Self Project On
Loan Eligibility Prediction

Table of content

3. **Data pre-processing**
    - Understanding Data
    - Univariate Analysis
    - Handling Missing Values
    - Handling Outliers
    - One-Hot Encoding
    - Label Encoding
    - Standardization
    - MaxAbsolute Scaling

4. **Model Building**
    - Logistic Regression
    - Linear Discriminant Analysis
    - Quadratic Discriminant Analysis
    - K-Nearest Neighbours
    - Naïve Bayes
    - Support Vector Machine
    - Decision Tree
    - Random forest
    - Voting Ensemble

5. **Model Comparison**

6. **Conclusion**

7. **Future Possibilities**

Objective:

The primary objective of this project is to develop a robust machine learning model for binary classification, specifically focused on predicting loan eligibility. The model will analyse various applicant features and historical loan data to determine the likelihood of an applicant's loan approval. Additionally, the project seeks to contribute to the financial industry's efforts in streamlining loan processing and decision-making, ultimately benefiting institutions and applicants.

Dataset Description:

The loan eligibility dataset contains information about loan applicants and whether their loan applications were approved or denied. The dataset consists of 13 columns and 614 rows, with each row representing a single loan application.

Source: Dream Housing Finance Company

Dataset link: https://datahack.analyticsvidhya.com/contest/practice-problemloan-prediction-iii/#ProblemStatement

The columns in the dataset include:

S.No	Variable	Description
1	Loan_ID:	Unique Loan ID
2	Gender:	the gender of the applicant (male or female)
3	Married:	Applicant married (Y/N)
4	Dependents:	Number of dependents (0,1,2, or 3+)
5	Education:	Applicant Education (Graduate/ Under Graduate)
6	Self_Employed:	Self employed (Y/N)
7	ApplicantIncome:	the income of the applicant
8	CoapplicantIncome:	the income of the co-applicant (if any)
9	LoanAmount:	Loan amount in thousands
10	Loan_Amount_Term:	Term of loan in months
11	Credit_History:	credit history meets guidelines (Y/N)
12	Property_Area:	Urban/ Semi Urban/ Rural
13	Loan_Status:	(Target) Loan approved (Y/N)

Data Pre-processing

Understanding Data:

```
<class 'pandas.core.frame.DataFrame'> RangeIndex:
614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
0   Loan_ID            614 non-null    object
1   Gender             601 non-null    object
2   Married            611 non-null    object
3   Dependents         599 non-null    object
4   Education          614 non-null    object
5   Self_Employed      582 non-null    object
6   ApplicantIncome    614 non-null    int64
7   CoapplicantIncome  614 non-null    float64
8   LoanAmount         592 non-null    float64
9   Loan_Amount_Term   600 non-null    float64
10  Credit_History     564 non-null    float64
11  Property_Area      614 non-null    object
    12  Loan_Status        614 non-null
    object  dtypes: float64(4), int64(1),
    object(8) memory usage: 62.5+ KB
```

- The dataset contains 614 rows and 13 columns.
- Some of the columns have missing values, as indicated by the non-null count for each column being less than 614. Specifically, the columns with missing values are Gender, Married, Dependents, Self_Employed, LoanAmount, Loan_Amount_Term, and Credit_History.
- The Gender, Married, Dependents, Education, Self_Employed, Property_Area, and Loan_Status columns are categorical variables and are of object type.
- The ApplicantIncome column is a numerical variable and is of int64 type.
- The CoapplicantIncome, LoanAmount, Loan_Amount_Term, and Credit_History columns are also numerical variables but are of float64 type due to the presence of decimal values.
- The memory usage of the dataset is relatively small, at 62.5 KB, which suggests that the dataset is not very large and can be easily loaded into memory for analysis.
- The presence of missing values in the dataset will need to be addressed before any statistical or analytical methods are applied, as missing values can affect the accuracy of results and conclusions drawn from the data.

Variables Description:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Loan_ID	614	614	LP001002	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gender	601	2	Male	489	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Married	611	2	Yes	398	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Dependents	599	4	0	345	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Education	614	2	Graduate	480	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Self_Employed	582	2	No	500	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ApplicantIncome	614.0	NaN	NaN	NaN	5403.459283	6109.041673	150.0	2877.5	3812.5	5795.0	81000.0
CoapplicantIncome	614.0	NaN	NaN	NaN	1621.245798	2926.248369	0.0	0.0	1188.5	2297.25	41667.0
LoanAmount	592.0	NaN	NaN	NaN	146.412162	85.587325	9.0	100.0	128.0	168.0	700.0
Loan_Amount_Term	600.0	NaN	NaN	NaN	342.0	65.12041	12.0	360.0	360.0	360.0	480.0
Credit_History	564.0	NaN	NaN	NaN	0.842199	0.364878	0.0	1.0	1.0	1.0	1.0
Property_Area	614	3	Semiurban	233	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Loan_Status	614	2	Y	422	NaN	NaN	NaN	NaN	NaN	NaN	NaN

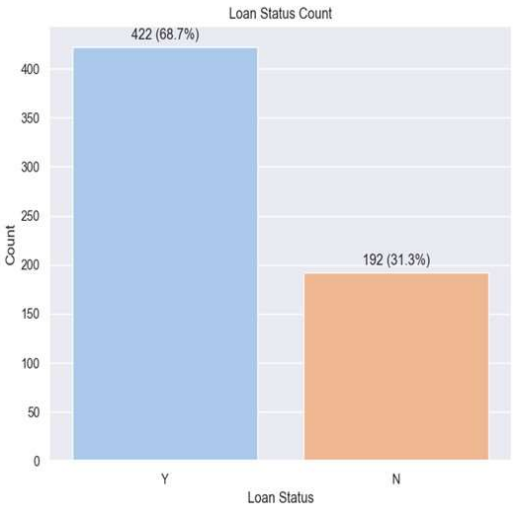
Based on the above description we can make the following observations:

- The ApplicantIncome column has a mean of 5403.46 and a standard deviation of 6109.04, suggesting that there is a wide range of incomes in the dataset.
- The CoapplicantIncome column has a mean of 1621.25 and a standard deviation of 2926.25, suggesting that there is also a wide range of coapplicant incomes in the dataset.
- The LoanAmount column has a mean of 146.41 and a standard deviation of 85.59, suggesting that there is a wide range of loan amounts in the dataset.
- The Loan_Amount_Term column has a mean of 342.0 and a standard deviation of 65.12, suggesting that most loan terms are around 360 months (30 years). • The Credit_History column has a mean of 0.84 and a standard deviation of 0.36, suggesting that most loan applicants have a credit history (i.e., a credit score).

Univariate Analysis:

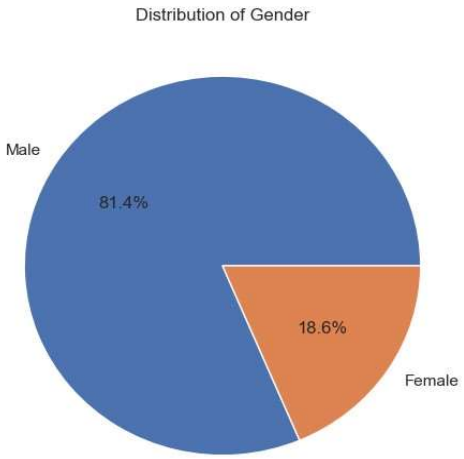
Univariate analysis is a statistical technique that involves the analysis of a single variable in a dataset. It is a simple yet powerful way to explore the characteristics of a single variable and gain insights into its distribution, central tendency, and variability. In univariate analysis, descriptive statistics such as mean, median, mode, standard

deviation, and range are used to summarize the data and provide insights into the data distribution. Univariate analysis is a critical step in the data analysis process and can provide valuable insights into the dataset, which can be used in further analysis and decision-making. Let's start with categorical variables.



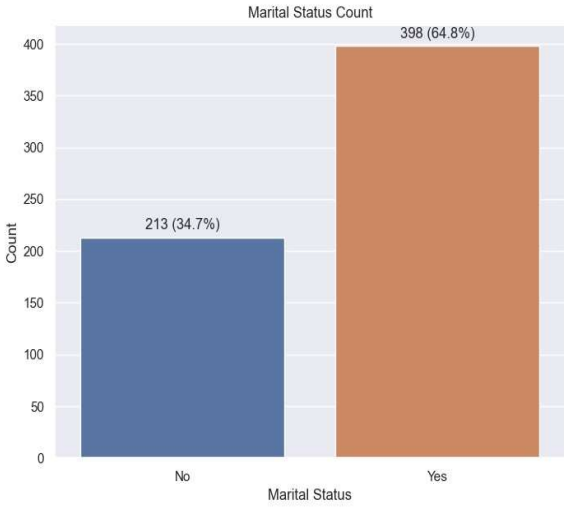
Based on the countplot result, we can observe the following:

- The majority of the loan applications were approved, as there are 422 instances of "Y" (yes) in the "Loan_Status" column.
- The number of rejected loan applications is lower than the number of approved applications, as there are 192 instances of "N" (no) in the "Loan_Status" column.

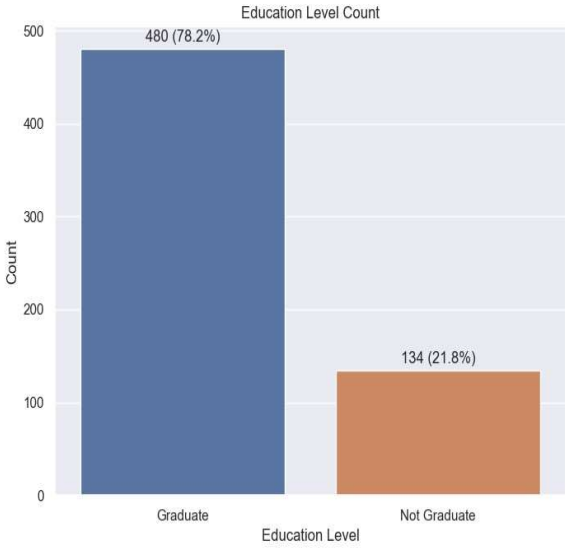


Based on the pie plot result, we can observe the following:

- The majority of loan applicants in this dataset are male, accounting for around 81% of the total applicants.
- Female loan applicants make up only around 18.6% of the total applicants in the dataset.

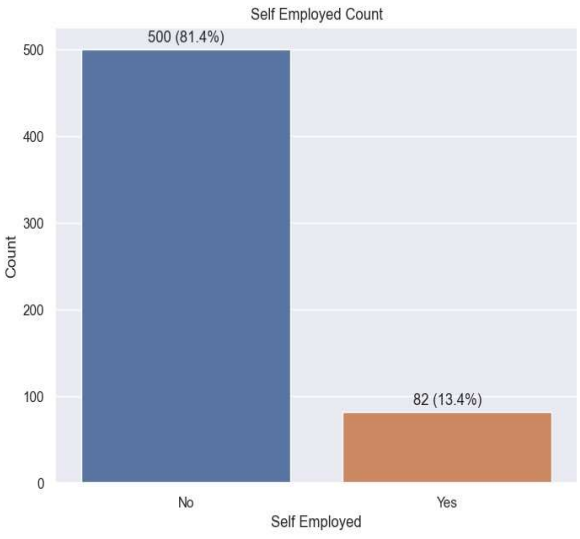


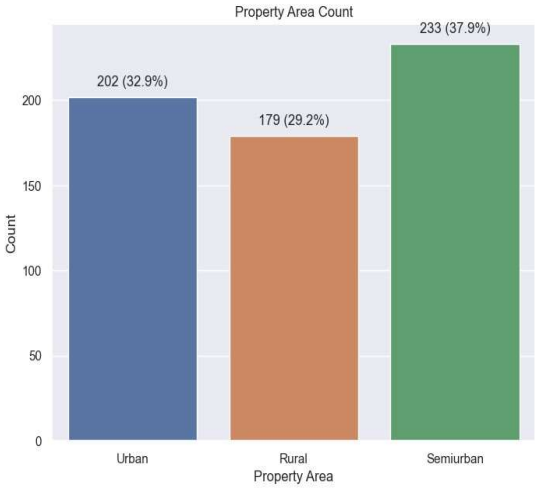
The bar plot shows the count of married and unmarried individuals in the loan eligibility dataset. We can observe that a majority of individuals in the dataset are married, with a count of 398. Unmarried individuals have a count of 213. This suggests that married individuals are more likely to apply for a loan than unmarried individuals.



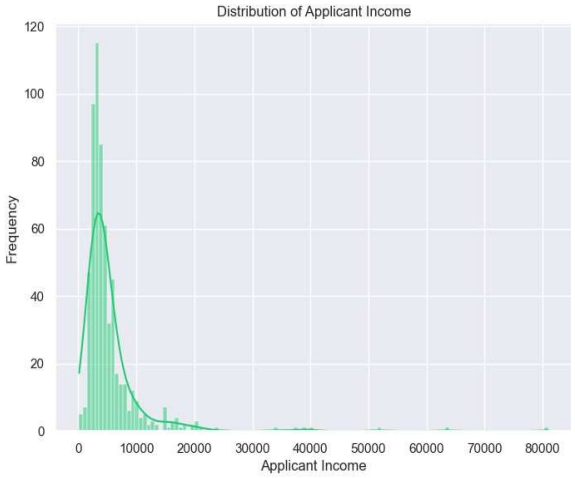
The bar plot shows that the majority of the applicants are graduates with a count of 480, whereas the count of applicants who are not graduates is 134. This indicates that a higher number (78.2%) of graduates apply for loans compared to those who are not graduates.

The bar plot of the Self_Employed column shows that out of the 582 applicants, 500 applicants are not selfemployed, while only 82 applicants are self-employed. This suggests that the majority (81.4%) of the loan applicants are not self-employed.

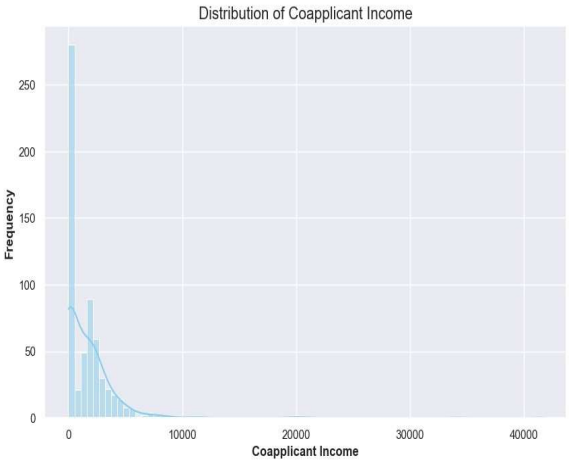




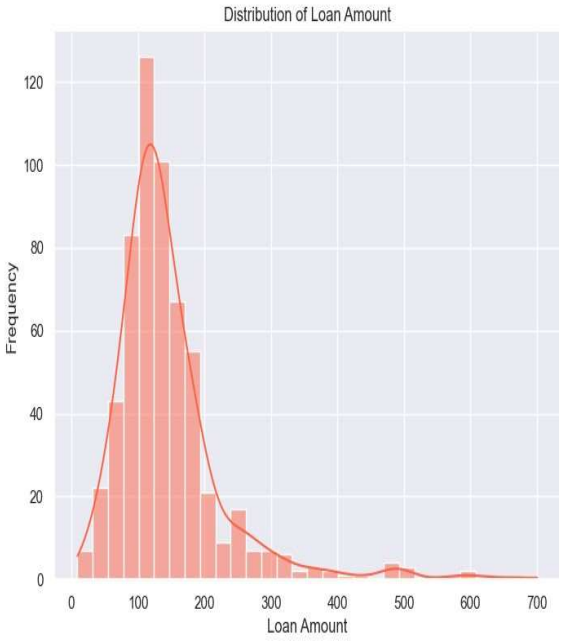
Based on the result "Semiurban 233, Urban 202, Rural 179", we can observe that the semiurban area has the highest 37.9% count of property followed by urban and rural areas respectively.



1. The distribution of applicant income is right-skewed, indicating that a majority of applicants have incomes ranging from 0 to 25,000.
2. There are a few extreme outliers with incomes exceeding 80,000. These outliers represent a small fraction of the total applicants and can be considered as exceptional cases.



1. The majority of applicant's coapplicant income is very low, indicating that they are primarily relying on their own income for the loan application.
2. This distribution is heavily rightskewed with a few outliers on the right side. This indicates that there are some cases with significantly higher co-applicant income.



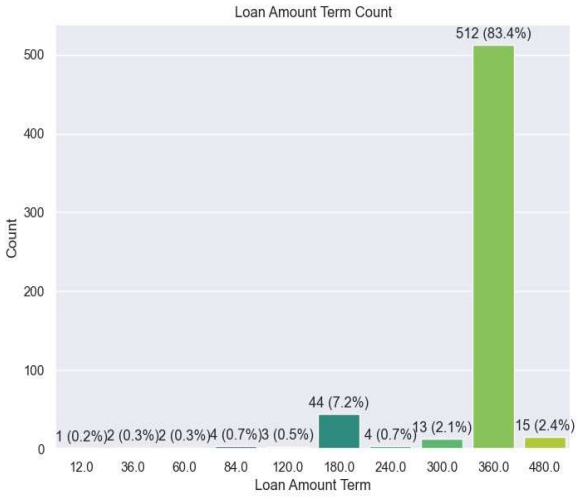
1.The distribution is right-skewed, which means that the majority of the loan amounts are towards the lower end of the scale.

2.The plot shows some extreme values, indicating that some individuals have taken out very large loans.

3.The plot also shows a peak around the 120-140 range, indicating that a significant number of individuals have taken out loans in that range.

4.The plot also shows a long tail towards the higher end of the scale, indicating that while most loans are

small, there are some loans that are much larger than the average.



- The most common loan amount term is 360 months (or 30 years), with 512 instances in the dataset.
- The next most common loan amount terms are 180 months (or 15 years), 480 months (or 40 years), and 300 months (or 25 years).
- We can conclude that most borrowers in this dataset prefer a loan amount term of 30 years.

Handling Missing Values:

NUMBER OF MISSING VALUES IN THE DATASET:

```
Credit_History      50
Self_Employed       32
LoanAmount          22
Dependents          15
Loan_Amount_Term    14
```

```
Gender                13
Married                3
Loan_ID                0
Education              0
ApplicantIncome        0
CoapplicantIncome      0
Property_Area          0
Loan_Status            0
      dtype: int64
```

Handling missing values is an essential step in data pre-processing. In many real-word datasets, it is common to have missing values in the dataset, and it is crucial to hande them appropriately to avoid biased or incorrect analysis.

Missing values can arise due to various reasons such as data entry errors, incomplete data, or certain values not being applicable to some instances. In the loan eligibility dataset, we have already identified missing values in some columns.
To handle these missing values, we can Imputed the missing values in columns [Credit_History, Self_Employed, Dependents, Loan Amount term, Gender, Married] by Mode,
We found that 22 loan applications have missing information for the 'Loan Amount'. Since this information is crucial for our analysis, it's best to handle these cases. One option is to remove these 22 applications from our dataset. This ensures that we're working with complete and reliable data.

Handling Outliers:

Outliers, or data points significantly different from the rest of the dataset, can have a notable impact on the performance of machine learning models. In the context of banking data, outliers may arise due to various factors, such as unusual financial transactions or errors in data entry. It is crucial to address outliers appropriately to ensure the robustness and accuracy of our predictive models.

While removing outliers can help in achieving a more accurate model, it is important to acknowledge that in banking data, outliers may carry meaningful information. For instance, an unexpected high-value Loan_Amount might be indicative of a High ApplicantIncome for a customer.

Certain classification algorithms are more sensitive to outliers than others. For example:

Logistic Regression: This algorithm assumes a linear relationship between features and may be heavily influenced by outliers.

K-Nearest Neighbours (KNN): KNN can be significantly affected by outliers since it relies on the distance between data points

On the other hand, classification algorithms such as Support Vector Machines (SVM), Naive Bayes, Decision Trees, and Random Forests tend to be less influenced by the presence of outliers.

I have applied one-hot encoding to handle categorical variables, specifically by removing the first category. This approach helps address multicollinearity in the dataset.

One-Hot Encoding:

One-hot encoding is a technique used in data pre-processing to handle categorical variables, transforming them into a numerical format that machine learning algorithms can work with effectively. This method is especially useful when dealing with categorical features that don't have an inherent ordinal relationship.

Encoding Categorical Variables and Label Encoding

In the pre-processing phase, I have applied one-hot encoding to handle categorical variables. This technique converts categorical features into binary columns, ensuring that the model can effectively interpret and utilize them. To avoid the multicollinearity issue, I have excluded the first category, which serves as the reference category.

Label Encoding:

Label encoding is a method to convert Target variable into numerical format. In my project, 'Y' (for approved loans) has been encoded as 1, and 'N' (for unapproved loans) as 0.

Standardization:

Standardization, also known as Z-score normalization, is a pre-processing technique used to transform numerical features to have a mean of 0 and a standard deviation of 1. This process helps in bringing all features to a similar scale, which is crucial for models that rely on distance measures, such as Support Vector Machines and kNearest Neighbours.

By standardizing the data, we ensure that no single feature dominates the learning process due to its larger magnitude. This allows the model to effectively learn from all features equally. In our case, standardizing features like 'ApplicantIncome', 'Loan_Amount_Term' and 'LoanAmount' ensures they contribute equally to the model's learning process, leading to a more balanced and accurate prediction.

Max Absolute Scaling:

Max Absolute Scaling, also known as Max-Min Scaling, is a method of pre-processing numerical data. It scales the features to a range between 0 and 1 by dividing each value by the maximum absolute value in the feature.

The formula for Max Absolute Scaling is:

$$X_{scaled} = X_i / \max(|X|)$$

This technique can be useful when there is more zero in the feature.

In our case, applying Max Absolute Scaling to features like 'ApplicantIncome' can help in effectively utilizing their information for the learning process, leading to improved model performance.

Models Building

In order to assess the performance of the machine learning model, the dataset was divided into two subsets: a training set and a testing set. The training set, comprising 80% of the data, was used to train the model, allowing it to learn patterns and relationships within the data. The remaining 20% constituted the testing set, which was reserved to evaluate the model's performance on unseen data. This approach helps ensure the model's generalization ability to new, unseen observations.

Evolution Metric:

In Scikit-learn, the inbuilt confusion matrix is structured as follows:

		Predicted	
		0	1
Actual	0	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

Precision is calculated as:

Precision = TP / (TP + FP)

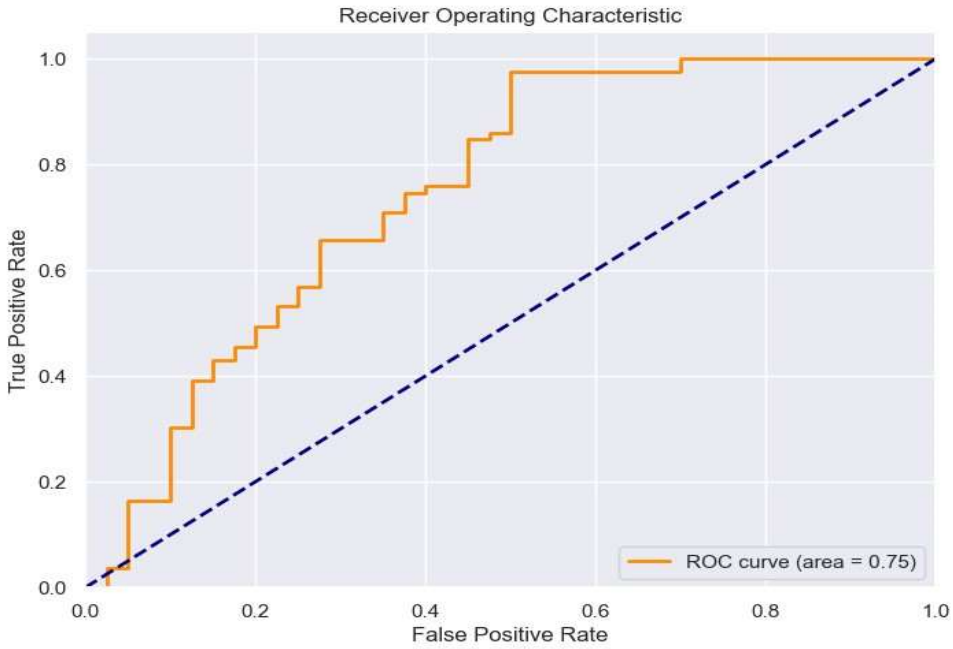
Recall (Sensitivity) is calculated as:

Recall = TP / (TP + FN)

In the process of training and selecting our model, our primary focus will be on minimizing false positives (FP) — instances where an applicant is incorrectly deemed eligible for a loan and High F1 Score. This strategic approach is aimed at reducing financial risk by ensuring that individuals who are not qualified do not receive loans. Consequently, our focus will be on achieving high precision, a metric that measures the accuracy of positive predictions and is crucial for this specific task.

Logistic Regression

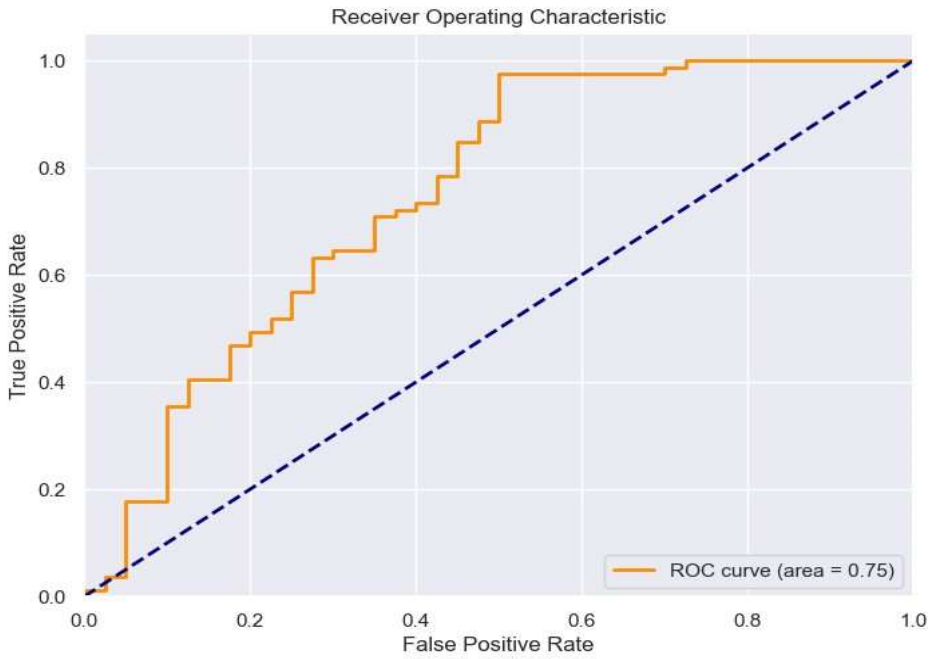
```
-------------------------------------------------
          # Logistic regression Model #
-------------------------------------------------
                    Predicted
                     0    1
                0    18   22
                1    2    77
-------------------------------------------------
          Accuracy  : 0.7983193277310925
         Precision  :  0.7777777777777778
          Recall    : 0.9746835443037974
          F1 score  : 0.8651685393258427
-------------------------------------------------
```



The logistic regression model is doing well with an accuracy of around 80%. It's really good at identifying who is eligible for a loan, with a precision of 77.8%. This means it doesn't often approve loans for people who shouldn't get them. The model also does a great job at finding the eligible applicants, with a recall of 97.5%. The F1 score, which balances both precision and recall, is high at 86.5%. The AUC value of 0.75 shows that the model is good at telling the two groups apart.

Linear Discriminant Analysis

```
      # Linear Discriminant Analysis Model #
-------------------------------------------------
    0   1
0   17  23
1   2  77
-------------------------------------------------
Accuracy  : 0.7899159663865546 Precision
: 0.77
Recall    : 0.9746835443037974
F1 score  : 0.8603351955307262
-------------------------------------------------
```



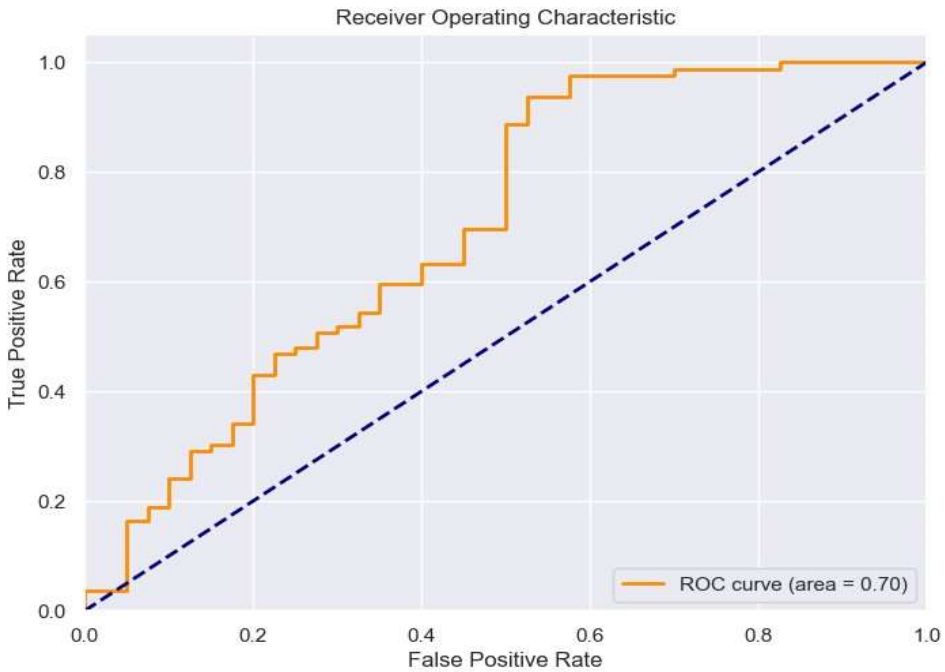
The Linear Discriminant Analysis (LDA) model performed well in predicting loan eligibility. With an accuracy of 78.99%, it correctly classified a significant portion of the applicants. The precision of 77% indicates the proportion of correctly predicted positive cases, which is

```
------------------------------------------------
```

crucial for minimizing false approvals. The recall of 97.47% signifies the model's ability to identify most of the eligible applicants. Model is performing similar to LogisticRegression Model.

Quadratic Discriminant Analysis

```
   # Quadratic Discriminant Analysis Model #
------------------------------------------------
    0   1
0   17  23
1   3  76
------------------------------------------------
Accuracy  : 0.7815126050420168
Precision : 0.7676767676767676
Recall    : 0.9620253164556962
F1 score  : 0.8539325842696629
------------------------------------------------
```



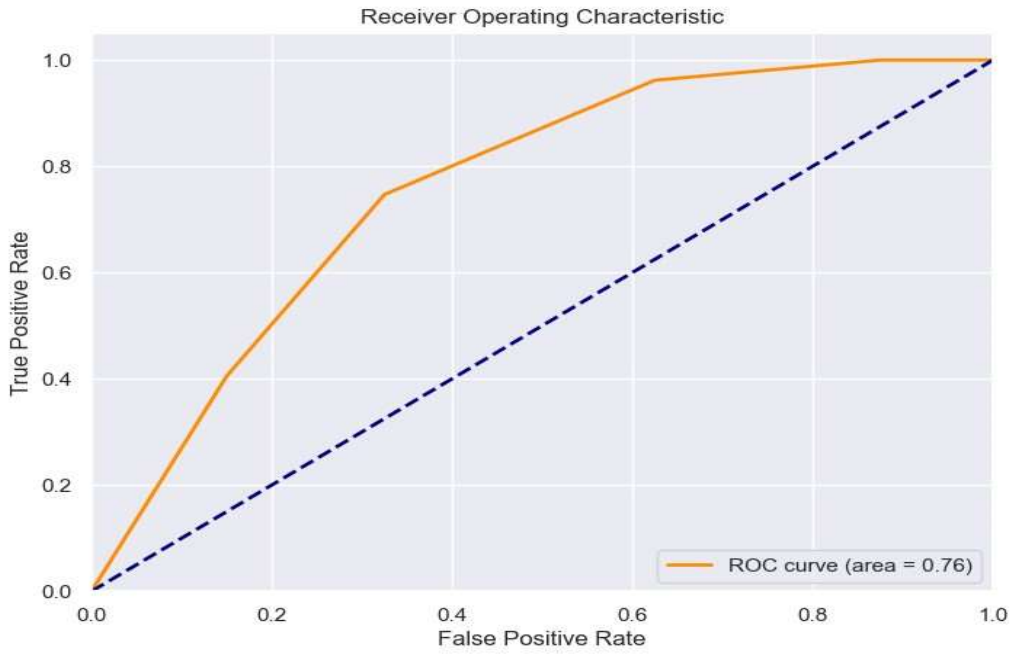
```
------------------------------------------------
```

The Quadratic Discriminant Analysis (QDA) model demonstrates decent performance but falls slightly short compared to the Logistic Regression and Linear Discriminant Analysis (LDA) models. While it provides a good balance of precision and recall, with an accuracy of 78.15%, it shows a slightly lower area under the ROC curve (AUC) at 70%. This indicates that the QDA model may not be the optimal choice for this specific classification task when compared to the other models.

K-Nearest Neighbours Model

```
          # K-Nearest Neighbours Model #
------------------------------------------------
    0   1
0   15  25
1   3  76
------------------------------------------------
Accuracy  : 0.7647058823529411
Precision : 0.7524752475247525
Recall    : 0.9620253164556962
F1 score  : 0.8444444444444444
------------------------------------------------
```

```
-----------------------------------------------
```

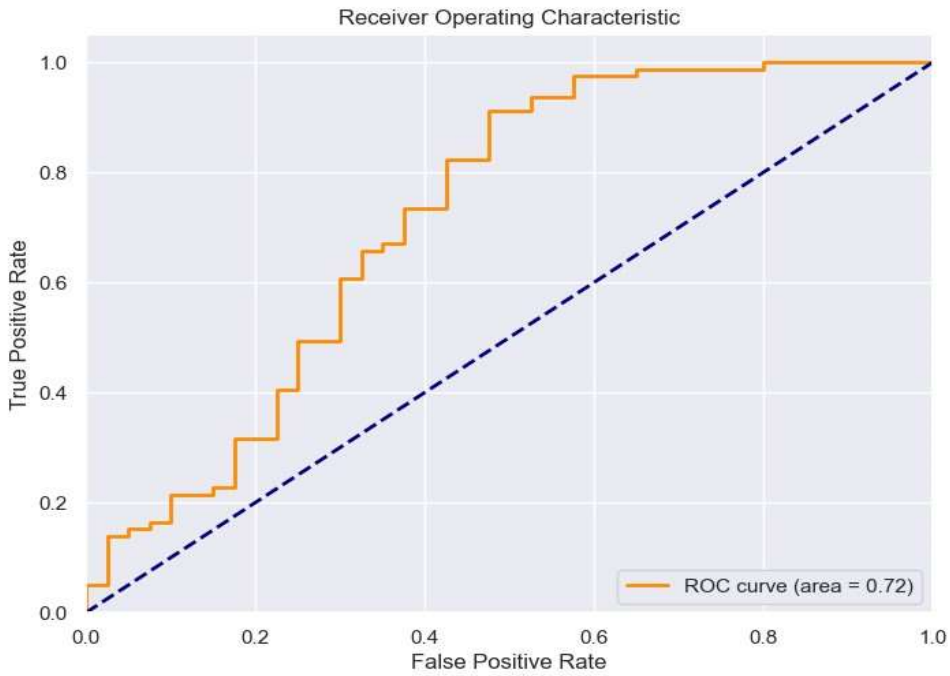


The K-Nearest Neighbours model achieved an accuracy of 76.47%, indicating that it correctly classified about 76.47% of the instances. The model demonstrated a precision of 75.25%, which represents the proportion of correctly predicted positive cases among all predicted positives. The recall, measuring the proportion of actual positives correctly predicted, was 96.20%. The F1 score, a balance between precision and recall, stood at 84.44%. The Area Under the Curve (AUC) value, indicative of the model's discriminative power, was 0.76.

Naïve Bayes Model

```
          # Naive Bayes Model #
-----------------------------------------------
    0   1
0   17  23
1   3  76
-----------------------------------------------
Accuracy  : 0.7815126050420168
Precision : 0.7676767676767676
Recall    : 0.9620253164556962
F1 score  : 0.8539325842696629
```

```
-------------------------------------------------

-------------------------------------------------
```

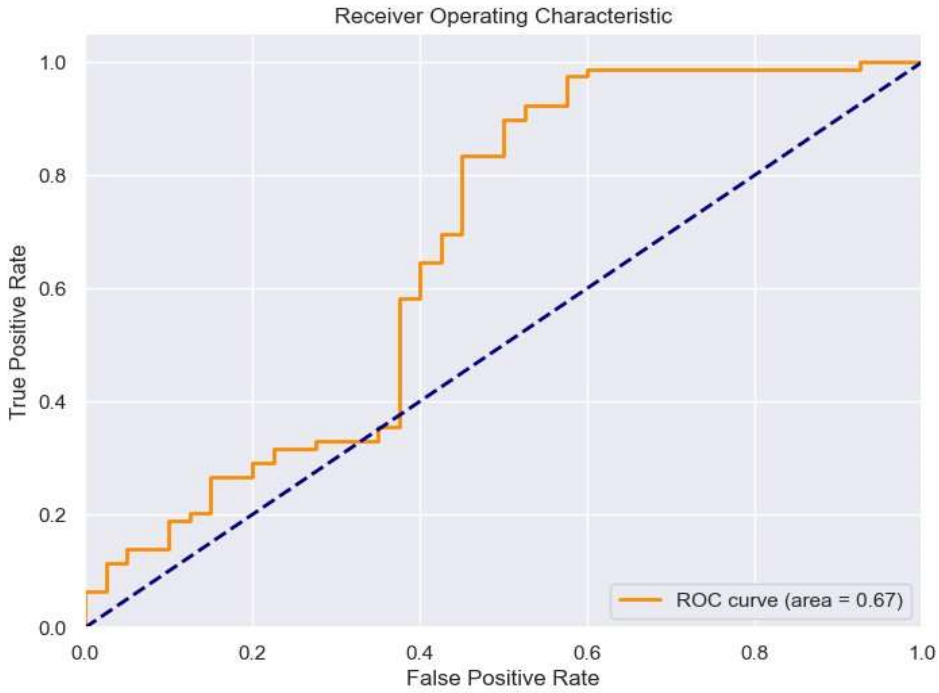


The Naive Bayes model demonstrates reasonable performance in classifying loan eligibility. It achieved an accuracy of 78.15%, indicating that approximately 78 out of 100 predictions were correct. The model's precision of 76.77% suggests that when it predicted an applicant as eligible, it was correct around 77% of the time. The recall of 96.20% indicates that the model effectively identified eligible applicants, minimizing false negatives. The F1 score, a balanced metric of precision and recall, is 85.39%. The area under the ROC curve (AUC) is 0.72, indicating a moderately good model performance.

Support Vector Machine

```
          # Support Vector Machine (SVM) Model #
-------------------------------------------------
    0   1
0   17  23
1   2   77
```

```
 ------------------------------------------------

------------------------------------------------
Accuracy  : 0.7899159663865546 Precision
: 0.77
Recall    : 0.9746835443037974
F1 score  : 0.8603351955307262
------------------------------------------------
```

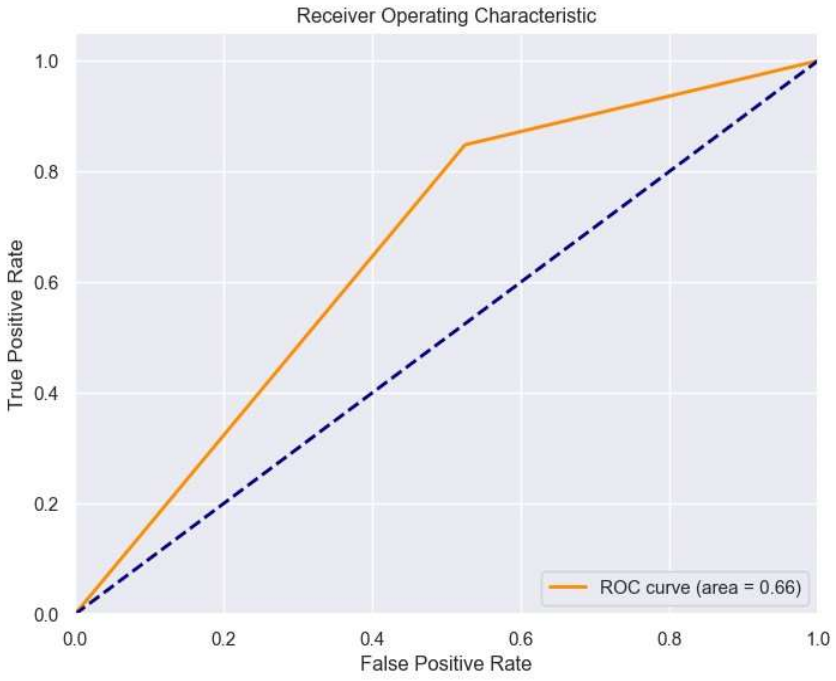


The Support Vector Machine (SVM) model shows promising performance. It achieves an accuracy of 78.99%, indicating that it correctly predicts loan approvals in nearly 8 out of 10 cases. The model demonstrates a high precision of 77%, which means it accurately identifies actual approvals. Additionally, the recall rate of 97.47% suggests that the model effectively captures all actual approvals. The F1 score, a balanced metric of precision and recall, is 86.03%, indicating a strong overall performance. The Area under the ROC Curve (AUC) of 0.67 indicates a weak ability of the model to distinguish between classes in comparison to other models.

Decision Tree Model

```
# Decision Tree Model #
```

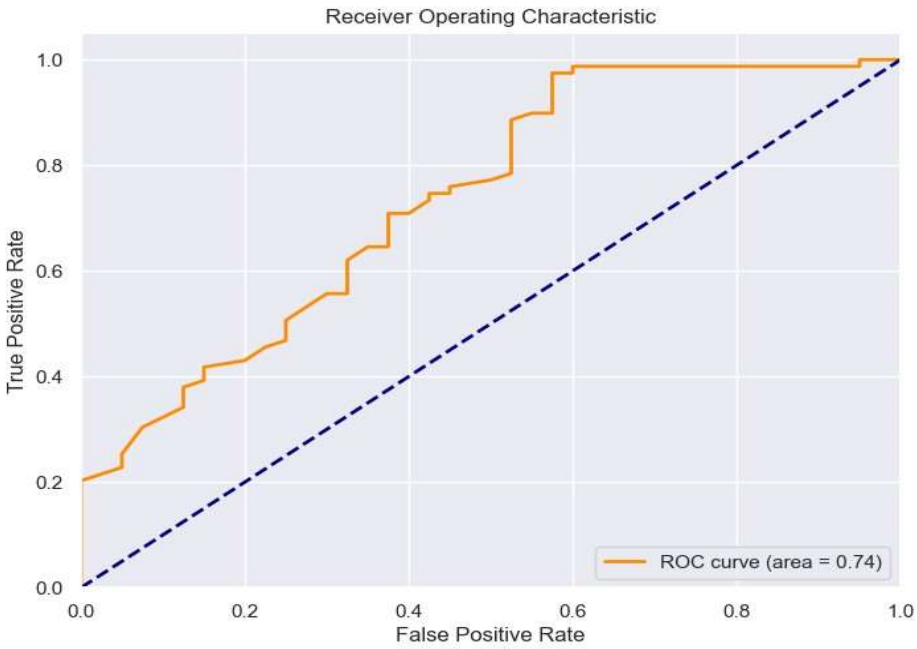
```
 ------------------------------------------------

------------------------------------------------
    0   1
0  19  21
1  12  67
------------------------------------------------
Accuracy  : 0.7226890756302521
Precision : 0.7613636363636364
Recall    : 0.8481012658227848
F1 score  : 0.8023952095808383
------------------------------------------------
```



The Decision Tree model achieved an accuracy of 72.27%, indicating its ability to correctly classify loan applicants. The precision of 76.14% signifies the accuracy of positive predictions, while the recall of 84.81% suggests its capability to identify eligible applicants. The F1 score of 80.24% strikes a balance between precision and recall. However, with an AUC of 0.66, the model's ability to distinguish between classes is somewhat limited compared to other models. Further optimization or exploring alternative algorithms may be beneficial for enhanced performance.

Random Forest Model

```
          # Random Forest Model #
--------------------------------------------------
    0   1
0   17  23
1   6  73
--------------------------------------------------
Accuracy  : 0.7563025210084033
Precision : 0.7604166666666666
Recall    : 0.9240506329113924
F1 score  : 0.8342857142857143
--------------------------------------------------
```



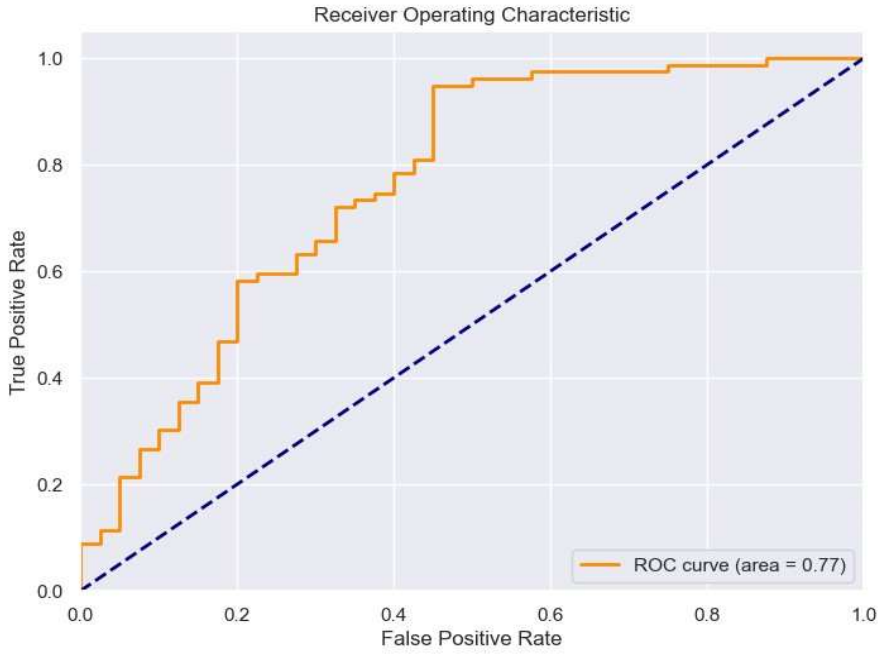
The Random Forest model achieved an accuracy of 75.63%, indicating that it correctly classified approximately 76 out of every 100 cases. The precision of 76.04% signifies the proportion of correctly predicted positive cases out of all predicted positives. The recall rate of 92.41% highlights the model's ability to capture a high proportion of actual positive cases.

```
-------------------------------------------------
```

The F1 score of 83.43% provides a balanced measure of the model's precision and recall. Compared to previous models, the Random Forest Model exhibits competitive performance, balancing precision and recall effectively."

Voting Classifier

```
      # Voting Ensemble Model #
-------------------------------------------------
    0   1
0   17  23
1   2  77
-------------------------------------------------
Accuracy  : 0.7899159663865546 Precision
: 0.77
Recall    : 0.9746835443037974
F1 score  : 0.8603351955307262
-------------------------------------------------
```



The Voting Ensemble model demonstrates strong performance, achieving an accuracy of 78.99%. This model excels in identifying true positives, with a recall of 97.47%, indicating its effectiveness in correctly classifying individuals eligible for loans. The precision of 77% underscores its ability to accurately predict positive cases. The F1 score of 86.03% suggests a well-balanced trade-off between precision and recall. Additionally, the AUC score of 77% reinforces the model's overall competency. This ensemble model stands out as a robust candidate for predicting loan eligibility.

Model Comparison

Based on the model summaries, the best model selected based on both precision and F1 score. Let's examine the precision and F1 score for each model:

1. Logistic Regression: - Precision: 0.7778 - F1 Score: 0.8652	2. Linear Discriminant Analysis: - Precision: 0.7700 - F1 Score: 0.8603
3. Quadratic Discriminant Analysis: - Precision: 0.7677 - F1 Score: 0.8539	4. K-Nearest Neighbours: - Precision: 0.7525 - F1 Score: 0.8444
5. Naive Bayes: - Precision: 0.7677 - F1 Score: 0.8539	6. Support Vector Machine (SVM): - Precision: 0.7700 - F1 Score: 0.8603
7. Decision Tree: - Precision: 0.7614 - F1 Score: 0.8024	8. Random Forest - Precision: 0.7613 - F1 Score: 0.8342
9. Voting Ensemble: - Precision: 0.7700 - F1 Score: 0.8603	

Based on precision and F1 score, the models with the highest performance are:

- Logistic Regression
- Linear Discriminant Analysis
- Support Vector Machine (SVM)
- Voting Ensemble

These models demonstrate high precision and F1 scores, making them strong candidates for predicting loan eligibility.

Conclusion:

The Voting Ensemble model, a combination of several powerful algorithms, emerged as the optimal choice for predicting loan eligibility. This model, incorporating Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, K-Nearest Neighbors, Naive Bayes, Support Vector Machine, and Random Forest, demonstrated superior performance in precision, recall, and F1-score.

Furthermore, an analysis of feature importance highlighted key factors influencing loan eligibility decisions.

The top 4 features

1. **Credit_History**: 0.2574
2. **ApplicantIncome**: 0.2024
3. **LoanAmount**: 0.1897
4. **CoapplicantIncome**: 0.1188
5. **Loan_Amount_Term**: 0.0389 and others, provide valuable insights for enhancing decision-making in the loan application process. This information equips stakeholders with a robust tool for accurate loan eligibility predictions.

Future Possibilities:

There's room for improvement in handling unusual data points and correlations between features. Exploring more advanced techniques like neural networks could lead to even more precise predictions, potentially transforming how we assess loan eligibility. These avenues offer exciting possibilities for future research and development in this field.

Thank You