UNIVERSIDAD SANTO TOMAS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

Facultad de Estadística
Trabajo de Grado
Febrero 2017

Comparación de modelos apilados bajo los esquemas de redes neuronales y árboles de clasificación.

Comparison of models stacked below the schemes of neural networks and classification trees.

Jhon Alexander Rincón Olmos.[a]
jhonrincon@usantotomas.edu.co

Daniel Leonardo Cruz Castro.[b]
daniel.cruz@usantotomas.edu.co

Resumen

Las redes neuronales y los árboles de decisión han demostrado ser métodos eficientes en aplicaciones relacionadas con la clasificación. En la actualidad siguen intentando combinar estos métodos con el propósito de obtener un algoritmo más eficiente que sea capaz de mejorar la precisión de sus estimaciones, por esta razón esta tesis tiene como propósito crear y comparar el rendimiento de los modelos anteriores, mediante la combinación de diferentes algoritmos de decisión a través del *método stacking* o *apilamiento*, el cual fue desarrollado por Wolper en 1992 para aumentar la precisión predictiva de los modelos, la idea de este método es recopilar las estimaciones de diferentes algoritmos de clasificación y posteriormente desarrollar el modelo final a partir de este nuevo conjunto de datos. En la terminología de Wolpert, los datos originales y los modelos construidos para ellos en el primer paso se denominan modelos y colección de datos de nivel 0, mientras que el conjunto de datos que contiene estimaciones de los modelos del nivel 0 junto con la variable a predecir original, se denomina datos de nivel 1, de igual manera que el algoritmo de aprendizaje creado para esta nueva base de datos toma el nombre de modelo de nivel 1.

***Palabras clave*:** Modelos apilados, redes neuronales, árboles de decisión, sesgo varianza..

Abstract

Neural networks and decision trees have shown to be efficient methods in applications related to classification, currently there are still attempts to combine these methods with the purpose of obtaining a more efficient algorithm able to improve the accuracy of its estimates, for this reason this thesis has a purpose of creating and comparing the performance of the neural networks models and decision trees, starting from the combination of different decision algorithms through the ensemble method, which was developed by Wolpert in 1992 to increase the predictive accuracy of the models, the idea of this method is to collate the estimation of different classification algorithms and afterwards develop a final model from this new dataset. In Wolpert's terminology, the original data and the models constructed for them in the first step are called models and data collection of level 0, while the data set contains estimates of the level 0 models along with the variable to predict original is denominated as level 1 data, in the same manner the learning algorithm created for this new database will take the name of level 1 model.

***Keywords*:** Ensemble methods, neural networks, decision trees, bias variance..

[a]Estudiante de estadística Universidad Santo Tomás Bogotá
[b]Docente de estadística Universidad Santo Tomás Bogotá

1. Introducción

Debido a que en la actualidad la información crece exponencialmente y la era del Big Data ha llegado con gran fuerza, es necesario implementar nuevas técnicas que sean capaces de analizar grandes cantidades de datos los cuales no pueden ser tratados de manera convencional, puesto que tomaría demasiado tiempo y su costo computacional sería demasiado grande, el objetivo del Big Data al igual que los sistemas analíticos convencionales el tratar los datos y convertirlos en información que facilite la toma de decisiones, inclusive en tiempo real. Esto adquiere gran importancia ya que le permite a las empresas a satisfacer las diferentes necesidades de sus clientes en el menor tiempo posible.

Las técnicas de Machine Learning o aprendizaje automático pertenecen a la disciplina de la ciencia de la computación y a la inteligencia artificial, las cuales mediante la construcción de algoritmos crean sistemas capaces de aprender automáticamente a través del reconocimiento de patrones, consiguiendo que las computadoras actúen sin haber sido programadas explícitamente. Es decir que la máquina que realmente aprende es el algoritmo el cual es capaz de predecir el comportamiento futuro de los datos debido a que el algoritmo aprende de los cálculos anteriores y de esta manera puede adaptar de manera independiente los nuevos datos.

2. Antecedentes

A pesar de que muchos algoritmos de machine learning han existido desde hace mucho tiempo, solo se han empezado a implementar desde los años 90, esto debido a que los avances tecnológicos permiten el procesamiento de grandes volúmenes de datos, en la actualidad estos métodos presentan una amplia gama de aplicaciones, entre los cuales se encuentran:

- Detección de fraudes en transacciones.
- Marketing (Seleccionar clientes potenciales basándose en comportamientos en las redes sociales, interacciones en la web, etc).
- Recomendación de productos en línea, tales como los de Amazon y Netflix.
- Auto conducción de coches.
- Decidir cuál es la mejor hora para llamar a un cliente.
- Hacer pre diagnósticos médicos basados en síntomas del paciente.

En muchas ocasiones las técnicas utilizadas en Machine Learning se complementan con los conceptos estadísticos, ya que ambas disciplinas se basan en el análisis de datos, un caso específico es la creación de ensemble models, en el cual se optimizan diferentes técnicas estadísticas los cuales son capaces de aprender de manera automática. Actualmente existen tres poderosos tipos de ensemble models (bagging, boosting y stacking) cuyas características son:

Bagging: Este método se refiere a la construcción de múltiples modelos (normalmente del mismo tipo) de diferentes sub muestras.

Boosting: Este conjunto de modelos se refiere a la construcción de modelos múltiples (típicamente del mismo tipo), cada uno de los cuales aprende a corregir los errores de predicción de un modelo anterior en la cadena, un ejemplo de este tipo de modelos es el RandomForest.

Stacking: Por último y siendo esta técnica la que se evaluara en este trabajo se encuentran los modelos apilados, los cuales consisten en la construcción de múltiples modelos (típicamente de diferentes tipos) y el modelo supervisor que aprende cómo combinar mejor las predicciones de los modelos primarios.

Los métodos ensemble, como el apilamiento o stacking (Wolpert (1992)) se diseñaron para aumentar la precisión predictiva al mezclar las predicciones de varios modelos de aprendizaje automático, para contextualizar un poco más acerca de estos métodos, debemos tener en cuenta que la combinación de clasificadores por stacking se consideran meta-aprendizaje; es decir que aprende sobre el aprendizaje, estos métodos también han sido utilizados para los modelos de regresión (Breiman (1996)) e incluso el aprendizaje no supervisado (Smyth & Wolpert (1997)). En la terminología de Wolpert, los datos originales y los modelos construidos para estos en el primer paso se denominan como datos y modelos de nivel 0, mientras que el conjunto de datos que contienen las estimaciones de los modelos primarios junto con su variable de clasificación verdadera y el algoritmo de aprendizaje de segunda etapa se denominan como datos y modelo de nivel 1, como se muestra en la Figura 1.

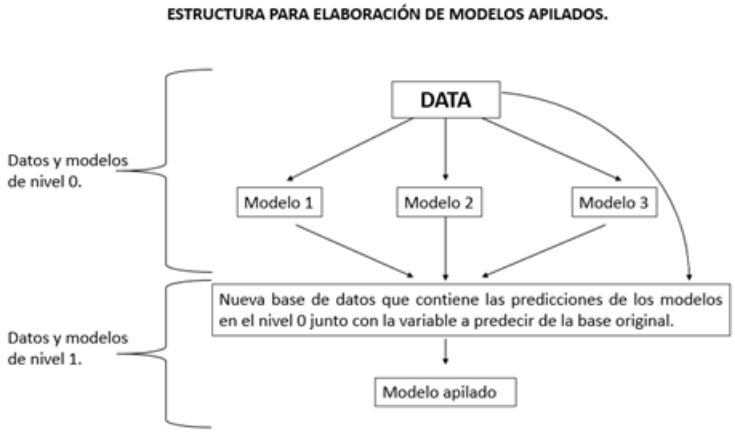

Figura 1: Estructura para elaboración de modelos apilados.

Una condición necesaria para crear un buen ensemble de clasificación es garantizar que los errores de las predicciones de los modelos del nivel 0 no están correlacionadas (Hansen & Salamon (1990)).

En el año 1999 Kai Ming Ting y Ian H. Witten, proponen una variante para el método apilado, el cual consiste en crear un meta-nivel para cada una de las predicciones del nivel 0, esto consiste en utilizar las distribuciones de probabilidad de los modelos del nivel 0 en lugar de una predicción de clase simple como meta-nivel (Ming & Witten (1999)).

Otra variante de apilado propuesta por Alexander K. Seewald y Johannes Furnkranz en 2001, es crear un clasificador de meta- nivel para cada clasificador en el nivel 0. La idea principal para este algoritmo se encuentra en que el modelo de nivel 1 pronostique si las predicciones de los clasificadores de nivel 0 son correctas (Seewald & Furnkranz (2001)).

Desde la aparición de estos métodos los resultados han sido favorables, un ejemplo reciente y el más destacado debido a la implementación de la mezcla de modelos fue la competencia de Netflix, el cual era un concurso abierto y consistía en mejorar el modelo de esta compañía el cual predecía la valoración de los usuarios para las películas. El objetivo principal era disminuir el RSE del modelo de Netflix en un 10 % y de esta manera llevarse el premio que otorgaba esta competencia, el cual consistía en un millón de dólares. Este premio fue ganado por el equipo caos pragmático de BellKor mediante la implementación de un modelo apilado de varias etapas el cual logro reducir el RSE del modelo de Netflix en 10.09 %.

3. Justificación

Las redes neuronales y los árboles de clasificación han demostrado ser métodos eficientes en aplicaciones relacionadas con la clasificación. En la actualidad estos métodos son algunos de los más usados en diferentes campos de acción como mercadeo, finanzas, medicina, entre otras disciplinas, es por esto que a menudo se realicen diferentes intentos por combinar estos métodos con el propósito de obtener un algoritmo más eficiente, gracias a las diferentes alternativas que se cuentan en el campo de la inteligencia artificial es posible implementar diferentes metodologías para optimizar el rendimiento de estos modelos. Este trabajo se enfoca en establecer la mejora que se obtiene en la exactitud de predicción cuando los algoritmos de redes neuronales y los árboles de clasificación sean combinados con diferentes tipos de modelos de clasificación a través del método de ensamble Stacked y Vote.

4. Marco Teórico

En este capítulo se explicaran cada uno de los modelos de clasificación utilizados en este trabajo, estos fueron implementados debido a su popularidad y buenos rendimientos que han presentado desde su desarrollo.

Los modelos implementados aquí para la construcción de los modelos apilados son; modelos de regresión logística, árboles de clasificación, método de vecinos más cercanos KNN, Naive Bayes y los modelos de redes neuronales.

4.1. Regresión Logística

La regresión logística (RL) forma parte de los métodos estadísticos para clasificar o predecir el resultado de una variable categórica, la cual puede ser dicotómica o politómica en función de las variables predictoras.

En la actualidad esta técnica es una de las más empleadas en la medicina clínica, epidemiologia, sector bancario entre otros. Su origen se remonta a la década de los sesenta (Confield et al. (1961)), su uso se expande a principios de los años 80, debido a los avances informáticos con los cuales se cuenta desde entonces.

4.1.1. Concepto

Mediante la RL se busca calcular la probabilidad de que ocurra un evento determinado en función de n variables que se consideran influyentes. Cocretamente, el objetivo fundamental de este método es encontrar los coeficientes $\beta_0, \beta_1 \dots \beta_n$ que mejor se ajustan a la siguiente expresión:

$$P(Y = 1) = \frac{1}{1 + exp(-\beta_0 - \beta_1 X_1 - \ldots - \beta_r X_r} \tag{1}$$

La cual también se puede expresar de la siguiente manera:

$$1 - p_i = \frac{e^{-(\beta_0 + \beta_1' x_i)}}{1 + e^{-(\beta_0 + \beta_1' x_i)}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1' x_i)}} \tag{2}$$

Por lo tanto al hacer la siguiente transformación tenemos un modelo lineal denominado logit.

$$g_i = log\frac{p_i}{1-p_i} = log\left(\frac{\frac{1}{1+e^{-(\beta_0+\beta_1' x_i)}}}{\frac{e^{-(\beta_0+\beta_1' x_i)}}{1+e^{-(\beta_0+\beta_1' x_i)}}}\right) = log\left(\frac{1}{e^{-(\beta_0+\beta_1' x_i)}}\right) = \beta_0 + \beta_1' x_i \tag{3}$$

4.1.2. Coeficiente de verosimilitud

Para que un modelo sea considerado adecuado, este debe atribuir una alta probabilidad de que produzca el evento de interés a aquellos individuos para los cuales, efectivamente se tiene $Y = 1$ o viceversa. Por lo tanto, una medida estadística que nos ayuda a valorar el grado de confiabilidad del modelo, es decir si los resultados obtenidos son coherentes con los datos usados para la construcción del modelo. Esta se haya multiplicando todas las probabilidades p_i (predichas por el modelo) de los n individuos de la muestra tengan la condición que realmente tienen.

Siendo p_i la probabilidad estimada por el modelo de que el i-ésimo individuo presenta cierta condición, y se tiene que d individuos tienen la condición, entonces lo anterior se puede representar de la siguiente manera:

$$V = [p_1 p_2 \dots p_d][(1-p_{d+1})(1-p_{d+2})\dots(1-p_n)] \tag{4}$$

Donde los primeros d factores corresponde a los individuos que contienen la condición y los demás a los que no la presentan.

El valor obtenido es un número entre cero y uno el cual es conocido como la verosimilitud del modelo. Por lo tanto, un modelo que clasifique perfectamente a los individuos corresponderá una verosimilitud de 1; en caso contrario un modelo deficiente tendría una verosimilitud cercana a 0. En consecuencia este estadístico nos indica que tan eficiente ha sido el ajuste del método para modelar la realidad.

4.1.3. Interpretación del modelo Logístico

Cuando se realiza una RL lo que se pretende es estimar los parámetros del modelo $(\beta_0, \beta_1, \beta_2 \dots \beta_r)Z$ así:

$$Z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_r X_r \tag{5}$$

Donde Z es el logaritmo neperiano (L_n) de la odds ratios o ratios de probabilidades de adquirir el producto promocionado por el banco, estos valores indican cuanto se modifican las probabilidades por unidad de cambio en las variables x. En efecto de la anterior expresión se deduce que:

$$O_i = \frac{p_i}{1-p_i} \tag{6}$$

4.2. Algoritmo de vecinos más cercanos (k-nn)

El método de vecinos más cercanos o k-nn fue desarrollado por Fix y Hodges en 1951 en una escuela de la fuerza aérea de los Estados Unidos, este es un método de clasificación supervisada, la idea de este método es clasificar un individuo x a la clase a la cual pertenezca la mayor cantidad de vecinos más cercanos (Fix & Hodges (1951)).

Al aplicar la regla de asociación, se explora en el conjunto de entrenamiento para determinar cuál será la clase a la cual pertenece un individuo nuevo, al aplicar el algoritmo k-nn utiliza la información suministrada por los k prototipos del conjunto de entrenamiento más cercanos de una muestra para su clasificación.

4.2.1. Definición

Dada una base de datos $D = t_1, t_2, t_3, \ldots, t_n$ de individuos y un conjunto de clases $C = C_1, C_2, C_3, \ldots, C_n$, se estima la función de densidad $f : D \to C$, tal que cada t_i es asignado a una clase C_j. Actualmente existen diferentes formas de calcular la distancia entre los individuos de estudio. Tradicionalmente este método utiliza la distancia euclidiana, la cual mide la distancia que existe entre dos puntos y se puede expresar de la siguiente manera:

$$d(A, B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \tag{7}$$

Donde A y B corresponden a las coordenadas de los individuos comparables.

Este método supone que los vecinos más cercanos nos dan la mejor clasificación y esto se hace utilizando todos los atributos; el problema de dicha suposición es que es posible que se tengan muchos atributos irrelevantes que dominen sobre la clasificación, es decir que dos atributos relevantes perderían peso entre otros veinte irrelevantes. A continuación se observa un ejemplo del algoritmo K-NN.

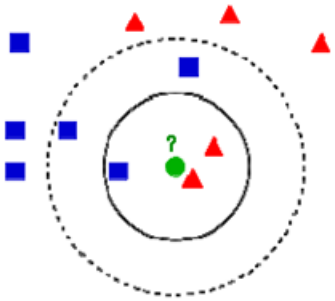

Figura 2: Algoritmo de clasificación K-NN.

En el ejemplo se desea clasificar el circulo verde para $k = 3$, este es clasificado con la clase triangulo, ya que sus 3 vecinos más cercanos están compuestos por 2 triángulos y 1 cuadrado. Caso contrario sucedería si $k = 5$, ya que este sería clasificado en la clase cuadrado, debido a que hay 3 cuadrados y 2 triángulos en el círculo que los contiene.

4.2.2. Eligiendo un k apropiado

La mejor elección del k depende fundamentalmente de los datos, la decisión de cuantos vecinos usar para el k-NN determina la eficiencia del modelo, es decir que tan bien clasificara los datos futuros. Generalmente valores grandes de k reducen el efecto del ruido en la clasificación, pero puede generar sesgo de aprendizaje debido ya que se corre riesgo de ignorar patrones pequeños, los cuales podrían ser importantes para realizar la discriminación de los diferentes grupos. No existe una fórmula para determinar el tamaño de k, por lo tanto este se debe realizar iterativamente y elegir aquel que presente mejor rendimiento de clasificación en los datos de prueba.

4.3. Naive Bayes

El algoritmo Naive Bayes pertenece a la familia de clasificadores supervisados, basado en la aplicación del teorema de bayes bajo el supuesto "naive" de independencia entre cada par de características. Estos

algoritmos de aprendizaje pueden ser extremadamente rápido comparado con otros métodos más sofisticados, a pesar de su simplicidad el clasificador Naive Bayes ha funcionado bastante bien en diferentes situaciones del mundo real, por lo cual es ampliamente utilizado y en muchas ocasiones supera a los métodos de clasificación mas sofisticados.

El modelo Naive Bayes se obtiene entonces de la siguiente manera. Dadas las variables aleatorias Y y $X_1 \dots X_n$, donde Y corresponde a la variable de clasificación, en este trabajo se tienen dos clases + (clase positiva), es decir el cliente adquiere el producto promocionado por el banco o - (la clase negativa) no lo adquiere. La tarea de este algoritmo es modelar la probabilidad conjunta de:

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n|y)}{P(x_1, \dots, x_n)} \tag{8}$$

Utilizando el supuesto naive de independencia tenemos que:

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y), \tag{9}$$

Para toda i, por lo que la anterior relación se simplifica así:

$$P(y|x_1, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \dots, x_n)} \tag{10}$$

Dado que $P(x_1, \dots, x_n)$ es constante entonces se puede expresar de la siguiente manera

$$\hat{y} = P(y)\prod_{i=1}^{n} P(x_i|y) \tag{11}$$

4.3.1. Algoritmo

Un modelo NB consiste en un entero j el cual especifica el número de etiquetas posibles y un entero m que especifica el número de atributos bajo los siguientes parámetros:

$$q(y) \tag{12}$$

Para cualquier $y \, \epsilon \{1 \dots j\}$. El parámetro $q(y)$ puede interpretarse como la probabilidad a priori de la etiqueta y, que cumpla con las siguientes propiedades:

$$q(y) \geq 0 \tag{13}$$

$$\sum_{y=1}^{j} q(y) = 1 \tag{14}$$

$$q_k(x/y) \tag{15}$$

Para cualquier $k \, \epsilon \{1 \dots m\}$, $x \, \epsilon \{-1, +1\}$, $y \, \epsilon \{1 \dots j\}$. El valor estimado para $q_k(x/y)$ puede interpretarse como la probabilidad condicional de que el atributo k tome el valor x dada la condición y, bajo las siguientes propiedades.

$$q_k(x/y) \geq 0 \tag{16}$$

$$\forall y \kappa \sum_{x \epsilon \{-1,+1\}} q(x/y) = 1 \tag{17}$$

Por lo tanto definimos la probabilidad para cualquier y, x_i como:

$$p(y, x_1 \dots x_m) = q(y) \prod_{k=1}^{m} q_k \left(\frac{x_k}{y} \right) \tag{18}$$

Una vez estimados los parámetros mediante un entrenamiento de aprendizaje se tiene que el clasificador N_b es:

$$p(y, x_1 \dots x_m) = q(y) \prod_{k=1}^{m} q_k \left(\frac{x_k}{y} \right) \tag{19}$$

4.3.2. Estimación de máxima-verosimilitud para Naive Bayes

Dada un conjunto de entrenamiento (x_i, y_i) $\forall$ $i = 1 \dots n$, la función de máxima-verosimilitud es:

$$\begin{aligned} L(\theta) &= \sum_{i=1}^{n} log\, p(x^{(i)}, y^{(i)}) \\ &= \sum_{i=1}^{n} log \left(q(y^{(i)}) \prod_{j=1}^{d} q_j(x_j^{(i)} | y^{(i)}) \right) \\ &= \sum_{i=1}^{n} log\, q(y^{(i)}) + \sum_{i=1}^{n} log \left(\prod_{j=1}^{d} q_j(x_j^{(i)} | y^{(i)}) \right) \\ &= \sum_{i=1}^{n} log\, q(y^{(i)}) + \sum_{i=1}^{n} \sum_{j=1}^{d} \log\, q_j(x_j^{(i)} | y^{(i)}) \end{aligned} \tag{20}$$

donde

$$q(y) = \frac{\sum_{i=1}^{n} [[y^{(i)} = y]]}{n} \tag{21}$$

y

$$q_j(x|y) = \frac{\sum_{i=1}^{n} [[y^{(i)} = y \qquad x_j^{(i)} = x]]}{\sum_{i=1}^{n} [[y^{(i)} = y]]} \tag{22}$$

4.4. Redes neuronales

Las Redes Neuronales (NN: Neural Networks) fueron originalmente una simulación abstracta del funcionamiento del sistema nervioso del cerebro humano, constituidos por un conjunto de unidades llamadas

neuronas o nodos conectados unos con otros, su funcionamiento se basa en tratar de resolver problemas no algorítmicos a partir de las experiencias almacenadas como conocimiento.

El primer modelo de red neuronal fue propuesto en 1943 por McCulloch y Pitts en términos de un modelo computacional de actividad nerviosa. Este modelo era un modelo binario, donde cada neurona tenía un escalón o umbral prefijado, y sirvió de base para los modelos posteriores (McCulloch & Pitts (1943)).

Las redes neuronales son más que otra forma de emular ciertas características propias de los humanos, como la capacidad de memorizar y de asociar hechos. Si se examinan con atención aquellos problemas que no pueden expresarse a través de un algoritmo, se observará que todos ellos tienen una característica en común: la experiencia. De acuerdo a las múltiples investigaciones sobre el comportamiento del cerebro se sabe que todas las actividades físicas y psíquicas del ser humano están relacionadas con este, y cuyo funcionamiento se debe a la interconexión de billones de neuronas. También se sabe que el ser humano es capaz de aprender. Siendo este aprendizaje la clave para resolver problemas que inicialmente no se lograban solucionar después de obtener información acerca del problema. Es por esto que las redes neuronales y en general los algoritmos de machine learning poseen la capacidad de aprender y mejorar su comportamiento.

Para entender el funcionamiento de las redes neuronales es necesario tener en cuenta los cuatro aspectos que caracterizan estos métodos: su topología, mecanismo de aprendizaje, tipo de asociación entre la información de entrada y salida, y la forma de representar esta información.

4.4.1. Topología

Consiste en la organización y disposición de las neuronas en la red formando capas o agrupaciones de neuronas. Los parámetros fundamentales de la red son: número de capas, número de neuronas por capa, grado de conectividad y tipo de conexión entre neuronas.

Al realizar una clasificación topológica de las redes neuronales se tiene que pueden ser monocapa o multicapa.

- Red monocapa: Estas se utilizan en tareas relacionadas con lo que se conoce como auto asociación.
- Red multicapa: Su construcción se realiza a partir del agrupamiento de las neuronas en varios niveles o capas, las conexiones entre neuronas pueden ser del tipo feedforward (conexión hacia adelante) o del tipo feedback (conexión hacia atrás).

4.4.2. Mecanismo de aprendizaje

Es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante la etapa de aprendizaje se reducen a la destrucción (el peso de la conexión toma el valor 0), modificación y creación (el peso de la conexión toma un valor distinto de 0) de conexiones entre las neuronas.

Se considera que el proceso de aprendizaje ha terminado cuando los valores de los pesos permanecen estables, es decir que $dw_j/d_t = 0$.

Existen dos tipos de reglas para determinar el mtodo de aprendizaje a utilizar; es decir cmo se van a modificar sus pesos, aprendizaje supervisado o aprendizaje no supervisado.

Aprendizaje supervisado

Se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo (supervisor) que determina la respuesta que debería generar la red a partir de una entrada determinada.

$$(w_{ji}) = \beta y_i (d_j - y_j) \tag{23}$$

El supervisor comprueba la salida generada por el sistema y en el caso de que no coincida con la esperada, se procederá a modificar los pesos de las conexiones. En este tipo de aprendizaje se suelen distinguir a su vez tres formas de llevarlo a cabo:

- Aprendizaje por corrección de error:

  Consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos en la salida, su correción se da de la siguiente manera:

  $$V_p \tag{24}$$

  Siendo

  $V_p(w_{ji})$: Variación en el peso de la conexión entre las neuronas i y j.

  y_i: Salida de la neurona i.

  d_j: Valor de salida deseado para la neurona j.

  y_j: Valor de salida obtenido en la neurona j.

  β: Factor de aprendizaje $(0 < \beta \leq 1)$ que regula la velocidad de aprendizaje.

- Aprendizaje por refuerzo:

  Este tipo de aprendizaje se basa en la idea de no tener un ejemplo completo del comportamiento deseado; es decir, de no indicar durante el entrenamiento cual es la salida exacta que se desea a través de la red ante una determinada entrada. Aquí el supervisor indica mediante una señal de refuerzo si la señal obtenida se ajusta a la deseada (éxito=1 o fracaso=-1) y en función de esta respuesta se ajustan los pesos basándose en un mecanismo de probabilidad.

- Aprendizaje estocástico:

  Aquí se basa en realizar cambios aleatorios en los valores de los pesos y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

Aprendizaje no supervisado

No requieren de influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta; son capaces de autoorganizarse. Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se pueden establecer entre los datos de la entrada.

En general se consideran dos tipos de algoritmos.

- Aprendizaje hebbiano: Por medio de este método se pretende medir la familiaridad o extraer características de los datos de entrada. Este aprendizaje consiste básicamente en el ajuste de los pesos de las conexiones de acuerdo con la correlación de los valores de activación (salidas) de las dos neuronas conectadas.

  $$V_p(w_{ji}) = y_i y_j \tag{25}$$

  Es decir que si dos neuronas y_i y y_j toman el mismo estado simultáneamente (ambas activas o ambas inactivas), el peso de la conexión entre ambas se incrementa. Si por el contrario, una es activa y la otra pasiva (salida negativa), se produce un debilitamiento de la conexión. Por tanto, la modificación de los pesos se realiza en función de los estados (salidas) de las neuronas, obtenidos tras la presentación de cierto estímulo (información de entrada), sin tener en cuenta si se deseaba obtener o no esos estados de activación.

- Aprendizaje competitivo o comparativo.

  Por medio de ese método se pretende que las neuronas compitan unas con otras con el fin de llevar a cabo una tarea dada, es decir que cuando se presente cierta información a la red, solo una parte se active. Por lo tanto las neuronas compiten por activarse. Es decir se realiza una clusterización o clasificación de los datos de entrada.

4.4.3. Tipo de asociación entre las informaciones de entrada y salida

Las redes neuronales son sistemas que almacenan cierta información aprendida; esta se registra de forma distribuida en los pesos asociados a las conexiones entre neuronas. Hay que establecer cierta relación o asociación entre la información presentada a la red y la salida ofrecida por esta. Es lo que se conoce como memoria asociativa.

Existen dos formas primarias de realizar esta asociación entrada/salida y que generan dos tipos de redes:

- Redes heteroasociativas:

  Las redes heteroasociativas, al asociar informaciones de entrada con diferentes informaciones de salida, precisan al menos de dos capas, una para captar y retener la información de entrada y otra para mantener la salida con la información asociada. Si esto no fuese así, se perdería la información inicial al obtenerse el dato asociado, lo cual no debe ocurrir, ya que en el proceso de obtención de la salida se puede necesitar acceder varias veces a esta información que, por tanto, deberá permanecer en la capa de entrada. En cuanto a su conectividad, pueden ser del tipo con conexión hacia adelante (o feedforward) o con conexión hacia atrás (feddforward/feedback), o bien con conexiones laterales.

- Redes autoasociativas:

  Una red autoasociativa asocia una información de entrada con el ejemplar más parecido de los almacenados conocidos por la red. Estos tipos de redes pueden implementarse con una sola capa de neuronas. Esta capa comenzará reteniendo la información inicial a la entrada, y terminará representando la información autoasociada. Si se quiere mantener la información de entrada y salida, se deberían añadir capas adicionales, sin embargo, la funcionalidad de la red puede conseguirse en una sola capa. En cuanto a su conectividad, existen de conexiones laterales y, en algunos casos, conexiones autorrecurrentes.

4.4.4. Representación de la información de entrada y salida

- Redes continuas:

  En un gran número de redes, tanto los datos de entrada como de salida son de naturaleza analógica (valores reales continuos y normalmente normalizados, por lo que su valor absoluto será menor que la unidad). En este caso las funciones de activación de las neuronas serán también continuas, del tipo lineal o sigmoidal.

- Redes discretas:

  Por el contrario, otras redes sólo admiten valores discretos $[0, 1]$ a la entrada, generando también en la salida respuestas de tipo binario. La función de activación en este caso es del tipo escalón.

- Redes híbridas:

  La información de entrada es continua pero a la salida ofrecen información binaria.

4.4.5. Arquitectura de redes neuronales

Estos modelos son construidos de acuerdo a las necesidades y naturaleza de los datos a tratar, a continuación se observa diferentes tipos de arquitectura de estos métodos los cuales se clasifican así:

- Según
  - Número y tipos de entradas.
  - Elementos ocultos.
  - Elementos de salida.
- Según conectividad entre capas
  - Feedforward (hacia adelante).
  - Redes Recurrentes.
  - Estructuras Enrejadas (Lattice).

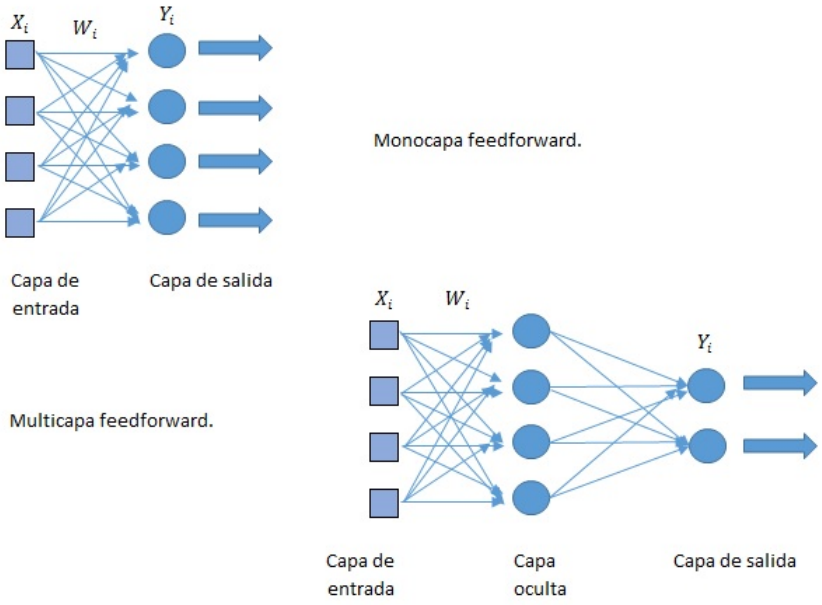

Figura 3: Esquema de redes neuronales.

4.5. Árboles de clasificación

Los métodos basados en árboles de decisión son bastante populares en minería de datos. Estos métodos se derivan de una metodología previa denominada automatic interaction detection.

Esta metodología llega a su mejor rendimiento cuando se tienen grandes masas de datos donde pueden revelar formas complejas en la estructura que no se pueden detectar con los métodos convencionales de regresión.

Algoritmo básico

A continuación a introducir las ideas fundamentales del denominado algoritmo TDIDT (Top Down Induction of Decision Trees) el cual puede ser contemplado como la base de la mayoría de los algoritmos de inducción de árboles de clasificación a partir de un conjunto de datos conteniendo patrones etiquetados.

La idea del algoritmo TDIDT es que mientras que todos los patrones que se correspondan con una determinada rama del árbol de clasificación no pertenezcan a una misma clase, se seleccione la variable que de entre las no seleccionadas en esa rama sea la más informativa o la más idónea con respecto de un criterio previamente establecido. La elección de esta variable sirve para expandir el árbol en tantas ramas como posibles valores toma dicha variable.

- Algoritmo C4.5 Quinlan (1993) propone una mejora del algoritmo ID3, al que denomina C4.5. El algoritmo C4.5 se basa en la utilización del criterio ratio de ganancia (gain ratio), definido como $I(X_i, C)/H(X_i)$. De esta manera se consigue evitar que las variables con mayor número de posibles valores salgan beneficiadas en la selección. Además el algoritmo C4.5 incorpora una poda del árbol de clasificación una vez que este ha sido inducido. La poda está basada en la aplicación de un test de hipótesis que trata de responder a la pregunta de si merece la pena expandir o no una determinada rama.

  Para determinar si es adecuado expandir el árbol es necesario llevar a cabo el siguiente test.

  - $N(t)$ número de ejemplos en el nodo t, en el que se está testando la expansión.
  - $e(t)$ número de ejemplos mal clasificados en el nodo t.
  - $n'(t)$ corrección por continuidad de $e(t)$, la cual se efectua sumando $1/2$ a $e(t)$. Es decir, $n'(t) = e(t) + 1/2$.

  Mientras que los tres términos anteriores $N(t)$, $e(t)$ y $n'(t)$ hacen referencia al nodo t, los siguientes están relacionados con el sub árbol T_t que se va a expandir a partir del nodo t.

  - $h(T_t)$ denota el número de hojas del subárbol T_t.
  - $n'(T_t)$ se obtiene a partir del número de errores existentes en las hojas terminales del subárbol T_t, y se define como:

$$n'(T_t) = \sum_{i=1}^{h(T_t)} e_i + \frac{h(T_t)}{2} \tag{26}$$

  - $S(n'(T_t))$ definido como la desviación de $n'(T_t)$ a partir de la siguiente fórmula:

$$S(n'(T_t)) = \sqrt{\frac{n'(T_t)[N(t) - n'(T_t)]}{N(t)}} \tag{27}$$

  La decisión acerca de expandir el nodo t, y contemplar el subarbol T_t se toma en base a la siguiente regla:

  El nodo t se expande si $n'(T_t) + S(n'(T_t)) < n'(t)$.

- Algoritmo C5.0

  Al igual que sus predecesores, este algoritmo construye los árboles en base a un conjunto de datos de entrenamiento optimizado bajo el criterio de ganancia de información y corresponde a una evolución de su versión anterior, el algoritmo C4.5. Las mayores ventajas de esta versión tienen que ver con la eficiencia en el tiempo de construcción de árbol, el uso de memoria y la obtención de árboles considerablemente más pequeños que en el C4.5 con la misma capacidad predictiva. Adicionalmente, tiene la opción de ponderar algunos atributos de manera de enfocar la construcción

del árbol y se pueden utilizar un aprendizaje penalizado en que es posible asignar un costo a los posibles resultados o matriz de resultados (Cost sensitive algorithm) (Weiss et al. (2008)). Los árboles obtenidos en este trabajo están modelados con el algoritmo C5.0 para poder utilizar gran cantidad de información, manejar mejor las variables continuas y los árboles obtenidos sean más simples y fáciles de entender.

4.6. Modelos apilados

Los métodos apilados consisten en la combinación de múltiples clasificadores generados mediante el uso de diferentes algoritmos de aprendizaje $L_1, \ldots, L_K$ en un solo conjunto de datos S, es decir, pares de vectores de características (x_i) y sus clasificaciones (y_i). En la primera fase, un conjunto de los clasificadores de nivel 0 de acuerdo a la terminología utilizada por (Wolpert (1992)) siendo estos clasificadores $C_1, C_2, \ldots, C_N$, donde $C_i = L_i(S)$, en el segundo paso un algoritmo de aprendizaje aprende a partir de las estimaciones obtenidas de los modelos de la fase anterior.

Este proceso puede ser visto como una especie de validación cruzada, en el cual K algoritmos son aplicados uno a uno a un conjunto de entrenamiento (train) y posteriormente se evalúan sus predicciones en el conjunto de datos seleccionados para este fin (test) $\forall_i = 1, \ldots, n : \forall k = 1, \ldots, N : C_k^i = L_k(S - s_i)$, las predicciones obtenidas para s_i $\hat{y}_i^k = C_k^i(x_i)$, son recopiladas en una nueva base de $((\hat{y}_i^1, \ldots, \hat{y}_i^n), y_i)$ datos de la siguiente forma ,donde las características ahora corresponden a las predicciones de los clasificadores del nivel base y la variable a predecir de la base original, estos datos corresponden ahora al nivel 1. Continuando con este proceso se procede a aplicar un algoritmo de clasificación a este nuevo conjunto de datos para obtener de estos un modelo M "gorrito" para y en función de los pronósticos de los Mk algoritmos anteriores, cuya salida son las predicciones obtenidas mediante el modelo M "gorrito", siendo este el proceso completo del método de generalización apilado propuesto por (Wolpert (1992)), y también utilizado por (Breiman (1996a); (LeBlanc & Tibshirani (1993)), cualquier proceso de generalización de esta forma se conoce como generalización apilada, este proceso puede iterarse dando como resultado niveles $p > 1$, es decir múltiples apilamientos.

La siguiente gráfica ilustra el proceso de validación cruzada de los K modelos primarios en el nivel 0 y la colección de estimaciones para aplicar el modelo M "gorrito" del nivel 1.

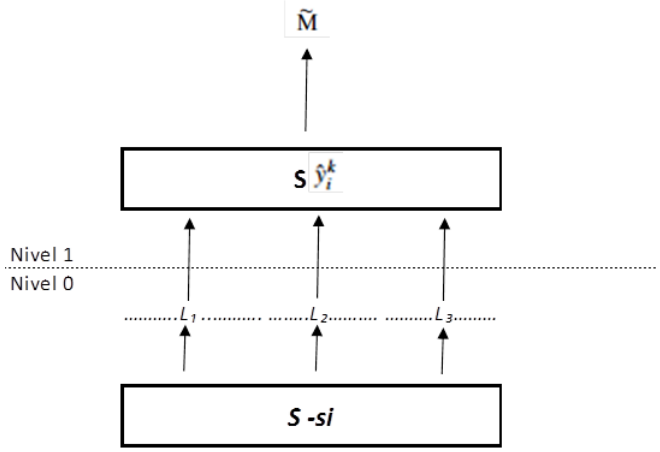

Figura 4: Esquema construcción modelo apilado.

También es importante tener en cuenta que muchos aspectos de la generalización apilada son catalogados como el arte negro, llamado así por Wolpert haciendo referencia a que no existe algún tipo de reglas que indiquen que tipo de clasificador es adecuado para derivar el modelo de nivel superior y el tipo de

atributos que deben utilizarse como entrada a estos algoritmos.

4.6.1. Estimación de la tasa de error

Los errores de los algoritmos de aprendizaje son estimados mediantes j validaciones cruzadas, este proceso se realiza j veces utilizando diferentes muestras aleatorias dando como resultado j diferentes conjuntos de estimaciones. Siendo esta una manera directa de usar la validación cruzada para asignar a un generalizador L_i junto con un conjunto de aprendizaje "theta" la tasa de error de L_i a partir del promedio de los errores estimados mediante validación cruzada.

4.6.2. Apilamiento de árboles de clasificación

Dado un conjunto de datos $Z = \{(y_n, x_n), n = 1, \ldots, N\}$, se seleccionan al azar un conjunto de prueba Z_{TS} resultando en $Z' = Z - Z_{TS}$. Posteriormente se dividen los datos Z' en R partes iguales $Z_1, \ldots, Z_J$ y defina $Z^{(J)} = Z' - Z_j$. Usando los datos en $Z^{(J)}$ se crea un árbol T_k^j de clasificación con K nodos. Es necesario podarlo hacia arriba de modo que T_k^j sea el sub árbol de T_K^j. El cual presenta un error de aprendizaje mínimo entre todos los k-nodos de los sub árboles de $T_K^j \cdot v_k^{(j)}$ (x) denota el predictor basado en T_K^j.

Tomar el $\{\alpha_k\}$ para minimizar:

$$\sum_j \sum_{(y_n, x_n)\epsilon\iota_j} (y_n - \sum_k \alpha_k \upsilon_k^{(j)}(x_n))^2 \tag{28}$$

Bajo las restricciones $\alpha_k >= 0$. Crear un árbol T_K basado en todos los datos de Z', con el error mínimo de los sub árboles T_k y las predicciones correspondientes $\{\alpha_k\}$. Posteriormente se estima el error en:

$$\upsilon(x) = \sum \alpha_k \upsilon_k(x) \tag{29}$$

Para

$$\frac{1}{N_{TS}} \sum_{(y_n, x_n)\epsilon\iota_{TS}} (y_n - \upsilon(x_n))^2 \tag{30}$$

Donde $N_{Ts} = |Z_{Ts}|$. Este procedimiento se repite varias veces con Z_{TS} y el Z_j seleccionado al azar y los errores obtenidos se promedian para dar una estimación del error de apilamiento. Al mismo tiempo, se selecciona la v_k que tiene la tasa de errores más pequeña. Su error está dado por:

$$\frac{1}{N_{TS}} \sum_{(y_n, x_n)\epsilon\iota_{TS}} (y_n - \upsilon(x_n))^2 \tag{31}$$

El cual también es promediado después de varias repeticiones.

4.7. Evaluación del modelo

El objetivo de evaluar un modelo de clasificación es determinar su desempeño en casos similares para extraer conclusiones o hipótesis. Típicamente se simulan condiciones futuras pidiendo al modelo que clasifique un conjunto de datos que se asemejen a las predicciones que se le pedirá que haga en el futuro. Observando las respuestas del modelo, se puede aprender sobre sus fortalezas y debilidades.

Los valores reales y predichos son la clave de la evaluación, se requiere saber la respuesta correcta para las predicciones de una máquina de aprendizaje. El objetivo es obtener dos vectores de datos: uno con los valores de clase correctos o reales y el otro con los valores de clase previstos. Ambos vectores deben tener el mismo número de valores almacenados en el mismo orden.

4.7.1. Matriz de confusión

Una matriz de confusión es una tabla que categoriza las predicciones de acuerdo a la coincidencia con el valor real. Una de las dimensiones de la tabla indica las posibles categorías de valores predichos, mientras que la otra dimensión indica lo mismo para los valores reales.

Cuando el valor predicho es el mismo que el valor real, es una clasificación correcta. Las predicciones correctas caen en la diagonal de la matriz de confusión. Las celdas de matriz fuera de la diagonal indican los casos en que el valor predicho difiere del valor real. Estas son predicciones incorrectas.

Las medidas de rendimiento más comunes consideran la capacidad del modelo para discernir una clase frente a todas las otras. La clase de interés se conoce como la clase positiva, mientras que todos los demás se conocen como negativos.

La relación entre la clase positiva y las predicciones de clase negativa puede ser representada como una matriz de confusión 2×2 que tabula si las predicciones caen en una de las cuatro categorías:

- Verdadero Positivo *(VP)*: Correctamente clasificado como la clase de interés.
- Verdadero Negativo *(VN)*: Correctamente clasificado como la clase negativa.
- Falso Positivo *(FP)*: Incorrectamente clasificado como la clase de interés.
- Falso Negativo *(FN)*: Incorrectamente clasificado como la clase negativa.

La matriz de confusión se puede ilustrar como se muestra en la Figura 2:

Figura 5: Matriz de confusión.

La matriz de confusión, es la base de varias medidas importantes del desempeño del modelo.

4.7.2. Accuracy

Accuracy (a veces llamada tasa de éxito) es la proporción del número total de predicciones correctas. Se determina usando la matriz de confusión de 2×2:

$$accuracy = \frac{VP + VN}{VP + VN + FP + FN} \tag{32}$$

En la Fórmula 28, los términos VP, VN, FP y FN se refieren al número de veces que las predicciones del modelo caen en cada una de las categorías. Por lo tanto, accuracy es una proporción que representa el número de verdaderos positivos y verdaderos negativos, divididos por el número total de predicciones.

4.7.3. Precisión

La precisión se utiliza principalmente en el contexto de recuperación de información, esta estadística pretende proporcionar una indicación de cuán interesantes y relevantes son los resultados de un modelo.

- ***Precisión*** se define como la proporción de ejemplos positivos que son verdaderamente positivos. En otras palabras, cuando un modelo predice la clase positiva, ¿Con qué frecuencia es correcto? Un modelo preciso sólo predecirá la clase positiva en los casos que es muy probable que sea positiva. Será un modelo digno de confianza.

$$precision = \frac{VP}{VP + FP} \tag{33}$$

4.7.4. Curva ROC

La curva ROC se utiliza comúnmente para examinar el equilibrio entre la detección de verdaderos positivos, evitando al mismo tiempo los falsos positivos. La curva ROC fue desarrollada por ingenieros en el campo de las comunicaciones; alrededor de la época de la Segunda Guerra Mundial, los operadores de radar y radio utilizaron curvas ROC para medir la capacidad de un receptor de discriminar entre señales verdaderas y falsas alarmas. La misma técnica es útil hoy para visualizar la eficacia de los modelos de aprendizaje automático.

Las curvas se definen en un gráfico con la proporción de verdaderos positivos en el eje vertical y la proporción de falsos positivos en el eje horizontal. Debido a que estos valores son equivalentes a sensibilidad y (1 - especificidad), respectivamente. Las características de un diagrama ROC se representan en la Figura 3.

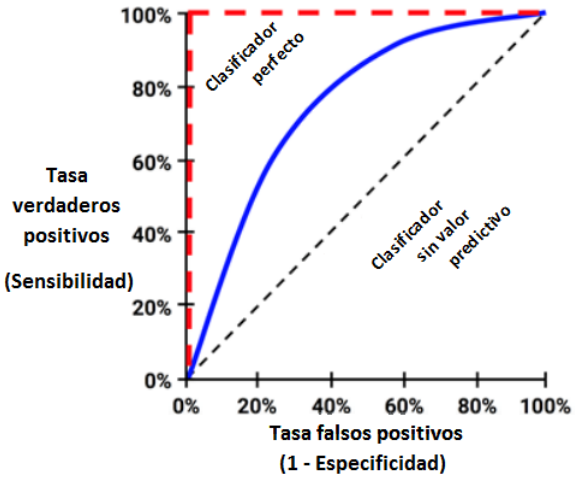

Figura 6: Curva ROC.

Tres clasificadores hipotéticos se contrastan en la gráfica anterior. En primer lugar, la línea diagonal desde la esquina inferior izquierda a la esquina superior derecha del diagrama representa un clasificador sin valor predictivo. Este tipo de clasificador detecta verdaderos positivos y falsos positivos exactamente a la misma velocidad, lo que implica que el clasificador no puede discriminar. Las curvas ROC que caen cerca de esta línea indican modelos que no son muy útiles. El clasificador perfecto tiene una curva que pasa a través del punto 100 por ciento de la tasa de verdaderos positivos y 0 por ciento de tasa de

falsos positivos, por lo tanto es capaz de identificar correctamente todos los positivos antes de clasificar incorrectamente cualquier resultado negativo.

Cuanto más cerca está la curva al clasificador perfecto, mejor se encuentra en la identificación de valores positivos, lo cual se puede medir usando una estadística conocida como el área bajo la curva ROC (abreviado AUC). El AUC trata el diagrama ROC como un cuadrado bidimensional y mide el área total bajo la curva, oscila entre 0.5 (para un clasificador sin valor predictivo) y 1.0 (para un clasificador perfecto). Una convención para interpretar los puntajes del AUC son los siguientes:

- Excepcional = 0.9 a 1.0
- Excelente / bueno = 0.8 a 0.9
- Aceptable / justo = 0.7 a 0.8
- Pobre = 0.6 a 0.7
- Sin discriminación = 0.5 a 0.6

5. Objetivos

5.1. Objetivo General

El objetivo principal de esta tesis es aplicar los modelos apilados mediante los esquemas de redes neuronales y árboles de clasificación a una campaña de mercadeo generada en un banco portugués a través de contacto telefónico, y comparar estos resultados con modelos clásicos de clasificación.

5.2. Objetivos específicos

- Comprobar si los modelos apilados mejoran las predicciones con respecto a diferentes modelos de clasificación sin apilar.
- Identificar el ajuste de cada uno de los modelos apilados, y determinar cual de ellos presenta mayor ajuste a los datos.
- Generar mayor conocimiento sobre las ventajas y desventajas que presentan los modelos apilados.

6. Metodología

Para probar el método de apilamiento para los algoritmos de redes neuronales y árboles de clasificación se utiliza la base de datos **bank marketing** (Moro et al. (2014)) del repositorio de datos UCI `https://archive.ics.uci.edu/ml/datasets.html/`, el cual contiene diferentes conjuntos de datos para el servicio de la comunidad de machine learning.

Esta investigación emplea los clasificadores *naive bayes*, *Knn*, *árboles de clasificación* y *regresión logística* como modelos del nivel 0, estos clasificadores son aplicados de forma independiente sobre el conjunto de datos de entrenamiento y evaluados en la partición de datos creada para valorar el rendimiento del modelo. Posteriormente, se crea una nueva base de datos para el nivel 1 con las estimaciones obtenidas a partir de lo algoritmos implementados en el nivel anterior, como modelos de apilamiento se aplico los algoritmos de ***redes neuronales*** y ***árboles de clasificación C5.0***.

6.1. Bank Marketing Data Set

Los datos de Bank Marketing Data Set corresponden a una campaña de mercadeo generada en un banco portugués a través de contacto telefónico. El objetivo de la campaña era predecir si un cliente estaría dispuesto a adquirir el producto promocionado, el cual es un depósito bancario. Durante la realización de la campaña de mercadeo fue necesario realizar contacto telefónico varias veces con los clientes para determinar si se deseaba adquirir el producto ofrecido. El conjunto de datos contiene 41.188 observaciones con 21 variables.

6.1.1. Selección de variables

Para la selección de variables se realiza un modelo logístico, teniendo en cuenta que la variable respuesta Y es binomial (Yes, No). Para realizar este análisis se comienza generando el modelo logit de la variable respuesta con todas las variables y se determina el grado de significancia mediante el *P-valor* el cual indica que variables no se deben incluir en el modelo, depurando aquellas que no son significativas y de esta manera no tenerlas en cuenta para estimar los modelos apilados. Las variables utilizadas para generar las estimaciones son las siguientes:

- Datos del cliente del banco
  - age: Edad (numérico)
  - job: Tipo de trabajo (categórica) - 'admin.', 'blue collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self employed', 'services', 'student', 'technician', 'unemployed', 'unknown'.
  - marital: Estado civil (categórica) - 'divorced', 'married', 'single', 'unknown'.
  - education: Educación (categórica) - 'basic.4y', 'basic.6y', 'basic. 9y', 'high.school', 'illiterate', 'professional.course', 'university.degree', 'unknown'.
  - default: ¿Tiene crédito en mora? (categórica) - 'no', 'yes', 'unknown'.
  - housing: ¿Tiene préstamo de vivienda? (categórica) - 'no', 'yes', 'unknown'.
  - loan: ¿Tiene préstamo personal? (Categórica) - 'no', 'yes', 'unknown'.
- Relacionado con el último contacto de la campaña en curso
  - contact: Tipo de comunicación de contacto (categórica) - 'cellphone', 'telephone'.
- Otros atributos
  - campaign: Número de contactos realizados durante esta campaña para el cliente (discreta) - incluye último contacto.
  - pdays: Número de días que pasaron por el cliente después de la última vez en contacto de una campaña anterior (discreta) - 999 significa que no se puso en contacto previamente al cliente.
  - previous: Número de contactos realizados antes de esta campaña y para el cliente (discreta).
  - poutcome: Resultados de la anterior campaña de marketing (categórica) - 'failure', 'nonexistent', 'success'.
- Variable de salida
  - Y: ¿Ha suscrito el cliente un depósito a plazo? (Binario) - 'Yes', 'No'.

6.2. Estimación de modelos

Para los datos de esta entidad bancaria se quiere predecir la aceptación de la oferta por parte de los clientes la cual es una variable binaria, con valor 1 si acepta la oferta y valor 0 si no se acepta. Este pronóstico se obtiene a través de una serie de variables como la edad del cliente, educación, créditos, tipo de trabajo, número de contactos realizados antes de la campaña, etc.

La variable respuesta del conjunto de datos está dada por las categorías *"Yes"* y *"No"*, de las cuales solo el 11.26 % corresponde a la categoría "Yes". Debido a este desbalance en las categorías se realizaran dos tipos de modelos apilados: uno modelo con la base de datos original y otro con una nueva base de datos en donde se equipara las dos clases a predecir. Esto con el objetivo de comparar el rendimiento de los modelos apilados mediante estos dos casos. Se genera la estimación primaria de los modelos KNN, naive bayes, regresión logística y árboles de clasificación a las bases de datos original y equiparada respectivamente, estos algoritmos corresponden al nivel 0 del proceso para crear los modelos finales.

Una vez obtenidas las estimaciones de los modelos primarios, se crea una nueva base de datos que contiene la variable respuesta original y esta será acompañada de las estimaciones realizadas por los modelos primarios y luego de esto se procede a entrenar el modelo apilado en esta nueva data.

6.3. Evaluación de los modelos

En esta investigación se emplea los algoritmos naive bayes, Knn, regresión logística, redes neuronales y árboles de clasificación por medio del software estadístico R Proyect `https://www.rstudio.com/`, estos algoritmos son evaluados de forma independiente en el conjunto de datos de entrenamiento "train" y control "test".

Posteriormente se crea un nuevo conjunto de datos a partir de las estimaciones obtenidas mediante los algoritmos anteriores, este proceso se realiza mediante el método de apilamiento "stacking" y evaluado sobre el conjunto de datos de control "test".

Debido a su popularidad y buen desempeño se toma como referencia el algoritmo de redes neuronales para comparar el rendimiento de los algoritmos construidos mediante el método de apilamiento, por lo cual se crean tres diferentes grupos de modelos los cuales están compuestos de la siguiente manera:

- Grupo 1.

  En este grupo se crean los modelos de redes neuronales apilada 1 y árbol de clasificación apilado 1, estos modelos están compuestos por dos niveles, los cuales contienen en el nivel 0 las estimaciones de los algoritmos de:

  - Árbol de clasificación función rpart de la librería tree, (Breiman et al. (1984)).
  - Árbol de clasificación función ctree de la librería party, (Torsten Hothorn et al.).
  - Redes neuronales función nnet de la librería nnet, (Ripley & Venables (1996)).
  - Regresión logística función glm de la librería stats, (Hastie & Pregibon (1992)).
  - Knn de la librería class, (Ripley & Venables (2002)).
  - Árbol de clasificación método C5.0, basado en el algoritmo C5.0 de Quinlan.
  - NB de la librería e1071, (Meyer).

  Es decir el nivel 0 de los modelos de redes neuronales apilada 1 y árbol de clasificación apilado 1 se entrenan a partir de las estimaciones obtenidas de los siete algoritmos anteriores.

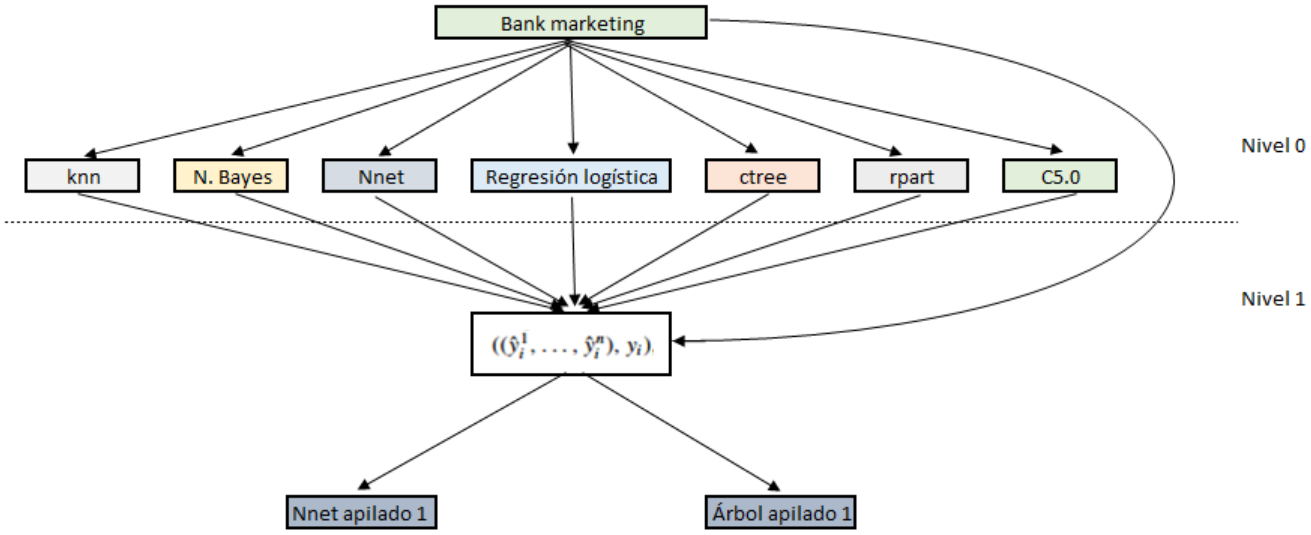

Figura 7: Modelos apilados grupo 1.

- Grupo 2.

  Para este grupo de modelos se modifica la base de entrenamiento para los modelos de redes neuronales 2 y árboles de clasificación 2, ya que, se eliminan las estimaciones del modelo knn, puesto que este algoritmo presento la menor precisión en las estimaciones.

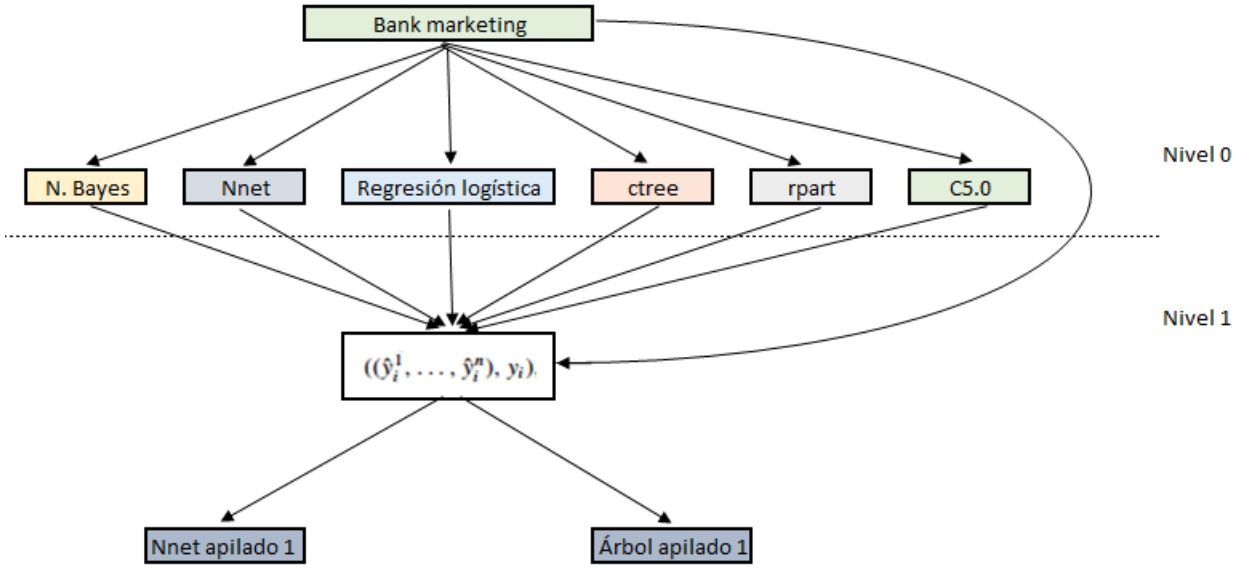

Figura 8: Modelos apilados grupo 2.

- Grupo 3.

  Para la construcción de los modelos de árboles de clasificación apilado 3 y redes neuronales apilada 3 se emplearon 3 capas de aprendizaje, es decir, una más que en los modelos de los grupos anteriores, en este grupo se tomaron las estimaciones de los modelos del grupo 2 para crear una capa intermedia antes de aplicar los modelos del estudio.

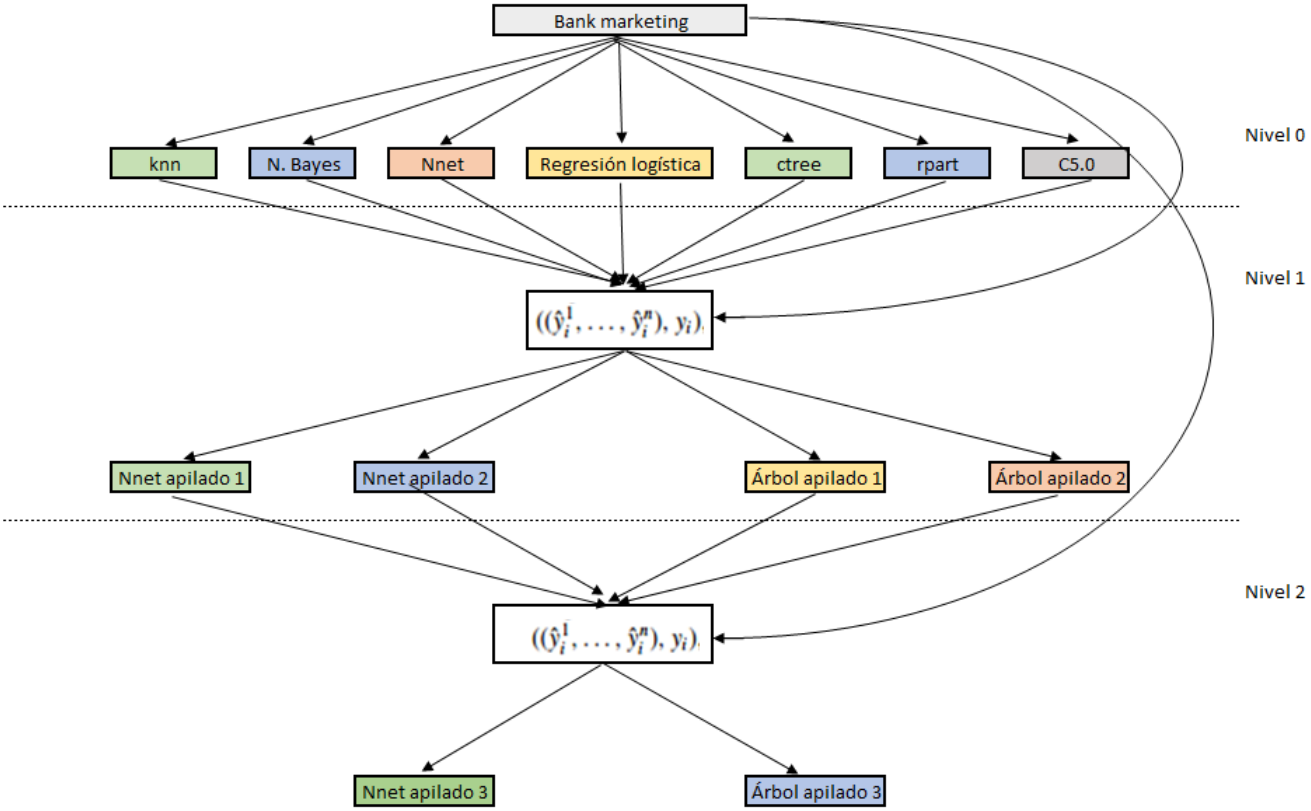

Figura 9: Modelos apilados grupo 3.

7. Resultados

Una vez seleccionadas las variables se aplican diferentes modelos de clasificación que permiten estimar la decisión de las personas en adquirir el producto promocionado. Para la calibración de los modelos y su validación se genera la división de la base de datos en dos partes, teniendo como base *train* o de entrenamiento al 70 % de los datos de la misma y como base *test* o de prueba al otro 30 % de los datos de la base, con esta última se realiza la validación cruzada.

Los resultados obtenidos mediante los diferentes clasificadores aplicados al caso de estudio, se resumen en la siguiente tabla, en la cual se observa que los algoritmos apilados mejoran significativamente sus estimaciones en cuanto a la precisión y acurracy, esto comparado contra los algoritmos sin apilar.

	Accuracy	Precisión	AUC
Árbol rpart	0.8432	0.3709	0.7497
Árbol ctree	0.8604	0.4142	0.7832
Regresión logística	0.7024	0.2232	0.6846
Knn	0.7913	0.2863	0.7471
NB	0.8385	0.3651	0.7743
Árbol C5.0	0.7794	0.2914	0.7862
Red neuronal	0.8393	0.3559	0.7202
Nnet apilado 1	0.8983	0.6326	0.7880
Nnet apilado 2	0.9001	0.7036	0.7967
Nnet apilado 3	0.9036	0.6907	0.7980
Árbol apilado 1	0.9026	0.7114	0.7437
Árbol apilado 2	0.9034	0.6874	0.7437
Árbol apilado 3	0.9064	0.6929	0.7458

En las figuras 7 y 8 se observa que todos los modelos ensemble aumentan considerablemente la proporción del total de predicciones correctas es decir el acurracy y la precisión de las predicciones positivas, en este caso se obtiene que los algoritmos apilados aprenden de manera más eficiente a medida que obtienen información de otros clasificadores. Así mismo, se observa que el rendimiento de un ensamble puede ser mayor al desempeño de los clasificadores individuales que lo forman.

Se tiene que el algoritmo que mayor proporción de aciertos presenta es el modelo apilado de 3 capas mediante la combinación de 15 algoritmos de aprendizaje repartidos en cada uno de sus niveles.

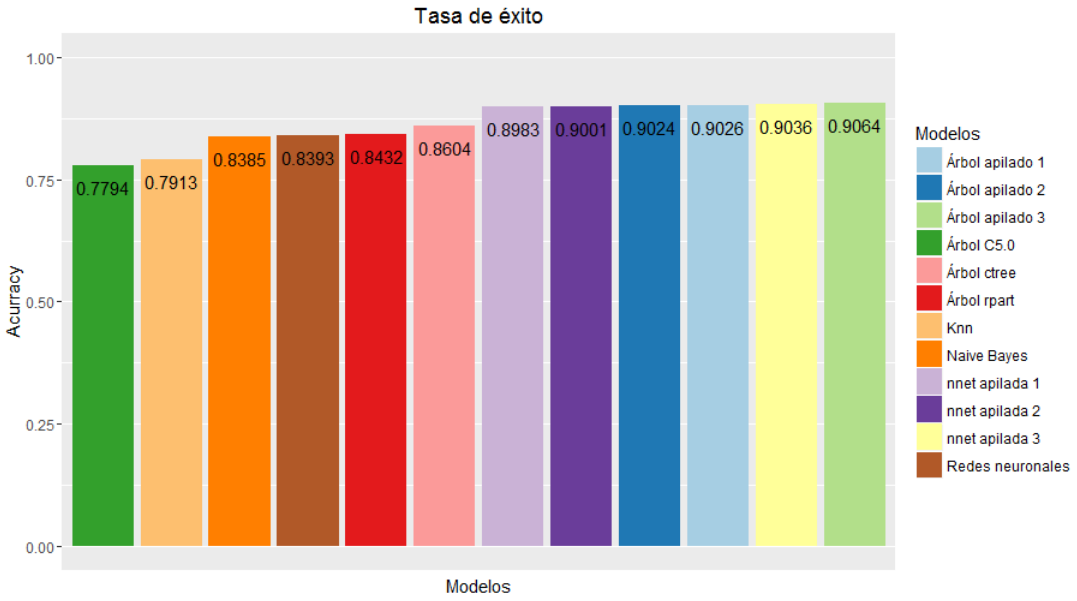

Figura 10: Proporción de aciertos.

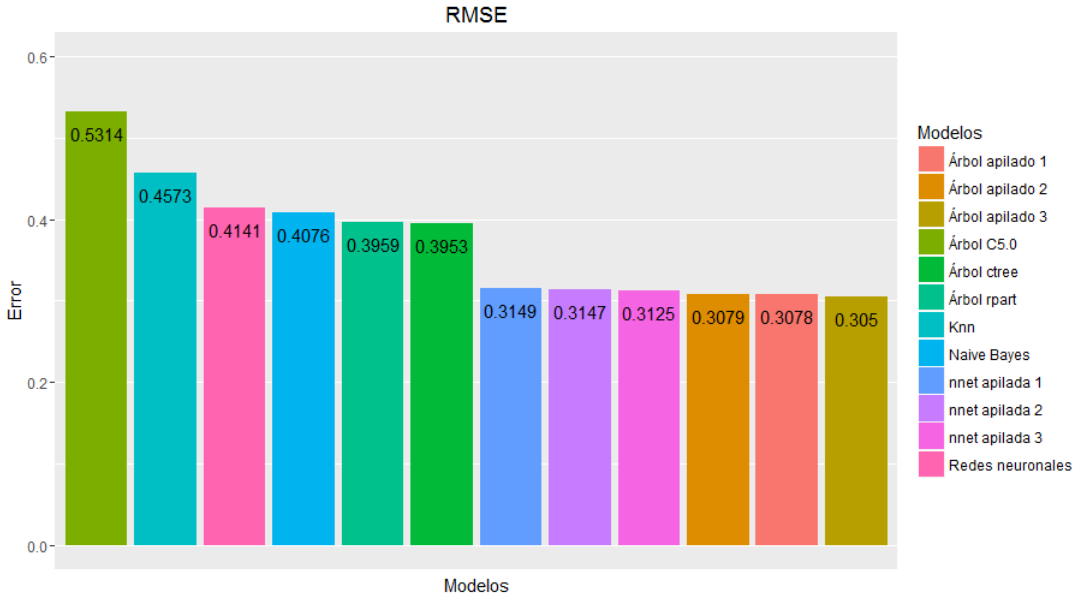

Figura 11: Error cuadrático medio.

De igual manera se tiene que los modelos ensembles aumentan considerablemente la precisión de las estimaciones positivas, como se evidencia en la figura 8, donde el algoritmo del árbol apilado 1 aumenta la precisión en 71 % más que el mejor modelo individual es decir el árbol de clasificación del método ctree. También se observa que los métodos ensemble superar de manera significativa la precisión del modelo bayesiano naive bayes y las redes neuronales, siendo estos modelos algunos de los más populares y que presentan mejor desempeño en la actualidad.

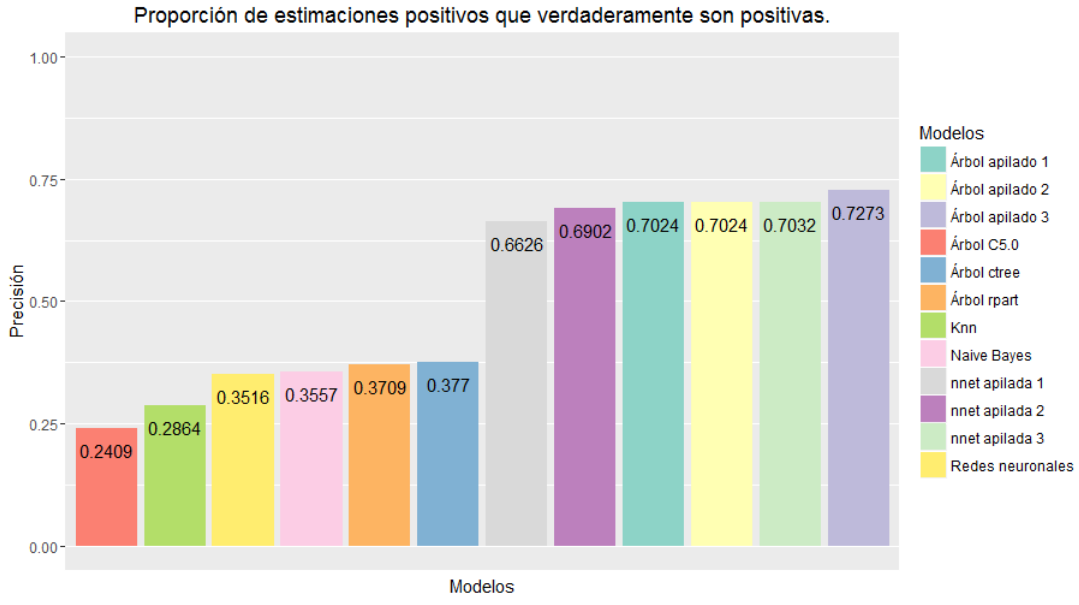

Figura 12: Proporción de predicciones positivas que realmente son positivas.

A continuación se presentan las curvas Roc obtenidas por medio de cada uno de los algoritmos implementados en el presente trabajo, finalizando con un gráfico para comparar los mejores modelos simples

y apilados:

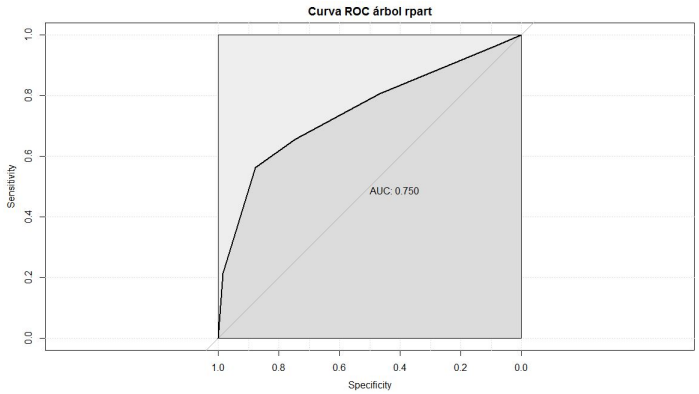

Figura 13: Árbol rpart.

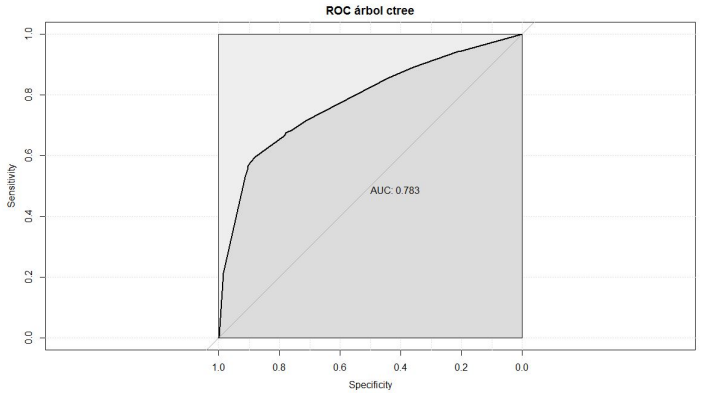

Figura 14: Árbol ctree.

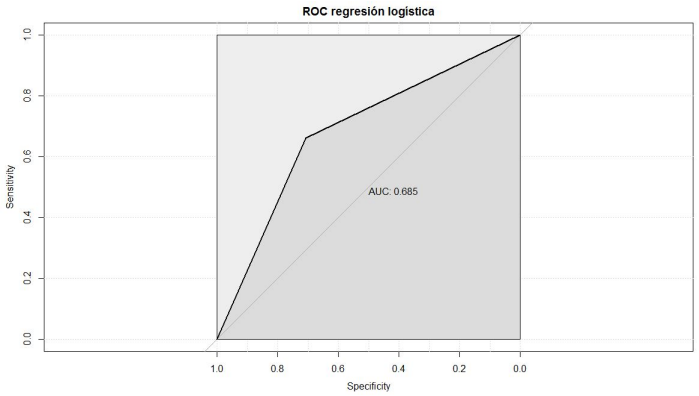

Figura 15: Regresión logística.

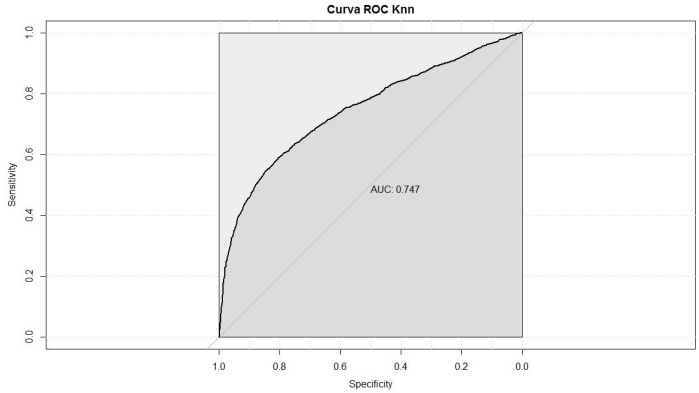

Figura 16: Knn.

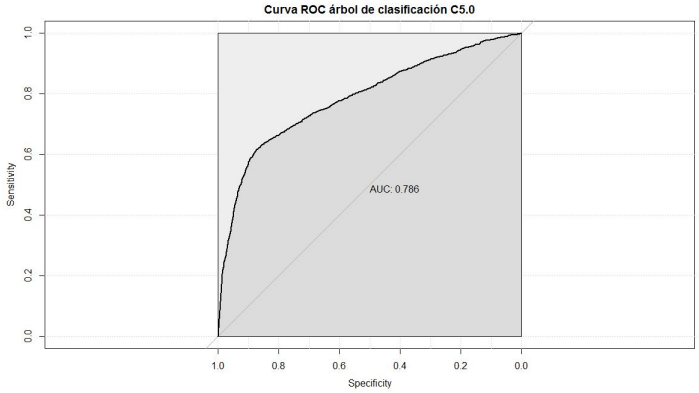

Figura 17: Árbol de clasificación C5.0.

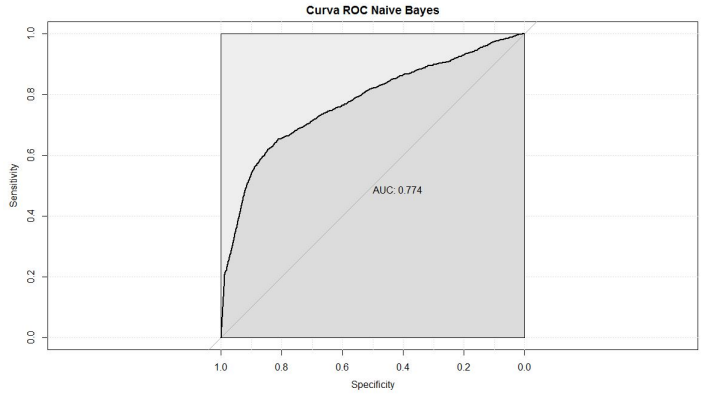

Figura 18: Naive Bayes.

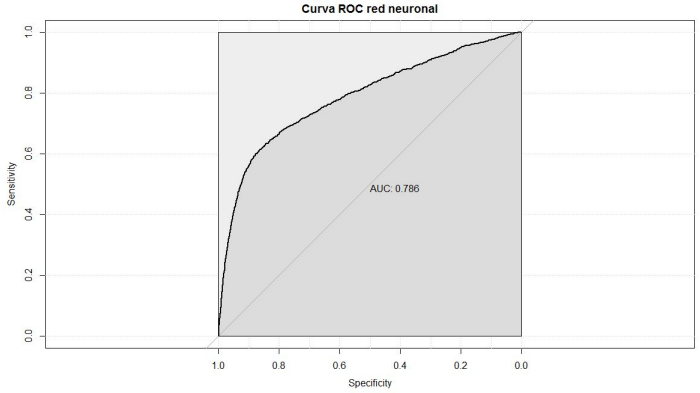

Figura 19: Red neuronal.

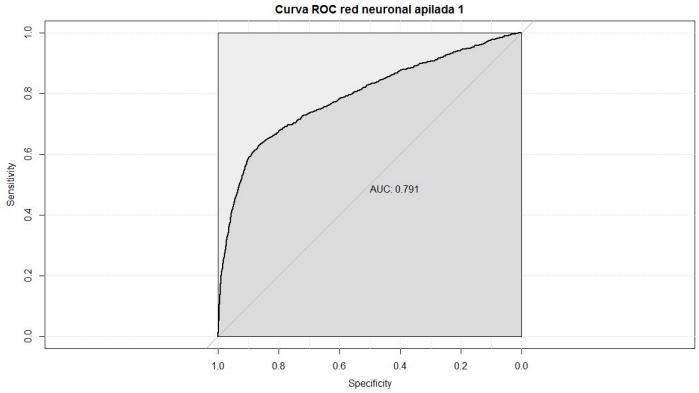

Figura 20: Red neuronal apilada 1.

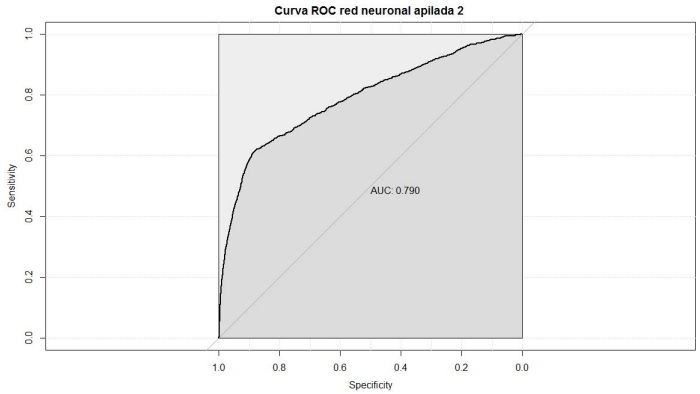

Figura 21: Red neuronal apilada 2.

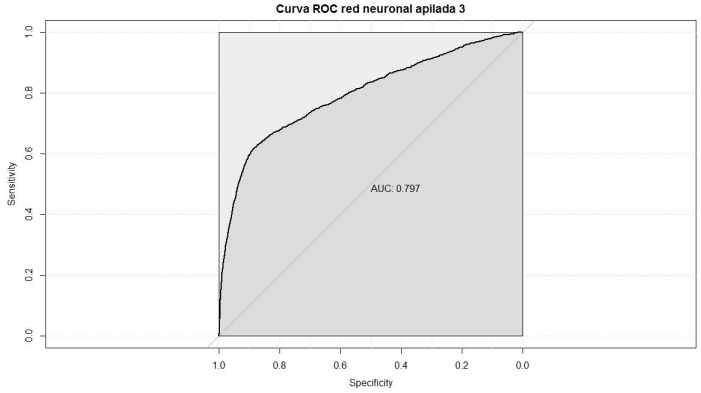

Figura 22: Red neuronal apilada 3.

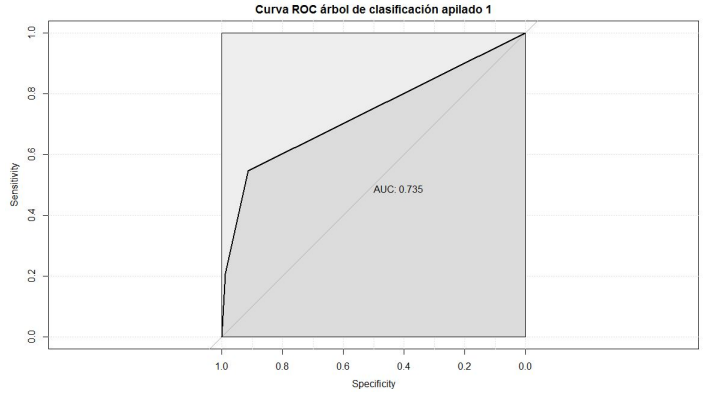

Figura 23: Árbol de clasificación apilado 1.

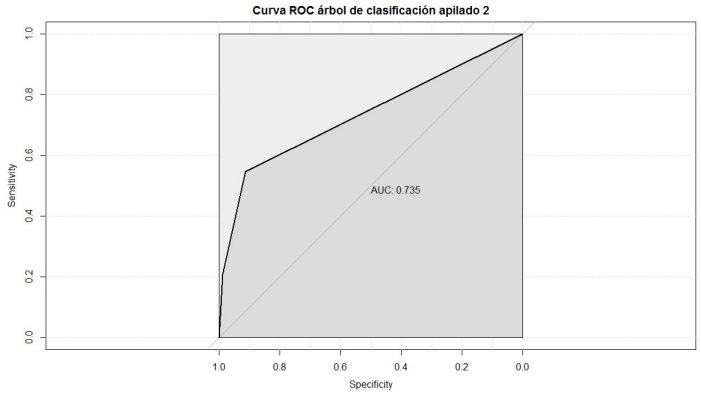

Figura 24: Árbol de clasificación apilado 2.

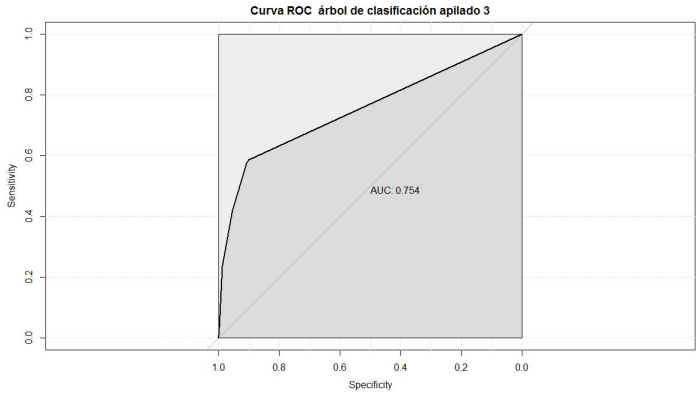

Figura 25: Árbol de clasificación apilado 3.

En la figura 22 se observa que los modelos ensemble presentan buena discriminación en cuanto al porcentaje de verdaderos positivos versus los falsos positivos. Siendo el algoritmo de red neuronal apilado de 3 capas el que tiene mayor área bajo la curva "Auc", aumentando la discriminación obtenida por los modelos individuales.

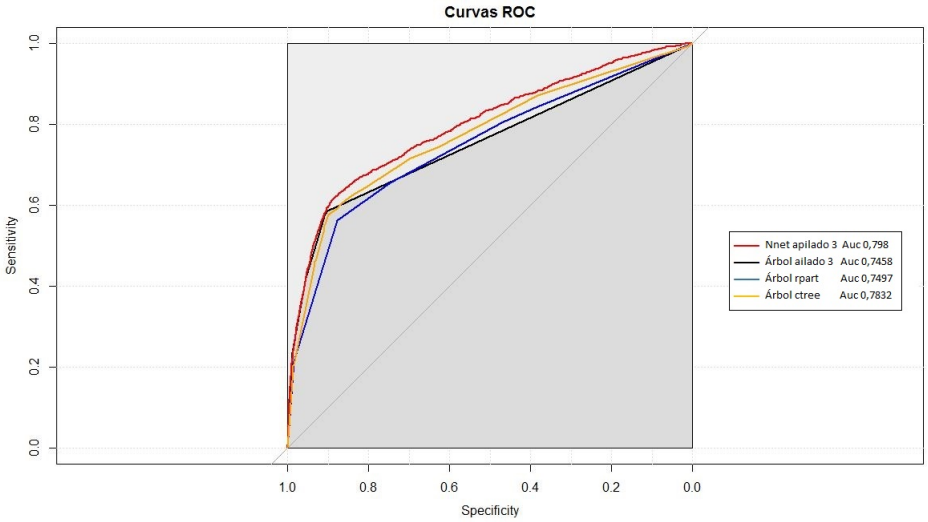

Figura 26: Curva Roc modelos apilados vs modelos individuales.

Se observa que todos los algoritmos ensamblados presentan un comportamiento similar antes el procesamiento del mismo conjunto de datos. En cada uno de los modelos apilados se tiene que la exactitud aumenta o disminuye en los diferentes algoritmos construidos mediante este método de ensemble. También se tiene que los algoritmos ensamblados tienden a mejorar su exactitud de clasificación a medida que se combinan con diferentes algoritmos de clasificación, así mismo, el tiempo de procesamiento también se incrementa.

8. Conclusiones

En el presente trabajo se evaluó empíricamente varios métodos de clasificación mediante la técnica ensemble de machine learning, la cual consiste en la combinación de múltiples algoritmos de clasificación de diferente tipo. En esta investigación nos encontramos con varias novedades en cuanto a la aplicación de los métodos ensemble. La primer situación fue encontrar que no existe una regla para definir la cantidad de algoritmos y mucho menos el número de niveles que debe tener un clasificador ensamble adecuado que garantice la construcción de un modelo óptimo para mejorar el rendimiento de los algoritmos de niveles inferiores, esta situación es denominada como el arte negro. Por otro lado también se encontró que los métodos ensamble presentan una gran flexibilidad y puede ser una buena alternativa cuando se pretende obtener un mejor desempeño en los diferentes modelos estadísticos ejecutados de manera individual. Sin embargo, el costo computacional para los métodos ensemble es significativamente mayor comparado con los algoritmos individuales, por lo cual es necesario contar con los equipos y programas adecuados para tratar grandes cantidades de datos y algoritmos complejos.

Como trabajos futuros, se recomienda combinar los tres métodos ensembles en diferentes capas del algoritmo, es decir, combinar métodos de stacking, bagging y boosting, si bien esto demanda bastante recurso computacional, incrementar el número de iteraciones provee mayor precisión sobre los resultados obtenidos.

Referencias

Breiman, L. (1996), 'Stacked regressions', *Machine Learning* **24**, 49–64.

Cornfield, J., Gordon, T. & Smith, W. (1961), 'Quantal response curves for experimentally uncontroled variables.', *Bulletin of the International Statistical Institute* **38**, 97–115.

Díaz, J. (2013), 'Comparación entre árboles de regresión cart y regresión lineal', *Comunicaciones en Estadística* **6, 21**.

Dzeroski, S. & Zenko, B. (2004), 'Is combining classifiers with stacking better than selecting the best one?', *Machine Learning* **54**, 225–273.

Fix, E. & Hodges, J. (1951), 'An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges', *International Statistical Review / Revue Internationale de Statistique* **57**, 233–238.

Hansen, L. & Salamon, P. (1990), 'Neural network emsembles', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**, 993–1001.

Jones, R. (1975), 'Probability estimation using a multinomial logistic function', *Journal of Statistical and Computer Simulation* **3**, 29–315.

Lackes & Mack, D. (1998), 'Computer based training', *Neuronal Networks* .

Ming, K. & Witten, I. (1999*a*), 'Issues in stacked generalization', *Journal of Artificial Intelligence Research* **10**, 271–289.

Ming, K. & Witten, I. (1999*b*), 'Issues in stacked generalization', *Journal of Artificial Intelligence Research* **10**, 271–289.

Piotte, M. & Chabbert, M. (2009), 'The pragmatic theory solution to the netflix grand prize', *Pragmatic Theory Inc* .

Satorra, A. & Bentler, P. (2001), 'A scaled difference chi-square test statistic for moment structure analysis', *Psychometrika* **66**, 507–514.

Weiss, G., McCarthy, K. & Zabar, B. (2007), 'Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?', *DMIN* pp. 35–41.

Wolpert, D. (1992), 'Stacked generalization', *Neural Networks* **5:2**, 241–260.