Instituto Tecnológico y de Estudios
Superiores de Monterrey

Analítica de Datos y Herramientas de Inteligencia Artificial II

Evidencia 2:
Análisis de aprendizaje automático

EQUIPO 2

André Calmus González	**A017333529**
Diego Sánchez Márquez	**A01734778**
Emilio Rugerio Pastrana	**A01737819**
Ximena Italya Jimenez Huerta	**A01277667**

Grupo 501

26, Mayo 2025

Análisis Exploratorio de Datos (EDA)

Objetivo:

Generar modelos de aprendizaje automático con la finalidad de clasificar la variable objetivo (y: Captación de clientes). De esta manera se ayudará a la dirección del banco para construir un sistema inteligente que logre captar nuevos clientes de diversas áreas. Es por esto que la clase objetivo será ‘yes’. Se busca maximizar el recall de la clase objetivo ‘yes’ para identificar a la mayor cantidad de clientes potenciales, por medio de modelos comparativos con VotingClassifier.

Principales hallazgos del EDA:

- Podemos tomar como valores nulos unknown, pero por mientras podemos destacar que el dataframe tiene 17 columnas con 45211 filas sin valores nulos. Algunas columnas son numéricas (age, balance, day, duration, etc.), mientras que otras son categóricas (job, marital, education, contact, etc.).

Valores "unknown" en las variables categóricas:

- La columna poutcome tiene 36,959 valores unknown, lo cual representa la mayoría de los datos en esa variable.
- Otras columnas como contact (13,020 unknown), education (1,857 unknown) y job (288 unknown) también tienen valores desconocidos.

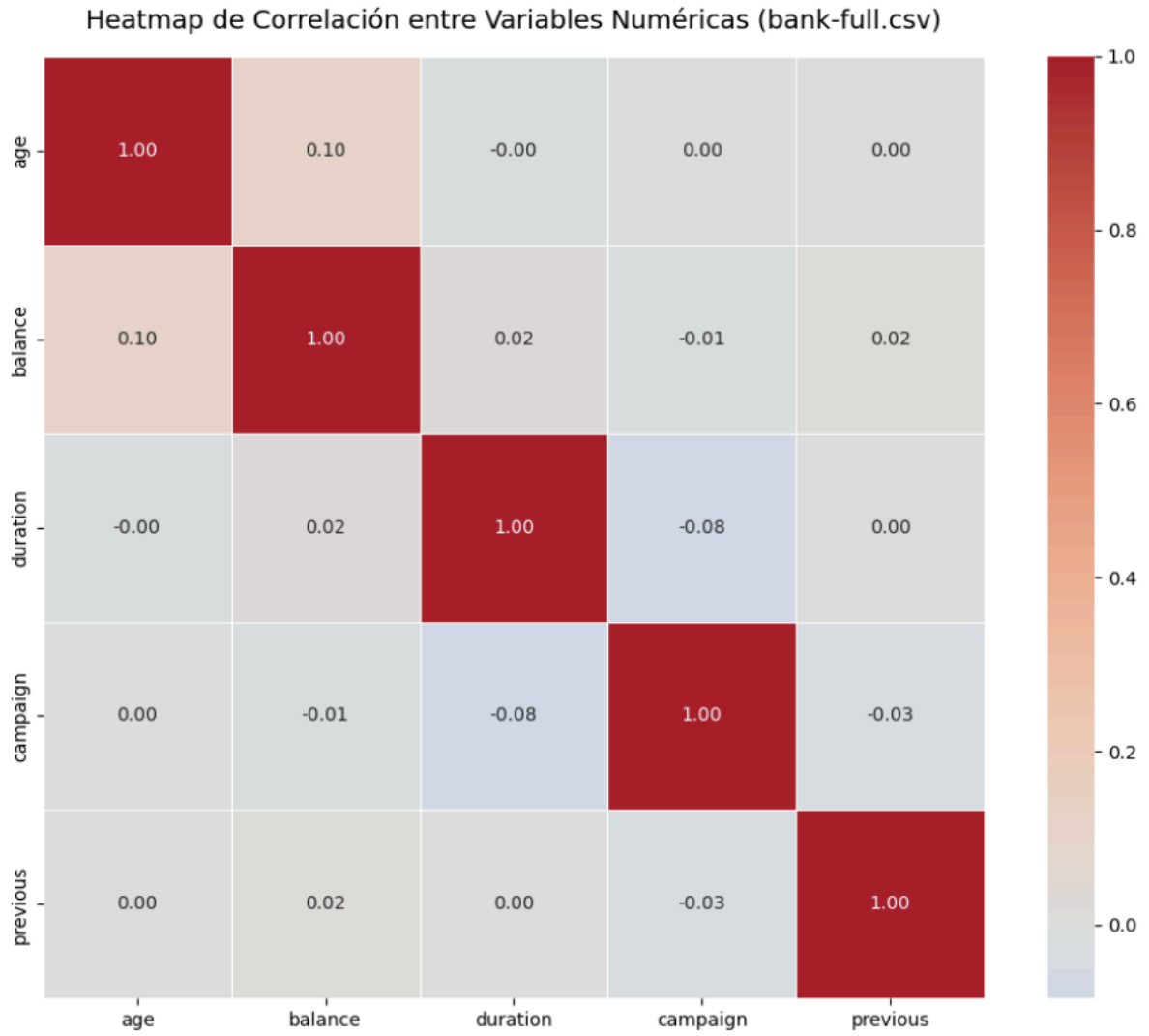
Distribución de la variable objetivo (y):

- La clase 'no' tiene 39,922 observaciones, mientras que 'yes' tiene 5,289.
- Esto sugiere un desbalance de clases, lo que puede afectar el desempeño de ciertos modelos de Machine Learning.

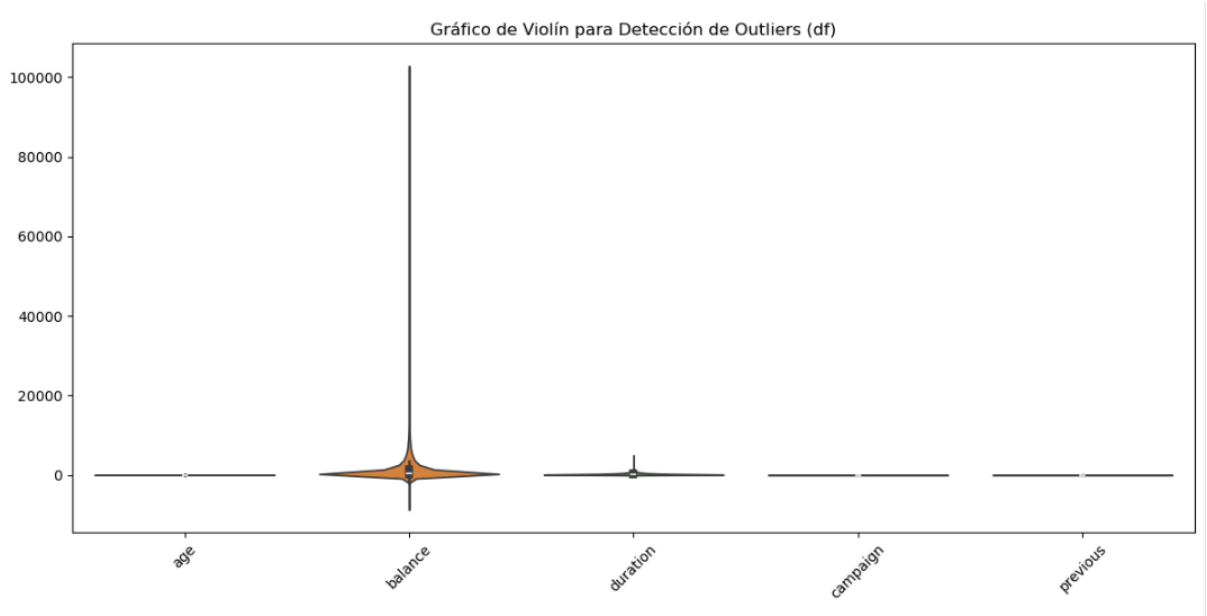
Análisis de correlación

Para analizar las relaciones lineales entre las variables numéricas, se generó un heatmap de correlación. Esto nos permite observar de forma sencilla qué variables están relacionadas entre sí. En el heatmap se observó que la mayoría de las variables muestran correlaciones muy bajas entre sí, esto nos indica poca dependencia entre sus valores. Por ejemplo, la correlación entre age y balance fue de apenas 0.10, mientras que la de duration y campaign fue de -0.08.

Esto nos dice que las variables no presentan una relación lo suficientemente fuerte como para poder concluir que aportan valor a la hora de explicarse entre sí.



Outliers:



- Edad: la mayoría de los clientes son entre 20-60 con mayor densidad entre 30-40, muy pocos por encima de 80 años.
- El balance indica que algunos clientes tienen saldos extremadamente altos en comparación con la mayoría
- Duración de llamada representa que la mayoría de las llamadas duran -500 segundos, con una densa concentración debajo de 200 segundos.
- Campaing la mayoría fueron contactados entre 1-5 veces pero hay ocasiones que van de 10 a 50 contactos

Transformación de datos

Para las columnas ‘job’, ‘education’ y ‘contact’ decidimos reemplazar ‘unknown’ por la moda. En el caso de la columna ‘poutcome’, debido a que el 80% de sus datos son ‘unknown’ decidimos borrar la columna.

- En la columna pdays, debido a que el -1 representa nunca haber sido contactado antes, decidimos generar una nueva columna, ‘pdays_contacted’, donde un valor de 1 representa un contacto previo y 0 representa que no hubo un contacto previo. Esto permite entender mejor dicha columna.
- Se realizó una codificación ordinal a ‘education’ donde primary = 1, secondary = 2, tertiary = 3 y unknown = 0. El nivel educativo puede tener un impacto directo en la

toma de decisiones financieras, por eso se codificó de forma que se mantuviera la jerarquía.

- Se agrupó la columna ‘job’ en categorías, donde student ahora se representa como ‘studying’, unemployed y unknown se representan como ‘unemployed’ y todos los demás tipos de trabajo se categorizaron como ‘working’.
- También se convirtieron los meses en números, dónde enero = 1, febrero = 2, etc.
- Se utilizó one-hot encoding para las variables nominales (job_grouped, marital, contact). Esto permite que los modelos traten las categorías como independientes
- Para las variables binarias (default, housing, loan) se utilizó label encoding. Al ser binarias, pueden representarse de gran manera con valores numéricos.
- Por último se eliminaron las variables irrelevantes para el análisis (marital_single, job) y se cambió el nombre de la columna ‘y’ a ‘clase’.

Sugerencias para mejorar la correlación entre las variables

Como podemos observar, la correlación entre las variables numéricas fue bastante baja, por eso realizamos algunas sugerencias para poder mejorar la calidad de los datos.

- Agrupar o transformar algunas variables ordinalmente en caso de que exista una jerarquía como en las variables de job o education.
- Segmentar el dataframe en grupos donde los registros compartan características similares.
- Deshacerse de los outliers para estabilizar cómo se relacionan las variables y para reducir la importancia de valores que son atípicos.
- Utilizar técnicas como undersampling para poder trabajar con datos más equilibrados entre sus clases.
- Escalar variables como age, duration y balance para que los modelos no se vean afectados por valores muy diferentes.

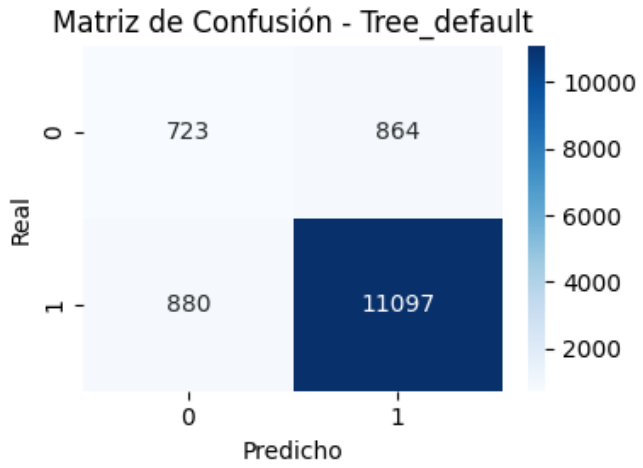
Estas sugerencias podrían ayudar a mejorar la calidad de los datos para que los modelos logren captar patrones con mayor precisión y logren un mejor desempeño.

Resultados de modelos individuales

Se compararon distintos modelos de clasificación con el objetivo de maximizar el recall de la clase ‘yes’, es decir, identificar correctamente a los clientes que aceptarían la oferta.

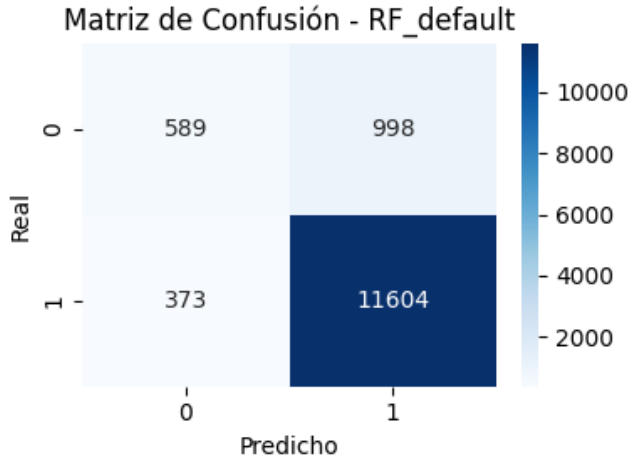
Decision Tree:

- Exactitud: 0.8714
- Precisión: 0.4510
- Recall: 0.4555 (el más alto entre modelos individuales)



Random Forest:

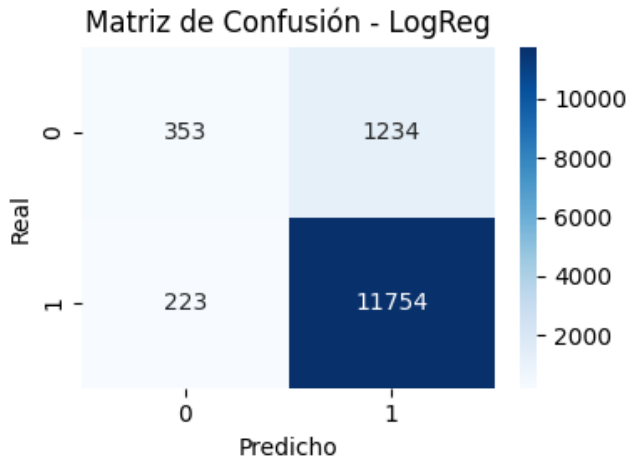
- Exactitud: 0.8989
- Precisión: 0.6122
- Recall: 0.3711
- Mejores métricas de precisión, pero menor recall.



Regresión Logística:

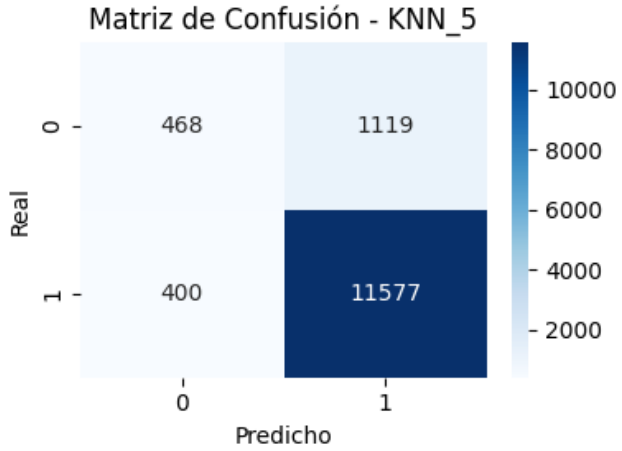
- Exactitud: 0.8925

- Precisión: 0.6128
- Recall: 0.2224
- Buen desempeño general pero mal desempeño en la detección de ‘yes’.



K-Nearest Neighbors (KNN):

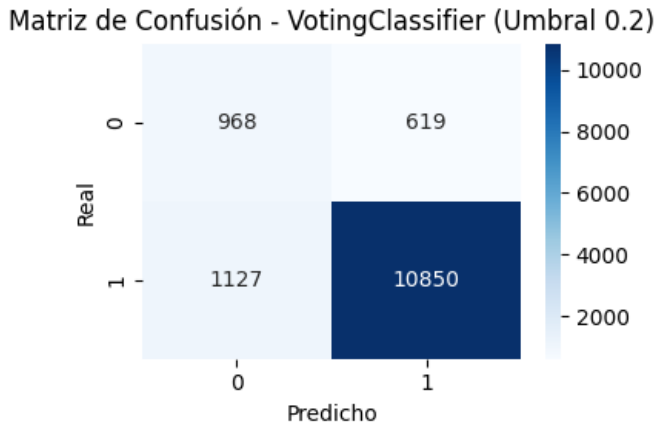
- Exactitud: 0.8880
- Precisión: 0.5391
- Recall: 0.2948
- Resultados buenos, pero peores que los modelos de Decision Tree y Random Forest.



Voting Classifier:

Combinación de modelos como Tree_default y Tree_depth10.

- Exactitud: 0.8713
- Precisión: 0.4621
- Recall: 0.61
- Logra el mejor recall general, pero se pierde algo de precisión.

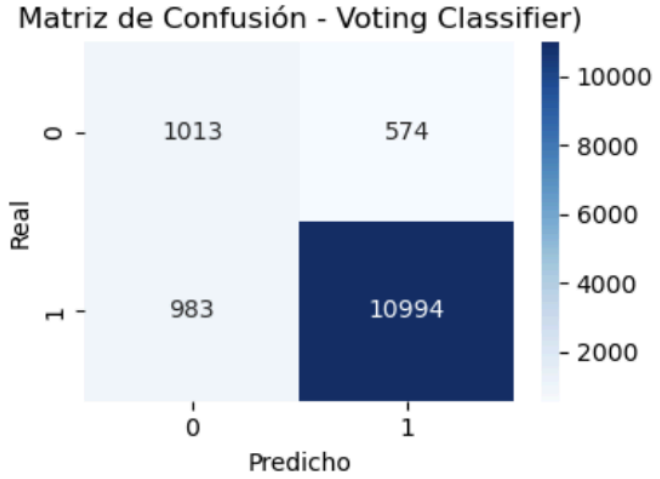


Modelos Destacados:

Comparación Entrenamiento y Evaluación.

RF_default y Tree_depth5:

- Exactitud: Entrenamiento 0.9615 | Evaluación 0.8852
- Recall: Entrenamiento 0.9973 | Evaluación 0.6383
- Precisión: Entrenamiento 0.7533 | Evaluación 0.5075
- F1-score: Entrenamiento 0.8583 | Evaluación 0.5654



Los modelos destacados, RF_default y Tree_depth5, mostraron un gran desempeño en entrenamiento, especialmente en recall, lo que indica una alta capacidad para identificar casos positivos de manera acertada. Sin embargo, en la evaluación se observan resultados más bajos en las métricas. Aun así, sigue teniendo un buen recall, lo cual es clave para el objetivo del proyecto: captar la mayor cantidad de clientes que aceptarían la oferta. Dicho esto se puede decir que se obtuvieron opciones buenas para poder cumplir con nuestro objetivo.

Aplicación del modelo en un caso real

El modelo podría ser utilizado para campañas de marketing del banco para decidir en qué clientes enfocarse a la hora de contactarlos por teléfono. Esto va de la mano con la matriz de costos, ya que un falso positivo nos estaría diciendo que el banco estaría contactando a alguien que no aceptaría la oferta, lo que se vería reflejado como un gasto innecesario. Por otro lado, un falso negativo nos indicaría que no se contactó a alguien que hubiera aceptado.

Al enfocarnos en maximizar el recall, nos estaríamos encargando de evitar dejar pasar clientes potenciales. Es por eso que nuestro modelo permitiría hacer campañas telefónicas más rentables al permitir al banco enfocar sus recursos en un grupo de personas que sean más probables a aceptar las ofertas.

Conclusión

Durante la realización de esta evidencia, utilizamos distintas técnicas para el procesamiento, la codificación y transformación de datos, además de implementar distintos algoritmos para clasificar. Esto se realizó con la finalidad de lograr un modelo capaz de capturar de manera correcta a los clientes que aceptarían una oferta del banco, dando prioridad a maximizar el recall sobre otras métricas.

El modelo obtenido mediante el VotingClassifier, se destacó por tener un recall de 0.61, siendo el resultado más alto para esta métrica de todos los modelos. Esto nos indica una gran capacidad del modelo para capturar correctamente a los clientes potenciales, para que de esta forma puedan enfocar sus recursos en un segmento de clientes en concreto.

Utilizar este tipo de modelos nos mostró su importancia, ya que pueden aportar valor a la hora de tomar decisiones, permitiendo realizar acciones más inteligentes y rentables fundamentadas por un buen análisis.