**Data analysis assessment**

**Dataset-**[Data\_Analyst\_Assignment\_Dataset.csv](https://prod-files-secure.s3.us-west-2.amazonaws.com/0fb337c8-6186-4010-911c-38ba2e525070/7e55b4b1-5878-4d1c-9dce-75419c39c4c5/Data_Analyst_Assignment_Dataset.csv)

|  |  |
| --- | --- |
| **Column**  **Name** | **Description** |
| Amount  Pending | This is the EMI amount pending. |
| State | The borrower’s state. |
| Tenure | Total tenure of the borrower. |
| Interest  Rate | Interest rate of the loan. |
| City | The city of the borrower. |
| Bounce  String | This is a string that explain’s customer’s bounce behaviour since the disbursal of the loan - bounce means that the customer did not end up making the payment • S or H- No bounce in that month • B or L - Bounce in that month • FEMI - first EMI - no known behaviour • Last character denotes the last month - first character denotes the first month on book - for example SSB means that customer was on book for 4 months and he has bounced the in the last month |
| Disbursed  Amount | The total disbursed amount of the loan. |
| Loan  Number | The unique identifier for the loan. |

**Tech Stack use:**

Kaggle Jupiter notebook **Hyperlinking notebook**

[**https://drive.google.com/file/d/1bMh8YmfTy4GRH5hsAvK8PCr1AiIoC4d/view?usp=sharing**](https://drive.google.com/file/d/1bMh8YmfTy4GRH5hsAvK8PCr1-AiIoC4d/view?usp=sharing)

import numpy as np import pandas as pd import os for dirname, \_, filenames in os.walk('/kaggle/input'): for filename in filenames:

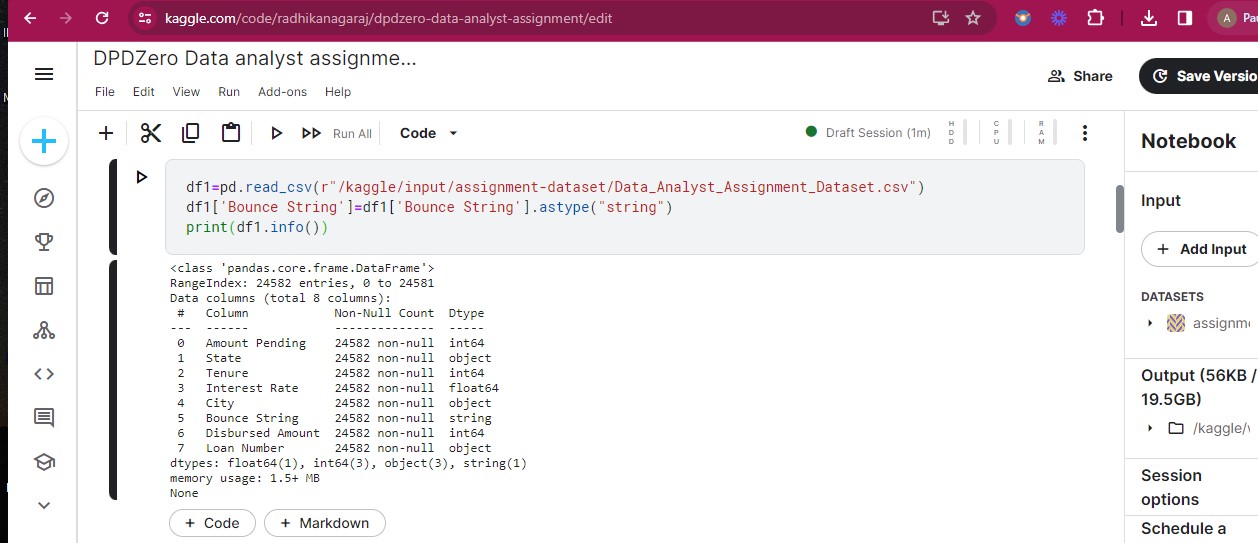
print(os.path.join(dirname, filename))

# This python script is set up to work in a Kaggle environment, which provides a convenient platform for data analysis and necessary libraries are imported to support analysis on a dataframe. df1=pd.read\_csv(r"/kaggle/input/assignmentdataset/Data\_Analyst\_Assignment\_Dataset.csv

")

df1['Bounce String']=df1['Bounce String'].astype("string") print(df1.info())

# Read data into the dataframe variable and start analysis



**Insights**

# Initial Data Cleaning

**Handling Empty and duplicate values:** No duplicate or missing values in the dataset **Data manipulation:** we create new calculated columns as required for analysis of key metrics.

If Tenure column is considered to be in years then the monthy EMI is as follows

df1.loc[df1['Interest Rate']==0,'Monthly\_EMI']=(df1['Disbursed

Amount']/df1['Tenure(months)']).round(2)

df1.loc[df1['Interest Rate']!=0,'Monthly\_EMI']=((df1['Disbursed Amount']\*((df1['Interest

Rate']/12)/100)\*((1+((df1['Interest

Rate']/12)/100))\*\*df1['Tenure(months)']))/(((1+((df1['Interest Rate']/12)/100))\*\*df1['Tenure(months)'])-1)).round(2)

df1['Balance Tenure']=(df1['Amount Pending']/df1['Monthly\_EMI']).round()

# Task 1: Risk Labels for all borrowers

|  |  |
| --- | --- |
| Unknown risk | New customers |
| Low risk | Customers who have not bounced in the last 6 months |
| Medium Risk | These are customers who have bounced max twice in the last 6 months - The bounce should not have occurred in the last month |
| High risk | every other customer |

Based on the Bounce String column we check for customer bounce behaviour and assign labels accordingly

def risk\_label(a): if a=="FEMI":

return 'unknown risk'

elif (a[6:].count('B')+a[-6:].count('L'))==0:

return 'low risk'

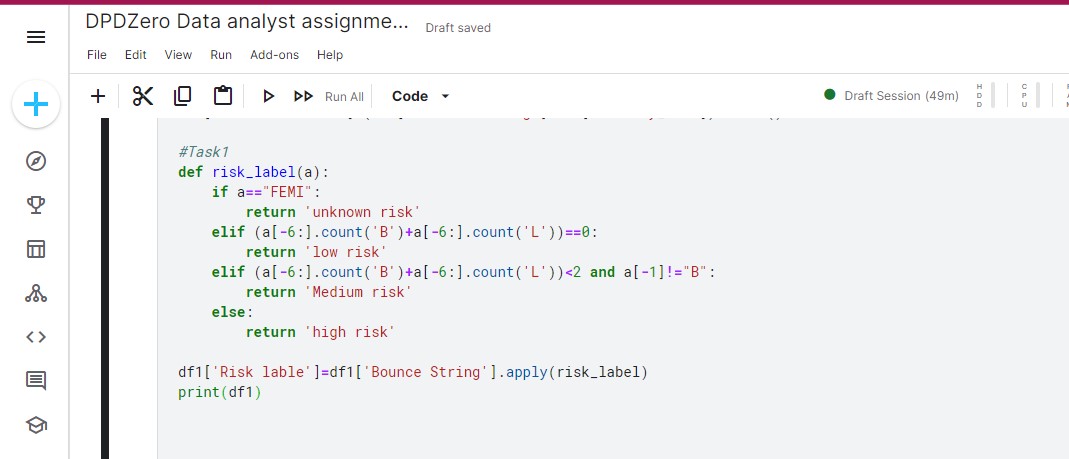
elif (a[-6:].count('B')+a[-6:].count('L'))<2 and a[-1]!="B":

return 'Medium risk' else:

return 'high risk'

df1['Risk lable']=df1['Bounce String'].apply(risk\_label)

print(df1)



# Plotting the summary of borrowers based on risk Import matplotlib.pyplot as plt

risk\_counts = df1['Risk lable'].value\_counts() risk\_counts.plot(kind='bar', color='skyblue')

plt.title('Summary of Borrowers Based on Risk')

plt.xlabel('Risk lable')

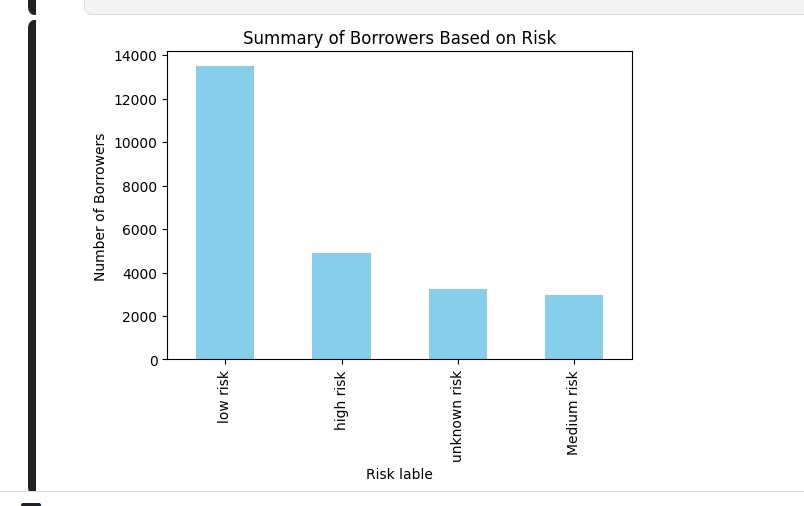
plt.ylabel('Number of Borrowers') plt.show()

Significant portion of borrowers fall into the low-risk category, it suggests that considerable

number of borrowers have exhibited good repayment behaviour

Moderate portion of borrowers fall into the medium-risk category may indicate some inconsistency I repayment behaviour or occasional instances of bouncing

High portion of borrowers fall into the high-risk category implying notable portion of borrowers have demonstrated poor repayment behaviour or frequent instances of bouncing Indicating the need for targeted interventions to mitigate potential defaults.



# Task 2: Tenure Labels for all borrowers

|  |  |
| --- | --- |
| Early tenure | Customers who are in the book for 3 months |
| Late tenure | Customers who are 3 months away from closing the loan |
| Mid tenure | Everyone else |

def Tenure\_label(a):

if len(a['Bounce String']) == 2 or a['Bounce String']=='FEMI' :

return 'early'

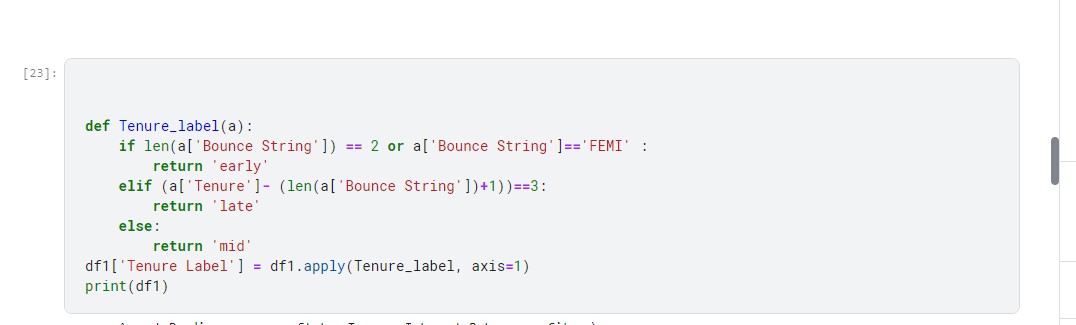
elif (a['Tenure']- (len(a['Bounce String'])+1))==3:

return 'late' else:

return 'mid'

df1['Tenure Label'] = df1.apply(Tenure\_label, axis=1)

print(df1)



tenure\_counts = df1['Tenure Label'].value\_counts()

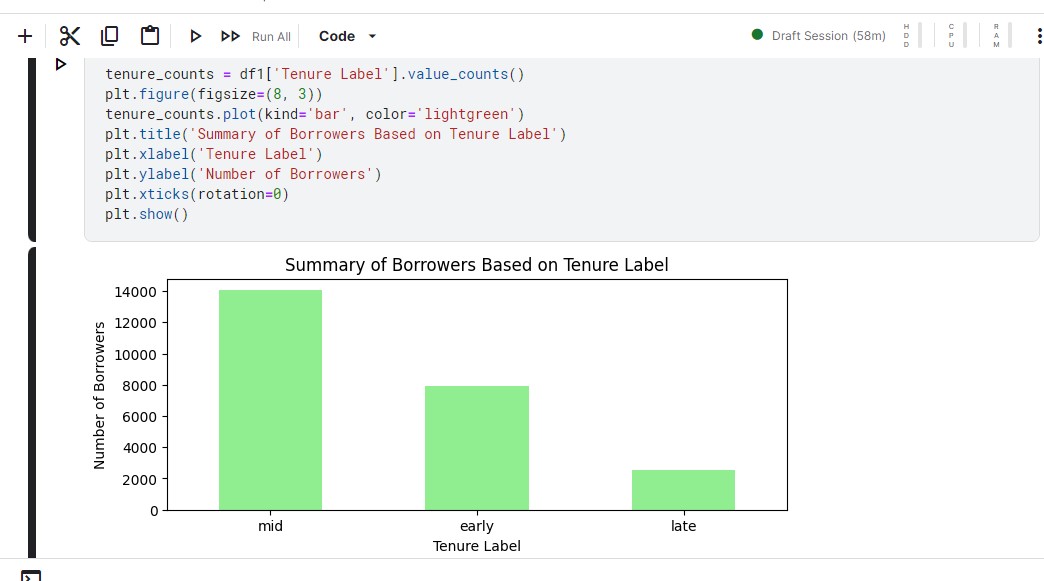
plt.figure(figsize=(8, 3))

tenure\_counts.plot(kind='bar', color='lightgreen') plt.title('Summary of Borrowers Based on Tenure Label') plt.xlabel('Tenure Label')

plt.ylabel('Number of Borrowers')

plt.xticks(rotation=0)

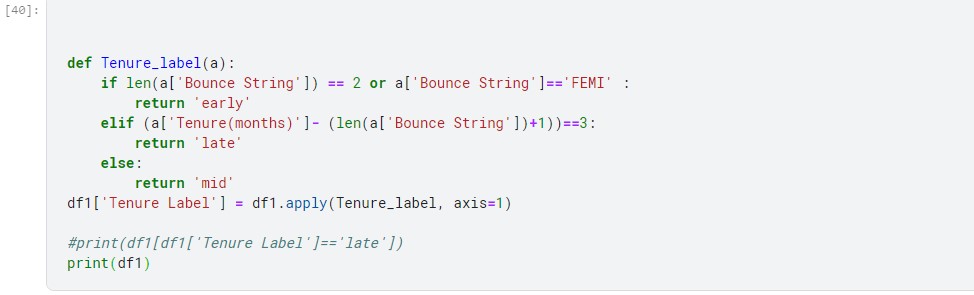
plt.show()

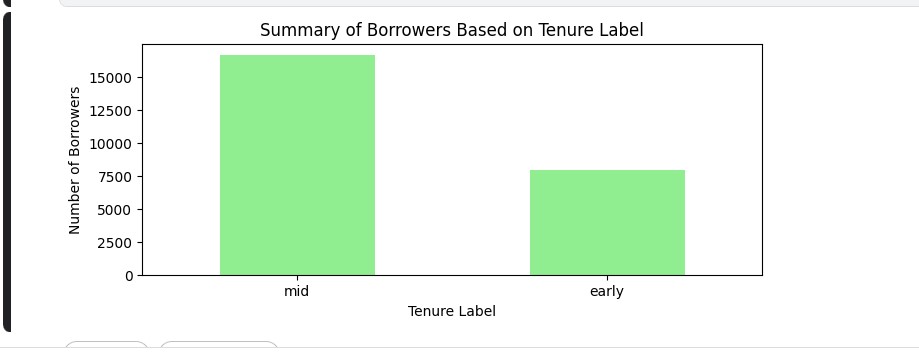


majority of borrowers are categorized as 'mid', followed by 'late' and 'early'.

There seems to be a relatively small number of borrowers categorized as 'late', indicating that fewer borrowers have a longer tenure length and are about to close their loan in 3 months.

Considering the column Tenure is the duration in months, above is the result, if the tenure is given in years then we have to convert the tenure into months by (Tenure\*12), which gives a different result, which we can see below, as per the result the ‘late’ category is vanished, indicating there are no borrowers that are 3 months away from closing their loans.





# Task 3: Ticket sizes Labels for all borrowers

Distribute the data into 3 cohorts based on ticket size. This is to be done such that sum of amount pending in each cohort should be approximately equal. Apply the following labels on each borrower based on this logic:

1. Low ticket size

2. Medium ticket size

3. High ticket size

Note that at the end of this exercise you would have a lot of folks with low ticket size and a few people in high ticket sizes - sum of amount pending for all these cohorts should be approximately equal

df1['Total Amount Pending'] = df1.groupby('Loan Number')['Amount Pending'].transform('sum')

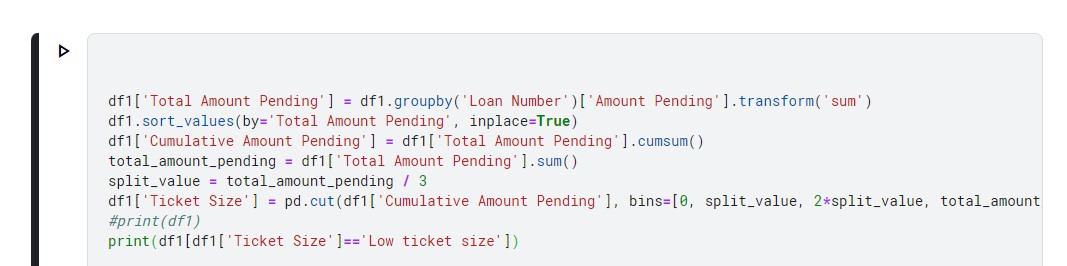
df1.sort\_values(by='Total Amount Pending', inplace=True)

df1['Cumulative Amount Pending'] = df1['Total Amount Pending'].cumsum() total\_amount\_pending = df1['Total Amount Pending'].sum()

split\_value = total\_amount\_pending / 3

df1['Ticket Size'] = pd.cut(df1['Cumulative Amount Pending'], bins=[0, split\_value, 2\*split\_value, total\_amount\_pending], labels=['Low ticket size', 'Medium ticket size', 'High ticket size'])

print(df1)



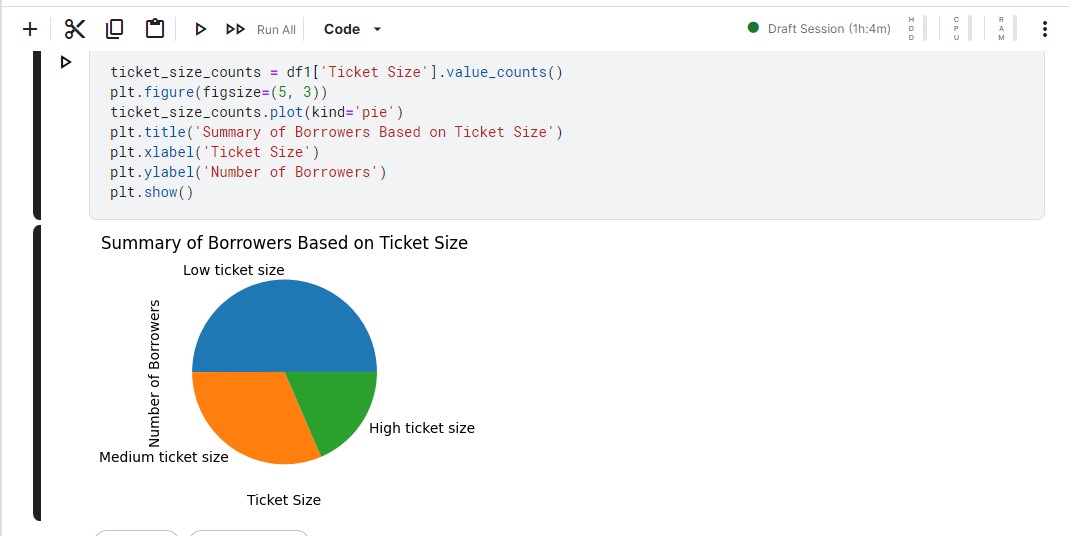
ticket\_size\_counts = df1['Ticket Size'].value\_counts() plt.figure(figsize=(5, 3))

ticket\_size\_counts.plot(kind='pie')

plt.title('Summary of Borrowers Based on Ticket Size') plt.xlabel('Ticket Size')

plt.ylabel('Number of Borrowers')

plt.show()



Based on the output graph, it appears that a significant portion of borrowers fall under the

“lower ticket size” indicating they have lower pending amounts to repay, followed by “Medium ticket size” and “high ticket size” with highest pending amounts, that may require closer attention or management strategies.

# Task 4: Segmenting borrowers to minimize total spend

# You are allowed to spend 3 kinds of resources to reduce the overall bounce

1. Whatsapp bot: This is the cheapest medium - it will cost 5 rupees per borrower
2. Voice bot: This is the mid-cost - it will cost 10 rupees per borrower
3. Human calling: This is the costliest option - it will cost 50 rupees per borrower

Whatsapp bot will work well in any of the following scenarios

1. Customers with great repayment behavior
2. Customers with first EMIs
3. Customers who have low EMIs

Voice bot will work well all the following conditions are met

1. Customer who know Hindi or English
   1. Metropolitan areas have high probability of english speakers
   2. People with low interest rates are also typically english speakers
   3. There are many states in India where the borrowers typically know Hindi
2. Customers who have had low bounce behaviour
3. Customers with low or medium sized EMIs

Human calling will work on all scenarios but is the costliest option and you need to use this channel only where absolutely necessary

Your job is to segment the borrowers into these 3 channels of spend category and minimise the overall spend while maximise on time repayment.

def spend(a):

if (a['Bounce String'].count('B')+a['Bounce String'].count('L'))==0 or a['Bounce String']=="FEMI" or a['Risk lable']=='low risk'or **a['Ticket Size']=='Low ticket size'**: return 'watsappbot'

elif a['Risk lable']=='Medium risk'or a['Ticket Size']=='Medium ticket size':

return 'voicebot' else:

return 'human calling'

df1['spend']=df1.apply(spend,axis=1)

print(df1)

channel\_counts = df1['spend'].value\_counts()

**Calculate the total cost for each channel**

whatsapp\_cost = channel\_counts['watsappbot'] \* 5 voice\_cost = channel\_counts['voicebot'] \* 10

human\_cost = channel\_counts['human calling'] \* 50

total\_cost = whatsapp\_cost + voice\_cost + human\_cost print(total\_cost)

The total cost to the company is **173490** rupees if the condition of low ticket size(low EMI) is considered for “watsapp bot” segment



def spend(a):

if (a['Bounce String'].count('B')+a['Bounce String'].count('L'))==0 or a['Bounce String']=="FEMI" or a['Risk lable']=='low risk':

return 'watsappbot'

elif a['Risk lable']=='Medium risk'or a['Ticket Size']=='Medium ticket size' or **a['Ticket Size']=='Low ticket size'**:

return 'voicebot' else:

return 'human calling'

df1['spend']=df1.apply(spend,axis=1)

The total cost to the company is **193890** rupees if the condition of low ticket size is considered for “voice bot” segment



The total cost to the company is **296130 rupees** if the condition of low ticket size is not considered

there might be cases where borrower would fall into “Voicebot’ segment or sometimes even “Human calling” segment and we can see the increase in the total cost. Hence, considering all possibilities before segmenting to spend on resources is very important,

when borrowers are punctually repaying loan every month without bouncing, the repayment rate is higher in such cases and they don’t need much attention and watsapp bot medium will work fine, “Human calling” will be required only when the bouncing rate is very high and we cannot risk further on borrower defaulting.

What happens with placing low ticket size in first if statement and second time is, when we take first time even thought low EMI the borrower would have not paid EMI and bounced a lot of times falling into high risk category, this could be little risky to not consider high bounce rate, hence placing the condition in second if statement along with medium risk we could be considering high risk profile with low “ticket size” meaning lesser EMI pending and can deal such borrowers with “voice bot” segment . they have high bounce rate and the total cost has increased only little compared to not considering low ticket size condition.

# Other Findings



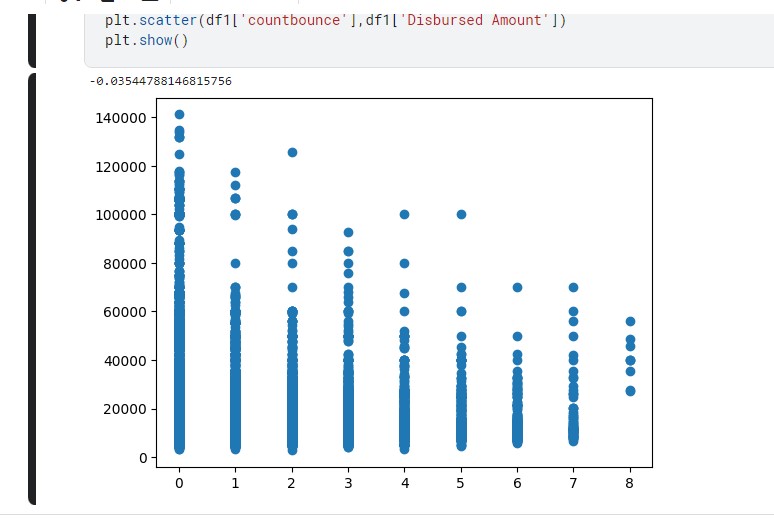
State Maharastra followed by madyapradesh and Tamil Nadu have the highest bounce rate among the states Kerala, Karnataka and Andrapradesh at the bottom having lowest bounce rates

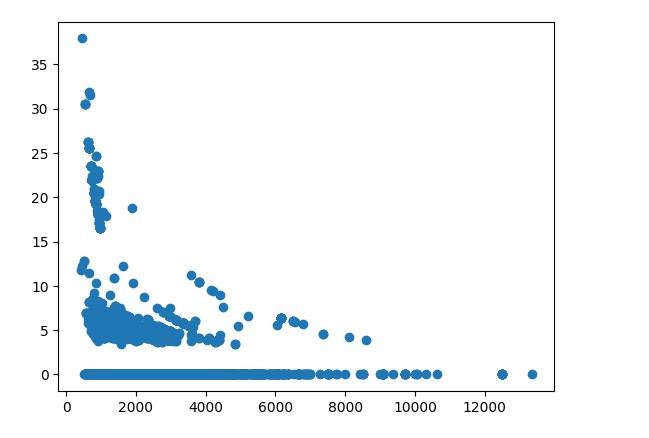


With respect to cities high risk or higher bounce rate is from pune followed by Aurangabad

**To see if high bounce rates are related to certain variables** Lets see If bounce rate is directly proportional to Loan amount. correlation coefficient of the two variables

is-0.03 which is very close to 0 indicating no relation and from the graph we can see the values are distributed all over the graph indicating no positive or negative correlation



Lets see If Amount Pending is directly proportional to 'Interest Rate' correlation coefficient of the two variables is -0.1492 which is very close to 0 indicating no linear relation and from the graph we can see the relation in the opposite direction, albeit weekely. ship is

Amount Pending vs Disbursed Amount

correlation coefficient of the two variables 0.8297 which is very close to 1 indicating positive linear relation and from the graph we can see he relationship is in the positive direction

