Introduction

The objective of this project is to develop a machine learning model that predicts the credit score of customers based on various attributes. This model aims to classify customers into appropriate credit score ranges, thereby assisting financial institutions in assessing creditworthiness and making informed lending decisions.

Exploring the data

1. Importing the Data and Viewing the First Few Rows

	ID	Customer_ID	Month	Name	Age	SSN	Occupation	Annual_Income	Monthly_Inhand_Salary	Num_Bank_Accounts	...	Credit_Mix	Outstanding_Debt	C
)	0x1602	CUS_0xd40	January	Aaron Maashoh	23	821-00-0265	Scientist	19114.12	1824.843333	3	...	_	809.98	
I	0x1603	CUS_0xd40	February	Aaron Maashoh	23	821-00-0265	Scientist	19114.12	NaN	3	...	Good	809.98	
!	0x1604	CUS_0xd40	March	Aaron Maashoh	-500	821-00-0265	Scientist	19114.12	NaN	3	...	Good	809.98	
)	0x1605	CUS_0xd40	April	Aaron Maashoh	23	821-00-0265	Scientist	19114.12	NaN	3	...	Good	809.98	
I	0x1606	CUS_0xd40	May	Aaron Maashoh	23	821-00-0265	Scientist	19114.12	1824.843333	3	...	Good	809.98	

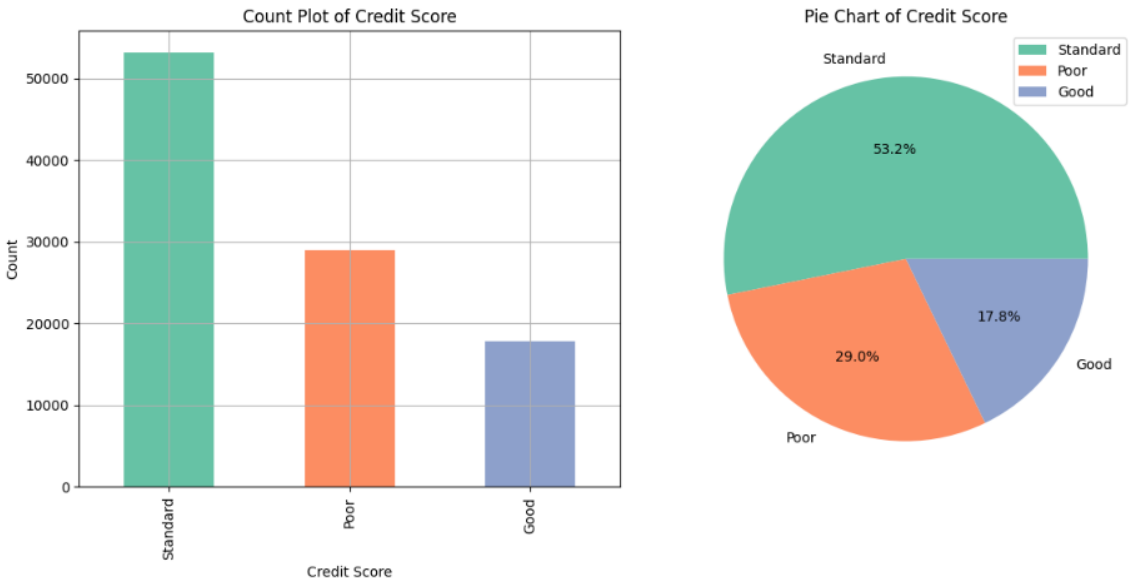
2. Checking for Missing Values

3. Checking for Duplicate Rows

Data Visualization

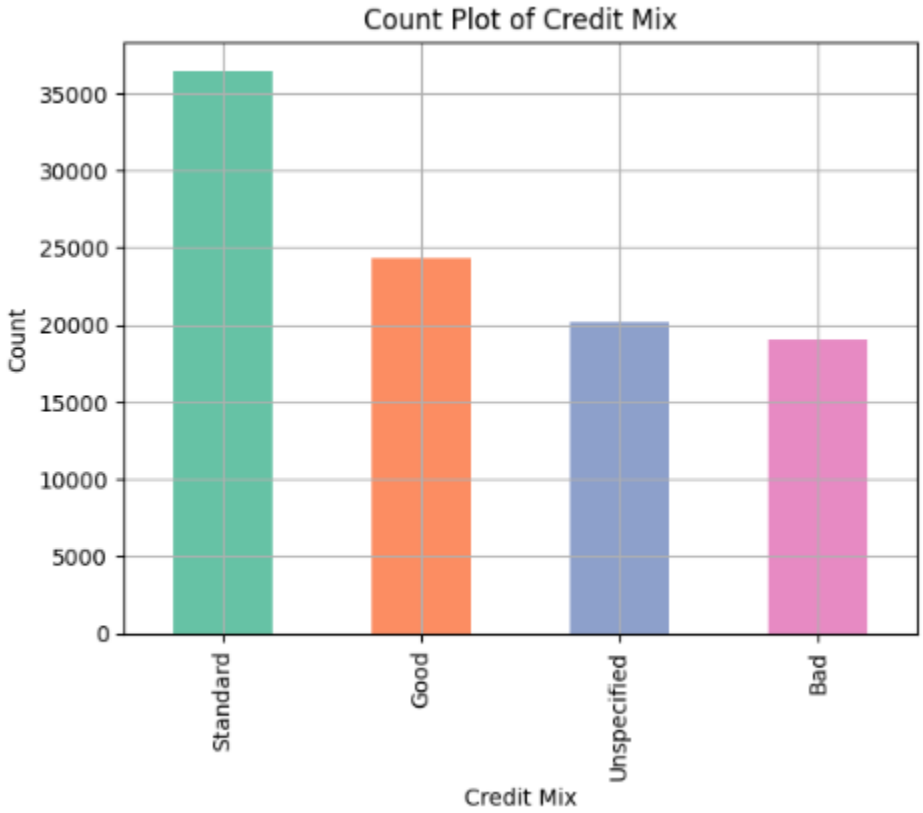
1) Count Plot & Pie Chart of Credit Scores

To understand the distribution of credit scores within the dataset, we use both a count plot and a pie chart. These visualizations help in identifying the frequency and proportion of different credit score categories. Here's a description of each plot:

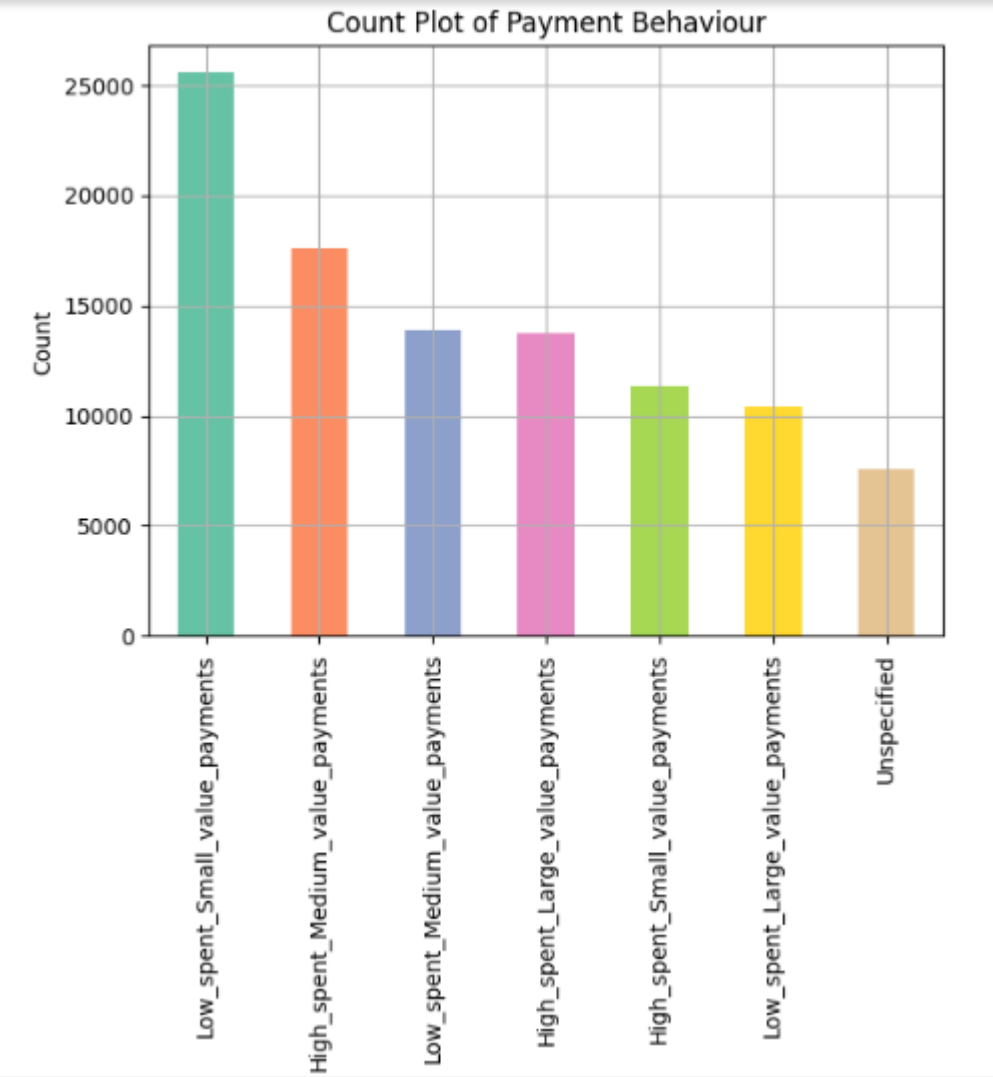


(Count Plot & Pie Chart of Credit Scores)

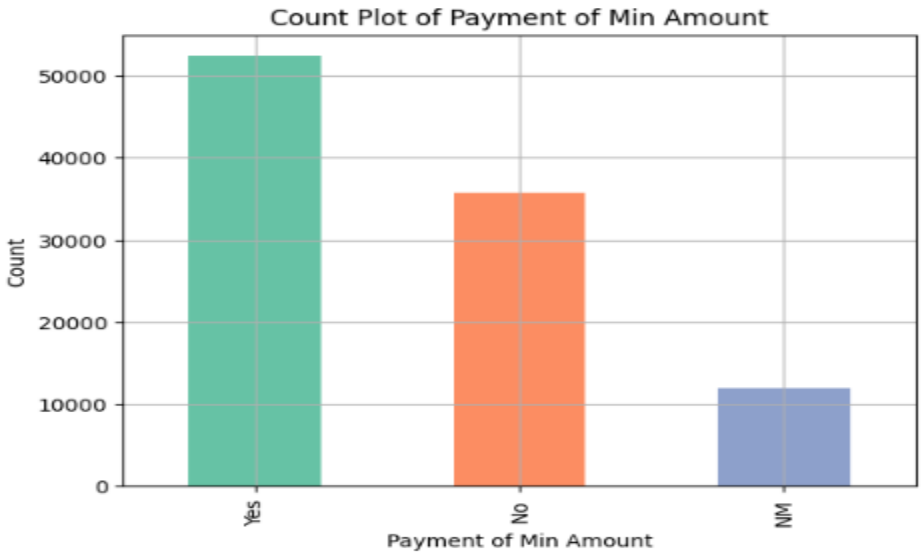
2) Count Plot of Credit Mix



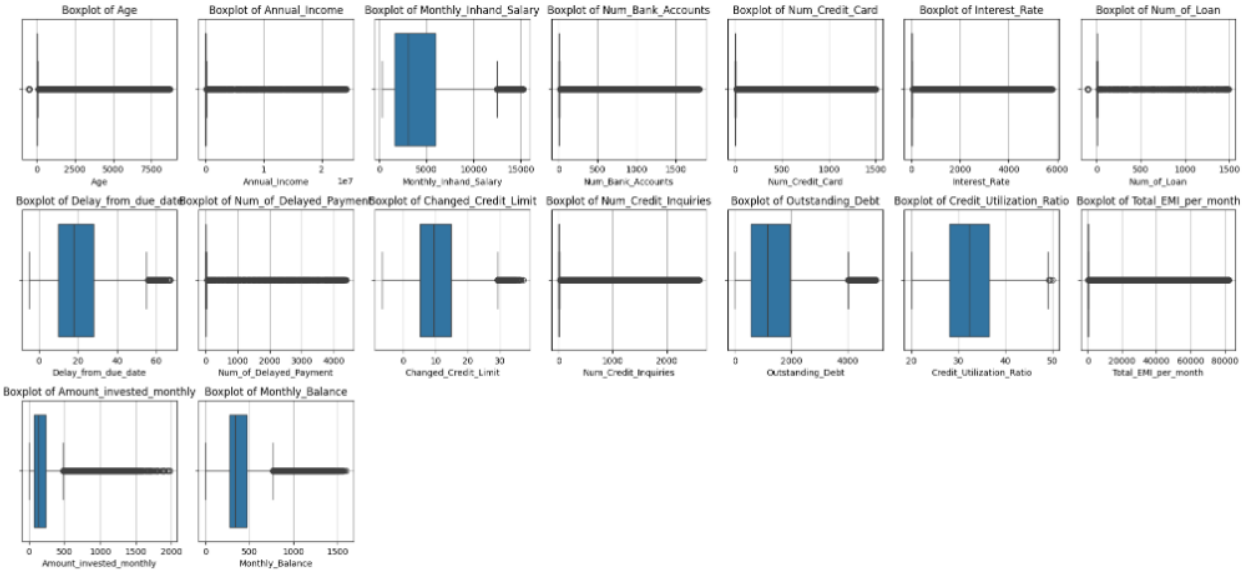
3) Count Plot of Payment Behavior


Count Plot of Payment Behaviour
Count
25000
20000
15000
10000
5000
0
Low_spent_Small_value_payments
High_spent_Medium_value_payments
Low_spent_Medium_value_payments
High_spent_Large_value_payments
High_spent_Small_value_payments
Low_spent_Large_value_payments
Unspecified

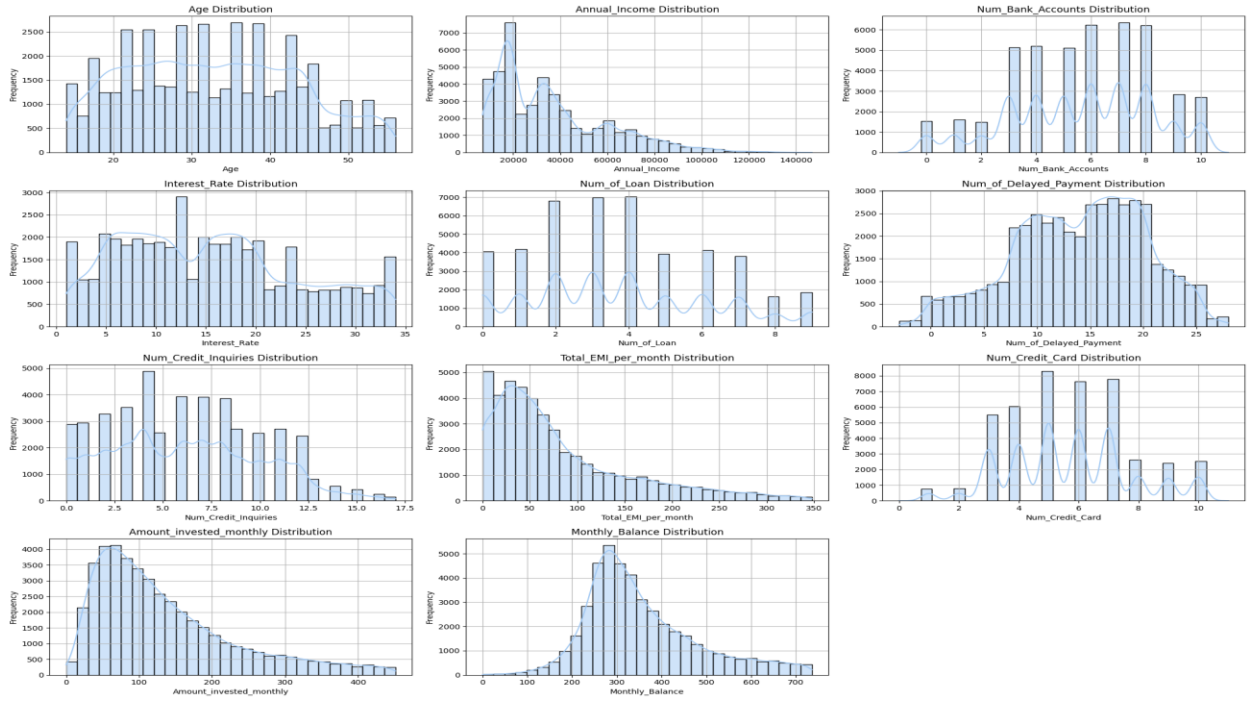
4) Count Plot of Payment of Min Amount



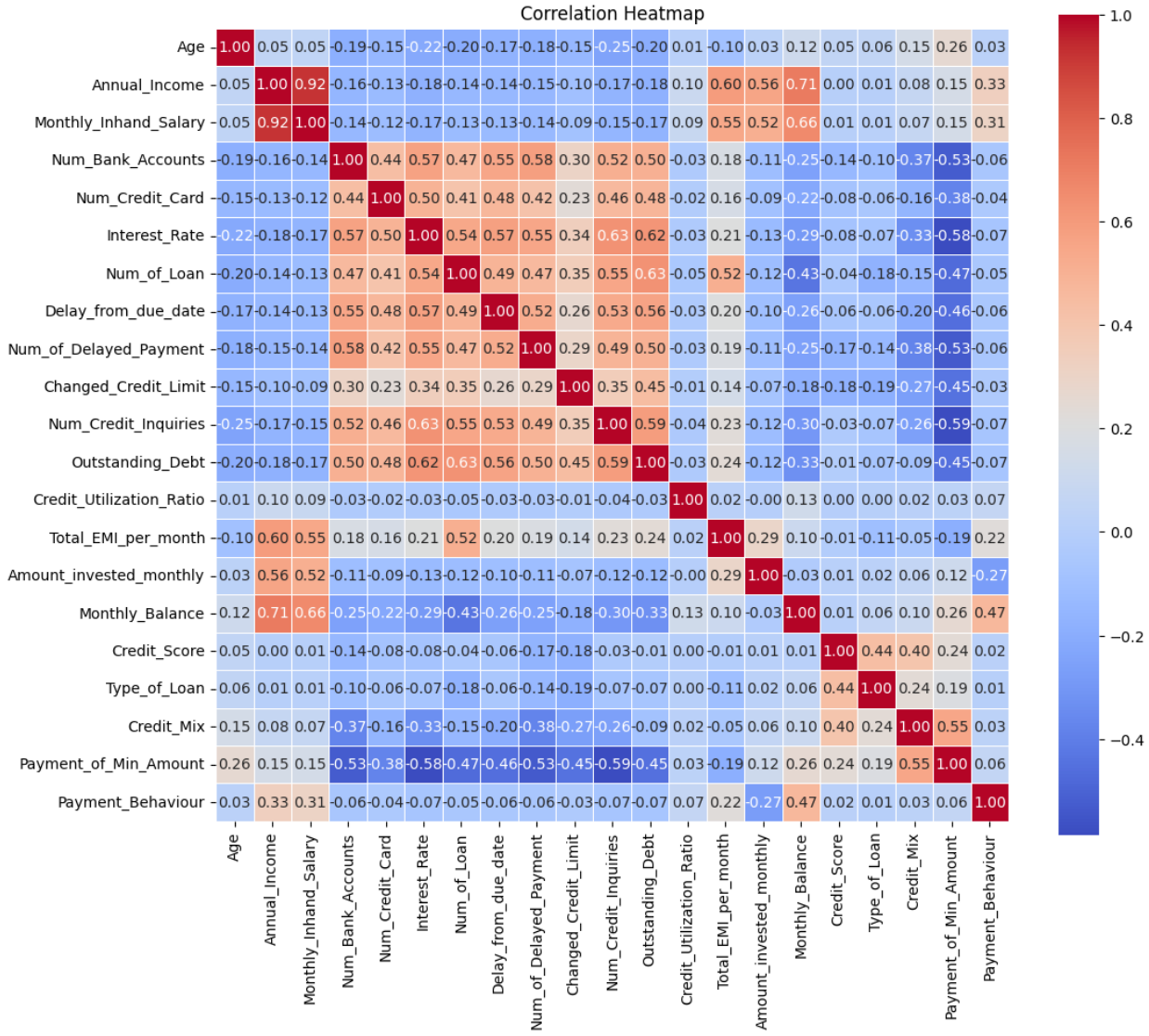
5) Boxplots for the Numerical Columns



6) Visualizing the Distribution of Numerical Columns



7) Correlation Heatmap



Data Preprocessing

1) **Converting Columns to Numeric**
2) **Outlier Detection and Removal**
   Identify and remove outliers from numerical columns using the Interquartile Range (IQR) method.

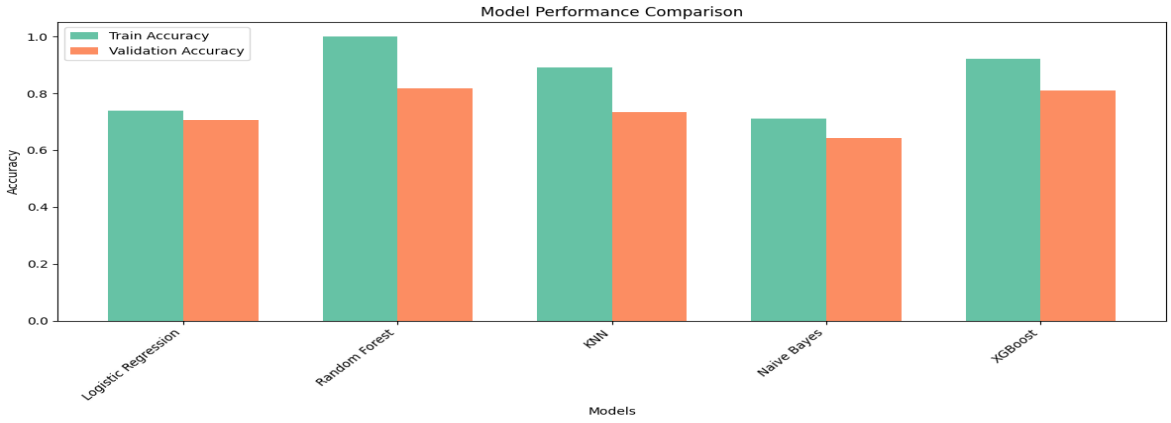
3) **Handling Missing Values in Numerical Columns**
   Fill null values in numerical columns with the mean value.
4) **Handling Missing Values in Categorical Columns**

Fill null values in categorical columns with the most frequent value (mode).

5) **Dropping Irrelevant Columns**
   Drop columns that are not needed for model training.
6) **Encoding Categorical Variables**
   Use Target Encoding for categorical variables to convert them into numeric values.
7) **Feature Scaling**
   Scale the features using StandardScaler to standardize the dataset.
8) **Handling Class Imbalance**
   Apply SMOTE (Synthetic Minority Over-sampling Technique) to balance the training data.

Modeling

We applied a variety of machine learning models to predict customer credit scores. The models included Logistic Regression, Random Forest, K-Nearest Neighbors (KNN), Naive Bayes, and XGBoost. Each model underwent training on resampled training data and subsequent evaluation on both the training and validation sets to assess their performance. Among the models evaluated, Random Forest and XGBoost emerged as the top performers, exhibiting the highest accuracy scores on both the training and validation set

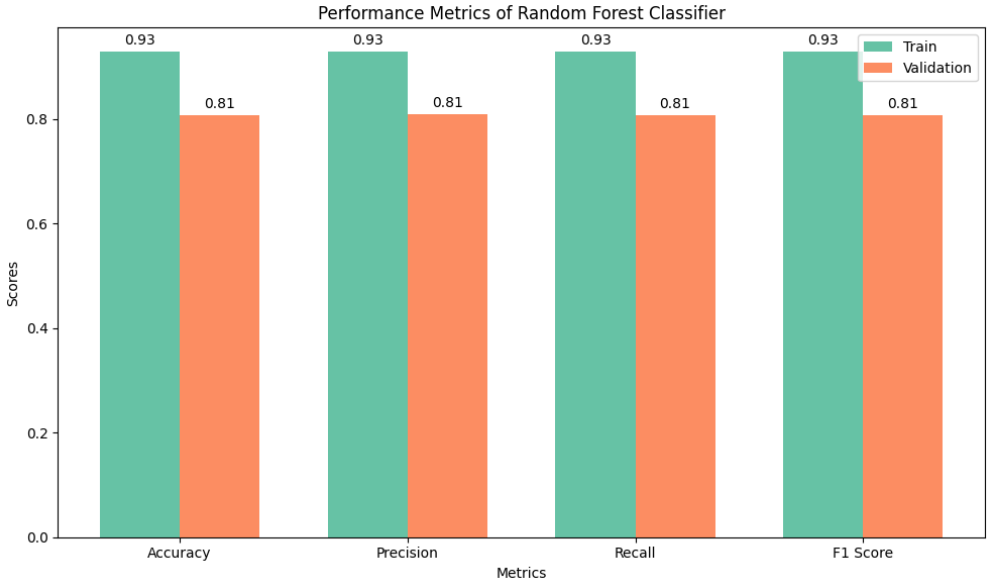


Hyperparameter Tuning for Best Models

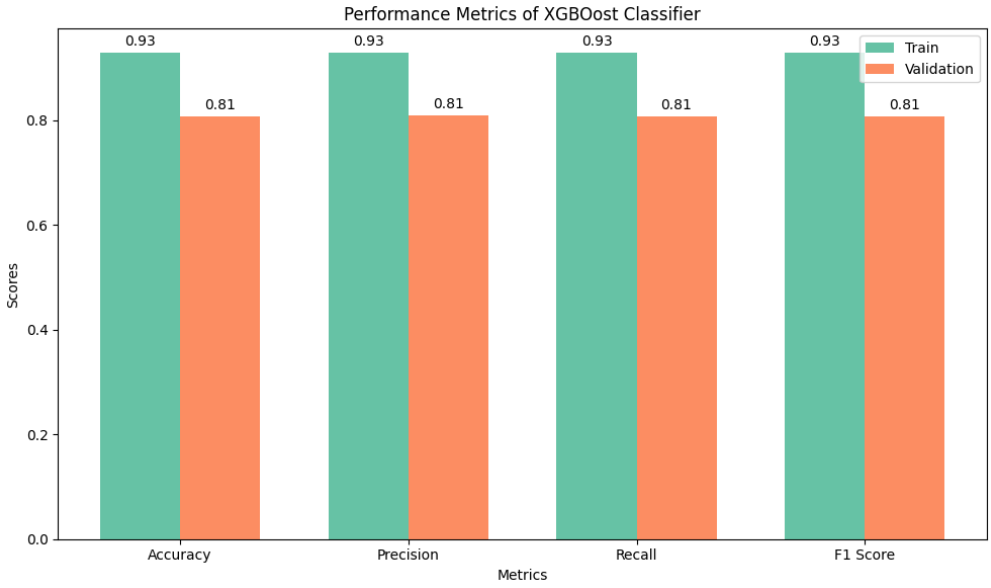
After initializing and fitting the Random Forest and XGBoost classifiers, we performed hyperparameter tuning to optimize their performance.

These evaluation metrics [Accuracy, Precision, Recall, F1-Score"] were calculated to assess the models' effectiveness:

1) Random Forest



2) XGBoost



API

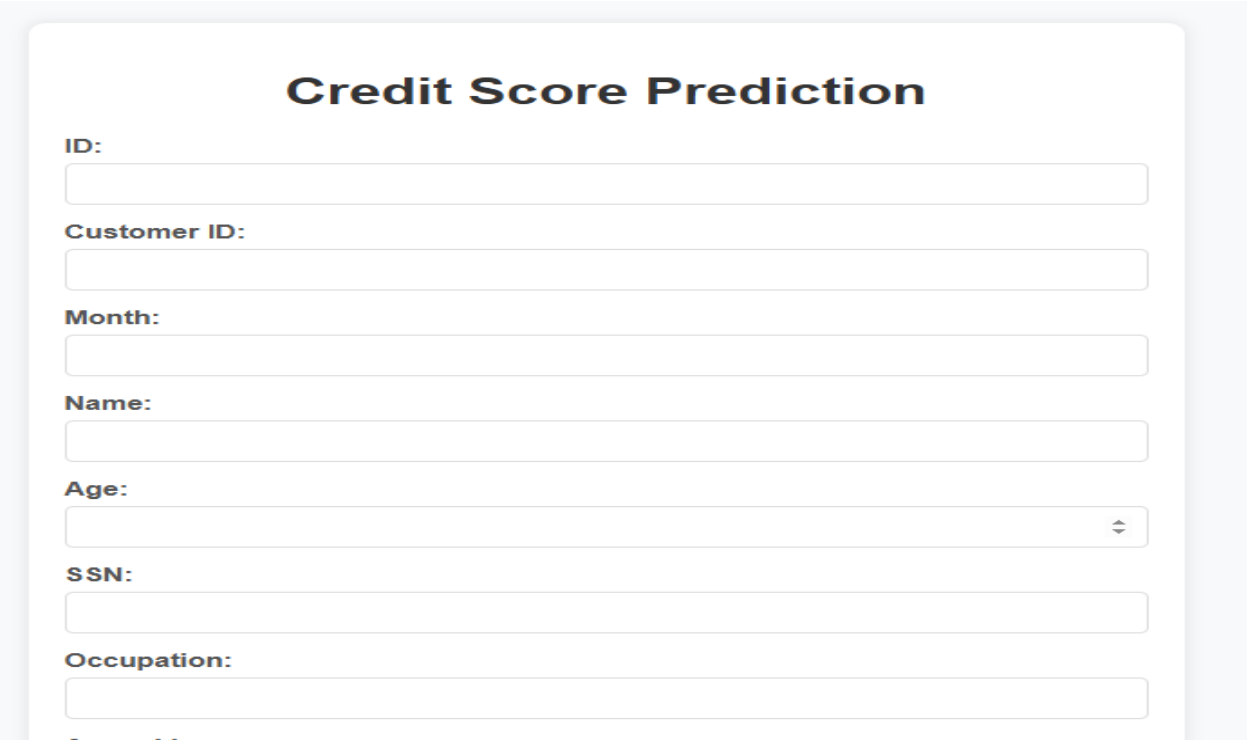
1. User Input: Users provide their information through the HTML form, including personal details, financial data, and payment behavior.

2. Data Submission: Upon clicking the "Submit" button, the form data is sent to the Flask application for processing.

3. Data Preprocessing: The Flask application preprocesses the incoming data using the same preprocessing steps applied to the training data before fitting it to the model. This preprocessing may involve tasks such as data cleaning, feature scaling, or encoding categorical variables.

4. Credit Score Prediction: The preprocessed data is passed to the trained XGBoost model, which predicts the customer's credit score (classifies the customer into one of three categories: standard, poor, or good credit) based on their attributes.

6. Result Display: The classification result is returned to the user interface, where it is displayed to the user.



Annual Income:

Monthly In-hand Salary:

Number of Bank Accounts:

Number of Credit Cards:

Interest Rate:

Number of Loans:

Type of Loan:

Delay from Due Date:

Number of Delayed Payments:

Outstanding Debt:

Credit Utilization Ratio:

Credit History Age:

Payment of Minimum Amount:

Yes

Total EMI per Month:

Amount Invested Monthly:

Payment Behaviour:

Monthly Balance:

Submit