SKILL VERTEX

MINOR PROJECT

-

LOAN PREDICTION MODEL

NAME: ARUNKUMAR A

BATCH: APRIL-2023

MAIL ID: arunkumarabinesh@gmail.com

MOBILE NO.: 7448449449

REPORT FOR LOAN PREDICTION MODEL

INTRODUCTION:

Financial institutions need to accurately predict loan outcomes to mitigate risk and improve operational efficiency. In this loan prediction report, we present the results of a project that aims to predict loan approval using various features, including credit history, loan amount, gender, education, marital status, and employment status. Our goal is to build a predictive model that can help these institutions make informed decisions on whether to approve or reject a loan application.

To achieve this goal, we first describe the data set used in this project, its size, structure, and features. We then detail the pre-processing steps taken to clean and prepare the data for modelling. Next, we conduct exploratory data analysis to gain insights into the data, including uni variate, bi-variate, and multivariate analysis.After analyzing the data, we evaluate the performance of logistic regression.

Our project has significant potential to provide value to financial institutions by improving loan decision-making processes and reducing financial risks. By building accurate predictive models, we can help these institutions make more informed decisions and ultimately improve their business outcomes.

DATA DESCRIPTION:

The model has been created with two sets of data in an excel file. Those data are train data and test data. There were these two data-sets separately splitted hence there was no necessary for the usage of train_test_split() syntax. It contains data about a loan applicant that are required to analyse the loan requirement of the applicant. These data has many categories of information that would be helpful for the model to predict and give the accurate prediction whether the loan is approved or not.

Columns in train data-set:

As mentioned above, there are various categories of column that gets the information of the applicant.

The columns are Loan_ ID, Gender, Married, Dependents, Education, Self-Employed, Applicant-Income, Co-applicant-Income, Loan Amount, Loan Amount Term, Credit History, Property Area, Loan status.These data are used to train the data in the loan prediction model.

Columns in test data-set:

The following categories are the columns in test data-set.

Loan_ ID, Gender, Married, Dependents, Education, Self-Employed, Applicant-Income, Co-applicant-Income, Loan Amount, Loan Amount Term, Credit History, Property Area.

The test data-set uses the same columns except the loan status category.

APPROACH:

- The main goal of the project is to create a loan prediction model that predicts whether the loan is approved for a particular applicant considering the data in the train and test data-set.
- As a first step required libraries like pandas, numpy and sklearn are imported. Next the train data-set is loaded using read_csv() syntax.
- The data is loaded and ready for further steps. This data is now ready for pre-processing.
- Once the data is pre-processed, they are now taken as x_train and y_train for the prediction method using logistic regression algorithm.
- The model is builded for the loan prediction,next is to deploy the model to the user by collecting the data (features) from the user and predict the loan sension or loan approval.

PRE-PREPROCESSING:

Step 1:

- The first step of pre-processing is deleting the unwanted column of data from the data-set for meaningful and accurate results.
- Using drop() to drop the unwanted data from the dataset.

Step 2:

- Next step is to find the null or empty values in the dataset and fill them using the mean for numerical data and frequency (max - occurred) for categorical data.
- .isna() used to find the null values present in the dataset. Sum them up so that we could check the total number of null values present.
- .fillna() is used to fill the null values with appropriate summary in this case we use mean for numerical data and the maximum number of data present in the column for the categorical data.

Step 3:

- In this step the categorical values are changed to numerical data by using LabelEncoder from sklearn.preprocessing module.
- Data is fitted to label encoder using fit_transform().

Label encoding:

Label Encoder is a machine learning technique used to convert categorical data into numerical data. It is a process of assigning a unique numerical label to each category or class in a given dataset. This technique is often used to transform text data, such as names, countries, and colors, into numerical values that can be understood and processed by machine learning algorithms.

- fit_transform() is used to perform data preprocessing on a training dataset. The fit() method learns the parameters of the transformation (such as mean and standard deviation in the case of normalization).

- transform() method applies those learned parameters to transform the data (such as subtracting the mean and dividing by the standard deviation to normalize the data).
- By doing so, the training data is transformed into a format that is suitable for use by machine learning algorithms.

Step 4:

- Split the data into x_train,y_train.
- Standardize the x_train data with the use of StandardScaler().

StandardScaler:

Standard Scaler is a data preprocessing technique used to standardize or normalize the features in a dataset. It transforms the data in such a way that the mean of each feature becomes zero and the standard deviation becomes one.

It is important to note that Standard Scaler should be applied only to the training data and then the same transformation should be applied to the test data. This is to ensure that the test data is transformed in the same way as the training data, which will improve the accuracy and reliability of the model.

By this the pre-processing ends and the same steps are applicable to test dataset.

ALGORITHM:

Logistic Regression is the algorithm used to fit the model to get prediction value.

LOGISTIC REGRESSION:

Logistic Regression is a machine learning algorithm used for binary classification problems, where the outcome variable takes only two values (e.g., yes/no, true/false, 0/1). It is a type of regression analysis that models the probability of a certain outcome based on one or more input features.

The logistic regression model works by fitting a sigmoid function to the input data, which maps the input features to a probability between 0 and 1. The sigmoid function has an S-shaped curve and is defined as follows:

$$h(x) = 1 / (1 + e^{(-z)})$$

Where,

(x) is the predicted probability of the outcome variable being positive

x is the input features

z is the linear combination of the input features and their corresponding weights.

During training, the model uses the input features and the actual outcome values to learn the weights that best fit the sigmoid function. This is typically done using a technique called maximum likelihood estimation.

Once the model is trained, it can be used to predict the outcome for new input data by passing the input features through the sigmoid function and classifying the output as positive (1) or negative (0) based on a decision threshold.

Logistic Regression has several advantages, including its simplicity, interpretability, and efficiency in handling large datasets. However, it has some limitations, such as the assumption of linearity between the input features and the log-odds of the outcome variable, which may not always hold in practice.

To use the algorithm in the model it must be imported using sklearn library.

Implementation In Python:

Sklearn.linear_model has the logistic regression algorithm which is imported as LogisticRegression.

The code is given below

```
from sklearn.linear_model import LogisticRegression

LoR = LogisticRegression(random_state =  0)

Model = LoR.fit(x_train,y_train)
```

random_state:

The "random_state" parameter is used to ensure that the same set of random numbers are generated each time the code is run, which can help make the results more reproducible and consistent. It is often used in situations where you want to be able to reproduce the exact same results, such as when testing and evaluating different models or algorithms.

.fit():

The `.fit()` method is typically used to fit a model to the training data, which involves finding the optimal parameters of the model that minimize the difference between the predicted values and the actual values in the training data. The process of finding the optimal parameters is often referred to as "model training" or "model fitting". After the `.fit()` method is called, the model is trained on the training data and the optimal parameters are learned. Once the model is trained, it can be used to make predictions on new data using the .predict() method.

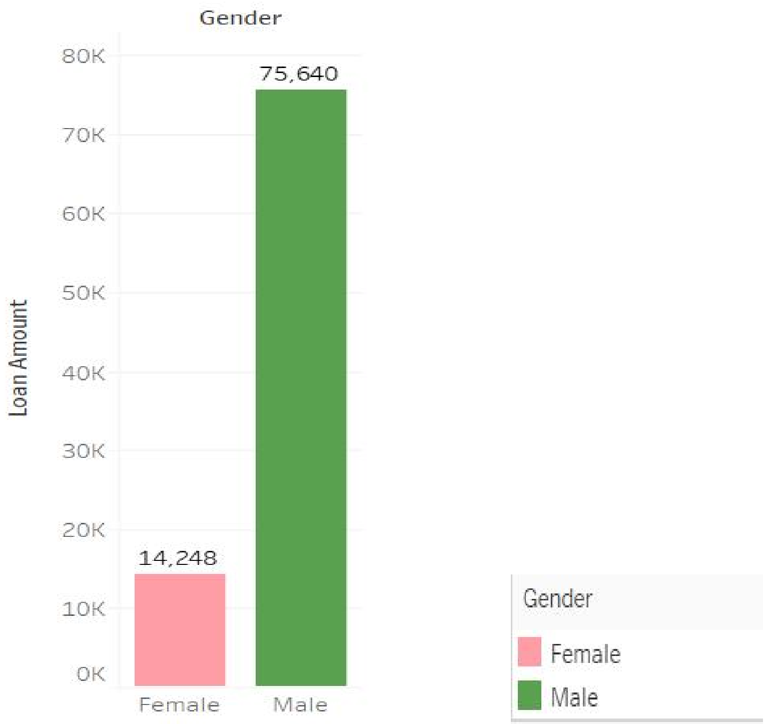
VISUALIZATION:

Visualization is a key aspect of any project report as it helps people understand more about the topic. For visualizing the data used in loan prediction model Tableau Public is used.

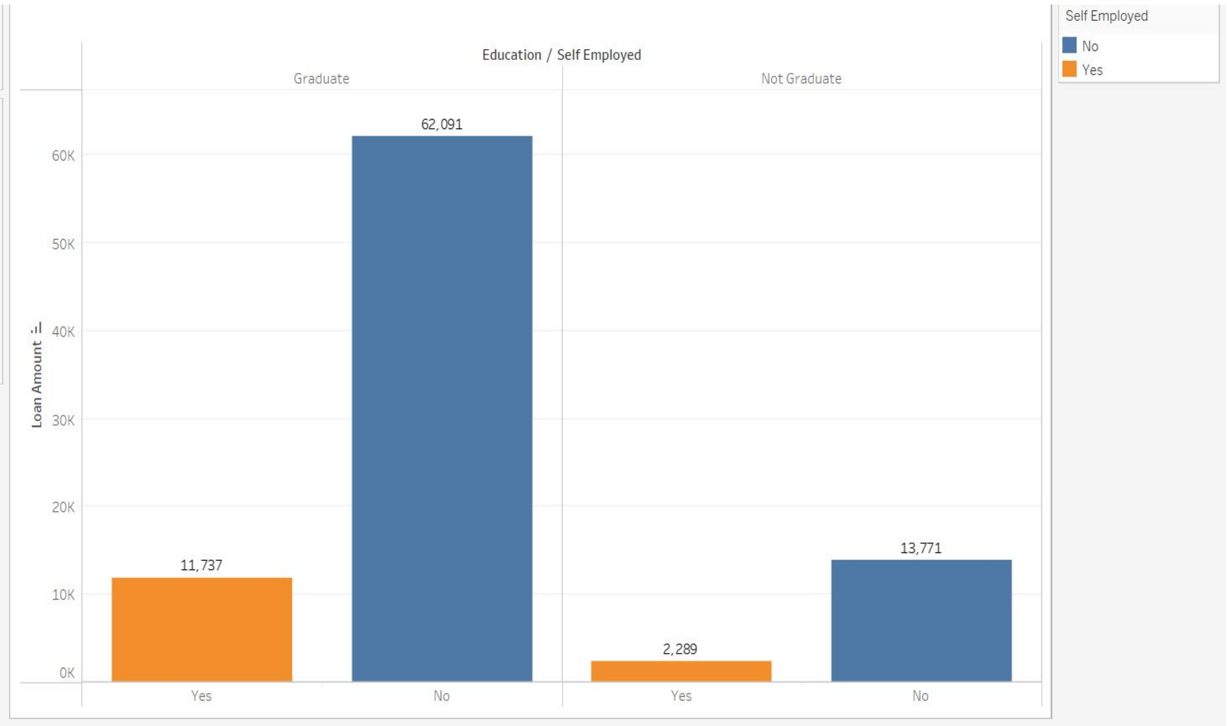
Tableau is a powerful data visualization tool that allows users to create interactive and visually appealing dashboards, reports, and charts. With Tableau, users can connect to various data sources, import and blend data, and create meaningful insights.One of the key features of Tableau is its ability to visualize data in a variety of ways. Tableau offers a wide range of charts and graphs, including bar charts, line charts, scatter plots, pie charts, heat maps, and many more. Users can also create custom charts using calculated fields and parameters.

Tableau offers a variety of tools for formatting and customizing visualizations, including color palettes, fonts, and labels. Users can also add annotations, highlight data, and create tooltips to provide additional context and information.

BAR CHART:

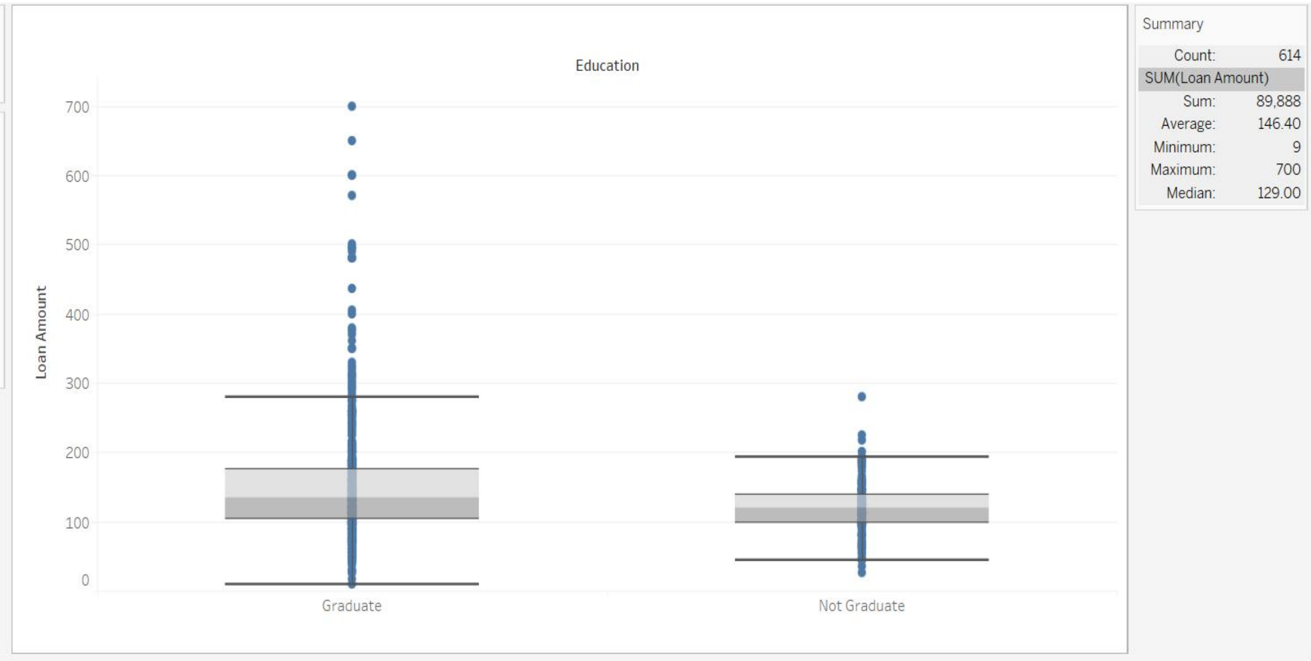


Sum of Loan Amount for each Gender. Color shows details about Gender. The marks are labeled by sum of Loan Amount. This chart shows the total loan amount of female and male in the dataset.



This chart is used to visualize the educational and self-employment details of the loan applicants. It is categorized as Graduates and Not Graduates who are self Employed.

BOX PLOT:

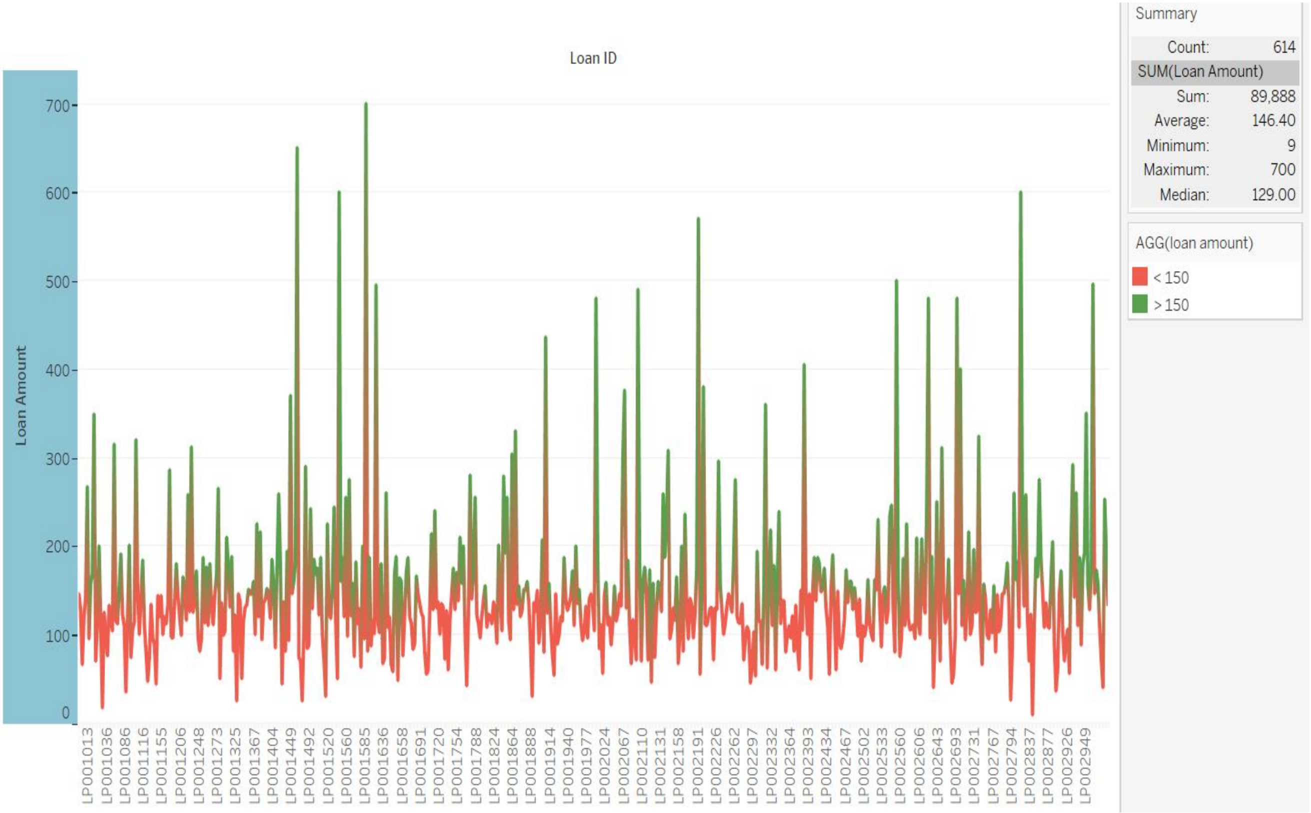


Summary

Count:	614
SUM(Loan Amount)	
Sum:	89,888
Average:	146.40
Minimum:	9
Maximum:	700
Median:	129.00

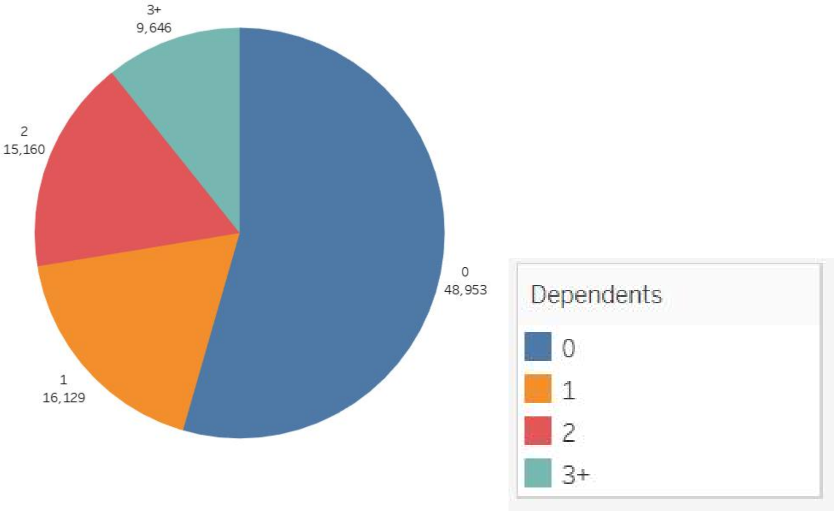
A box plot is used to show the educational status and loan amount categories. It depicts the summary of the data (i.e) SUM,AVERAGE,MINIMUM,MAXIMUM and MEDIAN. Its also useful in identifying the outliers in the data.

LINE CHART:



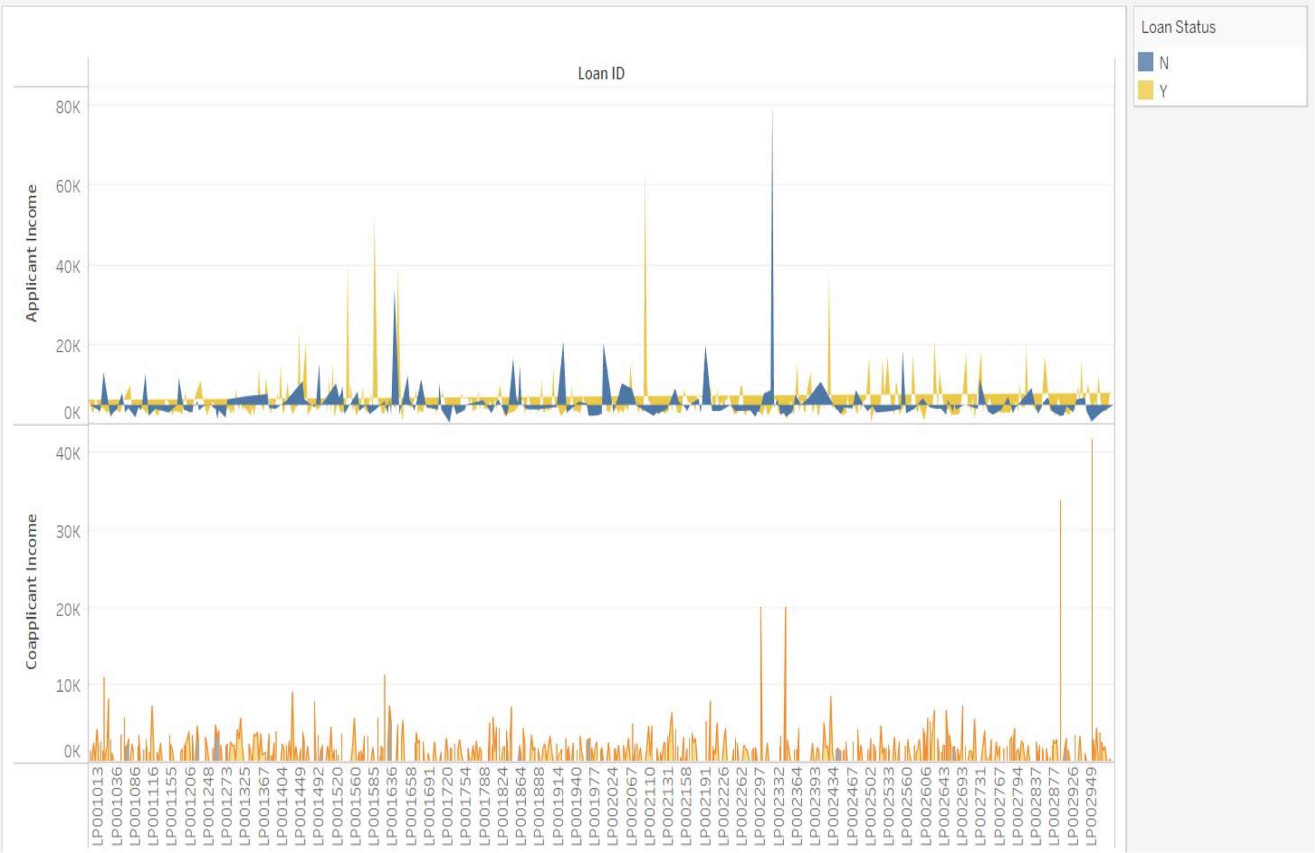
The line of sum of Loan Amount for Loan ID. Color shows details about loan amount.

PIE CHART:



Sum of Loan Amount for each Self Employed broken down by Education. Color shows details about Self Employed. The marks are labeled by sum of Loan Amount.

POLYGEN AND HISTOGRAM:



The above graph show the polygen curve of Applicant income and Loan ID with the color it shows the loan status and another graph i.e histogram it show the distribution o the co applicant income with the loan id.

CONCLUSION:

In conclusion, this study utilized logistic regression to predict loan approval outcomes. Logistic regression is a powerful statistical technique that allows us to understand the relationship between a set of independent variables and a binary dependent variable. By analyzing a dataset containing various features related to loan applications, we built a logistic regression model to predict whether a loan would be approved or not.

The visualization of the data played a crucial role in our analysis. We employed several visualization techniques to gain insights into the relationships between different variables and the loan approval status. These visualizations helped us identify patterns, trends, and potential predictors that influenced loan approval decisions.

Through our analysis, we observed that certain factors significantly influenced the likelihood of loan approval. These factors included the applicant's credit history, income, loan amount, and loan term. Visualizations such as bar charts, scatter plots, and correlation matrices provided us with a comprehensive understanding of the impact these variables had on loan approval outcomes.Our logistic regression model successfully learned the underlying patterns in the data and made accurate predictions regarding loan approval.

The findings of this study can be valuable for both lenders and loan applicants. Lenders can utilize the logistic regression model to assess the risk associated with loan applications and make informed decisions accordingly. Applicants, on the other hand, can better understand the factors that influence loan approval and work on improving their chances by focusing on aspects like maintaining a good credit history and ensuring a stable income.

In conclusion, our loan prediction analysis using logistic regression, along with the aid of visualization techniques, provided valuable insights into loan approval outcomes. By leveraging these insights, lenders can make more informed decisions, while loan applicants can increase their chances of approval by focusing on influential factors. This study highlights the importance of utilizing data-driven approaches and visualization tools in the field of loan prediction.