**MODULE 3 PROJECT REPORT**

**Module 3 Technique Practice**

Tanvik Vishwanath Poojary

College of Professional Studies, Northeastern University

Logo

Description automatically generated with low confidence

ALY6040: Data Mining Applications

Dr. Kasun Samarasinghe

Jun 12, 2023

Table of Contents

[Introduction 3](#_Toc138723241)

[Goals and Objectives 4](#_Toc138723242)

[Methodology 5](#_Toc138723243)

[Tools and Libraries: 5](#_Toc138723244)

[Data Extraction: 5](#_Toc138723245)

[Data Dictionary: 6](#_Toc138723246)

[Data Cleaning: 6](#_Toc138723247)

[Exploratory Data Analysis 7](#_Toc138723248)

[Age: 7](#_Toc138723249)

[Vintage: 8](#_Toc138723250)

[Average Account Balance: 9](#_Toc138723251)

[Gender: 10](#_Toc138723252)

[Age: 10](#_Toc138723253)

[Occupation: 11](#_Toc138723254)

[Channel Code: 11](#_Toc138723255)

[Credit Product: 12](#_Toc138723256)

[Active: 12](#_Toc138723257)

[Feature Engineering 13](#_Toc138723258)

[Correlation between features: 13](#_Toc138723259)

[Analysis 13](#_Toc138723260)

[Modelling 17](#_Toc138723261)

[Modelling methodology: 17](#_Toc138723262)

[Performance Evaluation 18](#_Toc138723263)

[Base Model: 18](#_Toc138723264)

[After Handling Class Imbalance (SMOTE) 19](#_Toc138723265)

[After Hyperparameter Tuning 21](#_Toc138723266)

[Feature Importance: 23](#_Toc138723267)

[Conclusion and Recommendation 23](#_Toc138723268)

[References 24](#_Toc138723269)

# Introduction

Finding potential customers who will likely accept credit card offers is a crucial task in the financial sector known as credit card lead prediction. Financial institutions can improve the profitability of their operations, increase customer acquisition, and optimize their marketing strategies by accurately predicting leads. In this study, we investigate the use of classification models, particularly Decision Trees, Random Forest, and XGBoost, for the prediction of credit card leads.

The report will take a methodical approach, starting with exploratory data analysis (EDA) to understand the dataset's characteristics and gain insights into them. Examining the distribution of variables, finding any missing or incorrect data, and examining relationships between features are all part of EDA. We'll then examine data preprocessing methods to make sure the dataset is clear, consistent, and prepared for model construction. Taking care of missing values, encoding categorical variables, and applying feature scaling or normalization, as necessary, are all included in this.

We will train and optimize classification models after the data has been prepared. Decision trees offer a clear and understandable method that helps us comprehend the decision-making process. In order to increase accuracy and handle complex relationships in the data, Random Forest combines multiple decision trees. The ensemble boosting algorithm XGBoost, which also uses gradient boosting techniques, provides strong predictive capabilities.

The main emphasis will be on model optimization, which will make use of tools like Standardization, Class Imbalance (SMOTE) , hyperparameter tuning, and model evaluation metrics. In light of the unique demands and priorities of credit card lead prediction, we will evaluate model performance using pertinent metrics like accuracy, precision, recall, F1-score, and AUC\_ROC plot.

Finally, the report will offer recommendations and actionable insights based on the top-performing model. These insights can help financial institutions target potential credit card leads more precisely and maximize their customer acquisition efforts. Marketing strategies can then be informed by these insights.

## Goals and Objectives

The goals of this project are to provide answers to the following queries, draw conclusions from data analysis:

1. To comprehend the many aspects of customers, to help the company find the best customers who are willing to accept credit card using Classification models.

# Methodology

We'll start by looking at the dataset for people who use credit cards. We will analyze the dataset to identify the variables that can be used to forecast the kind of customer lead. In addition, we’ll develop Decision trees, Random Forest, and XGBoost to forecast whether the customer will purchase the credit card. Is\_Lead is our Target variable where we have 2 classes:

* **0**: Not likely.
* **1**: highly likely.

## Tools and Libraries:

Language**:** Python.

Packages: Pandas, numpy, seaborn, sklearn.model\_selection , sklearn.metrics, sklearn.tree, matplotlib.pyplot, optuna, imblearn.

## Data Extraction:

Data source:

* Customer data: <https://www.kaggle.com/datasets/sajidhussain3/jobathon-may-2021-credit-card-lead-prediction?select=train.csv>

The data was obtained by importing it into a pandas dataframe using the read csv function. The train data “train.csv" and “test.csv” collection contains data about customers and their characteristics. The data in these datasets were extracted to evaluate the lead for customer credit and create data models, and it has 245725 rows and 9 attributes. These characteristics contain the components needed to provide the data-driven answers we seek.

**Numeric features:** 'Age', 'Vintage', 'Avg\_Account\_Balance'

**Categorical features:** 'Gender', 'Region\_Code', 'Occupation', 'Channel\_Code', 'Credit\_Product', 'Is\_Active'

## 

## Data Dictionary:

|  |  |  |
| --- | --- | --- |
| Column Attribute | Columns | Data Decription |
| **Target** | Is\_Lead | Where customer has a lead on credit card |
| **Customer Features** | Gender | Gender of the Customer |
|  | Age | Age of the Customer (in Years) |
|  | Region\_Code | Code of the Region for the customers |
|  | Occupation | Occupation Type for the customer |
|  | Channel\_Code | Acquisition Channel Code for the Customer (Encoded) |
|  | Vintage | Vintage for the Customer (In Months) |
|  | Credit\_Product | If the Customer has any active credit product |
|  | Avg\_Account\_Balance  Is\_Active | Average Account Balance for the Customer in last 12 Months  If the Customer is Active in last 3 Months |

## 

## Data Cleaning:

The data cleaning values in “Train Dataset” were converted as the methods below. The list of columns which has been Data Cleaned, Although the data set was pretty clean, only one column had missing values.

**Credit Product:** Credit product had missing values, earlier in the previous iteration during module 2 assignment; We had filled the data with the mode of the column which is “Yes”. But now I think that is not the right choice. Here we have created a new class called “Unknown” to preserve the data integrity which would help the model define that class of people accurately.

|  |
| --- |
| A picture containing screenshot, rectangle, diagram, software  Description automatically generated |

# Exploratory Data Analysis

This section's main objective is to perform exploratory data analysis on the dataset to find patterns in the variable distributions and relationships between the features. Here, we clean the data, get it ready for model fitting, and present it in an easily digestible manner through summary statistics and visuals.

## Age:

The age of the customers varies from 25 to 87 years. More customers are found between 27 to 31 years which shows that they are applying for the credit card and almost constant customer number of customers applying can be found between mid 40’s to their 60’s with 50 years of customer age applying more in this part of the age. We can see less people applying who past 70 years of age, which means less people of that age has interest in credit cards.

|  |
| --- |
| A picture containing plot, screenshot, diagram, line  Description automatically generated |
| Fig 3: Showing the count of values in Age Column |

## Vintage:

The vintage column indicates how long the customer has been a bank client. As seen in Fig. 4, the majority of the clients who are applying for credit cards have been clients of the bank for 20–21 months, or roughly two years. They have been asking for credit cards, and among those who are connected to the bank, the highest credit card request was made by someone between the ages of 18 and 35. And fewer individuals have consistently requested services from the customers who have been linked to them for 40 to 130 months in the bank. In Fig5 we can see that customers who are with the bank from 24 months to 85 months have been denied applying very less. Customers having a higher Vintage on an average are more likely to respond positively.

|  |  |
| --- | --- |
| A picture containing text, screenshot, plot, line  Description automatically generated | A picture containing screenshot, diagram, line, plot  Description automatically generated |
| Fig 4: Showing average vintage count of customers in months | Fig 5: Showing count plot between Vintage and Is\_Lead Column |

## Average Account Balance:

Avg\_Account\_Balance column shows the account balance of customers who has been keeping savings in the bank. In Fig 6 we can see that the greater number of customers have money around $200K in the bank. The customers are positively responding banks where they have a slightly higher average account balance on an average as compared to negatively responding customers.

|  |  |
| --- | --- |
|  | A picture containing diagram, screenshot, line  Description automatically generated |
| Fig 6: Showing average account balance of customers in the bank | Fig 7: Showing count plot between Avg\_Account\_Balance and Is\_Lead Column |

## Gender:

Gender column shows whether the customer is male or female. There are a greater number of observations from Male customers as compared to Female customers in the training data. According to the dataset, Male gender has better conversion ratio than Female gender and has a slightly high positive response for the Male gender.

|  |
| --- |
| A picture containing diagram, line, rectangle, screenshot  Description automatically generated |
| Fig 8: Showing the box plot of gender and Is\_Lead |

## Age:

Below shows the Age of customers. From the figure below we can see that on an average more aged people are likely to respond positively to our offer. The average age of male in dataset has greater average age of female. If we split by gender, the average age of positively responding male is greater than average of positively responding Female.

|  |  |
| --- | --- |
| A picture containing diagram, screenshot, line, pixel  Description automatically generated | A picture containing diagram, line, rectangle, screenshot  Description automatically generated |
| Fig 8: Showing the count plot of age and Is\_Lead | Fig 8: Showing the box plot of gender, age and Is\_Lead |

## Occupation:

This graph shows the occupation of customers who have offered for the credit card. The customer base in training data are majorly Self-Employed or Salaried. Entrepreneurial customers are way more likely to respond positively to Credit Card offer. Salaried people are least likely to respond positively to our offer.

|  |  |
| --- | --- |
| A picture containing screenshot, rectangle, diagram, text  Description automatically generated |  |
| Fig 9: Showing ratio of occupation of customers in the bank | Fig 10: Showing ratio of occupation of customers in the bank and response from Is\_Lead |

## Channel Code:

This Figure shows the channel which the customers has been offered the credit card also Acquisition Channel Code for the Customer. We can see that the Channel X3 has the highest leads and Customer acquired through X3 channel are most likely to respond positively and X1 are least likely.

|  |  |
| --- | --- |
| A picture containing screenshot, plot, rectangle, colorfulness  Description automatically generated | A picture containing screenshot, text, diagram, plot  Description automatically generated |
| Fig 11: Showing ratio of Channel\_Code of customers in the bank | Fig 12: Showing ratio of Channel of customers in the bank and response from Is\_Lead |

## Credit Product:

This shows if the customer had any active credit product such as house loans, Personal Loan or Credit Card. We can see that more number of customers have some sort of credit when offered a credit card. And people already having some kind of credit product are way more likely to respond positively to offer as compared to non-credit users.

|  |  |
| --- | --- |
| A picture containing screenshot, rectangle, text, plot  Description automatically generated | A picture containing screenshot, text, plot, diagram  Description automatically generated |
| Fig 13: Showing ratio of Credit\_Product of customers in the bank | Fig 14: Showing ratio of Credit of customers in the bank and response from Is\_Lead |

## Active:

This shows if the customers are active in the past 3 months. In Fig 16 we can see that the Active Customers(in last 3 months) are slightly more likely to respond positively as compared to inactive customers.

|  |  |
| --- | --- |
| A picture containing screenshot, rectangle, square  Description automatically generated | A picture containing screenshot, text, diagram, plot  Description automatically generated |
| Fig 16: Showing ratio of Active of customers in the bank | Fig 17: Showing ratio of Activeness of customers in the bank and response from Is\_Lead |

# Feature Engineering

## 

## Correlation between features:

We then created a correlation matrix to examine the relationships between features and the target variable, as well as identify any potential multicollinearity. From the correlation matrix, we observed that Channel Code, Vintage, and Age were highly correlated with the target variable (Is\_Lead) compared to others. In addition, Channel Code, Vinatge and Age were slightly correlated with each other(0.6). However, there was no significant multicollinearity between independent features.

A picture containing screenshot, text, square, colorfulness

Description automatically generated

# **Analysis**

The tools and techniques used in the analysis include:

**SMOTE (Synthetic Minority Over-Sampling Technique):** used to address the target variable's class imbalance problem. SMOTE balances the classes and enhances model performance by oversampling the minority class using synthetic instances. We can see the imbalance in the target variable where there is more 0: people who are not likely to accept than 1: people who are likely to accept the offer. Where about 76% of people are not likely to accept the offer and only 23% of people are likely to accept it.

|  |
| --- |
| A screenshot of a pie chart  Description automatically generated with medium confidence |

Fig: Pie chart showing Class Imbalance

SMOTE successfully addressed the class imbalance issue, resulting in a significant improvement in model performance. The accuracy and AUC-ROC scores increased after applying SMOTE, indicating better classification accuracy and a more reliable model.

**Logarithmic Transformation:**  Used to address skewed distribution in the "Avg Account Balance" column. The normalization of the data and creation of a more balanced distribution through this transformation improve the model's capacity to identify patterns and make precise predictions. In the earlier module, we had not changed the values and kept it as it is. But now closely looking at the data we decided to transform the data and use Logarithmic transformation to ’Avg\_Account\_Balance” column of the dataset. The reason is because of Skewed distribution that we can see in the below figure 1. Basically, each value is transformed based on this formula “**probability\_score = no\_of\_leads\_in\_region / no\_of\_customers\_in\_region**”. After applying the transformation, we can see a good distribution of the data.

|  |
| --- |
| **Figure showing the distribution of region code based on leads** |
|  |
| **Below figure showing the distribution after log transformation.** |
|  |

The logarithmic transformation of the "Region\_Code" column revealed a more suitable distribution for modeling, potentially capturing more nuanced relationships and improving the model's ability to distinguish between different classes.

**Outliers:** There were some outliers in the Average account balance, this does not have much impact on the target.

A picture containing screenshot, line, rectangle, plot

Description automatically generated

**Standard Scaling:** Applied standard scaling to numerical variables such as Age, Vintage, and Avg\_Account\_Balance. Scaling brings these variables to a similar range, preventing any feature from dominating the model and ensuring fair weighting during the analysis.We used standard scalar to scale the numerical variables such as Age, Vintage, and Avg\_Account\_Balance to normalize them.

A picture containing plot, diagram, line, screenshot

Description automatically generated

Standard scaling normalized the numerical variables, preventing any feature from dominating the model's predictions. It helped maintain fairness and consistency in the analysis.

Note: As the model was not overfitting before and after performance tuning Regularization implementation was not necessary.

# Modelling

Modelling methodology:First we had training and test data separately. So, we split the model using train\_test\_split function in 80:20 ratio. We then trained the model with the parameters which we had set accordingly without any base and predicted the result without any sampling technique as a base model. Then we used Synthetic Minority Oversampling Technique (SMOTE) method to remove class imbalance. To further improve the model, we used Optuna algorithm for hyperparameter tuning and get the best parameters. We then compared the performance metrics from all the models and concluded the best method and model to predict credit card lead to the company.

**Decision Trees:** Decision trees are a popular and widely used machine learning algorithm for both classification and regression tasks. They are versatile, easy to understand, and can handle both numerical and categorical data. In a decision tree, the data is split into different branches based on feature values, allowing the algorithm to make decisions or predictions at each node. The tree structure is built by recursively partitioning the data based on the selected features and their respective thresholds, with the goal of maximizing information gain or minimizing impurity.

**Random Forest:** Random Forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. By training each tree on a different subset of the data and considering only a subset of features at each split, Random Forest introduces randomness and diversity. The final prediction is determined by majority voting (for classification) or averaging (for regression) the predictions of all the trees. This approach improves accuracy and robustness compared to a single decision tree. Random Forest is less prone to overfitting and can handle a variety of data types. It also provides feature importance measures, aiding in feature selection and gaining insights from the data.

**Xtreme Gradient Boosting (XGBoost):**  XGBoost (Extreme Gradient Boosting) is a popular machine learning algorithm that belongs to the family of gradient boosting methods. It is known for its high performance and efficiency in handling structured/tabular data. Gradient boosting is a supervised learning process that combines the predictions of a number of weaker, simpler models to attempt to properly predict a target variable.

# Performance Evaluation

## Base Model:

Based on the provided metrics, we can summarize the performance of the three base models are: The ***decision tree*** model achieved an accuracy of 0.790, a recall of 0.562, a precision of 0.557, and an F1 score of 0.560. These metrics indicate moderate performance in predicting the target variable. However, the model's recall and precision are relatively low, suggesting that it may struggle to correctly identify positive cases and may have a higher false positive rate. The ***random forest*** model outperformed the decision tree model with an accuracy of 0.854, a recall of 0.578, a precision of 0.748, and an F1 score of 0.652. The higher accuracy and precision indicate improved overall performance and better ability to correctly identify positive cases. However, the recall is still relatively low, indicating that some positive cases are being missed. The ***XGBoost model*** achieved the highest performance among the three models with an accuracy of 0.860, a recall of 0.559, a precision of 0.792, and an F1 score of 0.655. It demonstrates good overall accuracy and precision, showing the ability to correctly classify positive cases. However, similar to the other models, the recall is relatively low, indicating some missed positive cases. Looking at the ROC Curve the Random Forest and XGBoost models had a decent Area under the curve, but the precision recall curve shows the models behaving worse in precision and recall metrics.

From the base models, the random forest and XGBoost models exhibit better performance compared to the decision tree model in terms of accuracy, precision, and F1 score. However, there is still room for improvement in terms of recall, particularly in correctly identifying positive cases.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Metrics** | **Decision Trees** | **Random Forest** | **Xtreme Gradient Boost (XGB)** |
| **Accuracy** | 0.790 | 0.854 | 0.860 |
| **Recall** | 0.562 | 0.578 | 0.559 |
| **Precision** | 0.557 | 0.748 | 0.792 |
| **F1 Score** | 0.560 | 0.652 | 0.655 |
| **Confusion matrix** | A picture containing text, font, graphics, typography  Description automatically generated | A picture containing text, font, white, typography  Description automatically generated | A picture containing text, font, graphics, typography  Description automatically generated |
| **AUC\_ROC Curve** | A graph with a line  Description automatically generated with low confidence | A picture containing text, line, plot, diagram  Description automatically generated | A picture containing text, line, plot, diagram  Description automatically generated |
| **Precision- Recall Curve** | A picture containing text, line, plot, diagram  Description automatically generated | A graph with a line  Description automatically generated with low confidence | A picture containing text, plot, line, diagram  Description automatically generated |

**A picture containing text, line, screenshot, plot

Description automatically generated**

Fig: Combined AUC\_ROC Curve of all models

A picture containing text, plot, line, diagram

Description automatically generated

Fig: Combined P-R\_Curve

## After Handling Class Imbalance (SMOTE)

Based on the provided metrics after handling class imbalance using SMOTE, we can summarize the updated performance of the three models as follows: The **decision tree** model achieved an accuracy of 0.859, a recall of 0.860, a precision of 0.858, and an F1 score of 0.859. These metrics indicate improved performance compared to the previous results. The model shows better accuracy in predicting the target variable, and both recall and precision have increased. However, the model's performance still shows some room for improvement. The **random forest** model further improved its performance with an accuracy of 0.894, a recall of 0.899, a precision of 0.887, and an F1 score of 0.893. The model demonstrates a significant increase in accuracy, recall, precision, and F1 score compared to the previous results. It performs well in correctly classifying positive cases and shows a balance between precision and recall. The **XGBoost model** also shows improvement with an accuracy of 0.902, a recall of 0.869, a precision of 0.932, and an F1 score of 0.899. It achieves the highest accuracy, precision, and F1 score among the three models after handling class imbalance. The model shows good performance in both precision and recall, indicating a better balance in correctly identifying positive cases. Looking at the ROC Curve the Random forest and XGBoost models had a decent Area under the curve, but the precision recall curve for decision tree is still worse and random forest, XGBoost models has a good PR curve.

Handling class imbalance using SMOTE has significantly improved the performance of all three models. The random forest and XGBoost models exhibit better overall performance, with the XGBoost model achieving the highest accuracy, precision, and F1 score. However, it's important to note that each metric should be considered in the context of the specific problem and its requirements.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Metrics** | **Decision Trees** | **Random Forest** | **Xtreme Gradient Boost (XGB)** |
| **Accuracy** | 0.859 | 0.894 | 0.902 |
| **Recall** | 0.860 | 0.899 | 0.869 |
| **Precision** | 0.858 | 0.887 | 0.932 |
| **F1 Score** | 0.8590 | 0.893 | 0.899 |
| **Confusion matrix** |  | A picture containing text, font, typography, graphics  Description automatically generated |  |
| **AUC\_ROC Curve** | A picture containing text, line, plot, screenshot  Description automatically generated | A picture containing text, line, plot, screenshot  Description automatically generated | A picture containing text, line, plot, screenshot  Description automatically generated |
| **Precision- Recall Curve** | A picture containing line, plot, diagram, text  Description automatically generated | A picture containing text, line, plot, diagram  Description automatically generated | A picture containing line, text, plot, diagram  Description automatically generated |

## After Hyperparameter Tuning

Based on the provided metrics after hyperparameter tuning, we can summarize the updated performance of the three models as follows: The decision tree model achieved an accuracy of 0.860, a recall of 0.825, a precision of 0.887, and an F1 score of 0.855. Compared to the previous results, the model shows a slight improvement in accuracy and precision. However, the recall has decreased, indicating a lower ability to correctly identify positive cases. The random forest model maintains its performance with an accuracy of 0.894, a recall of 0.890, a precision of 0.896, and an F1 score of 0.893. The model shows consistent performance compared to the previous results, with high accuracy and balanced recall and precision. The XGBoost model also demonstrates strong performance with an accuracy of 0.903, a recall of 0.869, a precision of 0.944, and an F1 score of 0.899. The model maintains its high accuracy while improving precision compared to the previous results. The recall remains relatively stable. Looking at the ROC Curve the Random forest and XGBoost models has a good Area under the curve, but the precision recall curve for decision tree is still worse and random forest, XGBoost models has a good PR curve.

In conclusion, after hyperparameter tuning, the models show similar or slightly improved performance across the board. The random forest model continues to perform consistently well, while the XGBoost model achieves the highest accuracy, precision, and F1 score among the three models. The decision tree model shows modest improvements but still lags behind the other two models in terms of overall performance.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Metrics** | **Decision Trees** | **Random Forest** | **Xtreme Gradient Boost (XGB)** |
| **Accuracy** | 0.860 | 0.894 | 0.903 |
| **Recall** | 0.825 | 0.890 | 0.869 |
| **Precision** | 0.887 | 0.896 | 0.944 |
| **F1 Score** | 0.855 | 0.893 | 0.899 |
| **Confusion matrix** | A picture containing text, font, graphics, typography  Description automatically generated | A picture containing text, font, typography, graphics  Description automatically generated | A picture containing text, font, graphics, typography  Description automatically generated |
| **AUC\_ROC Curve** |  | A picture containing text, line, plot, screenshot  Description automatically generated |  |
| **Precision- Recall Curve** |  |  |  |

A picture containing text, screenshot, line, plot

Description automatically generated

Fig: Combined AUC\_ROC Curve of all models

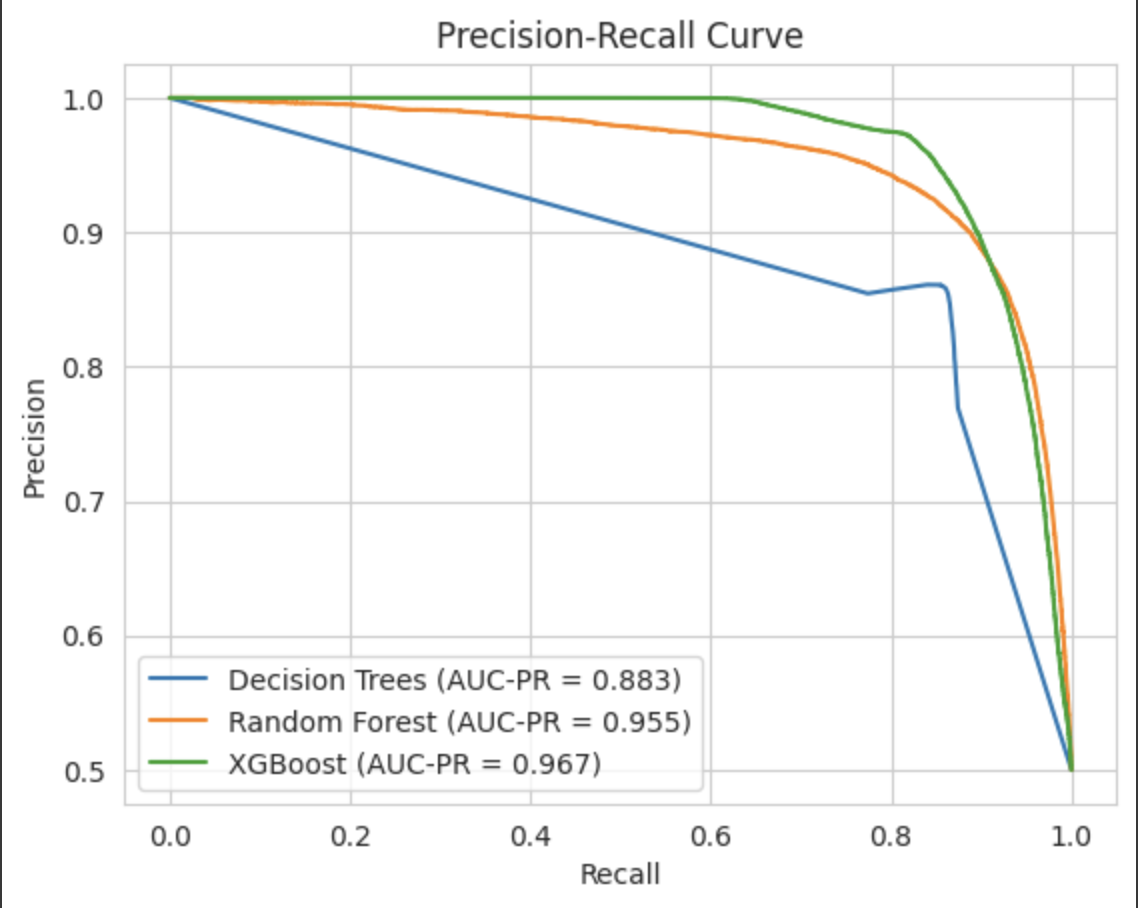


Fig: Combined P-R\_Curve

## Feature Importance:

From the below figure, we can see that the Credit\_Product is one of the important features for the credit card company to accurately determine which customer is good to get the credit card offer and the customer who is likely to accept it. In a way it makes sense, the more the debt you are in the more you need money to repay it.

Also vintage is the second important feature for the model as longer the person is associated with the bank more likely he will open up a credit in the bank.

A picture containing text, screenshot, software, diagram

Description automatically generated

# Conclusion and Recommendation

In conclusion, we gained more insight into the features and used different classification models, like Decision Trees, Random Forest, and XGBoost to assess whether customers are likely to accept credit card offers after following the machine learning life cycle to build our Credit Card Lead Prediction. The models were assessed using a variety of metrics, such as accuracy, recall, precision, and F1 score.

We observed significant improvements in model performance across all three models after addressing class imbalance with SMOTE and performing hyperparameter tuning. With regard to accuracy, recall, precision, and F1 score, the Random Forest and XGBoost models consistently outperformed the Decision Trees model.

After accounting for class imbalance, the Random Forest model showed the highest accuracy of 89.4%, while XGBoost attained an accuracy of 90.3% after hyperparameter tuning. Additionally, these models showed increased recall, precision, and F1 score, demonstrating improved performance in accurately identifying positive cases and reducing false positives.

These findings lead us to suggest the Random Forest or XGBoost model for the prediction of credit card leads due to their superior performance. It is significant to remember that the model selection may be influenced by the particular needs and priorities of the company. SMOTE was successfully used to identify the effects of class imbalance on model performance. We also highlighted the significance of data preprocessing operations like handling missing values and transforming skewed features.

My recommendation to the company is to used credit\_product where customers already have some sort of credit line to target and use XGBoost model to predict other customers for whom to target. Sampling and hyperparameter tuning has greatly helped the model to increase its performance.

# References

1. Patel, S. (2019). *K-means Clustering Algorithm: Implementation and Critical Analysis*. Scholars’ Press.
2. *seaborn: statistical data visualization — seaborn 0.12.1 documentation*. (n.d.). Retrieved October 23, 2022, from <https://seaborn.pydata.org/index.html>
3. *matplotlib.markers — Matplotlib 3.6.0 documentation*. (n.d.). Retrieved October 23, 2022, from <https://matplotlib.org/stable/api/markers_api.html>
4. Wes McKinney (2017) Python for Data Analysis, O’Reilly.
5. Aurelien Geron Hands-on (2019) Machine Learning with Scikit-Learn, Keras & TensorFlow 2nd edition, O’Reilly.
6. Jose Portilla- Python for Data Science and Machine Learning, Udemy from <https://www.udemy.com/share/101WaU3@Q75P8RKsTd2OYIwOE9om1bJedw9egQcfmeX9THYeBjvWQnNDGRFbmI3nQ8iZJBlS/>
7. <https://scikit-learn.org/stable/modules/generated/sklearn.tree.plot_tree.html>
8. <https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/>