Predicting Customer Response to Bank Marketing Campaigns

Machine Learning Assignment – 2

1) **Problem Statement**

Banks conduct telemarketing campaigns to promote term deposit subscriptions. Contacting every customer is expensive and inefficient. Therefore, predicting which customers are most likely to subscribe helps improve marketing efficiency and reduce costs.

The objective of this project is to build multiple machine learning classification models that predict whether a customer will subscribe to a term deposit based on demographic and campaign-related information. The trained models are then deployed using a Streamlit web application.

2) **Dataset Description *(1 Mark)***

Dataset: Bank Marketing Dataset (UCI Machine Learning Repository)

The dataset contains information about customers contacted during a bank marketing campaign.

Target Variable

- **y** → Did the customer subscribe to a term deposit?
  - yes = 1
  - no = 0

Dataset Characteristics

Property	**Value**
Number of Instances	45,211
Number of Features	17
Problem Type	Binary Classification
Data Types	Numerical + Categorical

Example Features

- Age
- Job
- Marital status
- Education
- Balance

- Housing loan
- Personal loan
- Contact type
- Campaign duration
- Previous campaign outcome

This dataset satisfies the assignment requirement of having more than 500 instances and at least 12 features.

3) Models Used & Evaluation Metrics *(6 Marks)*

The following classification models were implemented on the same dataset:

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

Evaluation Metrics

Each model was evaluated using:

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

Model Comparison Table

Model	Accuracy	AUC	Precision	Recall	F1	MCC
Logistic Regression	0.90	0.93	0.64	0.38	0.48	0.45

Model	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1**	**MCC**
Decision Tree	0.87	0.83	0.49	0.52	0.50	0.43
KNN	0.89	0.88	0.57	0.39	0.46	0.42
Naive Bayes	0.86	0.86	0.45	0.60	0.51	0.44
Random Forest	0.91	0.95	0.70	0.48	0.57	0.54
XGBoost	0.92	0.96	0.74	0.50	0.60	0.57

(Values may slightly vary after retraining)

4) **Model Performance Observations *(3 Marks)***

Model	**Observation**
Logistic Regression	Performs well as a baseline model but struggles with complex non-linear relationships.
Decision Tree	Captures non-linear patterns but prone to overfitting, resulting in slightly lower generalization performance.
KNN	Provides moderate performance but is sensitive to scaling and large dataset size.
Naive Bayes	Achieves good recall but lower precision due to strong feature independence assumption.
Random Forest	Performs significantly better by reducing overfitting and capturing complex patterns.
XGBoost	Best performing model with highest AUC and overall balanced performance across metrics.

5) **Streamlit Web Application**

The deployed Streamlit application provides:

- Dataset upload option (CSV)
- Model selection dropdown
- Display of evaluation metrics
- Confusion matrix and classification report

This demonstrates a complete end-to-end machine learning deployment workflow.

6) **Project Structure**

```
project-folder/

│— app.py

│— train_models.py

│— requirements.txt

│— README.md

│— model/

│— data/
```

7) **Conclusion**

This project demonstrates the complete machine learning pipeline from data preprocessing and model training to deployment. Ensemble methods such as Random Forest and XGBoost achieved the best performance, highlighting the effectiveness of ensemble learning for real-world classification problems.