Lending Club Case Study

By Jawad Sonalkar & Noushad Chono Kadavath

Goal

The aim is to identify patterns which indicate if a person is likely to default, which may be used for taking actions such as denying the loan, reducing the amount of loan, lending (to risky applicants) at a higher interest rate, etc by analysing the past loan applications.

1. Intro

In this case study, use EDA to understand how **consumer attributes** and **loan attributes** influence the tendency of default using the the data given which contains the information about past loan applicants and whether they ‘defaulted’ or not.

When a person applies for a loan, there are two types of decisions that could be taken by the company:

- ➔ Loan accepted: If the company approves the loan, there are 3 possible scenarios described below:
  - Fully paid: Applicant has fully paid the loan.
  - Current: Applicant is in the process of paying the instalments
  - Charged-off: Applicant has not paid the instalments in due time.
- ➔ **Loan rejected**: The company had rejected

Business Understanding

This company is the largest online loan marketplace, facilitating personal loans, business loans, and financing of medical procedures. Borrowers can easily access lower interest rate loans through a fast online interface. The credit loss is the amount of money lost by the lender when the borrower refuses to pay or runs away with the money owed. In other words, borrowers who default cause the largest amount of loss to the lenders. The company wants to understand the driving factors (or driver variables) behind loan default, i.e. the variables which are strong indicators of default

Tip

In this scenario, we're working with the data provided.

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

- **loss of business** by rejecting the genuine applicant.
- **financial loss** for the company by accepting wrong applicant.

Data understanding

```
loan.head()
```

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	...	num_tl_90g_dpd_24m	num_tl_op_past_12m
0	1077501	1296599	5000	5000	4975.0	36 months	10.65%	162.87	B	B2	...	NaN	NaN
1	1077430	1314167	2500	2500	2500.0	60 months	15.27%	59.83	C	C4	...	NaN	NaN
2	1077175	1313524	2400	2400	2400.0	36 months	15.96%	84.33	C	C5	...	NaN	NaN
3	1076863	1277178	10000	10000	10000.0	36 months	13.49%	339.31	C	C1	...	NaN	NaN
4	1075358	1311748	3000	3000	3000.0	60 months	12.69%	67.79	B	B5	...	NaN	NaN

5 rows × 111 columns

```
loan.shape
```

```
(39717, 111)
```

Tip

Identify the depth and nature of the dataset provided.

Used following Python modules to get these details.

- pandas

Data cleaning

In this step we'll make the clean and make the data ready by considering following factors,

- Remove invalid / irrelevant columns: As there are more than 100+ some columns ought to be irrelevant for our analysis, like columns which have all null values or columns which have a same value across all rows. We can remove these columns from the dataset
- Fix rows: Remove duplicate rows or rows which is having a missing value for our key analysis field ie loan status or changing the dtypes
- Removing columns with all null values
- Removing columns with all same values
- Removing columns with personal info field as we dont need personal info to take decision.
- Removing columns based on business understanding like post charge collection_recovery_fee, recoveries etc
- Removing current loan status as we are looking for fully paid vs Charged-off, current loan will not add any value, so lets remove those records

Tip

Go through the data and understand the nature and identify the details required and not required. When looking at the dataset we found out that there are some columns contains NA values like the very second column from data dictionary ie **acc_open_past_24mths**

Handling missing Values

- We can see columns like mths_since_last_delinq, mths_since_last_record and next_pymnt_d has more than 60% missing values. This will not add any benefits in our analysis, so lets drop the columns

```
(loan.isna().sum()/len(loan.index))*100
```

```
id                            0.000000
loan_amnt                     0.000000
funded_amnt                   0.000000
funded_amnt_inv               0.000000
term                          0.000000
int_rate                      0.000000
installment                   0.000000
grade                         0.000000
sub_grade                     0.000000
emp_length                    2.677761
home_ownership                0.000000
annual_inc                    0.000000
verification_status           0.000000
issue_d                       0.000000
loan_status                   0.000000
purpose                       0.000000
zip_code                      0.000000
addr_state                    0.000000
dti                           0.000000
mths_since_last_delinq       64.559193
mths_since_last_record       92.897322
next_pymnt_d                100.000000
pub_rec_bankruptcies          1.806776
dtype: float64
```

Tip

For now lets leave all the missing values as it is

Standardise Values

- In this section we'll try to standardise the values, like fix the dtypes

```
loan.dtypes
```

```
id                        int64
loan_amnt                 int64
funded_amnt               int64
funded_amnt_inv         float64
term                     object
int_rate                 object
installment             float64
grade                    object
sub_grade                object
emp_length               object
home_ownership           object
annual_inc              float64
verification_status      object
issue_d                  object
loan_status              object
purpose                  object
zip_code                 object
addr_state               object
dti                     float64
pub_rec_bankruptcies    float64
dtype: object
```

Tip

Here we can see the int_rate is of type object but its a quantitative variable, so lets convert the type by remove removing %

Data Analysing

In this section we'll try to understand different features and how these features are related to each other. We'll also find out which variable has the most impact on our **target** variable which is loan_status

```
def set_plot_labels(xlabel, ylabel, title
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.title(title)
```

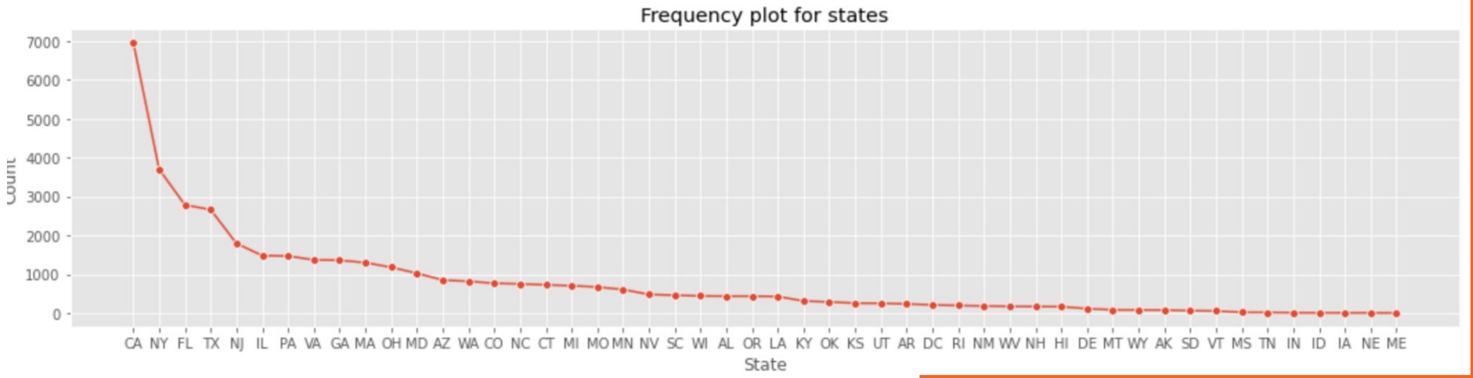
```
plt.style.use("ggplot")
```

Univariate

Let's perform univariate and do analysis of individual variable.

let's plot a frequency chart and see based on field addr_state which is an unordered variable.

```
plt.figure(figsize=(18,4))
sns.lineplot(data=loan.addr_state.value_counts(), marker="o")
set_plot_labels("State","Count","Frequency plot for states")
plt.plot()
```



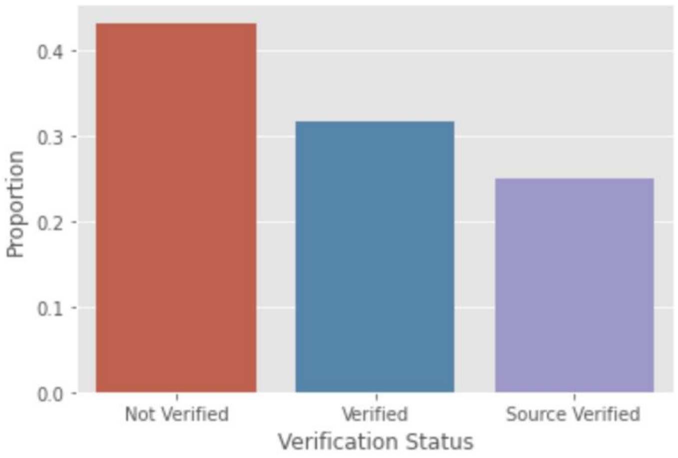
Tip

We can see that the california has the highest borrower followed by New York.

Data Analysing Cont...

Lets analyze few other variables

```
sns.barplot(data=loan.verification_status.value_counts(normalize=True).reset_index(), x = 'index', y='verification_status')
set_plot_labels("Verification Status", "Proportion")
plt.show()
```




```
loan.verification_status.value_counts(normalize=True)
```

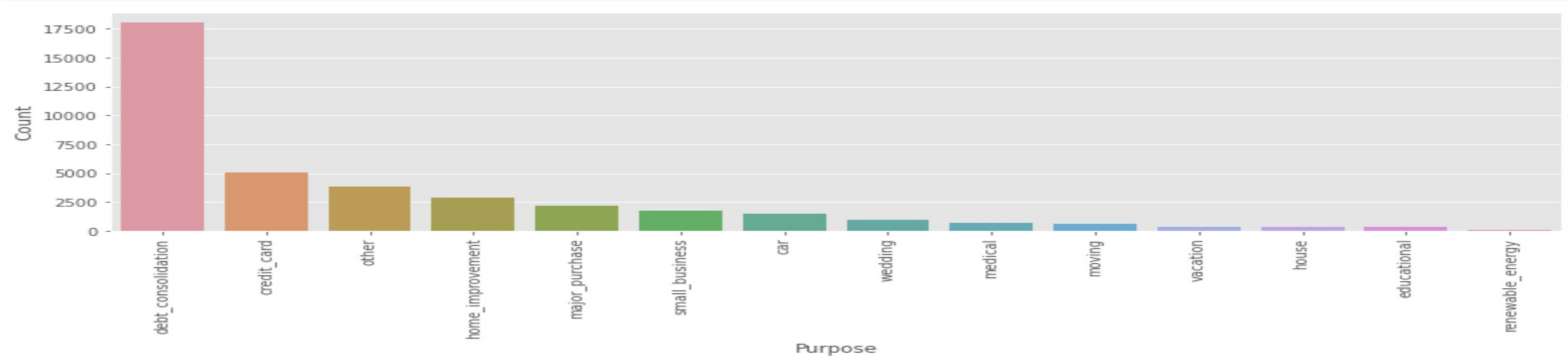
```
Not Verified       0.432745
Verified           0.316406
Source Verified    0.250849
Name: verification_status, dtype: float64
```

We can see the sum of Verified and source verified is coming to 56% and not verified is coming to 45%. Which is not really matching with 15% charged off, that means the unverified borrowed are not necessarily charging off

Data Analysing Cont...

More deeper analysis..

```
plt.figure(figsize=(15, 4))
sns.countplot(data=loan, x="purpose", order=loan.purpose.value_counts().index)
set_plot_labels("Purpose", "Count")
plt.xticks(rotation=90)
plt.show()
```



Looks like most of them are taking loan for *Debt Consolidation*

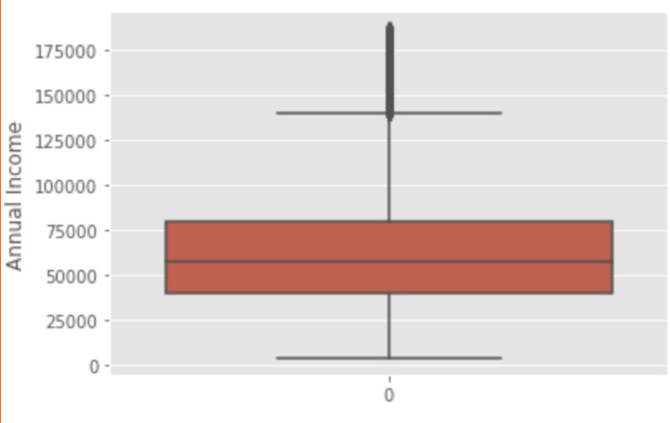
Data Analysing Cont...

Let's have a look at some quantitative variables like annual Income based chart.

```
sns.boxplot(data=loan.annual_inc)
plt.ylabel("Annual Income")
plt.show()
```

```
loan = loan[loan.annual_inc <= loan.annual_inc.quantile(0.98)]
```

```
sns.boxplot(data=loan.annual_inc)
plt.ylabel("Annual Income")
plt.show()
```




```
loan.annual_inc.describe()
```

```
count     37807.000000
mean      63882.857307
std       32295.149781
min        4000.000000
25%       40000.000000
50%       57600.000000
75%       80000.000000
max      187000.000000
Name: annual_inc, dtype: float64
```

We can see the mean annual income for the borrower are around 60,000

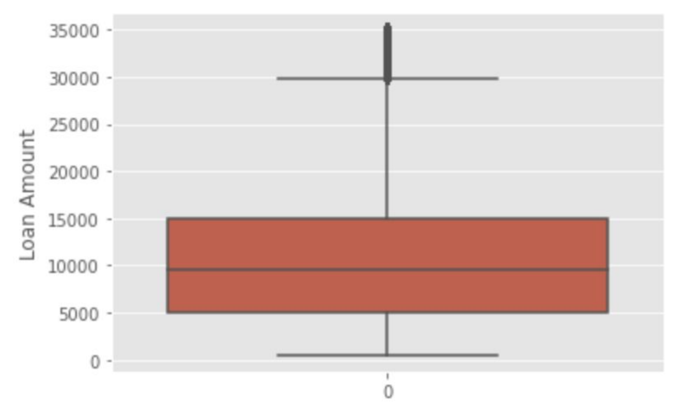
Data Analysing Cont...

Let’s have a look at the loan amount disbursed.

```
sns.boxplot(data=loan.loan_amnt)
plt.ylabel("Loan Amount")
plt.show()
```

```
loan.funded_amnt.describe()
```

```
count     37807.000000
mean      10631.478298
std        6956.351622
min         500.000000
25%        5000.000000
50%        9250.000000
75%       14750.000000
max       35000.000000
Name: funded_amnt, dtype: float64
```



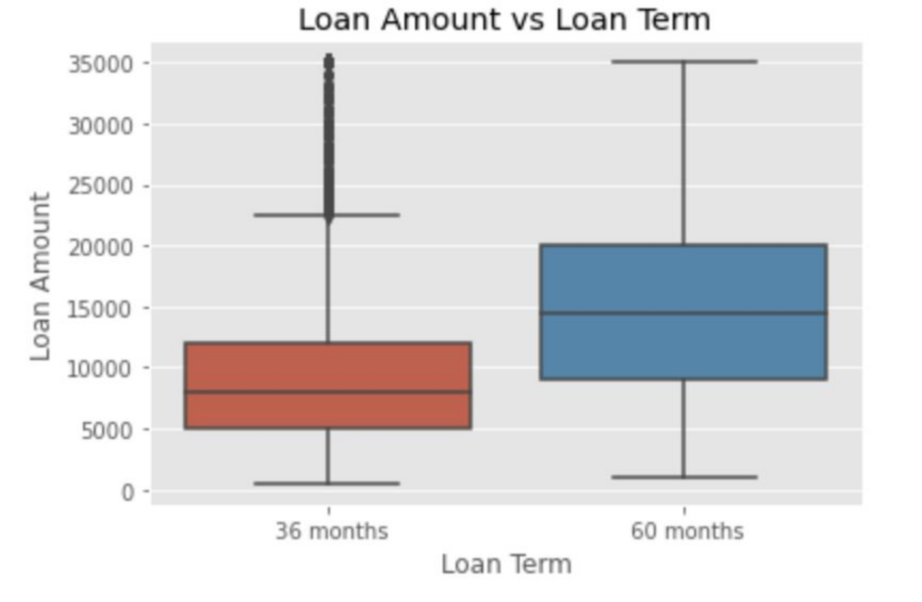
Average amount received to borrower is around 10,000

Multivariate Analysis.

Let's have a look at segmented and multivariate analysis to see a pattern

Lets see how term and loan amount are related

```
sns.boxplot(data=loan, x="term", y=loan.loan_amnt)
set_plot_labels("Loan Term", "Loan Amount", "Loan Amount vs Loan Term")
plt.show()
```



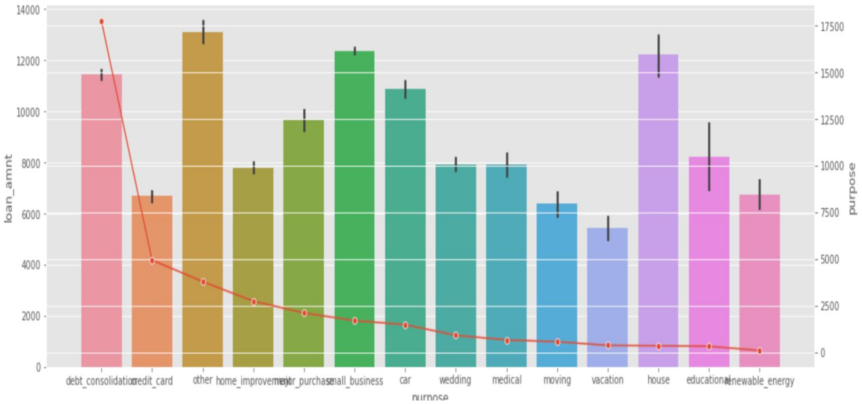
Here we can see that the Loan amount is more for higher term loan which make sense

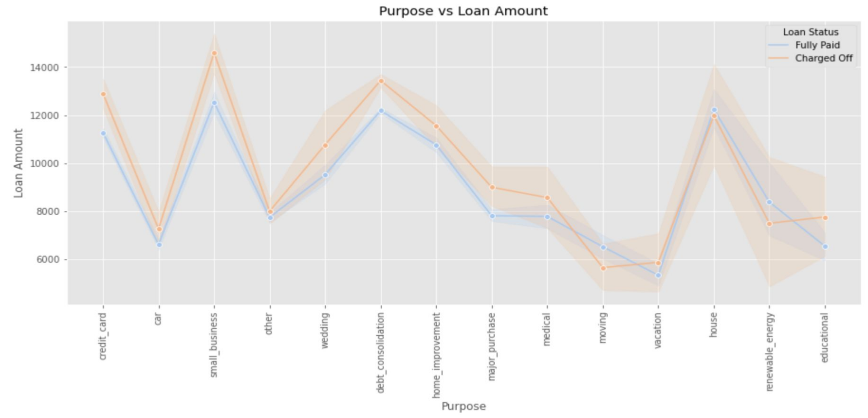
Multivariate Analysis.

Let's have a look at segmented and multivariate analysis to see a pattern

```
fig, ax1 = plt.subplots(figsize=(18,6))
sns.barplot(x="purpose", y="loan_amnt", data=loan, ax=ax1)
ax2 = ax1.twinx()
x=loan.purpose.value_counts().reset_index()
sns.lineplot(data=x, x="index", y="purpose", palette="pastel", ax=ax2, marker="o")
plt.show()
```

```
plt.figure(figsize=(15,6))
sns.lineplot(data =loan,y='loan_amnt', x='purpose', hue ='loan_status',palette="pastel", marker="o")
set_plot_labels("Purpose", "Loan Amount", "Purpose vs Loan Amount")
plt.xticks(rotation=90)
plt.legend(title="Loan Status")
plt.show()
```



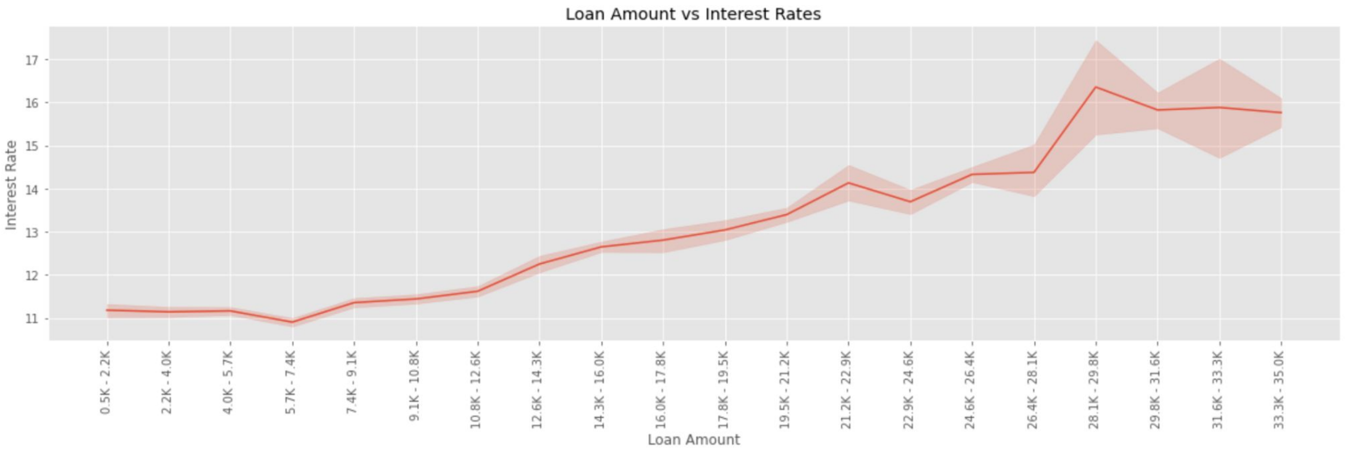


The number of loans for debt consolidation are higher plus the loan amount is also in the higher range so Higher loan amount is likely to charge of especially in case of small business and debt consolidation.

More Analysis.

Lets see how loan amount and interest rate are related

```
plt.figure(figsize=(20, 5))
sns.lineplot(y="int_rate", x="loan_amnt_bin", data=loan)
set_plot_labels("Loan Amount", "Interest Rate", "Loan Amount vs Interest Rates")
plt.xticks(rotation=90)
plt.show()
```

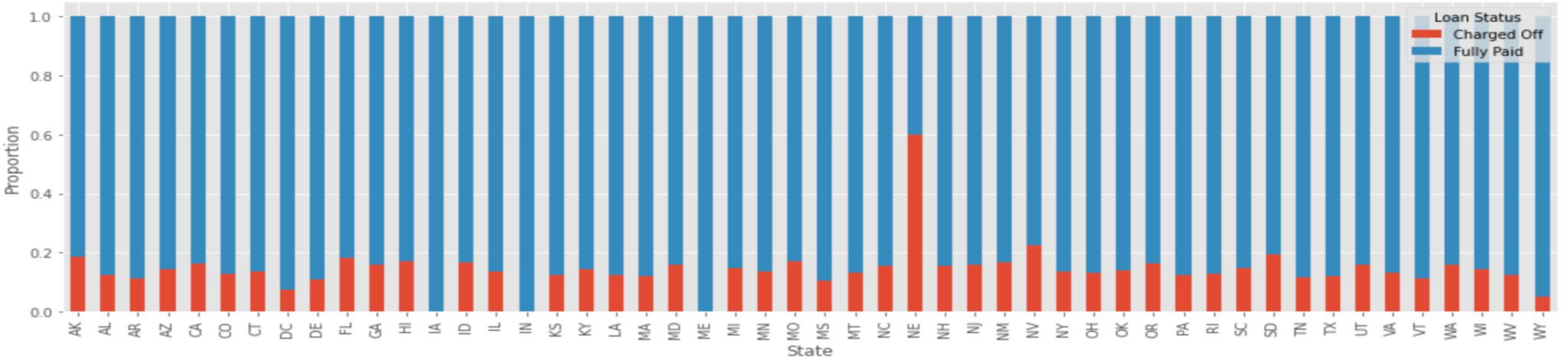


We can see increase in loan amount increase interest rate

More Analysis.

Lets see state wise loan status

```
state_vs_status = pd.crosstab(index=loan.addr_state, columns=loan.loan_status, normalize='index')
state_vs_status.plot(kind="bar", stacked=True, figsize=(18, 5), xlabel="State", ylabel="Proportion")
plt.legend(title="Loan Status")
plt.show()
```

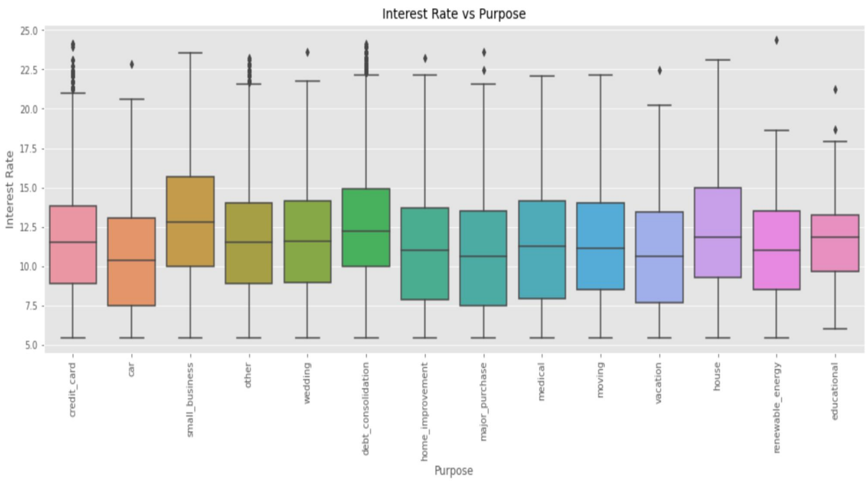


Althought proportion wise the state of NE has charged off the most, the count of the borrowers in NE is the least, so this value can be ignored so the state of the borrower does not influence the charge off.

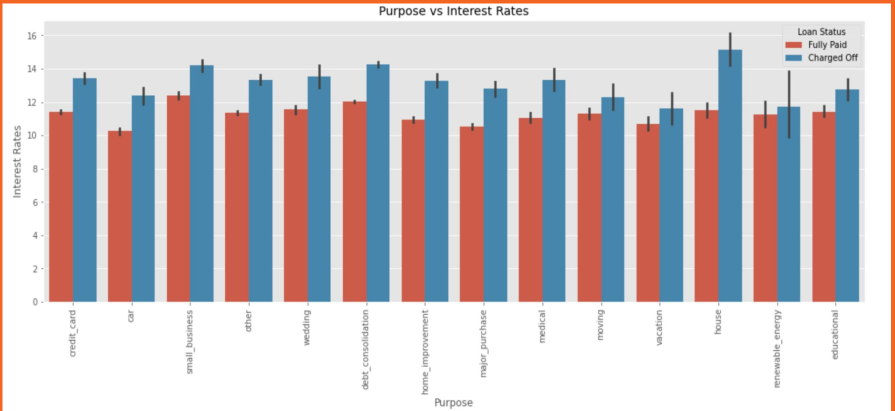
More Analysis.

Let's analys interest rate against purpose of the loan

```
plt.figure(figsize=(18, 6))
sns.boxplot(data=loan, x="purpose", y="int_rate")
set_plot_labels("Purpose", "Interest Rate", "Interest Rate vs Purpose")
plt.xticks(rotation=90)
plt.show()
```




```
plt.figure(figsize=(18, 6))
sns.barplot(data=loan, x="purpose", y="int_rate", hue="loan_status")
set_plot_labels("Purpose", "Interest Rates", "Purpose vs Interest Rates")
plt.legend(title="Loan Status")
plt.xticks(rotation=90)
plt.show()
```

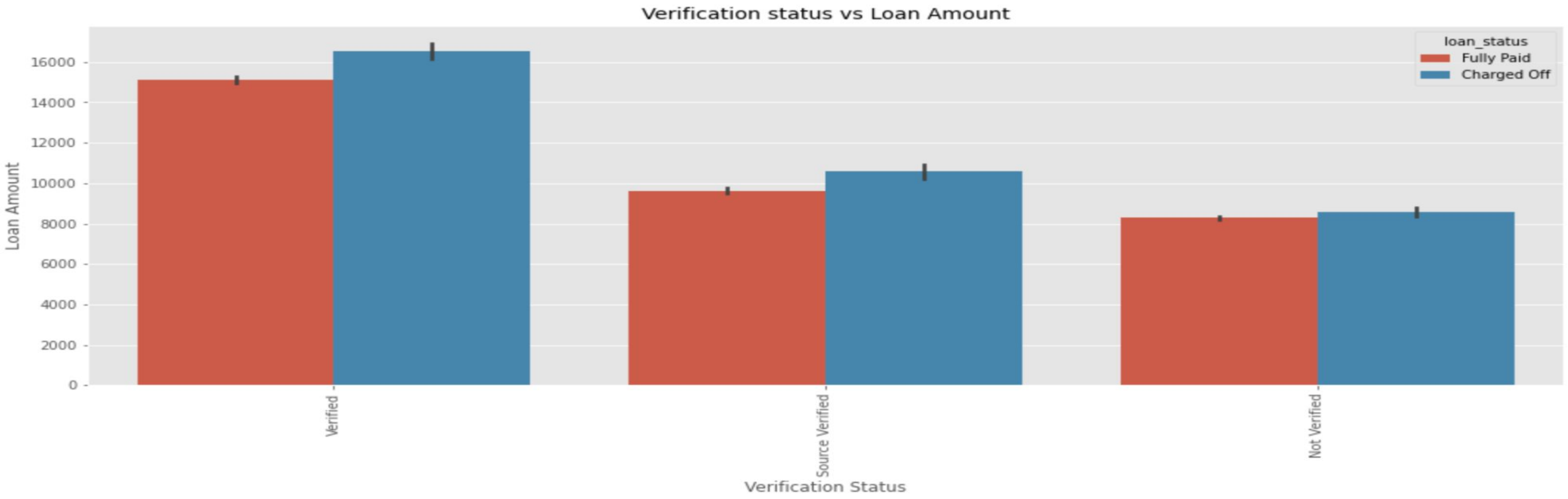


Although the average rate of interest for small_business is higher, when we drill down we realize that the average int_rate are higher in house category for those who charged off if a housing category is having higher rate of interest the borrower will likely charge off

More Analysis.

Let's have a look at how the verification status and loan amount is related to charge off

```
plt.figure(figsize=(18, 6))
sns.barplot(data=loan, x="verification_status", y="loan_amnt", hue="loan_status")
set_plot_labels("Verification Status", "Loan Amount", "Verification status vs Loan Amount")
plt.xticks(rotation=90)
plt.show()
```

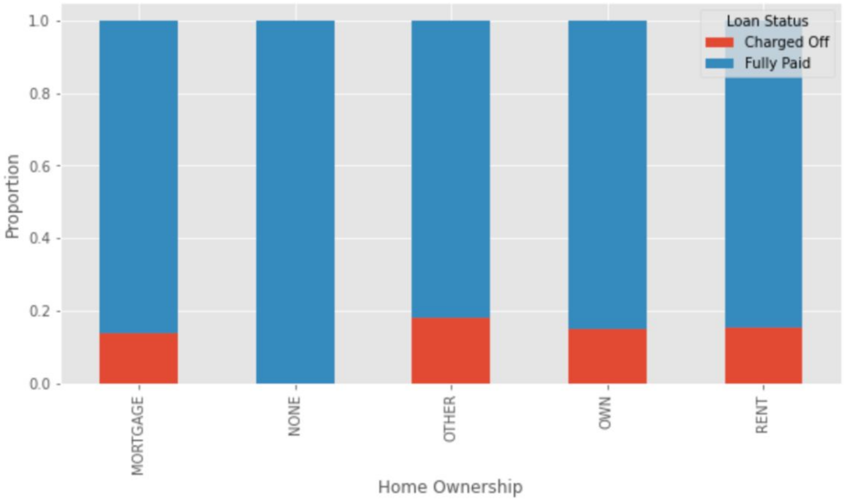


On an average if a loan amount is higher borrower is likely to charge-off, it does not depend whether borrower is verified or not

More Analysis.

Let's have a look how home_ownership are dependent on loan status

```
loan.groupby("home_ownership")["loan_status"].value_counts(normalize='index').unstack().plot(kind='bar',stacked=True, figsize=(10, 5), xla
plt.legend(title="Loan Status")
plt.show()
```

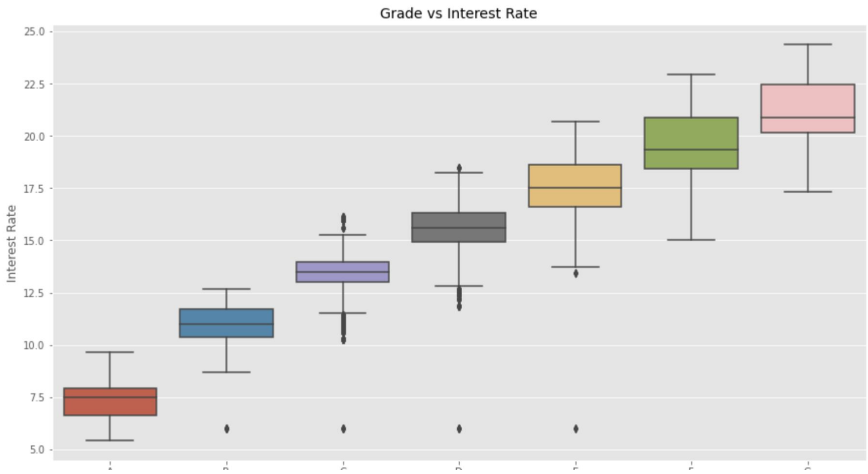


There seems to be no correlation between home_ownership and charge-offs

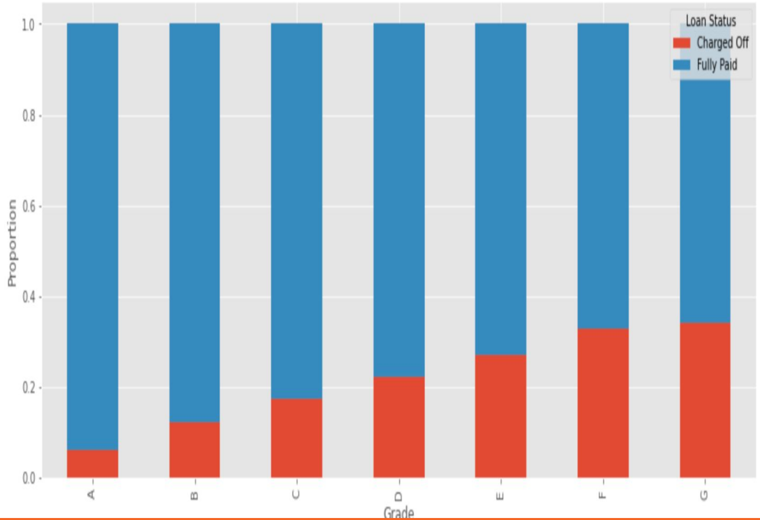
More Analysis.

Let’s analys how Grade influence interest rate

```
plt.figure(figsize=(15, 8))
order = loan.grade.unique()
order.sort()
sns.boxplot(x='grade', y="int_rate", order = order, data=loan)
set_plot_labels("Grade", "Interest Rate", "Grade vs Interest Rate")
plt.show()
```



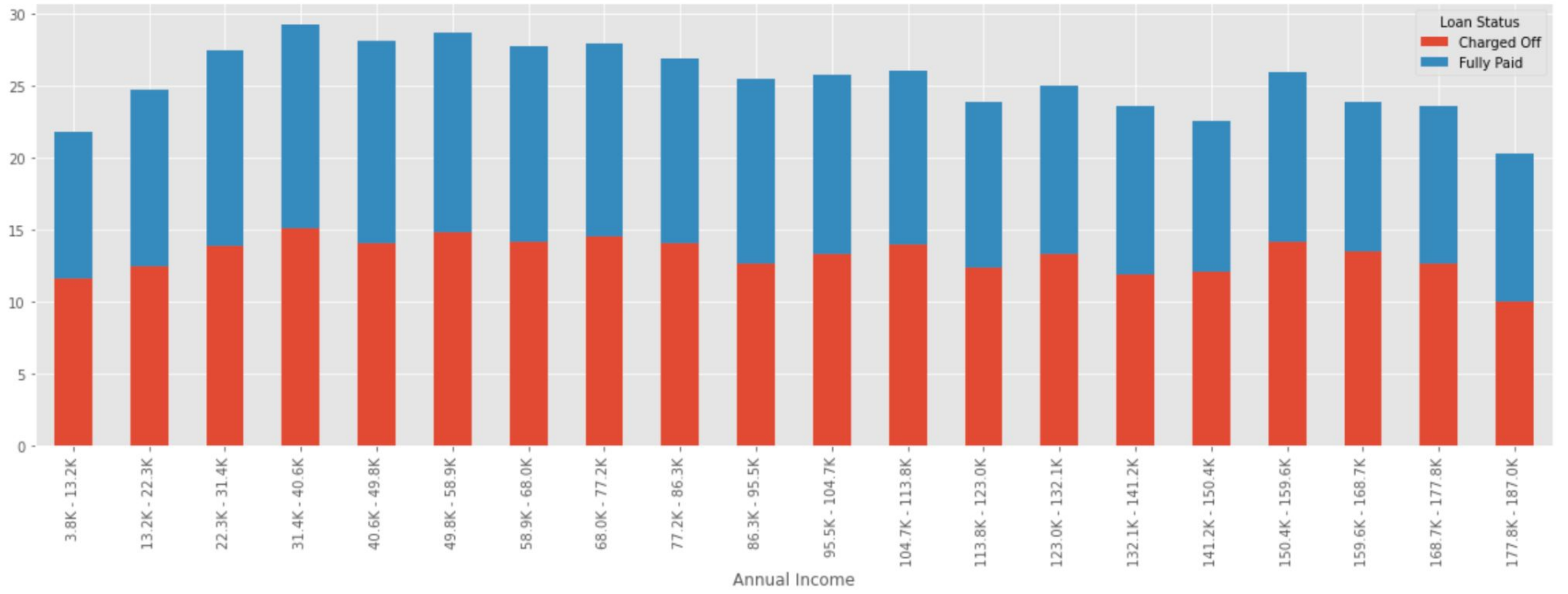

```
loan.groupby("grade")["loan_status"].value_counts(normalize=True).unstack().plot(kind='bar', stacked=True, figsize=(15, 6), xlabel="Grade'
plt.legend(title="Loan Status")
plt.show()
```



Loan with Lower grades generally has higher loan amount and are more likely to charge off

More Analysis.

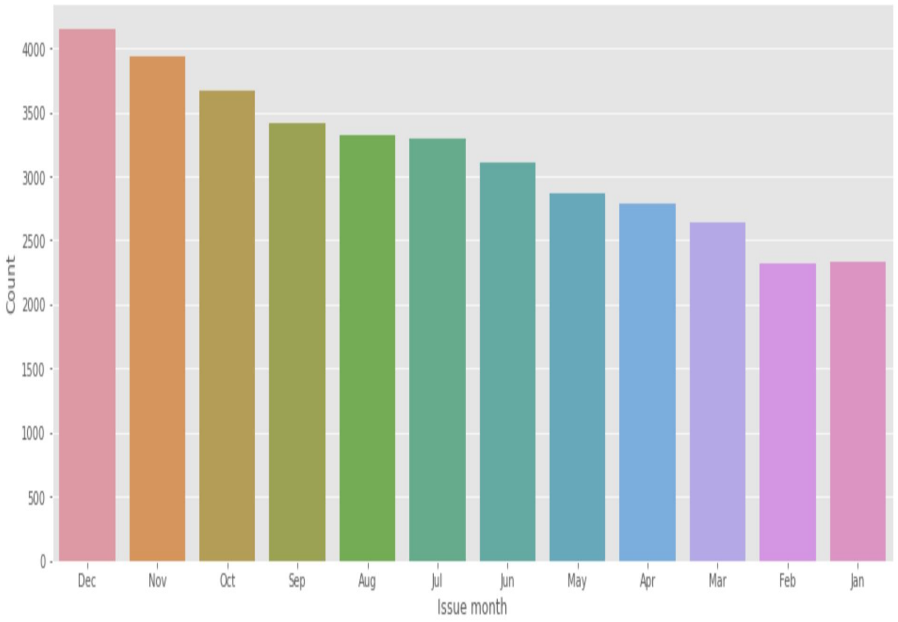
Lets analyze influence of annual income and DTI

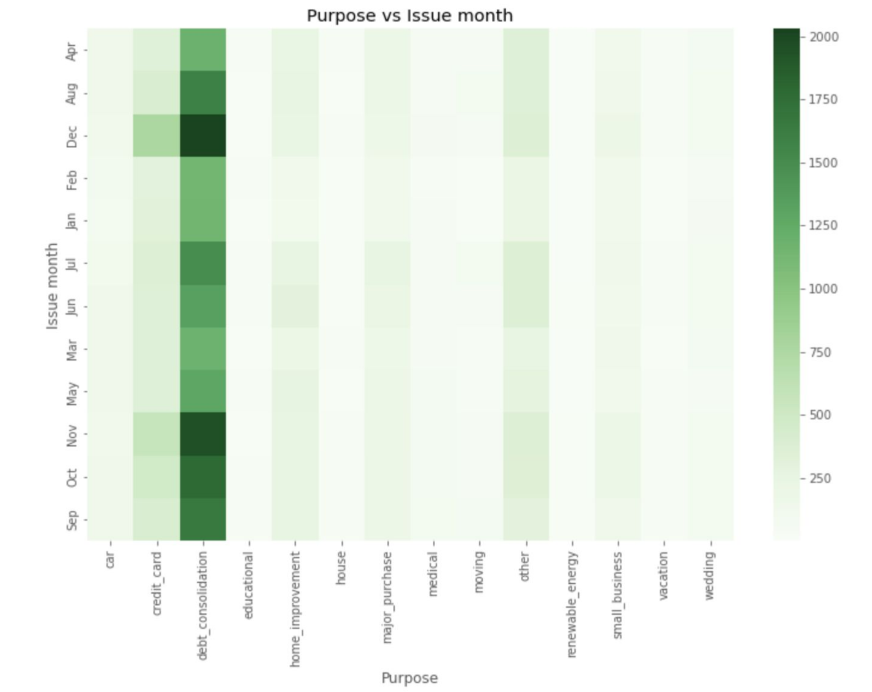


Borrowers with the annual income ranges from 31k to 40k have higher DTI rates and are like to charge off more

More Analysis.

Let's understand the trend in each month

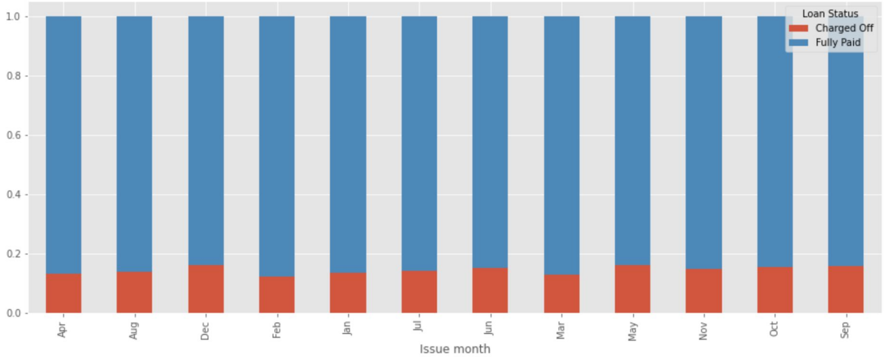




We can see in the month of Dec more loan is being taken, Here we can see that in month of Aug, Sept, Oct Nov, Dec borrowers are taking loan for debt_consolidation

More Analysis.

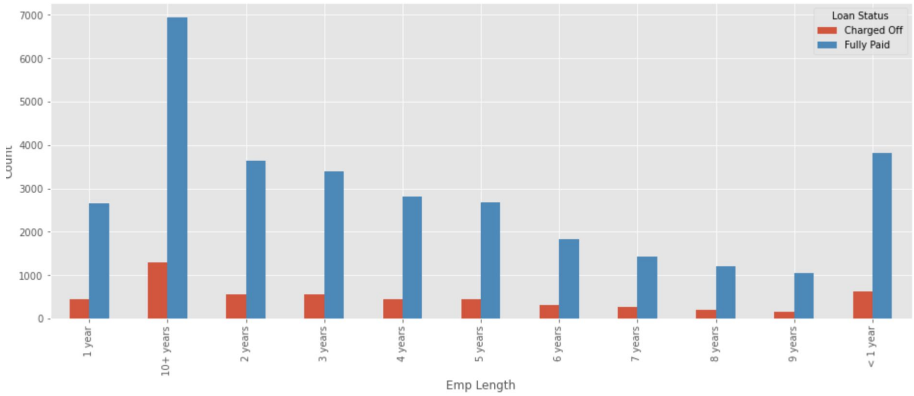
Let's understand the trend in each month

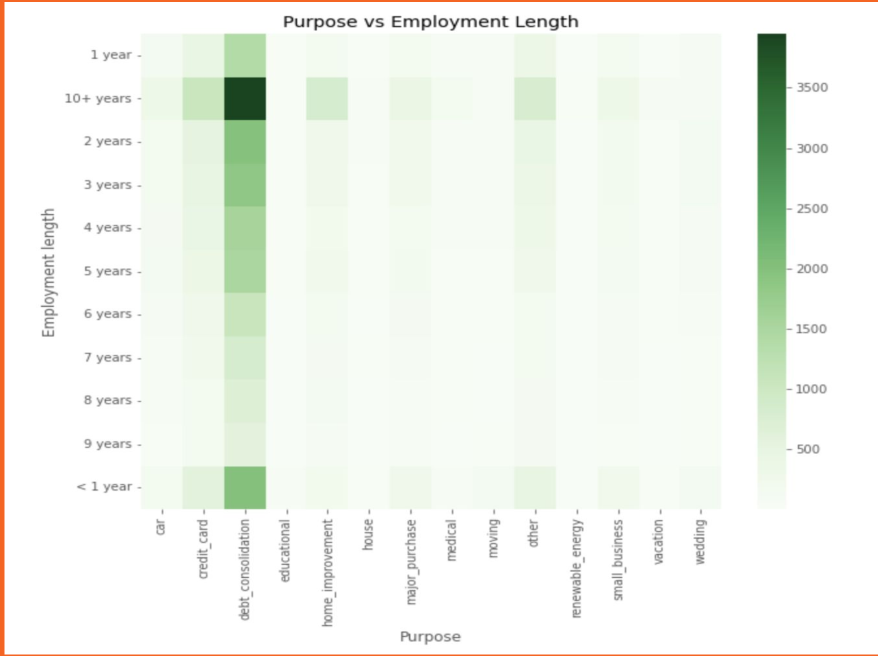


In month of Dec overall charge off is not significant,There's no significant increase in charge off to debt_consolidation in the last quarter

More Analysis.

Employee Length based analysis





Employees having exp greater than 10 year is borrowing the most and mainly for debt_consolidation

Observation and recommendation

Overall 15% of the borrower charges-off

Whether the source is verified or not it does not really impact the charge-offs

If a loan amount is higher the term is also more and there is a likely of charge-off

State of california has the highest borrower number. But overall charge-offs are independent of the state of the borrower. Although the state of Nebraska has higher percent of charge-off the number of borrowers are also very less, so it can be considered consequential

Most of the loans are taken for Debt Consolidation. Loan taken for debt consolidation and small businesses with higher loan amount are likely to charge off

Most of the loans are taken in the month of Oct, Nov and Dec again mainly for Debt Consolidation, but there is no unusal increase in charge during this period for Debt Consolidation purpose

Observation and recommendation

Verification status of the borrower does not guarantee that there will be no charge-offs

The rate of interest increases with the loan amount

Interest rates are higher for small business.

Average annual income of the borrowers are around 60k. Borrower with annual income ranges from 30k to 40k have higher DTI and are more likely to charge-off

If a borrower is taking a loan for housing and rate of interest is higher then the borrower is likely to charge-off

Loan with Lower grades generally has higher loan amount and are more likely to charge off

Employees having exp greater than 10 year is borrowing the most and mainly for debt_consolidation

Loan amount requested are based on annual income which make sense

Thank You !