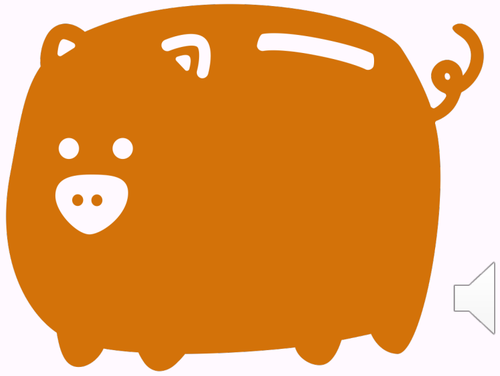
SC1015
Mini Project

Group 4:
Angel Chan Jin Xuan, Peh Yong Qi
Celeste and Sim Hui En Renee Nicole

TABLE OF CONTENTS



Backgound

Applying for loans are an inevitable process that we face in lives. At our stage in life, applying for Built-to-Order (BTO) is something many young couples are considering.

Unfortunately, as many of these applicants face rejection and remain unaware of the specific reasons for their unsuccessful loan applications.

Problem Definition

As such, there is a need for a solution that is able to:

1. accurately assess the probability of loan approval
2. identify key factors influencing rejection
3. provide actionable guidance for applicants to enhance their chances of success

Our Dataset

Source: Kaggle
Title: Loan-Approval-Prediction-Dataset
Author: Archit Sharma
Collaborators: Kai

Exploratory Analysis

PAIRPLOT

- LEGEND:
  - green: loan approved
  - red: loan rejected

OBSERVATION:

The general relationship between relationships can be seen.

SOME VARIABLES HAVE A POSITIVE CORRELATION WITH OTHERS.

Data Visualisation: Numerical

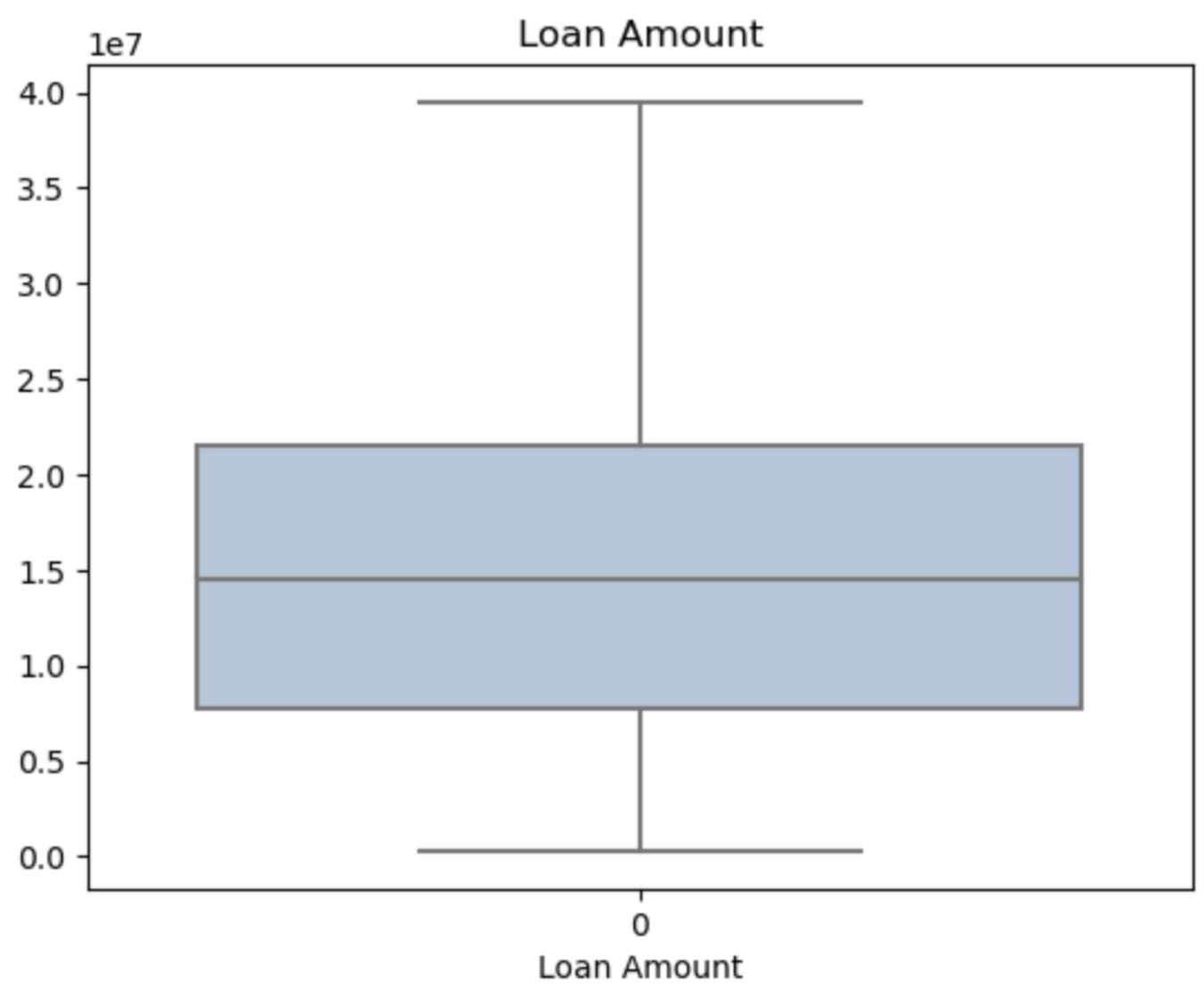
1. LOAN AMOUNT
2. CREDIT SCORE - CIBIL SCORE
3. RESIDENTIAL ASSETS VALUE, COMMERCIAL ASSETS VALUE, LUXRY ASSETS VALUE, BANK VALUE
4. LOAN TERM

1. Loan Amount

From the boxplot, it can be seen that there are no outliers in “loan_amount”

Statistical Description (10^7):

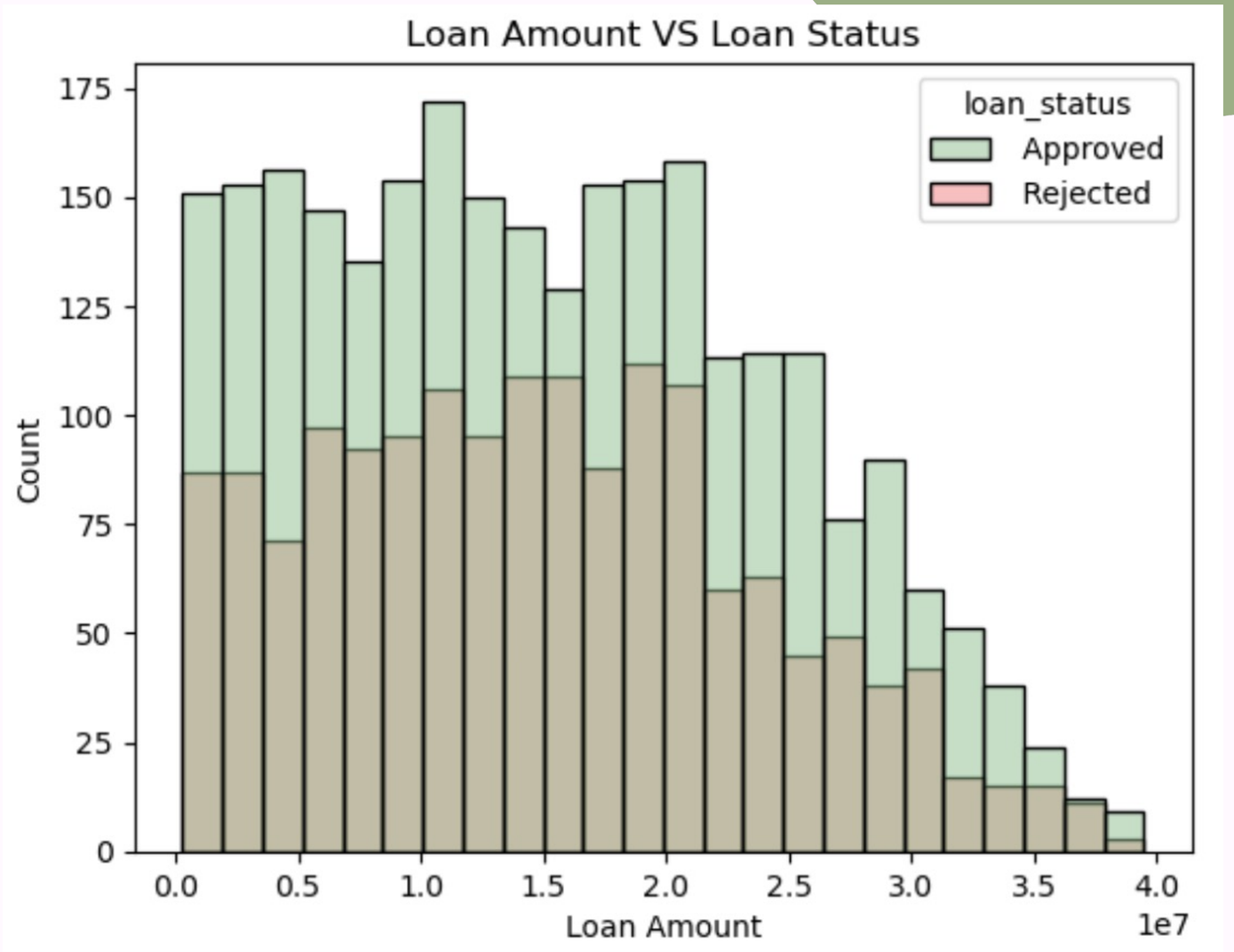
- Mean: 1.5
- 25%: 0.77
- 75%: 2.15



Relationship between Loan Amount and Loan Status

As loan amount increases, rate of loan acceptance and rejection decreases at around the same rate.

This shows that the relationship between loan status and loan amount is weak.



Relationship between Loan Status, Loan Amount and Annual Income

From the scatter plot:

1. when annual income increases, loan amount tends to increase
2. those with lower income tend to have a narrower range of loan amounts

Insight: Those with annual income on the higher end are still rejected when they apply for a lower loan amount

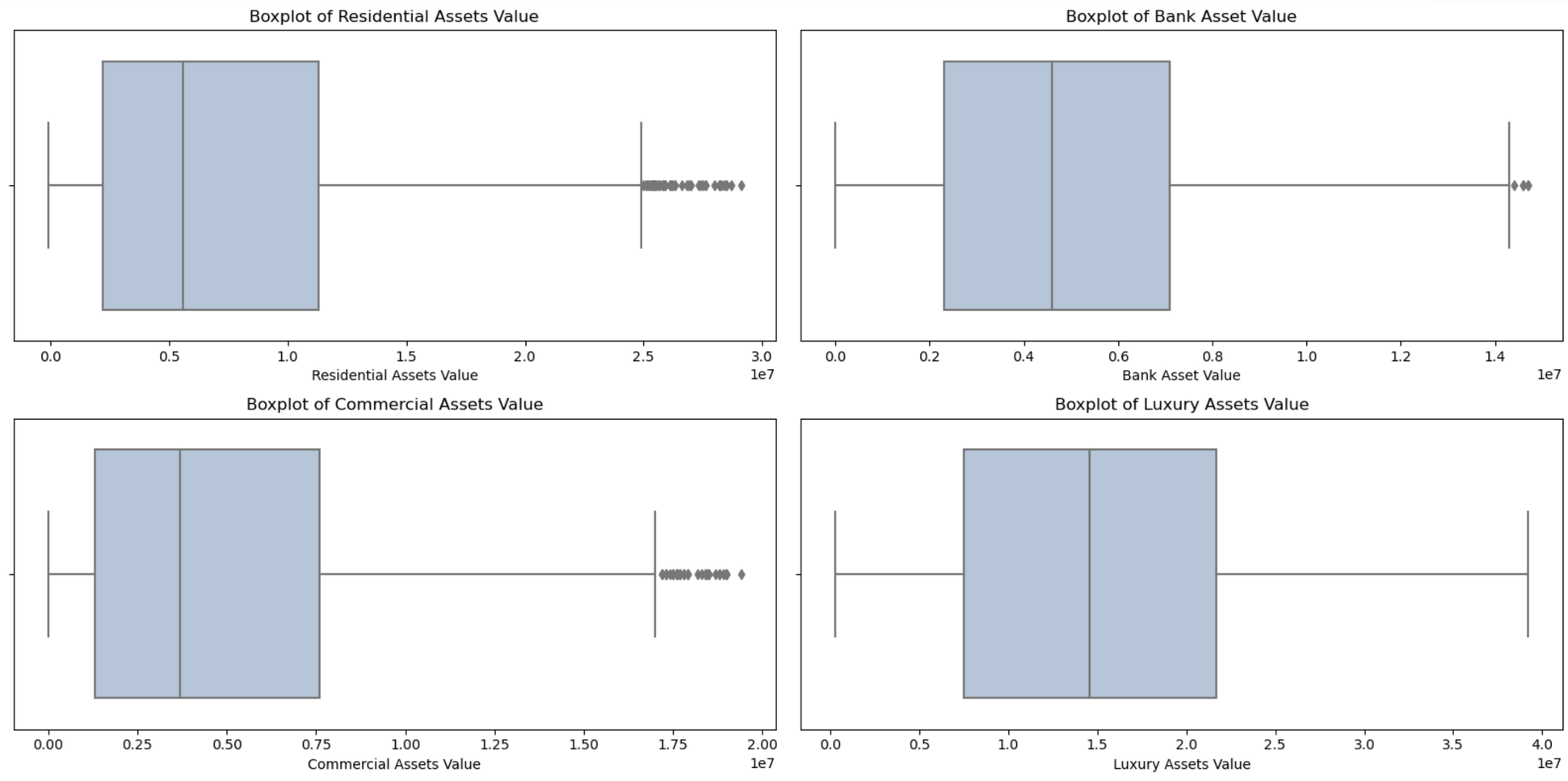
2. Credit Score - CIBIL Score

There is a strong relationship between credit score and loan status.

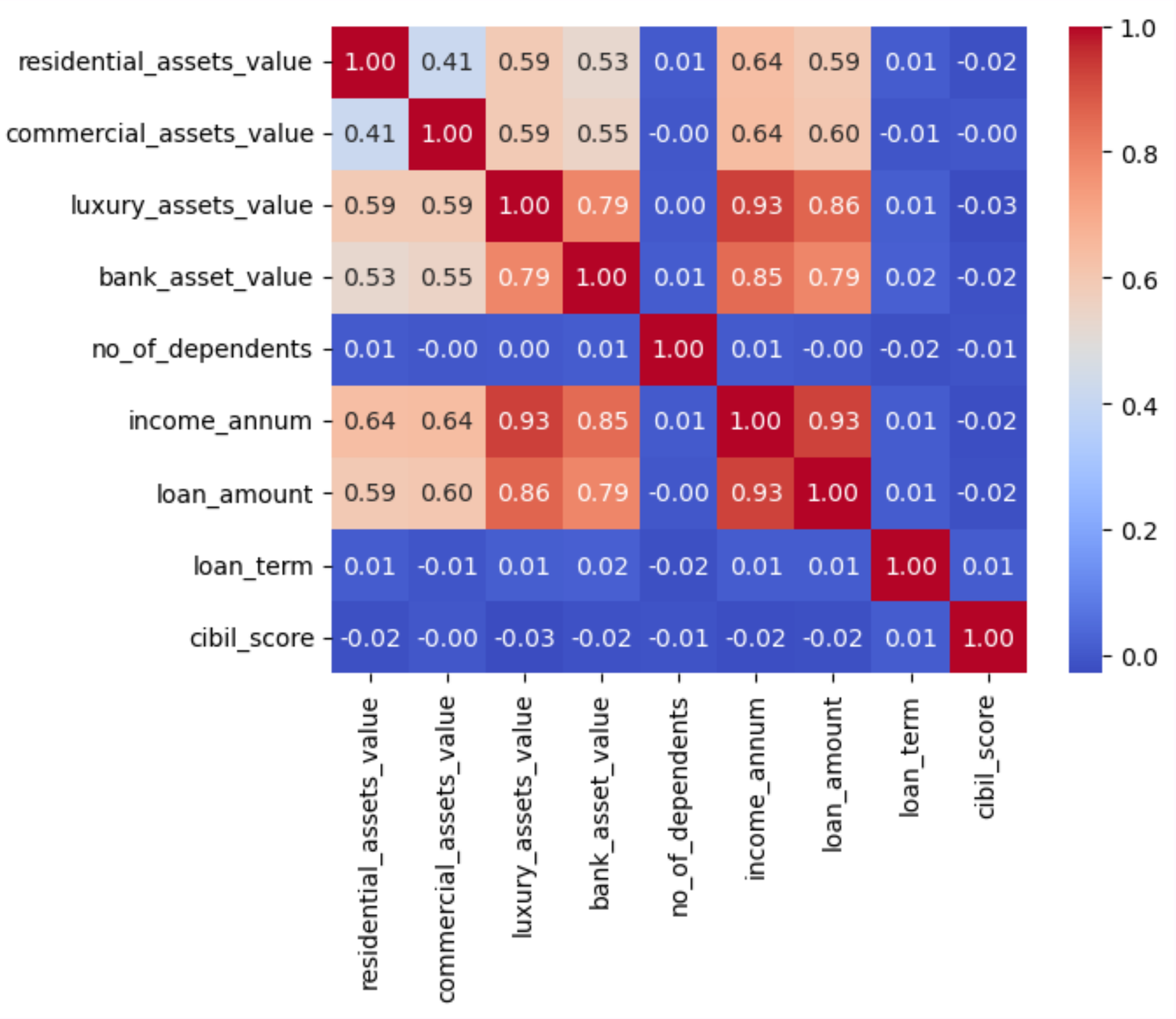
Separating score: 530-550

However, some with high credit scores were still rejected.

3. Residential Assets Value, Commercial Assets Value, Luxury Assets Value, Bank Asset Value



Outliers for residential, commercial and bank asset value are present.

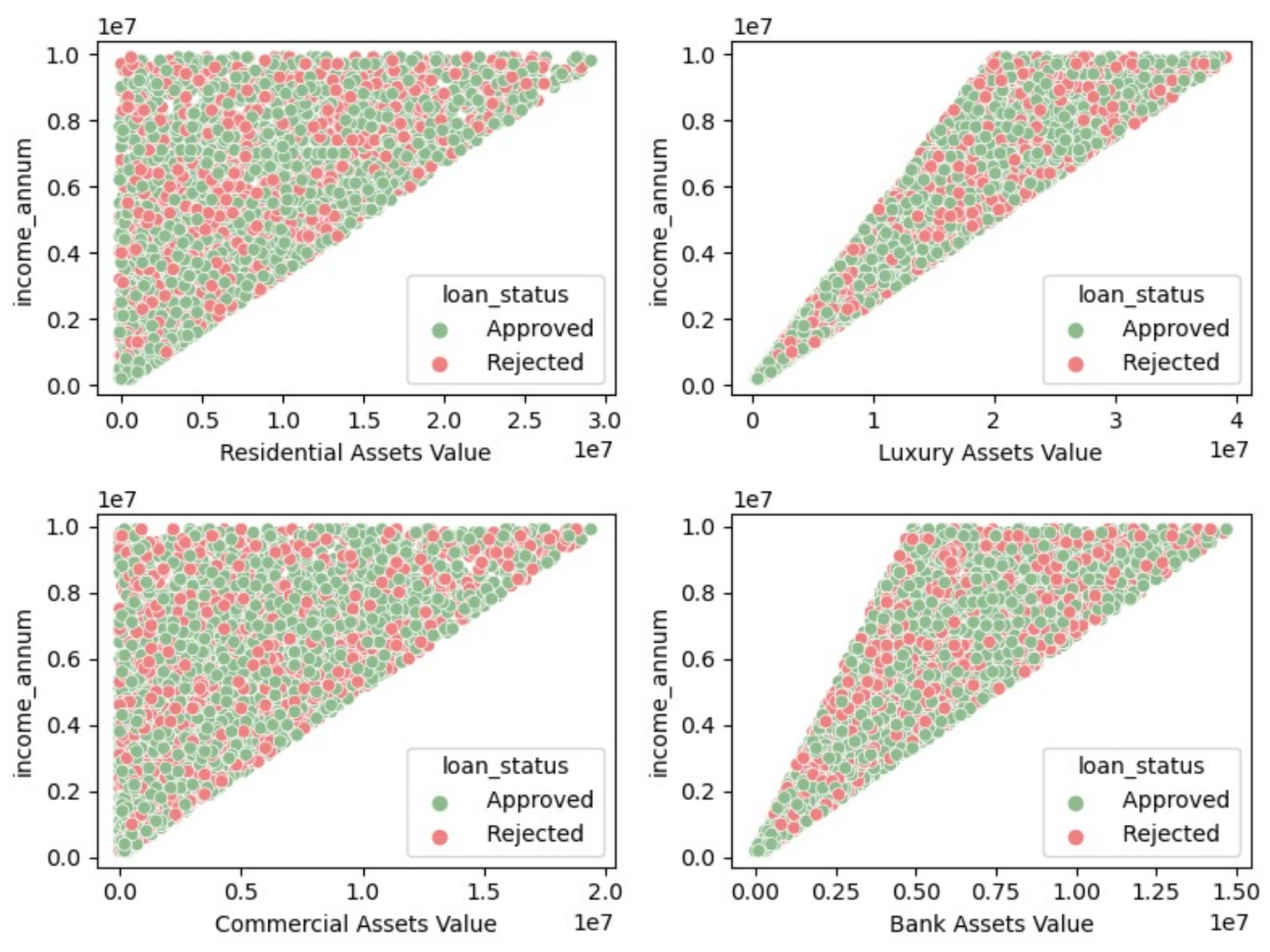


- all asset values have moderate to strong positive linear relationships with annual income
- CIBIL Score and Annual Income have a weak relationship

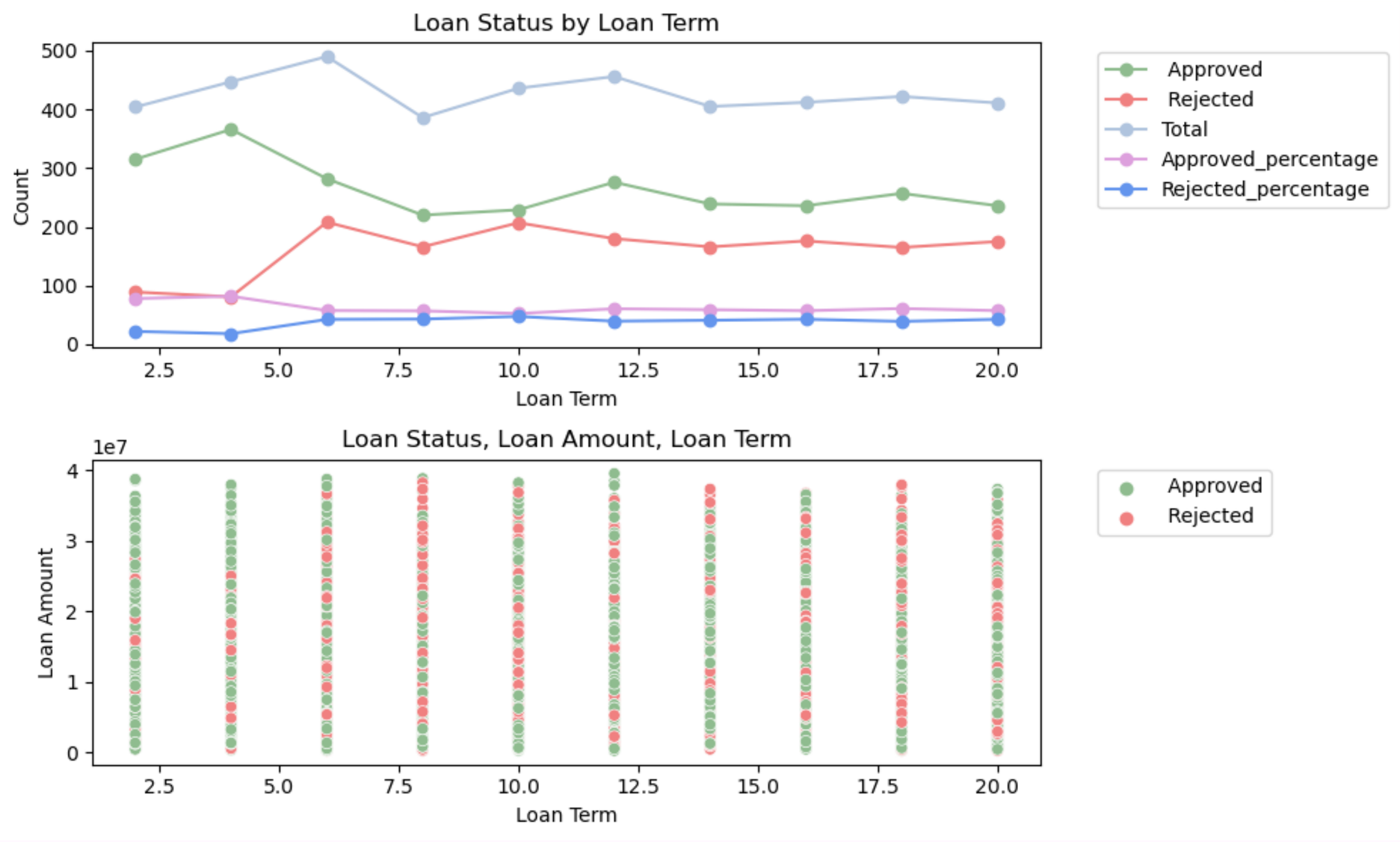
Threshold Effect:

- when Residential & Commercial asset value surpasses a certain threshold, there is a notable increase in annual income

This is not observed for Luxury and Bank assets where higher variability in annual income is observed.



4. Loan Term


Loan Status by Loan Term
Count
Loan Term
Approved
Rejected
Total
Approved_percentage
Rejected_percentage
Loan Status, Loan Amount, Loan Term
1e7
Loan Amount
Loan Term
Approved
Rejected

Reason for Rejection

After analysing our data, possible rejection reasons could be:

- low credit scores
- annual income

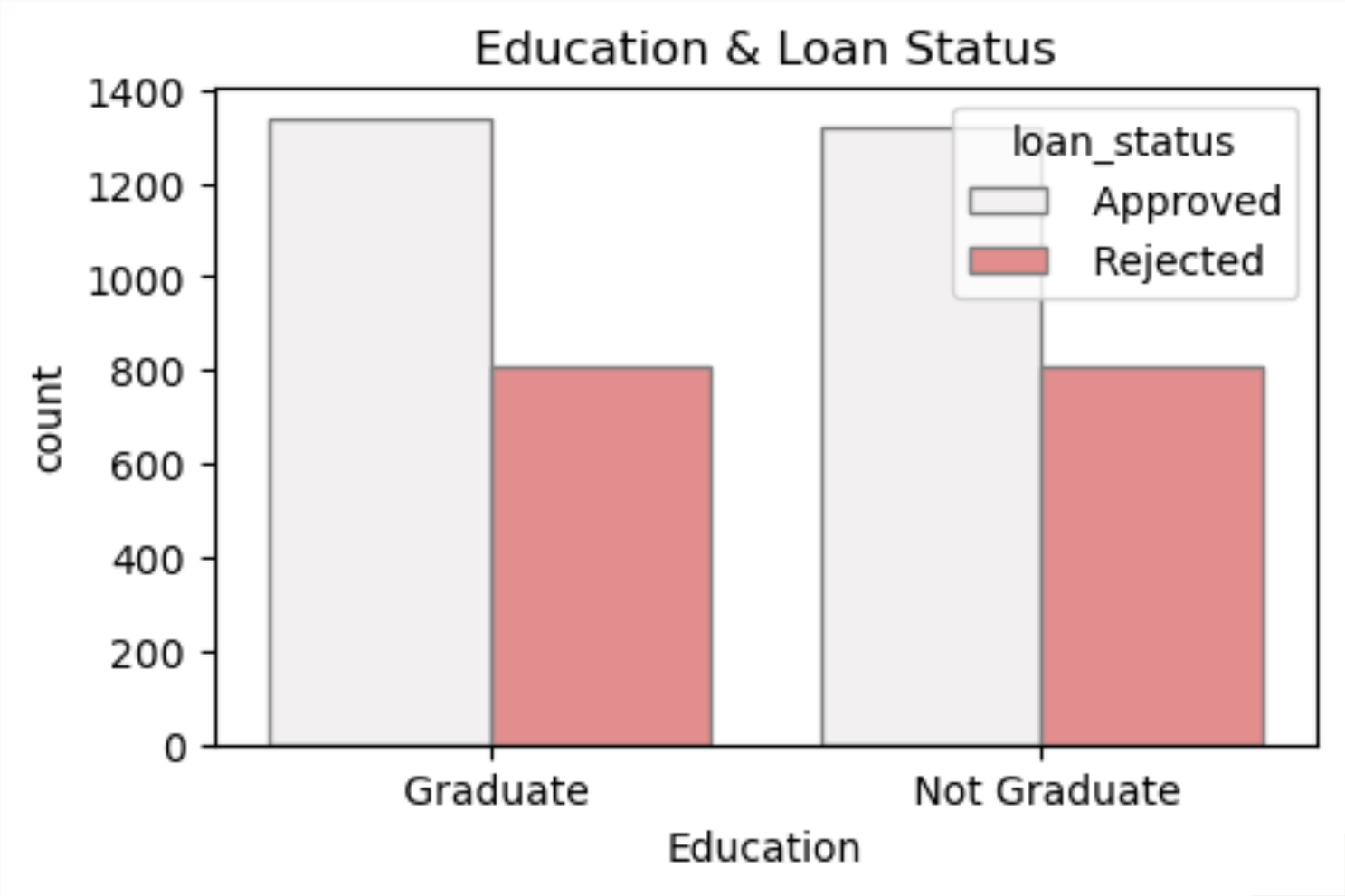
Data Visualisation: Categorical

1. EDUCATION
2. SELF-EMPLOYMENT
3. NUMBER OF DEPENDENTS

1. Education

Counts on Loan Status based on Education are approximately the same.

It appears that the education of an applicant has no direct effect on Loan Status.

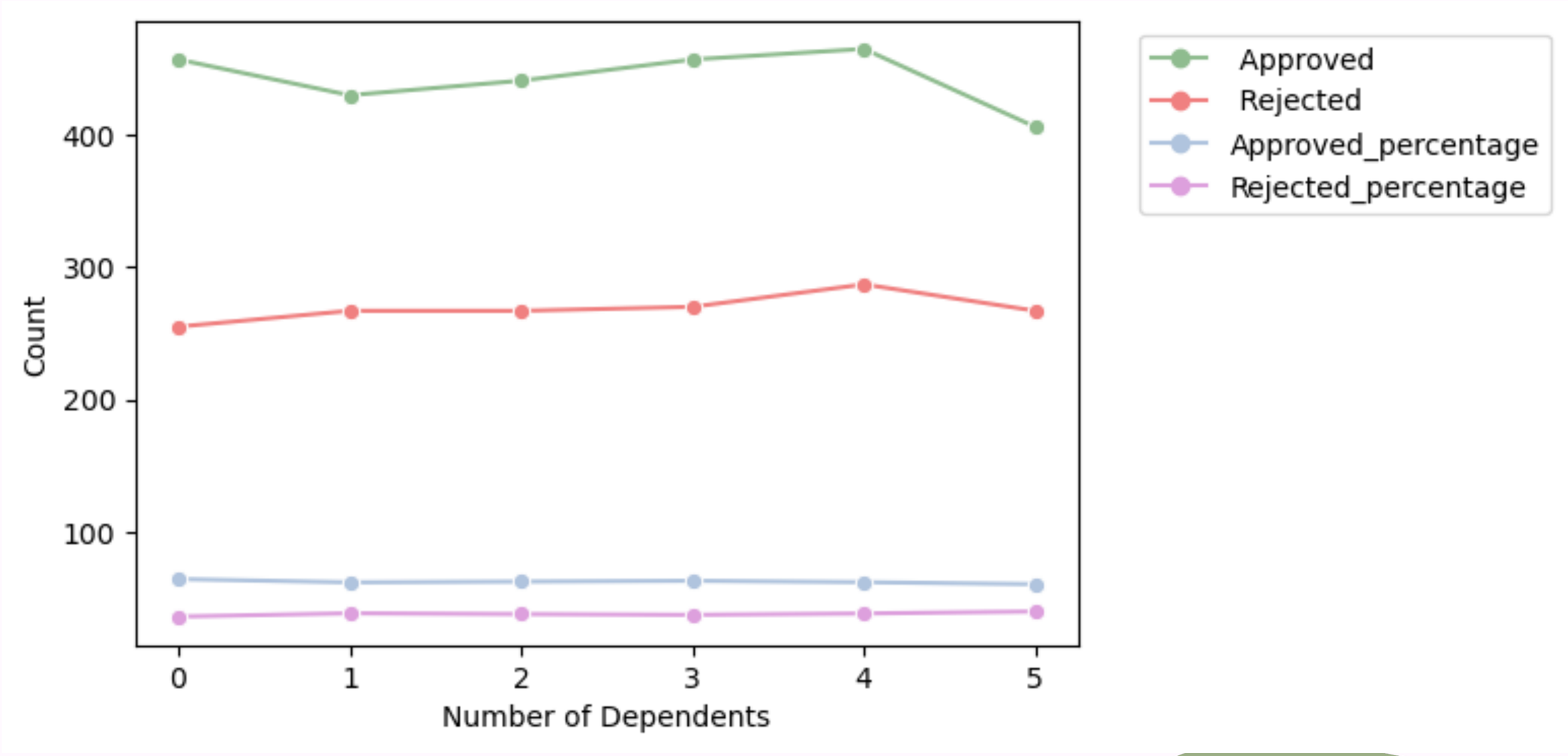


2. Self Employed

Counts on Loan Status based on Self Employment are approximately the same.

Self Employment appears to not have a direct effect on loan Status

3. Number of Dependents



Statistical Tests: Chi Square Test

A statistical method used to determine whether there is a significant association (or dependency) between categorical data in a contingency table.

Our Results

p-value > 0.05

- no significant evidence to reject the null hypothesis

```
no_of_dependents:
Chi-Square Value: 2.4542
p-value: 0.7834
Degrees of Freedom: 5
Expected Frequencies Table:
[[442.97774654 269.02225346]
 [433.6453502  263.3546498 ]
 [440.48910752 267.51089248]
 [452.31014289 274.68985711]
 [467.8641368  284.1358632 ]
 [418.71351605 254.28648395]]
```

```
education:
Chi-Square Value: 0.0840
p-value: 0.7720
Degrees of Freedom: 1
Expected Frequencies Table:
[[1333.91051769  810.08948231]
 [1322.08948231  802.91051769]]
```

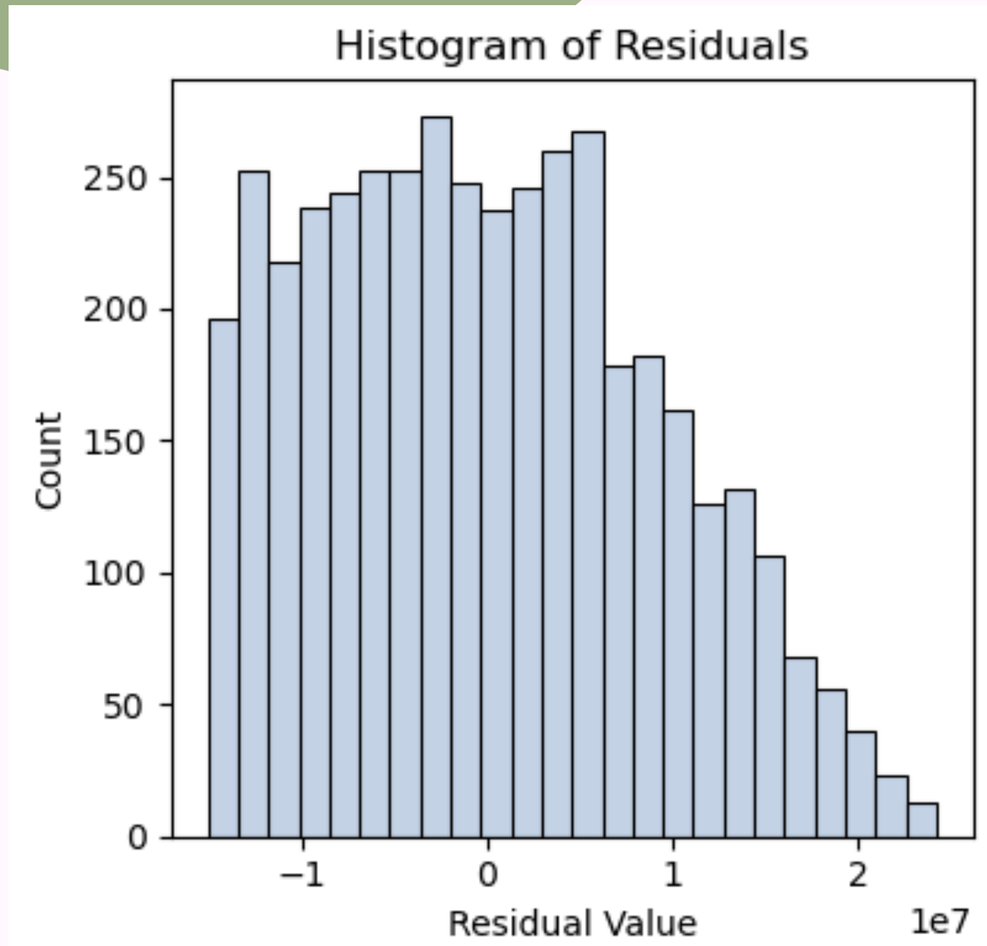
```
self_employed:
Chi-Square Value: 0.0000
p-value: 1.0000
Degrees of Freedom: 1
Expected Frequencies Table:
[[1318.35652378  800.64347622]
 [1337.64347622  812.35652378]]
```

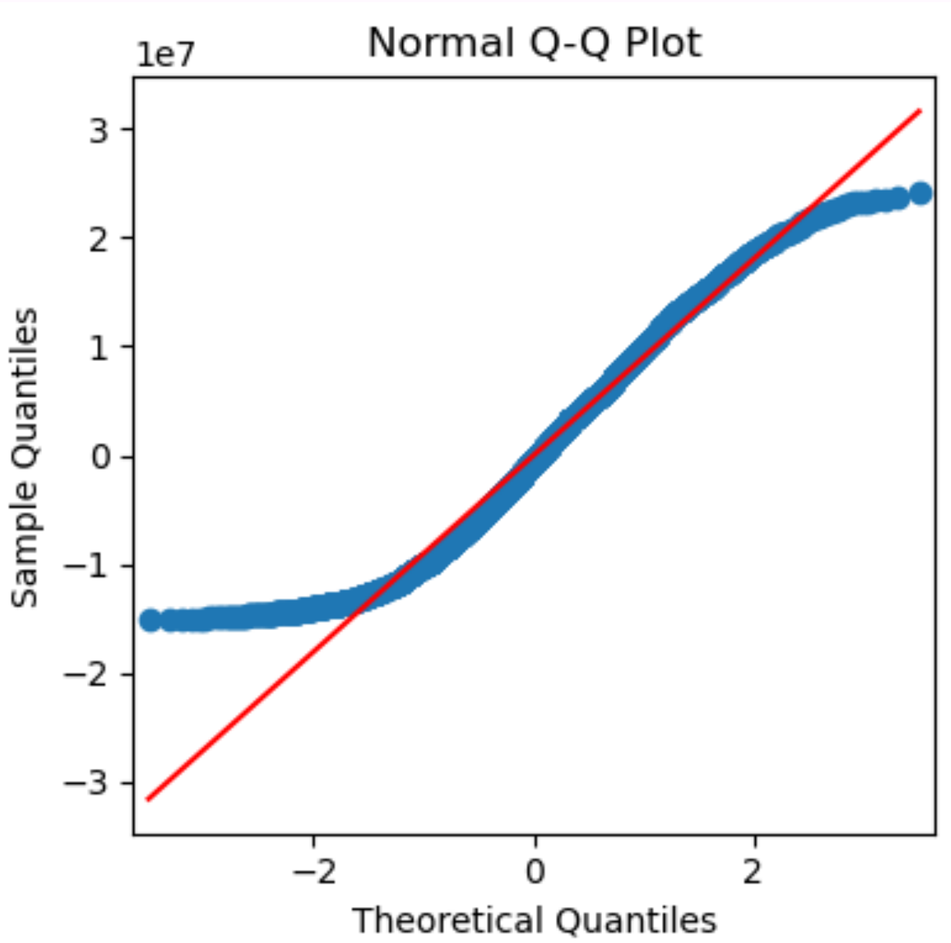
Data thus does not provide sufficient that there is a meaningful relationship between Loan Status and the three categorical variables.

Statistical Tests: ANOVA

Analysis of Variance (ANOVA) is a statistical method used to compare the Mean among three or more groups to determine whether there are statistically significant differences between these groups.

Our Results



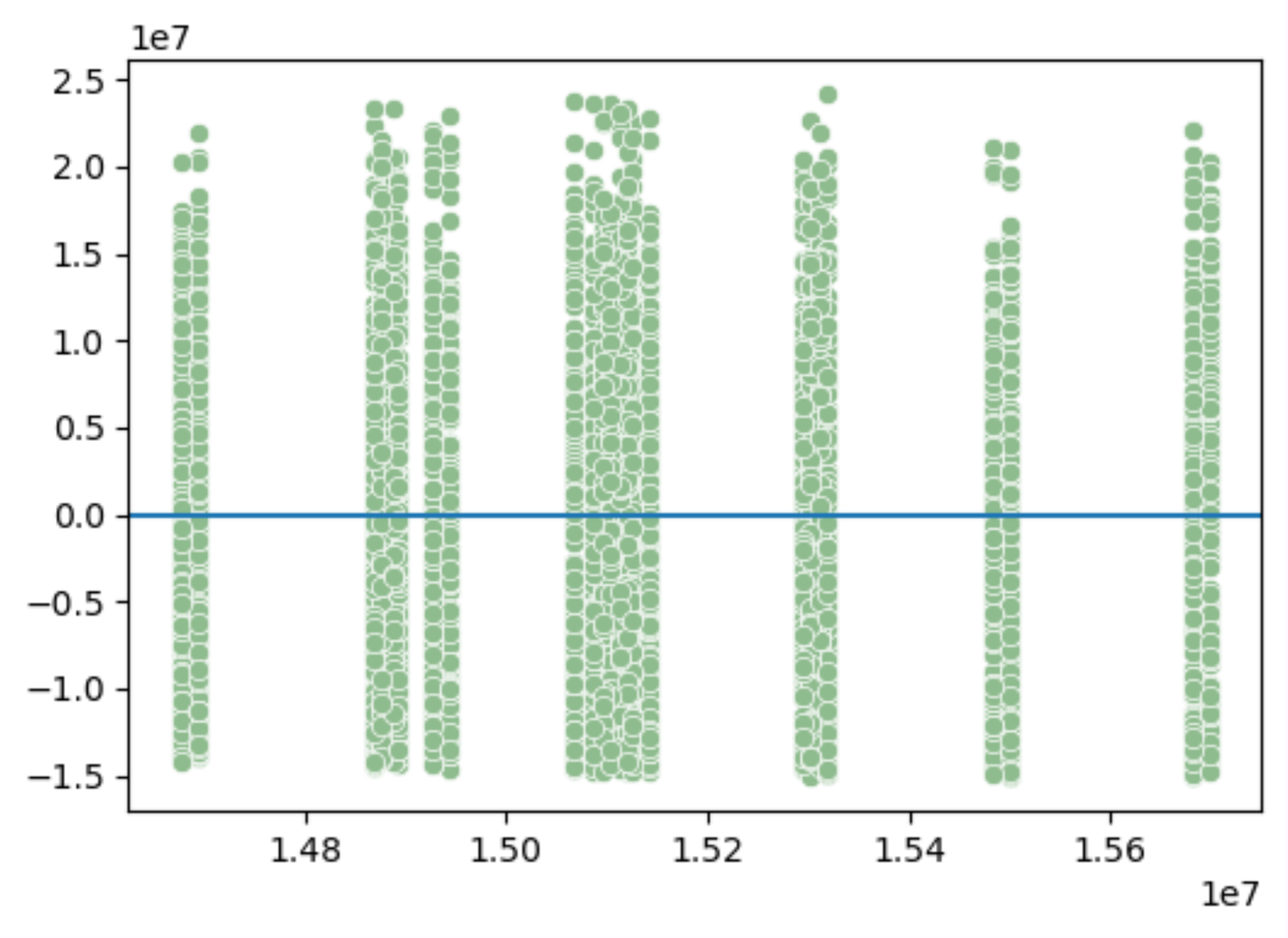


Histogram: right skewed distribution

Q-Q Plot: Blue markers closely align with red diagonal line

Homoscedascity

The assumption that the variability (or spread) of the residuals is consistent across all levels of the independent variables.



ANOVA Summary

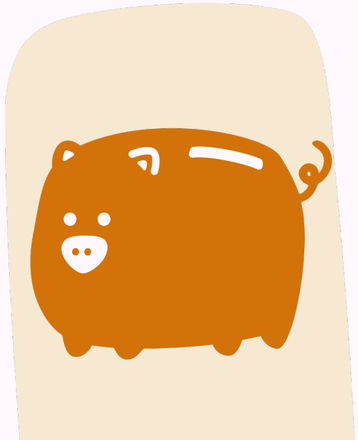
ANOVA Summary:

	sum_sq	df	F	PR(>F)
C(no_of_dependents)	2.747923e+14	5.0	0.671513	0.645062
C(education)	4.195183e+13	1.0	0.512591	0.474059
C(self_employed)	3.389126e+11	1.0	0.004141	0.948694
Residual	3.487321e+17	4261.0	NaN	NaN

Machine Learning

1. dropped variables that have no relationship

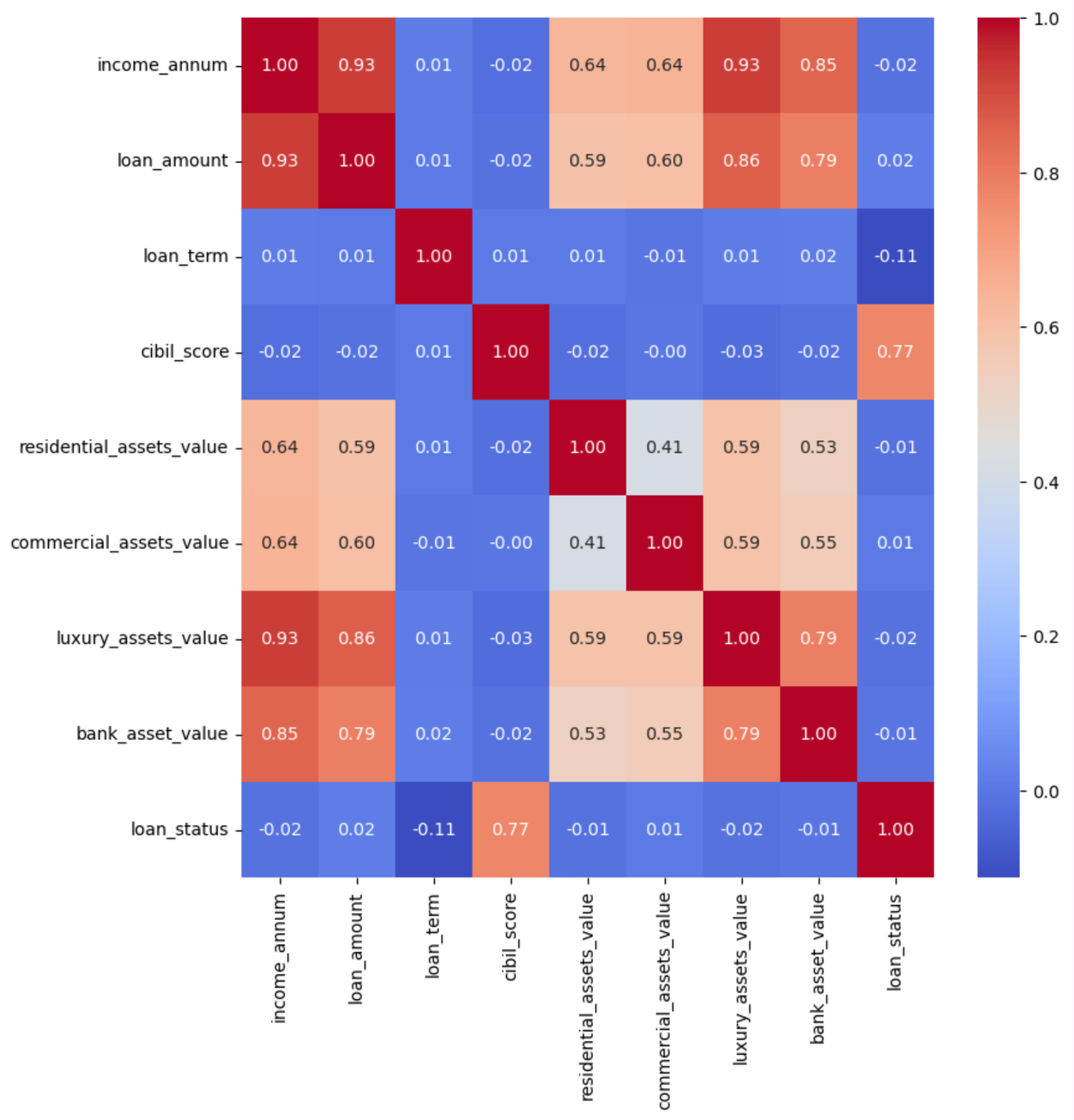
2. convert loan status categorical variable to dummy variable



3. removing loan_status_rejected column

4. standardising our data

5. machine learning models



Reviewing heatmap

- cibil_score has the highest positive correlation with loan_status
- strong linear relationships between income_annum and loan_amount, luxury_assets_value and bank_asset_value
- strong liner relationship between luxury_assets_value and bank_asset_value

Standardisation

```
#standardise dataset
stand_scaler = StandardScaler()
stand_scaler.fit(X_train)
X_train_stand = stand_scaler.transform(X_train)
X_val_stand = stand_scaler.transform(X_val)
X_test_stand = stand_scaler.transform(X_test)
```

Preprocessing technique commonly used in machine learning to transform numerical features so that they have a mean of 0 and a standard deviation of 1.

1. Logistic Regression

It is a type of machine learning algorithm used for binary classification where the goal is to predict the probability of a particular outcome.

It predicts probability that an instance belongs to a particular class.

EQUATION FOR LOGISTIC REGRESSION

$$f(x) = \frac{1}{1 + e^{-x}}$$

Results

Goodness of Fit of Model

Accuracy: 0.917
Precision: 0.934
Recall: 0.925
F1 Score: 0.930

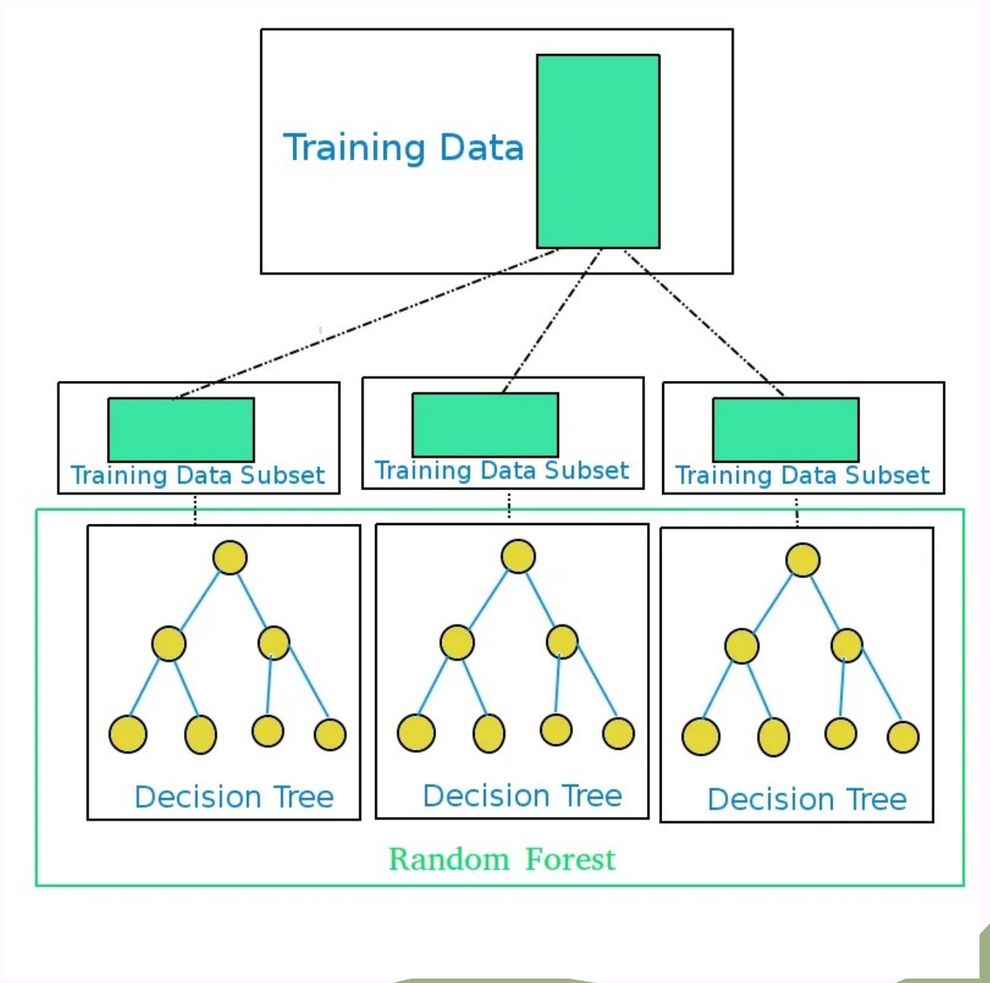
Coefficient Summary

Variables	Coefficient
income_annum	-1.66741
loan_amount	1.1483
loan_term	-0.865514
cibil_score	4.12359
residential_assets_value	0.0612439
commercial_assets_value	0.035288
luxury_assets_value	0.340216
bank_asset_value	0.184455
Intercept	1.82502

2. Random Forest

It is a supervised machine learning algorithm that is used for both classification and regression tasks.

It combines multiple decision trees to produce a more robust model,



Hyperparameter Tuning

```
#hyperparameter tuning
param_dist = {
    'n_estimators': [50, 100, 150],
#This hyperparameter (n_estimators) determines the number of trees
#The random search will explore these three values to determine the

    'max_depth': [None, 10, 20],
#This hyperparameter controls the maximum depth of each tree in the
#None represents no maximum depth (i.e., trees are expanded until a

    'min_samples_split': [2, 5, 10],
#This hyperparameter sets the minimum number of samples required to

    'min_samples_leaf': [1, 2, 4]
 #This hyperparameter specifies the minimum number of samples requi
}
```

Randomised Search Cross Validation

```
{'n_estimators': 150,
 'min_samples_split': 10,
 'min_samples_leaf': 4,
 'max_depth': 10},
```

This is done to retrieve the best parameters and estimators.

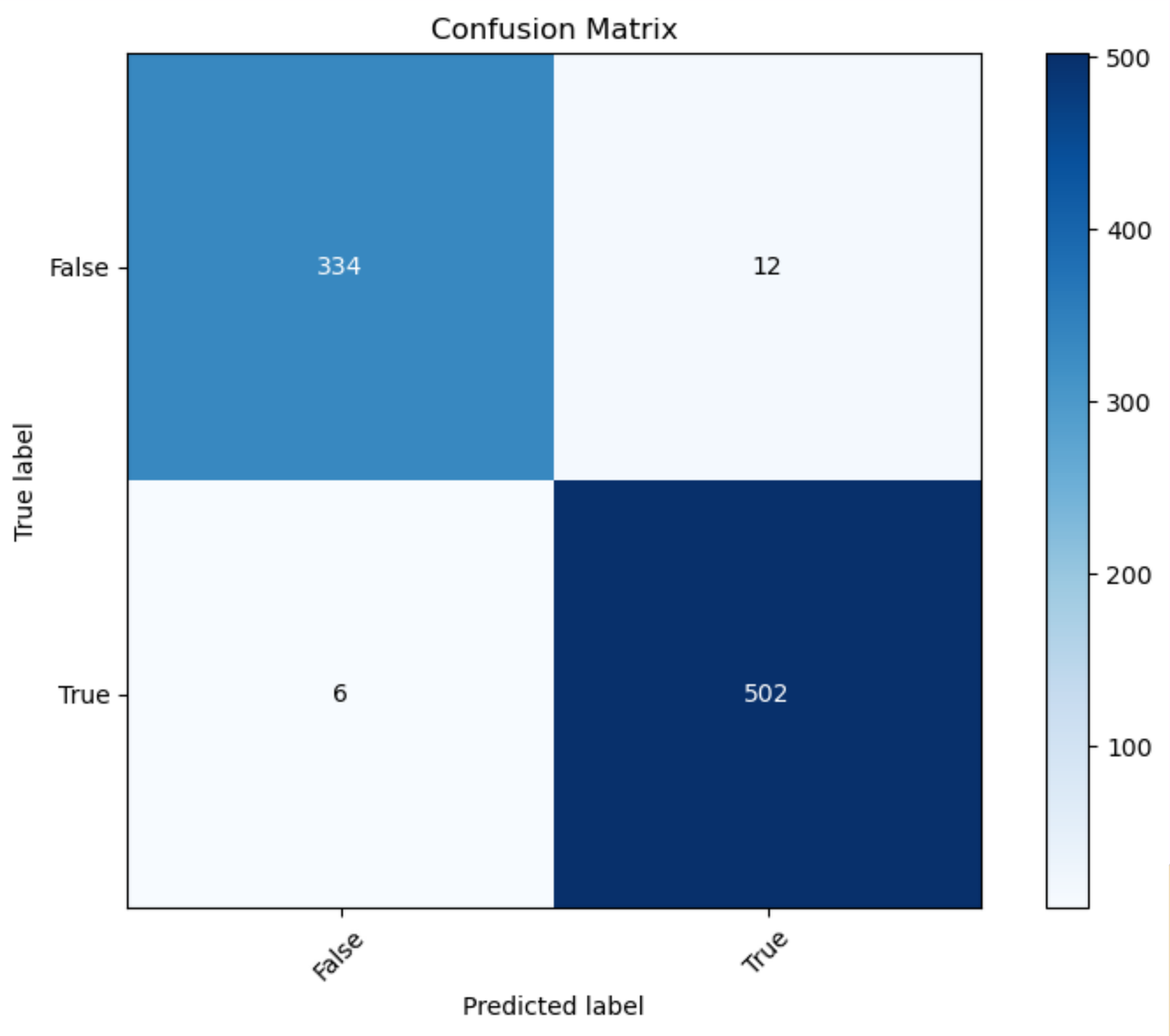
Results

Goodness of Fit of Model

Train Set:
Accuracy: 0.979
Precision: 0.977
Recall: 0.988
F1 Score: 0.982

Test Set:
Accuracy: 0.973
Precision: 0.976
Recall: 0.981
F1 Score: 0.979

Confusion Matrix

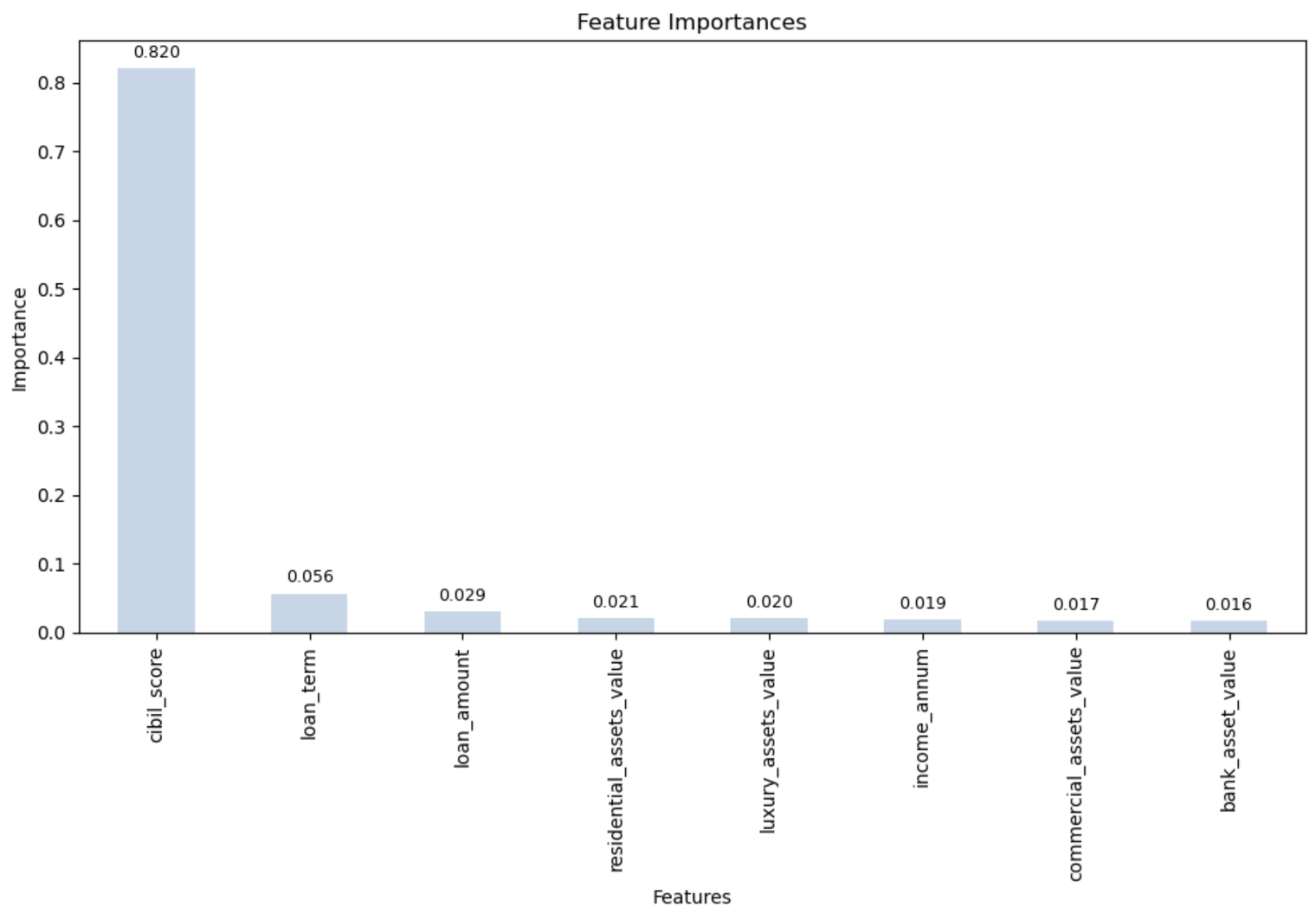


Overall

The Random Forest Model (RFM) performed better than the Logistic Regression Model.

Scores obtained on the RFM test set are extremely similar to the train set scores. This is a positive indication that the model's performance generalises well to new and unseen data.

Insights: Importance of variables



In our analysis, cibil_score emerges as the most important factor affecting the prediction of loan_status .

However, it is important to understand that other factors including data noise can also contribute to these findings.

Loan Status Prediction Program

Features of this Program:

- allows users to input their data for loan status prediction
- informs the user of the probability of approval
- provides users with the factors that contributed to their rejection in order of importance

Snippet of our code

```python
# Assume rf_opt is your trained Random Forest classifier
# Example:
# rf_opt = RandomForestClassifier(n_estimators=150, max_depth=None, min_samples_leaf=1, min_samples_split=5, random_
# Make sure to load your trained model here

# Define a function to preprocess user input
def preprocess_input(input_data):
    # Create a DataFrame from user input
    user_data = pd.DataFrame([input_data])
    return user_data

# Define a function to predict loan status and explain prediction
def predict_loan_status(user_data, model):
    # Use the model to predict loan status and probabilities
    prediction = model.predict(user_data)
    probabilities = model.predict_proba(user_data)

    # Determine loan status and probability of approval
    loan_status = "Approved" if prediction[0] == 1 else "Rejected"
    probability_approved = probabilities[0][1]  # Probability of approval (class 1)

    # Get feature importances
    feature_importances = model.feature_importances_
    features = user_data.columns.tolist()
    important_features = dict(zip(features, feature_importances))
    important_features_sorted = dict(sorted(important_features.items(), key=lambda x: x[1], reverse=True))

    return loan_status, probability_approved, important_features_sorted
```

Example output

```
Welcome to Loan Status Prediction Program!

Enter your annual income:
```

```
Welcome to Loan Status Prediction Program!

Enter your annual income: 2300000
Enter the loan amount: 98800000
Enter your loan term (in years): 18
Enter your credit score: 336
Enter your residential assets value: 1800000
Enter your commercial assets value: 800000
Enter your luxury assets value: 5200000
Enter your bank asset value: 1500000

Prediction Results:
Loan Status: Rejected
Probability of Approval: 0.08

Top Contributing Factors:
cibil_score: 0.8199
loan_term: 0.0562
loan_amount: 0.0294
residential_assets_value: 0.0212
luxury_assets_value: 0.0203
income_annum: 0.0192
commercial_assets_value: 0.0173
bank_asset_value: 0.0164

Thank you for using the Loan Status Prediction Program!
```

Thank You!

References:

1. Koehrsen, W. (2018, January 10). Hyperparameter tuning the random forest in python. Medium. https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74
2. Kai. (2023, July 16). Loan-approval-prediction-dataset. Kaggle. https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset/data