**Project #2: Comcast Telecom Consumer Complaints**

**Question1:** Import data into Python environment.

*Python Code:*

import numpy as np

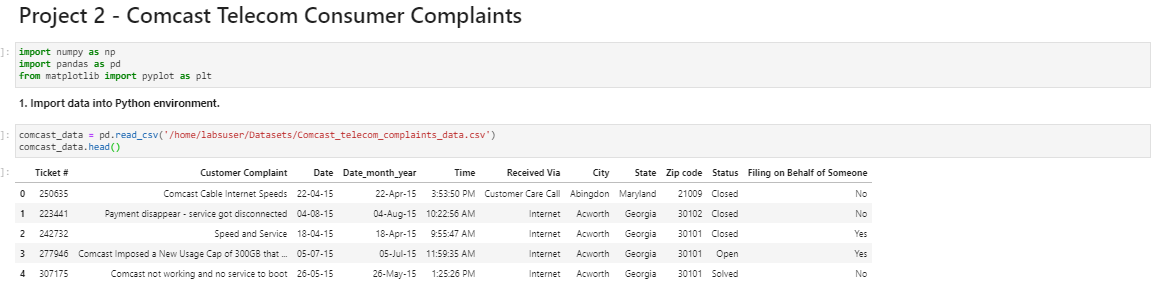
import pandas as pd

from matplotlib import pyplot as plt

comcast\_data = pd.read\_csv('/home/labsuser/Datasets/Comcast\_telecom\_complaints\_data.csv')

comcast\_data.head()

*Screenshot:*



**Question2:** Provide the trend chart for the number of complaints at monthly and daily granularity levels.

*Monthly Trend Code:*

comcast\_data['Month'] = pd.DatetimeIndex(comcast\_data['Date\_month\_year']).month

comcast\_data['Day'] = pd.DatetimeIndex(comcast\_data['Date\_month\_year']).day

grpd\_by\_month = comcast\_data.groupby('Month')

month\_wise\_df = pd.DataFrame(grpd\_by\_month.Month.count())

plt.figure(figsize=(20,10))

plt.title('Month vs Number of Complaints')

plt.plot(month\_wise\_df.index, month\_wise\_df['Month'])

plt.xlabel('Month')

plt.ylabel('Number of Complaints')

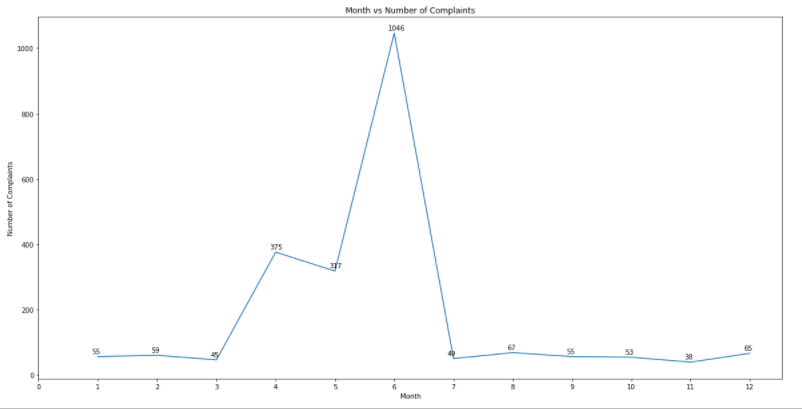
plt.xticks(np.arange(0,13,1))

for x,y in zip(month\_wise\_df.index, month\_wise\_df['Month']):

plt.annotate(str(y), (x,y), (x-0.1,y+10))

plt.show()

*Screenshot:*



*Daily Trend Code:*

grp\_dmy = comcast\_data.groupby('Date\_month\_year')

day\_wise\_filter\_df = pd.DataFrame(grp\_dmy['Customer Complaint'].agg(len))

day\_wise\_filter\_df.index = pd.to\_datetime(day\_wise\_filter\_df.index)

day\_wise\_filter\_df = day\_wise\_filter\_df.sort\_index()

day\_wise\_filter\_df.index = day\_wise\_filter\_df.index.strftime('%d-%m-%y')

plt.figure(figsize=(30,15))

plt.title('Date Vs Number of Complaints')

plt.plot(day\_wise\_filter\_df.index,day\_wise\_filter\_df['Customer Complaint'])

plt.xlabel('Date-->')

plt.ylabel('No of Complaints')

plt.xticks(range(0,len(day\_wise\_filter\_df.index)), day\_wise\_filter\_df.index, rotation=90)

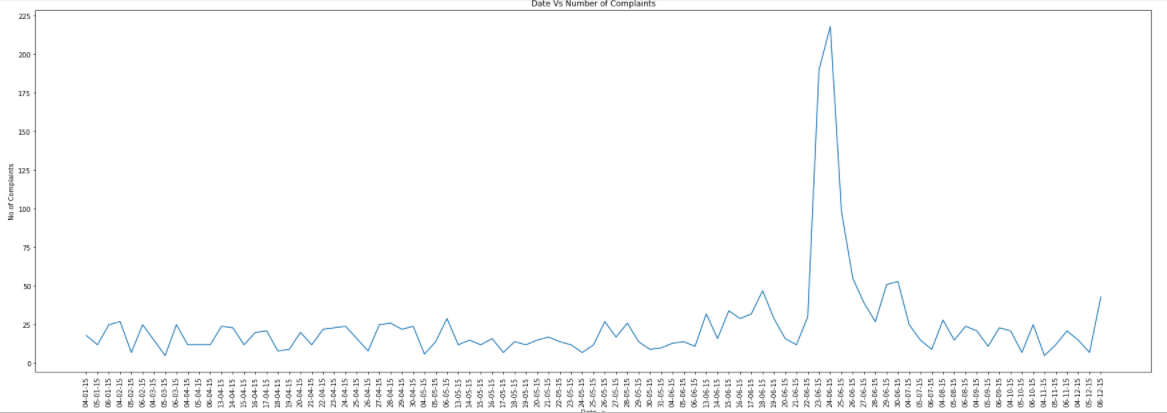
plt.yticks(np.arange(0,250,25))

#for x,y in zip(day\_wise\_filter\_df.index, day\_wise\_filter\_df['Customer Complaint']):

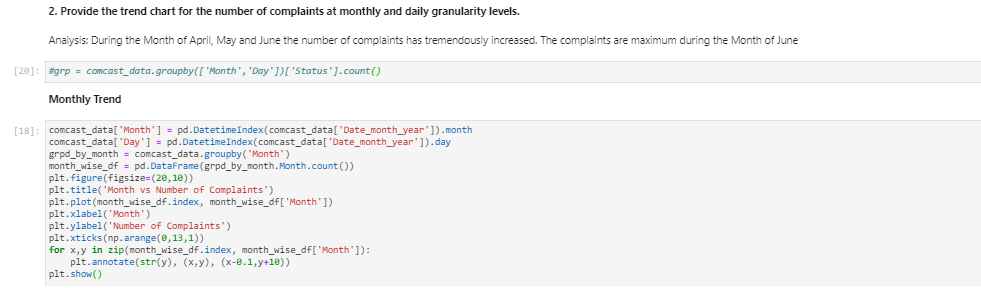
#plt.annotate(str(y), (x,y), (x,y))

plt.show()

*Screenshots:*



*Code Screenshots:*





*Question2 Insights:*

From Plot 1 – Monthly Trend, we can figure out, the number of complaints has tremendously increased during the months April, May and June. And, among these 3 months, the highest number of complaints are recorded in June.

From Plot 2 – Daily Trend, we can figure out, In the month of June, maximum number of complaints are received from 23 June to 26 June(both dates inclusive).

**Question3:** Provide a table with the frequency of complaint types.

* Which complaint types are maximum i.e., around internet, network issues, or across any other domains.

*Code:*

grp\_comp = comcast\_data.groupby('Customer Complaint')

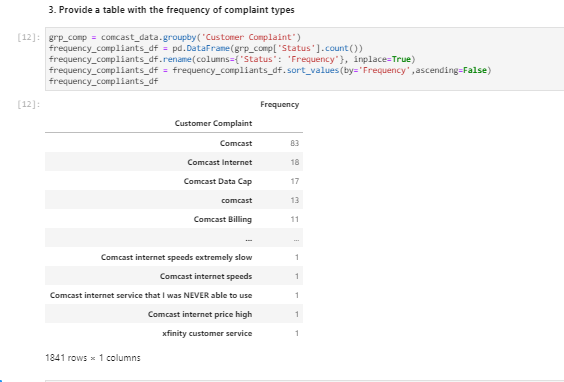
frequency\_compliants\_df = pd.DataFrame(grp\_comp['Status'].count())

frequency\_compliants\_df.rename(columns={'Status': 'Frequency'}, inplace=True)

#frequency\_compliants\_df[frequency\_compliants\_df['Frequency']==83]

frequency\_compliants\_df

*Screenshot:*



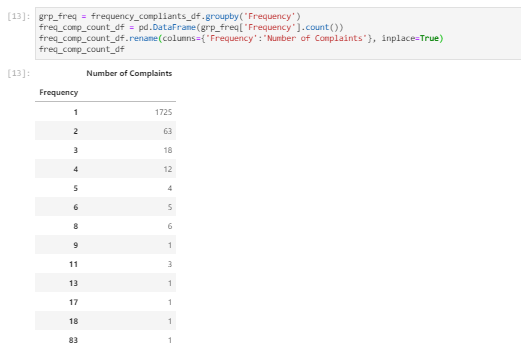
grp\_freq = frequency\_compliants\_df.groupby('Frequency')

freq\_comp\_count\_df = pd.DataFrame(grp\_freq['Frequency'].count())

freq\_comp\_count\_df.rename(columns={'Frequency':'Number of Complaints'}, inplace=True)

freq\_comp\_count\_df

*Screenshot:*



*Question3 Insights:*

The complaints for Comcast Internet are maximum with a total of 18 Complaints.

Though there are 83 complaints which has description as Comcast, as we cannot figure out which domain those complaints belong to, we are considering the next highest number 18 as the maximum complaints in Internet domain.

**Question4:** Create a new categorical variable with value as **Open**and **Closed**. Open & Pending is to be categorized as Open and Closed & Solved is to be categorized as Closed.

*Code:*

comcast\_data['Categorized\_Status'] = comcast\_data.Status

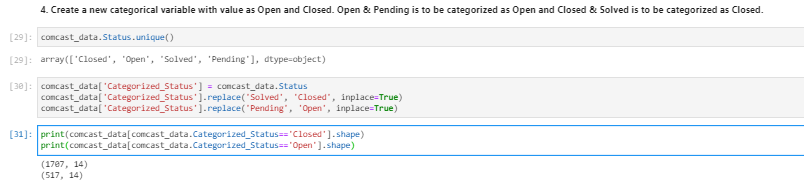
comcast\_data['Categorized\_Status'].replace('Solved', 'Closed', inplace=True)

comcast\_data['Categorized\_Status'].replace('Pending', 'Open', inplace=True)

print(comcast\_data[comcast\_data.Categorized\_Status=='Closed'].shape)

print(comcast\_data[comcast\_data.Categorized\_Status=='Open'].shape)

*Screenshot:*



**Question5:** Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3

*Code:*

states\_list = comcast\_data.State.unique()

listOfAlldata=[]

for state in states\_list:

freshlist=[]

freshlist.append(state)

Opencount = comcast\_data[(comcast\_data.State == state) & (comcast\_data.Categorized\_Status=='Open')].State.count()

Closedcount = comcast\_data[(comcast\_data.State == state) & (comcast\_data.Categorized\_Status=='Closed')].State.count()

TotalCount = Opencount + Closedcount

Percentage\_Open = (Opencount/TotalCount)\*100

Percentage\_Closed = (Closedcount/TotalCount)\*100

freshlist.append(Closedcount)

freshlist.append(Opencount)

freshlist.append(TotalCount)

freshlist.append(round(Percentage\_Open,2))

freshlist.append(round(Percentage\_Closed,2))

listOfAlldata.append(freshlist)

#listOfAlldata

new\_df = pd.DataFrame(listOfAlldata, columns=['StateName', 'ClosedCount', 'OpenCount', 'TotalCount', '% Open', '% Closed'])

new\_df = new\_df.sort\_values(by='StateName')

#new\_df.head()

plt.figure(figsize=(30, 10))

plt.title('State Vs. Number of Closed and Open Complaints')

plt.bar(new\_df.StateName, new\_df.ClosedCount, label='Closed')

plt.bar(new\_df.StateName, new\_df.OpenCount,bottom=new\_df.ClosedCount,label='Open')

plt.xticks(range(0, len(new\_df.index)), new\_df.StateName, rotation=90)

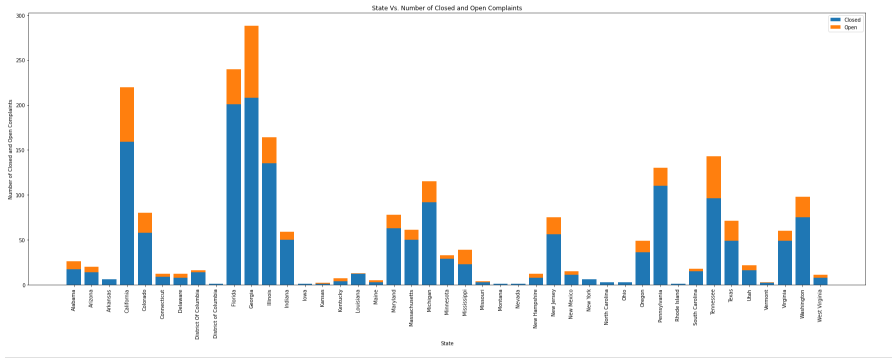
plt.xlabel('State')

plt.ylabel('Number of Closed and Open Complaints')

plt.legend()

plt.show()

*Screenshot:*



*Additional Screenshot of Data Frame Created for Q#5:*



**Question 5.1:**

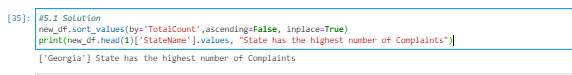
*Code:*

#5.1 Solution

new\_df.sort\_values(by='TotalCount',ascending=False, inplace=True)

print(new\_df.head(1)['StateName'].values, "State has the highest number of Complaints")

*Screenshot:*



**Question 5.2:**

*Code:*

#5.2 Solution

new\_df.sort\_values(by='% Open',ascending=False,inplace=True)

print(new\_df.head(1)['StateName'].values, "State has the highest percentage of Unresolved Complaints")

*Screenshot:*



*Question#5 Insights:*

Georgia State has the highest number of Complaints.

Kansas State has the highest percentage of Unresolved Complaints.

**Question 6:** Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

*Code:*

closed\_Internet = comcast\_data[(comcast\_data['Received Via']=='Internet') & (comcast\_data['Categorized\_Status']=='Closed')].Categorized\_Status.count()

open\_Internet = comcast\_data[(comcast\_data['Received Via']=='Internet') & (comcast\_data['Categorized\_Status']=='Open')].Categorized\_Status.count()

closed\_customercare = comcast\_data[(comcast\_data['Received Via']=='Customer Care Call') & (comcast\_data['Categorized\_Status']=='Closed')].Categorized\_Status.count()

open\_customercare = comcast\_data[(comcast\_data['Received Via']=='Customer Care Call') & (comcast\_data['Categorized\_Status']=='Open')].Categorized\_Status.count()

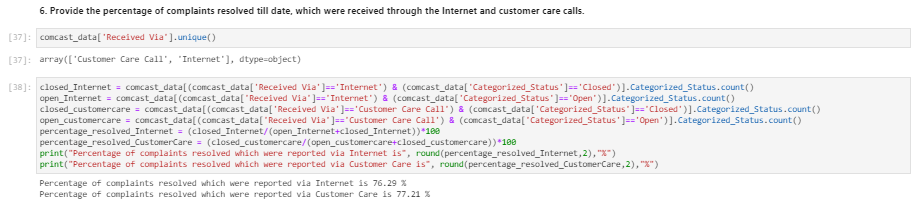
percentage\_resolved\_Internet = (closed\_Internet/(open\_Internet+closed\_Internet))\*100

percentage\_resolved\_CustomerCare = (closed\_customercare/(open\_customercare+closed\_customercare))\*100

print("Percentage of complaints resolved which were reported via Internet is", round(percentage\_resolved\_Internet,2),"%")

print("Percentage of complaints resolved which were reported via Customer Care is", round(percentage\_resolved\_CustomerCare,2),"%")

*Screenshot:*



*Question #6 Insights:*

Percentage of complaints resolved which were reported via Customer Care is 77.21 %

Percentage of complaints resolved which were reported via Internet is 76.29 %