CONFIDENTIAL & RESTRICTED

LENDING CLUB CASE STUDY

SUBMISSION

Group Name:

1. Koushik Kumar
2. Payal Mandulkar
3. Sindhu Mohana
4. Sourav Dutta

Lending Club Case Study

Business Understanding

- This case study is for a consumer finance company specializing in lending various types of loans to urban customers. Two types of loan applicants are of interest :
  - ❖ Likely to repay the loan, but not approving the loan, results in a loss of business to the company
  - ❖ Not likely to repay the loan, but approving the loan may lead to a financial loss to the company
- Decisions taken by Lending club when a loan application is received
  - ❖ Accept the loan - when the individual meets the criteria
    - ❖ Fully Paid - All the principal & interest are fully paid
    - ❖ Current - Ongoing Loan
    - ❖ Charged Off – Installment payment missed for a long time

Business Objectives

Business Goal :

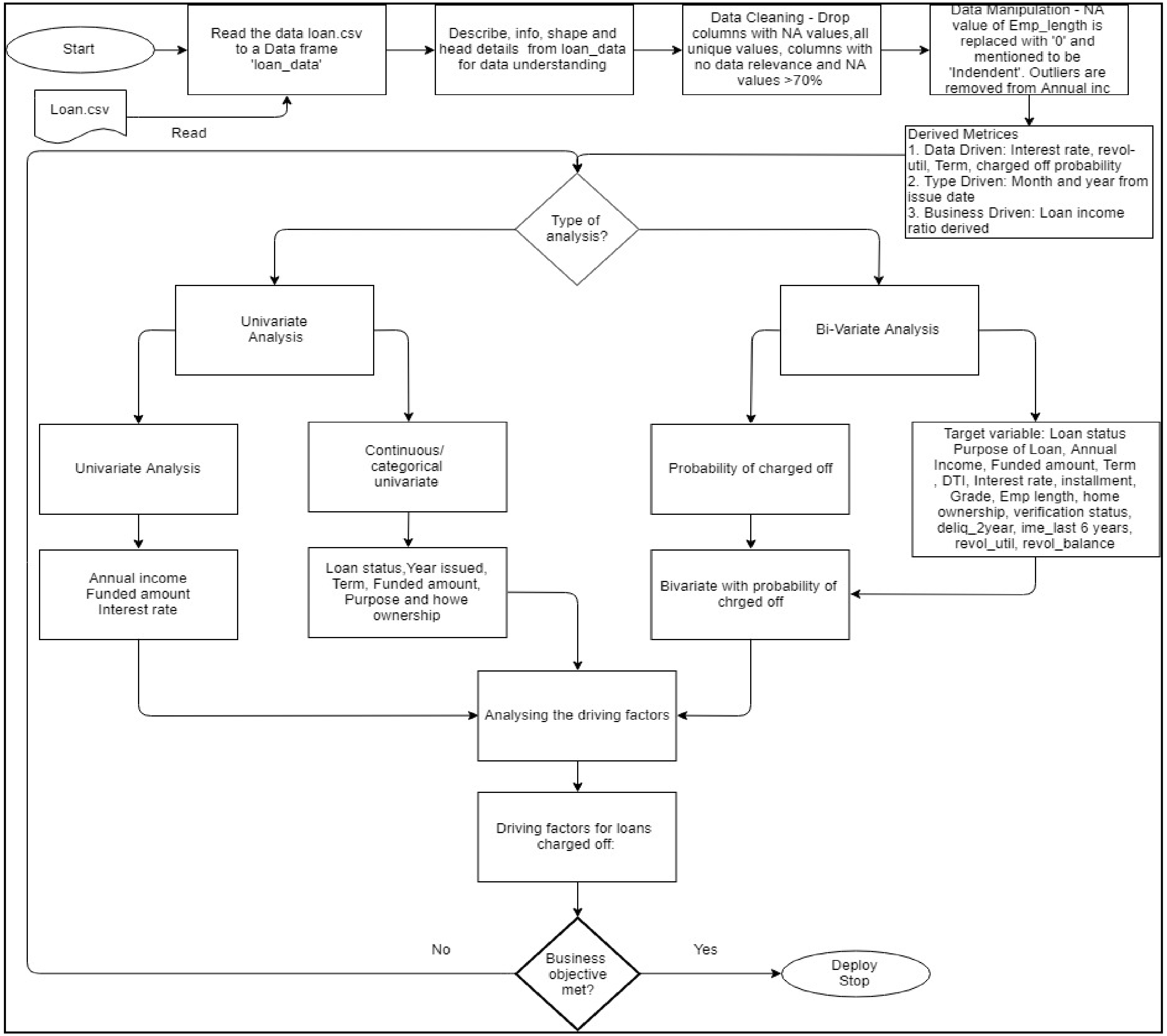
- Increase chances of Profit – Lending money to applicants likely to repay the loan
- Control the Financial loss in case of default
  - Cut down on the loans associated with risky applicants who default on loan
  - Utilize the knowledge of such loan applicants for long term portfolio and risk assessment.

Goal of the Exploratory Data Analysis

Analysis Goal :

- Understand how **consumer attributes** and **loan attributes** influence the tendency of default.
- Derive the strong indicators to loan default by performing EDA on historical data of loan applicants

Problem solving approach



Data Understanding

1. Data Understanding

Step 1: Reading the file : loan.csv has 39717 rows and 111 columns .

Summary :

1. On the first look we can see multiple categorical and numerical columns
2. 54 columns with all values null / NAN
3. Both Categorical (type = object) & Numerical columns are present (type = float64 , int64)

```
loan_data_Df = pd.read_csv('C:\ML & AI\Cohort\Lending Club Case Study\loan.csv', low_memory = False)

print(loan_data_Df.shape)

loan_data_Df.info(verbose = True, null_counts = True)

print("We have " + str((loan_data_Df.isnull().sum() == 39717).sum()) + " columns where all rows all null and these can be dropped
```

```
(39717, 111)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39717 entries, 0 to 39716
Data columns (total 111 columns):
id                             39717 non-null int64
member_id                      39717 non-null int64
loan_amnt                      39717 non-null int64
```

```
print("We have " + str((loan_data_Df.isnull().sum() == 39717).sum()) + " columns where all rows all null and these can be dropped
```

```
num_op_rev_tl                  0 non-null float64
num_rev_accts                  0 non-null float64
num_rev_tl_bal_gt_0            0 non-null float64
num_sats                       0 non-null float64
num_tl_120dpd_2m               0 non-null float64
num_tl_30dpd                   0 non-null float64
num_tl_90g_dpd_24m             0 non-null float64
num_tl_op_past_12m             0 non-null float64
pct_tl_nvr_dlq                 0 non-null float64
percent_bc_gt_75               0 non-null float64
pub_rec_bankruptcies           39020 non-null float64
tax_liens                      39678 non-null float64
tot_hi_cred_lim                0 non-null float64
total_bal_ex_mort              0 non-null float64
total_bc_limit                 0 non-null float64
total_il_high_credit_limit     0 non-null float64
dtypes: float64(74), int64(13), object(24)
memory usage: 33.6+ MB
We have 54 columns where all rows all null and these can be dropped
```

Observations

In this section, we have tried to read the "loan_csv" at the beginning. Some of the observations that came to light were –

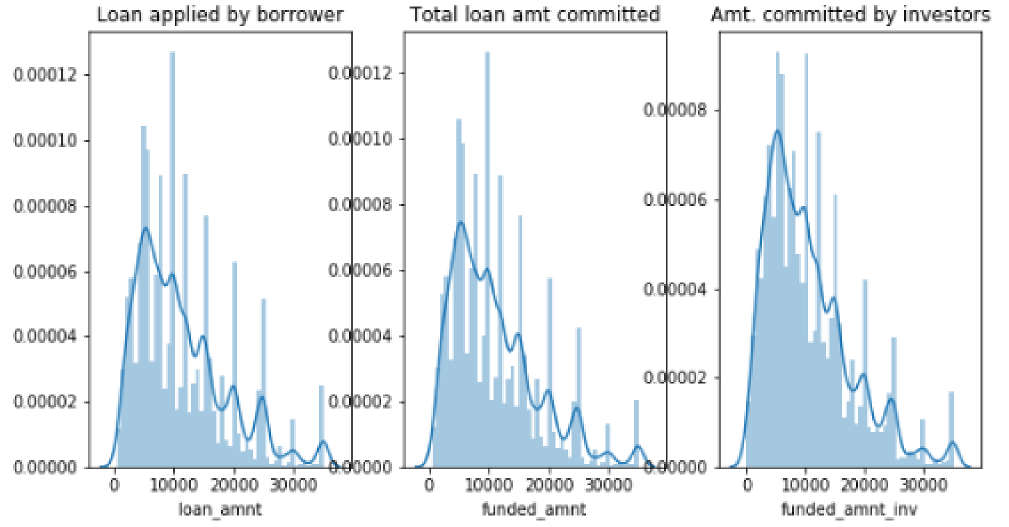
- There are multiple categorical & numerical columns
- The categorical columns had the type as "object" & Numerical had the type as "float64" & "int64" respectively
- Approx. 54 columns were identified were 100% of the values are null / NAN

UpGrad

Data Cleaning & Manipulation

Data Point (Columns)	Rationale for cleaning / manipulation
All NAN	No business significance, hence dropped
Having > 60% Null values	Less business significance as only 3 columns having > 60% Null values (post dropping columns with all NAN)
Containing a constant value	Such columns doesn't impact the analysis
Too textual	Too Textual columns are difficult to analyze and derive a pattern – 1 columns , 'Desc' is dropped
Multicollinear columns	For eg : "sub-grade" was dropped. "grade" is representative of subgrade as well. Dropped to avoid multi-collinearity
With no business significance	zipcode (last digits are xx) , id , member_id , url are dropped
remove '%'	Eg : int_rate , revol_util are treated to remove % and converted to float
Derived columns	1. Issue_year and issue_month are derived from issue_d 2. Columns created for 'bins' a. annual_income_range b. Funded_amount_range c. Int_rate_range d. dti_range 3. installment_ratio
Imputing	emp_length : values with NA are replace 0

	loan_amnt	funded_amnt	funded_amnt_inv
loan_amnt	1.000000	0.981578	0.940034
funded_amnt	0.981578	1.000000	0.958422
funded_amnt_inv	0.940034	0.958422	1.000000



Summary of Data Cleaning

Metric	Before Cleaning	After Data Cleaning
Numerical	87	22
Categorical	24	13
Derived Metrics	0	6

Univariate Analysis & Segmented Univariate Analysis

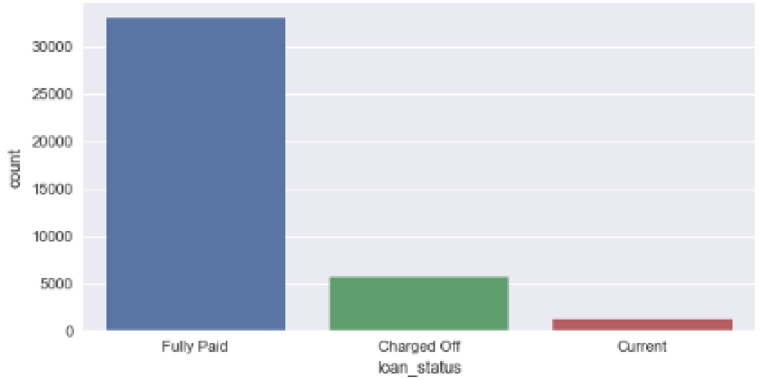
Univariate Analysis 1. Loan status

```
# Univariate Analysis : Loan status
#*******************************************
loan_data_Df['loan_status_num'] = np.where((loan_data_Df['loan_status'] == 'Charged Off') , 0 , 1)
plt.figure(num=None, figsize=(8, 4), dpi=80, facecolor='w', edgecolor='k')
ax = sns.countplot(x="loan_status", data=loan_data_Df, order = loan_data_Df['loan_status'].value_counts().index)


# Get the percentage of Charged off Loans
perc = round(100*(len(loan_data_Df.loc[(loan_data_Df['loan_status'] == 'Charged Off')])/len(loan_data_Df['loan_status'])),2)

print("percentage of Charged off Loans is ",str(perc))
plt.show()
```

```
percentage of Charged off Loans is  14.17
```

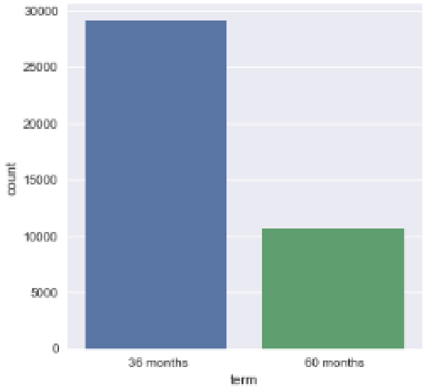


Univariate Analysis #4 : Term

```
plt.figure(figsize=(5, 5))
sns.countplot(loan_data_Df['term'])
plt.show()

# Get the percentage of Loans for 36 month term
perc = round(100*(len(loan_data_Df.loc[(loan_data_Df['term_mth'] == 36)])/len(loan_data_Df)),2)
print("percentage of Loans for 36 months ",str(perc))

# Get the percentage of Loans for 60 month term
perc = round(100*(len(loan_data_Df.loc[(loan_data_Df['term_mth'] == 60)])/len(loan_data_Df)),2)
print("percentage of Loans for 60 months ",str(perc))
```




```
percentage of Loans for 36 months   73.26
percentage of Loans for 60 months   26.74
```

Inference #6: 1. 73% applicants have chosen 36 months as a loan tenure and 27 % applicants have chosen 60 months as tenure

Univariate Analysis #5 : Home Ownership wise Loan

```
loan_data_Df['home_ownership'].unique()
ho=['OTHER','NONE']
loan_data_Df.drop(loan_data_Df[loan_data_Df['home_ownership'].isin(ho)].index,inplace=True)
loan_data_Df.home_ownership.unique()
(((loan_data_Df.groupby('home_ownership').purpose.count())/len(loan_data_Df))*100).sort_values(ascending=False)
sns.countplot(loan_data_Df['home_ownership'])

print("percentage of Loans for home ownership purpose : ", str(((loan_data_Df.groupby('home_ownership').purpose.count())/len(loan
```

```
percentage of Loans for home ownership purpose :  home_ownership
MORTGAGE    44.575424
OWN          7.719103
RENT        47.705473
Name: purpose, dtype: float64
```

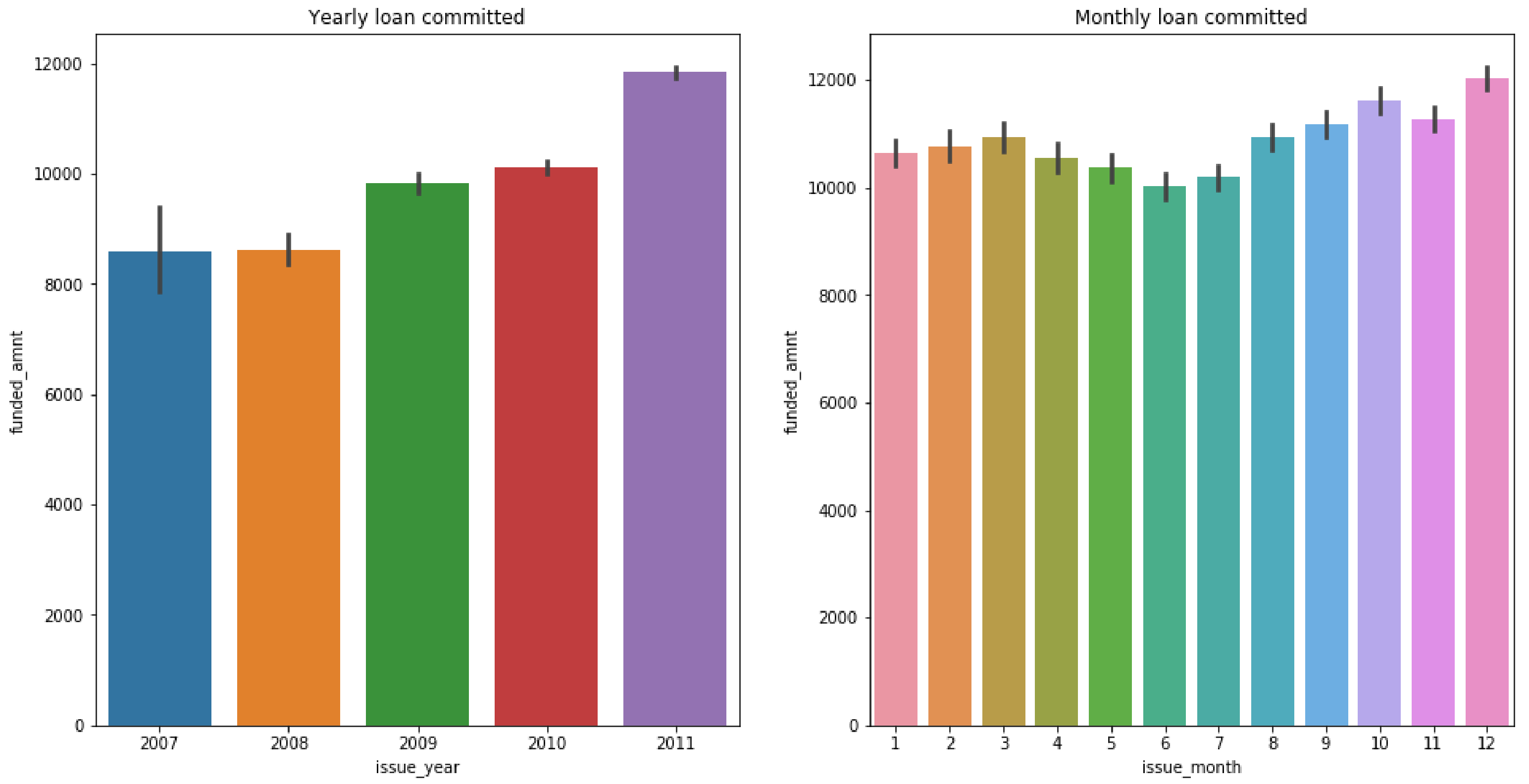
Inference #7: 1. Insights: Maximum no. of loans are taken by the people who are staying at Rent: 48% or Mortgage houses:45%

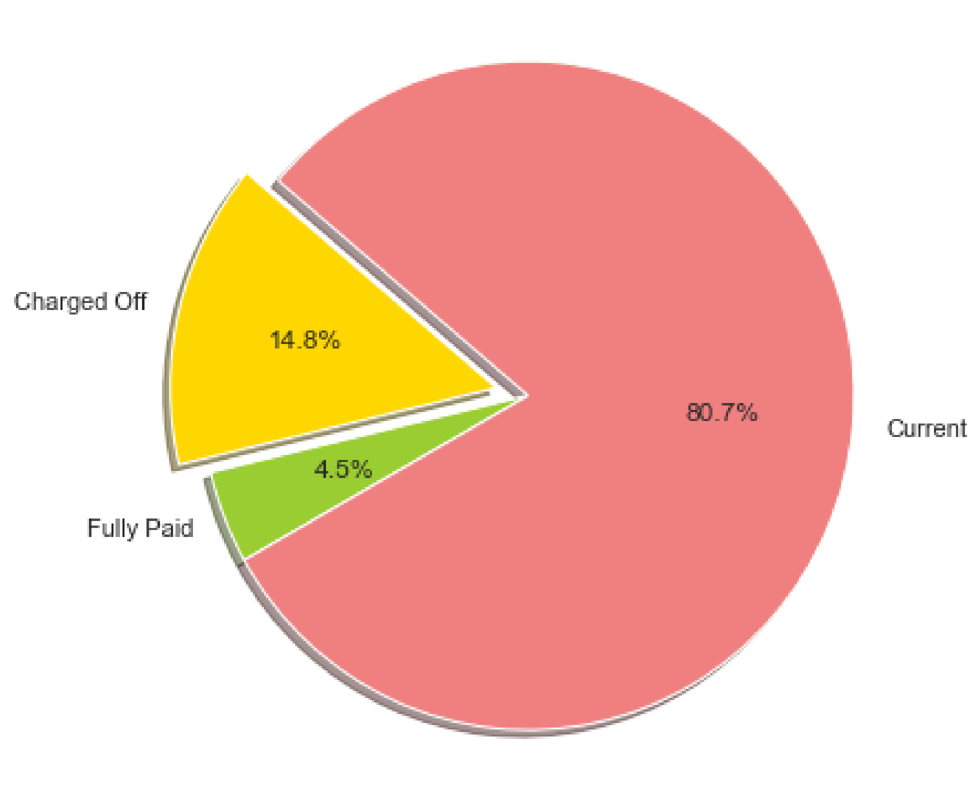
Observations

Some of the inferences from these charts are –

- % of Charged Off customers is 14%
- Range of customers Charged off is up to 5000
- 73% & 27% applicants has chosen 36 & 60 mths as loan tenure respectively
- Max. loans taken by people staying at Rent(48%) or Mortgage houses (45%)

Univariate Analysis & Segmented Univariate Analysis contd..

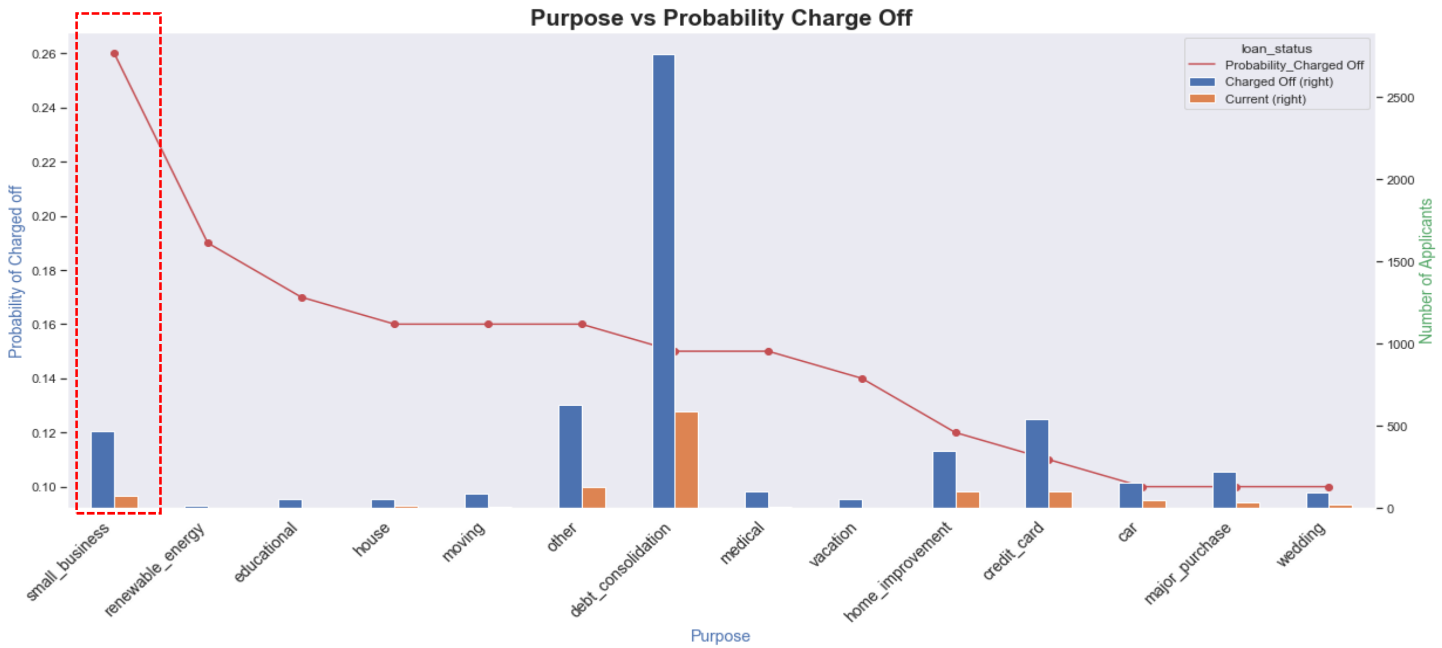


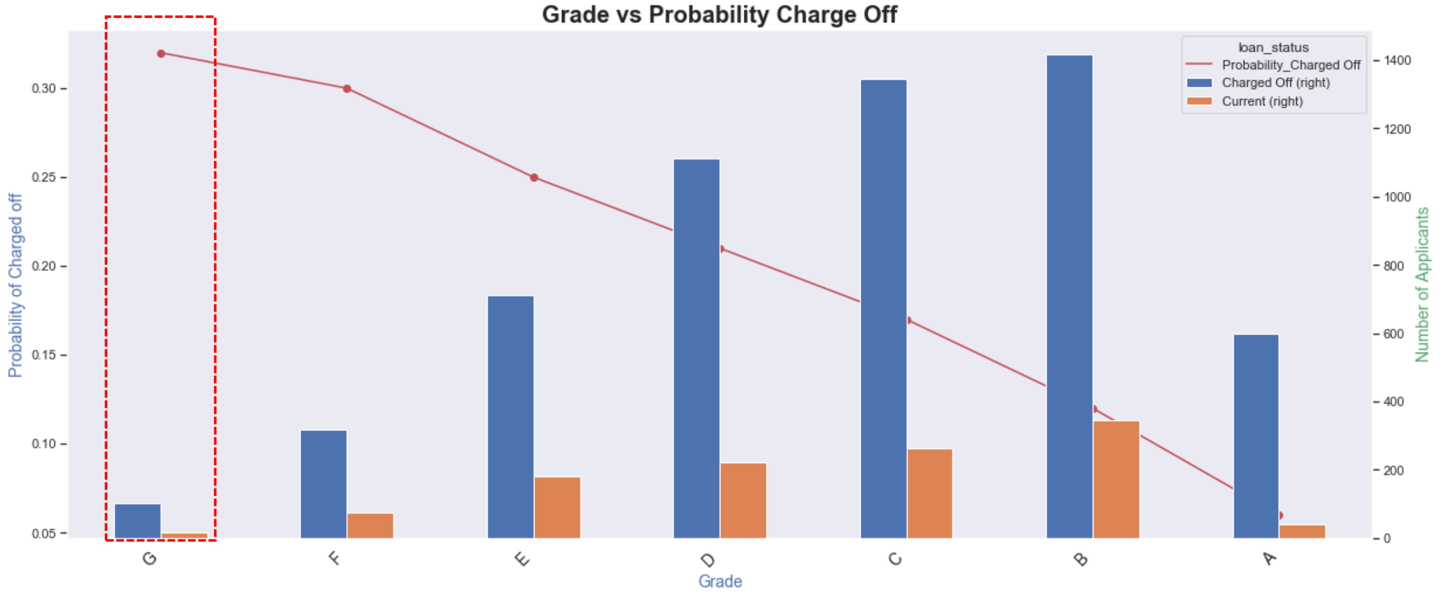


Observations

- Funded amount has been steadily increasing from 2007 to 2011 . 2011 has the highest funding amount
- Loan issuance is least in the Month of June
- The Loan issuance steadily comes down after first quarter & starts rising from July
- Charged Off Loans are 14.8 % of the Total Loans issued

Bivariate Analysis



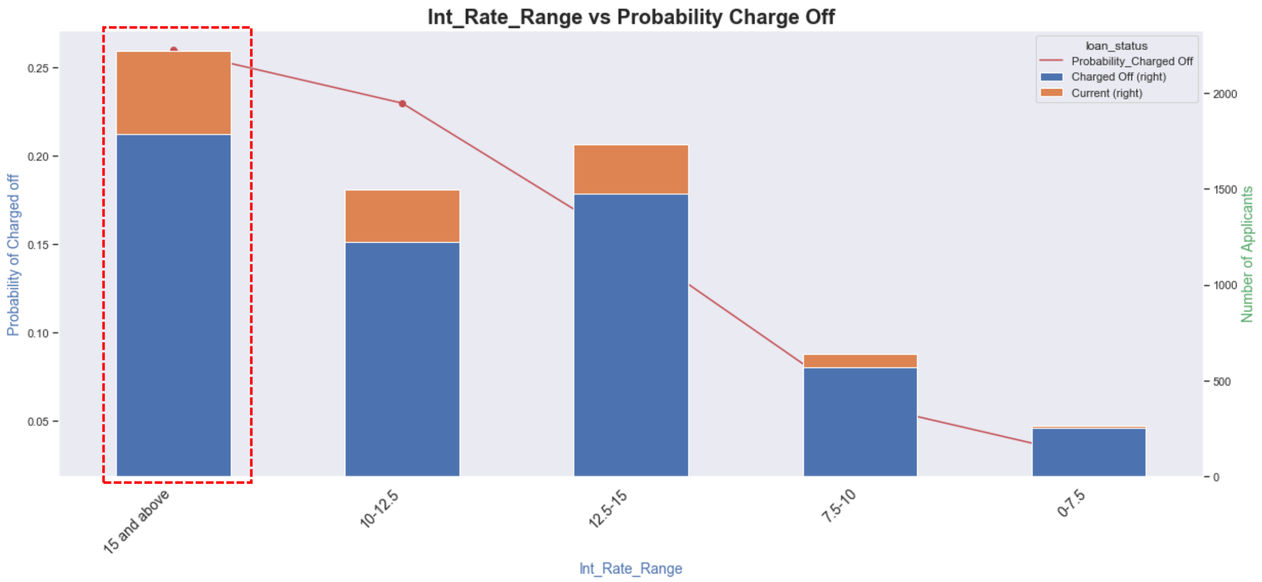


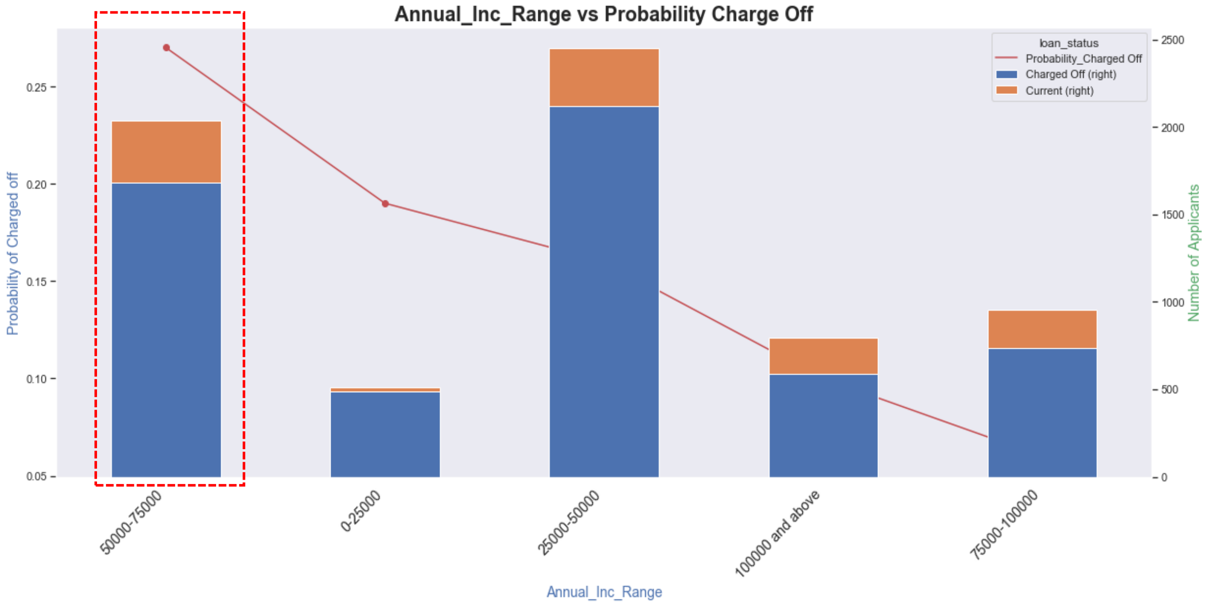
Observations

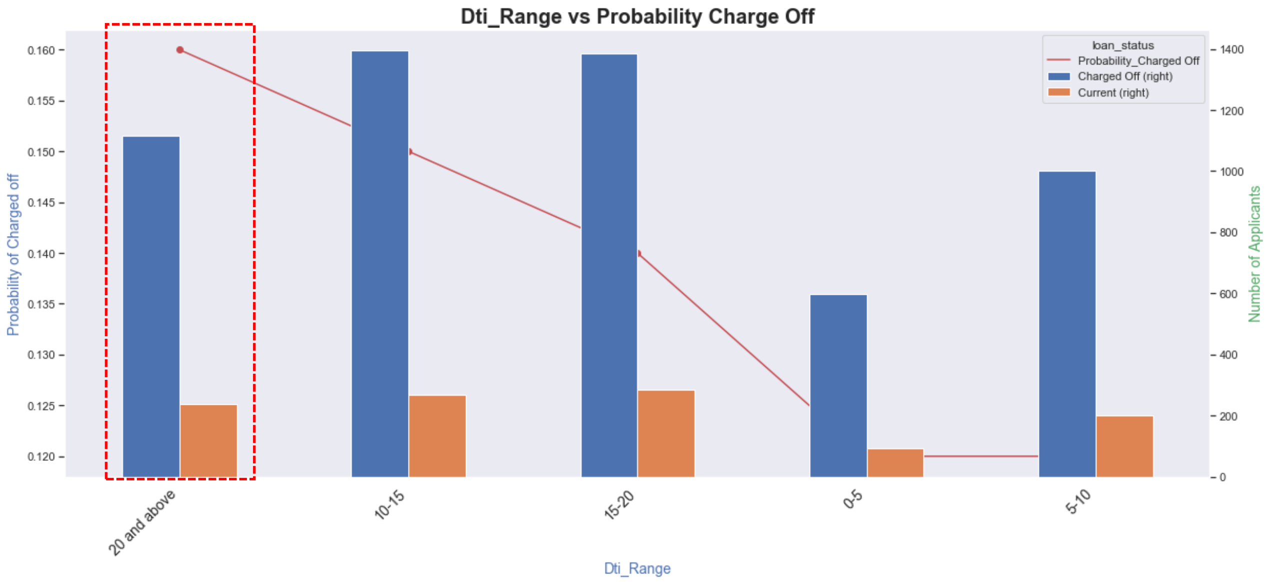
- ❖ The Loan applicants with the purpose of 'small business' has the highest likelihood of a Charge Off
- ❖ The Loan applicants with grade 'G' has the highest likelihood of a Charge Off

CONFIDENTIAL & RESTRICTED

Bivariate Analysis contd..







Observations

- The Loan applicants with the interest rate of '15 and above' has the highest likelihood of a Charge Off
- The Loan applicants with Annual Income between '50000-75000' has the highest likelihood of a Charge Off
- The Loan applicants with the Dti range between '20 & above' has the highest likelihood of a Charge Off

Conclusions

Important Driving factors causing Charged Off

Loan Data	Attribute Type	Value	Indicator for high probability of Charged Off
Purpose	Consumer	small_business	'small_business' - Loans taken for Small business are more likely to be charged off compared to other Loan purposes
Emp Length	Consumer	0 / Independent NB: Independent is the value imputed	The loans taken by Employees who have 0 years / are 'Independent' are most likely to be charged off
Grade	Loan	G	The grade type ‘G’ are more prone to be charged off
Annual Income range	Consumer	50000 - 75000	are most likely to be charged Off
Interest rate	Loan	15 and above	As rate of interest increases the probability of getting charged off increases steadily. The loans taken for interest rate of 15 and above are most likely to be charged Off
Dti	Loan	20 and above	DTI range of ‘20 and above’ is most likely to be charged off. As the number of Debts are higher compared to income, likelihood of being Charged Off is higher.
Addr_state	Consumer	NV	Consumers from ‘NV’ region are more likely to be Charged Off . This could potentially be due to specific consumer behavior in NV state