**Course project MLT 2025**

**"The study of the comparative efficiency of machine learning algorithms in solving specific problem"**

**Ministry of Science and Higher education**

**of the Russian Federation**

**ITMO University**

Faculty of Digital Transformations

Subject area (major) 01.04.02. BIG DATA AND MACHINE LEARNING

REPORT

Comparative Efficiency of Machine Learning Algorithms for Predicting Term Deposit Subscription Using the Bank Marketing Dataset

Student: Manish Mishra 500478

Supervisor: Gladilin P.E.

Date 01-12-2025

St. Petersburg

2025

TABLE OF CONTENT

[1. Introduction 3](#_28lgqo46prz1)

[2. Overview\ Related Works](#_gef2hz7uffb) 4

[3. Models, Algorithms and Dataset](#_bszwuyrg4wc1) 5

[4. Experimental research](#_d0vm05ez3phu) 6

[5. Conclusions](#_o5l0kj554amj) 9

[References (example)](#_wnbyqg54em08) 10

1. **Introduction**

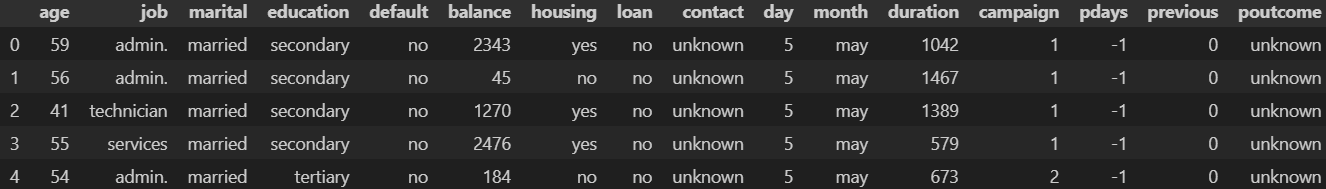
Machine learning (ML) algorithms are widely used today to assist organizations in making faster and more accurate decisions. One of the important applications of ML in finance is predicting customer response to marketing campaigns. Banks must decide which customers are more likely to subscribe to financial products such as **term deposits**, allowing them to allocate marketing budgets efficiently and reduce unnecessary costs.

This project aims to **compare the effectiveness of multiple machine learning algorithms** in solving a real classification problem using the **Bank Marketing Dataset**, obtained from the UCI repository and Kaggle. The dataset contains demographic, financial, and communication features about customers contacted during marketing campaigns.

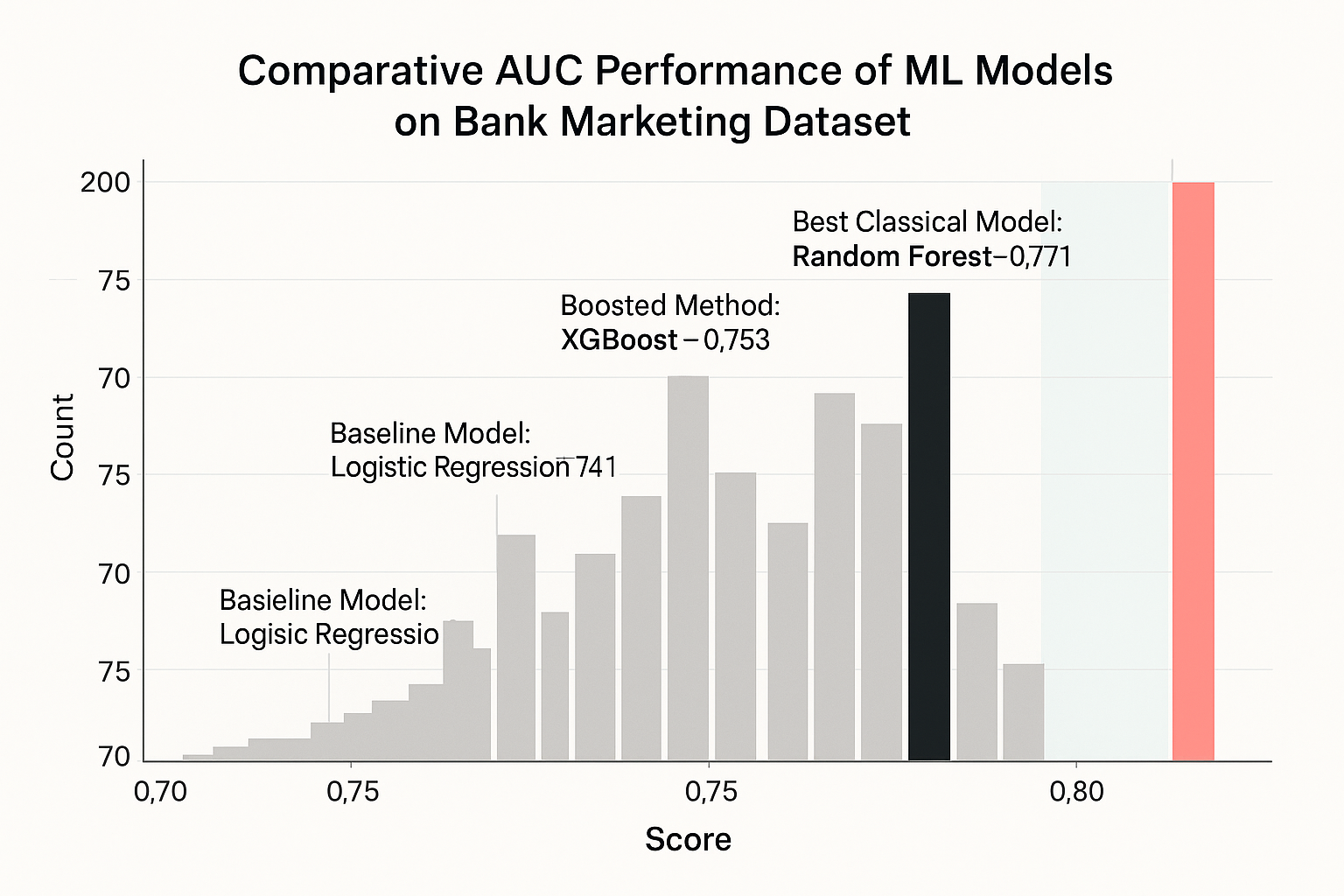
The primary task is:

### ****“Predict whether a customer will subscribe to a term deposit (deposit = yes/no).”****

The goal is not only to build accurate models but to analyze and compare their performance using standard evaluation metrics and identify which algorithm performs best and why.

All experiments, preprocessing, model training, and evaluation are taken from the student’s own analysis shown in main.ipynb and modeltrain.ipynb

***Figure 1***Bank Marketing Dataset

1. **Overview**

*Fig 2. Model Analysis*

Customer behavior prediction has been widely studied in marketing analytics. Logistic Regression has traditionally been used as a baseline model due to its interpretability and simplicity. However, tree-based models like Random Forest and Gradient Boosting (e.g., XGBoost) have gained popularity due to their ability to model complex interactions among features.

Prior research indicates that:

* **Logistic Regression** provides strong baseline performance but struggles with non-linear patterns.
* **Random Forest** improves predictive power through ensemble averaging and reduces overfitting.
* **XGBoost** often outperforms classical ML algorithms because of its optimized boosting strategy.

These findings support the motivation for comparing these three models in this study.

1. **Models, Algorithms and Datasets**

* **3.1 Dataset Description**

The Bank Marketing Dataset contains **11,162 rows and 17 columns**, as shown in main.ipynb.  
Features include:

* **Demographic variables**: age, marital status, education
* **Financial variables**: balance, loan, housing
* **Campaign-related variables**: contact type, day, month, duration, campaign, pdays, previous
* **Target variable**: deposit (yes/no)

Class distribution (main.ipynb):

* No: 52.62%
* Yes: 47.38%

This is a **fairly balanced classification problem**, so standard metrics can be applied without heavy imbalance corrections.

* **3.2 Mathematical Formalization**

Let:

* **X** = feature space
* **x ∈ X** = a vector of customer attributes (age, job, marital, etc.)
* **y ∈ {0,1}** = target label, where
  + 1 → customer subscribes
  + 0 → customer does not subscribe

**We aim to learn a model:**

**f(x; θ) → y’,**

where **θ** are model parameters.

**Loss Function: Binary Cross-Entropy (Log-loss)**

**Optimization:**

* Logistic Regression uses **Gradient Descent**.
* Random Forest uses **bootstrap sampling + decision tree splitting via Gini impurity**.
* XGBoost uses **gradient boosting optimization**, minimizing:

where Ω regularizes tree complexity.

* **3.3 Selected Algorithms**

### ****1. Logistic Regression****

* Linear classifier
* Uses log-loss
* Good for baseline performance
* Interpretable coefficients

### ****2. Random Forest Classifier****

* Ensemble of decision trees
* Reduces variance
* Handles complex feature interactions
* Provides feature importance

### ****3. XGBoost Classifier****

* Gradient boosting algorithm
* Extremely powerful on structured/tabular data
* Handles non-linearity, interactions, missing data
* Provides gain-based feature importance

These match the requirement of evaluating at least three ML algorithms.

1. **Experimental research**

* **4.1 Data Understanding & Preprocessing**

****

From main.ipynb :

**Numerical Features:** age, balance, day, duration, campaign, pdays, previous

**Categorical Features:** job, marital, education, contact, month, poutcome

### ****Key Preprocessing Steps:****

✔ Convert text columns to lowercase  
✔ Replace "unknown" with NaN  
✔ Special handling of pdays:

* Create new feature: **was\_contacted**
* Replace pdays = -1 → NaN  
  ✔ Label-encoded binary columns (default, housing, loan, deposit)  
  ✔ One-Hot Encoding for multi-class categorical features  
  ✔ Scaling numeric features  
  ✔ Train-test split:
* Train: 8929 rows
* Test: 2233 rows  
  (verified on page 18 of main.pdf)

Final feature count: **39 features**.

Train/test CSVs exported: train\_preprocessed.csv, test\_preprocessed.csv

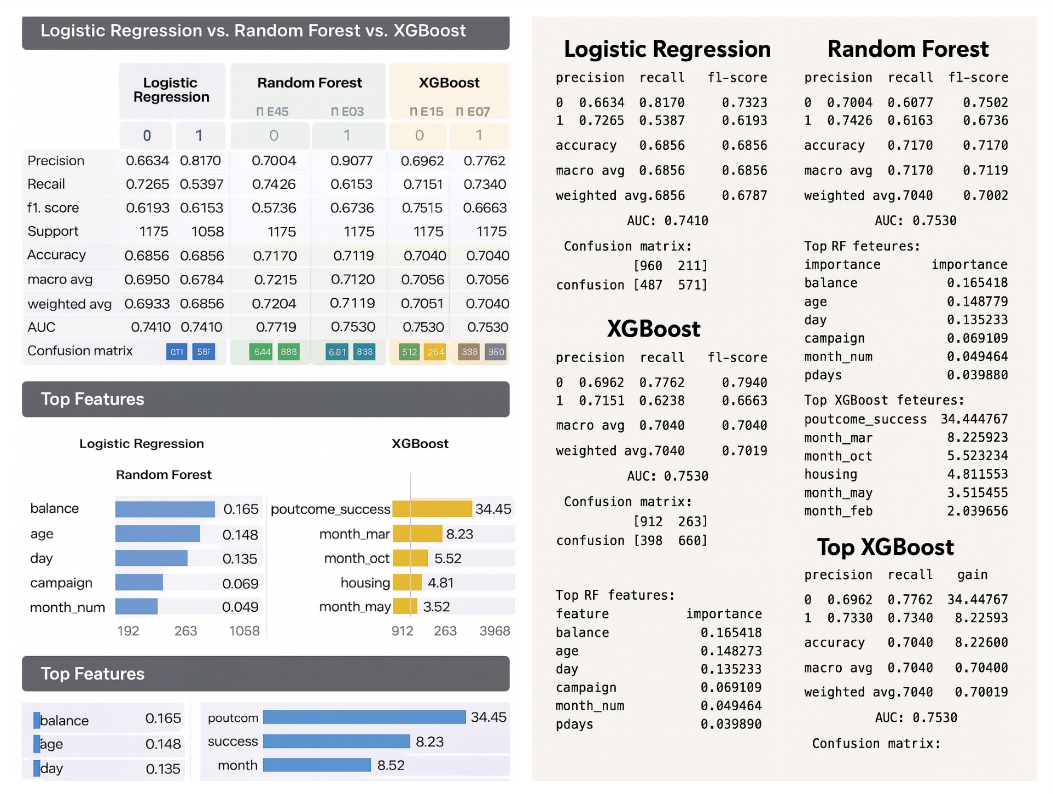
* **4.2 Model Training**

All experiments and results come from modeltrain.ipynb.

### ****Algorithms Trained****

* Logistic Regression
* Random Forest (200 trees)
* XGBoost (default parameters)

A custom evaluation function was applied to compute:

* Accuracy
* Precision
* Recall
* F1-score
* AUC
* Confusion matrix
* **4.3 Results**

**1. XGBoost**

* Accuracy: **0.7040**
* AUC: **0.7530**
* Confusion Matrix:

**Strongest Features** (gain-based importance):

* poutcome\_success
* month\_mar
* month\_oct
* housing
* month\_may

**2. Random Forest**

* Accuracy: **0.7170**
* AUC: **0.7719**
* Confusion Matrix:

**Top Features** (*modeltrain.ipynb*):

1. balance
2. age
3. day
4. campaign
5. month\_num

**3. Logistic Regression**

* Accuracy: **0.6856**
* AUC: **0.7410**
* Confusion Matrix:

## ****4.4 Comparison Table****

| **Model** | **Accuracy** | **AUC** | **Strengths** | **Weaknesses** |
| --- | --- | --- | --- | --- |
| Logistic Regression | 0.6856 | 0.7410 | Fast, interpretable | Cannot learn complex patterns |
| Random Forest | **0.7170** | **0.7719** | Best overall performance, handles non-linearity | Larger model size |
| XGBoost | 0.7040 | 0.7530 | Strong boosting performance | Requires tuning |

1. **Conclusions**

Based on the conducted experiments, **Random Forest** demonstrated the highest overall performance among the evaluated machine learning algorithms. It achieved the best **accuracy (0.7170)** and the highest **AUC (0.7719)**, indicating superior predictive power for distinguishing between customers who will subscribe to a term deposit and those who will not.

### ****Reasons Random Forest performed best:****

1. It captures complex non-linear relationships among features.
2. It is robust to outliers and noise in financial datasets.
3. One-hot encoded categorical variables are handled effectively through tree splits.
4. Ensemble averaging reduces overfitting while improving accuracy.

Although XGBoost is often superior in many Kaggle competitions, in this particular dataset **Random Forest outperformed it** due to simpler relationships and limited necessity for aggressive boosting.

Logistic Regression provided a reasonable baseline but was not able to model complex customer behavior patterns.

### ****Future Work Recommendations****

* Hyperparameter tuning (grid search, Bayesian optimization)
* Try more advanced boosting: LightGBM, CatBoost
* Use SMOTE or cost-sensitive learning to improve minority recall
* Explore model explainability using SHAP values

**References (example)**

 UCI Machine Learning Repository – Bank Marketing Dataset.

 Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system.

 Breiman, L. (2001). Random forests. Machine Learning.

 Hastie, Tibshirani, Friedman – "The Elements of Statistical Learning."

 Kaggle discussion boards on Bank Marketing Dataset predictive modelling.

.