PROYECTO SEMINARIO PROFESIONAL - BANK MARKETING

JOSE CARLOS ESPAÑA LUTIN - 22000123 | LESTER ALEJANDRO HERNANDEZ OCHAITA - 22003896 |
CRISTIAN ISAAC SACTIC HERRERA - 21001986

Introducción

El propósito de este proyecto fue implementar y comparar tres modelos de inteligencia artificial utilizando el conjunto de datos de Bank Marketing, el cual contiene información sobre campañas de llamadas telefónicas realizadas por un banco en Portugal. La tarea consistía en predecir si un cliente aceptaría o no un depósito a plazo fijo, siendo esta la variable objetivo.
El Dataset esta compuesto por 4 subsdatasets:

- **Bank-adittional-full**: Contiene al rededor de 41188 registros por 20 de entrada.
- **Bank-adittional**: Corresponde a una muestra aleatoria equivalente al 10 % del conjunto anterior.
- **Bank-Full**: Incluye los mismos registros que el anterior, pero con únicamente 17 variables de entrada.
- **Bank**: Es una muestra aleatoria del 10 % del conjunto *Bank-full*.

Código QR

Descripción del Dataset

El dataset es multivariable, con variables numéricas, categóricas y binarias. Estas describen información demográfica, financiera y del historial de contacto entre clientes y la entidad bancaria.

Tipos de variables:

- **Numéricas:** edad, balance, duración, etc.
- **Binarias:** default, housing, loan, etc.
- **Categóricas:** job, marital, education, month, etc.

Variable objetivo: `y`, indica si el cliente aceptó el depósito a plazo (`"yes" / "no"`).

Desafíos encontrados:

- Alta desbalance entre clases: mayoría de respuestas son `"no"`.
- Variables categóricas requieren codificación.
- Valores `unknown` en atributos como `job`, `loan` o `housing`.
- Diferencias de escala en variables numéricas (ej. `balance` vs. `duration`).
- Alta cardinalidad en algunas variables categóricas.

Metodología y Resultados

Idea de Modelado
El enfoque del modelado consistió en preparar y balancear el dataset bancario a través de varias etapas:

- **Carga y limpieza del dataset**
- **Codificación de variables categóricas a numéricas**
- **Normalización** de variables numéricas
- **Oversampling** para balancear la variable objetivo `Y`.

Desafíos encontrados:

- Riesgo de Overfitting al balancear los datos artificialmente
- Normalización de variables numéricas al combinarlos con Oversampling
- Desbalanceo en la variable objetivo (Y) al aplicar técnicas de Oversampling
- Datos faltantes o inconsistentes durante el preprocesamiento de la data
- Tiempos altos de cómputo por técnicas aplicadas sobre el modelo

Resultados
Se entrenaron tres modelos distintos. El Modelo 1 fue una red neuronal multiclase con varias capas densas, utilizando la función de activación **ReLU** en las capas ocultas y **sigmoide** en la capa de salida, adaptada para clasificación multiclase. El Modelo 2, basado en **XGBoost**, empleó árboles de decisión optimizados, siendo eficaz para datos estructurados y clases desbalanceadas.
El Modelo 3 fue una red neuronal binaria que también usó **ReLU** y **sigmoide**, incorporando además técnicas de *Dropout*, *Batch Normalization* y *EarlyStopping* para evitar el sobreajuste. El objetivo fue determinar cuál arquitectura ofrecía el mejor rendimiento sobre el conjunto de datos bancario, obteniendo los siguientes resultados:

Comparativa:

Modelo	Funciones de Activación	Número de Capas
Modelo 1	ReLU, Sigmoide	4
Modelo 2	Árboles de decisión	N/A
Modelo 3	ReLU, Sigmoide	4 (con Dropout y BatchNorm)

Modelo	Accuracy Entrenamiento	Accuracy Validación	Accuracy Prueba	Observaciones
Modelo 1	91.94 %	68.5 %	87 %	Bueno pero algo sobreajustado
Modelo 2	- %	- %	83 %	Mejor equilibrio y precisión
Modelo 3	- %	- %	66.5 %	Mejor desempeño y generalización

Conclusiones y mejoras a futuro

Conclusiones
El Modelo 1 fue el más efectivo al reconocer casos, con una alta precisión en el test (91.9 %), aunque tuvo dificultades en la evaluación (68.5 %). El Modelo 2 (XGBoost) mostró un rendimiento más bajo en general (83 % de precisión), pero presentó problemas al identificar correctamente la clase 1. El Modelo 3 tuvo un rendimiento intermedio (66.5 %), pero también necesitó mejoras para mejorar su generalización. En resumen, el Modelo 1 fue el más preciso.

Mejoras a futuro

- Optimización de hiperparámetros (**GridSearch, RandomSearch**).
- Probar modelos como **LightGBM** o **CatBoost**.
- Ampliar y diversificar el **dataset**.
- Implementar **evaluación cruzada** más rigurosa (k-fold).

Repositorio GitHub:
`https://github.com/LesterHernandez/SP1-AI/tree/main`