Introduction

This presentation explores loan default patterns on the Lending Club platform. We'll analyze borrower characteristics and their relation to loan repayment.

About Lending Club Data Set

The loan data contains data for all approved loans from 2007 to 2011. The data contains 3 types of loan statuses - Charged Off, Current, and Fully Paid. A data dictionary is provided in a separate file to tell us more about each column in the table.

Problem Statement

- **Goal:** Identify factors influencing loan defaults on Lending Club.
- **Business Impact:** Understanding borrower risk profiles allows for better loan decisions, minimizing defaults, and maximizing profits.

Analysis Approach

1. **Data Exploration and Data Cleaning:** We'll get familiar with the data, and understand borrower characteristics and loan details. And clean garbage data not required for analysis
2. **Univariate Analysis:** We'll examine individual variables (e.g., loan amount, income) to identify potential risk factors.
3. **Bivariate Analysis:** We'll explore relationships between borrower characteristics and loan default rates (e.g., how interest rate and annual income impact default rates).

Library imports

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

import warnings
warnings.filterwarnings("ignore")
```

✓ 1.3s

Initial Shape of Data: (39717, 111)

Data Cleaning

Handling Null Values

During the data cleaning process, one significant observation was the presence of numerous columns containing null values. To ensure the dataset's integrity and relevance, we implemented a method to identify and remove columns where all entries were null.

```
df = df.drop(axis=1, columns = df.columns[df.isna().all()].tolist())
```
✓ 0.0s

This step was crucial in ensuring that our dataset remained manageable and relevant, eliminating columns that did not contribute any meaningful information due to their entirely null values. By streamlining the dataset, we could focus on analyzing and processing columns with actual data, enhancing the overall efficiency and accuracy of subsequent data analysis tasks.

Fixing columns where the entry count is not 39717

To ensure consistency in our dataset, we addressed columns where the entry count did not match the expected total of 39,717 rows. We took specific actions to fill in missing values for critical columns, thereby preserving data integrity and facilitating accurate analysis.

```
df['emp_length'] = df['emp_length'].fillna(0)
df['emp_title'] = df['emp_title'].fillna('Unknown')
```
✓ 0.0s Python

Dropping unnecessary columns

To optimize our dataset for analysis, we identified and dropped columns that were deemed unnecessary due to various reasons such as having all or mostly zero values, containing redundant information, or having a high proportion of missing values.

```
Click to add a breakpoint liens (all values are 0),id,
# next_pymnt_d - only 1140 values are present
# chargeoff_within_12_mths - all values are 0
# pymnt_plan - all values are n
# out_prncp - most values are 0
# out_prncp_inv - most values are 0
# pub_rec_bankruptcies - most values are 0
# acc_now_delinq - all values are 0
# delinq_amnt - all values are 0
# initial_list_status - all values are f
# policy_code - all values are 1
# collections_12_mths_ex_med - all values are 0
# emp_title - 2459 unique values
# application_type - all values are individual
# delinq_2yrs - 32000 values are 0
# pub_rec - 34000 values are 0
df = df.drop(labels=['desc', 'tax_liens', 'id', 'url', 'title', 'zip_code', 'mths_since_last_delinq',
                     'mths_since_last_record','next_pymnt_d', 'chargeoff_within_12_mths', 'pymnt_plan', 'out_prncp', 'out_prncp_inv',
                     'pub_rec_bankruptcies', 'acc_now_delinq', 'delinq_amnt', 'initial_list_status',
                     'policy_code', 'collections_12_mths_ex_med', 'emp_title', 'application_type', 'delinq_2yrs',
                     'pub_rec'], axis=1)
```

Dropping redundant rows

To maintain the quality and consistency of our dataset, we identified and removed rows with missing values in certain critical columns. This approach helps ensure that the dataset remains clean and reliable for analysis.

```
# last_credit_pull_d - has 2 missing values - so deleting the rows
# revol_util - has 50 missing values - so deleting the rows
df = df[df['last_credit_pull_d'].notna()]
df = df[df['revol_util'].notna()]
```

Converting Object Data Types to Numeric or Float

To ensure that our dataset is in the appropriate format for analysis, we converted several columns from object types to numeric types. This conversion is crucial for performing numerical operations and analyses on these columns.

```
# term can be converted to int
df['term'] = df['term'].apply(lambda x: int(x.split()[0]))
```

✓ 0.0s Python

+ Code + Markdown

```
# int_rate can be converted to float
df['int_rate'] = df['int_rate'].apply(lambda x: float(x.split('%')[0]))
```

✓ 0.0s Python

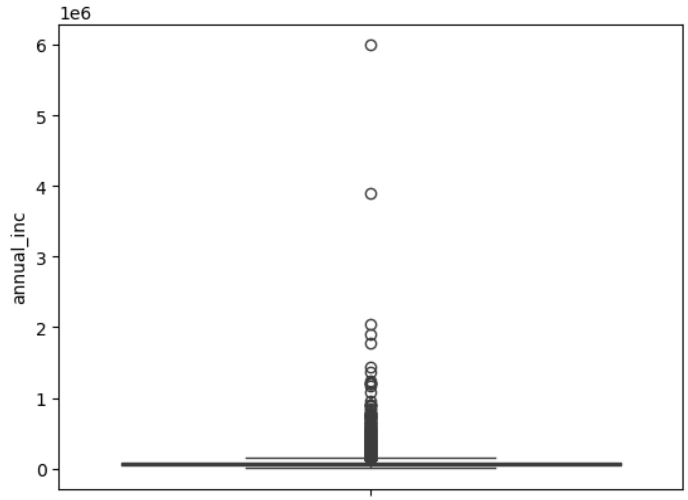
```
# revol_util can be converted to float
df['revol_util'] = df['revol_util'].apply(lambda x: float(x.split('%')[0]))
```

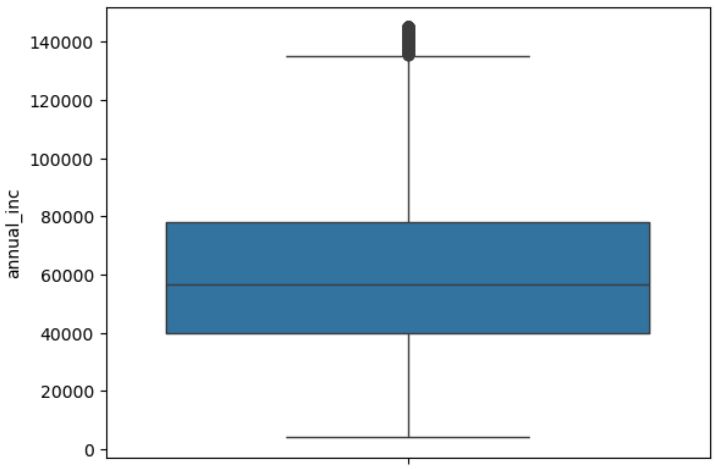
✓ 0.0s Python

Removing outliers

While doing univariate analysis we noticed that some of the data had outliers in them and could be removed. To improve the quality of our dataset, we identified and removed outliers in the `annual_inc` (annual income) and installment columns. This step helps in ensuring that the data is more representative of typical values, thereby enhancing the robustness of our analyses.
In

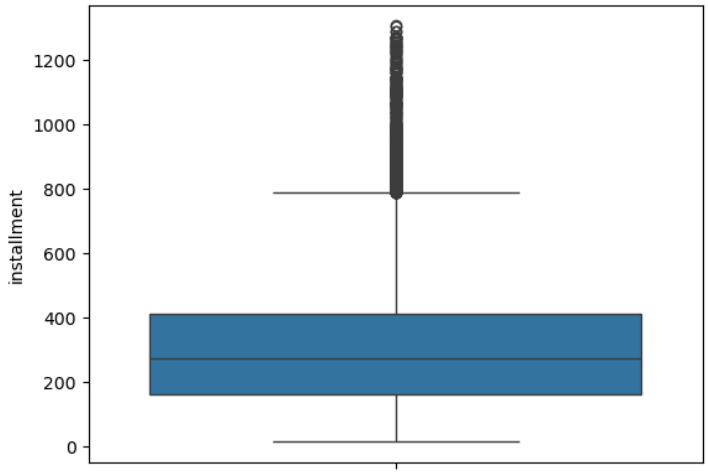
```python
# There's a possibility of outliers in annual_inc, came to know while doing univariate analysis

sns.boxplot(df['annual_inc'])
plt.show()

# Let's remove the outliers

q1 = df['annual_inc'].quantile(0.25)
q3 = df['annual_inc'].quantile(0.75)
iqr = q3 - q1
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr

df = df[(df['annual_inc'] > lower_bound) & (df['annual_inc'] < upper_bound)]

sns.boxplot(df['annual_inc'])
plt.show()
```

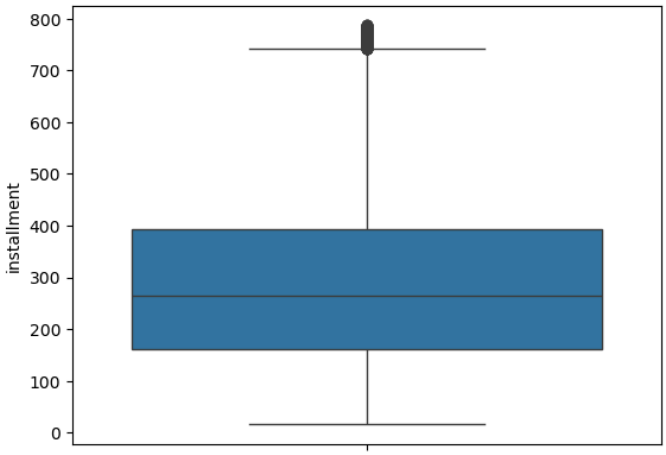





```python
# Remove outliers in installment

sns.boxplot(df['installment'])
plt.show()

q1 = df['installment'].quantile(0.25)
q3 = df['installment'].quantile(0.75)
iqr = q3 - q1
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr

df = df[(df['installment'] > lower_bound) & (df['installment'] < upper_bound)]

sns.boxplot(df['installment'])
plt.show()
```

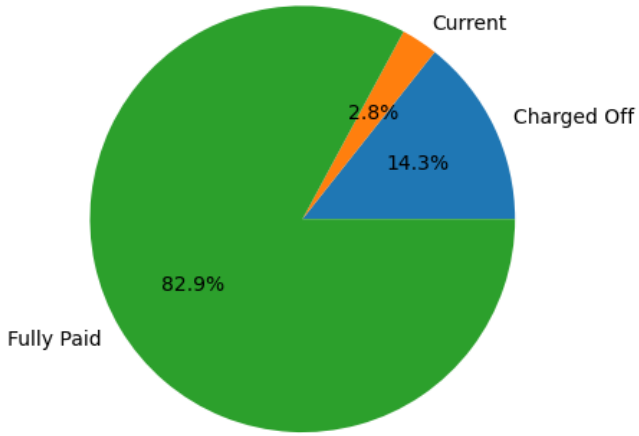
✓ 0.2s

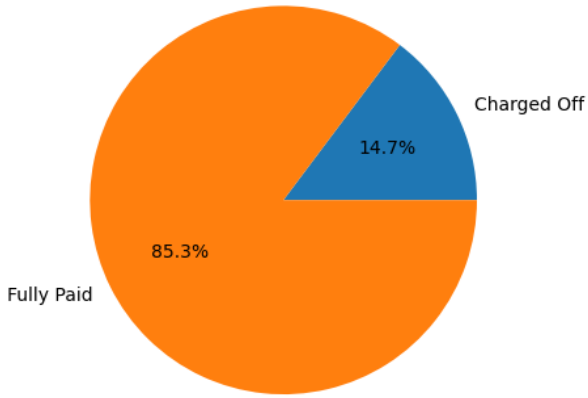




Visualizing the distribution of Loan Statuses

To analyze the distribution of loan statuses and focus on defaulters and non-defaulters, we first visualized the distribution of loan statuses. Subsequently, we filtered out borrowers who are currently paying off their loans, as they do not yet fall into the categories of defaulters or non-defaulters.

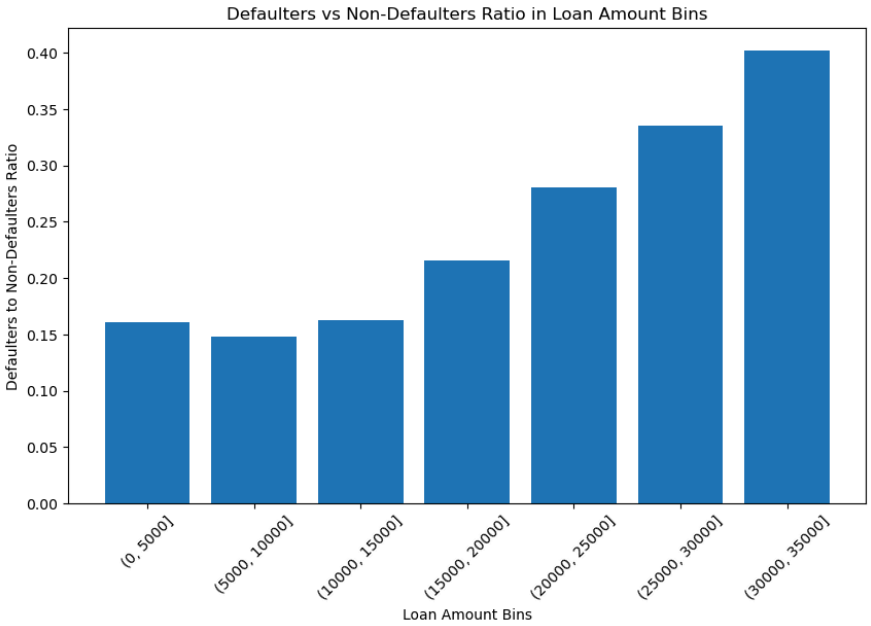




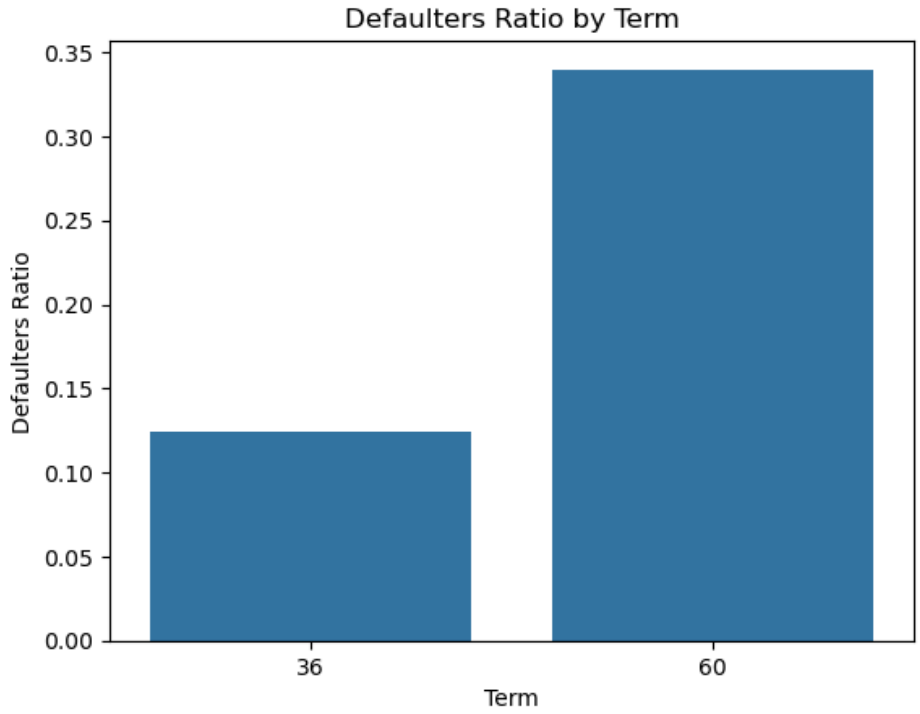
The final shape of data after cleaning the data: (35451, 34)

Observations of Univariate Analysis

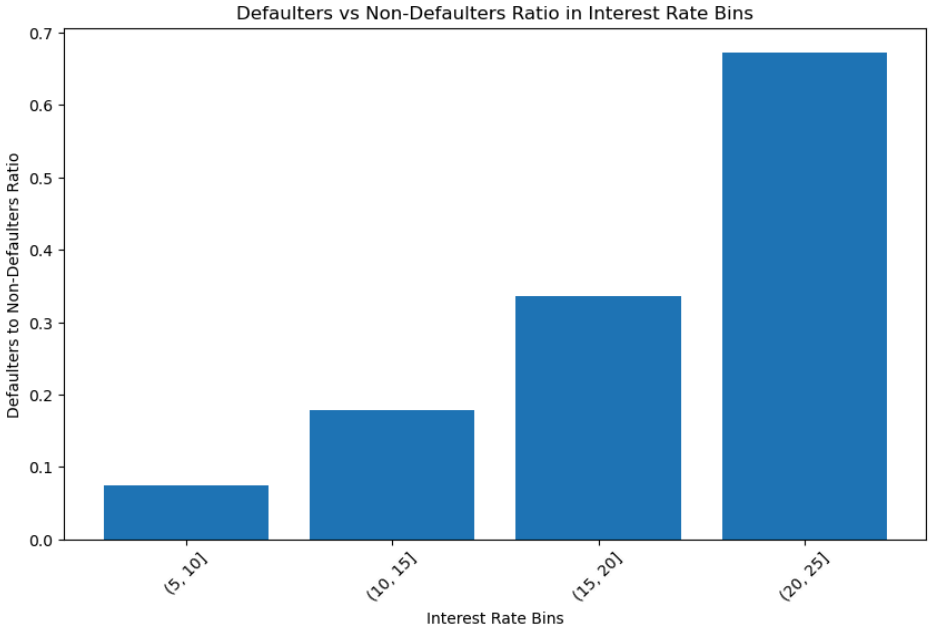
- Loan Amounts - We notice that ratio of defaulters increase as the loan amount increases



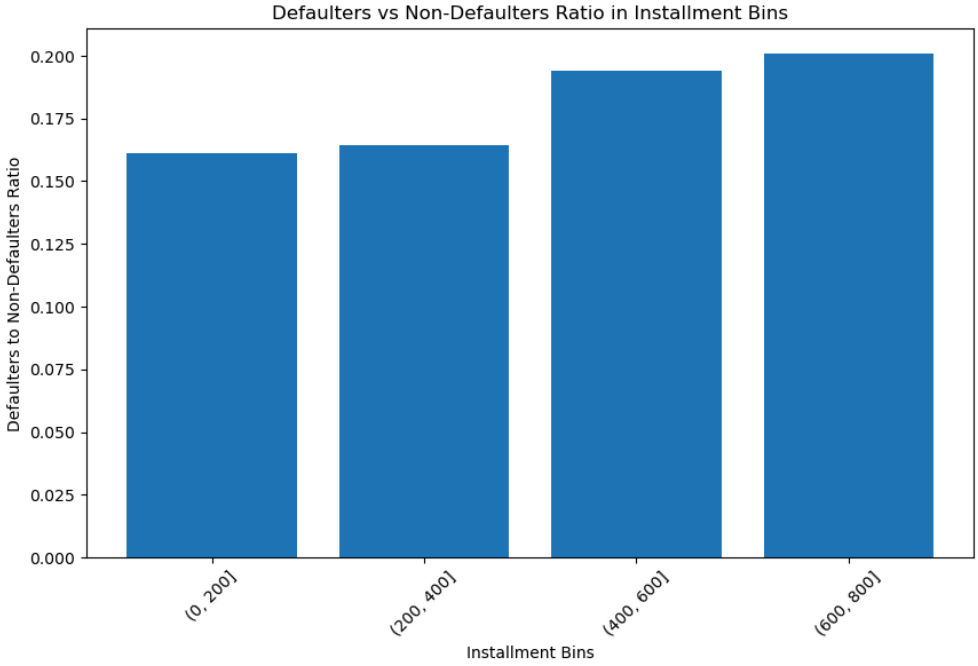
- Term (We notice that as term increases the number of defaulters to non-defaulters ratio also increases. This means that higher the term the higher the defaulters number)



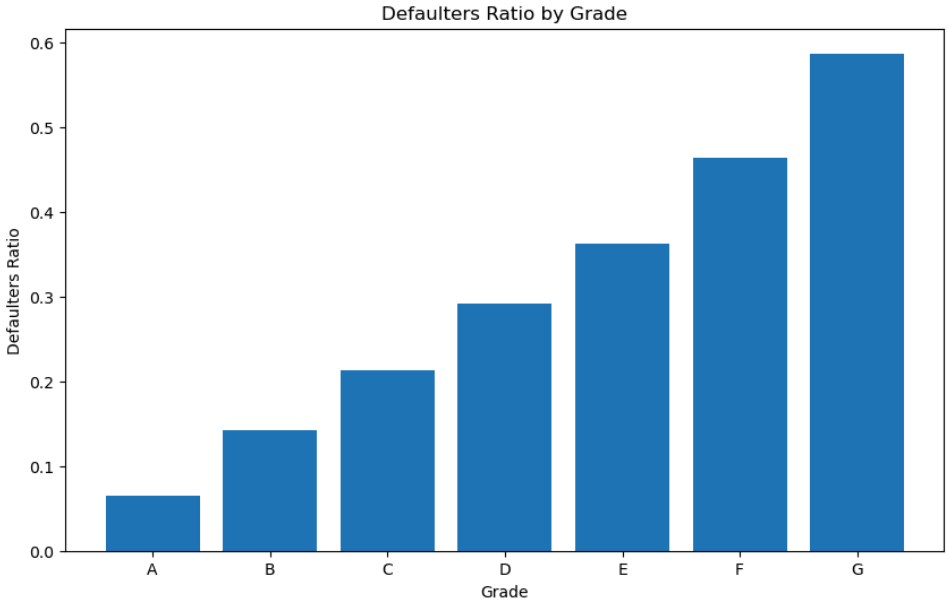
- Interest Rate (We observe that the ratio of defaulters also increases as the interest rate increases. Therefore we can say higher the interest rate higher the chances of defaulting)



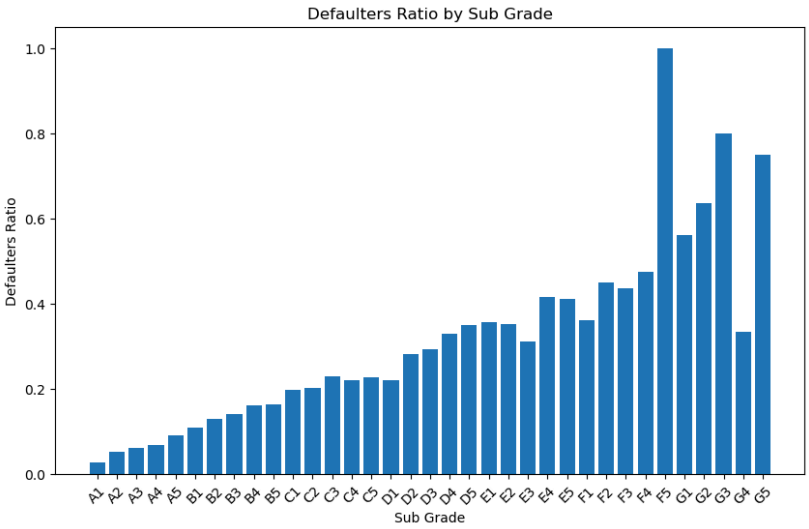
- Installment (We notice a trend for installment and defaulters ratio go hand in hand)



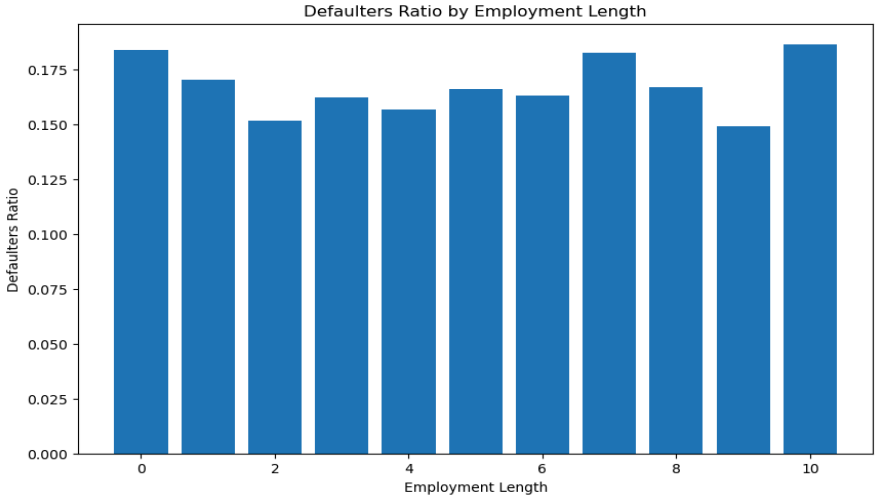
- Grade (We observe that the defaulters rate increases as the grade increases)



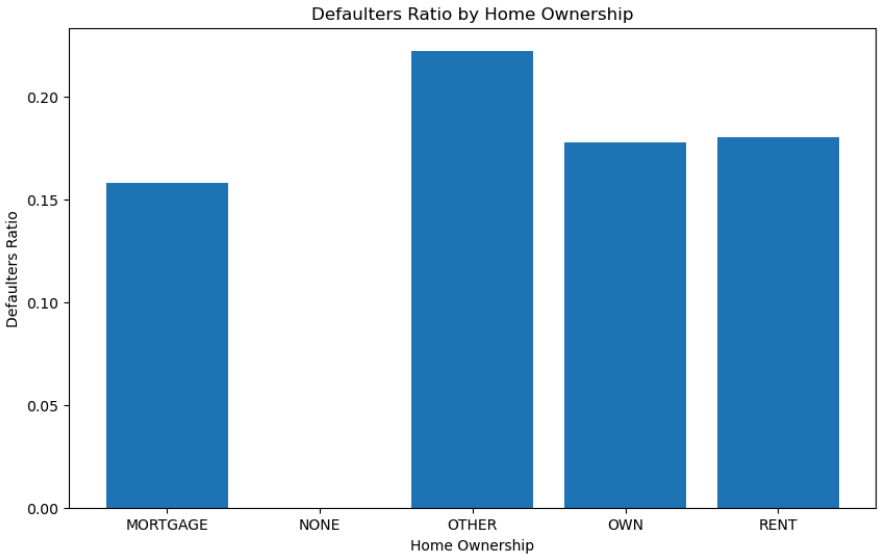
- Sub-Grade (We observe the trend that defaulters increase as sub-grade moves from A1 to G5)



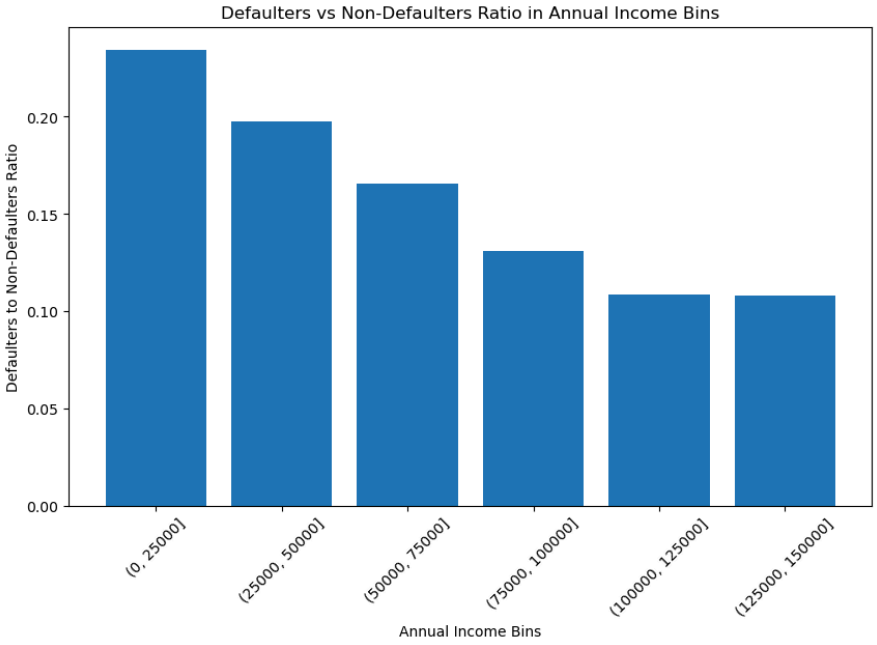
- Employment Length (We observe no correlation between defaulters and the length of their employment)



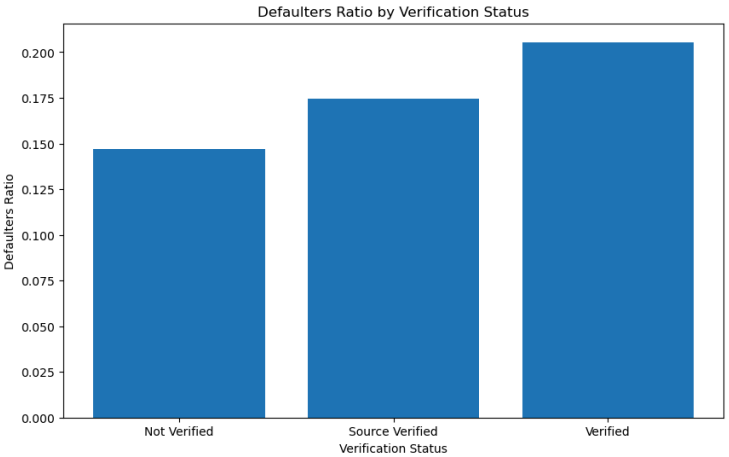
- Home Ownership (We notice no relation with home ownership and defaulters except that "Other" section has more of defaulters. But we don't know what values Others have in it)



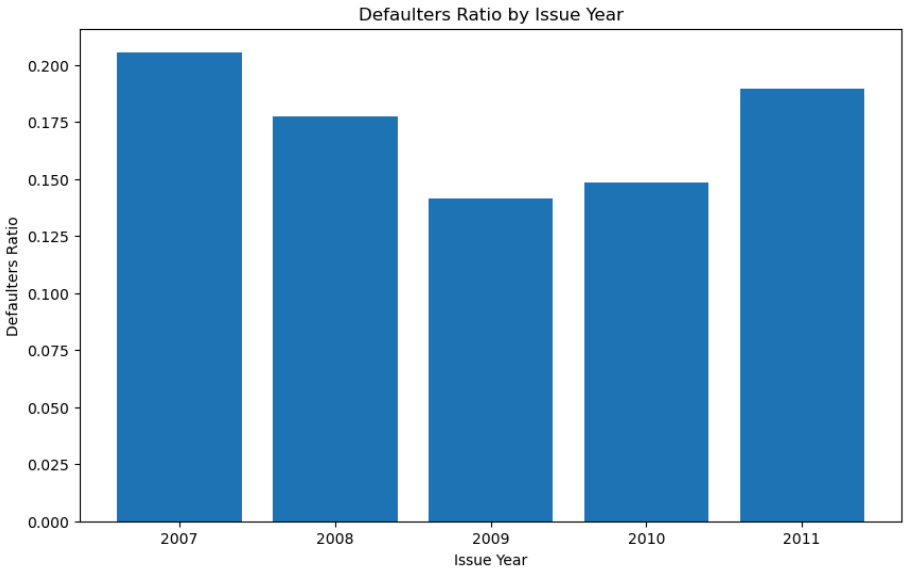
- Annual Income (We observe as annual salary increases ratio of defaulters decreases. There is a co-relation between salary and defaulters)



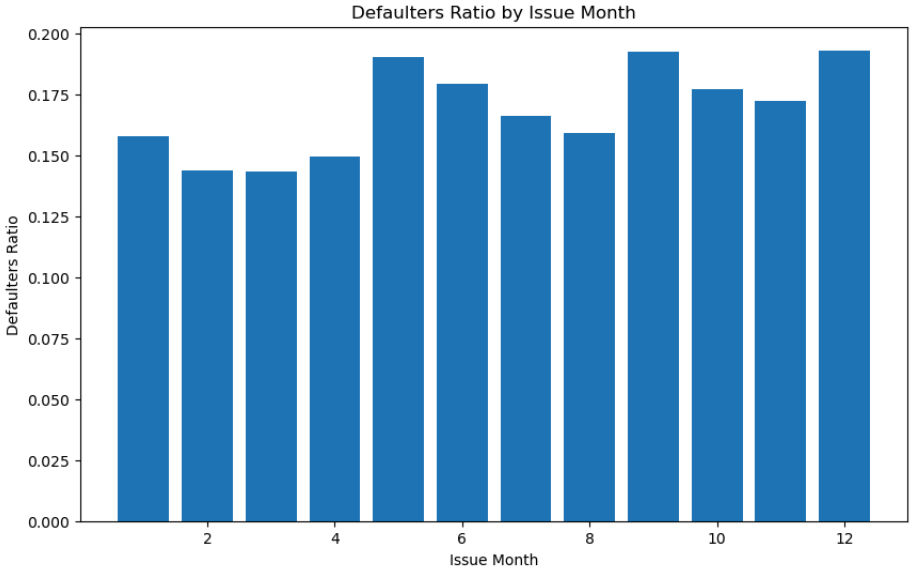
- Verification Status (We observe that defaulters rate increases for verified annual income. So there may be no relation between income verification and defaulters)



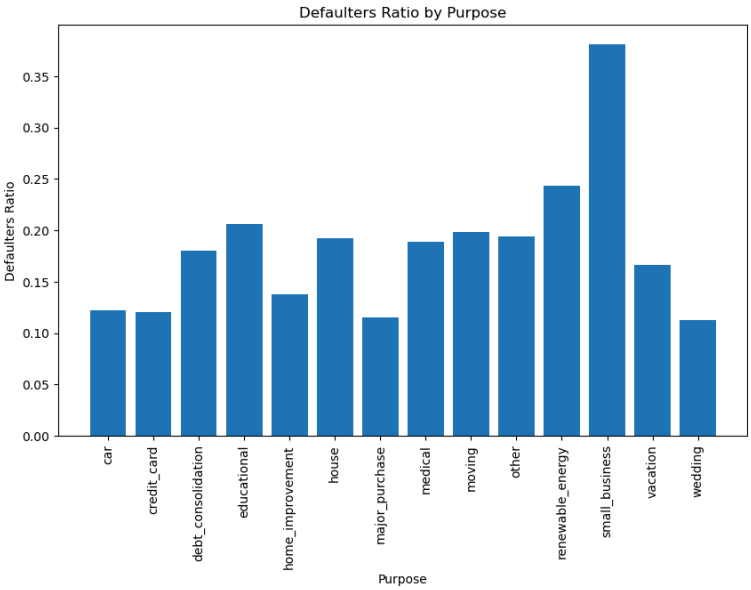
- Issue Year (No relation with issue year and defaulters)



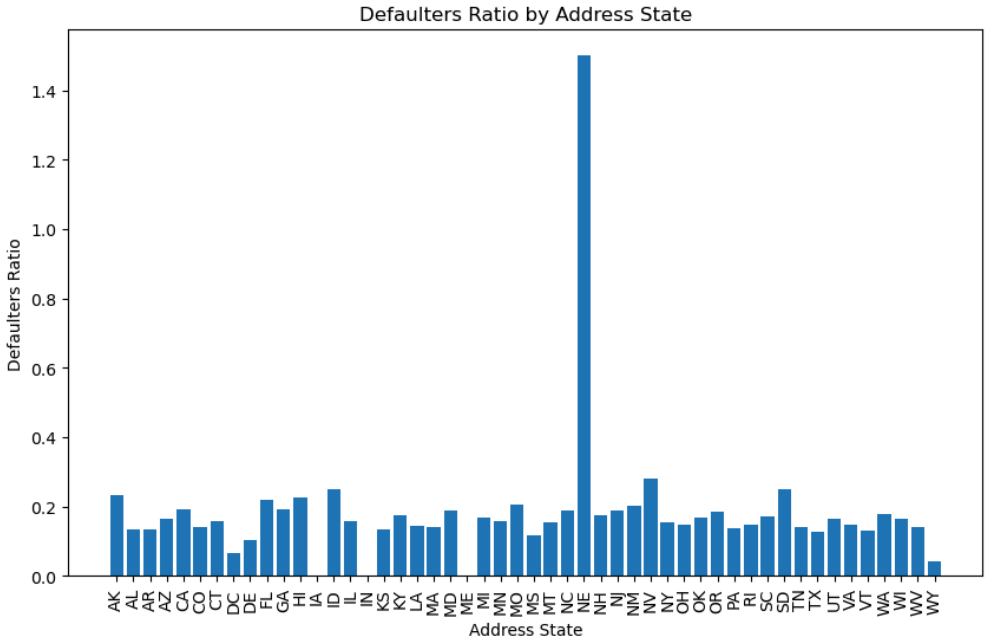
- Issue Month (No relation with issue month and defaulters)



- Purpose (Members taking loan for small business tend to default more)

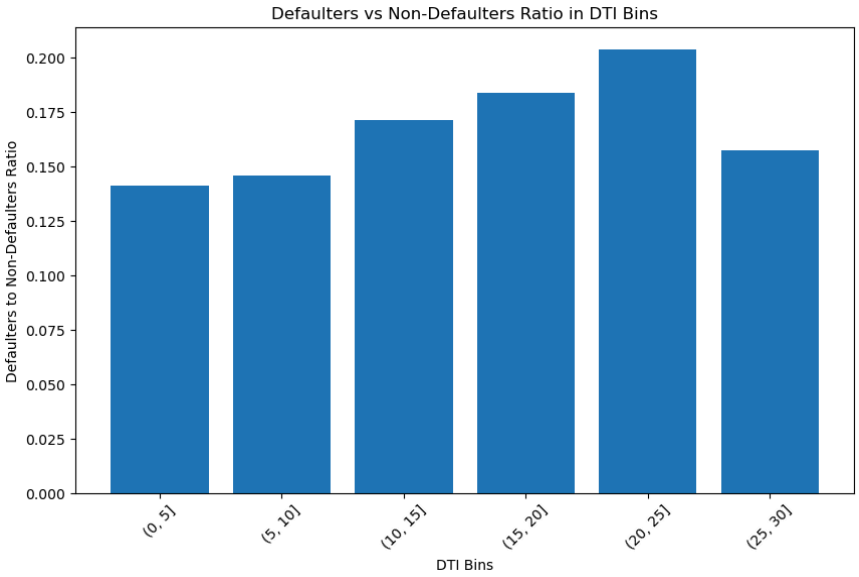


- Address State (We observe that members from Nebraska State tend to default more)

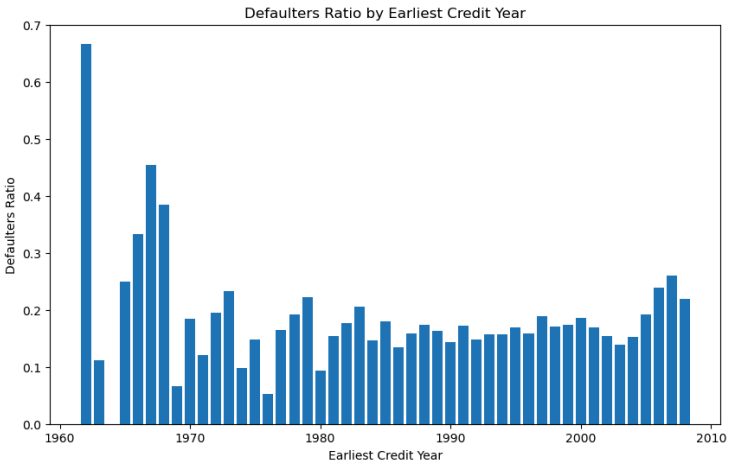


-

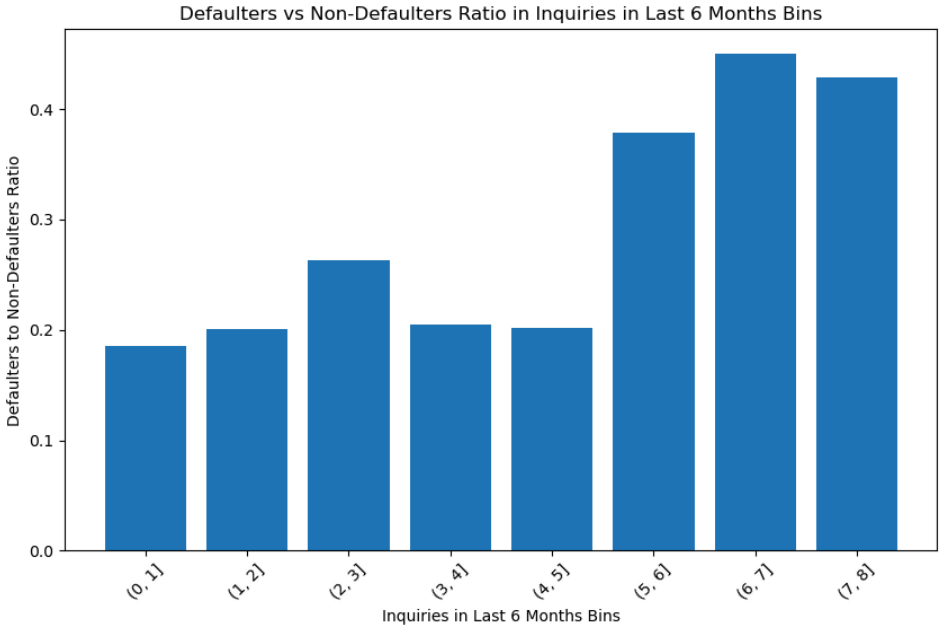
- DTI (We observe that as dti increases defaulters also increase)



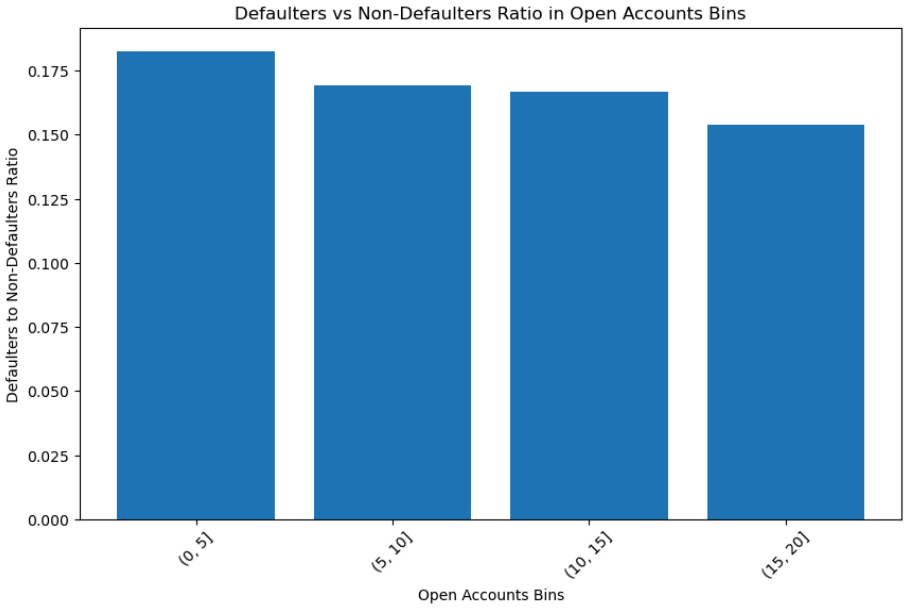
- Earliest credit year (We don't see any relation between year and earliest_cr_line. Only that in 196x we see high defaulters once)



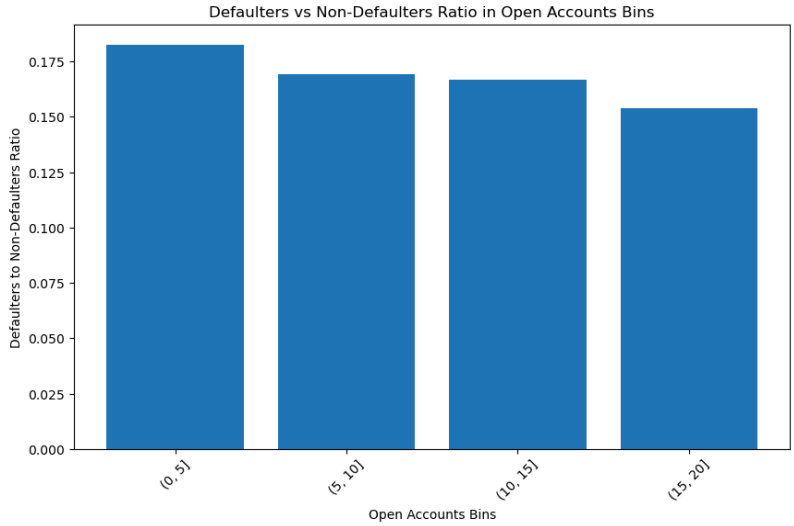
- Inq_last_6mths (No relation between inq_last_6mths and defaulters)



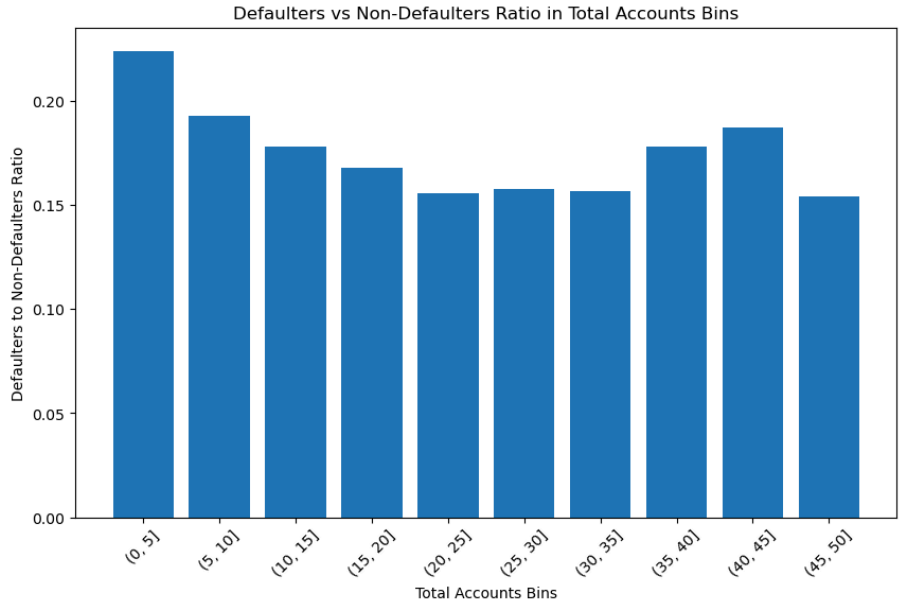
- Open_acc (No relation between open_acc and defaulter. It's almost constant)



- revol_util (We observe that the defaulters ratio increases as the revol_util increases)



- Total Accounts (No relation between Total account bins and defaulters)



Conclusion

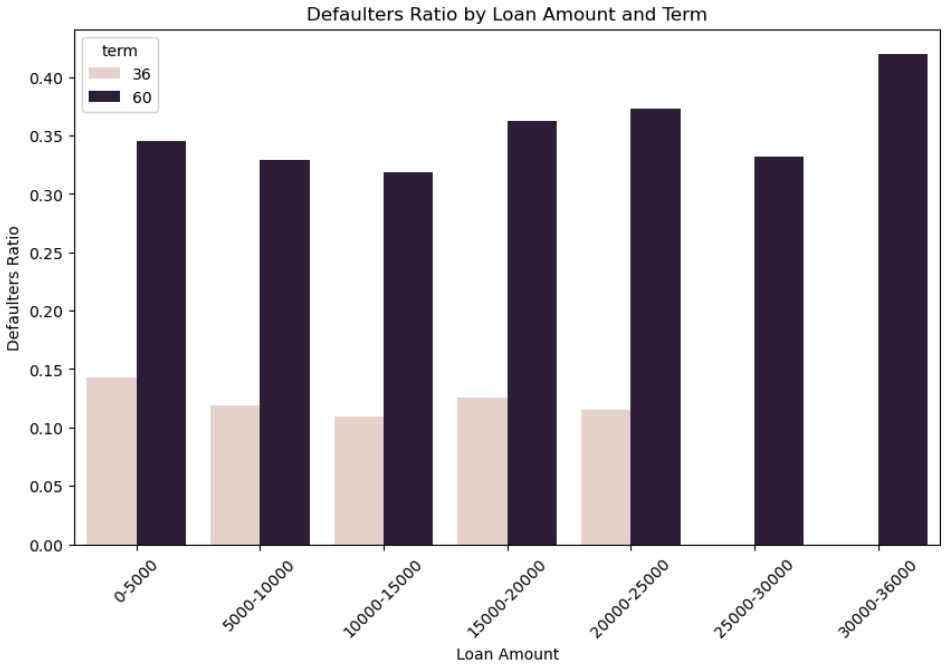
Therefore from the above univariate analysis, we deduced that the following are the important features we should look out for when giving out loans. As they individually have a direct impact on defaulters numbers:

- **loan_amnt**
- **term**
- **interest rate**
- **Installment**
- **grade**
- **sub-grade**
- **annual salary**
- **purpose= small business**
- **state=Nebraska**

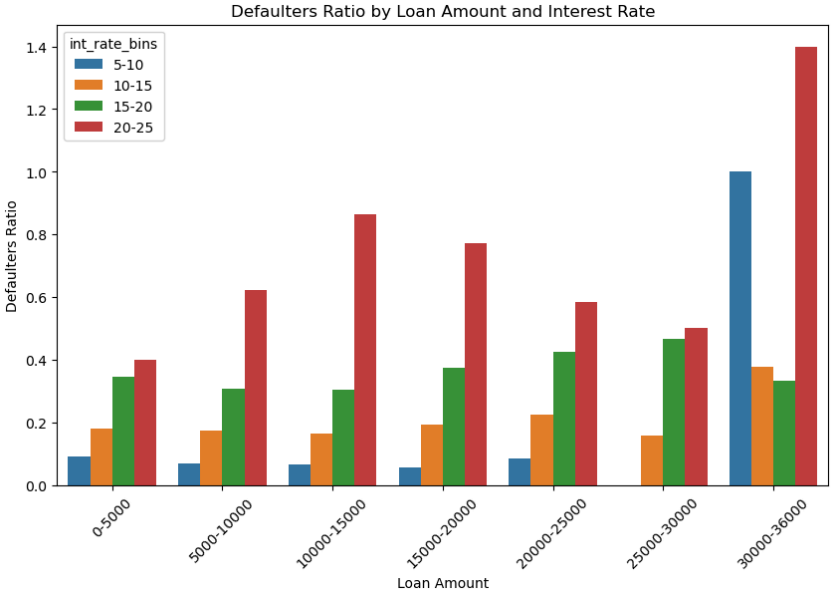
- **dti,**
- **revol_util**

Bivariate Analysis

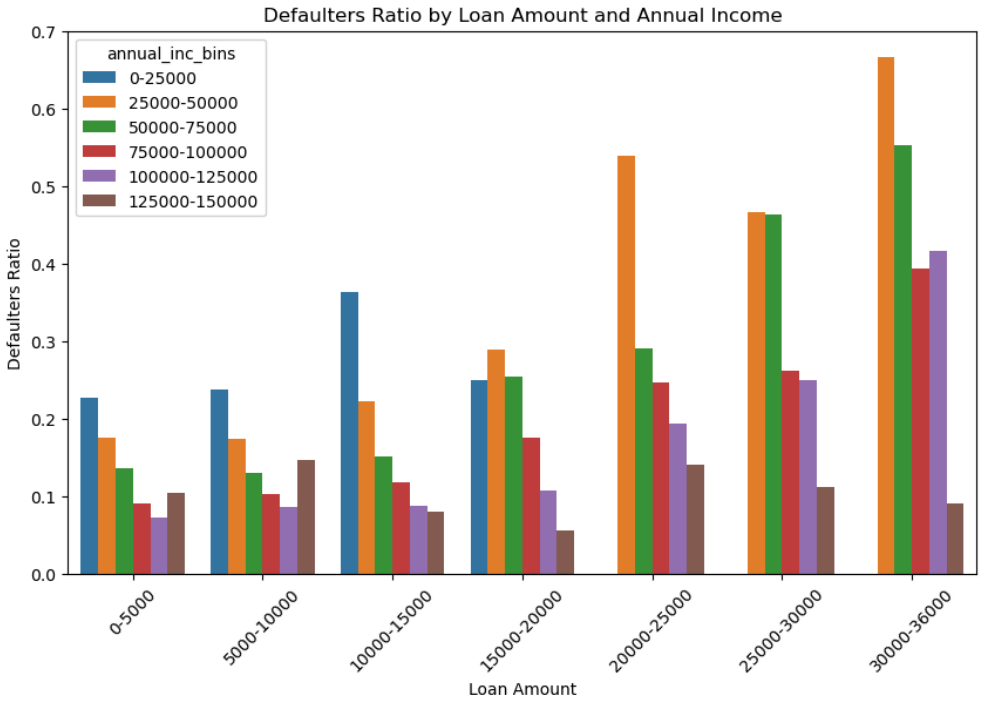
- Loan Amount and Term on Loan Status (We observe that the defaulters ratio increases for every term in each loan_amnt_bin and the trend is consistent. So we notice defaulters usually go for 60-month term period irrespective of the loan amount. That is something we can watch out for when giving out loans)



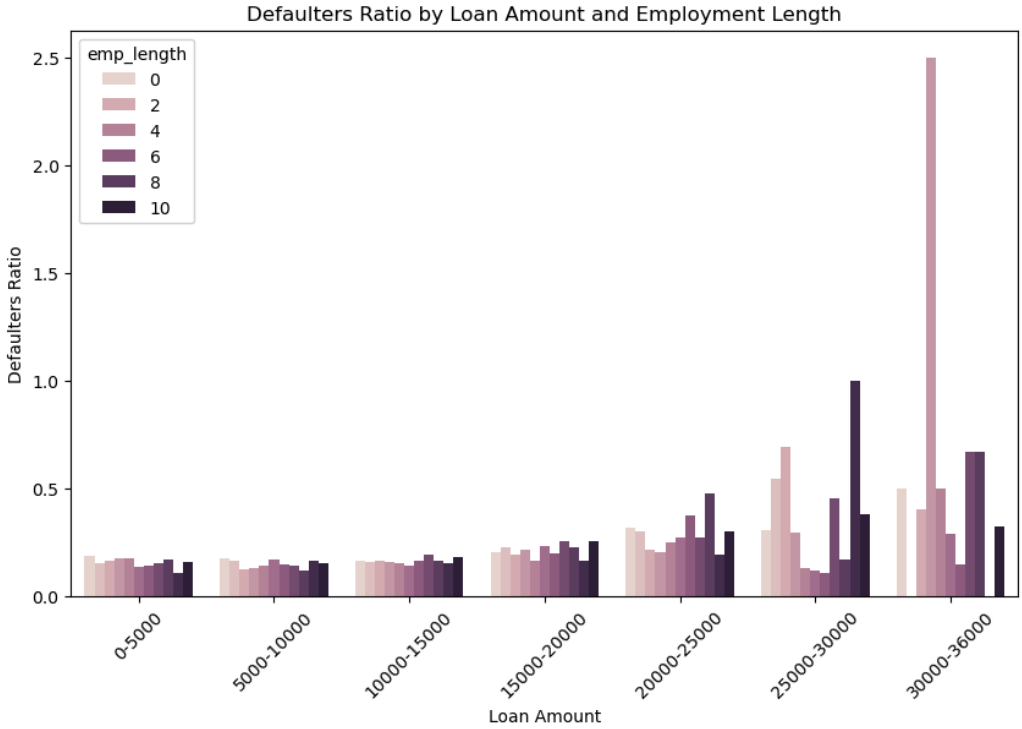
- Loan Amount and Interest Rate (We observe the increase in defaulters ratio as the interest rate increases in each loan_amnt_bins. So we can conclude that higher interest rate may lead to higher number of defaulters regardless of the loan amount)



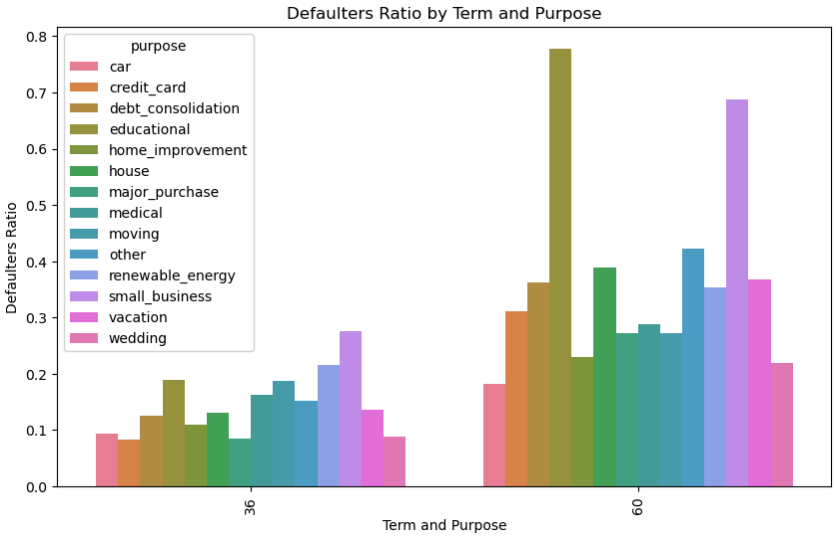
- Loan Amount and Annual Income (We can observe a trend that as annual income increases the defaulters ratio decreases regardless of the loan amount)



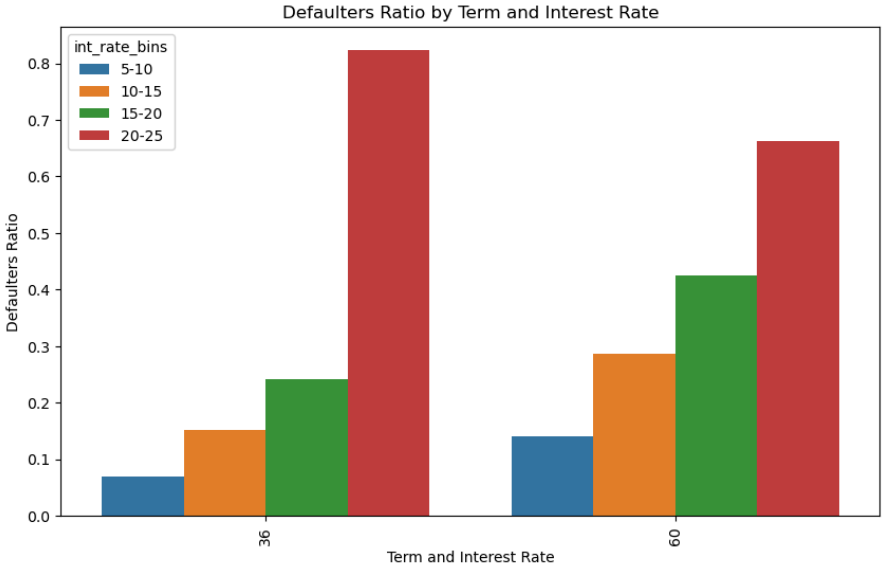
- Employment Length and Loan Amount (We do not observe any relation between employment length and Loan Amount with respect to defaulters ratio)



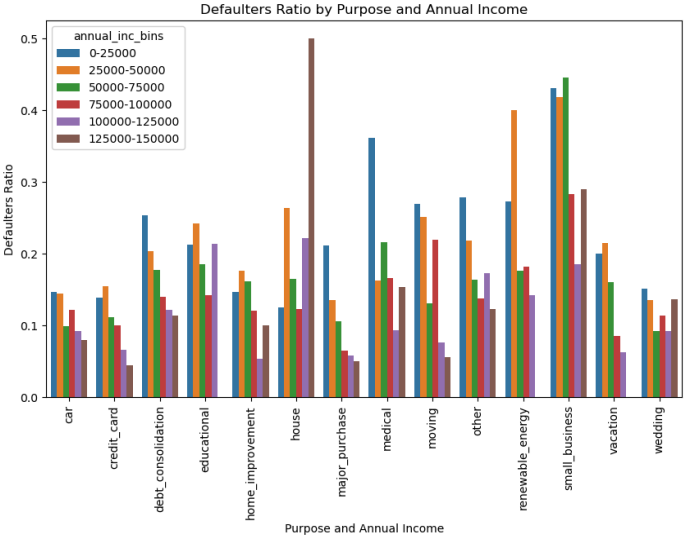
- Term and Purpose (We observe high defaulters ratio for purposes like "small_business" regardless of term.)



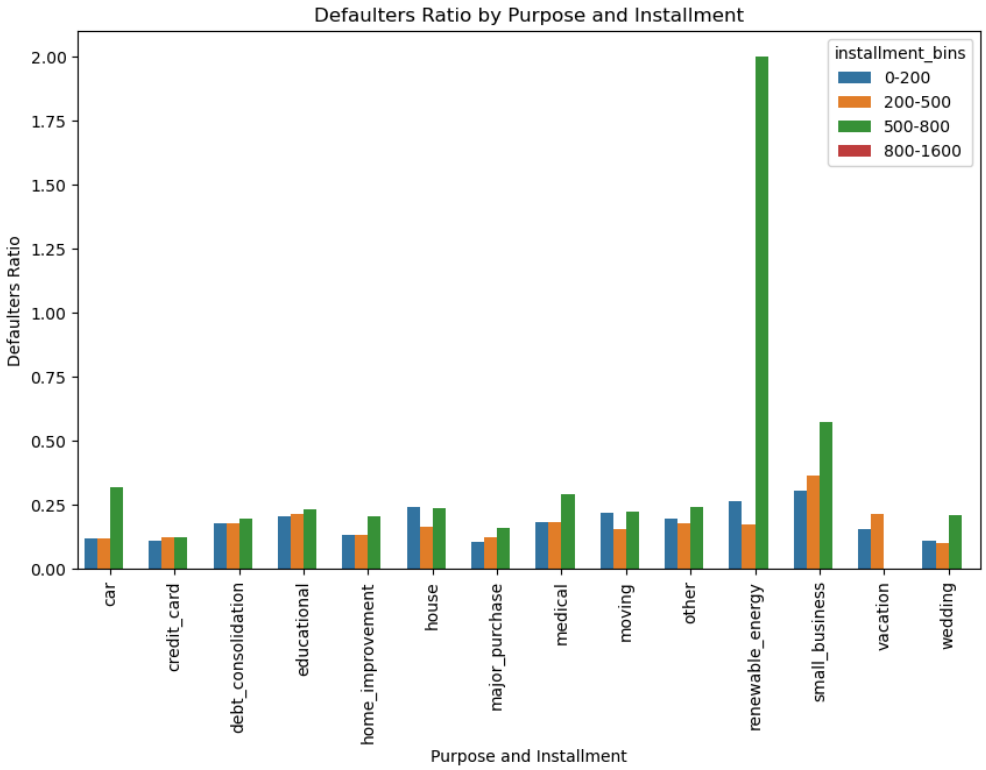
- Term and Interest Rate for loan status (We observe high defaulters ratio as the interest rate increases regardless of the term)



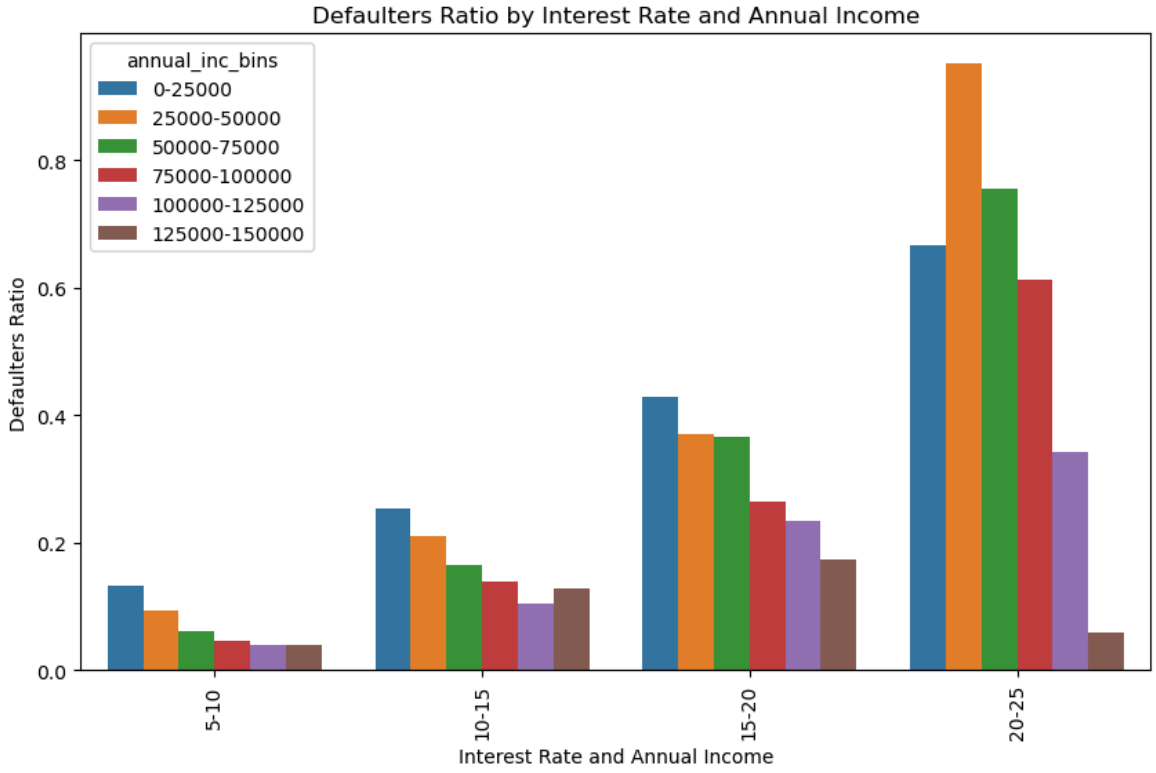
- Purpose and Annual Income for loan status (We don't observe any corellation between purpose and annual income wrt defaulters. Each purpose has different defaulters of different annual income bins)



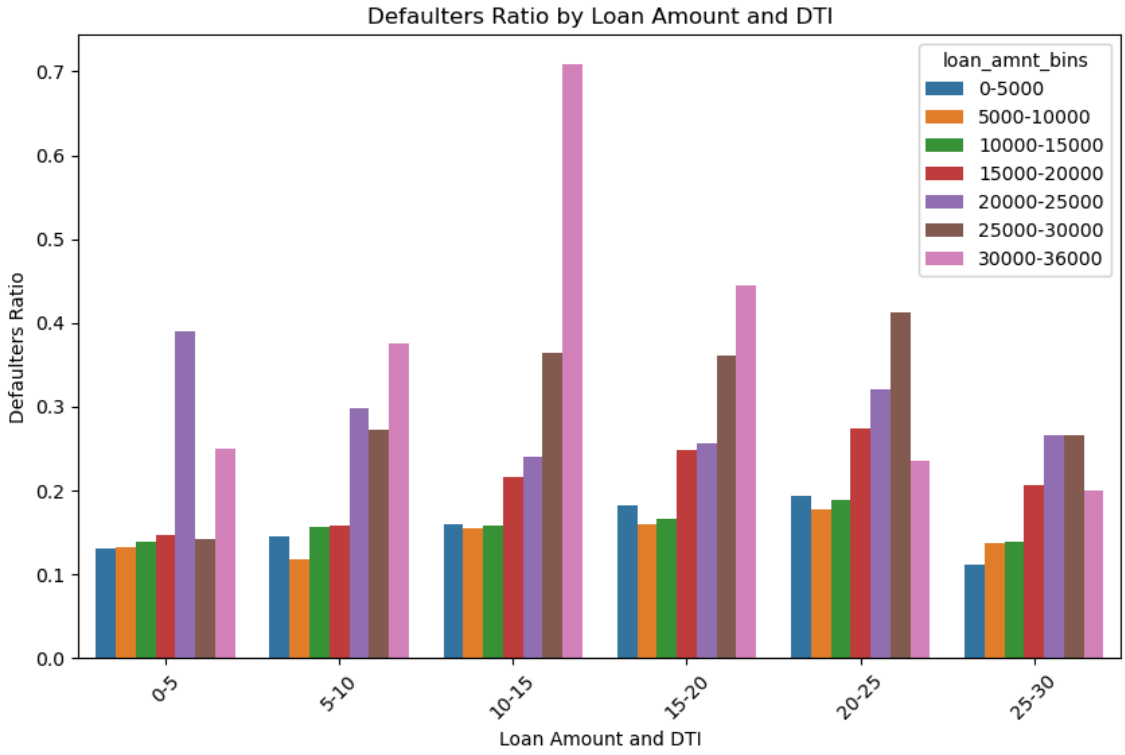
- Purpose and Installment for loan status (Higher the installment -> higher the defaulters for most of the purpose except "small business")



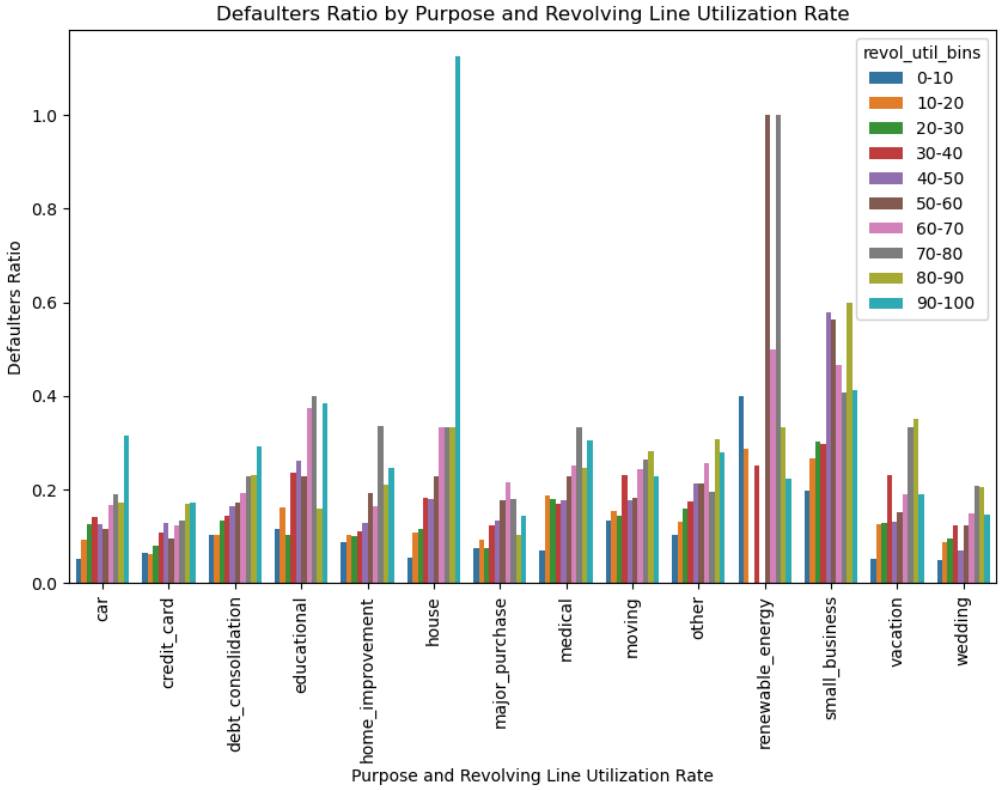
- Interest Rate and Annual Income for loan status (We observe as interest rate increases the defaulters increase and usually as annual income increases defaulters decrease and that is true for each interest rate bin)



- Loan Amount and dti for loan status (We don't observe increase in defaulters in each loan_amount_bins as dti increases)



- revol_util and purpose for loan status (For each purpose, generally as revol_util increases the defaulters also increase)

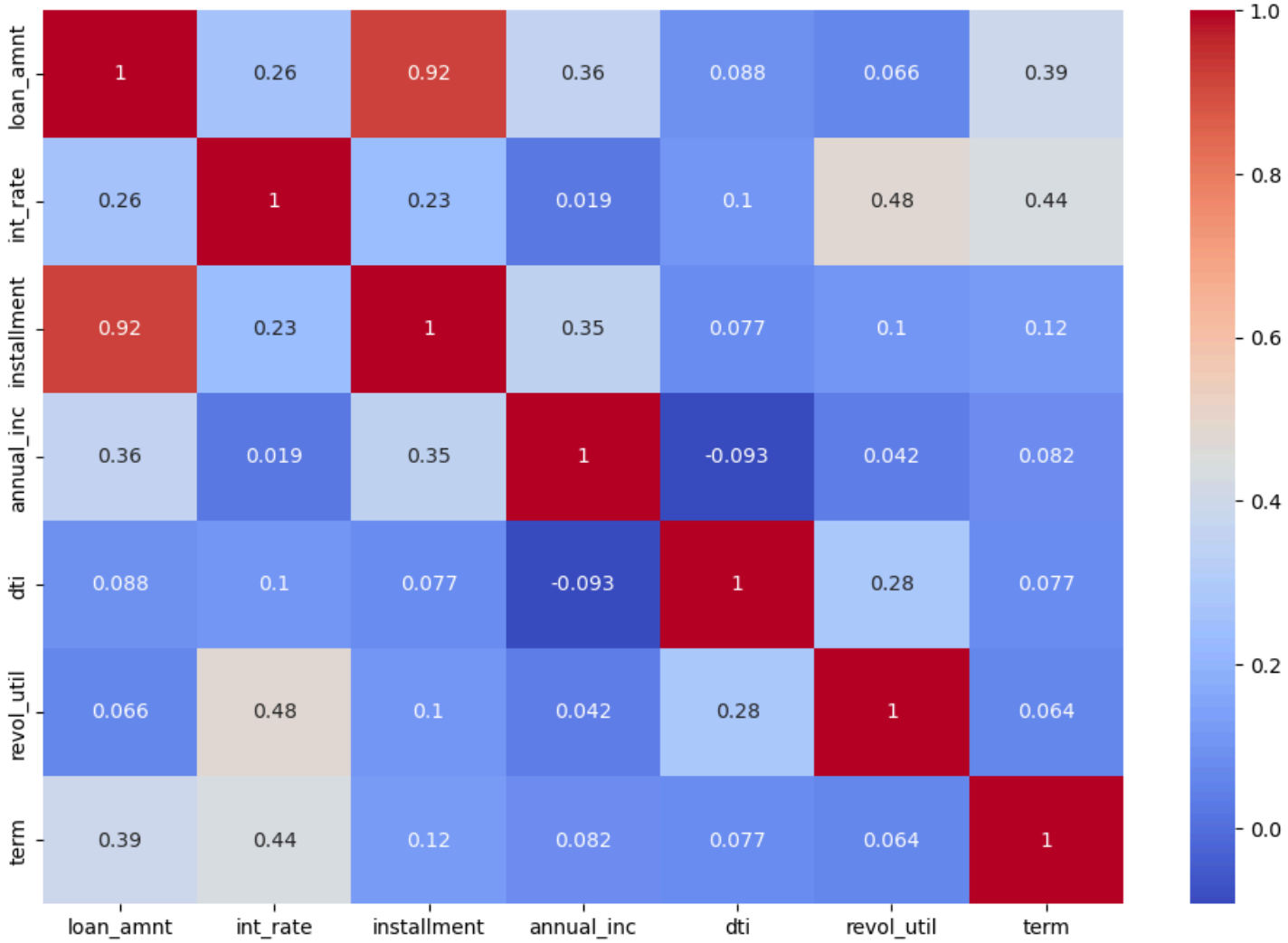


Conclusion

Therefore from the above bivariate analysis, we deduced that the following are the important features we should look out for when giving out loans. As they have an impact on defaulters numbers:

- Loan Amount
- Term
- Interest Rate
- Annual Income
- Purpose = Small Business
- Installment

Multi-Variate Analysis



We see high correlation between

- loan amount and installment

Final Conclusion

After analyzing all available parameters of the dataset, we can conclude on the main driving features for the defaulters in Lending Club Case Study:

- Loan Amount
- Installment
- Term
- Interest Rate
- revol_utl
- Annual income
- Purpose = Small Business
- Grade