Credit Risk Evaluation for Loan Approval

Kamthe Rupali[1], Lembhe Akshata[2], Deepali Akolkar[3]

[1,2,3]Assistant Professor, Department of Statistics, Dr.D.Y.Patil ACS College Pimpri Pune 18

Abstract

The loan approval process plays a vital role in the financial sector, requiring careful assessment of an applicant's financial background to determine their eligibility. This project introduces a machine learning-based model designed to predict loan approval outcomes using applicant data.

The model is developed using a dataset that includes key variables such as income, credit history, loan amount, and other relevant financial indicators. To build a reliable prediction system, we apply algorithms like decision trees and gradient boosting, along with cross-validation methods to improve accuracy and generalization.

The findings highlight the model's ability to accurately distinguish between approved and rejected applications. Incorporating this predictive tool into financial workflows can help institutions make informed decisions, improve operational efficiency, and manage risk more effectively.

Overall, the proposed system presents a data-driven solution to enhance the speed and accuracy of the loan approval process within banking environments.

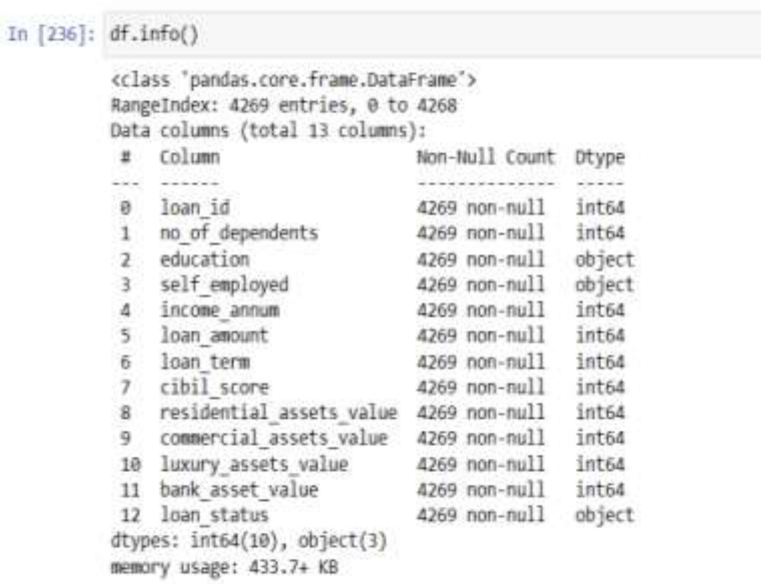
Keywords: Loan approval, Machine learning,Cibil Score.

Introduction

Loan approval is a critical process in the financial sector, where an applicant's creditworthiness is evaluated based on factors like income, credit score, and employment history. Traditional methods rely heavily on manual assessments, making the process slow and prone to bias. With the rise in application volumes, there is a growing need for more efficient and accurate evaluation methods. This project explores the use of machine learning to predict loan approvals by analyzing historical application data. Using models such as decision trees and gradient boosting, we develop a predictive system aimed at improving decision-making accuracy and efficiency. Our study evaluates model performance and discusses its potential integration into the loan approval workflow, offering benefits for both lenders and applicants. Further sections detail the dataset, methodology, results, and future directions for enhancing the model.

Analysis

The data for this project is available at Kaggle - **https://www.kaggle.com/code/adilhabib/loan-approval-model-comparison**

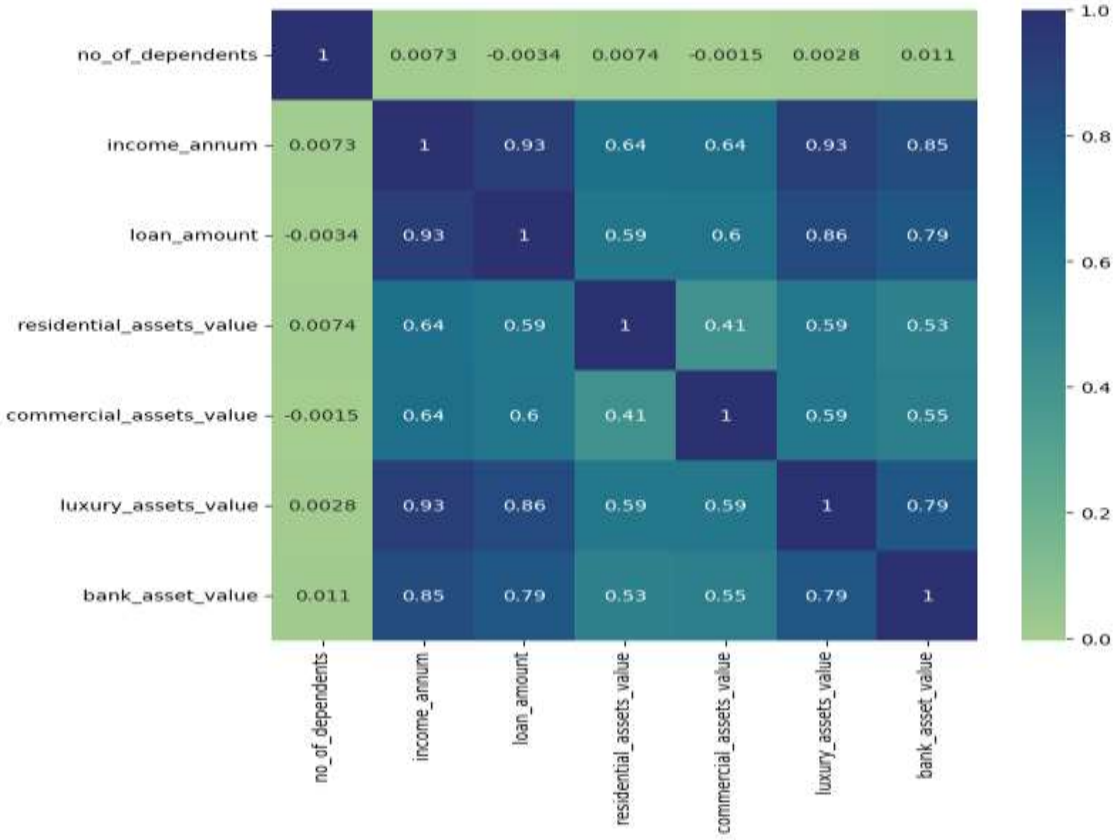


```
In [236]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4269 entries, 0 to 4268
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   loan_id                   4269 non-null   int64
 1   no_of_dependents          4269 non-null   int64
 2   education                 4269 non-null   object
 3   self_employed             4269 non-null   object
 4   income_annum              4269 non-null   int64
 5   loan_amount               4269 non-null   int64
 6   loan_term                 4269 non-null   int64
 7   cibil_score               4269 non-null   int64
 8   residential_assets_value  4269 non-null   int64
 9   commercial_assets_value   4269 non-null   int64
 10  luxury_assets_value       4269 non-null   int64
 11  bank_asset_value          4269 non-null   int64
 12  loan_status               4269 non-null   object
dtypes: int64(10), object(3)
memory usage: 433.7+ KB
```

The dataset contains 4269 samples with 13 features, one of them being whether the applicant got approved for a loan or not. It appears that there are no null values and the columns are of appropriate data types.

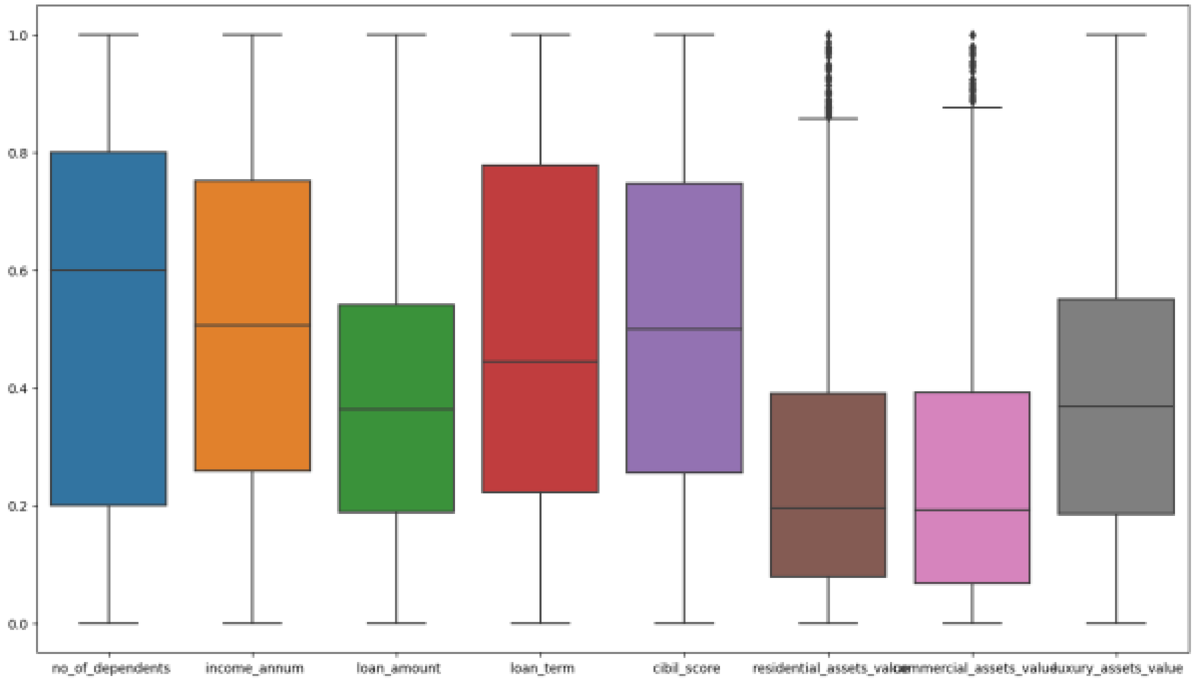
Explanatory Analysis

The initial visualization of the dataset involves constructing a correlation matrix to examine the connections among numeric variables.



Conclusion:

In this correlation heatmap we can see that income annum and loan amount are positively correlated with each other, also income annum and luxury assets are positively correlated**.**

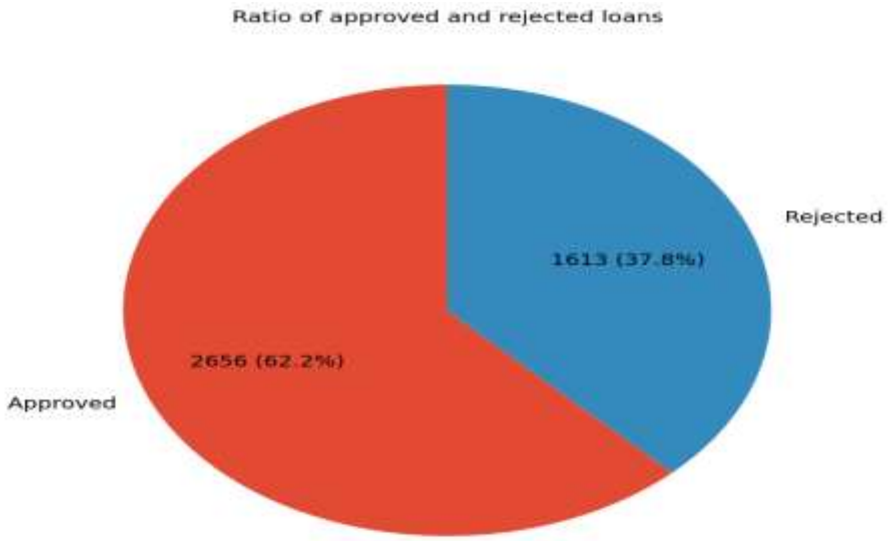


Conclusion:

The box plot displays the distribution of several numerical features related to loan applications, all normalized between 0 and 1. Here's what can be concluded:

- **Most features** like no_of_dependents, income_annum, loan_amount, loan_term, and cibil_score show a relatively balanced spread with no extreme outliers.
- **residential_assets_value, commercial_assets_value, and luxury_assets_value** display significant **outliers**, indicating that a few applicants have exceptionally high asset values compared to the majority.
- The **median values** across most features are centered between 0.3 and 0.6, suggesting moderate normalized values in the dataset.
- The **range** (interquartile spread) for variables like loan_term and no_of_dependents is wider, showing more variability in these attributes.
- Features with many outliers, especially those related to assets, may require transformation or special handling (e.g., scaling or log transformation) to prevent them from skewing model performance.

Overall, this graph helps identify variability, spread, and the presence of outliers across key financial and demographic factors, aiding in better data preprocessing and feature selection for predictive modelling.

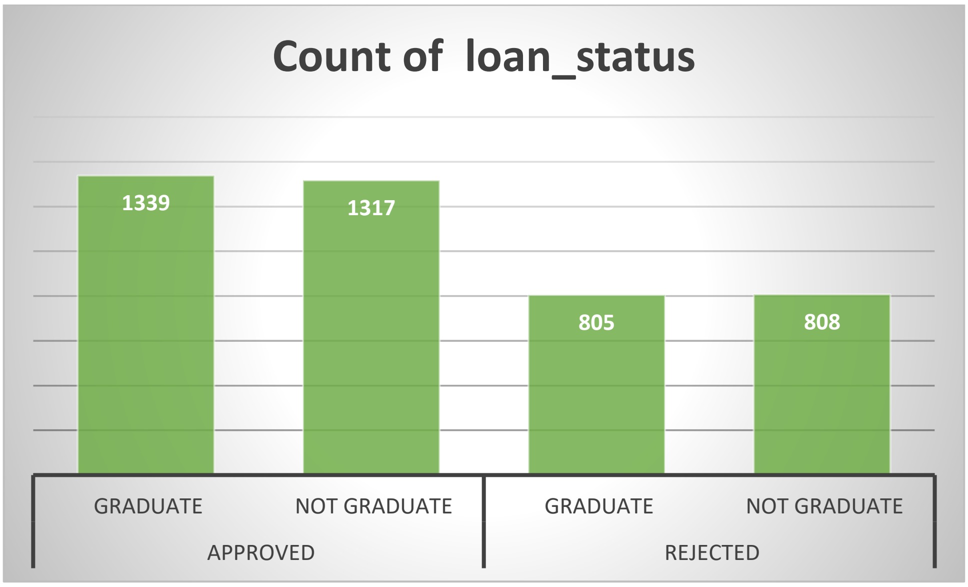


Conclusion:

The pie chart illustrates the distribution between approved and rejected loan applications. Out of the total applications:

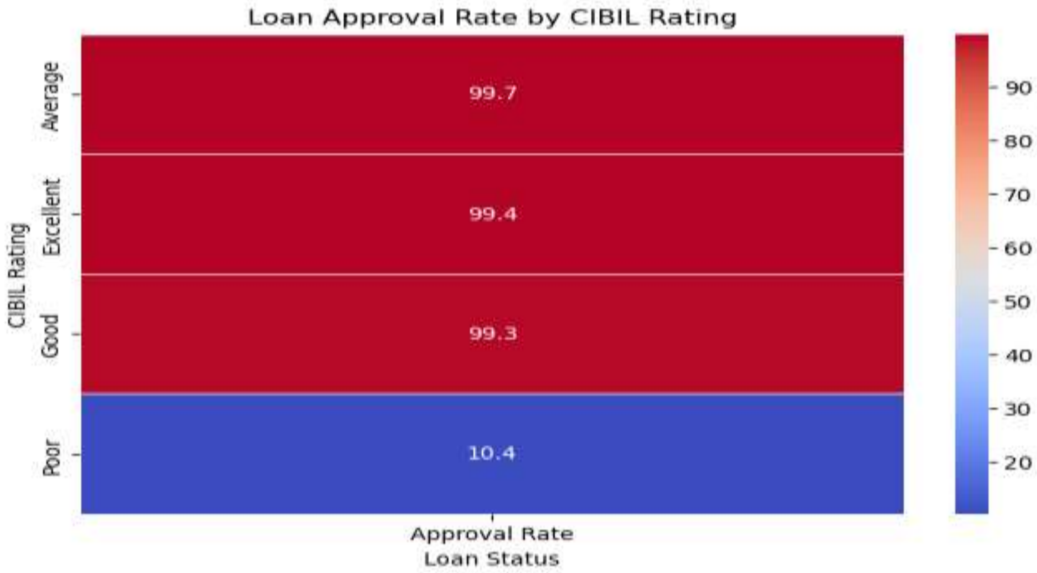
- **62.2% (2656)** were approved
- **37.8% (1613)** were rejected

This indicates that a majority of loan applications were approved, suggesting that most applicants met the necessary eligibility criteria. However, a significant portion—over one-third—were rejected, highlighting the importance of accurately assessing credit risk and eligibility to maintain responsible lending practices. This distribution also supports the need for predictive models that can help automate and optimize loan approval decisions.



Conclusion :

here the approval chances for graduate people are little bit more as compare to not graduate people.



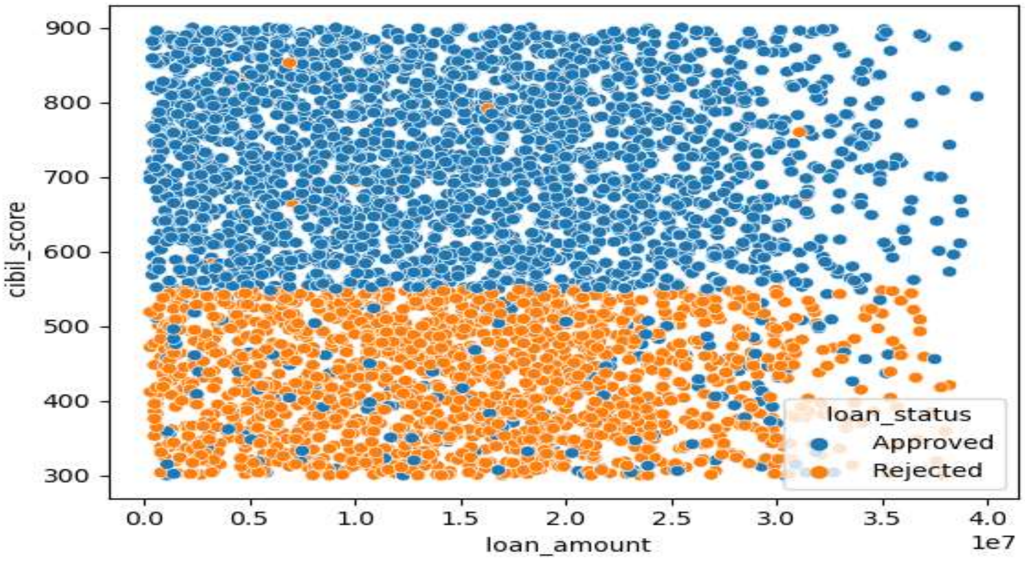
Conclusion:

This heatmap shows the **loan approval rates categorized by CIBIL rating**. The findings are as follows:

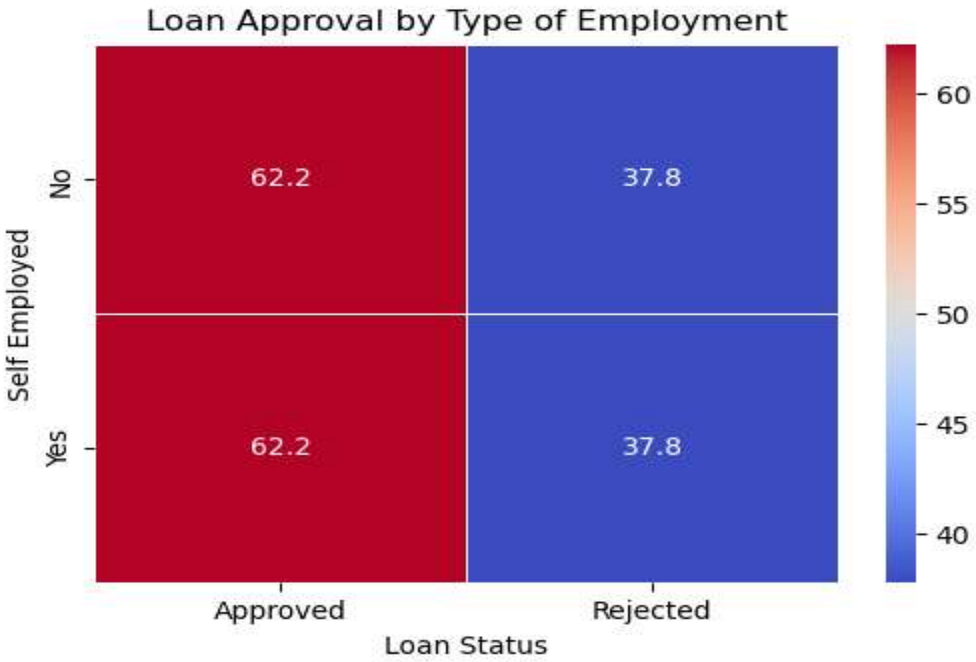
- Applicants with **Average (99.7%)**, **Excellent (99.4%)**, and **Good (99.3%)** CIBIL ratings have **very high approval rates**, indicating strong creditworthiness and a high likelihood of loan approval.
- In contrast, applicants with a **Poor CIBIL rating** have a dramatically lower approval rate of **only 10.4%**, highlighting a significant drop in approval chances.

Overall Conclusion:

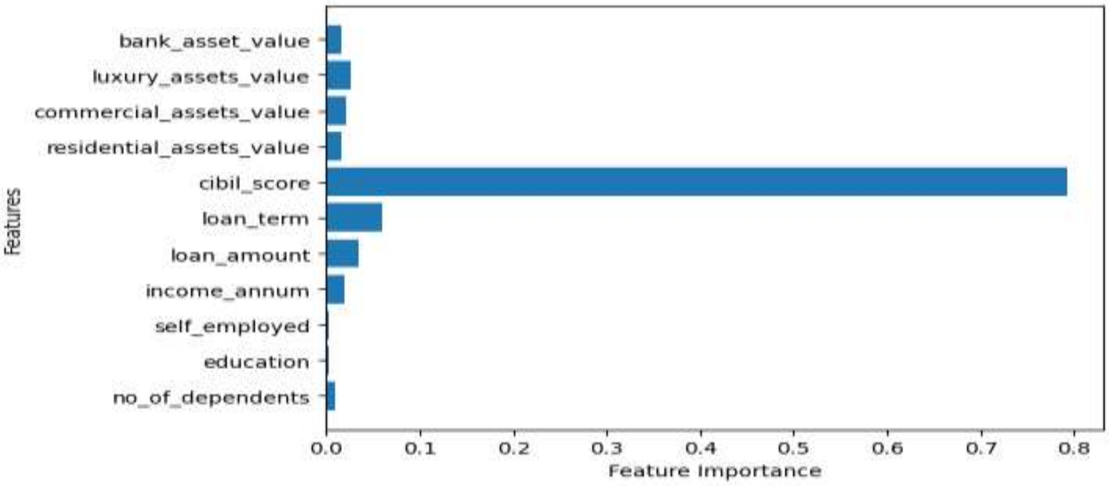
CIBIL rating plays a **critical role** in loan approval decisions. Higher credit ratings are strongly associated with higher chances of loan approval, while poor credit history significantly reduces the likelihood of approval. This emphasizes the importance of maintaining a good credit score for successful loan applications.



Conclusion: In this plot, cibil score is important term in loan approval status.



Conclusion: In this self employment its doesn't impact as same as cibil score in loan approval.



Conclusion: In loan approval the cibil score is most important feature.In the above depicted bar chart, we can see that before fitting the model feature scaling is important.

ALGORITHMS AND TECHNIQUES

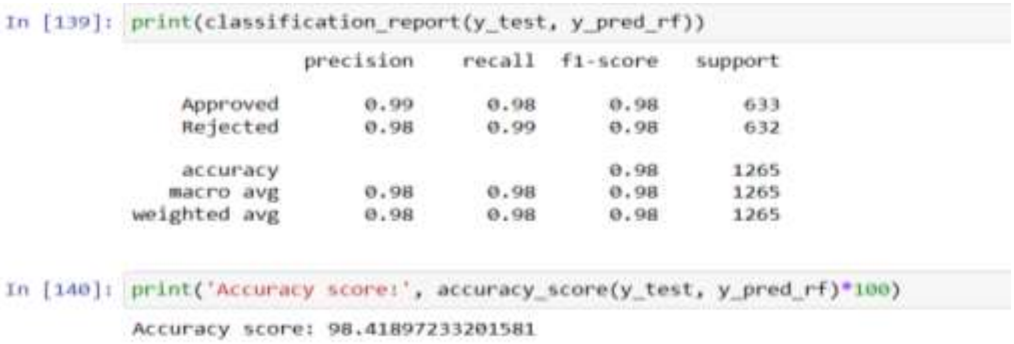
- **Logistic Regression:** Logistic Regression is a widely used classification algorithm that predicts binary outcomes (such as Yes/No, True/False, or 0/1) based on input features. It uses the logistic (sigmoid) function to estimate the probability of a certain outcome and helps establish a relationship between the input variables and the likelihood of a specific result.
- **Decision Trees:** Decision Trees are supervised learning models primarily applied to classification tasks. This algorithm splits the dataset into subsets based on feature values, aiming to create branches that lead to more uniform (homogeneous) groups. At each step, it selects the best attribute

to divide the data using criteria like Gini Index or Information Gain. As more nodes and branches are added, the classification accuracy generally improves by increasing the purity of each node with respect to the target variable.

- **Random Forest:** Random Forest is an ensemble learning technique that enhances the performance of decision trees by creating a collection (or "forest") of them. Each tree is built from a random subset of the dataset and a random selection of features. The final prediction is made by aggregating the predictions of all individual trees, typically through majority voting for classification or averaging for regression. This approach reduces overfitting and improves overall model robustness.
- **Naive Bayes:** Naïve Bayes is a probabilistic classifier based on Bayes' Theorem, commonly used for classification tasks. It assumes independence among features and calculates the probability of a data point belonging to a particular class. Despite its simplicity, Naïve Bayes is highly effective and is often used in real-world applications such as spam detection, document categorization, and sentiment analysis.
- **K-Nearest Neighbors (KNN):** KNN is a basic yet powerful supervised learning algorithm used for both classification and regression. Instead of training a model in the traditional sense, KNN stores the entire dataset and makes predictions based on the similarity of new data points to the existing data. When a new instance needs to be classified, the algorithm identifies the 'k' closest training examples and assigns the most common class among them

Random Forest Classification

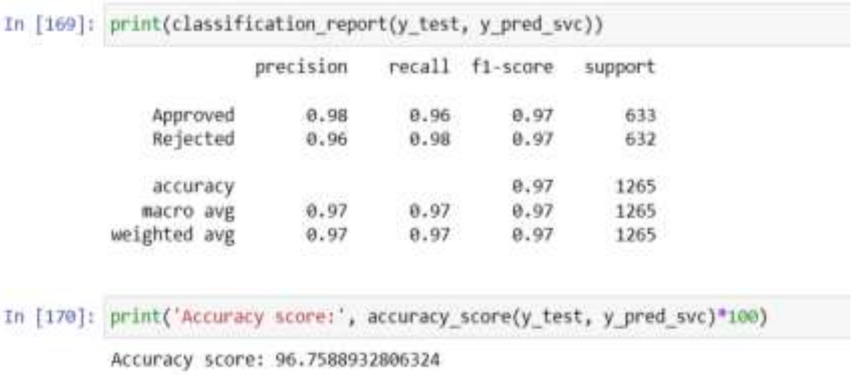
Accuracy: 98.41



```
In [139]: print(classification_report(y_test, y_pred_rf))

                        precision    recall  f1-score   support

              Approved       0.99      0.98      0.98       633
              Rejected       0.98      0.99      0.98       632

              accuracy                           0.98      1265
             macro avg       0.98      0.98      0.98      1265
          weighted avg       0.98      0.98      0.98      1265

In [140]: print('Accuracy score:', accuracy_score(y_test, y_pred_rf)*100)

          Accuracy score: 98.41897233201581
```

Support Vector Machine (SVM)

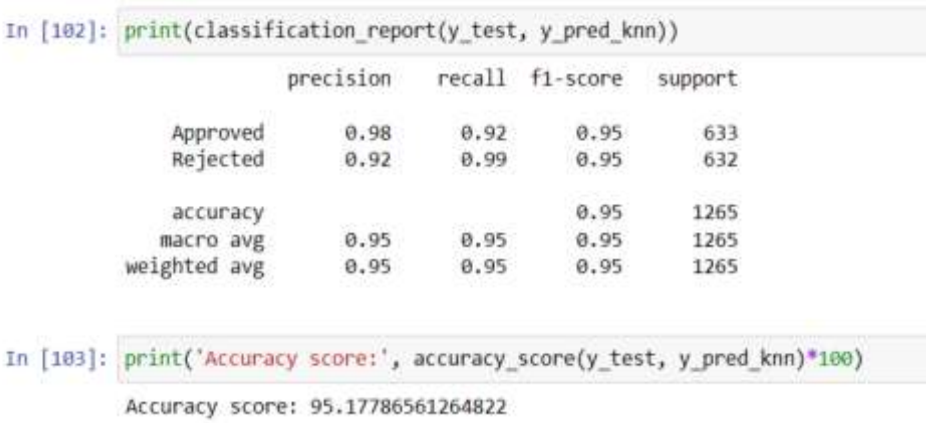
Accuracy: 96.75



```
In [169]: print(classification_report(y_test, y_pred_svc))

                        precision    recall  f1-score   support

              Approved       0.98      0.96      0.97       633
              Rejected       0.96      0.98      0.97       632

              accuracy                           0.97      1265
             macro avg       0.97      0.97      0.97      1265
          weighted avg       0.97      0.97      0.97      1265

In [170]: print('Accuracy score:', accuracy_score(y_test, y_pred_svc)*100)

          Accuracy score: 96.7588932806324
```

K-Nearest Neighbour (KNN)

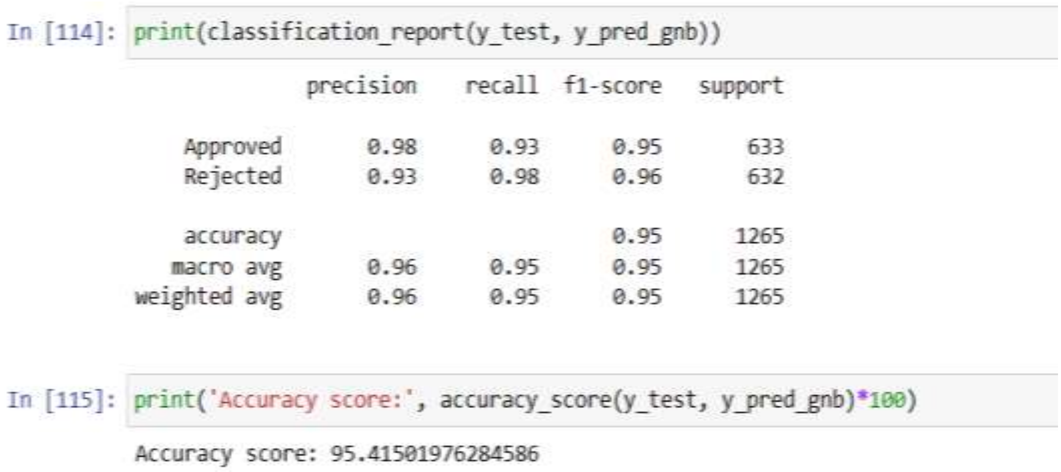
Accuracy:95.17



```
In [102]: print(classification_report(y_test, y_pred_knn))

                        precision    recall  f1-score   support

              Approved       0.98      0.92      0.95       633
              Rejected       0.92      0.99      0.95       632

              accuracy                           0.95      1265
             macro avg       0.95      0.95      0.95      1265
          weighted avg       0.95      0.95      0.95      1265


In [103]: print('Accuracy score:', accuracy_score(y_test, y_pred_knn)*100)

          Accuracy score: 95.17786561264822
```

Gaussian Naive Bayes

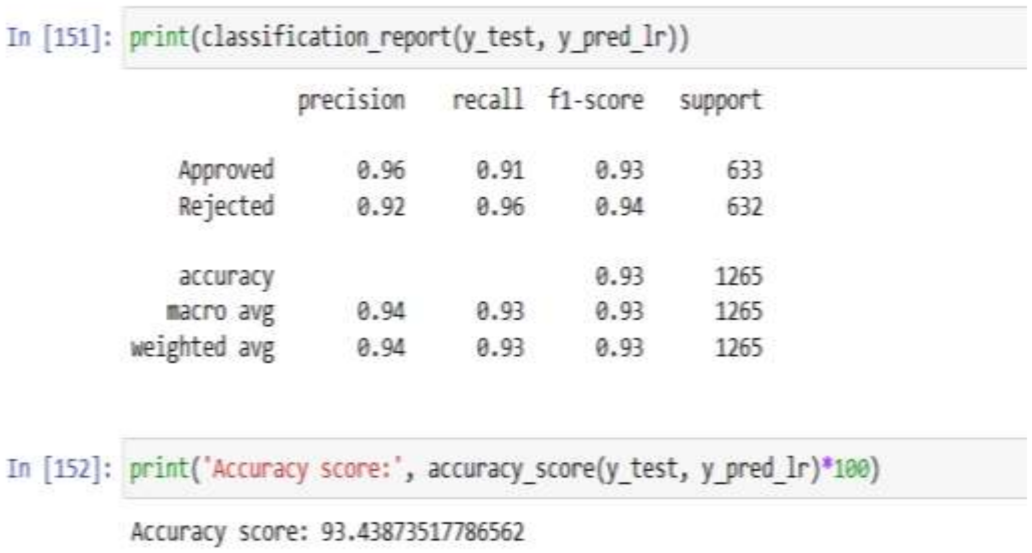
Accuracy:95.41



```
In [114]: print(classification_report(y_test, y_pred_gnb))

                        precision    recall  f1-score   support

              Approved       0.98      0.93      0.95       633
              Rejected       0.93      0.98      0.96       632

              accuracy                           0.95      1265
             macro avg       0.96      0.95      0.95      1265
          weighted avg       0.96      0.95      0.95      1265


In [115]: print('Accuracy score:', accuracy_score(y_test, y_pred_gnb)*100)

          Accuracy score: 95.41501976284586
```

Logistic Regression

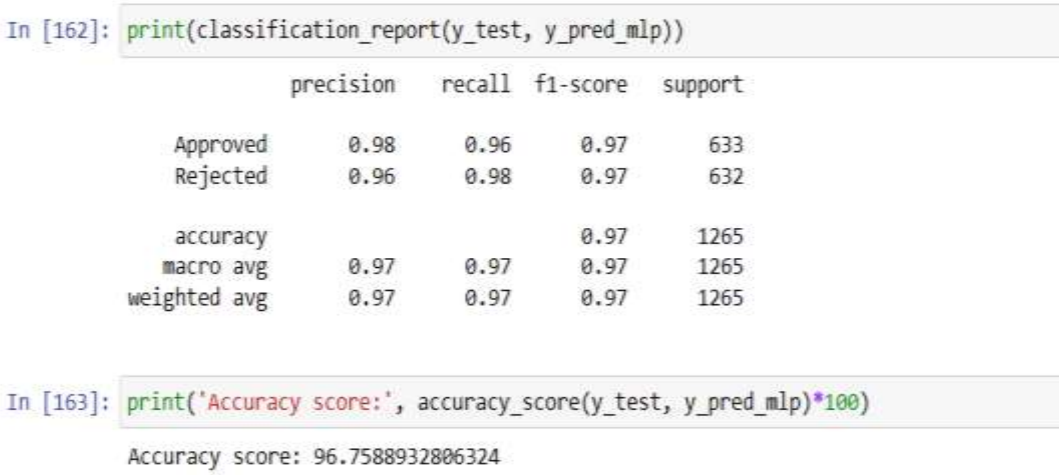
Accuracy:93.43



```
In [151]: print(classification_report(y_test, y_pred_lr))

                        precision    recall  f1-score   support

              Approved       0.96      0.91      0.93       633
              Rejected       0.92      0.96      0.94       632

              accuracy                           0.93      1265
             macro avg       0.94      0.93      0.93      1265
          weighted avg       0.94      0.93      0.93      1265


In [152]: print('Accuracy score:', accuracy_score(y_test, y_pred_lr)*100)

          Accuracy score: 93.43873517786562
```

Artificial neural network (ANN)

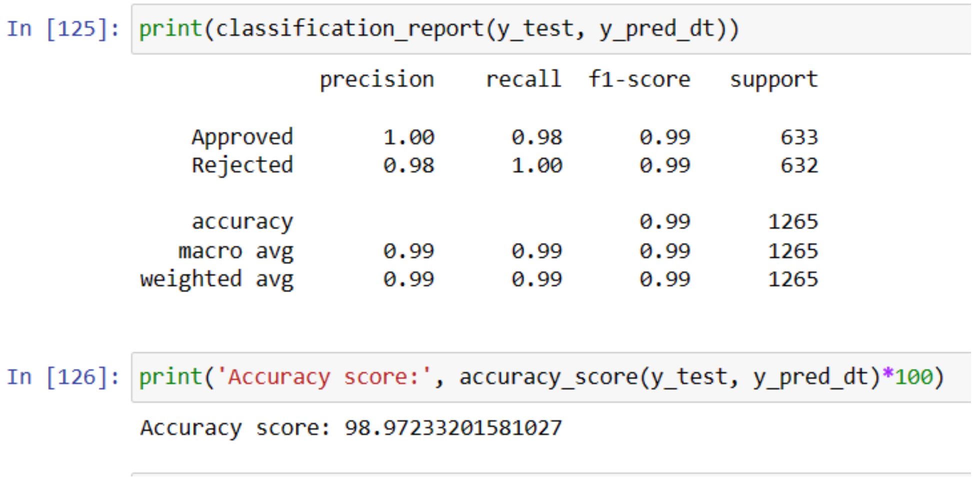
Accuracy: 96.75



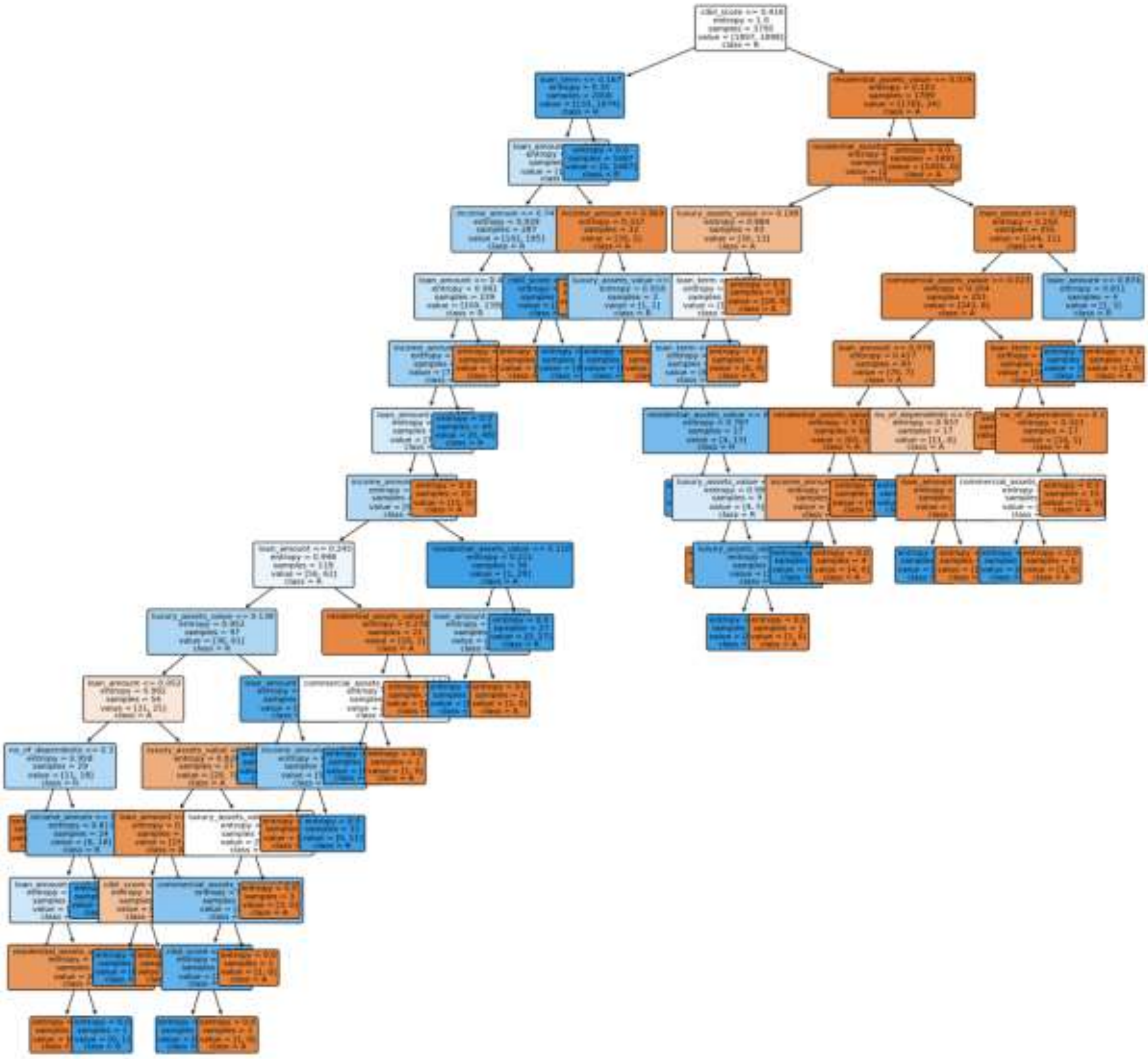
```
In [162]: print(classification_report(y_test, y_pred_mlp))

                  precision    recall  f1-score   support

      Approved       0.98      0.96      0.97       633
      Rejected       0.96      0.98      0.97       632

      accuracy                           0.97      1265
     macro avg       0.97      0.97      0.97      1265
  weighted avg       0.97      0.97      0.97      1265


In [163]: print('Accuracy score:', accuracy_score(y_test, y_pred_mlp)*100)

          Accuracy score: 96.7588932806324
```

Decision Tree Classifier

Accuracy: 98.97



```
In [125]: print(classification_report(y_test, y_pred_dt))

                  precision    recall  f1-score   support

      Approved       1.00      0.98      0.99       633
      Rejected       0.98      1.00      0.99       632

      accuracy                           0.99      1265
     macro avg       0.99      0.99      0.99      1265
  weighted avg       0.99      0.99      0.99      1265

In [126]: print('Accuracy score:', accuracy_score(y_test, y_pred_dt)*100)

          Accuracy score: 98.97233201581027
```



Conclusion :

Comparison of Models:

Decision Tree Accuracy score: 98.97233201581027

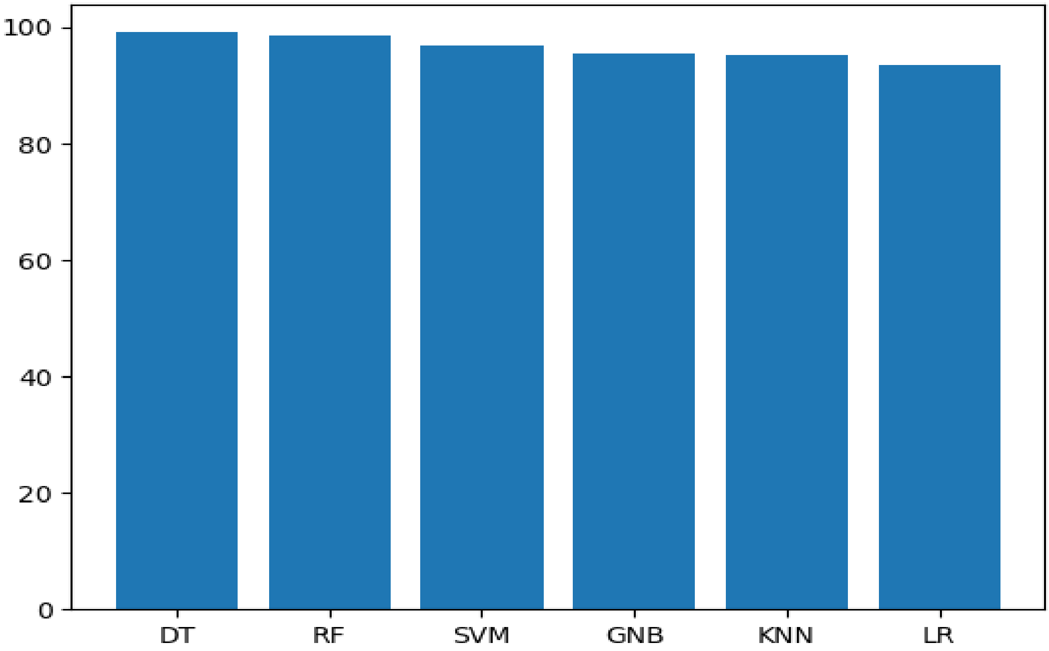
Random Forest Accuracy score: 98.41897233201581

Support Vector Machines Accuracy score: 96.7588932806324

Gaussian Naive Bayes Accuracy score: 95.41501976284586

K-Nearest Neighbour Accuracy score: 95.17786561264822

Logistic Regression Accuracy score: 93.43873517786562



Conclusion :

- The Random Forest model, which achieved an accuracy of 98.4%, offers a notable advantage by identifying the most influential features through its feature importance matrix. This helps in refining the model by focusing on the attributes that significantly impact prediction accuracy and reducing the risk of overfitting.
- Previously, an accuracy of 93% was observed, but by selecting key features based on their importance, model performance has been further optimized. Among the machine learning models used—Logistic Regression, Decision Tree, Random Forest, and K-Nearest Neighbors—their respective accuracies were approximately 93%, 98.97%, 98.4%, and 95.17%.
- Based on these results, the Decision Tree model demonstrated the highest performance, making it the most suitable and effective model for predicting loan approvals in this study.
- **Chi square Analysis :**

Determine relation between loan status and no. of independent

Chi Squared Test Statistics: 1.1296798036848839

P-Value Statistics: 0.5684511474541935

DOF: 2

There is no statistically significant relation between loan status and number of dependent

Conclusion

By creating a predictive risk assessment model that incorporates essential applicant factors, decision-making accuracy receives a significant boost, ensuring a more informed approach to lending.

Identification and utilization of influential factors in loan approval processes lead to optimization, streamlining operations, and enhancing efficiency in assessing and approving loans Assessing the impact of diverse asset types on loan approval not only sheds light on varying risk levels but also informs better decision-making regarding loan amounts and conditions Delving into the correlation between credit scores and borrower attributes yields deeper insights, enabling lenders to tailor offerings and strategies to individual borrower profiles Analyzing loan term dynamics in relation to borrower characteristics refines lending strategies, allowing for more tailored and flexible loan terms that meet borrower needs while managing risk effectively Investigating the relationship between credit scores, income, education, and asset values offers a comprehensive understanding of borrower creditworthiness, facilitating more accurate risk assessment and lending decisions

Suggestion

Building a user-friendly interface or integrating the predictive model into existing loan approval systems for seamless integration into operational processes.

Conducting thorough testing and validation of the predictive model using robust evaluation metrics to assess its accuracy and reliability Providing insights and recommendations based on the predictive model's outcomes to optimize loan approval processes and mitigate risk effectively.

Data availability and quality may pose challenges, especially if the dataset is limited or contains incomplete or inaccurate information, potentially impacting the predictive model's performance.

The predictive model's accuracy may be influenced by factors beyond the scope of the available data, such as economic fluctuations or unforeseen market dynamics.

Reference

1. **Data Source**: Dataset obtained from Kaggle. URL: https://www.kaggle.com/code/adilhabib/loan-approval-model-comparison
2. **Online Article**: *Bank Loan Approval Prediction Using Data Science Technique*. Available at: https://www.ijraset.com/research-paper/bank-loan-approval-prediction-using-data-science-technique
3. Supriya, Pidikiti, et al. (2019). *Loan Prediction Using Machine Learning Models*. *International Journal of Engineering and Techniques*, 5(2), pp. 144–147.
4. Arun Kumar, Garg Ishan, & Kaur Sanmeet. (2016). *Loan Approval Prediction Based on Machine Learning Approach*. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 18(3), pp. 18–21.
5. Tejaswini, J., et al. (2020). *Accurate Loan Approval Prediction Based on Machine Learning Approach*. *Journal of Engineering Science*, 11(4), pp. 523–532