


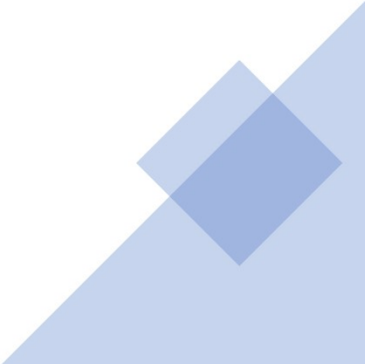


Lending_Club_Case_Study_Project_Ajit_Siddhant

Project done by :-
Ajit John
Siddhant Srivastava




Lending Club :EDA case study

- There are four major parts that are needed to be done for this case study:
 - 1. Data understanding
 - 2. Data cleaning (cleaning missing values, removing redundant columns etc.)
 - 3. Univariate analysis
 - 4. Bivariate analysis
 - 5. Observations and Results
- 



Data Cleaning

1. Firstly, we are going to check the percentage of missing values
 2. Secondly, we will remove all those with very high missing percentage
 3. Next ,for columns with less missing percentage we will be performing Imputations
 4. Then, we will just identify the correct metric to impute the column.
 5. Finally, we will be dropping rows where the missing percentage is quite high
- 

Data Analysis

- i. The objective is to identify predictors of default so that at the time of loan application, we can use those variables for approval/rejection of the loan.
 - i. There are broadly three types of variables –
 - ii. those which are related to the applicant (demographic variables such as age, occupation, employment details etc.),
 - iii. Loan characteristics (amount of loan, interest rate, purpose of loan etc.) and
 - iv. Customer behavior variables (those which are generated after the loan is approved such as delinquent 2 years, revolving balance, next payment date etc.).
- ii. Now, the customer behavior variables are not available at the time of loan application, and thus they cannot be used as predictors for credit approval.
- iii. The ones marked 'current' are neither fully paid not defaulted, so get rid of the current loans. Also, tag the other two values as 0 or 1 to make your analysis simple and clean.



Data Analysis

- **Data Analysis: Univariate Analysis**
 - For univariate analysis, we will check the default rate across various categorical features.
 - For continuous features, we will perform binning and then you may perform univariate analysis.
- **Data Analysis: Bivariate Analysis**
 - Here we are planning to choose two or more features to understand the Default variable

Variables those can be dropped

id , member_id, emp_title, url, desc, zip_code, addr_state, funded_amnt (as funded_amnt_inv is present), title (as purpose is present),

```
In [19]: # Dropping columns mentioned above
loan_data.drop(columns=["id", "member_id", "emp_title", "url", "desc", "zip_code", "addr_state", "funded_amnt", "title"], inplace=True)
print(loan_data.shape)
loan_data.columns
```

```
(38575, 33)
Out[19]: Index(['loan_amnt', 'funded_amnt_inv', 'int_rate', 'installment', 'grade',
               'sub_grade', 'emp_length', 'home_ownership', 'annual_inc', 'purpose',
               'dti', 'delinq_2yrs', 'inq_last_6mths', 'open_acc', 'revol_bal',
               'revol_util', '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', 'verified', 'term_36_not60', 'loan_paid',
               'issue_d_year', 'issue_d_month', 'credit_age', 'closed_acc'],
              dtype='object')
```

Removing skewed data

In [24]:

```
def getBoxPlot(df: pd.DataFrame, percentile: float) -> None:
    # for attr in ("loan_amnt", "int_rate", "installment", "annual_inc", "dti", "open_acc", "closed_acc"):
    for attr in ("installment", "annual_inc",):
        plt.figure()
        (df.loc[df[attr] < df[attr].quantile(percentile)]).boxplot(attr)
        print(attr, df[attr].quantile(percentile))

# for x in (1.0, 0.99, 0.98, 0.97):
for x in (0.94, 0.93, 0.9):
    print(f"percentile: {x}")
    getBoxPlot(loan_data, x)
```

```
percentile: 0.94
installment 709.8367999999999
annual_inc 134000.0
percentile: 0.93
installment 682.74
annual_inc 127500.0
percentile: 0.9
installment 620.9
```

Removing sub-categories with very less data

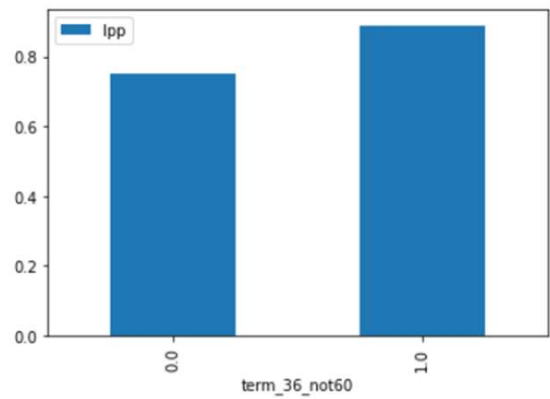
```
In [29]: ...  
Too less data for G under grade  
Too less data for NONE under home_ownership  
Too less data for OTHER under home_ownership  
Too less data for renewable_energy under purpose  
Too less data for 4 under inq_last_6mths  
Too less data for 5 under inq_last_6mths  
Too less data for 6 under inq_last_6mths  
Too less data for 7 under inq_last_6mths  
Too less data for 8 under inq_last_6mths  
...  
  
loan.drop(index=loan.index[loan["home_ownership"] == 'NONE'], inplace=True)  
loan.drop(index=loan.index[loan["home_ownership"] == 'OTHER'], inplace=True)  
loan.drop(index=loan.index[loan["inq_last_6mths"] >= 4], inplace=True)
```


Univariate & Segmented Univariate Analysis

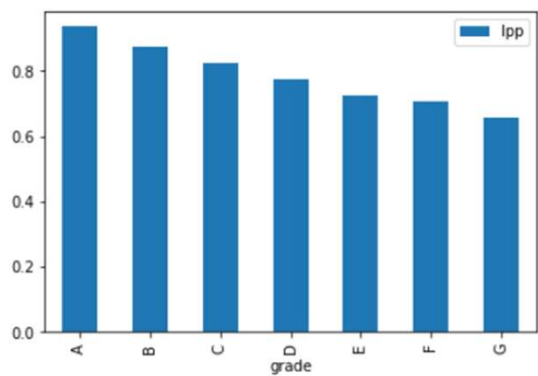
```
: 1 def getOrderedList(allVals: list) -> list:
2     months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun",
3               "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]
4     if type(allVals[0]) == str and (allVals[0].split('_')[-1]).isdigit():
5         sortedAllVals = sorted(allVals, key=lambda d: int(d.split('_')[-1]))
6     else:
7         if allVals[0] in months:
8             sortedAllVals = months
9         else:
10            sortedAllVals = sorted(allVals)
11    return sortedAllVals
```

```
1 for attr in ("funded_amnt_inv_groups",
2             "grade",
3             "sub_grade",
4             "emp_length",
5             "home_ownership",
6             "purpose",
7             "inq_last_6mths",
8             "term_36_not60",
9             "loan_paid",
10            "issue_d_year",
11            "issue_d_month",
12            "loan_amnt_groups",
13            "int_rate_groups",
14            "installment_groups",
15            "annual_inc_groups",
16            "dti_groups",
17            "credit_age_groups",
18            "closed_acc_groups",
19            "open_acc_groups",
20            "revol_util_groups",
21            ):
22     d = dict()
23     sortedAllVals = getOrderedList(loan[attr].value_counts().index.values)
24     for val in sortedAllVals:
25         if len(loan.loc[loan[attr] == val]) == 0:
26             print(f"No data for {val} under {attr}")
27             continue
28         elif len(loan.loc[loan[attr] == val]) < len(loan)/(10*len(loan[attr].value_counts().index.values)):
29             print(f"Too less data for {val} under {attr}")
30         d[val] = len(loan.loc[(loan[attr] == val) & (loan["loan_paid"] == 1)]/len(loan.loc[loan[attr] == val]))
31     plt.figure()
32     (pd.DataFrame({attr:d.keys(), "lpp" : d.values()})).plot.bar(x=attr, y="lpp")
33     plt.figure(figsize=(15, 5))
34     sns.countplot(x=attr, hue="loan_paid", data=loan, order=d.keys())
```

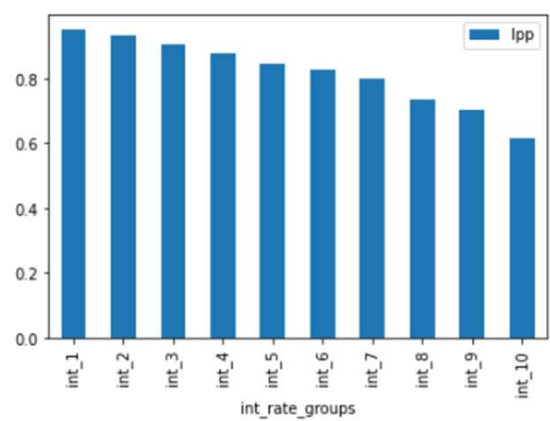
<Figure size 432x288 with 0 Axes>



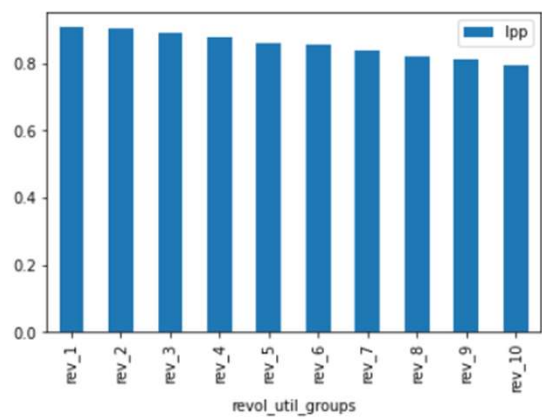
<Figure size 432x288 with 0 Axes>



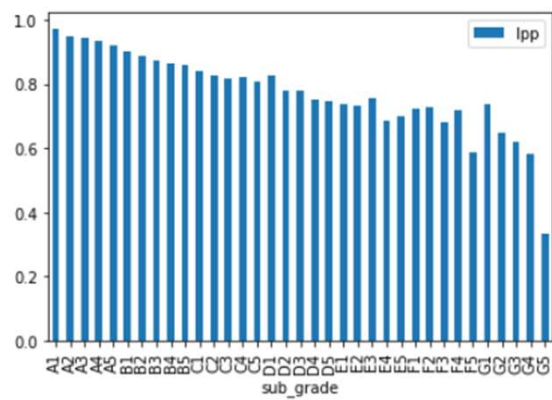
<Figure size 432x288 with 0 Axes>



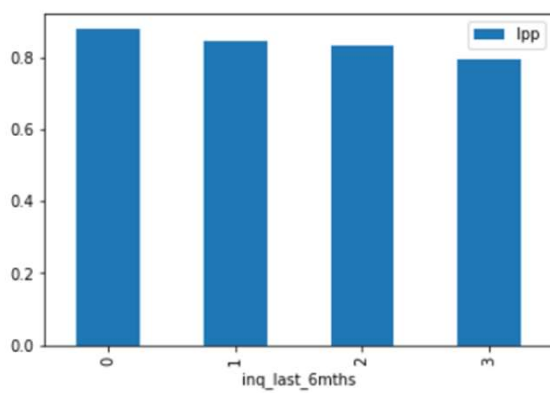
<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>





Observations (Univariate Analysis)

Indicators for loan payment:

Higher Grade or Sub-Grade

Lower interest-rate

Lower revolving utility percentage

Less enquired in last six months (inq_last_6mths).

Term is 36 months.

If purpose is "wedding", "major purchase", credit-card" & "car".

Indicators for loan default:

purpose is marked as 'other'

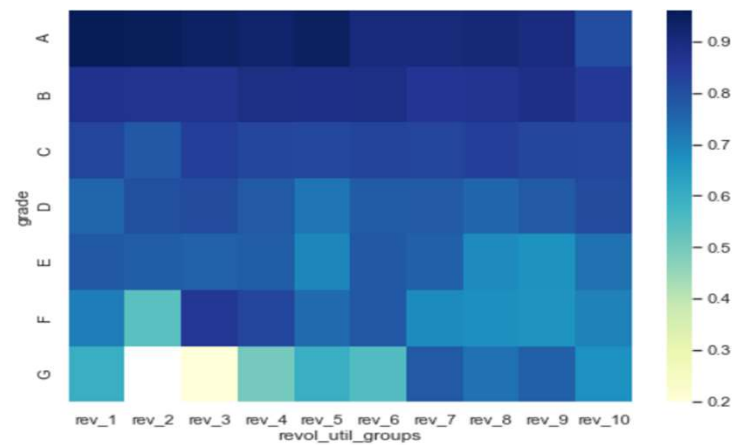
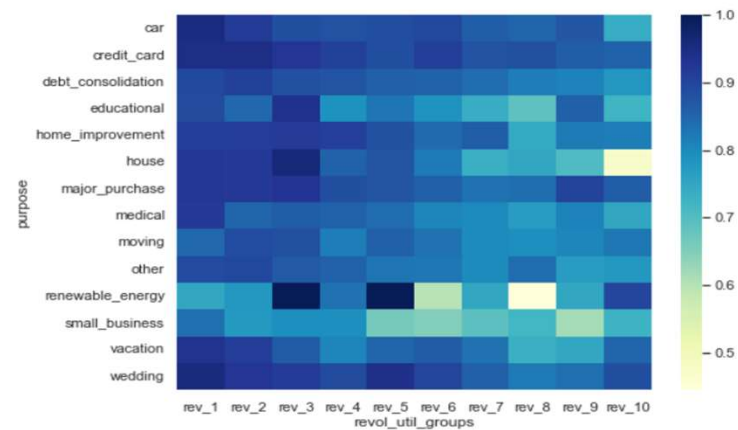
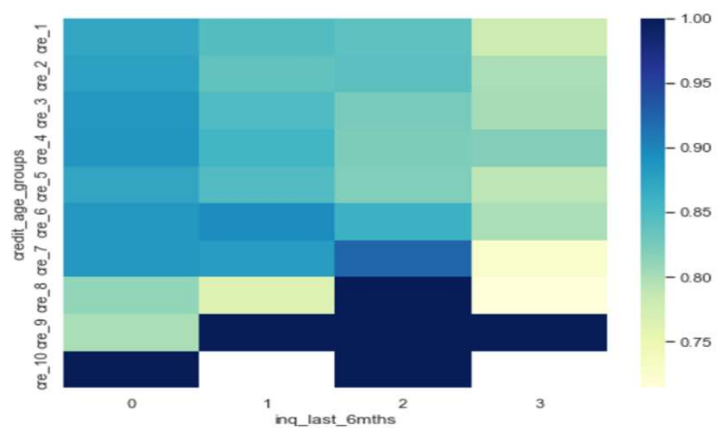
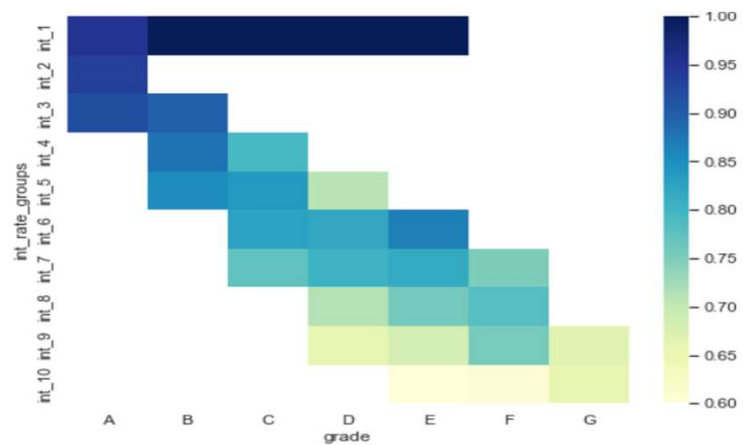


Bivariate Analysis

Now that we have analyzed each of the variables and its impact on the loan-status, let us take group of variables together and analyze their combined effect on the loan-status. These categories are based on our business understanding. The original distribution column shows the average trend in all the data and we compare that with the data after applying our conditions.

```
1 cat_var = ["grade", "sub_grade", "emp_length", "home_ownership", "purpose", "inq_last_6mths",
2           "verified", "term_36_not60", "issue_d_year", "issue_d_month", "loan_amnt_groups", "int_rate_groups",
3           "installment_groups", "annual_inc_groups", "dti_groups", "credit_age_groups", "funded_amnt_inv_groups",
4           "revol_util_groups"]
```

```
1 colors = ["#a9fea9", "#fca9a9"]
2 sns.set(style='whitegrid', palette=sns.color_palette(colors))
3 by = 'loan_status'
4 order = ['Fully Paid', 'Charged Off']
5 def bivariateCategoricalAna (data: pd.DataFrame, x_axis, y_axis, title=None, figsize=(8,6)) -> None:
6     """
7     Parameters:
8         data : defaulted at the 'loan' variable, but can be changed externally if needed
9         x, y : columns of data in the corresponding axis, both categorical
10        figsize : a default have is given, but can be overridden
11    Returns:
12        None, does a heatmap plot
13    """
14    fig, ax = plt.subplots(figsize=figsize)
15    pt = pd.pivot_table(data=data, values='loan_paid', index=y_axis, columns=x_axis)
16    sns.heatmap(pt, ax=ax, cmap='YlGnBu')
17    if title is not None:
18        plt.title(title)
19    plt.show()
```






Observations (BiVariate Analysis)

- *Interest-rate & grade is highly correlated : higher grade indicates lower interest rate.*
- *Grade is a strong indicator. Higher grade (A) will have high probability to pay loan.*
- *Revolving utility percentage (lower utility leads to higher loan pay probability) has low impact on "wedding", "major purchase", credit-card" & "car" purpose category.*

In higher credit-age group (554 month +), trend for inq_last_6mths acts opposite relative to univariate analysis outcome.

If term is 36 months, loan-payment probability is higher as indicated by univariate analysis.

Higher income group(102k+) have high probability of paying loan.



Recommendations

```
1
2 # No need to look into output of this. We wanted to give a glimpse how we did bivariate
3 # analysis. I am sure that there must be some better way to do it.
4 for i in range(2, len(cat_var)):
5     for j in range(1, len(cat_var)):
6         if i == j:
7             continue
8         bivariateCategoricalAna(loan, x_axis=cat_var[i], y_axis=cat_var[j], figsize=(10, 15))
9
```

Applicants with probability of default

- In general when term is “60 months” & especially when
 - “revol_util” is more than 70%.
 - Purpose is “educational” (though less data but default rate is 50%) or “small business”.
 - inq_last_6mths count is 2+.
 - emp_length is 5.
 - Interest rate is higher than 15%.
- If purpose is “small_business” &
 - Home ownership is “RENT”
 - verification_status is “verified”.
- In general when interest-rate is higher than 15% but especially when
 - Purpose is medical (though less data but default rate is higher than payment rate).