**A Project Report on**

**Loan eligibility prediction using Ensemble Learning**

Submitted in partial fulfillment for award of

**Bachelor of Technology**

Degree

in

**Computer Science and Engineering**

By

**K. Lahari (Y20ACS484) M. Sri Teja (Y20ACS507)**

**K. Kalyani (L21ACS412) N.M.S. Hemanth (Y20ACS519)**

A logo with a picture of a person working on a computer

Description automatically generated

Under the guidance of

**Mr. V Naveen kumar, M. Tech**

Asst. Professor

Department of Computer Science and Engineering

**Bapatla Engineering College**

(Autonomous)

(Affiliated to Acharya Nagarjuna University)

**BAPATLA – 522 102, Andhra Pradesh, INDIA**

**2023-2024**

**Department of**

**Computer Science and Engineering**

A logo with a picture of a person working on a computer

Description automatically generated

**CERTIFICATE**

This is to certify that the project report entitled **Loan eligibility prediction using Ensemble Learning** that is being submitted by K. Lahari (Y20ACS484), M. Sri Teja (Y20ACS507), K. Kalyani (L21ACS412), N.M.S. Hemanth (Y20ACS519) in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

**Signature of the Guide Signature of the HOD**

**Mr. V. Naveen Kumar Dr. M. Rajesh Babu**

**Asst. Professor Associate Professor**

**DECLARATION**

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

K. Lahari (Y20ACS484)

M. Sri Teja (Y20ACS507)

K. Kalyani (L21ACS412)

N.M.S. Hemanth (Y20ACS519)

**Acknowledgement**

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **Mr. V. Naveen Kumar**, Asst. Prof., Department of CSE, for his valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Dr. M. Rajesh Babu**, Associate Professor and Head of the Department for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal **Dr. Nazeer Shaik** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator **Dr. N. Sudhakar,** Prof. Dept. of CSE for his helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

K. Lahari (Y20ACS484)

M. Sri Teja (Y20ACS507)

K. Kalyani (L21ACS412)

N.M.S. Hemanth (Y20ACS519)

**Table of Contents**

[List of Tables viii](#_Toc165048718)

[List of Figures ix](#_Toc165048719)

[Abstract x](#_Toc165048720)

[1 Introduction 1](#_Toc165048721)

[1.1 Need of the study 2](#_Toc165048722)

[1.2 Problem statement 2](#_Toc165048723)

[1.3 Project overview / Specifications 2](#_Toc165048724)

[1.3.1 Objective 3](#_Toc165048725)

[1.3.2 Goal 4](#_Toc165048726)

[1.4 Machine Learning 4](#_Toc165048727)

[1.5 Django Framework 5](#_Toc165048728)

[2 Literature Survey 6](#_Toc165048729)

[3 Requirement Analysis 8](#_Toc165048730)

[3.1 Functional Requirements 8](#_Toc165048731)

[3.2 Non Functional Requirements 11](#_Toc165048732)

[3.3 Hardware Requirements 15](#_Toc165048733)

[3.4 Software Requirements 15](#_Toc165048734)

[4 Algorithms 16](#_Toc165048735)

[4.1 Logistic regression 16](#_Toc165048736)

[4.2 Support Vector Machine 17](#_Toc165048737)

[4.3 Naive Bayes 18](#_Toc165048738)

[4.4 Random Forest 19](#_Toc165048739)

[4.5 Dataset and Description 20](#_Toc165048740)

[5 System Design 22](#_Toc165048741)

[5.1 Existing System 23](#_Toc165048742)

[5.2 Proposed System 24](#_Toc165048743)

[5.3 Use Case Diagram 25](#_Toc165048744)

[5.4 Class Diagram 25](#_Toc165048745)

[5.5 Activity Diagram 26](#_Toc165048746)

[5.6 Sequence Diagram 28](#_Toc165048747)

[5.7 Collaboration Diagram 28](#_Toc165048748)

[6 Implementation 30](#_Toc165048749)

[6.1 Modules and Libraries 30](#_Toc165048750)

[6.1.1 matplotlib.pyplot 31](#_Toc165048751)

[6.1.2 Pandas 31](#_Toc165048752)

[6.1.3 Seaborn 31](#_Toc165048753)

[6.1.4 Numpy 32](#_Toc165048754)

[6.1.5 scikit-learn 32](#_Toc165048755)

[6.1.6 sklearn.svm 32](#_Toc165048756)

[6.1.7 sklearn.naive\_bayes 33](#_Toc165048757)

[6.1.8 sklearn.linear\_model 33](#_Toc165048758)

[6.1.9 sklearn.ensemble 34](#_Toc165048759)

[6.1.10 sklearn.calibration 34](#_Toc165048760)

[6.2 Code URL 36](#_Toc165048761)

[7 Evaluation 37](#_Toc165048762)

[7.1 Metrics for Evaluation 37](#_Toc165048763)

[8 Results 38](#_Toc165048764)

[8.1 Output 38](#_Toc165048765)

[9 Future Work 43](#_Toc165048766)

[10 Bibliography 45](#_Toc165048767)

**List of Tables**

[Table 4.1 Dataset and Description 20](#_Toc164751390)

**List of Figures**

[Figure 5.1 Loan Prediction Architecture 22](#_Toc164751345)

[Figure 5.2 Architecture Diagram 23](#_Toc164751346)

[Figure 5.3 Class Diagram 26](#_Toc164751347)

[Figure 5.4 Activity Diagram 27](#_Toc164751348)

[Figure 5.5 Sequence Diagram 28](#_Toc164751349)

[Figure 5.6 Collaboration Diagram 29](#_Toc164751350)

[Figure 6.1 Calibration curve 35](#_Toc164751351)

[Figure 6.2 Correlation matrix 36](#_Toc164751352)

[Figure 8.1 Interface 1 38](#_Toc164751353)

[Figure 8.2 Interface 2 39](#_Toc164751354)

[Figure 8.3 Interface 3 39](#_Toc164751355)

[Figure 8.4 Interface 4 40](#_Toc164751356)

[Figure 8.5 Interface 5 41](#_Toc164751357)

[Figure 8.6 Predicting Result 41](#_Toc164751358)

[Figure 8.7 Disapproved loan 42](#_Toc164751359)

[Figure 8.8 Approved loan 42](#_Toc164751360)

**Abstract**

Banks are making major part of profits through loans. Loan approval is a very important process for banking organizations. It is very difficult to predict the possibility of payment of loan by the customers because there is an increasing rate of loan default. And the banking authorities are finding it more difficult to correctly access loan requests and tackle the risks of people defaulting on loans. The term banking can be defined as receiving and protecting money that is deposited by the customers. This process also includes lending money to the customers which will be repaid within given time.

The primary objective of any bank is to provide their wealth in safer hands. In recent times banks approve loans only after thorough background checks and verifications of customer's credentials. Yet there is no guarantee that the customer is deserved for the loan. Banking processes use manual procedures to determine whether or not a borrower is suitable for a loan based on results.

Manual procedures were mostly effective, but they were insufficient when there were a large number of loan applications. This project aims to provide a loan to a deserving applicant out of all applicants. An efficient and non-biased system that reduces the bank's time employs checking every applicant on a priority basis.. The best part is that it is efficient for both banks and applicants.

Key words :- Django, Logistic Regression , Decision Trees, Random Forests, Support Vector Machines.

# Introduction

The loan eligibility prediction project aims to develop a machine learning model that can accurately determine whether a loan applicant qualifies for a loan based on specific parameters. The banking and finance sector, loan approval task is crucial for streamlining the loan approval process and reducing manual effort. By leveraging historical data on loan applicants, including factors such as income, credit history, loan amount, and more, the project seeks to automate the assessment process using machine learning techniques.

The dataset will undergo preprocessing to handle missing values, outliers, and categorical variables. Feature engineering will be employed to extract relevant information, and various machine learning algorithms such as logistic regression, support vector machines and random forests will be evaluated for model selection. In the preprocessing stage, missing values will be addressed through imputation techniques like mean, median, or mode replacement, while outliers may be handled using techniques such as trimming or transformation.

Categorical variables will undergo encoding or transformation to numerical representations suitable for machine learning algorithms. Feature engineering will involve creating new features or transforming existing ones to enhance model performance. Evaluation of machine learning algorithms like logistic regression, support vector machines, and random forests will include tuning hyperparameters, cross-validation, and performance metrics assessment to select the most suitable model for the dataset's characteristics and objectives.

## **Need of the study**

In today’s world, obtaining loans from financial institutions has become a very common phenomenon. Every day many people apply for loans, for a variety of purposes. But not all the applicants are reliable, and not everyone can be approved. Every year, there are cases where people do not repay the bulk of the loan amount to the bank which results in huge financial loss. The risk associated with making a decision on a loan approval is immense. Hence, the idea of this project is to gather loan data from the Lending Club website and use machine learning techniques on this data to extract important information and predict if a customer would be able to repay.

## Problem statement

While sanctioning monetary loans, banks, and other lending authorities need to have a surety of getting their money back along with interest on it. So, they need to know the credibility of the borrower before lending the money. this, the lending authorities need to verify the background and credibility of the borrower thoroughly. However, going through several variables and factors manually for every borrower is a time taking process and is highly inefficient.

## Project overview / Specifications

The aim of the loan eligibility prediction project is to utilize machine learning techniques to create a predictive model capable of automatically determining whether a loan applicant qualifies for a loan based on various factors. This project seeks to improve the efficiency of loan processing by replacing traditional manual assessment methods with a faster and more accurate automated system.

By leveraging predictive analytics, the goal is to mitigate risks associated with loan approvals by analyzing applicant characteristics and making data-driven decisions. Additionally, the project aims to enhance the overall customer experience by expediting loan approval processes, ensuring fairness, and providing transparency in decision-making. Financial institutions stand to benefit from reduced operational costs through optimized resource allocation and decreased reliance on labor-intensive reviews.

The project also aims to deliver a scalable solution capable of handling large volumes of loan applications and adapting to evolving market conditions and regulatory requirements. Ultimately, the project aims to provide financial institutions with a reliable decision support tool that facilitates consistent and objective loan approval decisions based on quantifiable data, thereby transforming and optimizing the loan approval process.

### Objective

The objective of a loan eligibility prediction project is to create a reliable computer model that can accurately determine whether an individual should be approved for a loan based on their personal and financial details. This model aims to provide clear and trustworthy decisions, taking into account factors like income, credit history, employment status, and other relevant information.

It should ensure fairness by treating all applicants equally and comply with legal regulations governing lending practices. Ensuring fairness and compliance with legal regulations will involve incorporating fairness-aware techniques into the model's development and implementing transparent decision-making processes. Additionally, the model should streamline the loan approval process, making it faster and more efficient for both applicants and lenders.

### Goal

The ultimate goal is to deploy a trained model capable of accurately predicting loan eligibility for new applicants, thereby improving efficiency and decision-making in the loan approval process within financial institutions. This deployment aims to revolutionize loan approval by leveraging data-driven insights to expedite the decision-making process. By accurately assessing loan eligibility, financial institutions can mitigate risks and allocate resources more effectively. Ultimately, this initiative seeks to enhance customer satisfaction by providing faster, fairer, and more transparent loan approval experiences.

## Machine Learning

Machine learning is a subset of artificial intelligence (AI) that focuses on the development of algorithms and models that allow computers to learn from and make predictions or decisions based on data. Instead of being explicitly programmed to perform a task, machines learn from patterns and insights derived from the data they process. It is a powerful tool that enables computers to learn from data and make decisions or predictions without being explicitly programmed for each specific task.

Machine learning's essence lies in its ability to empower computers to autonomously learn from data and refine their performance over time. By leveraging algorithms and models, machine learning enables computers to detect intricate patterns and derive meaningful insights from vast datasets. This iterative learning process allows machines to adapt and improve their predictions or decisions without human intervention continually. Moreover, machine learning excels in tasks where traditional rule-based programming falls short. Instead of relying on explicit instructions, machines learn from examples, adjusting their behavior based on feedback and experiences. This flexibility makes machine learning particularly valuable in dynamic environments where patterns evolve or data characteristics change over time.

## Django Framework

Django is a popular web framework for building web applications using Python. It provides a set of tools and libraries that streamline the development process, allowing developers to focus on creating their application's functionality rather than dealing with repetitive tasks. Django includes features such as Model View Template pattern (MVT) and includes an object-relational mapper (ORM) for interacting with databases, a built-in admin interface for managing content, a URL routing system for handling incoming requests, and support for forms and templates for generating HTML content.

Additionally, Django emphasizes security best practices to help developers build secure web applications. Django is known for its scalability, flexibility, and ease of use, making it a preferred choice for developers building a wide range of web applications. Django is a versatile and feature-rich web framework that simplifies the development of web applications. Its extensive documentation, vibrant community, and vast ecosystem of third-party packages make it a popular choice for developers worldwide.

# Literature Survey

In this section we provide relevant background on previous work on Loan eligibility prediction. Previous work on loan eligibility prediction has predominantly focused on leveraging various machine learning algorithms to analyze historical loan data and predict the likelihood of approval for new applicants. Studies have explored the effectiveness of algorithms such as logistic regression, decision trees, random forests, and neural networks in this context. Additionally, researchers have investigated feature selection techniques and model evaluation metrics to improve predictive accuracy and generalization performance. Despite advancements, challenges remain in ensuring fairness, interpretability, and regulatory compliance in loan approval models.

In the paper [1] The loan default prediction of the Chinese peer-to-peer (P2P) market using R.F, XG Boost, GBM, and Neural Network machine learning models. Their four models exceeded 90% accuracy, with RF being the superior model. This research is closely related to our study in terms of methods used and algorithms deployed.

In the paper [2] An Approach for Prediction of Loan Approval Using Machine Learning Algorithm, this paper has taken the data of previous customers of various banks to whom on a set of parameters loan were approved. So the machine learning model is trained on that record to get accurate results. Our main objective of this research is to predict the safety of loan to predict loan safety, the logistic regression algorithm is used.

In the paper [3] Approval Prediction System based on Machine Learning Approach, this model is used for the banking system or anyone who wants to apply for a loan. It will be very helpful in bank management. From the analysis of the data, it is very clear that it reduces all the frauds done at the time of loan approval.

Time is also very precious for everyone through this not only the bank but also the waiting time of the applicant will also reduce. As it seems, it will not deal with some special cases when only one parameter is enough for the decision, but it is quite efficient and reliable in some instant. In the future, this prediction module can be more improved and integrated. The system is prepared on the previous training data but in the future, it is possible to make changes to software, which can accept new testing data and should also take part in training data and predict accordingly.

In the paper [4] Prediction of Modernized Loan Approval System Based on Machine Learning Approach , this Prediction of a modernized loan approval system is incredibly helpful for banks and also the clients. This system checks the candidate on his priority basis. Customer can submit his application directly to the bank so the bank will do the whole process, no third party or stockholder will interfere in it.

And finally, the bank will decide that the candidate is deserving or not on its priority basis. The only object of this research paper is that the deserving candidate gets straight forward and quick results. In this system, we take some data from the user like his monthly income, marriage status, loan amount, loan duration, etc. Then the bank will decide according to its parameters whether the client will get the loan or not.

# Requirement Analysis

We developed automatic loan prediction using machine learning techniques. In this we will train the machine with previous dataset. So machine can analyze and understand the process. Then machine will check for eligible applicant and give us result. Our automatic loan prediction system utilizes machine learning techniques trained on historical datasets. By exposing the machine to past data, it comprehends the intricacies of the loan approval process. Once trained, the machine autonomously evaluates new applicants, determining their eligibility based on learned patterns and features from the training data. This approach streamlines the decision-making process, providing rapid and accurate results for loan approval.

## Functional Requirements

Functional requirements for a loan eligibility prediction system outline the specific features and capabilities that the system must possess to effectively assess loan applications and make accurate eligibility decisions. These requirements ensure that the system meets the needs of stakeholders and users involved in the loan approval process. Here are key functional requirements for a loan eligibility prediction system:

1. Input Data Handling:

1. Data Collection: Ability to gather relevant applicant information such as income, credit score, employment details, loan amount, etc.
2. Data Validation: Ensure data quality by checking for completeness, accuracy, and consistency.

2. Preprocessing and Feature Engineering:

1. Data Cleaning: Handle missing values, outliers, and inconsistencies in the input data.
2. Feature Selection: Identify and select the most relevant features for loan eligibility assessment.
3. Normalization/Scaling: Standardize numerical features to a common scale to ensure fair comparison.

3. Model Development and Training:

1. Model Selection: Choose appropriate machine learning algorithms (e.g., logistic regression, decision trees, neural networks) for loan prediction.
2. Training Process: Train the model using historical loan data with known outcomes (approved/rejected).
3. Hyperparameter Tuning: Optimize model performance by tuning parameters (e.g., learning rate, regularization) based on validation results.

4. Prediction and Decision Making:

1. Real-time Prediction: Assess loan applications in real-time to provide quick decisions.
2. Probability Estimation: Calculate and provide the probability of loan approval based on applicant information.
3. Decision Thresholding: Apply predefined decision thresholds (e.g., probability cutoff) to determine loan eligibility.

5. Integration and Deployment:

1. Integration with Front-end Systems: Interface with user-facing applications (e.g., loan application portals) to accept input and provide output.
2. Scalability: Handle varying levels of workload and adapt to increasing data volumes.
3. Deployment Flexibility: Deploy as a standalone system or integrate within existing loan processing workflows.

6. Monitoring and Maintenance:

1. Performance Monitoring: Continuously monitor model performance (e.g., accuracy, precision, recall) to ensure reliability and consistency.
2. Model Updates: Periodically update the model using new data to account for changing trends and patterns in loan applications.
3. Security and Compliance: Implement data security measures and adhere to regulatory guidelines (e.g., GDPR, data privacy laws) governing loan application processing.

7. Reporting and Analytics:

1. Dashboard and Reporting: Provide insights through dashboards and reports on loan approval trends, application characteristics, and model performance metrics.
2. Ad Hoc Analysis: Allow users to perform ad hoc analysis on loan data to understand decision factors and outcomes.

The functional requirements outlined here establish the scope and functionality of a loan eligibility prediction system. They are designed to ensure the system's ability to accurately assess loan applications, improve decision-making procedures, and facilitate streamlined loan processing workflows. These requirements serve as the foundation for developing a robust and efficient system that meets the needs of both applicants and lenders.

## Non Functional Requirements

Non-functional requirements for a loan eligibility prediction system focus on aspects that define how the system should behave or perform rather than specific functionalities. These requirements are essential for ensuring the system's quality, usability, reliability, and scalability. Here are key non-functional requirements for a loan eligibility prediction system. By addressing these non-functional requirements, the loan eligibility prediction system can ensure high quality, reliability, and usability, while also complying with regulatory and industry standards.

1. Performance:

1. Response Time: The system should provide quick responses to loan applications, ideally within milliseconds or seconds.
2. Throughput: Ability to handle a large volume of loan applications concurrently without performance degradation.
3. Scalability: The system should scale seamlessly to accommodate increasing numbers of users and data volumes.

2. Reliability:

1. Availability: The system should be available and accessible for loan processing operations at all times, with minimal downtime.
2. Fault Tolerance: Ability to recover from failures (e.g., server crashes, network issues) gracefully without data loss.
3. Data Integrity: Ensure the accuracy and consistency of data used for loan eligibility assessment.

3. Security:

1. Data Protection: Implement strong data encryption techniques to protect sensitive applicant information (e.g., personal, financial data).
2. Access Control: Ensure that only authorized personnel have access to the system and its functionalities.
3. Compliance: Adhere to regulatory standards and guidelines related to data privacy and security (e.g., GDPR, HIPAA).

4. Usability:

1. User Interface: Provide a user-friendly interface for loan officers or applicants to interact with the system effectively.
2. Accessibility: Ensure the system is accessible to users with disabilities and supports multiple devices (e.g., desktops, mobile devices).
3. Documentation: Maintain comprehensive and up-to-date documentation for system users and administrators.

5. Maintainability:

1. Modularity: Design the system using modular components that can be maintained, updated, or replaced independently.
2. Code Quality: Write clean, well-documented, and maintainable code to facilitate future enhancements and bug fixes.
3. Version Control: Implement version control practices to manage changes to the system's codebase and configuration.

6. Performance Testing:

1. Load Testing: Conduct performance testing under simulated high-load conditions to assess system behaviour and identify performance bottlenecks.
2. Stress Testing: Evaluate the system's resilience by subjecting it to extreme workload scenarios.

7. Integration:

1. Interoperability: Ensure compatibility and seamless integration with external systems (e.g., databases, third-party services) used in loan processing workflows.
2. API Design: Provide well-defined APIs for integrating the loan eligibility prediction system with other enterprise applications.

8. Compliance:

1. Regulatory Compliance: Ensure compliance with industry regulations and standards governing loan processing and eligibility assessment.
2. Ethical Considerations: Address ethical concerns related to fairness, bias, and transparency in loan decision-making processes.

These non-functional requirements are critical for delivering a robust, secure, and efficient loan eligibility prediction system that meets the needs and expectations of stakeholders, users, and regulatory bodies. They contribute to the overall quality, reliability, and performance of the system throughout its lifecycle.

Advantages of Random Forest for Loan Eligibility Prediction:

1. High Accuracy: Random Forest tends to yield high accuracy compared to individual decision trees.
2. Robust to Overfitting: The ensemble nature of Random Forest reduces overfitting by averaging out predictions from multiple trees.
3. Handles Missing Data: Random Forest can handle missing values in the dataset without the need for imputation.
4. Feature Importance: It provides insights into feature importance, helping to identify which factors (e.g., income, credit score) are most influential in determining loan eligibility.

Implementation Considerations:

1. Hyperparameter Tuning: Tuning parameters like the number of trees (n\_estimators), maximum depth of trees, and the number of features considered at each split can optimize Random Forest performance.
2. Interpretability: While Random Forest offers excellent predictive performance, interpreting the model may be challenging due to its ensemble nature.

In summary, Random Forest is a versatile and robust algorithm suitable for loan eligibility prediction due to its ability to handle complex datasets, reduce overfitting, and provide insights into feature importance for decision-making. Its ensemble approach leverages the collective wisdom of multiple decision trees to make accurate predictions on loan applications.

## Hardware Requirements

Hardware requirements are essential to ensure that the project's software can run efficiently and effectively. These requirements include the minimum specifications for the CPU, RAM, storage, and other hardware components. By meeting these requirements, the project's software can run smoothly, reducing the risk of crashes, errors, and other issues.

1. RAM (min 16GB)
2. Hard Disk (min 128GB)
3. CPU
4. X64 based Processor
5. 64-bit operating system

## Software Requirements

Software requirements are also critical to the success of a project. These requirements include the specific versions of software and operating systems that are compatible with the project's software. By ensuring that the project's software is compatible with the required software and operating systems, you can reduce the risk of compatibility issues and ensure that the project's software functions as intended.

1. Software: Python 3.10 or high version
2. IDE: Visual Studio Code
3. Web Framework**:** Django

# Algorithms

## Logistic regression

Logistic regression is a fundamental and widely used statistical technique for binary classification tasks such as loan eligibility prediction. Despite its name, logistic regression is a classification algorithm rather than a regression algorithm. Here's an explanation of how logistic regression works for loan eligibility prediction. In logistic regression, the goal is to model the probability that a loan application belongs to a certain class (e.g., approved or rejected) based on input features. The algorithm estimates the probability using a logistic (sigmoid) function, which outputs values between 0 and 1.

The logistic function is defined as ( sigma(z) = frac{1}{1 + e^{-z}} ), where ( z ) is a linear combination of the input features and model parameters. The logistic regression model learns the optimal parameters (coefficients) that minimize the difference between the predicted probabilities and the actual class labels in the training data. This optimization process is typically done using techniques like maximum likelihood estimation or gradient descent.

To make predictions, logistic regression uses a decision threshold (often 0.5). If the predicted probability is above the threshold, the loan application is classified as approved (positive class); otherwise, it's classified as rejected (negative class). One of the key advantages of logistic regression is its simplicity and interpretability. The model coefficients indicate the impact of each feature on the probability of loan approval.

Positive coefficients indicate a positive relationship with loan approval, while negative coefficients indicate a negative relationship. Logistic regression performs well when the relationship between the input features and the target variable is linear or can be approximated by a linear function. However, it may struggle with complex non-linear relationships unless feature engineering or transformation is applied.

## Support Vector Machine

The Support Vector Machine (SVM) algorithm is a powerful supervised learning technique used for loan eligibility prediction and various other classification tasks. SVM is particularly effective in scenarios where clear boundaries between classes exist within the dataset. Here's an explanation of how SVM works for loan eligibility prediction.

SVM aims to find the optimal hyperplane that separates different classes of data points in a high-dimensional space. In the context of loan eligibility prediction, each loan application is represented as a data point with multiple features such as income, credit score, and loan amount. The goal is to classify these data points into two categories: approved or rejected.

The key idea behind SVM is to identify a hyperplane that maximizes the margin between the nearest data points of different classes, known as support vectors. These support vectors are the data points closest to the decision boundary and play a crucial role in defining the hyperplane. The hyperplane is positioned such that it minimizes classification errors and generalizes well to unseen data.

In cases where the data points are not linearly separable in their original feature space, SVM can employ a technique called the kernel trick. This involves mapping the input features into a higher-dimensional space using a kernel function (e.g., polynomial kernel, radial basis function (RBF) kernel) to make the data linearly separable. The kernel function computes the dot product between the mapped feature vectors efficiently without explicitly transforming them, thereby avoiding the computational burden associated with high-dimensional spaces.

SVM has several advantages for loan eligibility prediction, including its ability to handle high-dimensional data, its effectiveness in finding complex decision boundaries, and its resistance to overfitting when appropriate regularization techniques are applied. However, parameter tuning, such as selecting the optimal kernel and regularization parameters, is crucial for achieving optimal performance and generalization.

## Naive Bayes

The Naive Bayes algorithm is a popular and effective method for classification tasks, especially in natural language processing and document categorization. Its strength lies in its simplicity and efficiency. The algorithm is based on Bayes' theorem, which describes the probability of a hypothesis given the evidence. In the context of classification, Naive Bayes calculates the probability of each class given a set of features, assuming that these features are conditionally independent.

This assumption simplifies the computation and makes the algorithm scalable even with large datasets. To use Naive Bayes for classification, we first calculate the prior probability of each class based on the training data. Then, we compute the likelihood of the features given each class. Finally, using Bayes' theorem, we combine these probabilities to estimate the posterior probability of each class for a new instance. Despite its simplicity and the "naive" assumption of feature independence.

Naive Bayes often performs surprisingly well in practice and can serve as a baseline model for more complex classification tasks. It's particularly effective for text classification, where features (words) can be treated as independent given the class label. However, Naive Bayes may not perform well if the feature independence assumption is violated or if the dataset is highly imbalanced. Nonetheless, it remains a widely used and studied algorithm in the machine learning community.

## Random Forest

The Random Forest algorithm is a powerful machine learning technique used for loan eligibility prediction and other classification tasks. It belongs to the ensemble learning methods, which combine multiple individual models (in this case, decision trees) to improve overall performance and generalization. Here's an explanation of how the Random Forest algorithm works for loan eligibility prediction:

1. Ensemble of Decision Trees:

1. Base Learner: The Random Forest algorithm is composed of a collection (ensemble) of decision trees.
2. Decision Trees: Each decision tree in the Random Forest is built using a subset of the training data and a subset of the features. This randomness helps to introduce diversity among the trees.

2. Random Sampling with Replacement (Bootstrap Aggregating):

1. Bagging Technique: The algorithm uses a technique called bootstrap aggregating (bagging). It involves creating multiple bootstrap samples (random samples with replacement) from the training dataset.
2. Training Trees: Each decision tree is trained on one of these bootstrap samples, which introduces randomness and variability into the training process.

3. Feature Randomness:

1. Feature Subset: When building each decision tree, only a random subset of features is considered at each split.
2. Feature Importance: By randomly selecting features, the algorithm learns which features are most informative for predicting loan eligibility.

4. Voting for Prediction:

1. Majority Voting: When making predictions for a new loan application, each decision tree in the Random Forest independently predicts the loan eligibility (e.g., approved or rejected).
2. Final Prediction: The final prediction is determined by taking a majority vote across all the individual trees. The most common prediction (e.g., approval) becomes the final prediction of the Random Forest.

## Dataset and Description

Table 4.1 Dataset and Description

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Description** | **Type** |
| **SeriousDlqin2yrs** | **Person experienced 90 days past due delinquency or worse** | **Y/N** |
| RevolvingUtilizationOfUnsecuredLines | Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits | percentage |
| age | Age of borrower in years | integer |
| NumberOfTime30-59DaysPastDueNotWorse | Number of times borrower has been 30-59 days past due but no worse in the last 2 years | integer |
| DebtRatio | Monthly debt payments, alimony,living costs divided by monthy gross income | percentage |
| MonthlyIncome | Monthly income | real |
| NumberOfOpenCreditLinesAndLoans | Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards) | integer |
| NumberOfTimes90DaysLate | Number of times borrower has been 90 days or more past due | integer |
| NumberRealEstateLoansOrLines | Number of mortgage and real estate loans including home equity lines of credit | integer |
| NumberOfTime60-89DaysPastDueNotWorse | Number of times borrower has been 60-89 days past due but no worse in the last 2 years | integer |
| NumberOfDependents | Number of dependents in family excluding themselves (spouse, children etc.) | integer |

# System Design

System modeling involves creating abstract representations of a system, with each model offering a unique viewpoint or aspect of the system's behavior, structure, or interactions. These models help stakeholders understand, analyze, and communicate various aspects of the system throughout its development lifecycle. By providing different perspectives, system modeling enables comprehensive understanding and effective management of complex systems.

**A diagram of a data processing process

Description automatically generated**

Figure 5.1 Loan Prediction Architecture

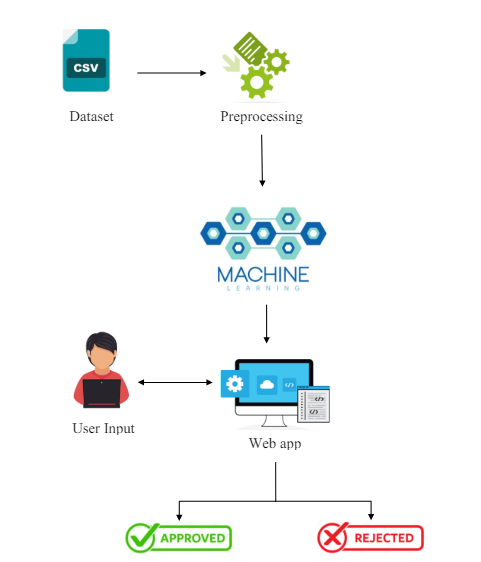


Figure 5.2 Architecture Diagram

## Existing System

Till now loans are processed by various banks through pen and paperwork. When the large no of customers’ apply for bank loan these bank take lot of time to approve their loan. After approval of loan by the banks, there is no surety that the chosen applicant is capable of paying loan or not. Many banks use their own software’s for the loan approval.

In existing system datamining algorithms for the loan approval; this is the old technique for the approval of loan. Multiple data sets are combined and form a generalized datasets, and different machine learning algorithms are applied to generate results. But these techniques are not up to the mark. Due to this huge banks are suffering from financial crises. To resolve this issue we introduce a new way for approval of loans.

## Proposed System

To deal with the problem, we developed automatic loan prediction using machine learning techniques. We will train the machine with previous dataset. So machine can analyze and understand the process. Then machine will check for eligible applicant and give us result. In response to the issue, we've devised an automatic loan prediction system employing machine learning methodologies.

Through training on historical datasets, the machine comprehends and analyzes the underlying processes. Subsequently, it evaluates new applicants, identifying eligible candidates and delivering prompt outcomes. This approach enhances efficiency and accuracy in the loan approval process. This automated loan prediction system serves as a pivotal solution to streamline the loan approval process.

By harnessing machine learning techniques and leveraging past data, the system gains insights into intricate patterns and trends. Empowered with this knowledge, it efficiently sifts through new applicant information, swiftly identifying those who meet eligibility criteria. This not only optimizes decision-making but also ensures a fair and transparent process for both applicants and lenders alike.

## Use Case Diagram

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. Actors are the external entities that interact with the system. The use cases are represented by either circles or ellipses. A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse. An actor is a person, organization or external system that plays a role in one or more interaction with the system.

The Figure 5.3 shows the use case representation of the system.

A diagram of a machine learning

Description automatically generated

Figure 5.3 Use Case Diagram

## Class Diagram

Class diagrams give an overview of a system by showing its classes and the relationships among them. Class diagrams are static – they display what interacts but not what happens when they do interact. In general a class diagram consists of some set of attributes and operations. Operations will be performed on the data values of attributes. The Figure 5.3 shows the class diagram representation of the system.

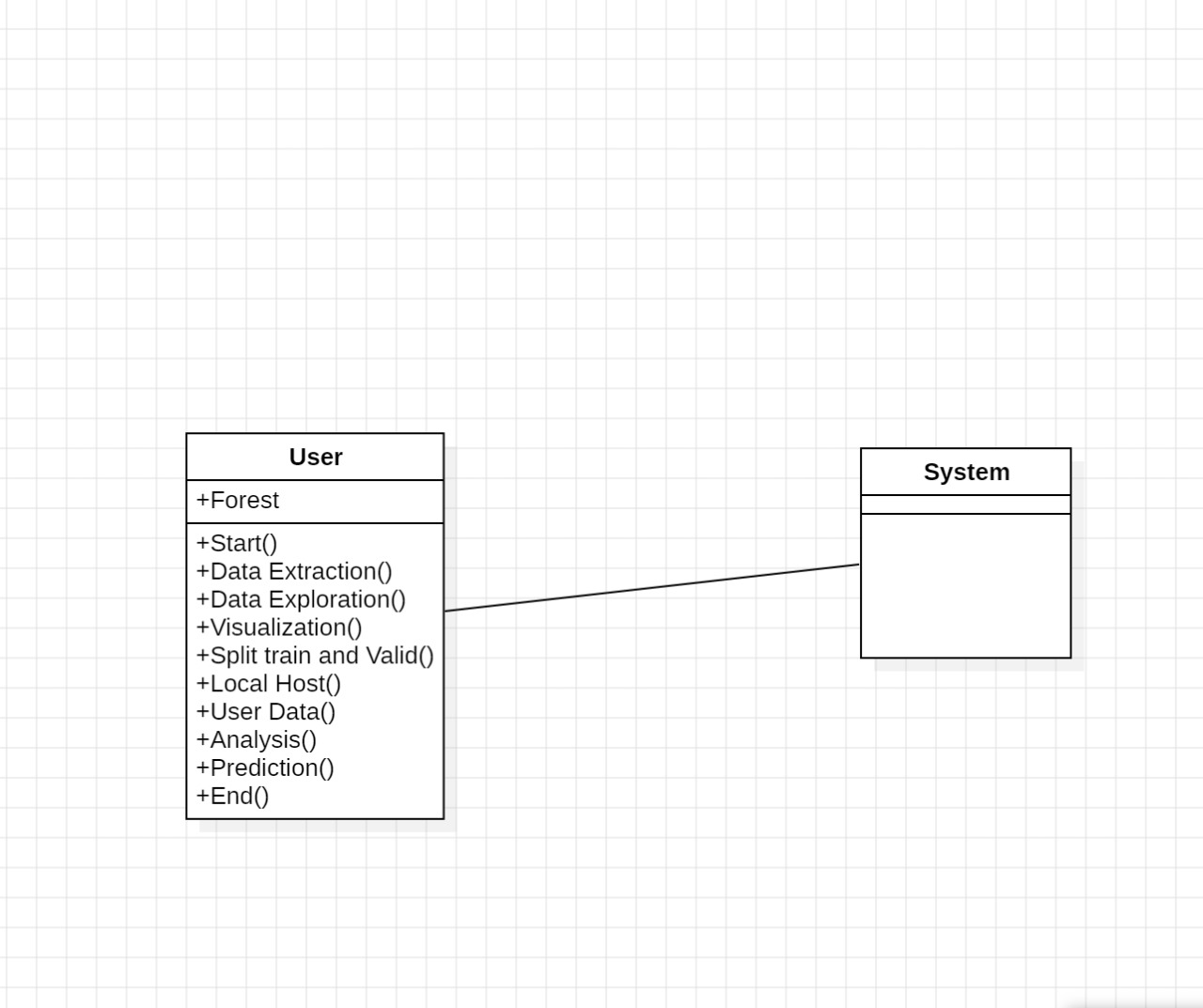


Figure 5.3 Class Diagram

## Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. In UML, an activity diagram provides a view of the behaviour of a system by describing the sequence of actions in a process. The most important shape types:

1. Rounded rectangles represent activities.
2. Diamonds represent decisions.
3. Bars represent the start or end of concurrent activities.
4. A black circle represents the start of the workflow.
5. An encircled circle represents the end of the workflow

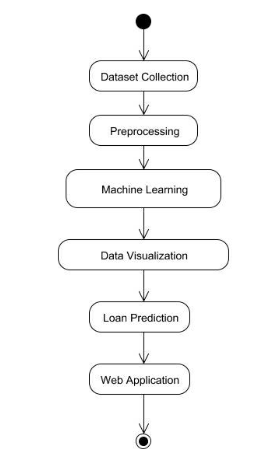
.

Figure 5.4 Activity Diagram

## Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. The Figure 5.6 shows the sequence diagram representation of the system.

A diagram of a computer program

Description automatically generated

Figure 5.5 Sequence Diagram

## Collaboration Diagram

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram is used to show the relationship between the objects in a system. The collaboration diagram illustrates messages being sent between classes and objects. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming.

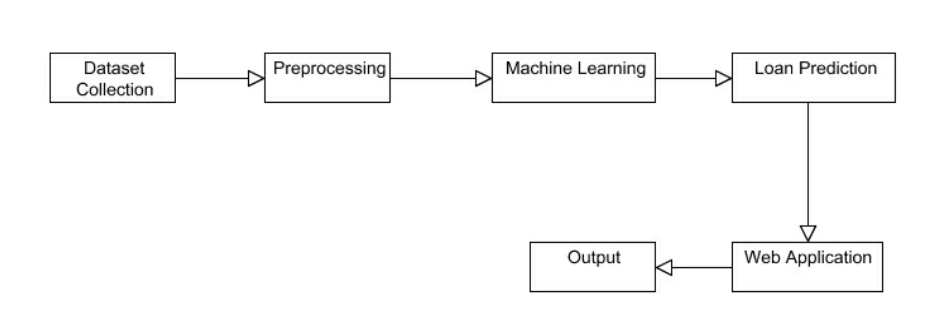


Figure 5.6 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is a type of UML (Unified Modeling Language) diagram that visualizes the interactions and relationships among objects or components within a system. It emphasizes the messages exchanged between objects to achieve a particular behavior or functionality.

Collaboration diagrams are useful for understanding the high-level interactions between objects in a system and can aid in the design and communication of system architecture and behavior. They provide a visual representation of how objects work together to achieve specific functionalities or behaviors within a software system.

# Implementation

Implementation process, it's essential to pay attention to data quality, model interpretability, fairness, and regulatory compliance, especially in the context of financial decision-making such as loan eligibility prediction. Additionally, consider using techniques such as cross-validation and model explainability methods to ensure that the model's predictions are reliable and understandable.

## Modules and Libraries

The terms "modules" and "libraries" refer to collections of pre-written code that provide functionalities to facilitate various aspects of the machine learning workflow. Modules are individual units of code within a library that serve specific purposes or functionalities, while libraries are collections of modules that provide a comprehensive set of tools and capabilities for performing machine learning tasks.

Together, modules and libraries enable practitioners to leverage existing implementations and functionalities to build, train, evaluate, and deploy machine learning models efficiently. In Python, a module is a file containing Python definitions and statements. These modules can consist of functions, classes, and variables that can be utilized by other Python scripts or modules.

Modules are typically organized around a specific set of tasks or functionalities. In machine learning, modules often refer to smaller units of code within a library that serve a particular purpose, such as implementing specific algorithms (e.g., linear regression, support vector machines) or providing utilities for data preprocessing, evaluation, or visualization.

A library is a collection of modules or packages that provide a set of functionalities to perform various tasks. Libraries in machine learning often contain implementations of machine learning algorithms, tools for data manipulation and preprocessing, evaluation metrics, visualization capabilities, and more. These libraries are designed to simplify and accelerate the development, training, evaluation, and deployment of machine learning models. Examples of popular machine learning libraries include scikit-learn, TensorFlow, PyTorch, Keras, pandas, and Matplotlib.

### ****matplotlib.pyplot****

This module offers a plotting framework similar to MATLAB, making it a popular choice for data visualization in Python. It supports a wide range of plot types, including line plots, scatter plots, bar charts, histograms, and more. By providing an intuitive interface and comprehensive functionality, it facilitates the creation of expressive and insightful visualizations for data analysis and presentation purposes.

### Pandas

This library is a robust resource for data manipulation and analysis tasks. Its key feature is the Data Frame data structure, designed for effortless handling of labeled data. Widely employed in data cleaning, transformation, and exploration, it simplifies complex operations and facilitates insightful insights extraction from datasets. Its versatility and efficiency make it an indispensable tool in various domains, from research and business analytics to machine learning and beyond.

### Seaborn

Seaborn, leveraging matplotlib as its foundation, offers a user-friendly interface for generating visually appealing and informative statistical graphics. By abstracting away the complexities, it streamlines the creation of intricate visualizations like heatmaps, violin plots, and categorical plots. This simplification enables researchers and data analysts to focus more on interpreting insights rather than grappling with technical intricacies, thus enhancing the efficiency and effectiveness of data exploration and communication.

### Numpy

NumPy is the fundamental package for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. NumPy serves as the cornerstone for numerical computation in Python. Its primary feature is the ability to handle large, multi-dimensional arrays and matrices efficiently. Moreover, NumPy offers a comprehensive suite of mathematical functions that operate seamlessly on these arrays, facilitating fast and reliable numerical operations.

### scikit-learn

Scikit-learn, often abbreviated as sklearn, stands as a leading machine learning library within the Python ecosystem. Renowned for its simplicity and efficiency, scikit-learn offers a plethora of tools tailored for data mining and analysis tasks. From classification and regression to clustering and dimensionality reduction, it encompasses a diverse array of algorithms to suit different machine learning needs. With its user-friendly interface and comprehensive documentation, scikit-learn has become a go-to choice for both beginners and experienced practitioners in the field of machine learning.

### sklearn.svm

Support Vector Machines (SVM) represent a supervised learning algorithm commonly employed for both classification and regression tasks. It endeavors to discern the optimal hyperplane within the feature space that effectively segregates distinct classes or predicts continuous values. By maximizing the margin between data points of different classes, SVM strives to achieve robust classification or regression performance, particularly in scenarios with complex data distributions or high-dimensional feature spaces.

### sklearn.naive\_bayes

Naive Bayes constitutes a family of probabilistic algorithms grounded in Bayes' theorem, under the "naive" assumption of feature independence. Primarily utilized for classification tasks, particularly prevalent in natural language processing (NLP) applications, Naive Bayes models estimate the likelihood of class labels given input features. Despite its simplistic assumption, Naive Bayes often yields satisfactory results and is favored for its efficiency and effectiveness, particularly in scenarios with high-dimensional feature spaces or limited training data.

### sklearn.linear\_model

This module encompasses a suite of linear models, encompassing Linear Regression, Logistic Regression, and Ridge Regression, among others. These models serve dual roles in regression and classification tasks, enabling prediction of continuous outcomes or categorical labels based on input features. Leveraging linear relationships between features and target variables, these models offer simplicity, interpretability, and computational efficiency, making them versatile choices across diverse domains, from finance and healthcare to marketing and beyond.

### sklearn.ensemble

The ensemble module in scikit-learn consolidates algorithms that amalgamate multiple base estimators to enhance generalization performance. Random Forest, an instance of ensemble learning, operates on the principle of decision trees. By constructing a multitude of decision trees and aggregating their predictions, Random Forest mitigates overfitting and enhances predictive accuracy. This approach renders Random Forest particularly effective in handling complex datasets and achieving robust performance across various machine learning tasks, including classification and regression.

### sklearn.calibration

This module offers utilities tailored to assess and enhance the calibration of probabilistic predictions generated by classifiers. One such function, calibration\_curve, facilitates the visualization of a calibration curve, illustrating the correspondence between predicted probabilities and actual probabilities. By plotting this curve, analysts can evaluate the reliability and accuracy of the classifier's probabilistic outputs, aiding in fine-tuning and optimizing model performance for tasks such as binary classification.

By visualizing the calibration curve, analysts can identify whether the classifier tends to be overconfident or underconfident in its predictions and make adjustments to improve its calibration, thereby optimizing model performance.

overall, the calibration curve function serves as a valuable tool for fine-tuning and optimizing the performance of classifiers, particularly in tasks such as binary classification where accurate probability estimates are essential.

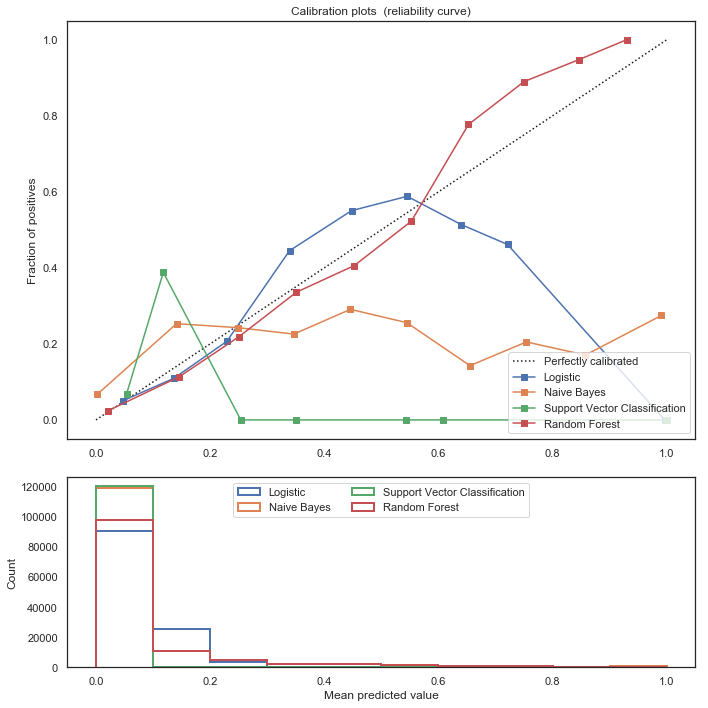


Figure 6.1 Calibration curve

A calibration curve is a graphical representation that illustrates the relationship between the predicted probabilities produced by a machine learning model and the actual outcomes observed in the data. It's particularly useful for models that provide probability estimates, such as logistic regression, support vector machines with probability outputs, or models trained using gradient boosting methods.

Calibration curves are important for assessing the reliability of probabilistic predictions from machine learning models, especially in applications where accurate probability estimates are crucial, such as risk assessment or medical diagnosis. They help to identify whether the model's predicted probabilities are well-calibrated and can be interpreted as reliable estimates of true probabilities.

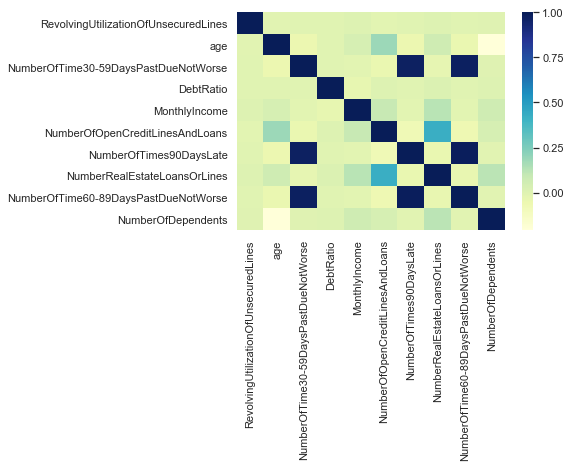


Figure 6.2 Correlation matrix

## Code URL

**GitHub**: A web-based hosting service for Git repositories, providing features such as code hosting, collaboration, issue tracking, and continuous integration.

The GitHub repository hosts code and data for Loan eligibility prediction. It includes Jupyter Notebook files which includes the important code for Loan eligibility prediction and python file for the web interface which is coded using Django framework. Below is the provided URL for accessing the repository.

<https://github.com/sriteja-28/Loan-eligibility-prediction-using-ensemble-learning>

# Evaluation

To assess the model's performance accurately, it's essential to employ a separate dataset, such as a validation or test set, unseen during training. This approach ensures an impartial evaluation of the model's ability to generalize to new data. These metrics collectively offer a comprehensive assessment of the model's effectiveness in classification tasks.

## Metrics for Evaluation

**Confusion Matrix:** Confusion Matrix is a performance measurement for the machine learning classification problems where the output can be two or more classes. It is a table with combinations of predicted and actual values. The confusion matrix provides a detailed breakdown of the model's predictions for each class.

1. TP: Instances correctly classified as positive by the model.
2. FP: Instances incorrectly classified as positive by the model.
3. TN: Instances correctly classified as negative by the model.
4. FN: Instances incorrectly classified as negative by the model.

**Accuracy:** Accuracy measures the proportion of correctly classified samples out of the total number of samples. While it's a straightforward metric, it might not be sufficient if the classes are imbalanced. The following Equation (6.1) represents the formula for calculation accuracy.

|  |  |  |
| --- | --- | --- |
|  | Results |  |

We use a machine learning algorithm on a loan prediction dataset and deploy the result using HTML, CSS, Python at the local server. To deploy the application on a network with multiple users you can create a robust and user-friendly loan prediction application that leverages machine learning techniques to provide valuable insights for users.

## Output

A screenshot of a computer

Description automatically generated

Figure 8.1 Interface 1

Figure 8.1 shows the loan prediction system of the applicants based on the value enter by the bank employee.

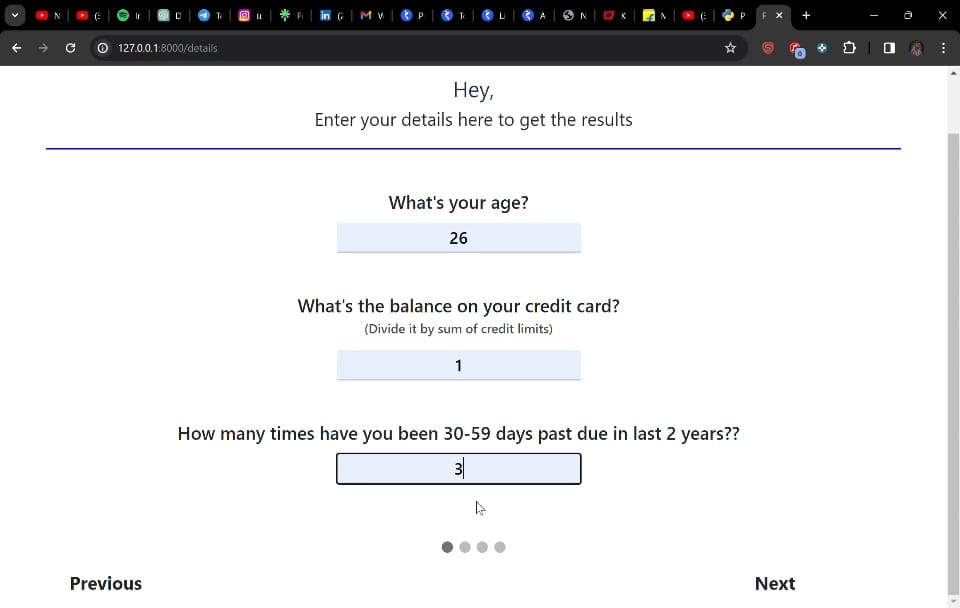


Figure 8.2 Interface 2

Figure 8.2 shows the loan prediction system where customer need to enter their age, balance on credit card and 30-59 days past due.

A screenshot of a computer screen

Description automatically generated

Figure 8.3 Interface 3

Figure 8.3 shows the loan prediction system where customer need to enter their monthly income,No.of open loans currently owe and the 90 days past due.

A screenshot of a computer screen

Description automatically generated

Figure 8.4 Interface 4

Figure 8.4 shows the loan prediction system where customer need to enter their mortgage and real estate loans , 60-89 days past due and dependents in their family.

A screenshot of a computer

Description automatically generated

Figure 8.5 Interface 5

Figure 8.5 shows the loan prediction system where customer need to enter their debt ratio,loan amount and pay back period.

A screenshot of a computer

Description automatically generated

Figure 8.6 Predicting Result

Figure 8.6 shows the loan prediction system result of the customer.

A screenshot of a computer

Description automatically generated

Figure 8.7 Disapproved loan

Figure 8.7 shows the loan disapproval for the person who can’t repay the loan amount.

A screenshot of a computer

Description automatically generatedFigure 8.8 Approved loan

Figure 8.8 shows the loan approval for a person who will repay the loan amount.

# Future Work

Today’s fast-growing IT industry needs to discover new technology and update the old technology that helps us to reduce human intervention and increase the efficiency of the work. This model is used for the banking system or anyone who wants to apply for a loan. It will be very helpful in bank management. From the analysis of the data, it is very clear that it reduces all the frauds done at the time of loan approval. Time is also very precious for everyone through this not only the bank but also the waiting time of the applicant will also reduce.

As it seems, it will not deal with some special cases when only one parameter is enough for the decision, but it is quite efficient and reliable in some instant. From a proper analysis of positive points and constraints on the component, it can be safely concluded that the product is a highly efficient component. This application is working properly and meeting to all Banker requirements.

This component can be easily plugged in many other systems. There have been numbers of cases of computer glitches, errors in content and most important weight of features is fixed in automated prediction system, So in the near future the so –called software could be made more secure, reliable and dynamic weight adjustment. In near future this module of prediction can be integrate with the module of automated processing system.

The system is trained on old training dataset in future software can be made such that new testing date should also take part in training data after some fix time. In the future, this prediction module can be more improved and integrated. The system is prepared on the previous training data but in the future, it is possible to make changes to software, which can accept new testing data and should also take part in training data and predict accordingly.

The main purpose of the project is to classify and analyze the nature of the loan applicants. From a proper analysis of data set and constraints of the banking sector, seven different graphs were generated and visualized. From the graphs, many conclusions have been made and information were inferred such as short-term loan was preferred by majority of the loan applicants and the clients majorly apply loan for debt consolidation.

This project work can be extended to higher level in future. Predictive model for loans that uses machine learning algorithms, where the results from each graph of the paper can be taken as individual criteria for the machine learning algorithm. The context emphasizes the transformative potential of machine learning in banking, particularly in loan approval processes, to reduce human intervention and enhance efficiency. The described model shows promise in minimizing fraud, reducing waiting times for applicants, and analyzing loan applicant data effectively through visualizations.

While not exhaustive, it addresses various parameters for decision-making, though it may not cover all special cases. Looking ahead, there's room for improvement in security, integration with other systems, and incorporating new data for training. Overall, the project underscores the value of machine learning in banking and the need for continuous refinement to meet evolving needs and challenges.

# Bibliography

|  |  |
| --- | --- |
| [1] | Ugochukwu .E. Orji ; "Machine Learning Models For Predicting Bank Loan Eligibility" 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON) |
| [2] | Anshika Gupta;Vinay Pant;Sudhanshu Kumar;Pravesh Kumar Bansal2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)] |
| [3] | Mohammad Ahmad Sheikh;Amit Kumar Goel;Tapas Kumar2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) |
| [4] | Raj, J. S., & Ananthi, J. V., "Recurrent neural networks and nonlinear prediction in support vector machine" Journal of Soft Computing Paradigm (JSCP), 1(01), 33-40, 2019. |
| [5] | Aakanksha Saha, Tamara Denning, VivekSrikumar, Sneha Kumar Kasera. "Secrets inSource Code: Reducing False Positives usingMachine Learning", 2020 InternationalConference on Communication Systems &Networks (COMSNETS), 2020. |
| [6] | X.Frencis Jensy,V.P.Sumathi,Janani Shiva Shri, "An exploratory Data Analysis for Loan Prediction based on nature of clients", International Journal of Recent Technology and Engineering (IJRTE),Volume-7 Issue-4S, November 2018. |
| [7] | Pidikiti Supriya, Myneedi Pavani, Nagarapu Saisushma,Namburi Vimala Kumari, k Vikash,"Loan Prediction by using Machine Learning Models", International Journal of Engineering and Techniques.Volume 5 Issue 2, Mar Apr 2019 . |
| [8] | Nikhil Madane, Siddharth Nanda,"Loan Prediction using Decision tree", Journal of the Gujrat Research History,Volume 21 Issue 14s, December 2019. |
| [9] | Vaidya and Ashlesha, Predictive and probabilistic approach using logistic regression: Application to prediction of loan approval, 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2017. |
| [10] | Amin, Rafik Khairul and Yuliant Sibaroni, Implementation of decision tree using C4. 5 algorithm in decision making of loan application by debtor (Case study: Bank pasar of Yogyakarta Special Region), 2015 3rd International Conference on Information and Communication Technology (ICoICT). IEEE, 2015. |