UNIVERSIDAD NACIONAL DE LA MATANZA

ESCUELA DE POSGRADO

TESIS DE MAESTRÍA

Título de Tesis: Estudio Comparativo de Técnicas de Minería de Datos para la predicción de la deserción universitaria.

Autor: Lic. Julio César Bossero

Director: Mg. Ing. Osvaldo Spositto.

Buenos Aires, Fecha

Tabla de contenido

Capítulo 1: 8
1. Introducción 8
1.1. Motivación del trabajo 11
1.2. Objetivos 13
1.3. Objetivo General 13
1.4. Objetivos específicos 13
1.5. Hipótesis 13
1.6. Hipótesis principal: 13
1.7. Hipótesis secundaria: 13
1.8. Estructura de la Tesis 13
1.9. Publicaciones vinculadas a esta tesis 14
1.10. Participación en Proyectos de investigación ProInce (UNLaM) 15
Capítulo 2. 17
2. Estado de Situación 17
2.1. Sistemas OLTP vs OLAP 18
2.2. Data Warehousing 20
2.3. Arquitectura de un DW 24
2.4. MOLAP (Multidimensional OLAP) 25
2.5. ROLAP (Relational OLAP) 26
2.6. Componentes del Modelo Multidimensional 27
2.7. Definición de la granularidad 29
2.8. Tablas de Hechos 30
2.9. Tablas de Dimensiones 30
2.10. Esquema en estrella: 31
2.11. Esquema en Copo de Nieve. 31
2.12. Data Warehouse y Data Mining 32
2.13. Descubrimiento de Conocimiento en Bases de Datos 32
2.14. Metodología para realizar MD 33
2.15. Fases en el proceso KDD 34
2.15.1. Fase de integración y recopilación: 34
2.15.2. Fase de selección, limpieza y transformación: 35
2.15.3. Fase de Minería de Datos: 35
2.15.4. Fase de evaluación e Interpretación: 35
2.15.5. Fase de difusión y uso: 36
2.16. Minería de Datos 37
2.17. Tareas de la Minería de Datos 37
2.17.1. Tareas descriptivas: 37
2.17.2. Tareas de Predicción: 38
2.18. Técnicas de Minería de Datos 39
2.19. Tipos de Aprendizajes: 39
2.20. Los algoritmos propuestos: 41
2.21. Red Neuronal Artificial. 42
2.22. Las Máquinas de Vectores de Soporte. 42
2.23. Herramienta de Minería de Datos 44
2.24. Trabajos similares: 46

Capítulo 3. 50
3. Modelos de Clasificación de Minería de Datos 50
3.1. Aprendizaje Automatizado (AA). 50
3.2. Modelos de Clasificación basados en Redes Neuronales Artificiales. 52
3.2.1. El Cerebro Humano 52
3.2.2. Sustancia blanca y Sustancia gris 54
3.2.3. La Neurona Artificial 54
3.2.4. Funciones de Activación. 55
3.2.5. Tipos de neuronas 57
3.2.6. ¿Qué es una Red Neuronal Artificial? 58
3.2.7. Breve reseña histórica. 58
3.2.8. ¿Qué son capaces de hacer la RNA? 59
3.2.9. Topología de las Redes Neuronales. 60
3.2.9.1. Redes Monocapa. 61
3.2.9.2. Redes Multicapa. 61
3.2.10. Mecanismos de aprendizajes. 62
3.2.11. El Modelo Perceptrón 64
3.2.11.1. La Regla de Aprendizaje del Perceptrón 65
3.2.12. El Modelo Perceptrón Multicapa 66
3.2.12.1. El algoritmo de retropropagación 68
3.2.13. Tamaño de las Redes Neuronales 69
3.3. Modelos de Clasificación basados en Máquinas de Vectores Soporte. 71
3.3.1. MVS para clasificación binaria de ejemplos separables linealmente 73
3.3.2. MVS para clasificación binaria de ejemplos No separables linealmente 76
3.3.3. Margen Hiperplano Máximo 79
3.3.4. Razones para un Margen Máximo 80
3.3.5. Minimización del Riesgo Estructural (SRM) 80
3.3.6. Algunas características de las Máquinas de Vectores Soporte (SVM) 81
3.3.7. MVS y la clasificación Binaria: Propiedades destacables. 82
3.3.8. Limitaciones de las MVS. 83
Capítulo 4: 84
4. Caso de Estudio y Experimentación 84
4.1. Integración y Recopilación de los Datos 86
4.2. Preparación de los datos. 86
4.3. Definición de deserción 88
4.4. Minería de Datos 90
4.5. Introducción a WEKA 92
4.5.1. Pestaña Pre-procesado (Preprocess) de los datos 93
4.5.1.1. Archivos ARFF 94
4.5.1.2. Construcción de los modelos 95
4.5.2. Pestaña Selección de Atributos (Select Attributes) 97
4.5.2.1. Filtrado y limpieza de datos. 99
4.5.3. Pestaña Clasificar (Classify) 103
4.6. Modelo de clasificación basado en RNA 105
4.6.1. Lista de resultados (Result list) 110
4.7. Modelo de clasificación basado en MVS 111
4.8. Comparación de los Modelos 115

4.9. Evaluación 118
4.9.1. Métricas de evaluación 118
4.9.1.1. Matriz de confusión 118
4.9.1.2. ¿Qué es una curva ROC? 120
4.9.1.3. Área bajo la curva ROC 121
4.10. Resultados en datos 122
Capítulo 5 123
5. Conclusiones y Trabajos Futuros 123
5.1. Conclusiones 123
5.2. Trabajos Futuros 124
6. Bibliografía 125
7. Anexos 131
- Anexo I. 131
- Anexo II 131

FIGURA 1: MÓDULOS QUE COMPONEN EL SIU-GUARANÍ. 22
FIGURA 2: ARQUITECTURA DEL DATA WAREHOUSE DEL DIIT [28]. 24
FIGURA 3: ESTRUCTURA DE TABLAS DEL ÁREA INTERMEDIA DE DATOS [28]. 27
FIGURA 4: REPRESENTACIÓN GRÁFICA DE UN CUBO 29
FIGURA 5: EJEMPLO DE UN ESQUEMA ESTRELLA 31
FIGURA 6: EJEMPLO DE UN ESQUEMA COPO DE NIEVE 32
FIGURA 7: PROCESO DE KDD. [5] 33
FIGURA 8: TÉCNICAS DE LA MINERÍA DE DATOS 41
FIGURA 9: ESTRUCTURA DE LA NEURONA ARTIFICIAL BÁSICA Y MODELADO DEL UMBRAL. 42
FIGURA 10: GRÁFICO DEL HIPERPLANO EQUIDISTANTE PARA UN CASO BIDIMENSIONAL. 43
FIGURA 11: TRANSFORMACIÓN MEDIANTE LA FUNCIÓN KERNEL PARA SEPARABILIDAD NO LINEAL. 43
FIGURA 12: VENTANA PRINCIPAL DE WEKA. 45
FIGURA 13. NEURONA BIOLÓGICA. [40] 53
FIGURA 14. MODELO DE NEURONA DE MCCULLOCH Y PITTS. 54
FIGURA 15. FUNCIÓN DE ACTIVACIÓN. 55
FIGURA 16. FUNCIÓN ESCALÓN 56
FIGURA 17. FUNCIÓN SIGMOIDAL 56
FIGURA 18. FUNCIÓN HIPERBÓLICA 57
FIGURA 19. FUNCIÓN DE TRANSFERENCIA LINEAL 57
FIGURA 20. CONEXIONES HACIA ADELANTE (FEED FORWARD), LATERALES Y HACIA ATRÁS (BACK PROPAGATION) 61
FIGURA 21. ARQUITECTURA DE RNA. 62
FIGURA 22. APRENDIZAJE SUPERVISADO. [45] 63
FIGURA 23 (A Y B). FUNCIONES DEL PERCEPTRÓN SIMPLE. 64
FIGURA 24. INFLUENCIA DE LA SALIDA DE LA NEURONA N_J EN LA ENTRADA DE LA NEURONA N_I. 66
FIGURA 14. EJEMPLO DE UNA TOPOLOGÍA DE UN PM CON DOS CAPAS OCULTAS. 67
FIGURA 27. ARQUITECTURAS Y REGIONES DE DECISIÓN. 70
FIGURA 28. HIPERPLANO DE SEPARACIÓN DE DATOS LINEALMENTE SEPARABLES. 71
FIGURA 29. HIPERPLANO PARA UN CASO LINEALMENTE SEPARABLE. 73
FIGURA 30. TRANSFORMACIÓN ESPACIAL DEL ESPACIO DE ENTRADA. 76
FIGURA 31. HIPERPLANOS DE DECISIÓN. 78
FIGURA 32. MARGEN HIPERPLANO.[52] 79
FIGURA 33. TIPOS DE AJUSTES 81
FIGURA 34: ETAPAS QUE COMPONEN UN PROCESO KDD. [1] 85

FIGURA 35. TABLA DE HECHO Y DE DIMENSIONES QUE CONFORMAN EL DW DEPARTAMENTAL. 87
FIGURA 36. VENTANA PRINCIPAL DEL PROGRAMA WEKA. 92
FIGURA 37. PANTALLA PRINCIPAL DEL EXPLORADOR DE WEKA 93
FIGURA 38. PANTALLA PRINCIPAL DEL EXPLORADOR DE WEKA 97
FIGURA 39. PANTALLA PRINCIPAL DEL FILTRADO DE ATRIBUTOS DE WEKA 98
FIGURA 40. PESTAÑA CLASSIFY DE WEKA 103
FIGURA 41. MENÚ DE SELECCIÓN DE ALGORITMOS 103
FIGURA 42. MENÚ DE SELECCIÓN DE ALGORITMOS Y CONFIGURACIÓN DE UNA RNA. 105
FIGURA 43: RESULTADOS PARA UNA RNA CON 5 NEURONAS EN LA CAPA OCULTA. 108
FIGURA 44: SALIDA GRÁFICA DE UNA RNA CON 5 NEURONAS EN LA CAPA OCULTA. 109
FIGURA 45: SALIDA GRÁFICA DE UNA RNA CON 5 NEURONAS EN LA CAPA OCULTA. 110
FIGURA 46: CONFIGURAR EL ALGORITMO SMO. 112
FIGURA 47: CONFIGURAR EL ALGORITMO SMO. 112
FIGURA 48: SALIDA DE WEKA USANDO LA FUNCIÓN KERNEL POLINOMIAL 114
FIGURA 49: SALIDA DE WEKA USANDO LA FUNCIÓN RBF KERNEL 114
FIGURA 50: SALIDA DE WEKA USANDO LA FUNCIÓN RBF KERNEL 115
FIGURA 51: SALIDA DE WEKA USANDO LA FUNCIÓN RBF KERNEL 116
FIGURA 52: SALIDA DE WEKA USANDO LA FUNCIÓN RBF KERNEL 116
FIGURA 53: SALIDA DE WEKA DEL ALGORITMO RNA. 117
FIGURA 54: SALIDA DE WEKA DEL ALGORITMO MVS. 117
FIGURA 55: RESULTADO DE UNA PRUEBA Y SU ESTADO RESPECTO A LA CLASE. 119
FIGURA 56: ESQUEMA EXPLICATIVO DE LAS CURVAS ROC. 121
FIGURA 57: ÁREA BAJO ROC DE RNA. 122
FIGURA 58: ÁREA BAJO DE MVS. 122

Estudia el pasado si quieres pronosticar el futuro.

Confucio

Capítulo 1:

1. Introducción

La Minería de Datos Educacional (MDE o EDM[1]), es una rama de la Minería de Datos (MD o DM[2]), que se ha dedicado a aplicar diversas técnicas para analizar datos provenientes de ambientes relacionados a la educación formal, y a extraer la mayor cantidad de conocimiento, con el objeto de entender mejor a los estudiantes, profesores y actores involucrados, y así mejorar los procesos educativos [1]. Algunos hechos que dan cuenta del aumento del interés en esta área son las conferencias internacionales de minería de datos educacionales que se vienen realizando desde el año 2008. En el 2009 se publicó el JEDM[3] y en julio del 2011 se fundó la *International Educational Data Mining Society* cuyo objetivo es *"…apoyar la colaboración y el desarrollo científico en esta nueva disciplina, a través de la organización de la serie de conferencias EDM, el Journal of Educational Data Mining, y listas de correo, así como también el desarrollo de recursos comunitarios para apoyar el intercambio de datos y técnicas…*" [2]. En el sitio de *Educational Data Mining*[4] se puede acceder a los temas expuestos en la 8va y 9va conferencia internacional de MDE, realizada en Madrid y Carolina del Norte. EU respectivamente.

En la actualidad, la tendencia en las universidades es trabajar con tecnologías MDE, que faciliten y mejoren el desarrollo de las actividades académicas, creándose de esta forma un aprendizaje electrónico añadido al tradicional. El uso de estos nuevos medios tiene grandes ventajas, una de ellas es que permite generar una gran cantidad de datos producidos por el estudiante durante el proceso de aprendizaje. Esta información procesada con las herramientas adecuadas permite predecir, prevenir y/o actuar para mejorar el rendimiento académico de los estudiantes. [3]

El descubrir información oculta en grandes cantidades de registros de bases de datos transaccionales es una de las tareas principales de la Minería de Datos, que según Piatetski-

[1] *por sus siglas en inglés Educational Data Mining.*
[2] *por sus siglas en inglés Data Mining.*
[3] *por sus siglas en inglés Journal of Educational Data Mining.*
[4] *http://www.educationaldatamining.org/EDM2015/*

Shapiro es *"…un conjunto de técnicas y herramientas aplicadas al proceso no trivial de extraer y presentar conocimiento implícito, previamente desconocido, potencialmente útil y humanamente comprensible, a partir de grandes conjuntos de datos, con objeto de predecir, de forma automatizada, tendencias o comportamientos y descubrir modelos previamente desconocidos…*" [4].

La Minería de Datos, entre otras técnicas, utiliza Inteligencia Artificial para encontrar patrones y relaciones entre los datos, permitiendo la creación de modelos y representaciones abstractas de la realidad. Esta es una etapa dentro del proceso mayor llamado "*Descubrimiento de Conocimiento en Base de Datos*" (DCBD o KDD[5]), aunque en la mayoría de los entornos, ambos términos se usan de manera indistinta. El proceso consiste en extraer patrones en forma de reglas o funciones, a partir de los datos, para que el usuario los analice. Las técnicas de Minería de Datos se emplean principalmente para clasificar y predecir datos en diversas áreas, como ser economía, sociedad, gobierno, medicina y otras, de cara a descubrir información novedosa, útil y válida de los datos que tienen almacenados en sus sistemas transaccionales [5].

El proceso completo de KDD incluye entre otras cosas, la preparación de los datos y la interpretación de los resultados obtenidos, los cuales dan un significado a los patrones identificados por las técnicas y algoritmos de MD. Así el valor real de los datos reside en la información que se puede extraer de ellos, información que ayude a tomar decisiones o mejorar nuestra comprensión de los fenómenos que nos rodean [6]. Para Hernández Orallo, la MD se encuentra dentro de las etapas de un proceso mucho más amplio conocido como DCBD y es básicamente un proceso automático o semiautomático, en el que se combinan descubrimiento y análisis [5].

De acuerdo a la bibliografía [5] y [7] las técnicas de MD más frecuentes pueden ser catalogadas en:

- *Descriptivas*: El objetivo de estos procedimientos es la búsqueda de la caracterización o discriminación de un conjunto de datos. Las técnicas más conocidas son: Agrupamiento

[5] *del inglés Knowledge Discovery in Databases.*

o Clustering, Reglas de Asociación, Análisis de Patrones Secuenciales, Análisis de Componentes Principales, Detección de Desviación.

- *Predictivas*: El propósito de estas técnicas es entrenar a un modelo o método por medio de diferentes datos para poder predecir una variable partiendo de estos mismos datos. Los algoritmos principales son: Regresión y Clasificación (Árboles de Decisión, Clasificación Bayesiana, Redes Neuronales, Algoritmos Genéticos, Máquina de Vectores Soporte, Conjuntos y Lógica Difusa, etc.). Cuando se realiza un proceso de MD, se necesita tener en cuenta el conocimiento previo; este puede derivar del proceso mismo (elección de variables, técnicas, algoritmos, interpretación de resultados) o del dominio de aplicación.

En este trabajo se realiza una introducción a dos técnicas de MD, por un lado Red Neuronal Artificial (RNA o ANN[6]) del tipo Perceptron Multicapa (PM), y por otro una Máquina de Vectores Soporte (MVS o SVM[7]), para explorar su aplicabilidad en el terreno de la MDE como instrumento de modelización y predicción no paramétrica. Con tal objeto, se pretende desarrollar ambos modelos, y luego ser posteriormente aplicados en la predicción de perfiles de alumnos desertores, utilizando datos reales provenientes de un Almacén de Datos (AD o DW[8]). Los resultados obtenidos serán comparados para determinar cuál de ellos es más eficaz y eficiente en la predicción y poder ser utilizado, de ser necesario, por las autoridades para tomar acciones anticipadas que ayuden a disminuir el índice de deserción o desgranamiento universitario.

Como definición teórica y práctica de deserción, para este trabajo se usará la de Vicent Tinto en [8], quien la define como una situación a la que se enfrenta un estudiante cuando aspira y no logra concluir su proyecto educativo. Asimismo, se considera como desertor a aquel individuo que, siendo estudiante del DIIT[9], no presenta actividad académica durante tres cuatrimestres académicos consecutivos.

[6] *Por sus siglas en inglés Artificial Neural Networks.*
[7] *Por sus siglas en inglés: Support Vector Machines*
[8] *Por sus siglas en inglés: Data Warehouse.*
[9] Departamento de Ingeniería e Investigaciones Tecnológicas

1.1. Motivación del trabajo

En los últimos años ha surgido en muchos países una preocupación ante el problema de la deserción de estudiantes universitarios, lo que conlleva a un creciente interés por determinar los múltiples factores que pueden influir en él. La mayoría de los trabajos que intentan resolver este problema [9-12] están enfocados en determinar cuáles son los factores que más afectan al rendimiento de los estudiantes (abandono y fracaso) en los diferentes niveles educativos (educación básica, media y superior), llegando en varios casos a la conclusión de que la deserción es un fenómeno extremadamente complejo, de naturaleza multidimensional y que no sólo afecta a los alumnos sino también a las instituciones, a la sociedad y con alcance a nivel nacional e internacional.

Uno de los mayores desafíos en MDE es predecir el desempeño de un alumno, dados sus datos históricos recopilados por la interacción con algún sistema computacional, de ejercitación y aprendizaje. Una de las técnicas más importantes en minería de datos educativa es la clasificación. La clasificación es una técnica predictiva, que realiza la predicción a partir de resultados conocidos que se encuentran en diferentes datos [5]. Como ya se mencionó, los modelos de predicción tienen el objetivo específico de predecir los valores desconocidos de las variables de interés, dados los valores conocidos de otras. Los modelos de predicción pueden ser considerados como el aprendizaje a partir de un mapeo de un conjunto de mediciones de vectores de entrada a una salida escalar. La clasificación mapea un conjunto de datos en grupos predefinidos de clases. Se conoce como aprendizaje supervisado, porque las clases se determinan antes de examinar los datos.

Varios trabajos y publicaciones [12-17] han abordado esta problemática desde la perspectiva de encontrar, a priori, perfiles de alumnos desertores, buscando en información histórica referida mayormente a su desempeño académico, para su tratamiento y análisis, aplicando una variada cantidad de técnicas. Si bien esto constituye una fuente de información valiosa de los alumnos, trae aparejado una dificultad, y es que hay muchos aspectos en los datos, tantos que rebasan las capacidades de procesamiento y asimilación de cualquier ser humano. Resulta de interés, entonces, la disposición de herramientas computacionales capaces de analizar de forma semiautomática y eficiente esa gran cantidad de información

acumulada en bases de datos. La cantidad de técnicas distintas que se han ensayado ponen en evidencia que no hay todavía ninguna que sea absoluta en sus conclusiones. Estudiar varias de estas técnicas sobre una misma población, con una problemática bien definida y en un contexto sobre el cual se pueda actuar, puede agregar luz sobre los fundamentos de las mismas y las condiciones que hacen a una, preferible sobre otra.

A partir de lo anteriormente planteado, en este trabajo se propone realizar un estudio comparativo sobre dos algoritmos de MD del tipo aprendizaje supervisado. Para ello fue creado un modelo también conocido con el nombre "Modelado o Vista Minable", para poder ser utilizado para testear dichos algoritmos. Un modelo es el conjunto de reglas, fórmulas o ecuaciones que se extraen de los datos de origen y le permite generar predicciones respecto a estos. Esa es la base del análisis predictivo [18]. En otras palabras, un Modelo no es otra cosa que "una manera de aplicar un tratamiento a una cantidad específica de datos para obtener información de ellos". Luego, se analizará cuál de ellos entrega una respuesta más beneficiosa. Está planeado para esta investigación, trabajar sobre la problemática de la deserción universitaria, mediante el proceso de revelar conocimiento útil y no evidente, explorando la información disponible en bases de datos de la UNLaM[10], para que este conocimiento se constituya en un apoyo esencial en la toma de decisiones. Así, como beneficio secundario de haber elegido este dominio de experimentación, el presente trabajo abordará la deserción universitaria que es una preocupante problemática en la mayoría de las Universidades Argentinas.

Dado que la carrera elegida refleja aptitudes, mentalidad, y en parte contexto social del alumno, se ha decidido comenzar con el alumnado de una única unidad académica. Así la investigación quedará acotada al estudio de los alumnos del DIIT de la UNLaM, pues sus autoridades se encuentran interesadas en tomar las directivas correctas para disminuir este fenómeno. Por otra parte, este tesista pertenece a esa unidad académica, lo que les da una oportunidad de conocimiento y mejor acceso a la información.

[10] Universidad Nacional de La Matanza

1.2. Objetivos

A continuación se describen los objetivos de la presente tesis.

1.3. Objetivo General

- Identificar y analizar predictivamente los perfiles de estudiantes con riesgo de deserción utilizando Redes Neuronales Artificiales y Máquinas de Vectores Soporte.

1.4. Objetivos específicos

- Llevar a cabo un estudio descriptivo/comparativo de las características particulares de los algoritmos de Minería de Datos propuestos.
- Desarrollar e implementar una vista minable para comparar los algoritmos propuestos.
- Realizar las transformaciones y limpieza de los datos originales con el fin de construir una vista minable.

1.5. Hipótesis

1.6. Hipótesis principal:

- Dos Modelos de Minería de Datos, uno basado en una Red Neuronal Artificial (RNA) del tipo Perceptron Multicapa, y otro del tipo Máquinas de Vectores Soporte (MVS), son capaces de identificar características y patrones de comportamiento, que permiten predecir la probabilidad de que algunos alumnos abandonen sus estudios universitarios.

1.7. Hipótesis secundaria:

- El modelo basado en MVS predice más eficaz y eficientemente la probabilidad de deserción que el modelo RNA del tipo Perceptrón Multicapa.

1.8. Estructura de la Tesis

El resto de este trabajo se encuentra organizado de la siguiente manera:

- **Capítulo II:**

  Se desarrollarán los siguientes conceptos: Qué es un Almacén de Datos; Qué es el proceso de descubrimiento de conocimiento en base de datos; descripción de las etapas. Que es la Minería de Datos; utilidades y los tipos de algoritmos que existen. Una breve introducción a los algoritmos propuestos. Que es Weka y por último se presenta una breve descripción de la situación del tema en estudio a nivel nacional y mundial.

- **Capítulo III**

  Se explican los conceptos de Modelos de Clasificación, Aprendizaje Automatizado. Se expondrá con más detalle cómo funcionan los distintos algoritmos. Modelos de Clasificación basados en RNA y los Modelos de Clasificación basados en Máquinas de Vectores Soporte. Que es el Margen Hiperplano Máximo. Razones para un Margen Máximo. Minimización del Riesgo Estructural (SRM). MVS y los caso separable linealmente y los caso no linealmente separables.

- **Capítulo IV:**

  Explicación de las herramientas tecnológicas utilizadas para el desarrollo del caso de estudio y de la metodología que se va a utilizar en la aplicación de los algoritmos, su evaluación y comparación. Exposición de las características exigidas a los datos recogidos para ser aceptados para realizar los experimentos. Después se mostrarán con detalle todos los experimentos realizados con el fin de lograr la clasificación de los alumnos, bien utilizando diversos tipos de redes neuronales artificiales y de distintos kernel de máquinas de vectores soporte.

- **Capítulo V: Conclusiones de la tesis**

  Se darán algunas líneas a trabajar en el futuro.

1.9. Publicaciones vinculadas a esta tesis

- "***Predicción del riesgo de abandono universitario utilizando métodos supervisados***" En colaboración con Edwards, Diego y Pérez, Silvia (UNLaM). Trabajo presentado en el Workshop de la V Jornadas Nacionales y I Latinoamericanas de Ingreso y Permanencia en Carreras Científico – Tecnológicas. Facultad Regional Bahía Blanca. Universidad Tecnológica Nacional. Bahía Blanca. Mayo de 2016. IPECyT 2016.

http://www.frbb.utn.edu.ar/ipecyt2016/

- "***Comparación de Algoritmos de Aprendizaje Supervisado para la obtención de perfiles de alumnos desertores***". En colaboración con el Ing. Osvaldo Spositto (UNLaM). Trabajo presentado en el "Workshop del IV Congreso Nacional de Ingeniería en Informática/Sistemas de Información. Publicación on line - ISSN 2347-0372. CONAIISI 2016. Salta. Argentina.
  http://www.ucasal.edu.ar/conaiisi2016/book/memorias.html
- "***Modelos de minería de datos para el diagnóstico de enfermedad de Parkinson mediante el análisis de voz***". En colaboración con el Ing. Osvaldo Spositto, Ing. Gabriel Blanco, Mg. Mónica Giuliano y el Ing. Luis Fernández (UNLaM). Trabajo presentado en el "Workshop del V Congreso Nacional de Ingeniería en Informática/Sistemas de Información. Publicación on line - ISSN. CONAIISI 2017. Santa Fe. Argentina. http://conaiisi2017.frsf.utn.edu.ar/index.php/memorias/

1.10. Participación en Proyectos de investigación ProInce (UNLaM)

Seguidamente, se enumeran por orden cronológico inverso tanto los proyectos vigentes como los concluidos.

- Sustitución de programación secuencial con cursores por nuevas formas sintácticas que impliquen paralelismo. Anuario de Investigaciones. Resúmenes Extendidos (RE) 2008. ISBN: 978-987-1635-01-6. Pp.125-128
- Aplicación de Técnicas de Minería de Datos para la evaluación del Rendimiento Académico y la Deserción Estudiantil. Anuario de Investigaciones. RE 2009. ISBN: 978-987-1635-23-8. Pp.15-24
- Utilización de Técnicas de Data Warehouse para la toma de decisiones en el área académica. Anuario de Investigaciones. RE 2010. ISBN: 978-987-1635-55-9. Pp.173-178 y RE 2011. ISBN: 978-987-1635-77-1. Pp.119-122
- Implementación de un Data Warehouse para la toma de decisiones en el área académica. Anuario de Investigaciones. R. Entendidos 2012. ISBN: 978-987-3806-01-8. Pp.161-166 y RE 2012. ISBN: 978-987-3806-30-8. Pp.73-79.

- Análisis Comparativo de Modelos de Clasificación de Minería de Datos (Data Mining). Su aplicación en la predicción de perfiles de alumnos en riesgo de deserción. Proyecto ProInce C176. 2015-2016.
- Uso de Minería de Datos para acelerar la recuperación de documentos. Proyecto ProInce C205. 2017-2018.

"Las herramientas de Query Ad-Hoc, poderosas como ellas son, sólo pueden ser entendidas y usadas efectivamente por un pequeño porcentaje de los potenciales usuarios del data warehouse del negocio".

Ralph Kimball.

Capítulo 2.

2. Estado de Situación

En el año 2015 uno de los diarios digitales de mayor consulta de la Argentina entregaba una nota cuyo título esgrimía "*Más matriculados, pero pocos graduados en las universidades*" [19]. Esta nota se refería a que mientras la población universitaria argentina creció el 22,5% en la última década, quienes terminan los estudios universitarios son apenas tres de cada diez ingresantes. Número que en Brasil asciende a cinco, y en Chile a seis. A esta conclusión llegó un informe publicado por el Centro de Estudios de Educación Argentina (CEEA), que considera la evolución de la matrícula y la graduación universitaria argentina entre 2003 y 2012 sobre la Base del Anuario de Estadísticas Universitarias, el Departamento de Información Universitaria, la Unesco, y la Organización para la Cooperación y el Desarrollo Económicos (OCDE).

Desde hace varios años, la deserción universitaria se ha convertido en un tema recurrente de ser tratado e investigado. Poder saber, a priori, si un alumno puede llegar a desertar, proporcionaría una herramienta importante para mejorar la toma de decisiones, ya que brindaría un sustento teórico a las acciones que se podrían proponer para desarrollar políticas de retención de los mismos.

Los rápidos avances en la tecnología de recopilación y almacenamiento de datos han permitido a las universidades acumular grandes cantidades de datos. Sin embargo, la extracción de información útil ha resultado sumamente difícil. A menudo, las herramientas y técnicas tradicionales de análisis de datos no pueden utilizarse debido al gran volumen del conjunto de datos, y otras, por la naturaleza "no tradicional" de los datos, hace que los enfoques "*tradicionales*" no se puedan aplicar aunque el conjunto de los datos sea relativamente pequeño.

La Minería de Datos es una tecnología que combina métodos de análisis de datos tradicionales con sofisticados algoritmos para procesar grandes volúmenes de datos. A su vez la Minería de Datos Educacionales, es una disciplina que se desprende de la MD tradicional y está dedicada a desarrollar métodos para explorar los datos provenientes de ambientes relacionados a la educación, para tratar de entender mejor a los estudiantes, profesores y actores en sus entornos educacionales, con el fin de mejorar el proceso educativo del aprendizaje. [1]

En algunos trabajos [14-17], entre otros, se observa como varias universidades nacionales y extranjeras, al disponer de sistemas computarizados con gran cantidad de información referida a los datos personales y de rendimiento académico de sus alumnos, utilizaron algoritmos de MD para obtener perfiles de alumnos con riesgo a abandonar sus estudios.

2.1. Sistemas OLTP vs OLAP

Descubrir información oculta en Bases de Datos (BD) transaccionales, también conocidas como Bases OLTP[11], es una de las tareas principales de la minería de datos. La MD es un proceso se encuadra dentro de la tecnología más amplia conocida como "*Descubrimiento de Conocimiento en Bases de Datos*" [5]. Y tienen por objetivo procesar y analizar la información para encontrar patrones repetitivos, tendencias, o reglas que expliquen el comportamiento de los datos en un determinado contexto.

Los sistemas OLTP trabajan con Bases de Datos relacionales y están diseñados para almacenar y modificar continuamente datos de la operación diaria de un negocio. Sin embargo, no todos los datos almacenados son relevantes para los reportes o análisis que los usuarios de las empresas buscan. Es decir, los OLTP son limitados para la toma de decisiones, y por lo tanto se requiere de un trabajo adicional sobre los datos para darles un formato o estructura que permita capturar el valor de la información [20].

[11] OLTP: Sigla en inglés de Procesamiento de Transacciones En Línea (On Line Transaction Processing)

Por los años 1990 aparece el concepto de OLAP[12], que son sistemas para extraer información relevante del negocio desde BD complejas, analizando la información y entregando una respuesta rápida a las consultas de los usuarios [5]. Los sistemas OLAP trabajan en línea con datos resumidos en una estructura multidimensional, a diferencia de los OLTP que tienen una estructura relacional. Por este motivo las bases OLAP obtienen resultados más ágiles de las consultas. Esta tecnología incluye técnicas de análisis de datos como pueden ser el resumen, la consolidación o la agregación, así como la posibilidad de ver la información desde distintas perspectivas. Este tipo de arquitectura fue diseñada específicamente para cubrir la brecha que presentan los sistemas OLTP para la entrega de información a los usuarios, consolidando las fuentes de información en una base única y consistente, en la cual se posee la información de forma agregada para responder rápidamente a las preguntas del negocio [5].

La información en los sistemas OLAP se obtiene desde múltiples fuentes y dispone en variados formatos, ya sean tablas, gráficos o reportes para que la utilicen áreas del negocio como ventas, marketing, control de gestión, etc. Las principales diferencias entre ambos sistemas (OLTP y OLAP) se resumen en la Tabla 1.

OLTP	OLAP
Orientado a la operación diaria y al funcionamiento de las aplicaciones transaccionales.	Orientado al usuario que toma las decisiones del negocio.
La información se almacena en bases de datos relacionales, con datos normalizados, y se accede principalmente para insertar, modificar o eliminar datos.	La información se almacena en estructuras multidimensionales, y se accede para hacer consultas.
Muchos usuarios acceden a modificar los datos constantemente.	Los datos permanecen estáticos hasta su próxima actualización, los usuarios sólo acceden para lectura de los datos.
El tamaño de la base de datos incrementa rápidamente, por lo cual se le da preferencia a los datos más actuales. Se busca tener la mínima redundancia posible y consistencia en los datos.	El tamaño de la base de datos puede ser muy grande debido a la redundancia de datos. Los datos históricos y actuales son igual de importantes.

Tabla 1: Diferencias entre OLTP y OLAP

[12] OLAP: Sigla en inglés de OnLine Analytical Processing.

A partir de lo anteriormente planteado, este trabajo tiene como objetivo realizar un estudio comparativo y realizará el análisis a partir de información almacenada en un Almacén de Datos, construido años atrás por el DIIT. Mediante el empleo de estas técnicas, se pretende determinar cuál de ellas es más eficiente en la formulación de patrones[13] predictivos respecto de alumnos que puedan llegar a abandonar sus estudios prematuramente. Los patrones pueden ser utilizados, además, realizar observaciones futuras o motivar la búsqueda de explicaciones a las observaciones pasadas, capacidades fundamentales para mejorar el comportamiento en relación a la deserción universitaria. El rendimiento de los algoritmos se medirá a través del análisis de la especificidad, sensibilidad, exactitud en la precisión, y el análisis de una Matriz de Confusión (MC). Se utilizará para este propósito una herramienta bajo licencia GPL (General Public License) denominada WEKA. Weka es una suite consistente en un conjunto de librerías diseñadas en JAVA por la universidad de Waikato (Nueva Zelanda) con diferentes métodos y técnicas de Minería de Datos [21]

A continuación se resumen los conceptos fundamentales empleados en el presente texto: DW, KDD y, sobretodo, Minería de Datos, así como sus principales características. Se continúa con una breve descripción de la herramienta utilizar y del estado de arte que se encuentra hoy la MDE.

2.2. Data Warehousing

Se define Data Warehousing al proceso por el cual las organizaciones extraen sentido y significado de sus Datos, a través del uso de un Almacén de Datos o Data Warehouse. Mientras que la definición de OLAP es, una tecnología que ayuda a los trabajadores del conocimiento a realizar con rapidez sus procesos empresariales y la toma de decisiones; permite a analistas y ejecutivos analizar los datos rápidamente, de forma interactiva y teniendo en cuenta varias entidades del negocio [22].

[13] Se entiende como patrón, a los hechos o las cosas recurrentes. Estos factores o elementos se repiten con previsibilidad y, por lo tanto, pueden funcionar como modelo para producir determinada cosa a partir de ellos.

La definición de Almacén de Datos ha sido planteada por varios autores. Estas definiciones enfatizan el uso de la información dentro de una organización para apoyar a la toma de decisiones gerenciales. Un AD es una colección de datos orientado a temas, integrado, no volátil, de tiempo variante, que se usa para el soporte del proceso de toma de decisiones gerenciales [23]. Un DW, en su simple percepción, no es más que una colección de las partes claves de información utilizada para administrar y especialmente para la toma de decisiones gerencial. Muchas de las definiciones se concentran en los datos, pero por definición de un AD es más que eso. Son también los procesos involucrados para obtener los datos desde las fuentes a las tablas, y desde las tablas al analista. En otras palabras un AD es el dato (metadato/ hecho/ dimensión/ agregación) y los administradores de proceso (carga/ almacén/ consulta) que hacen disponible la información, permitiendo a las personas tomar decisiones [24].

El DIIT viene desarrollando investigaciones desde el año 2009 en diferentes proyectos orientados a la Minería de Datos y a la construcción de un AD, con el objetivo de mejorar la toma de decisiones y lograr mejores resultados organizacionales. Estas tecnologías permiten realizar análisis más detallados de la información, y uno de las metas principales que se persigue es intentar disminuir los índices de deserción universitaria usando técnicas de MD [25]. Se presentaron varios proyectos en el marco del programa de Incentivos a Docentes Investigadores de la Secretaría de Políticas Universitarias (ProInce), en los que quién escribe participó como integrante [26-28]. El resultado de estas investigaciones permitió construir el primer AD departamental o Data Mart (DM) de la UNLaM. A continuación se realiza una breve descripción del mismo, que es la base de la que se extrajeron los datos para esta comparación.

La construcción del AD utilizó como origen de datos las tablas que componen el sistema OLTP conocido como el Sistema de Gestión Académica de Alumnos SIU-Guaraní[14]. Este es un sistema de gestión académica que registra y administra todas las actividades académicas de la Universidad y sus Facultades, desde que los alumnos ingresan como aspirantes hasta

[14] http://www.siu.edu.ar/nuestras-soluciones/gestion-academica-2/siu-guarani-2

que obtienen el diploma. Fue concebido para administrar la gestión de alumnos en forma segura.

El SIU (*Sistema Inter Universitario*) es un Consorcio de Universidades que desarrolla soluciones informáticas y brinda servicios para el Sistema Universitario Nacional y distintos organismos de gobierno. Su objetivo es contribuir a mejorar la gestión de las instituciones, permitiéndoles contar con información segura, íntegra y disponible, optimizar sus recursos y lograr que el software sea aprovechado en toda su potencialidad. En la figura 1 se muestra la totalidad de los módulos que componen al SIU-Guaraní.

Figura 1: Módulos que componen el SIU-Guaraní.

Como ya se mencionó la definición de Almacén de Datos ha sido planteada por varios autores. Esta definición enfatiza el uso de la información dentro de una organización para apoyar a la toma de decisiones gerenciales. Sin embargo fué Bill Inmon[15], ingeniero especializado en BD [29], quien acuñó por primera vez el término para hacer referencia a un almacén de información temática, orientado a cubrir las necesidades de aplicaciones de los Sistemas de Soporte de las Decisiones (DSS[16]) y los Sistemas de Información para Directivos (EIS[17]), que permite acceder a la información corporativa para la gestión, control y apoyo a la toma de decisiones. Este autor en [30], da la siguiente definición de un DW "*...es una colección de datos orientado a temas, integrado, no volátil, de tiempo variante, que se usa para el soporte del proceso de toma de decisiones gerenciales...*". A continuación se da una breve explicación de cada uno de los términos utilizados:

[15] Inmon, Bill. http://www.inmoncif.com/about/
[16] **DSS** por sus siglas en inglés: *Decision Support System.*
[17] **EIS** por sus siglas en inglés*: Executive Information Systems.*

- **Orientado a Temas**

  Una primera característica del DW es que la información se clasifica en base a los aspectos que son de interés para la organización. Siendo así, los datos tomados están en contraste con los clásicos procesos orientados a las aplicaciones. El ambiente operacional se diseña alrededor de las aplicaciones y funciones tales como datos personales de los alumnos, notas de cursadas, notas de exámenes, etc. para una institución universitaria. Las BD combinan estos elementos en una estructura que acomoda las necesidades de la aplicación. En el ambiente DW, se organiza alrededor de sujetos tales como alumnos, profesores, cátedras, cursos y comisiones.

- **Integración**

  El aspecto más importante del AD, es que la información almacenada esté siempre integrada. La integración de datos se muestra de muchas maneras, en convenciones de nombres consistentes, en la medida uniforme de variables, en la codificación de estructuras consistentes, en atributos físicos de los datos consistentes, fuentes múltiples y otros. A través de los años, los diseñadores de las diferentes aplicaciones han tomado sus propias decisiones sobre cómo se debería construir una aplicación. Los estilos y diseños personalizados se muestran de muchas maneras, diferenciándose en la codificación, las convenciones de nombramiento y otros.

  - ✓ **Codificación**. Los diseñadores de aplicaciones codifican el campo género en varias formas. Un diseñador representa el atributo como una "M" y una "F", otros como un "1" y un "0", otros como una "X" y una "Y" e inclusive, como "masculino" y "femenino". Es importante que sea de cualquier fuente de donde venga, el atributo género debe llegar al DW en un estado integrado uniforme.
  - ✓ **Medida de atributos.** Los diseñadores de aplicaciones miden las unidades de medida en una variedad de formas. Unos pueden expresarlas en Sistema Internacional, otros en el Anglo-sajón, etc. Al dar medidas a los atributos, la transformación traduce las diversas unidades de medida usadas en las diferentes Bases de Datos para uniformarlas unidades utilizando un sistema estándar común.
  - ✓ **Convenciones de Nombramiento.** Cuando un mismo elemento es frecuentemente referido por nombres diferentes en las diversas aplicaciones, el proceso de transformación asegurará que se use preferentemente el nombre de usuario.

- ✓ **Fuentes Múltiples.** Un mismo elemento puede provenir de fuentes múltiples. En este caso, el proceso de transformación deberá asegurar que la fuente apropiada sea la utilizada, documentada y movida al depósito. Cualquiera que sea la forma del diseño, el resultado es el mismo, la información necesita ser almacenada en el DW en un modelo globalmente aceptable y singular, aun cuando los sistemas operacionales subyacentes almacenen los Datos de manera diferente.

- **De Tiempo Variante**

  Toda la información del DW es requerida en algún momento. Esta característica básica de los dato, es muy diferente de la información encontrada en el ambiente operacional. En éstos, la información se requiere al momento de acceder. En otras palabras, en el ambiente operacional, cuando se acceda a una unidad de información, se espera que los valores requeridos se obtengan a partir del momento de acceso. Los Datos históricos son de poco uso en el procesamiento operacional. La información en el DW por el contraste, debe incluir los Datos históricos para usarse en la identificación y evaluación de tendencias.

2.3. Arquitectura de un DW.

Una Arquitectura DW establece el marco de trabajo, estándares y procedimientos a seguir para su construcción [28]. El objetivo de las actividades de la arquitectura es simple, integrar al DW las necesidades de información de la organización. En la Figura 2 se muestra la arquitectura empleada en los proyectos ProInce del DIIT.

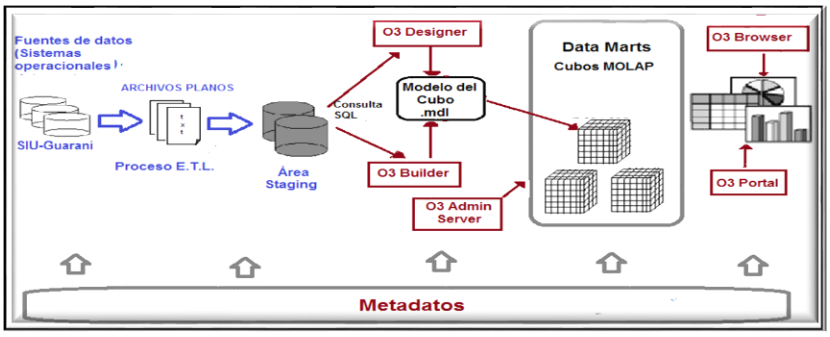


Figura 2: Arquitectura del Data Warehouse del DIIT [28].

Como se mencionó anteriormente, el sistema operacional del que se extraen los datos es el SIU-Guaraní. A través de consultas del tipo SQL se exportan los registros en

formato plano (texto) en archivos del tipo .txt o csv[18]. Los procesos ETL[19] son un término estándar que se utiliza para referirse al movimiento y transformación de datos. Se trata del proceso que permite a las organizaciones mover Datos desde múltiples fuentes, reformatearlos y cargarlos en otra BD que puede ser un DW o DM con el objeto de analizarlos. En definitiva, el principal objetivo de este proceso es facilitar el movimiento de los Datos y la transformación de los mismos, integrando los distintos sistemas y fuentes. Aunque existen muchas metodologías distintas para el desarrollo de estas bases, el DIIT eligió la metodología propuesta por Ralph Kimball[20] [31]. Esta metodología, denominada "*Bottom-Up*", comienza por el desarrollo de un DM para construir luego un DW como unión de estos. El autor define a un DM como: "*…una porción de la torta completa que representaría el DW…*". Kimball sugiere utilizar esta metodología, donde la información se extrae de los sistemas transaccionales para ser cargada en diferentes DM, cada uno de los cuales son independientes, están modelados en forma dimensional, y tienen foco departamental. Estos DM podrían ser implementados con tecnología ROLAP o MOLAP.

2.4. MOLAP (Multidimensional OLAP)

La arquitectura MOLAP usa Bases de Datos multidimensionales para proporcionar el análisis. MOLAP es, quizás, la forma OLAP original. Generalmente, si sólo se habla de OLAP, se entenderá que se trata de MOLAP. La "M" se puso más adelante para diferenciarlo de otros términos como ROLAP que se explicará más adelante. La arquitectura MOLAP se trata de cubos, es una arquitectura de datos diferente y casi siempre propietaria, la información se almacena multidimensionalmente, para poder ser visualizada en varias dimensiones de análisis. En las investigaciones del DIIT se usó esta tecnología empleando la herramienta *Ideasoft O3™ Business Intelligence*[21]. El sistema MOLAP utiliza una arquitectura de dos niveles, la Base de Datos multidimensionales, y el motor analítico. La Base de Datos multidimensional es la encargada del manejo, acceso y obtención del Dato, más adelante se ampliará este concepto.

[18] por sus siglas en inglés: *comma-separated values.*
[19] por sus siglas en inglés: *Extract, Transform and Load* («extraer, transformar y cargar»)
[20] Ralph Kimball. http://www.kimballgroup.com/
[21] https://www.ideasoft.biz

2.5. ROLAP (Relational OLAP)

La premisa de los sistemas ROLAP es que las capacidades OLAP se soportan mejor contra las Bases de Datos Relacionales (BDR). En ROLAP se utiliza una arquitectura de tres niveles. La BD relacional maneja el almacenamiento de datos, el motor OLAP proporciona la funcionalidad analítica, y alguna herramienta especializada es empleada para el nivel de presentación. Los usuarios finales ejecutan sus análisis multidimensionales, a través del motor ROLAP, que transforma dinámicamente sus consultas a SQL. Se ejecutan estas consultas SQL en las BDR, y sus resultados se consolidan mediante tablas cruzadas y conjuntos multidimensionales para devolverlos luego a los usuarios. La arquitectura ROLAP es capaz de usar datos pre-calculados, si estos están disponibles, o de generar dinámicamente los resultados desde los datos elementales, si es preciso. Antes de cargar los datos a las Bases MOLAP, se realiza un proceso que se encarga de la migración de los datos de origen. Estos vienen en un formato de archivos planos desde las tablas del SIU-Guaraní y se almacenan en unas tablas intermedias. Este proceso abarca las fases de diseño de las transformaciones, sumarización[22], conversión, controles, comprobación de Datos y verificación. Los datos se guardan en el Área intermedia de datos (DSA[23]). En esta área de almacenamiento auxiliar, es donde se le aplican a los datos un conjunto de procesos que limpian, transforman, combinan y preparan a los mismos para ser cargados al repositorio final. En el desarrollo de los proyectos ProInce se utilizó un servidor SQL Server 2008 R2 junto con *SQL Server Integration Services* para realizar todos los procesos ETL. En la siguiente figura se observa el modelo físico del AD del DIIT.

[22] La sumarización o agregación muestra los datos de una manera más resumida, permitiendo, precisamente, calcular valores agregados, que no son los datos directos registrados, sino datos derivados de ellos

[23] por sus siglas en inglés: *Data Staging Area.*

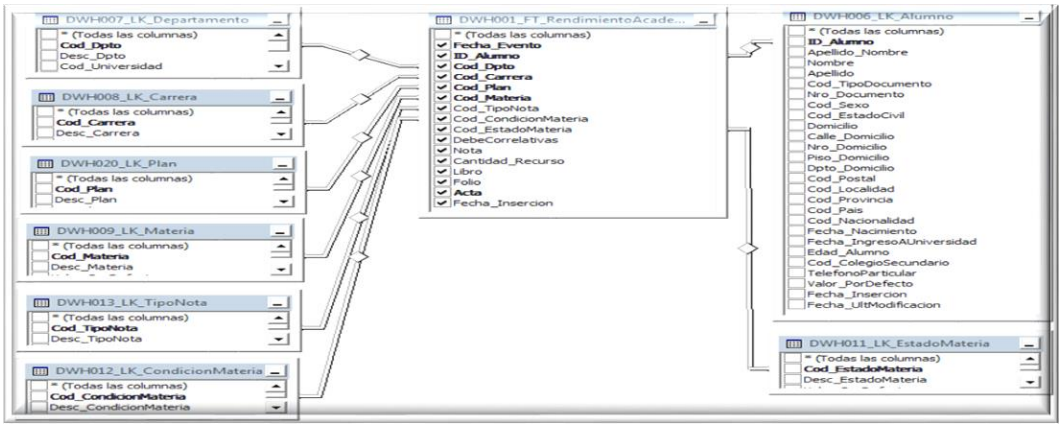


Figura 3: Estructura de tablas del Área intermedia de Datos [28]**.**

Los Metadatos (o Diccionario de Datos), son la información respecto a los objetos de la Base de Datos. Ésta describe el significado de la estructura de los datos del negocio, así como también la manera en que éstos son creados, accedidos y usados, lo que asegurará que los datos del negocio sean usados de manera completa y consistente. Una vez que se logra realizar el diseño conceptual del repositorio de datos, se conciben las estructuras multidimensionales que responden a los requerimientos generales de información.

2.6. Componentes del Modelo Multidimensional

El Modelo Entidad-Relación (MER), es una técnica poderosa para el diseño de sistemas transaccionales en el entorno de las BDR. Éste permite la normalización de la estructura de datos física, obteniéndose un diseño sin redundancias en los datos y ocupando el menor espacio de almacenamiento. Sin embargo, no contribuye en la habilidad del usuario al momento de consultar la BD [32].

En contraposición el Modelo Multidimensional (MM) es una técnica de diseño que busca organizar los datos en un marco estándar, que sea intuitivo y de acceso rápido [31]. Este modelo es mucho menos riguroso en cuanto a organización que el MER. Le permite a analistas, y diseñadores, más flexibilidad en el diseño para lograr un mayor desempeño, y optimizar la recuperación de la información desde un punto de vista más cercano al usuario final. Modela las particularidades de los procesos que ocurren en una organización, dividiéndolos en Mediciones y Dimensiones. Las medidas son, en su mayoría, numéricas, y se les denomina Hechos. Alrededor de estos hechos existe un contexto que describe en qué condiciones, y en qué momento, se registró este hecho. Aunque la dimensión se ve como un todo, existen registros lógicos de diferentes características que describen un hecho, por ejemplo, si el hecho referido es el registro de la nota de un alumno, se podría dividir el entorno que rodea al hecho de la nota obtenida en, la materia rendida, el alumno que la generó, el curso, y la fecha en que se realizó.

A estas divisiones se le denomina Dimensiones y, a diferencia de los hechos que son numéricos, éstos son fundamentalmente textos descriptivos. A esta tecnología se la denomina también Cubos. Por ejemplo, normalmente, se define un punto en el espacio por la intersección de sus coordenadas X, Y, Z. Si se le asignan valores a esas coordenadas, por ejemplo: X representa Alumno, Y representa el tiempo y Z representa la asignatura. Y luego se considera la siguiente combinación X = Alumno (Pedro Gómez); Y = Fecha (5/12/2013); Z = Materia (Base de Datos), se obtendrá un punto en el espacio el cual se podría definir como la Nota Obtenida en el examen de ese día. En la figura 4 se puede observar una representación gráfica de un cubo. Una característica importante que ha de definirse en los Modelos multidimensionales es la Granularidad. Este concepto representa el nivel de detalle en el que se almacenan los registros. La granularidad condiciona las posibilidades analíticas en el modelo resultante, extendiendo el examen en detalle hasta un nivel definido.

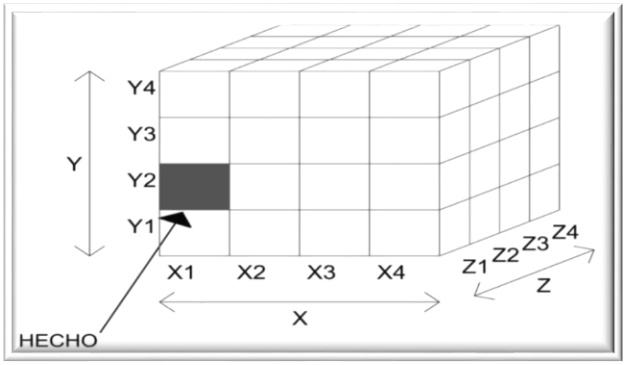


Figura 4: Representación gráfica de un cubo

2.7. Definición de la granularidad

Una vez que se ha definido el proceso de negocio, la siguiente tarea será la definición de la granularidad, o lo que es lo mismo, hasta qué nivel de detalle se quiere alcanzar en el modelo de DM y más concretamente en la tabla de hechos. Lo más recomendable en la metodología de Kimball es desarrollar el modelo en torno a una granularidad baja obtenida a partir del proceso de negocio. Es decir, el objetivo es estructurar el modelo en torno a una información lo más detallada posible, de tal manera que ésta ya no se pueda desglosar [31].

Para el DM del DIIT, la granularidad más apropiada es la que consiste en tener un registro por cada alumno que haya cursado una asignatura en un año determinado, de manera que se pueda estructurar la información en torno a los Datos que se puedan obtener del alumno, entre los que pueden encontrarse los siguientes:

- ✓ Fecha del Evento
- ✓ Carrera
- ✓ Plan de Estudio
- ✓ Materia
- ✓ Nota
- ✓ Condición
- ✓ Estado la Materia
- ✓ Etc.

Este modelo permitirá conocer la evolución de los alumnos a lo largo del tiempo dentro, en este caso, del DIIT. El DW departamental final construido o DM está compuesto por la Tabla de Hechos (F*act Table*), y tablas más pequeñas, que definirán las n-dimensiones o aperturas del Cubo, llamadas Tablas de Dimensiones. Las medidas, como se expresó

anteriormente, se registran en la Tabla de Hechos, siendo la clave de esta tabla, la combinación de las múltiples claves foráneas que hacen referencia a las dimensiones que describen la ocurrencia de este hecho. En otras palabras, cada una de las claves externas en la Tabla de Hechos, se corresponden con la clave primaria de una dimensión. A continuación se detallan los conceptos de Tablas de Hechos y Dimensiones [27-28].

2.8. Tablas de Hechos

Las T*ablas de Hechos*, representan los procesos que ocurren en la Organización, Y son independientes entre sí (no se relacionan unas con otras). En éstas, se almacenan las medidas numéricas de la organización. Cada medida, se corresponde con una intersección de valores de las dimensiones y generalmente se trata de cantidades numéricas, las que son aditivas y continuamente evaluadas. La razón de estas características, es que facilitan que todos los registros que involucran una consulta, sean comprimidos más fácilmente y se pueda dar, con rapidez, respuesta a una solicitud que abarque gran cantidad de información. La clave de la tabla de hechos, es una clave compuesta, debido a que se forma de la composición de las claves primarias de las tablas dimensionales a las que está unida.

2.9. Tablas de Dimensiones

Una *Tabla de Dimensión* contiene, por lo general, una clave simple y un conjunto de atributos que describen la dimensión. En dependencia del esquema multidimensional que se siga, pueden existir atributos que representen claves foráneas de otras tablas de dimensión. Es decir: establecen una relación de esta tabla con otra dimensión, como se verá más adelante en los tipos de esquemas. Las tablas de dimensión, como se expresó anteriormente, son las que alimentan a las tablas de hechos. La clave de un hecho es la composición de las claves de las dimensiones que están conectados a esta, por lo tanto, los atributos que conforman las tablas de dimensiones también describen el hecho. Los atributos dimensionales son fundamentalmente textos descriptivos, estos desempeñan un papel determinante; son la fuente de gran parte de todas las necesidades que deben cubrirse. Además, sirven de restricciones en la mayoría de las consultas que realizan los usuarios.

A continuación se hace mención a los tipos de esquemas existente:

2.10. Esquema en estrella:

Es el esquema de referencia para los DM. El corazón del esquema es la Tabla de Hechos, que son las operaciones que se van a analizar, que han sido registradas por las transacciones de los sistemas OLTP y que son, numerables, cuantificables numéricamente, o ambas. Cada registro de la tabla de hechos representa un hecho. La finalidad del DM consiste en permitir el análisis de los hechos a través de los ejes de análisis, llamados dimensiones. En el esquema en estrella cada dimensión se reduce a una tabla y cada registro de la tabla de hechos está vinculada a cada dimensión. La característica principal del esquema en estrella es el hecho de *aplanar* cada dimensión en una tabla única como resultado de la *desnormalización*. En el sistema OLTP, cada dimensión es objeto de multitud de tablas *normalizadas*. Por ejemplo el producto se modela, como mínimo, con tres tablas.

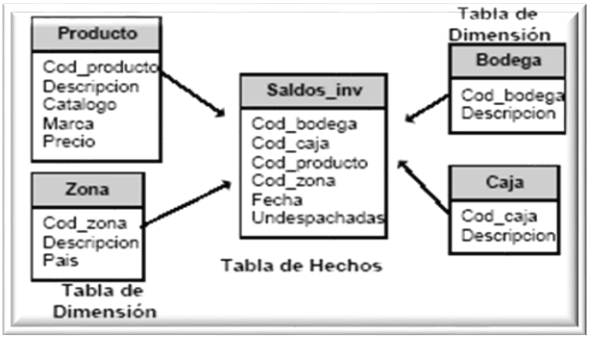


Figura 5: Ejemplo de un esquema estrella

Mientras que la modelización OLTP *normalizada* se concibe para la escritura. La modelización en estrella se crea para la lectura. Solamente debe crearse y actualizarse el DM. Cuando una categoría "A" de productos se desplaza a una nueva categoría en el sistema operacional, la actualización del esquema en estrella debe efectuarse para todos los productos de la subcategoría "A".

2.11. Esquema en Copo de Nieve.

En el caso de un DM que se actualice de forma periódica, la *desnormalización*[24] no debería suponer un problema. No obstante, a menudo se prefiere un esquema semi-desnormalizado. Un esquema de este tipo se conoce como Esquema en Copo de Nieve.

[24] Es el proceso de procurar optimizar el desempeño de una base de datos por medio de agregar datos redundantes. Para saber más se puede ingresar en: https://www.ibm.com/support/knowledgecenter/es/SSEPEK_10.0.0/intro/src/tpc/db2z_denormalizationfoperformance.html

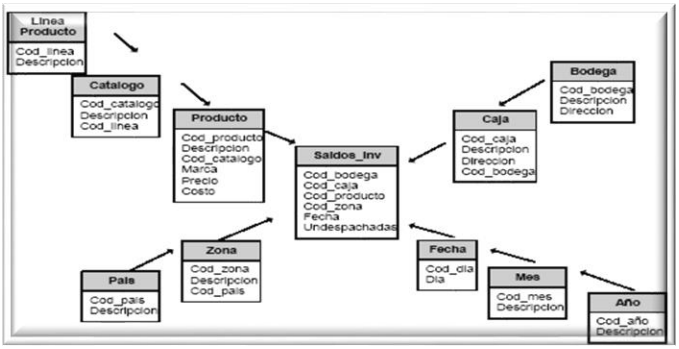


Figura 6: Ejemplo de un esquema copo de nieve

2.12. Data Warehouse y Data Mining

Como ya se mencionó, un DW almacena información heterogénea, de las BD, para que los usuarios consulten sólo un único aspecto. Las respuestas que un usuario consigue en una consulta dependen de los volúmenes del DW o DM. Estas estructuras, en general, no intentan extraer la información de los datos almacenados, si no que se encargan de organizar los datos para soportar funciones de administración. Es la Minería de Datos quien intenta extraer la información útil, así como predecir las tendencias de los datos. Si bien no es necesario construir un DW o DM para hacer MD, ya que también pueden aplicarse estas técnicas a las BDR. Sin embargo, como un DW o DM estructuran los datos de tal manera que facilita la utilización de técnicas de MD, por lo que en muchos casos es muy deseable tener un AD para llevar a cabo los procesos [5].

2.13. Descubrimiento de Conocimiento en Bases de Datos

Con la necesidad de poder manejar grandes cantidades de datos, surge un área de estudio que se denomina "Descubrimiento del Conocimiento en Bases de Datos" o KDD. En la literatura actual se puede encontrar un gran número de definiciones acerca de este tema, una de la más completas es la siguiente: *"…el Descubrimiento de Conocimiento en Bases de Datos es un campo de la inteligencia artificial de rápido crecimiento, que combina técnicas del aprendizaje de máquina, reconocimiento de patrones, estadística, Bases de Datos, y visualización para automáticamente extraer Conocimiento (o información), de un nivel bajo de Datos (Bases de Datos)…*" [7].

El KDD es un proceso que consta de una serie de etapas consecutivas, y funciona de forma iterativa e interactiva. Iterativa, ya que es posible regresar desde cualquier etapa a una

anterior para ajustar los parámetros o supuestos previos, e interactiva pues el usuario experto del negocio tiene que estar presente para aportar con su conocimiento en la preparación de los datos y en la validación de los resultados que se obtengan durante el proceso. Cabe aclarar que KDD supone la convergencia de distintas disciplinas de investigación; algunas tales como el aprendizaje automático, estadística, inteligencia artificial, sistemas de gestión de base de datos, técnicas de visualización de datos, los sistemas para el apoyo a la toma de decisión (DSS) o la recuperación de información, entre otras [33]. Las etapas de este proceso, según Hernández en [5], se muestran en la Figura 7 y se describen en el próximo apartado.

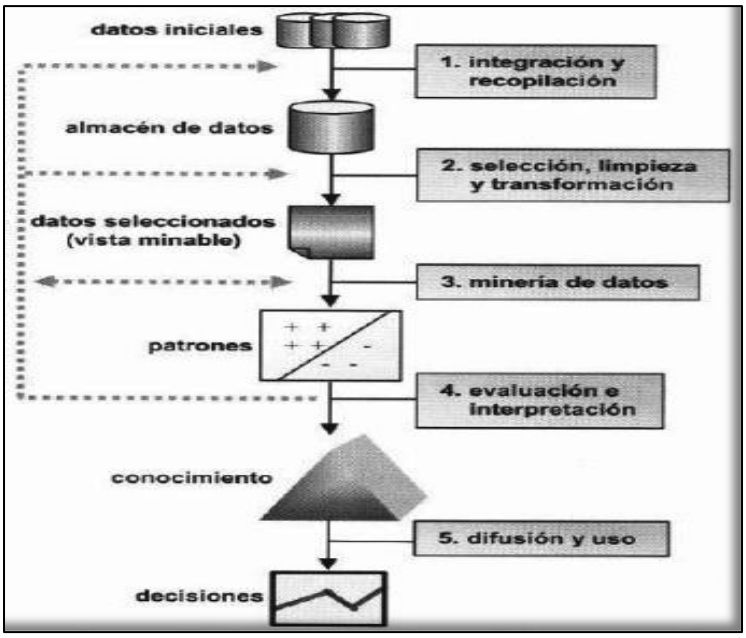


Figura 7: Proceso de KDD. [5]

2.14. Metodología para realizar MD

Para implementar una tecnología en un área determinada, se requiere de una metodología. Los métodos son definidos a partir de las experiencias y tomando lo mejor de los procedimientos más exitosos o populares. Las metodologías incluyen descripciones de las fases normales de un proyecto, las tareas necesarias en cada fase, y una explicación de las relaciones entre las tareas. Para el caso de proyectos de implementación de MD, hay varias metodologías entre las que se encuentran, *CRISP-DM* [25], *SEMMA*[26], *Catalyst* [27] y *KDD*, entre

[25] por sus siglas en inglés de: *Cross-Industry Standard Process for Data Mining*.
[26] Acrónimo de Sample, Explore, Modify, Model, assess.
[27] Conocida como P3TQ (Product, Place, Price, Time, Quantity)

otras. Estas metodologías permiten llevar a cabo el proceso de MD en forma sistemática. Asimismo, ayuda a las organizaciones a entender el proceso de descubrimiento de conocimiento y proveen una guía para la planificación y ejecución de los proyectos. A su vez, define las fases del proceso, las tareas específicas que deben realizarse, y cómo llevarlas a cabo.

Para este trabajo se eligió la metodología *KDD*, que consta de cinco fases diferenciadas, como lo describe Hernández [5], en particular se desarrollará principalmente una de ellas, la Minería de Datos. También se describirán las demás fases que componen este proceso, ya que no se puede hablar de M datos sin tener conocimientos previos de los datos en sí, y de cómo se seleccionan, limpian, transforman y almacenan los mismos, por supuesto que una vez concluida la minería, será necesario saber la manera de evaluar, e interpretar las conclusiones sacadas del estudio.

2.15. Fases en el proceso KDD.

2.15.1. Fase de integración y recopilación:

En esta primera fase del proceso de KDD, se trata de decidir dónde se obtendrán los datos que se utilizarán más adelante, es decir, qué fuentes de información van a resultar útiles. Las bases de datos y las aplicaciones basadas en el procesamiento tradicional, o OLTP, son suficientes para cubrir las necesidades diarias de una organización, como el armado de las listas de alumnos de los cursos, volcar las notas parciales y finales, etc.

Sin embargo, resultan insuficientes para otras funciones más complejas, como el análisis, la planificación y la predicción, es decir para tomar decisiones estratégicas a largo plazo [31]. La integración de datos, entonces, significa combinar información de múltiples procedencias, incluyendo diferentes tecnologías de BD, que podrían tener diferentes contenidos y formatos. La inconsistencia en el formato puede llevar una redundancia, e inconsistencia, en los atributos y valores de los datos. Normalmente cuando se trabaja en un problema de proceso de descubrimiento, es necesario primero formar un único conjunto con todos los datos que provienen de distintas fuentes.

2.15.2. Fase de selección, limpieza y transformación:

La calidad del conocimiento descubierto no solo depende del algoritmo de minería utilizado, sino también de la calidad de los datos minados. Pero, además de la irrelevancia, 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 anómalos pueden representar errores en los datos, o bien pueden ser valores correctos, que son simplemente diferentes a los demás. Algunos algoritmos de minería de datos ignoran estos datos, otros los descartan considerándolos *ruido* o *excepciones*, pero otros son muy sensibles y el resultado se ve claramente perjudicado por ello. Las transformaciones consisten principalmente en modificaciones sintácticas, llevadas a cabo sobre datos, sin que supongan un cambio para la técnica de minería aplicada. En algunos casos, las transformaciones discretas de los datos [33], tienen la ventaja de mejorar la comprensión de las reglas descubiertas al transformar los datos de bajo nivel, en datos de alto nivel, y también reduce significativamente el tiempo de ejecución del algoritmo de búsqueda. Su principal desventaja es que se puede reducir la exactitud del conocimiento descubierto, debido a que puede causar la pérdida de alguna información.

2.15.3. Fase de Minería de Datos:

Está fase está conformada por un conjunto de técnicas y algoritmos que sirven para hacer análisis de los conjuntos de datos, extrayendo patrones y relaciones entre ellos, convirtiéndolos en información valiosa y útil para quienes toman las decisiones. Los algoritmos que más se destacan son los de regresión, agrupamiento o clustering, y clasificación. Es muy importante darse cuenta que el uso de DM se debe entender como un apoyo para los analistas, y no reemplaza al conocimiento que tienen los expertos del negocio, ni elimina la necesidad de entender los datos. La MD no funciona por sí sola, ya que los patrones que se encuentren en los datos deben ser interpretados y validados por el usuario, constatar si responden a las consultas del negocio, y si son aplicables en el mundo real.

2.15.4. Fase de evaluación e Interpretación:

En esta fase se intenta identificar verdaderamente patrones interesantes que representan conocimiento, usando diferentes técnicas incluyendo análisis estadísticos y lenguajes de consultas. Por lo general, para entrenar y probar un modelo, se parten los datos en dos

conjuntos: el conjunto de entrenamiento (*training set*) y el conjunto de prueba o test (*test set*). Esta separación es necesaria para garantizar que la validación de la precisión del método es una medida independiente. Si no se usan conjuntos diferentes de entrenamiento y prueba, la precisión del modelo será sobreestimada, es decir que existirán estimaciones muy optimistas. El método que se usa normalmente es la *Validación Cruzada* con ***n*** pliegues (en inglés *n-fold cross validation*). En este método los datos se dividen aleatoriamente en ***n*** grupos o fold. Un grupo se reserva para el conjunto de prueba y con los ***n-1*** restantes (juntando todos sus datos) se construye un modelo y se usa para predecir el resultado de los datos del grupo reservado. Este proceso se repite ***n*** veces, dejando cada vez un grupo diferente para la prueba. Esto significa que se calculan ***n ratios***[28] de error independientes. Finalmente, se construye un modelo con todos los datos y se obtienen sus ***ratios de error*** y precisión promediando los ***n ratios de error*** disponibles.

2.15.5. Fase de difusión y uso:

Consiste en entender los resultados del análisis y sus implicaciones y puede llevar a regresar a algunos de los pasos anteriores. Una vez construido y validado el modelo puede usarse principalmente con dos funcionalidades: para que un analista recomiende acciones basándose en el modelo y sus resultados, o bien para aplicar al modelo a diferentes conjuntos de datos. Tanto en el caso de una aplicación manual o automática del modelo, es necesario su difusión, es decir que se distribuya, y se comunique a los posibles usuarios ya sea mediante los cauces habituales dentro de la organización, las reuniones, intranet, etc. El nuevo conocimiento extraído debe integrar el "*Know-how"* de la organización. También es importante medir lo bien que en el modelo evoluciona. Aun cuando el modelo funcione bien se deben, continuamente, comprobar las prestaciones del mismo. Esto se debe principalmente a que los patrones pueden cambiar. Por ello tanto, el modelo deberá ser monitorizado, lo que significa que de tiempo en tiempo, el modelo tendrá que ser re-evaluado, re-entrenado, y posiblemente reconstruido completamente. Estas fases serán desarrolladas en los próximos capítulos.

[28] Relación cuantificada entre dos magnitudes que refleja su proporción.

2.16. Minería de Datos

Como ya se mencionó, la Minería de Datos, corresponde a la etapa más importante del proceso conocido como KDD y según [5], está conformada por "…*un conjunto de técnicas y algoritmos que sirven para hacer análisis de conjuntos de datos, extrayendo patrones y relaciones entre ellos, convirtiéndolos en información valiosa y útil para quienes toman las decisiones..".* Para Ian H. Witten [34], la MD es "…*el proceso de extraer Conocimiento útil y comprensible desde grandes cantidades de Datos almacenados en distintos formatos…*". Por tanto, el objetivo principal es encontrar modelos inteligibles a partir de estos datos. Se debe tener en claro que el proceso de MD genera muchos tipos de patrones, pero lo más importante es determinar qué patrones son útiles e interesantes, y cuáles no. Un patrón es interesante si cumple con ciertas condiciones, por ejemplo, si es fácil comprenderlo; si es válido con cierto grado de certeza, si para otro conjunto de datos, ya sea nuevo o de prueba; tiene una utilidad potencial, y si expresa un conocimiento novedoso y no trivial [35].

2.17. Tareas de la Minería de Datos

Según Hernández la principal meta del proceso de la minería de datos es el descubrimiento de reglas, las cuales mostrarán nuevas relaciones entre las variables o excepciones según el negocio que utilice este proceso [5]. Las tareas pueden clasificarse como predictivas o descriptivas.

2.17.1. Tareas descriptivas:

Orientadas a describir un conjunto de Datos.

- ✓ **La Clasificación:** La clasificación de datos es el proceso por medio del cual se encuentran propiedades comunes entre un conjunto de objetos de una base de datos. Se los cataloga en diferentes clases, de acuerdo al modelo de clasificación. Este proceso se realiza en dos pasos, en el primer lugar se construye un modelo donde cada *tupla*[29] o registro, de un conjunto de tuplas de la base de datos, tiene una clase conocida (etiqueta). Cada *tupla* de este conjunto se denomina ejemplo de entrenamiento. El segundo paso, es utilizar el modelo para clasificar. Inicialmente, se estima la exactitud del modelo utilizando otro conjunto de tuplas o registros, cuya

[29] En la teoría de Bases de datos, una tupla se define como una función finita que *mapea* (asocia unívocamente) los nombres con algunos valores. Un objeto de este tipo es conocido también como **registro** (o *record* en inglés).

clase es conocida, denominado "C*onjunto de Prueba"*. Este conjunto es escogido al azar, y es independiente del conjunto de entrenamiento.

- ✓ **La Agrupación o Clustering:** El proceso de agrupar objetos físicos o abstractos en clases de objetos similares se llama agrupación, segmentación o clustering o clasificación no supervisada. Básicamente, la técnica agrupa un conjunto de datos, sin un atributo de clase predefinido, basado en el principio de: maximizar la similitud intra-clase, y minimizar la similitud inter-clase. La meta de la segmentación, es la partición de ésta en grupos o clusters de registros similares que comparten un número de propiedades, y son considerados homogéneos. Se entiende por heterogeneidad a que los registros en diferentes segmentos no son similares, de acuerdo a una medida ya establecida. Algunos de los algoritmos más utilizados para clustering son el K-Means, BDScan, y el EM (Expectation Maximization).
- ✓ **La Estimación:** Es la actividad donde dados unos datos de entrada, se debe estimar valores para algunas variables continuas desconocidas, tales como "ingreso", y "balance de una tarjeta de crédito" entre otros. La estimación es similar a la clasificación, salvo que la variable de destino es numérica, y no categórica. Los modelos son construidos usando registros "completos", que proporcionan el valor de la variable de destino así como los predictores. Entonces, para las nuevas observaciones, las estimaciones del valor de la variable son realizadas en base a los valores de las variables predictoras.
- ✓ **La Asociación:** Este tipo de técnicas se emplea para establecer las posibles relaciones o correlaciones entre distintas acciones o sucesos aparentemente independientes, pudiendo reconocer como la ocurrencia de un suceso, o acción, puede inducir o generar la aparición de otros. Son utilizadas cuando el objetivo es realizar análisis exploratorios, buscando relaciones dentro del conjunto de datos. Las asociaciones identificadas pueden usarse para predecir comportamientos, y permiten descubrir correlaciones y co-ocurrencias de eventos. Por lo general esta forma de extracción de conocimiento se fundamenta en técnicas estadísticas, como los análisis de correlación y de variación.

2.17.2. Tareas de Predicción:

Es el proceso que intenta determinar los valores de una, o varias variables, a partir de un conjunto de datos. La predicción de valores continuos puede planificarse por las técnicas estadísticas de regresión. Por ejemplo, para predecir el sueldo de un graduado de la universidad con 10 años de experiencia de trabajo, o las ventas potenciales de un nuevo producto dado su precio. Se pueden resolver muchos problemas por medio de la regresión lineal, y puede conseguirse todavía más, aplicando las transformaciones a las

variables para que un problema no lineal pueda convertirse en uno lineal. Más adelante, dentro de la clasificación, se estudiarán varias técnicas de MD que pueden servir para predicción numérica. De entre todas ellas, las más importantes se presentaran en la clasificación bayesiana, la basada en ejemplares y las redes de neuronas.

2.18. Técnicas de Minería de Datos

La Minería de Datos ha dado lugar a una paulatina sustitución del análisis de datos dirigido a la verificación, por un enfoque dirigido al descubrimiento del conocimiento. La principal diferencia entre ambos se encuentra en que en el último se descubre información sin necesidad de formular previamente una hipótesis. La aplicación automatizada de algoritmos de MD permite detectar fácilmente patrones en los datos, razón por la cual esta técnica es mucho más eficiente que el análisis dirigido a la verificación cuando se intenta explorar datos procedentes de repositorios de gran tamaño y complejidad elevada. Dichas técnicas emergentes se encuentran en continua evolución como resultado de la colaboración entre campos de investigación tales como bases de datos, reconocimiento de patrones, inteligencia artificial, sistemas expertos, estadística, visualización, recuperación de información, y computación de altas prestaciones. Los algoritmos de MD se clasifican en dos grandes categorías: de *Aprendizajes Supervisados o predictivos,* y de *Aprendizajes no Supervisados* o de Descubrimiento del Conocimiento [35].

2.19. Tipos de Aprendizajes:

El concepto de aprendizaje tiene un amplio abanico de definiciones debido a que es un término muy general. Para diferenciarlo del procedimiento biológico de aprendizaje de los seres vivos, se denomina “Aprendizaje Automático” al campo cuyo objetivo es el estudio de programas que mejoran con la experiencia, aunque a partir de ahora, sin posibilidad de confusión, se hará referencia al Aprendizaje Automático como Aprendizaje. Este campo de estudio es multidisciplinar. Los conceptos que se manejan abarcan diversas disciplinas como la estadística, la inteligencia artificial, la filosofía, la teoría de la información, la biología, la ciencia cognitiva, etc.

Aunque todas estas disciplinas pretendan explicar la naturaleza del problema desde distintos puntos de vista, todas tienen en común que el objetivo perseguido del aprendizaje es el de incrementar el Conocimiento o las habilidades para cumplir una determinada tarea. A lo largo de la bibliografía se encuentran diversas definiciones del término. Una de las definiciones de referencia más utilizada es la de Mitchell en [36], "…*Un programa de ordenador se dice que aprende de la experiencia E respecto a un tipo de tarea T y a la medida de rendimiento P, si su rendimiento en la tarea T, medida mediante P, mejora con la experiencia E*…".

Los tipos de algoritmos utilizados en aprendizaje automático se suelen dividir principalmente en dos categorías a saber:

- ✓ **Aprendizaje Supervisado**: Como ya mencioné, es una técnica para deducir una función a partir de datos de entrenamiento. El objetivo del aprendizaje es el de crear una función capaz de predecir el valor correspondiente, a cualquier objeto de entrada válida, después de haber visto una serie de ejemplos, o sea los datos de entrenamiento. Para ello se tiene que generalizar, a partir de los datos presentados, a las situaciones no vistas previamente.
- ✓ **Aprendizaje No Supervisado**: es un método de aprendizaje automático donde un modelo es ajustado a las observaciones. Se distingue del anterior por el hecho de que no hay un conocimiento previo. Es la tarea de estos algoritmos encontrar estructuras a priori desconocidas, que se encuentran en un conjunto de datos. Se desconocen las estructuras intrínsecas del conjunto de datos debido a la ausencia de un atributo que de alguna manera guie (supervise) la formación de dichas estructuras.

Una técnica constituye el enfoque conceptual para extraer la información de los datos y, en general, es implementada por varios algoritmos. Cada algoritmo representa, en la práctica, la manera de desarrollar una determinada técnica paso a paso, de forma que es preciso un entendimiento de alto nivel de los algoritmos para saber cuál es la técnica más apropiada para cada problema. En la siguiente figura se puede ver las principales técnicas.

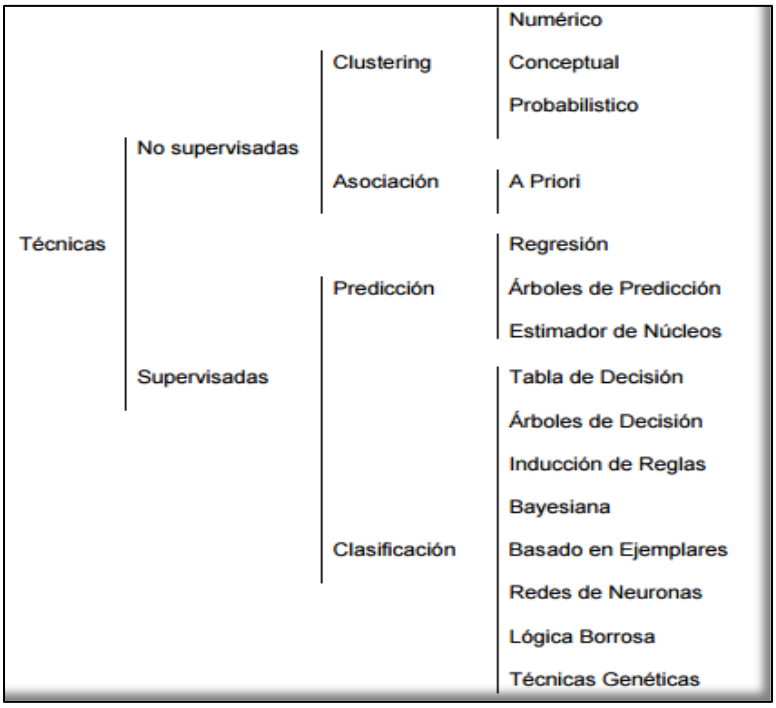


Figura 8: Técnicas de la Minería de Datos

2.20. Los algoritmos propuestos:

En este apartado se hará una introducción a los algoritmos que se utilizarán en el presente trabajo para la comparación.

Determinados algoritmos y métodos del aprendizaje computacional permiten obtener buenos resultados de predicción una vez entrenado el modelo con un número conocido de datos. De entre todos los modelos predictivos, las Redes Neuronales Artificiales (RNA) y las Máquinas de Vector Soporte (MVS) han constituido, en los últimos tiempos, un foco de investigación importante y con una actividad intensa, siendo un paradigma de aprendizaje computacional muy extendido en la resolución de problemas de diversas áreas de la Ingeniería y la Ciencia. La diferencia, quizás la principal, es que las RNA utilizan durante la fase de entrenamiento, el principio de Minimización del Riesgo Empírico (ERM), y las MVS se basan en el principio de Minimización del Riesgo Estructural (SRM), estas serán explicadas en los próximos apartados. Las MVS han mostrado un mejor desempeño que las RNA, ya que su técnica minimiza un límite superior al riesgo esperado, a diferencia del ERM que minimiza el error sobre los datos de entrenamiento [37].

2.21. Red Neuronal Artificial.

Es un sistema de aprendizaje y procesamiento automático inspirado en la forma como funciona el sistema nervioso animal, que parte de la interconexión de neuronas que colaboran entre sí para generar un resultado. En la Figura 9 se muestra la estructura de una neurona artificial.

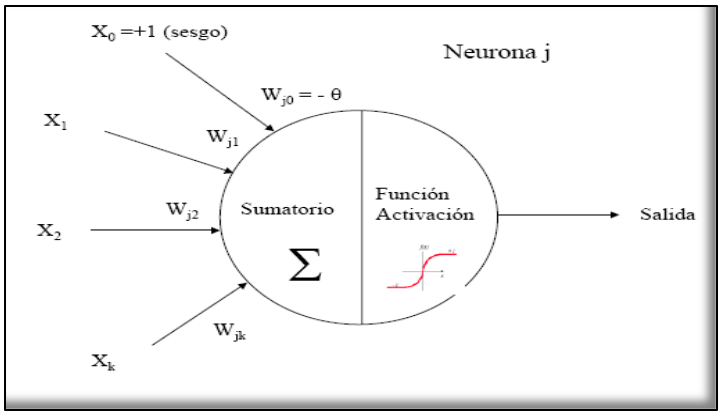


Figura 9: Estructura de la neurona artificial básica y modelado del umbral.

La salida o resultado de una neurona proviene de tres funciones [5].

1. **Propagación** (*función de excitación*): Es la sumatoria de cada entrada multiplicada por el peso de su interconexión (*valor neto*). Si el peso es positivo, la conexión se denomina *excitatoria*, y si es negativo, se denomina inhibitoria.
2. **Activación** (modifica a la función anterior): su existencia no es obligatoria, siendo en este caso la salida, la misma función de propagación.
3. **Transferencia**: se aplica al valor devuelto por la función de activación. Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que se desee dar a dichas salidas. Algunas de las más utilizadas son la función sigmoidea (para obtener valores en el intervalo [0,1]) y la tangente hiperbólica (para obtener valores en el intervalo [-1,1]).

2.22. Las Máquinas de Vectores de Soporte.

Las MVS o SVM, del inglés *Support Vector Machine*, buscan el límite que separa las clases con el mayor margen posible. Cuando no se pueden separar bien las dos clases, los algoritmos buscan el mejor límite que pueden. Las MVS hacen esto, sólo con una línea recta (usa un kernel lineal) y gracias a que hace esta aproximación lineal, se puede ejecutar con

bastante rapidez [37]. Figura 10. Además pueden separar las clases más rápidamente, y con menos sobreajuste, que con la mayoría de los otros algoritmos, además de que requieren solo una pequeña cantidad de memoria.

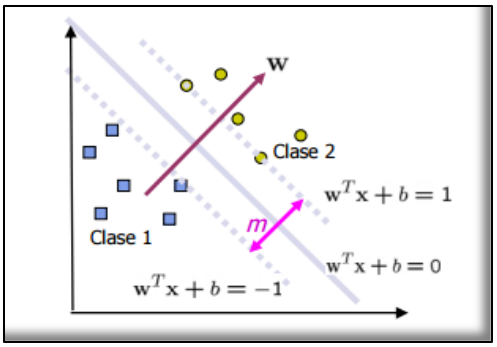


Figura 10: Gráfico del hiperplano equidistante para un caso bidimensional.

Cuando el conjunto de muestras etiquetadas no es linealmente separable, el objetivo será transformar mediante funciones kernel los vectores de entrada x_i de n-dimensiones en vectores de dimensión más alta donde las clases puedan ser linealmente separables. Ver Figura 11.

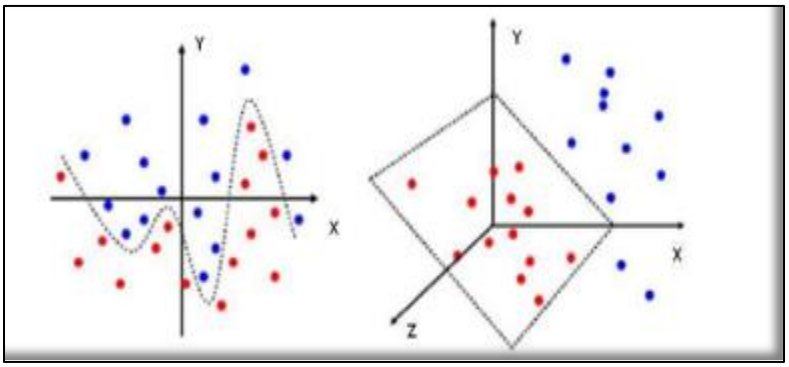


Figura 11: Transformación mediante la función kernel para separabilidad no lineal.

A continuación, un cuadro comparativo entre las RNA y MSV.

RNA	MSV
Las capas ocultas transforman a espacios de cualquier dimensión.	Kernels Transforma a espacios de dimensión muy alta.
El espacio de búsqueda tiene múltiples mínimos locales.	El espacio de búsqueda tiene un mínimo global.

El entrenamiento es costoso. Se establece el número de nodos y capas ocultas Alto funcionamiento en problemas típicos	El entrenamiento es altamente eficiente. Se diseña la función de kernel y el parámetro de coste C. Muy buen funcionamiento en problemas típicos Extremadamente robusto para generalización. Menos necesidad de emplear heurísticos en el entrenamiento.

Tabla 2: Cuadro comparativo entre las RNA y MSV

2.23. Herramienta de Minería de Datos

Existen diversas herramientas de software para realizar los procesos de MD. Algunas de ellas de muy buen rendimiento; otras que además cuentan con interfaces gráficas amigables y que, en general, ofrecen diversos algoritmos para trabajar, incluso permitiendo realizar pre-procesamiento de los datos.

Hernández en su libro [5], describe una gran cantidad de herramienta informática para llevar procesos de MD, para llevar adelante este trabajo se seleccionó la suite de WEKA[30]. Esta suite es desarrollada por la Universidad de Waikato (Nueva Zelanda). Está escrita en lenguaje Java para varias plataformas [34]. En el manual de Weka [38], se encuentra una explicación más detallada de las cuatro opciones que posee el producto, como se ve en su primera pantalla o *Weka GUI Chooser* . Figura 12:

- **Explorer**: El modo Explorador es el modo más usado y más descriptivo. Éste permite realizar operaciones sobre un sólo archivo de datos.
- **Experimenter**: El modo experimentador, es un modo muy útil para aplicar uno o varios métodos de clasificación sobre un gran conjunto de datos y, luego poder realizar contrastes estadísticos entre ellos y obtener otros índices estadísticos.
- **Knowledge flow**: Es similar a Explorer pero permite construir un *Workflow (flujo de trabajo)* del proceso de Minería de Datos.

[30] Waikato Environment for Knowledge Analisys. *http://www.cs.waikato.ac.nz/~ml/weka/index.html*

- **Simple CLI**: Es una abreviación de *Simple Client*. Esta interfaz proporciona una consola para poder introducir mandatos. A pesar de ser en apariencia muy simple es extremadamente potente porque permite realizar cualquier operación soportada por Weka de forma directa; no obstante, es muy complicada de manejar ya que es necesario un Conocimiento completo de la aplicación.

Figura 12: Ventana principal de Weka

Weka incluye toda una serie de funcionalidades, algunas de las cuales son:

- Implementación de amplia variedad de algoritmos de *machine learning*, entre los cuales está disponible el algoritmo clasificatorio propuestos, con la posibilidad de que varios parámetros pueden ser modificables por el usuario.
- Inclusión de una amplia variedad de herramientas de pre-proceso de los conjuntos de datos, (denominadas filtros en la documentación del programa). Podría llegar a ser posible pre-procesar los Datos, construir un modelo con ellos y analizar la eficiencia del modelo sin necesidad de crear ninguna aplicación adicional de ayuda.
- Existe disponible una amplia documentación en línea, que va siendo actualizada periódicamente a medida que se introducen cambios y se amplía la aplicación. Entre la documentación disponible existe documentación API generada mediante Javadoc[31], la cual describe exhaustivamente las clases Java que se usan en el proyecto. Esto permite que el usuario avanzado pueda llegar a desarrollar, e implementar, en la aplicación sus propios algoritmos, y filtros de pre-procesado de datos. Así, por ejemplo, para implementar un algoritmo clasificador puede usarse las clases del paquete java *Classifiers*.

[31] Javadoc es una utilidad de Oracle para la generación de documentación de APIs en formato HTML a partir de código fuente Java. Javadoc es el estándar de la industria para documentar clases de Java.

La clase diferida *Classifier* define la estructura que debe tener cualquier esquema clasificatorio que se quiera implementar como subclase. Para ello debe implementarse el método abstracto buildClassifier() en la subclase.

- La aplicación puede usarse tanto desde la línea de comandos del sistema operativo como desde las interfaces gráficas que proporciona.
- Dispone de herramientas de visualización y análisis gráfico de los Modelos construidos, lo cual facilita la toma de decisiones respecto la validez de los mismos.

Esta tesis se centrará particularmente en la aplicación *Explorer*. En el capítulo III, se explica cada uno de los procesos que se usaron para este trabajo.

2.24. Trabajos similares:

A continuación se enumeran algunos de los trabajos sobre MD y deserción.

- **Detección de Patrones de Deserción Estudiantil en Programas de Pregrado de Instituciones de Educación Superior con CRISP-DM.,** Timaran, R; Jimenez, J. Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación. ISBN: 978-84-7666-210-6 – Artículo 758. (Colombia. 2014)
- **La Minería de Datos como un Método Innovador para la detección de Patrones de Deserción Estudiantil en Programas de Pregrado en Instituciones de Educación Superior.** Ricardo Timarán Pereira, Andrés Calderón Romero. Javier Jiménez Toledo. International Federation of Engineering Education Societies (IFEES). (Colombia. 2013)
- **Análisis de la deserción estudiantil en la Universidad Simón Bolívar, facultad Ingeniería de Sistemas, con técnicas de Minería de Datos.** Kamagate Azoumana, Pensamiento Americano, Universidad Simón Bolívar Barranquilla. (Colombia. 2013).
- **Aplicación de técnicas de aprendizaje automático y Minería de Datos al problema de la deserción de alumnos.** Mg. Ing. Beatriz Parra de Gallo, Lic. Alejandra Carolina Cardoso. Universidad Católica de Salta. Facultad de Ingeniería e Informática. (Argentina. 2006).

- **La Deserción Escolar desde el punto de vista de la Minería de Datos. Caso de Estudio: Tesjo, Generaciones 2003—2008.** R. Alejo y otros. Congreso Internacional de Investigación Tijuana. Revista Aristas: Investigación Básica y Aplicada. ISSN 2007-9478, Vol. 4, Núm. 7. Año 2015. 18 al 20 de febrero 2015. Facultad de Ciencias Químicas e Ingeniería. UABC. Tijuana, Baja California, (México. 2015)
- **Predicción del Fracaso Escolar mediante Técnicas de Minería de Datos.** Carlos Márquez Vera, Cristóbal Romero Morales y Sebastián Ventura Soto. IEEE-RITA Vol. 7, Núm. 3, (España. 2012)
- **Construcción de Modelos matemáticos para determinar el nivel de deserción en los programas de pregrado de la universidad mariana.** Ivan Mauricio Argote Puetaman y otros. Universidad Mariana. (Colombia. 2013).
- **Descubrimiento de Conocimiento en la Base de Datos académica de una institución de educación superior usando redes neuronales**. Javier Hernández Cáceresa. Facultad de Ingeniería Industrial. Universidad Santo Tomás, Bucaramanga, (Colombia. 2013).
- **Impacto de Actividades Cotidianas en el Rendimiento Estudiantil**. Huerta Luis. International Journal of Innovation and Applied Studies. ISSN 2028-9324 Vol. 14 No. 4 Feb. 2016, pp. 927-935. (México. 2016).
- **Modelo de Minería de Datos para identificación de Patrones que influyen en el aprovechamiento Académico**. Jaime Ángel Hernández Cedano. Instituto Tecnológico de La Paz. (México. 2015).
- **Análisis de Deserción-Permanencia de Estudiantes Universitarios Utilizando Técnica de Clasificación en Minería de Datos.** Karina B. Eckert y Roberto Suénaga. Universidad Gastón Dachary, Departamento de Ingeniería y Ciencias de la Producción, (Argentina. 2015).
- **Análisis de deserción escolar con Minería de Datos.** José Luis Aguirre Mendiola y otros. Tecnológico de Estudios Superiores de Jocotitlán, Estado de México, (México. 2015).
- **Caracterización de la deserción estudiantil en educación superior con Minería de Datos.** Javier Alejandro Jiménez Toledo, Silvio Ricardo Timarán Pereira. Revista Tecnológica ESPOL – RTE, Vol. 28, N. 5, 447-463, (Colombia. 2015)

- **Detección de patrones de deserción en los programas de pregrado de la universidad mariana de san juan de pasto, aplicando el proceso de Descubrimiento de Conocimiento sobre Base de Datos (kdd) y su implementación en Modelos matemáticos de predicción.** Argote, Ivan, Jiménez, Robinson, Gómez, Jair. Universidad Mariana – (Colombia, 2014).
- **Implementación de Modelos de Minería de Datos Para la definición de tendencias de deserción y permanencia En la universidad nacional de Colombia**. López Guarín, Camilo Ernesto, Gallego Vega, Luis Eduardo, Casadiego, María Angélica. Universidad Nacional de Colombia – (Colombia. 2015).
- **Descubrimiento de Conocimientos en la Base de Datos académica de la Universidad Autónoma de Manizales aplicando redes neuronales.** Jairo Elias Gutierrez. Universidad Autónoma de Manizales. Manizales. (Colombia. 2012).
- **Estudio de variables que influyen en la deserción de estudiantes universitarios de primer año, mediante Minería de Datos.** Christian Zarria Torres, Christian Arce Ramos, Jaime Lam Moraga. Universidad Científica del Perú. (Perú. 2016).
- **Modelo de detección de estudiantes excluidos en carreras de ingeniería utilizando Minería de Datos.** Alveiro Alonso Rosado Gómez. Facultad de Ingenierías. Universidad Francisco de Paula Santander Ocaña. (Colombia. 2013).
- **Proyección de Estudiantes en Riesgo de Desertar Mediante Técnicas de Minería de Datos.** Jhon Jaime Méndez Alandete. Corporación Universitaria del Caribe – CECAR. (Colombia. 2015).
- **Predicción del fracaso y el abandono Escolar mediante técnicas de Minería de Datos.** Carlos Márquez Vera. Universidad de Córdoba. (Argentina. 2015).
- **Uso de Tecnologías de la Información para Detectar Posibles Deserciones Universitarias.** Marisa Fabiana Haderne. Universidad Nacional de Cuyo Universidad del Aconcagua. Mendoza, (Argentina. 2004)
- **Un modelo basado en Árboles de decisión para predecir la deserción estudiantil en la Educación Superior Privada**. Daza Vergaray, Alfredo. Facultad de Ingeniería de Sistemas. Universidad César Vallejo-Lima. (Perú. 2016).

- **Modelo predictivo de deserción estudiantil utilizando técnicas de Minería de Datos.** Yegny Karina Amaya Torradoa y otros Universidad Simón Bolívar, Barranquilla, (Colombia.2012).
- **Identificación de Factores en la Deserción y Reprobación Universitaria.** Ordonez Ordonez, Pablo & González González, Aníbal. Universidad de Alcalá, ISBN: 978-84-16133-42-0. (Ecuador. 2010)
- **Hacia un modelo de deserción universitaria: Análisis exploratorio de variables socioeconómicas y académicas de alumnos de primer y segundo año de la carrera Ingeniería en Sistemas de Información,** UTN Facultad Regional Rosario". Ing. Juan Miguel Moine, Ing. Luciano Valia, Ing. José Rostagno, Ing. Cristian Bigatti y otros (UTN FRRo). Trabajo presentado en el IV Congreso Nacional de Ingeniería en Informática (CoNaIISI), Simposio de Base de Datos. (Salta, Noviembre de 2016)
- **Abordaje del fenómeno de deserción universitaria con técnicas de Minería de Datos.** Ing. Juan Miguel Moine, Ing. José Rostagno, Ing. Cristian Bigatti (UTN FRRo).Revista Rumbos Tecnológicos, Vol 8, año 2016. ISSN 1852-7698.
- **Una herramienta para la evaluación y comparación de metodologías de Minería de Datos.** Ing. Juan Miguel Moine, Dra. Ana Haedo (UBA) y Dra. Silvia Gordillo (UNLP). Trabajo presentado en el "Workshop de Bases de Datos y Minería de Datos", XXI Congreso Argentino de Ciencias de la Computación 2015 (CACIC). (Junín, Octubre de 2015)

"La información es la gasolina del siglo XXI,
y la analítica de datos el motor de combustión".
Peter Sondergaard.

Capítulo 3.

3. Modelos de Clasificación de Minería de Datos

Últimamente las computadoras han pasado de resolver problemas matemáticos, a ayudar en casi todos los aspectos de la vida diaria. Aun así, hay problemas complejos en los que las computadoras no podrían ayudar si sólo se usaran los métodos tradicionales de programación. Uno de estos problemas es el reconocimiento de patrones[32], y una forma de solucionarlo es usando técnicas de minería de datos. Dado que existe una gran cantidad de métodos para el análisis de la información almacenada, es importante realizar un estudio comparativo entre las técnicas de *Aprendizaje Automatizado* más utilizadas históricamente, para el diagnóstico en la deserción universitaria, y otras técnicas de reciente surgimiento, con excelente desempeño en el reconocimiento de patrones. El objetivo es determinar la técnica con mayor capacidad en clasificar de forma correcta patrones, para el diagnóstico en la deserción universitaria, mediante el rendimiento académico de los estudiantes, a partir del análisis de los datos históricos provenientes de los mismos.

3.1. Aprendizaje Automatizado (AA).

Es la disciplina que estudia cómo construir sistemas computacionales que progresen automáticamente mediante la experiencia. De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de información, no estructurada, suministrada en forma de ejemplos. O sea, crear programas que puedan, en un sentido similar a lo realizado por los humanos, aprender por sí mismos. El aprendizaje automatizado abarca una gran variedad de técnicas para buscar y descubrir patrones estructurales. Las técnicas que se pretenden comparar en este trabajo pertenecen al paradigma de aprendizaje supervisado, es decir que necesitan una etapa de aprendizaje para construir un modelo, utilizando los datos de entrenamiento, para después usarlo en la predicción o inferencia de la categoría de ejemplos desconocidos.

[32] El **reconocimiento de patrones** es la ciencia que se ocupa de los procesos sobre ingeniería, computación y matemáticas relacionados con objetos físicos o abstractos, con el propósito de extraer información que permita establecer propiedades de entre conjuntos de dichos objetos.

3.2. Modelos de Clasificación basados en Redes Neuronales Artificiales.

Las Redes Neuronales Artificiales constituyen un modelo computacional inspirado en ciertas características de las redes neuronales biológicas, y permiten resolver diversos problemas de la vida real. Se agrupan dentro de las teorías de aprendizaje conexionista, y constituyen una de las especialidades más ampliamente difundidas. Estas herramientas matemáticas para la modelación de problemas, permiten obtener las relaciones funcionales subyacentes entre los datos involucrados en problemas de clasificación, reconocimiento de patrones, regresiones, etc. Son consideradas excelentes aproximadores de funciones esencialmente no lineales, siendo capaces de aprender las características relevantes de un conjunto de datos, para luego reproducirlas en entornos ruidosos o incompletos [39].

3.2.1. El Cerebro Humano

El cerebro, como todo en el organismo animal, está formado por células, pero las del cerebro son excepcionales por su impresionante diversidad, por la complejidad de sus formas, por la intrincadísima red que comunica a unas células con otras. Está compuesto por dos hemisferios y el cuerpo calloso que los une. Aunque no lo parezca, el cerebro humano tiene una superficie aproximada de $2m^2$, pero cabe en el cráneo debido a que está plegado de una forma muy peculiar. Por su función preponderante, es el único órgano completamente protegido por una bóveda ósea llamada "*cavidad craneal*".

Las células del cerebro se llaman *Neuronas*. La estructura y la comunicación de las neuronas, fueron descritas magistralmente en los albores de este siglo, por el sabio español Santiago Ramón y Cajal[33] [40]. A pesar de las diferencias en la forma de las neuronas, su estructura en los sitios en los que se comunican unas con otras es muy similar. La parte de la neurona que "*habla*" con otra neurona tiene siempre una estructura típica, y la región de la neurona que recibe ese contacto también tiene una forma característica. A esta zona de interacción de las neuronas se le llama "*sinapsis"* (del griego σύναψις = unión, enlace), y su

[33] Médico español, especializado en histología y anatomía patológica.
https://es.wikipedia.org/wiki/Santiago_RamCB3n_y_Cajal

funcionamiento es esencial para explicar prácticamente todas las acciones del cerebro, desde las más sencillas como ordenar a los músculos que se contraigan y se relajen en forma coordinada para llevar a cabo un simple movimiento, hasta las más complicadas tareas intelectuales, pasando también por las funciones que originan, controlan y modulan las emociones [41].

El término neurona procede del vocablo griego *neyron* (nervio), es una célula del sistema nervioso especializada en captar los estímulos provenientes del ambiente y, de transportar y transmitir impulsos nerviosos (mensajes eléctricos). La neurona está considerada como la unidad nerviosa básica, tanto funcional como estructural del sistema nervioso. La neurona no se divide, y su tasa de reproducción es extremadamente baja. El tamaño y forma de las mismas es muy variable, pero todas cumplen con la función de conducir impulsos nerviosos. Una neurona está constituida por un cuerpo celular o soma, que es su parte más ancha .y contiene un núcleo rodeado por citoplasma. Están también sus prolongaciones, o fibras conocidas como dendritas y axón. Las primeras son ramificaciones cortas y numerosas, que conducen el impulso hacia el cuerpo celular; y la segunda, es una ramificación larga que transmite dicho impulso desde el cuerpo celular, hasta la neurona próxima [41]. En la Figura 13 se puede observar un grupo de neuronas y sus sinapsis.

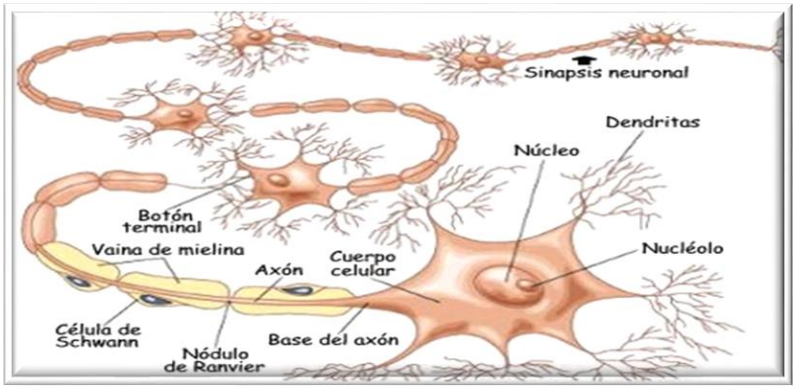


Figura 13. Neurona biológica. [40]

Como se mencionó, la conexión entre dos neuronas se denomina *sinapsis,* y se origina entre el botón terminal de un *axón,* y las *dendritas* iniciales de otra neurona. Como se sabe su función básica es la transmitir mensajes en impulsos nerviosos, a través de un proceso que puede ser de tipo de eléctrico cuando un impulso viaja a lo largo de una fibra nerviosa, o

de tipo químico cuando la señal es transmitida desde una neurona a otra, en ambos procesos sustancias denominadas neurotransmisores.

3.2.2. Sustancia blanca y Sustancia gris

La sustancia blanca es una parte del sistema nervioso central compuesta de fibras nerviosas mielinizadas, o sea recubiertas de mielina, sustancia que permite transmitir más rápidamente el impulso nervioso. Las fibras nerviosas contienen sobre todo axones, parte de la neurona encargada de la transmisión de información a otra célula nerviosa. La llamada "*sustancia gris*", en cambio, está compuesta por las dendritas y cuerpos neuronales. [42]. El lector interesado en profundizar en este tema, puede encontrar en internet dos referencias interesantes al respecto [43] y [45].

3.2.3. La Neurona Artificial

La neurona artificial o sintética, Figura 14, que es un modelo simplificado de la neurona biológica, fue propuesta por primera vez en el año 1943 por McCulloch[34] y Pitts[35] [40], en un trabajo publicado bajo el título: "*A logical calculus of the ideas immanent in nervous activity*"[36].

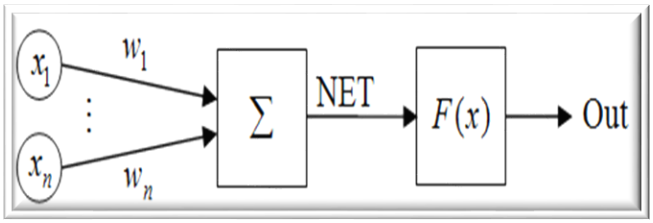


Figura 14. Modelo de neurona de McCulloch y Pitts.

En este Modelo, cada neurona consta de un conjunto de entradas, X_n, y una sola salida (***Out***). Cada entrada X_i está afectada por un coeficiente que se denomina ***peso*** y que se representa por w_i. El subíndice ***i*** refleja que el peso afecta a la entrada ***i.*** En el caso de que la entrada provenga de más de una neurona, el peso se representaría w_{ij}, donde el subíndice ***i*** refleja que el peso afecta a la entrada ***i***, y el subíndice ***j*** que se trata de la neurona ***j***.

[34] https://es.wikipedia.org/wiki/Warren_McCulloch
[35] https://es.wikipedia.org/wiki/Walter_Pitts
[36] http://www.cs.cmu.edu/~epxing/Class/10715/reading/McCulloch.and.Pitts.pdf

La cantidad calculada como la suma del producto de cada entrada multiplicada por su respectivo peso se denomina función de transferencia de la neurona x_i y devuelve la entrada ***Net.*** Las redes neuronales tienen muchas propiedades y la capacidad de aprender es la más destacada. El proceso de aprendizaje se reduce a cambiar los pesos w_i.

$$\text{Net} = \sum_{n} x_n w_n \qquad \textbf{(1)}$$

Net es la entrada neta de la neurona.

La entrada neta se transforma luego en la salida mediante la función de activación. En pocas palabras, puede considerarse una red neuronal como una "*caja negra*" que recibe señales como *entradas* y proporciona resultados como *salidas*. La fórmula se representa de la siguiente manera:

$$Salida(Out) = F(Net - \Theta \qquad \textbf{(2)}$$

Donde:

- ***F(x)*** es la función de activación;
- ***Net*** es la suma ponderada obtenida en la primera etapa del cálculo de la información de salida de una neurona;
- Θ es un valor de umbral de la función de activación. Solo se usa para la función de umbral y es igual a cero en otras funciones.

3.2.4. Funciones de Activación.

Existen varias funciones de activación que calculan la información de salida de una neurona, recuadro en rojo en el Figura 15. La entrada que recibe, representa la suma de todos los productos de las entradas, y sus respectivos pesos, en adelante "*suma ponderada*":

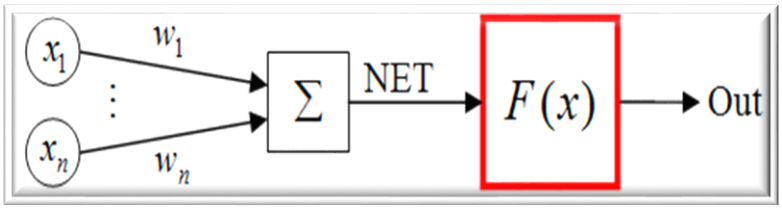
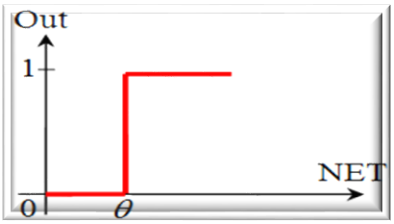


Figura 15. Función de activación.

- **Función Escalón**: se utiliza cuando las salidas de la red son binarias. La salida de una neurona se activa sólo cuando el estado de activación es mayor o igual que cierto valor

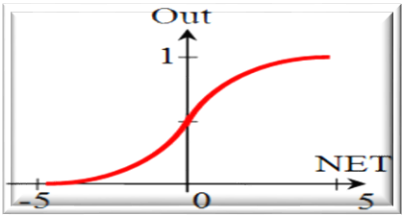
umbral *t* que representa la mínima entrada total ponderada necesaria para provocar la activación de la neurona. Figura 16.



$$\mathbf{output}_i = \begin{cases} \mathbf{0} & \mathbf{si\ Net} \leq \mathbf{T} \\ \mathbf{1} & \mathbf{si\ Net} > \mathbf{T} \end{cases}$$

Figura 16. Función Escalón

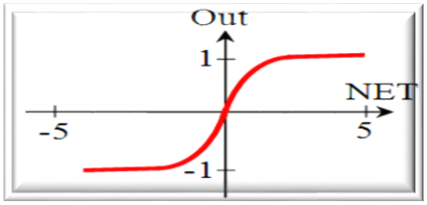
- **Función Sigmoidal:** es la más apropiada cuando se quiere como salida información analógica. Con esta función, para la mayoría de los valores del estímulo de entrada (variable independiente), el valor dado por la función es cercano a uno de los valores asintóticos. La importancia de esta función es que su derivada es siempre positiva y cercana a cero para los valores grandes positivos o negativos; además toma su valor máximo cuando x es 0. Esto hace que se puedan utilizar las reglas de aprendizaje en las cuales se usan derivadas. La expresión de esta función responde a la forma.



$$output = \frac{1}{1 + e^{-Net}}$$

Figura 17. Función Sigmoidal

- **Función Hiperbólica**. Esta es una de las funciones más utilizadas en las RNA por su flexibilidad y el amplio rango de resultados que ofrece. Las ventajas de utilizar una tangente sigmoidea frente a una sigmoidea reside en que la segunda solo ofrece resultados en el rango positivo entre 0 y 1. En cambio, la tangente sigmoidea da resultados entre 1 y -1, por lo que se amplía a los números negativos los posibles resultados.



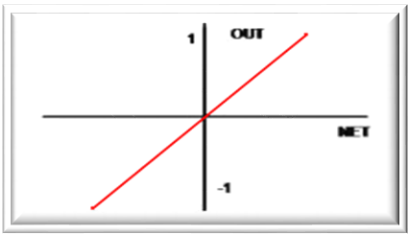
$$output = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}}$$

Figura 18. Función hiperbólica

3.2.5. Tipos de neuronas

La linealidad de las funciones que definen a los elementos de la red es quizás lo que va a proporcionar la característica más definitoria. Así, se pueden clasificar las neuronas en *lineales y no lineales*.

- **Neuronas lineales.** Una neurona es lineal cuando su salida es linealmente dependiente de sus entradas, es decir, proporcional a las funciones de transferencia y de activación. Figura 19.



$$output = x = y$$

Figura 19. Función de Transferencia Lineal

- **Neuronas no lineales**: En estas neuronas, o bien la función de activación, o bien la función de transferencia (o ambas) son funciones no lineales, dando lugar a que la respuesta de la neurona no sea función lineal de sus entradas. Este tipo de neuronas va a producir respuestas acotadas, desapareciendo los problemas de fluctuación y la falta de adecuación a señales pequeñas y grandes. Como ejemplo de funciones no lineales se pueden destacar las funciones vistas anteriormente: Escalón, Sigmoide e Hiperbólica Figuras 17,18 y 19 respectivamente [41].

3.2.6. ¿Qué es una Red Neuronal Artificial?

Podría decirse que *"…una red neuronal artificial es un conjunto de algoritmos matemáticos que procesan información y encuentran relaciones no lineales entre el conjunto de datos, y cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la neurona…"* [43], es decir, la teoría y el modelo de las RNA están inspirados en la estructura, el funcionamiento del sistema nervioso y en particular, de la neurona biológica. En este modelo tan sencillo puede observarse que la activación de la neurona depende del valor que tomen los pesos y las entradas, de forma tal que la variación de éstos, originan distintas salidas para la misma entrada a la neurona. En la práctica, los pesos de las neuronas se modifican sometiendo a la red a un entrenamiento, permitiendo que la red realice una función determinada. Esta es la característica que diferencia a una red neuronal de una máquina algorítmica clásica, una red neuronal no se programa, se «educa». La red es capaz de retener y asociar el conocimiento a través de la adaptación de los pesos de las neuronas siguiendo una regla de aprendizaje. Estas reglas son ecuaciones expresadas en función de las entradas y salidas de las neuronas, y describen la forma de variación de los pesos. En definitiva, son el instrumento empleado por las neuronas para adaptarse a la información que se le presenta

3.2.7. Breve reseña histórica.

- **1943**. primer paso dado por el neurofisiólogo Warren McCulloch y el matemático Walter Pitts quienes escribieron un documento en el cual explicaban el posible funcionamiento de las neuronas e hicieron un Modelo simple de una red neuronal con circuitos eléctricos.
- **1949**. Donald Hebb apoya el concepto *de Neurona y su funcionamiento,* escribiendo un libro titulado "*The Organization of Behavior***"**[37] en el cual comenta la actividad existente en las neuronas cada vez que son usadas. Donald Hebb presentaba su conocida "*regla de aprendizaje***"**.
- **1957**. Frank Rosenblatt. Comenzó el desarrollo del "Perceptrón". Esta es la red neuronal más antigua; utilizándose hoy en día para aplicación como identificador de patrones. Este modelo era capaz de generalizar, es decir, después de haber aprendido una serie de patrones podía reconocer otros similares, aunque no se le hubiesen presentado en el

[37] http://s-f-walker.org.uk/pubsebooks/pdfs/The_Organization_of_Behavior-Donald_O._Hebb.pdf

entrenamiento. Sin embargo, tenía una serie de limitaciones, por ejemplo, su incapacidad para resolver el problema de la función OR-exclusiva y, en general, era incapaz de clasificar clases no separables linealmente.

- **1959**. Frank Rosenblatt: Principios de Neurodinámica. En este libro confirmó que, bajo ciertas condiciones, el aprendizaje del perceptrón convergía hacia un estado finito (Teorema de convergencia del perceptrón).
- **1960 -** Bernard Widrow y Marcian Hoff de la Universidad de Stanford. Desarrollaron Teoría sobre la adaptación neuronal. Crearon un Modelo llamado "*Adaline*" (Adaptative Linear Neuron) y luego la "*Madaline*" (Múltiple Adealine). Esta fue la primera aplicación de las Redes a problemas reales: filtros adaptativos para eliminar ecos en las líneas telefónicas.
- **1969**. En este año casi se produjo la "muerte abrupta" de las redes neuronales; pues Minsky y Papert probaron (matemáticamente) en el libro ***Perceptrons,*** no era capaz de resolver problemas relativamente fáciles, tales como el Aprendizaje de una función no-lineal. Esto demostró que el perceptrón era muy débil, dado que las funciones no-lineales son extensamente empleadas en computación y en los problemas del mundo real.
- **1974**. Paul Werbos desarrolló la idea básica del algoritmo de aprendizaje de propagación hacia atrás (backpropagation); cuyo significado quedó definitivamente aclarado en 1985. Posteriormente se desarrollaron otros tipos de Redes: Kohonen en los 70 creó los mapas topológicos y las memorias asociativas. En 1982 Hopfield definió las *Redes de Hopfield*.
- **1986**. Rumelhart y McClelland desarrollaron el *perceptrón multicapa*, popularizándose así el algoritmo de *retropropagació*n. En 1989, Cybenko, Hornik et al. y Funahashi definieron el perceptrón multicapa como el *Aproximador Universal* [40].

3.2.8. ¿Qué son capaces de hacer la RNA?

En esencia las Redes Neuronales son capaces de realizar dos tareas diferentes: el reconocimiento de patrones y la síntesis funcional. Aunque parecen tener la misma capacidad de cómputo que una máquina de Turing[38], no se debe esperar que una red neuronal realice tareas que ya tienen una solución algorítmica buena, por ejemplo invertir una matriz.

[38] Una máquina de Turing es un dispositivo que manipula símbolos sobre una tira de cinta de acuerdo a una tabla de reglas. A pesar de su simplicidad, una máquina de Turing puede ser adaptada para simular la lógica de cualquier

- **Reconocimiento de patrones**: El reconocimiento de patrones, implica la clasificación de información según ciertas características. Aplicaciones típicas son la diferenciación de sonidos muy similares, el reconocimiento de escritura manuscrita, interpretación de encefalogramas, reconocimiento de la voz y procesamiento de imágenes.
- **Síntesis Funcional**: La aproximación de funciones, consiste en establecer relaciones entre varias entradas continuas (discretas) y una o más salidas continuas (discretas), por ejemplo, estimar la demanda de un producto, filtrar el ruido de una señal, controlar un proceso y simular el comportamiento de un sistema dinámico.

3.2.9. Topología de las Redes Neuronales.

La topología de una Red neuronal consiste en la organización y disposición de las neuronas en la misma, formando capas o agrupaciones de neuronas más o menos alejadas de la entrada y salida de dicha red. En este sentido, los parámetros fundamentales de la red son el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas. Como ya se mencionó, en una red neuronal artificial los nodos se conectan por medio de la sinapsis, estando el comportamiento de la red determinado por la estructura de conexiones sinápticas. Estas conexiones son direccionales, es decir, la información solamente puede propagarse en un único sentido, desde la neurona pre-sináptica a la pos-sináptica. En general las neuronas se suelen agrupar en unidades estructurales que se denominarán Capas. El conjunto de una o más capas constituye una red neuronal. Se distinguen tres tipos de capas, de entrada, de salida y ocultas.

Una capa de Entrada, también denominada Sensorial, está compuesta por neuronas que reciben datos o señales procedentes del entorno. Una capa de salida se compone de neuronas que proporcionan la respuesta de la red neuronal. Una capa oculta no tiene una conexión directa con el entorno, es decir, no se conecta directamente ni a órganos sensores ni a efectores. Este tipo de capa oculta proporciona grados de libertad a la red neuronal gracias a los cuales es capaz de representar más fehacientemente determinadas características del entorno que trata de modelar.

algoritmo de computador y es particularmente útil en la explicación de las funciones de una CPU dentro de un computador.

Teniendo en cuenta diversos conceptos se pueden establecer diferentes tipos de arquitecturas neuronales. Así, considerando su estructura, se puede hablar de Redes Monocapa o Multicapa. Teniendo en cuenta el flujo de datos, se puede distinguir entre redes unidireccionales (*feedforward*) y redes recurrentes o retroalimentadas (*feedback*), Figura 20. Mientras que en las redes unidireccionales la información circula en un único sentido, en las redes recurrentes la información puede circular entre las distintas capas de neuronas en cualquier sentido, incluso en el de salida-entrada [40].

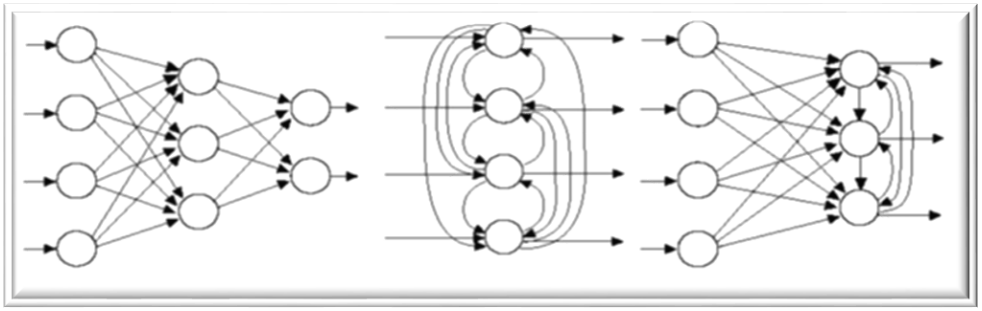


Figura 20. Conexiones hacia adelante (feed forward), laterales y hacia atrás (back propagation)

3.2.9.1. Redes Monocapa.

En las redes monocapa se establecen conexiones entre las neuronas que pertenecen a la única capa que constituye la red. Se utilizan generalmente en tareas relacionadas con lo que se conoce como auto-asociación. Regenerar información de entrada que se presenta a la red de forma incompleta, o distorsionada.

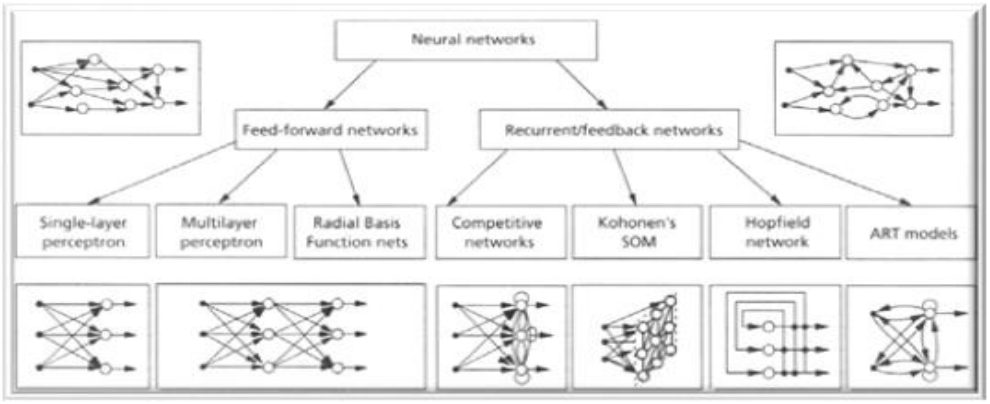
3.2.9.2. Redes Multicapa.

Las redes multicapas son aquellas que disponen de un conjunto de neuronas agrupadas en varios niveles, o capas (2, 3, etc.). En estos casos, una forma para distinguir la capa a la que pertenece una neurona, consistiría en fijarse en el origen de las señales que recibe a la entrada y el destino de la señal de salida. Normalmente, todas las neuronas de una capa reciben señales de entrada desde otra capa anterior, la cual está más cerca a la entrada de la red, y envían señales de salida a una capa posterior, que está más cerca a la salida de la red. Como se mencionó anteriormente, a estas conexiones se las denomina conexiones hacia adelante o feedforward. Sin embargo, en un gran número de estas redes también existe la

posibilidad de conectar la salida de las neuronas de capas posteriores a la entrada de capas anteriores; a estas conexiones se las denomina conexiones hacia atrás o feedback. La Figura 21 muestra algunos de los distintos tipos de arquitecturas posibles.

Figura 21. Arquitectura de RNA

3.2.10. Mecanismos de aprendizajes.



Una de las características más importantes de las Redes Neuronales, es su capacidad de aprender interactuando con su entorno, o con alguna fuente de información. Se ha visto que los datos de entrada se procesan a través de la red neuronal con el propósito de lograr una salida. También se dijo que estas redes extraen generalizaciones[39], desde un conjunto determinado de ejemplos anteriores de tales problemas de decisión. Una red neuronal debe aprender a calcular la salida correcta para cada constelación, arreglo o vector de entrada en el conjunto de ejemplos. Este proceso de aprendizaje se denomina: Proceso de Entrenamiento o Aprendizaje. El conjunto de datos, o conjunto de ejemplos, sobre el cual este proceso se basa es, por ende, llamado: Conjunto de Datos de Entrenamiento [5].

La topología de la red y las diferentes funciones de cada neurona (entrada, activación y salida) no pueden cambiar durante el aprendizaje, mientras que los pesos sobre cada una de las conexiones si pueden hacerlo; el aprendizaje de una red neuronal significa: Adaptación de los Pesos. En otras palabras el aprendizaje de la red es un proceso adaptativo, mediante el cual se van modificando los pesos sinápticos de la red, para mejorar el comportamiento de la misma. De manera general, una red neuronal va a modificar su peso sináptico wij

[39] Capacidad para aprender con cualquier conjunto de ensayo.

correspondiente a la conexión de la neurona i con la neurona j, mediante una regla de aprendizaje de la forma:

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}(k) \quad \textbf{(3)}$$

Es decir, el nuevo valor de los pesos sinápticos, se obtiene sumándole una cantidad (modificación) al valor antiguo. Este procedimiento básico sienta las bases de entrenamiento para los Sistemas basados en la neurona como unidad básica [43]. Se puede distinguir tres paradigmas de aprendizaje:

- **Aprendizaje supervisado**, en el que se va a disponer de un conjunto de patrones de entrenamiento para los que se conoce perfectamente la salida deseada de la red. Un objetivo para diseñar la regla de aprendizaje supervisada podrá ser minimizar el error cometido entre las salidas (respuestas) de la red y las salidas (respuestas) deseadas. Se tendrán así Reglas de Aprendizaje basadas en la corrección del error, como la regla de retropropagación del error, en el caso del perceptrón y el algoritmo de mínimos cuadrados, muy utilizados en problemas de clasificación y predicción. En la Figura 22, se puede ver el diagrama de bloques de Aprender con un Maestro; la parte de La figura impresa en celeste constituye un bucle de retroalimentación.

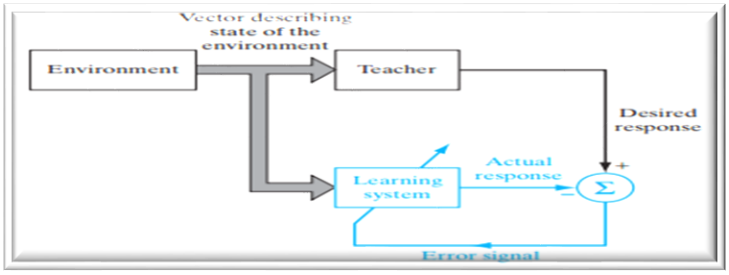


Figura 22. Aprendizaje supervisado. [45]

- **Aprendizaje no supervisado**, competitivo o auto-organizado, en el que se dispone de un conjunto de patrones de entrenamiento, pero no se conocerán las salidas deseadas de la red. La red buscará por sí misma el comportamiento más adecuado, atendiendo a cierto criterio, y encontrará estructuras o prototipos en el conjunto de patrones de entrenamiento. Como ejemplo se puede citar la "*Regla de Aprendizaje Competitivo No Supervisado*", utilizada en problemas de agrupación de patrones y obtención de

prototipos, la "*Regla de Kohonen*", utilizada en reconocimiento e identificación de patrones, y la "*Regla de Hebb*".

- **Aprendizaje por refuerzo**, basado en un proceso de prueba y error que busca maximizar el valor esperado de una función criterio conocida, como una señal de refuerzo. Si una acción supone una mejora en el comportamiento, entonces la tendencia a producir esta acción se refuerza, y en caso contrario de debilita. Por ello, se tiene un conjunto de patrones de entrenamiento, y sus correspondientes señales evaluativas, que suelen ser valores -1 o +1, en lugar de sus respuestas deseadas como en el caso supervisado. Dicha señal evaluativa informa a la unidad entrenada, sobre su comportamiento, con respecto a la entrada recibida. Es decir que evalúa la adecuación de su salida para dicha entrada.

3.2.11. El Modelo Perceptrón

En el Modelo propuesto por McCulloh-Pitts, la neurona tiene *n* entradas $\boldsymbol{x} = (\boldsymbol{x^1}, \ldots, \boldsymbol{x^n}) \in \boldsymbol{X} \subset \boldsymbol{R^n}$ y una salida $\mathbf{y} \in \{-\mathbf{1}, \mathbf{1}\}$. Este Modelo se puede ver en la Figura 9a. La salida está conectada con las entradas por una dependencia funcional:

$$\boldsymbol{y} = \boldsymbol{sign}\{(\boldsymbol{w} \cdot \boldsymbol{x}) - \boldsymbol{b}\} \qquad \textbf{(4)}$$

Dónde: $(\mathbf{w} \cdot \mathbf{x})$ es el producto interno de dos vectores, b es el valor de umbral o bias, y $\text{sign}(\mathbf{w} \cdot \mathbf{x}) = 1$ si $(\mathbf{w} \cdot \mathbf{x}) > 0$ y $\text{sign}(\mathbf{w} \cdot \mathbf{x}) = -1$ si $(\mathbf{w} \cdot \mathbf{x}) \leq 0$.

Desde el punto de vista geométrico, el perceptrón divide el espacio de entrada x en dos regiones, una región donde la salida y toma el valor de 1 y una región donde la salida y toma el valor de -1 (figura 23b). Estas dos regiones son separadas por un hiperplano:

$$(\boldsymbol{w} \cdot \boldsymbol{x}) - \boldsymbol{b} = \boldsymbol{0} \qquad \textbf{(5)}$$

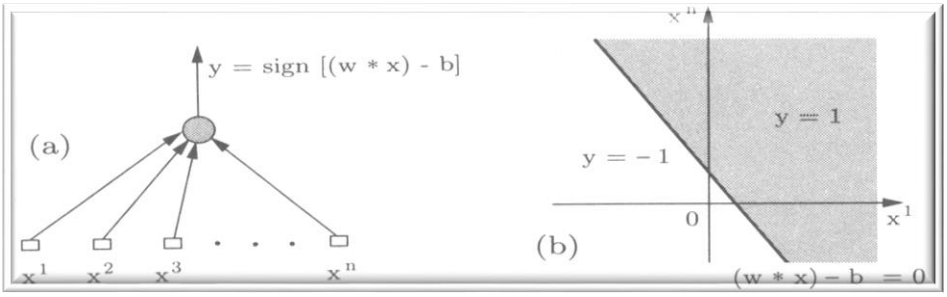


Figura 23 (a y b). Funciones del Perceptrón simple.

El vector ***w*** y el escalar ***b*** determinan la posición del hiperplano de separación, para poder realizar la separación adecuada de las dos clases, el perceptrón durante el proceso de aprendizaje selecciona los coeficientes apropiados de las neuronas.

Rosenblatt planteó un modelo de perceptrón que dividía el espacio ***x*** en dos partes separadas por trozos de una superficie, Figura 23b. Esto lo logra seleccionando los coeficientes adecuados para las neuronas de la red, Rosenblatt consideró para esta red, una estructura con varias neuronas. En este modelo, las salidas de las neuronas de la capa previa, eran las entradas de la neurona de la capa siguiente, y la capa de salida sólo tenía una neurona, es decir que era un perceptrón clásico con ***n*** entradas y una salida. Esta estructura se observa en la Figura 23a.

3.2.11.1. La Regla de Aprendizaje del Perceptrón

Es un tipo de *Aprendizaje por Corrección de Error* y, como ya se comentó, pertenece al tipo de *Aprendizaje Supervisado*. Éste se caracteriza porque el proceso se realiza mediante un entrenamiento controlado por un agente externo, supervisor o maestro, que determina la respuesta que debería generar la red, a partir de una entrada determinada. El supervisor controla la salida de la red, y en caso de que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones con el fin de conseguir que la salida obtenida, se aproxime a la deseada.

La regla consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos a la salida de la red, es decir, en función del error cometido en la salida. Esta es una regla muy simple, a cada neurona en la capa de salida se le calcula la desviación a la salida objetivo como el error, ***δ***. El mismo se utiliza luego para cambiar los pesos sobre la conexión de la neurona precedente. El cambio de los pesos por medio de la regla de aprendizaje del perceptrón se realiza según:

$$\mathbf{\Delta w_{ij} = \sigma * out_j * \left(a_{qi} - out_i\right)} \qquad \textbf{(6)}$$

Donde:

a_{qi} es la salida, deseada/objetivo, de la neurona de salida N_i, $\delta_i = (a_{qi} - out_i)$ la desviación objetivo de la neurona N_i, y σ el aprendizaje.

La salida de la neurona N_j (out_j) se utiliza, porque este valor influye en la entrada global y, por ende, en la activación y luego en la salida de la neurona N_i. Esto es semejante a un "*efecto en cadena*".

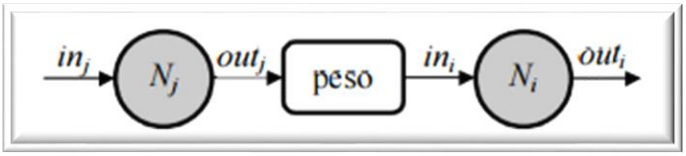


Figura 24. Influencia de la salida de la neurona N_j en la entrada de la neurona N_i

Un perceptrón simple asociado a un patrón de entrada particular, $\boldsymbol{x^p}$, se tiene una salida $\boldsymbol{o^p}$ y un "blanco" o salida correcta $\boldsymbol{t^p}$. El algoritmo de aprendizaje tiene la siguiente estructura: [6]

1. La red comienza en un estado aleatorio. Los pesos entre neuronas poseen valores aleatorios y pequeños (entre -1 y 1).
2. Seleccionar un vector de entrada, x_1, a partir del conjunto de ejemplos de entrenamiento.
3. Se propaga la activación hacia adelante a través de los pesos en la red para calcular la salida op = w . x^p
4. Si $\boldsymbol{o^p} = \boldsymbol{t^p}$, es decir, si la salida de la red es correcta, volver a 2.
5. En caso contrario el cambio de los pesos se realiza atendiendo a la siguiente expresión: $\boldsymbol{\Delta w_i = \eta x_i^p \left(t^p - o^p\right)}$ donde $\boldsymbol{\eta}$ es un número pequeño positivo conocido como coeficiente de aprendizaje. Volver al paso 2.

Lo que se hace, por tanto, es ajustar los pesos de una manera en la que las salidas de la red, $\boldsymbol{o^p}$, se vayan haciendo cada vez más semejantes al valor de los blancos, $\boldsymbol{t^p}$, a medida que cada entrada se $\boldsymbol{x^p}$, se va presentando a la red.

3.2.12. El Modelo Perceptrón Multicapa

El Perceptrón simple es capaz de resolver problemas de clasificación e implementar funciones lógicas, como por ejemplo, la función AND y OR, pero es incapaz de implementar la función lógica XOR. Sobre estas limitaciones, Minsky y Papert en 1969 publicaron un libro titulado "*Perceptrons*", que supuso el abandono por parte de muchos científicos de la

investigación en Redes Neuronales, pues no se encontraba un algoritmo de aprendizaje capaz de implementar funciones de este tipo [5][40].

Las limitaciones de las redes de una sola capa hicieron que se plantease la necesidad de implementar otro tipo de redes en las que se pudiera aumentar el número de capas. El introducir capas intermedias, o capas ocultas, entre la capa de entrada y la capa de salida, originó la posibilidad de aplicar cualquier función que tuviera el grado de precisión deseado, es decir, se consiguió que las redes multicapa fuesen aproximadores universales de funciones, lo que implica que dada una serie de patrones ***x***, y una serie de respuestas deseadas ***d***, el sistema encuentra los pesos ***w*** para ajustarse a las especificaciones. Estas redes, por lo tanto, tendrán la capacidad de conseguir reproducir el comportamiento de cualquier función matemática. Que ello suceda depende del correcto montaje de la red [45]. Esta capacidad fue la que las hizo populares rápidamente, como una herramienta de tipo "*caja negra*", para modelar relaciones entre variables. Por ejemplo, con un perceptrón multicapa se puede implementar la función lógica XOR. Entonces, una manera de solventar las limitaciones del perceptrón simple, es por medio de la inclusión de capas ocultas, obteniendo de esta forma una red neuronal que se denomina Perceptrón Multicapa (*PM* o MLP (*Multi-Layer Perceptron*) [5][40].

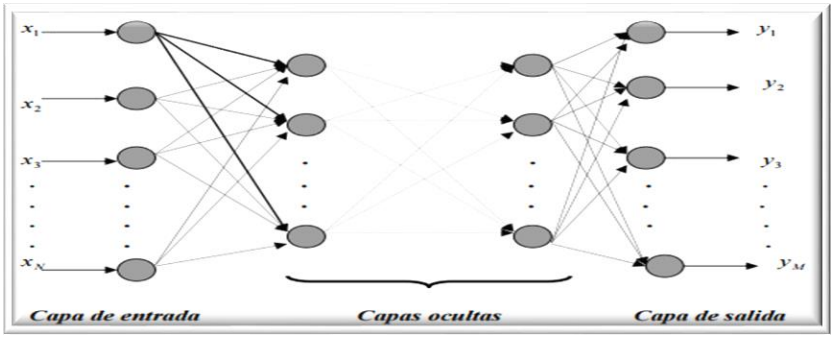


Figura 25. Ejemplo de una topología de un PM con dos capas ocultas.

La topología más complicada de estas redes dificulta la forma de encontrar los pesos correctos, ya que es el proceso de aprendizaje quién decide qué características de los patrones de entrada serán representadas por la capa oculta de neuronas.

En 1986 se abrió un nuevo panorama en el campo de las redes neuronales con el redescubrimiento por parte de Rumerlhard, Hinton y Williams del algoritmo de Retro-

propagación del error hacia atrás, más conocido como *BP*[40]. El algoritmo BP es un método eficiente para el entrenamiento de un perceptrón multicapa [5].

3.2.12.1. El algoritmo de retropropagación

Como se mencionó, en 1986, Rumelhart, Hinton y Williams, formalizaron un método para que una red neuronal aprendiera la asociación que existe entre los patrones de entrada y las clases correspondientes, utilizando varios niveles de neuronas. El método está basado en la generalización de la Regla Delta [40], y a pesar de sus limitaciones ha ampliado, de forma considerable, el rango de aplicaciones de las RNA. La BP trabaja bajo aprendizaje supervisado, y por tanto necesita un set de entrenamiento que le describa cada salida. El algoritmo funciona en dos fases, una hacia adelante, y una hacia atrás. En la fase de entrada, el patrón se presenta a la red, y se propaga a través de las capas hasta llegar a la capa de salida, obteniéndose los valores de salida de la red. La segunda fase (hacia atrás), inicia cuando se comparan los valores obtenidos con los valores de salida esperados, para así obtener el error. Esta fase transmite *"hacia atrás"* el error, a partir de la capa de salida, hacia todas las neuronas de la capa intermedia que contribuyen directamente a la salida, recibiendo el porcentaje de error aproximado a la participación de la neurona intermedia en la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido un error que describa su aportación relativa al error total. Basándose en el valor del error recibido, se reajustan los pesos de conexión de cada neurona, de manera que la siguiente vez que se presente el mismo patrón, la salida esté más cercana a la deseada y, por tanto, disminuya el error [10].

El algoritmo finaliza cuando se verifica su condición de parada, ya sea porque el error calculado de la salida es inferior al permitido, o porque se ha superado el número de iteraciones. Ante esto, se considera que se deberán hacer ajustes al diseño de la red, pues o no tiene solución, o se debe ampliar el número de iteraciones. Generalmente, la función que se utiliza es *Sigmoidal*. Este algoritmo se conforma de la tiene la siguiente manera: [5]

1. Inicializar los pesos a valores aleatorios y pequeños.
2. Escoger el patrón de entrada $\boldsymbol{x^P}$ y presentarlo a la capa de entrada.

[40] por sus siglas en inglés *Back Propagación*

3. Propagar la actividad hacia delante a través de los pesos hasta que la actividad alcance las neuronas de la capa de salida.
4. Calcular los valores de "δ" para las capas de salida:

$$\delta_i^p = \left(t_j^P - o_j^P\right) f'\left(Act_j^P\right) \quad \textbf{(7)}$$

usando los valores de los blancos deseados para el patrón de entrada seleccionado.

5. Calcular los valores de "δ" para la capa de oculta :

$$\delta_i^p = \sum_{j=1}^{N} \delta_i^p w_{ji} f'\left(Act_i^P\right) \quad \textbf{(8)}$$

6. Actualizar los pesos de acuerdo con $\Delta_P w_{ij} = \gamma \delta_i^P o_j^P$ **(9)**
7. Repetir del paso 2 al 6 para todos los patrones de entrada.

El algoritmo encuentra su valor mínimo de error, local o global, mediante la aplicación de pesos descendentes (algoritmo del gradiente descendente). Con el gradiente descendente, cada vez que se realizan cambios a los pesos de la red, se asegura el descenso por la superficie del error hasta encontrar el valle más cercano, lo que hace que el proceso de aprendizaje se detenga en un mínimo local de error. Uno de los problemas que presenta este algoritmo de entrenamiento, es que busca minimizar la función de error, pudiendo caer en un mínimo local, o en algún punto estacionario. Lo que significa que no se llegará a encontrar el mínimo global de la función de error. [46]

3.2.13. Tamaño de las Redes Neuronales

Un perceptrón multicapa es una red compuesta de varias capas de neuronas entre la entrada y la salida de la misma, esta red permite establecer regiones de decisión mucho más complejas que las de dos semiplanos, como lo hace el perceptrón de un solo nivel. El tamaño de las redes depende del número de capas y del número de neuronas ocultas por capa. El número de unidades ocultas está directamente relacionado con las capacidades de la red. Si bien se ha demostrado que la propiedad de aproximador universal de funciones de la red requiere de un máximo de dos capas ocultas, en la mayoría de los casos una única capa oculta resulta suficiente para conseguir óptimos resultados [47].

Para determinar el número de neuronas ocultas de cada capa se suele utilizar reglas "*ad hoc*" que, aunque no resulten matemáticamente justificables, han demostrado un buen comportamiento en diversas aplicaciones prácticas. Entre las que se encuentran:

- **La regla de la pirámide geométrica**: se basa en la suposición de que la capa oculta ha de ser inferior al total de variables de entrada, pero superior al número de variables de salida. $\sqrt{N * M}$ **(10)**

  Siendo ***N*** el número de variables de entrada, y ***M*** el total de neuromas de salida.
- **La regla de la capa oculta-capa entrada:** según esta regla, el número de capas ocultas está relacionado con el número de neuronas de entrada. En particular suele aplicarse la regla **2 x 1**, de forma que el número de neuronas ocultas no puede ser superior al doble del número de variables de entrada.

Las capacidades del perceptrón multicapa con dos, y tres capas, y con una única neurona en la capa de salida se muestran en la Figura 27, muestra las regiones de decisión que se obtienen para distintas arquitecturas de redes neuronales considerando dos neuronas en la capa inicial. Así por ejemplo para una arquitectura de perceptrón simple, la región de decisión es una recta, mientras que el perceptrón multicapa, con una única capa de neuronas ocultas, puede discriminar regiones convexas. Por otra parte el perceptrón multicapa, con dos capas de neuronas ocultas, es capaz de discriminar regiones de forma arbitraria. El perceptrón se suele entrenar por medio del algoritmo de retro-propagación de errores o BP.

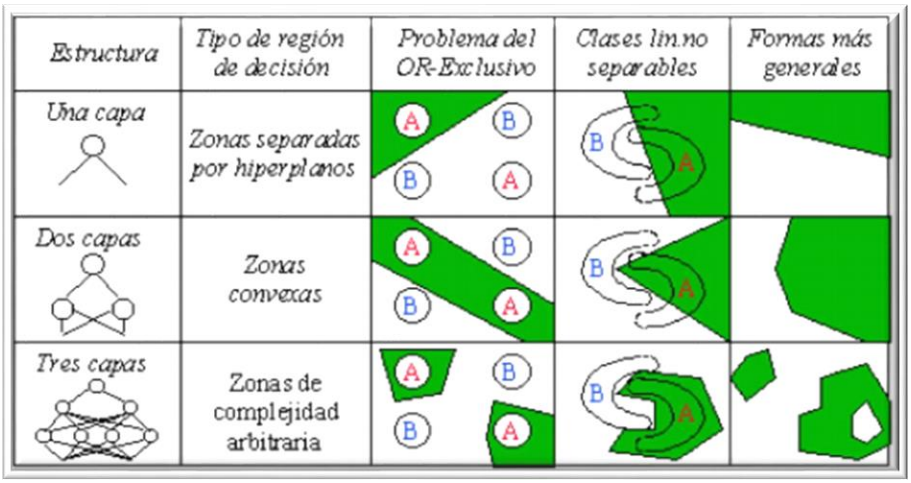


Figura 26. Arquitecturas y regiones de decisión.

Hasta la fecha, no se ha diseñado un ordenador que sea consciente de lo que está haciendo;
pero, la mayor parte del tiempo, nosotros tampoco lo somos.
Marvin Minsky

3.3. Modelos de Clasificación basados en Máquinas de Vectores Soporte.

Las Máquinas de Vectores Soporte, son un nuevo sistema de aprendizaje que ha tenido un desarrollo muy significativo en los últimos años, tanto en la generación de nuevos algoritmos, como en las estrategias para su implementación. Estos son un conjunto de algoritmos de aprendizaje supervisado desarrollados en 1963 por Vapnik y Lerneren, en los laboratorios AT&T [5].

Este algoritmo generaliza el método "*Generalized Portrait*", para la resolución de problemas de clasificación linealmente separables, mediante lo que se denomina hiperplano óptimo de separación (*Optimal Hyperplane Decision Rule*, OHDR) [49]. Pero no fue hasta los años noventa, con la publicación del primer paper por Boser en 1992 y por Cortes y Vapnik en 1995, cuando fue desarrollado y generalizado [5]. Fue ideada originalmente para la resolución de problemas de clasificación binarios, en los que las clases eran linealmente separables. El algoritmo consistía en generar una solución en la que se clasificaban de manera correcta todas las muestras, colocando el hiperplano de separación lo más lejos posible de todas ellas [51]. En la Figura 28 se puede observar un hiperplano con un ejemplo de separación binaria. Dentro de la tarea de clasificación, las MVS pertenecen a la categoría de los clasificadores lineales, puesto que inducen separadores lineales o hiperplanos. Esto sucede ya sea en el espacio original de los ejemplos de entrada, si éstos son separables o cuasi- separables (ruido), o en un espacio transformado, o espacio de características, si es que los ejemplos no son separables linealmente en el espacio original.

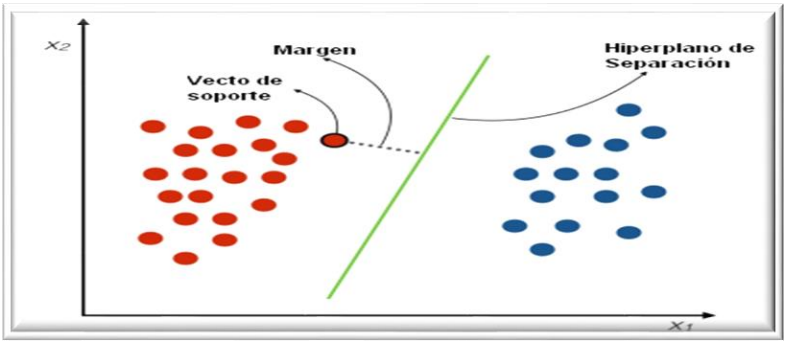


Figura 27. Hiperplano de separación de datos linealmente separables.

Para definir el concepto de hiperplano se usará la siguiente: *"...en geometría, Hiperplano es una generalización del concepto de plano. En un espacio de una única dimensión (como una recta), un hiperplano es un punto; divide una línea en dos líneas. En un espacio bidimensional (como el plano xy), un hiperplano es una recta; divide el plano en dos mitades. En un espacio tridimensional, un hiperplano es un plano corriente; divide el espacio en dos mitades. Este concepto también puede ser aplicado a espacios de cuatro dimensiones y más, donde estos objetos divisores se llaman simplemente hiperplanos, ya que la finalidad de esta nomenclatura es la de relacionar la geometría con el plano..."*

Mientras la mayoría de los métodos de aprendizaje se centran en minimizar los errores cometidos por el modelo generado a partir de los ejemplos de entrenamiento (error empírico), el sesgo inductivo asociado a las MVS radica en la minimización del denominado riesgo estructural. El sesgo inductivo de un algoritmo de aprendizaje, es el conjunto de afirmaciones que el algoritmo utiliza para clasificar instancias nuevas. La idea es seleccionar un hiperplano de separación que equidista de los ejemplos más cercanos de cada clase para, de esta forma, conseguir lo que se denomina un *margen máximo* a cada lado del hiperplano. Además, a la hora de definir el hiperplano, sólo se consideran los ejemplos de entrenamiento de cada clase, que se encuentran justo en la frontera de dichos márgenes. Estos ejemplos reciben el nombre de *vectores soporte*. Desde un punto de vista práctico, el hiperplano separador de margen máximo ha demostrado tener una buena capacidad de generalización, evitando en gran medida el problema del sobreajuste a los ejemplos de entrenamiento. Desde un punto de vista algorítmico, el problema de optimización del margen geométrico, representa un problema de optimización cuadrático con restricciones lineales, que puede ser resuelto mediante técnicas estándar de programación cuadrática. Es el problema de optimizar, reduciendo al mínimo, o maximizando una función cuadrática, varias variables conforme a apremios lineales en estas variables. La propiedad de convexidad exigida para su resolución garantiza una solución única, en contraste con la no unicidad de la solución producida por una red neuronal artificial entrenada con un mismo conjunto de ejemplos [50]. Este trabajo sólo abarca una pequeña porción del extenso campo que trata con las máquinas vectores soporte. Se describirá el problema de clasificación sólo para el caso de clases binarias.

3.3.1. MVS para clasificación binaria de ejemplos separables linealmente

Se dice que un problema es linealmente separable cuando, para cualquier conjunto de muestras existe, un único hiperplano que clasifica con error cero. De entre todo el conjunto de muestras, se extraen una serie de vectores, los vectores soporte, que son los únicos que se necesitan, de entre todos los datos, para definir la frontera de decisión [5].

Usando multiplicadores de Lagrange[41], es posible representar el hiperplano deseado como combinación lineal de los propios datos. Estos son un método para trabajar con funciones de varias variables que interesa maximizar o minimizar, y está sujeta a ciertas restricciones. Este método reduce el problema restringido en ***n*** variables en uno sin restricciones de ***n*** + 1 variables, cuyas ecuaciones pueden ser resueltas. Este método introduce una nueva variable escalar desconocida, el *multiplicador*, para cada restricción y forma una combinación lineal involucrando los multiplicadores como coeficientes. Su demostración involucra derivadas parciales. Se demuestra que la gran mayoría de los *coeficientes de Lagrange* serán nulos, y que únicamente serán distintos de cero aquellos puntos situados exactamente a la distancia marcada por el margen, estos son los vectores soporte, que actuarán como resumen de todo el conjunto de datos, en el sentido que la solución únicamente depende de ellos. De este modo, si se entrena una máquina sólo teniendo en cuenta los vectores soporte, se obtendrán los mismos resultados que entrenando con todo el conjunto de datos. En la Figura 17 se puede observar un caso binario linealmente separable.

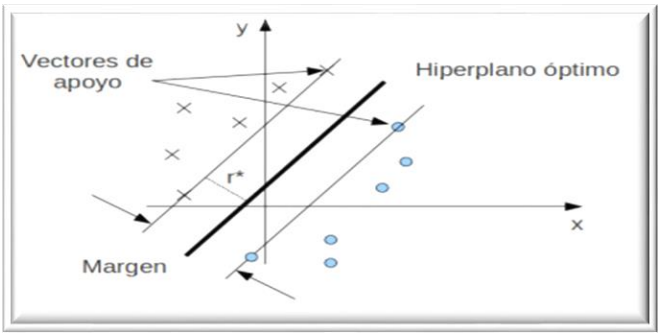


Figura 28. Hiperplano para un caso linealmente separable.

[41] Son usados en problemas de optimización, el método es llamado así en honor a Joseph Louis Lagrange. [20]

Matemáticamente, dadas N muestras, x_i, con sus correspondientes etiquetas o labels asociados, y_i, que definirán a qué clase pertenece cada muestra, tenemos

$$(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_n, \mathbf{y}_n);\ \mathbf{x}_1 \in \mathbf{R}^j, \mathbf{y} \in \{+\mathbf{1}, -\mathbf{1}\} \quad \textbf{(11)}$$

siendo J el número de dimensiones o componentes de los vectores que contienen los datos.

Un clasificador es lineal si su función de decisión puede expresarse mediante una función lineal en x. Así pues, la ecuación del hiperplano de separación será el lugar de los puntos x en los que se cumple

$$H : w \cdot x + b = 0 \quad \textbf{(12)}$$

siendo b una constante que indica la posición del plano respecto al origen de coordenadas. Esta constante recibe el nombre de sesgo. w es el vector normal al hiperplano y tiene la forma:

$$\mathbf{w}^* = \sum_{\mathbf{i}=\mathbf{1}}^{\ell} \boldsymbol{\alpha}_\mathbf{i}^* \mathbf{y}_\mathbf{i} \mathbf{x}_\mathbf{i} \quad \textbf{(13)}$$

donde los α_i son los multiplicadores de Lagrange, los cuales como se ha mencionado anteriormente, serán nulos en su mayoría, salvándose únicamente los que sean Vectores Soporte. La clasificación se realiza determinando en qué zona del hiperplano está el punto a clasificar. Así pues, el clasificador puede representarse mediante las expresiones

$$\mathbf{w} \cdot \mathbf{x}_\mathbf{i} + \mathbf{b} \geq +\mathbf{1} \quad \forall \mathbf{y}_\mathbf{i} = +\mathbf{1} \quad \textbf{(14)}$$

$$\mathbf{w} \cdot \mathbf{x}_\mathbf{i} + \mathbf{b} \geq -\mathbf{1} \quad \forall \mathbf{y}_\mathbf{i} = -\mathbf{1} \quad \textbf{(15)}$$

Si el resultado de la operación es positivo, la muestra pertenecerá a una clase, y si es negativo a la otra. Las expresiones anteriores pueden combinarse en una única

$$y_i \{w \cdot x_i + b\} \geq 1 \quad \textbf{(16)}$$

En un problema linealmente separable habrá infinitos hiperplanos que cumplan esta condición. El que se busca, es aquel que tenga un mayor margen. Es decir, se quiere maximizar la distancia entre los datos y la frontera de decisión. Los puntos que caen sobre cada uno de los hiperplanos son los que cumplen:

$$\mathbf{H}_\mathbf{i} : \mathbf{w} \cdot \mathbf{x}_\mathbf{i} + \mathbf{b} = +\mathbf{1} \quad \textbf{(17)}$$

$$\mathbf{H_i : w \cdot x_i + b = -1} \quad \textbf{(18)}$$

Por tanto estos dos hiperplanos son paralelos entre sí, y paralelos al hiperplano ***H***. El margen será la distancia entre ***H1*** y ***H2***. Lo que se busca aquí es aumentar la generalización, es decir, que una vez entrenada la máquina, clasifique correctamente las muestras nuevas.

Para ello, cuanto mayor sea la distancia entre los datos y la frontera de clasificación, mejor. Si se desea que los datos más cercanos a la frontera tengan una salida ±1, la distancia de los datos al plano será $d = 1/\|w\|$, que será la distancia de ***H1*** y ***H2***, a ***H***. Así pues, el margen del hiperplano ***H*** (distancia a los vectores más cercanos pertenecientes a diferentes clases) a maximizar es $2/\|w\|$. Maximizar este margen es equivalente a minimizar la norma de w. De este modo, el problema de encontrar el hiperplano óptimo puede formularse como:

$$L(w) = min\left\{\frac{1}{2}\|w\|^2\right\} \quad \textbf{(19)}$$

con la restricción (15), que garantiza que el hiperplano separará las muestras de distintas clases. Si usamos los multiplicadores de Lagrange, e incorporamos la restricción (16), la expresión a minimizar es

$$L_d = \frac{1}{2}\|w^2\| - \sum_{i=1}^{n} \alpha_i (y_i (x_i . w + b) - 1) \quad \textbf{(20)}$$

Así pues, para minimizar la expresión anterior, se deberá derivar con respecto a w y b e igualar a 0. Esto deja dos ecuaciones:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \quad \textbf{(21)}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \quad \textbf{(22)}$$

Teniendo en cuenta las ecuaciones anteriores, la MVS actuando como clasificador, y para el caso en el que el problema sea linealmente separable, puede escribirse como:

$$f(x) = sign\left(\sum_{i=1}^{n} \alpha_i y_i (x_i x) + b\right) \quad \textbf{(23)}$$

3.3.2. MVS para clasificación binaria de ejemplos No separables linealmente

La MVS no lineal se puede interpretar como una generalización del hiperplano óptimo de decisión, ya que permite la resolución de problemas no separables y trazar fronteras de clasificación no lineales [50]. Así, cuando los datos no son linealmente separables, puede aplicarse una transformación $\boldsymbol{\Theta}(\boldsymbol{x})$ sobre el espacio de trabajo, con el objetivo de obtener un espacio de características generalmente de dimensión superior, donde sí sean separables las muestras, y donde se debe trazar el hiperplano óptimo de separación. La formulación es básicamente la misma, únicamente reemplazando $\boldsymbol{x}$ por $\boldsymbol{\Theta}(\boldsymbol{x})$. Figura 30.

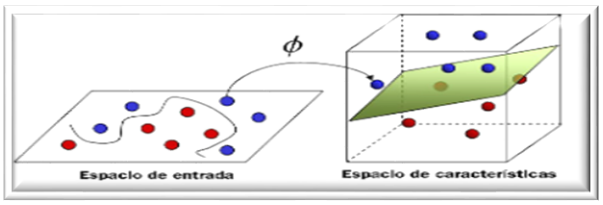


Figura 29. Transformación espacial del espacio de entrada.

El *Teorema de Cover* [50] proporciona la justificación al aumento de las posibilidades de disponer de un conjunto de datos separables, mediante una transformación no lineal hacia un espacio de mayor dimensión, si estos no lo eran en el espacio de entrada. Para construir una SVM en el espacio resultante, este debe ser un Espacio de Hilbert.

Para definir el concepto de Espacio de Hilbert se usará, nuevamente, la siguiente definición "*...en matemáticas, el concepto de espacio de Hilbert es una generalización del concepto de espacio euclídeo. Esta generalización permite que nociones y técnicas algebraicas y geométricas aplicables a espacios de dimensión dos y tres se extiendan a espacios de dimensión arbitraria, incluyendo a espacios de dimensión infinita. Ejemplos de tales nociones y técnicas son la de ángulo entre vectores, ortogonalidad de vectores, el teorema de Pitágoras, proyección ortogonal, distancia entre vectores y convergencia de una sucesión. El nombre dado a estos espacios es en honor al matemático David Hilbert quien los utilizó en su estudio de las ecuaciones integrales...*". Este espacio debe cumplir:

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}_i) = \boldsymbol{\Theta}(\boldsymbol{x}), \boldsymbol{\Theta}(\boldsymbol{x}_i) \qquad \textbf{(24)}$$

donde la función $\boldsymbol{k}$ es llamada *Función núcleo* o *Función kernel*.

Teniendo esta función, es posible aplicar el algoritmo de entrenamiento de las MVS sin conocer $\boldsymbol{\Theta}$. Existen distintas funciones kernel que permiten adaptar la MVS a cada

conjunto de muestras, con el fin de obtener mejores resultados. Es decir, este clasificador puede trabajar de modo lineal o no lineal. Para hacer esta diferenciación, el clasificador se basa en la utilización de diferentes *Kernel*, entendiendo por *kernel* a la forma que tendrá la separación generada entre clases dentro del hiperplano. En la literatura referente a este concepto, se destacan tres tipos de *kernel* cuyo uso se encuentra más extendido, y son utilizados en la gran mayoría de los estudios realizados actualmente:

- El primer *kernel* que se desarrolló se denomina ***kernel lineal***. Es el más básico y sencillo de todos y el que se utiliza habitualmente. Este *kernel* sigue la ecuación (24) y su funcionamiento se basa en el cálculo de la función que defina la línea, plano o hiperplano de grado ***n*** mediante el cual la separación entre clases del set de muestras sea óptima, es decir, que maximice la distancia entre la separación y los puntos de entrenamiento.

$$K(x, x_i) = (x * x_i) \qquad \textbf{(25)}$$

- El ***kernel polinomial*:** este es un kernel muy utilizado para modelar relaciones no lineales. Sin embargo, a medida que aumenta el parámetro ***d*** (grado del polinomio) la superficie de clasificación se hace más compleja.

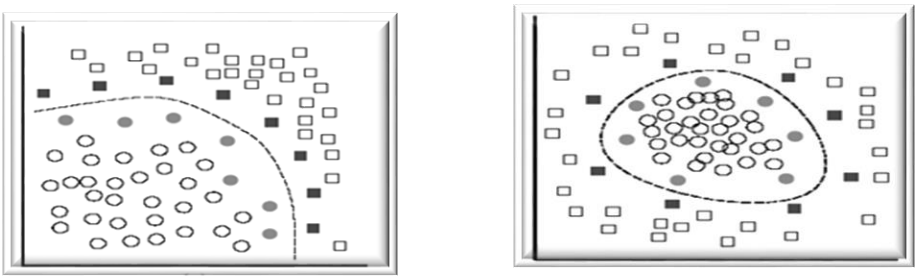
$$K(x,\ x_i) = (1 + x,\ x_i)^d \qquad \textbf{(26)}$$

- Por último, se encuentran los kernel basados en ***Funciones de Base Radial*** (RBF[42]). De manera genérica, siguen la ecuación (24) y, en concreto, uno de los más utilizados es el GBF (*Gaussian Radial Basis function*) que sigue la variante (26) de la ecuación general. Un ejemplo gráfico de la diferencia que existe entre el kernel polinominal (a) y el presentado en este punto (b), puede observarse en la Figura 31.

$$K(x_i, x_j) = exp\left(-\gamma \left\| x_i - x_j \right\|^2\right), \quad \textbf{para } \gamma > 0 \qquad \textbf{(27)}$$

$$K(x_i, x_i) = exp\left(-\frac{\left\| x_i - x_j \right\|^2}{2\sigma^2}\right) \qquad \textbf{(28)}$$

[42] por sus siglas en inglés *Radial Basis Function*.



a) **b)**

Figura 30. Hiperplanos de decisión.

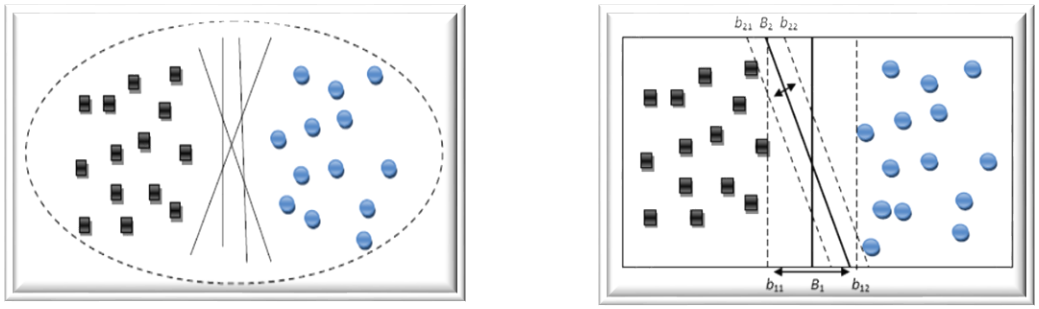
Las MVS han ganado popularidad como herramienta para la identificación de sistemas lineales y no lineales. Esto último, debido principalmente a que el algoritmo está basado en el principio de *Minimización del Riesgo Estructural*. Principio originado en la Teoría de Aprendizaje Estadístico y desarrollada por Vapnik en 1998 [50], el cual como ya fue expuesto, ha demostrado ser superior al principio de *Minimización del Riesgo Empírico* utilizado por las RNA. Algunas de las razones por las que este método ha tenido éxito, es que no padece de mínimos locales y el modelo sólo depende de los datos con más información, llamados vectores de soporte. Algunas de las ventajas que tienen las MVS son [5]:

- Excelente capacidad de generalización, debido a la minimización del riesgo estructurado.
- Pocos parámetros a ajustar; el Modelo solo depende de los Datos con mayor información.
- La estimación de los parámetros se realiza a través de la optimización de una función de costo convexa, lo cual evita la existencia de un mínimo local.
- La solución de MVS es "*Sparse*" (*Datos dispersos*). Esto significa que la mayoría de las variables son 0 en la solución de MVS. Así, el Modelo final puede ser escrito como una combinación de un número muy pequeño de vectores de entrada, llamados vectores de soporte.

Mientras la mayoría de los métodos de aprendizaje se centran en minimizar los errores, cometidos por el modelo generado a partir de los ejemplos de entrenamiento, el sesgo inductivo asociado a las MVS radica en la minimización del denominado riesgo estructural. La idea es seleccionar un hiperplano de separación que equidiste de los ejemplos más cercanos de cada clase para, de esta forma, conseguir lo que se denomina un *margen máximo* a cada lado del hiperplano.

3.3.3. Margen Hiperplano Máximo

En el trabajo de Luis González Abril [52], se presenta una gráfica de ejemplo, donde un conjunto de elementos que pertenecen a dos diferentes clases, representados como cuadrados y círculos, son linealmente separables si se puede encontrar un hiperplano tal, que todos los cuadrados residan en un lado del hiperplano, y todos los círculos residan en el otro lado. Como se muestra en la Figura 32(a) existen infinitos hiperplanos posibles. Aunque sus errores de entrenamiento fueran cero, no hay garantía de que los hiperplanos respondan igualmente bien en ejemplos no vistos previamente. Por tanto, para representar su límite de decisión, el clasificador debe elegir uno de estos hiperplanos, con la intención de que responda acertadamente con los ejemplos de prueba, es decir posea un error de generalización mínimo. Para tener un panorama claro de cómo las elecciones posibles de hiperplanos diferentes afectan los errores de generalización, se consideran los dos límites de decisión, $\boldsymbol{B_1}$ y $\boldsymbol{B_2}$ mostrados en la Figura 32(b).



32 a. **32 b.**

Figura 31. Margen Hiperplano.[52]

Ambos límites de decisión pueden separar los ejemplos de entrenamiento en sus clases respectivas sin cometer ningún error de mala clasificación. Cada límite de decisión $\boldsymbol{B_i}$ está asociado con un par de hiperplanos, denominados como $\boldsymbol{b_{i1}}$ y $\boldsymbol{b_{i2}}$, respectivamente.

El hiperplano $\boldsymbol{b_{i1}}$ se obtiene moviendo un hiperplano paralelo lejos del límite de decisión, hasta que toca el/los cuadrado/s más cercano/s, mientras que $\boldsymbol{b_{i2}}$ se obtiene moviendo el hiperplano hasta que toca a él/los círculo/s más cercano/s.

La distancia entre estos dos hiperplanos es conocida como "*margen del clasificador*". Desde el diagrama mostrado en la Figura 32(b), se observa que el margen para $\boldsymbol{B_1}$ es

considerablemente mayor que para $\boldsymbol{B_2}$. En ese ejemplo $\boldsymbol{B_1}$, llega a ser el margen máximo del hiperplano para los ejemplos de entrenamiento.

3.3.4. Razones para un Margen Máximo

Los límites de decisión con márgenes grandes, tienden a tener menores errores de generalización que aquellos con márgenes pequeños. Intuitivamente, si el margen es pequeño, entonces cualquier leve perturbación del límite de decisión, puede tener un impacto significativo en su clasificación. Sin embargo, una separación perfecta no siempre es posible y, si lo es, el resultado del modelo no puede ser generalizado para otros datos. Esto se conoce como sobreajuste (*overfitting*) [52]. El principio de aprendizaje estadístico *SRM* provee un límite superior al error de generalización de un clasificador (R). En términos de su error de entrenamiento (R_e), el número de ejemplos de entrenamiento (N) y la complejidad del Modelo, conocida también como su capacidad (h). Más específicamente, con una probabilidad de 1 - n, el error de generalización del clasificador puede ser, como máximo:

$$R \leq R_e + \phi\,(h / N,\ log(h) / N) \qquad (29)$$

donde ϕ es una función creciente monótona de la capacidad h.

Con esta consideración, el SRM expresa el error de generalización R, en relación al error de entrenamiento y la capacidad del modelo. La capacidad de un Modelo lineal es inversamente proporcional a su margen. Los modelos con márgenes pequeños tienen capacidades mayores porque son más flexibles, y pueden encajar con varios conjuntos de entrenamiento distinto a los modelos con márgenes grandes.

De acuerdo al principio SRM, si aumenta la capacidad, el límite de error de generalización también aumentará. Por lo tanto, es deseable diseñar clasificadores lineales que maximicen los márgenes de sus límites de decisión, que a su vez minimizará la capacidad del modelo, asegurando que los errores de generalización también sean mínimos.

3.3.5. Minimización del Riesgo Estructural (SRM)

El riesgo estructural nace como necesidad de incorporar la capacidad de generalización de manera explícita en la construcción de un modelo predictivo, y prevenir de esta forma, el

problema de sobreajuste [46]. Se define la capacidad de una máquina de aprendizaje como la habilidad para aprender cualquier conjunto de entrenamiento sin error. En el ejemplo de la Figura 33 se observa que, si se busca clasificar una muestra de árboles de acuerdo a si son de color verde o no, el poder discriminatorio de esta función es muy bajo, asumiendo que casi todos los árboles son verdes, y por ende esta clasificación no logra hacer una diferencia importante. Por otro lado, si se clasifica de acuerdo al número de hojas, es probable que ningún árbol tenga el mismo número que otro, por lo tanto, la función de clasificación se está ajustando demasiado a los datos y perdiendo su capacidad de generalización. En el diamante de la Figura 33, se define la clasificación usada en los datos, la cual genera dos o más subconjuntos de acuerdo a si cumplen, o no, la regla de clasificación, y son simbolizados por el árbol sin cruz y con cruz respectivamente.

Figura 32. Tipos de ajustes.

Para solucionar este problema, los métodos que aplican este criterio y en particular las MVS, buscan el balance correcto entre precisión y capacidad de generalización. Para ello se restringe la complejidad de la función de clasificación. Intuitivamente una función simple que explique más de los datos, es preferible a una más compleja. En el trabajo de Galvão [45], se demuestra que maximizando el margen de separación entre clases, es posible generar funciones de clasificación que logren este balance.

3.3.6. Algunas características de las Máquinas de Vectores Soporte (SVM)

1- Al maximizar, las SVM, el margen del límite de decisión, determinan la tarea de aprendizaje. No obstante el usuario todavía debe proveer otros parámetros, tales como el tipo de función kernel a usar, y la penalidad asociada a la función *C*.

2- Las MVS pueden ser aplicadas a datos categóricos introduciendo variables ficticias (*dummy*). Por ejemplo si "*Sexo*" toma dos valores; *Masculino*, *Femenino*, podemos introducir una variable binaria para cada valor del atributo "Sexo".

3- La formulación de MVS presentada en este trabajo, es realizada para problemas de clasificación binaria.

3.3.7. MVS y la clasificación Binaria: Propiedades destacables.

- El entrenamiento de una MVS es básicamente un problema de programación cuadrática (QP) convexa, que es atractivo por dos motivos, su eficiente computación, existen paquetes software que permiten su resolución eficientemente, y por la garantía de encontrar un extremo global de la superficie de error, ya que nunca alcanzará mínimos locales. La solución obtenida es única y la más óptima para los Datos de entrenamiento dados.
- A la vez que minimiza el error de clasificación en el entrenamiento, maximiza el margen para mejorar la generalización del clasificador.
- No tiene el problema de *Sobreentrenamiento* (*overfitting*) como podría ocurrir en las redes neuronales. La solución no depende de la estructura del planteamiento del problema.
- Permite trabajar con relaciones no lineales entre los datos (genera funciones no lineales, mediante kernel). El producto escalar de los vectores transformados se puede sustituir por el kernel por lo que no es necesario trabajar en el espacio extendido. Generaliza muy bien con pocas muestras de entrenamiento.

3.3.8. Limitaciones de las MVS.

Pese a los excelentes resultados obtenidos, las SVM tienen algunas limitaciones:

- La elección de un núcleo adecuado es todavía un área abierta de investigación. Una vez elegido el núcleo, los clasificadores basados en MVS tienen como único parámetro a ajustar por el usuario, la penalización del error C.
- La complejidad temporal y espacial, tanto en el entrenamiento como en la evaluación, son también una limitación. Es un problema sin resolver el entrenamiento con grandes conjuntos de datos, del orden de millones de vectores soporte. Los algoritmos existentes para resolver esta familia de problemas tardan un tiempo que depende cuadráticamente del número de puntos.
- Fue inicialmente creado para clasificación binaria. Aunque hay ya algunos trabajos que estudian el entrenamiento de SVM multiclase en una sola pasada, aún se está lejos de diseñar un clasificador multiclase óptimo basado en SVM.
- La SVM siempre soluciona un problema bloque, un cambio en los patrones de entrenamiento supone obtener una nueva SVM, pues pueden surgir distintos vectores soporte, aunque existen ya algunas alternativas para el entrenamiento on-line de SVM capaces de encajar modificaciones del conjunto de datos sin necesidad de reentrenar el sistema.

Capítulo 4:

4. Caso de Estudio y Experimentación

La deserción universitaria es un problema que aqueja tanto a la educación pública, como privada en la Argentina. En este trabajo el objetivo de la investigación se fundamentó en utilizar técnicas de MD, Redes Neuronales Artificiales, y Máquinas de Vector Soporte, para clasificar a los estudiantes de los primeros años de las carreras de Ingeniería de la Universidad Nacional de La Matanza en dos clases o categorías, con posibilidad o no de desertar. Esta clasificación se realizó a partir de la información referida al rendimiento académico, y a algunas variables socio-económicas de los alumnos de las que se tenía disponibilidad, como edad, sexo, etc. Como ya se comentó, aprender cómo clasificar objetos a una de las categorías, o clases, previamente establecidas, es una característica de la MD, y una de sus principales características reside en que la habilidad de realizar una clasificación, y de aprender a clasificar, otorga el poder de tomar decisiones [5]. Francisco José García González en [53], da la siguiente definición: **"…***Sea E un conjunto de datos, el objetivo de la clasificación es aprender una función, L: X→Y, denominada clasificador, que represente la correspondencia existente en los ejemplos entre los vectores de entrada y el valor de salida correspondiente…".* La función aprendida será capaz entonces, de determinar la clase para cada nuevo ejemplo sin etiquetar. Es importe resaltar que el éxito de un algoritmo de aprendizaje para la clasificación depende en gran medida de la calidad de los datos que se le proporcionan [5].

En este trabajo se presenta la aplicación de la metodología KDD para la extracción de reglas de clasificación de alumnos universitarios, usando como datos de entrada un almacén de datos proporcionado por el DIIT. En muchas ocasiones, dentro de un conjunto numeroso de datos existen patrones que no son observables a simple vista, y que pueden ser útiles en el estudio de un campo determinado. Las técnicas de MD ya se han empleado con éxito para crear modelos de predicción del rendimiento de los estudiantes, obteniendo resultados prometedores que demuestran cómo determinadas características sociológicas, económicas, y educativas de los alumnos, pueden afectar en el rendimiento académico. En varios estudios

aplicados a la deserción Universitaria, se han usado distintos métodos de clasificación. Siendo las (RNA), una de las utilizadas [17, 54-58]. Por otro lado, las MVS constituyen nuevas estructuras de aprendizaje automático que han demostrado un excelente desempeño en aplicaciones de clasificación [59-61].

Este trabajo presenta una metodología formal para la extracción del conocimiento, a partir de los datos que dispone una universidad. Los datos que residen en las bases de datos corporativas, pueden ser una de las fuentes de conocimiento más importantes que hay en las organizaciones, por lo que su administración eficiente es de especial importancia. Inicialmente se explicará la metodología aplicada, en la cual se brindan los principales alcances relacionados con la deserción universitaria, y los modelos de clasificación que serán evaluados. Posteriormente se desarrollarán las mejores configuraciones para cada clasificador, sección en la que se crea cada uno de los modelos. Luego se procederá a desarrollar el análisis comparativo de los modelos de clasificación Para ello, se presentarán los resultados obtenidos mediante la aplicación de los modelos guardados previamente, a partir de una data de entrada compuesta por 1499 registros. Posteriormente se presentará la comparación de los modelos de clasificación, señalando las bondades y dificultades de la aplicación de cada uno. Finalmente se brindarán las conclusiones, y recomendaciones, derivadas del presente artículo. La metodología propuesta es la KDD, que consta de seis fases como se describió en el Capítulo I. Estas fases se observan en la Figura 34.

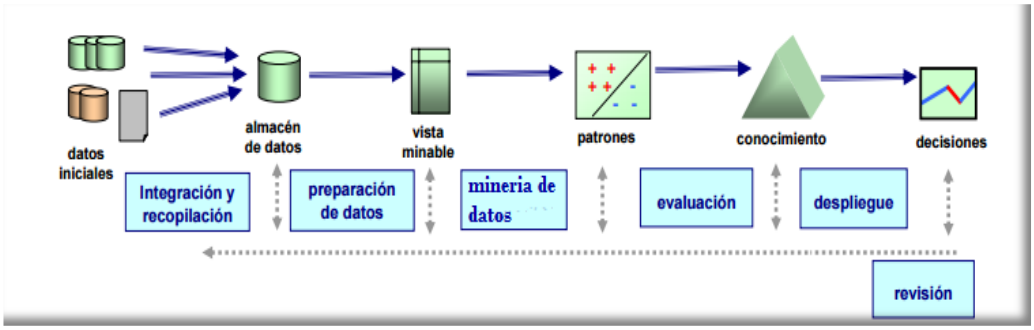


Figura 33: Etapas que componen un proceso KDD. [1]

A continuación se detallan los procesos que se llevaron a cabo siguiendo la metodología planteada:

4.1. Integración y Recopilación de los Datos

Las universidades utilizan sistemas de información adecuados a sus necesidades y características propias, tales como estructura de los planes, modalidad de cursada, y constitución geográfica de sus sedes, entre otras características. Si bien existen diferencias notables entre las distintas instituciones, es posible mantener criterios comunes para el desarrollo de sistemas informáticos, que permitan la gestión de los datos de los alumnos desde sus primeros días en la universidad, hasta que culminan sus estudios.

El Sistema de Información Universitario (SIU[43]), es el organismo que desarrolla, desde el año 1996, sistemas informáticos para el Sistema Universitario Nacional, la Secretaria de Políticas Universitarias y distintas áreas del Ministerio de Educación de la Nación. En particular el SIU Guaraní[44], es el sistema utilizado para el llenado del almacén de datos departamental, que se emplea para este trabajo. Este sistema de información permite el seguimiento de todas las actividades que realiza un estudiante, desde su inscripción a exámenes y cursadas, pasando por la reinscripción a carreras, consulta de inscripciones, consulta de plan de estudios e historia académica, consulta de cronograma de evaluaciones parciales, consulta de créditos, notas de evaluaciones parciales, materias regulares, actualización de datos censales, y hasta incluso la recepción de mensajes.

4.2. Preparación de los datos.

Esta etapa concuerda en parte con una de los principales proceso para la construcción de un DW, y se conoce como el proceso de "*Extraer, Transformar y Cargar"* , frecuentemente abreviado como ETL, por sus siglas en inglés *Extract, Transform and Load*. Este es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, como el sistema transaccional SIU-Guaraní, para reformatearlos, limpiarlos, y cargarlos en otra base de datos, data mart, almacén de datos, o data warehouse, para analizarlos, o también en otro sistema operacional para apoyar un proceso de negocio, es decir, para lograr que sean accesibles para el análisis y toma de decisiones [25-27]. Cabe recordar que los almacenes de

[43] http://www.siu.edu.ar/
[44] http://portal.comunidad.siu.edu.ar/micrositios/siu-guarani

datos deben ser adecuados para el procesamiento analítico en lineamiento (OLAP) [29]. La Figura 35 describe el modelo datos que compone el DW institucional del DIIT.

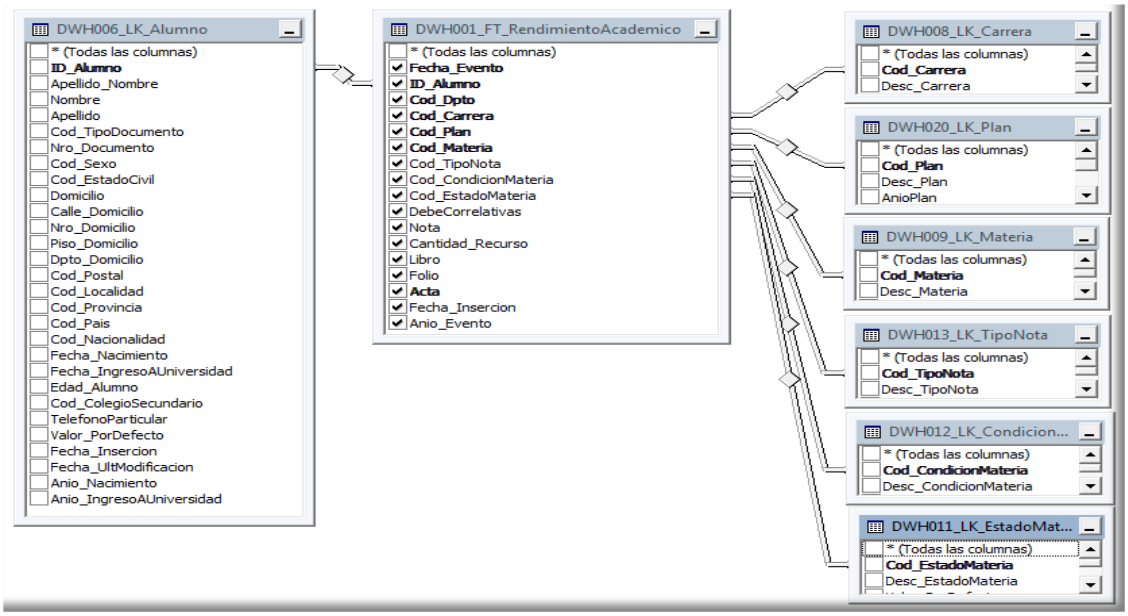


Figura 34. Tabla de Hecho y de dimensiones que conforman el DW departamental.

Para este estudio se analizaron los datos de alumnos de las carreras de Ingeniería en Informática[45], Civil[46], Electrónica[47] e Industrial[48], haciendo foco en la cohorte 2013, 2014 y 2015. Se contó con una población de 718, 781 y 849 alumnos, respectivamente. En la Tabla 3 se muestra la relación entre los alumnos que no abandonaron y los que sí lo hicieron.

Año	No Abandonan	Porcentaje	Abandonan	Porcentaje	Totales
2013	**599**	**83.43%**	**119**	**16.57%**	**718**
2014	**669**	**85.66%**	**112**	**14.34%**	**781**
2015	**727**	**85.63%**	**122**	**14.37%**	**849**

Tabla 3. Porcentajes de alumnos por año

4.3. Definición de deserción

Como ya mencioné, no existe una única definición de deserción que pueda captar en su totalidad la complejidad de este fenómeno. De acuerdo con Tinto [8] la definición de la deserción estudiantil puede analizarse desde varias perspectivas y de acuerdo con los diferentes tipos de abandono. Estas perspectivas dependen de las partes involucradas e interesadas en el proceso, como son los estudiantes, los funcionarios de las instituciones de educación superior y los responsables de la política nacional de educación.

De acuerdo con lo anterior, y para el desarrollo de los modelos que a continuación se realizarán, dentro del presente documento se definirá la deserción utilizando la siguiente perspectiva, será considerado un desertor aquel estudiante que abandona la institución educativa durante tres períodos consecutivos de inscripción, como resultado de la interacción o del efecto individual, y el combinado de diferentes categorías de variables: individuales, académicas, institucionales, y socioeconómicas. En términos cuantitativos, y considerando que en la UNLaM por año calendario se realizan 3 inscripciones, un desertor es el estudiante que en el tiempo (t=0), está matriculado en un programa dentro de una institución determinada, pero en los tres (3) momentos siguientes del tiempo (t=1, t=2 *y* t=3), no se encuentra matriculado en ese mismo programa, o en otro programa, dentro de la misma

[45] http://www.unlam.edu.ar/descargas/32_planingenierainformtica2009.pdf
[46] http://www.unlam.edu.ar/descargas/397_planingenieracivil.pdf
[47] http://www.unlam.edu.ar/descargas/33_planingenieraelectrnica2009.pdf
[48] http://www.unlam.edu.ar/descargas/34_planingenieraindustrial2009.pdf

institución. Por lo tanto, los cambios de programa al interior de una misma institución no son considerados como deserción, sino como movilidad intra-institucional. Igualmente, las interrupciones temporales durante un período no son consideradas deserción.

Para la selección de asignaturas, se tomó como guía el trabajo realizado por Diego J. Edwards Molina [62], y luego por el mismo autor, y quién suscribe, en un documento presentado en el *IPECyT* 2016 [63], donde se eligieron, como variables descriptoras de la situación académica (Tabla 1), a las correspondientes asignaturas comunes del primer año de las carreras mencionadas.

Las asignaturas se seleccionaron entre las del ciclo general de materias básicas, y son comunes a todas las carreras. La nómina de asignaturas son Álgebra y Geometría Analítica I y II; Análisis Matemático I y II, Computación Transversal Nivel I, Elementos de Programación I, Matemática Discreta, Química General, Tecnología, Ingeniería y Sociedad, y Sistemas de Representación. A las variables utilizadas, respecto a las materias que cursó o no cursó, y su correspondiente situación académica, se le sumaron otras variables pertinentes para el análisis, como la edad, el estado civil y el género. Además, todas las variables correspondientes a la situación académica, de tipo numérica, se transformaron a variables discretas, con el objetivo de proporcionar una visión más comprensible de la información. Por ejemplo, los valores numéricos de las notas obtenidas por los estudiantes en cada asignatura, fueron transformadas en las siguientes categorías (Tabla 4):

Variable	Tipo de datos	Tipo de contenido	Valores
Cod_sexo	**Numérico**	**Discreta**	**1-2**
Estcivil	**Numérico**	**Discreta**	**1-2-3**
Carrera	**Numérico**	**Discreta**	**201-202-203-207**
Edad	**Numérico**	**Continuo**	**17 a 65**
Alg_Geo_Analitica_I	**Texto**	**Discreta**	**Cursada Promocionada**
Alg_Geo_Analitica_II	**Texto**	**Discreta**	**Final Aprobado**
Analisis_Mat_I	**Texto**	**Discreta**	
Analisis_Mat_II	**Texto**	**Discreta**	**Cursada Aprobada**
Computacion_Niv_I	**Texto**	**Discreta**	**Final Reprobado**
Elementos_Prog_I	**Texto**	**Discreta**	**Final Ausente**
Mat_Discreta	**Texto**	**Discreta**	

Química_Gral	**Texto**	**Discreta**	**Cursada Reprobada**
Tecn_Ing	**Texto**	**Discreta**	**Cursada Ausente**
Sist_Rep	**Texto**	**Discreta**	**No Cursada**
Situacion	**Texto**	**Discreta**	**Abandono** **No_Abandono**

Tabla 4. Nómina de variables utilizadas y sus valores.

Desde el punto de vista de la recolección de la información primaria, este caso de estudio, se clasifica como una investigación descriptiva. La variable dependiente o de respuesta, fue la variable dicotómica[49] denominada "***Situación***" que corresponde al estado del alumno, respecto a su condición de regularidad como cursante. Esta variable puede tener dos posibles valores: "***Abandonó****" o "****No_Abandonó****"*. Entonces, el valor "***Abandonó***" para este trabajo, se entiende como la no inscripción en ninguna de las asignaturas de la carrera en las que se ha inscripto el estudiante, dejando de asistir a las clases, y de cumplir con las obligaciones establecidas. Se escribió el código necesario para construir este atributo ya que el cubo no contenía esa información explícitamente, pero sí permite obtenerla.

Categoría	Significado
Cursada Promocionada	Cuando lo aprobó o promocionó la materia
Final Aprobado	Cuando aprobó el examen final
Cursada Aprobada	Si resta aprobar el examen final
Final Reprobado	Cuando no aprobó el examen
Final Ausente	Cuando se anotó y no rindió el examen
Cursada Reprobada	Cuando no aprobó la cursa
Cursada Ausente	Cuando se anotó y luego abandonó la materia
No Cursada	Cuando no se inscribió a la materia

Tabla 5. Posible valores que contenga la asignatura.

4.4. Minería de Datos

La etapa de la minería de datos, ha dado lugar a una paulatina sustitución del análisis de *datos dirigido a la verificación*, por un enfoque orientado hacia el *descubrimiento del*

[49] Las variables cualitativas pueden ser **dicotómicas** cuando sólo pueden tomar dos valores posibles.

conocimiento. La principal diferencia entre ambos se encuentra en que en el último, se descubre información sin necesidad de formular previamente una hipótesis. La aplicación automatizada de algoritmos de minería de datos permite detectar fácilmente patrones en los datos, razón por la cual esta técnica es mucho más eficiente que el análisis dirigido a la verificación, cuando se intenta explorar datos procedentes de repositorios de gran tamaño. y complejidad elevada. Dichas técnicas emergentes se encuentran en continua evolución. siendo el resultado de la colaboración entre campos de investigación, tales como bases de datos, reconocimiento de patrones, inteligencia artificial, sistemas expertos, estadística, visualización, recuperación de información, y computación de altas prestaciones [64].

La herramienta WEKA fue elegida para desarrollar esta etapa. Es una aplicación elaborada con Java, que contiene una colección de algoritmos de "Machine Learning" utilizados principalmente para resolver problemas de MD. Cabe aclarar que este capítulo tiene un enfoque práctico y funcional, pretendiendo servir de guía de utilización de esta herramienta, desde su interfaz gráfica. Se profundizará en los detalles técnicos, y específicos, sólo de los diferentes algoritmos propuestos. Cabe destacar, entonces, que se centrará en la aplicación, configuración, y análisis de los mismos, dentro de la herramienta, y sólo para aquellos parámetros que se consideraron propicios para la experimentación. Además, se comentarán, sin entrar en detalles, las características, y parámetros de configuración restantes, no usados en las experimentaciones. A las técnicas se le aplicarán ejemplos concretos, explicando los pasos necesarios para realizar las pruebas desde la herramienta, es decir, se presentará a modo ilustrativo cada corrida (ciclo), para reforzar el carácter práctico de este capítulo, el cual se verá como un formato de tipo tutorial.

4.5. Introducción a WEKA

Como ya se mencionó, para realizar las pruebas en la etapa siguiente de este proceso, se eligió el WEKA. En este trabajo se empleó la versión 3.6.13, que requiere una versión de Java 1.8 o superior. En la Figura 3 se muestra la pantalla de inicio del programa. WEKA está constituido por una serie de paquetes de código abierto con diferentes técnicas de pre-procesado, clasificación, agrupamiento, asociación y visualización. Se trata de un software desarrollado en la Universidad de Waikato (Nueva Zelanda) bajo licencia GNU-GPL[50], lo cual ha impulsado que sea una de las suites más utilizadas en el área en los últimos años. Este es uno de los programas recomendados por Hernández Orallo, J., en su libro "*Introducción a la minería de datos"* [5].

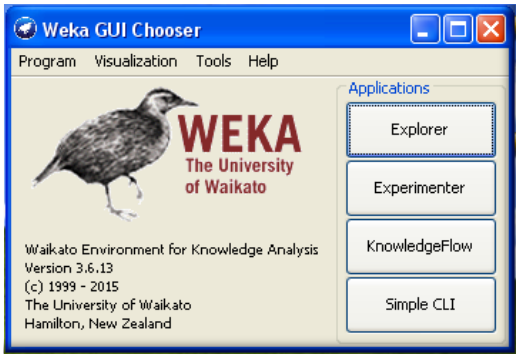


Figura 35. Ventana principal del programa WEKA.

La primera pantalla de WEKA muestra una serie de opciones en su parte superior, como así también, en el lateral derecho se encuentran las Aplicaciones (*Aplications*). Esta son un conjunto de sub-herramientas de WEKA. Para esa memoria la opción que se desarrollarán los algoritmos *Explorer*, para explorar los datos.

[50] La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License (o por sus siglas en inglés GNU GPL) es la licencia de derecho de autor más ampliamente usada en el mundo del software libre y código abierto,6 y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software.

4.5.1. Pestaña Pre-procesado (Preprocess) de los datos

Esta es la primera parte por la que se debe pasar, antes de realizar ninguna otra operación, ya que se precisan datos para poder llevar a cabo cualquier análisis. La disposición de la parte de pre-procesado del Explorer, Preprocess, es la que se indica en la Figura 37.

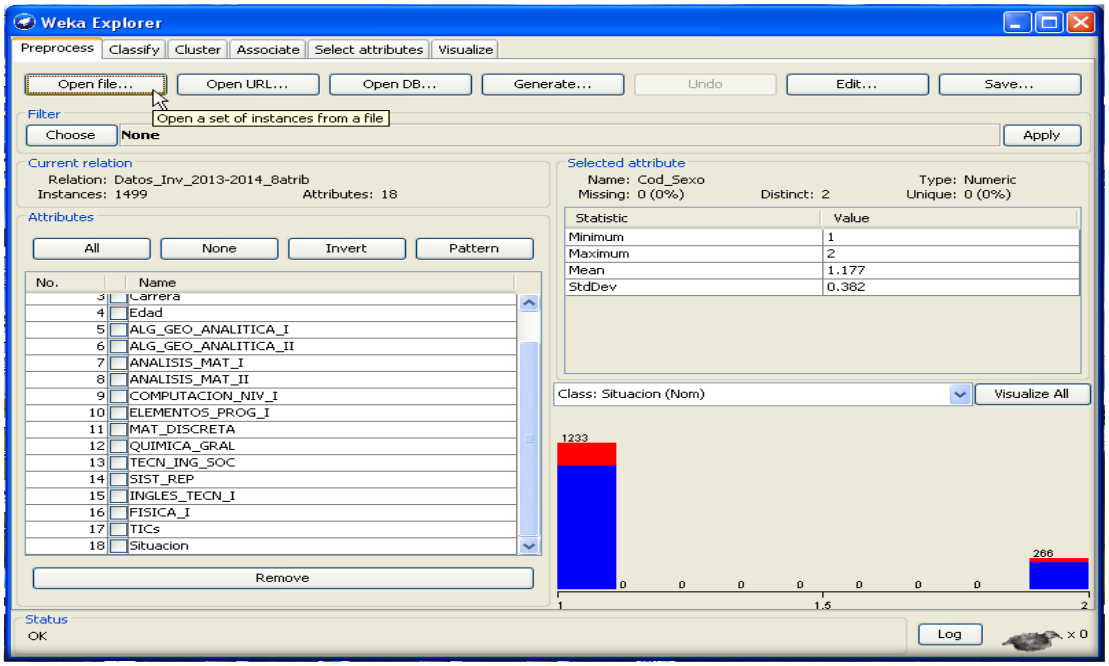


Figura 36. Pantalla principal del Explorador de WEKA

El Explorer permite visualizar, y aplicar, distintos algoritmos de aprendizaje para un conjunto de datos. Cada una de las tareas de minería de datos viene representada por una pestaña en la parte superior. Estas son:

- *Preprocess:* visualización y pre-procesado de los datos (aplicación de filtros)
- *Classify*: Aplicación de algoritmos de clasificación y regresión
- *Cluster*: Agrupación
- *Associate*: Asociación
- *Select Attributes*: Selección de atributos
- *Visualize:* Visualización de los datos por parejas de atributos

Es aquí donde se cargan los datos. Se usó un archivo tipo *arff*[51]., ya que este es el formato de datos de WEKA. El objetivo de este punto es construir un clasificador que permita clasificar a los alumnos en dos clases, Abandonó y No Abandonó, en términos de sus datos de rendimiento académico, y datos socio-económicos. Se cargó el fichero mediante la opción *Open File* (arriba a la izquierda). Si bien existen otras opciones para cargar datos desde una página web, desde una base de datos, o incluso para generar datos artificiales adaptados a clasificadores concretos, no fueron utilizadas.

4.5.1.1. Archivos ARFF

Los datos que se extraen del almacén de datos, y que se usarán en la comparación en este trabajo, deben introducirse mediante un archivo de texto tipo ASCII[52] en formato arff. WEKA admite varios tipos de archivos, pero el formato propio de la aplicación, y con el que mejor se entiende, es el ".arff". En el Anexo I, se encuentran un compacto del archivo *Datos_2013-2014.arff*, encabezado y una porción de los datos, usados en esta memoria. En este formato de archivo, las instancias de datos correspondientes a los registros de una base de datos, son independientes entre sí, no implicando el orden del listado de las mismas ningún orden, ni relación entre ellas necesariamente [34]. Generalmente hay un atributo que será tomado como la clase a clasificar. Este puede ser el último atributo de la serie, o bien tener el nombre Clase (*Class).* El formato arff permite dos tipos de datos básicos, el tipo Nominal

[51] Por sus siglas en ingles Attribute-Relation File Format.

[52] El código ASCII sigla en inglés de American Standard Code for Information Interchange (Código Estadounidense Estándar para el Intercambio de Información).

(datos categóricos), y el tipo Numeric (datos numéricos). La estructura del mismo es simple, consistente en un listado de cadenas de caracteres, y puede dividirse en tres partes:

- **Cabecera**; Indicada por una única cadena inicial de tipo:

  @relation <nombre_de_la_relacion>

- **Declaración de atributos o variables**. A través de:

  @attribute <nombre-variable>

  siendo el valor de tipo: string, numeric, integer, date o nominal.

- **Sección de datos**. Definidos de la siguiente forma:

  @data

Donde se tendrá una línea para cada registro, los valores estarán separados por coma, y los valores perdidos se representan mediante el carácter "?". Además, es posible escribir comentarios en ese fichero, precedidos del caracter "%".

4.5.1.2. Construcción de los modelos

Con los datos ya presentados en el formato definido por el software, se obtiene la primera Vista Minable (VM), que es la consolidación en una única tabla de todas las observaciones (datos), y los atributos sobre los que se aplicarán los algoritmos de minería de datos. Entonces el modelado de una VM se refiere a un conjunto de datos que, según su relevancia, son los más importantes dentro del repositorio de datos, y que los agrupan de una forma en que se puedan procesar con más facilidad. A partir de esta VM, el siguiente paso consiste en la construcción de los modelos de minería de datos, usando una RNA y una MVS, con el objeto de tener un parámetro de comparación, respecto a qué modelo presenta una mejor respuesta frente a la estimación de la variable *Situación* (el atributo clase), esto se debe a que para poder aplicar los algoritmos de minería, se necesita que la vista minable tenga un atributo que exprese a qué categoría pertenece, y así poder entrenar al algoritmo. De esta sección es que se puede seleccionar el archivo de datos con formato arff, que incluirá los ejemplos de entrenamiento y testeo.

WEKA permite ver cómo están distribuidos los distintos atributos del conjunto de entrenamiento, así como editarlo, eliminando atributos [34]. Una vez cargados los datos, aparece un cuadro resumen, "*Current relation*", con el nombre de la relación que se indica en la línea @relation del fichero arff), el número de instancias, y el número de atributos.

Luego aparecen listados todos los atributos disponibles, incluyendo los nombres especificados en el fichero, de modo que se pueden seleccionar para ver sus detalles y propiedades.

En la parte derecha, aparecen las propiedades del atributo seleccionado. Si es un atributo simbólico, se presenta la distribución de valores de ese atributo, número de instancias que tienen cada uno de los valores. Si es numérico aparecen los valores máximo, mínimo, valor medio, y desviación estándar. Otras características que se destacan del atributo seleccionado, son el tipo (*Type*), número de valores distintos (*Distinct*), número y porcentaje de instancias con valor desconocido para el atributo (*Missing*, codificado en el fichero arff con "?"), y valores de atributo que solamente se dan en una instancia (*Unique*). Además, en la parte inferior se presenta gráficamente el histograma con los valores que toma el atributo. Si éste es simbólico, se visualiza la distribución de frecuencia de los valores, si es numérico se muestra un histograma con intervalos uniformes. En el histograma se puede presentar además con colores distintos la distribución de un segundo atributo para cada valor del atributo en pantalla. Por último, hay un botón que permite observar los histogramas de todos los atributos simultáneamente. En la Figura 38, se puede ver la distribución de los datos con su correspondiente clase, respecto a cada variable,

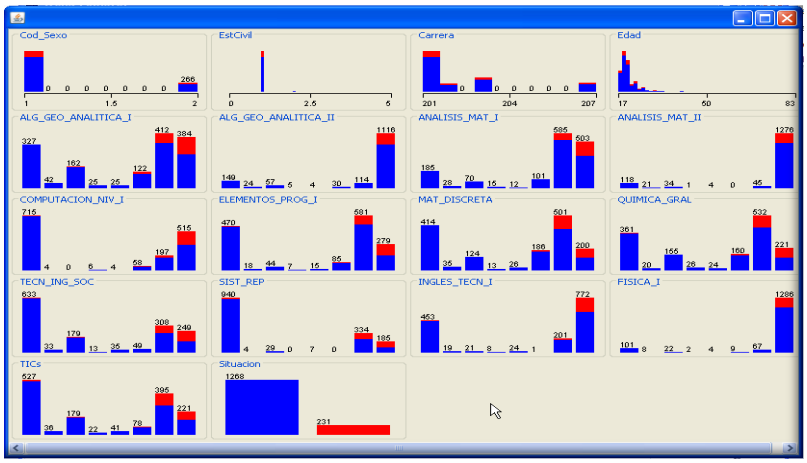


Figura 37. Pantalla principal del Explorador de WEKA

En esta ventana también es posible seleccionar filtros, para los datos y los atributos (selección de filtros en la parte superior izquierda), además de eliminar algunos atributos. Estos filtros disponibles no serán tenidos en cuenta como opción. Para la reducción de atributos, eliminación de atributos redundantes e irrelevantes, se utilizará la pestaña de selección de atributos.

4.5.2. Pestaña Selección de Atributos (Select Attributes)

Los algoritmos de selección de características tienen dos objetivos principales:

- Reducir el coste computacional asociado tanto al aprendizaje como al propio modelo de conocimiento generado (eliminando atributos irrelevantes o redundantes)
- Aumentar la precisión de dicho modelo (eliminando atributos perjudiciales para el aprendizaje).

Si existe un número excesivo de atributos, puede que el modelo sea demasiado complejo, y se produzca *sobreajuste (*en inglés *Overfitting)* [34]. En MD, es frecuente emplear este término, que es el efecto de sobreentrenar un algoritmo de aprendizaje con unos ciertos datos para los que se conoce el resultado deseado. En WEKA, la selección de atributos puede hacerse de varias maneras. En la Figura 39 se observa la pestaña Selección de atributos *(Attribute Selection).*

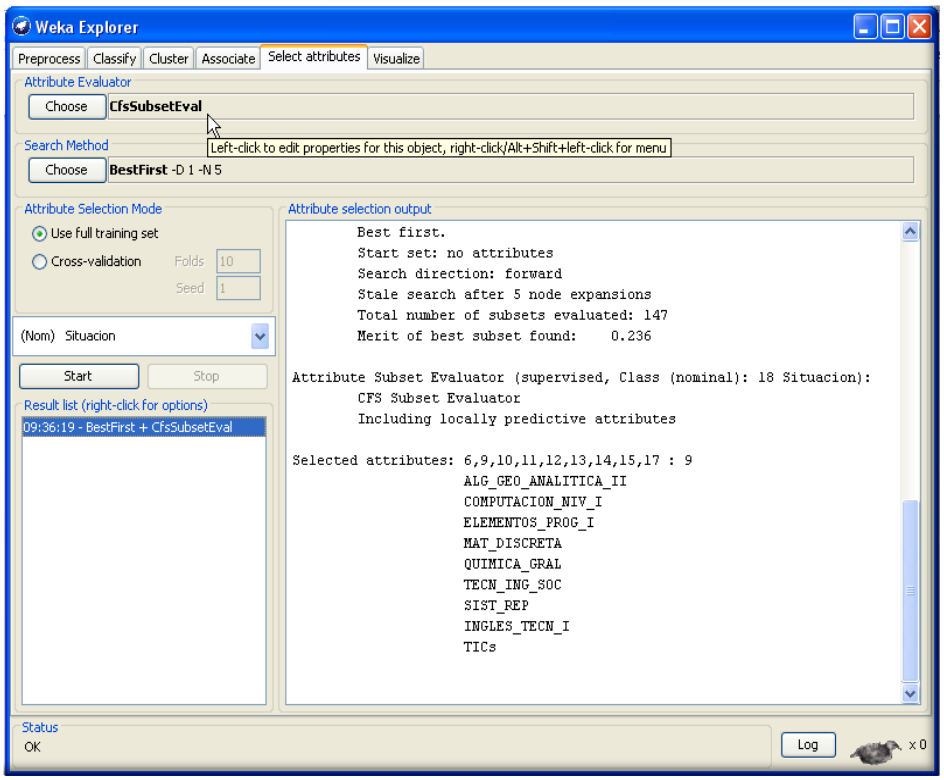


Figura 38. Pantalla principal del filtrado de atributos de WEKA

En esta pestaña se trata que WEKA ayude a determinar qué atributos formarán parte del modelo; es decir, poder eliminar aquellos atributos que resulten, como se mencionó anteriormente, redundantes e irrelevantes para realizar la clasificación. Además, si hay un número excesivo de atributos, puede implicar obtener un modelo demasiado complejo, y se produzca sobreajuste. En WEKA, la selección de atributos se puede hacer de varias maneras, siendo la más directa la que se realiza a través de esta pestaña. En ella tenemos que seleccionar:

- El método de evaluación (***Attribute Evaluator***): es la función que determina la calidad del conjunto de atributos para discriminar la clase. Se pueden distinguir entre los métodos que directamente utilizan un clasificador específico, para medir la calidad del subconjunto de atributos mediante la tasa de error del clasificador, y los que no.
- El método de búsqueda (***Search Method***): es la manera de realizar la búsqueda de conjuntos de forma eficiente Dado que la evaluación exhaustiva de todos los subconjuntos es un problema combinatorio inabordable, conforme crece el número de atributos, aparecen estrategias que permiten realizar la búsqueda de forma eficiente.

Para esta tesis, se expone en forma sintética este tema, aunque tanto en Witten [34], como García Gutiérrez [65], lo desarrollan en más detalle.

Hay dos tipos de evaluadores principales, los evaluadores de subconjuntos o selectores (*SubsetEval*), y Prorreteadores de atributos (*AtributteEval*). Los primeros tienen que elegir una estrategia, o método de búsqueda de subconjuntos (*Search Method*), para encontrar el mejor subconjunto de atributos que proporcione los mejores resultados en un modelo de clasificación. Significa que se obtendrán aquellos atributos con los cuales se ha realizado una mejor clasificación. El segundo tipo de evaluadores sólo pueden combinarse con un *Ranker*, ya que no seleccionan los atributos, sino que los ordenan por su relevancia a la hora de clasificar el atributo objetivo. En general, estos algoritmos pueden ser clasificados por varios criterios.

Una categorización popular es aquella en la que los algoritmos se distinguen por su forma de evaluar atributos, y se clasifican en, *Filtros*, donde se seleccionan y evalúan los atributos en forma independiente del algoritmo de aprendizaje, y *Wrappers* (envoltorios) , los cuales usan el desempeño de algún clasificador, algoritmo de aprendizaje, para determinar lo deseable de un subconjunto. Se los denominan métodos *Wrappers*, porque "envuelven" al clasificador para explorar la mejor selección de atributos que optimiza sus prestaciones, son muy costosos porque necesitan un proceso completo de entrenamiento y evaluación, en cada paso de búsqueda.

4.5.2.1. Filtrado y limpieza de datos.

Para empezar un método de selección de atributos, es necesario primero seleccionar el método de evaluación de atributos (*Attribute evaluator*), de la Figura 39. Este método será el encargado de evaluar cada uno de los casos a los que se le enfrente, y dotar a cada atributo de un peso específico. El funcionamiento para seleccionar este método es el mismo que con otros métodos en WEKA, se selecciona el método con el botón *Choose* situado dentro del cuadro *Attribute Evaluator*. Una vez seleccionado podemos acceder a las propiedades del mismo, pulsando sobre el nombre de la etiqueta que muestra el nombre del método seleccionado. El siguiente paso será elegir el método de búsqueda que será el encargado de

generar el espacio de pruebas. El funcionamiento es el mismo al caso anterior. Una vez seleccionado el método de evaluación, y el de generación del espacio de pruebas, sólo falta elegir el método de prueba, el atributo que representa la clasificación conocida, y pulsar Start [38].

Para esta selección se siguió el procedimiento empleado en la investigación del Proyecto PROINCE C176, de la UNLaM [66]. Se utilizaron los cuatro (4) algoritmos evaluadores de subconjuntos de atributos disponibles en el WEKA, los dos primeros clasificados como Filtros, y los restantes como *Wrappers*. Se ejecutaron en combinación con el método de búsqueda Best First, el cual busca en el espacio de los subconjuntos de atributos utilizando la estrategia *Greedy Hillclimbing*, con *Backtracking*. La dirección de la búsqueda, realizada por *BestFirst*, fue hacia adelante partiendo del conjunto vacío de atributos. En la Tabla 6 se pueden observar los atributos que arrojaron cada una de las pruebas realizadas.

• Evaluator: WEKA.attributeSelection.CfsSubsetEval • Search:WEKA.attributeSelection.BestFirst -D 1 -N 5	Attribute Subset Evaluator (supervised, Class (nominal): 18 Situacion): CFS Subset Evaluator Including locally predictive attributes. Selected attributes: 6,9,10,11,12,13,14,15,17 : 9 ALG_GEO_ANALITICA_II - COMPUTACION_NIV_I ELEMENTOS_PROG_I - MAT_DISCRETA, QUIMICA_GRAL - TECN_ING_SOC, SIST_REP - INGLES_TECN_I - TICs
• Evaluator: WEKA.attributeSelection.ConsistencySubsetEval • Search:WEKA.attributeSelection.BestFirst -D 1 -N 5	Attribute Subset Evaluator (supervised, Class (nominal): 18 Situacion): Consistency Subset Evaluator. Selected attributes: 4,5,7,9,10,11,12,13,14,15,17 : 11 Edad - ALG_GEO_ANALITICA _I - ANALISIS_MAT_I COMPUTACION_NIV_I - ELEMENTOS_PROG_I, MAT_DISCRETA - QUIMICA_GRAL, TECN_ING_SOC - SIST_REP, INGLES_TECN_I - TI
• Evaluator: WEKA.attributeSelection.ClassifierSubsetEval -B WEKA.classifiers.functions.SimpleLogistic -T -H "Click to set hold out or test instances" -- -I 0 -M 500 -H 50 -W 0.0 • Search:WEKA.attributeSelection.BestFirst -D 1 -N 5	Attribute Subset Evaluator (supervised, Class (nominal): 18 Situacion):Classifier Subset Evaluator Learning scheme: WEKA.classifiers.functions.SimpleLogistic Scheme options: -I 0 -M 500 -H 50 -W 0.0 Hold out/test set: Training data Accuracy estimation: classification error Selected attributes: 2,4,5,8,9,12,13,14,15,16,17 : 11 EstCivil, Edad , ALG_GEO_ANALITICA_I,ANALISIS_MAT_II - COMPUTACION_NIV_I,QUIMICA_GRAL - TECN_ING_SOC SIST_REP - INGLES_TECN_I - FISICA_I - TICs
• Evaluator: WEKA.attributeSelection. WrapperSubsetEval –B WEKA.classifiers.functions.SimpleLogistic -F 5 -T 0.01 -R 1 -- -I 0 -M 500 -H 50 -W 0.0 Search:WEKA.attributeSelection.BestFirst -D 1 -N 5 • Search Method: Best first. Start set: no attributes. Search direction: forward Stale search after 5 node expansions. Total number of subsets evaluated: 184	Attribute Subset Evaluator (supervised, Class (nominal): 18 Situacion): WrapperSubset Evaluator earning scheme: WEKA.classifiers.functions.SimpleLogistic Scheme options: -I 0 -M 500 -H 50 -W 0.0 Subset evaluation: classification accuracy Number of folds for accuracy estimation: 5 Selected attributes: 2,4,8,9,10,12,13,14,15 : 9 EstCivil - Edad - ANALISIS_MAT_II, COMPUTACION_NIV_I - ELEMENTOS_PROG_I, QUIMICA_GRAL - TECN_ING_SOC SIST_REP - INGLES_TECN_I,

Tabla 6. Nómina de atributos por método de selección.

Una vez que se obtuvieron los distintos conjuntos de atributos (Tabla 6), se procedió a utilizarlos con cada uno de los algoritmos supervisados propuestos, para encontrar aquel con cuál se obtiene mejor porcentaje de casos correctamente clasificados. En esta comparación se incluyó además una prueba con todos los atributos originales. La opción *Percentage Split*, fue la elegida para probar los algoritmos en cada conjunto de atributos propuesto. En la Tabla 7, se pueden observar los porcentajes obtenidos, siendo el *WapperSubsetEva* el que mejor porcentaje obtuvo.

Método	RNA	MVS
Porcentaje de instancias correctamente clasificadas		
Todos los atributos	**89.8599 %**	**91.7945 %**
CfsSubsetEval:	**89.9933 %**	**91.7278 %**
ConsistencySubsetEval	**89.7932 %**	**91.7945 %**
ClassifierSubsetEval	**88.9927 %**	**91.5944 %**
WapperSubsetEva	**96.7312 %**	**92.1948 %**

Tabla 7. Porcentajes de Instancias correctamente clasificadas según algoritmo.

Es de destacar de este tabla, que el uso de la totalidad de los atributos no significó ninguna mejora en la calidad de la clasificación, lo que muestra la conveniencia de recurrir a una selección de atributos por el ahorro de tiempo que habrá en la clasificación, sin pérdida de calidad en la misma. Cabe aclarar además, que para este ensayo se usaron los 4 clasificadores, adoptando en sus parámetros propios, los valores sugeridos por WEKA. Tampoco, como se puede observar, hubo ningún método que haya tenido gran diferencia en cuanto a los resultados obtenidos. Se optó entonces, por el método *WapperSubsetEva*.

Determinados algoritmos, y métodos del aprendizaje computacional, permiten obtener buenos resultados de predicción una vez entrenado el modelo, con un número conocido de datos, aplicando alguna técnica de presentación de los mismos, esperando que el modelo produzca una respuesta, sino idéntica, sí cercana a la correcta. Por tal motivo de entre estos modelos predictivos, las Redes Neuronales Artificiales (RNA) y la Máquinas de Vector Soporte (MVS). Vamos ahora a construir un primer modelo para los datos propuesto.

4.5.3. Pestaña Clasificar (Classify)

Pulsando en la segunda pestaña del explorador (parte superior izquierda de la ventana Explorer), se encuentra el modo de clasificación, en la siguiente Figura se puede ver la ventana, ya con un clasificador elegido.

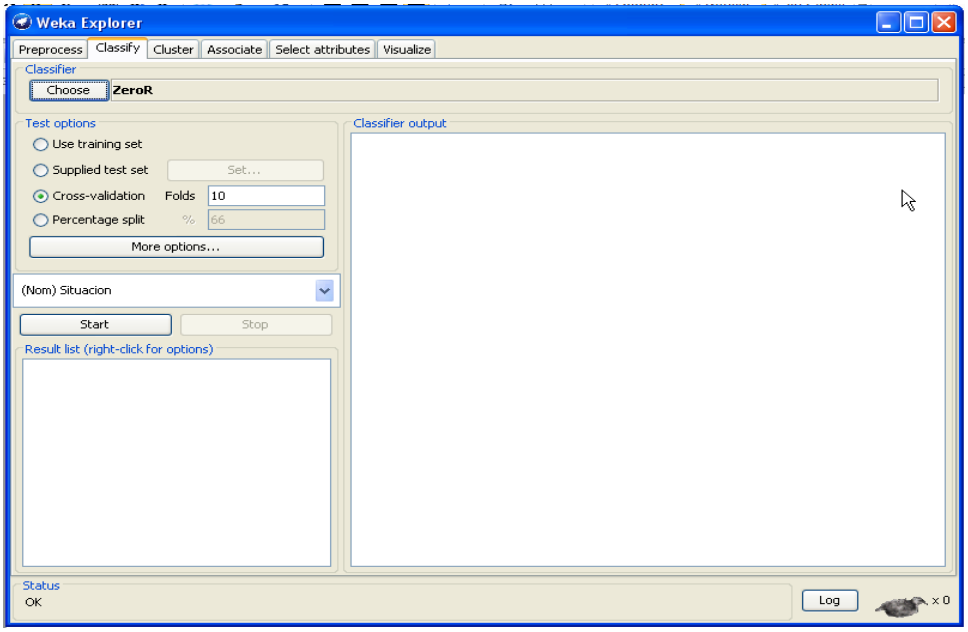


Figura 39. Pestaña Classify de WEKA

Primero se deberá elegir el clasificador en *Choose*, en la Figura 40 se ve preseleccionado el clasificador Zero, se abre entonces la ventana que permite elegir de una serie de clasificadores, como se muestra en la Figura 41.

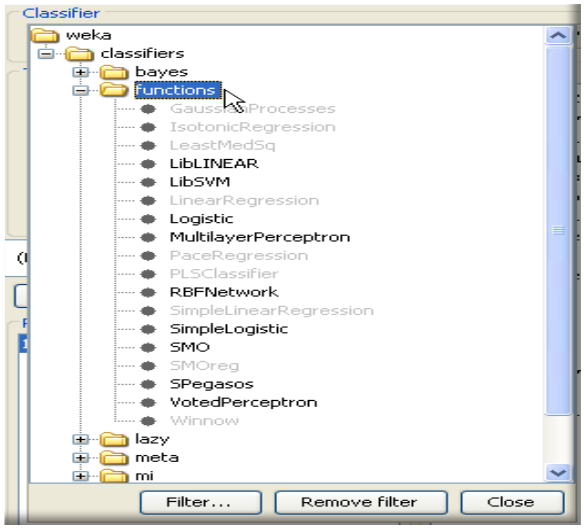


Figura 40. Menú de selección de algoritmos

Una vez seleccionado el algoritmo, WEKA ofrece una cantidad de opciones con las cuales se podrán incluir, y hasta modificar, parámetros asociados al clasificador elegido en referencia al lote de prueba que usará el algoritmo. Esto se encuentra en el *Test Options*, en la Figura 7 se pueden ver las siguientes opciones disponibles:

- *Use training set*: En esta opción se entrenará el método con todos los datos disponibles y luego se aplicará sobre los mismos.
- *Supplied test set*: Marcando esta opción se tendrá la oportunidad de seleccionar un fichero de datos, con el que se probará el clasificador obtenido a partir del método de clasificación usado, y los datos iniciales.
- *Cross validation*: La herramienta realizará una validación cruzada, estratificada del número de particiones dado. En primer lugar, se divide el conjunto de entrenamiento en subconjuntos disjuntos, llamados "pliegues". El número de subconjuntos se pueden introducir en el campo *Folds*, diez (10) es el valor predeterminado, y en general proporciona mejores estimaciones que otras opciones. Una vez que los datos se han dividido en los pliegues de, aproximadamente, igual tamaño, todos menos uno de los pliegues se utilizan para la formación, y el restante se utiliza para la prueba.
- *Percentage split*: Se define un porcentaje de los datos, con el que se construirá el clasificador, y con la parte restante se realizarán las pruebas. Es decir, los datos son elegidos al azar, y luego se dividen en un conjunto de entrenamiento, y otrode pruebas, de acuerdo a la proporción especificada. En la práctica, esta es una buena alternativa a la validación cruzada, si el tamaño del conjunto de datos haga a la validación cruzada demasiado lenta. [38]

Se generaron dos modelos distintos para el conjunto de datos disponible (1499 registros y 9 atributos, más el atributo Clase), resultado de aplicar el evaluador de WEKA *WapperSubsetEva.* Estos modelos son los que se usaran para realizar la comparación en esta memoria.

4.6. Modelo de clasificación basado en RNA

Para elegir una Red Neuronal Artificial, debemos elegir los algoritmos basados en funciones. Pinchamos en *functions* y escogemos el *Perceptrón Multi-capa*. Una vez hecho esto, como se muestra la Figura 42, en el panel de texto que está dentro del panel *Classifier* veremos que aparece la leyenda:

MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

y si pinchamos ahí aparecerá una ventana que permite configurar el algoritmo, con parámetros como el *ratio de aprendizaje*, el *momentum*, el tiempo destinado para aprender, etc. .

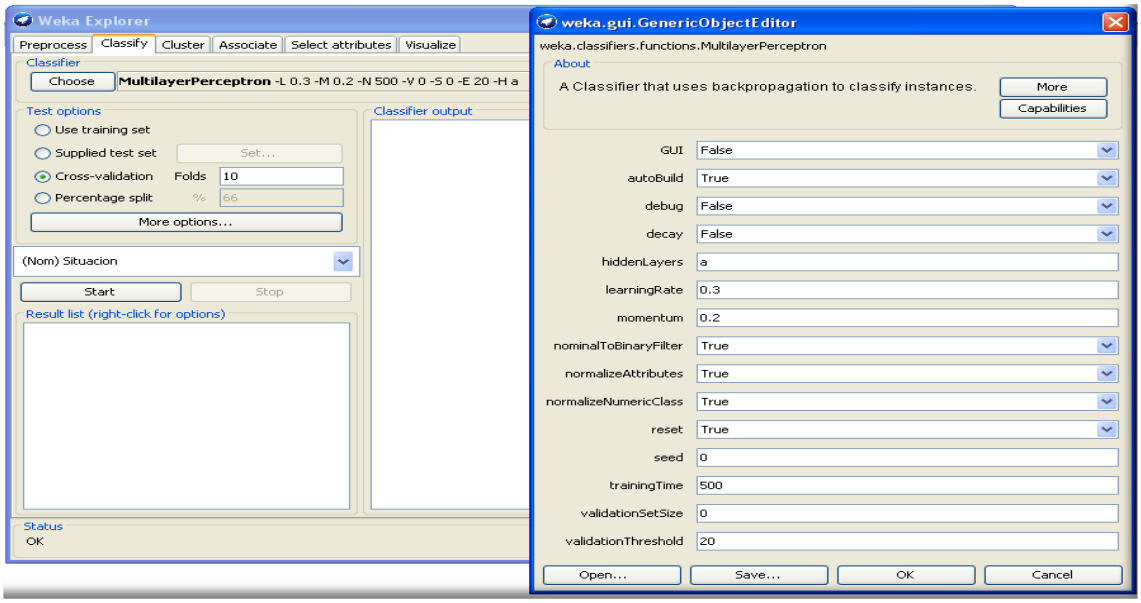


Figura 41. Menú de selección de algoritmos y configuración de una RNA.

Siendo las opciones configurables las siguientes:

- *GUI*: muestra una interfaz GUI. Esto permitirá la pausa y la alteración de la red neuronal durante el entrenamiento.
- *Autobuild*: añade y conecta las capas ocultas de la red.
- *Debug*: se mostrará información adicional en la pantalla de salida.
- *Decay*: hace que la tasa de aprendizaje disminuya. Esto puede ayudar a mejorar el rendimiento general.
- *HiddenLayers*: define las capas ocultas de la red neuronal. Solo puede usarse si está activo autoBuild.
- *LearningRate*: actualiza los pesos.
- *Momentum:* añade cierto impulso a los pesos durante la actualización.

- *NominalToBinaryFilter:* aplica el filtro de pre-procesamiento que convierte los atributos nominales a atributos nominales de dos valores (binarios).
- *NormalizeNumericClass*: normaliza las clases numéricas.
- *Reset*: permite a la red restablecer la tasa de aprendizaje. Esta opción solo está disponible si no está activa GUI.
- *Seed*: inicializa el generador de números aleatorios, que sirven para ajustar los pesos iniciales de las conexiones entre nodos.
- *TrainingTime*: número de veces que se entrenará el método.
- *ValidationSetSize*: define un conjunto de validación de tamaño especificado. El entrenamiento continuará hasta que se observe que el error en el conjunto de validación empeora de forma constante.
- *ValidationTreshold*: realiza un test de validación. Este valor determina cuántas veces el conjunto de validación empeora en la misma línea antes de que el entrenamiento acabe.

La técnica estándar para entrenamiento de redes feed-forward es Backpropagation, y de hecho es la que implementa WEKA en su clasificador MultilayerPerceptron [32]. En la literatura se han propuesto distintas estrategias de entrenamiento para obtener bajos errores de clasificación. La arquitectura del perceptrón multicapa se compone de, número de neuronas en la capa de entrada, el cual depende del número de componentes del vector de entrada; cantidad de capas ocultas y número de neuronas de cada una de ellas; y el número de neuronas en la capa de la salida, el cual depende del número de componentes del vector de salida o patrón objetivo. Teniendo en cuenta que tanto el vector de entrada, como el de salida, están definidos por el problema a resolver, nueve (9) y dos (2) respectivamente, en la Figura 34 se puede ver gráficamente.

Respecto al criterio a tener en cuenta para la selección de las neuronas de las capas ocultas, surge una interrogante, este número en general debe ser lo suficientemente grande como para que se forme una región compleja que pueda resolver el problema, sin embargo no debe ser muy grande, pues la estimación de los pesos puede ser no confiable para el conjunto de los patrones de entrada disponibles. La mayor eficiencia del perceptrón multicapa, en cuanto a su arquitectura, se logra principalmente con una mayor experiencia de su diseñador, aunque también existen varios criterios para realizar esta selección. A partir de una serie de criterios consultados en la bibliografía, para realizar esta serie de

experimentos, respecto a la cantidad de capas ocultas, se siguió el siguiente criterio de configuraciones para elegir cual es el mejor evalúa:

- Una RNA sin capa oculta.
- Una sola capa de neuronas ocultas, ya que de esta manera se pueden representar todas las funciones matemáticas conocidas hasta el momento. Para la cantidad de neuronas se empleó la regla de la pirámide geométrica, explicada en el capítulo anterior, la fórmula es: $\sqrt{N * M}$ **,** dando como resultado 5 neuronas.
- Una sola capa de neuronas ocultas, para la cantidad de neuronas ahora usamos la regla de la capa oculta-capa entrada, también explicada en el capítulo III, en este caso se aplica la regla **2 x 1**, de forma que el número de neuronas ocultas no puede ser superior al doble del número de variables de entrada, dando como resultado 18 neuronas.
- Por último se probó con dos capas de neuronas ocultas, empleado la cantidad de neuronas de los experimentos anteriores. 5 y 18 respectivamente.

Una modificación simple, pero eficiente, consiste en utilizar una tasa de aprendizaje η alta para ejemplos incorrectamente clasificados, y una tasa baja para los ejemplos correctamente clasificados. Inicialmente se fijaron algunos valores necesarios en el entrenamiento de la red, como es el caso del *momentum* con 0.2, y la *velocidad de aprendizaje* con 0.3, estos valores son los propuestos por WEKA, los mismos fueron reemplazos, ya que se vio que los mejores valores se situaron en 0.4 para cada uno de los parámetros. Entonces, entre los parámetros que se configuraron para este trabajo, se encuentran:

- Número de capas ocultas (*hidden layers*): se ofrecen varias posibilidades para aproximar correctamente el modelo buscado.
- Tasa de aprendizaje (*learning rates*): factor de actualización de pesos cuando han de ser modificados, un valor mayor converge más rápido pero puede provocar oscilaciones.
- *(Momentum)*: para determinar cuándo dar por acabado el test de validación, cuántas veces en una fila el error puede empeorar antes de terminar el entrenamiento.

Luego de seleccionados los valores con los que se van a entrenar, y de definir la arquitectura de la red, la *velocidad de aprendizaje* y el *momentum*, teniendo como cantidad de patrones de entrada los 9 atributos y los dos de salida, se realizó el entrenamiento. La aplicación genera aleatoriamente los pesos sinápticos. Cabe recordar que cada patrón de entrada se hace pasar a través de la estructura activando cada neurona, generando salidas en estas. Dichas salidas son multiplicadas por los pesos sinápticos, y constituyen la entrada de las neuronas de la capa siguiente. Así sucesivamente hasta llegar a la capa de salida, donde

el resultado final se compara con el resultado esperado, generando un error, el cual es propagado por toda la red (backpropagation), hasta llegar al origen, corrigiendo los valores sinápticos. Esto sucede con cada patrón, hasta que todos hayan pasado a través de la red, esto constituye una *iteración* o *epoch.* Este valor es 500, y también es el valor por default de WEKA. Para realizar los siguientes experimentos, se utilizó la vista minable compuesta por los 1499 registros de los alumnos de las carreras de Ingeniería del DIIT, de las cohortes 2013 y 2014. En cuanto a los diversos modos de realizar la evaluación, se decidió utilizar el método *Percentage split*, el cual divide los datos en dos grupos, de acuerdo con el porcentaje indicado, en este caso el 66 %. Este valor indica el porcentaje de instancias para construir el modelo, que seguidamente es evaluado sobre las que se han dejado aparte. Para correr cada configuración, en esta pestaña basta pulsar el botón *Start*, mostrado en la Figura 9, entonces WEKA comenzará el entrenamiento. Al cabo de un cierto tiempo, el algoritmo entrega el resultado. El mismo se muestra en ventana principal de la pantalla, de la pestaña *Classify* en el recuadro *Classifier Output.* Figura 43.

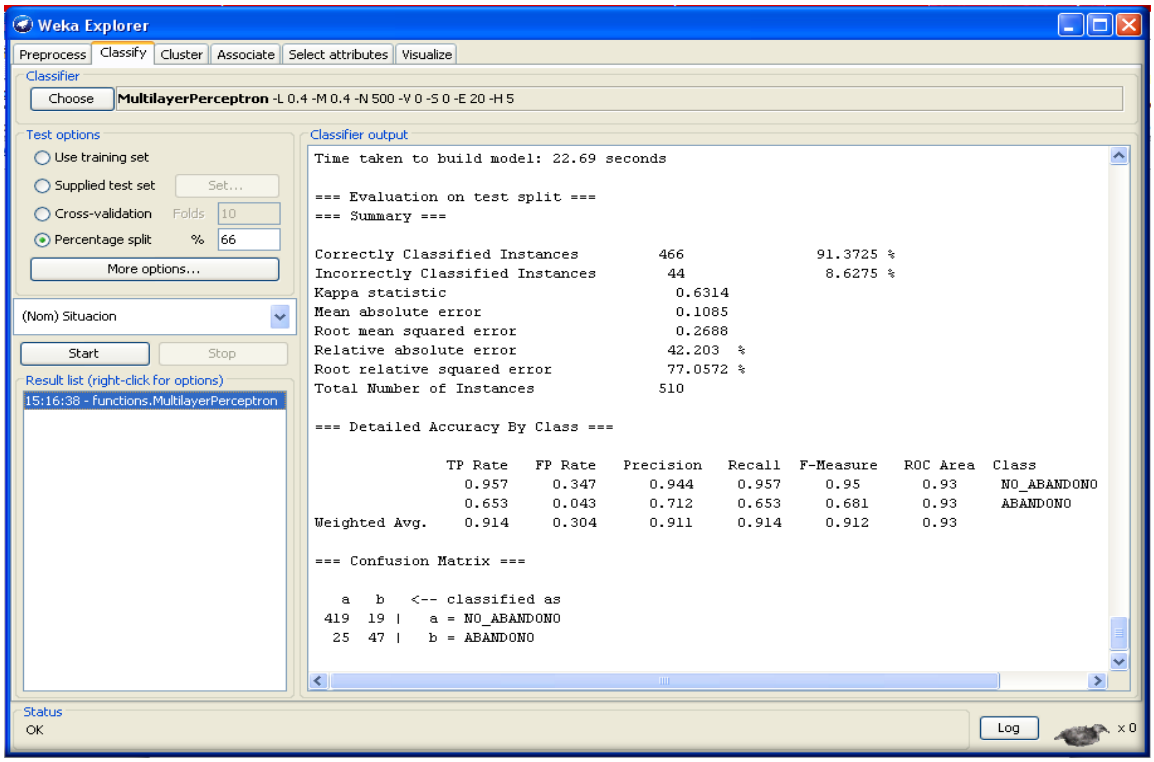


Figura 42: Resultados para una RNA con 5 neuronas en la capa oculta.

En la Tabla 6, se muestran los resultados arrojados por WEKA al comparar las distintas configuraciones.

Número de capas ocultas	Neuronas por Capas	*Momentum*	Tasa de aprendizaje	% bien Clasificados
Ninguna	**0**	***0.4***	**0.4**	**89.8039 %**
1	**5**	**0.4**	**0.4**	**91.3725 %**
1	**18**	**0.4**	**0.4**	**89.6078 %**
2	**5 -18**	**0.4**	**0.4**	**90.3922 %**

Tabla 8. Porcentajes de Instancias correctamente clasificadas según distintas configuraciones.

Como se puede observar, la configuración de la RNA con una capa, oculta con cinco (5) neuronas, fue la que mejor porcentaje de aciertos obtuvo. En la Figura 44, se muestra el diseño de las RNA que obtuvo el mejor porcentaje de clasificación. Podemos distinguir varios elementos en esta interfaz:

- Las etiquetas verdes a la izquierda son las entradas.
- Los nodos rojos son las capas ocultas.
- Los nodos naranjas representan los nodos de salida y las etiquetas naranjas de la derecha muestran el nodo que es.

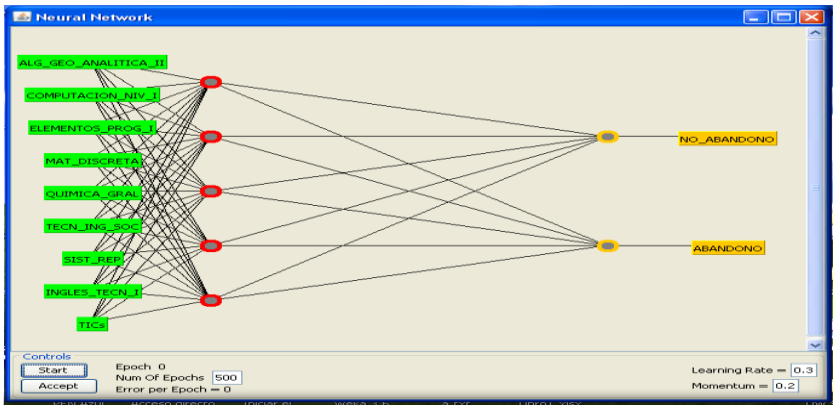


Figura 43: Salida gráfica de una RNA con 5 neuronas en la capa oculta.

Esta interfaz es interactiva, y podemos cambiarla según ciertas normas. Habrá que tener en cuenta, que las alteraciones a la red neuronal sólo se pueden hacer mientras la red no está funcionando, tanto tasa de aprendizaje como demás campos. Para añadir un nodo, bastará con pulsar el botón izquierdo del mouse, configurado para personas diestras,, mientras que para eliminar un nodo, se utilizará el derecho . Para seleccionar un nodo, basta con pulsar sobre él. Para seleccionar varios, es necesario mantener la tecla Control (*Ctrl)* pulsada mientas se selecciona con el mouse. Para deseleccionar todos, pulsar el botón derecho en cualquier parte de la pantalla desmarcará todos los que estén seleccionados, o bien para

desmarcar sólo uno, pulsar el botón derecho + *Ctrl* sobre aquel que se quiera desmarcar. Para conectar uno. o más nodos. a otro, primero se deben seleccionar los que serán conectados, y después pulsar sobre el nodo destino, o en un espacio vacío. Esto creará un nuevo nodo que estará conectado con los nodos seleccionados. El estado de selección de nodos será el mismo después de la conexión. Para desconectarlos, se selecciona uno de los nodos, y se pulsa el botón derecho sobre el otro nodo.

Hay que destacar, que no todos los modelos neuronales son aptos para todas las tareas de minería de datos. De manera general, cada método tiene características específicas por construcción, y se destacan en diversas áreas como el control, optimización, y visión, entre otras. Lo más probable, es que siempre la arquitectura de una red neuronal esté estrechamente relacionada con el algoritmo de aprendizaje de la misma.

4.6.1. Lista de resultados (Result list)

Una vez que finalizado el entrenamiento, se procede a guardar el modelo mejor clasificado, para ser utilizado en la comparación. Como se muestra en la Figura 45, con el botón izquierdo de mouse, y posicionándose sobre el entrenamiento que se desee guardar, se abre una ventana. En ella se encuentra una serie de opciones. Por ejemplo, *Save Model*, ésta nos permite guardar el modelo para ser utilizado con otros lotes de datos.

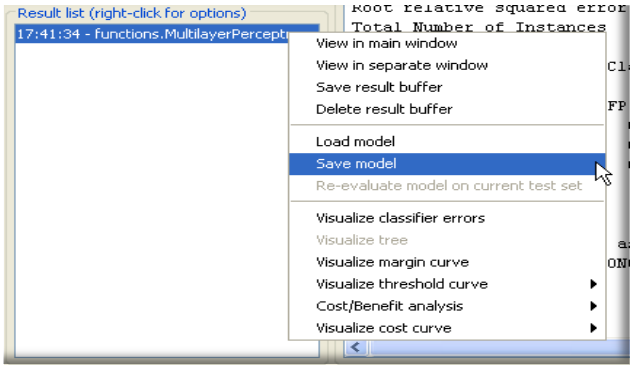


Figura 44: Salida gráfica de una RNA con 5 neuronas en la capa oculta.

Aquí va una breve descripción de cada una de las opciones:

- *View in main window:* mostrará la salida en la pantalla.
- *View in separate window*: abrirá una nueva ventana con los datos de los resultados.
- *Save result* buffer: salva el resultado
- *Delete result buffer*: borra el resultado de la lista.

- *Load model*: carga un modelo guardado anteriormente.
- *Save model*: guarda nuestro modelo para poder usarlo en clasificaciones posteriores.
- *Re-evaluate model on current test set*: si cambiamos el conjunto de datos del test, podremos reevaluar el modelo con esos nuevos datos.
- *Visualize classifier errors*: muestra una gráfica de errores.
- *Visualize graph / Visualize tree*: Visualiza árboles o gráficas.
- *Visualize margin curve*: curva sobre la diferencia entre la probabilidad de clase predicha y la probabilidad máxima de otras clases.
- *Visualize treshold* curve: gráfica de proporciones de variación de la clase escogida.
- *Cost/Benefit analysis*: permite predecir los costos y beneficios de manera analítica.
- *Visualize cost curve*: genera una gráfica sobre la probabilidad de las compensaciones sobre el costo. Al igual que el anterior, debemos tratar con datos con costos.

4.7. Modelo de clasificación basado en MVS

El aprendizaje de máquinas de vectores soporte (MVS), es un método supervisado que ha demostrado buenas propiedades [5]. Es una de las técnicas más poderosas del aprendizaje automático, que a pesar de su sencillez ha demostrado ser un algoritmo robusto, que generaliza bien en problemas de la vida real. En WEKA esta técnica figura como un algoritmo SMO (*Sequential Minimal Optimization*), que se ha utilizado para entrenar MVS usando un kernel polinomial, SMO implementa el algoritmo de entrenamiento mediante núcleos polinomiales, o gaussianos [37].

Como se explicó en el Capítulo II, el modelo SVM representa los puntos de la muestra en el espacio n-dimensional (conjunto de hiperplanos), con ello, en función de la cercanía se puede realizar la clasificación en la clase correspondiente. Si existe una buena separación entre las clases, se permitirá una clasificación correcta, a la hora de inferir la clase correspondiente a una instancia. Los clasificadores MVS poseen un entrenamiento y clasificación muy eficientes, tienen un buen funcionamiento con problemas típicos, y son extremadamente robustos para generalización, con menos necesidad de heurísticos para entrenamiento. El entrenamiento de una MVS requiere la solución a un gran problema de programación cuadrática (QP)[53]. El SMO divide este gran problema en una serie de

[53] La programación cuadrática (QP) es un tipo especial en la matemática de optimización de problemas. Es el problema de optimizar (reduciendo al mínimo o maximizando) una función cuadrática de varias variables conforme a apremios lineales en estas variables.

problemas más pequeños QP. Estos problemas más pequeños son resueltos de forma analítica, lo cual reduce significativamente el tiempo del ciclo interno de procesamiento. La cantidad de memoria requerida para SMO es lineal, en el tamaño del conjunto de entrenamiento, lo cual permite entrenar con grandes conjuntos de datos [66]. En WEKA el algoritmo SMO tiene varios parámetros para configurar, al igual que con la RNA, pulsando sobre el clasificador abre una venta para modificar los valores (Figura 14).

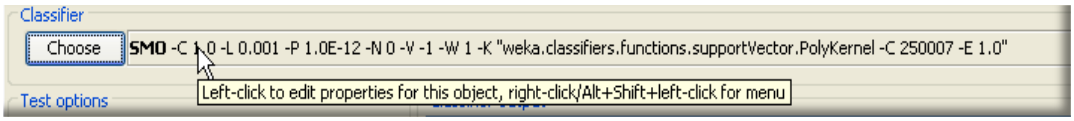


Figura 45: Configurar el algoritmo SMO.

En la Figura 15, se muestra la ventana para configurar el clasificador, y se describen brevemente cada uno de los parámetros. En la página *http://wiki.pentaho.com/display/datamining/mism*, se puede acceder a más información sobre los parámetros. Además, para mayor información sobre el algoritmo SMO, se puede consultar el libro de Bernhard Scholkopy y Alexander J. Smola [67]

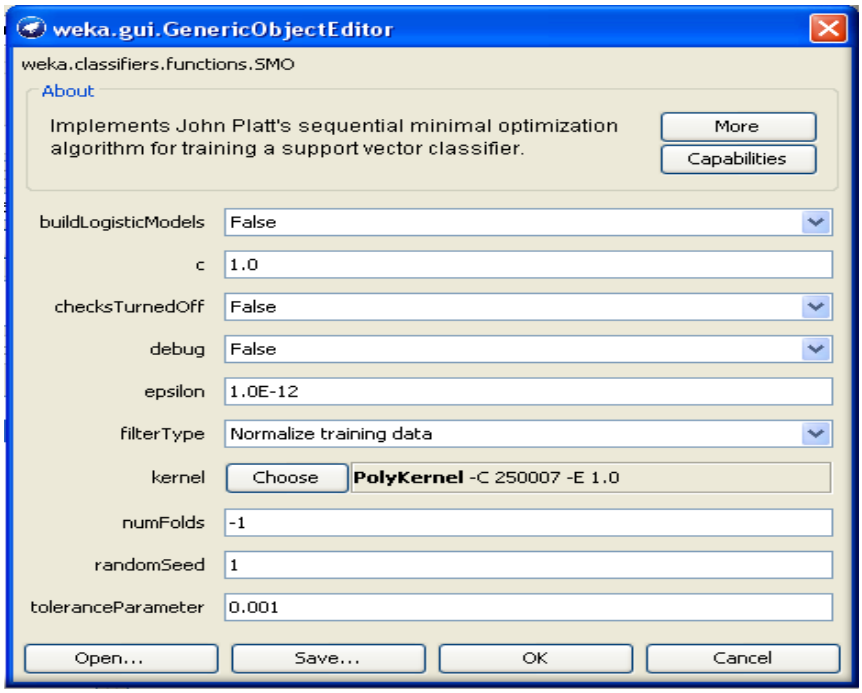


Figura 46: Configurar el algoritmo SMO.

- *BuildLogisticModels*: Si se deben adaptar modelos logísticos a los resultados (para estimaciones de probabilidad adecuadas).
- *C*: Parámetro de la complejidad. Relacionado al aumento del tiempo computacional (C = 1.0);

- *ChecksTurnedOff*: Desactiva cheques que consumen mucho tiempo: úselos con precaución.
- *D*ebug: Si se establece en verdadero, el clasificador puede generar información adicional a la consola.
- Epsilon: Factor para redondeo de error, que no debe ser modificado (P = 1.0E-12).
- *FilterType*: Determina si los datos serán transformados.
- *Kernel:* Permite seleccionar el kernel a usar.
- *NumFolds* - Número de pliegues para la validación cruzada, utilizados para generar datos de entrenamiento de modelos logísticos (V = -1: significa utilizar los datos de entrenamiento);
- *RandomSeed* - Semilla de números aleatorios para su uso método de validación cruzada (W = 1);
- *ToleranceParameter:* Parámetro de tolerancia, relacionado con la cantidad de iteraciones para el aprendizaje de la clase (WEKA). No debe cambiarse (L = 0.0010).

En WEKA, el propio SMO normaliza las características de entrada nominales, de acuerdo con sus necesidades, ya que no son características admitidas directamente. Al igual que para el experimento anterior, se eligió para realizar el testeo, en las opciones "*Modo Test", la opción: Percentage split,* con el valor configurado a 66.0% para realizar entrenamiento y test, en vista de sacar conclusiones sobre los resultados obtenidos.

Primeramente aplicamos a la base de datos la función Kernel Polinomial, y luego la función de Base Radial (RBF), con los parámetros por defecto que propone WEKA, para luego buscar la mejor configuración de los parámetros del mejor kernel. En la Figura 48, se muestra los resultados que arrojó el software, usando la función Kernel Polinomial:

$$\mathbf{K(x,y) = <x,y>} \qquad \mathbf{(30)}$$

y en la Figura 49 los resultados de la función RBF kernel:

$$\mathbf{K(x,y) = e^{-(0.01* <x-y,x-y>^2)}} \qquad \mathbf{(31)}$$

Como se puede observar la primer experimentación obtuvo mejor resultado usando por defecto los valores: C = 1, filtro de normalización del conjunto de datos, y p = 1.

```
Time taken to build model: 1.23 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances         470               92.1569 %
Incorrectly Classified Instances        40                7.8431 %
Kappa statistic                          0.6649
Mean absolute error                      0.0784
Root mean squared error                  0.2801
Relative absolute error                 30.513  %
Root relative squared error             80.2928 %
Total Number of Instances               510

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.961     0.319      0.948     0.961     0.955      0.821    NO_ABANDONO
                 0.681     0.039      0.742     0.681     0.71       0.821    ABANDONO
Weighted Avg.    0.922     0.28       0.919     0.922     0.92       0.821

=== Confusion Matrix ===

   a   b   <-- classified as
 421  17 |   a = NO_ABANDONO
  23  49 |   b = ABANDONO
```

Figura 47: Salida de WEKA usando la función Kernel Polinomial

```
Time taken to build model: 2.8 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances         457               89.6078 %
Incorrectly Classified Instances        53               10.3922 %
Kappa statistic                          0.4699
Mean absolute error                      0.1039
Root mean squared error                  0.3224
Relative absolute error                 40.4297 %
Root relative squared error             92.424  %
Total Number of Instances               510

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.977     0.597      0.909     0.977     0.942      0.69     NO_ABANDONO
                 0.403     0.023      0.744     0.403     0.523      0.69     ABANDONO
Weighted Avg.    0.896     0.516      0.885     0.896     0.883      0.69

=== Confusion Matrix ===

   a   b   <-- classified as
 428  10 |   a = NO_ABANDONO
  43  29 |   b = ABANDONO
```

Figura 48: Salida de WEKA usando la función RBF kernel

Como resultado de la comparación entre ambos, la función Polinomial de Kernel es la elegida como el mejor Kernel para esta clasificación con SMO. Entonces, primero es necesario ajustar adecuadamente los valores de los parámetros *C y p*. Ambos valores, se ensayan empíricamente, con el mismo *Modo Test* usado anteriormente, buscando cual es el que arroja la mejor precisión, que será el que luego se use en la comparación con la RNA. Se aplicará, entonces, la función Kernel Polinomial, dada en la ecuación 3.1, buscando la mejor configuración de los parámetros. WEKA trae por defecto los parámetros C = 1, filtro de

normalización del conjunto de datos y p = 1. Variando primeramente el tipo filtro obtenemos los siguientes niveles de precisión (C = 1, p = 1). Procediendo a estandarizar los datos de entrenamiento, y utilizando todos los casos, variamos los parámetros p y C, contrastando los niveles de exactitud arrojados por las respectivas matrices de confusión (Tabla 96, Figura 50).

	p=0,5	*p=1*	*p=2*	*p=3*
C=0,5	**92.1569 %**	**85.8824 %**	**85.8824 %**	**85.8824 %**
C=1	**92.1569 %**	**91.1274 %**	**85.8824 %**	**85.8824 %**
C=2	**92.3529 %**	**85.8824 %**	**85.8824 %**	**85.8824 %**
C=3	**92.1569 %**	**85.8824 %**	**85.8824 %**	**85.8824 %**

Tabla 9. Porcentajes de Instancias correctamente clasificadas según valores de C y p.

A fin de reducir la complejidad del modelo, escogemos como la mejor configuración de parámetros C = 2 y p = 0.5, cuyo modelo arroja para el conjunto de entrenamiento una exactitud del 92.3529 % y la siguiente matriz de confusión:

```
=== Confusion Matrix ===

   a   b   <-- classified as
 421  17 |   a = NO_ABANDONO
  22  50 |   b = ABANDONO
```

Figura 49: Salida de WEKA usando la función RBF kernel

Al igual que en el ensayo anterior, se guarda el modelo creado para su posterior uso con los datos de la cohorte 2015.

4.8. Comparación de los Modelos

Con el fin de realizar un análisis comparativo del desempeño de un clasificador del tipo RNA, frente a una MVS, se construyeron dos modelos, uno para cada clasificador, basados en los datos de las cohortes 2013-2014. A continuación, se describen los pasos para realizar la carga de esos modelos, para luego ser probados con datos nuevos, los que corresponden a la cohorte 2015. En la Figura 51 se puede observar cómo se carga un modelo previamente guardado.

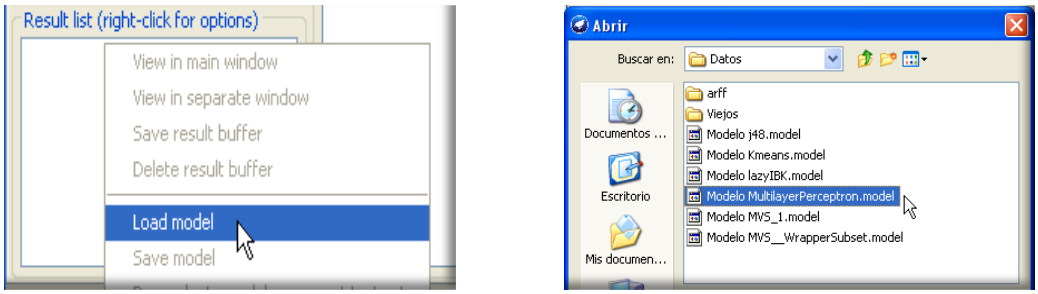


Figura 50: Salida de WEKA usando la función RBF kernel

En el recuadro "Opciones de prueba" (*Test Options*), tenemos que seleccionar "Conjunto de prueba suministrado" (*Supplied test set*), donde se abre una ventana que nos permite cargar el archivo con los datos de testeo (Figura 20).

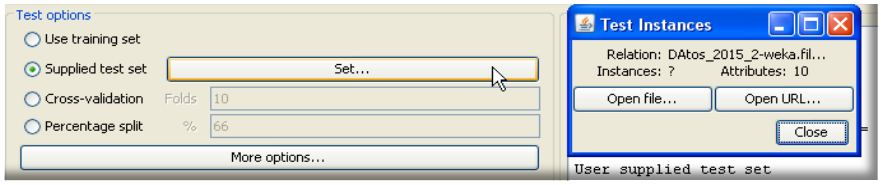
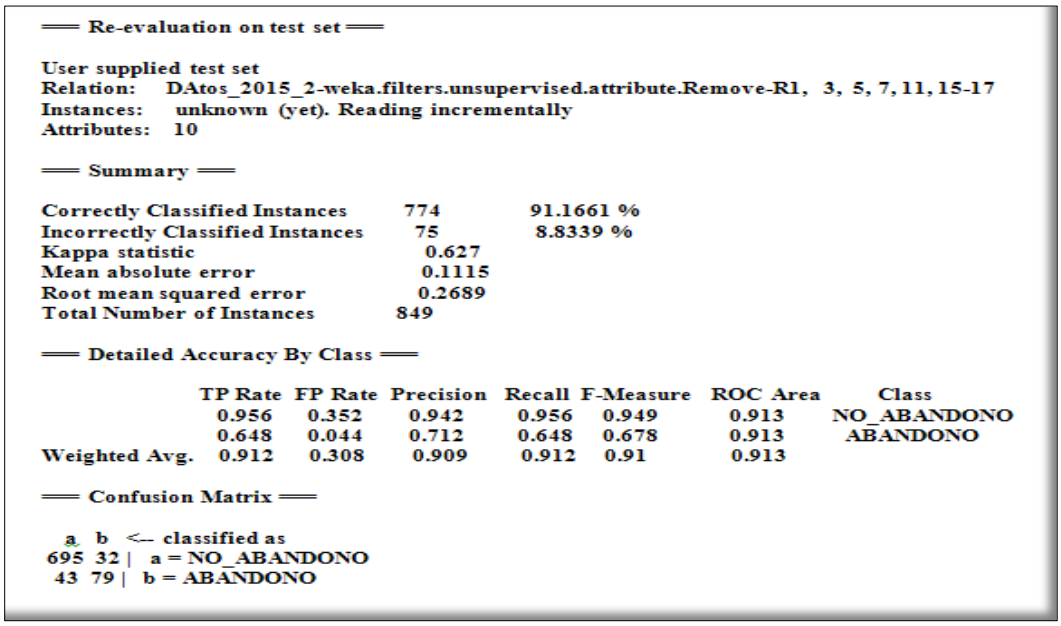


Figura 51: Salida de WEKA usando la función RBF kernel

Finalmente, pulsando el botón derecho en el modelo, y se ejecuta Re-evaluar el modelo, con el conjunto de prueba actual (*Re-evaluate model on current test set*). Este proceso se repitió para ambos modelos. En el Anexo II, se puede ver el resultado total de cada corrida.

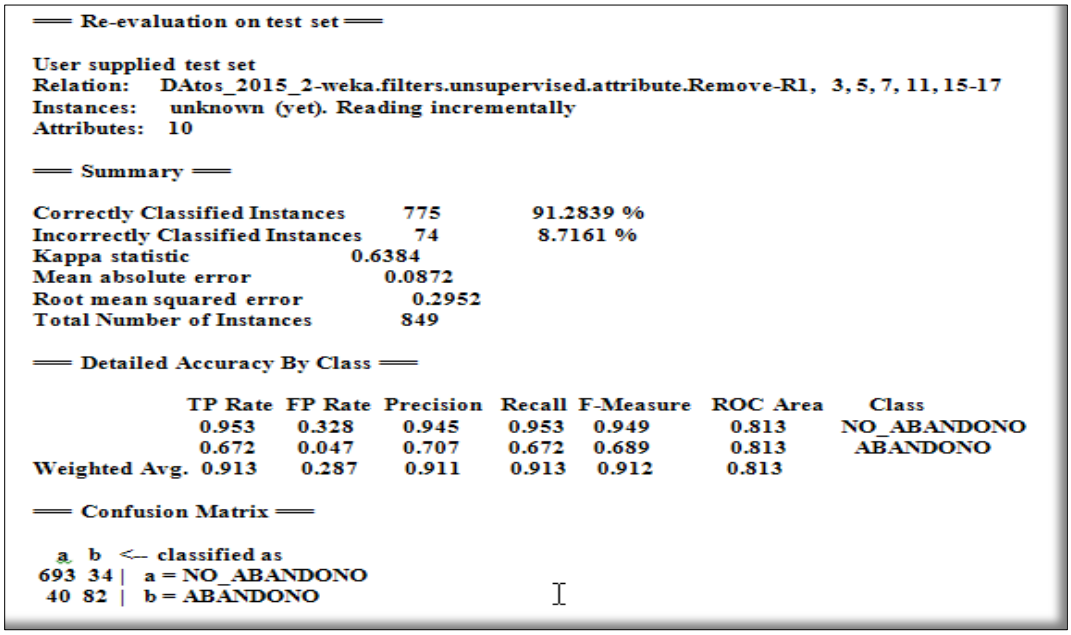
La prueba de reconocimiento se realizó con una MVS, y una RNA, programadas para WEKA. Cabe destacar que una MVS es esencialmente un clasificador binario no lineal, capaz de determinar si un vector de entrada "x" pertenece a una clase 1 (la salida deseada sería entonces y = 1) o a una clase 2 (y = −1). Para la realización de estas pruebas, además, se utilizó una RNA con un algoritmo de retropropagación, lo que la hace un tipo de red con aprendizaje supervisado, el mismo emplea un ciclo propagación-adaptación de dos fases. Una vez aplicado un patrón de entrenamiento a la entrada de la red, este se propaga desde la primera capa, a través de las capas subsecuentes de la red, hasta generar una salida, la que es comparada con la salida deseada. Se calcula una señal de error para cada una de las salidas, que a su vez, es propagada hacia atrás, empezando desde la capa de salida, y hacia todas las capas de la red hasta llegar a la capa de entrada. Todo esto con la finalidad de actualizar los pesos de conexión de cada neurona, y lograr que la red converja a un estado que le permita clasificar correctamente todos los patrones de entrenamiento. La estructura general se

muestra en la Figura 11. En las Figuras 53 y 54, se muestran los resultados de la ejecución de los modelos con los nuevos datos.



```
=== Re-evaluation on test set ===

User supplied test set
Relation:     DAtos_2015_2-weka.filters.unsupervised.attribute.Remove-R1,  3, 5, 7, 11, 15-17
Instances:    unknown (yet). Reading incrementally
Attributes:   10

=== Summary ===

Correctly Classified Instances         774               91.1661 %
Incorrectly Classified Instances        75                8.8339 %
Kappa statistic                          0.627
Mean absolute error                      0.1115
Root mean squared error                  0.2689
Total Number of Instances              849

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
                   0.956    0.352      0.942   0.956      0.949     0.913    NO_ABANDONO
                   0.648    0.044      0.712   0.648      0.678     0.913    ABANDONO
Weighted Avg.      0.912    0.308      0.909   0.912      0.91      0.913

=== Confusion Matrix ===

   a   b   <-- classified as
 695  32 |   a = NO_ABANDONO
  43  79 |   b = ABANDONO
```

Figura 52: Salida de WEKA del algoritmo RNA.



```
=== Re-evaluation on test set ===

User supplied test set
Relation:     DAtos_2015_2-weka.filters.unsupervised.attribute.Remove-R1,  3, 5, 7, 11, 15-17
Instances:    unknown (yet). Reading incrementally
Attributes:   10

=== Summary ===

Correctly Classified Instances         775               91.2839 %
Incorrectly Classified Instances        74                8.7161 %
Kappa statistic                          0.6384
Mean absolute error                      0.0872
Root mean squared error                  0.2952
Total Number of Instances              849

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
                   0.953    0.328      0.945   0.953      0.949     0.813    NO_ABANDONO
                   0.672    0.047      0.707   0.672      0.689     0.813    ABANDONO
Weighted Avg.      0.913    0.287      0.911   0.913      0.912     0.813

=== Confusion Matrix ===

   a   b   <-- classified as
 693  34 |   a = NO_ABANDONO
  40  82 |   b = ABANDONO
```

Figura 53: Salida de WEKA del algoritmo MVS.

4.9. Evaluación

La validez de esta prueba, se define como la habilidad para identificar correctamente aquellos alumnos que tienen posibilidad de abandonar sus estudios, y aquellos que no la tienen.

4.9.1. Métricas de evaluación

La validez de una clasificador posee dos componentes importantes que brindan la exactitud de la misma: la *sensibilidad* y la *especificidad* [ ]. Cuando se utiliza una prueba dicotómica, una cuyos resultados se puedan interpretar directamente como positivos o negativos, la *sensibilidad* es la probabilidad de clasificar correctamente a un individuo, cuyo estado real sea el definido como positivo respecto a la condición que estudia la prueba, razón por la que también es denominada *Fracción de Verdaderos Positivos* (FVP), o simplemente *Verdaderos Positivos* (VP). La *especificidad* es la probabilidad de clasificar correctamente a un individuo, cuyo estado real sea el definido como negativo. Es igual al resultado de restar a uno (1), la *Fracción de Falsos Positivos* (FFP) [ ]. Cuando los datos de una muestra se clasifican en una matriz de confusión, o tabla de contingencia por el resultado de la prueba, y su estado respecto a la clase u etiqueta, es fácil estimar a partir de ella la *sensibilidad*, y la *especificidad* de la prueba.

4.9.1.1. Matriz de confusión

A continuación se procederá a explicar una matriz de confusión, en base a los resultados de clasificación que se obtendrían a partir de una clase con 2 categorías, *Abandona o No Abandona, sí o no, compra, no compra, acepta o rechaza,* etcétera. Además, una de las categorías de la clase debe ser considerada como la categoría de interés, la principal, la otra sería la categoría secundaria. Continuando con el ejemplo presentado, se determina que:

i. **Categoría principal (+): A.**
ii. **Categoría secundaria (–): B.**

Entonces, la clasificación realizada a una nueva instancia puede corresponder a uno de los siguientes cuatro (4) tipos de resultados: verdadero positivo, falso positivo, verdadero negativo, y falso negativo [20]. Estos tipos de resultados se muestran en la siguiente matriz de confusión figura 23:

		Verdadero Diagnóstico	
		Abandona	No Abandona
Resultado de la Prueba	Prueba Positiva	Verdadero Positivo (VP)	Falso Positivo (FP)
	Prueba Negativa	Falso Negativo (FN)	Verdadero Negativo (VN)
		VP + FN	VN + FP
Sensibilidad	= VP/(VP + FN) = FVP (fracción de verdaderos positivos)		
Especificidad	= VN/(VN + FP) = FVN (fracción de verdaderos negativos) = 1 - FFP (fracción de falsos positivos)		

Figura 54: Resultado de una prueba y su estado respecto a la clase.

Los verdaderos positivos (VP), y verdaderos negativos (VN), son las instancias clasificadas correctamente. Los falsos positivos (FP) son las instancias clasificadas como la categoría de interés (positivo), cuando es de la categoría secundaria (negativo). Los falsos negativos (FN) son las instancias clasificadas como la categoría secundaria (negativo), cuando es de la categoría principal (positivo) [20]. Estos resultados, contextualizados al ejemplo aplicativo se presentan a continuación:

• *Verdadero positivo* (VP). Señala la cantidad de instancias clasificadas como la categoría A, perteneciendo realmente a dicha categoría.
• *Verdadero negativo* (VN). Señala la cantidad de instancias clasificadas como la categoría B, perteneciendo realmente a dicha categoría.
• *Falso positivo* (FP). Señala la cantidad de instancias clasificadas como la categoría A, perteneciendo realmente a la categoría B.
• *Falso negativo* (FN). Señala la cantidad de instancias clasificadas como la categoría B, perteneciendo realmente a la categoría A.

Medida	Forma de cálculo		Interpretación
Éxito (Exactitud)	Clasificaciones acertadas con respecto al total de instancias.	$\frac{VP + VN}{total}$	Proporción de instancias bien clasificadas
Error	Clasificaciones erradas con respecto al total de instancias.	$\frac{FP + FN}{Total}$	Proporción de instancias mal clasificadas
Sensibilidad	Clasificaciones acertadas con respecto al total de instancias de la categoría de interés.	$\frac{VP}{VP + FN}$	Probabilidad de clasificar correctamente una instancia en la categoría de interés (+).
Especificidad	Clasificaciones acertadas con respecto al total de instancias de la categoría secundaria.	$\frac{VN}{VN + FP}$	Probabilidad de clasificar correctamente una instancia en la categoría secundaria (–).
Mean Absolute error (MAE)	Se define error absoluto de una medida la diferencia entre el valor medio obtenido y el hallado en esa medida todo en valor absoluto.	$\frac{\sum_{i=1}^{n} y_i - x_i}{n}$	Es la suma de los errores absolutos de clasificación en cada uno de los sujetos llevados a promedio. El clasificador que arroje mayor cifra (mayor a 0.1), define un error de clasificación alto, por lo cual no se debe considerar por sobre los que arrojen una cifra menor.
Tiempo de creación del modelo	Medido en segundos, es la cantidad de tiempo que demora en construir la arquitectura del clasificador, y en arrojar resultados.		Puede que un clasificador se defina como eficiente, si el tiempo que emplea en emitir resultados es menor a 5 segundos.

Tabla 10. Medidas de evaluación.

Otra forma de evaluar el rendimiento del clasificador utilizado es a través de su curva ROC [].

4.9.1.2. ¿Qué es una curva ROC?

Una curva de características de funcionamiento del receptor, o curva ROC (Receiver Operating Characteristics), es una técnica utilizada para la visualización, organización y selección de sistemas de clasificación basándose en su rendimiento. El análisis de las curvas ROC ha sido principalmente usado en el área de la medicina, con el fin de analizar el comportamiento de sistemas diseñados para el diagnóstico de enfermedades. Recientemente, el análisis de las curvas ROC ha incrementado su uso en el área del aprendizaje de máquina. Esto se debe principalmente, a que a partir de estos análisis se pueden obtener métricas que permiten la evaluación del rendimiento de un modelo de clasificación []. En la Figura 24, se pueden observar distintas curvas ROC, y su significado respecto a los posibles resultados.

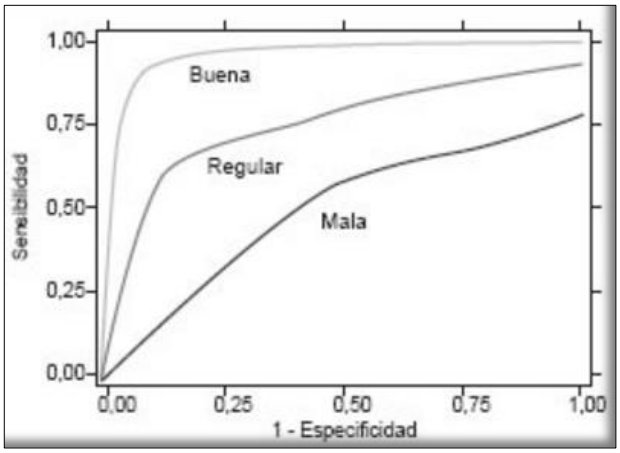


Figura 55: Esquema explicativo de las curvas ROC.

4.9.1.3. Área bajo la curva ROC

El área bajo la curva ROC (Area Under the Curve, AUC), es el mejor indicador global de la precisión de una prueba diagnóstica. Hace factible expresar el desempeño de una prueba mediante un número simple. Esta área es siempre mayor, o igual a 0,5. El rango de valores se mueve entre 1, discriminación perfecta, y 0,5, no hay diferencias en la distribución de los valores de la prueba entre los 2 grupos. La interpretación del valor del área sería del modo siguiente, un área de 0,8 significa que un individuo seleccionado aleatoriamente del grupo de clase A (Categoría Principal), tiene un valor de la prueba mayor que uno seleccionado aleatoriamente del grupo de clase B (Categoría secundaria), en el 80 % de las veces.

Mediante una prueba de hipótesis, y/o de la estimación del intervalo de confianza para el área, es posible evaluar la precisión de un proceder diagnóstico. Rechazar la hipótesis de que el área teórica es igual a 0,5, $p < 0,05$ y/o intervalo de confianza que no contiene al 0,5, proporciona evidencia de que la prueba diagnóstica tiene la habilidad para distinguir entre los 2 subgrupos.

Utilizando una prueba de hipótesis es posible comparar varias áreas bajo la curva ROC, lo que permite hacer distinciones entre el poder discriminatorio de 2, o más procederes diagnósticos, cuando estos se han realizado en el mismo grupo de datos.

4.10. Resultados en datos

Los resultados obtenidos en los anteriores procesos de clasificación se resumen en la Tabla 9:

Métrica	RNA	MVS
Éxito (Exactitud)	**91.16 %**	**91.28 %**
Error	**8.83 %**	**8.71 %**
Sensibilidad	**94.17 %**	**94.51 %**
Especificidad	**71.17 %**	**70.69 %**
(MAE)	**0.1115**	**0.0872**
ROC Area	**0.913**	**0.813**
Tiempo de creación	**21.09’’**	**1.55’’**

Tabla 11. Comparación de los resultados obtenidos.

El AUC se define como la probabilidad de clasificar correctamente un par de individuos, Abandona y No Abandona, seleccionados al azar de la población, mediante los resultados obtenidos al aplicar la prueba diagnóstica [33,y 34]. Los datos ofrecidos en las Figuras 24 y 25, muestran que el valor de la AUC se mantuvo por encima del 83% entre los dos clasificadores. En la ejecución de las iteraciones de prueba, el clasificador RNA obtuvo un 91.3% de AUC, mayor valor obtenido..

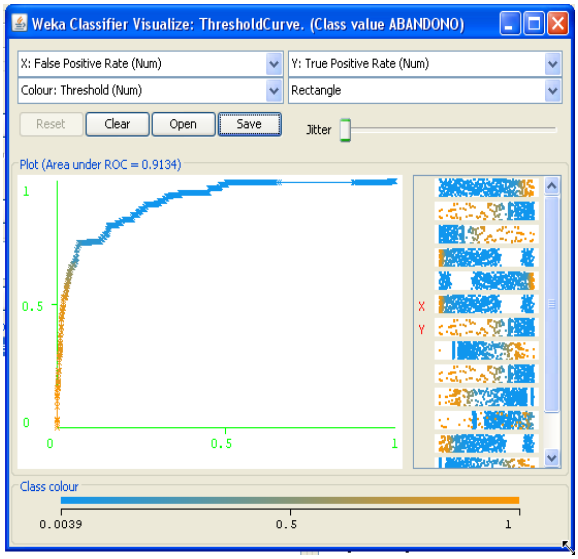


Figura 56: Área bajo ROC de RNA.

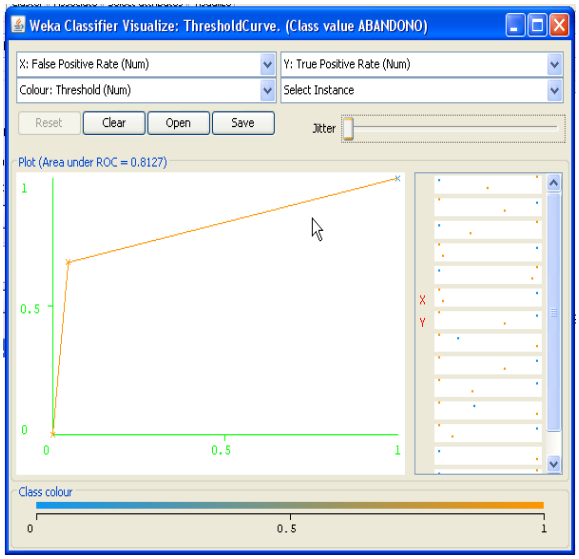


Figura 57: Área bajo de MVS.

Ninguna cantidad de experimentos pueden probar que tengo razón;
un solo experimento puede probar que me he equivocado.
Albert Einstein

Capítulo 5

5. Conclusiones y Trabajos Futuros

El objetivo primario de esta investigación es utilizar la metodología de descubrimiento de conocimiento en bases de datos (KDD), en la construcción de un modelo de minería de datos educacional Con el fin de conocer cual tiene el mejor resultado de clasificación para estos tipos de datos, se comparó entre dos métodos de clasificación automática,

5.1. Conclusiones

1. La construcción de modelos usando MVS y RNA, es un proceso bastante arduo, que implica encontrar primero los rangos de operación en los cuales va a trabajar el sistema, para posteriormente, mediante sucesivas pruebas encontrar los principales parámetros para configurar los algoritmos.
2. No existe un modelo clasificador mejor que otro, de manera general. Es por esto que han surgido varias medidas para evaluar la clasificación, y comparar los modelos empleados para un problema determinado. Los resultados mostraron que ambos clasificadores pueden alcanzar prestaciones similares en la identificación de esta problemática.
3. Ambos clasificadores incorporan la información de los datos mediante modelos matemáticos, o teóricos, construidos a partir de la (o teoría, o teorema, las dos juntas no) de minimización de riesgo, la cual como se observó, ofrece un modelo que minimiza la probabilidad total del error.
4. Para la construcción de modelos de clasificación, se comprobó que se puede utilizar una menor cantidad de atributos, del total inicial, para obtener un resultado considerado aceptable por el investigador. Sin embargo, el problema radica en la elección, y el número de atributos a utilizar, debido a que determina la efectividad del modelo de discriminación construido.
5. El clasificador MVS tiene un ratio de acierto ligeramente superior al del clasificador MLP, y el proceso de entrenamiento es más rápido, sobre todo para conjuntos de

entrenamiento grandes. Sin embargo, el proceso de clasificación es más lento que en un clasificador MLP, y necesita mucha más memoria.

6. Debido a la insuficiencia de trabajos en esta área, con estos tipos de datos, no ha sido posible realizar una comparación objetiva de los resultados del trabajo presente; sin embargo, al analizar las ventajas del sistema propuesto, se puede visualizar una nueva gama de aplicaciones, que tiendan a una correcta identificación de los alumnos en situaciones de riesgo de deserción.
7. Por último, se puede utilizar el porcentaje de aciertos para elegir el mejor modelo, Existe otra medida, que se basa en las curvas ROC, y que es inmune al desequilibrio en la muestra, el AUC. Cuanto mayor sea esta área, más cerca está la curva ROC de la esquina superior izquierda, y por tanto mejor es la separación de los datos, independientemente del desequilibrio de la muestra. En este caso el modelo de RNA, resultó mejor que la MVS.

5.2. Trabajos Futuros

Coincidiendo con las recomendaciones que hace Erwin Sergio Fischer Angulo [17], se propone que se incorporen nuevos atributos, sobre todo los asociados a factores individuales, institucionales, y socioeconómico. Asimismo, se recomienda que estos sean capturados al momento que el estudiante se matricula en la institución, con el objeto poder crear un repositorio de registro histórico de las condiciones de entrada, completándolo con el resultado del desempeño de los estudiantes en su carrera, indicando si es desertor o no.

Además, como trabajo futuro se propone aplicar nuevas configuraciones a los modelos estudiados, como así también, experimentar la aplicación de otros métodos de clasificación a las vistas minables creadas, para conocer si algún método resulta más adecuado a la estructura de estos datos, y dar una mejor precisión en la clasificación.

6. Bibliografía

[1] The 8th International Conference on Educational Data Mining Edm 2015. [En línea] http://www.educationaldatamining.org/ EDM2015/, ultimo acceso el 05/11/2017.

[2] Molen Moris, J. (2013). Minería de datos educacionales: modelos de predicción del desempeño escolar en alumnos de enseñanza básica. Santiago, Chile: Universidad de Chile - Facultad de Ciencias Físicas y Matemáticas. Descargado desde: http://www.repositorio.uchile.cl/handle/2250/113034, el 05/11/2017.

[3] García Tinisaray, D. (2015). Construcción de un modelo para determinar el rendimiento académico de los estudiantes basado en learning analytics (análisis del aprendizaje), mediante el uso de técnicas multivariantes. Tesis doctoral. Dpto. de Estad. e Inv. Operativa. Univ. de Sevilla. Descargado desde: http://hdl.handle.net/11441/40436, el 05/11/2017.

[4] Piatetski-Shapiro, G., Frawley, W.J., Matheus, C.J. (1991). Knowledge discovery in databases: an overview. AAAI-MIT Press, Menlo Park, California. AI Magazine Volume 13 Number 3 (1992) (© AAAI). Descargado el 05/11/2017desde: https://pdfs.semanticscholar.org/7a7b/51b86e22d0077215287980c7ba793b09e4cd.pdf, el.

[5] Hernández Orallo, J. (2014), Introducción a la minería de datos. Pearson Education, Valencia. 2004. Editorial Pearson, 2004. ISBN: 84 205 4091 9.

[6] Valcárcel Asencios, V. (2004). Data Mining y el descubrimiento del conocimiento. Revista de la Facultad de Ingeniería Industrial Vol. (7) 2: pp. 83-86. UNMSM ISSN: 1560-9146 (impreso) / ISSN: 1810-9993 (electrónico). Descargado el 05/11/2017 desde: http://sisbib.unmsm.edu.pe/bibvirtualdata/publicaciones/indata/Vol7_n2/Pdf/a13.pdf.

[7] Fayyad, U.M., Piatetsky-Shapiro and P. Smyth, (1996). "From Data Mining to Knowledge Discovery: An Overview" AI Magazine Vol. 17 Number 3. Descargado desde: https://www.aaai.org/ojs/index.php/aimagazine/article/viewFile/1230/1131, el 05/11/2017.

[8] Tinto, V. (1982). Definir la Deserción: Una Cuestión de Perspectiva. Rev de Ed. Sup. Descargado el 05/11/2017 desde: http://www.alfaguia.org/alfaguia/files/1342823160_52.pdf.

[9] Lorenzano, C. La deserción universitaria en la Universidad Nacional de Tres de Febrero. Dpto. de Cs. Sociales. U. Nacional de Tres de Febrero. Descargado desde: http://www.untref.edu.ar/documentos/AutoevaluacionLadesercion.pdf, el 05/11/2017.

[10] Ferreyra, A. y González, E. (2000). Reflexiones Sobre La Deserción Universitaria. Grupo de Enseñanza de la Ciencia y la Tecnología (GECYT). Facultad de Matemática,

Astronomía y Física (FaMAF). Universidad Nacional de Córdoba. Descargado desde: http://www.raco.cat/index.php/ensenanza/article/viewFile/21655/21490, el 05/11/2017.
[11] Dra. De Felippi. I. (2011). Ingreso y retención universitaria. Gestión Universitaria ISSN 1852-1487. Vol.:04 Nro.:01. Descargado desde: http://www.gestuniv.com.ar/gu_10/v4n1a2.htm, el 05/11/2017.
[12] Giuliano, M. (2014]. Factores que afectan la permanencia de los estudiantes en las carreras de ingeniería de la UNLaM. Descargado desde: http://ingenieria.unlam.edu.ar/index.php?seccion=4, el 05/11/2017.
[13] del Alcázar León D. (2014). Sistema inteligente para perfilar la deserción en estudiantes universitarios de carreras técnicas. Número 10. Primer semestre 2014 Serie: Cuadernos del Contrato Social por la Educación. Descargado desde: http://contratosocialecuador.org/images/publicaciones/cuadermos/10.pdf, el 05/11/2017.
[14] Timarán Pereira, R. (2014). Detección de Patrones de Deserción Estudiantil en Programas de Pregrado de Instituciones de Educación Superior con CRISP-DM. Univ. de Nariño, Pasto-Nariño-Colombia. C. I. de Ciencia, T., I. y E. ISBN: 978-84-7666-210-6 Descargado desde: www.oei.es/congreso2014/memoriactei/758.pdf, el 05/11/2017.
[15] Formia, S. (2012). Evaluación de técnicas de Extracción de Conocimiento en Bases de Datos y su aplicación a la deserción de alumnos universitario. Trabajo final presentado para obtener el grado de Especialista en Tecnología Informática aplicada en Educación. Facultad de Informática. Universidad Nacional de La Plata. Descargado el 05/11/2017 desde: http://sedici.unlp.edu.ar/bitstream/handle/10915/26772/Documento_completo.pdf?sequence=1.
[16] Pautsch, Jesús G. (2009). Minería de Datos aplicada al análisis de la deserción en la Carrera de Analista en Sistemas de Computación. Tesis de grado Licenciatura en Sistemas de Inf. U.N. de Misiones. Fac. de Cs. Exactas, Químicas y Naturales. Descargado desde: http://exa.unne.edu.ar/informatica/SO/TFAGermanPAUTSCHFinal.pdf, el 05/11/2017.
[17] Fischer Angulo E.S. (2012). Modelo para la automatización del proceso de determinación de riesgo de deserción en estudiantes universitarios. Santiago De Chile U. de Chile. Descargado desde: http://repositorio.uchile.cl/bitstream/handle/2250/111188/cf-fischer_ea.pdf?sequence=1, el 05/11/2017.
[18] Modelado y minería de datos. (2015). IBM Knowledge Center. En línea: http://www.ibm.com/support/knowledgecenter/es/SS3RA7_16.0.0/com.ibm.spss.

[19] http://www.lanacion.com.ar/1779480-mas-matriculados-pero-pocos-graduados-en-las-universidades.
[20] Martínez Álvarez, C. (2012). Aplicación de Técnicas de Minería de Datos para mejorar el proceso de control de gestión en Entel. Universidad de Chile. Disponible en: http://repositorio.uchile.cl/bitstream/handle/2250/112065/cf-martinezca.pdf?sequence=1
[21] García Herrero, J. y Molina López, J. (2012). Técnicas de análisis de datos aplicaciones prácticas utilizando Microsoft Excel y Weka. Disponible en: http://ocw.uc3m.es/ingenieria-informatica/analisis-de-datos/libroDataMiningv5.pdf
[22] Tejeda Ávila, E. (2010). Data Warehousing con procesamiento de datos textuales. Universidad de Granada. Disponible en: http://www.tesisenred.net/handle/10803/17316, el 24/11/2017.
[23] Inmon, Bill. (1996). Building The Datawarehouse. Edición 2. Editorial Wiley Computer Publishing.
[24] Guevara Lenis Jorge Eduardo, Valencia Arcos Janeth del Carmen, "Data Warehouse para el Análisis Académico de la Escuela Politécnica Nacional". Escuela Politécnica Nacional. Escuela de Ingeniería en Sist. Informáticos y de Computación. Quito, Junio 2007.
[25] Spositto, O. y otros. (2010). Aplicación de técnicas de minería de datos para la evaluación del rendimiento académico y la deserción estudiantil. 9º Conferencia Iberoamericana en Sist., Cibernética e Inf., CISCI 2010. Descargado el 05/11/2017: http://www.iiis.org/CDs2010/CD2010CSC/CISCI2010/PapersPdf/ CA156FK.pdf.
[26] Rykeboer, H. y otros. (2009). Aplicación de técnicas de minería de datos para la evaluación del rendimiento académico y la deserción estudiantil. Anuario de Investigaciones. Resúmenes Extendidos. ISBN: 978-987-1635-23-8
[27] Rykeboer, H. y otros. (2010). Utilización de técnicas de Data Warehouse para la toma de decisiones en el área académica. Anuario de Investigaciones. Resúmenes Extendidos. ISBN: 978-987-1635-55-9
[28] Rykeboer, H. y otros. (2013). Implementación de un Data Warehouse para la toma de decisiones en el área académica. Anuario de Investigaciones. Resúmenes Extendidos. ISBN: 978-987-3806-30-8
[29] Inmon, W. and Richard D. (1994). Using the Data Warehouse. 1ra Edición. Ed: John Wiley & Sons, ISBN-13: 978-0471059660

[30] Inmon, W. (1996). Building The Datawarehouse. 2da Edición. Editorial Wiley Computer Publishing. ISBN: 0-471-08130-2

[31] Kimball & Ross. (2002). The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (2da Edition), New York, Wiley. ISBN: 978-1-118-53080-1

[32] Elmasri, R. y Navathe, S. (2007). Fundamentos de sistemas de bases de datos. Editorial: Addison-Wesley. ISBN: 9788478290857.

[33] Heughes, V. Escobar J. (2007). Minería Web de Uso y Perfiles de Usuario: Aplicaciones con Lógica Difusa. U. de Granada. ISBN: 978-84-338-4707-2.

[34] Witten, I. and Frank, E. (2000). Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann. 2da Edición. ISBN: 0-12-088407-0

[35] Weiss, S. and Indurkhya, N. (1988). Predictive Data Mining: A Practical Guide. Morgan Kaufmann Publishers Inc., San Francisco, USA. ISBN-13: 978-1558604032

[36] Mitchell, M. (1997). Machine Learning, McGraw Hill. ISBN-13: 978-0070428072

[37] Cervantes Canales, J. (2009). Clasificación de grandes conjuntos de datos vía Máquinas de Vectores Soporte y aplicaciones en sistemas biológicos. Tesis Doctoral. Descargado de: https://www.cs.cinvestav.mx/TesisGraduados/2009/tesisJairCanales.pdf, el 24/11/2017.

[38] García Morate, D. Manual De Weka. Descargado el 24/11/2017 de: http://ucua.ujaen.es/jnavas/web_recursos/archivos/weka%20master%20recursos%20naturales/tutorial2%20weka%20mediano.pdf.

[39] Caicedo, E. y López, J.; (2009).Una aproximación práctica a las Redes Neuronales Artificiales. Cali: Universidad del Valle. Descargado desde: http://bibliotecadigital.univalle.edu.co/, el 24/11/2017.

[40] Del Brio, B.; Sanz Molina, A; (2007), Redes Neuronales y Sistemas Borrosos (3ª ed.). Edit. Ra-Ma, ISBN 978-970-15-1250-0

[41] Pasantes, H., De Neuronas, Emociones y Motivaciones. Instituto Latinoamericano de la Comunicación Educativa. Disponible en: http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen3/ciencia3/158/html/neuronas.html, última consulta. 24/11/2017

[42] Manes, F.; Niro, M. (204). Usar el cerebro. Conocer nuestra mente para vivir mejor., Bogotá, Editorial Planeta, ISBN: 978-997-47-0079-6.

[43] Haykin S. Neural Networks, McMaster University, Ontario, Canada, 1994. Descargado desde: https://cours.etsmtl.ca/sys843/REFS/Books/ebook_Haykin09.pdf, el 24/11/2017.

[44] Raúl Rojas. (1996). Neural Networks - A Systematic Introduction. Editorial Springer Verlag, New-York. Descargado desde: https://page.mi.fu-berlin.de/rojas/neural/neuron.pdf, el 24/11/2017.

[45] Galvão, V. (1999). Sistemas inteligentes: aplicações a recursos hídricos y ciencias

ambientales. Porto Alegre. ABRH/Editora de Universidade/UFRGS.
[46] Hilera J., Martínez V. (1995). Redes Neuronales Artificiales: Fundamentos, Modelos y aplicaciones. Madrid, España: RAMA Editorial. ISBN 978-84-7897-155-8
[47] Flórez L. R, Fernández L. (2008). Las Redes Neuronales Artificiales Metodología y Análisis de Datos en Ciencias Sociales. Editor Netbiblo. ISBN 978-84-9745-246-5.
[48] Bertona L.F. (2005), Entrenamiento de Redes Neuronales basado en Algoritmos Evolutivos. Tesis de grado en Ingeniería Informática Facultad de Ingeniería Universidad de Bs As. Descargado desde: http://laboratorios.fi.uba.ar/lsi/bertona-tesisingenieriainformatica.pdf, el 24/11/2017.
[49] Verma, A., Singh T. and Maheshwar,S. (2014). Comparative Study of Intelligent Prediction Models for Pressure Wave Velocity. Journal of Geosciences and Geomatics. Disponible en http://pubs.sciepub.com/jgg/2/3/9/index.html, última consulta 24/11/2017.
[50] Carmona Suárez, E. (2013). Tutorial sobre Máquinas de Vectores Soporte (SVM) Dpto. de Inteligencia Artificial, Univ. Nac. de Educación a Distancia (UNED), Madrid (España). Descargado desde: http://www.ia.uned.es/~ejcarmona/publicaciones/[2013-Carmona]%20SVM.pdf, el 24/11/2017.
[51] Pérez Ortiz, J. (1999). Clasificación con discriminantes: un enfoque neuronal. Dpto. de Lenguajes y Sistemas Informáticos Universidad de Alicante. Descargado desde: http://www.dlsi.ua.es/~japerez/pub/pdf/cden1999.pdf, el 24/11/2017.
[52] González Abril, L. (2014). Modelos de Clasificación basados en Máquinas de Vectores Soporte. Departamento de Economía Aplicada I Universidad de Sevilla. Descargado desde: https://www.researchgate.net/publication/267402740, el 24/11/2017.
[53] García González, F. (2013). Aplicación de técnicas de Minería de Datos a datos obtenidos por el Centro Andaluz de Medio Ambiente (CEAMA). España 2013. Descargado desde: http://masteres.ugr.es/moea/pages/tfm-1213/tfm_garciagonzalezfrancisco_1/!, el 24/11/2017.
[54] Santín González, D. (2008). Detección de alumnos de riesgo y medición de la eficiencia de centros escolares mediante redes neuronales. Universidad Complutense de Madrid. Descargado desde: http://eprints.ucm.es/6674/1/9902.pdf, el 24/11/2017.
[55] Barreto, S. y otros. (2004). Redes Neuronales para predecir el rendimiento académico de los alumnos ingresantes a la carrera de Bioquímica de la FACENA-UNNE en función de

sus conocimientos matemáticos previos. Facultad de Ciencias Exactas y Naturales y Agrimensura. Universidad Nacional del Nordeste. Corrientes. Descargado desde: http://sedici.unlp.edu.ar/bitstream/handle/10915/23739/Documentocompleto.pdf?sequence=1, el 24/11/2017.

[56] Amaya Torradoa, K. y otros. (2014). Modelo predictivo de deserción estudiantil utilizando técnicas de minería de datos. Universidad Simón Bolívar, Barranquilla, Colombia. Descargado desde: ttp://documentos.redclara.net/bitstream/10786/759/1/124-22-3-2014-Modelo%20predictivo%20de%20deserci%C3%B3n%20estudiantil%20utilizando%20t%C3%A9cnicas%20de%20miner%C3%ADa%20de%20datos.pdf, el 24/11/2017.

[57] Zambrano Matamala, C. (2011). Análisis de rendimiento académico estudiantil usando data warehouse y redes neuronales. Revista a de ingeniería, vol. 19 Nº 3, 2011, pp. 369-381. Descargado desde: http://www.scielo.cl/pdf/ingeniare/v19n3/art07.pdf, el 24/11/2017.

[58] Hernández Cáceresa, J. (2013). Descubrimiento de conocimiento en la base de datos académica de una institución de educación superior usando redes neuronales. Universidad Santo Tomás, Bucaramanga. Descargado desde: http://vip.ucaldas.edu.co/vector/downloads/Vector6_2.pdf, el 24/11/2017.

[59] Madrid Echeverry, J. (2017). Propuesta de un modelo estadístico para caracterizar y predecir la deserción estudiantil Universitaria. Universidad Nacional de Colombia Facultad de Minas, Departamento de Ingeniería de la Organización Medellín, Colombia Descargado desde: http://www.bdigital.unal.edu.co/58059/1/71787491.2017.pdf, el 24/11/2017.

[60] Ahumada, H. y otros. (2015). Minería de datos para un sistema de alerta temprana de deserción en carreras de Ingeniería. (RedUNCI). Descargado desde: http://sedici.unlp.edu.ar/handle/10915/45515, el 24/11/2017.

[61] Ordoñez Briceño, K. (2013). Aplicación de técnicas de minería de datos para predecir la deserción de los estudiantes de primer ciclo de la Modalidad Abierta y a Distancia de la UTPL. Ecuador. Descargado desde: http://dspace.utpl.edu.ec/bitstream/123456789/7897/1/Ordonez%20Brice%C3%B1o%20Karla-%20Informatica.pdf, el 24/11/2017.

[62] Edwards Molina, D. y otros. (2015). Modelo Predictivo para el Análisis de Deserción en Carreras de Ingeniería. Congreso Arg. de Estadística. ISSN 2451-8131
Descargado desde: http://www.cae2015.s-a-e.org.ar/Arch/img_53.pdf, el 24/11/2017.
[63] Edwards Molina, D. y otros. (2015). Predicción del riesgo de abandono universitario utilizando métodos supervisados. IPECyT 2016. Descargado desde:
http://www.frbb.utn.edu.ar/ipecyt2016/, el 24/11/2017.
[64] Lezcano, D. Minería De Datos. Univ Nacional del Nordeste. Descargado desde:
http://exa.unne.edu.ar/informatica/SO/MineriaDatosLezcano.pdf, el 24/11/2017.
[65] García Gutiérrez, J. (2016). Comenzando con Weka: Filtrado y selección de subconjuntos de atributos basada en su relevancia descriptiva para la clase.
Descargado desde: https://www.researchgate.net/publication/308141950, el 24/11/2017.
[66] Rykeboer, H. y otros. (2016). Análisis Comparativo de Modelos de Clasificación de Minería de Datos (Data Mining). Su aplicación en la predicción de perfiles de alumnos en riesgo de deserción. PROINCE C176
[66] Cortez Vásquez, A. (2013). Categorización de Textos mediante Máquinas de Soporte Vector. Universidad Nacional Mayor de San Marcos, ISSN 1816-3823(versión electrónica)
Descargado desde:
http://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/viewFile/5711/4942, el 24/11/2017.
[67] Scholkopf, B. and Smola, A. (2001). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA. ISBN: 0262194759

7. Anexos.

- **Anexo I.** Contenido del archivo *Datos_2013-2014.arff*
  Se describe el contenido que se va a presentar en el Modelo de datos o vista minable.
- **Anexo II.** Detalle de valores de cada clasificador
  Presenta los datos arrojados por Weka, en las distintas fases del entrenamiento y del test.