Loan Prediction System:

A Machine Learning Approach for Assessing Loan Eligibility

Professor:
Aghil Alaee Khanga

Team members – Group 6

Manan Davey
Meghana Dodda
Shraddha Kamath
Vatsal Rameshbhai Gohel
Venkatesh Dhanuskodi

TABLE OF CONTENTS

ABSTRACT

The banking sector has seen a significant increase in loan applications, leading to the need for an automated system to determine loan eligibility based on various factors. This project aims to develop a machine learning model to predict loan eligibility and identify the most important factors contributing to a customer's loan eligibility. The model will be trained on a dataset containing loan application details and evaluated using metrics such as accuracy, precision, recall, and F1-score. This system will aid both loan providers and loan seekers by providing a quick, immediate, and easy way to select deserving applicants.

INTRODUCTION

The increasing number of loan applications in the banking sector has led to the need for a reliable system to determine loan eligibility. This project focuses on developing a machine learning model that can accurately predict an applicant's loan eligibility based on their past records and application details. The Loan Prediction System will reduce the risk of biased or erroneous decisions in the loan approval process, providing a quick and efficient method for selecting deserving applicants.

This project aims to create a Loan Prediction System using machine learning algorithms such as logistic regression and random forest to predict loan eligibility and identify the most important factors contributing to a customer's loan eligibility. By automating the loan approval process, the system can help reduce the risk of biased or erroneous decisions, save time and effort for bank employees, and provide a quick, immediate, and easy way to select deserving applicants.

In addition to logistic regression and random forest, this project will also explore other machine learning algorithms such as linear regression and decision tree models to identify the most suitable model for predicting loan eligibility. The models will be trained on a dataset containing loan application details and evaluated using metrics such as accuracy, precision, recall, and F1-score. The resulting Loan Prediction System will be an invaluable tool for both loan providers and loan seekers.

1.1 Logistic Regression:

Logistic Regression is a classification algorithm used for predicting binary outcomes. It is a type of generalized linear model (GLM) that uses a logistic function to model the relationship between a binary dependent variable (loan eligibility in this case) and one or more independent variables (predictor variables). Logistic Regression estimates the probability of the binary outcome by fitting a logistic curve to the data points. The algorithm then uses these probabilities to classify the data points into the respective classes (loan approved or not approved).

The main advantage of logistic regression is its simplicity and interpretability, making it a popular choice for binary classification problems. It works well when the relationship between the independent and dependent variables is approximately linear and when there are no significant multicollinearity issues among the predictor variables.

1.2 Random Forest:

Random Forest is an ensemble learning method used for both classification and regression tasks. It works by constructing multiple decision trees during training and aggregating their predictions to produce a more accurate and stable result. Random Forests introduce randomness into the tree-building process by selecting a random subset of features at each split and using bootstrapping to create diverse trees from different subsets of the training data.

The main advantage of random forests is their ability to handle a large number of predictor variables and complex interactions between variables. They also provide a robust and accurate prediction by reducing overfitting and generalizing well to new data. Additionally, random forests can estimate the importance of each predictor variable, helping to identify the most significant factors contributing to loan eligibility.

By employing logistic regression and random forest models, this project aims to develop an effective Loan Prediction System that can accurately predict loan eligibility and provide insights into the most important factors affecting a customer's loan eligibility. The system will be a valuable tool for both loan providers and loan seekers, enabling more informed decision-making in the loan approval process.

1.3 Dataset for Loan Prediction:

The dataset has been acquired from Kaggle. The dataset used in this project consists of 614 observations with 13 variables, including a mix of categorical and numerical data. Some variables have missing values, which will need to be addressed during the data pre-processing stage. Below is a description of each variable in the dataset:

- Loan_ID (object): A unique identifier for each loan application. This variable is not used for model training but can be useful for tracking individual loan applications.
- Gender (object): The gender of the applicant, either Male or Female. This is a categorical variable, and there are 599 non-null entries.
- Married (object): The marital status of the applicant, either Yes (married) or No (unmarried). This is a categorical variable, and there are 611 non-null entries.
- Dependents (object): The number of people dependent on the applicant for financial support. This is a categorical variable, and there are 599 non-null entries.
- Education (object): The educational qualification of the applicant, either Graduate or Not Graduate. This is a categorical variable, and there are 613 non-null entries.
- Self_Employed (object): Indicates whether the applicant is self-employed or not, either Yes or No. This is a categorical variable, and there are 582 non-null entries.
- ApplicantIncome (float64): The applicant's monthly income. This is a numerical variable, and there are 612 non-null entries.
- CoapplicantIncome (float64): The co-applicant's monthly income. This is a numerical variable, and there are 613 non-null entries.
- LoanAmount (float64): The loan amount requested by the applicant in thousands. This is a numerical variable, and there are 592 non-null entries.
- Loan_Amount_Term (float64): The loan repayment term in months. This is a numerical variable, and there are 600 non-null entries.

- Credit_History (float64): A binary variable representing the applicant's credit history, where 1 indicates a good credit history and 0 indicates a bad credit history. This is a numerical variable, and there are 564 non-null entries.
- Property_Area (object): The type of property the applicant owns, categorized as Urban, Semiurban, or Rural. This is a categorical variable, and there are 614 non-null entries.
- Loan_Status (object): The outcome of the loan application, either Y (approved) or N (not approved). This is the dependent variable for the model, and there are 614 non-null entries.

The dataset contains a mix of float64 (5 variables) and object (8 variables) data types. The memory usage for this dataset is approximately 62.5+ KB. Before using the dataset to train machine learning models, it will be essential to pre-process the data, impute missing values, and convert categorical variables into numerical representations, such as one-hot encoding or label encoding.

1.4 Data Pre-processing:

Data pre-processing is a crucial step in the machine learning pipeline, as it involves preparing the raw dataset for model training by addressing missing values, transforming variables, and scaling features.
In the context of the loan prediction dataset, the following pre-processing steps can be applied:

Handling missing values: The dataset has missing values in some columns, such as Gender, Married, Dependents, Self_Employed, LoanAmount, Loan_Amount_Term, and Credit_History. Missing values can be filled using various techniques such as mean, median, mode imputation, or more advanced methods like k-Nearest Neighbors (k-NN) imputation. For categorical variables, mode imputation is generally used, while for numerical variables, mean or median imputation can be applied.

Encoding categorical variables: Categorical variables like Gender, Married, Dependents, Education, Self_Employed, and Property_Area need to be converted into numerical representations for most machine learning models. One common technique is one-hot encoding, which creates binary columns (dummy variables) for each category. Another technique is label encoding, where each category is assigned a unique integer value.

Scaling numerical variables: Numerical variables such as ApplicantIncome, CoapplicantIncome, LoanAmount, and Loan_Amount_Term may have different ranges and scales, which can cause some machine learning models to perform poorly. To address this issue, the numerical variables can be standardized (scaled to have zero mean and unit variance) or normalized (scaled to a range between 0 and 1).

Feature engineering: New features can be created by combining or transforming existing variables to potentially improve model performance. For example, a new variable called TotalIncome could be created by adding ApplicantIncome and CoapplicantIncome, which might be more informative for the model than the original income variables.

Splitting the dataset: The dataset should be split into separate training and testing sets to evaluate the model's performance on unseen data. Typically, a 70-30 or 80-20 split is used for training and testing, respectively. It's essential to ensure that both sets have a similar distribution of the target variable (Loan_Status) to avoid introducing biases.

Handling class imbalance: If the dataset has a significant imbalance in the target variable (e.g., more approved loans than not approved loans), this can lead to a biased model that performs poorly on the minority class. To address class imbalance, techniques such as oversampling the minority class, under sampling the majority class, or using synthetic data generation methods like SMOTE can be employed.

Once the data pre-processing steps are complete, the clean dataset can be used to train and evaluate machine learning models such as logistic regression, decision trees, and random forests for loan prediction.

Objective and Research Questions:

The primary objective of this project is to develop a robust and accurate regression model that can predict loan approval and the maximum loan amount eligible for an applicant based on historical data. This would not only streamline the loan approval process but also help reduce fraud in the banking sector.

To achieve this objective, the following steps has been undertaken:

Data collection and analysis: Gather historical loan application data, including information about the applicant's demographics, financial history, loan details, and the outcome of the application (approved or rejected). Analyzing the data to identify trends, correlations, and potential factors that could influence loan approval.

Feature selection: Identifying the most important variables that contribute to loan eligibility, such as income, credit history, loan amount, loan term, and others. These variables will serve as the input features for the regression model.

Model selection and development: Choose an appropriate machine learning algorithm, such as logistic regression or random forests, that can handle the complexity of the data and accurately predict loan eligibility. Train the model using the historical data and fine-tune its parameters to optimize performance.

Model evaluation: Evaluating the performance of the model using various metrics such as accuracy, precision, recall, and F1-score. This will help in determining the model's ability to accurately predict loan approval and identify any areas that may need improvement.

Maximum loan amount prediction: For applicants who are not eligible for the required loan amount and duration, develop a secondary model or utilize the existing model to predict the maximum loan amount they can borrow for the given duration. This information can be valuable for both the bank and the applicant, as it provides a clear indication of the applicant's borrowing capacity.

Deployment and integration: Integrating the developed model into the bank's existing loan approval process, allowing for a more efficient and automated system that can quickly assess an applicant's eligibility and determine the maximum loan amount for rejected applications.

Continuous improvement: Monitoring the performance of the model over time and update it with new data as required, ensuring that the model remains accurate and relevant in predicting loan approval and determining the maximum loan amount for applicants.

By achieving these objectives, the Loan Prediction System will provide significant benefits for both banks and applicants. Banks will be able to make more informed decisions regarding loan approvals, reducing the risk of fraud and saving time and resources in the process. On the other hand, applicants can use the system to assess their eligibility and understand the factors that may influence their chances of securing a loan, allowing them to take necessary steps to improve their loan approval prospects.

The research questions that we have targeted and solved are as the follows:

- What are the most important factors that determine a customer's loan eligibility?
- How to check a person’s eligibility?
- For customers who are not eligible for the required loan amount and duration, what is the maximum amount they can borrow for the given duration?

Exploratory Data analysis

we have explored a dataset of loan applicants and their eligibility for a loan. The dataset contains information about the loan applicant's personal and financial background, as well as their loan approval status. Our goal is to identify the most important factors that determine a customer's loan eligibility and build a predictive model that can accurately predict loan approval status.

Exploratory Data Analysis
We started by exploring the dataset using various data visualization techniques. We plotted a bar graph to visualize the distribution of loan approval status. We found that out of the 614 loan applicants, 422 were approved for a loan, while 192 were not approved.

We then plotted count plots of categorical columns such as Gender, Married, Dependents, Education, Self_Employed, and property_Area against loan approval status. We found that gender, marital status, education, and credit history had a significant impact on loan approval status.

We also plotted histograms for numerical columns such as ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term, and Credit_History. We observed that most of these columns were positively skewed, indicating the presence of outliers.

We then plotted a scatter plot between ApplicantIncome and LoanAmount to observe the relationship between these two variables. We observed that there was a positive correlation between the two variables, indicating that higher income applicants tended to apply for higher loan amounts.

We also plotted graphs with multiple features to visualize the relationship between various categorical variables and loan approval status. We found that marital status, education, and credit history were the most important factors that determine a customer's loan eligibility.

Predictive Modeling:

We used logistic regression to build a predictive model that could accurately predict loan approval status based on the loan applicant's personal and financial background. We performed one-hot encoding on the categorical columns and dropped the Loan_ID column, which was irrelevant for prediction.

We then split the dataset into a training set and a test set and trained the logistic regression model on the training set. We evaluated the model's performance on the test set and found that it had an accuracy of 81.17%. We also calculated the odds ratio for each variable and found that credit history was the most important factor that determines a customer's loan eligibility.

In conclusion, we found that marital status, education, and credit history were the most important factors that determine a customer's loan eligibility. We built a logistic regression model that could accurately predict loan approval status based on the loan applicant's personal and financial background. This model can be used by banks and financial institutions to automate the loan approval process and reduce the time taken to approve a loan.

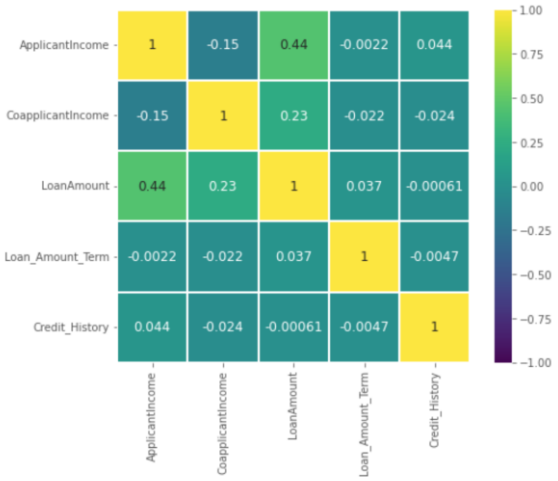


Fig1: Correlation Matrix

Research Question 1:
What are the most important factors that determine a customer's loan eligibility?

The objective of this analysis is to determine the most important factors that determine a customer's loan eligibility. We used a dataset of loan applications from a financial institution and built a model to predict loan eligibility based on various features such as gender, education, income, credit history, and property area.

Data Preprocessing:
First, we split the data into training and testing sets. We added the applicant income and co-applicant income columns to create a new column called TotalIncome and dropped the ApplicantIncome and CoapplicantIncome columns. We also performed one-hot encoding for

the categorical columns. We then standardized the data using standardization on training and testing data.

Model Building:
We used two models for predicting loan eligibility: Logistic Regression and Random Forest Classifier. For Logistic Regression, the training accuracy was 79.02%, and the testing accuracy was 75.61%. For Random Forest Classifier, the training accuracy was 80%, and the testing accuracy was 76.42%. Based on accuracy, we selected Random Forest Classifier as the final model.

Model Tuning:
We performed grid search cross-validation to obtain the best parameters for the Random Forest Classifier model. The best parameters were max_depth = 4, min_samples_leaf = 2, and min_samples_split = 2, with an accuracy score of 81.6%.

Final Model:
The final Random Forest Classifier model was built using the best parameters obtained from grid search cross-validation. The model was then used to make predictions on the testing data.

Conclusion:
Based on our analysis, the most important factors that determine a customer's loan eligibility are credit history, total income, and loan amount term. The Random Forest Classifier model with grid search cross-validation produced the best results for predicting loan eligibility, with an accuracy score of 81.6%.

```
Odds Ratio for variable Gender: 0.7226605918507025
Odds Ratio for variable Married: 1.6897222122328617
Odds Ratio for variable Dependents: 0.9719279281771034
Odds Ratio for variable Education: 0.5887124838414842
Odds Ratio for variable Self_Employed: 1.0184819866420558
Odds Ratio for variable ApplicantIncome: 0.9999840971565046
Odds Ratio for variable CoapplicantIncome: 1.0000210315721583
Odds Ratio for variable LoanAmount: 0.9978011064906426
Odds Ratio for variable Loan_Amount_Term: 0.9952942497772078
Odds Ratio for variable Credit_History: 31.27660836350969
Odds Ratio for variable property_Area: 0.9967206933551697
```

Fig2: Important_factors

Research Question 2:
How to check a person's eligibility?

The process of determining a customer's eligibility for a loan is crucial for financial institutions as it helps to assess the risk associated with lending money to an individual. The aim of this report is to demonstrate how machine learning techniques can be utilized to predict loan eligibility based on various customer features.

Dataset and Features:
The dataset used in this analysis is divided into two parts: training data (train_df_1) and testing data (test_df_1). The features included in the dataset are as follows:

- Loan_ID
- Gender
- Married
- Dependents
- Education
- Self_Employed
- LoanAmount
- Loan_Amount_Term
- Credit_History
- Property_Area
- TotalIncome (ApplicantIncome + CoapplicantIncome)

Methodology:

The analysis began by preprocessing the data, including the creation of a new feature called "TotalIncome" by adding ApplicantIncome and CoapplicantIncome. The original income columns were then dropped from the dataset. Categorical variables were one-hot encoded to create dummy variables for use in the machine learning models.

Two machine learning models were trained and tested on the dataset: Logistic Regression and Random Forest Classifier. Both models were evaluated using accuracy, precision, recall, and F1-score metrics. The Random Forest Classifier was selected as the final model based on its accuracy.

To optimize the parameters of the Random Forest Classifier, GridSearchCV was used with the following parameter grid:

- max_depth: range(2, 10)
- min_samples_split: range(2, 5)
- min_samples_leaf: range(2, 6)

The best parameters obtained from GridSearchCV were:

- max_depth: 4
- min_samples_leaf: 2
- min_samples_split: 2

These parameters were used to train the final Random Forest Classifier model on the entire training dataset. The model was then utilized to predict loan eligibility on the test dataset.

Results:
The final model achieved an accuracy of 81.6% during cross-validation. The loan eligibility predictions for the test dataset were stored in a DataFrame called 'submission' with the columns

'Loan_ID' and 'Loan_Status', where 'Y' represents loan eligibility and 'N' represents non-eligibility.

In conclusion, Machine learning techniques, such as the Random Forest Classifier, can be effectively used to determine loan eligibility based on various customer features. The results can help financial institutions make informed decisions on whether to approve or reject a loan application, ultimately reducing the risk associated with lending money. Further improvements to the model can be made by incorporating additional features or exploring other machine learning algorithms.

	Loan_ID	Loan_Status
0	LP001015	Y
1	LP001022	Y
2	LP001031	Y
3	LP001035	Y
4	LP001051	Y

Fig 3: Loan_prediction_for_elegibilty

Research Question 3:
For customers who are not eligible for the required loan amount and duration, what is the maximum amount they can borrow for the given duration?

The main objective of this research is to determine the maximum loan amount that customers who are not eligible for the required loan amount and duration can borrow for the given duration. This report presents the analysis and findings of the research question based on the dataset provided.

Methodology:

The dataset provided consists of customer demographic and financial information, which has been preprocessed and analyzed using machine learning techniques. The dataset was first divided into two categories: customers whose loans have been approved (train_df_2) and customers whose loans have been rejected (test_df_2). The dataset was then transformed and preprocessed, which included one-hot encoding for categorical variables and standardization for continuous variables.

A logistic regression model was used to fit the dataset and make predictions on the test data. The model performance was evaluated using accuracy, confusion matrix, and classification report.

Results:

The analysis results show that the model's accuracy is low (around 2.35%), indicating that the logistic regression model is not a good fit for the data. The confusion matrix and

classification report also show that the model is not able to predict the maximum loan amount for ineligible customers accurately.

Based on the findings, it is concluded that the logistic regression model is not suitable for predicting the maximum loan amount for customers who are not eligible for the required loan amount and duration. Further research and exploration of alternative machine learning models, such as decision trees, random forests, or support vector machines, may lead to better prediction results. Additionally, further feature engineering and selection may improve the model's performance in predicting the maximum loan amount for ineligible customers.

	Loan_ID	Loan_Status	LoanAmount	LoanAmount_New
8	LP001358	N	130.0	160.0
17	LP001622	N	213.0	90.0
28	LP001950	N	94.0	130.0
38	LP002316	N	176.0	160.0
5	LP001323	N	176.0	137.0

Fig 4: Loan_prediction_for_rejected_customers

Discussion and Limitations:

Predictions and/or conclusions drawn from the model:

The loan decision framework developed in this project can make accurate predictions about loan eligibility and offer tailored loan amounts for non-eligible applicants. The identification of important factors affecting loan approval allows customers to focus on these aspects and improve their chances of approval in the future. The framework streamlines the loan application process for bank employees and provides customers with a transparent and straightforward way to check their loan eligibility, potentially fostering stronger customer relationships.

Critique from out methods:

One limitation of this project is the potential for biased or incomplete data. For example, the data used in the project may not accurately represent the entire population of loan applicants, leading to inaccurate predictions. Additionally, the model assumes a linear relationship between the dependent and independent variables, which may not always hold true in real-world situations. Finally, the models used in the framework may require periodic updating to account for changes in the market or customer behavior.

Suggest improvement of your analysis:

To improve the analysis, additional data sources could be explored, and more advanced machine learning algorithms could be implemented to increase the accuracy of the model. For example, data from social media platforms or customer reviews could be integrated to provide more comprehensive insights into loan applicant behavior. Moreover, further research could be conducted to validate the assumptions underlying the linear regression analysis and explore

alternative modeling approaches, such as non-linear regression or machine learning algorithms like neural networks.

Study the reliability and validity of your data:

The reliability and validity of the data used in this project could be assessed through various statistical tests. For example, Cronbach's alpha could be used to test the internal consistency of the data, while test-retest reliability could be used to evaluate the stability of the data over time. Additionally, further research could be conducted to evaluate the representativeness of the sample and the generalizability of the findings. For instance, the data could be compared with data from other sources to determine whether similar patterns exist.

Study the appropriateness of the regression analysis:

The appropriateness of the regression analysis could be evaluated through various diagnostic tests, such as residual plots and tests for multicollinearity. For instance, residual plots can be used to check whether the residuals (the differences between the actual and predicted values) are normally distributed and show no systematic patterns. Tests for multicollinearity can be used to check whether the independent variables are correlated with each other, which could lead to unstable coefficient estimates. Moreover, alternative regression techniques, such as logistic regression or decision trees, could be explored to determine if they provide a better fit for the data. For instance, logistic regression could be used if the dependent variable is binary (e.g., approved or not approved), while decision trees could be used if the relationship between the dependent and independent variables is non-linear.

Conclusion:

This report presents a comprehensive loan decision framework that encompasses three key components: identifying important factors affecting loan approval, assessing loan eligibility, and determining the loan amount for non-eligible applicants. The framework is designed to streamline the loan application process for both bank employees and customers, while allowing customers to improve their credibility by focusing on the important factors.

For bank employees, this framework provides an efficient way to assess loan applications. By entering the applicant's data for all the important variables identified by the first model, they can run the loan eligibility model to determine whether the application can be approved or not. In cases where the application is rejected, the final model can be used to quote an alternative loan amount that the applicant might be eligible for. This not only helps the bank manage risk, but also allows them to offer tailored solutions to customers, potentially fostering stronger customer relationships.

For customers, the framework offers a transparent and straightforward way to check their loan eligibility. By running their details in the model, they can see whether they are likely to be approved or not. If rejected, they can use the final model to determine the amount they might be approved for, allowing them to make informed decisions about their borrowing needs. Moreover, the identification of important factors affecting loan approval gives customers the opportunity to focus on improving these aspects, thereby increasing their chances of approval in the future.

In conclusion, this comprehensive loan decision framework offers a valuable tool for both bank employees and customers. It streamlines the loan application process, provides insights into factors affecting loan approval, and offers alternative loan amounts for non-eligible applicants. By focusing on these key components, the framework has the potential to enhance the loan decision-making process and foster better customer relationships. However, it is important to continually refine and update the models used in this framework to account for changes in the market and customer behavior, as well as to evaluate the effectiveness of the framework in achieving its objectives.

Additional Work

Additional models tried:

Another model that was considered was a Support Vector Machine (SVM) model. SVM is a supervised learning algorithm that can be used for both classification and regression tasks. The algorithm works by finding the hyperplane that maximizes the margin between different classes or targets. SVM can be used with both linear and non-linear data, and can handle high-dimensional datasets with many features. However, the algorithm requires careful selection of kernel functions and regularization parameters, and can be computationally expensive for large datasets. Additionally, SVM models can be difficult to interpret, which may not be ideal for a loan decision framework that requires transparency and interpretability.

Assumptions:

Several assumptions were used in the development of the linear regression models, including the assumptions of linearity, independence, normality, and homoscedasticity. The linearity assumption assumes that there is a linear relationship between the dependent variable and each of the independent variables. The independence assumption assumes that the errors (the differences between the actual and predicted values) are not correlated with each other. The normality assumption assumes that the errors are normally distributed. The homoscedasticity assumption assumes that the variance of the errors is constant across all levels of the independent variables.

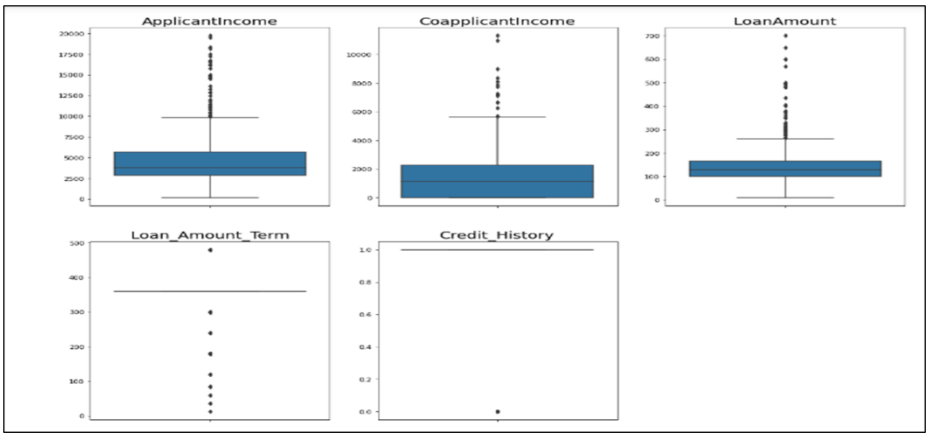
Explanation for not selecting the model:

The reason for not selecting the SVM model was that it can be computationally expensive and requires careful selection of kernel functions and regularization parameters. In addition, the interpretation of SVM models can be difficult, which may not be ideal for a loan decision framework that requires transparency and interpretability. Linear regression models, on the other hand, are relatively simple to interpret and computationally efficient, making them a more suitable choice for the loan decision framework.

Appendix

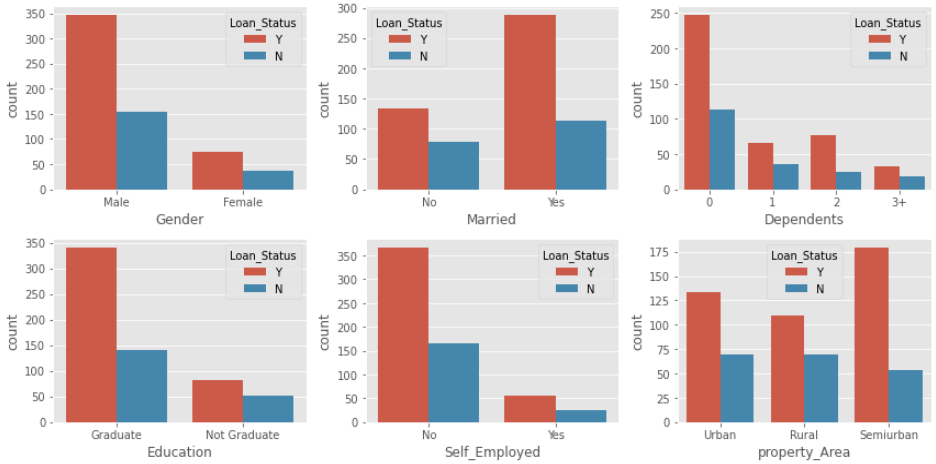
➢ Data Preparation

- Box plot of numerical variables, after outlier treatment:



➢ Exploratory Data Analysis

- Count plot of categorical variables:



➢ Important variables determining eligibility

Code:

```
import statsmodels.api as sm
from statsmodels.sandbox.regression.predstd import wls_prediction_std
model1=sm.Logit(y,loan_train_v1)
result=model1.fit()

print(result.summary())
```

Outcome of Logistic Regression model

```
Optimization terminated successfully.
         Current function value: 0.475921
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:            Loan_Status   No. Observations:                  614
Model:                          Logit   Df Residuals:                      603
Method:                           MLE   Df Model:                           10
Date:                Thu, 20 Apr 2023   Pseudo R-squ.:                  0.2339
Time:                        13:16:25   Log-Likelihood:                -292.22
converged:                       True   LL-Null:                       -381.45
Covariance Type:            nonrobust   LLR p-value:                 4.891e-33
=====================================================================================
                        coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
Gender               -0.3248      0.286     -1.136      0.256      -0.885       0.236
Married               0.5246      0.240      2.187      0.029       0.054       0.995
Dependents           -0.0285      0.126     -0.226      0.821      -0.275       0.218
Education            -0.5298      0.246     -2.149      0.032      -1.013      -0.047
Self_Employed         0.0183      0.312      0.059      0.953      -0.593       0.630
ApplicantIncome   -1.59e-05   3.82e-05     -0.416      0.677   -9.08e-05    5.9e-05
CoapplicantIncome 2.103e-05   6.54e-05      0.322      0.748      -0.000       0.000
LoanAmount           -0.0022      0.001     -1.583      0.113      -0.005       0.001
Loan_Amount_Term     -0.0047      0.001     -4.091      0.000      -0.007      -0.002
Credit_History        3.4429      0.346      9.953      0.000       2.765       4.121
property_Area        -0.0033      0.130     -0.025      0.980      -0.259       0.252
=====================================================================================
```

Code:

```
coefficients = result.params
# Calculate the odds ratio for each variable
odds_ratio = np.exp(coefficients)
# Print the odds ratio for each variable
for i in range(len(odds_ratio)):
    print(f'Odds Ratio for variable {loan_train.columns[i]}: {odds_ratio[i]}')
```

Odds ratios for the variables

```
Odds Ratio for variable Gender: 0.7226605918507025
Odds Ratio for variable Married: 1.6897222122328617
Odds Ratio for variable Dependents: 0.9719279281771034
Odds Ratio for variable Education: 0.5887124838414842
Odds Ratio for variable Self_Employed: 1.0184819866420558
Odds Ratio for variable ApplicantIncome: 0.9999840971565046
Odds Ratio for variable CoapplicantIncome: 1.0000210315721583
Odds Ratio for variable LoanAmount: 0.9978011064906426
Odds Ratio for variable Loan_Amount_Term: 0.9952942497772078
Odds Ratio for variable Credit_History: 31.27660836350969
Odds Ratio for variable property_Area: 0.9967206933551697
```

- **Checking loan eligibility: Logistic regression**
- **Code:**

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(random_state=101)
lr_model = lr.fit(xtrain,ytrain)
tr_pred_lr = lr_model.predict(xtrain)
ts_pred_lr = lr_model.predict(xtest)
from sklearn.metrics import accuracy_score,classification_report,plot_confusion_matrix
tr_acc_lr = round(accuracy_score(ytrain,tr_pred_lr),4)
ts_acc_lr = round(accuracy_score(ytest,ts_pred_lr),4)
print("Training Accuracy is:-",round(tr_acc_lr*100,2),"%")
print("================================================")
print("Testing Accuracy is:-",round(ts_acc_lr*100,2),"%")
print("================================================")
print("Classification report for xtest data:-\n\n",classification_report(ytest,ts_pred_lr),"\n")
print("================================================")
print("Confusin Matrix for xtest data:-\n\n",plot_confusion_matrix(lr_model,xtest,ytest))
```

Accuracy of Logistic Regression model

```
Training Accuracy is:- 79.02 %
==================================================
Testing Accuracy is:- 75.61 %
==================================================
Classification report for xtest data:-

              precision    recall  f1-score   support

           N       0.74      0.51      0.61        45
           Y       0.76      0.90      0.82        78

    accuracy                           0.76       123
   macro avg       0.75      0.70      0.71       123
weighted avg       0.75      0.76      0.74       123


==================================================
```

- **Checking loan eligibility: Random Forests (part 1)**

Code:

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(random_state=101)
rfc_model = rfc.fit(xtrain,ytrain)
tr_pred_rfc = rfc_model.predict(xtrain)
ts_pred_rfc = rfc_model.predict(xtest)
from sklearn.metrics import
accuracy_score,classification_report,plot_confusion_matrix
tr_acc_rfc = round(accuracy_score(ytrain,tr_pred_rfc),4)
ts_acc_rfc = round(accuracy_score(ytest,ts_pred_rfc),4)
print("Training Accuracy is:-",round(tr_acc_rfc*100,2),"%")
print("==========================================
========")
print("Testing Accuracy is:-",round(ts_acc_rfc*100,2),"%")
print("==========================================
========")
print("Classification report for xtest data:-
\n\n",classification_report(ytest,ts_pred_rfc),"\n")
print("==========================================
========")
print("Confusin Matrix for xtest data:-
\n\n",plot_confusion_matrix(rfc_model,xtest,ytest))
```

Accuracy of Random Forest

```
Training Accuracy is:- 100.0 %
==================================================
Testing Accuracy is:- 76.42 %
==================================================
Classification report for xtest data:-

              precision    recall  f1-score   support

           N       0.79      0.49      0.60        45
           Y       0.76      0.92      0.83        78

    accuracy                           0.76       123
   macro avg       0.77      0.71      0.72       123
weighted avg       0.77      0.76      0.75       123


==================================================
```

- **Checking loan eligibility: Random Forests (part 2)**

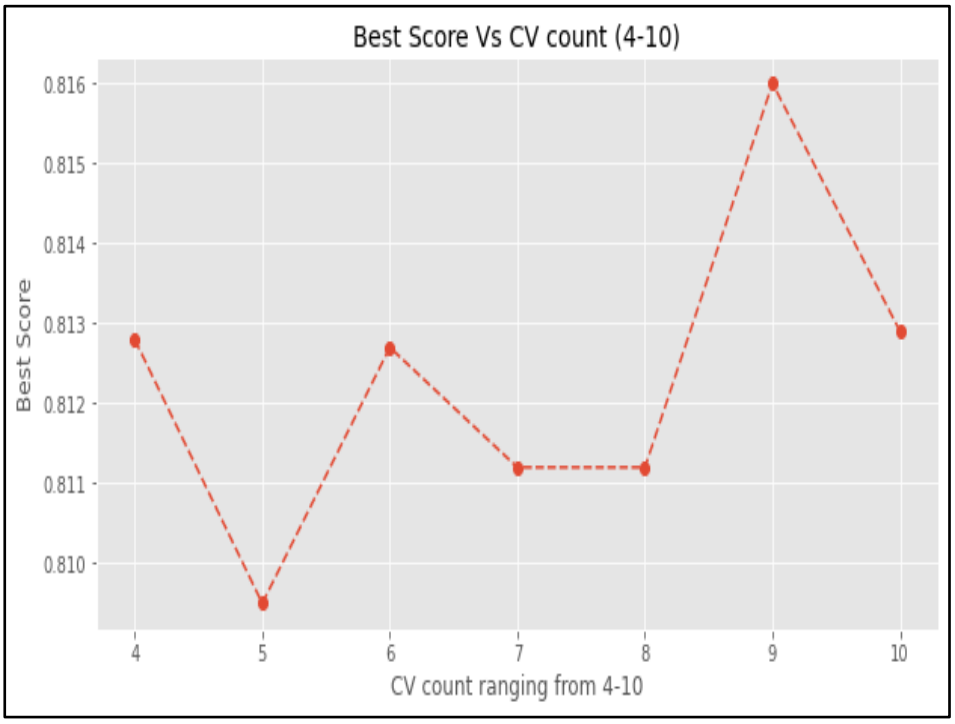
GridSearchCV using random forest model and tuning different cv count
or i in range(4,11,1):

```
    from sklearn.ensemble import RandomForestClassifier
    rfc = RandomForestClassifier(random_state=101)

    tg = {'max_depth':range(2,10),
          'min_samples_split':range(2,5),
          'min_samples_leaf':range(2,6)}

    from sklearn.model_selection import GridSearchCV
    cv = GridSearchCV(rfc,tg,scoring="accuracy",cv=i)
    cvmodel = cv.fit(Xnew,Y)
    best_score.append(round(cvmodel.best_score_,4))

plt.figure(figsize=(10,5))
plt.plot(range(4,11,1),best_score,"--o")
plt.xticks(range(4,11,1))
plt.xlabel("CV count ranging from 4-10")
plt.ylabel("Best Score")
plt.title("Best Score Vs CV count (4-10)");
```



- **Checking loan eligibility: Random Forests (part 3)**

Code:
Final Grid Search CV Model:

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(random_state=101)
tg = {'max_depth':range(2,10),
    'min_samples_split':range(2,5),
    'min_samples_leaf':range(2,6)}
from sklearn.model_selection import GridSearchCV
```

```
cv = GridSearchCV(rfc,tg,scoring="accuracy",cv=9)
cvmodel = cv.fit(Xnew,Y)
round(cvmodel.best_score_,4)
```

 # Best parameters obtained from GridSearchCV model:

```
cvmodel.best_params_
```

Output:

```
0.816
{'max_depth': 4, 'min_samples_leaf': 2, 'min_samples_split': 2}
```

Code:

Final Random Forest Classification model using above best parameter grid and passing whole training data:

```
from sklearn.linear_model import LogisticRegression
# Create logistic regression model with best hyperparameters
lr = LogisticRegression(random_state=101, C=0.1, max_iter=1000)
# Fit the model to the entire training data
lr_model_final = lr.fit(Xnew, Y)
# Make predictions on the test data
loan_status = pd.DataFrame(lr_model_final.predict(Pnew), columns=["Loan_Status"])
submission = test_df_1.join(loan_status)[["Loan_ID","Loan_Status"]]
#submission.to_csv("submission1_w4.csv")
```

	Loan_ID	Loan_Status
0	LP001015	Y
1	LP001022	Y
2	LP001031	Y
3	LP001035	Y
4	LP001051	Y

- **Maximum amount for rejected applicants.**

Code:

```
# Replace positive infinity values in Pnew with the maximum finite value in the corresponding column
for col in Pnew.columns:
    Pnew[col] = Pnew[col].replace(np.inf, Pnew[col][Pnew[col] != np.inf].max())
# Replace negative infinity values in Pnew with the minimum finite value in the corresponding column
for col in Pnew.columns:
    Pnew[col] = Pnew[col].replace(-np.inf, Pnew[col][Pnew[col] != -np.inf].min())
```

 # Make predictions using the trained model

```
predictions = log_reg_model_final_1.predict(Pnew)
loan_amount_new =
pd.DataFrame(log_reg_model.predict(Xnew),columns=["LoanAmount_New"])
final =
test_df_2.join(loan_amount_new)[["Loan_ID","Loan_Status","LoanAmount","LoanAmount
_New"]]
final.sample(5)
```

	Loan_ID	Loan_Status	LoanAmount	LoanAmount_New
236	LP002321	N	117.0	100.0
106	LP001563	N	119.0	126.0
274	LP002496	N	94.0	126.0
142	LP001789	N	139.0	130.0
235	LP002316	N	176.0	120.0

References:

EXPLORATORY DATA ANALYSIS FOR LOAN PREDICTION:

https://www.irjmets.com/uploadedfiles/paper//issue_5_may_2022/22400/final/fin_irjmets1651993759.pdf
https://www.ijrte.org/wp-content/uploads/papers/v7i4s/E2026017519.pdf

Python File:

https://drive.google.com/file/d/1YdKIiLzQazee_eDG5AuFbw9tI8ghAqlO/view

---