RISK ANALYTICS IN BANKING FINANCIAL SERVICES



Created by- Team08

## Manish Rao (002818773) Anshul Singh (002747160)

[rao.mani@northeastern.edu](mailto:rao.mani@northeastern.edu) [singh.anshul@northeastern.edu](mailto:singh.anshul@northeastern.edu)

# **Introduction**

## Background

Bank loans and credits are a vital demand in modern society. Banks gain a large portion of the entire profit from this alone. It is advantageous for students to manage their educational and living expenditures, as well as for individuals to purchase any type of luxury such as houses, vehicles, and so on.

However, while determining whether the applicant's profile is important to be granted a loan or not banks must deal with numerous issues. So, in this case, we will use Machine Learning with Python to facilitate their work and predict whether a candidate's profile is appropriate or not based on crucial variables such as Marital Status, Education, Applicant Income, Credit History, and so on.

## Motivation

The motivation behind this case study lies in the pursuit of practical knowledge application within a real-world business context. We seek to explore the implementation of Exploratory Data Analysis (EDA) techniques, extending beyond the theoretical understanding gained in our EDA module.

Moreover, this case study offers a unique opportunity to delve into the intricate domain of risk analytics within the banking and financial services sector.

It allows us to gain insights into how data can be strategically harnessed to minimize the inherent risks associated with lending to customers.

## Goals

Loan prediction often entails the lender reviewing various background information about the applicant and choosing whether or not to provide the loan. Credit score, loan amount, lifestyle, career, and assets are all determining factors in loan approval. If people with comparable requirements to yours have paid their dues on time in the past, it is more likely that your loan will be approved as well.

Machine learning algorithms can take advantage of this reliance on previous experiences and comparisons with other applicants to create a data science challenge that predicts the loan status of a new application based on comparable principles.

Several collections of data from previous loan applicants employ various factors to determine loan status. A machine learning model can investigate.

# **Methodology**

Data Pre-processing and Cleaning: Dataset will be pre-processed to have data usable and accurate. Identify and address data quality issues. Check for missing values, duplicated rows, and outliers. Decide on strategies for handling missing data, such as imputation or removal, and deal with duplicates and outliers appropriately.

Data Exploration: Create visualizations, including histograms, scatter plots, and bar charts, to better understand variable distributions, relationships, and patterns within the data. Use libraries like matplotlib and seaborn to generate visual insights.

Feature Engineering: Engineer new features or transform existing ones to enhance the dataset for analysis and modelling. This step may involve feature scaling, one-hot encoding, or other data transformations to improve model performance.

Model Selection and Assessment: In our analysis of the dataset, we will employ a range of machine learning algorithms, including Logistic Regression, Binary and Multiclass Classifications, Decision Trees, KNN Classification, Support Vector Machines, and Random Forest.

Hyperparameter Tuning: The hyperparameters of the models are tuned to improve their performance. This is done using techniques like GridSearchCV and RandomSearchCV, which search through a predefined space of hyperparameters to find the best ones.

Model Evaluation : The performance of the model is evaluated using various metrics like accuracy, precision, recall and F1 score. The models are also evaluated using confusion matrices.

Resampling: As the dataset is highly imbalanced, SMOTE(Synthetic Minority Over sampling Technique) is used to oversample the minority class. This helps to improve the performance of the models on the minority class.

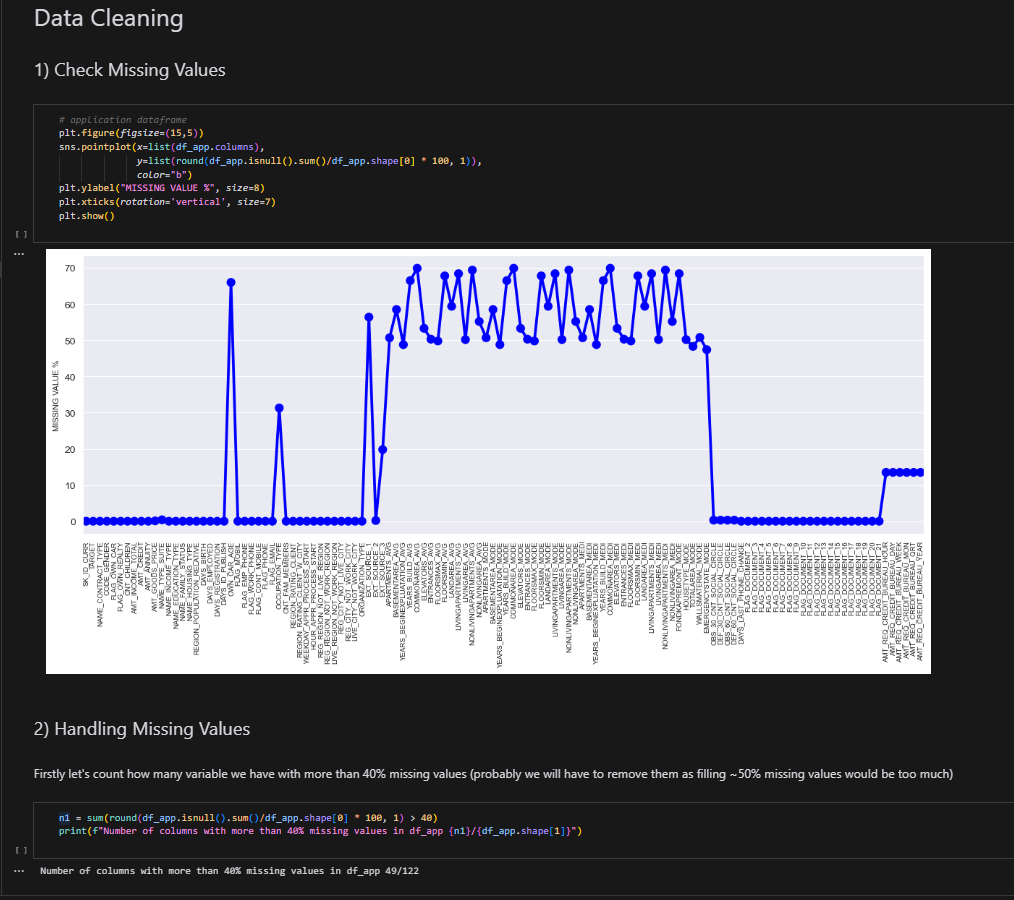
# **Results and Analysis**

In the result and analysis section we are going to deep dive into the project where we show every step we have taken to reach the end goal **of the desired project**

## Exploratory Data Analysis

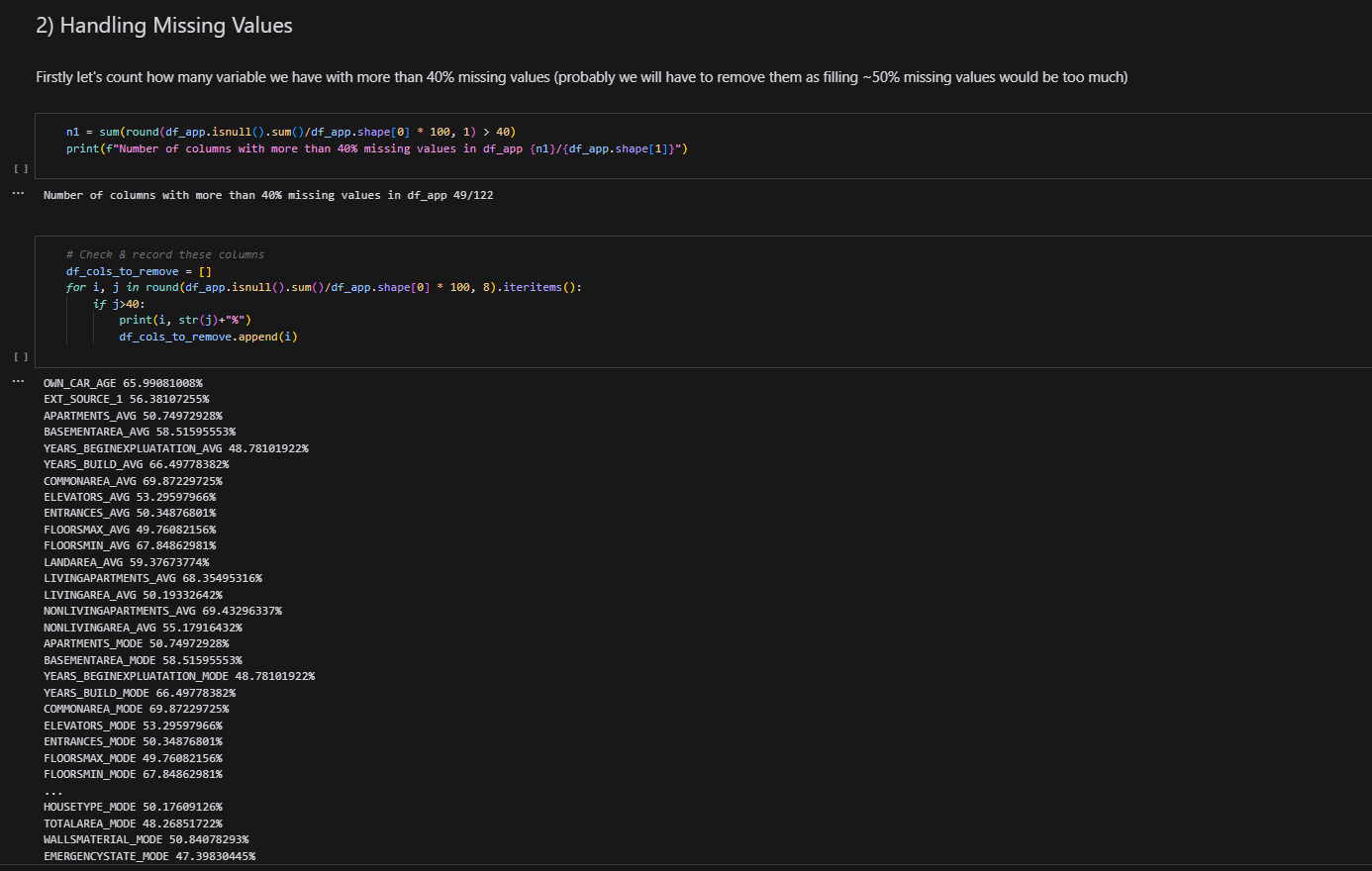
Started out by importing all the necessary files, libraries required for the project and setting the display options for the data frame.

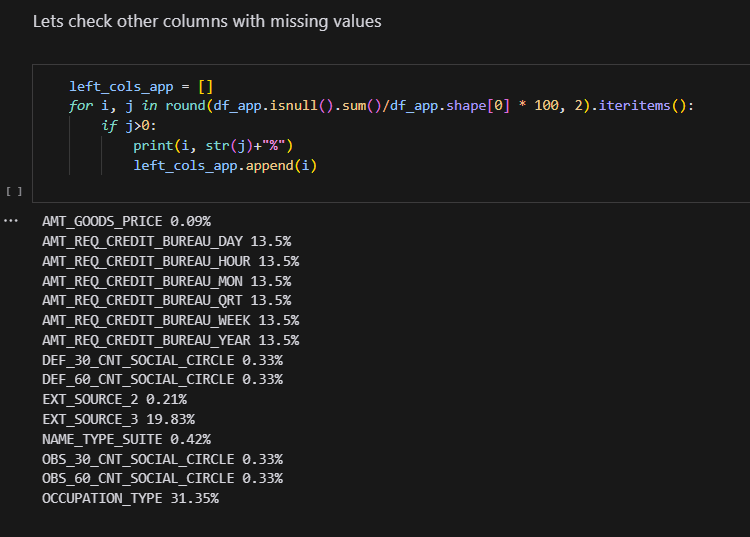
**Data Cleaning**-



* Analysed the data and detected the null values in the data frame, because more than 50% missing values would be too much. Checked how many features had more than 40% null values.

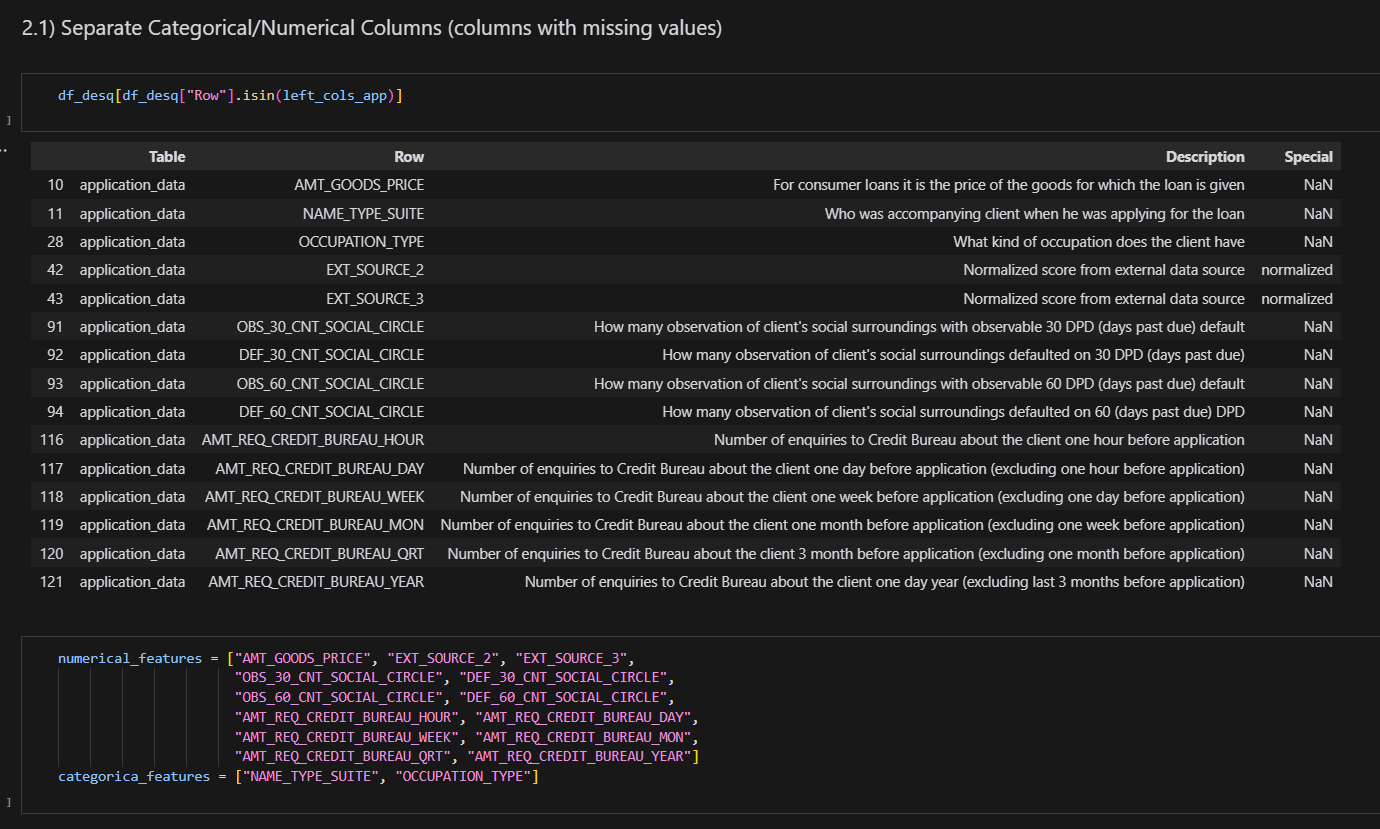
**Handling Missing Values**





* Removed the columns that have a null percent of more than 40%.
* Every column in the dataset and correlate it with the target column to check if the column has any relation with the user being a defaulter or not. We use this column to compare it with other columns

**Separating Categorical and Numerical Columns**

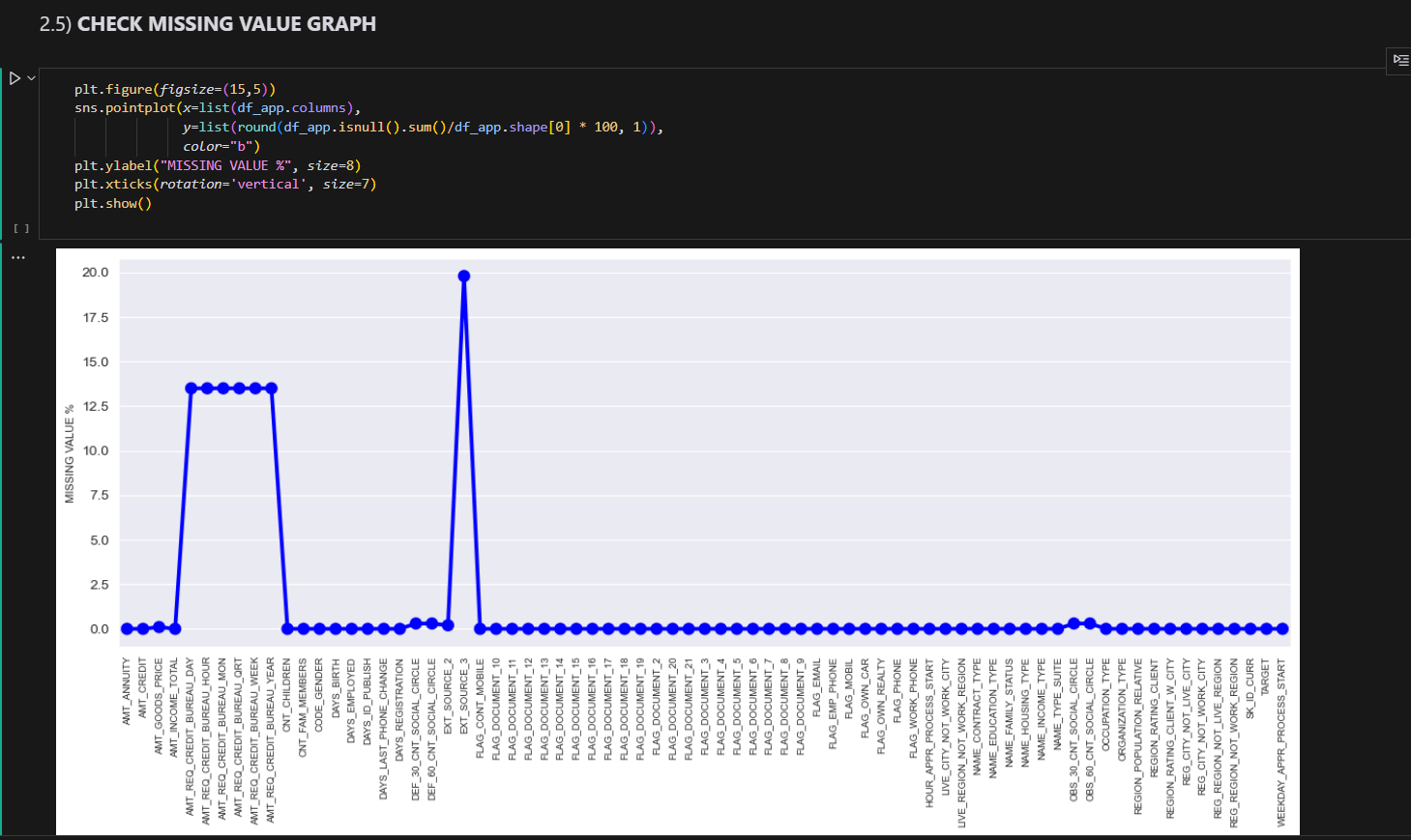


**Filling Missing Values**

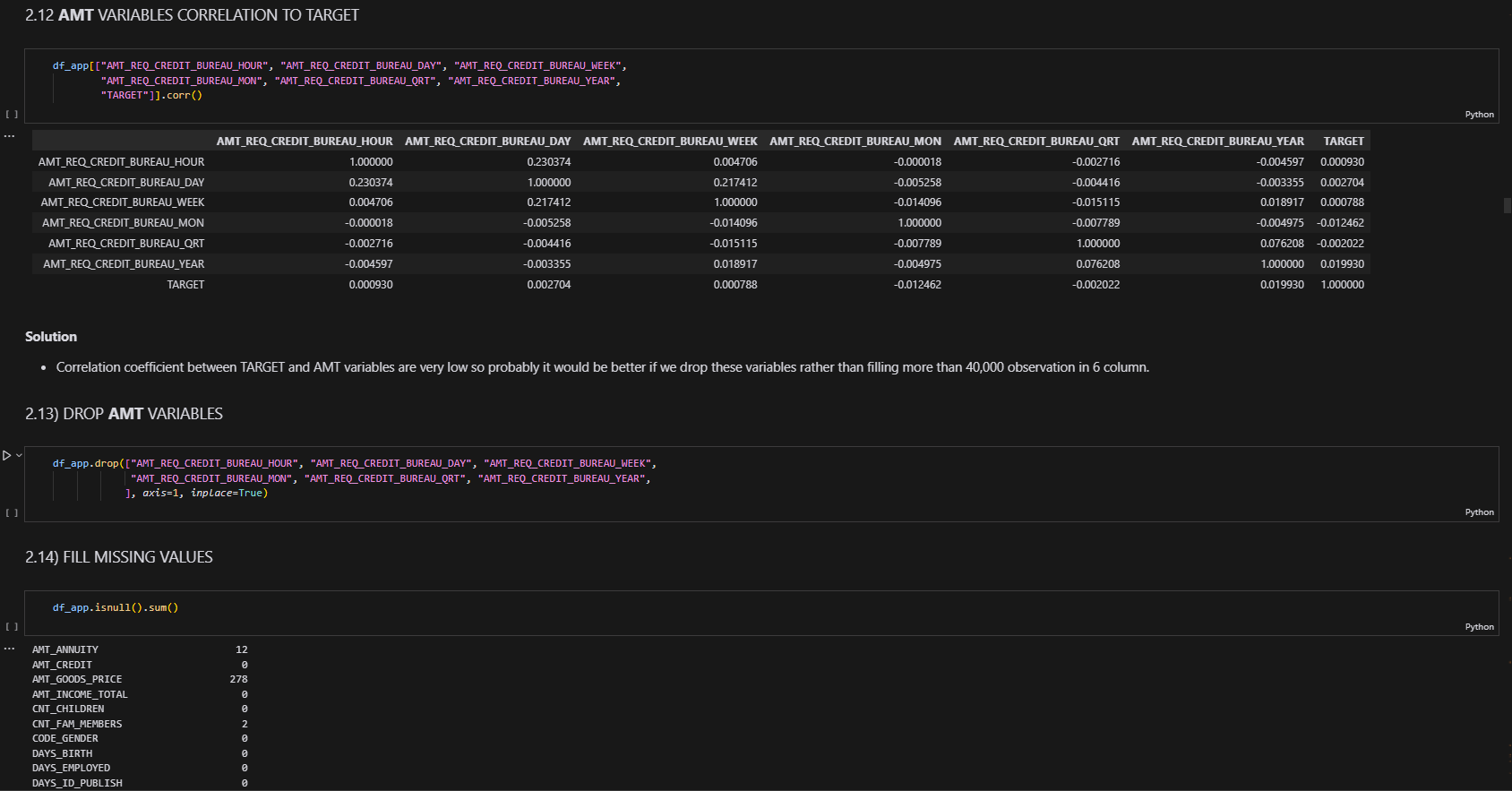
A screenshot of a computer

Description automatically generated

**Checking for Missing Value Graph**

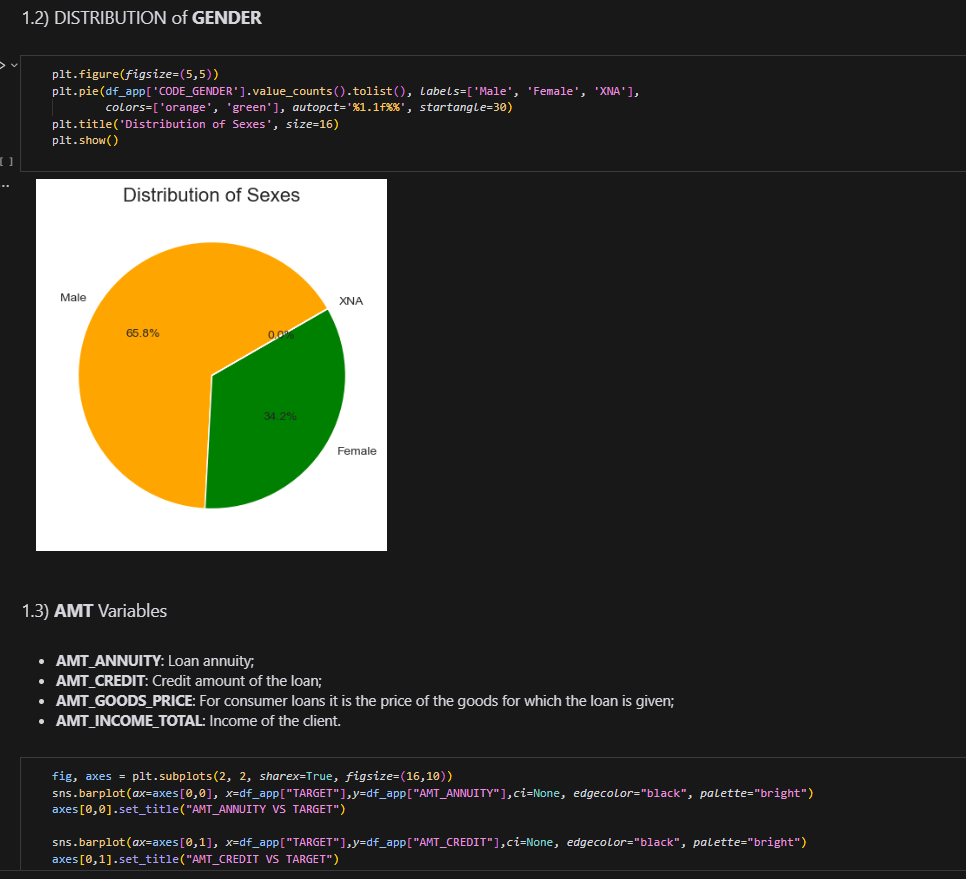


There are features that include a variety of documents that had to be submitted during the application process and now we will analyse these features. For this analysis we have used the ‘itertools’ library and selecting all the documents and checking which of these documents has the least number of defaulters. By this we will understand that this document is important and not many defaulters come up with this document.

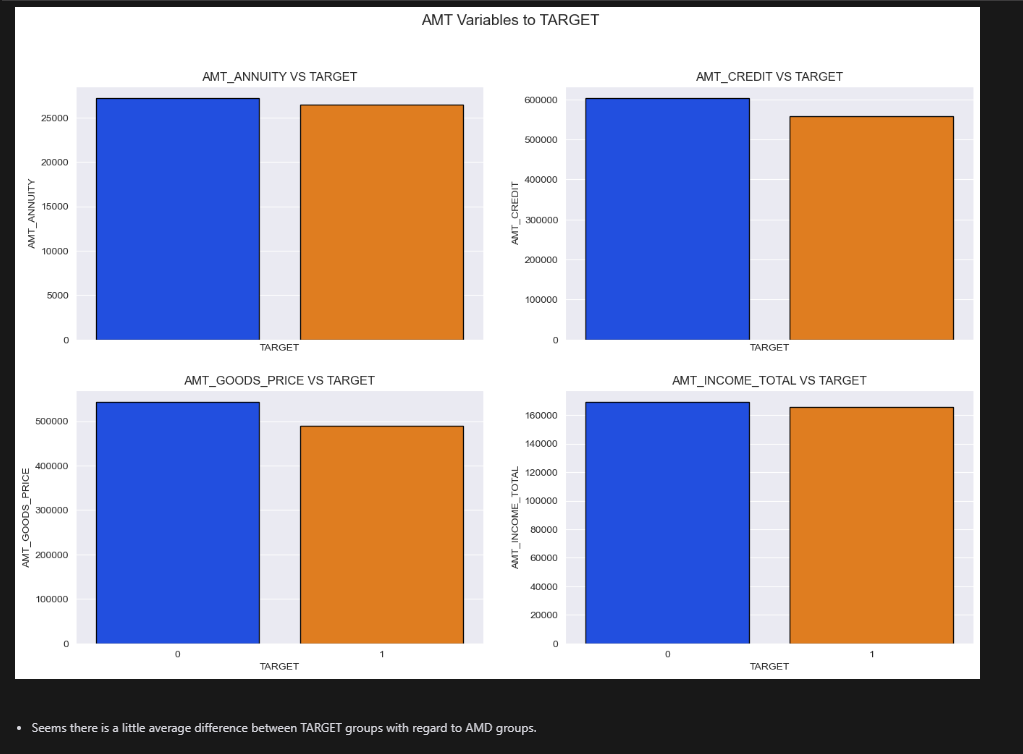




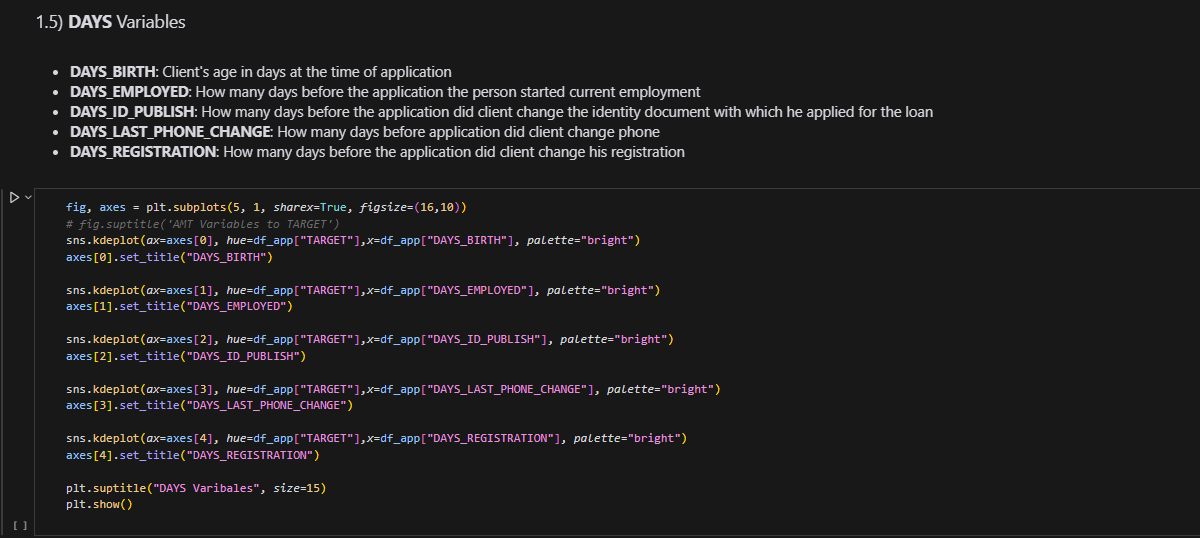
**Shows gender distribution in dataset**

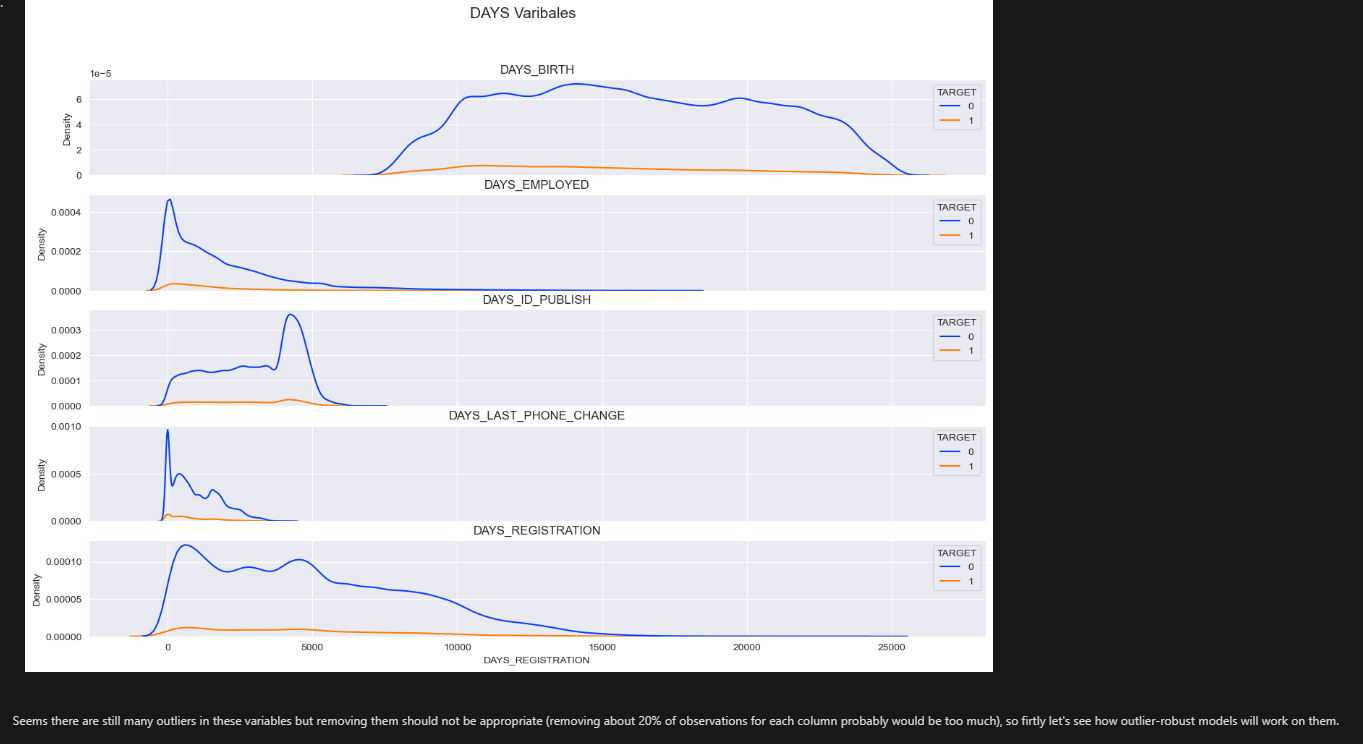


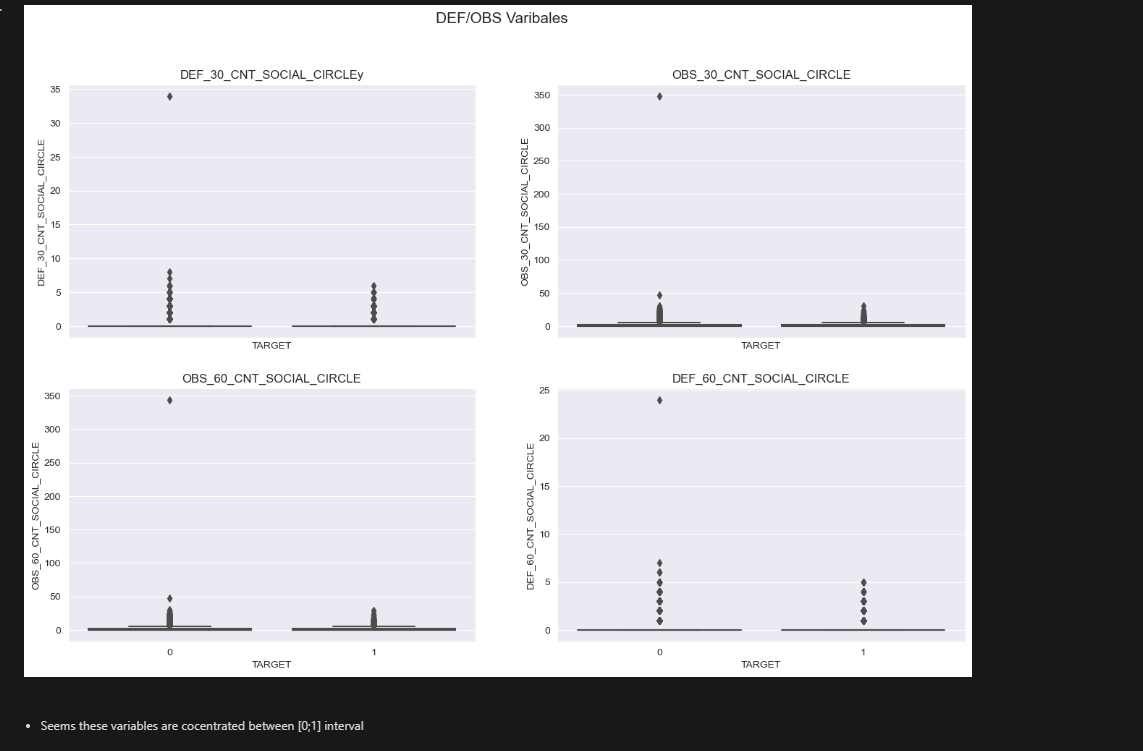
Detecting the outliers for the features like AMT\_INCME\_TOTAL, AMT\_CREDIT, AMT\_ANNUITY, AMT\_GOODS\_PRICE, and analyse the values with other features.



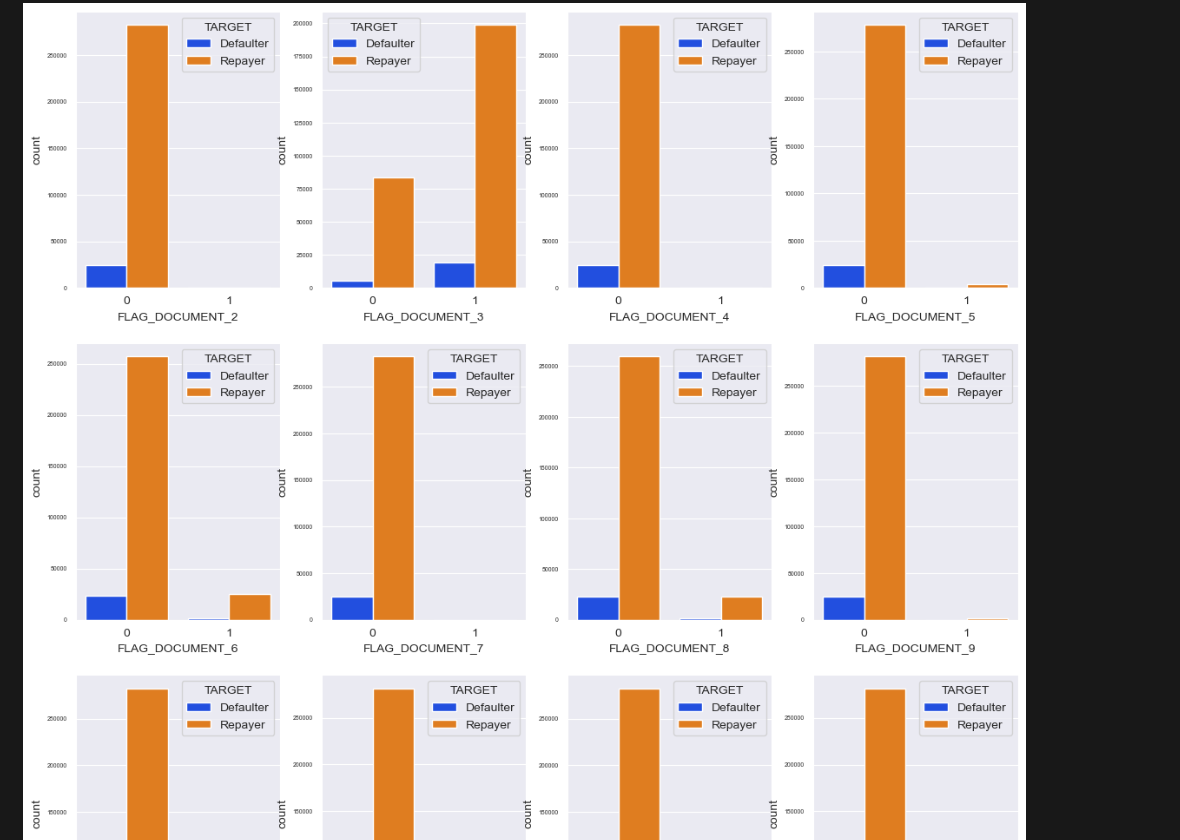
Age, Contract type, Income type, Education type, Family Status, Housing type, Gender and Work Experience.







From the visualization below we can see that the ‘FLAG\_DOCUMENT\_3’ has the least number of defaulters and hence we will remove all the other features.



## Machine learning Models

Logistic Regression :

1.Model Definition: A Logistic Regression model was defined with the 'saga' solver and a maximum iteration limit of 200 to 300. The 'saga' solver is a variant of Stochastic Average Gradient descent and is suitable for large datasets.  
  
2. Hyperparameter Tuning: GridSearchCV was used to find the optimal value for the 'C' parameter, which is the inverse of regularization strength. Regularization is a technique used to prevent overfitting by adding a penalty term to the loss function.  
  
3. Model Training: The model was trained on the resampled training data. Resampling was done using the SMOTE technique to handle class imbalance in the dataset.  
  
4. Model Evaluation: The performance of the model was evaluated using various metrics, including accuracy, precision, recall, and the F1 score. These metrics were calculated for the predictions made on the test data.  
  
5. Threshold Adjustment: The threshold for classifying an instance as class 1 was adjusted to improve the model's performance. The F1 score, accuracy, precision, and recall were calculated for various thresholds to find the optimal threshold.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Decision Tree Classifier:

1. Model Definition: A Decision Tree classifier was defined using the sklearn library.

2. Model Training: The model was trained on the training data. The fit method was used to fit the model to the training data.  
  
3. Model Evaluation: The model's performance was evaluated on both the training and testing data. Metrics such as accuracy, precision, and recall were calculated.  
  
4. Hyperparameter Tuning: GridSearchCV was used to find the optimal hyperparameters for the Decision Tree model. The parameters tuned included 'criterion' (the function to measure the quality of a split), 'max\_depth' (the maximum depth of the tree), 'min\_samples\_split' (the minimum number of samples required to split an internal node), 'min\_samples\_leaf' (the minimum number of samples required to be at a leaf node), and 'max\_features' (the number of features to consider when looking for the best split).  
  
5. Model Evaluation with Best Parameters: The best model obtained from the grid search was evaluated on the test set. The classification report and confusion matrix were printed to assess the model's performance

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

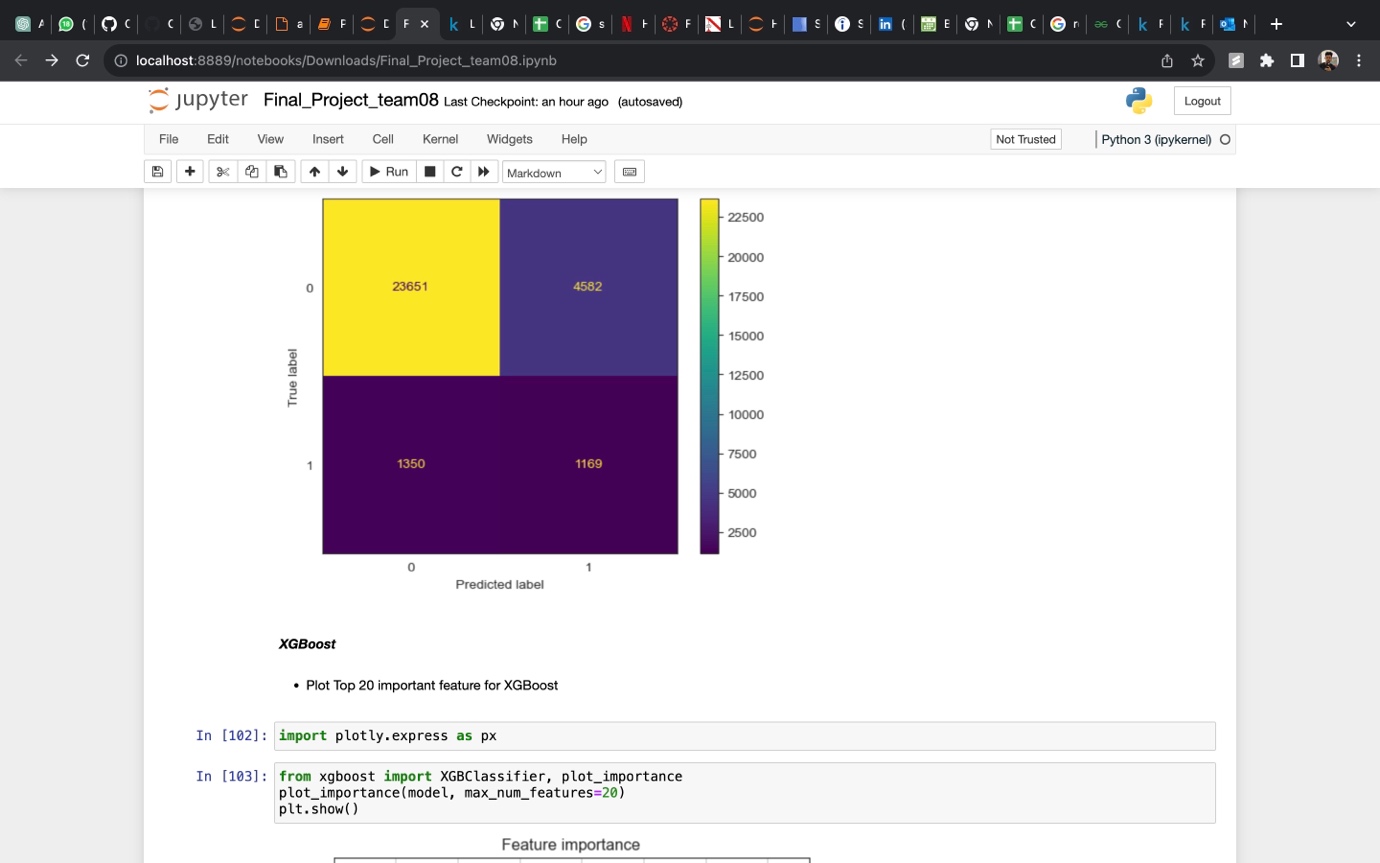
Description automatically generated

XGBoost Model:

1. Model Definition:An XGBoost classifier was defined with specific parameters such as learning rate, max depth, n\_estimators, and others. These parameters control various aspects of the model such as the learning rate, the maximum depth of the trees, and the number of trees to be used, respectively.  
  
2. Model Training: The model was trained on the training data using the fit method.  
  
3. Model Evaluation: The model's performance was evaluated on the test data. Metrics such as accuracy were calculated. Additionally, a classification report was generated to provide detailed performance metrics.  
  
4. Feature Importance: The importance of the different features was visualized using the plot\_importance function from the XGBoost library. This function plots the importance of features based on the fitted model.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

K-Nearest Neighbours:

1. Data Pre-processing: The KNN algorithm requires all input data to be numerical and not contain any missing values. Therefore, any missing values in the dataset were filled using the SimpleImputer.  
  
2. Model Definition: A KNN classifier was defined using the sklearn library. The number of neighbors to consider (n\_neighbors) was set to 5.  
  
3. Model Training: The model was trained on the training data using the fit method.  
  
4. Model Evaluation: The model's performance was evaluated on the test data. Metrics such as accuracy were calculated. Additionally, a classification report and confusion matrix were generated to provide detailed performance metrics.  
  
5. Hyperparameter Tuning: RandomizedSearchCV was used to find the optimal hyperparameters for the KNN model. The parameters tuned included 'n\_neighbors', 'weights', 'algorithm', and 'p'.  
  
6. Model Evaluation with Best Parameters: The best model obtained from the randomized search was evaluated on the test set. The classification report and confusion matrix were printed to assess the model's performance

A screenshot of a computer

Description automatically generatedA screenshot of a computer

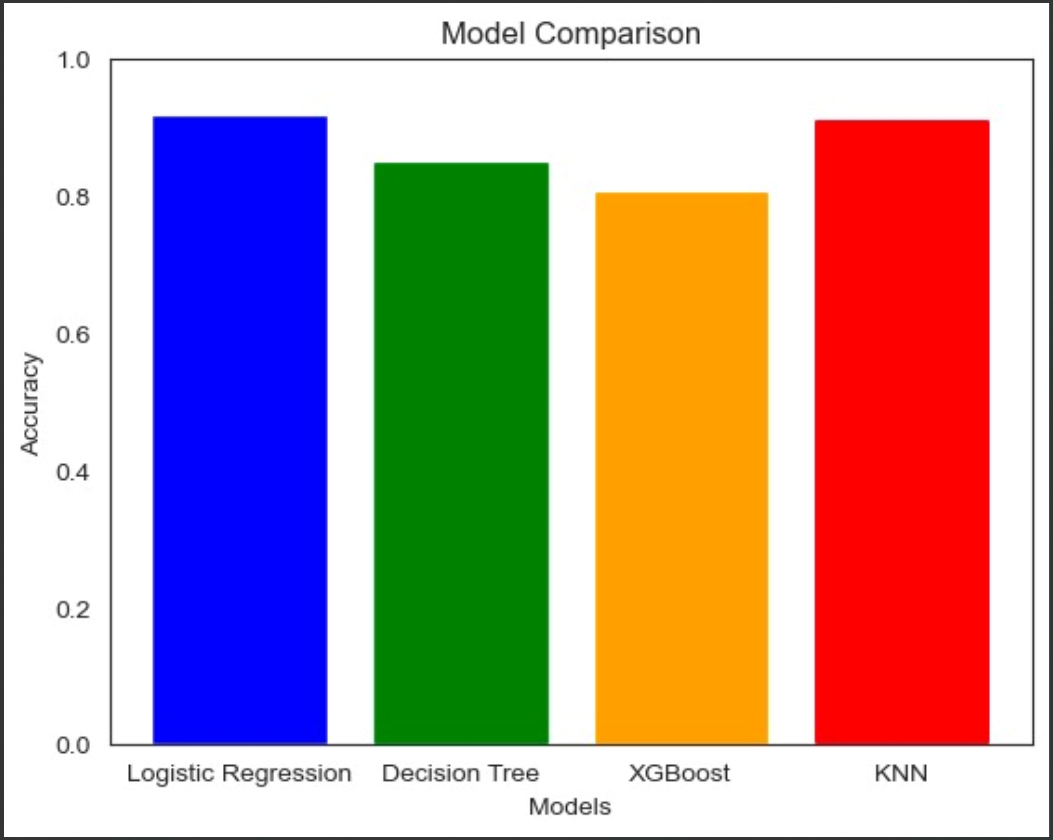
Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

# Model Comparison



* The Logistic Regression model had a high overall accuracy, but it struggled to correctly identify instances of class 1 (defaulters).
* Decision Trees can overfit the training data leading to lower performance on the test data.
* The XGBoost model was evaluated based on its accuracy and the detailed classification report, due to the low accuracy score could not be the best model
* The KNN model was evaluated based on its accuracy and the detailed classification report turned out to be the best overall performing model between all of the other models, We can choose KNN Model after comparison with all other models because the Accuracy is 92% (Model predicts costumers status with 92% accuracy), F-1 Score: 96% (Model can classify Positive or Negative accurately )

# Conclusion

Finally, our machine learning project focused on developing a robust loan eligibility assessment system using a large dataset containing critical information about borrowers. The strategic use of various Python libraries, such as NumPy for numerical operations, Pandas for efficient data manipulation, Matplotlib and Seaborn for insightful data visualization, and Scikit-Learn for implementing machine learning models, was critical to the success of our project.

We meticulously processed the raw data throughout the project, extracting and engineering features that were critical in determining loan eligibility. This meticulous feature engineering aims to provide rich and relevant information to our machine learning models—Logistic Regression, XGBoost, Decision Tree and KNN—in order for them to make accurate predictions.

KNN Model identifies most of the Defaulters but with a cost of non-default costumers: only 10% of predicted Defaulters will be actually defaulter, Model identifies only 10% of Defaulters but on the other hand it won't lose as much non-default costumers as in Case I if we go with Logistic Regression.

We were able to investigate and compare the performance of various machine learning models, providing insights into which algorithms were most effective for our specific use case. The Logistic Regression, Random Forest, and Decision Tree models were chosen for their suitability for binary classification tasks, with each contributing unique strengths to our system's overall predictive power.

The machine learning models demonstrated commendable accuracy and efficiency in evaluating loan eligibility as a result of our collaborative efforts. The combination of these models, combined with the extensive feature engineering process, allowed us to develop a dependable and versatile system capable of assisting financial institutions in making informed decisions about borrower eligibility.

# References

* <https://www.forbes.com/sites/robertfarrington/2018/09/25/student-loan-defaulters-strategic-default/?sh=43370930229c>
* <https://www.businessinsider.com/personal-finance/how-long-to-get-a-student-loan>
* <https://www.kaggle.com/code/gauravduttakiit/risk-analytics-in-banking-financial-services-1>
* [https://www.learndatasci.com/glossary/binary-classification/#:~:text=each%20binary%20classifier-,What%20is%20Binary%20Classification%3F,Application](file:///Users/anshulsingh/Downloads/%20https:/www.learndatasci.com/glossary/binary-classification)
* <https://www.mathworks.com/campaigns/offers/next/choosing-the-best-machine-learning-classification-model-and-avoiding-overfitting.html>