



PROYECTO FINAL

Churn Modelling con Redes Neuronales

José Roberto
Torres Bello

Proyecto II

Profesor: María Fernanda Sánchez Puig
Ayudante: Sergio Miguel Fernández Martínez

enero 2022

Facultad de Ciencias
Universidad Nacional Autónoma de México

Índice general

1	Introducción	1
1.1	Objetivo	2
2	Marco Teórico Parte I	7
2.1	Atributos	8
2.1.1	Entendimiento y Preparación de los Datos	8
2.2	Valores Faltantes	9
3	Marco Teórico Parte II	13
3.1	Regresión Logística	14
3.1.1	Ventajas de la regresión logística	17
3.1.2	Desventajas de la regresión logística	18
3.2	Perceptrón Multicapa	18
4	Resultados	25
4.1	Análisis Exploratorio de Datos	25
4.1.1	Construcción del Modelo	30
4.1.2	Evaluación del Modelo	32
4.1.3	Curva ROC	34
5	Conclusiones	37
	Bibliografía	39

Capítulo 1

Introducción

A las empresas no les gusta perder clientes valiosos en particular a los bancos. Los clientes están directamente relacionados con las ganancias, las instituciones financieras deben evitar la pérdida de clientes mientras adquieren nuevos clientes.

El Churn (abandono) es uno de los más grandes problemas en cualquier empresa que ofrezca algún bien o servicio. Dado que es mucho más costoso atraer clientes nuevos que retener los existentes se deben crear estrategias que permitan de manera proactiva predecir y prevenir el Churn, permitiendo a su vez la fidelización del cliente. En este trabajo se describe el problema de abandono y se elabora un modelo predictivo de Churn.

Las técnicas de aprendizaje automático o Machine Learning suelen ser herramientas bastante efectivas para afrontar este problema, permitiéndo determinar qué clientes tienen mayor probabilidad de cancelar un producto, dejar de usar un servicio, etc. Identificar a los clientes que se van a ir es importante, pero es igual de importante qué hacer después con dichos clientes para que no se vayan.

La pérdida de clientes es un problema fundamental para las empresas.

Se define como la pérdida de clientes porque se trasladan a la competencia.

- Una empresa debe tener herramientas para tratar de predecir el comportamiento de rotación de los clientes y que le de una una visión muy valiosa para retener y aumentar su base de clientes.
- Uno de los enfoques más directos y efectivos para mantener a los clientes actuales es que la empresa debe poder prever los posibles abandonos a tiempo y reaccionar rápidamente.

Harvard Business Review cree que al reducir la tasa de deserción de clientes en un



Figura 1.1: Abandono de Clientes.

5 %, las empresas pueden aumentar las ganancias entre un 25 % y un 85 %, mientras que Business Week cree que las ganancias aumentarán en un 140 %.



Para el proyecto se hará uso de una dataset llamado **Bank Customer Churn Prediction**, es un dataset público que se puede consultar en <https://www.kaggle.com/kmalit/bank-customer-churn-prediction/data> perteneciente al repositorio: <https://www.kaggle.com/>.

1.1. Objetivo

El objetivo del presente proyecto es desarrollar un modelo Churn de clientes usando técnicas de Machine Learning que permita, de manera proactiva, predecir el abandono de clientes por lo cual nos enfrentamos a un problema de clasificación binaria.

¿Qué es Churn Modelling? Es un modelo predictivo que estima, a nivel de clientes individuales, la propensión (o susceptibilidad) que tienen a irse.



¿Por qué es importante?

- 1. Las empresas ahorran dinero en marketing.
- 2. Repetir compras de clientes habituales significa repetir beneficios.
- 3. Publicidad gratuita de boca en boca.
- 4. Los clientes retenidos proporcionan comentarios valiosos.

Beneficios:

- Identificar problemas de la empresa.
- Mejorar la reputación de la marca.
- Aumentar los ingresos.
- Primero, se comienza cargando y analizando el conjunto de datos.
- Se verá que el conjunto de datos tiene una combinación de características numéricas y categóricas, que contiene valores de diferentes rangos.
- Se preprocesa el conjunto de datos para asegurarse de que el modelo de aprendizaje automático que se eligió pueda hacer buenas predicciones.
- Una vez que el conjunto de datos esté en buena forma, se hará un análisis exploratorio de datos.
- Finalmente, se creará un modelo de aprendizaje automático que puede predecir si un cliente abandonará o no la relación con el banco.

Técnicamente, es un clasificador binario que divide a los clientes en dos grupos (clases): los que se van y los que no. Es ventajoso para los bancos saber qué lleva a un cliente a la decisión de dejar la empresa.

El dataset a analizar se encuentra en formato CSV en kaggle pero pero se ha alojado en un repositorio creado en Github esto con la finalidad de que el dataset sea portable, replicable y puesta en producción de una forma rápida. Se puede consultar en https://raw.githubusercontent.com/jrtorresb/Proyecto_Final_FC_Proyecto_II/main/Churn_Modelling.csv.



Figura 1.2: GitHub.

La metodología utilizada será la CRISP-DM que se puede visualizar en la figura 1.3.

Capítulo 2

Marco Teórico Parte I

Como se mencionó en la introducción, se hará uso del dataset llamado **Bank Customer Churn Prediction**, es un dataset público que se puede consultar en <https://www.kaggle.com/kmalit/bank-customer-churn-prediction/data> perteneciente al repositorio: **KAGGLE**.

Este archivo se refiere a clientes de un banco que abandonaron la relación con el mismo.

El término dataset es una representación de datos conocido en español como conjunto de datos o serie de datos. Esta representación viene dada en una única tabla de base de datos o matriz de datos y se podría definir como una representación de datos residentes en memoria. El conjunto de datos se almacena por columnas y filas, siendo cada columna una variable (atributo como color, talla, edad, género...) y cada fila representa a un miembro determinado del conjunto de datos. La unión de todas las filas y columnas proporciona el conjunto de todos los valores que pueden tener las variables. El tipo de datos que pueden representar puede ser tanto texto, números o multimedia, por ejemplo.

Este conjunto de datos es interesante porque hay una buena combinación de atributos: continuo, entero, nominal con un número reducido de valores y nominal con un número mayor de valores.

La base se conforma de 10,000 registros donde cada registro es la información de un cliente que pudo o no abandonar la relación con el banco.

2.1. Atributos

2.1.1. Entendimiento y Preparación de los Datos

La base cuenta con 14 atributos los cuales se enlistan a continuación con su tipo de dato:

- RowNumber: Número de fila de tipo entero.
- CustomerId: Id del cliente, de tipo entero.
- Surname: Apellido del cliente, de tipo object.
- CreditScore: Calificación del cliente, de tipo entero.
- Geography: Nacionalidad del cliente, de tipo object.
- Gender: Género del cliente, de tipo object.
- Age: Edad del cliente, de tipo entero.
- Tenure: Estado del cliente, de tipo entero.
- Balance: Balance en la cuenta del cliente, de tipo flotante.
- NumOfProducts: Número de productos que tiene el cliente con el banco, de tipo entero.
- HasCrCard: Si tiene asociada una tarjeta de crédito, de tipo entero.
- IsActiveMember: Si el cliente es activo, de tipo entero.
- EstimatedSalary: Salario estimado del cliente, de tipo flotante.
- Exited: Si el cliente abandonó al banco (1) o no (0) de tipo entero.

Nos encontramos frente a un problema de aprendizaje supervisado, la variable objetivo, es decir, la variable a predecir es la columna **Exited** la cual se conforma de dos categorías: 0, 1 (atributo de clase) donde:

- **0** El cliente no abandonó la relación con el banco.
- **1** El cliente abandonó la relación con el banco..

El utilizar métodos de aprendizaje automático o machine learning para predecir el abandono de clientes conduce a un proceso más rápido que al hacerlo de manera manual, dicha tarea se puede automatizar y en cuestión de minutos el banco puede saber si un cliente es propenso a abandonar la relación para así tomar medidas y tratar de retenerlo, lo cual representa un ahorro para los bancos optimizando tiempo y costos.

Lo primero que se puede hacer es eliminar los atributos que no sean relevantes para el análisis, tal es el caso de RowNumber, CustomerId y Surname por lo cual ya no se considerarán desde ahora.

2.2. Valores Faltantes

Es común trabajar con bases de datos que no están completas, es decir, se tiene la presencia de datos faltantes, esto suele ser muy común por diversas razones, es por ello que se tienen que aplicar técnicas de imputación e inclusive eliminar los registros que incluyen datos faltantes.

En la figura [2.1](#) se puede observar un mapa de calor de la base en donde se pintan de color amarillo los datos faltantes.

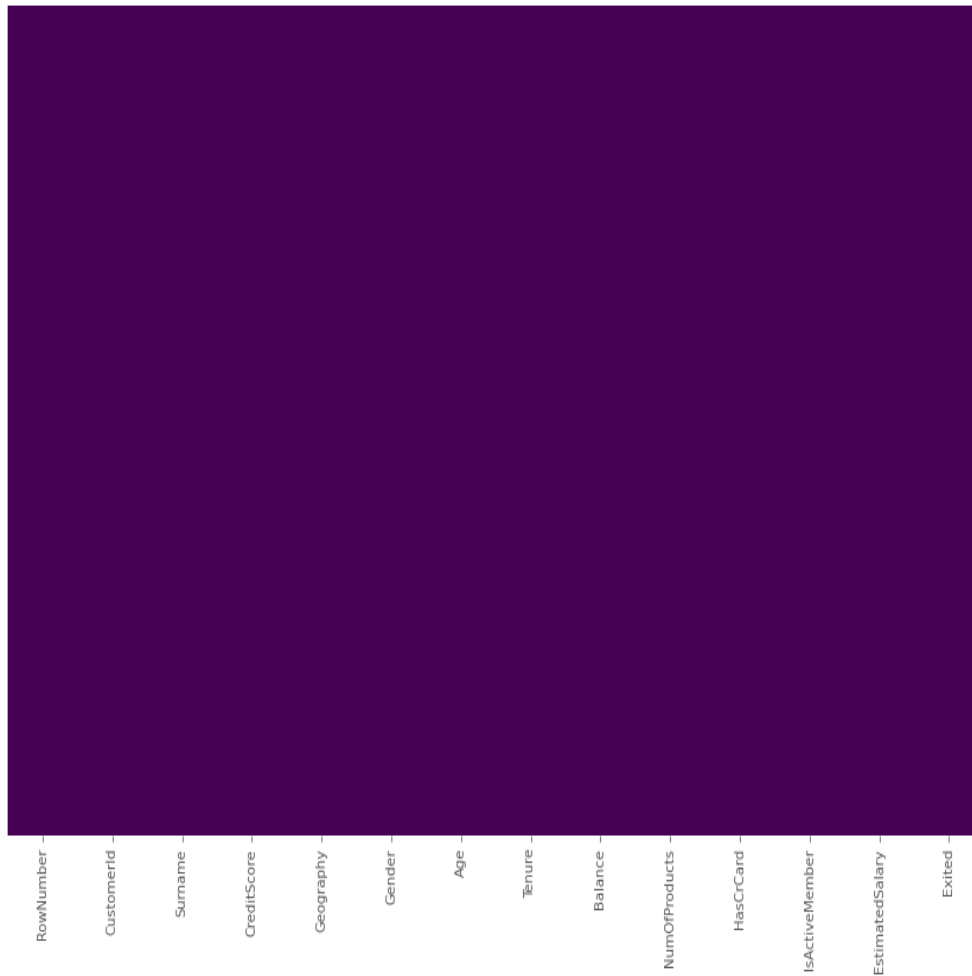


Figura 2.1: Mapa Datos Faltantes.

Afortunadamente en la base no hay datos faltantes por lo cual podemos avazar con el análisis.

Ignorar los datos faltantes puede afectar en gran medida el rendimiento de un modelo de aprendizaje automático, el modelo puede perder información sobre el conjunto de datos que puede ser útil para su entrenamiento.

	total	pctg
CreditScore	0	0.0
Geography	0	0.0
Gender	0	0.0
Age	0	0.0
Tenure	0	0.0
Balance	0	0.0
NumOfProducts	0	0.0
HasCrCard	0	0.0
IsActiveMember	0	0.0
EstimatedSalary	0	0.0
Exited	0	0.0

Figura 2.2: Número Datos Faltantes.

El total y porcentaje de datos faltantes se pueden visualizar en la figura 2.2 y notamos que no hay datos faltantes, si existieran datos faltantes para evitar este problema, se imputar los valores faltantes con una estrategia llamada imputación de la media que aplica únicamente para variables numéricas. La técnica se trata de imputar los valores faltantes con la media del atributo. Para variables categóricas se pueden imputar los valores faltantes con los valores más frecuentes presentes en las columnas respectivas. Esta es una buena práctica cuando se trata de imputar valores faltantes para datos categóricos en general. Haciendo uso de estas técnicas de imputación se tendría una base de datos completa.

Capítulo 3

Marco Teórico Parte II

En este capítulo se detallan los métodos utilizados, una síntesis de cómo funciona el algoritmo.

Los algoritmos de clasificación son capaces de clasificar de manera automática nuevos eventos a partir del conocimiento de eventos pasados, para que esto sea posible se debe contar con información suficiente para identificar las características que permiten que este se ubique en una etiqueta u otra. Como se dijo previamente, el enfoque de este proyecto busca profundizar en el funcionamiento de un algoritmo de clasificación binario el cual determina si un cliente tiene propensión hacia el Churn a partir del conocimiento de las características de aquellos que ya han hecho Churn.

Las técnicas avanzadas de aprendizaje automático (ML) y ciencia de datos (DS) pueden aprender del comportamiento pasado del cliente y los factores desencadenantes externos que llevaron a la deserción y utilizar este aprendizaje para predecir la ocurrencia futura de un evento similar a una deserción.

El aprendizaje supervisado es cuando se tienen variables de entrada X y una variable de salida Y y se utiliza un algoritmo para aprender la función f de mapeo de la entrada a la salida, midiendo el ajuste mediante una función de pérdida.

$$Y = f(X)$$

El objetivo es aproximar la función de mapeo f tan bien que cuando se tengan nuevos datos de entrada X se puedan predecir las variables de salida Y para esos datos.

3.1. Regresión Logística

La Regresión Logística es un algoritmo de aprendizaje supervisado y se utiliza para clasificación. En términos de este problema particular, este método trata de explicar la probabilidad de que un cliente abandone o no la relación con el banco en función de una serie de variables explicativas. El valor principal de éste método reside en la buena interpretabilidad, otra ventaja reside en la modelización de probabilidades y en el hecho de que los modelos de regresión logística son menos sensibles a outliers.

La variable dependiente presenta dos categorías que representan la ocurrencia y la no ocurrencia del acontecimiento definido, en este caso el abandono o no abandono condiciándose con los valores uno y cero respectivamente. En lo que se refiere a las variables explicativas o predictoras, pueden ser tanto numéricas como categóricas.

El modelo expresa la variable dependiente como la ocurrencia o no de un acotenci-
miento en términos de probabilidad, haciendo uso de la función logística para estimar la probabilidad de que ocurra el acontecimiento mediante la formulación:

$$\pi_i = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n}} \quad (3.1)$$

siendo π_i la probabilidad de pertenecer a las clase buena y x_i las variables explicativas o características de un cliente.

Puesto que el modelo anterior no es lineal respecto a las variables independientes, se considera la inversa de la función logística, a lo que se llama **logit**, definiéndose como en cociente entre la probabilidad de que ocurra un acontecimiento y la probabilidad de que no ocurra, que es su complemento, como se puede observar:

$$g(x) = \ln \left(\frac{\pi_i}{1 - \pi_i} \right) = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n \quad (3.2)$$

Dicha formulación hace más clara la interpretación del modelo y de sus coeficientes, que reflejan el cambio en el logit correspondiente a un cambio unitario en la variable independiente.

La probabilidad π_i obtenida por la ecuación 3.2 es el límite de la clasificación. El cliente es propenso a abandonar si es mayor a 0.5 o no propenso si es menor a 0.5. En este tipo de aplicación financiera, se puede asociar el término $\beta_i x_i$ a la calidad del cliente.

Se puede llegar a la regresión logística partiendo de una regresión lineal. Considere una regresión lineal con una variable independiente X y una variable dependiente y con las siguientes características:

$$y = \alpha + \beta \cdot X \quad (3.3)$$

- $y \in \{0, 1\}$
- $X \in [-\infty, \infty]$

Si se considera que P es la probabilidad condicionada de éxito o de fracaso condicionada a la presencia de la variable X .

- $P \in [0, 1]$
- $P = \alpha + \beta \cdot X$

$$\frac{P}{1-P} = \alpha + \beta \cdot X \in [0, +\infty]$$

$$\ln\left(\frac{P}{1-P}\right) = \alpha + \beta \cdot X \in [-\infty, +\infty]$$

$$\begin{cases} \frac{P}{1-P} \in [0, 1] \Rightarrow \ln\left(\frac{P}{1-P}\right) \in [-\infty, 0] \\ \frac{P}{1-P} \in [1, \infty] \Rightarrow \ln\left(\frac{P}{1-P}\right) \in [0, \infty] \end{cases}$$

$$\ln\left(\frac{P}{1-P}\right) = \alpha + \beta \cdot X$$

$$\frac{P}{1-P} = e^{\alpha + \beta \cdot X}$$

$$P(y|X) = \frac{1}{1 + e^{-(\alpha + \beta \cdot X)}} \quad (3.4)$$

- Si $a + bX$ es muy pequeño (negativo), entonces P tiende a 0.
- Si $a + bX = 0$, $P = 0,5$
- Si $a + bX$ es muy grande (positivo), entonces P tiende a 1.

En la regresión logística múltiple el caso es análogo al tener más de una variable independiente:

$$P(y|X) = \frac{1}{1 + e^{-(\alpha + \sum_{i=1}^n \beta_i \cdot X_i)}} \quad (3.5)$$

Donde P es la probabilidad condicional de éxito o de fracaso condicionada la presencia del vector de características X .

La idea del algoritmo es encontrar los parámetros α y β_i que mejor se ajusten al conjunto de datos, para hacerlo se implementa el método de la máxima verosimilitud para la regresión logística, en donde se define la función de entorno $L(b)$ (también llamada función de pérdida).

$$L(\beta) = \sum_{i=1}^n P_i^{y_i} (1 - P_i)^{1-y_i} \quad (3.6)$$

En donde se calculan las probabilidades para cada observación

$$P_i = P(x_i) = \frac{1}{1 + e^{-\sum_{j=0}^k \beta_j \cdot x_i}} \quad (3.7)$$

Se calcula la matriz diagonal W

$$W = \text{diag}(P_i \cdot (1 - P_i))_{i=1}^n \quad (3.8)$$

Se utilizan distintos algoritmos para la solución, uno de ellos es el método de Newton-Raphson.

$$\beta_{n+1} = \beta_n - \frac{f(\beta_n)}{f'(\beta_n)}$$

$$f(X) = X(Y - P)$$

$$f'(X) = XWX^T$$

Cabe mencionar que también se puede definir un umbral para la clasificación.

$$\varepsilon \in (0, 1), Y_p = \begin{cases} 0 & \text{si } p \leq \varepsilon \\ 1 & \text{si } p > \varepsilon \end{cases}$$

Como se puede observar, la regresión logística lleva en el núcleo de su método la función sigmoide; esta función es una curva en forma de S, que puede tomar cualquier

número real y dar como resultado cualquier número entre cero y uno.

En el caso de machine learning, la función sigmoide relaciona la variable dependiente con las variables independientes; es una curva que puede tomar cualquier valor entre 0 y 1 y nunca valores por fuera de estos límites, así la ecuación que define la función sigmoidea es:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.9)$$

Una de las particularidades de esta función es de que cumple con la ecuación diferencial $f'(x) = f(x)(1 - f(x))$ por lo cual es fácil de evaluar su derivada para los problemas de optimización que involucran la primera derivada.

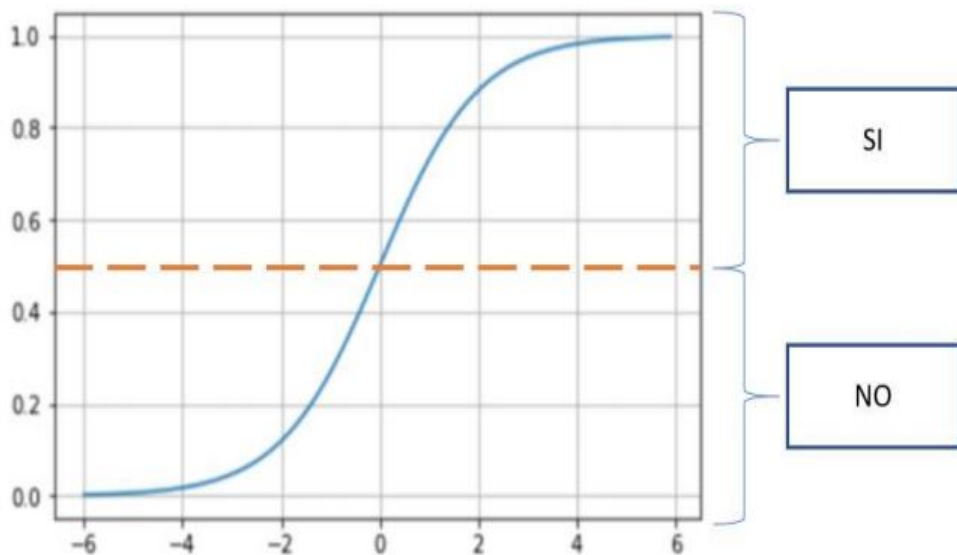


Figura 3.1: Función sigmoide.

3.1.1. Ventajas de la regresión logística

- Modelo de fácil interpretación.
- Costo computacional bajo.

La regresión logística es una técnica muy empleada por los científicos de datos debido a su eficacia y simplicidad, es los algoritmos más sencillos de aplicar ya que los resultados son altamente interpretables. No se necesita contar con grandes recursos computacionales. Siendo esta una de sus principales ventajas respecto a otras técnicas

más complejas. El peso de cada una de las características determina la importancia que tiene en la decisión final.

El funcionamiento de la regresión logística, al igual que la regresión lineal, es mejor cuando se utilizan atributos relacionados con la de salida. También es importante eliminar las características que muestran una gran multicolinealidad entre sí. Por lo que la selección de las características previa al entrenamiento del modelo es clave. Siendo aplicables las técnicas de ingeniería de características también utilizadas en la regresión lineal.

3.1.2. Desventajas de la regresión logística

En cuanto a sus desventajas se encuentra la imposibilidad de resolver directamente problemas no lineales. Esto es así porque la expresión que toma la decisión es lineal. Por ejemplo, en el caso de que la probabilidad de una clase se reduzca inicialmente con una característica y posteriormente suba no puede ser registrado con un modelo logístico directamente. Siendo necesario transformar esta característica previamente para que el modelo pueda registrar este comportamiento no lineal. En estos casos es mejor utilizar otros modelos.

Una cuestión importante es que la variable objetivo esta ha de ser linealmente separable. En caso contrario el modelo de regresión logística no clasificará correctamente. Es decir, en los datos han de existir dos regiones con una frontera lineal.

Otra desventaja es la dependencia que muestra en las características. Ya que no es una herramienta útil para identificar las características más adecuadas. Siendo necesario identificar estas mediante otros métodos

Finalmente, la regresión logística tampoco es uno de los algoritmos más potentes que existen. Pudiendo ser superado fácilmente por otros más complejos.

3.2. Perceptrón Multicapa

El perceptrón multicapa es una red neuronal artificial (RNA) formada por múltiples capas, de tal manera que tiene capacidad para resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón (también llamado perceptrón simple).

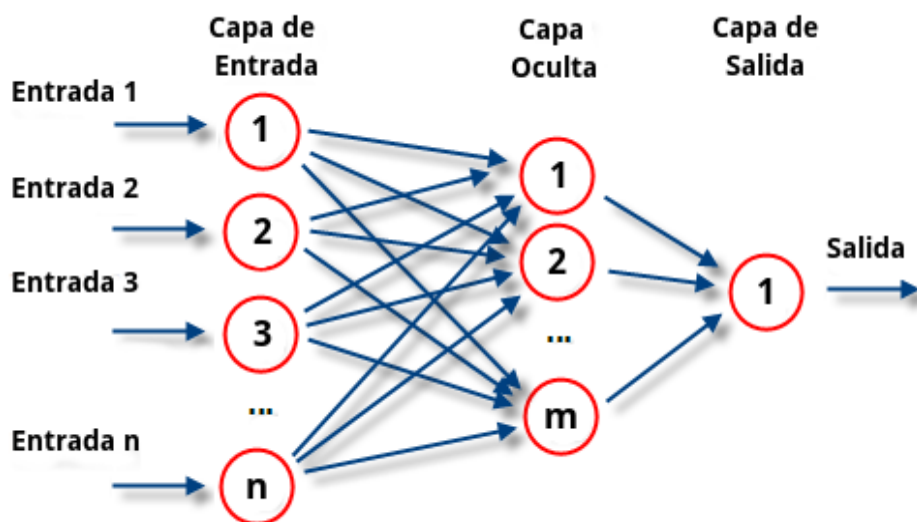


Figura 3.2: Perceptrón Multicapa.

El Perceptrón multicapa es una red de alimentación hacia adelante (feedforward) compuesta por una capa de unidades de entrada (sensores), otra capa de unidades de salida y un número determinado de capas intermedias de unidades de proceso, también llamadas capas ocultas porque no se ven las salidas de dichas neuronas y no tienen conexiones con el exterior. Cada sensor de entrada está conectado con las unidades de la segunda capa, y cada unidad de proceso de la segunda capa está conectada con las unidades de la primera capa y con las unidades de la tercera capa, así sucesivamente. Las unidades de salida están conectadas solamente con las unidades de la última capa oculta. Con esta red se pretende establecer una correspondencia entre un conjunto de entrada y un conjunto de salidas deseadas, de manera que

$$(x_1, x_2, \dots, x_N) \in \mathbb{R}^N \rightarrow (y_1, y_2, \dots, y_M) \in \mathbb{R}^M$$

Para ello se dispone de un conjunto de p patrones de entrenamiento, de manera que se sabe perfectamente que al patrón de entrada $(x_1^k, x_2^k, \dots, x_N^k)$ le corresponde la salida $(y_1^k, y_2^k, \dots, y_M^k)$, $k = 1, 2, \dots, p$. Es decir, conocemos dicha correspondencia para p patrones, así el conjunto de entrenamiento será:

$$\{(x_1^k, x_2^k, \dots, x_N^k) \rightarrow (y_1^k, y_2^k, \dots, y_M^k) : k = 1, 2, \dots, p\}$$

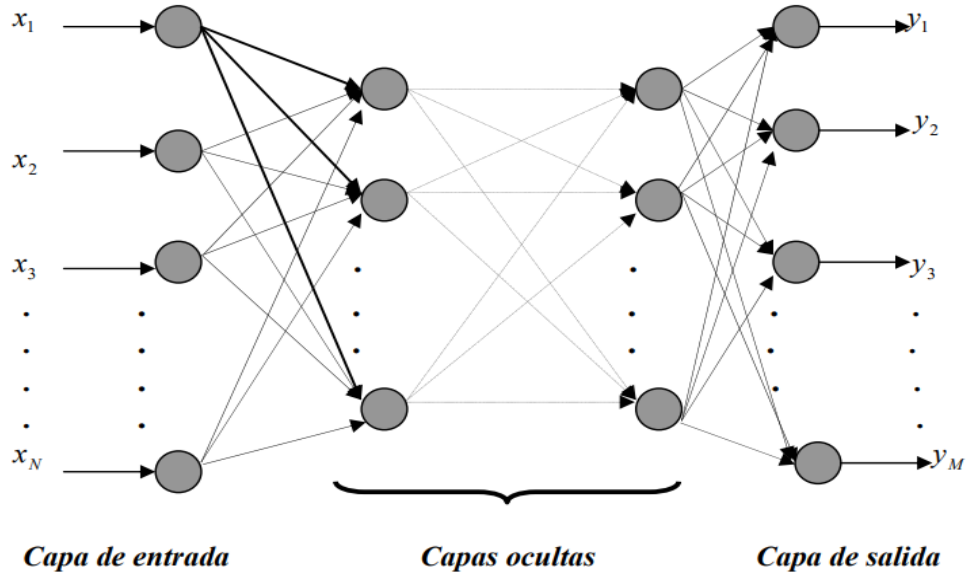


Figura 3.3: Topología Perceptrón Multicapa.

Para implementar dicha relación, la primera capa tendrá tantos sensores como componentes tenga el patrón de entrada, es decir, N ; la capa de salida tendrá tantas unidades de proceso como componentes tengan las salidas deseadas, es decir, M , y el número de capas ocultas y su tamaño dependerán de la dificultad de la correspondencia implementar.

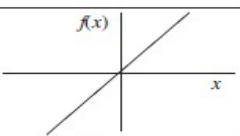
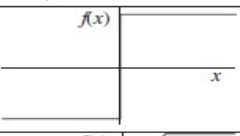
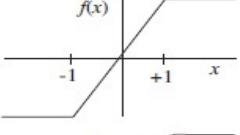
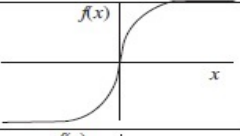
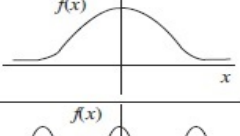
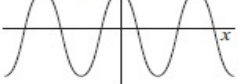
La capa oculta contiene nodos de red no observables (unidades). Cada unidad oculta es una función de la suma ponderada de las entradas. La función de activación y los valores de las ponderaciones se determinan mediante el algoritmo de estimación. Si la red contiene una segunda capa oculta, cada unidad oculta de la segunda capa es una función de la suma ponderada de las unidades de la primera capa oculta. La misma función de activación se utiliza en ambas capas. Un perceptrón de capas múltiples puede tener una o dos capas ocultas. La función de activación o función de transferencia enlaza las sumas ponderadas de las unidades de una capa a los valores de las unidades de la capa sucesiva. La capa de salida contiene las variables (dependientes) de destino.

Como las entradas a las unidades de proceso de una capa son las salidas de las unidades de proceso de la capa anterior, el perceptrón multicapa con sólo una capa oculta implementa la siguiente función:

$$y_i = g_1 \left(\sum_{j=1}^L w_{ij} s_j \right) = g_1 \left(\sum_{j=1}^L w_{ij} \left(g_2 \left(\sum_{r=1}^N t_{jr} x_r \right) \right) \right) \quad (3.10)$$

donde w_{ij} es el peso sináptico de la conexión entre la unidad de salida i y la unidad de proceso j de la capa oculta; L es el número de unidades de proceso de la capa oculta; g_1 es la función de transferencia de las unidades de salida, que puede ser una función logística, una función tangente hiperbólica o la función identidad; t_{jr} es el peso sináptico que conecta la unidad de proceso j de la capa oculta con el sensor de entrada r y g_2 es la función de transferencia de las unidades de la capa oculta, que puede ser también una función logística, una función tangente hiperbólica o la función identidad.

Principales funciones de activación

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(\omega x + \varphi)$	$[-1, +1]$	

Fuente: Hilera (1994).

Figura 3.4: Funciones de Activación o de Transferencia.

Una vez definida la topología de la red, la determinación de los pesos sinápticos lleva al diseño completo de la red. Para ello se sigue un proceso de entrenamiento, mediante el cual se va introduciendo cada una de las entradas y evaluando el error que se comete entre las salidas obtenidas por la red y las salidas deseadas; entonces se irán modificando los pesos sinápticos según el error cometido.

Se trata de determinar los pesos de las conexiones sinápticas entre las unidades de proceso de manera que las salidas de la red coincidan con las salidas deseadas, o por

lo menos, sean lo más proximas posibles. Es decir, se trata de determinar los pesos de manera que el error total sea mínimo:

$$E = \frac{1}{2} \sum_{k=1}^p \sum_{i=1}^M (z_i(k) - y_i(k))^2$$

El algoritmo de retropropagación utiliza el método del descenso por el gradiente y realiza un ajuste de los pesos comenzando por la capa de salida, según el error cometido, y se procede propagando el error a las capas anteriores, de atrás hacia adelante, hasta llegar a la capa de las unidades de entrada. Una característica importante de este algoritmo es su capacidad para organizar el conocimiento de la capa intermedia de manera que se pueda conseguir cualquier correspondencia entre la capa de entrada y la de salida.

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

donde

$$\Delta w_{ij}(k) = -\eta \frac{\partial E}{\partial w_{ij}}(k)$$

Dado un patrón de entrada se aplica como estímulo a la primera capa de neuronas de la red, se va propagando por las siguientes capas hasta que llega a la capa de salida, donde se compara la salida obtenida con la deseada.

Si se regresa al ejemplo de la ecuación 3.10 supongamos que estamos en la iteración k donde se ha introducido el patrón cuya salida de unidad i es $y_i(k)$ y la salida deseada $z_i(k)$, siendolos pesos sinápticos $w_{ij}(k)$ y $t_{jr}(k)$, $i = 1, 2, \dots, M$, $j = 1, 2, \dots, L$, $r = 1, 2, \dots, N$. Entonces la regla de modificación de los pesos sinápticos de la capa de salida será:

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

donde

$$\Delta w_{ij}(k) = -\eta \frac{\partial E}{\partial w_{ij}}(k) = \eta [z_i(k) - y_i(k)] g'_1(h_i) s_j(k)$$

$$h_i = \sum_{j=1}^L w_{ij}(k) s_j(k)$$

Es aquí donde juega un papel importante la función de activación g_1 ya que sí es la función logística entonces

$$g_1'(h_i) = 2\beta g_1(h_i)[1 - g_1(h_i)]$$

si se toma la función tangente hiperbólica entonces

$$g_1'(h_i) = 2\beta g_1(h_i)[1 - g_1(h_i)]$$

y si se toma la función identidad,

$$g_1'(h_i) = h_i$$

lo que simplifica los calculos.

Capítulo 4

Resultados

4.1. Análisis Exploratorio de Datos

El principal propósito del análisis exploratorio de datos es tener una idea de cómo son nuestros datos, antes de decidir qué técnica de Ciencia de Datos o de Machine Learning se usará. Se debe entender su contenido, cuáles son las variables más relevantes y cómo se relacionan unas con otras, comenzar a ver algunos patrones y extraer conclusiones acerca de todo este análisis. Y todo esto es precisamente el análisis exploratorio de datos, que es en resumen una forma de entender, visualizar y extraer información relevante del conjunto de datos para poder decidir cuál será la ruta o técnica más adecuada para su posterior procesamiento, este es siempre el paso cero en cualquier proyecto de Machine Learning o Ciencia de Datos.

En la figura [4.1](#) se han graficado las distribuciones de frecuencia de las variables categóricas, con la finalidad de detectar cuáles son las categorías con el mayor número de observaciones.

En la figura [4.2](#) se han graficado las tablas de contingencia normalizadas desgareadas por la variable objetivo.

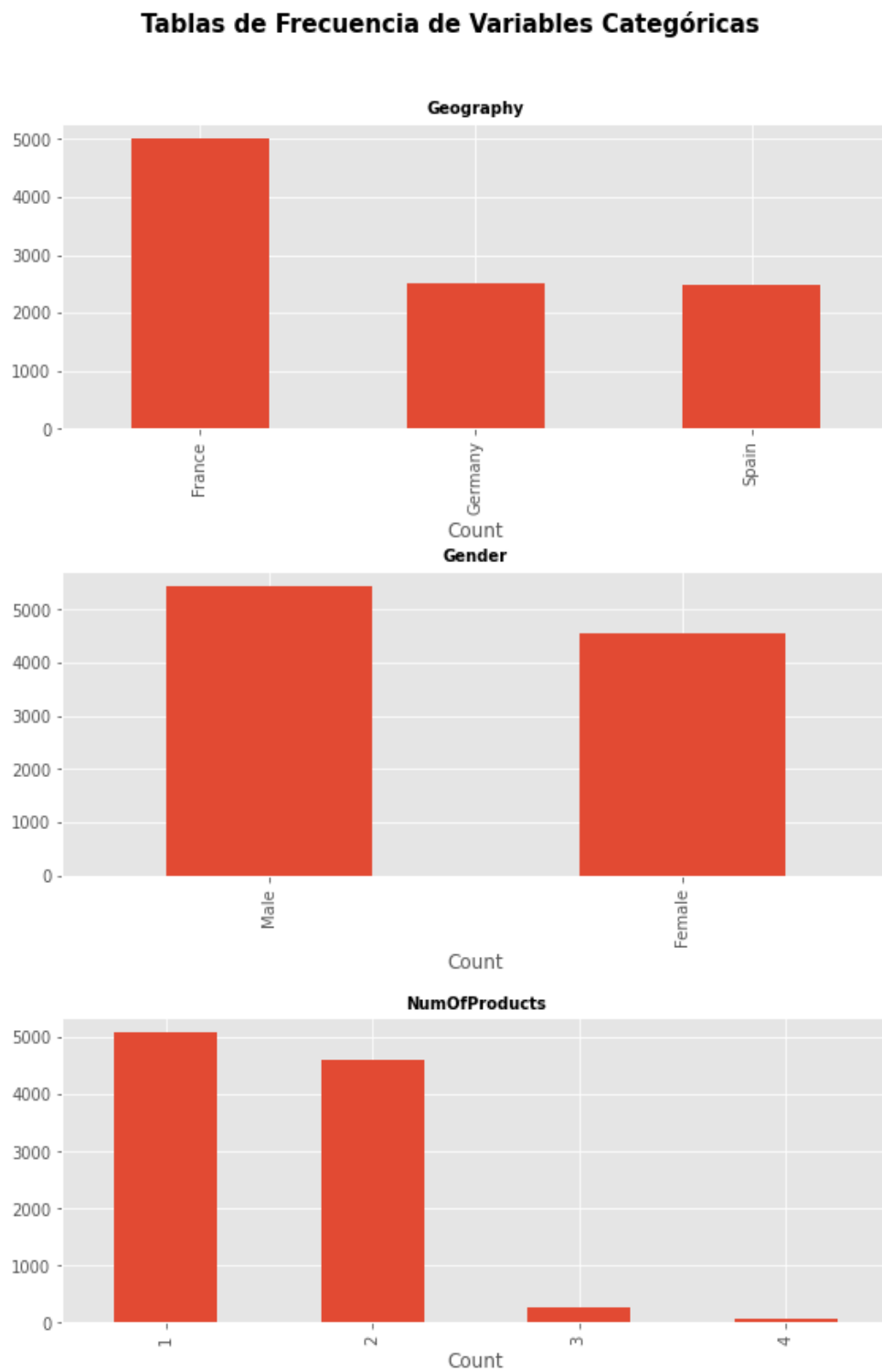


Figura 4.1: Tablas de Frecuencia de Variables Categóricas.

Tablas de Frecuencia de Variables Categóricas Desagregada por Abandono

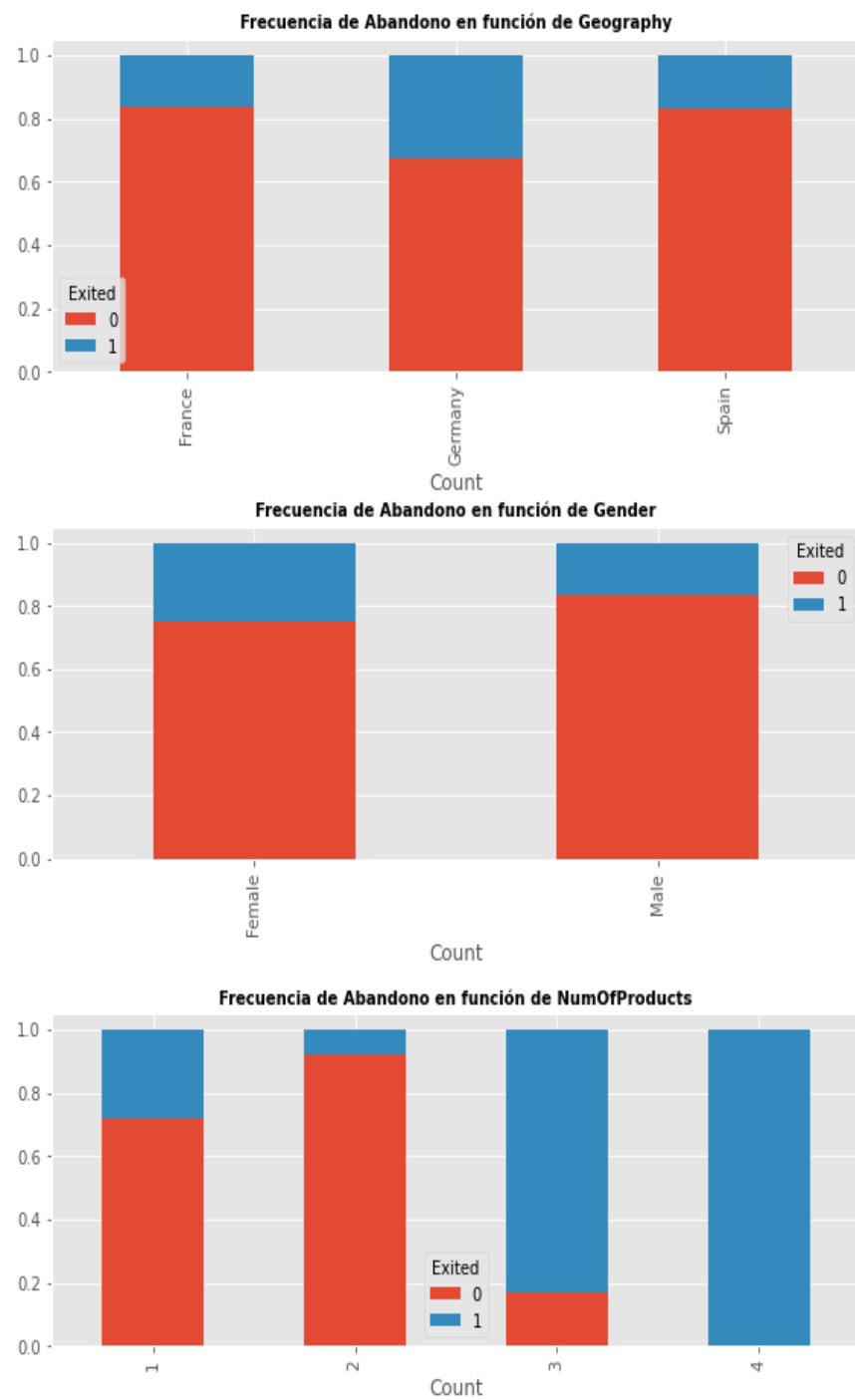


Figura 4.2: Tablas de Frecuencia de Variables Categóricas Desagregada por Abandono.

	count	mean	std	min	25%	50%	75%	max
CreditScore	10000.0	650.528800	96.653299	350.00	584.00	652.000	718.0000	850.00
Age	10000.0	38.921800	10.487806	18.00	32.00	37.000	44.0000	92.00
Tenure	10000.0	5.012800	2.892174	0.00	3.00	5.000	7.0000	10.00
Balance	10000.0	76485.889288	62397.405202	0.00	0.00	97198.540	127644.2400	250898.09
NumOfProducts	10000.0	1.530200	0.581654	1.00	1.00	1.000	2.0000	4.00
HasCrCard	10000.0	0.705500	0.455840	0.00	0.00	1.000	1.0000	1.00
IsActiveMember	10000.0	0.515100	0.499797	0.00	0.00	1.000	1.0000	1.00
EstimatedSalary	10000.0	100090.239881	57510.492818	11.58	51002.11	100193.915	149388.2475	199992.48
Exited	10000.0	0.203700	0.402769	0.00	0.00	0.000	0.0000	1.00

Figura 4.3: Estadísticos Básicos.

En la figura 4.3 se tienen los estadísticos básicos de las variables numéricas como media, desviación estándar, mínimo, máximo y primer, segundo y tercer cuartil. Donde se puede observar que la edad promedio es de 39 años y un ingreso de 100,090.23, donde también se han graficado sus distribuciones en las figuras 4.4 y 4.5.

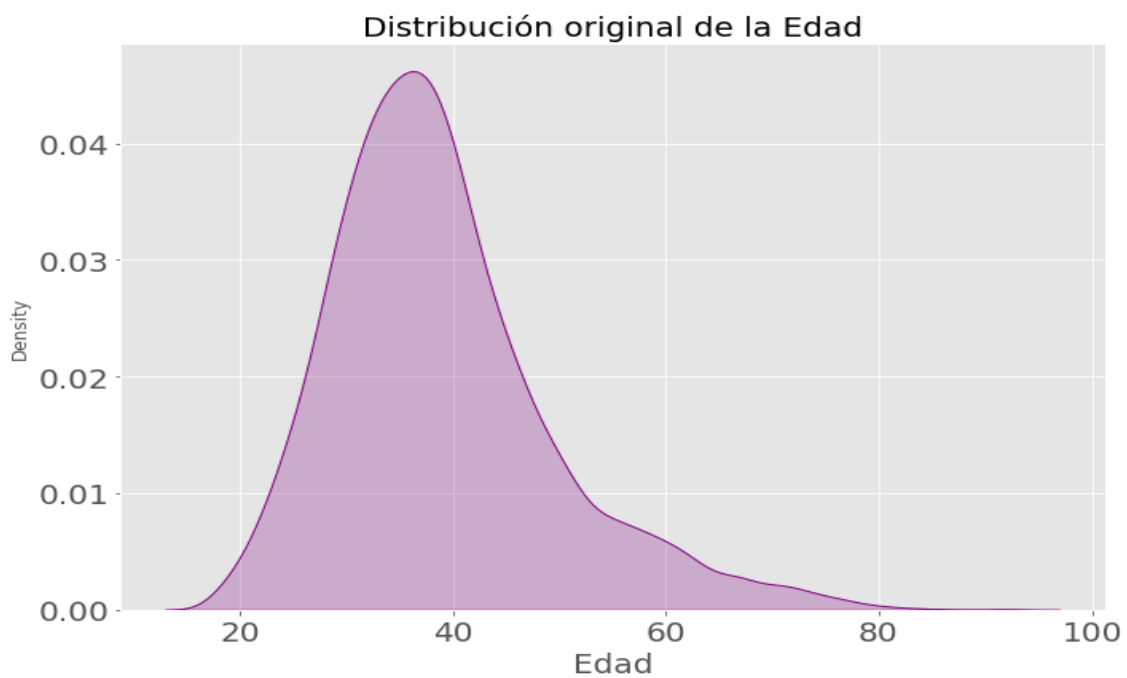


Figura 4.4: Distribución de la Edad.

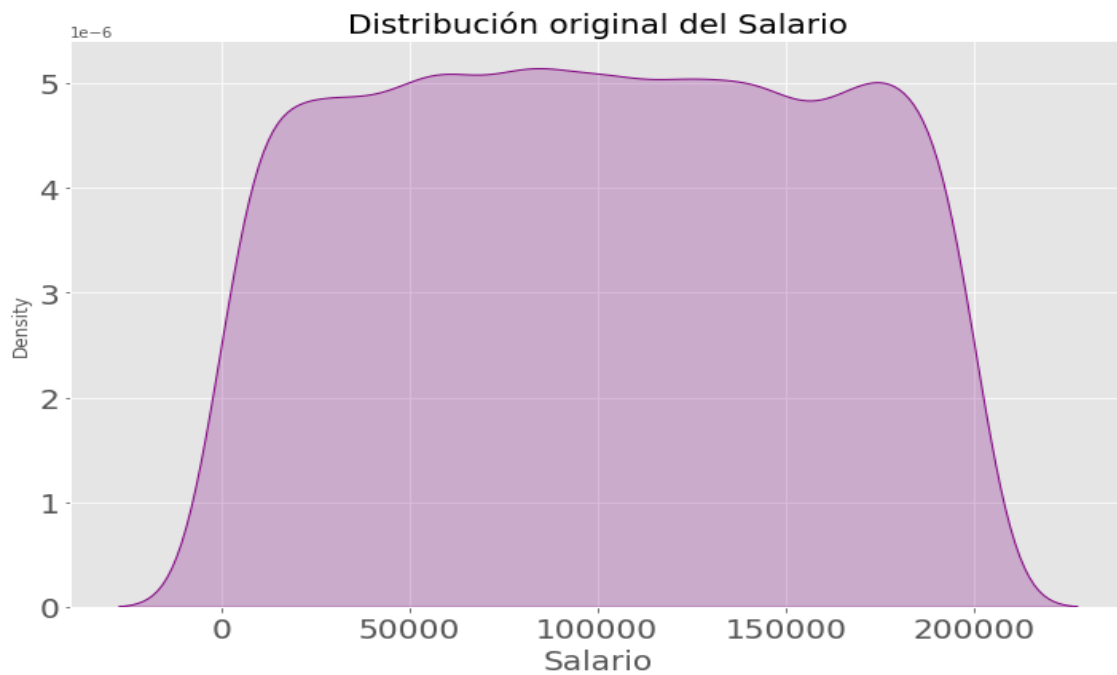


Figura 4.5: Distribución del Ingreso.

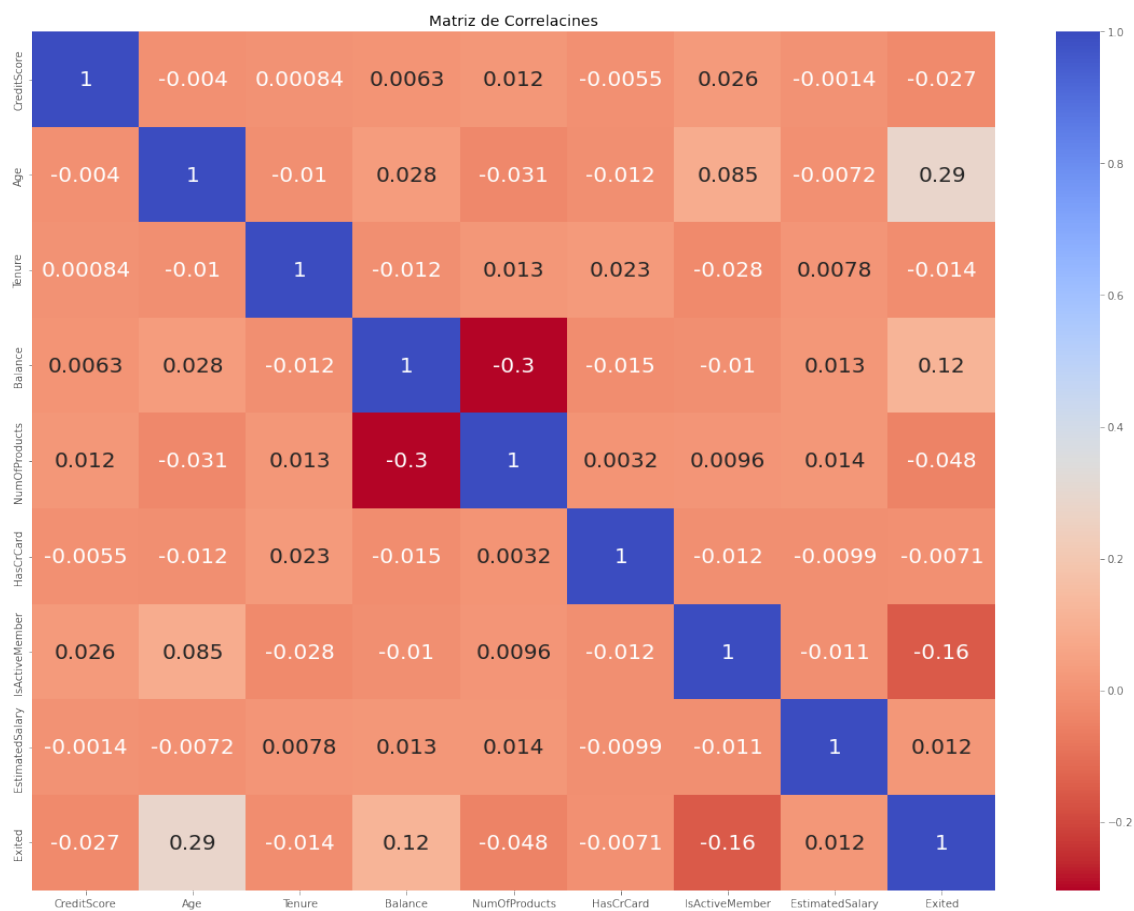


Figura 4.6: Matriz de Correlaciones.

En la figura 4.6 se muestra la matriz de correlaciones de las variables numéricas en la que se puede observar que no se presentan problemas de multicolinealidad.

En la tabla 4.1 se tiene la frecuencia absoluta y relativa de la variable objetivo, donde se puede observar que no se presenta un problema de desbalanceo de clases.

Cuadro 4.1: Distribuciones.

Valor	Total	Porcentaje
No Abandona	7963	79.63 %
Aprobada	2037	20.37 %

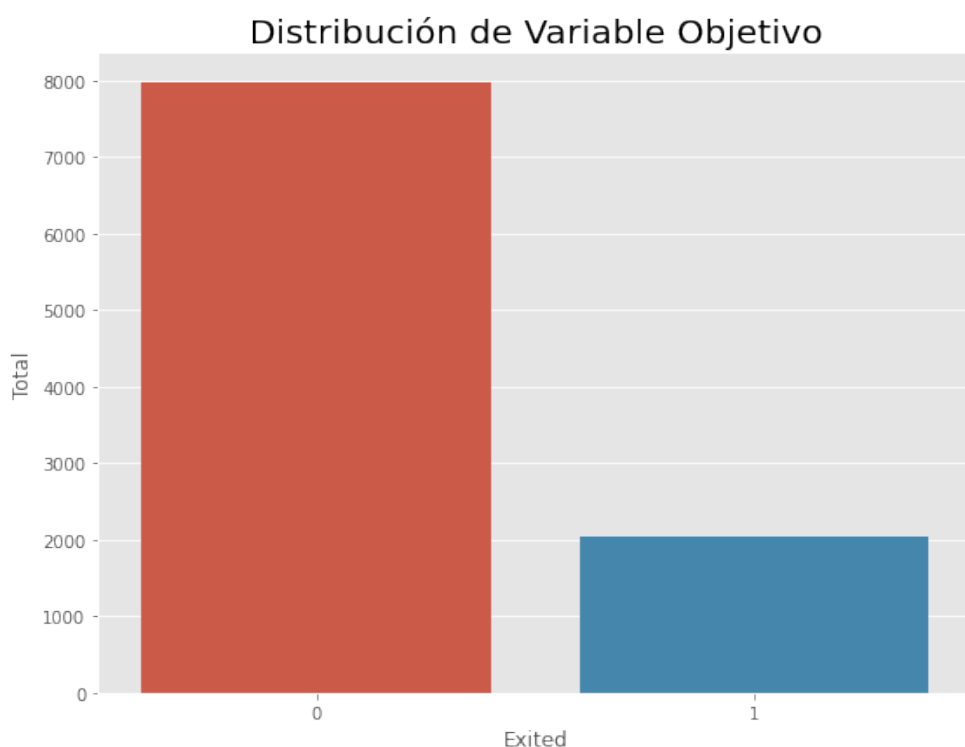


Figura 4.7: Distribución de la Variable Objetivo.

4.1.1. Construcción del Modelo

Una vez limpia la base y realizado un análisis exploratorio, se procede al tratamiento de las variables categóricas, es decir, codificar características categóricas como una matriz numérica para poder aplicar el algoritmo ya que solamente recibe valores numéricos, esto se puede hacer una una función de la librería de sklearn llamada *OneHotEncoder*.

Una vez hecho esto ahora sí se procede a separar los datos en datos de entrenamiento con los cuales se va a entrenar el modelo y datos de test con los que se va a evaluar el

modelo. Una de las convenciones es separar los datos de la siguiente manera:

- Entrenamiento: 70 % Un subconjunto para entrenar el modelo, este subconjunto se utilizará en la práctica para ajustar los parámetros en los algoritmos.
- Test: 30 % Un subconjunto para probar el modelo entrenado, este subconjunto de datos se utilizará en la práctica para proporcionar una evaluación imparcial de un último ajuste del modelo y análisis de los resultados.

Como modelo se propone una perceptrón multicapa con tres capas densas ocultas y funciones de activación relu y sigmoide. Se utilizará la librería Keras debido a su sencillez y practicidad.

Todo esto se realizó en una tubería con el uso de Pipeline de sklearn la cual se puede ver en la figura 4.8.

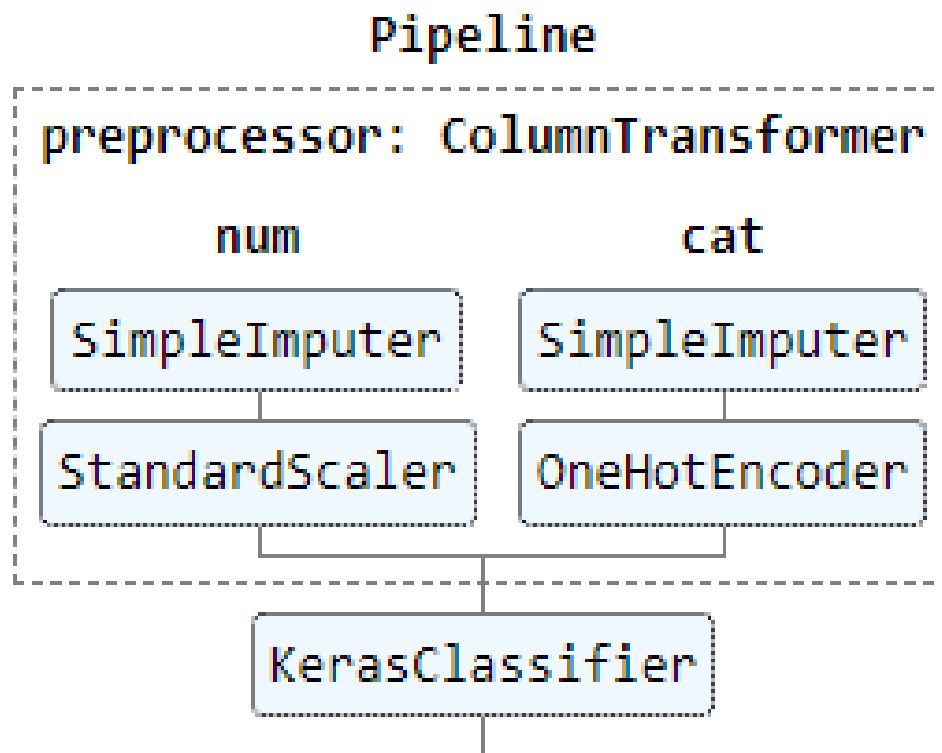


Figura 4.8: Pipeline del modelo.

Al entrenar el modelo la primera métrica que se considera es el Accuracy, que es el total de predicciones correctas entre el total de predicciones.

A la hora del uso de los modelos es de gran importancia el uso de los conjuntos de datos que se están usando. Es aquí donde entran en práctica los términos overfitting o underfitting conocidos en castellano como sobreajuste y subajuste.

- Subajuste: se dice de un modelo estadístico o un algoritmo de aprendizaje automático en el que se obtiene un bajo rendimiento cuando al no proporcionar los suficientes datos no puede captar la tendencia subyacente de los datos. El modelo probablemente realizará un gran número de predicciones erróneas.

Causas:

- Modelo demasiado simple, no es capaz de aprender relación entre datos de entrada y salida
 - Falta de datos, si no se cuenta con los datos necesarios.
 - Atributos relevantes insuficientes, es posible tener todos los datos pero que surja la necesidad de transformar estos para una mejor comprensión del modelo entre la relación de entrada salida
- Sobreajuste: se dice de un modelo estadístico o un algoritmo de aprendizaje automático en el que se obtiene un bajo rendimiento cuando se entrena con un número demasiado alto de datos. Cuando un modelo se entrena con datos de más, comienza a aprender del noise o ruido y de las entradas de datos inexactos y no es capaz de categorizar los datos correctamente debido a demasiados detalles.

Causas:

- Modelo demasiado complejo, podrá aprender muchos de los datos de memoria.
- Los datos tienen noise, es decir, como si hubiera valores atípicos y errores en los datos.
- El tamaño de los datos utilizados para los datos de entrenamiento puede que no sean suficientes.

4.1.2. Evaluación del Modelo

Al utilizar la red neuronal los resultados al evaluar el modelo con los siguientes:

La matriz de confusión es una herramienta que nos muestra el desempeño de un algoritmo de clasificación, describiendo cómo se distribuyen los valores reales y las predicciones hechas por el modelo mediante 4 distintos casos basados en 4 variables que se comentan a continuación:

- Verdadero positivo (VP): Se predice que es positivo y es verdad, fue clasificado correctamente.
- Verdadero negativo (VN): Se predice que es negativo y es verdad, fue clasificado correctamente.
- Falso positivo (FP): Se predice que es positivo y es falso, fue clasificado incorrectamente.
- Falso negativo (FN): Se predice que es negativo y es falso, fue clasificado incorrectamente.

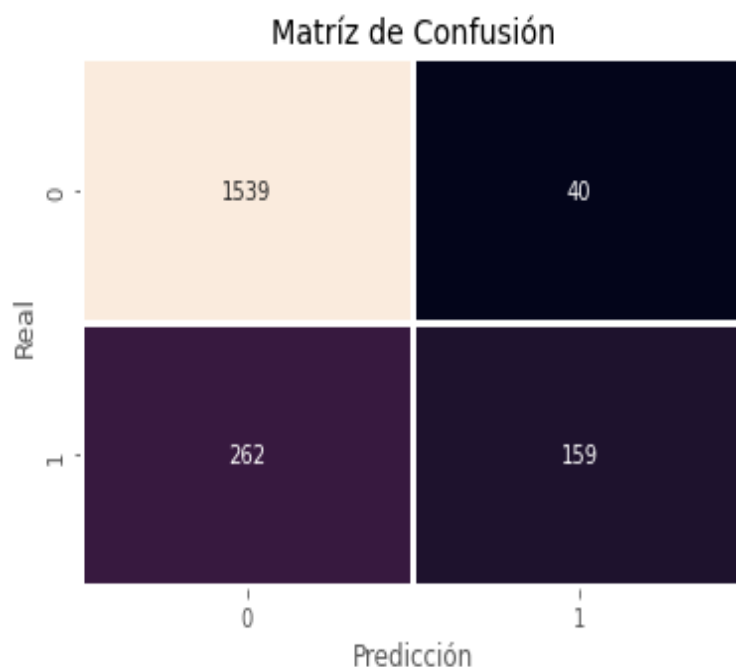


Figura 4.9: Matriz de Confusión.

A partir de la matriz de confusión se pueden obtener la precision, recall y f1-score que se pueden observar en la tabla 4.2. Una buena métrica que pondera la precision y el recall es la f1-score y para ambas clases es bastante alta por lo cual el modelo tiene un ajuste considerablemente bueno, es decir, es capaz de distinguir las dos clases.

Cuadro 4.2: Métricas.

Clase	Precision	Recall	f1-score
0 (No Abandona)	0.85	0.97	0.91
1 (Abandona)	0.80	0.38	0.51

Al aplicar una técnica de validación cruzada se obtiene un Accuracy promedio de 83.79 % con una desviación $\pm 0.79\%$

4.1.3. Curva ROC

Una curva ROC (curva de característica operativa del receptor) es una gráfica que muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación. Esta curva representa dos parámetros:

La curva ROC-AUC es una medida de rendimiento para el problema de clasificación en varios umbrales. ROC es una curva de probabilidad y AUC representa el grado o medida de separabilidad. Será a través de la métrica AUC (Area Under the ROC Curve) que representa un valor del área que queda por debajo de la curva ROC, la encargada de comparar unos modelos con otros indicando cuánto es capaz el modelo de distinguir entre clases. Cuanto más alto es el AUC, mejor es el modelo para predecir verdaderos como verdades y falsos como falsos. La curva ROC se traza con la Sensibilidad frente la Especificidad donde la Sensibilidad está en el eje y , y el eje x está compuesto por 1-Especificidad, la evaluación de esta medida se clasifica de la siguiente forma:

- Valor cerca de 1: El modelo es excelente, tiene una gran capacidad de separabilidad. En este caso es perfectamente capaz de distinguir entre la clase positiva y la clase negativa.
- Valor cerca de 0.5: El modelo no tiene ninguna capacidad de separación de clases. Es la peor situación y el modelo no tiene capacidad de discriminación para distinguir entre clase positiva y negativa.

La gráfica de la curva ROC del modelo se puede observar en la figura [4.10](#) donde se obtiene un AUC de 0.86 bastante aceptable.

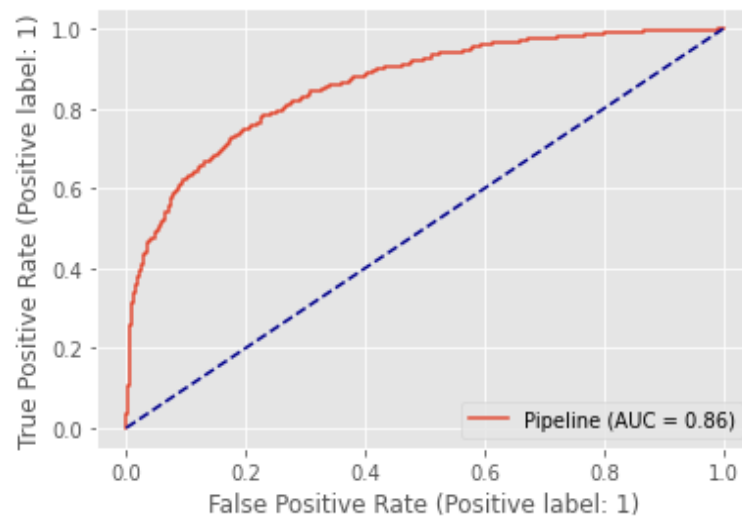


Figura 4.10: Curva ROC.

Capítulo 5

Conclusiones

El machine learning o aprendizaje automático abre para las organizaciones posibilidades sin precedentes que facilitan la automatización, la eficiencia y la innovación. Tal es el caso en el sector financiero en donde la complejidad en la gestión de los riesgos financieros ha aumentado durante los últimos años, lo cual ha generado, en algunos casos, grandes pérdidas económicas por parte de las instituciones financieras derivadas de fallos en los modelos de medición de riesgos y en los esquemas de toma de decisiones soportados por estos. En especial, el abandono de clientes supone una de las principales preocupaciones para las entidades. Para reducir este tipo de problemas, muchas entidades utilizan sistemas automáticos de clasificación de clientes. Como resultado de la evolución de la tecnología, los modelos de machine learning están cada vez más presentes en el sector financiero.

En este trabajo se presentó el procedimiento necesario para implementar un algoritmo de machine learning capaz de predecir de manera proactiva el abandono de clientes con el fin abordar una problemática que significa grandes pérdidas tanto en capital como en esfuerzo para el sector bancario, los modelos predictivos maximizan la efectividad de las campañas de retención y a su vez permiten un enfoque más centrado en el cliente y sus necesidades.

A la hora de afrontar un problema de Machine Learning es necesario tener claro que este no ofrece soluciones perfectas, no siempre es posible implementar un algoritmo de Machine Learning y que el tiempo y costo de implementación puede ser muy alto si no se tiene claro el alcance del proyecto. Para que la implementación sea exitosa es preferible definir sprints ya que estos ayudan a tener claras las etapas del desarrollo, en este sentido el modelo CRISP-DM es altamente recomendable de utilizar ya que segmenta de manera concisa los pasos necesarios para un proyecto de este tipo

En este proyecto se consideró una dataset público conformado por clientes que decidieron abandonar un banco. Se planteó un problema de aprendizaje supervisado en donde la variable objetivo se representa con 1 si el cliente abandona 0 en caso contrario. Una vez limpia la base y al aplicar un análisis exploratorio de datos se aplicó un perceptrón multicapa. Con respecto al modelo el 83.79% de los casos fueron correctamente clasificados utilizando un umbral de 0.5. También se calculó la curva ROC y se obtuvo un AUC de 0.86 que es bastante cercano a 1 por lo que se concluye que el modelo es capaz de hacer una buena clasificación.

Cabe mencionar que este algoritmo, al igual que cualquier otro, depende en gran parte de los datos, por lo que sin una buena base de información de gran valor los sistemas no serán capaces de obtener un modelo confiable, de manera que sus limitaciones están condicionadas por otros factores.

Una de las líneas a seguir es tratar de agregar más capas al modelo para tratar de mejorar las métricas y obtener un mejor modelo para su posterior puesta en producción que es el último paso en la metodología CRISP-DM.

El código se encuentra en un repositorio de GitHub en la siguiente url:

- https://github.com/jrtorresb/Proyecto_Final_FC_Proyecto_II

A tomar en cuenta

- En general, la deserción se expresa como un grado de inactividad o desvinculación del cliente, observado durante un tiempo determinado.
- Otro aspecto del problema empresarial es ¿qué tan temprano desea que prediga el modelo? Una predicción que está demasiado lejos puede ser menos precisa. Por otro lado, un horizonte de predicción corto puede tener una mejor precisión, pero podría ser demasiado tarde para intervenir una vez que el cliente ya haya tomado una decisión.
- Es crucial determinar si la deserción debe definirse a nivel de producto (es probable que el cliente se desvincule de un producto en particular, como discontinuar una tarjeta de crédito) o al nivel de la relación (es probable que el cliente se desvincule del propio banco).

Bibliografía

- [1] Wright, R. E. (1995). Logistic regression.
- [2] Menard, S. (2002). Applied logistic regression analysis (Vol. 106). Sage.
- [3] Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3), 4626-4636.
- [4] Qian, Z., Jiang, W., & Tsui, K. L. (2006). Churn detection via customer profile modelling. *International Journal of Production Research*, 44(14), 2913-2933.
- [5] Shaaban, E., Helmy, Y., Khedr, A., & Nasr, M. (2012). A proposed churn prediction model. *International Journal of Engineering Research and Applications*, 2(4), 693-697.
- [6] Tsai, C. F., & Lu, Y. H. (2009). Customer churn prediction by hybrid neural networks. *Expert Systems with Applications*, 36(10), 12547-12553.