
# UNIVRSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES

ACATLÁN

*"Selección de Características usando un Algoritmo Genético para problemas de Minería de Datos en Weka"*

TESIS Y EXAMEN PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE

**Licenciada en Matemáticas Aplicadas y Computación**

PRESENTA

*Zureyma Alejandra Flores Rodríguez*

*Asesor: Dra. Katya Rodríguez Vázquez*

Marzo, 2014.

*Para mi mamá Petra, el Ángel que me guía,*
*y para mi papá Alejandro,*
*a quienes quiero con todo mi corazón.*

# AGRADECIMIENTOS

*A Dios padre.*

A mi asesor de tesis, la *Dra. Katya Rodríguez Vázquez* por la orientación, el apoyo y la disposición que me brindó en todo momento para la realización de esta tesis. Ha sido un honor desarrollar a bien este trabajo de investigación.

A la *Universidad Nacional Autónoma de México* por haberme permitido ser parte de su planilla estudiantil y haberme otorgado todos los conocimientos necesarios para la realización de este trabajo.

A mi esposo, *Dmitry Slabko*, que sin su apoyo incondicional difícilmente hubiera podido llegar hasta este momento final.

Y a mi querido padre, *Alejandro Flores Álvarez* que, a través de sus incansables ánimos y apoyo, ha sido la inspiración principal para que el día de hoy pueda presentar este trabajo de tesis.

*¡Muchas Gracias!*

## ABSTRACT.

Hoy en día, la mayoría de las empresas cuentan con grandes repositorios de información, gracias a la facilidad de la obtención de los datos y a la diversidad de los dispositivos de entrada para conseguirlos. Las compañías alrededor del mundo, tienen contenedores con miles o millones de datos referentes al giro del negocio, de donde se puede obtener información o conocimiento latente, como soporte tanto para los procesos internos del negocio, como para los externos. Sin embargo, dada la magnitud de los datos, su manipulación, visualización y aprovechamiento es casi imposible, si no se cuenta con herramientas que ayuden en el manejo y uso de los mismos.

Así, el proceso de Extracción de Conocimiento de las Bases de Datos (KDD, del inglés Knowledge Discovery in Databases), nació con la necesidad de utilizar diversas técnicas matemáticas para analizar ese mundo de datos, con el fin de que aporten conocimiento útil en la toma de decisiones de las diversas áreas de una compañía. Dentro del proceso de KDD pueden existir las siguientes fases: Integración y Recopilación; Selección, Limpieza y Transformación; Minería de Datos; Evaluación e Interpretación; y Difusión y Uso. El presente trabajo de tesis se enfoca en la tarea de *Selección de Atributos*, dentro de la segunda fase del proceso de KDD, que extrae los atributos más apropiados de la base de datos**,** en relación a una meta definida.

A continuación se presenta un sistema programado en Java de una técnica para seleccionar características dentro de una BD, inspirada en la Teoría de la Evolución de las especies de Darwin. Esta técnica, llamada Algoritmos Genéticos, ha demostrado ser muy eficiente y eficaz en la búsqueda de soluciones a problemas complejos, donde el espacio de soluciones es muy grande y no existe una definición determinística para obtener el resultado deseado, y si la hay, es muy costosa. Los Algoritmos Genéticos son una clase de algoritmos pertenecientes al paradigma de Computación Evolutiva, y tienen como objetivo simular la evolución 'natural' de conjuntos de soluciones, usando la aleatoriedad como su herramienta principal.

La importancia de la fase de Selección, Limpieza y Transformación, radica en buscar que la calidad de los datos seleccionados sea la adecuada para obtener reglas o patrones valiosos de información, al aplicar un algoritmo de Minería de Datos. En particular, es necesario realizar una selección de datos, dado que la mayoría de los algoritmos de minería de datos, apenas pueden lidiar con un modesto número de 10 atributos y los recursos de la mayoría de los sistemas son limitados.

La Tarea de *Selección de Atributos* busca disminuir la cardinalidad de la 'vista minable', que se usará como materia prima del algoritmo de Minería de Datos. Por tanto, gracias a esta tarea, aumenta la precisión de la clasificación o predicción de dichos algoritmos, se reduce su tiempo de ejecución y el consumo de los recursos computaciones y permiten obtener patrones de conocimiento más comprensibles, acertados y coherentes.

# ÍNDICE GENERAL

# ÍNDICE DE FIGURAS.

# ÍNDICE DE TABLAS.

## NOMENCLATURA.

| | |
|---|---|
| ADN | Ácido Desoxirribonucleico |
| AE's | Algoritmos Evolutivos |
| AG | Algoritmo Genético |
| BD | Bases de Datos |
| BMV | Bolsa Mexicana de Valores |
| CE | Computación Evolutiva |
| DM | Data Mining – Minería de Datos |
| DW | Data Warehouse – Mercado de Datos |
| EM | Expectation Maximization<br>Algoritmo de Maximización de la Esperanza |
| ES's | Evolutive Strategies – Estrategias Evolutivas |
| ETL | Extraction, Transformation, Load System<br>Sistema de Extracción, Transformación y Carga |
| FN | Falsos Negativos |
| FP | Falsos Positivos |
| FS | Feature Selection – Selección de Características |
| IA | Inteligencia Artificial |
| ID | Identifier – Identificador |
| IDE | Integrated Development Environment<br>Ambiente de Desarrollo Integrado |
| KDD | Knowledge Discovery in Databases<br>Extracción de Conocimiento de las Bases de Datos |
| MDL | Minimun Description Length<br>Longitud de Descripción Mínima |
| M5P | Árbol de decisión con poda, M5 |
| MOLAP | Multidimencional OLAP – OLAP Multidimensional |
| NBTree | Árbol de Decisión con clasificadores Naïve Bayes en las hojas |
| OLAP | Online Analytical Processing<br>Procesamiento Analítico en Línea |
| PE | Programación Evolutiva |
| ROLAP | Relational OLAP – OLAP Relacional |
| SQL | Structured Query Language<br>Lenguaje de Consulta Estructurado |
| VN | Verdaderos Negativos |
| VP | Verdaderos Positivos |
| WEKA | Waikato Environment for Knowledge Analysis<br>Ambiente de Análisis del Conocimiento de Waikato |

# CAPÍTULO I: INTRODUCCIÓN.

El conocimiento obtenido a través de un conjunto de datos es importante en la medida en que puede ayudar a vislumbrar y comprender el comportamiento de ella misma y del entorno y así, favorecer el proceso de la toma de decisiones. Muchas empresas, instituciones u organizaciones tienen dentro de sus archivos grandes cantidades de datos de diversa procedencia, los cuales representan valiosos patrones de información oculta e inservible hasta que no sea extraída, examinada y decodificada, de acuerdo a una meta establecida.

La relación que existe entre el grupo de técnicas de aprendizaje durante el proceso completo de extracción de información, y el paradigma de la Computación Evolutiva y los métodos asociados a ésta, es muy estrecha, dado que estos últimos ofrecen soluciones confiables y robustas a muchos de los problemas que se encaran en el proceso de Extracción de Conocimiento de Bases de Datos. Hoy en día, existe una gran necesidad de aprender a manejar datos, interpretarlos, y tomar decisiones con base a dichas interpretaciones; sin embargo, la cantidad y diversidad de estos datos complica la aplicación de técnicas matemáticas determinísticas y computacionales tradicionales dando cabida a que se exploren los diversos paradigmas de la Computación Evolutiva para la solución de dichos problemas. Dentro de las líneas de investigación de la minería de datos se encuentran, entre otros, los Algoritmos Genéticos. Los algoritmos genéticos son preferentemente utilizados como métodos de búsqueda de soluciones óptimas, y han sido empleados con éxito en la solución de problemas de optimización combinatoria, optimización de funciones reales y como mecanismos de aprendizaje de máquina, entre otros.

Así, la computación evolutiva es un área de investigación dentro de las Ciencias de la Computación que trata de resolver problemas inspirándose en el proceso de la evolución natural y los modelos de los Procesos Evolutivos. La *Computación Evolutiva* es la conjunción entre la ciencia y la concepción suprema como Dios la concibió. Empieza a

finales de los 1950's y principios de los 60's y abarca los campos de investigación: "Programación Evolutiva", "Estrategias de Evolución" y "Algoritmos Genéticos". [Fogel 1997]

La aplicación de las técnicas de Computación Evolutiva dentro del proceso de extracción de conocimiento de las bases de datos puede ser muy diversa. Los algoritmos evolutivos han demostrado tener un buen desempeño tanto en los métodos de la preparación de datos, como en la búsqueda de patrones de información desde una vista minable. Lo anterior es gracias a su habilidad de explorar todo el espacio de soluciones, a través de saltos aleatorios, evitando caer en máximos o mínimos locales; así como, por su característica de evolucionar sin la necesidad de tener una amplia información acerca del problema en específico. Entre algunas aplicaciones relacionadas se encuentran: 'Algoritmos evolutivos para el descubrimiento de reglas de predicción en la mejora de sistemas educativos adaptativos basados en Web', 'Diseño e implementación de un modelo de aprendizaje mediante Algoritmos Genéticos', 'Entorno para la extracción de conocimiento basado en Algoritmos de aprendizaje genético y evolutivo', 'Algoritmos Evolutivos para el descubrimiento de reglas de predicción', 'Programación genética para el descubrimiento de reglas de predicción', 'Algoritmos evolutivos para clustering', 'Algoritmos evolutivos para la selección de características', 'Algoritmos evolutivos de la ponderación de atributos', 'Programación genética para la construcción de atributos' y 'Algoritmos evolutivos para el descubrimiento de reglas difusas'. [Freitas 2002]

La presente tesis se enfoca en la tarea de la "Selección de Características" mediante un Algoritmo Genético, tarea que pertenece a la Preparación de los Datos de la segunda fase del Proceso de Extracción de Conocimiento dentro de las bases de datos, la cual es llamada "Limpieza, Transformación, Exploración y Selección".

## 1.1 MOTIVACIÓN.

El presente trabajo es una oportunidad de explorar las áreas de estudio concernientes a la Computación Evolutiva y Minería de Datos. La primera, una rama de las ciencias de la

computación que ha sido desarrollada en los últimos 60 años y que ofrece alternativas diferentes, y a nuestro nivel muy poco exploradas, en la solución de problemas de la vida real.

El segundo campo, también nombrado Minería de Datos, Análisis Predictivo, Análisis del Negocio, Inteligencia del Negocio ó Ciencia de la Información; pertenece al proceso de Extracción de Conocimiento de las Bases de Datos, KDD *(Knowledge Discovery in Databases)*, y surge gracias a la necesidad del manejo y aprovechamiento de los datos, así como del actual incremento exponencial de la información. Dicho proceso, contiene un enfoque multidisciplinario y muy completo, el cuál además se encuentra en pleno desarrollo. Los investigadores de este campo llevan en acción apenas 18 años, sin embargo ha tenido un desarrollo acelerado debido a lo rápido que han evolucionado últimamente la mayoría de las ciencias computacionales. Así, este campo promete un gran crecimiento con un amplio grado de aplicación en muchos sectores: académico, empresarial, financieros, de servicios, escolar, médico, etc.

## 1.2 PROBLEMA DE INVESTIGACIÓN.

Teniendo $\boldsymbol{n}$ atributos en una tabla de datos numérica, limpia y sin ruido, encontrar a través de un Algoritmo Genético aquel subconjunto de $\boldsymbol{n}$, digamos $\boldsymbol{A}$, donde $\boldsymbol{A_j}$ es el individuo $\boldsymbol{j}$ que toma los valores de $[\boldsymbol{1}, \boldsymbol{m}]$, y $\boldsymbol{m}$ es el número total de individuos de la población del AG; tal que $\boldsymbol{A_j(k) = i}$ sea el subconjunto de atributos relevantes y no redundantes de la vista minable, que mejor clasifique al atributo de clase booleano $\boldsymbol{C}$, para la mayoría de las observaciones de una base de datos, en los conjuntos de entrenamiento y prueba. Siendo $\boldsymbol{k}$ la posición dentro del individuo $\boldsymbol{A_j}$ que toma los valores de $\boldsymbol{\{1, \ldots, p\}}$ con $\boldsymbol{p = \{ 6, 8, 10 \}}$, e $\boldsymbol{i}$ es la codificación del atributo que toma los valores de $\boldsymbol{\{1, n-1\}}$.

## 1.3 JUSTIFICACIÓN.

Hoy en día, los repositorios de información con los que cuentan las instituciones y empresas de cualquier rubro, crecen de manera exponencial. La facilidad de adquirir

cualquier tipo de información debido al constante desarrollo de la tecnología, permite que ahora las empresas y compañías tengan mundos de información de todo tipo. Sin embargo, lo anterior, en vez de representar un beneficio se esta convirtiendo en un problema real, ya que las compañías aunque tienen mucha información, debido a la cantidad y a los problemas de acceso a dichos repositorios de información, son incapaces de extraer por sí mismas conocimiento real y valioso que pueda ayudar a un problema en específico. [Hernández-Orallo et al. 2007] El proceso de KDD, como comúnmente se le conoce, trata de dar solución a este problema de una manera muy específica y sistemática, en donde se trata de utilizar diferentes técnicas tanto matemáticas como computacionales para obtener conocimiento, desde un conjunto de datos, que realmente sea el adecuado en relación a un problema en específico. Como se estudiará en la sección 2.2.1 (página 22), la metodología de KDD no es la única, empero es la que seguiremos en este trabajo.

La presente Tesis es una investigación que propone utilizar una de las técnicas del Cómputo Evolutivo, los Algoritmos Genéticos [Goldberg 1989], al aplicarla a una de las fases del proceso de Extracción de Conocimiento en las Bases de Datos (KDD), la cual es la de Preparación de los Datos, en donde se busca seleccionar los atributos o características más importantes y no redundantes del almacén de datos, para crear una *Vista Minable* que será utilizada por algún algoritmo de Minería de Datos en el banco de herramientas de software libre, 'Weka', [Frank et al. 2005]. El uso de los Algoritmos Genéticos, como algoritmo de selección de características, está basado en su capacidad eficiente de búsqueda aleatoria sobre el espacio de soluciones, cuándo éste cuenta con un amplio número de variables.

Una de las razones más importantes que justifican la presente investigación, es la 'Maldición de la Dimensionalidad', la cual expresa que las técnicas para el análisis de la información totalmente eficientes para bajas dimensiones, no pueden proveer ningún resultado significativo cuando el número de registros va más allá de un modesto tamaño de 10 atributos. [Chizi & Maimon 2005]

La importancia de realizar una selección de características radica en las siguientes razones: 1) La necesidad de mejorar la calidad de los datos que se proporcionarán como materia

prima dentro de un algoritmo de Minería de Datos, para obtener reglas o patrones coherentes; 2) cuando se busca patrones significativos en las bases de datos, el proceso de Minería de Datos requiere un alto costo computacional cuando manipula grandes conjuntos de datos ocupando los recursos del sistema; 3) Ayuda a lidiar con la maldición de la dimensionalidad, que constituye un serio obstáculo para la eficiencia de la mayoría de los algoritmos de Minería de Datos.

El problema de tener muchos atributos en una vista minable es que la gran mayoría de los métodos de minería de datos pueden perderse entre tantas características en un espacio que, al tener alta dimensionalidad, resulta estar más desierto y obtiene modelos ajustados a las particularidades de los datos de entrenamiento y no de los datos en general. Sin embargo, la gran ventaja de los métodos de reducción de dimensionalidad a través de la Selección de Características, es que el conjunto final de características es un subconjunto de los atributos originales y, por tanto, los modelos extraídos con posterioridad se definirán en función de los atributos originales, sin perder la comprensibilidad del modelo.

## 1.4 HIPÓTESIS.

Los Algoritmos Genéticos son algoritmos robustos que han demostrado ser, en promedio, bastante eficientes, en una gran variedad de problemas. Dado este antecedente, se plantean como mecanismos o herramientas para ser implementados en la fase de Selección de Características dentro del proceso de Extracción de Conocimiento de una Base de Datos, llamada Selección de Características (Feature Selection). [Chizi & Maimon 2005]

## 1.5 OBJETIVOS.

### 1.5.1 Objetivo General.

Mostrar mediante un sistema programado en Java, que la Computación Evolutiva a través de su rama llamada 'Algoritmos Genéticos', es una herramienta capaz de producir resultados eficientes en la Selección de Atributos o Características, dentro de la fase de 'Preparación de los Datos' para la extracción de conocimiento de una Base de Datos.

### 1.5.2 Objetivos Particulares.

1. Estudiar el problema de la Reducción de la Dimensionalidad, a través de la Selección de Características de una base de datos, para crear una Vista Minable adecuada que nutra a la fase de Minería de Datos, dentro del proceso de extracción de conocimiento de una base de datos (KDD).
2. Estudiar la técnica de los Algoritmos Genéticos dentro del paradigma de la Computación Evolutiva, de forma amplia y concreta.
3. Desarrollar una herramienta basada en los Algoritmos Genéticos, para resolver el problema de Selección de Características.
4. Aplicar un algoritmo de Minería de Datos, mediante el banco de herramientas de Weka, a la solución óptima obtenida del AG presentado.

## 1.6 GUÍA DE LA TESIS:

Capítulo I – Introducción - Se presentará un panorama general del presente trabajo de tesis, así como su orientación, a través del tiempo y la importancia del mismo en el área de las ciencias computacionales. Se explicarán las razones por las cuáles se esta abordando este tema en particular, así como los objetivos que se pretenden alcanzar con la realización de este trabajo de tesis y la presentación de la hipótesis a confrontar.

Capítulo II – Marco Teórico: En este capítulo se presentará la base teórica completa sobre la cuál se respalda este trabajo. El problema en sí se trata de resolver con una técnica perteneciente al paradigma de Computación Evolutiva que, a su vez se encuentra dentro del área de conocimiento de la Inteligencia Artificial. Además, se explicará a grandes rasgos el proceso completo de la extracción de conocimiento de las bases de datos, haciéndose énfasis en la segunda fase, la cual es objeto de estudio de la presente tesis.

Capítulo III – Desarrollo del Modelo: Se describe el problema a resolver por el presente trabajo de tesis, así como todos los parámetros que se utilizaron para definir y crear el

modelo, el cual implementará un Algoritmo Genético sobre la fase de Selección de Características del proceso de extracción de conocimiento de las bases de datos.

Capítulo IV –Recopilación, Integración y Limpieza de los datos: En este capítulo se describen todas las técnicas y medidas utilizadas en la preparación de los datos de cada base de datos, que servirán para probar el sistema aquí propuesto.

Capítulo V - Evaluación del Modelo y Resultados: En este capítulo se muestran y desarrollan todas las pruebas hechas al modelo anteriormente creado, así como la evaluación del mismo, y se presentan de manera detallada los resultados de las pruebas previamente hechas con las bases de datos obtenidas en el capítulo anterior. Además, se comparan las soluciones con las obtenidas por Weka, para la selección de características para validar el modelo.

Capítulo VI – Discusión y Conclusiones: En este capítulo se discutirán los resultados obtenidos de las evaluaciones realizadas en el capítulo anterior, así como las implicaciones del modelo. Se contrastarán los resultados con los objetivos deseados y los estándares conocidos previamente; así como, las conclusiones a las cuáles se ha llegado, una vez hecha la evaluación del modelo creado, sugerencias y alcances del mismo para trabajos futuros sobre el tema. También, se aplican diversos algoritmos de Minería de Datos del banco de herramientas de Weka (Anexo B), a las soluciones óptimas obtenidas por el AG.

Anexos: Se presentan 5 anexos donde se abordarán los temas relacionados con la tesis, hasta cierto nivel de detalle, que no son propios del trabajo de investigación, cómo son: el código en Java del sistema 'Attribute_Classification' (Anexo A), el banco de trabajo 'Weka' (Anexo B), el Análisis de Correlación (Anexo C) y la Teoría sobre Portafolios de Inversión (Anexo D). Además, se presenta un Glosario de Términos en el Anexo E y los Diccionarios de Datos de las tres bases de datos en el Anexo F.

# CAPÍTULO II: MARCO TEÓRICO.

## 2.1 COMPUTACIÓN EVOLUTIVA.

Es una rama de las Ciencias de la Computación llamada *Inteligencia Artificial (IA)*, la IA tiene como principal objetivo desarrollar el "Aprendizaje de Máquina" y se define como el campo que estudia y diseña los agentes inteligentes, donde un *agente inteligente* es un sistema que percibe su entorno y toma acciones para maximizar (optimizar) las posibilidades de su éxito.

La Computación Evolutiva (CE), en inglés *Evolutionary Computation*, empieza a finales de los 50's y principios de los 60's como una serie de experimentos aislados para modelar el proceso evolutivo; pero no es sino hasta los años 90´s cuando empieza a madurar y a definirse propiamente como la conocemos hoy en día. Este paradigma, consiste de a*lgoritmos de búsqueda estocástica* que están basados en abstracciones del proceso de Evolución Darwiniano [Bäck 2000a; De Jong 2000; De Jong et al. 2000].

Según [Fogel 1997] los esfuerzos por orientarse en la Computación Evolutiva comúnmente derivan, principalmente de las siguientes motivaciones: ***i)*** **Mejoramiento de la Optimización**: Dado que la evolución es un proceso de optimización que no implica perfección pero que si puede descubrir soluciones funcionales muy precisas para diversos problemas de la vida real. ***ii)*** **Adaptación Robusta**: El mundo verdadero nunca es estático, y los problemas de optimización donde el tiempo es variable, son de los más difíciles. A través de sus estrategias de búsqueda genética, [Holland 1975] enfatiza la flexibilidad de estos algoritmos para ajustarse a la ejecución basada sobre la retroalimentación de su medio ambiente. ***iii)*** **Inteligencia Mecánica**: La inteligencia puede ser definida como la capacidad de un sistema para adaptar su comportamiento y así, alcanzar ciertas metas en un rango de entornos [Fogel 1997]. El comportamiento inteligente entonces requiere predicción, para adaptarse a futuras circunstancias y tomar decisiones adecuadas. Tomando a los seres vivos como ejemplo, estos modelos se enfocan en las reglas que estos seres siguen, o sus conexiones neurológicas, para simular la evolución sobre una clase de algoritmos

predictivos, lo que es una aproximación alternativa para generar inteligencia mecánica. ***iv)*** **Facilitación de un gran entendimiento de Biología**: Misma que sirve de inspiración en el desarrollo de sus modelos, existiendo un deseo de capturar la esencia de la evolución en una simulación computarizada y usar dicha simulación para obtener nuevos conocimientos en la física de los procesos evolutivos naturales [Ray 1991]. El éxito de este paradigma plantea la posibilidad de estudiar los sistemas biológicos alternativos que son meramente imágenes plausibles de lo que la vida debe ser de alguna forma.

La CE (por sus siglas en inglés) abarca las áreas de conocimiento llamadas *Programación Evolutiva, Estrategias de Evolución y Algoritmos Genéticos*, las cuales tienen en común el desarrollo de los siguientes procesos: la recombinación genética, variación aleatoria, competición y selección de contendientes individuales, a través de la evaluación de su aptitud dentro de una población de individuos, o soluciones candidatas, lo cual forma la esencia de la evolución. Una vez que estos procesos están en su lugar, el algoritmo evolucionará inevitablemente. Además, gracias a su factor aleatorio al aplicar los operadores de selección, cruza o mutación, ayuda a evitar que caiga en máximos o mínimos locales al encontrar la solución óptima.

Un Algoritmo Evolutivo mantiene una población de individuos, siendo cada uno una solución candidata a un problema dado. Además, desarrollan ciertos operadores como el de Cruza y Mutación, los cuáles son estocásticos y gracias a ellos, el individuo de mejor calidad podrá copiar con mayor probabilidad su material genético a la siguiente generación. Para obtener más información de este campo de estudio, se recomienda consultar [De Jong et al. 2000, Eshelman 1997, Rudolph 1997, Porto 1997].

### 2.1.1 Programación Evolutiva.

La Programación Evolutiva (PE) pertenece a la clase de paradigmas para simular la evolución y los conceptos de Darwin. En un principio, fue desarrollada para evolucionar máquinas de estado finito; hoy en día, su objetivo principal es generar, incremental e iterativamente, soluciones apropiadas desde un entorno de cambios estáticos y dinámicos, a

través de vectores de valores reales. En lugar de desarrollar un conjunto complejo de reglas que son derivadas de los expertos humanos, como son en las heurísticas, PE desarrolla un conjunto de soluciones que exhibe el comportamiento óptimo bajo un entorno y una función de pago deseada. Dicho de otra forma, la Programación Evolutiva puede ser considerada como una técnica de optimización, donde los algoritmos obtienen comportamientos, parámetros y otras construcciones óptimas. Como en otros algoritmos, es importante entender que el punto de optimalidad es completamente independiente del algoritmo de búsqueda y es únicamente determinado por la topografía adaptativa [Porto 1997]. En su forma estándar, un programa evolutivo utiliza cuatro componentes principales: Inicialización, variación, evaluación y selección.

El padre de la PE, en inglés *(Evolutionary Programming)* es Lawrence J. Fogel. Alrededor de 1960, la Inteligencia Artificial estuvo principalmente concentrada en Heurística y en la simulación de Redes Neuronales Primitivas. Para Fogel fue claro que esas aproximaciones son limitadas porque modelan sólo a los humanos en lugar del proceso esencial que produce criaturas con incremento del intelecto: Evolución. Fogel consideró que la inteligencia está basada en comportamientos adaptativos para alcanzar metas en un rango de entornos. Por otro lado, la predicción fue vista como un ingrediente clave para el comportamiento inteligente y sugirió una serie de experimentos sobre el uso de la evolución simulada de *máquinas de estado – finito*, para pronosticar series de tiempo con respecto a criterios arbitrarios. [Fogel & Fogel 1986]

El problema evolutivo fue definido como el desarrollo de un algoritmo (esencialmente un programa) que podría operar sobre la secuencia de símbolos hasta el momento observado de tal manera que pueda producir un símbolo de salida, que es posiblemente para maximizar la ejecución del algoritmo, a la luz tanto del siguiente símbolo a aparecer en el entorno como de una función bien definida de rentabilidad. Las máquinas de estados finitos proveyeron una representación útil del comportamiento requerido.

La propuesta fue como sigue: Una población de máquinas de estado finito es expuesta al entorno, que es, la secuencia de símbolos que han sido observados durante el tiempo actual.

Para cada máquina pariente, como cada símbolo de entrada es ofrecido a la máquina, cada símbolo de salida es comparado con el siguiente símbolo de entrada. El valor de esta predicción es medido entonces, con respecto a la función de rentabilidad (ej. todo – nada, error absoluto, error cuadrático, o cualquier otra expresión del significado de los símbolos). Después de que la última predicción es hecha, una función de rentabilidad para cada símbolo indica la aptitud de la máquina, por ejemplo, la rentabilidad promedio por símbolo. [Fogel et al. 1996]

Las máquinas descendientes son creadas aleatoriamente mutando cada máquina pariente. Cada padre produce un descendiente (esto fue originalmente implementado como un solo descendiente simplemente, por conveniencia). Hay 5 posibles modos de mutación aleatoria que naturalmente resultan de la descripción de la máquina: cambiar un símbolo de salida, cambiar una transición del estado, añadir un estado, borrar un estado, o cambiar el estado inicial. Las acciones de 'borrar un estado' y 'cambiar el estado inicial' son sólo permitidas cuando la máquina padre tiene más de un estado. Las mutaciones son escogidas con respecto a la distribución de probabilidad, que es típicamente uniforme. El número de mutaciones por descendiente es también escogido con respecto a la distribución de probabilidad o quizá fijado 'a priori'. Esos descendientes son entonces evaluados sobre el entorno existente de la misma manera que sus Parientes. Otras mutaciones, tales como, el funcionamiento de acoplamiento de la lógica sobre tres o más máquinas, también fueron propuestos por Fogel pero no implementadas.

Las máquinas que proveen la mayor rentabilidad son retenidas para convertirse en parientes de la siguiente generación. Típicamente, la mitad del total de las máquinas fueron guardadas, así que la población de parientes permaneció como un tamaño constante. Este proceso es iterado hasta que una predicción actual del siguiente símbolo, hasta ahora inexperto en el entorno, es requerido. La mejor máquina genera esta predicción, el nuevo símbolo es añadido al ambiente experimental y el procedimiento es repetido. Fogel usó la evolución *'no regresiva'*. Para ser conservada, una máquina tiene que clasificarse en la mejor mitad de la población. Este procedimiento general fue exitosamente aplicado a problemas de predicción, identificación y control automático y fue extendido a la

simulación de la co-evolución de poblaciones. El diagrama de flujo de la programación evolutiva es muy similar al de los algoritmos genéticos (figura 2.2, página 17), con las diferencias de que únicamente se tiene el operador de Mutación a nivel individuo y su codificación son máquinas de estado finito (ver Anexo E).

### 2.1.2 Estrategias Evolutivas.

Las Estrategias Evolutivas, (ES's) por sus siglas en inglés *(Evolution Strategies)* fueron desarrolladas en 1964 por un grupo de estudiantes alemanes, encabezados por Ingo Rechenberg, de la Universidad de Berlín, Alemania. Su objetivo principal era resolver problemas de mecánica de fluidos de alto grado de dificultad. [Rechenberg 1965]

A grandes rasgos, en la versión original de esta técnica *Estrategia (1+1)*, el *arquetipo* de las ES's tomaba la siguiente forma: Un individuo *a* (padre) consistiendo de un elemento $X \in \mathbb{R}^n$ es mutado añadiendo un vector aleatorio con distribución normal $Z \sim \mathcal{N}(0, l_n)$ que es multiplicado por un escalar $\sigma > 0$, ($l_n$ denota una matriz unitaria con rango *n*). El nuevo punto (hijo) es aceptado si es mejor o igual que el viejo (si el hijo es mejor o igual al padre), de otra forma, el viejo punto pasa a la siguiente iteración. La decisión para la selección está basada en una simple comparación entre los valores de los puntos viejo y nuevo, de la función objetivo; a este tipo de selección se le llama *extintiva*, porque los peores individuos tienen una probabilidad *0* de ser seleccionados. [Rudolph 1997]. Su diagrama de flujo es muy similar al de la figura 2.2, de la página 17, con la diferencia que su codificación es Real, y su operador principal originalmente fue la mutación, aunque después se implementó la cruza.

Asumiendo que la función objetivo $f: \mathbb{R}^n \longrightarrow \mathbb{R}$ será minimizada, la ES más simple, empezando en algún punto $X_0 \in \mathbb{R}^n$, es determinada por el siguiente esquema iterativo:

$$X_{(t+1)} = \begin{cases} X_t + \sigma_t Z_t & if \quad f(X_t + \sigma_t Z_t) \leq f(X_t) \\ X_t & de\ otra\ forma \end{cases}$$

donde $t \in \mathbb{N}_0$ denota el contador de la iteración y donde $Z_t : t \geq 0$ es una secuencia de vectores aleatorios con distribución normal estándar independientes e idénticamente distribuidos.

La mayoría de los métodos relacionados a las ES's difieren en el mecanismo para ajustar el parámetro $\sigma_l$ , que es usado para controlar la fortaleza de la mutación. La solución de Rechenberg para controlar dicho parámetro es conocida como "*La regla del éxito 1/5"*: La razón entre mutaciones exitosas y el total de mutaciones debe ser de 1/5; si es más, entonces debe incrementarse la desviación estándar y si es menos, entonces debe haber un decremento en ésta.

Básicamente, en esta técnica, los cromosomas se componen, al menos, de dos partes:

- Variables objeto: $x_1, \cdots, x_n$ .
- Los parámetros de la estrategia: $\sigma_1, \cdots, \sigma_{n_0}$ .

También, los operadores de recombinación de las ES's pueden ser de dos formas:

- *Sexuales:* el operador actúa sobre los individuos elegidos aleatoriamente de la población de padres.
- *Panmíticos:* se elige un sólo padre al azar, y se mantiene fijo hasta que se elige a un segundo padre (de entre toda la población) para cada componente de sus vectores.

Además de este tipo de estrategia, existen también algunas variaciones como las ES $(\mu + 1)$, $(\mu + \lambda)$ , $(\mu \cdot \lambda)$, y otras que son combinaciones de las anteriores. [De Jong et al. 2000].

### 2.1.3 Algoritmos Genéticos.

Los algoritmos genéticos (AG's) pertenecen a una clase de algoritmos evolutivos propuestos y analizados por primera vez por John Holland en 1975. Estos métodos heurísticos de búsqueda inspirados en lo que sabemos acerca del proceso de la evolución natural, son apropiados para resolver problemas donde el dominio de la solución pueda resultar demasiado extenso.

Los AG son métodos de optimización que están basados en la Teoría de la Evolución de Darwin que dice que: "La selección natural obra solamente mediante la conservación y acumulación de pequeñas modificaciones heredadas, provechosas todas al ser conservado", y dio nombre a este proceso: *"El origen de las especies, selección natural"* [De Jong et al. 2000]. Estas "modificaciones heredadas", señaladas por Darwin, como las generadoras de organismos mejores, son llamadas *mutaciones*. Un organismo mutante ha sufrido una modificación que lo hace diferente al resto de sus congéneres; esta modificación puede ser un inconveniente para él o, por otro lado, que le confiera alguna cualidad que le permita sobrevivir más fácilmente que al resto de los individuos de su especie. Con el tiempo, gracias a la competencia, los organismos que en un principio eran raros, se volverán comunes a costa de la desaparición del modelo anterior.

La principal cualidad del proceso natural de la evolución es la de generar organismos óptimos sobre los que influyen infinidad de variables. Estos algoritmos generan un conjunto de soluciones para un problema en términos de un nivel de aptitud '*fitness'*, es decir, una función de aptitud (la cuál nos indica que tan "buena" o "mala" es cierta respuesta). La eficacia del algoritmo será plenamente dependiente de los criterios que se consideren para la determinación de la '*aptitud'*, así como de su representación. Una característica que debe tener esta función es que debe ser capaz de "castigar" a las malas soluciones y de "premiar" a las buenas, de forma que sean estas últimas las que se propaguen con mayor rapidez. La función de aptitud no es más que la función objetivo de un problema de optimización.

Son principalmente tres los rasgos que diferencian a un AG Simple de otros algoritmos evolutivos: el primero es que son usadas cadenas de bits para representar a los individuos (posibles soluciones) de una población, el segundo es que su método de selección es la *selección proporcional*; y tercero, sus métodos principales para producir variaciones son la *mutación* y la *cruza*, siendo ésta última lo que hace distintivos a los AG's [Eshelman 1997]. A diferencia de los procedimientos matemáticos tradicionales, los AG's no suponen nada o casi nada acerca del problema a resolver, como es el caso de los métodos

tradicionales en los que se exige que el modelo matemático del problema consista de una función claramente definida y con ciertas propiedades. Además, gracias a la variación aleatoria, trabajan muy bien con problemas donde el número de variables es amplio, lo cual, desde el punto de vista determinístico, es una complicación para encontrar su solución.

Los AG's, al usar una analogía directa con el comportamiento natural, dado que trabajan con una población de individuos, deben decidir cuidadosamente que tipo de solución candidata será representada por un individuo y que función de aptitud será usada para evaluarlos. En la figura 2.1 indica de qué elementos se conforma usualmente un AG:

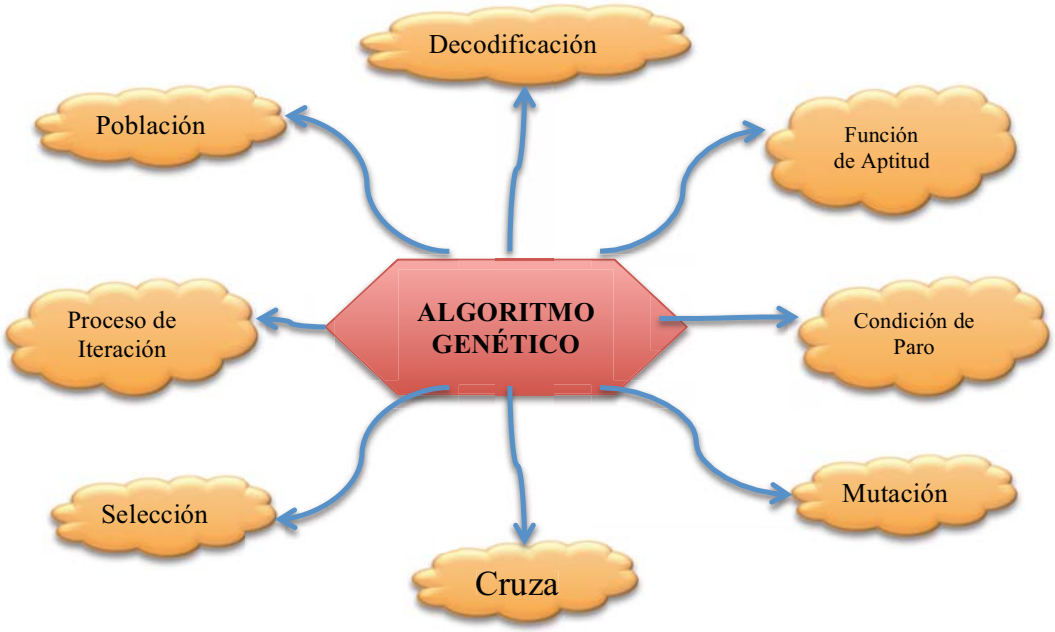



*Figura 2.1 – Elementos de un Algoritmo Genético.*

Un AG trabaja con un conjunto de individuos llamado *Población*. Cada estructura individual de la población, también llamado *Cromosoma*, codifica como *genotipo* mapeado en el ADN para ser manipulado por el algoritmo, las posibles características de las soluciones del problema en cuestión, conocidas como el *fenotipo*. Típicamente, pero no necesariamente, los cromosomas son cadenas de bits. El valor en cada lugar sobre la cadena de bits o *Gen* es referido como un *Alelo*. La representación de las soluciones candidatas suele ser muy importante, dado que debe permitir mapear absolutamente todas las posibles

soluciones, de tal forma, que sean fáciles de manipular por la computadora y al mismo tiempo descifrables.

Para iniciar el procedimiento, el AG crea una población inicial, la cual es evaluada por una función que asigna a cada cromosoma un valor de *aptitud*, también llamado *fitness*. Una vez que el algoritmo se cerciora de no haber llegado al máximo o mínimo deseado, se seleccionan los individuos de acuerdo a su fitness, dándole preferencia a los individuos mejor calificados de ser seleccionados y pasar a la siguiente población. Entre los métodos más comunes de *Selección* se encuentran: Ruleta, Estocástico Universal y Torneo. A continuación, se aplica el primer operador, la *Cruza*, que consiste en recombinar el material genético de dos individuos, para crear dos nuevos individuos. La probabilidad de recombinación sobre la población suele ser alta, 90% ó 95%, con lo cual se busca variar la información genética de cada población. Entre los métodos más destacados de recombinación, tenemos: cruza a uno o dos puntos, cruza multipunto o cruza uniforme. Después de la recombinación, se aplica el operador de *Mutación*, que consiste en cambiar aleatoriamente un alelo de un gen, con la intensión de recuperar material genético importante que pudiera perderse a través de las generaciones. Aunque este operador es también importante en el AG, su probabilidad de aplicación es muy baja, pues es apenas de 5% normalmente. Se pueden aplicar tantos operadores de cruza y mutación como se quiera. Una vez aplicados los operadores correspondientes, los individuos generados, así como los que han permanecido intactos, pasan a formar parte de una nueva población. A veces, puede existir *elitismo*, es decir, se asegura de que el mejor individuo de la población anterior pase directamente a la nueva población sin ningún cambio genético. A continuación, se evalúan nuevamente los individuos con el fin de saber si se ha encontrado la solución buscada y, de no ser así, se continúa con la siguiente iteración donde se repetirán los procesos anteriores hasta que se encuentre una solución efectiva o hasta que se llegue a un criterio de paro. Entre los parámetros a tomar en cuenta en un AG, se encuentran: el tamaño de la población, el criterio de paro del AG y las probabilidades de cruza y mutación. Para ilustrar este procedimiento, en la figura 2.2 se muestra el diagrama de flujo de un AG.

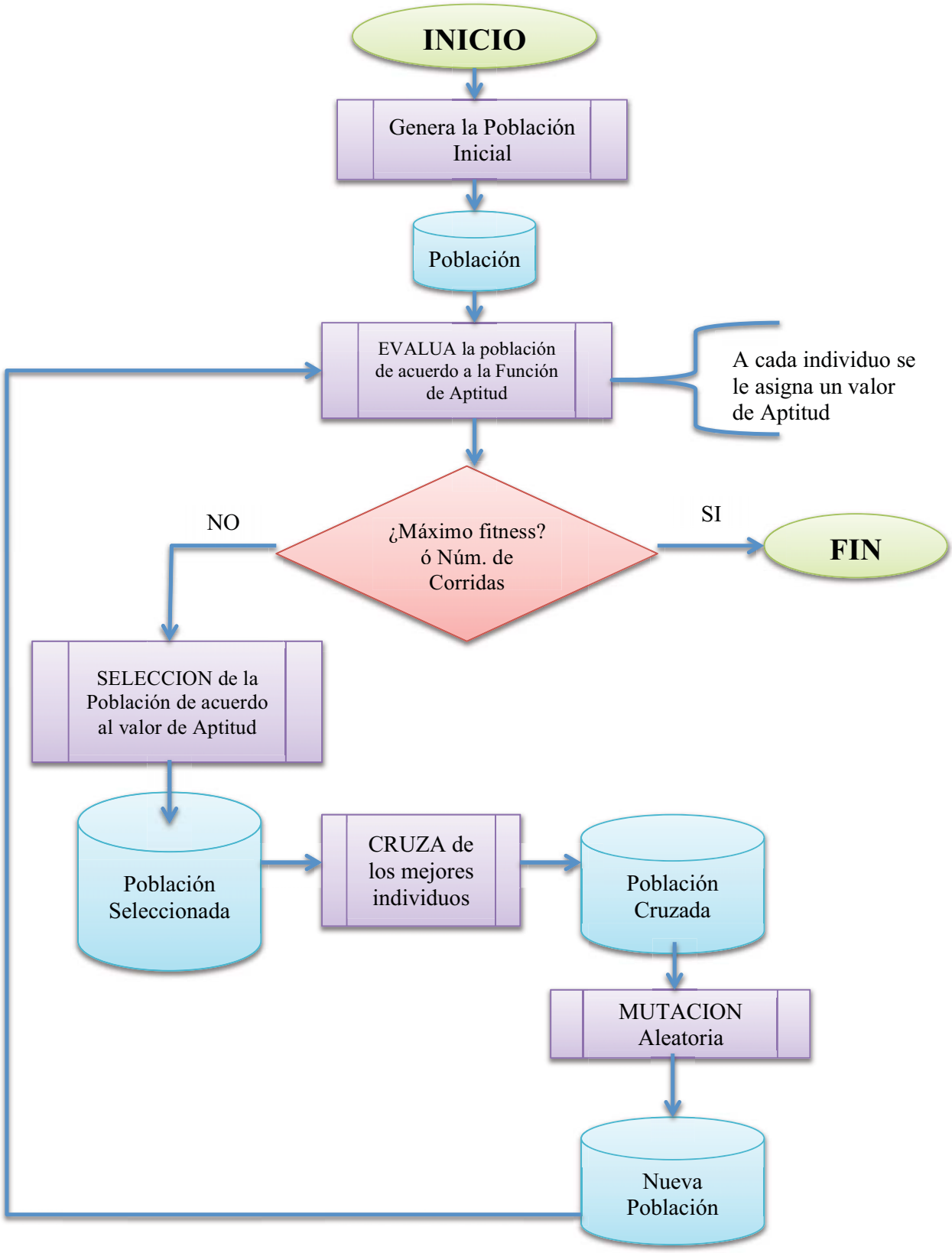



*Figura 2.2 – Diagrama de Flujo de un AG.*

En el capítulo III, que corresponde a la descripción del modelo, se hablará más del AG propuesto que se empleará para seleccionar las características más importantes dentro de una base de datos.

### 2.1.4 Aplicaciones de los Algoritmos Evolutivos.

Puesto que la Computación Evolutiva es un paradigma que propone soluciones confiables a problemas de optimización, su campo de aplicación es bastante extenso. Así, podemos encontrar aplicaciones en diferentes áreas del conocimiento como el problema del agente viajero y todos sus derivados, o planear la trayectoria o navegación que un robot debe seguir. Otro ejemplo muy común, es la programación de cierto número de eventos o actividades durante un periodo de tiempo, donde cada actividad necesita diferente cantidad de recursos y tiempo. Los algoritmos evolutivos también han sido aplicados a problemas de la mochila y al diseño de sistemas digitales y electrónicos que implementen una frecuencia de respuestas deseadas, sistemas de procesamiento de señal, diseños de circuitos integrados y diseño de topología de redes. [Beasley 1997]

En computación, también se han usado en el diseño de autómatas celulares para coordinar los procesos globales con mecanismos apropiados de comunicación, al ejecutarlos en computación paralela. En el área de simulación, los algoritmos evolutivos son usados para determinar cómo se comportará un sistema, o para probar la precisión de algún modelo, por ejemplo, en los campos de química, biología, medicina, economía, física, etc.

En el área de control, estos algoritmos son usados para diseñar el controlador de un sistema, o trabajan como rutinas de optimización dentro del mismo proceso de control. Asimismo, existe un vasto campo de aplicación en los sistemas de clasificación, que usualmente son el corazón de muchos otros tipos de sistemas; como ejemplos tenemos, los juegos de computadora, torneos, maniobras de combate aéreo o en procesamiento de imagen.

## 2.2 EXTRACCIÓN DE CONOCIMIENTO DE LAS BASES DE DATOS (KDD).

El conocimiento obtenido por cualquier tipo de organización o institución es importante en la medida en que puede ayudar a conocer y comprender el comportamiento de ésta y de su entorno, favoreciendo la toma de decisiones. Anteriormente, las decisiones en una institución o empresa eran tomadas de acuerdo al giro de la misma y a la experiencia e intuición personal de sus directivos, etc. Hoy en día, también sabemos que para ejercer una toma de decisiones adecuada de acuerdo a ciertas metas existen, además de los ya mencionados, muchos otros factores en juego, tanto externos como internos. Entre los externos podemos mencionar la competencia con otras empresas e instituciones del mismo giro, la situación económica de la entidad a la cual es dirigido cierto producto, la demanda y la oferta, etc. Dentro de los factores internos, sin duda podemos señalar toda la información que se ha recolectado a través de los años y, que dentro de sus datos, yace una cantidad de conocimiento que puede ser de gran utilidad para el mismo proceso de la Toma de Decisiones.

En la actualidad, es cada vez más común ver la obtención de conocimiento útil y valioso a partir de los enormes contenedores de datos que se posee. Lo anterior, crea la oportunidad y necesidad de semi-automatizar los métodos que descubren cierto conocimiento. Dicho conocimiento es adquirido a través de un proceso estructurado que puede ser explicado bajo diferentes metodologías.

En el presente trabajo se explora una de ellas, conocida en el mundo de habla inglesa como *Knowledge Discovery in Databases, KDD (Descubrimiento del Conocimiento en las Bases de Datos).* En [Hernández-Orallo et al. 2007] se define al KDD como "el proceso de extracción no trivial de información potencialmente útil de las grandes bases de datos". El KDD, que tiene un enfoque multidisciplinario, es un análisis complejo, exploratorio, automático e iterativo que además de obtener modelos o patrones a través de los datos (Minería de Datos), se encarga también de la preparación de los mismos y de la interpretación de los resultados. Una definición más detallada se muestra en [Fayyad et al. 1996a] como "un proceso no trivial de identificar patrones válidos, novedosos,

potencialmente útiles y, en última instancia, comprensibles a partir de los datos". En la siguiente figura se ilustra este razonamiento:

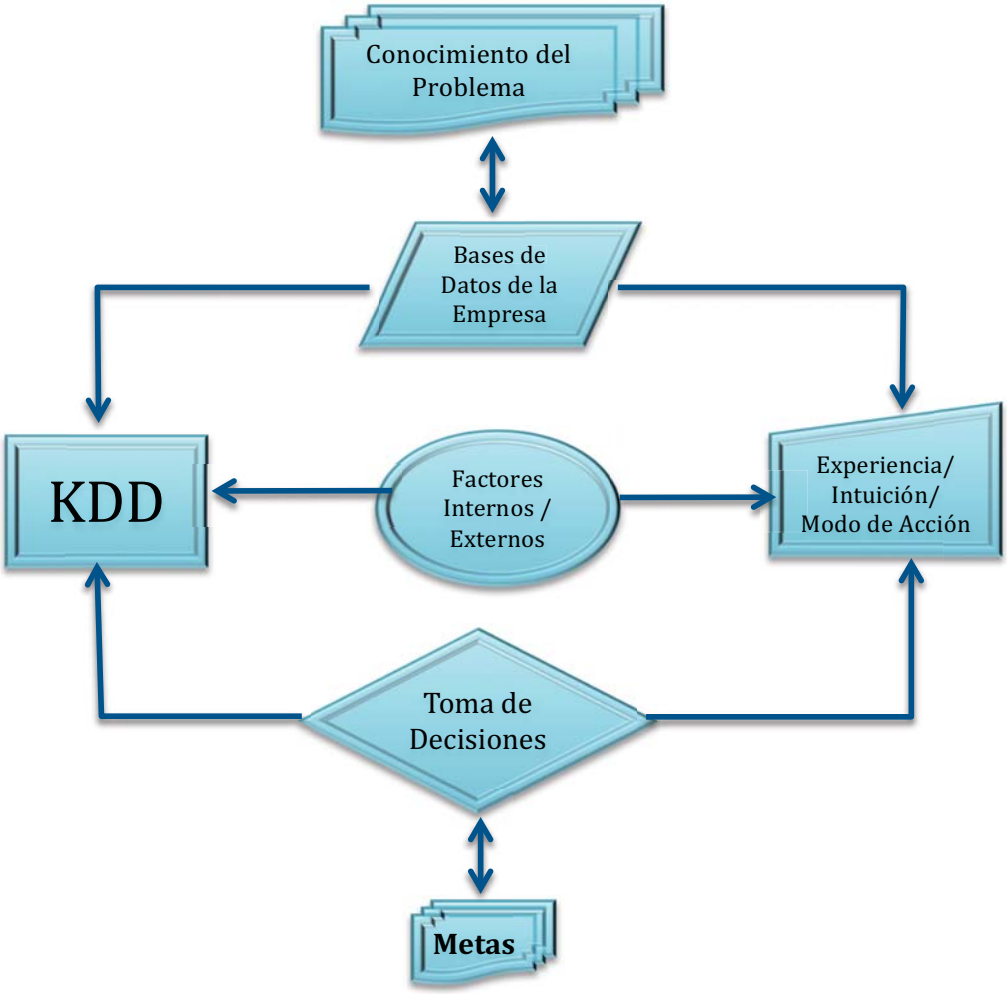



*Figura 2.3 – Proceso de Toma de Decisiones.*

Según las características mencionadas anteriormente acerca de los patrones de datos que se extraen con el KDD, según [Fayyad et al. 1996a], se dice que deben ser **válidos ó precisos** porque dichos patrones deben seguir siendo correctos para los datos nuevos y no sólo para los datos de prueba. **Novedosos**, porque aportan conocimiento desconocido o nuevo en relación a los datos de entrada. **Potencialmente útiles**, porque dicho conocimiento debe contribuir en cierta medida a los procesos de toma de decisiones reportando algún tipo de beneficio para la compañía; y por último, **comprensibles**, es decir, que sea conocimiento coherente para facilitar su interpretación e incorporación en algún sistema de la vida real.

Es importante acentuar que este proceso no tendría razón de ser si no tuviéramos bases de datos con voluminosos repositorios de información, como las hay en la actualidad, en donde los datos "*en crudo*"; es decir, con ruido, ausentes, intratables y volátiles se tornen difíciles de analizar manualmente en el estado en que se encuentran, siendo muy complicado sacar provecho de dicha información. El valor real de los datos, entonces, reside en la información que podamos extraer de ellos. Para obtener conocimiento útil para una empresa es necesario, antes que nada, hacer un análisis de las necesidades de la organización con lo cual se busca definir un problema en el que se establecen las metas que se desean alcanzar con la minería de datos. Sin un objetivo claro de lo que se busca obtener, el KDD no serviría de mucho.

Lo anterior ha permitido a investigadores de diferentes ámbitos de la ciencia desarrollar diversas técnicas, entre las que destacan: el análisis automático, análisis estadístico de datos, redes neuronales, cómputo evolutivo, visualización de datos, técnicas de representación del conocimiento, razonamiento basado en casos, razonamiento aproximado, computación paralela, sistemas de toma de decisiones, sistemas de recuperación de información y bases de datos, entre muchas otras disciplinas [Agrawal & Shafer 1996]. Como es dicho en [Witten et al. 2011], la Minería de Datos es una ciencia experimental.

Además, se tendrá que definir medidas cuantitativas para los patrones obtenidos, como pueden ser de precisión, utilidad, costo, etc. Y también se deberá establecer alguna medida de interés que considere la validez, utilidad y simplicidad de los patrones obtenidos mediante alguna de las técnicas de Minería de Datos.

Este proceso puede ser muy útil en las compañías para resolver algunos problemas de negocios como: Optimizar procesos de producción a través de la información histórica que se guarde, mantener una ventaja competitiva estudiando las ventajas propias y las de los competidores, aumentar las ganancias por actividad y administrar las relaciones con el cliente, de acuerdo a las necesidades de éste.

### 2.2.1 Proceso de la Extracción del Conocimiento de las Bases de Datos (KDD)

En el mundo de la extracción del conocimiento desde los datos, existen varias metodologías que pueden guiar a las empresas e instituciones, durante este proceso [KDnuggets™]. Una de las comunidades sobre Minería de Datos más grandes del área, nos presenta los resultados de una encuesta hecha en Agosto del 2007, acerca de las metodologías más usadas, así como la frecuencia con la que fueron usadas cada una de ellas:

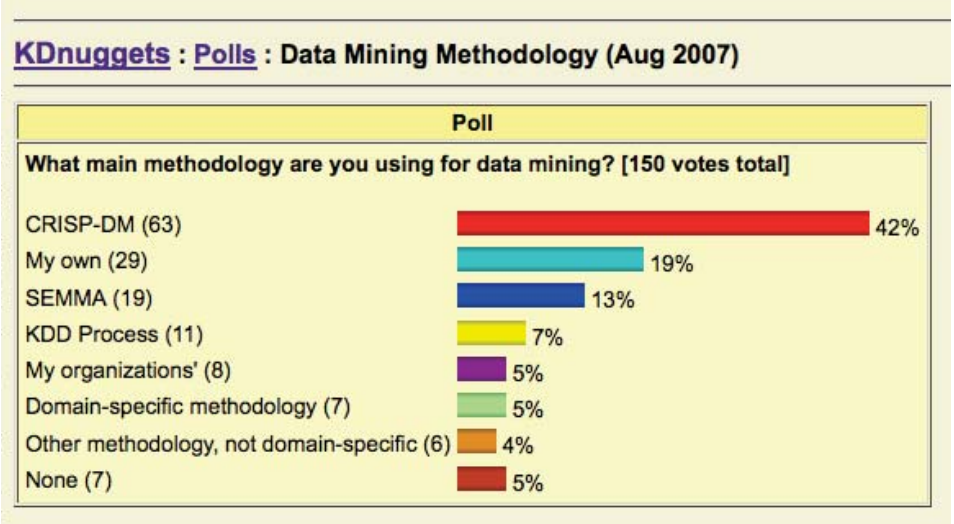



*Tabla 2.1 – Metodologías en Minería de Datos, agosto del 2007. [KDnuggets™ 2007]*

Como se ve en la tabla 2.1, el proceso de KDD ocupa el lugar número 4, como metodología a seguir. En este trabajo, se seguirá esta metodología porque es una de las más ilustrativas y didácticas en el proceso de extracción de conocimiento de las bases de datos.

El KDD como una herramienta de apoyo en la toma de decisiones, ayuda a comprender y modelar de una manera más eficiente y precisa, el contexto de los posibles campos de acción con base en la información con la que se cuenta. Al final se trata de ser capaces de extraer patrones, de describir tendencias y regularidades (o irregularidades), así como de predecir comportamientos.

Es importante entender que al ser una metodología en desarrollo, podemos encontrar información un tanto diferente en relación al proceso del KDD; sin embargo, según [Hernández-Orallo et al. 2007] este proceso se desarrolla a partir de un sistema de información, donde el primer paso a realizar es la 'Preparación de los Datos', permitiendo la selección, limpieza, transformación y exploración de los mismos; después, se aplican las herramientas matemáticas y computacionales para la obtención de modelos o patrones adecuados al conocimiento que se desea extraer, a esto se le llama 'Minería de Datos'; y por último, una vez que se han obtenido los patrones y/o reglas, se evalúan e interpretan a través, entre otras cosas, de la visualización de éstos para obtener el nuevo conocimiento. Dentro de este último paso, también está la labor de consolidar el conocimiento y hacerlo disponible para su uso. También existe un paso *0*, que es un paso inicial. En este paso se obtiene un entendimiento de lo que debe ser hecho en relación a las decisiones que se tomarán durante el proceso de KDD y a las metas finales a alcanzar. Conforme el proceso se desarrolle, puede haber la necesidad de regresar a este paso en varias ocasiones.

De la misma manera, [Hernández-Orallo et al. 2007] organiza el proceso del KDD en torno a 5 fases, ilustradas en la figura 2.4.

#### 2.2.1.1 Recopilación e Integración

Es la primera fase del proceso del KDD, y es en donde se determinan las fuentes de información que pueden ser útiles, ya que puede ser que los datos indispensables se encuentren distribuidos en diferentes sistemas de información y, de esta forma, es necesario recolectarlos. "En general, el problema de reunir un conjunto de datos que posibilite la extracción de conocimiento requiere decidir, entre otros aspectos, qué tipo de información se necesita, de qué fuentes, internas y externas se van a obtener los datos, cómo se van a organizar, bajo qué formato, cómo se mantendrán con el tiempo y, finalmente, de qué forma se va a poder extraer dicho conocimiento total o parcial, en detalle o agregado, ya sea usando distintas *"vistas minables"* a las cuales se le aplicarán herramientas concretas de Minería de Datos." [Hernández-Orallo et al. 2007]

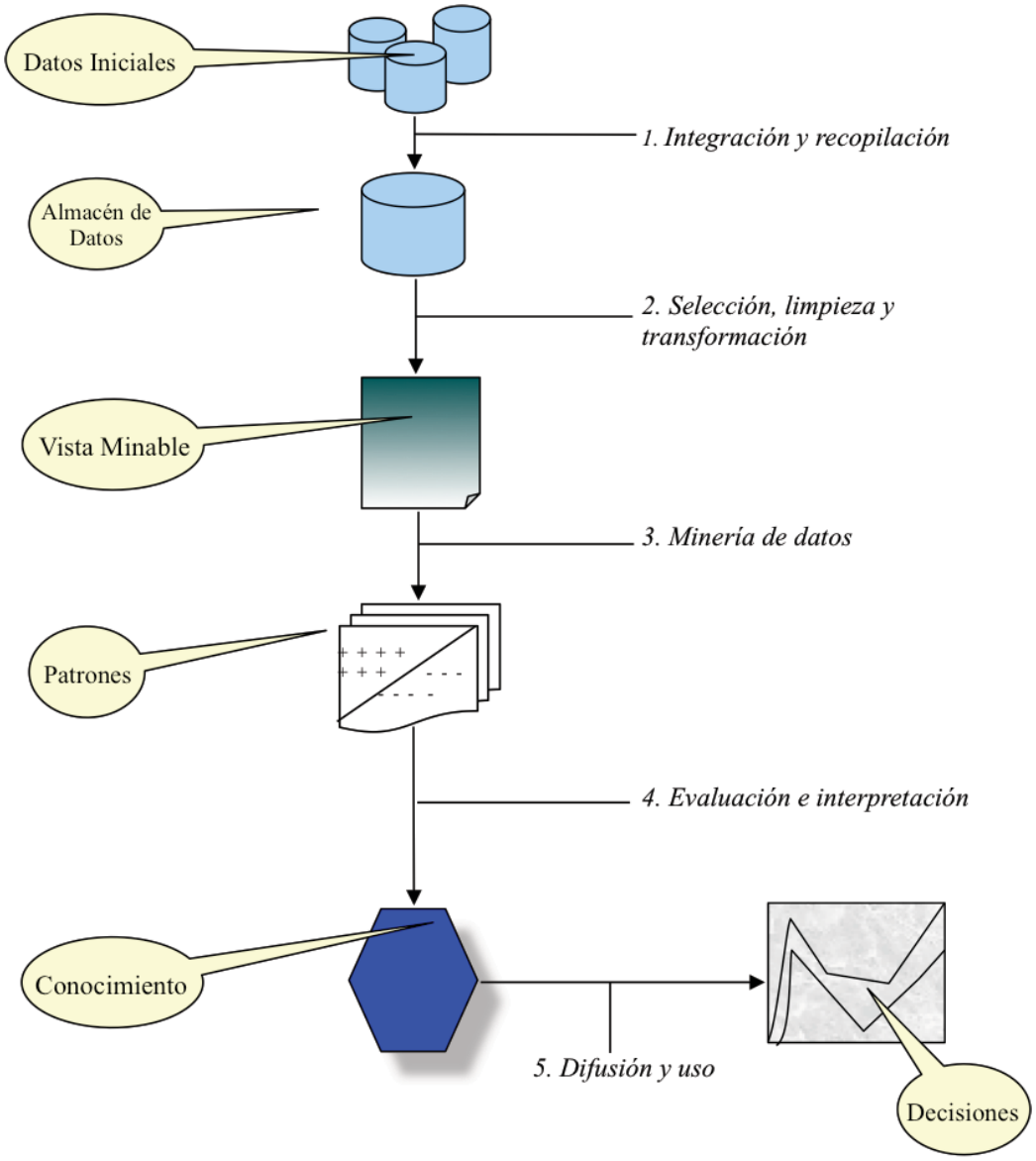
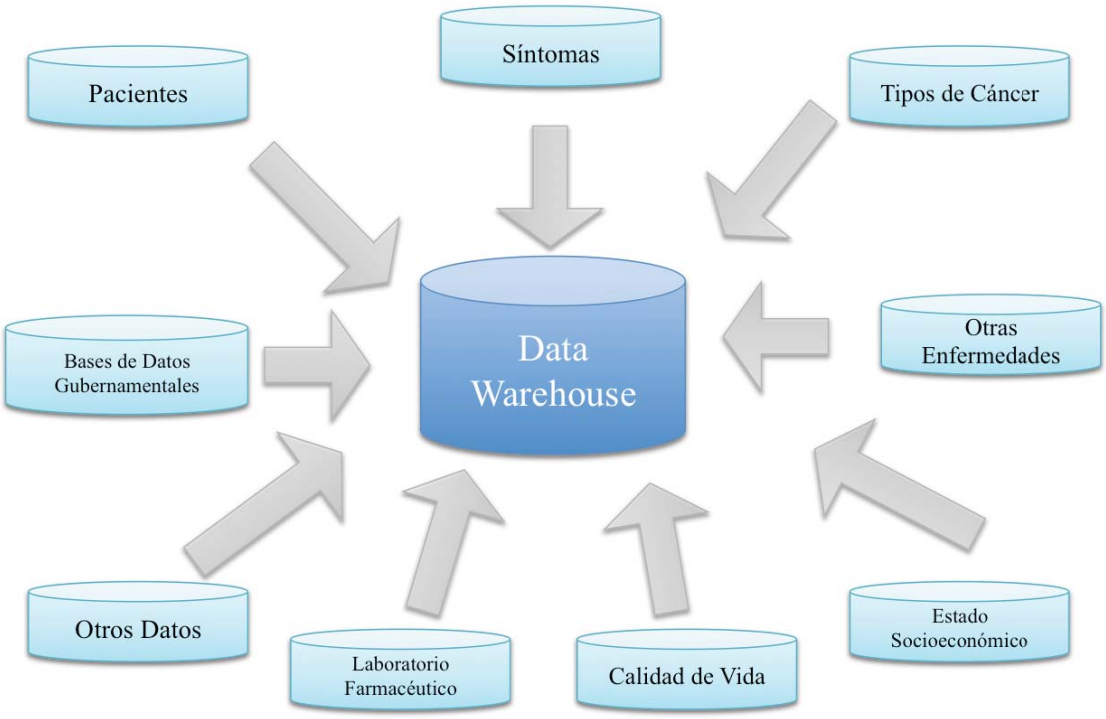



*Figura 2.4 Fases del Proceso de KDD. [Hernández-Orallo et al. 2007]*

La diversidad en el tipo de datos y en la forma en que estos se encuentran en las diferentes bases de datos, representa un reto, ya que cada fuente de datos usa diferentes formatos de registro, diferentes grados de agregación de los datos, diferentes claves primarias, diferentes tipos de error, etc., lo primero, por tanto, es integrar todos estos datos. La idea de la integración de múltiples bases de datos ha dado lugar a la tecnología de *Almacenes de Datos 'DW' (Data Warehousing)*. Este término hace referencia a la tendencia actual en las

empresas e instituciones de coleccionar datos de las bases de datos transaccionales para hacerlos accesibles para el análisis y la toma de decisiones. Un almacén de datos es un repositorio de información coleccionada desde varias fuentes, almacenada bajo un esquema unificado que normalmente reside en un único emplazamiento. Existen varias formas de mezclar las distintas bases de datos para crear el repositorio, pero para no limitar las ventajas para acceder a bases de datos heterogéneas, generalmente los almacenes de datos se construyen vía un proceso de integración y almacenamiento en un nuevo esquema integrado. Esencialmente, los almacenes de datos se utilizan para poder agregar y cruzar eficientemente la información de manera sofisticada. En la figura 2.5 se ilustra este concepto tomando como ejemplo una de las BD del sector salud.




*2.5 Data Warehouse (ejemplo).*

La importancia de un almacén de datos radica en la unificación de la información dentro de un diseño específico que servirá, de forma iterativa, como materia prima a las técnicas de Minería de Datos. Un almacén de datos es muy aconsejable para la Minería de Datos,

aunque no imprescindible, sus ventajas se vislumbran a mediano y largo plazo, lo cual se hace más evidente cuando nos enfrentamos a grandes volúmenes de datos. En algunos casos, en especial cuando el volumen no es muy grande, se puede trabajar con los datos originales o en formatos heterogéneos, como son los archivos de texto, hojas de cálculo, etc. Un almacén de datos es un conjunto de datos históricos, internos o externos, y descriptivos de un contexto o área de estudio, que están integrados y organizados de tal forma que permiten aplicar eficientemente herramientas para resumir, describir y analizar los datos con el fin de ayudar en la toma de decisiones estratégicas. [Hernández-Orallo et al. 2007] Una definición más antigua es la de [Inmon 1992], la cual indica que "un almacén de datos es una colección de datos, orientada a un dominio, integrada, no volátil y variante en el tiempo para ayudar en las decisiones de dirección".

A consecuencia de que gran parte de la información dentro de los almacenes de datos proviene de la base de datos transaccional, la BD sobre la que se pueden realizar todo tipo de operaciones y consultas en tiempo real, se ha desarrollado una tecnología de vaciado y mantenimiento de datos desde la base de datos transaccional al almacén de datos. La arquitectura de los almacenes de datos puede ser de dos formas: con base en un **Modelo Multidimensional**, que permite obtener información desde diferentes niveles de agregación en torno a los hechos mismos; o sobre **Datamarts *(Mercado de Datos)***, donde la información se encuentra organizada en dimensiones bajo una estructura de estrella.

Con el fin de que el análisis sobre los datos se realice de forma eficiente, los sistemas de almacenes de datos pueden implementarse bajo la tecnología de cubos de información, **OLAP (Online Analytical Processing)**, utilizando dos tipos de esquemas físicos: **ROLAP (OLAP Relacional)** y **MOLAP (OLAP Multidimensional)** [Hernández-Orallo et al. 2007]. El primero, se construye sobre una base de datos relacional, que representa los resultados de los informes y consultas como tablas, facilitando su utilización en sistemas de gestión de bases de datos genéricos y, por tanto, el costo de su implementación es menor. Por su parte, los sistemas MOLAP, como también se le conocen, se construyen sobre estructuras basadas en matrices multidimensionales, permitiendo la especialización de

entidades, y una mejor correspondencia entre el nivel lógico y el físico, siendo sistemas en general, mucho más eficientes.

Una vez que se ha diseñado un almacén de datos y ya está implementado, el siguiente paso es cargar los datos. El proceso más tradicional es el de *Migración*, aunque exista un mantenimiento posterior. En realidad, la carga y mantenimiento de un almacén de datos es uno de los aspectos más delicados y que más esfuerzo requieren. El sistema *ETL,* de Extracción, Transformación y Carga*, (*en inglés, *Extraction, Transformation, Load),* es un sistema especializado para esta tarea. La construcción del ETL es responsabilidad del equipo de desarrollo del almacén de datos y se realiza específicamente para cada almacén de datos. El sistema ETL se encarga de hacer algunas tareas como: lectura de datos transaccionales, incorporación de datos externos, creación de claves primarias, integración de datos, obtención de agregaciones, limpieza y transformación de datos, creación y mantenimiento de metadatos, identificación de cambios, planificación de la carga y mantenimiento, indización y pruebas de calidad.

El almacén de datos facilita enormemente la navegación y visualización previa de los datos, para discernir qué aspectos puede interesar que sean estudiados. Este paso es muy importante, porque dependiendo de qué datos se escogen y cómo se escogen, será la calidad del conocimiento descubierto por las técnicas y los modelos construidos durante la fase de Minería de Datos. Si algunos atributos faltan, el estudio completo puede fallar recurriendo a la necesidad de replantear los pasos anteriores.

#### 2.2.1.2 Limpieza, Transformación, Exploración y Selección

Los datos provenientes de diferentes fuentes, pueden ser de baja calidad; es decir, contener valores erróneos, redundantes o faltantes, lo cuál da cabida a la siguiente fase. La segunda fase es la de *"Limpieza, Transformación, Exploración y Selección"*, en la que se eliminan o corrigen los datos incorrectos y se decide la estrategia a seguir con los datos incompletos. Además, se proyectan los datos para considerar únicamente aquellas variables o atributos que van a ser relevantes, con el objetivo de hacer más fácil la tarea propia de minería y para

que los resultados de la misma sean más útiles. El objetivo de esta fase es el de mejorar la calidad de los datos a ser minados, para facilitar la aplicación de los algoritmos de Minería de Datos.

En la práctica, esta fase y la anterior suelen consumir alrededor del 60% al 70% del total del tiempo empleado en el proceso completo del KDD. Las dos primeras fases se suelen englobar bajo el nombre de *"Preparación de Datos"*. Como se explicará en el capítulo III (página 54), este trabajo gira alrededor de esta fase en relación a la *Selección de Características o Atributos*.

La calidad del conocimiento descubierto no sólo depende del algoritmo de minería utilizado, sino también de la calidad de los datos minados, es decir, que éstos estén íntegros, completos y sean consistentes. Por ello, después de la recopilación, el siguiente paso en el proceso KDD es la limpieza y transformación de los datos, que se concentrará en obtener un subconjunto de datos que es el que se va a minar, al cual se le conoce como V*ista Minable*.

Este paso es necesario ya que algunos datos coleccionados en la etapa anterior son irrelevantes e innecesarios para la tarea de minería que se desea realizar. Además, existen otros problemas que afectan a la calidad de los datos. Uno de estos problemas es la presencia de valores que no se ajustan al comportamiento general de los datos (*outliers*). Estos datos anormales pueden representar errores en los datos o pueden ser valores correctos que son simplemente diferentes a los demás. Algunos algoritmos de minería de datos, como los árboles de decisión, ignoran completamente estos datos; otros los descartan considerándolos ruido o excepciones; pero otros, como la regresión y las redes neuronales, son muy sensibles y no los toleran, además de que el resultado se ve claramente perjudicado por ellos. Sin embargo, no siempre es conveniente eliminarlos, ya que, en algunas aplicaciones los eventos raros pueden ser más interesantes que los regulares, (aplicaciones de tarjeta de crédito o predicción de inundaciones, por ejemplo). La integración también produce una disparidad de formatos, nombres, rangos, etc. que podría no existir, o en menor medida, en las fuentes originales. La identificación de datos faltantes

o perdidos puede ser también un problema pernicioso que puede conducir a resultados poco precisos. No obstante, es necesario reflexionar primero sobre el significado de los valores faltantes antes de tomar alguna decisión sobre cómo tratarlos ya que éstos pueden deberse a causas muy diversas, como a un mal funcionamiento de un dispositivo que hizo la lectura del valor, a cambios efectuados en los procedimientos usados durante la recolección de los datos o al hecho de que los datos se recopilen desde fuentes diversas. Por ellos, existen muchas aproximaciones para manejar los datos faltantes.

Estos problemas son sólo algunos ejemplos que muestran la necesidad de limpiar, transformar o seleccionar los datos; es decir, de mejorar su calidad, proporcionando a los métodos de minería de datos, el subconjunto de datos más adecuado para resolver un problema dado. La selección de atributos relevantes, la cual es el tema central de esta tesis, es uno de los métodos de pre-procesamiento más importantes, ya que es crucial que los atributos utilizados sean los necesarios para la tarea de minería de datos, que va a depender del conocimiento que se desea adquirir. Tal conocimiento sobre el dominio del problema puede permitirnos hacer correctamente muchas de esas selecciones. Aunque al principio algunos algoritmos de minería de datos ignoran automáticamente las variables irrelevantes, en la práctica, incluir variables irrelevantes o redundantes complica el modelo de minería de datos obtenido, siendo complejo, confuso y difícil de poner en práctica. A continuación, se hablará de cada una de estas subtareas.

#### 2.2.1.2.1 Limpieza de los datos.

La limpieza de los datos, que se lleva a cabo inmediatamente después de la integración de los datos, puede en muchos casos, detectar y solucionar problemas no resueltos durante la etapa de integración.

Cuando tenemos integrados todos los datos, lo primero que podemos realizar es una exploración de los mismos. Como lo hemos dicho anteriormente, sin conocimiento del dominio y sin observar el dato a detalle, no podemos determinar si dicho dato es simplemente un dato anómalo pero correcto o erróneo.

A la hora de hablar de valores faltantes, perdidos o ausentes *(missing values),* debemos también tratar acerca de su detección y tratamiento. Esta tarea parece sencilla, si los datos proceden de una misma base de datos, basta mirar en la tabla de resumen de atributos/características y ver la cantidad de nulos que tiene cada atributo. El problema es que a veces los campos faltantes no están representados como nulos. Aunque existen campos en los que las restricciones de integridad del sistema evitan introducir códigos fuera del formato para representar valores faltantes, esto al final ocurre en muchos otros, especialmente en campos sin formato. A veces son las propias restricciones de integridad las que causan el problema. Este tipo de problemas complican de sobremanera la detección de valores faltantes, introduciendo sesgo en el conocimiento extraído. Las razones más comunes para los valores faltantes suelen ser las siguientes:

- Algunos valores faltantes expresan características relevantes, es decir, el que un valor no se encuentre en un campo puede ser debido a una razón intencional.
- Muchos valores faltantes existen en la realidad pero otros no, por lo que se convierten en valores no existentes.
- Si los datos vienen de fuentes diferentes, al combinarlos se suele hacer la unión y no la intersección de campos, con lo que muchos datos faltantes representan que dichas duplas provengan de una o más fuentes diferentes al resto.
- En la mayoría de las ocasiones, los datos faltantes se deben a un mal diseño y/o implementación del o los almacenes de datos.

Finalmente, si se ha conseguido detectar los datos faltantes e, idealmente, sus causas, procederemos a su tratamiento. Las posibles acciones sobre datos faltantes son:

- Ignorarlos: Algunos algoritmos son robustos a datos faltantes.
- Eliminar, filtrar o reemplazar toda la columna: Es una solución extrema, pero en ocasiones la proporción de nulos es tan alta que la columna no tiene arreglo.
- Filtrar la fila: Esto claramente sesga los datos, porque muchas veces las causas de un dato faltante están relacionadas con casos o tipos especiales.

- Reemplazar el valor: Se puede intentar reemplazar el valor manualmente, en caso de que no haya muchos, o automáticamente por un valor que preserve la media o la varianza, en el caso de valores numéricos o, por el valor de la moda, en el caso de valores nominales. Una manera más sofisticada de estimar un valor es *predecirlo* a partir de otros ejemplos, utilizando cualquier técnica predictiva de aprendizaje automático (clasificación o predicción) o técnicas más específicas (como el algoritmo EM, '*Maximización de la Esperanza'*. Si sustituimos un dato faltante por uno estimado, lo que realmente pasa es que perdemos información y, al mismo tiempo, inventamos información con los riesgos que pueda tener de que sea errónea. Una solución para este problema puede ser crear un nuevo atributo lógico indicando si el atributo original era nulo o no, lo cuál permite saber que el dato era faltante y, por tanto, que el valor hay que tomarlo con cautela.
- Segmentar: Se segmentan las tuplas por los valores que tienen disponibles. Se obtienen modelos diferentes para cada segmento y luego se combinan.
- Modificar las políticas de calidad de datos y esperar hasta que los datos faltantes estén disponibles.

En general, si la muestra de datos con la que se cuenta es muy grande, se puede prescindir del registro completo donde se encuentre el dato faltante (si éste esta en un campo clave) o, de otro modo, tratar de predecirlo por diversos métodos estadísticos. Del mismo modo que para los campos faltantes, para los campos erróneos o inválidos, se ha de distinguir entre la detección y su tratamiento. La detección de campos erróneos se puede realizar de maneras muy diversas, dependiendo del formato y origen del campo. En el caso de datos nominales, la detección dependerá fundamentalmente de conocer el formato o los posibles valores del campo. Parece evidente que aquellos datos erróneos que sí se ajusten al formato serán mucho más difíciles, sino imposibles de detectar. Es importante destacar que no detectar un valor anómalo puede ser un problema importante si el atributo se normaliza posteriormente, ya que la mayoría de los atributos estarán en un rango muy pequeño y puede haber poca precisión o sensibilidad para algunos métodos de minería de datos. Si un sistema de información original fuese diseñado con buenas restricciones de integridad, no podría introducirse campos erróneos. En este caso, la detección de campos erróneos debido al

formato no sería necesario ya que todos los datos se ajustan al formato correcto y sería más fácil concentrarse en otro tipo de errores.

En algunos casos, sin embargo, es posible detectar campos erróneos por su contenido, buscando que el dato registrado sea coherente con las características del atributo. También mediante la tabla de resumen de atributos o histogramas, podemos detectar valores erróneos en relación al tipo de dato (nominal o numérico). En el caso de detectar valores erróneos en atributos numéricos, se empieza por buscar valores anómalos, atípicos o extremos, llamados *datos aislados, exteriores o periféricos*. Es importante destacar que un valor erróneo y un valor anómalo no son lo mismo; existen casos en los que los valores extremos se categorizan como anómalos estadísticamente pero son correctos, es decir, representan un dato fidedigno de la realidad; no obstante, pueden ser un inconveniente para algunos métodos que se basan en el ajuste de pesos, como las redes neuronales. Por el contrario, también puede haber datos erróneos que caen en la normalidad y, por tanto, es mucho más difícil detectarlos.

Otras técnicas para la detección de valores anómalos consisten en definir una distancia (incluyendo valores nominales y numéricos) y ver los individuos con mayor distancia media al resto de los individuos (individuo cuyo vecino más próximo o cuyos k-vecinos más próximos está más lejos) o bien, donde una fracción de los otros individuos está lejos. Ésta es una técnica más general que la de los Histogramas y se ha desarrollado en gran medida últimamente [Knorr & Ng 1998].

También, podemos utilizar técnicas de *Clustering* parciales (no todos los elementos caen dentro de un grupo). Aquellos elementos aislados que no entren en un grupo o aquellos grupos minoritarios pueden ser considerados valores extremos. También se puede realizar una detección por predicción; es decir, se realiza un modelo de clasificación o regresión y se predice el valor del atributo para cada ejemplo. Aquellos ejemplos cuya predicción está más lejos del valor que existe en la instancia puede denotar un error a ser inspeccionado. Para profundizar en la gran cantidad de métodos para detectar valores anómalos se puede consultar [Peña & Prieto; Peña 2002; Barnett & Lewis 1994; Rousseeuw & Leroy 2003].

En lo concerniente al tratamiento de los valores anómalos o erróneos, los más usados son los siguientes:

- Ignorarlos.
- Filtrar la columna: Es una solución extrema, pero a veces existe otra columna dependiente con datos de mayor calidad. Una columna con valores erróneos es muy poco confiable para alimentar un modelo, por tanto, es mejor hacerla a un lado.
- Filtrar la fila: Esto puede sesgar los datos, porque muchas veces las causas de un dato erróneo están relacionadas con casos o tipos especiales. Sin embargo, al mismo tiempo, muchos algoritmos de minería de datos podrían arrojar resultados incoherentes gracias a este tipo de datos.
- Reemplazar el valor: Por el valor "nulo" si el algoritmo lo trata bien, o por máximos, mínimos o medias, dependiendo del contexto de los datos.
- Discretizar: Transformar un valor continuo en uno discreto.

Los atributos erróneos son mucho más graves cuando afectan a un atributo que va a ser utilizado con clase o como valor de salida de una tarea predictiva de minería de datos, por ejemplo si un grupo de pacientes tiene cáncer o no, ya que pueden sesgar el resultado. Dentro de estos casos, existe un tipo peculiar todavía más grave: todos los atributos de dos o más ejemplos son idénticos, excepto la clase, usualmente en atributos numéricos. Este tipo de errores se consideran como *inconsistencias* e incluso algunos métodos de minería de datos no pueden "digerirlos", por lo que es importante, en estos casos, eliminarlos.

#### 2.2.1.2.2 Transformación de los datos.

La transformación de datos engloba cualquier proceso que modifique la forma de los datos. Prácticamente todos los procesos de preparación de datos entrañan algún tipo de transformación. Específicamente en relación a la transformación de atributos, tenemos las que incrementan y decrementan la dimensionalidad, cambian el tipo de dato o el rango, y algunas de otro tipo para el tratamiento de variables no lineales y/o espaciales.

La *reducción de la dimensionalidad*, es una de las tareas principales en la transformación de datos. Una dimensionalidad alta representa un gran problema a la hora de aplicar alguna de las técnicas de Minería de Datos. Si tenemos muchas dimensiones (atributos) respecto a la cantidad de instancias o ejemplos, al tener un gran número de variables (grados de libertad), los patrones extraídos pueden ser caprichosos y poco robustos. Afectando la eficiencia y dificultando la representación de los datos. Una forma de reducción es la Selección de Atributos, tarea que se analizará a fondo en el capítulo III.

La Reducción de la Dimensionalidad a través de la transformación de los datos, sustituye el conjunto de atributos iniciales por otros diferentes, a lo cual, geométricamente, se denomina *Proyección*. Existen muchas técnicas para realizar este tipo de proyecciones, entre los que se encuentran: el análisis de componentes principales, la cuál es la técnica más tradicional, conocida y eficiente para reducir la dimensionalidad por transformación y sólo es aplicable a atributos numéricos; y algunas técnicas del análisis factorial, como son: basadas en mínimos cuadrados, máxima verosimilitud, factorización de ejes principales, factorización alfa y factorización de imagen, etc. [Peña 2002; Renche 2002].

En muchos casos parece razonable reducir la dimensionalidad; sin embargo, en otros lo que interesa es aumentarla, generando nuevos atributos que, incluso, son combinación de otros. Un buen ejemplo es la construcción de llaves o ID, a partir de dos o más atributos. Si aumentamos la dimensionalidad conseguimos que los datos se separen en el espacio, facilitando la creación de fronteras lineales donde antes no las había. Esto quiere decir que, si realizamos un aumento de dimensionalidad adecuado, podemos convertir algunos problemas no lineales o incluso irresolubles, en problemas lineales, al "aclararse el espacio".

Uno de los aspectos más importantes en la transformación de un atributo, es el tipo. El hecho de que un atributo sea nominal o numérico determina, en gran medida, como va a ser tratado por las herramientas de minería de datos. En algunas ocasiones, puede ser conveniente convertir un atributo numérico a nominal, llamado *Discretización*; o viceversa, N*umeración*.

Existen muchas otras transformaciones de los datos que pueden permitir, en muchos casos, extraer patrones más fáciles e incluso más comprensibles. Entre ellas se encuentra el *Escalado Multidimensional*, una técnica del análisis multivariado [Krzanowski 1988; Peña 2002; Johnson 1999]; ó las transformaciones especiales para variables temporales o en forma de series; la transformada de Fourier, las transformaciones de *wavelets* discretas u otras transformaciones que cambian los datos del dominio de la amplitud, una serie, al dominio de las frecuencias, un espectro.

#### 2.2.1.2.3 Exploración de datos.

Además de recopilar, integrar y limpiar los datos, es conveniente, para realizar una tarea de minería de datos, hacer una exploración y selección de los mismos, con el objetivo de conocerlos mejor de cara a la tarea de minería de datos y obtener una *"vista minable con tarea asignada"*, es decir, una vista minable con entradas y salidas e instrucciones sobre qué datos trabajar, qué tarea realizar y de qué manera obtener el conocimiento. Aquí, es donde se debe tener muy en claro que objetivo se persigue al minar los datos y que conocimiento se está buscando. Para poder definir nuestra meta, [Hernández-Orallo et al. 2007] sugiere dar respuesta a las siguientes interrogantes:

- *¿Qué parte de los datos es pertinente analizar?* Por ejemplo, un producto en específico de una empresa. La mayoría de los métodos de minería de datos sólo son capaces de tratar con una tabla relativamente pequeña en cada tarea, recogiendo toda y únicamente la información necesaria para desarrollar el modelo.
- *¿Qué tipo de conocimiento se desea extraer y cómo se debe presentar?* Por ejemplo, si se desea analizar una muestra de datos de los últimos 6 meses, presentada quincenalmente. Se trata de decidir qué **tarea** (clasificación, regresión, agrupamiento, reglas de asociación, etc.), cuáles son las entradas y salidas en las tareas predictivas, qué **método** de entre los existentes para cada tarea se usará, (árboles de decisión, redes neuronales, regresión logística, etc.), y de qué manera se van a **presentar** los resultados.

- *¿Qué conocimiento puede ser válido, novedoso e interesante?* Por ejemplo, una nueva estrategia para un segmento nuevo de la población. Esto se refiere a los **criterios de calidad** de los métodos aplicados. Como los criterios de comprensibilidad de los modelos (número de reglas máximo), criterios de fiabilidad (la confianza para las reglas de asociación, la precisión para la clasificación, el error cuadrático medio para la regresión, etc.), criterios de utilidad (basados en medidas de cuándo son aplicables, como el soporte, qué beneficios se obtiene, matrices de costos, etc.), y criterios de novedad o interés (medidas subjetivas).
- *¿Qué conocimiento previo me hace falta para realizar esta tarea?* Por ejemplo, la consulta hacia bases de datos externas que pueden enriquecer el estudio. Conforme avanza el proceso, puede surgir la necesidad de recurrir a cierto conocimiento previo, como añadir otras tablas u otros modelos anteriores como apoyo para construir uno nuevo, reconociendo el conocimiento que podría ser útil. Lo importante en este punto es entender que se está dentro de un proceso iterativo, que irá siendo más sencillo a medida en que se conocen los datos, el contexto, los usuarios, las técnicas de exploración y de minería de datos.

En el caso específico de la exploración de los datos, se debe de poner cierto énfasis en el reconocimiento del dominio (entorno y posibles escenarios), de los usuarios (quienes usarán el modelo) y de los datos, a través del análisis de los mismos, tarea también llamada en inglés, *data survey, exploratory data analysis, data fishing, prospection.* Su objetivo es reconocer los datos, guiado por el interés y las necesidades establecidas en el reconocimiento del dominio, para saber qué datos son relevantes y qué tareas pueden ser útiles. Entre las técnicas usadas para la exploración de los datos se encuentran: visualización (por ejemplo, histogramas, diagramas de caja o gráficas de dispersión), visualización multidimensional, descripción, generalización, agregación, pivotamiento y selección, en esta última nos enfocaremos en el siguiente subtema.

#### 2.2.1.2.4 Selección de datos.

El proceso de Selección de datos muchas veces se engloba dentro de un concepto más amplio, denominado *Reducción de los Datos*. Sin embargo, la selección de datos no tiene únicamente como objetivo la reducción del tamaño para hacer más rápido el algoritmo de minería de datos sino que, en muchos casos, puede permitir mejorar el resultado, tanto en precisión y costo, como en comprensibilidad.

La selección de datos no sólo abarca el decidir qué tablas o archivos se necesitarán, sino que también es muy importante decidir que atributos/variables e instancias van a ser necesarias. El problema existente es precisamente que si se selecciona como "vista minable" todo aquello que pueda ser relevante, al final se estaría contando con cientos de atributos y millones de registros.

En general, cuando tratamos de datos sobre una tabla (atributo-valor), hay dos tipos de selección aplicables: la selección vertical, donde se aplica técnicas de reducción de dimensionalidad en la eliminación de atributos. En el subcapítulo 2.3, haremos énfasis en este tipo de selección, la cuál es el tema central del presente trabajo. Y la selección horizontal, donde se puede aplicar técnicas de muestreo para la eliminación de algunos individuos (filas). Consecuentemente, una buena idea es usar una muestra a partir de algunos datos. La selección de la muestra debe ser hecha aleatoriamente para no sesgar dicha muestra.

### 2.2.1.3 Minería de Datos

La tercera fase se llama *"Minería de Datos"* y es la más conocida. Este concepto se suele confundir en gran medida con el proceso completo de la extracción de conocimiento de las bases de datos, y dado que es más conciso, rentable y comercial se le ha decidido nombrar como Minería de Datos.

No obstante, la Minería de Datos es una de las fases más importantes del proceso de KDD, ya que se encarga de aplicar técnicas matemáticas y computacionales para obtener modelos, reglas o patrones a partir de los datos recopilados en los procedimientos anteriores [Hernández-Orallo et al. 2007], y se lleva a cabo, una vez que se han preparado los datos y se ha obtenido una *vista minable*. El nombre "Minería de Datos" − *Data Mining*, en el inglés original− viene de las similitudes entre la búsqueda de información de gran valor para el negocio en grandes bases de datos y la búsqueda de oro y metales preciosos en las minas de las montañas.

Una buena definición sobre la Minería de Datos es la presentada en [Fayyad et al. 1996a]: 'Es el empleo de algoritmos y procedimientos para sacar a la luz asociaciones, correlaciones, reglas, patrones e incluso excepciones interesantes o potencialmente útiles, desconocidas y escondidas en las grandes bases de datos'.

Es importante comprender que la Minería de Datos es una herramienta o un asistente para el análisis de datos; no elimina la necesidad de conocer el problema y los datos, así como el objetivo que dirige dicha búsqueda. En la figura 2.6, se ilustra el proceso de Minería de Datos:

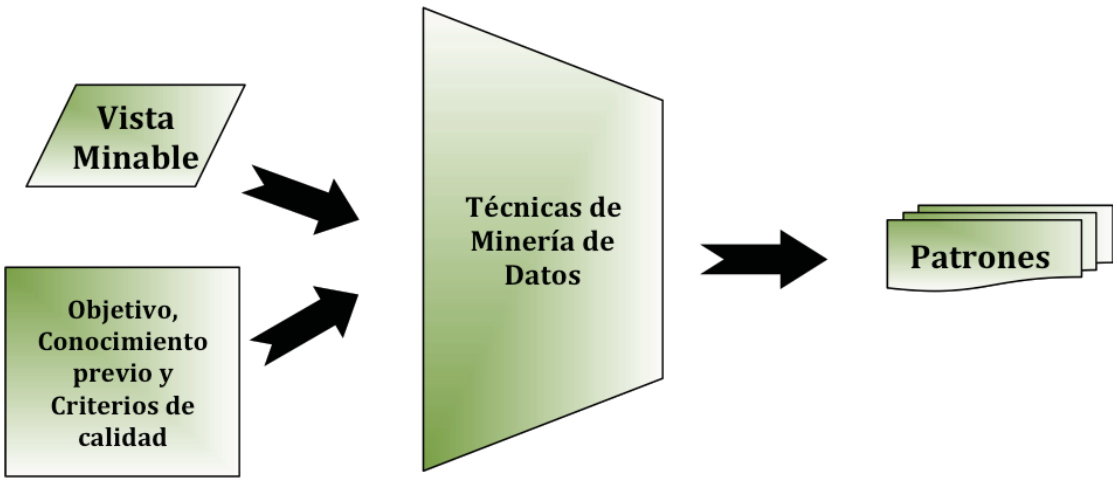



*Figura 2.6 Proceso de Minería de Datos. [Hernández-Orallo et al. 2007]*

Aunque en la imagen 2.6 el proceso parece simple, en la realidad, las técnicas que extraen patrones a partir de los datos son computacionalmente costosas, al ser un proceso iterativo y, cuanto más expresivos, novedosos, comprensibles e interesantes son los patrones tanto más se incrementa dicho costo.

Las raíces de la Minería de Datos son el Aprendizaje de Máquina y la Estadística. Recordemos que el Aprendizaje de Máquina es una rama de la Inteligencia Artificial *(IA)* que gira en torno al diseño y aplicación de algoritmos de aprendizaje que tratan de interpretar grandes cantidades de datos. Por su parte, la Estadística puede ser definida como la metodología de la extracción de información de los datos, su correlación y la expresión de la incertidumbre en las decisiones que tomamos. [Johnson 1990] La Minería de Datos contiene una amplia gama de técnicas, y los aspectos que nos ayudan para comparar éstas son la *expresividad, comprensibilidad, robustez, facilidad de uso, precisión, eficiencia,* entre otras.

Algunos de los modelos de Minería de Datos son: Árboles de Decisión, Inducción de Reglas, Redes Neuronales Artificiales, Aprendizaje basado en Instancias, Aprendizaje Bayesiano, Programación Lógica Inductiva, Computación Evolutiva y algunos Algoritmos Estadísticos. Cada modelo posee diferentes características y cada característica puede ser adecuada para algunos conjuntos de datos e inadecuada para otros. Por lo tanto, ningún algoritmo de Minería de Datos es universal y el mejor para todos los conjuntos de datos.

El primer paso en el desarrollo de un algoritmo de minería de datos es definir cuál es la tarea en cuestión. En el proceso completo de búsqueda de información, podemos encontrar tareas y métodos. Una *Tarea*, es un tipo de problema, con sus propios requisitos y características, para los cuáles existen diferentes *Métodos* o algoritmos de minería de datos, que pueden resolverlo. Los métodos pueden ser en general predictivos o descriptivos. El carácter predictivo nos sirve para prever el comportamiento futuro de algún tipo de entidad, obteniendo como valores de salida una clase, categoría, valor numérico o un orden entre ellos, sobre los cuáles se tomarán decisiones a futuro; mientras que el carácter descriptivo nos ayuda a la comprensión de los eventos en el presente mediante la representación de los

datos. Según [Maimon & Rokach 2005a], la *predicción* es referida frecuentemente como minería de datos supervisada, mientras que la Minería de Datos *descriptiva* incluye la parte de la visualización y los métodos no supervisados de la Minería de Datos. Entre las tareas predictivas encontramos la clasificación y la regresión mientras que el agrupamiento (*clustering*), las reglas de asociación, las reglas de asociación secuenciales y las correlaciones son tareas descriptivas. Un modelo de minería de datos, puede componerse de 1 o más métodos predictivos y/o descriptivos.

Los algoritmos de minería de datos suelen tener tres componentes:

- El modelo, que contiene parámetros que han de fijarse a partir de los datos de entrada.
- El criterio de preferencia, que sirve para evaluar el modelo, o compararlo con otros modelos alternativos.
- El algoritmo de búsqueda.

Las principales diferencias entre los algoritmos de minería de datos se hallan en el modelo de representación escogido y la función objetivo del mismo. Es por eso que es tan importante la determinación adecuada de qué técnica o procedimiento se va a usar en el tratamiento de los datos, lo cual depende de cuáles son las metas del proceso de KDD, del conocimiento del problema y de la experiencia obtenida al aplicar los pasos anteriores.

#### 2.2.1.4 Evaluación e Interpretación

La cuarta fase es la de *"Evaluación e Interpretación"*. A través de esta fase, uno puede saber si un determinado modelo es lo suficiente válido para nuestros propósitos. Aquí, se evalúan los patrones resultantes de los algoritmos de Minería de Datos y son interpretados por los expertos; si es necesario, se vuelve a las fases anteriores para una nueva iteración donde se busca resolver posibles conflictos con la información que se disponía anteriormente. En esta fase, se mide principalmente la calidad de los patrones descubiertos, lo cuál no es un problema trivial, ya que dentro de los criterios utilizados, algunos de ellos pueden ser muy subjetivos.

#### 2.2.1.4.1 Evaluación del Modelo.

La etapa de evaluación de modelos es crucial para la aplicación real de técnicas de Minería de Datos. Sin embargo, establecer medidas justas y exhaustivas no es tarea sencilla, dado el grado de subjetividad que muchas técnicas involucran y la falta de una clase objetivo a ser medida. Existen muchas sugerencias de cómo evaluar diferentes técnicas de Minería de Datos, a continuación presentamos las que han sido usadas en este trabajo de investigación:

- *Evaluación de Clasificadores*: El objetivo del Aprendizaje Automático Supervisado es el aprendizaje de una función ***f***, considerando un espacio de posibles hipótesis ***H*** , de donde se toma una muestra ***S***, formada por ejemplos de la función objetivo ***f*** de acuerdo con una distribución ***D*** [Mitchell 1997]. Para evaluar la calidad de una *hipótesis*, se pueden considerar los siguientes métodos:
    - *Evaluación basada en la Precisión*: Esta técnica se enfoca en la maximización de la precisión de una hipótesis o, lo que es lo mismo, la minimización del porcentaje de error de la hipótesis con respecto a la función ***f***. Existen dos tipos de error: el *error de la muestra* $\left(error_s(h)\right)$, utilizando los datos que se poseen de ***f*** ó, el *error verdadero* $\left(error_t(h)\right)$, que utiliza la distribución ***D***, y es el que se desearía conocer acerca de una determinada hipótesis. Sin embargo, el error de la muestra es todo lo que se puede conocer acerca de la precisión de una hipótesis, debido a que generalmente se desconoce la función objetivo ***f***.
    - *Validación Cruzada*: Esta técnica permite reducir la dependencia que existe entre el resultado del experimento y la forma en que se ha realizado la partición de los subconjuntos. La validación cruzada consiste en dividir el conjunto de la evidencia en ***k*** subconjuntos disjuntos de tamaño similar. Luego, se realiza la unión de todos los ***k-1*** subconjuntos y se utiliza el conjunto resultante para calcular el error de la muestra parcial. Este procedimiento se repite ***k*** veces, utilizando un conjunto diferente para estimar el error parcial. El error final de la muestra se calcula como la media

aritmética de los ***k*** errores parciales. Usualmente, ***k*** toma el valor de 10. Otra ventaja de esta técnica es que la varianza de los ***k*** errores de muestras parciales, permite estimar la variabilidad del método de aprendizaje con respecto a la evidencia [Hernández-Orallo et al. 2007]. Existe una aproximación de validación cruzada 5x2, donde el número de repeticiones es 5 con ***k=2***. También se pueden establecer intervalos de confianza siguiendo una distribución binomial o normal, o utilizar diferentes modificaciones a este tipo de técnicas.

- *Maximización de la Tasa de Precisión*: La Tasa de Precisión, una medida de la Precisión Predictiva para el enfoque de clasificación, para un atributo que tiene sólo dos clases a ser predichas, una positiva (+) y otra negativa (-), es a través del análisis de VP, VN, FP, FN. Puesto que se intenta clasificar cada instancia con sólo dos clases, una salida primaria o positiva *'1'*, y una secundaria o negativa *'0'*; la función de aptitud para la precisión predictiva puede resumirse en una Matriz de Confusión [Hand 1997; Weiss & Kulikowski 1991; Freitas 2002]. La clase predicha por una instancia dada es *C* (positiva), si la instancia satisface la condición *"A"*; de otra forma es negativa. La matriz de confusión contiene información respecto al desempeño de una posible solución o conjunto de soluciones y podría definirse de la siguiente forma:

| | | *Clase Real* | |
|---|---|---|---|
| | | **C** | **No C** |
| *Clase Predicha* | **C** | VP | FP |
| | **No C** | FN | VN |

*Tabla 2.2 – Medida de Precisión para el Enfoque de Clasificación [Freitas 2002]*

Donde:

**VP =** Verdaderos Positivos: Número de instancias que satisfacen "A" y tienen la clase 'C'.

**FP =** Falsos Positivos: Número de instancias que satisfacen "A" pero no tienen la clase 'C'.

**FN** = Falsos Negativos: Número de instancias que NO satisfacen "A" y tienen la clase 'C'.

**VN** = Verdaderos Negativos: Número de instancias que NO satisfacen "A" y no tienen la clase 'C'.

Dados los valores para VP, FP, FN y VN, se pueden definir las siguientes medidas:

$$Precisión = {VP}/{(VP + FP)}$$

$$Tasa\ de\ Verdaderos\ Positivos = {VP}/{(VP + FN)}$$

$$Tasa\ de\ Verdaderos\ Negativos = {VN}/{(VN + FP)}$$

$$Tasa\ de\ Precisión = {(VP + VN)}/{(VP + FP + FN + VN)}$$

La Precisión también puede ser llamada *consistencia ó confianza*. A la tasa de verdaderos positivos también se le llama *sensibilidad* y a la de verdaderos negativos, *especificidad*. Una función de aptitud puede usar cualquier combinación de las medidas anteriores, sin embargo algunas combinaciones son más sensibles que otras. [Freitas 2002]. Un ejemplo de esta matriz puede observarse en la figura 5.1 del subcapítulo 5.3 sobre el Análisis de los Resultados.

- *Interés*: El interés de un modelo intenta medir la capacidad de ese modelo de suscitar la atención y aprobación del usuario en relación al modelo. Pueden ser de dos tipos: subjetivas y objetivas [Freitas 2002]. Son *subjetivas* cuando se tienen en cuenta los conocimientos y expectativas del usuario acerca del conocimiento que se desea extraer. Las medidas *objetivas* utilizan los datos minados para estimar el grado de interés del usuario hacia el modelo.
- *Novedad*: Es un criterio relacionado con la capacidad de un modelo de sorprender al usuario con respecto al conocimiento previo que tenía sobre determinado problema.
- *Comprensibilidad o Inteligibilidad*: La comprensibilidad de un modelo es la capacidad de comprensión que un usuario obtiene acerca del modelo; por

tanto, es una cuestión mucho más subjetiva, ya que un modelo puede ser poco comprensible para un usuario y muy comprensible para otro.

- *Simplicidad*: Este criterio se basa en establecer el tamaño o complejidad del modelo. Entre más simple, mejor.
- *Aplicabilidad*: La calidad de un modelo se basa en su capacidad de ser utilizado con éxito en el contexto real donde va a ser aplicado, al ver como influyen los resultados del modelo dentro del proceso de toma de decisiones.

#### 2.2.1.4.2 Interpretación de la información.

Una vez que la etapa de la Evaluación ha terminado, la información que es aportada por los modelos aprendidos no se puede utilizar directamente, sino que se necesita de una fase de refinamiento que permita concretar cuál es el conocimiento que aportan. A esta etapa se le denomina "Interpretación" y es muy útil para sustentar la toma de decisiones final.

Los modelos que se producen a partir de la fase de Minería de Datos, pueden ser tan complejos o su representación es tan abstracta, que prácticamente no se conoce su comportamiento interno. Es por ello que, en algunas ocasiones, se intenta explorar otras técnicas de minería de datos para facilitar la interpretación de las reglas y patrones obtenidos.

Los sistemas de reglas son considerados una de las representaciones que permiten comprender más fácilmente el comportamiento de un modelo, mientras el número de reglas no sea demasiado grande, teniendo en cuenta que dichas reglas son expresadas a través de los atributos del propio problema. La desventaja de este tipo de métodos, es que algunas veces su medida de precisión no es la deseable ó pueden ser modelos de lento aprendizaje; por tanto, lo que se aconseja en estos casos, es usar una estrategia llamada "Extracción de Reglas Comprensibles", que consiste en tratar de aprender el modelo con diferentes métodos, y escoger el que obtenga los mejores resultados; después, intentar ajustar el comportamiento del modelo obtenido con un conjunto de reglas.

Por otra parte, los diferentes métodos y técnicas de visualización, ayudan en gran medida en la fácil interpretación de los resultados, ya que permiten que los usuarios puedan identificar fácil y directamente, los patrones más significativos. Hoy en día, se han definido representaciones para Árboles de Decisión, Reglas de Asociación, Redes Bayesianas, Modelos de Regresión, Clasificación (menor a tres dimensiones), y Modelos de Agrupamiento, por mencionar algunas. En la sección 6.1.2 de este trabajo de investigación, se pueden observar dos Árboles de Decisión Naïve-Bayes aplicados a uno de los mejores individuos obtenidos a través del Algoritmo 'Attribute_Classification'.

#### 2.2.1.5 Difusión y Uso

Por último, la quinta fase es la *"Fase de Difusión y Uso"*. Su objetivo principal es hacer uso del nuevo conocimiento y se distribuye entre todos los usuarios que podrían obtener algún beneficio de él, como el tomar o recomendar acciones basándose en los resultados del modelo. Este conocimiento extraído a partir del KDD debe integrar el contexto del problema. También, puesto que la naturaleza de los datos cambia constantemente con el tiempo, es importante medir que tan bien evoluciona el modelo; aún cuando el modelo funcione adecuadamente, debemos continuamente monitorearlo, lo que significa que de tiempo en tiempo el modelo tendrá que ser re-evaluado, re-entrenado y posiblemente reconstruido completamente.

En relación al uso del conocimiento extraído del proceso de Minería de Datos, su principal utilidad radica en el soporte de la toma de decisiones, actuales o a futuro; así como, detectar nuevas tendencias positivas o negativas en el comportamiento de los datos. El conjunto de elementos que apoyan en la toma de decisiones [Mladenic et al. 2003] constituye un área multidisciplinaria cuyo objetivo es la introducción de métodos y/o herramientas que ayuden a los humanos en la toma de decisiones clave. El impacto de la información que se obtenga del proceso completo del KDD, estará basado en gran medida, en qué tan claro esté definido el objetivo final por parte del usuario para hacer uso de dicha información.

## 2.3 SELECCIÓN DE CARACTERÍSTICAS (FS)

Entendemos por Selección de Características, variables o atributos, llamada en inglés *Feature Selection*, a la búsqueda de subconjuntos de atributos que se realiza en el espacio de búsqueda para reducirlo, evaluando cada uno, conforme a una función objetivo ligada a una o varias clases objetivo determinadas, para las cuáles se desea la obtención de información [Pyle 1999].

El tamaño del espacio de búsqueda para la tarea de Selección de Características, es decir, el número de subconjuntos de atributos candidatos es de $2^n$, donde $n$ es el número total de atributos. Por tanto, el tamaño del espacio de búsqueda crece exponencialmente con el número de atributos. La justificación principal para realizar una tarea de selección de atributos, es lo que se le conoce como "La maldición de la dimensionalidad"; es decir, la alta dimensionalidad constituye un serio obstáculo para la eficiencia de la mayoría de los algoritmos de Minería de Datos.

La selección de los atributos, es un problema cercanamente relacionado a la reducción de la dimensionalidad. El objetivo de la selección de atributos es identificar los atributos más importantes del conjunto de datos, y descartar cualquier otro atributo irrelevante o con información redundante, ofreciendo una ejecución más rápida y una representación más compacta y fácil de interpretar del concepto perseguido [Hall 1999]. Por otro lado, la pérdida de información y el incremento en el tiempo de ejecución y los recursos que consume, son parte de las desventajas de hacer una selección de atributos. Además, de que elegir los atributos incorrectos pueden sesgar el modelo o proveer resultados incoherentes a la hora de predecir nuevas instancias.

La selección de atributos, tiene los siguientes objetivos principales:

1. Reducir el tamaño de los datos al eliminar características o atributos de todas las instancias que puedan ser irrelevantes o redundantes.

2. Una buena selección de características puede mejorar la calidad del modelo al permitir al método de minería de datos, centrarse en las características relevantes.
3. Permite expresar el modelo resultante en función de menos variables; lo cual es especialmente importante cuando se desean modelos comprensibles (árboles de decisión, regresión lineal, etc.).
4. Para facilitar la visualización de los datos, utilizando los dos o tres atributos más significativos.
5. Deshacerse más fácilmente de los atributos que contienen muchos datos erróneos o faltantes.
6. Deshacerse de atributos identificadores, los cuáles no son relevantes para las técnicas de minería de datos.

El problema de tener muchos atributos en una vista minable es, que la gran mayoría de métodos de minería de datos pueden *perderse* entre tantas características en un espacio que, al tener alta dimensionalidad, resulta estar más desierto y obtiene modelos ajustados a las particularidades de los datos de entrenamiento y no de los datos en general. Sin embargo, la gran ventaja de los métodos de reducción de dimensionalidad a través de la Selección de Características, es que el conjunto final de características es un subconjunto de los atributos originales y, por tanto, los modelos extraídos con posterioridad se definirán en función de los atributos originales, sin perder la comprensibilidad del modelo.

Algunos atributos son fáciles de eliminar, por ejemplo si un atributo es constante, es decir, tiene el mismo valor para todas las instancias es claramente eliminable. Existen dos reglas generales para eliminar características, en especial si los atributos son nominales:

- **Eliminación de claves candidatas:** La regla general es eliminar cualquier atributo que pueda ser clave primaria de la tabla, parcial o totalmente. Una manera sencilla de saber si un atributo nominal es demasiado específico y debe ser eliminado es ver si tiene casi tantos valores distintos como observaciones. Si no se elimina esta clase de atributos, puede ser especialmente problemático para tareas de clasificación o regresión.

- **Eliminación de atributos dependientes:** En una base de datos, cuando existen dependencias funcionales entre atributos se intenta normalizar en varias tablas. Por motivo de que los datos con los que trabajamos para minería de datos provienen de una *vista minable* que ha desnormalizado los datos, es necesario decidir si realmente se necesitan todos los atributos, ya que muchos de ellos son redundantes. No eliminar los atributos dependientes es especialmente nefasto a la hora de establecer reglas de asociación o para las tareas de agrupamiento.

Una vez que se han eliminado tanto las claves primarias como los atributos dependientes, es necesario eliminar otras características que no son relevantes ni redundantes. En general, detectados los atributos irrelevantes y redundantes, el siguiente paso sería utilizar técnicas más sofisticadas si queremos seguir reduciendo la dimensionalidad, en particular para los atributos numéricos. Existen dos tipos generales de métodos para seleccionar características:

- **Métodos de Filtrado:** Es una de las aproximaciones más antiguas, para la selección de atributos. Éstos métodos usan heurísticas basadas en características generales de los datos, en lugar de un algoritmo de aprendizaje para evaluar el método de los subconjuntos de atributos. Es decir, se filtran los atributos irrelevantes antes de cualquier proceso de minería de datos y, en cierto modo, independiente de él, sin tomar en cuenta el algoritmo de clasificación que será usado después. Las técnicas son fundamentalmente estadísticas ó heurísticas como: medidas de información, distancia, dependencia o inconsistencia. El criterio para establecer el subconjunto de características "óptimo" se basa en medidas de calidad previas que se calculan a partir de los mismos datos. El subconjunto seleccionado de atributos puede ser usado por diferentes algoritmos de clasificación y al final escoger el mejor, o una mezcla de ellos. Como consecuencia, estos métodos son mucho más rápidos que los de envoltura, y es mucho más práctico usarlos sobre las bases de datos con alta dimensionalidad.
- **Métodos basados en Modelo:** También llamados "de envolvente". Usan un algoritmo de inducción para estimar la precisión de los subconjuntos de atributos.

Aquí la selección de atributos es ejecutada tomando en cuenta el algoritmo de clasificación que será aplicado a los atributos seleccionados. Este algoritmo se ejecuta muchas veces con diferentes subconjuntos de atributos, y su desempeño en cada ejecución es usado para evaluar la calidad del correspondiente subconjunto de atributos. Entonces, el algoritmo de clasificación es visto como una caja negra. La justificación de este enfoque es que, el método de inducción que usará el subconjunto de atributos final, debe proveer una mejor estimación de la precisión, que una medida separada, que tiene un sesgo propio [Langley & Sage 1994]. Lógicamente, este tipo de técnicas requieren mucho más tiempo que las otras, ya que para evaluar hay que entrenar un modelo, ejecutando repetidamente el algoritmo de inducción; sin embargo, frecuentemente alcanzan mejores resultados que las de filtrado una vez que se sabe qué algoritmo de clasificación se usará, debido a la sintonía que existe entre el método de clasificación y el de selección. Este método implica cierta pérdida de generalidad, dado que los Atributos seleccionados son optimizados por un solo algoritmo de clasificación, aunque tiende a converger a una *Precisión Predictiva* más alta.

Para entrenar un modelo predictivo, normalmente se divide el conjunto completo de datos en dos o más subconjuntos, dependiendo del enfoque a seguir. En el enfoque de filtrado, se divide el conjunto de datos en un *conjunto de entrenamiento*, y otro de *prueba*, y al obtener el subconjunto de atributos más adecuado después de ejecutar el proceso de selección de atributos, se usa un método de inducción, con una partición distinta, para realizar la predicción. En cambio, en el enfoque de envoltura, se realiza una tercera partición; es decir, debido a que aquí no se puede utilizar un subconjunto de prueba para evaluar el subconjunto de atributos actual, es necesario dividir el subconjunto de entrenamiento en dos, es decir, el subconjunto de entrenamiento del conjunto de Entrenamiento, y el subconjunto de prueba del conjunto de Entrenamiento. Siendo estos dos utilizados por separado durante el procedimiento de la selección de características y juntos para el proceso del clasificador. El conocimiento descubierto por el algoritmo de clasificación es entonces evaluado sobre el conjunto de prueba, el cuál no ha sido visto durante el proceso completo de la Selección de Atributos. Una característica importante de estos dos métodos, es que

pueden ser iterativos, eliminando o recuperando atributos hasta que se obtiene una combinación que maximiza la precisión de la clasificación. La partición debe ser significativa del conjunto completo de instancias y aleatoria. Este tipo de segmentación, evita el '*sobreajuste*' del modelo, que es cuando el modelo da mejores resultados para el conjunto de entrenamiento que para el de prueba; no obstante, el modelo aún es demasiado dependiente del modo en el cual se ha realizado la partición de los subconjuntos

En general, el tipo de método de Selección de Atributos a ser usado depende totalmente de la tarea de Minería de Datos. Mientras que en el método de selección de atributos, la correlación de los atributos con el 'Outcome' es importante, en muchos de los modelos de minería de datos, se tratará de minimizar dicha correlación. Algunos de los métodos clásicos que se utilizan para establecer la independencia o importancia de las variables originales son el análisis correlacional, el análisis por modelo lineal, el análisis de frecuencias y el análisis de Correspondencias.

El sistema WEKA incorpora una gran cantidad de métodos para estudiar la relevancia de atributos y realizar una selección automática de los mismos, entre ellos podemos mencionar: RelieFAttributeEval, InfoGainAttributeEval, GainRatioAttributeEval, OneRAttributeEval, WrapperSubSetEval. Se habla más extensamente de este sistema en el Anexo B [Witten & Frank 2000].

### 2.3.1 La Maldición de la Dimensionalidad.

La *dimensionalidad* es el número de atributos, columnas ó variables (grados de libertad), que están disponibles para crear una predicción. En la práctica, los problemas de Minería de Datos son frecuentemente masivos en dimensión y computacionalmente intensos.

La alta dimensionalidad de la entrada de datos, incrementa el tamaño del espacio de búsqueda de una manera exponencial, aumentando también las posibilidades de que el algoritmo de minería de datos encuentre clasificaciones falsas que generalmente son inválidas. Además, es bien conocido que el número requerido de muestras para la

clasificación supervisada se incrementa como una función de la dimensionalidad [Jiménez & Landgrebe 1998]. Así, se ha estimado que, al aumentar el número de dimensiones, el tamaño de la muestra necesita incrementarse exponencialmente, para obtener una estimación efectiva de las densidades multivariadas [Hwang et al. 1994]. De esta forma, la dimensionalidad constituye un serio obstáculo para la eficiencia de la mayoría de los algoritmos de Minería de Datos. [Maimon & Last 2000]

La Maldición de la Dimensionalidad se refiere al crecimiento exponencial en los datos requeridos para poblar densamente un espacio, incrementándose la dimensión [Maimon & Rokach 2005b]. Un espacio 100-dimensional sería similar a galaxias distantes. La maldición de la dimensionalidad limita la habilidad práctica para ajustar un modelo flexible a datos con ruido, cuando existe un gran número de variables de entrada; esta limitante se encuentra tanto a nivel del software como a nivel del hardware, aunque menos común en este último. Un espacio de entrada densamente poblado es necesario para ajustar modelos altamente complejos. Cuando se valoran los datos que están disponibles para la minería de datos, se debe considerar, antes que cualquier cosa, la dimensión del problema. Asimismo, sabemos que las técnicas para el análisis de la información totalmente eficientes para bajas dimensiones, muchas veces no pueden proveer ningún resultado significativo cuando el número de variables va más allá de un modesto tamaño de 10 atributos. [Chizi & Maimon 2005]

En general, los procedimientos de Minería de Datos requieren un alto costo computacional al tratar con conjuntos de datos grandes [Chizi & Maimon 2005]. Al reducir el número de atributos, se puede reducir este costo de forma efectiva. La meta es: reducir la dimensionalidad con un mínimo de pérdida de información. Para evitar la 'maldición de la dimensionalidad', un modelo debe seleccionar entradas útiles; esto se puede hacer erradicando variables irrelevantes y redundantes. Reducir la dimensionalidad puede ayudar a lidiar con la maldición de la dimensionalidad; sin embargo, también puede ignorar información importante. Hoy en día, se han desarrollado una gran variedad de algoritmos con el fin de seleccionar las variables más apropiadas para la minería de datos, tal es el caso del presente trabajo de investigación.

### 2.3.2 Aplicación de los Algoritmos Genéticos dentro de la tarea de Selección de Atributos.

Cómo se dijo en el subcapítulo 2.1, los Algoritmos Genéticos son algoritmos de búsqueda estocástica que están basados en abstracciones del proceso evolutivo Darwiniano. Los AG se han aplicado en áreas como el descubrimiento de Reglas, Clustering, preparación de los datos, entre otras. En la mayoría de las aplicaciones de los algoritmos genéticos, la población es generada aleatoriamente, esto incrementa su grado de no-determinismo. Los individuos representan conjuntos de soluciones, por ejemplo, conjuntos de reglas, instancias, atributos, etc.

En el caso particular de la Reducción de Dimensionalidad, los Algoritmos Genéticos pueden ser adecuados para la Selección de Atributos debido a su habilidad para ejecutar una búsqueda global y lidian de mejor manera con la interacción entre atributos, en contraste con los métodos de selección de atributos con búsqueda convencional, secuencial y local. Además, los algoritmos genéticos son más flexibles, que los métodos de búsqueda convencional, por permitir naturalmente la evaluación de un subconjunto de atributos candidatos, basados en la combinación de diferentes criterios. Su espacio de búsqueda consiste de todos los posibles subconjuntos de atributos. La mayoría de los Algoritmos Genéticos siguen un enfoque de Envolvimiento.

La codificación estándar de un Algoritmo Genético Simple es una codificación binaria con una longitud de cadena fija; sin embargo, también existen otras formas de codificar un subconjunto de atributos. Su principal ventaja es su simplicidad, dado que cualquier operador de cruza y mutación servirá, aunque no cualquiera es la mejor opción para el problema a tratar.

Se sabe que los métodos de Selección de Atributos pueden ser clasificados dentro de los enfoques de envolvimiento y filtrado. En el enfoque de envolvimiento, la función de aptitud envuelve una medida de desempeño de un algoritmo clasificador, usando sólo el subconjunto de atributos seleccionados por el individuo del Algoritmo Genético

correspondiente. También es posible definir una función de aptitud que siga un enfoque de filtrado puro, ignorando la tasa de precisión del clasificador. Esta propuesta tiene la ventaja de reducir el tiempo de procesamiento de un Algoritmo Genético, debido a que en general, la computación de un criterio orientado a filtrado para un subconjunto de atributos es más rápido en su ejecución. Un AG tiene un aspecto de ambos enfoques, dependiendo de lo que la función de aptitud esta midiendo; si ésta envuelve una medida de desempeño entonces se trata del enfoque de envolvimiento, y si envuelve una medida más que es independiente del clasificador, también puede ser considerado dentro del enfoque de filtrado.

En [Freitas 2002] se encontró la tabla 2.3 donde se muestran algunos de los criterios usados en diversas investigaciones siguiendo un enfoque de envolvimiento para la Selección de Atributos:

| **Referencia** | **Algoritmo de Clasificación** | **Criterios de la Función de Aptitud** |
|---|---|---|
| Bala et al. 1995 | Árbol de Decisión | Precisión Predictiva, No. de atributos seleccionados |
| Bala et al. 1996 | Árbol de Decisión | Precisión Predictiva, No. de atributos seleccionados, Información del Contenido |
| Chen et al. 1999 | Tabla de Decisión Euclidiana | Precisión Predictiva, No. de atributos seleccionados |
| Cherkauer & Shavlik 1996 | Árbol de Decisión | Precisión Predictiva, No. de atributos seleccionados, tamaño medio del Árbol de Decisión |
| Terano & Ishino 1998 | Árbol de Decisión | Evaluación Subjetiva, Precisión Predictiva, Tamaño del Conjunto de Reglas |
| Vafaie & DeJong 1998 | Árbol de Decisión | Precisión Predictiva |
| Ishibuchi & Nakashima 2000 | Vecino más Cercano | Precisión Predictiva, No. de atributos seleccionados, No. de instancias seleccionadas |
| Emmanouilidis et al. 2000 | Redes Neuronales | Precisión Predictiva, No. de atributos seleccionados, Evaluación Multiobjetivo |

*Tabla 2.3 – Criterios usados en la Función de Aptitud para el Enfoque de Envolvimiento. [Freitas 2002]*

# CAPÍTULO III: DESARROLLO DEL MODELO.

## 3.1 DESCRIPCIÓN DEL MODELO.

El modelo que se presenta en este trabajo es de tipo predictivo y tiene como tarea la clasificación a través de un Algoritmo Genético, siguiendo un enfoque de envoltura. Este modelo opera sobre un dominio de dos clases encontrando, de un total de ***m*** individuos, un individuo o subconjunto óptimo de atributos representativos $\boldsymbol{A_j(k) = i}$, dentro de una base de datos con **n** columnas/atributos, que mejor clasifiquen al atributo clase u objetivo ***C***, en función del resto de los atributos ***(n-1)***, donde ***k*** es la posición dentro del individuo $\boldsymbol{A_j}$ que toma los valores de ***{1, … , p}*** con ***p = { 6, 8, 10 }***, e ***i*** es la codificación del atributo que toma los valores de ***{1, n-1}***. Se parte de la premisa que las bases de datos son numéricas, están limpias y no contienen ruido, la variable identificadora *id* o datos faltantes. El modelo divide los datos en dos subconjuntos, el conjunto de entrenamiento y el de prueba, para poder entrenar el modelo con el primero, y supervisar la eficiencia del mismo con el segundo. En la figura 3.1 se ilustra lo anterior:

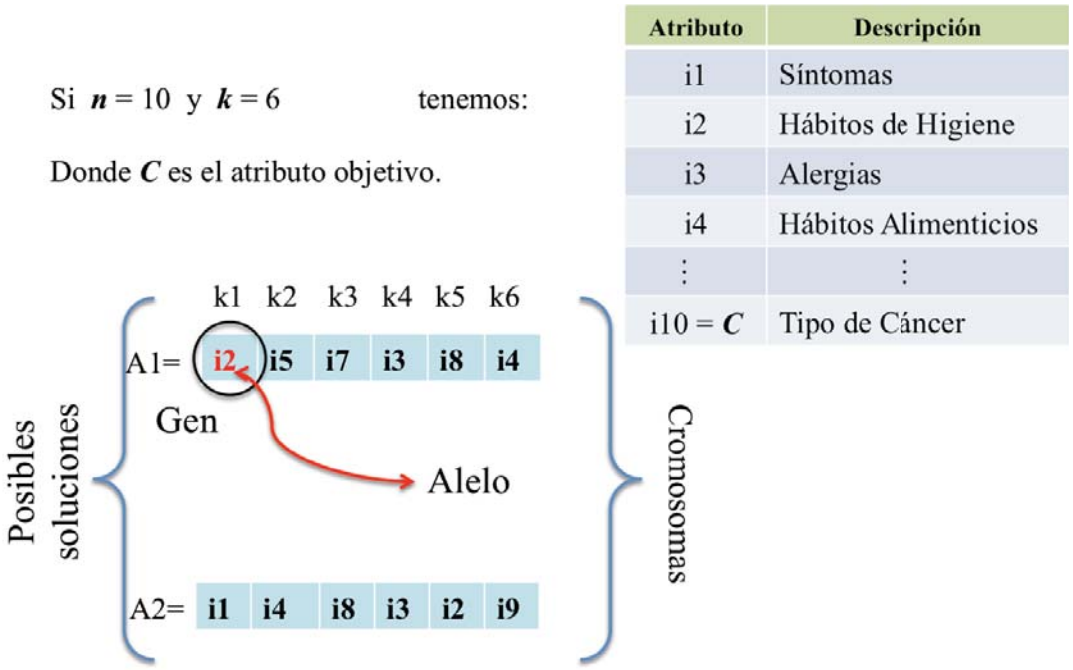


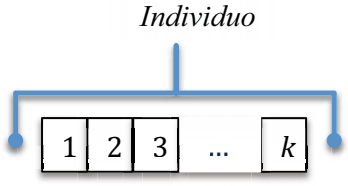

*Figura 3.1 – Representación Gráfica de la Decodificación.*

Los Algoritmos Genéticos son métodos de búsqueda colectiva en un espacio de soluciones y encuentran conjuntos de soluciones que, conforme pasan las iteraciones, van mejorando de generación en generación, porque los individuos que representan las soluciones más adecuadas al problema, tienen más posibilidades de sobrevivir. Un AG debe contener una representación adecuada de las posibles soluciones o conjuntos de soluciones del problema; una función de aptitud que establezca una forma de evaluar las posibles soluciones para saber cuál se adapta mejor al problema de optimización dado; y la definición de operadores de Selección, Cruza y Mutación que premien o beneficien la evolución de los mejores individuos, a través de herramientas como la aleatoriedad o la combinación de su material genético. En el subcapítulo 2.1.3 (página 13), se explica como funciona un AG.

### 3.1.1 Algoritmo Genético Planteado.

El Algoritmo Genético propuesto en este trabajo, presenta las siguientes características:

a) *Codificación:* La forma en que se representarán las posibles soluciones del problema será a través de la codificación entera o permutaciones sin reemplazo, siendo más conveniente esta representación, ya que el conjunto de características o atributos de una BD es un conjunto finito que puede ser cuantificado con números enteros [Whitley 1997; Angeline & Fogel 1997]. Por tanto, será representado en la computadora en un arreglo de tamaño $\boldsymbol{k}$, al cuál llamaremos '*Individuo*', donde $\boldsymbol{k}$ es un número relativamente pequeño, con $\boldsymbol{k \in \{1, \dots, p\}}$ y $\boldsymbol{p = \{ 6, 8, 10 \}}$, siendo $\boldsymbol{A_j(k)}$ el subconjunto de atributos más relevantes dentro de la vista minable. Cada atributo será un gen, por lo que cada individuo tendrá $\boldsymbol{k}$ genes. El experto decidirá el número de individuos que conformarán la población del AG.




$$\forall\ \boldsymbol{k} \in \mathbb{N}, \boldsymbol{k} \in \{\mathbf{1}, \dots, \boldsymbol{p}\} \text{ y } \boldsymbol{p} = \{\mathbf{6}, \mathbf{8}, \mathbf{10}\}$$

b) *Función Objetivo:* Todo problema de optimización esta expresado en una función objetivo que define las propiedades de un problema. Cuando la tarea de búsqueda es un problema de cobertura, la meta usualmente es encontrar un conjunto de puntos en *S* que juntos optimicen (maximicen o minimicen) alguna función, ó satisfagan algunas propiedades. Si se desea maximizar, la función entonces es una función de aptitud o utilidad; en caso contrario, su función es de costo o energía. La función objetivo debe reflejar los aspectos más relevantes del problema, estableciéndose las condiciones que restringen los resultados proporcionados por el algoritmo. De este modo, la función objetivo para el presente trabajo se encuentra definida de la siguiente manera:

$$\mathcal{O} : \mathbb{P}(\mathcal{S}) \rightarrow \mathcal{R}$$

donde $\mathbb{P}(\mathcal{S})$ es el conjunto que contiene todos los subconjuntos de soluciones expresados a través de alguna función de $\mathcal{S}$, llamada *Función de Aptitud* y asocia un valor del conjunto $\mathcal{R}$ con un subconjunto del espacio de búsqueda [Radcliffe 1997]. En el presente trabajo se pretende maximizar la suma de la Tasa de Precisión (T.P.) con una calificación obtenida en función del coeficiente de correlación (C.C.C.) de cada atributo dentro del individuo $\boldsymbol{A_j(k)}$ que clasifique correctamente una columna de dos clases ***C***, para el mayor número de instancias dentro de la vista minable; y el espacio de búsqueda esta conformado por todos los subconjuntos de atributos de tamaño $\boldsymbol{k_i}$, representados en el AG por la combinación de los números enteros, donde $i \in \{1, n-1\}$ y ***C*** es el atributo clasificador.

$$Optimizar \;\; \mathcal{O} : \mathbb{P}(\mathcal{S}) \implies Max: \;\; f_j = T.P + C.C.C.$$

$$s.a. \begin{cases} \overline{a_j} \neq \overline{\emptyset}, \qquad n \neq \emptyset \\ a_j(k) \leq \{i_1, i_2, \cdots, i_k\} \;\; donde \;\; i_k \neq i_{k+1} \\ 0 < k \leq p \\ p = \{6,8,10\} \\ i \in \{1, n-1\} \\ \bar{C} \in \{1,2\}, \bar{C} \neq \overline{\emptyset} \end{cases}$$

donde:

$$f_j = T.P + C.C.C.$$

$$TP_j = Tasa\ de\ Precisión = (VP + VN)/(VP + FP + FN + VN)$$

$$\forall\, TP_j,\ \ donde\ \ j \in (1, m),$$

$$\boldsymbol{j}\ es\ el\ individuo\ a\ evaluar$$

$y$ $\boldsymbol{m}$ es el número total de individuos, en cada generación.

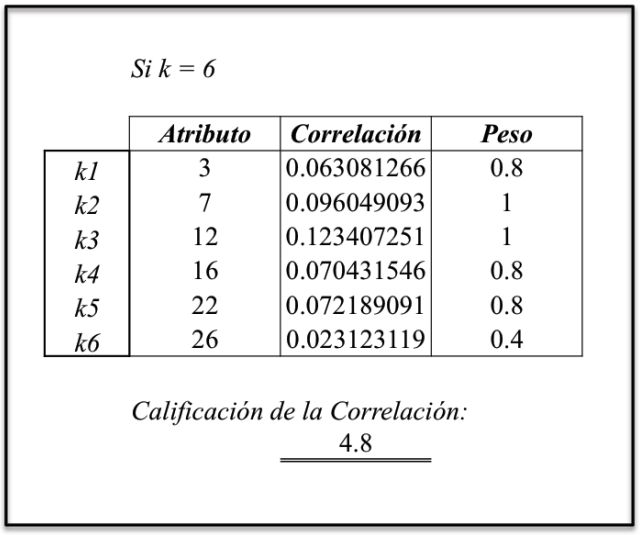
Y la calificación con base en el Coeficiente de Correlación (C.C.C.) se obtiene de la *Función CalcCorrelación* que otorga una *'calificación'* a cada individuo de una población, de acuerdo al coeficiente de correlación de Pearson, por cada atributo. Para hacerlo, se definieron diferentes pesos que cubrieron el rango de valores para las correlaciones de los atributos con la clase, en cada base de datos, y al final se sumaron dichos pesos de cada atributo, para obtener la calificación de la correlación por individuo. Dicha calificación es la que se busca maximizar dentro de la función de evaluación, al ser sumada con la Tasa de Precisión de un individuo determinado.

En la tabla 3.1 se explican los pesos de los intervalos para cada base de datos, y en la figura 3.2 se ilustra el proceso anterior:

| *BD Portafolios de Inversión* | | *BD WDBC* | | *BD WPBC* | |
|---|---|---|---|---|---|
| **Coeficientes de Correlación** | **Peso** | **Coeficientes de Correlación** | **Peso** | **Coeficientes de Correlación** | **Peso** |
| Correl <= 0 | 0.1 | Correl <= 0.1 | 0.1 | -0.01 < Correl < 0.01 | 0.1 |
| 0 < Correl <= 0.04 | 0.4 | 0.1 < Correl <= 0.3 | 0.4 | -0.05 < Correl <= -0.01 ó 0.01 <= Correl < 0.05 | 0.4 |
| 0.04 < Correl < 0.06 | 0.6 | 0.3 < Correl < 0.5 | 0.6 | -0.1 < Correl <= -0.05 ó 0.05 <= Correl < 0.1 | 0.6 |
| 0.06 <= Correl < 0.08 | 0.8 | 0.5 <= Correl < 0.7 | 0.8 | -0.2 < Correl <= -0.1 ó 0.1 <= Correl < 0.2 | 0.8 |
| Correl >= 0.08 | 1 | Correl >= 0.7 | 1 | -0.2 >= Correl >= 0.2 | 1 |

*Tabla 3.1 Pesos para los intervalos del coeficiente de correlación, de cada base de datos.*

Como se ve en la tabla 3.1, los pesos se obtuvieron en función de los rangos del coeficiente de correlación para cada base de datos, donde los atributos con mayor presencia tendrán más peso y los que son insignificantes, en relación al atributo categórico a predecir, "outcome", se castigan con un menor peso.



*Si k = 6*

| | *Atributo* | *Correlación* | *Peso* |
|---|---|---|---|
| *k1* | 3 | 0.063081266 | 0.8 |
| *k2* | 7 | 0.096049093 | 1 |
| *k3* | 12 | 0.123407251 | 1 |
| *k4* | 16 | 0.070431546 | 0.8 |
| *k5* | 22 | 0.072189091 | 0.8 |
| *k6* | 26 | 0.023123119 | 0.4 |

*Calificación de la Correlación:*
4.8

*Figura 3.2 – Ejemplo de la función para calcular la calificación de un individuo de la base de datos de Portafolios de Inversión.*

c) *Función de Aptitud* (Criterios de Evaluación): La función de aptitud es la función objetivo en los problemas de optimización. Ésta se debe maximizar o minimizar, encontrando valores para los diferentes parámetros que resulten óptimos al ser reemplazados en la función objetivo. El valor de la función de aptitud representa la calidad de la solución, la aptitud o el *fitness* de cada individuo. [Beasley et al. 1993] El éxito de un AG radica entre otros aspectos, de una buena elección de la función de aptitud. Una función de aptitud utilizada en el descubrimiento del conocimiento en una BD debe incorporar principalmente tres criterios: precisión, comprensibilidad e interés. [Hernández-Orallo et al. 2007] Idealmente, debe medir la calidad de un individuo de forma tan precisa como sea posible, sujeta a restricciones como la capacidad de procesamiento disponible o limitaciones propias del problema. La Función de Aptitud del algoritmo genético presentado en este trabajo, esta en función de la Aptitud Relativa y es representada de la siguiente forma:

$$F = Aptitud\ Total = \sum_{j=1}^{j=m} f_j$$

$$\therefore \qquad r_j = Aptitud\ Relativa = \frac{f_j}{F} \qquad donde \quad \sum_{j=1}^{j=m} r_j = 1$$

d) *Función de Clasificación:* Esta función clasifica cada registro de la base de datos, proporcionando una salida de 0 ó 1, que se utilizará en la obtención de la tasa de precisión, a través de la comparación con el atributo clase ó 'outcome' de la base de datos, de acuerdo con los siguientes pasos (Ver ejemplo en la figura 3.3):

1) Se obtiene la media y la desviación estándar de cada atributo perteneciente a cada individuo de la generación actual.
   (a) Regla 1: Si el valor en cada registro es <= a la media +/- la desviación estándar, es decir, si el valor se encuentra dentro del intervalo anterior, el resultado es 1, para el atributo en curso, en caso contrario es 0.
2) Una vez que se obtienen los 1's y 0's, se suman para obtener un valor para cada individuo. Posteriormente, se compara dicho valor con la columna de outcome de la siguiente manera:
   (a) Regla 2: Si el valor de la columna de salida 'outcome' es 1, entonces se realiza la siguiente comparación:
      (i) si el valor obtenido de la suma es mayor al número total de atributos dividido entre 2, la clasificación final es positivo, de otro modo es 0.
   (b) Si el valor de la columna de salida 'outcome' es 0, entonces se realiza la siguiente comparación:
      (i) si el valor obtenido de la suma es menor al número total de atributos dividido entre 2, la clasificación final es positivo, de otro modo es 0.

**Ejemplo:**

Número total de atributos: ***n=30*** / Número total de genes en el individuo: ***k=5***

Individuo:

| 8 | 21 | 4 | 9 | 13 |
|---|---|---|---|---|

Si $\mu j-\sigma j \geq ri,j \leq \mu j+\sigma j$ entonces $aj=1$. De otra forma, $aj=0$.

Para $j=4,8,9,13,21$, si $\mu j-\sigma j = 0.334$ y $\mu j+\sigma j = 0.856$, entonces:

| | 4 | 8 | 9 | 13 | 21 | Outcome |
|---|---|---|---|---|---|---|
| | 0.91 | 0.226 | 0.568 | 0.446 | 0.832 | 1 |
| **Nuevo individuo:** | 0 | 0 | 1 | 1 | 1 | |

Por lo tanto, *cont_Indiv_1=3*

Si *cont_Indiv_1* $>k/2$ ➔ $3>2.5$, entonces $C=1$

Por lo tanto,

| Outcome | Clasif_AG |
|---|---|
| 1 | 1 |

*Figura 3.3 – Ejemplo de la aplicación de la Función Clasificación.*

e) *Operador de Selección:* La selección esta estrictamente relacionada con el valor de aptitud (*fitness*) de cada individuo. El método escogido en este trabajo para llevar a cabo la selección de forma proporcional fue el Método de la Ruleta donde la selección es representativa y aleatoria, y donde se divide un círculo (ruleta) entre las probabilidades de selección de los individuos, es decir, su *aptitud relativa* $r_j$, una vez que han sido evaluados por la función de aptitud. Los rangos son definidos a través de la suma acumulada de probabilidades relativas, es decir, se agrupan los individuos de acuerdo a su valor de aptitud y se definen las probabilidades en relación a la cantidad de individuos que cada valor de aptitud agrupó. Una vez que se tienen los rangos, la selección sin reemplazo se lleva a cabo generando un número aleatorio entre 0 y 1, el cual se ubicará dentro del rango correspondiente, y será elegido un individuo del mismo, para pasar intacto a la siguiente generación. Cabe destacar, que el AG propuesto utiliza *elitismo*, para asegurar que el individuo con mejor valor de aptitud pueda pasar intacto, desde un principio, a la siguiente generación. Así es como se completa la población de la siguiente generación. [Deb 2000; Rodríguez-Vázquez et al. 2007]

f) *Operador de Cruza:* Es el más importante de los operadores de un AG, y trabaja por medio del intercambio de información genética entre pares de individuos. Normalmente, la *Probabilidad de Cruza 'Pc'* es cercana al 100%. En este trabajo se presenta un operador de cruza a dos puntos, porque garantiza el mayor movimiento de su material genético, y en donde dos puntos son seleccionados aleatoriamente en dos individuos cortando cada arreglo en tres segmentos. El primer y tercer segmento de ambos padres se copian directamente al primer y tercer segmento de los individuos "hijo", mientras que el segmento de en medio del padre 1 se copia al segmento de en medio del hijo 2 y el segmento de en medio del padre 2, se copia al segmento de en medio del hijo 1, de tal forma que de dos padres se obtienen, dos nuevos individuos hijos. El operador de cruza presentado no es sensible al orden, pero sí es sensible a los valores duplicados; por tanto, para nuestro problema en específico, esta función primero separa los duplicados, y copia el gen directamente a los nuevos individuos, y después, con las posiciones restantes, corta los dos

segmentos y continúa el procedimiento arriba descrito. A continuación se presenta la Figura 3.4 para ilustrar este operador:

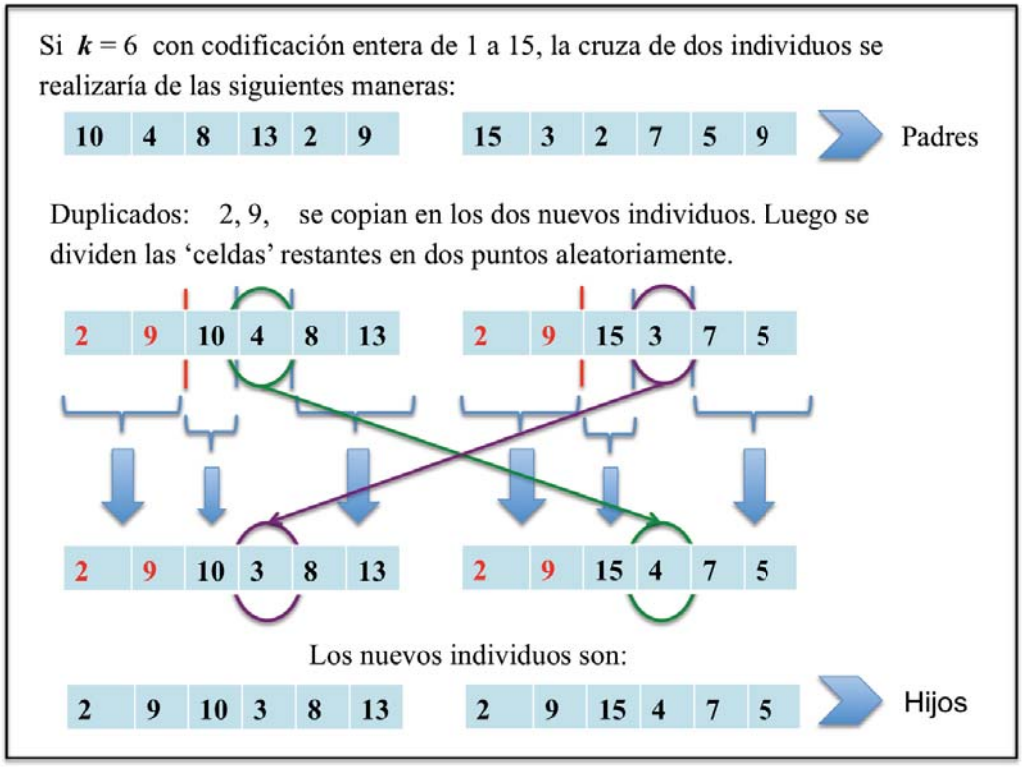



*Figura 3.4 – Operador de Cruza.*

g) *Operador de Mutación:* Este operador es una protección contra las pérdidas prematuras de material genético, como resultado de la reproducción y del operador de cruzamiento. Al mismo tiempo, intenta simular los cambios estocásticos y sorpresivos, dentro del proceso evolutivo de las especies. Generalmente, la *Probabilidad de Mutación 'Pm'* para cada gen del cromosoma es muy pequeña con el fin de que está no afecte significativamente a la población. La probabilidad de mutación puede ser constante a través de todas las generaciones ó, variar conforme van pasando las generaciones. En el caso del operador propuesto en este trabajo, se aplica en cada generación con una probabilidad constante menor al 5%. Se selecciona un gen al azar y se cambia su material genético por los posibles valores restantes. Esta función se cerciora de que no escoja el mismo valor o algún otro que

se encuentre dentro del mismo cromosoma, para evitar repeticiones. En la siguiente figura 3.5 se ilustra el operador de mutación:

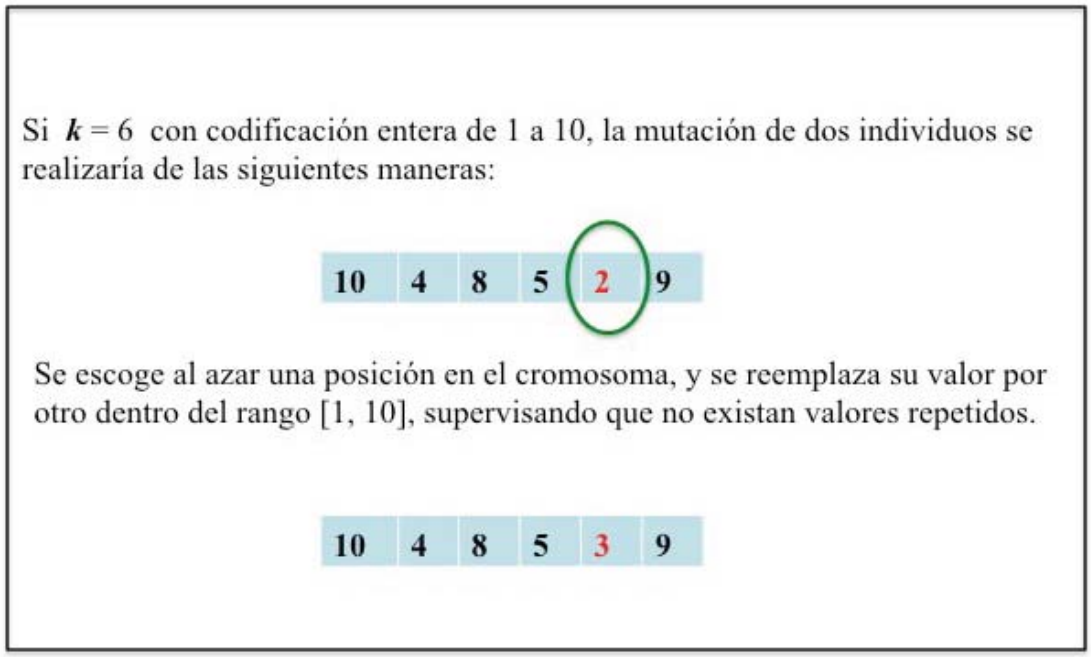



*Figura 3.5 – Operador de Mutación.*

h) *Función de Muestreo:* Para evitar el sobreajuste o sesgo de la predicción, el algoritmo divide el conjunto de datos en 2 subconjuntos, uno de entrenamiento y otro de prueba, de la siguiente manera: Se toma el 70% para entrenar el modelo y para obtener los mejores individuos, tras una serie de corridas (100). Y el 30% restante para probar al '*mejor individuo*'. Las muestras son aleatorias, independientes y representativas de las clases. El usuario puede modificar tanto los porcentajes, en la variable '*porcent_Entrenam*', como el número de corridas, en el archivo '*config.txt*', que se explica a continuación.

i) *Parámetros de Configuración:* Los parámetros más importantes y generales que nutren al modelo presentado, están contenidos en un archivo de texto que el usuario puede personalizar, llamado '*config.txt*'. En él se encontrarán los siguientes parámetros junto con sus valores por default:

| | *Parámetro* | *Valor por Default* | *Notas* |
|---|---|---|---|
| **Parámetros de la Base de Datos** | Database | *La base de datos anterior* | Ej. Data/BD_Portafolio.csv |
| | cont_atrib | *34* | Total de atributos |
| | Lugar_colum_outcome | *33* | Lugar que ocupa la columna de salida |
| | Lector_db | *'default'* | Sentencia que ordena que se lea la BD |
| | Omitir_columnas | *0* | Omite la columna del ID (el 0 cuenta como 1) |
| **Parámetros pre-modelo** | modo_Correlacion | *1* | Indica los criterios para calificar los individuos, de acuerdo al coef. de correlación. 1 – Portafolio, 2 – Wdbc, 3 – Wpbc |
| | muestra | *3* | Valores: 1 –directo, 2 –random, 3 –estadística |
| | porcent_Entrenam | *70* | Porcentaje de instancias para el conjunto de entrenamiento; el resto será para el conjunto de prueba. |
| | eval_cont | *100* | Indica cuantas veces se ejecutará el AG en el conjunto de entrenamiento. |
| **Parámetros del Algoritmo Genético** | salida_positiva | *1* | Puede ser diferente en otra BD (M = Maligno, R = Recurrente) |
| | salida_negativa | *0* | Puede ser diferente en otra BD (B = Benigno, N = No Recurrente) |
| | Probabilidad_Mutacion | *0.05* | |
| | Probabilidad_Cruza | *0.90* | |
| | Tamanio_Poblacion | *200* | |
| | Individuo_tamanio | *6* | Valor de ***k*** |
| | Cont_Max_Generacion | *1000* | Criterio de paro |
| **Salida de los Resultados** | Guardar_File | *0* | 0 – No guardar<br>1 – Guardar en 'Data/result.txt' |
| | File_Resultados | *Data/result.txt* | Nombre del archive de salida |

*Tabla 3.2 Valores por default en el archivo de lectura, 'config.txt'.*

# CAPÍTULO IV: RECOPILACIÓN, INTEGRACIÓN Y LIMPIEZA DE LOS DATOS.

Este proceso corresponde a la primera fase de la extracción de conocimiento de las bases de datos, conocida como: 'Integración y Recopilación de los datos'. A continuación se explicará la extracción y limpieza de las tres bases de datos con las cuales se exploró el sistema propuesto en este trabajo.

## 4.1 BASE DE DATOS FINANCIERA.

Se trata de obtener una base de datos financiera de los rendimientos de un grupo de acciones que cotizan en la BMV durante un periodo en específico, para calcular el rendimiento de acuerdo a un Portafolio de Inversión y poder 'probar' el algoritmo genético propuesto en este trabajo con dichos datos.

### 4.2.1 Recopilación.

Se recopilaron los precios diarios de cierre de las acciones de 40 empresas que cotizan en la BMV, tomados del Portal de Yahoo Finanzas durante el periodo del 1ro. de julio del 2008 al 29 de junio del 2012 [Yahoo Finanzas 2012]. La lista inicial de empresas es la siguiente:

1.- Alfa, S.A.B. de C.V. (ALFAA.MX)
2.- Alsea SAB de CV (ALSEA.MX)
3.- America Movil, S.A.B. de C.V. (AMXL.MX)
4.- Consorcio Ara SAB de CV (ARA.MX)
5.- Compañía Minera Autlan SAB de CV (AUTLANB.MX)
6.- Axtel SAB de CV (AXTELCPO.MX)
7.- Industrias Bachoco SAB de CV (BACHOCOB.MX)
8.- Grupo Bimbo SAB de CV (BIMBOA.MX)
9.- Cemex, S.A.B. de C.V. (CEMEXCPO.MX)
10.- Grupo, S.A.B. de C.V. (CIDMEGA.MX)
11.- Corporacion Moctezuma SAB de CV (CMOCTEZ.MX)
12.- Corporacion Mexicana de Restaurantes SAB de CV (CMRB.MX)
13.- Controladora Comercial Mexicana, S.A.B. de C.V. (COMERCIUBC.MX)
14.- Cydsa SAB de CV (CYDSASAA.MX)
15.- Grupo Elektra, S.A. de C.V. (ELEKTRA.MX)
16.- Fomento Economico Mexicano SAB de CV (FEMSAUBD.MX)
17.- Corporativo Fragua SAB de CV (FRAGUAB.MX)
18.- Grupo Carso, S.A.B. de C.V. (GCARSOA1.MX)
19.- Corporacion Geo, S.A.B. de C.V. (GEOB.MX)

20.- Grupo Financiero Inbursa, S.A.B. de C.V. (GFINBURO.MX)
21.- Grupo Financiero Banorte SAB de CV (GFNORTEO.MX)
22.- Grupo Mexicano de Desarrollo, S.A.B. (GMD.MX)
23.- Grupo Mexico, S.A.B. de C.V. (GMEXICOB.MX)
24.- Grupo Modelo, S.A.B. de C.V. (GMODELOC.MX)
25.- Grupo Herdez, S.A.B. de C.V. (HERDEZ.MX)
26.- Desarrolladora Homex SAB de CV (HOMEX.MX)
27.- Empresas ICA, S.A.B. de C.V. (ICA.MX)
28.- Kimberly - Clark de Mexico S.A.B. de C.V. (KIMBERA.MX)
29.- Coca-Cola Femsa, S.A.B. de C.V. (KOFL.MX)
30.- Grupo Lamosa, S.A.B. de C.V. (LAMOSA.MX)
31.- El Puerto de Liverpool, S.A.B. de C.V. (LIVEPOLC-1.MX)
32.- Mexichem, S.A.B. de C.V. (MEXCHEM.MX)
33.- Promotora Ambiental, S.A.B. de C.V. (PASAB.MX)
34.- Qualitas Compañia de Seguros, S.A. de C.V. (QCPO.MX)
35.- Telefonos de Mexico,S.A.B. de C.V. (TELMEXL.MX)
36.- Grupo Televisa, S.A. (TLEVISACPO.MX)
37.- Tenaris SA (TS.MX)
38.- Urbi Desarrollos Urbanos, S.A.B. de C.V. (URBI.MX)
39.- Value Grupo Financiero, S.A.B. de C.V. (VALUEGFO.MX)
40.- Wal - Mart de Mexico, S.A.B. de C.V. (WALMEXV.MX)

### 4.2.2 Integración y Limpieza:

A continuación, se realizan simultáneamente los subprocesos de integración y limpieza de los datos, con el objetivo de obtener la base de datos adecuada para trabajar con el sistema propuesto en esta tesis.

Para comenzar, se eliminaron 8 de las 40 empresas, bajo el criterio de contar con más de 100 registros faltantes, es decir, cada una de las 8 empresas abajo enlistadas, cuenta con menos de 900 registros para el experimento.

| | | |
|---|---|---|
| *CIDMEGA.MX* | *GMD.MX* | *TS.MX* |
| *CMRB.MX* | *LAMOSA.MX* | *VALUEGFO.MX* |
| *FRAGUAB.MX* | *PASAB.MX* | |

Después, se grafica cada empresa para ver el comportamiento de sus datos a través del tiempo. A continuación se presentan las gráficas de algunas empresas tomadas al azar con su respectiva tendencia durante el periodo del 1ro. de julio del 2008 al 29 de junio del 2012. (Figuras 4.1, 4.2 y 4.3):

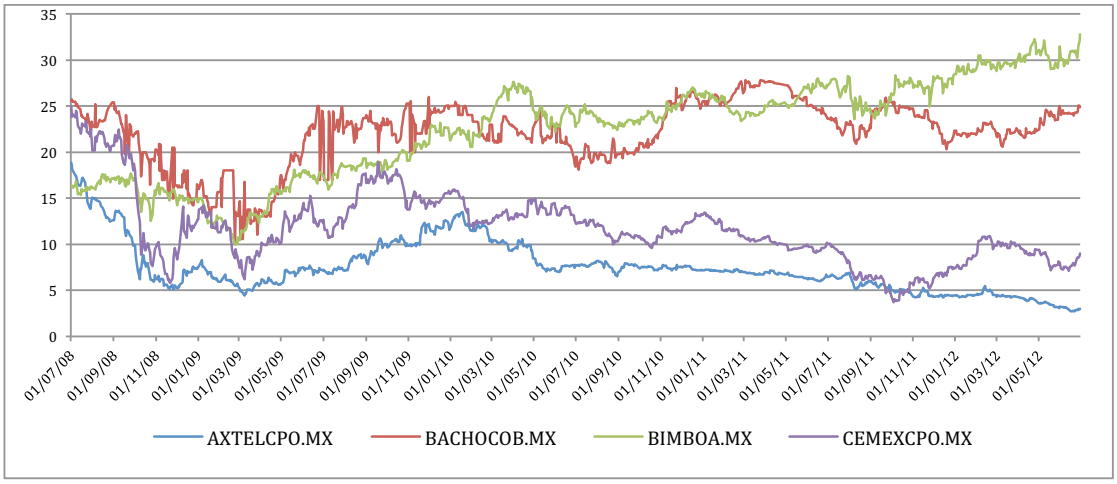



*Figura 4.1 - Precios al cierre de las acciones de las empresas AXTELCPO, BACHOCOB, BIMBOA y CEMEXCPO, que cotizan en la Bolsa Mexicana de Valores.*

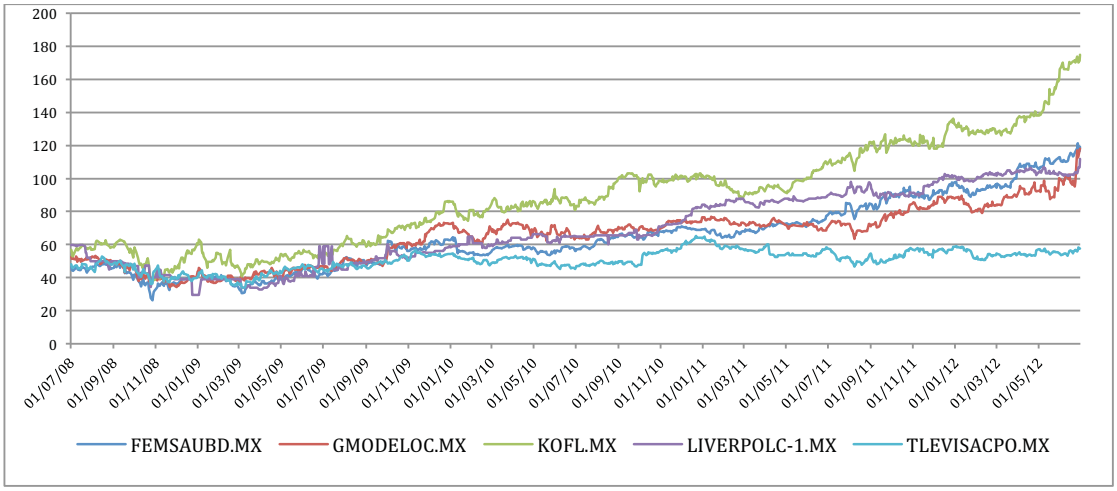



*Figura 4.2 – Precios al cierre de las acciones de las empresas FEMSAUBD, GMODELOC, KOFL, LIVERPOLC-1 y TLEVISACPO, que cotizan en la Bolsa Mexicana de Valores.*

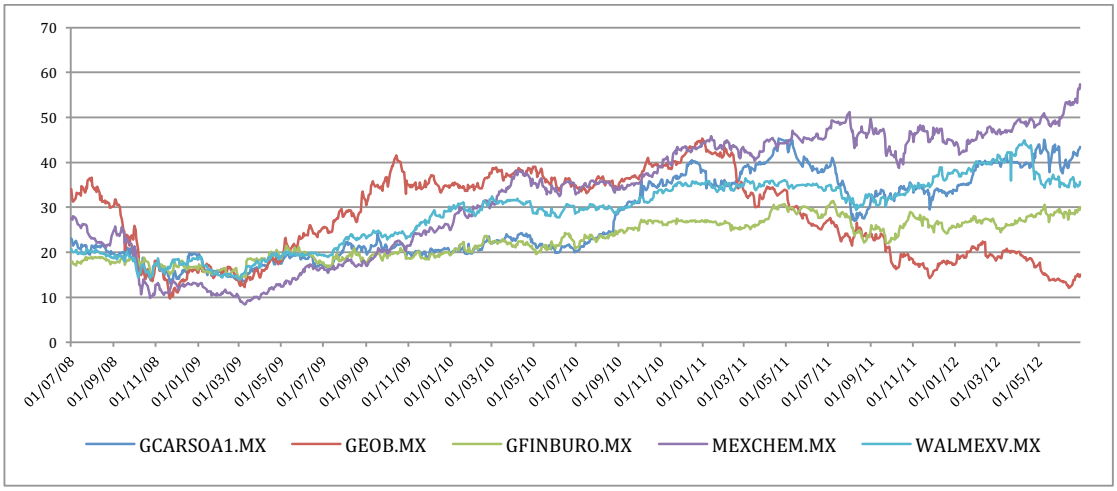



*Figura 4.3 – Precios al cierre de las acciones de las empresas GCARSOA1, GEOB, GFINBURO, MEXCHEM y WALMEXV, que cotizan en la Bolsa Mexicana de Valores.*

Dentro de los criterios de Integración de los datos recopilados, es preciso añadir aquellos datos faltantes para todas las empresas de acuerdo a los días laborados por la mayoría, a fin de conseguir una base de datos de calidad, con los precios de cierre de las empresas, con la cuál se pretende obtener los rendimientos de un portafolio de 32 empresas. La mayoría de las empresas tienen datos 'en blanco' o campos vacíos en diferentes fechas durante el periodo del 1ro. de julio de 2008 al 29 de junio del 2012. Por tanto, un criterio a tomar es, si se encuentra un dato faltante, se escribe el precio de cierre del día anterior. No es posible poner '0' porque ello significaría una variación drástica en la tendencia de los precios. Sin embargo, repetir el precio del cierre anterior, supondría que no hubo movimientos para esa empresa en específico, en ese día en específico.

La lista final de empresas que constituirán la BD final, dentro de un periodo del 1ro. de julio de 2008 al 29 de junio del 2012, con un total de 1020 filas y 32 atributos, es:

1.- Alfa, S.A.B. de C.V. (ALFAA.MX)
2.- Alsea SAB de CV (ALSEA.MX)
3.- America Movil, S.A.B. de C.V. (AMXL.MX)
4.- Consorcio Ara SAB de CV (ARA.MX)
5.- Compañía Minera Autlan SAB de CV (AUTLANB.MX)
6.- Axtel SAB de CV (AXTELCPO.MX)
7.- Industrias Bachoco SAB de CV (BACHOCOB.MX)
8.- Grupo Bimbo SAB de CV (BIMBOA.MX)
9.- Cemex, S.A.B. de C.V. (CEMEXCPO.MX)

10.- Corporacion Moctezuma SAB de CV (CMOCTEZ.MX)
11.- Controladora Comercial Mexicana, S.A.B. de C.V. (COMERCIUBC.MX)
12.- Cydsa SAB de CV (CYDSASAA.MX)
13.- Grupo Elektra, S.A. de C.V. (ELEKTRA.MX)
14.- Fomento Economico Mexicano SAB de CV (FEMSAUBD.MX)
15.- Grupo Carso, S.A.B. de C.V. (GCARSOA1.MX)
16.- Corporacion Geo, S.A.B. de C.V. (GEOB.MX)
17.- Grupo Financiero Inbursa, S.A.B. de C.V. (GFINBURO.MX)
18.- Grupo Financiero Banorte SAB de CV (GFNORTEO.MX)
19.- Grupo Mexico, S.A.B. de C.V. (GMEXICOB.MX)
20.- Grupo Modelo, S.A.B. de C.V. (GMODELOC.MX)
21.- Grupo Herdez, S.A.B. de C.V. (HERDEZ.MX)
22.- Desarrolladora Homex SAB de CV (HOMEX.MX)
23.- Empresas ICA, S.A.B. de C.V. (ICA.MX)
24.- Kimberly - Clark de Mexico S.A.B. de C.V. (KIMBERA.MX)
25.- Coca-Cola Femsa, S.A.B. de C.V. (KOFL.MX)
26.- El Puerto de Liverpool, S.A.B. de C.V. (LIVEPOLC-1.MX)
27.- Mexichem, S.A.B. de C.V. (MEXCHEM.MX)
28.- Qualitas Compañia de Seguros, S.A. de C.V. (QCPO.MX)
29.- Telefonos de Mexico,S.A.B. de C.V. (TELMEXL.MX)
30.- Grupo Televisa, S.A. (TLEVISACPO.MX)
31.- Urbi Desarrollos Urbanos, S.A.B. de C.V. (URBI.MX)
32.- Wal - Mart de Mexico, S.A.B. de C.V. (WALMEXV.MX)

### 4.2.3 Procedimiento para la obtención de la BD de Portafolios de Inversión.

Una vez que se ha obtenido una BD depurada, es preciso empezar el procedimiento para la obtención de la última base de datos que probará el sistema propuesto. Esta BD, contendrá los rendimientos de un portafolios conformado por las 32 empresas seleccionadas en el subcapítulo anterior durante el periodo del 1ro. de julio de 2008 al 29 de junio del 2012. En lo sucesivo, se presenta dicho procedimiento:

Se ordenan cronológicamente los precios de las 32 empresas seleccionadas, creando una matriz de precios diarios ***P*** de (1021x32), como se muestra a continuación:

$$P_{1021,32} = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & \cdots & p_{1,32} \\ p_{2,1} & p_{2,2} & p_{2,3} & \cdots & p_{2,32} \\ p_{3,1} & p_{3,2} & p_{3,3} & \cdots & p_{3,32} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{1021,1} & p_{1021,2} & p_{1021,3} & \cdots & p_{1021,32} \end{bmatrix}$$

Se obtiene el rendimiento diario de cada empresa, utilizando la siguiente expresión matemática, tomada de [Ross et al. 2000]:

$$R_{i+1,j} = \frac{P_{i+1,j} - P_{i,j}}{P_{i,j}} \quad para \quad i = 1,2,\dots 1020 \quad y \quad j = 1,2,\dots,32$$

Obteniendo la siguiente matriz de rendimiento diario ***R*** de 1020X32:

$$R_{1020,32} = \begin{bmatrix} r_{2,1} & r_{2,2} & r_{2,3} & \cdots & r_{2,32} \\ r_{3,1} & r_{3,2} & r_{3,3} & \cdots & r_{3,32} \\ r_{4,1} & r_{4,2} & r_{4,3} & \cdots & r_{4,32} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{1021-1,1} & r_{1021-1,2} & r_{1021-1,3} & \cdots & r_{1021-1,32} \end{bmatrix}$$

Cabe señalar que, en este punto, se siguió la siguiente estrategia a nivel instancia: Se eliminó la primera fila correspondiente al 1ro. de julio del 2008, a causa de que se obtiene un valor de rendimiento desde la segunda instancia. Además, esta matriz puede contener rendimientos negativos, por lo que para obtener la matriz de pesos ***W*** que determinará el rendimiento del portafolio, es necesario calcular primero la matriz de rendimientos desplazados $\widetilde{\boldsymbol{R}}_{\boldsymbol{1020,32}}$, de la siguiente forma: Se busca el valor mínimo de la matriz de rendimientos diarios ***min***, se multiplica por 2 en valor absoluto y se le suma dicho valor a todos los valores de $R_{i,j}$, es decir: $\tilde{r}_{i,j} = r_{i,j} + |min| * 2$

Para obtener la matriz de pesos ***W*** que determinará los pesos que cada empresa tendrá diariamente para invertir en el portafolio, se suman los rendimientos de las empresas por día, $S_i = \sum_{j=1}^{32} R_{i,j} \ \ con \ \ i = 1,2,3,\cdots,1020$. Los pesos de los rendimientos se determinan así: $w_{i,j} = \frac{R_{i,j}}{S_i}$. La matriz de pesos queda de la siguiente manera:

$$W_{1020,32} = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & \cdots & w_{1,32} \\ w_{2,1} & w_{2,2} & w_{2,3} & \cdots & w_{2,32} \\ w_{3,1} & w_{3,2} & w_{3,3} & \cdots & w_{3,32} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{1020,1} & w_{1020,2} & w_{1020,3} & \cdots & w_{1020,32} \end{bmatrix}$$

La matriz de pesos para la base de datos de Portafolios de Inversión se encuentra en el archivo *'Obtencion_BD_Portafolio.xlsx'* y en la tabla 4.1 tenemos un ejemplo de la matriz de pesos ***W***:

| Date | 1) ALFAA.MX | 2) ALSEA.MX | 3) AMXL.MX | 4) ARA.MX | 5) AUTLANB.MX | 6) AXTELCPO.MX | 7) BACHOCOB.MX | 8) BIMBO.MX | 9) CEMEXCPO.MX |
|---|---|---|---|---|---|---|---|---|---|
| 01/07/08 | | | | | | | | | |
| 02/07/08 | 0.03152112 | 0.03173860 | 0.03125191 | 0.03133019 | 0.03072456 | 0.03109798 | 0.03165878 | 0.03155049 | 0.03056876 |
| 03/07/08 | 0.03112944 | 0.03156620 | 0.03141802 | 0.03134837 | 0.03104915 | 0.03095112 | 0.03127489 | 0.03126053 | 0.03179415 |
| 04/07/08 | 0.03113160 | 0.03130027 | 0.03104427 | 0.03109479 | 0.03147436 | 0.03123836 | 0.03125010 | 0.03118487 | 0.03113654 |
| 07/07/08 | 0.03058455 | 0.03086274 | 0.03122687 | 0.03135742 | 0.03189666 | 0.03056954 | 0.03117074 | 0.03168823 | 0.03091614 |
| 08/07/08 | 0.03063880 | 0.03135859 | 0.03138060 | 0.03092409 | 0.03186901 | 0.03116539 | 0.03078398 | 0.03157213 | 0.03182862 |
| 09/07/08 | 0.03132176 | 0.03097996 | 0.03060696 | 0.03188093 | 0.03115939 | 0.03100792 | 0.03155482 | 0.03060352 | 0.03118560 |
| 10/07/08 | 0.03140385 | 0.03106764 | 0.03168129 | 0.03118980 | 0.03165270 | 0.03111403 | 0.03149932 | 0.03122350 | 0.03093720 |
| 11/07/08 | 0.03131282 | 0.03071920 | 0.03135859 | 0.03126131 | 0.03144045 | 0.03101089 | 0.03117883 | 0.03064356 | 0.03079213 |
| 14/07/08 | 0.03129349 | 0.03073164 | 0.03130985 | 0.03068661 | 0.03124066 | 0.03107531 | 0.03112340 | 0.03154071 | 0.03086825 |
| ⋮ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ |

*Tabla 4.1 – Matriz de Pesos para la base de datos de Portafolios de Inversión.*

Una vez que se tiene la matriz de pesos $\boldsymbol{W}$, se puede calcular el rendimiento del portafolio diario $\boldsymbol{Rp_i}$ (para i=1,…,1020 días), a través de la siguiente expresión:

$$Rp_i = \sum_{j=1}^{32} \tilde{r}_{i,j} \cdot w_{i,j} \quad , \quad \text{con i} = 1, \cdots, 1020,$$

$$y \;\; j = 1, \cdots, 32 \;\; empresas.$$

Es decir, se obtiene el producto interno del vector-renglón de la matriz de rendimientos $\boldsymbol{R}$, por el vector-renglón correspondiente de la matriz de pesos $\boldsymbol{W}$ para cada día, dando como resultado un vector de $Rp_{1020,1}$.

Bajo la comparación diaria entre el rendimiento del portafolios de inversión con 32 empresas y el rendimiento diario de la Bolsa Mexicana de Valores, se determinará un vector booleano diagnóstico $\boldsymbol{D}$ de (1020x1), que calificará al portafolio como 'Aceptable' o 'No Aceptable', siguiendo la regla de asociación que a continuación se presenta:

- Si el rendimiento del portafolio diario es mayor al Rendimiento promedio obtenido por la BMV en ese día en específico, es aceptado (buen rendimiento),
- de otra forma es rechazado (mal rendimiento), es decir:

$$if(Rp_i > Rd)\ then\ 1, else\ 0.$$

El valor 1 en la columna de salida (outcome), significa un valor de rendimiento mayor al rendimiento de referencia, que en este caso es el rendimiento de la BMV; por lo que puede haber un Portafolio con menos empresas que mejore dicho rendimiento.

Por último, se eliminaron 20 filas que contenían el valor de 0 en todas las casillas, viendo que ninguna empresa tuvo movimientos esos días y el rendimiento para las 32 empresas fue de 0; por tanto, se obtuvo una base de datos final de 1000 registros y 33 atributos que contiene el rendimiento diario de un portafolio de inversión con 32 empresas; así como una columna clase de salida (outcome) que 'aprueba' o no el portafolio en un día en específico.

La lista de fechas concernientes a estas 20 filas eliminadas se presenta a continuación:

| **No. Reg.** | **Fecha** | **No. Reg.** | **Fecha** | **No. Reg.** | **Fecha** |
|---|---|---|---|---|---|
| 55 | 16 de Sept. 2008 | 202 | 9 de Abr. 2009 | 876 | 12 de Dic. 2011 |
| 99 | 17 de Nov. 2008 | 203 | 10 de Abr. 2009 | 916 | 6 de Feb. 2012 |
| 118 | 12 de Dic. 2008 | 349 | 2 de Nov. 2009 | 946 | 19 de Mar. 2012 |
| 127 | 25 de Dic. 2008 | 414 | 1ro. De Feb. 2010 | 959 | 5 de Abr. 2012 |
| 132 | 1ro. De Ene 2009 | 458 | 2 de Abr. 2010 | 960 | 6 de Abr. 2012 |
| 154 | 2 de Feb. 2009 | 578 | 17 de Sept. 2010 | 977 | 1ro. De May. 2012 |
| 184 | 16 de Mar. 2009 | 861 | 21 de Nov. 2011 | | |

*Tabla 4.2 – Lista de registros borrados.*

Finalmente, la vista minable que alimentará al sistema propuesto en este trabajo de tesis, estará compuesta por 34 atributos con los rendimientos diarios de las 32 empresas seleccionadas, la columna de fecha como identificador (útil sólo como referencia), y una columna diagnóstico que servirá como guía para obtener el valor de aptitud de la vista minable. Esta base de datos se puede encontrar con el nombre de '*BD_Portafolio.csv*' y su Diccionario de Datos se puede encontrar en el Anexo F.

## 4.2 BASES DE DATOS MÉDICAS: WDBC Y WPBC.

Los registros de ambas bases de datos, WDBC y WPBC, fueron obtenidos de la investigación hecha por los científicos de la Universidad de Wisconsin, Dr. William H. Wolberg, W. Nick Street y Olvi L. Mangasarian, llamada "Wisconsin Prognostic Breast Cancer (WPBC)" en el año de 1995. El Diccionario de Datos de ambas bases se pueden encontrar en el Anexo F.

La primera base de datos, Winsconsin Diagnostic Breast Cancer (WDBC), [Wolberg et al. 1995], esta compuesta por 569 instancias, y 32 atributos con los siguientes nombres:

| | | |
|---|---|---|
| 1. Id_number | 12. Dimensión_Fractal_promedio | 23. Peor/más_largo_Radio |
| 2. Salida | 13. Error_Estandar_Radio | 24. Peor/más_larga_Textura |
| 3. Radio_promedio | 14. Error_Estandar_Textura | 25. Peor/más_largo_Perimetro |
| 4. Textura_promedio | 15. Error_Estandar_Perimetro | 26. Peor/más_largo_Area |
| 5. Perimetro_promedio | 16. Error_Estandar_Area | 27. Peor/más_largo_Suavidad |
| 6. Area_promedio | 17. Error_Estandar_Suavidad | 28. Peor/más_largo_Compacidad |
| 7. Suavidad_promedio | 18. Error_Estandar_Compacidad | 29. Peor/más_largo_Concavidad |
| 8. Compacidad_promedio | 19. Error_Estandar_Concavidad | 30. Peor/más_largo_Punto_Concavo |
| 9. Concavidad_promedio | 20.Error_Estandar_Puntos_Concavos | 31. Peor/más_largo_Simetría |
| 10. Puntos_Concavos_promedio | 21. Error_Estandar_Simetría | 32.Peor/más_largo_Dimensión_Fractal |
| 11. Simetría_promedio | 22. Error_Estand_Dimensión_Fractal | |

La base de datos 'WDBC' se puede encontrar de forma electrónica bajo el nombre de '*wdbc.csv*'. Dado que el primer atributo (Id_number) trata de un paciente en específico, el programa no lo toma en cuenta y trabaja con un total de 31 atributos. El atributo de salida, cuenta con dos tipos de registros: B = Benigno y M = Maligno, de los cuales existe un total de 357 y 212 registros respectivamente.

Por su parte, la segunda base de datos, Wisconsin Prognostic Breast Cancer (WPBC) [Wolberg et al. 1995a], esta compuesta por 198 instancias, 35 atributos y existen 4 datos faltantes en el atributo 'Estado del Nodo Linfático', que se trataron como 0´s (moda), debido a que eliminar cada observación sería muy costoso para la investigación, dada la

escasez de datos y a que se perdería información importante. La lista de atributos es la siguiente:

1. Id_number
2. Salida
3. Tiempo
4. Radio_promedio
5. Textura_promedio
6. Perimetro_promedio
7. Area_promedio
8. Suavidad_promedio
9. Compacidad_promedio
10. Concavidad_promedio
11. Puntos_Concavos_promedio
12. Simetría_promedio
13. Dimensión_Fractal_promedio
14. Error_Estandar_Radio
15. Error_Estandar_Textura
16. Error_Estandar_Perimetro
17. Error_Estandar_Area
18. Error_Estandar_Suavidad
19. Error_Estandar_Compacidad
20. Error_Estandar_Concavidad
21. Error_Estandar_Puntos_Concavos
22. Error_Estandar_Simetría
23. Error_Estand_Dimensión_Fractal
24. Peor/más_largo_Radio
25. Peor/más_larga_Textura
26. Peor/más_largo_Perimetro
27. Peor/más_largo_Area
28. Peor/más_largo_Suavidad
29. Peor/más_largo_Compacidad
30. Peor/más_largo_Concavidad
31. Peor/más_largo_Punto_Concavo
32. Peor/más_largo_Simetría
33. Peor/más_largo_Dimensión_Fractal
34. Tamaño del Tumor
35. Estado del Nodo Linfático

Nuevamente, el primer atributo que es el Id del paciente, es ignorado, así que se trabaja con un total de 34 atributos. El atributo de salida, “outcome”, es tomado como atributo clasificador, y cuenta con dos tipos de registros: N = no recurrente y R = Recurrente, de los cuales existe un total de 151 y 47 registros respectivamente. Esta base de datos se puede encontrar en el archivo con nombre ‘*wpbc.csv*’.

# CAPÍTULO V: EVALUACIÓN DEL MODELO Y RESULTADOS.

## 5.1 RESULTADOS DEL MODELO.

Una vez que ya se cuenta con las bases de datos de aplicación, integradas y limpias de atributos erróneos, ID's y faltantes, se procede a aplicar el algoritmo genético propuesto en este trabajo, llamado "Attribute_Classification". Las bases de datos usadas en este trabajo de investigación se pueden encontrar en el Anexo E.

Se trabajó con la siguiente serie de parámetros:

- La Base de Datos
- Número de Atributos **k**
- # de Iteraciones
- Población (Total de Individuos)
- %'s de subconjuntos de Entrenamiento y Prueba
- # de Generaciones dentro del AG
- % de Cruza
- % de Mutación

Se hicieron 16 diferentes pruebas con diferentes tamaños de individuos, cambiando los valores de los parámetros; los únicos parámetros que se mantuvieron constantes fueron el tamaño de la población (total de individuos) con 200 individuos, y los porcentajes de los subconjuntos de entrenamiento y prueba, los cuáles fueron 70/30, respectivamente. Los demás, tomaron los siguientes valores:

| **Bases de Datos (3)** | | |
|---|---|---|
| *BD_Portafolio* | *WDBC* | *WPBC* |
| **k** = 10, 8 y 6 | **k** = 10, 8 y 6 | **k** = 10, 8 y 6 |
| **# de Iteraciones** = 50, 100 | | **% de Cruza** = 90 y 95. |
| **# de Generaciones** = 500, 1000 | | **% de Mutación** = 3 y 5. |

*Tabla 5.1 – Valores de los parámetros para cada ejecución del AG.*

A continuación se presenta una serie de tablas que resumen los criterios tomador para evaluar los resultados del modelo para cada base de datos, presentando los parámetros bajo los cuales se ejecutó el sistema, el tiempo de ejecución que le tomó, el número total de mejores individuos, la tasa de precisión y el ó los mejor(es) individuo(s) en las tablas 5.2, 5.3 y 5.4; y añadiendo en las tablas de la 5.5 a la 5.13, el porcentaje de error al clasificar

dichos individuos en el conjunto de prueba, y los valores de aptitud, verdaderos positivos, verdaderos negativos, falsos positivos, falsos negativos, tasa de valores positivos, tasa de valores negativos, precisión y la calificación de la correlación para cada mejor individuo, al ser analizado en el conjunto de entrenamiento.

### 5.1.1 Base de Datos de Portafolio de Inversión.

Para la base de datos 'BD_Portafolio.csv' con 32 atributos, el atributo ID y la clase 1 y 0, una población de 200 individuos, y la división del 70% de los datos para el conjunto de entrenamiento y el resto para el de prueba. Se realizaron un total de 48 ejecuciones, con k = 10, 8 y 6:

| ID | K | ID Corrida | Entren / Prueba | Poblac | Total Iterac. | # Generac. | % Cruza | % Mutac. | Tiempo Ejecución | # mejores individuos |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10 | 1 | 70/30 | 200 | 50 | 500 | 90% | 5% | 7 min. 46 seg. | 1 |
| 2 | 10 | 2 | 70/30 | 200 | 50 | 1000 | 90% | 5% | 16 min. 13 seg. | 1 |
| 3 | 10 | 3 | 70/30 | 200 | 100 | 500 | 90% | 5% | 15 min. 56 seg. | 1 |
| 4 | 10 | 4 | 70/30 | 200 | 100 | 1000 | 90% | 5% | 31 min. 32 seg. | 3 |
| 5 | 10 | 5 | 70/30 | 200 | 50 | 500 | 95% | 5% | 8 min. 17 seg. | 1 |
| 6 | 10 | 6 | 70/30 | 200 | 50 | 1000 | 95% | 5% | 16 min. 15 seg. | 1 |
| 7 | 10 | 7 | 70/30 | 200 | 100 | 500 | 95% | 5% | 16 min. 2 seg. | 1 |
| 8 | 10 | 8 | 70/30 | 200 | 100 | 1000 | 95% | 5% | 33 min. 58 seg. | 1 |
| 9 | 10 | 9 | 70/30 | 200 | 50 | 500 | 90% | 3% | 8 min. 1 seg. | 1 |
| 10 | 10 | 10 | 70/30 | 200 | 50 | 1000 | 90% | 3% | 15 min. 47 seg. | 1 |
| 11 | 10 | 11 | 70/30 | 200 | 100 | 500 | 90% | 3% | 15 min. 52 seg. | 1 |
| **12** | **10** | **12** | **70/30** | **200** | **100** | **1000** | **90%** | **3%** | **38 min. 48 seg.** | **1** |
| 13 | 10 | 13 | 70/30 | 200 | 50 | 500 | 95% | 3% | 8 min. 27 seg. | 1 |
| 14 | 10 | 14 | 70/30 | 200 | 50 | 1000 | 95% | 3% | 18 min. 43 seg. | 1 |
| 15 | 10 | 15 | 70/30 | 200 | 100 | 500 | 95% | 3% | 15 min. 47 seg. | 1 |
| 16 | 10 | 16 | 70/30 | 200 | 100 | 1000 | 95% | 3% | 31 min. 53 seg. | 3 |
| 17 | 8 | 1 | 70/30 | 200 | 50 | 500 | 90% | 5% | 6 min. 48 seg. | 1 |
| 18 | 8 | 2 | 70/30 | 200 | 50 | 1000 | 90% | 5% | 13 min. 41 seg. | 1 |
| 19 | 8 | 3 | 70/30 | 200 | 100 | 500 | 90% | 5% | 13 min. 18 seg. | 1 |
| 20 | 8 | 4 | 70/30 | 200 | 100 | 1000 | 90% | 5% | 28 min. 40 seg. | 1 |
| 21 | 8 | 5 | 70/30 | 200 | 50 | 500 | 95% | 5% | 6 min. 58 seg. | 1 |
| 22 | 8 | 6 | 70/30 | 200 | 50 | 1000 | 95% | 5% | 13 min. 24 seg. | 1 |
| **23** | **8** | **7** | **70/30** | **200** | **100** | **500** | **95%** | **5%** | **14 min. 7 seg.** | **1** |
| 24 | 8 | 8 | 70/30 | 200 | 100 | 1000 | 95% | 5% | 29 min. 11 seg. | 1 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 25 | 8 | 9 | 70/30 | 200 | 50 | 500 | 90% | 3% | 6 min. 46 seg. | 1 |
| 26 | 8 | 10 | 70/30 | 200 | 50 | 1000 | 90% | 3% | 13 min. 27 seg. | 1 |
| 27 | 8 | 11 | 70/30 | 200 | 100 | 500 | 90% | 3% | 13 min. 30 seg. | 1 |
| 28 | 8 | 12 | 70/30 | 200 | 100 | 1000 | 90% | 3% | 26 min. 58 seg. | 1 |
| 29 | 8 | 13 | 70/30 | 200 | 50 | 500 | 95% | 3% | 6 min. 48 seg. | 1 |
| 30 | 8 | 14 | 70/30 | 200 | 50 | 1000 | 95% | 3% | 13 min. 30 seg. | 1 |
| 31 | 8 | 15 | 70/30 | 200 | 100 | 500 | 95% | 3% | 13 min. 35 seg. | 1 |
| 32 | 8 | 16 | 70/30 | 200 | 100 | 1000 | 95% | 3% | 27 min. 6 seg. | 1 |
| 33 | 6 | 1 | 70/30 | 200 | 50 | 500 | 90% | 5% | 5 min. 29 seg. | 1 |
| **34** | **6** | **2** | **70/30** | **200** | **50** | **1000** | **90%** | **5%** | **10 min. 52 seg.** | **1** |
| 35 | 6 | 3 | 70/30 | 200 | 100 | 500 | 90% | 5% | 11 min. 6 seg. | 2 |
| 36 | 6 | 4 | 70/30 | 200 | 100 | 1000 | 90% | 5% | 24 min. 15 seg. | 2 |
| 37 | 6 | 5 | 70/30 | 200 | 50 | 500 | 95% | 5% | 5 min. 28 seg. | 2 |
| 38 | 6 | 6 | 70/30 | 200 | 50 | 1000 | 95% | 5% | 13 min. 12 seg. | 1 |
| 39 | 6 | 7 | 70/30 | 200 | 100 | 500 | 95% | 5% | 11 min. 18 seg. | 2 |
| 40 | 6 | 8 | 70/30 | 200 | 100 | 1000 | 95% | 5% | 22 min. 47 seg. | 3 |
| 41 | 6 | 9 | 70/30 | 200 | 50 | 500 | 90% | 3% | 6 min. 12 seg. | 2 |
| 42 | 6 | 10 | 70/30 | 200 | 50 | 1000 | 90% | 3% | 11 min. 42 seg. | 1 |
| 43 | 6 | 11 | 70/30 | 200 | 100 | 500 | 90% | 3% | 12 min. 15 seg. | 2 |
| 44 | 6 | 12 | 70/30 | 200 | 100 | 1000 | 90% | 3% | 23 min. | 1 |
| 45 | 6 | 13 | 70/30 | 200 | 50 | 500 | 95% | 3% | 6 min. 11 seg. | 1 |
| 46 | 6 | 14 | 70/30 | 200 | 50 | 1000 | 95% | 3% | 11 min. 40 seg. | 2 |
| 47 | 6 | 15 | 70/30 | 200 | 100 | 500 | 95% | 3% | 11 min. 38 seg. | 3 |
| 48 | 6 | 16 | 70/30 | 200 | 100 | 1000 | 95% | 3% | 21 min. 46 seg. | 5 |

*Tabla 5.2 – Resumen de los resultados obtenidas por el AG durante 48 ejecuciones para la BD del Portafolios de Inversión.*

### 5.1.2 Base de Datos Médica de Wisconsin Diagnosis Breast Cancer 'WDBC'.

Para la base de datos médica 'wdbc.csv' con 30 atributos, el atributo ID y la clase 'B' para Benigno y 'M' maligno, una población de 200 individuos, y la división del 70% de los datos para el conjunto de entrenamiento y el resto para el de prueba. Se realizaron un total de 48 ejecuciones, con k = 10, 8, 6:

| ID | K | ID Corrida | Entren / Prueba | Poblac | Total de Iterac. | # Generac. | % Cruza | % Mutac. | Tiempo Ejecución | # mejores individuos |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10 | 1 | 70/30 | 200 | 50 | 500 | 90% | 5% | 3 min. 14 seg. | 1 |
| 2 | 10 | 2 | 70/30 | 200 | 50 | 1000 | 90% | 5% | 6 min. 25 seg. | 2 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 10 | 3 | 70/30 | 200 | 100 | 500 | 90% | 5% | 6 min. 24 seg. | 1 |
| 4 | 10 | 4 | 70/30 | 200 | 100 | 1000 | 90% | 5% | 12 min. 58 seg. | 1 |
| 5 | 10 | 5 | 70/30 | 200 | 50 | 500 | 95% | 5% | 3 min. 13 seg. | 1 |
| 6 | 10 | 6 | 70/30 | 200 | 50 | 1000 | 95% | 5% | 6 min. 27 seg. | 1 |
| 7 | 10 | 7 | 70/30 | 200 | 100 | 500 | 95% | 5% | 6 min. 26 seg. | 1 |
| 8 | 10 | 8 | 70/30 | 200 | 100 | 1000 | 95% | 5% | 12 min. 45 seg. | 1 |
| 9 | 10 | 9 | 70/30 | 200 | 50 | 500 | 90% | 3% | 3 min. 15 seg. | 1 |
| 10 | 10 | 10 | 70/30 | 200 | 50 | 1000 | 90% | 3% | 6 min. 24 seg. | 1 |
| 11 | 10 | 11 | 70/30 | 200 | 100 | 500 | 90% | 3% | 6 min. 26 seg. | 2 |
| 12 | 10 | 12 | 70/30 | 200 | 100 | 1000 | 90% | 3% | 12 min. 34 seg. | 2 |
| 13 | 10 | 13 | 70/30 | 200 | 50 | 500 | 95% | 3% | 3 min. 18 seg. | 2 |
| 14 | 10 | 14 | 70/30 | 200 | 50 | 1000 | 95% | 3% | 6 min. 27 seg. | 2 |
| 15 | 10 | 15 | 70/30 | 200 | 100 | 500 | 95% | 3% | 6 min. 35 seg. | 1 |
| **16** | **10** | **16** | **70/30** | **200** | **100** | **1000** | **95%** | **3%** | **12 min. 36 seg.** | **1** |
| 17 | 8 | 1 | 70/30 | 200 | 50 | 500 | 90% | 5% | 2 min. 45 seg. | 1 |
| 18 | 8 | 2 | 70/30 | 200 | 50 | 1000 | 90% | 5% | 5 min. 32 seg. | 1 |
| 19 | 8 | 3 | 70/30 | 200 | 100 | 500 | 90% | 5% | 5 min. 33 seg. | 1 |
| 20 | 8 | 4 | 70/30 | 200 | 100 | 1000 | 90% | 5% | 11 min. 2 seg. | 1 |
| 21 | 8 | 5 | 70/30 | 200 | 50 | 500 | 95% | 5% | 2 min. 47 seg. | 1 |
| 22 | 8 | 6 | 70/30 | 200 | 50 | 1000 | 95% | 5% | 5 min. 32 seg. | 1 |
| 23 | 8 | 7 | 70/30 | 200 | 100 | 500 | 95% | 5% | 5 min. 24 seg. | 1 |
| **24** | **8** | **8** | **70/30** | **200** | **100** | **1000** | **95%** | **5%** | **11 min. 8 seg.** | **1** |
| 25 | 8 | 9 | 70/30 | 200 | 50 | 500 | 90% | 3% | 2 min. 46 seg. | 2 |
| 26 | 8 | 10 | 70/30 | 200 | 50 | 1000 | 90% | 3% | 5 min. 30 seg. | 1 |
| 27 | 8 | 11 | 70/30 | 200 | 100 | 500 | 90% | 3% | 5 min. 26 seg. | 1 |
| 28 | 8 | 12 | 70/30 | 200 | 100 | 1000 | 90% | 3% | 11 min. 6 seg. | 1 |
| 29 | 8 | 13 | 70/30 | 200 | 50 | 500 | 95% | 3% | 2 min. 45 seg. | 1 |
| 30 | 8 | 14 | 70/30 | 200 | 50 | 1000 | 95% | 3% | 5 min. 26 seg. | 2 |
| 31 | 8 | 15 | 70/30 | 200 | 100 | 500 | 95% | 3% | 5 min. 32 seg. | 1 |
| 32 | 8 | 16 | 70/30 | 200 | 100 | 1000 | 95% | 3% | 10 min. 55 seg. | 2 |
| 33 | 6 | 1 | 70/30 | 200 | 50 | 500 | 90% | 5% | 2 min. 13 seg. | 1 |
| 34 | 6 | 2 | 70/30 | 200 | 50 | 1000 | 90% | 5% | 4 min. 17 seg. | 1 |
| 35 | 6 | 3 | 70/30 | 200 | 100 | 500 | 90% | 5% | 4 min. 22 seg. | 1 |
| 36 | 6 | 4 | 70/30 | 200 | 100 | 1000 | 90% | 5% | 8 min. 53 seg. | 1 |
| 37 | 6 | 5 | 70/30 | 200 | 50 | 500 | 95% | 5% | 2 min. 10 seg. | 1 |
| 38 | 6 | 6 | 70/30 | 200 | 50 | 1000 | 95% | 5% | 4 min. 20 seg. | 1 |
| 39 | 6 | 7 | 70/30 | 200 | 100 | 500 | 95% | 5% | 4 min. 21 seg. | 1 |
| 40 | 6 | 8 | 70/30 | 200 | 100 | 1000 | 95% | 5% | 8 min. 55 seg. | 1 |
| 41 | 6 | 9 | 70/30 | 200 | 50 | 500 | 90% | 3% | 2 min. 12 seg. | 1 |
| 42 | 6 | 10 | 70/30 | 200 | 50 | 1000 | 90% | 3% | 4 min. 17 seg. | 1 |
| 43 | 6 | 11 | 70/30 | 200 | 100 | 500 | 90% | 3% | 4 min. 22 seg. | 1 |
| 44 | 6 | 12 | 70/30 | 200 | 100 | 1000 | 90% | 3% | 8 min. 42 seg. | 1 |
| 45 | 6 | 13 | 70/30 | 200 | 50 | 500 | 95% | 3% | 2 min. 12 seg. | 2 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 46 | 6 | 14 | 70/30 | 200 | 50 | 1000 | 95% | 3% | 4 min. 30 seg. | 1 |
| 47 | 6 | 15 | 70/30 | 200 | 100 | 500 | 95% | 3% | 4 min. 28 seg. | 1 |
| **48** | **6** | **16** | **70/30** | **200** | **100** | **1000** | **95%** | **3%** | **8 min. 48 seg.** | **1** |

*Tabla 5.3 – Resumen de los resultados obtenidas por el AG durante 48 ejecuciones para la BD de WDBC.*

### 5.1.3 Base de Datos Médica de Wisconsin Prognosis Breast Cancer 'WPBC'.

Para la base de datos médica 'wpbc.csv' con 33 atributos, el atributo ID y la clase 'N' No Recurrente y 'R' para Recurrente, una población de 200 individuos, y la división del 70% de los datos para el conjunto de entrenamiento y el resto para el de prueba. Se realizaron un total de 48 ejecuciones, con k = 10, 8, 6:

| ID | K | ID Corrida | Entren/ Prueba | Poblac | Total de Iterac. | # Generac. | % Cruza | % Mutac. | Tiempo Ejecución | # mejores individuos |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10 | 1 | 70/30 | 200 | 50 | 500 | 90% | 5% | 4 min. 6 seg. | 1 |
| 2 | 10 | 2 | 70/30 | 200 | 50 | 1000 | 90% | 5% | 8 min. 15 seg. | 2 |
| 3 | 10 | 3 | 70/30 | 200 | 100 | 500 | 90% | 5% | 8 min. 4 seg. | 1 |
| 4 | 10 | 4 | 70/30 | 200 | 100 | 1000 | 90% | 5% | 16 min. 45 seg. | 1 |
| 5 | 10 | 5 | 70/30 | 200 | 50 | 500 | 95% | 5% | 4 min. 11 seg. | 1 |
| 6 | 10 | 6 | 70/30 | 200 | 50 | 1000 | 95% | 5% | 8 min. 14 seg. | 3 |
| 7 | 10 | 7 | 70/30 | 200 | 100 | 500 | 95% | 5% | 8 min. 16 seg. | 1 |
| 8 | 10 | 8 | 70/30 | 200 | 100 | 1000 | 95% | 5% | 16 min. 34 seg. | 1 |
| 9 | 10 | 9 | 70/30 | 200 | 50 | 500 | 90% | 3% | 4 min. 8 seg. | 1 |
| 10 | 10 | 10 | 70/30 | 200 | 50 | 1000 | 90% | 3% | 8 min. 15 seg. | 1 |
| 11 | 10 | 11 | 70/30 | 200 | 100 | 500 | 90% | 3% | 8 min. 21 seg. | 1 |
| **12** | **10** | **12** | **70/30** | **200** | **100** | **1000** | **90%** | **3%** | **16 min. 13 seg.** | **1** |
| 13 | 10 | 13 | 70/30 | 200 | 50 | 500 | 95% | 3% | 4 min. 13 seg. | 1 |
| 14 | 10 | 14 | 70/30 | 200 | 50 | 1000 | 95% | 3% | 8 min. 32 seg. | 3 |
| 15 | 10 | 15 | 70/30 | 200 | 100 | 500 | 95% | 3% | 8 min. 11 seg. | 2 |
| 16 | 10 | 16 | 70/30 | 200 | 100 | 1000 | 95% | 3% | 16 min. 22 seg. | 2 |
| 17 | 8 | 1 | 70/30 | 200 | 50 | 500 | 90% | 5% | 3 min. 32 seg. | 3 |
| 18 | 8 | 2 | 70/30 | 200 | 50 | 1000 | 90% | 5% | 7 min. 1 seg. | 1 |
| 19 | 8 | 3 | 70/30 | 200 | 100 | 500 | 90% | 5% | 6 min. 53 seg. | 1 |
| 20 | 8 | 4 | 70/30 | 200 | 100 | 1000 | 90% | 5% | 14 min. 44 seg. | 1 |
| 21 | 8 | 5 | 70/30 | 200 | 50 | 500 | 95% | 5% | 3 min. 42 seg. | 1 |
| 22 | 8 | 6 | 70/30 | 200 | 50 | 1000 | 95% | 5% | 7 min. 23 seg. | 1 |
| 23 | 8 | 7 | 70/30 | 200 | 100 | 500 | 95% | 5% | 8 min. 37 seg. | 2 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 24 | 8 | 8 | 70/30 | 200 | 100 | 1000 | 95% | 5% | 14 min. 27 seg. | 1 |
| 25 | 8 | 9 | 70/30 | 200 | 50 | 500 | 90% | 3% | 3 min. 47 seg. | 3 |
| 26 | 8 | 10 | 70/30 | 200 | 50 | 1000 | 90% | 3% | 7 min. 4 seg. | 1 |
| 27 | 8 | 11 | 70/30 | 200 | 100 | 500 | 90% | 3% | 7 min. 35 seg. | 1 |
| **28** | **8** | **12** | **70/30** | **200** | **100** | **1000** | **90%** | **3%** | **14 min. 18 seg.** | **1** |
| 29 | 8 | 13 | 70/30 | 200 | 50 | 500 | 95% | 3% | 4 min. 10 seg. | 1 |
| 30 | 8 | 14 | 70/30 | 200 | 50 | 1000 | 95% | 3% | 8 min. | 1 |
| 31 | 8 | 15 | 70/30 | 200 | 100 | 500 | 95% | 3% | 7 min. 19 seg. | 2 |
| 32 | 8 | 16 | 70/30 | 200 | 100 | 1000 | 95% | 3% | 13 min. 40 seg. | 1 |
| 33 | 6 | 1 | 70/30 | 200 | 50 | 500 | 90% | 5% | 2 min. 53 seg. | 1 |
| 34 | 6 | 2 | 70/30 | 200 | 50 | 1000 | 90% | 5% | 5 min. 40 seg. | 1 |
| **35** | **6** | **3** | **70/30** | **200** | **100** | **500** | **90%** | **5%** | **5 min. 35 seg.** | **1** |
| **36** | **6** | **4** | **70/30** | **200** | **100** | **1000** | **90%** | **5%** | **11 min. 34 seg.** | **1** |
| 37 | 6 | 5 | 70/30 | 200 | 50 | 500 | 95% | 5% | 2 min. 53 seg. | 1 |
| 38 | 6 | 6 | 70/30 | 200 | 50 | 1000 | 95% | 5% | 5 min. 47 seg. | 1 |
| 39 | 6 | 7 | 70/30 | 200 | 100 | 500 | 95% | 5% | 5 min. 44 seg. | 1 |
| 40 | 6 | 8 | 70/30 | 200 | 100 | 1000 | 95% | 5% | 11 min. 26 seg. | 1 |
| 41 | 6 | 9 | 70/30 | 200 | 50 | 500 | 90% | 3% | 2 min. 52 seg. | 3 |
| 42 | 6 | 10 | 70/30 | 200 | 50 | 1000 | 90% | 3% | 5 min. 39 seg. | 2 |
| 43 | 6 | 11 | 70/30 | 200 | 100 | 500 | 90% | 3% | 6 min. 12 seg. | 1 |
| 44 | 6 | 12 | 70/30 | 200 | 100 | 1000 | 90% | 3% | 11 min. 9 seg. | 1 |
| **45** | **6** | **13** | **70/30** | **200** | **50** | **500** | **95%** | **3%** | **2 min. 58 seg.** | **1** |
| 46 | 6 | 14 | 70/30 | 200 | 50 | 1000 | 95% | 3% | 5 min. 43 seg. | 1 |
| 47 | 6 | 15 | 70/30 | 200 | 100 | 500 | 95% | 3% | 5 min. 53 seg. | 1 |
| 48 | 6 | 16 | 70/30 | 200 | 100 | 1000 | 95% | 3% | 11 min. 31 seg. | 1 |

*Tabla 5.4 – Resumen de los resultados obtenidas por el AG durante 48 ejecuciones para la BD de WPBC.*

## 5.2 EVALUACIÓN DE LOS RESULTADOS.

A continuación se presentarán los resultados obtenidos en este trabajo de investigación, para las tres bases de datos con las que se trabajó.

Los criterios sobre los que se evaluó el mejor individuo, de acuerdo al análisis de verdaderos positivos, fueron la tasa de precisión, calificación de la correlación, confianza, y especificidad, sobre el conjunto de entrenamiento; y el error al clasificar nuevas observaciones sobre el conjunto de prueba.

Para la base de datos de Portafolios de Inversión, se presentan los siguientes resultados, para k=10, 8 y 6:

a) Para un individuo con 10 genes (atributos) con 16 ejecuciones:

| ID | # Gener | % Cruza | % Mut. | # MI | Mejor Indiv. del AG en cto. de entrenam. | % de Error | Tasa de Prec. | CCC | VP | VN | FP | FN | TVP | TVN | Precis. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 500 | 90% | 5% | 1 | 4.7.12.17.19.22.25.28.29.30 | 8.0268 | 0.9211 | 9.8 | 329 | 313 | 17 | 38 | 0.8965 | 0.9485 | 0.9509 |
| 2 | 1000 | 90% | 5% | 1 | 4.7.12.15.17.18.19.21.29.30 | 8.3612 | 0.9197 | 9.8 | 327 | 314 | 16 | 40 | 0.8910 | 0.9515 | 0.9534 |
| 3 | 500 | 90% | 5% | 1 | 1.4.7.12.15.17.19.20.21.30 | 8.6957 | 0.9268 | 9.8 | 330 | 316 | 14 | 37 | 0.8992 | 0.9576 | 0.9593 |
| 4 | 1000 | 90% | 5% | 3 | 3.7.12.17.19.20.24.28.29.32 | 7.3579 | 0.9182 | 9.8 | 330 | 310 | 20 | 37 | 0.8992 | 0.9394 | 0.9429 |
| | | | | | 3.7.12.17.19.20.24.25.28.29 | 7.0234 | 0.9182 | 9.8 | 329 | 311 | 19 | 38 | 0.8965 | 0.9424 | 0.9454 |
| | | | | | **7.12.14.15.19.20.24.25.28.29** | **6.6890** | **0.9182** | **9.6** | **328** | **312** | **18** | **39** | **0.8937** | **0.9455** | **0.9480** |
| **5** | **500** | **95%** | **5%** | **1** | **1.7.12.18.19.20.21.29.30.32** | **6.6890** | **0.9211** | **9.8** | **330** | **312** | **18** | **37** | **0.8992** | **0.9455** | **0.9483** |
| 6 | 1000 | 95% | 5% | 1 | 1.7.12.15.19.20.21.24.25.29 | 7.6923 | 0.9182 | 9.8 | 326 | 314 | 16 | 41 | 0.8883 | 0.9515 | 0.9532 |
| **7** | **500** | **95%** | **5%** | **1** | **1.4.7.12.18.20.24.25.28.29** | **6.6890** | **0.9225** | **9.8** | **331** | **312** | **18** | **36** | **0.9019** | **0.9455** | **0.9484** |
| 8 | 1000 | 95% | 5% | 1 | 1.4.7.12.15.19.24.28.29.32 | 8.6957 | 0.9211 | 9.8 | 328 | 314 | 16 | 39 | 0.8937 | 0.9515 | 0.9535 |
| 9 | 500 | 90% | 3% | 1 | 4.7.12.13.17.18.20.21.22.30 | 8.0268 | 0.9197 | 9.8 | 327 | 314 | 16 | 40 | 0.8910 | 0.9515 | 0.9534 |
| **10** | **1000** | **90%** | **3%** | **1** | **1.7.12.18.19.20.21.29.30.32** | **6.6890** | **0.9211** | **9.8** | **330** | **312** | **18** | **37** | **0.8992** | **0.9455** | **0.9483** |
| **11** | **500** | **90%** | **3%** | **1** | **1.4.7.12.20.22.24.28.29.30** | **6.3545** | **0.9225** | **9.8** | **328** | **315** | **15** | **39** | **0.8937** | **0.9545** | **0.9563** |
| 12 | 1000 | 90% | 3% | 1 | 1.7.12.17.19.20.21.24.29.30 | 7.3579 | 0.9297 | 9.8 | 331 | 317 | 13 | 36 | 0.9019 | 0.9606 | 0.9622 |
| 13 | 500 | 95% | 3% | 1 | 1.4.7.12.17.18.20.21.22.25 | 8.0268 | 0.9268 | 9.8 | 328 | 318 | 12 | 39 | 0.8937 | 0.9636 | 0.9647 |
| **14** | **1000** | **95%** | **3%** | **1** | **1.4.7.12.13.15.20.21.24.30** | **6.3545** | **0.9268** | **9.8** | **329** | **317** | **13** | **38** | **0.8965** | **0.9606** | **0.9620** |
| 15 | 500 | 95% | 3% | 1 | 1.3.4.7.12.19.24.28.29.32 | 9.0301 | 0.9225 | 9.8 | 329 | 314 | 16 | 38 | 0.8965 | 0.9515 | 0.9536 |
| 16 | 1000 | 95% | 3% | 3 | 1.12.15.18.20.21.25.28.29.30 | 7.6923 | 0.9168 | 9.6 | 326 | 313 | 17 | 41 | 0.8883 | 0.9485 | 0.9504 |
| | | | | | 1.7.12.13.17.18.19.21.22.24 | 11.0368 | 0.9168 | 9.8 | 325 | 314 | 16 | 42 | 0.8856 | 0.9515 | 0.9531 |
| | | | | | 1.3.4.7.12.19.20.22.24.28 | 9.6990 | 0.9168 | 9.8 | 324 | 315 | 15 | 43 | 0.8828 | 0.9545 | 0.9558 |

*Tabla 5.5 – Análisis de los mejores individuos para k=10 en la BD Financiera.*

En la tabla 5.5, se presentan 20 mejores individuos junto con sus características. El rango de su tasa de precisión en el conjunto de entrenamiento oscila entre 91.68% y 92.97%; su calificación de correlación es de 9.6 y 9.8 de un total de 10 y el porcentaje de error en el conjunto de prueba se encuentra entre el 6.3545% y 11.0368%.

b) Para un individuo con 8 genes (atributos) con 16 ejecuciones:

| ID | # Generac. | % Cruza | % Mutac. | # mejores indivi's. | Mejor Indiv. del AG en cto. de entrenam. | % de Error | Tasa de Precisión | CCC | VP | VN | FP | FN | TVP | TVN | Precis. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 500 | 90% | 5% | 1 | 1.3.7.12.21.24.29.30 | 10.0334 | 0.9139 | 7.8 | 319 | 318 | 12 | 48 | 0.8692 | 0.9636 | 0.9637 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1000 | 90% | 5% | 1 | 4.7.12.15.22.25.28.29 | 9.3645 | 0.9240 | 7.8 | 328 | 316 | 14 | 39 | 0.8937 | 0.9576 | 0.9591 |
| 3 | 500 | 90% | 5% | 1 | 4.7.12.15.17.25.28.29 | 9.6990 | 0.9254 | 7.8 | 327 | 318 | 12 | 40 | 0.8910 | 0.9636 | 0.9646 |
| **4** | **1000** | **90%** | **5%** | **1** | **4.7.12.18.20.24.28.32** | **7.6923** | **0.9211** | **7.8** | **328** | **314** | **16** | **39** | **0.8937** | **0.9515** | **0.9535** |
| 5 | 500 | 95% | 5% | 1 | 1.4.7.12.19.21.24.32 | 11.0368 | 0.9182 | 7.8 | 322 | 318 | 12 | 45 | 0.8774 | 0.9636 | 0.9641 |
| 6 | 1000 | 95% | 5% | 1 | 1.4.7.12.18.21.22.25 | 13.0435 | 0.9254 | 7.8 | 326 | 319 | 11 | 41 | 0.8883 | 0.9667 | 0.9674 |
| 7 | 500 | 95% | 5% | 1 | 1.4.7.12.19.20.28.32 | 9.0301 | 0.9211 | 7.8 | 326 | 316 | 14 | 41 | 0.8883 | 0.9576 | 0.9588 |
| **8** | **1000** | **95%** | **5%** | **1** | **4.7.12.13.20.21.22.25** | **7.0234** | **0.9311** | **7.8** | **331** | **318** | **12** | **36** | **0.9019** | **0.9636** | **0.9650** |
| 9 | 500 | 90% | 3% | 1 | 7.12.15.17.24.25.28.30 | 8.6957 | 0.9211 | 7.8 | 324 | 318 | 12 | 43 | 0.8828 | 0.9636 | 0.9643 |
| 10 | 1000 | 90% | 3% | 1 | 3.7.12.13.22.24.28.30 | 8.0268 | 0.9168 | 7.8 | 326 | 313 | 17 | 41 | 0.8883 | 0.9485 | 0.9504 |
| 11 | 500 | 90% | 3% | 1 | 7.12.15.20.22.24.28.32 | 8.3612 | 0.9154 | 7.8 | 323 | 315 | 15 | 44 | 0.8801 | 0.9545 | 0.9556 |
| 12 | 1000 | 90% | 3% | 1 | 1.7.12.17.18.20.28.32 | 9.3645 | 0.9154 | 7.8 | 323 | 315 | 15 | 44 | 0.8801 | 0.9545 | 0.9556 |
| **13** | **500** | **95%** | **3%** | **1** | **7.12.13.20.22.28.29.32** | **5.6856** | **0.9182** | **7.8** | **325** | **315** | **15** | **42** | **0.8856** | **0.9545** | **0.9559** |
| 14 | 1000 | 95% | 3% | 1 | 1.4.12.15.20.21.24.30 | 8.6957 | 0.9197 | 7.8 | 323 | 318 | 12 | 44 | 0.8801 | 0.9636 | 0.9642 |
| 15 | 500 | 95% | 3% | 1 | 3.4.7.12.19.21.22.24 | 11.3712 | 0.9211 | 7.8 | 326 | 316 | 14 | 41 | 0.8883 | 0.9576 | 0.9588 |
| 16 | 1000 | 95% | 3% | 1 | 4.7.12.19.22.24.28.29 | 10.3679 | 0.9225 | 7.8 | 326 | 317 | 13 | 41 | 0.8883 | 0.9606 | 0.9617 |

*Tabla 5.6 – Análisis de los mejores individuos para k=8 en la BD Financiera.*

En la tabla 5.6, se observan 16 mejores individuos, uno diferente por cada ejecución. Nuevamente, para el conjunto de entrenamiento, la tasa de precisión se encuentra entre el 91.39% y el 93.11%; además, todos los individuos tienen una calificación de correlación del 7.8, de un máximo de 8, y el error de clasificación para el conjunto de prueba está en el intervalo del 5.69% al 13.04%, siendo un intervalo más amplio que el anterior (para k=10).

c) Para un individuo con 6 genes (atributos) con 16 ejecuciones:

| ID | # Gener. | % Cruza | % Mutac. | # mejores indivi's. | Mejor Indiv. del AG en cto. de entrenam. | % de Error | Tasa de Precisión | CCC | VP | VN | FP | FN | TVP | TVN | Precisión |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | **500** | **90%** | **5%** | **1** | **4.7.12.20.22.30** | **11.0368** | **0.9110** | **6.0** | **318** | **317** | **13** | **49** | **0.8665** | **0.9606** | **0.9607** |
| 2 | 1000 | 90% | 5% | 1 | 1.7.12.20.21.22 | 11.3712 | 0.9197 | 5.8 | 319 | 322 | 8 | 48 | 0.8692 | 0.9758 | 0.9755 |
| **3** | **500** | **90%** | **5%** | **2** | **4.7.12.20.29.30** | **11.0368** | **0.9125** | **6.0** | **322** | **314** | **16** | **45** | **0.8774** | **0.9515** | **0.9527** |
| | | | | | **4.7.12.20.22.29** | **9.6990** | **0.9125** | **6.0** | **319** | **317** | **13** | **48** | **0.8692** | **0.9606** | **0.9608** |
| 4 | 1000 | 90% | 5% | 2 | 4.7.12.20.29.30 | 11.0368 | 0.9125 | 6.0 | 322 | 314 | 16 | 45 | 0.8774 | 0.9515 | 0.9527 |
| | | | | | 1.7.12.15.19.20 | 15.7191 | 0.9125 | 6.0 | 315 | 321 | 9 | 52 | 0.8583 | 0.9727 | 0.9722 |
| 5 | 500 | 95% | 5% | 2 | 4.7.12.20.22.29 | 9.6990 | 0.9125 | 6.0 | 319 | 317 | 13 | 48 | 0.8692 | 0.9606 | 0.9608 |
| | | | | | 1.7.12.15.19.20 | 15.7191 | 0.9125 | 6.0 | 315 | 321 | 9 | 52 | 0.8583 | 0.9727 | 0.9722 |
| 6 | 1000 | 95% | 5% | 1 | 4.7.12.20.29.30 | 11.0368 | 0.9125 | 6.0 | 322 | 314 | 16 | 45 | 0.8774 | 0.9515 | 0.9527 |
| **7** | **500** | **95%** | **5%** | **2** | **4.7.12.20.22.29** | **9.6990** | **0.9125** | **6.0** | **319** | **317** | **13** | **48** | **0.8692** | **0.9606** | **0.9608** |
| | | | | | 4.7.12.15.20.30 | 11.3712 | 0.9125 | 6.0 | 317 | 319 | 11 | 50 | 0.8638 | 0.9667 | 0.9665 |
| 8 | 1000 | 95% | 5% | 3 | 4.7.12.20.29.30 | 11.0368 | 0.9125 | 6.0 | 322 | 314 | 16 | 45 | 0.8774 | 0.9515 | 0.9527 |
| | | | | | 1.7.12.15.19.20 | 15.7191 | 0.9125 | 6.0 | 315 | 321 | 9 | 52 | 0.8583 | 0.9727 | 0.9722 |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 4.7.12.15.20.30 | 11.3712 | 0.9125 | 6.0 | 317 | 319 | 11 | 50 | 0.8638 | 0.9667 | 0.9665 |
| **9** | **500** | **90%** | **3%** | **2** | **4.7.12.20.22.29** | **9.6990** | **0.9125** | **6.0** | **319** | **317** | **13** | **48** | **0.8692** | **0.9606** | **0.9608** |
| | | | | | 7.12.15.19.20.30 | 12.7090 | 0.9125 | 6.0 | 319 | 317 | 13 | 48 | 0.8692 | 0.9606 | 0.9608 |
| 10 | 1000 | 90% | 3% | 1 | 7.12.19.21.25.30 | 12.7090 | 0.9168 | 5.8 | 322 | 317 | 13 | 45 | 0.8774 | 0.9606 | 0.9612 |
| 11 | 500 | 90% | 3% | 2 | 7.12.15.19.20.30 | 12.7090 | 0.9125 | 6.0 | 319 | 317 | 13 | 48 | 0.8692 | 0.9606 | 0.9608 |
| | | | | | **4.7.12.20.22.29** | **9.6990** | **0.9125** | **6.0** | **319** | **317** | **13** | **48** | **0.8692** | **0.9606** | **0.9608** |
| 12 | 1000 | 90% | 3% | 1 | 4.7.12.20.29.30 | 11.0368 | 0.9125 | 6.0 | 322 | 314 | 16 | 45 | 0.8774 | 0.9515 | 0.9527 |
| **13** | **500** | **95%** | **3%** | **1** | **4.7.12.20.22.30** | **11.0368** | **0.9110** | **6.0** | **318** | **317** | **13** | **49** | **0.8665** | **0.9606** | **0.9607** |
| 14 | 1000 | 95% | 3% | 2 | 4.7.12.19.20.29 | 12.3746 | 0.9125 | 6.0 | 321 | 315 | 15 | 46 | 0.8747 | 0.9545 | 0.9554 |
| | | | | | 4.7.12.15.20.30 | 11.3712 | 0.9125 | 6.0 | 317 | 319 | 11 | 50 | 0.8638 | 0.9667 | 0.9665 |
| 15 | 500 | 95% | 3% | 3 | 4.7.12.19.20.29 | 12.3746 | 0.9125 | 6.0 | 321 | 315 | 15 | 46 | 0.8747 | 0.9545 | 0.9554 |
| | | | | | **4.7.12.20.22.29** | **9.6990** | **0.9125** | **6.0** | **319** | **317** | **13** | **48** | **0.8692** | **0.9606** | **0.9608** |
| | | | | | 1.7.12.15.19.20 | 15.7191 | 0.9125 | 6.0 | 315 | 321 | 9 | 52 | 0.8583 | 0.9727 | 0.9722 |
| 16 | 1000 | 95% | 3% | 5 | 7.12.15.19.20.30 | 12.7090 | 0.9125 | 6.0 | 319 | 317 | 13 | 48 | 0.8692 | 0.9606 | 0.9608 |
| | | | | | **4.7.12.20.29.30** | **11.0368** | **0.9125** | **6.0** | **322** | **314** | **16** | **45** | **0.8774** | **0.9515** | **0.9527** |
| | | | | | 1.7.12.15.19.20 | 15.7191 | 0.9125 | 6.0 | 315 | 321 | 9 | 52 | 0.8583 | 0.9727 | 0.9722 |
| | | | | | 4.7.12.15.20.30 | 11.3712 | 0.9125 | 6.0 | 317 | 319 | 11 | 50 | 0.8638 | 0.9667 | 0.9665 |
| | | | | | **4.7.12.20.22.29** | **9.6990** | **0.9125** | **6.0** | **319** | **317** | **13** | **48** | **0.8692** | **0.9606** | **0.9608** |

*Tabla 5.7 – Análisis de los mejores individuos para k=6 en la BD Financiera.*

En la tabla 5.7, podemos observar 31 mejores individuos, con la siguiente variabilidad: entre un 91.1% y 91.96% en la tasa de precisión y con calificaciones de su correlación de 5.8 y 6, de un máximo de 6, para las ejecuciones sobre el conjunto de entrenamiento; y un rango de 9.689% y 15.719%. del porcentaje de error sobre el conjunto de prueba. Cabe señalar, que tanto el rango de valores de la tasa de precisión como los porcentajes de error cambiaron, en comparación con las ejecuciones para k=10 y 8, el primero se redujo y los segundos aumentaron, debido a la pérdida de información al tomar sólo 6 atributos de los 32 existentes.

Para la base de datos médica 'WDBC', se presentan los siguientes resultados con k=10,8,6:

a) Para un individuo con 10 genes (atributos) con 16 ejecuciones:

| ID | # Gener. | % Cruza | % Mut. | # mejores indivi's. | Mejor Indiv. del AG en cto. de entrenam. | % de Error | Tasa de Precis. | CCC | VP | VN | FP | FN | TVP | TVN | Precisión |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | **500** | **90%** | **5%** | **1** | **3.6.8.11.13.14.23.24.26.29** | **13.5294** | **0.8557** | **8.6** | **93** | **245** | **3** | **54** | **0.6327** | **0.9879** | **0.9688** |
| 2 | 1000 | 90% | 5% | 2 | 3.7.8.11.13.14.21.22.24.27 | 14.7059 | 0.8430 | 8.8 | 92 | 241 | 7 | 55 | 0.6259 | 0.9718 | 0.9293 |
| | | | | | 6.8.11.13.14.21.22.23.24.28 | 15.2941 | 0.8430 | 9.0 | 91 | 242 | 6 | 56 | 0.6190 | 0.9758 | 0.9381 |
| 3 | 500 | 90% | 5% | 1 | 2.6.8.11.13.14.21.22.23.24 | 14.1176 | 0.8532 | 8.8 | 93 | 244 | 4 | 54 | 0.6327 | 0.9839 | 0.9588 |
| 4 | 1000 | 90% | 5% | 1 | 5.6.7.13.14.21.23.24.27.29 | 14.7059 | 0.8532 | 8.2 | 93 | 244 | 4 | 54 | 0.6327 | 0.9839 | 0.9588 |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 500 | 95% | 5% | 1 | 1.4.6.8.11.13.14.21.24.26 | 17.0588 | 0.8456 | 8.8 | 92 | 242 | 6 | 55 | 0.6259 | 0.9758 | 0.9388 |
| 6 | 1000 | 95% | 5% | 1 | 1.2.3.6.8.13.14.22.23.26 | 18.8235 | 0.8380 | 8.8 | 88 | 243 | 5 | 59 | 0.5986 | 0.9798 | 0.9462 |
| 7 | 500 | 95% | 5% | 1 | 3.6.7.9.11.14.21.23.24.30 | 16.4706 | 0.8456 | 8.4 | 96 | 238 | 10 | 51 | 0.6531 | 0.9597 | 0.9057 |
| 8 | 1000 | 95% | 5% | 1 | 3.6.8.13.14.18.22.23.24.30 | 15.8824 | 0.8557 | 8.4 | 95 | 243 | 5 | 52 | 0.6463 | 0.9798 | 0.9500 |
| 9 | 500 | 90% | 3% | 1 | 3.6.7.13.14.21.22.24.25.26 | 16.4706 | 0.8532 | 8.4 | 93 | 244 | 4 | 54 | 0.6327 | 0.9839 | 0.9588 |
| 10 | 1000 | 90% | 3% | 1 | 6.8.11.13.14.21.22.23.24.28 | 15.2941 | 0.8430 | 9.0 | 91 | 242 | 6 | 56 | 0.6190 | 0.9758 | 0.9381 |
| 11 | 500 | 90% | 3% | 2 | 1.7.11.13.22.23.24.26.27.28 | 18.2353 | 0.8380 | 8.8 | 90 | 241 | 7 | 57 | 0.6122 | 0.9718 | 0.9278 |
| | | | | | 1.4.6.8.13.14.18.21.24.26 | 20 | 0.8380 | 8.6 | 92 | 239 | 9 | 55 | 0.6259 | 0.9637 | 0.9109 |
| 12 | 1000 | 90% | 3% | 2 | 2.3.6.7.11.13.14.21.24.28 | 14.7059 | 0.8506 | 8.8 | 93 | 243 | 5 | 54 | 0.6327 | 0.9798 | 0.9490 |
| | | | | | 3.7.11.13.14.21.22.24.26.28 | 14.7059 | 0.8506 | 8.8 | 93 | 243 | 5 | 54 | 0.6327 | 0.9798 | 0.9490 |
| 13 | 500 | 95% | 3% | 2 | 1.3.8.9.11.13.14.21.24.29 | 14.1176 | 0.8481 | 8.6 | 92 | 243 | 5 | 55 | 0.6259 | 0.9798 | 0.9485 |
| | | | | | **2.3.11.13.14.22.23.24.26.28** | **12.9412** | **0.8481** | **8.8** | **91** | **244** | **4** | **56** | **0.6190** | **0.9839** | **0.9579** |
| 14 | 1000 | 95% | 3% | 2 | 2.6.11.13.14.21.23.24.27.28 | 14.1176 | 0.8506 | 8.8 | 94 | 242 | 6 | 53 | 0.6395 | 0.9758 | 0.9400 |
| | | | | | 5.11.13.14.21.22.23.24.27.28 | 14.1176 | 0.8506 | 8.6 | 94 | 242 | 6 | 53 | 0.6395 | 0.9758 | 0.9400 |
| 15 | 500 | 95% | 3% | 1 | 3.6.7.11.13.14.21.24.26.28 | 16.4706 | 0.8532 | 8.8 | 94 | 243 | 5 | 53 | 0.6395 | 0.9798 | 0.9495 |
| **16** | **1000** | **95%** | **3%** | **1** | **1.3.6.8.11.13.14.24.26.30** | **12.3529** | **0.8582** | **8.6** | **95** | **244** | **4** | **52** | **0.6463** | **0.9839** | **0.9596** |

*Tabla 5.8 – Análisis de los mejores individuos para k=10 en la BD del WDBC.*

La tabla 5.8 nos muestra un total de 21 mejores individuos, con calificaciones de su correlación con la clase de 8.2, 8.4, 8.6, 8.8 y 9, de un total de 10. Su tasa de precisión se encuentra entre el 83.79% y el 85.82%, y un porcentaje de error en el conjunto de prueba que varía entre el 12.3529% y el 20%.

b) Para un individuo con 8 genes (atributos) con 16 ejecuciones:

| ID | # Gener. | % Cruza | % Mut. | # MI | Mejor Indiv. del AG en cto. de entrenam. | % de Error | Tasa de Precis. | CCC | VP | VN | FP | FN | TVP | TVN | Precis. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 500 | 90% | 5% | 1 | 2.8.11.14.21.22.24.28 | 17.0588 | 0.8253 | 7.2 | 85 | 241 | 7 | 62 | 0.5782 | 0.9718 | 0.9239 |
| **2** | **1000** | **90%** | **5%** | **1** | **1.2.6.8.11.13.24.27** | **14.1176** | **0.8354** | **7.0** | **88** | **242** | **6** | **59** | **0.5986** | **0.9758** | **0.9362** |
| 3 | 500 | 90% | 5% | 1 | 1.9.11.13.14.21.24.28 | 15.8824 | 0.8456 | 7.0 | 90 | 244 | 4 | 57 | 0.6122 | 0.9839 | 0.9574 |
| **4** | **1000** | **90%** | **5%** | **1** | **2.8.13.14.18.21.22.24** | **12.9412** | **0.8506** | **6.8** | **93** | **243** | **5** | **54** | **0.6327** | **0.9798** | **0.9490** |
| 5 | 500 | 95% | 5% | 1 | 3.8.11.13.14.23.24.30 | 15.2941 | 0.8506 | 7.0 | 89 | 247 | 1 | 58 | 0.6054 | 0.9960 | 0.9889 |
| 6 | 1000 | 95% | 5% | 1 | 1.8.11.13.14.24.26.28 | 14.7059 | 0.8456 | 7.2 | 92 | 242 | 6 | 55 | 0.6259 | 0.9758 | 0.9388 |
| **7** | **500** | **95%** | **5%** | **1** | **2.7.11.13.14.21.24.28** | **14.1176** | **0.8506** | **7.0** | **91** | **245** | **3** | **56** | **0.6190** | **0.9879** | **0.9681** |
| 8 | 1000 | 95% | 5% | 1 | 3.8.11.13.14.21.28.30 | 15.2941 | 0.8532 | 7.0 | 91 | 246 | 2 | 56 | 0.6190 | 0.9919 | 0.9785 |
| 9 | 500 | 90% | 3% | 2 | 1.6.8.11.13.14.21.28 | 18.8235 | 0.8405 | 7.2 | 90 | 242 | 6 | 57 | 0.6122 | 0.9758 | 0.9375 |
| | | | | | 1.11.13.22.23.24.26.27 | 14.7059 | 0.8405 | 7.0 | 90 | 242 | 6 | 57 | 0.6122 | 0.9758 | 0.9375 |
| 10 | 1000 | 90% | 3% | 1 | 1.2.8.13.14.22.24.28 | 16.4706 | 0.8278 | 7.2 | 84 | 243 | 5 | 63 | 0.5714 | 0.9798 | 0.9438 |
| 11 | 500 | 90% | 3% | 1 | 2.3.6.11.14.24.27.28 | 15.8824 | 0.8405 | 7.0 | 89 | 243 | 5 | 58 | 0.6054 | 0.9798 | 0.9468 |
| 12 | 1000 | 90% | 3% | 1 | 1.4.8.11.13.14.24.25 | 16.4706 | 0.8430 | 6.8 | 91 | 242 | 6 | 56 | 0.6190 | 0.9758 | 0.9381 |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 500 | 95% | 3% | 1 | 1.2.6.8.11.13.14.21 | 14.7059 | 0.8506 | 7.0 | 91 | 245 | 3 | 56 | 0.6190 | 0.9879 | 0.9681 |
| 14 | 1000 | 95% | 3% | 2 | 2.3.11.13.21.24.26.28 | 18.2353 | 0.8253 | 7.2 | 87 | 239 | 9 | 60 | 0.5918 | 0.9637 | 0.9063 |
| | | | | | 1.5.8.11.14.23.24.28 | 20 | 0.8253 | 7.2 | 88 | 238 | 10 | 59 | 0.5986 | 0.9597 | 0.8980 |
| 15 | 500 | 95% | 3% | 1 | 3.7.8.11.13.14.24.28 | 16.4706 | 0.8380 | 7.2 | 90 | 241 | 7 | 57 | 0.6122 | 0.9718 | 0.9278 |
| 16 | 1000 | 95% | 3% | 2 | 3.8.11.14.16.22.23.24 | 15.8824 | 0.8354 | 6.8 | 89 | 241 | 7 | 58 | 0.6054 | 0.9718 | 0.9271 |
| | | | | | 1.8.13.14.21.22.24.26 | 16.4706 | 0.8354 | 7.2 | 88 | 242 | 6 | 59 | 0.5986 | 0.9758 | 0.9362 |

*Tabla 5.9 – Análisis de los mejores individuos para k=8 en la BD del WDBC.*

En la tabla anterior, podemos ver los 19 mejores individuos obtenidos de la serie de 16 ejecuciones, Las calificaciones de la correlación de los atributos con la clase varían del 6.8 al 7.2, de un total de 8. La tasa de precisión sobre el conjunto de entrenamiento oscila entre el 82.53% y el 85.32%. Por su parte, el porcentaje de error sobre el conjunto de prueba se encuentra entre el 12.9412% y el 20%.

c) Para un individuo con 6 genes (atributos) con 16 ejecuciones:

| **ID** | **# Generac.** | **% Cruza** | **% Mutac.** | **# MI** | **Mejor Indiv. del AG en cto. de entrenam.** | **% de Error** | **Tasa de Precisión** | **CCC** | **VP** | **VN** | **FP** | **FN** | **TVP** | **TVN** | **Precis.** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 500 | 90% | 5% | 1 | 7.8.11.14.21.23 | 18.8235 | 0.8203 | 5.4 | 86 | 238 | 10 | 61 | 0.5850 | 0.9597 | 0.8958 |
| 2 | 1000 | 90% | 5% | 1 | 3.6.11.14.23.24 | 18.2353 | 0.8278 | 5.4 | 85 | 242 | 6 | 62 | 0.5782 | 0.9758 | 0.9341 |
| **3** | **500** | **90%** | **5%** | **1** | **1.2.8.11.14.24** | **15.2941** | **0.8380** | **5.4** | **88** | **243** | **5** | **59** | **0.5986** | **0.9798** | **0.9462** |
| 4 | 1000 | 90% | 5% | 1 | 3.6.13.14.23.28 | 20.5882 | 0.8380 | 5.4 | 87 | 244 | 4 | 60 | 0.5918 | 0.9839 | 0.9560 |
| 5 | 500 | 95% | 5% | 1 | 3.8.13.14.22.24 | 15.8824 | 0.8456 | 5.4 | 88 | 246 | 2 | 59 | 0.5986 | 0.9919 | 0.9778 |
| 6 | 1000 | 95% | 5% | 1 | 1.2.8.13.14.24 | 15.8824 | 0.8405 | 5.4 | 88 | 244 | 4 | 59 | 0.5986 | 0.9839 | 0.9565 |
| 7 | 500 | 95% | 5% | 1 | 1.6.8.11.14.23 | 19.4118 | 0.8380 | 5.4 | 89 | 242 | 6 | 58 | 0.6054 | 0.9758 | 0.9368 |
| 8 | 1000 | 95% | 5% | 1 | 8.11.13.21.24.26 | 16.4706 | 0.8354 | 5.4 | 88 | 242 | 6 | 59 | 0.5986 | 0.9758 | 0.9362 |
| 9 | 500 | 90% | 3% | 1 | 1.3.8.13.14.24 | 18.8235 | 0.8152 | 5.6 | 84 | 238 | 10 | 63 | 0.5714 | 0.9597 | 0.8936 |
| **10** | **1000** | **90%** | **3%** | **1** | **8.11.13.14.21.23** | **15.2941** | **0.8532** | **5.4** | **90** | **247** | **1** | **57** | **0.6122** | **0.9960** | **0.9890** |
| 11 | 500 | 90% | 3% | 1 | 1.8.11.14.24.28 | 17.6471 | 0.8203 | 5.6 | 86 | 238 | 10 | 61 | 0.5850 | 0.9597 | 0.8958 |
| 12 | 1000 | 90% | 3% | 1 | 3.8.11.13.14.21 | 16.4706 | 0.8557 | 5.4 | 91 | 247 | 1 | 56 | 0.6190 | 0.9960 | 0.9891 |
| 13 | 500 | 95% | 3% | 2 | 3.7.8.11.22.24 | 18.8235 | 0.8152 | 5.4 | 83 | 239 | 9 | 64 | 0.5646 | 0.9637 | 0.9022 |
| | | | | | 1.8.11.14.23.24 | 18.2353 | 0.8152 | 5.6 | 84 | 238 | 10 | 63 | 0.5714 | 0.9597 | 0.8936 |
| 14 | 1000 | 95% | 3% | 1 | 1.7.8.11.14.24 | 17.0588 | 0.8380 | 5.4 | 90 | 241 | 7 | 57 | 0.6122 | 0.9718 | 0.9278 |
| 15 | 500 | 95% | 3% | 1 | 11.14.21.23.24.28 | 19.4118 | 0.8127 | 5.6 | 85 | 236 | 12 | 62 | 0.5782 | 0.9516 | 0.8763 |
| **16** | **1000** | **95%** | **3%** | **1** | **8.11.13.14.24.28** | **14.1176** | **0.8582** | **5.4** | **93** | **246** | **2** | **54** | **0.6327** | **0.9919** | **0.9789** |

*Tabla 5.10 – Análisis de los mejores individuos para k=6 en la BD del WDBC.*

La tabla 5.10, muestra un total de 17 mejores individuos. Las calificaciones de la correlación obtenidas por los individuos son 5.4 y 5.6, de un total de 6. La tasa de precisión

varía entre el 81.26% y 85.82%, y el porcentaje de error en la clasificación sobre el conjunto de entrenamiento se encuentra en el rango del 14.1176% al 20.5882%. Durante el conjunto de 48 ejecuciones, la tasa de precisión se mantuvo entre el 81% y el 85 %, y el porcentaje de error, entre el 12% y el 20%, lo cual demuestra que el sistema obtuvo resultados aceptables, en comparación con la base de datos anterior, tomando en cuenta la pérdida de información al disminuir el total de atributos de 8 a 6.

Para la BD médica 'WPBC', se presentan los siguientes resultados para k=10, 8 y 6:

a) Para un individuo con 10 genes (atributos) con 16 ejecuciones:

| ID | # Gener. | % Cruza | % Mutac. | # MI | Mejor Indiv. del AG en cto. de entrenam. | % de Error | Tasa de Prec. | CCC | VP | VN | FP | FN | TVP | TVN | Precis. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 500 | 90% | 5% | 1 | 1.2.3.13.14.16.18.24.26.33 | 11.8644 | 0.9111 | 8.0 | 28 | 95 | 9 | 3 | 0.9032 | 0.9135 | 0.7568 |
| **2** | **1000** | **90%** | **5%** | **2** | **1.2.3.11.12.16.17.18.26.27** | **8.4746** | **0.8889** | **7.8** | **25** | **95** | **9** | **6** | **0.8065** | **0.9135** | **0.7353** |
| | | | | | 1.2.3.4.7.11.16.18.27.33 | 11.8644 | 0.8889 | 7.6 | 25 | 95 | 9 | 6 | 0.8065 | 0.9135 | 0.7353 |
| 3 | 500 | 90% | 5% | 1 | 1.2.3.12.13.16.24.27.32.33 | 10.1695 | 0.9037 | 7.8 | 27 | 95 | 9 | 4 | 0.8710 | 0.9135 | 0.7500 |
| 4 | 1000 | 90% | 5% | 1 | 1.2.3.13.14.16.25.26.27.29 | 10.1695 | 0.9111 | 7.6 | 26 | 97 | 7 | 5 | 0.8387 | 0.9327 | 0.7879 |
| 5 | 500 | 95% | 5% | 1 | 2.7.11.13.16.18.21.24.27.32 | 18.6441 | 0.9111 | 7.4 | 30 | 93 | 11 | 1 | 0.9677 | 0.8942 | 0.7317 |
| 6 | 1000 | 95% | 5% | 3 | 1.2.3.13.14.16.20.26.27.32 | 10.1695 | 0.8889 | 7.8 | 25 | 95 | 9 | 6 | 0.8065 | 0.9135 | 0.7353 |
| | | | | | 1.2.3.13.14.17.20.24.27.32 | 11.8644 | 0.8889 | 7.8 | 25 | 95 | 9 | 6 | 0.8065 | 0.9135 | 0.7353 |
| | | | | | 1.2.7.11.13.14.17.18.26.32 | 11.8644 | 0.8889 | 7.8 | 26 | 94 | 10 | 5 | 0.8387 | 0.9038 | 0.7222 |
| 7 | 500 | 95% | 5% | 1 | 1.2.7.11.17.18.21.26.27.33 | 11.8644 | 0.9037 | 7.6 | 25 | 97 | 7 | 6 | 0.8065 | 0.9327 | 0.7813 |
| 8 | 1000 | 95% | 5% | 1 | 1.2.3.18.20.24.26.27.32.33 | 11.8644 | 0.9111 | 7.6 | 26 | 97 | 7 | 5 | 0.8387 | 0.9327 | 0.7879 |
| 9 | 500 | 90% | 3% | 1 | 1.2.11.13.14.18.19.24.26.33 | 15.2542 | 0.8889 | 7.6 | 29 | 91 | 13 | 2 | 0.9355 | 0.8750 | 0.6905 |
| 10 | 1000 | 90% | 3% | 1 | 1.2.3.6.9.13.16.18.24.27 | 13.5593 | 0.8963 | 7.8 | 26 | 95 | 9 | 5 | 0.8387 | 0.9135 | 0.7429 |
| 11 | 500 | 90% | 3% | 1 | 1.2.3.6.7.11.16.18.22.26 | 13.5593 | 0.9111 | 7.4 | 26 | 97 | 7 | 5 | 0.8387 | 0.9327 | 0.7879 |
| **12** | **1000** | **90%** | **3%** | **1** | **1.2.3.12.13.14.17.18.22.26** | **6.7797** | **0.9259** | **7.4** | **26** | **99** | **5** | **5** | **0.8387** | **0.9519** | **0.8387** |
| 13 | 500 | 95% | 3% | 1 | 1.2.3.7.13.16.17.22.24.27 | 15.2542 | 0.8889 | 8.0 | 25 | 95 | 9 | 6 | 0.8065 | 0.9135 | 0.7353 |
| 14 | 1000 | 95% | 3% | 3 | 1.2.3.10.13.16.17.26.27.29 | 13.5593 | 0.8741 | 7.6 | 25 | 93 | 11 | 6 | 0.8065 | 0.8942 | 0.6944 |
| | | | | | 3.13.14.16.18.20.22.24.26.27 | 11.8644 | 0.8741 | 7.4 | 24 | 94 | 10 | 7 | 0.7742 | 0.9038 | 0.7059 |
| | | | | | 1.2.3.6.14.17.18.20.24.26 | 10.1695 | 0.8741 | 7.8 | 24 | 94 | 10 | 7 | 0.7742 | 0.9038 | 0.7059 |
| **15** | **500** | **95%** | **3%** | **2** | **1.2.3.4.13.14.21.26.27.32** | **8.4746** | **0.8963** | **7.8** | **25** | **96** | **8** | **6** | **0.8065** | **0.9231** | **0.7576** |
| | | | | | 1.2.3.4.12.13.16.19.24.27 | 11.8644 | 0.8963 | 7.8 | 26 | 95 | 9 | 5 | 0.8387 | 0.9135 | 0.7429 |
| 16 | 1000 | 95% | 3% | 2 | 1.2.6.8.13.14.18.24.32.33 | 18.6441 | 0.9111 | 7.4 | 29 | 94 | 10 | 2 | 0.9355 | 0.9038 | 0.7436 |
| | | | | | **1.2.3.12.13.16.17.18.24.27** | **8.4746** | **0.9111** | **8.0** | **27** | **96** | **8** | **4** | **0.8710** | **0.9231** | **0.7714** |

*Tabla 5.11 – Análisis de los mejores individuos para k=10 en la BD del WPBC.*

La tabla 5.11, muestra 21 mejores individuos obtenidos durante las 16 ejecuciones para k=10, con calificaciones de las correlaciones de 7.4 al 8, de un total de 10. La tasa de precisión sobre el conjunto de entrenamiento, varía entre el 87.4% y el 92.59%; y el porcentaje de error se encuentra entre el rango del 6.779% al 18.644%.

b) Para un individuo con 8 genes (atributos) con 16 ejecuciones:

| ID | # Gener. | % Cruza | % Mutac. | # MI | Mejor Indiv. del AG en cto. de entrenam. | % de Error | Tasa de Precis. | CCC | VP | VN | FP | FN | TVP | TVN | Precisión |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 500 | 90% | 5% | 3 | 1.2.3.6.7.16.17.18 | 13.5593 | 0.8889 | 6.2 | 26 | 94 | 10 | 5 | 0.8387 | 0.9038 | 0.7222 |
| | | | | | **2.3.13.16.17.18.24.27** | **8.4746** | **0.8889** | **6.4** | **26** | **94** | **10** | **5** | **0.8387** | **0.9038** | **0.7222** |
| | | | | | 1.2.3.4.14.17.18.26 | 10.1695 | 0.8889 | 6.4 | 25 | 95 | 9 | 6 | 0.8065 | 0.9135 | 0.7353 |
| 2 | 1000 | 90% | 5% | 1 | 1.2.4.16.17.27.32.33 | 11.8644 | 0.8815 | 6.4 | 25 | 94 | 10 | 6 | 0.8065 | 0.9038 | 0.7143 |
| 3 | 500 | 90% | 5% | 1 | 1.2.7.12.14.17.18.27 | 10.1695 | 0.8963 | 6.4 | 25 | 96 | 8 | 6 | 0.8065 | 0.9231 | 0.7576 |
| 4 | 1000 | 90% | 5% | 1 | 1.2.3.6.13.17.21.27 | 10.1695 | 0.9111 | 6.2 | 25 | 98 | 6 | 6 | 0.8065 | 0.9423 | 0.8065 |
| 5 | 500 | 95% | 5% | 1 | 1.2.3.6.17.18.24.26 | 11.8644 | 0.8815 | 6.6 | 24 | 95 | 9 | 7 | 0.7742 | 0.9135 | 0.7273 |
| 6 | 1000 | 95% | 5% | 1 | 1.2.3.6.13.14.24.32 | 11.8644 | 0.9037 | 6.4 | 26 | 96 | 8 | 5 | 0.8387 | 0.9231 | 0.7647 |
| 7 | 500 | 95% | 5% | 2 | 1.2.7.13.16.18.27.33 | 15.2542 | 0.9037 | 6.4 | 26 | 96 | 8 | 5 | 0.8387 | 0.9231 | 0.7647 |
| | | | | | 1.2.4.14.17.18.24.33 | 11.8644 | 0.9037 | 6.4 | 26 | 96 | 8 | 5 | 0.8387 | 0.9231 | 0.7647 |
| 8 | 1000 | 95% | 5% | 1 | 1.2.8.14.17.18.24.27 | 11.8644 | 0.8963 | 6.4 | 24 | 97 | 7 | 7 | 0.7742 | 0.9327 | 0.7742 |
| 9 | 500 | 90% | 3% | 3 | 1.2.6.17.18.24.26.32 | 16.9492 | 0.8593 | 6.6 | 24 | 92 | 12 | 7 | 0.7742 | 0.8846 | 0.6667 |
| | | | | | 1.2.4.7.13.14.18.24 | 23.7288 | 0.8593 | 6.6 | 25 | 91 | 13 | 6 | 0.8065 | 0.8750 | 0.6579 |
| | | | | | 1.2.4.10.14.18.24.26 | 20.3390 | 0.8593 | 6.4 | 24 | 92 | 12 | 7 | 0.7742 | 0.8846 | 0.6667 |
| 10 | 1000 | 90% | 3% | 1 | 1.2.3.13.17.18.26.27 | 10.1695 | 0.9037 | 6.6 | 24 | 98 | 6 | 7 | 0.7742 | 0.9423 | 0.8000 |
| **11** | **500** | **90%** | **3%** | **1** | **1.2.3.13.16.17.18.26** | **6.7797** | **0.9111** | **6.4** | **26** | **97** | **7** | **5** | **0.8387** | **0.9327** | **0.7879** |
| **12** | **1000** | **90%** | **3%** | **1** | **1.2.3.13.16.17.18.24** | **6.7797** | **0.9185** | **6.4** | **27** | **97** | **7** | **4** | **0.8710** | **0.9327** | **0.7941** |
| 13 | 500 | 95% | 3% | 1 | 1.2.4.13.14.16.18.26 | 10.1695 | 0.8815 | 6.6 | 26 | 93 | 11 | 5 | 0.8387 | 0.8942 | 0.7027 |
| 14 | 1000 | 95% | 3% | 1 | 1.2.7.13.16.17.18.27 | 11.8644 | 0.8963 | 6.6 | 25 | 96 | 8 | 6 | 0.8065 | 0.9231 | 0.7576 |
| 15 | 500 | 95% | 3% | 2 | 1.2.7.11.13.14.18.24 | 11.8644 | 0.9111 | 6.4 | 27 | 96 | 8 | 4 | 0.8710 | 0.9231 | 0.7714 |
| | | | | | **2.3.11.13.16.24.27.32** | **8.4746** | **0.9111** | **6.2** | **27** | **96** | **8** | **4** | **0.8710** | **0.9231** | **0.7714** |
| 16 | 1000 | 95% | 3% | 1 | 1.2.3.4.13.18.26.27 | 15.2542 | 0.8815 | 6.6 | 23 | 96 | 8 | 8 | 0.7419 | 0.9231 | 0.7419 |

*Tabla 5.12 – Análisis de los mejores individuos para k=8 en la BD del WPBC.*

Esta tabla muestra 22 mejores individuos, en donde la tasa de precisión oscila entre el 85.92% y el 91.85%; las calificaciones de la correlación de los atributos con la clase son 6.2, 6.4 y 6.6, de un total de 8. Por último, el porcentaje de error en la clasificación del conjunto de prueba se encuentra entre el 6.779% y el 23.7288, ampliándose así el intervalo

de error en comparación a la serie de ejecuciones anteriores, aunque el mínimo error alcanzado para k=8, es el mismo mínimo error obtenido para k=10.

c) Para un individuo con 6 genes (atributos) con 16 ejecuciones:

| ID | # Generac. | % Cruza | % Mutac. | # mejores indivi's. | Mejor Indiv. del AG en cto. de entrenam. | % de Error | Tasa de Precisión | CCC | VP | VN | FP | FN | TVP | TVN | Precisión |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 500 | 90% | 5% | 1 | 1.2.7.14.18.24 | 11.8644 | 0.8815 | 5.0 | 25 | 94 | 10 | 6 | 0.8065 | 0.9038 | 0.7143 |
| 2 | 1000 | 90% | 5% | 1 | 1.2.3.13.26.27 | 11.8644 | 0.8741 | 5.2 | 22 | 96 | 8 | 9 | 0.7097 | 0.9231 | 0.7333 |
| 3 | 500 | 90% | 5% | 1 | 1.2.7.17.18.27 | 11.8644 | 0.9037 | 5.0 | 25 | 97 | 7 | 6 | 0.8065 | 0.9327 | 0.7813 |
| 4 | 1000 | 90% | 5% | 1 | 1.2.3.21.24.26 | 13.5593 | 0.9037 | 4.8 | 22 | 100 | 4 | 9 | 0.7097 | 0.9615 | 0.8462 |
| 5 | 500 | 95% | 5% | 1 | 1.2.14.18.24.27 | 10.1695 | 0.8667 | 5.2 | 22 | 95 | 9 | 9 | 0.7097 | 0.9135 | 0.7097 |
| 6 | 1000 | 95% | 5% | 1 | 1.2.13.18.26.27 | 15.2542 | 0.8815 | 5.2 | 22 | 97 | 7 | 9 | 0.7097 | 0.9327 | 0.7586 |
| **7** | **500** | **95%** | **5%** | **1** | **1.2.13.17.27.32** | **8.4746** | **0.8963** | **5.0** | **24** | **97** | **7** | **7** | **0.7742** | **0.9327** | **0.7742** |
| 8 | 1000 | 95% | 5% | 1 | 1.2.4.14.18.26 | 11.8644 | 0.8815 | 5.0 | 24 | 95 | 9 | 7 | 0.7742 | 0.9135 | 0.7273 |
| 9 | 500 | 90% | 3% | 3 | 1.2.7.14.26.32 | 13.5593 | 0.8667 | 5.0 | 24 | 93 | 11 | 7 | 0.7742 | 0.8942 | 0.6857 |
| | | | | | 1.2.6.13.14.27 | 15.2542 | 0.8667 | 5.2 | 24 | 93 | 11 | 7 | 0.7742 | 0.8942 | 0.6857 |
| | | | | | 1.2.12.13.17.24 | 10.1695 | 0.8667 | 5.0 | 24 | 93 | 11 | 7 | 0.7742 | 0.8942 | 0.6857 |
| 10 | 1000 | 90% | 3% | 2 | 1.2.12.13.16.26 | 11.8644 | 0.8519 | 5.0 | 23 | 92 | 12 | 8 | 0.7419 | 0.8846 | 0.6571 |
| | | | | | 1.2.17.26.27.29 | 10.1695 | 0.8519 | 5.0 | 21 | 94 | 10 | 10 | 0.6774 | 0.9038 | 0.6774 |
| 11 | 500 | 90% | 3% | 1 | 1.7.13.14.18.27 | 18.6441 | 0.8889 | 5.0 | 25 | 95 | 9 | 6 | 0.8065 | 0.9135 | 0.7353 |
| 12 | 1000 | 90% | 3% | 1 | 1.2.3.7.14.26 | 11.8644 | 0.8815 | 5.0 | 23 | 96 | 8 | 8 | 0.7419 | 0.9231 | 0.7419 |
| **13** | **500** | **95%** | **3%** | **1** | **1.2.3.13.16.27** | **6.7797** | **0.9037** | **5.0** | **24** | **98** | **6** | **7** | **0.7742** | **0.9423** | **0.8000** |
| **14** | **1000** | **95%** | **3%** | **1** | **1.2.3.24.27.32** | **8.4746** | **0.8815** | **5.0** | **23** | **96** | **8** | **8** | **0.7419** | **0.9231** | **0.7419** |
| 15 | 500 | 95% | 3% | 1 | 1.2.7.16.18.24 | 11.8644 | 0.8815 | 5.0 | 25 | 94 | 10 | 6 | 0.8065 | 0.9038 | 0.7143 |
| 16 | 1000 | 95% | 3% | 1 | 1.2.6.13.18.27 | 16.9492 | 0.8963 | 5.0 | 24 | 97 | 7 | 7 | 0.7742 | 0.9327 | 0.7742 |

*Tabla 5.13 – Análisis de los mejores individuos para k=6 en la BD del WPBC.*

Finalmente, en la tabla 5.13, tenemos 19 mejores individuos, que obtuvieron las calificaciones de correlación del 4.8, 5 y 5.2, de un total de 6 puntos. El porcentaje de error sobre el conjunto de prueba se mantuvo estable en comparación a las pruebas anteriores, con valores que varían del 6.779% al 18.644%, exactamente el mismo intervalo, que el obtenido para la serie de ejecuciones donde k=10. Por su parte, la tasa de precisión de la clasificación sobre el conjunto de entrenamiento osciló entre el 85.19% y el 90.37%, disminuyendo en comparación con las pruebas anteriores.

## 5.3 ANÁLISIS DE LOS RESULTADOS.

A continuación, se analizarán los resultados de la evaluación para cada base de datos, con particular interés en el estudio de los mejores individuos, la redundancia de los atributos y la efectividad del algoritmo presentado.

### 5.3.1 Análisis de la Base de Datos de Portafolio de Inversión.

Como se puede observar en la tabla 5.5, del total de resultados con 10 atributos, existen 5 mejores individuos, en donde el porcentaje de error es mínimo, de 6.689% y 6.3545% y la tasa de precisión es máxima, 92.68%, 92.25%, 92.11% y 91.82%. Los mejores individuos obtenidos fueron los siguientes:

| Mejores Individuos del AG en conjunto de entrenamiento | % de Error | Tasa de Precisión | Calif. de Correl. |
|---|---|---|---|
| 7.12.14.15.19.20.24.25.28.29 | 6.6890 | 0.9182 | 9.6 |
| 1.7.12.18.19.20.21.29.30.32 | 6.6890 | 0.9211 | 9.8 |
| 1.4.7.12.18.20.24.25.28.29 | 6.6890 | 0.9225 | 9.8 |
| 1.4.7.12.20.22.24.28.29.30 | 6.3545 | 0.9225 | 9.8 |
| 1.4.7.12.13.15.20.21.24.30 | 6.3545 | 0.9268 | 9.8 |

*Tabla 5.14 – Mejores Individuos con k=10, para la BD Financiera.*

Para escogerlos, se dio preferencia a minimizar el porcentaje de error sobre el conjunto de prueba para después escoger el individuo con mayor tasa de precisión. El primer individuo de la tabla 5.14, se descarta debido a que tiene la menor calificación de correlación de los 5 individuos enlistados. Dado que no se tienen un mejor individuo con tasa de precisión de 1 y porcentaje de error de 0, es necesario analizar cada uno de los individuos y de ahí obtener el mejor. En la siguiente figura nos enfocamos en los 4 restantes:

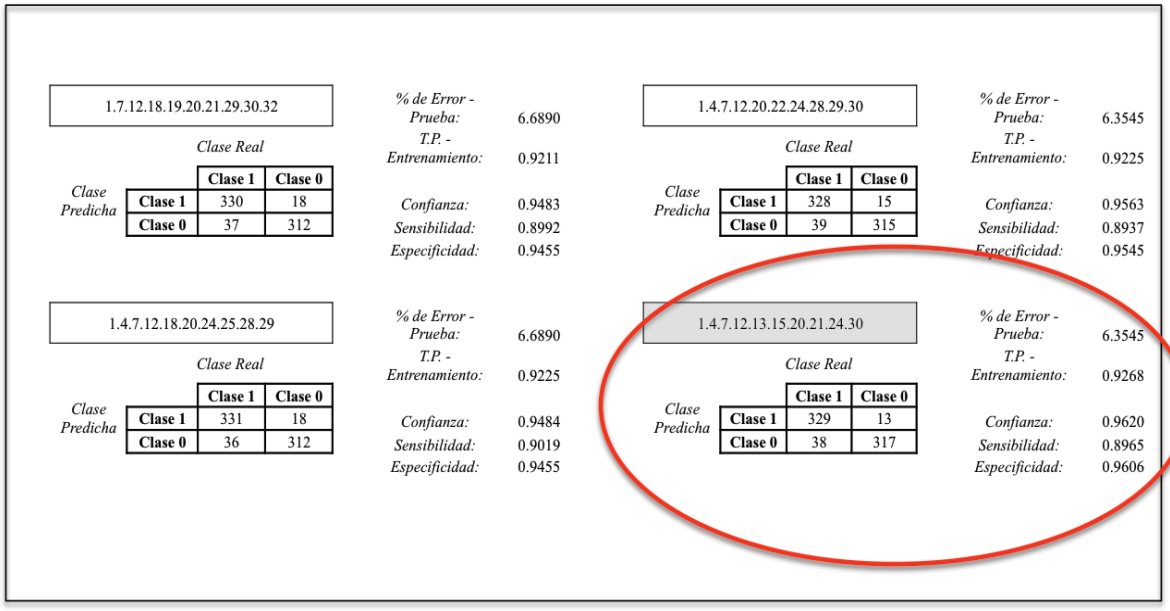



*Figura 5.1 – Análisis de 4 mejores individuos con k=10 para la BD Financiera.*

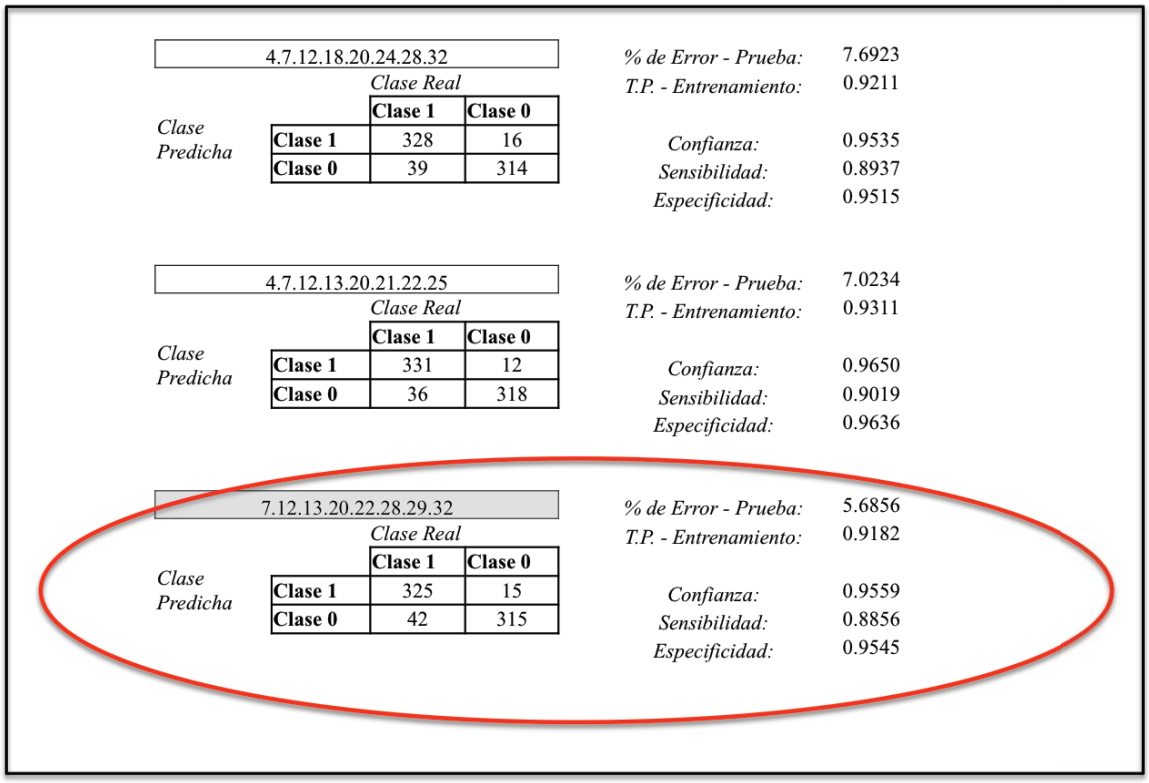
Como podemos ver en la figura 5.1, el individuo '1.4.7.12.13.15.20.21.24.30' es el mejor individuo para k=10, porque tiene el mínimo porcentaje de error y, al mismo tiempo, obtuvo la mayor tasa de precisión sin sobreajustar la clasificación al conjunto de entrenamiento. A su vez, tiene las mayores tasas de confianza y especificidad, y su sensibilidad dista en 45 centésimas del mejor.

Para el caso de k=8, es decir, individuos con 8 atributos, la tabla 5.15 muestra los tres mejores individuos para esta serie de ejecuciones:

| Mejores Individuos del AG en conjunto de entrenamiento | % de Error | Tasa de Precisión | Calif. de Correl. |
|---|---|---|---|
| 4.7.12.18.20.24.28.32 | 7.6923 | 0.9211 | 7.8 |
| 4.7.12.13.20.21.22.25 | 7.0234 | 0.9311 | 7.8 |
| 7.12.13.20.22.28.29.32 | 5.6856 | 0.9182 | 7.8 |

*Tabla 5.15 – Mejores Individuos con k=8, para la BD Financiera.*

En comparación a los mejores individuos obtenidos en la serie de ejecuciones para k=10, podemos observar que los atributos 1, 14, 15 y 30 desaparecen en la tabla 5.15, con una diferencia casi nula en el porcentaje de error y la tasa de precisión. A continuación, se analizarán más a fondo los tres individuos de la tabla 5.15:



4.7.12.18.20.24.28.32

| Clase Predicha \ Clase Real | Clase 1 | Clase 0 |
|---|---|---|
| **Clase 1** | 328 | 16 |
| **Clase 0** | 39 | 314 |

*% de Error - Prueba:* 7.6923
*T.P. - Entrenamiento:* 0.9211
*Confianza:* 0.9535
*Sensibilidad:* 0.8937
*Especificidad:* 0.9515

4.7.12.13.20.21.22.25

| Clase Predicha \ Clase Real | Clase 1 | Clase 0 |
|---|---|---|
| **Clase 1** | 331 | 12 |
| **Clase 0** | 36 | 318 |

*% de Error - Prueba:* 7.0234
*T.P. - Entrenamiento:* 0.9311
*Confianza:* 0.9650
*Sensibilidad:* 0.9019
*Especificidad:* 0.9636

7.12.13.20.22.28.29.32

| Clase Predicha \ Clase Real | Clase 1 | Clase 0 |
|---|---|---|
| **Clase 1** | 325 | 15 |
| **Clase 0** | 42 | 315 |

*% de Error - Prueba:* 5.6856
*T.P. - Entrenamiento:* 0.9182
*Confianza:* 0.9559
*Sensibilidad:* 0.8856
*Especificidad:* 0.9545

*Figura 5.2 – Análisis de 3 mejores individuos con k=8 para la BD Financiera.*

La figura 5.2 muestra que el mejor individuo para k=8 es '7.12.13.20.22.28.29.32' que, curiosamente, mantiene sólo 4 atributos, del mejor individuo para k=10. Éste individuo, contiene la más baja tasa de precisión de los 3 presentados; sin embargo, exhibe un menor porcentaje de error al clasificar nuevas instancias, razón por la cuál ha sido escogido.

La última prueba que se hizo con la base de datos de Portafolios de Inversión, fue para k=6, y en la tabla 5.7 se pueden observar 3 mejores individuos con una tasa de precisión del 91.1% y 91.25%, y un porcentaje de error de 9.699% y 11.0368%, como se resume en la tabla 5.16. La calificación de su correlación fue de 6, la máxima para k=6.

| Mejores Individuos del AG en conjunto de entrenamiento | % de Error | Tasa de Precisión | Calif. de Correl. |
|---|---|---|---|
| 4.7.12.20.22.30 | 11.0368 | 0.9110 | 6 |
| 4.7.12.20.29.30 | 11.0368 | 0.9125 | 6 |
| 4.7.12.20.22.29 | 9.6990 | 0.9125 | 6 |

*Tabla 5.16 – Mejores Individuos con k =6, para la BD Financiera.*

De igual forma, en la figura 5.3, se analizan los 3 mejores individuos para k=6:

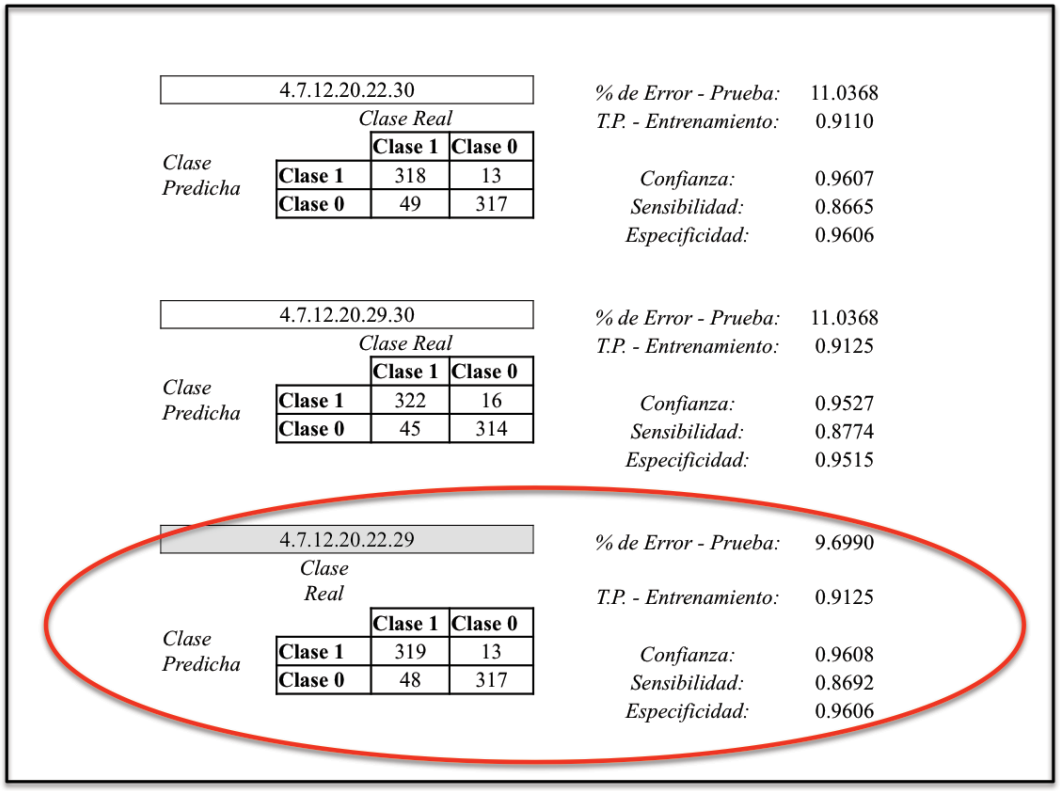



*Figura 5.3 – Análisis de 3 mejores individuos con k =6 para la BD Financiera.*

En el caso de la figura 5.3, es fácil observar que el mejor individuo es '4.7.12.20.22.29', ya que posee tanto la mejor tasa de precisión de la clasificación sobre el conjunto de entrenamiento, como el mínimo error al clasificar el conjunto desconocido de prueba. En cuanto a los atributos que conforman al mejor individuo para k=6, se observa que regresa el atributo 4 que también esta incluido en el mejor individuo para k=10, los atributos 22 y 29

que también pertenecen al mejor individuo para k=8, y los atributos 7, 12 y 20, que se mantienen constantes en los 3 mejores individuos finales.

A continuación se presenta el histograma de las tres series de ejecuciones (Figura 5.4), para k=10,8,6, que muestra los atributos predominantes así como su frecuencia. También, se presenta una tabla con el mejor individuo para cada serie de ejecuciones (Tabla 5.17).

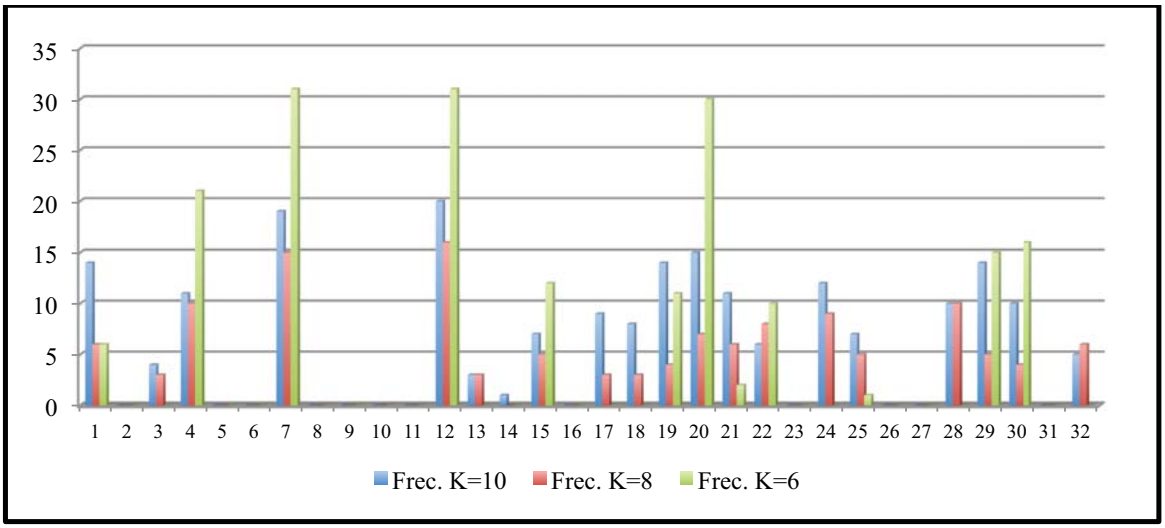



*Figura 5.4 – Frecuencias de los atributos de la BD Financiera, con k=10,8,6.*

| K | Mejor Individuo |
|---|---|
| **10** | 1.4.7.12.13.15.20.21.24.30 |
| **8** | 7.12.13.20.22.28.29.32 |
| **6** | 4.7.12.20.22.29 |

La figura 5.4 y la tabla 5.17, muestran que los atributos más representativos que cumplen con la clase objetivo son el 1, 4, 7, 12, 20, 29 y 30.

*Tabla 5.17 – Mejores Individuos para la BD Financiera.*

### 5.3.2 Análisis de la Base de Datos Médica 'WDBC'.

Al analizar la tabla 5.8, que presenta los mejores individuos de las 16 ejecuciones para k=10, se obtienen tres mejores individuos con el porcentaje de error más bajo sobre el conjunto de prueba, de 12.3529%, 12.9412% y 13.5294%, y una mayor tasa de precisión de 84.81%, 85.57% y 85.82%, los cuales se muestran a continuación:

| Mejores Individuos del AG en conjunto de entrenam. | % de Error | Tasa de Precisión | Calif. de Correl. |
|---|---|---|---|
| 3.6.8.11.13.14.23.24.26.29 | 13.5294 | 0.8557 | 8.6 |
| 2.3.11.13.14.22.23.24.26.28 | 12.9412 | 0.8481 | 8.8 |
| 1.3.6.8.11.13.14.24.26.30 | 12.3529 | 0.8582 | 8.6 |

*Tabla 5.18 – Mejores Individuos con k=10, para la BD WDBC.*

Podría decidirse eliminar los individuos con menor calificación de su correlación con la clase, sin embargo, es prudente analizar los tres con el fin de escoger al mejor, con base en todos sus parámetros. En la figura 5.5, se presenta su análisis:

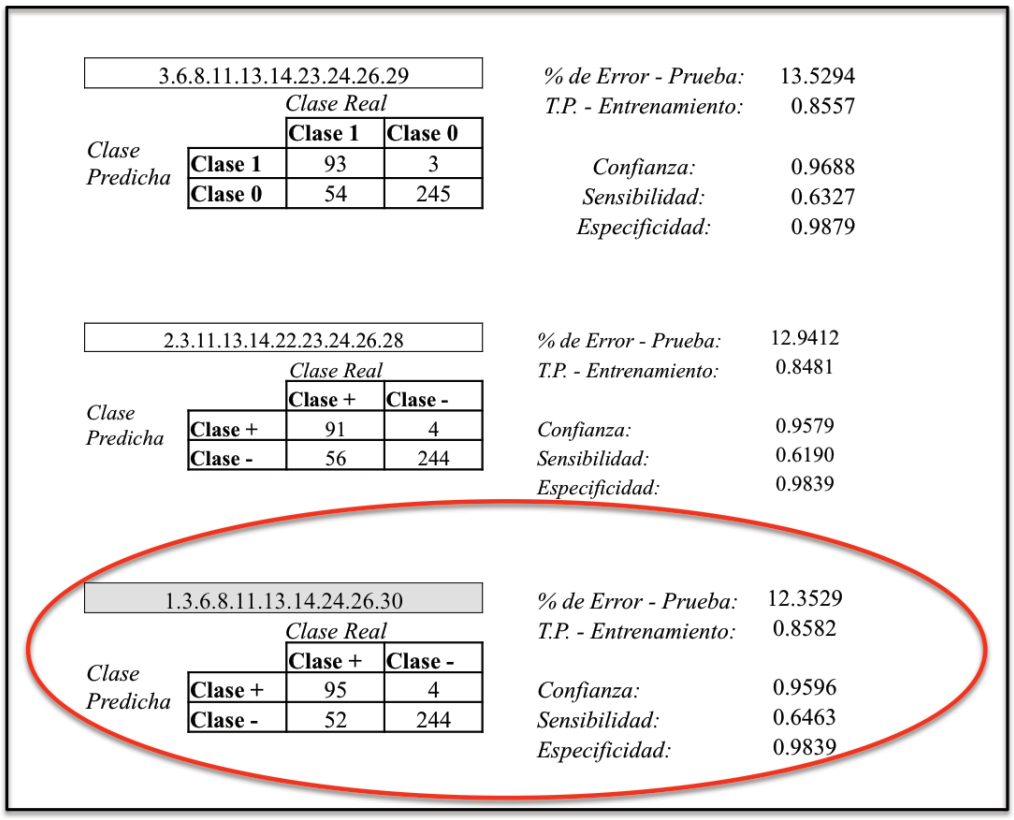



*Figura 5.5 – Análisis de 3 mejores individuos con k=10 para la BD WDBC.*

La figura 5.5, muestra que el individuo '1.3.6.8.11.13.14.24.26.30' es el mejor individuo para k=10, viendo que tiene la mejor tasa de precisión, confianza, sensibilidad y

especificidad, sobre el conjunto de entrenamiento, y el menor porcentaje de error en el conjunto de prueba, aunque no tenga la mejor calificación de correlación.

En relación a la segunda prueba para esta base de datos, k=8, en la tabla 5.9 se puede observar 3 individuos, que pueden ser considerados como los mejores, puesto que alcanzaron los mejores valores para los parámetros de evaluación, como se muestra en la tabla 5.19:

| Mejores Individuos del AG en conjunto de entrenamiento | % de Error | Tasa de Precisión | Calif. de Correl. |
|---|---|---|---|
| 1.2.6.8.11.13.24.27 | 14.1176 | 0.8354 | 7.0 |
| 2.8.13.14.18.21.22.24 | 12.9412 | 0.8506 | 6.8 |
| 2.7.11.13.14.21.24.28 | 14.1176 | 0.8506 | 7.0 |

*Tabla 5.19 – Mejores Individuos con k=8, para la BD WDBC.*

Los mejores parámetros obtenidos para el porcentaje de error fueron: 14.1176% y 12.94%; para el caso de la tasa de precisión, tenemos: 83.54% y 85.06%. En el caso de las calificaciones de la correlación, fueron: 6.8 y 7, de un total de 8. En la siguiente figura, se analizan cada uno de estos individuos, para obtener al mejor con k=8:

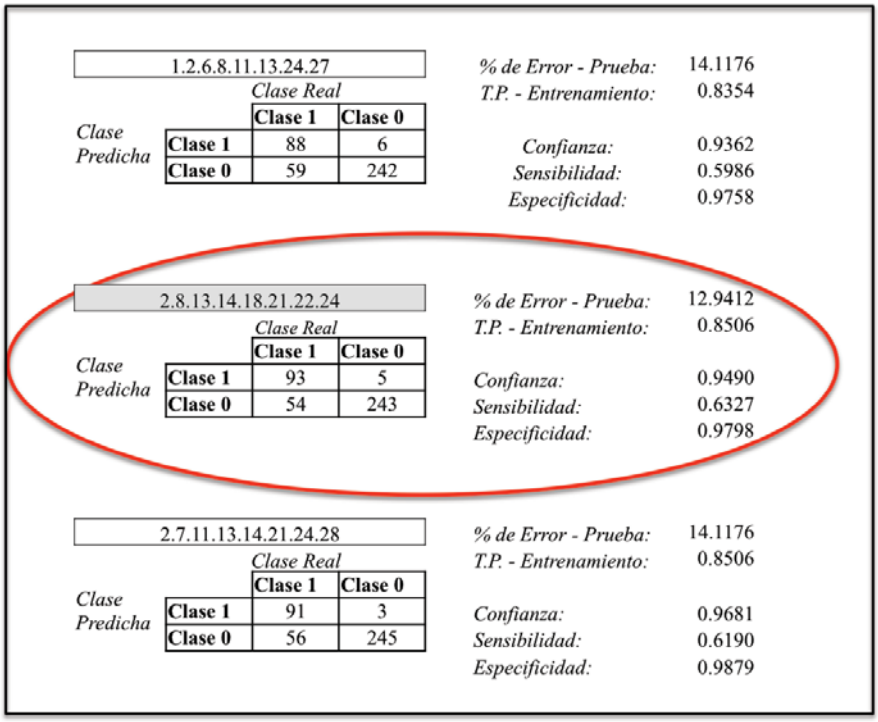



*Figura 5.6 – Análisis de 3 mejores individuos con k=8 para la BD WDBC.*

El mejor individuo es ‘2, 8, 13, 14, 18, 21, 22, 24’ que posee tanto la mejor tasa de precisión (85.06%), como el menor porcentaje de error (12.94%), no obstante, de obtener la menor calificación de correlación de los 3 analizados. Si se compara este individuo con el mejor para k=10, los atributos que permanecieron fueron 8, 13, 14 y 24; mientras que los nuevos fueron: 2, 18, 21 y 22. El porcentaje de error apenas se incremento en un 0.588% en comparación con el mejor individuo para k=10, y su tasa de precisión se redujo sólo un 0.76%.

Para el caso de k=6, la última serie de prueba que se hicieron sobre esta base de datos médica, se puede observar en la tabla 5.10, los 17 mejores individuos con sus diferentes frecuencias. De la misma forma, sobresalen 3 individuos, con los mejores valores para los parámetros de evaluación; todos con una calificación de su correlación con la clase del 5.4, de un total de 6 puntos. En la tabla 5.20, se presentan los 3 individuos junto con sus parámetros:

| Mejores Individuos del AG en conjunto de entrenamiento | % de Error | Tasa de Precisión | Calif. de Correl. |
|---|---|---|---|
| 1.2.8.11.14.24 | 15.2941 | 0.8380 | 5.4 |
| 8.11.13.14.21.23 | 15.2941 | 0.8532 | 5.4 |
| 8.11.13.14.24.28 | 14.1176 | 0.8582 | 5.4 |

*Tabla 5.20 – Mejores Individuos con k=6, para la BD WDBC.*

De la tabla anterior, se deduce fácilmente que el mejor individuo para k=6, sobre la base de datos en cuestión es, ‘8.11.13.14.24.28’, dado que es el individuo que obtuvo el menor porcentaje de error del 14.1176% y la mayor tasa de precisión sobre el conjunto de entrenamiento, del 85.82%. La figura 5.7, muestra el análisis de éste mejor individuo:

8.11.13.14.24.28

| Clase Predicha | Clase Real: Clase + | Clase Real: Clase - |
|---|---|---|
| Clase + | 93 | 2 |
| Clase - | 54 | 246 |

| | |
|---|---|
| *% de Error - Prueba:* | 14.1176 |
| *T.P. - Entrenamiento:* | 0.8582 |
| *Confianza:* | 0.9789 |
| *Sensibilidad:* | 0.6327 |
| *Especificidad:* | 0.9919 |

*Figura 5.7 – Análisis del mejor individuo con k=6 para la BD WDBC.*

A continuación se presenta el Histograma de las tres series de ejecuciones (Figura 5.8), para k=10,8,6, que muestra los atributos predominantes así como su frecuencia. También, se presenta una tabla con el mejor individuo para cada serie de ejecuciones (Tabla 5.21).

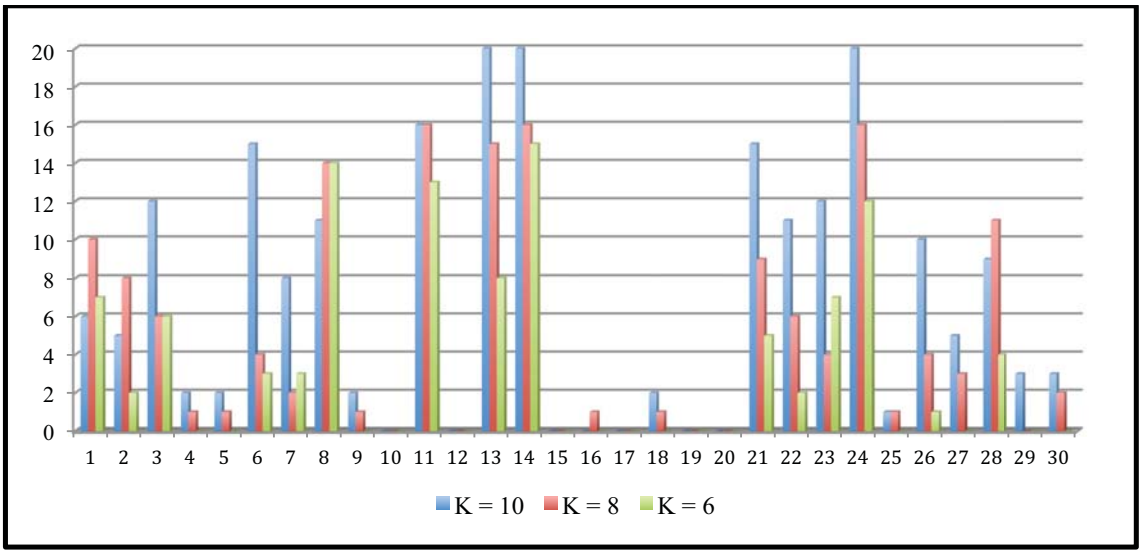



*Figura 5.8 – Frecuencias de los atributos de la BD WDBC, con k=10,8,6.*

En el histograma de la figura 5.8, se puede ver claramente que los atributos más representativos son: 1, 2, 3, 6, 8, 11, 13, 14, 21, 22, 23, 24 y 28 y que los mejores individuos, estén compuestos de algunos de estos atributos o de una combinación de ellos.

La tabla 5.21 muestra los mejores individuos para *k=10,8 y* 6:

| K | Mejor Individuo |
|---|---|
| **10** | 1.3.6.8.11.13.14.24.26.30 |
| **8** | 2.8.13.14.18.21.22.24 |
| **6** | 8.11.13.14.24.28 |

*Tabla 5.21 – Mejores Individuos para la BD WDBC.*

Es interesante también observar, que los atributos 8, 13, 14 y 24 son constantes para los tres mejores individuos, mientras que el atributo 28, aparece sólo hasta el final, con k=6, sustituyendo quizá al 26 y 30, del mejor individuo para k=10.

### 5.3.3 Análisis de la Base de Datos Médica 'WPBC'.

El análisis de los mejores individuos para la base de datos médica 'WPBC', con k=10 puede basarse en la tabla 5.11, donde podemos observar 23 mejores individuos obtenidos de las 16 ejecuciones realizadas para k=10. De ellos, 4 poseen los parámetros de evaluación más óptimos, como se muestra en la tabla 5.22:

| Mejores Individuos del AG en conjunto de entrenamiento | % de Error | Tasa de Precisión | Calif. de Correl. |
|---|---|---|---|
| 1.2.3.11.12.16.17.18.26.27 | 8.4746 | 0.8889 | 7.8 |
| 1.2.3.12.13.14.17.18.22.26 | 6.7797 | 0.9259 | 7.4 |
| 1.2.3.4.13.14.21.26.27.32 | 8.4746 | 0.8963 | 7.8 |
| 1.2.3.12.13.16.17.18.24.27 | 8.4746 | 0.9111 | 8 |

*Tabla 5.22 – Mejores Individuos con k=10, para la BD WPBC.*

En la figura, 5.9 se analizan con mayor detalle estos 4 mejores individuos:

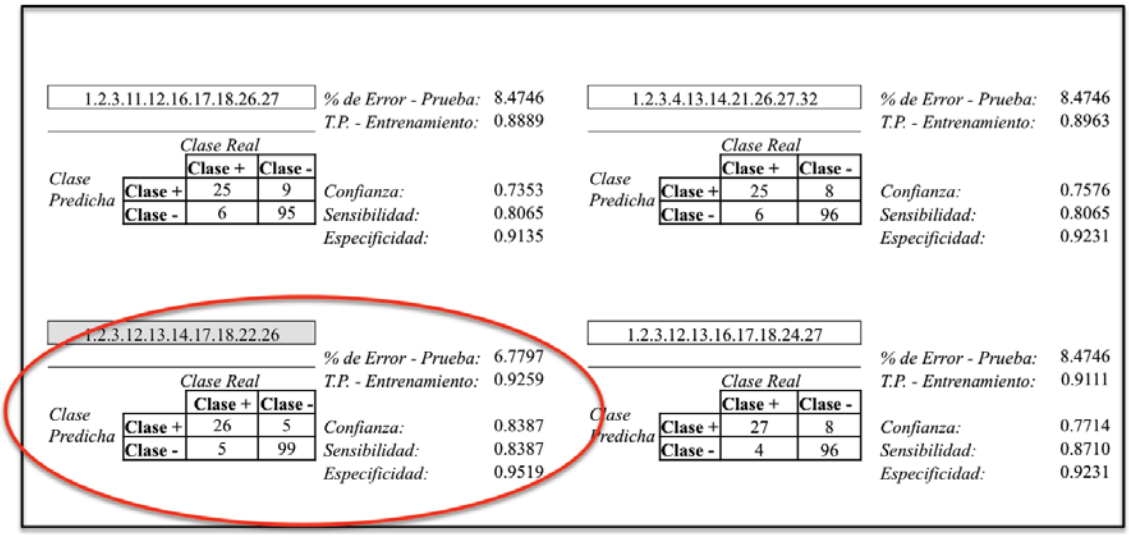



*Figura 5.9 – Análisis de 4 mejores individuos con k=10 para la BD WPBC.*

No obstante, el individuo '1, 2, 3, 12, 13, 14, 17, 18, 22 y 26' tiene la menor calificación de su correlación conjunta con la clase (7.4), posee el porcentaje de error más bajo sobre el conjunto de prueba (6.7797%) y la mayor tasa de precisión de la clasificación sobre el conjunto de entrenamiento (92.59%), por lo que se le considera como el mejor individuo con k=10, para esta segunda base de datos médica.

Para el caso en el que k=8, donde los individuos están formados por 8 atributos, la tabla 5.12 muestra 22 mejores individuos obtenidos durante esta serie de 16 ejecuciones, de los cuales 4, poseen un porcentaje de error de 6.7797% y 8.4746%; una tasa de precisión de 88.89%, 91.11% y 91.85% y una calificación de su correlación de 6.2 y 6.4, de un total de 8 puntos.

| Mejores Individuos del AG en conjunto de entrenamiento | % de Error | Tasa de Precisión | Calif. de Correl. |
|---|---|---|---|
| 2.3.13.16.17.18.24.27 | 8.4746 | 0.8889 | 6.4 |
| 1.2.3.13.16.17.18.26 | 6.7797 | 0.9111 | 6.4 |
| 1.2.3.13.16.17.18.24 | 6.7797 | 0.9185 | 6.4 |
| 2.3.11.13.16.24.27.32 | 8.4746 | 0.9111 | 6.2 |

*Tabla 5.23 – Mejores Individuos con k=8, para la BD WPBC.*

Asimismo, en la figura 5.10 se presenta el análisis de estos 4 mejores individuos:

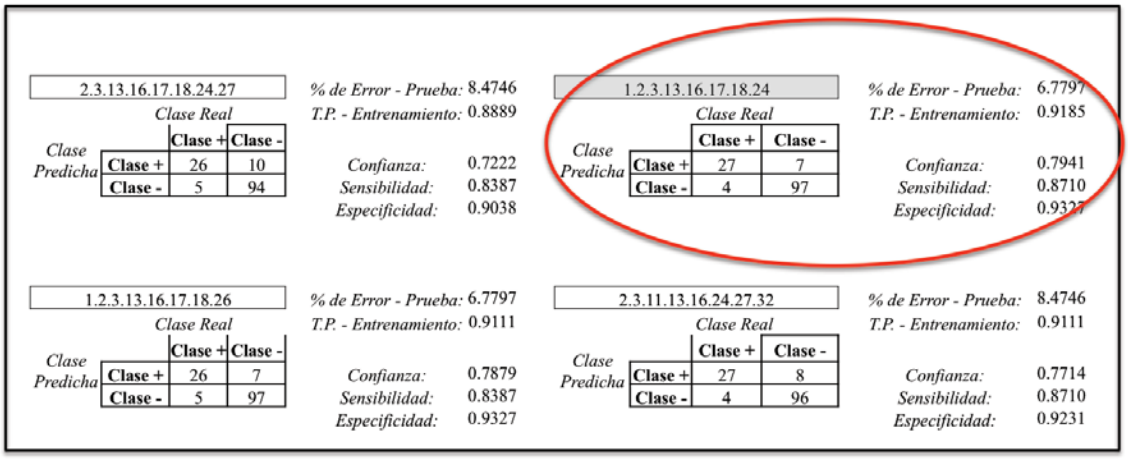



*Figura 5.10 – Análisis de 4 mejores individuos con k=8 para la BD WPBC.*

Como muestra la figura 5.10, no obstante el individuo '1.2.3.13.16.17.18.26' tiene el mismo porcentaje mínimo de error (6.7797%) que el individuo '1.2.3.13.16.17.18.24', éste último posee una mejor tasa de precisión de 91.85%, razón por la que se le considera el mejor individuo para k=8. Cabe señalar, que el porcentaje mínimo de error de la clasificación sobre el conjunto de entrenamiento para ésta serie de ejecuciones, fue le mismo que el obtenido cuando k=10.

La última prueba que se hizo con la base de datos médica WPBC, fue para k=6, donde se tiene un total de 19 mejores individuos, como se puede observar en la tabla 5.13. De los cuáles 3, poseen los parámetros más óptimos en el porcentaje de error y la tasa de precisión, como se muestra en la siguiente tabla:

| Mejores Individuos del AG en conjunto de entrenamiento | % de Error | Tasa de Precisión | Calif. de Correl. |
|---|---|---|---|
| 1.2.13.17.27.32 | 8.4746 | 0.8963 | 5.0 |
| 1.2.3.13.16.27 | 6.7797 | 0.9037 | 5.0 |
| 1.2.3.24.27.32 | 8.4746 | 0.8815 | 5.0 |

*Tabla 5.24 – Mejores Individuos con k=6, para la BD WPBC.*

Aquí se observa una calificación constante de la correlación conjunta de los individuos de 5, de un total de 6 puntos. Los valores mínimos para la tasa de precisión son de 88.15%, 89.63% y 90.37%; y el porcentaje de error fue de 6.7797% y 8.4746%. De igual forma, es

fácil deducir que el individuo '1, 2, 3, 13, 16, 27' es el mejor individuo con k=6, de los tres presentados en la tabla 5.24, para ésta última base de datos, porque posee el menor porcentaje de error en la clasificación del conjunto de prueba y una mayor tasa de precisión sobre el conjunto de entrenamiento. La figura 5.11 muestra los detalles de sus parámetros:

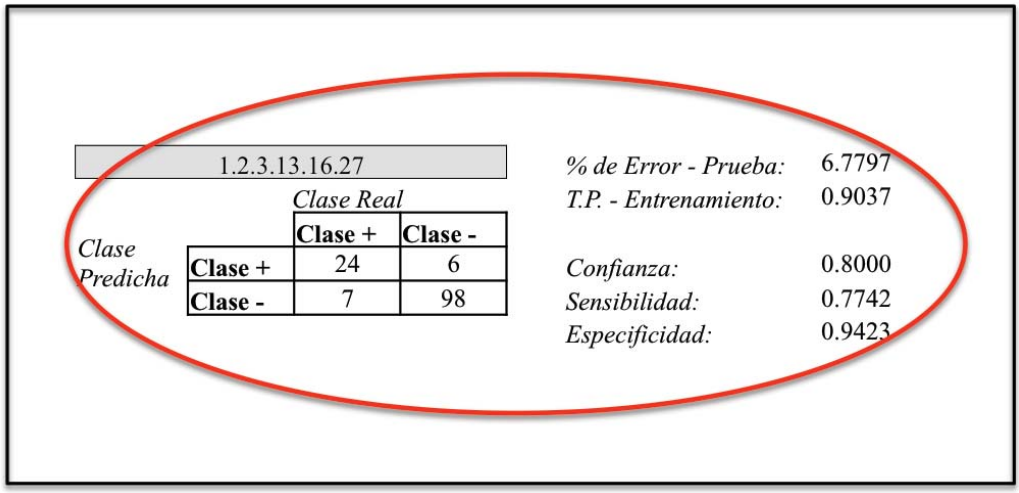



*Figura 5.11 – Análisis del mejor individuo con k=6 para la BD WPBC.*

Finalmente, se presenta el histograma que muestra las frecuencias de los atributos durante las 16 ejecuciones para cada uno de los diferentes valores k=10, 8 y 6. También, se presenta la tabla 5.25 con el mejor individuo para cada serie de ejecuciones:

| K | Mejor Individuo |
|---|---|
| **10** | 1.2.3.12.13.14.17.18.22.26 |
| **8** | 1.2.3.13.16.17.18.24 |
| **6** | 1.2.3.13.16.27 |

En esta tabla se puede observar cierta coherencia con los atributos que pertenecen a los mejores individuos elegidos por el sistema.

*Tabla 5.25 – Mejores Individuos para la BD WPBC.*

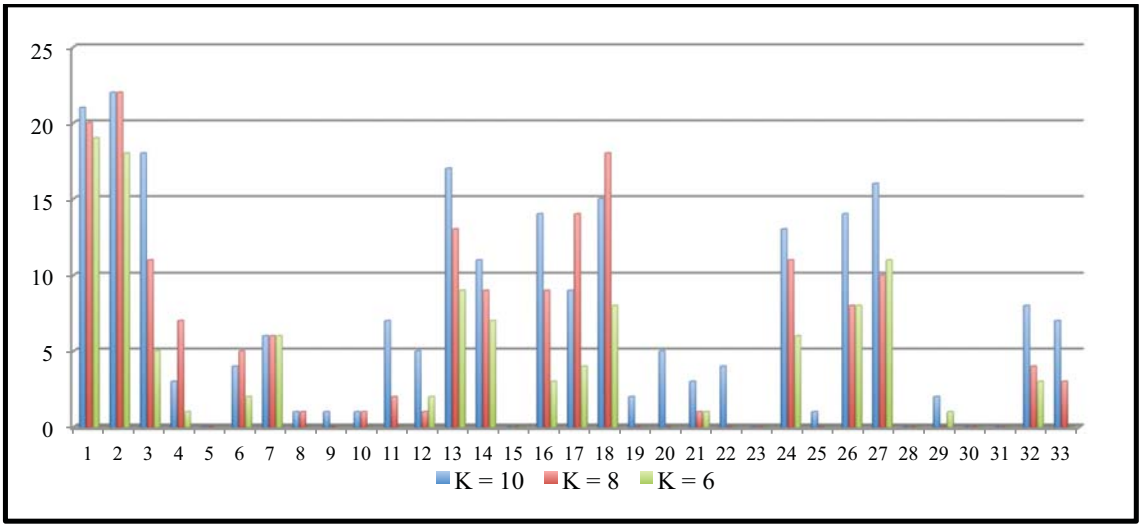



*Figura 5.12 – Frecuencias de los atributos de la BD WPBC, con k=10,8,6.*

En la figura 5.12, se muestra una clara tendencia de los atributos más representativos de la base de dato WPBC, los cuales son: 1, 2, 3, 13, 14, 16, 17, 18, 24, 26 y 27; atributos que en su mayoría se encuentran en los mejores individuos obtenidos por el sistema 'Attribute_Classification'.

## 5.4 SELECCIÓN DE CARACTERÍSTICAS EN WEKA.

Para validar los resultados del modelo propuesto en este trabajo de tesis, se aplicará un algoritmo de selección de características en Weka a las tres bases de datos con las que se ha trabajado en lo largo de esta Tesis. Dentro del panel para la selección de atributos de "Weka" se encuentran varios métodos para la selección de subconjuntos de atributos en conjunción con diferentes métodos de búsqueda. A continuación, se presentan los resultados obtenidos al aplicar el método de selección *'CfsSubsetEval'*, que considera el valor predictivo de cada atributo junto con la redundancia en el conjunto completo; con el método de búsqueda '*BestFirst*', que se encarga de realizar una búsqueda exhaustiva hacia delante. (Anexo B).

En la siguiente figura, se presentan los resultados en Weka, para la base de datos de Portafolios de Inversión:

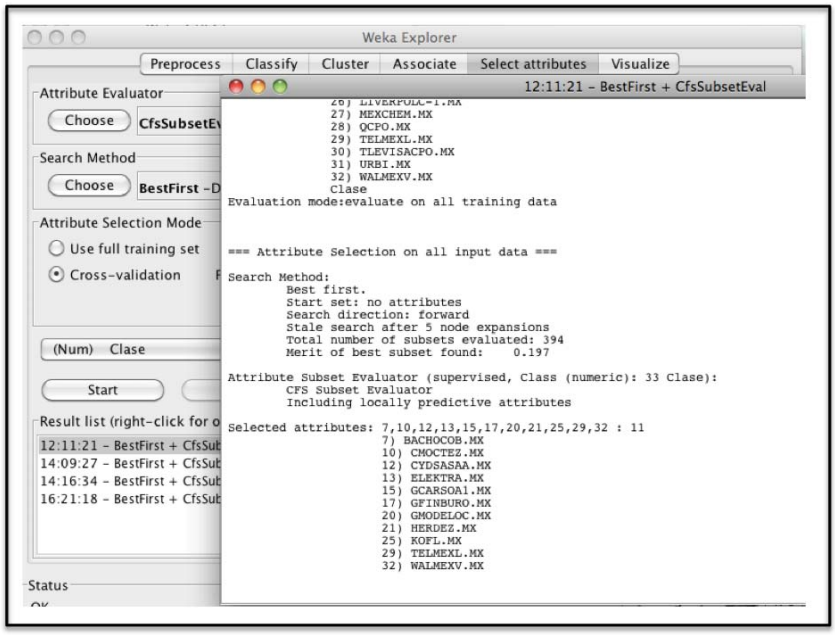



*Figura 5.13 – Resultado de la selección de atributos con Weka, para BD_Portafolios.*

| K = 10 | Atributo | K = 8 | Atributo | K = 6 | Atributo |
|---|---|---|---|---|---|
| 1 | ALFAA.MX | 7 | **BACHOCOB.MX** | 4 | ARA.MX |
| 4 | ARA.MX | 12 | **CYDSASAA.MX** | 7 | **BACHOCOB.MX** |
| 7 | **BACHOCOB.MX** | 13 | **ELEKTRA.MX** | 12 | **CYDSASAA.MX** |
| 12 | **CYDSASAA.MX** | 20 | **GMODELOC.MX** | 20 | **GMODELOC.MX** |
| 13 | **ELEKTRA.MX** | 22 | HOMEX.MX | 22 | HOMEX.MX |
| 15 | **GCARSOA1.MX** | 28 | QCPO.MX | 29 | **TELMEXL.MX** |
| 20 | **GMODELOC.MX** | 29 | **TELMEXL.MX** | | |
| 21 | **HERDEZ.MX** | 32 | **WALMEXV.MX** | | |
| 24 | KIMBERA.MX | | | | |
| 30 | TLEVISACPO.MX | | | | |

*Tabla 5.26 – Mejores Individuos seleccionador por el sistema propuesto. (BD_Portafolios)*

De todos los atributos seleccionados en Weka, sólo 3 atributos no figuran dentro de los mejores individuos seleccionados por *'Attribute_Classification'*, los cuales son: 10)CMOCTEZ.MX, 17)GFINBURO.MX y 25)KOFL.MX.

Para la base de datos médica WDBC, los atributos seleccionados por Weka, se presentan a continuación:

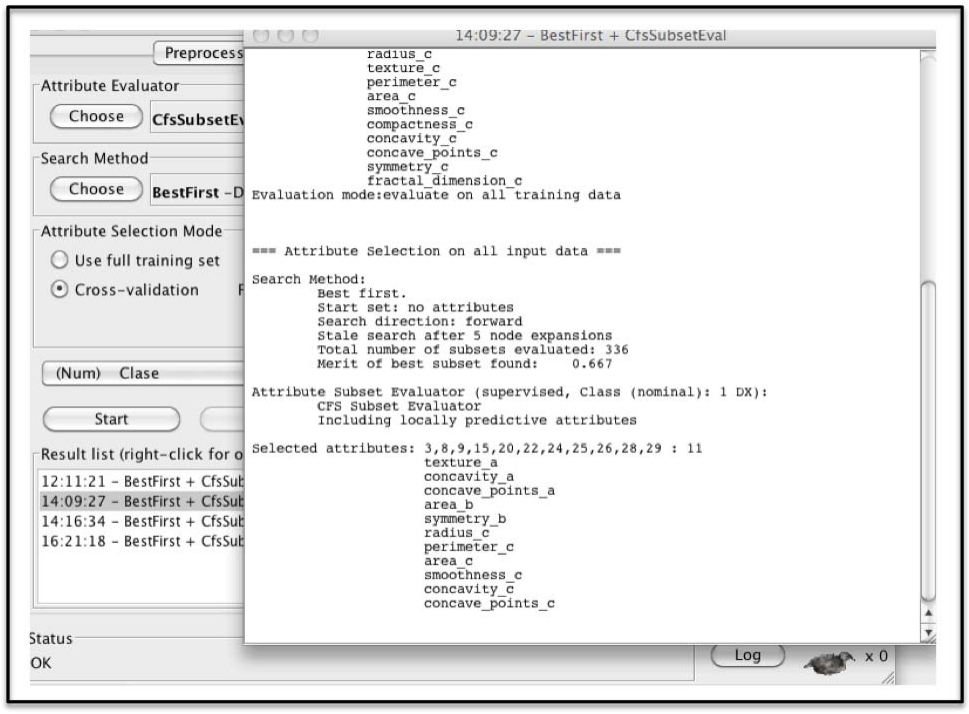



*Figura 5.14 – Resultado de la selección de atributos con Weka, para BD_WDBC.*

| K = 10 | Atributo | K = 8 | Atributo | K = 6 | Atributo |
|---|---|---|---|---|---|
| 1 | radius_a | 2 | **texture_a** | 8 | **concave_points_a** |
| 3 | perimeter_a | 8 | **concave_points_a** | 11 | radius_b |
| 6 | compactness_a | 13 | perimeter_b | 13 | perimeter_b |
| 8 | **concave_points_a** | 14 | **area_b** | 14 | **area_b** |
| 11 | radius_b | 18 | concave_points_b | 24 | **area_c** |
| 13 | perimeter_b | 21 | **radius_c** | 28 | **concave_points_c** |
| 14 | **area_b** | 22 | texture_c | | |
| 24 | **area_c** | 24 | **area_c** | | |
| 26 | compactness_c | | | | |
| 30 | fractal_dimension_c | | | | |

*Tabla 5.27 – Mejores Individuos seleccionador por el sistema propuesto. (BD_WDBC)*

Para esta base de datos en particular, fueron 5, de 11, los atributos que el sistema '*Attribute_Classification*' no seleccionó, los cuales son: Concavity_a, Symmetry_b, Perimeter_c, Smoothness_c y Concavity_c.

En el caso de la última base de datos, WPBC, los atributos seleccionados por Weka, fueron sólo 2, 'Time' y 'Lymph_node_status', como lo muestra la figura 5.15:

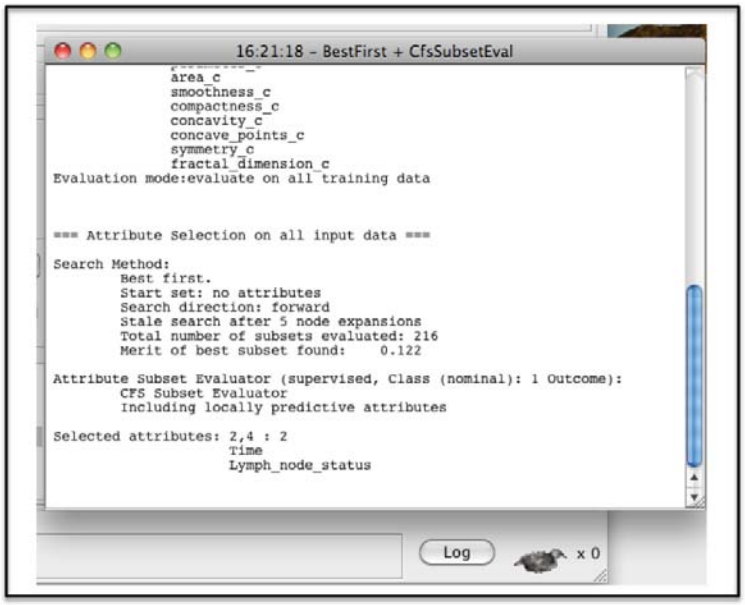



*Figura 5.15 – Resultado de la selección de atributos con Weka, para BD_WPBC.*

Estos dos atributos fueron seleccionados dentro de los tres mejores individuos, para k=10,8,6, por el sistema '*Attribute_Classification*', como se muestra en la tabla 5.28:

| **K = 10** | **Atributo** | **K = 8** | **Atributo** | **K = 6** | **Atributo** |
|---|---|---|---|---|---|
| 1 | **Time** | 1 | **Time** | 1 | **Time** |
| 2 | Tumor_size | 2 | Tumor_size | 2 | Tumor_size |
| 3 | **Lymph_node_status** | 3 | **Lymph_node_status** | 3 | **Lymph_node_status** |
| 12 | symmetry_a | 13 | fractal_dimension_a | 13 | fractal_dimension_a |
| 13 | fractal_dimension_a | 16 | perimeter_b | 16 | perimeter_b |
| 14 | radius_b | 17 | area_b | 27 | area_c |
| 17 | area_b | 18 | smoothness_b | | |
| 18 | smoothness_b | 24 | radius_c | | |
| 22 | symmetry_b | | | | |
| 26 | perimeter_c | | | | |

*Tabla 5.28 – Mejores Individuos seleccionador por el sistema propuesto. (BD_WPBC)*

En el capítulo VI, utilizaremos algunos de los mejores individuos de cada base de datos, para aplicar una técnica de Minería de Datos, siguiendo con el procedimiento de la extracción de conocimiento de las bases de datos.

# CAPÍTULO VI: DISCUSIÓN Y CONCLUSIONES.

## 6.1 DISCUSIÓN.

### 6.1.1 Impacto del presente trabajo.

En este capítulo, ahondaremos en los resultados obtenidos por el sistema propuesto en este trabajo de investigación, llamado *"Attribute_Classification"*, y mostrados en el capítulo anterior. El algoritmo evaluó a cada individuo con dos parámetros: la maximización de la tasa de precisión más la calificación de correlación, al clasificar los registros en el conjunto de entrenamiento; y la minimización del error al clasificar los registros en el conjunto de prueba. Por tanto, el algoritmo logró resultados significativos al clasificar la base de datos financiera 'BD_Portafolios', debido a que su precisión se mantuvo por arriba del 91% de éxito, mientras que el porcentaje de error no ascendió más allá del 16%. Para el caso de la serie de ejecuciones sobre la base de datos médica WDBC, los resultados fueron menos plausibles, aunque no del todo mal; su tasa de precisión se encontró entre el 81.26% y el 85.82% y en ocasiones, su porcentaje de error llegó al máximo de 20.6%, siendo ésta la base de datos que obtuvo los más 'bajos' parámetros. En cambio, la segunda base de datos médica, WPBC, mejoró un tanto el desempeño de su precisión, obteniendo un mínimo porcentaje en la tasa de precisión del 85.19%, y un máximo en el porcentaje de error del 23.7288%. Estos resultados, muestran la variabilidad y eficacia del sistema al ser aplicado en diversas bases de datos.

Para la base de datos financiera, cabe señalar que, aún cuando se pretende maximizar el rendimiento del individuo, la mayoría de las empresas obtenidas por el algoritmo genético, son empresas conservadoras que no tuvieron mucha variación en sus precios originales, concentrándose alrededor de la media; en cambio, otras empresas, pese a que tendrían un precio más alto de sus acciones, sufrieron variaciones extremas, con caídas intempestivas de los mismos, ocasionando un rendimiento negativo. Por tanto, se puede deducir que el algoritmo genético presentado tiene un carácter conservador para problemas de ésta índole. Asimismo, se sugiere para futuras investigaciones, el hacer de este trabajo un problema

multiobjetivo, con el fin de minimizar el riesgo del portafolio de inversión final; un aspecto de suma importancia dentro del mundo de las finanzas.

En el caso de las bases de datos médicas, fue interesante observar que cuando el algoritmo elegía un atributo que pertenece al grupo de las 'medias' (_a) ó de los peores valores (_c), también elegía el mismo atributo dentro del grupo de los errores estándares (_b), proveyendo un panorama más general hacia el problema.

Una de las razones más importantes que justifican la presente investigación, es la 'Maldición de la Dimensionalidad', estableciendo que las técnicas de Minería de Datos totalmente eficientes para bajas dimensiones, no pueden proveer ningún resultado significativo cuando el número de características va más allá de un modesto tamaño de 10 atributos. [Chizi & Maimon 2005]

Teniendo en cuenta que la dimensionalidad constituye un serio obstáculo para la eficiencia de la mayoría de los algoritmos de Minería de Datos (Maimon & Last 2000), al disminuir las bases de datos de 33, 32 y 31 atributos activos, a sólo 10, 8 ó 6, un tercio del total de los atributos, con un costo en la pérdida de la información, de un 10% y 20%, se cumple con la meta establecida de presentar un algoritmo robusto que provea resultados significativos para el problema de la Reducción de la Dimensionalidad, dentro del proceso de KDD.

En general, una de las ventajas de este sistema, es la búsqueda aleatoria que realiza sobre el espacio de posibles soluciones, ejecutando saltos que permiten encontrar soluciones globales óptimas en lugar de soluciones locales, característica especialmente útil cuando el conjunto de atributos es muy grande. Estos métodos heurísticos de búsqueda, son apropiados para resolver problemas donde el dominio de la solución pueda resultar demasiado extenso. Además, estos algoritmos no requieren un amplio conocimiento del problema en cuestión, por lo que se amplía su área de aplicabilidad, dependiendo en gran medida de la determinación adecuada de su función de aptitud. Sin una función de aptitud adecuada, el algoritmo podría estar proporcionando resultados erróneos o falsos.

Es importante subrayar, que el algoritmo presentado en esta tesis, trabaja en dos niveles distintos sobre la base de datos en cuestión, siguiendo un enfoque de envolvimiento. Primero, trabaja directamente sobre los atributos al maximizar la correlación de cada uno de ellos con la clase a predecir. Y segundo, trabaja sobre los registros en función de su media y desviación estándar, 'barriendo' así, el conjunto completo de datos de entrenamiento. Además, previene el sobreajuste de las soluciones, al 'entrenar' al mejor individuo sobre el conjunto de entrenamiento, para después 'probarlo', sobre un conjunto de prueba completamente desconocido.

### 6.1.2 Aplicación de 3 Técnicas de Minería de Datos en WEKA.

Para continuar con el procedimiento de extracción de la información de una base de datos, se han aplicado diferentes técnicas de Minería de Datos a los mejores individuos elegidos por el sistema *"Attribute_Classification"*, para k=10 y para la base de datos original, con el fin de explicar la utilidad de éste.

Para el primer conjunto de datos, BD_Portafolio, se aplicó un árbol de decisión podado M5P (véase el Glosario de este trabajo de investigación), tanto al conjunto de datos conformado por el individuo '7.12.13.20.22.28.29.32' más la clase; como a la base de datos completa de 32 atributos predictores más la clase. Como resultado, el árbol de decisión arrojó 130 reglas para la base de datos original, y sólo una regla para el conjunto de datos del mejor individuo. La regla obtenida por el árbol de decisión para el subconjunto de datos del mejor individuo es:

Clase = 0.7164 * 7) BACHOCOB.MX + 1.2259 * 12) CYDSASAA.MX + 0.9322 * 13) ELEKTRA.MX
+ 1.123 * 15) GCARSOA1.MX + 0.4919 * 21) HERDEZ.MX + 0.5188

La cual expone los pesos que contendría cada empresa elegida en una inversión de capital para que dicha inversión sea mayor ó igual a la de la Bolsa Mexicana de Valores. La figura 6.1 muestra la comparación de los dos modelos obtenidos:

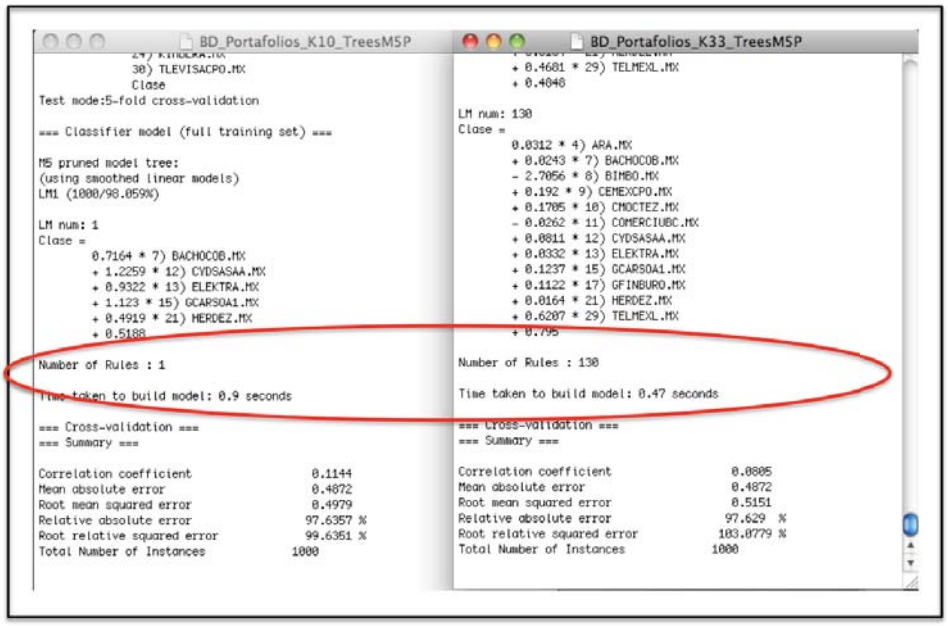



*Figura 6.1 – Reporte del árbol M5P, mejor individuo (izq.) y B.D. original (der.).*

En relación a la primera base de datos médica, WDBC, se aplicó un árbol de decisión NBT que utiliza la probabilidad condicional de Naïve Bayes como clasificador. Para la base de datos original de 31 atributos predictores, se creó un árbol de 23 nodos y 12 ramas, calificando correctamente el 95.9% de instancias; mientras que para el conjunto de datos del mejor individuo con k=10, el resultado fue un árbol de 7 nodos y 4 ramas, calificando correctamente el 93.567% de instancias. La figura 6.2 muestra ambos árboles de decisión:

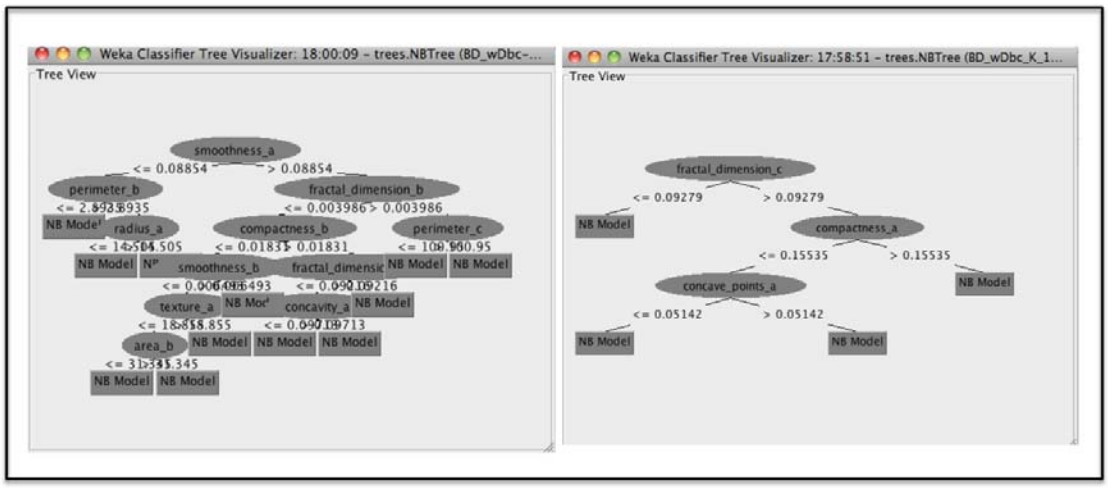



*Figura 6.2 – Árbol de Decisión NBT, base de datos original (izq.) y mejor individuo (der.).*

El árbol construido desde el conjunto de datos del mejor individuo, es sólo 2% menos preciso que el árbol construido del conjunto de datos original, con una diferencia en su dimensionalidad de 21 atributos; ambos clasificando correctamente más del 90% de instancias. Asimismo, la ventaja del árbol derecho sobre el izquierdo en la figura 6.2, es que es más fácil de interpretar, sobre todo para los usuarios finales que aplicaran dichas reglas para una decisión concreta. Recordemos, que no todos los modelos largos y complejos son los mejores; entre más simples y entendibles sean, mejor.

Por último, se aplicó una regresión logística simple, tanto para la tercera base de datos original, WPBC, como para el mejor individuo elegido con k=10; donde se clasificaron correctamente el 81.36% para el conjunto del mejor individuo y 79.66% para la base de datos original, mejorando el rendimiento de la clasificación con sólo 10 atributos, como lo muestra la siguiente figura:

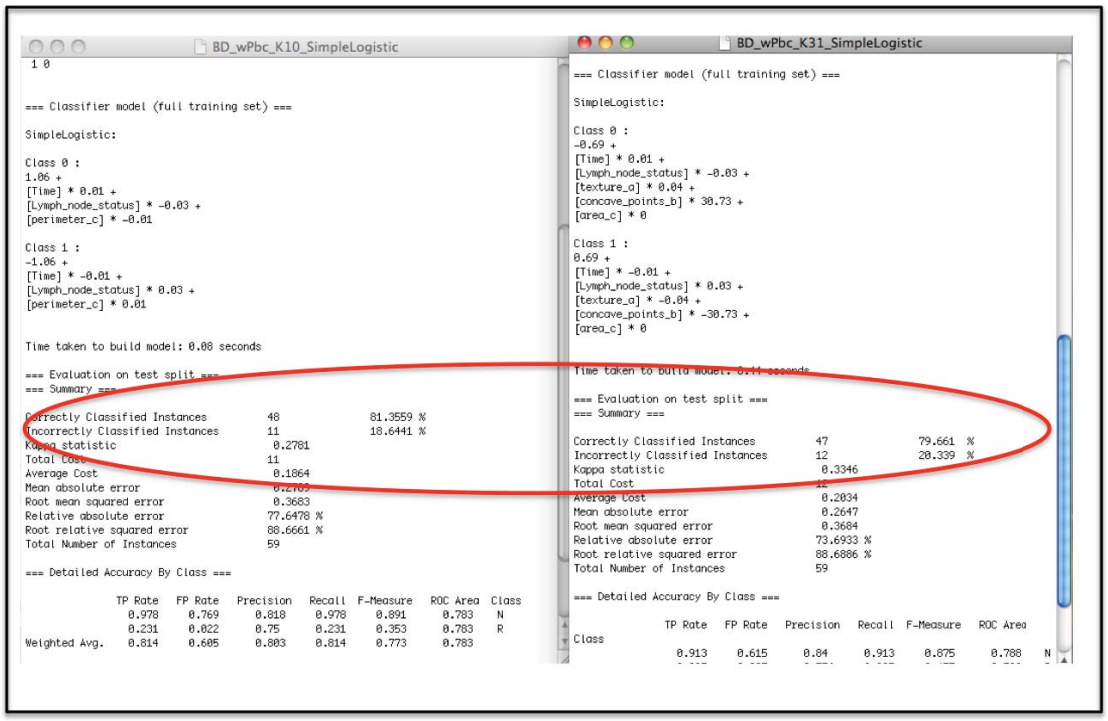



*Figura 6.3 – Reporte Regresión logística, mejor individuo (izq.) y B.D. original (der.).*

Como se observa en la figura 6.3, los resultados obtenidos justifican en gran medida el desempeño del sistema *'Attribute_Classification'*, mejorando la clasificación de instancias posteriores con una dimensionalidad mucho menor.

Con los tres ejemplos anteriores, se muestra que el algoritmo presentado en este trabajo de investigación, provee resultados significativos para el problema de la 'Reducción de la Dimensionalidad' en grandes bases de datos. Además, una de las principales desventajas de los sistemas de minería de datos, como lo es el banco de trabajo WEKA, es que la mayor parte de su funcionalidad es aplicable solamente si el conjunto completo de datos es mantenido en la memoria principal. Razón por la cuál, es mucho mejor trabajar con una base de datos con una dimensionalidad reducida, que con las bases de datos originales. Como hemos visto en los ejemplos anteriores, la reducción de la dimensionalidad mantiene y en algunos casos, mejora el rendimiento al aplicar técnicas de minería de datos sobre los subconjuntos reducidos, con una pérdida mínima en la información ya que, es mejor utilizar los recursos en el algoritmo de minería de datos para tener una mejor exactitud en el conocimiento de salida, que en la manipulación de los datos en sí.

## 6.2 CONCLUSIONES.

### 6.2.1 Conclusiones Generales.

" ...Y si la evolución natural permitió que una pequeña e insignificante cadena núcleo proteínica, con el paso del tiempo hiciera que sus descendientes llegaran a ser hombres, podría hacer que la solución de un modelo matemático converja a la solución óptima."
David A. Fogel [De Jong et al. 1997]

El estudio de la rama de las Ciencias de la Computación, llamada "*Computación Evolutiva*" ha sido sumamente interesante, no sólo porque sus organismos, los seres vivos y los procesos naturales evolutivos en general, son los sistemas óptimos por excelencia, con una capacidad inteligente de aprendizaje desarrollada a través de millones de años; sino también, por todos los misterios y retos que imponen a la ciencia actual, con la meta de obtener un mejor entendimiento de los sistemas de optimización presentes a nuestro alrededor.

En particular, los "*Algoritmos Genéticos*", con su habilidad de trabajar con poblaciones de individuos a evolucionar, proveen herramientas útiles para problemas de optimización dinámicos, al encontrar los mejores individuos, a través de la recombinación de sus genes más exitosos, heredando el conocimiento ganado a través de las generaciones ó iteraciones y, ayudándose del factor aleatorio para converger más rápidamente a una solución adecuada. Así, la naturaleza nos enseña a resolver problemas, en su mayoría no determinísticos encontrando a través de un procedimiento natural, una solución "aceptable" que en algunos casos es la óptima.

Por otro lado, dentro del paradigma de la Extracción de Conocimiento de las bases de datos KDD, la fase de '*Preparación de los Datos*' es la fase que consume más tiempo y trabajo; debido a que es necesario contar con los datos adecuados y necesarios para obtener un modelo que exprese acciones a tomar en un entorno de operación. Dentro de esta fase, nos encontramos con problemas como la 'Maldición de la Dimensionalidad', que limita la

habilidad para ajustar diferentes modelos de Minería de Datos, cuando existe un gran número de variables de entrada, un problema muy recurrente, dado el acelerado incremento de la información con la que se cuenta hoy en día. Sin embargo, también se corre el riesgo de ignorar información importante, teniendo la disyuntiva de pagar cierto costo en la precisión de la predicción, a costa de una mejor manipulación de los datos. Lo anterior ha dado paso a desarrollar numerosas investigaciones en el área, como algoritmos de reducción de la dimensionalidad, sistemas que manejen cantidades de información estratosféricas, y herramientas como BigData.

De esta forma, este trabajo de investigación se enfocó en la tarea de Selección de Características a través de un Algoritmo Genético. El modelo presentado redujo exitosamente la dimensionalidad de tres bases de datos. El modelo, siguió un enfoque de envolvimiento, clasificando todos los registros, sólo para los individuos en cuestión (con 6, 8 y 10 atributos por individuo). Por la misma razón, este modelo se presta para ser usado sobre bases de datos con alta dimensionalidad, ya que permite a los algoritmos de aprendizaje operar más rápido y eficientemente. Así, uno de los frutos de este algoritmo es permitir la correcta aplicación de las técnicas de minería de datos, obteniendo, patrones, reglas, o herramientas gráficas que nos ayudan a lidiar con un problema en específico. Además, se sugiere realizar una futura investigación sobre construir un Algoritmo Genético Multicriterio para la Selección de Características en una base de datos, donde el tamaño de ***k*** sea dinámico y elegido por el AG, optimizando tanto el total de atributos a ser escogidos, como los atributos más representativos de la base de datos.

### 6.2.2 Conclusiones Particulares.

Recordando que la meta es “reducir la dimensionalidad con un mínimo de pérdida de información”; se ha demostrado que el algoritmo genético para la selección de características en una base de datos, desarrollado en este trabajo de tesis, provee resultados significativos para el problema de la Reducción de la Dimensionalidad; ya que, al reducirse el número de atributos, los algoritmos de minería de datos pueden manipular los datos de manera más efectiva, probando diferentes modelos a fin de obtener el mejor de ellos. En

algunos casos, una vez que se ha reducido la dimensionalidad, la precisión de un algoritmo de minería de datos, sobre la futura clasificación puede ser mejorada; otra veces, el resultado es una representación más compacta y fácil de interpretar del concepto perseguido.

También es necesario considerar que la selección de atributos es un proceso costoso, y contradice la asunción de que toda la información (incluyendo los atributos) es requerida para alcanzar la máxima precisión; sin embargo, el basto incremento de la información, significa un verdadero problema en la extracción de conocimiento de las bases de datos. Por lo tanto, a veces es preferible pagar un mínimo costo en la precisión con un reducido conjunto de datos, a obtener modelos indescriptibles e igualmente no confiables.

Por otro lado, los algoritmos genéticos son algoritmos de optimización robustos de búsqueda aleatoria, fáciles de comprender, con una estructura bien definida, y en donde el problema principal es saber definir correctamente la Función de Aptitud que determinará quién es el mejor individuo, en este caso, el subconjunto de datos que mejor clasifique la clase objetivo. También es de suma importancia, definir correctamente la codificación de los individuos, porque de ello dependerá la optimización de su ejecución y la correcta decodificación de los resultados. Gracias a sus operadores de cruza, mutación y selección, pueden recorrer el espacio de soluciones, previniendo caer en óptimos locales, todo ello, manipulando en cada iteración, sólo a los individuos y sus datos, y no a la base de datos completa.

Por último, la Licenciatura de Matemáticas Aplicadas y Computación ofrece una gama de conocimientos bastante completa, tanto en el área de las matemáticas como de las ciencias de la computación, para la investigación y desarrollo del proceso completo de la extracción de conocimiento desde las Bases de Datos, así como la oportunidad de abordar nuevas áreas de conocimiento dentro de la Estadística, el Cómputo Evolutivo y el Aprendizaje de Máquina, creando modelos eficientes y confiables en la exploración, análisis y solución de problemas del mundo real.

# REFERENCIAS.


[Aluja 2001] Aluja, T. "La Minería de Datos, entre la Estadística y la Inteligencia Artificial", *Qüestió Vol. 25.3*, pp. 479-498, Cataluña España, 2001.

[Agrawal & Shafer 1996] Agrawal, R. y J.C. Shafer. "Parallel Mining of Association Rules", *IEEE Transaction on Knowledge and Data Engineering 8*, pp. 962-969, 1996. En: Maimon, O. y L. Rokach. *The Data Mining and Knowledge Discovery Handbook*, Springer, USA, 2005.

[Almuallim & Dietterich 1992] Almuallim, H. y T.G. Dietterich. "Efficient algorithms for identifying relevant features". En: *Proceedings of the Ninth Canadian Conference on Artificial Intelligence*, pp. 38-45, Morgan Kaufmann, 1992.

[Angeline 1997] Angeline, P.J. "Competitive Fitness Evaluation", Cap. C4.3. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Angeline & Fogel 1997] Angeline, P.J. y D.B. Fogel. "Other Representations", Cap. C1.8. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Bäck et al. 1997] Bäck, T., Fogel, D.B., Whitley, D. y P.J. Angeline. "Mutation", Cap. C3.2. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Bäck et al. 2000] Bäck, T., Fogel, D.B. y T. Michalewicz (Eds.). *Evolutionary Computation 1: Basic Algorithms and Operators*, pp. 59-63, Institute of Physics Publishing, 2000.

[Bäck 2000a] Bäck T. "Introduction to Evolutionary Algorithms". En: Back, T., Fogel, D.B. y T. Michalewicz (Eds.). *Evolutionary Computation 1: Basic Algorithms and Operators*, pp. 59-63, Institute of Physics Publishing, 2000.

[Baeza-Yates & Ribeiro-Neto 1999] Baeza-Yates, R. y B. Ribeiro-Neto. *Modern Information Retrieval*, Addison-Wesley, 1999.

[Bala et al. 1995] Bala, J., De Jong, K., Huang, J., Vafaie, H. y H. Wechsler. "Hybrid Learning using Genetic Algorithms and Decision Trees for Pattern Classification", *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '95)*, pp. 719-724, 1995.

[Bala et al. 1996] Bala, J., De Jong, K., Huang, J., Vafaie, H. y H. Wechsler. "Using learning to facilitate the evolution of features for recognizing visual concepts", *Evolutionary Computation 4(3)*, pp. 297-312, 1996.

[Barnett & Lewis 1994] Barnett, V. y T. Lewis. *Outliers in Statistical Data*, John Wiley and Sons, 1994.

[Beasley 1997] Beasley, D. "Possible applications of evolutionary computation", Cap. A1.2. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Beasley et al. 1993] Beasley, D., Bull, D. R. y R. R. Martin. "An Overview of Genetic Algorithms: Part 1, fundamentals" 1993. En: http://delta.cs.cinvestav.mx/~ccoello/bibgen.htm.

[Blickle 1997] Blickle, T. "Tournament Selection", Cap. C2.3. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Cardie 1995] Cardie, C. "Using decision trees to improve cased-based learning". En: *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1995.

[Carmona 2007] Carmona, E. "Algoritmos Evolutivos para Minería de Datos", *2nd Escuela del Observatorio Virtual Español: Minería de Datos*, España, 2007.

[Chen & Liu 1999] Chen, K. y H. Liu. "Towards an Evolutionary Algorithm: A Comparison of two Feature Selection Algorithms", *Proceedings of the Congress on Evolutionary Computation (CEC '99)*, pp. 1309-1313, Washington D.C., 1999.

[Chen et al. 1999] Chen, S., Guerra-Salcedo, C. y S.F. Smith. "Non-standard crossover for a standard representation – commonality-based feature subset selection", *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '99)*, pp. 129-134, Morgan Kaufmann, 1999.

[Cherkauer & Shavlik 1996] Cherkauer, K.J. y J.W. Shavlik. "Growing Simpler Decision Trees to facilitate knowledge discovery", *Proceedings of the 2nd International Conference on Knowledge Doscovery and Data Mining (KDD '96)*, pp. 315-318, AAAI Press, 1996.

[Chizi & Maimon 2005] Chizi, B. y O. Maimon. "Dimension Reduction and Feature Selection", Cap. 5, pp. 93 – 111. En: Maimon, O. y L. Rokach. *The Data Mining and Knowledge Discovery Handbook*, Springer, USA, 2005.

[Coleman 1986] Coleman, R. *Procesos Estocásticos: Selección de Problemas resueltos*, Limusa, México, 1986.

[Daniel 1999] Daniel, W.W. *Bioestadística: Base para el análisis de las ciencias de la salud*, Uteha Noriega Editores, 1999.

[Date 1986] Date, C.J. *Introducción a los Sistemas de Bases de Datos*, Addison-Wesley Iberoamericana, México, 1986.

[Davis 1991] Davis, L. (Ed.). *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

[De Jong et al. 1993] De Jong, K., Spears, W.M. y D.F. Gordon. "Using Genetic Algorithms for Concept Learning", *Machine Learning 13*, pp. 161-188, 1993.

[De Jong et al. 1997] De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[De Jong et al. 1997] De Jong K., Fogel D.B. y H.P. Schwefel. "A History of Evolutionary Computation", Cap. A2.3. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[De Jong 2000] De Jong, K. "Evolutionary Computation: An Unified Overview", *Genetic and Evolutionary Computation Conference Tutorial Program*, pp. 471-479, Las Vegas, N.V. USA, 2000.

[De Jong et al. 2000] De Jong, K., Fogel, D.B. y H.P. Schwefel. "A history of Evolutionary Computation". En: Back, T., Fogel, D.B. y T. Michalewicz (Eds.). *Evolutionary Computation 1: Basic Algorithms and Operators*, pp. 40-58, Institute of Physics Publishing, 2000.

[Deb 1997] Deb K. "Limitations of Evolutionary Computation Methods", Cap. B2.9. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Deb 1997a] Deb K. "Encoding and Decoding Functions", Cap. C4.2. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Deb 2000] Deb, K. "Introduction to Selection". En: Back, T., Fogel, D.B. y T. Michalewicz (Eds.). *Evolutionary Computation 1: Basic Algorithms and Operators*, pp. 166-171, Institute of Physics Publishing, 2000.

[Dhar et al. 2000] Dhar, V., Chou, D. y F. Povost. "Discovering interesting patterns for investment decision making with GLOWER – a genetic learner overlaid with entropy reduction", *Data Mining and Knowledge Discovery 4(4)*, pp. 251-280, 2000.

[Emmanouilidis et al. 2000] Emmanouilidis, C., Hunter, A. y J. MacIntyre, "A Multiobjetive Evolutionary setting for Feature Selection and a commonality-based crossover operator", *Proceedings of the 2000 Congress on Evolutionary Computation (CEC '2000)*, pp. 309-316, IEEE, 2000.

[Eshelman 1997] Eshelman, L.J. "Genetic Algorithms", Cap. B1.2. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Fayyad et al. 1996] Fayyad, U.M., Piatetsky-Shapiro, G. y P. Smyth. "The KDD Process for Extracting Useful Knowledge from Volumes of Data", *Communications of the ACM*, 39(11): 27-34, 1996.

[Fayyad et al. 1996a] Fayyad, U.M., Piatetsky-Shapiro, G. y P. Smyth. "From Data Mining to Knowledge Discovery: An Overview". En: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. y Uthurusamy R. (Eds.). *Advances in Knowledge Discovery and Data Mining,* pp. 1-34, AAAI/MIT Press 1996.

[Fidelis et al. 2000] Fidelis, M.V., Lopes, H.S. y A.A. Freitas. "Discovering comprehensible classification rules with a genetic algorithm", *Proceedings of the Congress on Evolutionary Computation – 2000 (CEC '2000)*, pp. 805-810, La Jolla CA, USA, IEEE, 2000.

[Flores 2011] Flores A.A. "Portafolios de Inversión Óptimos a través de Redes Neuronales Artificiales y Lógica Difusa", *Tesis de Maestría en Ciencias con Especialidad en Ingeniería de Sistemas*, ESIME-SEPI/IPN, 88 págs., México, 2011.

[Fogel & Fogel 1986] Fogel, L.J. y D.B. Fogel. *Artificial Intelligence through Evolutionary Programming*, U.S. Army Research Institute, Final Report Contract, PO-9-X56-1102C-1, 1986.

[Fogel et al. 1996] Fogel, L.J., Angeline, P.J. y T. Bäck, (eds). "Evolutionary Programming V", *Proc. 5th Ann. Conf. on Evolutionary Programming*, Cambridge, MA: MIT Press, 1996.

[Fogel 1997] Fogel, D.B. "Introduction", Cap. A1.1. En: De Jong K., Fogel L. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Fogel 1997a] Fogel, D.B. "Principles of evolutionary processes", Cap. A2.1. En: De Jong K., Fogel L. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Frank et al. 2005] Frank E., Hall M., Holmes G., Kirkby R., Pfahringer B., Witten I.H. y L. Trigg. "WEKA: A Machine Learning Workbeanch for Data Mining", Cap. 62, pp. 1305-1314. En: Maimon, O. y L. Rokach. *The Data Mining and Knowledge Discovery Handbook*, Springer, USA, 2005.

[Freitas 2002] Freitas, A.A. *Data Mining and Knowledge Discovery with Evolutionary Algorithms*, Natural Computing Series, Springer-Verlag, Berlin, 2002.

[Freitas 2005] Freitas, A.A. "Evolutionary Algorithms for Data Mining", Cap. 20, pp. 435-467. En: Maimon, O. y L. Rokach. *The Data Mining and Knowledge Discovery Handbook*, Springer, USA, 2005.

[Goldberg 1989] Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[Guerra-Salcedo & Whitley 1998] Guerra-Salcedo, C. y D. Whitley. "Genetic Search for feature subset selction: a comparison between CHC and GENESIS", *Genetic Programming 1998: Preceedings of the 3rd Annual Conference*, pp. 504-509, Morgan Kaufmann, 1998.

[Guerra-Salcedo et al. 1999] Guerra-Salcedo, C., Chen, S., Whitley, D. y S. Smith. "Fast and Accurate Feature Selection using Hybrid Genetic Strategies". En *Proceedings of the Congress on Evolutionary Computation (CEC '99)*, pp. 177-184, Washington D.C., USA, 1999.

[Guerra-Salcedo & Whitley 1999a] Guerra-Salcedo, C. y D. Whitley. "Genetic Approach to Feature Selection for Ensemble Creation". En *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '99)*, pp. 236-243, Morgan Kaufmann, 1999.

[Guerra-Salcedo & Whitley 1999b] Guerra-Salcedo, C. y D. Whitley. "Feature Selection Mechanisms for Ensemble Creation: A Genetic Perspective". En: Freitas, A.A. (Eds.), *Data Mining with Evolutionary Algorithms: Research Directions – Papers from the AAAI '99/GECCO '99 Workshop Technical Report WS-99-06*, pp. 13-17, AAAI Press, 1999.

[Hall 1999] Hall, M. *Correlation-based feature selection for machine learning*, Ph.D. Tesis, Department of Computer Science, University of Waikato, 1999.

[Hand 1997] Hand D.J. *Construction and Assessment of Classification Rules*, Wiley, 1997.

[Hekanaho 1996] Hekanaho, J. "Background Knowledge in GA-based Concept Learning", *Proceedings of the 13h International Conference on Machine Learning (1CML '96)*, pp. 234-242, 1996.

[Hernández-Orallo et al. 2007] Hernández-Orallo, J., Ramírez Quintana, M.J. y C. Ferri Ramírez. *Introducción a la Minería de Datos*, 656 págs., Pearson Prentice Hall, 2007.

[Higuchi & Manderick 1997] Higuchi T. y B. Manderick. "Hardware realizations of Evolutionary Algorithms", Cap. E2.3. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Hinterding 2000] Hinterding, R. "Representarion, Mutation and Crossover Issues in Evolutionary Computation", *Proceedings of the 2000 Congress on Evolutionary Computation (CEC '2000)*, pp. 916-923, IEEE, 2000.

[Holland 1975] Holland, J.H. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[Holmes & Nevill-Manning 1995] Holmes, G. y C.G. Nevill-Manning. "Feature Selection via the discovery of simple classification rules". En: *Proceedings of the Symposium on Intelligent Data Analysis*, Baden-Baden, Alemania, 1995.

[Holte 1993] Holte, R.C. *Very simple classification rules perform well on most commonly used datasets*, Machine Learning, 11:63-91, 1993.

[Hwang et al. 1994] Hwang, J., Lay, S. y A. Lippman. "Nonparametric multivariate density estimation: A comparative study", *IEEE Transaction on Signal Proccesing,* 42(10): 2795 – 2810, 1994.

[Ibaraki 1997] Ibaraki, T. "Comparison with existing Optimization Methods", Cap. D3.6. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Iglesia et al. 1996] Iglesia, B., Debuse, J.C.W. y V.J. Rayward-Smith. "Discovering Knowledge in Commercial Databases using Modern Heuristic Techniques", *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD '96)*, pp. 44-49, AAAI Press, 1996.

[Inmon 1992] Inmon, W.H. "EIS and the data warehouse: A simple approach to building an effective foundation for EIS", *Database Programming and Design, 5(11):70-73*, 1992.

[Ishibuchi & Nakashima 2000] Ishibuchi, H. y T. Nakashima. "Multi-objetive Pattern and Feature Selection by a Genetic Algorithm", *Proceedings of the 2000 Genetic and Evolutionary Computation Conference (GECCO '2000)*, pp. 1069-1076, Morgan Kaufmann, 2000.

[Janikow 1993] Janikow, C.Z. "A Knowledge-Intensive Genetic Algorithm for Supervised Learning", *Machine Learning 13*, pp. 189-228, 1993.

[Jaramillo 2003] Jaramillo López Judith, *Curso Básico de SQL*, Notas de Curso, FES Acatlán, UNAM, México, 2003.

[Jimenez & Landgrebe 1998] Jimenez, L.O. y D.A. Landgrebe, "Supervised Classification in High – Dimensional Space: Geometrical, Statistical and Asymptotical Propierties of Multivariate Data", *IEEE Transaction on Systems Man and Cybernetics - Part C*, Applications and Reviews, 28:39-54, 1998.

[John et al. 1994] John G.H., Kohavi R. y K. Pfleger. "Irrelevant features and the subset selection problem", *Proceedings of the Eleventh International Conference Machine Learning*, pp. 121-129, Morgan Kaufmann, 1994.

[Johnson 1990] Johnson, R. *Estadística Elemental*, Grupo Editorial Iberoamérica, 1990.

[Johnson 1999] Johnson, D.E. *Métodos Multivariados aplicados al análisis de datos*, Paraninfo, 1999.

[Jourdan et al. 2001] Jourdan L., Dhaenens C. y E. Talbi. "A Genetic Algorithm for Feature Selection in Data Mining for Genetics", *4th Metaheuristic International Conference (MIC '2001)*, pp. 29-33, Porto, Portugal, 2001.

[Kass 1975] Kass, G.V. "Significance testing in automatic interaction detection", *Applied Statistics 24(2):*178-189, 1975.

[Kim et al. 2002] Kim, Y., Street, N. y F. Menczer. *Feature Selection in Data Mining*, Universidad de Iowa, USA, 2002.

[Kira & Rendell 1992] Kira, K. y L.A. Rendell. "A practical approach to feature selection". En: *Machine Learning: Proceedings of the Ninth International Conference*, 1992.

[Knorr & Ng 1998] Knorr, E. y R. Ng. "Algorithms for mining distance-based outliers in large datasets". E*n: Proc. 1998 Intl. Conference on Very Large Data Bases (VLDB'98)*, pp. 392 – 403, 1998.

[KDnuggets™ 2007] *KDnuggets™: Data Mining Community's Top Resource*, 2007. http://www.kdnuggets.com/polls/2007/data_mining_methodology.htm

[Kohavi & John 1998] Kohavi, R. y G.H. John. "The Wrapper Approach". En: H. Liu y H. Motoda (Eds.). *Feature Extraction, Construction and Selection: A Data Mining Perspective*, pp. 33-50, Kluwer, 1998.

[Kohavi 1995] Kohavi, R., *Wrappers for Performance Enhancement and Oblivious Decision Graphs*, Tesis Doctoral, Universidad de Stanford, 1995.

[Koller & Sahami 1996] Koller, D. y M. Sahami. "Toward Optimal Feature Selection", *Proceedings of the Thirteenth International Conference Machine Learning*, Morgan Kaufmann, 1996.

[Korth & Silberschatz 1988] Korth, Henry F. y A. Silberschatz. *Fundamentos de Bases de Datos*, McGraw- Hill, México, 1988.

[Krzanowski 1988] Krzanowski, W.J. *Principles of Multivariate Analysis: A User´s Perspective*, Number 3 in Oxford Statistical Sciences Series, Oxford University Press, Oxford, 1988.

[Langley 1996] Langley, P. *Elements of Machine Learning*, Morgan Kaufmann, 1996.

[Langley & Sage 1994] Langley, P. y S. Sage. "Scaling to domains with irrelevant features". En: R. Greiner, Editor, *Computational Learning Theory and Natural Learning Systmes*, volume 4, MIT Press, 1994.

[Lavrac & Zupan 2005] Lavrac, N. y B. Zupan. "Data Mining in Medicine", Cap. 52, pp. 1107-1138. En: Maimon, O. y L. Rokach. *The Data Mining and Knowledge Discovery Handbook*, Springer, USA, 2005.

[Li & Vitányi 1997] Li, M. y P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*, Springer Verlag; 2nd. Edition, 1997.

[Liu & Motoda 1998] Liu, H. y H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer, 1998.

[Liu & Setiono 1996] Liu, H. y R. Setiono. "A probabilistic approach to feature selection: A filter solution". En: *Machine Learning: Proceedings of the Thirteenth International Conference on Machine Learning*, Morgan Kaufmann, 1996.

[Maimon & Last 2000] Maimon, O, y M. Last. *Knowledge Discovery and Data Mining – The Info-Fuzzy Network (INF) Methodology*, Kluwer, 2000.

[Maimon & Rokach 2005] Maimon, O. y L. Rokach. *The Data Mining and Knowledge Discovery Handbook*, Springer, USA, 2005.

[Maimon & Rokach 2005a] Maimon, O. y L. Rokach. "Introduction to Knowledge Discovery in Databases", Cap. 1, pp. 1-17. En: Maimon, O. y L. Rokach. *The Data Mining and Knowledge Discovery Handbook*, Springer, USA, 2005.

[Maimon & Rokach 2005b] Maimon, O. y L. Rokach. "Introduction to Supervised Methods", Cap. 8, pp. 149-164. En: Maimon, O. y L. Rokach. *The Data Mining and Knowledge Discovery Handbook*, Springer, USA, 2005.

[Mansilla et al. 1999] Mansilla, E.B., Mekaouche, A. y J.M.G. Guiu. "A Study of a Genetic Classifier System based on the Pittsburgh Approach on a Medical Domain", *Proceedings of the 12th International Conference Industrial and Engineering Applications of Artificial Intelligence and Exp. Syst. (IEA/AIE '99) – Lecture Notes in Artificial Intelligence 1611*, pp. 175-184, Springer, 1999.

[Martin-Bautista & Vila 1999] Martin-Bautista, M.J. y M.A. Vila. "A Survey of Genetic Feature Selection in Mining Issues", *Proceedings of the Congress on Evolutionary Computation (CEC '99)*, pp. 1314-1321, IEEE, 1999.

[McCullagh & Nelder 1989] McCullagh, P. y J.A. Nelder. "Generalized Linear Models", number 37. En: *Monographs on Statistics and Applied Probability*, Chapman and Hall, London, 2nd. Edition, 1989.

[Mendenhall et al. 1986] Mendenhall W., Scheaffer R.L. y D.D. Wackerly. *Estadística Matemática con Aplicaciones*, Grupo Editorial Iberoamérica, México, 1986.

[Miller & Freund 1987] Miller, I. y J.E. Freund. *Probabilidad y Estadística para Ingenieros*, Prentice Hall, México, 1987.

[Mitchell 1996] Mitchell, T.M. *An Introduction to Genetic Algorithms*, MIT Press, 1996.

[Mitchell 1997] Mitchell, T.M. *Machine Learning*, McGraw-Hill, 1997.

[Mladenic et al. 2003] Mladenic D., Lavrac N., Bohanec M. y S. Moyle. *Data Mining and Decision Support: Integration and Collaboration*, Kluwer Academic Publishers, 2003.

[Moore & Lee 1994] Moore, A.W. y M.S. Lee. "Efficient algorithms for minimizing cross validation error". En: *Machine Learning: Proceedings of the Eleventh International Conference*, Morgan Kaufmann, 1994.

[Moore et al. 1992] Moore, A.W., Hill, D.H. y M.P. Johnson. "An empirical investigation of brute force to choose feature, smoothers and function approximations". En: Hanson, S. et al., editors, *Computational Learning Theory and Natural Learning Systems*, volumen 3, MIT Press, 1992.

[Moser & Murty 2000] Moser, A. y M.N. Murty. "On the scalability of genetic algorithms to very large-scale feature selection", *Proceedings of the Real-World Applocations of Evolutionary Computing (EvoWorkshops 2000). Lecture Notes in Computer Science 1803*, pp.77-86, Springer, 2000.

[Netbeans 2012] Netbeans, 2012. www.netbeans.com

[Ng et al. 1998] Ng, R., Lakshmanana, L.V.S., Han, J. y A. Pang. "Exploratory Mining and Pruning Optimizations of Constrained Asociation Rules", *Proc. ACM SIGMOD Int'l Conf. Management of Data*, ACM Press, pp. 13-24, New York, 1998.

[Noda et al. 1999] Noda E., Freitas, A.A. y H.S. Lopes. "Discovering interesting prediction rules with a genetic algorithm", *Proceedings of the Congress on Evolutionary Computation – 1999 (CEC '99)*, pp. 1322-1329. Washington D.C., USA, 1999.

[Núñez 1988] Núñez, M. "Economic Induction: A case study". En: *Proceedings of the 3rd. European Working Session on Learning, EWSL-98*, volumen 55, pp. 139-145, Morgan Kaufmann, 1988.

[Ochoa & Sandra 2008] Ochoa G., Sandra I., *El Modelo de Markowitz en la Teoría del Portafolios de Inversión,* Tesis de Maestría en Ciencias en Administración, UPIICSA, IPN, México D.F., 2008.

[Pappa & Freitas 2010] Pappa, G.L. y A.A. Freitas. *Automating the Design of Data Mining Algorithms: An Evolutionary Computation Approach*, Springer, Berlín, Alemania, 2010.

[Pazzani 1995] Pazzani, M. "Searching for dependencies in Bayesian Classifiers". En: *Proceedings of the Fifth International Workshop on AI and Statistics*, 1995.

[Pei et al. 1997] Pei, M., Goodman, E.D. y W.F. Punch. "Pattern Discovery from Data using Genetic Algorithms", *Proceedings of the 1st Pacific Asia Conference on Knowledge Discovery and Data Mining*, Feb. 1997.

[Peña & Prieto 2001] Peña, D. y F.J. Prieto. "Robust covariance matrix estimation and multivariate outlier detection", *Technometrics 3*, pp. 286-310, 2001.

[Peña 2002] Peña, D. *Análisis de datos multivariantes*, McGraw Hill, 2002.

[Pfahringer 1995] Pfahringer, B. "Compression-based feature subset selection". En: *Proceedings of the IJCAI-95 Workshop on Data Engineering for Inductive Learning*, pp. 109-119, 1995.

[Porto 1997] Porto, V.W. "Evolutionary Programming", Cap. B1.4. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Pyle 1999] Pyle, D. *Data Preparation for Data Mining*, Morgan Kaufmann, pp. 540, U.S.A., 1999.

[Quinlan 1986] Quinlan, J.R. *Induction of Decision Trees*, Machine Learning, 1:81-106, 1986.

[Radcliffe 1997] Radcliffe, N.J. "Introduction: Theoretical Foundations and Properties of Evolutionary Computations", Cap. B2.1 De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Ray 1991] Ray, T. "An Approach to the synthesis of life", *Artificial Life II,* Ed. C.G. Langton, C. Taylor, J.D. Farmer and S. Rasmussen (Reading, MA: Addison-Wesley), pp. 371-408, 1991.

[RAE 2012] Real Academia Española, Diccionario, 2012. http://www.rae.es/rae.html

[Rechenberg 1965] Rechenberg, I. *Cybernetic Solution Path of an Experimental Problem,* Royal Aircraft Establishment Library, Translation, 1122, 1965.

[Renche 2002] Renche, A.C. *Methods of Multivariate Analysis*, 2nd. Edition, Wiley-Interscience, 2002.

[Rissanen 1978] Rissanen, J. "Modeling by shortest data description", *Automatica*, vol. 14, pp. 465-471, 1978.

[Rodríguez-Vázquez et al. 2007] Rodríguez-Vázquez K., Meneses Matilde A., Torres Cruz C.E. y K. Castro Carrera, *Introducción a los Algoritmos Genéticos. Implementación en C++, Java y MATLAB*, Notas de Curso, IIMAS-UNAM, México, 2007.

[Ross et al. 2000] Ross, S.A., Westerfield, R.W. y J.F. Jaffe. *Finanzas Corporativas*, McGraew Hill, 5° edición, México, 1049, 2000.

[Rousseeuw & Leroy 2003] Rousseeuw, P.J. y A.M. Leroy. *Robust Regression and Outlier Detection*, John Wiley and Sons, New York, 1987. 2nd. Edition, Wiley-Interscience, 2003.

[Royden 1968] Royden, H. L. *Real Analysis*, Mac Millan Publishing Co. INC, 2nd. Ed., New York, 1968.

[Rudolph 1997] Rudolph, G. "Evolution Strategies", Cap. B1.3. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Rudolph & Ziegenhirt 1997] Rudolph, G. y J. Ziegenhirt. "Computation Time of Evolutionary Operators", Cap. E2.2. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Salton & McGill 1983] Salton, G. y M.J. McGill. *Introduction to modern information retrieval*, McGraw Hill, New York, 1983.

[Schnier & Yao 2000] Schnier, T. y X. Yao. "Using multiple representarions in Evolutionary Algorithms", *Proceedings of the 2000 Congress on Evolutionary Computation (CEC '2000)*, pp. 479-486, IEEE, 2000.

[Schwefel 1997] Schwefel, H.P. "Advantages (and Disadvantages) of Evolutionary Computation over other Approaches", Cap. A1.3. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997.

[Setiono & Liu 1996] Setiono, R y H. Liu. "Chi2: Feature Selection and Discretization of numeric attributes". En: *Proceedings of the Seventh IEEE International Conference on Tools of Artificial Intelligence*, 1995.

[Shannon & Johannes 1976] Shannon, R. y J.D. Johannes. "Systems simulation: the art and science", *IEEE Transactions on Systems, Man and Cybernetics*, 6(10), pp. 723-724.

[Sharpe & Glover 1999] Sharpe, P.K. y R.P. Glover. "Efficient GA based Techniques for Classification", *Applied Intelligence 11*, pp. 277-284, 1999.

[Singh & Provan 1996] Singh, M. y G.M. Provan. "Efficien learning of selective Bayesian classifiers". En: *Machine Learning: Proceedings of the Thirteeth International network Conference on Machine Learning*, Morgan Kaufmann, 1996.

[Sun et al. 2008] Sun, Y., Todorovic, S. y S. Goodinson. *A Feature Selection Algorithm Capable of Handing Extremely Large Data Dimensionality*, SIAM, Illinois, USA, 2008.

[Terano & Ishino 1998] Terano, T. y Y. Ishino. "Interactive Genetic Algorithm based Feature Selection and its Application to Marketing Data Analysis". En: Liu, H. y H. Motoda (Eds.). *Feature Extraction,Construction and Selection*, pp. 393-406, Kluwer, 1998.

[Tomlin 1990] Tomlin, C.D. *Geographic Information Systems and Cartographic Modeling*, Prentice Hall, 1990.

[Turney 2000] Turney, P. "Types of Cost in Inductive Concept Learning", *Proceedings Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning* (WCSL at ICML-2000), pp. 15-21, 2000.

[Vafaie & De Jong 1995] Vafaie, H. y K. De Jong. "Genetic algorithms as a tool for restructuring feature space representations". En: *Proceedings of the International Conference on Tools with AI*, IEEE Computer Society Press, 1995.

[Vafaie & De Jong 1998] Vafaie, H. y K. De Jong. "Evolutionary Feature Space Transformation". En: Liu, H. y H. Motoda (Eds.). *Feature Extraction, Construction and Selection*, pp. 307-323, Kluwer, 1998.

[Venturini 1993] Venturini, G. "SIA: A Supervised Inductive Algorithm with Genetic Search for Learning Attributes based Concepts", *Machine Learning: Proceedings of the 1993 European Conference (ECML '93) – Lecture Notes in Artificial Intelligence 667*, pp. 280-296, Springer, 1993.

[Weiss & Kulikowski 1991] Weiss, S.M. y C.A. Kulikowski. *Computer Systems that Learn*, Morgan Kaufmann, San Mateo, 1991.

[WEKA-Online 2005] WEKA: Online Documentation, Universidad de Waikato, 2005. http://www.cs.waikato.ac.nz/ml/weka/documentation.html

[Whitley 1997] Whitley, D. "Permutations", Cap. C1.4. En: De Jong K., Fogel D.B. y H.P. Schwefel. *Handbook of Evolutionary Computation*, Publishing Ltd and Oxford University Press, 1997

[Wikipedia 2012] Wikipedia, 2012. http://en.wikipedia.org/wiki/Main_Page

[Witten & Frank 2000] Witten I.H. y E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, 2000.

[Witten et al. 2011] Witten I.H., Frank E. y M.A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*, 3ra. Ed., Morgan Kaufmann, USA, 2011.

[Wolberg et al. 1995] Wolberg, Dr. W.H., Nick Street, W. y O.L. Mangasarian. *Breast Cancer Wisconsin Diagnostic Data Set - WDBC*, Universidad de Wisconsin, Noviembre, 1995. ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/cancer/WDBC/

[Wolberg et al. 1995a] Wolberg, Dr. W.H., Nick Street, W. y O.L. Mangasarian. *Breast Cancer Wisconsin Diagnostic Data Set - WPBC*, Universidad de Wisconsin, Diciembre, 1995. ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/cancer/WPBC/

[Yang & Honavar 1997] Yang, J. y V. Honavar. "Feature Subset Selection using a Genetic Algorithm", *Genetic Programming 1997: Proceedings of the 2nd Annual Conference (GP '97)*, pp. 380-385, Morgan Kaufmann, 1997.

[Yahoo Finanzas 2012] Yahoo Finanzas: http://mx.finanzas.yahoo.com/indices

# ANEXO A.

## A. CÓDIGO FUENTE DEL SISTEMA "ATTRIBUTE_CLASSIFICATION".

El programa llamado "Attribut_Classification", es un algoritmo genético para la selección de atributos programado en java, sobre la plataforma IDE de NetBeans, y que contiene 7 clases básicas. A continuación se describen éstas clases:

*Package: Attribute_Classification*

Clases:

1. *CPrincipal.java*
2. *CConfig.java*
3. *CProblema.java*
4. *CGeneracion.java*
5. *CIndividuo.java*
6. *CDatabase.java*
7. *CRegistro.java*
8. Config.txt (Archivo de Texto)

**********

### *A.1 CPrincipal.java*

Es la clase principal que llama a iniciar el programa. Lee la base de datos, guarda los mejores individuos de cada ejecución, imprime las estadísticas de éstas y extrae el porcentaje de datos determinado para el conjunto de entrenamiento. Crea un objeto de las clases CConfig y 'CProblema' para llamar a otras funciones, y así iniciar el proceso para ejecutar el algoritmo genético y encontrar al individuo óptimo. También se encarga de clasificar al mejor individuo en el conjunto de Prueba.

```
package tesis;
import java.io.*;
import java.util.*;
/**
 * @author Zure
 * Se presenta un trabajo de Tesis de Licenciatura llamado "Attribute_Classification", basado en el desarrollo
 * de un Algoritmo Genetico para la Seleccion de Caracteristicas de una BD.
 * En esta clase se genera una nueva instancia d la clase CProblema, se encuentra una solucion y se imprimen
 * los resultados.
 */

public class CPrincipal {
   private static CConfig config;
   public static void main(String[] args) throws IOException
   {  config = new CConfig("Data/config.txt");
      // Aqui se lee la BD
      CDatabase db = new CDatabase(config);
      if (!db.Leer())
      {  System.out.println("No se puede leer la Base de Datos");
         return;
      }
      boolean guardar = config.Obten_GuardarFile();
```

```
      String resultFile = config.Obten_ResultFile();
      int porcentEntrenam = config.Obten_PorcentEntrenam();
      PrintWriter writer = guardar ?
         new PrintWriter(new FileWriter(resultFile)) : new PrintWriter(System.out);
       PrintWriter resumen = config.Obten_GuardarResumen() ?
          new PrintWriter(new FileWriter(config.Obten_ResumenFile())) :
          new PrintWriter(new StringWriter(65536));
      db.Muestreo(porcentEntrenam);
      ArrayList<CIndividuo> listaMejores = Obten_Mejor(db, writer, resumen);
      db.Limpiar();
      imprime_EstadsAtrib(listaMejores, writer, resumen);
      ArrayList<CIndividuo> lista_LosMejores = Obten_Mejores(listaMejores, false);
      imprimeElMejor(lista_LosMejores, writer, resumen, false);
      db.Leer();
      db.Muestreo(-config.Obten_PorcentPrueba());
      calificaElMejor(db, lista_LosMejores, writer, resumen);
      writer.close();
       if (resumen != null) resumen.close();
      db.Limpiar();
   }

   //Se obtienen los mejores individuos de cada 'corrida'
   private static ArrayList<CIndividuo> Obten_Mejor(CDatabase db, PrintWriter writer, PrintWriter
resumen) throws IOException
   {  int eval_cont = config.Obten_EvalCont();
      ArrayList<CIndividuo> listaMejores = new ArrayList<CIndividuo>();
      for (int i = 0; i < eval_cont; i++)
      {  CProblema proceso = new CProblema(db);
         proceso.EncontrarSolucion();
         CIndividuo mejor = proceso.Solucion == null ? proceso.Actual.Mejor : proceso.Solucion;
         mejor.Desconectar();
         Iterator<CIndividuo> iterac = listaMejores.iterator();
         boolean duplicado = false;
         while (iterac.hasNext())
         {  CIndividuo otro = iterac.next();
            duplicado = mejor.IgualQue(otro);
            if (duplicado)
            {  otro.Encontrado++;
               break;
            }
         }
         if (!duplicado) listaMejores.add(mejor);
         proceso.ImprimirResult(i+1, writer, resumen);
      }
      return listaMejores;
   }

   // Imprime las estadisticas del AG ejecutado en el conjunto de entrenamiento
   private static void imprime_EstadsAtrib(ArrayList<CIndividuo> listMejor, PrintWriter writer, PrintWriter
resumen)
   {  Iterator<CIndividuo> iterac = listMejor.iterator();
      // Estadisticas de los atributos - contador total para cada atributo encontrado
      TreeMap<Integer, Integer> estadisticas = new TreeMap<Integer, Integer>();
      while (iterac.hasNext())
      {  CIndividuo actual = iterac.next();
         int[] datos = actual.Datos;
```

```
            for (int i = 0; i < datos.length; i++)
            {  Integer clave = new Integer(datos[i]);
               Integer valor;
               if (estadisticas.containsKey(clave))
               {  valor = estadisticas.get(clave);
                  valor += actual.Encontrado;
               }
               else {
                  valor = new Integer(actual.Encontrado);
               }
               estadisticas.put(clave, valor);
            }
         }
         // Imprimiendo las estadisticas de los atributos
         Iterator<Integer> Ids = estadisticas.keySet().iterator();
         writer.println();
         writer.println("Estadisticas de los Atributos");
          resumen.println();
          resumen.println("Estadisticas de los Atributos");
         while (Ids.hasNext())
         {  Integer clave = Ids.next();
            Integer valor = estadisticas.get(clave);
            writer.println("    Atributo # " + clave + " ha sido encontrado " + valor + " veces");
              resumen.println("Atributo #," + clave + ",frecuencia," + valor);
         }
      }

      // Obtiene los mejores individuos para clasificarlos en el conjunto de Prueba
      private static ArrayList<CIndividuo> Obten_Mejores(ArrayList<CIndividuo> listaMejores, boolean
ctoPrueba)
      {   //Aqui encontramos la TP mayor
         Iterator<CIndividuo> iterac = listaMejores.iterator();
         double maxTP = 0;
         while (iterac.hasNext())
         {  double tp = ctoPrueba ? iterac.next().Obten_TP2() : iterac.next().Obten_TP();
            if (tp > maxTP)
            {  maxTP = tp;
            }
         }
         // Aqui obtenemos a todos los individuos con la TP mas alta
         ArrayList<CIndividuo> listaElMejor = new ArrayList<CIndividuo>();
         iterac = listaMejores.iterator();
         while (iterac.hasNext())
         {  CIndividuo actual = iterac.next();
            double tp = ctoPrueba ? actual.Obten_TP2() : actual.Obten_TP();
            if (tp == maxTP)
            {  listaElMejor.add(actual);
            }
         }   return listaElMejor;
      }

      // Aqui se imprime en pantalla el mejor individuo despues de ser calificado en ambos conjuntos
      private static void imprimeElMejor(ArrayList<CIndividuo> listaElMejor, PrintWriter writer, PrintWriter
resumen, boolean analizado)
      {  writer.println();
         int cont = listaElMejor.size();
```

```
      String texto = cont == 1 ? " Mejor individuo " : " Mejores individuos ";
       writer.print("Encontrado " + cont + texto + "con la Tasa de Precision = " +
listaElMejor.get(0).Obten_TP());
      if (analizado)
      {  writer.print(" (despues de ser analizado en el conjunto de prueba)");
      }
      writer.println();
      Iterator<CIndividuo> iterac = listaElMejor.iterator();
      while (iterac.hasNext())
      {   CIndividuo next = iterac.next();
          writer.println("    " + next.Texto_Completo());
      }
   }

   // Se califica al mejor individuo en el conjunto de Prueba
   private static void calificaElMejor(CDatabase db, ArrayList<CIndividuo> lista_Mejores, PrintWriter
writer, PrintWriter resumen)
   {   writer.println();
      Iterator<CIndividuo> iterac = lista_Mejores.iterator();
      // realizando la etapa de prueba y evaluando los datos de todos los individuos
      while (iterac.hasNext())
      {
         CIndividuo actual = iterac.next();
         writer.println("Probando el mejor individuo:");
         writer.println(actual.Texto_Completo());
         actual.califica = true;
         ArrayList<String> lista = db.Clasificar(actual);
         writer.println("A continuacion, se muestran los resultados de las pruebas por registro...");
         Iterator<String> IteradorCadena = lista.iterator();
         while (IteradorCadena.hasNext())
         {  writer.println("    " + IteradorCadena.next());                    }
         writer.println("Despues de la Prueba, estos son los resultados de su nueva evaluacion:");
         writer.println(actual.Texto_Completo2());
          resumen.println("Despues de la Prueba, estos son los resultados de su nueva evaluacion:");
          int error = actual.FN2 + actual.FP2;
          double falla = 100.0 * error / actual.Obten_Total(true);
          resumen.println("Errores," + error + ",% de falla," + falla);
          resumen.println(actual.Texto_CSV2());
         writer.println();
      }
      // Encuentra al mejor nuevo individuo
      lista_Mejores = Obten_Mejores(lista_Mejores, true);
      imprimeElMejor(lista_Mejores, writer, resumen, true);
   }

   // Crea un nuevo objeto de la clase CConfig, que configurar los parametros para iniciar el AG
   public static CConfig Configurar()
   {   return config;                                    }
}
```

**********

## *A.2 CConfig.java*

Lee, línea por línea, el archivo de configuración ‘config.txt’ para establecer los parámetros iniciales del AG. Inicializa las variables, evalúa que sea una BD adecuada y determina las condiciones del ciclo.

```
package tesis;
import java.io.*;
import java.util.*;
/**
 * * @author Zure
 * Este programa establece la configuracion de los parametros del programa a traves de un archivo de texto
 */

public class CConfig {
   public enum LectorBD { Default };
   private double Probab_Mutacion;
   private double Probab_Cruza;
   private int Tamanio_Poblacion;
   private int Tamanio_Individuo;
   private int Cont_Max_Generacion;
   private String CadenaCorrecta;
   private String CadenaFalsa;
   private String DatabasePath;
   private int ContadorColumna;
   private int ColumnaClase;
   private ArrayList<Integer> saltaColumna = new ArrayList<Integer>();
   private LectorBD lector;
   private boolean guardarFile;
   private String resultFile;
   private String resumenFile;
   private int porcent_Entrenam;
   private int muestra;
   private int eval_cont;
   private int modoCorrelacion;

   // Lee todos los parametros de configuracion
   public CConfig(String path)
   {  boolean valido = true;
      estableceDefaults();
      try
      {   FileReader lector = new FileReader(path);
         BufferedReader lector_b = new BufferedReader(lector);
         String linea;
         try
         {   while ((linea = lector_b.readLine()) != null)
            {   analizaLinea(linea);                              }
            lector_b.close();
            lector.close();
         }
         catch (IOException x)
         {   valido = false;                                       }
      }
      catch (FileNotFoundException x)
      {   valido = false;                                          }
      if (!valido) estableceDefaults();
      Config = this;
   }

   // Analiza las lineas para ver si son las correctas a ser cargadas
   private boolean analizaLinea(String linea)
   {   //Esta func. responde 'true' si la linea es valida,aunque este vacia y 'false' si contiene datos incorrectos
```

```
if (linea == null) return true;
linea = linea.trim();
if (linea.length() == 0) return true;
int comentarioPos = linea.indexOf(';');
if (comentarioPos == 0) return true;
// Si la linea es un comentario, la salta
if (comentarioPos > 0)
{  linea = linea.substring(0, comentarioPos);
   linea = linea.trim();
   if (linea.length() == 0) return true;
}
int posicion = linea.indexOf('=');
if (posicion <= 0) return false;
if (posicion == linea.length() - 1) return false;
String nombre = linea.substring(0, posicion);
String valor = linea.substring(posicion + 1);
nombre = nombre.trim();
valor = valor.trim();
if (nombre.length() == 0 || valor.length() == 0) return false;
boolean resultado = true;
if (nombre.equalsIgnoreCase("Database")) DatabasePath = valor;
else if (nombre.equalsIgnoreCase("Cont_atrib"))
{  try
   {   ContadorColumna = Integer.parseInt(valor);          }
   catch (NumberFormatException ex)
   {   resultado = false;                                  }
}
else if (nombre.equalsIgnoreCase("Salida_positiva"))
{   CadenaCorrecta = valor;                                }
else if (nombre.equalsIgnoreCase("Salida_negativa"))
{   CadenaFalsa = valor;                                   }
else if (nombre.equalsIgnoreCase("Probabilidad_Mutacion"))
{  try
   {   Probab_Mutacion = Double.parseDouble(valor);    }
   catch (NumberFormatException ex)
   {   resultado = false;                                  }
}
else if (nombre.equalsIgnoreCase("Probabilidad_Cruza"))
{  try
   {   Probab_Cruza = Double.parseDouble(valor);          }
   catch (NumberFormatException ex)
   {   resultado = false;                                  }
}
else if (nombre.equalsIgnoreCase("Tamanio_Poblacion"))
{  try
   {   Tamanio_Poblacion = Integer.parseInt(valor);       }
   catch (NumberFormatException ex)
   {   resultado = false;                                  }
}
else if (nombre.equalsIgnoreCase("Individuo_tamanio"))
{  try
   {   Tamanio_Individuo = Integer.parseInt(valor);       }
   catch (NumberFormatException ex)
   {   resultado = false;                                  }
}
else if (nombre.equalsIgnoreCase("Cont_Max_Generacion"))
```

```
{  try
   {   Cont_Max_Generacion = Integer.parseInt(valor);   }
   catch (NumberFormatException ex)
   {   resultado = false;                               }
}
else if (nombre.equalsIgnoreCase("Guardar_File"))
{   guardarFile = "1".equals(valor);                    }
else if (nombre.equalsIgnoreCase("File_Resultados"))
{   resultFile = valor;                                 }
 else if (nombre.equalsIgnoreCase("File_Resumen"))
 {    resumenFile = valor;                                    }
else if (nombre.equalsIgnoreCase("Omitir_columnas"))
{
   String[] indices = valor.split(",");
   int sz = indices.length;
   for (int i = 0; i < sz; i++)
   {  try
      {   saltaColumna.add(new Integer(Integer.parseInt(indices[i])));        }
      catch (NumberFormatException ex)
      {   resultado = false;                            }
   }
}
else if (nombre.equalsIgnoreCase("Lugar_colum_outcome"))
{  try
   {   ColumnaClase = Integer.parseInt(valor);          }
   catch (NumberFormatException ex)
   {   resultado = false;                               }
}
else if (nombre.equalsIgnoreCase("Lector_db"))
{
   if (valor.equalsIgnoreCase("Default")) lector = LectorBD.Default;
   else resultado = false;
}
else if (nombre.equalsIgnoreCase("Porcent_entrenam"))
{  try
   {   porcent_Entrenam = Integer.parseInt(valor);      }
   catch (NumberFormatException ex)
   {   resultado = false;                               }
}
else if (nombre.equalsIgnoreCase("Muestra"))
{  try
   {   muestra = Integer.parseInt(valor);               }
   catch (NumberFormatException ex)
   {   resultado = false;                               }
}
else if (nombre.equalsIgnoreCase("Eval_cont"))
{  try
   {   eval_cont = Integer.parseInt(valor);             }
   catch (NumberFormatException ex)
   {   resultado = false;                               }
}
else if (nombre.equalsIgnoreCase("correlacion_modo"))
{  try
   {  modoCorrelacion = Integer.parseInt(valor);        }
   catch (NumberFormatException ex)
   {  resultado = false;                                }
```

```
    }   return resultado;
}

// Establece los valores por default desde el codigo
private void estableceDefaults()
{  muestra = 3;
   porcent_Entrenam = 71;
   Probab_Mutacion = 0.15;
   Probab_Cruza = 0.95;
   Tamanio_Poblacion = 200;
   Tamanio_Individuo = 6;
   Cont_Max_Generacion = 1000;
   CadenaCorrecta = "1";
   CadenaFalsa = "0";
   guardarFile = false;
   resultFile = "Data/results.txt";
   eval_cont = 1;
   modoCorrelacion = 1;
   DatabasePath = "";
   ContadorColumna = 30;
   lector = LectorBD.Default;
   ColumnaClase = -1;
   saltaColumna.clear();
}

// Se obtienen los parametros desde Config.txt, para las sig. funciones:
public double ObtenProbab_Mutacion()
{   return Probab_Mutacion;                }

public double ObtenProbab_Cruza()
{   return Probab_Cruza;                   }

public int ObtenTamanio_Poblacion()
{   return Tamanio_Poblacion;              }

public int ObtenIndividuo_Tamanio()
{   return Tamanio_Individuo;              }

public int ObtenCont_Max_Generacion()
{   return Cont_Max_Generacion;            }

public String ObtenCaden_Verdadera()
{   return CadenaCorrecta;                 }

public String ObtenCadena_Negativa()
{   return CadenaFalsa;                    }

public String Obten_DatabasePath()
{   return DatabasePath;                   }

public int Obten_ContColumna()
{   return ContadorColumna;                }

public boolean Obten_GuardarFile()
{   return guardarFile;                    }
```

```
   public String Obten_ResultFile()
   {   return resultFile;                          }

   public String Obten_ResumenFile()
    {   return resumenFile;                         }

   public boolean Obten_GuardarResumen()
   {     return resumenFile != null && !resumenFile.isEmpty();      }

   public LectorBD Obten_LectorDB()
   {   return lector;                              }

   public int Obten_ColumnaClase()
   {   return ColumnaClase;                        }

   public ArrayList<Integer> Obten_ColumIgnoradas()
   {  ArrayList<Integer> lista = new ArrayList<Integer>();
      lista.add(new Integer(ColumnaClase));
      lista.addAll(saltaColumna);
      return lista;
   }
   public int Obten_PorcentEntrenam()
   {   return porcent_Entrenam;                    }

   public int Obten_PorcentPrueba()
   {   return 100 - porcent_Entrenam;              }

   public int Obten_Muestra()
   {   return muestra;                             }

   public int Obten_EvalCont()
   {   return eval_cont;                           }

   public int ObtenCorrelacionModo()
   {   return modoCorrelacion;                     }
}

******************
```

## *A.3 CProblema.java*

Crea la primera generación e inicia el algoritmo genético para encontrar la solución, con la función Iniciar_proc(). Además, contiene las siguientes funciones: Iteracion(), EncontrarSolucion() e ImprimirResult(). La función de Iteracion() permite que el programa realice las iteraciones necesarias para encontrar la solución o para abortar el programa si se ha llegado al límite de iteraciones. EncontrarSolucion() está ligado a la clase CIndividuo, que es la que reporta el mejor individuo del AG; ImprimirResult(), imprime los resultados en pantalla y GuardarResult() guarda los resultados en los archivos '*result.txt*' y *'resumen.csv'*.

```
package tesis;
import java.io.*;
import java.util.*;
/**
 * @author Zure
 * Esta parte del programa maneja todos los aspectos generales del AG.
 */
```

```
public class CProblema {
   public final double Probab_Mutacion;
   public final double Probab_Cruza;
   public final int Tamanio_Poblacion;
   public final int Individuo_tamanio;
   public final int Cont_Max_Generacion;
   public final int min_valor = 1;
   public int max_valor;
   public final int Porcent_Entrenam;
   public final int Porcent_Prueba;
   public CGeneracion Primera = null;
   public CGeneracion Actual = null;
   public CIndividuo Solucion = null;
   private CDatabase basededatos;

   // Obtiene los parametros necesarios del archivo: Config.txt
   public CProblema(CDatabase DB)
   {  basededatos = DB;
      max_valor = DB.min_num_columnas;
      CConfig config = CPrincipal.Configurar();
      Probab_Mutacion = config.ObtenProbab_Mutacion();
      Probab_Cruza = config.ObtenProbab_Cruza();
      Tamanio_Poblacion = config.ObtenTamanio_Poblacion();
      Individuo_tamanio = config.ObtenIndividuo_Tamanio();
      Cont_Max_Generacion = config.ObtenCont_Max_Generacion();
      Porcent_Entrenam = config.Obten_PorcentEntrenam();
      Porcent_Prueba = config.Obten_PorcentPrueba();
   }

   // Crea la primera generacion con todos sus parametros
   public boolean Iniciar_proc()
   {  CGeneracion.Probabilidad_Mutacion = Probab_Mutacion;
      CGeneracion.Probabilidad_Cruza = Probab_Cruza;
      Primera = new CGeneracion(Tamanio_Poblacion, Individuo_tamanio, min_valor, max_valor,
basededatos);
      Actual = Primera;
      Primera.Poblar();
      Primera.Evaluar();
      if (Primera.Mejor.Evaluacion == 0) Solucion = Primera.Mejor;
      return Solucion != null;
   }

   // Determina las iteraciones del AG
   public boolean Iteracion()
   {  Actual = Actual.Generacion_siguiente();
      Actual.Evaluar();
      if (Actual.Mejor.Evaluacion == 0) Solucion = Actual.Mejor;
      Actual.Antecesores.Eliminar();
      return Solucion != null;
   }

   // Obtiene la mejor solucion y la devuelve.
   public CIndividuo EncontrarSolucion()
   {  boolean encontrado = Iniciar_proc();
      while (!encontrado && Actual.Cont < Cont_Max_Generacion)
         encontrado = Iteracion();
```

```
        return Solucion;
    }

    // Imprime el resultado de la mejor solucion
    public void ImprimirResult(int iteracion, PrintWriter imprimir, PrintWriter resumen) throws IOException
    {  if (iteracion == 1)
        {  imprimir.println("Problema de Seleccion de Atributos de una BD");
           imprimir.println(
              "Se hace una seleccion de los " + CPrincipal.Configurar().ObtenIndividuo_Tamanio() +
              " atributos mas representativos de una BD con " + CPrincipal.Configurar().Obten_ContColumna() +
              " atributos");
        }
        imprimir.println("Iteracion # " + iteracion);
        resumen.println("Iteracion # " + iteracion);

        if (Solucion != null)
        {  imprimir.println("La Solucion es: " + Solucion.Texto_Completo());
           imprimir.println("Pertenece a la generacion # " + Solucion.Obten_GenCont());
           resumen.println("La Solucion es," + Solucion.Texto_CSV());
           resumen.println("Pertenece a la generacion # " + Solucion.Obten_GenCont());
        }
        else {
           imprimir.println("No se encontro solucion al problema");
           imprimir.println("El mejor resultado encontrado es: " + Actual.Mejor.Texto_Completo());
           imprimir.println("Pertenece a la generacion # " + Actual.Mejor.Obten_GenCont());
           resumen.println("El mejor resultado encontrado es," + Actual.Mejor.Texto_CSV());
           resumen.println("Pertenece a la generacion # " + Actual.Mejor.Obten_GenCont());
        }
        imprimir.flush();
    }
}

***************
```

## *A.4 CGeneracion.java*

Aquí se encuentran todos los procedimientos propios de cada generación dentro de un Algoritmo Genético, como son: Las funciones de Poblar(), Evaluar(), Seleccionar(), Cruzar(), Mutar(), Eliminar() y la acción de obtener la información necesaria de la generación anterior, así como, pasar a la siguiente generación. En la función Poblar() se crean ***n*** nuevos individuos de la clase CIndividuo, para luego evaluarlos en Evaluar(), a través de la tasa de precisión más una calificación acorde al coeficiente de correlación de cada atributo de un individuo. Además, a través de la aptitud relativa se proporcionan las probabilidades de selección que la clase Seleccionar() utilizará, y en donde se aplica elitismo, para después seleccionar el resto de los individuos a través del Método de la Ruleta; después, se aplican los operadores de Cruzar() y Mutar() a la generación actual; y por último, se limpia la generación actual con la función Eliminar(), para poder pasar a una nueva generación.

```
package tesis;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Random;
import java.util.Set;
import java.util.TreeMap;
/**
 * @author Zure
 * Esta clase contine todos los operadores de AG a nivel Generacion
 */
```

```
public class CGeneracion {
   private static int Cont_generacion = 0;
   public static double Probabilidad_Mutacion = 0.15;
   public static double Probabilidad_Cruza = 0.75;
   private int minValor;
   private int maxValor;
   public int Cont;
   public int Tamanio;
   public int Cont_Set_Atrib;
   public CIndividuo[] Poblacion;
   public CIndividuo Mejor = null;
   public boolean Evaluado = false;
   public boolean Poblado = false;
   public double Total_Evaluacion = -1;
   public CGeneracion Antecesores = null;
   public CGeneracion Sucesores = null;
   public CDatabase DB;

   // Declara los parametros necesarios de la clase
   public CGeneracion(int tamanio, int atrib_cont, int min_valor, int max_valor, CDatabase database)
   {  Cont_generacion = 0;
      DB = database;
      minValor = min_valor;
      maxValor = max_valor;
      Cont_Set_Atrib = atrib_cont;
      Tamanio = tamanio;
      Poblacion = new CIndividuo[Tamanio];
      Cont = ++Cont_generacion;
   }

   // Estos son los parametros de la generacion anterior
   private CGeneracion(CGeneracion fuente)     // fuente = el antecesor
   {  Cont_Set_Atrib = fuente.Cont_Set_Atrib;
      Tamanio = fuente.Tamanio;
      Poblacion = new CIndividuo[Tamanio];
      Cont = ++Cont_generacion;
      Antecesores = fuente;
      minValor = fuente.minValor;
      maxValor = fuente.maxValor;
      DB = fuente.DB;
   }

   // Crea la poblacion de individuos
   public void Poblar()
   {  for (int i = 0; i < Tamanio; i++)
      {  Poblacion[i] = new CIndividuo(this);
         Poblacion[i].Generador(minValor, maxValor, Cont_Set_Atrib);
      }
      Poblado = true;
   }

   // Metodo de evaluacion a nivel generacion
   public void Evaluar()
   {  if (!Poblado) return;
      Total_Evaluacion = 0;
```

```
    double max = 0;
    //encuentra al mejor individuo y evalua c/uno
    for (int i = 0; i < Tamanio; i++)
    {  // val = TP + Correlacion
       double val = Poblacion[i].Evaluar();
       if ((Mejor == null || val > max) && !(Mejor != null && Mejor.Evaluacion == 0))
       {  Mejor = Poblacion[i];                                    }
       if (val > max) max = val;
       Total_Evaluacion += val;
    }
    // Calcula la aptitud relativa
    for (int i = 0; i < Tamanio; i++)
    {  CIndividuo individuo = Poblacion[i];
       individuo.Aptitud_Rel = individuo.Obten_Eval() / Total_Evaluacion;
    }
    Evaluado = true;
}

// Metodo de Seleccion de la Ruleta
public void Seleccionar()
{   // Toma al mejor individuo de la generacion anterior como numero 0
    Poblacion[0] = new CIndividuo(this);
    Poblacion[0].Heredar(Antecesores.Mejor);
    TreeMap<Double, ArrayList<CIndividuo>> ruleta = new TreeMap<Double, ArrayList<CIndividuo>>();
    // this contiene un cto d listas d indiv's agrupados x su valor d aptit.
    // cada registro en el TreeMap es una Lista de individuos (ArrayList)
    //y es definida por una clave
    for (int i = 0; i < Tamanio; i++)
    {
       CIndividuo fuente = Antecesores.Poblacion[i];
       if (fuente == Antecesores.Mejor) continue;
       // se ignora al mejor individuo
       Double clave = new Double(fuente.Aptitud_Rel);
       // Java no permite que se usen tipos primitivos como clave
       // Doble es una clase de java que simula el tipo primitivo
       ArrayList<CIndividuo> Lista;
       if (ruleta.containsKey(clave))
       {  Lista = ruleta.get(clave);                               }
       else
       {  Lista = new ArrayList<CIndividuo>();
          ruleta.put(clave, Lista);
       }
       Lista.add(fuente);
    }
    Set<Double> Cto_claves = ruleta.keySet();
    double[] claves = new double[Cto_claves.size()];
    Iterator<Double> iteradorClaves = Cto_claves.iterator();
    int indice = 0;
    double sumaClaves = 0;
    while (iteradorClaves.hasNext())
    {  claves[indice] = iteradorClaves.next();
       sumaClaves += claves[indice++];
    }
    Random rnd = new Random();
    int contador = 1;
    while(contador < Tamanio)
```

```
    {  double n = rnd.nextDouble() * sumaClaves;
       double claveAnterior = 0;
       double clave = 0;
       int contClaves = claves.length;
       for (int i = 0; i < contClaves; i++)
       {  if ((n >= claveAnterior && n <= claves[i] + claveAnterior) || i == contClaves - 1)
          {  clave = claves[i];
             break;
          }
          claveAnterior += claves[i];
       }
       Double Clave = new Double(clave);
       if (ruleta.containsKey(Clave))
       {  ArrayList<CIndividuo> Lista = ruleta.get(Clave);
          if (Lista.size() > 0)
          {  CIndividuo objeto = Lista.get(0);
             Lista.remove(0);
             CIndividuo obj_nuevo = new CIndividuo(this);
             obj_nuevo.Heredar(objeto);
             Poblacion[contador++] = obj_nuevo;
          }
          if (Lista.isEmpty()) ruleta.remove(Clave);
       }
    }  Poblado = true;
}

// Metodo de Cruza a nivel generacion
public void Cruzar()
{  for (int i = 1; i < Tamanio - 1; i += 2)
    {   Poblacion[i].Cruzar(Probabilidad_Cruza, Poblacion[i+1]);       }
}

// Metodo de Mutacion a nivel generacion
public void Mutar()
{  for (int i = 0; i < Tamanio; i++)
    {   Poblacion[i].Mutar(Probabilidad_Mutacion);                     }
}

// Pasa a la siguiente generacion
public CGeneracion Generacion_siguiente()
{  if (!Poblado) return null;
   // this referencia a la generacion en la que se esta trabajando
   CGeneracion siguiente = new CGeneracion(this);
   this.Sucesores = siguiente;
   siguiente.Seleccionar();
   siguiente.Cruzar();
   siguiente.Mutar();
   return siguiente;
}

// Limpia todos los parametros de la generacion actual
public void Eliminar()
{  DB = null;
   Antecesores = null;
   Sucesores = null;
   Mejor = null;
```

```
      if (Poblacion != null)
      {  for (int i = 0; i < Poblacion.length; i++) Poblacion[i].Eliminar();
         Poblacion = null;
      }
   }
}
```

**********

## *A.5 CIndividuo.java*

Contiene todas las funciones propias de un Algoritmo Genético a nivel individuo; es decir, la creación de un objeto o 'individuo', el generador de los genes del individuo (atributos), la función de herencia, la función de evaluación a nivel 'gen', que compara el 'outcome' obtenido con el valor de salida de cada registro de la BD; los operadores de cruza y mutación a nivel 'gen', dos funciones de manejo de cadenas a texto, el cálculo de la calificación de cada individuo de acuerdo al coeficiente de correlación de cada atributo, y la función de Evaluación, que nos sirve para evaluar ó emitir un resultado final para cada individuo, de acuerdo a la tasa de precisión y la correlación de los atributos con la clase.

```
package tesis;
import java.io.*;
import java.util.*;
/**
 * @author Zure
 * Aqui se ejecutan todas las funciones relacionadas con el individuo de un AG
 */

public class CIndividuo {
   private static Random celdas = new Random();
   public int Datos[];
   public int Evaluacion = -1;
   public double Aptitud_Rel = -1;
   public double Correlacion = 0;
   public boolean Mutado = false;
   public boolean Cruzado = false;
   public int VP = 0;
   public int VN = 0;
   public int FP = 0;
   public int FN = 0;
   public boolean califica = false;
   public int VP2 = 0;
   public int VN2 = 0;
   public int FP2 = 0;
   public int FN2 = 0;
   public int Encontrado = 1;
   public CGeneracion Generacion;
   private CDatabase DB;
   private int valMin, valMax;
   private int generacCont;

   // Obtiene los parametros basicos de la generacion
   public CIndividuo(CGeneracion num_generacion)
   {  Generacion = num_generacion;
      DB = Generacion.DB;
      generacCont = Generacion.Cont;
   }
```

```
// Es la funcion que se encarga de hacer el 'Elitismo'
public void Heredar(CIndividuo fuente)
{  Datos = fuente.Datos.clone();
   valMin = fuente.valMin;
   valMax = fuente.valMax;
   CalcCorrelacion();
}

// Metodo que genera los individuos
public void Generador(int Val_min, int Val_max, int tamanio)
{   // genera #'s entre Val_min y Val_max q pueden estar en cualquier
   //posicion dentro del arreglo de un individuo
   valMin = Val_min;
   valMax = Val_max;
   Datos = new int[tamanio];
   // Genera el contenido del individuo
   for (int i = 0; i < tamanio; i++)
   {  int valor;
      boolean Valor_existente = false;
      do
      {  valor = celdas.nextInt(Val_max) + Val_min;
         for (int j = 0; j < i; j++)
         {
            Valor_existente = Datos[j] == valor;
            if (Valor_existente) break;
         }
      }  while (Valor_existente);
      Datos[i] = valor;
   }
   CalcCorrelacion();
}

// Es la funcion que evalua a cada individuo de acuerdo a su tasa de precisión y el coef. de correl.
public double Evaluar()
{  Evaluacion = 0;
   for (int i = 0; i < DB.Cont_registros(); i++)
   {  CRegistro registro = DB.ObtenRegistro(i);
      // Comparamos la clasificacion calculada con el valor d la clase d la BD
      boolean clase = Clasificar(registro);
      if (!clase)
      {  Evaluacion++;
         if (califica)
         {  if (registro.Condicion) FN2++;
            else FP2++;
         }   else
         {    if (registro.Condicion) FN++;
              else FP++;
         }
      }   else
      {
         if (califica)
         {  if (registro.Condicion) VP2++;
            else VN2++;
         }   else
         {   if (registro.Condicion) VP++;
             else VN++;
```

```
            }
        }
    }  return Obten_Eval();
}

// Es el metodo que se encarga de mutar al individuo
public boolean Mutar(double ProbMutacion)
{  double oportunidad_mutar = celdas.nextDouble();
   Mutado = oportunidad_mutar < ProbMutacion;
   if (Mutado)
   {  int max = Datos.length;
      int indice = celdas.nextInt(max);
      int valor;
      boolean Valor_existente = false;
      do
      {  valor = celdas.nextInt(valMax) + valMin;
         for (int j = 0; j < max; j++)
         {  Valor_existente = Datos[j] == valor;
            if (Valor_existente) break;
         }
      } while (Valor_existente);
        Datos[indice] = valor;
   }
   CalcCorrelacion();
   return Mutado;
}

// Este metodo se encarga de cruzar a un par de individuos
public boolean Cruzar(double Prob_Cruza, CIndividuo pareja)
{   // genero cualquier no. aleatorio
   double oportunidad_cruzar = celdas.nextDouble();
   Cruzado = oportunidad_cruzar < Prob_Cruza;
   if (Cruzado)
   {  ArrayList<Integer> padre1 = new ArrayList<Integer>();
      ArrayList<Integer> padre2 = new ArrayList<Integer>();
      for (int i = 0; i < Datos.length; i++)
      {  padre1.add(new Integer(Datos[i]));
         padre2.add(new Integer(pareja.Datos[i]));
      }
      ArrayList<Integer> duplicados = new ArrayList<Integer>();
      Iterator<Integer> iterador = padre1.iterator();
      while (iterador.hasNext())
      {  Integer valor = iterador.next();
         if (padre2.contains(valor)) duplicados.add(valor);
      }
      ArrayList<Integer> hijo1 = new ArrayList<Integer>();
      ArrayList<Integer> hijo2 = new ArrayList<Integer>();
      hijo1.addAll(duplicados);
      hijo2.addAll(duplicados);
      duplicados.clear();
      int max_tam = padre1.size();
      int ind = hijo1.size();
      int diferencia = max_tam - ind;
      if (diferencia > 0)
      {  int rango = celdas.nextInt(diferencia);
         int tamRango = diferencia - rango;
```

```
            if (tamRango > 1) tamRango = celdas.nextInt(tamRango);
            iterador = padre1.iterator();
            Iterator<Integer> pIterador = padre2.iterator();
            for (int i = ind; i < max_tam; i++)
            {  if (i >= ind + rango && i < ind + rango + tamRango)
               {   // Aqui se copian los datos de la pareja (this - pareja)
                  Integer valor = iterador.next();
                  while (hijo2.contains(valor)) valor = iterador.next();
                  hijo2.add(valor);
                  valor = pIterador.next();
                  while (hijo1.contains(valor)) valor = pIterador.next();
                  hijo1.add(valor);
               }
               else   {
                  // Aqui se toman los datos q pertenecen a los individuos (this - this, padra 1 - padre 2)
                  Integer valor = iterador.next();
                  while (hijo1.contains(valor)) valor = iterador.next();
                  hijo1.add(valor);
                  valor = pIterador.next();
                  while (hijo2.contains(valor)) valor = pIterador.next();
                  hijo2.add(valor);
               }
            }
         }
         iterador = hijo1.iterator();
         for (int i = 0; i < Datos.length; i++) Datos[i] = iterador.next();
         iterador = hijo2.iterator();
         for (int i = 0; i < pareja.Datos.length; i++) pareja.Datos[i] = iterador.next();
         CalcCorrelacion();
         pareja.CalcCorrelacion();
      }
      return Cruzado;
   }

   // Convierte el arreglo numerico en texto para imprimirlo
   public String Convierte_enTexto()
   {  String resultado = "";
      for (int i = 0; i < Datos.length; i++)
      {  resultado += Datos[i];
         if (i < Datos.length - 1) resultado += ".";
      }  return resultado;
   }

   // Convierte el arreglo numerico en texto de formato CSV para imprimirlo
   public String Convierte_enCSV()
   {  String resultado = "";
      for (int i = 0; i < Datos.length; i++)
      {  resultado += Datos[i];
         if (i < Datos.length - 1) resultado += ",";
      }  return resultado;
   }

   // Imprime el contenido y muestra el no. de fallas y su aptitud relativa
   public String Texto_Completo()
   {  return "Datos: " + Convierte_enTexto() + " (Aptitud = " + Aptitud_Rel +
         ", VP = " + VP + ", VN = " + VN + ", FP = " + FP + ", FN = " + FN + ", TP = " + Obten_TP() +
```

```
          ", TVP = " + Obten_TVP() + ", TVN = " + Obten_TVN() + ", Precision = " + Obten_Precision() + ",
          Correlacion = " + Correlacion + "," + correlacionText + ")";
   }

    // Imprime el contenido y muestra el no. de fallas y su aptitud relativa
   public String Texto_Completo2()
   {  return "Datos: " + Convierte_enTexto() + " (Aptitud = " + Aptitud_Rel +
            ", VP = " + VP + ", VN = " + VN + ", FP = " + FP + ", FN = " + FN + ", TP = " + Obten_TP() +
            ", TVP = " + Obten_TVP() + ", TVN = " + Obten_TVN() + ", Precision = " + Obten_Precision() + ",
          Correlacion = " + Correlacion + ", VP2 = " + VP2 + ", VN2 = " + VN2 + ", FP2 = " + FP2 + ", FN2
          = " + FN2 + ", TP2 = " + Obten_TP2() + ", TVP2 = " + Obten_TVP2() + ", TVN2 = " +
          Obten_TVN2() + ", Precision2 = " + Obten_Precision2() + ")";
   }

   // Imprime el contenido en el formato CSV y muestra el no. de fallas y su aptitud relativa
   public String Texto_CSV()
   {  return "Individuo," + Convierte_enCSV() + ",Aptitud," + Aptitud_Rel +
            ",VP," + VP + ",VN," + VN + ",FP," + FP + ",FN," + FN + ",TVP," + Obten_TVP() +
            ",TVN," + Obten_TVN() + ",Precision," + Obten_Precision() + ",TP," + Obten_TP() +
            ",Correlacion," + Correlacion + "," + correlacionText;
   }

   // Imprime el contenido en el formato CSV y muestra el no. de fallas y su aptitud relativa
   public String Texto_CSV2()
   {  StringWriter sw = new StringWriter(1024);
      PrintWriter printer = new PrintWriter(sw);
      printer.println(Texto_CSV());
      for (int i = 0; i < Datos.length + 3; i++) printer.print(",");
      printer.print(
         "VP2," + VP2 + ",VN2," + VN2 + ",FP2," + FP2 + ",FN2," + FN2 + ",TVP2," + Obten_TVP2() +
         ",TVN2," + Obten_TVN2() + ",Precision2," + Obten_Precision2() + ",TP2," + Obten_TP2() );
      return sw.toString();
   }

   // 'Limpia' los datos y parametros de la generacion
   public void Eliminar()
   {  Generacion = null;
      DB = null;
      Datos = null;
   }

   // Metodo que se encarga del calculo de la Correlación de cada atributo del individuo dentro del AG
private void CalcCorrelacion()
   {  Correlacion = 0;
      StringWriter cad = new StringWriter(256);
      PrintWriter printer = new PrintWriter(sw);
      int modo = CConfig.Config.ObtenCorrelacionModo();
      for (int j = 0; j < Datos.length; j++)
      {  int ind = Datos[j]-1;
         double correl = DB.Correl[ind];
         if (j > 0) printer.print(",");
         printer.print(correl);
         switch (modo)
         {  case 1: // portofolio
               if (correl <= 0) correl = 0.1;
               else if (correl <= 0.04) correl = 0.4;
```

```
            else if (correl < 0.06) correl = 0.6;
            else if (correl < 0.08) correl = 0.8;
            else correl = 1;
            break;
         case 2: // wdbc
            if (correl <= 0.1) correl = 0.1;
            else if (correl <= 0.3) correl = 0.4;
            else if (correl < 0.5) correl = 0.6;
            else if (correl < 0.7) correl = 0.8;
            else correl = 1;
            break;
         case 3: // wpbc
            if (correl > -0.01 && correl < 0.01) correl = 0.1;
            else if ((correl >= 0.01 && correl < 0.05) || (correl <= -0.01 && correl > -0.05))
            {  correl = 0.4;
            }
            else if ((correl >= 0.05 && correl < 0.1) || (correl <= -0.05 && correl > -0.1))
            {  correl = 0.6;
            }
            else if ((correl >= 0.1 && correl < 0.2) || (correl <= -0.1 && correl > -0.2))
            {  correl = 0.8;
            }
            else correl = 1;
            break;
      }  Correlacion += correl;
   }  correlacionText = cad.toString();
}

// Metodo que se encarga de Clasificar al individuo dentro del AG
public boolean Clasificar(CRegistro registro)
{  int Recurrente = 0;
   for (int j = 0; j < Datos.length; j++)
   {   if (registro.Regla_Clasifica(Datos[j])) Recurrente++;              }
   int limite = Datos.length / 2;
   boolean resultado = registro.Condicion ? Recurrente > limite : Recurrente < limite;
   return resultado == registro.Condicion;
   if (califica)
   {  if (!resultado)
      {  Evaluacion++;
         if (registro.Condicion) FN2++;
         else FP2++;
      }
      else
      {  if (registro.Condicion) VP2++;
         else VN2++;
      }
   }   return resultado;
}

// Obtiene la Precision para el cto. de Entrenamiento
public double Obten_Precision()
{   return VP + FP == 0 ? 0 : VP / (double)(VP + FP);                    }

// Obtiene la Tasa de Verdaderos Positivos para el cto. de Entrenam
public double Obten_TVP()
{   return VP + FN == 0 ? 0 : VP / (double)(VP + FN);                    }
```

```
// Obtiene la Tasa de Verdaderos Negativos para el cto. de Entrenamiento
public double Obten_TVN()
{   return VN + FP == 0 ? 0 : VN / (double)(VN + FP);                    }

// Obtiene la Tasa de Precision para el cto. de Entrenamiento
public double Obten_TP()
{   return VP + VN + FP + FN == 0 ? 0 : (VP + VN) / (double)(VP + VN + FP + FN);
}

public int Obten_Total(boolean clasificacion)
{   return clasificacion ? VP2 + VN2 + FP2 + FN2 : VP + VN + FP + FN;     }

// Obtiene la Precision para el cto. de Prueba
public double Obten_Precision2()
{   return VP2 + FP2 == 0 ? 0 : VP2 / (double)(VP2 + FP2);               }

// Obtiene la Tasa de Verdaderos Positivos para el cto. de Prueba
public double Obten_TVP2()
{   return VP2 + FN2 == 0 ? 0 : VP2 / (double)(VP2 + FN2);               }

// Obtiene la Tasa de Verdaderos Negativos para el cto. de Prueba
public double Obten_TVN2()
{   return VN2 + FP2 == 0 ? 0 : VN2 / (double)(VN2 + FP2);               }

// Obtiene la Tasa de Precision para el cto. de Prueba
public double Obten_TP2()
{   return VP2 + VN2 + FP2 + FN2 == 0 ? 0 : (VP2 + VN2) / (double)(VP2 + VN2 + FP2 + FN2);
}

// Toma los mejores individuos de las 'corridas'
public void Desconectar()
{  for (int i = 0; i < Generacion.Poblacion.length; i++)
   {  if (this == Generacion.Poblacion[i])
      {  Generacion.Poblacion[i] = null;
         break;
      }
   }
   Generacion = null;
   DB = null;
   TreeSet<Integer> lista = new TreeSet<Integer>();
   for (int i = 0; i < Datos.length; i++)
   {   lista.add(new Integer(Datos[i]));                                 }
   int ndx = 0;
   Iterator<Integer> iterador = lista.iterator();
   while (iterador.hasNext())
   {   Datos[ndx++] = iterador.next();                                   }
}

// Obtiene los mejores individuos que se repitan
public boolean IgualQue(CIndividuo otro)
{  boolean resultado= false;
   for (int i = 0; i < Datos.length; i++)
   {  resultado = Datos[i] == otro.Datos[i];
      if (!resultado) break;
   }    return resultado;
```

```
    }

    // Regresa el numero de la generacion en donde pertenece el mejor indiv.
    public int Obten_GenCont()
    {   return generacCont;                                          }

    // Regresa los parámetros de la evaluación del mejor individuo
    public double Obten_Eval()
    {  return Obten_TP() + Correlacion;                              }
}

**********
```

## *A.6 CDatabase.java*

Aquí se encuentran todos los procedimientos propios del manejo de una base de datos como son: la obtención de los parámetros de configuración, la lectura y limpieza de los datos; además de funciones como obtener y contar registros, normalizarlos (como parte de la fase de 'Preparación de los Datos'). Parte el conjunto de datos en dos subconjuntos, de acuerdo a 3 posibilidades: Directo, Aleatorio o Estadístico, para que los conjuntos de entrenamiento y prueba contengan una proporción balanceada de clases positivas y negativas. También, encontramos aquí la función '*Clasificar'*, que nos sirve para contrastar nuestro diagnóstico con el del estudio original o *outcome.*

```
package tesis;
import java.io.*;
import java.util.*;
import org.apache.commons.math3.stat.correlation.PearsonsCorrelation;
/**
 * @author Zure
 * Aqui se encuentran todas las funciones concernientes al manejo de la BD
 */

public class CDatabase {
    private List<CRegistro> registros = new ArrayList<CRegistro>();
    private String Path;
    public int min_num_columnas = 30;
    public double[] Media_m;
    public double[] Desv_Stand;
    public double[] Correl;
    public String CadenaCorrecta;
    public String CadenaFalsa;
    public int Muestra;
    // Aqui se definen todos los parametros relacionados con el manejo de la BD
    public CDatabase(CConfig config)
    {  Path = config.Obten_DatabasePath();
       min_num_columnas = config.Obten_ContColumna() - config.Obten_ColumIgnoradas().size();
       CadenaCorrecta = config.ObtenCaden_Verdadera();
       CadenaFalsa = config.ObtenCadena_Negativa();
       Muestra = config.Obten_Muestra();
    }

    // Aqui se lee la BD
    public boolean Leer()
    {  Limpiar();
       ArrayList<String> lineas = new ArrayList<String>();
       try
       {  FileReader lector = new FileReader(Path);
```

```
        BufferedReader lector_b = new BufferedReader(lector);
        String renglon;
        try
        {  while ((renglon = lector_b.readLine()) != null)
             {   lineas.add(renglon);                                    }
           lector_b.close();
           lector.close();
        }
        catch (IOException x)
        {   return false;                                                }
    }
    catch (FileNotFoundException x)
    {   return false;                                                    }
    for (int i = 0; i < lineas.size(); i++)
    {  registros.add(new CRegistro(lineas.get(i), this, CadenaCorrecta));
    }
    return true;
}

// Cuenta los registros de la BD
public int Cont_registros()
{   return registros.size();                                             }

// Carga los registros de la BD
public CRegistro ObtenRegistro(int indice)
{   return indice >= 0 && indice < Cont_registros() ? registros.get(indice) : null;
}

// Limpia los parametros
public void Limpiar()
{  registros.clear();
   Media_m = null;
   Desv_Stand = null;
   Correl = null;
}

// Aqui se dividen los datos en los ctos. de Entrenamiento y Prueba
public void Muestreo(double porciento)
{  if (registros.isEmpty()) return;
   switch (Muestra)
   {  case 1:
         Selecc_Directo(porciento);
         break;
      case 2:
         Selecc_Aleator(porciento);
         break;
      case 3:
         Selecc_Estadistico(porciento);
         break;
   }
   int cont = registros.size();
   int tam_regist = registros.get(0).Obtener_Tam();
   double [] sumas = new double [tam_regist];
   double[][] matriz = new double[tam_regist][];
   double[] cad_salida = new double[cont];
   for (int i = 0; i < tam_regist; i++)
```

```
    {  sumas[i] = 0;
       matriz[i] = new double[cont];
    }
    for (int i = 0; i < cont; i++)
    {  CRegistro registro = registros.get(i);
       for (int j = 0; j < tam_regist; j++)
       {  sumas[j] += registro.Datos[j];
          matriz[j][i] = registro.Datos[j];
          cad_salida[i] = registro.Condicion ? 1 : 0;
       }
    }
    Correl = new double [tam_regist];
    PearsonsCorrelation pc = new PearsonsCorrelation();
    for (int i = 0; i < tam_regist; i++)
    {  Correl[i] = Math.abs(pc.correlation(matriz[i], cad_salida));          }

    //Normalizacion propuesta
    for (int i = 0; i < cont; i++)
    {  CRegistro registro = registros.get(i);
       for (int j = 0; j < tam_regist; j++)
          if (sumas[j] != 0) registro.Datos[j] /= sumas[j];
    }
    Media_m = new double [tam_regist];
    Desv_Stand = new double [tam_regist];
    for (int i = 0; i < tam_regist; i++) Media_m[i] = Desv_Stand[i] = 0;
    for (int i = 0; i < cont; i++)
    {  CRegistro registro = registros.get(i);
       for (int j = 0; j < tam_regist; j++) Media_m[j] += registro.Datos[j];
    }
    for (int i = 0; i < tam_regist; i++) Media_m[i] /= cont;
    for (int i = 0; i < cont; i++)
    {  CRegistro registro = registros.get(i);
       for (int j = 0; j < tam_regist; j++)
       {  double x = registro.Datos[j] - Media_m[j];
          Desv_Stand[j] += x*x;
       }
    }  int div = cont - 1;
    for (int i = 0; i < tam_regist; i++) Desv_Stand[i] = Math.sqrt(Desv_Stand[i] / div);
}

// Se dividen los datos en dos simples conjuntos
private void Selecc_Directo(double porciento)
{  int cont = registros.size();
    if (porciento > 0)
    {  cont *= porciento/100;
       while (registros.size() > cont) registros.remove(registros.size() - 1);
    }
    else  {
       int tam_completo = cont;
       cont *= (100+porciento)/100;
       registros = registros.subList(cont, tam_completo);
       cont = tam_completo - cont;
    }
}

// Se seleccionan los datos aleatoriamente
```

```
private void Selecc_Aleator(double porciento)
{  if (porciento < 0) porciento = -porciento;
   int cont = registros.size();
   cont *= porciento/100;
   Random rnd = new Random();
   while (registros.size() > cont) registros.remove(rnd.nextInt(registros.size()));
}

// Se selecc. los datos de forma representativa e independiente de la clase
private void Selecc_Estadistico(double porciento)
{  int cont = registros.size();
   List<CRegistro> positivo = new ArrayList<CRegistro>();
   List<CRegistro> negativo = new ArrayList<CRegistro>();
   for (int i = 0; i < cont; i++)
   {  CRegistro registro = registros.get(i);
      if (registro.Condicion) positivo.add(registro);
      else negativo.add(registro);
   }
   registros.clear();
   if (porciento > 0)
   {  cont = positivo.size();
      cont *= porciento/100;
      registros = positivo.subList(0, cont - 1);
      cont = negativo.size();
      cont *= porciento/100;
      registros.addAll(negativo.subList(0, cont - 1));
   }
   else   {
      cont = positivo.size();
      int tam_completo = cont - 1;
      cont *= (100+porciento)/100;
      registros = positivo.subList(cont, tam_completo);
      cont = negativo.size();
      tam_completo = cont - 1;
      cont *= (100+porciento)/100;
      registros.addAll(negativo.subList(cont, tam_completo));
   }
}

// Es la funcion que clasifica los registros a nivel BD
public ArrayList<String> Clasificar(CIndividuo mejor)
{  int total = registros.size();
   int cont_aciertos = 0;
   int cont_fallas = 0;
   ArrayList<String> lista = new ArrayList<String>();
   for (int i = 0; i < total; i++)
   {  CRegistro registro = registros.get(i);
      boolean clasificacion = mejor.Clasificar(registro);
      String Clase;
      if (clasificacion)
      {  cont_aciertos++;
         Clase = registro.Condicion ? CadenaCorrecta : CadenaFalsa;
      }
      else {
         cont_fallas++;
         Clase = registro.Condicion ? CadenaFalsa : CadenaCorrecta;
```

```
            Clase += " ; - Error";
        }
        Formatter dar_formato = new Formatter();
        String datos =  dar_formato.format("%1$4d ;   ", i + 1) + (registro.Condicion ? CadenaCorrecta :
CadenaFalsa) + "   ;   " + Clase;
        lista.add(datos);
      }  String Resultados_Finales = "Clasificando " + total + " Registros: " + cont_aciertos +
            " registros clasificados correctamente, " + cont_fallas +
            " registros clasificados incorrectamente, % de falla: " + (100.0 * cont_fallas / total);
      lista.add(Resultados_Finales);
      return lista;
   }
}
```

**********

## *A.7 CRegistro.java*

Contiene las funciones propias de las características de los registros, como: el tamaño del registro y la BD, estadísticos necesarios obtenidos de los datos, y la función que determina si el registro forma parte del conjunto de entrenamiento o del de prueba.

```
package tesis;
import java.util.ArrayList;
/**
 * @author Zure
 * Esta clase contiene los metodos sobre los registros de la BD para manejarla.
 */

public class CRegistro {
   public boolean Condicion;
   public double [] Datos;
   private CDatabase DB;
   //'Carga' la BD con la que se trabajara
   public CRegistro(String datos, CDatabase DB, String cadena_correcta)
   {  this.DB = DB;
      CConfig config = CPrincipal.Configurar();
      int ColumnaClase = config.ObtenerColumnaClase();
      ArrayList<Integer> salta_Columnas = config.Obten_Colum_Ignoradas();
      String[] partes = datos.split(",");
      int tamanio = partes.length - salta_Columnas.size();
      Datos = new double[tamanio < DB.min_num_columnas ? DB.min_num_columnas : tamanio];
      Condicion = partes[ColumnaClase].equals(cadena_correcta);
      int ndx = 0;
      for (int i = 0; i < partes.length; i++)
      {  if (salta_Columnas.contains(new Integer(i))) continue;
         double valor_dato;
         try
         {   valor_dato = Double.parseDouble(partes[i]);                    }
         catch (Exception x)
         {   valor_dato = 0;                                                }
         Datos[ndx++] = valor_dato;
      }
       if (tamanio < DB.min_num_columnas)
      {   for (int i = tamanio; i < DB.min_num_columnas; i++) Datos[i] = 0;
      }
   }
```

```
   //Obtiene el tamanio de la BD
   public int Obtener_Tam()
   {   return Datos.length;                                          }

   // Es usado por la funcion del AG que clasifica al individuo.
   public boolean Regla_Clasifica(int ind)
   {  ind--;
      double valor = Datos[ind];
      double sd = DB.Desv_Stand[ind];
      double media = DB.Media_m[ind];
      return ((valor >= media - sd) && (valor <= media + sd));
   }
}
```

**********

## *A.8 Config.txt (Archivo de texto)*

Archivo de Texto que contiene todos los parámetros que el algoritmo "Attribute_Classification" necesita para obtener los ***k*** atributos más representativos de la base de datos de entrada.

```
; Notas:
; El texto después del ; es un comentario.
; Cuando un valor no puede ser leído desde el archivo de configuración, es porque está comentado o hay un
error - y su valor por default será tomado.
; Los espacios son removidos cuando se leen los valores.
; Parámetros de la Base de Datos - path y contador de columnas

; Configuración para WDBC y WPBC (Data/wdbc.csv) (Data/wpbc.csv)
;Database = Data/wdbc.csv
;cont_atrib = 32   ;35
;Lugar_colum_outcome = 1 ; lugar de la column de salida 'outcome' usada para la clasificación, -1 si no hay
; estos 2 parámetros son usados para determinar si el objectivo binario es (+) o (-), en la columna de salida.
;salida_positiva = M  ;R
;salida_negativa = B  ;N
;omitir_columnas = 0

; Configuración para BD_Portafolio.csv
Database = Data/BD_Portafolio.csv
cont_atrib = 34
Lugar_colum_outcome = 33 ; lugar de la column de salida 'outcome' usada para la clasificación, -1 si no hay
; estos 2 parámetros son usados para determinar si el objectivo binario es (+) o (-), en la columna de salida.
Salida_positiva = 1
Salida_negativa = 0
omitir_columnas = 0

Lector_db = default ; define una forma de leer la base de datos
correlacion_modo = 3 ; 1 - portofolio, 2 - wdbc, 3 - wpbc
muestra = 3          ; 1 - direct, 2 - random, 3 - estadistica
porcent_Entrenam = 70
eval_cont = 100

; Parámetros del Algoritmo Genético
Probabilidad_Mutacion = 0.05
Probabilidad_Cruza = 0.90
Tamanio_Poblacion = 200
```

```
Individuo_tamanio = 6
Cont_Max_Generacion = 1000

; Configuración de la Salida de los Resultados
Guardar_File = 0 ; 0 - no guardar los resultados en un archivo, 1 - guardar los resultados en:
File_Resultados = Data/result.txt
File_Resumen = Data/resumen.csv
```

## ANEXO B.

# B. WEKA.

El Ambiente de Waikato para el Análisis del Conocimiento, por sus siglas en inglés *WEKA,* es un toolkit ó conjunto de herramientas de trabajo programado en Java y distribuido bajo los términos de la Licencia General Pública GNU, creado por un grupo de desarrolladores del Departamento de las Ciencias de la Computación de la Universidad de Waikato en Hamilton, Nueva Zelanda. [Frank et al. 2005]

Se encuentra orientado principalmente en dos aspectos: la primera es en el desarrollo experimental tanto matemático como computacional de las técnicas utilizadas en las primeras cuatro fases del proceso del KDD, la extracción de conocimiento de grandes bases de datos (véase el capítulo II en la parte del KDD ); y la segunda, es el dar la posibilidad de tener amplio acceso a un software gratuito competitivo para extraer conocimiento de los grandes repositorios de datos. Ambos objetivos colocan a Weka como un software de vanguardia *(state-of-the-art)*.

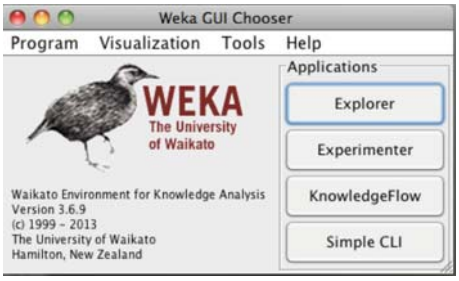



*Figura B.1 – Weka GUI – Interfaz de Bienvenida.*

Weka incluye varios métodos para prácticamente todos los problemas estándar en Minería de Datos: Regresión, clasificación, clustering y reglas de asociación. De la misma manera, este software proporciona algunas formas de visualización y varias herramientas de pre-procesamiento de datos, entre las que se encuentra la selección de atributos. Un importante aspecto de su arquitectura es su *'modularidad'*, lo cual permite a los algoritmos ser combinados de muchas formas diferentes. La mayoría de los algoritmos tiene una o más opciones que pueden ser especificadas, además de técnicas para medir el desempeño de

dichos algoritmos. Según [Frank et al. 2005], Weka puede ser usado de varias formas, como son:

1. Aplicar un método de aprendizaje a un conjunto de datos y analizar su salida para aprender más respecto de los datos.
2. Otro es, usar los modelos de aprendizaje para generar predicciones sobre instancias nuevas.
3. Una última forma es, aplicar diferentes métodos de aprendizaje y comparar su desempeño para escoger uno y para utilizarlo como predictor.

Weka tiene 3 principales ventajas. La primera es que es una fuente abierta, lo que significa que no sólo es gratuita sino que puede ser conservada y modificada en cualquier momento por los desarrolladores del área. La segunda es que provee algoritmos de vanguardia que están siendo desarrollados en el momento. Y la tercera es que esta implementada completamente en Java y puede correr bajo cualquier plataforma. Ha sido probado bajo Linux, Windows, Macintosh y aún sobre un asistente personal digital. Su principal desventaja es que la mayor parte de su funcionalidad es aplicable solamente si el conjunto completo de datos es mantenido en la memoria principal, aunque no siempre es así.

Como un conjunto de herramientas diverso y comprehensivo, es accesado a través de una interfaz común y uniforme para que los usuarios puedan comparar diferentes algoritmos de aprendizaje, junto con métodos de pre y pos – procesamiento, así como la evaluación de los resultados sobre uno o varios conjuntos de datos; y así, el usuario identifique aquellos que son más apropiados para el problema en turno. La implementación de los esquemas de aprendizaje junto con los filtros (herramientas para el pre-procesamiento de los datos), son los aspectos más valiosos del paquete.

Todos los algoritmos toman los datos de entrada de una tabla relacional en el formato ARFF, que puede ser generada de un archivo de texto o a través de un query, aunque también lee los formatos *'.csv'* y se puede realizar una conexión directa, vía una consulta SQL, con una base de datos.

Weka contiene un conjunto completo de algoritmos útiles para una gama de tareas en Minería de Datos. Esto incluye herramientas para la ingeniería de datos, llamadas *'filtros'*, algoritmos para selección de atributos, clustering, aprendizaje de reglas de asociación, clasificación y regresión, que pueden ser combinados de muchas formas diferentes. [Frank et al. 2005]

## B.1 Estructura de Weka.

Para hacer las operaciones lo más flexiblemente posibles, Weka fue diseñado bajo una arquitectura orientada a objetos; por lo tanto, si queremos entender la estructura de Weka, es necesario conocer como están organizados los programas en Java.

Cada programa en Java es implementado como una clase; en Weka, una clase puede encapsular un algoritmo de aprendizaje en particular, aunque los programas más largos usualmente están fragmentados en más de una clase. Weka está conformado por una gran cantidad de clases y para facilitar su navegación, Java permite organizar dichas clases en paquetes. Los paquetes están organizados en forma ordenada en correspondencia a un directorio jerárquico.

Weka esta compuesto por más de 40 paquetes, de los cuáles, algunos de los más importantes son:

- **weka.core:** Es el paquete central del sistema y sus clases son accesadas desde cualquier otra clase. Sus clases más importantes son: atributos, instancia e instancias (la base de datos).
- **weka.clasiffiers:** Contiene implementaciones de la mayoría de los algoritmos para clasificación y predicción numérica. La clase más importante es Classify, que define la estructura general de cualquier esquema para la clasificación o predicción numérica.
- **weka.associations:** Contiene los esquemas de aprendizaje de reglas de asociación.

- **weka.clusterers:** Contiene los métodos de aprendizaje no supervisado.
- **weka.estimators:** Contiene subclases de una clase genérica llama '*Estimator'* que computa diferentes tipos de distribuciones probabilísticas.
- **weka.filters:** Donde la clase *'Filter'* define la estructura general de las clases que contienen los algoritmos de filtrado.
- **weka.attributeSelection:** Que es la que contiene las diferentes clases para la selección de atributos.

Existe una documentación on-line que se puede acceder al descargar el software, el cuál incluye la explicación de todos los paquetes desarrollados, su estructura y su funcionalidad. [WEKA-Online 2005]

## B.2 Ambiente de Trabajo.

Al ingresar a Weka, uno debe escoger una de las siguientes opciones de interfaz, para acceder a los algoritmos de Minería de Datos:

1. Línea de Comandos.
2. El Explorador (Explorer).
3. El Flujo de Conocimientos (Knowledge Flow).
4. El Experimentador (Experimenter).

### B.2.1 Línea de Comandos.

Es la forma básica y más rudimentaria de interactuar con estos métodos, invocándolos desde la línea de comandos e ingresando a todas las características del sistema a través de comandos textuales, lo que implica el tener un amplio conocimiento de las instrucciones de cada algoritmo, así como de su funcionalidad y sus características.


*Figura B.2 – Weka GUI – Línea de Comandos.*

### B.2.2 El Explorador.

Es la interfaz de usuario gráfica principal y la más fácil de usar. Da acceso a todas sus facilidades empleando el menú de selección y las formas de llenado. Una de sus características más importantes, es que es un buen guía, forzando a seguir una secuencia apropiada de algoritmos, además que los valores por default aseguran que uno pueda obtener resultados coherentes. Sin embargo, al mismo tiempo es necesario saber lo que se está haciendo para entender lo que significan los resultados.

Contiene 6 paneles diferentes, presentados en diferentes pestañas, los cuáles son:

1. *Pre-procesamiento*: Carga el conjunto de datos y lo modifica de diferentes maneras.
2. *Clasificador*: Entrena los esquemas de aprendizaje que ejecutan la clasificación o regresión, y los evalúa.
3. *Cluster:* Encuentra grupos de datos en el conjunto de datos.
4. *Asociación:* Aprende reglas de asociación para los datos y las evalúa.
5. *Selección de Atributos:* Selecciona los atributos más relevantes en el conjunto de datos.
6. *Visualización:* Muestra diferentes gráficas de los datos en dos dimensiones e interactúa con ellas.

La desventaja principal del Explorador, es que mantiene toda la información en la memoria principal; es decir, al abrir el conjunto de datos, inmediatamente se carga todo

completamente, lo cual aumenta el tiempo de ejecución, y limita que Weka sólo pueda ser usado con bases de datos pequeñas o medianas.

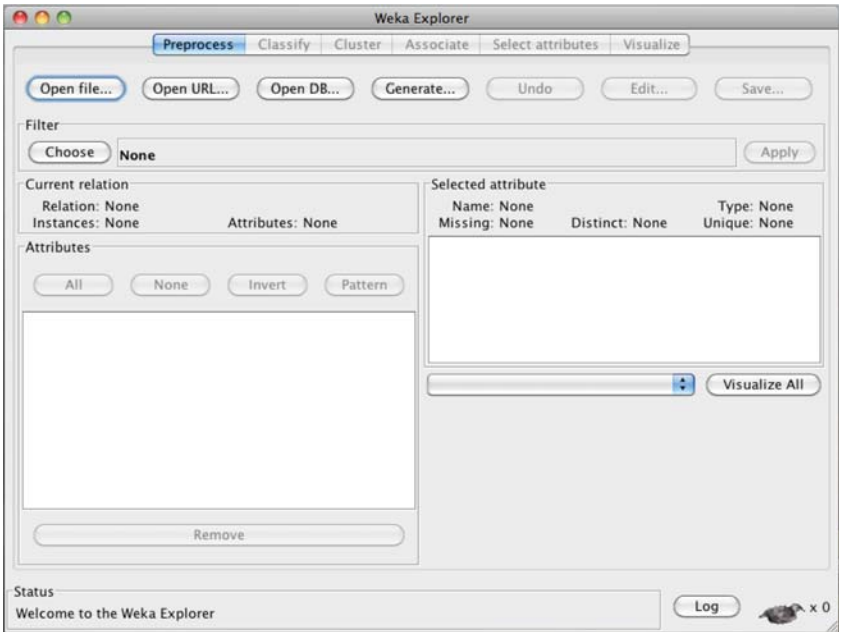



*Figura B.3 – Weka GUI – El Explorador.*

### B.2.3 El Flujo de Conocimientos.

Esta interfaz permite diseñar configuraciones especiales para el procesamiento de datos. Es decir, selecciona los componentes Weka que representan a los algoritmos de aprendizaje y fuentes de datos, desde una barra de herramientas para colocarlos en un lienzo dentro de la configuración que se requiera y conectándolos en una gráfica dirigida que procesa y analiza los datos, para obtener una corriente o flujo de datos. Lo anterior es muy útil, si es necesario entender cómo fluyen los datos a través del sistema y así tener una mayor comprensión acerca de los resultados.

Un sistema de componentes puede trabajar un conjunto de datos grandes si todos sus componentes en el flujo operan incrementalmente. Los sistemas incrementales pueden trabajar grandes conjuntos de datos debido a que, en lugar de leer los datos antes de que el sistema empiece, como lo hace el Explorador, cada componente del flujo lee la entrada

instancia por instancia y, así los va pasando a través de la cadena del Flujo de Conocimientos.

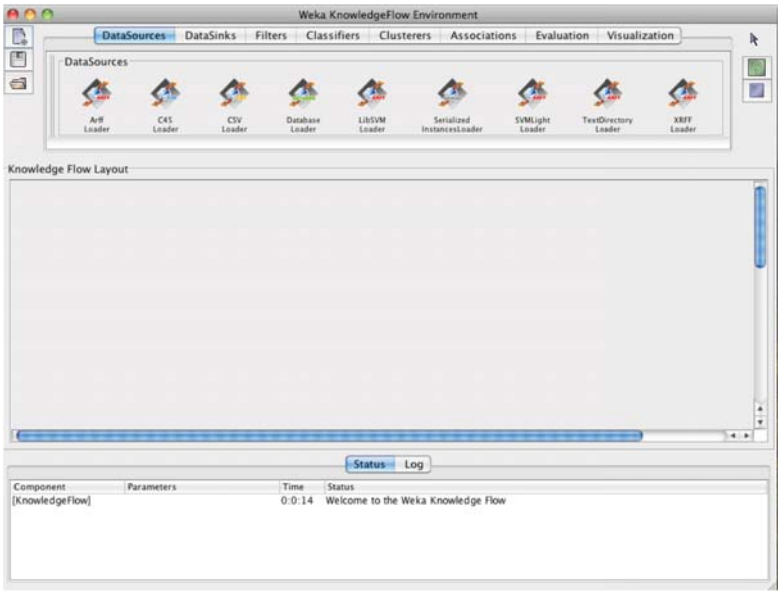



*Figura B.4 – Weka GUI – El Flujo de Conocimiento.*

Cada componente puede ser manipulado a través de tres opciones: Editar, Conexiones y Acciones. La mayoría de los componentes de esta interfaz son similares a las del Explorador; sin embargo, su verdadero valor esta en su potencial para la operación incremental, debido a que contiene algunos algoritmos incrementales que pueden ser usados para procesar conjuntos de datos muy grandes. Weka contiene varios clasificadores que pueden manejar incrementalmente los datos.

La configuración particular de esta interfaz puede procesar archivos de entrada de cualquier tamaño, aún aquellos que no caben en la memoria principal de la computadora; sin embargo, todo depende de cómo opera el clasificador internamente, dado que, aún cuando sean incrementales, muchos sistemas de aprendizaje basados en instancia almacenan el conjunto entero de datos internamente. Para saber con exactitud que algoritmos y filtros son incrementales, es necesario acudir a la documentación on-line. [WEKA-Online 2005]

### B.2.4 El Experimentador.

Los ambientes del Explorador y el Flujo de Conocimientos, ayudan a determinar que tan bueno es el desempeño de los esquemas de aprendizaje de máquina sobre conjuntos de datos dados. Sin embargo, los trabajos serios de investigación involucran experimentos más substanciales, corriendo varios esquemas de aprendizaje sobre diferentes conjuntos de datos y con varios ajustes en los parámetros. Y las interfaces antes mencionadas no son las adecuadas en este sentido. El experimentador, por su parte, esta diseñado para ayudar a responder una pregunta básica y práctica cuando se aplican las técnicas de clasificación y regresión: ¿Qué métodos y valores de los parámetros trabajan mejor para un problema dado?

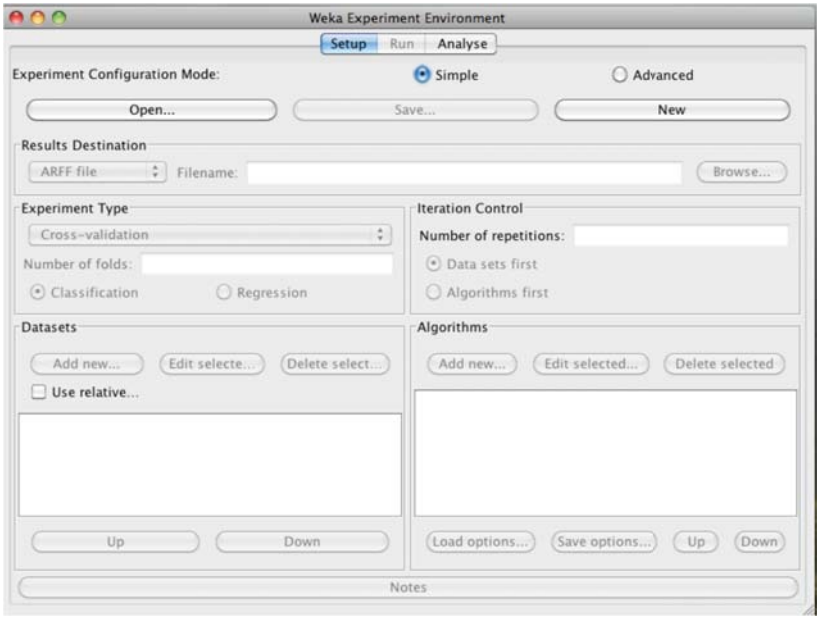



*Figura B.5 – Weka GUI – El Experimentador.*

Es muy difícil responder a la pregunta anterior dada la diversidad que existe entre los algoritmos de aprendizaje y entre los mismos conjuntos de datos. Sin embargo, una de las razones más importantes por la que se desarrolló Weka, fue para proveer un ambiente de trabajo que permita a los usuarios comparar una variedad de técnicas de aprendizaje, lo cual puede ser llevado a cabo en la interfaz del Experimentador. En este ambiente, se pueden automatizar los procesos, haciendo más fácil el correr los clasificadores y filtros con

diferentes ajustes de parámetros sobre diferentes conjuntos de datos, y también, muestra el desempeño estadístico de los diferentes esquemas de aprendizaje.

Usuarios más avanzados pueden emplear el Experimentador, para distribuir la carga computacional de experimentos a larga escala, sobre múltiples máquinas usando el Método de Invocación Remota de Java (RMI), empezarlos a ejecutar, dejarlos correr y regresar, una vez que ya haya terminado, para analizar su desempeño estadístico.

Mientras que el Flujo de Conocimientos trasciende las limitaciones del espacio, permitiendo correr los esquemas de aprendizaje sin que se haya cargado completamente el conjunto de datos, el Experimentador trasciende las limitaciones del tiempo, ya que permite hacer uso del cómputo distribuido.

## B.3 Selección de Atributos en Weka

En Weka, existe un panel especial para la tarea de selección de atributos donde uno puede especificar tanto el método de búsqueda, como el de la evaluación de los atributos. También hay la posibilidad de evaluar los atributos individualmente y después ordenarlos, aunque es un criterio más rápido y menos preciso.

En [Witten et al. 2011] se presentan dos tablas con la información de dichos métodos:

| **Métodos de Evaluación de Atributos para la Selección de Atributos** | | | |
|---|---|---|---|
| Tipo de Método | Nombre | Función | |
| Evaluador de Subconjuntos de Atributos | *CfsSubsetEval* | Considera el valor predictivo de cada atributo individualmente, junto con el grado de redundancia sobre todos ellos, prefiriendo los conjuntos de atributos que se encuentran altamente correlacionados con la clase pero tienen menor intercorrelación entre ellos. | Métodos de Filtrado |
| | *ConsistencySubsetEval* | Evalúa los conjuntos de atributos por su grado de consistencia en los valores de la clase cuando las instancias entrenadas son | |

| | | | |
|---|---|---|---|
| | | proyectadas hacia dentro del conjunto. | |
| | *ClassifierSubsetEval* | Usa un clasificador para evaluar los conjuntos de atributos sobre el conjunto de entrenamiento. | Métodos de Envolvimiento |
| | *WrapperSubsetEval* | También usa un clasificador para evaluar los subconjuntos de atributos, pero emplea la validación cruzada para estimar la precisión del esquema de aprendizaje para cada conjunto. | |
| Evaluador de Atributos Individuales | *ReliefFAttributeEval* | Es un evaluador de atributos basado en instancia. Toma muestras aleatorias de instancias y checa las instancias vecinas de la misma y de diferentes clases. Opera sobre clases continuas y discretas. | |
| | *InfoGainAttributeEval* | Evalúa los atributos midiendo su información ganada con respecto a la clase, discretizando los atributos numéricos con el método MDL. | |
| | *ChiSquaredAttributeEval* | Calcula el estadístico de Chi Cuadrada para cada atributo con respecto a la clase. | |
| | *GainRatioAttributeEval* | Evalúa los atributos midiendo su relación de ganancia con respecto a la clase. | |
| | *SymmetricalUncertAttributeEval* | Evalúa una atributo A midiendo su incertidumbre simétrica con respecto a la clase C. | |
| | *OneRAttributeEval* | Usa la medida de precisión adoptada por el clasificador OneR. | |
| | *SVMAttributeEval* | Evalúa los atributos usando la función recursiva de eliminación con una máquina vectorial de soporte lineal. Los atributos son seleccionados uno por uno basándose en el tamaño de sus coeficientes: complejidad, épsilon, tolerancia y el método de filtrado usado. | |
| | *PrincipalComponents* | Ejecuta el análisis de Componentes Principales para transformar el conjunto de atributos. Los nuevos atributos son enlistados de acuerdo a sus eigen-valores. | |

*Tabla B.1 – Métodos de Evaluación de Atributos para la Selección de Atributos.*

A continuación se explicará a grandes rasgos, cada uno de estos tipos de métodos de evaluación:

- **Métodos de Evaluación de Subconjuntos de Atributos:** Estos métodos toman un subconjunto de atributos y regresan una medida numérica que guía la búsqueda. Existen métodos tanto para el enfoque de filtrado como el de envoltura. En general, los primeros son evaluados por la habilidad predictiva de cada atributo y el grado de redundancia entre ellos, prefiriendo aquellos atributos que se encuentran altamente correlacionados con la clase, pero tienen una baja correlación entre ellos; no obstante, existen otras opciones de evaluación, como la tasa de grados de consistencia cuando las instancias de entrenamiento son proyectadas sobre los datos y trabaja en conjunto con alguna búsqueda aleatoria o exhaustiva. También, existen diferentes funciones para tratar los valores faltantes. Mientras que los segundos, usan algún clasificador para evaluar los subconjuntos de atributos sobre los conjuntos de entrenamiento y prueba.
- **Métodos de Evaluación de Atributos Individuales:** Las técnicas de evaluación para atributos individuales son usadas normalmente con el método de tipificación, *'Ranker'*, para generar una lista ordenada, en donde el ranker puede eliminar un número dado de atributos. Algunos de estos métodos están basados en el vecino más cercano, en la medida de la información ganada con respecto a la clase, calculando la Ji-Cuadrada, la técnica de validación cruzada, la eliminación de atributos recursiva con una máquina de soporte vectorial lineal ó, la técnica de componentes principales, que transforma el conjunto de atributos original en el conjunto de sus eigen-valores. Algunos de los métodos anteriores, pueden lidiar con atributos discretos o continuos y valores faltantes.

| **Métodos de Búsqueda para la Selección de Atributos** | | |
|---|---|---|
| Tipo de Método | Nombre | Función |
| Métodos de Búsqueda | *BestFirst* | Ejecutan una búsqueda exhaustiva de subir la colina con marcha hacia atrás; además, uno puede especificar cuántos nodos consecutivos no mejorables deben ser encontrados antes de que el sistema retroceda. Puede buscar hacia delante desde el conjunto de atributos vacío, retroceder desde el conjunto lleno, o empezar en un punto medio y buscar en ambas direcciones. |
| | *ExhaustiveSearch* | Busca a través del espacio de subconjuntos de atributos empezando desde el conjunto vacío, realizando una búsqueda exhaustiva; y reporta el mejor |

| | | |
|---|---|---|
| | | subconjunto encontrado. |
| | *GeneticSearch* | Usa un Algoritmo Genético Simple [Goldberg, 1989], con los siguientes parámetros: Tamaño de la población, número de generaciones y probabilidad de cruza y mutación. |
| | *GreedyStepwise* | Ejecuta una búsqueda exhaustiva de subir la colina sin marcha hacia atrás. Busca codiciosamente a través del espacio de subconjuntos de atributos. En un modo alternativo, clasifica los atributos atravesando el espacio de lo vacío a lo lleno, o viceversa, registrando el orden en que los atributos son seleccionados. |
| | *RaceSearch* | Usado con el clasificador ClassifierSubsetEval, calcula el error de validación cruzada de la competencia de subconjuntos de atributos usando la búsqueda de la carrera. Implementa las 4 técnicas de búsqueda conocidas: selección hacia delante, eliminación hacia atrás, esquemas y carreras de rango. |
| | *RandomSearch* | Busca aleatoriamente en el espacio de subconjuntos de atributos. Si un conjunto inicial es suministrado, busca por subconjuntos que mejoren (o igualen) el punto inicial. |
| | *RankSearch* | Clasifica los atributos usando un evaluador de atributos individual y entonces ordena los subconjuntos prometedores usando un evaluador de subconjuntos de atributos. Este procedimiento tiene una baja complejidad computacional. |
| Método de Tipificación | *Ranker* | No es un método de búsqueda de subconjuntos de atributos, sino un esquema de categorías para los atributos individuales. Ordena los atributos por sus evaluaciones individuales y debe ser usado en conjunción con uno de los evaluadores de atributos individualmente. Este procedimiento no sólo ordena los atributos sino también ejecuta la selección de atributos removiendo los de hasta abajo. |

*Tabla B.2 – Métodos de Búsqueda para la Selección de Atributos.*

A continuación se explicará brevemente, estos 2 tipos principales de métodos de búsqueda para la selección de atributos:

- **Métodos de Búsqueda:** Los métodos de búsqueda atraviesan el espacio de atributos para encontrar un buen subconjunto y un evaluador de subconjunto de atributos mide su calidad. Cada método de búsqueda puede ser configurado con el editor de objetos de Weka. Entre los métodos de búsqueda establecidos, se encuentran: búsqueda escalada codiciosa hacia delante y hacia atrás, y algunas heurísticas.
- **Método de Tipificación:** El método *'Ranker'*, no es un método de búsqueda para subconjuntos de atributos, sino un esquema de ordenamiento para los atributos

individuales. Por tanto, ordena cada atributo de acuerdo a su evaluación individual, y debe ser usado junto con algún método de evaluación de atributos individuales. Se considera con un método de selección de atributos, porque elimina o conserva, los atributos con puntaje bajo o alto, respectivamente, de acuerdo a un umbral o tope de atributos.

## ANEXO C.

### C. ANÁLISIS DE CORRELACIÓN.

El Análisis de Correlación es una técnica estadística frecuentemente usada en la Minería de Datos, en técnicas como el modelo de regresión; y en particular, es de gran utilidad en la selección de atributos. En realidad, un estudio correlacional permite, además de ayudar en la selección, comprender los datos [Hernández-Orallo et al. 2007], y poder tomar decisiones sobre los mismos.

Es posible que el investigador desee tener una medida de la intensidad de la relación lineal entre dos variables cuando se elimina la influencia de las variables restantes [Daniel 1999]. El objetivo principal del análisis de correlación lineal es medir la intensidad de una relación lineal entre dos variables. [Johnson 1990]

Dado que el análisis de correlación se desarrolla entre dos variables, en este caso llamadas atributos, y en la presente investigación se estudian conjuntos de atributos, es necesario recurrir al Análisis Estadístico Multivariado, que puede ser expresado mediante una matriz de correlación [Flores 2011]. La Matriz de Correlación se construye a través de la matriz de covarianzas $\mathbf{C}=[\boldsymbol{\sigma_{i,j}}]$, donde $\sigma_{i,j} = E\left[\left(A_{i,k} - \mu_i\right)\left(A_{j,k} - \mu_j\right)\right]$; $\mu_i = E\left(A_{i,k}\right)$ y $\mu_j = E\left(A_{j,k}\right)$.

La Matriz de Covarianzas es una matriz simétrica, es decir, $\sigma_{i,j} = \sigma_{j,i}$, en donde su diagonal contiene las varianzas $\sigma_{i,j}^2$ con $i=j$ de cada atributo. Así, la matriz de covarianza para $\boldsymbol{n}$ atributos quedaría de la siguiente manera:

$$C_{nxn} = \begin{bmatrix} \sigma_{1,1}^2 & \sigma_{1,2} & \sigma_{1,3} & \sigma_{1,4} & \cdots & \sigma_{1,n} \\ \sigma_{2,1} & \sigma_{2,2}^2 & \sigma_{2,3} & \sigma_{2,4} & \cdots & \sigma_{2,n} \\ \sigma_{3,1} & \sigma_{3,2} & \sigma_{3,3}^2 & \sigma_{3,4} & \cdots & \sigma_{3,n} \\ \sigma_{4,1} & \sigma_{4,2} & \sigma_{4,3} & \sigma_{4,4}^2 & \cdots & \sigma_{4,n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{n,1} & \sigma_{n,2} & \sigma_{n,3} & \sigma_{n,4} & \cdots & \sigma_{n,n}^2 \end{bmatrix}$$

La matriz anterior representa una relación de un conjunto finito de atributos, $X = \{A_1, A_2, A_3, \cdots, A_n\}$ consigo mismo, es decir, es una matriz cuadrada de tamaño $n \times n$, con las propiedades de reflexividad y simetría. La matriz de covarianzas se puede transformar en la Matriz de Coeficientes de Correlación $\boldsymbol{\rho_{i,j}}$, mediante la fórmula:

$$\rho_{i,j} = \frac{\sigma_{i,j}}{\sigma_i \sigma_j}$$

Donde $\sigma_{i,j}$ es la covarianza entre los atributos $\boldsymbol{i}$ y $\boldsymbol{j}$; y $\sigma_i$, $\sigma_j$ son las desviaciones estándar del total de las instancias de los atributos $\boldsymbol{i}$ y $\boldsymbol{j}$ respectivamente.

El coeficiente de correlación se denota por $\rho_{i,j}$ , donde $-1 \leq \rho_{i,j} \geq 1$; y es empleado para determinar el grado de correlación ó dependencia entre un par de elementos, en el caso del presente trabajo, entre un par de atributos, teniendo como consecuencia los siguientes tres casos:

1) Cuando $\rho_{i,j} \in (0, 1]$, significa que los atributos varían en forma directamente proporcional.
2) Cuando $\rho_{i,j} \in [-1, 0)$, significa que los atributos varían en forma inversamente proporcional.
3) Cuando $\rho_{i,j} = 0$, indica una ausencia de correlación o dependencia entre los atributos $\boldsymbol{i}$ y $\boldsymbol{j}$.

Para realizar el análisis de correlación entre el conjunto total de atributos, es necesario construir la matriz de correlación $\mathbf{T}=[\boldsymbol{\rho_{i,j}}]$, de la siguiente manera:

$$T_{nxn} = \begin{bmatrix} 1 & \rho_{1,2} & \rho_{1,3} & \rho_{1,4} & \cdots & \rho_{1,n} \\ \rho_{2,1} & 1 & \rho_{2,3} & \rho_{2,4} & \cdots & \rho_{2,n} \\ \rho_{3,1} & \rho_{3,2} & 1 & \rho_{3,4} & \cdots & \rho_{3,n} \\ \rho_{4,1} & \rho_{4,2} & \rho_{4,3} & 1 & \cdots & \rho_{4,n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{n,1} & \rho_{n,2} & \rho_{n,3} & \rho_{n,4} & \cdots & 1 \end{bmatrix}$$

Esta matriz mantiene las propiedades de simetría y reflexividad, teniendo en la diagonal principal la unidad.

## ANEXO D.

## D. TEORÍA SOBRE PORTAFOLIOS DE INVERSIÓN.

La teoría de portafolios de inversión aborda el problema de la inversión financiera, como la aportación de recursos financieros a un mercado organizado para obtener un beneficio futuro, diversificando el riesgo resultante al invertir recursos monetarios, en un conjunto de empresas que coticen en dicho mercado y obteniendo el mejor rendimiento de la inversión en dicho conjunto. [Flores 2011]

Para elaborar un portafolios de inversión se debe fijar un objetivo de rendimiento sobre una moneda de referencia y tomar en cuenta la tolerancia al riesgo, de acuerdo a la liquidez con la que se cuenta y el plazo determinado.

Un portafolios de inversión no es más que un conjunto de empresas que otorgan un porcentaje de rendimiento generado en un conjunto de empresas, en relación a un porcentaje de riesgo admitido. El *Rendimiento* "es una utilidad generada sobre una inversión, ya sea de capital o en valores"; el *Riesgo*, por su parte, "es la posibilidad de obtener un resultado distinto al que se pretendía conseguir al efectuar una acción determinada", por tanto, el riesgo es una medida de la variabilidad del rendimiento [Ross et al. 2000]. Una cartera de inversión, como también se le llama, puede tener un perfil de riesgo muy distinto del de sus componentes individuales, debido a la correlación que puede existir entre ellos. Así, la Teoría Moderna de Portafolios, tiene como objetivo obtener el mejor conjunto de empresas que tenga la mejor combinación de rendimientos de diversas inversiones para un nivel de riesgo dado. [Flores 2011]

El rendimiento esperado de una cartera es un promedio ponderado de los rendimientos esperados de los títulos individuales. Para **m** activos:

$$\overline{R}_P = W_1\overline{R}_1 + W_2\overline{R}_2 + \cdots + W_m\overline{R}_m \quad donde\ W_1 + W_2 + \cdots + W_m = 1$$

Donde $\boldsymbol{W_1}, \boldsymbol{W_2} \cdots \boldsymbol{W_m}$ son las ponderaciones de los rendimientos y, $\overline{\boldsymbol{R}}_{\boldsymbol{1}}, \overline{\boldsymbol{R}}_{\boldsymbol{2}} \cdots \overline{\boldsymbol{R}}_{\boldsymbol{m}}$ son los rendimientos promedio de las **m** empresa.

## D.1 El Modelo Matemático del Portafolio de Inversión.

## [Ochoa & Sandra 2008]

Maximizar el Rendimiento del Portafolio.

a) $\boldsymbol{Max\ E[R_p]} = \sum_{i=1}^{n} \boldsymbol{w_i E[R_i]}$

Sujeto a $\boldsymbol{\sigma_p^2} = \sum_{i=1}^{n} \sum_{j=1}^{n} \boldsymbol{w_i w_j \rho_{ij} \sigma_i \sigma_j} = \boldsymbol{cte}$

$\sum_{i=1}^{n} \boldsymbol{w_i} = \boldsymbol{1}$

$\boldsymbol{w_i} \geq \boldsymbol{0} \qquad \boldsymbol{para\ i} = \boldsymbol{1},\ \boldsymbol{2}, \cdots, \boldsymbol{n}$

La primera restricción, expresa la condición de lograr el nivel de riesgo tolerado. La segunda, conocida como limitación de presupuesto requiere que el total del presupuesto sea invertido en el portafolio. Y la última, conocida como condiciones de no negatividad, expresa que no son permitidas las ventas en corto, es decir, no se permite dar o pedir prestado dinero, y:

Minimizar el Riesgo del Portafolio.

b) $\boldsymbol{Min\ \sigma_p^2} = \sum_{i=1}^{n} \sum_{j=1}^{n} \boldsymbol{w_i w_j \rho_{ij} \sigma_i \sigma_j}$

Sujeto a $\boldsymbol{E[R_p]} = \sum_{i=1}^{n} \boldsymbol{w_i E[R_i]} = \boldsymbol{R^*}$

$\sum_{i=1}^{n} \boldsymbol{w_i} = \boldsymbol{1}$

$\boldsymbol{w_i} \geq \boldsymbol{0} \quad \boldsymbol{para\ i} = \boldsymbol{1}, \boldsymbol{2}, \cdots, \boldsymbol{n}$

La primera restricción, expresa la condición de lograr el nivel de rendimiento esperado. Las otras restricciones son igual al modelo (a).

# ANEXO E.

## E. GLOSARIO.

| Palabra | Referencia | Definición |
|---|---|---|
| **Agentes Inteligentes** | Wikipedia | Es una entidad capaz de percibir su entorno, procesar tales percepciones y responder o actuar dentro de su entorno de manera racional, es decir, de manera correcta y tendiendo a maximizar un resultado esperado. |
| **Agrupamiento** | Hernández-Orallo et al. 2007 | Clustering |
| **Alelo** | Wikipedia | Es cada una de las formas alternativas que puede tener un gen, que se diferencian en su secuencia y que se puede manifestar en modificaciones concretas de la función de ese gen. |
| **Algoritmos de Búsqueda Estocástica** | Coleman 1986 | Son algoritmos que realizan la búsqueda de soluciones en el espacio de estados, de forma aleatoria, es decir, sin un orden o dependencia entre un resultado y el anterior. Estos algoritmos son aplicables a cualquier sistema que comprenda variabilidad al azar con el transcurso del tiempo. |
| **Algoritmos Genéticos** | Eshelman 1997 | Son métodos heurísticos de búsqueda inspirados en lo que sabemos acerca del proceso de la evolución natural, son apropiados para resolver problemas donde el dominio de la solución pueda resultar demasiado extenso. Estos algoritmos utilizan una codificación adecuada de las posibles soluciones de un problema, un método de selección de acuerdo a un valor de aptitud y aplican los operadores de mutación y cruza como medidas para aplicar la variación aleatoria |
| **Almacén de Datos** | Hernández-Orallo et al. 2007 | Es un conjunto de datos históricos, internos o externos, y descriptivos de un contexto o área de estudio, que están integrados y organizados de tal forma que permiten aplicar eficientemente herramientas para resumir, describir y analizar los datos con el fin de ayudar en la toma de decisiones estratégicas. |
| **Análisis de Componentes Principales** | Hernández-Orallo et al. 2007 | También llamado "Método de Karhunen-Loeve", forma parte del conjunto de técnicas estadísticas del Análisis Multivariado clásico para tratar con un conjunto de variables, que tienen sus orígenes en el Álgebra Lineal y la Geometría. Consiste en un cambio de ejes en la representación de los atributos o variables originales, donde los nuevos atributos se generan de tal manera que sean independientes sí, y además, que los primeros tengan más relevancia que los últimos. |
| **Análisis de la Correlación** | Hernández-Orallo et al. 2007 | Es el análisis de una Matriz de Correlaciones entre los atributos, utilizando la correlación entre los diferentes pares de ellos. Si la correlación es positiva y cercana a uno, los pares de atributos son directamente dependientes. Si la correlación es negativa y cercana a -1, los pares de atributos son inversamente dependientes, y si es cercana a 0, casi no son dependientes entre sí. |
| **Aprendizaje de Máquina** | Mitchell 1997 | Un programa de computadora se dice que aprende de una experiencia E con respecto a una clase de tarea T y una medida de ejecución P, si su ejecución en la tarea T, medida por P, mejora con la experiencia E. |
| **Aptitud (Fitness)** | Freitas 2002 | Es una medida de desempeño que mide la calidad de una solución. |

| | | |
|---|---|---|
| **Árbol de Decisión** | Freitas 2002 | Es una estructura de representación del conocimiento que consiste de nodos y ramas organizados en forma de árbol, que permite tomar decisiones siguiendo de manera ordenada y esquematizada las ramas del árbol. |
| **Árbol de Decisión M5P** | Frank et al. 2005 | Es un árbol de inducción para modelos de regresión, que combina un árbol de decisión con una función de regresión lineal, permitiendo la poda como optimizador |
| **Árbol de Decisión NBT** | Frank et al. 2005 | Es un árbol de decisión híbrido, que clasifica en las hojas del árbol los registros, por medio de la probabilidad condicional de Naïve Bayes. |
| **Arquetipo** | Wikipedia | Es el patrón ejemplar del cual otros objetos, ideas o conceptos se derivan. En biología, son los tipos primitivos y originarios a partir de la cual se produce y determina la diversidad orgánica. |
| **Arquitectura de los Almacenes de Datos** | Propia | Es la técnica que se enfoca en el diseño, construcción y mantenimiento de los Almacenes de Datos |
| **Atributo** | Date 1986; Korth & Silberschatz 1988; Wikipedia | Son todas las características que definen y describen a una entidad, es decir, las columnas de una tabla, dentro de una Base de Datos. Una Entidad es un objeto que existe y puede distinguirse de otros objetos. Una entidad está representada por un conjunto de atributos. un atributo es una especificación que define una propiedad de un Objeto, elemento o archivo. También puede referirse o establecer el valor específico para una instancia determinada de los mismos. |
| **Atributo Categórico** | Pyle 1999 | Atributo Nominal |
| **Atributo Irrelevante** | Hernández-Orallo et al. 2007 | Desde el punto de vista de Minería de Datos, es un atributo que no aportará ningún tipo de información al proceso de extracción de conocimiento dentro de una BD, y que hay que eliminar con la finalidad de no 'entorpecer' ó 'distraer' el proceso de extracción de conocimiento. |
| **Atributo Nominal** | Pyle 1999 | Son los atributos medidos a través de una escala nominal, la cual trata de medir los valores que el atributo tomará, a través de nombres, categorías o etiquetas que describen el comportamiento de esta variable. |
| **Atributo Numérico** | Wikipedia | Son los atributos que son descritos a través de valores numéricos. |
| **Atributo Redundante** | Hernández-Orallo et al. 2007 | Desde el punto de vista de Minería de Datos, es el atributo que aporta la misma información al proceso de extracción de conocimiento dentro de una BD, y que hay que eliminar en beneficio del proceso. |
| **Cardinalidad** | Wikipedia | Indica el número o cantidad de elementos de un conjunto, ya sea finita o infinita, constituyendo una generalización del conjunto de números Enteros. |
| **Clasificación** | Hernández-Orallo et al. 2007 | Es la tarea de Minería de Datos más usada; en ella, cada instancia pertenece a una clase, la cual se indica mediante el valor de un atributo. Su objetivo es predecir la clase de nuevas instancias de las que se desconoce la clase, de acuerdo con las instancias clasificadas anteriormente. |
| **Clustering** | Hernández-Orallo et al. 2007 | También conocido como 'Agrupamiento', es la tarea descriptiva por excelencia y consiste en obtener grupos 'naturales' a partir de los datos. Los datos son agrupados basándose en el principio de maximizar la similitud entre los elementos de un grupo, minimizando la similitud entre los distintos grupos. |

| | | |
|---|---|---|
| **Computación Evolutiva** | Fogel 1997, Rodríguez-Vázquez et al. 2007 | Es el paradigma de computación que fue definido como tal en 1991, y representa un esfuerzo para juntar a los investigadores que han seguido diferentes aproximaciones para simular varios aspectos de la evolución. Sus principales técnicas envuelven la reproducción, variación aleatoria, competitividad y selección de individuos contendientes dentro de una población. Esta inspirada en los mecanismos de la evolución natural, a la cual imita parcialmente. |
| **Computación Paralela** | Wikipedia | Es una técnica de programación en la que muchas instrucciones se ejecutan simultáneamente. Se basa en el principio de que los problemas grandes se pueden dividir en partes más pequeñas que pueden resolverse de forma concurrente ("en paralelo"). Existen varios tipos de computación paralela: paralelismo a nivel de bit, paralelismo a nivel de instrucción, paralelismo de datos y paralelismo de tareas. |
| **Control Automático** | Mitchell 1997 | Procedimientos que aprenden a controlar procesos para optimizar objetivos predefinidos, y que aprenden a predecir el siguiente estado del proceso que están controlando. |
| **Correlaciones** | Hernández-Orallo et al. 2007 | Es la tarea descriptiva de Minería de Datos que se usa para examinar el grado de similitud de los valores de dos variables numéricas, analizando el comportamiento del coeficiente de correlación *r*, es decir, si *r* es positivo, las variables tienen un comportamiento similar y cuando es negativo, su comportamiento es opuesto. |
| **Cromosoma** | Wikipedia | Son estructuras organizadas del ADN de un individuo que contienen los genes, elementos regulatorios y otras secuencias de nucleótidos. |
| **Cruza** | Wikipedia | Es el operador de recombinación o reproducción y es el más importante dentro de un Algoritmo Genético. Representa la reproducción sexual, operando sobre dos cromosomas a la vez para generar dos descendientes, donde se combinan las características de ambos cromosomas padres. |
| **Dato Anómalo** | Hernández-Orallo et al. 2007 | También llamados *Outliers.* Es un dato correcto pero puede estar fuera del rango usual de valores que un atributo o característica tiene, es decir, estos valores no se ajustan al comportamiento general de los datos |
| **Dato Incorrecto** | Hernández-Orallo et al. 2007 | Son datos incorrectos que posiblemente se deben a una captura errónea de los mismos, o a la falta de restricciones dentro de los criterios de captura y/o vaciado de los campos dentro de una BD. |
| **Data Survey** | Hernández-Orallo et al. 2007 | Es el proceso de reconocimiento y exploración de los datos. Su objetivo es conocer los datos a través de diferentes técnicas como son: la visualización, agregación, pivotamiento y descripción generalizada, entre otros. |
| **Data Warehouse** | Hernández-Orallo et al. 2007 | Es un almacén de datos o repositorio de fuentes heterogéneas de datos, integrados y organizados bajo un esquema unificado para facilitar su análisis y dar soporte a la toma de decisiones. Incluye operaciones de procesamiento analítico en línea (OLAP). |
| **Datamart** | Hernández-Orallo et al. 2007 | Es una forma de organizar la información contenida dentro de un almacén de datos. La idea general es que para cada subámbito de la organización se va a construir una estructura de estrella. Por tanto, el almacén de datos estará formado por muchas estrellas, jerárquicas o no, formando una constelación y cada una de sus estrellas son llamadas 'Datamart'. |

| | | |
|---|---|---|
| **Dimensionalidad en una tabla** | Hernández-Orallo et al. 2007 | La dimensionalidad de una tabla se encuentra en función del número de atributos o conjunto de atributos *n* de una tabla o Vista Minable. |
| **Discretización** | Hernández-Orallo et al. 2007 | Es la conversión de un valor numérico en un valor nominal ordenado. No obstante, el orden del atributo nominal puede ser preservado y utilizado por los pasos subsiguientes o bien puede olvidarse y tratarse el atributo como un valor nominal sin orden. |
| **Distribución de Probabilidad** | Miller & Freud 1987 | Es la función que muestra la probabilidad de una variable aleatoria, ya sea entera o continua, de que tome cualquier valor de su rango. |
| **Dominio de una solución** | Royden 1968 | Dado que 'f' es una función de 'X' en 'Y' f : X->Y, el conjunto X es llamado el Dominio de una función 'f' y contiene todos los valores posibles que 'x' puede tomar dentro de la función. |
| **Eficaz /Eficacia** | Diccionario de la Real Academia Española | Capacidad de lograr el efecto que se desea o espera. |
| **Eficiencia** | Diccionario de la Real Academia Española - RAE | Capacidad de disponer de alguien o de algo para conseguir un efecto determinado. Lograr cumplir el objetivo con el mínimo de tiempo, materiales y esfuerzo. |
| **Enfoque de Envolvimiento** | Freitas 2002 | En este enfoque, la selección de atributos es ejecutada tomando en cuenta el algoritmo de clasificación que será aplicado a los atributos seleccionados; por tanto, su meta es seleccionar un subconjunto de atributos que sea optimizado por un algoritmo de clasificación dado. |
| **Enfoque de Filtrado** | Freitas 2002 | En este enfoque, la selección de atributos es ejecutada sin tomar en cuenta el algoritmo de clasificación que será aplicado a los atributos seleccionados. Así, su objetivo es seleccionar un subconjunto de atributos que preserven, tanto como sea posible, la información relevante. |
| **Error Cuadrático Medio** | Hernández-Orallo et al. 2007 | Se utiliza para evaluar la salida de un modelo, y se obtiene elevando al cuadrado el error del valor predicho respecto al valor que se utiliza como validación. Esto promedia los errores y toma más en cuenta aquellos errores que se desvían más del valor predicho. |
| **Error de la Muestra** | Mitchell 1997 | El error de la muestra de una hipótesis con respecto a alguna muestra S de instancias obtenidas a partir de X, es la fracción de S que es clasificada erróneamente. |
| **Error Verdadero** | Mitchell 1997 | Es lo que se desearía conocer acerca de una determinada hipótesis. Sin embargo, el error de la muestra es todo lo que se puede conocer acerca de la precisión de una hipótesis, dado que generalmente se desconoce la función objetivo ***f***. |
| **Espacio de Búsqueda** | Radcliffe 1997 | Un espacio de búsqueda *S*, es un conjunto de objetos a ser considerados durante la búsqueda. Puede ser finito, infinito, continuo o discreto. La meta de un problema de búsqueda es encontrar uno o más puntos en el espacio de búsqueda teniendo alguna característica específica. |
| **Estadística** | Johnson 1990 | Es la metodología de la extracción de Información de los datos, su correlación y la expresión de la incertidumbre en las decisiones que tomamos. |

| | | |
|---|---|---|
| **Estrategias de Evolución** | [Rudolph 1997] | Es un algoritmo de Computación Evolutiva que consiste en un individuo **a** (padre) es mutado añadiendo un vector aleatorio con distribución normal, que es multiplicado por un escalar. El nuevo punto (hijo) es aceptado si es mejor o igual que el viejo (si el hijo es mejor o igual al padre), de otra forma, el viejo punto pasa a la siguiente iteración. La decisión para la selección está basada en una simple comparación entre los valores de los puntos viejo y nuevo, de la función objetivo, a éste tipo de selección se le llama extintiva, porque los peores individuos tienen una probabilidad 0 de ser seleccionados. |
| **Evolución** | Fogel 1997a | Puede ser visto como un proceso de optimización, ya que es vista como la inevitable salida o resultado de la interacción de 4 procesos esenciales: Reproducción, competición, mutación y selección. También se conoce como Evolución a la adaptación de una especie a su ambiente. |
| **Fenotipo** | Rodríguez-Vázquez et al. 2007 | Es el conjunto de características físicas, fisiológicas, etológicas y del medio ambiente contenidas en el ADN de cada individuo, que diferencian a una especie de otra. |
| **Función de Aptitud** | Mitchell 1997 | Función de un GA que asigna una evaluación de la aptitud del algoritmo, de acuerdo a una hipótesis dada. |
| **Función Objetivo** | Wikipedia | Es la función a ser optimizada (maximizada ó minimizada) que va a evaluar la solución candidata de un problema computacional dado. |
| **Gen** | Rodríguez-Vázquez et al. 2007 | Son secciones dentro de la larga molécula de ADN ensamblados en estructuras denominadas cromosomas, que contienen información para una determinada característica del fenotipo y que pueden codificar cada uno de los veinte aminoácidos para ensamblar una proteína. |
| **Genotipo** | Wikipedia, Rodríguez-Vázquez et al. 2007 | Es la totalidad de información genética que un organismo hereda de sus padres, es decir, son todos los genes que se encuentran contenidos en los cromosomas de un individuo de una especie en particular. |
| **GNU** | Wikipedia | La Licencia Pública General, es una licencia publicada por la Fundación de Software Gratuita 'FSF', que otorga a todos los destinatarios de un programa, el derecho de correr, copiar modificar y distribuir dicho software, y al mismo tiempo, prohíbe el imponer restricciones sobre copias futuras que ellos distribuyan. |
| **Heurística** | Wikipedia | Es la capacidad de un sistema para realizar de forma inmediata innovaciones positivas para sus fines. La capacidad heurística es un rasgo característico de los humanos desde cuyo punto de vista puede describirse como el arte y la ciencia del descubrimiento y de la invención o de resolver problemas mediante la creatividad y el pensamiento lateral o pensamiento divergente. |
| **Histograma** | Wikipedia, Miller & Freud 1987 | Es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados. En el eje vertical se representan las frecuencias, y en el eje horizontal los valores de las variables, normalmente señalando las marcas de clase, es decir, la mitad del intervalo en el que están agrupados los datos. / Son gráficas de dos ejes rectangulares que describen el comportamiento de dos o más variables aleatorias representadas en el eje 'x', en relación con el valor de una función, representada en el eje 'y'. |
| **IDE** | Netbeans | Es un conjunto de programas que le permiten al usuario desarrollar programas. Integrated Development Environment. |

| | | |
|---|---|---|
| **Independencia de la Muestra** | Johnson 1990 | La independencia de una muestra está determinada por las fuentes utilizadas para obtener los datos. Si se utiliza el mismo conjunto de fuentes, o conjuntos relacionados, para obtener los datos que representan a ambas poblaciones, se tiene un muestreo dependiente. Si se utilizan conjuntos de fuentes no relacionados, un conjunto para cada población, se tiene un muestreo independiente. |
| **Instancia** | Korth & Silberschatz 1988 | En Bases de Datos es la información almacenada en una celda de una tabla relacional. |
| **Inteligencia Artificial** | Mitchell 1997; Wikipedia | Aprendizaje de las representaciones simbólicas de conceptos - Es la ciencia e ingeniería de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes. |
| **Intervalos de Confianza** | Wikipedia | En estadística, es un par de valores, uno más grande que el otro, que determinan un intervalo, entre los cuáles se estima que se encuentra cierto valor desconocido con una determinada probabilidad de acierto. |
| **KDD** | Hernández-Orallo et al. 2007 | Es un proceso iterativo e interactivo cuyo objetivo es el descubrimiento de conocimiento valioso, útil y comprensible de una base de datos. |
| **Máquinas de Estado Finito** | Wikipedia | Es un modelo matemático que realiza cómputos en forma automática sobre una entrada para producir una salida. Este modelo está conformado por un alfabeto, un conjunto de estados y un conjunto de transiciones entre dichos estados. Su funcionamiento se basa en una función de transición, que recibe a partir de un estado inicial una cadena de caracteres pertenecientes al alfabeto (la entrada), y que va leyendo dicha cadena a medida que el autómata se desplaza de un estado a otro, para finalmente detenerse en un estado final o de aceptación, que representa la salida. |
| **Máxima Verosimilitud** | Hernández-Orallo et al. 2007; Wikipedia | En estadística, la estimación por máxima verosimilitud es un método habitual para ajustar un modelo y encontrar sus parámetros, buscando aquellos que hacen más verosímiles los datos recogidos dada una hipótesis. |
| **Medidas Cualitativas** | Wikipedia | Son las medidas de una variable cualitativa, es decir, las variables que expresan distintas cualidades, características o modalidad. Cada modalidad que se presenta se denomina atributo o categoría y la medición consiste en una clasificación de dichos atributos. |
| **Medidas Cuantitativas** | Wikipedia | Son las medidas de una variable cuantitativa, es decir, las variables que se expresan mediante cantidades numéricas, que pueden ser discretas o continuas. |
| **Métodos Basados en Casos** | Hernández-Orallo et al. 2007 | Son métodos de aprendizaje automático que tratan de resolver un problema a partir de información extraída de un conjunto de ejemplos existentes previamente. Los ejemplos son los que aportarán la información necesaria para poder predecir el comportamiento de un nuevo dato no perteneciente al conjunto de ejemplos. |
| **Métodos de Búsqueda Estocástica** | Wikipedia | Son métodos de optimización que generan y usan variables aleatorias. Para problemas estocásticos, las variables aleatorias aparecen en la formulación de un problema de optimización en sí mismo y envuelven funciones objetivo aleatorias. Además, incluye métodos con iteraciones aleatorias. Este tipo de métodos introducen la aleatoriedad dentro de los procesos de búsqueda para acelerar el progreso, haciendo el método menos sensitivo a los errores de modelado. |

| | | |
|---|---|---|
| **Métodos de Envolvimiento** | Hernández-Orallo et al. 2007 | También llamados Métodos basados en modelo, son métodos para la selección de características donde la bondad de la selección misma se evalúa respecto a la calidad de un modelo de minería de datos o estadístico, extraído a partir de los datos. Este tipo de técnicas requieren mucho más tiempo que las otras, ya que para evaluar hay que entrenar un modelo. Además, el método de minería de datos utilizado para la selección de características no tiene que ser el mismo, que el que se utilizará finalmente. |
| **Métodos de Filtrado** | Hernández-Orallo et al. 2007 | También llamados Métodos Previos, son métodos para la selección de características que filtran los atributos irrelevantes antes de cualquier proceso de Minería de Datos y, en cierto modo, independiente de él. Las técnicas son fundamentalmente estadísticas y el criterio para establecer el subconjunto de características "óptimo" se basa en medidas de calidad previa que se calculan a partir de los datos mismos. |
| **Métodos de Minería de Datos** | Maimon & Rokach 2005a | Son las técnicas que permiten resolver la tarea, por ejemplo: Árboles de Decisión o Redes Neuronales. |
| **Métodos Heurísticos de Búsqueda** | Wikipedia | Se refiere a un grupo de técnicas basadas en la experiencia para resolver problemas, aprender y descubrir. Donde una búsqueda exhaustiva es impráctica, los métodos heurísticos son usados para acelerar el proceso de encontrar una solución satisfactoria. Además este tipo de técnicas permiten explorar más ampliamente el dominio del problema. |
| **Minería de Datos** | Hernández-Orallo et al. 2007 | Es un conjunto de técnicas que tienen como objetivo analizar los datos para extraer conocimiento (patrones, reglas, clasificaciones, etc) de una Base de Datos. |
| **Mínimos Cuadrados** | Miller & Freund 1987 | Es un método estadístico que estima ***y*** a través de la ecuación de la línea que mejor ajusta un conjunto dado de datos apareados. Sus estimadores ***a*** y ***b*** tienen una varianza muy pequeña, es decir, están sujetos a variaciones aleatorias muy pequeñas, por lo que son muy confiables. |
| **Modelo de Dependencias** | Hernández-Orallo et al. 2007 | Tarea de Minería de Datos que enfatiza la dependencia entre los atributos y suele utilizar un espacio de búsqueda mucho más grande. Aquí, cualquier regla descubierta puede contener cualquier subconjunto de atributos, además cualquier atributo puede aparecer tanto en la premisa del antecedente como en la premisa del precedente, pero no en ambas partes de la misma regla. |
| **Modelo Relacional de Base de Datos** | Korth & Silberschatz 1988 | También llamado "Modelo Entidad-Relación", se basa en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados *entidades*, y de las *relaciones* entre esos objetos. Una Entidad es un objeto que existe y puede distinguirse de otros; la distinción se logra asociando a cada entidad un conjunto de atributos que describen al objeto. Una Relación es una asociación entre varias entidades. |
| **Modelos Descriptivos de Minería de Datos** | Hernández-Orallo et al. 2007 | Identifican patrones que explican o resumen los datos, es decir, sirven para explorar las propiedades de los datos examinados, no para predecir nuevos datos. |
| **Modelos Predictivos de Minería de Datos** | Hernández-Orallo et al. 2007 | Pretenden estimar valores futuros o desconocidos de variables objetivo o dependientes, usando otras variables o campos de la base de datos (variables independientes). |

| | | |
|---|---|---|
| **Muestra Aleatoria Simple** | Johnson 1990 | Es aquella seleccionada de manera que cada elemento de la población tiene la misma probabilidad de ser seleccionada. De manera equivalente, todas las muestras de tamaño *n* tienen la misma probabilidad de ser seleccionadas. |
| **Mutación** | Rodríguez-Vázquez et al. 2007 | Es un cambio aleatorio en la secuencia de bases en el ADN, de los cuales, la minoría son benéficos; sin embargo, estos pueden aportar variedad genética a una especie, con lo cual es posible obtener nuevas características fenotípicas. En los Algoritmos Genéticos, la mutación permite explorar ampliamente el conjunto de soluciones. |
| **Niveles de Agregación** | Wikipedia | Es una abstracción a través de la cual las relaciones se tratan como entidades de un nivel más alto. Se utiliza para expresar relaciones entre relaciones o entre entidades y relaciones. Se representa englobando la relación abstraída y las entidades que participan en ella en un rectángulo. |
| **Numerización** | Hernández-Orallo et al. 2007 | Es el proceso inverso a la discretización, y puede ser extremadamente útil cuando el método de Minería de datos que se va a utilizar no admite datos nominal. |
| **Optimización Combinatoria** | Wikipedia | Es una rama de la optimización en matemáticas aplicadas y en ciencias de la computación, relacionada a la investigación de operaciones, teoría de algoritmos y teoría de la complejidad computacional. También está relacionada con otros campos, como la inteligencia artificial e ingeniería de software. Los algoritmos de optimización combinatoria resuelven instancias de problemas que se creen ser difíciles en general, explorando el espacio de soluciones (usualmente grande) para estas instancias. Los algoritmos de optimización combinatoria logran esto reduciendo el tamaño efectivo del espacio, y explorando el espacio de búsqueda eficientemente. |
| **Outliers** | Hernández-Orallo et al. 2007 | Son los valores de una instancia que no se ajustan al comportamiento general de los datos de un atributo en particular. |
| **Pivotamiento** | Hernández-Orallo et al. 2007 | Es una operación OLAP de preparación de datos de una Vista Minable, la cual consiste en cambiar sus filas por sus columnas. |
| **Portafolios de Inversión** | Ross et al. 2000 | También llamado *Cartera*, es una postura combinada de un inversionista que incluye a más de una acción, de un bono, de un active de un bien raíz, u otro active. |
| **Precisión** | Wikipedia | Es la capacidad de una técnica/modelo de dar un resultado adecuado durante una periodo de prueba o un experimento, realizadas bajo las mismas condiciones. En Minería de Datos, es un parámetro muy importante para medir la predicción de clases. |
| **Preparación de los Datos** | Hernández-Orallo et al. 2007 | Es la etapa anterior a la de Minería de Datos, dentro del proceso de KDD, donde se busca mejorar la calidad de los datos de una Vista Minable para poder aplicar las técnicas en busca de patrones coherentes, comprensibles, novedosos e interesantes. |
| **Procedimientos** | Wikipedia | Un procedimiento es un conjunto de acciones u operaciones que tienen que realizarse de la misma forma, para obtener siempre el mismo resultado bajo las mismas circunstancias (por ejemplo, procedimiento de emergencia). |
| **Proceso** | Wikipedia | Un proceso es un conjunto de actividades o eventos (coordinados u organizados) que se realizan o suceden (alternativa o simultáneamente) bajo ciertas circunstancias con un fin determinado. |

| | | |
|---|---|---|
| **Programación Evolutiva** | Porto 1997 | Genera incremental e iterativamente, soluciones apropiadas desde un entorno de cambios estáticos y dinámicos, a través de vectores de valores reales; es decir, desarrolla un conjunto de soluciones que exhibe el comportamiento óptimo bajo un entorno y una función de pago deseada. Pertenece al Paradigma de Algoritmos Evolutivos. |
| **Query** | Wikipedia, Jaramillo 2003 | Es una petición exacta de recuperación de información desde una base de datos o un sistema de información. Es una consulta a una base de datos. |
| **Razonamiento Aproximado** | Wikipedia | Es una capacidad del razonamiento humano por la cual es capaz de obtener conclusiones útiles a partir de información incompleta o con cierto grado de incertidumbre. La lógica tradicional se fundamenta en los métodos de razonamiento deductivo en los que no se contempla que tanto la información de entrada como las propias reglas puedan no ser ciertas con carácter absoluto. Tampoco es capaz de tratar información vaga o imprecisa. Dentro del área de la inteligencia artificial, se han desarrollado modelos teóricos que simulan la capacidad humana para realizar razonamiento aproximado. Entre los más conocidos están la lógica difusa y los métodos bayesianos. |
| **Razonamiento Basado en Casos** | Hernández-Orallo et al. 2007 | Es un modelo computacional de razonamiento por analogía, basado en casos históricos existentes. Una de sus premisas es que gran parte de las soluciones a problemas encontradas por los especialistas, son variaciones sobre un problema tipo. |
| **Recombianción Genética** | Rodríguez-Vázquez et al. 2007 | Es el proceso en el cual se intercambia parcialmente la información genéticas de los individuos de una especie, para diversificar el material genético de ésta. En la recombinación, se unen dos cromosomas homólogos mediante estructuras proteicas para generar un nuevo individuo que contiene algo del material genéticos de los padres. Como resultado, se obtienen las variaciones de la población y su diversificación. |
| **Redes Neuronales Artificiales** | Mitchell 1997 | Es un paradigma de aprendizaje de máquina que inspirado, en parte, por las observaciones de que los sistemas de aprendizaje biológicos se encuentran construidos sobre redes de neuronas interconectadas muy complejas. Las RNA están construidas sobre un conjunto densamente interconectado de unidades simples, donde cada unidad toma un cierto número de entradas y produce una salida en particular |
| **Reglas de Asociación** | Hernández-Orallo et al. 2007 | También es una tarea descriptiva de Minería de Datos, que tiene como objetivo identificar relaciones no explícitas entre atributos categóricos, aunque no impliquen una relación causa-efecto. La formulación más común es: "Si el atributo *X* toma el valor de *d* entonces el atributo *Y* toma el valor de *b*". |
| **Reglas de Predicción** | Freitas 2002 | Es una representación del conocimiento de la forma: IF (condición_1 y ... condición_i … y condición_m) ENTONCES (Predicción) |
| **Regresión** | Hernández-Orallo et al. 2007 | Es una tarea de Minería de Datos que consiste en aprender una función real que asigna a cada instancia un valor numérico real; ésta es la principal diferencia con respecto a la clasificación; el valor a predecir es numérico. |
| **Regresión Logística** | Frank et al. 2005 | Construye un modelo de Regresión Logística Lineal, usada para lidiar con modelos logísticos (0,1). |
| **Rendimiento** | Ross et al. 2000 | Utilidad generada sobre una inversión de capital o sobre una inversión en valores. |

| | | |
|---|---|---|
| **Reproducción** | Wikipedia, Rodríguez-Vázquez et al. 2007 | Es un proceso biológico que permite la creación de nuevos organismos, transmitiendo su información genética a su progenie; existen dos tipos de reproducción: la sexual y la asexual. |
| **Ruido** | Wikipedia, Freitas 2002 | Es un proceso estocástico o señal aleatoria que se caracteriza por el hecho de que sus valores de señal en dos tiempos diferentes no guardan correlación estadística, por lo tanto su gráfica es plana. El ruido coloreado es aquel que si guarda correlación estadística. El ruido es un caso particular de la caminata aleatoria. / En Minería de Datos, usualmente se refiere a los errores en los datos producidos al recopilarlos o introducirlos dentro de la Base de Datos. |
| **Selección** | Rodríguez-Vázquez et al. 2007 | Es el operador de un Algoritmo Genético que se encarga de favorecer a los individuos mejor calificados, teniendo más posibilidades de ser seleccionados y pasar a la siguiente población. Algunos métodos de selección son: Ruleta, Estocástico Universal y Torneo. |
| **Selección de Características** | Hernández-Orallo et al. 2007 | Es una tarea del Proceso de Extracción de Conocimiento de las Bases de Datos, perteneciente a su segunda fase que busca obtener el subconjunto de atributos o características más significativos para una tarea específica de Minería de Datos. |
| **Simulación** | Shannon & Johannes 1976, Wikipedia | Es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias -dentro de los límites impuestos por un cierto criterio o un conjunto de ellos - para el funcionamiento del sistema |
| **Sistemas de Información** | Wikipedia | Es un conjunto de elementos enfocados al tratamiento y administración de datos e información, generados para cubrir un objetivo. |
| **Sobreajuste** | Freitas 2002 | Dado una tarea de clasificación ejecutada sobre el conjunto de prueba, los datos estarán *sobreajustados*, si muestran una solución que parece correcta, pero que no lo es. |
| **Subajuste** | Freitas 2002 | Dada una tarea de clasificación ejecutada sobre el conjunto de prueba, los datos estarán *subajustados*, si muestran una solución que parece incorrecta, pero que no lo es. |
| **Tarea de Minería de Datos** | Maimon & Rokach 2005a | Es un tipo de problema de Minería de Datos, por ejemplo: Clasificar las clases de un atributo de acuerdo a un valor específico. |
| **Tareas Descriptivas (Algoritmos Descriptivos)** | Maimon & Rokach 2005a | También llamadas Minería de Datos No Supervisada. El carácter descriptivo nos ayuda a la comprensión de los eventos en el presente mediante la descripción de estos. |
| **Tareas Predictivas (Algoritmos Predictivos)** | Maimon & Rokach 2005a | También llamada Minería de Datos Supervisada. El carácter predictivo nos sirve para prever el comportamiento futuro de algún tipo de entidad, obteniendo como valores de salida una clase, categoría, valor numérico o un orden entre ellos, sobre los cuáles se tomarán decisiones a futuro |
| **Técnicas de Optimización** | Wikipedia | Son las técnicas de programación matemática que resuelven un tipo general de problemas matemáticos, donde se desea elegir el mejor entre un conjunto de elementos. Un problema de optimización trata entonces de tomar una decisión óptima para maximizar (ganancias, velocidad, eficiencia, etc.) o minimizar un criterio determinado (costos, tiempo, riesgo, error, etc.), contando con restricciones que implican que no cualquier decisión es posible. |

| | | |
|---|---|---|
| **Toma de Decisiones** | Wikipedia | Es el proceso en el cual se realiza una elección entre las opciones o formas para resolver un problema en particular, de cualquier índole, utilizando metodologías y herramientas que apoyen el análisis de las diferentes alternativas de acción, a fin de elegir la mejor para la situación dada. |
| **Valores Ausentes (Missing Values)** | Hernández-Orallo et al. 2007 | Campos vacíos dentro de una BD. |
| **Vecino más cercano** | Freitas 2002 | Es un algoritmo de búsqueda aleatoria que clasifica las instancias de los datos usando los datos disponibles. Dicha búsqueda se realiza alrededor de las instancias más cercanas ó más similares. |
| **Vista Minable** | Hernández-Orallo et al. 2007 | Es una tabla relacional que reúne toda la información requerida para una tarea particular de Minería de Datos, desde diferentes tablas o bases de datos, posiblemente a través de un query. |
| **Volatilidad en los datos** | Wikipedia | Es una medida de la frecuencia e intensidad de las variaciones entre diferentes tipos y valores de datos |
| **WEKA** | Frank et al. 2005 | El Ambiente de Waikato para el Análisis del Conocimiento 'WEKA', es una colección organizada de los algoritmos de vanguardia más sofisticados de Aprendizaje de Máquina y de las herramientas de Preprocesamiento de datos, escritos en Java bajo los términos de la licencia de GNU. |
| **World Wide Web Consortium** | Wikipedia | El W3C, por sus siglas en inglés, es la principal organización internacional de estándares, para la Red Informática Mundial WWW, en donde desarrollan estándares para ésta. |

# ANEXO F.

## F. DICCIONARIO DE DATOS.

Se presenta a continuación el diccionario de datos para la Base de Datos de Portafolios de Inversión:

| TableName | FieldName | Type | Length | Decimals | Nulls | Description | Minimum | Maximum |
|---|---|---|---|---|---|---|---|---|
| BD_Portafolio.csv | Date | Date | dd/mm/aa | 0 | 0 | En días | 01/07/08 | 29/06/12 |
| BD_Portafolio.csv | 1) ALFAA.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.144638404 | 0.256559767 |
| BD_Portafolio.csv | 2) ALSEA.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.131648936 | 0.242424242 |
| BD_Portafolio.csv | 3) AMXL.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.100640439 | 0.133742331 |
| BD_Portafolio.csv | 4) ARA.MX | Decimal | 20 | 17 | 0 | Rendimiento | -0.2112150 | 0.281990521 |
| BD_Portafolio.csv | 5) AUTLANB.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.270771408 | 0.288905298 |
| BD_Portafolio.csv | 6) AXTELCPO.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.153381643 | 0.206043956 |
| BD_Portafolio.csv | 7) BACHOCOB.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.444444444 | 0.589403974 |
| BD_Portafolio.csv | 8) BIMBO.MX | Decimal | 21 | 18 | 1 | Rendimiento | -0.09726637 | 0.139517345 |
| BD_Portafolio.csv | 9) CEMEXCPO.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.189496097 | 0.268806419 |
| BD_Portafolio.csv | 10) CMOCTEZ.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.260689655 | 0.349502488 |
| BD_Portafolio.csv | 11) COMERCIUBC.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.739759036 | 0.549828179 |
| BD_Portafolio.csv | 12) CYDSASAA.MX | Decimal | 21 | 18 | 1 | Rendimiento | -0.172413793 | 0.340000000 |
| BD_Portafolio.csv | 13) ELEKTRA.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.19296875 | 0.209790442 |
| BD_Portafolio.csv | 14) FEMSAUBD.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.131674853 | 0.134005764 |
| BD_Portafolio.csv | 15) GCARSOA1.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.113065327 | 0.245706737 |
| BD_Portafolio.csv | 16) GEOB.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.1672742 | 0.175552666 |
| BD_Portafolio.csv | 17) GFINBURO.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.081415929 | 0.240000000 |
| BD_Portafolio.csv | 18) GFNORTEO.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.215291751 | 0.31025641 |
| BD_Portafolio.csv | 19) GMEXICO-B.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.168539326 | 0.190883191 |
| BD_Portafolio.csv | 20) GMODELOC.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.103080569 | 0.194486983 |
| BD_Portafolio.csv | 21) HERDEZ.MX | Decimal | 21 | 18 | 5 | Rendimiento | -0.438508065 | 0.693895871 |
| BD_Portafolio.csv | 22) HOMEX.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.23014763 | 0.284873022 |
| BD_Portafolio.csv | 23) ICA.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.241887906 | 0.344017094 |
| BD_Portafolio.csv | 24) KIMBERA.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.107063197 | 0.151079137 |
| BD_Portafolio.csv | 25) KOFL.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.166666667 | 0.108585346 |
| BD_Portafolio.csv | 26) LIVERPOLC-1.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.243589744 | 0.322033898 |
| BD_Portafolio.csv | 27) MEXCHEM.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.180319149 | 0.226628895 |
| BD_Portafolio.csv | 28) QCPO.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.4092827 | 0.675000000 |
| BD_Portafolio.csv | 29) TELMEXL.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.075650118 | 0.080110497 |
| BD_Portafolio.csv | 30) TLEVISACPO.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.078138163 | 0.162021858 |
| BD_Portafolio.csv | 31) URBI.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.223897912 | 0.187766714 |
| BD_Portafolio.csv | 32) WALMEXV.MX | Decimal | 21 | 18 | 0 | Rendimiento | -0.142857143 | 0.173333333 |
| BD_Portafolio.csv | Diagnostico | Binary | 2 | 0 | 0 | Categoría | 0 | 1 |

*Tabla F.1 – Diccionario de Datos para la base de datos de BD_Portafolios.*

El diccionario de datos para la Base de Datos de WDBC se encuentra en la Tabla F.2:

| Table Name | FieldName | Type | Length | Decimal | Null | Description | Minimum | Maximum |
|---|---|---|---|---|---|---|---|---|
| wdbc.csv | Id_number | Integer | 9 | . | 0 | PK - paciente | 8670 | 911320502 |
| wdbc.csv | Salida | Char | 1 | . | 0 | Categoría | M | B |
| wdbc.csv | Radio_promedio | Decimal | 6 | 3 | 0 | Media | 6.981 | 28.110 |
| wdbc.csv | Textura_promedio | Decimal | 5 | 2 | 0 | Media | 9.71 | 39.28 |
| wdbc.csv | Perimetro_promedio | Decimal | 6 | 2 | 0 | Media | 43.79 | 188.50 |
| wdbc.csv | Area_promedio | Decimal | 6 | 1 | 0 | Media | 143.5 | 2501.0 |
| wdbc.csv | Suavidad_promedio | Decimal | 7 | 5 | 0 | Media | 0.05263 | 0.16340 |
| wdbc.csv | Compacidad_promedio | Decimal | 7 | 5 | 0 | Media | 0.01938 | 0.34540 |
| wdbc.csv | Concavidad_promedio | Decimal | 9 | 7 | 13 | Media | 0 | 0.4268000 |
| wdbc.csv | Puntos_Concavos_promedio | Decimal | 8 | 6 | 13 | Media | 0 | 0.201200 |
| wdbc.csv | Simetría_promedio | Decimal | 6 | 4 | 0 | Media | 0.1060 | 0.3040 |
| wdbc.csv | Dimensión_Fractal_ promedio | Decimal | 7 | 5 | 0 | Media | 0.04996 | 0.09744 |
| wdbc.csv | Error_Estandar_Radio | Decimal | 6 | 4 | 0 | Error Estándar | 0.1115 | 2.8730 |
| wdbc.csv | Error_Estandar_Textura | Decimal | 6 | 4 | 0 | Error Estándar | 0.3602 | 4.8850 |
| wdbc.csv | Error_Estandar_Perimetro | Decimal | 7 | 4 | 0 | Error Estándar | 0.7570 | 21.9800 |
| wdbc.csv | Error_Estandar_Area | Decimal | 7 | 3 | 0 | Error Estándar | 6.802 | 542.200 |
| wdbc.csv | Error_Estandar_Suavidad | Decimal | 8 | 6 | 0 | Error Estándar | 0.001713 | 0.031130 |
| wdbc.csv | Error_Estandar_ Compacidad | Decimal | 8 | 6 | 0 | Error Estándar | 0.002252 | 0.135400 |
| wdbc.csv | Error_Estandar_Concavidad | Decimal | 9 | 7 | 13 | Error Estándar | 0 | 0.3960000 |
| wdbc.csv | Error_Estandar_Puntos_ Concavos | Decimal | 8 | 6 | 13 | Error Estándar | 0 | 0.052790 |
| wdbc.csv | Error_Estandar_Simetría | Decimal | 8 | 6 | 0 | Error Estándar | 0.007882 | 0.078950 |
| wdbc.csv | Error_Estand_Dimensión_ Fractal | Decimal | 9 | 7 | 0 | Error Estándar | 0.0008948 | 0.0298400 |
| wdbc.csv | Peor/más_largo_Radio | Decimal | 5 | 2 | 0 | Peor/Más Grande | 7.93 | 36.04 |
| wdbc.csv | Peor/más_larga_Textura | Decimal | 5 | 2 | 0 | Peor/Más Grande | 12.02 | 49.54 |
| wdbc.csv | Peor/más_largo_Perimetro | Decimal | 6 | 2 | 0 | Peor/Más Grande | 50.41 | 251.20 |
| wdbc.csv | Peor/más_largo_Area | Decimal | 6 | 1 | 0 | Peor/Más Grande | 185.2 | 4254.0 |
| wdbc.csv | Peor/más_largo_Suavidad | Decimal | 7 | 5 | 0 | Peor/Más Grande | 0.07117 | 0.22260 |
| wdbc.csv | Peor/más_largoCompacidad | Decimal | 7 | 5 | 0 | Peor/Más Grande | 0.02729 | 1.05800 |
| wdbc.csv | Peor/más_largo_Concavidad | Decimal | 8 | 6 | 13 | Peor/Más Grande | 0 | 1.252000 |
| wdbc.csv | Peor/más_largo_Punto_ Concavo | Decimal | 8 | 6 | 13 | Peor/Más Grande | 0 | 0.291000 |
| wdbc.csv | Peor/más_largo_Simetría | Decimal | 6 | 4 | 0 | Peor/Más Grande | 0.1565 | 0.6638 |
| wdbc.csv | Peor/más_largo_Dimensión_ Fractal | Decimal | 7 | 5 | 0 | Peor/Más Grande | 0.05504 | 0.20750 |

*Tabla F.2 – Diccionario de Datos para la base de datos de WDBC.*

Por último, el diccionario de datos para la Base de Datos de WPBC se detalla a continuación:

| Table Name | FieldName | Type | Length | Decimal | Null | Description | Minimum | Maximum |
|---|---|---|---|---|---|---|---|---|
| wpbc.csv | ID number | Integer | 7 | . | 0 | PK - paciente | 8423 | 9411300 |
| wpbc.csv | Outcome (R = recurrent, N = no recurrent) | Char | 1 | . | 0 | Categoría | R | N |
| wpbc.csv | Tiempo | Integer | 3 | . | 0 | En días | 1 | 125 |
| wpbc.csv | Radio_promedio | Decimal | 5 | 2 | 0 | Media | 10.95 | 27.22 |
| wpbc.csv | Textura_promedio | Decimal | 5 | 2 | 0 | Media | 10.38 | 39.28 |
| wpbc.csv | Perimetro_promedio | Decimal | 5 | 2 | 0 | Media | 71.9 | 182.10 |
| wpbc.csv | Area_promedio | Decimal | 5 | 2 | 0 | Media | 361.6 | 2250.00 |
| wpbc.csv | Suavidad_promedio | Decimal | 7 | 5 | 0 | Media | 0.07497 | 0.14470 |
| wpbc.csv | Compacidad_promedio | Decimal | 7 | 5 | 0 | Media | 0.04605 | 0.31140 |
| wpbc.csv | Concavidad_promedio | Decimal | 7 | 5 | 0 | Media | 0.02398 | 0.42680 |
| wpbc.csv | Puntos_Concavos_promedio | Decimal | 7 | 5 | 0 | Media | 0.02031 | 0.20120 |
| wpbc.csv | Simetría_promedio | Decimal | 6 | 4 | 0 | Media | 0.1308 | 0.3040 |
| wpbc.csv | Dimensión_Fractal_ promedio | Decimal | 7 | 5 | 0 | Media | 0.05025 | 0.09744 |
| wpbc.csv | Error_Estandar_Radio | Decimal | 6 | 4 | 0 | Error Estándar | 0.1938 | 1.8190 |
| wpbc.csv | Error_Estandar_Textura | Decimal | 6 | 4 | 0 | Error Estándar | 0.3621 | 3.5030 |
| wpbc.csv | Error_Estandar_Perimetro | Decimal | 6 | 3 | 0 | Error Estándar | 1.153 | 13.280 |
| wpbc.csv | Error_Estandar_Area | Decimal | 6 | 2 | 0 | Error Estándar | 13.99 | 316.00 |
| wpbc.csv | Error_Estandar_Suavidad | Decimal | 8 | 6 | 0 | Error Estándar | 0.002667 | 0.031130 |
| wpbc.csv | Error_Estandar_ Compacidad | Decimal | 8 | 6 | 0 | Error Estándar | 0.007347 | 0.135400 |
| wpbc.csv | Error_Estandar_Concavidad | Decimal | 7 | 5 | 0 | Error Estándar | 0.01094 | 0.14380 |
| wpbc.csv | Error_Estandar_Puntos_ Concavos | Decimal | 8 | 6 | 0 | Error Estándar | 0.005174 | 0.039270 |
| wpbc.csv | Error_Estandar_Simetría | Decimal | 8 | 6 | 0 | Error Estándar | 0.007882 | 0.060410 |
| wpbc.csv | Error_Estand_Dimensión_ Fractal | Decimal | 8 | 6 | 0 | Error Estándar | 0.001087 | 0.012560 |
| wpbc.csv | Peor/más_largo_Radio | Decimal | 5 | 2 | 0 | Peor/Más Grande | 12.84 | 35.13 |
| wpbc.csv | Peor/más_larga_Textura | Decimal | 5 | 2 | 0 | Peor/Más Grande | 16.67 | 49.54 |
| wpbc.csv | Peor/más_largo_Perimetro | Decimal | 6 | 2 | 0 | Peor/Más Grande | 85.10 | 232.20 |
| wpbc.csv | Peor/más_largo_Area | Decimal | 6 | 1 | 0 | Peor/Más Grande | 508.1 | 3903.0 |
| wpbc.csv | Peor/más_largo_Suavidad | Decimal | 7 | 5 | 0 | Peor/Más Grande | 0.08191 | 0.22260 |
| wpbc.csv | Peor/más_largoCompacidad | Decimal | 7 | 5 | 0 | Peor/Más Grande | 0.05131 | 1.05800 |
| wpbc.csv | Peor/más_largo_Concavidad | Decimal | 7 | 5 | 0 | Peor/Más Grande | 0.02398 | 1.17000 |
| wpbc.csv | Peor/más_largo_Punto_ Concavo | Decimal | 7 | 5 | 0 | Peor/Más Grande | 0.02899 | 0.29030 |
| wpbc.csv | Peor/más_largo_Simetría | Decimal | 6 | 4 | 0 | Peor/Más Grande | 0.1565 | 0.6638 |
| wpbc.csv | Peor/más_largo_Dimensión_ Fractal | Decimal | 7 | 5 | 0 | Peor/Más Grande | 0.05504 | 0.20750 |
| wpbc.csv | Tamaño del Tumor | Decimal | 4 | 1 | 0 | Diámetro del Tumor | 0.4 | 10.0 |
| wpbc.csv | Estado del Nodo Linfático | Integer | 2 | . | 91 | Observado | 0 | 27 |

*Tabla F.3 – Diccionario de Datos para la base de datos de WPBC.*