CREDIT SCORING SYSTEM

USING XGBOOST AND LIGHTBGM

A DESIGN PROJECT REPORT

Submitted by

MONISH VIDYARTHI. R

MADHUMITHA .P

KANNIGA SARASWATHY .M

SUBALATHA .A

in partial fulfilment for the award of the

degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENGE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, NewDelhi)

SAMAYAPURAM–621112

NOVEMBER 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)
SAMAYAPURAM–621112

BONAFIDE CERTIFICATE

Certified that this design project report titled **"CREDIT SCORING SYSTEM USING XGBOOST AND LIGHTBGM"** is the bonafide work of **KANNIGA SARASWATHY.M (REG NO: 811721243024), MADHUMITHA.P (REG NO: 811721243026), MONISH VIDYARTHI.R (REG NO: 811721243035), SUBALATHA.A (REG NO: 811721243054)** who carried out the project under my supervision.

SIGNATURE

Dr.T.Avudaiappan M.E., Ph.D.,
HEAD OF THE DEPARTMENT,
Associate Professor, Department of AI,
K.Ramakrishnan College of Technology (Autonomous),
Samayapuram – 621112.

SIGNATURE

Mr.P.B.Aravind Prasad M.Tech.,
SUPERVISOR
Assistant Professor, Department of AI,
K.Ramakrishnan College of Technology (Autonomous),
Samayapuram – 621112.

Submitted for the viva-voce examination held on ……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"CREDIT SCORING SYTEM USING XGBOOST AND LIGHTBGM "** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This design project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF TECHNOLOGY**.

SIGNATURE

_____

MONISH VIDYARTHI R

_____

MADHUMITHA P

_____

KANNIGA SARASWATHY M

_____

SUBALATHA A

PLACE : SAMAYAPURAM

DATE :

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in - debt to our institution "**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)**", for providing us with the opportunity to do this project.

We are glad to credit honorable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA., Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

We would like to thank our principal **Dr. N. VASUDEVAN, M.E., Ph.D.,** who gave opportunity to frame the project the full satisfaction.

We whole heartily thanks to **Dr. T. AVUDAIAPPAN**, **M.E., Ph.D.,** HEAD OF THE DEPARTMENT, **ARTIFICAL INTELLIGENCE** for providing his encourage pursuing this project.

I express my deep and sincere gratitude to my project guide **Mr P.B.ARAVIND PRASAD M.Tech.,**ASSISTANT PROFESSOR, **ARTIFICIAL INTELLIGENCE** for her incalculable suggestions, creativity, assistance and patience which motivated me to carry out the project successfully.

I render my sincere thanks to my project coordinator **Mrs. P.B. ARAVIND PRASAD M.Tech.,**,other faculties and non-teaching staff members for providing valuable information during the course. I wish to express my special thanks to the officials & Lab Technicians of our departments who rendered their help during the period of the work progres

ABSTRACT

Credit scoring systems play an essential role in financial institutions by assessing the creditworthiness of individuals and businesses, directly influencing lending decisions, risk management, and financial stability. This project investigates the implementation of a credit scoring system leveraging advanced machine learning models, specifically LightGBM and XGBoost, to improve the accuracy, scalability, and reliability of credit risk evaluations. Both models have proven capabilities in handling high-dimensional data and capturing complex non-linear relationships, making them well-suited for this application.The proposed system begins with comprehensive data preprocessing, involving handling missing values, addressing data imbalance, and performing feature engineering to extract relevant financial and behavioral indicators such as credit history, income levels, outstanding debt, and repayment behavior. These features are then utilized to train the models, with LightGBM excelling in its efficiency and scalability, particularly for large datasets, and XGBoost offering robust performance through gradient boosting techniques and effective handling of overfitting via regularization Metrics such as accuracy, precision, recall, and the area under the ROC curve (AUC-ROC) are used to evaluate the models' effectiveness. In addition to model training, the project integrates explainability techniques such as SHAP (SHapley Additive exPlanations) values to provide insights into the contribution of each feature to the model’s predictions. .The credit scoring system is designed to support real-time decision-making, making it highly applicable for integration into online financial services and credit monitoring platforms. It is also adaptable to various financial environments, catering to the needs of different economies and demographic groups.Despite its strengths, the implementation of this system presents challenges.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

XGBoost	Extreme Gradient Boosting
LightGBM	Light Gradient Boosting Machine
ROC	Receiver Operating Characteristic
AUC	Area Under Curve
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
TPR	True Positive Rate
FPR	False Positve Rate

CHAPTER 1

INTRODUCTION

A credit scoring system is a critical component in modern financial services, using machine learning to evaluate creditworthiness by analyzing data about an individual's financial behavior. The implementation of XGBoost and LightGBM for credit scoring has brought precision and efficiency to the process, leveraging advanced features for robust predictions. By integrating these algorithms into a web-based platform, credit scoring systems can provide seamless, real-time evaluations, making them indispensable in lending, insurance, and risk assessment. This study examines the role of XGBoost and LightGBM in building accurate, scalable credit scoring models, highlighting their importance in improving financial decision-making.

1.1 BACKGROUND

Credit scoring has evolved from traditional rule-based systems to sophisticated data-driven approaches. Initially relying on linear regression models, the process has expanded to include advanced machine learning techniques. XGBoost and LightGBM have emerged as leading algorithms in credit scoring due to their ability to handle large datasets, account for complex feature interactions, and deliver high predictive accuracy. These gradient-boosting algorithms excel in processing imbalanced data, a common challenge in credit scoring, and ensure faster computation with scalable solutions.The integration of XGBoost and LightGBM into web-based platforms has further transformed the landscape, enabling real-time credit evaluations accessible to users globally. These platforms combine feature engineering, model training, and visualization to deliver an intuitive and transparent user experience. Such advancements make credit scoring systems a cornerstone of financial innovation, driving inclusivity

1.2 PROBLEM STATEMENT

The challenge with existing credit scoring systems lies in their inability to address several critical issues effectively. Feature selection and engineering often fail to capture the full complexity of financial behavior, leading to models that are either overfitted or biased. Additionally, the inherent imbalance in credit scoring datasets, where the majority of data represents low-risk borrowers, makes it difficult for models to accurately predict outcomes for minority high-risk groups. Many machine learning models, particularly advanced ones like XGBoost and LightGBM, are often criticized for their lack of interpretability. Furthermore, the growing demand for real-time credit evaluations poses challenges in optimizing computational pipelines without compromising accuracy. Finally, the dynamic nature of financial markets and regulatory requirements necessitates systems that are adaptable and can evolve to meet changing needs. Overcoming these challenges is essential to create a robust and reliable credit scoring system.

1.3 AIM AND OBJECTIVES

1.3.1 AIM

To develop an advanced web-based credit scoring system leveraging XGBoost and LightGBM, providing accurate and interpretable credit risk assessments using optimized feature selection and real-time analytics.

1.3.2 OBJECTIVES

1.Implement Advanced Feature Engineering: Utilize domain-specific knowledge and automated methods to create a feature set that improves model performance and robustness.

2.Optimize for Imbalanced Data**:** Apply techniques such as Synthetic Minority Oversampling Technique (SMOTE) or class-weight adjustments to address imbalances in the dataset.

CHAPTER 2
LITERATURE REVIEW

2.1 CREDIT SCORING PREDICTION SYSTEM USING DEEP LEARNING AND K-MEANS ALGORITHMS

Author Name Ashwani Kumar, D. Shanthi Pronaya Bhattacharya

Year of Publication: 2022

Abstract

In financial markets, credit rating and risk assessment tools are used to minimize potential risk up to some extent for credit score. Nowadays, the banking and financial industry has experienced rapid expansion. Therefore, with this growth, the numbers of credit card applications with various credit products are increasing day by day because many people want to avail these services for their personal interest. The challenge here is to identify insights on the performance of a finance industry by using deep learning algorithms as they directly affect the viability of that industry. The scheme contains a predictive model which uses feature selection(FS) classification and deep learning applications simultaneously to train the proposed model to perform effectively.

Advantages:

Risk Mitigation, Profitability, Eficiency, Accuracy and Data insights.

Disadvantages:

Data dependence, Bias Risk, Complexity, Model Overfitting, Limited scope.

2.2 CREDIT SCORING USING MACHINE LEARNING AND DEEP LEARNING-BASED MODELS

Author Name **S**ami Mestiri

Year of Publication: 2024

Abstract

Credit scoring is a useful tool for assessing the capability of customers repayments. The purpose of this paper is to compare the predictive abilities of six credit scoring models: Linear Discriminant Analysis (LDA), Random Forests (RF), Logistic Regression (LR), Decision Trees(DT), Support Vector Machines (SVM) and Deep Neural Network (DNN). To compare these models, an empirical study was conducted using a sample of 688 observations and twelve variables. The performance of this model was analyzed using three measures: Accuracy rate, F1 score, and Area Under Curve (AUC). In summary, machine learning techniques exhibited greater accuracy in predicting loan defaults compared to other traditional statistical models.

Advantages:

Simple. interpretable, High accuracy, Easy interpretation, Easy visualization

Disadvantages:

Limited flexibility, Computational expensive, Limited non-linearity, overfitting prone

2.3 A RECENT REVIEW ON OPTIMIZATION METHODS APPLIED TO CREDIT SCORING METHODS

Author Name **E**lias Shohel Kamimura, Anderson Rogerio faia pinto and Marcelo Seido Nagano

Year of Publication: 2023

Abstract

The research methodology employed technical procedures based on

bibliographic and exploratory analyses. A traditional investigation was carried out using the Scopus, ScienceDirect and Web of Science databases. The findings showed that CSMs are usually formulated using Financial Analysis, Machine Learning, Statistical Techniques, Operational Research and Data Mining Algorithms. The main databases used by the researchers were banks and the University of California, Irvine. The analyses identified 48 methods used by CSMs, the main ones being: Logistic Regression (13%), Naive Bayes (10%) and Artificial Neural Networks (7%).. As it was aimed to demonstrate the application of optimisation methods, it is highlyconsiderable that legal and ethical issues should be better adapted to CSMs.

Advantages:

Comprehensive review, Risk management, predictive power and customer evaluation

Disadvantages:

Data dependence, Limited applicability, uncertainity and implementation costs.

2.4 ANALYZING MACHINE LEARNING METHODS FOR CREDIT SCORING WITH EXPLAINABLE AI AND OPTIMIZING INVESTMENT DECISIONS

Author Name Swathi Tyagi

Year of Publication: 2023

Abstract

This paper examines two different yet related questions related to explainable AI (XAI) practices. Machine learning (ML) is increasingly important in financial services, such as pre-approval, credit underwriting, investments, and various front-end and back-end activities. Machine Learning can automatically detect non-linearities and interactions in training data, facilitating faster and more accurate credit decisions. However, machine learning models are opaque and hard to explain, which are critical elements needed for establishing a reliable technology. The study compares various machine learning models, including single classifiers (logistic regression, decision trees, LDA, QDA), heterogeneousensembles (AdaBoost, Random Forest), and sequential neural networks. The results indicate that ensemble classifiers and neural networks outperform.

Advantages:

Accuracy, Faster decisions and Non-linear detection.

Disadvantages:

Data dependence, Model opacity, complex interpretation and Overfitting risk

2.5 CREDIT SCORING METHODS USING DEEP LEARNING MODULES

Author Name Anton Markov, Zinaida Seleznyova and Victor Lapshin

Year of Publication: 2022

Abstract

Credit risk is the most significant risk by impact for any bank and financial institution. Accurate credit risk assessment affects an organisation's balance sheet and income statement, since credit risk strategy determines pricing, and might even influence seemingly unrelated domains, e.g. marketing, and decision-making. Credit risk assessment is a sensitive subject for any bank and financial institution for several reasons. Firstly, credit risk is subject to external evaluation, since central banks and auditors rigorously monitor how financial institutions comply with Basel and International Financial Reporting Standards (IFRS) requirements. Secondly, precise credit risk estimation is key to an organisation's profitability. If the bank fails to estimate a risk correctly, it either overprices loans and loses its market share, or sets interest rates too low to cover the expected losses, which leads to poor financial results. Finally, since credit risk is a vital part of the net present value (NPV) of financial instruments.

Advantages:

Accuracy, Superior performance, handles complexity and robust testing.

Disadvantages:

Overfitting, Less effective, resource-intensive and interpretability

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing credit scoring systems often rely on traditional machine learning algorithms or simpler statistical models. While these approaches can provide a baseline assessment of credit risk, they suffer from several limitations. Firstly, these systems may lack the ability to capture complex patterns and relationships within the data, which are crucial for accurate credit scoring. Secondly, the scalability of these systems is often a challenge when dealing with large datasets or high-dimensional data. Thirdly, traditional systems may struggle with interpretability, making it difficult to explain decisions to stakeholders or meet regulatory compliance requirements. Finally, they often depend heavily on manual feature engineering, which is time-consuming and prone to errors.

3.1.1 Drawbacks

- Limited Predictive Power: Traditional algorithms may not capture non-linear relationships effectively.
- Scalability Issues: Struggles with large or high-dimensional datasets
- Interpretability Challenges: Difficult to explain decisions to stakeholders.
- Manual Feature Engineering: Requires significant time and expertise.

3.2 PROPOSED SYSTEM

The proposed credit scoring system leverages advanced machine learning models, specifically XGBoost and LightGBM, to overcome the limitations of traditional systems. XGBoost, known for its high performance and ability to handle complex data structures, offers improved predictive power and efficiency. Similarly, LightGBM provides exceptional scalability and speed, making it suitable for large-scale credit scoring tasks. Both models are capable of automated feature selection and importance ranking, reducing the dependency on manual feature engineering. Moreover, these systems include built-in mechanisms for handling missing data and imbalanced datasets, which are common in credit scoring applications. Their interpretability features, such as SHAP (SHapley Additive exPlanations), allow transparent decision-making and regulatory compliance.

3.2.1 Advantages

- High Predictive Accuracy: Models capture complex patterns and relationships.
- Scalability: Efficient for large datasets with LightGBM's tree-based learning.
- Speed: Fast training and prediction processes.
- Robustness: Handles missing data and imbalances effectively.
- Feature Importance: Automated ranking for better insights.
- Interpretability: Use of SHAP values for transparent decisions.
- Customization: Easily tuned to adapt to specific credit scoring requirements.

CHAPTER 4

SYSTEM SPECIFICATIONS

4.1 HARDWARE REQUIREMENTS

- Computer : Intel Core i7/i9 (10th Gen or newer) or AMD Ryzen 7/9.
- Memory : 16-32 GB for handling large datasets.
- Storage : At least 512 GB, though 1 TB is recommended if the dataset is large.
- GPU(Optional) : NVIDIA GeForce RTX 2060 or higher with at least 6 GB VRAM.

4.2 SYSTEM REQUIREMENTS

- Python programming language : Python 3.x installed on the server.
- Operating system : Windows, Linux or MacOS.
- Python libraries : numpy, pandas, scikit-learn, matplotlib, xgboost and lightbgm.
- Version control : Git for version control or track changes.

4.3 SOFTWARE DESCIPTION

The proposed credit scoring system utilizes advanced machine learning algorithms, specifically LightGBM and XGBoost, to enhance the process of assessing creditworthiness. The software integrates several key modules, including data preprocessing, model training, prediction, and explainability. The preprocessing module handles tasks like missing value imputation, outlier detection, and feature engineering to prepare raw financial

data for analysis. The training module leverages LightGBM for its efficiency in handling large datasets and fast computation, and XGBoost for its gradient boosting capabilities and regularization techniques that prevent overfitting. The prediction module ensures accurate and real-time credit scoring by analyzing input features and delivering risk evaluations. To ensure transparency, the software incorporates explainability tools such as SHAP values, allowing users to interpret the contributions of individual factors to the overall credit score.

4.3.1 DEVELOPING ENVIRONMENT

To develop a credit scoring system using XGBoost and LightGBM, the software environment must include Python 3.x as the programming language installed on the server. The system should run on a supported operating system such as Windows, Linux (e.g., Ubuntu or CentOS), or macOS (Monterey or newer). Core Python libraries like numpy for numerical computations, pandas for data manipulation, and scikit-learn for preprocessing and evaluation are essential, alongside xgboost and lightgbm for implementing the machine learning models. Visualization libraries such as matplotlib and seaborn can be used for data analysis and insights. For model explainability, libraries like shap are recommended, while tools like optuna assist with hyperparameter optimization. Version control is managed using Git, with platforms like GitHub or GitLab for repository hosting and collaboration. Developers can use IDEs such as PyCharm, Jupyter Notebook, or Visual Studio Code for efficient coding and debugging.

4.3.2 LIBRARY

- **Pandas** for data manipulation and preprocessing, such as cleaning and organizing large datasets.
- **Numpy** used for numerical computations and array-based operations.
- **Scikit-Learn** provides tools for data preprocessing, model evaluation, and building machine learning pipelines.
- **Matplotlib** used for visualizing data distribution and understanding feature importance.
- **LightBGM** A gradient boosting framework optimized for speed and efficiency.
- **XGBoost** another gradient boosting algorithm designed for scalability and performance. Well-suited for structured and tabular data.

CHAPTER 5

ARCHITECTURE DESIGN

5.1 SYSTEM ARCHITECTURE

A credit scoring system using LightGBM and XGBoost typically follows a multi-layered architecture to ensure scalability, accuracy, and interpretability. At the data ingestion layer, raw data is collected from various sources, such as financial histories, demographic information, and transactional records. This data is then passed through a preprocessing layer where missing values are handled, features are encoded or normalized, and unnecessary variables are removed. The cleaned data enters the modeling layer, where both LightGBM and XGBoost models are trained in parallel. These models are combined or compared using techniques like model stacking or ensembling to deliver a final, optimized credit score. The system also includes a model evaluation layer that ensures the models generalize well to unseen data, using performance metrics like accuracy, precision, recall, F1-score, AUC-ROC, and log loss. Explainability and interpretability are key in credit scoring systems, so tools like SHAP or LIME are applied to provide insights into the factors driving the model's predictions. Once the models are trained and optimized, they are deployed in a production environment, typically as APIs or cloud-based solutions, to handle real-time credit score requests. The system is designed for scalability, ensuring that it can handle large volumes of data and requests concurrently. Continuous monitoring and periodic retraining are necessary to maintain the accuracy of predictions as new data becomes available.

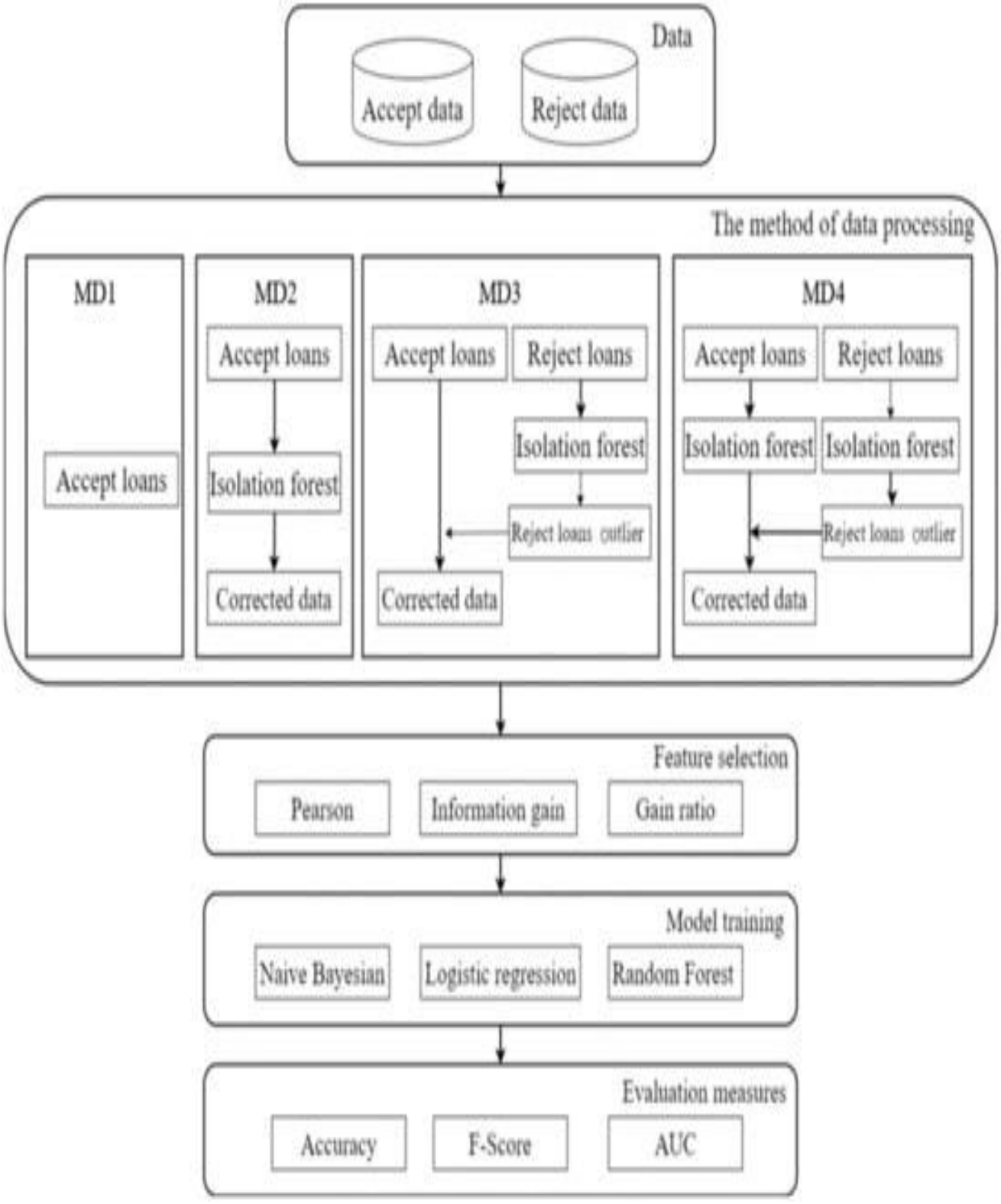

FIGURE NO 5.1 .SYSTEM ARCHITECTURE

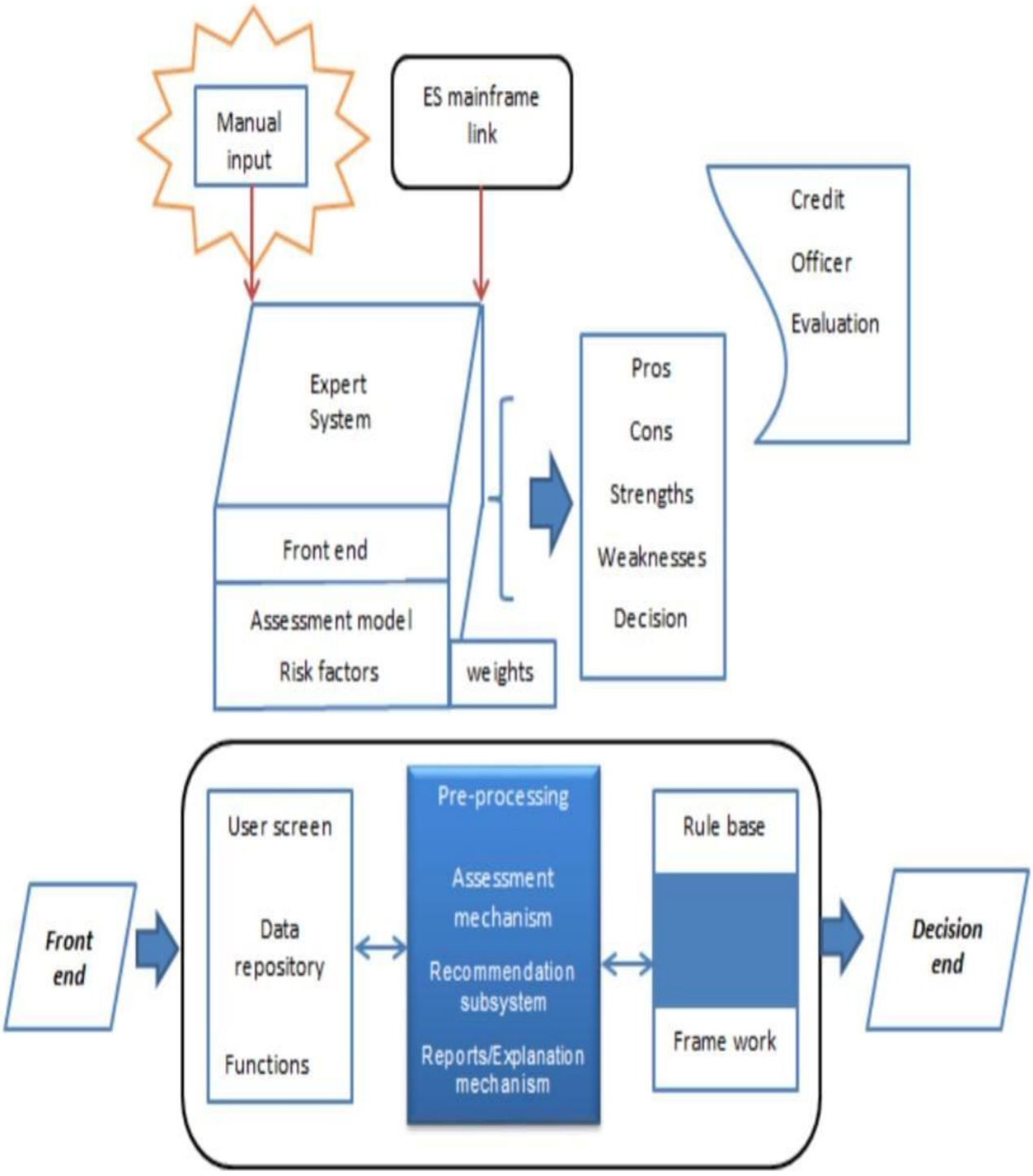

FIGURE NO 5.2. WEB INTERFACE

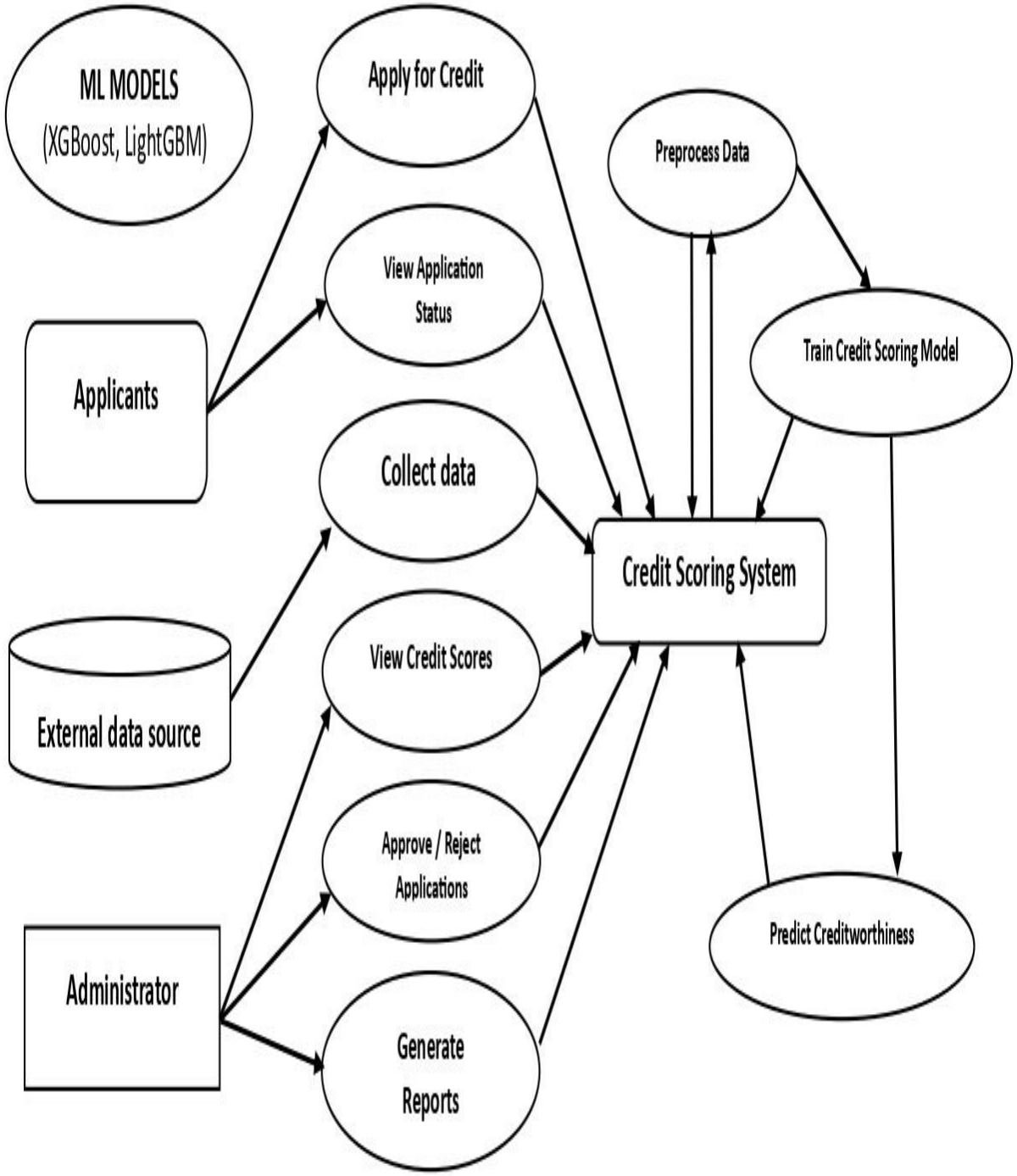

FIGURE NO 5.3 .USE CASE DIAGRAM

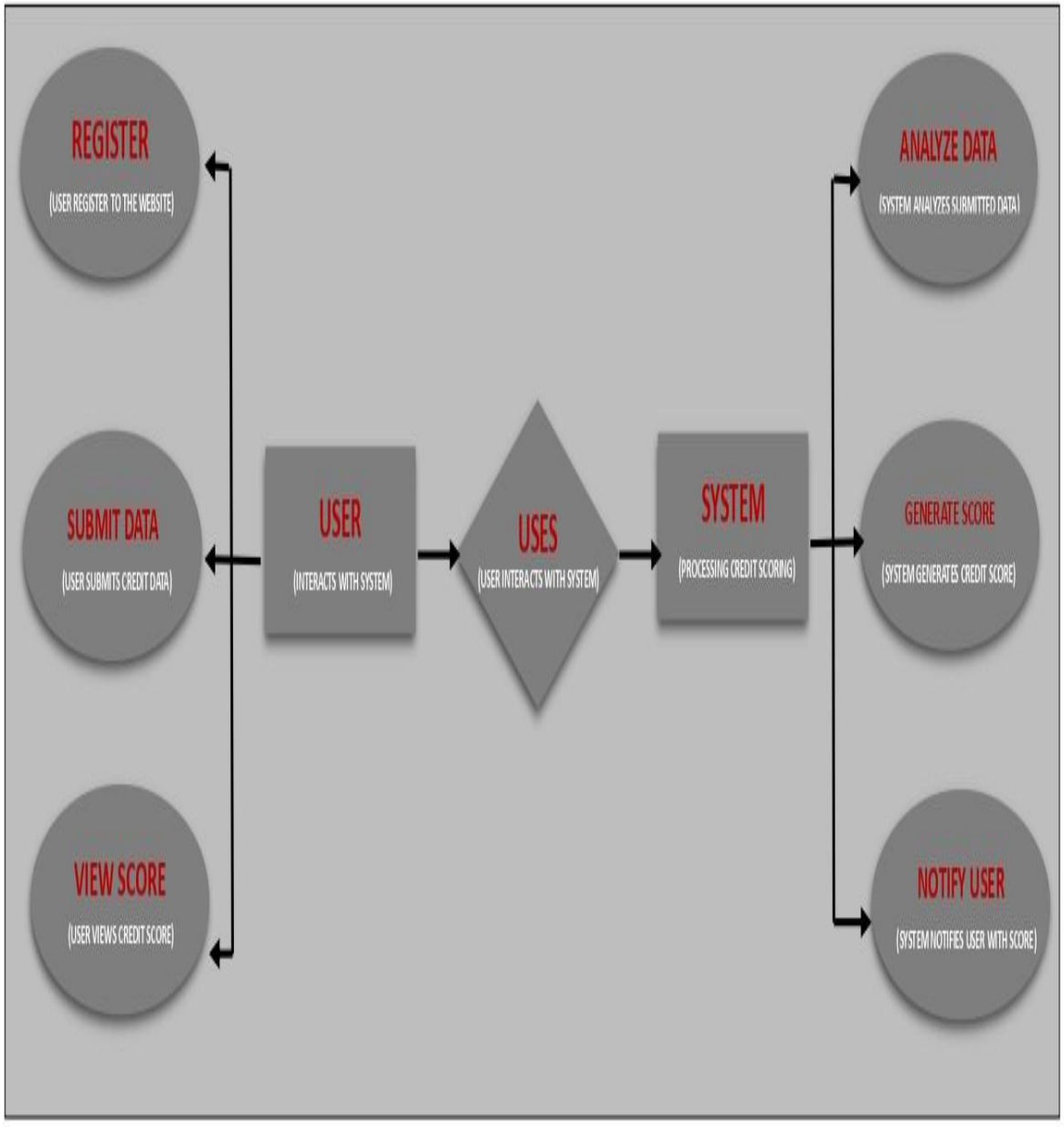

FIGURE NO 5.4. BLOCK DIAGRAM

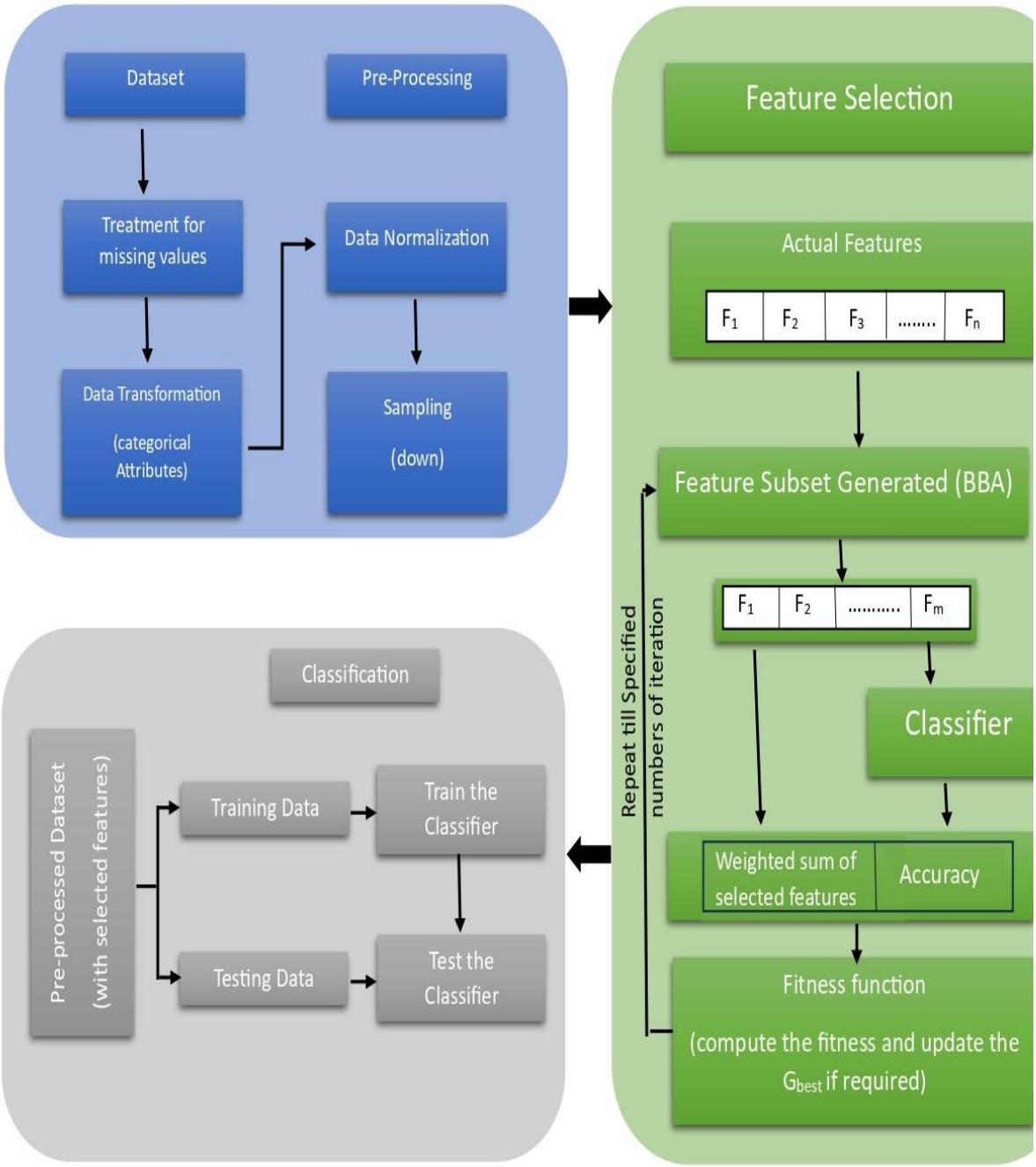

FIGURE NO 5.5. FEATURE ANALYSIS DIAGRAM

CHAPTER 6

MODULE DESCRIPTION

6.1 1. Data Processing Module

The system begins with a robust Data Processing Module that segregates data into two categories: Accepted Loans and Rejected Loans. Using advanced anomaly detection techniques such as the Isolation Forest algorithm, this module identifies outliers that could compromise the analysis. The corrected data is filtered into multiple stages (MD1 to MD4) to ensure accuracy and reliability. This module ensures that any noisy or inconsistent data is isolated and handled effectively, forming the foundation for further analysis.

1. Data Segregation: Divides the dataset into categories such as "Accepted Loans" and "Rejected Loans" for focused processing.
2. Outlier Detection: Uses the Isolation Forest algorithm to identify and eliminate anomalies that could skew analysis.
3. Data Correction: Cleanses and standardizes data to ensure consistency across all records.
4. Multi-Stage Filtering: Processes data through multiple layers (e.g., MD1 to MD4) to ensure accurate handling and reduce redundancy.
5. Data Readiness: Prepares a robust dataset free of noise and inaccuracies, enabling precise feature extraction and model training.

6.1.2. Feature Selection Module

To maximize the predictive power of the model, the Feature Selection Module identifies the most relevant attributes from the dataset. Techniques like Pearson Correlation, Information Gain, and Gain Ratio are employed to

evaluate the importance of features. This step not only reduces dimensionality but also ensures that only impactful variables, such as income level, debt-to-income ratio, credit history, and repayment behavior, are used for modeling. Feature selection ensures a streamlined and efficient learning process.

1. Relevance Scoring: Identifies and ranks features based on their correlation to creditworthiness (e.g., Pearson Correlation).

2. Dimensionality Reduction: Removes irrelevant or redundant attributes, simplifying the dataset for improved model performance.

3. Information Metrics: Uses methods like Information Gain and Gain Ratio to measure the predictive power of each feature.

4. Feature Prioritization: Selects variables such as credit history, income level, and repayment patterns for inclusion in the model.

5. Optimization: Enhances the efficiency of the machine learning model by reducing computational complexity and focusing on impactful features.

6. Iterative Learning: Continuously updates the model using feedback from past performance to improve prediction accuracy.

7. Algorithm Comparison: Evaluates the strengths of each model to identify the most suitable one for real-world deployment.

6.1.3. Model Training Module

The Model Training Module is the core of the credit scoring system, utilizing machine learning algorithms such as Naive Bayes, Logistic Regression, and

Random Forest. Each model is trained on the cleaned and feature-optimized dataset to predict an individual's likelihood of repaying loans. This module iteratively improves its accuracy using historical data and fine-tuning techniques, ensuring that the predictions are both precise and reliable.Algorithm Selection: Implements machine learning algorithms like XGboost and lightBGM ,Naive Bayes, Logistic Regression, and Random Forest for credit scoring.

1. Model Training: Trains each algorithm using the processed and feature-optimized dataset to predict loan repayment probability.

2. Hyper parameter Tuning: Adjusts algorithm parameters to achieve optimal performance and minimize prediction errors.

3. **Iterative Learning:** Continuously updates the model using feedback from past performance to improve prediction accuracy.

4. Algorithm Comparison: Evaluates the strengths of each model to identify the most suitable one for real-world deployment.

6.1.4. Evaluation and Validation Module

The Evaluation and Validation Module measures the performance of the trained models using metrics like Accuracy, F-Score, and Area Under the Curve (AUC). This module ensures that the system delivers high-quality predictions by comparing the outcomes of different models and selecting the most optimal one. Additionally, cross-validation techniques are used to prevent overfitting and to ensure that the model generalizes well across different datasets.

1. Performance Metrics: Uses Accuracy, F-Score, and AUC (Area Under the Curve) to assess model effectiveness.

2. Cross-Validation: Splits data into training and testing sets to validate the model's generalizability and avoid overfitting.

3. Model Benchmarking: Compares performance across multiple algorithms to select the best-performing model.

4. Error Analysis: Identifies and addresses areas where the model underperforms, improving prediction reliability.

5. Quality Assurance: Ensures the final model meets accuracy and reliability standards for deployment in credit assessment.

6.1.5. Prediction and Decision Module

Finally, the Prediction and Decision Module delivers real-time creditworthiness scores for individuals. Based on the trained model, this module analyzes input data, such as credit history and income, to generate risk profiles. The results are displayed in an intuitive interface, allowing financial institutions to make informed decisions on loan approvals. The system also provides insights into the factors influencing an individual's credit score, ensuring transparency and aiding in risk management.

1.**Real-Time Prediction:** Generates individual creditworthiness scores based on live or inputted data.

2. **Risk Profiling:** Classifies individuals into risk categories, aiding in loan approval or rejection decisions.

3. **Transparent Insights:** Provides explanations for the factors influencing an individual's credit score, ensuring clarity.

4. **Integration:** Seamlessly integrates with banking systems to automate credit assessments for faster decision-making.

5. **Reporting:** Offers a user-friendly dashboard for visualizing credit scores, trends, and actionable insights for stakeholders.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7 .1 CONCLUSION:

Our implementation of credit scoring systems using XGBoost and LightGBM demonstrates the potential for accurate and efficient prediction of creditworthiness. These machine learning models excel in handling large datasets with diverse features, thanks to their ability to manage missing values, prioritize important features, and handle imbalanced datasets effectively. XGBoost offers robust performance through its gradient boosting framework and extensive parameter tuning, while LightGBM provides enhanced speed and memory efficiency due to its histogram-based algorithm. Both models outperform traditional credit scoring methods by offering higher predictive accuracy and scalability. Credit scoring systems play a crucial role in modern financial ecosystems, enabling lenders to evaluate the creditworthiness of potential borrowers with precision and efficiency. The integration of machine learning techniques like XGBoost and LightGBM has revolutionized credit scoring by offering highly accurate, scalable, and interpretable models. Both algorithms are gradient-boosting techniques that optimize decision tree ensembles, but each brings unique strengths to the credit scoring domain. XGBoost is known for its flexibility and robust handling of missing data, which makes it well-suited for datasets with diverse features and incomplete information. On the other hand, LightGBM excels in computational efficiency and scalability, making it ideal for large-scale datasets where quick predictions are crucial. Overall, their application enhances decision-making in financial institutions, fostering better risk assessment and informed lending strategies.

7.2 FUTURE ENHANCEMENT:

Future enhancements for credit scoring systems utilizing XGBoost and LightGBM can focus on improving model performance, interpretability, and adaptability. To enhance performance, advanced feature engineering techniques, su.ch as automated feature selection and domain-specific transformations, can be applied to capture nuanced patterns in customer data. Hybrid modeling approaches that combine XGBoost and LightGBM with deep learning architectures, like autoencoders or transformers, may further improve predictive accuracy by uncovering complex, non-linear relationships. Model interpretability can be enhanced using SHAP (SHapley Additive exPlanations) values or LIME (Local Interpretable Model-Agnostic Explanations) to provide transparent insights into feature contributions, enabling regulatory compliance and fostering trust among stakeholders. Adaptability can be improved by integrating real-time data streams, allowing the system to recalibrate dynamically and handle changing credit behaviors. Additionally, employing techniques like online learning and transfer learning can help the model generalize better to new market segments or regions. Ethical considerations, such as bias detection and mitigation strategies, can ensure fairness in decision-making, while secure integration with blockchain technology can enhance data integrity and transparency in credit evaluation processes.

APPENDIX 1 SAMPLE CODE

```
import pandas as pdimport plotly.graph_objects as goimport plotly.express
as pximport plotly.io as piopio.templates.default = "plotly_white"
data = pd.read_csv("C:/Users/anike/OneDrive/Desktop/Projects/Machine
Learning/Credit Scoring/credit_scoring.csv")print(data.head())
def calculate_credit_score(age, income, debt, credit_history,
employment_status):
# This function takes various parameters as inputs and calculates a credit
score.
# Define weights for each parameter
age_weight = 0.2
income_weight = 0.3
debt_weight = 0.2
credit_history_weight = 0.2
employment_status_weight = 0.1
# Calculate the weighted sum
weighted_sum = (age * age_weight) + (income * income_weight) - (debt *
debt_weight) + (credit_history * credit_history_weight) +
(employment_status * employment_status_weight)
```

```
# Map the weighted sum to a credit score scale (e.g., 350 to 800)
min_score = 350
max_score = 800
# Ensure the calculated score is within the defined range
credit_score = max(min_score, min(max_score, weighted_sum))
return credit_score
# Sample usage
applicant_age = 30
applicant_income = 60000
applicant_debt = 10000
applicant_credit_history = 3  # Years of credit history
applicant_employment_status = 1  # Employed (1) or unemployed (0)
# Get credit score
result = calculate_credit_score(applicant_age, applicant_income,
applicant_debt, applicant_credit_history, applicant_employment_status)
# Print the result
print("Credit Score: {result}")
import pandas as pd
from sklearn.model_selection import train_test_split
# Load the dataset
credit_data = pd.read_csv('credit_data.csv')
# Handle missing values
```

```
credit_data = credit_data.dropna()
# Encode categorical variables
credit_data = pd.get_dummies(credit_data, columns=['education', 'employment_status'])
# Normalize numerical features
credit_data['income'] = (credit_data['income'] - credit_data['income'].mean()) / credit_data['income'].std()
# Split the data into features and target variable
X = credit_data.drop('target_variable', axis=1)
y = credit_data['target_variable']
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
model = RandomForestClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Model Accuracy: {accuracy}')
from flask import Flask, render_template, request
import pickle
import numpy as np

app = Flask(__name__)

# Load the pre-trained ML model (LightGBM or XGBoost)
```

```
model_path = "model/credit_model.pkl"
model = pickle.load(open(model_path, "rb"))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':
        # Collect form data
        income = float(request.form['income'])
        debt = float(request.form['debt'])
        credit_history = int(request.form['credit_history'])
        repayment_behavior = float(request.form['repayment_behavior'])

        # Prepare data for prediction
        input_data = np.array([[income, debt, credit_history,
repayment_behavior]])
        prediction = model.predict(input_data)[0]

        # Map prediction to creditworthiness
        if prediction == 1:
            result = "Creditworthy"
        else:
            result = "Not Creditworthy"
```

```
    return render_template('result.html', prediction=result)

if __name__ == '__main__':
    app.run(debug=True)
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Credit Scoring System</title>
    <link rel="stylesheet" href="/static/style.css">
</head>
<body>
    <div class="container">
        <h1>Credit Scoring System</h1>
        <form action="/predict" method="post">
            <label for="income">Monthly Income:</label>
            <input type="number" step="0.01" name="income" id="income"
required>

            <label for="debt">Debt-to-Income Ratio:</label>
            <input type="number" step="0.01" name="debt" id="debt" required>

            <label for="credit_history">Credit History (Years):</label>
            <input type="number" name="credit_history" id="credit_history"
required>
```

```
        <label for="repayment_behavior">Repayment Behavior (Score):</label>
        <input type="number" step="0.01" name="repayment_behavior" id="repayment_behavior" required>

        <button type="submit">Check Creditworthiness</button>
      </form>
    </div>
</body>
</html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Result</title>
    <link rel="stylesheet" href="/static/style.css">
</head>
<body>
    <div class="container">
        <h1>Credit Scoring System</h1>
        <p>The individual is <strong>{{ prediction }}</strong>.</p>
        <a href="/">Check Another</a>
    </div>
</body>
</html>
```

APPENDIX 2 SCREENSHOTS

```
import plotly.express as px
import plotly.io as pio
pio.templates.default = "plotly_white"

data = pd.read_csv("C:/Users/anike/OneDrive/Desktop/Projects/Machine Learning/Credit S
print(data.head())
```

```
   Age  Gender Marital Status Education Level Employment Status  \
0   60    Male        Married          Master          Employed
1   25    Male        Married     High School        Unemployed
2   30  Female         Single          Master          Employed
3   58  Female        Married             PhD        Unemployed
4   32    Male        Married        Bachelor     Self-Employed

   Credit Utilization Ratio  Payment History  Number of Credit Accounts  \
0                      0.22           2685.0                          2
1                      0.20           2371.0                          9
2                      0.22           2771.0                          6
3                      0.12           1371.0                          2
4                      0.99            828.0                          2

   Loan Amount  Interest Rate  Loan Term   Type of Loan
0      4675000           2.65         48  Personal Loan
1      3619000           5.19         60      Auto Loan
2       957000           2.76         12      Auto Loan
3      4731000           6.57         60      Auto Loan
4      3289000           6.28         36  Personal Loan
```

FIGURE NO A.2.1

```
In [2]: print(data.describe())
        print(data.head())
```

```
                Age  Credit Utilization Ratio  Payment History  \
count  1000.000000                1000.000000      1000.000000
mean     42.702000                   0.509950      1452.814000
std      13.266771                   0.291057       827.934146
min      20.000000                   0.000000         0.000000
25%      31.000000                   0.250000       763.750000
50%      42.000000                   0.530000      1428.000000
75%      54.000000                   0.750000      2142.000000
max      65.000000                   1.000000      2857.000000

       Number of Credit Accounts   Loan Amount  Interest Rate    Loan Term
count                1000.000000  1.000000e+03    1000.000000  1000.000000
mean                    5.580000  2.471401e+06      10.686600    37.128000
std                     2.933634  1.387047e+06       5.479058    17.436274
min                     1.000000  1.080000e+05       1.010000    12.000000
25%                     3.000000  1.298000e+06       6.022500    24.000000
50%                     6.000000  2.437500e+06      10.705000    36.000000
75%                     8.000000  3.653250e+06      15.440000    48.000000
max                    10.000000  4.996000e+06      19.990000    60.000000
   Age  Gender Marital Status Education Level Employment Status  \
0   60    Male        Married          Master          Employed
1   25    Male        Married     High School        Unemployed
2   30  Female         Single          Master          Employed
```

FIGURE NO A.2.2

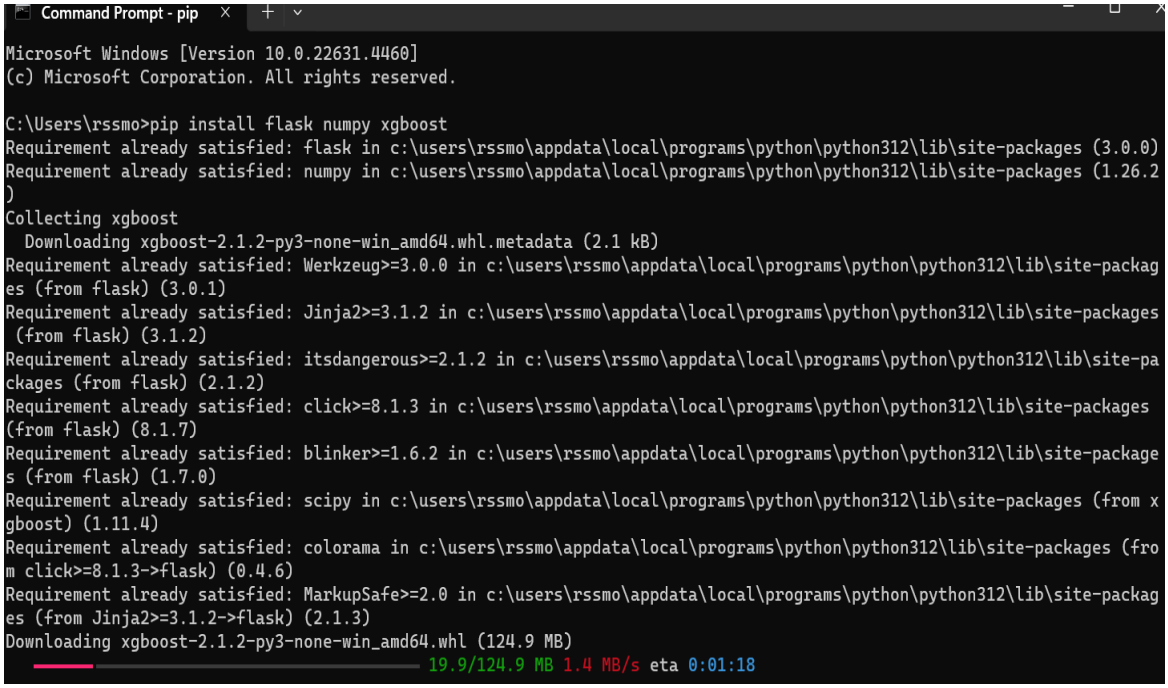

FIGURE NO A.2.3

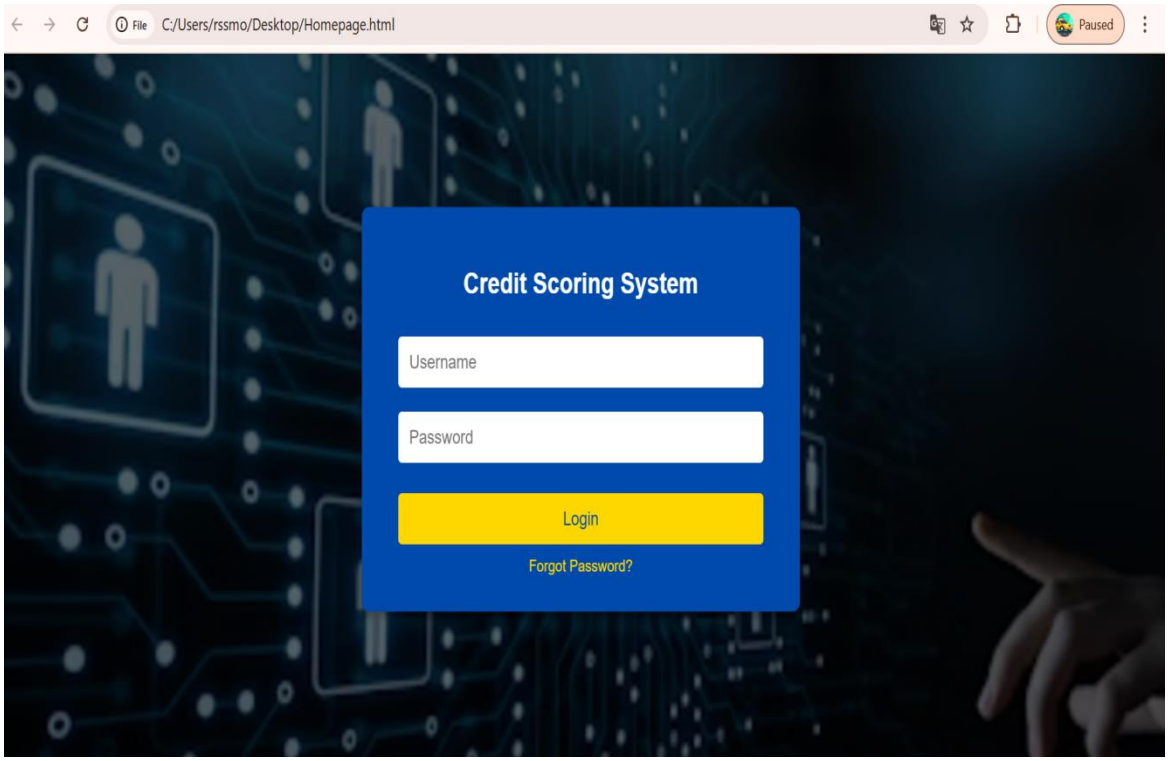

FIGURE NO A.2.4

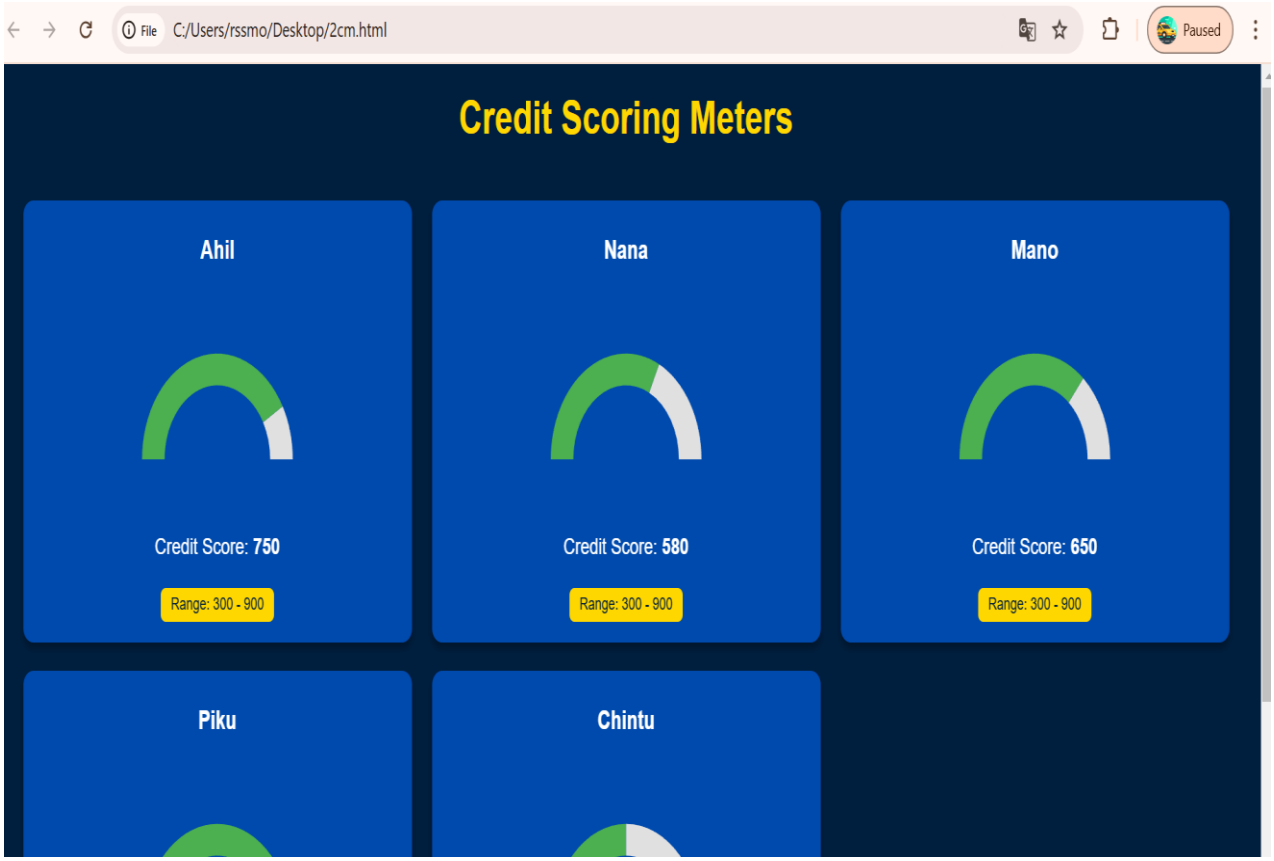

FIGURE NO A.2.5

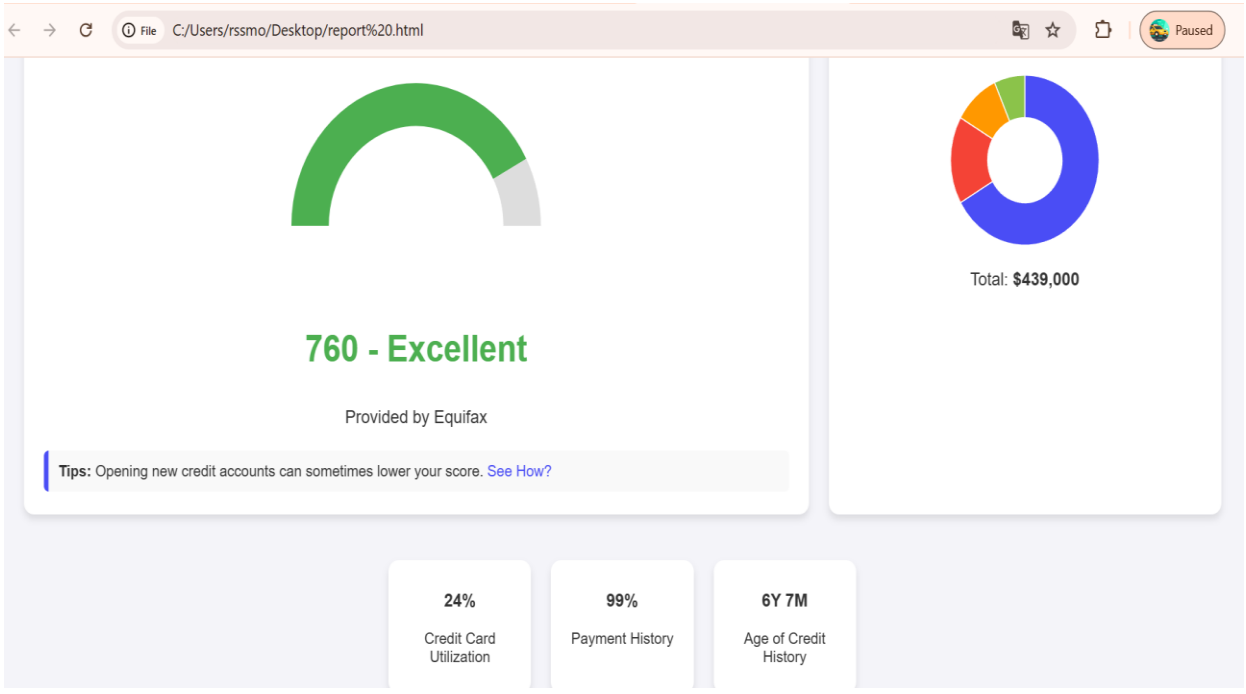

FIGURE NO A.2.6

REFERENCES

[1] Credit Score Prediction System using Deep Learning and KMeans Algorithms -Tripathi, D., Edla, D. R., Cheruku, R., & Kuppili, V. (2022).

[2] Credit scoring using machine learning and deep Learning-Based models-Deng L, Yu D (2024) Deep Learning: Methods and Applications. Found Trends Signal Proc 7: 197–387.

[3] A recent review on optimisation methods applied to credit scoring models-Abdou, A.J. and Pointon, H.A. (2023),

[4] Analyzing Machine Learning Models for Credit Scoring with Explainable AI and Optimizing Investment Decisions-Doshi-Velez, F.,Kim, B. (2021).

[5] Credit scoring methods: Latest trends and points to consider-Xiao H, Xiao Z, Wang Y. Ensemble classification based on supervised clustering for credit scoring. (2020)