**Data Analysis Report on**

**PRCP-1003-Customer Transaction Prediction**

**BY:** Sameera Tanveer,

Arunmozhi Muthamizhan

Date: 21-10-2024

Table of Contents:

1. Introduction
2. Data Preprocessing & Feature Engineering
3. Modelling
4. Conclusion

Introduction

Purpose of the report:

The purpose of the report is to give you the complete data analysis report on the given data.

Overview of the dataset:

* The dataset is provided to identify which customer will make a specific transaction in the future, irrespective of the amount of money transacted.
* The domain of the dataset is “Banking”.
* The dataset is anonymized and contains total of 201 features.
* The target column contains 0 and 1 value, where 0 represents that Customer will not do the transaction, and 1 represents that the Customer will do the transaction.
* This makes this problem to be solved as a Classification Problem narrowing down the possible algorithms or ML model applicable to solve this problem.

Objectives:

The main objective task is to get the best model for the production which classifies “Whether or not the Customer will do the transaction?”.

2. Data Preprocessing & Feature Engineering.

- Data Cleaning:

The ID\_code column is irrelevant.

Now, (the dataset’s shape is reduced to 200000, 201) in which 200 are input features and 1 is target column.

- Handling Missing Values:

* No. of Missing Values in the dataset = 0.
* No. of Duplicates in the dataset = 0.
* Hence there is nothing to clean and drop.

- Handling Outliers:

* We have used Isolation Forest technique to identify the outlier and used the identified outlier column to drop them.
* After performing this step, the data has reduced to (190000,202)
* It is 202 columns because With Isolation Forest Method, one more column had been added to identify the outlier.
* Later this column has been dropped to clean the dataset from having any irrelevant columns.
* After dropping the Outlier column, the dataset has reduced to (190000,201).

- Feature Scaling:

Before feeding the data to the model, we have applied Standard Scalar on the dataset, because we had created an EDA report using sweetViz Python library, in which we have observed that the distributions of the features are normal or near to normal distribution.

Hence, we decided to apply the scaling on the dataset using Standard Scaler.

Feature scaling has been done after the splitting of the data, to avoid any data leakage which can later on lead to overfitting and wrong high-performance model that perform well on the test data, but not very well on the actual unseen data from the real world.

Modelling

For this specific dataset, by considering some facts about the data, we have decided to use Random Forest Classifier and then XG Boost Classifier.

Considerations:

1. Dataset is high-dimensional data.
2. Hard to say about multicollinearity.
3. The dataset is highly imbalanced dataset.
4. Not much domain knowledge.
5. **Random Forest Classifier**:

After scaling the X\_train and X\_test separately for avoiding data leakage. We have trained X\_train\_scaled and y\_train.

After training the model, we tested the model by letting the model predict X\_test\_scaled.

Later then we have generated the Classification report, and concluded that the overall performance of the model is pretty good.

This conclusion is supported from the accuracy score of the testing data to the actual data, which is 92.15%.

1. **XG Boost Classifier:**

After scaling the data, we started training the data, this time using XG Boost Classifier, which is highly popular for its advantages of handling large volume of data, high-dimensional data, fast training and better performance than many other algorithms.

After training the model, we tested the model and then calculated the Classification report.

After seeing the Classification report, we have concluded that, it is performing just slightly better than random forest but not much difference in their metrics score.

But we have seen the positive changes in minority class metrics, which shows that XG boost performed well than Random Forest.

This conclusion is supported from the accuracy score of the tested data which is 92.81%.

* **Hyperparameter tuning:**

1. We have decided to perform hyperparameter tuning on XG boost classifier.
2. After performing Random Search, we have not yielded much better accuracy, which led us to go with the existing XG boost model which was already trained.

* **Saving the Model and Variables:**

1. After all of these, we have saved our both models and some important variables using pickle and Joblib which reduces our task of rerunning all the cells again.
2. We have opted this approach as a solution for our issue of losing our work on the daily basis.

Conclusion

* Key Insights:

1. **Problem**: The problem we are solving is a Classification problem where -

0 - Customer will not do the transaction.

1 - Customer will do the transaction.

1. **Data Imbalance:** The data is very highly an imbalanced, therefore it surely creates bias and problems.
2. **Selection of ML model:** Select a Machine Learning algorithm for training that handles imbalanced dataset.
3. **Model Performance:** The XG Boost model showed an accuracy of 93%.
4. **Comparing Models:** Both Random Forest and XG Boost has performed well, but XG boost has shown better performance in minority class metrics which makes it the robust for this problem that we are tackling.
5. **Hyperparameter Tuning:** Performed Hyperparameter tuning using Random Search which did not yield much different accuracy than without Hyperparameter tuning XG boost did.

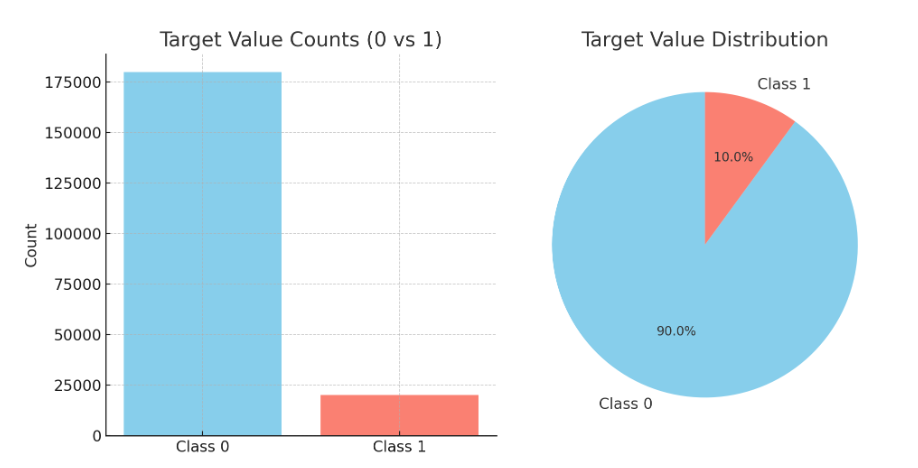
Recommendations:

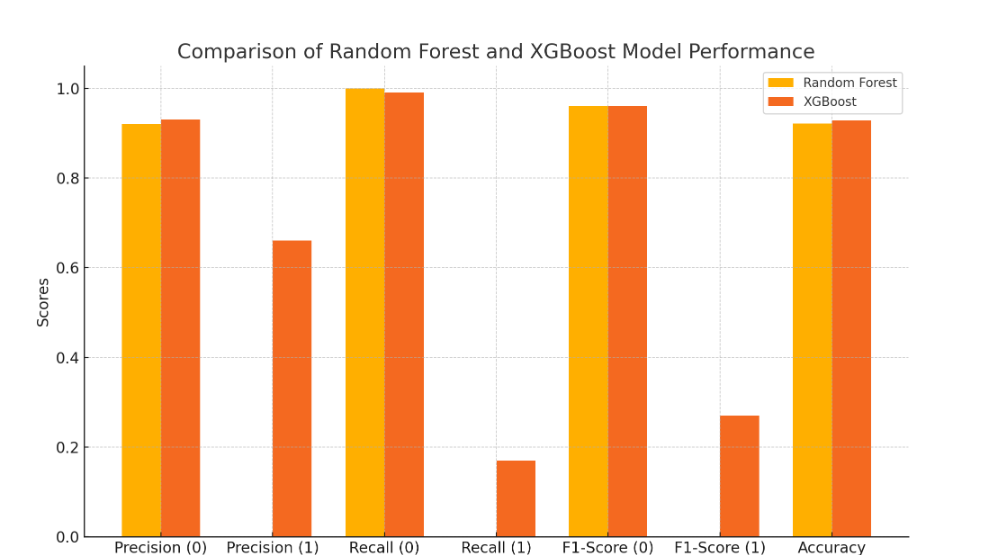
1. Save your work like trained model using pickle for time-saving.
2. Use n\_jobs > 1, when dealing with high-dimensional data and large data to speed up the training process.
3. Scaling after splitting the data, to avoid data leakage.
4. Choose the model based on the characteristics of the data in the dataset.
5. First go with the Random Search hyperparameter tuning then go for Grid search due to its computational expensive.
6. Use Sweetviz or Dtale for faster and basic EDA understanding.

Challenges:

1. The model is totally dependent on statistics, due to less domain exposure in the features of the dataset.

Visualization of Analysis:





* **Random Forest** shows strong performance for class 0, but it struggles with class 1, which results in lower metrics like recall and F1-score for that class.
* **XG Boost** improves precision, recall, and F1-score for class 1, making it better at handling the minority class.

**Report on Challenges faced:**

1. It was hard to apply any domain knowledge for feature selection due to anonymous data.
2. For high-dimensionality we had applied PCA but not yielded good results. Hence, we decided to go with whole data.
3. There was no control for us as a data scientist to choose the methods, it is solely depend on the existing method automatically.

**Conclusion:**

We have successfully, solved the problem by creating the predictive model for predicting Customer Transaction prediction.