Lending Club Case Study

EPGP in ML & AI - April 2023

Agenda

This document intends to cover the approach taken for **Lending Club Case Study** from an Exploratory Data Analysis (EDA) perspective. The document is divided in following high-level sections:

- Problem Statement
- Data Understanding
- Data Cleaning
- Data Analysis
- Recommendations (Prescriptive Insights)

Problem Statement

A consumer finance company lends various types of loans to urban customers. On receipt of loan application company makes a decision for loan approval based on applicant's profile. Two types of risk are associated with the bank's decision:

- If the applicant is **likely to repay the loan**, then not approving the loan results in 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 financial loss to the company.

Objective

Objective of this case study is to use Exploratory Data Analysis to understand how **consumer attributes** and **loan attributes** influence the tendency to default and identify risky loan applications to reduce the credit loss.

In other words, Lending Club wants to understand the driving factors behind loan default.

Data Understanding

Data Understanding - Preliminary Observations

Total number of rows	39717
Total number of columns	111
Total number of columns of type Float 64	74
Total number of columns of type Int 64	13
Total number of columns of type Object	24
Total number of columns with missing values	68

Data Understanding - Column Analysis

Column Name(s)	Facts	Observation / Decision
‘id’ and ‘member_id’	These are identifier columns, each row has a unique value for these columns.	Columns will be dropped for data analysis
loan_status	Possible values: Charged Off, Current, Fully Paid	This is the target variable. As we are predicting the likelihood of a new loan to be repaid, we will ignore the rows where value of loan status is ‘Current’
‘term’	Unique values for term of the loan is 36 months and 60 months	This column is a good candidate for bivariate analysis with other columns
pymnt_plan	Only one possible value: ‘n’	Will be dropped for data analysis
‘url’ and ‘desc’	Url represents the link for each loan and desc holds the description of the loan application	Will be dropped for data analysis
'delinq_2yrs', 'earliest_cr_line', 'inq_last_6mths', 'open_acc', 'pub_rec', 'revol_bal', 'revol_util', 'total_acc', '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', 'application_type'	These data points will not be available at the time of loan application.	Due to unavailability at the time of loan application these data points cannot be used for data analysis related to decisioning for approval or rejection of a loan application.

Data Understanding - Column Analysis contd..

Column Name(s)	Facts	Observation / Decision
‘int_rate’	A continuous column	Remove the ‘%’ at the end of the column value and convert the values to float
'installment', 'grade', 'sub_grade'		Check the null values

Data Cleaning

Data Cleaning - Preliminary Actions and Observations

- Remove the rows where loan status is ‘Current’
- Remove the columns identified as not to be considered for data analysis in Data Understanding section
- Remove ‘%’ at the end of ‘int_rate’ column values and convert the values to float

Total number of rows	38577
Total number of columns	85
Total number of columns of type Float 64	65
Total number of columns of type Int 64	5
Total number of columns of type Object	15
Total number of columns with missing values	64

Columns with missing values

'emp_title', 'emp_length', 'title', 'mths_since_last_delinq', 'mths_since_last_record', 'next_pymnt_d', 'collections_12_mths_ex_med', 'mths_since_last_major_derog', 'annual_inc_joint', 'dti_joint', 'verification_status_joint', 'tot_coll_amt', 'tot_cur_bal', 'open_acc_6m', 'open_il_6m', 'open_il_12m', 'open_il_24m', 'mths_since_rcnt_il', 'total_bal_il', 'il_util', 'open_rv_12m', 'open_rv_24m', 'max_bal_bc', 'all_util', 'total_rev_hi_lim', 'inq_fi', 'total_cu_tl', 'inq_last_12m', 'acc_open_past_24mths', 'avg_cur_bal', 'bc_open_to_buy', 'bc_util', 'chargeoff_within_12_mths', 'mo_sin_old_il_acct', 'mo_sin_old_rev_tl_op', 'mo_sin_rcnt_rev_tl_op', 'mo_sin_rcnt_tl', 'mort_acc', 'mths_since_recent_bc', 'mths_since_recent_bc_dlq', 'mths_since_recent_inq', 'mths_since_recent_revol_delinq', 'num_accts_ever_120_pd', 'num_actv_bc_tl', 'num_actv_rev_tl', 'num_bc_sats', 'num_bc_tl', 'num_il_tl', 'num_op_rev_tl', 'num_rev_accts', 'num_rev_tl_bal_gt_0', 'num_sats', 'num_tl_120dpd_2m', 'num_tl_30dpd', 'num_tl_90g_dpd_24m', 'num_tl_op_past_12m', 'pct_tl_nvr_dlq', 'percent_bc_gt_75', 'pub_rec_bankruptcies', 'tax_liens', 'tot_hi_cred_lim', 'total_bal_ex_mort', 'total_bc_limit', 'total_il_high_credit_limit'

Data Cleaning - Missing Value Handling

Action	Command	Total / Count
Find all columns having no values	data.isnull().sum()[data.isnull().sum() == len(data)]	55
Drop the 55 columns identified above	data.drop(data.isnull().sum()[data.isnull().sum() == len(data)].index, axis=1, inplace=True)	30 columns remaining
Find columns having missing values of the remaining 30 columns	data.isnull().sum()[data.isnull().sum() > 0]	9 columns mths_since_last_delinq has 24905 missing values mths_since_last_record has 35837 missing values
Drop columns mths_since_last_delinq, mths_since_last_record	data.drop(['mths_since_last_delinq', 'mths_since_last_record'], axis=1, inplace=True)	28 columns remaining

Remaining Columns which have missing values with number of missing values

emp_title - 2386	chargeoff_within_12_mths - 56	pub_rec_bankruptcies - 697	title - 11
emp_length - 1033	collections_12_mths_ex_med - 56	tax_liens - 39	

Data Analysis

Data Analysis - Preliminary Observations

Total number of rows	38577
Total number of columns	28
Total number of columns of type Float 64	9
Total number of columns of type Int 64	5
Total number of columns of type Object	14
Total number of columns with missing values	7

```
data.isnull().sum()[data.isnull().sum() > 0]
✓ 0.1s

emp_title                     2386
emp_length                    1033
title                           11
collections_12_mths_ex_med      56
chargeoff_within_12_mths        56
pub_rec_bankruptcies           697
tax_liens                       39
dtype: int64
```

- Drop the rows where emp_length column value is missing
- Drop all other columns where any values are missing as it is not used for analysis

Data Analysis - Derived Columns

- Add a new column 'result' based on 'loan_status': if status = 'Fully Paid' then 1 else 0
- Add a new column 'annual_inc_in_mills' based on 'annual_inc' column holding annual income in millions
- Add a new column 'term_months' which holds the integer value from 'term' column
- On right is the result after adding above columns

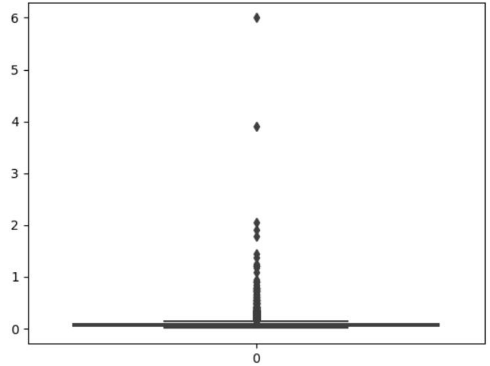
```
Int64Index: 37544 entries, 0 to 39716
Data columns (total 25 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   loan_amnt            37544 non-null  int64
 1   funded_amnt          37544 non-null  int64
 2   funded_amnt_inv      37544 non-null  float64
 3   term                 37544 non-null  object
 4   int_rate             37544 non-null  float64
 5   installment          37544 non-null  float64
 6   grade                37544 non-null  object
 7   sub_grade            37544 non-null  object
 8   emp_length           37544 non-null  object
 9   home_ownership       37544 non-null  object
 10  annual_inc           37544 non-null  float64
 11  verification_status  37544 non-null  object
 12  issue_d              37544 non-null  object
 13  loan_status          37544 non-null  object
 14  purpose              37544 non-null  object
 15  zip_code             37544 non-null  object
 16  addr_state           37544 non-null  object
 17  dti                  37544 non-null  float64
 18  initial_list_status  37544 non-null  object
 19  policy_code          37544 non-null  int64
 20  acc_now_delinq       37544 non-null  int64
 21  delinq_amnt          37544 non-null  int64
 22  result               37544 non-null  int64
 23  annual_inc_in_mills  37544 non-null  float64
 24  term_months          37544 non-null  int64
```

Data Analysis - Annual Income Analysis

```
## describe annual income in millions
data['annual_inc_in_mills'].describe()
```

```
count    37544.000000
mean         0.069407
std          0.064677
min          0.004000
25%          0.041000
50%          0.060000
75%          0.083000
max          6.000000
Name: annual_inc_in_mills, dtype: float64
```

```
sns.boxplot(data['annual_inc_in_mills'])
plt.show()
```

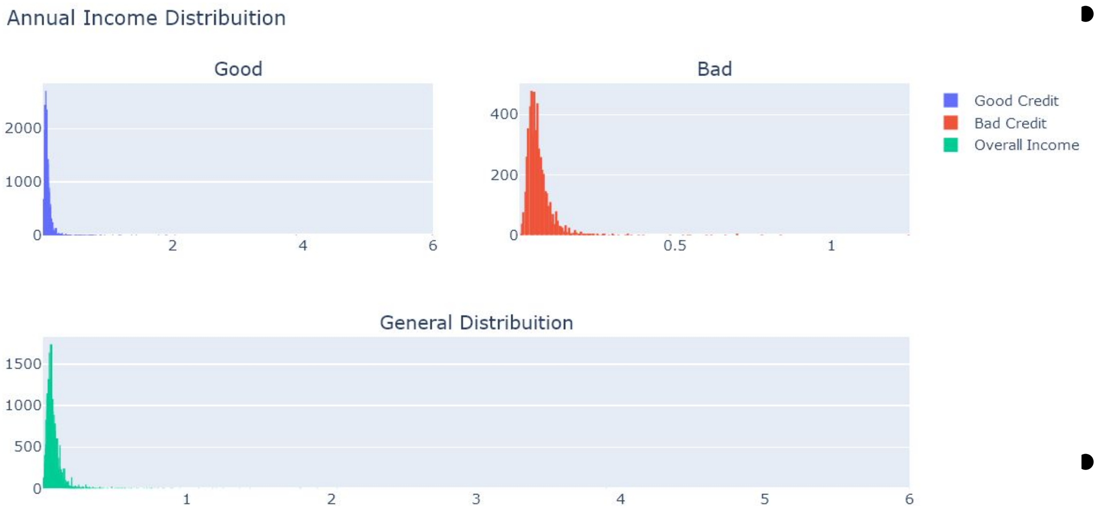


- There is a huge difference between 75 percentile and max value evident from above table.
- As visible from box plot there are certain outliers
- There are 22 rows where income is more than 0.8M and out of 22 rows 2 rows are present which are 'Charged Off'.
- We will remove these 22 rows as this is skewing the data due to high income

```
In [97]: ## Let's remove these 22 rows i.e. further analysis
         ## will be done considering rows where annual income <= .8M
         data = data[data['annual_inc_in_mills'] <= 0.8]
         data.shape

Out[97]: (37522, 25)
```

Data Analysis - Annual Income Analysis contd..



- Let's divide the dataset into three -
  - borrowers with annual income between 0 and 0.1 - Low Income borrowers
  - borrowers with annual income between 0.1 and 0.2 - Medium Income borrowers
  - borrowers with annual income between 0.2 and above - High Income borrowers
- We will create a derived variable 'income_group' using above criteria

Inference - Borrowers with annual income between 0M to 0.2M are the largest chunk of borrowers

Data Analysis - Low Annual Income Group Analysis

Below are the observations from above:

- There are specific peaks at 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09 (all in millions)
- There are specific peaks at mid range i.e. 0.035, 0.045, 0.055, 0.065, 0.075 (all in millions) etc.

Same observations are applicable for Medium Annual Income group borrowers

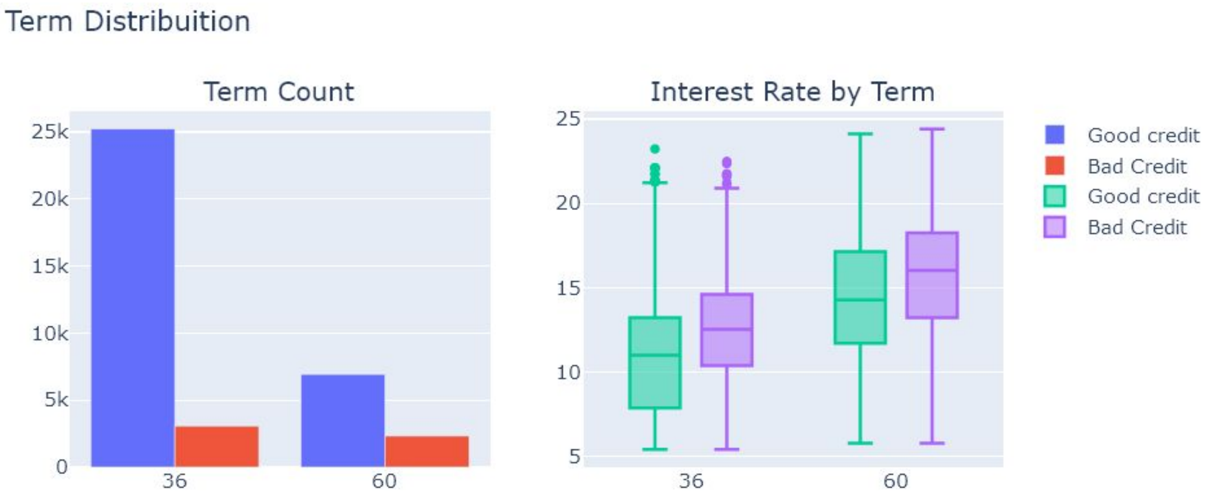
- **Inference - At specific income levels for low income group there is a spurt in the loan offering that is the loan approval is not linearly increasing as the income increases**
- **Inference - There is a peak at 0.06M income level for the number of loans in the low income group**

Data Analysis - Interest Rate against Income Group (derived)

Observe that interquartile range for good credit loans is always below the interquartile range for bad credit loans for different income groups

- **Inference - higher interest rate loans are more likely to default across income groups**

Data Analysis - Interest Rate against Term



Observe that mostly loans are offered with 36 months tenure

Also the percentage charged off loans are more for 60 months term loans compared to 30 months term loans

Interest rate Interquartile range spread for 60 months term loan is higher than 30 months term loans both for good and bad credit

- **Inference - For both term types higher interest rate are likely to result in default**
- **Inference - 60 month term loans are more likely to be defaulted compared to 30 months term loans**

Data Analysis - Interest Rate against Purpose & Grade

```
## Let's check interest rate against loan purpose and loan amount
pd.pivot_table(data=data, values='int_rate', index='grade', columns='purpose', aggfunc=np.mean)
```

purpose	car	credit_card	debt_consolidation	educational	home_improvement	house	major_purchase	medical	moving	other	renewable_ener
grade											
A	7.148417	7.456471	7.359154	8.356456	7.224418	7.465222	7.101599	7.243265	7.436585	7.489459	7.1475
B	10.814956	11.006890	11.044700	11.260408	10.968079	10.955152	10.965819	10.927538	11.010057	11.074093	10.9619
C	13.558798	13.472739	13.572164	12.928444	13.548357	13.661515	13.463869	13.523154	13.650642	13.450531	13.5140
D	15.751721	15.656323	15.678166	14.745161	15.625704	15.915000	15.532342	15.880506	15.564286	15.622753	16.1544
E	17.023704	17.645097	17.725170	16.326429	17.695663	17.680000	17.430645	16.966250	17.787143	17.651463	17.0240
F	18.929000	19.570532	19.703668	17.245000	19.937455	19.479286	18.971538	19.343846	19.285556	19.704324	18.4700
G	21.255000	21.624286	21.431481	21.270000	21.024667	21.190000	21.922500	20.620000	21.038000	21.617778	23.0200

- **Inference - Interest Rates increase with the grade of the loan i.e. grade A loan will have lower interest rate compared to grade G loan, across loan purposes.**

Data Analysis - Employee Length Distribution

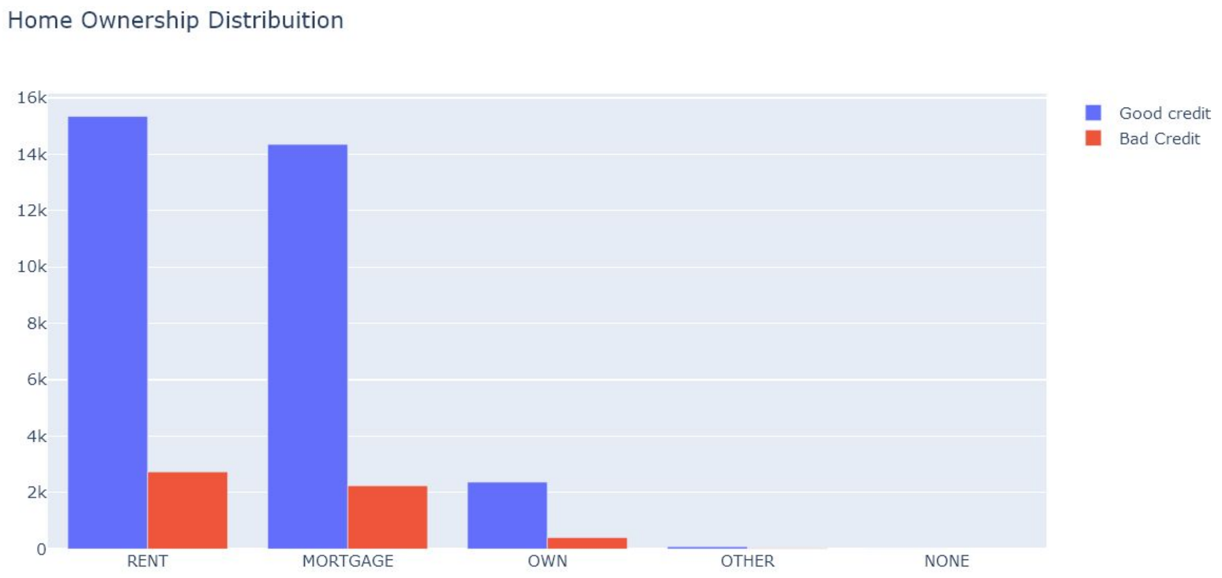
Most of the loans are offered to borrowers having employment length more than 10 years or less than 1 year

Most defaults are by borrowers having employment length more than 10 years or less than 1 year

Also the percentage charged off loans are more for 60 months term loans compared to 30 months term loans

Inference - Number of loans offered to borrowers with experience of 1 to 5 years is more compared to borrowers with experience of 6 to 9 years

Data Analysis - Home Ownership Distribution



- **Inference - Most number of loans are offered to borrowers who have either rented or mortgaged their home**

Data Analysis - Loan Amount Distribution

There are specific peaks at rounded numbers of loan amount for e.g. at 5k, 10k, 20k, 25k, 30k, 35k

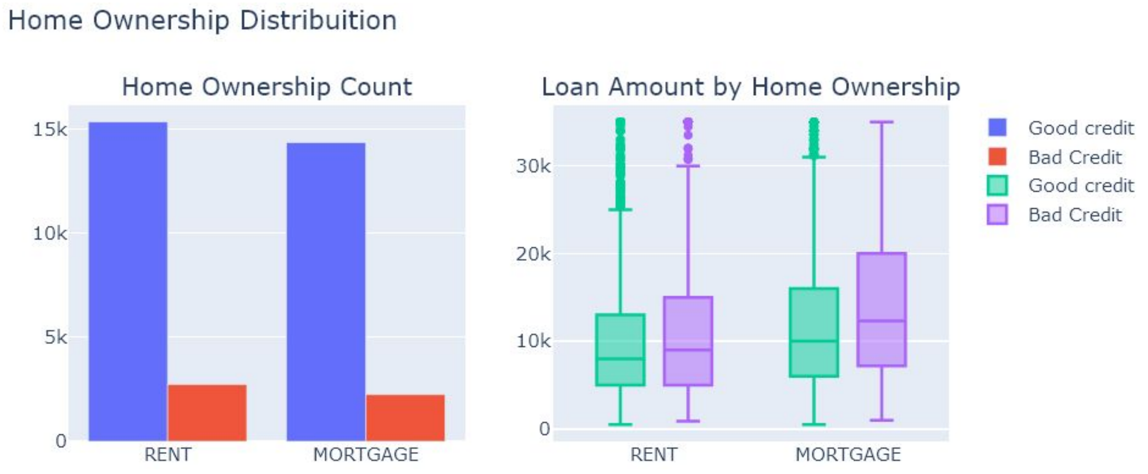
There are less loan offered for loan amount between 20k to 35k compared to 0 to 20k

```
data['loan_amnt'].value_counts().head(10)
```

```
10000    2742
12000    2213
5000     1959
6000     1834
15000    1805
8000     1524
20000    1514
25000    1308
4000     1086
7000      984
Name: loan_amnt, dtype: int64
```

Inference - Most number of loans are offered to borrowers who have either rented or mortgaged their home

Data Analysis - Loan Amount vs Home Ownership (Rent, Mortgage)



For both rented and mortgage home ownership, interquartile range for loan amount is bigger for bad credit compared good credit

Loan Amount Interquartile range for borrowers with mortgaged home ownership is bigger compared borrowers with rented home ownership

- **Inference - For both RENT and MORTGAGE home ownership type, higher loan amount can result in bad credit**
- **Inference - Borrowers who already have a mortgage (home loan) are more likely to default if the loan amount is higher**

Data Analysis - Issue Date Analysis

Two new columns are added to capture the month and year in which loan is issued (issue_month, issue_year)

```
data['issue_year'].value_counts()
```

```
2011    19801
2010    11214
2009     4716
2008     1562
2007      251
Name: issue_year, dtype: int64
```

```
data['issue_month'].value_counts()
```

```
12    4120
11    3890
10    3637
9     3394
8     3321
7     3253
6     3094
5     2838
4     2756
3     2632
1     2331
2     2278
Name: issue_month, dtype: int64
```

There are more loans issued in year 2011 compared to combined totals of 2010, 2009, 2008 and 2007

There are more loans issued in 2nd half of the year compared to 1st half of the year

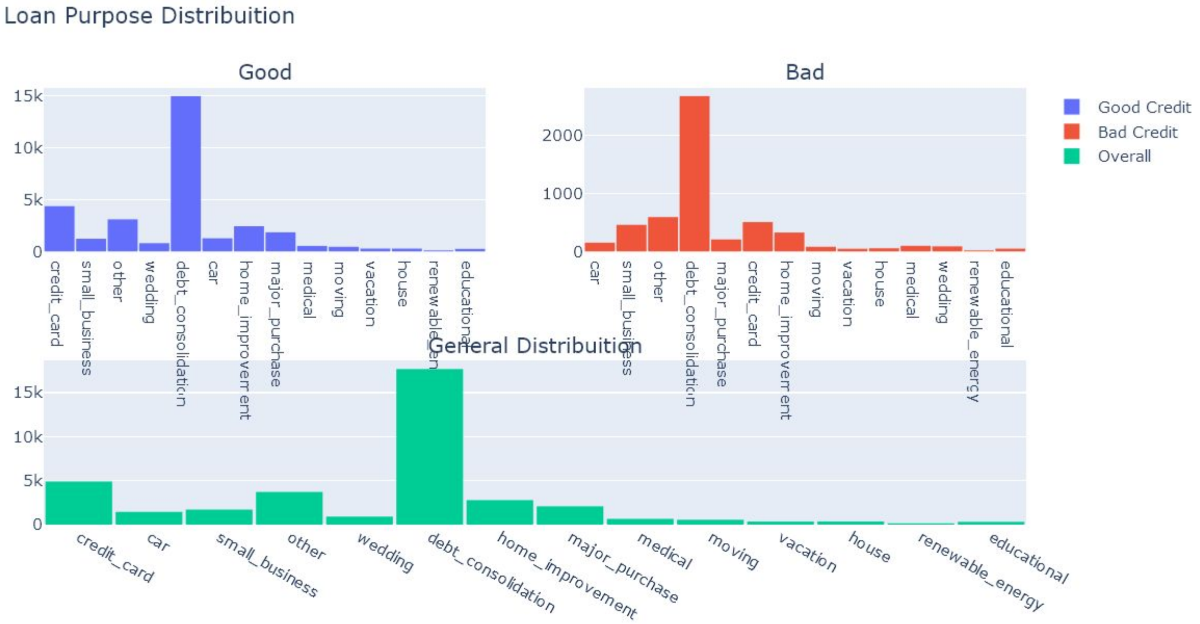
Data Analysis - Loan Issue Month Analysis

Number of loans issued are increasing from Jan to Dec for all years

Number of loans issued in Feb is least compared to other months, Jan is a close second

Inference - One reason for this almost linear increase may be to due to pressure to achieve yearly sales target

Data Analysis - Loan Purpose Analysis




```
data['purpose'].value_counts()

debt_consolidation    17675
credit_card            4899
other                  3713
home_improvement       2785
major_purchase         2080
small_business         1710
car                    1448
wedding                 913
medical                 656
moving                  552
house                   354
vacation                348
educational             317
renewable_energy         94
```

- **Inference - Maximum number of loans issued is for the purpose of debt consolidation, with next two purposes for loans being credit card and others**

Data Analysis - Debt to Income (DTI) Analysis

Understanding of DTI is how much of a customer's income goes towards fulfilling current debt

As a guideline it is recommended that dti should be lower than 36%

Ideally, a person having a high value of dti should not be given a high loan amount

Create a derived variable from 'dti' called 'dti_group' with below criteria

- 0 <= dti < 5 - very low;
- 5 <= dti < 10 - low;
- 10 <= dti < 15 - medium;
- 5 <= dti < 20 - high;
- 20 <= dti < 25 - very high;
- 25 <= dti - extremely high;

Data Analysis - Debt to Income (DTI) Analysis


```
data['dti_group'].value_counts()
```

```
Medium            9394
High              8614
Low               7643
Very High         6421
Very Low          4874
Extremely High     598
Name: dti_group, dtype: int64
```

Maximum # of loans are issued for medium, high and low dti group

Inference - For high loan amount, even with Low or Very Low DTI borrowers can default

Recommendations (Prescriptive Insights)

Recommendations

- **Create products targeted for high income group customers**
  - **This will increase the market share of the high income group customers**
  - **They are more likely to repay compared to low income group customers**

- **Loans issued for round figure incomes are considerably higher, there is a need to understand why this is happening**
  - **Improvement in verification process**
  - **Improvement in KYC process to collect more accurate data**

- **Uniform sales across the year**
  - **Offer better incentives and place better processes for sales team to perform uniformly across the year rather than slog at the end of the year compromising the quality of the loan issued**

Recommendations contd..

- Offer more loans for categories other than debt consolidation
  - This will diversify portfolio for Lending Club and hence reduce overall credit risk

- There is a higher risk exposure due to the number of loans issued to borrowers who have a mortgage
  - Diversify portfolio by extending loan to borrowers who have less credit to pay

Thank You