UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

Comparación de modelos Machine Learning aplicados al riesgo de crédito

POR

Tamahí Constanza Martínez Fernández

Tesis presentada a la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Concepción para optar al título profesional de Ingeniera Civil Matemática

Profesores: Jorge Figueroa, Guillermo Ferreira, Reinaldo González

Abril de 2022
Concepción, Chile

Agradecimientos

Quiero comenzar dando las gracias al equipo de riesgo de crédito de KPMG por confiar en mí y darme la oportunidad de realizar mi práctica y memoria de título, gracias por todo el apoyo, comprensión y calidad humana.

Agradecer a mis profesores, por su tiempo y buena disposición. Como también a todo el Departamento de Ingeniería Matemática y la Universidad de Concepción por brindarnos las herramientas y conocimientos para nuestra formación como Ingenieros.

Por último y no menos importante, doy las gracias a mi familia y a todos mis seres queridos.

A mi mamá y papá, gracias por su cariño y apoyo incondicional durante todos estos años, a mi hermano Omar, gracias por todos tus consejos y tu constante preocupación, a nuestro integrante peludo de la familia Puntito, gracias por ser nuestro fiel compañero y siempre alegrarnos el corazón. Estoy eternamente agradecida de ustedes por creer en mis capacidades y por siempre cuidar de mí.

A mi pololo Fernando, gracias por siempre confiar en mí, por tu paciencia y tu cariño, por compartir tus conocimientos conmigo y que siempre estuvieras dispuesto a ayudarme, ya sea estudiando juntos o animándome con tus palabras de apoyo.

A mis compañeros y amigos de la carrera, a mis amigas de la pensión, gracias por alegrar nuestra vida universitaria, por todos esos pequeños momentos de distracción que nos hacían olvidar el estrés y por siempre acompañarnos y apoyarnos mutuamente.

A cada uno de ustedes, gracias por todo.

Resumen

De acuerdo al marco regulatorio que rige a las instituciones financieras, es necesario que a la hora de evaluar el riesgo de crédito las empresas establezcan de forma clara modelos que estimen la probabilidad de que un cliente falle con el objetivo de constituir provisiones necesarias que permitan cubrir eventuales pérdidas. Comúnmente la técnica estadística adoptada para este próposito en la industria financiera corresponde a la regresión logística, sin embargo, en los últimos años se ha prestado una atención creciente a los algoritmos de aprendizaje automático (Machine Learning) para desafiar y explorar nuevas soluciones a la modelación de la probabilidad de incumplimiento. Es por esto que el objetivo de la presente memoria de título consiste en comparar la capacidad predictiva de siete algoritmos de Machine Learning para la clasificación de deudores según su probabilidad de incumplimiento. Específicamente los algoritmos estudiados fueron regresión logística, análisis discriminante lineal, árboles de decisión, random forest, gradient boosting, extreme gradient boosting y support vector machines.

Índice General

Índice de Figuras

Índice de Tablas

1. Introducción

Al momento de conceder un crédito existe la posibilidad de que un cliente no pueda cumplir con sus obligaciones de pago, a esta posibilidad se le conoce como riesgo de crédito. Cabe destacar que según el Compendio de Normas Contables [1] se considera que un cliente entra en la cartera de incumplimiento cuando muestra una deteriorada o nula capacidad de pago, cuando es necesaria una reestructuración forzosa de sus deudas (disminuyendo o postergando el pago principal o los intereses) y cuando presenta un atraso igual o superior a 90 días en el pago de algún crédito.

Dada esta posibilidad de incumplimiento de los clientes, es clave para las instituciones empresariales y financieras contar con procedimientos que evaluen la calidad crediticia de los clientes y ayuden a la toma de decisiones, ya que una correcta decisión puede aumentar las utilidades y disminuir la posibilidad de sufrir pérdidas financieras importantes al no recuperar el monto prestado, asegurándose de mantener un nivel de provisiones suficiente para sustentarlas.

Una forma de manejar el riesgo de crédito es a través de modelos estadísticos, los cuales a partir de una muestra de clientes clasificados a priori como “buenos” o “malos”, “incumplimiento” o “no incumplimiento”, predicen la probabilidad de que un cliente falle . Entre las técnicas estadísticas que se utilizan para predecir la probabilidad de incumplimiento es posible mencionar el análisis discriminante y la regresión logística, siendo la regresión logística la técnica más utilizada por la industria financiera. Sin embargo, actualmente las técnicas de Machine Learning han sido utilizadas para la construcción de modelos predictivos y para la comprensión de patrones de un determinado segmento de clientes, ya que tienen la ventaja de no tener demasiados requerimientos y supuestos para las variables de entrada, además permiten una reducción significativa en los tiempos de ejecución de los distintos procesos financieros para la medición de la calidad crediticia. [2]

En particular, los métodos de Machine Learning se clasifican en dos tipos de aprendizaje: aprendizaje supervisado y aprendizaje no supervisado. La diferencia fundamental entre estos dos tipos de aprendizaje radica en si los datos entregados al algoritmo están etiquetados o no. Por un lado, los algoritmos de aprendizaje supervisado se utilizan en el área del riesgo de crédito para encontrar la relación entre las características del cliente y el incumplimiento crediticio y luego predecir la clasificación (incumplimiento o no incumplimiento). En cambio, los algoritmos

de aprendizaje no supervisado, que en la mayoría de los casos se refieren a los algoritmos de agrupación en clústeres, se utilizan como una técnica para agrupar los datos en grupos de características similares en lugar de proporcionar predicciones directamente, por lo tanto, estos algoritmos se utilizan a menudo como herramientas complementarias a los algoritmos de aprendizaje supervisado. [3]

1.1. Revisión Bibliográfica

Hoy en día, gracias a la rapidez de procesamiento de grandes volúmenes de datos y la automatización que nos entregan las herramientas de Machine Learning, se han realizado diversos estudios de comparación de algoritmos para predecir el incumplimiento financiero. A continuación se mencionan algunas referencias relevantes:

En [2] se compararon los modelos de regresión logística, árboles de decisión, redes neuronales y support vector machines. En este estudio los resultados obtenidos con técnicas complejas (árboles de decisión y redes neuronales) no fueron sustancialmente superiores a aquellos obtenidos con regresión logística.

En [4] se comparan 8 algoritmos de Machine Learning obteniendo mejor desempeño el algoritmo de ensamble (boosting) heterogéneo. Además se menciona en la literatura relacionada que Yeh y Lien (2009) han realizado la comparación de 6 algoritmos: k-Nearest Neighbors, regresión logística, análisis discriminante, naive Bayesian classifier, redes neuronales y árboles de clasificación, encontrando que las redes neuronales dan los mejores resultados. En tanto, Bellotti y Crook (2009) establecieron que support vector machines tienen mejor desempeño que métodos tradicionales como regresión logística y análisis discriminante.

En [7] se comparan random forest, gradient boosting y redes neuronales, mientras que en [8] se comparan los modelos de regresión logística, random forest, gradient boosting y 4 versiones de redes neuronales. En ambos estudios se encontró que el mejor modelo obtenido es gradient boosting.

Por otro lado, en [9] se aplicaron los algoritmos random forest y extreme gradient boosting (XGBoost) sobre una base de solicitudes de tarjetas de crédito, obteniendo mayor precisión con XGBoost.

1.2. Objetivos

Como podemos observar, los estudios mencionados anteriomente no obtienen el mismo modelo con mejor desempeño, esto se debe a que los algoritmos de Machine Learning dependen en gran medida de los datos con los que se entrenan, por lo que no se puede definir un mejor modelo globalmente.

Es por esto que en colaboración con KPMG se define esta memoria de título, con la finalidad de entrenar algoritmos de Machine Learning como metodologías para el riesgo de crédito en el software estadístico R, proponiendo realizar una calificación de los clientes mediante los siguientes algoritmos de aprendizaje supervisado: Regresión logística, Análisis discriminante lineal, Árboles de decición, Random Forest, Gradient Boosting, Extreme Gradient Boosting y Support Vector Machines.

Por lo tanto, se definen los siguientes objetivos:

1. Estudiar e implementar algoritmos de aprendizaje supervisado de clasificación que nos permitan determinar si un cliente entrará o no en incumplimiento.
2. Comparar los algoritmos de clasificación en base a su bondad de ajuste, con la finalidad de determinar la metodología con mejor rendimiento para predecir el incumplimiento crediticio.

2. Marco Teórico

2.1. Comisión para el Mercado Financiero

La Comisión para el Mercado Financiero (CMF) es un organismo público, regulador y supervisor financiero de Chile, que tiene como objetivo velar por el correcto funcionamiento, desarrollo y estabilidad del mercado financiero. Una de las principales funciones de la CMF es la dictación de normas o instrucciones propias para el mercado y las entidades que lo integran, con el fin de establecer un marco regulador. [10]

En particular, las instrucciones contables vigentes impartidas a los Bancos e instituciones financieras se encuentran en el Compendio de Normas Contables (CNC). De acuerdo a esta normativa, se indica que el riesgo de crédito se debe medir en base a la pérdida esperada a fin de constituir provisiones necesarias y suficientes para cubrirlas. Esta pérdida esperada se calcula mediante la siguiente fórmula:

$$PE = PI \cdot PDI \cdot EAI \tag{2.1}$$

Donde

PE: corresponde a la Pérdida Esperada por riesgo de crédito.
PI: corresponde a la Probabilidad de Incumplimiento.
PDI: corresponde a la Pérdida dado el Incumplimiento.
EAI: corresponde a la Exposición al Incumplimiento.

Específicamente, el estudio y análisis de esta memoria de título se centra en la calidad crediticia de los deudores, dada por la capacidad que tienen para cumplir con sus obligaciones, por lo que se utilizarán distintos métodos, definidos en las secciones siguientes, para obtener la probabilidad de incumplimiento (PI) de los clientes.

Para ello, se debe tener en consideración la siguiente definición de incumplimiento, la cual se encuentra establecida en el Capítulo B1 del CNC: “La Cartera en incumplimiento incluye a los deudores y sus créditos para los cuales se considera remota su recuperación, pues muestran una deteriorada o nula capacidad de pago. Forman parte de esta cartera los deudores que han dejado de pagar a sus acreedores (en default) o con indicios evidentes de que dejarán de

hacerlo, así como también aquellos para los cuales es necesaria una reestructuración forzosa de sus deudas, disminuyendo la obligación o postergando el pago del principal o los intereses y, además, cualquier deudor que presente un atraso igual o superior a 90 días en el pago de intereses o capital de algún crédito". [1]

2.2. Técnicas estadísticas

2.2.1. Regresión logística

La técnica estadística más utilizada por la industria financiera para evaluar la calidad crediticia corresponde a la regresión logística (Thomas et al., 2002). Esta técnica analiza la relación entre un conjunto de variables independientes y una variable dependiente categórica al estimar la probabilidad de ocurrencia de un evento. [12]

Suponiendo que se tiene una muestra de n observaciones independientes de p variables predictoras o independientes $X = (X_1, X_2, ..., X_p)$ y que la variable dependiente o variable de respuesta $Y = (y_1, y_2, ..., y_n)^T$ representa la ocurrencia de un evento, es decir,

$$y_i = \begin{cases} 1, \text{ si el evento ocurre,} \\ 0, \text{ si el evento no ocurre.} \end{cases}$$

El modelo de regresión logística expresa la probabilidad condicional de que el evento ocurra $P(Y = 1 \mid X) = \pi(X)$ de la siguiente manera:

$$\pi(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + ... + \beta_p X_p)}} \tag{2.2}$$

Otra forma de expresar la relación entre las variables predictoras y la variable de respuesta es realizando una transformación en términos de $\pi(x)$ para obtener la siguiente relación lineal en los parámetros $\boldsymbol{\beta}$

$$g(X) = \ln\left[\frac{\pi(X)}{1 - \pi(X)}\right] = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p \tag{2.3}$$

donde $g(X) = \ln\left[\frac{\pi(X)}{1-\pi(X)}\right]$ es conocido como el *logit* del modelo de regresión logística, o también es llamado log-odds ya que es el logaritmo de la razón de problabilidades (odds) $\frac{\pi(X)}{1-\pi(X)}$. [13]

En paticular, para el caso del estudio del riesgo de crédito, las variables predictoras son variables cuantitativas o cualitativas que se consideran desencadenantes del incumplimiento de los clientes, por ejemplo, los días o meses de mora, o el monto pendiente adeudado, por otro lado, la respuesta binaria representa el incumplimiento de un individuo (*default*) [14], como sigue

$$y_i = \begin{cases} 1, \ \textit{default}, \\ 0, \ \text{no } \textit{default}. \end{cases}$$

Por lo tanto, para obtener la probabilidad de incumplimiento del cliente $P(Y = 1 \,|\, X)$ se debe ajustar el modelo de regresión logística (2.2) a un conjunto de datos estimando el valor de $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^T$.

El método utilizado para estimar los parámetros $\boldsymbol{\beta}$ es conocido como máxima verosimilitud y consiste en predecir los parámetros desconocidos maximizando la función de verosimilitud. Ya que se supone que las observaciones son independientes, la función de verosimilitud para el modelo de regresión logística está dada por:

$$\ell(\boldsymbol{\beta}) = \prod_{i=1}^{n} \pi(x)^{y_i} \ [1 - \pi(x_i)]^{1-y_i} \tag{2.4}$$

Matemáticamente, maximizar la verosimilitud es equivalente a maximizar la log verosimilitud definida como

$$L(\boldsymbol{\beta}) = \ln[\ell(\boldsymbol{\beta})] = \sum_{i=1}^{n} \{y_i \ln[\pi(x_i)] + (1 - y_i) \ln[1 - \pi(x_i)]\} \tag{2.5}$$

Luego, para encontrar el valor de $\boldsymbol{\beta}$ que maximiza $L(\boldsymbol{\beta})$ se deriva la ecuación (2.5) con respecto a los $p + 1$ coeficientes y se iguala a 0. Así, la estimación del parámetro $\boldsymbol{\beta}$ se obtiene al resolver las siguientes ecuaciones:

$$\sum_{i=1}^{n} [y_i - \pi(x_i)] = 0 \tag{2.6}$$

$$\sum_{i=1}^{n} x_{ij} [y_i - \pi(x_i)] = 0 \tag{2.7}$$

donde $j = 1, 2, ..., p + 1$.

Estas ecuaciones no son lineales en los parámetros $\boldsymbol{\beta}$, por lo que requieren métodos especiales para su solución. Sin embargo, en general, la regresión logística y otros modelos se pueden ajustar fácilmente utilizando un software estadístico como R. [13]

2.2.1.1. Supuestos

Los supuestos que se aplican a la regresión logística (Bewick Cheek y Ball (2005), Peng y So (2002)) son los siguientes:

- Las observaciones deben ser independientes entre sí.
- El modelo debe presentar poca o nula multicolinealidad, ya que la multicolinealidad entre los predictores puede llevar a estimaciones sesgadas y a errores estándar inflados. [17]
- La relación entre las variables independientes y el logaritmo de las probabilidades (2.3) debe ser lineal.
- La regresión logística requiere tamaños de muestra grandes, ya que, los estimadores por máxima verosimilitud son menos robustos que los estimadores por mínimos cuadrados ordinarios usados para estimar los parámetros de un modelo de regresión lineal.

Otro punto importante a considerar, es que en la regresión logística no es necesario el supuesto distribucional de normalidad, sin embargo, la solución puede ser más estable si los predictores tienen una distribución normal multivariante. [17]

2.2.2. Análisis discriminante lineal

Como se define en la subsección anterior, la variable de respuesta Y puede tomar 2 posibles valores, 1 si el incumplimiento ocurre y 0 si no. La regresión logística implica modelar directamente la probabilidad $P(Y = 1 \,|\, X = x)$ usando la función logística.

El análisis discriminante lineal o LDA por su sigla en inglés (*Linear discriminant analysis*) es una técnica alternativa que nos permite estimar las probabilidades $P(Y = k \,|\, X = x)$, con $k = \{0, 1\}$ y clasificar a los individuos en una de las $K = 2$ clases de acuerdo a estas probabilidades.

Sea π_k la probabilidad a priori de que una observación pertenezca a la $k - ésima$ clase y $f_k(X) = P(X = x \,|\, Y = k)$ la función de densidad de X para una observación que pertenece a la clase k. Aplicando el teorema de Bayes, LDA estima la probabilidad de que la variable de respuesta pertenezca a una de las k clases dado un determinado valor de los predictores como sigue:

$$p_k(x) = P(Y = k \,|\, X = x) = \frac{P(Y = k)P(X = x \,|\, Y = k)}{\sum_{j=0}^{1} P(Y = j)P(X = x \,|\, Y = j)} = \frac{\pi_k f_k(x)}{\sum_{j=0}^{1} \pi_j f_j(x)} \tag{2.8}$$

Esta probabilidad $p_k(x)$ se define como la probabilidad posteriori de que una observación $X = x$ pertenezca a la $k - ésima$ clase. [18]

Entonces, a partir de (2.8) LDA estima las probabilidades de que la variable de respuesta sea 0 y de que la variable de respuesta sea 1, y clasifica una observación dentro del grupo que mayor probabilidad posteriori tenga, para ello se debe realizar una estimación de π_k y $f_k(x)$.

Para estimar la función de densidad $f_k(x)$ se requiere hacer algunas suposiciones, en particular, cuando se tienen múltiples predictores, se asume que $X = (X_1, X_2, ..., X_p)$ sigue una distribución normal multivariante con vector de medias μ y matriz de covarianza Σ, por lo tanto, su función de densidad está dada por

$$f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{2.9}$$

Además, suponiendo que las observaciones para cada clase siguen una distribución normal multivariante $N(\mu_k, \Sigma_k)$ y que las clases tienen igual matriz de covarianza $\Sigma_k = \Sigma$, para todo $k = \{0, 1\}$, se reemplaza la función de densidad para cada una de las clases $f_k(X = x)$ (2.9) en la ecuación (2.8), se aplica el logaritmo y se obtiene el siguiente clasificador de Bayes:

$$\delta_k(x) = x^T \Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T \Sigma^{-1}\mu_k + \log \pi_k \tag{2.10}$$

Luego, debemos estimar los parámetros desconocidos $\mu_1, \mu_2, ..., \mu_k, \pi_1, ..., \pi_k$ y Σ a partir de las siguientes fórmulas:

$$\begin{aligned} \hat{\pi_k} &= \frac{n_k}{n} \\ \hat{\mu_k} &= \frac{1}{n_k}\sum_{i:y_i=k} x_i \\ \hat{\Sigma} &= \frac{1}{n-K}\sum_{k=0}^{1}\sum_{i:y_i=k}(x_i - \hat{\mu_k})(x_i - \hat{\mu_k})^T \end{aligned} \tag{2.11}$$

donde n_k es el número de observaciones de la $k - ésima$ clase $(k = 0, 1)$, n es el número total de observaciones y K es el número de clases, en este caso como Y es dicotómica se tiene que $K = 2$.

Así, para asignar una clase a una observación $X = x$, LDA obtiene una estimación $\hat{\delta_k}(x)$ para cada una de las clases y clasifica de acuerdo a la clase que obtiene el mayor $\hat{\delta_k}(x)$.

Cabe mencionar, que $\delta_k(x)$ en (2.10) es una función lineal de x y se conoce como *función discriminante*, y es por está función que el clasificador LDA recibe el nombre *lineal*, ya que la regla de decisión de LDA depende únicamente de x a través de una combinación lineal de sus elementos. [18][19]

2.2.2.1. Supuestos

De lo anterior se sigue que las condiciones que se deben cumplir para que un modelo de Análisis Discriminante Lineal sea válido son las siguientes:

- Cada predictor distribuye normal en cada una de las clases de la variable respuesta. En el caso de múltiples predictores, las observaciones siguen una distribución normal multivariante en todas las clases.
- La varianza del predictor es igual en todas las clases de la variable respuesta. En el caso de múltiples predictores, la matriz de covarianza es igual en todas las clases. [20]

2.2.2.2. Ventajas y Desventajas

Algunas de las ventajas que presenta LDA versus a la regresión logística son las siguientes:

- Cuando existe una separación sustancial entre las dos clases, las estimaciones de los parámetros para el modelo de regresión logística son sorprendentemente inestables. El método LDA no sufre este problema.
- Si la distribución de los predictores X es aproximadamente normal en cada una de las clases y el tamaño de la muestra es pequeño, entonces LDA es más estable que la regresión logística.

Sin embargo, cuando se trata de un problema de clasificación con solo dos niveles, ambos métodos suelen llegar a resultados similares. [20]

Como desventaja se tiene que LDA no permite trabajar directamente con variables categóricas las cuales son comúnmente utilizadas en problemáticas de riesgo de crédito. La regresión logística es menos restrictiva en este sentido ya que es posible incorporar tanto variables categóricas como variables continuas. [2]

Las técnicas estadísticas, por lo general, tienen una serie de supuestos estadísticos que deben ser cumplidos a cabalidad para que el modelo construido tenga cierta validez, supuestos que rara vez en los problemas reales se cumplen. Es por esta razón que en los últimos años, se han estudiado una serie de técnicas de Machine Learning para la construcción de modelos de riesgo de crédito ya que tienen la ventaja de no tener demasiados requerimientos y supuestos para las variables de entrada, aumentando su validez. [2]

En la siguiente sección presentaremos los métodos de clasificación de Machine Learning utilizados en esta memoria de título, los cuales nos permiten predecir cuando un cliente entrará en incumplimiento, estos son: Árboles de decisión, Random Forest, Gradient Boosting, Extreme Gradient Boosting y Support vector machines.

2.3. Aprendizaje supervisado

2.3.1. Árboles de decisión

Como se puede apreciar en la Figura 2.1, el nombre de este algoritmo se debe a que el árbol de decisión representa un árbol al revés con muchas bifurcaciones, donde las divisiones del árbol representan una serie de decisiones lógicas. [21] Los métodos basados en árboles son conceptualmente simples y poderosos gracias a su interpretación gráfica. [19] Algunos de los conceptos importantes que definen la estructura de un árbol son:

- Nodo raíz: representa a toda la población o muestra.
- Nodo de decisión: cuando un subnodo se divide en subnodos adicionales, se denomina nodo de decisión.
- Nodo hoja (o terminal): los nodos que no se dividen se llaman nodo hoja (o terminal).
- Rama: subsección del árbol.
- Poda: cuando se eliminan subnodos de un nodo de decisión, este proceso se llama poda. [14]

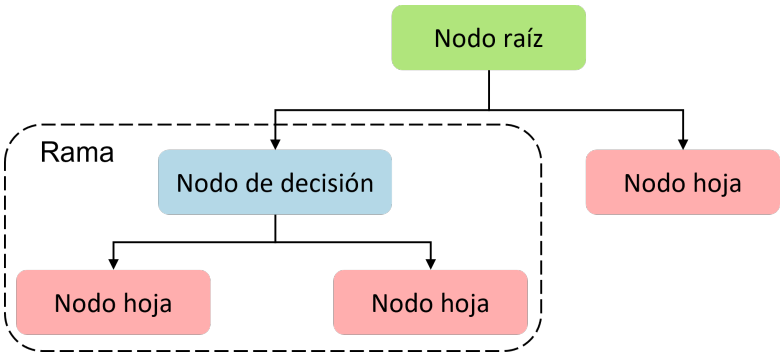


Fig. 2.1: Estructura de un árbol
(Fuente: Elaboración propia)

Los árboles de decisión se pueden aplicar tanto a problemas de regresión como de clasificación. La principal diferencia entre un árbol de regresión y clasificación es el tipo de variable de respuesta. En este caso, cuando se tiene una variable de respuesta cualitativa categórica (incumplimiento o no incumplimiento) se utilizan los árboles de decisión de clasificación, cuyo objetivo es predecir la clase $k = 0$ o $k = 1$ de la variable dependiente.

La idea principal de los árboles de decisión es dividir las observaciones del nodo raíz en conjuntos homogéneos (nodos hojas) mediante reglas binarias a partir de las variables explicatorias más importantes. [21] Es decir, el proceso de construcción de un árbol se puede resumir en dos grandes pasos:

1. División sucesiva del espacio de predictores $X_1, X_2, .., X_p$ en J subconjuntos disjuntos $R_1, R_2, \ldots, R_J$.

2. Predicción para cada observación en la región R_J.

Para construir las regiones $R_1, R_2, ..., R_J$ que corresponden a los nodos terminales, el algoritmo de árboles de decisión utiliza el método de división binaria recursiva (*recursive binary splitting*), este método comienza en el nodo raíz, donde todas las observaciones pertenecen a una misma región, luego, se consideran todos los predictores $X_1, X_2, \ldots, X_p$ y todos los posibles valores de división s para cada uno de ellos, y se selecciona un predictor X_j y un punto de división s de manera de dividir el nodo raíz en dos semiplanos:

$$R_1(j, s) = \{X \mid X_j \leq s\} \quad \text{y} \quad R_2(j, s) = \{X \mid X_j > s\} \tag{2.12}$$

Esta división se visualiza a través de dos nuevas ramas en el árbol. Luego se repite el proceso en cada una de las dos regiones y así sucesivamente se divide el espacio de los predictores. [18]

Cabe destacar que en cada paso del proceso de construcción, se debe elegir la mejor división a realizar. En clasificación, los criterios de selección de la variable X_j y el punto de división s se basan en la ganancia de información o reducción de la impureza [21], es decir, se elige la división que entrega nodos hojas lo más homogéneos o puros posible (un nodo se considera puro cuando todas sus observaciones pertenecen a una misma clase).

Existen 3 índices principales para evaluar la impureza los cuales son: el Error de Clasificación, el Índice de Gini y la Entropía. Definiendo $\hat{p}_{mk}$ como la proporción de observaciones que pertenecen a la k-ésima clase en el m-ésimo nodo, las medidas de impureza se definen como sigue:

$$\begin{aligned} \text{Error} &= 1 - \max_k \hat{p}_{mk} \\ \text{GINI} &= \sum_k \hat{p}_{mk}(1 - \hat{p}_{mk}) \\ \text{Entropía} &= -\sum_k \hat{p}_{mk} \log \hat{p}_{mk} \end{aligned} \tag{2.13}$$

De estos 3 índices, normalmente se utilizan el índice de Gini o la entropía para evaluar la calidad de la división, ya que son más sensibles a la pureza del nodo que el error de clasificación [18]. En la práctica el índice más utilizado corresponde al índice de Gini [14], sin embargo, independientemente de la medida empleada como criterio de selección de las divisiones, el proceso siempre es el mismo:

1. Para cada posible división se calcula el valor de la medida en cada uno de los dos nodos resultantes.

2. Después de que el índice se calcula en cada nodo, el valor total se calcula como el promedio ponderado

$$I_{SPLIT} = \frac{n_1}{n} \cdot I_1 + \frac{n_2}{n} \cdot I_2 \tag{2.14}$$

   donde I_j corresponde al índice de impureza de cada nodo resultante de la división, n_j corresponde al número de observaciones en cada nodo para $j = 1, 2$ y n es el número total de observaciones.

3. Luego, se calcula el índice de Información Ganada (IG), que es la resta de la impureza del nodo padre menos el promedio ponderado (2.14) de las impurezas de los nodos hijos.

$$\Delta = IG = I_{padre} - I_{SPLIT} \tag{2.15}$$

4. Finalmente, se elige como mejor división de los datos aquella con mayor información ganada.

En particular, el proceso de división continúa hasta que se alcance algún criterio de parada, por ejemplo, alcanzar una profundidad máxima, que ninguna región contenga menos de n observaciones o que el árbol tenga un máximo de nodos terminales. De esta manera se construye el árbol eligiendo la mejor división binaria en cada paso y las observaciones quedan agrupadas en las J distintas regiones que corresponden a los nodos terminales. Luego, para realizar una predicción se recorre el árbol en función del valor de los predictores hasta llegar a un nodo terminal, el valor de predicción será la clase más frecuente de las observaciones en aquel nodo. [22]

En la Figura 2.2 se muestra un ejemplo de división binaria recursiva

1. Primero comienza en el nodo raíz y de acuerdo a los criterios de impureza se selecciona el predictor X_1 y el punto de división t_1 como mejor división y se crean las dos regiones

$$\{X_1 \,|\, X \leq t_1\} \quad \text{y} \quad \{X_1 \,|\, X > t_1\}$$

2. Luego, para ambas regiones resultantes de la primera división se repite el proceso de división eligiendo el mejor predictor y punto de división. Para la primera región se elige el predictor X_2 y el punto de división t_2 y se crean las regiones

$$R_1 = \{X \,|\, X_1 \leq t_1 \wedge X_2 \leq t_2\} \quad \text{y} \quad R_2 = \{X \,|\, X_1 \leq t_1 \wedge X_2 > t_2\}$$

   Para la segunda región resultante se elige el predictor X_1 y el punto de división t_3 y se crean las regiones

$$R_3 = \{X \,|\, X_1 > t_1 \wedge X_1 \leq t_3\} \quad \text{y} \quad R_4 = \{X \,|\, X_1 > t_1 \wedge X_1 > t_3\}$$

3. De estas 4 regiones, la región R_4 se divide eligiendo el predictor X_2 y el punto de división t_4 generando las regiones

$$R_4 = \{X \,|\, X_1 > t_1 \wedge X_1 > t_3 \wedge X_2 \leq t_4\} \quad \text{y} \quad R_5 = \{X \,|\, X_1 > t_1 \wedge X_1 > t_3 \wedge X_2 > t_4\}$$

   R_1, R_2, R_3 no se dividen más.

4. El resultado de este proceso es una partición en las cinco regiones que representan los nodos terminales $R_1, R_2, \ldots, R_5$ que se muestran en la Figura.

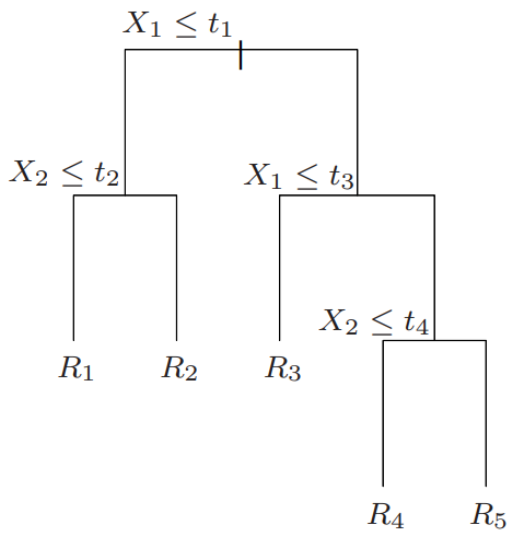


Fig. 2.2: Ejemplo partición de un árbol
(Fuente: T. Hastie, R. Tibshirani, J. Friedman, 2008)

2.3.1.1. Ventajas y Desventajas

Los métodos basados en árboles de decisión tienen una serie de ventajas sobre las técnicas estadísticas clásicas vistas en la sección anterior. Por lo general, requieren mucha menos limpieza y preprocesado de los datos, ya que al tratarse de métodos no paramétricos no es necesario que se cumpla ningún tipo de distribución específica, pueden manejar tanto variables continuas como categóricas sin crear variables *dummy* (ficticias) y pueden imputar los valores faltantes, además, son simples y útiles para la interpretación ya que se pueden representar gráficamente y son capaces de seleccionar predictores de forma automática entregando la importancia de cada uno de ellos. [18][21]

Sin embargo, normalmente no son competitivos en términos de precisión de predicción, esto se debe a su tendencia al sobreajuste y alta varianza, además, un pequeño cambio en los datos puede causar un gran cambio en la estimación final del árbol, por lo que pueden ser muy poco robustos. A menudo combinar una gran cantidad de árboles puede resultar en mejoras dramáticas en la precisión de la predicción. Por lo tanto, en esta sección también presentamos los algoritmos Boosting y Random Forest, los cuales implican producir varios árboles que luego se combinan para producir una sola predicción de consenso. [18]

2.3.2. Random Forest

Como se menciona anteriormente, los árboles de decisión sufren de una alta varianza, dado que al ajustar un árbol a distintos conjuntos de datos los resultados pueden ser bastante diferentes. La agregación *bootstrap* o *bagging* es una técnica para reducir la varianza de un método de aprendizaje estadístico, particularmente útil para procedimientos de alta varianza y bajo sesgo, como lo son los árboles de decisión.

Random Forest es un modelo propuesto por Breiman (2001) que utiliza el método *bagging* para reducir la variación y aumentar la precisión, construyendo un modelo de árboles de decisión para varios conjuntos de entrenamiento tomando repetidas muestras del conjunto de datos de acuerdo al siguiente algoritmo:

Algoritmo: Random Forest para regresión y clasificación.

---

1. Para $b = 1$ hasta B:

   a) Obtener una muestra bootstrap X^* de tamaño N desde la base de entrenamiento.

   b) Se construye un árbol de bosque aleatorio T_b para los datos de la muestra bootstrap, repitiendo recursivamente los siguiente pasos para cada nodo terminal del árbol, hasta alcanzar el tamaño mínimo n_{min} del nodo:

      i. Seleccionar m variables al azar de las p variables.

      ii. Elegir la mejor variable y punto de división de las m variables.

      iii. Dividir el nodo en dos nodos hijos.

2. Salida del conjunto de árboles $\{T_b\}_1^B$.

   La predicción en un nuevo punto x, si el problema es de:

   Regresión: $\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$

   Clasificación: Sea $\hat{C}_b(x)$ la predicción de la clase del $b-$*ésimo* árbol del bosque aleatorio, la predicción del bosque aleatorio es $\hat{C}_{rf}^B(x) =$ voto mayoritario $\{\hat{C}_b(x)\}_1^B$.

Por lo general, Random Forest se estabiliza alrededor de los 200 árboles, donde cada árbol creado se deja crecer hasta su máxima profundidad y no se poda. Específicamente, para la clasificación el valor predeterminado de variables escogidas para realizar la división de los árboles es $m = \sqrt{p}$ y el tamaño mínimo del nodo es 1.

En resumen, cuando Random Forest se utiliza para la clasificación, se toman B muestras aleatorias del conjunto de datos, para cada una de ellas se entrena un árbol, se obtiene una predicción de cada árbol y luego clasifica usando la clase más frecuente (voto mayoritario). [19][18]

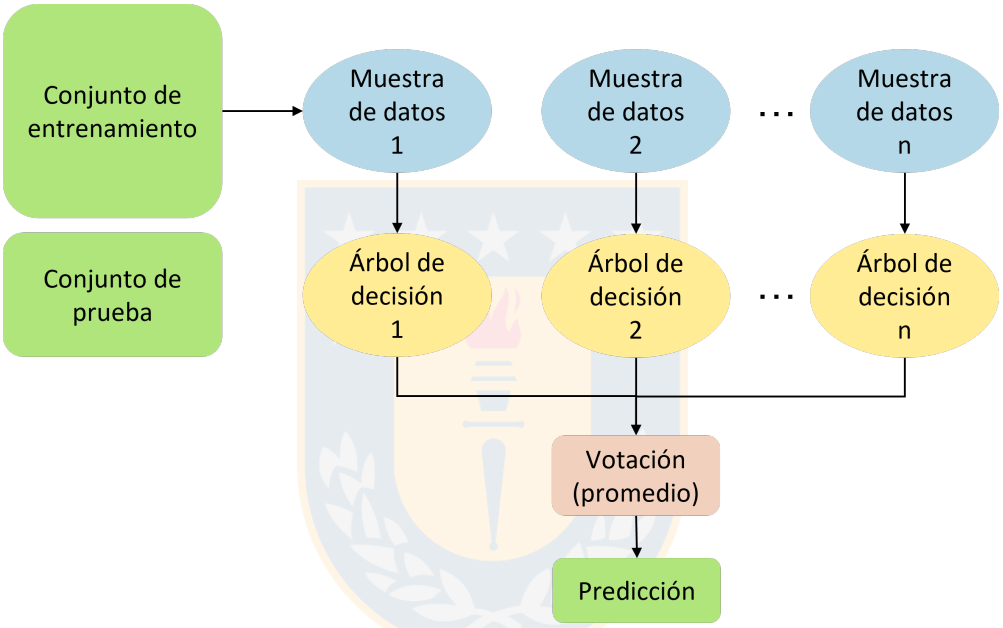


Fig. 2.3: Representación gráfica algoritmo Random Forest
(Fuente: Elaboración propia)

2.3.2.1. Ventajas y Desventajas

Ya que Random Forest está basado en árboles de decisión, presenta las mismas ventajas descritas en la subsección 2.3.1.1, además, mejora la varianza de los árboles de decisión, puede aplicarse a conjuntos de datos con un elevado número de observaciones y estima el error de validación sin necesidad de recurrir a estrategias computacionalmente costosas como la validación cruzada.

Pese a estas múltiples ventajas, se tiene que al combinar múltiples árboles se pierde la interpretabilidad que tiene un único árbol, por lo que se dice que el modelo se comporta como una caja negra para los modeladores estadísticos.

2.3.3. Gradient Boosting

Otro enfoque para mejorar las predicciones resultantes de un árbol de decisión, corresponde al método Boosting, cuya idea general es entrenar árboles de decisión secuencialmente, donde cada nuevo árbol se ajusta en función de los residuos de los árboles anteriores.

El algoritmo de Gradient Boosting propuesto por Friedman (2001) utiliza el descenso del gradiente para minimizar los errores de los residuos. Dado una muestra de entrenamiento el objetivo es encontrar una función $F^*(x)$ tal que el valor esperado de una función de pérdida especificada $L(y, F(x))$ sea minimizado, esto es

$$F^*(x) = \underset{F(x)}{\operatorname{argmin}} E_{y,x} L(y, F(x)) \tag{2.16}$$

A continuación se describe el algoritmo por el cual Gradient Boosting aproxima la función $F^*(x)$ por una expansión aditiva de la forma

$$F^*(x) = \sum_{m=0}^{M} \beta_m h(x; a_m) \tag{2.17}$$

donde cada una de las funciones $h(x; a_m)$ corresponden a un pequeño árbol de regresión llamado "aprendiz débil".

Algoritmo: Gradient Boosting para clasificación.

Entrada: Conjunto $\{(x_i, y_i)\}_{i=1}^n$ y función de pérdida $L(y_i, F(x))$.

1. El modelo inicia con $F_0 = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma)$
2. Para $m = 1$ hasta M:
   - *a)* Calcular $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ para $i = 1, \ldots, n$.
   - *b)* Ajustar un árbol de regresión a los pseudo residuos r_{im} y crear las regiones R_{jm} para $j = 1, \ldots, J_m$.
   - *c)* Para cada región calcular

$$\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$$

   - *d)* Actualizar $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_m 1(x \in R_{jm})$.

Salida: $F_M(x)$.

Para clasificación, la función de pérdida L corresponde a la función de verosimilitud (2.5) negativa, ya que minimizar la función de verosimilitud negativa es equivalente a maximizar la función de verosimilitud, así se tiene que

$$L(y_i, F(x)) = -\sum_{i=1}^{n} y_i \ln(p(x_i)) + (1 - y_i) \ln(1 - p(x_i)) \tag{2.18}$$

donde $p(x_i)$ es la probabilidad $P(y_i = 1 \mid x)$. De la relación entre la probabilidad y el logaritmo de la razón de probabilidades se realiza una transformación a la ecuación (2.18) de modo que sea función del $\log(odds)$, por lo que se tiene que la función de pérdida también puede ser escrita de la siguiente manera

$$L(y_i, F(x)) = -y_i \log(odds) + log(1 + e^{\log(odds)}) \tag{2.19}$$

Así, la predicción inicial F_0 con la que comienza el modelo está dada por el logaritmo de la razón de probabilidades: $\log(odds)$. [25] De aquí se siguen los siguientes pasos:

En el segundo paso del algoritmo de Gradient Boosting, se calcula el pseudo residuo para cada observación, el cual se obtiene al derivar la función de pérdida (2.19) con respecto a $\log(odds)$, así se tiene que los pseudo residuos son la diferencia entre el valor de la variable y la predicción

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} = \left[y_i - \frac{e^{\log(odds)}}{1 + e^{log(odds)}}\right]_{F(x)=F_{m-1}(x)} = y_i - p_{m-1}(x_i) \tag{2.20}$$

donde i corresponde a la $i-ésima$ observación en los datos y m corresponde a la $m-ésima$ iteración. Una vez obtenidos los pseudo residuos, se ajusta un árbol para estimarlos y se calcula el valor de salida para cada hoja

$$\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} -y_i[F_{m-1}(x_i) + \gamma] + log(1 + e^{F_{m-1}(x_i)+\gamma}) \tag{2.21}$$

Para ello, se aproxima la función de pérdida mediante el polinomio de Taylor de orden 2, se deriva con respecto a γ y se iguala a 0, así se tiene que

$$L(y_i, F_{m-1}(x_i) + \gamma) \approx L(y_i, F_{m-1}(x_i) + \gamma) + \frac{\partial L}{\partial F}(y_i, F_{m-1}(x_i))\gamma + \frac{1}{2}\frac{\partial^2 L}{\partial F^2}(y_i, F_{m-1}(x_i))\gamma^2$$

$$\frac{\partial}{\partial \gamma} L(y_i, F_{m-1}(x_i) + \gamma) \approx \frac{\partial L}{\partial F}(y_i, F_{m-1}(x_i)) + \frac{\partial^2 L}{\partial F^2}(y_i, F_{m-1}(x_i))\gamma = 0$$

$$\gamma = \frac{-\frac{\partial L}{\partial F}(y_i, F_{m-1}(x_i))}{\frac{\partial^2 L}{\partial F^2}(y_i, F_{m-1}(x_i))} = \frac{y_i - p_{m-1}(x_i)}{p_{m-1}(x_i)(1 - p_{m-1}(x_i))}$$

Por lo tanto, el residuo predicho por cada hoja del árbol se obtiene a través de la fórmula

$$\gamma_{jm} = \frac{\sum\limits_{x_i \in R_{ij}} y_i - p_{m-1}(x_i)}{\sum\limits_{x_i \in R_{jm}} p_{m-1}(x_i)(1 - p_{m-1}(x_i))} \tag{2.22}$$

[24] [26]

Notar que el algoritmo de Gradient Boosting es susceptible al sobreajuste dado que su objetivo es ir minimizando los residuos iteración tras iteración, para evitar este problema y controlar el aprendizaje del procedimiento, se emplea un valor conocido como tasa de aprendizaje (*learning rate* en inglés) que limita la influencia de cada árbol individual. Por tal motivo la predicción del árbol (2.22) se escala por la tasa de aprendizaje ν en cada iteración y se suma a la predicción anterior.

Específicamente, Friedman (1999) encontró de manera empírica que valores pequeños de la tasa de aprendizaje ($\nu \leq 0,1$) conducen a un error de generalización menor. [27]

Así, el proceso anteriormente expuesto se repite M veces, donde en cada iteración se calculan los residuos, se ajusta un árbol de decisión a los residuos recientemente calculados, luego se calcula la predicción del árbol la cual se suma a la predicción anterior, de manera que la predicción final del modelo está dada por

$$\begin{aligned} &\textit{Predicción inicial} + \textit{Residos predichos}_1 \times \textit{tasa de aprendizaje} \\ &+ \textit{Residuos predichos}_2 \times \textit{tasa de aprendizaje} + \ldots \end{aligned}$$

donde el subíndice del residuo predicho denota el $i - \acute{e}simo$ árbol para $i = 1, 2, \ldots, M$.

Por último, cabe destacar que al utilizar la función de pérdida en función del $log(odds)$, la predicción final del modelo es una predicción logarítmica, por lo que se debe convertir en una probabilidad mediante la función logística.

De este modo, Gradient Boosting estima las probabilidades de que un evento ocurra, y comúnmente utiliza un umbral de probabilidad de 0,5 para tomar decisiones de clasificación, es decir, si la probabilidad final predicha es mayor a 0,5 entonces el evento ocurre, en caso contrario el evento no ocurre. [28]

A continuación, en la Figura 2.4 se observa una representación gráfica del procedimiento del algoritmo.

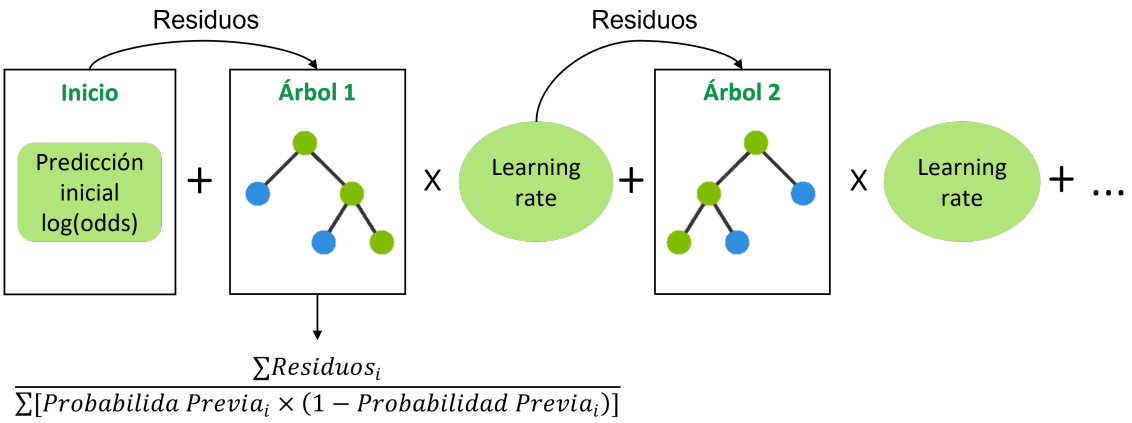


Fig. 2.4: Representación gráfica algoritmo Gradient Boosting
(Fuente: Elaboración propia)

2.3.3.1. Ventajas y Desventajas

Al igual que Random Forest, Gradient Boosting es un algoritmo basado en árboles de decisión, por lo que también presenta las ventajas descritas en la subsección 2.3.1.1, como por ejemplo, imputar valores faltantes e identificar de forma rápida y eficiente las variables predictoras más importantes. Además, es un algoritmo generalizado ya que funciona para diversas funciones de pérdida que sean diferenciables y a menudo proporciona mejores predicciones que otros algoritmos.

Como se menciona anteriormente, una de las mayores ventajas de los modelos de Árbol de decisión es la interpretabilidad, sin embargo, Gradient Boosting carece de esta característica al combinar múltiples árboles de decisión. Una forma de interpretar las aproximaciones de este método es utilizar gráficas de dependencia o de importancia relativa de las variables, que si bien no proporcionan una descripción completa, al menos ofrecen una idea de la naturaleza de la relación entre la variables de entrada y la variable de salida.

Adicionalmente a la pérdida de interpretabilidad, se tiene que dentro de las desventajas de este método se encuentran la sensibilidad a valores atípicos y la tendencia al sobreajuste si el número de árboles es demasiado grande. [24][28]

2.3.4. Extreme Gradient Boosting

Extreme Gradient Boosting es una versión del algoritmo Gradient Boosting desarrollada por Chen y Guestrin (2016), que incluye un término de regularización para la función de pérdida con el objetivo de penalizar la complejidad del modelo y evitar el sobreajuste, por lo tanto, para este algoritmo la función de pérdida está dada por

$$\mathcal{L}(\theta) = \sum_i L(y_i, F(x_i)) + \sum_k \Omega(f_k) \tag{2.23}$$

donde L corresponde al negativo de la función de verosimilitud y Ω es la función de regularización adicional que se define como sigue

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|\omega\|^2 \tag{2.24}$$

donde γ y λ son los parámetros que controlan el grado de regularización y se utilizan en combinación para reducir la sensibilidad del árbol a las observaciones, T representa las hojas del árbol y ω el peso de cada hoja.

Al igual que Gradient Boosting, se utiliza el descenso del gradiente minimizando la función de pérdida a través de la siguiente aproximación de segundo orden

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[g_i F_t(x_i) + \frac{1}{2} h_i F_t^2(x_i) \right] + \Omega(f_t) \tag{2.25}$$

donde $g_i = \frac{\partial L}{\partial F}(y_i, F_{t-1}(x_i))$ y $h_i = \frac{\partial^2 L}{\partial F^2}(y_i, F_{t-1}(x_i))$ son los gradientes de primer y segundo orden.

De aquí se obtiene que los pseudo residuos corresponden a la diferencia entre el valor de la variable de respuesta y la predicción, y el valor de predicción de cada hoja del árbol de decisión puede ser calculado por la siguiente fórmula

$$\omega_{jm} = \frac{\sum\limits_{x_i \in R_{jm}} g_i}{\sum\limits_{x_i \in R_{jm}} h_i + \lambda} = \frac{\sum\limits_{x_i \in R_{ij}} y_i - p_{m-1}(x_i)}{\sum\limits_{x_i \in R_{jm}} p_{m-1}(x_i)(1 - p_{m-1}(x_i)) + \lambda} \tag{2.26}$$

Así, la predicción final del modelo está dado por una función aditiva de árboles de decisiones que ajustan el error de cada árbol anterior como se muestra en la representación gráfica de la Figura 2.5.

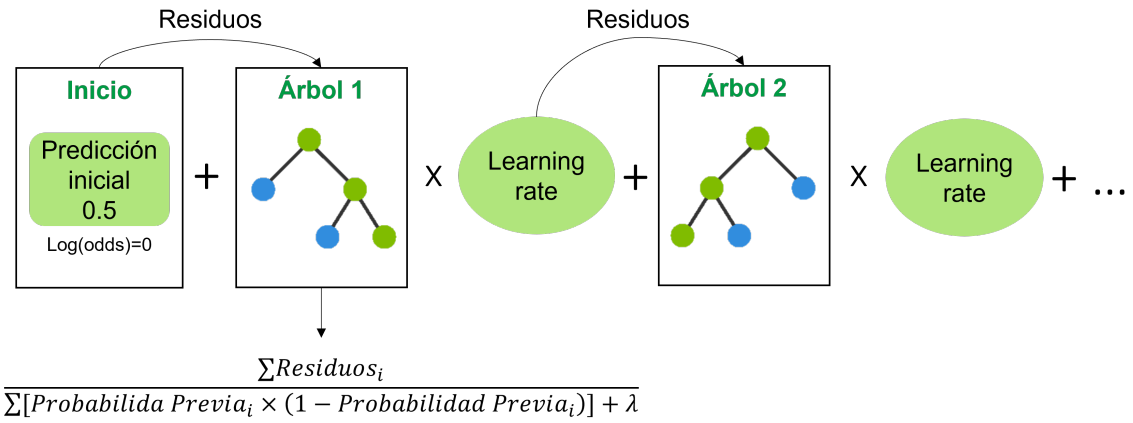


Fig. 2.5: Representación gráfica Extreme Gradient Boosting
(Fuente: Elaboración propia)

Por último, es importante señalar que en este caso, además de la regularización de la función de pérdida, Extreme Gradient Boosting se diferencia en la predicción inicial dada por una probabilidad igual a 0,5 o equivalentemente a $\log(oods) = 0$ y en la construcción del árbol que ajusta a los residuos. Específicamente, Gradient Boosting emplea las medidas estándar para determinar la mejor división en la construcción de árbol (Error, Gini o Entropía), por el contrario, Extreme Gradient Boosting usualmente utiliza la siguiente fórmula para evaluar las divisiones del árbol

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] = \frac{1}{2}\frac{\sum_{x_i\in R_{ij}} (y_i - p_{m-1}(x_i))^2}{\sum_{x_i\in R_{jm}} p_{m-1}(x_i)(1 - p_{m-1}(x_i))} \quad (2.27)$$

[29] [30]

2.3.4.1. Ventajas y Desventajas

Este algoritmo ha demostrado ser un método altamente efectivo para la clasificación de datos [31] y actualmente es uno de los algoritmos más utilizados debido a su éxito incomparable en las competiciones de Kaggle [32].

Las principales ventajas de este algoritmo corresponden al aumento de la precisión y velocidad de ejecución, además el manejo de grandes bases de datos con múltiples variables. Por el contrario, dentro de las principales desventajas se encuentra el ajuste de parámetros del algoritmo que permiten minimizar el error de precisión y evitar sobreajuste del modelo, el gran consumo de recursos computacionales en grandes bases de datos, por lo que se recomienda de-

terminar cuáles son las variables que aportarán más información antes de aplicar esta técnica a grandes bases de datos, por otro lado, este algoritmo solo trabaja con vectores numéricos, por lo que se requiere convertir previamente los datos no numéricos a numéricos. [9]

2.3.5. Support Vector Machines

Hasta el momento se han visto distintas técnicas que permiten clasificar los datos, como lo son regresión logística, análisis discriminante lineal y los modelos basados en árboles de decisión. En esta subsección, se estudiará el algoritmo de Support Vector Machines propuesto inicialmente por Vapnik et al. (1992), el cual establece que es posible construir un hiperplano que separe las observaciones de un conjunto de datos de acuerdo a la etiqueta de sus clases con el mayor margen posible.

Esto es, dado las observaciones de entrenamiento $x_1, \ldots, x_n \in \mathbb{R}^p$ y sus etiquetas asociadas $y_1, \ldots, y_n \in \{-1, 1\}$, donde -1 representa una clase y 1 la otra clase, se define el hiperplano de separación

$$w^T x_i + b = 0, \quad i = \{1, \ldots, n\} \tag{2.28}$$

tal que se satisface lo siguiente

$$\begin{cases} w^T x_i + b > 0, & \text{si } y_i = 1 \\ w^T x_i + b < 0, & \text{si } y_i = -1 \end{cases} \tag{2.29}$$

donde $b \in \mathbb{R}$ y $w \in \mathbb{R}^n$ es el vector perpendicular al hiperplano. De esta manera, el hiperplano puede emplearse a modo de clasificador binario, donde las observaciones que pertenecen a una clase queden por encima y las que quedan por debajo pertenecen a la otra, así el clasificador es de la forma

$$y(x) = \text{sgn}(w^T x + b) \tag{2.30}$$

por lo que el problema de clasificación consiste en encontrar los parámetros b y w. Sin embargo, este problema no tiene solución única ya que existen infitos hiperplanos de separación, en efecto, si b y w son solución entonces λw y λb, con $\lambda > 0$ también son solución.

En particular, para que exista una única solución, Support Vector Machines elige el hiperplano de máximo margen, es decir, aquel que entregue la máxima separación entre los datos correspondientes a cada clase. Para ello se definen los vectores de soporte (*support vectors*) como sigue

$$x_+ = \{x \in \mathbb{R}^n \mid w^T x + b = 1\} \tag{2.31}$$

$$x_- = \{x \in \mathbb{R}^n \mid w^T x + b = -1\} \tag{2.32}$$

Luego, el ancho del margen m es igual a la diferencia entre ambos vectores de soporte, proyectada en la dirección normal del hiperplano, así

$$\begin{aligned} m &= \|\text{proy}_w(x_+ - x_-)\| \\ &= \|x_+ - x_-\| \cos(\theta) \\ &= \|x_+ - x_-\| \left(\frac{w^T(x_+ - x_-)}{\|w\| \cdot \|x_+ - x_-\|} \right) \\ &= \frac{1}{\|w\|} \left((w^T x_+) - (w^T x_-) \right) \\ &= \frac{1}{\|w\|} ((1-b) - (-1-b)) \\ &= \frac{2}{\|w\|} \end{aligned} \tag{2.33}$$

Además, se considera que deben cumplirse las siguientes condiciones

$$y_i = 1 \Leftrightarrow w^T x_i + b \geq 1 \tag{2.34}$$

$$y_i = -1 \Leftrightarrow w^T x_i + b \leq -1 \tag{2.35}$$

Como se muestra en la siguiente Figura

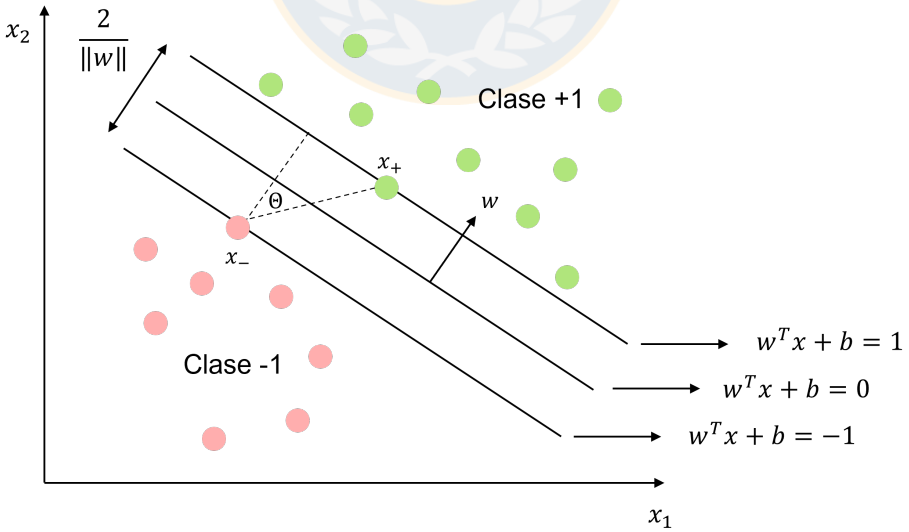


Fig. 2.6: Bosquejo del margen SVM
(Fuente: Elaboración propia)

Por consiguiente, el problema de clasificación se resuelve maximizando el margen m, lo que es equivalente a minimizar la norma $\|w\|$ mediante el siguiente problema de optimización

$$\begin{aligned} \min_{w,b} \quad & \frac{\|w\|^2}{2} \\ \text{s.a} \quad & y_i(w^T x_i + b) \geq 1, \quad i \in \{1, \ldots, n\} \end{aligned} \tag{2.36}$$

Este problema de optimización puede ser transformado a su forma dual por medio del Lagrangiano, el cual está dado por

$$L(w, b, \alpha) = \frac{\|w\|^2}{2} - \sum_{i=1}^{n} \alpha_i \left(y_i(w^T x_i + b) - 1 \right) \tag{2.37}$$

donde $\alpha = (\alpha_1, \ldots, \alpha_n)^T$, $\alpha_i \geq 0$ son los multiplicadores de Langrange o coeficientes de Kühn-Tucker que satisfacen las condiciones

$$\alpha_i(y_i(w^T x_i + b) - 1) = 0, \quad i = 1, 2, \ldots, n. \tag{2.38}$$

Derivando L con respecto a w y b e igualando a 0 se tiene que

$$\frac{\partial L}{w} = w^T - \sum_{i=1}^{n} \alpha_i y_i x_i^T = 0 \Rightarrow \bar{w} = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{2.39}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.40}$$

Luego, sustituyendo (2.39) y (2.40) en (2.37) se obtiene la siguiente formulación dual

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\ s.a \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & \alpha_i \geq 0 \end{aligned} \tag{2.41}$$

Este problema de optimización dual se resuelve mediante técnicas numéricas de programación cuadrática, a través del algoritmo *Sequential Minimal Optimization* y su solución es única, ya que la matriz $y_i y_j \langle x_i, x_j \rangle$ es definida positiva. Además, como el problema de optimización (2.36) es convexo, dado que su función objetivo es cuadrática y las restricciones son lineales para los parámetros w y b, las soluciones del problema primal y dual son equivalentes.

Por lo tanto, una vez que se encuentran los valores óptimos para α a través de la solución del problema dual, se utilizan las ecuaciones (2.39) y (2.38) para obtener las soluciones de w y b. Luego, el clasificador es de la forma

$$\hat{y}(x) = \text{sgn}(\bar{w}^T x + b) = \text{sgn}\left(\left[\sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle \right] + b \right) \tag{2.42}$$

De esta manera, la predicción de la clasificación de un vector depende solo de los vectores de soporte que se encuentran en el margen. [33] [34] [35]

2.3.5.1. Margen Suave

En la mayoría de los casos los datos no se pueden separar exactamente en las dos clases, tal como se muestra en la Figura (2.7), y en consecuencia el problema de optimización descrito anteriormente no tiene solución.

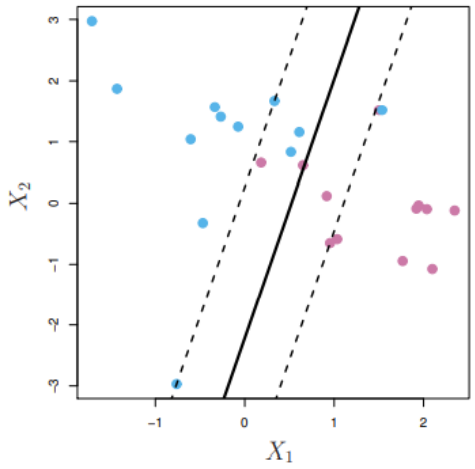


Fig. 2.7: Datos no separables por un hiperplano
(Fuente: G. James et al., 2013)

En esta situación, se debe encontrar un hiperplano de separación utilizando un margen suave, es decir, permitir que algunos puntos de ambas clases queden dentro del margen. Para ello se introducen variables de holgura $\{\xi_i\}_{i=1}^n$ que admiten datos incorrectamente clasificados y un hiperparámetro de regularización c que representa la importancia que se da a la suma de las variables de holgura versus el ancho del margen, de este modo el problema de optimización es el siguiente

$$\begin{aligned} \min_{w,b} \quad & \frac{\|w\|^2}{2} + c\sum_{i=1}^{n}\xi_i \\ \text{s.a} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i \in \{1,\ldots,n\} \\ & \xi_i \geq 0, \quad i \in \{1,\ldots,n\} \end{aligned} \tag{2.43}$$

donde $c > 0$, $\xi_i = 0$ si la observación x_i se encuentra del lado correcto del hiperplano, $0 < \xi_i < 1$ si la observación x_i se encuentra en el margen y del lado correcto y $\xi_i > 1$ si se encuentra al lado incorrecto.

Luego, procediendo de la misma forma que en el caso anterior, el problema dual es el siguiente

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\ s.a \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq c \end{aligned} \tag{2.44}$$

El cual también tiene una única solución y se resuelve con técnicas de programación cuadrática. [34]

2.3.5.2. Truco del Kernel

De acuerdo a lo anteriormente expuesto, la solución del clasificador SVM siempre existe, es única y se puede encontrar, sin embargo, en la práctica es común enfrentarse a datos que no son separables linealmente, tal como se muestra en la Figura 2.8, en estos casos no es posible obtener una solución utilizando las formulaciones anteriores, por lo que para solucionar este problema se utiliza el truco del kernel o *kernel trick*.

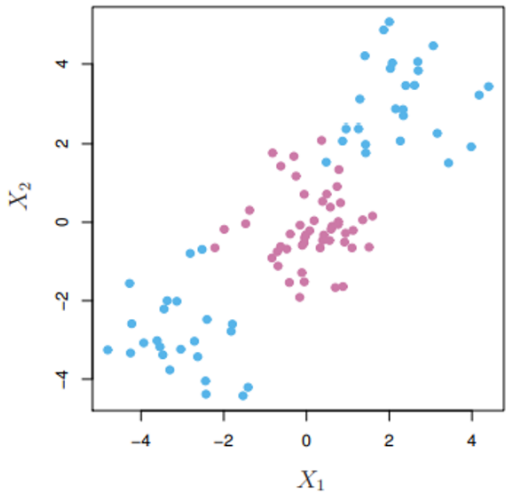


Fig. 2.8: Datos no linealmente separables
(Fuente: G. James et al., 2013)

El truco del kernel consiste en trasladar los datos a un espacio de dimensión mayor donde si sean linealmente separables utilizando una función $\Phi : x \to \phi(x)$ como se muestra en la Figura 2.9. Así, en lugar de aplicar el algoritmo SVM al conjunto de datos original x se aplica a $\phi(x)$.

Esto se logra reemplazando el producto interno $\langle x_i, x_j \rangle$ del problema de optimización por la función kernel K que define el producto interno del mapeo de las características como sigue

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{2.45}$$

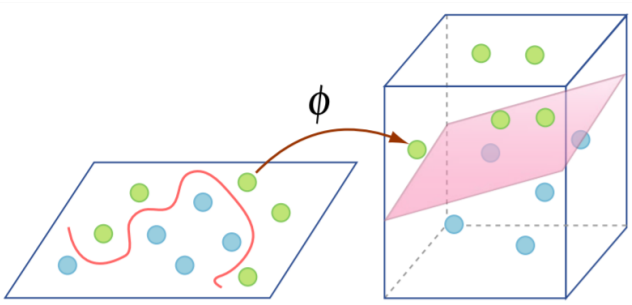


Fig. 2.9: Separación lineal en un espacio de dimensión mayor
(Fuente: MIT 15.097 course)

Por lo tanto, la *kernelización* del SVM tiene una formulación primal dada por

$$\begin{aligned} \min_{w,b} \quad & \frac{\|w\|^2}{2} + c\sum_{i=1}^{n} \xi_i \\ \text{s.a} \quad & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \quad i \in \{1, \ldots, n\} \\ & \xi_i \geq 0, \quad i \in \{1, \ldots, n\} \end{aligned} \tag{2.46}$$

Mientras que su formulación dual tiene la forma

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle \\ s.a \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq c \end{aligned} \tag{2.47}$$

Luego, ocupando el truco del kernel, el problema de optimización dual se convierte en

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ s.a \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq c \end{aligned} \tag{2.48}$$

El cual al igual que el SVM lineal, es un problema de programación cuadrática con solución única, pues el kernel K es definido positivo y por ende el funcional de optimización es cuadrático y cóncavo. [18] [34] [36]

En consecuencia, la función de decisión que clasifica una observación es la siguiente

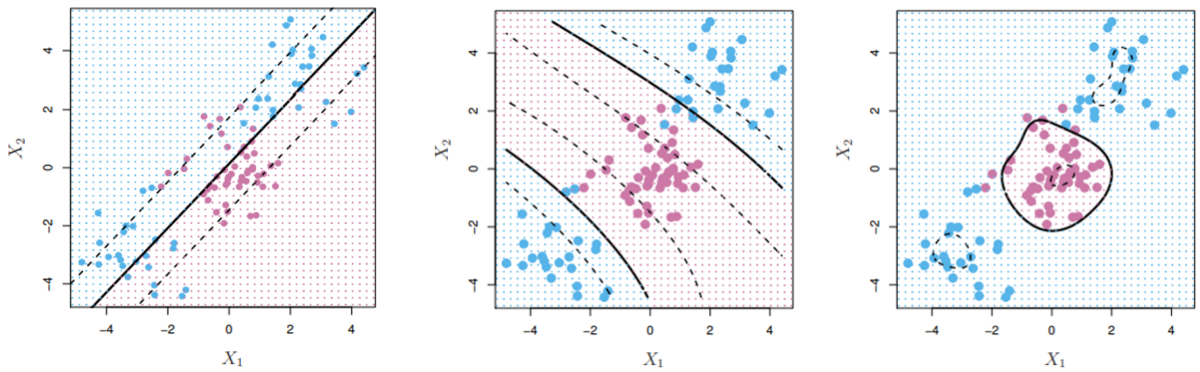
$$\hat{y}(x) = \text{sgn}\left(\left[\sum_{i=1}^{n} \alpha_i y_i K(x_i, x)\right] + b\right) \tag{2.49}$$

En particular, existen distintas funciones kernel para la transformación del espacio vectorial, de las cuales las más conocidas y utilizadas son la función lineal, las funciones polinomiales y la función de base radial (RBF). [37] Estas funciones tienen una gran ventaja, ya que transforman los datos a un espacio de mayor dimensión sin definir explícitamente la función $\phi(x)$.

A continuación, en la Tabla 2.1 se definen los 3 tipos de kernel: lineal, polinomial y radial, mientras que en la Figura 2.10 se puede observar gráficamente la aplicación de estos a un conjunto de datos no separable linealmente.

Kernel	Fórmula
Lineal	K(x,y)=$x \cdot y$
Polinomial	$K(x, y) = (x \cdot y + 1)^d$
Radial	$K(x, y) = \exp(-\gamma\|x - y\|^2)$

Tabla 2.1: Tipos de Kernel más utilizados
(Fuente: G. James et al., 2013)



(a) Kernel Lineal (b) Kernel Polinomial (c) Kernel Radial

Fig. 2.10: Ejemplos de SVM con funciones Kernel
(Fuente: G. James et al., 2013)

2.3.5.3. Ventajas y Desventajas

Debido a que se pueden especificar diferentes funciones kernel para la función de decisión, la cual solo depende de los vectores de soporte, las principales ventajas de Support Vector Machines corresponden a la versatilidad, eficiencia en memoria y efectividad en espacios de alta dimensión. [38] Sin embargo, la elección del Kernel no es trivial y por lo general para seleccionar los parámetros del kernel y el parámetro c, se utiliza una búsqueda mediante validación cruzada. [37]

Adicionalmente, se tiene que este algoritmo no proporciona directamentente una estimación de probabilidad, por lo que para calcularla se realiza una validación cruzada de 5 iteraciones, por lo tanto, la estimación de las probabilidades resulta en un elevado costo computacional y tiempo de ejecución. Esto se convierte en una desventaja importante del algoritmo Support Vector Machines ya que en el caso de la evaluación de la calidad crediticia de los deudores, más que la clasificación se desea obtener la probabilidad de pertenecer a una de las dos clases (clientes buenos o malos). [38] [39]

Por último, cabe destacar que a pesar que el algoritmo Support Vector Machines permite el error de clasificación a través de las variables de holgura, es un algoritmo poco robusto al ruido, es decir, no funciona bien cuando las clases se superponen. [40]

2.4. Selección de variables

Al momento de ajustar los modelos de Machine Learning, el aprendizaje del algoritmo depende exclusivamente de las variables que se le entrega para su entrenamiento. Por lo tanto, es necesario entender qué atributos realmente contribuyen a que el algoritmo aprenda a desarrollar la tarea de la mejor manera posible para así mejorar el rendimiento de los modelos.

Existen diferentes enfoques en la selección de variables útiles para el aprendizaje del algoritmo:

1. El primero es basado en conocimiento experto, es decir, un experto en el problema puede guiar la búsqueda óptima de las variables necesarias a incorporar.

2. El segundo enfoque está basado en aplicar algunos test estadísticos que indican si una determinada variable contribuye a aprender sobre el problema o no.

Actualmente se adoptan las siguientes métricas para identificar variables con alto poder discriminatorio: área bajo la curva (AUC), índice de Gini y valor de información (IV). Sin embargo, en la práctica los analistas de crédito a menudo se basan en umbrales de IV para el proceso de inclusión o exclusión de variables para el modelo. Específicamente, una variable con poder de predicción débil se asocia con $IV < 0,10$, el poder discriminatorio medio se asocia comúnmente con $0{,}1 < IV < 0{,}4$, mientras que $IV > 0{,}4$ denota un fuerte poder de predicción. Por lo tanto, se pueden excluir todas las variables con $IV < 0,10$. [14]

Matemáticamente, el valor de información (IV) se define de la siguiente manera

$$IV = \sum_{j=1}^{J} (DistributionGood_j - DistributionBad_j) \cdot \ln\left(\frac{DistributionGood_j}{DistributionBad_j}\right)$$

donde $DistributionGood_j$ es la proporción de buenos (cuentas no incumplidas) sobre el número total de buenos, mientras que $DistributionBad_j$ es la proporción de malos (cuentas en incumplimiento) sobre el número total de malos, además, cabe señalar que el segundo término de la ecuación se denota como peso de la evidencia (WOE).

3. Un tercer enfoque para la selección de variables tiene que ver con algoritmos que, dentro de su proceso de aprendizaje, realizan un proceso de selección de atributos para encontrar la combinación de atributos que mejor le permiten aprender la tarea, no obstante, esto último no es válido para todos los algoritmos de aprendizaje. [41]

2.5. Criterios de evaluación de modelos

Tanto para técnicas estadísticas como para las técnicas de Machine Learning, el modelo se evalúa en términos de su capacidad predictiva en la discriminación entre clientes buenos y malos, obteniendo la comparación entre la clase predicha por el modelo respecto a la clase real definida a priori.

Para ello, luego de seleccionar las variables que se utilizarán para modelar, el conjunto de datos disponible se debe separar en un conjunto de datos de entrenamiento y de prueba, ya que se necesita disponer de un conjunto de observaciones de las que se conozca la variable respuesta, pero que el modelo no haya “visto”, para saber si el modelo realmente aprendió a desarrollar la tarea que se buscaba aprender, o simplemente “memorizó” los patrones en los datos que se utilizaron para el entrenamiento.

En particular, el tamaño adecuado de las particiones depende en gran medida de la cantidad de datos disponibles y la seguridad que se necesite en la estimación del error, aunque normalmente un proporción de 80 % - 20 % suele dar buenos resultados. [42].

Por lo tanto, la validación de un modelo se conoce como el proceso en el que un modelo entrenado se evalúa con un conjunto de datos de prueba. A continuación, se proporcionan detalles sobre técnicas y métricas utilizadas para la evaluación del desempeño de los métodos de clasificación.

2.5.1. Matriz de confusión

La matriz de confusión que se presenta en la Tabla 2.2 es una matriz de dimensión 2×2 que compara las predicciones del modelo versus la clase real a la que pertenecen los individuos de los datos de prueba.

		Predicción	
		Positivo	Negativo
Observación	Positivo	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativo	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Tabla 2.2: Matriz de confusión
(Fuente: Elaboración propia)

Cada columna de la matriz representa el número de predicciones obtenidas por el modelo, mientras que cada fila representa la clase real. En la diagonal principal se pueden observar los individuos clasificados correctamente (verdaderos positivos y negativos), mientras que la diagonal secundaria nos indica los errores de la clasificación (falsos positivos y negativos), estos 4 posibles resultados se definen a continuación:

- Verdaderos Positivos (VP): Número de observaciones que se clasificaron correctamente como "positivos".
- Verdaderos Negativos (VN): Número de observaciones que se clasificaron correctamente como "negativos".

- Falsos Positivos (FP): También conocido como error tipo I, es el número de observaciones que se clasificaron incorrectamente como "positivos".
- Falsos Negativos (FN): También conocido como error tipo II, es el número de observaciones que se clasificacion incorrectamente como "negativos".

2.5.2. Métricas de evaluación

Las métricas para evaluar el rendimiento y la calidad de los modelos derivan de los valores de la matriz de confusión. A continuación, se presentan sus definiciones y fórmulas:

1. Exactitud: Proporción de predicciones correctas.

$$Exactitud = \frac{VP + VN}{Total} = \frac{VP + VN}{VP + FP + FN + VN}$$

2. Tasa de Error: Proporción de observaciones clasificadas incorrectamente.

$$\text{Tasa de Error} = 1 - Exactitud = \frac{FP + FN}{Total}$$

3. Sensibilidad: También conocido como tasa de verdaderos positivos, es la proporción de casos positivos que fueron correctamente identificados.

$$Sensibilidad = \frac{VP}{\text{Total positivos}} = \frac{VP}{VP + FN}$$

4. Especificidad: También conocido como tasa de verdaderos negativos, es la proporción de casos negativos correctamente identificados.

$$Especificidad = \frac{VN}{\text{Total negativos}} = \frac{VN}{VN + FP}$$

5. Tasa de Falsos positivos: También conocido como Error tipo I, es la probabilidad de que se dé un resultado positivo cuando el valor verdadero es negativo.

$$TFP = \text{Error Tipo I} = 1 - especificidad = \frac{FP}{\text{Total negativos}} = \frac{FP}{VN + FP}$$

6. Tasa de Falsos negativos: También conocido como Error tipo II, es la probabilidad de que la prueba pase por alto un verdadero positivo, es decir, que se dé un resultado negativo cuando el verdadero valor es positivo.

$$TFN = \text{Error tipo II} = 1 - sensibilidad = \frac{FN}{\text{Total positivos}} = \frac{FN}{VP + FN}$$

2.5.3. Curva de ROC

Antes de explicar qué es una curva de ROC, cabe destacar que al realizar una clasificación de individuos a partir de la probabilidad de que pertenezca a una clase, es necesario establecer un umbral de discriminación conocido como punto de corte, es decir, es necesario determinar el nivel de probabilidad a partir del cual un individuo pertenece a una de las clases. En general, el valor predeterminado de los algoritmos de clasificación es $0,5$, esto quiere decir que cuando la probabilidad predicha sea mayor o igual a $0,5$ el individuo será clasificado en la clase 1 y cuando la probabilidad sea menor a $0,5$ será clasificado en la clase 0. Por lo tanto, la predicción de la clasificación que realice el algoritmo depende del valor del punto de corte, y en consecuencia, se obtendrán matrices de confusión y métricas distintas para cada valor posible.

Dicho esto, la curva de ROC (*Receirver Operating Characteristic* en inglés) es una representación gráfica, donde el eje Y corresponde a la tasa de vedaderos positivos (*sensibilidad*) y el eje X corresponde a la tasa de falsos positivos ($1 - especificidad$) de cada uno de los posibles puntos de corte. [43]

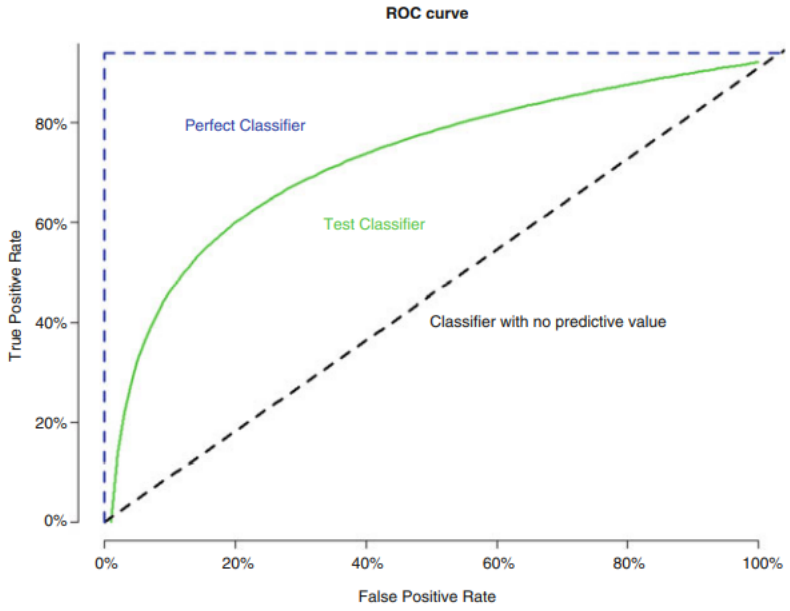


Fig. 2.11: Ejemplos Curvas de ROC
(Fuente: Ivo D. Dinov, 2018)

En el gráfico de la Figura 2.11 podemos observar 3 curvas, la curva de color azul representa una clasificación perfecta donde se tiene $0\,\%$ de falsos positivos y $100\,\%$ de verdedaros positivos, la curva de color verde representa un ejemplo de curva de ROC para un modelo de clasificación, y la diagonal de referencia de color negro, también llamada línea de no-discriminación, describe

lo que sería la curva ROC de un modelo incapaz de discriminar entre las clases positivas y negativas debido a que cada punto de corte que la compone determina la misma proporción de verdaderos positivos y de falsos positivos [21], esto quiere decir, que el modelo realiza una clasificación aleatoria análogo al lanzamiento de una moneda.

Del gráfico de estas 3 curvas de ROC presentadas se observa cómo esta representación gráfica es útil para determinar el mejor modelo predictivo de acuerdo a su capacidad para distinguir entre clases. La manera de cuantificar este rendimiento de las curvas de ROC es a través del área bajo la curva (AUC, *area under the curve*):

$$AUC = \frac{sensibilidad - (1 - especificidad) + 1}{2}$$

Este valor refleja que tan bueno es el modelo para discriminar entre clases. Notando que el área de un clasificador perfecto sería 1 y el área de un clasificador sin valor predictivo sería 0.5, a medida que el AUC se acerque al valor 1 mayor será la capacidad discriminativa del modelo, por el contrario si la curva de ROC coincide con la diagonal de referencia entonces el modelo se considera no discriminativo.

En la siguiente Tabla, se muestra la valoración del modelo de acuerdo el área bajo las curvas de ROC.

AUC	**Desempeño**
$0,5 - 0,6$	Sin discriminación
$0,6 - 0,7$	Malo
$0,7 - 0,8$	Regular
$0,8 - 0,9$	Bueno
$0,9 - 1,0$	Excelente

Tabla 2.3: Valoración AUC
(Fuente: Ivo D. Dinov, 2018)

2.5.4. Índice KS

Otro índice ampliamente utilizado para determinar la eficacia de la capacidad discriminativa de los modelos es el índice KS, el cual corresponde a la máxima diferencia absoluta entre

la distribución acumulada de dos grupos diferentes. En este caso, al estar modelando la PI (probabilidad de incumplimiento) los grupos corresponden a los clientes en incumplimiento y los clientes sin incumplimiento, por lo tanto el índice KS se puede definir como sigue

$$KS = \text{máx}|\,\%Buenos\ acumulados - \%Malos\ acumulados| \tag{2.50}$$

En general, como referencia se considera que un KS mayor a 40 % indica un poder de discriminación razonable en un modelo multivariado ya que cuanto mayor sea la diferencia de las entre los dos grupos, mayor será la capacidad discriminante del modelo. [45]

A continuación, se presenta la valoración del modelo de acuerdo a los valores del índice KS.

KS	Desempeño
$< 20\,\%$	Malo
$20\,\% - 40\,\%$	Regular
$41\,\% - 50\,\%$	Bueno
$51\,\% - 60\,\%$	Muy Bueno
$61\,\% - 75\,\%$	Excelente
$> 75\,\%$	Sospechoso

Tabla 2.4: Valoración KS
(Fuente: Elizabeth Mays, 2001)

2.5.5. Índice de Youden

El índice de Youden refleja la diferencia entre la tasa de verdaderos positivos y falsos positivos y se define como

$$J = Sensibilidad + Especificidad - 1 \tag{2.51}$$

Los valores posibles para el índice J pertenecen al intervalo $[0, 1]$, donde valores cercanos a 0 corresponden a un modelo con poca capacidad discriminatoria y valores cercanos a 1 indicarían que no hay falsos positivos por lo que todas las observaciones son clasificadas correctamente. Por lo tanto, para obtener un buen modelo es deseable obtener un alto valor del índice de Youden.

Dado lo anterior, el índice de Youden es utilizado conjuntamente con la curva de ROC para seleccionar el valor del punto de corte óptimo, es decir, para cada punto de corte posible de la

curva de ROC se calcula el índice de Youden y se selecciona aquel que presente el mayor índice, determinando la sensibilidad y especificidad más alta **en conjunto** de la curva.

Gráficamente, el punto de corte identificado por el índice de Youden corresponde al punto de la curva más cercano al ángulo superior izquierdo del gráfico, es decir, más cercano al punto del gráfico cuya sensibilidad = 100 % y especificidad = 100 %. [43] [44]

A continuación, se muestra un ejemplo donde el punto de corte óptimo determinado por el índice de Youden es 0,551, de esta manera al clasificar de acuerdo a este punto de corte se obtiene la sensibilidad y especificidad más alta para el modelo ($0,849$ y $0,854$ respectivamente).

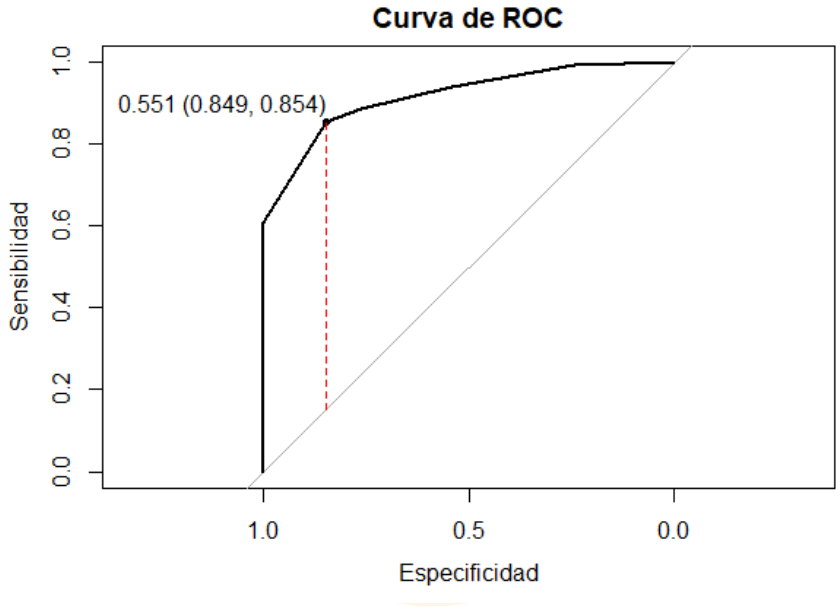


Fig. 2.12: Ejemplo punto de corte óptimo determinado por el índice de Youden (Fuente: elaboración propia)

2.6. Sobreajuste

Al utilizar las métricas de evaluación recientemente descritas se puede detectar cuando el modelo entrenado se ajusta exactamente a los datos de entrenamiento perdiendo la generalidad de la estimación, es decir, se puede detectar cuando el algoritmo logra predecir con precisión los datos de entrenamiento pero al aplicarlo a nuevos datos no es capaz de predecir los resultados a partir de lo aprendido y presenta rendimientos relativamente bajos para los datos de validación, este fenómeno se conoce como *sobreajuste*.

Dado que la generalización del modelo a nuevos datos es lo que permite realizar predicciones, el sobreajuste es un fenómeno que se debe monitorear a partir de la diferencia que existe entre el error en el conjunto de entrenamiento y validación, ya que una fuerte señal de sobreajuste es cuando el error de entrenamiento es bajo y el error en la validación es muy alto. [41]

Por lo general, los procedimientos recomendados para evitar el sobreajuste corresponden a la regularización y optimización de hiperparámetros y la validación cruzada. Sin embargo, las desventajas de utilizar estos procedimientos son el alto costo computacional y tiempo de ejecución, pues el modelo se debe entrenar repetidas veces.

3. Análisis de datos

En este capítulo, se detalla el preprocesamiento de los datos llevado a cabo para posteriormente entrenar los métodos de clasificación definidos en el capítulo anterior, con el objetivo de obtener la probabilidad de incumplimiento de los deudores y así discriminar entre clientes buenos y malos. El análisis de datos y posterior ajuste y validación de modelos fue realizado en el software estadístico R Studio versión 4.0.5.

3.1. Base de datos

Un problema de los modelos de calificación crediticia que debe enfatizarse es la falta de disponibilidad de datos crediticios del mundo real, ya que los datos crediticios de los clientes son confidenciales en la mayoría de las instituciones financieras. [3] Es por esta razón que para implementar los algoritmos se utilizó un subconjunto de la base de datos del libro *IFRS 9 and CECL Credit Risk Modelling and Validation* [14] que contiene la información de la cartera de $n = 18135$ clientes descrita por las siguientes variables:

1. DICOM: Puntaje entregado por la empresa DICOM que indica si una persona tiene un buen crédito (mientras mayor sea el puntaje menor es el riesgo).
2. Monto moroso: Monto en millones de pesos con 30 o más días de morosidad.
3. Utilización TC: Ratio entre el monto utilizado y el cupo otorgado de la tarjeta de crédito.
4. Renta anual: Valor en dólares de la renta líquida anual de cada cliente.
5. Meses mora: Cantidad de meses que han pasado desde la última morosidad del cliente.
6. Default: Incumplimiento del cliente (90 o más días de atraso).

En este caso la variable Default codifica el evento del incumplimiento del cliente y se define como variable dependiente $Y = (y_1, y_1, \ldots, y_n)^T$ tal que

$$y_i = \begin{cases} 1, & \text{si el i-ésimo cliente entra en incumplimiento,} \\ 0, & \text{en otro caso.} \end{cases}$$

Las variables DICOM, Monto moroso, Utilización TC, Renta anual y Meses mora se definen como variables independientes X_1, X_2,...,X_5 respectivamente.

A continuación se presenta una tabla resumen de las cinco variables independientes, donde se visualizan algunas medidas de tendencia, dispersión, máximos y mínimos, así como los datos faltantes para cada variable y su respectivo porcentaje.

Variable	Datos faltantes	Media	Desv. estándar	Mín.	Q_1	Q_2	Q_3	Máx.
DICOM	286 (1,6 %)	391,1	110,8	-125,0	306,0	403,0	484,0	702,0
Monto moroso	0	0,2	0,5	0,0	0,0	0,0	0,0	3,0
Utilización TC	0	0,5	0,2	0,2	0,4	0,5	0,6	1,3
Renta anual	0	63691,1	16655,2	20053,0	52271,5	63601,0	75118,0	135140,0
Meses mora	0	9,0	3,9	2,0	6,0	9,0	12,0	20,0

Tabla 3.1: Estadísticos descriptivos
(Fuente: Elaboración propia)

3.1.1. Datos Faltantes

Los datos faltantes es un problema común al que se enfrentan casi todas las instituciones financieras, esto se debe a que es posible que algunos datos no se puedan recopilar, que el sistema de recopilación de datos se haya alterado o que los clientes no envíen algunos elementos opcionales cuando completan formularios. [3] Cuando el conjunto de datos contiene valores ausentes, se puede realizar lo siguiente:

1. Eliminar las observaciones que contengan valores ausentes.
2. Eliminar aquellas variables que contengan valores ausentes.
3. Estimar los valores ausentes empleando el resto de información disponible (imputación).

Las primeras dos opciones implican perder información, por lo que en este caso se estimarán los datos faltantes a través de la imputación múltiple.

Por lo general, la imputación múltiple es la alternativa más favorecida por la literatura sobre datos faltantes. Esta técnica consiste en crear $m > 1$ versiones completas del conjunto de datos, lo que da lugar a m estimaciones que se agrupan en una estimación final para reemplazar los datos faltantes. [46][47]

Una de las ventajas de la imputación múltiple, es que un número pequeño de imputaciones (entre 3 y 5) puede proporcionar buenos resultados. [48] Además, es posible concluir que la imputación resultó adecuada comprobando que no se generaron cambios relevantes en las características estadísticas de la variable de análisis y en la forma de su distribución. [49]

En paticular, de la Tabla 3.1 se observa que la única variable que consta de valores faltantes es la variable DICOM. Específicamente, se tienen 286 valores faltantes que corresponden a un $1,86\,\%$ del total de la base, de los cuales 272 corresponden a clientes clasificados como buenos ($y = 0$) y 14 corresponden a clientes clasificados como malos ($y = 1$).

Variable	% Datos faltantes	% Clientes buenos	% Clientes malos
DICOM	1,86 %	1,5 %	0,08 %

Tabla 3.2: Porcentaje de datos faltantes
(Fuente: Elaboración propia)

A continuación, se muestran los resultados obtenidos de la estimación de los valores faltantes para la variable DICOM a través de imputación múltiple, para la cual se utilizó la función `mice()` con el método y número de imputaciones predeterminados, es decir, se realizaron $m = 5$ imputaciones a través del método *pmm*: predictive mean matching.

En la tabla 3.3 se observa que los estadísticos descriptivos de la variable DICOM al ser imputada se mantienen constantes, excepto por la media que varía de $391,1$ a $391,0$, además, se observa gráficamente en la Figura 3.1 que no hay cambios relevantes en la distribución de la variable, por lo tanto, se concluye que la imputación que se ha realizado es adecuada.

Variable	Media	Mín.	Q_1	Mediana	Q_3	Máx.
DICOM original	391,1	-125,0	306,0	403,0	484,0	702,0
DICOM imputada	391,0	-125,0	306,0	403,0	484,0	702,0

Tabla 3.3: Estadísticos descriptivos variable DICOM
(Fuente: Elaboración propia)

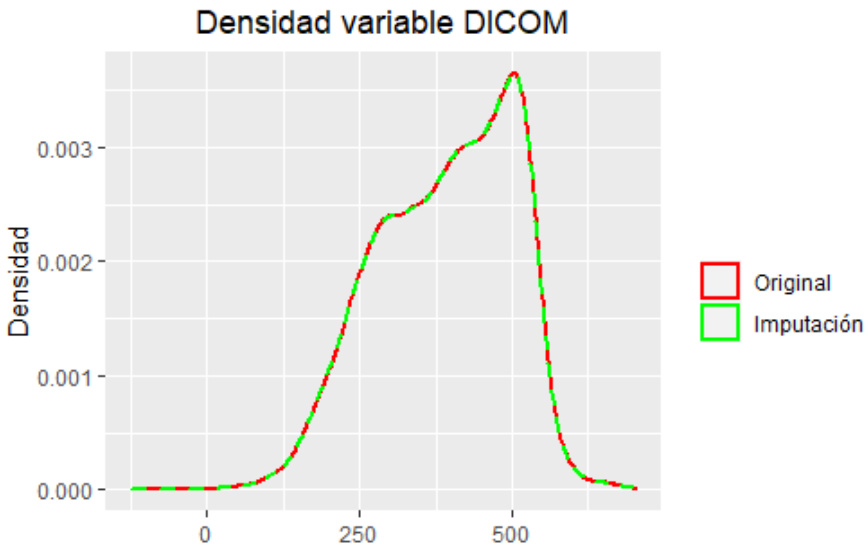


Fig. 3.1: Gráfico de densidad de la variable DICOM
(Fuente: Elaboración propia)

3.1.2. Poder predictivo de las variables

Luego de estimar los datos faltantes, se procede a analizar el poder predictivo de cada una de las variables con el fin de identificar su importancia en la contribución del aprendizaje de los algoritmos, para ello se calcula el valor de información, también conocido como *information value* (IV).

En este caso, dado que las variables independientes son todas continuas y numéricas, los pasos para calcular el IV son los siguientes:

1. Se realiza una discretización de la variable.
2. Se calcula el número de observaciones de buenos (y=1) y malos (y=0) que caen en cada grupo de la discretización.
3. Se calcula el porcentaje de la distribución de buenos y malos en cada grupo de la discretización.
4. Se calcula el IV mediante la fórmula definida en la sección 2.4 del Marco Teórico.

En particular, se utiliza el paquete de R llamado `smbinning`, el cual realiza las agrupaciones óptimas a través de árboles de decisión y calcula los valores de WOE e IV.

De esta manera, se obtienen los siguientes valores de información para cada variable:

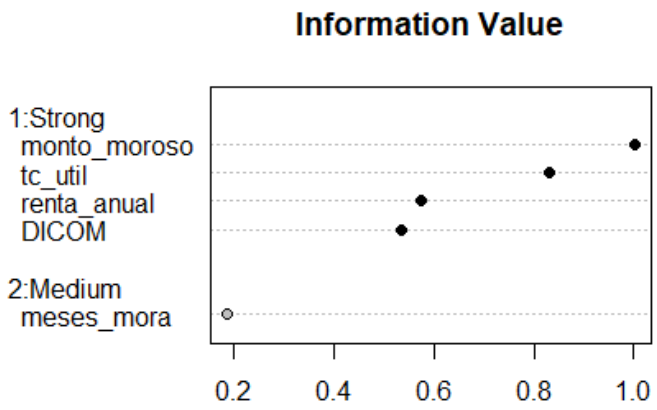


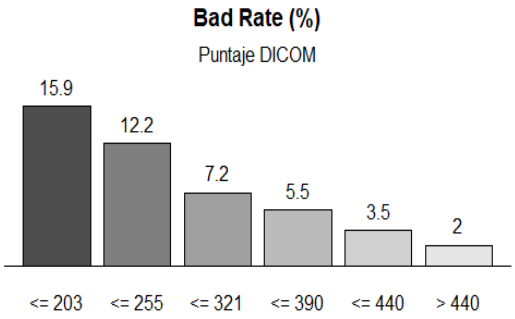
Fig. 3.2: Gráfico IV de las variables independientes
(Fuente: Elaboración propia)

Variable	IV
DICOM	0,54
Monto moroso	1,00
Utilización TC	0,83
Renta anual	0,58
Meses mora	0,18

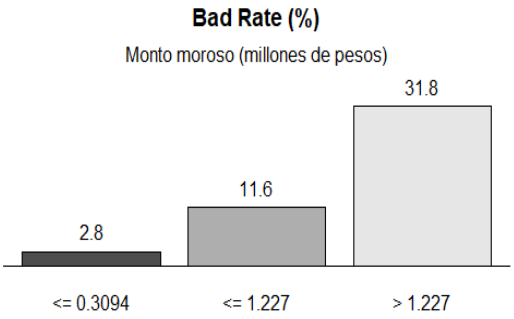
Tabla 3.4: Information Value de las variables independientes
(Fuente: Elaboración propia)

De la Figura 3.2 y la Tabla 3.4 se observa que todas las variables poseen un IV$> 0,1$, por lo tanto, no se excluye ninguna variable. Específicamente, la variable con menor poder de predicción corresponde a la variable Meses de mora con un nivel de discriminancia medio, mientras que las demás variables presentan un fuerte poder de predicción.

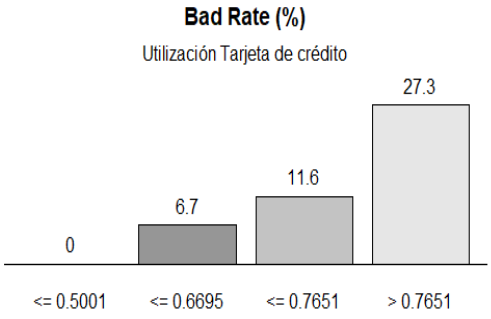
Notar que estos valores de IV son obtenidos a través de las agrupaciones óptimas de cada variable, las cuales también son un factor importante en el análisis ya que cuando los datos se discretizan, las agrupaciones deben tener un orden lógico y tener sentido de riesgo. A continuación, se presentan los gráficos de la tasa de incumplimiento de la cartera de clientes para las distintas agrupaciones:



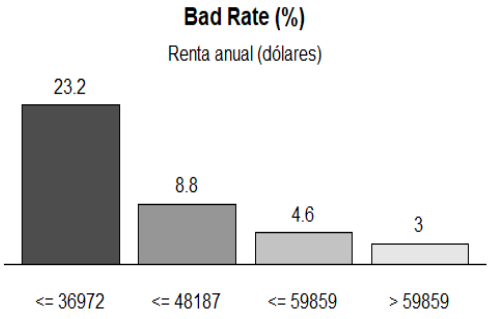
(a)



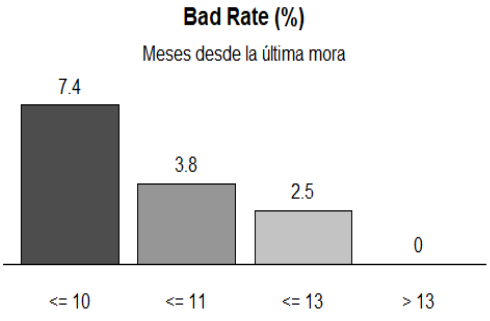
(b)



(c)



(d)



(e)

Fig. 3.3: Tasas de incumplimiento variables independientes
(Fuente: Elaboración propia)

En este caso, se tiene que todas las variables siguen un sentido lógico de riesgo, ya que tienen una relación monótona creciente (o decreciente) con el incumplimiento. En efecto, al observar la Figura 3.3 se sigue que:

a) A mayor puntaje DICOM la tasa de incumplimiento de los clientes disminuye, lo cual concuerda con la definición de la variable presentada en la sección 3.1 (a mayor puntaje menor riesgo).

b) A mayor monto moroso la tasa de incumplimiento aumenta.

c) Mientras mayor sea el ratio entre el monto utilizado y el cupo otorgado de la tarjeta de crédito, mayor es la tasa de incumplimiento.

d) A mayor renta anual la tasa de de incumplimiento disminuye.

e) Mientras mayor sea la cantidad de meses desde que el cliente dejo de incumplir, menor es la tasa de incumplimiento.

Otra herramienta útil para el análisis de importancia de las variables es la representación gráfica de la distribución en función del incumplimiento de los clientes mediante el uso de boxplots, ya que visualizar las diferencias entre los cuartiles de los distintos valores de la variable dependiente ayuda a tener una idea de qué variables pueden ser buenos predictores.

A continuación, en la Figura 3.4 se presentan los boxplot de cada variable independiente, donde se puede observar que la mayor diferencia entre los cuartiles de clientes buenos y malos se obtiene para la variable Monto moroso, seguida de Utilización TC, lo cual coincide con los valores de información más altos obtenidos en la Tabla 3.4. En general, se observa que las 5 variables independientes logran discriminar entre clientes buenos y clientes malos, por lo que, complementado a los análisis anteriores se consideran como buenos predictores para el problema de clasificación.

De igual manera, los boxplot complementan el análisis del sentido lógico de las variables. En efecto, se observa que los clientes que caen en incumplimiento presentan un menor puntaje DICOM, mayores montos morosos, mayor ratio ocupado de la tarjeta de crédito, presentan una renta líquida anual menor y han pasado menos meses desde la última mora, es decir, llevan menos meses cumpliendo con sus pagos.

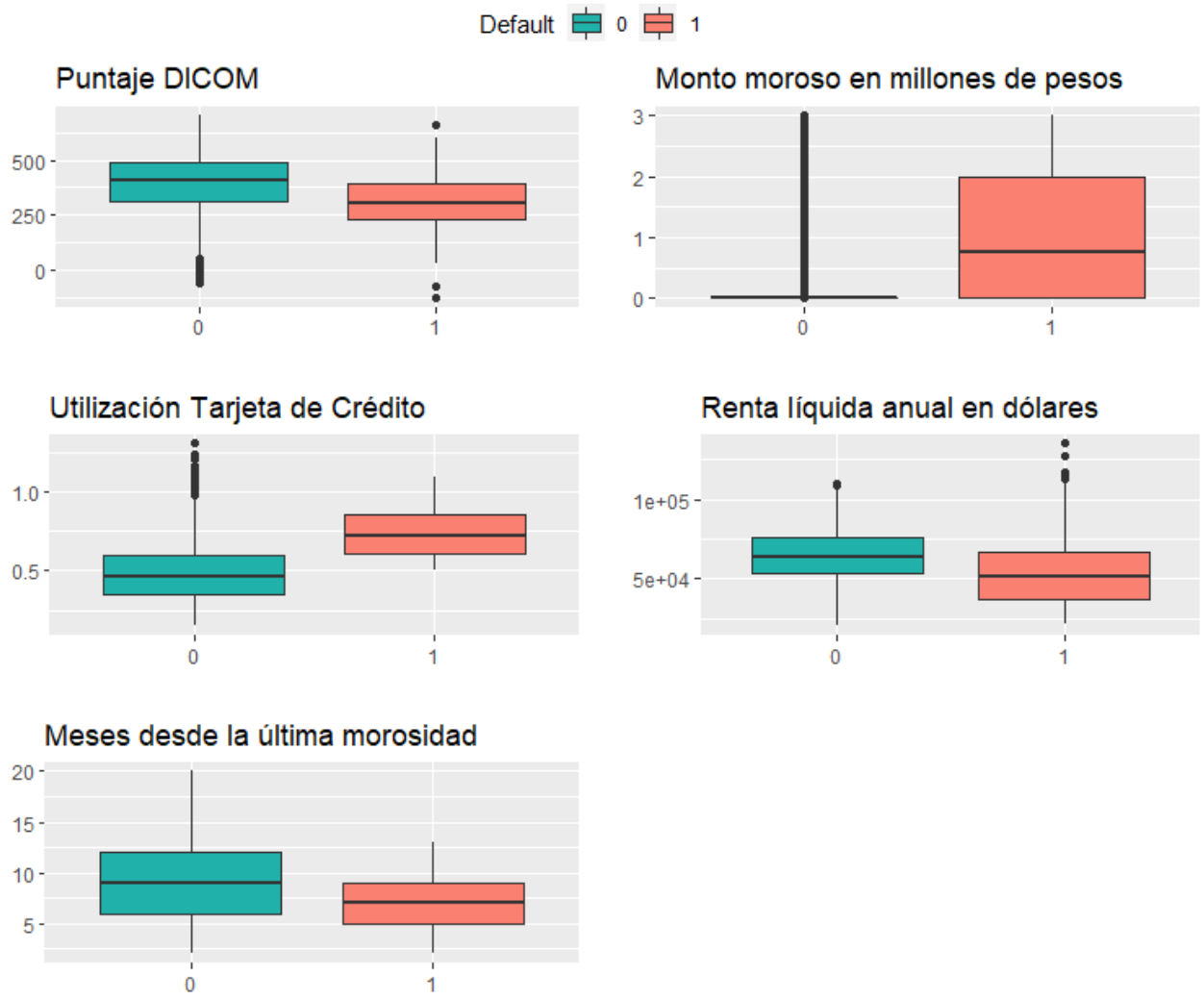


Fig. 3.4: Boxplot de las variables independientes
(Fuente: Elaboración propia)

3.1.3. Distribución variable dependiente

Dado que el objetivo del estudio es obtener una clasificación de los clientes de acuerdo a su probabilidad de incumplimiento, la variable dependiente que interesa predecir a partir de las variables independientes es la variable Default, la cual se distribuye de la siguiente manera: un 5,2 % de la cartera total de clientes pertenece a clientes marcados con incumplimiento, es decir, tienen más de 90 días de morosidad, mientras que el 94,8 % restante ha tenido un buen comportamiento de pago de sus obligaciones, esta distribución de la variable Default se puede observar en el siguiente gráfico.

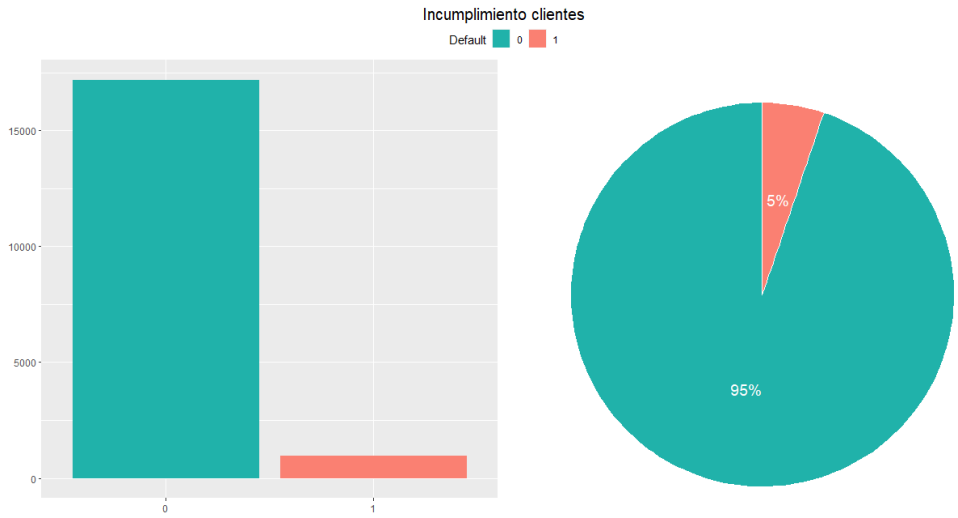


Fig. 3.5: Representación clases de la variable dependiente
(Fuente: Elaboración propia)

De aquí se desprende que la clase mayoritaria en la base corresponde al no incumplimiento de los clientes (*default* = 0).

3.1.4. Supuestos estadísticos

Tal como se menciona en el Marco Teórico, técnicas estadísticas como regresión logística y análisis discriminante lineal dependen del cumplimiento de supuestos estadísticos, específicamente para este caso se deben verificar los supuestos de normalidad univariante y multivariante, homogeneidad en la matriz de covarianzas y realizar una revisión de la multicolinealidad de las variables. A continuación, se presenta el análisis del estudio de los supuestos estadísticos mencionados.

Las herramientas más utilizadas para evaluar la normalidad corresponden a respresentaciones gráficas y test de hipotésis que contrastan la siguiente dócima:

H_0: Los datos distribuyen normalmente
H_1: Los datos no siguen una distribución normal

En primer lugar, para estudiar la normalidad univariante, se realiza el histograma junto con la normalidad esperada y el gráfico de cuantiles QQ-plot para cada una de las variables independientes por cada clase de la variable dependiente (y=0 o y=1).

De la Figura 3.6 se observa que la única variable que presenta distribución aproximadamente normal corresponde a la renta anual de los clientes con buen comportamiento crediticio (y=0), mientras que las demás variables se encuentran sesgadas donde la mayoría presenta una distribución con asimetría positiva. Esta asimetría de las distibuciones se puede relacionar con el sentido lógico del riesgo de las variables, por ejemplo, la mayoría de las personas con buen comportamiento crediticio tendrá montos morosos relativamente bajos o nulos y muy pocos tendrán montos morosos altos lo que explica la asimetría positiva de la variable Monto moroso para la clase 0, de igual manera, se espera que la mayoría de las personas que incumplen con sus obligaciones de pago tengan menores rentas anuales por lo que también se obtiene una districión de asimetría positiva de la variable Renta anual para la clase 1.

Adicionalmente al análisis del histograma, se analizan los gráficos QQ-plot donde se pueden observar los cuantiles teóricos de la distribución normal para cada variable y los cuantiles observados. Por lo general, los gráficos QQ-plot pueden tener las siguientes apariencias [50]:

- Un gráfico con forma de "S" implica que la distribución de los datos tiene colas más largas que la distribución normal.
- Un gráfico con forma de "J" implica que la distribución de los datos está sesgada. Específicamente, la forma de "J" se obtiene para distribuciones asimétricas positivas y la forma de "J" invertida para distribuciones asimétricas negativas.
- Una línea recta implica que la distribución de los datos es la misma que la distribución teórica, en este caso, la distribución normal.

De esta manera, se observa de la Figura 3.7 que para la mayoría de las variables se obtienen desviaciones de los cuantiles teóricos, obteniendo gráficos en forma de "S" y "J" lo cual indicaría posibles desviaciones de la distribución normal, complementando así el análisis gráfico anteriormente expuesto de los histogramas.

Además de las desviaciones de los cuantiles se puede observar un comportamiento atípico de los cuantiles teóricos para el Monto moroso de los clientes con buen comportamiento de pago, esto se debe a que el 87 % de las observaciones de esta variable tienen el valor 0 lo que provoca que los cuantiles sean 0 como se muestra en la tabla de estadísticos descriptivos 3.1, lo cual tiene sentido ya que es de esperar que los clientes que no se encuentren en estado de incumplimiento no presenten montos morosos.

Fig. 3.6: Histogramas variables independientes
(Fuente: Elaboración propia)

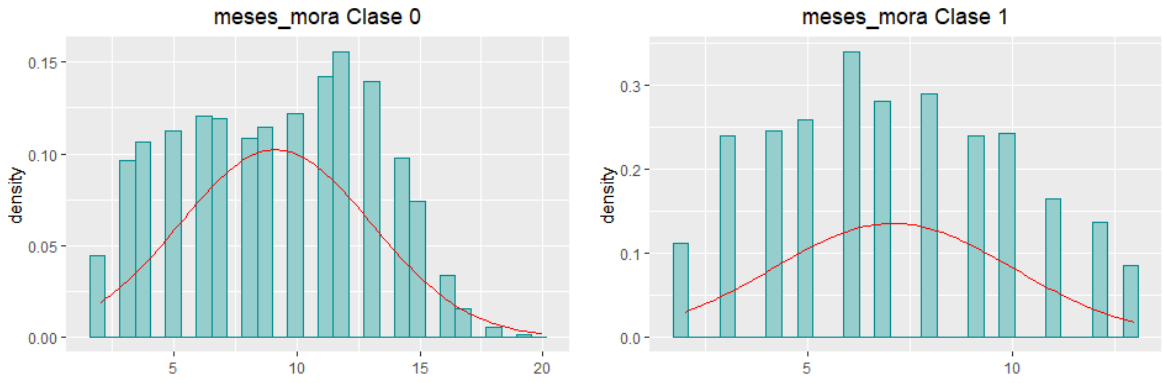


Fig. 3.6: Histogramas variables independientes
(Fuente: Elaboración propia)

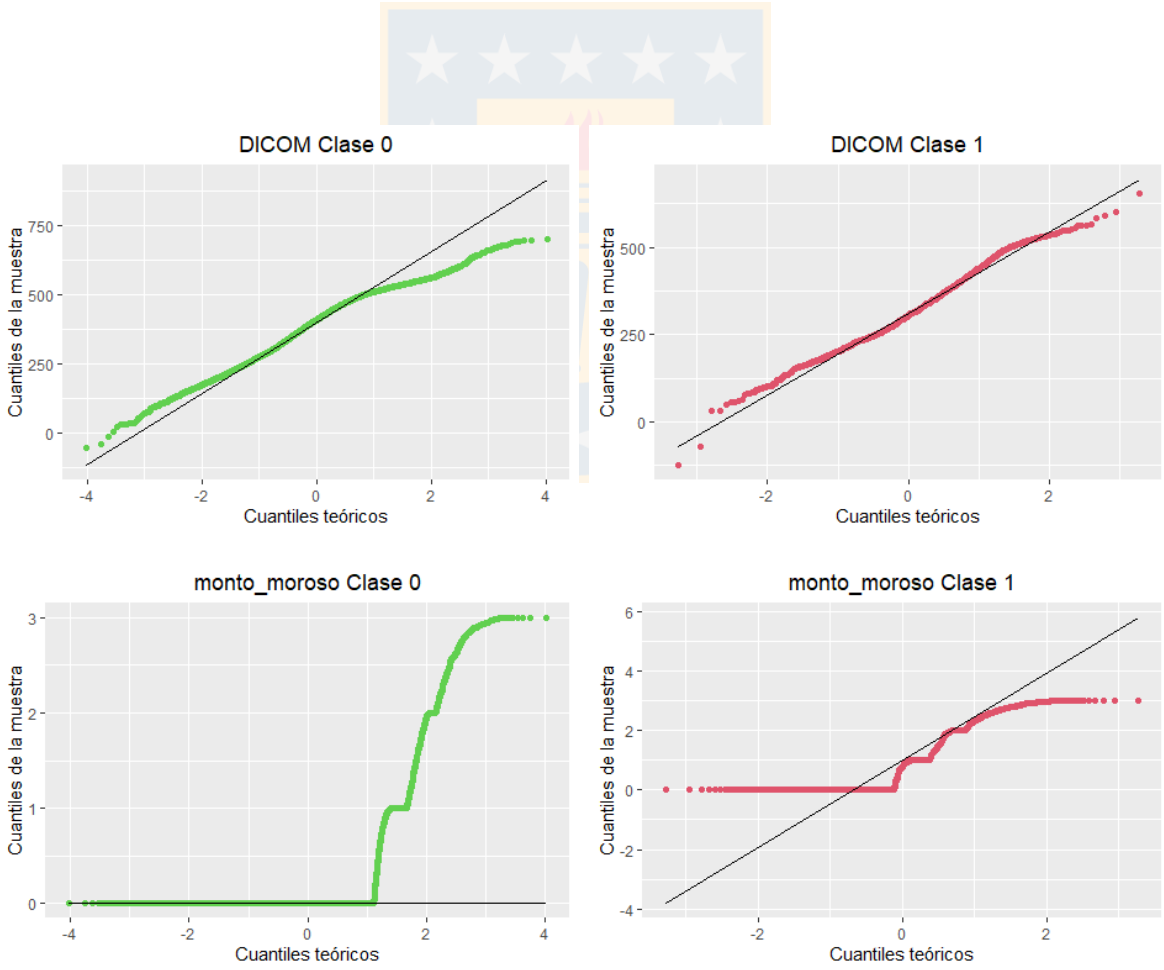


Fig. 3.7: QQ-plots variables independientes
(Fuente: Elaboración propia)

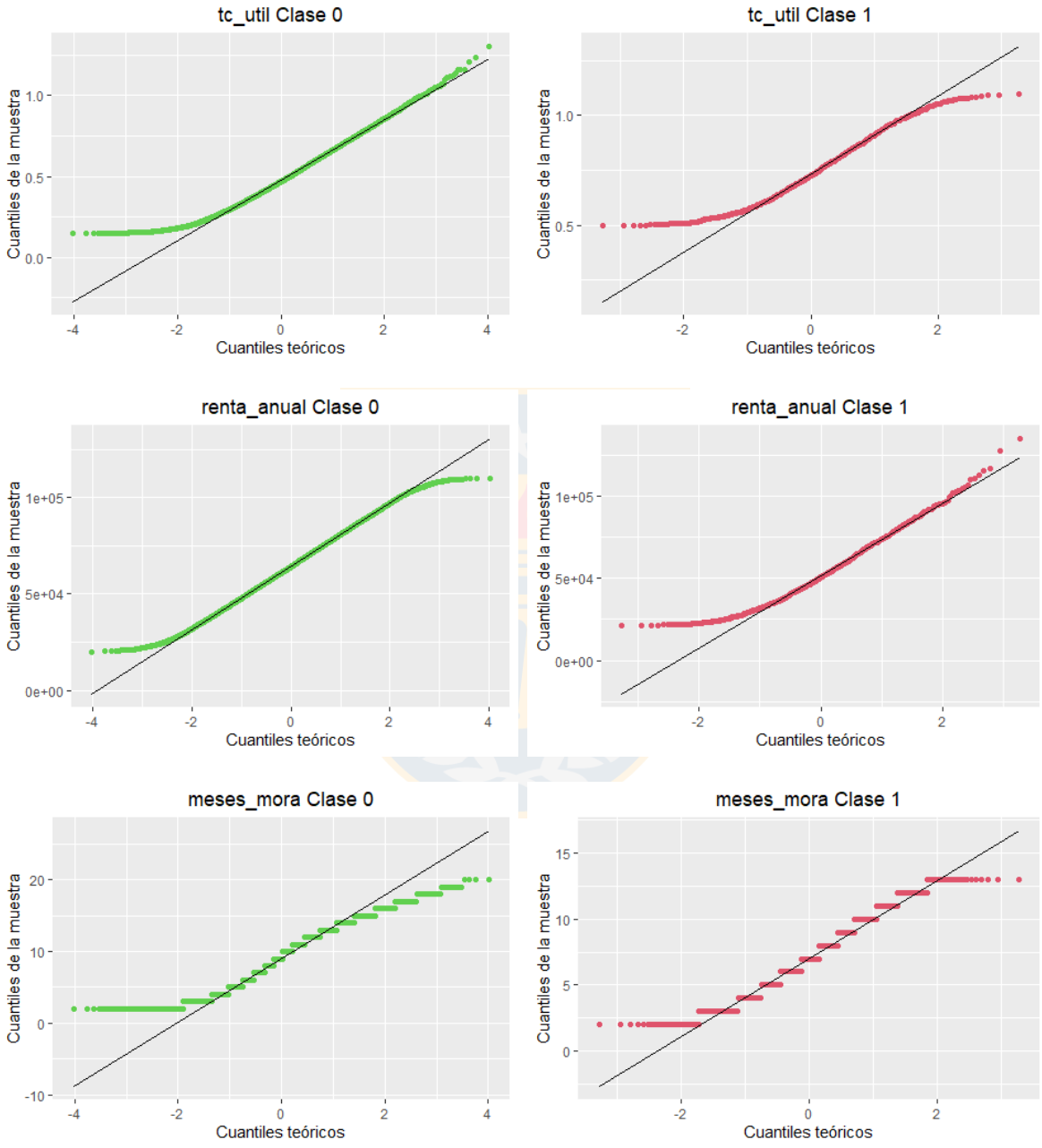


Fig. 3.7: QQ-plots variables independientes
(Fuente: elaboración propia)

Luego de realizar el análisis gráfico previo, se procede a realizar el test de hipótesis, considerando como hipótesis nula que los datos sí proceden de una distribución normal y utilizando el p-valor de la prueba estadística como regla de decisión, es decir, si el p-valor es menor que un nivel de significancia α, el resultado se considera estadísticamente significativo y por lo tanto, permite rechazar la hipótesis nula (se concluye que los datos no distibuyen normalmente), en cambio si el p-valor es mayor que α se concluye que los datos si distribuyen normal.

El test de Shapiro Wilk se considera como el más potente para el contraste de la normalidad, sin embargo, este test se emplea cuando el tamaño del conjunto de datos es pequeño (entre 3 a 5000 casos). Por lo tanto, debido a que la distribución de la variable dependiente corresponde a 17.189 observaciones clasificadas como 0 y 946 clasificadas como 1, no es posible aplicar el test Shapiro Wilk a las muestras del conjunto de datos que pertenecen a la clase 1.

No obstante, se sabe que el test de Shapiro Wilk es equivalente al test de Kolmogorov-Smirnov para muestras grandes. En particular, el test Kolmogorov-Smirnov es conservador y poco potente ya que se asume que se conoce la media y la varianza poblacional lo que en la mayoría de los casos no sucede, por lo que para solucionar este problema se desarrolló una modificación conocida como test Lilliefors, el cual asume que la media y varianza son desconocidas y está especialmente desarrollado para testear la normalidad. [51] [52] Por lo tanto, para verificar la normalidad univariante por clases mediante el contraste de hipótesis, se utilizó la función `lillie.test()` obteniendo los resultados que se presentan en la siguiente tabla:

Variables	Incumplimiento	p-valor
DICOM	0	$1,22 \times 10^{-201}$
	1	$1,35 \times 10^{-5}$
Monto moroso	0	0,00
	1	$1,20 \times 10^{-197}$
Utilización TC	0	$1,39 \times 10^{-42}$
	1	$5,32 \times 10^{-8}$
Renta anual	0	0,12
	1	$1,09 \times 10^{-7}$
Meses mora	0	0,00
	1	$2,92 \times 10^{-24}$

Tabla 3.5: P-valor contraste de hipótesis de normalidad
(Fuente: Elaboración propia)

Por lo tanto, utilizando un nivel de significancia de $\alpha = 0{,}05$ se concluye que existen evidencias significativas de falta de normalidad univariante en todas las variables empleadas como predictores, a excepción de la variable Renta anual cuando la variable Default es 0.

Como consecuencia de la falta de normalidad univariante, se puede deducir que la observaciones no siguen una distribución normal multivariante. Sin embargo, a fin de corroborar esta hipótesis se utilizó la función `mvn()` que implementa las 3 pruebas más utilizadas, las cuales corresponden a los test de Mardia, Henze-Zirkler y Royston. Específicamente, las pruebas de Henze-Zirkler y Royston son sugeridas para evaluar normalidad multivariante debido a su buen control de errores tipo I y potencia, por su parte, el test de Mardia es útil para identificar el motivo de la desviación de la normalidad (asimetría y/o curtosis). [53]

Cabe destacar que el test de Royston depende del estadístico Shapiro-Wilk, por lo tanto, como se menciona anteriormente, no es posible utilizarlo debido al tamaño de los datos, por tal razón, el test empleado en la función `mvn` es el de Henze-Zirkler obteniendo un p-valor = 0, por lo que bajo un nivel de significacia de 0,05 existe evidencia significativa de que los datos no siguen una distibución normal multivariante.

Por otro lado, el segundo supuesto que se debe verificar para las técnicas estadísticas empleadas, es el supuesto de homogeneidad de las matrices de covarianzas. Por lo general, el método empleado para comparar matrices de covarianzas más citado por la literatura es la prueba de Bartlett, específicamente el test Box M que fue desarrollado como una extensión de esta para escenarios multivariantes, el cual permite contrastar la igualdad de las matrices entre los grupos. Sin embargo, este test es sensible a las violaciones del supuesto de normalidad multivariante, por lo que se recomienda utilizar un nivel de significancia de $\alpha = 0{,}001$. [54] [55]

De esta manera, usando la función `boxM()` para contrastar la homogeneidad de covarianzas se obtiene un p-valor = $2{,}2 \times 10^{-16} < 0{,}001$ y dado que la hipótesis nula es que las matrices de covarianzas son iguales, se concluye que existe evidencia significativa para rechazar H_0, por lo tanto las matrices de covarianzas de los datos por grupo no son iguales.

Esta desigualdad de matrices de covarianzas y la falta de normalidad de los datos se debe tener en cuenta a la hora de sacar conclusiones con respecto a la metodología de Análisis discriminante lineal.

Por último, dado que el modelo de regresión logística debe presentar poca o nula multicolinealidad se estudia la correlación entre las variables independientes mediante la siguiente matriz obtenida en R:

Fig. 3.8: Matriz de correlación variables independientes
(Fuente Elaboración propia)

De aquí se sigue que la utilización de la tarjeta de crédito y los meses que han pasado desde la última mora del cliente están inversamente correlacionadas, es decir, a mayor cantidad de meses sin morosidad menor es el ratio utilizado de la tarjeta de crédito y viceversa. Sin embargo, en vista de que solamente se tienen 5 variables independientes, se decidió utilizar toda la información disponible para construir los modelos de clasificación, a pesar de la correlación de las variables Utilización TC y Meses mora.

3.1.5. División de la base de datos

Finalmente, luego de haber realizado el análisis de datos y seleccionar las variables a utilizar, se procede a dividir la base en los llamados conjuntos de entrenamiento y validación, con el objetivo de entrenar y posteriormente evaluar la calidad de clasificación de los modelos a implementar.

Para ello se generó la siguiente división del 80 % y 20 % de los datos, utilizando la función `createDataPartition()`.

Conjunto de datos	Total	Buenos	Malos	Tasa de buenos	Tasa de malos
Entrenamiento (80 %)	14508	13762	746	94,86 %	5,14 %
Validación (20 %)	3627	3427	200	94,49 %	5,51 %

Tabla 3.6: Partición del Conjunto de datos
(Fuente: Elaboración propia)

De la Tabla 3.6 se observa que la distribución de la variable dependiente (clientes buenos/cumplimiento y clientes malos/incumplimiento) es similar en el conjunto de entrenamiento y en el conjunto de validación. Además, se mantiene la proporción de la base completa la cual correspondía a un 95 % de clientes buenos y un 5 % de clientes malos.

Por lo tanto, ahora que los datos han sido preprocesados se emplean los algoritmos de Machine Learning que permiten crear un modelo capaz de representar los patrones presentes en los datos de entrenamiento y generalizarlos a nuevas observaciones.

4. Resultados

En este capítulo, se muestran los resultados de la aplicación de cada uno de los modelos considerando la información de la base de datos estudiada en el capítulo anterior, además se presentan las matrices de confusión y la comparación de los modelos de clasificación a partir las métricas que miden el rendimiento de los algoritmos que nos permiten calcular la probabilidad de incumplimiento de los clientes de la cartera.

Cabe mencionar que para el problema abordado los valores positivos de la matriz de confusión corresponden a los clientes con buen comportamiento de pago, mientras que los valores negativos corresponden a los clientes marcados en incumplimiento, por lo tanto se tiene que:

- Los verdaderos positivos corresponderán a los clientes buenos que han sido clasificados como buenos.
- Los verdaderos negativos corresponderán a los clientes malos que han sido clasificados como malos.
- Los falsos positivos corresponderán a los clientes malos que han sido clasificados como buenos.
- Los falsos negativos corresponderán a los clientes buenos que han sido clasificados como malos.

De estos valores es que derivan las principales métricas de evaluación de los modelos que se mencionan en el Marco Teórico, las cuales fueron calculadas para cada uno de los modelos entrenados, sin embargo, en este informe solo se presentan la exactitud, el error tipo I y el error tipo II, además del AUC y del índice KS, ya que fueron seleccionadas para la comparación de los algoritmos de acuerdo al contexto de predecir el incumplimiento en el pago de un crédito.

De esta manera estaremos comparando la proporción de clientes que son clasificados correctamente sin importar su clasificación, las proporciones de clientes clasificados erróneamente dependiendo de su clasificación, y la capacidad de los modelos para discriminar entre clientes buenos y malos.

Inicialmente los algoritmos fueron entrenados bajo 5 escenarios distintos, eligiendo finalmente el escenario en el cual se entrenan los datos y se aplica el punto de corte identificado por el índice de Youden para obtener la matriz de confusión y las métricas de evaluación. A continuación, se presentan los resultados de los modelos construidos bajo este escenario, aplicando los algoritmos sobre la base de entrenamiento con los parámetros predeterminados de cada uno en R, y también, se presentan los resultados de la aplicación de los modelos entrenados tanto a la base de entrenamiento como a la base de validación.

El detalle de las funciones, paquetes y parámetros utilizados para el entrenamiento de los algoritmos se puede encontrar en el Anexo A.1.

4.1. Regresión logística

Los estimadores de los coeficientes β obtenidos al entrenar el modelo de Regresión logística y sus valores p asociados se presentan en la Tabla 4.1, donde se observa que todos los coeficientes de las variables son significativos bajo un nivel de significancia de $\alpha = 0{,}05$, por lo tanto, se concluye que todos los coeficientes son distintos de 0.

Variable	Coeficiente	p valor
Intercepto	$-4,064$	$< 2 \times 10^{-16}$
DICOM	$-3,516 \times 10^{-3}$	$2,66 \times 10^{-16}$
Monto Moroso	$1,047$	$< 2 \times 10^{-16}$
Utilización TC	$7,614$	$< 2 \times 10^{-16}$
Renta anual	$-4,206 \times 10^{-5}$	$< 2 \times 10^{-16}$
Meses mora	$-3,042 \times 10^{-2}$	$0,0376$

Tabla 4.1: Coeficientes del modelo de Regresión logística

Una vez que se ha entrenado el modelo y se ha verificado la significancia estadística de los coeficientes de la Regresión logística, se procede a calcular las probabilidades predichas para posteriormente identificar el punto de corte óptimo de la base de entrenamiento y la base de validación a través del índice de Youden, obteniendo un punto de corte de 5,3 % y 4,8 % respectivamente.

Luego, se realizan las clasificaciones correspondientes para cada una de las bases, clasificando a los clientes con probabilidad mayor al punto de corte como clientes en incumplimiento. De

esta manera se obtienen las siguientes matrices de confusión:

		Predicción	
		0	1
Real	0	11562	2200
	1	137	609

Tabla 4.2: Matriz de confusión conjunto de entrenamiento Regresión logística

		Predicción	
		0	1
Real	0	2823	604
	1	29	171

Tabla 4.3: Matriz de confusión conjunto de validación Regresión logística

Específicamente, de las matrices de confusión obtenidas para el modelo de Regresión logística se puede concluir que del total de observaciones de la cartera, 15165 observaciones fueron clasificadas correctamente mientras que 2970 fueron clasificadas erróneamente.

4.2. Análisis discriminante lineal

Al entrenar el modelo de Análisis discriminante lineal se obtienen los siguientes parámetros estimados que conforman la función discriminante (2.10):

1. Las probabilidades a priori estimadas de pertenecer a uno de los dos grupos corresponden a $\pi_0 = 0,949$ y $\pi_1 = 0,514$.

2. En la Tabla 4.4 se presentan las estimaciones de μ_k, las cuales corresponden al promedio de cada predictor dentro de cada clase.

Incumplimiento	DICOM	Monto Moroso	Utilización TC	Renta anual	Meses mora
0	395,62	0,16	0,48	64366,43	9,12
1	312,89	0,93	0,74	52542,46	7,13

Tabla 4.4: Estimaciones μ_k del modelo Análisis discriminante lineal

3. Por último, de acuerdo a los factores obtenidos se tiene que la combinación lineal de las observaciones $X = x$ es la siguiente:

$$-1,29 \times 10^{-3} X_1 + 1,21 X_2 + 4,21 X_3 - 1,95 \times 10^{-5} X_4 + 4,38 \times 10^{-2} X_5 \tag{4.1}$$

Utilizando estos estimadores se realizan las predicciones de acuerdo a la regla de decisión del clasificador de Bayes y se aplican los puntos de corte 2,0 % y 2,3 % (identificados a partir del índice de Youden) a las probabilidades predichas de la base de entrenamiento y validación respectivamente, obtienendo así las matrices de confusión 4.5 y 4.6, de las cuales se puede concluir que del total de observaciones de la cartera, 14615 observaciones fueron clasificadas correctamente mientras que 3520 fueron clasificadas erróneamente.

		Predicción	
		0	1
Real	0	11045	2717
	1	130	616

Tabla 4.5: Matriz de confusión conjunto de entrenamiento Análisis discriminante lineal

		Predicción	
		0	1
Real	0	2785	642
	1	31	169

Tabla 4.6: Matriz de confusión conjunto de validación Análisis discriminante lineal

4.3. Árboles de decisión

A continuación, en la Figura 4.1 se observa la representación gráfica del árbol de clasificación creado a partir de la base de entrenamiento, donde las divisiones fueron generadas a partir de la ganancia de información del índice de Gini. De aquí se observa que los predictores elegidos para las reglas de decisión corresponden a Utilización TC, Monto moroso y Renta anual, de acuerdo a lo visto anteriormente en el análisis de datos, estás 3 variables son las que mayor valor de información presentan por lo tanto discriminan mejor entre clientes malos y buenos, por lo que tiene sentido que sean las variables que presenten mayor ganancia de información.

La interpretación del árbol de decisión obtenido en la Figura 4.1 es la siguiente: cada rectángulo corresponde a un nodo del árbol en el cual se muestra la categoría predicha de acuerdo a

las reglas de decisión, la proporción de casos de cada categoría y el porcentaje de observaciones en el nodo. Por ejemplo, el 89 % de las observaciones totales presenta un ratio utilizado de su tarjeta de crédito menor al 77 % y un monto moroso menor a 1.8 millones de pesos y fueron clasificadas como clientes sin incumplimiento en el nodo terminal N°4 (rectángulo inferior izquierdo del gráfico), de estas observaciones el 98 % corresponde realmente a clientes buenos mientras que el 2 % corresponde a clientes malos. Por lo tanto, se observa como en cada nodo se puede obtener una idea de la precisión del modelo al momento de hacer predicciones.

Así, utilizando las reglas de decisión y aplicando el punto de corte 6,7 % (identificado a partir del índice de Youden) a la base de entrenamiento y validación, se obtienen las matrices de confusión 4.7 y 4.8, de las cuales se puede concluir que del total de observaciones de la cartera, 16507 observaciones fueron clasificadas correctamente mientras que 1628 fueron clasificadas erróneamente.

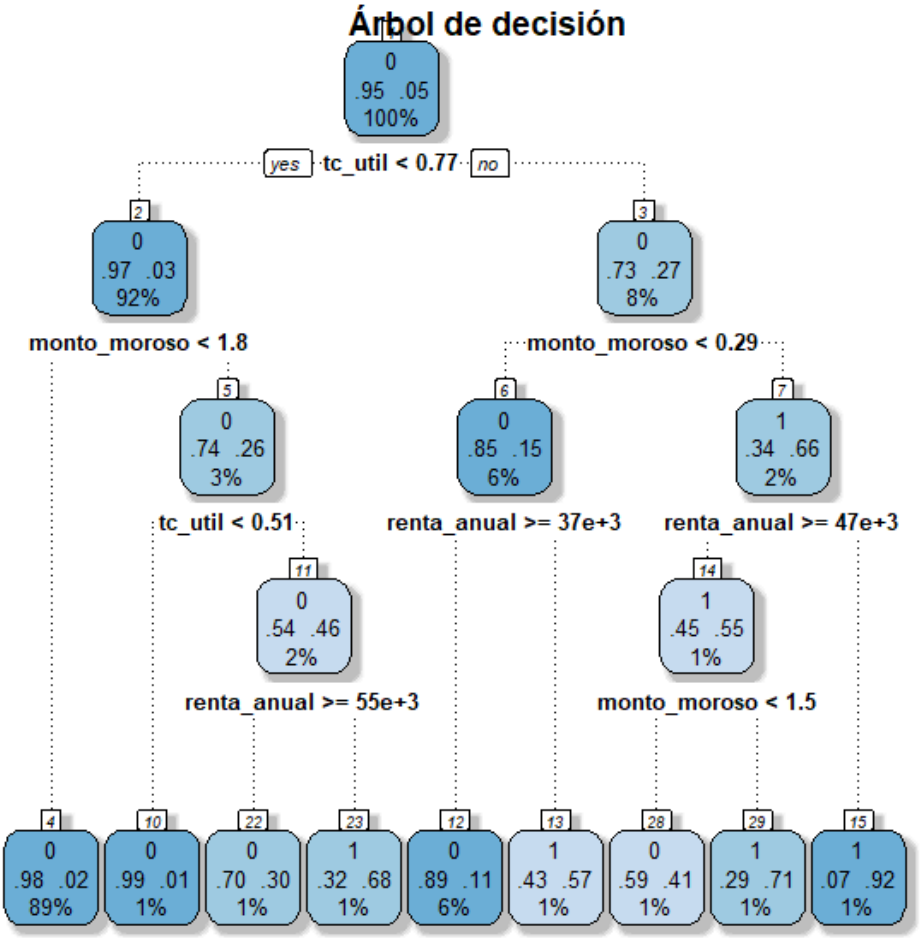


Fig. 4.1: Partición del árbol y reglas de decisión

		Predicción	
		0	1
Real	0	12785	977
	1	324	422

Tabla 4.7: Matriz de confusión conjunto de entrenamiento Árbol de decisión

		Predicción	
		0	1
Real	0	3176	251
	1	76	124

Tabla 4.8: Matriz de confusión conjunto de validación Árbol de decisión

4.4. Random Forest

El algoritmo Random Forest es entrenado aplicando 200 árboles de decisión, ya que como se menciona en el Marco Teórico los resultados se estabilizan alrededor de este número (Hastie et al., 2018).

Dado que una de las desventajas de este algoritmo es la pérdida de interpretabilidad al combinar los resultados de múltiples árboles, no es posible interpretar las reglas de decisión del bosque aleatorio, sin embargo, se puede visualizar la importancia de las variables a través de la disminución media del índice de Gini en la Figura 4.2.

En particular, cuanto mayor sea el valor de la disminución media de Gini mayor será la importancia de la variable en el modelo, por lo tanto, a partir de la Figura se puede concluir que la variable que más aporta al aprendizaje del modelo corresponde al ratio utilizado de la tarjeta de crédito, mientras que la variable que menos aporta corresponde a la cantidad de meses desde la última mora.

Una vez que se ha entrenado el modelo, se procede a realizar las predicciones de probabilidad y aplicar los puntos de corte identificados a partir del índice de Youden, los cuales corresponden a un 44,5 % y un 5,8 %, para la base de entrenamiento y validación respectivamente, obteniendo así las matrices de confusión 4.9 y 4.10, de las cuales se puede concluir que del total de observaciones de la cartera, 17540 observaciones fueron clasificadas correctamente mientras que 595 fueron clasificadas erróneamente.

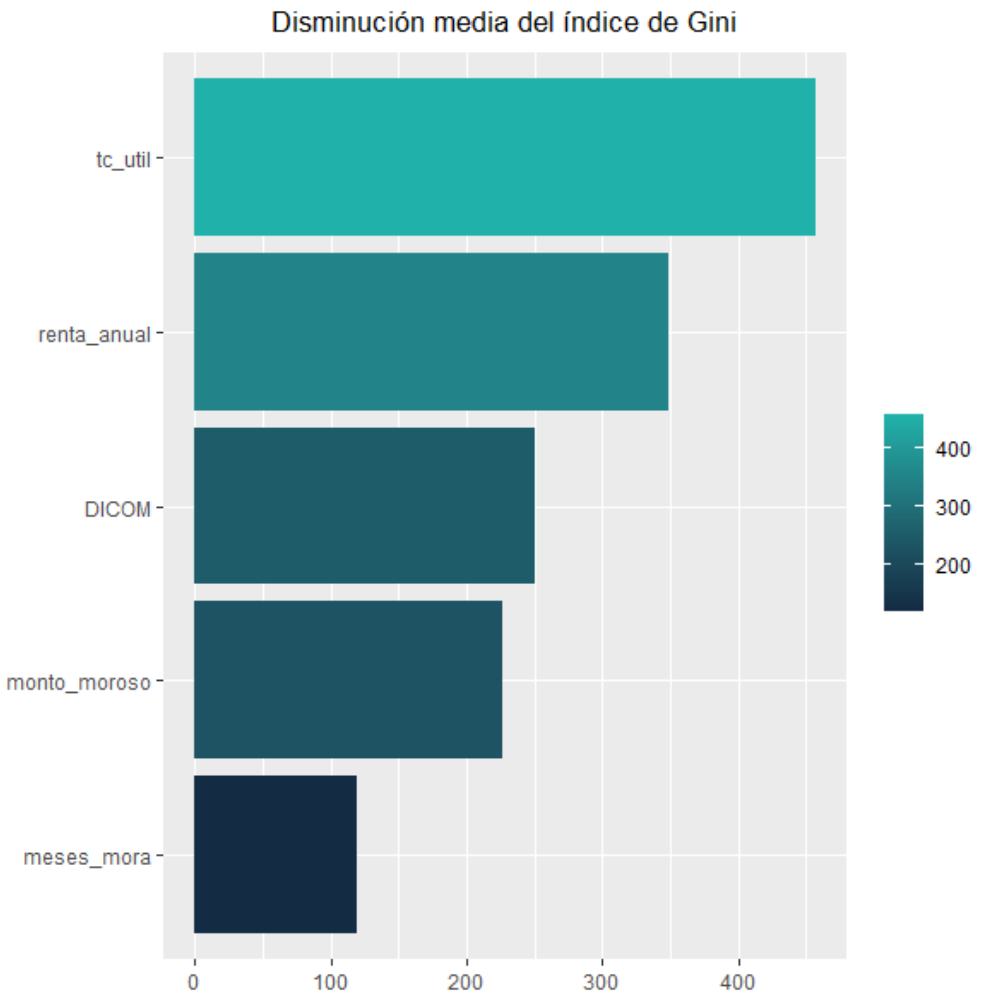


Fig. 4.2: Disminución media de Gini del modelo Random Forest

		Predicción	
		0	1
Real	0	13762	0
	1	0	746

Tabla 4.9: Matriz de confusión conjunto de entrenamiento Random Forest

		Predicción	
		0	1
Real	0	2864	563
	1	32	168

Tabla 4.10: Matriz de confusión conjunto de validación Random Forest

4.5. Gradient Boosting

Para el caso de Gradient Boosting el algoritmo es entrenado aplicando los valores predeterminados de la función de R los cuales corresponden a 100 árboles y una tasa de aprendizaje igual a $0,1$. De igual manera que en Random Forest, Gradient Boosting pierde interpretabilidad, sin embargo, en la Figura 4.3 se visualiza la importancia de las variables del modelo obtenido, donde se observa que 4 de las 5 variables explicativas tiene influencia en el modelo, específicamente, la variable que más aporta en el aprendizaje del modelo es el ratio utilizado de la tarjeta de crédito, mientras que el aporte de la cantidad de meses desde la última mora es nulo.

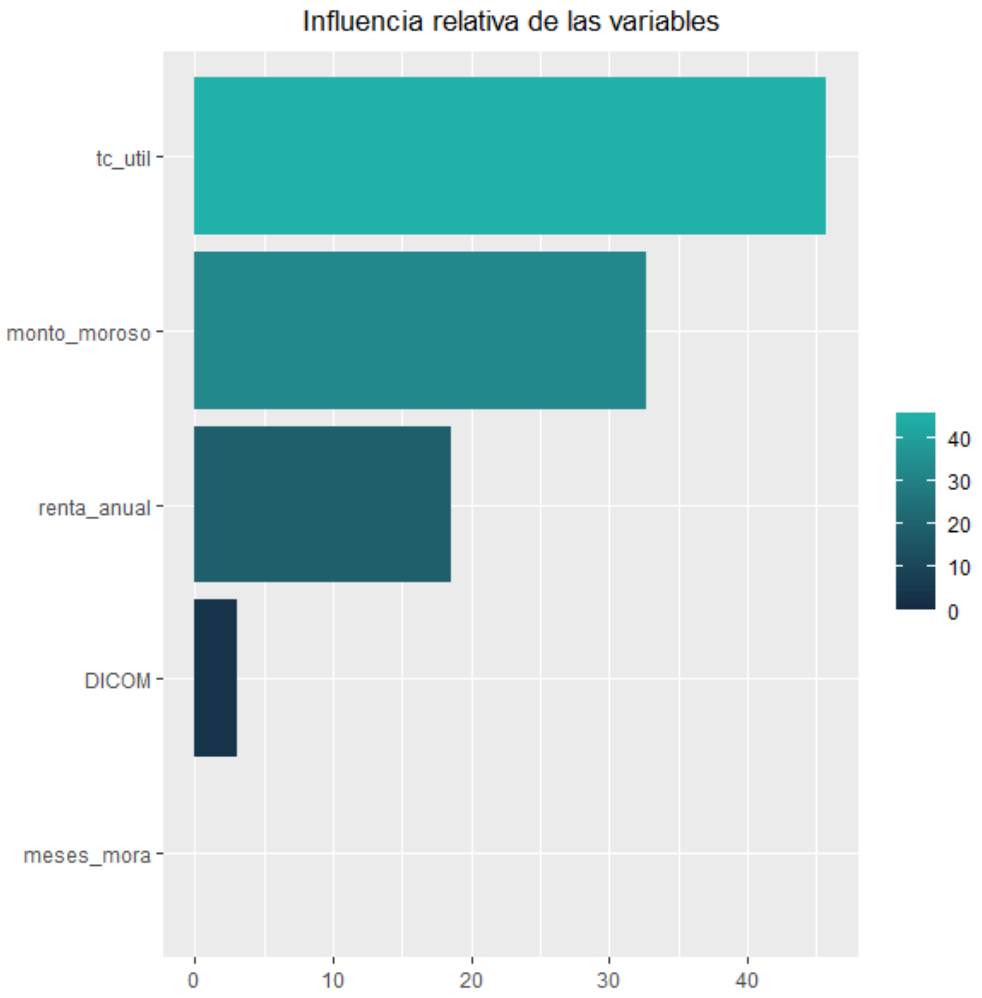


Fig. 4.3: Influencia relativa de las variables del modelo Gradient Boosting

Luego de entrenar el modelo, se aplican los puntos de corte 6,1 % y 4,9 % (identificados a partir del índice de Youden) a las probabilidades predichas de la base de entrenamiento y validación respectivamente, obteniendo las matrices de confusión 4.11 y 4.12, de las cuales se puede

concluir que del total de observaciones de la cartera, 15575 observaciones fueron clasificadas correctamente mientras que 2561 fueron clasificadas erróneamente.

		Predicción	
		0	1
Real	0	11949	1813
	1	135	611

Tabla 4.11: Matriz de confusión conjunto de entrenamiento Gradient Boosting

		Predicción	
		0	1
Real	0	2837	591
	1	22	178

Tabla 4.12: Matriz de confusión conjunto de validación Gradient Boosting

4.6. Extreme Gradient Boosting

De igual manera que Gradient Boosting, el algoritmo Extreme Gradient Boosting es entrenado con un número de iteraciones igual a 100, pero con una tasa de aprendizaje igual a 0,3 ya que es el valor predeterminado de la función. En la Figura 4.4 se encuentra la representación de la importancia de las variables a partir de la ganancia de información del modelo, donde se puede observar nuevamente que la variable con mayor influencia en el modelo es el ratio de la utilización de la tarjeta de crédito, mientras que la variable que menor influencia tiene en el modelo es la cantidad de meses desde la última mora.

Con este modelo, se procede a realizar las predicciones de probabilidad y aplicar los puntos de corte identificados a partir del índice de Youden, los cuales corresponden a un 9,6 % y un 4,5 %, para la base de entrenamiento y validación respectivamente, obteniendo las matrices de confusión 4.13 y 4.14.

De las matrices de confusión obtenidas se puede concluir que del total de observaciones de la cartera, 16564 observaciones fueron clasificadas correctamente mientras que 1571 fueron clasificadas erróneamente.

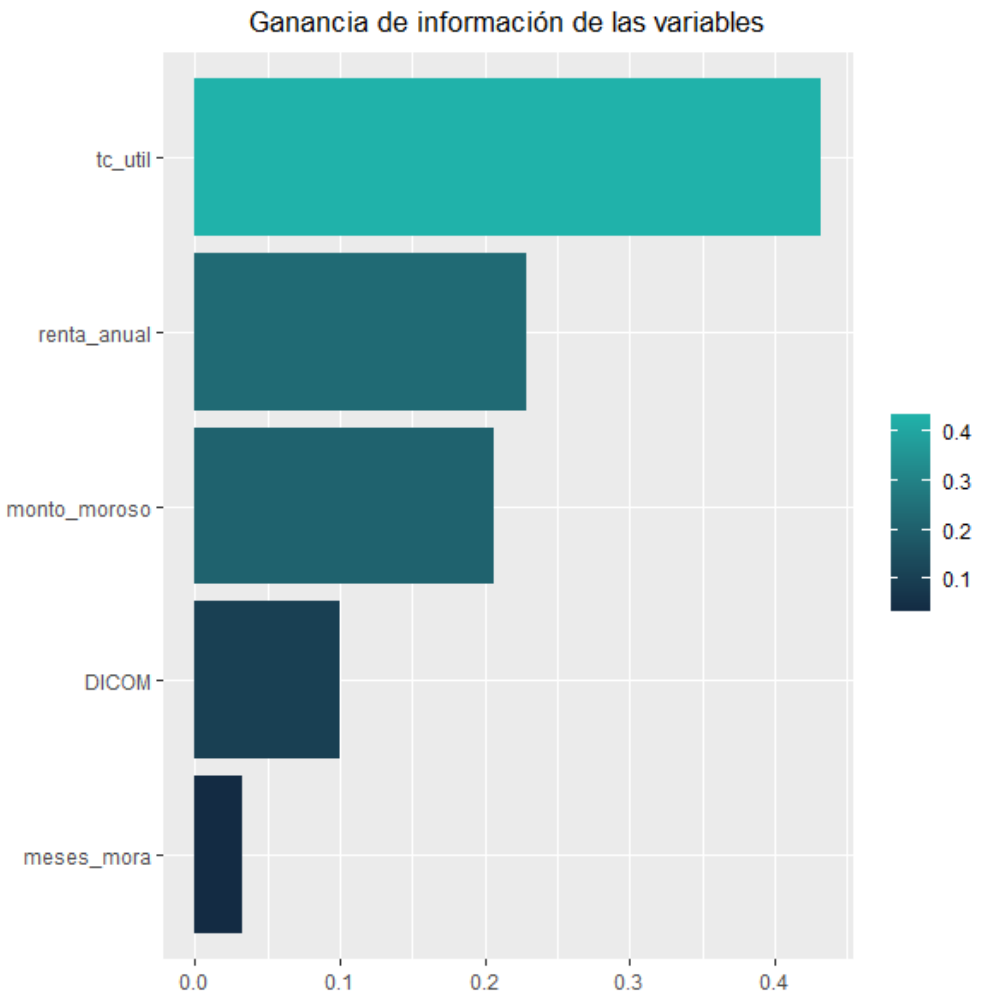


Fig. 4.4: Ganancia de información de las variables del modelo Extreme Gradient Boosting

		Predicción	
		0	1
Real	0	12837	925
	1	51	695

Tabla 4.13: Matriz de confusión conjunto de entrenamiento Extreme Gradient Boosting

		Predicción	
		0	1
Real	0	2861	566
	1	29	171

Tabla 4.14: Matriz de confusión conjunto de validación Extreme Gradient Boosting

4.7. Support Vector Machine

En el caso del algoritmo Support vector machine se entrenaron 3 modelos con las 3 funciones kernel más utilizadas: lineal, polinomial y radial. A continuación se presentan los resultados de los modelos entrenados.

4.7.1. Lineal

A modo de ejemplo, en la Figura 4.5 se muestra la representación gráfica del hiperplano de separación y las dos regiones en las que queda dividido el espacio muestral entre las variables Monto moroso y Renta anual para el algoritmo Support vector machine entrenado con kernel lineal.

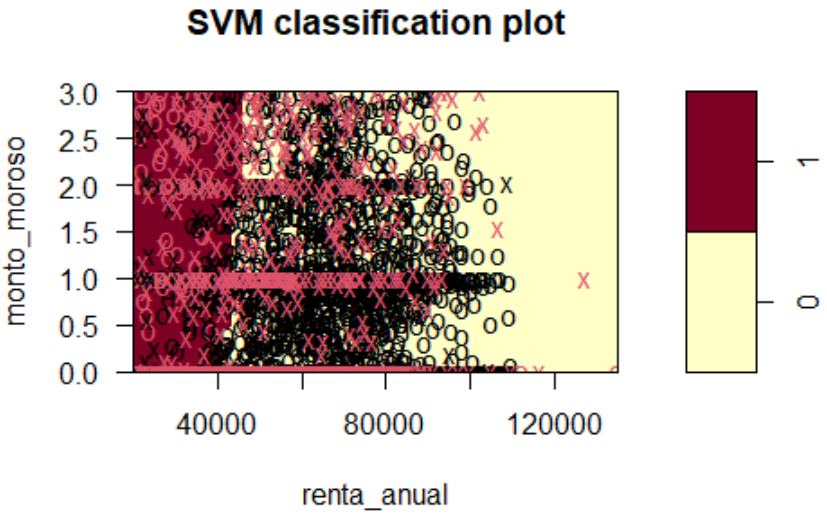


Fig. 4.5: Hiperplano de separación lineal entre el Monto moroso y la Renta anual

En particular, el modelo obtenido cuenta con 1295 vectores de soporte, de los cuales 649 pertenecen a las observaciones clasificadas como no incumplimiento y 646 a las observaciones clasificadas como incumplimiento. Con estos vectores de soporte se predicen las clases a las que pertenecen las observaciones y se obtienen las probabilidades utilizando el booleano *probability* de la función de R, para posteriormente aplicar los puntos de corte 5,0 % y 5,6 % (identificados a partir del índice de Youden) a la base de entrenamiento y validación, obteniendo las matrices de confusión 4.15 y 4.16, de las cuales se puede concluir que del total de observaciones de la cartera,

13163 observaciones fueron clasificadas correctamente mientras que 4972 fueron clasificadas erróneamente.

		Predicción	
		0	1
Real	0	9971	3791
	1	246	500

Tabla 4.15: Matriz de confusión conjunto de entrenamiento SVM lineal

		Predicción	
		0	1
Real	0	2563	864
	1	71	129

Tabla 4.16: Matriz de confusión conjunto de validación SVM lineal

4.7.2. Polinomial

Para el algoritmo Support Vector Machines entrenado con kernel polinomial, se muestra a modo de ejemplo la representación gráfica del hiperplano de separación y las dos regiones en las que queda dividido el espacio muestral entre las variables Monto moroso y Renta anual en la siguiente Figura.

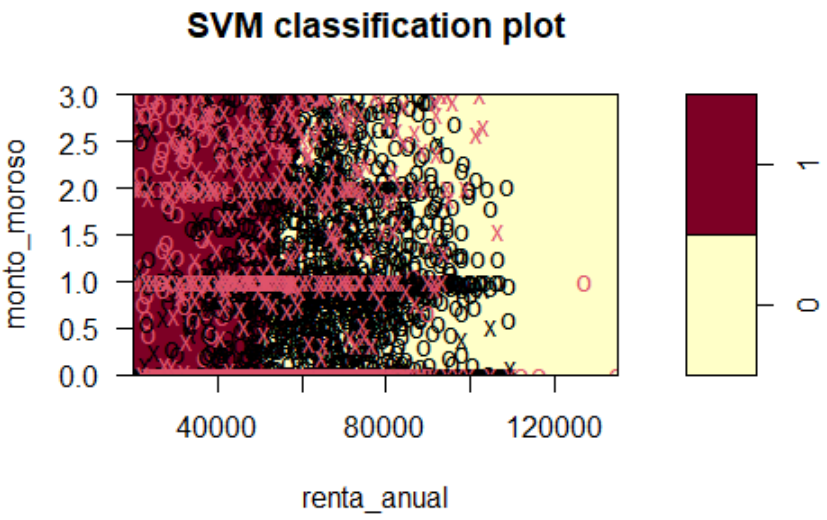


Fig. 4.6: Hiperplano de separación polinomial entre el Monto moroso y la Renta anual

En particular, el modelo obtenido cuenta con 1195 vectores de soporte, de los cuales 598 pertenecen a las observaciones clasificadas como no incumplimiento y 597 a las observaciones clasificadas como incumplimiento. Con estos vectores de soporte se predicen las clases a las que pertenecen las observaciones y se obtienen las probabilidades utilizando el booleano *probability* de la función de R, para posteriormente aplicar los puntos de corte 6,0 % y 6,3 % (identificados a partir del índice de Youden) a la base de entrenamiento y validación, obteniendo las siguientes matrices de confusión, de las cuales se puede concluir que del total de observaciones de la cartera, 12839 observaciones fueron clasificadas correctamente mientras que 5296 fueron clasificadas erróneamente.

		Predicción	
		0	1
Real	0	9686	4076
	1	239	507

Tabla 4.17: Matriz de confusión conjunto de entrenamiento SVM polinomial

		Predicción	
		0	1
Real	0	2516	911
	1	70	130

Tabla 4.18: Matriz de confusión conjunto de validación SVM polinomial

4.7.3. Radial

Por último, se presenta a modo de ejemplo la representación gráfica del hiperplano de separación entre las variables Monto moroso y Renta anual del algoritmo Support Vector Machines entrenado con kernel radial en la Figura 4.7.

En particular, el modelo obtenido cuenta con 14508 vectores de soporte, de los cuales 13762 pertenecen a las observaciones clasificadas como no incumplimiento y 746 a las observaciones clasificadas como incumplimiento. Con estos vectores de soporte se predicen las clases a las que pertenecen las observaciones y se obtienen las probabilidades utilizando el booleano *probability* de la función de R, para posteriormente aplicar los puntos de corte 5,0 % y 5,2 % (identificados a partir del índice de Youden) a la base de entrenamiento y validación, obteniendo las matrices de confusión 4.19 y 4.20, de las cuales se puede concluir que del total de observaciones de la cartera,

15464 observaciones fueron clasificadas correctamente mientras que 2671 fueron clasificadas erróneamente.

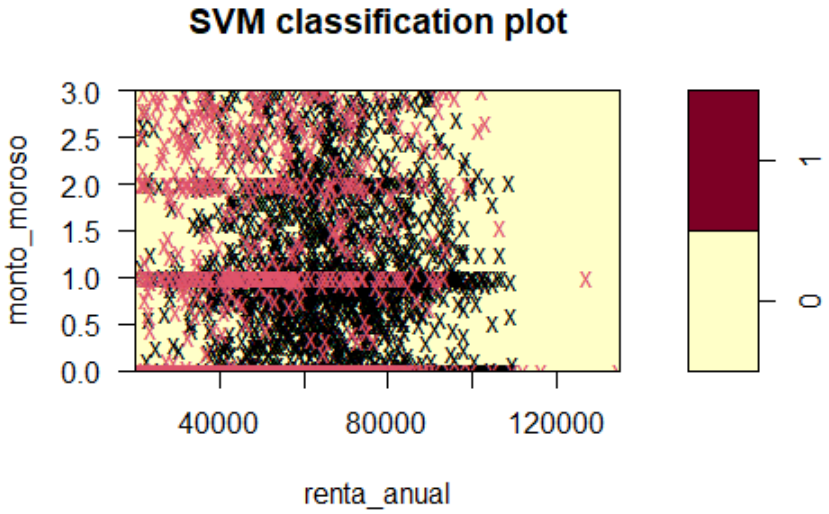


Fig. 4.7: Hiperplano de separación radial entre el Monto moroso y la Renta anual

		Predicción	
		0	1
Real	0	13762	0
	1	0	746

Tabla 4.19: Matriz de confusión conjunto de entrenamiento SVM radial

		Predicción	
		0	1
Real	0	787	2640
	1	31	169

Tabla 4.20: Matriz de confusión conjunto de validación SVM radial

4.8. Comparación de los modelos

A continuación se presentan las métricas obtenidas a partir de las matrices de confusión anteriormente expuestas, con la finalidad de comparar los modelos que permiten predecir la probabilidad de incumplimiento en base a su bondad de ajuste.

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Regresión logística	83,9 %	18,4 %	16,0 %	91,6 %	65,6 %
Análisis discriminante lineal	80,4 %	17,4 %	19,7 %	90,3 %	62,8 %
Árboles de decisión	91,0 %	43,4 %	7,1 %	76,2 %	49,5 %
Random Forest	**100,0 %**	**0,0 %**	**0,0 %**	**100,0 %**	**100,0 %**
Gradient Boosting	86,6 %	18,1 %	13,2 %	93,2 %	68,7 %
Extreme Gradient Boosting	93,3 %	6,8 %	6,7 %	98,4 %	86,4 %
SVM lineal	72,2 %	33,0 %	27,5 %	75,1 %	39,5 %
SVM polinomial	70,3 %	32,0 %	29,6 %	74,6 %	38,3 %
SVM radial	**100,0 %**	**0,0 %**	**0,0 %**	**100,0 %**	**100,0 %**

Tabla 4.21: Métricas de evaluación conjunto de entrenamiento

Cabe destacar que debido al desbalance de clases en los problemas de calificación crediticia, el interés se centra en poder identificar correctamente a los malos clientes, por lo tanto, una de las métricas claves para concluir es el error tipo I ya que describe la probabilidad de clasificar un mal cliente como un buen cliente.

En general, de la Tabla 4.21 se observa que para el conjunto de entrenamiento los modelos que presentan un mejor rendimiento corresponden a Random Forest y Support Vector Machines radial ya que logran predecir a la perfección la clasificación de los clientes obteniendo un 100 % de exactitud y 0 % en los errores. Por el contrario, se observa que el modelo que mayor error tipo I presenta es Árboles de decisión, seguido de los modelos Support Vector Machine lineal y polinomial, los cuales obtienen el peor rendimiento en cuanto a las métricas de exactitud, error tipo II y capacidad de discriminancia.

Dado que los modelos de Support Vector Machines lineal y polinomial no logran ajustarse bien a los datos de entrenamiento se puede deducir que al aplicarlos a datos desconocidos tampoco logran realizar buenas predicciones, sin embargo, para evaluar el verdadero desempeño de los modelos se deben analizar las métricas obtenidas a partir del conjunto de datos de validación presentadas en la Tabla 4.22.

En particular, de la Tabla 4.22 se puede observar que el error tipo I mayor se obtiene para Árboles de decisión, seguido de los modelos de Support Vector Machine lineal y polinomial, además, se puede concluir que estos 3 modelos tienen una capacidad regular para discriminar entre clientes buenos y malos ya que presentan un AUC entre 70 % y 80 %.

Adicionalmente, se observa que Support Vector Machine radial obtiene el peor desempeño, realizando predicciones aleatoriamente por lo que no es capaz de discrminar entre clientes buenos y malos (AUC = 50 %). De aquí se sigue que los modelos que presentaban el mejor rendimiento en el conjunto de entrenamiento disminuyen considerablemente su desempeño, por lo tanto, para revisar el posible sobreajuste de los modelos se procede a calcular la variación porcentual entre el conjunto de entrenamiento y validación obteniendo los resultados de la Tabla 4.23.

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Regresión logística	82,5 %	14,5 %	17,6 %	92,7 %	67,9 %
Análisis discriminante lineal	81,4 %	15,5 %	18,7 %	91,6 %	65,8 %
Árboles de decisión	91,0 %	38,0 %	7,3 %	78,7 %	54,7 %
Random Forest	83,6 %	16,0 %	16,4 %	92,4 %	67,6 %
Gradient Boosting	**83,1 %**	**11,0 %**	**17,2 %**	**93,9 %**	**71,8 %**
Extreme Gradient Boosting	83,6 %	14,5 %	16,5 %	92,8 %	69,0 %
SVM lineal	74,2 %	35,5 %	25,2 %	72,7 %	39,3 %
SVM polinomial	73,0 %	35,0 %	26,6 %	72,1 %	38,4 %
SVM radial	26,4 %	15,5 %	77,0 %	53,6 %	7,5 %

Tabla 4.22: Métricas de evaluación conjunto de validación

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Regresión logística	-1,7 %	-21,2 %	10,0 %	1,2 %	3,5 %
Análisis discriminante lineal	1,2 %	-10,9 %	-5,1 %	1,4 %	4,8 %
Árboles de decisión	0,0 %	-12,4 %	2,8 %	3,3 %	10,5 %
Random Forest	-16,4 %	100,0 %	100,0 %	-7,6 %	-32,4 %
Gradient Boosting	-4,0 %	-39,2 %	30,3 %	0,8 %	4,5 %
Extreme Gradient Boosting	-10,4 %	113,2 %	146,3 %	-5,7 %	-20,1 %
SVM lineal	2,8 %	7,6 %	-8,4 %	-3,2 %	-0,5 %
SVM polinomial	3,8 %	9,4 %	-10,1 %	-3,4 %	0,3 %
SVM radial	-73,6 %	100,0 %	100,0 %	-46,4 %	-92,5 %

Tabla 4.23: Variación porcentual entre las métricas del conjunto de entrenamiento y validación

En vista de que los valores de la Tabla 4.23 se interpretan según su signo, es decir, si el valor obtenido es positivo representa un incremento respecto a la métrica del conjunto de entrenamiento, por el contrario si el valor es negativo representa una disminución, se destaca el

aumento en los errores y la disminución en la capacidad discriminativa de los modelos asociados a Random Forest, Extreme Gradient Boosting y Support Vector Machines radial, por lo que se concluye que estos modelos presentan sobreajuste al conjunto de entrenamiento. Notar que el ejemplo más claro de sobreajuste es Support Vector Machine radial, es más si analizamos los resultados obtenidos anteriormente se tiene que este modelo es entrenado a partir de 14508 vectores de soporte, lo cual es equivalente al tamaño total de observaciones del conjunto de entrenamiento, esto significa que el algoritmo aprendió los patrones específicos del conjunto de entrenamiento y no logra generalizar los resultados a datos nuevos.

En resumen, se tiene que los modelos asociados a Árboles de decisión, Random Forest, Extreme Gradient Boosting, Support Vector Machines lineal, radial y polinomial no se consideran en el análisis comparativo, ya sea porque no logran diferenciar a los clientes malos de los buenos o debido a que se ajustaron demasiado bien a los datos de entrenamiento y no son capaces de generar predicciones de nuevos datos con precisión. Además, es importante recordar que la base de datos estudiada no cumple con los supuestos estadísticos necesarios para validar el modelo de Análisis discriminante lineal, por lo que también queda fuera del análisis comparativo.

De esta manera, el análisis se reduce a la comparación de los modelos de Regresión logística y Gradient Boosting, los cuales poseen una excelente capacidad de discriminación entre clientes buenos y malos, dado que ambos modelos obtienen un AUC sobre 90 % y un índice KS sobre 60 %. Sin embargo, de la Tabla 4.21 y 4.22 se puede observar que tanto para el conjunto de entrenamiento como para el conjunto de validación, el modelo que mejores resultados presenta es Gradient Boosting, obteniendo mayor exactitud al predecir la clasificación de las observaciones y un menor porcentaje de error que Regresión logística.

Finalmente, la Figura 4.8 muestra las curvas de ROC obtenidas para el conjunto de validación, con el fin de reforzar el análisis recientemente expuesto y visualizar la comparación de los algoritmos en términos del error tipo I (eje X) y las clasificaciones correctas (eje Y). De aquí, se puede observar claramente que los modelos asociados a Árboles de decisión y Support Vector Machines tienen la peor capacidad de discriminancia y que la curva más cercana al punto de 100 % sensibilidad y 100 % especificidad corresponde a la metodología Gradient Boosting.

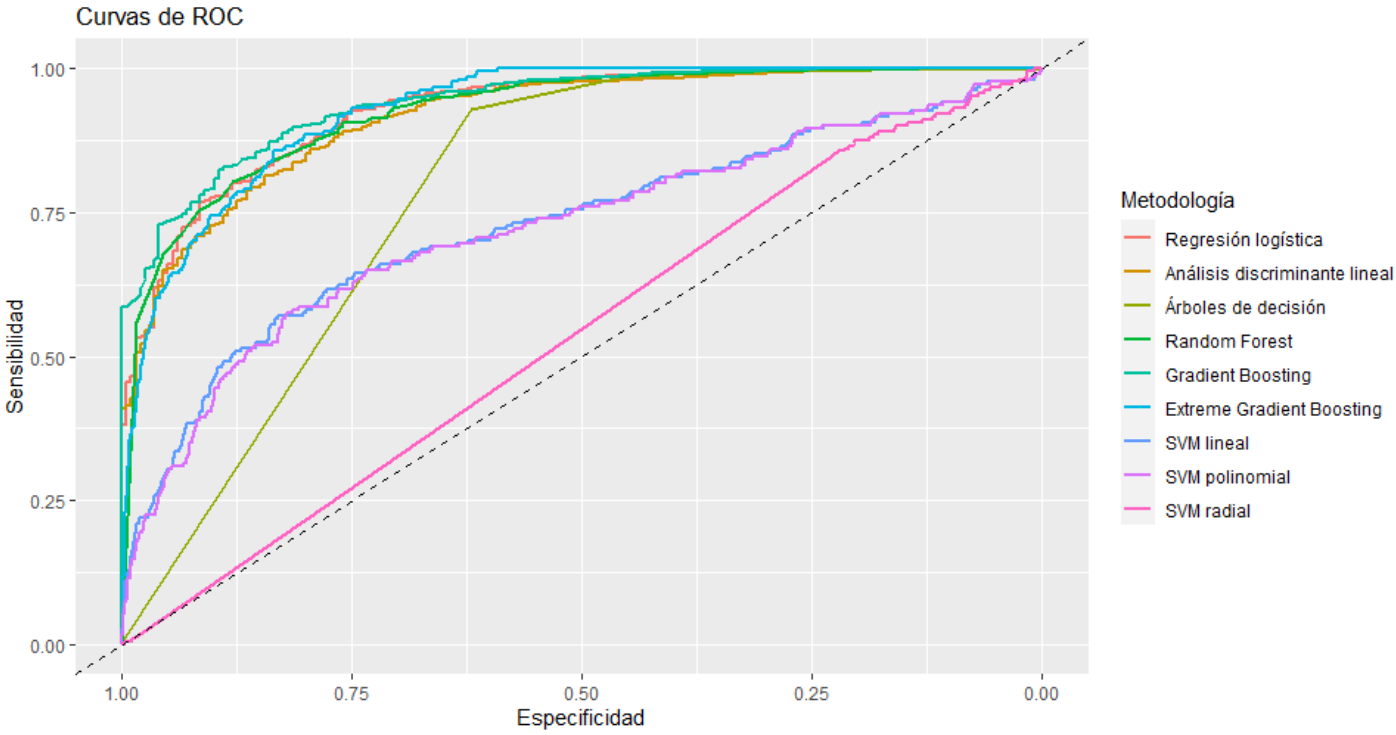


Fig. 4.8: Curvas de ROC conjunto de validación

5. Aplicación con datos reales

Luego del análisis e implementación de los modelos de Machine Learning al conjunto de datos estudiado, se creó una función en R llamada `modelos` que permite dividir la base de datos en un conjunto de entrenamiento y validación (80 % - 20 %) para entrenar los modelos de Machine Learning vistos en el capítulo anterior y finalmente obtener las métricas y sus curvas de ROC, donde el único argumento de entrada corresponde a una base de datos previamente preprocesada en formato SPSS Statistics (*.sav), es decir, una base que contiene solo las variables seleccionadas para entrenar los modelos y la variable de incumplimiento.

Por lo tanto, en este capítulo se verá un resumen de los resultados obtenidos de la aplicación de la función a 4 bases de datos de distintas entidades financieras proporcionadas por KPMG, con el objetivo de realizar una comparación con la actual metodología utilizada por las instituciones para estimar la probabilidad de incumplimiento, la cual corresponde a un modelo de regresión logística en base a una categorización de las variables seleccionadas utilizando indicadores tales como: el odds, las correlaciones, el Information Value (IV) y el bade rate (tasa de malos).

Es importante recordar que la información utilizada es de carácter confidencial, por lo tanto los resultados son presentados de manera de garantizar la privacidad de los datos y que no sea posible identificar a la institución o a los clientes correspondientes.

5.1. Institución financiera 1

A continuación, se presenta el ánalisis de los resultados obtenidos a partir de los modelos entrenados para la base de la primera institución financiera.

De la Tabla 5.1 se puede observar que para el conjunto de entrenamiento el modelo Random Forest presenta el mejor rendimiento, obtiendo mayor exactitud y una mayor capacidad de discriminancia en comparación con el modelo obtenido por la institución financiera. Por el contrario, se observa que los modelos Support Vector Machines presentan el peor desempeño en cuanto a las métricas de exactitud y errores tipo I y II, independientemente del Kernel utilizado, por lo tanto, estos modelos no serán considerados en el ánalisis comparativo.

Por otro lado, se observa de la Tabla 5.2 que para el conjunto de validación todos los modelos presentan un índice similar, sin embargo, el modelo asociado al Árbol de Decisión presenta la peor performance en cuanto a las métricas de exactitud, error tipo I e índices de discriminación, por lo que este método tampoco será considerado en el ánalisis comparativo.

Luego, se procede a calcular la variación porcentual entre el conjunto de entrenamiento y validación obteniendo los resultados de la Tabla 5.3, en la cual se observa que todos los modelos estadísticos disminuyen su exactitud y capacidad discriminativa, destacando en particular el aumento en los errores de los modelos asociados a las metodologías Random Forest y Extreme Gradient Boosting, por lo que se concluye que estas técnicas presentan sobreajuste al conjunto de entrenamiento y no serán consideradas en el análisis comparativo.

Además, dado que el modelo asociado a Análisis discriminante lineal necesita la verificación de supuestos estadísticos para que sea válido, no se considera en el análisis por pasimonia.

De esta manera, se obtiene que los modelos que mejores resultados presentaron fueron los modelos obtenidos por la Institución, Regresión logística y Gradient Boosting, los cuales poseen una capacidad de discriminación Regular de acuerdo a la valoración de los índices AUC y KS.

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Institución	66,7 %	28,9 %	35,9 %	72,4 %	35,3 %
Análisis discriminante lineal	68,5 %	35,6 %	29,2 %	72,8 %	35,3 %
Regresión logística	68,6 %	35,8 %	28,9 %	72,8 %	35,4 %
Árboles de decisión	67,0 %	38,7 %	29,7 %	68,5 %	31,6 %
Random Forest	94,2 %	8,6 %	4,2 %	97,2 %	87,2 %
Gradient Boosting	68,9 %	31,7 %	30,7 %	74,5 %	37,6 %
Extreme Gradient Boosting	83,5 %	18,8 %	15,2 %	91,5 %	66,1 %
SVM lineal	33,2 %	65,4 %	67,7 %	71,7 %	33,0 %
SVM polinomial	35,0 %	71,1 %	61,4 %	70,2 %	32,5 %
SVM radial	26,0 %	72,2 %	75,1 %	80,3 %	47,3 %

Tabla 5.1: Métricas de evaluación conjunto de entrenamiento Institución financiera 1

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Institución	66,2 %	30,8 %	35,5 %	71,0 %	33,7 %
Análisis discriminante lineal	68,1 %	34,5 %	30,5 %	72,5 %	35,0 %
Regresión logística	67,7 %	33,5 %	31,7 %	72,4 %	34,8 %
Árboles de decisión	64,9 %	41,9 %	31,4 %	66,5 %	26,8 %
Random Forest	68,6 %	32,5 %	30,8 %	73,4 %	36,7 %
Gradient Boosting	68,0 %	32,2 %	31,9 %	73,5 %	35,9 %
Extreme Gradient Boosting	67,8 %	36,0 %	30,1 %	71,0 %	34,0 %

Tabla 5.2: Métricas de evaluación conjunto de validación Institución financiera 1

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Institución	-0,7 %	6,6 %	-1,1 %	-1,9 %	-4,5 %
Análisis discriminante lineal	-0,6 %	-3,1 %	4,5 %	-0,4 %	-0,8 %
Regresión logística	-1,3 %	-6,4 %	9,7 %	-0,5 %	-1,7 %
Random Forest	-27,2 %	277,9 %	633,3 %	-24,5 %	-57,9 %
Gradient Boosting	-1,3 %	1,6 %	3,9 %	-1,3 %	-4,5 %
Extreme Gradient Boosting	-18,8 %	91,5 %	98,0 %	-22,4 %	-48,6 %

Tabla 5.3: Variación porcentual entre el conjunto de entrenamiento y de validación Institución financiera 1

5.2. Institución financiera 2

A continuación, se presenta el ánalisis de los resultados obtenidos a partir de los modelos entrenados para la base de la segunda institución financiera.

En general, de la Tabla 5.4 se puede observar que para el conjunto de entrenamiento, el modelo Random Forest presenta el mejor rendimiento, ya que, obtiene la mayor exactitud, menor error tipo I y II y posee una mayor capacidad de discriminancia que el modelo obtenido por la institución financiera. Por el contrario, los modelos Support Vector Machines, independiente del kernel utilizado, presentan la peor performance en cuanto a las métricas de exactitud y errores tipo I y II, por lo tanto estos modelos no serán considerados en el análisis comparativo.

De la Tabla 5.5, se observa que todos los modelos presentan un índice similar, sin embargo, el modelo asociado al Árbol de Decisión presenta el peor desempeño en las métricas de exactitud, error tipo I e índices de discriminación.

Luego, de la variación porcentual entre el conjunto de entrenamiento y validación de la Tabla 5.6 se observa que todos los modelos estadísticos aumentan su exactitud y su capacidad discriminativa, a excepción de los modelos Random Forest y Extreme Gradient Boosting, por lo que no serán considerados en el análisis debido a la pérdida en su capacidad discriminativa.

Además, dado que el modelo asociado a Análisis discriminante lineal necesita la verificación de supuestos estadísticos para que sea válido, tampoco se considera en el análisis por pasimonia.

De esta manera, se obtiene que los modelos que mejores resultados presentaron fueron los modelos obtenidos por la Institución, Regresión logística y Gradient Boosting, los cuales poseen un poder de discriminación razonable, de acuerdo a la valoración del índice KS.

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Institución	69,9 %	28,9 %	31,3 %	76,7 %	39,8 %
Análisis discriminante lineal	69,1 %	29,7 %	32,1 %	76,4 %	38,2 %
Regresión logística	69,5 %	29,5 %	31,5 %	76,6 %	39,0 %
Árboles de decisión	69,4 %	33,8 %	27,4 %	74,0 %	38,9 %
Random Forest	76,6 %	22,7 %	24,1 %	82,8 %	53,2 %
Gradient Boosting	70,1 %	29,5 %	30,3 %	77,2 %	40,2 %
Extreme Gradient Boosting	73,4 %	27,3 %	26,0 %	81,9 %	46,7 %
SVM lineal	30,8 %	71,1 %	67,3 %	76,4 %	38,4 %
SVM polinomial	55,3 %	40,3 %	49,0 %	57,1 %	10,7 %
SVM radial	24,8 %	78,4 %	72,0 %	82,0 %	50,5 %

Tabla 5.4: Métricas de evaluación conjunto de entrenamiento Institución financiera 2

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Institución	70,6 %	26,6 %	32,2 %	77,8 %	41,2 %
Análisis discriminante lineal	70,5 %	27,8 %	31,2 %	77,5 %	40,9 %
Regresión logística	70,7 %	27,4 %	31,2 %	77,7 %	41,4 %
Árboles de decisión	69,8 %	29,9 %	30,6 %	75,3 %	39,5 %
Random Forest	71,1 %	25,4 %	32,3 %	76,3 %	42,3 %
Gradient Boosting	71,1 %	28,7 %	29,1 %	78,2 %	42,2 %
Extreme Gradient Boosting	70,8 %	27,4 %	31,0 %	77,8 %	41,6 %

Tabla 5.5: Métricas de evaluación conjunto de validación Institución financiera 2

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Institución	1,0 %	-8,0 %	2,9 %	1,4 %	3,5 %
Análisis discriminante lineal	2,0 %	-6,4 %	-2,8 %	1,4 %	7,1 %
Regresión logística	1,7 %	-7,1 %	-1,0 %	1,4 %	6,2 %
Random Forest	-7,2 %	11,9 %	34,0 %	-7,9 %	-20,5 %
Gradient Boosting	1,4 %	-2,7 %	-4,0 %	1,3 %	5,0 %
Extreme Gradient Boosting	-3,5 %	0,4 %	19,2 %	-5,0 %	-10,9 %

Tabla 5.6: Variación porcentual entre el conjunto de entrenamiento y de validación Institución financiera 2

5.3. Institución financiera 3

A continuación, se presenta el ánalisis de los resultados obtenidos a partir de los modelos entrenados para la base de la tercera institución financiera.

De la Tabla 5.7 se puede observar que para el conjunto de entrenamiento el modelo Random Forest presenta un rendimiento casi perfecto. Por el contrario, los modelos de Support Vector Machines lineal y polinomial, presentan el peor rendimiento, por lo que no será considerados en el análisis comparativo.

Por otro lado, se puede observar de la Tabla 5.8 que el modelo asociado al Árbol de Decisión presenta la peor performance en las métricas del error tipo I e índices de discriminación, por lo tanto, este modelo tampoco será considerado.

Luego, de la variación porcentual entre los conjuntos de entrenamiento y validación obtenidos en la Tabla 5.9 se observa que todos los modelos estadísticos disminuyen su exactitud y su capacidad discriminativa y se destaca en particular el aumento en los errores de los modelos asociados a las metodologías Random Forest, Extreme Gradient Boosting y Support Vector Machines radial, por lo que se concluye que estas técnicas presentan sobreajuste al conjunto de entrenamiento y no serán consideradas en el análisis comparativo.

Además, dado que el modelo asociado a Análisis discriminante lineal necesita la verificación de supuestos estadísticos, no se considera en el análisis por parsimonia. Así, se obtiene que los modelos que mejores resultados presentaron fueron los modelos obtenidos por la Institución, Regresión logística y Gradient Boosting, los cuales poseen una buena capacidad de discriminación de acuerdo a la valoración de los índices AUC y KS.

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Institución	84,0 %	26,7 %	13,0 %	85,5 %	60,3 %
Análisis discriminante lineal	79,9 %	21,7 %	19,7 %	87,4 %	58,6 %
Regresión logística	80,4 %	22,6 %	18,8 %	87,2 %	58,6 %
Árboles de decisión	84,8 %	34,5 %	9,8 %	79,4 %	55,7 %
Random Forest	99,5 %	0,9 %	0,4 %	99,9 %	98,7 %
Gradient Boosting	82,0 %	18,6 %	17,8 %	89,6 %	63,6 %
Extreme Gradient Boosting	89,5 %	10,2 %	10,5 %	96,2 %	79,3 %
SVM lineal	31,7 %	74,0 %	66,8 %	69,3 %	40,7 %
SVM polinomial	29,0 %	67,9 %	71,8 %	70,0 %	39,7 %
SVM radial	86,5 %	18,1 %	12,3 %	88,5 %	69,7 %

Tabla 5.7: Métricas de evaluación conjunto de entrenamiento Institución financiera 3

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Institución	82,5 %	31,5 %	13,7 %	83,2 %	54,8 %
Análisis discriminante lineal	78,3 %	24,0 %	21,1 %	85,5 %	54,9 %
Regresión logística	78,2 %	24,6 %	21,0 %	85,5 %	54,4 %
Árboles de decisión	83,7 %	39,6 %	10,0 %	76,7 %	50,4 %
Random Forest	82,0 %	23,6 %	16,5 %	87,8 %	59,9 %
Gradient Boosting	80,3 %	21,5 %	19,2 %	87,8 %	59,3 %
Extreme Gradient Boosting	80,4 %	20,6 %	19,3 %	88,2 %	60,1 %
SVM radial	79,3 %	26,3 %	19,2 %	81,1 %	54,6 %

Tabla 5.8: Métricas de evaluación conjunto de validación Institición financiera 3

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Institución	-1,8 %	18,0 %	5,4 %	-2,7 %	-9,1 %
Análisis discriminante lineal	-2,0 %	10,6 %	7,1 %	-2,2 %	-6,3 %
Regresión logística	-2,7 %	8,8 %	11,7 %	-1,9 %	-7,2 %
Random Forest	-17,6 %	2522,2 %	4025,0 %	-12,1 %	-39,3 %
Gradient Boosting	-2,1 %	15,6 %	7,9 %	-2,0 %	-6,8 %
Extreme Gradient Boosting	-10,2 %	102,0 %	83,8 %	-8,3 %	-24,2 %
SVM radial	-8,3 %	45,3 %	56,1 %	-8,4 %	-21,7 %

Tabla 5.9: Variación porcentual entre el conjunto de entrenamiento y validación Institución financiera 3

5.4. Institución financiera 4

Finalmente, se presenta el análisis de los resultados obtenidos a partir de los modelos entrenados para la cuarta institución financiera.

En particular, de la Tabla 5.10 se observa que los modelos Support Vector Machines, presentan el peor desempeño en cuanto a las métricas de exactitud y errores tipo I y II, por lo tanto no serán considerados en el análisis comparativo.

En cuanto a los indicadores del conjunto de validación presentados en la Tabla 5.11 , se observa que todos lo modelos presentan métricas similares, sin embargo, el modelo asociado al Árbol de Decisión presenta el peor rendimiento, por lo que no será considerado en el ánalisis.

Luego, de la variación porcentual entre el conjunto de entrenamiento y validación presentada en la Tabla 5.12, se observa que todos los modelos estudiados aumentan su exactitud y su capacidad de discriminación, a excepción de Random Forest y Extreme Gradient Boosting, por lo tanto, no serán considerados en el análisis debidos a la pérdida en su poder discriminativo.

Adicionalmente, dado que el modelo asociado a Análisis discriminante lineal necesita la verificación de supuestos estadísticos para su validez, tampoco se considera en el análisis por parsimonia. Así, se obtiene que los modelos que mejores resultados presentaron fueron los modelos obtenidos por la Institución, Regresión logística y Gradient Boosting, los cuales poseen un poder de discriminación razonable de acuerdo a la valoración del índice KS.

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Institución	70,8 %	28,1 %	30,2 %	77,4 %	41,7 %
Análisis discriminante lineal	70,0 %	28,9 %	31,2 %	76,3 %	40,0 %
Regresión logística	70,3 %	27,4 %	32,0 %	76,7 %	40,6 %
Árboles de decisión	69,8 %	31,7 %	28,7 %	73,3 %	39,6 %
Random Forest	73,5 %	25,4 %	27,6 %	82,4 %	47,0 %
Gradient Boosting	71,1 %	26,1 %	31,7 %	77,8 %	42,2 %
Extreme Gradient Boosting	73,4 %	25,1 %	28,2 %	81,2 %	46,7 %
SVM lineal	30,0 %	72,4 %	67,7 %	76,8 %	40,1 %
SVM polinomial	33,2 %	65,0 %	68,5 %	71,7 %	33,5 %
SVM radial	26,1 %	75,7 %	72,1 %	81,1 %	47,9 %

Tabla 5.10: Métricas de evaluación conjunto de entrenamiento Institución financiera 4

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Institución	70,3 %	27,9 %	31,4 %	77,0 %	41,5 %
Análisis discriminante lineal	70,9 %	29,8 %	28,5 %	76,5 %	41,7 %
Regresión logística	70,9 %	28,5 %	29,7 %	76,8 %	41,8 %
Árboles de decisión	69,2 %	33,2 %	28,5 %	73,3 %	38,3 %
Random Forest	70,7 %	27,5 %	31,0 %	76,4 %	41,5 %
Gradient Boosting	71,2 %	25,4 %	32,1 %	77,4 %	42,5 %
Extreme Gradient Boosting	70,0 %	24,1 %	35,7 %	76,2 %	40,2 %

Tabla 5.11: Métricas de evaluación conjunto de validación Institución financiera 4

Modelo	Exactitud	Error tipo I	Error tipo II	AUC	KS
Institución	-0,7 %	-0,7 %	4,0 %	-0,5 %	-0,5 %
Análisis discriminante lineal	1,3 %	3,1 %	-8,7 %	0,3 %	4,2 %
Regresión logística	0,9 %	4,0 %	-7,2 %	0,1 %	3,0 %
Random Forest	-3,8 %	8,3 %	12,3 %	-7,3 %	-11,7 %
Gradient Boosting	0,1 %	-2,7 %	1,3 %	-0,5 %	0,7 %
Extreme Gradient Boosting	-4,6 %	-4,0 %	26,6 %	-6,2 %	-13,9 %

Tabla 5.12: Variación porcentual entre el conjunto de entrenamiento y validación Institución financiera 4

6. Conclusiones

En resumen, dado que el objetivo principal de esta memoria de título es la comparación de algoritmos de Machine learning, se realizó en primer lugar una revisión bibliográfica de la aplicación de los algoritmos que se utilizan generalmente para predecir la probabilidad de incumplimiento, evidenciando que han sido ampliamente estudiados debido al rápido crecimiento de la industria. Posteriormente se realizó una revisión teórica y metodológica de los algoritmos propuestos considerando las ventajas y desventajas de cada uno de ellos, para luego implementarlos al conjunto de datos detallado en la sección 3.1 y finalmente realizar una automatización para la aplicación de los modelos a conjuntos de datos reales.

En base al análisis de datos y a los resultados, se puede concluir que los modelos entrenados dependen de la calidad de los datos utilizados, verificando lo expuesto en la Introducción y en la sección 2.4, ya que por lo general se observó que las variables con mayor valor de información tuvieron mayor influencia en el aprendizaje de los algoritmos utilizados.

Adicionalmente, a partir de los resultados y los análisis realizados en la comparación de los modelos, tanto para la base de datos estudiada como para las bases de datos de las distintas instituciones financieras, se puede concluir que las metodologías más adecuadas para predecir la probabilidad de incumplimiento corresponden a la Regresión logística y Gradient Boosting, sin embargo, la elección del mejor modelo depende principlamente del punto de vista del negocio tomando en consideración las ventajas y desventajas de ambos algoritmos, en este sentido, cobra relevancia la interpretabilidad de los resultados ya que de acuerdo al Capítulo B1 del CNC de la CMF se define que: “La elección de los algoritmos, métodos o modelos utilizados para la estimación, se encuentran fundamentados, documentados de manera inteligible, tanto para los usuarios internos, como para las entidades externas y supervisores”. [1]

Por lo tanto, considerando el análisis estadístico y normativo se concluye que los modelos asociados a la Regresión logística proporcionan una mejor estimación de los clientes en incumplimiento ya que reflejan de manera conjunta los siguientes aspectos:

1. Sus métricas, en las muestras de entrenamiento y validación, son mejores que el promedio obtenido por las otras seis metodologías indicadas en la presente memoria.
2. No presenta sobreajuste al conjunto de entrenamiento.

3. Cumple con los supuestos teóricos de un modelo estadístico.

4. Es una metodología que estima parámetros fijos.

5. Adicionalmente tiene la ventaja que es un método conocido y ampliamente utilizado en la industria financiera.

Bibliografía

[1] *Compendio de Normas Contables para Bancos (Versión Año 2022)*, Comisión para el Mercado Financiero (CMF), Octubre 2021.

[2] Pablo Coloma, Richard Weber, José Guajardo, Jaime Miranda, *Modelos analíticos para el manejo del riesgo de crédito*, Trend Management, (2006).

[3] Wang Bao, Ning Lianjua, Kong Yue, *Integration of unsupervised and supervised machine learning algorithms for credit risk assessment*, Expert Systems with Applications, (2019).

[4] Miguel Biron y Víctor Medina, *Comparación de algoritmos de clasificación para el incumplimiento crediticio. Aplicación al sistema bancario chileno*, Superintendencia de Bancos e Instituciones Financieras de Chile (SBIF), (2018).

[5] I-Cheng Yeh, Che-hui Lien, *The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients*, Expert Systems with Applications 36: 2473–2480, (2009).

[6] T. Belloti y J. Crock, *Support vector machines for credit scoring and discovery of significant features*, Expert Systems with Applications 36:3302-3308, (2009).

[7] Dinesh Bacham, Janet Zhao, *Machine Learning: Challenges, Lessons, and Opportunities in Credit Risk Modeling*, Moody's Analitics, (2017).

[8] Peter Martey Addo, Dominique Guegan and Bertrand Hassani, *Credit Risk Analysis Using Machine and Deep Learning Models*, Revista Risks, (2018).

[9] Javier Espinosa-Zúñiga, *Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito*, Ingeniería Investigación y tecnología, (2020).

[10] *¿Qué es la CMF?*, Portal educativo de la Comisión para el Mercado Financiero. Recuperado de: `https://www.cmfchile.cl/educa/621/articles-25484_recurso_1.pdf`

[11] Thomas, L.C., Edelman, D.B., & Crook, J.N., *Credit scoring and its applications*, SIAM Monographs on Mathematical Modeling and Computation, Philadelphia, (2002).

[12] Park, Hyeoun-Ae, *An introduction to logistic regression: from basic concepts to interpretation with particular attention to nursing domain*, Journal of Korean Academy of Nursing, 43(2), 154-164, (2013).

[13] David W. Hosmer, Stanley Lemeshow, *Applied Logistic Regression*, Second Edition, John Wiley & Sons, (2000).

[14] T. Bellini, *IFRS 9 and CECL Credit Risk Modelling and Validation*, Academic Press, (2019).

[15] Bewick, V., Cheek, L., & Ball, J., *Statistics review 14: Logistic regression*, Critical Care (London, England), 9(1), 112-118, (2005).

[16] Peng, C. J., & So, T. H., *Logistic regression analysis and reporting: A primer*, Understanding Statistics: Statistical Issues in Psychology, Education, and the Social Sciences, 1(1), 31-70, (2002).

[17] *Regresión logística*, IBM. Recuperado de: `https://www.ibm.com/docs/es/spss-statistics/SaaS?topic=regression-logistic`

[18] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning with Applications in R*, Second Edition, Springer, (2013).

[19] Trevor Hastie, Robert Tibshirani, Jerome Friedman *The Elements of Statistical Learning*, Second Edition, Springer (2008).

[20] Joaquín Amat Rodrigo, *Análisis discriminante lineal (LDA) y análisis discriminante cuadrático (QDA)*, (2016). Recuperado de: `https://www.cienciadedatos.net/documentos/28_linear_discriminant_analysis_lda_y_quadratic_discriminant_analysis_qda`

[21] Ivo D. Dinov, *Data Science and Predictive Analytics*, Springer, (2018).

[22] Joaquín Amat Rodrigo, *Árboles de decisión con Python: regresión y clasificación*, (2020). Recuperado de: `https://www.cienciadedatos.net/documentos/py07_arboles_decision_python.html`

[23] Breiman, L., *Random forests*, Machine Learning vol. 45: 5–32, (2001).

[24] J. H. Friedman., *Greedy function approximation: A gradient boosting machine*, Annals of Statistics, 29(5):1189–1232, (2001).

[25] Pedregosa *et al.*, *Scikit-learn: Machine Learning in Python, 1.11. Ensemble methods*, Journal of Machine Learning Research vol 12. Recuperado de `https://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting`

[26] Josh Starmer, *Gradient Boost Part 4: Classification Details*, (2019). Recuperado de `https://statquest.org/gradient-boost-part-4-classification-details/`

[27] J. H. Friedman., *Stochastic Gradient Boosting*, Computational Statistics & Data Analysis 38(4):367-378, (2002).

[28] Aratrika Pal, *Gradient Boosting Trees for Classification: A Beginner's Guide*, (2020). Recuperado de `https://affine.ai/gradient-boosting-trees-for-classification-a-beginners-guide/`

[29] Chen T. & Guestrin C., *XGBoost: A Scalable Tree Boosting System*, arXiv:1603.02754v3, (2016).

[30] Xin Yu Liew, Nazia Hameed, Jeremie Clos, *An investigation of XGBoost-based algorithm for breast cancer classification*, Machine Learning with Applications vol 6, (2021).

[31] Parashar, J., Sumiti, & Rai, M., *Breast cancer images classification by clustering of ROI and mapping of features by CNN with XGBOOST learning*, Materials Today: Proceedings, (2020).

[32] Corey Wade, *Hands-On Gradient Boosting with XGBoost and Scikit-learn: XGBoost and scikit-learn*, Packt Publishing Ltd, (2020).

[33] Bernhard E. Boser, Isabelle M. Guyon, Vladimir N. Vapnik, *A Training Algorithm for Optimal Margin Classifiers*, Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, (1992).

[34] Felipe Tobar, *Notas de clase: Aprendizaje de Máquinas*, Centro de Modelamiento Matemático, Universidad de Chile, (2021). Recuperado de `github.com/GAMES-UChile/Curso-Aprendizaje-de-Maquinas`

[35] Felipe Bravo *Clasificación Support Vector Machines*, Ciencias de la Computación, Universidad de Chile, (2010). Recuperado de `https://github.com/dccuchile/CC5206/blob/master/slides/Clase_8_clasi_SVM.pdf`

[36] Cynthia Rudin, *15.097 Prediction: Machine Learning and Statistics*, Massachusetts Institute of Technology: MIT OpenCourseWare, (2012). Recuperado de `https://ocw.mit.edu`

[37] Sebastián Maldonado, Richard Weber, *Modelos de Selección de Atributos para Support Vector Machines*, Revista Ingeniería de Sistemas, Vol XXVI, (2012).

[38] Pedregosa *et al.*, *Scikit-lern: Machine Learning in Python, 1.4. Support Vector Machines*, Journal of Machine Learning Research vol 12. Recuperado de `https://scikit-learn.org/stable/modules/svm.html#kernel-functions`

[39] S.Archana *et al.*, *Survey of Classification Techniques in Data Mining*, International Journal of Computer Science and Mobile Applications, Vol.2 Issue. 2, 65-71, (2014).

[40] Sunil Ray, *Understanding Support Vector Machine(SVM) algorithm from examples (along with code)*, Analytics Vidhya, (2017). Recuperado de `https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/`

[41] *Curso Machine Learning with Python Essencials: II. Aprendizaje supervisado parte I*, Virtual Labx, (2021).

[42] Joaquín Amat Rodrigo, *Machine Learning con R y caret*, (2020). Recuperado de: `https://www.cienciadedatos.net/documentos/41_machine_learning_con_r_y_caret`

[43] Jaime Cerda y Lorena Cifuentes, *Uso de curvas ROC en investigación clínica. Aspectos teórico-prácticos*, Revista Chilena de Infectología 29 (2): 138-141, (2012).

[44] Ana Rocío del Valle Benavides, *Curvas ROC (Receiver-Operating-Characteristic) y sus aplicaciones* (Trabajo Fin de Grado), Déposito de Investigación Universidad de Sevilla (2017).

[45] Elizabeth Mays, *Handbook of credit scoring*, Glenlake Publishing Company, (2001).

[46] Jorge Bacallao Guerra y Jorge Bacallao Gallestey, *Imputación múltiple en variables categóricas usando data augmentation y árboles de clasificación*, Revista investigación operacional, vol 31, No. 2, 133-139, (2010).

[47] Yubar Marín, *Imputación de datos*, Rpubs. Recuperado de: `https://rpubs.com/ydmarinb/429757`

[48] Rubin, D. & Schenker, N., *Multiple imputation for interval estimation from simple random samples with ignorable nonresponse*, Journal of the American Statical Association, vol 81, No. 394, (1986).

[49] Fernando Medina, Marco Galván, *Imputación de datos: teoría y práctica*, CEPAL - Serie Estudios estadísticos y prospectivos No. 54 (2010).

[50] Richard M. Heiberger, Burt Holland, *Statistical Analysis and Data Display, An Intermediate Course with Examples in R*, Second Edition, (2015).

[51] Flores Tapia, Carlos Ernesto; Flores Cevallos, Karla Lissette, *Pruebas para comprobar la normalidad de datos en procesos productivos: Anderson Darling, Ryan-Joiner, Shapiro-Wilk y Kolmogorov-Smirnov*, Societas, Revista de Ciencias Sociales y Humanísticas, Universidad de Panamá, (2021).

[52] Joaquín Amat Rodrigo, *Análisis de normalidad: gráficos y contrastes de hipótesis*, (2016). Recuperado de: https://www.cienciadedatos.net/documentos/8_analisis_normalidad#Test_de_Shapiro-Wilk

[53] Selcuk Korkmaz, Dincer Goksuluk and Gokmen Zararsiz, *MVN: An R Package for Assessing Multivariate Normality*, The R Journal Vol. 6/2, (2014).

[54] Richard A.Johnson, Dean W. Wichern, *Applied Multivariate Statistical Analysis*, 6th Edition, Prentice Hall, (2007).

[55] Charles Zaiontz, *Box's M Test Basic Concepts*. Recuperado de: https://www.real-statistics.com/multivariate-statistics/boxs-test/boxs-test-basic-concepts/

A. Anexo

A.1. Librerías utilizadas

En el cuadro a continuación se presentan las librerías utilizadas en el software R para cada metodología.

Metodología	Función	Librería
Regresión Logística	`glm()`	`stats`
Análisis discriminante lineal	`lda()`	`MASS`
Árboles de decisición	`rpart()`	`rpart`
Random Forest	`randomForest()`	`randomForest`
Gradient Boosting	`gbm()`	`gbm`
Extreme Gradient Boosting	`xgboost()`	`xgboost`
Support Vector Machines	`svm()`	`e1071`

Tabla A.1: Librerías utilizadas para el entrenamiento de los algoritmos

A continuación se especifican los parámetros utilizados para efectos de entrenar cada uno de los modelos presentados.

A.1.1. Regresión logística

Los principales agumentos utilizados para entrenar el algoritmo de regresión logística son los siguientes:

- **fórmula:** Expresión de la forma $y \sim x_1 + x_2 + ... + x_n$, donde la variable de respuesta es el factor de agrupación y el lado derecho especifica las variables explicativas.
- **data:** Representa el conjunto de datos del que se tomarán preferentemente las variables especificadas en la fórmula.
- **family:** : Especifica la distribución de la variable de respuesta y la función de enlace que se utilizará en el modelo, en este caso, la distribución es binomial y la función de enlace

corresponde a la función logística.

- **method:** Método que se utilizará para ajustar el modelo. El método predeterminado "glm.fitütiliza mínimos cuadrados ponderados iterativamente.

A.1.2. Análisis discriminante lineal

Los principales argumentos utilizados para entrenar el algoritmo de análisis discriminante lineal son los siguientes:

- **fórmula:** Expresión de la forma $y \sim x_1 + x_2 + ... + x_n$, donde la variable de respuesta es el factor de agrupación y el lado derecho especifica las variables explicativas.
- **data:** Representa el conjunto de datos del que se tomarán preferentemente las variables especificadas en la fórmula.
- **prior:** Son las probabilidades previas de pertenencia a una clase. En este caso, se utilizan las proporciones de clase para el conjunto de entrenamiento.

A.1.3. Árboles de decisión

Los principales argumentos utilizados para entrenar el algoritmo de árboles de decisión son los siguientes:

- **fórmula:** Expresión de la forma $y \sim x_1 + x_2 + ... + x_n$, donde la variable de respuesta es el factor de agrupación y el lado derecho especifica las variables explicativas.
- **data:** Representa el conjunto de datos del que se tomarán preferentemente las variables especificadas en la fórmula.
- **params:** Corresponde a los parámetros opcionales para la función de división. En este caso el criterio de división predeterminado corresponde al índice de Gini.

A.1.4. Random Forest

Los principales argumentos utilizados para entrenar el algoritmo de random forest son los siguientes:

- **fórmula:** Expresión de la forma $y \sim x_1 + x_2 + ... + x_n$, donde la variable de respuesta es el factor de agrupación y el lado derecho especifica las variables explicativas.
- **data:** Representa el conjunto de datos del que se tomarán preferentemente las variables especificadas en la fórmula.
- **ntree:** Especifica el número de árboles.
- **mtry:** Número de variables muestreadas aleatoriamente como candidatas en cada división. El valor por defecto para clasificación es $\sqrt{p}$ donde p es el número de variables explicativas.
- **nodesize:** Tamaño mínimo de los nodos terminales. El valor por defecto es diferente para clasificación (1) y regresión (5).

A.1.5. Gradient Boosting

Los principales argumentos utilizados para entrenar el algoritmo de gradient boosting son los siguientes:

- **fórmula:** Expresión de la forma $y \sim x_1 + x_2 + ... + x_n$, donde la variable de respuesta es el factor de agrupación y el lado derecho especifica las variables explicativas.
- **data:** Representa el conjunto de datos del que se tomarán preferentemente las variables especificadas en la fórmula.
- **distribution**: Especifica el nombre de la distribución a usar. En este caso, la distribución utilizada es bernoulli ya que la variable de respuesta solo tiene 2 valores únicos.
- **n.trees**: Número de árboles a ajustar. El valor por defecto es 100.
- **shrinkage:** Parámetro de contracción aplicado a cada árbol en la expansión, también conocido como tasa de aprendizaje; normalmente, 0,001 a 0,1 funciona, pero una tasa de aprendizaje menor normalmente requiere más árboles. El valor predeterminado es 0,1.

A.1.6. Extreme Gradient Boosting

Los principales argumentos utilizados para entrenar el algoritmo de extreme gradient boosting son los siguientes:

- **data:** Representa el conjunto de datos de entrenamiento, donde se acepta solo una matriz xgb.DMatrix como entrada.
- **objective:** Especifica la tarea y el objetivo de aprendizaje correspondiente, definiendo la función de interés. En este caso, se utiliza la función objetivo para clasificación binaria *binary:logistic*.
- **nrounds:** Número máximo de iteraciones boosting.
- **max.depth:** Máxima profundidad del árbol (predeterminado: 6).
- **eta:** Tasa de aprendizaje (predeterminado: 0,3).

A.1.7. Support Vector Machine

Los argumentos utilizados para entrenar el algoritmo de Support Vector Machine son los siguientes:

- **fórmula:** Expresión de la forma $y \sim x_1 + x_2 + ... + x_n$, donde la variable de respuesta es el factor de agrupación y el lado derecho especifica las variables explicativas.
- **data:** Representa el conjunto de datos del que se tomarán preferentemente las variables especificadas en la fórmula.
- **cost:** Es el costo de violación de las restricciones, es decir, es la constante c del término de regularización de la fórmula de Lagrange. El valor predeterminado es $c = 1$.
- **kernel:** Especifica el kernel utilizado para entrenar y predecir.

  Según el tipo de kernel se consideran los siguientes parámetros:

  - **degree:** Parámetro necesario para el kernel polinomial, el valor por defecto es 3.
  - **gamma:** Parámeto necesario para todos los tipos de kernel excepto el lineal, el valor por defecto es $1/(\textit{dimensión de los datos})$.

- **coef0**: Parámetro necesario para el kernel polinomial, el valor por defecto es 0.

- **probability:** Expresión lógica que indica si el modelo debe permitir predicciones de probabilidad.