Universidad del Bío-Bío, Chile

Facultad de Ciencias Empresariales

Departamento de Sistemas de Información

# Generación de un modelo de predicción para la deserción estudiantil en la Universidad del Bío-Bío a través del análisis de diferentes herramientas de predicción analítica.

Tesis presentada por Eduardo Javier Lizama Núñez

Para obtener el título de Ingeniero Civil en Informática

Dirigida por Elizabeth Grandón Toledo

Abril, 2019

**Resumen**

La deserción estudiantil universitaria es un término comúnmente utilizado para definir aquella situación en la que un estudiante después de ingresar a un plan de estudios superior, se retira de éste sin obtener su título. El objetivo de este trabajo es generar un modelo de predicción de deserción estudiantil en la Universidad del Bío-Bío, que pueda identificar las características de los estudiantes con mayor riesgo a desertar. Este trabajo estudió los conceptos y pasos involucrados en la predicción de la deserción estudiantil y realizó una comparación de modelos de clasificación de minería de datos para determinar el modelo más adecuado. Los pasos involucrados en este estudio son: una revisión bibliográfica sobre la deserción, familiarización, preprocesamiento y preparación de los datos, evaluación y comparación de los modelos puestos a prueba. Los algoritmos empleados fueron árboles de decisión, redes neuronales y los softwares utilizados fueron SAP Predictive Analytics, Weka y RapidMiner. Los resultados de la evaluación muestran que la red neuronal y el árbol de decisión entrenado en RapidMiner son los modelos con mejor rendimiento para la predicción de estudiantes desertores con una tasa de sensibilidad de 62% y 21% respectivamente

## Abstract

Student drop out is a term commonly used to define the situation in which a student, after entering a higher education plan, withdraws from it without obtaining a degree. The objective of this work is to generate a prediction model of student desertion at the Universidad del Bío-Bío, which can identify the characteristics of students with higher risk of desertion. This work studies the concepts and steps involved in the prediction of student desertion and conduct a comparison of data mining classification models to determine the most appropriate model. The steps involved in this study are: a literature review on desertion, familiarization, preprocessing and data preparation, evaluation and comparison of the models put to the test. The algorithms used were decision trees, neural networks and the software used were SAP Predictive Analytics, Weka and RapidMiner. The results of the evaluation show that the neural network and the decision tree trained in Rapidminer are the models with the best performance for the prediction of dropout students with a sensitivity rate of 62% and 21% respectively

# Índice

# Capítulo 1: Introducción

## 1.1 Introducción

La deserción universitaria es un fenómeno con repercusiones institucionales, personales y sociales muy relevantes. En el ámbito institucional contribuye a la disminución de índices de calidad y eficiencia; a nivel personal limita las ventajas que trae la educación para el desarrollo del individuo y en el ámbito social aporta a la desigualdad y el aumento del desempleo (IESALC, 2007). Este fenómeno es un problema que está vigente en las instituciones de educación superior chilenas y latinoamericanas. De acuerdo a cifras de la Organización para la Cooperación y el Desarrollo Económicos OECD (2018) en 2016 solo un 49% de los jóvenes estudiantes (incluidos los estudiantes internacionales) pueden esperar graduarse de la educación superior al menos una vez. Por otra parte, en el entorno nacional chileno, según datos del Servicio de Información de Educación Superior SIES (2018) en 2017 las universidades chilenas tenían una tasa de retención en primer año del 78,7%, seguidos por los institutos profesionales (IP) con un 70,9% y por último los centros de formación técnica (CFT) con un 68,7%; lo que se traduce en que aproximadamente un 27,2% de los estudiantes que ingresaron en 2017 a la educación superior chilena desertaron en primer año de su plan de estudio.

Por lo expuesto anteriormente es que la deserción universitaria actualmente es un tema de interés para las instituciones de educación superior, por ende, también para la Universidad del Bío-Bío. Según el Anuario Estadístico 2017, la UBB muestra que la tasa de deserción definitiva de primer año, para la cohorte de ingreso 2017, es de 9,5% y un 16,3% si se suma con la deserción temporal. La deserción temporal se refiere a los estudiantes que solicitan retiro temporal por un máximo de dos semestres consecutivos o tres alternados durante la permanencia en su carrera. Con

el fin de disminuir los índices de deserción la Universidad del Bío-Bío ha visto la necesidad de tener mecanismos que pueda determinar el riesgo de deserción de los estudiantes y con esto desarrollar acciones que aporten a la retención de los estudiantes más propensos a desertar.

La Universidad del Bío-Bío ha desarrollado técnicas de análisis predictivo para prever el riesgo de deserción asociado a los estudiantes. De esta manera la institución ha utilizado datos generados en el proceso de admisión para poder identificar qué características de los estudiantes son más determinantes para la deserción. Dentro de trabajos desarrollados en esta área podemos destacar el Modelo de Deserción de Retamal y Rubilar (2017) que utiliza Regresiones Logísticas para el análisis de la deserción y el trabajo realizado por Pérez et al. (2018) basado en árboles de decisión en el software SAP Predictive Analytics, temas desarrollados en sus respectivas tesis de pregrado.

Considerando los trabajos mencionados anteriormente, este proyecto propone comparar y proponer nuevas técnicas de predicción para deserción estudiantil en la Universidad del Bío-Bío utilizando herramientas de Minería de Datos (MD) con el fin de evaluar qué algoritmos y software son más eficientes para la predicción de este fenómeno estudiantil. A continuación, se procede a definir los objetivos de este proyecto, luego se profundiza en conceptos relacionados con la deserción estudiantil y minería de datos y, por último, luego de la comparación de modelos se propone un nuevo modelo de predicción para la deserción universitaria en la Universidad del Bío Bío.

## 1.2 Trabajos Relacionados

A continuación, se describen tres trabajos similares a este donde se ha estudiado la deserción con minería de datos:

Fischer (2012) en su trabajo "Modelo para la automatización del proceso de determinación de riesgo de deserción en estudiantes universitarios" propone una metodología que permita identificar en forma automática a los estudiantes con mayor riesgo de deserción en las carreras de ingeniería en la Universidad de las Américas. Para la implementación de este proyecto se adoptó la metodología CRISP-DM y además se utilizó el módulo de Analysis Services de SQL Server 2008. Se aplicaron los modelos de redes neuronales, árbol de decisión y Cluster K-mediana para analizar el comportamiento de los estudiantes, evaluando factores como el puntaje promedio obtenido en la PSU, el promedio de notas obtenido en la enseñanza media, la edad a la fecha de ingreso a la institución y género de los estudiantes. La exactitud de los modelos fue calculada a partir de un conjunto de datos de pruebas, los cuales indicaron que ninguno de los modelos evaluados arrojó resultados positivos, debido a esto se analizó el proceso y se llegó a la conclusión que es muy probable que los datos de entrada no eran suficientemente confiables.

Con minería de datos, Dekker, Pechenizkiy, & Vleeshouwers (2009) estudiaron datos educativos destinados a predecir la deserción estudiantil en estudiantes de Ingeniería Eléctrica de la Universidad Tecnológica de Eindhoven. Para el desarrollo de este trabajo se utilizaron árbol de decisión y sus resultados experimentales muestran que los árboles decisión dan un resultado útil con precisiones entre 75 y 80% que es difícil de superar con otros modelos más sofisticados.

Yukselturk (2014) examinó la predicción de abandonos a través de enfoques de minería de datos en un programa en línea. El tema del estudio se seleccionó de un total de 189 estudiantes que

se inscribieron en el Programa de Certificación de Tecnologías de la Información en línea en 2007-2009. Los datos se recopilaron a través de cuestionarios en línea (Encuesta demográfica, Escala de autoeficacia de tecnologías en línea, Cuestionario de preparación para el aprendizaje en línea, Escala de locus de control y Cuestionario de conocimientos previos). Los datos recopilados incluyeron 10 variables, que fueron género, edad, nivel educativo, experiencia previa en línea, ocupación, autoeficacia, preparación, conocimiento previo, locus de control y el estado de abandono como la etiqueta de la clase (abandono / no). Para clasificar a los estudiantes que abandonaron la escuela, se aplicaron cuatro métodos de minería de datos basados en k-Nearest Neighbor (k-NN), Decision Tree (DT), Naive Bayes (NB) y Neural Network (NN). Estos métodos fueron entrenados y probados utilizando 10 veces la validación cruzada. Las sensibilidades de detección de los clasificadores 3-NN, DT, NN y NB fueron del 87%, 79.7%, 76.8% y 73.9% respectivamente. Además, el uso del método de selección de características basado en el algoritmo genético (GA), la autoeficacia de las tecnologías en línea, la preparación para el aprendizaje en línea y la experiencia en línea previa se encontraron como los factores más importantes para predecir los abandonos.

Pérez et al. (2018) propone un modelo de predicción que permite determinar los estudiantes con mayor riesgo de desertar en la Universidad del Bío-Bío. Para el desarrollo de este modelo se utilizó la metodología KDD y además se utilizó el software SAP Predictive Analytics. Se entrena un árbol de decisión para analizar el comportamiento de los estudiantes, evaluando atributos como carrera, puntaje PSU matemáticas, NEM, orden de postulación, becas de arancel MINEDUC, domicilio universitario, motivación a trabajar, género, comuna de domicilio y gratuidad. El desempeño del modelo se mide evaluando los índices de la matriz de confusión resultante. Como

resultado se obtuvo que el árbol de decisión entrenado puede predecir los estudiantes desertores con un 87% de sensibilidad y una precisión de 97%.

### 1.3 Objetivo general

- Generar un modelo de predicción para la deserción estudiantil en la Universidad del Bío-Bío a través del análisis de diferentes herramientas de predicción analítica.

### 1.4 Objetivos específicos

- Realizar revisión literaria sobre deserción estudiantil.
- Estudiar modelos actuales para predicción de deserción estudiantil en la Universidad del Bío-Bío
- Estudiar algoritmos de predicción aplicables para el caso de la deserción estudiantil en la Universidad del Bío-Bío.
- Implementar modelos de predicción en diferentes herramientas analíticas.
- Comparar resultados de los modelos generados.
- Proponer un nuevo modelo de predicción basado en la comparación efectuada.

### 1.3 Alcance y limites

El presente proyecto está circunscrito en la Universidad del Bío-Bío, incluyendo todas sus sedes distribuidas en las ciudades de Concepción y Chillán. Se aplican técnicas de minería de datos con modelos de árboles de decisión y redes neuronales con datos de admisión comprendiendo el periodo de ingreso 2014-2015. Se utilizan los sofware SAP Predictive Analytics, Weka y Rapidminer; que contienen las características y herramientas necesarias de minería de datos, con los algoritmos requeridos para esta investigación.

# Capítulo 2: Marco Teórico

Para comprender mejor el desarrollo de esta investigación será necesario plantear y estudiar algunas ideas que sirvan como ejes conceptuales para el desarrollo del análisis. Por lo anterior este capítulo desarrolla la teoría que va a sustentar esta investigación y fue construido en base a una revisión bibliográfica de la literatura de distintas bases de datos. A continuación, se procede a describir el proceso de revisión bibliográfica, la conceptualización de elementos relacionados a la deserción estudiantil, modelos aplicables al estudio y herramientas que son de utilidad para la construcción de los modelos de predicción.

## 2.1 Descripción de la problemática

Para la Universidad del Bío-Bío la deserción estudiantil se ha convertido en un factor importante de estudio a causa del número considerable de estudiantes que desertan temporal o definitivamente de sus programas académicos. Esto conlleva a un gran costo tanto para la institución como para los estudiantes, ya que las instituciones disminuyen sus índices de calidad y eficiencia y en los estudiantes puede generar gran desigualdad respecto de oportunidades y su desarrollo académico (IESALC, 2007).

Como muestra la Ilustración 1, la tasa de deserción entre los años 2013-2017 alcanza un promedio de aproximadamente un 15% y tiende a aumentar. Es por esto que la institución ha tomado acciones respecto de este tema implementando distintos programas de nivelación y acompañamiento a los estudiantes de primer año. Uno de los programas que ha sido más relevante en la mitigación de la deserción es el Programa de Tutores, el cual genera un acompañamiento en los estudiantes que cursan su primer año en la universidad mejorando su rendimiento académico y su desarrollo universitario.

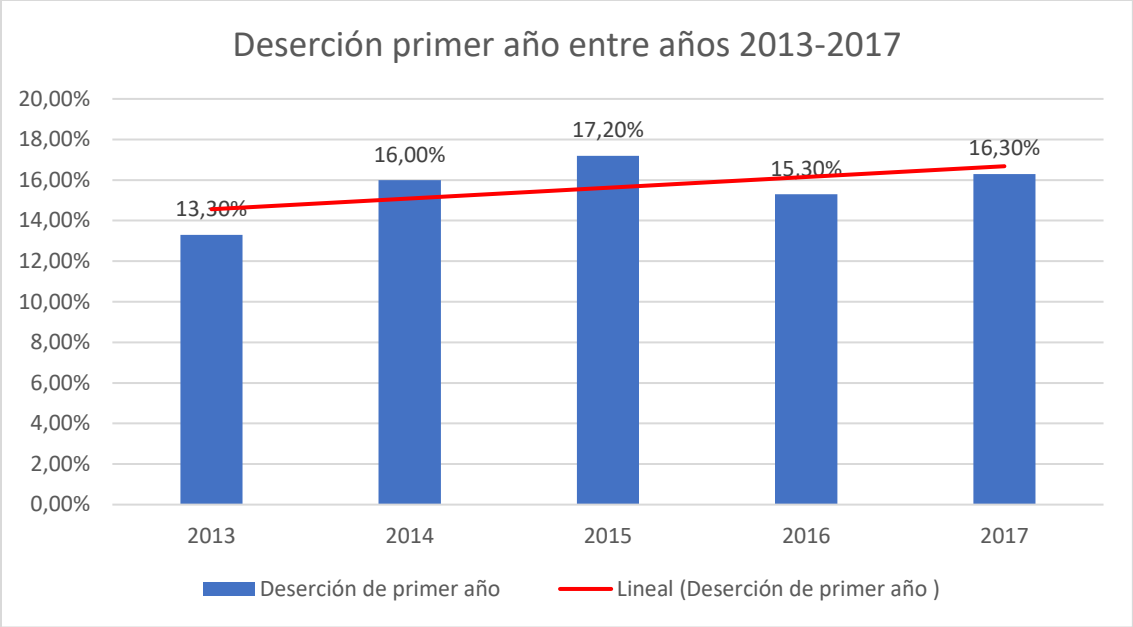


*Ilustración 1 Tasa de deserción en la Universidad del Bío-Bío. Fuente: anuarios estadísticos UBB.*

Como se puede apreciar en la Tabla 1, la deserción se comporta de distintas maneras en cada una de las facultades de la universidad, donde la Facultad de Ciencias tiene los mayores índices de deserción alcanzando un 31,2% en 2014 y la Facultad de Ciencias de la Salud y de los Alimentos con los índices más bajos alcanzando un 9,8% en 2017.

*Tabla 1 Deserción por facultad entre los años 2013-2017. Fuente: anuarios académicos UBB.*

| Facultad | 2013 | 2014 | 2015 | 2016 | 2017 |
|---|---|---|---|---|---|
| Facultad de Arquitectura, Construcción y Diseño | 19,5% | 26,7% | 23,9% | 16,9% | 17,2% |
| Facultad de Ciencias | 21,5% | 31,2% | 28,9% | 26,3% | 30,3% |
| Facultad de Educación y Humanidades | 12,3% | 16,2% | 14,1% | 10,7% | 12,7% |
| Facultad de Ingeniería | 13,4% | 14,6% | 21,5% | 16,4% | 20,7% |
| Facultad de Ciencias de la Salud y de los Alimentos | 11,1% | 10,5% | 11,7% | 15,7% | 9,8% |
| Facultad de Ciencias Empresariales | 12,8% | 13,4% | 12,2% | 15,4% | 12,8% |

De acuerdo al último anuario estadístico realizado en 2017, la tasa de deserción definitiva de primer año para los estudiantes matriculados en el mismo año es de 9,5% y si adicionalmente

se considera los estudiantes que se retiran temporalmente, los estudiantes desertores llegarían a un 16,3%. La Tabla 2 presenta la tasa de deserción y retención para la cohorte de ingreso 2017.

*Tabla 2 Retención y deserción de estudiantes de primer año. Fuente: Anuario estadístico 2017 UBB.*

| **Total, Cohorte 2017** | **Retención** | | **Deserción** | | | | |
|---|---|---|---|---|---|---|---|
| | Matriculados 2018 | | Temporal | | Definitiva | | Global |
| | N° | % | N° | % | N° | % | % |
| 2432 | 2043 | 83.6% | 167 | 6.9% | 230 | 9.5% | 16.3% |

Analizando los índices de los últimos años y la tendencia al aumento que tiene la deserción en la UBB, se hace necesario desarrollar continuamente herramientas que permitan mejorar los índices de retención de estudiantes, sobre todo en las facultades que tienen los índices más elevados. Para esto, la universidad ha orientado sus esfuerzos a la prevención de la deserción mediante modelos que sean capaces de determinar qué estudiantes tienen más posibilidades de abandonar sus planes de estudios.

Siguiendo la misma línea de los modelos mencionados anteriormente, esta investigación busca generar un modelo que sea más preciso a la hora de predecir la deserción estudiantil, en base a la comparación de distintos algoritmos y software de minería de datos.

## 2.2 Revisión de la bibliografía

A continuación, se presenta una revisión bibliográfica del fenómeno de la deserción estudiantil en la educación superior y el desarrollo de herramientas que permiten estudiar esta variable.

### 2.2.1 Deserción estudiantil universitaria

Para Tinto (1982) la deserción universitaria se define como el fracaso para alcanzar una meta deseada en pos de la cual un sujeto ingresó a una institución de educación superior. De forma similar, Himmel (2002) asevera que la deserción estudiantil se refiere al abandono prematuro de un programa de estudios antes de alcanzar el título, y además, considera un tiempo suficientemente largo como para descartar la posibilidad de que el estudiante se reincorpore. Sin embargo, se pueden reconocer dos tipos de abandonos en los estudiantes de educación superior: respecto al tiempo y al espacio (Castaño, Gallón, Gómez, & Vásquez, 2004).

Con respecto al tiempo, se clasifican de la siguiente manera:

1. Deserción precoz: cuando el estudiante que, habiendo sido admitido en la institución, no se matricula.
2. Deserción temprana: cuando el estudiante abandona sus estudios en los cuatro primeros semestres de la carrera.
3. Deserción tardía: cuando el estudiante abandona los estudios en los últimos seis semestres de su carrera.

Por otra parte, la deserción con respecto al espacio se divide en:

1. Deserción interna: se refiere al estudiante que cambia su programa académico que ofrece la misma institución universitaria.
2. Deserción institucional: es el caso en el que el estudiante abandona la universidad para matricularse en otra.
3. Deserción total del sistema educativo: es el caso en el que el estudiante abandona definitivamente sus estudios.

Considerando que existen variadas definiciones acerca de la deserción estudiantil, Tinto (1989) afirma que ninguna definición de deserción logra capturar totalmente este fenómeno, por lo que deja en manos de los investigadores la elección de la definición que mejor se ajusta a la investigación que lleven a cabo. Apoyando esta idea Simpson (2004) afirma que el estudio de la deserción de estudiantes universitarios es muy complejo y que ninguna definición de deserción logra capturar completamente la complejidad del problema.

Según Ramírez & Grandón (2018), el fenómeno de la deserción universitaria se ha estudiado principalmente en dos líneas de investigación. En una de ellas se encuentran los trabajos relacionados a modelos teóricos, que explican los factores que influyen en la intención de desertar de los programas académicos. En la otra línea de investigación encontramos trabajos que han estudiado el fenómeno de la deserción con el uso de herramientas de minería de datos a través de modelos analíticos descriptivos o predictivos. Algunos de estos últimos son desarrollados en la sección 1.2 Trabajos relacionados.

Por el lado de los modelos teóricos, Fishbein & Ajzen (1975) propone que las intenciones de una persona son el resultado de sus creencias, las que son determinantes sobre su forma de actuar y llevan a expresar un comportamiento. Por ende, la decisión de desertar o persistir en un programa de estudios se ve influida por conductas anteriores. Ethington (1990) propone un modelo más completo, basándose en los anteriores e incorporando una teoría más general sobre las "conductas de logro", las cuales comprenden atributos de perseverancia, la elección y el desempeño. Spady (1970) enfatiza no solo en los factores psicológicos, sino que también en la influencia de factores externos al individuo en la retención. Tinto (1975) propone un modelo que a medida que el alumno evoluciona a través del programa de estudios, diversas variables contribuyen a reforzar su adaptación dentro de la institución, ya que al momento de ingresar

comienza con un conjunto de características que influyen sobre su experiencia en la educación universitaria. Bean (1985), en base a la teoría de Tinto (1975), propuso un modelo integrado que explica la deserción. Entre los factores influyentes en la deserción, Bean considera los factores académicos, psicosociales, ambientales y de socialización.

### 2.2.2 Knowledge Discovery in Databases

En la literatura actual se puede encontrar un gran número de definiciones acerca del descubrimiento de conocimiento en bases de datos (Knowledge discovery in databases - KDD). Una de las definiciones 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)" (Fayyad, 1997). El KDD es un proceso centrado en el usuario, que tiene la propiedad de ser altamente interactivo, y que debe ser guiado por las decisiones que toma el usuario, o también por un agente inteligente (Nigro, Xodo, Corti, & Terren, s. f.).

El proceso general de búsqueda e interpretación de patrones a partir de datos implica la aplicación repetida de los siguientes pasos (M. Fayyad, Piatetsky-Shapiro, & Smyth, 1996):

1. Desarrollando un entendimiento de
    - el dominio de la aplicación.
    - el conocimiento previo relevante.
    - los objetivos del usuario final.

2. Creación de un conjunto de datos de destino: selección de un conjunto de datos, o enfoque en un subconjunto de variables, o muestras de datos, en el que se realizará el descubrimiento.
3. Limpieza y preprocesamiento de datos.
    - Eliminación de ruidos o valores atípicos.
    - Recopilación de información necesaria para modelar o tener en cuenta el ruido.
    - Estrategias para el manejo de campos de datos faltantes.
    - Contabilización de la información de secuencia de tiempo y cambios conocidos.
4. Reducción y proyección de datos.
    - Encontrar características útiles para representar los datos en función del objetivo de la tarea.
    - Usar métodos de reducción o transformación de dimensionalidad para reducir el número efectivo de variables bajo consideración o para encontrar representaciones invariantes para los datos.
5. Elegir la tarea de minería de datos.
    - Decidir si el objetivo del proceso KDD es clasificación, regresión, agrupación, etc.

La Minería de Datos es un campo de la estadística que intenta obtener patrones o comportamientos a partir de los datos almacenados en bases de datos (Maimon & Rokach, 2005). Estas técnicas constituyen al 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. Asimismo, es preciso entender los parámetros y las características de los algoritmos para preparar los datos a analizar. Decidir si los modelos resultantes son de utilidad o no suele requerir una validación subjetiva del investigador. Según Weiss & Indurkhya (1998) las técnicas de minería de datos se clasifican en dos grandes categorías: supervisadas o predictivas y no supervisadas o descriptivas (véase Tabla 3).

*Tabla 3 Clasificación Técnicas de Minería de datos.*

| | Aprendizaje | Técnica | Algoritmos |
|---|---|---|---|
| **Técnicas de MD** | No supervisadas | Clustering | Numérico<br>Conceptual<br>Probabilístico |
| | | Asociación | A Priori |
| | Supervisadas | Predicción | Regresión<br>Árbol de Predicción<br>Estimador de Núcleos |
| | | Clasificación | Tabla de decisión<br>Árbol de decisión<br>Inducción de Reglas<br>Bayesiana<br>Basado en Ejemplares<br>Redes Neuronales<br>Lógica Borrosa<br>Técnicas Genéticas<br>Regresión logística |

Las predicciones son utilizadas para predecir comportamientos futuros de algún tipo de variable mientras que una descripción puede ayudar a su comprensión. Sin embargo, los modelos predictivos pueden ser descriptivos (hasta donde sean comprensibles por personas) y los modelos descriptivos pueden emplearse para realizar predicciones. De esta forma, hay algoritmos o técnicas que pueden servir para distintos propósitos, por lo que la figura anterior únicamente representa para qué propósito son más utilizadas las técnicas (Molina & García, 2006). Por ejemplo,

las redes neuronales pueden servir para predicción, clasificación e incluso para aprendizaje no supervisado.

6. Elegir los algoritmos de minería de datos.
    - Selección de los métodos que se utilizarán para buscar patrones en los datos.
    - Decidir qué modelos y parámetros pueden ser apropiados.
    - Hacer coincidir un método particular de minería de datos con los criterios generales del proceso KDD.
7. Minería de datos.
    - Buscar patrones de interés en una forma de representación particular o en un conjunto de representaciones tales como reglas de clasificación o árboles, regresión, agrupamiento, etc.
8. Interpretando patrones minados.
9. Consolidando el conocimiento descubierto

### 2.2.3 Algoritmos de minería de datos

Con el fin de analizar los trabajos de Retamal & Rubilar (2017) y Pérez et al. (2018), en esta sección se describen los algoritmos de regresión logística y árboles de decisión. Además, se describen las redes neuronales ya que es un algoritmo que se utiliza en la comparación de los modelos para la predicción de la deserción.

#### *2.2.3.1 Regresión Logística*

Los métodos de regresión son un proceso estadístico que permite estimar relaciones entre una variable de respuesta y una o más variables explicativas (Hosmer, Lemeshow, & Sturdivant,

2013). Generalmente cuando se quiere poner una variable en función de otra(s), se acude al bien conocido recurso de la regresión lineal (simple o múltiple). Esta función utiliza normalmente el método de mínimos cuadrados y funciona fluidamente desde el punto de vista aritmético. Cuando la variable a explicar sólo puede tomar dos valores, es decir, la ocurrencia o no de un cierto proceso, al evaluar la función para valores específicos de las variables independientes se obtendrá un número que será diferente de 1 y de 0 (los valores posibles de la variable dependiente), lo cual carece de todo sentido. En este caso, la regresión lineal debe ser descartada, en cambio la regresión logística se ajusta adecuadamente a esta situación (Fiuza & Rodríguez, 2000).

Según Salas (1996) los objetivos del modelo de regresión logística son, principalmente, tres: (i) determinar la existencia o ausencia de relación entre una o más variables independientes (Xi) y una variable dependiente dicotómica (Y), es decir, que solo admite dos categorías que definen opciones o características mutuamente excluyentes u opuestas. Las variables independientes pueden ser cualitativas binarias o categóricas, y cuantitativas o continuas. (ii) Medir el signo de dicha relación, en caso de que exista y (iii) estimar o predecir la posibilidad de que se produzca el suceso o acontecimiento definido como "Y=1" en función de los valores que adoptan las variables independientes.

#### *2.2.3.2 Árbol de decisión*

Según Rokach & Maimon (2008) los árboles de decisión son simples pero exitosos para predecir y explicar la relación entre variables independientes y una variable objetivo. Además, el autor afirma que son herramientas muy eficaces en áreas como la minería de textos, la extracción de información, el aprendizaje automático y el reconocimiento de patrones. Para Hernández, Ramírez & Ferri (2004) los árboles de decisión son una técnica de minería de datos que establecen un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión

final a tomar se puede determinar siguiendo condiciones que se cumplen desde la raíz del árbol, llegando hasta alguna de sus hojas. Hernández explica que los árboles de decisión crecen a través de la división iterativa de grupos discretos, donde la meta es maximizar la "distancia" entre grupos por cada división. Una de las distinciones entre los diferentes métodos de "división" es cómo miden estas distancias. Se puede pensar que cada división de los datos en nuevos grupos debe ser diferente uno a otro tanto como sea posible. Este método es llamado "purificación de grupos" (Hernández et al., 2004). Para predecir variables del tipo categóricas los árboles son llamados árbol de clasificación, y los árboles usados para predecir variables del tipo continuas son llamados árboles de regresión. Los árboles de decisión manejan datos no numéricos muy bien.

En la Ilustración 2 se puede apreciar un ejemplo de árbol de decisión para la clasificación de posibles muertes hospitalaria en hombres, el cual fue el resultado de un modelo entrenado con 7 variables que son: paciente en shock, edad, fibrilación ventricular, edad, insuficiencia cardiaca, sexo, accidente cerebro vascular e insuficiencia renal. Al igual que como se explica en el párrafo anteriores, este modelo puede clasificar posibles mortalidades si se siguen sus condiciones desde la raíz, un ejemplo de esto podría ser el siguiente. Si un hombre no presenta shock, y tiene menos de 68,5 años y fibrilación ventricular, el modelo le asigna una probabilidad de mortalidad hospitalaria del 19,75%.

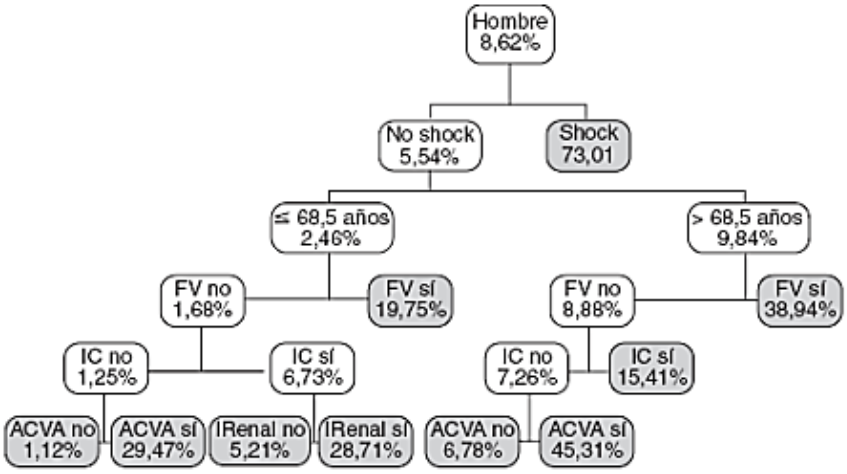


*Ilustración 2 Ejemplo de árbol de decisión, caso "Mortalidad hospitalaria del infarto agudo de miocardio caso Hombres". Fuente:*(Trujillano et al., 2008)

#### *2.2.3.5 Redes Neuronales*

Las Redes Neuronales Artificiales o simplemente Redes Neuronales son una técnica de minería de datos para resolver problemas de forma individual o combinadas con otros métodos, para aquellas tareas de clasificación, identificación, diagnóstico, optimización o predicción en las que el balance datos/conocimiento se inclina hacia los datos y donde, adicionalmente, puede haber la necesidad de aprendizaje en tiempo de ejecución y de cierta tolerancia a fallos (Salas, 2017). Según Izaurieta & Saavedra (2019) las redes neuronales, están motivadas en imitar la forma de procesamiento de la información en sistemas nerviosos biológicos. Las redes neuronales están conformadas por nodos interconectadas y arregladas en tres capas. Los datos ingresan por medio de la "capa de entrada", pasan a través de la "capa oculta" y salen por la "capa de salida. Cabe mencionar que la capa oculta se puede conformar por varias capas.

En la Ilustración 3, la capa de entrada está constituida por aquellos nodos que introducen los patrones de entrada en a la red. En estos nodos no se produce procesamiento. Las capas ocultas están formadas por aquellos nodos cuyas entradas provienen de capas anteriores y cuyas salidas

pasan a nodos de capas posteriores. Por último, la capa de salida está compuesta de nodos cuyos valores de salida corresponde con las salidas de toda la red.

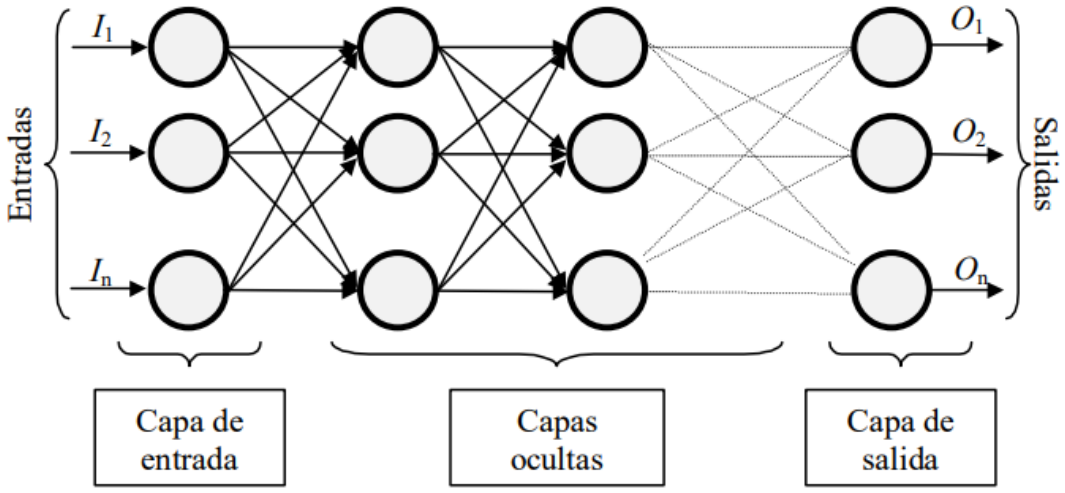


*Ilustración 3 Ejemplo red neuronal. Fuente:* Matich (2001)

Matich (2001) afirma que las redes neuronales son sistemas dinámicos autoadaptativos. Son adaptables debido a la capacidad de autoajuste de los elementos procesales que componen el sistema. Son dinámicos, pues son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones. En el proceso de aprendizaje, los enlaces ponderados de los nodos se ajustan de manera que se obtengan ciertos resultados específicos. Una red neuronal no necesita un algoritmo para resolver un problema, ya que ella puede generar su propia distribución de pesos en los enlaces mediante el aprendizaje. También existen redes que continúan aprendiendo a lo largo de su vida, después de completado su período de entrenamiento.

### 2.2.4 Softwares de minería de datos

A continuación, se revisarán tres software de minería de datos, los cuales se utilizaron para desarrollar este trabajo: Weka, SAP Predictive Analytics y RapidMiner.

#### *2.2.4.1 Weka*

El entorno de Waikato para el análisis del conocimiento (Weka por su sigla en inglés) es un conjunto de bibliotecas de clases Java que implementan una variedad de algoritmos de extracción de datos y aprendizaje automático de última generación (Witten et al., 1999). Weka está disponible gratuitamente en la web y viene acompañado de un texto sobre minería de datos que documenta y explica completamente todos los algoritmos que contiene. Las aplicaciones escritas con las bibliotecas de clase Weka pueden ejecutarse en cualquier computadora con capacidad de navegación web. Esto permite a los usuarios aplicar técnicas de aprendizaje automático a sus propios datos, independientemente de la plataforma de la computadora (Witten et al., 1999).

##### *2.2.4.1.1 Formato. ARFF*

Para la importación de datos Weka trabaja nativamente con un formato denominado arff, acrónimo de Attribute-Relation File Format. Este formato está compuesto por una estructura diferenciada en tres partes:

1. Cabecera. Se define el nombre de la relación. Su formato es el siguiente:

@relation <nombre-de-la-relación>

Donde <nombre-de-la-relación> es de tipo String. Si dicho nombre contiene algún espacio es necesario expresarlo entre comillas.

2. Declaraciones de atributos. En esta sección se declaran los atributos que componen el archivo junto a su tipo. La sintaxis es la siguiente:

@attribute <nombre-del-atributo> <tipo>

Donde <nombre-del-atributo> es de tipo String teniendo las mismas restricciones que el caso anterior. Weka acepta diversos tipos, estos son:

a. NUMERIC: expresa números reales.
b. INTEGER: expresa números enteros
c. DATE: expresa fechas, para ello este tipo debe ir precedido de una etiqueta de formato entre comillas. La etiqueta de formato está compuesta por caracteres separadores (guiones y/o espacios) y unidades de tiempo:

- dd Día.
- MM Mes.
- yyyy AÑO.
- HH Horas.
- mm Minutos.
- ss Segundos.

d. STRING
e. NOMINAL El identificador de este tipo consiste en expresar entre llaves y separados por comas los posibles valores (caracteres o cadenas de caracteres) que puede tomar el atributo: Por ejemplo: @attribute tiempo {soleado,lluvioso,nublado}

3. Sección de datos. Se declaran los datos que componen la relación separando entre comas los atributos y con saltos de línea las relaciones.

*2.2.4.1.2 Interfaces de Weka*

A continuación, se procede a explicar cada una de las interfaces que provee WEKA (véase ilustración 4).

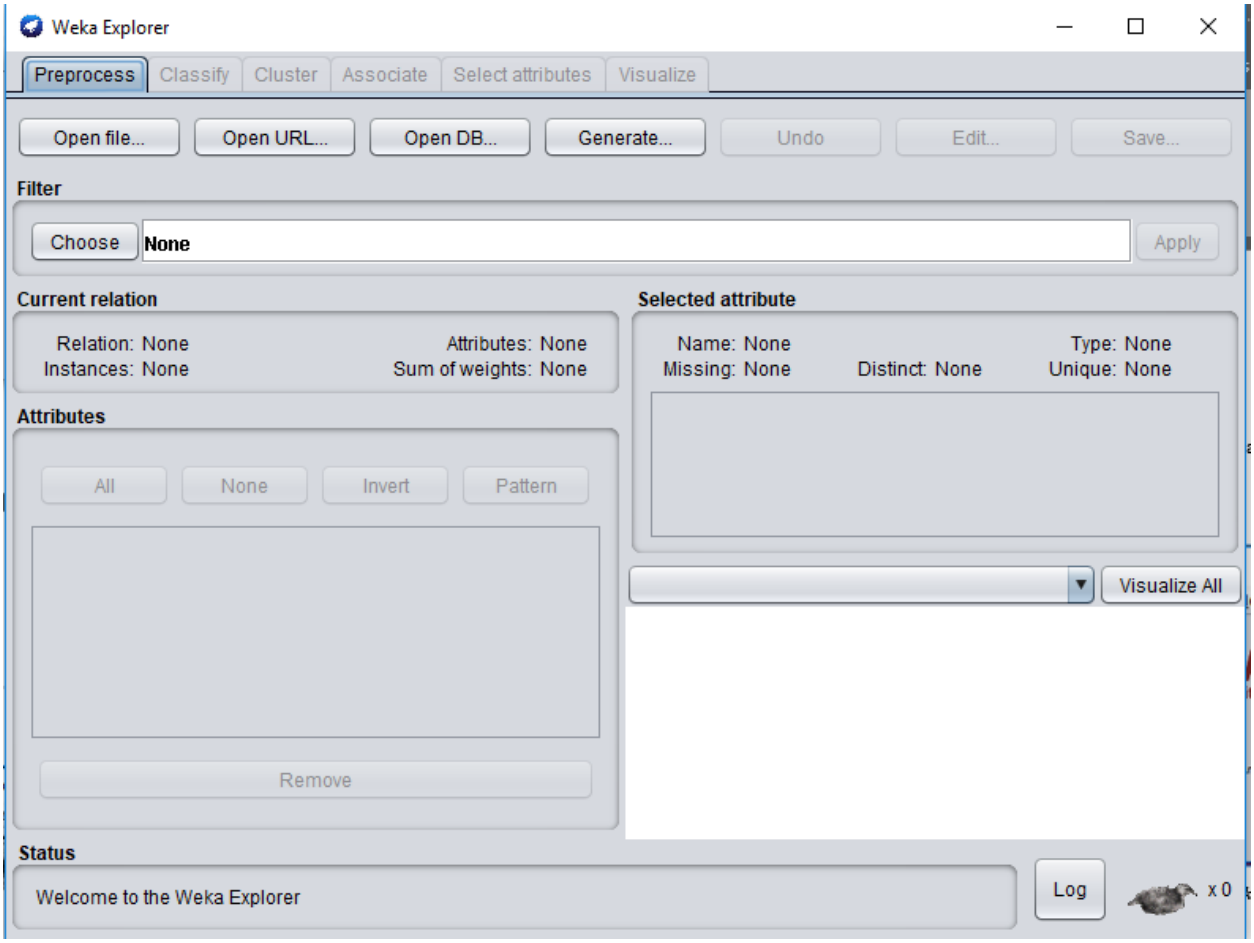


*Ilustración 4 Weka Explorer.*

- 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.
- Explorer: Es el modo más usado y más descriptivo. Éste permite realizar operaciones sobre un sólo archivo de datos. El explorador permite tareas de:
    - 1. Preprocesado de los datos y aplicación de filtros.
    - 2. Clasificación.

    - 3. Clustering.
    - 4. Búsqueda de Asociaciones.
    - 5. Selección de atributos.
    - 6. Visualización 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: Esta última interface de Weka es quizá la más cuidada y la que muestra de una forma más explícita el funcionamiento interno del programa. Su funcionamiento es gráfico y se basa en situar en el panel de trabajo, elementos base de manera que creemos un "circuito" que defina nuestro experimento.

*2.2.4.1.3 Algoritmos Weka*

Algunos algoritmos de Weka tienen nombres no estándar, por lo que es posible que exista confusión con sus nombres en el software. A continuación, se incluye una lista de los 7 principales algoritmos de aprendizaje automático disponibles en el entorno Weka, por su nombre estándar y su nombre utilizado en Weka.

- Regresión lineal: function.LinearRegression
- Regresión Logística: function.Logistic
- Naive Bayes: bayes.NaiveBayes
- Árbol de decisión: trees.J48
- K-vecinos más cercanos: lazy.IBk
- Máquinas de vectores de soporte: functions.SMO
- Red Neuronal: functions.MultilayerPerceptron

#### *2.2.4.2 SAP Predictive Analytics*

SAP Predictive Analytics es un software de pago y permite hacer inteligencia empresarial desarrollado por la empresa SAP. Está diseñado para permitir a las organizaciones analizar grandes conjuntos de datos y predecir futuros resultados y comportamientos. Este software puede ayudar a dar sentido a grandes almacenes de datos mediante la creación de modelos de análisis predictivo para identificar oportunidades imprevistas, comprender mejor los clientes y descubrir riesgos ocultos. SAP PA trabaja bajo una herramienta de código abierto denominado R. R es un entorno y lenguaje de programación orientado a objetos que proporciona un amplio abanico de herramientas estadísticas y gráficas.

SAP Predictive Analytics se lanzó por primera vez en 2015 y combina las características de dos productos de análisis previamente separados. El primero de ellos es SAP Predictive Analysis (nota: no Predictive Analytics), que SAP lanzó en 2012. Es una herramienta de análisis avanzada principalmente para científicos de datos que automatizan el análisis manual. Con el análisis predictivo, los usuarios pueden analizar y visualizar datos a través de algoritmos predefinidos escritos en el lenguaje de programación estadística de código abierto, y luego encadenar los modelos gráficamente para realizar un análisis complejo. La segunda parte de SAP Predictive Analytics es SAP InfiniteInsight, un producto que SAP absorbió cuando adquirió la empresa KXEN en 2013. InfiniteInsight automatiza la preparación de datos, el modelado predictivo y la calificación, lo que puede ayudar a los usuarios comerciales a analizar datos sin el modelado manual. Los algoritmos de InfiniteInsight se encargan de la preparación de datos y el trabajo de modelado que realizan los científicos de datos (Rouse, 2016).

SAP PA tiene dos modos de uso, uno de ellos es el Automated Analytics que permite al usuario realizar minería de datos de forma automática, dejando al software orientar las variables y

modificaciones del algoritmo. El otro modo es el Experts Analytics el cual también realiza minería de datos, pero de una forma más compleja, manual y extensible. Manipula los tipos de datos y los modelos entrenados.

*2.2.4.2.1 Importación de datos SAP PA*

SAP PA permite importar set de datos de variadas maneras y variados formatos. La importación de datos se puede hacer desde archivos guardados localmente, bases de datos SQL y desde base de datos SAP. A continuación, se explican las maneras de importación de datos de SAP.

- Tabla de datos Microsoft: Cargar una hoja de trabajo Excel como conjunto de datos, donde las filas son los datos y las columnas los atributos, considerando que la primera fila contiene los nombres de los atributos.
- Texto: Cargar un archivo de texto (.csv, .txt, .log, .prn, .tsv) como conjunto de datos
- Copiar datos temporalmente almacenados en el portapapeles que contengan un formato de filas y columnas con datos y atributos respectivamente
- Conectarse a SAP HANA: Conectarse a una URL de servidor de SAP HANA para visualizar y crear datos.
- Consulta con SQL: Ejecutar Freehand SQL en una base de datos para descargar un conjunto de datos.

*2.2.4.2.2 Interfaces SAP PA*

Como se puede ver en la ilustración 5, SAP PA proporciona una interfaz de usuario bastante amigable. Este software pone especial énfasis en la simplicidad a la hora de construir un proceso para análisis de datos, ya que cada uno de sus operadores se puede arrastrar al panel de trabajo

para ir construyendo el proceso. A continuación, se explica cada uno de los componentes del panel de trabajo de SAP Predictive Analytics.

Preparar: Esta pestaña permite visualizar los datos previamente importados para verificar que sean correctos, de lo contrario permite modificar los tipos de atributos y visualizar la cantidad de datos importados.

Predecir: Esta pestaña es una de las más importantes, ya que aquí se lleva a cabo la construcción, entrenamiento y evaluación de los modelos. Para construir el proceso de análisis, esta pestaña permite diseñarlo simplemente arrastrando elementos desde su barra lateral donde se encuentran todos los algoritmos, operadores para la preparación de datos, escritores de datos y modelos previamente guardados. Por último, en el botón Resultados se pueden visualizar todos los resultados de los modelos entrenados.

Visualizar: Permite generar gráficas con los distintos atributos, con el fin de poder ver información escondida en los datos.

Crear: Esta pestaña sirve para crear infografías incorporando todos los elementos de análisis realizados.

Compartir: Es utilizada para publicar o enviar por correo análisis realizados.

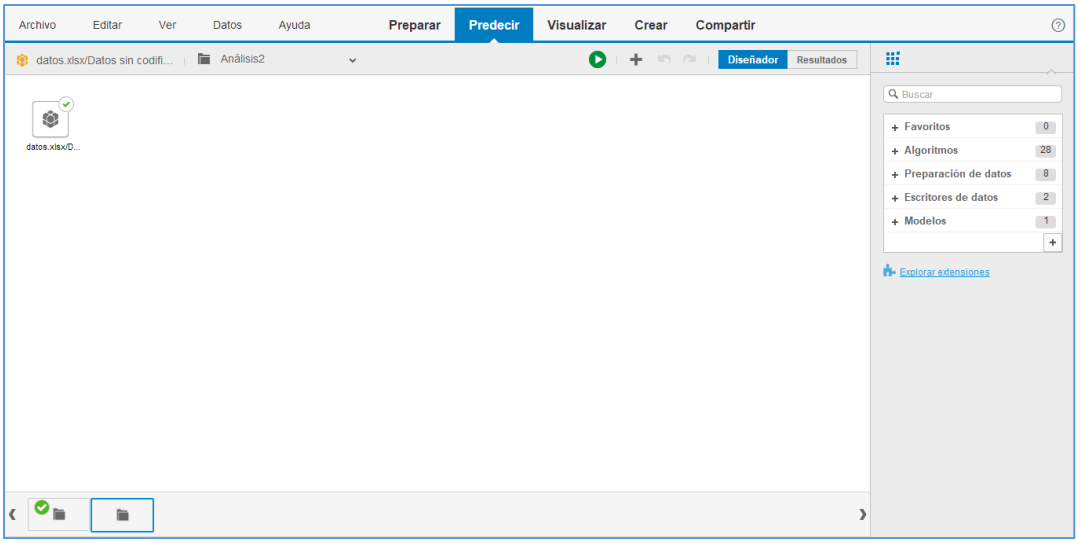


*Ilustración 5 Panel de diseño SAP PA.*

*2.2.4.2.3 Algoritmos SAP PA*

SAP PA ofrece una gran cantidad de algoritmos para el análisis de datos, a continuación, en la Tabla 4 se muestran cada uno de ellos y su funcionalidad.

*Tabla 4 Algoritmos de SAP PA, clasificados por función*.

| Función | Algoritmo |
|---|---|
| Realizar predicciones basadas en tiempo | Algoritmos de series temporales<br>Suavizado exponencial sencillo<br>Suavizado exponencial doble<br>Suavizado exponencial triple |
| Predicción de variables continuas basadas en otras<br>variables del conjunto de datos<br>Relaciones, causa y efecto (Correlación y Regresión) | Algoritmos de regresión<br>Regresión lineal<br>Regresión exponencial<br>Regresión geométrica<br>Regresión algorítmica<br>Regresión lineal múltiple<br>Regresión polinómica<br>Regresión logística<br>Exponential Smoothing |
| Localizar patrones frecuentes de conjuntos de elementos en grandes conjuntos de datos transaccionales para generar reglas de asociaciones. | Algoritmos de asociación<br>Apriori<br>AprioriLite |
| Agrupar observaciones en grupos de conjuntos de datos similares (Análisis de Cluster) | Algoritmos de agrupación en clúster K-Means<br>ABC Análisis |
| Clasificar y predecir una o más variables discretas sobre la base de otras variables del conjunto de datos | Árboles de decisiones<br>HANA C 4.5<br>Árbol CNR R (CART)<br>CHAID |
| Detectar valores atípicos del conjunto de datos | Algoritmos de detección de valores atípicos<br>Rango intercuartil<br>Valor atípico de vecino más próximo<br>Detección de anomalías<br>Prueba de varianza |
| Predicción, clasificación y reconocimiento de patrón estadístico | Algoritmos de red neuronal<br>Red neuronal NNet R<br>Red neuronal MONMLP R |

#### *2.2.4.3 RapidMiner*

RapidMiner (RM) es un entorno para experimentos de aprendizaje automático y minería de datos que respalda el paradigma de creación rápida de prototipos. Los experimentos pueden estar compuestos por un gran número de operadores anidables arbitrariamente y su configuración se describe mediante archivos XML que se pueden crear fácilmente con una interfaz gráfica de usuario. Las aplicaciones de RM cubren tanto la investigación como las tareas de minería de datos en el mundo real (Mierswa, Wurst, Klinkenberg, Scholz, & Euler, 2006). RapidMiner proporciona más de 500 operadores orientados al análisis de datos, incluyendo los necesarios para realizar operaciones de entrada y salida, preprocesamiento de datos y visualización. También permite utilizar los algoritmos incluidos en Weka.

##### *2.2.4.3.1 Importación de datos*

RapidMiner ofrece un gran número de maneras para poder importar datos que el investigador desee estudiar. Los datos pueden ser importados en distintos formatos en una máquina local como también pueden ser extraídos desde una base de datos. A continuación, se explicarán las formas y formatos que permite Rapidminer para la importación de datos.

- Importar archivo CSV

  CSV es una abreviatura en ingles de “valores separados por coma”. Los archivos CSV almacenan datos en forma de texto plano, los que pueden ser numéricos o texto. Todos los valores correspondientes se almacenan como una línea del archivo CSV. La forma más

adecuada de importar datos en CSV es utilizando el operador “READ CSV” disponible en la sección “Operators”.

- Importar archivo Excel

  El operador “READ EXCEL” puede leer datos de Excel 95, 97, 2000, XP y 2003. El usuario debe definir cuál de las hojas de cálculo del libro de trabajo debe usarse como tabla de datos. La tabla debe tener un formato tal que cada fila sea una tupla de datos y cada columna represente un atributo. Por último, es necesario tener en cuenta que la primera fila de la hoja Excel podría usarse para nombre de los atributos.

- Importar archivo ARFF

  El operador “READ ARFF” puede leer los archivos ARFF, formato explicado en la sección 2.2.4.1.1 Formato. ARFF. Un archivo ARFF es un archivo de texto ASCII que describe una lista de instancias que comparten un conjunto de atributos.

- Importar archivo SPSS

  El operador “READ SPSS” permite leer archivos creados por SPSS (Paquete estadístico para las ciencias sociales), una aplicación utilizada para el análisis estadístico. Los archivos de SPSS se guardan en un formato binario y contienen un conjunto de datos. Estos archivos guardan los datos por ‘casos’(filas) y ‘variables’(columnas) y tienen una extensión de archivo ‘.SAV’

- Importar archivo XML

  El operador “READ XML” puede leer archivos XML, donde los datos están representados por elementos que coinciden con un XPath (lenguaje que permite construir expresiones que recorren y procesan un documento XML) determinando, las características, el

contenido de texto de cada elemento y sus subelementos. Este operador intenta determinar un tipo apropiado de atributos leyendo los primeros elementos y verificando los valores que ocurren. Si todos los valores son enteros, el atributo se convertirá en entero, si se producen números reales, será de tipo real. Las columnas que contienen valores que no pueden interpretarse como números serán nominales, siempre que no coincidan con el patrón de fecha y hora del parámetro de formato de fecha. Si lo hacen, este atributo se analizará automáticamente como fecha y la característica correspondiente será de tipo fecha.

- Importar datos desde BD SQL

  El operador “READ DATABASE” permite al usuario conectarse a una base de datos y leer un set de datos desde una base de datos SQL. El usuario debe tener al menos una comprensión básica de las bases de datos, las conexiones de las bases de datos y las consultas para poder utilizar este operador correctamente.

*2.2.4.3.2 Interfaces RapidMiner*

RapidMiner provee una serie de interfaces (véase ilustración 6) que permiten hacer uso de la herramienta. A continuación, se explican cada una de ellas.

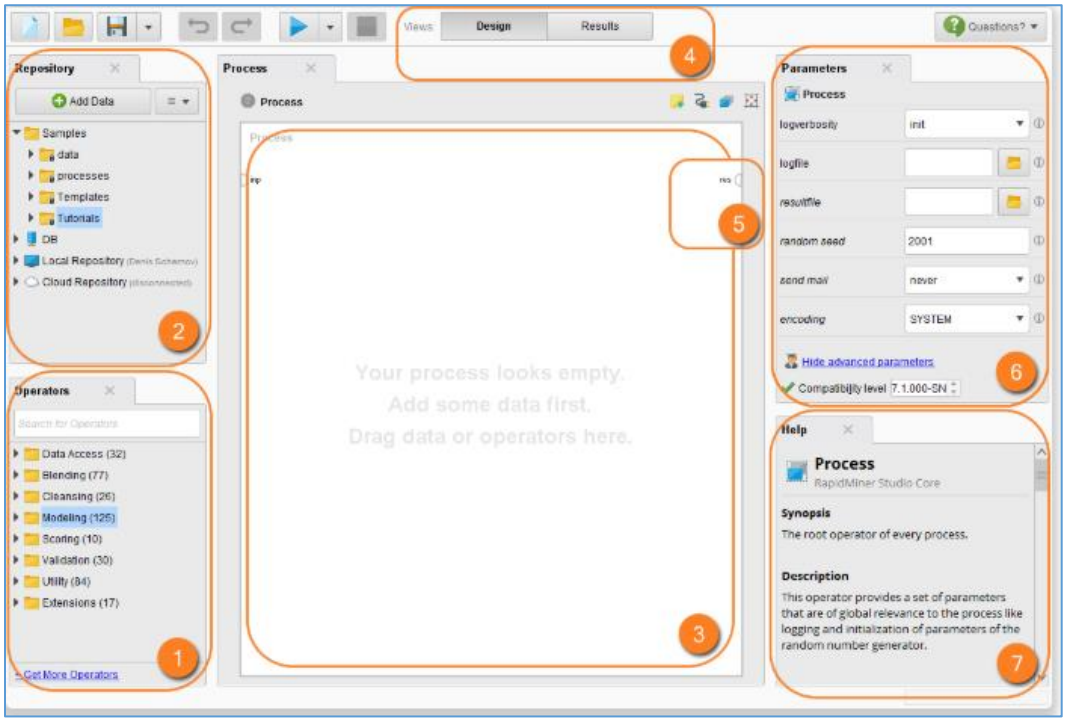


*Ilustración 6 Panel de diseño RapidMiner.*

- 1. Operadores: En esta sección se encuentran cada uno de los "bloques de construcción" que dispone la herramienta para crear procesos.
- 2. Repositorio: Esta sección permite almacenar datos y procesos de Rapidminer.
- 3. Panel de proceso: Área de trabajo para procesos constructivos.
- 4. Vistas: Área de trabajo para acceder a funcionalidades específicas de diseño y resultados.
- 5. Puertos: Mecanismos de entrada y salida para operadores y procesos.
- 6. Parámetros: Configuraciones que modifican el comportamiento del operador seleccionado.
- 7. Ayuda: Ayuda sensible al contexto del operador seleccionado.

*2.2.4.3.3 Algoritmos RapidMiner*

RapidMiner cuenta con una variedad de algoritmos representado en operadores, los que se usan en el diseño de un proceso. Cada uno de los tipos de algoritmos es agrupado en carpetas con su nombre genérico, a continuación, se procede a mencionarlos.

- Lazy Learning: Algoritmos de aprendizaje perezoso como por ejemplo el algoritmo k-NN
- Bayesian: Algoritmos bayesianos como por ejemplo el algoritmo Naive Bayes.
- Trees: Árbol de decisión donde se encuentran algoritmos como CHAID, Decisión Stump, Decisión Tree, Gradient Boosted Trees, ID3, Random Forest, Random Tree
- Rules: Algoritmos para el aprendizaje de reglas como Rule Induction, Subgroup Discovery y Tree to Rules
- Neural Nets: Redes neuronales como Deep Learning, Neural Net y Perceptron
- Functions: Gaussian Process, Generalized Linear Model, Local Polynomial Regression, Polynomial Refression, Relevance Vector Machine y Vector Linear Regression.
- Logistic Regression: Regresiones logísticas como, por ejemplo, Linear Discriminant Analysis, Quadratic Discriminant Analysis y Regularized Discriminant Analysis
- Support Vector Machines: Máquinas de vectores de soporte como Fast Large Margin, Hyper Hyper, Support Vector Machine, Support Vector Machine (Evolutionary), Support, Vector Machine (LibSVM), Support Vector Machine (Linear), Support Vector Machine (PSO).
- Discriminant Analysis: Algoritmos de análisis discriminante como Linear Discriminant Analysis, Quadratic Discriminant Analysis, Regularized Discriminant Analysis.

### *2.2.4.4 Comparación de softwares*

A continuación, se hace una comparación de los tres software anteriormente descritos, basada en los indicadores de Mojarrango & Chapalbay (2016):

*Tabla 5 Comparación de software de minería de datos*

| Indicador | Weka | SAP PA | RapidMiner |
|---|---|---|---|
| **Permite la importación de Archivos de Texto (.txt, .csv)** | 1 | 1 | 1 |
| **Permite la importación de Excel/spreadsheet** | 0 | 1 | 1 |
| **Permite la importación de Tabla de base de datos** | 1 | 1 | 1 |
| **Permite la importación de datos desde una URL** | 1 | 0 | 1 |
| **Permite la importación de Otras fuentes de datos** | 1 | 1 | 1 |
| **Permite la visualización previa de la información** | 1 | 1 | 1 |
| **Permite obtener el número de registros** | 1 | 1 | 1 |
| **Permite obtener el número de las variables** | 1 | 1 | 1 |
| **Permite la identificación de las variables** | 1 | 1 | 1 |
| **Permite la descripción del tipo inicial de las variables** | 1 | 1 | 1 |
| **Permite la creación de gráficos de distribución estadística** | 1 | 1 | 1 |
| **Permite obtener la cantidad de valores nulos** | 1 | 1 | 1 |
| **Permite encontrar valores erróneos** | 1 | 1 | 1 |
| **Permite seleccionar un subconjunto de los datos adquiridos de acuerdo a la verificación de la calidad de los datos** | 1 | 1 | 1 |
| **Permite la normalización de los datos** | 1 | 1 | 1 |
| **Permite la discretización de campos numéricos** | 1 | 1 | 1 |
| **Permite el tratamiento de valores ausentes** | 1 | 1 | 1 |
| **Permite la reducción del volumen de datos** | 1 | 1 | 1 |
| **Permite generar nuevos atributos a partir de los ya existentes** | 1 | 1 | 1 |
| **Permite integrar nuevos registros** | 1 | 1 | 1 |
| **Permite la transformación de valores para atributos ya existentes** | 1 | 1 | 1 |
| **Permite la generación de nuevos campos a partir de otros existentes** | 1 | 1 | 1 |
| **Permite la creación de nuevos registros** | 0 | 1 | 0 |
| **Permite la fusión de tablas campos o nuevas tablas** | 1 | 1 | 0 |
| **Permite la reordenación de los registros** | 1 | 1 | 1 |
| **Permite la reordenación de los campos** | 1 | 1 | 1 |
| **Permite utilizar algoritmo/s de Árboles de Decisión** | 1 | 1 | 1 |
| **Permite utilizar algoritmo/s para Clustering** | 1 | 1 | 1 |
| **Permite utilizar algoritmo/s para Series Temporales** | 1 | 1 | 1 |
| **Permite utilizar algoritmo/s Clustering de Secuencia** | 1 | 1 | 1 |
| **Permite utilizar algoritmo/s para Asociación** | 1 | 1 | 1 |
| **Permite utilizar algoritmo/s para Redes Bayesianas** | 1 | 1 | 1 |
| **Permite utilizar algoritmo/s de Redes Neuronales** | 1 | 1 | 1 |
| **Permite la construcción gráfica del modelo** | 1 | 1 | 1 |
| **Permite la ejecución del modelo con una interfaz gráfica** | 0 | 0 | 0 |
| **Permite la creación de un flujo gráfico del modelo** | 1 | 1 | 1 |
| **Permite evaluar el performance del modelo** | 1 | 1 | 1 |
| **Genera gráficos de evaluación** | 1 | 1 | 1 |
| **Permite analizar el costo beneficio del modelo** | 1 | 1 | 1 |
| **Permite crear gráficos que ilustren los resultados obtenidos** | 1 | 1 | 1 |

Como se observa en la Tabla 5, las tres herramientas de comparadas poseen características bastante similares respecto de la importación, entrenamiento y resultados. Además, Weka, SAP PA y RM permiten utilizar algoritmos de árbol de decisión y redes neuronales por lo que son apropiadas para este estudio.

### 2.2.5 Modelo de deserción basado Regresión logística UBB

Retamal & Rubilar (2017) postulan un modelo de deserción basado en el método ScoreCard, aplicando la estimación de parámetros a través de un modelo de regresión logística. Para el desarrollo de este trabajo se contó con una base de datos de las cohortes 2014 y 2015 con 4519 sujetos y 32 variables, entre ellas la variable respuesta.

Según Rayo Cantón, Lara Rubio, & Camino Blasco (2010) el ScoreCard es un método estadístico que han utilizado las empresas como bancos y/o retail para otorgar créditos o préstamos a sus clientes, mediante la clasificación de estos individuos en los tipos de riesgo “bueno” y “malo”. Dichos procedimientos estadísticos son algoritmos que de manera automática evalúan el riesgo de crédito de un solicitante de financiamiento o de alguien que ya es cliente de la entidad.

La selección de las variables se hizo mediante los indicadores IV y WoE. Valor de la información (IV) es un indicador que consiste en una medida que aparece en la teoría de información e indica una proporción de buenos y malos criterios en los atributos de cada característica. Peso de la evidencia (WoE) es un indicador donde el poder de predicción de cada atributo o grupo se calculó con los pesos de las evidencias, es una medida de diferencia de las proporciones de buenos y malos en cada atributo. Luego se seleccionaron 10 atributos, pues eran la mejor combinación de variables que dieran cuenta del comportamiento de los datos.

Para cada estudiante se calculó un puntaje o score, de estos puntajes obtenidos se elige un punto de corte para que de esta forma se obtenga un porcentaje total de los clasificados y clasificados con éxito. Considerando que este modelo busca clasificar estudiantes desertores, en riesgo de desertar y regulares, se determina que el puntaje de corte es 662 puntos. Con este último, el modelo arroja como resultado una exactitud (predicción general) de 68% una especificidad (predicción desertores) de 57%.

Por otro lado, la institución establece un porcentaje que se encuentra dispuesto a aceptar como máximo de error, debido a que busca un punto de corte con una mayor tasa de éxito (sensibilidad). Después de un análisis de curva de precisión el punto de corte escogido es 640, debido a que aumenta su tasa de éxito, con un total de 60% de clasificados y un 69% de desertores clasificados con éxito (ver Tabla 6 y Tabla 7).

*Tabla 6 Matriz de confusión Modelo de Regresión Logística UBB.*

| | Predicción | | |
|---|---|---|---|
| **Observación** | **SI** | **NO** | **Total** |
| **SI** | 105 | 47 | 152 |
| **NO** | 315 | 437 | 752 |
| **Total** | 420 | 484 | 904 |

*Tabla 7 Resultados análisis de rendimiento Modelo de Regresión Logista UBB.*

| Evaluación | Resultados | Porcentaje |
|---|---|---|
| **Exactitud** | 0.599 | 60% |
| **Tasa de error** | 0.400 | 40% |
| **Sensibilidad** | 0.581 | 58% |
| **Especificidad** | 0.690 | 69% |
| **Precisión** | 0.901 | 90% |
| **Predicción negativa** | 0.250 | 25% |
| **Índice de Youden** | 0.271 | 27% |

Dado lo anterior se pudo obtener un puntaje de corte adecuado, con el propósito de beneficiar la correcta clasificación de éxito, es decir la correcta clasificación de alumnos desertores, etiquetados como desertores.

### 2.2.6 Modelo de deserción basado en árboles de decisión UBB

Pérez et al. (2018) en su trabajo denominado "Análisis comparativo de técnicas de predicción para la deserción estudiantil: el caso de la Universidad del Bío-Bío" postula dos modelos predictivos basados en árbol de decisión para la predicción de la deserción estudiantil. Este análisis buscaba generar un modelo que permitiera predecir que estudiantes tenían más posibilidades de desertar de sus planes de estudios y que además fuera más eficiente que el modelo existente. Para llevar a cabo este trabajo, los autores utilizan el software SAP Predictive Analytics y un conjunto de datos de 4519 estudiantes ingresados a la UBB (sede Chillán y sede Concepción) en los años 2014 y 2015. Entre esos datos incluye datos de tipo personal, académico, institucional y socioeconómicos.

Primeramente, para la construcción de uno de los modelos considera las siguientes 10 variables de los estudiantes: carrera, puntaje PSU matemáticas, NEM, orden de postulación, becas de arancel MINEDUC, domicilio universitario, motivación a trabajar, género, comuna de domicilio y gratuidad. Luego, en SAP PA, entrena los datos con el algoritmo R-CNR Tree lo que arroja como resultado un Árbol de decisión y una matriz de confusión (véase tabla 8). Esta última utilizada para evaluar el desempeño del modelo.

*Tabla 8 Matriz de confusión Modelo de árboles de decisión V1 UBB.*

| | Predicción | | |
|---|---|---|---|
| **Observación** | **SI** | **NO** | **Total** |
| **SI** | 252 | 105 | 357 |
| **NO** | 541 | 3621 | 4162 |
| **Total** | 793 | 3726 | 4519 |

Como muestra la tabla 9 este primer modelo tiene una exactitud de 86% y un 87% de sensibilidad para predecir a los estudiantes que SI van a desertar, sin embargo, este modelo fue puesto a prueba con los mismos datos con los cuales se entrenó, por lo que lo resultados no son muy confiables.

*Tabla 9 Resultados análisis de rendimiento Modelo de árboles de decisión V1 UBB.*

| Evaluación | Resultados | Porcentaje |
|---|---|---|
| **Exactitud** | 0.857 | 86% |
| **Tasa de error** | 0.143 | 14% |
| **Sensibilidad** | 0.870 | 87% |
| **Especificidad** | 0.705 | 70% |
| **Precisión** | 0.971 | 97% |
| **Predicción negativa** | 0.317 | 32% |
| **Índice de Youden** | 0.575 | 57% |

Con el objetivo de optimizar el modelo propuesto, el segundo modelo se construye de la misma manera solo que se utilizaron los siguientes 17 atributos: NEM, puntaje PSU matemáticas, puntaje PSU lenguaje, ranking, carrera, edad, duración de carrera, tipo de enseñanza, orden de postulación, años de acreditación, tipo de colegio, género, gratuidad, sede, becas arancel MINEDUC, beca excelencia académica y discapacidad. Este modelo arroja la siguiente matriz de confusión (ver Tabla 10).

*Tabla 10 Matriz de confusión Modelo de árboles de decisión V2 UBB.*

| | Predicción | | |
|---|---|---|---|
| **Observación** | **SI** | **NO** | **Total** |
| **SI** | 591 | 132 | 723 |
| **NO** | 202 | 3594 | 3796 |
| **Total** | 793 | 3726 | 4519 |

En este segundo modelo se obtuvieron mejores resultados de exactitud con un 93% y un 95% de sensibilidad para la predicción de estudiantes desertores (ver Tabla 11). Pérez concluye afirmando que su modelo entrega resultados prometedores en comparación al modelo previo

existente en la UBB, considerando que el software utilizado para este estudio es orientado a estrategias de negocio.

*Tabla 11 Resultados análisis de rendimiento Modelo de árboles de decisión V2 UBB.*

| Evaluación | Resultados | Porcentaje |
|---|---|---|
| **Exactitud** | 0.926 | 93% |
| **Tasa de error** | 0.073 | 7% |
| **Sensibilidad** | 0.946 | 95% |
| **Especificidad** | 0.817 | 82% |
| **Precisión** | 0.964 | 96% |
| **Predicción negativa** | 0.745 | 75% |

# Capítulo 3: Diseño y construcción de modelos

El siguiente paso consiste en la construcción de los modelos de minería de datos, usando árbol de decisión y redes neuronales, con el objetivo de tener un parámetro de comparación respecto a qué modelo presenta una mejor respuesta frente a la estimación de la variable deserción.

## 3.1 Caracterización de datos

Para este estudio, se facilitó un archivo. xslx que contiene un conjunto de datos provenientes desde la Dirección de Admisión y Registro Académico (DARCA). Este set de datos está conformado por 4519 registros, que contienen antecedentes de los estudiantes ingresados en los años 2014-2015 y 32 variables considerando la variable "Deserción", que en este caso es la variable objetivo. En la ilustración 7 se puede apreciar un resumen de la población total, distribuida en estudiantes que desertaron y estudiantes que no desertaron en la muestra disponible.

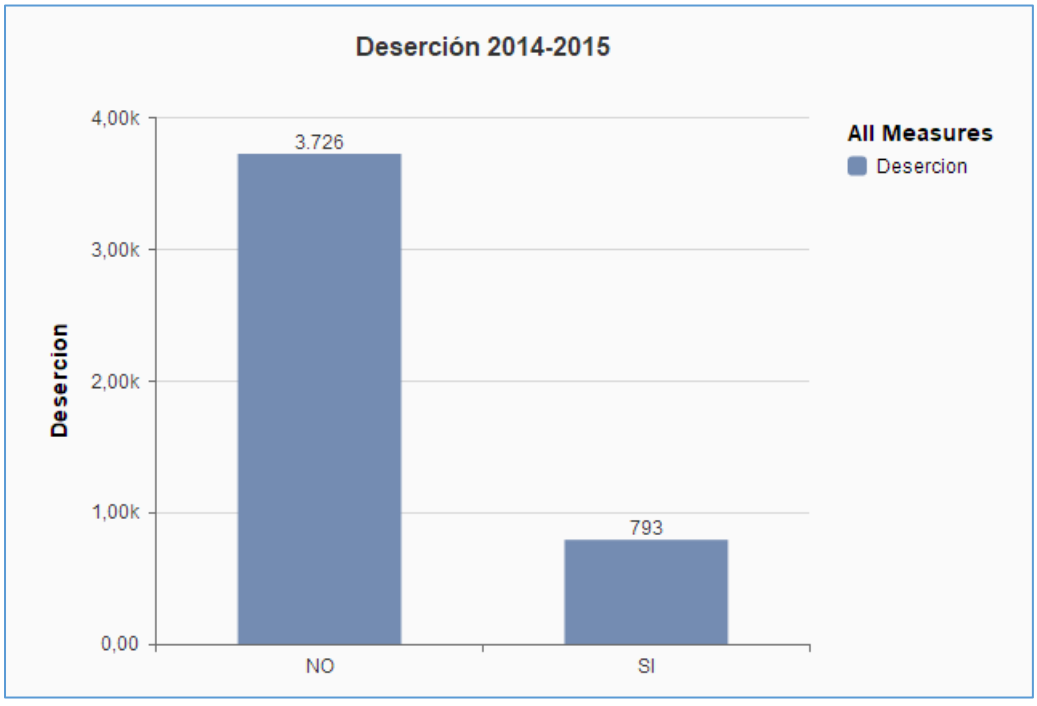


*Ilustración 7 Caracterización de la variable deserción en la UBB entre los años 2014-2016.*

La selección de variables independientes a utilizar en la construcción de los modelos se basa en las variables utilizadas en los trabajos de Pérez et al. (2018) y Retamal & Rubilar (2017). Estos autores, basándose en un análisis bibliográfico, consideraron que 10 de las 32 variables disponibles en el set de datos constituían una mejor combinación para tener mayor éxito en la predicción de la deserción (ver tabla 12). Estas variables son de tipo personales, académicas, institucionales y socioeconómicas.

*Tabla 12 Variables independientes.*

| Tipo | Variable | Descripción |
|---|---|---|
| **Personales** | Región | Region de domicilio de la familia |
| | Vive_año_actual | Domicilio actual del año académico |
| | Género | Sexo del estudiante |
| | Deserción | Estado actual del estudiante |
| | Facultad | Numero correlativo a la facultad |
| **Académicas** | PSU_Mate | Puntaje PSU Matemáticas |
| | NEM | Promedio de notas de enseñanza media |
| **Institucionales** | Orden_postulación | Orden de selección de preferencia en la postulación a la carrera |
| **Socioeconómicas** | Becas_arancel_mindeuc | Posee Beca Arancel Mineduc |
| | Gratuidad | Posee gratuidad |
| **Otras** | Motivación_trabjar | Motivación por lo cual trabajará durante la carrera |

Por último, se recalca que la variable objetivo es la “Deserción” y la entenderemos según los siguientes criterios:

- Estudiantes que ingresaron a una carrera en particular y el año inmediatamente posterior se inscribieron en una carrera distinta.
- Retiro definitivo.
- Perdida de carrera.
- No inscripción de asignaturas en los 3 semestres subsiguientes por causas distintas al retiro temporal.

- No descripción de asignaturas por retiro temporal (se desconoce si volverán a estudiar en semestres subsiguientes).

De igual forma a los alumnos no desertores se les caracterizará como estudiantes que continuaron sus planes de estudios normalmente en la carrera que iniciaron ya sea en los semestres subsiguientes para las carreras semestrales y al año siguiente para las anuales.

### 3.2 Estandarización de las variables independientes

En esta sección se caracterizarán todas las variables que se utilizarán en el modelo, estandarizando dichos valores y agrupando valores atípicos que éstas presenten (Pérez et al., 2018).

- Región de procedencia familiar

  Variable: Región

  Tipo: Cualitativa nominal

  Descripción: número de la región donde el alumno registra el domicilio familiar.

- Domicilio año académico.

  Variable: Vive año actual

  Tipo: Cualitativa nominal

  Descripción: Corresponde con quien vivirá el alumno durante el año académico.

  Observación: Los seis atributos para esta característica son:

  - Atributo 1: Padres (uno o ambos).

  - Atributo 2: Amistades.

  - Atributo 3: Aún no definido, solo y otro.

  - Atributo 4: Familiares.

- Atributo 5: Pareja.

- Género.

  Variable: Género Tipo: Cualitativa nominal

  Descripción: Género correspondiente a los estudiantes objeto de estudio.

  Observación: Esta característica posee dos atributos los cuales son: masculino y femenino.

  - 1: Masculino.

  - 2: Femenino.

- Deserción

  Variable: Deserción

  Tipo: Cualitativa Nominal

  Descripción: Alumno con situación académica de disección definitiva.

  Observación: los atributos para esta característica son:

  - Atributo 0: Retenido

  - Atributo 1: Desertado

- Facultad.

  Variable: Facultad.

  Tipo de variable: Cuantitativa continúa

  Descripción: Correlativo para identificar la facultad

  Observación: se utilizan números del 1 al 6.

  - Atributo 1: Arquitectura, Construcción y Diseño

  - Atributo 2: Ingeniería

  - Atributo 3: Ciencias Empresariales

- Atributo 4: Educación y Humanidades

- Atributo 5: Ciencias de la Salud y de los alimentos

- Atributo 6: Ciencias

- Puntaje PSU

  Matemática. Variable: PSU_Mate

  Tipo de variable: Cuantitativa continúa

  Descripción: Puntaje obtenido en la prueba de selección universitaria de matemática (PSU)

  Observación: Para manipular esa variable en el modelo de deserción, se ingresaron las variables numéricas desde los rangos de 120 hasta 850 puntos, tomando todo el espectro de puntajes.

- Notas de enseñanza media

  Variable: NEM

  Tipo: Cuantitativa continua

  Descripción: Promedio de notas de los 4 años de enseñanza media del alumno.

  Observación: Para trabajar esta variable en el modelo de deserción, se manipularon las notas y se trasformaron a puntajes, para que todos los valores fuesen estandarizados con el valor del puntaje correcto.

- Orden de postulación.

  Variable: Orden_postulación

  Tipo: Cualitativa ordinal

  Descripción: Orden de prioridad que el alumno le da a la carrera al momento de postular.

Observación: Los seis atributos para esta característica son:

- Atributo 0: No postulo a ninguna carrera (0), Sin Información (SinInf), preferencia 8, 9 y 10.

- Atributo 1: Preferencia 1.

- Atributo 2: Preferencia 2.

- Atributo 3: Preferencia 3.

- Atributo 4: Preferencia 4, 5, 6 y 7.

- Beca arancel MINEDUC.

  Variable: Beca arancel MINEDUC

  Tipo: Cualitativa nominal

  Descripción: Es un apoyo económico que entrega el Ministerio de Educación para financiar parte del costo de los estudios, cubriendo el total o parte del arancel anual de la carrera.

  Observación: Los dos atributos para esta característica son:

  - Atributo 0: Estudiantes sin Beca arancel MINEDUC.

  - Atributo 1: Estudiantes con Beca arancel MINEDUC.

- Gratuidad.

  Variable: gratuidad.

  Tipo: Cualitativa nominal.

  Descripción: Beneficio otorgado a las familias correspondientes al 50% más vulnerable de la población, cuyos miembros estudien en universidades, Centros de Formación Técnica o Institutos Profesionales adscritos a la Gratuidad, no deberán pagar el arancel ni la matrícula en su institución durante la duración formal de la carrera.

  Observación: Los dos atributos para esta característica son:

- Atributo 0: Estudiantes sin gratuidad.

- Atributo 1: Estudiantes con gratuidad.

- Motivación trabajar.

  Variable: Motivacion_trabajar

  Tipo: Cualitativa nominal

  Descripción: Se caracteriza al tipo de estudiante que tiene la necesidad de trabajar durante el año académico, orientado la necesidad algunos de los atributos mencionados.

  Observación: Los dos atributos para esta característica son:

  - Atributo 0: Ninguna.

  - Atributo 1: Aportar al ingreso familiar.

  - Atributo 2: Pagar estudios.

  - Atributo 3: Para gustos personales.

  - Atributo 4: Desarrollar habilidades laborales.

  - Atributo 5: otro.

### 3.3 Clasificación

A partir de un conjunto de ejemplos de los que conocemos su valor objetivo se aplica una técnica de aprendizaje supervisado llamado clasificación. La clasificación intenta encontrar una función que permita asignar un valor objetivo a ejemplos que el sistema no ha visto anteriormente (y de los que, generalmente, no se conoce el valor de salida correcto).

### 3.4 Método de evaluación y comparación

Para poder comparar y evaluar la calidad de los modelos entrenados es necesario definir una métrica que lo permita. Por ende, se definió la matriz de confusión y sus índices derivados como herramienta de evaluación y comparación entre las tres herramientas seleccionadas.

La matriz de confusión es una herramienta que permite visualizar el desempeño de un algoritmo de aprendizaje. Como muestra la tabla 13, cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa las instancias en la clase real. Para poder utilizar este método de evaluación de un modelo de clasificación fue necesario separa nuestro conjunto de datos en dos datasets, uno denominado "train" (80% datos) con el cual se entrenaron los modelos y otro llamado "test" (20% datos) con el que probaron los modelos. Esto nos ayuda a medir cómo se comportan los modelos cuando evalúa con un conjunto de datos nuevo.

*Tabla 13 Matriz de confusión.*

| | **Predicción** | |
|---|---|---|
| **Observación** | **Positivos** | **Negativos** |
| **Positivos** | Verdaderos Positivos (VP) | Falsos Negativos (FN) |
| **Negativos** | Falsos Positivos (FP) | Verdaderos Negativos (VN) |

La comparación de los modelos, se basó en los índices de rendimiento (Fawcett, 2006) derivados de la matriz de confusión, los cuales son explicados a continuación:

- Exactitud (acurrancy), es definida como el porcentaje de aciertos del modelo. Es decir, todos aquellos valores correctos que se encuentran debidamente clasificados y pertenecen a dicho clasificador. La siguiente fórmula define la exactitud:

$$\textbf{Exactitud} = \frac{\textbf{VP} + \textbf{VN}}{\textbf{Total}}$$

- Tasa de error (misclassification rate), es definida como el porcentaje erróneo que ha clasificado el modelo. Es decir, todos aquellos valores incorrectos que se encuentran indebidamente clasificados y no pertenecen a dicho clasificador. A continuación, se incluye la fórmula de cálculo de este indicador:

$$\textbf{Tasa de error} = \frac{\textbf{FP} + \textbf{FN}}{\boldsymbol{Total}}$$

- Sensibilidad, (sensitivity, recall), es definida como la eficiencia en la clasificación de todos los elementos que son de una clase. Es decir, entrega el porcentaje que es capaz de clasificar de todas las observaciones positivas que son correctamente clasificadas como positivas, de acuerdo a la siguiente fórmula:

$$\textbf{Sensibilidad} = \textbf{S} = \frac{\textbf{VP}}{\textbf{Total Positivo}}$$

- Especificidad, (especificity), conocida como tasa de verdaderos negativos, es definida como el porcentaje de captura que es capaz de clasificar el modelo dentro de todas las observaciones negativas correctamente clasificadas como negativas, la siguiente fórmula muestra la ecuación que define la especificidad.

$$\boldsymbol{Especificidad} = \boldsymbol{E} = \frac{\boldsymbol{VN}}{\boldsymbol{Total\ Negativo}}$$

- Precisión, es definida como la medida o porcentaje de calidad de la respuesta del clasificador. Es decir, nos mide el porcentaje de resultados positivos que son correctas, correspondiendo a muestras positivas. La siguiente formula define este valor:

$$\boldsymbol{Precisión = \frac{VP}{Total\ Clasificados\ Positivos}}$$

- Predicción negativa, es definida como el porcentaje negativo de la respuesta del clasificador, es decir entrega el porcentaje de resultados negativos que son correctamente clasificados como muestras negativas:

$$\boldsymbol{Predicción\ negativa = \frac{VN}{Total\ Clasificados\ Negativos}}$$

- Índice de Youden, es definido como un análisis estadístico que entrega el comportamiento o rendimiento de una prueba diagnóstica. Este índice se basa en maximizar las diferencias entre verdaderos positivos y verdaderos negativos. Por lo tanto, se buscará el comportamiento en el modelo que posea un compromiso aceptable, el rango posible de valores es de 0 a 1.

$$\boldsymbol{Índice\ de\ Youden = IY = S + E - 1}$$

## 3.5 Construcción de modelos

En esta sección se procede a describir el proceso de entrenamiento y evaluación de cada modelo en particular en los tres softwares seleccionados para comparar.

### 3.5.1 Modelos Weka

Como se mencionó anteriormente Weka es una herramienta para análisis de datos, contiene implementaciones de algoritmos de aprendizaje y además incluye herramientas para la transformación de conjuntos de datos. Se utiliza herramienta para evaluar el desempeño de sus métodos de clasificación árbol de decisión y red neuronal.

En primer lugar, el módulo que se utiliza es el explorador de Weka (ver ilustración 8). Luego, dentro del explorador con la opción “Open file” se importa el set de datos que previamente había sido transformado de formato .xlsx a .arff como se muestra en la sección 2.2.4.1.1 formato ARFF. En la pestaña “Preprocess” se permite visualizar y modificar información del set de datos importado, en este caso se verificó que la cantidad de datos fuera correcta y que las 11 variables definidas para el estudio fueran del tipo correcto.

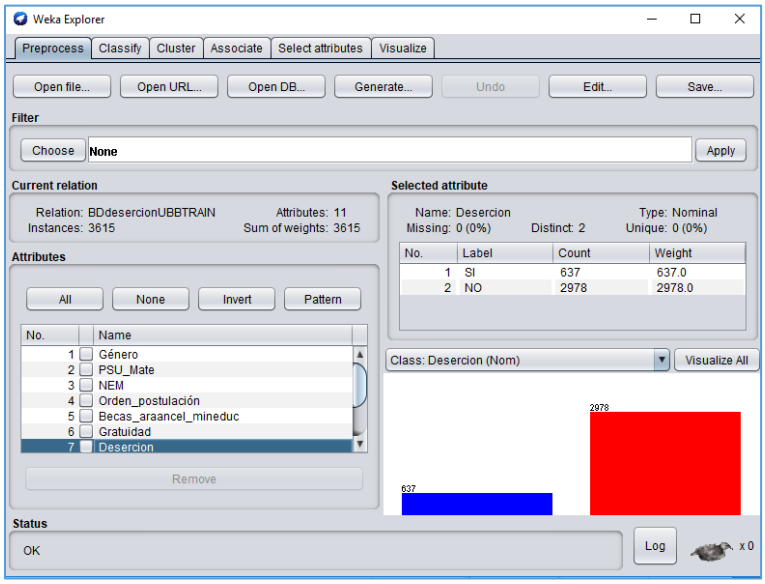


*Ilustración 8 Weka Explorer, Pestaña Preprocess.*

Seguidamente, en el mismo Weka explorer, se ingresó a la pestaña “Classify”, la cual da la opción de elegir un algoritmo para clasificar presionando el botón “Choose” y seleccionando el algoritmo en la lista que se despliega al presionarlo. Considerando que la importación de los datos de entrenamiento es similar en los dos algoritmos solo se explicará una vez. A continuación, se revisarán el entrenamiento del árbol de decisión y red neuronal por separado.

#### *3.5.1.1 Árbol de decisión Weka*

En la pestaña de clasificación se procede a la elección del algoritmo de árbol de decisión, en este caso el algoritmo que se utilizó y está disponible en Weka es el REPTree. El algoritmo REPTree construye un árbol de decisión/regresión utilizando la ganancia / varianza de la información y lo poda utilizando la reducción de errores. Solo ordena los valores de los atributos numéricos una vez. Los valores que faltan se tratan dividiendo las instancias correspondientes en partes (es decir, como en C4.5).

Para el entrenamiento de este modelo utilizaremos los valores por defecto. A continuación, se explican los parámetros utilizados:

*Tabla 14 Descripción parámetros del algoritmo REPTree.*

| Parámetro REPTree | Valor | Descripción |
|---|---|---|
| **batchSize** | 100 | El número preferido de instancias para procesar si se está realizando la predicción por lotes. |
| **debug** | False | Si se establece en true, el clasificador puede generar información adicional en la consola. |
| **doNotCheckCapabilities** | False | Si se establece, las capacidades del clasificador no se verifican antes de que se compile el clasificador |
| **initialCount** | 0.0 | Cuenta inicial del valor de la clase. |
| **maxDepth** | -1 | La profundidad máxima del árbol. |
| **minNum** | 2.0 | Los pesos totales mínimos de las instancias en una hoja. |
| **minVarianveProp** | 0.001 | La proporción mínima de la varianza en todos los datos que deben estar presentes en un nodo para que la división se realice en árboles de regresión. |
| **noPruning** | true | Si se realiza la poda. |
| **numDecimalPlaces** | 2 | El número de decimales que se utilizarán para la salida de números en el modelo. |
| **numFolds** | 3 | Determina la cantidad de datos utilizados para la poda. |
| **seed** | 1 | La semilla utilizada para aleatorizar los datos. |
| **spreadInitialCount** | False | distribuye el recuento inicial en todos los valores en lugar de utilizar el recuento por valor. |

Luego de definir los parámetros del algoritmo REPTree, se procede a seleccionar el método de prueba. En el módulo “Test options” se selecciona la opción “Percentage split” y se setea en 80%, esto permite la utilizar un 80% de datos de entrenamiento y el resto, un 20%, para test. Por último, al volver a la pestaña “Classify” se seleccionó el botón “Start”, el cual pone en marcha el entrenamiento de los datos y la prueba del modelo entrenado con los datos de prueba automáticamente (ver ilustración 9). Cabe mencionar que en la pantalla de salida se puede observar una serie de resultados incluyendo la matriz de confusión y una representación del árbol de decisión.

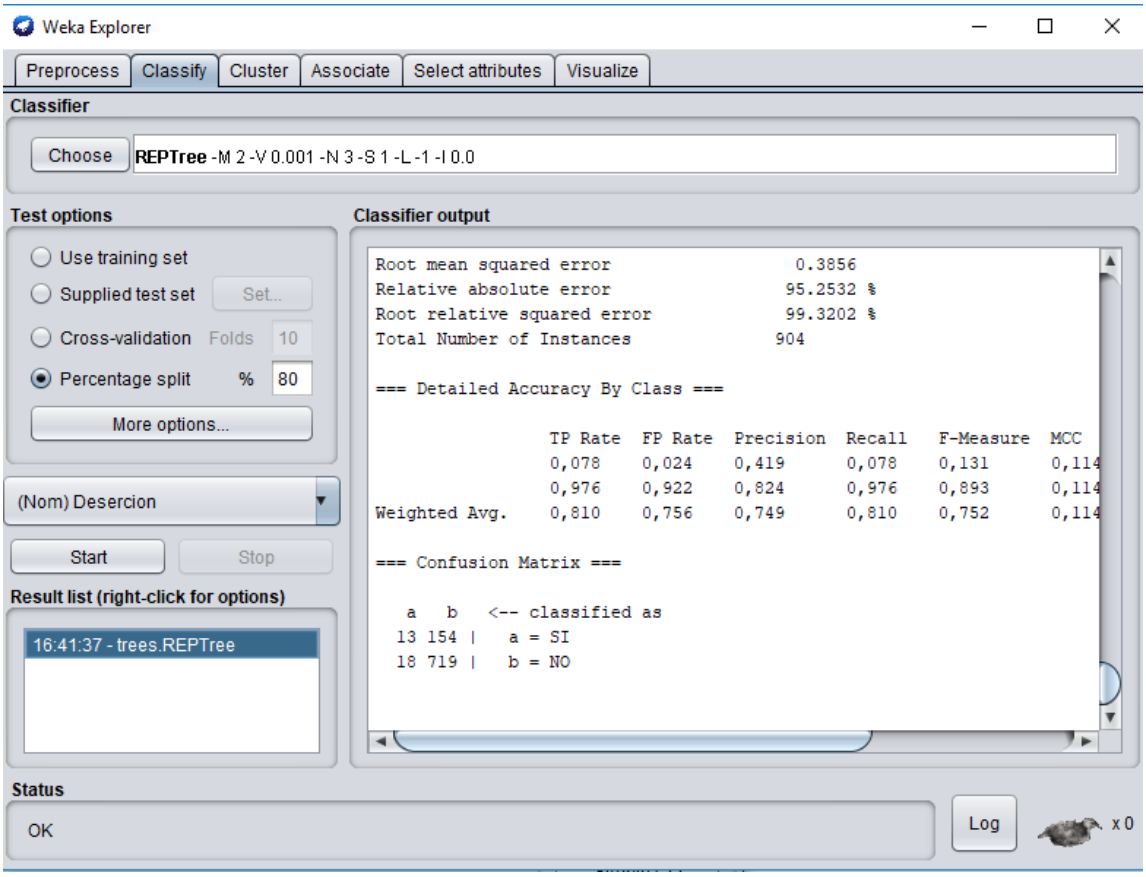


*Ilustración 9 Weka Explorer, pestaña Classify.*

#### *3.5.1.2 Red Neuronal Weka*

La red neuronal que dispone Weka lleva por nombre Multilayer Perceptron y es un algoritmo backpropagation, esto quiere decir que emplea un ciclo de propagación. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas siguientes de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Para entrenar la red neuronal, primero se importaron los datos de entrenamiento desde un archivo .arff de igual forma a como se explicaba en el Árbol de decisión. Luego en la pestaña “Classify” se seleccionó el algoritmo Multilayer Perceptron y se definieron sus parámetros como muestra la Tabla 15.

*Tabla 15 Parámetros algoritmo MultilayerPerceptron.*

| Parámetros Multilayer Perceptron | Valor | Descripción |
|---|---|---|
| **GUI** | False | Trae una interfaz GUI |
| **autoBuild** | True | Agrega y conecta capas ocultas en la red. |
| **batchSize** | 100 | El número preferido de instancias para procesar si se realizan predicciones por lotes |
| **debug** | False | Si se establece en true, el clasificador puede generar información adicional en la consola |
| **decay** | False | Esto hará que la tasa de aprendizaje disminuya. |
| **doNotCheckCapabilities** | False | Si está configurado, las capacidades del clasificador se verifican antes de que se construya |
| **hiddenLayers** | A | Define las capas ocultas de la red neuronal. |
| **learningRate** | 0.3 | Rapidez con la que cambian los pesos |
| **momentum** | 0.2 | Impulso con aplicado la actualización de pesos |
| **nominalToBinaryFilter** | True | Esto pre procesará las instancias con el filtro nominal a binario |
| **normalizeAttributes** | True | Esto normalizará los atributos |
| **normalizeNumericClass** | True | Esto normalizará la clase si es numérico. |
| **numDecimalPlaces** | 2 | El número de decimales que se utilizarán para la salida de número en el modelo |
| **reset** | True | Esto permitirá que la red aumente con una baja tasa de aprendizaje. |
| **seed** | 0 | Semilla utilizada para inicializar el generador de números aleatorios |
| **trainingTime** | 500 | El número de épocas para entrenar. |
| **validationSetSize** | 0 | El porcentaje de tamaño del conjunto de validación |
| **validationThreshold** | 20 | Utilizado para terminar las pruebas de validación |

Al igual que en el árbol de decisión, se selecciona y setea la opción “Percentage split” en 80%, para entrenar los datos importados (80% datos) y ponerlos a prueba con los datos de entrenamiento (20% datos) (ver iIlustración 10). En la pantalla de salida se pudo observar la matriz de confusión y una representación de la red neuronal entrenada.

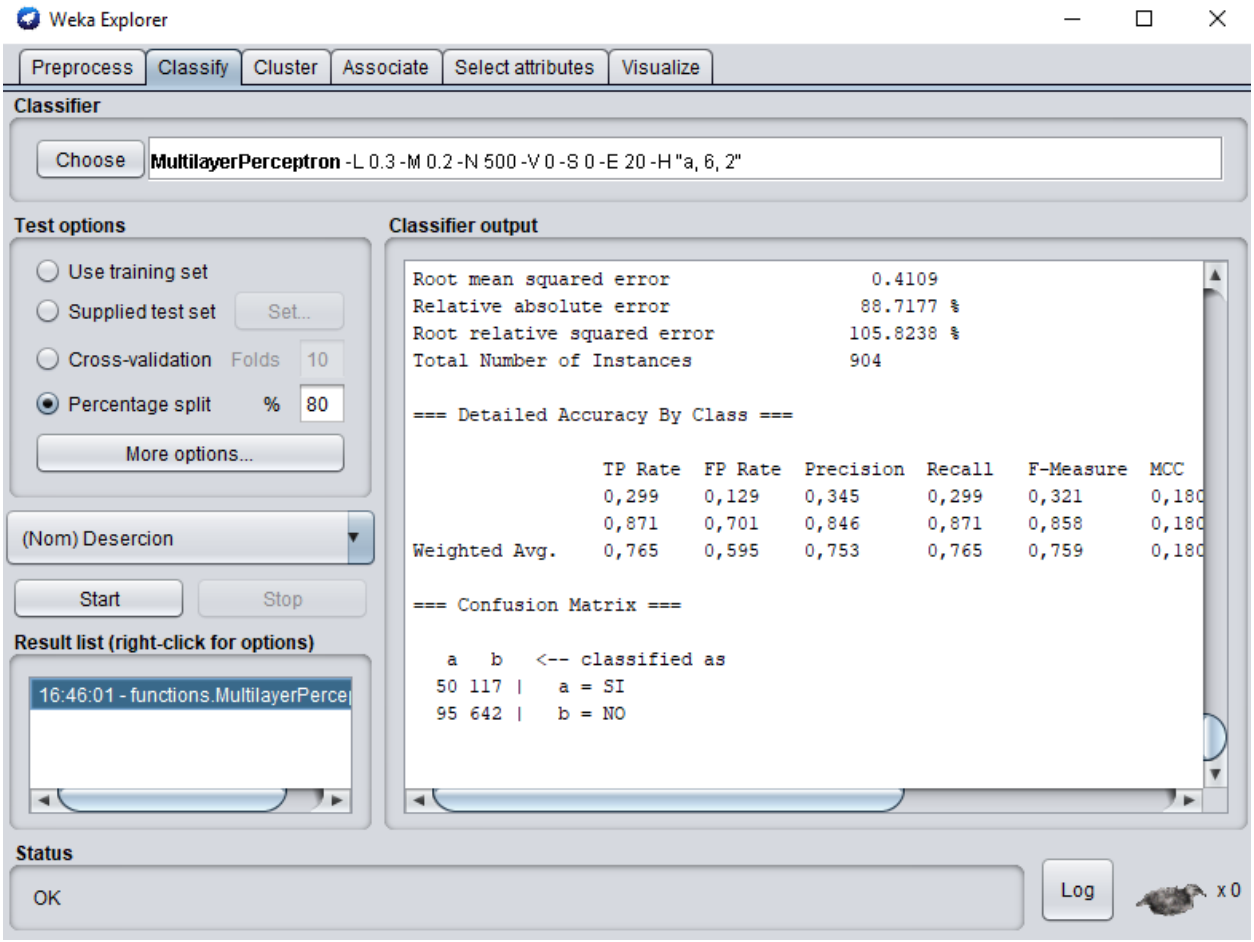


*Ilustración 10 Weka Explorer, Resultados de la clasificación con MultilayerPerceptron.*

### 3.5.2 Modelos SAP PA

Como se menciona en la revisión bibliográfica, SAP Preditive analytics es una herramienta de inteligencia de negocios que permite crear procesos de minería de datos de manera gráfica. Permitiendo a los usuarios diseñar un flujo de datos y ejecutarlo. En esta herramienta se pusieron a prueba algoritmos de árbol de decisión y red neuronal.

### *3.5.2.1 Árbol de decisión SAP PA*

En primer lugar, se importan los datos desde un archivo .xlsx donde se encuentra la totalidad de los datos (entrenamiento y prueba) y luego en la pestaña "preparación" se verifica que los datos estén correctos y estén las 11 variables definidas. El operador que representa el set de datos se ve de esta manera (Ilustración 11):


*Ilustración 11 Conjunto de datos en SAP PA.*

De acuerdo a la metodología adoptada, se utilizó un operador llamado "Sample" que sirve para particionar los datos importados, de esta manera se divide la muestra con 80% para datos de entrenamiento y el 20% para datos de prueba (ver Ilustración 12 y Tabla 16).


*Ilustración 12 Partición datos de entrenamiento SAP PA.*

*Tabla 16 Parámetros de la partición de entrenamiento.*

| Propiedades "Partition-train" | Valor | Descripción |
|---|---|---|
| **Tipo de muestra** | Primeros N | Seleccionar tipo de muestreo |
| **Limitar por fila** | Porcentaje de filas | Método para limitar filas |
| **Porcentaje de filas** | 80 | Porcentaje de filas que utilizará en la partición |
| **Filas máximas** | (opcional) | Cantidad de filas máximas |

Luego de particionar los datos de entrenamientos se agrega el operador R-CNR Tree el cual es la representación del algoritmo a entrenar (ver lustración 13) y se configura sus parámetros (ver tabla 17).

*Tabla 17 Parámetros del algoritmo R-CNR Tree de SAP PA.*

| Propiedades R-CNR Tree | Valor | Descripción |
|---|---|---|
| **Modo de salida** | Tendencia | Seleccionar el modo para mostrar los resultados |
| **Tipo de algoritmo** | Clasificación | Seleccionar tipo de análisis |
| **Funciones** | Seleccionar todas las variables menos la variable objetivo | Seleccionar columnas de entrada para el análisis |
| **Variable destino** | Deserción | Seleccionar la columna destino para el análisis |
| **Nombre de columna prevista** | PredictedValues | Nombre de la columna para datos pronosticados |
| **Valores pedidos** | Rpart | Seleccionar método para gestionar las entradas que faltan |
| **División mínima** | 10 | Número mínimo de observaciones para dividir un nodo |
| **Parámetro de complejidad** | 0.005 | Introducir el parámetro de complejidad que guarda el tipo de cálculo al evitar las divisiones que no mejoren el ajuste |
| **Profundidad máxima** | (opcional) | Introducir el nivel máximo del nodo en el árbol final |
| **Dividir criterios** | Gini | Seleccionar los criterios de división para el nodo |
| **Validación cruzada** | (opcional) | Introduzca el número de validaciones cruzadas |
| **Probabilidad anterior** | (opcional) | Introduzca el vector de probabilidades anteriores |
| **Usar sustituto** | (opcional) | Seleccionar los sustitutos que se usaron en el proceso de división |

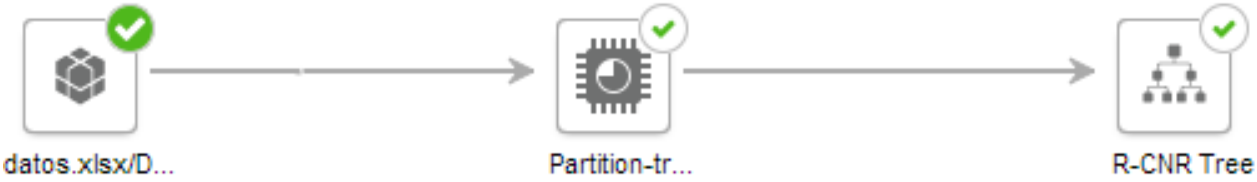


*Ilustración 13 R-CNR Tree de SAP PA.*

Luego, al presionar el botón de "ejecutar" se ejecuta el entrenamiento del modelo. Posterior a entrenar el modelo, este se guarda quedando disponible para su puesta a prueba con los datos de prueba. Nuevamente se utilizará una muestra para dividir los datos de prueba (ver ilustración 14) y se parametrizó como lo muestra la tabla 18.

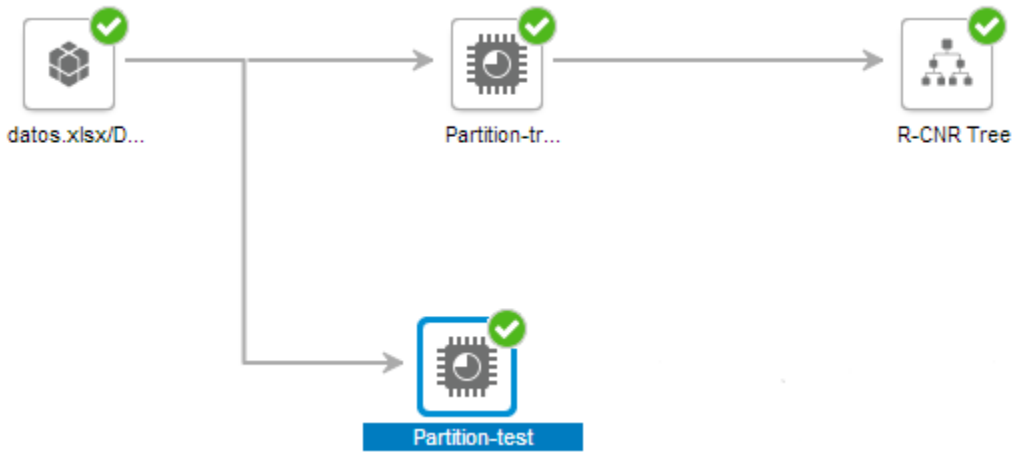


*Ilustración 14 Partición de datos de prueba SAP PA.*

*Tabla 18 Parámetros de la partición de prueba.*

| Parámetros "Partition-test" | Valor | Descripción |
|---|---|---|
| **Tipo de muestra** | Últimos N | Seleccionar tipo de muestreo |
| **Limitar por fila** | Porcentaje de filas | Método para limitar filas |
| **Porcentaje de filas** | 20 | Porcentaje de filas que utilizará en la partición |
| **Filas máximas** | (opcional) | Cantidad de filas máximas |

Por último, se agregó al diseño el modelo entrenado y un módulo de matriz de confusión (Ilustración 15). Estos últimos dos permitieron evaluar el modelo con un 20% de datos de prueba que el modelo desconoce y se parametrizaron como muestra la tabla 19 y 20 respectivamente. En

la opción “resultados” quedaron almacenados la matriz de confusión y la representación del modelo

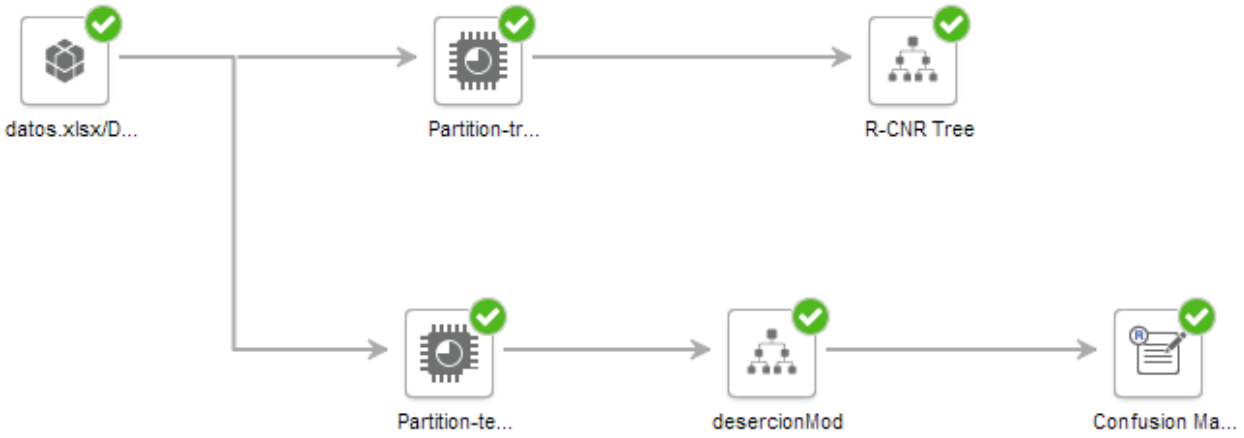


*Ilustración 15 Proceso de entrenamiento y prueba del árbol de decisión de SAP PA*

*Tabla 19 Parámetros del modelo entrenado.*

| Parámetros “desercionMod” | Valor | Descripción |
|---|---|---|
| **Modo de salida** | Predecir | Seleccionar el modo para mostrar los resultados |
| **Funciones** | Cada variable con su homónima | Seleccionar columnas de entrada para el análisis |
| **Nombre de columna prevista** | PredictedValues | Nombre de la columna para datos pronosticados |
| **Valores perdidos** | Rpart | Seleccionar método para gestionar las entradas que faltan |

*Tabla 20 Parámetros del módulo de matriz de confusión.*

| Parametros “Matrix Confusion” | Valor | Descripción |
|---|---|---|
| **Actuals** | Deserción | Variable objetivo |
| **Predictions** | PredictedValues | Columna de predicción |

#### *3.5.2.2 Red neuronal SAP PA*

SAP PA cuenta con dos algoritmos para el entrenamiento de redes neuronales. R-NNet y MONMLP R pero en el caso particular de esta investigación solo se utilizó el algoritmo

R-NNet ya que es el más utilizado. Este algoritmo permite entrenar una red neuronal de alimentación directa con una sola capa oculta, y para modelos multinomiales.

Primeramente y al igual que el algoritmo de árbol de decisión de SAP PA, se importa la totalidad de los datos verificando el tipo de las variables independientes y la cantidad de datos. Luego con el operador "Sample" se dividin los datos en dos muestras, una de entrenamiento con el 80% de los datos y la segunda para pruebas con un 20% de los datos (ver ilustración 16). Las muestras de entrenamiento y prueba son parametrizadas como se ve en la tabla 21 y 22 respectivamente

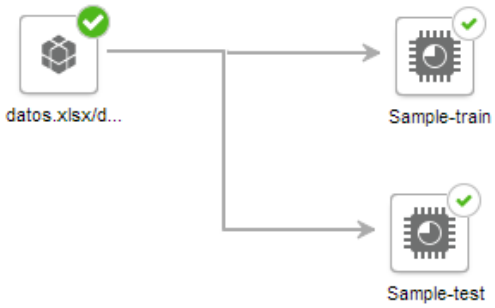


*Ilustración 16 Particiones de datos de entrenamiento y prueba en SAP PA*

*Tabla 21 Parámetros de la partición de entrenamiento.*

| Propiedades "Sample-train" | Valor | Descripción |
|---|---|---|
| **Tipo de muestra** | Primeros N | Seleccionar tipo de muestreo |
| **Limitar por fila** | Porcentaje de filas | Método para limitar filas |
| **Porcentaje de filas** | 80 | Porcentaje de filas que utilizará en la partición |
| **Filas máximas** | (opcional) | Cantidad de filas máximas |

*Tabla 22 Parámetros de la partición de prueba.*

| Parametros "Sample-test" | Valor | Descripción |
|---|---|---|
| **Tipo de muestra** | Últimos N | Seleccionar tipo de muestreo |
| **Limitar por fila** | Porcentaje de filas | Método para limitar filas |
| **Porcentaje de filas** | 20 | Porcentaje de filas que utilizará en la partición |
| **Filas máximas** | (opcional) | Cantidad de filas máximas |

Seguido de la partición de los datos, se agrega el operador R-NNet Neural Network para entrenar una red neuronal. Este operador se parametriza como muestra la tabla 23.

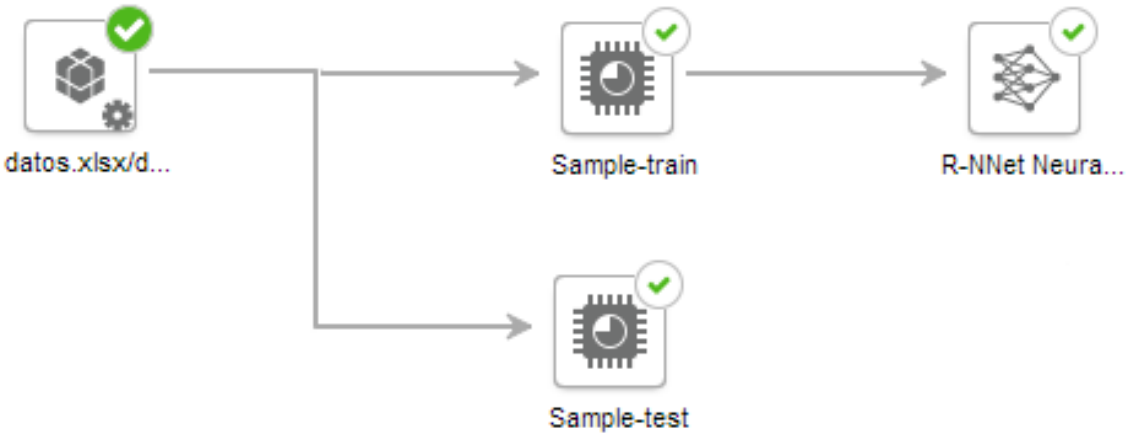


*Ilustración 17 R-NNet SAP PA.*

*Tabla 23 Parámetros de la red neuronal de SAP PA.*

| Parametros R-NNet | valores | descripción |
|---|---|---|
| **Modo de salida** | Tendencia | Seleccionar el modo para mostrar los resultados |
| **Tipo de algoritmo** | Clasificación | Seleccionar tipo de análisis |
| **Funciones** | Seleccionar todas las variables independientes menos la variable objetivo | Seleccionar columnas de entrada para el análisis |
| **Variable destino** | Deserción | Seleccionar la columna destino para el análisis |
| **Neuronas capa oculta** | 5 | Numero de nodos de la capa oculta |
| **Nombre columna prevista** | PredictedValues | Columna que contiene los valores previstos |
| **Valores perdidos** | Omitir | Método para gestionar las entradas que faltan |
| **Omitir capa oculta** | (opcional) | Agregar conexiones de capa |
| **Salida lineal** | Falso | Para obtener una salida lineal |
| **Usar softmax** | Verdadero | Para llevar a cabo la clasificación mediante el uso de los ajustes “modelo de registro lineal” |
| **Usar entropía** | (opcional) | Para usar el ajuste “Probabilidad condicional máxima” |
| **Usar censurado** | (opcional) | Variante de softmax |
| **Rango de peso aleatorio** | (opcional) | Rango inicial de pesos aleatorios |
| **Caída de peso** | (opcional) | Valor para calcular nuevos pesos |
| **Matriz Hessian necesaria** | Falso | Para obtener el indicado Hessian con el mejor conjunto de pesos |
| **Contrastes** | (opcional) | Seleccionar los contrastes que se usan para los factores que aparecen como variable en el modelo |
| **Repeticiones máximas** | (opcional) | Número máximo de iteraciones |
| **Pesos máximos** | (opcional) | Peso máximo |

Posterior al entrenamiento de la red neuronal, esta se guarda como modelo y se pone a prueba con la muestra de datos de prueba. Su rendimiento se puede ver a través del módulo

Confusión Matrix, que es un componente de R que no viene por defecto en SAP PA por lo que se debe que descargar e instalarlo. Este componente arroja una matriz de confusión con las predicciones de prueba. Por último, en la opción resultados se puede observar la matriz de confusión y representación del modelo de red neuronal.

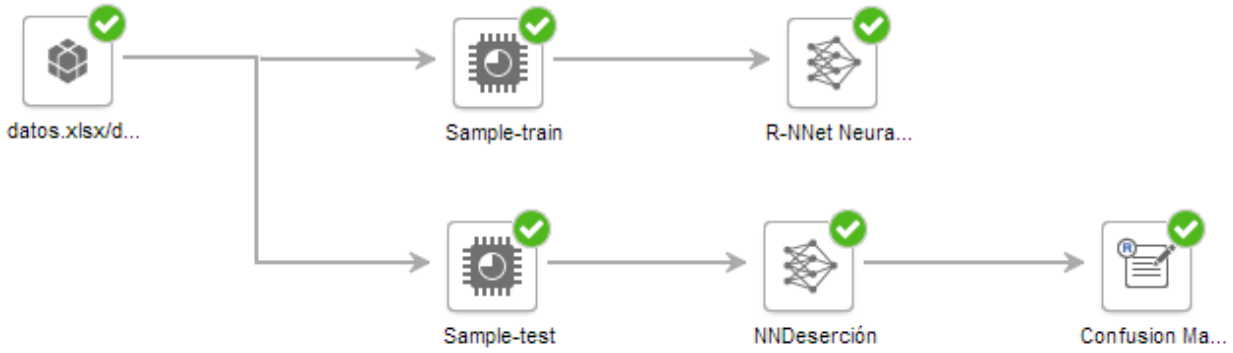


*Ilustración 18 Proceso de entrenamiento y prueba de la red neuronal de SAP PA.*

### 3.5.3 Modelos RapidMiner

Rapidminer es un software que permite el desarrollo de procesos de análisis de datos mediante el encadenamiento de operadores a través de un entorno gráfico. Al igual que los demás softwares dispone de árboles de decisión y redes neuronales. En las siguientes secciones se describe el proceso de construcción de los modelos en esta herramienta.

#### *3.5.3.1 Árbol de decisión Rapidminer*

RapidMiner dispone de un operador llamado “Decision Tree”. Este operador permite entrenar árbol de decisión y puede procesar los conjuntos de ejemplo que contienen atributos nominales y numéricos. Para poder ver el desempeño de este operador fue necesario seguir los siguientes pasos:

En primer lugar, se importa el set de datos con el operador Read Excel (ilustración 19), el cual lee un set de datos .xlsx. Este operador permite seleccionar las variables utilizadas en el estudio, verificar su tipo e identificar la variable objetivo su parametrización se muestra en la Tabla 24.

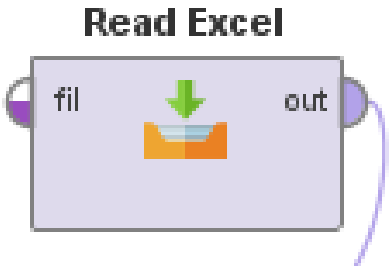


*Ilustración 19 Operador Read Excel de Rapid Miner.*

*Tabla 24 Parámetros operador Read Excel*.

| Parámetros "Read Excel" | Valor | Descripción |
|---|---|---|
| **import_configuration_wizard** | - | Esta opción le permite configurar este operador por medio de un asistente. Este asistente fácil de usar facilita el uso de este operador. |
| **excel_file** | Ubicación del archivo excel | La ruta del archivo de Excel se especifica aquí. Se puede seleccionar usando el botón elegir un archivo. |
| **sheet_selection** | sheet number | Esta opción le permite cambiar la selección de hoja entre el número de hoja y el nombre de la hoja. |
| **sheet_number** | 1 | El número de la hoja que desea importar debe especificarse aquí. |
| **imported_cell_range** | A1 | El rango de celdas a importar de la hoja especificada |
| **encoding** | system | Codificación utilizada para leer o escribir archivos. |
| **first_row_as_names** | Seleccionado | Si esta opción se establece en true, se supone que la primera línea del archivo Excel tiene los nombres de los atributos. Luego, los atributos se nombran automáticamente y la primera línea del archivo Excel no se trata como una línea de datos. |
| **date_format** | - | El formato de fecha y hora |
| **time_zone** | system | Este es un parámetro experto. Se proporciona una larga lista de zonas horarias; Los usuarios pueden seleccionar cualquiera de ellos. |
| **locale** | English (United States) | Este es un parámetro experto. Se proporciona una larga lista de locales; Los usuarios pueden seleccionar cualquiera de ellos. |
| **read_all_values_as_polynominal** | No seleccionado | Esta opción le permite deshabilitar el manejo de tipo para este operador. |
| **data_set_meta_data_information** | Aquí se definen los tipos de datos | Le permite ajustar los metadatos del ExampleSet creado a partir del archivo Excel especificado. |

Luego se agrega el operador "Split Data" el cual permite dividir los datos en dos muestras, una de entrenamiento y otra de prueba (véase Ilustración 20).

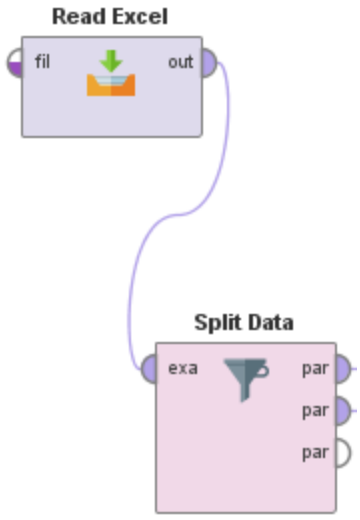


*Ilustración 20 Operador Split Data.*

Con los datos ya importados y divididos se agrega al proceso el operador “Decisión Tree”. Este operador genera un modelo de árbol de decisión, que puede utilizarse para la clasificación. Como se observa en la Ilustración 21, el operador “Spliy Data” entrega los datos de entrenamiento por medio de un flujo al operador “Decisión Tree”. Los parámetros para ese operador se muestran en la Tabla 25.

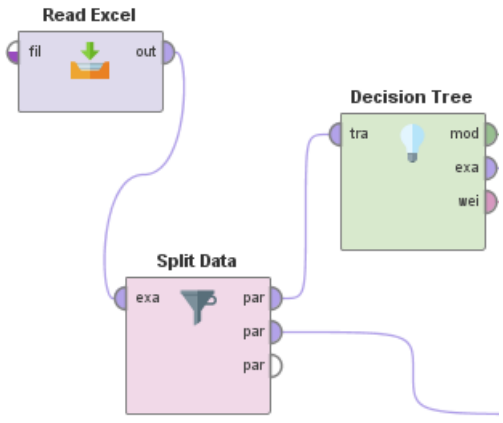


*Ilustración 21 Operador Decisión Tree de RapidMiner.*

*Tabla 25 Parámetros del operador Decisión Tree de Rapid Miner.*

| Parámetros "Decision Tree" | Valor | Descripción |
|---|---|---|
| **crirerion** | Information_gain | Selecciona el criterio en el que se seleccionarán los atributos para dividir |
| **maximal depth** | 20 | Este parámetro se utiliza para restringir la profundidad del árbol de decisión |
| **apply pruning** | Seleccionado | El modelo de árbol de decisión puede ser podado después de la generación. Si se marca, algunas ramas se reemplazan por hojas de acuerdo con el parámetro de confianza. |
| **confidence** | 0.25 | Este parámetro especifica el nivel de confianza utilizado para el cálculo del error pesimista de la poda. |
| **apply prepruning** | Seleccionado | Este parámetro especifica si se deben usar más criterios de detención |
| **minimal gain** | 0.01 | La ganancia de un nodo se calcula antes de dividirlo. El nodo se divide si su ganancia es mayor que la ganancia mínima. Un valor más alto de ganancia mínima produce menos divisiones y, por lo tanto, un árbol más pequeño. Un valor demasiado alto evitará completamente la división y se generará un árbol con un solo nodo. |
| **minimal leaf size** | 2 | El tamaño de una hoja es el número de ejemplos en su subconjunto. El árbol se genera de tal manera que cada hoja tenga al menos el número mínimo de Ejemplos de hojas. |
| **minimal size for split** | 4 | El tamaño de un nodo es el número de ejemplos en su subconjunto. Solo se dividen aquellos nodos cuyo tamaño es mayor o igual que el tamaño mínimo para el parámetro dividido |
| **number of prepruning alternatives** | 3 | este parámetro ajustará el número de nodos alternativos probados para la división |

Luego de entrenar los datos, se procedió a probar el modelo construido, para esto fue necesario utilizar otro operador llamado "Apply Model" que sirve para aplicar un set de prueba al modelo entrenado (ver ilustración 22). Este último operador se parametrizó como lo muestra la tabla 26.

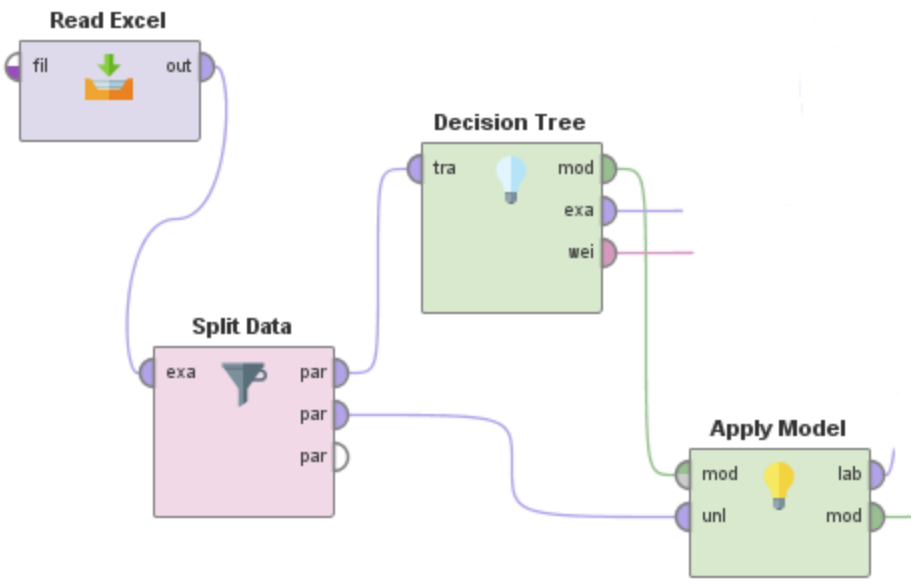


*Ilustración 22 Apply Model de RapidMiner.*

*Tabla 26 Parámetros del Operador Apply Model de RapidMiner.*

| Parametros "Apply Model" | Valor | Descripción |
|---|---|---|
| **Application parameters** | (opcional) | Este parámetro puede cambiar la configuración de ciertos modelos antes de que se apliquen al set de prueba proporcionado |
| **create view** | No seleccionado | Si esta opción está marcada, la aplicación del modelo se retrasa hasta que se necesiten las transformaciones |

Por último, con el fin de poder observar el rendimiento del modelo, se incluye un último operador llamado "Performance", este operador se utiliza para la evaluación del rendimiento estadístico de las tareas de clasificación y fue parametrizado como se muestra en la Tabla 27. Como se puede ver en la Ilustración 23 el operador "Performance", el operador "Apply Model" y el operador "Decisión Tree" tienen un flujo conectado a los puertos "res". Estos flujos de salida

son los que reciben el resultado del desempeño del modelo, el árbol de decisión y la importancia de las variables.

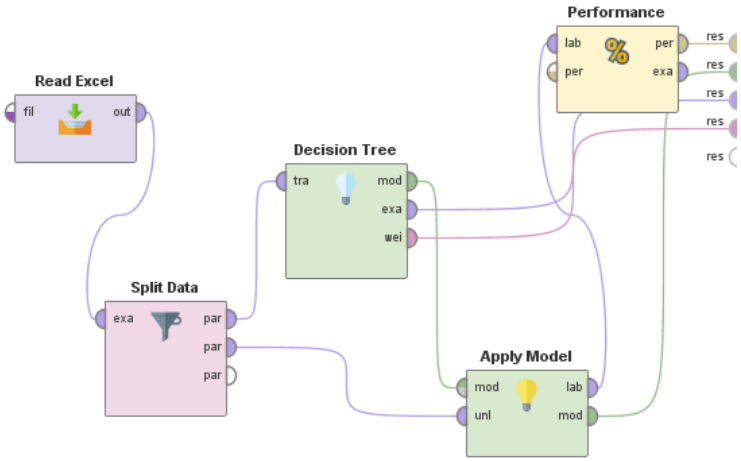


*Ilustración 23 Proceso de entrenamiento y prueba del árbol de decisión de RapidMiner.*

*Tabla 27 Parámetros del operador Performance de RapidMiner.*

| Parametros "Perfonrmance" | Valor | Descripción |
|---|---|---|
| **Accuracy** | Seleccionado | Número relativo de las instancias clasificadas correctamente o, en otras palabras, porcentaje de predicciones correctas |
| **Skip undefined labels** | Seleccionado | si se establece en true, las instancias con etiquetas indefinidas se omiten |
| **Use example weights** | seleccionado | este parámetro permite usar ejemplos de ponderaciones para cálculos de rendimiento estadístico, si es posible |

#### *3.5.3.2 Red neuronal Rapidminer*

Respecto a las redes neuronales, Rapidminer dispone de un operador llamado “Deep Learning” que entrena una red neuronal con un algoritmo llamado H2O. Este operador se basa en una red neuronal artificial de alimentación avanzada de múltiples capas que se entrena con el descenso de gradiente estocástico utilizando la propagación hacia atrás. La red puede contener un gran número de capas ocultas que consisten en neuronas con las funciones de activación de tanh, rectificador y maxout.

Primero para poder utilizar el operador de redes neuronales se importa el set de datos desde un archivo .xlsx, se verificaron las variables para el estudio y se identifica la variable objetivo; todo esto con el operador “Read Excel”.

*Tabla 28 Parámetros del operador Read Excel de RapidMiner*

| Parámetros "Read Excel" | Valor | Descripción |
|---|---|---|
| **import_configuration_wizard** | - | Esta opción le permite configurar este operador por medio de un asistente. Este asistente fácil de usar facilita el uso de este operador. |
| **excel_file** | Ubicación del archivo Excel | La ruta del archivo de Excel se especifica aquí. Se puede seleccionar usando el botón elegir un archivo. |
| **sheet_selection** | sheet number | Esta opción le permite cambiar la selección de hoja entre el número de hoja y el nombre de la hoja. |
| **sheet_number** | 1 | El número de la hoja que desea importar debe especificarse aquí. |
| **imported_cell_range** | A1 | El rango de celdas a importar de la hoja especificada |
| **encoding** | system | Codificación utilizada para leer o escribir archivos. |
| **first_row_as_names** | Seleccionado | Si esta opción se establece en true, se supone que la primera línea del archivo Excel tiene los nombres de los atributos. Luego, los atributos se nombran automáticamente y la primera línea del archivo Excel no se trata como una línea de datos. |
| **date_format** | - | El formato de fecha y hora |
| **time_zone** | system | Este es un parámetro experto. Se proporciona una larga lista de zonas horarias; Los usuarios pueden seleccionar cualquiera de ellos. |
| **locale** | English(United States) | Este es un parámetro experto. Se proporciona una larga lista de locales; Los usuarios pueden seleccionar cualquiera de ellos. |
| **read_all_values_as_polynominal** | No seleccionado | Esta opción le permite deshabilitar el manejo de tipo para este operador. |
| **data_set_meta_data_information** | Aquí se definen los tipos de datos | Le permite ajustar los metadatos del ExampleSet creado a partir del archivo Excel especificado. |

Para el entrenamiento y prueba de datos, se trabajó de la misma forma que el resto de los modelos, un 80% de los datos para entrenamiento y un 20% de los datos para las pruebas. Para lograr lo

anterior se utilizó el operador “Split Data”, el cual permite crear subconjuntos de datos a partir de un set de datos (ver ilustración 24 y Tabla 29).

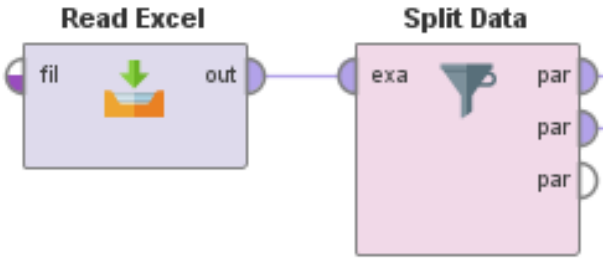


*Ilustración 24 Split Data de RapidMiner.*

*Tabla 29 Parámetros del operador Split data de RapidMiner.*

| Parámetros “Split data” | Valor | Descripción |
|---|---|---|
| **partitions** | 80% train<br>20% test | Este es el parámetro más importante de este operador. Especifica el número de particiones y la relación relativa de cada partición. |
| **Sampling type** | Linear sampling | Muestreo lineal:<br>El muestreo lineal simplemente divide el ExampleSet en particiones sin cambiar el orden de los ejemplos, es decir, se crean subconjuntos con ejemplos consecutivos. |
| **Use local random seed** | - | Indica si se debe utilizar una semilla aleatoria local para aleatorizar ejemplos de un subconjunto |

Seguido de la partición de los datos se agrega el operador “Deep Learning” (ver Ilustración 25) el cual es parametrizado (ver tabla 30), recibe los datos de entrenamiento y entrena la Red Neuronal.

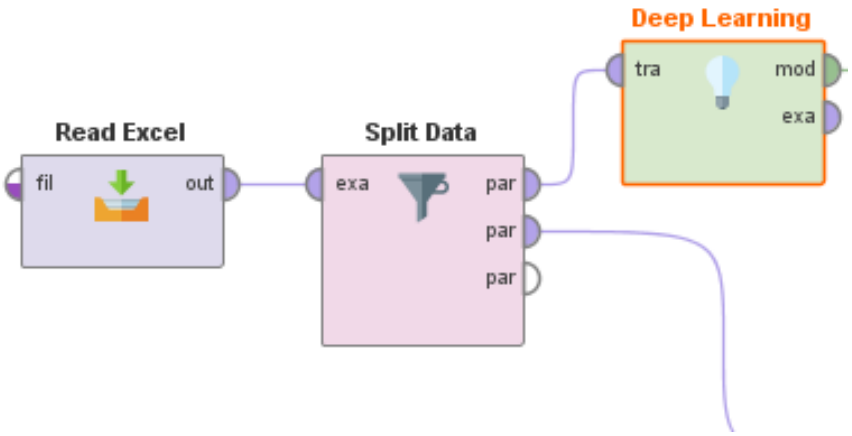


*Ilustración 25 Operador Deep Learning de RapidMiner.*

*Tabla 30 Parámetros del operador Deep Learning de RapidMiner.*

| Parámetros "Deep Learning" | Valor | Descripción |
|---|---|---|
| **activation** | tanh | Función de activación. |
| **Hidden layer sizes** | 6, 2 | Número y dimensión de capas ocultas. |
| **reproducible** | Sin seleccionar | Fuerza reproducibilidad en pequeños datos. |
| **epochs** | 10.0 | Cuántas veces el conjunto de datos debe ser iterado. |
| **Compute variable importances** | seleccionado | Calcular la importancia de las variables. |
| **Train simples per iteration** | -2 | El número de filas de datos de entrenamiento que se procesarán por iteración. |
| **Adaptive rate** | Seleccionado | Aprendizaje adaptativo. |
| **epsilon** | 1.0E-8 | Los valores típicos están entre 1e-10 y 1e-4. Este parámetro solo está activo si la velocidad de aprendizaje adaptativo está habilitada. |
| **rho** | 0.99 | Los valores típicos están entre 0.9 y 0.999. Este parámetro solo está activo si la velocidad de aprendizaje adaptativo está habilitada. |
| **standarize** | seleccionado | Si está habilitado, estandarice automáticamente los datos. Si está deshabilitado, el usuario debe proporcionar datos de entrada escalados adecuadamente. |
| **L1** | 1.0E-5 | Un método de regularización que restringe el valor absoluto de los pesos y tiene el efecto neto de eliminar algunos pesos de un modelo para reducir la complejidad y evitar el sobreajuste. |
| **L2** | 0.0 | Un método de regularización que restringe la suma de los pesos al cuadrado. |
| **max w2** | 10.0 | Un máximo en la suma de los pesos entrantes al cuadrado en cualquier nodo. |
| **loss function** | Automatic | La función de pérdida (error) debe ser minimizada por el modelo. |
| **distribution function** | AUTO | La función de distribución de los datos de entrenamiento. |
| **early stopping** | Sin seleccionar | Si es verdadero, se deben especificar los parámetros para la parada temprana. |

| | | |
|---|---|---|
| **Missing values handling** | MeanImputation | Manejo de valores faltantes. |
| **Max runtime seconds** | 0 | Tiempo de ejecución máximo permitido en segundos para entrenamiento modelo. Se utiliza 0 para deshabilitar. |

Seguido del entrenamiento del modelo se agrega el operador “Apply Model” quien recibe la Red Neuronal entrenada previamente y el set de datos de prueba que se particiona con el “Split Data”. Con los datos anteriores el “Apply Model” hace pruebas con las predicciones del modelo. Por último, y para poder observar el desempeño, se agrega el operador “Performance” quien arroja información estadística de las pruebas (ver ilustración 26). En la tabla 31 y 32 se puede ver la parametrización de los operadores “Appley Model” y “Performance” respectivamente.

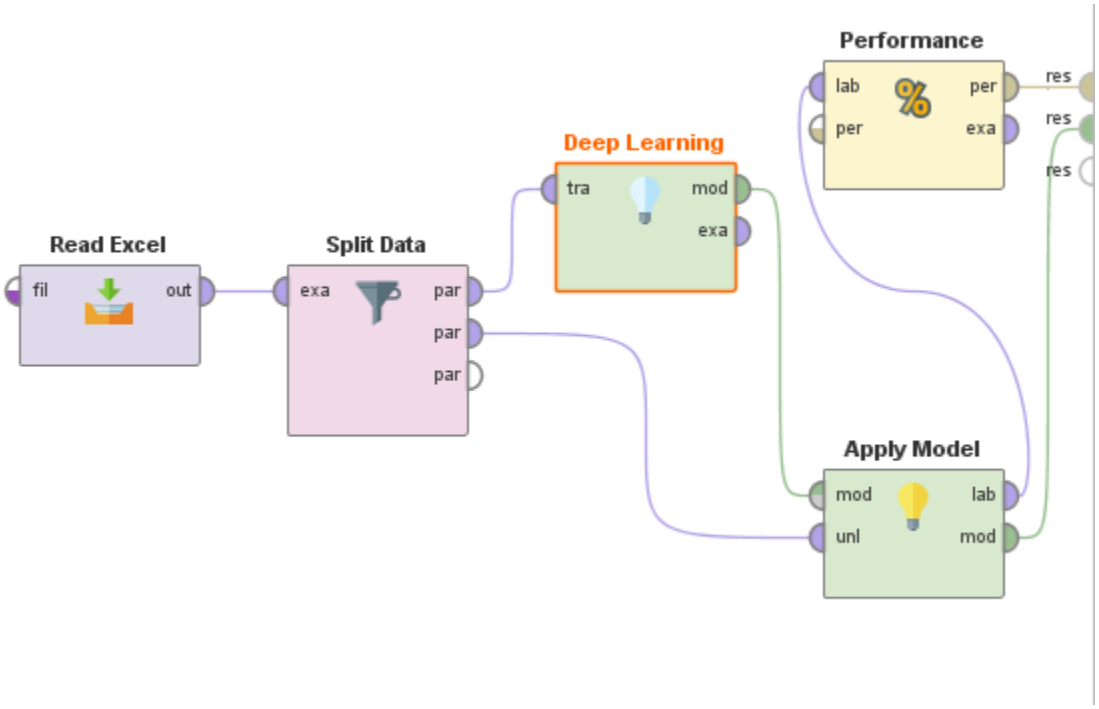


*Ilustración 26 Proceso de entrenamiento y prueba de la red neuronal de RapidMiner*.

*Tabla 31 Parámetros del operador Apply Model de RapidMiner.*

| Parámetros "Apply Model" | Valor | Descripción |
|---|---|---|
| **Application parameters** | (opcional) | Este parámetro puede cambiar la configuración de ciertos modelos antes de que se apliquen al set de prueba proporcionado |
| **create view** | No seleccionado | Si esta opción está marcada, la aplicación del modelo se retrasa hasta que se necesiten las transformaciones |

*Tabla 32 Parámetros del operador Performance de RapidMiner.*

| Parámetros "Perfonrmance" | Valor | Descripción |
|---|---|---|
| **Accuracy** | Seleccionado | Número relativo de las instancias clasificadas correctamente o, en otras palabras, porcentaje de predicciones correctas |
| **Skip undefined labels** | Seleccionado | si se establece en true, las instancias con etiquetas indefinidas se omiten |
| **Use example weights** | seleccionado | este parámetro permite usar ejemplos de ponderaciones para cálculos de rendimiento estadístico, si es posible |

# Capítulo 4: Resultados

## 4.1 Desempeño

Para analizar el desempeño de cada uno de los modelos entrenados, se extrajeron datos de la matriz de confusión y se calcularon los siguientes índices: Exactitud, Tasa de error, Sensibilidad, Especificidad, Precisión e Indice de Youden definidos en la sección 3.4 . A continuación, se muestra el análisis de desempeño de los modelos entrenados previamente.

### 4.1.1 Árbol de decisión Weka

La Tabla 33 muestra la matriz de confusión entregada por Weka entrenando un árbol de decisión con el algoritmo REPTree. Este modelo pudo predecir 13 estudiantes desertores de los 167 existentes en los datos de prueba.

*Tabla 33 Matriz de confusión modelo árbol decisión Weka..*

| | Predicción | | |
|---|---|---|---|
| **Observación** | **SI** | **NO** | **Total** |
| **SI** | 13 | 154 | 167 |
| **NO** | 18 | 719 | 737 |
| **Total** | 31 | 873 | 904 |

Como muestra la Tabla 34, el árbol de decisión de Weka arrojó un 81% de exactitud; lo que se traduce en que un 81% de las clasificaciones son correctas y 19% de ellas son erradas. La precisión de este modelo fue de 42%, lo que quiere decir que un 42% de las predicciones positivas son correctas.

*Tabla 34 Resultados análisis de rendimiento modelo árbol de decisión Weka.*

| Evaluación | Porcentaje |
|---|---|
| **Exactitud** | 81% |
| **Tasa de error** | 19% |
| **Sensibilidad** | 8% |
| **Especificidad** | 98% |
| **Precisión** | 42% |
| **Predicción negativa** | 82% |
| **Índice de Youden** | 0.053 |

### 4.1.2 Árbol de decisión SAP PA

La Tabla 35 muestra la matriz de confusión entregada por SAP PA entrenando un árbol de decisión con el algoritmo RCN-Tree. Este modelo pudo predecir 5 estudiantes desertores de los 162 existentes en los datos de prueba.

*Tabla 35 Matriz de confusión modelo árbol decisión SAP PA.*

| | Predicción | | |
|---|---|---|---|
| **Observación** | **SI** | **NO** | **Total** |
| **SI** | 5 | 157 | 162 |
| **NO** | 2 | 740 | 742 |
| **Total** | 7 | 897 | 904 |

Como muestra la Tabla 36, el árbol de decisión de SAP PA arrojó un 82% de exactitud; lo que se traduce en que un 82% de las clasificaciones son correctas y 18% de ellas son erradas. La precisión de este modelo fue de 71%, lo que quiere decir que un 71% de las predicciones positivas son correctas.

*Tabla 36 Resultados análisis de rendimiento modelo árbol de decisión SAP PA.*

| Evaluación | Porcentaje |
|---|---|
| **Exactitud** | 82% |
| **Tasa de error** | 18% |
| **Sensibilidad** | 3% |
| **Especificidad** | 100% |
| **Precisión** | 71% |
| **Predicción negativa** | 82% |
| **Índice de Youden** | 0.028 |

### 4.1.3 Árbol de decisión RapidMiner

La Tabla 37 muestra la matriz de confusión entregada por RapidMiner entrenando un árbol de decisión con el operador Decisión Tree. Este modelo pudo predecir 34 estudiantes desertores de los 159 existentes en los datos de prueba.

*Tabla 37 Matriz de confusión modelo árbol decisión RapidMiner*

| | Predicción | | |
|---|---|---|---|
| **Observación** | **SI** | **NO** | **Total** |
| **SI** | 34 | 125 | 159 |
| **NO** | 65 | 680 | 745 |
| **Total** | 99 | 805 | 904 |

Como muestra la Tabla 38, el árbol de decisión de RapidMiner arrojó un 79% de exactitud; lo que se traduce en que un 79% de las clasificaciones son correctas y 21% de ellas son erradas. La precisión de este modelo fue de 34%, lo que quiere decir que un 34% de las predicciones positivas son correctas.

*Tabla 38 Resultados análisis de rendimiento modelo árbol de decisión RapidMiner*

| Evaluación | Porcentaje |
|---|---|
| **Exactitud** | 79% |
| **Tasa de error** | 21% |
| **Sensibilidad** | 21% |
| **Especificidad** | 91% |
| **Precisión** | 34% |
| **Predicción negativa** | 84% |
| **Índice de Youden** | 0.127 |

### 4.1.4 Red Neuronal Weka

La Tabla 39 muestra la matriz de confusión entregada por Weka entrenando una Red Neuronal con el algoritmo MultilayerPerceptron. Este modelo pudo predecir 50 estudiantes desertores de los 167 existentes en los datos de prueba.

*Tabla 39 Matriz de confusión modelo red neuronal de Weka.*

| | Predicción | | |
|---|---|---|---|
| **Observación** | **SI** | **NO** | **Total** |
| **SI** | 50 | 117 | 167 |
| **NO** | 95 | 642 | 737 |
| **Total** | 145 | 759 | 904 |

Como muestra la Tabla 40, la Red Neuronal de Weka arrojó un 77% de exactitud; lo que se traduce en que un 77% de las clasificaciones son correctas y 23% de ellas son incorrectas. La precisión de este modelo fue de 34%, lo que quiere decir que un 34% de las predicciones positivas son correctas.

*Tabla 40 Resultados análisis de rendimiento modelo red neuronal de Weka.*

| Evaluación | Porcentaje |
|---|---|
| **Exactitud** | 77% |
| **Tasa de error** | 23% |
| **Sensibilidad** | 30% |
| **Especificidad** | 87% |
| **Precisión** | 34% |
| **Predicción negativa** | 85% |
| **Índice de Youden** | 0.171 |

### 4.1.5 Red Neuronal SAP PA

La Tabla 41 muestra la matriz de confusión entregada por SAP PA entrenando una Red Neuronal con el algoritmo RNNet. Este modelo no pudo predecir estudiantes desertores de los 156 existentes en los datos de prueba.

*Tabla 41 Matriz de confusión modelo red neuronal de SAP PA.*

| | Predicción | | |
|---|---|---|---|
| **Observación** | **SI** | **NO** | **Total** |
| **SI** | 0 | 156 | 156 |
| **NO** | 0 | 748 | 748 |
| **Total** | 0 | 904 | 904 |

Como muestra la Tabla 42, la Red Neuronal de SAP PA arrojó un 83% de exactitud; lo que se traduce en que un 83% de las clasificaciones son correctas y 17% de ellas son incorrectas. Sin embargo, ninguna de las clasificaciones fue de estudiantes desertores.

*Tabla 42 Resultados análisis de rendimiento modelo red neuronal de SAP PA.*

| Evaluación | Porcentaje |
|---|---|
| **Exactitud** | 83% |
| **Tasa de error** | 17% |
| **Sensibilidad** | 0% |
| **Especificidad** | 100% |
| **Precisión** | 0% |
| **Predicción negativa** | 83% |
| **Índice de Youden** | 0.000 |

### 4.1.6 Red Neuronal RapidMiner

La tabla 43 muestra la matriz de confusión entregada por RapidMiner entrenando una Red Neuronal con el operador Neural Net. Este modelo pudo predecir 98 estudiantes desertores de los 159 existentes en los datos de prueba.

*Tabla 43 Matriz de confusión modelo red neuronal de RapidMiner*

| | Predicción | | |
|---|---|---|---|
| **Observación** | **SI** | **NO** | **Total** |
| **SI** | 98 | 61 | 159 |
| **NO** | 280 | 465 | 745 |
| **Total** | 378 | 526 | 904 |

Como muestra la Tabla 44, la Red Neuronal de RapidMiner arrojó un 62% de exactitud; lo que se traduce en que un 62% de las clasificaciones son correctas y 38% de ellas son incorrectas. La precisión de este modelo fue de 26%, lo que quiere decir que un 26% de las predicciones positivas son correctas.

*Tabla 44 Resultados análisis de rendimiento modelo red neuronal de RapidMiner*

| Evaluación | Porcentaje |
|---|---|
| **Exactitud** | 62% |
| **Tasa de error** | 38% |
| **Sensibilidad** | 62% |
| **Especificidad** | 62% |
| **Precisión** | 26% |
| **Predicción negativa** | 88% |
| **Índice de Youden** | 0.241 |

## 4.2 Comparación de los modelos

A continuación, se comparan los índices de cada modelo. La exactitud es el índice que muestra la cantidad de clasificaciones correctas de todas las clases de la variable objetivo, en este caso la clasificación de desertores como no desertores. Esta variable se comportó de manera similar en la mayoría de los modelos, sin contar la red neuronal de Rapidminer tuvo el índice más bajo de un 62% (ver ilustración 27).

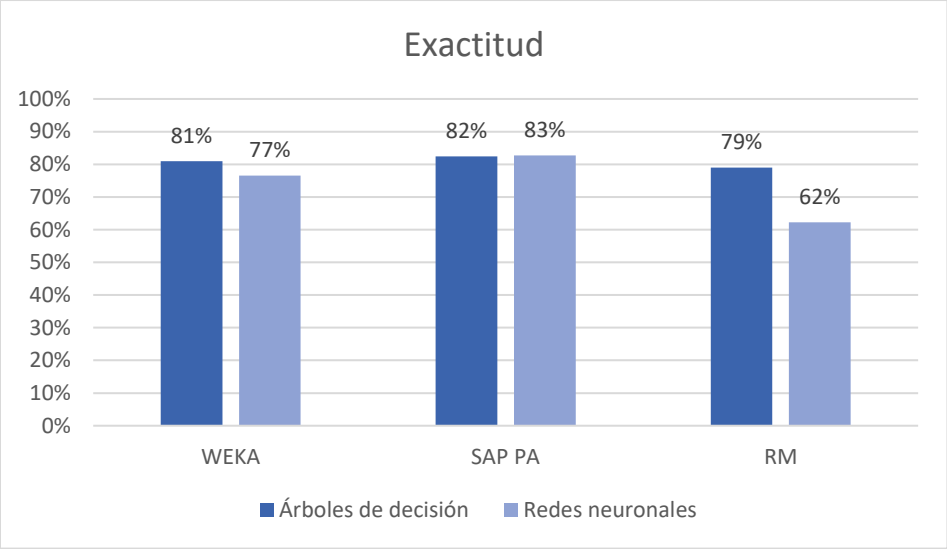


*Ilustración 27 Gráfica comparativa del índice de exactitud.*

La Tasa de error es el índice inverso a la exactitud por ende informa de las clasificaciones erradas de las dos clases de la variable objetivo. Al igual que en la exactitud, esta variable se comportó de manera similar en todos los modelos, sin embargo, tal como muestra la ilustración 28 la red neuronal de Rapidminer se diferenció de los demás modelos con un 38% de error.

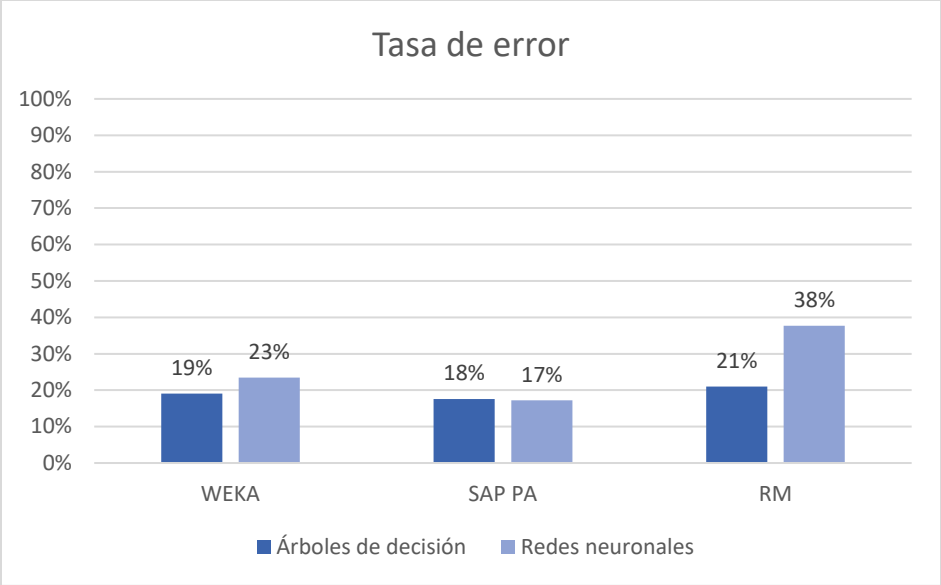


*Ilustración 28 Gráfica comparativa de la tasa de error.*

La Sensibilidad es uno de los índices más importantes en este estudio, ya que muestra la proporción de clasificaciones correctas de los estudiantes que SI desertaron respecto del total de desertores en la muestra de prueba. Como se puede apreciar en la ilustración 29, esta variable tuvo índices bajos para los árboles de decisión, sin embargo, las redes neuronales, en particular el de RapidMiner, tuvieron índices más altos, alcanzando un 62%.

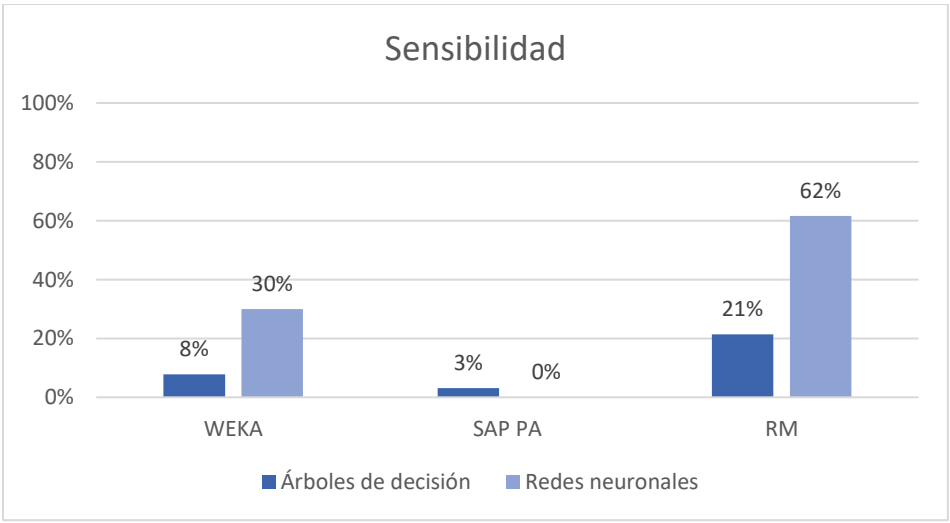


*Ilustración 29 Gráfica comparativa del índice de sensibilidad.*

La Especificidad es el índice que mide la cantidad de clasificaciones correctas para los estudiantes que NO desertaron. Como muestra la ilustración 30, este índice alcanza números bastante altos alcanzando 100% en los modelos de SAP PA. Esto se traduce en que los modelos tienen un buen desempeño a la hora de predecir a los estudiantes que NO desertarán.

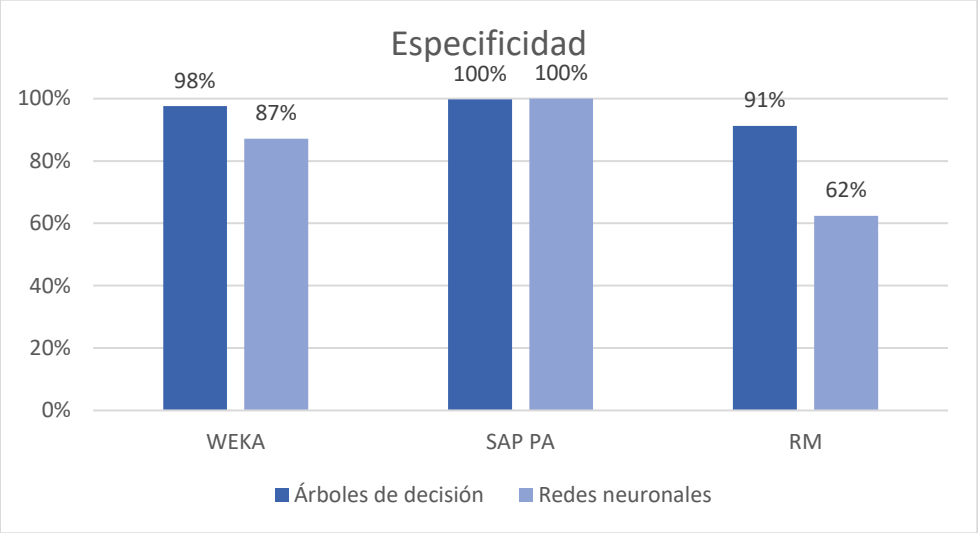


*Ilustración 30 Gráfica comparativa del índice de Especificidad.*

La Precisión es el índice que mide el porcentaje de predicciones positivos que son correctos, es decir, se enfoca en predecir los casos de estudiantes que desertarán. Esta variable tiene su mayor porcentaje en el árbol de decisión de SAP PA con un 71% y en algunos modelos como la red neuronal de SAP tuvo un 0% ya que estos modelos no pudieron predecir desertores.

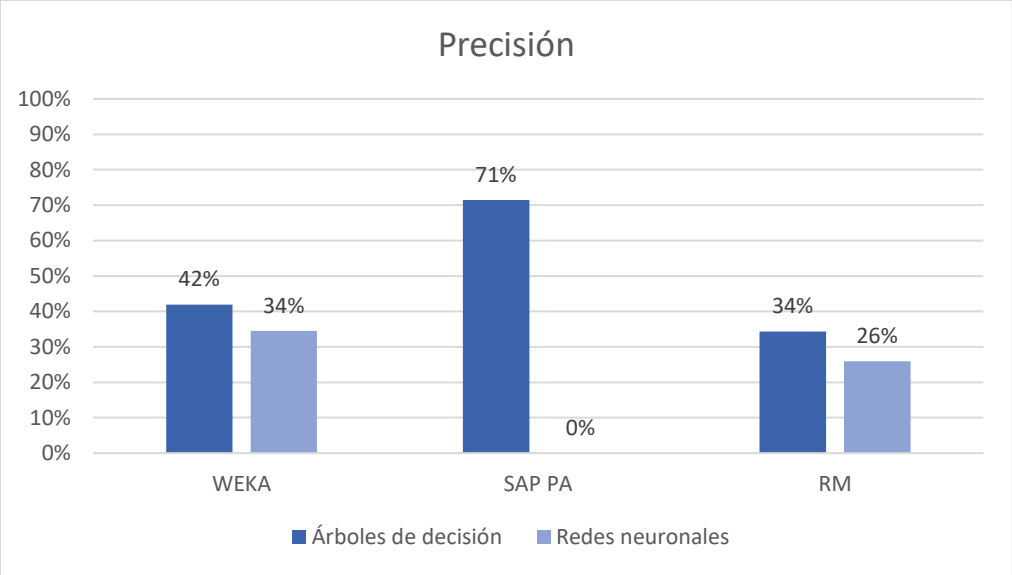


*Ilustración 31 Gráfica comparativa del índice de precisión.*

La predicción negativa es el índice inverso de la precisión pues mide el porcentaje de predicciones negativas que son correctas. Esta variable es similar en todos los modelos bordeando los 84%.

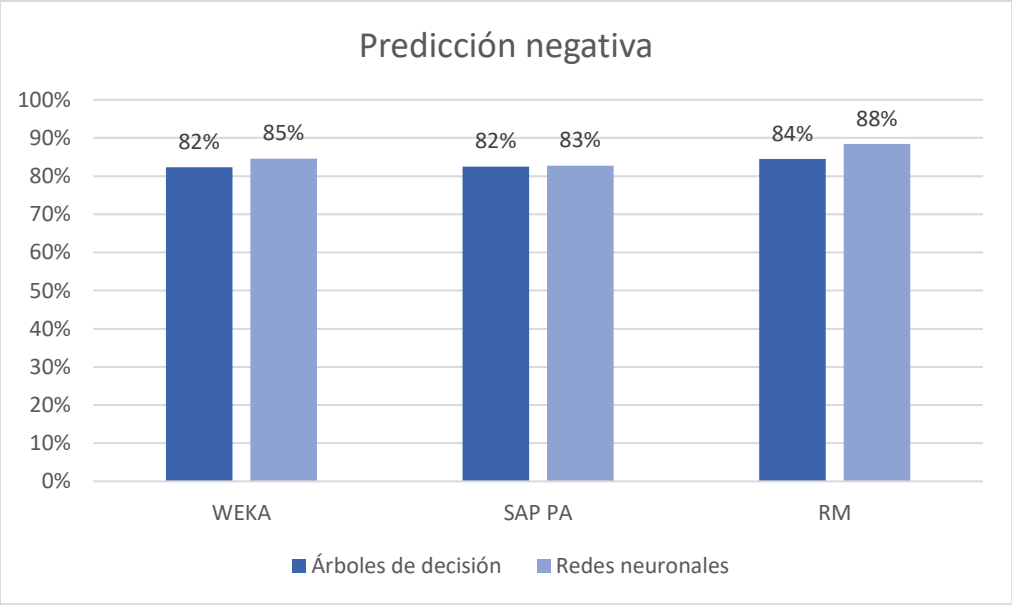


*Ilustración 32 Gráfica comparativa de la predicción negativa.*

Por último, se analiza el índice de Youden, el cual pretende resumir el desempeño de una prueba a un modelo. Su valor puede ser de -1 a 1 y tiene valor cero cuando una prueba da la misma proporción de resultados positivos para grupos desertores y no desertores. Un valor 1 indica que no hay falsos positivos o falsos negativos, por lo tanto, la prueba es perfecta. En este caso la mayoría de los modelos presentaron valores cercanos a 0 por lo que la prueba no arrojaría ningún resultado significativo.

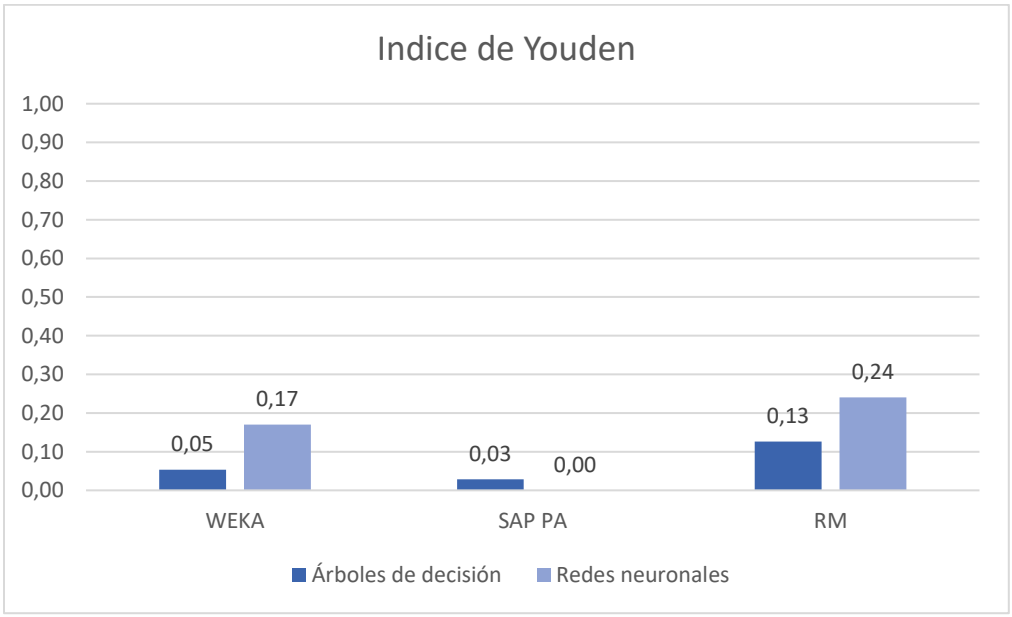


*Ilustración 33 Gráfica comparativa del índice de Youden.*

La siguiente tabla muestra una tabla resumen con los indicadores para cada herramienta analítica y técnica.

*Tabla 45 Comparación de resultados análisis de rendimiento.*

| | Árbol de decisión | | | Redes neuronales | | |
|---|---|---|---|---|---|---|
| | Weka | SAP PA | RM | Weka | SAP PA | RM |
| **Exactitud** | 81% | 82% | 79% | 77% | 83% | 62% |
| **Tasa de error** | 19% | 18% | 21% | 23% | 17% | 38% |
| **Sensibilidad** | 8% | 3% | 21%* | 30% | 0% | 62%* |
| **Especificidad** | 98% | 100% | 91% | 87% | 100% | 62% |
| **Precisión** | 42% | 71% | 34% | 34% | 0% | 26% |
| **Predicción Negativa** | 82% | 82% | 84% | 85% | 83% | 88% |
| **Índice de Youden** | 0.053 | 0.028 | 0.127 | 0.171 | 0.000 | 0.241 |

Entendiendo que el fin de este estudio es poder clasificar de manera correcta los estudiantes en el problema de la deserción, en especial a los estudiantes desertores, la sensibilidad es el índice más relevante ya que indica que porcentaje de estudiantes desertores pudo clasificar el modelo correctamente. En esta línea, el árbol de decisión y la red neuronal de RapidMiner son los modelos con mayores porcentajes de sensibilidad de sus respectivos tipos, por ende, son considerados los modelos con mejor desempeño. A continuación, se procede a analizar las variables importantes que estos modelos arrojan como resultado luego de su procesamiento.

### 4.3 Caracterización de los atributos de mayor relevancia para determinar la deserción.

La Tabla 46 muestra los pesos de las variables luego de ser procesadas en árbol de decisión de RapidMiner. Estos indican la importancia de las variables en el modelo, por ende, determinan la ubicación de las variables en el árbol de decisión.

*Tabla 46 Pesos de atributos árbol de decisión RapidMiner*

| Variables | Pesos |
|---|---|
| **NEM** | 0.36 |
| **PSU_Matemáticas** | 0.33 |
| **Gratuidad** | 0.1 |
| **Motivación_trabajar** | 0.055 |
| **Género** | 0.052 |
| **Orden_postulación** | 0.043 |
| **Región** | 0.03 |
| **Becas_arancel_MINEDUC** | 0.012 |
| **Vive_año_actual** | 0.012 |
| **Facultad** | 0.005 |

El árbol de decisión resultante es poco descriptivo, ya que es muy extenso, entonces no permite caracterizar el comportamiento de las variables. Para poder construir arboles más descriptibles se replica el análisis, pero ya no en la universidad en general, sino que, por cada una de las facultades, ya que como se evidencia la descripción del problema, la deserción se comporta de distinta forma en las facultades. Este último análisis arroja árboles mucho más descriptivos facilitando la caracterización del comportamiento de las variables.

La Tabla 47 muestra los pesos de las variables arrojados por los árboles entrenados por facultad y las variables están ordenadas desde la más importante a la menos importante. Según estos datos, el NEM, seguido por la PSU de matemáticas y por último el género, en promedio, son las variables más determinantes.

*Tabla 47. Pesos de las variables por facultad*

| Variables | FARCODI | FI | FACE | FEDUC | FACSA | FC |
|---|---|---|---|---|---|---|
| **NEM** | 0,314 | *0,515 | *0,282 | *0,590 | *0,465 | *0,551 |
| **PSU_matemáticas** | *0,391 | 0,294 | 0,273 | 0,141 | 0,441 | 0,289 |
| **Género** | 0,141 | 0,043 | 0,154 | 0,102 | 0,000 | 0,079 |
| **Motivación_trabajar** | 0,050 | 0,011 | 0,113 | 0,041 | 0,062 | 0,031 |
| **Gratuidad** | 0,083 | 0,085 | 0,013 | 0,029 | 0,025 | 0,019 |
| **Región** | 0,000 | 0,007 | 0,144 | 0,000 | 0,000 | 0,010 |
| **Becas_arancel_Mineduc** | 0,001 | 0,003 | 0,014 | 0,058 | 0,003 | 0,009 |
| **Orden_postulación** | 0,016 | 0,004 | 0,002 | 0,036 | 0,000 | 0,007 |
| **Vive_año_actual** | 0,004 | 0,038 | 0,004 | 0,002 | 0,004 | 0,006 |

*FARCODI: Facultad de Arquitectura, Construcción y Diseño, FI: Facultad de Ingeniería, FACE: Facultad de Ciencias Empresariales, FEDUC: Facultad de Educación y Humanidades, FACSA Facultad de Ciencias Salud y de los Alimentos, FC: Facultad de Ciencias.*

A modo de análisis se describen las ramas con mayores posibilidades de desertar de los árboles de decisión de la Facultad de Ciencias empresariales y la Facultad de Ciencias de la Universidad del Bío-Bío. La primera rama representa la rama que contiene los nodos hoja con mayor posibilidad de deserción de la Facultad de Ciencias Empresariales (véase Ilustración 34). La estructura de esta rama comienza en el nodo principal del árbol el cual clasifica primeramente a los estudiantes con puntajes de PSU de matemáticas menores o igual a 515.5pts.

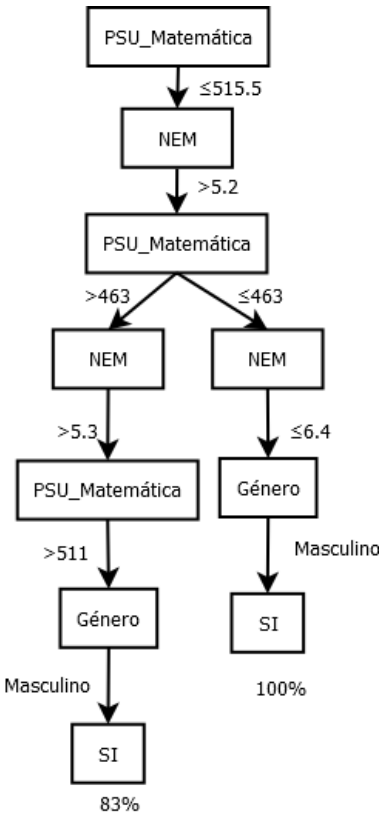


*Ilustración 34. Rama con mayor posibilidad de deserción del árbol de decisión de la FACE.*

Según esta rama los estudiantes con mayor posibilidad de deserción en la FACE son los siguientes:

- Estudiantes hombres con un puntaje de PSU de matemáticas entre los 511pts y 515.5pts y un NEM sobre 5.3 tienen un 83% de posibilidad de desertar.
- Estudiantes hombres con puntaje de PSU de matemáticas menor a 463pts y NEM menor a 6.4 tienen un 100% de posibilidad de desertar.

La segunda rama representa la rama que contiene los nodos hoja con mayor posibilidad de deserción de la Facultad de Ciencias (véase Ilustración 35). La estructura de esta rama comienza en el nodo principal del árbol el cual clasifica primeramente a los estudiantes con puntajes de PSU de matemáticas mayor y menores o igual a 487pts.

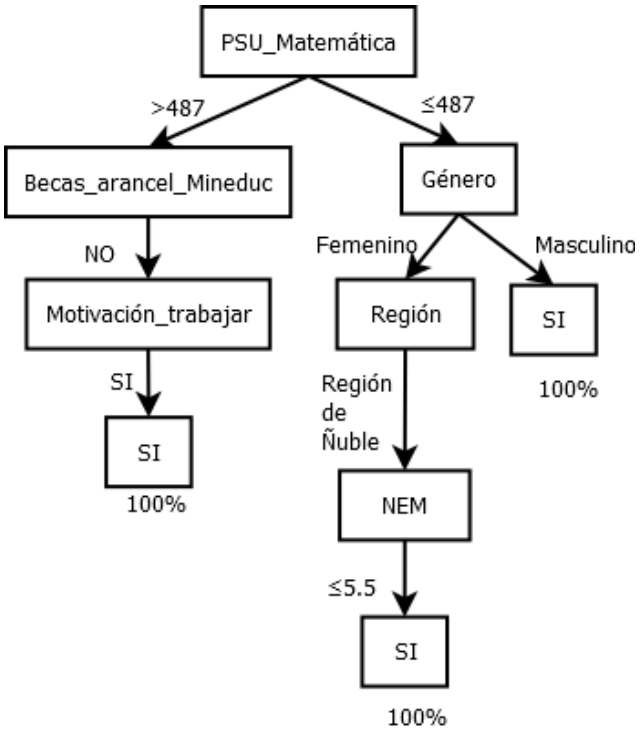


*Ilustración 35. Rama con mayor posibilidad de deserción en la FC.*

Según esta rama los estudiantes con mayor posibilidad de deserción en la FACE son los siguientes:

- Estudiantes con un puntaje de PSU de matemáticas superior a 457pts, sin becas de arancel MINEDUC y con necesidad de trabajar mientras estudia tiene un 100% de posibilidad de desertar.
- Estudiantes mujeres de la región de Ñuble con un puntaje de PSU inferior a 487 y un NEM inferior a 5.5 tiene un 100% de posibilidad de desertar.
- Estudiantes hombres con puntaje de PSU inferior a 5.5 tiene un 100% de posibilidad de desertar.

Respecto de la importancia de las variables en la red neuronal entrenada en RapidMiner, es complejo de analizar, ya que una red neuronal es una caja negra en el sentido de que, aunque se puede crear un modelo con un muy buen desempeño, el estudio de su estructura, no da ningún

conocimiento de cómo se comportan las variables. De todas maneras, esta red neuronal tiene un método que permite mostrar la importancia de la variable representada en pesos.

La Ilustración 36 muestra importancia de las variables de entrada en la red neuronal entrenada en RapidMiner. La importancia de las variables en este modelo considera los pesos que conectan las variables de entrada a las dos primeras capas ocultas de la red neuronal.

Variable Importances:

| Variable | Relative Importance | Scaled Importance | Percentage |
|---|---|---|---|
| Becas_arancel_MINEDUC.NO | 1.000000 | 1.000000 | 0.041023 |
| Región.3 | 0.911441 | 0.911441 | 0.037390 |
| PSU_Matemáticas | 0.894179 | 0.894179 | 0.036682 |
| Facultad.2 | 0.793426 | 0.793426 | 0.032549 |
| Motivación_trabajar.NO | 0.748869 | 0.748869 | 0.030721 |
| Región.16 | 0.737491 | 0.737491 | 0.030254 |
| Orden_postulación.0 | 0.712118 | 0.712118 | 0.029213 |
| Región.2 | 0.672742 | 0.672742 | 0.027598 |
| Facultad.1 | 0.670721 | 0.670721 | 0.027515 |
| Facultad.5 | 0.648688 | 0.648688 | 0.026611 |

*Ilustración 36, pesos de las variables importantes de la red neuronal de RM.*

Las variables que muestran mayor importancia luego del procesamiento de la red neuronal son, la Beca de arancel MINEDUC (NO = Sin becas), Región (3 = Región de Atacama y 2 = Región de Antofagasta), PSU de matemáticas, Facultad (2 = Facultad de Ingeniería, 1 = Facultad de Arquitectura, Construcción y Diseño y 5 = Ciencias de la Salud y de los Alimentos), Motivación a trabajar (NO = sin necesidad de trabajar).

# Capítulo 5: Discusión

A continuación, se presenta la discusión de los resultados obtenidos en esta investigación. Esta discusión está orientada a buscar el modelo que obtuvo mejor rendimiento frente a las pruebas aplicadas y sus atributos más determinantes, por ende, a medida que avanza el desarrollo de las ideas se van identificando los índices de desempeño más importantes y se van descartando los modelos que tengan menores rendimientos. Luego de seleccionar los modelos con mejores rendimientos se analiza la importancia de los atributos.

La exactitud y la tasa de error son índices que se comportan de manera similar en todos los modelos puestos a prueba. En promedio, la exactitud bordea el 84% y la tasa de error un 20%, números bastante buenos. Sin embargo, estos índices no reflejan el desempeño para predecir posibles estudiantes desertores, ya que la exactitud y la tasa de error consideran clasificaciones correctas en estudiantes desertores como no desertores.

La sensibilidad es un índice muy importante para comparar los modelos, puesto que refleja el porcentaje de los desertores clasificados correctamente respecto del total de desertores en la muestra de prueba. En este caso la red neuronal de RapidMiner es el que presenta mejor índice de sensibilidad con un 62% seguido por la red neuronal de Weka con un 30%. Otro es el caso de los árboles de decisión, el árbol de decisión de SAP PA obtuvo 3%, el árbol de RapidMiner un 21% y el de Weka un 8%.

La especificidad presentó números bastante altos en todos los modelos, por sobre los 87%. Esto quiere decir que los modelos son capaces de predecir correctamente gran cantidad de NO desertores, sin embargo, este índice no es de gran importancia porque la investigación está enfocada en la predicción de los desertores. Lo mismo sucede con la Predicción negativa que es

un índice que muestra el porcentaje de predicciones negativas que son correctamente clasificados como muestras negativas.

La precisión es una variable que también hay que considerar ya que indica la calidad de la predicción de estudiantes que SI desertaron. El árbol de decisión de Weka es el que presenta mayor porcentaje de precisión con un 71%, sin embargo, su sensibilidad es muy baja.

Respecto de las redes neuronales, la red neuronal de RM es la que presenta mejor rendimiento clasificando estudiantes desertores, con una sensibilidad de 62%, la sigue la red neuronal de Weka con un 30% y por último la red de SAP PA que no tiene la capacidad de predecir estudiantes desertores. Respecto de los árboles de decisión, el árbol de RM es el que presenta mejor rendimiento clasificando estudiantes desertores, con una sensibilidad de 21%, seguido por el modelo de Weka con un 8% y por último el árbol de SAP PA con un 3%. El árbol de decisión y la red neuronal de RM poseen los mejores rendimientos en sus respectivos tipos.

Según el árbol de decisión entrenado con toda la muestra en RM, la PSU de matemáticas, NEM y la gratuidad son los atributos más importantes, sin embargo, el árbol resultante no es descriptivo, debido a su extensión, entonces no permite caracterizar los atributos. Por lo anterior, se divide la muestra por facultades y se entrenan arboles más descriptibles. Los arboles resultantes por facultad tiene distintos pesos para cada atributo, pero en promedio los atributos más importantes son el NEM, la PSU de matemáticas y Género. Según la red neuronal entrenada en RM, las Becas de arancel MINEDUC, Región, PSU de matemáticas son los atributos más importantes, sin embargo, no es posible caracterizarlos ya que la red neuronal no da indicios de las características de los atributos.

# Capítulo 6: Conclusiones

En este capítulo se presenta las conclusiones obtenidas a partir de la realización de este trabajo. Primero se muestra el cumplimiento de objetivos de la investigación y luego las conclusiones generales.

## 6.1 Cumplimiento de objetivos.

Como se mencionó en el capítulo 2, el objetivo general de este trabajo es "Generar un modelo de predicción para la deserción estudiantil en la Universidad del Bío-Bío a través del análisis de diferentes herramientas de predicción analítica". A continuación, se muestran los objetivos específicos y la manera en que dio el cumplimiento a estos.

- Realizar revisión literaria sobre deserción estudiantil.

  Se realizó una revisión bibliográfica con la cual se pudieron reunir los principales conceptos respecto de la deserción y herramientas de minería de datos. Además, se recopilaron y detallaron los modelos de deserción desarrollados previamente en la universidad.
- Estudiar modelos actuales para predicción de deserción estudiantil en la Universidad del Bío-Bío

  Se estudiaron y describieron los trabajos de Pérez et al. (2018) y Retamal & Rubilar (2017) , quienes estudiaron la deserción estudiantil en la UBB con árboles de decisión y regresiones logísticas respectivamente.
- Estudiar algoritmos de predicción aplicables para el caso de la deserción estudiantil en la Universidad del Bío-Bío.

Se estudiaron y describieron algoritmos de árboles de decisión, redes neuronales y regresión logística, todos ellos aplicados al problema de la clasificación de estudiantes desertores.

- Implementar modelos de predicción en diferentes herramientas analíticas.

  Se entrenaron árboles de decisión y redes neuronales en tres softwares de minería de datos.

- Comparar resultados de los modelos generados.

  Se compararon análisis de rendimiento de cada uno de los modelos entrenados y se caracterizaron sus atributos determinantes.

- Proponer un nuevo modelo de predicción basado en la comparación efectuada.

  En base a la comparación de análisis de rendimiento se propone el modelo con mejores resultados.

## 6.2 Limitaciones

Este estudio presenta algunas limitaciones tales como el tamaño de la muestra, que podría ampliarse, así como hacer más extensivo este estudio a los contextos particulares de las carreras. También los atributos definidos, son restrictivos en tanto quedan limitados a respuestas específicas.

## 6.3 Trabajos Futuros.

Basándose en este trabajo, se recomienda los siguientes trabajos futuros:

- Entrenar un modelo de clasificación para la predicción de la deserción estudiantil en la UBB en Rapidminer, utilizando distintas variables a las que se utilizaron en este trabajo y una muestra más grande.

- Ejecutar el análisis utilizando distintos algoritmos de clasificación disponibles en las herramientas utilizadas en este estudio.

Basándose en trabajo similares, se recomiendan los siguientes trabajos futuros:

- Entrenar modelos de predicción para el desempeño de los estudiantes analizando resultados y calificaciones
- Entrenar modelos de predicción para estimación de las habilidades adquiridas por el alumno.
- Entrenar modelos para la creación de cursos/secciones en base a las características de los estudiantes.

**6.4 Conclusiones generales.**

A modo de síntesis, se resumen las principales ideas generadas durante el desarrollo del presente trabajo. A partir de ellas se podrán reconocer con qué nivel fueron alcanzados los objetivos propuestos. En base a la comparación del desempeño de los modelos de deserción entrenados en este estudio, se concluye que la red neuronal y el árbol de decisión de Rapidminer son los que arrojan mejores resultados en sus categorías. Estos modelos obtienen un índice de sensibilidad de 62% y 21% respectivamente. Según los árboles de decisión entrenados por facultad, el NEM, la PSU de matemáticas y el género son los atributos más importantes y se comporta de distintas maneras en las facultades. Los modelos seleccionados pueden predecir con precisión a los estudiantes que no desertaron, sin embargo, su precisión es baja para predecir a los posibles desertores. Si bien ninguno de los dos modelos mencionados anteriormente es confiable para clasificar a posibles desertores, se propone que se complemente el set de datos utilizados en

este trabajo y se entrenen nuevamente los dos algoritmos de Rapidminer, ya que fueron los que presentaron mejores resultados.

## Referencias


A. Fishbein, M., & Ajzen, I. (1975). *Belief, attitude, intention and behaviour: An introduction to theory and research*.

Castaño, E., Gallón, S., Gómez, K., & Vásquez, J. (2004). *Deserción estudiantil universitaria: una aplicación de modelos de duración*. *Lecturas de Economía*. Antioquia.

Dekker, G. W., Pechenizkiy, M., & Vleeshouwers, J. M. (2009). *Predicting Students Drop Out: A Case Study*.

Ethington, C. A. (1990). A Psychological Model of Student Persistence. *Research in Higher Education*.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Fayyad, U. (1997). Data mining and knowledge discovery in databases: implications for scientific databases (pp. 2–11).

Fischer, E. (2012). *Modelo para la automatización del proceso de determinación de riesgo de deserción en estudiantes universitarios*. Santiago de Chile.

Fiuza, M. D., & Rodríguez, J. C. (2000, diciembre 1). La regresión logística: una herramienta versátil. *Nefrología*, *20*(6), 477–565.

Hernández, J., Ramírez, M. J., & Ferri, C. (2004). *Introducción a la minería de datos*. Pearson Educación.

Himmel, E. (2002). Modelo de análisis de la deserción estudiantil en la educación superior. *Calidad en la Educación*, *0*(17), 91.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression.*

IESALC. (2007). *Informe sobre la Educación Superior en América Latina y el Caribe (2000-2005)*. Caracas.

Izaurieta, F., & Saavedra, C. (2019). *Redes Neuronales Artificiales*.

M. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). *From Data Mining to Knowledge Discovery in Databases*. *AI Magazine* (Vol. 17).

Maimon, O., & Rokach, L. (2005). *Data Mining and Knowledge Discovery Handbook*. Berlin, Heidelberg: Springer-Verlag.

Matich, D. (2001). *Redes Neuronales: Conceptos Básicos y Aplicaciones*. Rosario.

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006). *YALE: Rapid Prototyping for Complex Data Mining Tasks*.

Mojarrango, N., & Chapalbay, J. (2016). *Análisis Comparativo de las Plataformas Weka y Microsoft Analysis Services para Optimizar el Desarrollo De Minería de Datos en la Empresas PRASOL "Lácteos Santillán".* Riobamba.

Molina, J., & García, J. (2006). Técnicas de análisis de datos.

Nigro, H. O., Xodo, D., Corti, G., & Terren, D. (s. f.). *KDD (Knowledge Discovery in Databases): Un proceso centrado en el usuario*. Tandil.

OECD. (2018). *Education at a Glance 2018: OECD Indicators*. Paris: OECD Publishing.

P. Bean, J. (1985). *Interaction Effects Based on Class Level in an Explanatory Model of College Student Drop Out Syndrome*. *American Educational Research Journal - AMER EDUC RES J* (Vol. 22).

Pérez, A., Grandón, E., & Vargas, G. (2018). Análisis Comparativo de Técnicas de Predicción para Determinar la Deserción Estudiantil : Regresión Logística vs Árboles de Decisión.

Ramírez, P. E., & Grandón, E. E. (2018). Predicción de la Deserción Académica en una Universidad Pública Chilena a través de la Clasificación basada en Árboles de Decisión con Parámetros Optimizados. *Formación universitaria*, *11*(3), 3–10.

Rayo Cantón, S., Lara Rubio, J., & Camino Blasco, D. (2010). Un Modelo de Credit Scoring para instituciones de microfinanzas en el marco de Basilea II. *Journal of Economics, Finance and Administrative Science*.

Retamal, J. C., & Rubilar, A. M. (2017). Desarrollo de un Modelo de deserción para los estudiantes de primer año de la Universidad del Bío Bío.

Rokach, L., & Maimon, O. (2008). *Data mining with decision trees : theory and applications*. World Scientific.

Rouse, M. (2016). ¿Qué es SAP Predictive Analytics?

Salas, M. (1996). *La regresión logística. Una aplicación a la demanda de estudios universitarios* (Vol. 38). Instituto Nacional de Estadística.

Salas, R. (2017). *Redes Neuronales Artificiales*. Valparaíso.

SIES. (2018). Informe Retención de 1er año de Pregrado, 9. Recuperado de www.mifuturo.cl.

Simpson, S. D. (2004). *A Study of Attrition in Higher Education with Implications for Supportive Services*.

Spady, W. G. (1970). Dropouts from higher education: An interdisciplinary review and synthesis. *Interchange*, *1*(1), 64–85.

Tinto, V. (1975). Dropout from Higher Education: A Theoretical Synthesis of Recent Research. *Review of Educational Research*, *45*(1), 89–125.

Tinto, V. (1982). Definir la deserción: una cuestión de perspectiva, 1–9.

Tinto, V. (1989). *Misconceptions Mar Campus Discussions of Student Retention - The Chronicle of Higher Education* (1ª ed.). Academic Research Library.

Trujillano, J., Sarria-Santamera, A., Esquerda, A., Badia, M., Palma, M., & March, J. (2008). Aproximación a la metodología basada en árboles de decisión (CART): Mortalidad hospitalaria del infarto agudo de miocardio . *Gaceta Sanitaria* . scieloes .

Weiss, S. M., & Indurkhya, N. (1998). *Predictive data mining : a practical guide*. Morgan Kaufmann Publishers.

Witten, I. H., Frank, E., Trigg, L., Hall, M., Holmes, G., & Cunningham, S. J. (1999). *Weka: Practical Machine Learning Tools and Techniques with Java Implementations*.

Yukselturk, E. (2014). Pedicting Dropout Student: An Application of Data Mining Methods in an Online Education Program. *European Journal of Open, Distance and e-Learning*, *17*(1).