Bank Marketing Campaign Prediction

Umar Faruk Abdullahi[1]

Abstract

In this project, we build and evaluate two supervised machine learning models designed predict whether the customer of a bank would accept a fixed term deposit. The project intricately details the criteria employed for feature selection, data preprocessing and provides a comprehensive analysis of the performance metrics associated with the selected machine learning models. As a culminating highlight, the model compares and discusses the performance of the selected machine learning models.

Link to Google Colab.

Keywords: Data Science, Machine Learning, Bank Customers

1. Introduction

Financial institutions, particularly banks, place a significant emphasis on comprehending the intricacies of their customer base. The wealth of demographic data at their disposal serves as a cornerstone for this endeavor, offering insights into the diverse profiles and preferences of their clientele. Armed with this knowledge, banks can fine-tune their marketing strategies to resonate more effectively with specific audience segments, ensuring that their campaigns are targeted towards those most likely to respond positively.

By leveraging data-driven insights, banks can optimize the allocation of both human and technical resources, directing them towards initiatives with the highest potential for success. This strategic approach not only maximizes gains and profitability but also fosters a more tailored and personalized customer experience. As a result, banks can cultivate stronger relationships with their customers, bolstering satisfaction levels and loyalty in the process.

In this project, data is collected from the marketing campaigns of a bank. The project's objective was multifaceted, involving a thorough exploration of the dataset, a detailed examination of relationships between features, the identification of key features, pre-processing of the features and the subsequent training of two machine learning algorithms using a supervised learning approach. Following this, we evaluated the performance of the models, fine-tuning their parameters to optimize and enhance their accuracy.

In the final phase of the project, we compared the performance of the trained models using different approaches.

2. Data

In this phase of the project, we identify the type of the data and distribution of the features with respect to the target variable. The data collected contained numerous variables including:

- 20 features
- 1 target value

The 20 features were largely divided into four groups namely:

1. Client information: age, marital status, education etc.
2. Data related with last contact with customer: month, dayofweek, duration etc.

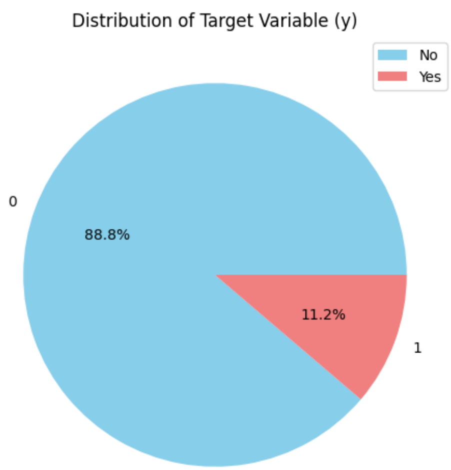

Figure 1: Class Distribution

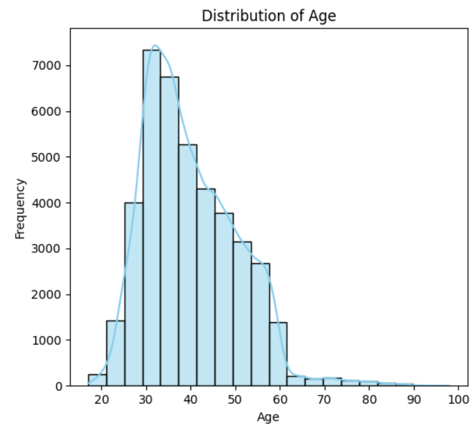

Figure 2: Age

3. Other attributes: previous contact, outcome of previous contact etc
4. Social and economic attributes: consumer price index, consumer confidence index etc.

The target value is the outcome of the contact with the customer. The target variable is a binary variable that can take either **yes** or **no** as shown in a distribution in Figure 1. Due to the categorical nature of the final grade, it is therefore pertinent to treat this as a **classification** problem.

2.1. Data Analysis

We begin by examining the distributions of the key demographic customer data. From our analysis, the graphs offer valuable insights into the demographics of the bank's customer base. Analyzing them collectively, several key patterns emerge. Firstly, a predominant proportion of customers belong to the middle-aged category. Secondly, marriage appears to be a common status among the clientele, with the majority identified as married individuals. Additionally, a notable portion of customers are employed in the industry sector, indicating a significant presence within this workforce segment. Moreover, the data suggests that a considerable proportion of customers possess higher levels of education, implying a certain degree of academic attainment within the customer base.

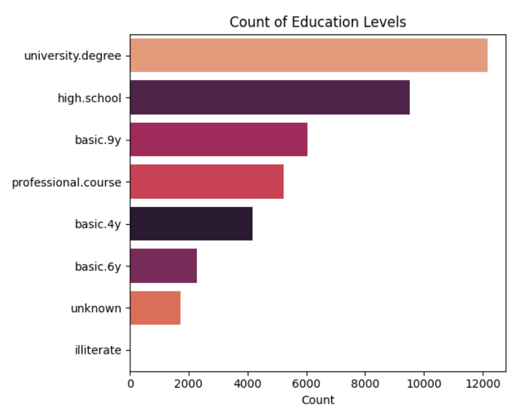

Figure 3: Education

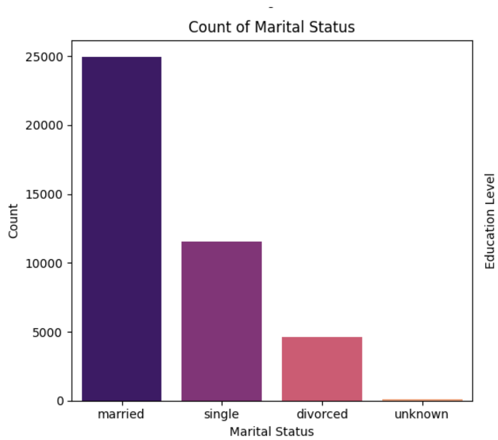

Figure 4: Marital Status

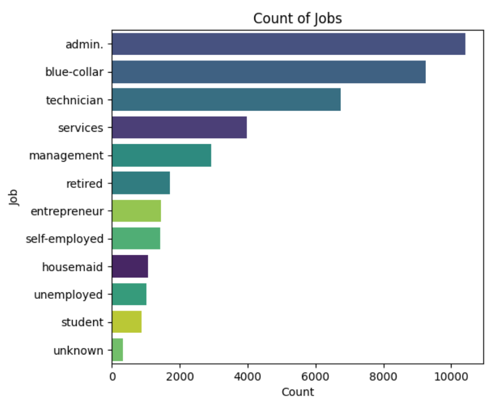

Figure 5: Job Category

In summary, the bank's typical customer profile paints a picture of a middle-aged, married individual with a higher educational background, often employed in the industry sector. Understanding these demographic trends is crucial as it equips the bank with insights to tailor its services effectively, ensuring they resonate with the needs and preferences of its core customer base.

Our analysis revealed a noteworthy trend indicating that longer calls tend to yield more positive responses from customers as shown in Figure 6. However, it's essential to note that there are exceptions within both groups, suggesting that this correlation may not be universally applicable. While longer calls may generally lead to better

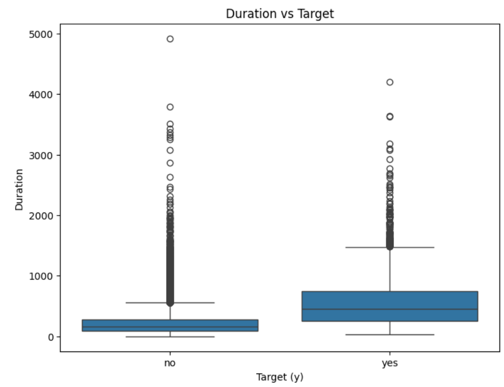

Figure 6: Duration vs Target

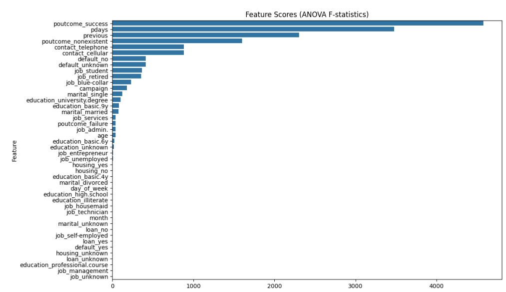

Figure 7: Feature scores

outcomes, it's evident that additional factors may influence customer response.

2.2. *Feature Selection*

Because this is a classification task, we employ the ANOVA (Analysis of Variance) f-classif test. In the context of feature selection, particularly in machine learning, the f-classif test is employed to evaluate whether the means of a particular feature vary significantly across different classes or categories.

We conduct feature selection exclusively on the dataset's internal features, excluding external variables such as social and economic indicators. By focusing solely on internal features, we aim to identify the most informative predictors relevant to our dataset's objectives, thereby refining our model's predictive accuracy and interpretability.

The feature denoted as **poutcome-success** stands out with the highest score. This denotes an important insight that customers who responded **yes** to a previous campaign are more likely to respond postively again.

We also dropped the **duration** feature as suggested by the instructions. This choice is because the duration of a call becomes known only after the call has ended. Since our goal is to build a predictive model, including this feature would distort our model and hinder its predictive accuracy, as the duration is not available beforehand.

2.3. Data Preprocessing

A large proportion of the features in our dataset were represented in categorical format. For training machine learning models, such categorical information must be correctly represented in formats that are suitable for training with machine learning models. One of such representations is **Encoding**. In machine learning, encoding refers to the process of converting categorical data into a numerical format that machine learning algorithms can understand and process. While there are numerous forms of encoding, we selected the following approaches for our project:

1. One-hot encoding: In one-hot encoding, each category or level of a categorical variable is represented by a binary vector. This vector has the length equal to the number of categories, and each position in the vector corresponds to a specific category. The value 1 is assigned to the position corresponding to the category of the observation, while all other positions are filled with zeros. We apply one-hot encoding to features that have no ordinal manner i.e **job, marital, education, default, housing, loan, contact and poutcome**
2. Ordinal encoding: In ordinal encoding, the features within a categorical variable are mapped to integer values based on their order or predefined ranking. This method replaces the categorical labels with numerical codes, allowing algorithms to interpret the data in a format more conducive to analysis and modeling. We apply ordinal encoding to features that posses a predefined order i.e **month, dayofweek**
3. Manual encoding: In this case, we convert the categorical values in the column to appropriate numerical equivalents. This is applied on **pdays** and the target variable **y**

All other numerical values in dataset are left AS IS.

3. Model Training and Evaluation

After carefully selecting the most important features and pre-processing the data for our machine learning task, we proceed to selecting and training two supervised machine learning algorithms. For this project, we selected the following popular algorithms:

1. Random Forest Classifier
2. Extreme Gradient Boosting (XGBoost) Classifier

Before training our machine learning models, we initially tested a **DecisionTree** classifier on the data. Although the model achieved an overall accuracy of 84%, it revealed notably low precision and recall scores of 0.30 each for the minority class - **yes**. Recognizing this as indicative of an **imbalanced dataset**, we sought to rectify the issue by implementing the **BorderlineSMOTE** oversampling technique. This approach aimed to rebalance the dataset by generating synthetic samples for the minority class, fostering a more equitable distribution for our models to learn from. Subsequently, we trained and evaluated our models on this refined dataset to enhance their predictive performance.

3.1. Random Forest Classifier

In the first iteration, a random forest classifier was trained on the sampled with a balanced class weight, maximum depth of 12, 100 estimators and random state of 42. The model achieved 87% accuracy. To evaluate the model, we plot the multi-class **ROC curve** in Figure 8.

To further enhance the performance and accurately measure it's performance across diverse data, a cross-validation was performed with a shuffling folding strategy. A total of 5 folds were trained iteratively with the most performative of the models having an accuracy of 86% - a 1% decline from our initially trained model. The graphs denoting the performance as shown in the comparison in Figure 10.

3.2. XGBoost Classifier

In this section, an XGBoost classifier was trained on the sampled dataset with a logistic objective, maximum depth of 5, learning rate of 0.1 and 100 estimators. The model

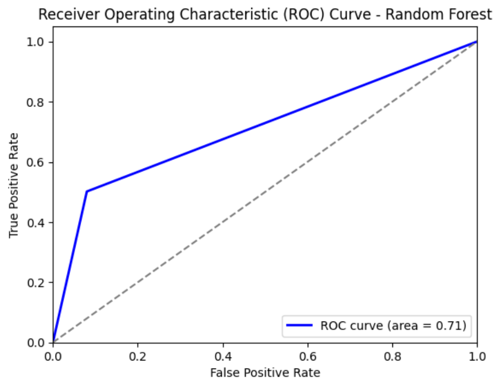

Figure 8: ROC Curve Random Forest

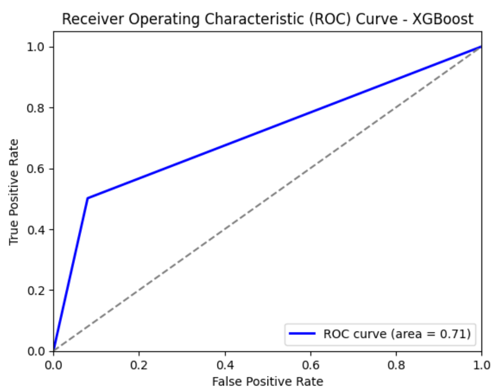

Figure 9: ROC Curve XGBoost

achieved 89% accuracy - a 2% jump from the Random Forest classifier. To evaluate the model, we plot the multi-class **ROC curve** in Figure 9.

As previously employed in the Random Forest Classifier, a cross-validation was also performed with a shuffling folding strategy. A total of 5 folds were trained iteratively with the most performative of the models having an accuracy of 90% - a 1% jump from the initially trained model and 4% from the random forest cross validation set. The graphs denoting the performance as shown in the comparison in Figure 11.

Table 1: Comparison of Model Performance

Model	**Accuracy (%)**	**Mean Squared Error**
Decision Tree	84.0	0.159
Random Forest	87.0	0.127
XG Boost	89.0	0.113

4. Model Comparison

After training the RandomForest and XGBoost classifiers, the XGBoost classifier showed slightly higher accuracy, achieving 89%, compared to the RandomForest classifier's 87%. However, both models demonstrated a similar Area under the ROC curve of 0.71.

Upon applying cross-validation to assess their robustness, the XGBoost model maintained its lead with an average accuracy of 90%, whereas the RandomForest model lagged slightly behind with an average accuracy of 86%. However, a closer inspection revealed that while XGBoost boasted of superior accuracy, it exhibited lower recall than the RandomForest model. This suggests that the RandomForest model, despite its slightly lower accuracy, possessed a superior ability to correctly identify positive instances.

XGBoost slightly surpassed the RandomForest in performance due to its scalability and efficiency that makes it adept at handling large and complex datasets, enabling it to effectively capture intricate relationships within the data. Its iterative nature allows for continuous improvement, as each subsequent iteration builds upon the weaknesses of the previous models, leading to refined predictions and enhanced generalization capabilities.

5. Conclusion

In this project, we began by exploring the dataset to understanding feature relationships. We then pre-procsssed the data to a format compatible with machine learning models and then trained two machine learning algorithms, fine-tuned them for better accuracy, and finally, compare the performance of the models. We were able to train atleast three machine learning models based on three algorithms performing really well with the least accuracy of our model being at 86%.

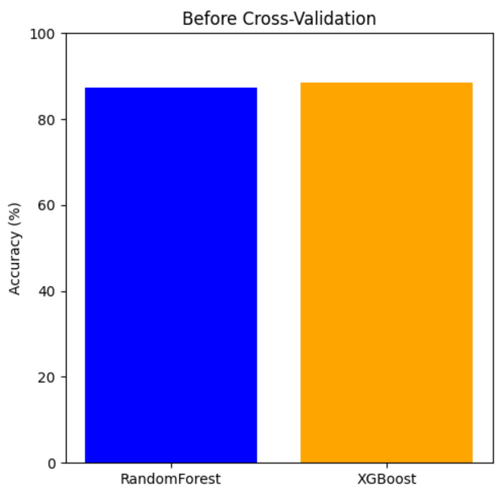

Figure 10: Model Comparison - Without cross-validation

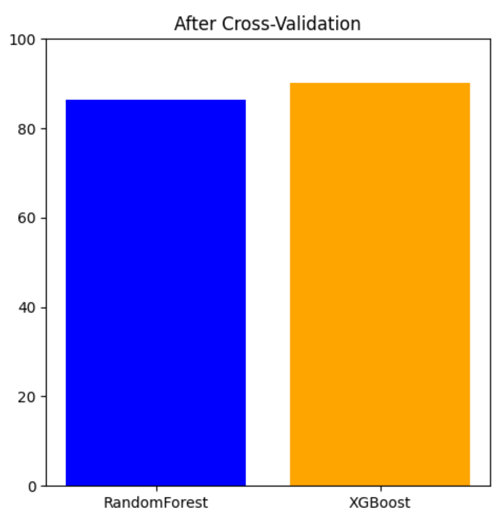

Figure 11: Model Comparison - With cross-validation

Looking ahead, one avenue for expansion lies in the augmentation of our dataset. With the current imbalanced dataset, even after applying several mitigation strategies, the models suffer to generalize well on the minority dataset. With a larger and balanced dataset, more robust machine learning models could be built that will perform consistently in production level environments and the real world.