Análisis y propuesta de solución para un problema derivado del telemarketing bancario

Gaia Ramirez Hincapié[1], Silvio José Otero Guzman[1], Sara Galván Ortega[1]
[1]Programa de ingeniería de sistemas. Facultad de Ingeniería. Universidad de Antioquia. Medellín

En el presente trabajo se explica de manera suscinta el problema de clasificación asociado al data set *"Bank Marketing Dataset"*, esto es la construcción de un modelo predictivo para asociar a cada cliente contactado la probabilidad de adquisición de un depósito a término fijo; también se explica la composición del data set (qué datos lo componen y su naturaleza) así como un posible modelo o paradigma aplicable a la solución de este problema. Finalmente, se realiza un breve estado del arte.

***Index Terms*—Depósitos a término fijo, SVMs, Decicion Tree, Multilayer perceptron**

I. Introduction

A DÍA de hoy las campañas de marketing representan una parte vital del fortalecimiento de cualquier negocio, de forma general las campañas de telemarketing se basan en la interacción con los usuarios, dicha interacción, generalmente se produce con un objetivo específico, sin embargo; muchas veces dicho objetivo no se cumple, debido a diversos motivos. Una manera de aportar a la solución de este problema es el desarrollo de modelos basados en datos que permitan conocer la probabilidad de éxito *previa* a la interacción del usuario, con la campaña de marketing.

II. Descripción del problema

Dentro del contexto de la base de datos "Bank Marketing Dataset" [2] que se va a manejar se observan los registros de "ventas" teléfonias a clientes de su banco para ofrecerles cDTs o depósitos a plazo (mantener cierta cantidad de dinero en una cuenta que genera intereses pero no puede ser usada por un tiempo). Cada llamada que se realiza con este fin implica diferentes gastos como: tiempo, y recursos humanos, pero no todas representan un caso exitoso. Desde el área del Machine Learning se podría predecir qué clientes tienen más probabilidad de aceptar este tipo de servicio. Esto permitiría: optimizar recursos al solo llamar a clientes potenciales, y mejorar la experiencia del cliente ya que no sería una oferta irrelevante para el usuario potencial. De este modo cada servicio podría estar mejor orientado a un público específico, lo que a su vez se reflejaría en una posible mejora en la reputacion del banco y una mejor asignación de recursos mas eficiente.

Corresponding author: S. Galván (email: sara.galvan@udea.edu.co).

A. Composición de la base de datos

La base de datos que será abordada en este proyecto presenta un conjunto de datos que contiene 41,188 muestras, con un total de 20 variables, 19 de ellas correspondientes a variables de entrada (predictoras) y una sola variable de salida (objetivo). Las variables presentes pueden establecerse en categorías para un mejor entendimiento de ellas.

Datos relativos a los clientes del banco:

- *Age* (numérica): Edad del cliente.
- *job* (categórica): Tipo de trabajo.
- *marital* (categórica): Estado civil ('divorced' incluye viudos).
- *education* (categórica): Nivel educativo.
- *default* (categórica): ¿Tiene créditos en incumplimiento?
- *housing* (categórica): ¿Tiene préstamo de vivienda?
- *loan* (categórica): ¿Tiene préstamo personal?

Datos relacionados al último contacto de la campaña actual:

- *contact* (categórica): Tipo de comunicación utilizada.
- *month* (categórica): Mes del último contacto.
- *day_of_week* (categórica): Día de la semana del último contacto.

Otros atributos de campaña:

- *campaign* (numérica): Número de contactos durante esta campaña (incluye el último contacto).
- *pdays* (numérica): Días desde el último contacto previo (999 indica que no fue contactado previamente).
- *previous* (numérica): Número de contactos anteriores.

- *poutcome* (categórica): Resultado de la campaña de marketing anterior.

Contexto socio-económico:

- *emp.var.rate* (numérica): Tasa de variación del empleo.
- *cons.price.idx* (numérica): Índice de precios al consumidor.
- *cons.conf.idx* (numérica): Índice de confianza del consumidor.
- *euribor3m* (numérica): Tasa Euribor a 3 meses.
- *nr.employed* (numérica): Número de empleados.

Variable de salida:

- *y* (binaria): ¿El cliente ha suscrito un depósito a plazo? ('yes' o 'no').

Los datos faltantes no están presentes de manera explícita (nulos o vacíos) en el conjunto de datos, sin embargo, algunas variables categóricas incluyen el valor "unknown", que cumple la función de representar datos faltantes implícitos. Esto ocurre en las variables job, marital, education, default, hosting y loan.

De acuerdo a la información presentada con respecto a esta base de datos no se aplicó una estrategia explícita de imputación, los valores faltantes fueron codificados como "unknown" y ya es responsabilidad del propio analista decidir si imputarlos, eliminarlos o tratarlos como una categoría válida durante el modelado.

Al referirse al tipo de codificación usado para las variables, en el caso de las numéricas, estas permanecen en su forma original (sin codificación adicional necesaria). Por otro lado, las variables categóricas están representadas como etiquetas de texto (string) pero una medida que es comúnmente tomada con este tipo de datos es codificarlas con One-hot encoding (para variables sin orden inherente) o label encoding (útil al momento de ahorrar espacio). Por último, la variable objetivo fue codificada como texto, siendo este "yes" o "no", dando pie al uso de clasificación binara al transformar esas respuestas en ceros y unos.

B. Primera aproximación a la solución del problema

En términos generales podemos decir que el objetivo sobre este data set es construir un modelo basado en datos que aprenda una función subyacente desconocida que mapea varias variables de entrada que caracterizan un elemento (p. ej., un cliente bancario) con un objetivo de salida etiquetado (p. ej., tipo de venta de depósito bancario: "fracaso" o "éxito").

Para este tipo de problemas de clasificación es necesario analizar cuáles son los paradigmas aplicables. De forma inicial, se puede señalar que este es un problema de aprendizaje *supervisado* dado que en el dataset por cada variable independiente, tenemos un valor de salida, (coloquialmente se puede decir que por cada x tenemos una y). Además se determinó que este es un problema *desbalanceado*, dado que la cantidad de valores correspondientes al *'yes'* es mucho menor a los que corresponden al *'no'*, en términos porcentuales los *'yes'* representan 11.26% de los datos mientras que los *'no'* 88.73% y como es conocido, este tipo de datos pueden producir modelos con sesgos hacia los datos mayoritarios.

TABLE I
MÉTRICAS DE DESEMPEÑO PARA TODOS LOS MODELOS.

Clasificador	Clase	Presición	Recall	Ov. Accuracy
Logistic Regression	0	93.31%	97.39%	91.48%
	1	68.61%	44.99%	
Decision Tree	0	93.79%	96.97%	89.91%
	1	58.82%	57.92%	
Multilayer Perceptron	0	91.90%	97.43%	90.10%
	1	61.55%	32.39%	

III. ESTADO DEL ARTE

A. Predicting the Accuracy for Telemarketing Process in Banks Using Data Mining

El articulo Predicting the Accuracy for Telemarketing Process in Banks Using Data Mining, aborda varios modelos desarrollados sobre la base de datos de telemarketing en un banco. Se emplea un paradigma de aprendizaje supervisado, ya que los modelos se entrenan con una base de datos que tienen la etiqueta donde la variable objetivo es la aceptación o rechazo de los clientes a un servicio particular ofrecido. Inicialmente se uso un modelo de Regresión Logística, luego una red neuronal y por ultimo arbol de decisión para predecir la posible aceptación de la oferta. Se utiliza la estrategia de validación cruzada de 10 folds para evaluar el rendimiento de los modelos. Se divide el conjunto de datos en 10 subconjuntos, entrenando el modelo en 9 y validándolo en 1, repitiendo el proceso 10 veces [1]. Se emplearon las métricas de Precision, Recall y Accuracy, todas vistas anteriormente en clase. Los resultados obtenidos fueron:

B. A Data-Driven Approach to Predict the Success of Bank Telemarketing

En este artículo se emplea un paradigma de aprendizaje supervisado, ya que el objetivo que se plantea es predecir si un cliente suscribirá o no un depósito a plazo fijo a partir de un conjunto de atributos conocidos. Para resolver este problema de clasificación binaria, se

utilizaron diversas técnicas de aprendizaje automático, específicamente: regresión logística, árboles de decisión, redes neuronales artificiales y máquinas de vectores de soporte. Estas técnicas fueron comparadas para determinar cuál ofrecía el mejor desempeño predictivo sobre el conjunto de datos. Con respecto a la metodología de validación, se aplicó una validación cruzada estratificada de 10 particiones sobre el conjunto de datos, lo que permite evaluar el rendimiento de los modelos de manera robusta y evita sobreajuste[2]. Las métricas de evaluación utilizadas incluyen el Área Bajo la Curva ROC (AUC) y la curva de Lift acumulativa (ALIFT). Estas métricas permiten no solo medir la capacidad predictiva general del modelo, sino también su eficacia en identificar correctamente a los clientes más propensos a suscribirse. el lift acumulado (ALIFT) evalúa cuánto mejora el modelo la identificación de casos positivos respecto a una selección aleatoria. Para un cierto percentil p, se define como:

$$Lift(p) = \frac{TP_p/N_p}{P/N} \tag{1}$$

Donde TP_p es el número de verdaderos positivos acumulados hasta el percentil p, N_p el número total de casos en ese percentil, P el total de positivos y N el número total de casos. De acuerdo a los resultados obtenidos, el modelo basado en redes neuronales fue el que obtuvo el mejor rendimiento, alcanzando un AUC de 0.8 y un ALIFT de 0.7. Esto significa que el modelo fue capaz de identificar correctamente el 79% de los suscriptores reales seleccionando solo al 50% de los clientes mejor clasificados.

IV. Configuración experimental

A. Metodología de validación escogida

Todas las variables categóricas fueron transformadas mediante codificación one-hot, dado que no presentan un orden jerárquico que justifique el uso de label encoding. Esta codificación genera una columna binaria por cada categoría, lo cual facilita su procesamiento por modelos lineales y redes neuronales, que no pueden operar directamente con texto. Las variables numéricas para tener en cuenta se estandarizaron utilizando la técnica de Z-score, es decir, se transformaron para tener media cero y desviación estándar uno. Esto permite mejorar el rendimiento de modelos sensibles a la escala, como Support Vector Machines y redes neuronales artificiales.

Adicionalmente, se llevó a cabo un análisis exploratorio de la distribución de clases, observándose un notable desbalance donde solo cerca del 11% de las muestras pertenecen a la clase positiva, o a clientes que adquirieron el servicio ofertado. Debido a esto se planteó la inclusión de técnicas de sobremuestreo para corregir el desbalance en las etapas de entrenamiento.

Para realizar un entrenamiento adecuado y minimizar riesgos de sobreajuste, se uso una estrategia de validación con 10 particiones (10-fold). Este método divide el conjunto de datos en 10 subconjuntos con la misma proporción de clases que el total. En cada iteración, se entrenan los modelos con 9 de esos subconjuntos y se validan con el restante. Este proceso se repite 10 veces, de forma que cada instancia es usada una vez para validación y nueve veces para entrenamiento.

Dado que el conjunto de datos está desbalanceado, se incorporó la técnica de sobremuestreo. Esto se aplicó exclusivamente al conjunto de entrenamiento de cada fold, para evitar modificación de los datos en la validación.

En la tabla II se presenta un resumen de los clasificadores a implementar y sus hiperparámetros y en la tabla IV se presenta un análisis comparativo de los valores de los hiperparámetros correspondientes a cada modelo.

V. Métricas de desempeño para la evaluación de los modelos

Las métricas fueron seleccionadas para la evaluación completa desde múltiples perspectivas, permitiendo analizar no solo cuántas predicciones son correctas, sino también qué tipo de errores comete el modelo, lo que es muy importante a la hora de toma de decisiones en un contexto como el del problema de telemarketing bancario. La idea es poder observar la capacidad de cada modelo para capturar el verdadero valor predictivo. En las secciones subsiguientes se explican cuáles métricas se han seleccionado para lograr estos objetivos.

A. F1-score

Esta métrica es la media armónica entre la precisión y la sensibilidad. Es útil en problemas desbalanceados, ya que penaliza tanto los falsos positivos (clientes contactados que realmente no estaban interesados) como los falsos negativos (clientes que sí habrían adquirir el producto pero fueron clasificados como no interesados)

B. Área bajo la curva ROC (AUC-ROC)

Esta métrica evalúa la capacidad del modelo para discriminar entre clases, mostrando cómo varían los verdaderos positivos frente a los falsos positivos a medida que se modifica el umbral de decisión.

TABLE II
BREVE DESCRIPCIÓN DE LOS HIPERPARÁMETROS CONCERNIENTES A CADA MODELO.

Clasificador	Hiperparámetro	Descripción	Valores evaluados
Regresión Logística	C	{0.01, 0.1, 1, 10 }	{0.01, 0.1, 1, 10 }
	penalty	Tipo de penalización aplicada.	$\{'l2'\}$
K-Nearest Neighbors	n_neighbors	Número de vecinos más cercanos usados para clasificar.	{ 3, 5, 7, 11 }
	metric	Distancia utilizada para calcular similitud entre puntos.	{ 'euclidean' }
Random Forest	n_estimators	Número de árboles que componen el ensamble.	{ 100, 200 }
	max_depth	Profundidad máxima de los árboles.	{ None, 10, 20 }
	max_features	Número de variables consideradas en cada división de nodo.	{ 'sqrt', 'log2' }
Red Neuronal MLP	hidden_layer_sizes	Estructura de las capas ocultas (número de neuronas por capa).	{ (50,), (100,), (100, 50) }
	activation	Función de activación usada en las capas ocultas.	{ 'relu', 'tanh' }
	learning_rate_init	Tasa de aprendizaje inicial.	{ 0.001, 0.01 }
SVM	C	Penalización por error. Controla el margen de separación.	{ 0.1, 1, 10 }
	kernel	Tipo de núcleo que transforma los datos.	{ 'linear', 'rbf' }
	gamma	Parámetro que define la influencia de cada ejemplo de entrenamiento (en RBF).	{ 'scale', 'auto' }

C. Precisión

Representa el porcentaje de predicciones positivas que realmente pertenecen a la clase positiva. Para evitar desperdiciar recursos, ya que una baja precisión llevaría a contactar a muchos clientes sin interés real.

D. Sensibilidad (Recall)

Mide la proporción de verdaderos positivos detectados entre todos los positivos reales. En el problema abordado, representa qué tan bien el modelo identifica correctamente a los clientes que sí contratarian el producto.

E. Matriz de confusión

Permitirá observar con detalle los errores falsos positivos y falsos negativos, lo cual es fundamental para interpretar el comportamiento del modelo dentro del contexto del problema.

VI. RESULTADOS

Con relación a los resultados obtenidos se puede señalar que de acuerdo a la matriz de confusión fig 1 se obtuvo un gran número de *falsos negativos*, esto quiere decir que la predicción obtenida del modelo fué buena en comparación con los datos reales, lo mismo se concluye con relación a los resultados positivos.

Por otro lado en la fig. 2 se presenta la curva ROC la cual determina qué tan bien clasifica el modelo tanto valores positvos (*'yes'*) como los negativos (*'no'*) obteniéndose un valor para el AUC de 0.8082 lo cual se considera bueno.

Por otro lado, en la tabla III se presentan los resultados de las métricas explicadas en la sección V que explica las métricas de desempeño.

TABLE III
RESULTADOS DE LAS MÉTRICAS DE DESEMPEÑO.

	Presición	Recall	f1-score	Support
No	0.95	0.90	0.93	7310
Sí	0.45	0.62	0.52	928
Accuracy			0.87	8238
Macro avg	0.70	0.76	0.72	8238
Weighted	0.89	0.87	0.88	8238

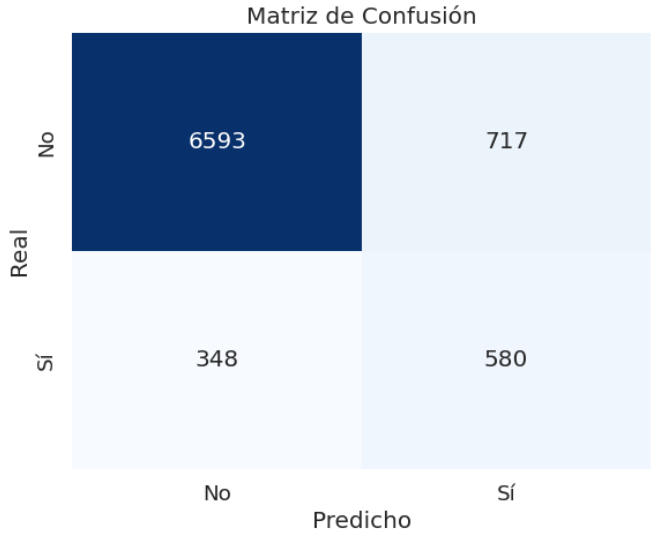

Fig. 1. Matriz de confución.

BIBLIOGRAFÍA

[1] Fawaz J Alsolami, Farrukh Saleem, and AL Abdullah. "Predicting the accuracy for telemarketing process in banks using data mining". In: *Comp. It. Sci* 9 (2020), pp. 69–83.

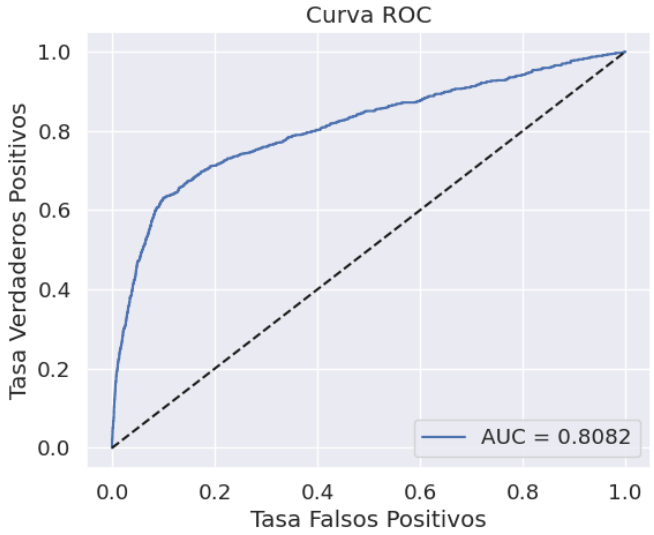

Fig. 2. Curva ROC

TABLE IV
COMPARACIÓN DE LOS VALORES PROPUESTOS PARA LOS HIPERPARÁMETROS DE CADA MODELO.

Modelo	Hiperp.	V. eval[1]	V.E.[2]	V.A.[3]	V.D.[4]
Regresión Logística	C	{0.01, 0.1, 1, 10}	{0.01, 0.1, 1, 10, 100 }	No reportado, se usó BFGS (rminer)	**1.0** (valor por defecto)
	penalty	$\{'l2'\}$	$\{'l1','l2'\}$	$\{'l2'\}$	**'l2'**
K-NN	n_neighbors	{ 3, 5, 7, 11 }	{ 3, 5, 7, 9, 11 }	No reportado	5
	metric	{ 'euclidean' }	{ 'euclidean', 'manhattan'}	No reportado	**'minkowski'** (equivale a Euclidean si p=2)
Random Forest	n_estimators	{ 100, 200 }	{ 100, 200, 500, 1000 }	No reportado	100
	max_depth	{ None, 10, 20 }	{ None, 5, 10, 20, 30 }	No reportado	**None** (crece hasta que la hoja es pura o min_samples)
	max_features	{ 'sqrt', 'log2' }	{ 'sqrt', 'log2', None }	No reportado	**'sqrt'** (para clasificación)
Red Neuronal MLP	hidden_layer_sizes	{(50,), (100,), (100, 50)}	{(10,), (50,), (100,), (100, 50)}	{11, 22} neuronas ocultas	**(100,)**
	activation	{ 'relu', 'tanh'}	{ 'relu', 'tanh', 'logistic' }	No especificado	**'relu'**
	learning_rate_init	{ 0.001, 0.01 }	{ 0.001, 0.01, 0.1 }	No especificado	**0.001**
SVM	C	{ 0.1, 1, 10 }	{ 0.1, 1, 10, 100 }	Usaron C = 3	1.0
	kernel	{'linear', 'rbf'}	{'linear', 'rbf', 'poly'}	Usaron RBF kernel	**'rbf'**
	gamma	{ 'scale', 'auto' }	$\{ 2^k, k \in [-15, 3] \}$	gamma evaluada como 2^k	**'scale'**

[1] Valores evaluados.
[2] Valores estándar (literatura).
[3] Valores reportados por artículos relacionados.
[4] Valores por defecto o con mínimos cambios (Scikit-learn).

[2] Sérgio Moro, Paulo Cortez, and Paulo Rita. "A data-driven approach to predict the success of bank telemarketing". In: *Decision Support Systems* 62 (2014), pp. 22–31.