Design and Development of an Intelligent Loan Eligibility Prediction System using Machine Learning & Explainable AI

by

ARNOB KUMAR DEY
22173012

A project submitted to the Department of Computer Science and Engineering in partial fulfillment of the requirements for the degree of
M.Engg. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
February 2025

Declaration

It is hereby declared that

1. The project submitted is my/our own original work while completing degree at Brac University.
2. The project does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The project does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Arnob Kumar Dey
22173012

Approval

The project titled "Design and Development of an Intelligent Loan Eligibility Prediction System using Machine Learning & Explainable AI" submitted by

1. Arnob Kumar Dey (22173012)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Engg in Computer Science and Engineering on February 3, 2025.

Examining Committee:

Supervisor:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md Sadek Ferdous, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

It can be seen that the value of assets is rising daily. That often necessary a lot of capital to buy the whole asset. Many times it becomes impossible to buy even from our savings. So we can apply for a loan to get the money we need because, through the loan application, we can easily get the money for buying assets. However, obtaining a loan is a lengthy procedure. The application must go through several steps before being accepted, and approval is not guaranteed. Many loan prediction models have been created to reduce the time and also reduce the risk attached to the loan. The main goal of my project was to compare different types of prediction models and then finalize which one is the best for loan-eligible prediction with the lowest amount of mistakes. Also used SHAP method for Feature importance of model prediction. After that choose the best ML model for deployment. So, I used eight machine learning models for my research paper. I get Logistic Regression is the best among the eight machine learning models with high accuracy and F1 score. For my application, I used this logistic regression model for prediction and also used SHAP for feature importance. After that, used text generation model L Lama-3.3-70b-versatile with Groq API to explain the reason for the prediction and give suggestions based on the prediction.

Keywords: Machine Learning, Loan Eligibility Prediction, XAI, Groq API.

Dedication

This work is lovingly dedicated to my dear mother and Dr. Golam Rabiul Alam Sir, whose guidance, support, and unwavering patience have been a source of strength and inspiration throughout this journey.

Acknowledgement

First and foremost, thanks to the Almighty God, this project was finished without any significant setbacks.

Second, I would like to express my sincere gratitude to my supervisor, Md. Golam Rabiul Alam, Professor in the Department of Computer Science and Engineering at BRAC University, for his helpful advice, academic direction, unwavering encouragement, and cooperative cooperation during the entire project's development. Without his priceless guidance, I could not have completed my assignment.

Thirdly, I also want to express my gratitude to everyone of the department's concerned instructors and staff for their direct and indirect support throughout various work-related occasions.

Lastly, without my mother unwavering support, it would not have been feasible. I am currently approaching our post-graduation with her kind assistance and prayers.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Banks make a chunk of their money through loans. Revenue comes when banks provide loans because they take interest on loans. Banks offer customers money to keep at a high interest rate. And this money is offered to customers as bank loans from which the bank earns interest. Besides, loans are also important in the economy by financing the people and businesses for investment and expanding as well as growing. This in turn leads to economic development which in turn is good for the banks in that they get more business opportunities. Banks offer multiple types of loan products (personal loans, mortgages, business loans, etc.) set up as a way to rebalanced funds between people and businesses and help reduce risk for lenders. This diversification is important so that the banks do not suffer losses in case of a sector wise business decline. Banks can use offering loans to build up strong relationships with customers, sometimes exponentially, since customers who take loans will often buy other financial products that banks offer, such as insurance, investment services and more. That's Why loans are important for banks. On the other hand, we need loans because it is important for an individual, a company, and an economy. All types of businesses need loans to get up and running, to grow, or to sustain operations. Such as this is from funding to purchase equipment, to the hiring of staff, to managing inventory and to venture into new markets, etc. Because, a number of businesses would also find it difficult to start and grow without loans.
Loans allow individuals to make significant life purchases but these purchases were not possible as it required access to funds that were not readily available sometimes to the borrower. But personal loans can also go toward debt consolidation, smoothing cash flow, or improving the quality of one's life.
For most people, it is financially impossible to buy a home outright with cash. Individuals make home loans, known as mortgages, that help them pay their home loan in installments while enabling them to own a home. Not only does this investment give you a place to live but it does so in a way where you are also building equity that can be a very valuable asset.
Automobiles are costly and also essential for going to work or handling life. Auto loans allow people to provide installment of car's loan throughout several years and giving them reliable transportation without using all of their savings at once. Investing in education is an important step in one's future, but it's expensive. The funds to enable students to pay tuition fees, books, and other charges are provided by student loans to allow as many people as possible to get to college. This means having better access to the education

necessary to attain a well paying job at a higher earning potential and greater stability financially.
Medical emergencies, sudden job loss or urgent home repairs can be quite unexpected and suddenly turn crippling on finances. Personal loans facilitate loans, in the sense that they help people take care of these crises before they resort to high interest options like credit cards or payday loans. This access can stop financial instability and give you peace of mind during difficult times.
An agricultural loan is a financial product used to finance crop cultivation, livestock rearing, land development, building of storage facilities as well as purchasing of equipment. Short, medium or long term loans with subsidized interest rates at the rate affordable and flexible payment terms that can be matched to the farming cycle. Small scale farmers are empowered by governments through special schemes meant to encourage rural development. Modernizing farming, food security and rural economy depends a lot on agricultural loans.
However, at present, in many banks loans are manually authorized by a bank representative. For a number of reasons, manually predicting loan eligibility is always considered risky. Traditional ways are already there and exist for a long period of time, but they offer numerous drawbacks to both the lender and the borrower. Manual loan prediction, in particular, requires human judgment, which is inherently error-prone, inconsistent and biased. An applicant presented to different loan officers may get evaluated differently based on the personal experience of the loan officer, his intuition or his understanding of the financial metrics. If it's not consistent, it means that some good borrowers will be denied and some unsafe borrowers will be granted a loan. Additionally, Manual assessments are usually based on a few select data points. While this narrow approach can easily miss key insights and result in suboptimal decisions regarding an applicant's ability which is sometimes incorrect.[13].
Manual assessment of loan applications tends to be slow and laborious, requiring a detailed perusal of financial papers, credit histories and other supporting material. Delays in loan approval can affect the applicant in a negative way and can eventually lead to losing business to competitors who offer slicker and effective processes. This type of method is usually a time and resource consuming process, and hence is relatively expensive. On the other hand, a lender can process much larger volumes of applications much more efficiently, and with far greater accuracy, by using an automated system, reducing costs to the lender[18].
Higher Default Rates created by manually loan prediction. Manual predictions of the risk profile of an applicant might not be complete, giving higher probability to loan defaults even if it has been approved. Most lenders contend that, in the absence of thorough loan assessment, they are exposed to more financial risk for they attract more loans that are likely to default and affect their profit margins.
Because manual loan prediction depends entirely on human judgment, only limited data analysis is available, and human biases may not be completely removed. These risks can result in poor decisions, financial loss and compliance issues, meaning it is an increasingly outmoded way to operate, even in the age of data-driven decision making.
For this reason, it is very important to have the ability to have the loan authorized in the shortest time possible, with the fewest number of mistakes in risk prediction is critically feasible.
In recent years, data science, and machine learning specifically, has brought in powerful tools to improve the process of loan approval. Using historical data, the machine learning

models have created which can spot these very complex relationships between those kinds of variables: income, employment history, credit behavior, and other socio economic factors. With high accuracy, these models can predict the likelihood of loan repayment so lenders can automate decisions, reduce processing time and increase consistency in evaluation. Therefore, we need a loan prediction model. Thus, this type of model can predict in time, whether the given applicant's loan will be approved or not.

1.2 Aims and Objectives

The project's main goal is to evaluate loan eligibility prediction using multiple machine learning algorithms and then select the best one that may reduce loan approval time and also risk. And also using the XAI method to explain the prediction. After that choose the best model for deployment. It has been completed by forecasting whether or not the loan can be provided to that individual based on a variety of factors such as Married, Education, Loan Amount, Applicant Income, and so on. The prediction model benefits both the applicant as well as the bank through reducing risk and lowering defaulters number.

1.3 Organization of the Report

The report is organized in the following sequence:

Chapter-1 describes the motivation, aims and objective of my project work, the organization of the report structured,.
Chapter-2 discuss about the related work about loan eligibility prediction.
Chapter-3 demonestrates Exploratory Data Analysis and visualization
Chapter-4 discuss about the Methodology, Data Preprocessing, Model Specification & Explainable AI
Chapter-5 show Result Analysis
Chapter-6 describes the Project Development process.
Chapter-7 presents the conclusion.

Chapter 2

Related Work

The prediction of loan eligibility has become a very important area of research in the area of financial services where accurate decision-making is required to approve the loan. Predicting loan eligibility became a lot more efficient and accurate while using machine learning techniques than that traditional methods provide. Traditionally, credit scoring models were almost entirely based on historical credit data, neglecting other important ones, such as income and employment status. However, these modern approaches include a wider set of attributes, and utilize sophisticated algorithms in order to produce more accurate predictions.

Uddin, Nazim, et al. developed a machine learning (ML) powered loan prediction system to automatically spot eligible candidates for loans by addressing this problem. The thorough research examines data preprocessing procedures and efficient SMOTE balancing alongside an exploration of multiple machine learning models including Decision Trees, SVM, KNN, Gaussian Naive Bayes, AdaBoost, Gradient Boosting and deep learning techniques such as recurrent neural networks, deep neural networks and long short-term memory models. The investigators carefully assess the accuracy and recall performance and F1 score measurements of their model. The experimental findings from their investigation demonstrate that Extra Trees produce superior results compared to alternative models. By combining the top three ML models into an ensemble voting model they achieve exceptional accuracy for bank loan default predictions which exceed Extra Trees by 0.62. They developed a user-focused desktop application which improves interface usability[10].

The objective of Özkurt, Cem in his research paper is to improve the forecast of eligibility for the personal loan using fancy machine learning algorithms. In order to minimize risk and expedite the lending process, financial companies necessarily need to accurately calculate creditworthiness. He used a large dataset including financial and demographic data to assess three algorithms: XGBoost, GradientBoost and AdaBoost. The highest performing on this criterion was found to be XGBoost with an F1 score of 0.95%, accuracy, precision, and recall of 95%. We find that these results demonstrate how well XGBoost can predict which borrowers have the probability of paying their loans back, hence making it a useful tool for loan approval decision-making. XGBoost may therefore help banks decrease the risk of default, deliver better customer service, simplify processes, and ultimately come up with better and more reliable lending practices[15].

In order to forecast loan eligibility, Orji, Ugochukwu .E., et al. in their paper, provide six (6) machine learning algorithms. The models were trained on a historical "Loan Eligible Dataset" which is a released database under the Database Contents License (DbCL) v1.0 available on Kaggle. Their studies showed superior accuracy with the Random Forest scoring 95.55%, Which is the best score & Logistic Regression at 80%, which is the lowest score. Their models performed better than the two in terms of precision-recall and accuracy among three different loan prediction models that were published in the literature[7]

For his research, Chen, Guangxuan selects a Kaggle dataset that is open to the public, cleans data meticulously, before training eight frequently used models simultaneously. Accuracy, precision and F1 score are the measures that are used in this case. From which, AdaBoost was the best model as it achieved the highest F1-score 0.8957 and the highest accuracy 84.95%. Finding the best precision performance was accomplished using XGBoost. This paper demonstrates how the concept of machine learning can indeed improve sound financial decision-making and provide a more precise means of loan approval. It assists in simplifying the loan approval procedures, increasing loan accessibility, and enabling loan applicants assessment for the chances of getting the loans by the relevant financial institutions.[18]

In their study, P, Krishnaraj, et al. investigate many machine learning approaches applied in different contexts, review their benefits, drawbacks and performance metrics, including loan approval.s. In order to predict and categorize the objective variable, they used three well-known machine learning models in this research. Which are Logistic regression, Decision trees, and their extension, Random forest. The general goal was to do a comparative analysis for different models and to find out which model works best for the given job. After a deep analysis and also comparing the model results, they found that the Logistic Regression machine learning model was marginally better compared to the other models. Their evaluation yielded an 86% accuracy with the logistic regression method. Test accuracy is the percentage of times the test dataset accurately predicted instances. Logistic regression model has an accuracy of 86% making it pretty decent to produce predictions depend on unseen data[13]

For the purpose of classifying loan approvals, Noviandy, Teuku Rizky, et al. used LightGBM, a gradient-boosting framework that was verified using 10-fold cross-validation and refined through the use of Random Search hyperparameter tuning in their paper. To tackle the interpretability problem in machine learning, we implemented the Shapley Additive exPlanations (SHAP) framework. But in recall (97.17%), accuracy (98.13%), precision (97.78%), and F1-score (97.48%), the LightGBM model performs better than traditional approaches (Decision Tree, Random Forest, AdaBoost, and Extra Tree). The study shows that the accuracy and transparency of loan approval decisions may be greatly increased by utilizing an interpretable machine learning technique with LightGBM and SHAP. With the use of this technique, financial institutions may improve their loan approval processes and make decisions in the digital economy more dependable, effective, and transparent.[14]

Loans are a vital source of revenue for the economic industry however they entail serious economic hazards. A sizable amount of a bank's assets are made up of loan interest.

Globally, there is an increasing need for loans, and businesses are coming up with effective ways to draw in new customers. Dansana, Debabrata, et al. in their article's goal is to decide whether to lend money to particular people or businesses. To gauge performance and find qualified borrowers for loan approval, the Random Forest Regressor model has been applied. According to the model, bankers should take into account additional consumer attributes that are crucial for determining defaults on loans and issuing credit, in addition to focusing on wealthy customers. The study looks at a number of factors that affect loan acceptance, including marital status, employment type, business kind, gender, educational background, and loan term. The research also examines the quantity of loans that are granted, drawn, and denied, which offers important information about approval of loans and forecasting.[11]

Rohini, Dr.S.Gomathi alias et al. used The Jupyter Notebook cross-platform Integrated Development Environment (IDE) and Python 3.7 and associated libraries to create the system. They used four machine learning model. Based on their average balance amount in the bank during the period and their financial background, the system authorizes the loan amount to the anticipated applicants having a good credit record. The system's ability to forecast applicants' financial worth was determined. The accuracy of the Naive Bayes, SVM, and XGBoost algorithms in predicting creditworthiness was 79%, 81%, and 84%, respectively.[19]

Madaan, Mehul, et al. conducted a comprehensive and comparative study regarding two algorithms, Random Forest and Decision Trees. Both approaches made use of the same dataset. An even closer look found that the Random Forest algorithm performed better comparison the Decision Tree, with the latter showing a much greater success rate. The Random Forest Classifier gave them an accuracy of 80%, one the other hand the Decision Tree method gave them an accuracy of 73%. Consequently, it seems that the Random Forest is a more appropriate option for this kind of data. This work investigated, examined, and created a machine learning algorithm to precisely ascertain a person's likelihood of loan default based on a collection of attributes.[4]

Tejaswini, J., et al. provide a comprehensive analysis of the prediction of a customer's loan application acceptance based on 3 machine learning algorithms. They use Random Forest and Decision Tree as well as Logistic Regression. According to the experiment's results, the Decision Tree algorithm is more accurate than the Random Forest and Logistic Regression machine learning algorithms. What's nice about this part is that it's easily integrable with a big range of different systems. [1]

In their research paper, Bansode, Nikhil et al. employ five different machine learning algorithms to predict customer loan approvals. Nevertheless, logistic regression is the most accurate algorithm with 84.376% accuracy. It handles flexibility, categorical values, and it may deal with overfitting, while it's also quite good at visualizing the data via a confusion matrix. It can also finish the missing values of datasets. If applicants have the chance of loan default, they have a greater possibility of being denied with bad credit.[5]

Yussuph, Toyyibat T. in his study forecasts loan defaults and determines the optimal loan levels for small firms using advanced machine learning techniques, including the XGBoost and Random Forest algorithms. A dataset with 27 variables—including loan

features, borrower information, and loan outcomes—is examined. The findings show that Random Forest and XGBoost are both good at predicting loan default, with XGBoost having a small advantage. For eligible borrowers, loan amounts are also estimated using linear regression. The study pinpoints the main causes of loan defaults, and Random Forest projections heavily rely on variables including term, disbursement gross, SBA approval, and gross approval .[20]

Zuama, Robi Aziz, et al. in their study examined how the use of machine learning model can be applied to predict loans based on consumer behavior. In this paper they explore the role ML can play in loan result forecasting. In particular, ML has been proven to be a revolutionary tool for Loan default prediction using more sophisticated approaches to analyses the historical data related to the behaviours of the consumer. Findings show that with 89% accuracy rate, XGB Outperforms other machine learning methods. Logistic regression and random forest have an accuracy of 88%. KNN and decision trees follow, with both slightly lower accuracy rates (87%).[17]

In order to assist in future decision-making, Perera, C. L., et al. in their research suggested an approach for evaluating the credit risks connected to loans. To offer insights, a data analysis based on exploration was conducted. In order to predict loan credit risks using Machine Learning (ML) algorithms, this study evaluated consumer profiles according to the demographic as well as geographic information of the consumers. Finally, the efficiency of the model was evaluated using evaluation metrics. The best training and test accuracy of 0.99% and 0.78%, respectively, were achieved by the Stacking Ensemble, outperforming the other methods. The research presented here is interesting since it collected extensive data from one of Sri Lanka's top financial institutions. The most important steps in the research are feature selection, machine learning algorithms, hyperparameters, and how to assess. Additionally, voting-based ensemble learning—a unique machine learning technique-was suggested as a way to improve results.[16]

Chapter 3

Exploratory Data Analysis and visualization

3.1 Data Overview

The dataset Is collected from the Kaggle. In my datasets, there are 13 columns and 614 rows available. Table Table 3.1 describes the key information of the data set.

Attribute Name	Description
Gender	Male/Female
Married	Yes/No
Dependents	Number of Dependents
Education	Graduate/Under- Graduate
Self-Employed	Yes/No
Applicant Income	Applicant's Income
Co-applicant Income	co-applicants Income
Loan Amount	Loan Amount in Thousands
Loan Amount Term	Term of Loan in Months
Credit History	Good or Bad
Property Area	Urban/Semi- Urban/Rural
Loan Status	Loan Approved or Not

Table 3.1: Overview of the dataset

In the data set, we can see that there are some string-type of data and some int-type of data existing. In Table.3.2, we see what type of data is in our dataset. Also we can see that some of the columns have some missing values.

Attribute Name	Non-Null Count	Type
Loan ID	614	String
Gender	601	String
Married	611	String
Dependents	599	String
Education	614	String
Self-Employed	582	String
Applicant Income	614	Integer
Co-applicant Income	614	Integer
Loan Amount	592	Integer
Loan Amount Term	600	Integer
Credit History	564	Integer
Property Area	614	String
Loan Status	614	String

Table 3.2: Data info

3.2 Bar Plot Analysis

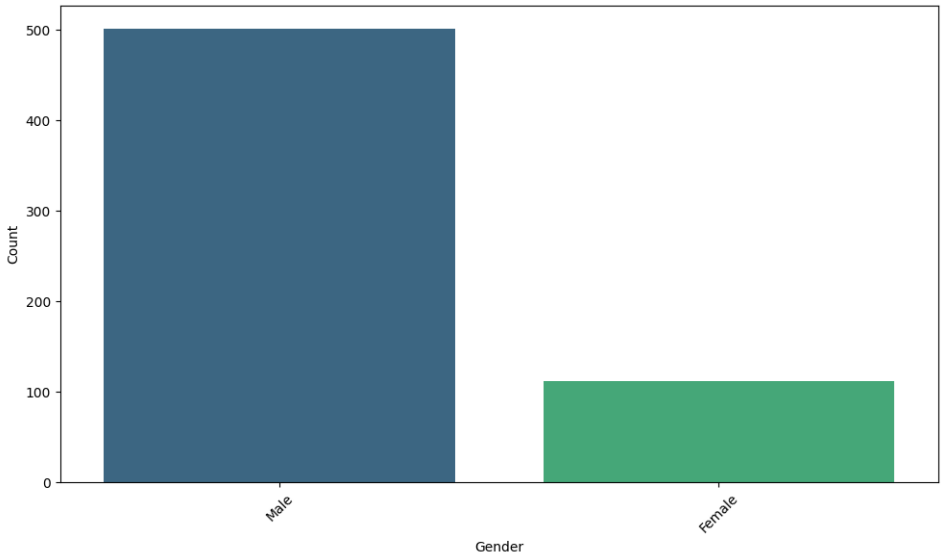


Figure 3.1: Bar plot of Gender

Figure 3.1 plots the gender distribution within a data set. In this plot, the gender categories (Male and Female) are represented by the x-axis. Here, y-axis represent the frequency or count of each category. There are about 500 counts in the male category and about 100 counts in the "Female" category. It's show that large number of male in the dataset.

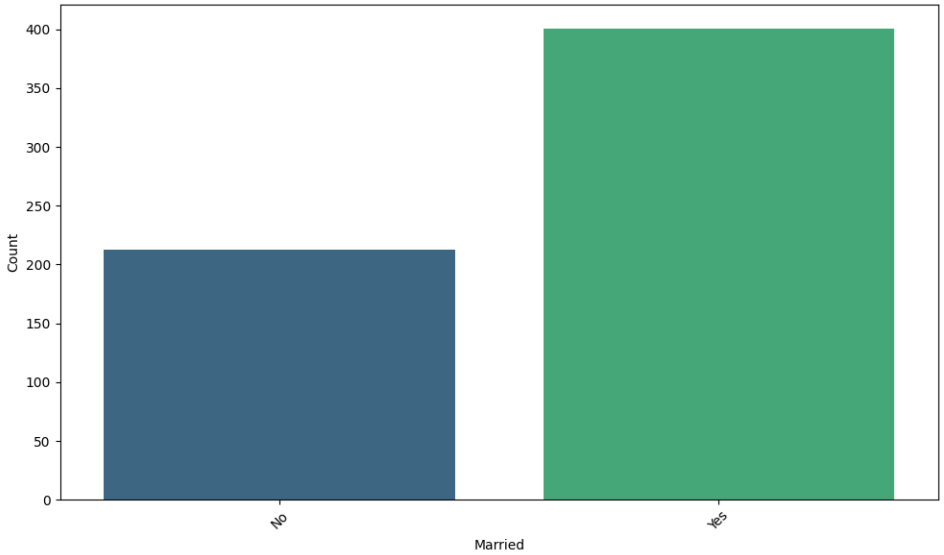


Figure 3.2: Bar plot of Married

The married distribution within a dataset is plotted in Figure 3.2. The x-axis in this visualization represents the Married categories (Yes and No). Here, the y-axis is the frequency or count of each category. The "No" category has above 200 people, the "Yes"about 400 people. This means that in the dataset there were more married people than not married.

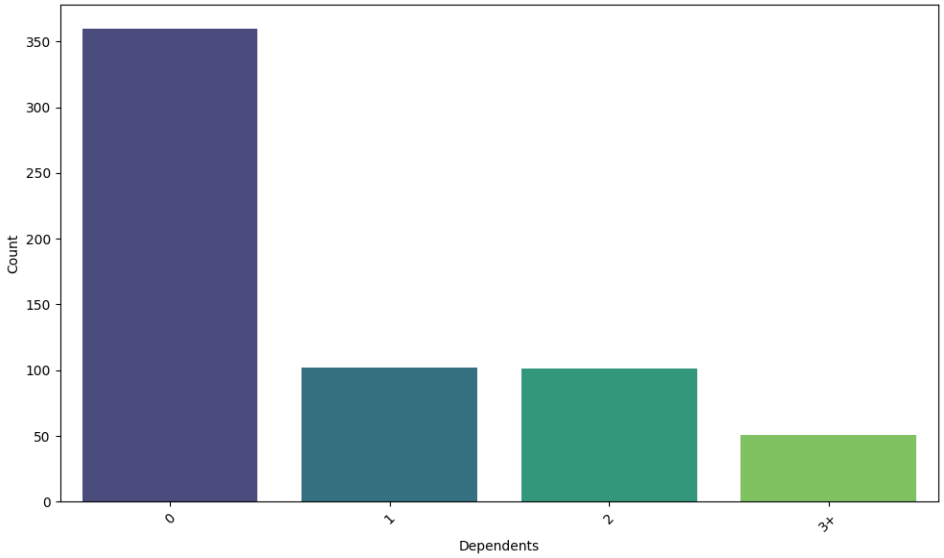


Figure 3.3: Bar plot of Dependents

This bar plot (Figure 3.3) represents the distribution of dependent variables in a data set. The number of dependents is shown on the x-axis and on the y-axis is the count or the frequency of each of the kinds. Most people have 0 dependents, and the number is more than 350. From this, we conclude that most of the individuals in our dataset are predominantly dependent-free. Out of the three categories, one and two have counts which are just near 100, whereas, lacking enough count marginally above 100, the 3+ group has the lowest number, which implies fewer numbers, around 50 of people having

3 or more dependents.

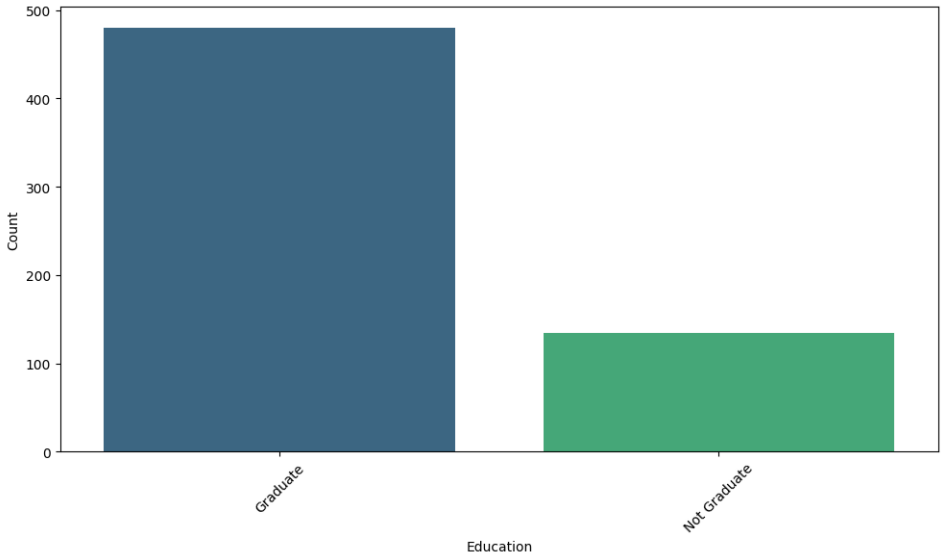


Figure 3.4: Bar plot of Education

Figure 3.4 represents the distribution of educational credentials of a dataset. The x-axis represents two categories. are respectively “Graduate ” and “Not Graduate”. On the other hand, the y-axis shows how many people (or frequency) are in each category. The “Not Graduate” group has a count of less than 200 and the “Graduate” group has a count of approximately 500.

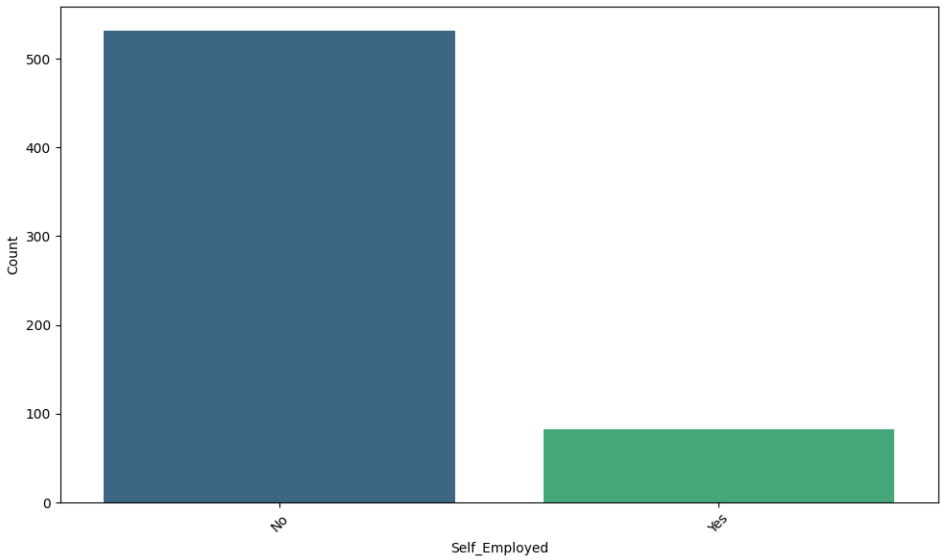


Figure 3.5: Bar plot of Self_Emlpoyed

This Figure 3.5 illustrate the self-employed population distribution of this dataset. On x-axis we see the Self_Emlpoyed categories, i.e. Yes, No, and on the y-axis the count or the frequency of each category. The majority of the people are not self-employed which is approximately 500 and about 100 are self-employed.

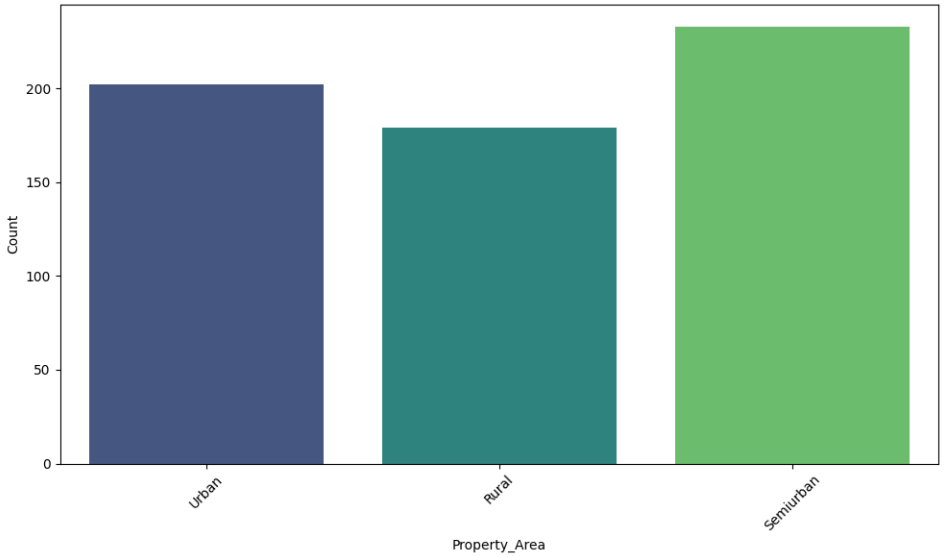


Figure 3.6: Bar plot of Property_Area

The distribution of the Property_Area variable among three classes: urban, rural, and semi-urban are demonstrated in Figure 3.6. On the y-axis is the number of data points (frequency) and on the x-axis are the categories. Semi_urban has the highest count, indicating that it is the most popular property area category in the dataset. The next most prevalent category is urban. Rural is the least prevalent category, with the lowest count.

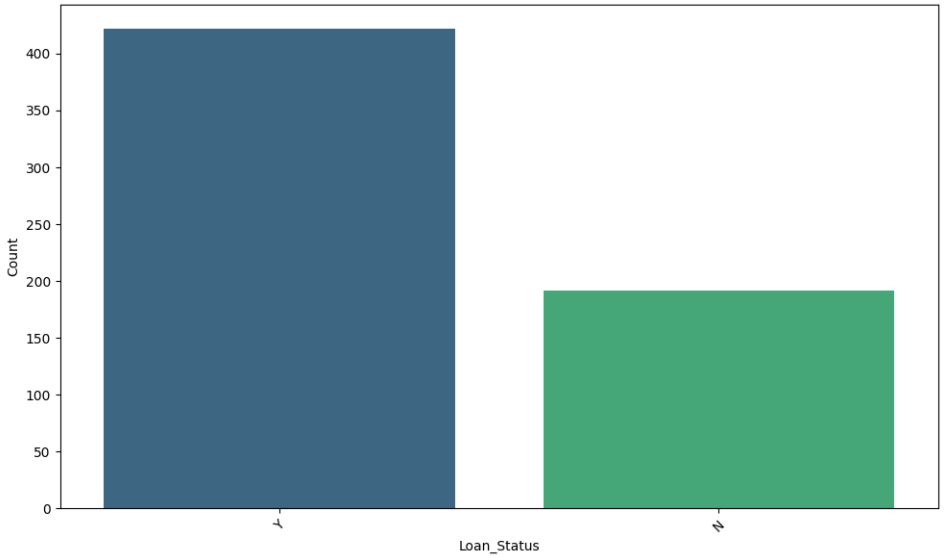


Figure 3.7: Bar plot of Loan Status

Figure 3.7 shows the overall distribution of the Loan_Status variable in a loan eligibility prediction dataset. The two categories that are seen on the x-axis represent "Y" and "N" both, where "Y" stands for approved loans and "N" stand for unapproved loans. On the y-axis, the number of times each category of the variable occurred is shown (frequency). Y: 500, it was closer to, Which means that most of the loan applications in the dataset were accepted. Fewer applications were rejected a low count in the Loan Not Approved

(N) category.

3.3 Histogram Plot Analysis

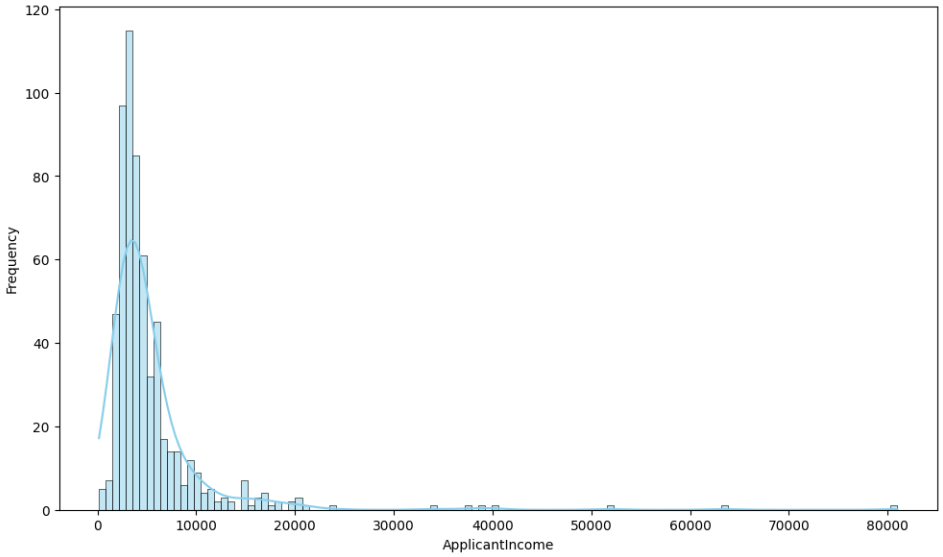


Figure 3.8: Distribution of Applicant income

The histogram above Figure 3.8 represents the "ApplicantIncome" variable available in the loan prediction dataset. Finally, the range of possible applicant income values is plotted along the x-axis, and the number of applications within each income in the monthly range is plotted along the y-axis. Smooth the distribution by making a line overlay (most likely a density plot). The vast majority of applicants make less than 10,000. There is some extreme outliers with very high earnings (over 40,000) that pushes the tail of the distribution to the right. The majority of candidates earn between 2,000 and 6,000.

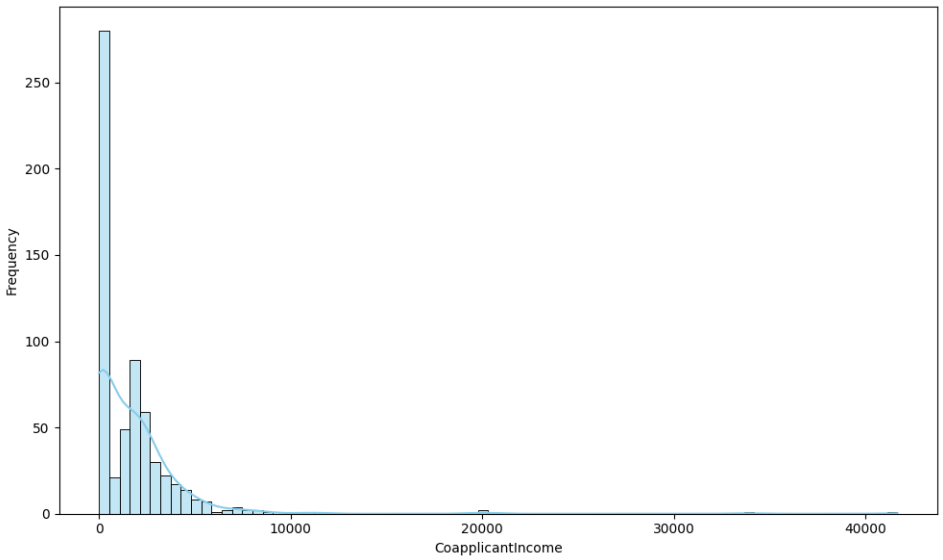


Figure 3.9: Distribution of Co-applicant income

Figure 3.9 shows the distribution of values of the "CoapplicantIncome" variable in a loan prediction dataset. Here the x-axis depicts co-applicant income levels in monthly, while the y-axis depicts the frequency (count) of co-applicants within each income bracket. A density curve is superimposed to provide a smooth depiction of the distribution. The highest frequency is shown in the first bar (at Co-applicant Income = 0). This indicate that a large number of candidates lack a co-applicant to provide income. This histogram is right-skewed, with a long tail extending towards higher incomes and the majority of values concentrated at lower income levels, much like the applicant income distribution. In contrast to the vast majority, the tail displays a tiny percentage of extroverted co-applicants with substantially higher earnings (e.g., above 10,000). Most of the Co-applicant's earning income 0 to 10000 range.

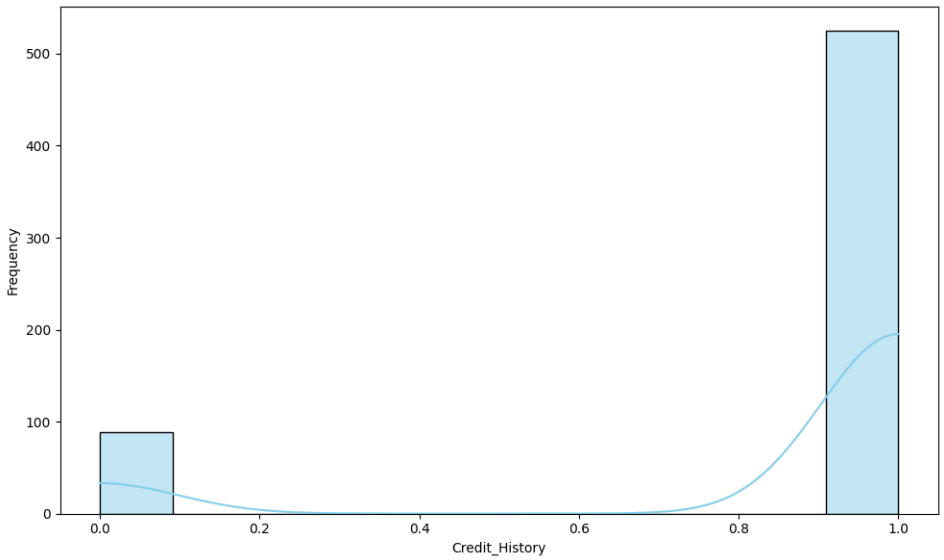


Figure 3.10: Distribution of Credit History

One important aspect in determining loan eligibility is the "Credit_History" variable, whose distribution is shown in Figure 3.10. Here Credit_History = 1, it means that the applicant has a solid credit history and has previously made responsible loan or debt repayments. The majority of customers have strong credit histories (a big bar at 1). On the other hand, Credit_History = 0 denotes the lack of previous credit conduct or an untrustworthy or bad credit history. Candidates who were with little or no credit record are represented by a smaller group (the bar at 0), which is near 100.

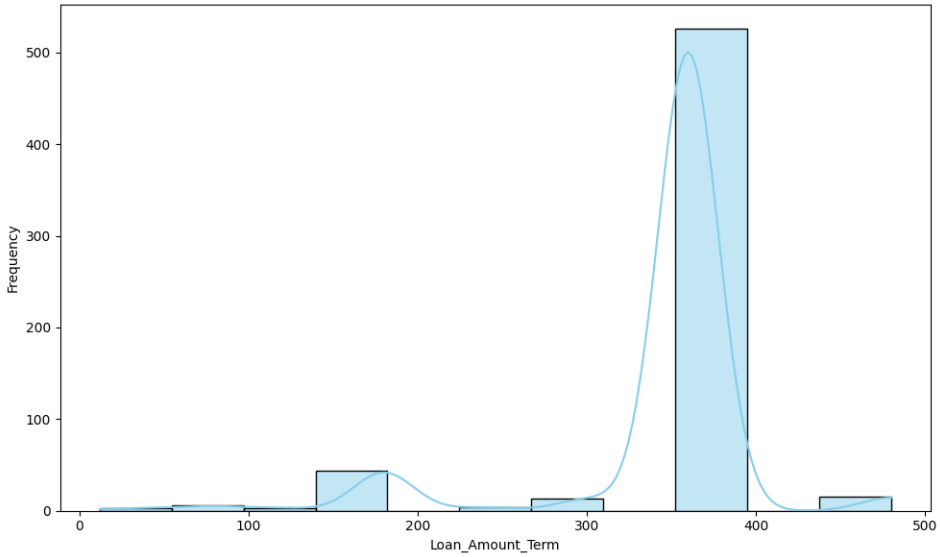


Figure 3.11: Distribution of Loan amount term

Figure 3.11 illustrates the distribution of the "Loan_Amount_Term" feature of a loan eligibility prediction dataset. The x-axis (Loan_Amount_Term) reflects the loan period, which is often expressed in days (12, 36, 60, etc.). The y-axis (frequency) shows how often every single loan term period comes up in the dataset. With a notable increase close to one specific figure (360 Days), the graph is heavily biased to the right, suggesting that most loans have a long duration (about 30 years, a typical loan period). For other periods, such as for 180 numbers, the frequencies are less. Loan terms as short as 100 months (8 years) and beyond 400 months (33 years) are very rare.

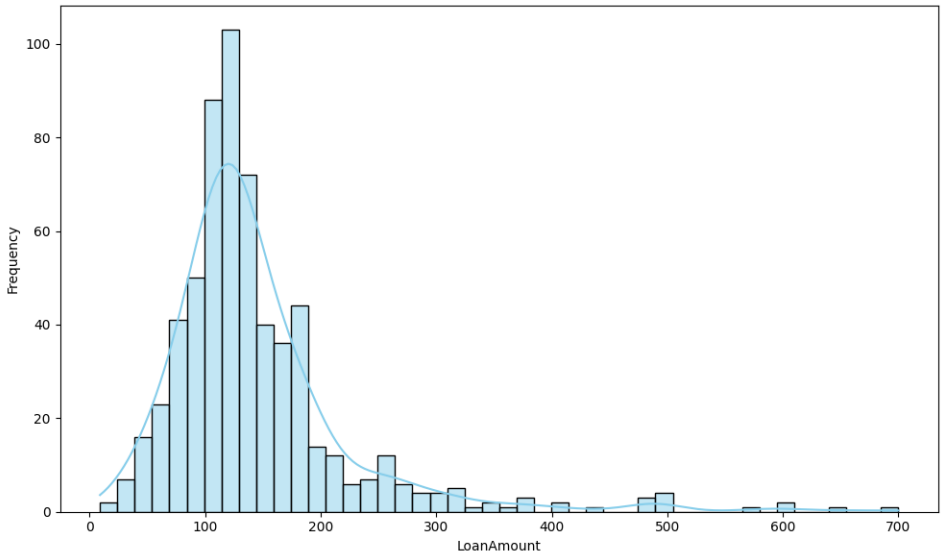


Figure 3.12: Distribution of Loan Amount

"Loan Amount" is a major determinant in forecasting a customers loan eligibility and the histogram of the distribution of this factor is shown by Figure 3.12. The loan amounts values are represented by the x-axis (LoanAmount). The loan amount distribution is right-skewed(positively skewed). Here, Loan amount is represented in thousands. With limited examples of bigger loan amounts (more than 300), the majority of loan amounts

come under a lower range, usually within 100 and 200 (in thousands). Most of the loan amount is in this range. Above this level, the frequency sharply declines while the loan amount rises. Higher loan amounts (over 400, up to 700) remain quite rare.

3.4 Scatter Plot Analysis

For the relation between two variables, now I visualize the scatter plot for all numerical data in my dataset.

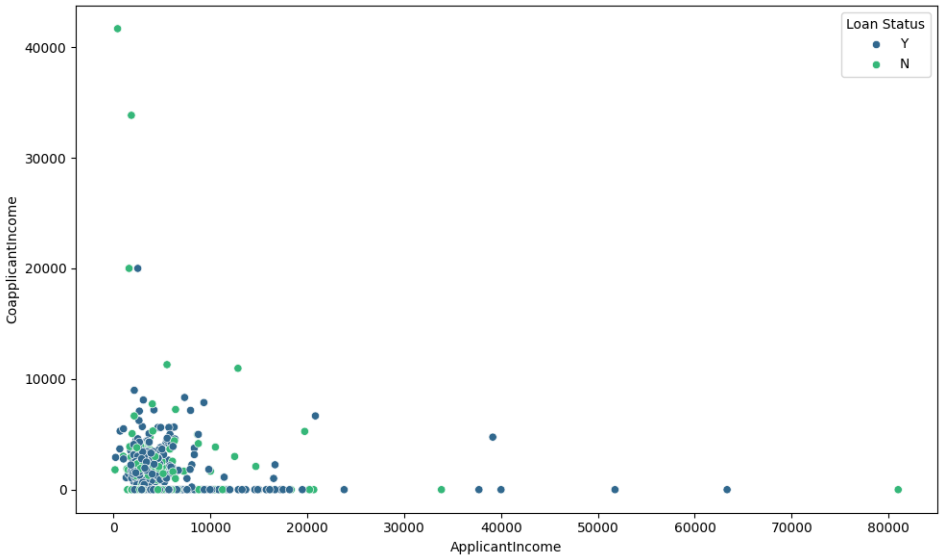


Figure 3.13: Scatter Plot: ApplicantIncome vs CoapplicantIncome

Figure 3.13 show "ApplicantIncome" against "CoapplicantIncome" with a categorical label of loan eligibility. X-axis (ApplicantIncome) It is the income of the applicant in the loan application in the monetary unit. Y-axis (CoapplicantIncome), Explains the co-applicant's income in terms of monetary unit. The every dot on the graph represents a loan application. The color of the points indicates Loan Status: Blue (Y): Loan approved. Green (N): Loan not approved. In this plot, most of the points lie toward the bottom-left quadrant meaning low Applicant Income and low Co-applicant Income. This in turn implies that most of the applicants and co-applicants earn less than this income earning level. The majority of the loans are presented in the area with low Applicant Income and Co-applicant Income and many of them are approved (blue color points). Hence, green points here seem to be indicative of a fair degree of dispersion but are actually more clearly discernible as the income values rise. Meaning that apart from income, credit history or other financial indicators constitute highly important factors in the loan decision.

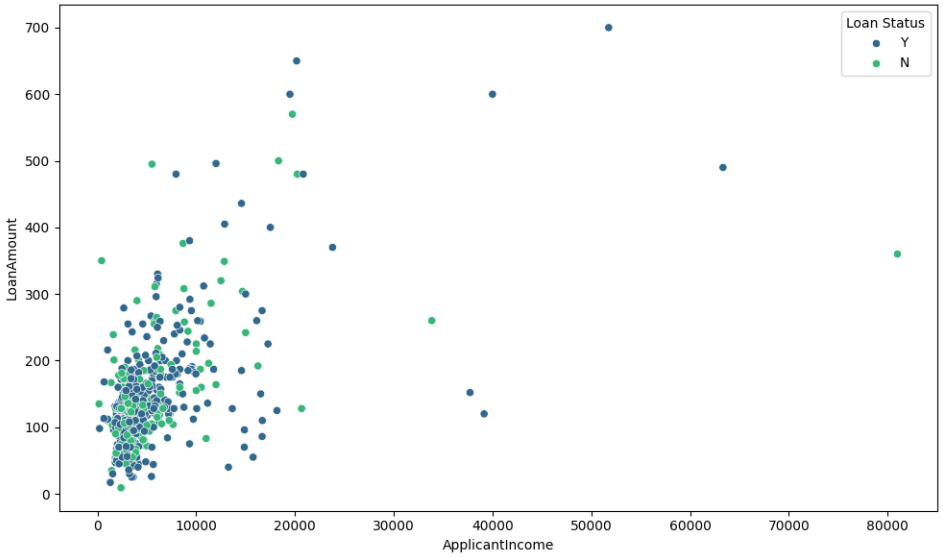


Figure 3.14: Scatter Plot: ApplicantIncome vs LoanAmount

In this case, a scatter plot is plotted with "ApplicantIncome" and "LoanAmount" along with a categorical variable loan status which is show in Figure 3.14. In both ApplicantIncome (0 to 10,000) and LoanAmount (0 to 200)(in thousands), we see a large cluster of points accumulating in the lower ranges. So, it seems that most loan applicants request lower loan amounts and also have lower incomes. The densely populated region below shows that loan approvals (blue) vastly outnumber loan denials, indicating that loans are being approved more often for lower-income applicants requesting smaller loan amounts. We see loan rejections (green) scattered throughout the graph, but are more visible for higher Loan Amount values or outliers.

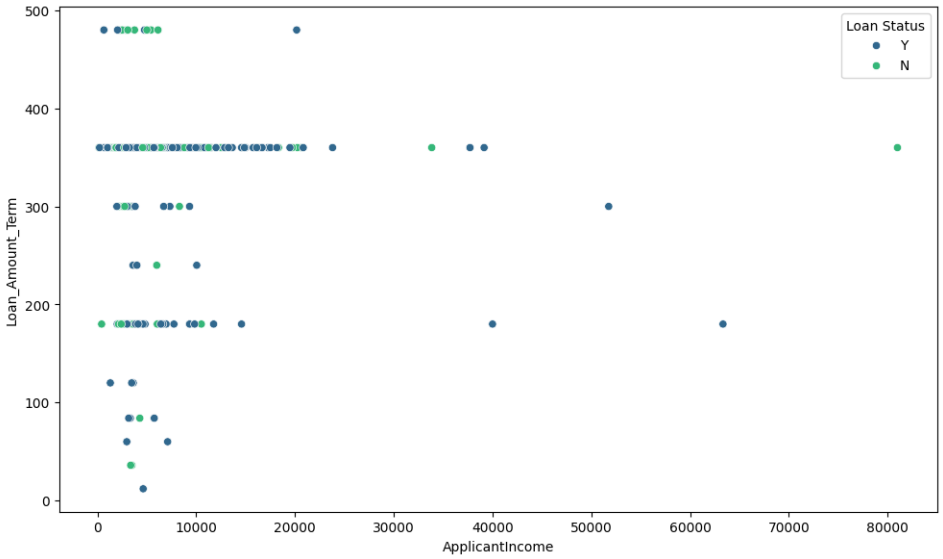


Figure 3.15: Scatter Plot: ApplicantIncome vs Loan_Amount_Term

This scatter plot(Figure 3.15) explores the relationship between "ApplicantIncome" and "Loan_Amount_Term", with the loan eligibility status. We can see that most of the data points are occurring around 360 months (30 years) loan amount term, irrespective of Applicant Income, which means that 360 months (30 years) is the standard loan amount

term for most loans. The remaining loans have terms of 180 months (15 years), 300 months (25 years), and shorter term loans throughout the income spectrum. Furthermore, the Loan Status (approved or rejected) does not exhibit a clear pattern on Applicant Income as the count of loans approved (Y) and rejected (N) across different Applicant Earnings are relatively similar, and there is no clear pattern of count of loans approved (Y) and rejected (N) across different Loan Amount Term means.

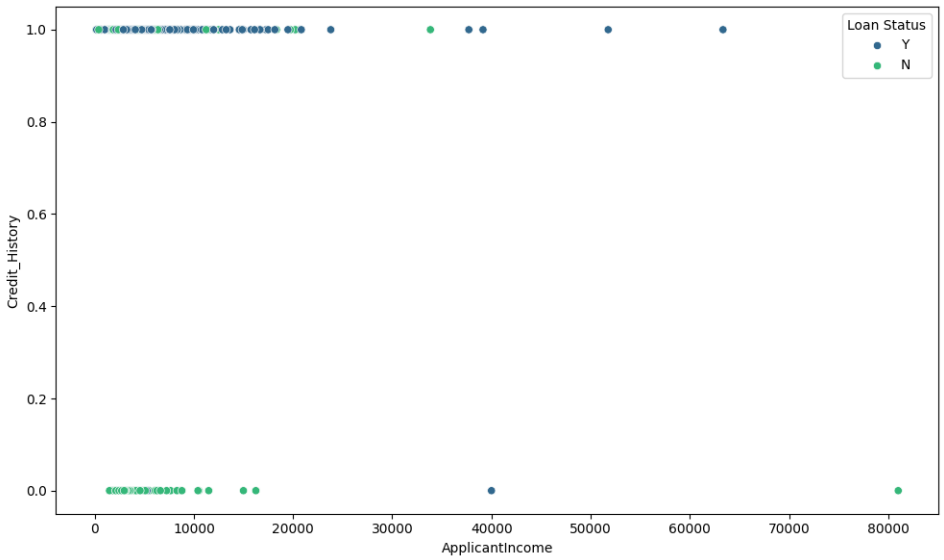


Figure 3.16: Scatter Plot: Applicant Income vs Credit History

"ApplicantIncome" and "Credit_History" relationship in predicting loan eligibility as visualized by Figure 3.16 on the x and y-axis. Loan Approval Status is represented by color coding. We observe that applicants with a Credit History equal to 1 (good credit history) are more likely to have their loans approved (blue dots having high density, in $y = 1$). The more green dots at $y = 0$ means that, applicants with Credit History $= 0$ (no or bad credit history) are more likely to be denied loans. We don't see a strong linear relationship between Applicant Income and loan approval. We see people who got approved and rejected across a very wide range of incomes. Here, Credit_History seems to be an important predictor of approval: there is a very skewed approval distribution with respect to Credit_History $= 1$.

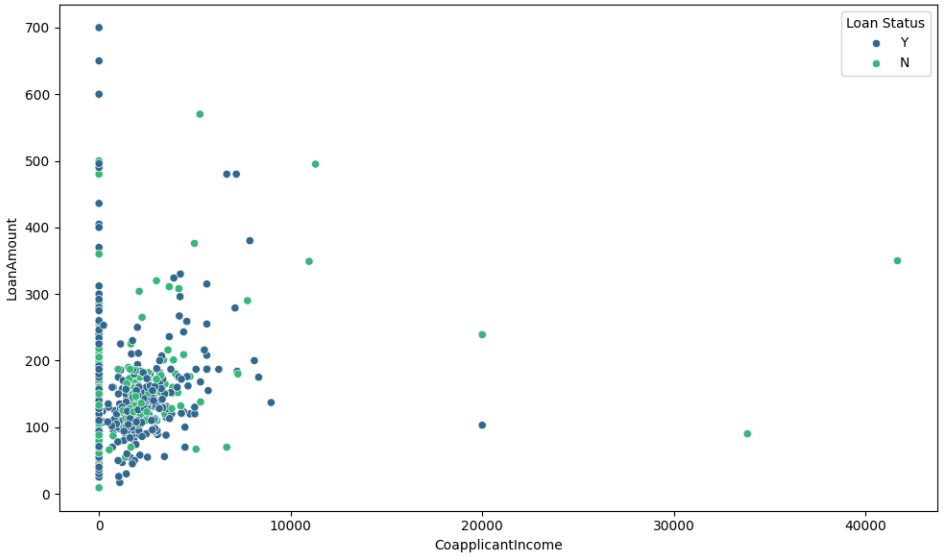


Figure 3.17: Scatter Plot: Coapplicant Income vs Loan Amount

Figure 3.17 represents the relationship between "CoapplicantIncome" (x-axis) and "Loan Amount" (y-axis) against loan status by coloring the data points. We also see many points clustered near Co-applicant Income = 0 which suggests that a large proportion of people did not have a co-applicant, or that the co-applicant had no income. These loans usually have smaller loan amounts and commonly are under 200 (in thousands). At this point, there is loan approval (blue), and loan denial (green), hinting co-applicant income is not the sole criterion to approve or deny a loan. Also, as the Co-applicant Income increases the range of loan amounts increases, which indicates that higher Co-applicant income could also be associated with higher loan amounts. Nevertheless, there are some outliers represented by those with very high values of the project amount (700 thousand) or income (40000), however, which happens infrequently.

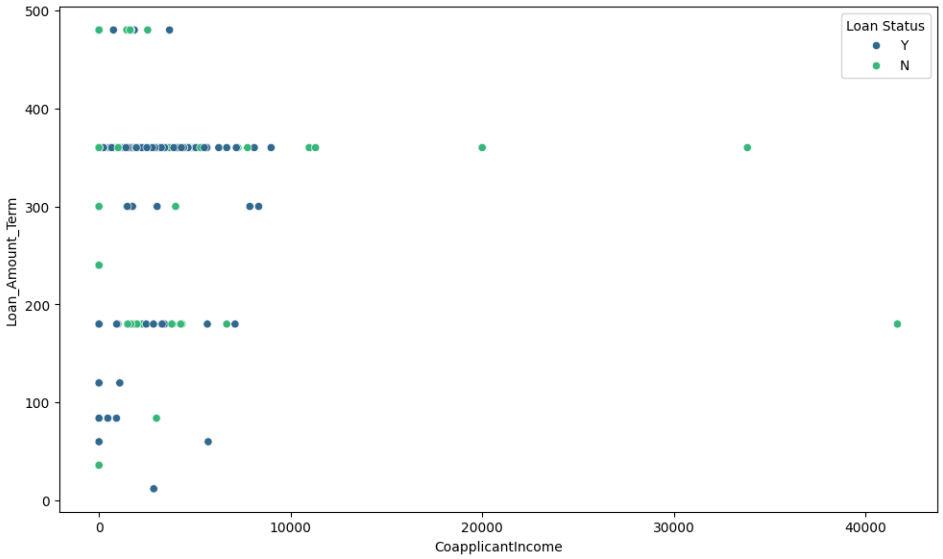


Figure 3.18: Scatter Plot: Coapplicant Income vs Loan_Amount_Term

The Figure 3.18 represent the scatter plot that there is some relationship between "CoapplicantIncome" and "Loan_Amount_Term", where there are some loans concentrated at a couple of terms such as 360 months and 180 months. The loan term does not directly determine loan status, and co-applicant income is not the sole determinant. Loans with terms of 360 months dominate the dataset, reflecting the preference for long-term loans. Higher co-applicant incomes do not guarantee loan approval, but denials appear slightly more frequent at extremely high incomes for long-term loans.

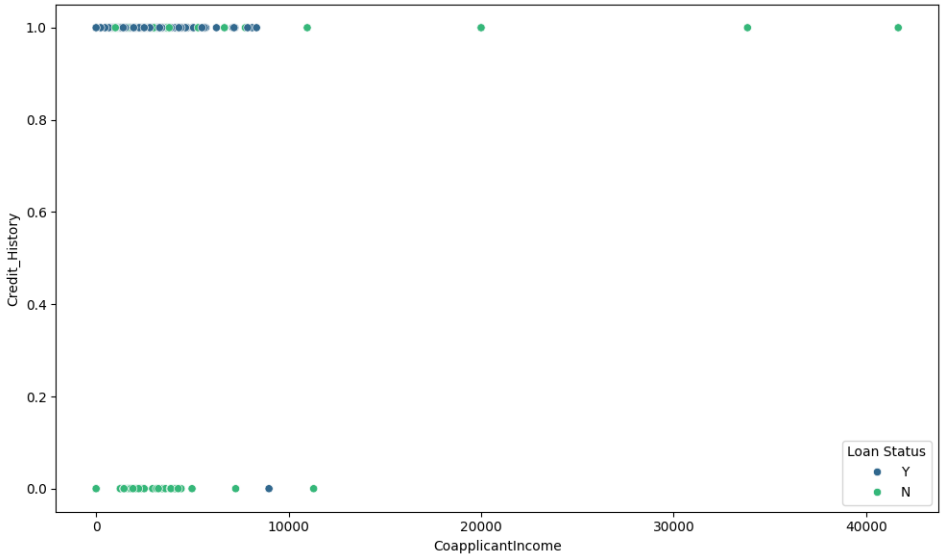


Figure 3.19: Scatter Plot: Co-applicant Income vs Credit History

In this scatter plot(Figure 3.19), "Credit_History" (y-axis) is plotted against "CoapplicantIncome" (x axis) and color-coded by Loan Status. Credit History = 1 (Good Credit History): A significant number of data points are concentrated here, and most of them belong to approved loans (blue dots) instead of poor Credit History. Thus, it is shown that good credit history has a close correlation with loan approval. And bad Credit history has fewer data points, which mainly represent the loan not being approved. Also most data points are clustered near Co-applicant Income 0 to 5000. Both approvals and denials exist in this range, with credit history playing a more decisive role.

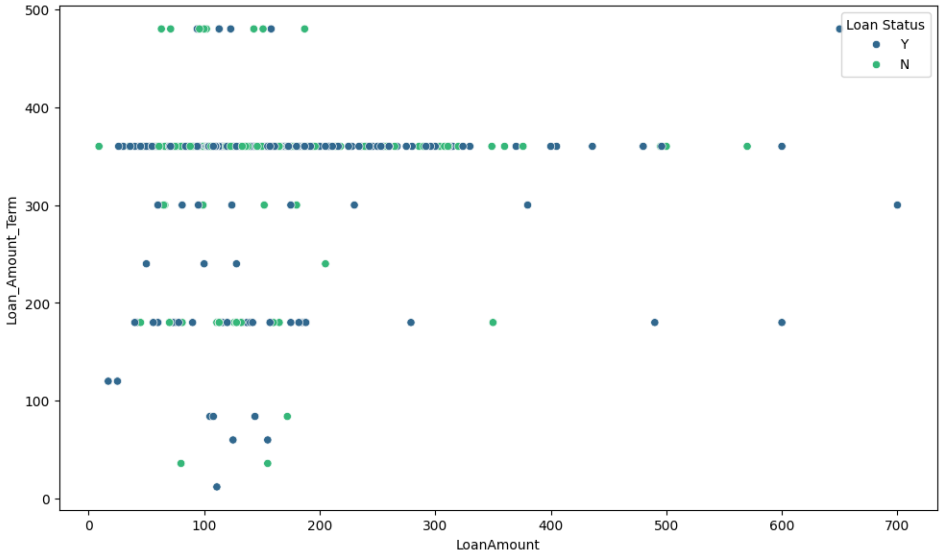


Figure 3.20: Scatter Plot: Loan Amount vs Loan_Amount_Term

Here i present a scatter plot examining the relationship between "LoanAmount" (on the x-axis) and "Loan_Amount_Term" (on the y-axis). The data points are color-coded based on Loan Status, where, Blue points (Y) Represent approved loans and Green points (N): Represent rejected loans. We can see that most of the data points are located in very specific regions meaning that they are often common loan amounts and terms. The most frequent values of Loan Amount Term are standard loan durations: 360 months (30 years) and also 180 months (15 years). The blue dots show that for smaller loan amounts (less than 200), more of them seem to have been approved. But many loans, whether approved or not, focus on low to mid-range loan amounts and standard terms, like 360 months. Alongside those, we also have a few outliers with terms and loan amounts that are high.

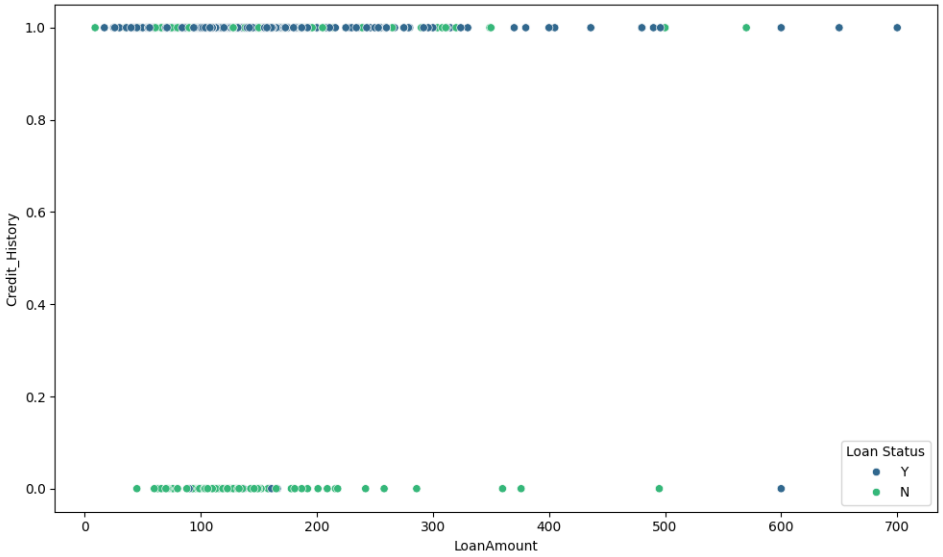


Figure 3.21: Scatter Plot: LoanAmount vs Credit_History

This scatter plot visualizes the relationship between "LoanAmount" and "Credit_History" in a loan dataset. We see a very strong correlation between having a credit history and getting a loan approval. The majority of loans are approved (1.0) for loans with a positive

credit history, irrespective of the loan amount. On the other hand, loans with a negative credit history (0.0) are mostly rejected, and the rate of approving them is very low. Loan amounts vary widely within each of these categories, but the clustering of approved loans is primarily observed at higher values for the credit history. This means that having a good credit history is very important to whether your loan will be approved or not. Some of the loans with poor credit history going through approval suggest outliers there could be exceptions or some other influence play.

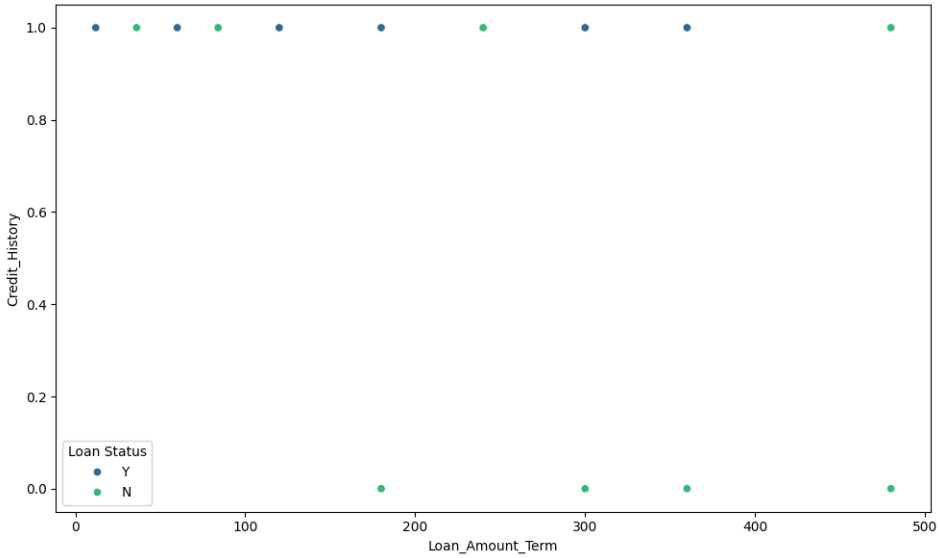


Figure 3.22: Scatter Plot: Loan_Amount_Term vs Credit_History

Figure 3.22, scatter plot represents the relationship between "Loan_Amount_Term" and "Credit_History", and their effect on Loan Status (Y/N). The x-axis represents Loan_Amount_Term, which is the duration of the loan in months. The y-axis represents Credit_History, where, 1 means the applicant has a good credit history and 0 means the applicant has no or poor credit history. In general, blue dots dominate the area of applicants with a credit history of type 1 (good credit) - which translates to better chances of getting a loan approved. Applicants with no or poor credit history (Credit_History = 0) are less likely to have their loans approved (green dots dominate this area). It does not seem that there is a good relationship between the duration of the loan term and loan approval, as the points are all over the x-axis.

3.5 Correlation Matrix

Figure 3.23 plots a correlation matrix, which shows the linear relationships among different features in the dataset and the correlation values are in the range from -1 to 1. Red shows strong positive correlations, which means as one variable increases then the other generally also increases, and blue shows strong negative correlations. The numeric value of the correlation between "ApplicantIncome" and "LoanAmount" is 0.57 which is slightly positive and indicates the positive relation or as higher applicant income, the larger loan amount. "CoapplicantIncome" shows a very weak positive correlation with "LoanAmount" (0.19) hence a very lower impact. There are some features like Gender, Married, Dependents, Self_Employed, etc., which cause negligible correlation with other

variables therefore they do not interact and or predict other features in a linear way. we observe that "Loan_Amount_Term" is also weakly correlated with all features, which demonstrates its independence. We note a slight negative correlation between "Education" and "ApplicantIncome"(-0.14) here perhaps indicating that higher incomes are associated with lower levels of formal education in this dataset. The matrix shows, overall, that most features are relatively independent of each other, and they have only a few moderate correlations, which may assist in the avoidance of multicollinearity problems in the predictive modeling. The weak correlations also imply that no one feature on its own will likely overwhelm the explanatory power for outcomes. The results of this analysis help you identify feature interactions, detecting possible multicollinearity, and selecting key predictors, that is, necessary information for model designing and getting to know data structure.

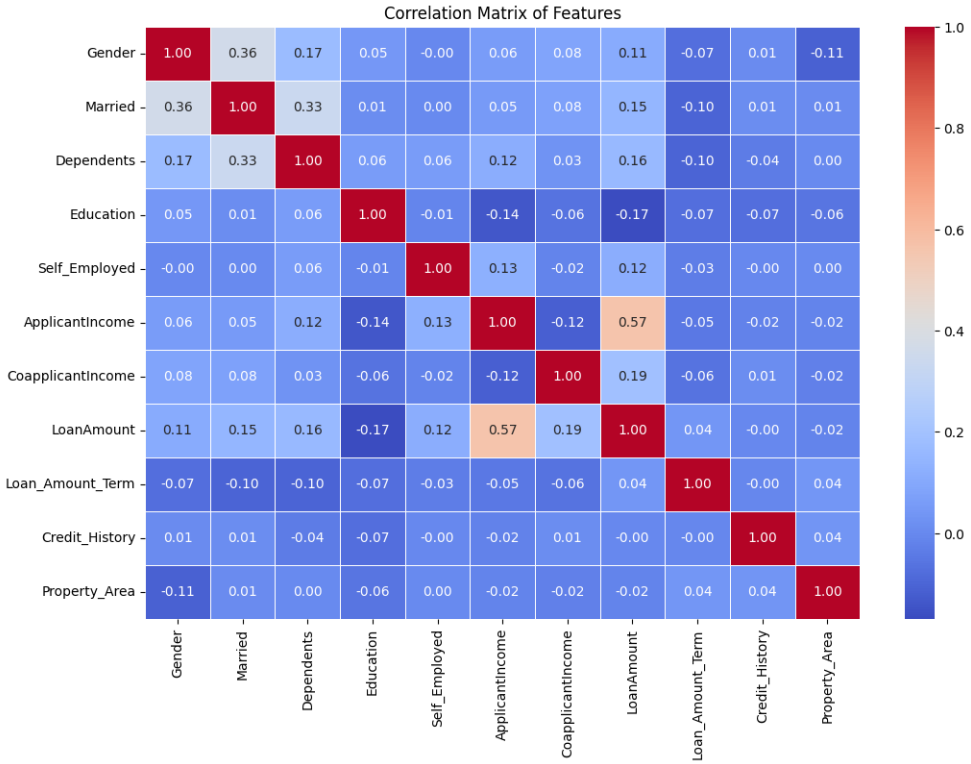


Figure 3.23: Correlation matrix

Chapter 4

METHODOLOGY

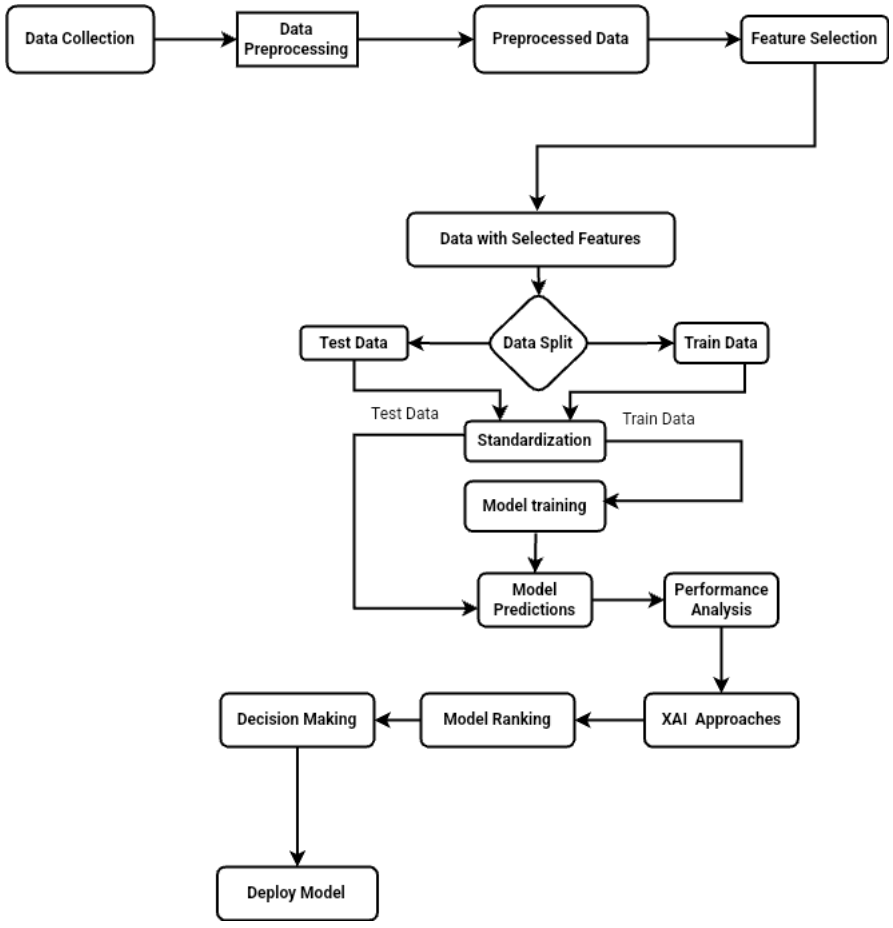


Figure 4.1: Work flow of this Project

In this part, I discuss my methodology. Figure 4.1 we can see the overall proposed system of my project. So, our primary task is data acquisition or sourcing it from a reputable database. After collecting the database we need to analyze the data of the database. Here I mainly used bar plots, histogram plots, scatter plots and confusion matrices for data visualization. Because this plot helps us to understand how data is distributed. After visualization we need to pre-process and then based on this preprocessed data, we need to feature selections. In the dataset, I choose the top most features that impact on results analysis. So for feature selections, I use the MRMR algorithm which stands for

"Minimum Redundancy - Maximum Relevance". This means it is designed to find the smallest relevant subset of features for a given Machine Learning task. So after using the MRMR algorithm, I get 7 features that impact on output. And I collected the selected feature datasets for my model performance analysis. Before model training, we need to split the dataset. So using train test split to split the dataset. So in my selected features train dataset, 80% of the data will be used for training the model and 20% data will be used for testing the model. After that, I used a standard scaler for three features, which are mainly large-scale. Use this scaling to both train and test datasets. Then all combined features are used for training and testing. Then fit the model for prediction. In my project, I use eight machine learning algorithms which are SVM, Decision Tree, KNN, Adaboost, Logistic Regression, MLP, XGBoost and Random Forest. After using this model I generate our results. If the result is satisfactory then I analyze the model performance. For model performance, I use a performance matrix, curve and plotting. And also use the XAI method to explain the feature's importance against accuracy. And then the best model to use for deployment.

4.1 Data Preproccesing

Data preparation means taking raw and unprocessed data and converting it into structured, understandable, and useful data for machine learning modeling. This is a critical step as data in the real world frequently has no key patterns, and also has inconsistency and incomplete information which can prevent the building of an effective model. Basically, it means that good preprocessing ensures that data is correct, meaningful, and well structured to allow for the machine learning model to actually extract good knowledge. Left untreated, missing data can also hurt machine learning models. Imputation, removal, or replacement methods are rolled out in order to avoid missing values and help use the data available with models in the best possible way. Furthermore, the preprocessing, including -feature selection, scaling, and normalization- is suitable enough to combat over-fitting and thus ensures that the model learns real patterns instead of the noise or random information. Data preprocessed is suitable for learning, improves the model's accuracy, and hence results in better predictions.

4.1.1 Data Imputation

Missing values are one of the typical problem with a dataset that is why their removal or imputation is an essential step of data preparation. If not addressed, missing values can cause normally distributed values to shift and skewed the model's result. This is a way of filling gaps left in data for the purpose of making sure that all data collected are reliable.

Imputation Technique: Median

For missing numerical values i used median imputation in my research. In this method, i replace missing values by the median of the observed (non missing) values per feature. Compared to the mean, the median being the middle value of a sorted data sample is very resilient to outliers and is a much more robust measure of central tendency.
Steps for Median Imputation:
Find the features or columns having missing values.

For each affected feature, calculate the median value of observed (not missing) data. It replaces the missing values with the median calculated.

Imputation Technique: Mode

For missing categorical values, i used mode imputation in my research. The Mode imputer replaces missing values on categorical columns with the most frequent value (mode) in that column. A frequent and natural way to deal with missing values in categorical data for which this assumption holds is simply to infer the missing values as the most common category.

4.1.2 Data Encoding: Categorical Variable Transformation

So in the pre-processing stage on 1st, we noticed that in my train data set there are some columns that have numerical data and on the other hand there are some columns which have Categorical Data. In categorical data, there are two types of data in our dataset. Where few columns have Binary Categorical Variables and few columns have multi-class categorical variables.

Label Encoding

Label Encoding converts binary categorical variables with only two unique categories (Yes/No, Male/Female) to numeric labels (0 and 1). Machine learning algorithms can't directly process categorical text data. So, they need to be transformed. In the dataset used for my research, binary categorical columns were identified and converted to numeric values using the LabelEncoder method.

Target encoding

Multi-class categorical variables have more than two distinct categories. These categories, which represent groups or levels of a feature, are transformed using Target encoding. Target encoding is the technique of transforming categorical variables into numerical values using the statistical information of the target variable, often the mean, but it can also be the median or standard deviation in many cases. This encoding takes advantage of the feature and the target variable information together so that it can help machine learning models learn relationships better. Firstly, the data set is grouped by the unique categories of the variable. Next, a statistic (often the mean of the target variable) is calculated over each category. After that, the original categorical values are assigned with their analyzed statistics, while in fact transforming the variable to a numeric one inclusive of correlation with the target.

4.1.3 MRMR

The mRMR algorithm, a feature selection approach, is extremely common to use in Machine learning. It tries to extract a subset of these as being highly relevant to their target variable by reducing the redundancy among the characteristics chosen.

Relevance:
Mrmr mainly used mutual information to calculate the relationship between the feature

and target variable. When these two variables depended on each other then mutual information measured the strength of their relationship. If the value of mutual information is high between Xi and Y, then it means this feature has a strong relationship with the target variable.[12]

$$\text{Relevance}(X_i, Y) = I(X_i, Y) \tag{4.1}$$

Redundancy:
In this part mrmr measure the relationship between the feature, it means it measure how much information share each other X_i and X_j. If the mutual information is high between this two feature then it mean they provide the same information. Then it call high redundancy. So that MRMR ignore that feature which mainly shares the same information. Because it main goal is to reduce overfitting in machine learning algorithms.

$$\text{Redundancy}(X_i, X_j) = I(X_i, X_j) \tag{4.2}$$

Average Redundancy:
Here mainly measures the redundancy within the new feature X_i and the feature S which are already calculated for redundancy. It help to identify whether the feature X_i share any information with the feature of S.

$$\text{Average Redundancy}(X_i, S) = \frac{1}{|S|} \sum_{X_j \in S} \text{Redundancy}(X_i, X_j) \tag{4.3}$$

Finally, MRMR score is calculated based on relevance and Redundancy. Where it measures the high relevant to the target variable. Also, measure the low redundancy among the features.[12]

$$\text{mRMR Score}(X_i) = \text{Relevance}(X_i, Y) - \frac{1}{|S|} \sum_{X_j \in S} \text{Redundancy}(X_i, X_j) \tag{4.4}$$

4.1.4 Standardization

Standard Scaling is also known as standardization. It is a data preprocessing technique used in machine learning to transform numerical features so that they have a mean of 0 and a standard deviation of 1. This puts all the features at an equal footing to contribute to the model without some features dominating because of their larger scales.

4.2 Model Specification

4.2.1 Logistic regression

Logistic regression is a convenient supervised machine learning algorithm to use for classification problems when we have a binary result (0 or 1). Logistic Regression is mainly a statistical technique used to estimate the likelihood of a target falling within a particular category given a set of predictors (independent variables). In this post, we will look at Logistic Regression (LR), the most common kind of regression as a supervised machine learning approach. LR predicts the likelihood of a binary result by using a sigmoid function to a linear combination of input data. As the output probability value is from 0 to 1 it better suits with classification problem.

The logistic function is defined as:

$$P(Y|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}} \tag{4.5}$$

where β_0 and β_1 are parameters learned during training, and X represents the input features.

4.2.2 SVM

SVM is a powerful machine learning tool when speaking about classification and regression. It is good for complex high dimensional datasets. The support vector machine approach attempt to identify a perfect hyperplane that separates the data into one of the different groups. When choosing the hyperplane, we try to maximize the margin. Which is mainly the distance between the hyperplane as well as the closest data points of every class. Kernel functions allow SVM to deal with data that is separable in a linear or non linear manner. The method may apply kernel functions in order to transform the original feature space to a higher dimensional space, in order for the classes to be linearly separable.[7]
The SVM formula is given as:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \frac{1}{\|\mathbf{w}\|_2} \left[\min_n y_n \left(\mathbf{w}^T \phi(\mathbf{x}) + b \right) \right] \tag{4.6}$$

Where:

$$\min_n y_n \left(\mathbf{w}^T \phi(\mathbf{x}) + b \right)$$

stands for the shortest margin between a data point and decision border.

$$\arg\max_{\mathbf{w}}$$

mainly stands for the highest point in the domain of a function

4.2.3 Decision Tree

A Decision Tree is a supervised learning method that underutilized for solving both classification and regression problems but is more preferred in classification problems. Indeed, it is a tree-based model, where the tree nodes represent features of the given dataset, decision branches are defined by the tree branches, and the decision outcomes are defined at the leaf nodes. The model comprises two kinds of nodes: which are called one is a Decision Node and the other is a Leaf Node. Decision Nodes decide the path for decision trees and have more than one branch while the Leaf Node are the final results options of decision trees that cannot branch any further. Based on the specific attributes of the data set, the decisions or the tests within the tree are taken. A Decision Tree is a tool that is used to model all possible options to decide the best solution of a particular problem based on the conditions provided. It is given a brand name and called a "tree" because it has roots that give birth to other branches that spread all over in a tree like manner. For future class formation of the dataset, the first step starts at the root node and checks the root attribute with the attributes of the dataset. Accordingly, if this is

the case, it moves to the next branch of a tree and then continue with the same process. This process goes on right to the end where the algorithm attains a terminal node which displays the forecast. The Decision Tree algorithm is a decision-making technique that uses the feature or attributes of a data set in making the best decision. Its main objective is to gather as much information as possible, which influences selecting features of splitting nodes. The algorithm first chooses the feature which provides the maximum information gain and uses that as the first split.[7] The formula for information gain is expressed as:

$$IG(T, a) = H(T) - H(T \mid a) \tag{4.7}$$

Where $H(T \mid a)$ represents the conditional entropy of T, given attribute a.

4.2.4 K-nearest neighbors

KNN algorithm is one kind of supervised machine learning method which is mainly applied to resolve classification as well as regression problem. It is among the most used machine learning algorithms because it is simple also easy to implementations. No assumptions on the underlying data distribution are required to work with it. It is also able to process both numerical and categorical features. This is a nonparametric method, that predicts a new point based on the similarity of the point in your given dataset. Other algorithms however are more sensitive to outliers and KNN is not. KNN approach uses a distance metric and it is called the Euclidean distance, which computes the K number of closest neighbors of a given data point. The data point assigned the class or value which corresponds to the majority vote or average value of the closest K neighbors. It can adjust to different patterns and predict based on the local structure of the data with this approach.
The first thing is to determine the number of neighbors. In step two you will calculate the Euclidean distance between the k neighbors. The third step is to select the K neighbors according to the estimated Euclidean distance when it's the closest to you. In these K neighbors, count the number of data points that fall into each category. And this is the fourth step. The fifth step is to assign the newly acquired data points to the category that has the biggest number of neighbors. Tuples are the focus of the KNN classifier. Specifically, it involves taking into account k training tuples that are located in the pattern space that are the closest to the unknown tuple under consideration. In other words, these k-training tuples are literally the K nearest neighbors of the unknown tuple. [3] $X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ and $X_2 = (x_{21}, x_{22}, \ldots, x_{2n})$ are examples of elements that make up a Euclidean distancecite. This distance is also known as the distance between two tuples. And this Euclidean distance formula may be used to determine the distance between two tuples, as follows:

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \tag{4.8}$$

4.2.5 AdaBoost

AdaBoost (Adaptive Boosting), an ensemble learning technique that works best for classification problems, aggregates multiple weak classifiers such as decision stumps to form a powerful classifier. This is done in a round-robin process where at the beginning each

datum is given equal weight, and it trains what can be referred to as weak learners on the weighted dataset. The weak learner computes the weighted error rare, assigns a weight to the performance, and adjusts the weights assigned to the misclassified samples towards the harder ones. Thus combination of all such weak classifiers makes the final strong classifier, through weighted voting.[15] Adaboost is easy to use, improves accuracy, and doesn't need feature scaling. However, it is sensitive to outliers and is highly time-consuming on large datasets. Some of its main uses include face recognition, filtering of spam emails, diagnosing disease, and credit rating.

Initial Weight Assignment:

$$w_i^{(i)} = \frac{1}{n}, \quad \forall i = 1, \ldots, n \tag{4.9}$$

here, $w_i^{(i)}$ represents the initial weight for each sample in the dataset. One the other hand n the entire amount of instances which are mainly used for training.

Updating weights

$$w_i^{(t+1)} = w_i^{(t)} \cdot \exp\left(\alpha_t \cdot \mathbf{1}\left\{h_t(x_i) \neq y_i\right\}\right) \tag{4.10}$$

- Here, α_t is the contribution of weak learner h_t at iteration t, which is computed as: $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ with ϵ_t representing the error rate of h_t weighted by the sample weights.
- The function $\mathbf{1}\{h_t(x_i) \neq y_i\}$ serves as an indicator, returning 1 if $h_t(x_i)$ is not equal to y_i, and 0 if the prediction is correct.
- During iteration, iteration t, the weight of the sample x_i is denoted by the symbol $w_i^{(t)}$

Final Classification Rule:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot h_t(x)\right) \tag{4.11}$$

where, $H(x)$ is the combined Strong classifier produced by AdaBoost. Here T represents the overall number of boosting rounds or iterations.

4.2.6 Multilayer Perceptron

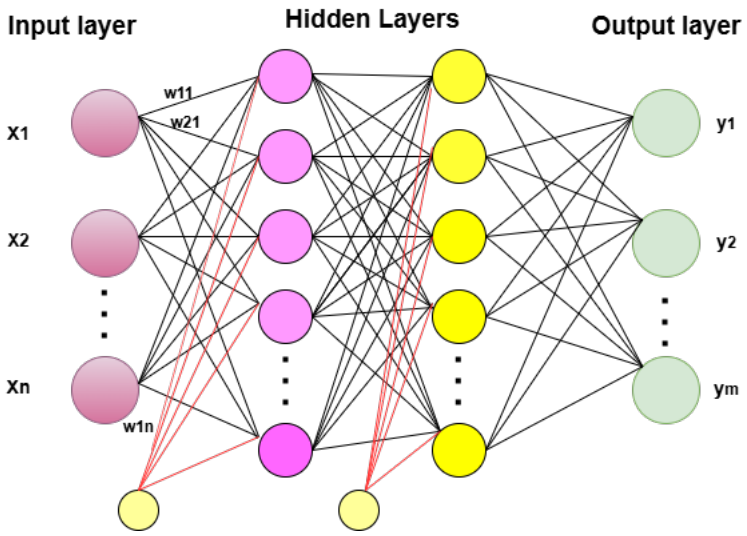


Figure 4.2: Multilayer Perceptron (MLP) Algorithm

A Multilayer Perceptron (MLP) is a type of feedforward neural network widely used in supervised learning for both classification as well as regression tasks. It consists of an input layer, one or more hidden layers and an output layer. An MLP with two hidden layers-an input layer and an output layer is shown in Figure 4.2. Every node exhibits some bias in its interactions. The input layer consists of n variables X, written as $\{x_1, x_2, \ldots, x_n\}$. Also the output layer consists of m variables Y, written as $\{y_1, y_2, \ldots, y_m\}$. It is possible to determine how many parameters make up an MLP.[8]

$$n \cdot h_1 + \sum_{k=1}^{N_h - 1} h_k \cdot h_{k+1} + h_{N_h} \cdot n \tag{4.12}$$

4.2.7 XGBoost

XGBoost's robust functionality and remarkable flexibility have garnered it widespread acclaim in several industries, including healthcare, marketing, and finance. Its ability to deal with missing information, use of parallel and distributed computation, and built-in regularization to prevent over fitting are some of its unique characteristics. Furthermore, XGBoost includes a set of hyper parameters that may be adjusted by a practitioner to get the best possible model performance.
One of the main things about XGBoost is that we have a regularization strategy which helps us avoid overfitting the dataset. Regularization adds in a penalty term to the loss function to make sure that it doesn't get too focused on the training data and to lose its ability to generalize to new inputs. To improve the performance of XGBoost, a tree pruning technique is applied to reduce the tree structure. The XGBoost objective function is described as:

$$L(\theta) = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_k) \tag{4.13}$$

where:

- $L(\theta)$ represents the entire target function.
- $L\left(y_i, \hat{y}_i^{(t)}\right)$ represents the loss function, which mainly quantifies the disparity within the actual value y_i as well as the anticipated value $\hat{y}_i^{(t)}$ at iteration t.
- $\Omega(f_k)$ is the word that is used to regularize the k-th tree f_k.
- θ is the parameters of the model.

An additional definition of the regularization term $\Omega(f_k)$ is:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{4.14}$$

where:

- γT charges the tree complexity with respect to its leaves with T leaves.
- $\frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$, It aids in reducing overfitting by penalizing the leaf weights w_j,

XGBoost combines these elements with the goal to create a model that will have good training performance and good adaptability to new unseen data. XGBoost seems to work well for stock market prediction because the regularization term does play an important role as to not overfit.[15]

4.2.8 Random Forest

The ensemble learning technique Random Forest is useful for problems which involve classification or regression. Instead, multiple decision trees are built during training time, since this helps increase accuracy and reduce over fitting. It is mainly ensemble of decision trees. Firstly it randomly choose samples in using provided data or training set. This algorithm will construct decision tree with each set of training data. At the end, the outcome of the vote will be decided by averaging the decision tree. For the last, final forecast, pick the one with the most votes.

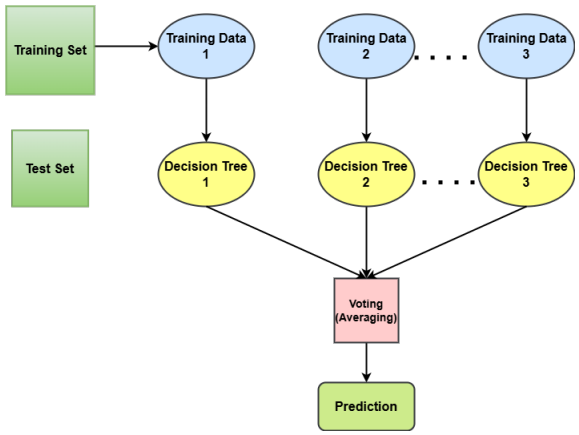


Figure 4.3: Random Forest Algorithm

4.3 Explainable AI

4.3.1 SHAP

Among machine learning interpretation techniques SHAP (SHapley Additive ExPlanations) has gained significant recognition. Prior to producing a prediction SHAP evaluates feature participation levels then relays these relevance assessments to explain specific outcomes. SHAP uses cooperative game theory through Shapley values to evaluate how much each player adds when they participate in collaborative gameplay. SHAP provides interpretation capabilities to a wide range of linear, tree-based along with deep learning mathematical models. The method creates an adaptable interpretability solution that works with any modeling approach[2].
The SHAP value formula for a feature i, derived for machine learning explainability, is as follows:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} \left[f_x(z') - f_x(z' \setminus \{i\}) \right] \tag{4.15}$$

Where

- $\phi_i(f, x)$: The SHAP value for feature i, quantifying its contribution to the prediction of the model f for input instance x.
- $z' \subseteq x'$: All possible subsets of the simplified input features x'.
- M: The total number of simplified input features.
- $f_x(z')$: The model's output when conditioned on the subset z' of features.
- $f_x(z' \setminus \{i\})$: The model's output when the subset z' excludes feature i.
- $\frac{|z'|!(M-|z'|-1)!}{M!}$: A combinatorial weight ensuring fairness in feature attribution.

Chapter 5

Result analysis

5.1 Performance Evaluation Metrics

If the projected class value is the same as the actual class value, and both say yes, then we say it more specifically a true positive (TP). As an example, what we term "True Positives" (TP) happens when both the actual and projected classes point to the same conclusion: It shows that this applicant is qualified for a loan. In the case of True Negatives (TN), the actual and projected values of the class are negative. For example, suppose the actual class is that the applicant does not qualify for a loan and the projected class also matches with this class. Our actual class will be different from our anticipated class, and we will get false positives and false negatives. FP is the name given to the situation where the anticipated class is yes, while the actual class is no. First, let us take the case when the actual class says the Applicant does not qualify for this loan but the anticipated class does. The term used for the actual class is positive and the projected class is negative is called False Negatives (FN). This could happen, if the applicant's actual class value denotes him eligible for the loan; the predicted class value denotes him not eligible for the loan. After understanding these four attributes, they can be calculated for Accuracy, Precision, Recall, and F1 score.[3]
When dealing with machine learning and classification tasks there are numerous important measures to be used when evaluating the performance of a model. Among them are accuracy, precision, recall, and F1 score, etc. The definitions as well as explanations for each are provided in the following:
1. Accuracy For classification tasks accuracy is the most common metric. It tells us how much the model is correct overall, how often the model makes correct predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

2. Precision A precision metric estimates how much of the instances predicted as positives are actually positives. It showcases accuracy of the positive predictions of the model.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.2}$$

3. Recall (Sensitivity) Recall, which is sometimes referred to as sensitivity, is a measurement that determines the percentage of real positive cases by which the model accurately recognized them. Its primary objective is to find every positive situations that are

relevant.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.3}$$

4.F1 Score

The harmonic mean of the accuracy and recall scores is what makes up the F1 Score. When we need to strike a balance between accuracy and recall, as well as when there are imbalances in the data, it is helpful to have this tool. An F1 score may vary anywhere from 0 (the poorest) to 1 (the greatest).

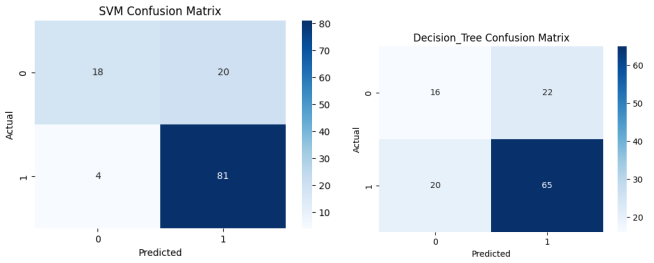
$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.4}$$

5.2 Classifiers Performance Analysis

I measured machine learning model performance through Accuracy, Precision, Recall and F1-Score metrics. In my research, I used eight machine-learning models. which are SVM, Decision Tree, KNN, AdaBoost, Logistic Regression, MLP, XG-Boost and Random Forest. Table 5.1 shows output values for True Negatives (TN), False Positives (FP) and False Negatives(FN), True Positives (TP) derived from confusion matrices. Based on this confusion matrix, I calculated Accuracy, Precision, Recall, and F1-Score for all model. Which is in Table 5.2. And if we noticed then see that, through comparison of different Machine Learning models, Logistic Regression (LGR) was the top performing model with the highest accuracy of 0.83 and F1 score of 0.89 and excellent recall of 0.98 which shows, it is able to identify almost all positive cases correctly. Other strong-performing models include AdaBoost, Multi-Layer Perceptron (MLP), SVM with balanced metrics and high recall (0.93, 0.94, 0.95) and F1 scores (0.87). We also see that Random Forest and K-Nearest Neighbors (KNN) follow closely, with good recall (0.88-0.91) but lower precision and accuracy compared to the top models. XGBoost performs well, where Accuracy (.80), precision (0.84) and F1 score (0.86); and Decision Tree has the lowest performance across all metrics, specifically accuracy (0.66) and F1 score (0.76). Logistic Regression proved the most robust overall with the best general performance, and is the most fitting model for applications with high accuracy and F1 score.

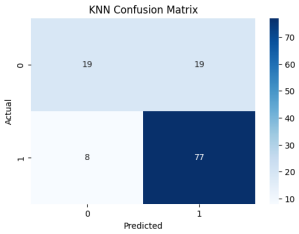
ML Model	**TN**	**FP**	**FN**	**TP**
SVM	18	20	4	81
DT	16	22	20	65
KNN	19	19	8	77
AdaBoost	20	18	6	79
Logistic Regression	19	19	2	83
MLP	19	19	5	80
XGBOOST	23	15	9	76
Random Forest	21	17	10	75

Table 5.1: TN, FP, FN, TP Table for different ML Models

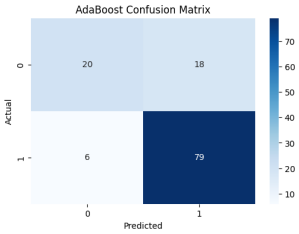


(a) SVM

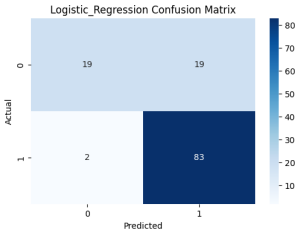
(b) Decision Tree



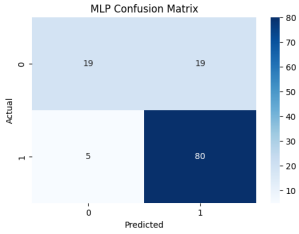
(c) KNN



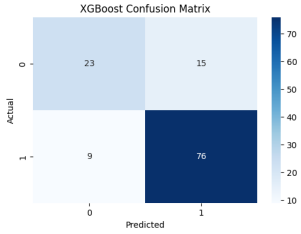
(d) AdaBOOST



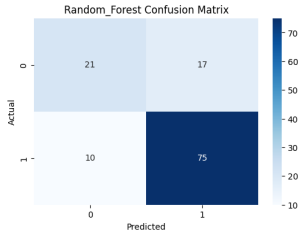
(e) Logistic Regression



(f) MLP



(g) XGBoost



(h) Random Forest

Figure 5.1: Confusion matrix for (a) SVM, (b) DT, (c) KNN, (d) AdaBOOST, (e) Logistic Regression, (f) MLP, (g) XGBoost and (h) Random Forest.

ML Model	**Accuracy**	**Precision**	**Recall**	**F1 SCORE**
SVM	0.80	0.80	0.95	0.87
Decision Tree	0.66	0.75	0.76	0.76
KNN	0.78	0.80	0.91	0.85
AdaBoost	0.80	0.81	0.93	0.87
Logistic Regression	0.83	0.81	0.98	0.89
MLP	0.80	0.81	0.94	0.87
XGBoost	0.80	0.84	0.89	0.86
Random Forest	0.78	0.82	0.88	0.85

Table 5.2: Accuracy, precision, Recall and F1 score comparison table for Different Machine Learning algorithm

5.2.1 ROC Curve

An ROC (Receiver Operating Characteristic) Curve for each model is presented here to evaluate the classification performance. On the Y-axis, plot the proportion of actual positives correctly identified, also is known as sensitivity or recall, TPR = True Positives / (True Positives + False Negatives). False Positive Rate (FPR), on the X-axis, is the proportion of actual negatives incorrectly classified as positives and given by FPR = False Positives / (False Positives + True Negatives). The red dashed line is the performance of a random guessing model. An effective model's curve should, therefore, ideally be above this line. Thus the blue ROC curve defines the trade off between TPR and FPR for different threshold value, a curve nearer to the top left corner means better performance. AUC (Area Under the Curve), ranging from 0 to 1, with 1 meaning perfect model, 0.5-random guessing, and values less than 0.5 meaning worse performance than random guessing.
We see the below ROC curves(Figure 5.2- 5.9) for model performance comparison on different classification models like Support Vector Machines (SVM), Decision Tree, K-Nearest Neighbors (KNN), AdaBoost, Logistic Regression, Multi Layer Perceptron (MLP), XGBoost, Random Forests. For each curve, I plot the True Positive Rate (TPR) on the Y-axis versus False Positive Rate (FPR) on the X-axis, and the choice of the optimal threshold is exhaustively investigated from the trade off between the TPR and FPR by changing the threshold value.
The results of each model's performance are represented as a number, the AUC (Area Under the Curve) values. KNN (0.7910) and SVM (0.7858) responded with higher AUC values: implying better discriminatory power and more ability to discriminate between positive and negative incidents. On the other hand, Decision Tree model (AUC = 0.5929) does only slightly better than a random guess (AUC = 0.5). Logistic Regression, AdaBoost, XGBoost, Random Forest models turn out to perform slightly above average with the scores ranging from 0.74 to 0.76 which seems to be a decent level of classifier effectiveness.

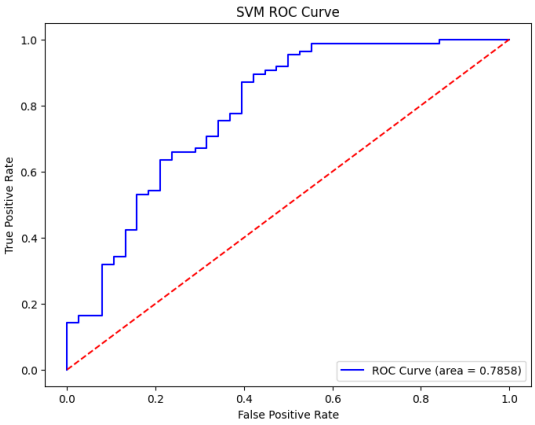


Figure 5.2: ROC Curve for SVM

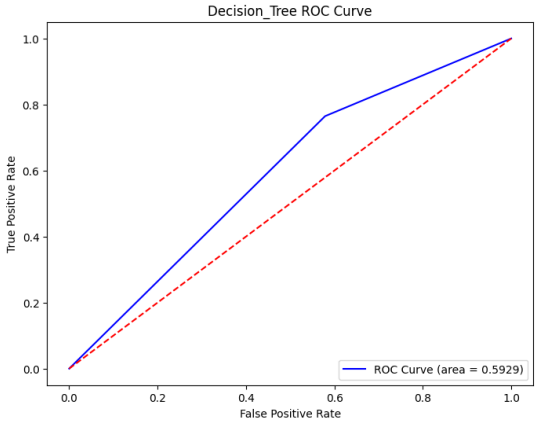


Figure 5.3: ROC Curve for Decision Tree

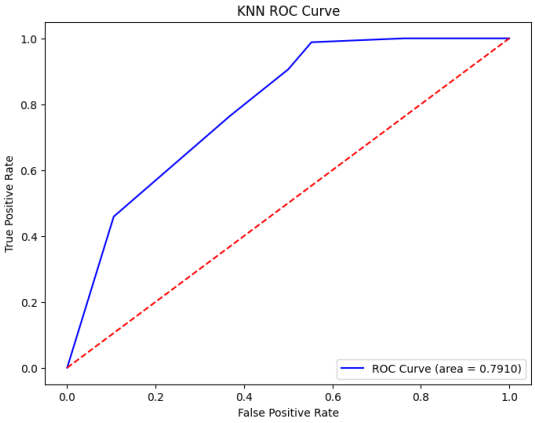


Figure 5.4: ROC Curve for KNN

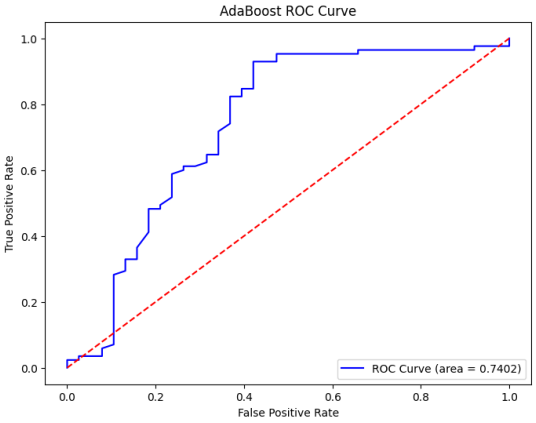


Figure 5.5: ROC Curve for AdaBoost

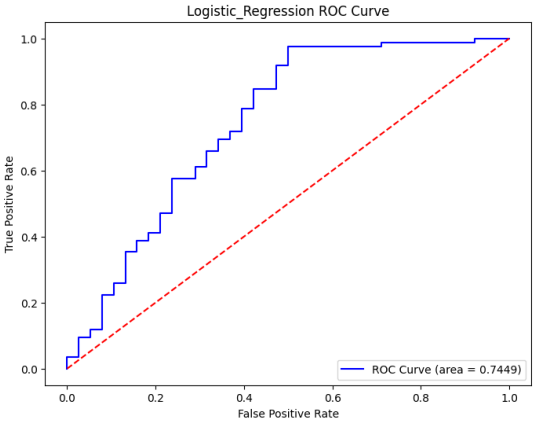


Figure 5.6: ROC Curve for Logistic Regression

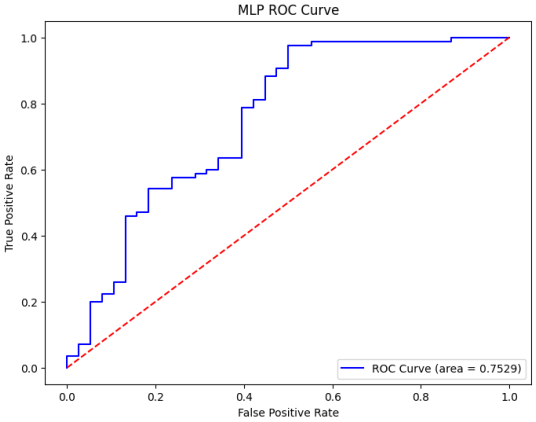


Figure 5.7: ROC Curve for MLP

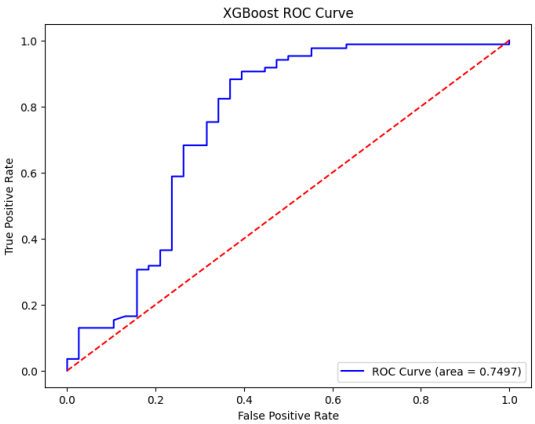


Figure 5.8: ROC Curve for XGBOOST

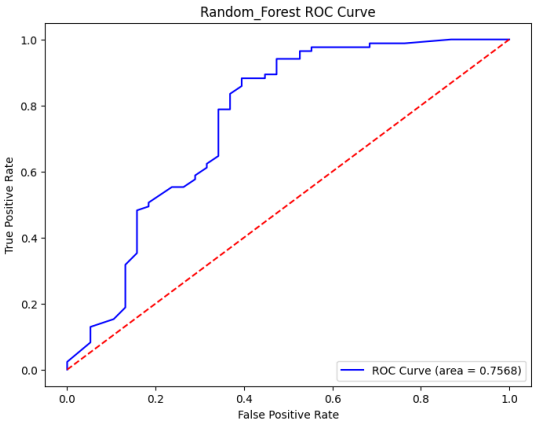


Figure 5.9: ROC Curve for Random Forest

5.2.2 Model Explanation

I used SHAP, which stands for Shapley Additive explanations, in order to provide an explanation for the various models as well as the predictions that were generated by them. In order to do this, i scrutinized the SHAP-provided feature significance for each

model.[9]

Shap

The basics of feature significance are to determine how significant a feature is. To do so, we first compute the average absolute Shapley value for each feature, and plot these in decreasing order. To give a global view of the feature importance of a machine learning model, you take the mean SHAP value plot, that summarizes the contributions of each feature across the whole dataset. They allow us to rank features by the total amount they contribute to the model's prediction. The model output is more sensitive to higher mean SHAP value features, making these features more important. Mean SHAP value plot is a simplified summary of the entire dataset, instead of analyzing SHAP values for each instance (local interpretability).[6]
The support vector-based feature importance graph is shown in Figure 5.10 With the provided statistics, it's easy to see that Credit_History is the most important aspect. The second most important thing is the ApplicantIncome. Next comes the Loan_Amount_Term. In Figure 5.11, we see that Credit_History is the most important feature. Then ApplicantIncome and CoapplicantIncome most important. For KNN(Figure 5.12), Credit_History is the most significant feature, with the highest mean SHAP value, indicating its dominant role in the model's predictions. followed by looked-after Applicant Income and Loan_Amount_Term. For AdaBoost(Figure 5.13), Credit_History is the most important feature and followed by AplicantIncome and Loan_Amount_Term. For Logistic Regression(5.14), Credit_History play the most important role and Coapplicantincome is the second most significant feature. From Figure 5.15 for MLP, we can notice that the Credit_History is the most influential variable according to the highest mean SHAP value which influences the model's prediction. Coapplicantincome and Loan_Amount_Term are both ranks as the second most significant feature, although its impact is much smaller than Credit_History. A SHAP summary plot in Figure 5.16 for the XGBoost model shows that Credit_History has the greatest mean SHAP value, making it the most influential feature in The model's predictions. ApplicantIncome and CoapplicantIncome follow closely, contributing significantly to the predictions. Meanwhile, other features have smaller SHAP values. Finally, for Random forest (Figure 5.17), Credit_History has the greatest mean SHAP value. ApplicantIncome and CoapplicantIncome is the second and third importance feature.

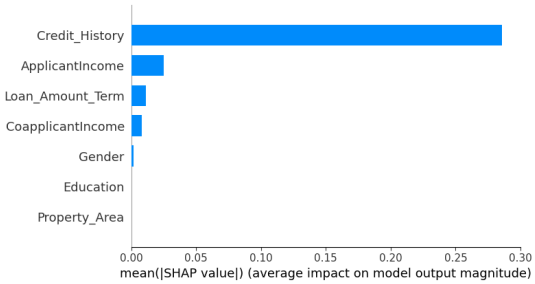


Figure 5.10: Feature importance for SVM

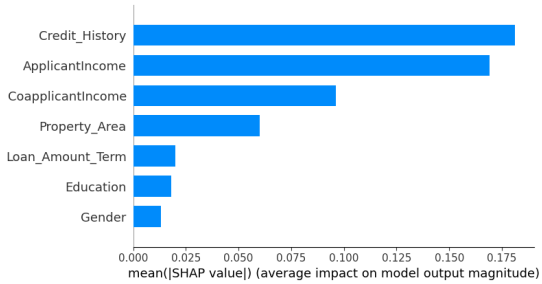


Figure 5.11: Feature importance for Decision Tree

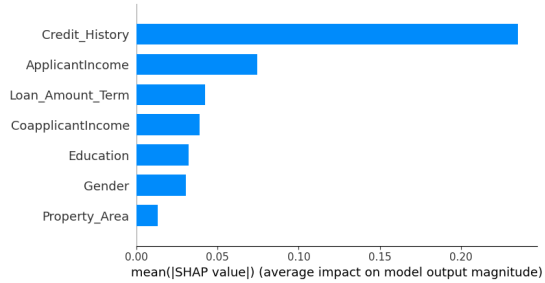


Figure 5.12: Feature importance for k-Nearest Neighbors

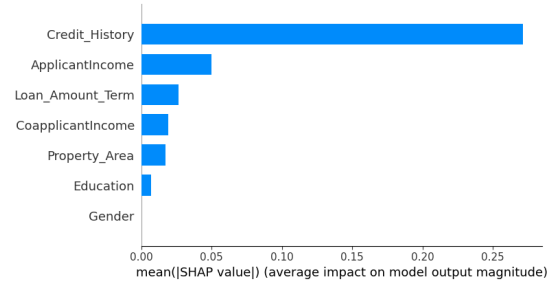


Figure 5.13: Feature importance for AdaBoost

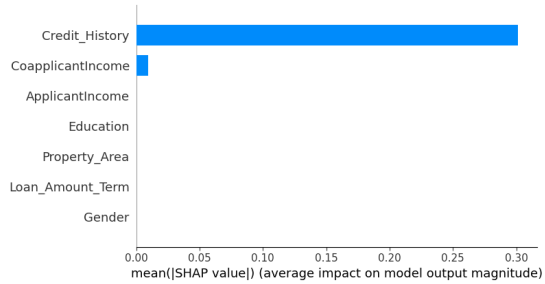


Figure 5.14: Feature importance Logistic Regression

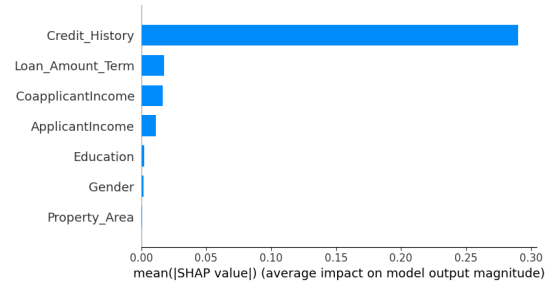


Figure 5.15: Feature importance for Multi-layer Perceptron

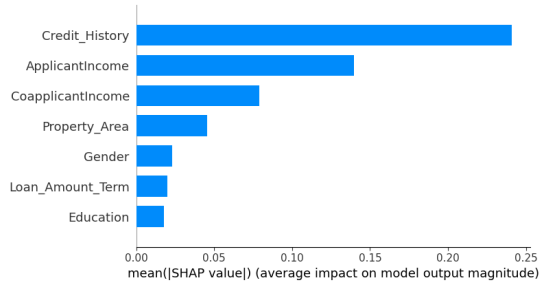


Figure 5.16: Feature importance for XGBoost

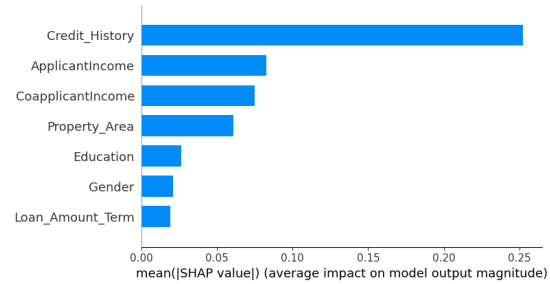


Figure 5.17: Feature importance for Random Forest

Chapter 6

Project Development

6.1 Project Overview

This project centers on designing and implementing an infrastructure that uses machine learning algorithms to forecast loan approvals from financial and demographic user input. The system conducts loan prediction through data analysis of information users give regarding their finances. The development uses Python language combined with machine learning models made deployable through Streamlit framework. SHAP-based feature importance visualizations used here. Used Recommendations generated through the Groq API, as a result user can understand why their loan is approve or reject. Also admin can view and download database.

6.2 Problem Statement

Financial services manage loan evaluation as a cornerstone of the financial expertise in order to effectively allocate credit in a responsible manner. In reality manual evaluation processes involve a lot of manpower as well as they tend to be less transparent and this results in applicant dissatisfaction and inefficiencies. This process can be automated and sped up, and the accuracy and fairness improved, when done through machine learning, using a data-driven approach.

6.3 Requirements

At the 1st stage I analysis the application functionality or required features. The main purpose of this application is to allow users a secure login and input their loan related information. After that provide a prediction with reason and explanation. Also, with an admin dashboard, the admin can view & download user information and prediction results.

Functional Requirements:

- The system requires a user authentication method through a login module based on user and admin roles.
- The system requires input fields to gather user information about their income, credit history, loan term duration, and property area extent.

- An admin dashboard is needed for viewing and analyzing data, including downloading Data.
- Integration of a pre-trained machine learning model (Logistic Regression) for predictions.
- Data-driven explanations of model decisions demonstrated using SHAP-based feature importance visualizations.
- Recommendations generated through the Groq API.

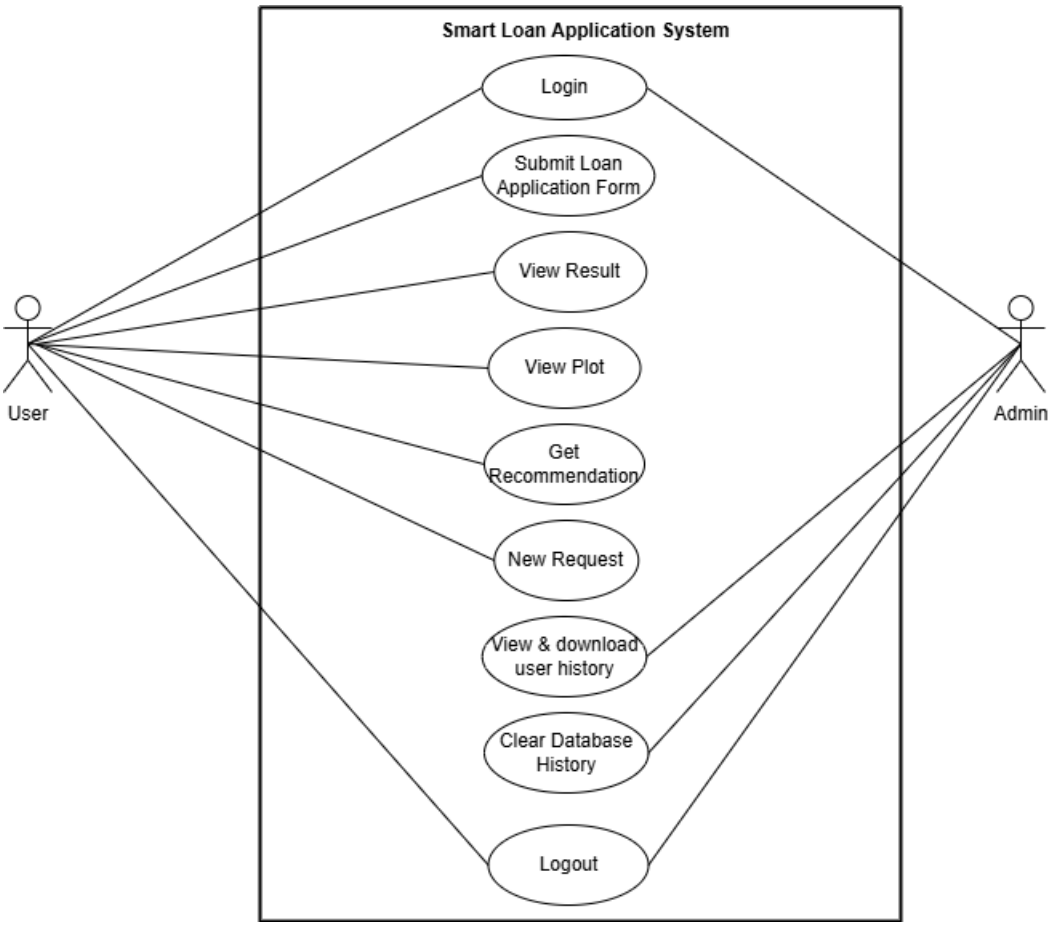


Figure 6.1: Use Case Model

In this Use Case Model there are two actors, the first one is the user and the second one is the admin. Here user and admin have two different dashboard. From this Use Case diagram, we can easily understand the key requirements of the user and admin. Now describe the Use Case Model.

Use Case 1: Login
Actor: User & Admin
Here, the user and admin first need to log in with their user ID and password to enter into their dashboard.

Use Case 2: Submit Loan Application Form
Actor: User

After login, the user can easily view and fill in the loan application form for loan prediction.

Use case 3: View Result
Actor: User
After the submission form, the user can easily view the result of loan approval or rejection.

Use Case 3: View Plot
Actor: User
The user can also see the plot of feature importance.

Use Case 5: Get Recommendation
Actor: User
Here mainly users can see the reason and also get suggestions for future loan approval.

Use Case 6: New Request
Actor: User
If the user wants a new request for prediction, then he can do it.

Use Case 7: View & Download user history
Actor: Admin
In the admin dashboard, the admin can view the user input for loan prediction and also see the result after prediction. Also, the admin can download the user history.

Use Case 7: Clear Database History
Actor: Admin
In the admin dashboard, the admin can also clear the Database if he or she want.

Use Case 8: Logout
Actor: User & Admin
In admin and user can easily leave the dashboard by Logout button

6.4 Design Phase

The design phase focused on building the application's structure:

6.4.1 Front-end Design

I designed an interactive interface using Streamlit which users would find user-friendly. Figure 6.2 demonstrates the front-end workflow diagram.

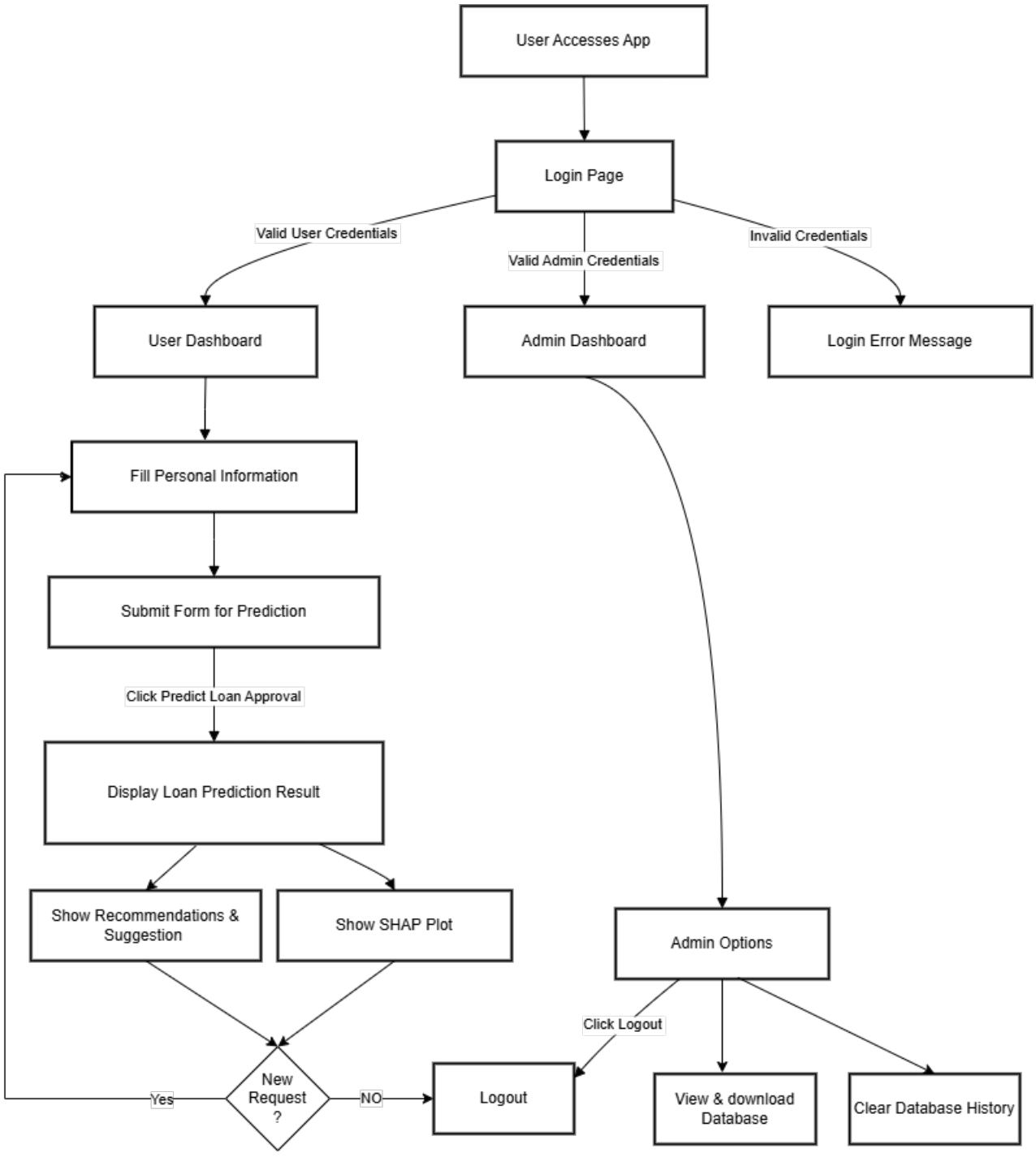


Figure 6.2: Front-end Workflow Diagram

The flow chart in Figure 6.2 defines the front-end workflow for a loan prediction application and shows the steps that a user and admin will take. When the user accesses the app and then he/she enter the login page is the start of the process. If the user has certain credentials, depending on them, he is redirected to the user or admin dashboard. In case of invalid credentials, there is an error message with the invalid credentials message. After Successfully logging, users can input their personal details in their user dashboard, and they can submit their loan applications for prediction where the results together with their SHAP plot and also use can get recommendations and suggestions based on their input and prediction results. As a result, the user can easily understand why their loan is approved or rejected. In the user dashboard, there is a new request button for new loan prediction if they want. On the other hand, admins get a dedicated dashboard to view and download the SQLite database for further investigation. Also, have a clear database option for admin. If they want, they can easily clear the database. The logout option is secure enough for both users and admins to terminate their session. By following this workflow, various roles are able to seamlessly experience it while also retaining the functionality for prediction and data management.

6.4.2 Backend Design:

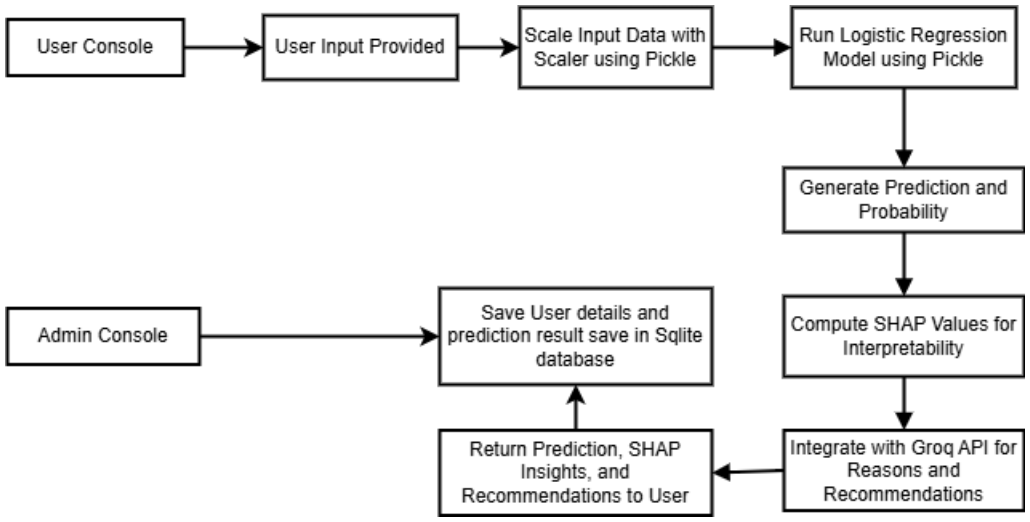


Figure 6.3: Backend Workflow Diagram

- Here I used Python's pickle module to load a Logistic Regression model and data scaler for scaling data for the model.
- providing users with visual insights, I used XAI method SHAP.
- For generating Reason and recommendations based on user data, Groq API integration was designed.
- To view and download the database, here use SQLite.

6.5 Development Phase

The development phase involved implementing the features using Python in VS Code:

Login System: Simplicity dictates a hardcoded username and password solution was implemented alongside Streamlit's session state management for login authentication.

Input Form: Streamlit widgets (e.g., st.text_input, st.number_input, and st.selectbox) were used to create a form for user input.
Prediction Model: A pre-trained logistic regression model was integrated to predict loan status.

Visualization: SHAP was used to generate bar charts showing which features influenced the decision.
Recommendations: The Groq API(L_Lama-3.3-70b-versatile) was used to provide suggestions, helping users understand how to improve their chances of loan approval. For this, I used the prompt here. This prompt is created based on user input and prediction. Based on this prompt text generator model provide the reasons and suggestions.

Key tools and libraries:

Streamlit for UI and interactivity.
sklearn for loading the logistic regression model and scaling input data.
SHAP for feature importance visualizations.
Groq API for generating recommendations & decision explanations.
matplotlib.pyplot for visualize SHAP Plot in the app.
sqlite3 for managing an SQLite database.

6.5.1 Database Description

Using SQLite database technology, the application stores user loan application information in a database named user_history.db. Such applications flourish with SQLite because it operates as a lightweight self-contained serverless database engine. The following section presents an overview of the database structure along with its operational features.

Database Schema

The database features one distinct table known as user_history. The user_history table functions as a database storage location for user information together with loan application entries and predicted values. The schema of the user_history table is as follows:

Column Name	Data Type	Description
Customer_Name	TEXT	The name of the customer applying for the loan.
Credit_History	INTEGER	Indicates the credit history of the customer
Education	INTEGER	Education level of the customer.
ApplicantIncome	REAL	Monthly income of the applicant (in dollars).
CoapplicantIncome	REAL	Monthly income of the co-applicant (if applicable).
Loan_Amount_Term	INTEGER	The loan repayment term in months.

Property_Area	REAL	Encoded numerical value representing the area type (Urban, Semiurban, Rural).
Gender	INTEGER	Gender of the applicant.
Loan_Amount	REAL	Requested loan amount in thousands of dollars.
Marital_Status	TEXT	Marital status of the applicant (Married or Unmarried).
Dependents	INTEGER	Number of dependents financially dependent on the applicant.
Self_Employment	TEXT	Employment type of the applicant (Yes/No).
Loan_Status	TEXT	The loan approval status predicted by the model (Approved or Declined).
Approval_Probability	REAL	The probability score of loan approval predicted by the model.

Table 6.1: Overview of the Database Schema

Functionality of the Database

The Following operations for primary functions in the database system.

Data Storage
The save_to_sqlite() function stores user loan application data with input fields along with prediction results to the user_history table. The system preserves each entry made by users with permanent storage capabilities for later use.

Data Retrieval
The system provides the admin with visual access to stored data by displaying it through the admin dashboard for making decisions. Through the displayed DataFrame administrators can access loan application history for review purposes.

Data Deletion
Through clear_database() administrators can completely delete records stored in the user_history table. The clear_database() function executes automatically to completely erase stored records which exist in SQLite database and persistent session state.

Export Capability
Under administrator privileges the user_history.db SQLite file can be obtained for backup purposes and external analysis. The st.download_button() function enables administrators to obtain complete database file downloads.

6.6 Testing Phase

The application was thoroughly tested to ensure its functionality and usability.

6.6.1 Functional Testing:

- A successful test confirmed the correct operation of the login system when users and admin log in with valid credentials.
- Confirmed that the input fields handle data properly as well as being user-friendly.
- Additionally, I validated that the forecast values are consistent with the Logistic Regression model.
- Verified that the admin can download the database.

6.6.2 Usability Testing:

- Confirmed that the user can easily navigate the interface and also they can understand easily when they use this application. Also, the admin can easily view the database and download it. After that they can log out of their session if they want.
- Also Ensured that the explanation and visualization are clear and understandable for the user. They can easily understand why their loan is rejected and after rejecting what they need to do.

6.6.3 Error Handling:

Also checked error handling for the application. Like.

- Checked that the application handles the invalid input properly. like if a user misses the fill any field then the application handles it properly.
- Also verified for error messages. If the user logs in not properly then it gives an error message, also for API failures.

6.7 Deployment

The application was developed and tested locally using Streamlit in VS Code. For deployment, it is hosted on Streamlit Cloud using GitHub. Future plans may include expanding to other cloud platforms like AWS for wider accessibility.

6.8 UI or Front-end View

6.8.1 Starting Login Screen

Here is the login process page. The application comprises a login system to ensure the user is authenticated securely and allocated an appropriate set of permissions or capabilities depending on their role (admin or normal user). If the user or admin doesn't provide the correct user password then it shows "Invalid credentials" message. Figure 6.4 shows the login page for both the User and Admin.

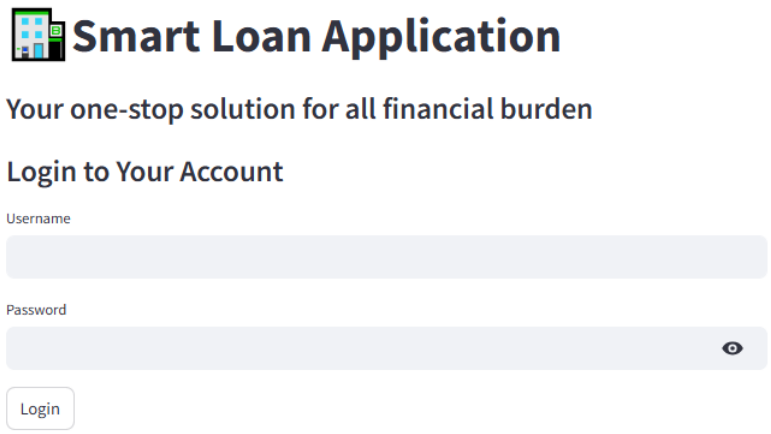


Figure 6.4: Login page for User and Admin

6.8.2 Loan prediction Process for User

If any user wants to predict that he or she is eligible or not eligible for the loan, then at 1st log in as a user. Then after logging in he or she fill-up their personal information which is mainly the application provided. After fill-up then press the prediction button for the result. All the figures show the process of a user step by step and also application shows the result and also explains.

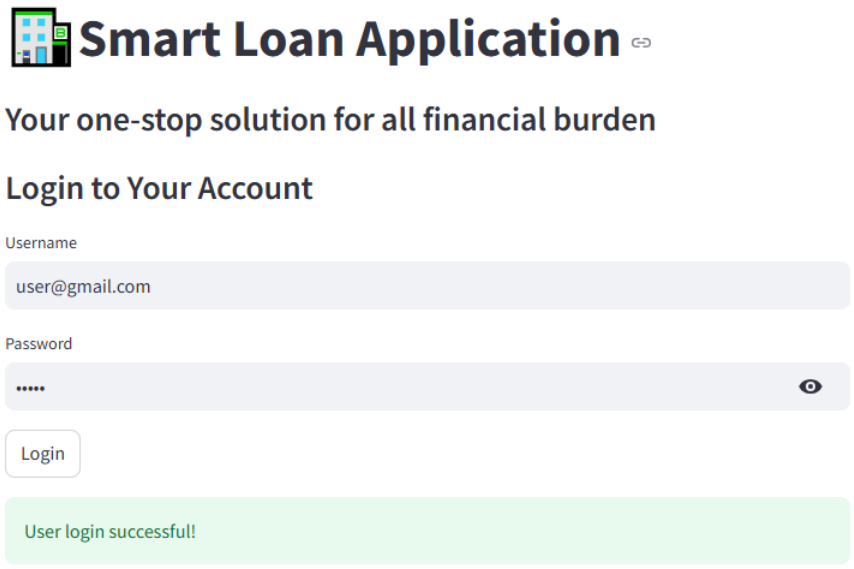


Figure 6.5: Login page for User

After successfully logging in as a user, then the user enters into the next step where the app provides a form for filling up personal information.

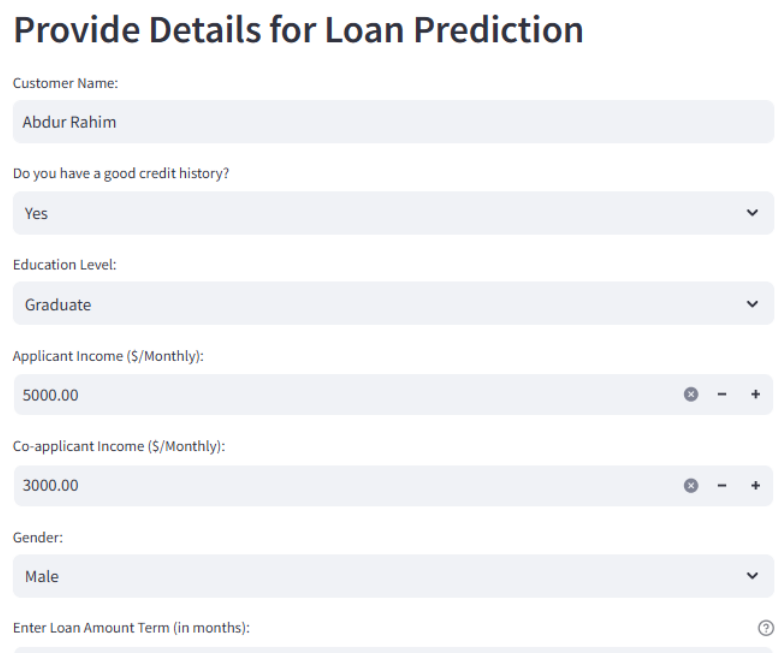


Figure 6.6: Personal information form

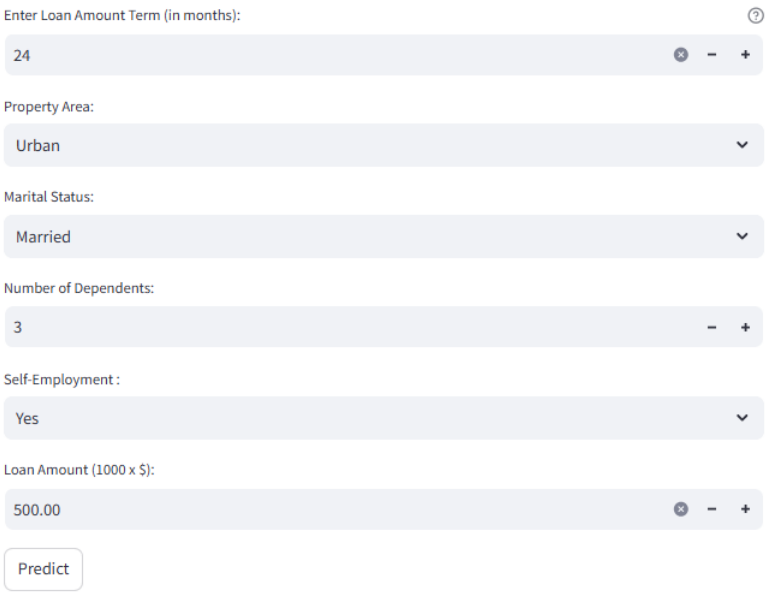


Figure 6.7: Personal information form

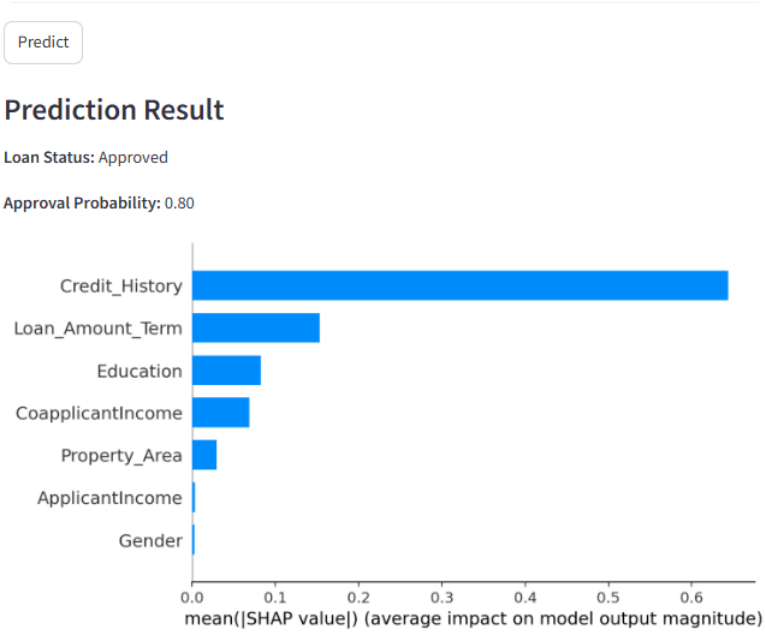


Figure 6.8: Result and SHAP Explanation for Approved

After filling up all the information the user can show the prediction when he or she presses the "Predict" button. In Figure 6.8, we can see the result after prediction. The application also provides a SHAP plot with it.

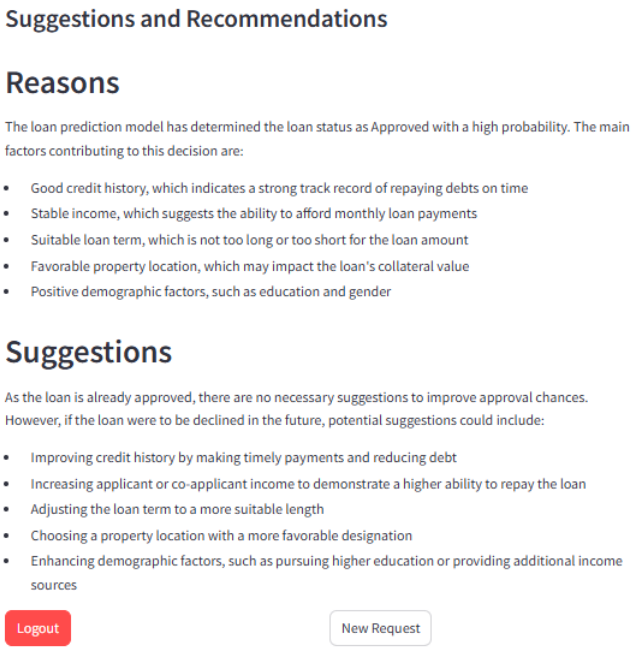


Figure 6.9: Suggestions and Recommendations for Approved

The application also provides reasons and Suggestions for the user. As a result, users can easily understand why they are eligible or not eligible. In Figure 6.9, see Suggestions and Recommendations based on prediction for loan approval.

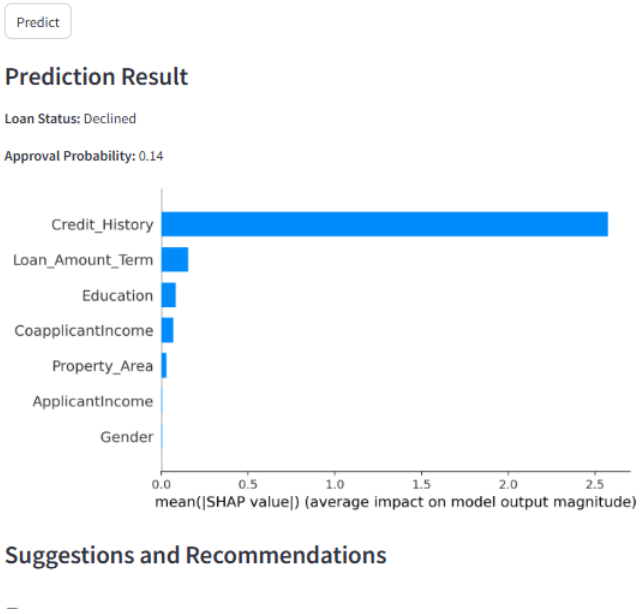


Figure 6.10: SHAP Explanation for Rejection

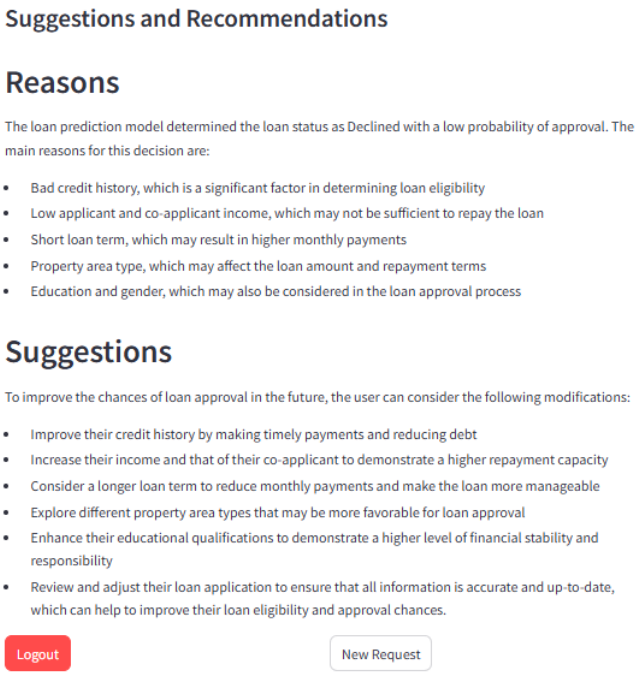


Figure 6.11: Suggestions and Recommendations for Rejection

Also, it's the same for the loan rejection. If the loan is rejected then provide a SHAP explanation for rejection. In Figure 6.10, we can see that it. And also provides a reason why the loan is rejected and provides suggestions for the user, on how he or she can get a loan. Figure 6.11 shows the suggestions and Recommendations. There is also a logout button for exit at any time. And also has a New Request button for the user.

6.8.3 Admin Dashboard

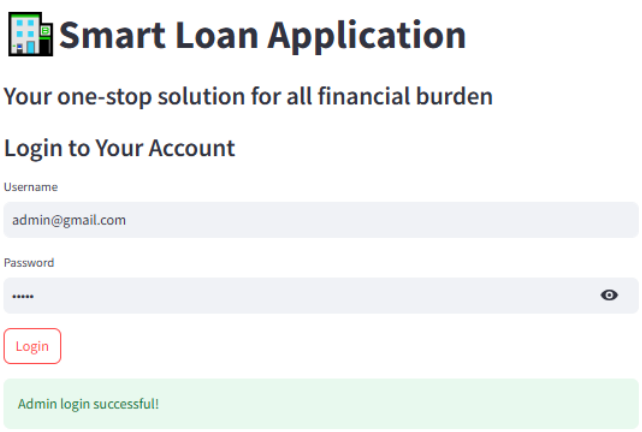


Figure 6.12: Login page for Admin

In Figure 6.12 we can see the login page for Admin. After successfully logging in, the admin can enter into the admin dashboard.

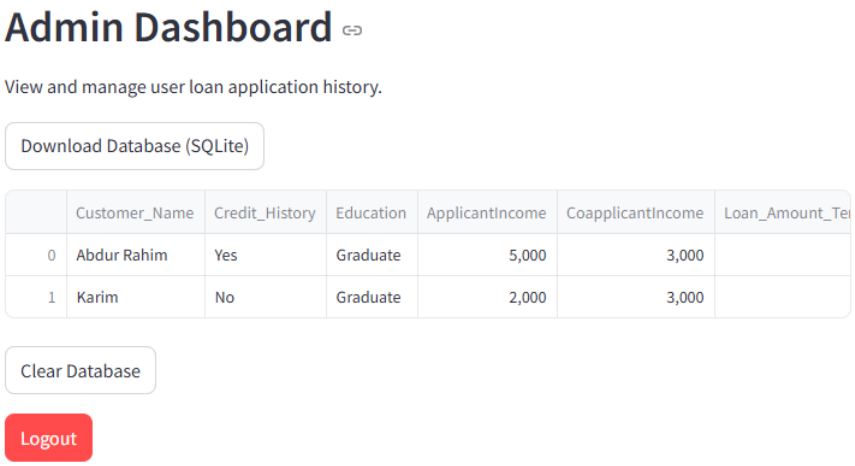


Figure 6.13: Admin Dashboard

In above Figure 6.13 we can see the Admin can view and download the database, where all user information and predictions are stored. Also, there is a Clear Database button for removing all user history from the database if the admin wants.

6.9 Future working

For future work, I would like to make some changes in my application. Like, change static credentials to an authentication method that stores credentials in a database system. The

predictive performance will improve by adding an advanced machine learning model. New input fields in addition to new options will enable more precise and customized prediction results. A better user experience requires improvements in graphical user interface design alongside performance optimization.

Chapter 7

Conclusion

Through loan services, banks accumulate substantial revenue by applying interest rates and creating multiple loan options that support personal finance as well as business development together with economic expansion. Customers depend on loans to carry out big transactions and finance schooling costs and handle unexpected situations and businesses require loans to execute vital transactions and function as a base for economic development. The financial activity creates development via investment promotion and growth expansion that promotes mutual advantages for bank clients and customers. Numerous problems arise from manual loan approval processes which slow the process and introduce inconsistencies and inaccuracies that increase default rates and financial losses. Advanced systems built with data science elements together with machine learning algorithms provide a faster and error-free approach. Research models that study interconnected financial variables predict personal loan repayments while improving banking workflows and lowering operational risk levels for contemporary financial institutions. For this reason, I used eight machine-learning models in my research. Also used SHAP for features importance. After prediction, I found that the Logistic Regression is the best prediction model. After that, I choose this Logistic Regression model for my application. Where i used also SHAP and text generation model L_Lama-3.3-70b-versatile with Groq API to explain the reason for the prediction and give suggestions based on the prediction. As a result, customer can easily understand why their loan is approved or rejected. If their loan is rejected, then what should they do, they can also know from the application. The users interact with the system using an interactive interface, and admins have role-based access to stored predictions. Results show its usefulness to automate the processes of decision-making in financial institutions. In the future, the research could also apply other models that might improve its accuracy, including Gradient Boosting or more sophisticated Neural Networks. Hyperparameter optimizations using Grid Search or Random Search could be developed to fine-tune the results of the applied models. Expansion of feature set inclusion of other external factors-demographic or economic indicators-can make the prediction even stronger. For application, a better user experience necessitates improvements in both graphical user interface design and performance optimization.

Bibliography

[1] J. Tejaswini, T. M. Kavya, R. D. N. Ramya, P. S. Triveni, and V. R. Maddumala, "Accurate loan approval prediction based on machine learning approach," *Journal of Engineering Science*, vol. 11, no. 4, pp. 523–532, 2020.

[2] M. Bugaj, K. Wrobel, and J. Iwaniec, "Model explainability using shap values for lightgbm predictions," in *2021 IEEE XVIIth international conference on the perspective technologies and methods in MEMS design (MEMSTECH)*, IEEE, 2021, pp. 102–106.

[3] M. A. Haque, I. J. Dristy, and M. G. R. Alam, "Symptomatic & non-symptomatic hepatocellular carcinoma prediction using machine learning," in *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, IEEE, 2021, pp. 1–6.

[4] M. Madaan, A. Kumar, C. Keshri, R. Jain, and P. Nagrath, "Loan default prediction using decision trees and random forest: A comparative study," in *IOP conference series: materials science and engineering*, IOP Publishing, vol. 1022, 2021, p. 012 042.

[5] N. Bansode, A. Verma, A. Sharma, and V. Bhole, "Predicting loan approval using ml," *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, vol. 4, no. 05, 2022.

[6] A. K. Dey, A. S. Alam, M. G. R. Alam, S. Zaman, *et al.*, "Implementation of explainable ai in mental health informatics: Suicide data of the united kingdom," in *2022 12th International Conference on Electrical and Computer Engineering (ICECE)*, IEEE, 2022, pp. 457–460.

[7] U. E. Orji, C. H. Ugwuishiwu, J. C. Nguemaleu, and P. N. Ugwuanyi, "Machine learning models for predicting bank loan eligibility," in *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)*, IEEE, 2022, pp. 1–5.

[8] K. Y. Chan, B. Abu-Salih, R. Qaddoura, *et al.*, "Deep neural networks in the cloud: Review, applications, challenges and research directions," *Neurocomputing*, vol. 545, p. 126 327, 2023.

[9] V. Sharma and D. Midhunchakkaravarthy, "Xgboost classification of xai based lime and shap for detecting dementia in young adults," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2023, pp. 1–6. DOI: 10.1109/ICCCNT56998.2023.10307791.

[10] N. Uddin, M. K. U. Ahamed, M. A. Uddin, M. M. Islam, M. A. Talukder, and S. Aryal, "An ensemble machine learning based bank loan approval predictions system with a smart application," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 327–339, 2023.

[11] D. Dansana, S. G. K. Patro, B. K. Mishra, V. Prasad, A. Razak, and A. W. Wodajo, "Analyzing the impact of loan features on bank loan prediction using r andom f orest algorithm," *Engineering Reports*, vol. 6, no. 2, e12707, 2024.

[12] L. C. Jaffrin and J. Visumathi, "Impact of feature selection techniques on machine learning and deep learning techniques for cardiovascular disease prediction-an analysis," *Edelweiss Applied Science and Technology*, vol. 8, no. 5, pp. 1454–1471, 2024.

[13] P. Krishnaraj, S. Rita, and J. Jaiswal, "Comparing machine learning techniques for loan approval prediction," in *Proceedings of the 1st International Conference on Artificial Intelligence, Communication, IoT, Data Engineering and Security, IACIDS 2023, 23-25 November 2023, Lavasa, Pune, India*, 2024.

[14] T. R. Noviandy, G. M. Idroes, and I. Hardi, "Enhancing loan approval decision-making: An interpretable machine learning approach using lightgbm for digital economy development," *Malaysian Journal of Computing (MJOC)*, vol. 9, no. 1, pp. 1734–1745, 2024.

[15] C. Özkurt, "Enhancing financial decision-making: Predictive modeling for personal loan eligibility with gradient boosting, xgboost, and adaboost," *Information Technology in Economics and Business*, vol. 1, no. 1, pp. 7–13, 2024.

[16] C. Perera and S. Premaratne, "An ensemble machine learning approach for forecasting credit risk of loan applications," *WSEAS Transactions on Systems*, vol. 23, pp. 31–46, 2024.

[17] R. A. Zuama, N. Ichsan, A. B. Pohan, M. S. Azis, and M. Lase, "An implementation of machine learning on loan default prediction based on customer behavior," *Jurnal Info Sains: Informatika dan Sains*, vol. 14, no. 01, pp. 157–164, 2024.

[18] G. Chen, "Predicting loan eligibility approval using machine learning algorithms,"

[19] S. G. alias Rohini and S. Mohanavel, "Predicting credit worthiness in bank loan approval process,"

[20] T. T. Yussuph, "Leveraging machine learning algorithm to enable access to credit for small businesses in the united states of america,"