

GitHub Info	
GitHub Repository Link containing	https://github.com/nskabra-git/ML_Assignment2_BankMarketing
Complete source code	https://github.com/nskabra-git/ML_Assignment2_BankMarketing/tree/main/model
requirements.txt	https://github.com/nskabra-git/ML_Assignment2_BankMarketing/blob/main/requirements.txt
A clear README.md	https://github.com/nskabra-git/ML_Assignment2_BankMarketing/blob/main/README.md

Streamlit App Info	
Live Streamlit App Link	https://2025aa05719-nitin-shriram-kabra-ml-assignment2.streamlit.app/

Screen Shot of BITS Virtual Lab Usage

The screenshot shows a Linux desktop environment with a dark theme. On the left, there's a dock with icons for Firefox, anaconda_projects, Visual Studio Code, Home, and Trash. The main screen has a large watermark of the "INSTITUTE OF TECHNOLOGY & SCIENCE, DABANGAL" logo. A Streamlit application window titled "New feedback for assignment" is open, displaying a dashboard with various metrics. Below it, a Firefox browser window shows the GitHub repository "nskabra-git / ML_Assignment2_BankMarketing". The GitHub page lists the repository details, including the last commit by "nskabra-git" and a brief description: "Bank Marketing using different models".

Screen Shot of BITS Virtual Lab Usage

Fork ⚙️ ⋮

Bank Marketing Classification App

User Controls Default Model - Evaluation Metrics

Select Model
Logistic Regression

Download Sample Test Dataset

Upload Test Dataset

Upload Test Dataset (CSV)
Drag and drop file here
Limit 200MB per file • CSV
Browse files

Evaluation Metrics

	Value
Accuracy	0.846
Precision	0.419
Recall	0.814
F1 Score	0.553
MCC	0.509
AUC	0.908

Confusion Matrix

		No Subscription	Subscribed
Actual	No Subscription	6789	1196
	Subscribed	197	861

Technology & Science, Pilani

परमं बलम्



README.md

ML Assignment 2 – Bank Marketing Classification

Problem Statement

The objective of this project is to build and compare multiple machine learning models to predict whether a client will subscribe to a term deposit based on marketing campaign data.

Target variable:

$y = 1 \rightarrow$ Client subscribed to term deposit

$y = 0 \rightarrow$ Client did not subscribe

Dataset



Bank Marketing

Donated on 2/13/2012

The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe a term deposit (variable y).

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Business	Classification
Feature Type	# Instances	# Features
Categorical, Integer	45211	16

- Target: Binary classification (yes / no)
- Class imbalance:
 - No ≈ 88%
 - Yes ≈ 12%
- Source: <https://archive.ics.uci.edu/dataset/222/bank+marketing>

Due to class imbalance, multiple evaluation metrics were used instead of relying only on accuracy.

Project Structure

```
ML_Assignment2_BankMarketing/
├── streamlit_app.py
├── requirements.txt
├── README.md
├── assets/
│   └── bank_marketing_dataset.png
├── data/
│   └── bank-full.csv
├── notebooks/
│   └── exploration.ipynb
└── model/
    ├── preprocessing.py
    ├── evaluate.py
    ├── train_models.py
    ├── logistic_regression_model.pkl
    ├── decision_tree_model.pkl
    ├── knn_model.pkl
    ├── naive_bayes_model.pkl
    ├── random_forest_model.pkl
    ├── xgboost_model.pkl
    ├── sample_test_data.csv
    └── sample_test_with_target.csv
```

Models Implemented

Six classification models were implemented and compared:

1. Logistic Regression
2. Decision Tree
3. K-Nearest Neighbors (KNN)
4. Naive Bayes
5. Random Forest
6. XGBoost

All models were built using a shared preprocessing pipeline.

- One-hot encoding for categorical variables
- Standard scaling for numerical variables
- Stratified train-test split (80/20)

Evaluation Metrics

The following metrics were used:

- Accuracy
- Precision
- Recall
- F1 Score
- AUC (ROC)
- Matthews Correlation Coefficient (MCC)

Accuracy alone was not sufficient due to class imbalance.

Model Comparison (Test Set)

Model	Accuracy	AUC	F1 Score	MCC
Logistic Regression	0.846	0.908	0.553	0.509
Decision Tree	0.878	0.713	0.488	0.419
KNN	0.896	0.837	0.444	0.407
Naive Bayes	0.864	0.809	0.456	0.380
Random Forest	0.904	0.927	0.486	0.456
XGBoost	0.908	0.929	0.561	0.515

Key Observations

- **XGBoost** achieved the best overall performance across Accuracy, AUC, F1 Score, and MCC.
 - **Logistic Regression** achieved the highest Recall, making it suitable when minimizing false negatives is important.
 - Tree-based ensemble methods (Random Forest, XGBoost) handled class imbalance better than simple models.
 - Accuracy alone would have been misleading due to the 88/12 class distribution.
-

Model-wise Performance Observations

ML Model Name	Observation about model performance
Logistic Regression	Achieved strong AUC (0.908) and the highest Recall (~0.81), meaning it correctly identifies most subscribing clients. However, Precision is moderate, indicating more false positives. Suitable when minimizing missed potential subscribers is important.
Decision Tree	Shows improved Accuracy compared to Logistic Regression but relatively low AUC (0.713), indicating weaker ranking ability. Tends to overfit and does not generalize as well as ensemble methods.
KNN	Achieves high Accuracy but relatively low Recall, meaning it misses many actual subscribers. Performance is sensitive to class imbalance and feature scaling. Not ideal for this dataset.
Naive Bayes	Provides balanced but moderate performance across metrics. Assumes feature independence, which may not hold in this dataset, leading to limited predictive strength compared to ensemble models.
Random Forest (Ensemble)	Improves Accuracy and AUC significantly compared to single Decision Tree. Handles non-linear relationships well. However, Recall is lower than Logistic Regression, meaning it misses some subscribers.
XGBoost (Ensemble)	Best overall performer with highest Accuracy (0.908), AUC (0.929), F1 Score, and MCC. Provides the best balance between Precision and Recall. Most robust model for this dataset.

Streamlit Application

The deployed application allows:

- Model selection via dropdown
- Downloading a sample test dataset
- Uploading a custom test dataset (CSV)
- Viewing:
 - Evaluation metrics
 - Confusion matrix (heatmap)
 - Tabulated classification report
 - Downloadable predictions

Live App:

<https://2025aa05719-nitin-shriram-kabra-ml-assignment2.streamlit.app/>

Installation (Local Run)

`pip install -r requirements.txt`

`python -m streamlit run streamlit_app.py`