Bank Term Deposit Predictor

Task Overview

The goal of this task is to predict if a user will subscribe to a term deposit scheme for a bank. This will help guide decisions into which customers to focus on when trying to pitch the scheme. The dataset that was used was the Bank Marketing Dataset that can be found on Kaggle.

Initial Data Exploration

The data was loaded into a Pandas DataFrame and resulted in the following findings:

- The data consists of 45211 Rows and 17 Columns.
- No N/A values.
- A y columns that informs us if a customer has subscribed to a term deposit scheme.
- The target variable class is heavily imbalanced with 39922 No and 5289 Yes.

This dataset was chosen due to its decently large number of features and large number of rows which give us more training data to work with in order to create a better model.
The distributions of all variables can be seen in Figure 1.

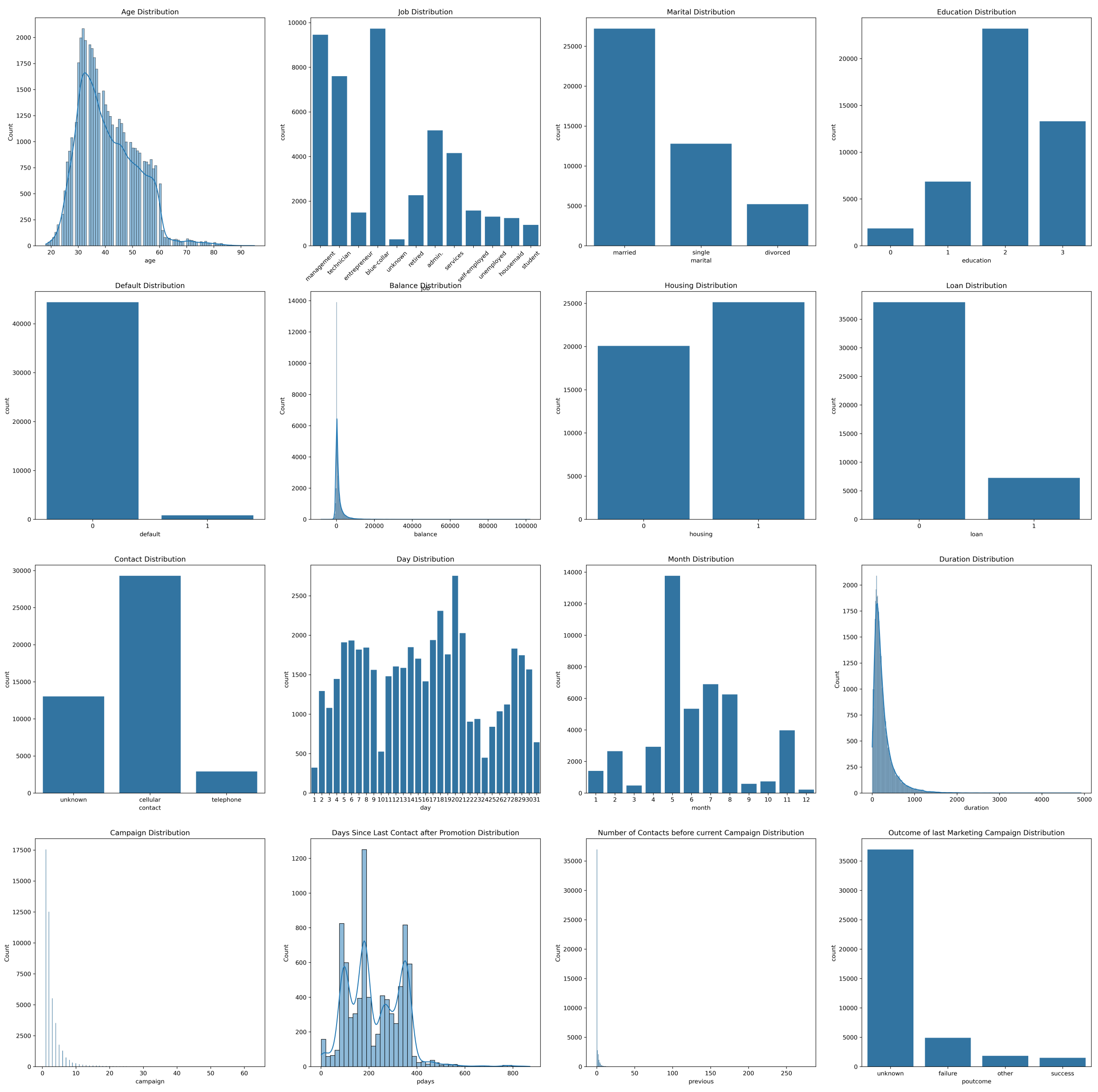


Figure 1: Variable Distributions.

Variables such as age, balance, duration, previous, pdays and campaign do not resemble normal distributions and will need to undergo feature engineering to bring them closer to the desired normal distribution.

Data Cleaning and Feature Engineering

The data cleaning step began with changing the education and month columns to numeric values as there are ordinal and their relations can be explained by numbers.
Next, the balance feature was transformed to better approximate a normal distribution. Due to the presence of negative values and a heavy right-skew, several transformation methods were evaluated, as shown in Figure 2. The most effective approach was the sklearn's QuantileTransformer, which applies a non-linear, rank-based transformation. While this transformation alters the original scale and complicates direct interpretation in monetary units, the results remain interpretable in terms of relative position within the balance distribution. Consequently, model interpretations are framed using a rank-based perspective, reflecting users' balance relative to others rather than absolute values.

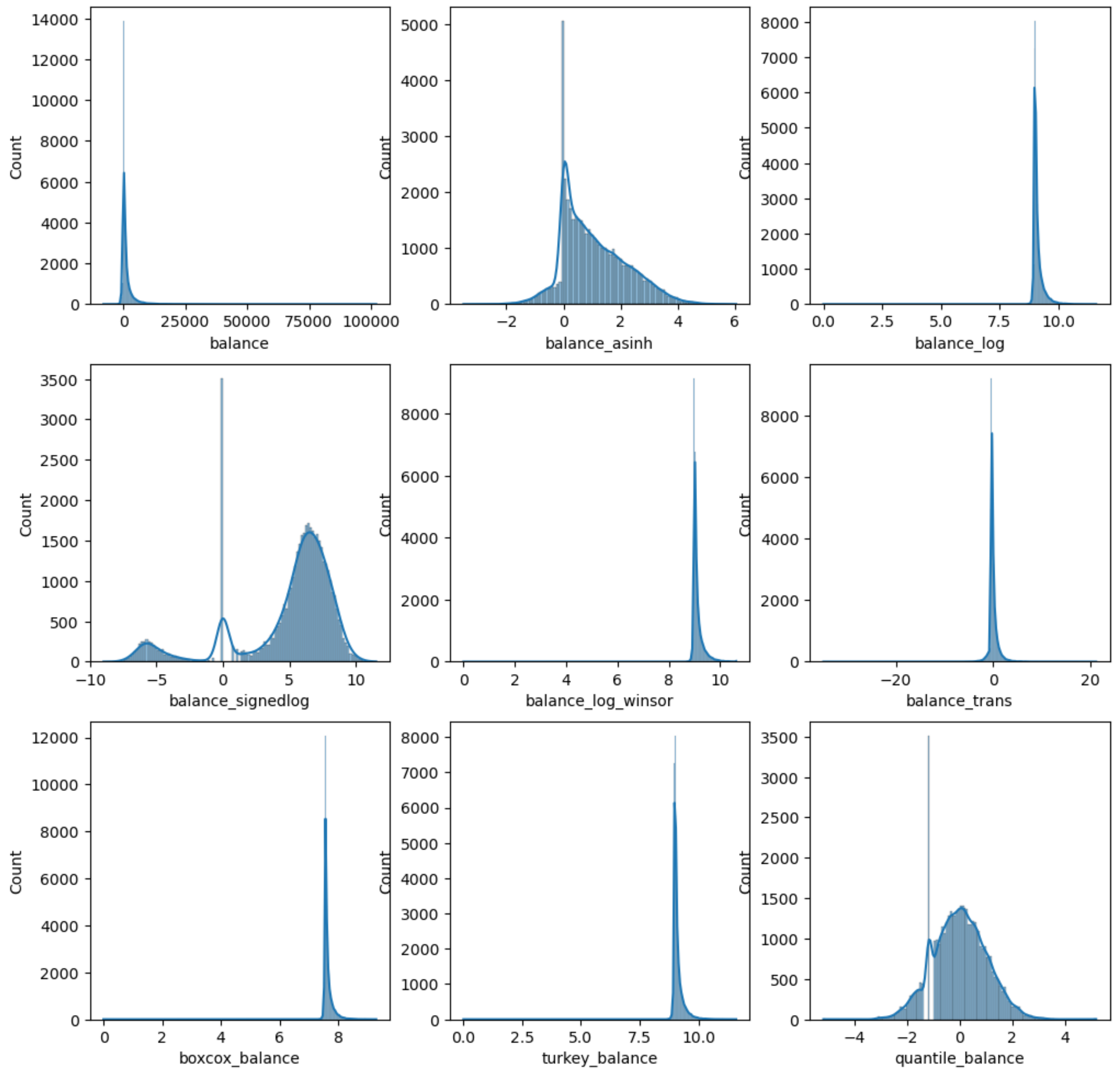
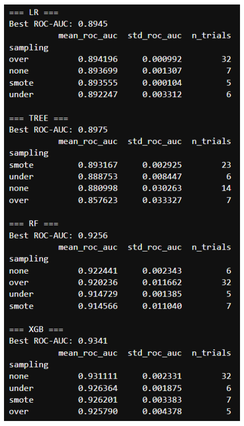


Figure 2: Distributions after applying various transformations to the balance feature.

Age and duration simply had log scales applied to them which shifted them into the desired normal distribution while the campaign and pdays columns where put into bins. The previous column had a much larger concentration of 0 values than any other so it was changed into a binary variable which just indicated if the customer has ever been contacted.
All other categorical variables were one hot encoded using a OneHotEncoder, which resulted in the final dataset consisting of X columns.

Model Selection

To choose the best model we created 4 classes of models: Linear Regression, Decision Tree Classifiers, Random Forests Classifiers and XGBoost Classifiers. Then all of these used various methods to combat the class balance such as SMOTE, undersampling and oversampling. These were then cross validated using a 5 fold cross validation with hyperparameter tuning being performed via optuna.
The results for this step can be seen in Figure 3. This shows that XGBoost achieves the highest AUC-ROC score, without any type of sampling needed to change class balance. The confusion matrix in Figure 4 shows the results of the each model with their best sampling method on the test set.



```
=== LR ===
Best ROC-AUC: 0.8945
          mean_roc_auc  std_roc_auc  n_trials
sampling
over          0.894196     0.000992        32
none          0.893699     0.001307         7
smote         0.893555     0.000104         5
under         0.892247     0.003312         6

=== TREE ===
Best ROC-AUC: 0.8975
          mean_roc_auc  std_roc_auc  n_trials
sampling
smote         0.893167     0.002925        23
under         0.888753     0.008447         6
none          0.880998     0.030263        14
over          0.857623     0.033327         7

=== RF ===
Best ROC-AUC: 0.9256
          mean_roc_auc  std_roc_auc  n_trials
sampling
none          0.922441     0.002343         6
over          0.920236     0.011662        32
under         0.914729     0.001385         5
smote         0.914566     0.011040         7

=== XGB ===
Best ROC-AUC: 0.9341
          mean_roc_auc  std_roc_auc  n_trials
sampling
none          0.931111     0.002331        32
under         0.926364     0.001875         6
smote         0.926201     0.003383         7
over          0.925790     0.004378         5
```

Figure 3: ROC-AUC Scores for models.

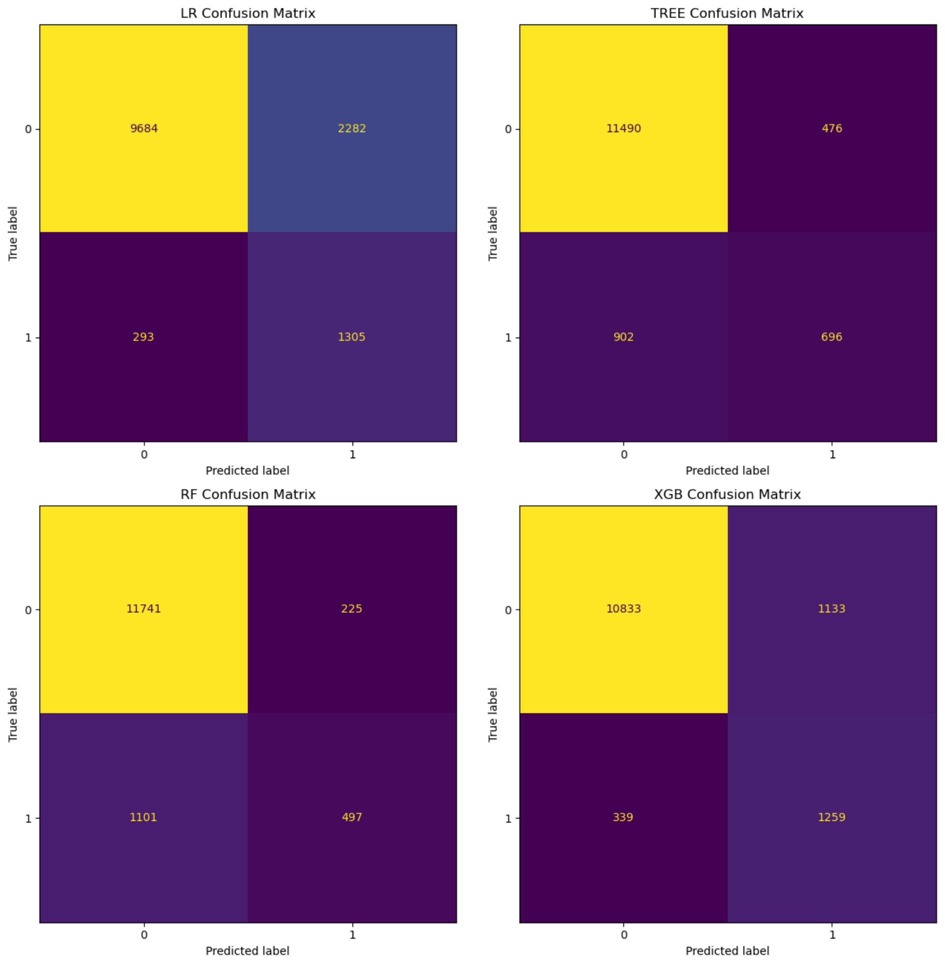


Figure 4: Confusion Matrices for best performing models.

The confusion matrices show that although the Logistic Regression model has a lower ROC-AUC score due to a higher number of errors, most of these errors are Type II errors, which are less critical in this case. When trying to encourage customers to subscribe to a term deposit, it is better to target customers who are less likely to subscribe rather than miss potential subscribers.
In contrast, the Random Forest model produces a larger number of Type I errors, which is undesirable. The XGBoost model, on the other hand, maintains a similarly low number of Type I errors while also

reducing Type II errors.
Overall, the XGBoost model is recommended because it minimizes mistakes, meaning less manpower is required to approach customers unlikely to subscribe, while still accurately identifying most of the customers who will subscribe.
This model is further explained by its SHAP values which can be seen in Figure 5. These show the feature importance for the top 15 most important features. The red indicate a high value or a 1 in the case of binary variables and the blue indicate a low value of a variable or a 0 in the case of a binary variable. The vertical size of the line indicate how that value for the feature affects the model.

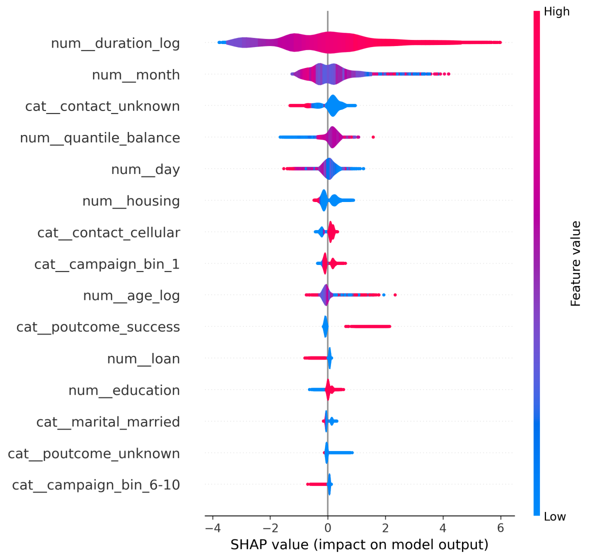


Figure 5: SHAP Values for Feature Importance

Key Findings, Insights and Next Steps

What this work found is that we can relatively accurately predict if a customer will subscribe to a term deposit. The findings show that the XGBoost model works best, with the some of the main features being the duration of the customers previous contract, the month that they have been previously contacted and their age amongst others.
This gives a clear insight into what areas to focus on to increase the number of customers that would subscribe to a term deposit. By focusing on areas such as contact with the customer, more customers may be inclined to subscribe to a term deposit.
The next steps are to deploy the model and continuously monitor it for any changes. If the bank decides to adjust how frequently they contact their customers or the types of customers they contact more frequently, the models predictions may change and some changes may need to be made so continuous monitoring is vital for this task.