Loan Default Case Study

AMEET KUMAR

Problem Statement

A consumer finance company specializes in lending various types of loans to urban customers. When the company receives a loan application, the company has to make a decision for loan approval based on the applicant's profile.

Two types of risks are associated with the bank's decision:

•If the applicant is likely to repay the loan, then not approving the loan results in a loss of business to the company
•If the applicant is not likely to repay the loan, i.e. he/she is likely to default, then approving the loan may lead to a financial loss for the company
Analysis has to be conducted summarizing the risk assessment the bank could carry out to reduce the risk of loan default.

Approach

1. Understand the data using the dictionary

2. Identify data issues

3. Clean data and create derived fields, bins, etc.

4. Find outliers and take appropriate action

5. Perform correlation exercise if required

6. Analyze columns for its effect on bad loans

Preparation for Analysis

Categorical

- addr_state
- dti - make bins
- emp_length
- grade, sub_grade
- home_ownership
- open_acc, total_acc - check correlation
- pub_rec, pub_rec_bankruptcies - check correlation
- delinq_2yrs
- purpose
- verification_status

Contiguous

- annual_inc - make bins
- loan_amnt - make bins

PREP - DERIVED METRICS

1. loan_status_count (Based on loan_status) := 1 if charged-off, 0 otherwise
2. annual_inc_bin = 0 to 25000, 25001 to 50000, 50001 to 75000 and > 75000
3. loan_amnt = 0 to 5000, 5001 to 8000, 8001 to 12000, 12001 to 23000 and >23000

1. Columns were categorized as ‘categorical’ or ‘contiguous’
2. Bins were created for few columns
3. Columns that needed correlation check were identified

Few Observations from Data Analysis

Lot of columns were null

```
# listing null value counts of each column

df1=df0.isnull().sum()
print(df1.values)

```

```
[    0     0     0     0     0     0     0     0     0     0  2459  1075
     0     0     0     0     0     0     0 12940     0    11     0     0
     0     0     0     0 25682 36931     0     0     0    50     0     0
     0     0     0     0     0     0     0     0     0    71     0 38577
     2    56 39717     0     0 39717 39717 39717     0 39717 39717 39717
 39717 39717 39717 39717 39717 39717 39717 39717 39717 39717 39717 39717
 39717 39717 39717 39717 39717 39717    56     0 39717 39717 39717 39717
 39717 39717 39717 39717 39717 39717 39717 39717 39717 39717 39717 39717
 39717 39717 39717 39717 39717 39717 39717 39717 39717   697    39 39717
 39717 39717 39717]
```

Many columns have lot of null values. There's one column with 2459 null values but that's about 6%. So let's delete columns with null values more than that

Correlation was performed

```

pub_corr = df0_cf['pub_rec'].corr(df0_cf['pub_rec_bankruptcies'])
print(pub_corr)
# Seems strongly correlated. Will use  'pub_rec_bankruptcies'
```

```
0.8585914697282653
```

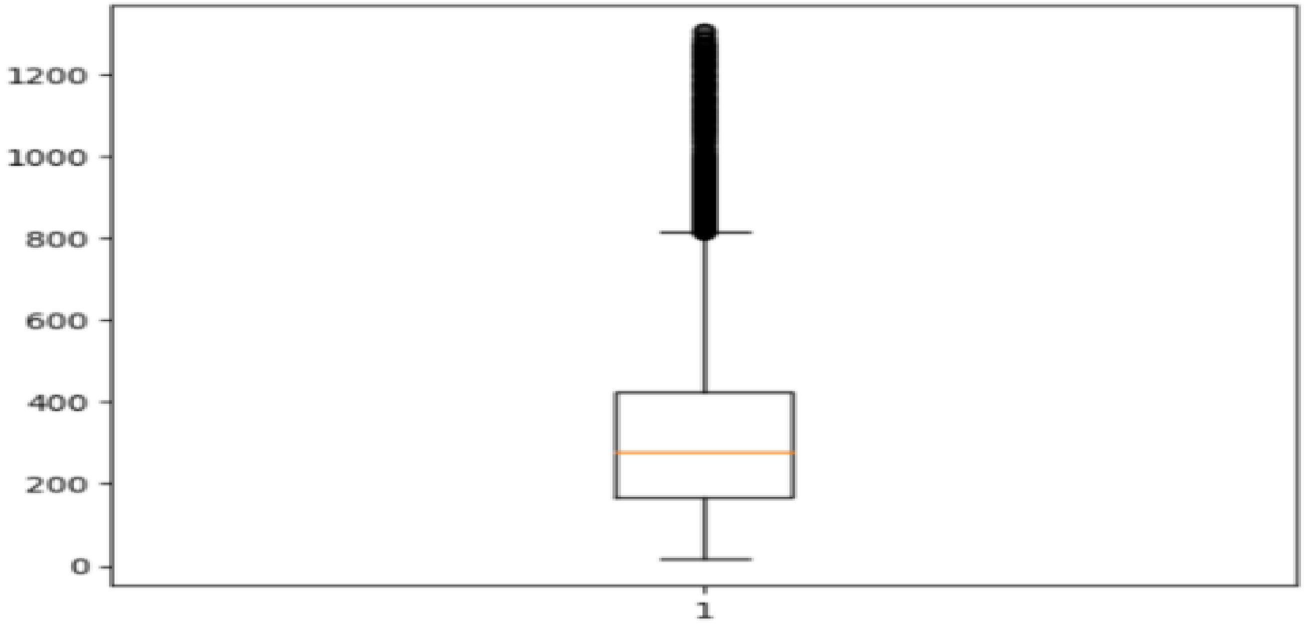
Column values were analyzed

```
df0['term'].value_counts()
df0['int_rate'].value_counts()
df0['grade'].value_counts()

df0['sub_grade'].value_counts()
df0['emp_title'].value_counts()
df0['emp_length'].value_counts()

df0['home_ownership'].value_counts() # There are few entries that don't make sense. We delete such rows
df0.drop(df0[df0['home_ownership'].isin(['OTHER','NONE'])].index, inplace=True)

df0['verification_status'].value_counts()
df0['issue_d'].value_counts()
df0['loan_status'].value_counts() # The analysis will focus on paid and charged-off. Hence delete others
df0.drop(df0[df0['loan_status'].isin(['Current'])].index, inplace=True)

df0['pymnt_plan'].value_counts()  # This has single value. Hence drop this column
df0.drop(['pymnt_plan'], axis=1, inplace=True)

df0['purpose'].value_counts()
df0['title'].value_counts()
df0['dti'].value_counts()

df0['earliest_cr_line'].value_counts()
df0['revol_util'].value_counts()

df0['initial_list_status'].value_counts() # This has single value. Hence drop this column
df0.drop(['initial_list_status'], axis=1, inplace=True)

df0['last_pymnt_d'].value_counts()
df0['last_credit_pull_d'].value_counts()

df0['application_type'].value_counts() # This has single value. Hence drop this column
df0.drop(['application_type'], axis=1, inplace=True)
```

Few Observations from Data Analysis Ctd.

Outliers were identified

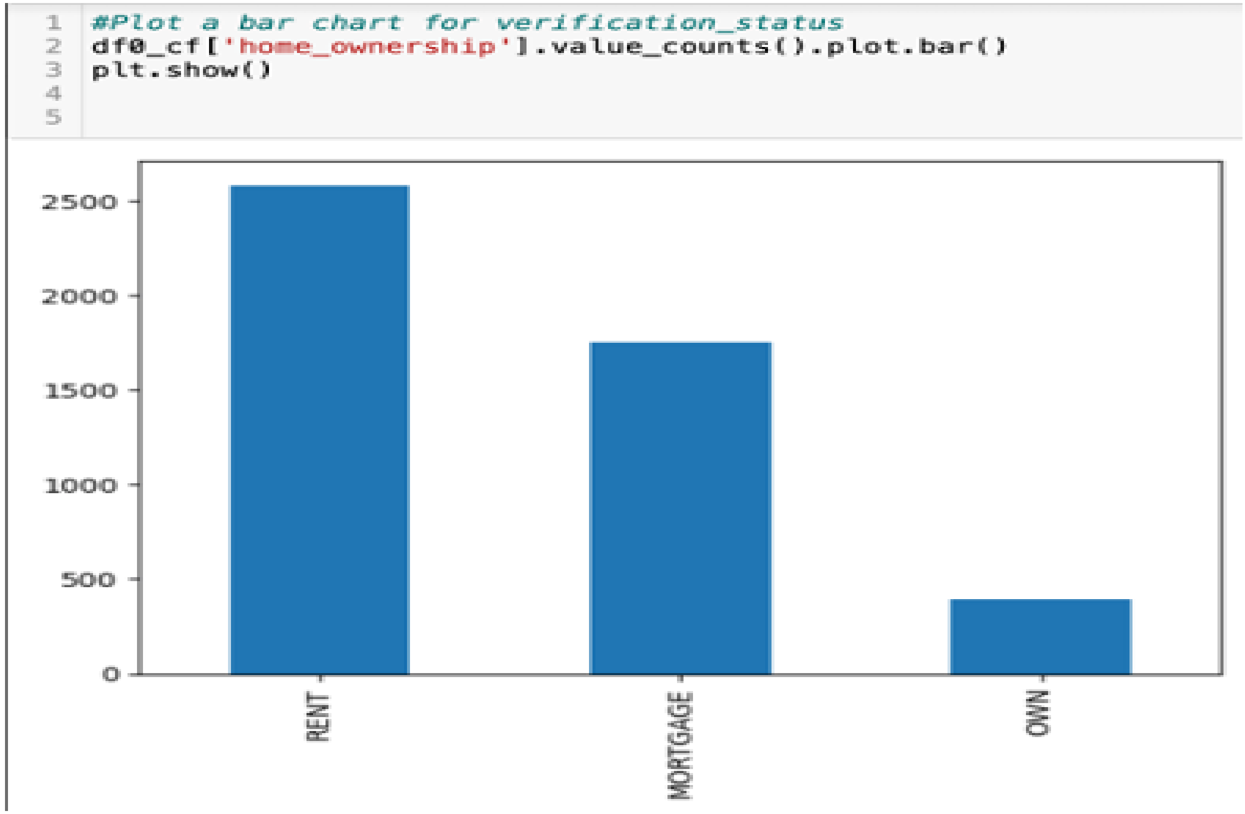
```
#Create a box plot for the installment column
plt.boxplot(df0['installment'])
plt.show()

#df0['installment'].describe()

```



Looks like there are lot of outliers. So we delete rows where installment > 1000

Few Observations of Analysis

People living in rented or mortgaged property tend to default more



Few Observations of Analysis Ctd.

Does purpose of loan offer important insight?

```
#Plot a bar chart for purpose
df0_cf['purpose'].value_counts()
```

```
debt_consolidation    2353
other                  528
credit_card            442
small_business         378
home_improvement       271
major_purchase         202
car                    139
medical                 94
moving                  83
wedding                 81
vacation                49
educational             46
house                   45
renewable_energy        17
Name: purpose, dtype: int64
```

debt_consolidation, other, credit_card, small_business, home_improvement, major_purchase and car are focus areas. Particularly - debt_consolidation, other, credit_card and small_business

Conclusion

- Grades B, C and D are more likely to default

- People who own home don't tend to default as mush as others do. But others should be analyzed

- Loans taken for 'debt_consolidation', 'other', 'credit_card' and 'small_business' are more likely to default

- People who have been working for about 5 years or more than 10 years tend to default more on loan. Especially 10+

- People who have 4 to 11 credit lines are more likely to default

- People living in state code CA are more likely to default