**TALLERES: TRABAJO FINAL**

**Clasificación Supervisada con el Dataset de Marketing Bancario**

**1. Breve descripción del problema**

El presente análisis busca resolver un problema de clasificación supervisada utilizando el conjunto de datos del Bank Marketing Dataset de Kaggle. Este dataset proviene de una campaña de marketing directo de una institución financiera y contiene información sobre clientes, sus características personales y el historial de contacto. El objetivo es construir modelos predictivos capaces de determinar si un cliente suscribirá un depósito a plazo ('term deposit') o no, en función de sus características.

La solución a este problema tiene una utilidad práctica significativa para entidades bancarias, ya que permite optimizar campañas de marketing, segmentar mejor a los clientes y reducir costos al enfocar esfuerzos solo en clientes con alta probabilidad de conversión. Para lograr este objetivo, se probaron distintos algoritmos de clasificación y se evaluaron sus desempeños mediante métricas estándar.

**2. Descripción del dataset**

Nombre del dataset: Bank Marketing Dataset (Fuente: Kaggle)

Cantidad de registros: 11,162

Tipo de problema: Clasificación binaria

Variable objetivo: deposit (valores originales “yes” y “no”, convertidos a 1 y 0)

Descripción de algunas variables principales:

age: Edad del cliente (numérica).

job: Tipo de ocupación del cliente (categórica).

marital: Estado civil del cliente (categórica).

education: Nivel educativo (categórica).

default: Si tiene créditos en default (binaria: yes/no).

balance: Saldo promedio anual en la cuenta (numérica).

housing: Si tiene préstamo hipotecario (binaria: yes/no).

loan: Si tiene otro tipo de préstamo (binaria: yes/no).

contact: Tipo de contacto (categórica).

day, month: Día y mes del último contacto (categóricas).

duration: Duración del último contacto en segundos (numérica).

campaign: Número de contactos durante esta campaña (numérica).

pdays, previous, poutcome: Información de campañas anteriores.

deposit: Variable objetivo que indica si el cliente contrató un depósito a plazo (yes/no).  
  
**3. Algoritmos usados**

Antes de entrenar los modelos, se realizó un preprocesamiento riguroso y ordenado. Primero, se verificó la presencia de valores nulos en el dataset; al no encontrarse valores faltantes, se continuó con la transformación de la variable objetivo deposit, convirtiéndola en una variable binaria donde 1 representa que el cliente contrató un depósito a plazo y 0 lo contrario. Luego, se aplicó la técnica de codificación OneHotEncoding a las variables categóricas, permitiendo su uso en algoritmos de clasificación. Finalmente, las variables numéricas fueron escaladas utilizando StandardScaler para normalizar su rango y mejorar el rendimiento de los modelos.

Se implementaron los siguientes modelos de clasificación:

1. Regresión Logística.

2. Árbol de Decisión.

3. Random Forest.

4. XGBoost.

Todos fueron entrenados con un 80% de los datos y evaluados sobre el 20% restante.

**4. Métricas y análisis de resultados**

Las métricas utilizadas fueron: precisión, recall, f1-score y matriz de confusión.

|  |  |  |  |
| --- | --- | --- | --- |
| Modelo | Precisión | Recall | F1-score |
| Regresión Logística | 0.808 | 0.78 | 0.80 |
| Árbol de Decisión | 0.783 | 0.76 | 0.77 |
| Random Forest | 0.841 | 0.87 | 0.84 |
| XGBoost | 0.844 | 0.86 | 0.84 |

El modelo XGBoost fue el que obtuvo el mejor rendimiento general, con una precisión de 0.844 y un recall de 0.86, lo que indica que no solo acierta en la mayoría de sus predicciones positivas, sino que también identifica correctamente a una gran parte de los clientes que efectivamente realizaron una compra. Esto lo convierte en una excelente opción para el problema planteado.

Muy cerca le siguió el modelo de Random Forest, que también mostró métricas sólidas y equilibradas. En cambio, Regresión Logística y Árbol de Decisión tuvieron un desempeño aceptable, pero con valores más bajos, especialmente en recall, lo que implica que dejaron pasar más casos positivos sin detectar.

En resumen, los modelos de ensamble como XGBoost y Random Forest se destacaron por su mayor capacidad predictiva, siendo más efectivos para identificar correctamente a los clientes que se suscribirán a un depósito a plazo, lo cual es clave para orientar mejor las estrategias de marketing.

**5. Instrucciones para correr el modelo localmente:**

Para ejecutar este proyecto en un entorno local (por ejemplo, Google Colab), se recomienda seguir los siguientes pasos:

1. Subir el archivo del dataset a Colab: Asegúrate de tener el archivo CSV del dataset. Puedes subirlo directamente al entorno de Colab con el botón “Subir archivo”.

2. Instalar las librerías necesarias: Si no están previamente instaladas, utiliza los siguientes comandos:

!pip install xgboost sweetviz scikit-learn

3. Importar las librerías necesarias para análisis, preprocesamiento y modelado:

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from xgboost import XGBClassifier

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

4. Leer y explorar los datos:

data = pd.read\_csv('nombre\_del\_archivo.csv')

data.head()

5. Preprocesar los datos:

- Verificar y tratar valores nulos.

- Convertir la variable objetivo (`deposit`) a formato binario (1/0).

- Aplicar codificación `OneHotEncoding` a las variables categóricas.

- Escalar las variables numéricas con `StandardScaler`.

6. Separar variables predictoras (X) y objetivo (y), y luego dividir los datos en entrenamiento y prueba.

7. Entrenar y evaluar los modelos: Regresión Logística, Árbol de Decisión, Random Forest y XGBoost. Usar métricas como precisión, matriz de confusión y reporte de clasificación.

**Se adjunta notebook (código fuente) para replicar todo el proceso.**