PROJECT: LOAN APPROVAL ANALYSIS

Introduction to Artifical Intelligence

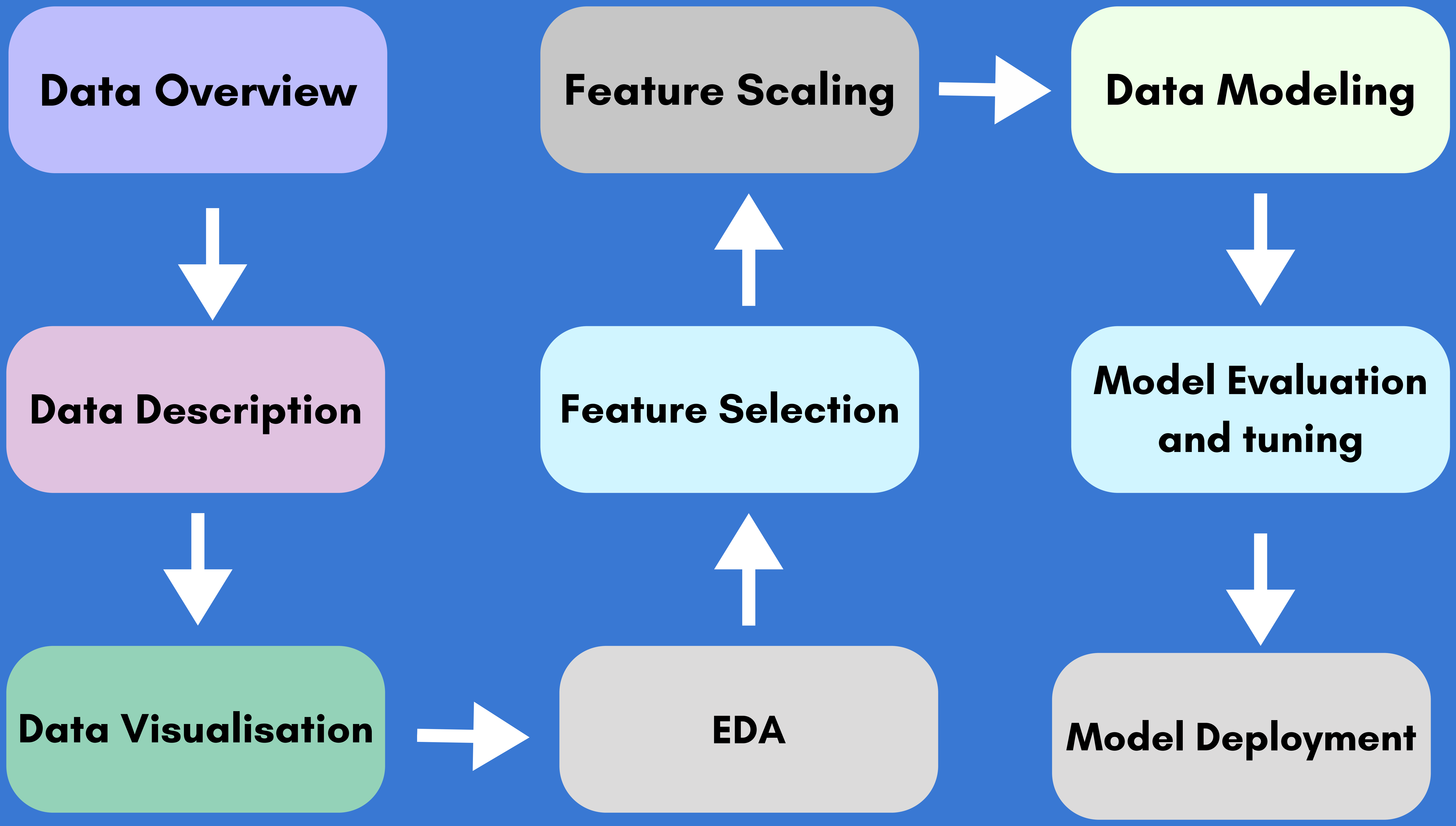
Group Members:

1. Ayush Sati
2. Chaitanya Chaniyara
3. Neha Roy Choudhury
4. Prasad Hadbe
5. Vamsi Krishna Pirati

Problem Statement:

To build a robust machine learning model that accurately predicts loan approval using a comprehensive dataset of financial records. By analyzing factors such as CIBIL score, income levels, employment status, loan terms, and asset values, the model aims to optimize decision-making for loan eligibility, enhancing efficiency and reliability in lending practices.

WorkFlow:


Data Overview
Data Description
Data Visualisation
EDA
Feature Selection
Feature Scaling
Data Modeling
Model Evaluation and tuning
Model Deployment

Dataset Overview:

loan_id	no_of_dependents	education	self_employed	income_annum	loan_amount	loan_term	cibil_score	residential_assets_value	commercial_assets_value
1	2	Graduate	No	9600000	29900000	12	778	2400000	17600000
2	0	Not Graduate	Yes	4100000	12200000	8	417	2700000	2200000
3	3	Graduate	No	9100000	29700000	20	506	7100000	4500000
4	3	Graduate	No	8200000	30700000	8	467	18200000	3300000
5	5	Not Graduate	Yes	9800000	24200000	20	382	12400000	8200000

- Dataset consists of 4269 rows and 13 columns
- Columns include: 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, and loan_status
- Target variable for analysis: loan_status

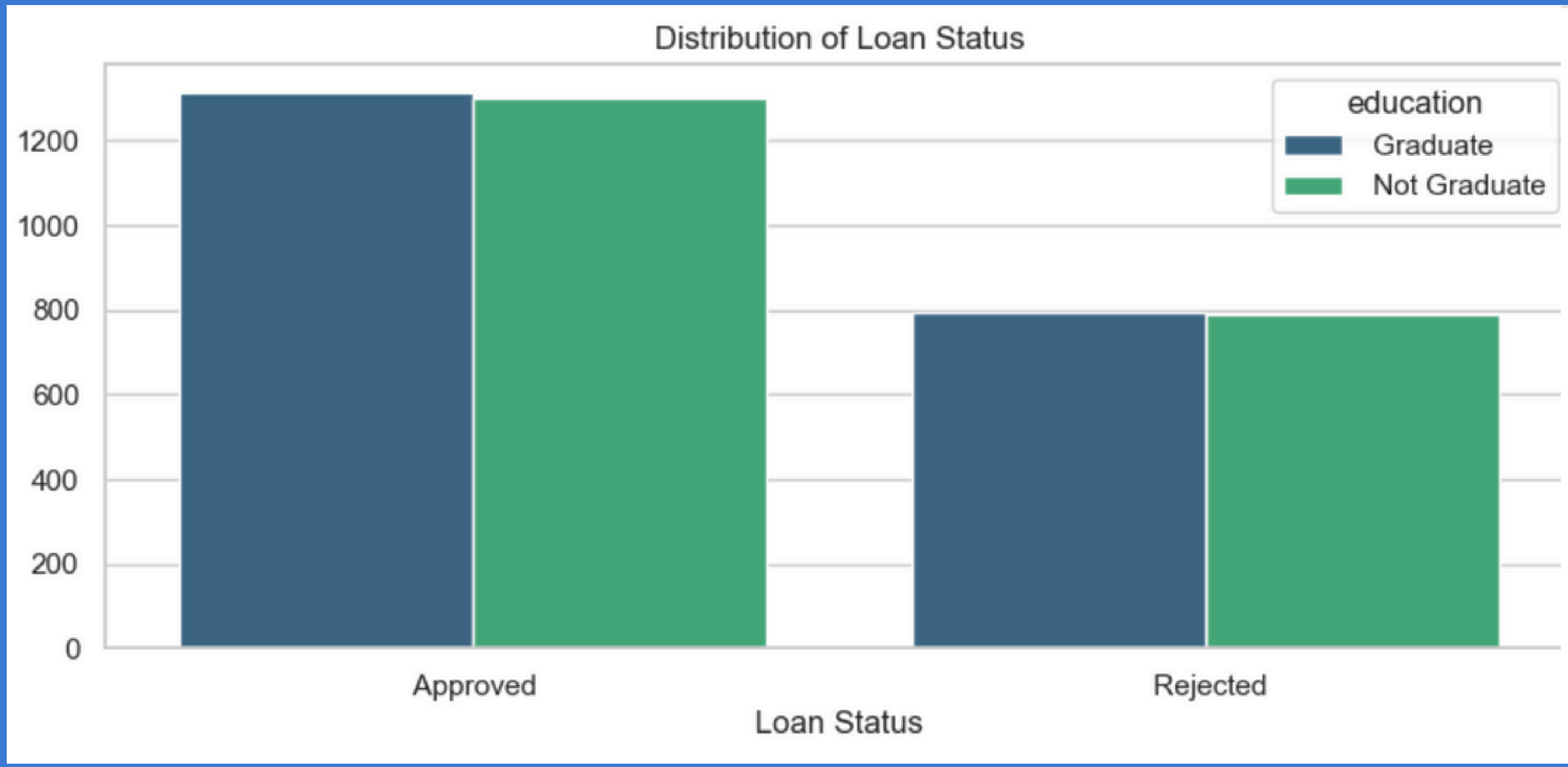
Exploratory Data Analysis

Descriptive Statistics:

	loan_id	no_of_dependents	income_annum	loan_amount	loan_term	cibil_score	residential_assets_value	commercial_assets_value	luxury_assets_value
count	4269.000000	4269.000000	4.269000e+03	4.269000e+03	4269.000000	4269.000000	4.269000e+03	4.269000e+03	4.269000e+03
mean	2135.000000	2.498712	5.059124e+06	1.513345e+07	10.900445	599.936051	7.472617e+06	4.973155e+06	1.512631e+07
std	1232.498479	1.695910	2.806840e+06	9.043363e+06	5.709187	172.430401	6.503637e+06	4.388966e+06	9.103754e+06
min	1.000000	0.000000	2.000000e+05	3.000000e+05	2.000000	300.000000	-1.000000e+05	0.000000e+00	3.000000e+05
25%	1068.000000	1.000000	2.700000e+06	7.700000e+06	6.000000	453.000000	2.200000e+06	1.300000e+06	7.500000e+06
50%	2135.000000	3.000000	5.100000e+06	1.450000e+07	10.000000	600.000000	5.600000e+06	3.700000e+06	1.460000e+07
75%	3202.000000	4.000000	7.500000e+06	2.150000e+07	16.000000	748.000000	1.130000e+07	7.600000e+06	2.170000e+07
max	4269.000000	5.000000	9.900000e+06	3.950000e+07	20.000000	900.000000	2.910000e+07	1.940000e+07	3.920000e+07

Data Visualisations:

Fig: Barplot showing Distribution of Loan depending on Education level



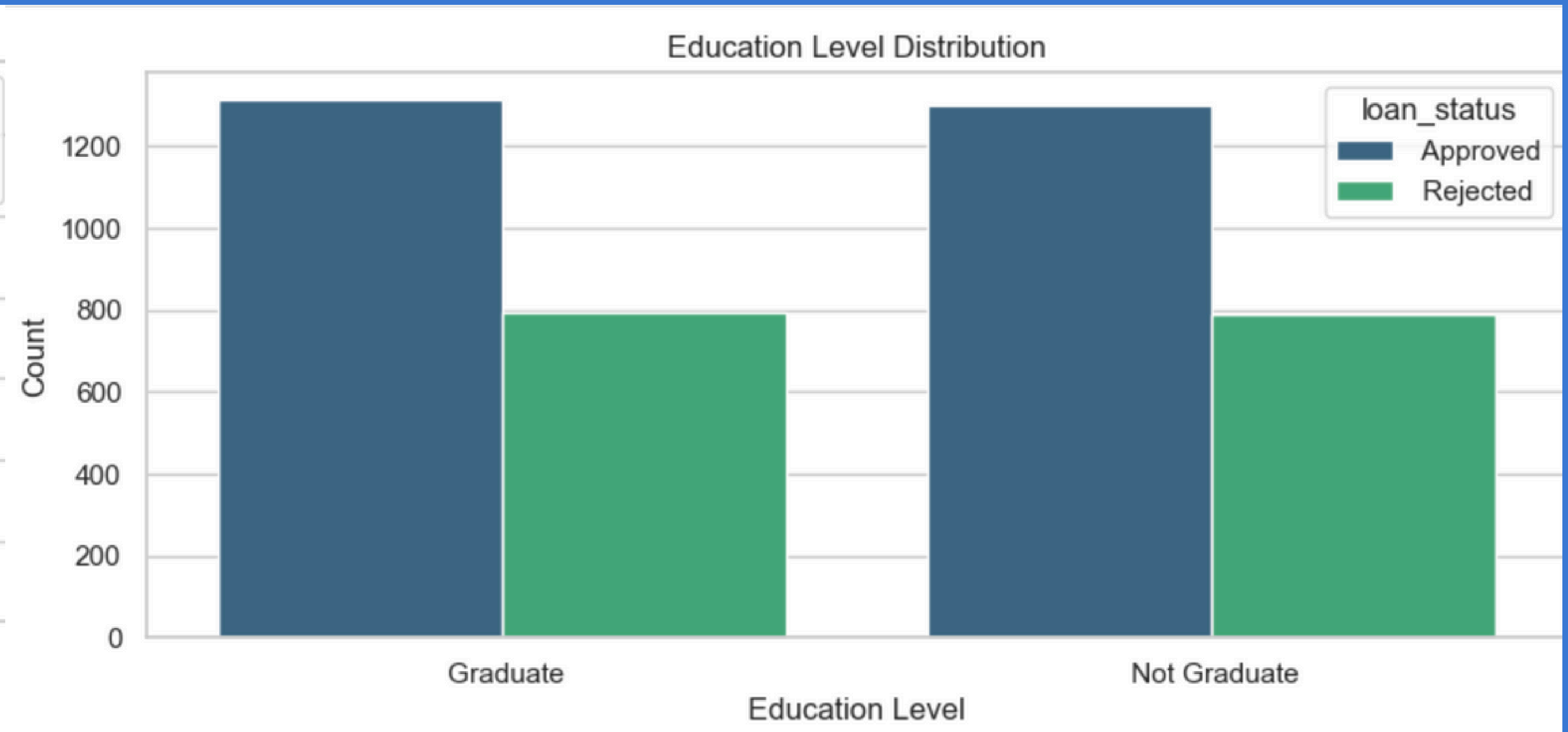
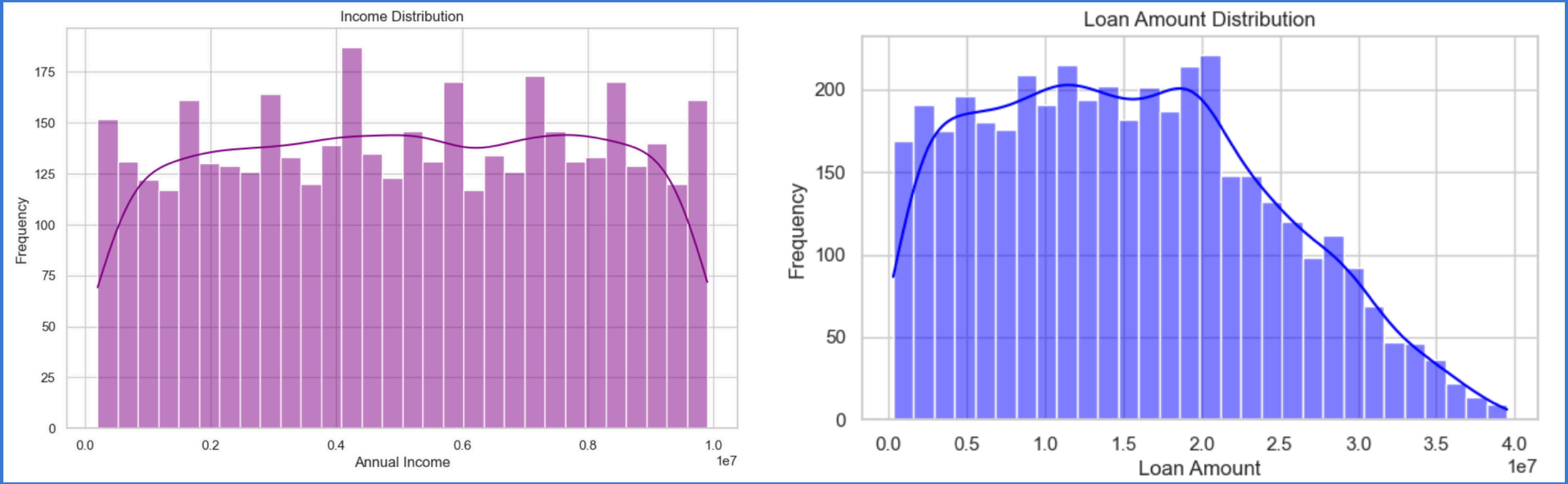


Fig: Histogram highlighting Income Distribution and Loan amount Distribution frequency



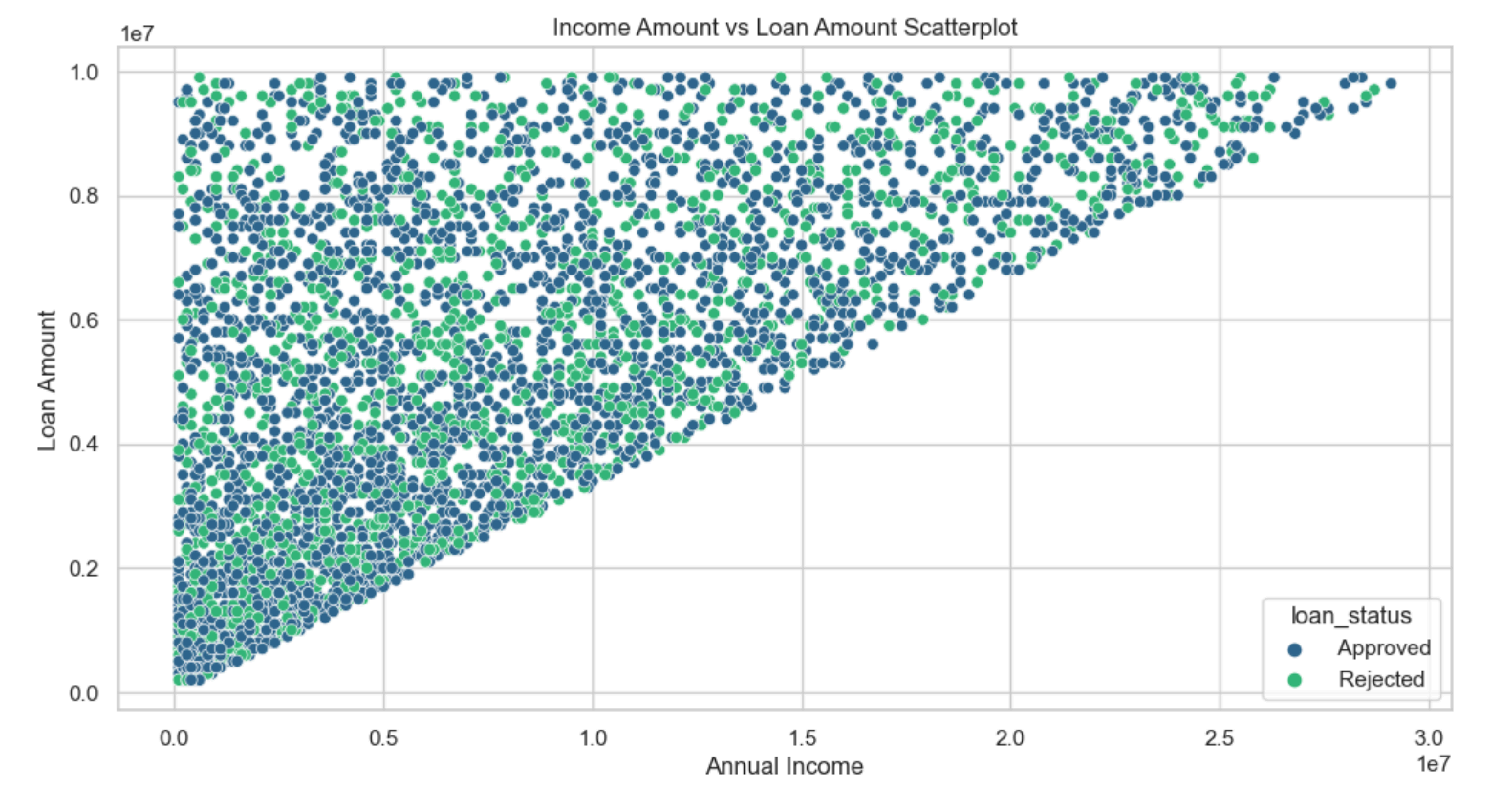
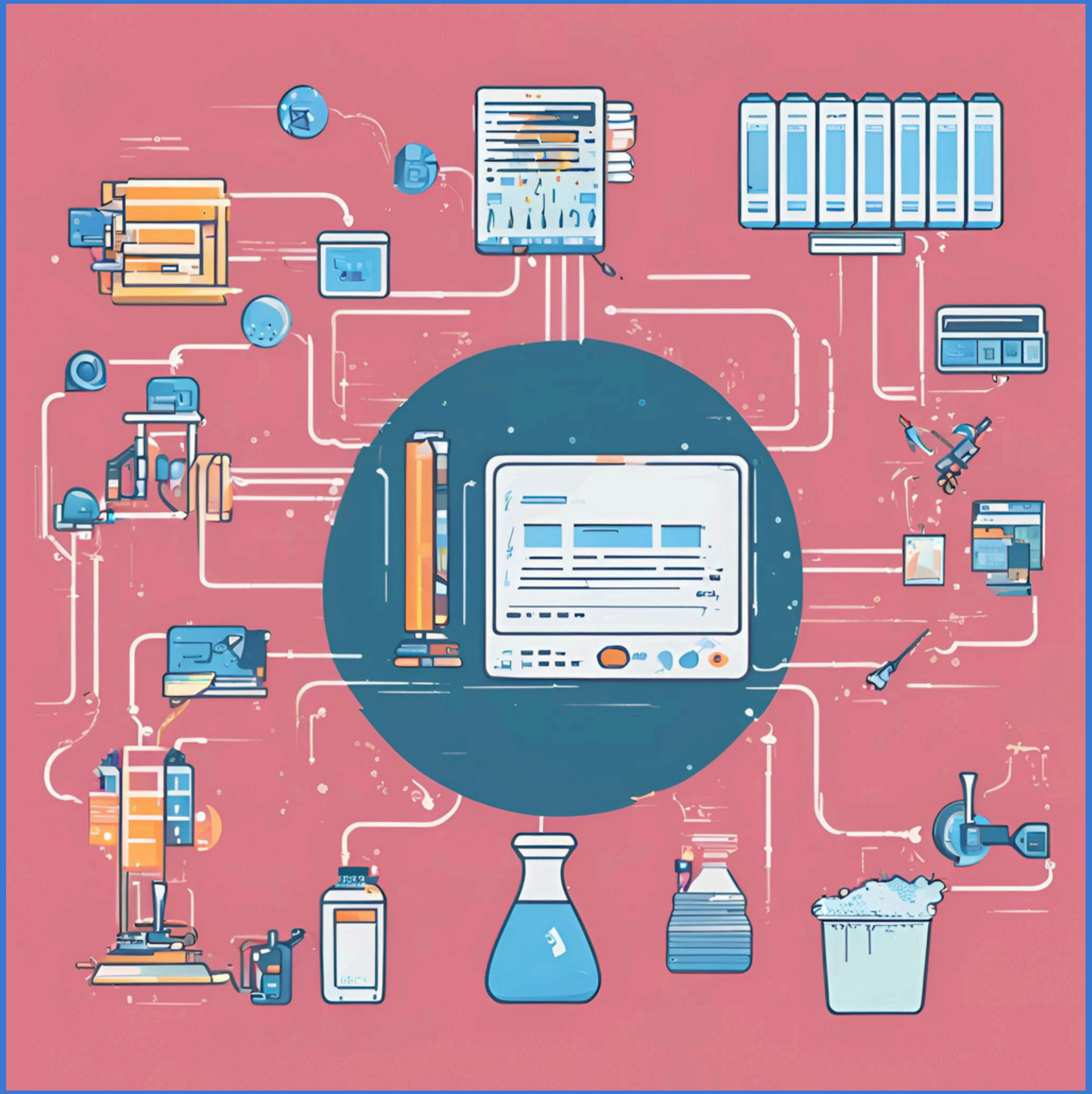


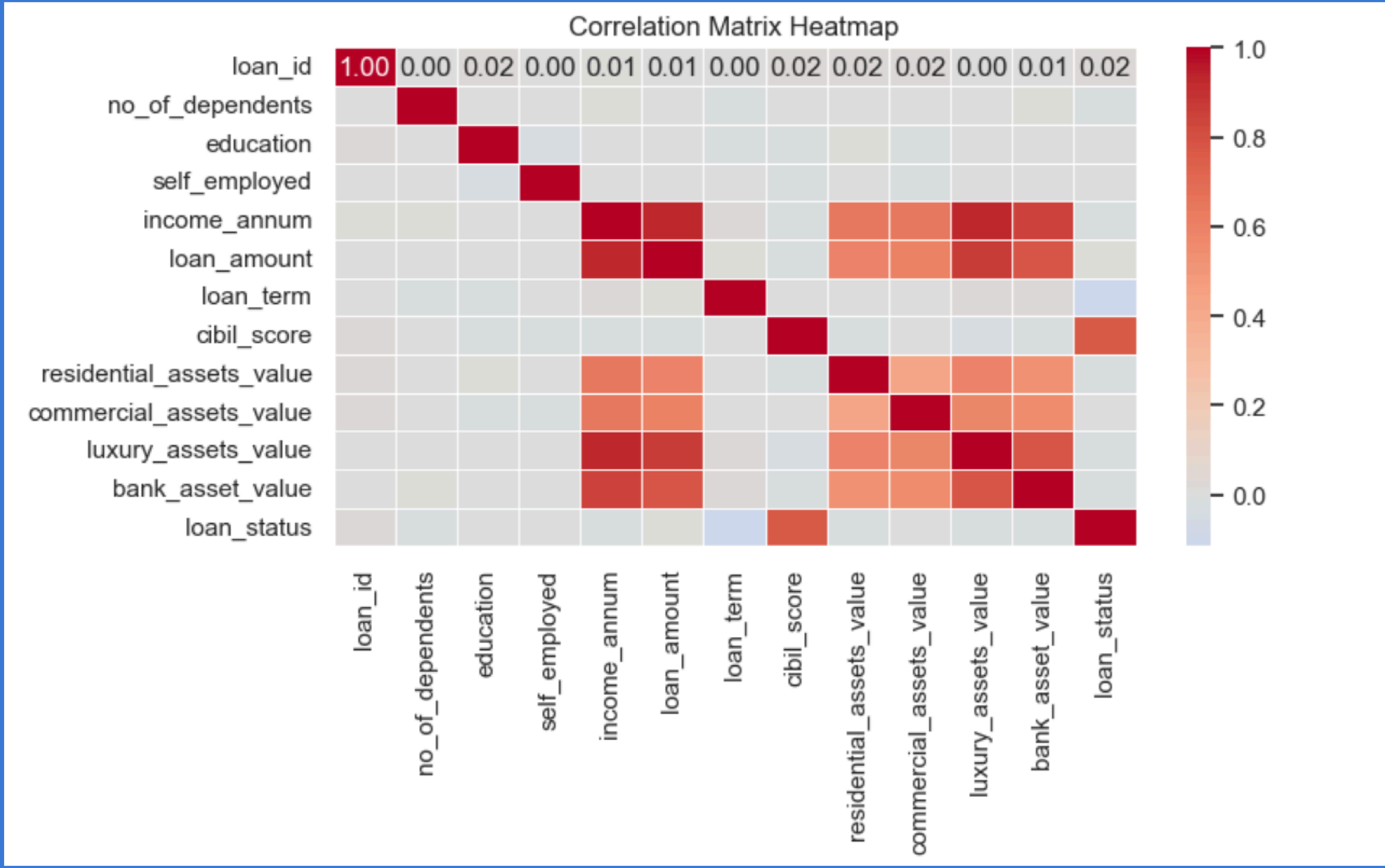
Fig: Scatterplot representing Loan amount distribution frequency with income



Data Cleaning and Preprocessing:

- Checking for Duplicates
- Missing Value Treatment
- Outlier Treatment
- Categorical Variables to Numeric Encoding

Feature Selection:



A correlation matrix was used to identify relationships between features.

```python
filtered_df, removed_features = filter_features_by_correlation(df, 0.8)
```

```python
filtered_df.head()
```

	loan_id	no_of_dependents	education	self_employed	income_annum	loan_term	cibil_score	residential_assets_value	commercial_assets_value	loan_status
0	1	2	1	0	9600000	12	778	2400000	17600000	1
1	2	0	0	1	4100000	8	417	2700000	2200000	0
2	3	3	1	0	9100000	20	506	7100000	4500000	0
3	4	3	1	0	8200000	8	467	18200000	3300000	0
4	5	5	0	1	9800000	20	382	12400000	8200000	0

```python
removed_features
```

```
{'bank_asset_value', 'loan_amount', 'luxury_assets_value'}
```

- Features with a correlation greater than 0.8 were identified as highly correlated.
- Highly correlated features were removed to reduce multicollinearity.
- Removing these features improved model performance and interpretability, and reduced complexity for modelling and prediction.

Feature Scaling: Normalisation

```
from sklearn.preprocessing import MinMaxScaler
```

```
mms = MinMaxScaler()
```

Min-Max scaling was applied to normalize column values between 0 and 1, ensuring all features contribute equally without skewing due to varying scales.

	loan_id	no_of_dependents	education	self_employed	income_annum	loan_term	cibil_score	residential_assets_value	commercial_assets_value	loan_status
0	1	2	1	0	0.969072	0.555556	0.796667	0.079310	0.907216	1
1	2	0	0	1	0.402062	0.333333	0.195000	0.089655	0.113402	0
2	3	3	1	0	0.917526	1.000000	0.343333	0.241379	0.231959	0
3	4	3	1	0	0.824742	0.333333	0.278333	0.624138	0.170103	0
4	5	5	0	1	0.989691	1.000000	0.136667	0.424138	0.422680	0

Scaled Columns: income_annum, cibil_score, residential_assets_value and commercial_assets_value

Data Modelling:

Train-Test Splitting

```
# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=56)
```

The dataset was split into 70% training and 30% testing sets to ensure effective model training and evaluation.

Models Applied

1. Decision Tree Classifier
2. Random Forest CLassifier
3. Logistic Regression
4. Support Vector Classifier
5. K Nearest Neighbour

Tabular Representation of Accuracy Metrics

- The models were compared based on their accuracy scores.
- Decision Tree Classifier showed the best performance, followed by Random Forest Classifier.

	model name	accuracy score
0	DecisionTreeClassifier	0.958697
1	RandomForestClassifier	0.957903
2	LogisticRegression	0.908658
3	SVC	0.921366
4	KNeighborsClassifier	0.892772

Conclusion

- The project involved comparing the performance of various machine learning models on a dataset.
- Data preprocessing, feature selection, and exploratory data analysis were key steps in preparing the data.
- Models applied included Decision Tree, Random Forest, Logistic Regression, SVC, and KNN.
- Among the models tested, the Decision Tree Classifier achieved the highest accuracy in the project.
- Its singular decision-making process proved effective, outperforming other models in this regard.