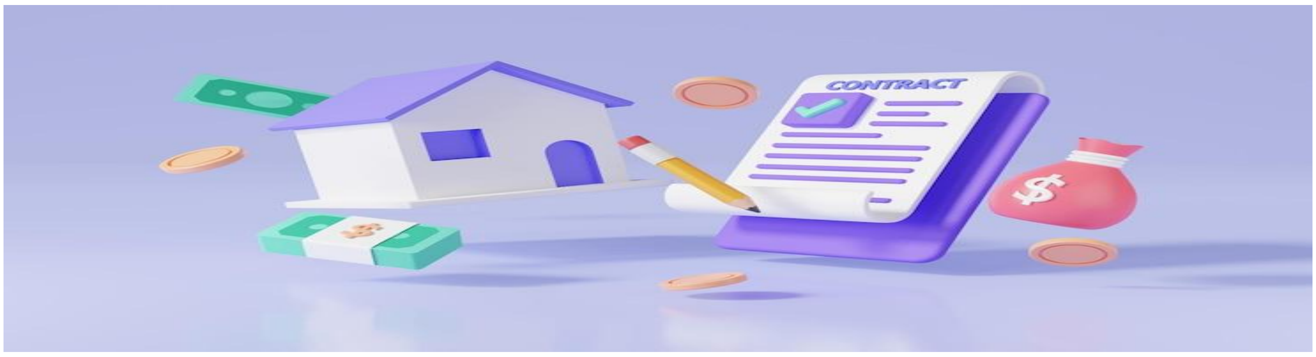
ML PROJECT – "CAPSTONE PROJECT OF LOAN APPROVAL"

- PRESENTED BY NUTAN ZINE

Problem Statement: "Capstone Project Of Loan Approval Prediction"

Financial institutions and banks receive thousands of loan applications every day. Manually evaluating each application based on multiple parameters such as applicant income, credit history, employment status, and loan amount is time-consuming and susceptible to human bias. To streamline the loan approval process and reduce risks associated with defaulted loans, there is a growing need for an automated, accurate, and data-driven loan approval system.

The goal of this capstone project is to develop a **machine learning model** that predicts whether a loan should be **approved or rejected**, based on historical loan application data. This model will help financial institutions make faster, more consistent, and data-backed lending decisions, improving both operational efficiency and customer satisfaction.



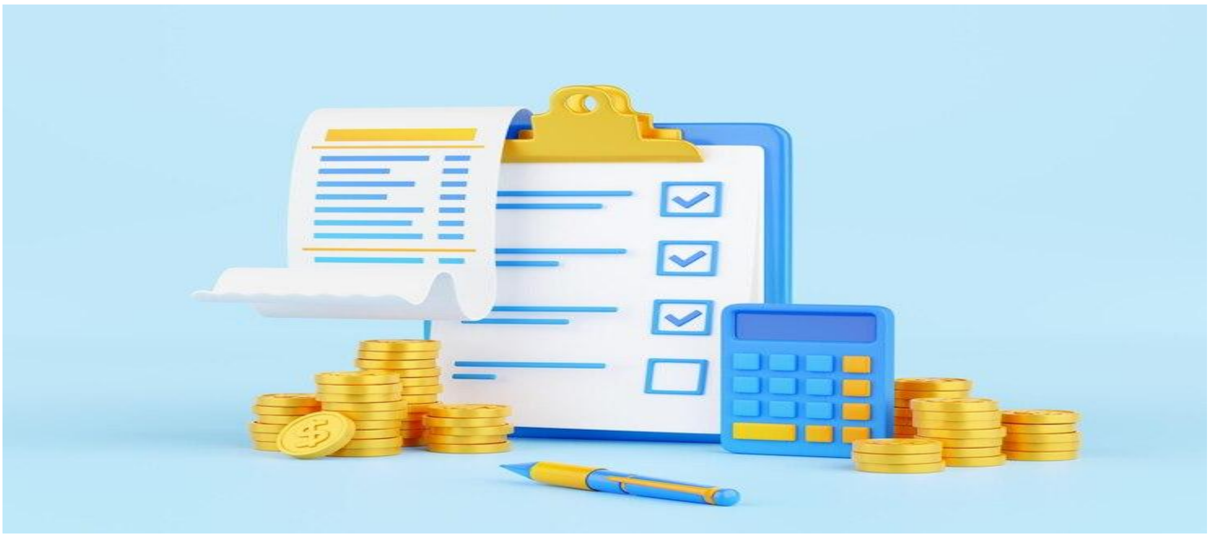
OBJECTIVES :

1. **Understand the Dataset**
   Analyse and explore historical loan application data to identify key features that influence loan approval decisions.
2. **Preprocess the Data**
   Perform data cleaning, handling of missing values, outlier detection, encoding of categorical variables, and normalization/scaling as necessary.
3. **Feature Engineering and Selection**
   Create new meaningful features, evaluate feature importance, and select the most relevant ones to improve model performance.
4. **Build Predictive Models**
   Train and evaluate multiple machine learning algorithms (e.g., Logistic Regression, Decision Tree, Random Forest, etc.) to predict loan approval outcomes.
5. **Optimize Model Performance**
   Use hyperparameter tuning, cross-validation, and performance metrics (accuracy, precision, recall, F1-score, ROC-AUC) to select the best-performing model.
6. **Develop a User Interface (Optional)**
   Create a simple web-based interface or dashboard

OBJECTIVES :

where users can input applicant details and receive instant loan approval predictions.

7. **Document the Project**
   Provide comprehensive documentation including data exploration, model development, evaluation, interpretation, and deployment details.



DATASET COLUMNS -

There are Total 13 Columns in
“Capstone Project of Loan Approval”.

'loan_id', ' no_of_dependents', '
education', ' self_employed', '
income_annum', ' loan_amount', '
loan_term', ' cibil_score', '
residential_assets_value', '
commercial_assets_value', '
luxury_assets_value', '
bank_asset_value', ' loan_status' .

STEPS TAKEN IN DATASET ANALYSIS AND MODEL BUILDING

Table Of Contents –

EDA - TARGET VARIABLE: LOAN_STATUS

IN EDA Catplot, Displot, Box Plot, Sub plot,Rel Plot, HeatMap charts have taken.

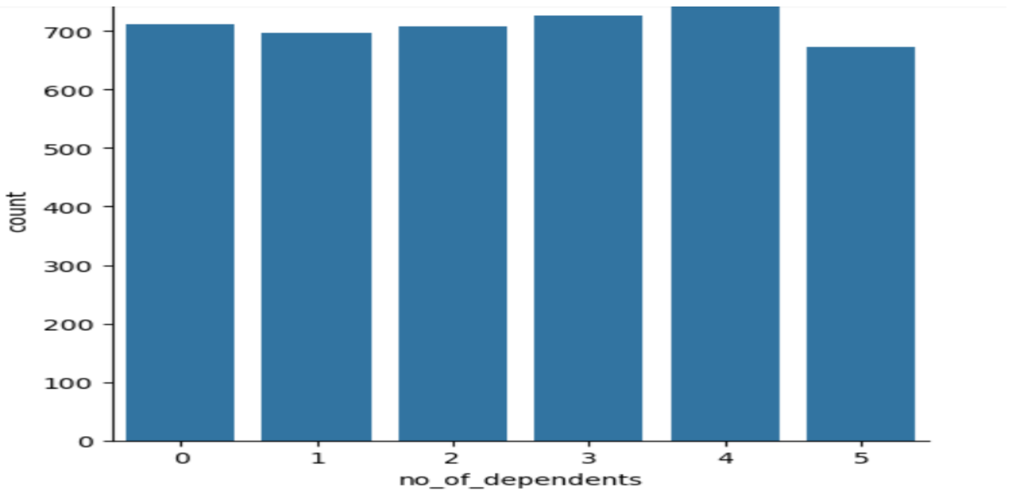
No Of Dependence , Self Employeed status,Loan status,Education Status have found.
Label Encoding has done & Outlier Removal has to be done on Box Plot.
In this way, EDA Is Done.

GRAPHS USED FOR EXPLORATORY DATA ANALYSIS :

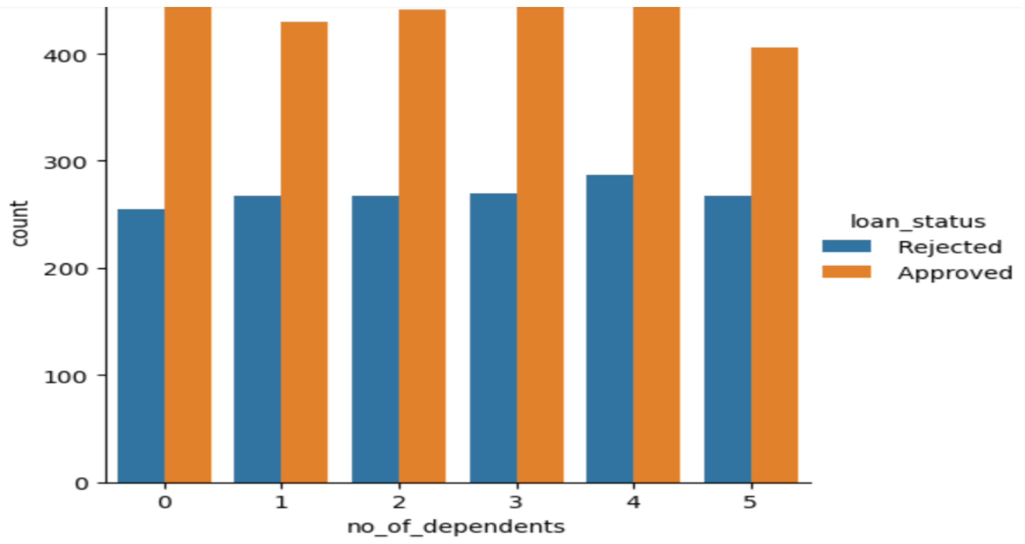
1. Categorical Plot
2. Distribution plot
3. Sub-Plot (Pie Chart)
4. Box Plot
5. Relational Plot
6. Heat – Map Plot

CATEGORICAL PLOTS ARE SHOWN HERE

Insight : From this cat plot with no of dep =4 is highest and dep=5 is lowest

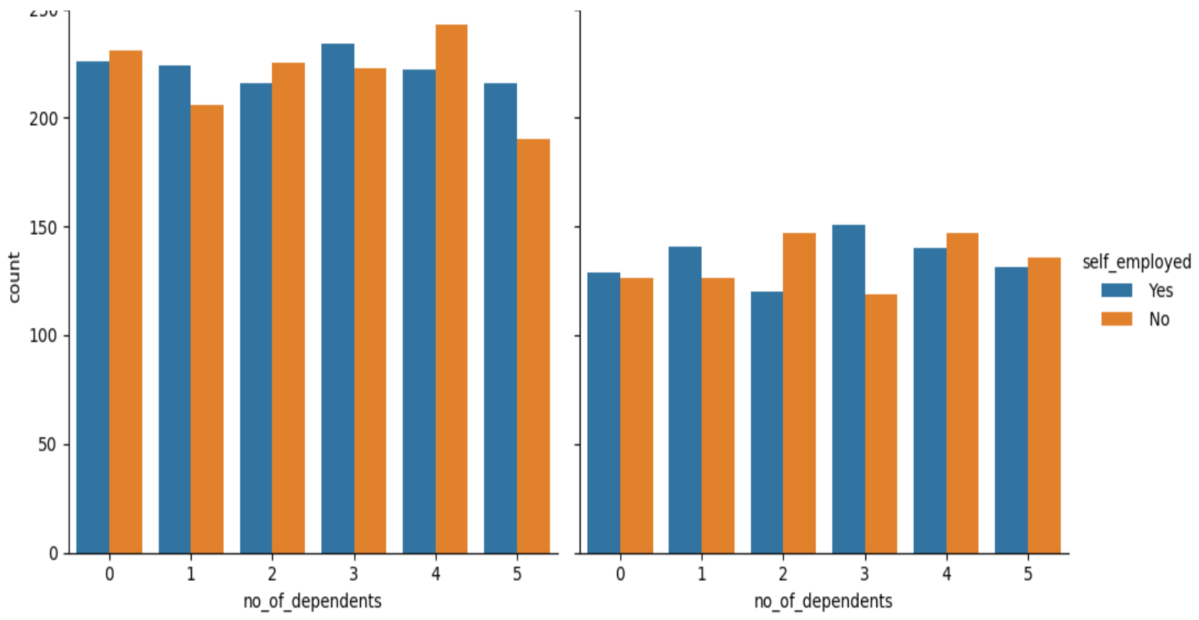


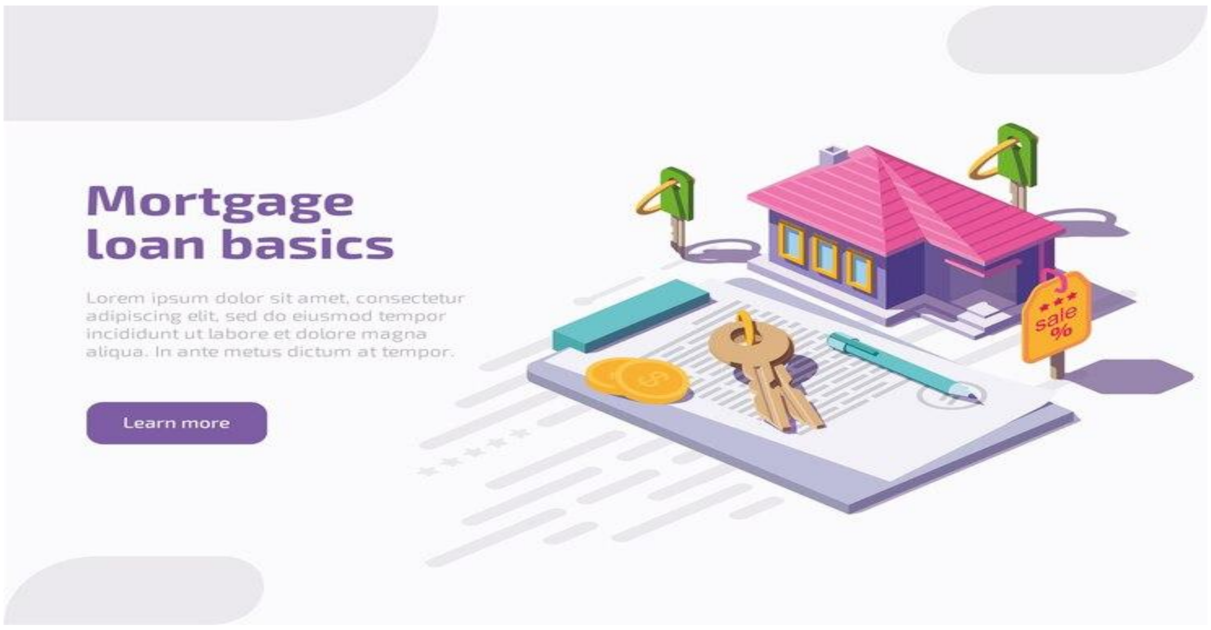
Insight : From here we can observe if the person has less dependents the probability that his loan will be approved is more .



CATEGORICAL PLOTS ARE SHOWN HERE

Insight : Here, no_of_dependents of loan_status = Approved chart has more count as compared to loan_status = Rejected.

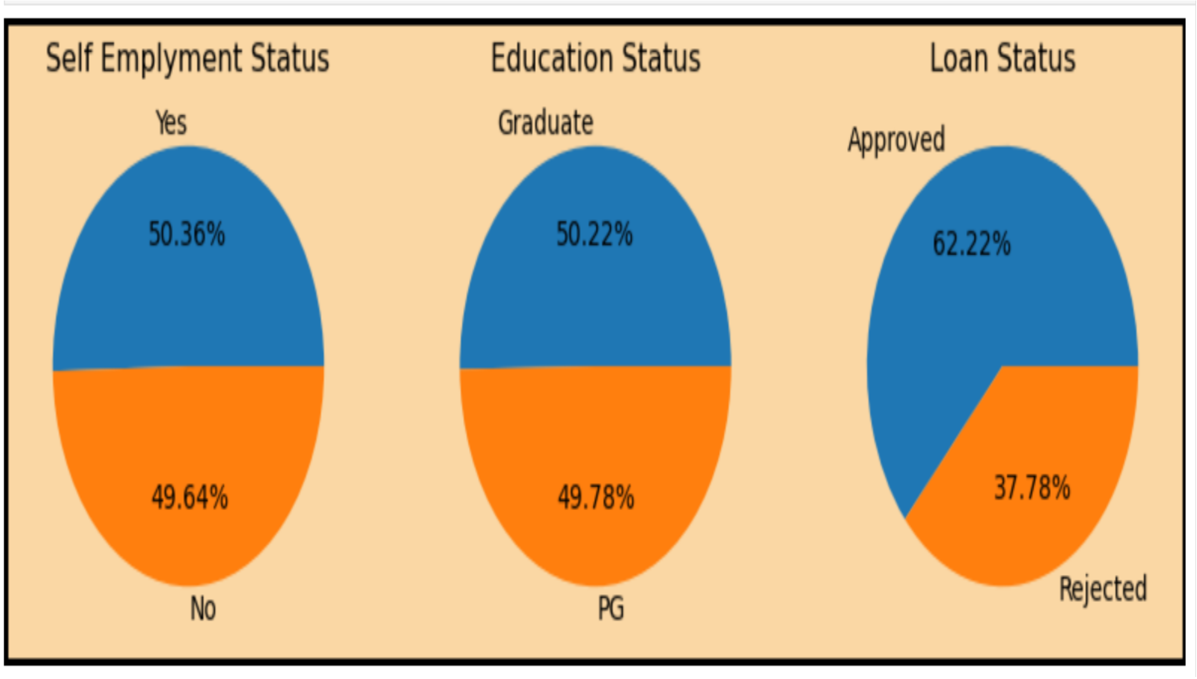




Mortgage loan basics
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. In ante metus dictum at tempor.
Learn more
sale %

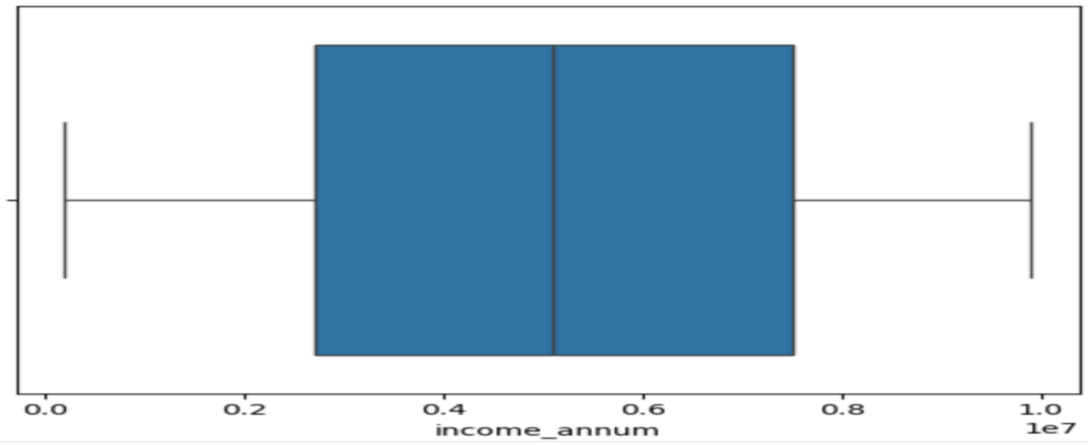
HERE, SUBPLOT ARE SHOWN. IN THAT, THERE ARE PIE CHART ARE SHOWN.

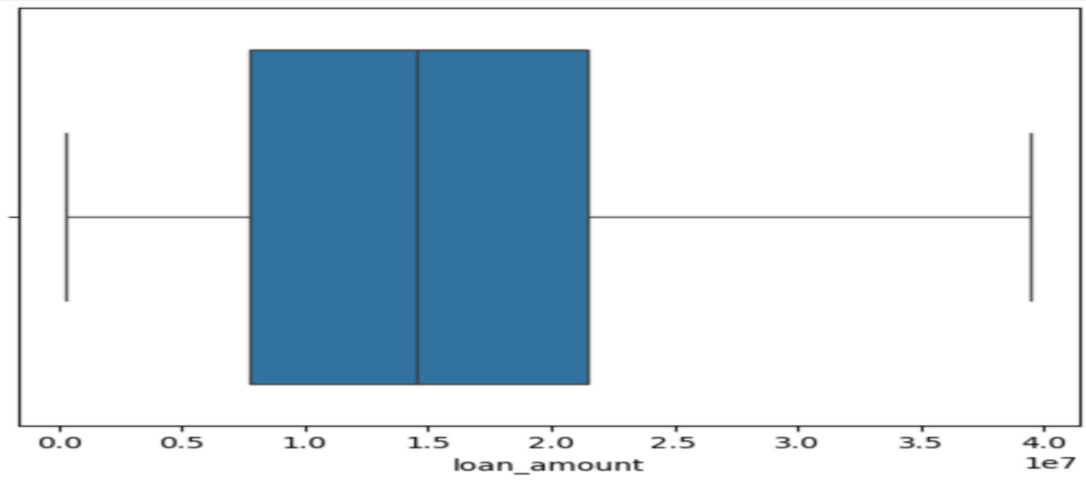
- Insights : Here, Self Employment Status is more as compared to non-self employment status. Education Status of Graduate is more as compared to PG Student. So, finally the Loan Status Of Approved is more i.e. 62.22% as compared to Rejected i.e. 37.78%



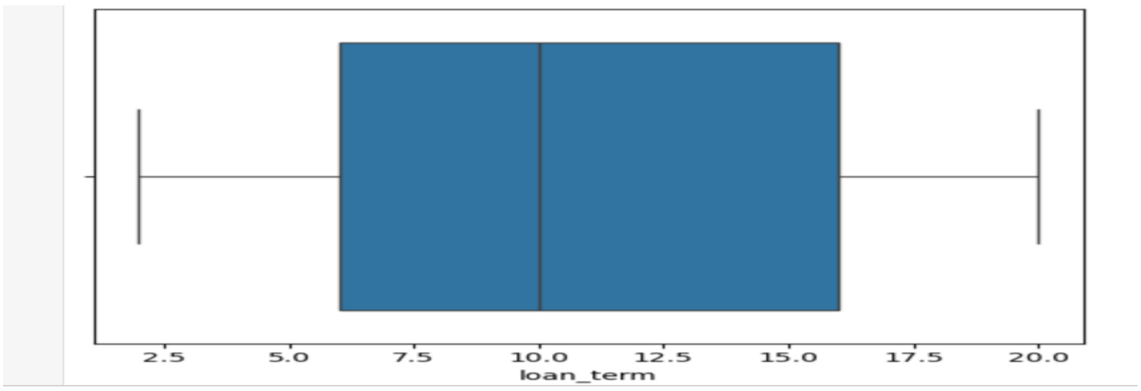
BOX PLOT ARE SHOWN HERE

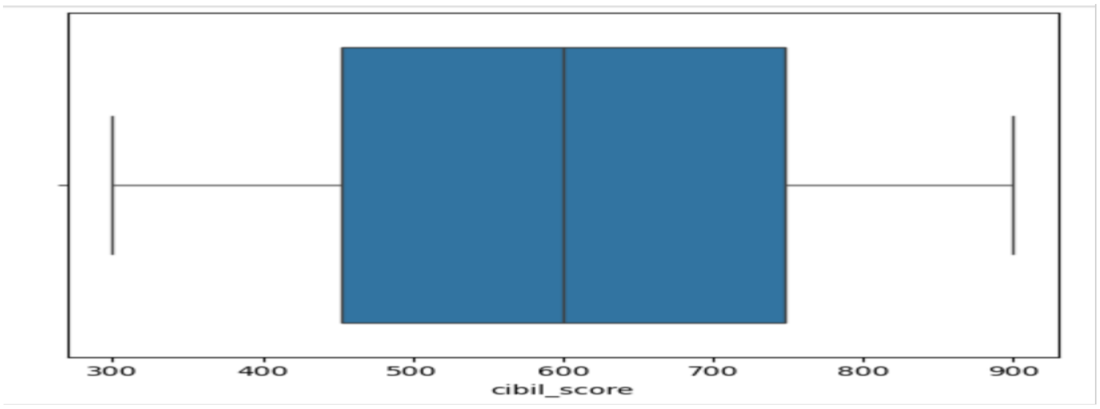
Insights : Here, Box Plot have been shown for income_annum, loan_amount, loan_term, cibil_score, residential_assets_value, commercial_assets_value, luxury_assets_value & bank_asset_value.

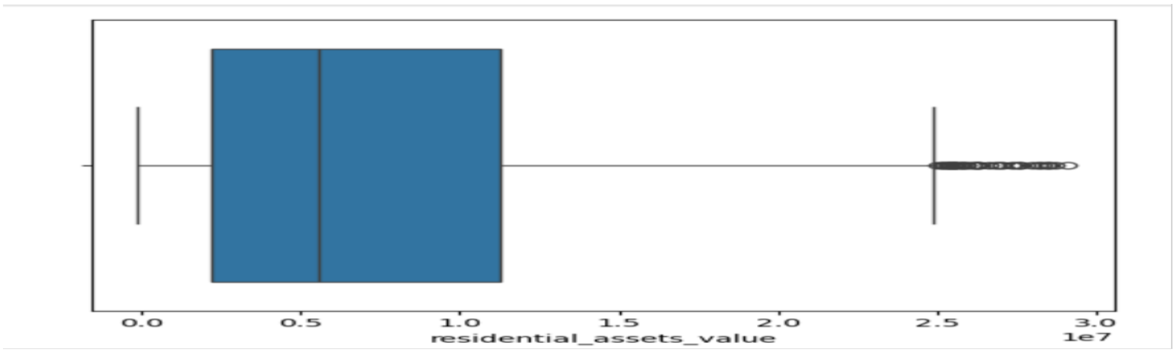


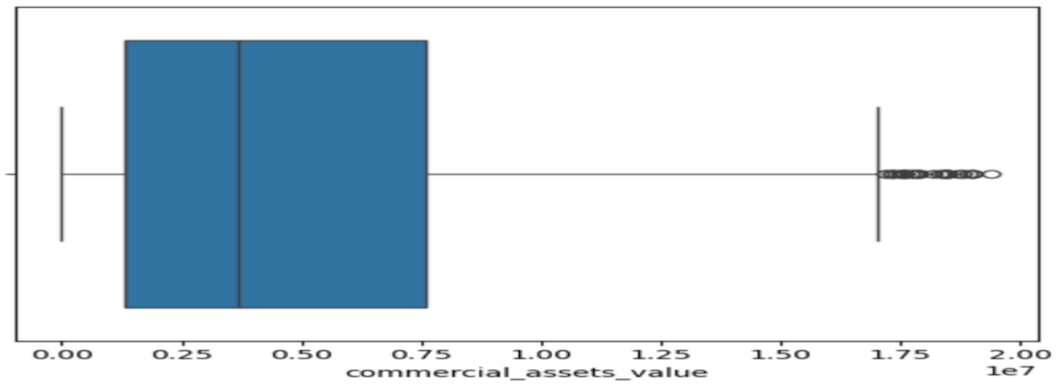


BOX PLOT ARE SHOWN HERE

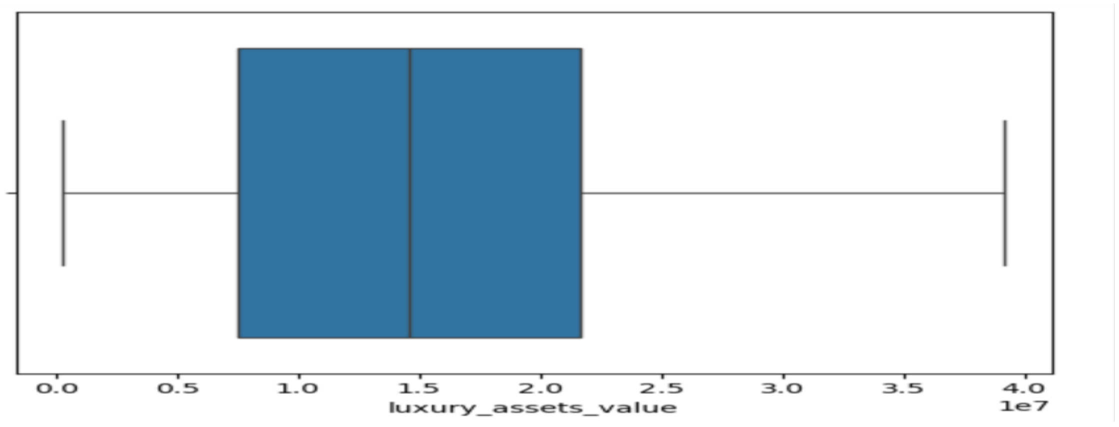


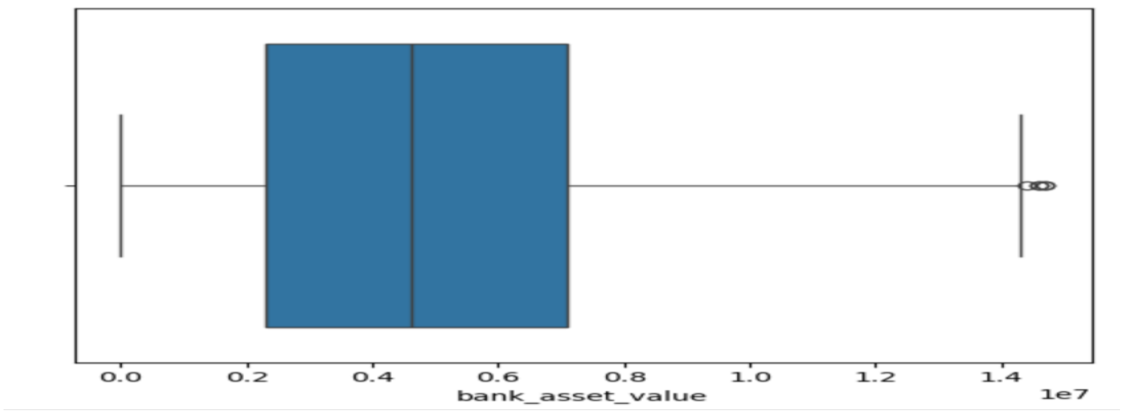


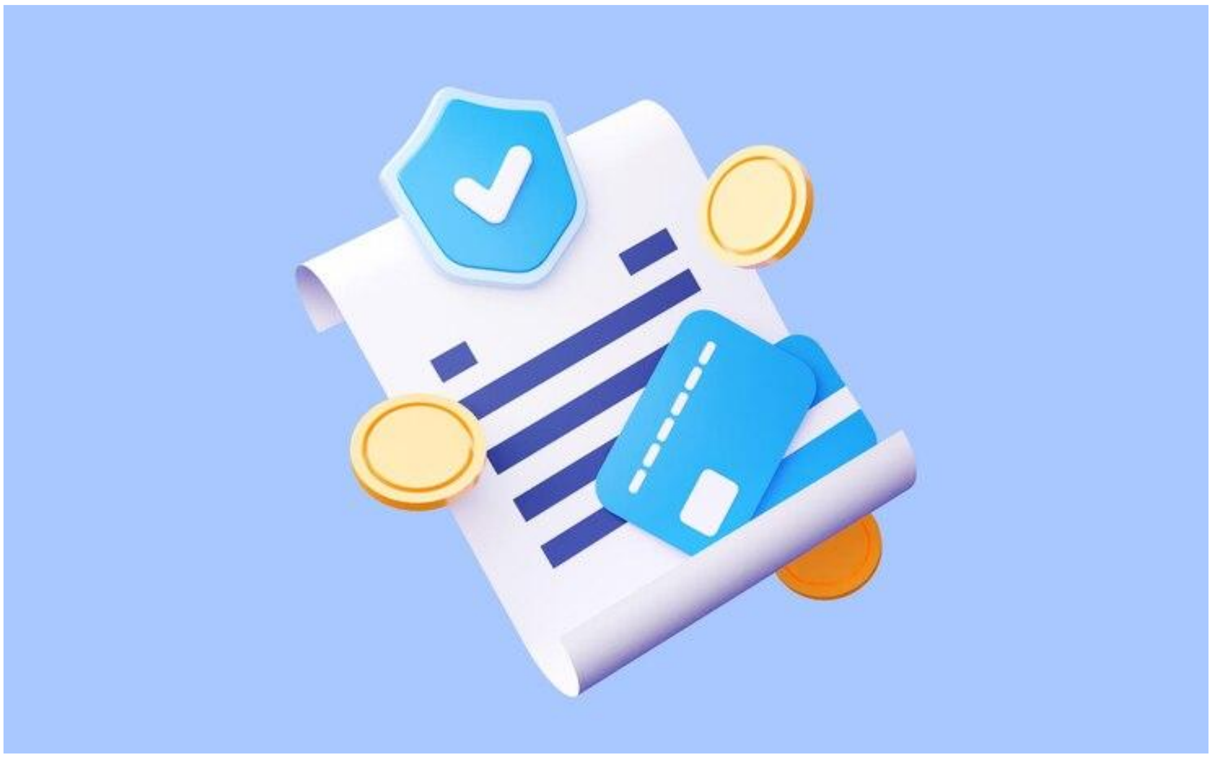




BOX PLOT ARE SHOWN HERE

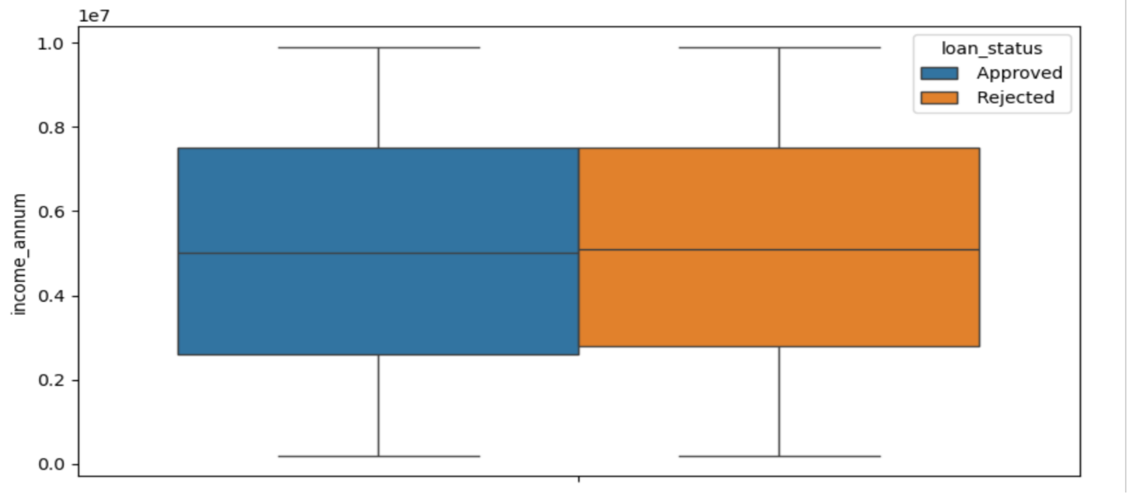


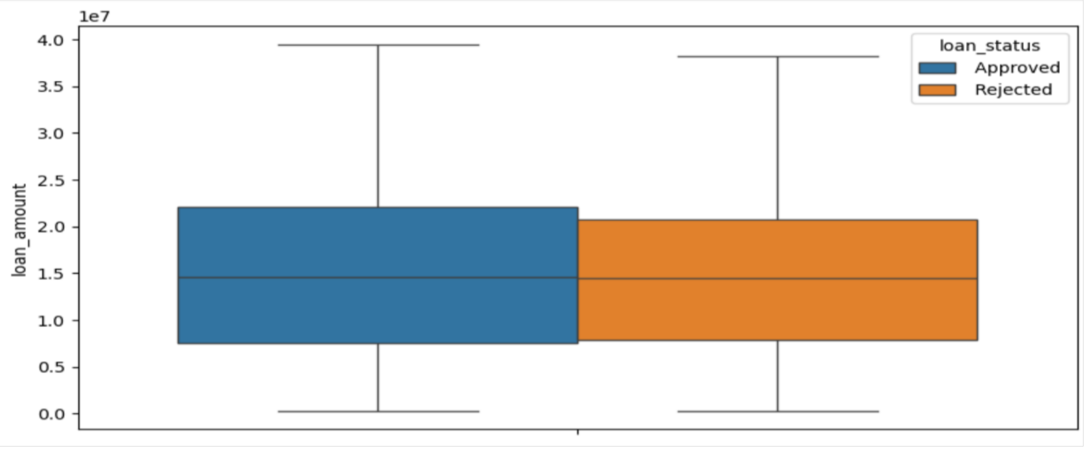




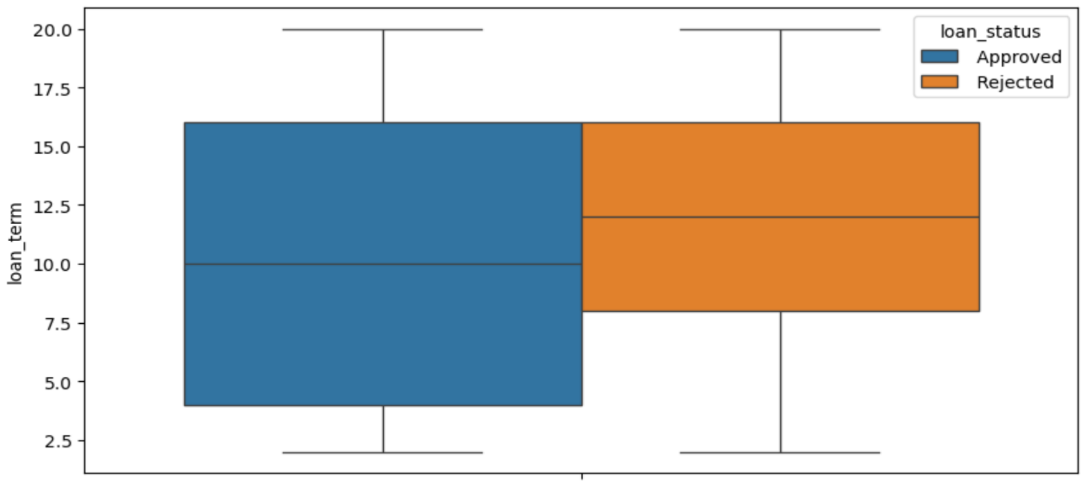
BOX PLOT ARE SHOWN HERE

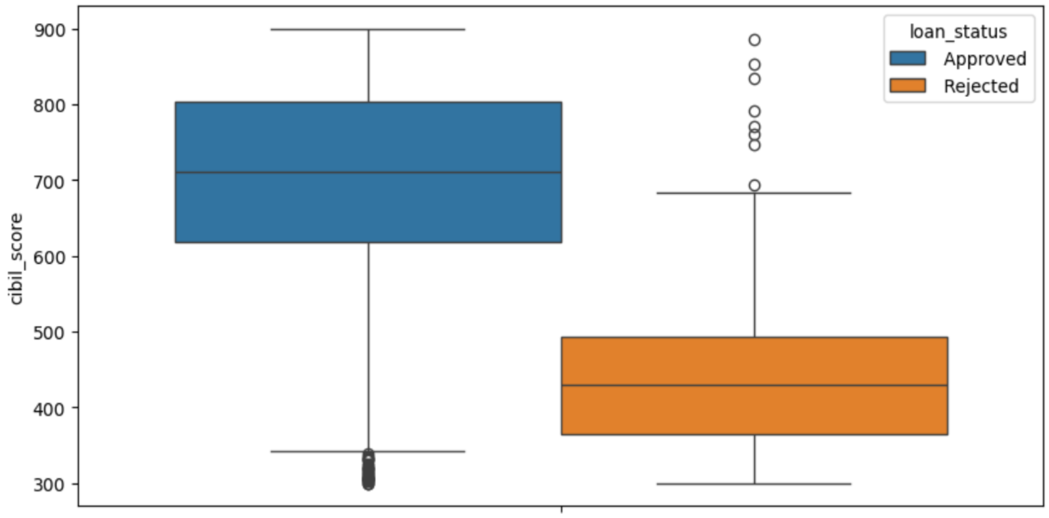
Insights : Here, Box plot of loan_status have been shown. Approved loan_status have shown in Blue Color & Rejected loan_status have shown in Orange Color. Here, loan_status are shown for different categories differently.

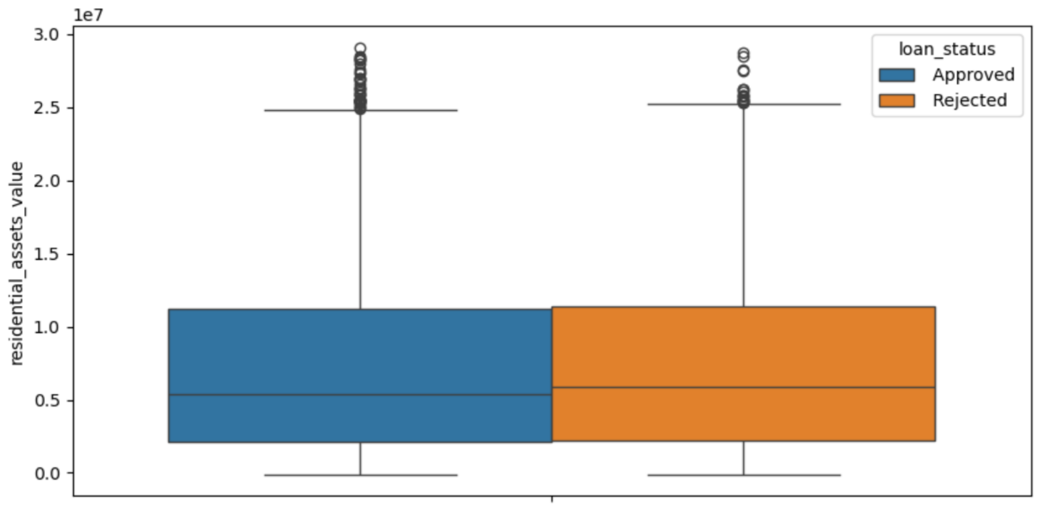




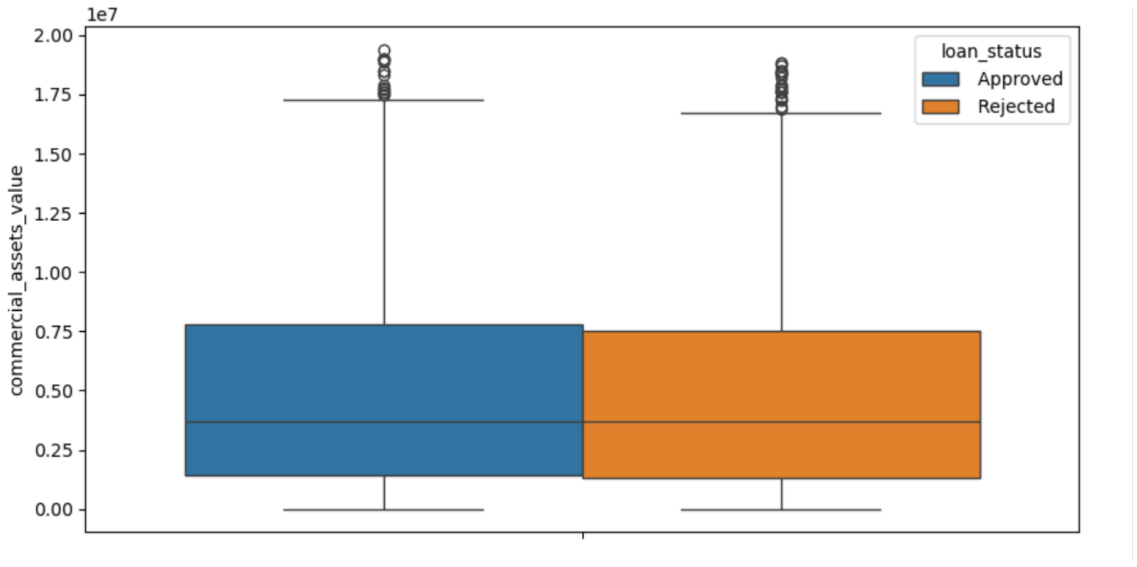
BOX PLOT ARE SHOWN HERE

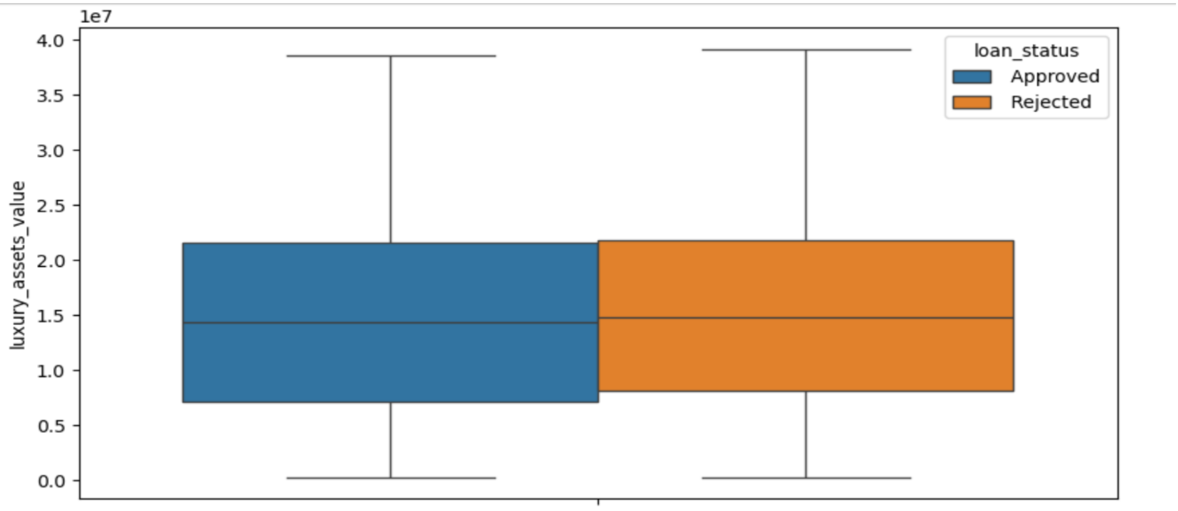


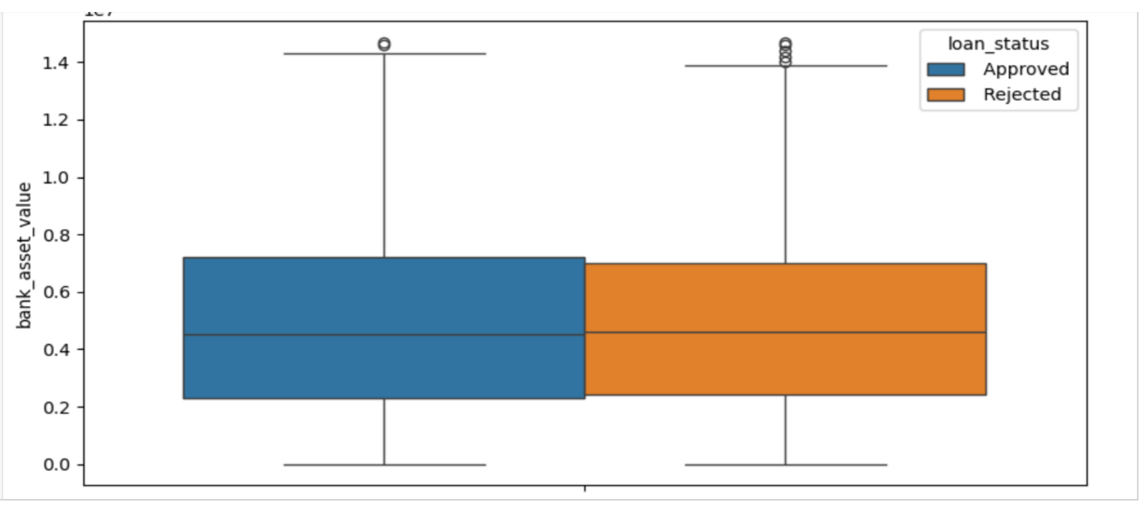




BOX PLOT ARE SHOWN HERE

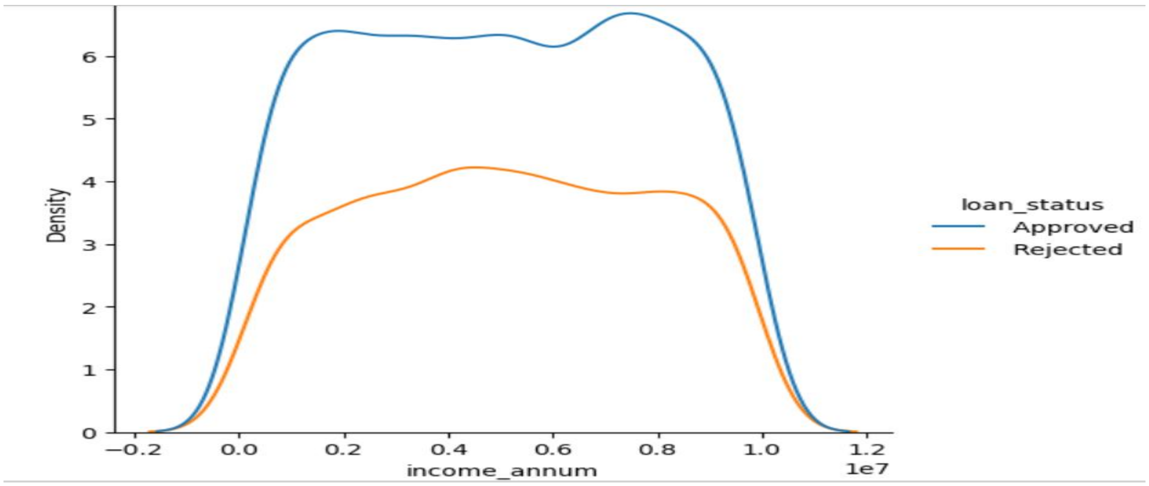


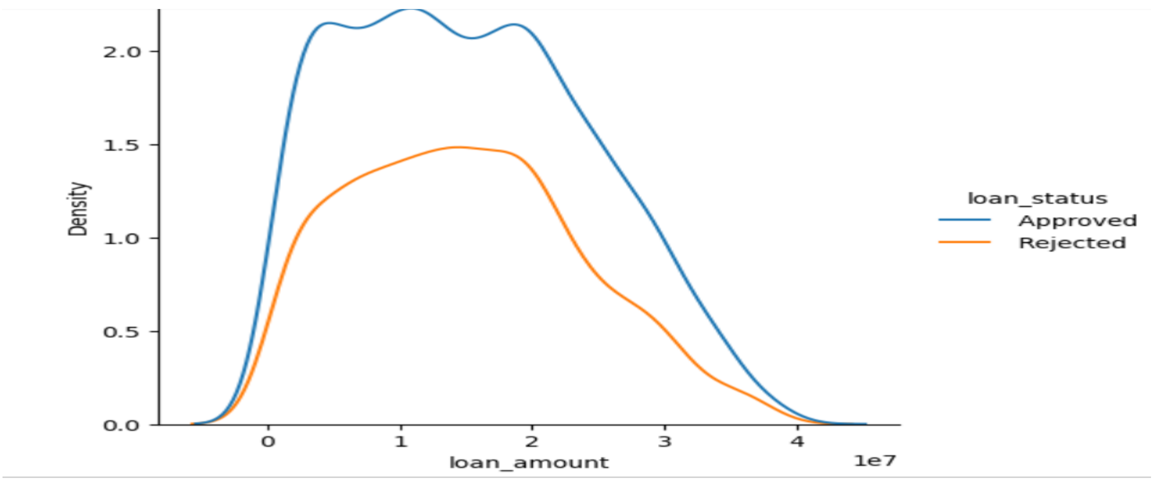




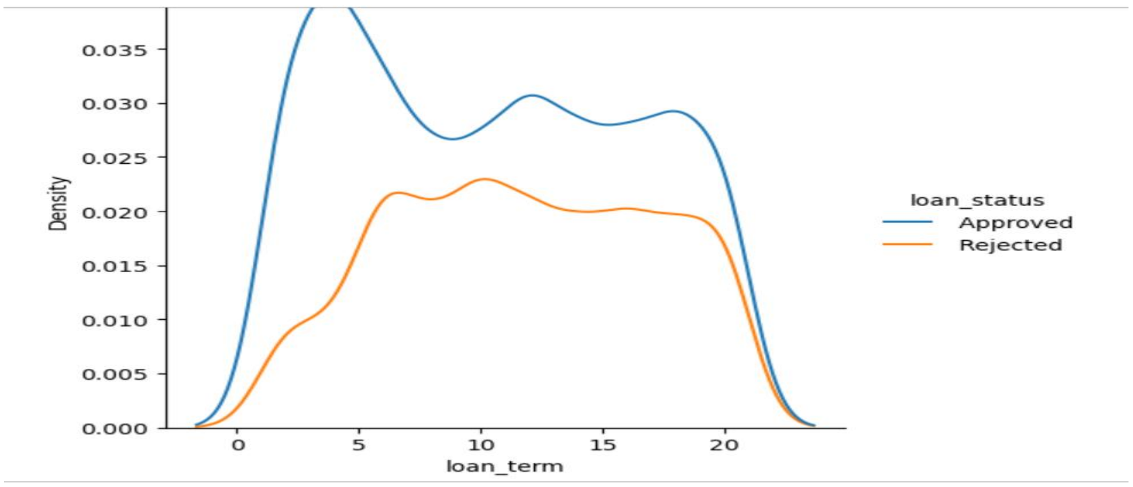
DISTRIBUTION PLOT ARE SHOWN HERE

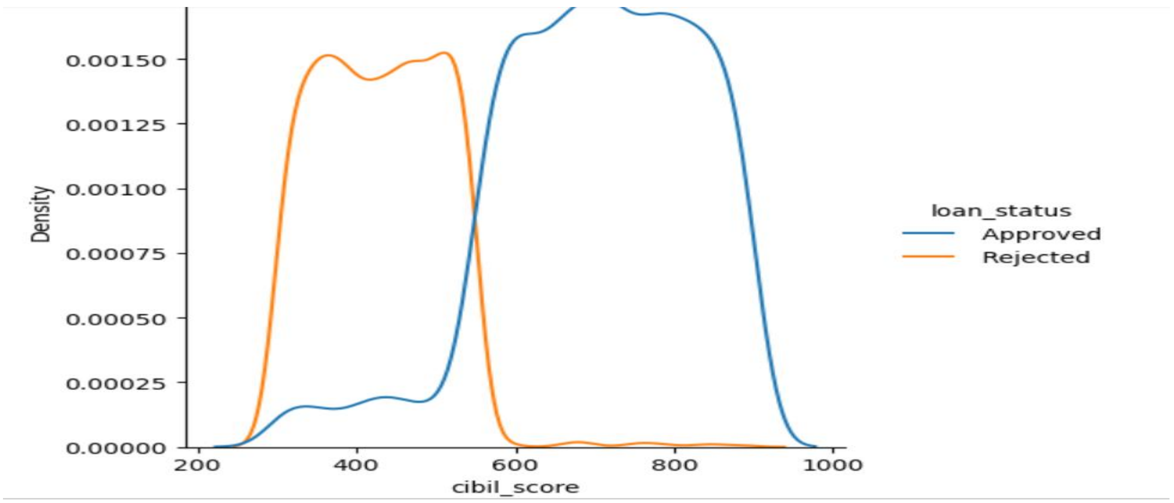
Insights : Here, Distribution Plot shows the Density wise loan_status. Also, loan_status of Approved have shown in Blue color & loan_status of Rejected have shown in Orange color by list .

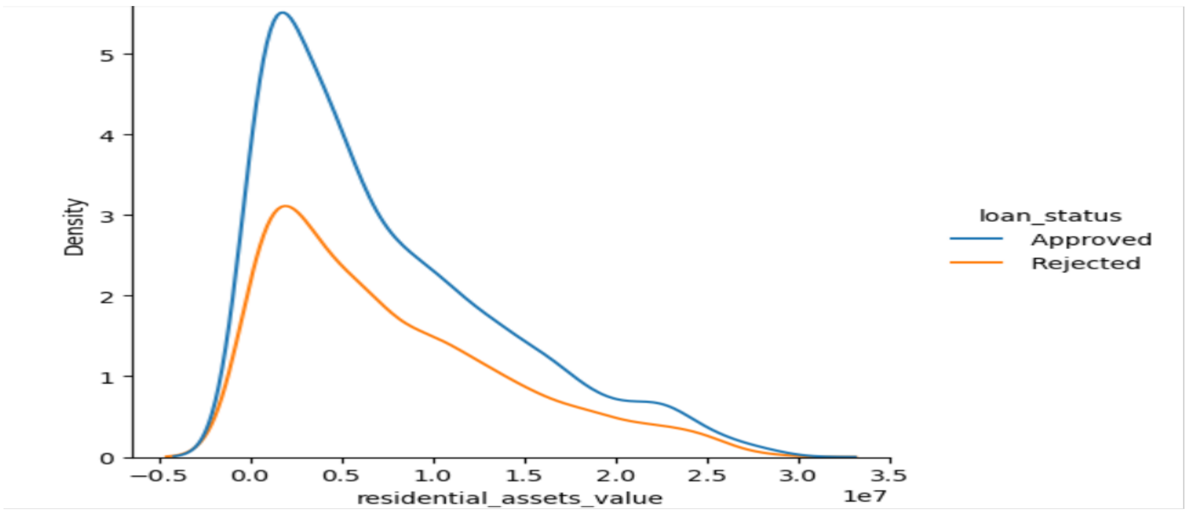




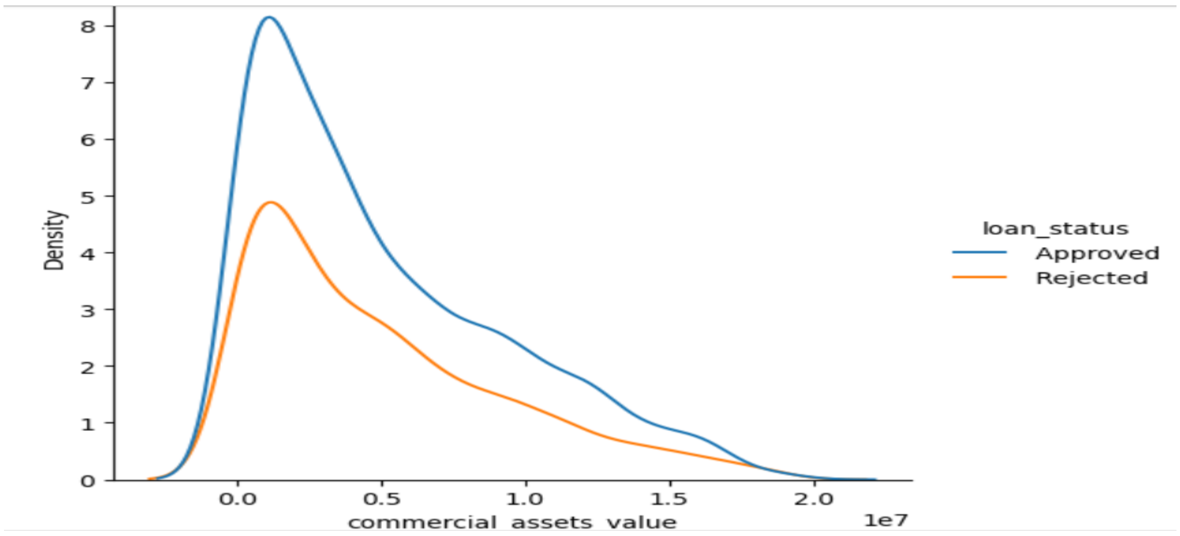
DISTRIBUTION PLOT ARE SHOWN HERE

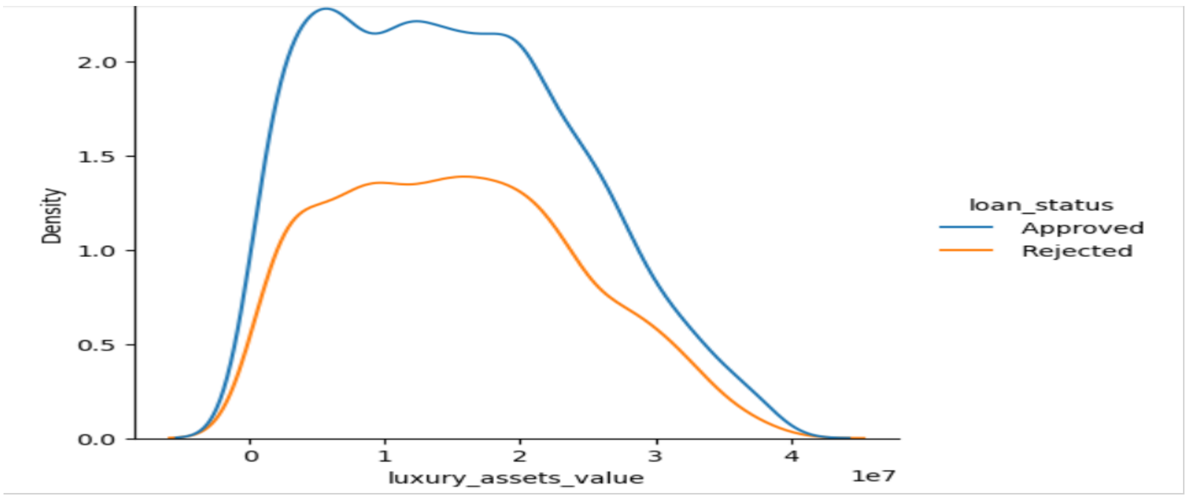


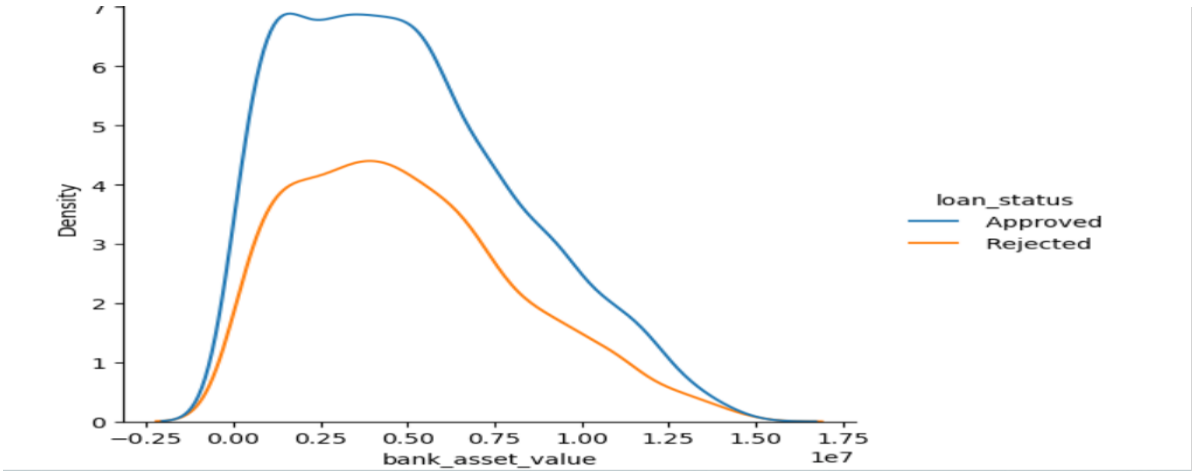




DISTRIBUTION PLOT ARE SHOWN HERE

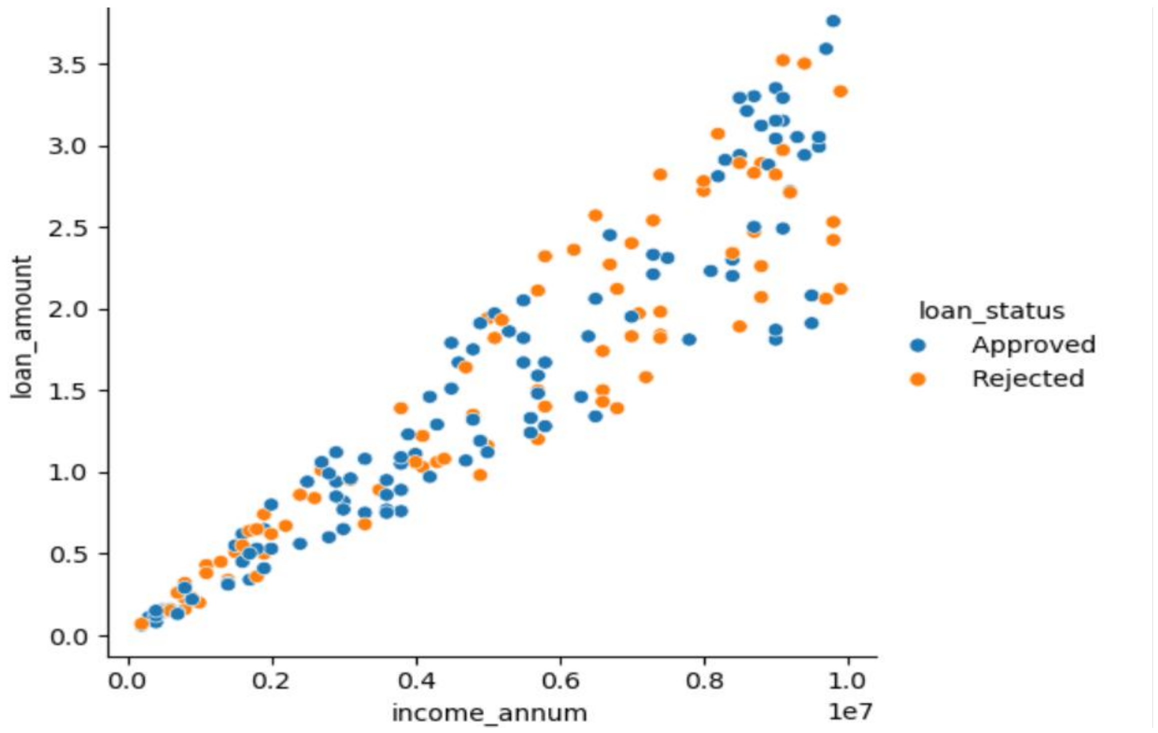


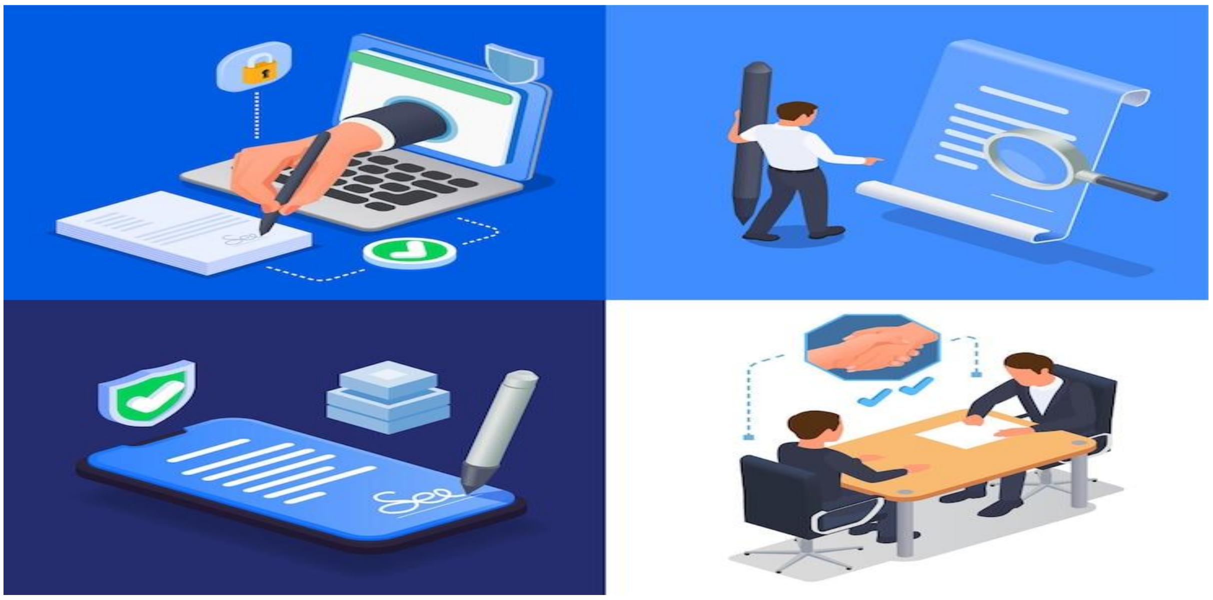




RELATIONAL PLOT ARE SHOWN HERE

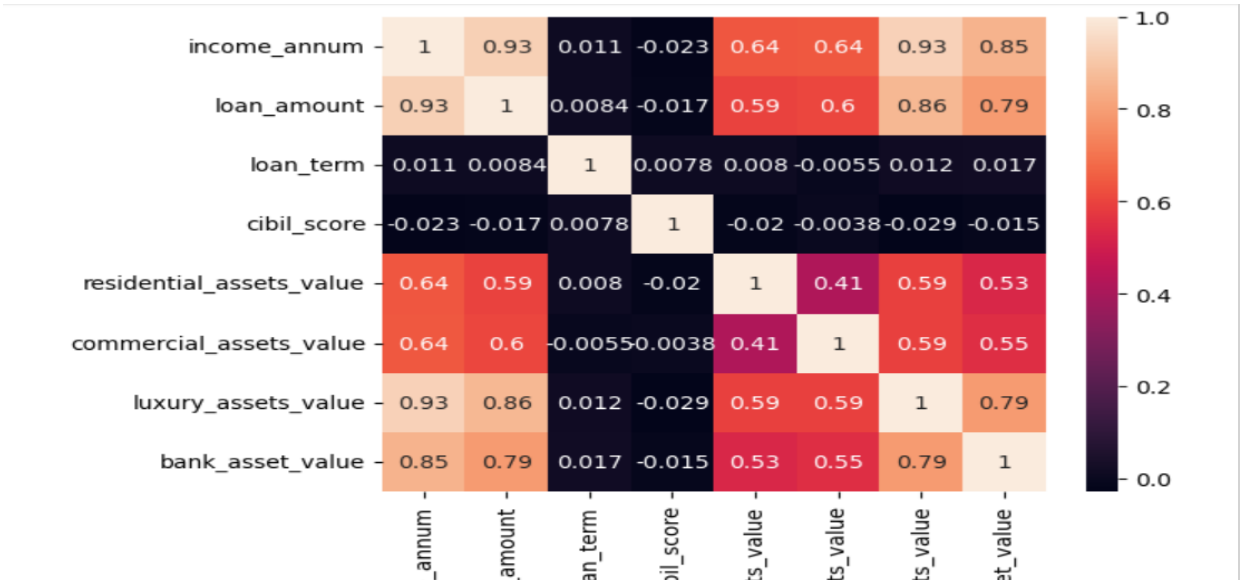
Insight : As the income increases the loan amount also increases .





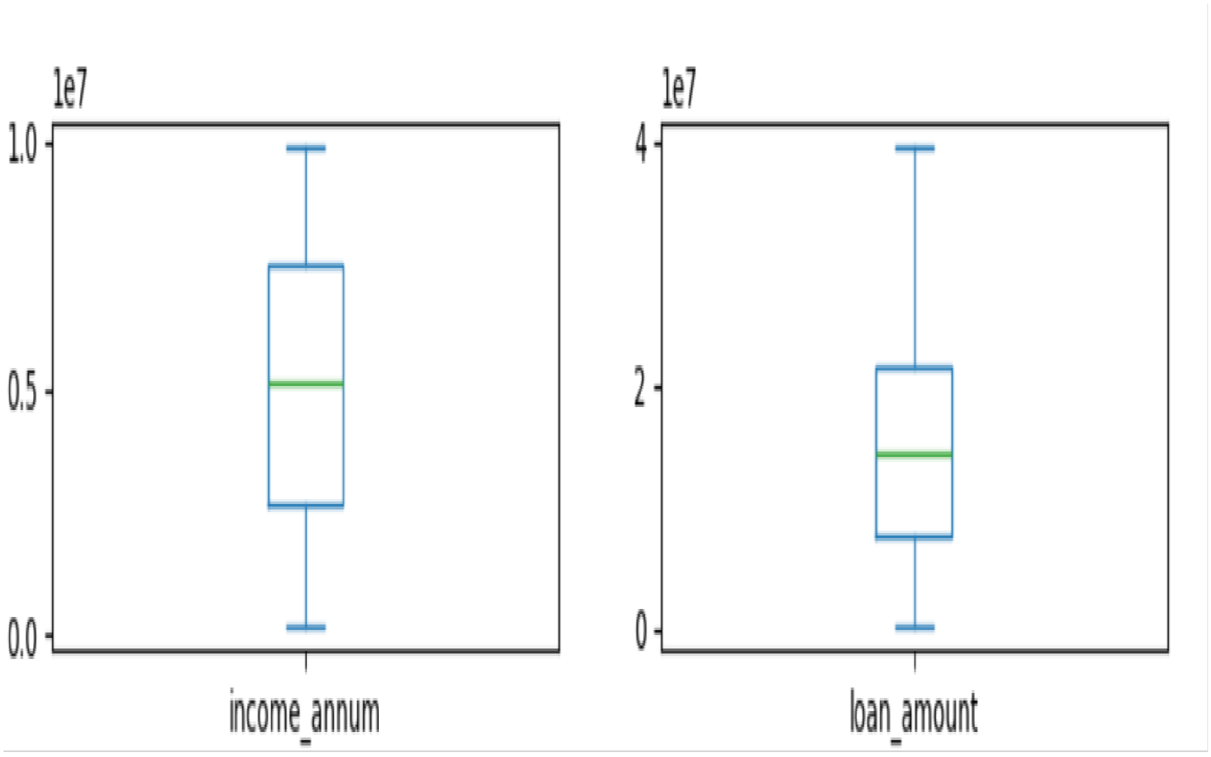
HEAT MAP ARE SHOWN HERE

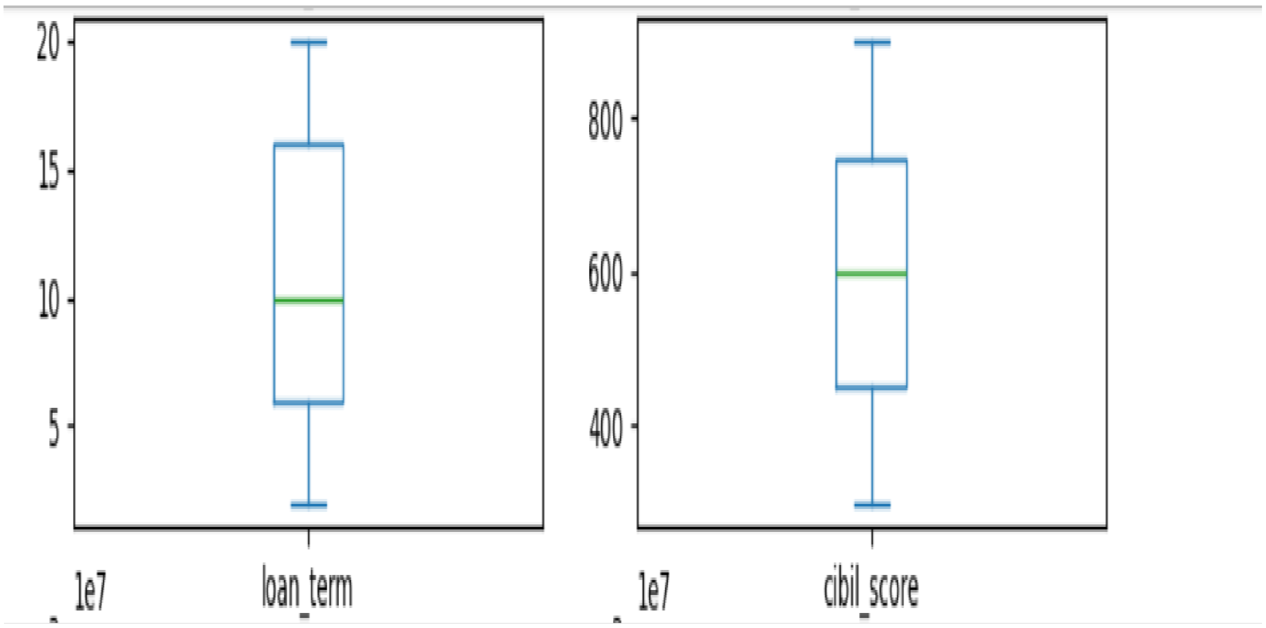
Insights : Here, Heat Map shows the Correlation Matrix. From this we can observe that most of the asset values and annual income are highly correlated. cibil score, loan term has no relation with annual income .



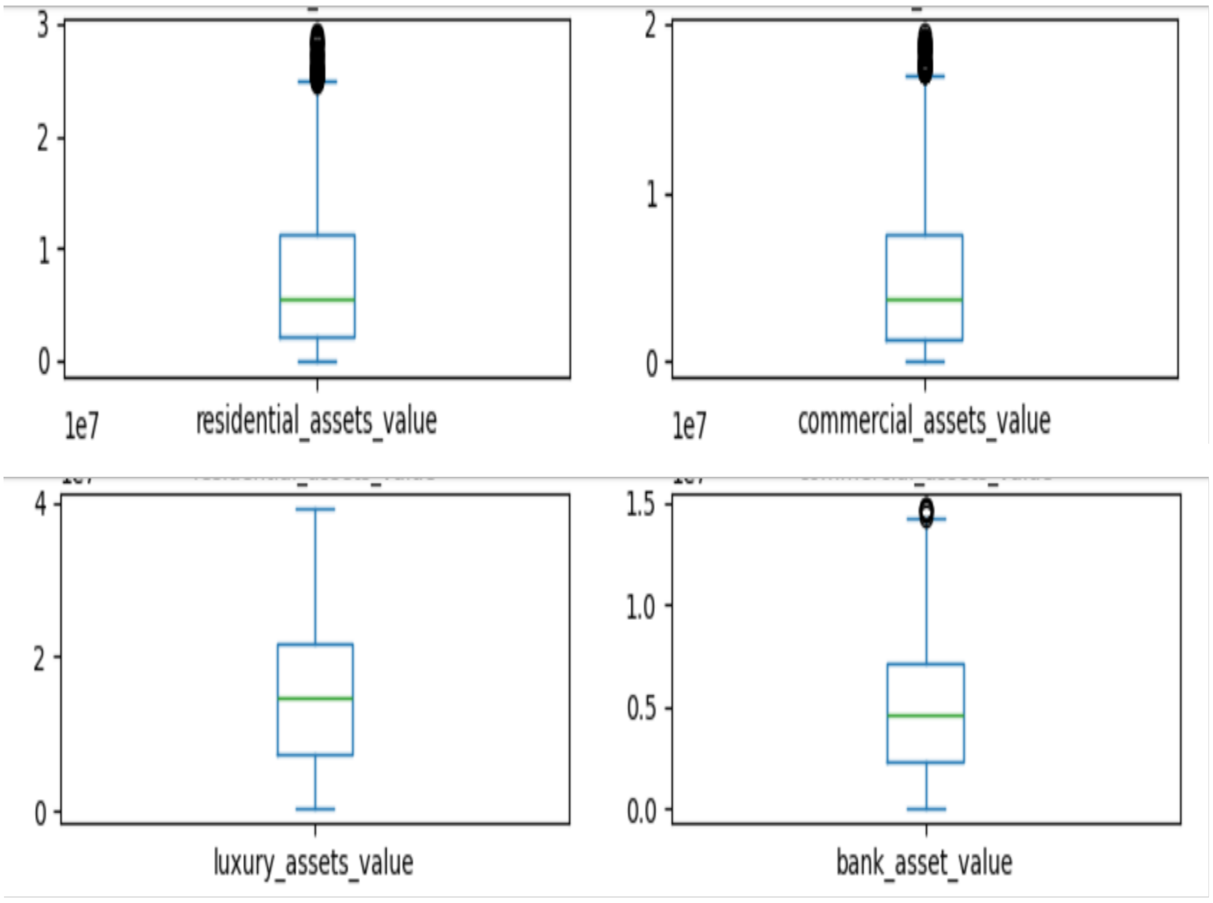
BOX PLOT ARE SHOWN HERE

Insights : Here, Box Plot shows the Outlier Removal Treatment.





BOX PLOT ARE SHOWN HERE

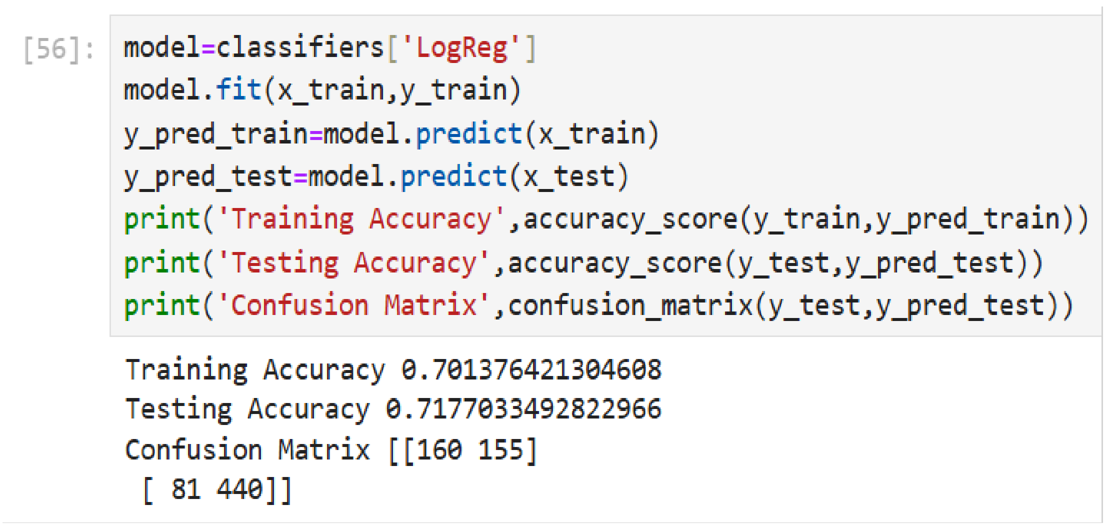


HERE, LABEL ENCODING, SCALING & HYPERPARAMETER TUNNING ARE USED

Algorithms used :

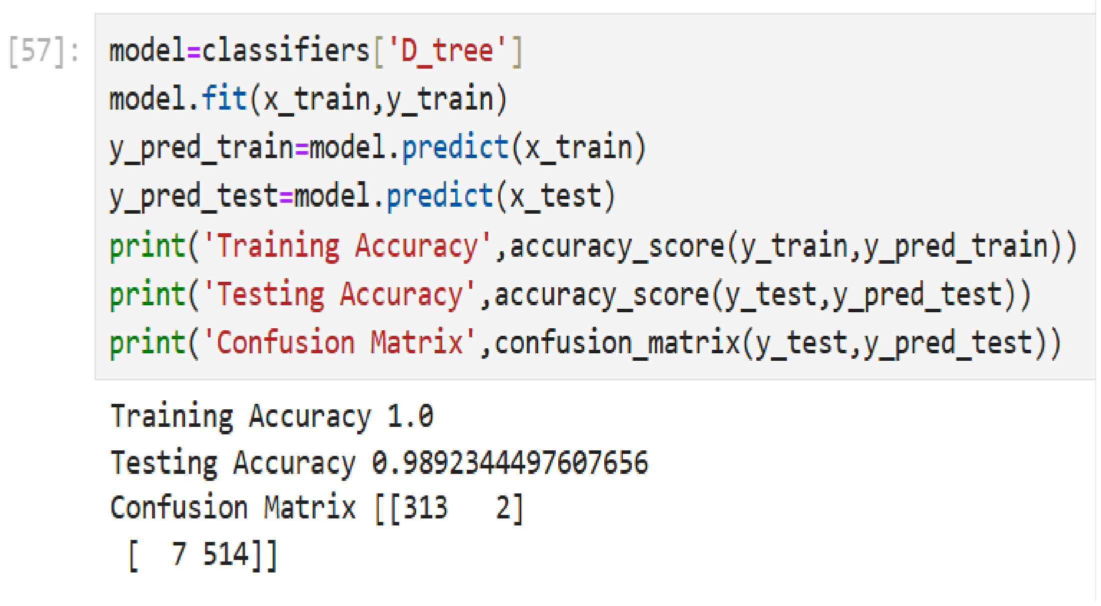
1. Logistic Regression
2. Decision Tree
3. Random Forest
4. SVM
5. Naive-Bayes
6. K-Nearest Neighbors

ALGORITHMS & THEIR ACCURACY



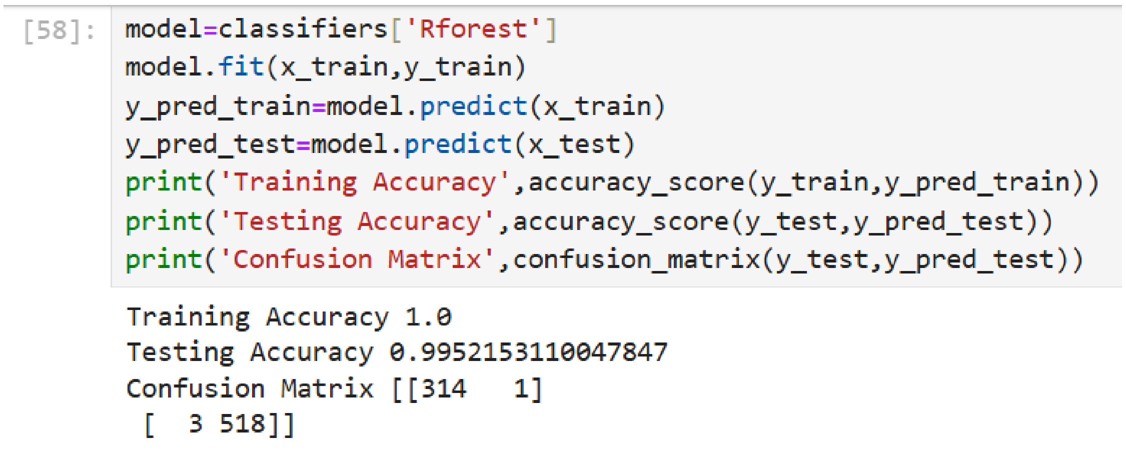
```
[56]: model=classifiers['LogReg']
      model.fit(x_train,y_train)
      y_pred_train=model.predict(x_train)
      y_pred_test=model.predict(x_test)
      print('Training Accuracy',accuracy_score(y_train,y_pred_train))
      print('Testing Accuracy',accuracy_score(y_test,y_pred_test))
      print('Confusion Matrix',confusion_matrix(y_test,y_pred_test))
```

```
Training Accuracy 0.701376421304608
Testing Accuracy 0.7177033492822966
Confusion Matrix [[160 155]
 [ 81 440]]
```



```
[57]: model=classifiers['D_tree']
      model.fit(x_train,y_train)
      y_pred_train=model.predict(x_train)
      y_pred_test=model.predict(x_test)
      print('Training Accuracy',accuracy_score(y_train,y_pred_train))
      print('Testing Accuracy',accuracy_score(y_test,y_pred_test))
      print('Confusion Matrix',confusion_matrix(y_test,y_pred_test))
```

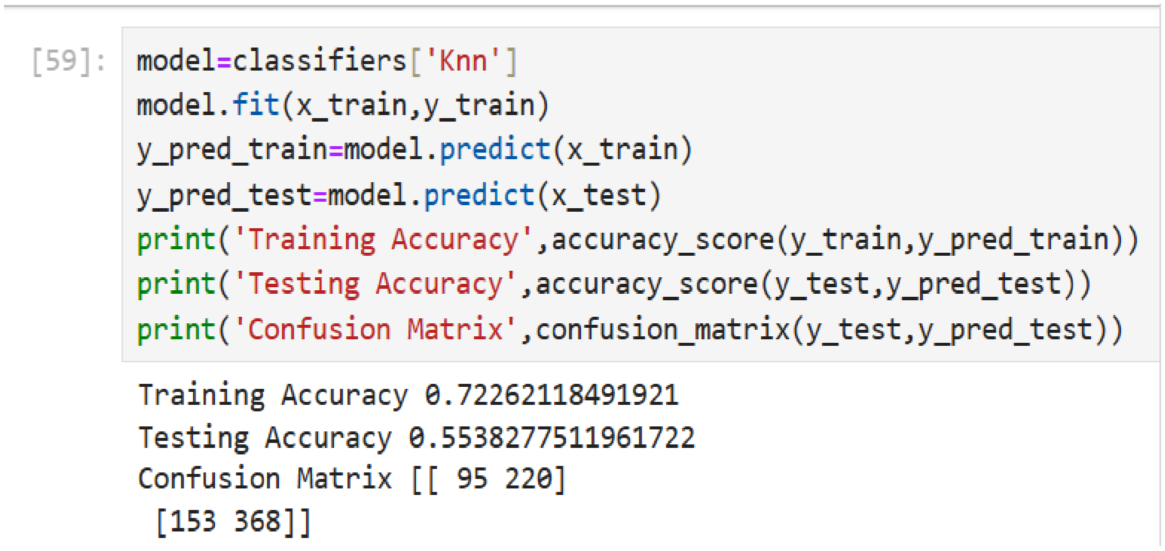
```
Training Accuracy 1.0
Testing Accuracy 0.9892344497607656
Confusion Matrix [[313   2]
 [  7 514]]
```



```
[58]: model=classifiers['Rforest']
      model.fit(x_train,y_train)
      y_pred_train=model.predict(x_train)
      y_pred_test=model.predict(x_test)
      print('Training Accuracy',accuracy_score(y_train,y_pred_train))
      print('Testing Accuracy',accuracy_score(y_test,y_pred_test))
      print('Confusion Matrix',confusion_matrix(y_test,y_pred_test))
```

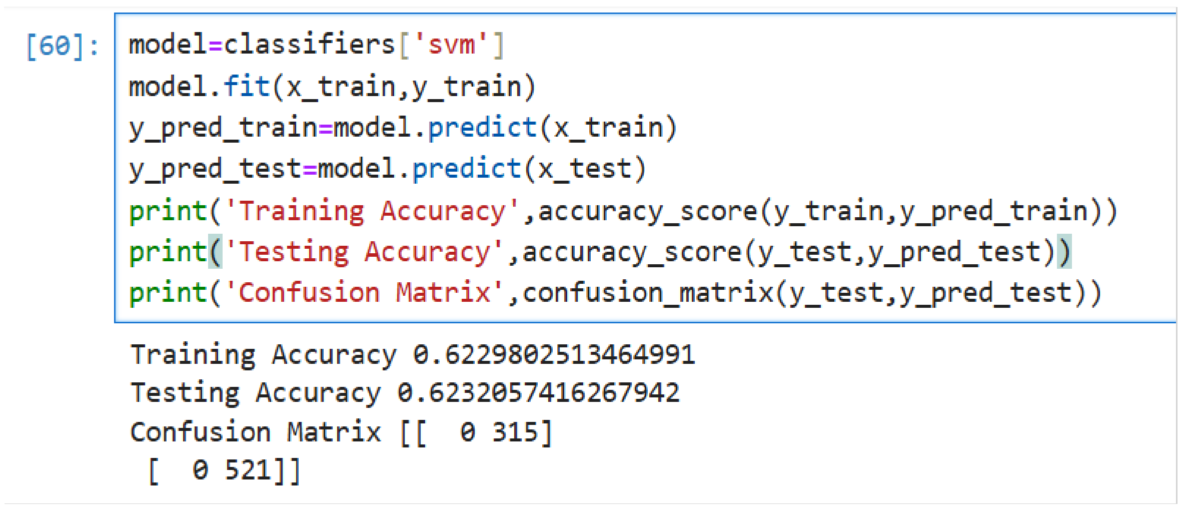
```
Training Accuracy 1.0
Testing Accuracy 0.9952153110047847
Confusion Matrix [[314   1]
 [  3 518]]
```

ALGORITHMS & THEIR ACCURACY



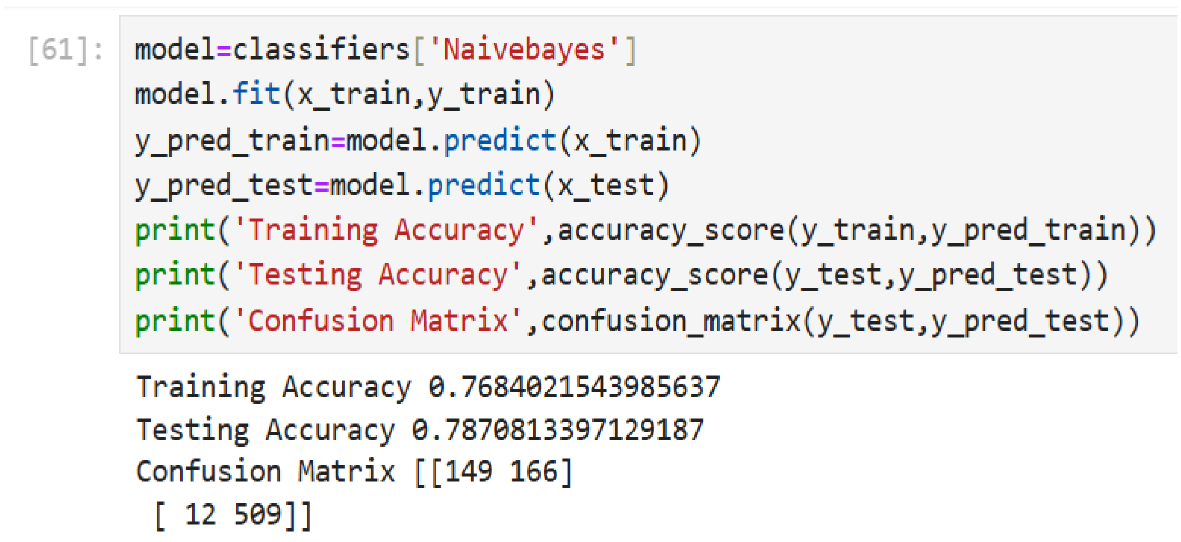
```
[59]: model=classifiers['Knn']
      model.fit(x_train,y_train)
      y_pred_train=model.predict(x_train)
      y_pred_test=model.predict(x_test)
      print('Training Accuracy',accuracy_score(y_train,y_pred_train))
      print('Testing Accuracy',accuracy_score(y_test,y_pred_test))
      print('Confusion Matrix',confusion_matrix(y_test,y_pred_test))
```

```
Training Accuracy 0.72262118491921
Testing Accuracy 0.5538277511961722
Confusion Matrix [[ 95 220]
 [153 368]]
```



```
[60]: model=classifiers['svm']
      model.fit(x_train,y_train)
      y_pred_train=model.predict(x_train)
      y_pred_test=model.predict(x_test)
      print('Training Accuracy',accuracy_score(y_train,y_pred_train))
      print('Testing Accuracy',accuracy_score(y_test,y_pred_test))
      print('Confusion Matrix',confusion_matrix(y_test,y_pred_test))
```

```
Training Accuracy 0.6229802513464991
Testing Accuracy 0.6232057416267942
Confusion Matrix [[  0 315]
 [  0 521]]
```



```
[61]: model=classifiers['Naivebayes']
      model.fit(x_train,y_train)
      y_pred_train=model.predict(x_train)
      y_pred_test=model.predict(x_test)
      print('Training Accuracy',accuracy_score(y_train,y_pred_train))
      print('Testing Accuracy',accuracy_score(y_test,y_pred_test))
      print('Confusion Matrix',confusion_matrix(y_test,y_pred_test))
```

```
Training Accuracy 0.7684021543985637
Testing Accuracy 0.7870813397129187
Confusion Matrix [[149 166]
 [ 12 509]]
```

CONCLUSION

1.From The above slide we can say that Logistic Regression, Decision Tree, Random Forest Classifier, SVM, Naïve Bayes & K Nearest Neighbors gives Accuracy & Confusion Matrix.

2. But, the Decision Tree, Random Forest Classifier & Naïve Bayes gives Best Accuracy.

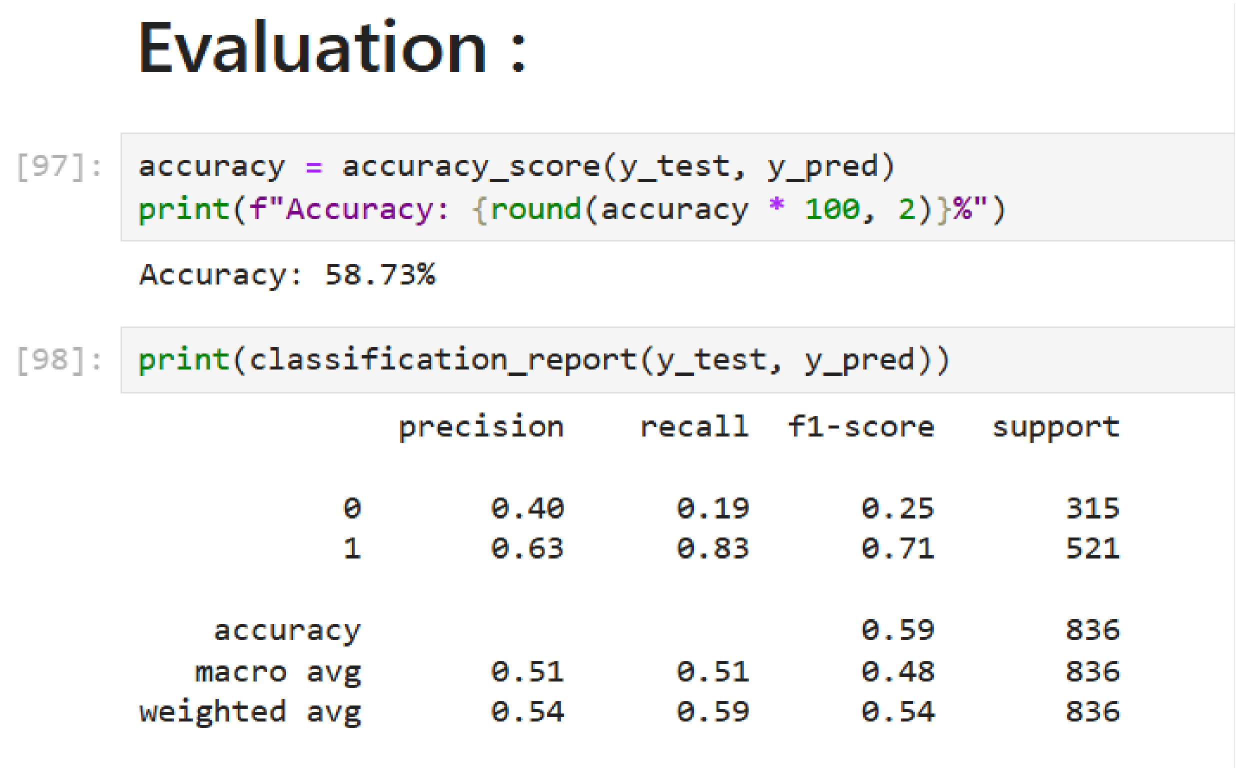
3. I have Applied KNN Algorithm For Classification Report. Precision, Recall, F1-Score, & Support are Good in K Nearest Neighbors Algorithm

4. Accuracy Of Training is 72% & Testing Set Score is 55%. i.e. Our Model have Good Training & Testing Accuracy for KNN Imputer.

EVALUATION

From The Below Evaluation slide, we can say that Precision, Recall, F-1 Score & Support is Good for our Model .

Class 0 Recall Value is 19% i.e It is Low But Good Not So Bad. & Overall Accuracy Of our Model is 58.73% i.e. Low Bot good Not So Bad For KNN Algorithm. So, Our Model has Good Accuracy. Our Model is Good to Perform.



Evaluation :

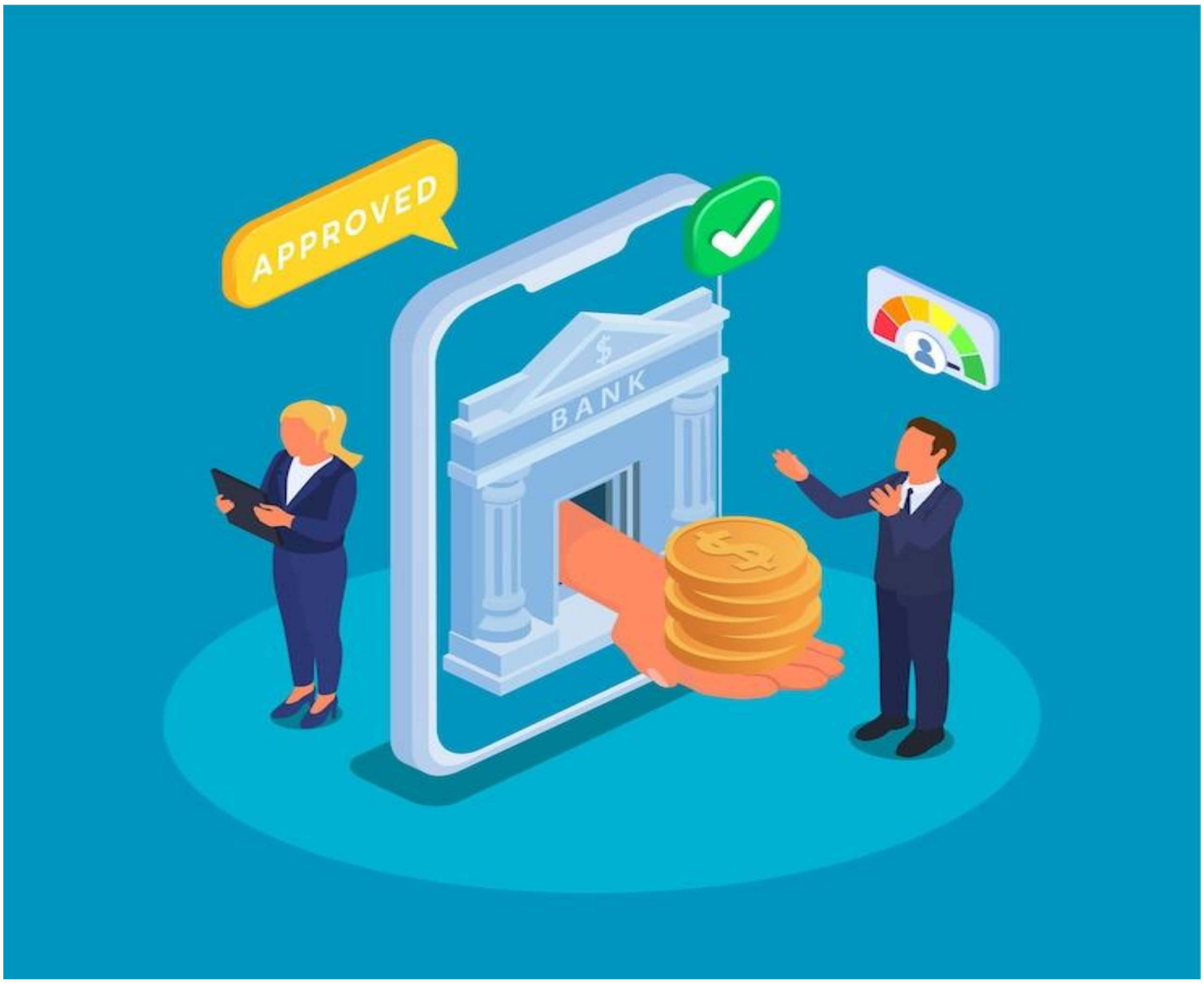
```
[97]: accuracy = accuracy_score(y_test, y_pred)
      print(f"Accuracy: {round(accuracy * 100, 2)}%")

Accuracy: 58.73%
```

```
[98]: print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

           0       0.40      0.19      0.25       315
           1       0.63      0.83      0.71       521

    accuracy                           0.59       836
   macro avg       0.51      0.51      0.48       836
weighted avg       0.54      0.59      0.54       836
```

INSIGHTS

On the Real – Time Data Analysis, Based on Target Column i.e. loan_status. Here, Approved & Rejected Loan_Status are shown by picture.

Loan Status Approved Person



INSIGHTS

Loan Status Rejected Person

FUTURE_SCOPE

1. **Integration with Banking Systems :**
   The model can be integrated into existing banking software to automate real-time loan approvals, making the system more efficient and scalable.
2. . **Expansion to Different Loan Types :**
   The system can be extended to handle various types of loans like home loans, personal loans, car loans, education loans, etc., with tailored prediction models for each.
3. **Incorporation of Additional Features :**
   Future iterations can include more features such as credit scores from

FUTURE_SCOPE

external agencies, social media activity (for informal sector applicants), and alternative income verification.

4. **Model Optimization and Tuning :**
   Advanced hyperparameter tuning and usage of ensemble methods can further improve the performance and reliability of predictions.

5. **Real-time Fraud Detection :**
   The system can be enhanced to flag potentially fraudulent applications based on anomalous patterns in the data.

6. **Deployment on Cloud Platforms :**
   For wider accessibility and scalability, the model can be deployed on using Streamlit library & using App1.py file &

FUTURE_SCOPE

using Anaconda Prompt for deployment.

7. **Mobile and Web Application Development :**
A user-friendly app or web portal can be developed for customers to check eligibility and get instant loan approval predictions.

8. **Continuous Learning :**
The model can be set up to retrain periodically with new data, improving its accuracy and adapting to changing trends and policies.

9. **Regulatory Compliance and Fairness Audits :**
Future systems can include mechanisms to ensure compliance with financial

FUTURE_SCOPE

regulations and conduct fairness audits to avoid biased decisions

BEST ALGORITHM FOR PREDICTION : RANDOM FOREST I.E. TRAINING ACCURACY : 100% & TESTING ACCURACY : 99.28%

THANK YOU!