«Aplicación de técnicas de aprendizaje automático supervisado para hallar los potenciales clientes a un nuevo servicio ofrecido por una entidad bancaria»

by

«Luis Pablo Urizar Catalán»

A thesis submitted in conformity with the requirements
for the MSc in Economics, Finance and Computer Science

University of Huelva & International University of Andalusia

September 2022

«Aplicación de técnicas de aprendizaje automático supervisado para hallar los potenciales clientes a un nuevo servicio ofrecido por una entidad bancaria»

«Luis Pablo Urizar Catalán»

Máster en Economía, Finanzas y Computación

«Diego Martin Santos y Manuel Emilio Gegúndez Arias»

Universidad de Huelva y Universidad Internacional de Andalucía

2022

Abstract

The objective of this work is the application of supervised learning techniques to identify the clients with the greatest probability of subscribing to a loan service offered by a bank. Seven learning models are analyzed: K Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), Naive Bayes, Support Vector Machine (SVM), Decision Tree, Tree-Bagging and Random Forest. These models were applied with different configurations (training with balanced and unbalanced classes using all the available instances, models contemplating all the attributes and models working with a selection of these.). For this, a database composed of two data sets has been used, one for training with 30,488 observations and the other for testing with 3,090 instances, each containing 19 input parameters that described the client and the macroeconomic context. The model with the best results has been the Random Forest using the complete set of observations and all the available attributes. This model can work with a success rate of 94%, presenting a sensitivity and specificity of 95% and 94%, respectively. These values indicate that the model is optimal for correctly identifying 352 potential clients out of 370 potential clients, at the cost of including a 6% error in their predictions (misclassifying 161 out of 3668 clients who do not intend to subscribe to

the loan). In this way, the selected algorithm satisfies the necessary requirements to be used by a banking entity to focus its marketing campaign.

JEL classification: E5, E7, G2

Key words: sensibilidad, aprendizaje supervisado, modelos de clasificación, punto de umbral optimo, métodos de ensamble.

Resumen

El objetivo del presente trabajo es la aplicación de técnicas de aprendizaje supervisado para identificar a los clientes con mayor probabilidad de suscribirse a un servicio de préstamo ofrecido por un banco. Se analizan 7 modelos de aprendizaje: K Vecinos más cercanos (KNN), Análisis Discriminante Lineal (LDA), Naive Bayes, Maquinas de Vector Soporte (SVM), Árbol de Decisión, Tree-Bagging y Random Forest. Se aplicaron estos modelos con distintas configuraciones (entrenamiento con clases balanceadas y sin balancear utilizando todas las instancias disponibles, modelos contemplando todos los atributos y modelos trabajando con una selección de estos.). Para ello se ha utilizado una base de datos compuesta por 2 conjuntos de datos, uno de entrenamiento con 30.488 observaciones y otro de prueba con 3.090 instancias, cada uno contenía 19 parámetros de entrada que describían al cliente y el contexto macroeconómico. El modelo con mejores resultados ha sido el Random Forest utilizando el conjunto completo de observaciones y todos los atributos disponibles. Este modelo es capaz de trabajar con una tasa de acierto del 94%, presentando una sensibilidad y especificidad del 95% y 94% respectivamente. Estos valores indican que el modelo es óptimo para identificar correctamente 352 clientes potenciales de 370 clientes potenciales, a costa de incluir un 6% de error en las predicciones de estos (clasificando erróneamente 161 de 3668 clientes que no tienen intención de suscribir al préstamo). De esta forma el algoritmo seleccionado satisface los requisitos exigibles para poder ser utilizado por una entidad bancaria para focalizar su campaña de marketing.

Índice

1 Propuesta del Trabajo de Fin de Máster 1

1.1 «Motivación» 1

1.2 «Objetivos» 2

1.3 «Organización del documento» 3

2 Marco Teórico 5

2.1 «Técnicas de Aprendizaje Supervisado» 5

2.2 «Modelos de Clasificación» 6

2.2.1 «K Vecinos Más Cercanos (KNN)» 6

2.2.2 «Análisis Discriminante Lineal (LDA)» 7

2.2.3 «Naive Bayes» 8

2.2.4 «Arboles de Decisión (Tree)» 10

2.2.5 «Tree-Tree-Bagging» 12

2.2.6 «Bosques Aleatorios (*Random forest*)» 12

2.2.7 «Máquinas de Vector Soporte (SVM)» 14

2.3 «Selección de Atributos» 17

2.3.1 «Metodología de Ranking» 18

2.3.2 «Metodología de Filtro» 18

2.4 «Evaluación de los Modelos de Aprendizaje» 19

2.4.1 «Holdout» 20

2.4.2 «Validación Cruzada de k iteraciones (K-fold)» 20

2.4.3 «Leave-One-Out» 21

2.4.4 «Métricas de Clasificación: Clasificación Binaria» 21

2.4.5 «Evaluación Clasificadores Binarios: Curva ROC» 23

3 Descripción de la Base de Datos 25

4 Metodología 30

4.1 «Preprocesamiento de datos» 30

4.1.1 «Datos Desconocidos» 31

4.1.2 «Normalización de Datos» 31

4.1.3 «Conjunto Balanceado» 32

4.2 «Proceso de Experimentación» 32

4.2.1 «Generación de los modelos de clasificación utilizando todos los parámetros de entrada» 32

4.2.2 «Contraste de los modelos utilizando un proceso de selección de atributos y utilizando todas las instancias disponibles» 35

4.2.3 «Implementación del modelo final» 36

5 Pruebas y Resultados 39

5.1 «Preprocesamiento de datos» 39

5.2 «Generación de los modelos de clasificación utilizando todos los parámetros de entrada» 41

5.2.1 «Análisis Discriminante Lineal» 41

5.2.2 «K Vecinos más cercanos» 42

5.2.3 «Naive Bayes» 44

5.2.4 «Máquinas de Vector Soporte» 46

5.2.5 «Árbol de Decisión» 47

5.2.6 «Tree-Tree-Bagging» 48

5.2.7 «Random Forest» 50

5.2.8 «Comparación de los Resultados» 51

5.3 «Contraste de los modelos utilizando un proceso de selección de atributos y utilizando todas las instancias disponibles» 52

5.3.1 «Generación de los modelos utilizando un proceso Selección de Atributos» 52

5.3.2 «Generación de los modelos utilizando todas las instancias disponibles» 61

5.4 «Implementación del modelo final» 67

5.4.1 «Contraste del modelo utilizando el punto de umbral óptimo de la curva ROC» 67

5.4.2 «Contraste del modelo modificando el punto de umbral en base a la sensibilidad» 70

6 Conclusiones 74

Referencias 76

Apéndice 80

Lista de Tablas

Tabla 01. Parámetros con valores faltantes en el conjunto de entrenamiento 39

Tabla 02. Parámetros con valores faltantes en el conjunto de prueba 40

Tabla 03. Metricas de Clasificación Binaria del modelo LDA con todos los atributos 41

Tabla 04. Metricas de Clasificación Binaria del modelo KNN con todos los atributos 43

Tabla 05. Metricas de Clasificación Binaria del modelo KNN con todos los atributos, utilizando un K=18, hallado por el método de LOO 43

Tabla 06. Metricas de Clasificación Binaria del modelo Naive Bayes con todos los atributos 45

Tabla 07. Metricas de Clasificación Binaria del modelo SVM con todos los atributos 46

Tabla 08. Metricas de Clasificación Binaria del modelo de Árbol de Decisión con todos los atributos 47

Tabla 09. Metricas de Clasificación Binaria del modelo de Bagging con todos los atributos .. 49

Tabla 10. Metricas de Clasificación Binaria del modelo de Random Forest con todos los atributos 50

Tabla 11. Comparación de los métodos 7 métodos generados con todos los atributos 51

Tabla 12. Metricas de Clasificación Binaria del modelo SVM con selección de atributos por filtro 54

Tabla 13. Metricas de Clasificación Binaria del modelo de Árbol de Decisión con selección de atributos por filtro 56

Tabla 14. Metricas de Clasificación Binaria del modelo de Bagging con selección de atributos por filtro 57

Tabla 15. Metricas de Clasificación Binaria del modelo de Random Forest con selección de atributos por filtro..... 59

Tabla 16. Resumen de las métricas de clasificación binaria obtenida por todos los modelos .. 60

Tabla 17. Metricas de Clasificación Binaria del modelo de Árbol de Decisión sin utilizar muestra balanceada..... 62

Tabla 18. Metricas de Clasificación Binaria del modelo SVM sin utilizar muestra balanceada 63

Tabla 19. Metricas de Clasificación Binaria del modelo Bagging sin utilizar muestra balanceada 64

Tabla 20. Metricas de Clasificación Binaria del modelo de Random Forest sin utilizar muestra balanceada 65

Tabla 21. Comparación de las Metricas de Clasificación Binaria con los modelos que no utilizan una muestra balanceada (ME)..... 66

Tabla 22. Metricas de Clasificación Binaria de los 3 modelos seleccionados utilizando el umbral optimo 69

Tabla 23. Niveles de Sensibilidad 71

Tabla 24. Metricas de Clasificación Binaria del modelo Random utilizando el cambio de umbral..... 72

Lista de Ilustraciones

Ilustración 1. Ejemplo de un modelo KNN..... 7

Ilustración 2 Ejemplo de un árbol de decisión..... 10

Ilustración 3 Hiperplano de separación máximo para un modelo SVM..... 15

Ilustración 4. Demostración método de Holdout..... 20

Ilustración 5. Demostración del método de validación cruzada de k=6 iteraciones..... 21

Ilustración 6 Curvas ROC donde se muestra clasificador perfecto, clasificador típico y clasificador equivalente a una suposición aleatorio..... 24

Ilustración 7. Curva ROC del modelo LDA, utilizando todos los atributos..... 42

Ilustración 8. Curva ROC modelo KNN, utilizando todos los atributos..... 44

Ilustración 9. Curva ROC del modelo Naive Bayes, utilizando todos los atributos..... 45

Ilustración 10. Curva ROC del modelo SVM, utilizando todos los atributos..... 46

Ilustración 11. Árbol de decisión optimo utilizando todos los atributos..... 47

Ilustración 12. Curva ROC del modelo de Árbol de decisión, utilizando todos los atributos..... 48

Ilustración 13. Curva ROC del modelo de Tree-Tree-Bagging, utilizando todos los atributos..... 49

Ilustración 14. Curva ROC del modelo de Random Forest, utilizando todos los atributos..... 50

Ilustración 15. Ranking de predictores utilizando el test de chi-cuadrado..... 52

Ilustración 16. Curva ROC del modelo SVM, utilizando selección de atributos..... 55

Ilustración 17. Árbol de decisión optimo, utilizando selección de atributos..... 56

Ilustración 18. Curva ROC del modelo de Árbol de decisión, utilizando selección de atributos..... 57

Ilustración 19. Curva ROC del modelo de Tree-Tree-Bagging, utilizando selección de atributos 58

Ilustración 20. Curva ROC del modelo de Random Forest, utilizando selección de atributos..... 59

Ilustración 21. Árbol de decisión optimo utilizando todas las instancias sin muestra balanceada 62

Ilustración 22. Curva ROC del modelo de Árbol de decisión sin utilizar muestra balanceada.... 63

Ilustración 23. Curva ROC del modelo SVM sin utilizar muestra balanceada 64

Ilustración 24. Curva ROC del modelo de Tree-Tree-Bagging sin utilizar muestra balanceada.. 65

Ilustración 25. Curva ROC del modelo de Random Forest sin utilizar muestra balanceada..... 66

Ilustración 26. Curva ROC del modelo Tree-Tree-Bagging Optimo..... 68

Ilustración 27. Curva ROC del modelo Random Forest Optimo utilizando una muestra balanceada..... 68

Ilustración 28. Curva ROC del modelo Random Forest Optimo no utilizando una muestra balanceada..... 68

Ilustración 29. Curva ROC del modelo optimo seleccionado..... 70

Ilustración 30. Matriz de confusión utilizando una sensibilidad del 90% 71

Ilustración 31. Matriz de confusión utilizando una sensibilidad del 90% 71

Ilustración 32. Matriz de confusión utilizando una sensibilidad del 97% 72

Ilustración 33. Matriz de confusión utilizando una sensibilidad del 99% 72

Lista de Apéndices

Apendice 1, Árbol de decisión optimo encontrado utilizando todos los atributos 80

Apendice 2, Árbol de decisión optimo encontrado utilizando solo los atributos seleccionados por la metodología de Filtro utilizando el test de Chi-cuadrado 81

Apendice 3, Árbol de decisión optimo utilizando todas las instancias del conjunto de entrenamiento, sin utilizar una muestra balanceada..... 82

1 Propuesta del Trabajo de Fin de Máster

1.1 «Motivación»

Las campañas de marketing sobre ventas constituyen una estrategia típica para mejorar las ventas de un negocio. Las compañías utilizan el marketing dirigido cuando se tiene como objetivo un segmento de consumidores y se contacta con ellos para alcanzar un objetivo específico.

Existen dos acercamientos principales de las empresas para promover sus productos y/o servicios, una de ellas es a través de las campañas masivas, donde no hay un público objetivo definido y el segundo, por medio del marketing directo o dirigido, donde se tiene como objetivo un nicho especifico.

Las campañas de telemarketing son de las más utilizadas para contactar directamente con los consumidores. Uno de los organismos que utilizan esta técnica son las entidades bancarias, que lo utilizan para ofrecer diferentes servicios, como son tarjetas, préstamos y suscripciones a depósitos a largo plazo. Sin embargo, debido a las ultimas crisis financieras que se han desarrollado, existe una presión sobre los bancos para aumentar sus activos financieros.

Una de las estrategias adoptadas por estas entidades es ofrecer atractivas solicitudes de depósitos bancarios a largo plazo, con buenas tasas de interés, mediante estas técnicas de marketing dirigidas; se busca que estas sean lo más eficientes posibles para realizar la menor cantidad de contactos a los clientes posible, pero manteniendo un número aproximado de clientes que aceptan y se suscriben a este servicio.

Para poder cumplir con este objetivo, la selección del mejor conjunto de clientes, es decir, buscar aquellos que sean más probables que se suscriban a un producto, es sumamente importante. Existen diferentes tecnologías y metodologías que pueden ayudar a hallar a estos clientes, a través de técnicas de *data mining*.

La clasificación es la tarea más común en el *data mining*. El objetivo de estas metodologías es construir un modelo a través de los datos que aprenda una función que mapea los parámetros de entrada que caracterizan a una instancia, acompañada con una etiqueta de salida, representando la clase a la que pertenece la instancia.

Existen diferentes metodologías, que se encargan de construir un modelo predictivo que puede etiquetar una serie de datos en una de varias clases. En el presente trabajo se busca analizar estas metodologías de aprendizaje supervisado, y hallar el que presente la solución óptima para el banco, ayudando a focalizar sus campañas de telemarketing al ofrecer una nueva suscripción a un depósito bancario. Hallando que EL modelo de aprendizaje final debe discriminar con precisión entre aquellos clientes que se suscriben al servicio de los que no. Los datos utilizada para generar estos modelos durante la investigación es recolectada por campañas de marketing relacionados con la suscripción de un depósito bancario en Portugal, durante los años 2008 a 2010.

1.2 «Objetivos»

El objetivo general de esta investigación es hallar el modelo de clasificación óptimo para discriminar y predecir la mayor cantidad de clientes que posean un alto potencial de suscribirse a un nuevo servicio de depósito bancario, brindando así una solución al banco para focalizar sus campañas de telemarketing.

Para lograr esto se plantean los siguientes objetivos específicos:

A. Encontrar un método óptimo para tratar los posibles datos faltantes en los conjuntos de datos, evitando comprometer información que se tiene acerca de los clientes que se desean suscribir al servicio.

B. Contrastar si un proceso de selección de atributos, para seleccionar aquellos más eficientes y eliminar los que menos influencia tengan, y a través de esto mejorar su rendimiento de discriminación y exactitud global.

C. Dado que el conjunto de datos está muy desbalanceado entre las 2 clases de nuestro problema, analizar si el balanceo de instancias genera una mejor predicción al clasificar los clientes.

D. Hallar a través de la Curva ROC, el punto de trabajo óptimo del modelo para clasificar las instancias que mejor se ajuste a los posibles criterios bancarios.

1.3 «Organización del documento»

Además de este primer capítulo de introducción al trabajo, esta memoria se ha dividido en los siguientes cuatro capítulos

1.3.1 «Capitulo 2: Marco Teórico»

En este capítulo, se desarrollan todas aquellas definiciones y conceptos que son relevantes para nuestra investigación. Brindado una explicación de cada uno de estos para brindar un mayor contexto y facilitar la comprensión de la investigación.

Los conceptos que se mencionan en este capítulo se pueden agrupar en 4 grandes rubros, el primero, con los conceptos básicos del aprendizaje supervisado. En el segundo se agrupan las diferentes técnicas de clasificación que se utilizaran a lo largo de la investigación: K vecinos más cercanos, Análisis discriminante lineal, Naive Bayes, Árbol de decisión, Maquinas de vector soporte, Tree-Bagging y Random Forest.

El tercer conjunto se describen las diferentes técnicas para la selección de atributos al momento de generar los modelos de clasificación. Mientras que en el último rubro su agrupan los conceptos acerca de diferentes técnicas, métodos y métricas que se pueden utilizar para evaluar un modelo, como lo es la Sensibilidad, Especificidad y el Área Bajo la Curva (AUC).

1.3.2 «Capitulo 3: Descripción de la Base de Datos»

En el desarrollo de este capítulo, se describe la base de datos utilizada, brinda más información acerca de los dos conjuntos de datos que se utilizaron durante la investigación (conjunto de entrenamiento y conjunto de prueba). Dando a conocer el origen de todos estos datos, el número de instancias con el que contaba cada conjunto, así como los diferentes parámetros que se tienen disponibles acompañados de una breve descripción de estos.

1.3.3 «Capitulo 4: Metodología Implementada»

Durante este capítulo se narran los diferentes pasos a seguir durante la investigación. Se desarrolla una explicación de cada una de las etapas que se elaboraron, desde la fase del preprocesamiento de los datos, hasta la implementación del modelo optimo encontrado.

En este capítulo se describen cada una de las acciones, pruebas y contrastes que se harán con la base de datos, brindando las características acerca de las diferentes metodologías de clasificación que se implementaran y como se generaron sus respectivos modelos, así como exponer las diferentes consideraciones que se deben tener al momento de generar cada uno de estos.

Adicionalmente se describe cual fue el proceso para evaluar y contrastar los diferentes modelos y configuraciones generados, para ir descartando estos, hasta hallar el que se puede considera más optimo como solución para el banco.

1.3.4 «Capitulo 5: Pruebas y Resultados»

En este capítulo, en base a lo descrito en la metodología, se exponen todos los resultados hallados durante las diferentes etapas de investigación. Se muestran todas las tablas, gráficos y resultados relevantes que se encontraron en cada una de las diferentes etapas de investigación, acompañados de una breve discusión de los hallazgos.

A través de este capítulo se demostrará con los resultados, como fue el proceso y cuáles eran los criterios que dictaban el descarte de diferentes configuraciones y modelos, hasta determinar cuál de estos, es el modelo optimo y porque es la solución óptima al problema planteado al inicio de la investigación.

1.3.5 «Capitulo 6: Conclusiones»

A partir de los resultados obtenidos durante todo el proceso de investigación, se forman y concretan todas las conclusiones o *insights* hallados a través de los capítulos anteriores. Estos además dan una respuesta a los objetivos que se plantean al inicio de la investigación determinando si estos se llegaron a cumplieron o no y el porqué. Adicionalmente en este capítulo se describe la solución óptima hallada para encontrar los clientes con mayor potencial a suscribirse al depósito ofrecido.

2 Marco Teórico

Las técnicas de clasificación en la minería de datos tienen como objetivo el buscar y obtener patrones, para así generar modelos a partir de una serie de datos recopilados. Estas tecinas se dividen en 2 categorías importantes, las técnicas supervisadas o también conocidas como predictivas y las técnicas no supervisadas, también llamadas descriptivas (Molina & García, 2006). Durante esta investigación se trabajará con la primera de estas.

Estas técnicas se utilizan, como bien su nombre lo dice para predecir el comportamiento que tendrá alguna entidad o evento.

2.1 «Técnicas de Aprendizaje Supervisado»

Los algoritmos de aprendizaje automático supervisado se caracterizan por contar con un conjunto de datos u observaciones, en los que cada observación viene acompañada por un valor o etiqueta, esta nos indica la clase a la que pertenece esta instancia. Esta clase nos indica cual es resultado final del evento. (Contreras, Fuentes & Trujillo, 2021)

Estos algoritmos se pueden dividir en dos categorías, las técnicas de clasificación y las de regresión. La primera de estas se caracteriza por contar con un conjunto de posibles clases o salidas finitas, que se interpretan como la clase a la que pertenece la observación. Mientras que la técnica de regresión tiene como objetivo establecer un método para relacionar diferentes características a una variable continua, que se considera como variable objetivo. (Contreras, Fuentes & Trujillo, 2021). Durante esta investigación nos enfocaremos en las técnicas de clasificación.

Los algoritmos de clasificación cuentan con 2 procesos principales muy importantes, el proceso de Entrenamiento y el proceso de Prueba o Clasificación. El proceso de entrenamiento tiene como objetivo, utilizar los datos de entrenamiento para que a través de diferentes configuraciones encontrar la mejor combinación de parámetros e hiperparámetro para generar así el algoritmo y evaluar el nivel de fiabilidad de este. Mientras que el proceso de prueba es en el cual el algoritmo ya entrenado es aplicado, a un nuevo conjunto de datos que se desea procesar, para hacer una predicción/clasificación a cada observación sobre su resultado final. (Baviera, 2016)

2.2 «Modelos de Clasificación»

A continuación, hablaremos sobre algunos de los modelos y algoritmos comúnmente utilizados en este tipo de técnicas: Modelo de K-vecinos más cercanos, Análisis discriminante lineal, Árboles de decisión, Máquinas de vector soporte, Naive Bayes, Tree-Bagging y Bosques Aleatorios.

2.2.1 «K Vecinos Más Cercanos (KNN)»

Este algoritmo tiene como objetivo clasificar una nueva observación a través de la distancia de esta nueva instancia con respecto al resto de las observaciones del conjunto de entrenamiento y la clase a las que estas pertenecen.

Esta nueva observación obtendrá la misma clasificación que tienen las observaciones más cercanas del conjunto de entrenamiento. Se conoce como "*KNN"* debido a que, para realizar la clasificación de las nuevas observaciones, se tienen que observar las K observaciones más cercanas a esta nueva instancia. Por lo que el hiperparámetro K nos indica cuantos vecinos hay que observar para poder predecir a que clase pertenece esta nueva observación. (Berástegui, 2018)

Para este tipo de clasificadores es importante normalizar los datos, ya que cada atributo puede tener un rango de valores distinto al de otra variable y para evitar que los datos de diferentes atributos se solapen, se normalizan los datos para que tomen valores entre 0 y1, de esta manera lograr que todos los parámetros afecten de igual manera en el cálculo de la distancia, es importante recalcar que esta metodología, solo utiliza parámetros numéricos. (Berástegui, 2018)

Posterior a la normalización de los atributos, se debe calcular la distancia de esta nueva observación con respecto al resto de las observaciones de entrenamiento. Existen diferentes formas de calcular la distancia.

La que se utilizara durante esta experimentación es la distancia Euclídea, que se define la distancia entre dos puntos, como la línea recta que une estos.

En la siguiente imagen, se muestra un ejemplo de una gráfica donde se utiliza esta metodología, donde *X* representa la nueva observación que se quiere clasificar. Se puede observar que, si se toman las 10 observaciones más cercanas a X, se determina que pertenecería a la categoría de "versicolor".

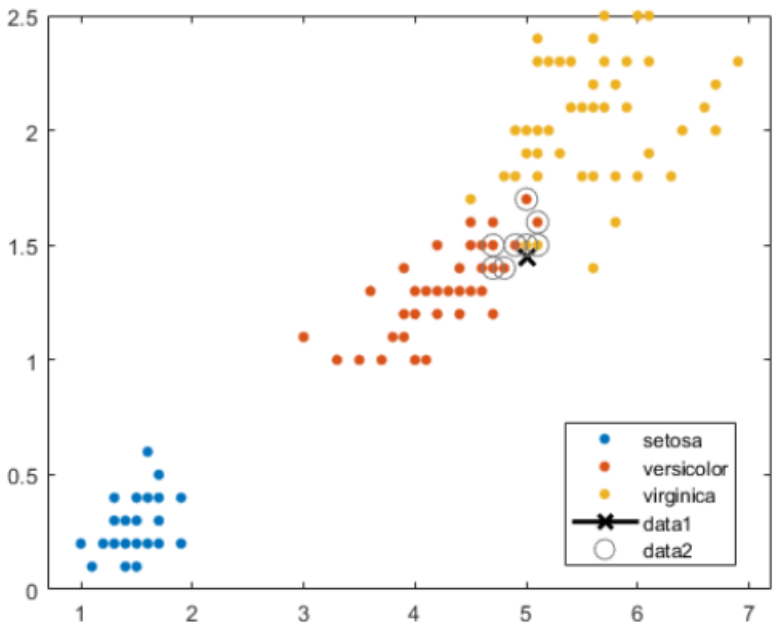


Ilustración 1. Ejemplo de un modelo KNN. Source: 'Clasification Using Nearest Neighbors', Matlab & Simullink

2.2.2 «Análisis Discriminante Lineal (LDA)»

Esta es una metodología de clasificación que utiliza únicamente variables numéricas y en la que dos o más de sus clases son conocidos a priori. Al momento de ingresar una nueva observación, esta se clasificará en una de estas en función de sus características. Utilizando el teorema de Bayes, se estima que la probabilidad dado un determinado valor en los predictores asignara la clase de pertenencia a cada una de las clases de la variable, $P(Y=k|X=x)$, donde se asigna a esta nueva observación a la clase que presente una probabilidad mayor. (Amat, 2016).

Esta metodología presenta algunas ventajas en comparación a la regresión logística, ya que el LDA es más estable, en el caso de que las clases se encuentren bien separadas o en el caso de que se cuenta con un numero alto de observaciones y que la distribución de los predictores sea aproximadamente normal para cada una de las clases.

El proceso de esta metodología se podría agrupar en 4 grandes pasos. El primero, calcular la proporción esperada de observaciones que pertenecerán a cada una de las clases, también denominadas *prior probabilities*. Posterior a esto se debe de observar si la matriz de covarianza es homogénea para todos los grupos, ya que de esto de penderá si se utiliza el método LDA o QDA

(análisis discriminante cuadrático), este a diferencia del LDA, asume que cada clase tiene su propia matriz de covarianzas. Posteriormente se estiman los parámetros necesarios para las funciones de probabilidad condicional. Por último, se calcula el resultado de la función discriminante, el resultado de esta asignara a cada observación la clase a la que pertenece. (Amat, 2016).

El teorema de bayes se puede enfocar para la clasificación, para que esto pueda suceder es necesario conocer la probabilidad poblacional de que una observación pertenezca a cada clase, así como también la probabilidad poblacional de que una observación que pertenece a la clase k adquiera el valor x en el predictor. (Amat, 2016).

En el caso que se desee clasificar una nueva observación en una de las k clases de una variable *Y*, a partir de un solo predictor *X*. Se cuenta con la probabilidad previa π_k, que representa la probabilidad de que una observación pertenezca a la clase *k*. Adicionalmente se cuenta con la función de densidad de probabilidad condicional de *X* para una observación que pertenece a la clase *k* $(f_k(X) = P(X = x|Y = k))$ y entre mayor sea el valor obtenido, mayor será la probabilidad de que esta observación de la clase k adquiera el valor de x. Por último, se necesitan las probabilidades a posteriori, estas se definen como la probabilidad de que una observación pertenezca a la clase *K*, conociendo el valor *x* del predictor. $(f_k(X) = P(Y = k|X = x))$, que utilizando el Teorema de bayes podemos encontrar este valor con la siguiente ecuación

$$P(Y = k|X = x) = \frac{\pi_k P(X = x|Y = k)}{\sum_{j=1}^{K} \pi_j P(X = x|Y = j)}$$

Dado que el denominador de esta fórmula será igual para todas las clases, se puede decir que a la observación se le asignará la clase de aquel grupo que presente un valor mayor en $\pi_k P(X = x|Y = k)$. (Amat, 2016).

2.2.3 «Naive Bayes»

Este modelo al igual que el LDA, utiliza el teorema Bayes, junto con una fuerte suposición en que los atributos son condicionalmente independientes dada una clase, a diferencia del LDA. Esta suposición a menudo se viola al experimentar con conjunto de datos verdaderos, a pesar de esto suele proporcionar una mejor precisión en la clasificación de las observaciones que otros modelos. (Mosquera, Castrillón & Parra, 2018)

Este modelo utiliza como base la teoría de probabilidades, así como la frecuencia para poder calcular las probabilidades condicionales y poder realizar predicciones sobre nuevos casos.

Para representar esto, suponiendo un ejemplo donde la observación *E* es representado por un conjunto de atributos con valores *x1, x2, …, xn*. En el que x_i es el valor del atributo *Xi* y adicionalmente se cuenta con el valor *C*, que es la etiqueta de clasificación y suponemos que *c* es el valor de *C*, y esta tienen 2 posibles clases diferentes, la clase positiva (+) y la clase negativa (-). (Zhang, 2004)

Desde la perspectiva de la probabilidad y según las reglas de Bayes, la probabilidad que tiene la observación *E= (x1, x2, …, xn)* de pertenecer a la mencionada clase c, viene dada por la siguiente formula:

$$p(c|E) = \frac{p(E|c)p(c)}{p(E)}.$$

E puede pertenecer a la clase *C* positiva (C=+), si y solo si

$$f_b(E) = \frac{p(C = +|E)}{p(C = -|E)} \geq 1,$$

Por lo que deducimos que, en este caso, *E* pertenecerá a la clase *C*, que tenga una mayor probabilidad condicional. (Zhang, 2004)

Esta formula $f_b(E)$ se conoce como clasificador Bayesiano. Como se mencionaba anteriormente este modelo, asume que todos los atributos son independientes dado el valor de una clase, esto se puede representar con la siguiente por la ecuación $p(E|c) = p(x_1, x_2, \dots, x_n|c) = \prod_{i=1}^{n} p(x_i|c)$, en base a esto el clasificador sigue la siguiente formula:

$$f_{nb}(E) = \frac{p(C = +)}{p(C = -)} \prod_{i=1}^{n} \frac{p(x_i|C = +)}{p(x_i|C = -)}$$

(Zhang, 2004)

Algunas de características que tiene este modelo, es que en este se pueden utilizar tanto como atributos numéricos, como categóricos, a diferencia de los 2 modelos explicados anteriormente.

2.2.4 «Arboles de Decisión (Tree)»

Este es un modelo que guarda semejanzas a los sistemas de predicción basados en reglas, ya que sirven para representar y categorizar una serie de condiciones que suceden de forma sucesiva para así obtener una solución a un problema.

Estos modelos se representan mediante un árbol, esto quiere decir que se representa por un conjunto de nodos, hojas y ramas. El árbol inicia con el nodo raíz, que es por el cual se inicia el proceso de clasificación. Cada nodo representado dentro del árbol corresponde a un cuestionamiento acerca de un parámetro del problema en particular, dependiendo de la respuesta a este cuestionamiento se varía el recorrido por el árbol, cada una de las respuestas produce un nodo hijo. Las ramas que se originan de cada nodo van acompañadas por una etiqueta con los posibles valores del atributo. Las hojas o también vistos como nodos finales, corresponden con una decisión, en nuestro caso, esta decisión es la clase a la que pertenecerá la nueva observación. (Barrientos, et al. 2009)

En la siguiente, imagen se puede observar un ejemplo de un árbol de decisión:

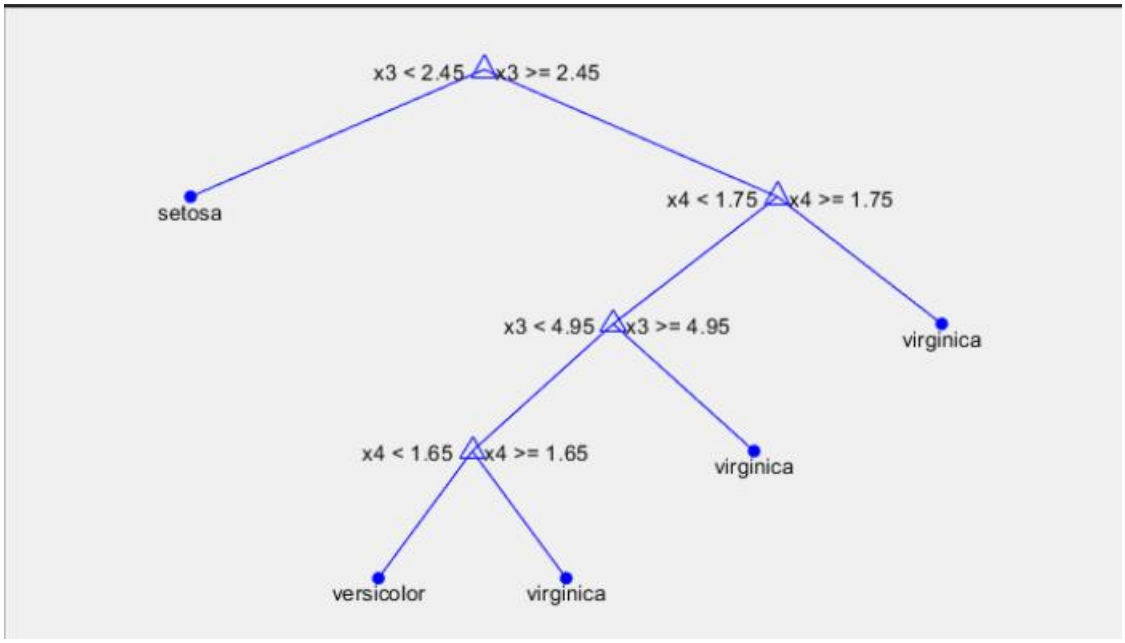


Ilustración 2 Ejemplo de un árbol de decisión. Source "Ver árbol de decisión" Matlab & Simulink

Existen 4 métodos de división para los árboles:

A. CHAID (Chi-square automatic interaction detector)

B. CHAID exhaustive

C. Árboles de clasificación y regresión

D. QUEST (Quick, unbiased, efficient, statistical tree)

Durante la experimentación nos enfocaremos únicamente en los árboles de clasificación y regresión. Este método se basa en un algoritmo de árbol binario que hace particiones de los datos y genera subconjuntos precisos y homogéneos, los datos se dividen en segmentos para que sean lo más homogéneos posibles con respecto a la variable dependiente. (Berlanga, Rubio, Vila, 2013)

La generación de un árbol de decisión se compone de 2 etapas, la primera es la inducción del árbol, es la etapa donde se construye el árbol a partir de los datos de entrenamiento que se poseen. El árbol inicia generando su nodo raíz, seleccionando uno de los atributos de prueba y a partir de este dividir los datos en dos o más subconjuntos, en cada partición se genera un nuevo nodo donde se repite el proceso y así sucesivamente. Cuando a un nodo llegan observaciones de más de una clase, se genera un nodo interno y se dividen estas observaciones de nuevo, mientras que, si al nodo únicamente llegan observaciones de una clase, este se transforma en una hoja y se le asigna la etiqueta de la clase. En la segunda etapa del proceso cada observación nueva es clasificada por el árbol anteriormente construido, el camino seguido se determinará a partir de las decisiones en cada nodo. (Barrientos, et al. 2009)

Estos árboles tienden a producir buenas predicciones en el conjunto de entrenamiento, que es con el cual se genera, por lo que puede llegar a ser un modelo muy ajustado a los mismos, por los que un árbol más pequeño, que presente menos divisiones podría mejorar la capacidad del árbol para generalizar el modelo global, a este proceso en el que se busca hacer el árbol más pequeño se le conoce como *Poda*. Existen 2 estrategias para podar un árbol, están los métodos Pre-poda y los Post-poda. Los primeros se realizan al construir un árbol y se dejan regiones cuando se cumpla un criterio basado en cuantificar la mejora que produce la división en términos de error. La desventaja principal de este es que no se puede tomar en cuenta que al realizar una división que al inicio no tiene mucho valor para el árbol, puede ir seguida por una división que sea importante o buena para

el árbol. Mientras que los post-poda, se basan en generar un árbol completo primero y luego podarlo para obtener el subárbol más optimo. (James, et. Al. 2017).

2.2.5 «Tree-Bagging»

Este modelo presentado por Breiman, realiza la evaluación de un conjunto de árboles de decisión distintos, generados a través de distintos conjuntos de datos por medio de remuestreo. Esto se hace con el objetivo de reducir la variabilidad que se tienen en los árboles de clasificación. (Cardona, 2019).

Esta metodología utiliza la técnica de *Bootstrap,* que es la encargada del remuestreo, se encarga de generar muestras extraídas de una muestra original, contemplando reemplazamiento, esto quiere decir que entre las diferentes muestras se pueden repetir observaciones u instancias, y cada una de estas muestras presenta el mismo tamaño. (Cardona, 2019).

El objetivo del *Tree-Bagging* es el de reducir la varianza que se tiene al utilizar únicamente un árbol de decisión. Supongamos que se cuenta con un conjunto de *n* instancias independientes $A_1, \ldots, A_n$, y cada una de estas con una varianza de σ^2. La varianza de la media $\bar{A}$ de las observaciones viene dada por σ^2/n. Esto nos indica que el promedio de un conjunto de observaciones reduce la varianza. Bueno pues en este caso se replica esto con los árboles de decisión, se busca generar un conjunto de árboles, para reducir la varianza que se tiene en un único árbol y a través de registrar la clase predicha por cada uno de los árboles del conjunto y decidirse por la clase mayoritaria entre las predicciones. (Blasco, 2018)

Una de las desventajas que tiene este modelo, es que pierde una cualidad que tenían los árboles de clasificación individuales, que era el poder utilizar el diagrama de un árbol para interpretarlo y entenderlo, se pierde interpretabilidad al usar el modelo de *Tree-Bagging*, pero se compensa haciendo que la precisión del predictor aumente en comparación a la del árbol individual.

2.2.6 «Bosques Aleatorios (*Random Forest*)»

Esta es una técnica de aprendizaje supervisada no paramétrico. Se conocen también como un método de ensamble, ya que se consideran capaces de impulsar métodos débiles y convertirlos en métodos fuertes y de esta manera presentar predicciones más precisas. (Zhou, 2012).

Los bosques aleatorios son una generalización de la metodología de *Tree-Bagging*, en la cual cada árbol del bosque se construye siguiendo el algoritmo CART, a los cuales posteriormente combina todas las predicciones o clasificaciones realizadas por cada uno de estos árboles, de esta forma el objetivo es reducir la alta varianza que tiene un árbol individualmente. Esta técnica se suele usar cuando se tiene una alta cantidad de observaciones y también cuando se tienen varias variables de entrada. (Cardona, 2019)

El algoritmo CART se refiere a que los árboles son de clasificación y regresión, además de que las diferentes divisiones que se forman en cada uno de los árboles se cuantifican a través de una medida de impureza. Una división se puede considerar como pura, si posterior a la división, todas las instancias de una elección de una rama pertenecen a la misma clase. La intención de esto es reducir la alta varianza que se tiene como respuesta de únicamente un solo árbol y mejorar el desempeño del método.

Esta técnica se basa en la producción de *n* remuestras aleatorias a partir del conjunto de datos de entrenamiento. Se debe contemplar que para la generación de estas *n* remuestras se debe de permitir el reemplazo, como se realiza en la metodología de *Bootstrap*, de esta manera se logra agregar a estas muestras ligeras variaciones aleatorias, pero cada muestra seguirá reflejando el comportamiento que tiene el conjunto de datos original. (Cardona, 2019).

A cada una de estas *n* remuestras se le genera un árbol de decisión, pero en este caso se tiene la cualidad de que, al llegar a un nodo o una división, se escoge una muestra aleatoria de *m* predictores candidatos, del conjunto de todos los predictores disponibles en el conjunto de datos original. Comúnmente se suele utilizar en cada una de estas divisiones un $m = \sqrt{p}$, donde *p* es la cantidad de predictores que se tienen disponibles en el conjunto de datos original. (Garcia, 2018)

Por lo que, en cada división del árbol, no se tomaran en cuenta todos los predictores disponibles, sino un grupo limitado, esto para evitar que en el caso de existir un predictor muy fuerte, la mayoría de los árboles utilizaran este predictor como principal en la división inicial, lo que puede desencadenar que muchos de los árboles generados tengan un aspecto similar entre ellos, por lo que la reducción de la varianza no sería óptima. (García, 2018)

Esto no sucedería en el caso de los bosques aleatorios, ya que, al forzar la utilización de diferentes subconjuntos de predictores, no siempre se consideran los predictores más fuertes y se podrá

observar el comportamiento a partir de otros predictores. Por lo que esta metodología reduce la correlación entre los árboles, y logrando que, al utilizar el promedio de estos, se reduzca la variabilidad del modelo.

2.2.7 «Máquinas de Vector Soporte (SVM)»

Esta es una técnica desarrollada por Vladimir Vapnik en los años 90, originalmente esta era utilizada para la resolución de problemas de clasificación binaria, pero en la actualizada este se llega a utilizar incluso en problemas de regresión y multiclasificación. (Carmona, 2016).

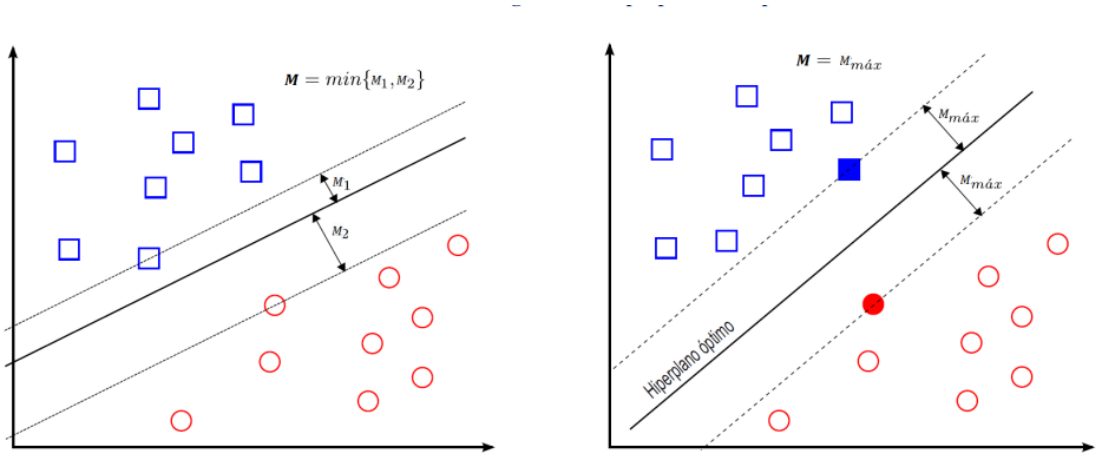
El objetivo de este algoritmo es el encontrar un hiperplano que logre dividir las clases de forma óptima, por esta razón esta técnica pertenece a la categoría de clasificadores lineales. Si el espacio y clases que se quieren dividir, no son linealmente separables, la búsqueda de este hiperplano de separación optimo se hará de forma implícita, utilizando alguna función de kernel.

Este método de aprendizaje se centra en minimizar el riesgo estructural, partiendo de que el objetivo es seleccionar un hiperplano de separación que equidista de los ejemplos más cercanos de cada clase para lograr encontrar un margen máximo a cada lado del hiperplano, al momento de conformar, solo se toman en cuenta aquellas observaciones de entrenamiento de cada clase que se presentan justo en la frontera de estos márgenes y estas observaciones se conocen como vectores soporte. (Carmona, 2016).

El hiperplano de separación está definido por la ecuación:

$$f(x) = w^T x + b$$

En la cual el w^T es el vector soporte y la variable b es la que representa el error. Este error es el que permite el movimiento del hiperplano hasta encontrar la ubicación ideal, que es el que minimiza este error. (Harrington, 2012; Ben-Hur & Weston, 2010; Yang et al, 2008).



Hiperplano de separación no óptimo y su margen asociado M (no máximo)

Hiperplano de separación óptimo o de margen máximo

Ilustración 3 Hiperplano de separación máximo para un modelo SVM. Source: "Tutorial sobre Máquinas de Vector Soporte (SVM)", Carmona, J. (2016)

Algunas de las ventajas de este método, es que la capacidad de generalización, debido a que lo que busca es minimizar el riesgo estructurado. Adicionalmente el modelo depende únicamente de los datos que contienen más información los más cercanos al hiperplano. En esta técnica el modelo final, podría ser descrito como una combinación de un número muy pequeño de vectores de entrada, los mencionados vectores de soporte. (Santibáñez, 2015)

En el caso de que las clases no puedan ser separadas por un hiperplano, se utilizan los clasificadores de vector soporte o también conocidos como clasificador de margen suave. En el caso de estos, si está permitido que ciertas observaciones se encuentren en lado incorrecto del margen e incluso el lado incorrecto del hiperplano, no como en caso del *Clasificador de margen máximo*, que busca que dividir ambas clases en su totalidad. Esta clase de clasificador cuentan con parámetro de ajuste *C*, este se encarga de medir el grado de violación del margen o hiperparámetro que serán toleradas. Entre mayor sea el valor que toma este parámetro el margen crecerá, con lo que aumenta la existencia de tolerancia a observaciones en el lado incorrecto del margen, mientras si es pequeño el modelo es más ajustado evitando y limitando la cantidad de observaciones en el lado incorrecto del hiperplano. (Santibáñez, 2015)

El hiperplano posee *N-1* dimensiones, donde la *N* es el número de variables que se utilizaran. En el caso de que lo que se quiere estudiar cuente con más de 2 dimensiones, o como anteriormente

se menciona, las clases no sean linealmente separables, es necesario utilizar una función que se encargue de pasar el problema a un nuevo espacio de características, normalmente agregando nuevas dimensiones, la función que se encargara de esto, es conocida como función kernel, utilizando estas es posible encontrar este hiperplano de separación. (Santibáñez, 2015).

Algunos de estos kernels son:

A. Kernel Lineal, que sigue la siguiente formula:

$$K(x_i, x_j) = x_i^T x_j.$$

B. Kernel Polinomial, que sigue la siguiente formula:

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$$

C. Kernel Base Radial, que sigue la siguiente formula:

$$K(x_i, x_j) = exp(-\gamma \|x_i x_j\|^2), \gamma > 0$$

Donde γ es el ancho del kernel radial, y d es el grado del kernel polinomial, y estos son los encargados de cuan ajustados serán los clasificadores y determinar la flexibilidad del clasificador. Si γ aumenta esto hará que el clasificador presente una mayor flexibilidad al ajustar el hiperplano y si este disminuye, este pierde flexibilidad. (Ben-Hur & Weston, 2010).

2.2.7.1 «Máquinas de Vector Soporte – Lineal (SVML)»

Esta metodología, utiliza el kernel clásico, que se describía con anterioridad, en este se busca que las observaciones de cada clase sean separadas a través de un hiperplano lineal. En este caso, como se mencionaba con anterioridad, se busca minimizar el riesgo estructural, encontrando el margen máximo a cada lado del hiperplano. (Carmona, 2016).

En este caso se utilizará la metodología de margen suave, agregando un hiperparámetro C, que nos permitirá hacer que nuestro modelo no sea tan restrictivo y aumentar su flexibilidad permitiendo una cantidad C de observaciones que se encuentran en el lado incorrecto del hiperplano permitidas. (Santibañez, 2015)

2.2.7.2 «Máquinas de Vector Soporte - Gaussiano (SVMG)»

Este modelo, como se mostró anteriormente, se encuentra definido por la siguiente función:

$$K(x_i, x_j) = exp(-\gamma \|x_i x_j\|^2), \gamma > 0$$

Donde el valor γ es el encargado de controlar el comportamiento del kernel. Adicional a esto este modelo también puede presentar el hiperparámetro C, que es el que se encarga de hacer el modelo más flexible, con un margen suave. En el caso de que ambos parámetros presenten valores muy altos puede haber una tendencia a que el modelo se sobreajuste a los datos de entrenamiento y el modelo puede presentar un alto porcentaje de error al discriminar las observaciones. (Argañaraz et al. 2011)

2.2.7.3 «Máquinas de Vector Soporte - Polinomial (SVMP)»

Las máquinas de vector soporte con kernel polinomial, como se presentó con anterioridad se definen con la siguiente formula $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$

Donde *d* es el grado del kernel polinomial. Si *d* fuera igual a 1, se está hablando de un clasificador de vector soporte lineal. Mientras que, si *d* es mayor a 1, equivale a ajustar un clasificador de vector soporte en un espacio de mayor dimensión, involucrando polinomios de grado *d*, en lugar del espacio característico original.

Este tipo de kernel, permite generar una frontera de decisión no lineal presentando mayor flexibilidad y capacidad de ajuste a los datos, que en el caso de las máquinas de vector soporte con kernel lineal. (Martin, 2016)

2.3 «Selección de Atributos»

La selección de atributos es una de las etapas del procesamiento de datos más importante, porque, aunque teóricamente el tener una alta cantidad de atributos que describen un evento, nos brindaría un mayor poder discriminatorio, esto no necesariamente se cumple y no siempre es lo más beneficioso. Al tener una alta cantidad de parámetros que describen un mismo evento, es probable que se tengan atributos redundantes y/o irrelevantes, que podrían afectar nuestras experimentaciones debido a que no se optimiza la utilización de recursos e incluso puede elevar los tiempos de ejecución de las metodologías de aprendizaje automático. Por lo que la eliminación de estas variables redundantes y/o irrelevantes pueden mejorar el comportamiento y la eficacia de

los modelos de clasificación, debido a que estas variables pueden confundir a los algoritmos de aprendizaje y se generan modelos más complejos. (Toledo,2016)

El objetivo es encontrar un subconjunto mínimo de atributos que logre que la tasa de aciertos no descienda significativamente y que la distribución de la clase resultante sea lo más semejante posible a la distribución real de la clase. (Ruiz, Aguilar & Riquelme, 2014)

Existen diferentes metodologías para delimitar cuales son estos parámetros, algunas de estas son:

A. Metodología de Ranking

B. Metodología de Filtro

2.3.1 «Metodología de Ranking»

Los algoritmos de ranking tienen como objetivo proporcionar una lista de parámetros de manera ordenada siguiendo una determinada medida de evaluación, como por ejemplo consistencia, información, distancia, dependencia y exactitud de algún algoritmo de aprendizaje. Para esto se le asigna un peso a cada uno de los atributos y se ordenan basándose en la relevancia de este atributo con respecto al concepto destino. El subconjunto de parámetros estará conformado por los *k* primeros atributos. (Ruiz, Riquelme & Aguilar, 2002)

La exactitud de la predicción de los modelos de clasificación en este caso depende del número (*k*) y de la calidad de las características que se seleccionaron en el ranking, por lo que el éxito de clasificación puede variar dependiendo de este número.

Esta técnica puede no cumplir con la minimización de variables, no eliminando aquellas que son redundantes, ya que, si el peso de 2 parámetros que miden lo mismo es suficientemente alto para entrar entre los *k* parámetros seleccionados, ambos se tomaran en cuenta, aunque sean redundantes. (Ruiz, Riquelme & Aguilar, 2002)

2.3.2 «Metodología de Filtro»

La selección por medio de las técnicas de filtro se hace previa a la utilización del modelo de aprendizaje, estos son similares a los métodos de ranking, ya que se encarga de establecer un ranking de las variables de entrada expresando la relevancia que tiene cada una de estas para poder discriminar de manera correcta las clases de la variable de salida. (Dorado, 2019)

Las variables que quedan por debajo de un umbral definido previamente son las variables que se van descartando. Normalmente en las técnicas de filtros se utilizan atributos estadísticos para la selección de los atributos o la entropía.

Algunas de las técnicas más comunes de filtro se encuentran:

A. Ganancia de Información: Este se basa en la disminución que tiene la entropía (medida de incertidumbre sobre la clasificación) en la clasificación de la variable respuesta después de que uno de los predictores se hiciera su aporte. Una desventaja sobre esta técnica es que la medida puede favorecer a las variables que tengan muchos valores. (Dorado, 2019)

B. Incertidumbre simétrica: Esta técnica se basa en el cálculo de la técnica de ganancia de información y la suma de la entropía en la variable de respuesta con la entropía de la variable de entrada y esto luego se multiplicar por un factor de corrección para estandarizar el valor en un intervalo [0,1], esta técnica a diferencia del anterior favorece a las variables con pocos valores. (Dorado, 2019)

C. Chi cuadrado: En este caso esta es una técnica que utiliza un test estadístico que se enfoca en medir la asociación entre dos variables. La hipótesis nula en esta prueba sugiere independencia entre estas variables. Entre más alto sean los valores obtenidos del estadístico, existe más evidencia para rechazar esta hipótesis nula, por lo que estas son las variables que son dependientes y tienen una influencia alta y formaran parte del subconjunto de parámetros seleccionados. (Dorado, 2019)

2.4 «Evaluación de los Modelos de Aprendizaje»

Evaluar un modelo significa que se debe estimar cual será la precisión al procesar datos nuevos o datos futuros, esto debido a que si se evalúa el modelo con los datos que se entreno puede presentar un muy buen funcionamiento, pero esto no significa que sea así de optimo al usarlo con nuevas observaciones, ya que el modelo pude haberse sobre ajustado a los datos de entrenamiento. Por lo mismo lo ideal es realizar una división de los datos que tenemos, en 2 conjuntos, el Conjunto de Entrenamiento, que agrupa los datos que se utilizaran para la fase de aprendizaje/entrenamiento del modelo; Y el conjunto de prueba, que agrupa los datos que se utilizar para evaluar el rendimiento del modelo entrenado. A continuación, se comentarán algunas técnicas de evaluación.

2.4.1 «Holdout»

En esta metodología se reserva un porcentaje de datos para la evaluación, mientras que el resto se utilizara para el entrenamiento del modelo de aprendizaje. El conjunto de prueba debe de ser lo más representativo posible. Con esta metodología puede llegar a existir una variación en la estimación del error u no resultar representativos los datos de entrenamiento y prueba. Se puede mejorar este método si se utiliza adicionalmente la técnica de validación cruzada. (Ochoa, 2019)

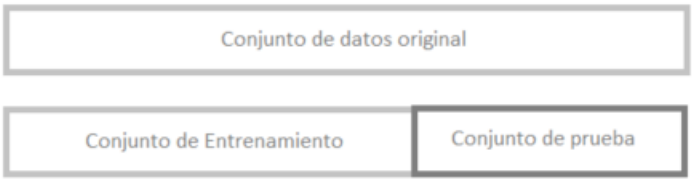


Ilustración 4. Demostración método de Holdout Source: Elaboración propia

La validación cruzada es otro método de evaluación, este tiene como objetivo utilizar todas las instancias tanto para entrenamiento como para pruebas. (Ochoa, 2019)

2.4.2 «Validación Cruzada de k iteraciones (K-fold)»

Esta metodología se basa en dividir el conjunto de datos originales en *k* subconjuntos. Al momento de entrenar el modelo, se va a utilizar un conjunto *k*, como el conjunto de datos de prueba, mientras que el resto de los subconjuntos se utilizara para entrenar el modelo. Esto se repetirá *k* veces hasta que todas las iteraciones hayan sido utilizadas como dato de entrenamiento y como dato de prueba. Posterior a esto se utilizará el promedio de los resultados de precisión y el error obtenido para cada subconjunto que se utilizó de prueba. (Ochoa, 2019)

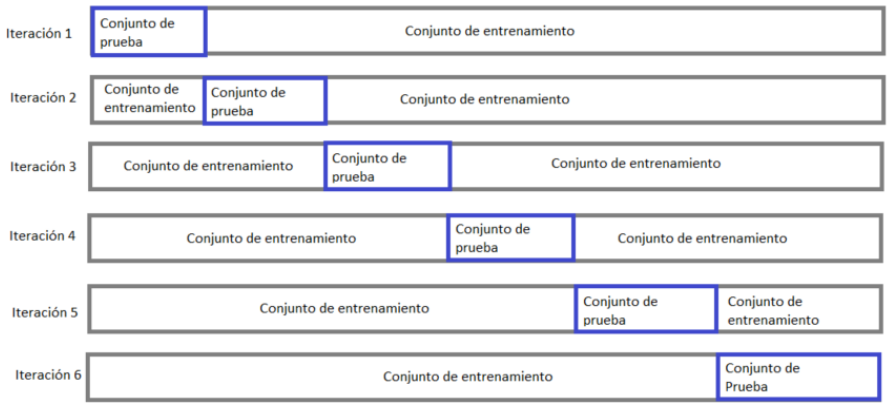


Ilustración 5. Demostración del método de validación cruzada de k=6 iteraciones. Elaboración propia

2.4.3 «Leave-One-Out»

En esta metodología se debe dejar una instancia fuera y el resto se utiliza como instancias de entrenamiento. Se utiliza la instancia restante para evaluar el modelo, esto se repite *k* veces hasta que se utilizan todas las instancias del conjunto, posteriores promedian los resultados y ese promedio representa la estimación de error final.

Esta metodología busca utilizar la mayor cantidad de instancias posibles para entrenar el modelo y de esta forma aumentar su rendimiento. Una de las desventajas que presenta esta metodología es que tiene un costo computacional y no es factible para conjuntos muy grandes de instancias, debido a la cantidad de repeticiones y procesamientos de datos que se tendría que hacer, por lo que es una metodología ideal para problemas con conjuntos pequeños de datos. (Ochoa, 2019)

2.4.4 «Métricas de Clasificación: Clasificación Binaria»

Las métricas de clasificación son diferentes cálculos que nos ayudaran a observar de manera general, que tan optimo ha sido nuestro modelo de aprendizaje posterior a ser probado con el conjunto de test. Estas métricas son importantes al momento de comparar diferentes modelos, para encontrar y tener una base para medir cual de estos tiene un mejor rendimiento discriminando cada una de las categorías de manera correcta. En este caso nos enfocaremos en las métricas de la clasificación binaria debido a que solo existen 2 posibles resultados, la clase positiva o la clase negativa.

Existen 4 datos muy importantes, que se utilizan en el cálculo de las métricas que nos ayudaran a comparar los modelos de aprendizaje, estos datos son:

A. *Verdadero Positivo (VP)*: Son aquellas instancias que están correctamente clasificadas por el sistema.

B. *Falso Positivo (FP):* Son aquellas instancias que el modelo clasifico como positiva, pero su instancia real pertenece a la clase negativa.

C. *Verdadero Negativo (VN):* Son todas aquellas instancias que el modelo de aprendizaje predice que pertenecen a la clase negativa, y la instancia realmente pertenece a esta clase.

D. *Falso Negativo (FN):* Son todas aquellas instancias que fueron clasificadas como que pertenecen a la clase negativa, pero la instancia en realidad pertenece a la clase positiva.

Por lo que podemos ver que los errores que ocurren en nuestro modelo son medidos por los valores de FN y FP. Estos valores nos serán útiles para calcular otras métricas que nos ayudarán a comprender y tener un mejor criterio acerca del rendimiento de nuestro modelo. Algunas de estas métricas son:

A. Tasa de Aciertos (Acc): Nos da la tasa de las instancias clasificadas correctamente por el modelo $Acc= (VP +VN / VP + FP +FN +VN)$

B. Sensibilidad (R): Este nos dice que, de todas las instancias positivas, la tasa que el modelo clasifico correctamente. $R=VP/VP+FN$

C. Especificidad (Sp): Se encarga de medir la tasa de observaciones que fueron predicas como clase negativas y se clasificaron correctamente. $Sp=VN/FP+VN$

D. Tasa Falsos Positivos (FPR): Nos brinda la tasa de observaciones predichas como negativas, que fueron erróneamente clasificadas por el modelo. FPR = FP/FP+TN

E. Precisión (P): Este nos indica la tasa de las observaciones que el modelo clasifica como positivas y fueron correctamente clasificadas P=VP/VP+FP

F. Medida F (F-score): Esta métrica es una combinación de 2 anteriores, Precisión y Sensibilidad, esto nos permite obtener una combinación más sencilla recudimiento combinado entre estas 2 métricas. F1=2*P*S/P+S (Amorós, 2021)

2.4.5 «Evaluación Clasificadores Binarios: Curva ROC»

El procedimiento que constituye el análisis ROC es la de elección de distintos niveles de decisión o valores de corte que permitan la clasificación dicotómica de los valores de la prueba según sean superiores o inferiores al valor elegido. Por lo que en este caso se tendrá un conjunto de pares correspondientes de sensibilidad y especificidad para cada uno de los distintos niveles de decisión. La curva ROC se obtiene representando para cada posible elección de valor de corte, la sensibilidad en las ordenadas y (1-especificidad) en las abscisas. (Lopez & Pita, 2001)

Por lo que podemos decir que la curva ROC es un resumen del conjunto de información que se necesitaría para describir el rendimiento de un clasificador sobre todos sus posibles valores de umbrales.

Esta curva es necesariamente creciente, para así reflejar la relación entre sensibilidad y especificidad, ya que, si se modifica el valor de corte para obtener mayor sensibilidad, se hace a expensas de disminuir la especificada. La exactitud de la prueba aumenta a medida que la curva se desplaza desde la diagonal hacia el vértice superior izquierdo, lo más cercano al punto (0,1), este es el punto en el que se tendría un rendimiento ideal. (Lopez & Pita, 2001)

Uno de los índices principales es el Área bajo la curva (AUC), como se comentó anteriormente la mayor exactitud de la prueba se encuentra en el punto (0,1) de la curva, y el área bajo esta curva se puede utilizar como un índice para medir la exactitud global de la prueba o en este caso de nuestro modelo de aprendizaje. La exactitud máxima correspondería a un AUC de 1. Mientras que la mínima es de 0.5, en el caso de tener un valor menor, se debería de invertir el criterio de positividad de la prueba. (Lopez & Pita, 2001)

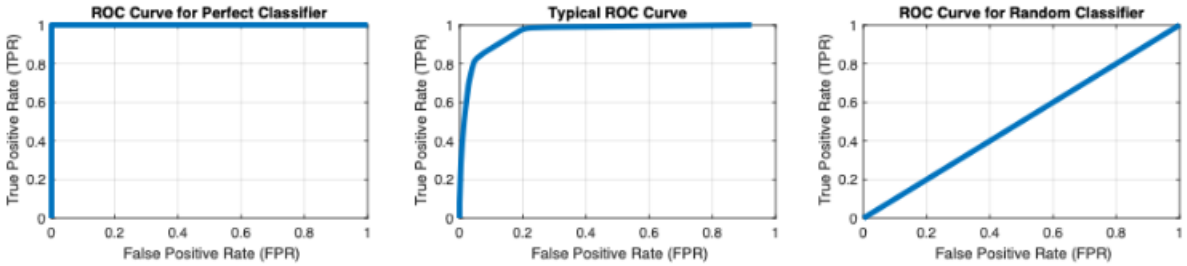


Ilustración 6 Curvas ROC donde se muestra (de izquierda a derecha) clasificador perfecto, clasificador típico y clasificador equivalente a una suposición aleatorio. Source: "Evaluación del rendimiento de los modelos de clasificación de machine learning", Matlab & Simulink

3 Descripción de la Base de Datos

La base de datos utilizada durante la experimentación está compuesta por 2 conjuntos de datos, estos utilizan información recolectada de diferentes campañas de telemarketing realizadas durante el periodo de mayo del 2008 a noviembre del 2010, realizadas por una entidad bancaria en Portugal.

Se cuentan con 2 conjuntos de datos prestablecido que se utilizaran durante esta investigación. Un conjunto de entrenamiento, que se utilizara para entrenar y generar los distintos modelos que se pondrán a prueba utilizando el otro es el conjunto brindado, el conjunto de test.

El conjunto de entrenamiento cuenta con un total de 41.188 instancias. Donde cada instancia cuenta con 20 parámetros o variables explicativas del modelo y una variable adicional, que es la variable respuesta, en este caso representa la clase a la que pertenece al cliente, indicando si el cliente se suscribió al servicio ofrecido en la campaña, o el cliente lo rechazo. Mientras que el conjunto de prueba únicamente cuenta con un equivalente al 10% de la cantidad de observaciones del conjunto de entrenamiento, es decir que este presenta 4,119 instancias. Con los mismos parámetros de entrada y de salida.

Las variables explicativas se pueden agrupar en 3 grandes categorías dependiendo lo que estas describen:

A. Variables relacionadas al cliente: Se cuentan con 7 variables que describen diferentes al cliente con el que se contactó el cliente para ofrecerle el servicio. De las 7 únicamente 1 de las variables es de carácter numérico.

    a. **AGE:** Variable numérica continua, que nos indica la edad del cliente. Se encontró que la edad media de los clientes es de 38 años y los clientes más jóvenes tiene 17 años.

    b. **JOB:** Esta es una variable categórica que nos indica el tipo de trabajo que tiene el cliente, se tienen 13 categorías para esta variable:

        i. Admin (Puesto administrativo)

        ii. Blue-collar (Clase obrera)

        iii. Entrepreneur (Emprendimiento)

iv. Housemaid (Empleada de casa)

v. Management

vi. Retired (Retirado)

vii. Self-employed (Autónomo)

viii. Services (Servicios)

ix. Student (Estudiante)

x. Technician (Técnico)

xi. Unemployed (Se encuentra desempleado)

xii. Unkown (Desconocido)

El tipo de trabajo más frecuente es el de puesto administrativo.

c. **MARITAL:** Estado marital del cliente, esta es una variable categórica, que cuenta con 4 clases diferentes:

i. Divorced (Divorciado)

ii. Married (Casado)

iii. Single (Soltero)

iv. Unknown (Desconocido)

Se encontró que la clase más frecuente con un 60% de los datos, es la clase de clientes casados.

d. **EDUCATION:** Variable categórica que nos indica el nivel de escolaridad del cliente, se cuentan con 8 categorías diferentes:

i. Basic.4y (Cursaron 4 años de estudios básicos)

ii. Basic.6y (Cursaron 4años de estudios básicos)

iii. Basic.9y (Cursaron 9 años de estudios básicos)

iv. High.school (Completaron los estudios secundarios)

v. Illiterate (Analfabetas)

vi. Professional.course (Cursaron un curso profesional)

vii. University.degree (Completaron una licenciatura)

viii. Unkown (Desconocido)

e. **DEFAULT:** Variable categórica, que nos indica si el cliente tiene un crédito de mora, esta puede tomar los siguientes resultados:

   i. Yes (Si)

   ii. No

   iii. Unkown (Desconocido)

   Se encontró que el 80% de los clientes no cuentan con una mora.

f. **HOUSING:** Esta es una variable categórica que nos indica si el cliente cuenta con préstamo para vivienda. Esta puede tomar los siguientes valores:

   i. Yes (Si)

   ii. No

   iii. Unkown (Desconocido)

   Se encontró que el 52% de los clientes cuentan con un préstamo para vivienda.

g. **LOAN:** Variable categórica que nos indica si el cliente cuenta con algún préstamo personal, al igual que en los casos de las variables anteriores, este puede tomar los posibles valores de:

   i. Yes (Si)

   ii. No

   iii. Unkown (Desconocido)

B. Variables relacionadas con el último contacto de la campaña actual

a. **CONTACT:** Esta es una variable categórica, que nos indica el método de contacto de comunicación utilizado, que puede ser contacto al teléfono celular o al teléfono fijo del cliente. Siendo el más común al teléfono celular, presente en un 63% de los datos.

b. **MONTH:** Variable categórica, que nos indica el mes del año en el que se realizó el último contacto con el cliente. Se encontró que, en mayo, es el mes en el que más contactos se realizan, mientras que en los meses de enero y febrero no se realiza ningún contacto.

c. **DAY_OF_WEEK**: Variable categórica que nos indica el día de la semana en el que ocurrió el último contacto, únicamente se toman en cuenta los días laborables de lunes a viernes, y en promedio se hace la misma cantidad de contactos cada día, presentando cada día alrededor del 20% de los contactos semanales.

d. **DURATION:** Variable numérica, continua, que nos indica la duración del último contacto con el cliente en segundos. Este atributo afecta significativamente a los resultados y objetivos, sabemos que, si la duración es de 0 segundos, entonces la variable de predicción es "no". La duración no se sabe hasta que la llamada haya sido hecha, además después de que la llamada es terminada, la variable a predecir ya es obvia. Por lo que este parámetro solo debería de ser incluido como benchmarking y debería de ser descartado si la intención es hacer un modelo de predicción realista.

e. **CAMPAIGN**: Variable numérica, que nos indica el número de contactos realizados durante la campaña a este cliente. La media de contactos realizados es de 2 contactos por cliente.

f. **PREVIOUS:** Variable numérica, que nos indica el número de contactos realizados antes de esta campaña al cliente, se encuentra que el mayor número presentado de contactos realizados con anterioridad a esta campaña es de 7 contactos.

g. **POUTCOME:** Variable categórica que nos indica el resultado del contacto en las campañas de marketing anteriores, esta variable puede tomar los valores de:

   i. Éxito

   ii. Fracaso

   iii. No existe, en este caso es en el que no se contactó al cliente en la última campaña.

   Se encontró que a el 86% de los clientes no se les contacto en campañas anteriores.

C. Atributos de contexto social y económico

a. **EMPVARRATE:** Variable numérica que nos indica la tasa de variación del empleo al momento en el que se hizo el contacto, este es un indicador trimestral.

b. **CONS.PRICE.IDX**: Variable numérica que nos indica el índice de precios al consumidor, al momento que se contactó con el cliente, este es un indicador mensual.

c. **CONSCOFIDX**: Variable numérica que nos señala el índice de confianza del consumidor, que es un indicador mensual, y se nos muestra el valor de este índice al momento de realizar el contacto con el cliente.

d. **EURIBOR3M**: Variable numérica que nos muestra la tasa Euribor, índice de referencia que informa sobre los diferentes tipos de interés al que los bancos se prestan entre sí. Este es un índice a 3 meses, y la variable nos indica el valor de este índice al momento del contacto con el cliente.

e. **NREMPLOYED:** Variable numérica que nos indica el número de empleados, este es indicador trimestral y se encuentra que la media es de 5167.

Como se comentó el atributo duration afecta significativamente a los resultados por lo que este parámetro será descartado de la experimentación con la intención es hacer un modelo de predicción optimo. Por lo que solo se tomaran en cuenta las 19 variables restantes. Se puede encontrar que se cuentan con 10 variables categóricas en total (*Job, Marital, Education, Default, Housing, Loan, Contact, Month, Day_of_week y Poutcome*). Estos son datos importantes debido a que no todos los modelos trabajan con esta clase de variables.

Adicional a estas variables, los conjuntos de datos cuentan con la variable respuesta, que nos indica si el cliente se ha suscrito (“Éxito”) o no (“Fracaso”) al servicio del depósito. Es una variable binaria donde si el evento es un “Éxito” la variable toma el valor 2, y en el caso de que se clasifique como un “Fracaso”, esta toma el valor de 1.

Se encontró que en el conjunto de entrenamiento únicamente se cuenta con 4,640 instancias en las que el cliente se suscribe al servicio, mientras que en 36,548 instancias en las que el cliente decide no suscribirse. Por lo que podemos observar que los éxitos representan únicamente el 11% de las instancias.

4 Metodología

El objetivo de esta investigación es encontrar el modelo de aprendizaje que pueda discriminar de mejor manera aquellos clientes que potencialmente se suscribirán al servicio del préstamo por parte del banco, de aquellos que no, para poder enfocar las campañas en las que se ofrece el servicio a los interesados y así aumentar el número de "Éxitos "con la menor cantidad de contactos posible.

Para conseguir esto, se utilizarán las bases de datos que son brindadas por el banco. Se cuenta con un conjunto de datos de Entrenamiento, compuesto por 19 parámetros de entrada y la variable respuesta. Los conjuntos de datos inicialmente cuentan con 41,187 instancias, en el caso del conjunto de entrenamiento, mientras que el conjunto de prueba está compuesto de 4,119 instancias. A partir de estos conjuntos se generarán 4 agrupaciones:

A. Parámetros de entrenamiento: Agrupando las observaciones de los 19 parámetros de entrada del conjunto de entrenamiento

B. Variable respuesta del entrenamiento: Agrupando las observaciones de la variable de salida del conjunto de entrenamiento

C. Parámetros de test: Agrupando las observaciones de los 19 parámetros de entrada del conjunto de test

D. Variable respuesta de prueba: Agrupando las observaciones de la variable de salida del conjunto de test

Estos fueron utilizados durante toda la experimentación y la primera acción fue realizar un estudio a estos datos y hacer un preprocesamiento a estos

4.1 «Preprocesamiento de datos»

Se realizaron dos acciones principales durante el preprocesamiento, la primera, la evaluación de los conjuntos de datos para observar el comportamiento de estos, mientras que el segundo proceso será el de la estandarización de los datos de las variables numéricas que se tienen en la base de datos.

4.1.1 «Datos Desconocidos»

Durante el estudio de la base de datos, se encontró que la base de datos contaba con datos faltantes, estos se encuentran representados como "*Unkown*" en diferentes instancias.

Por lo que se realizara un estudio de estos datos para encontrar la forma óptima de tratarlos. Esto se hará con el objetivo de perder la menor cantidad de información posible de todas aquellas instancias que están etiquetadas como "Éxito", esto quiere decir que el cliente se suscribe al servicio ofrecido por el banco. Esto debido a que es información limitada en nuestra base de datos, ya que solo representa un 11% del conjunto total de instancias y esta es la información más relevante para el estudio, ya que es la clase objetivo en la investigación.

4.1.2 «Normalización de Datos»

La normalización de los datos numéricos es un proceso que se realiza debido a que muchos de estos parámetros se miden en diferentes escalas, y esto podrían terminar creando un sesgo y para evitar esto se realiza la normalización y colocar todos los valores en una escala común que conserve la diferencia en los rangos de valores que tenían estos valores.

A cada uno de los parámetros numéricos del conjunto de variables mencionados anteriormente se le estandarizaran los valores, utilizando la media y desviación estándar de cada parámetro de la matriz. Se estandarizan siguiendo la siguiente formula:

$$Z = \frac{X(i) - media(i)}{desviación\ estandar\ (i)}$$

Esto se hace con cada uno de los parámetros de entrada (i), donde X(i) es el valor de la instancia, del parámetro de entrada (i). Esto se realiza para normalizar todas las variables y que todas se encuentren tipificadas con media 0 y desviación típica 1. Esto se realizará tanto con el conjunto de datos de entrenamiento como en el de datos de prueba.

Por lo que durante la experimentación se utilizarán los conjuntos de datos con los parámetros de entrada de entrenamiento y de prueba ya estandarizados, para realizar el modelado con las diferentes técnicas de aprendizaje.

4.1.3 «Conjunto Balanceado»

El conjunto de datos con el que contamos para entrenar los diferentes modelos únicamente cuenta con un 11% de instancias donde el cliente ha aceptado el servicio ofrecido por el banco. Este es un número limitado de información comparada con la información acerca de clientes que rechazan el servicio, por lo que se puede decir que se cuenta con una muestra de datos desbalanceada que puede presentar un problema a los diferentes modelos, dificultando la capacidad para discriminar entre ambas clases.

Esto se evitará, limitando la cantidad de datos que se utilizan para generar los diferentes modelos, se buscara generar una muestra balanceada, donde se tomaran en cuenta todas las instancias de "Éxito" y se tomara la misma cantidad de instancias de manera aleatoria de observaciones que se consideren como "Fracasos".

Este conjunto de datos es el que se utilizara en los diferentes modelos, se generara una muestra para cada modelo, utilizando de esta manera una mayor cantidad de instancias de "fracaso", ya que se cuenta con un nivel alto de instancias para esta clase.

A partir de estas modificaciones elaboradas en la base de datos, se continuará con el proceso de generar y evaluar los diferentes modelos de aprendizaje.

4.2 «Proceso de Experimentación»

Con los conjuntos de datos modificados, se procedió a la generación de los diferentes modelos que se pondrán a prueba para encontrar el que mejor pueda discriminar entre ambos tipos de clientes, así como tener un enfoque en hallar el que logre identificar de forma más optima a los clientes que sí que se suscriben al servicio ofrecido.

Para hallar esto la experimentación se puede agrupar en 3 diferentes procesos:

4.2.1 «Generación de los modelos de clasificación utilizando todos los parámetros de entrada»

Durante esta experimentación se generarán 7 modelos diferentes de aprendizaje y se compararán para encontrar el modelo optimo que cumpla con nuestro objetivo. Los modelos que se utilizaran son los siguientes:

A. Análisis Discriminante Lineal (LDA)
B. K Vecinos Más Cercanos (KNN)
C. Naive Bayes
D. Máquinas de Vector Soporte (SVM)
E. Árbol de Decisión
F. Tree-Bagging
G. Bosques Aleatorios

Para generar estos modelos se utilizarán los 19 parámetros de entrada con los que se cuentan. En el caso de los modelos LDA y KNN, que funcionan únicamente utilizados parámetros numéricos, solo se utilizaran 10 de los 19 parámetros, ya que el resto son parámetros categóricos. Los parámetros que no se utilizaran son:

A. *Tipo de Trabajo*
B. *Estado Marital*
C. *¿Tiene crédito en mora?*
D. *¿Tiene préstamo para vivienda?*
E. *¿Tiene préstamos personales?*
F. *Tipo de contacto de comunicación utilizado*
G. *Mes del año en el que se realizó el último contacto*
H. *Dia de la semana en el que ocurrió el último contacto*
I. *Resultado en las campañas de marketing anteriores*.

En el caso del modelo KNN, que cuenta con hiperparámetro variantes como lo son la cantidad de vecinos (K), que se utilizan para clasificar las nuevas observaciones y el tipo de distancia que se utiliza. Se utilizarán algoritmos de Matlab, para encontrar el número de vecinos óptimo para nuestro modelo, tomando en cuenta que se utilizara la distancia euclídea, para medir la distancia entre el nuevo punto y los vecinos. Estos modelos buscan encontrar estos hiperparámetro minimizando el error a través de una validación cruzada de 5 iteraciones, esto quiere decir que mide la inexactitud que tiene el modelo, variando el número de vecinos que tiene y se escoge aquel con el cual tenga el valor mínimo de las iteraciones realizadas.

En el caso del modelo SVM, donde existen más hiperparámetros, se calcularán cuáles son sus valores óptimos, calculando el valor que debe de tomar C, que nos dará la flexibilidad del modelo, así como nos dará el kernel que usaremos. En el caso del kernel, se van variando pruebas entre el kernel gaussiano, linear y el polinomial, mientras que en el caso del hiperparámetro C, va variando entre los valores de $[[1e^{-3}, 1e^{3}]$. Al igual que en el caso del KNN, se hará a traves de encontrar la inexactitud o error que tienen y encontrar a través de una validación cruzada de 5 iteraciones, la combinación que obtenga el mínimo error.

En el caso del Naive Bayes, para encontrar el modelo optimo se hacen variaciones en la distribución que se utilizara, si se utiliza una normal, o por una estimación de la densidad de suavizado del kernel, así como también se hacen variaciones en el "*Width*" que es el ancho de la ventana de suavizado del kernel. Como en los casos anteriores, se hacen estas variaciones para encontrar el modelo que minimice el error en la clasificación a través de una validación cruzada de 5 iteraciones.

En el caso del modelo de árbol, al utilizar el código para obtener el modelo más optimo, adicionalmente nos ayuda a podar el árbol optimo seleccionado, esto lo hará utilizando la tecina de "*tamaño mínimas de hojas*", que lo que busca es tener el mínimo número de nodos hojas, por lo que en este caso se generan diferentes árboles y se ponen a prueba a través una validación cruzada de 5 iteraciones, buscando aquel en el que se obtenga el error mínimo.

Por último, en el caso del modelo de Tree-Bagging y Random Forest, se configuran para que cada uno genere 350 árboles, para entrenar el modelo y se especifica que al momento de hacer las 350 remuestras para generar los árboles, se permite el que una misma instancia se encuentre en más de una única remuestra.

Una vez generados todos los modelos, se ponen todos estos a prueba utilizando el conjunto de datos de prueba. En el caso del LDA y KNN, también solo se utilizan los parámetros numéricos. Una vez que todos los modelos se pusieron a prueba y realizaron la predicción, se utilizaran las métricas de clasificación binaria, para comparar los diferentes modelos. Se podrán ver la tasa de aciertos (Acc), la sensibilidad (Se), especificidad (Sp), entre otras.

Para completar la comparación entre estos modelos, se obtiene la curva ROC de cada uno de estos, para obtener el Área bajo la curva de cada uno de los modelos (AUC), que esto nos dará un valor general del comportamiento del modelo y que tan optimo es este.

Con estos datos se realizará una comparación entre el rendimiento de cada uno de los modelos para encontrar cual es el que mejor se adapta a nuestro objetivo. Se seleccionarán aquellos 2 que presentan mejores métricas, y que son óptimos para nuestro objetivo, Estos modelos seleccionados son los utilizados en otros pasos de la experimentación.

4.2.2 «Contraste de los modelos utilizando un proceso de selección de atributos y utilizando todas las instancias disponibles»

4.2.2.1 «Generación de los modelos utilizando un proceso selección de atributos»

En este caso se generarán 4 modelos, pero estos a diferencia de los obtenidos en la experimentación 4.2.1, estos tendrán un proceso de selección de atributos previo a la generación del modelo. La selección de atributos se hará a través de una técnica de filtro utilizando la prueba de chi-cuadrado, que se enfoca en medir la asociación entre dos variables. La hipótesis nula en esta prueba sugiere independencia entre estas variables. Entre más alto sean los valores obtenidos del estadístico, existe más evidencia para rechazar esta hipótesis nula, por lo que estas son las variables que son dependientes y tienen una influencia alta y formaran parte del subconjunto de parámetros seleccionados. A través de los datos del estadístico obtenidos para los 19 parámetros, se decidirá el número de parámetros óptimos a utilizar en los modelos.

Los 4 modelos que se obtendrán serán:

A. Árbol de Decisión
B. Máquinas de Vector Soporte (SVM)
C. Tree-Bagging
D. Random Forest

Estos seguirán los mismos parámetros utilizados en la experimentación 1, con la única diferencia que, al generar los modelos, estos utilizaran una menor cantidad de parámetros, utilizando solo los seleccionados como óptimos y que tienen una mayor relevancia sobre la variable dependiente.

Después de generar los modelos, se repetirá el mismo proceso que en la experimentación 1, utilizando cada uno de estos modelos obtenidos y para evaluar y medir que tan bueno es su rendimiento se obtienen las métricas de clasificación binaria de estos.

Posterior se obtendrá la curva ROC de estos modelos también, para así poder obtener el valor de AUC, que nos servirá para la comparación de los modelos.

Por último, estos resultados se unirán a los obtenidos por la experimentación 4.2.1 para hacer una comparación entre los modelos generados, esto se hace para ver si existe una mejora en los modelos al eliminar las variables menos relevantes y si este proceso de selección de atributos es necesario para nuestros modelos.

4.2.2.2 «Generación de los modelos utilizando todas las instancias disponibles»

En este caso se generan nuevamente 4 modelos de clasificación, a través de las mismas técnicas que en el inciso anterior (SVM, Árbol de Decisión, Tree-Bagging y Random Forest). Pero en esta ocasión no se hace una selección de atributos. La diferencia de estos modelos es que se generaran utilizando las 30.488 instancias con las que se cuentan.

Esto se realiza para constatar si utilizar todas las instancias, a pesar de ser una muestra desbalanceada y tener una mayor cantidad de información para generar los modelos, genera modelos con mejor rendimiento y que puedan ser mejores a los obtenidos por las experimentaciones anteriores.

Al igual que en los casos anteriores, posterior a generar estos modelos, se utilizarán con el conjunto de datos de prueba para obtener sus respectivas métricas. Al igual que se obtendrá la curva ROC de estos, para hacer comparaciones entre los modelos.

4.2.3 «Implementación del modelo final»

4.2.3.1 «Contraste del modelo utilizando el punto de umbral óptimo de la curva ROC»

Al encontrar aquellos 2 modelos óptimos y con mejor rendimiento para nuestro problema, se estudiará su curva ROC, y se buscara cual es el punto óptimo para el umbral de corte, esto quiere decir, cambiar el punto en el que se decide cuando una observación se clasifica como “Éxito” o

“Fracaso”, normalmente Matlab, utiliza el punto 0.5 que es el punto medio, para hacer la división entre ambas clases.

El punto óptimo, se obtiene del punto de la curva que es el más cercano al valor (0,1), que es el punto que se considera ideal en la curva ROC, aquel donde se presenta una sensibilidad del 100%. Por lo que se obtendrá este valor de los 2 modelos seleccionados y se modificará el punto de corte de ambos.

Por lo que, al obtenerlo, se utilizaran los scores de las predicciones, esto quiere decir que se utilizaran la probabilidad que tiene cada instancia para pertenecer a cada una de las clases, en este caso nos interesa el de la clase positiva, por lo que, en este caso, se modificara, para que cuando el valor del score de una instancia sea mayor o igual al del umbral optimo obtenido de la curva ROC, la instancia sea clasificada como “Éxito”. Modificando este punto de umbral, y obteniendo las predicciones con este, se hace otra vez una comparación con los valores reales de los valores de prueba y se obtiene las métricas de clasificación binarias, para compararlas con los modelos antes de este cambio y encontrar entre estos cual es el óptimo.

4.2.3.2 «Contraste del modelo modificando el punto de umbral en base a la sensibilidad para distintas sensibilidades objetivo»

Se hará una experimentación similar a la del paso anterior modificando el valor del umbral de corte, pero ahora no se utilizará el punto óptimo. Esto debido a que el punto óptimo se encarga de buscar el punto donde mejore de manera general el modelo a través de su tasa de aciertos, pero este no siempre conlleva la mejor sensibilidad, que es una métrica que nos interesa ya que esta nos indica la tasa de observaciones positivas clasificadas correctamente. Nosotros al busca un modelo que de mejor manera discrimine las clases pero que al mismo tiempo nos ayude a identificar la mayor cantidad de “Éxitos” posibles de manera correcta. Por lo que se obtendrán y evaluaran 4 puntos de cortes diferentes, para aumentar la sensibilidad del modelo, y compararlos con todas sus métricas, ya que, al aumentar la sensibilidad, podemos estar comprometiendo otros parámetros aumentando la cantidad de falsos positivos, y se debe evaluar hasta qué punto esto es conveniente.

Para esto se repetirán los pasos del proceso 4.2.3, 4 veces, cambiando el umbral a estos parámetros:

A. Punto de umbral donde se obtiene un 90% de sensibilidad.
B. Punto de umbral donde se obtiene un 95% de sensibilidad.

C. Punto de umbral donde se obtiene un 97% de sensibilidad.

D. Punto de umbral donde se obtiene un 99% de sensibilidad.

Al encontrar las métricas con cada uno de estos se puede tomar una decisión y encontrar aquel modelo que más se ajuste a nuestro objetivo y pueda representar una mejor solución para el banco.

5 Pruebas y Resultados

5.1 «Preprocesamiento de datos»

A través del estudio realizado por medio de RStudio se encontraron que en el conjunto de datos existían instancias que contenían valores vacíos. Los siguientes parámetros son los que contaban con datos faltantes en el conjunto de entrenamiento:

Tabla 01. Parámetros con valores faltantes en el conjunto de entrenamiento

Parámetro	N.º De Instancias Con Valores Faltantes
Default	8,597 instancias
Education	1,731 instancias
Housing	990 instancias
Loan	900 instancias
Job	330 instancias
Marital	80 instancias

Con todos estos datos, se encontró que, en 10,700 instancias del conjunto de entrenamiento, existen datos faltantes, lo que equivale a un 25% de las bases de datos. Se observo como afectaría el descartar estas instancias, debido al bajo nivel de instancias de "Éxitos" que hay en el conjunto de entrenamiento, se busca minimizar la perdida de información de estas instancias, por lo que se observó que porcentaje de estas instancias se perdería si se descartan las instancias con valores desconocidos. Se encontró que únicamente se pierde el 16% de las instancias, conservando un 84%, es decir 3.859 de los 4.640 éxitos.

Mientras que, en el caso de las instancias clasificadas como "Fracaso", se encontró que se conservaría un 72% de los datos, por lo que este presenta una mayor pérdida de información con un 28% de información perdida. Debido a esto y encontrar que no existe una alta perdida de información de clientes que se suscriben al servicio, que son los datos de los cuales nos interesan tener la mayor información posible. Así como hay suficientes instancias de fracaso, que el descarte

de las instancias con datos perdidos no reduciría de manera significativa la información de esta clase. Se optó por descartar las instancias que presentaban algún dato faltante. Por lo que el conjunto de entrenamiento con el que se trabajara cuenta únicamente con 30.488 observaciones, de las cuales 3.859, son instancias de clientes que se suscriben al servicio, mientras que 26.629 instancias son de clientes que no se suscriben al servicio.

De la misma forma se estudió el conjunto de prueba, que presenta 4.119 instancias, por lo que es un conjunto de pruebas que equivale al 10% de los datos utilizados de entrenamiento. En este caso se encontró también con instancias que presentaban datos faltantes:

Tabla 02. Parámetros con valores faltantes en el conjunto de prueba

Parámetro	N.º De Instancias Con Valores Faltantes
Default	803 instancias
Education	167 instancias
Housing	105 instancias
Loan	105 instancias
Job	39 instancias
Marital	11 instancias

En total se encontró que en 1.029 de las 4.119 instancias se presentaban datos faltantes, por lo que, si se descartan, se perdería el 25% de los datos. Pudiendo utilizar únicamente 3.090 instancias. Se observo nuevamente si este descarte de instancias afectaría y reduciría drásticamente el número de instancias de clientes que se suscriben al servicio ofrecido en la campaña.

Se encontró que originalmente el conjunto de prueba cuenta con 451 instancias de “Éxitos” y 3.668 instancias de “Fracaso”. Al momento de hacer el descarte se pierde únicamente el 18% de las instancias de “Éxito” conservando 370 de estas. Mientras que en el caso de las instancias de “Fracaso” se encontró que se conserva únicamente un 61% de sus instancias. A pesar de que en las instancias de “Fracaso” hubiera un descenso alto de instancias, al contener un alto nivel de información en comparación de las instancias de clientes que, si se suscriben al servicio, se optó

también por realizar un descarte de estas variables, por lo que ahora el conjunto de prueba contiene 3.090 instancias.

Para dar soporte a esta decisión, se hicieron evaluaciones previamente en la que se utilizaron los conjuntos de datos de entrenamiento y prueba con datos faltantes para generar un modelo y ver si este presentaba un buen rendimiento en comparación de si lo hacía con los mismos conjuntos, pero eliminando las instancias con datos faltantes, encontrando que esta segunda opción presentaba mejores métricas y modelos óptimos.

Posterior a estos descartes y obtener los nuevos conjuntos de datos de entrenamiento y de prueba, se estandarizaron los datos de los parámetros numéricos de estos, a través de la función “scale” de RStudio.

Antes de iniciar cada uno de los procesos para generar los modelos de aprendizaje, se generará una muestra balanceada de datos, en este caso al haber reducido el conjunto de entrenamiento, ahora la muestra balanceada se hará con todos los datos etiquetados como “Éxitos” (3.859) y se utiliza la misma cantidad de instancias de clientes que no se suscriben al servicio ofrecido. Por lo que nuestros conjuntos de entrenamiento contendrán 7.718 instancias. Durante la experimentación se utilizará el termino *ME*, para identificar los modelos que hayan utilizado esta muestra balanceada.

5.2 «Generación de los modelos de clasificación utilizando todos los parámetros de entrada»

5.2.1 «Análisis Discriminante Lineal»

En el caso del análisis discriminante lineal, se utilizaron los parámetros numéricos, para obtener el modelo que divide ambas clases.

A partir de este modelo se pueden obtener los siguientes resultados al ponerlo a prueba con el conjunto de prueba, en este caso también solo se utilizan los 9 parámetros numéricos.

Tabla 03. Métricas de Clasificación Binaria del modelo LDA con todos los atributos

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.72362	0.27638	0.73514	0.26486	0.72206	0.27794	0.26459	0.38913

Podemos observar que tiene una sensibilidad 0.73, por lo que podemos observar que el modelo presenta un valor alto como tasa de clientes que se clasifican como que se suscriben al servicio y que en realidad si se suscriben a este. Pero podemos observar que tiene una precisión baja de 0.26 por lo que tiene una tasa baja de clientes predichos como "Éxitos" que fueron clasificados correctamente.

A partir de estas predicciones se obtiene la probabilidad a posteriori de pertenencia a cada una de las clases, para cada instancia, y se obtuvo la curva ROC. De la cual obtenemos que tiene un área bajo la curva de 0.77, este nos servirá para medir exactitud global del modelo, al igual que podemos encontrar que su punto óptimo de umbral se encuentra en el punto (0.0118, 0.1865), que podemos encontrar que se encuentra alejado del punto óptimo de la curva ROC (0,1).

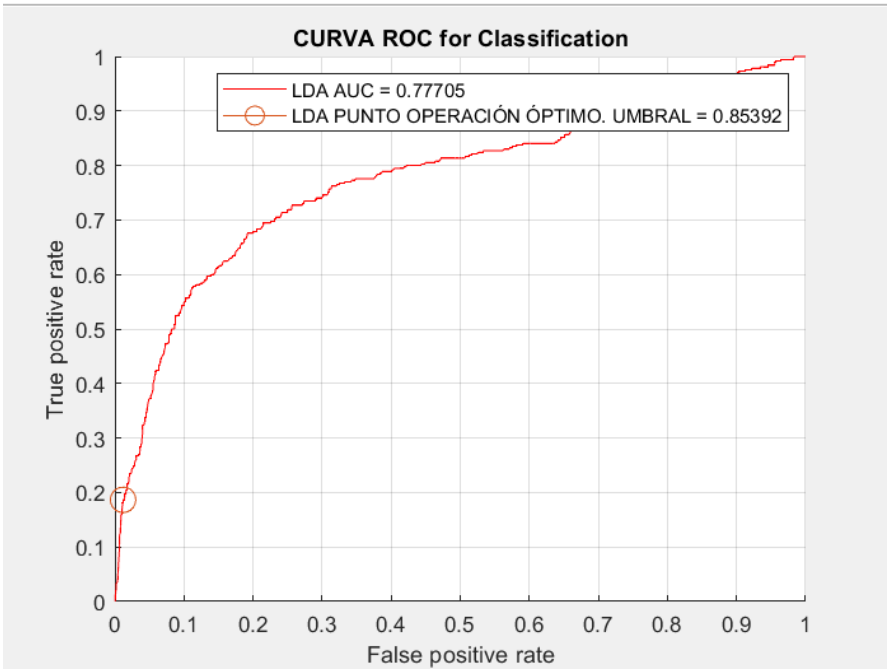


Ilustración 7. Curva ROC del modelo LDA, utilizando todos los atributos

5.2.2 «K Vecinos más cercanos»

Al generar el modelo óptimo de K vecinos más cercanos, se utilizaron también únicamente las 9 variables numéricas con los que se cuenta, en este caso se buscó por medio Matlab el valor optimo del hiperparámetro K utilizando una distancia euclídea, y este es de 30 vecinos (K=30), encontrando que con este se producía el mínimo error.

A partir de este modelo generado, se utiliza junto con el conjunto de datos de prueba y realizar una predicción a la que pertenece cada instancia, para compararlo con los datos reales, y se obtuvieron las siguientes métricas para observar el rendimiento que se obtuvo

Tabla 04. Métricas de Clasificación Binaria del modelo KNN con todos los atributos

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.81909	0.18091	0.66486	0.33514	0.84007	0.15993	0.36123	0.46813

Se puede observar que se tiene una tasa de falsos negativos del 33% por lo que existe una cantidad significativa de valores en los cuales, se clasifican de manera errónea, considerándolos como clientes que no se suscriben al servicio, cuando en realidad lo hacía y esto es importante de considerar, ya que nuestro objetivo es asegurarnos de clasificar correctamente la mayor cantidad de “Éxitos” posibles, mientras que en el caso de la tasa de falsos positivos se tiene una tasa de 16%, por lo que podemos decir que el modelo es más exigente para clasificar a una instancia como ”Éxito”. En comparación a la clase de “Fracaso”.

Adicionalmente se realizó una variación en este modelo en el cual, a través de un método de *Leave One Out*, se buscó el valor K que presentaba la mayor tasa de aciertos, haciendo iteraciones y variado el valor de K de 1 a 5 con saltos de una unidad. Encontrando que el K con mayor tasa de aciertos se encuentra con un K=18. Se utilizo este valor en el modelo, y se utilizó este modelo con el conjunto de prueba obteniendo los siguientes resultados:

Tabla 05. Métricas de Clasificación Binaria del modelo KNN con todos los atributos, utilizando un K=18, hallado por el método de LOO

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.81553	0.18447	0.66216	0.33784	0.8364	0.1636	0.35507	0.46226

Se puede observar que son resultados muy similares, variando insignificantemente, por lo que se utilizara este modelo en el resto de la experimentación.

Por último, se obtuvo su curva ROC, para obtener la métrica del AUC. Se encontró que el AUC posee un valor de 0.801. Como se puede observar en la gráfica, se encuentra que el punto óptimo en (0.0210, 0.2892), que también se encuentra lejano al optimo (0,1), se puede comprobar en este punto que se tendrá un modelo exigente para clasificar los "Éxitos" en nuestros datos, ya que se contará con una tasa de falsos negativos muy baja, pero conllevando también una tasa de verdaderos positivos también baja, menor al 30%.

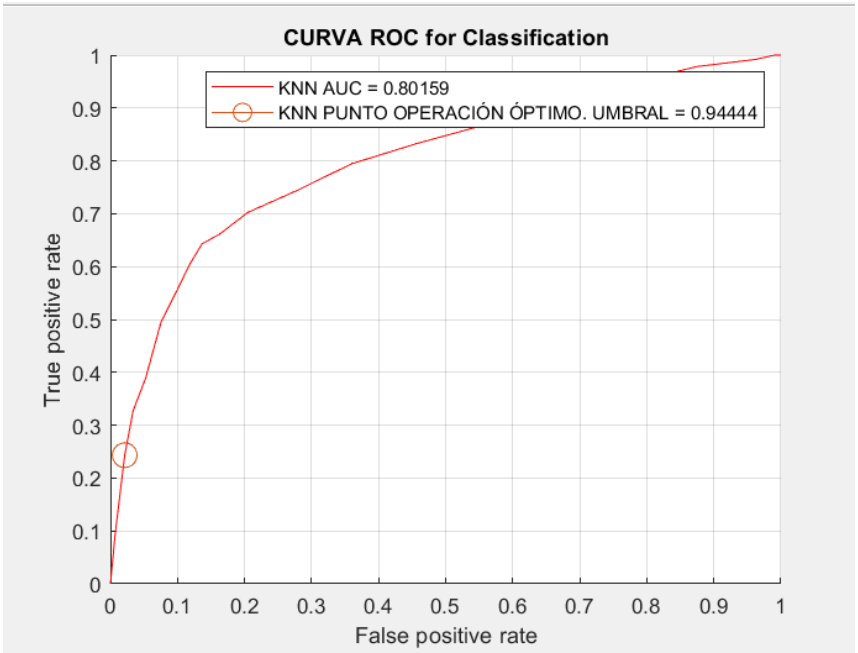


Ilustración 8. Curva ROC modelo KNN, utilizando todos los atributos

5.2.3 «Naive Bayes»

En el caso del modelo Naive Bayes, al momento de encontrar el modelo optimo, se optimizaron diferentes valores, como lo eran la distribución que se utilizara una distribución normal, con un ancho de la ventana de suavizado del kernel, del $9.79e^{-04}$.

A partir de este modelo generado se realizaron predicciones utilizando el conjunto de prueba, para evaluar que tan bueno es el rendimiento de este modelo. Se obtuvieron los siguientes resultados en las métricas:

Tabla 06. Métricas de Clasificación Binaria del modelo Naive Bayes con todos los atributos

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.83625	0.16375	0.33243	0.66757	0.90478	0.095221	0.32199	0.32713

Se puede observar que se tienen una sensibilidad muy baja, por lo que se tiene una tasa de acierto de la clase positiva muy baja, solo acertando en un 30% de todos los clientes que se suscriben al servicio. Esto acompañado de una tasa de falsos positivos, se puede deducir que se el modelo es muy exigente para clasificar un valor como "Éxito", ya que podemos observar que los valores que clasifica como positivos, en su mayoría son correctos y hay muy pocos datos en los que el modelo clasifica una instancia como "Éxito" y en realidad era el dato de un cliente que no se suscribía a el servicio ofrecido.

Se obtuvo la siguiente curva ROC:

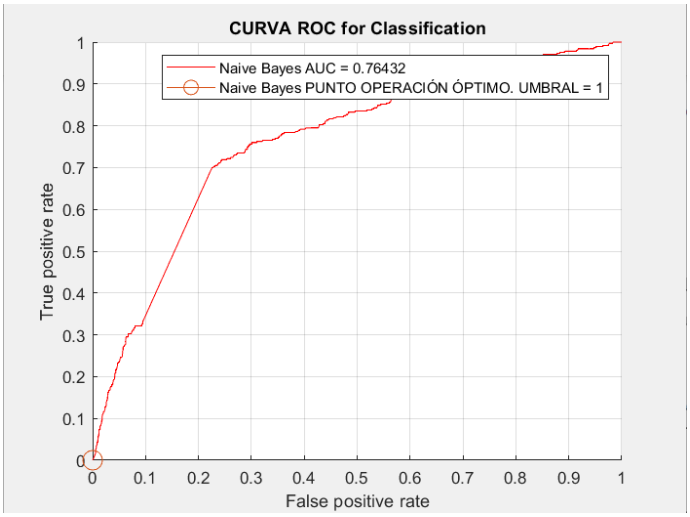


Ilustración 9. Curva ROC del modelo Naive Bayes, utilizando todos los atributos

Se puede observar que esta curva se aleja de lo ideal, estando muy lejana al punto (0,1), se encuentra que el punto óptimo para el modelo está en el punto (0,0), tomando el valor de 1, podemos observar que en este modelo no es óptimo, tomando este optimo se clasificaran todas las instancias como "Fracasos", ya que es donde se obtendrá la mayor tasa de aciertos, pero esto se debe a que se acertarían un 90% de las instancias, por la baja tasa de "Éxitos" que se tiene en el conjunto de prueba. Así como se encontró que el AUC, tiene un valor de 0.764.

5.2.4 «Máquinas de Vector Soporte»

El modelo SVM al igual que el modelo KNN, cuentan con diferentes hiperparámetros que se deben de definir a la hora de generar un modelo. Los hiperparámetros que se optimizaron por medio de algoritmos de Matlab, fueron el tipo de kernel que usara el modelo, y el valor del hiperparámetro C que dictara la flexibilidad del modelo. En este caso posterior a la prueba de varios valores por la validación cruzada para encontrar el mínimo error, se encontró que se utilizaría un kenel Lineal, con un valor de C de 0.0016.

A partir de este modelo y utilizarlo con el conjunto de prueba se obtuvieron los siguientes resultados en las métricas de clasificación binaria:

Tabla 07. Métricas de Clasificación Binaria del modelo SVM con todos los atributos

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.82686	0.17314	0.63514	0.36486	0.85294	0.14706	0.37008	0.46766

Encontrando una sensibilidad del 63%, por lo que podemos observar que este modelo identifica correctamente más de la mitad de los clientes que se suscriben al servicio, y mantiene una tasa de falsos positivos no muy alta, del 15%.

Adicionalmente se obtuvo la curva ROC, que se muestra a continuación:

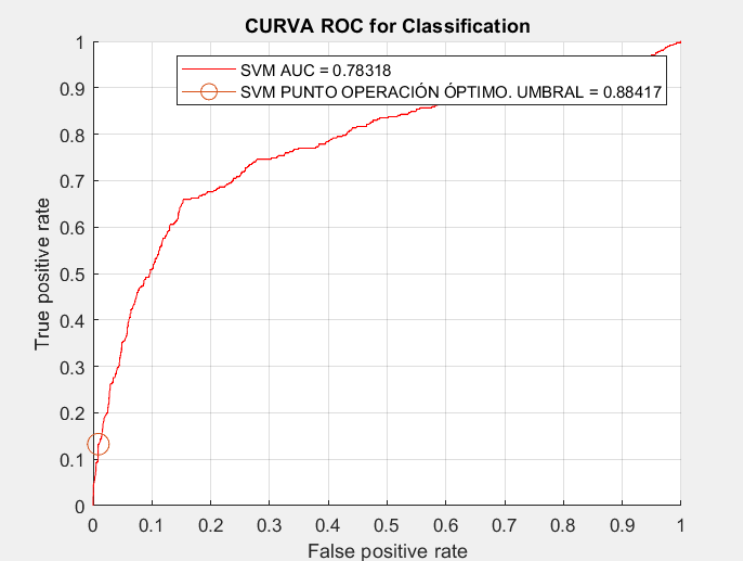


Ilustración 10. Curva ROC del modelo SVM, utilizando todos los atributos

Encontrando un valor de área bajo la curva (AUC) del 0.78, por lo que podemos observar que el modelo cuenta de manera global una buena exactitud caso aproximadamente del 80%. Aunque se puede seguir observando que el punto óptimo, se encuentra alejado del punto (0, 1).

5.2.5 «Árbol de Decisión»

En el caso del modelo del árbol de decisión se utilizó un algoritmo para hallar cual era el árbol optimo, ya podado, que era aquel que presentaba el menor error. Encontrando el siguiente árbol:

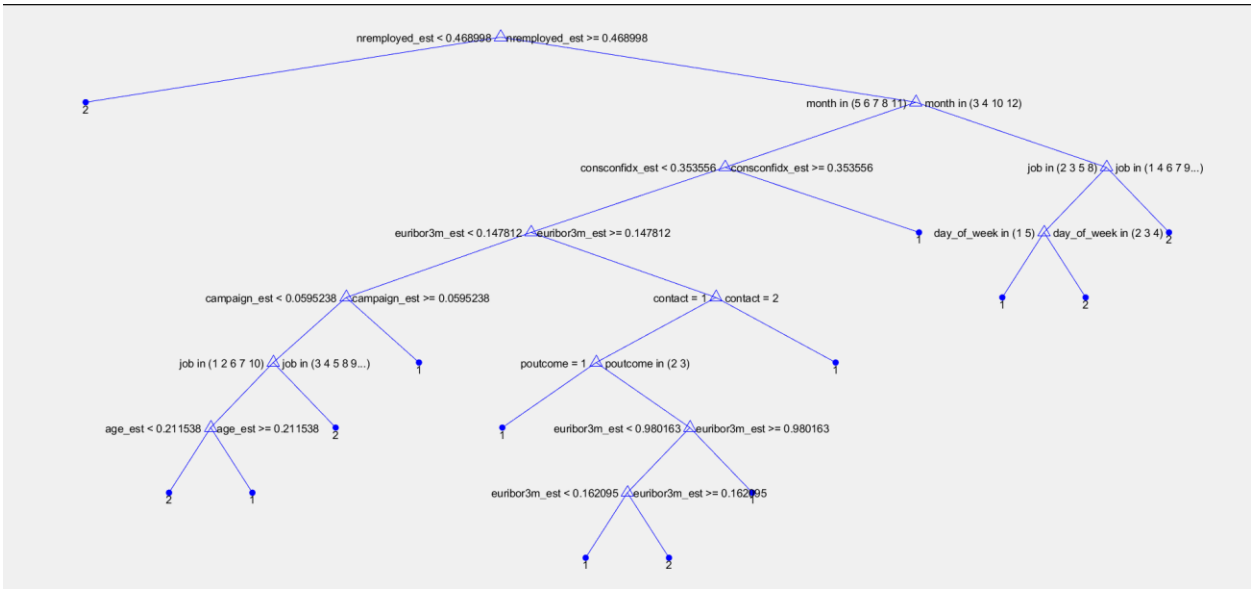


Ilustración 11. Árbol de decisión optimo utilizando todos los atributos[1]

Utilizando este árbol optimo con los datos de prueba, se pudo obtener las siguientes métricas al comparar las predicciones con los datos reales:

Tabla 08. Métricas de Clasificación Binaria del modelo de Árbol de Decisión con todos los atributos

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.80712	0.19288	0.70541	0.29459	0.82096	0.17904	0.34893	0.46691

[1] Ilustración 11, agregada como apéndice 1

Se puede observar que utilizar este árbol simplificado, se puede obtener en general unos resultados decentes, teniendo una sensibilidad del 70% por lo que acierta en la clasificación de una buena cantidad de datos. Pero todavía se fallan en casi un tercio de los éxitos que tenemos, y que son los datos que queremos identificar de manera más optima.

Adicionalmente se obtuvo la curva ROC, con este modelo, que presentaba los siguientes valores:

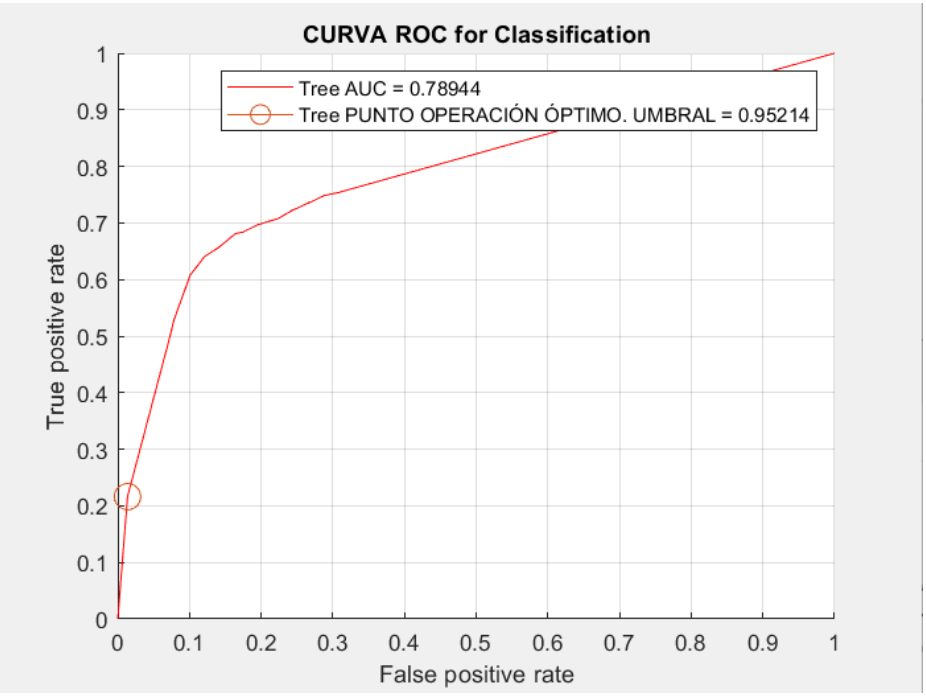


Ilustración 12. Curva ROC del modelo de Árbol de decisión, utilizando todos los atributos

Donde se puede observar un AUC del aproximadamente 0.79. Pero también podemos observar que se encuentra que el punto de umbral optimo es 0.95, este puede considerarse que es un valor muy alto, esto generara que se tenga una cantidad muy baja de falsos positivos, por lo que las que se predigan como clientes que se suscriben al servicio, serán correctas, pero esto conlleva en las predicciones que muy pocas instancias sean encontradas, por lo que se estaría volviendo un modelo muy exigente para clasificar los "Éxitos" acertando únicamente en el 20% de estos, que es clase en la que buscamos identificar y predecir la mayor cantidad posible.

5.2.6 «Tree-Bagging»

Para la generación del modelo Tree-Bagging, se generaron 350 árboles de decisión para poder encontrar mejores predicciones y reducir el error que se tiene al utilizar únicamente 1 árbol de

decisión. A partir de este modelo, se pudieron obtener los siguientes resultados, tras haberlo utilizado con los datos de prueba:

Tabla 09. Métricas de Clasificación Binaria del modelo de Tree-Bagging con todos los atributos

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.84854	0.15146	0.93784	0.062162	0.8364	0.1636	0.43813	0.59725

Se puede observar a través de estos, que este es un buen modelo para discriminar y clasificar a los clientes que tienen mayor potencial de suscribirse al servicio, esto se ve reflejado al presentar una sensibilidad del 93% que es una sensibilidad muy alta. Acompañado por una especificidad alta también, del 0.83, por lo que se tiene una tasa de falsos positivos del 16%, pero se mantiene siendo una tasa alta de acierto para la clase negativa.

Adicionalmente se obtuvo la curva ROC de este modelo

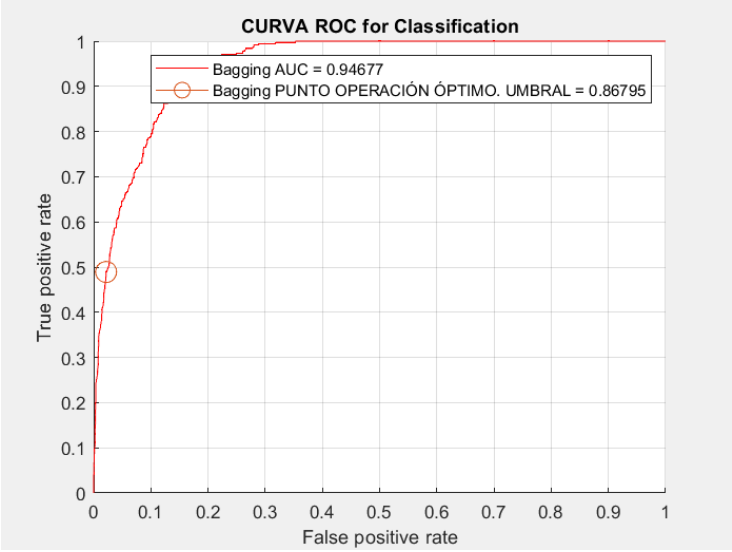


Ilustración 13. Curva ROC del modelo de Tree-Bagging, utilizando todos los atributos

Se puede observar un AUC muy bueno, teniendo un valor de 0,96, siendo el ideal de 1. Adicionalmente podemos encontrar que tiene un punto de umbral optimo en el punto (0,0217, 0,4892) que podemos observar que se puede tener una mejor predicción en la que se puede clasificar de manera correcta al 50% de los clientes que se suscriben al servicio, teniendo una tasa

mínima y casi 0 de falsos positivos, por lo que, al identificar una instancia como éxito, es muy probable que esta la sea.

5.2.7 «Random Forest»

En el caso de los bosques aleatorios, al igual que en el modelo de Tree-Bagging se generaron 350 remuestras para generar los árboles de este modelo y poder minimizar la variación y que se tiene utilizando únicamente un solo árbol de decisión y darles oportunidad a todos los parámetros para evitar sesgos de parámetros muy importantes. A partir de este modelo generado, se obtuvieron como resultados posteriores a la predicción con los valores de prueba:

Tabla 10. Métricas de Clasificación Binaria del modelo de Random Forest con todos los atributos

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.85502	0.14498	0.94054	0.059459	0.84338	0.15662	0.44961	0.60839

Junto con los datos de Tree-Bagging se encontró, que presentan la mayor tasa de sensibilidad presentando un valor del 94%, por lo que se puede observar que clasifica y acierta de manera óptima a aquellos clientes que tienen la mayor probabilidad de suscribirse al servicio del banco. Esto se puede acompañar con un F-Score alto que nos da una muestra general de que tan bueno es el modelo para clasificar la clase positiva y obtiene un valor 0.60.

Adicionalmente se obtiene la curva ROC de este modelo, que se presenta a continuación:

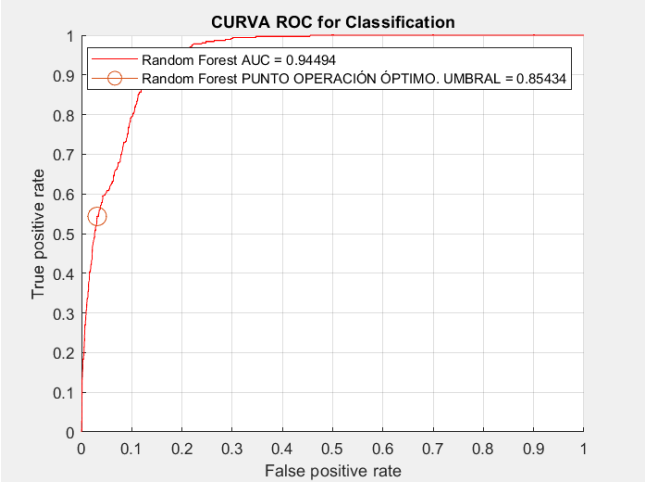


Ilustración 14. Curva ROC del modelo de Random Forest, utilizando todos los atributos

Se puede encontrar un AUC alto con un valor de 0.94, que nos demuestra que el modelo tiene una muy buena exactitud, siendo el ideal de 1. Así como podemos observar que se acerca más a una curva ROC ideal, llegando al punto (0,1), en este caso el punto más cerca es (0.0313, 0.5432). Por lo que en este punto se puede observar que se puede clasificar correctamente más de la mitad de los clientes que se suscriben al servicio, y que tiene una tasa de falsos positivos casi 0, por lo que al predecir que pertenece a esta clase, es casi seguro de acertar.

Posterior a la obtención de todos estos modelos, ahora se observará que sucede si se utiliza un proceso de selección de atributos, para encontrar si mejora el rendimiento de estos modelos, para posteriormente comparar los resultados obtenidos por todos.

5.2.8 «Comparación de los Resultados»

Tabla 11. Comparación de los métodos 7 métodos generados con todos los atributos

	TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore	AUC
LDA	0.72362	0.27638	0.73514	0.26486	0.72206	0.27794	0.26459	0.38913	0.77705
KNN	0.81553	0.18447	0.66216	0.33784	0.8364	0.1636	0.35507	0.46226	0.80159
Naive Bayes	0.83625	0.16375	0.33243	0.66757	0.90478	0.095221	0.32199	0.32713	0.76432
SVM	0.82686	0.17314	0.63514	0.36486	0.85294	0.14706	0.37008	0.46766	0.78318
Arbol de Decisión	0.80712	0.19288	0.70541	0.29459	0.82096	0.17904	0.34893	0.46691	0.79089
Bagging	0.84854	0.15146	0.93784	0.062162	0.8364	0.1636	0.43813	0.59725	0.94677
Random Forest	0.85502	0.14498	0.94054	0.059459	0.84338	0.15662	0.44961	0.60839	0.94494

Se puede observar a través de la tabla, que los 2 modelos que de manera general presentan las mejores métricas, son los métodos de Tree-Bagging y de Random Forest, presentando no únicamente el AUC más alto siendo aproximadamente del 94%, que nos indica la exactitud global del modelo, teniendo un muy buen valor comparado con el resto, siendo el siguiente más alto de 80% del KNN. Adicionalmente cuentan con un alto nivel de sensibilidad en estos modelos, presentando también una tasa del 94%, lo que nos indica que estos modelos son eficientes al momento de discriminar a los clientes que, si se suscriben al servicio del depósito ofrecido, por lo que en base a la exactitud global que estos 2 modelos (Random Forest y Tree-Bagging) son los seleccionados para utilizar posteriormente en la experimentación.

5.3 «Contraste de los modelos utilizando un proceso de selección de atributos y utilizando todas las instancias disponibles»

5.3.1 «Generación de los modelos utilizando un proceso Selección de Atributos»

En este caso se utilizó una selección de atributos por medio de la prueba de chi-cuadrado, donde se obtuvieron las puntuaciones de que tan relevantes son estos, en la siguiente grafica se puede resumir los datos hallados.

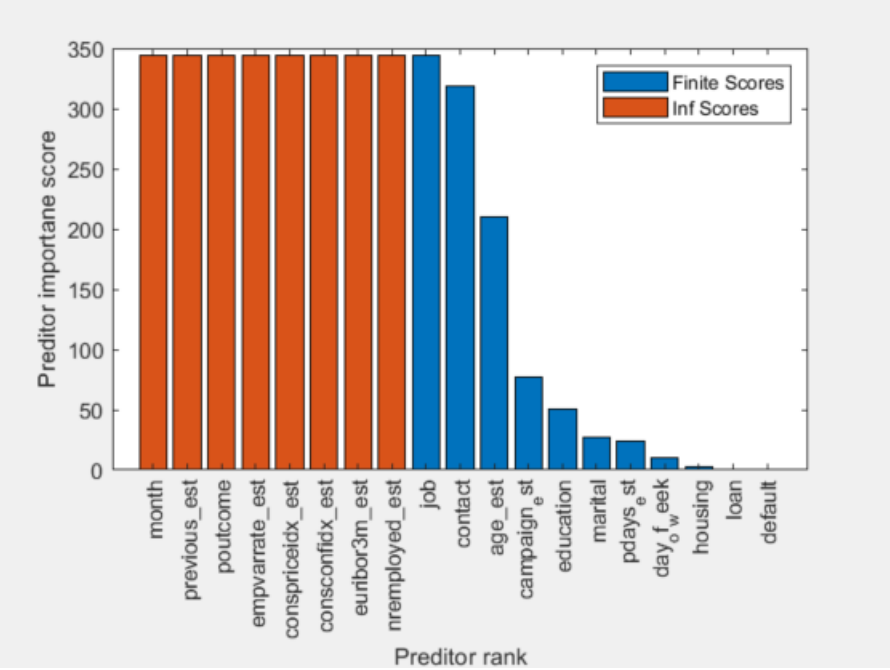


Ilustración 15. Ranking de predictores utilizando el test de chi-cuadrado

Se encontró que los primeras 8 parámetros, que presentan la relevancia más alta, se muestran como "Inf Scores" en la gráfica anterior, esto se debe a que al calcular el valor que se obtiene de la prueba, para ranquearlos, Matlab lo identifica como Inf ("infinito"), esto se debe a que en el test de chi-cuadrado, entre más pequeño sea el valor p obtenido, se indica mayor relevancia y dependencia de la variable dependiente, pero las valoraciones que obtiene Matlab para ordenar el ranking utiliza el cálculo de -log (valor p), y para los parámetros que presentan un valor p, muy pequeño, menor a eps(0), Matlab lo clasifica como infinito, identificando estos parámetros como los más relevantes para el modelo.

En este caso se encontraron que los parámetros que presentaban este caso eran 8:

A. Month
B. Previos
C. Pout_come
D. Empvarrate
E. Conspriceidx
F. Conconfidx
G. Euribor3m
H. Nremployed

A través de esta gráfica, se decidió tomar los 12 parámetros con mayor relevancia, ya que presentaban una mayor influencia en nuestro modelo según esta prueba y adicionalmente se puede observar que, a partir de este parámetro, existe un descenso considerable en el nivel de relevancia de los parámetros.

Por lo que el conjunto de datos de prueba solo se utilizaron los parámetros de *month, previous, poutcome, empvarrate, conspriceidx, consconfidx, euribor3m, nremployed, job, contact, age, compact.* Estos se utilizaron para construir los siguientes modelos.

Se puede observar y recalcar de los parámetros seleccionados, que la mayoría de estos parámetros son parámetros numéricos, siendo *month, job, poutcome y contact* las únicas categorías categóricas.

Adicionalmente podemos observar que de los parámetros numéricos que tenían mayor relevancia, se encuentran todos los parámetros que nos brindan información del contexto social y macroeconómicos, así que estos tienen una alta relación e influencia en lo que puede decidir un usuario. Esto se debe a que los usuarios al estar en una crisis económica, crea incertidumbre en las personas, y al generar esta, los clientes y usuarios tienden a preferir a mantener el efectivo o dinero en lugar de ingresarlo a una entidad bancaria, ya que, durante estas recesiones, necesitan del dinero y sienten más seguridad al tenerlo presente, en lugar de tenerlo guardado en un lugar independiente. Esta también puede ser una de las razones de porque hay una tasa tan baja de aceptación al servicio del depósito a largo plazo ofrecido por el banco.

Una vez determinado el vector de predictores óptimos, se continuó con la generación de los modelos.

5.3.1.1 «Máquinas de Vector Soporte»

Al igual que en la experimentación anterior se generó un modelo optimo utilizando ahora únicamente los parámetros más relevantes. En este caso se continúa utilizando un kernel lineal, así como ahora se utiliza un valor de C de 0.68

A partir de este modelo, y utilizarlo con el conjunto de datos de prueba para encontrar ver el rendimiento de este, se obtuvieron los siguientes resultados de métricas de clasificación binaria.

Tabla 12. Métricas de Clasificación Binaria del modelo SVM con selección de atributos por filtro

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.82686	0.17314	0.63514	0.36486	0.85294	0.14706	0.37008	0.46766

Donde se puede observar una tasa de acierto del 0.82, pero podemos ver que viene acompañado de una sensibilidad de 0,63 y una especificidad del 0,85. Por lo que podemos observar en ambos tienen unos buenos niveles de predicción tanto de clase positiva como de clase negativa. Por lo que podemos observar que se mantiene una buena tasa de aciertos, pero todavía se cuenta con una tasa de falsos negativos considerable del 40%.

En este caso se obtuvo también la curva ROC de este modelo

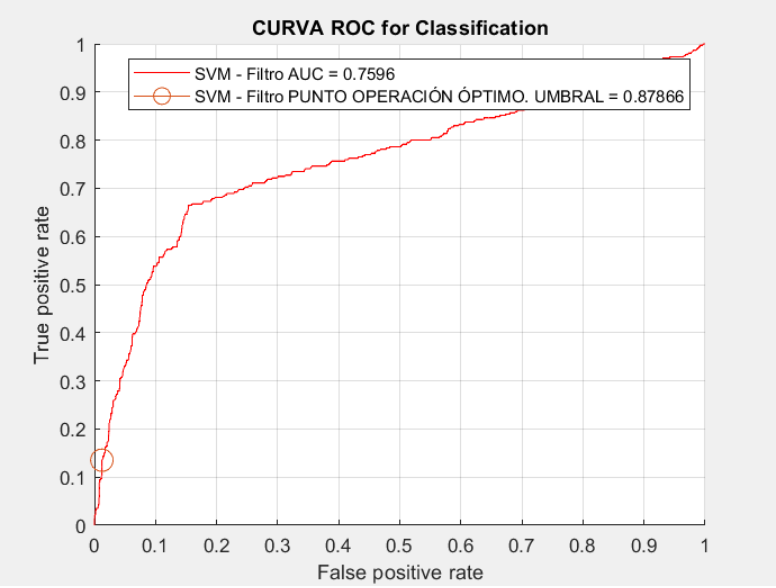


Ilustración 16. Curva ROC del modelo SVM, utilizando selección de atributos

En este caso se puede observar que se tiene un AUC de 0.75, por lo que no mantiene una exactitud global baja, y se puede observar que se encuentra alejado del punto óptimo de la curva ROC (0,1), siendo el más próximo a este es el punto (0.0121, 0.1351), donde puede presentar una mayor tasa de aciertos, pero esto no conlleva una buena tasa de sensibilidad ya que podemos observar que solo clasificaría y encontraría alrededor del 15% de los datos etiquetados como “Éxito”.

5.3.1.2 «Árbol de Decisión»

En este caso se volvió a generar un árbol podado optimo, pero ahora utilizando únicamente los 12 parámetros seleccionados. El árbol resultante es el siguiente:

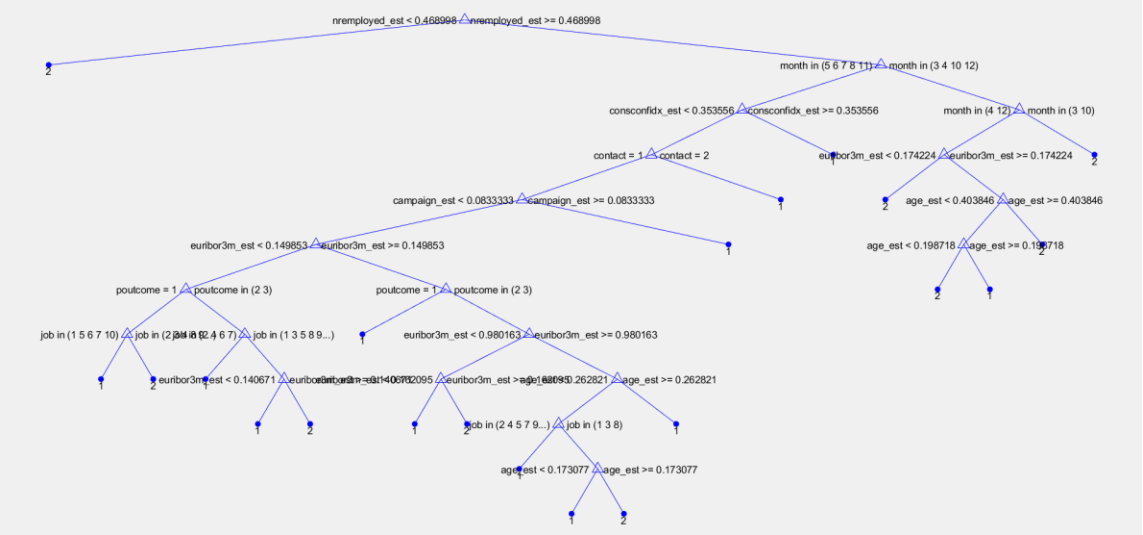


Ilustración 17. Árbol de decisión optimo, utilizando selección de atributos[2]

Donde se muestra un árbol más complejo que en el caso anterior, donde se utilizaban todos los parámetros.

En este caso, se pudo obtener el siguiente resultado al utilizar este árbol para predecir los datos del conjunto de prueba.

Tabla 13. Métricas de Clasificación Binaria del modelo de Árbol de Decisión con selección de atributos por filtro

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.811	0.189	0.68108	0.31892	0.82868	0.17132	0.35097	0.46324

En este caso se puede observar que se tiene una sensibilidad del 68%, por lo que se puede observar que se obtiene se hallan y clasifican correctamente más de la mitad de los clientes que se suscribe. Al igual que se tiene una tasa baja de falsos positivos, por lo que se puede decir que cuando el modelo predice que la instancia pertenece a un cliente con potencial de suscribirse tiene una alta posibilidad de acertar.

A este modelo se le obtuvo la siguiente curva ROC_

[2] Ilustración 17 agregada como apéndice 2

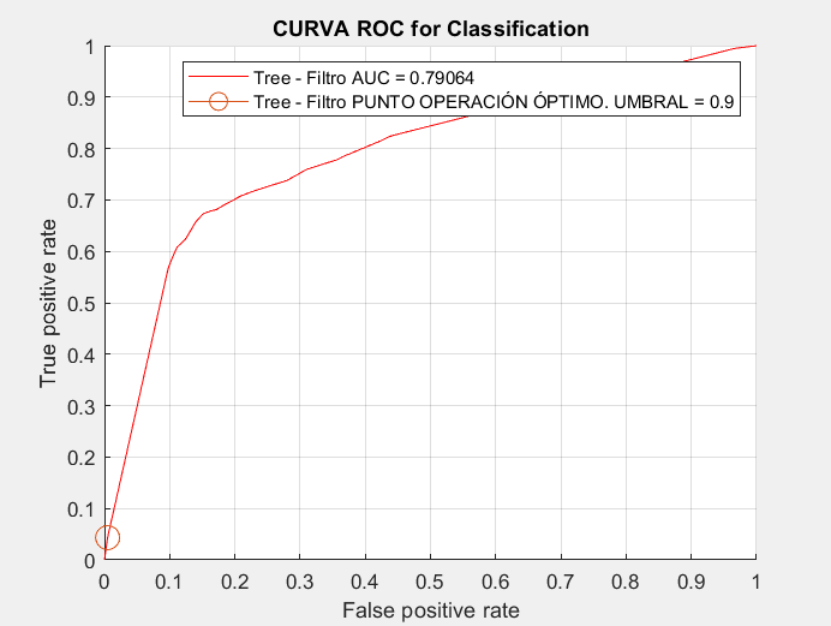


Ilustración 18. Curva ROC del modelo de Árbol de decisión, utilizando selección de atributos

Donde se puede obtener un AUC del 0.79, lo que hace que tenga una exactitud global, pero podemos observar que el punto óptimo de operación no es conveniente, debido a que a pesar de ser el óptimo, no ayuda nuestro problema, ya que este punto brindaría una mejor tasa de aciertos, pero conllevaría no encontrar casi a ningún de los clientes que se suscriben al servicio, clasificando a la mayoría de estos como “Fracaso”, que es lo contrario a lo que buscamos.

5.3.1.3 «Tree-Bagging»

En la generación del modelo Tree-Bagging, se continuó utilizando 350 remuestras para generar los árboles de este método, pero utilizando únicamente los 12 parámetros seleccionados. A partir de la generación de estos y del modelo, se pusieron a prueba con los datos de prueba, y se obtuvieron los siguientes valores de las métricas de clasificación binario.

Tabla 14. Métricas de Clasificación Binaria del modelo de Tree-Bagging con selección de atributos por filtro

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.83657	0.16343	0.74054	0.25946	0.84963	0.15037	0.40117	0.52042

Se puede observar que el modelo tiene una sensibilidad del 74%, obteniendo en este caso uno menor que al utilizar todos los parámetros, que era del 93%, también viene acompañado por una tasa baja de falsos positivos, por lo que al predecir que la instancia pertenece a una clase positiva, tiene una alta probabilidad de acertar. Adicionalmente se puede ver que sigue manteniendo una buena predicción de los clientes que no se suscriben con una especificidad del 0.83.

Adicionalmente se obtuvo la curva ROC de este modelo.

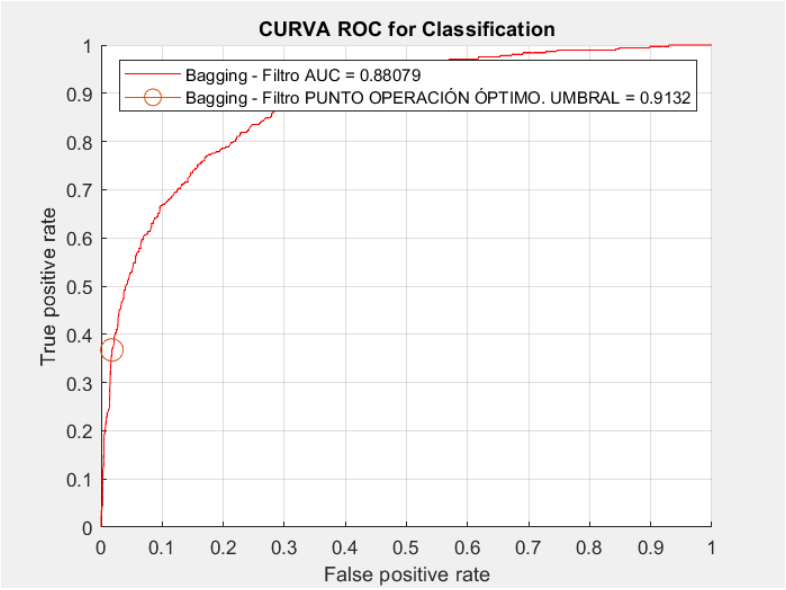


Ilustración 19. Curva ROC del modelo de Tree-Bagging, utilizando selección de atributos

Donde se puede observar que en este modelo se tiene una buena exactitud global, presentando un AUC del 0.8879. así como también se puede observar que tiene un punto más cercano al optimo (0,1), en comparación a modelos anteriores. Por lo que se puede decir que es un buen modelo para la discriminación de nuestras categorías.

5.3.1.4 «Random Forest»

Al igual que en la experimentación anterior, al generar este modelo se utilizaron 350 remuestras para generar los diferentes árboles de decisión variando los parámetros que se utilizan en los modos para minimizar la varianza. En este caso la diferencia cae en los parámetros que se utilizan, solo utilizando los óptimos.

A través de este modelo generado se obtuvieron los siguientes resultados:

Tabla 15. Métricas de Clasificación Binaria del modelo de Random Forest con selección de atributos por filtro

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.8343	0.1657	0.74865	0.25135	0.84596	0.15404	0.39799	0.5197

Se puede observar que continúa siendo un modelo con un buen rendimiento de clasificación teniendo una sensibilidad del 0.74, y una especificidad del 0.84, por lo que logra clasificar correctamente un buen nivel de datos, de ambas clases. La sensibilidad viene acompañada de una tasa de falsos positivos del 15%, por lo que se tiene una cantidad baja de error al predecir una instancia como un "Éxito".

Adicionalmente se obtuvo la curva ROC de este último modelo:

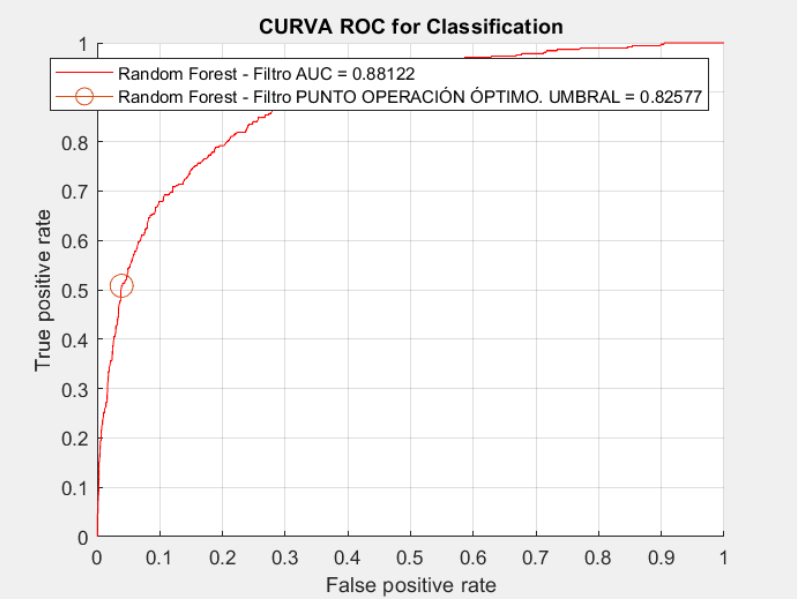


Ilustración 20. Curva ROC del modelo de Random Forest, utilizando selección de atributos

Se puede observar una curva ROC que presenta un buen punto de umbral optimo presentando en el punto (0,0390, 0,5081), en este caso se puede observar que es en el que se obtendrá una mejor exactitud global, ya que es el punto más cercano al punto (0,1). Adicionalmente se encuentra un AUC del 0.8812, por lo que presenta una alta exactitud global en comparación a otros modelos.

5.3.1.5 «Comparación de los Resultados»

En la siguiente tabla se comparan todos los resultados de las métricas de clasificación binaria obtenidos por los 4 modelos generados con la selección de atributos, acompañados de las métricas obtenidos por estos modelos, pero utilizando todos los parámetros, para así poder determinar cuáles de estos son los más optimo y presentan el mayor rendimiento y cumplen con nuestro objetivo de investigación.

Tabla 16. Resumen de las métricas de clasificación binaria obtenida por todos los modelos

	TasaAcierto	Sensibilidad	Especificidad	FScore	AUC
SVM	0.82686	0.63514	0.85294	0.46766	0.78318
Arbol de Decisión	0.80712	0.70541	0.82096	0.46691	0.79089
Bagging	0.84854	0.93784	0.8364	0.59725	0.94677
Random Forest	0.85502	0.94054	0.84338	0.60839	0.94494
Arbol de Decisión - Filtro	0.811	0.68108	0.82868	0.46324	0.7596
SVM - Filtro	0.82686	0.63514	0.85294	0.46766	0.79064
Bagging - Filtro	0.83657	0.74054	0.84963	0.52042	0.88079
Random Forest - Filtro	0.8343	0.74865	0.84596	0.5197	0.88122

Al momento de comparar los 8 modelos generados, incluyendo los que utilizaban una selección de atributos previa, se puede encontrar que los modelos en los que se utilizó la selección de atributos por medio de la técnica de filtros, a pesar de obtener buenos datos de sensibilidad (63-75%) y especificidad (82-84%) en comparación que algunos modelos como el KNN y Naive Bayes, que utilizan todos los parámetros.

Estos en comparación con sus respectivos modelos equivalentes, se puede encontrar que se presenta un mejor rendimiento del modelo y para nuestro objetivo en los modelos que utilizan todos los parámetros, esto debido a que hay modelos presentan una mayor sensibilidad. Este es un parámetro que para nuestra investigación es sumamente importante ya que refleja la tasa de observaciones positivas, es decir de clientes que se suscriben al servicio, clasificados correctamente y esto es lo que se desea lograr predecir el mayor número de clientes que se suscriben.

Se puede observar que en el caso de los modelos que utilizaron la selección de atributos, los mejores modelos también eran el de Tree-Bagging y Random Forest, presentando la tasa de sensibilidad y especificidad más altas de estos conjuntos. Obteniendo ambos modelos una

sensibilidad del 74% y una especificidad del 84%, y al igual obtener un AUC del 0.88, por lo que estos modelos presentan una muy buena exactitud global.

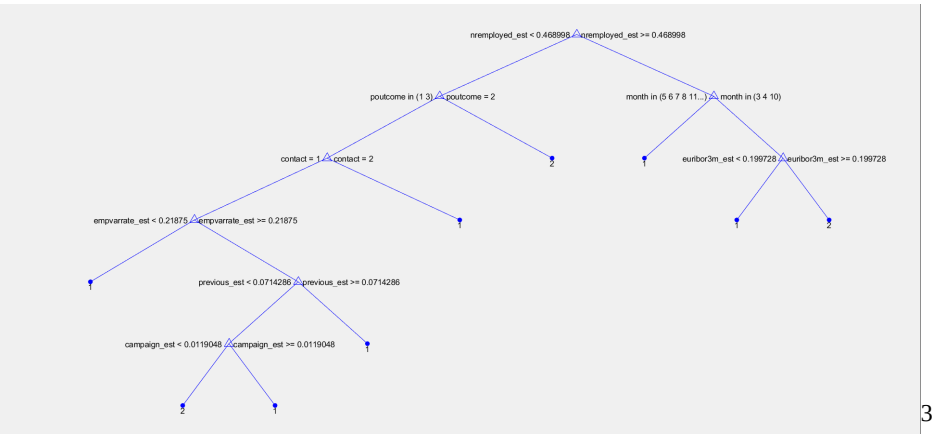
Pero al comparar los diferentes modelos, se encontró que los modelos que utilizan todos los parámetros tienen una mejor discriminación entre las clases, así como presentar una mejor diferenciación para clasificar a un mayor número de clientes que se suscriben al servicio, por lo que se optó por estos modelos en el resto de la investigación, este descarte se realizó conociendo que en los 4 modelos que no utilizaban selección de atributos se presentaban mayores valores de AUC, que sus contrapartes donde si se utilizaba.

5.3.2 «Generación de los modelos utilizando todas las instancias disponibles»

En este caso se quiere observar la influencia de la cantidad de instancias utilizadas para generar los modelos. En estos modelos no se utilizará la muestra balanceada, utilizando toda la información disponible. Este es el resultado hallado en los diferentes modelos que se generaron (SVM, Árbol de decisión, Tree-Bagging y Random Forest):

5.3.2.1 «Árbol de Decisión»

En el caso del árbol, se puede encontrar que, al optimizar el modelo, utilizando las 30.000 instancias, se obtuvo el siguiente árbol de clasificación optimo:



[3]

Ilustración 21. Árbol de decisión optimo utilizando todas las instancias sin muestra balanceada

Utilizando este árbol seleccionado como optimo, ahora con el conjunto de datos de prueba, se obtuvieron los siguientes resultados en las métricas de clasificación binaria:

Tabla 17. Métricas de Clasificación Binaria del modelo de Árbol de Decisión sin utilizar muestra balanceada

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.89838	0.10162	0.27838	0.72162	0.98272	0.017279	0.68667	0.39615

Se puede observar que este no es un modelo optimo, principalmente ubicando la sensibilidad es del 0.27, considerando que es la métrica más importante para nuestra investigación. Pero podemos observar que tiene una tasa de aciertos del 0.89, es alta, y esto se debe a que tiene una especificidad del 98%, por lo que acierta casi todas las instancias de clientes que no se suscriben que son casi el 90% del conjunto de datos.

Adicionalmente se obtuvo su curva ROC, presentando un valor de exactitud global del 0.76.

[3] Ilustración 21, agregada como apéndice 3

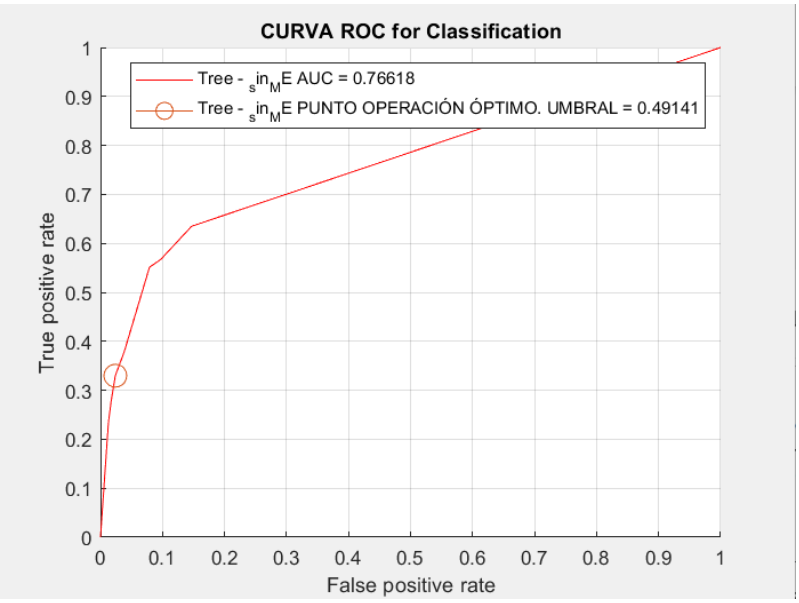


Ilustración 22. Curva ROC del modelo de Árbol de decisión sin utilizar muestra balanceada

5.3.2.2 «SVM»

En el caso de la metodología SVM, se utilizó de nuevo un kernel lineal, que fue encontrado como optimo, al generar el modelo con todas las instancias disponibles. A partir de este modelo entrenado, se utilizó con el conjunto de prueba, con las predicciones elaboradas, se obtuvieron las siguientes métricas de clasificación binaria:

Tabla 18. Métricas de Clasificación Binaria del modelo SVM sin utilizar muestra balanceada

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.89191	0.10809	0.22162	0.77838	0.98309	0.016912	0.64063	0.32932

Se puede observar al igual que en el caso del árbol, una tasa de acierto alta, casi del 90% pero va acompañado de una sensibilidad muy baja de 0.22. Por lo que este modelo no discrimina de manera correcta a los clientes que si se suscriben al servicio ofrecido por el banco. Adicionalmente se obtiene la curva ROC, donde se puede observar una exactitud de 0.70, con un punto óptimo de umbral alejado del (0,1).

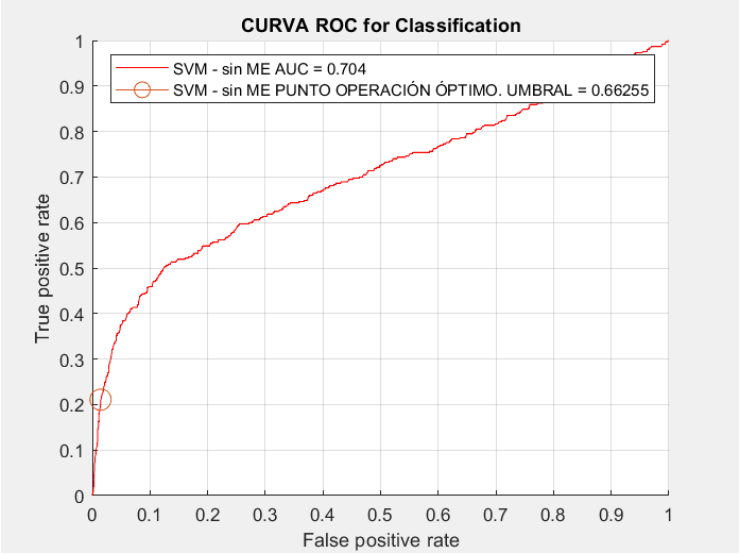


Ilustración 23. Curva ROC del modelo SVM sin utilizar muestra balanceada

5.3.2.3 «Tree-Bagging»

En este caso se volvieron a utilizar 350 remuestras para generar los árboles y entrenar el modelo utilizando las 30.488 instancias, para posteriormente utilizarlo con los datos de prueba, con estas predicciones, obtuvimos los siguientes resultados en las métricas de clasificación binaria.

Tabla 19. Métricas de Clasificación Binaria del modelo Tree-Bagging sin utilizar muestra balanceada

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.95469	0.045307	0.63514	0.36486	0.99816	0.0018382	0.97917	0.77049

Se puede observar una muy buena tasa de acierto y una casi perfecta tasa de especificidad, del 0,99, adicionalmente podemos ver una sensibilidad de 0,65, acompañada de una tasa de falsos positivos de 0,0018, por lo que este modelo al predecir una instancia como “Éxito”, es casi seguro que esta instancia lo sea.

Adicionalmente se obtuvo su curva ROC, donde podemos observar que es una curva casi perfecta, presentando un AUC de 0,9886.

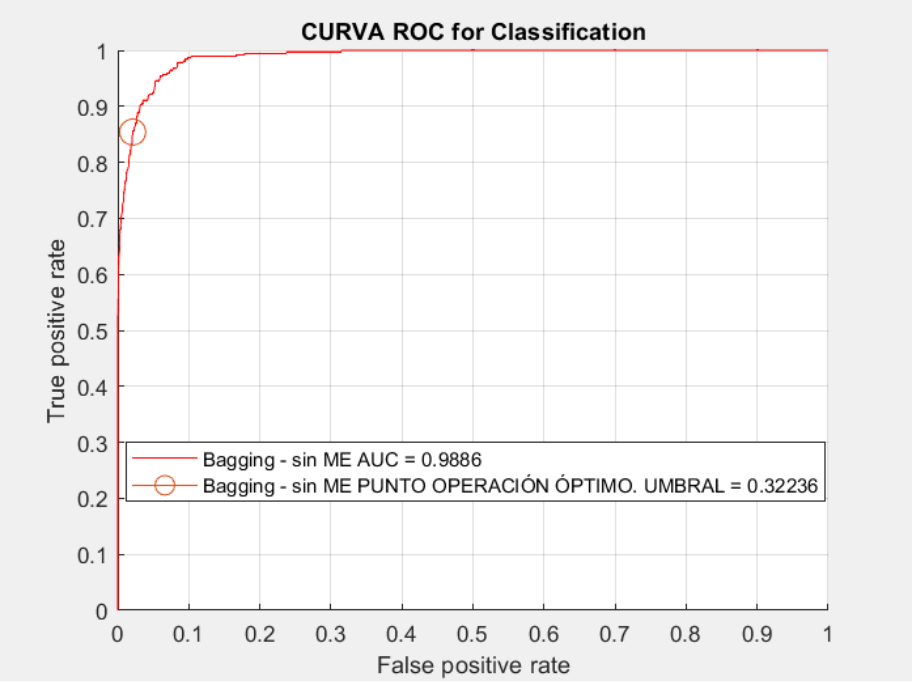


Ilustración 24. Curva ROC del modelo de Tree-Bagging sin utilizar muestra balanceada

5.3.2.4 «Random Forest»

En el caso de la metodología del Random Forest, se utilizaron de nuevo 350 árboles, para generar el modelo. Una vez entrenado con toda la información del conjunto de datos obtenido, se utilizó el modelo con los datos de prueba y se obtuvieron las siguientes métricas:

Tabla 20. Métricas de Clasificación Binaria del modelo de Random Forest sin utilizar muestra balanceada

TasaAcierto	TasaError	Sensibilidad	TasaFalsosNeg	Especificidad	TasaFalsosPos	Precision	FScore
0.95469	0.045307	0.63784	0.36216	0.99779	0.0022059	0.97521	0.77124

Al igual que en el caso del Tree-Bagging, se puede observar una tasa de acierto del 95%, acompañada de una especificidad de 0,99, por lo que nuestro modelo clasifica correctamente a todos los usuarios que no se suscriben al servicio. Esto acompañado de una sensibilidad de 0,63, que no es la más alta presentada hasta el momento.

Adicionalmente se obtuvo su curva ROC, esta presenta un AUC de 0,98, por lo que es casi perfecta, con un punto óptimo de 0,3267.

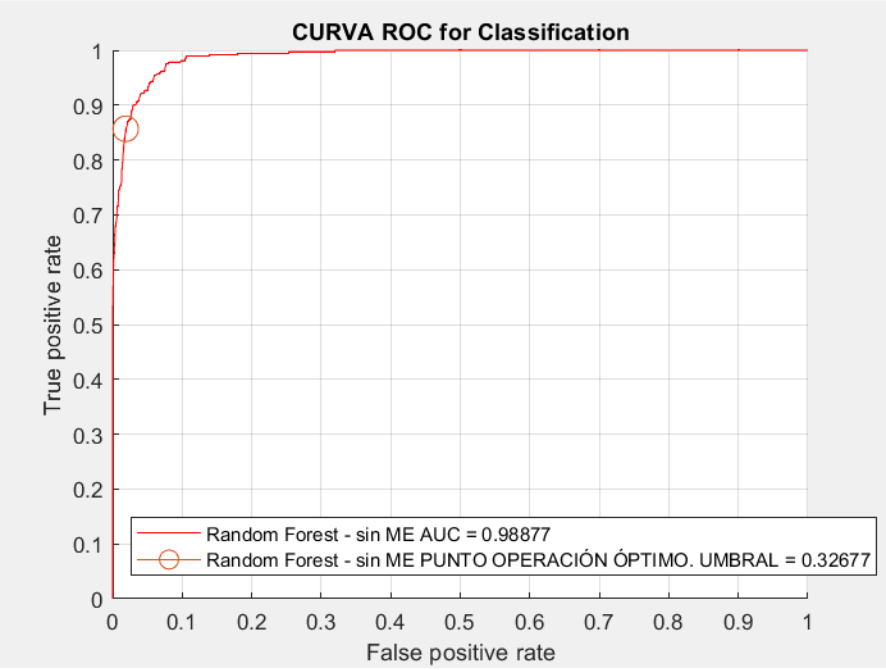


Ilustración 25. Curva ROC del modelo de Random Forest sin utilizar muestra balanceada

5.3.2.5 «Comparación de los resultados»

Tabla 21. Comparación de las Métricas de Clasificación Binaria con los modelos que no utilizan una muestra balanceada (ME)

	TasaAcierto	Sensibilidad	Especificidad	FScore	AUC
SVM	0.82686	0.63514	0.85294	0.46766	0.78318
Arbol de Decisión	0.80712	0.70541	0.82096	0.46691	0.79089
Bagging	0.84854	0.93784	0.8364	0.59725	0.94677
Random Forest	0.85502	0.94054	0.84338	0.60839	0.94494
Arbol de Decisión - sin ME	0.89838	0.27838	0.98272	0.39615	0.704
SVM - sin ME	0.89191	0.22162	0.98309	0.32932	0.76618
Bagging - sin ME	0.95469	0.63514	0.99816	0.77049	0.9886
Random Forest - sin ME	0.95469	0.63784	0.99779	0.77124	0.98877

Al comparar los resultados obtenidos por los modelos que se elaboraron utilizando una muestra balanceada con una de no balanceada, se pueden observar que dependiendo del modelo se obtuvieron resultados distintos.

En el caso de los modelos SVM y Árbol de decisión, se puede observar que de manera general los modelos que utilizan una muestra balanceada presentan mejores resultados, presentando una mayor sensibilidad y AUC, que los modelos que utilizan todas las instancias disponibles.

Pero al observar los modelos de Tree-Bagging y Random Forest, podemos observar que se da el caso contrario, encontrándonos que el no utilizar una muestra balanceada presenta mejores resultados, obteniendo una tasa de 0.95, en comparación del 0.85 en el caso de cuando se usa una muestra balanceada. Así como también hay un aumento es la especificidad del 84% al 99%, adicionalmente presentan una exactitud global (AUC) del 99%, a pesar de que disminuye el nivel de sensibilidad, bajando del 0.94 que se obtienen en el caso de los modelos que utilizan una muestra balanceada por 0.64 con estos modelos que no lo utilizan, el alto AUC, nos indican que estos modelos tienen un mayor rendimiento.

En base a estos resultados y las curvas ROC obtenidas por los modelos de Tree-Bagging y Random Forest sin utilizar muestra balanceada y al presentar una curva casi perfecta, se puede decir que los modelos que no utilizan la muestra balanceada presentan mejores resultados, y se agregó el modelo de Random Forest, a los modelos seleccionados, como posible modelo final en la experimentación

5.4 «Implementación del modelo final»

En esta etapa se utilizarán los 3 modelos óptimos encontrados, que son los modelos generados por la metodología de Tree-Bagging y Random Forest en la experimentación 1, utilizando todos los parámetros y utilizando una muestra balanceada, y el modelo de Random Forest, cuando no se utiliza una muestra balanceada.

5.4.1 «Contraste del modelo utilizando el punto de umbral óptimo de la curva ROC»

A través de la curva ROC, se puede encontrar el punto óptimo de umbral, este es el que se encuentra más cercano al punto óptimo de la curva ROC (0,1). Este punto nos indicara el mejor punto de umbral, esto quiere decir el umbral al que tiene que llegar la probabilidad de una instancia a pertenecer a una de las 2 clases, para poder clasificarla en alguna de estas. Se considera como el punto óptimo debido a que utilizando este punto es en el que se obtendrá la mayor exactitud posible.

En este caso se modificó el punto de umbral óptimo para los modelos seleccionados, normalmente Matlab utiliza un umbral de 0,5 para ambas clases. En este caso el punto de umbral para el modelo de Tree-Bagging será de 0,8679; Mientras que para el modelo de Random Forest, el punto de

umbral será 0,8543 y en el Random Forest sin utilizar muestra balanceada, se tiene un punto de umbral óptimo de 0,3267.

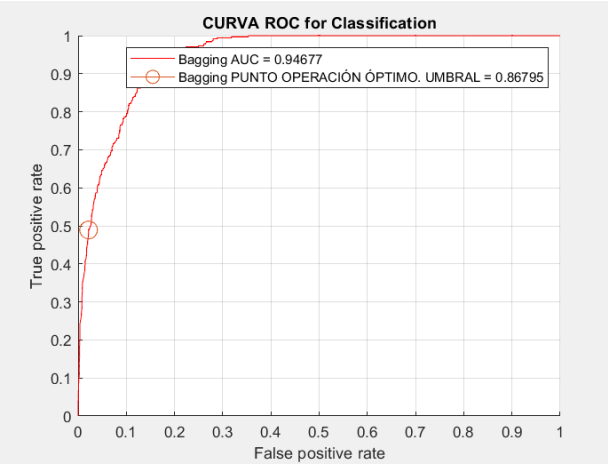


Ilustración 26. Curva ROC del modelo Tree-Bagging Optimo

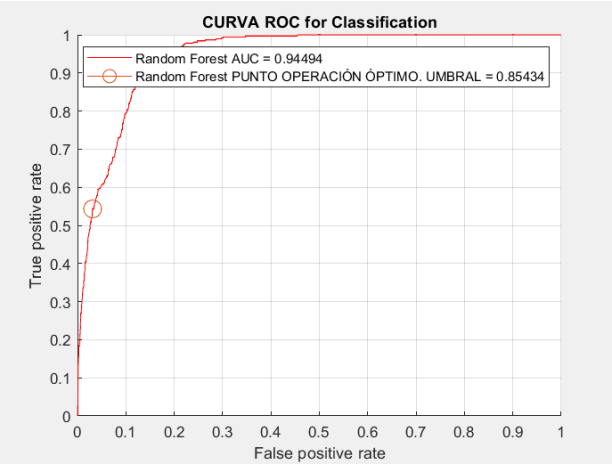


Ilustración 27. Curva ROC del modelo Random Forest Optimo utilizando una muestra balanceada

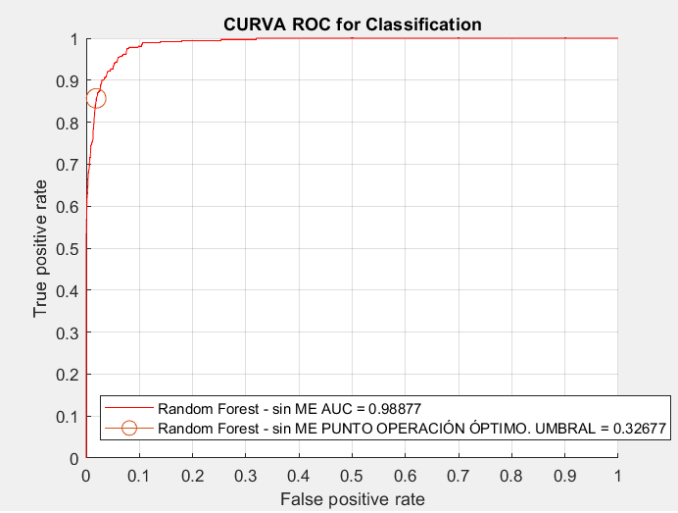


Ilustración 28. Curva ROC del modelo Random Forest Optimo no utilizando una muestra balanceada

A partir de este cambio se realizó nuevamente la clasificación de las predicciones hechas por ambos modelos, pero ahora se clasificarán utilizando estos nuevos umbrales. A partir de estos cambios se obtuvieron los siguientes resultados en las métricas de clasificación binaria.

Tabla 22. Métricas de Clasificación Binaria de los 3 modelos seleccionados utilizando el umbral optimo

	TasaAcierto	Sensibilidad	Especificidad	FScore	AUC
Bagging	0.84854	0.93784	0.8364	0.59725	0.94677
Bagging Umbral Optimo	0.91974	0.48919	0.97831	0.59344	0.94677
Random Forest	0.85502	0.94054	0.84338	0.60839	0.94494
Random Forest Umbral Optimo	0.9178	0.54324	0.96875	0.6128	0.94494
Random Forest sin ME	0.95469	0.63784	0.99779	0.77124	0.98877
Random Forest Umbral Optimo sin ME	0.96634	0.85676	0.98125	0.85908	0.98877

Se puede observar que, en el caso de los modelos donde se utilizó una muestra balanceada, en los modelos en los que se modificó el punto de umbral por el punto óptimo presentan una tasa de aciertos mayor, al que de los modelos a los que no se les modifico. Un cambio importante en estas métricas, y es que, para llegar a este nivel de acierto mayor, se comprometió a la sensibilidad del modelo, en la que en ambos casos decayó del 94% que tenían los modelos aproximadamente, ahora presentando valores alrededor del 50%. Al igual que un aumento en la especificidad de los modelos presentando antes una tasa del 84% aproximadamente en ambos modelos, y ahora del casi 97% en ambos casos.

Esto no es lo ideal para nuestro problema, ya que a pesar de tener una tasa de aciertos muy buena del 91%, se compromete a la sensibilidad que es la métrica más relevante en nuestra investigación, ya que es la que se encarga de medir la tasa de instancias positivas clasificadas correctamente. Que en nuestro caso sería la tasa de clientes que se suscriben al servicio que son clasificados correctamente y es lo que buscamos maximizar, encontrar la mayor cantidad de estos clientes para enfocarnos en ellos.

Esta tasa tan buena de acierto se debe a que en la muestra únicamente el 12% de los datos de la muestra son instancias de clientes que se suscriben mientras que el resto son de clientes que no, por lo que si se tiene una muy alta especificidad se puede obtener una muy buena tasa de aciertos ya que representan la mayoría de los datos.

Ahora en el caso de observar el modelo donde no se utilizó una muestra balanceada se dio un caso diferente, en este caso el umbral optimo mejoro el modelo, no solo en la tasa de aciertos subiendo de 0,95 a 0,96, sino adicionalmente mejorando la sensibilidad, que hasta el momento de los

modelos seleccionados era el que presentaba la sensibilidad más baja con un valor de 0,63, mientras que al utilizar el punto de umbral optimo, se aumenta a 0,85, y en este caso la especificidad se encuentra insignificantemente disminuida pasando de 0,99 a 0,98. Por estos motivos se escoge esta opción como la óptima y con la que se continuara la experimentación.

A pesar de que este modelo no presenta la sensibilidad más alta, como su contraparte del modelo de Random Forest utilizando una muestra balanceada, pero aun así, observando las curvas ROC de ambos modelos, se puede observar que el modelo que no utiliza una muestra balanceada, puede brindar una predicción más optima, y se puede aumentar la sensibilidad de este, modificando su punto de umbral, pero esta vez, no buscando la tasa de aciertos más alta, sino modificándolo en base a aumentar la sensibilidad del modelo.

5.4.2 «Contraste del modelo modificando el punto de umbral en base a la sensibilidad»

A través de la curva ROC de este modelo, se obtuvieron 4 puntos diferentes de umbral, para observar el comportamiento del modelo y el rendimiento de este, al variar y mejorar la sensibilidad en el modelo. Se buscaba el punto de umbral donde la sensibilidad llegar al 90, 95, 97 y 99%.

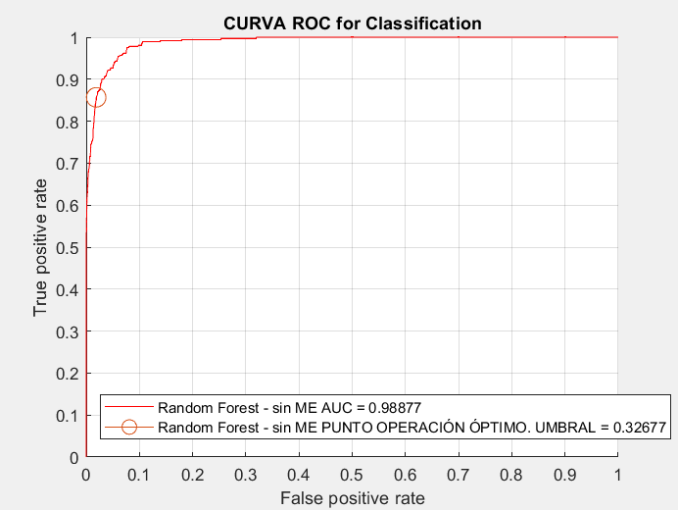


Ilustración 29. Curva ROC del modelo optimo seleccionado

Los puntos de umbral con cada una de estas sensibilidades es el siguiente:

Tabla 23. Niveles de Sensibilidad

Nivel De Sensibilidad	Punto De Umbral
90%	0,2797
95%	0,2215
97%	0,1821
99%	0,1134

Para cada uno de estos niveles de sensibilidad se obtuvo su matriz de confusión, presentando los siguientes resultados:

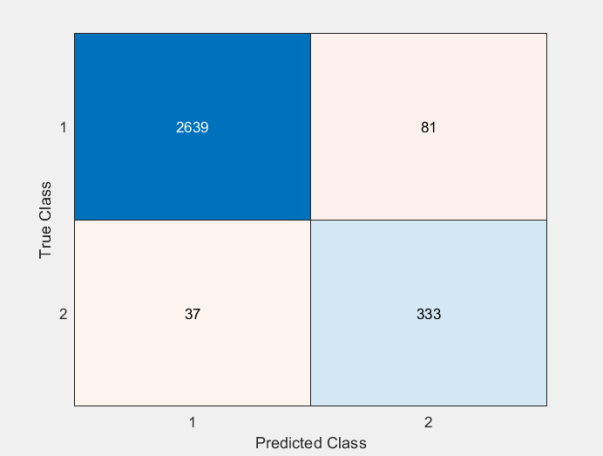


Ilustración 30. Matriz de confusión utilizando una sensibilidad del 90%

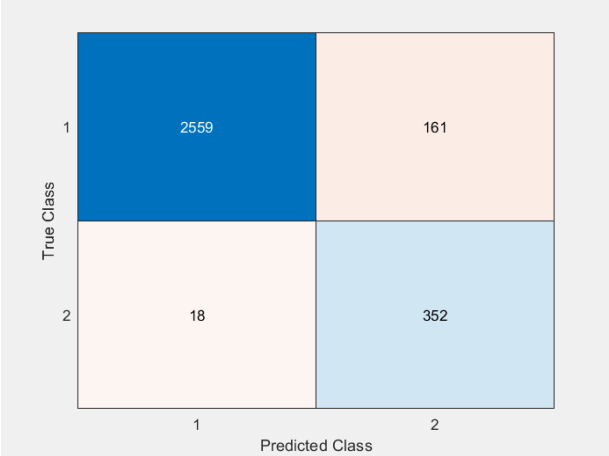


Ilustración 31. Matriz de confusión utilizando una sensibilidad del 95%

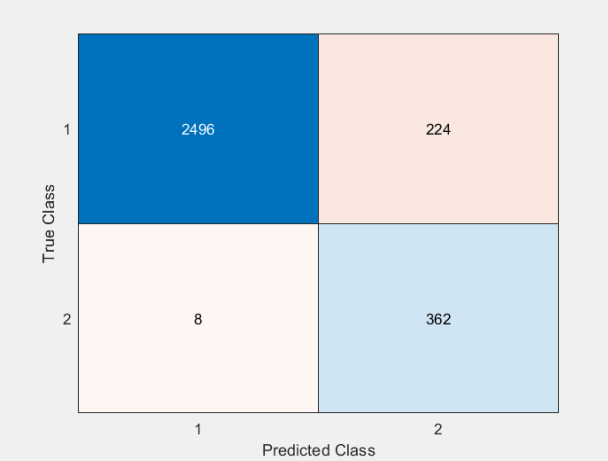


Ilustración 32. Matriz de confusión utilizando una sensibilidad del 97%

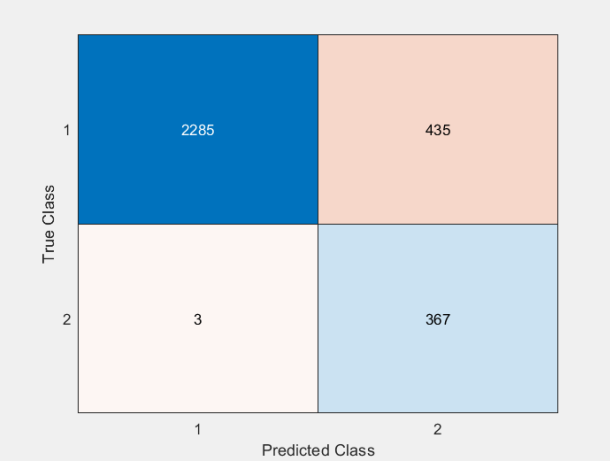


Ilustración 33. Matriz de confusión utilizando una sensibilidad del 99%

Adicionalmente a partir de la modificación del umbral y obtener las métricas para cada uno de estos umbrales, se encontraron los siguientes resultados.

Tabla 24. Métricas de Clasificación Binaria del modelo Random utilizando el cambio de umbral

	TasaAcierto	Sensibilidad	Especificidad	FScore	AUC
Random Forest sin ME	0.95469	0.63784	0.99779	0.77124	0.98877
Random Forest Umbral Optimo sin ME	0.96634	0.85676	0.98125	0.85908	0.98877
Random Forest sin ME con umbral Sensibilidad 90%	0.96181	0.9	0.97022	0.84949	0.98877
Random Forest sin ME con umbral Sensibilidad 95%	0.94207	0.95135	0.94081	0.79728	0.98877
Random Forest sin ME con umbral Sensibilidad 97%	0.92492	0.97838	0.91765	0.75732	0.98877
Random Forest sin ME con umbral Sensibilidad 99%	0.85825	0.99189	0.84007	0.62628	0.98877

A partir de esto resultados se pudo encontrar que el modelo con una sensibilidad del 95% era la opción más considerada como solución al banco. Tomando de base los datos de prueba, se puede observar que, de las 370 observaciones, 352 fueron acertadas correctamente, y únicamente 18 no lo han sido. Si comparamos estos datos con los resultados obtenidos por los modelos que tienen una mayor sensibilidad, se puede observar que pasa de 352 aciertos a 362 en el caso del 97% y a 367 en el caso de una sensibilidad del 99%.

Por lo que el aumentar la sensibilidad cambia en únicamente 10 instancias en el caso del 97% y de 15 en el caso de utilizar la sensibilidad al 99%. Por lo que podemos observar que son muy pocas instancias las que se dejarían fuera al escoger la sensibilidad del 95%.

Pero adicionalmente se debe de tomar en cuenta, la cantidad de falsos positivo, estos son importantes ya que se debe de tomar el costo de tiempo y de recursos que conllevara para el banco, llamar a estos usuarios que se han clasificado de manera incorrecta, y no se suscribirán al servicio. En el caso de usar una sensibilidad del 95% se encontró que 161 usuarios fueron clasificados como "Éxito" cuando en realidad no se suscribirán al servicio. Pero podemos ver que hay un aumento de 63 usuarios al elevar la sensibilidad al 97%, teniendo 224 usuarios como falsos positivos. Mientras que si se comparar con el valor obtenido con una sensibilidad del 99%, se encuentra una diferencia mayor a 270 usuarios, presentando una cantidad de 435 usuarios mal clasificados.

Tomando de base estos factores se podría considerar el de 95% como el modelo optimo ya que podemos observar que es mejor para el banco, ya que el aumentar el acierto 10 o 15 instancias y que esto conlleve hasta 270 clientes que se encontraran mal calificados y presentaran un gasto para el banco. Adicionalmente conociendo que las instancias de clientes que se suscriben en comparación con los que no mantendrá una relación baja y el aumentar la sensibilidad, hará que se presente este efecto donde se aumentara la cantidad de aciertos en cantidades muy bajas, y conllevara una tasa de falsos positivos muy alta, no será conveniente.

Por lo que el modelo óptimo para la discriminación de los clientes que tienen un mayor potencial para suscribirse al servicio ofrecido por el banco del depósito bancaria y obtención de la sensibilidad más optima es la del modelo Random Forest, utilizando todos los parámetros de entrada disponible, modificando el umbral de corte de las clases, a uno asegurando la sensibilidad del 95%.

6 Conclusiones

Las principales conclusiones que se derivan de la realización del presente trabajo pueden resumirse como sigue;

A. Utilizar un modelo de Random Forest, utilizando todas las instancias disponibles en el conjunto de datos, acompañado de un proceso para modificar el umbral para la clasificación de las instancias buscando una tasa de sensibilidad objetivo del 95%, es la solución óptima para hallar la mayor cantidad de clientes posible que se suscriben al servicio y reducir la cantidad de contactos.

B. La selección de atributos para eliminar las variables de menor influencia en la variable de salida de nuestro problema no siempre genera modelos más optimo. Durante la experimentación el utilizar todos los parámetros disponibles brindaban una mejor sensibilidad y AUC, que los modelos que utilizaban selección de atributos.

C. Los métodos de ensemble (Tree-Bagging y Random Forest) presentan una mayor exactitud y rendimiento al utilizar toda la información e instancias disponible en lugar de utilizar un conjunto balanceado, pero esta mejora depende del modelo ya que modelos como el SVM, que presentan un comportamiento contrario.

D. Modificar el punto de trabajo ajustando el umbral de salida del modelo, utilizando el punto óptimo de la curva ROC, genera los mejores modelos en base a la tasa de aciertos, pero esto puede no reflejar la mejor solución a un problema ya que podría ser de más utilidad trabajar con otros niveles de sensibilidad/especificidad según el criterio de la entidad bancaria.

E. Los métodos de ensemble pueden generan modelos con mayor rendimiento, fortaleciendo modelos débiles como los árboles de decisión, al generar y utilizar conjunto de modelos reduciendo así la varianza y error a la hora de clasificar las instancias. Como lo fueron los modelos de Random Forest y Tree-Bagging durante nuestra investigación, presentando las mayores tasas de sensibilidad y especificidad.

F. Los factores macroeconómicos, influyen en la decisión del cliente a suscribirse al servicio del depósito a largo plazo ofrecido por el banco. Si estos reflejan un ambiente económico

donde los clientes sienten incertidumbre económica, como lo puede ser una recesión, los clientes preferirán quedarse con su dinero en lugar de darlo a una entidad bancaria, explicando por qué hay una tasa tan alta de rechazo al servicio ofrecido.

Referencias

Santibáñez-Arellano, Tomás. (2015). Validez predictiva de un clasificador basado en máquina de soporte vectorial para éxito o fracaso en la extubación de pacientes conectados a ventilación mecánica invasiva en una unidad de paciente crítico adulto. 10.13140/RG.2.1.4939.2082.

Martin Guareño, Juan José. (2016). Support Vector Regression: Propiedades y Aplicaciones. Universidad de Sevilla. Facultad de Matemática.

López Díaz, Miguel. (2018). Fundamentos Matemáticos de los Métodos Kernel para Aprendizaje Supervisado. Universidad de Sevilla. Facultad de Matemáticas.

Berástegui Arbeloa, Gonzalo & Galar Idoate, Mikel. (2018). Implementación del Algoritmo de los K vecinos más cercanos (K-NN) y estimación del mejor valor local de K para su cálculo. Universidad Pública de Navarra. Facultad de Ingeniería.

Hanneman, Robert A. and Mark Riddle. (2005). Introduction to social network methods. Riverside, CA: University of California, Riverside.

Mayorga Ortiz, P., Druzgalski, C., Criollo Arellano, M. A., & Ganzález Arriaga, O. H. (2013). GMM y LDA Aplicado a la Detección de Enfermedades Pulmonares. Revista Mexicana de Ingeniería Biomédica, 34(2), 133–144.

Guevara, S., Bouchet, A., Brun, M., & Ballarin, V. (2019). Diseño Automático de un Clasificador para Filtrado de Ruido en Imágenes Binarias Utilizando Análisis Discriminante Lineal. Revista Digital del Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de La Matanza, 4(1), 1–9.

Análisis discriminante lineal (LDA) y análisis discriminante cuadrático (QDA) by Joaquín Amat Rodrigo, available under a Attribution 4.0 International (CC BY 4.0)

Mosquera, Rodolfo, Castrillón, Omar D., & Parra, Liliana. (2018). Máquinas de Soporte Vectorial, Clasificador Naïve Bayes y Algoritmos Genéticos para la Predicción de Riesgos

Psicosociales en Docentes de Colegios Públicos Colombianos. Información tecnológica, *29*(6), 153-162.

Zhang, Harry. (2004), The optimality of naïve Bayes. Aa, 1(2), 3

Barrientos, R., Curz, N., Acosta, H., Rabatte, I., Gogeascoechea, M. Pavón, P. & Blázquez, S. (2009). Árboles de decisión como herramienta en el diagnóstico médico. Revista Médica de la Unicersidad de Veracruz, 9(2), 19-24.

Berlanga, V., Rubio, M. & Vilà, R., (2013). Cómo aplicar árboles de decisión en SPSS. Revista d'Innovación i Recerca en Educació, 6(1), 65-79

Gareth, J., Witten, D., Hastie, T. & Tibshirani, R. (2013). An Introduction to Statisticar Learning with Applications in R. Springer Science+Business Media, New York.

Cardona, N. (2019). Predicción y selección de variables con bosques aleatorios en presencia de variables correlacionadas. Universidad Nacional de Colombia. Facultad de Ciencias.

Blasco Santos, Sandra. (2018). Aplicación de técnicas Boosting al modelado y predicción de la serie de precios de la energía eléctrica. Universidad Politécnica de Madrid, Escuela técnica superior de ingenieros industriales, Madrid, España.

Perez Garcia, Manuel. (2018). Técnicas Boosting. Universidad de Sevilla, Departamento de Estadística e Investigación Operativa. Sevilla, España.

Medina, R. & Nique, C. (2017). Bosques Aleatorios como Extensión de los Árboles de Clasificación con los Programas R y Python. Revista Interfases de la Carrera de ingeniería de Sistemas de la Universidad de Lima. 10(1), 165-189.

Argañaraz, J.P. & Entraigas, I. (2011). Análisis comparativo entre las máquinas de vector soporte y el clasificador de máxima probabilidad para la discriminación de cubiertas de suelo. Revista de Teledetección, 36(1), 26-39.

Toledo, A. (2016). Métodos de selección de atributos para clasificación supervisada basados en teoría de información. Instituto Superior Politécnico “José Antonio Echeverría”, Facultad de Ingeniería informática, Cuba.

Ruiz, R. Aguilar-Ruiz, J. & Riquelme, J. (2002). Evaluación de Rankings de Atributos para Clasificación. Universidad de Sevilla, Sevilla, España.

Dorado, H. (2019). Wrapper para la construcción de modelos de aprendizaje supervisado basado en arreglos de cobertura que permite la estimación de la importancia de las variables de entrada y la selección de atributos. Universidad del Cauca, Facultad de Ingeniería Electrónica y Telecomunicaciones, México.

Ortiz, E., Hiram, J., Arguijo, P. & Melendez-Armenta. R. (2020). Selección de características con método wrapper para un sistema de detección de intruso: caso CICIDS-2017. Instituto Tecnológico Superior de Misantla, México. ISSN 1870-4069.

Medel, C. (2015). Probabilidad clásica de sobreajuste con criterios de información: estimaciones con series macroeconómicas chilenas. Revista de Análisis Económico, 30(1), 57-72.

Laura, L. (2019). Evaluación de Algoritmos de Clasificación utilizando Validación Cruzada. Industry, Innovation, and Infrastructure for Sustainable Cities and Communities: Proceedings of the 17th LACCEI International Multi-Conference for Engineering, Education and Technology, Jamaica.

Borja-Robalino, R., Monleón-Getino, Antonio & Rodellar, J. (2020). Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning. Revista Ibérica de Sistemas e Tecnologías de Informação. 30(1).

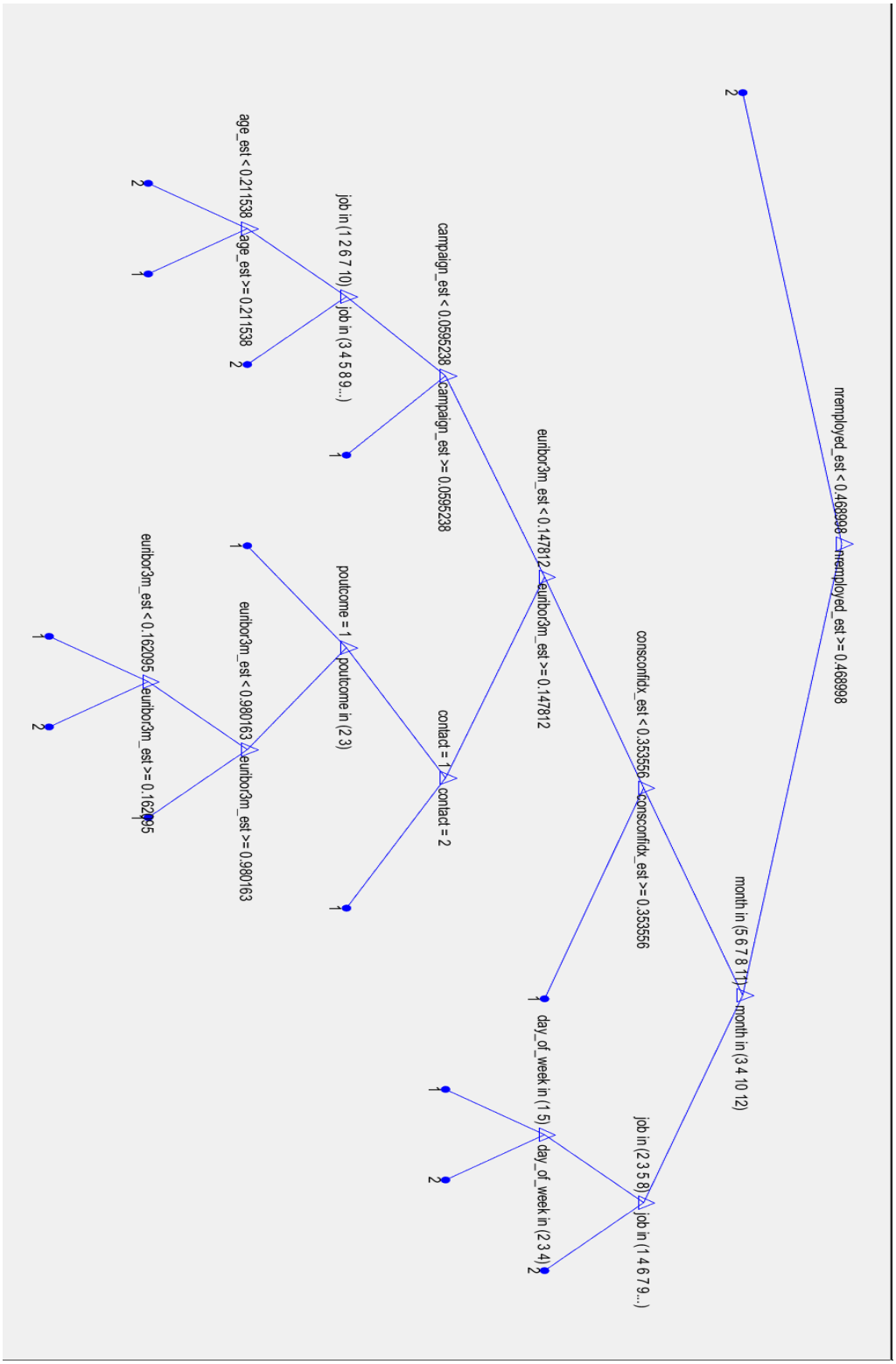
Corso, C. (2009), Aplicación de algoritmos de clasificación supervisada usando Weka. Universidad Tecnológica Nacional, Argentina.

Amoros, Pablo. (2021). Desbalanceo de datos en redes de clasificación binaria. Universidad de Alicante, España.

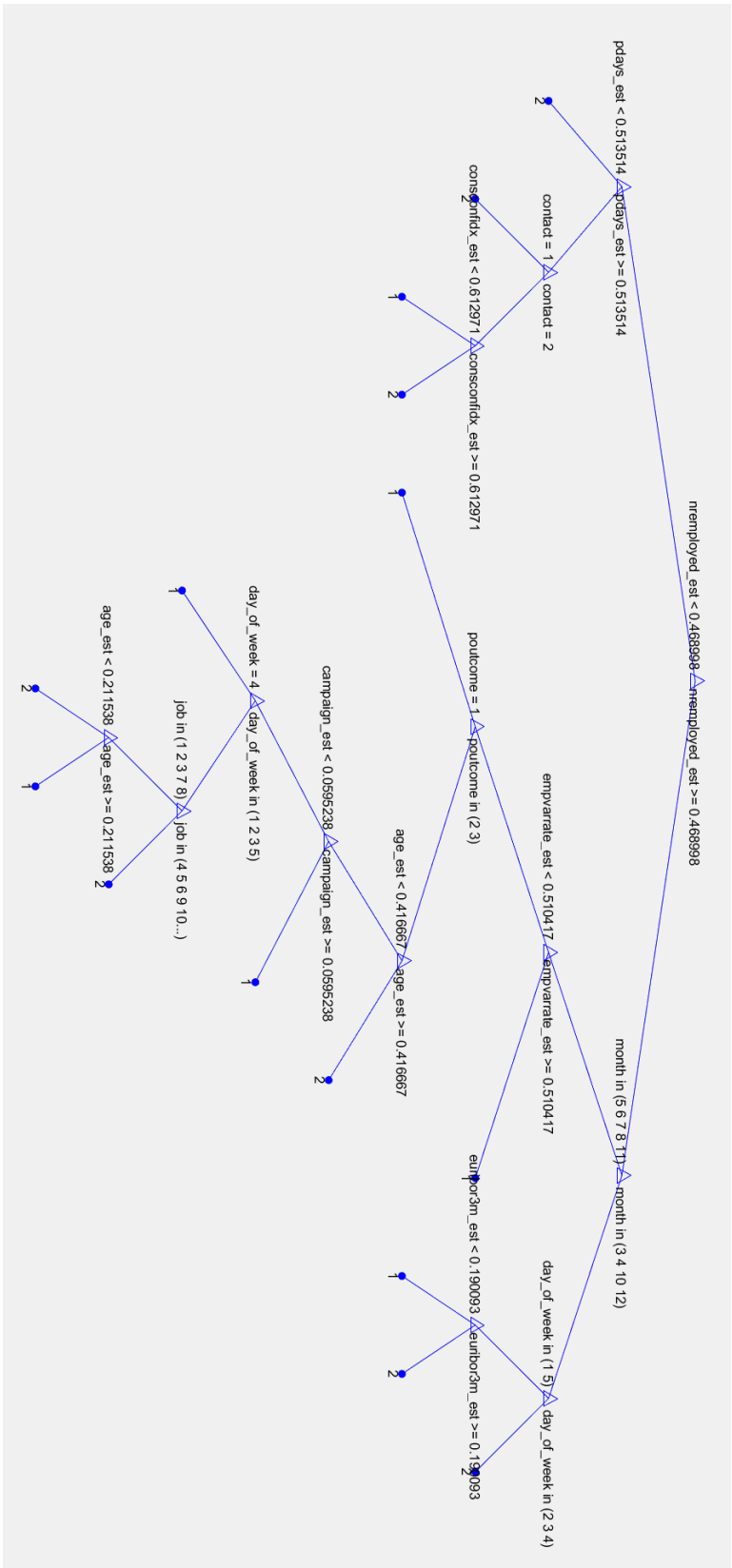
Lopez de Ullibarri, G. & Pita, S. (2001). Curvas ROC. Universidad de Epidemiologia Clínica y Bioestadística. A Coruña, España.

Apéndice

1. Árbol de decisión optimo encontrado utilizando todos los atributos



2. Árbol de decisión optimo encontrado utilizando solo los atributos seleccionados por la metodología de Filtro utilizando el test de Chi-cuadrado



3. Árbol de decisión optimo utilizando todas las instancias del conjunto de entrenamiento, sin utilizar una muestra balanceada.