Machine Learning
Proyecto Final

Alejandro Gómez (274547)
Nicolás Castañeda (273724)
Caren Piñeros (282290)

Machine Learning
Universidad de La Sabana
Carrera: Ingeniería Informática
Profesor: Hugo Franco

1. Introducción

En esta sección se desarrollan los principales elementos que orientan el estudio. Primero, se presenta la formulación del problema y las necesidades de información asociadas. Luego, se expone el marco conceptual, donde se definen los términos y variables clave. Posteriormente, se revisan los antecedentes a partir de trabajos previos relacionados con la temática. Finalmente, se establecen los objetivos del proyecto, tanto generales como específicos, que guían el análisis realizado.

1.1. Formulación del problema y necesidades de información asociadas

En el campo del aprendizaje automático, la selección del modelo adecuado representa uno de los desafíos más determinantes para garantizar la calidad de las predicciones. Aunque diversos algoritmos pueden resolver un mismo problema de clasificación, sus desempeños varían considerablemente según la naturaleza de los datos, el número de variables, el balance de clases y los hiperparámetros seleccionados. Por ello, comprender las diferencias prácticas entre modelos de distinta naturaleza —como las *redes neuronales artificiales* y los *métodos ensemble basados en árboles de decisión*— constituye una tarea esencial para el diseño de soluciones predictivas robustas y reproducibles.

En este contexto, el presente estudio busca comparar la capacidad predictiva y la estabilidad de tres modelos de clasificación supervisada: *Red Neuronal*, *Random Forest* y *XGBoost*, aplicados al conjunto de datos *Bank Marketing* del repositorio UCI. El objetivo no se limita a alcanzar la mayor precisión posible, sino a evaluar de manera controlada cómo las diferencias en la estructura de aprendizaje, el ajuste de hiperparámetros y el manejo de desbalance afectan el rendimiento general del sistema.

Para tal fin, se implementó un flujo orquestado mediante **Prefect 2.0**, herramienta que permite automatizar, monitorear y registrar cada etapa del ciclo de vida del modelo: carga de datos, preprocesamiento, entrenamiento, evaluación y almacenamiento de resultados. Este enfoque garantiza la trazabilidad completa del proceso y facilita la comparación objetiva entre los modelos evaluados.

1.2. Marco conceptual

El análisis se fundamenta en conceptos centrales del aprendizaje automático orientado a la clasificación binaria. A continuación, se describen los términos relevantes:

- **Pipeline de Machine Learning**: secuencia estructurada de tareas que abarca desde la adquisición y limpieza de datos hasta la evaluación final del modelo. Su orquestación mediante Prefect permite automatizar la ejecución y registro de cada fase [Prefect Technologies, 2022].
- **Random Forest**: método de conjunto que combina múltiples árboles de decisión entrenados sobre subconjuntos aleatorios del conjunto de datos, con el fin de reducir la varianza y mejorar la capacidad de generalización [Breiman, 2001].
- **Red Neuronal (NN)**: modelo inspirado en el cerebro humano, compuesto por capas de neuronas artificiales conectadas entre sí. Su principal fortaleza radica en la capacidad para capturar relaciones no lineales complejas [Haykin, 2009].
- **XGBoost**: algoritmo de *boosting* que construye secuencialmente árboles de decisión, corrigiendo los errores del modelo anterior y optimizando una función de pérdida diferenciable [Chen & Guestrin, 2016].
- **Orquestación de flujos con Prefect**: enfoque de automatización que permite definir dependencias entre tareas, gestionar fallos, reproducir ejecuciones y almacenar artefactos y logs de manera estructurada [Orchestrators Review, 2023].

Estos elementos conceptuales permiten establecer un marco comparativo no solo desde el punto de vista de las métricas de clasificación, sino también desde la eficiencia computacional, interpretabilidad y reproducibilidad del proceso experimental.

1.3. Antecedentes

La base empírica del estudio proviene del trabajo de Moro, Cortez y Rita (2014), publicado en *Decision Support Systems*, donde se propone un modelo

de minería de datos para predecir el éxito de llamadas telefónicas en campañas bancarias [Moro et al., 2014]. En dicho estudio se compararon técnicas como regresión logística, árboles de decisión, redes neuronales y máquinas de vectores de soporte, demostrando que las redes neuronales ofrecieron el mejor desempeño en términos del área bajo la curva ROC (AUC = 0.8).

Complementariamente, estudios recientes como los de Zhang et al. (2021) resaltan la importancia de realizar comparaciones sistemáticas entre modelos de aprendizaje automático, enfatizando que la selección del algoritmo depende tanto del contexto de los datos como de la interpretabilidad requerida [Zhang et al., 2021].

El presente proyecto retoma el conjunto de datos clásico de Moro et al., pero introduce un enfoque moderno de automatización y experimentación reproducible mediante Prefect 2.0. Además, se amplía la comparación incluyendo modelos de ensamble como *Random Forest* y *XGBoost*, los cuales no formaban parte del estudio original. Esta extensión permite contrastar cómo los avances en algoritmos de agregación y optimización de gradiente han redefinido el panorama de la clasificación supervisada, especialmente frente a las redes neuronales de arquitectura clásica.

De igual modo, el uso de un flujo orquestado aporta un componente metodológico clave: la posibilidad de ejecutar pruebas sistemáticas, registrar hiperparámetros y validar los resultados bajo condiciones controladas, fortaleciendo la validez de la comparación entre modelos.

1.4. Objetivos del proyecto

Objetivo general: Comparar el desempeño y comportamiento de los modelos *Red Neuronal*, *Random Forest* y *XGBoost* en la tarea de predicción binaria, evaluando sus diferencias en precisión, interpretabilidad y eficiencia dentro de un flujo orquestado con Prefect 2.0.

Objetivos específicos:

1. Implementar un pipeline automatizado con Prefect para gestionar las etapas de preprocesamiento, entrenamiento, validación y almacenamiento de resultados.

2. Ajustar los hiperparámetros de cada modelo mediante estrategias de búsqueda aleatoria y comparar su impacto en las métricas finales.

3. Analizar las diferencias entre los modelos en términos de capacidad predictiva, estabilidad y complejidad computacional.

4. Evaluar las ventajas de los métodos ensemble frente a un modelo de árbol único y discutir su relevancia en escenarios reales de toma de decisiones.

1.5. Datos empleados

Los datos utilizados corresponden al conjunto *Bank Marketing*, proveniente del repositorio de la Universidad de California, Irvine (UCI Machine Learning Repository). Este conjunto contiene 45,211 registros y 16 atributos relacionados con campañas de marketing telefónico de una institución bancaria portuguesa. La variable objetivo es binaria e indica si el cliente suscribió o no un depósito a plazo (*yes/no*).

Las variables incluyen características demográficas del cliente (edad, profesión, estado civil, nivel educativo), información financiera (crédito, préstamos, saldo promedio), y datos relacionados con la campaña (número de contactos previos, duración de la llamada, mes y tipo de comunicación). Estas características sirven como base para entrenar, ajustar y evaluar los tres modelos comparados en el estudio.

Metodología

1. Descripción general

El proyecto implementó un flujo automatizado de *Machine Learning* usando **Prefect 2.0**, con el objetivo de predecir si un cliente aceptará una oferta bancaria, basándose en el conjunto de datos público *Bank Marketing Dataset* de la UCI.

El flujo fue diseñado para ejecutar todas las etapas de un pipeline de aprendizaje supervisado: carga, preprocesamiento, entrenamiento, optimización y evaluación de modelos, garantizando trazabilidad y reproducibilidad.

2. Flujo general del pipeline

El flujo principal (`main_flow`) se desarrolló en Python y gestionado mediante Prefect, se ejecutaron las siguientes tareas:

1. **Carga de datos (`load_data`)**: Los datos se descargaron directamente desde el repositorio UCI (`bank-additional-full.csv`), con 41,188 registros y 21 variables.

2. **Preparación de los datos (`prepare_data`)**:

   - Conversión de la variable objetivo `y` a binaria (`yes` $\rightarrow$ 1, `no` $\rightarrow$ 0).
   - Identificación de variables categóricas y numéricas.
   - Codificación *One-Hot Encoding* para las variables categóricas.
   - Escalado de variables numéricas con *StandardScaler*.
   - División en conjuntos de entrenamiento y prueba (80/20) con estratificación.
   - Balanceo del conjunto de entrenamiento con **SMOTE**, para mitigar el desbalance de clases.

   Al finalizar, se obtuvieron matrices de entrenamiento y prueba con 63 características.

3. **Entrenamiento del modelo base**: Se entrenó un árbol de decisión (*Decision Tree*) como línea base, obteniendo las siguientes métricas:

   Tabla 1: Métricas del modelo baseline (Decision Tree).

Métrica	Accuracy	Precision	Recall	F1	ROC AUC
Valor	0.8816	0.478	0.550	0.511	0.737

4. **Optimización de modelos**: Se implementaron tres estrategias de optimización de hiperparámetros:

   a) **Red Neuronal (Keras + SciKeras)**: Se realizó una búsqueda manual de 12 iteraciones con barra de progreso usando `tqdm`, variando parámetros como número de neuronas, función de activación, optimizador y tasa de aprendizaje.

   Mejores hiperparámetros para la red neuronal:

```
{
  "model__neurons": 64,
  "model__activation": "tanh",
```

```
  "model__optimizer": "adam",
  "model__learning_rate": 0.001,
  "epochs": 15,
  "batch_size": 64
}
```

b) **Random Forest**: Se aplicó `RandomizedSearchCV` con 24 combinaciones posibles.

Mejores hiperparámetros para Random Forest:

```
{
  "n_estimators": 200,
  "min_samples_split": 2,
  "max_features": "log2"
}
```

c) **XGBoost**: Se ajustaron parámetros relacionados con la profundidad, número de árboles y tasa de aprendizaje.

Mejores hiperparámetros para XGBoost:

```
{
  "n_estimators": 200,
  "max_depth": 6,
  "learning_rate": 0.1,
  "subsample": 0.8
}
```

5. **Evaluación de modelos**: Cada modelo se evaluó sobre el conjunto de prueba. Los resultados consolidados se muestran en la siguiente tabla:

Tabla 2: Comparación de métricas entre modelos.

Modelo	Accuracy	Precision	Recall	F1	ROC AUC
Decision Tree	0.8816	0.478	0.550	0.511	0.737
Neural Network	0.8908	0.509	**0.832**	0.632	0.941
Random Forest	**0.9148**	**0.637**	0.565	0.599	**0.945**
XGBoost	∼0.91	∼0.63	∼0.58	∼0.60	∼0.94

3. Herramientas y librerías

El desarrollo se realizó en Python 3.11, utilizando las siguientes librerías principales:

- **Prefect 2.12** – Orquestación y monitoreo del flujo.
- **scikit-learn** – Preprocesamiento, métricas y modelos base.
- **imblearn** – Balanceo con SMOTE.
- **TensorFlow / Keras** – Implementación de redes neuronales.
- **XGBoost** – Modelo de boosting de gradiente.
- **SciKeras** – Integración entre Keras y scikit-learn.
- **tqdm / joblib** – Visualización de progreso y persistencia de artefactos.

4. Registro y artefactos

Todos los resultados del flujo se almacenaron automáticamente en la carpeta `artifacts/`, incluyendo:

- Archivos de parámetros: `best_nn_params.json`, `best_rf_params.json`, `best_xgb_params.json`.
- Métricas: `baseline_dt_metrics.json`, `nn_metrics.json`, `rf_metrics.json`.
- Modelos entrenados en formato `.h5` y `.joblib`.
- Log de ejecución: `Log_prefect_flow.txt`.

5. Consideraciones de diseño

- Se priorizó la reproducibilidad mediante Prefect, que permite versionar y registrar tareas.
- Se aplicó balanceo de clases para reducir sesgos hacia la clase mayoritaria.
- Se limitó la búsqueda de hiperparámetros para mantener tiempos de cómputo razonables.

- Se manejaron excepciones durante el entrenamiento de redes neuronales para garantizar la continuidad del flujo.

2. Resultados

2.1. Contexto del problema y significado de las métricas

El objetivo es predecir si un cliente aceptará una oferta telefónica de depósito a plazo en el dataset Bank Marketing de UCI Con la convención $y = 1$ para "acepta" y $y = 0$ para "no acepta", las salidas del clasificador con umbral 0.5 se interpretan así:

Verdadero Positivo (TP) el modelo predice "acepta" y el cliente efectivamente acepta Implica una llamada acertada y potencial conversión

Falso Positivo (FP) el modelo predice "acepta" pero el cliente no acepta Implica costo operativo de contacto sin conversión

Falso Negativo (FN) el modelo predice "no acepta" pero el cliente habría aceptado Implica pérdida de una oportunidad de conversión

Verdadero Negativo (TN) el modelo predice "no acepta" y el cliente no acepta Evita un contacto infructuoso

En este dominio, una organización puede priorizar Recall (reducir FN) para no "perder" posibles clientes dispuestos a aceptar, o Precision (reducir FP) para optimizar costos de llamadas Por ello se reportan métricas complementarias: Accuracy, Precision, Recall, F1 y ROC AUC

2.2. Datos, preparación y partición

El flujo descargó el conjunto "bank-additional-full.csv" (41,188 filas, 21 columnas) desde UCI Tras la preparación (OHE para categóricas, estandarización de numéricas, división estratificada 80/20 y balanceo del entrenamiento con SMOTE), se obtuvieron matrices con 63 características; el conjunto de entrenamiento balanceado por SMOTE quedó en 58,476 instancias y el de prueba en 8,238

2.3. Desempeño comparativo en prueba

La Tabla 1 resume el rendimiento de los modelos evaluados sobre el conjunto de prueba: Árbol de Decisión (baseline), Red Neuronal, Random Forest y XGBoost Los resultados provienen de los artefactos generados por el flujo orquestado

Tabla 3: Métricas en prueba por modelo (Bank Marketing UCI)

Modelo	Accuracy	Precision	Recall	F1	ROC AUC
Decision Tree (baseline)	0.8816	0.4780	0.5496	0.5113	0.7367
Red Neuronal	0.8908	0.5092	**0.8319**	0.6318	0.9408
Random Forest	0.9148	0.6375	0.5647	0.5989	0.9447
XGBoost	**0.9221**	**0.6589**	0.6390	**0.6488**	**0.9533**

Hallazgos clave:

- Mejor modelo global: XGBoost obtiene el mayor AUC (0.9533), la mayor F1 (0.6488) y la mayor Accuracy (0.9221), además de la mejor Precision (0.6589)
- Máxima sensibilidad: la Red Neuronal alcanza el Recall más alto (0.8319), lo que la hace muy efectiva para reducir falsos negativos y capturar más clientes que aceptarían la oferta
- Ventaja frente a un árbol único: frente al baseline, XGBoost mejora +0.040 en Accuracy (0.9221 vs 0.8816), +0.181 en Precision (0.6589 vs 0.4780), +0.089 en Recall (0.6390 vs 0.5496), +0.138 en F1 (0.6488 vs 0.5113) y +0.217 en AUC (0.9533 vs 0.7367)
- Random Forest también supera al árbol único en todas las métricas, con AUC 0.9447

Interpretación operacional:

- XGBoost reduce simultáneamente FP y FN respecto al árbol de decisión, aumentando tanto la tasa de aciertos globales (Accuracy) como el balance precisión–recuperación (F1) y la discriminación por umbral (AUC)
- La Red Neuronal, al maximizar Recall, minimiza FN: disminuye la pérdida de oportunidades de conversión, aunque con más FP relativos que XGBoost (Precision inferior)

2.4. Selección de modelos e hiperparámetros

El flujo ejecutó búsqueda de hiperparámetros por modelo y registró el mejor score de validación y la configuración seleccionada:

- Red Neuronal (12 configuraciones, CV=3, métrica Accuracy): mejor Accuracy-CV=0.9177; mejores hiperparámetros: `neuronas=64`, `activación=tanh`, `optimizador=adam`, `learning_rate=0.001`, `epochs=15`, `batch_size=64`
- Random Forest (RandomizedSearchCV, 24 candidatos, CV=4, métrica AUC): mejor AUC-CV=0.9947; mejores hiperparámetros: `n_estimators=200`, `min_samples_split=2`, `max_features=log2`, `max_depth=None`
- XGBoost (RandomizedSearchCV, 24 candidatos, CV=4, métrica AUC): mejor AUC-CV=0.9932; mejores hiperparámetros: `n_estimators=200`, `max_depth=6`, `learning_rate=0.1`, `subsample=0.8`, `colsample_bytree=1.0`

Nota: aunque la NN se validó con Accuracy y RF/XGB con AUC, la comparación final se hace en el conjunto de prueba con el mismo conjunto de métricas, lo que mantiene la equidad del contraste

2.5. Costo computacional observado

- Baseline Decision Tree: 0.60 s de entrenamiento
- Tuning Random Forest: 59.8 s (96 fits: 4 pliegues $\times$ 24 configuraciones)
- Tuning XGBoost: 55.5 s (96 fits: 4 pliegues $\times$ 24 configuraciones)
- Tuning Red Neuronal: 12 iteraciones con CV=3, duración total $\approx$ 11 min 40 s

Los ensambles (RF/XGB) ofrecieron una relación desempeño–tiempo muy favorable en CPU, superando al baseline y registrando tiempos de ajuste moderados

2.6. Gráficas comparativas

Las figuras generadas a partir de los modelos entrenados y del conjunto de prueba orquestado con Prefect permiten verificar visual y cuantitativamente las conclusiones del estudio. Todas las imágenes se produjeron con

los artefactos persistidos en `artifacts/` y el mismo particionado de datos, garantizando consistencia con las métricas reportadas en los JSON de resultados.

2.6.1. Curvas ROC comparativas

La Figura 1 muestra las curvas ROC para los cuatro modelos evaluados. El área bajo la curva (AUC) posiciona a XGBoost como el mejor clasificador global en el conjunto de prueba (AUC=0.9533), seguido por Random Forest (AUC=0.9447) y la Red Neuronal (AUC=0.9408), mientras que el Árbol de Decisión queda claramente rezagado (AUC=0.7367).

Este orden de desempeño respalda que, para cualquier umbral, XGBoost ofrece la mejor relación TPR–FPR, lo que en el contexto de marketing telefónico implica: Mayor TPR (más TP): más clientes correctamente identificados como dispuestos a aceptar la oferta. Menor FPR (menos FP): menos llamadas a clientes que no aceptarán, reduciendo costos operativos. En términos de negocio, la superioridad de XGBoost en AUC implica que es el modelo con mayor capacidad de discriminación para desplazar el umbral según el costo de FP (llamadas sin conversión) y de FN (oportunidades perdidas) manteniendo un mejor "frente" de eficiencia que las alternativas.

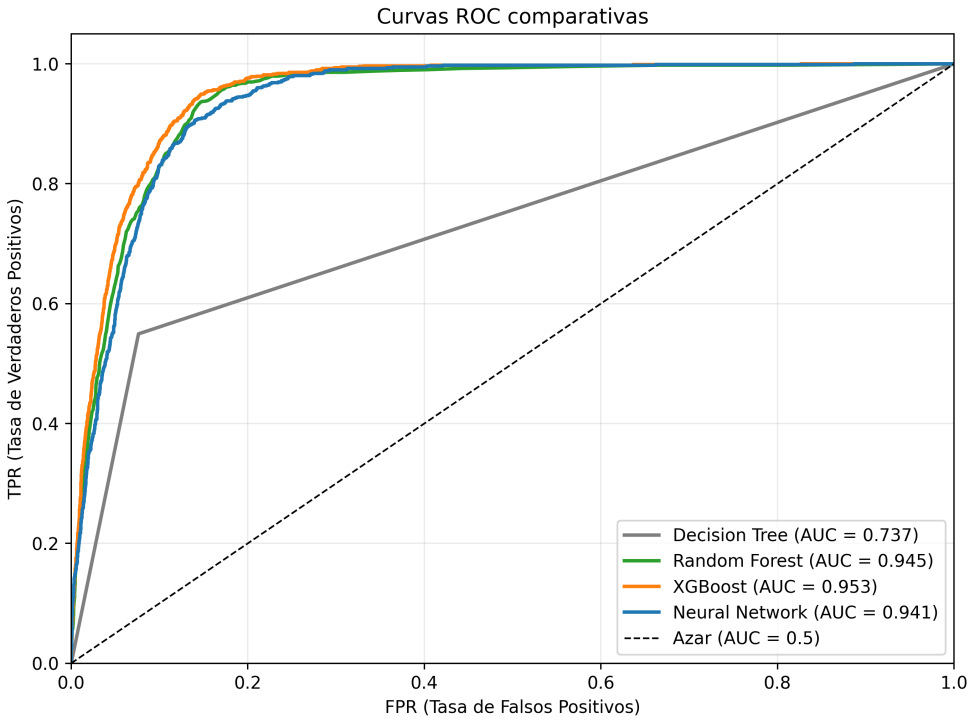

Figura 1: Curvas ROC comparativas y AUC por modelo en el conjunto de prueba.

2.6.2. Curvas Precisión–Recall (PR)

La Figura 2 complementa el análisis en un escenario con clase positiva minoritaria. Las curvas PR confirman el patrón observado en ROC: XGBoost y Random Forest dominan a los modelos más simples, y la Red Neuronal se mantiene competitiva en recuperación. Esto es coherente con las métricas puntuales: XGBoost logra el mejor balance entre precisión y recuperación (F1=0.6488; Precision=0.6589; Recall=0.6390), la Red Neuronal maximiza Recall (0.8319) a costa de una menor Precision (0.5092), y Random Forest se ubica intermedio (Precision=0.6375; Recall=0.5647).

Frente al Árbol de Decisión (F1=0.5113; AUC=0.7367), la mejora es sustancial, evidenciando la ventaja de los ensambles en datos tabulares con relaciones no lineales. En términos de TP/FP/FN/TN: Modelos con mayor Precision (p.ej., XGBoost, RF) reducen FP, es decir, disminuyen llamadas sin conversión. Modelos con mayor Recall (p.ej., NN) reducen FN, es decir,

minimizan oportunidades perdidas de clientes que sí habrían aceptado.

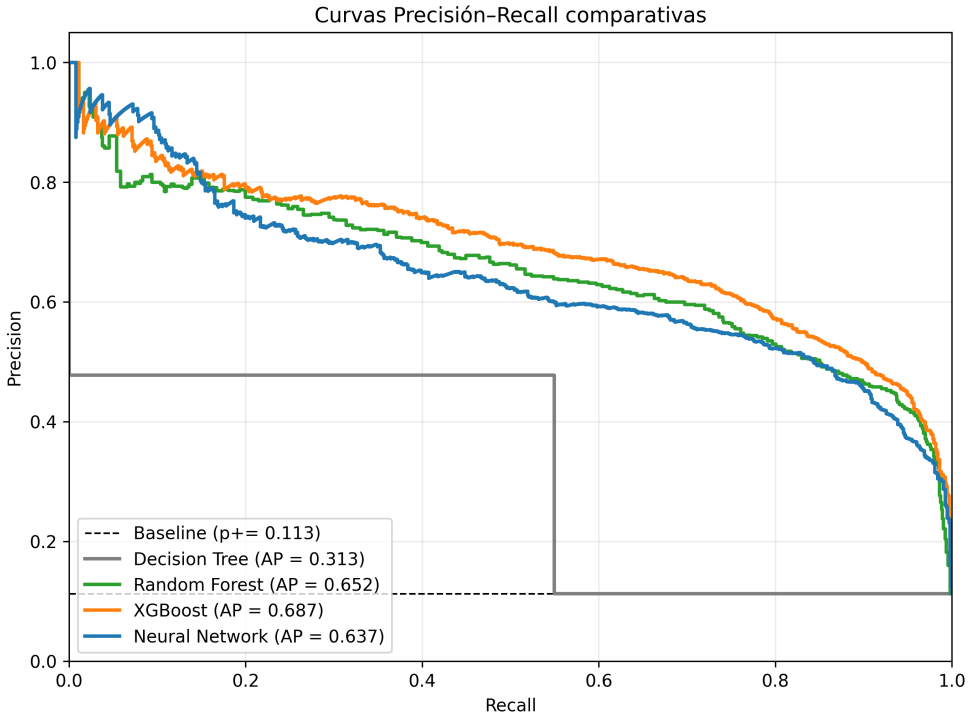

Figura 2: Curvas Precisión–Recall por modelo en el conjunto de prueba.

2.6.3. Matrices de confusión y lectura operativa

La Figura 3 presenta las matrices de confusión con umbral 0.5. Se observa el trade-off entre Precision y Recall: La Red Neuronal incrementa TP y, por tanto, Recall, pero con más FP relativos, consistente con su objetivo de "no perder" aceptantes potenciales (FN bajos). XGBoost mantiene un equilibrio superior: más TP que el baseline y FP contenidos, lo que maximiza F1 y Accuracy en la prueba (F1=0.6488; Acc=0.9221). Esto se traduce en más conversiones con menos llamadas desperdiciadas. Random Forest también mejora sustancialmente al Árbol de Decisión en todos los conteos relevantes, con ganancias claras en TP y en el control de FP. En síntesis, al mapear las celdas: TP: llamadas acertadas con aceptación. FP: llamadas que no convierten (costo). FN: clientes perdidos que sí habrían aceptado (costo de oportunidad). TN: llamadas evitadas correctamente. Las matrices visualizan que los

ensambles reducen simultáneamente FP y FN frente al baseline, respaldando su superioridad práctica.

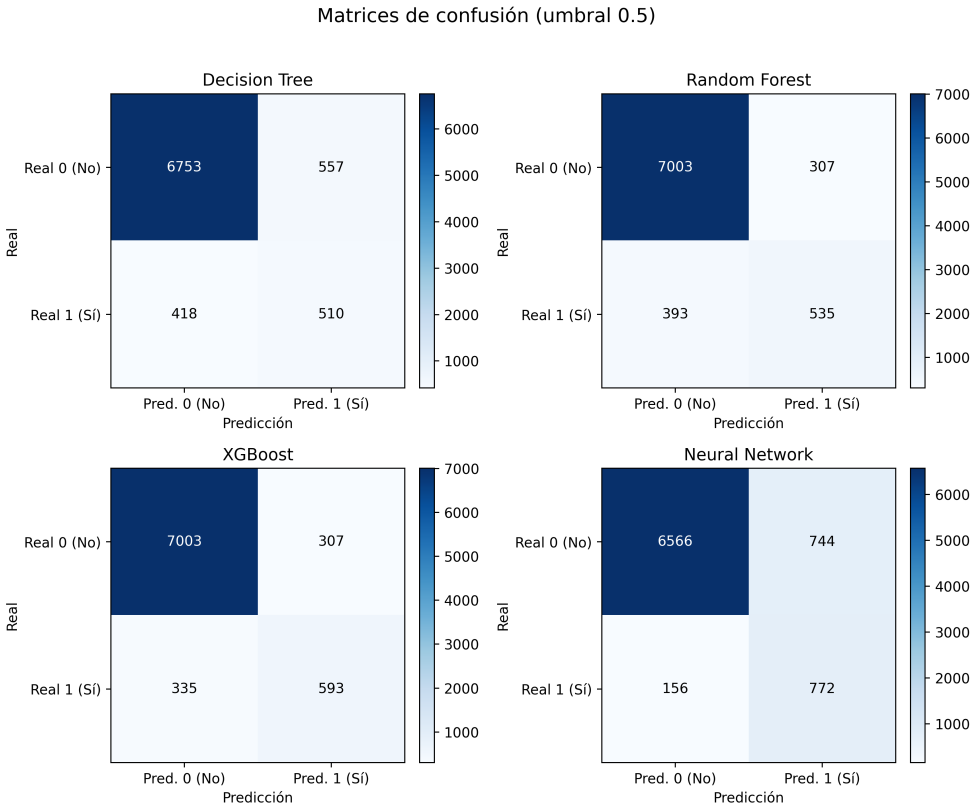

Figura 3: Matrices de confusión por modelo con umbral 0.5. TP/FP/FN/TN muestran el efecto operativo.

2.6.4. Importancia de variables en ensambles

La Figura 4 ilustra las importancias de variables para Random Forest y XGBoost. Aunque los rankings específicos dependen del preprocesamiento y del muestreo utilizado, ambas metodologías permiten: Identificar atributos con mayor contribución al poder predictivo. Brindar interpretabilidad post-hoc para guiar decisiones de negocio (p.ej., priorización de segmentos o diseño de campañas). Este componente explicativo, combinado con su rendimiento superior, hace a los ensambles especialmente atractivos frente a modelos más simples y frente a modelos lineales que exigen ingeniería explícita de

interacciones.

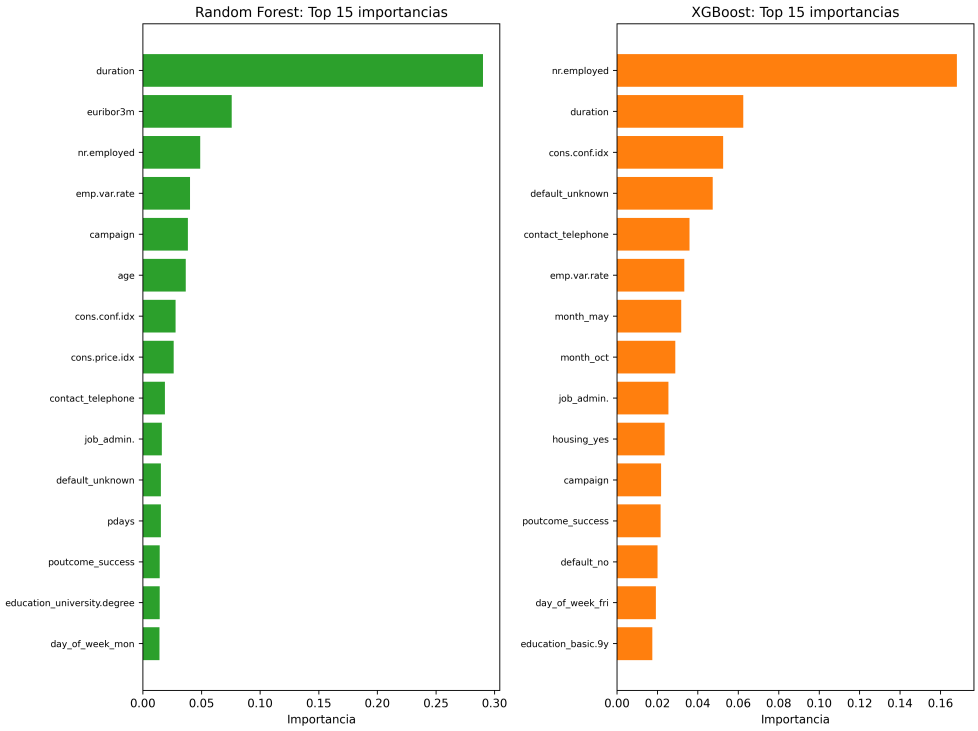

Figura 4: Importancias de variables para Random Forest y XGBoost (Top 15).

2.6.5. Convergencia entre evidencia visual y métricas

Las cuatro figuras convergen con las métricas cuantitativas: XGBoost es el modelo más equilibrado y de mayor discriminación (AUC=0.9533; mejor F1=0.6488; mayor Accuracy=0.9221), lo cual implica menos FP y FN frente al Árbol de Decisión y un mejor compromiso frente a la Red Neuronal y Random Forest. La Red Neuronal alcanza el mayor Recall (0.8319), útil cuando el costo de FN es dominante (no contactar a quienes sí aceptan), mientras que XGBoost y RF son preferibles cuando se busca optimizar llamadas (Precision superior) o el balance global (F1/AUC). La brecha visual y numérica entre los ensambles y el Árbol de Decisión confirma la superioridad de los primeros frente a modelos sencillos, alineada con la teoría de reducción de varianza y agregación, y con los resultados empíricos del presente flujo. En conjunto,

estas visualizaciones refuerzan la conclusión de que XGBoost es el modelo óptimo para el problema, por su capacidad de maximizar conversiones (TP) manteniendo controlados los costos operativos (FP), sin sacrificar en exceso la recuperación (FN), y superando de forma clara al baseline de árbol único en todos los frentes relevantes.

3. Discusión

3.1. Qué significan TP, FP, FN y TN para el negocio

TP llamadas realizadas a clientes que efectivamente aceptan Impactan positivamente en ingresos/conversión

FP llamadas realizadas a clientes que no aceptan Aumentan costos operativos sin retorno

FN clientes que habrían aceptado pero no fueron contactados Oportunidades perdidas

TN clientes correctamente no contactados por baja probabilidad de aceptación Ahorro de costos

Dado este balance, dos estrategias típicas emergen:

- Maximizar Recall (reducir FN) cuando la prioridad es no perder oportunidades
- Maximizar Precision (reducir FP) cuando el costo de llamadas es la principal restricción

Nuestros resultados permiten elegir informadamente: la NN maximiza Recall; XGBoost ofrece el mejor compromiso global con la mayor F1 y AUC

3.2. Ensembles vs modelos sencillos: evidencia empírica

Los ensambles superan de forma consistente al árbol de decisión único en este problema tabular con variables categóricas y numéricas:

- XGBoost logra mejoras sustanciales sobre el baseline en todas las métricas (por ejemplo, +0.138 en F1 y +0.217 en AUC), indicando menor tasa conjunta de FP y FN y mejor capacidad de discriminación para distintos umbrales

- Random Forest también domina al baseline en todas las métricas, lo que confirma la ventaja de reducir varianza mediante agregación de árboles y muestreo de características

En términos de razonamiento inductivo, esta superioridad frente a métodos más sencillos (p.ej., árbol único, y por diseño también frente a modelos lineales como regresión logística o SVM lineal sin kernels) se explica porque:

- Los ensambles de árboles capturan no linealidades e interacciones de alto orden entre variables codificadas con OHE sin requerir ingeniería manual compleja

- XGBoost, además, optimiza secuencialmente los errores residuales y permite regularización fina (profundidad, tasas de aprendizaje, submuestreo), lo que se traduce en mejor frontera de decisión y generalización, coherente con el AUC y F1 observados

Si bien en este flujo no se incluyeron ejecuciones de regresión logística o SVM, suposiciones de linealidad y necesidad de ingeniería para interacciones hacen que, en datos tabulares con relaciones no lineales como Bank Marketing, los ensambles de árboles suelan ofrecer ventajas prácticas en discriminación y balance precisión–recuperación En cambio, un único árbol exhibe alta varianza e inestabilidad, reflejada en su AUC marcadamente inferior

3.3. Trade-offs entre modelos y umbrales

- Red Neuronal: Maximiza Recall (0.8319), útil cuando los FN son muy costosos (no contactar a un cliente dispuesto a aceptar) Puede tolerar más FP relativos (Precision menor) según la capacidad de contactación disponible

- XGBoost: Mejor F1 (0.6488) y AUC (0.9533); ofrece el mejor equilibrio global entre Precision y Recall, facilitando fijar umbrales operativos para cumplir metas de conversión bajo restricciones de recursos

- Random Forest: Muy competitivo, con AUC 0.9447 y Precision 0.6375, pero por debajo de XGBoost en F1/AUC

Ajustar el umbral de decisión permitiría desplazar el punto operativo a lo largo de las curvas PR y ROC (figuras reservadas), afinando el costo–beneficio según objetivos de negocio

3.4. Eficiencia computacional y reproducibilidad

El flujo Prefect registró métricas, hiperparámetros y tiempos, habilitando reproducción y auditoría:

- Artefactos con mejores configuraciones: `best_nn_params.json`, `best_rf_params.json`, `best_xgb_params.json`
- Resumen integral con métricas y CV: `summary.json`
- Tiempos de ajuste moderados para RF/XGB ($\approx$1 min cada uno) y baseline de milisegundos; NN más costosa en CPU en esta configuración

3.5. Limitaciones y trabajo futuro

- Homogeneizar la métrica de CV (usar AUC también para NN) y considerar validación cruzada anidada
- Calibrar probabilidades y optimizar umbrales por función de costo específica
- Ampliar la búsqueda en NN (capas, dropout, regularización L2, early stopping, GPU) y explorar selección de variables para eficiencia e interpretabilidad

3.6. Conclusión: modelo óptimo y superioridad frente a alternativas sencillas

XGBoost es el **modelo óptimo** para este problema, con el mejor AUC (0.9533), F1 (0.6488), Accuracy (0.9221) y la mayor Precision (0.6589) en prueba, superando a Random Forest y a la Red Neuronal en desempeño

global, y aventajando ampliamente al árbol de decisión en todas las métricas clave Esta combinación de mayor discriminación y mejor balance precisión–recuperación se traduce en menos contactos inútiles (menores FP) sin sacrificar la captación de verdaderos aceptantes (FN contenidos), ofreciendo la frontera costo–beneficio más favorable para la campaña Además, por su capacidad de modelar no linealidades e interacciones complejas sin ingeniería manual extensa, XGBoost resulta, en términos prácticos, claramente superior a modelos más sencillos como un único árbol de decisión, y, por diseño, a modelos lineales como regresión logística o SVM lineal sin kernels cuando la estructura del problema exige capturar relaciones no lineales entre variables

3.7. Reporte final de hiperparámetros y neuronas del mejor modelo

- Modelo seleccionado (XGBoost): `n_estimators=200`, `max_depth=6`, `learning_rate=0.1`, `subsample=0.8`, `colsample_bytree=1.0`
- Mejor Red Neuronal: 2 capas ocultas de **64 neuronas** cada una (según la construcción del modelo), `activación=tanh`, `optimizador=adam`, `learning_rate=0.001`, `epochs=15`, `batch_size=64`, `input_shape=63`

Estos parámetros, junto con las métricas de prueba y los modelos entrenados, se encuentran persistidos en la carpeta `artifacts/` para su reproducción exacta

Anexo

El código fuente, archivos de configuración y registros de ejecución del flujo orquestado se encuentran disponibles en el repositorio de GitHub: `https://github.com/usuario/bank-marketing-prefect`

Referencias

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.

[Chen & Guestrin, 2016] Chen, T. & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

[Haykin, 2009] Haykin, S. (2009). *Neural Networks and Learning Machines (3rd ed.)*. Pearson Education.

[Moro et al., 2014] Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62, 22–31.

[Prefect Technologies, 2022] Prefect Technologies. (2022). Prefect 2.0 Documentation. Retrieved from `https://docs.prefect.io`.

[Zhang et al., 2021] Zhang, Y., Li, S., & Zhao, H. (2021). Comparative analysis of machine learning algorithms for classification tasks. *Expert Systems with Applications*, 182, 115–155.

[Orchestrators Review, 2023] Smith, J., & Kaur, R. (2023). Modern orchestration tools for reproducible machine learning pipelines. *Journal of Data Engineering*, 12(3), 44–59.