Lending Club Case Study

Submitted by-
Arnav Sharma
Arun Gambhir

Table of Content

Problem Statement & Solution

<u>Problem</u>:

Lending Club aims to ensure that applicants are likely to repay their loans before approving them.

<u>Solution</u>:

By analyzing historical loan data, we can identify factors that typically lead to defaults. This insight will enable us to make informed recommendations on whether to approve a loan for a new applicant.

Project Structure

- Group_Facilitator_Arun_Gambhir.ipynb: Main script containing the analysis code.
- loan.csv: Loan dataset.
- README.md: Project documentation.

Import Necessary Libraries & Load Data Frame

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#from datetime import datetime
```

Python

This is to display all the columns & avoid warning while updating the copied dataframe

```python
pd.set_option('display.max_columns', None)
pd.options.mode.chained_assignment = None
```

Python

```python
# load the csv into dataframe & used low_memory=False for the memory warning
loan_df = pd.read_csv("loan.csv", low_memory=False)
loan_df.head()
```

Python

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_length	home_ownership	annual_inc	verification_status	issu
0	1077501	1296599	5000	5000	4975.0	36 months	10.65%	162.87	B	B2	NaN	10+ years	RENT	24000.0	Verified	Dec

Data Understanding

Data Understanding:

1. Data quality issue - **Selecting correct** columns, identification & report
take care of null or out of range values(outliers)

2. Understand **Data Dictionary** & correct them where required in comment section.

Data Cleaning and Preparation

- Loading and inspecting the data.
- Column selection and removal of irrelevant columns.
- Data cleanup: Removal of 'Current' loan statuses.
- Handling missing values and outliers.
- Creating bins for key variables.

Data Cleanup Remove Columns (NAN >35%)

1. Data Understanding

Removing columns that are NA, duplicate & have null values more than threshload.

For this exercise we are taking the threshload as 35%

Then remove the columns that doesn't seem relevant for this execise

[8]

```
def findNonNullColumns(loan_df, fraction):
    return loan_df[loan_df.columns[loan_df.isnull().sum() < fraction * len(loan_df)]].columns
```

[9]

```
print(findNonNullColumns(loan_df, 0.35))
```

```
Index(['id', 'member_id', 'loan_amnt', 'funded_amnt', 'funded_amnt_inv',
       'term', 'int_rate', 'installment', 'grade', 'sub_grade', 'emp_title',
       'emp_length', 'home_ownership', 'annual_inc', 'verification_status',
       'issue_d', 'loan_status', 'pymnt_plan', 'url', 'desc', 'purpose',
       'title', 'zip_code', 'addr_state', 'dti', 'delinq_2yrs',
       'earliest_cr_line', 'inq_last_6mths', 'open_acc', 'pub_rec',
       'revol_bal', 'revol_util', 'total_acc', 'initial_list_status',
       'out_prncp', 'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv',
       'total_rec_prncp', 'total_rec_int', 'total_rec_late_fee', 'recoveries',
       'collection_recovery_fee', 'last_pymnt_d', 'last_pymnt_amnt',
       'last_credit_pull_d', 'collections_12_mths_ex_med', 'policy_code',
       'application_type', 'acc_now_delinq', 'chargeoff_within_12_mths',
       'delinq_amnt', 'pub_rec_bankruptcies', 'tax_liens'],
      dtype='object')
```

Actions & Observation

- After analysing all the columns here are the one that we shortlisted & will do analysis on them.
- Term & Interest rates were string & we are converting those to integers for the analysis.
- For the **loan status as ‘Current’**, we can’t do analysis on those records as they neither good nor defaulter so we removed those records.

Univariate Analysis

- Analysis of key columns like Loan Status, Home Ownership, Purpose, Term, Issue Month, Address State, Grade, etc.
- Visualization using count plots and box plots.
- Insights on the distribution of values in each column.

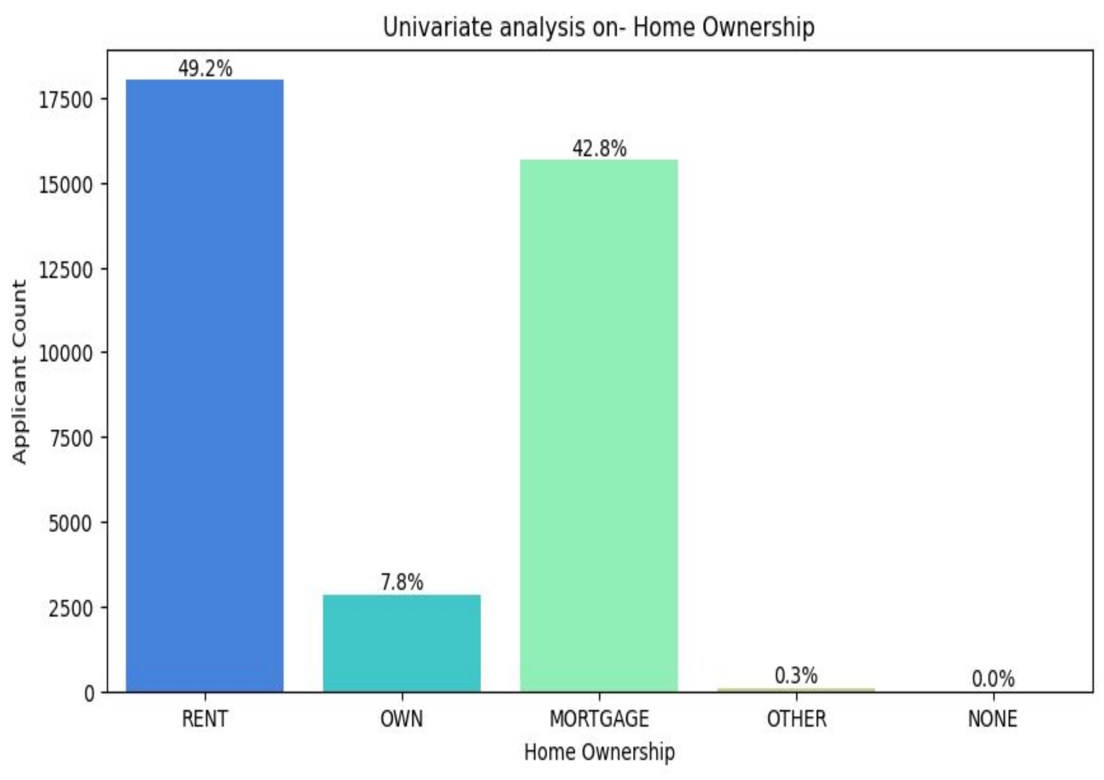
Univariate Analysis (UA) Main Function

```python
#function to create Count plot for univariate analysis
def createCountPlot(loan_df, parameter, label_rotation=False, show_per=True):
    plt.figure(figsize=(10, 5))
    ax = sns.countplot(x=parameter, data=loan_df, hue=parameter, palette='rainbow', legend=False)
    ax.set(title="Univariate analysis on- " + parameter.replace("_", " ").title())
    ax.set_xlabel(parameter.replace("_", " ").title())
    ax.set_ylabel('Applicant Count')

    if show_per:
        # Calculate total
        total = len(loan_df[parameter])
        # Add percentage labels
        for p in ax.patches:
            height = p.get_height()
            percentage = f'{(height / total) * 100:.1f}%'
            ax.annotate(percentage, (p.get_x() + p.get_width() / 2., height),
                        ha='center', va='bottom')

    if label_rotation:
        plt.xticks(rotation=90)

    plt.show();
```

0.0s

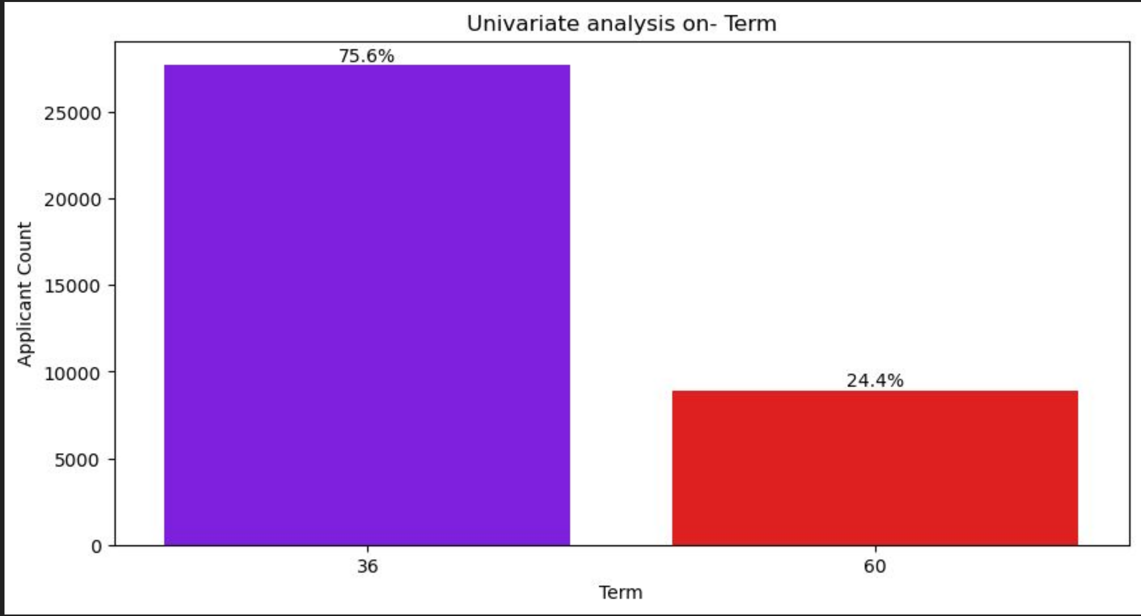
UA on Home Ownership

Observation:
Graph shows that Renters have majority of users for loan takers followed by mortgage and Homeowner.



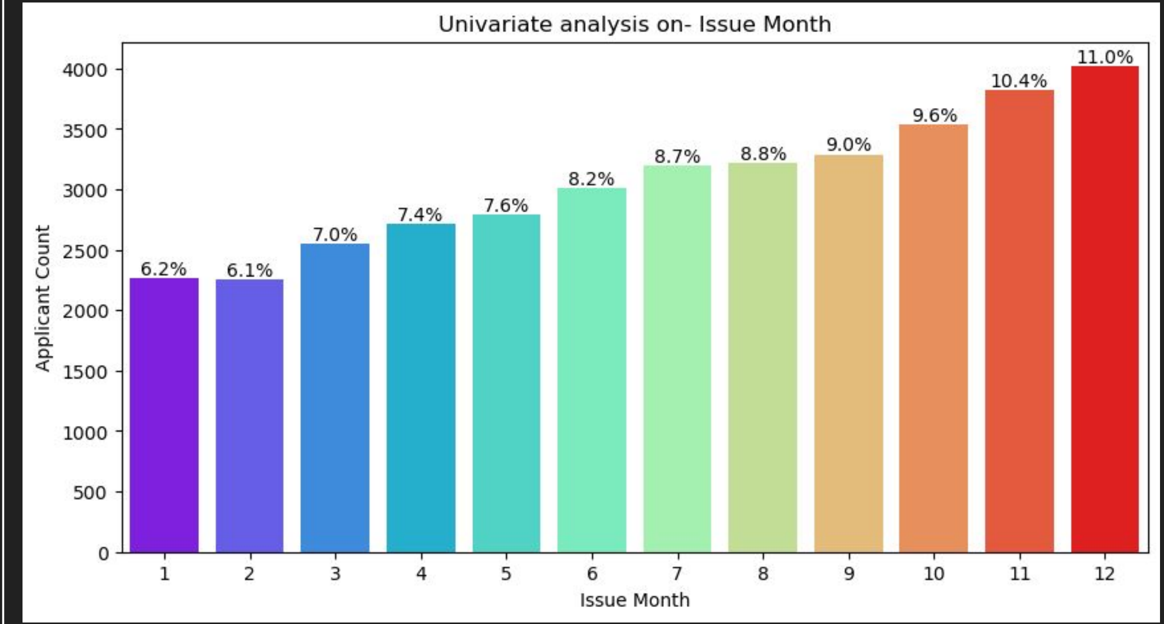
UA on Term Length

Observation:
Graph shows that majority(~75%) of the loans have term length of 36 months and rest (~25%) have length of 60 months.



UA on Issue Month

Observation:
Graph shows that as the Month gets closure to end of the year count of loans issued starts increasing. Therefore, Dec is the month of Majority loans whereas Jan & Feb being the lowest ones.



Bivariate Analysis (BA)

- Relationships between loan_status and other variables.
- Key factors analyzed: Loan Amount, Term, Interest Rate, Grade, etc.
- Visualizations highlight trends in default risk.

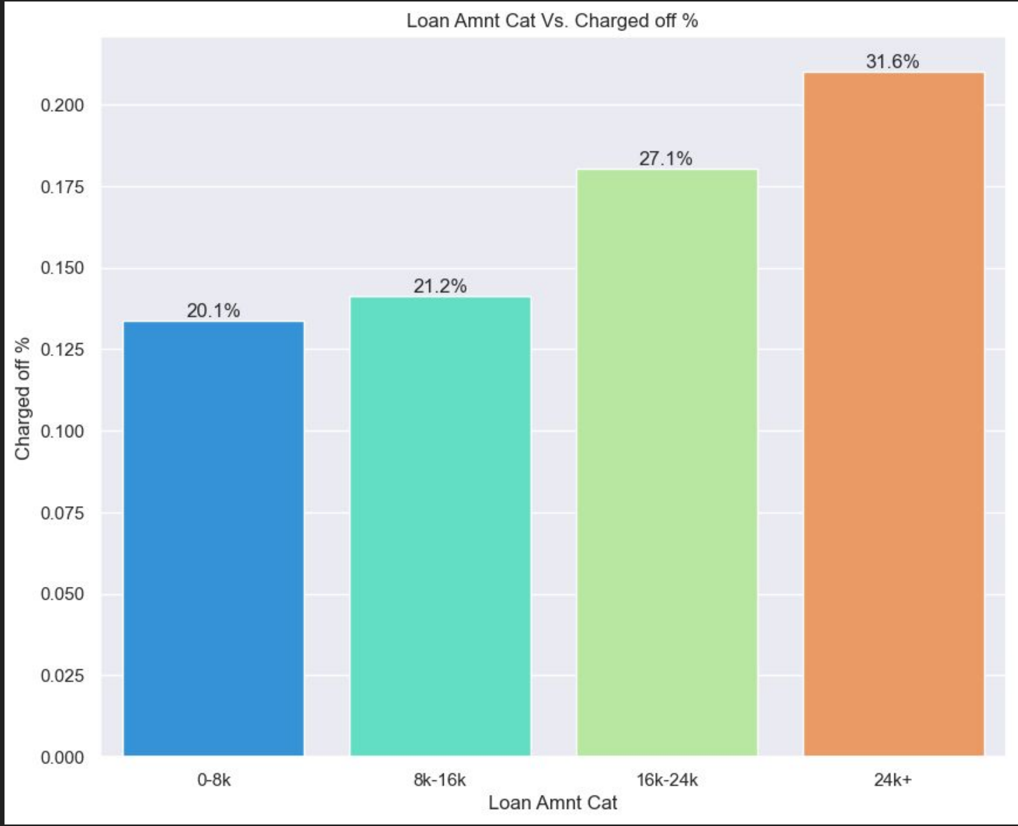
Bivariate Analysis (BA) Function

3.3 Bivariate Analysis

```python
#Function to do Bivariate analysis on loan status = Charged off.
def createBivariantePlot(df_data,feature,label_rotation=False, show_per=True):
    plt.figure(figsize=(10,8))
    sns.set(font_scale=1)

    # Calculate the percentage of loan_status_new =1 per category value
    cat_perc = df_data[[feature, 'loan_status_num']].groupby([feature], as_index=False, observed=True).mean()
    cat_perc.sort_values(by='loan_status_num', ascending=False, inplace=True)
    s = sns.barplot(x = feature, y='loan_status_num', data=cat_perc, hue=feature, palette='rainbow', legend=False)

    if show_per:
        # Calculate total
        total = sum(cat_perc['loan_status_num'])
        # Add percentage labels
        for p in s.patches:
            height = p.get_height()
            percentage = f'{(height / total) * 100:.1f}%'
            s.annotate(percentage, (p.get_x() + p.get_width() / 2., height),
                        ha='center', va='bottom')

    s.set(title= feature.replace("_"," ").title()+' Vs. Charged off %')
    plt.ylabel('Charged off %')
    plt.xlabel(feature.replace("_"," ").title())
    #plt.tick_params(axis='both', which='major')
    #plt.subplots_adjust(wspace=0.2, top=0.9)

    if(label_rotation):
        plt.xticks(rotation=90)
    plt.show();
```

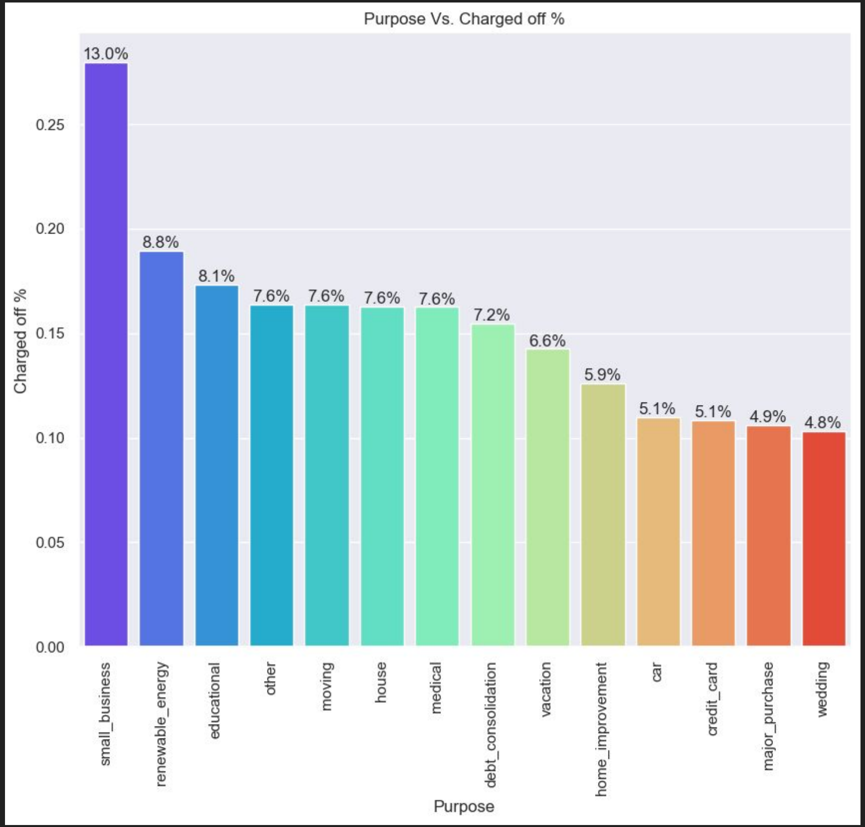
BA on Loan Amount

Observation:
Graph shows that more the loan amount higher the chances of default, suggestion is to keep the loan amount less than 24K USD to avoid high charged off%.



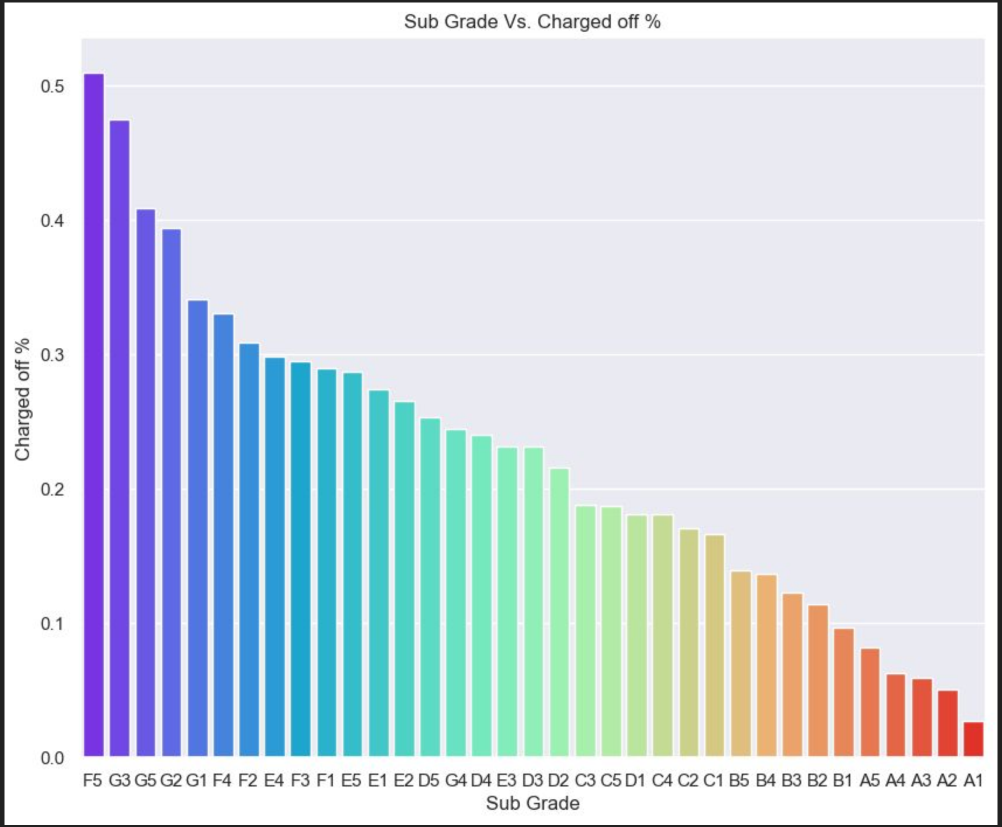
BA on Purpose of Loan

Observation:
Graph shows that some purposes like Car, Credit card, Vacation has lower Charged off % whereas small business, renewable energy has slightly higher charged off %.



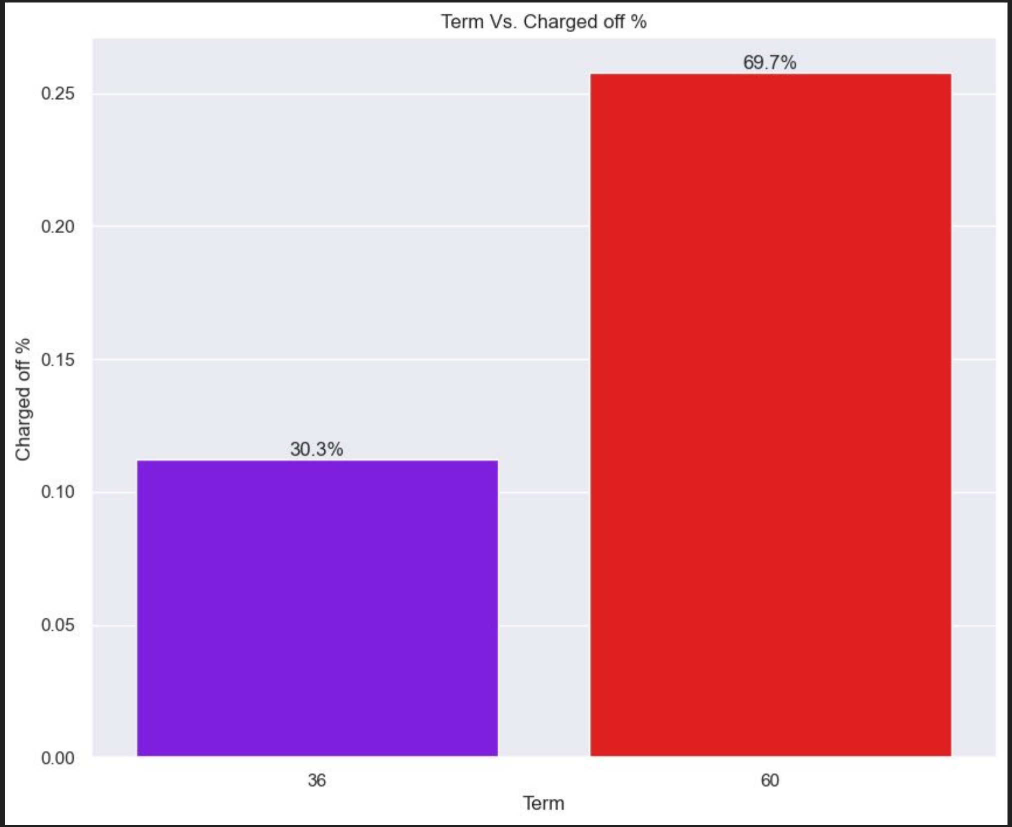
BA on Subgrade

Observation:
Graph shows that lower the Grade length lesser is the Charged Off%, therefore it is recommended to give loans for lower Sub Grades (A1 till C5).

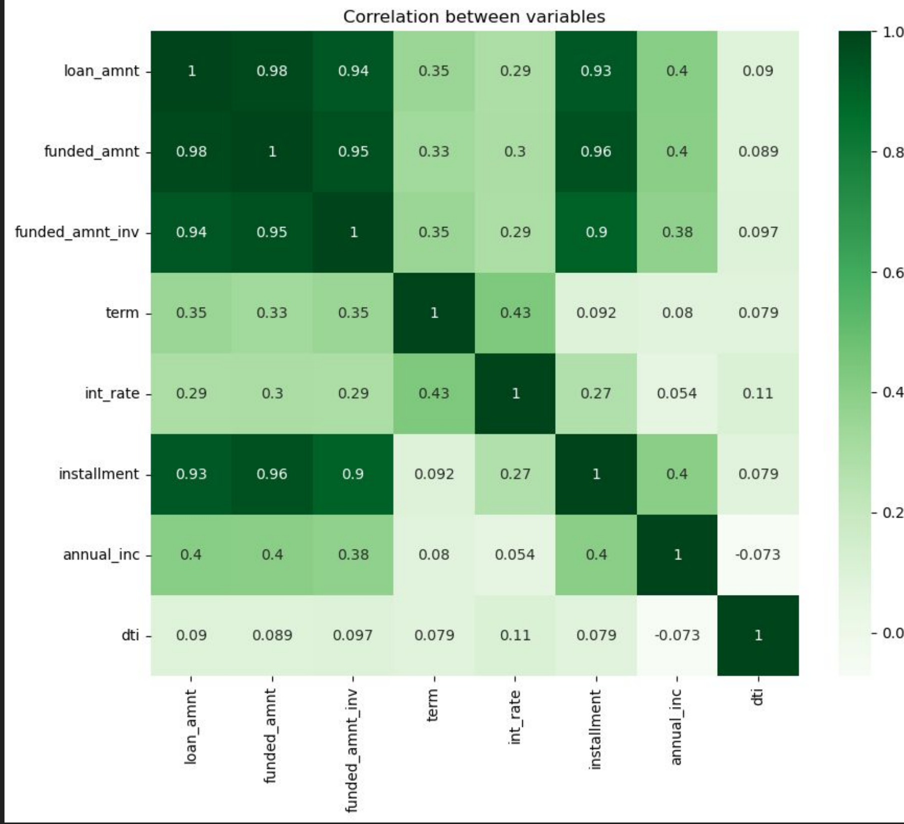


BA on Term Length

Observation:
Graph shows that more term length higher is the Charged Off%, therefore it is recommended to give loans for lower term length (36 Months).



Correlation Analysis



Correlation between two variables lie in the range of -1 to +1 where +1 is positive correlation and -1 is negative correlation and 0 means no correlation.

```
# Create a heatmap of the correlation matrix
plt.figure(figsize=(10,8))
sns.heatmap(loan_df_final[['loan_amnt','funded_amnt', 'funded_amnt_inv','term','int_rate','installment','annual_inc', 'dti']].corr(),cmap='Greens
plt.title("Correlation between variables")
plt.show()
```

Correlation Analysis

Observations for Previous Correlation Matrix:

1. **High Correlation Among Loan Amount, Funded Amount, and Funded Amount Inv:** These values move together, indicating that larger loan requests generally result in higher funding.
2. **Positive Correlation Between Installment and Loan Amount:** Bigger loans lead to larger monthly payments.
3. **Slight Negative Correlation Between Annual Income and DTI:** Higher incomes are associated with lower Debt-to-Income ratios.
4. **Positive Correlation Between Annual Income and Funded Amount:** Higher earners tend to receive larger loan amounts.

Conclusion & Recommendations

- Cap loan amount at $24,000 USD unless the profile is strong.
- Prefer 36-month loan tenure due to lower charged off%.
- Avoid high interest loans as it could lead to higher charged off %.
- Prioritize higher-grade applicants as they have lower charged off%..
- Favor applicants with home ownership followed by mortgage and rent.
- Improve the verification process.
- Validate loan processes in states with high default rates.
- Scrutinize loans for small businesses and renewable energy purposes.
- Be cautious with loans issued during holiday seasons.

Contact Information

For questions, feedback, or further information, contact:

Arun Gambhir (arungambhir@gmail.com)

Arnav Sharma (arnavsharma7991@gmail.com)