Bank Marketing Dataset Analysis Report:

1.Introduction

The Bank Marketing dataset comes from a Portuguese bank's direct marketing campaigns. These campaigns involved contacting clients (mostly by phone) to promote **term deposits**.

- **Objective**: Predict whether a client will subscribe to a term deposit (`yes` or `no`).
- **Type of Problem**: Binary classification.
- **Algorithm Used**: Logistic Regression.

2. Dataset Overview

- **File Used**: `bank-full.csv`
- **Records**: 45,211 rows
- **Features**: 16 input variables + 1 target (`y`)
- **Target Variable**:
  - `y = yes` → client subscribed to term deposit
  - `y = no` → client did not subscribe

Features:

1. **Client Attributes**: `age`, `job`, `marital`, `education`, `default`, `housing`, `loan`
2. **Current Campaign**: `contact`, `month`, `day_of_week`, `duration`, `campaign`
3. **Previous Campaigns**: `pdays`, `previous`, `poutcome`
4. **Economic Indicators**: `emp.var.rate`, `cons.price.idx`, `cons.conf.idx`, `euribor3m`, `nr.employed`

3. Exploratory Data Analysis (EDA)

3.1 Target Distribution

- **No**: ~89%
- **Yes**: ~11%
  The dataset is **highly imbalanced**.

3.2 Numerical Features

- `age`: Most clients between 30–40 years old.
- `duration`: Strongly linked with outcome (longer calls often lead to "Yes").
- `campaign`: Most clients contacted fewer than 5 times.
- `balance`: Skewed distribution with some very high values.

3.3 Categorical Features

- **Job**: Common jobs include admin, blue-collar, and technician.

- **Marital**: Majority are married.
- **Education**: Most have secondary education.
- **Housing Loans**: Many clients have housing loans.

3.4 Correlation

- `duration` shows the strongest correlation with subscription.
- Other features (`pdays`, `previous`, `euribor3m`) also show some influence.

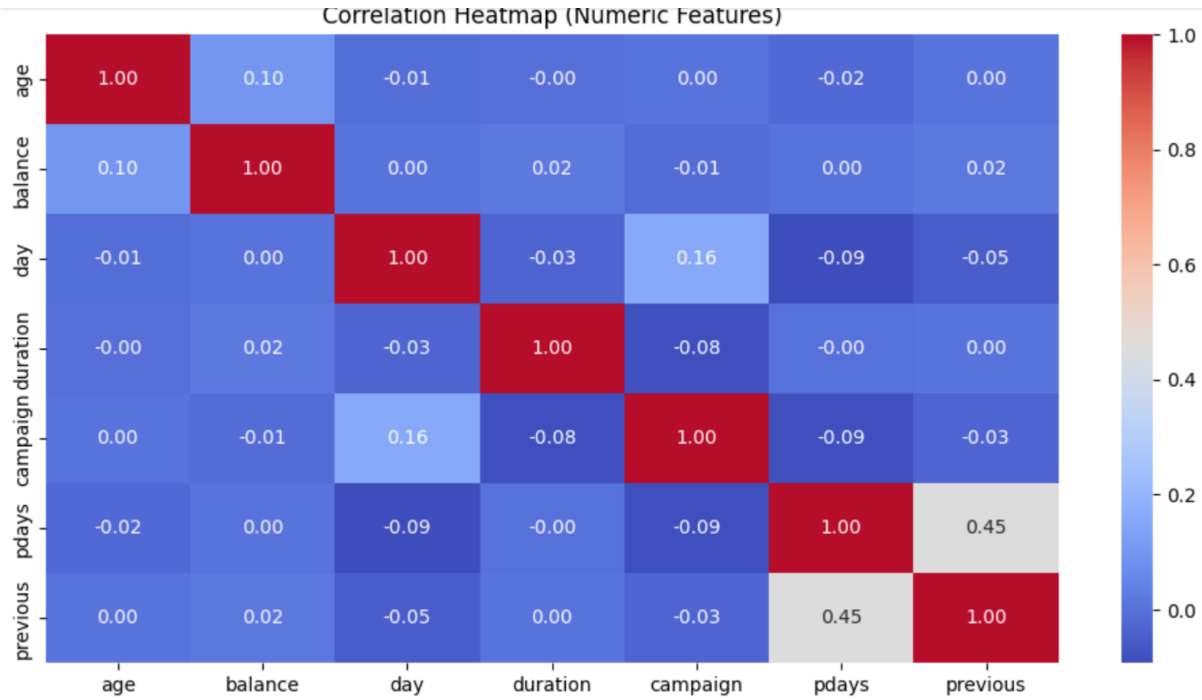


Fig: Correlation Heatmap

4. Data Preprocessing

- **Encoding**: All categorical variables converted into numeric form using Label Encoding.
- **Splitting**: 70% training data, 30% testing data.
- **Scaling**: Not applied (logistic regression with categorical encoding doesn't strictly require it).

5. Model Training

- **Model**: Logistic Regression
- **Hyperparameters**:
  - `solver = liblinear`
  - `max_iter = 500`
- **Training**: Model fitted on training dataset.

```
""" # Import Libraries """

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

"""# Load and Exploration of Dataset"""

df = pd.read_csv("bank-full.csv", sep=';')
print("Dataset Shape:", df.shape)
print("\nFirst 5 rows:\n", df.head())
print("\nColumn Info:\n")
print(df.info())
print("\nMissing Values:\n", df.isnull().sum())
print("\nTarget value counts:\n", df['y'].value_counts())
sns.countplot(x="y", data=df)
plt.title("Target Distribution (Subscribed: Yes/No)")
plt.show()

print("\nStatistical Summary:\n", df.describe(include='all'))
print("\nUnique Values per Column:\n")
for col in df.columns:
    print(f"{col}: {df[col].nunique()} unique values")
plt.figure(figsize=(12, 6))
sns.heatmap(df.select_dtypes(include=np.number).corr(), annot=True,
cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap (Numeric Features)")
plt.show()

df.select_dtypes(include=np.number).hist(bins=20, figsize=(15, 10),
edgecolor="black")
plt.suptitle("Numeric Feature Distributions")
plt.show()
categorical_cols = df.select_dtypes(include='object').columns
for col in categorical_cols:
    plt.figure(figsize=(8, 4))
    sns.countplot(data=df, x=col, order=df[col].value_counts().index)
    plt.title(f"Distribution of {col}")
    plt.xticks(rotation=45)
    plt.show()

for col in categorical_cols:
```

```
 plt.figure(figsize=(8, 4))
    sns.countplot(data=df, x=col, hue="y", order=df[col].value_counts().index)
    plt.title(f"{col} vs Subscription (Target)")
    plt.xticks(rotation=45)
    plt.show()
print("\nTarget Variable Distribution (with percentages):")
print(df['y'].value_counts(normalize=True) * 100)

"""# Encoding and Training of Logistic Model"""

df_encoded = df.copy()
for col in df_encoded.select_dtypes(include=['object']).columns:
    df_encoded[col] = LabelEncoder().fit_transform(df_encoded[col])
X = df_encoded.drop("y", axis=1)
y = df_encoded["y"]
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
)
log_reg = LogisticRegression(max_iter=500, solver='liblinear')
log_reg.fit(X_train, y_train)

"""# Model Evaluation"""

y_pred = log_reg.predict(X_test)

print("\nClassification Report:\n", classification_report(y_test, y_pred))

cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=["No", "Yes"],
            yticklabels=["No", "Yes"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrices")
plt.show()
```

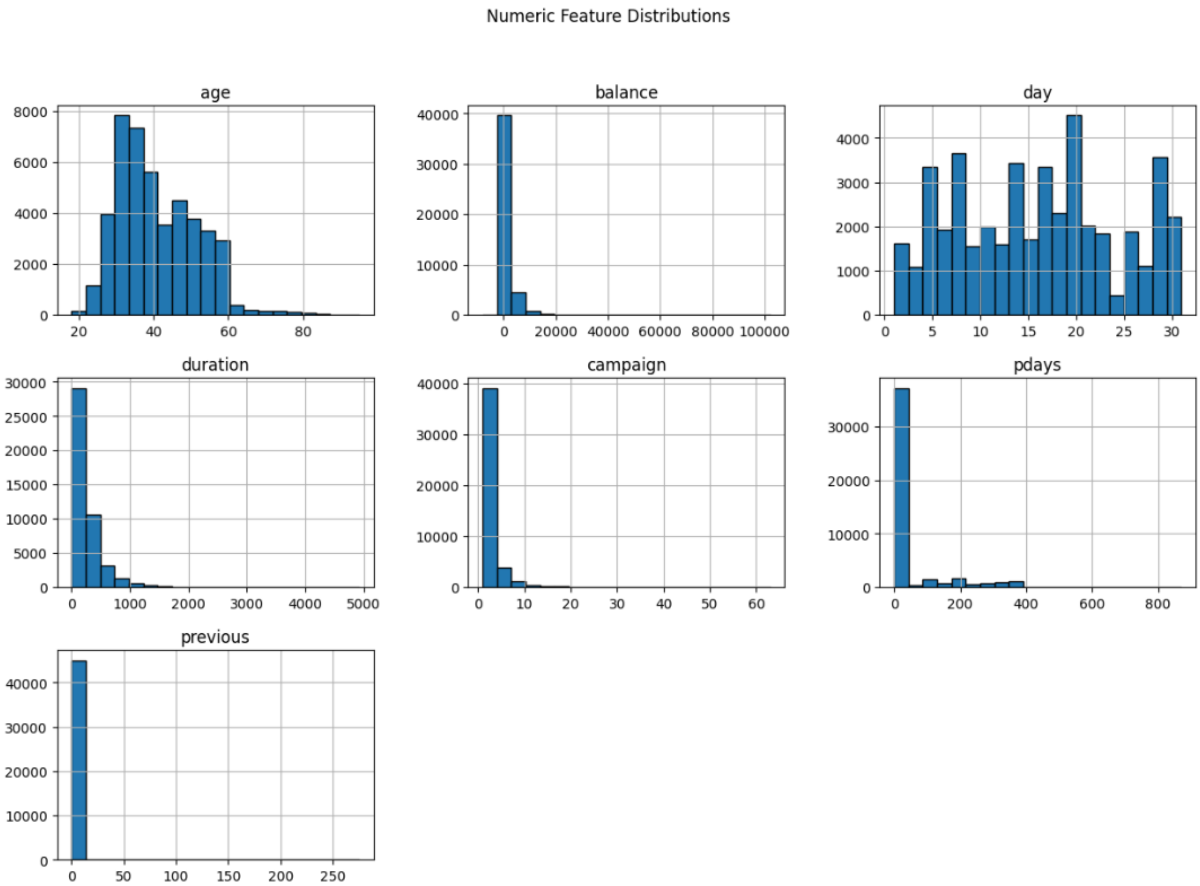
6.Code

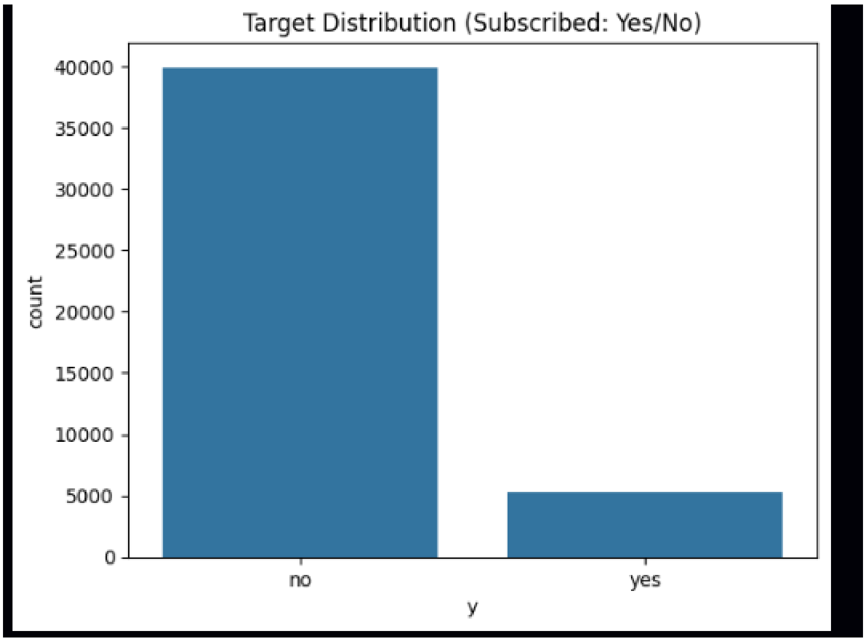
7. Output --

```
Dataset Shape: (45211, 17)

First 5 rows:
   age           job  marital  education default  balance housing loan
0   58    management  married   tertiary      no     2143     yes   no
1   44    technician   single  secondary      no       29     yes   no
2   33  entrepreneur  married  secondary      no        2     yes  yes
3   47   blue-collar  married    unknown      no     1506     yes   no
4   33       unknown   single    unknown      no        1      no   no

   contact  day month  duration  campaign  pdays  previous poutcome   y
0  unknown    5   may       261         1     -1         0  unknown  no
1  unknown    5   may       151         1     -1         0  unknown  no
2  unknown    5   may        76         1     -1         0  unknown  no
3  unknown    5   may        92         1     -1         0  unknown  no
4  unknown    5   may       198         1     -1         0  unknown  no

Column Info:
```

```
Column Info:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        45211 non-null  int64
...
 y
no     39922
yes     5289
Name: count, dtype: int64
```

```
Statistical Summary:
                 age          job  marital  education default        balance  \
count   45211.000000        45211    45211      45211   45211   45211.000000
unique           NaN           12        3          4       2            NaN
top              NaN  blue-collar  married  secondary      no            NaN
freq             NaN         9732    27214      23202   44396            NaN
mean       40.936210          NaN      NaN        NaN     NaN    1362.272058
std        10.618762          NaN      NaN        NaN     NaN    3044.765829
min        18.000000          NaN      NaN        NaN     NaN   -8019.000000
25%        33.000000          NaN      NaN        NaN     NaN      72.000000
50%        39.000000          NaN      NaN        NaN     NaN     448.000000
75%        48.000000          NaN      NaN        NaN     NaN    1428.000000
max        95.000000          NaN      NaN        NaN     NaN  102127.000000

       housing  loan   contact           day  month      duration  \
count    45211  45211     45211  45211.000000  45211  45211.000000
unique       2      2         3           NaN     12           NaN
top        yes     no  cellular           NaN    may           NaN
freq     25130  37967     29285           NaN  13766           NaN
mean       NaN    NaN       NaN     15.806419    NaN    258.163080
std        NaN    NaN       NaN      8.322476    NaN    257.527812
min        NaN    NaN       NaN      1.000000    NaN      0.000000
25%        NaN    NaN       NaN      8.000000    NaN    103.000000
50%        NaN    NaN       NaN     16.000000    NaN    180.000000
...
pdays: 559 unique values
previous: 41 unique values
poutcome: 4 unique values
y: 2 unique values
```





8. Model Evaluation

8.1 Classification Report (Test Data)

	precision	recall	f1-score	support
0	0.90	0.98	0.94	11977
1	0.60	0.21	0.31	1587
accuracy			0.89	13564
macro avg	0.75	0.60	0.63	13564
weighted avg	0.87	0.89	0.87	13564

8.2 Confusion Matrix (Interpretation)

- **True Negatives (TN)**: Very high → Model correctly identifies most `No`.
- **True Positives (TP)**: Very low → Model misses many `Yes`.
- **False Negatives (FN)**: High → Many actual subscribers are predicted as `No`.

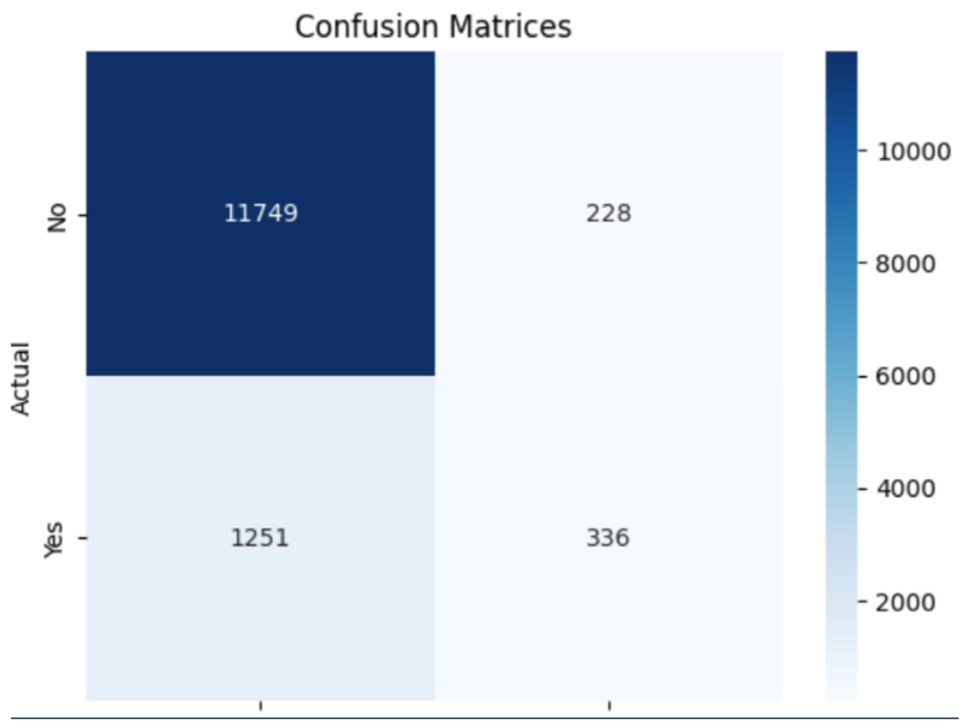


Fig: Confusion Matrices

9. Results

- The model achieves **89% accuracy** overall.
- Very strong in predicting **non-subscribers (class 0)**:
  - Precision = 0.90, Recall = 0.98

- Weak in predicting **subscribers (class 1)**:
  - Recall = 0.21 → Model captures only ~21% of actual subscribers.
  - F1-score = 0.31 → Poor performance for minority class.
- This happens due to **class imbalance** (only 11% "Yes").

10. Observation

- Logistic Regression achieved **high accuracy (89%)** but **performed poorly on predicting actual subscribers (Yes)**.
- For marketing, missing potential subscribers is costly, so future improvements should focus on increasing **Recall for the minority class**.
- This baseline model highlights the importance of handling imbalanced datasets in real-world classification problems.