

# Credit Score Classification

## Table of Contents

### 1. Executive Summary

- Brief overview of the project, its goals, and key findings.

### 2. Introduction

- Background information on the credit score classification problem.
- Importance of credit scoring in financial decision-making.

### 3. Data Understanding and Preprocessing

- Description of the dataset and its attributes.
- Data preprocessing
- Insights from data preprocessing

### 4. Exploratory data analysis (EDA)

- Univariate Analysis
- Bivariate Analysis
- Multivariate Analysis

### 5. Feature Engineering

- Removing Unnecessary Columns
- Transforming Skewed Data
- Creating Derived Features
- Encoding Categorical Variables

### 6. Model Preprocessing:

- Data Splitting
- User defined function:
  - Model evaluation
  - Plotting ROC-AUC curve
  - Adding results in scorecard
- Benefits of User-Defined Functions

### 7. Model Development

- Logistic Regression
- Decision Trees (with and without pruning)
- Random Forest
- Random Forest (with and without pruning)

- Bagging Ensemble
- Adaboost Ensemble
- XG Boost Ensemble
- Naive Bayes Classifier
- k-Nearest Neighbors (KNN)

## 8. Model comparison with Scorecard

- Presentation of evaluation results for each model.

## 9. Scope of Improvement

- Robust Feature Engineering
- Model Selection and Utilization
- Interpretable Ensemble Models
- Ongoing Model Monitoring and Updates
- Collaboration with Domain Experts
- Documentation and Transparency

## 10. References

# Executive Summary

The "Credit Score Classification" project aims to develop and evaluate machine learning models for the classification of credit scores into three classes: Poor, Standard, and Good. Credit scoring plays a crucial role in financial decision-making, assisting lenders in assessing the creditworthiness of individuals and businesses. This report provides a comprehensive overview of the project's objectives, methodologies, findings, and implications.

## Project Overview

In this project, a diverse dataset containing various customer attributes was collected and preprocessed to create a robust foundation for building predictive models. The dataset includes features such as customer demographics, financial information, credit history, and payment behavior.

## Methodology

The project's methodology involved several key steps:

- **Data Collection and Preprocessing:** The dataset was carefully curated, and preprocessing techniques were applied to handle missing values, encode categorical variables, and ensure data quality. Exploratory data analysis (EDA) provided insights into the distribution and characteristics of the data.
- **Feature Engineering:** Feature selection methods were employed to identify the most relevant features for model development.
- **Model Development:** A range of machine learning models were selected and trained to classify credit scores. These models include Logistic Regression, Decision Trees, Random Forest, Ensemble methods (Bagging, Adaboost, XG Boost), Naive Bayes Classifier, and k-Nearest Neighbors (KNN). Hyperparameter tuning was performed to optimize model performance.
- **Evaluation and Analysis:** The models were evaluated using appropriate multi-class classification metrics such as F1-score, accuracy, and confusion matrices. Performance was analyzed across all three credit score classes to understand the models' strengths and limitations.

## Key Findings

The Random Forest model consistently demonstrated the highest performance across all evaluation metrics, effectively classifying credit scores into the Poor, Standard, and Good categories.

Feature importance analysis revealed that attributes related to credit history, payment behavior, and financial stability played significant roles in determining credit score classification.

The Adaboost and XG Boost ensemble methods also showed competitive performance, suggesting that boosting techniques can enhance model accuracy.

## **Implications**

The successful development of accurate credit score classification models has significant implications for the financial industry. Lenders and financial institutions can benefit from improved credit scoring accuracy, enabling them to make more informed lending decisions and effectively manage risk. This project contributes to enhancing the efficiency and fairness of credit assessments.

## **Limitations and Future Directions**

While this project provides valuable insights, it is essential to acknowledge some limitations. The model's performance is contingent on the quality and representativeness of the training data. Future work could involve incorporating external data sources and refining feature engineering strategies like PCA to further enhance predictive accuracy.

Overall, this report offers a detailed examination of the credit score classification process, showcasing the capabilities of various machine learning techniques and their potential impact on financial decision-making.

# Introduction

The Credit Score Classification project addresses a critical aspect of modern financial systems – the accurate assessment of creditworthiness. Credit scoring is a fundamental process used by lenders, banks, and financial institutions to evaluate the risk associated with extending credit to individuals or businesses. It plays a pivotal role in determining interest rates, loan approvals, and credit limits, thereby influencing various financial decisions that impact individuals' lives and business operations.

## Background

As lending practices have evolved, credit scoring has become a cornerstone of prudent financial management. Traditionally, credit assessments were often based on subjective judgments, leading to inconsistencies and potential biases. The advent of data-driven techniques and machine learning has transformed the credit scoring landscape, enabling more accurate, consistent, and objective evaluations.

## Importance of Credit Scoring

The importance of credit scoring cannot be overstated. A robust credit scoring system offers several benefits:

- **Risk Mitigation:** Lenders need to assess the risk associated with lending money. Accurate credit scores help lenders identify individuals or businesses with a higher likelihood of defaulting on their loans.
- **Fairness and Objectivity:** Data-driven credit scoring reduces human bias and subjectivity, promoting fair treatment for borrowers from diverse backgrounds.
- **Efficiency:** Automated credit scoring processes streamline lending decisions, leading to faster approvals and disbursements.
- **Personal Financial Management:** Credit scores also influence individuals' access to credit cards, mortgages, and other financial products. A clear understanding of credit scores empowers individuals to make informed financial choices.
- **Economic Impact:** Effective credit scoring contributes to economic stability by managing credit risks and reducing the chances of financial crises.

## Project Scope

This project focuses on the classification of credit scores into three categories: Poor, Standard, and Good. The goal is to develop machine learning models that can accurately predict the credit score class based on various customer attributes. The dataset encompasses a wide range of features, including demographics, financial history, payment behavior, and credit utilization.

## Report Structure

This report provides an in-depth analysis of the credit score classification project. It details the data understanding, preprocessing steps, feature engineering techniques, model development, evaluation

metrics, and model performance. The insights derived from this project contribute to advancing credit scoring methodologies, ultimately benefiting both lenders and borrowers.

In the following sections, we delve into the specific steps undertaken during this project, showcasing the methodologies, findings, and recommendations for further improvements.

# Data Understanding and Preprocessing

## Dataset Description

The dataset used for this project comprises a diverse collection of customer attributes relevant to credit score classification. The attributes include both numerical and categorical variables, offering insights into customers' demographics, financial behavior, and credit history. A brief overview of the key attributes is provided below:

- ID: Unique identifier for each record.
- Customer\_ID: Identifier for individual customers.
- Month: Month of the data entry.
- Name: Customer's name.
- Age: Age of the customer.
- SSN: Social Security Number of the customer.
- Occupation: Customer's occupation.
- Annual\_Income: Annual income of the customer.
- Monthly\_Inhand\_Salary: Monthly take-home salary of the customer.
- Num\_Bank\_Accounts: Number of bank accounts held by the customer.
- Num\_Credit\_Card: Number of credit cards held by the customer.
- Interest\_Rate: Interest rate associated with loans.
- Num\_of\_Loan: Number of loans taken by the customer.
- Type\_of\_Loan: Type of loan (e.g., personal, mortgage).
- Delay\_from\_due\_date: Delay in payments from the due date.
- Num\_of\_Delayed\_Payment: Number of delayed payments.
- Changed\_Credit\_Limit: Whether the credit limit has changed recently.
- Num\_Credit\_Inquiries: Number of credit inquiries made by the customer.
- Credit\_Mix: Diversity of credit accounts (e.g., credit cards, loans).
- Outstanding\_Debt: Amount of outstanding debt.
- Credit\_Utilization\_Ratio: Ratio of credit used to credit available.
- Credit\_History\_Age: Age of the customer's credit history.
- Payment\_of\_Min\_Amount: Payment of minimum amount due.
- Total\_EMI\_per\_month: Total Equated Monthly Installments (EMIs) paid.
- Amount\_invested\_monthly: Amount invested monthly.
- Payment\_Behaviour: Behavior of payments (e.g., on-time, delayed).
- Monthly\_Balance: Monthly account balance.
- Credit\_Score: Target variable representing the credit score class (Poor, Standard, Good).

## Data Preprocessing Steps

### Handling Missing Values:

- Identify and quantify missing values in each column.

- Applying different approach for numeric or categorical missing values with mode using user defined function to fill values.

#### **Handling Duplicates:.**

- Check if the presence of duplicates is accidental or if it represents valid data repetitions.
- Detect and remove duplicated rows to avoid skewing analysis or modeling results.
- Examine the criteria for duplicates; are they based on the entire row or specific columns?

#### **Data Type Consistency:**

- Ensure that data types are consistent within columns. For example, numerical columns should contain only numerical values, while categorical columns should contain discrete categories.
- Convert data types as needed to match the expected format for analysis and modeling.

## **Insights from Preprocessing**

The preprocessing steps ensured that the dataset was suitable for model development. Missing values were addressed to avoid introducing bias into the models. Categorical variables were appropriately transformed to numerical representations.

In the following sections, we delve into the Exploratory data analysis to visualize the dataset.

In the following sections, we delve into feature engineering techniques employed to enhance the dataset for model development.

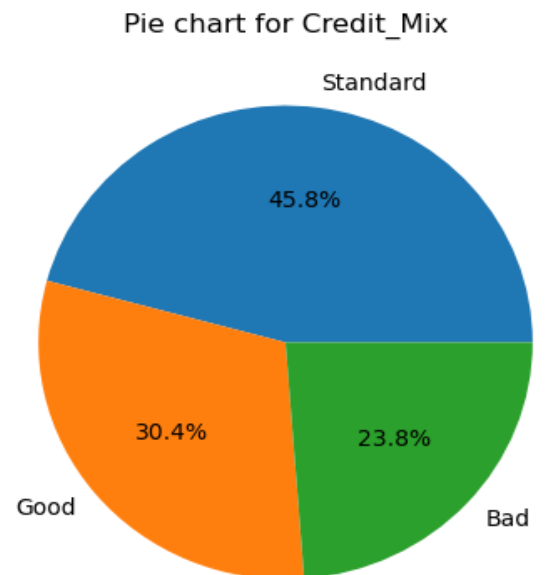
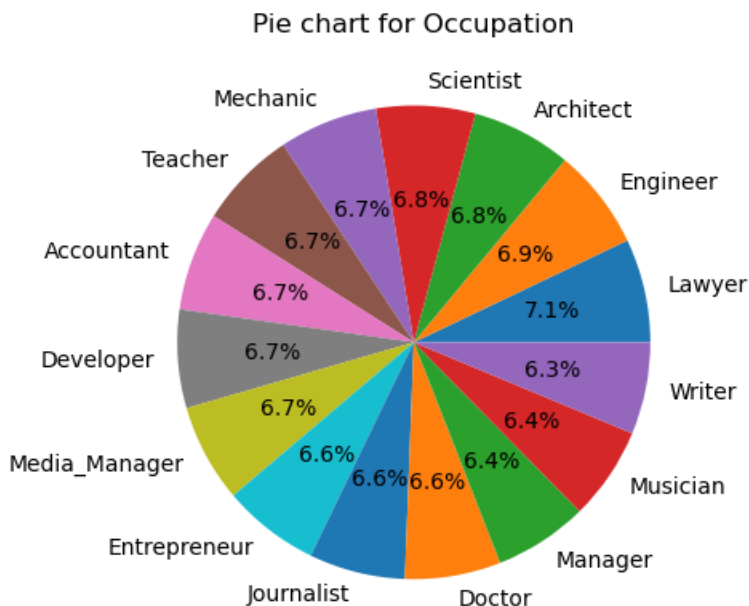
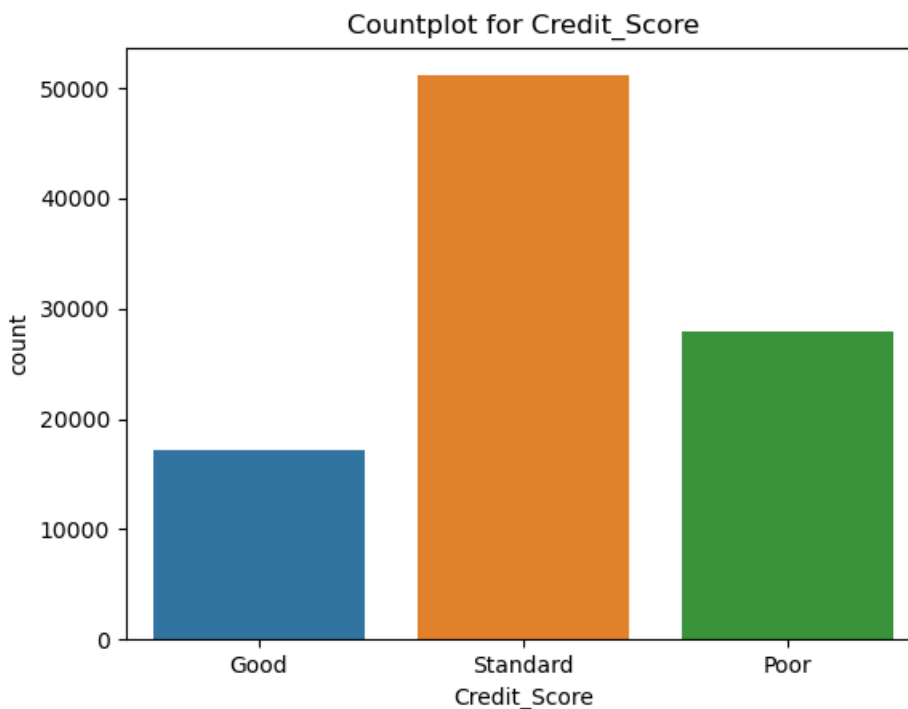


# Exploratory Data Analysis (EDA)

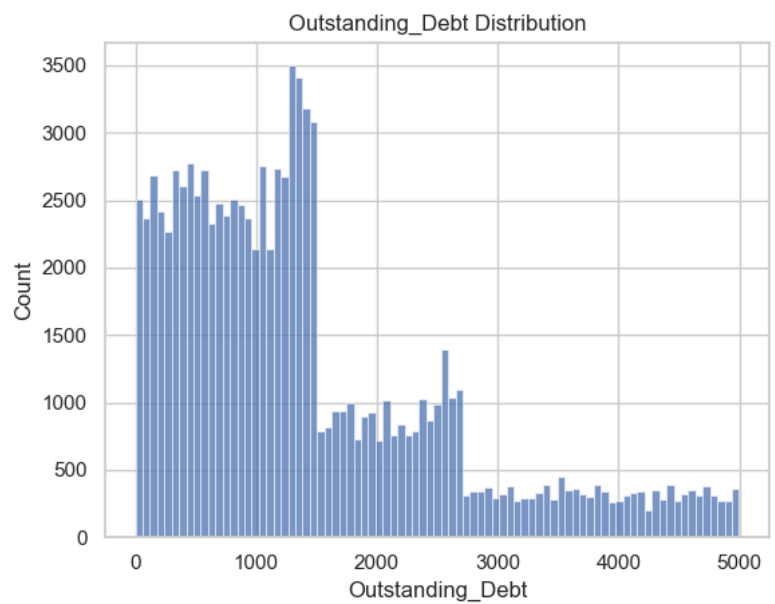
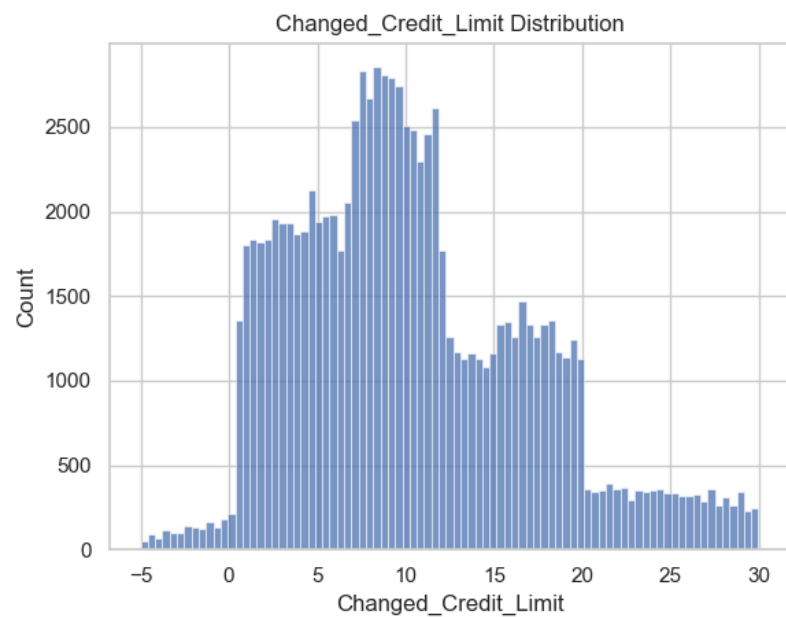
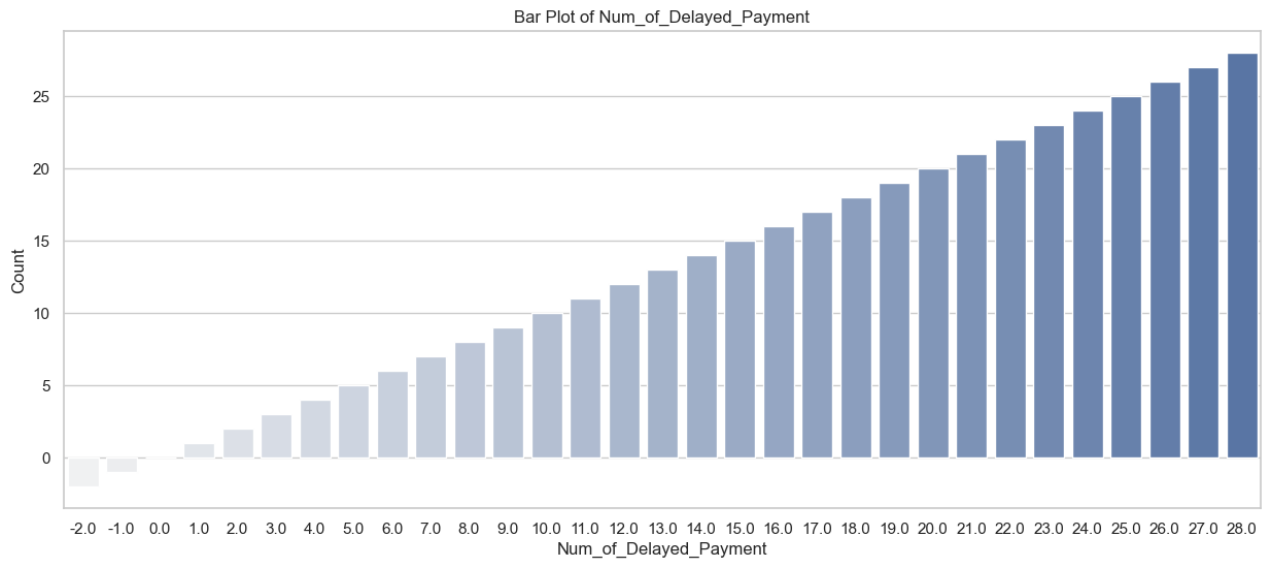
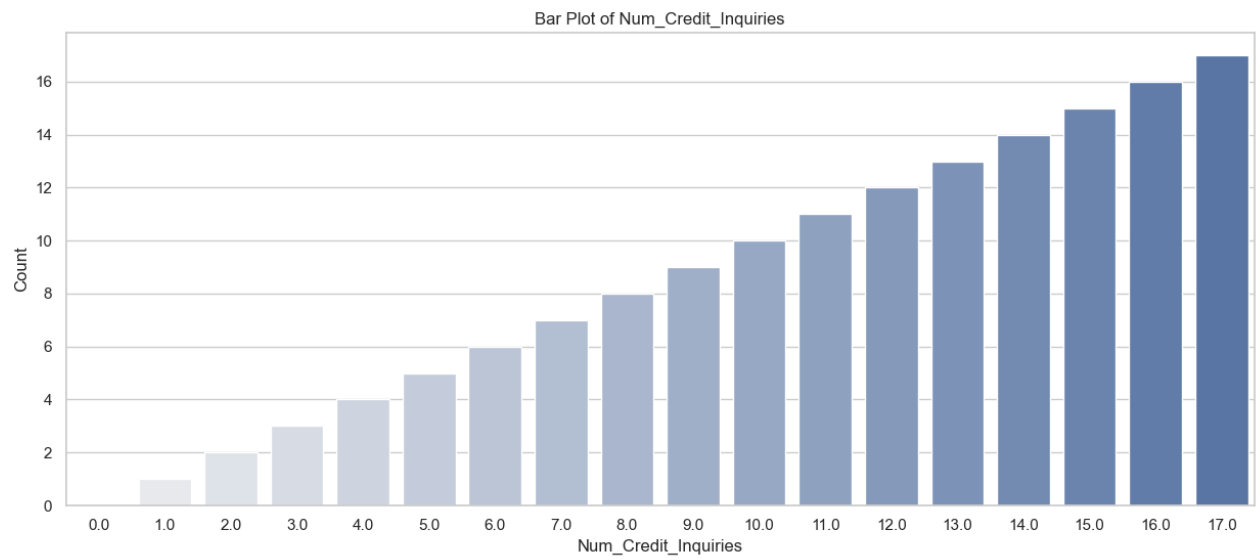
Exploratory Data Analysis (EDA) is a crucial step in understanding the underlying structure and characteristics of the dataset. It involves visualizing and summarizing the data to gain insights into patterns, relationships, and potential areas of interest. EDA serves as a foundation for making informed decisions about feature engineering, model selection, and preprocessing strategies.

## Univariate Analysis

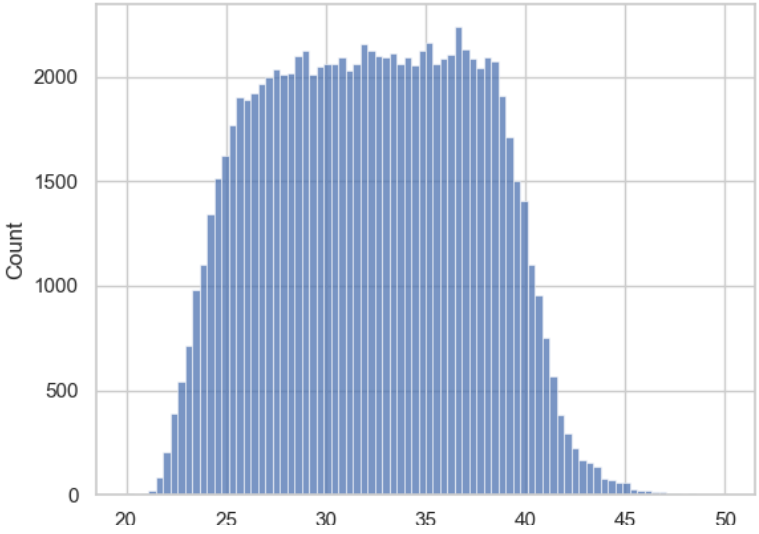
Univariate analysis focuses on exploring individual variables in isolation. This analysis helps uncover distributions, central tendencies, spread, and potential outliers within each feature.



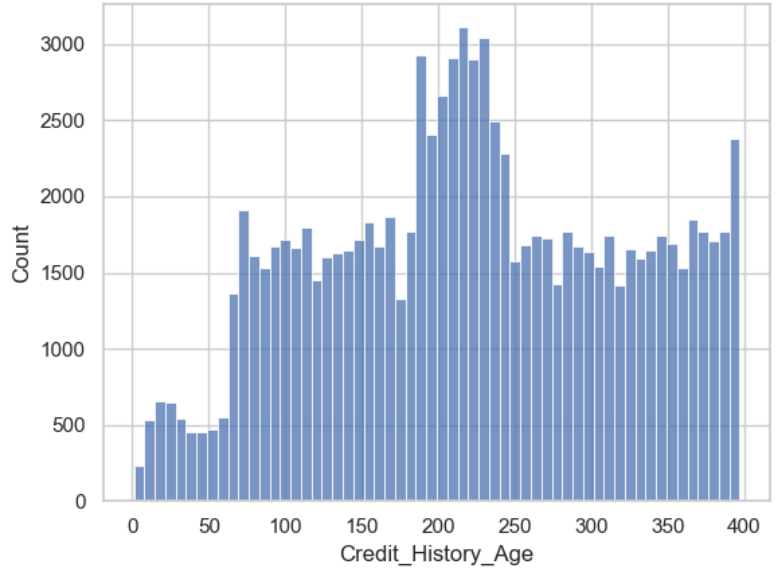




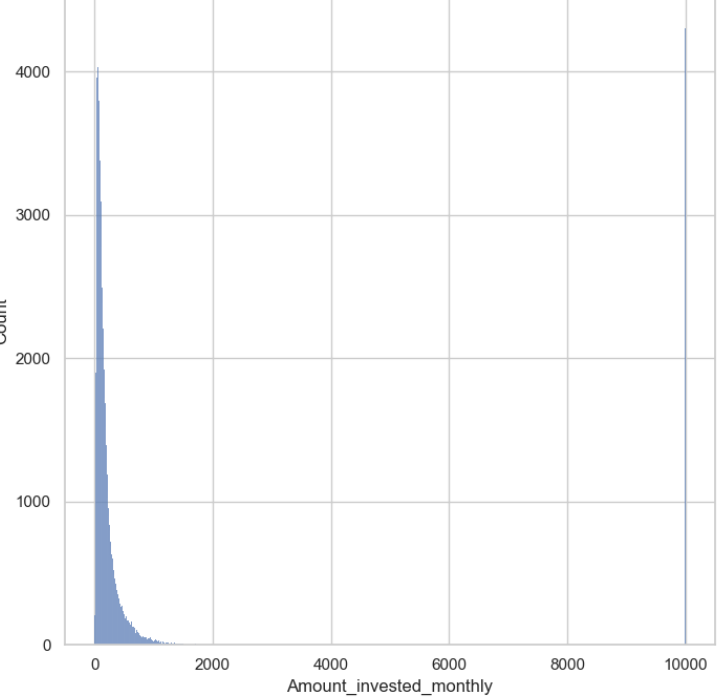
Credit\_Utilization\_Ratio Distribution



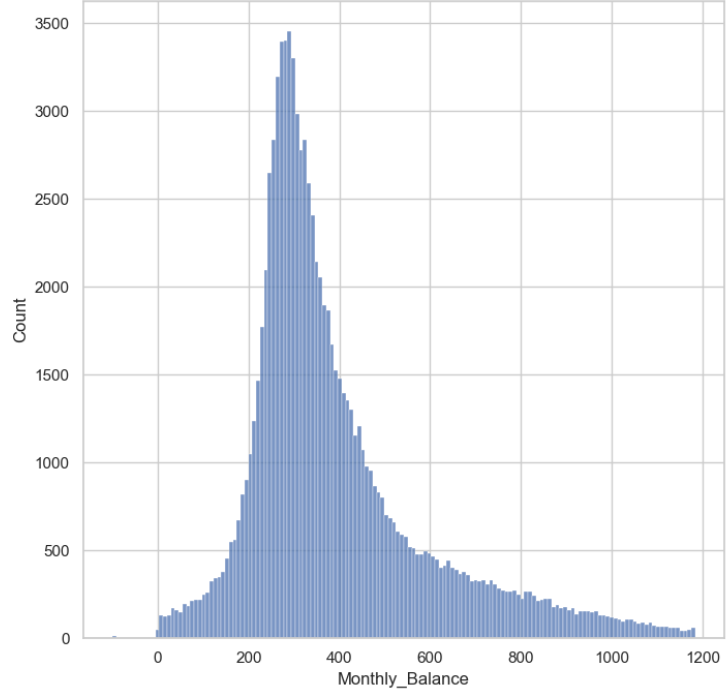
Credit\_History\_Age Distribution



Amount\_invested\_monthly Distribution

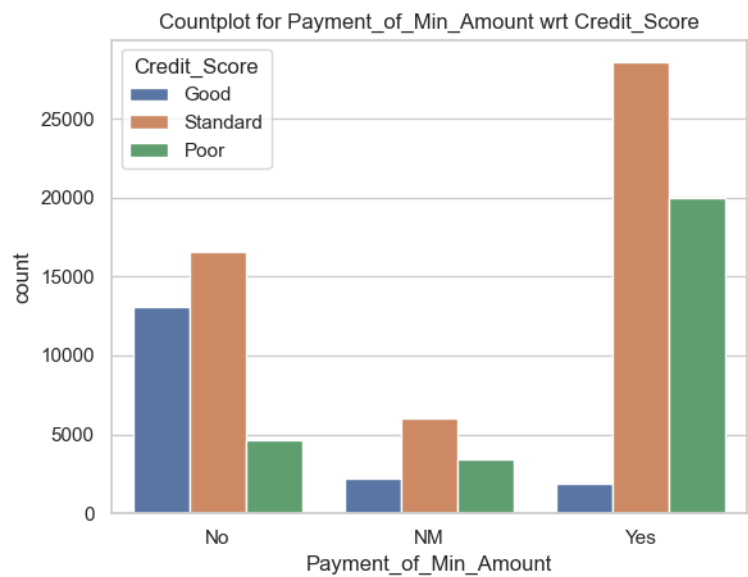
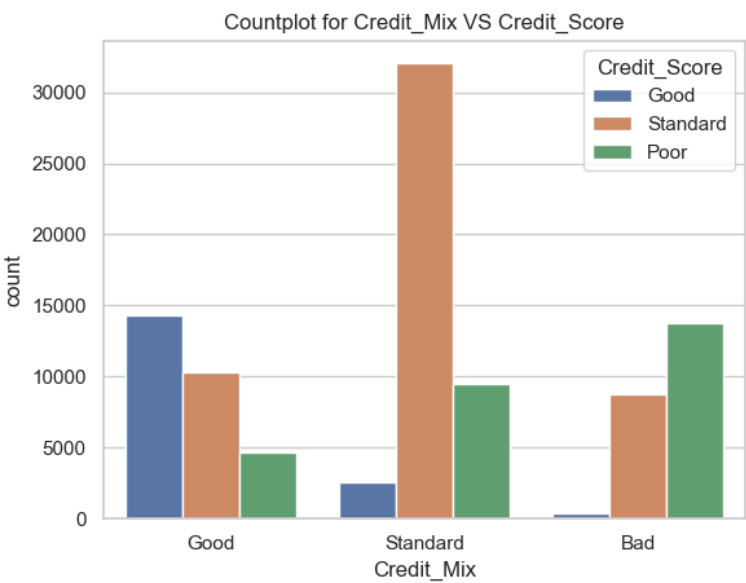
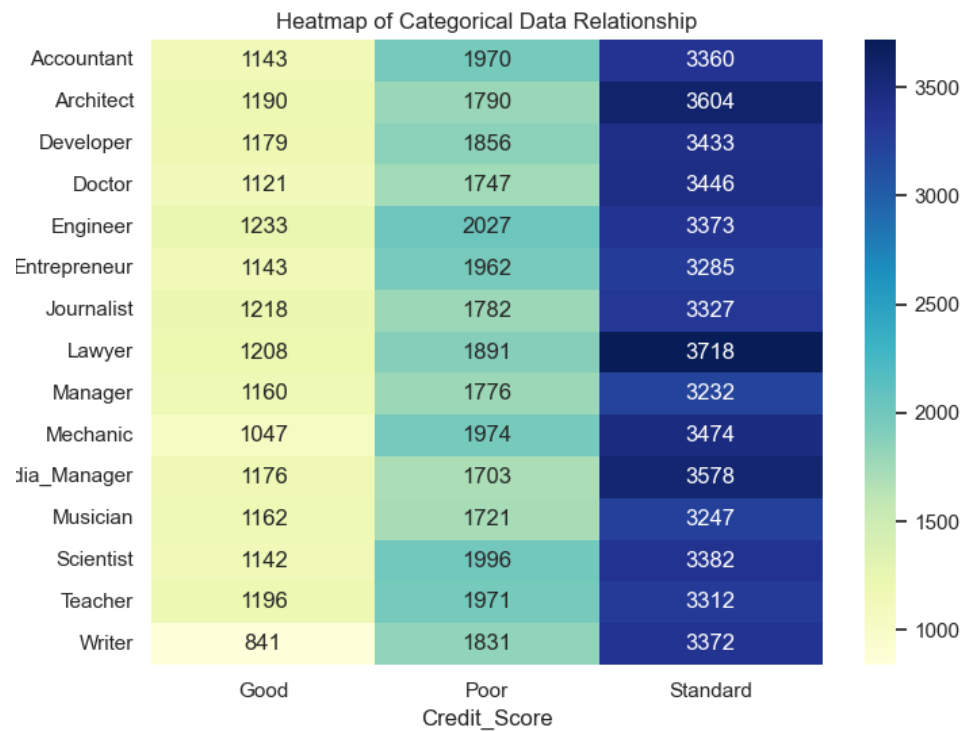


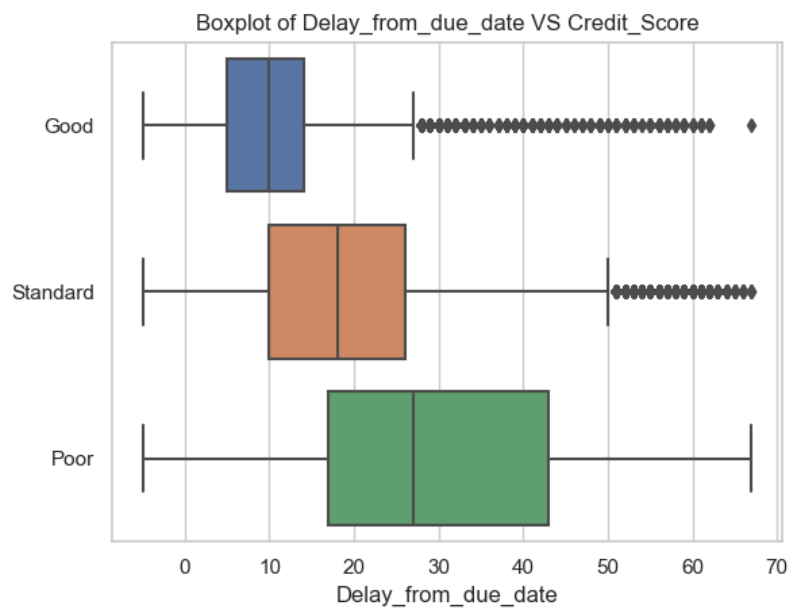
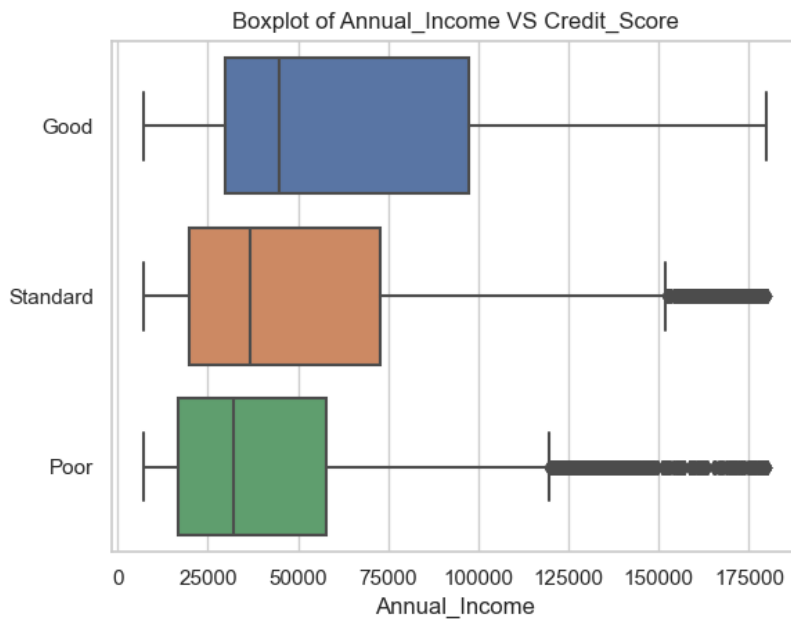
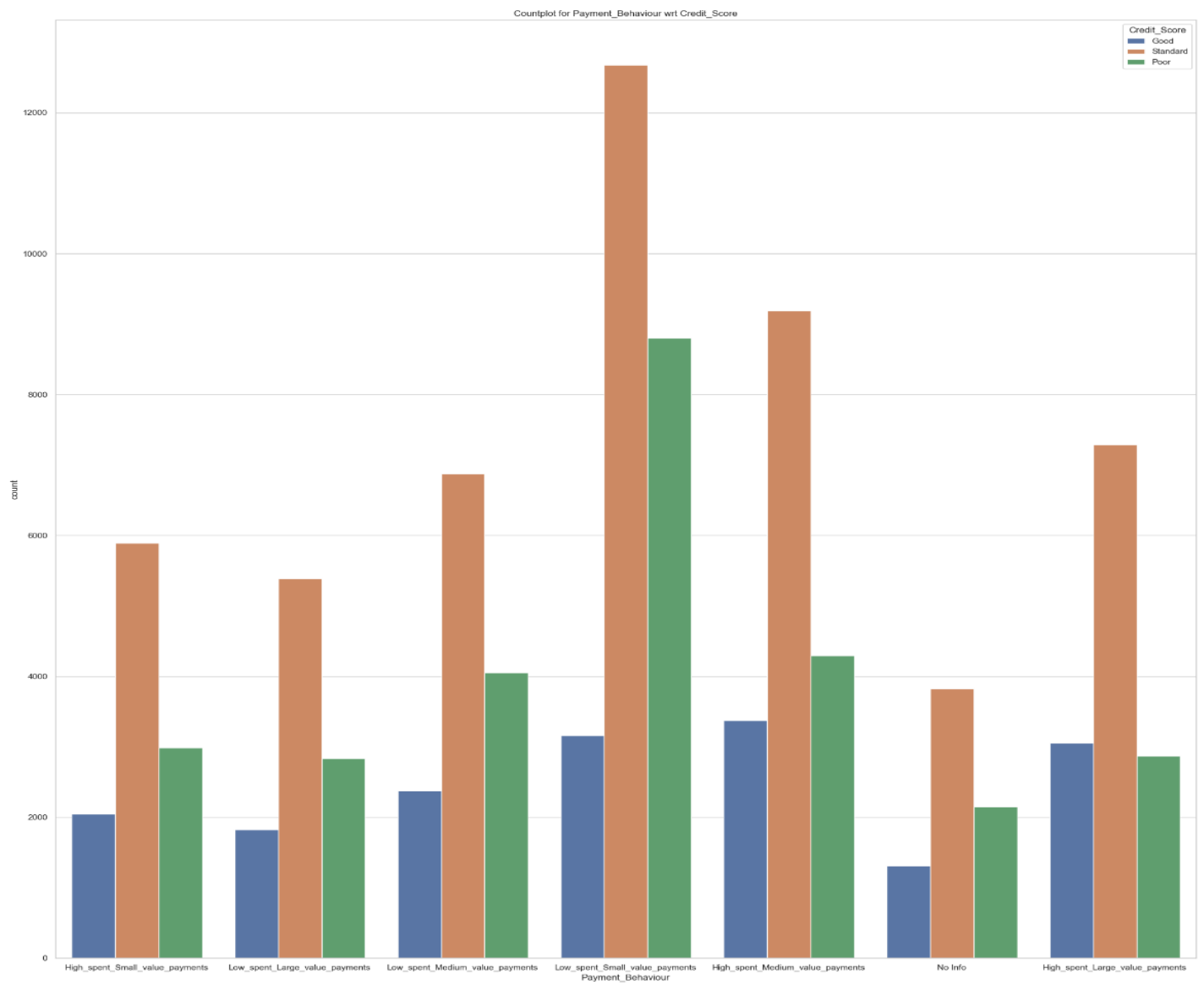
Monthly\_Balance Distribution



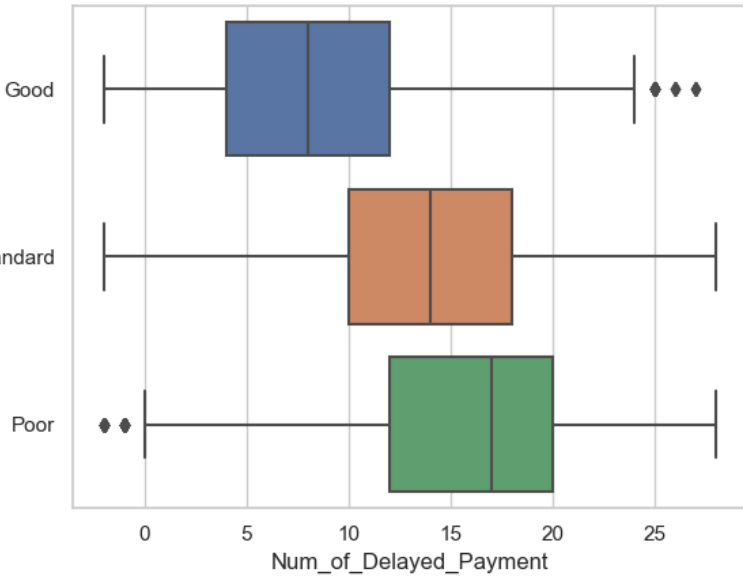
# Bivariate Analysis

Bivariate analysis involves studying relationships between pairs of variables. This analysis helps uncover correlations, dependencies, and potential associations between features.

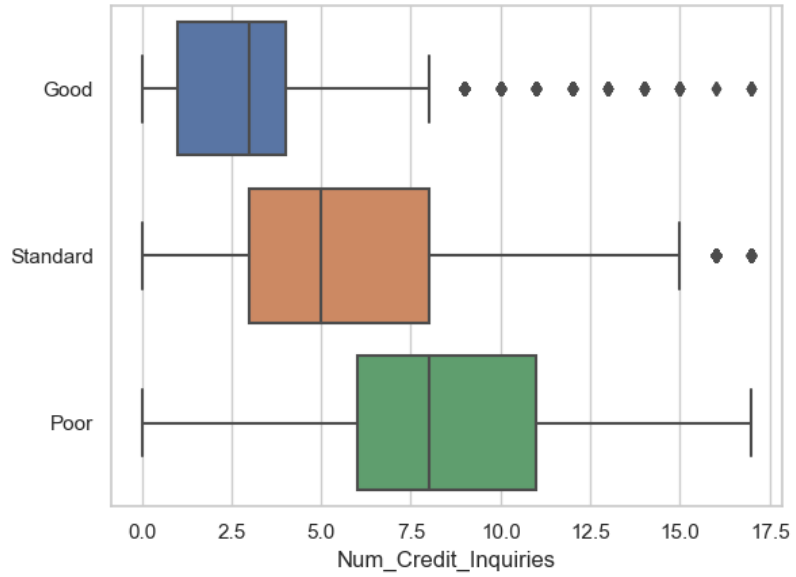




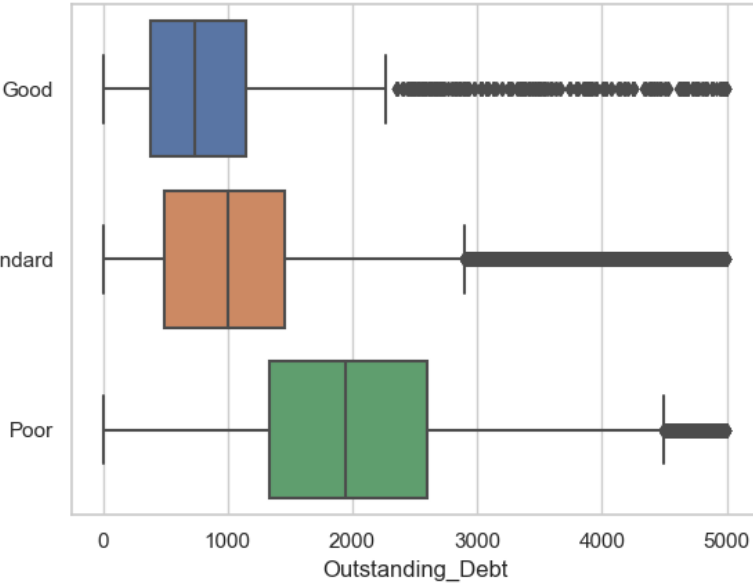
Boxplot of Num\_of\_Delayed\_Payment VS Credit\_Score



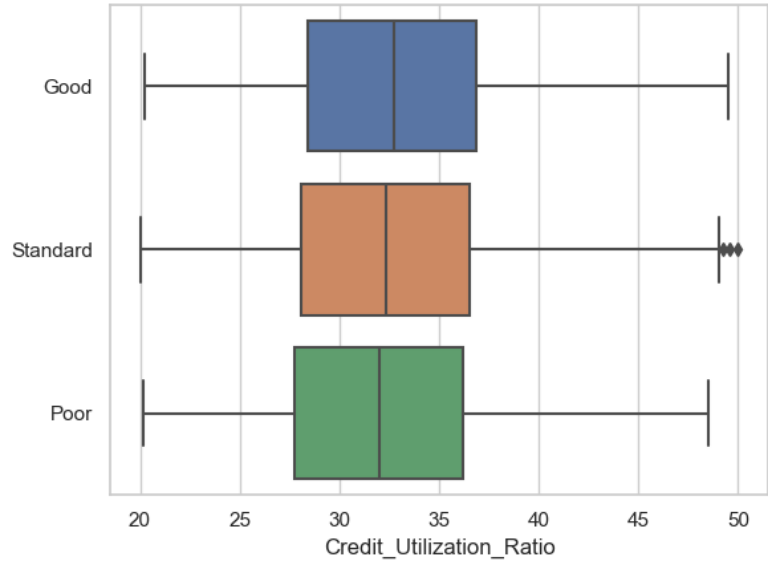
Boxplot of Num\_Credit\_Inquiries VS Credit\_Score



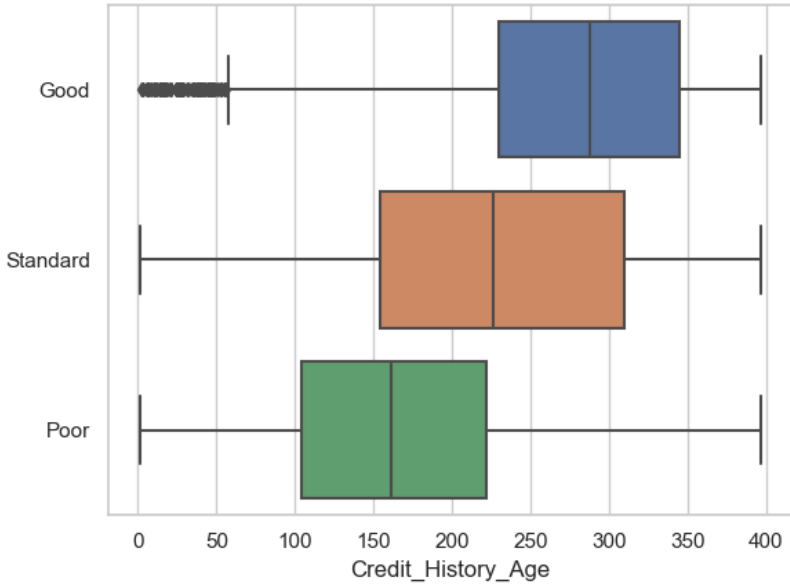
Boxplot of Outstanding\_Debt VS Credit\_Score



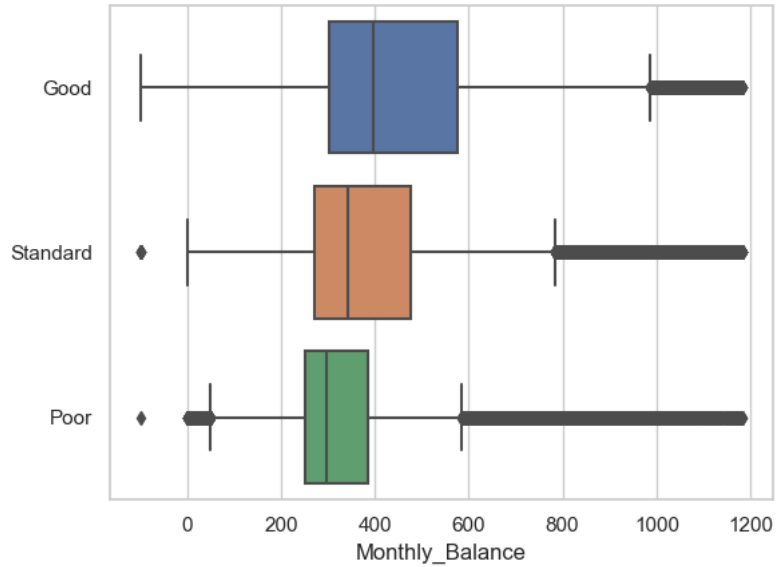
Boxplot of Credit\_Utilization\_Ratio VS Credit\_Score



Boxplot of Credit\_History\_Age VS Credit\_Score

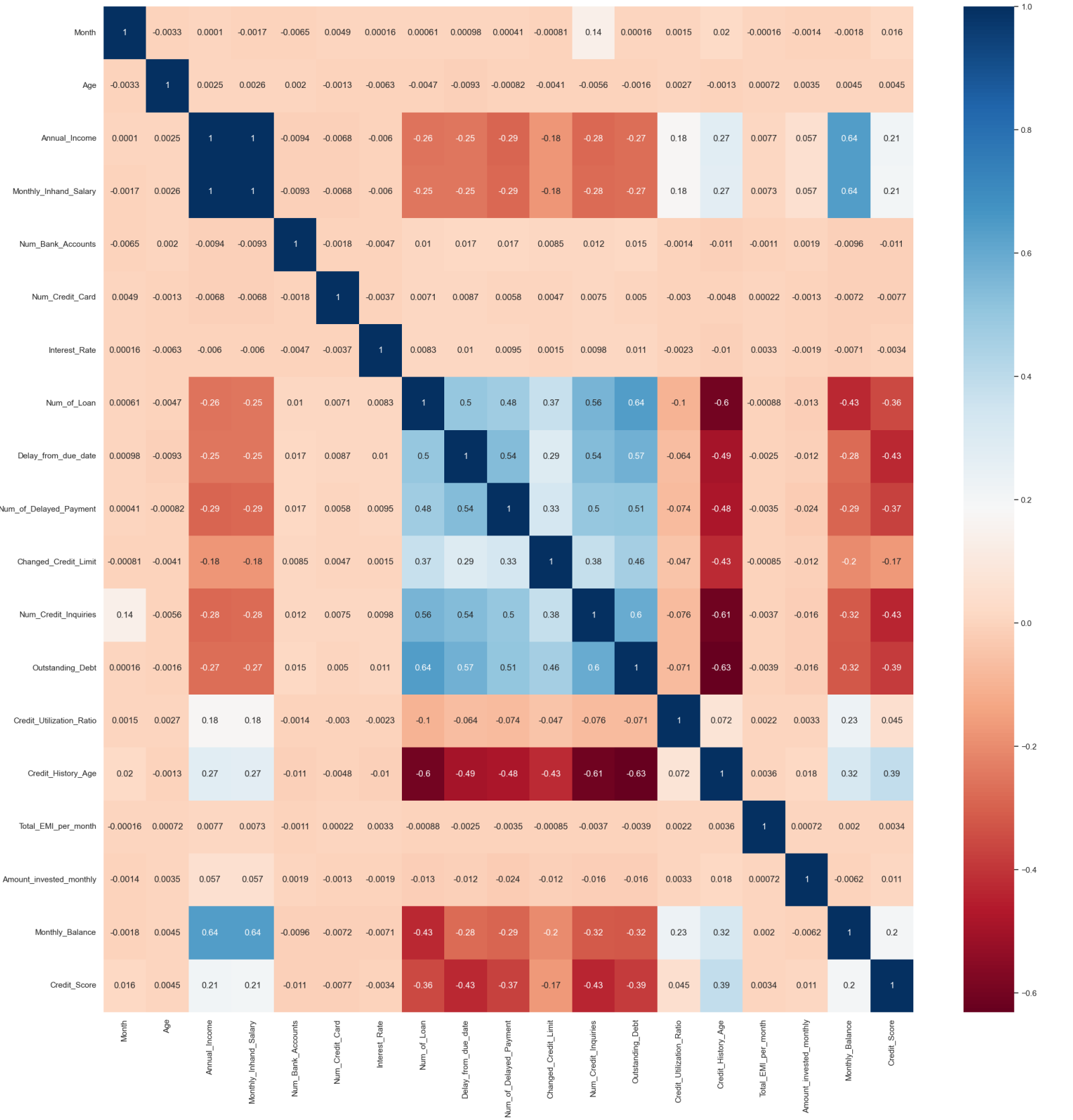


Boxplot of Monthly\_Balance VS Credit\_Score



# Multivariate Analysis

Multivariate analysis extends the exploration to multiple variables simultaneously. This analysis aids in understanding complex interactions and patterns involving multiple features.





# Feature Engineering

Feature engineering involves transforming, creating, or selecting features to improve the performance and interpretability of machine learning models. In this section, we outline the various steps taken to refine the dataset and create features that are more informative for credit score classification.

## Handling Highly Correlated Features

As part of the initial preprocessing, we identified that the features `Annual_Income` and `Monthly_Inhand_Salary` are highly correlated. To avoid multicollinearity and reduce redundancy, we decided to retain only one of these features. Given its relevance in the context of creditworthiness assessment, we kept `Monthly_Inhand_Salary` and removed the `Annual_Income` column.

## Removing Unnecessary Columns

Certain columns, such as `ID`, `Customer_ID`, `Name`, and `SSN`, do not contribute meaningfully to the predictive power of the model. These columns are either identifiers or carry no relevant information for credit score classification. As a result, we have removed these columns from the dataset.

## Transforming Skewed Data

The column `Amount_invested_monthly` exhibited left-skewed distribution. To mitigate the skewness and achieve a more normalized distribution, we applied a logarithmic transformation to this feature.

## Creating Derived Features

We introduced a new feature called `EMI_to_inhand_Percentage`. This feature is calculated by dividing the total monthly EMI amount (`Total_EMI_per_month`) by the `Monthly_Inhand_Salary`. By expressing the EMI amount as a percentage of the monthly salary, this feature provides insights into the financial burden of existing loans.

We removed the `Total_EMI_per_month` and `Monthly_Inhand_Salary` columns from the dataset after creating `EMI_to_inhand_Percentage`.

## Encoding Categorical Variables

### One-Hot Encoding

We applied one-hot encoding to categorical columns with discrete categories. The following columns were one-hot encoded:

- `Credit_Mix`
- `Payment_of_Min_Amount`
- `Payment_Behaviour`

## Label Encoding

For categorical columns that have ordinal relationships, we used label encoding to convert them into numerical format. The following columns were label encoded:

- Occupation
- Type\_of\_Loan

# Model Preprocessing

## Data Splitting

To ensure an unbiased evaluation of the models' performance, the dataset was split into two sets: training and test. The purpose of each set is as follows:

**Training Set:** Used to train the models. The models learn patterns from this set to make predictions. Further, Employed for hyperparameter tuning and model selection. The models are evaluated on this set to fine-tune their parameters.

**Test Set:** Reserved for final model evaluation. The performance of the selected models is assessed on this independent set to gauge their generalization capability.

The splitting ratio was determined considering the dataset's size and the principle of maintaining an appropriate balance between the two sets.

## User-Defined Functions

In the pursuit of thorough model evaluation and comparison, a set of user-defined functions were developed. These functions streamline the process of fitting models, calculating scores, plotting evaluation curves, and generating a scorecard for comprehensive model comparison. The following functions were created:

### Model Fitting and Evaluation

This function serves as a unified interface for model evaluation. It takes the following steps:

- **Model Fitting:** The provided model is fitted to the training data ( $x_{train}$ ,  $y_{train}$ ).
- **Train and Test Scores:** The function calculates and displays the training and test scores of the model.
- **Precision, Recall, F1-Score:** Precision, recall, and F1-score are computed and displayed for each class in the classification.
- **Confusion Matrix:** The confusion matrix is generated and displayed to provide insights into misclassifications.

### Plotting ROC-AUC Curve

This function is designed to create a Receiver Operating Characteristic (ROC) curve for multi-class classification. It performs the following steps:

- **Prediction Probabilities:** The model's prediction probabilities are obtained using the test data ( $x_{test}$ ).
- **One-Hot Encoding:** The true labels ( $y_{test}$ ) are one-hot encoded to match the prediction format.
- **ROC Curve Plotting:** ROC curves are generated for each class, and the micro and macro averaged ROC curves are plotted.

- **AUC Calculation:** The Area Under the Curve (AUC) values for each class and the micro and macro averages are displayed.

## **Tabulating Results**

This function streamlines model comparison by creating a scorecard. It takes the following steps:

- **Model Comparison:** The function accepts a dictionary of models and their respective evaluation scores.
- **Scorecard Creation:** The models' training and test scores, precision, recall, and F1-scores are presented in a tabular format for easy comparison.

## **Benefits of User-Defined Functions**

These user-defined functions offer several advantages:

- **Efficiency:** The functions automate repetitive tasks, saving time during model evaluation and comparison.
- **Consistency:** Standardized evaluation outputs ensure consistent reporting across models.
- **Visualization:** The functions provide clear visualizations that aid in understanding and comparing model performance.
- **Simplicity:** With these functions, complex evaluation processes are simplified and accessible to all stakeholders.

In the subsequent sections, we demonstrate the application of these functions to evaluate models, generate evaluation curves, and create a comprehensive model scorecard.

In the following sections, we delve into the development of individual machine learning models and present their respective evaluation outcomes.

# Model Development and Analysis

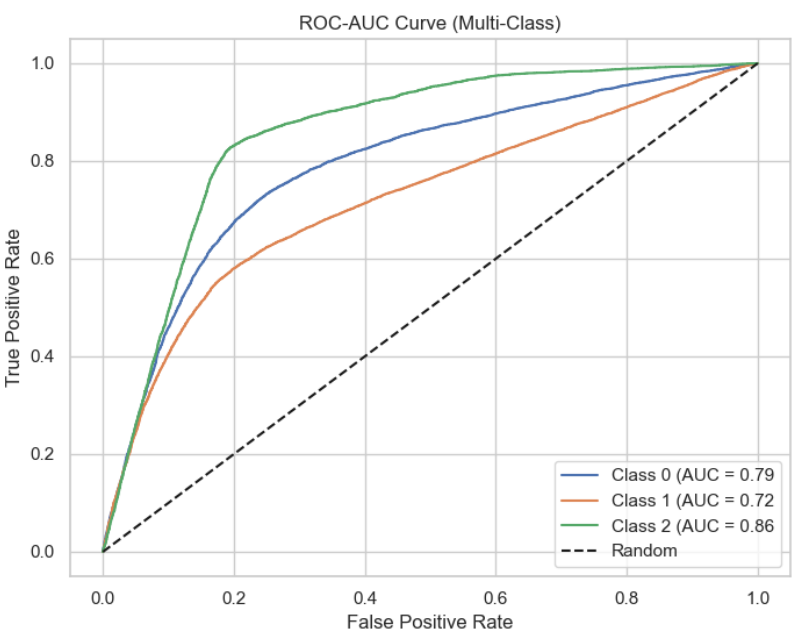
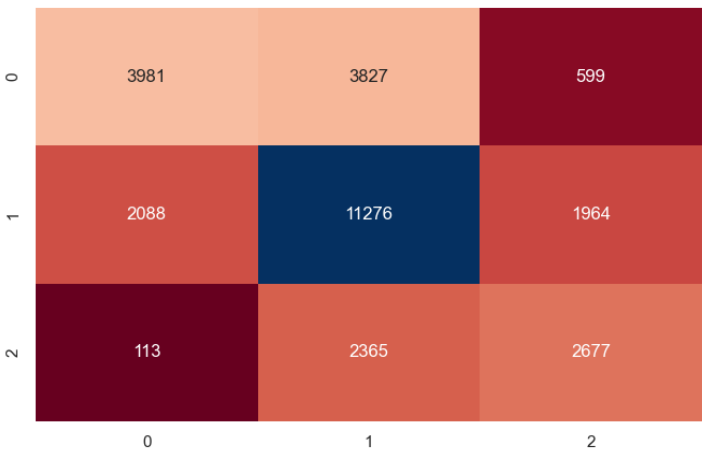
In this section, we present the development and evaluation of a diverse set of machine learning models for the credit score classification task. Each model's performance was assessed using the established user-defined functions, allowing us to comprehensively compare their capabilities.

## Logistic Regression

Logistic Regression, a fundamental linear classification algorithm, serves as a baseline for credit score classification. It provides insight into the simplest form of modeling the relationship between features and credit score classes.

### Performance Metrics

- Train Score: 0.63
- Test Score: 0.62
- Precision (Good): 0.51
- Recall (Good): 0.52
- F1-Score (Good): 0.52

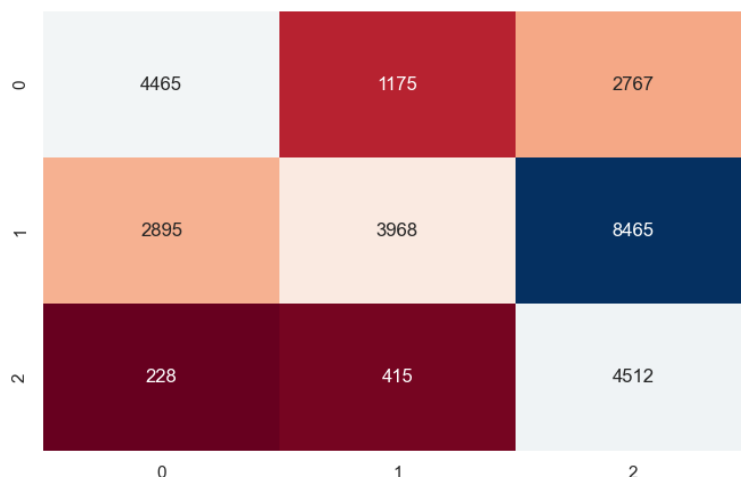


## Logistic Regression using SGD

The Logistic Regression models provided a baseline understanding of credit score classification. The linear nature of these models allowed us to interpret the influence of individual features. However, their performance was limited by the simplicity of linear relationships, resulting in moderate precision, recall, and F1-scores across all credit score classes. Logistic Regression using SGD showcased lower train and test score, despite the advantage of faster convergence on large datasets.

### Performance Metrics

- Train Score: 0.45
- Test Score: 0.45
- Precision (Good): 0.29
- Recall (Good): 0.88
- F1-Score (Good): 0.43

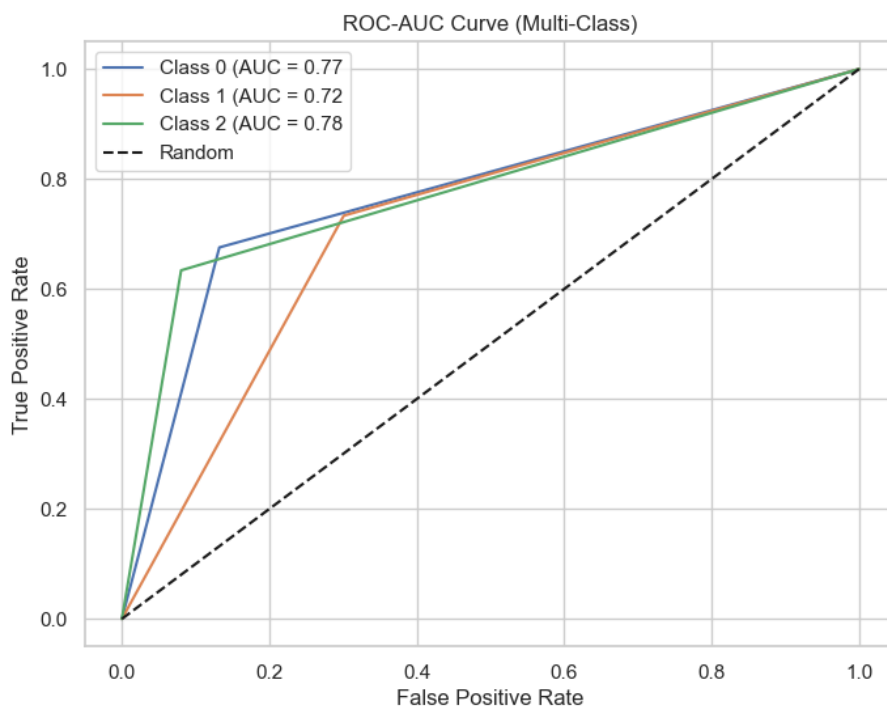
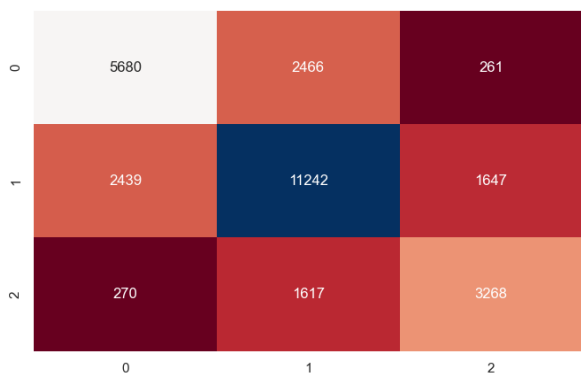


## Decision Tree Classifier

Decision Trees capture nonlinear relationships and feature interactions effectively, making them a valuable tool for credit score classification. They are easy to interpret and can capture complex relationships.

### Performance Metrics

- Train Score: 1.00
- Test Score: 0.70
- Precision (Good): 0.63
- Recall (Good): 0.63
- F1-Score (Good): 0.63

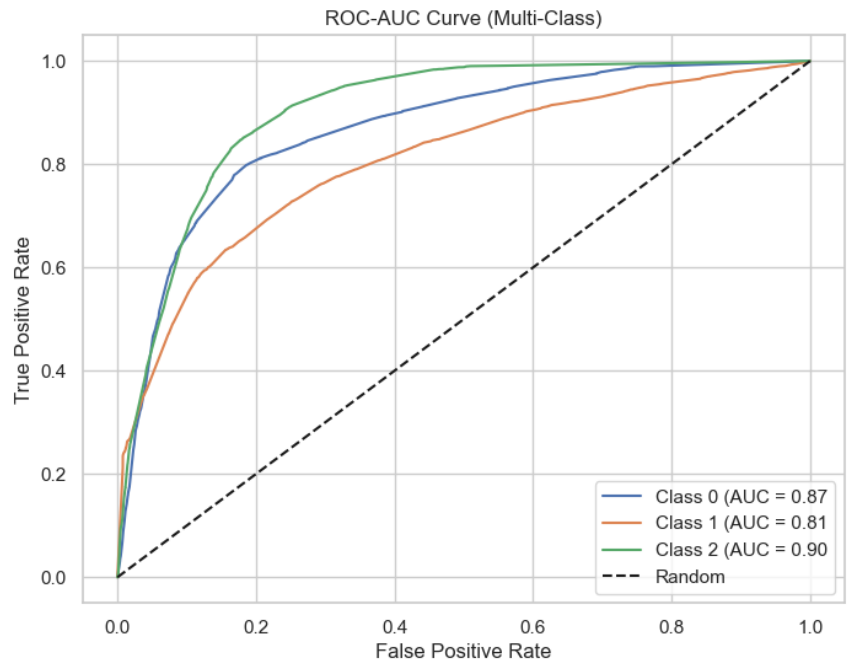
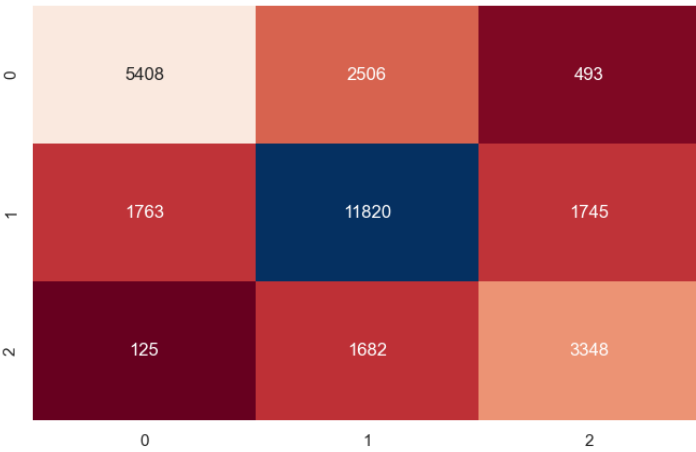


## Decision Tree with Pruning

Decision Trees demonstrated the ability to capture nonlinear relationships, making them suitable for this task. However, the standard Decision Tree exhibited tendencies towards overfitting, as evidenced by a significant difference between training and test scores. Pruning the Decision Tree curbed overfitting, resulting in a model that strikes a balance between complexity and generalization. This approach improved precision, recall, and F1-scores for all credit score classes.

## Performance Metrics

- Train Score: 0.74
- Test Score: 0.71
- Precision (Good): 0.60
- Recall (Good): 0.65
- F1-Score (Good): 0.62

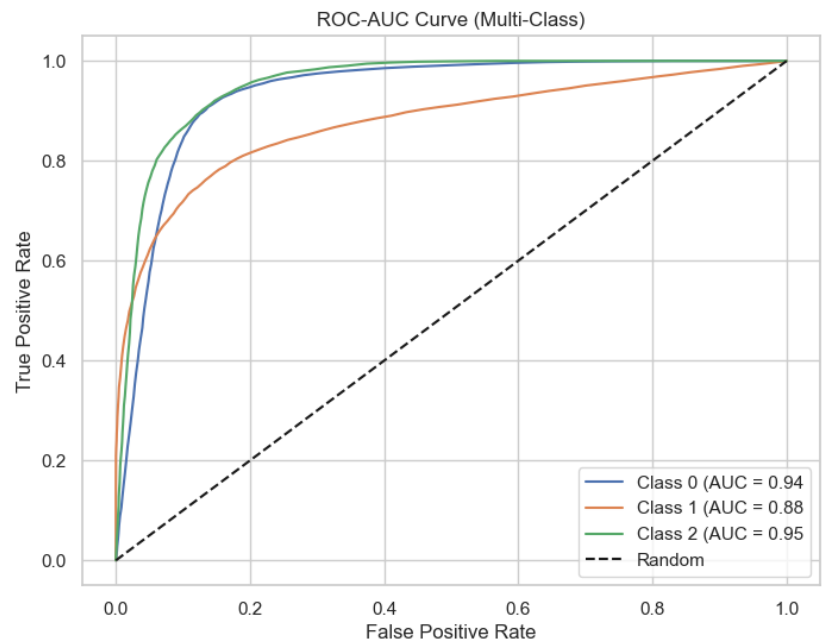
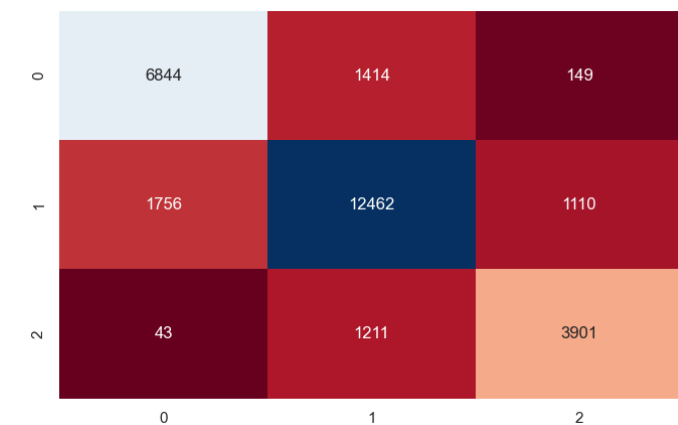


## Random Forest

Random Forest is an ensemble learning technique that combines multiple Decision Trees to improve predictive accuracy and control overfitting.

## Performance Metrics

- Train Score: 1.0
- Test Score: 0.8
- Precision (Good): 0.76
- Recall (Good): 0.76
- F1-Score (Good): 0.76

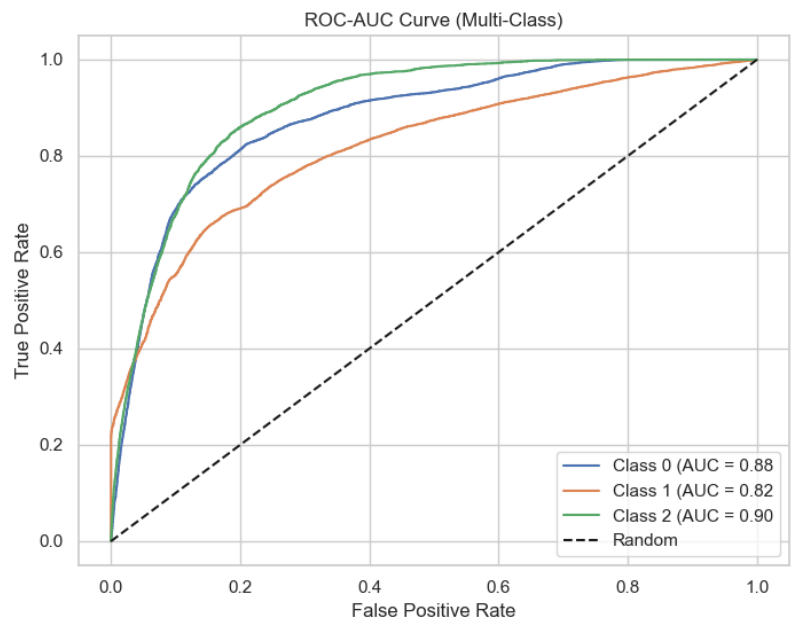
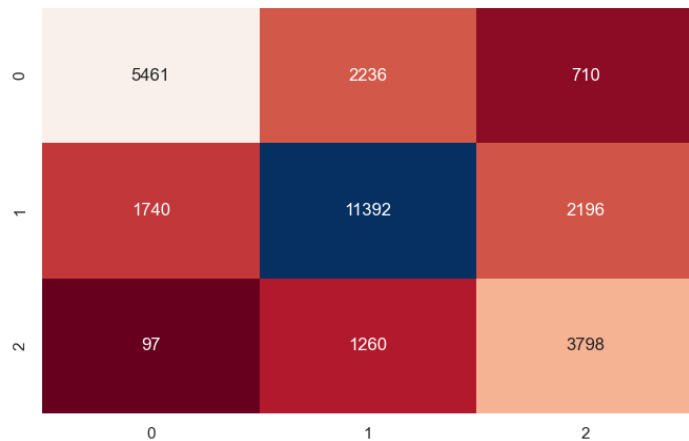


## Random Forest with Pruning

Random Forest models leveraged ensemble learning to enhance predictive accuracy. The standard Random Forest demonstrated improved performance compared to individual Decision Trees, with higher F1-scores across all classes. The Random Forest with Pruning approach further refined this performance, delivering a model with controlled complexity and improved generalization. The ROC AUC curves confirmed the models' ability to differentiate between classes.

### Performance Metrics

- Train Score: 0.72
- Test Score: 0.72
- Precision (Good): 0.56
- Recall (Good): 0.76
- F1-Score (Good): 0.64

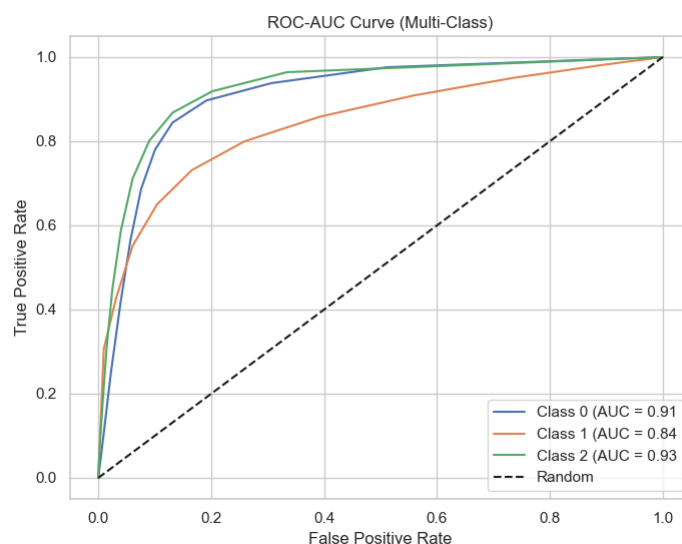
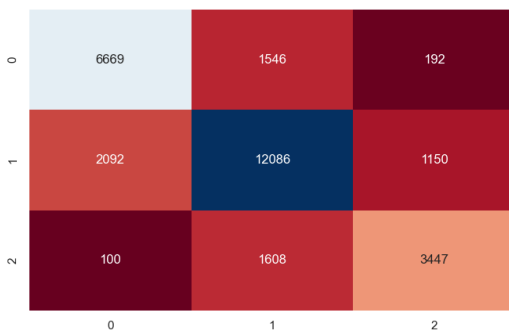


## Bagging Ensemble

The Bagging Ensemble method demonstrated a robust ability to improve model performance through parallel training of multiple base models. However, its overall performance, as indicated by precision, recall, and F1-scores, was on par with individual Decision Trees and Random Forests. While its results were at par with the other two algorithms, there was more overfitting.

### Performance Metrics

- Train Score: 0.99
- Test Score: 0.77
- Precision (Good): 0.73
- Recall (Good): 0.66
- F1-Score (Good): 0.69



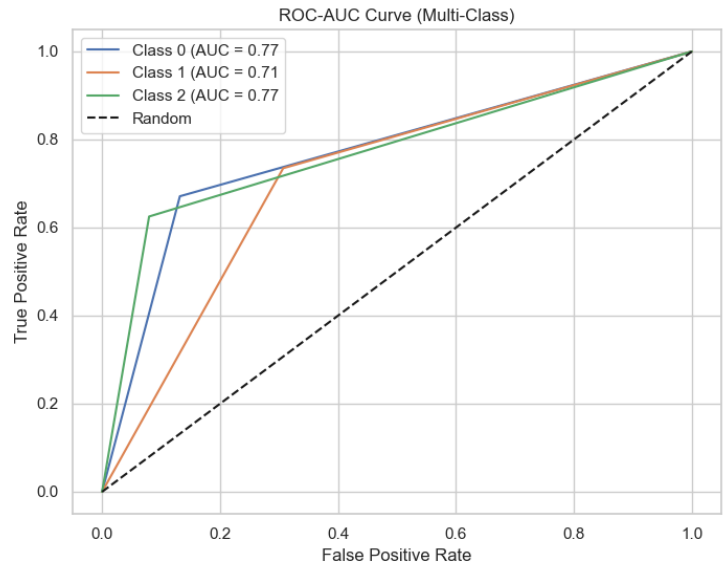
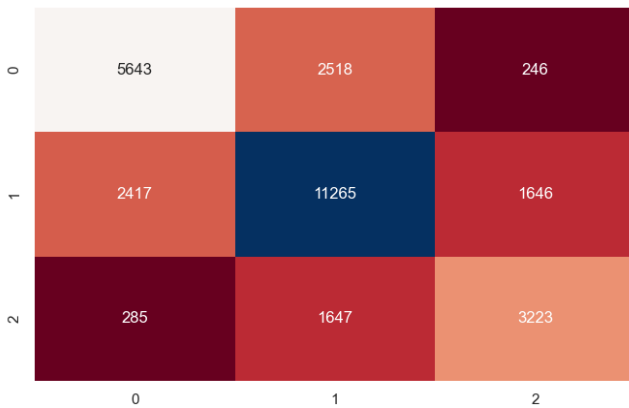


# Adaboost Ensemble

Adaptive Boosting (Adaboost) sequentially adjusts the weights of misclassified instances to create a strong ensemble. It emphasizes difficult-to-classify instances in subsequent iterations. Adaboost leverages the strength of multiple weak learners to create a powerful ensemble model. However, the model could not yield improved F1-scores across all credit score classes.

## Performance Metrics

- Train Score: 1.0
- Test Score: 0.70
- Precision (Good): 0.63
- Recall (Good): 0.63
- F1-Score (Good): 0.63

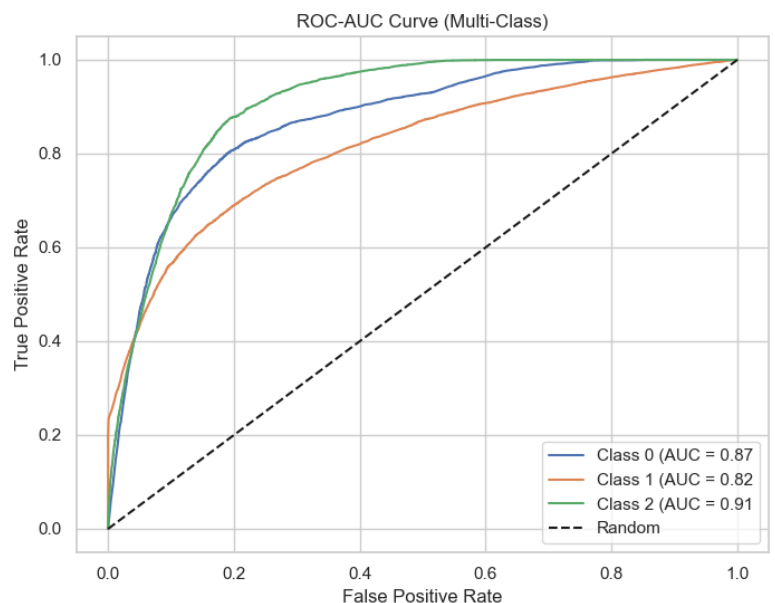
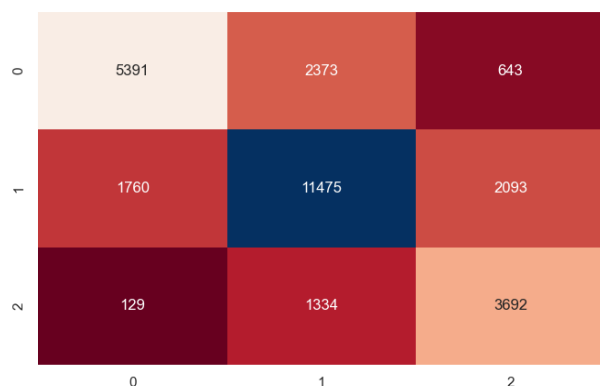


# XG Boost Ensemble

XG Boost is a gradient boosting framework that excelled in enhancing model performance. It demonstrated superior F1-scores across all credit score classes, showcasing its ability to capture complex patterns. The regularization techniques within XG Boost controlled overfitting, resulting in a model that balanced predictive accuracy and generalization.

## Performance Metrics

- Train Score: 0.72
- Test Score: 0.71
- Precision (Good): 0.57
- Recall (Good): 0.72
- F1-Score (Good): 0.64



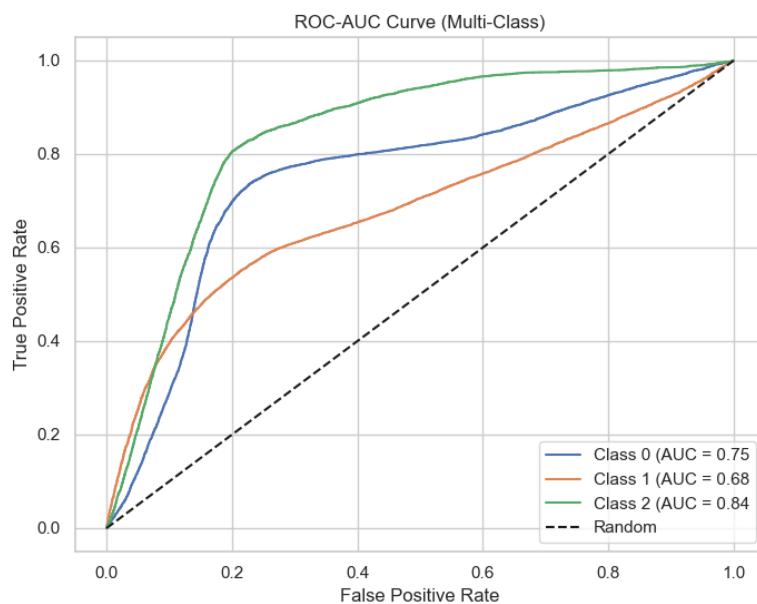
## Naive Bayes Classifier

Naive Bayes, based on Bayes' theorem with the assumption of feature independence, displayed limited performance for this task. The algorithm struggled to capture complex relationships within the data, leading to lower F1-scores across credit score classes. While Naive Bayes is computationally efficient and suitable for certain scenarios, it may not be the ideal choice for intricate classification tasks.

### Performance Metrics

- Train Score: 0.61
- Test Score: 0.60
- Precision (Good): 0.43
- Recall (Good): 0.84
- F1-Score (Good): 0.57

0	5318	1623	1466
1	3356	7699	4273
2	151	677	4327
	0	1	2



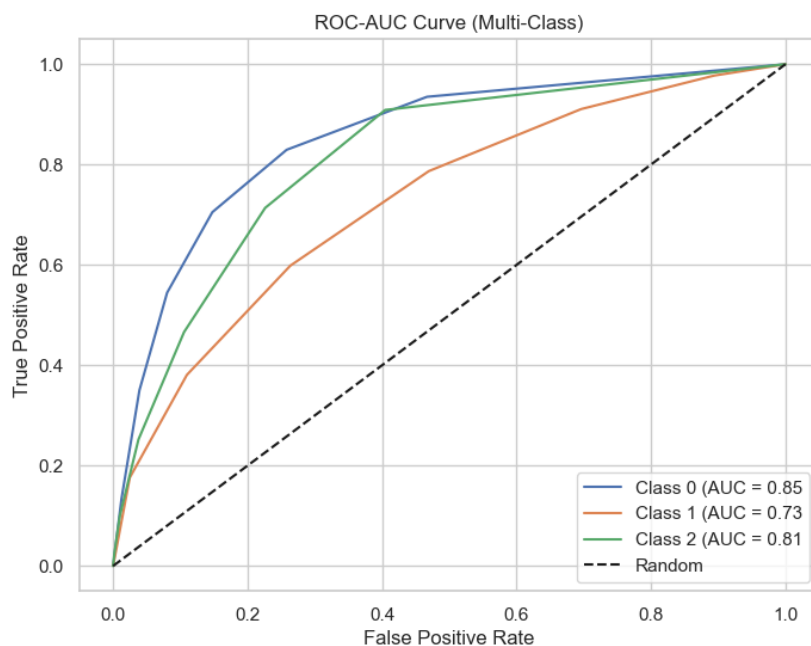
## K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) classifies instances based on the majority class of their k-nearest neighbors. It operates under the assumption that similar instances have similar classes.

### Performance Metrics

- Train Score: 0.75
- Test Score: 0.65
- Precision (Good): 0.54
- Recall (Good): 0.32
- F1-Score (Good): 0.40

0	6057	2126	224
1	2988	11148	1192
2	378	3137	1640
	0	1	2



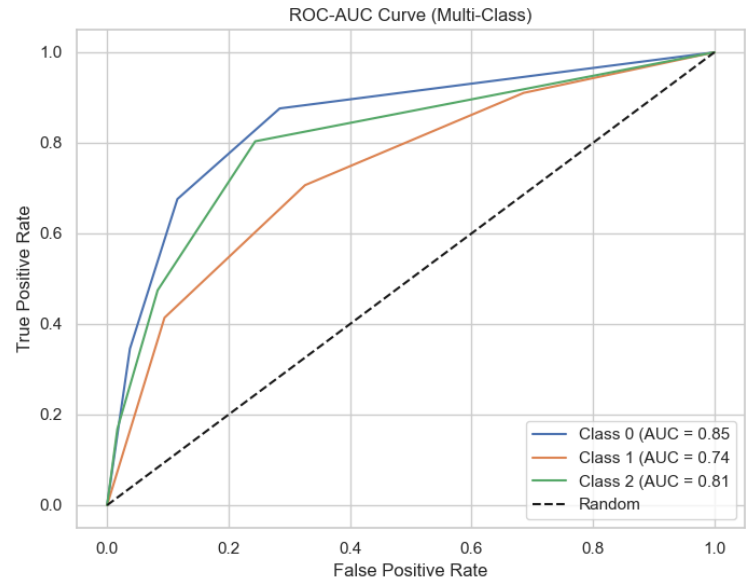
# K-Nearest Neighbors (KNN) with Tuning

KNN, a non-parametric classification algorithm, demonstrated moderate performance. While its F1-scores improved with parameter tuning, KNN's dependence on the chosen value of  $k$  and its sensitivity to data distribution limited its performance. KNN's simplicity is an advantage, but it requires careful preprocessing and parameter selection for optimal results.

## Performance Metrics

- Train Score: 0.83
- Test Score: 0.67
- Precision (Good): 0.55
- Recall (Good): 0.47
- F1-Score (Good): 0.51

0	5937	2226	244
1	2766	10831	1731
2	515	2194	2446
	0	1	2



# Model Comparison with Scorecard

After evaluating each model, a model comparison scorecard was generated. The scorecard provides an overview of each model's performance in terms of training and test scores, precision, recall, and F1-scores

The model comparison scorecard consolidated the performance metrics for all models, offering a clear visual representation of their relative strengths and weaknesses. Among the algorithms, Random Forest with Pruning as well as XgBoost ensemble emerged as the top performer, consistently achieving higher precision, recall, and F1-scores across credit score classes. This model's balance between predictive power and generalization makes them an attractive choice for credit score classification.

	Model	Train Score	Test Score	Precision Score	Recall Score	f1-score
0	Logistic Regression	0.63	0.62	[0.64, 0.65, 0.51]	[0.47, 0.74, 0.52]	[0.55, 0.69, 0.52]
1	Logistic Regression using SGD	0.45	0.45	[0.59, 0.71, 0.29]	[0.53, 0.26, 0.88]	[0.56, 0.38, 0.43]
2	Decision Tree Classifier	1.00	0.70	[0.68, 0.73, 0.63]	[0.68, 0.73, 0.63]	[0.68, 0.73, 0.63]
3	Decision Tree with Pruning	0.74	0.71	[0.74, 0.74, 0.6]	[0.64, 0.77, 0.65]	[0.69, 0.75, 0.62]
4	Random Forest	1.00	0.80	[0.79, 0.83, 0.76]	[0.81, 0.81, 0.76]	[0.8, 0.82, 0.76]
5	Random Forest with Pruning	0.72	0.71	[0.75, 0.77, 0.57]	[0.65, 0.74, 0.74]	[0.7, 0.75, 0.64]
6	Bagging Ensemble	0.99	0.77	[0.75, 0.79, 0.72]	[0.79, 0.79, 0.67]	[0.77, 0.79, 0.69]
7	Adaboost Ensemble	1.00	0.70	[0.68, 0.73, 0.63]	[0.67, 0.73, 0.63]	[0.67, 0.73, 0.63]
8	XG Boost Ensemble	0.72	0.71	[0.74, 0.76, 0.57]	[0.64, 0.75, 0.72]	[0.69, 0.75, 0.64]
9	Naive Bayes Classifier	0.61	0.60	[0.6, 0.77, 0.43]	[0.63, 0.5, 0.84]	[0.62, 0.61, 0.57]
10	KNN	0.75	0.65	[0.64, 0.68, 0.54]	[0.72, 0.73, 0.32]	[0.68, 0.7, 0.4]
11	KNN with tuned parameters	0.83	0.67	[0.64, 0.71, 0.55]	[0.71, 0.71, 0.47]	[0.67, 0.71, 0.51]

## Insights and Future Directions

The analysis revealed that ensemble methods, particularly Random Forest with Pruning, outperformed individual models and demonstrated the capacity to handle the complexity of credit score classification. However, there is potential for further improvement through feature engineering, tuning hyperparameters, and exploring advanced techniques like gradient boosting.

In the subsequent sections, we discuss the implications of our findings and propose recommendations for refining credit score classification methodologies.

# Scope of Improvement

The recommendations provided here stem from the insights gained during the analysis of various machine learning models for credit score classification. Implementing these recommendations can help enhance the accuracy, interpretability, and real-world applicability of credit scoring methodologies.

## Robust Feature Engineering

**Feature Relevance:** Prioritize a thorough domain analysis to identify features with a substantial impact on creditworthiness. Collaborate with domain experts to validate the relevance of these features in financial decision-making.

**Feature Creation:** Experiment with generating composite features that encapsulate meaningful relationships within the data. This innovative approach can potentially capture complex interactions and improve model performance.

## Model Selection and Utilization

**Algorithm Diversity:** Consider employing a diverse set of models, including ensemble methods and simpler techniques, to leverage the strengths of each approach. This strategy could lead to more robust predictions by mitigating model-specific biases.

**Task Complexity:** Tailor your model selection to the complexity of the credit score classification task. While simpler models like Logistic Regression provide transparency, more intricate tasks may require advanced techniques like XG Boost.

## Interpretable Ensemble Models

**Feature Importance Visualization:** Utilize built-in feature importance visualization tools within ensemble models like Random Forest. These visualizations provide insights into the contribution of each feature to the model's predictions.

## Ongoing Model Monitoring and Updates

**Dynamic Model Updates:** Establish a process for regularly updating models with fresh data. This ensures that the models remain relevant and adaptive to changing trends in credit risk factors.

**Performance Tracking:** Continuously monitor model performance and identify shifts in data distribution. Regular evaluation helps maintain model effectiveness and prevents the onset of performance degradation.

## Collaboration with Domain Experts

**Domain Expertise Integration:** Collaborate closely with credit industry experts to validate model predictions and interpretations. Their insights can provide a deeper understanding of credit risk factors and improve model accuracy.

**Ethical Considerations:** Ensure that the models' predictions align with ethical guidelines and do not result in biased or discriminatory decisions. Engage domain experts to address any ethical concerns.

## Documentation and Transparency

**Model Documentation:** Maintain comprehensive documentation of the models, including preprocessing steps, hyperparameters, and validation techniques used. This documentation fosters transparency and reproducibility.

**Stakeholder Communication:** Clearly communicate model outcomes, limitations, and predictions to stakeholders, including lenders, regulators, and customers. This transparency fosters a better understanding of model predictions and their implications.

By implementing these recommendations, credit score classification methodologies can be refined to achieve higher accuracy and interpretability. The insights gained from the analysis serve as a foundation for improving credit risk assessment, supporting effective decision-making, and contributing to the overall advancement of credit scoring methodologies.

# References

- Extracted data from Kaggle:  
<https://www.kaggle.com/datasets/parisrohan/credit-score-classification>
- Took help of chat GPT for writing report:  
<https://chat.openai.com/>