Credit Score Classification To Issue Loan Using Random Forest Algorithm

A project report submitted in partial fulfillment
of the requirements for the award of degree in

M.Sc. Data Science

Submitted by

Pandi Sudeer

(VP21CSCIO200077)

Under the esteemed guidance of

Mr. Satyanarayana Botsa

Assistant Professor

Department of Computer Science

GITAM Institute of Science

GITAM (Deemed to be University)

Visakhapatnam - 530045, A.P

(2022-23)

CERTIFICATE

This is to certify that the project entitled **“Credit Score Classification To Issue Loan Using Random Forest Algorithm”** is a bonafide work done by **Pandi Sudeer**, (**VP21CSCI0200077)** during **December 2022** to **April 2023** in partial fulfillment of the requirement for the award of degree of **Master of Science, Data Science** in the Department of Computer Science, GITAM School of Science, GITAM (Deemed to be University), Visakhapatnam.

Internal Guide
Mr. Satyanarayana Botsa
Assistant Professor
Department of Computer Science
Gitam School of Science

Head of the Department
Dr. T. Uma Devi
Associate Professor
Department of Computer Science
Gitam School of Science

DECLARATION

I, **Pandi Sudeer,** (**VP21CSCI0200077)** hereby declare that the project entitled **"Credit Score Classification To Issue Loan Using Random Forest Algorithm"** is an original work done in the partial fulfilment of the requirements for the award of degree of **Master of Science, Data Science** in **GITAM School of Science, GITAM (Deemed to be University), Visakhapatnam.** I assure that this project work has not been submitted towards any other degree or diploma in any other colleges or universities.

Pandi Sudeer

(VP21CSCI0200077)

ACKNOWLEDGEMENT

I want to express my gratitude towards my project mentor, **Mr. Satyanarayana Botsa**, Assistant Professor, for his exceptional support and guidance throughout the completion of my project. His trust and encouragement gave me an incredible opportunity to excel in this endeavor.

I must also express my sincere appreciation to my project coordinator, **Smt. V. Satyasaivani**, Associate Professor, whose unwavering encouragement and assistance played a crucial role in the project's accomplishment.

Additionally, I am delighted to acknowledge **Dr. T. Uma Devi**, our Head of Department, for her constant inspiration and efforts in providing us with all the essential resources and facilities necessary for the project's success.

I express my gratitude to **Prof. K. Vedavathi**, our beloved principal, for the facilities provided by her throughout the course.

Furthermore, I would like to thank the entire **Faculty and Staff** of the Computer Science Department for their unyielding support and guidance throughout my academic journey. Their commitment to teaching and mentorship has been a continuous inspiration for me.

Pandi Sudeer
(VP21CSCI0200077)

INDEX

CONTENTS **PAGE**

ABSTRACT

The aim is to provide a brief overview of the use of credit scoring in assessing a person’s creditworthiness and likelihood of repaying a loan on time, based on their credit history (CH). The prime objective is to find out the credit score for all existing customers from existing data and help lenders to allocate funds using a good prediction. Banks and credit card companies calculate credit score (CS) to determine creditworthiness. It helps banks and credit card companies immediately to issue loans to customers with good creditworthiness. Credit scoring is a way to predict borrower’s behaviour or future possibility of delay using input and historical information. It affects the government's economy thus financial companies should give loans only to responsible and solvent part of the population. In credit score classification, there are three credit scores that banks and credit card companies use to label their customers:

- Good [2]
- Standard [1]
- Poor [0]

A person with a good credit score (CS) will get loans from any bank and financial institution. In this project I developed a Classification model with Machine Learning using Python. The goal of this project is to predict credit score by giving inputs through a web application to our model according to the features we used in training the model.

1. INTRODUCTION

1.1 Background

Credit scoring is a method that helps to decide whether to provide loans to consumers, it is a probability of person's debt repay in a timely manner, based on person's credit history. The process typically involves analysing a consumer's credit history, which includes information such as credit card balances, payment history, and outstanding loans. Advanced statistical and mathematical methods, including machine learning algorithms and artificial intelligence, are often used to make these evaluations.

Credit scoring model predictions have now become an important component of the commercial world. The important aspect of assets used in banking directly comes from the profit earned from the distribution of credit cards among the customers.

Fig 1.1: Credit Score Measure

The best step of any banking system is to identify the worthy stakeholders from which they can get maximum profit from the investment in the assets. The field of banking is affecting the lives of the credit card holder loan holder by its services. Financial companies issued credit cards following a thorough verification and validation process, but there is no guarantee that the credit cards were granted to deserving candidates. And is estimated based on applicants' historical data which helps credit lenders in the granting of credit products.

People are considered financially reliable when their score is higher. Credit scoring eliminates the human factor and uses only reliable data. Advanced statistical and mathematical methods provide fast and automatic tools that help to make effective decisions.

Currently, financial institutions are adopting various risk assessment tools and techniques for credit scoring systems to minimize the risk up to some extent. The scheme uses deep learning-based algorithm to map the input weight with hidden biases and uses an effective classifier for taking intelligent decisions and perform a rigorous training to train the predictive model.

1.2 Motivation (Problem Statement)

The problem statement of this project is to predict the creditworthiness of a customer based on their credit score. The aim is to help banks and financial institutions in decision-making regarding issuing loans to customers. The motivation behind this project is to provide a reliable and efficient way to evaluate the creditworthiness of customers, which can save time and reduce the risk of bad debt.

By using machine learning techniques like random forest, this project can help banks and financial institutions to make informed decisions based on historic data. This can lead to faster processing of loan applications, reduced risk of bad debt, and better customer satisfaction.

The motivation is to improve the efficiency and accuracy of the loan issuing process, while also providing a fair and transparent evaluation of customers' creditworthiness. Machine learning models finds patterns and relationships from the training data set and yields predictions about the future. The goal of credit scoring is to identify consumers who are less likely to default on their loans, and to help financial institutions make more informed decisions about providing credit.

1.3 Existing System

1.3.1 Overview of existing system & drawbacks

- Some previous studies proposed credit scoring is a conventional decision model and its focus is on risk approximation approach associated with credit products such as credit card, loans, etc. and is estimated based on applicants' historical data which helps credit lenders in granting credit products.

- Credit scoring model is a classification problem where the dependent variable is dichotomous and assigns “false” to failed loans and “true” to non-failed loans. figure below shows the various credit risk components.

- In this a customer generally, borrower will default on debit like credit card and mortgage loan over a time. It basically returns the expected probability of users who fail to repay the loan back to the bank. In this 0% and 100% percentage is used to represent the Probability. Greater probability shows the greater chance of default.

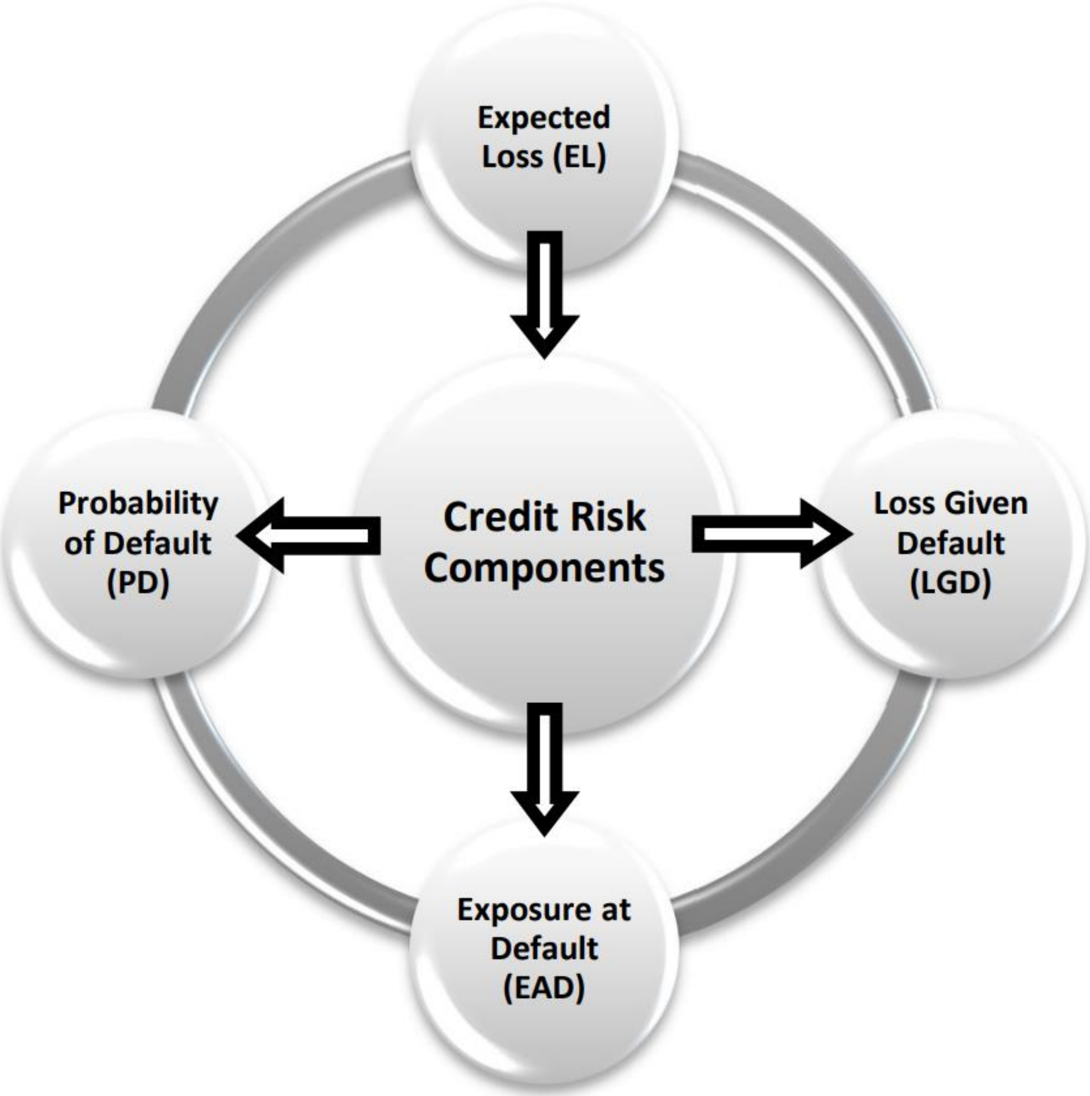


Fig 1.3.1: Credit Risk Components

- The current credit score system has limitations in terms of factors used to determine creditworthiness. It's based on a limited number of factors such as payment history, credit utilization, length of credit history, types of credit, and recent credit inquiries. This oversimplified view may not accurately reflect a borrower's ability to repay a loan. Moreover, the current system lacks customization, meaning it's a one-size-fits-all approach that doesn't consider individual circumstances such as recent job loss or medical emergency that could affect the borrower's ability to repay the loan.

- Another issue with the current credit score system is the concern that it may be biased against certain groups, including minorities or low-income individuals who have limited access to credit, leading to limited opportunities to build a strong credit history. Finally, credit scores are based on credit reports that may contain errors or outdated information. This inaccuracy can lead to an inaccurate credit score classification that negatively impacts a borrower's ability to access credit.

1.4 Proposed System

1.4.1 Overview of proposed system & limitations

Our proposed system aims to make loan decisions based on a customer's credit score. However, the data collected may be inconsistent, which is why we will implement various techniques to clean and prepare the data before building a model.

To ensure the accuracy of our predictive analytics, we will consider several techniques such as handling imbalanced data, conducting exploratory data analysis, and dimensionality reduction. These techniques will help us to better understand the data and improve the accuracy of our model.

Once we have prepared the data, we will build a machine learning model using classification techniques. This model will consider the features we have gathered and make predictions based on those features. By implementing these techniques and processes, we hope to develop a more reliable and accurate loan decision system.

1.5 Aim & purpose of the project

The aim is to provide a brief overview of the use of credit scoring in assessing a person's creditworthiness and likelihood of repaying a loan on time, based on their credit history and its importance in the commercial world and banking industry. It provides an overview of how credit scoring works and its importance in the financial industry and help banks and financial institutions in decision-making regarding issuing loans to customers.

The system will take input data such as income, age, employment status, credit history, and other relevant information to generate a credit score that can be used to assess the likelihood of loan repayment.

The ultimate purpose of this project is to help lenders make more informed decisions about loan approvals and reduce the risk of default. The project also aims to integrate the credit score classification system with a web API for easy access and scalability.

1.6 Scope of the project

The scope of the project is to develop a web application that integrates credit worthiness rating model and to automate the loan approval process. The project aims to leverage machine learning algorithms to analyze the past credit history of an applicant and provide an instant rating, which helps banks and financial institutions make informed decisions about granting loans.

The project also includes the development of a web API that can be used to integrate the credit rating system into machine learning model. The project's scope is to provide a reliable and efficient system that reduces the turnaround time for loan approvals, making the process more efficient and user-friendly for both the financial institution and the applicant.

1.7 Objectives

- The prime objective is to find out the credit score for all the existing customers from existing data and help lenders to allocate funds for a non-default user by making use of good prediction.

- To describe the use of credit scoring as a method to assess a person's creditworthiness and likelihood of repaying a loan on time, based on their credit history.

- To explain how credit scoring models are used in the commercial world, particularly in banking, to identify profitable customers and make intelligent decisions.

- To provide an overview of how credit scoring works and its importance in the financial industry.

2. SYSTEM REQUIREMENT SPECIFICATIONS

2.1 Purpose of the System

The purpose of the system is to provide a way to decide whether to issue loans to customers based on their credit score. The system uses advanced statistical and mathematical methods, including machine learning algorithms and artificial intelligence, to analyze a consumer's credit history and make predictions about their likelihood of repaying a loan on time. The goal is to help financial institutions make more informed decisions about providing credit and minimize the risk of default. The proposed system aims to address some of the limitations of the existing credit scoring system, such as limited factors, lack of customization, bias, and data inaccuracy.

2.2 Feasibility analysis

Feasibility analysis of credit score classification involves assessing the practicality and viability of implementing a credit scoring system. Firstly, it is important to determine whether there is a need for a credit scoring system. If there is a high demand for credit services, then implementing a credit scoring system can help financial institutions manage risk and make better lending decisions.

Secondly, it is important to assess the availability and quality of data. Credit scoring models rely on historical credit data, so the quality of the data is crucial. If the data is incomplete, inaccurate or outdated, then the credit scoring system may not be effective.

Thirdly, it is important to consider the cost of implementing a credit scoring system. Developing and maintaining a credit scoring system can be expensive, and the costs may outweigh the benefits if the system is not used frequently or if there are other, cheaper alternatives available.

Fourthly, it is important to consider the legal and regulatory environment. Depending on the country or region, there may be laws and regulations governing the collection and use of credit data, and it is important to ensure that the credit scoring system complies with these regulations.

Lastly, it is important to assess the potential impact of the credit scoring system on individuals and society. Credit scoring can have both positive and negative effects on individuals, and it is important to consider these effects in the feasibility analysis.

Overall, the feasibility analysis of credit score classification involves assessing the need for a credit scoring system, the availability and quality of data, the cost of implementation, the legal and regulatory environment, and the potential impact on individuals and society.

2.3 Hardware Requirements

Processor	: intel dual core, i3 or higher generation
HDD	: 80 Gb or higher
RAM	: 4 Gb or higher

2.4 Software Requirements

Operating System	: Windows 7 or higher
Programming Language	: Python 3.7
IDE	: Jupiter Notebook, Spyder
Python packages	: Pandas, Numpy, Plotly, Scikit-learn, Flask

2.5 Functional Requirements

Functional requirements describe the specific features and capabilities that a system or software must have to meet the needs of its users.

- The system should allow users to input personal and financial information for credit score calculation.
- The system should be able to process and analyze the input data using machine learning algorithms to generate a credit score.
- The system should provide users with an easy-to-understand credit score report that includes the factors affecting their score and recommendations for improving it.
- The system should allow users to securely store their personal and financial information.
- The system should have a user authentication and authorization feature to ensure secure access to the credit score report.
- The system should be scalable to handle a large number of users and data processing requests.
- The system should be reliable and available at all times to ensure users can access their credit score report whenever needed.
- The system should be compliant with relevant regulations and standards for data privacy and security.
- The system should be compatible with various devices and web browsers to enable a seamless user experience.

2.6 Non- Functional Requirements

Non-functional requirements are attributes or qualities that a system should possess to perform efficiently and provide value to its users.

- **Performance**: The system should have fast response times, be able to handle a large volume of data, and have a high level of availability

- **Security**: The system should be designed with security in mind to protect sensitive customer information and prevent fraud.

- **Usability**: The system should be easy to use and navigate for users, with clear instructions and error messages.

- **Scalability**: The system should be able to handle increasing amounts of data as the user base grows.

- **Compatibility**: The system should be compatible with different platforms, operating systems, and browsers.

- **Accessibility**: The system should be accessible to users with disabilities, with support for assistive technologies and other accessibility features.

3. ABOUT THE SOFTWARE

3.1 Pandas

A pandas is an open-source library licenced under the Berkeley Software Distribution (BSD). In the domain of data science, this well-known library is widely used. They're mostly used for analysis, manipulation, and cleaning of data, among other things. Pandas allow us to perform simple data modelling and analysis without having to swap to another language like R, Pandas is an open-source library licenced under the Berkeley Software Distribution (BSD). In the domain of data science, this well know library is widely used. They're mostly used for analysis, manipulation, and cleaning of data, among other things. Pandas allow us to perform simple data modelling and analysis without having to switch to another language like R.

3.2 Numpy

NumPy is a widely used Python library for scientific computing that provides powerful and efficient tools for working with arrays and numerical operations. It was first released in 2006, and since then, it has become one of the most important libraries for data analysis and scientific research.

The main feature of NumPy is its ability to work with multidimensional arrays and perform complex mathematical operations on them. This makes it ideal for tasks such as data analysis, machine learning, image processing, and scientific computing. NumPy provides a wide range of functions for basic arithmetic operations, linear algebra, Fourier transforms, and random number generation.

NumPy is designed to be fast and efficient, and it is optimized for large datasets and high-performance computing. It is also open-source, which means that it is freely available for use and modification by anyone.

3.3 Sckit-learn

Scikit-learn is an open-source machine learning library based on Python. Both supervised and unsupervised learning processes can be used in this library. Popular algorithms and the SciPy, NumPy, and Matplotlib packages are all already precluded in this library. The most well-known Scikit-learn application is for Spotify music recommendations.

3.4 Plotly

The plotly python library is an interactive open-source library. This can be a very helpful tool for data visualization and understanding the data simply and easily. plotly graph objects are a high-level interface to plotly which are easy to use.

It can plot various types of graphs and charts like scatter plots, line charts, bar charts, box plots, histograms, pie charts, etc. Plotly has hover tool capabilities that allow us to detect any outliers or anomalies in a large number of data points. It is visually attractive that can be accepted by a wide range of audiences. It allows us for the endless customization of our graphs that makes our plot more meaningful and understandable for others.

3.5 Flask

Flask is a Python-based micro web framework that enables developers to create small to medium-sized web applications quickly and easily. It provides a simple and flexible architecture that is easy to learn, making it an ideal choice for building prototypes, small-scale applications, and RESTful APIs.

Flask includes features like URL routing, templating, debugging, and extension support, which makes it easy for developers to create dynamic web pages and RESTful APIs. Flask is widely used in various industries, including education, healthcare, finance, and e-commerce, and has proven to be a reliable and flexible solution for building web applications.

4. SYSTEM ANALYSIS AND REQUIREMENTS DOCUMENTATION

4.1 Overview

System analysis is the process of studying a system to identify its components, behaviours, and interactions. It involves analysing the existing system and identifying its strengths and weaknesses, and then proposing improvements or a new system that better meets the needs of the organization or user. In the context of this project, the system analysis would involve examining the current credit scoring process and identifying areas where it can be improved, such as by incorporating new data sources or using more advanced machine learning techniques.

The requirements for this project would include both functional and non-functional requirements. Functional requirements would specify the system's capabilities, such as the ability to analyse credit history data and generate a credit score. Non-functional requirements would specify the system's characteristics, such as its performance, reliability, and security. For example, the system would need to be able to process large amounts of data quickly and securely, and it would need to comply with relevant regulations and standards.

To develop a successful credit scoring system, it would be important to involve stakeholders from various areas, such as finance, data science, and IT. Their input would be valuable in understanding the requirements and constraints of the system and in ensuring that the system meets the needs of all stakeholders. The requirements would need to be carefully documented and managed throughout the development process to ensure that the system meets its intended goals and objectives.

4.2 Proposed System Architecture

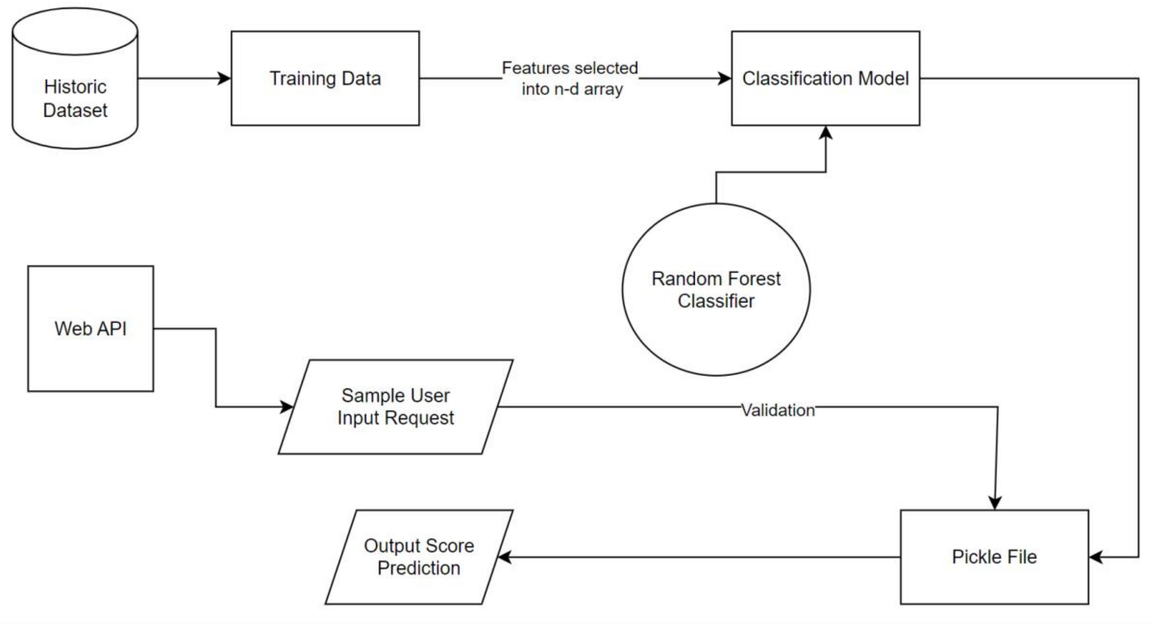


Fig 4.2: System Architecture of the Proposed System

4.3 Modules Description

4.3.1 Credit Scoring Analysis

The credit scoring analysis involves various steps, including data collection, cleaning, pre-processing, and feature engineering. The process is complex and iterative and requires technical expertise, domain knowledge, and business acumen.

Using the right tools and techniques, lenders and financial institutions can make informed credit decisions that balance risk and reward. Ultimately, the credit scoring analysis plays an essential role in managing credit risk and maintaining a healthy lending portfolio.

Methodology:

Input: The input dataset is taken where the data is collected from kaggle. Then further cleaning and preprocessing techniques takes part in this module.

Output: The output of this module will be the filtered data itself where all the techniques are applied in feature engineering.

Process: It is a process used by lenders and financial institutions to assess the creditworthiness of an individual or company. This evaluation is based on factors such as credit history, financial behavior, and other relevant information. The primary goal of this analysis is to determine the probability of a borrower defaulting on a loan or credit line.

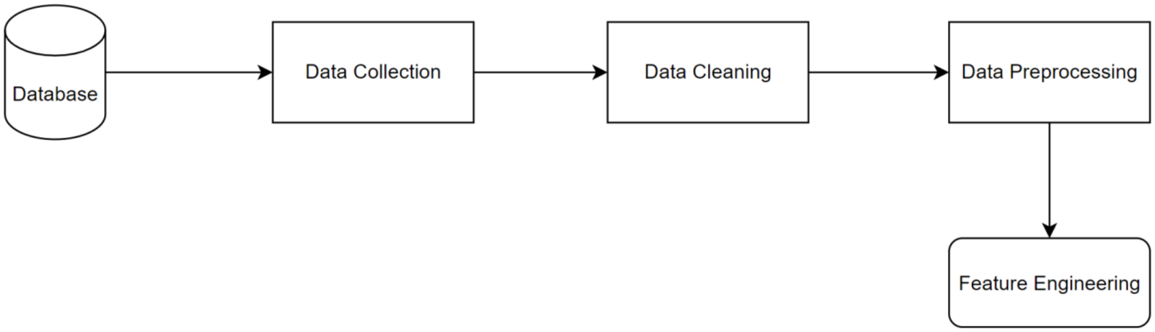


Fig 4.3.1: Module 1 System Design

4.3.2 Credit History Data Representation

It discusses use of various data visualization techniques to represent credit history data on plots. One of the useful visualization techniques discussed in the text is the use of box plots. Box plots are used to represent the distribution of numerical data by showing their quartiles, outliers, and minimum and maximum values.

It further highlights that box plots are helpful in identifying the distribution of data and any potential outliers or anomalies. In the case of credit history data, box plots can be used to represent the distribution of numerical features such as credit limit, age, and monthly income. Other numerical features such as age or monthly income can also be represented using box plots to provide further insights into the credit history of customers.

Lastly, the text mentions that a major step in credit history data representation involves feature selection, which is necessary for building models. This means that selecting the most relevant features from the data analysis is crucial in building effective credit history models.

Methodology:

Input: This input can be used to create a box plot showing the distribution of credit limit values for the sample of customers. The data with credit limit values are used for representation.

Output: The output resulting visualization can help identify the range of credit limits and any potential outliers or anomalies in the data.

Process: This module discusses the use of data visualization techniques to represent credit history data, with a focus on box plots. Box plots are a useful tool for showing the distribution of numerical data by displaying quartiles, outliers, and minimum and maximum values. They can be used to identify the distribution of data and any potential anomalies or outliers. In the context of credit history data, box plots can be used to represent numerical features such as credit limit, age, and monthly income. They provide insights into the distribution of these features among customers and can help in identifying patterns and trends in the data.

It emphasizes the importance of feature selection in credit history data representation for model building. This involves selecting the most relevant features from the data analysis to build effective credit history models. By combining data visualization techniques like box plots with feature selection, analysts can gain a deeper understanding of the credit history data and build more accurate and effective models. Overall, data visualization and feature selection are important steps in understanding credit history data and building models that can help in making informed decisions.

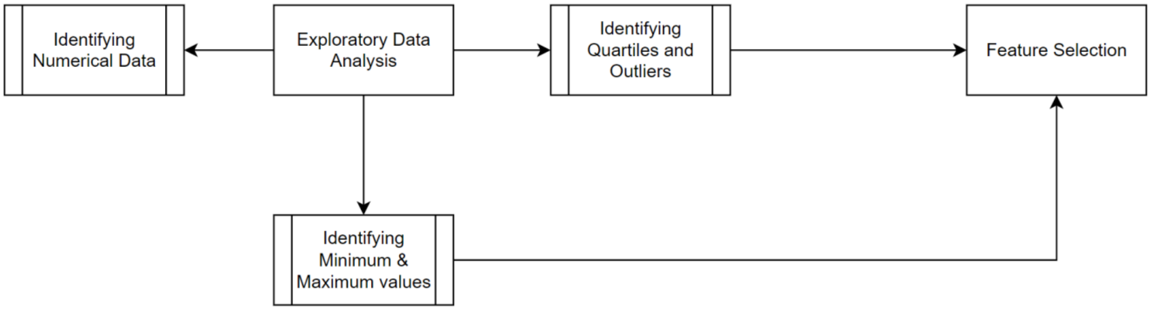

Identifying Numerical Data
Exploratory Data Analysis
Identifying Quartiles and Outliers
Feature Selection
Identifying Minimum & Maximum values

Fig 4.3.2: Module 2 System Design

4.3.3 Risk Classification Of Credit Score

The module explains the use of the Random Forest algorithm for credit risk classification, which is a popular technique in machine learning for identifying and predicting credit risk. This ensemble learning algorithm combines multiple decision trees to improve the accuracy of the model and can handle large amounts of data with many features. It can also provide important insights into the most important features that contribute to credit risk and help inform credit decision-making processes.

The module highlights the importance of the Credit Mix feature in determining credit scores and explains how it is transformed into a numerical feature for training a machine learning model. The data is split into features and labels, with the important features selected for the model. Finally, the data is split into training and test sets and used to train a credit score classification model. Overall, the module demonstrates how machine learning techniques like Random Forest can be used to build effective credit scoring models and inform credit risk assessment processes.

Methodology:

Input: The input for this module is the process of transforming the Credit Mix feature into a numerical feature for training a machine learning model

Output: The Credit Mix feature is identified as an important feature for credit score determination. The data is split into training and test sets and used to train the model, which can provide insights into the most important features that contribute to credit risk and inform credit decision-making processes.

Process: The module describes the process of using Random Forest algorithm for credit risk classification, which involves training the model on data by selecting important features such as Credit Mix, transforming categorical features into numerical features, and splitting the data into training and test sets to build an effective credit scoring model that can account for non-linear relationships between features and the target variable, providing important insights into the most important features that contribute to credit risk and informing credit decision-making processes.

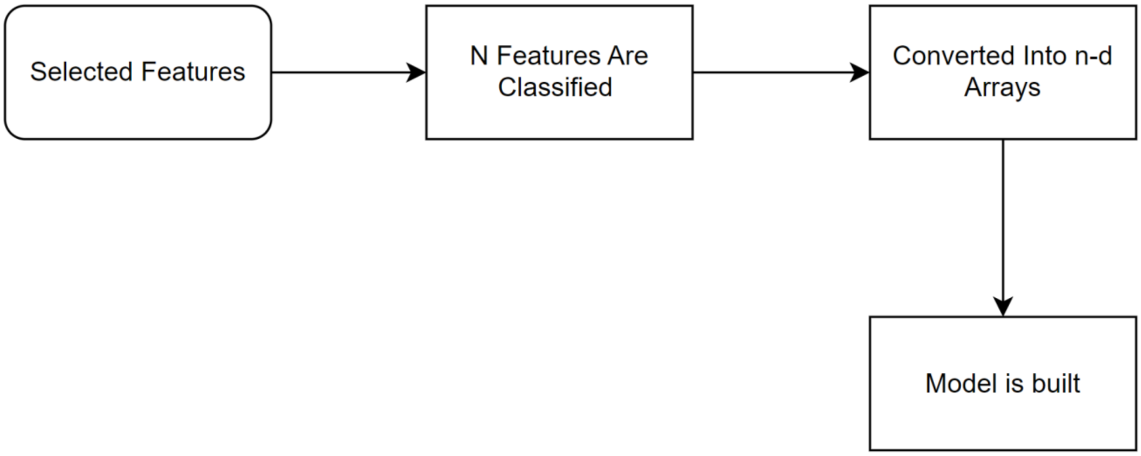


Fig 4.3.3: Module 3 System Design

4.3.4 Creditworthiness Rating Model Integrated With Web API

This module explains the concept of credit worthiness rating, which is an assessment model used by lenders to evaluate a borrower's ability to repay debts on time. The rating is expressed as a credit score, and is based on various factors such as credit history, income, payment history, credit utilization, length of credit history, types of credit, and new credit accounts. Integrating a creditworthiness rating system with a web application streamlines the lending process, reduces risks, and improves access to credit for borrowers, making it easier for both lenders and borrowers to manage the loan application and evaluation process.

Methodology:

Input: The input to this module is input data from web API

Output: The output of this module is calculated credit score

Process: The module provides an overview of credit worthiness rating, a system used by lenders to assess the credit risk associated with a borrower, and the factors that contribute to the credit score of a borrower. It also explains the benefits of integrating a creditworthiness rating system with a web application, which improves the lending process by streamlining it, reducing risks, and making it easier for both lenders and borrowers to manage the loan application and evaluation process. Overall, the module presents a concise and informative description of credit worthiness rating and its role in the lending process.

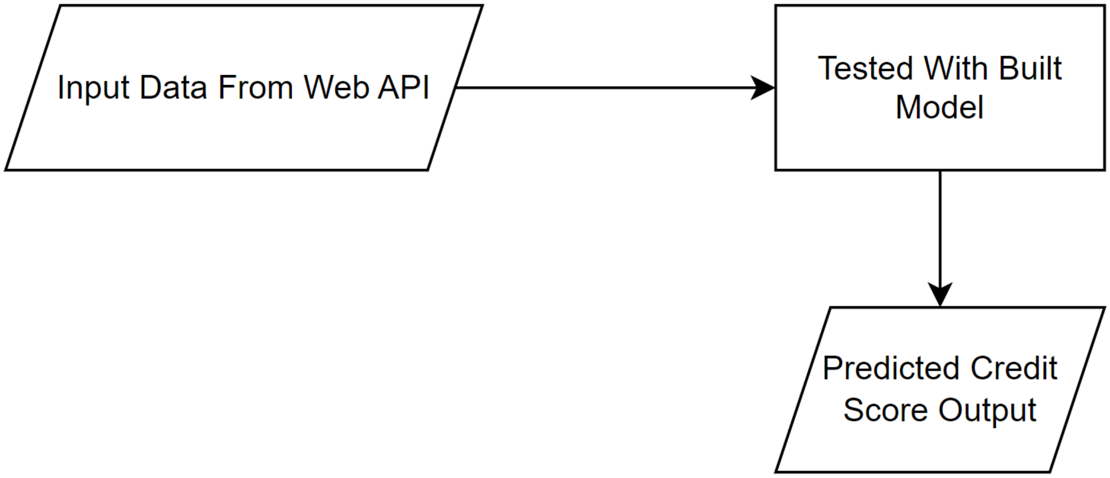


Fig 4.3.4: Module 4 System Design

4.4 UML Diagrams

UML stands for Unified Modeling Language. The purpose of a use case diagram is to capture the dynamic aspect of a system. It is used to gather the requirements of a system. It used to be possible to get an outside view of a system. Identify the external and internal factors influencing the system.

UML diagrams are categorized into two main groups: structural diagrams and behavioral diagrams.

4.4.1 Use case diagram

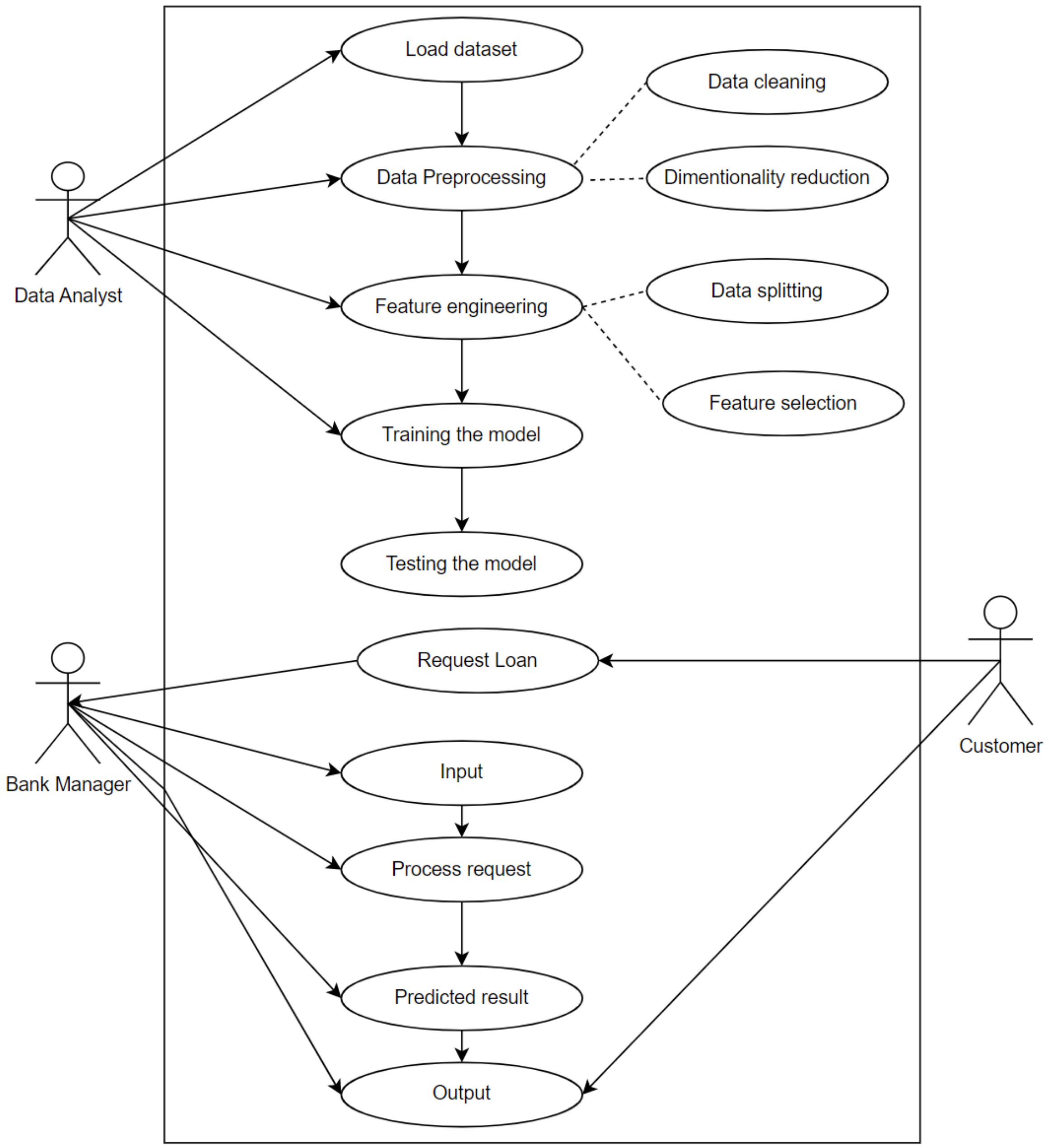


Fig 4.4: Use Case diagram of the proposed system

4.4.2 Sequence Diagram

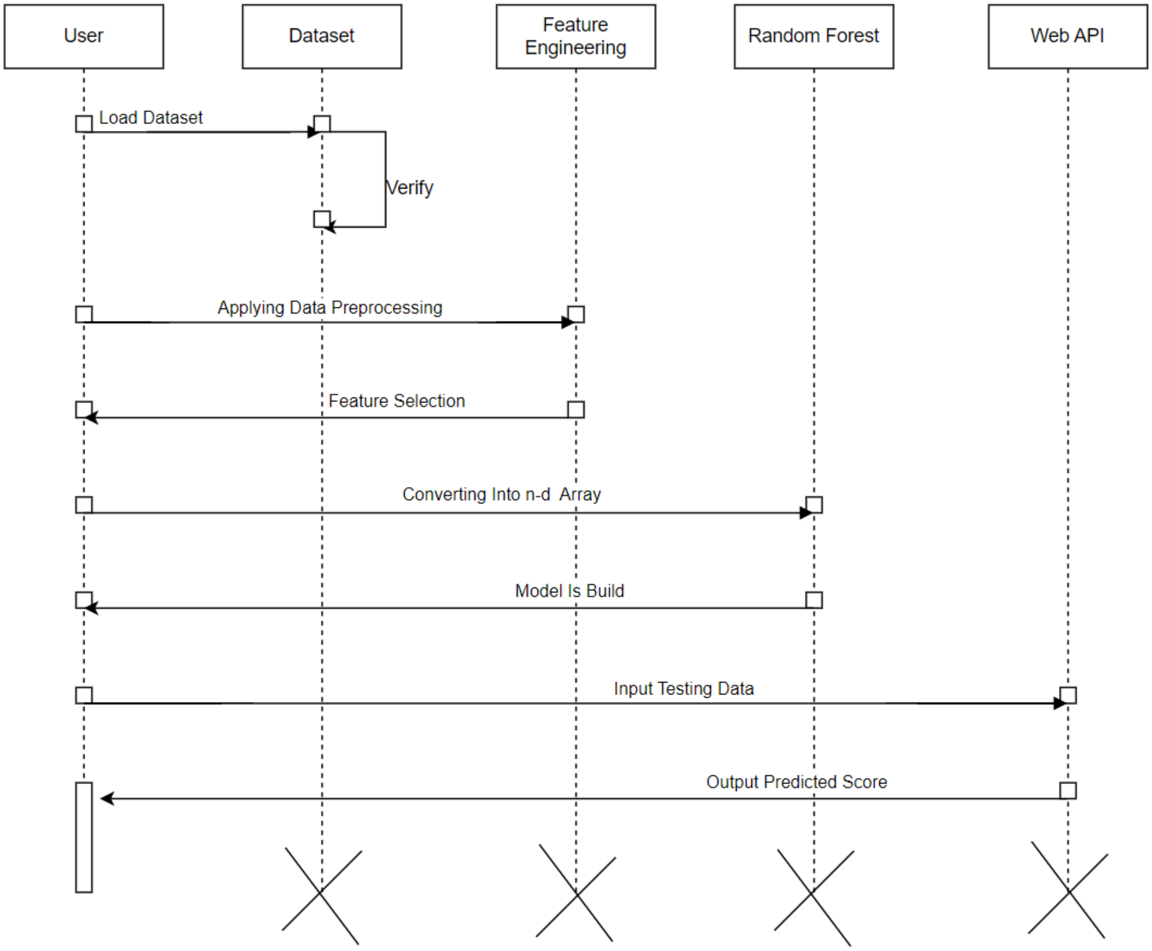


Fig 4.4.2: Sequence diagram of the proposed system

4.4.3 State Chart Diagram

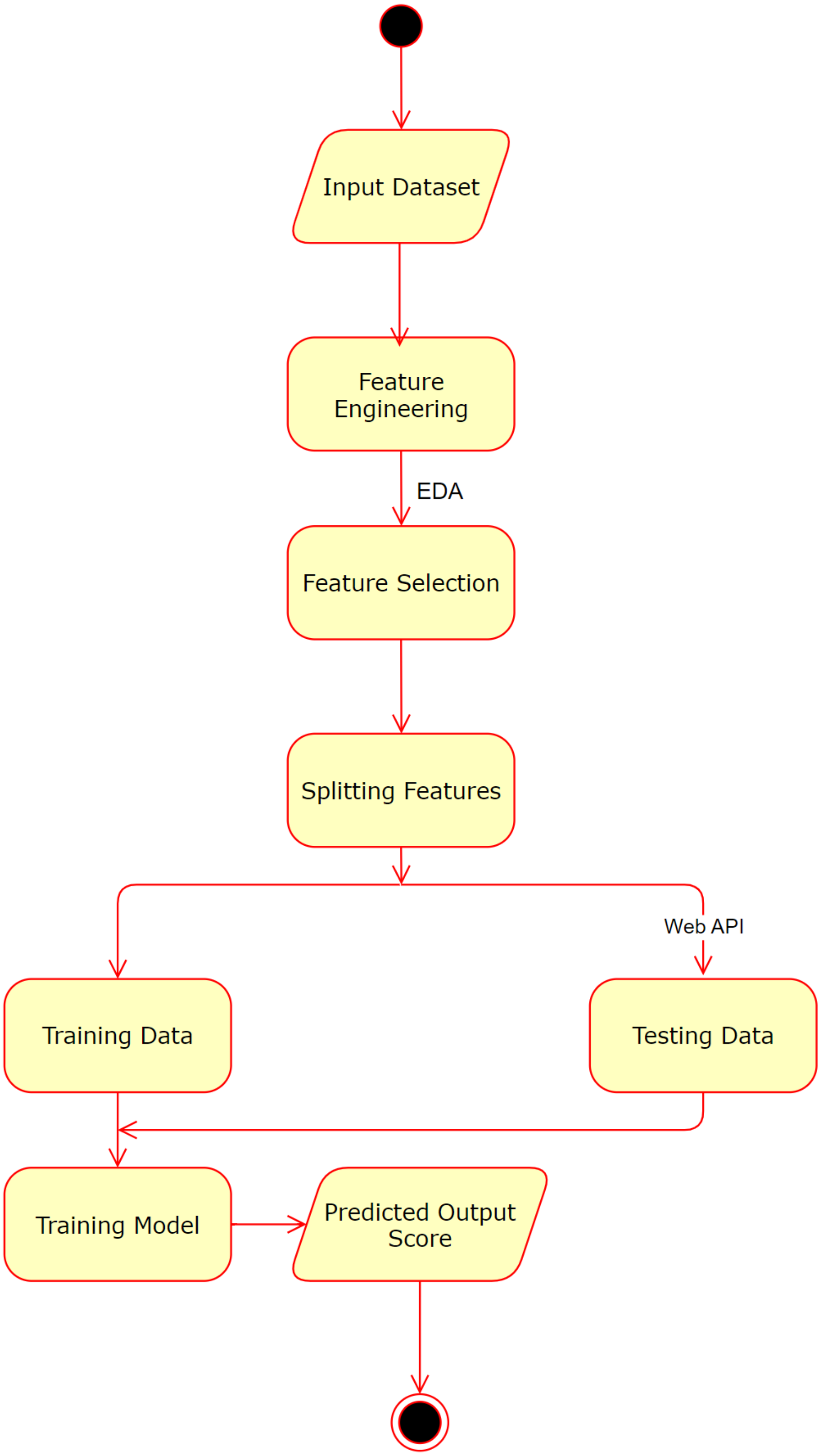


Fig 4.4.3: State Chart diagram of the proposed system

4.5 Structured Diagrams

A structure diagram is a conceptual modeling tool used to document the different structures that make up a system such as a database or an application. It shows the hierarchy or structure of the different components or modules of the system and shows how they connect and interact with each other.

A structure diagram visualizes how a system works from the initial input, to processing and, finally, to the desired output. It is especially useful in determining all the interfaces involved between the different parts and helps developers agree on how each part should be connected based on the models being shown on the structure diagram.

4.5.1 Class diagram

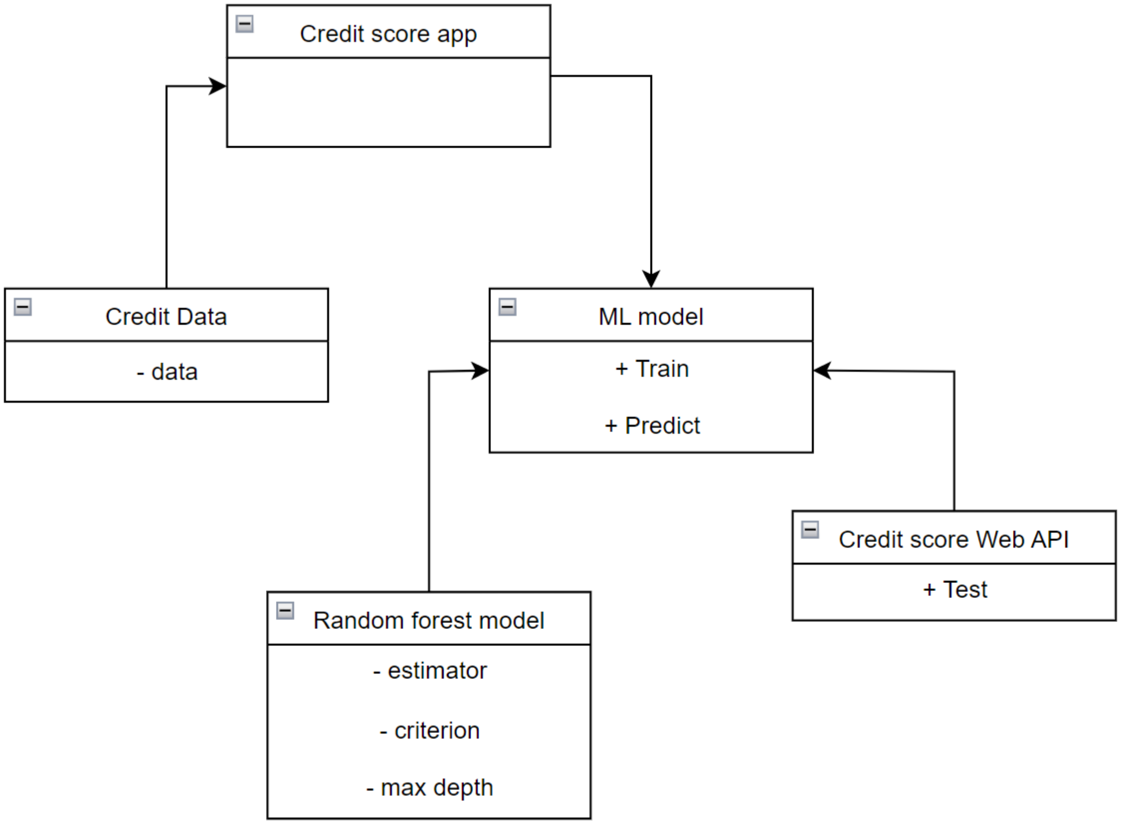


Fig 4.5.1: Class diagram of the proposed system

5. System Design and Documentation

5.1 Data Collection

The first step in any machine learning project is to collect the data that will be used to train the model. In this case, the data will include information about borrowers, such as their credit history, income, and employment status.

5.2 Data Pre-processing

Once the data is collected, it will need to be pre-processed to clean and transform it into a format that can be used by the machine learning algorithms. This may include tasks such as removing missing values, encoding categorical variables, and scaling numerical features.

5.3 Feature engineering

In this step, the data is transformed to extract features that are relevant and meaningful for the credit score classification task. This can include creating new features from existing ones or selecting the most important features using techniques like correlation analysis or feature importance rankings.

5.4 Model Training

This step involves using the training data to train the chosen machine learning model. The model learns to identify patterns and relationships between the features and the target variable, which in this case is the credit score category.

5.5 Model Evaluation

Once the model is trained, it needs to be evaluated to ensure it is performing well. This may involve splitting the data into training and testing sets, using metrics such as accuracy, precision, and recall to evaluate the model's performance.

5.6 Model Deployment

After the model is trained and evaluated, it is deployed in a production environment where it can be used to classify credit scores in real-time. This can involve integrating the model with other systems and ensuring it is performing as expected in the live environment.

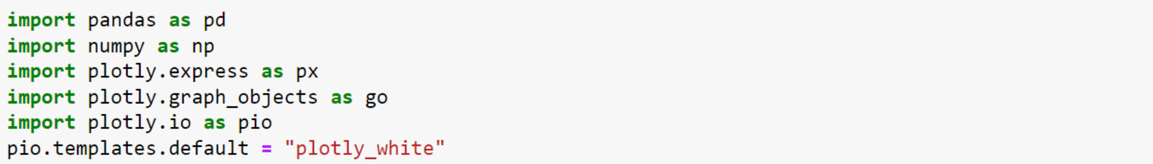
5.7 Integration

Integration with a web API is a critical step in deploying machine learning models to production. This step involves creating an endpoint for the model, which can be accessed by external applications through HTTP requests.

6. Code and Implementation

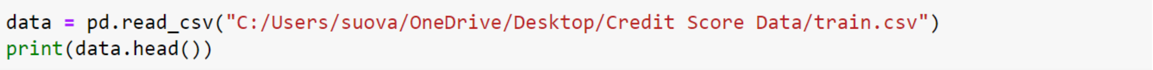
6.1 Python Implementation:

Import Section:



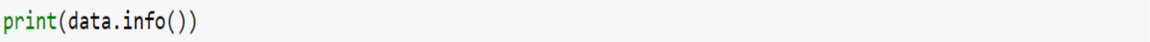
```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
pio.templates.default = "plotly_white"
```

Reading dataset:

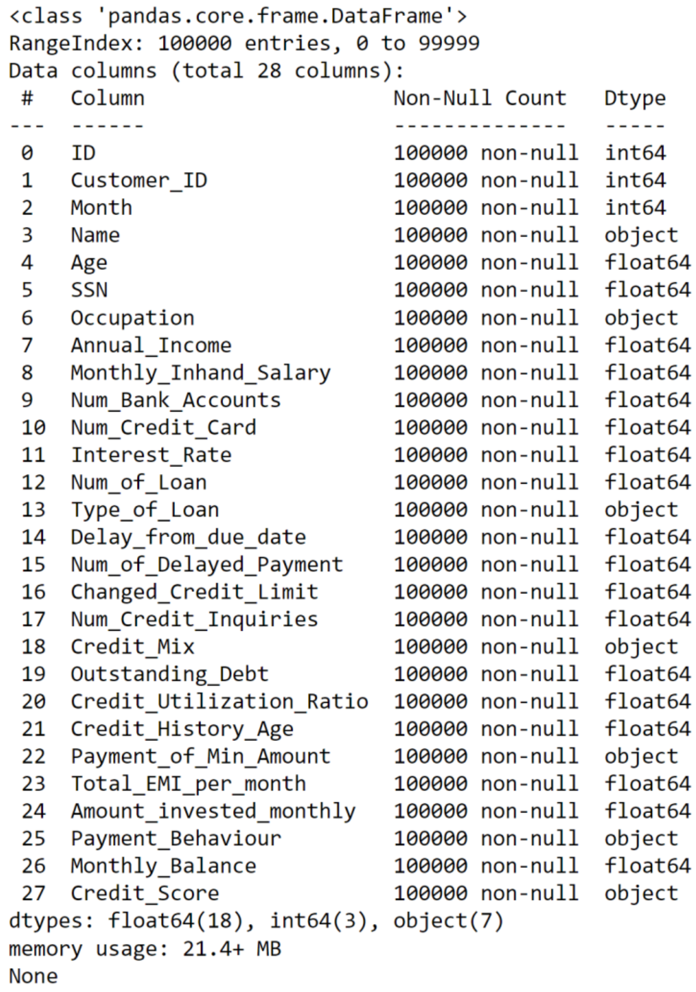


```
data = pd.read_csv("C:/Users/suova/OneDrive/Desktop/Credit Score Data/train.csv")
print(data.head())
```

Displaying columns in the dataset:

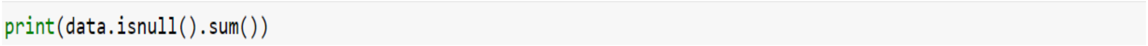


```
print(data.info())
```

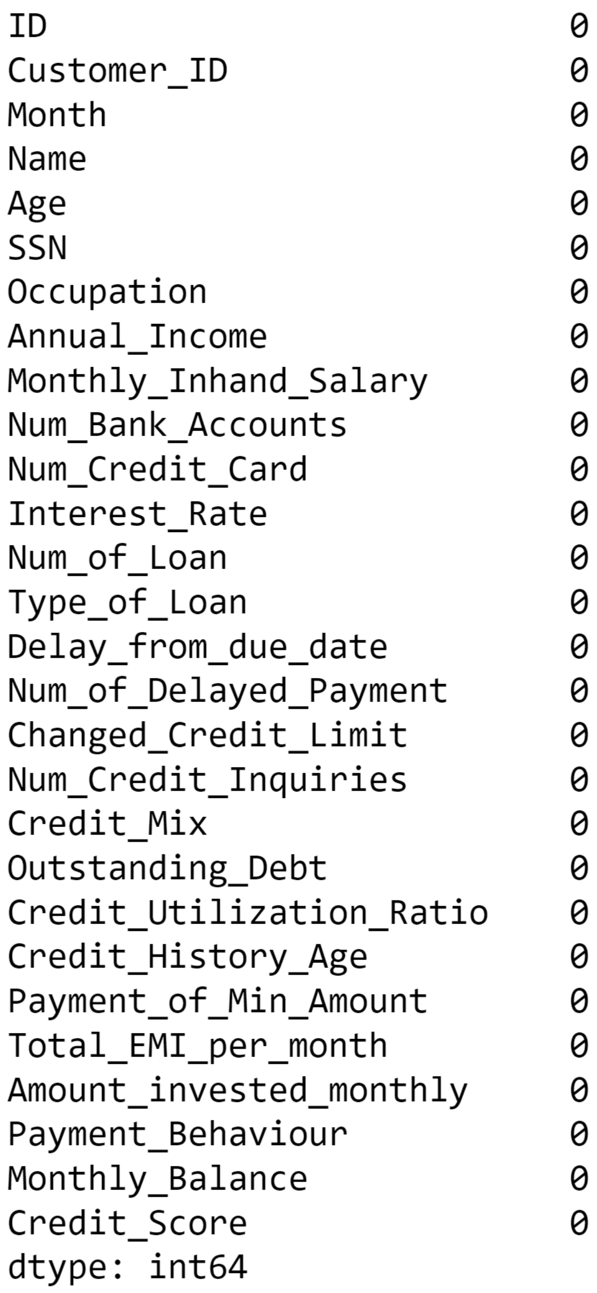


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 28 columns):
 #   Column                    Non-Null Count   Dtype  
---  ------                    --------------   -----  
 0   ID                        100000 non-null  int64  
 1   Customer_ID               100000 non-null  int64  
 2   Month                     100000 non-null  int64  
 3   Name                      100000 non-null  object 
 4   Age                       100000 non-null  float64
 5   SSN                       100000 non-null  float64
 6   Occupation                100000 non-null  object 
 7   Annual_Income             100000 non-null  float64
 8   Monthly_Inhand_Salary     100000 non-null  float64
 9   Num_Bank_Accounts         100000 non-null  float64
 10  Num_Credit_Card           100000 non-null  float64
 11  Interest_Rate             100000 non-null  float64
 12  Num_of_Loan               100000 non-null  float64
 13  Type_of_Loan              100000 non-null  object 
 14  Delay_from_due_date       100000 non-null  float64
 15  Num_of_Delayed_Payment    100000 non-null  float64
 16  Changed_Credit_Limit      100000 non-null  float64
 17  Num_Credit_Inquiries      100000 non-null  float64
 18  Credit_Mix                100000 non-null  object 
 19  Outstanding_Debt          100000 non-null  float64
 20  Credit_Utilization_Ratio  100000 non-null  float64
 21  Credit_History_Age        100000 non-null  float64
 22  Payment_of_Min_Amount     100000 non-null  object 
 23  Total_EMI_per_month       100000 non-null  float64
 24  Amount_invested_monthly   100000 non-null  float64
 25  Payment_Behaviour         100000 non-null  object 
 26  Monthly_Balance           100000 non-null  float64
 27  Credit_Score              100000 non-null  object 
dtypes: float64(18), int64(3), object(7)
memory usage: 21.4+ MB
None
```

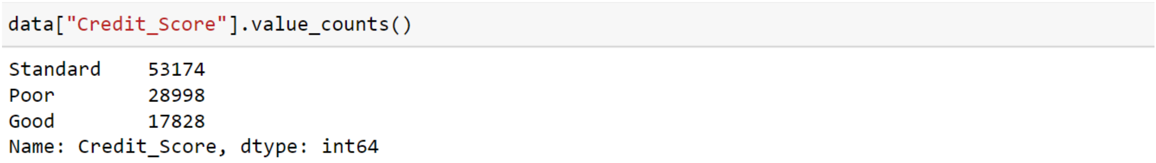
Checking the values are null or not:



```
print(data.isnull().sum())
```



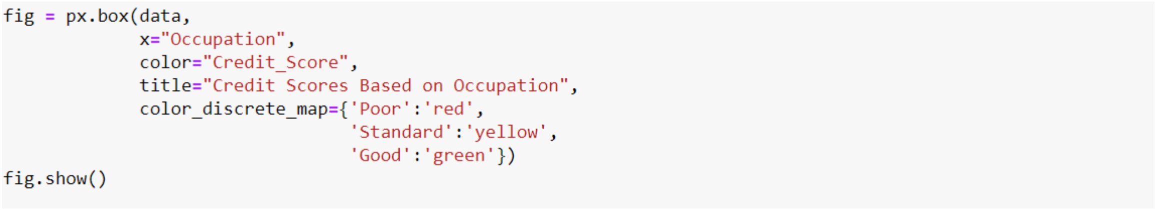
```
ID                          0
Customer_ID                 0
Month                       0
Name                        0
Age                         0
SSN                         0
Occupation                  0
Annual_Income               0
Monthly_Inhand_Salary       0
Num_Bank_Accounts           0
Num_Credit_Card             0
Interest_Rate               0
Num_of_Loan                 0
Type_of_Loan                0
Delay_from_due_date         0
Num_of_Delayed_Payment      0
Changed_Credit_Limit        0
Num_Credit_Inquiries        0
Credit_Mix                  0
Outstanding_Debt            0
Credit_Utilization_Ratio    0
Credit_History_Age          0
Payment_of_Min_Amount       0
Total_EMI_per_month         0
Amount_invested_monthly     0
Payment_Behaviour           0
Monthly_Balance             0
Credit_Score                0
dtype: int64
```



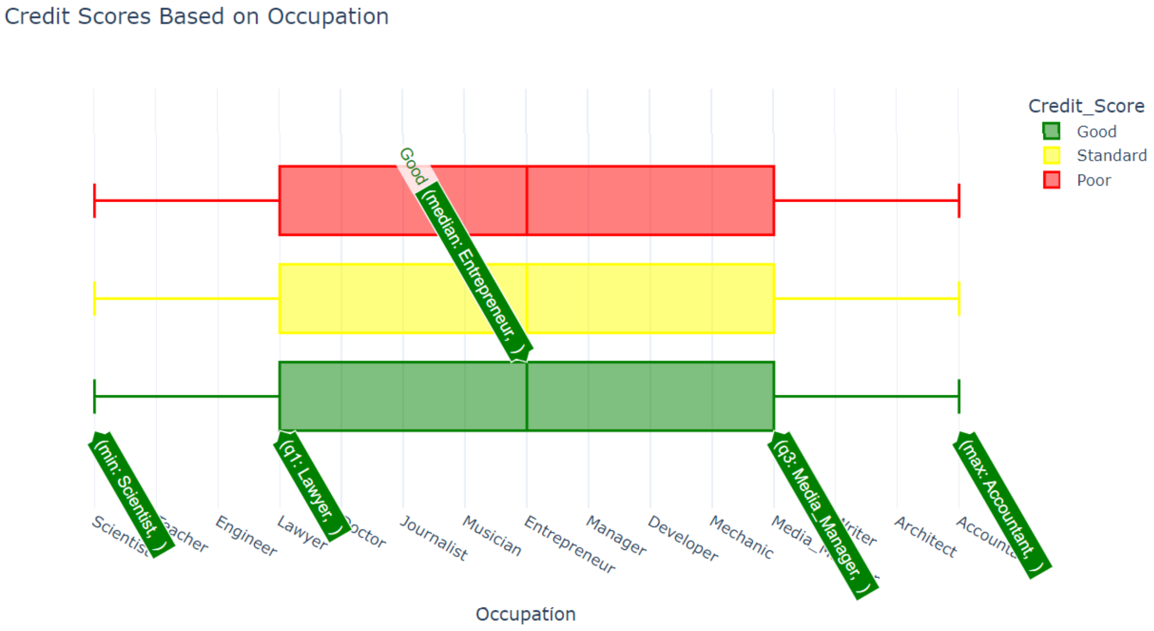
```
data["Credit_Score"].value_counts()
```

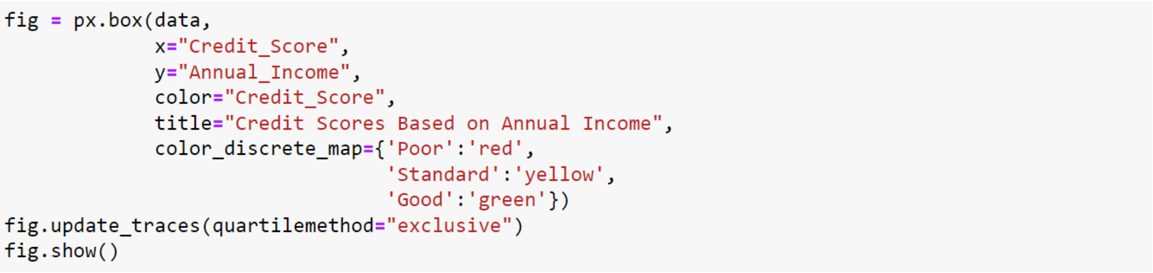
```
Standard    53174
Poor        28998
Good        17828
Name: Credit_Score, dtype: int64
```

Data Exploration:

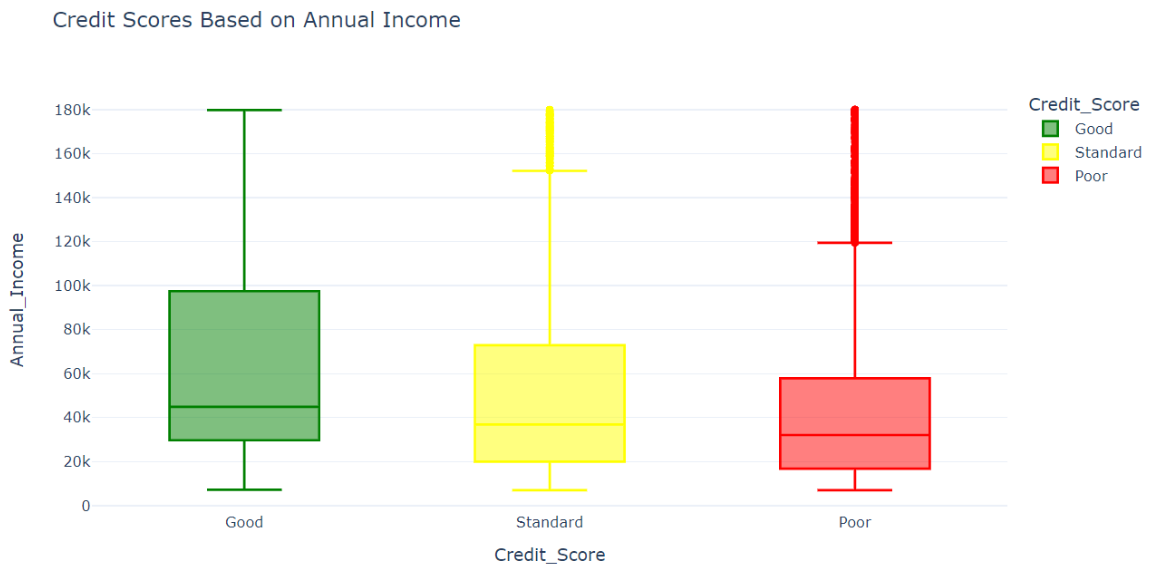


```
fig = px.box(data,
             x="Occupation",
             color="Credit_Score",
             title="Credit Scores Based on Occupation",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.show()
```

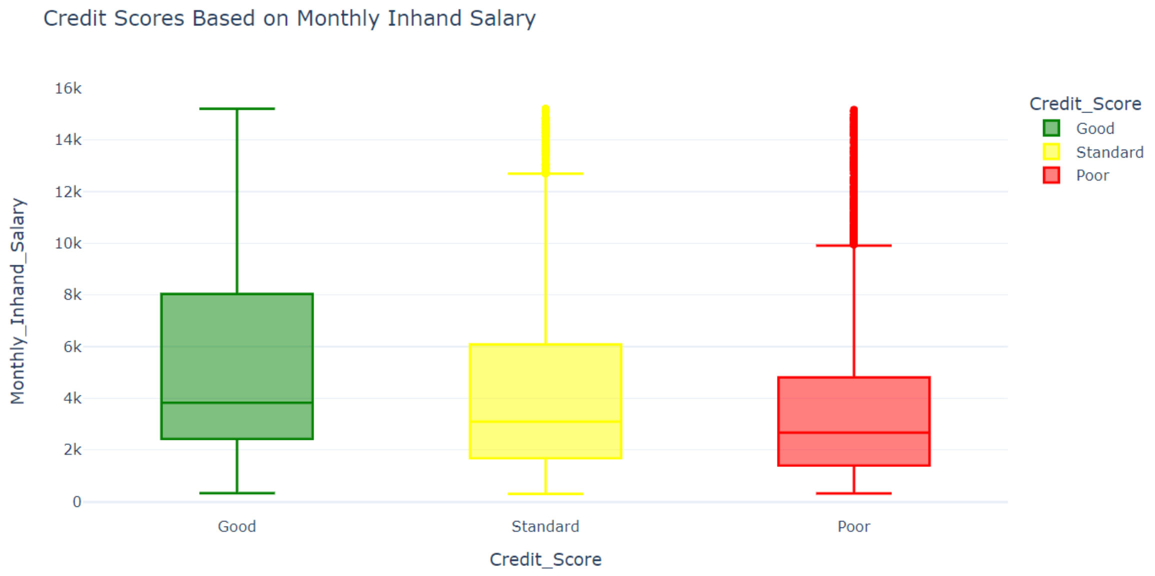




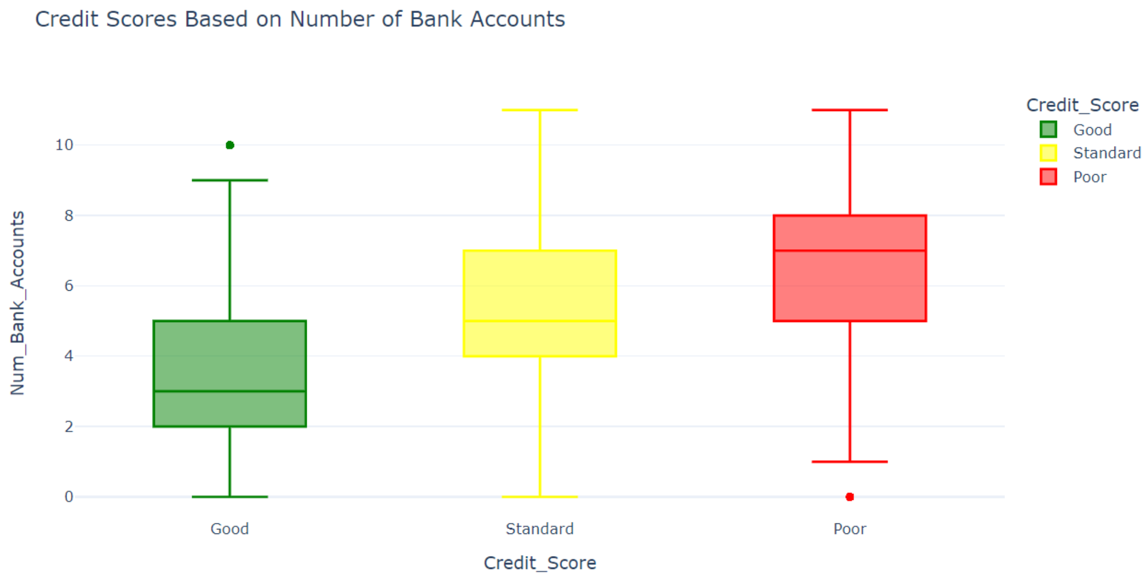
```
fig = px.box(data,
             x="Credit_Score",
             y="Annual_Income",
             color="Credit_Score",
             title="Credit Scores Based on Annual Income",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



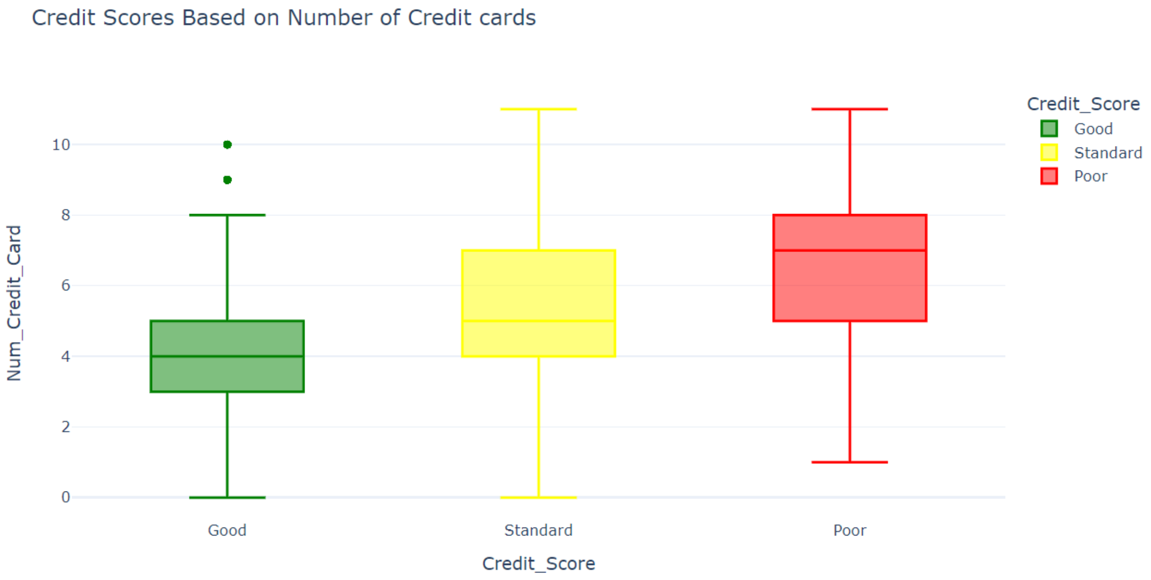
```
fig = px.box(data,
             x="Credit_Score",
             y="Monthly_Inhand_Salary",
             color="Credit_Score",
             title="Credit Scores Based on Monthly Inhand Salary",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



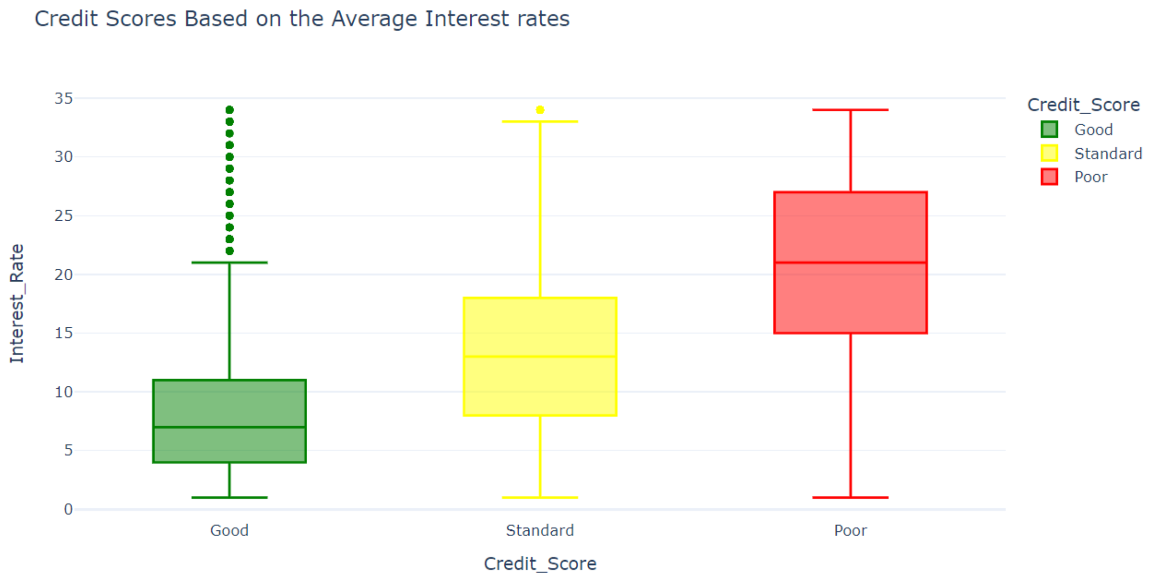
```
fig = px.box(data,
             x="Credit_Score",
             y="Num_Bank_Accounts",
             color="Credit_Score",
             title="Credit Scores Based on Number of Bank Accounts",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



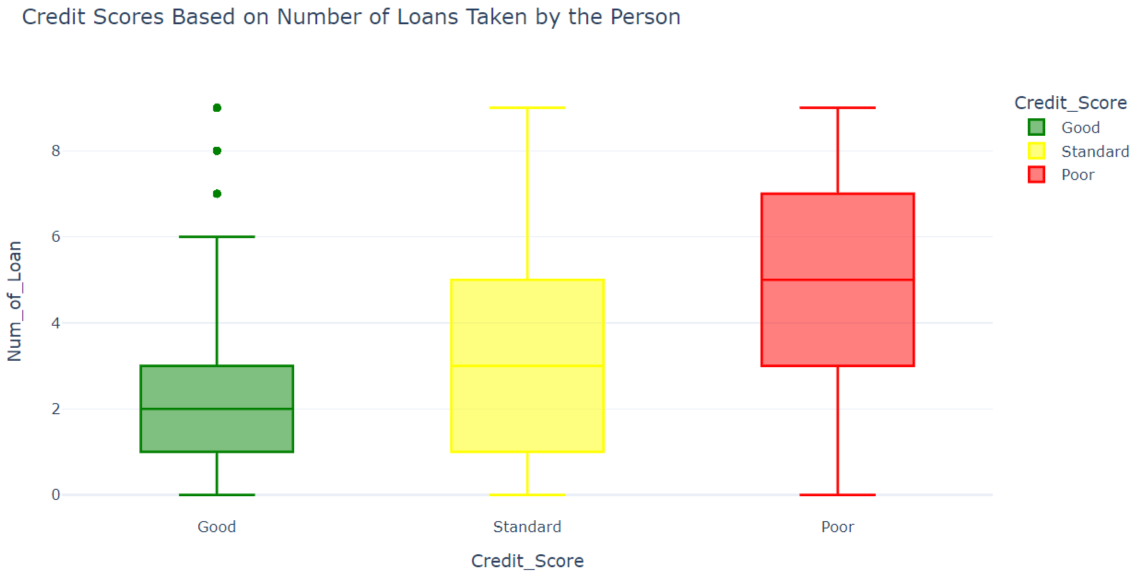
```
fig = px.box(data,
             x="Credit_Score",
             y="Num_Credit_Card",
             color="Credit_Score",
             title="Credit Scores Based on Number of Credit cards",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



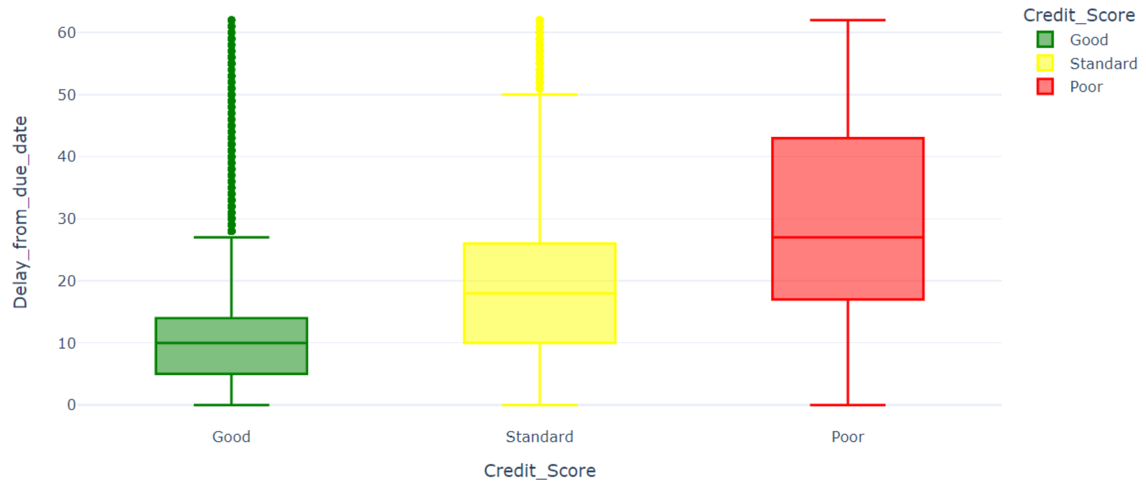
```
fig = px.box(data,
             x="Credit_Score",
             y="Interest_Rate",
             color="Credit_Score",
             title="Credit Scores Based on the Average Interest rates",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



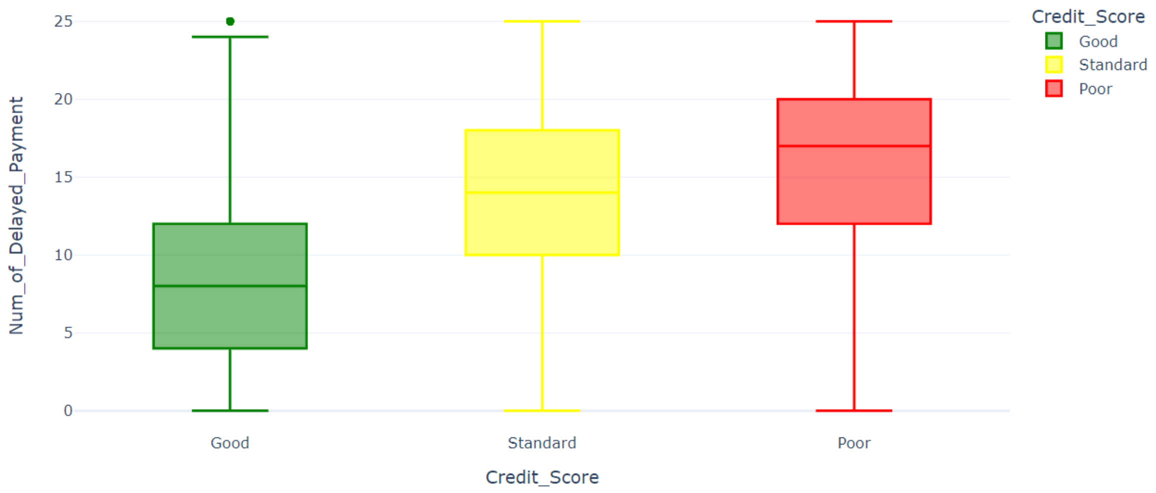
```
fig = px.box(data,
             x="Credit_Score",
             y="Num_of_Loan",
             color="Credit_Score",
             title="Credit Scores Based on Number of Loans Taken by the Person",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```


Credit Scores Based on Number of Loans Taken by the Person

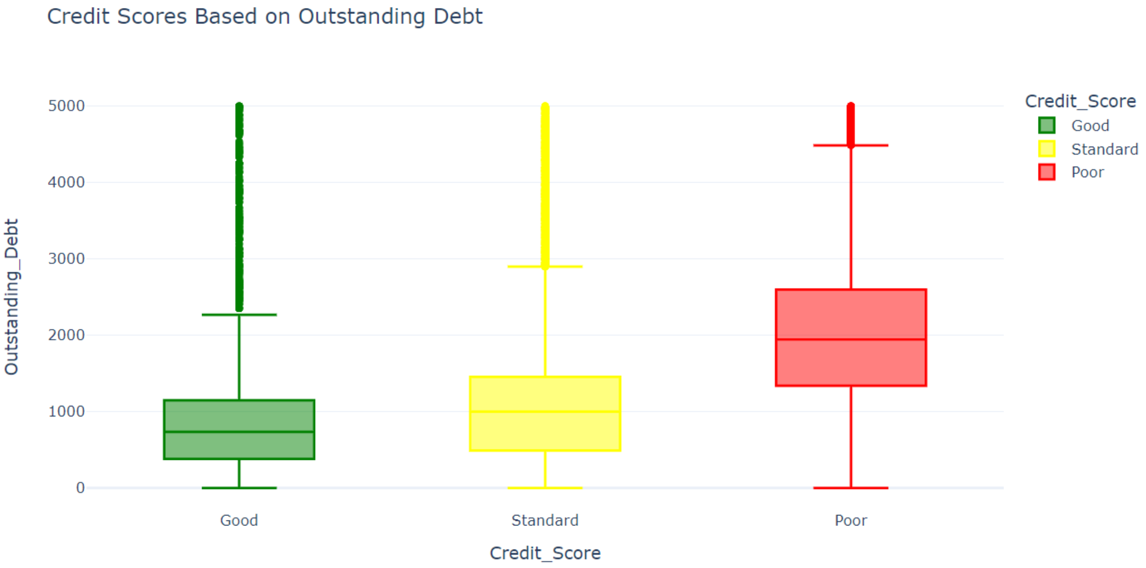
```
fig = px.box(data,
             x="Credit_Score",
             y="Delay_from_due_date",
             color="Credit_Score",
             title="Credit Scores Based on Average Number of Days Delayed for Credit card Payments",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```


Credit Scores Based on Average Number of Days Delayed for Credit card Payments

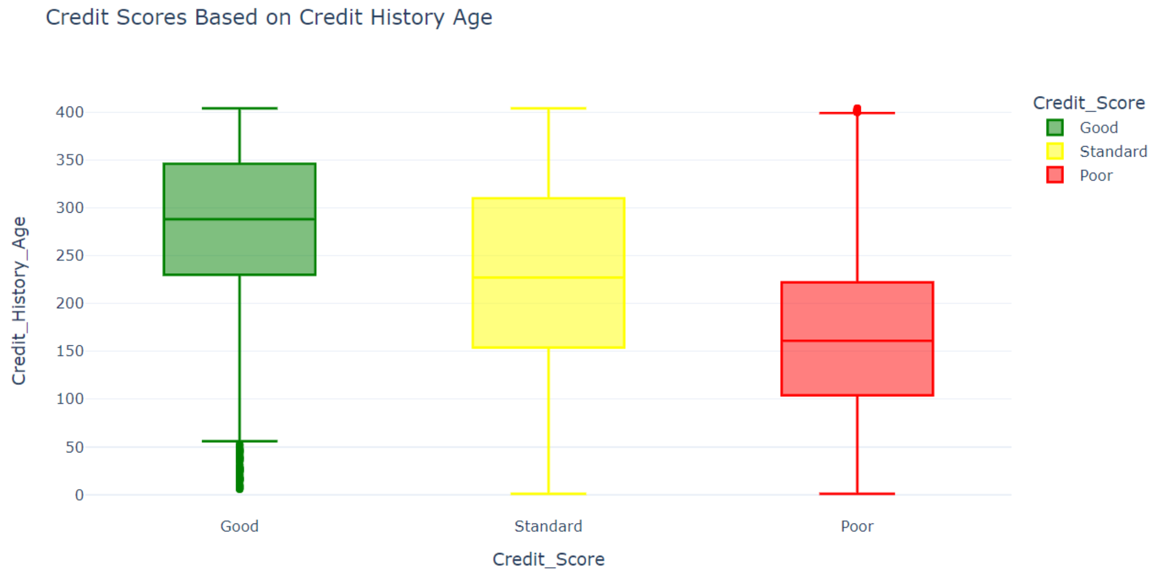
```
fig = px.box(data,
             x="Credit_Score",
             y="Num_of_Delayed_Payment",
             color="Credit_Score",
             title="Credit Scores Based on Number of Delayed Payments",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```


Credit Scores Based on Number of Delayed Payments

```
fig = px.box(data,
             x="Credit_Score",
             y="Outstanding_Debt",
             color="Credit_Score",
             title="Credit Scores Based on Outstanding Debt",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```

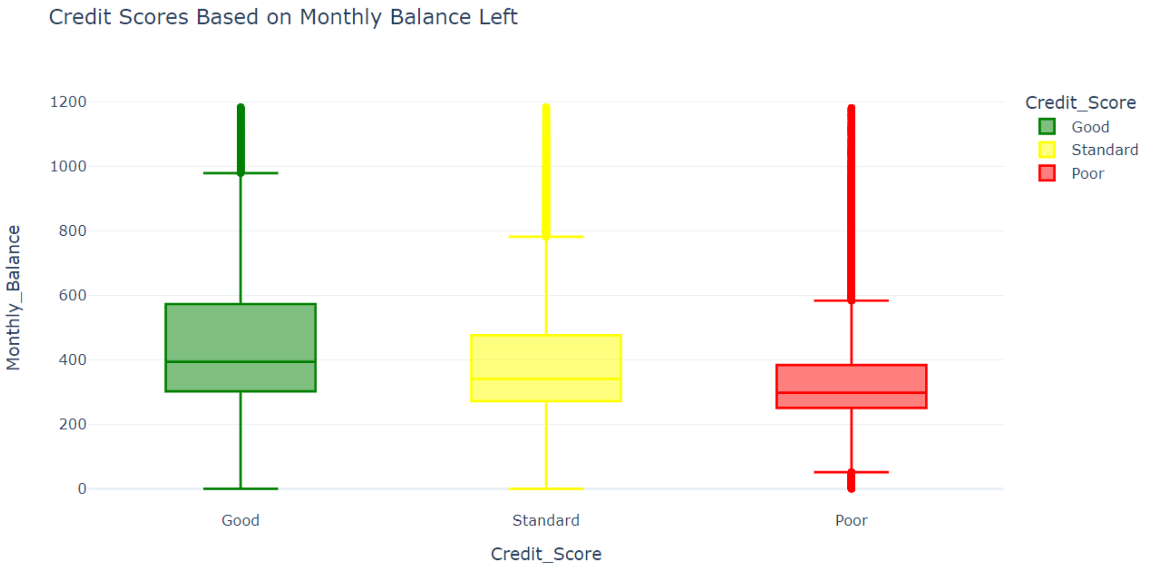


```
fig = px.box(data,
             x="Credit_Score",
             y="Credit_History_Age",
             color="Credit_Score",
             title="Credit Scores Based on Credit History Age",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



Credit Scores Based on Credit History Age

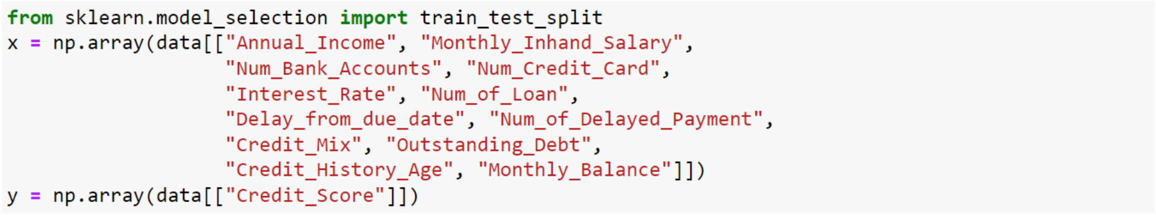
```
fig = px.box(data,
             x="Credit_Score",
             y="Monthly_Balance",
             color="Credit_Score",
             title="Credit Scores Based on Monthly Balance Left",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



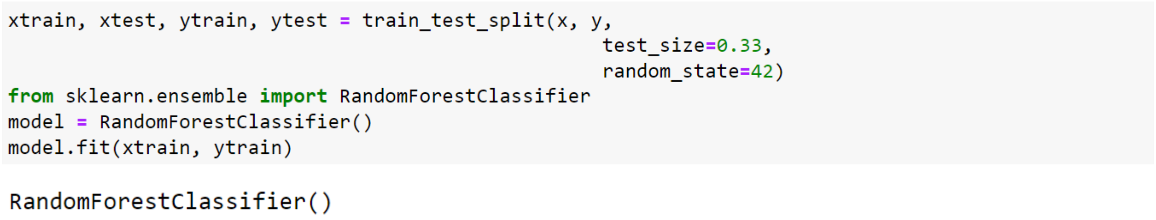
Credit Scores Based on Monthly Balance Left

```
data["Credit_Mix"] = data["Credit_Mix"].map({"Standard": 1,
                               "Good": 2,
                               "Bad": 0})
```

Model Building:



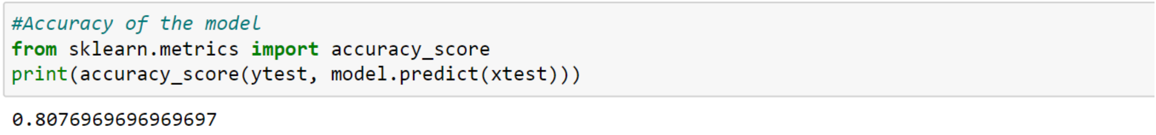
```
from sklearn.model_selection import train_test_split
x = np.array(data[["Annual_Income", "Monthly_Inhand_Salary",
                   "Num_Bank_Accounts", "Num_Credit_Card",
                   "Interest_Rate", "Num_of_Loan",
                   "Delay_from_due_date", "Num_of_Delayed_Payment",
                   "Credit_Mix", "Outstanding_Debt",
                   "Credit_History_Age", "Monthly_Balance"]])
y = np.array(data[["Credit_Score"]])
```



```
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                                                    test_size=0.33,
                                                    random_state=42)
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(xtrain, ytrain)
```

```
RandomForestClassifier()
```

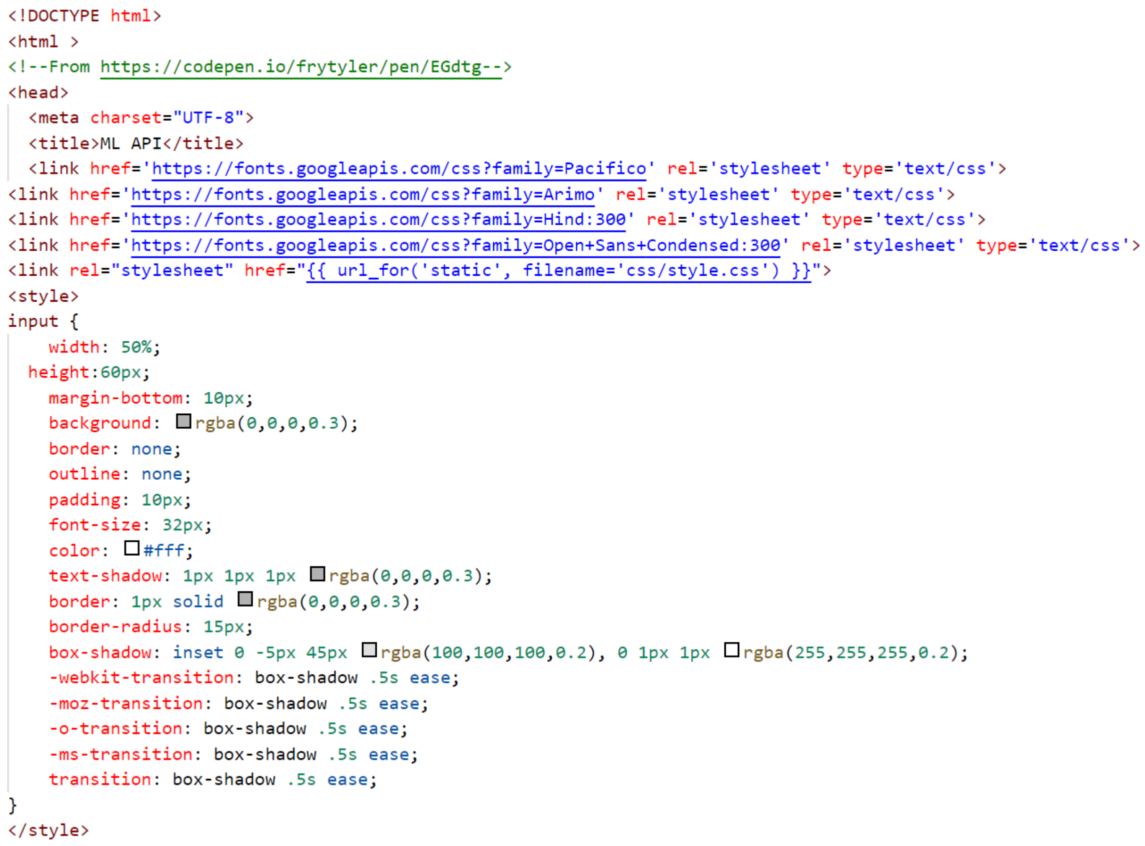
Accuracy of model:



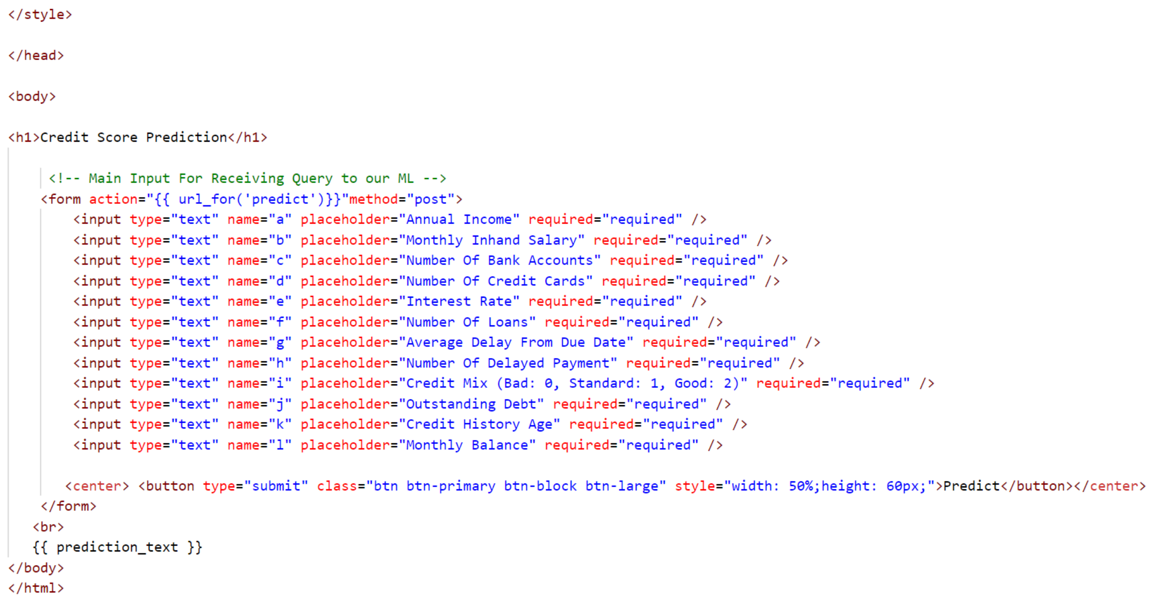
```
#Accuracy of the model
from sklearn.metrics import accuracy_score
print(accuracy_score(ytest, model.predict(xtest)))
```

```
0.8076969696969697
```

6.2 HTML Implementation:

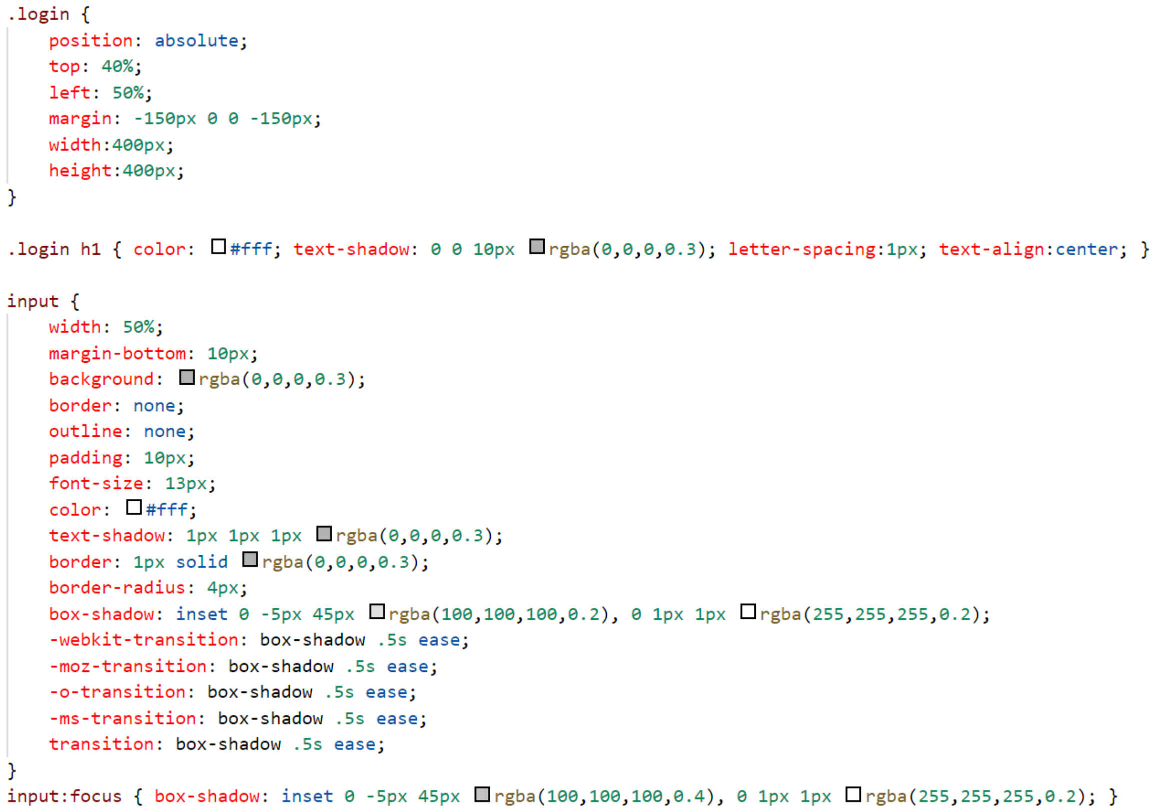


```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
<style>
input {
    width: 50%;
  height:60px;
    margin-bottom: 10px;
    background: rgba(0,0,0,0.3);
    border: none;
    outline: none;
    padding: 10px;
    font-size: 32px;
    color: #fff;
    text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
    border: 1px solid rgba(0,0,0,0.3);
    border-radius: 15px;
    box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
    -webkit-transition: box-shadow .5s ease;
    -moz-transition: box-shadow .5s ease;
    -o-transition: box-shadow .5s ease;
    -ms-transition: box-shadow .5s ease;
    transition: box-shadow .5s ease;
}
</style>
```



```
</style>

</head>

<body>

<h1>Credit Score Prediction</h1>

    <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
        <input type="text" name="a" placeholder="Annual Income" required="required" />
        <input type="text" name="b" placeholder="Monthly Inhand Salary" required="required" />
        <input type="text" name="c" placeholder="Number Of Bank Accounts" required="required" />
        <input type="text" name="d" placeholder="Number Of Credit Cards" required="required" />
        <input type="text" name="e" placeholder="Interest Rate" required="required" />
        <input type="text" name="f" placeholder="Number Of Loans" required="required" />
        <input type="text" name="g" placeholder="Average Delay From Due Date" required="required" />
        <input type="text" name="h" placeholder="Number Of Delayed Payment" required="required" />
        <input type="text" name="i" placeholder="Credit Mix (Bad: 0, Standard: 1, Good: 2)" required="required" />
        <input type="text" name="j" placeholder="Outstanding Debt" required="required" />
        <input type="text" name="k" placeholder="Credit History Age" required="required" />
        <input type="text" name="l" placeholder="Monthly Balance" required="required" />

       <center> <button type="submit" class="btn btn-primary btn-block btn-large" style="width: 50%;height: 60px;">Predict</button></center>
    </form>
   <br>
   {{ prediction_text }}
</body>
</html>
```

6.3 CSS Implementation:



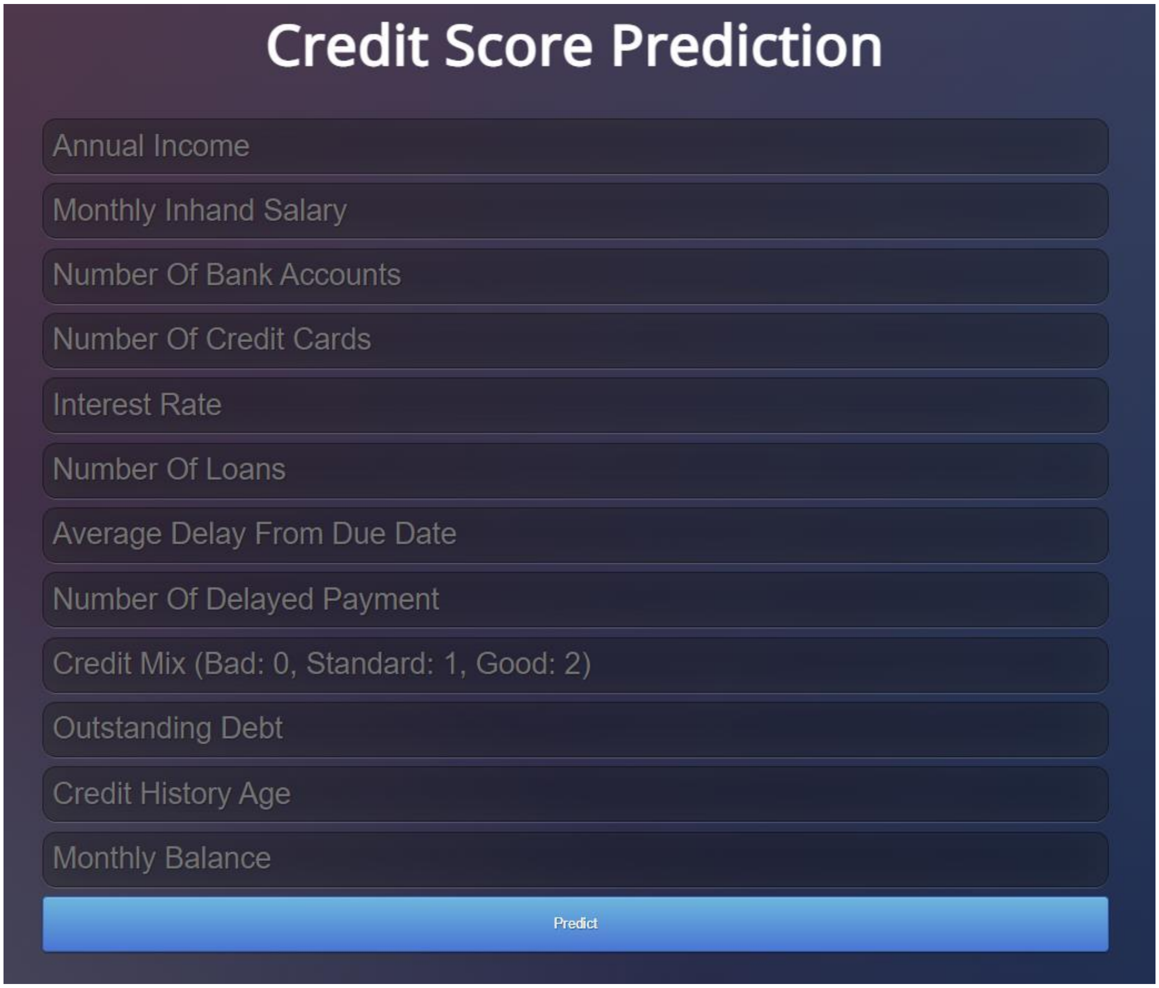
```
.login {
    position: absolute;
    top: 40%;
    left: 50%;
    margin: -150px 0 0 -150px;
    width:400px;
    height:400px;
}

.login h1 { color: □#fff; text-shadow: 0 0 10px ■rgba(0,0,0,0.3); letter-spacing:1px; text-align:center; }

input {
    width: 50%;
    margin-bottom: 10px;
    background: ■rgba(0,0,0,0.3);
    border: none;
    outline: none;
    padding: 10px;
    font-size: 13px;
    color: □#fff;
    text-shadow: 1px 1px 1px ■rgba(0,0,0,0.3);
    border: 1px solid ■rgba(0,0,0,0.3);
    border-radius: 4px;
    box-shadow: inset 0 -5px 45px □rgba(100,100,100,0.2), 0 1px 1px □rgba(255,255,255,0.2);
    -webkit-transition: box-shadow .5s ease;
    -moz-transition: box-shadow .5s ease;
    -o-transition: box-shadow .5s ease;
    -ms-transition: box-shadow .5s ease;
    transition: box-shadow .5s ease;
}
input:focus { box-shadow: inset 0 -5px 45px ■rgba(100,100,100,0.4), 0 1px 1px □rgba(255,255,255,0.2); }
```

7. Test Cases

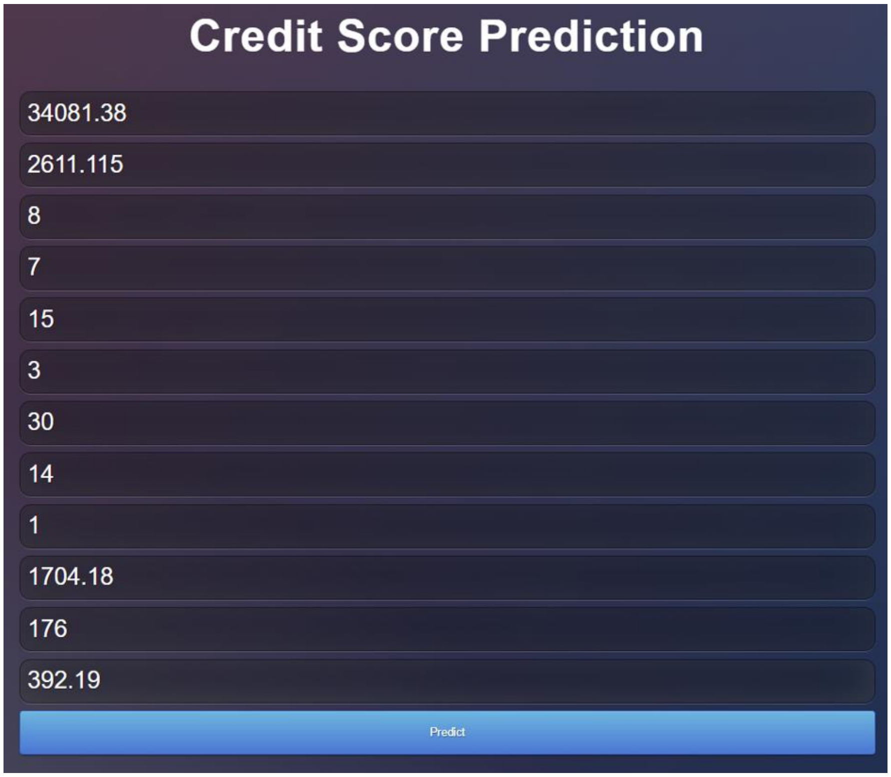
Test Trail ID	Test Scenario	Expected Result	Result
Trail 1	The inputs for prediction are purposely given to get poor credit score	Poor	Pass
Trail 2	The inputs for prediction are purposely given to get good credit score	Good	Pass
Trail 3	The inputs for prediction are purposely given to get standard credit score	Standard	Pass

8. Screenshots

Web UI:



Trail 1:



Credit Score Prediction

34081.38

2611.115

8

7

15

3

30

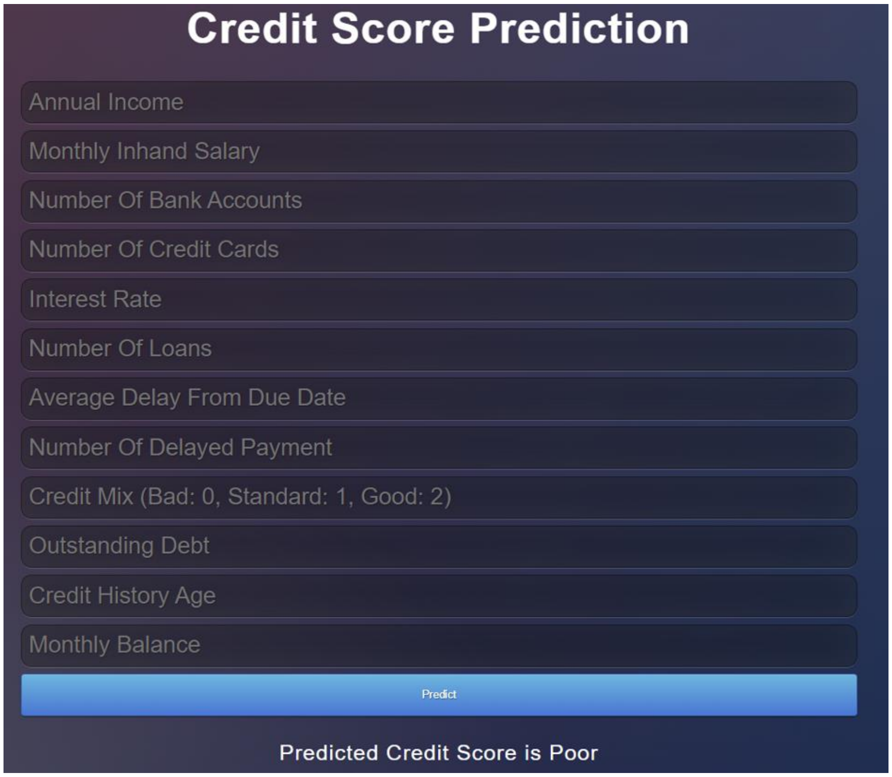
14

1

1704.18

176

392.19



Credit Score Prediction

Annual Income

Monthly Inhand Salary

Number Of Bank Accounts

Number Of Credit Cards

Interest Rate

Number Of Loans

Average Delay From Due Date

Number Of Delayed Payment

Credit Mix (Bad: 0, Standard: 1, Good: 2)

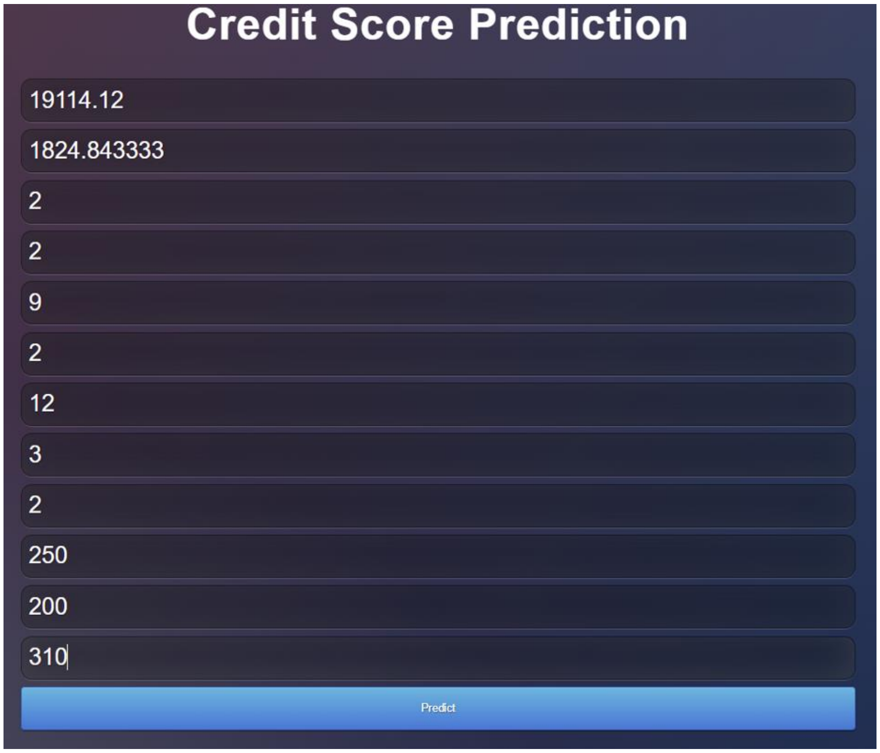
Outstanding Debt

Credit History Age

Monthly Balance

Predicted Credit Score is Poor

Trail 2:



Credit Score Prediction

19114.12

1824.843333

2

2

9

2

12

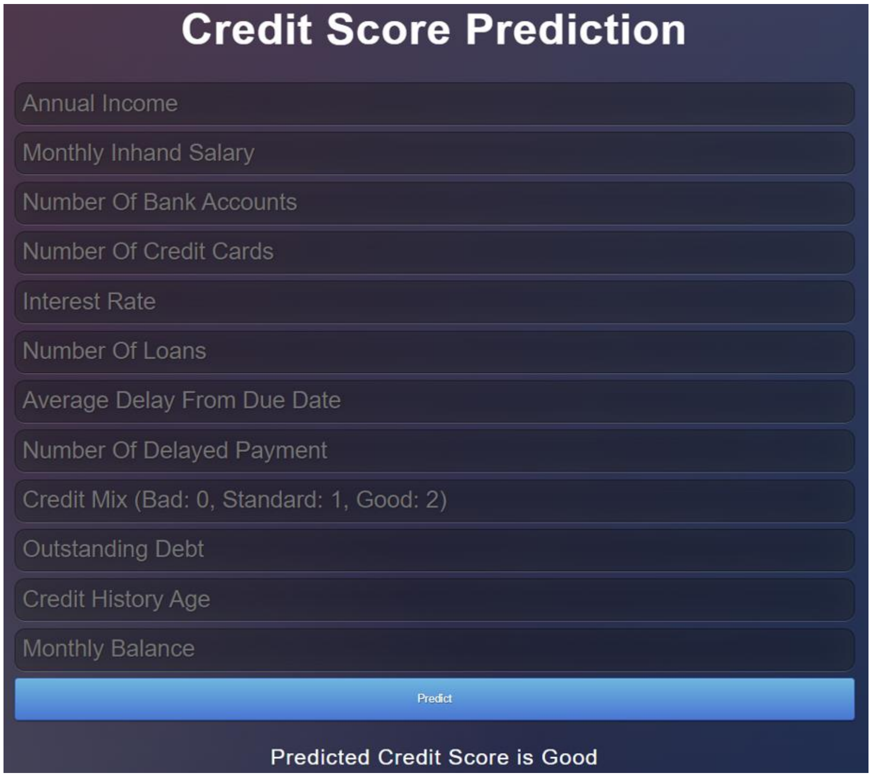
3

2

250

200

310



Credit Score Prediction

Annual Income

Monthly Inhand Salary

Number Of Bank Accounts

Number Of Credit Cards

Interest Rate

Number Of Loans

Average Delay From Due Date

Number Of Delayed Payment

Credit Mix (Bad: 0, Standard: 1, Good: 2)

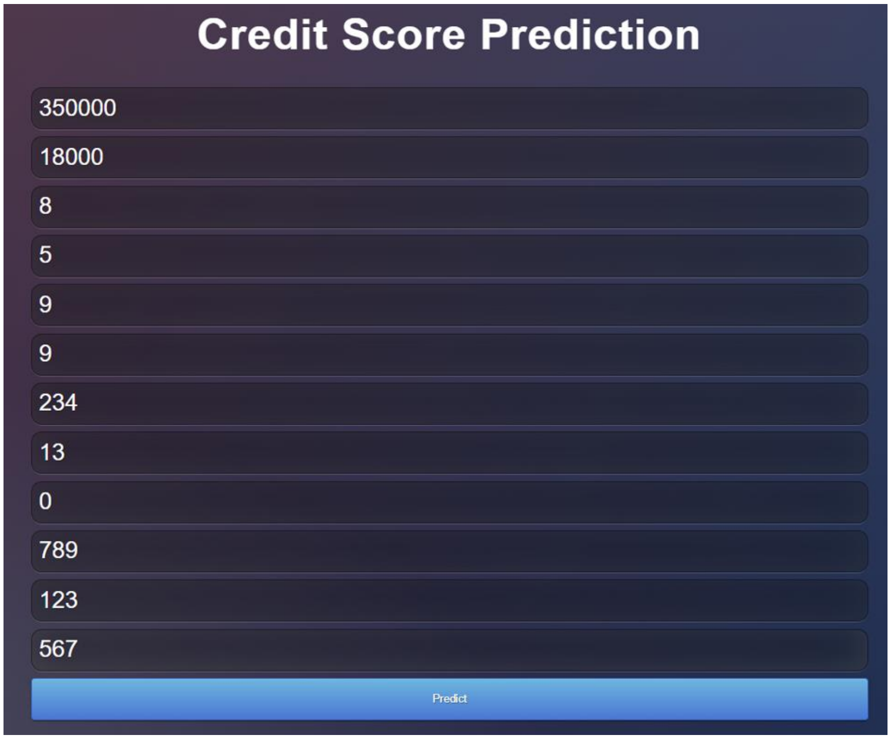
Outstanding Debt

Credit History Age

Monthly Balance

Predicted Credit Score is Good

Trail 3:



Credit Score Prediction

350000

18000

8

5

9

9

234

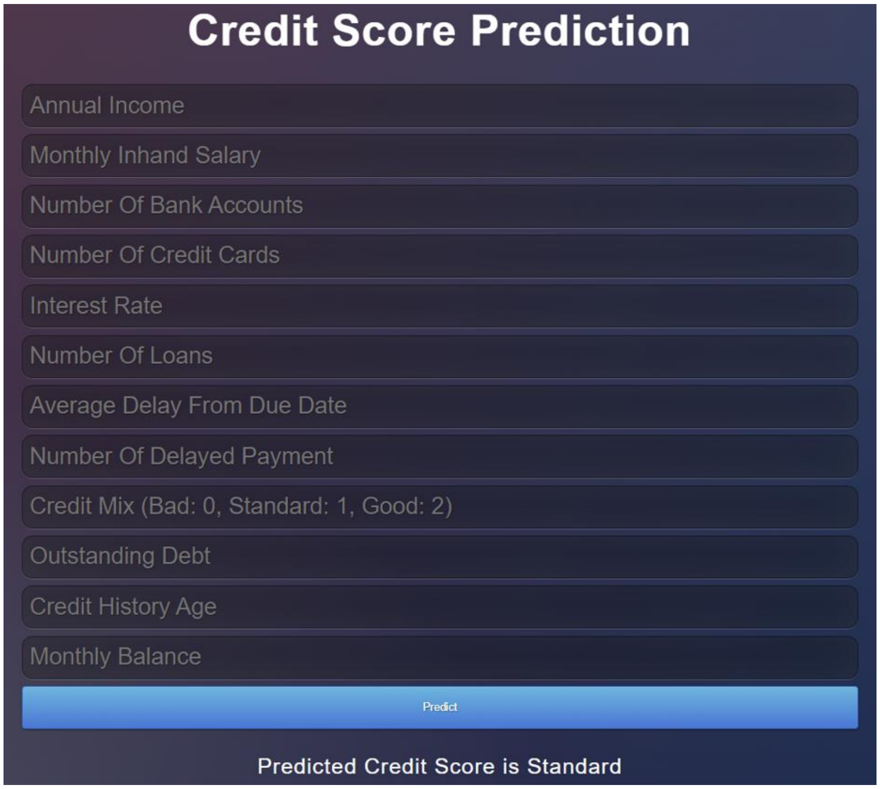
13

0

789

123

567



Credit Score Prediction

Annual Income

Monthly Inhand Salary

Number Of Bank Accounts

Number Of Credit Cards

Interest Rate

Number Of Loans

Average Delay From Due Date

Number Of Delayed Payment

Credit Mix (Bad: 0, Standard: 1, Good: 2)

Outstanding Debt

Credit History Age

Monthly Balance

Predicted Credit Score is Standard

9. Conclusion

The conclusion of this project is credit score classification can be an effective way for banks and credit card companies to make lending decisions. By analysing historical data and identifying key features that impact creditworthiness, the model can make predictions about a customer's likelihood of repayment. Good credit scores increase the chances of getting loans from various financial institutions. This information can be used to determine whether to issue a loan or not to that customer.

Overall, this project demonstrates the potential benefits of using machine learning algorithms for financial decision-making. Classifying customers based on their credit scores is crucial for banks to assess creditworthiness and issue loans accordingly. The model developed in this project uses several features to make predictions and can be used as a tool to assess creditworthiness of customers.

10. Future Scope

The further scope of this project could involve expanding the dataset used for training the model to improve its accuracy in predicting credit scores. This could be useful for individuals seeking to improve their creditworthiness or for financial institutions looking to automate the loan application process. Another potential direction for the project could be exploring other machine learning algorithms and comparing their performance with the random forest model.

The accuracy of the model can be further improved by incorporating more relevant features and using more advanced machine learning algorithms. Moreover, the model can be integrated with other financial systems to automate the loan approval process, thereby reducing the time and effort required to evaluate loan applications. This can help financial institutions to make more informed and objective decisions, while also ensuring a faster and more efficient loan approval process for their customers. Overall, this project has the potential to revolutionize the credit scoring system and streamline the lending process for both borrowers and lenders.

11. Bibliography

[1.] Tripathi, D., Edla, D. R., Cheruku, R., & Kuppili, V. (2019). A novel hybrid credit scoring model based on ensemble feature selection and multilayer ensemble classification Computational Intelligence, 35(2), 371-394.

[2.] Svozil, D., Kvasnicka, V., &Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks Chemometrics and intelligent laboratory systems, 39(1), 43-62.

[3.] C. Serrano-Cinca and B. Guti´errez-Nieto, The use of profit scoring as an alternative to credit scoring systems in peer-topeer (p2p) lending, Decision Support Systems, vol. 89, pp. 113–122, 2016.

[4]. Galindo J, Tamayo P 2000 Credit risk assessment using statistical and machine learning: Basic methodology and risk modeling applications

[5]. Hand D, Henley W 1997 Statistical classification methods in consumer credit scoring: A review Journal of the Royal Statistical Society