Bank Marketing Prediction Machine Learning Project Presentation

Aditya Bagri
2022029
IIIT Delhi
aditya22029@iiitd.ac.in

Arav Amawate
2022091
IIIT Delhi
arav22091@iiitd.ac.in

Rishit
2022405
IIIT Delhi
rishit22405@iiitd.ac.in

Abstract

Github – https://github.com/adityabagrii/ML-Project
The dataset associated with this report pertains to direct marketing campaigns conducted by a Portuguese banking institution, primarily through phone calls. The objective is to predict whether a client will subscribe to a term deposit based on various client and campaign-related attributes.

1. Introduction

Direct marketing campaigns are crucial for financial institutions looking to expand their customer base and product subscriptions. This report analyzes data from a Portuguese bank's telephone campaigns, focusing on client interactions to predict whether a client will subscribe to a bank term deposit ('yes' or 'no'). This predictive modeling improves customer engagement strategies and marketing effectiveness.

2. Literature Survey

2.1. A Machine Learning Framework towards Bank Telemarketing Prediction

This paper Link to paper introduces a Class Membership-Based (CMB) classifier for predicting bank telemarketing success. It preprocesses heterogeneous data effectively and achieves high accuracy (97.3%) and AUC (95.9%), outperforming Decision Trees, KNN, and SVM. The focus is on transparent preprocessing and managing non-significant attributes, which benefits direct marketing campaigns.

2.2. A Data-driven Approach to Predict the Success of Bank Telemarketing

In this paper Link to paper, the authors utilize a dataset from a Portuguese retail bank (2008-2013) to predict the success of telemarketing calls for long-term deposits. They compare logistic regression, decision trees, neural networks, and support vector machines, finding neural networks most effective and predicting 79% of buyers. The study underscores the importance of feature engineering and robust evaluation methods in enhancing campaign effectiveness.

2.3. Quantification Under Prior Probability Shift: The Ratio Estimator and its Extensions

This research Link to paper presents a ratio estimator for quantification under prior probability shift, generalizing the Adjusted Count (AC) estimator. It minimizes risk and enhances consistency with limited data. The authors also introduce improvements like the combined and regression ratio estimators. The ratio estimator shows greater robustness in real-world applications, including sentiment analysis and marketing trends, compared to "classify and count."

3. About Dataset

The dataset used in this analysis pertains to direct marketing campaigns by a Portuguese bank, featuring attributes about clients, campaign details, and socio-economic factors that may affect a client's decision to subscribe to a term deposit.

3.1. Attribute Information

3.1.1 Bank Client Data

The bank client data includes several features that describe the clients. Age is represented as a numeric value. Job type is categorized into various occupations such as 'admin.', 'blue-collar', and 'entrepreneur', among others. Marital status indicates whether the client is 'divorced', 'married', 'single', or 'unknown', where 'divorced' also encompasses widowed individuals. The education level is classified based on the highest attained, with options ranging from 'illiterate' to 'university.degree'. Additionally, there are indicators for credit default, housing loans, and personal loans, each with options of 'yes', 'no', or 'unknown'.

3.1.2 Related to Last Contact of the Current Campaign

The features include the type of communication used (cellular or telephone), the month and day of the last contact, and the duration of that contact in seconds. The duration is particularly important, as a value of zero usually indicates a 'no' response, but this information is only available after the call.

3.1.3 Other Attributes

The campaign involves tracking the number of contacts made with a specific client, including the last interaction. The "pdays" metric indicates the days since the client was last contacted in a past campaign, with a value of 999 showing that the client has never been contacted before. The "previous" metric counts the number of contacts made prior to the current campaign, and "poutcome" reflects the outcome of the last marketing campaign, which can be 'failure', 'nonexistent', or 'success'.

3.1.4 Social and Economic Context Attributes

The data includes several economic indicators: the employment variation rate, a quarterly numeric indicator; the consumer price index, provided monthly as a numeric value; the consumer confidence index, also a monthly numeric indicator; the Euribor 3-month rate, recorded daily as a numeric value; and the number of employees, counted quarterly as a numeric figure.

3.1.5 Output Variable (Desired Target)

The output variable indicates whether the client has subscribed to a term deposit ("yes" or "no")

3.2. Pre-Processing

The dataset was thoroughly explored, and information about its columns was gathered. A heat-map was created to visualize null values, leading to the decision to drop the "default" column due to significant class imbalance, which had 32,588 false values, 3 true values, and 8,597 missing values. Missing values in the "housing" column were filled with the mode, while rows with missing "loan" values were removed due to class imbalance.

Box plots were generated for numerical columns, revealing outliers in the "age" and "cons.conf.idx" columns. The "duration" column exhibited class imbalance that hindered outlier removal, while "previous," "pdays", and "campaign" columns were identified as discrete values, making outlier removal unfeasible. Count plots provided insights into the distribution of categorical data.

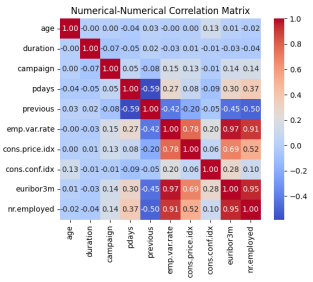

Figure 1. Numerical-Numerical Correlation Heatmap

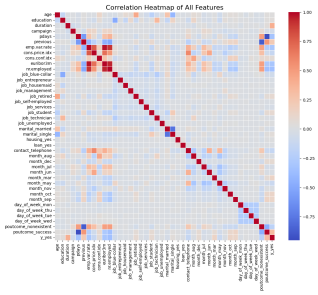

Figure 2. Correlation heatmap of all features

A correlation matrix for numerical features was plotted, and label encoding was applied to the "education" column with custom mapping, while One-Hot Encoding was utilized for several categorical columns, increasing the total number of columns to 41. A correlation heat-map was generated for the 41 features, and a density plot using KDE was created for the "education" column before and after preprocessing to analyze the probability density curve.

Outlier removal was conducted using the IQR technique on the "age" and "cons.conf.idx" columns. A log transformation was performed on the "duration" and "campaign" columns, successfully removing skewness from the "duration" column. The data was then split into train and test sets. For numerical columns having normal distribution like "age", "cons.conf.idx", and "duration", Standard Scaling was applied. For numerical columns that didn't have a normal distribution and lay within a certain range, Min-Max Scaling was applied. To address the class imbalance, SMOTENC was employed, resulting in an increase in the number of rows in the training dataset to 56378.

4. Methodology and Model Details

We used four types of models, including Logistic Regression, Random Forests, Gradient Boosting (XGBoosting) and Multi-Layer Perceptron. We ran Grid Search on them using a parameter grid, which generated around 800+ total fits to find the best hyper-parameters for each model training based on the accuracy of the models. After this, we trained the models on the best hyper-parameters found

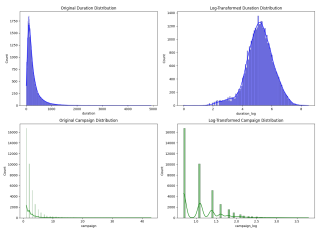

Figure 3. Log transformation of duration and campaign columns

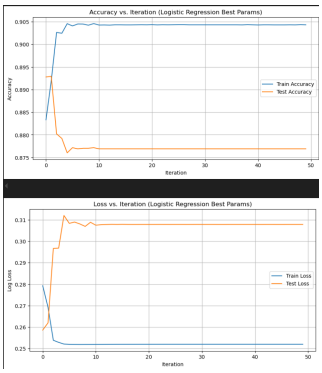

Figure 4. The loss vs epochs curve for Logistic Regression on training and validation sets.

```
Classification Report:
              precision    recall  f1-score   support

       False       0.97      0.89      0.93      7006
        True       0.47      0.81      0.59       866

    accuracy                           0.88      7872
   macro avg       0.72      0.85      0.76      7872
weighted avg       0.92      0.88      0.89      7872

Accuracy: 0.8769
Precision: 0.4657
Recall: 0.8083
F1-Score: 0.5910
Log Loss: 0.3079
AUC: 0.9251
```

Figure 5. Classification report for Logistic Regression

and computed some metrics like accuracy, loss, confusion matrix and others, which are shown below for each of the model types.

1. **Logistic Regression:** The best hyper-parameters found for Logistic Regression are L2 regularization with a regularization strength, i.e. C = 1, maximum iterations = 50 and using the solver saga. These gave a model with an accuracy of 0.8769 on the test set. Figures 4, 5, and 6 show the other computed metrics.

2. **Random Forest:** The best hyper-parameters found for Random Forests training are bootstrapping = False, max depth = None, max features = log2, min samples leaf = 1, min n samples split = 2 and n estimators = 100. These hyper-parameters gave a model with an accuracy of 0.9084. Other metrics are reported in figures 7, 8, and 9.

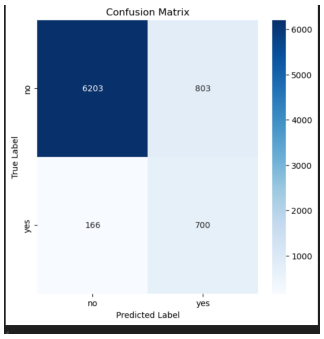

Figure 6. Confusion Matrix for Logistic Regression

```
Classification Report:
              precision    recall  f1-score   support

       False       0.95      0.95      0.95      7006
        True       0.58      0.59      0.59       866

    accuracy                           0.91      7872
   macro avg       0.77      0.77      0.77      7872
weighted avg       0.91      0.91      0.91      7872
```

Figure 7. Classification Report for Random forests

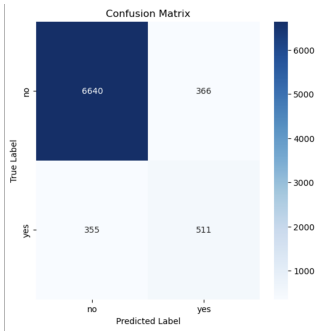

Figure 8. Confusion Matrix for Random Forests

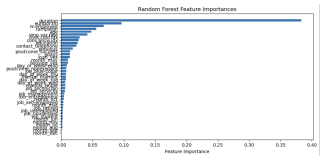

Figure 9. Feature Importance graph for Random Forests

```
Classification Report:
              precision    recall  f1-score   support

       False       0.96      0.94      0.95      7006
        True       0.58      0.67      0.62       866

    accuracy                           0.91      7872
   macro avg       0.77      0.80      0.78      7872
weighted avg       0.92      0.91      0.91      7872
```

Figure 10. Classification report for XGBoosting

3. **XG Boosting:** The best hyper-parameters found for XGBoosting are colsample bytree = 0.8, learning rate = 0.1, max depth = 10, n estimators = 100 and sub samples = 0.9. These hyper-parameters gave an accuracy of 0.9098 and other metrics for XG Boost are given in figures 10, 11 and 12.

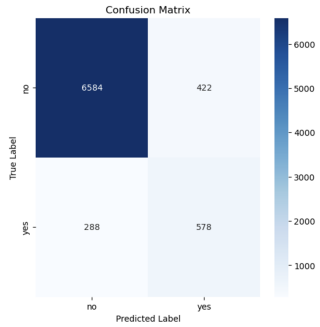

Figure 11. Confusion Matrix for XGBoosting

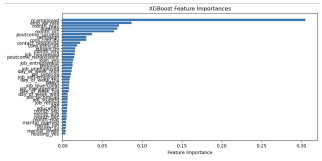

Figure 12. Feature Importance graph for XGBoosting

Classification Report:	precision	recall	f1-score	support
False	0.95	0.92	0.94	7223
True	0.47	0.61	0.53	843
accuracy			0.89	8066
macro avg	0.71	0.76	0.73	8066
weighted avg	0.90	0.89	0.89	8066

Figure 13. Classification report for MLP

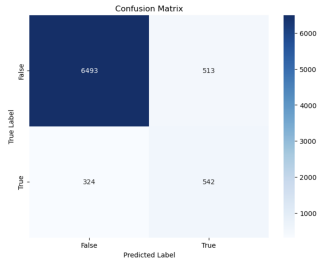

Figure 14. Confusion Matrix for MLP

4. **MLP:** The best parameters for MLP are activation function = 'relu', hidden layer sizes = (100, 100), learning rate = 'constant', max iter= 100 and solver = 'adam'. This results in a model with an accuracy of 0.8872 and other metrics are reported in figures 13, 14 and 15.

5. Results and Analysis

Logistic regression yields a good accuracy score of 0.8769 but its precision value for the minority class is quite low making it less effective in cases where identifying the true value is very critical. Random forests also yield a great accuracy score of 0.9084 but the recall of the class Yes is lowered compared to the XGBoosting model. Similarly MLP also struggles with the precision score due the its comparitively poor performance in the Yes class making XGBoosting a good choice for our dataset with an accuracy of 91 percent and precision score for No = 0.96 and Yes = 0.58.

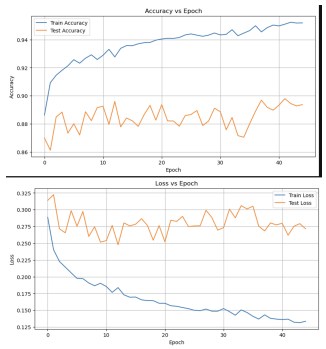

Figure 15. Loss vs Epoch graph for MLP

6. Conclusion

After the comprehensive evaluation of all the four models, i.e., Logistic Regression, Random Forests, XGBoosting and Multi-Layer perceptron. It becomes clear the XGBoost performs great on the given dataset. We can come to this conclusion based on the detailed metrics we observed for each model and noticed the XGBoost has outperformed all the other models in accuracy, precision and recall scores for both the Yes and No classes. It also has an high AUC Score which implies that it can distinguish between subscribers and non-subscribers across various threshold settings which enhances the reliability of the model. XG Boost also identifies critical features which in turn helps to increase the predictive capability of the model. Therefore we can clearly observe from the metrics and performances that XGBoost outperforms and is clearly the best model with excellent predictive capabilities run on the dataset and gives a significantly high accuracy which can be employed by the banks to make their telemarketing tasks more efficient and more productive.

References

1. A Machine Learning Framework towards Bank Telemarketing Prediction (MDPI Journal)
2. A Data-driven Approach to Predict the Success of Bank Telemarketing (ScienceDirect)
3. Quantification Under Prior Probability Shift: The Ratio Estimator and its Extensions (arXiv)
4. Bank Marketing Dataset (Kaggle)