Comcast Telecom Consumer Complaints

#1. Import data into python environment

import pandas as py

data = py.read\_csv('Comcast\_telecom\_complaints\_data.csv')

#2. Provide the trend chart for the number of complaints at monthly granularity levels.

import numpy as np

import datetime

Complaint\_data = py.DataFrame(data)

Complaint\_data['Month'] = py.DatetimeIndex(Complaint\_data['Date\_month\_year']).month

Complaint\_data.head()

import matplotlib.pyplot as plt

Month\_Complaint\_count= Complaint\_data.groupby('Month').count()

plt.bar(Month\_Complaint\_count.index.values, Month\_Complaint\_count['Status'])

#2.b. Provide the trend chart for the number of complaints at Daily granularity levels.

Complaint\_data['Date'] = py.DatetimeIndex(Complaint\_data['Date\_month\_year']).date

Complaint\_data.head()

Date\_Complaint\_count= Complaint\_data.groupby("Date")["Date"].count()

Date\_Complaint\_count.plot()

#3. Provide a table with the frequency of complaint types.

Total\_Complaint\_data = Complaint\_data['Customer Complaint'].count()

Total\_Complaint\_data

Complaint\_data\_frequency = Complaint\_data['Customer Complaint'].str.lower().value\_counts()

Complaint\_data\_frequency

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

Internet\_count\_data = Complaint\_data['Customer Complaint'].str.lower().str.contains("internet").value\_counts()

Internet\_count\_data

#Total internet complaints are 532

Network\_count\_data = Complaint\_data['Customer Complaint'].str.lower().str.contains("network").value\_counts()

Network\_count\_data

# Total network complaints are 2

Other\_domain\_count\_data = Total\_Complaint\_data-(Internet\_count\_data + Network\_count\_data)

Other\_domain\_count\_data

# Complaints other than internet and network are 1690

#Total Maximum are 1690 complaints of other domain data

# 4. 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.

Complaint\_data.loc[(Complaint\_data['Status'] == 'Open') | (Complaint\_data['Status'] =='Pending'), 'new\_status'] = 'Open'

Complaint\_data.loc[(Complaint\_data['Status'] == 'Closed') | (Complaint\_data['Status'] =='Solved'), 'new\_status'] = 'Closed'

Complaint\_data.head()

#5. Provide state wise status of complaints in a stacked bar chart.

State\_Complaint\_count= Complaint\_data.groupby('State').count()

plt.bar(State\_Complaint\_count.index.values,State\_Complaint\_count['Status'])

# 5.a. Which state has the maximum complaints

Max\_State\_Complaint\_Data = State\_Complaint\_count['Status'].idxmax()

Max\_State\_Complaint\_Data

#Maximum complaints are raised by Georgia state

#5.b. Which state has the highest percentage of unresolved complaints

Unresolved\_complaints = Complaint\_data['State'][Complaint\_data['new\_status'] == 'Open']

Unresolved\_complaints\_count = Unresolved\_complaints.count()

Unresolved\_complaints\_count

Unresolved\_complaints.value\_counts().idxmax()

#Maximum unresolved complaints are by Georgia

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

Complaint\_count = Complaint\_data['State'].count()

Complaint\_count

Resolved\_complaints = Complaint\_count - Unresolved\_complaints\_count

Resolved\_complaints

Resolved\_complaints\_percentage = (Resolved\_complaints/Complaint\_count)\*100

Resolved\_complaints\_percentage

# Percentage of complaints resolved is 76.75%