PROYECTO DE AULA: “APLICACIÓN DEL APRENDIZAJE AUTOMÁTICO PARA LA BASE DE DATOS BANK MARKETING”

Modelos y Simulación de Sistemas II

Entregable II

Est. Rony Santiago Bañol Rico
Est. Johan Sebastian Henao Cañas
Universidad de Antioquia, Colombia
rony.banol@udea.edu.co
johan.henao1@udea.edu.co

I. RESUMEN

Este estudio se enfoca en la predicción del éxito de las campañas de telemarketing bancario para la venta de depósitos a plazo. El objetivo es construir un modelo de clasificación que pueda predecir si un cliente se suscribe o no a un depósito a plazo. Se emplearon técnicas de aprendizaje automático utilizando modelos de Gradient Boosted Tree, Random Forest y Decision Tree. Los datos utilizados fueron recopilados durante cinco años por una institución bancaria portuguesa. Se enfrentaron desafíos como el desequilibrio de clases, abordado mediante técnicas de submuestreo y sobremuestreo. Los resultados muestran que el modelo Gradient Boosted Tree alcanzó la mejor precisión, con un área bajo la curva de 0.8. Este estudio destaca el potencial de la aplicación de aprendizaje automático en la optimización de campañas de telemarketing bancario.

Palabras clave: *Telemarketing bancario, Predicción de suscripción, Depósito a plazo, Aprendizaje automático, Gradient Boosted Tree, Random Forest, Decision Tree, Desbalanceo de clases, Submuestreo, Sobremuestreo, Optimización de campañas, Modelo de clasificación, Análisis de variables, Selección de características, Sobreajuste (Overfitting), Subajuste (Underfitting), Evaluación de modelos, Métricas de rendimiento.*

II. INTRODUCCIÓN

En el ámbito de la industria bancaria, las campañas de telemarketing representan una estrategia fundamental para promover productos financieros y aumentar la participación de los clientes. En este contexto, la capacidad de predecir si un cliente potencial se suscribirá o no a un depósito a plazo puede ser crucial para optimizar los recursos y mejorar la efectividad de dichas campañas. En este estudio, nos centramos en la aplicación de técnicas de aprendizaje automático para abordar este desafío en el contexto de una institución bancaria portuguesa.

El objetivo principal de este trabajo es construir un modelo de clasificación preciso que pueda predecir la suscripción de los clientes a depósitos a plazo basándose en una serie de variables seleccionadas previamente. Para lograr este objetivo, llevamos a cabo un análisis exhaustivo de datos recopilados durante campañas de telemarketing realizadas por la institución bancaria, abarcando un período que incluye los efectos de la crisis financiera global.

Nuestra metodología se centra en el uso de tres modelos de aprendizaje automático: Gradient Boosted Tree, Random Forest y Decision Tree. Estos modelos fueron seleccionados debido a su capacidad para manejar conjuntos de datos complejos y su eficacia en problemas de clasificación. Además, abordamos el desafío del desbalance de clases en nuestros datos utilizando técnicas de submuestreo y sobremuestreo para mejorar la capacidad predictiva de nuestros modelos.

A través de este estudio, buscamos proporcionar a los gerentes de campañas de telemarketing una herramienta valiosa para optimizar sus estrategias y mejorar la eficacia de sus campañas. Además de la construcción de modelos, también exploramos la importancia de las variables seleccionadas y su impacto en las decisiones de suscripción de los clientes.

III. METODOLOGÍA

Para llevar a cabo este proyecto, utilizamos un conjunto de datos recopilados durante las campañas de telemarketing de una institución bancaria portuguesa. Este conjunto de datos proviene del repositorio de Machine Learning de la UCI (UCI Machine Learning Repository) y contiene información demográfica y de contacto de los clientes, así como el resultado de las campañas de telemarketing anteriores.

El conjunto de datos original consta de 45211 muestras y 17 variables, incluyendo características como edad, estado civil, nivel educativo, saldo bancario, duración de la llamada, entre otros. La variable objetivo es binaria y representa si el cliente se suscribió o no a un depósito a plazo.

Se realiza preliminarmente el análisis exploratorio de las variables para la bases de datos, se empieza con el análisis individual de las variables, sus datos para cada registro y si

tienen datos atípicos o poco influyentes con respecto a la variable objetivo.

Se opta por la eliminación de las variables "pdays" y "Poutcome" debido que los valores faltantes corresponden al 81% de los registros de la base de datos, además, se decide eliminar la variable "contact" dado que el 28% de los registros de la base de datos tienen datos faltantes para esta variable. Su exclusión no se espera que tenga un impacto significativo en la variable objetivo "Y" "**Tabla 1**".

Los valores faltantes se sustituyen por "Unknown" según la fuente de la base de datos, donde "NaN" representa "Unknown". Se opta por mantener estos registros en lugar de eliminarlos, ya que aunque la información es desconocida, sigue siendo relevante para el análisis al mantener la proporción relativa respecto a las otras categorías.

Columna	Porcentaje de NaN
age	0.00%
job	0.64%
marital	0.00%
education	4.11%
default	0.00%
balance	0.00%
housing	0.00%
loan	0.00%
contact	28.80%
day_of_week	0.00%
month	0.00%
duration	0.00%
campaign	0.00%
pdays	81.74%
previous	0.00%
poutcome	81.75%
y	0.00%

Tabla 1: *Datos faltantes por variable.*

Se realiza el análisis de las distribuciones de cada variable respecto a la variable objetivo, (Figura 1, Figura 2)

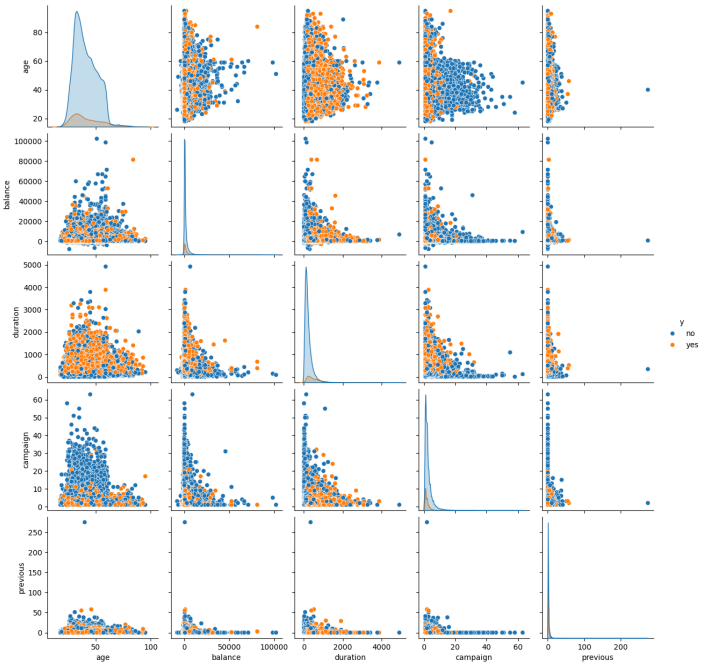

Figura 1: *Distribución de variables numéricas.*

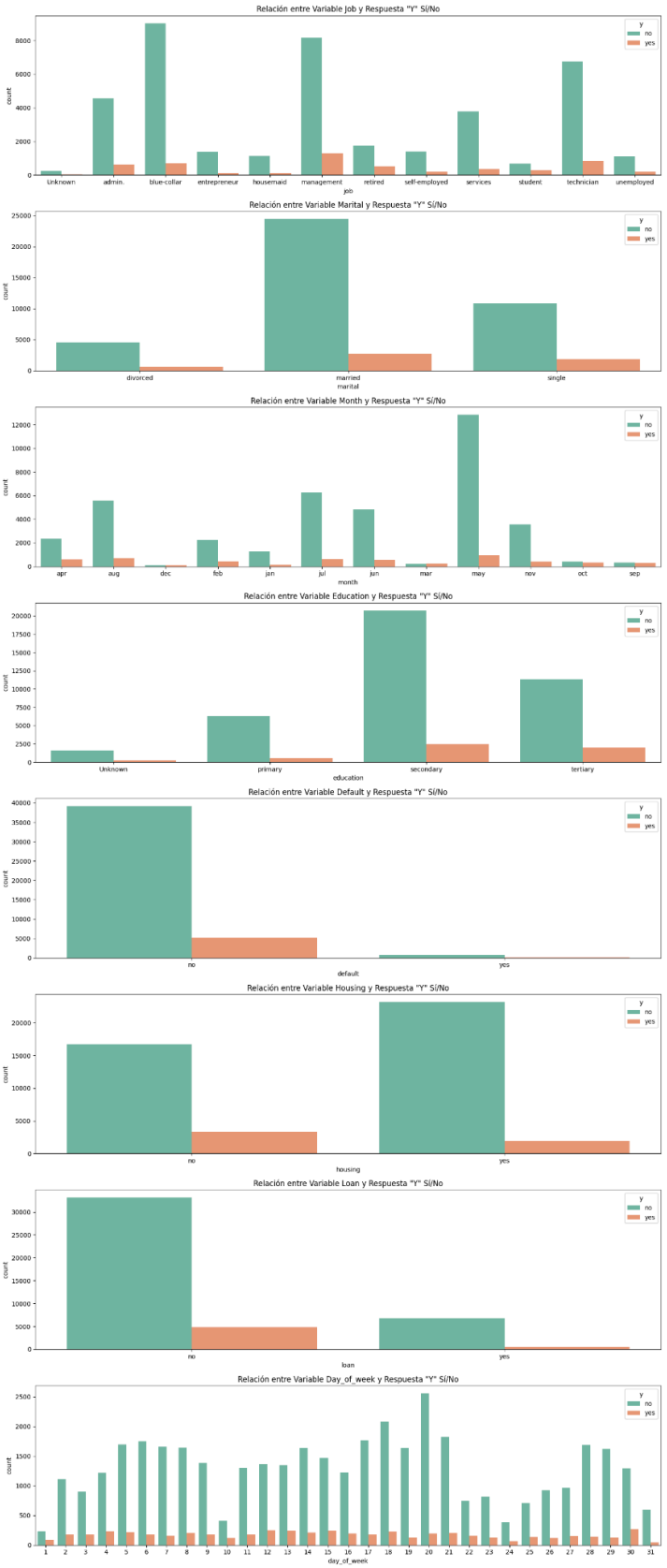

Figura 2: *Distribución de variables categóricas.*

La tendencia de todas las categorías de las variables es tener un mayor número de registros con "no" en la variable respuesta, lo cual indica lo desbalanceada que está la base de datos.
Las variables numéricas muestran un comportamiento similar a las variables categóricas, donde la mayoría de los valores resultan en un "no" en la variable respuesta. Por lo tanto, se procede a realizar un análisis individual de las variables y las correlaciones entre ellas.

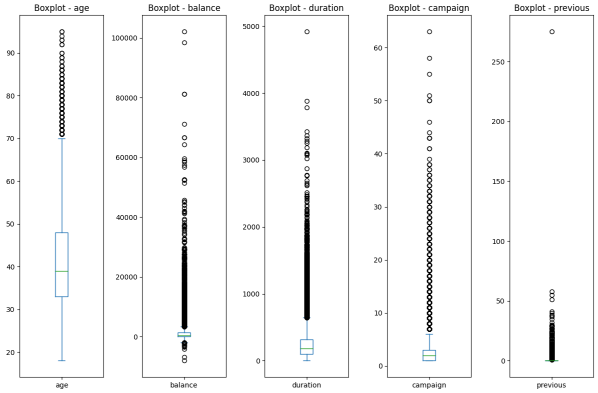

Figura 3: *Boxplot de variables numéricas.*

Se observan valores atípicos en todas las variables, los cuales tienen sentido según el contexto de cada una. Por lo tanto, no se opta por eliminarlos. Solo se eliminó un registro en el cual se mostraba que una persona había recibido más de 250 llamadas.
Para implementar las métricas de validación y la implementación de los modelos de predicción se usaron las bases de datos original, la base de datos con submuestreo y la base de datos con sobremuestreo, así es posible comparar el desempeño de los datos desbalanceados (Figura 4) respecto a la base de datos balanceada.

Figura 4: *Clases desbalanceadas de la base de datos.*

A. Métricas de Evaluación:

Se implementaron 3 modelos de predicción para cada una de las bases de datos: Árbol de Decisión, Bosque Aleatorio y Árbol de Impulso Gradiente, antes de realizar el análisis individual de cada modelo, se hace el análisis de las métricas de evaluación, mediante la librería Pycaret, se calculan las métricas de desempeño: Accuracy, AUC, Recall, Prec, F1, Kappa, MCC y se tabulan los resultados en la tabla 2.

Modelo	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
Modelo con base de datos original.							
Gradient Boosting C.	0.8987	0.9094	0.3701	0.6110	0.4608	0.4086	0.4243
Random Forest	0.8977	0.9111	0.3485	0.6106	0.4431	0.3914	0.4105
Decision Tree C.	0.8626	0.6785	0.4381	0.4171	0.4274	0.3494	0.3495
Modelo con base de datos con sobremuestreo.							
Gradient Boosting C.	0.9530	0.9975	0.9936	0.9189	0.9548	0.9059	0.9089
Random Forest	0.9427	0.9427	0.9912	0.9035	0.9453	0.8853	0.8895
Decision Tree C.	0.8470	0.9153	0.8695	0.8321	0.8504	0.6940	0.6947
Modelo con base de datos con submuestreo.							
Gradient Boosting C.	0.8401	0.9065	0.8590	0.8277	0.8430	0.6802	0.6808
Random Forest	0.8362	0.9098	0.8522	0.8257	0.8388	0.6723	0.6727
Decision Tree C.	0.7667	0.7667	0.7615	0.7695	0.7654	0.5335	0.5337

Tabla 2. *Resultado de métricas de desempeño*

Los resultados resaltan la importancia de abordar el desbalance de clases en conjuntos de datos desbalanceados. El submuestreo emerge como una técnica efectiva para mejorar el rendimiento de los modelos en este contexto, especialmente para el modelo Random Forest Classifier. Estos hallazgos subrayan la necesidad de considerar técnicas de validación adecuadas y métricas de evaluación específicas para conjuntos de datos desbalanceados al desarrollar modelos de aprendizaje automático para la clasificación de datos

La métrica principal de evaluación es el accuracy, comúnmente utilizado para medir la precisión global de un modelo de clasificación. Sin embargo, dado el desbalance de clases presente en la base de datos original, es importante considerar métricas adicionales que proporcionen una evaluación más completa del rendimiento del modelo.

Además del accuracy, se utilizarán las siguientes métricas:

- Recall (Sensibilidad): Indica la proporción de casos positivos correctamente identificados por el modelo.
- Precision (Precisión): Indica la proporción de casos positivos identificados correctamente por el modelo, entre todos los casos positivos identificados.
- F1-score: Es la media armónica entre la precisión y la sensibilidad, proporcionando una medida balanceada entre ambas métricas.

Estas métricas serán especialmente útiles para evaluar el rendimiento de los modelos en los conjuntos de datos con desbalance de clases, como la base de datos original. Permitirán una evaluación más precisa del rendimiento del modelo, teniendo en cuenta la capacidad del modelo para identificar correctamente las muestras positivas y negativas, así como la proporción de falsos positivos y falsos negativos.

Además, se utilizará la técnica de validación cruzada con k-folds = 4 para mitigar cualquier sesgo en la evaluación del modelo debido a la partición de los datos. Esto asegurará una evaluación robusta del rendimiento del modelo en diferentes subconjuntos de datos.

B. Análisis y Resultados:

Decision tree:

Para cada evaluación, se registraron la eficiencia de entrenamiento y prueba, así como la desviación estándar asociada (**Tabla 3**). Además, se realizó un análisis visual del árbol de decisión generado para el conjunto de datos original (**Figura 6**).

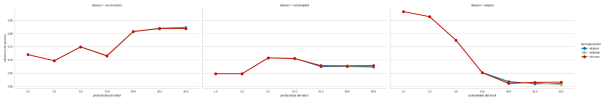

Figura 5: *Análisis de profundidad del árbol.*

Para el conjunto de datos original, se generó un árbol de decisión con una profundidad máxima de 5 niveles. Esta profundidad moderada indica un equilibrio entre la capacidad de generalización y la complejidad del modelo, lo que ayuda a evitar tanto el sobreajuste como el subajuste.

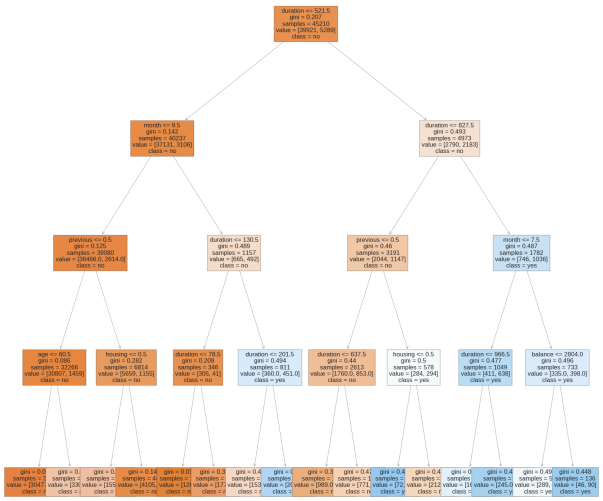

Figura 6: *Árbol de Decision Tree*

Dataset	Train Efc.	Train Std	Test Efc.	Test Std
oversampled	0.793763	0.404603	0.792160	0.405762
subsampled	0.797684	0.401727	0.792060	0.405833
original	0.896124	0.305100	0.893386	0.308621

Tabla 3: *Evaluación del modelo Decision Tree.*

La eficiencia del modelo fue consistente en los conjuntos de datos oversampled y subsampled, aunque ligeramente más baja en comparación con el conjunto de datos original. Y la desviación estándar en los resultados de eficiencia de entrenamiento y prueba refleja la variabilidad inherente en los datos y el proceso de evaluación del modelo

Random Forest:

Dataset	# de árboles	Va. Umbral	Train Efc	Std(train)	Test Efc	Std(test)
oversampled	30.0	2.0	0.999962	0.000022	0.876079	0.067183
oversampled	20.0	2.0	0.999875	0.000049	0.874964	0.067779
oversampled	5.0	2.0	0.996410	0.000591	0.868589	0.062576
subsampled	30.0	2.0	0.999275	0.000164	0.744573	0.061500
subsampled	30.0	5.0	0.999055	0.000289	0.736632	0.061104
subsampled	20.0	5.0	0.997574	0.000431	0.730866	0.063401
original	30.0	2.0	0.999241	0.000067	0.788540	0.068880
original	20.0	2.0	0.997818	0.000379	0.784138	0.074061
original	5.0	2.0	0.988107	0.001607	0.760912	0.082812

Tabla 4: *Evaluación del modelo Random Forest.*

Observamos que la eficiencia de entrenamiento es bastante alta en todos los conjuntos de datos, lo que indica que el modelo es capaz de ajustarse bien a los datos de entrenamiento. Sin embargo, al evaluar el modelo en los datos de prueba, vemos una disminución en la eficiencia, lo que sugiere cierto grado de sobreajuste en el modelo. Pero en general, el aumento en el número de árboles tiende a mejorar la eficiencia del modelo, pero también puede aumentar el tiempo de entrenamiento. Además, la selección adecuada de variables para la división de nodos puede influir en la capacidad del modelo para generalizar bien a datos no vistos.

También vemos que la desviación estándar de las métricas de rendimiento (tanto para el entrenamiento como para la prueba) es relativamente baja en general, lo que indica una cierta consistencia en el rendimiento del modelo en diferentes divisiones de los datos. Sin embargo, en algunos casos, especialmente en el conjunto de datos original, la desviación estándar puede ser un poco más alta, lo que sugiere una mayor variabilidad en el rendimiento del modelo.

El rendimiento del modelo varía según el método de manipulación del desequilibrio de clases y tiende a obtener mejores resultados en los conjuntos de datos subsampled y oversampled en comparación con el conjunto de datos original, lo que indica que la estrategia de balanceo de clases puede mejorar el rendimiento del modelo en este caso.

Gradient Boosted Tree:

Dataset	# de árboles	Train Efc	Std(train)	Test Efc	Std(test)
oversampled	5.0	0.788186	0.015299	0.745709	0.056499
oversampled	20.0	0.816942	0.018669	0.733760	0.063929
oversampled	30.0	0.830828	0.018200	0.739571	0.068376
subsampled	5.0	0.797504	0.015011	0.747301	0.036266
subsampled	20.0	0.816002	0.010326	0.771126	0.039042
subsampled	30.0	0.829710	0.010391	0.778218	0.039124
original	5.0	0.883020	0.000015	0.882990	0.000045
original	20.0	0.900516	0.006731	0.845786	0.023195
original	30.0	0.904984	0.008702	0.831319	0.036476

Tabla 5 *Evaluación del modelo Gradient Boosted Tree.*

Muestra un buen rendimiento en términos de eficiencia, pero es importante considerar el equilibrio entre la eficiencia y la capacidad de generalización del modelo. Además, la consistencia en las métricas de rendimiento, respaldada por las bajas desviaciones estándar, indica que el modelo es robusto y confiable en diferentes divisiones de los datos.

Se observa una tendencia hacia una mejora en la eficiencia de prueba al aumentar el número de árboles en el modelo. Sin embargo, este aumento en la eficiencia no es significativo después de cierto punto, lo que sugiere que agregar más árboles puede ofrecer rendimientos marginales en términos de mejora del rendimiento.

IV. CONCLUSIONES

Comparación entre los 3 variables modelos de predicción, con los 3 mejores configuraciones por modelo.:

dataset	# de árboles	Train Efc.	std(train)	Test Efc.	std(test)
Gradient Boosted Tree					
original	5.0	0.883020	0.000015	0.882990	0.000045
original	20.0	0.900516	0.006731	0.845786	0.023195
original	30.0	0.904984	0.008702	0.831319	0.036476
Random Forest					
oversampled	30.0	0.999962	0.000022	0.876079	0.067183
oversampled	20.0	0.999875	0.000049	0.874964	0.067779
oversampled	5.0	0.996410	0.000591	0.868589	0.062576
Decision Tree					
oversampled		0.793763	0.404603	0.792160	0.405762
subsampled		0.797684	0.401727	0.792060	0.405833
original		0.896124	0.305100	0.893386	0.308621

Tabla 6 *Comparación mejores configuraciones por cada modelo.*

- Las técnicas de oversampling y subsampling ayudaron a abordar el desequilibrio de clases en el conjunto de datos, lo que resultó en mejoras notables en la eficiencia de prueba y una mayor capacidad de generalización del modelo.

- Se observó que cada modelo tiene sus propias ventajas y limitaciones, y la elección del modelo adecuado depende en gran medida de las características específicas del conjunto de datos y los objetivos del proyecto. Por ejemplo, se encontró que el modelo de gradient boosted trees mostró un buen rendimiento en términos de eficiencia de prueba en el conjunto de datos original, mientras que random forest fue más efectivo después de aplicar técnicas oversample y subsample.

- Los modelos Random Forest tienden a sobreajustarse en el conjunto de datos sobremuestreado evidenciado a una eficiencia de entrenamiento 0.999, lo que resulta en una baja eficiencia de prueba

- Los modelos de Decision Tree tienen un rendimiento más bajo en comparación con Gradient Boosted Tree y Random Forest en ambos conjuntos de datos de muestreo.

V. REFERENCIAS

[2] UCI Machine Learning Repository, "Bank Marketing Dataset", https://archive.ics.uci.edu/dataset/222/bank+marketing, 2024..

[2] TodoBI, "PyCaret: paso a paso," TodoBI, https://todobi.com/pycaret-paso-a-paso/, 2024.

[3] JohanSH7, "Proyecto_BankMarketing," GitHub, 2022. [Online]. Available: https://github.com/JohanSH7/Proyecto_BankMarketing/tree/main

.