Risk Analysis Case Study

[Consumer Finance Company]

Business Understanding

- 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

Case Statement

In this case study, you will use EDA to understand how **consumer attributes** and **loan attributes** influence the tendency of default.

Data Cleaning and Manipulations

Initially the total columns were 111

After removing all the columns with all values with NAN, total no of columns came down to 57

After that found out the total percentage of null values in columns having NAN, then fixed the threshold to 30 and removed all the columns having greater that 30% null values.

For rest with less than 30% null values, replaced the missing values by calculations as in next slide.

Replacing Missing values

1. Emp_title: This column had around 6% null values. Since this is a categorical variable, I calculated the mode and replaces the missing values with mode - US Army
2. title: This column had around title 0.03 % null values. Since this is a categorical variable, I calculated the mode and replaces the missing values with mode - Debt Consolidation
3. revol_util:This column had missing values of 0.13%, since its numerical value and value id too small percentage, replacing it with 0.0
4. last_pymnt_d: Since this is a date column and number is too small, replacing this with mode
5. collections_12_mths_ex_med, chargeoff_within_12_mths, pub_rec_bankruptcies, tax_liens – Since these are numerical values and percentage of nulls are too low, replacing all the NAN values with 0.0
6. emp_length: replacing this categorical variable with mode

Replacing the Types of columns for univariate analysis

- Getting all the type details with loans_df.info(verbose=True)
- Replacing int_rate to float using > loans_df['int_rate'].str.rstrip("%").astype(float)
- Parsing and using only year for analysis for issued_year column using loans_df["issued_year"] = loans_df["issue_d"].str[4:] and loans_df["issued_year"].astype(int)

Univariate Analysis

First we need to create the separate dataframes for analysis.

For this we consider loan_status column, one df with chargedOff and df column with FullyPaid for analysis.

Default/Non-Default analysis

```
100*(loans_df.loan_status.value_counts())/(len(loans_df))
```

By using above calculations> result was,

Fully Paid 82.961956

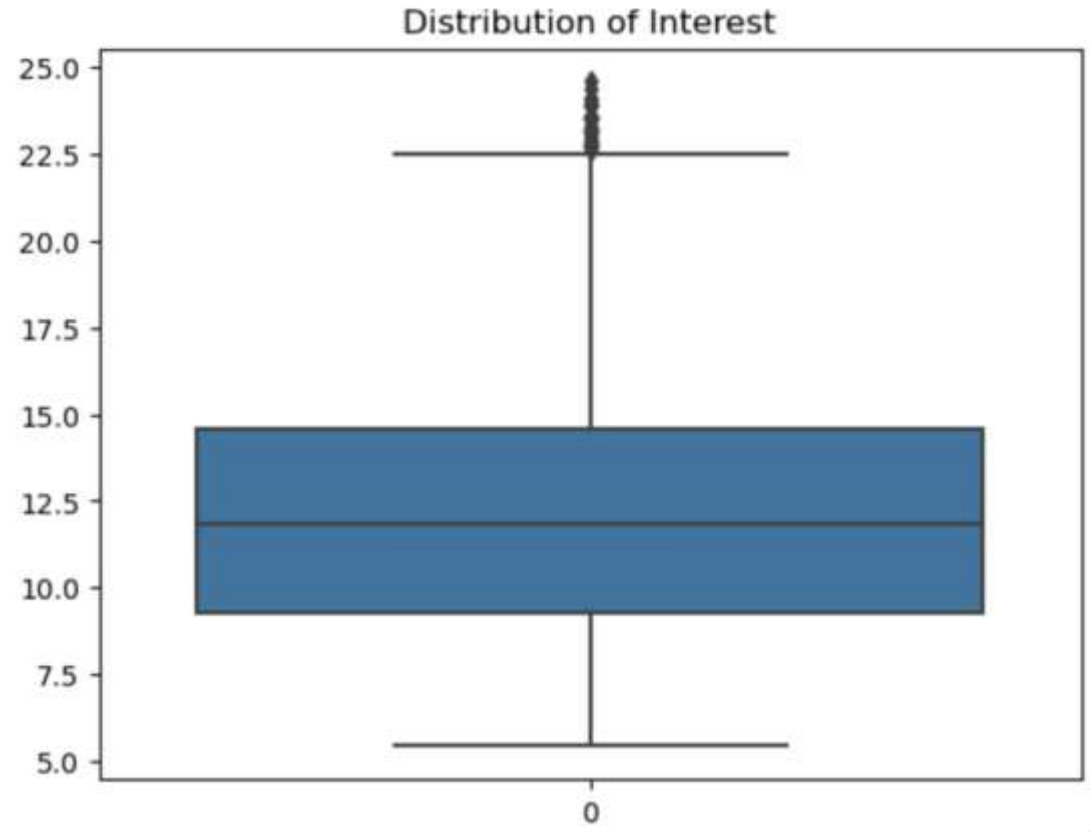
Charged Off 14.167737

Current 2.870307

Around 86% are non-default and around 14% is default

Univariate Analysis – Int_Rate

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.boxplot(loans_df.int_rate)
plt.title('Distribution of Interest')
plt.show()
```



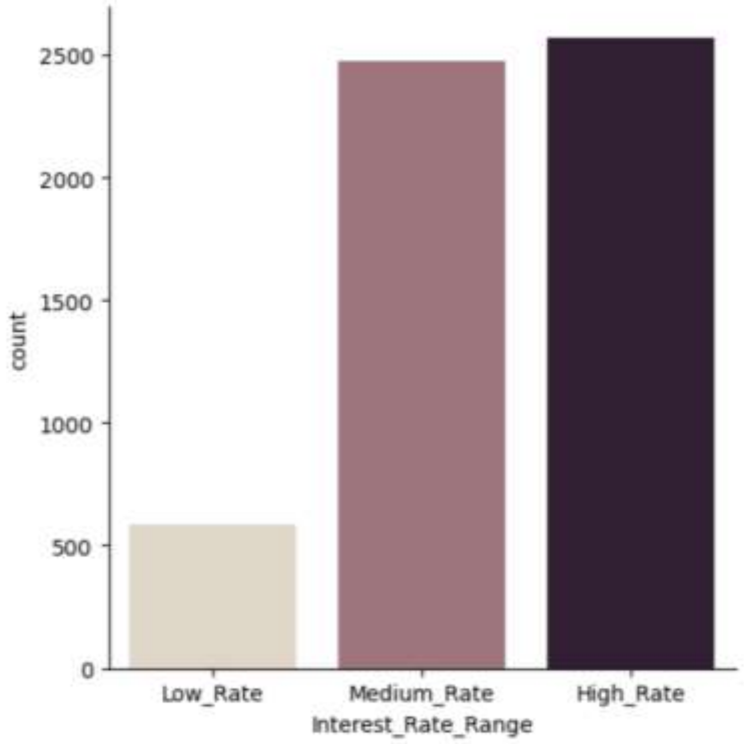
- Plot shows the distribution of interest
- Bin the interest rates to low, medium and high and
- doing univariate analysis

Univariate analysis - Int_Rate

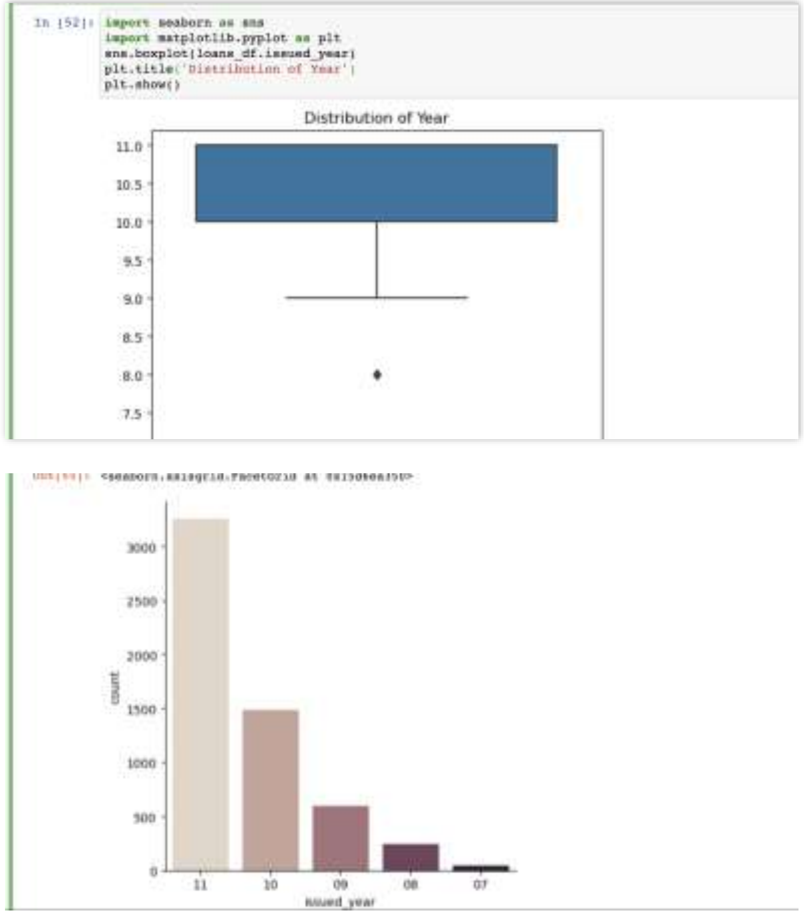
One with higher interest range is to default more

```
sns.catplot(data=charged_off, x="Interest_Rate_Range", kind="count", palette="ch:.25")
```

<seaborn.axisgrid.FacetGrid at 0x1604c53d0>



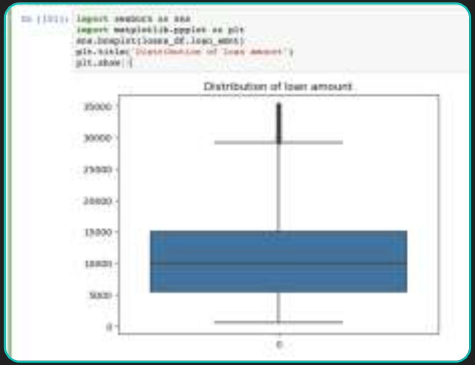
Univariate Anlaysis - issue_d

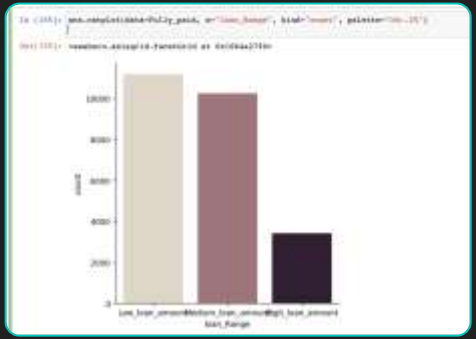


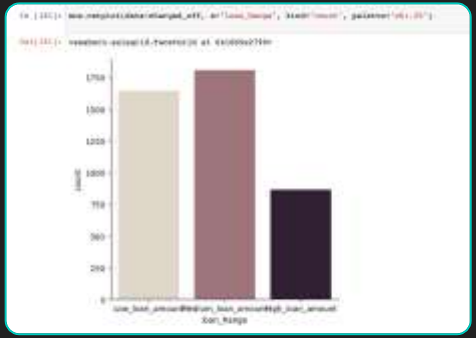
- Compared to year on year,
- every every has increasing rate of defaults

Univariate analysis – laon_amt

- Binning the loan values to low, medium and higher range
- One who falls under medium loan range is said to
- default the most
- One with low loan amount has paid fully.

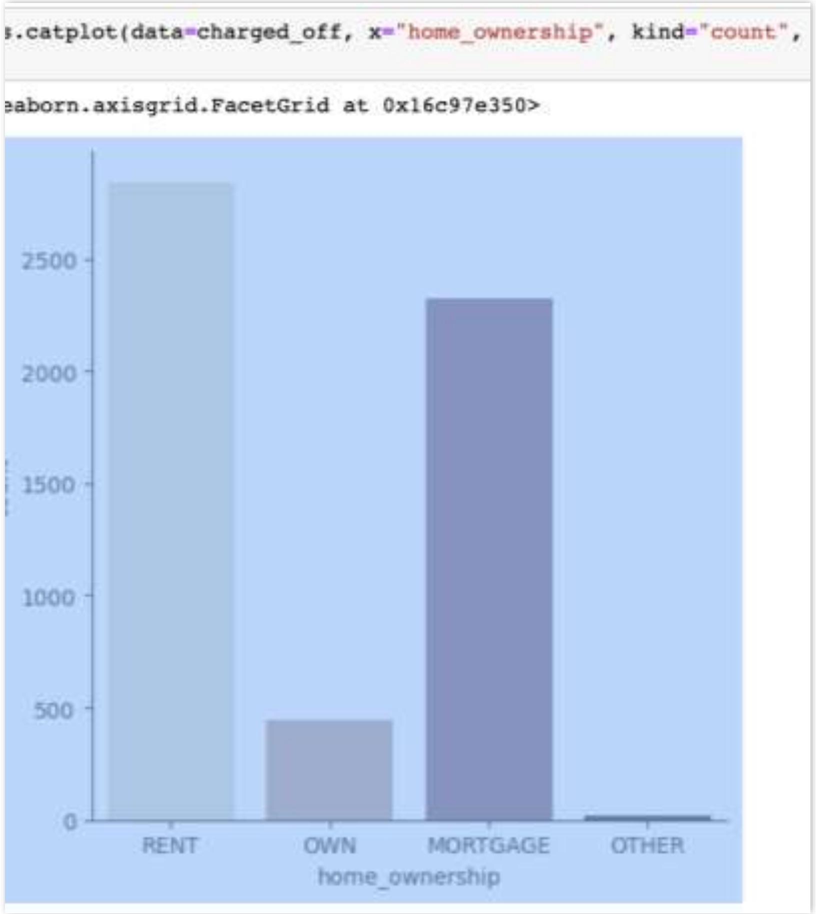




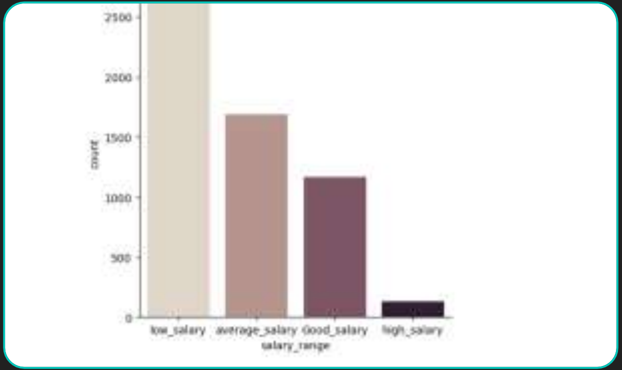


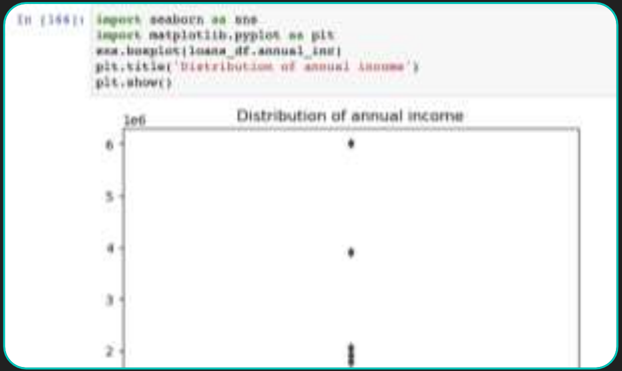
Univariate Analysis – Home_ownership

- One who are in rent/mortage are to
- default the most



Univariate Analysis – Salary range

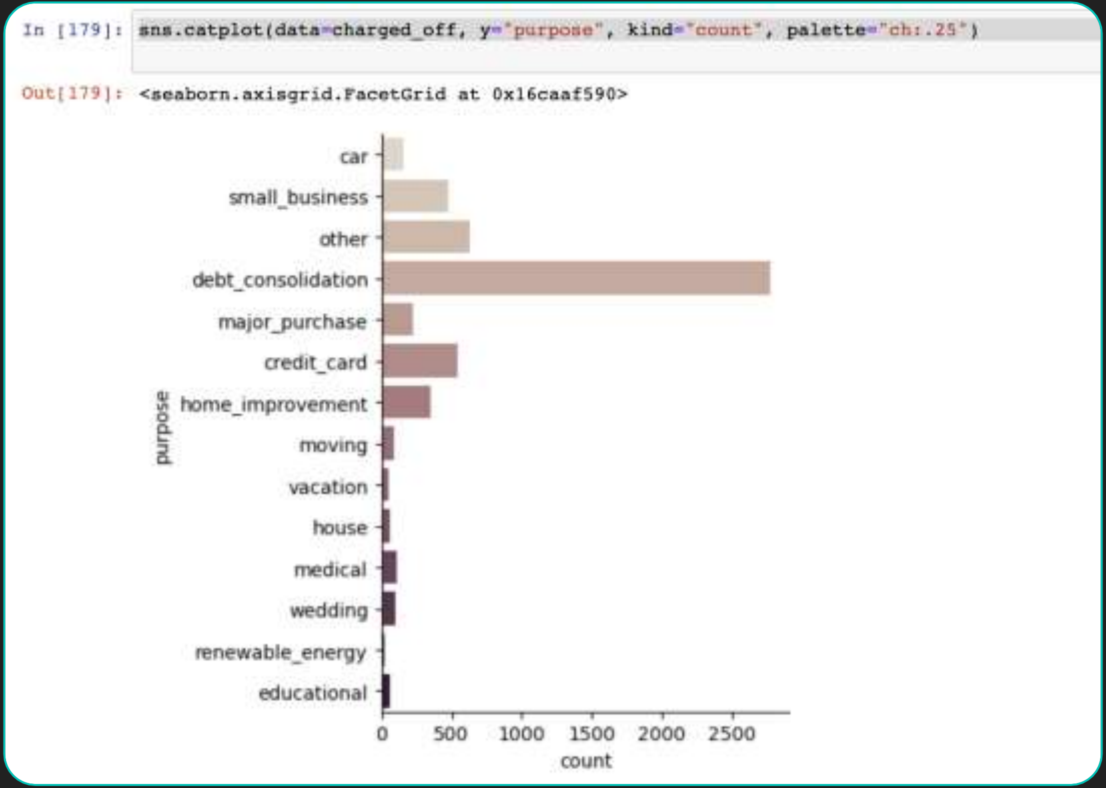




- One with low salary is to default the
- most compared to one with high salary

Univariate Analysis - purpose

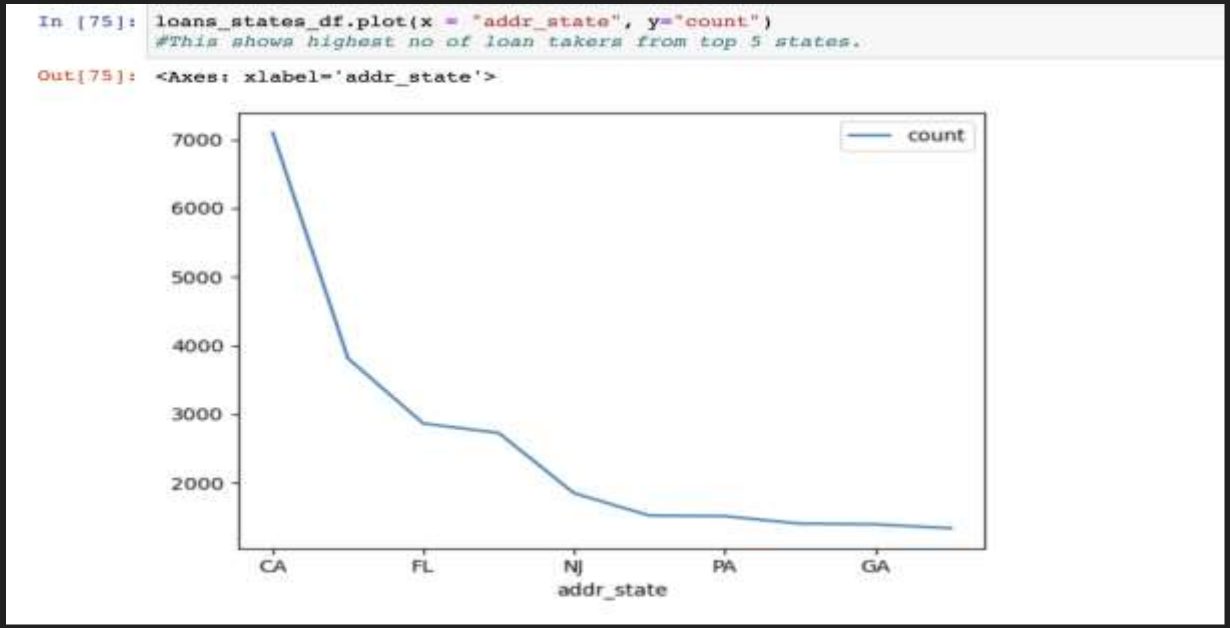
- One who has taken loan for purpose of
- debt consolidation is to default the most
- compared to others



Segmented Univariate – addr_state

Grouping by state Grouping by state and checking which top 5 states has taken most loans

As we can see, California, Florida, New Jersey, Pennsylvania, Georgia has taken more loans



Bivariate analysis – default correlation

As we can see, the top 10 highly correlated variables for defaults

	VAR1	VAR2	Correlation
32	member_id	id	0.99
98	funded_amnt	loan_amnt	0.98
593	total_pymnt_inv	total_pymnt	0.97
195	installment	funded_amnt	0.95
194	installment	loan_amnt	0.93
131	funded_amnt_inv	funded_amnt	0.93
130	funded_amnt_inv	loan_amnt	0.91
625	total_rec_prncp	total_pymnt	0.91
657	total_rec_int	total_pymnt	0.90
658	total_rec_int	total_pymnt_inv	0.89

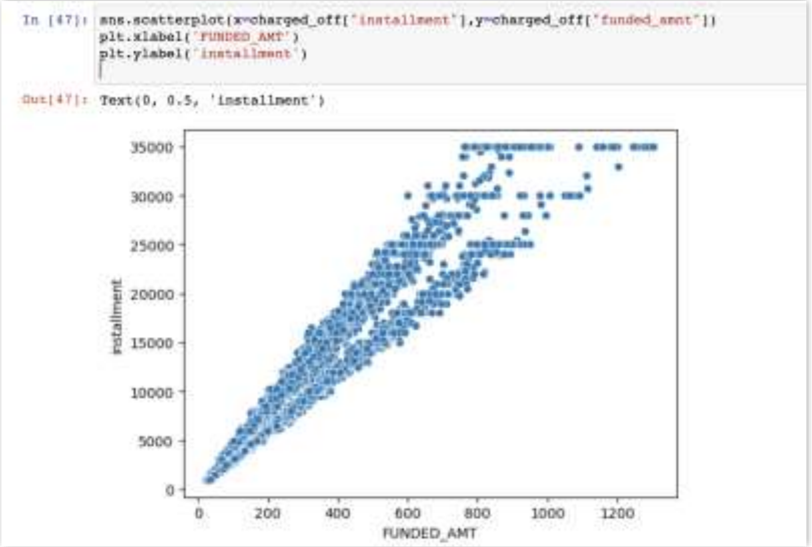
Bivariate Analysis – Non-defaults correlation

As we can see, the top 10 highly correlated variables for Non - Defaults

	VAR1	VAR2	Correlation
611	total_rec_prncp	funded_amnt	1.00
32	member_id	id	0.99
610	total_rec_prncp	loan_amnt	0.98
580	total_pymnt_inv	funded_amnt_inv	0.98
625	total_rec_prncp	total_pymnt	0.98
547	total_pymnt	funded_amnt	0.98
98	funded_amnt	loan_amnt	0.98
593	total_pymnt_inv	total_pymnt	0.97
546	total_pymnt	loan_amnt	0.97
131	funded_amnt_inv	funded_amnt	0.96

Example of correlation b/w installment and funded amount

- As you can see, the graph is almost linear showing
- that both are highly correlated



Bivariate analysis - Conclusions

- Correlation with defaults and non defaults is almost the same, with slight differences.

Thank You