Upgrad
Lending Club Case Study

FROM
SWATI GUPTA (GSW68521@GMAIL.COM)
DEEPAK CHANDRU (DEEPAKC1133@GMAIL.COM)

Problem Statement

Lending loans to ‘risky’ applicants is the largest source of financial loss(called credit loss). The credit loss is the amount of money lost by the lender when the borrower refusesto pay or runs away with the money owed. The main objective is to be able to identify these risky loan applicants, then such loans can be reduced thereby cutting down the amount of credit loss. Identification of such applicants using EDA is the aim of this case study. Perform an analysis to understand the driving factors (or driver variables)behind loan default, i.e.the variables which are strong indicators of default. The company can utilise this knowledge for its portfolio and risk assessment.

Analysis

The above analysis with respect to the charged off loans. There is a more probability of defaulting when :

•Applicants taking loan for 'home improvement' and have income of 60k -70k
•Applicants whose home ownership is 'MORTGAGE and have income of 60-70k
•Applicants who receive interest at the rate of 21-24% and have an income of 70k-80k
•Applicants who have taken a loan in the range 30k - 35k and are charged interest rate of 15-17.5 %
•Applicants who have taken a loan for small business and the loan amount is greater than 14k
•Applicants whose home ownership is 'MORTGAGE and have loan of 14-16k
•When grade is F and loan amount is between 15k-20k
•When employment length is 10yrs and loan amount is 12k-14k
•When the loan is verified and loan amount is above 16k
•For grade G and interest rate above 20%

jupyter SwatiGupta_Deepak Last Checkpoint: 13 minutes ago

File Edit View Run Kernel Settings Help Trusted

Markdown JupyterLab Python 3 (ipykernel)

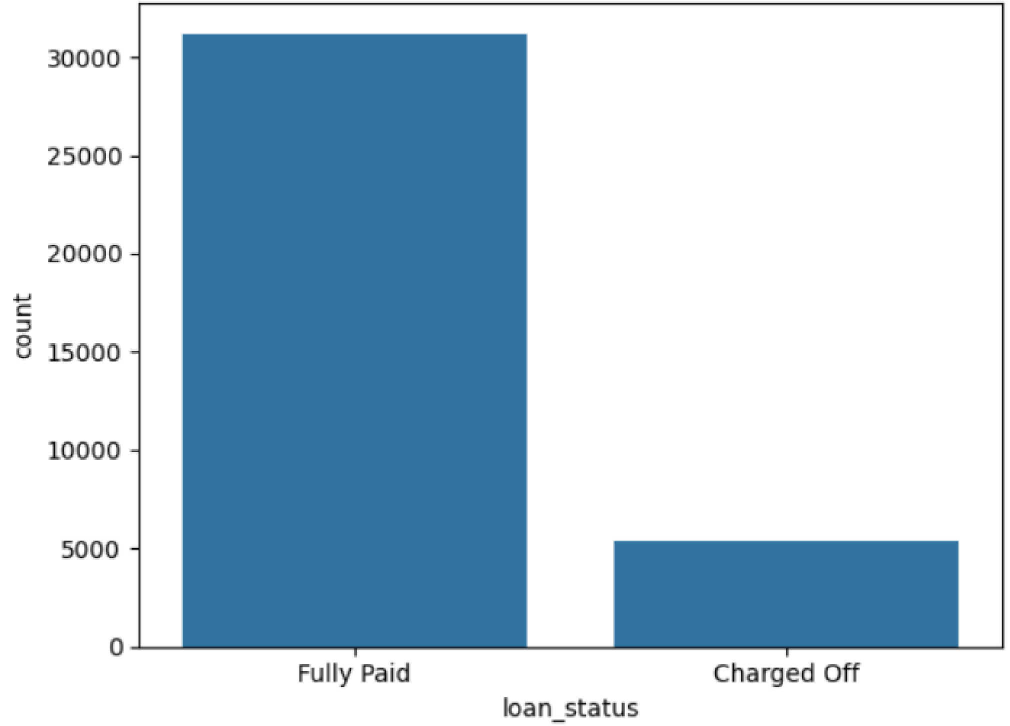
Visualizing Categorical Data

As we already have grade column, extracting only subgrade (int level value) from the sub_grade variable

- We are analyzing and visualizing only the defaulter data. So subsetting the data while plotting only for 'Charged Off' loan_status for below plots

[40]:
```
sns.countplot(x = 'loan_status', data = loan_data)
```

[40]:
```
<Axes: xlabel='loan_status', ylabel='count'>
```



jupyter SwatiGupta_Deepak Last Checkpoint: 13 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Markdown

JupyterLab Python 3 (ipykernel)

Analyzing home_ownership

[44]:
```
#checking unique values for home_ownership
loan_data['home_ownership'].unique()
```

[44]:
```
array(['RENT', 'OWN', 'MORTGAGE', 'OTHER', 'NONE'], dtype=object)
```

There are only 3 records with 'NONE' value in the data. So replacing the value with 'OTHER'

[45]:
```
#replacing 'NONE' with 'OTHERS'
loan_data['home_ownership'].replace(to_replace = ['NONE'],value='OTHER',inplace = True)
```

[46]:
```
#checking unique values for home_ownership again
loan_data['home_ownership'].unique()
```

[46]:
```
array(['RENT', 'OWN', 'MORTGAGE', 'OTHER'], dtype=object)
```

[47]:
```
fig, ax = plt.subplots(figsize = (6,4))
ax.set(yscale = 'log')
sns.countplot(x='home_ownership', data=loan_data[loan_data['loan_status']=='Charged Off'])
```

Analyzing purpose

[48]:
```
fig, ax = plt.subplots(figsize = (12,8))
ax.set(xscale = 'log')
sns.countplot(y ='purpose', data=loan_data[loan_data.loan_status == 'Charged Off'])
```

[48]: <Axes: xlabel='count', ylabel='purpose'>

jupyter SwatiGupta_Deepak Last Checkpoint: 15 minutes ago

File Edit View Run Kernel Settings Help

Trusted

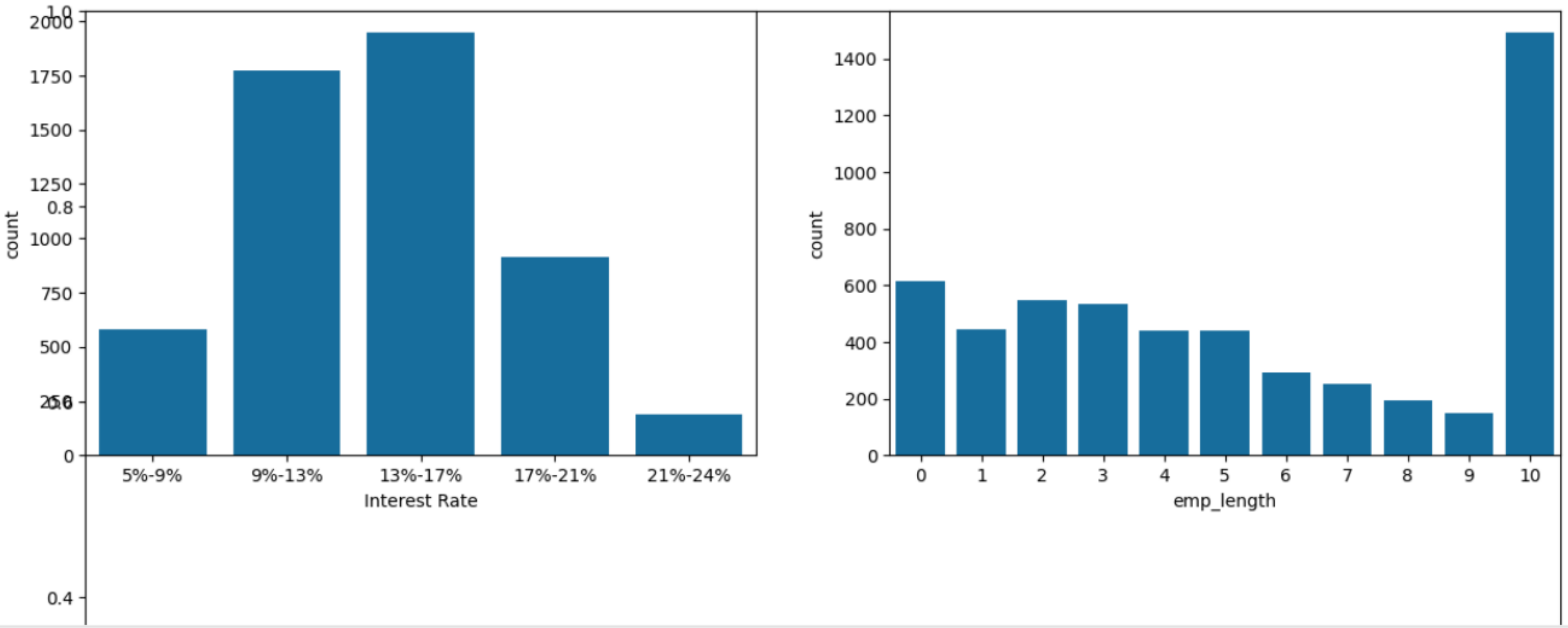
Markdown

JupyterLab Python 3 (ipykernel)

Analyzing interest rate wrt the interest rate bins created

[51]:
```
fig, ax = plt.subplots(figsize = (15,10))
plt.subplot(221)
sns.countplot(x='int_rate_groups', data=loan_data[loan_data.loan_status == 'Charged Off'])
plt.xlabel('Interest Rate')
plt.subplot(222)
sns.countplot(x='emp_length', data=loan_data[loan_data.loan_status == 'Charged Off'])
```

[51]:
```
<Axes: xlabel='emp_length', ylabel='count'>
```



jupyter SwatiGupta_Deepak Last Checkpoint: 15 minutes ago

File Edit View Run Kernel Settings Help

Trusted

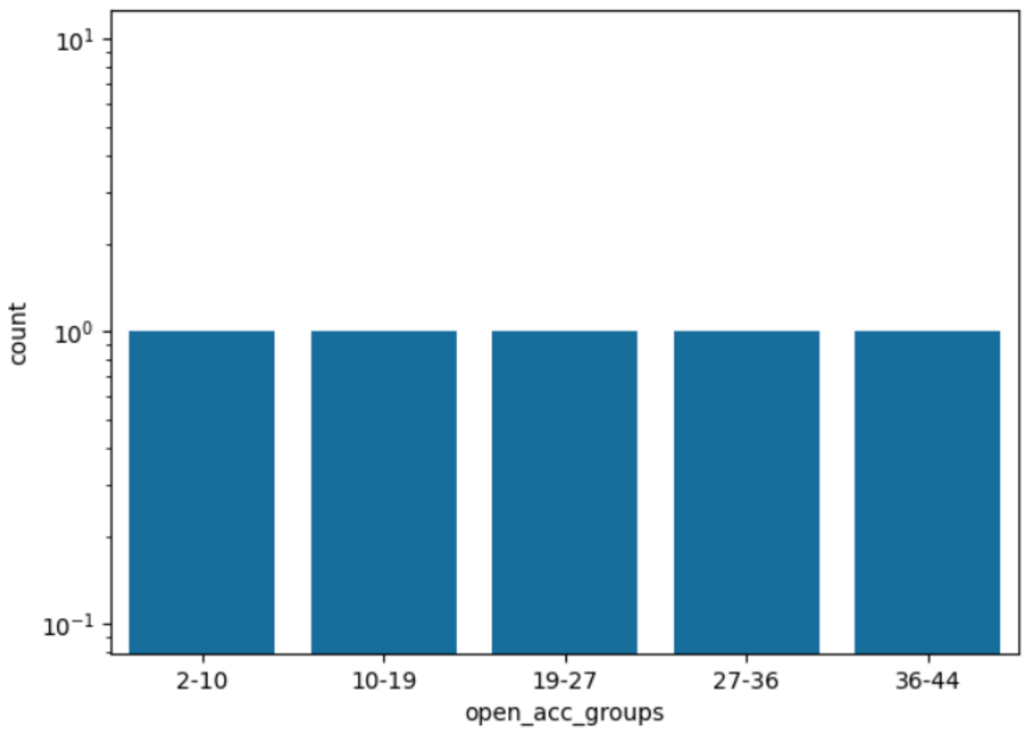
Markdown

JupyterLab Python 3 (ipykernel)

Similarly analyzing open_acc,revol_util,total_acc,annual_inc

[52]:
```
fig, ax = plt.subplots(figsize = (7,5))
ax.set_yscale('log')
sns.countplot(x='open_acc_groups', data=loan_data[loan_data.loan_status == 'Charged Off'])
```

[52]: <Axes: xlabel='open_acc_groups', ylabel='count'>



jupyter SwatiGupta_Deepak Last Checkpoint: 15 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Markdown

JupyterLab Python 3 (ipykernel)

Observations ¶

The above analysis with respect to the charged off loans for each variable suggests the following. There is a more probability of defaulting when :

- Applicants having house_ownership as 'RENT'
- Applicants who use the loan to clear other debts
- Applicants who receive interest at the rate of 13-17%
- Applicants who have an income of range 31201 - 58402
- Applicants who have 20-37 open_acc
- Applicants with employement length of 10
- When funded amount by investor is between 5000-10000
- Loan amount is between 5429 - 10357
- Dti is between 12-18
- When monthly installments are between 145-274
- Term of 36 months
- When the loan status is Not verified
- When the no of enquiries in last 6 months is 0
- When the number of derogatory public records is 0
- When the purpose is 'debt_consolidation'
- Grade is 'B'
- And a total grade of 'B5' level.

Also there is a very interesting observation from the date issued. The late months of an year indicated the high

jupyter SwatiGupta_Deepak Last Checkpoint: 16 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Markdown

JupyterLab Python 3 (ipykernel)

Also there is a very interesting observation from the date issued. The late months of an year indicated the high possibility of defaulting.

- The high number of loan defaults in 2011 could be due to the financial crisis in USA (Assuming the data is of US origin)

Analysing annual income with other columns for more insights

1.Annual income vs loan purpose

[68]:

```
plt.figure(figsize=(10,10))
sns.barplot(data =loan_data,x='annual_inc', y='purpose', hue ='loan_status',palette="deep")
plt.show()
```