Lending Club Case Study

APFE23704419 - Prasad YVVSN Pechetti

26-June-2024

Lending Club Case Study– Overview

Problem Statement

Business problems are solved using EDA. In this case study, apart from applying the techniques in EDA, also a basics of risk analytics in banking and financial services and data is used to minimize the risk of losing money while lending to customers.

Business Objectives

This company is the largest online loan marketplace, facilitating personal loans, business loans, and financing of medical procedures. Borrowers can easily access lower interest rate loans through a fast online interface.

Like most other lending companies, lending loans to 'risky' applicants is the largest source of financial loss (called credit loss). Credit loss is the amount of money lost by the lender when the borrower refuses to pay or runs away with the money owed. In other words, borrowers who default cause the largest amount of loss to the lenders. In this case, the customers labelled as 'charged-off' are the 'defaulters'.

If one is able to identify these risky loan applicants, then such loans can be reduced thereby cutting down the amount of credit loss. Identification of such applicant's using EDA is the aim of this case study.

In other words, the company wants to understand the driving factors (or driver variables) behind loan default, i.e. the variables which are strong indicators of default. The company can utilize this knowledge for its portfolio and risk assessment.

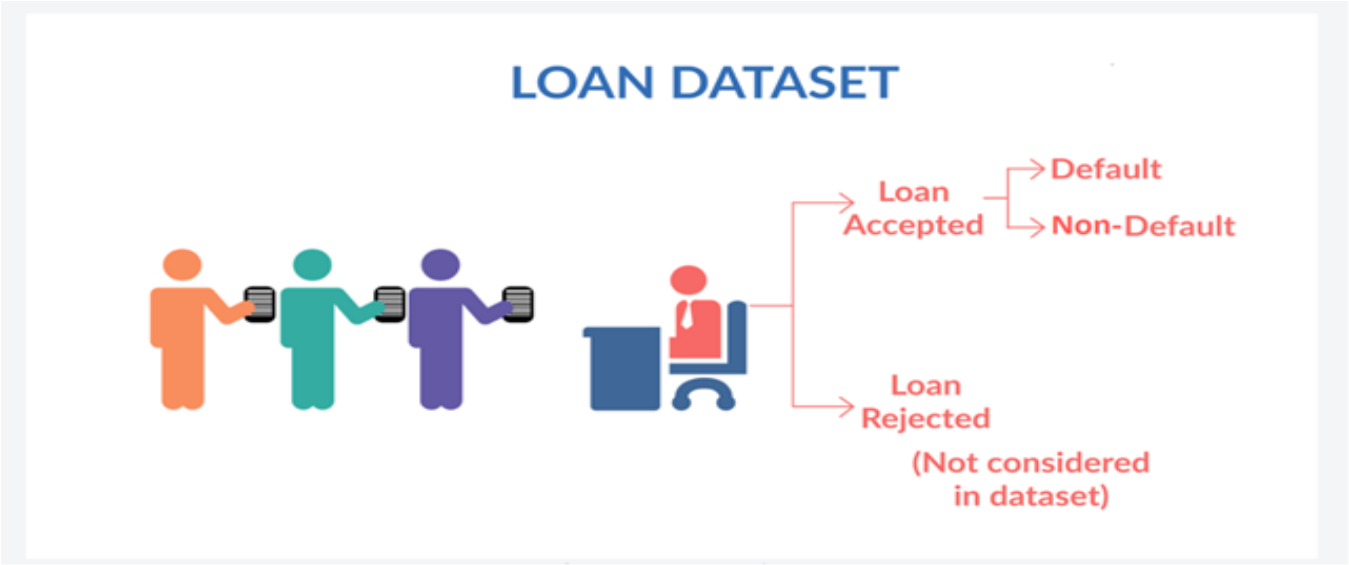
To develop your understanding of the domain, you are advised to independently research a little about risk analytics (understanding the types of variables and their significance should be enough).

Lending Club Case Study– Loan DataSet

About the dataset:

When a person applies for a loan, there are **two types of decisions** that could be taken by the company:

1. **Loan accepted:** If the company approves the loan, there are 3 possible scenarios described below:
    - **Fully paid**: Applicant has fully paid the loan (the principal and the interest rate)
    - **Current**: Applicant is in the process of paying the instalments, i.e. the tenure of the loan is not yet completed. These candidates are not labelled as 'defaulted'.
    - **Charged-off**: Applicant has not paid the instalments in due time for a long period of time, i.e. he/she has **defaulted** on the loan
2. **Loan rejected**: The company had rejected the loan (because the candidate does not meet their requirements etc.). Since the loan was rejected, there is no transactional history of those applicants with the company and so this data is not available with the company (and thus in this dataset)



The dataset contains a large number of columns (111) and rows. Here's a summary of the first few rows to help us understand the types of variables present.

Overview of the dataset:

1. Missing Values:

There are 68 columns with missing values.
Many columns have a significant proportion of missing values (some 100%).

2. Data Types:

The dataset contains a mix of integers, floats, and objects (strings).
Some columns representing numerical data are inappropriately stored as objects and will need conversion (e.g., int_rate).

3. Summary Statistics:

Provides insights into the central tendency and dispersion of numerical variables.
Important variables for our analysis (e.g., loan_amnt, funded_amnt, annual_inc, etc.)

Lending Club Case Study– Dataset Analysis

The dataset has been cleaned by removing columns with excessive missing values and converting relevant columns to the appropriate data types. Here are the key changes:

- Columns with more than 90% missing values were dropped.
- The percentage signs were removed and values converted to floats for int_rate and revol_util.
- Rows with missing values in the loan_status column were dropped.

1. Data Cleaning:

- Handle missing values (drop columns with excessive missingness, impute where appropriate).
- Convert data types as needed.

2. Exploratory Data Analysis:

- Analyze the distribution of key variables.
- Investigate the relationships between the target variable (loan_status) and other predictors.
- Identify potential drivers of loan default.

3. Target Variable:

- Focus on understanding and isolating the loan_status variable, especially identifying 'charged-off' loans as defaulters.

Lending Club Case Study– Dataset Cleanup

The dataset has been cleaned by removing columns with excessive missing values and converting relevant columns to the appropriate data types. Here are the key changes:

- Columns with more than 90% missing values were dropped.
- The percentage signs were removed and values converted to floats for int_rate and revol_util.
- Rows with missing values in the loan_status column were dropped.

```
import pandas as pd

# Load the dataset
file_path = 'loan.csv'
loan_data = pd.read_csv(file_path, low_memory=False)

# Display the first few rows of the dataset
loan_data.head()

# Drop columns with more than 90% missing values
 columns_to_drop = missing_values_summary[missing_values_summary['Percentage'] > 90].index
 loan_data_cleaned = loan_data.drop(columns=columns_to_drop)


# Convert appropriate columns to numeric, removing the '%' sign and converting to float for `int_rate` and `revol_util`
loan_data_cleaned['int_rate'] = loan_data_cleaned['int_rate'].str.replace('%', '').astype(float)
loan_data_cleaned['revol_util'] = loan_data_cleaned['revol_util'].str.replace('%', '').astype(float)

# Drop rows with missing values in the target variable `loan_status`
loan_data_cleaned = loan_data_cleaned.dropna(subset=['loan_status'])

# Display the cleaned dataset info to verify changes
loan_data_cleaned.info()
```

Lending Club Case Study– Distribution of Loan Amount

Graph shows that the Distribution of Loan amounts based on the frequency of the loans

Python code :

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
sns.histplot(loan_data_cleaned['loan_amnt'], kde=True)
plt.title('Distribution of Loan Amounts')
plt.xlabel('Loan Amount')
plt.ylabel('Frequency')
plt.show()
```

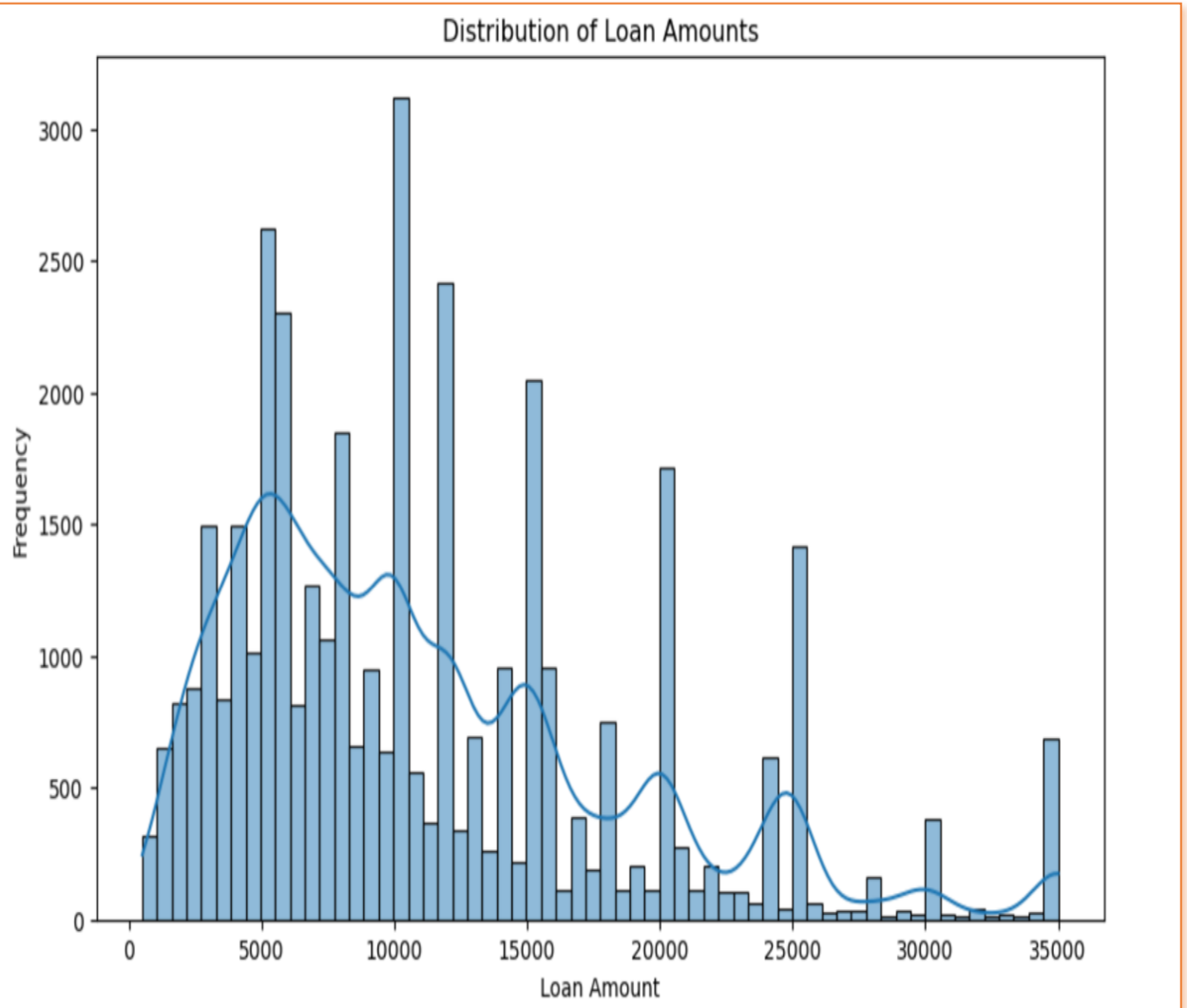
The graph displays a histogram overlaid with a line graph. Here are the key details:

- **Horizontal Axis (X-axis):** Represents loan amounts ranging from 0 to 35,000.
- **Vertical Axis (Y-axis):** Represents frequency (i.e., the number of loans).
- **Bars (Histogram):** Show the distribution of loans within specific loan amount ranges.
- **Line Graph:** Indicates the trend or distribution curve of these loan amounts.

From the graph, we can observe the following:

1. The majority of loans fall within the lower loan amount ranges (e.g., 0 to 5,000).
2. As loan amounts increase, the frequency decreases.
3. There's a peak around 10,000, suggesting that loans in this range are common.

Output :



Lending Club Case Study– Distribution of Interest Rates

Graph shows that the Distribution of Loans based on the Interest Rates of the loans

Python code :

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
sns.histplot(loan_data_cleaned['int_rate'], kde=True)
plt.title('Distribution of Interest Rates')
plt.xlabel('Interest Rate (%)')
plt.ylabel('Frequency')
plt.show()
```

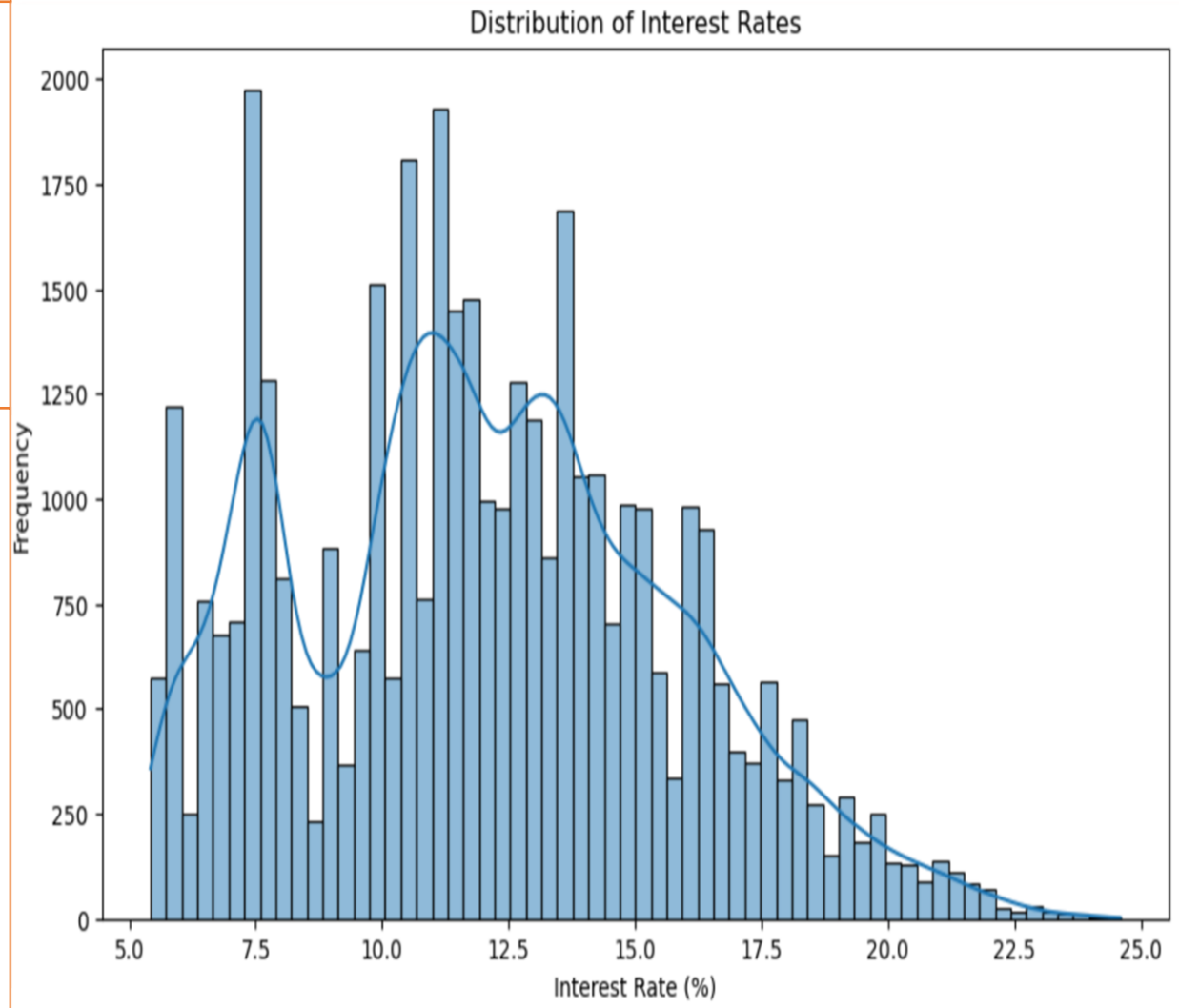
The graph displays a histogram overlaid with a line graph. Here are the key details:

- **Horizontal Axis (X-axis):** Represents interest rates in percentages, ranging from 5.0% to 25.0%.
- **Vertical Axis (Y-axis):** Indicates the frequency of each interest rate occurrence.
- **Bars (Histogram):** Shows the distribution of different interest rates, with the majority clustered between approximately 7.5% and 12.5%.
- **Line Graph:** Indicates the trend or average within the distribution. It peaks around the same range where most bars are concentrated and then tapers off towards higher interest rates.

From the graph, we can observe the following:

1. The most common interest rates fall within the 7.5% to 12.5% range.
2. As interest rates increase beyond this range, their occurrence decreases

Output :



Lending Club Case Study– Loan Status Changed off /Defaulters

Graph shows that the three distinct bars based on the loan status

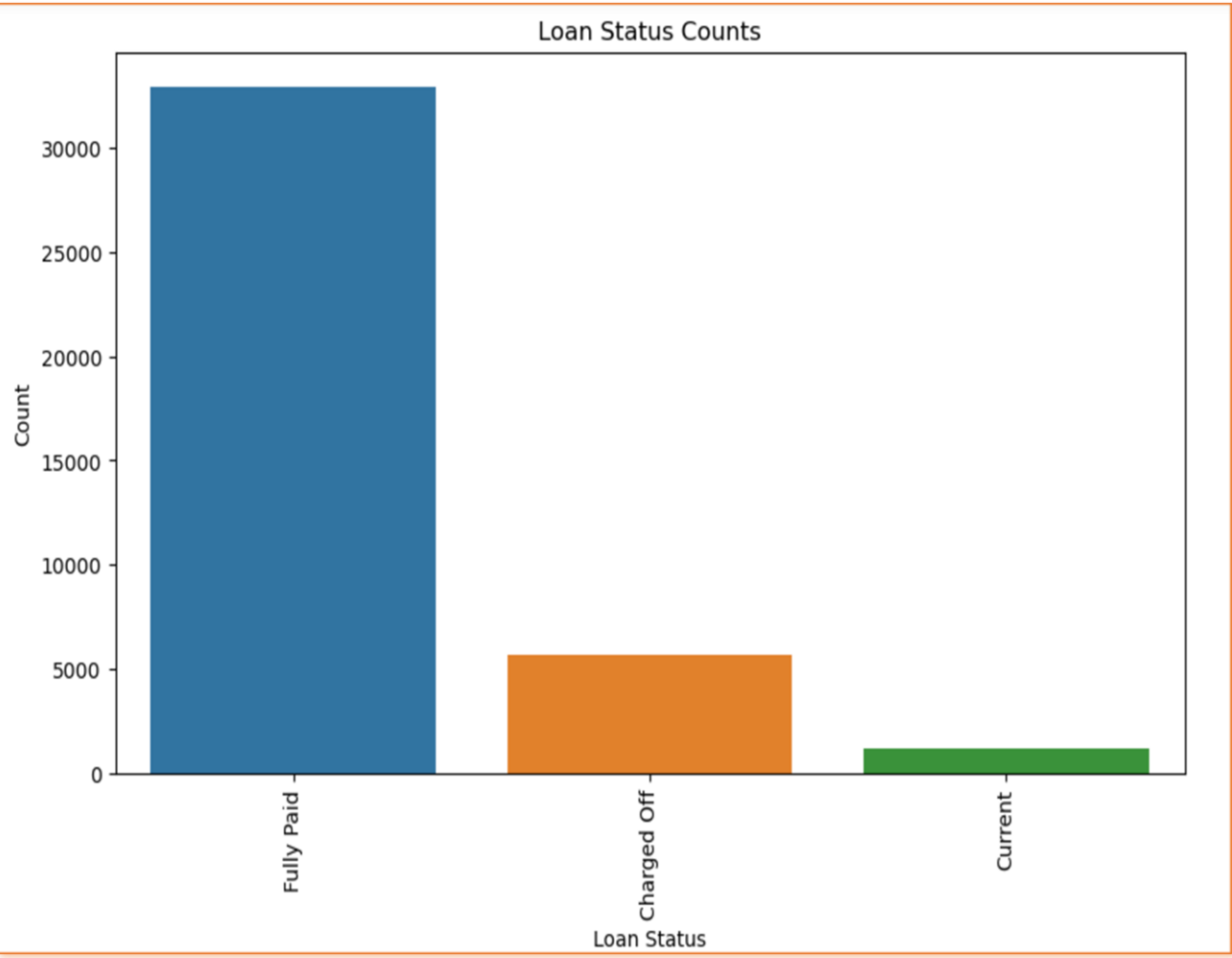
Python code :

```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
sns.countplot(data=loan_data_cleaned, x='loan_status',
order=loan_data_cleaned['loan_status'].value_counts().index)
plt.title('Loan Status Counts')
plt.xlabel('Loan Status')
plt.ylabel('Count')
plt.xticks(rotation=90)
```

The graph displays a bar chart with three distinct bars representing different loan statuses:

1. **Fully Paid:** This status has the highest count, exceeding 30,000 loans.
2. **Charged Off:** The count for this status is significantly lower than “Fully Paid.”
3. **Current:** The “Current” status has the lowest count, close to zero.

Output :



Lending Club Case Study– Correlation Heatmap

Graph shows that the correlation matrix - the relationships between different variables

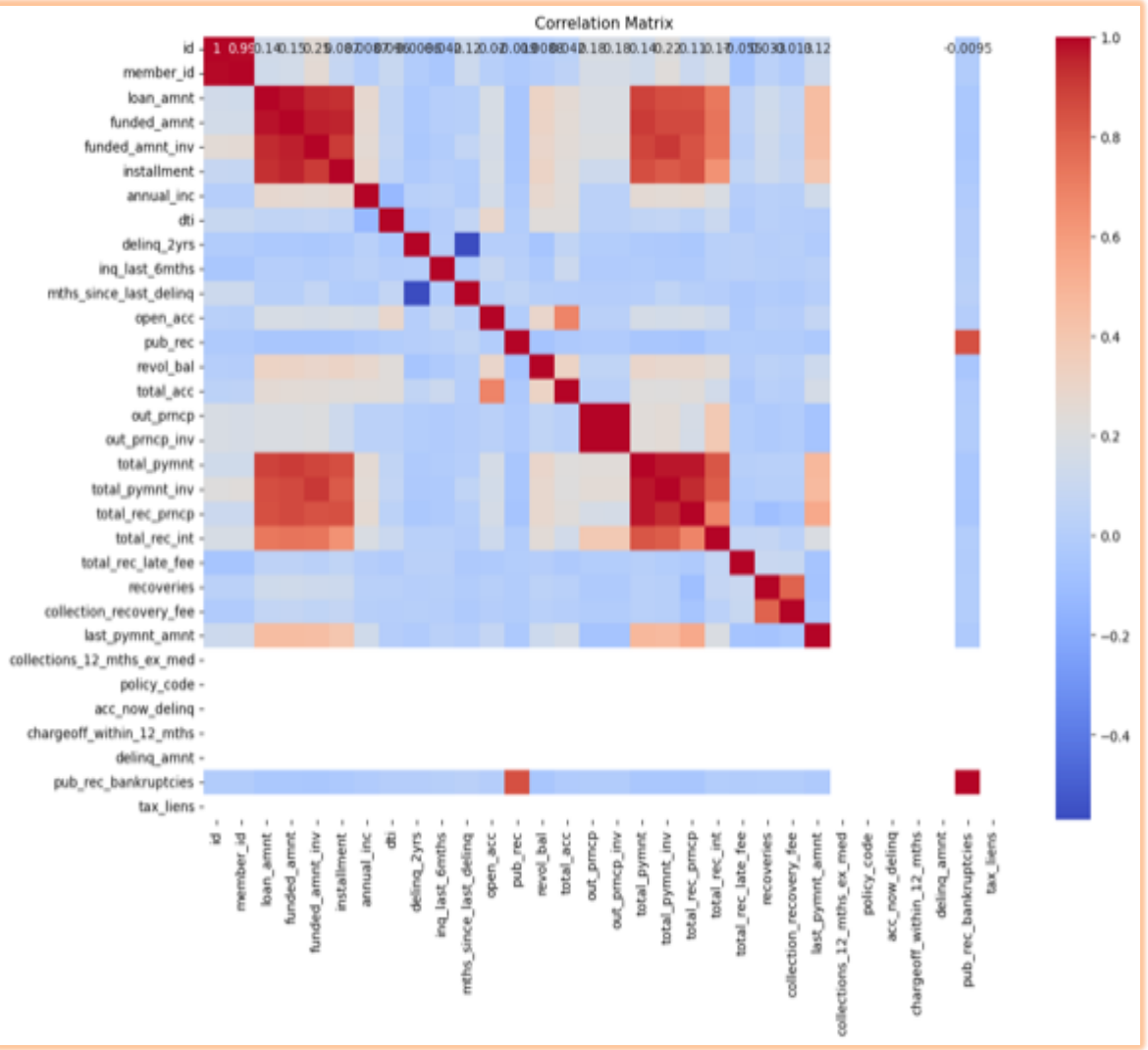
Python code :

```
plt.figure(figsize=(14, 10))
correlation_matrix = loan_data_cleaned.corr()
sns.heatmap(correlation_matrix, annot=False,
cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

A correlation matrix shows the relationships between different variables. Here are some key points based on the matrix:

1. Positive Correlations:
   - Variables with darker red cells have a strong positive correlation. For example, “funded_amnt” and “installment” are positively correlated (0.99).
   - Similarly, “annual_inc” and “total_acc” have a positive correlation (0.96).
2. Negative Correlations:
   - Variables with darker blue cells have a strong negative correlation. For instance, “pub_rec” and “total_acc” are negatively correlated (-0.2).
   - “dti” (debt-to-income ratio) and “pub_rec” also show a negative correlation (-0.35).
3. No Correlation:
   - The diagonal cells (from top left to bottom right) represent perfect positive correlations (value of 1.0) since they compare each variable with itself.

Output :



Thank You

Github : https://prasadpechetti/