**Comcast Telecom Consumer Complaints**

DESCRIPTION

Comcast is an American global telecommunication company. The firm has been providing terrible customer service. They continue to fall short despite repeated promises to improve. Only last month (October 2016) the authority fined them a $2.3 million, after receiving over 1000 consumer complaints. The existing database will serve as a repository of public customer complaints filed against Comcast. It will help to pin down what is wrong with Comcast's customer service.

Importing necessary Libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

**Task 1** : Import data into Python environment

comcast\_df = pd.read\_csv('C:/Users/Workstation/OneDrive/Desktop/Data Science\_Material/Data Science with Python/Python\_Projects/Comcast Telecom Consumer Complaints/Comcast\_telecom\_complaints\_data.csv')

comcast\_df.head()

comcast\_df.info()

comcast\_df.isna().sum()

**Task 2** : Provide the trend chart for the number of complaints at monthly and daily granularity levels.

To perform this task, 'Date\_month\_year' would be our target attribute.

* ***We have to change the datatype from 'object' to 'datetime'***

comcast\_df['Date\_month\_year'] = pd.to\_datetime(comcast\_df['Date\_month\_year'])

* ***Extract "Month", "date" and "Days of week" using lambda function***

comcast\_df['Month']=comcast\_df['Date\_month\_year'].apply(lambda x: x.month)

comcast\_df['Date']=comcast\_df['Date\_month\_year'].apply(lambda x: x.day)

comcast\_df['Day\_of\_week']=comcast\_df['Date\_month\_year'].apply(lambda x: x.dayofweek)

* ***map the weeks and months as follows:***

d={0:'Mon', 1: 'Tue', 2: 'Wed', 3: 'Thur', 4: 'Fri', 5: 'Sat', 6: 'Sun'}

comcast\_df['Day\_of\_week']=comcast\_df['Day\_of\_week'].map(d)

s = { 1: 'Jan', 2: 'Feb', 3: 'Mar', 4: 'Apr', 5: 'May', 6: 'Jun', 7: 'Jul', 8: 'Aug', 9: 'Sep', 10: 'Oct', 11: 'Nov', 12: 'Dec' }

comcast\_df['Month']=comcast\_df['Month'].map(s)

comcast\_df.head(10)

***Number of Complaints month wise***

Months\_df = comcast\_df.groupby(comcast\_df['Month']).size()

Months\_df

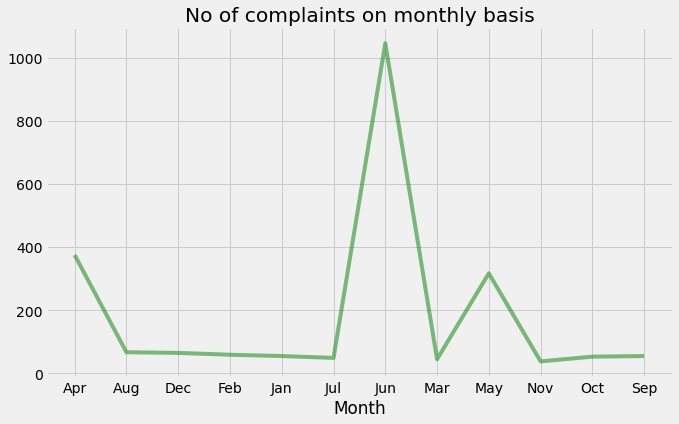
from matplotlib import style

style.use('fivethirtyeight')

plt.figure(figsize=(10,6))

sns.lineplot(data = Months\_df, color = "green", alpha = 0.5)

plt.title("No of complaints on monthly basis")



***Number of Complaints on daily basis***

Daily\_df = comcast\_df.groupby(comcast\_df['Date']).size()

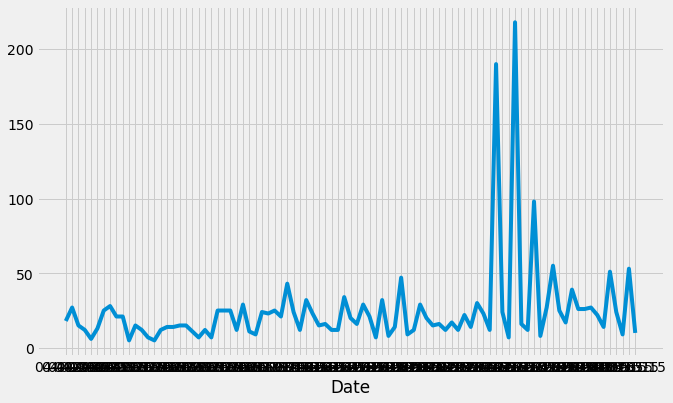
Daily\_df

from matplotlib import style

style.use('fivethirtyeight')

plt.figure(figsize=(10,6))

sns.lineplot(data = Daily\_df)

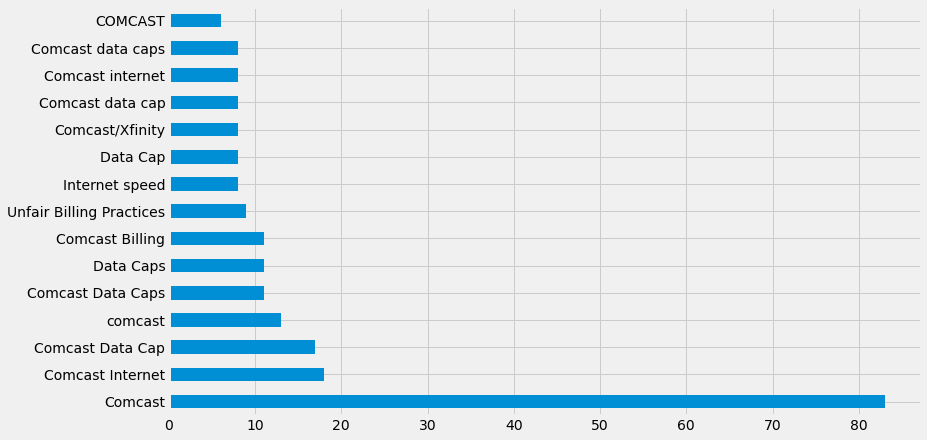


**Task 3** - Provide a table with the frequency of complaint types

Complaints\_df = comcast\_df[('Customer Complaint')].value\_counts()[:15]

Complaints\_df

Complaints\_df.plot(kind="barh", figsize=(12,7))



**Task - 4** : Which complaint types are maximum i.e., around internet, network issues, or across any other domains.

* 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.

comcast\_df['Highlevel\_Status'] = ["Open" if Status=="Open" or Status=="Pending" else "Closed" for Status in comcast\_df["Status"]]

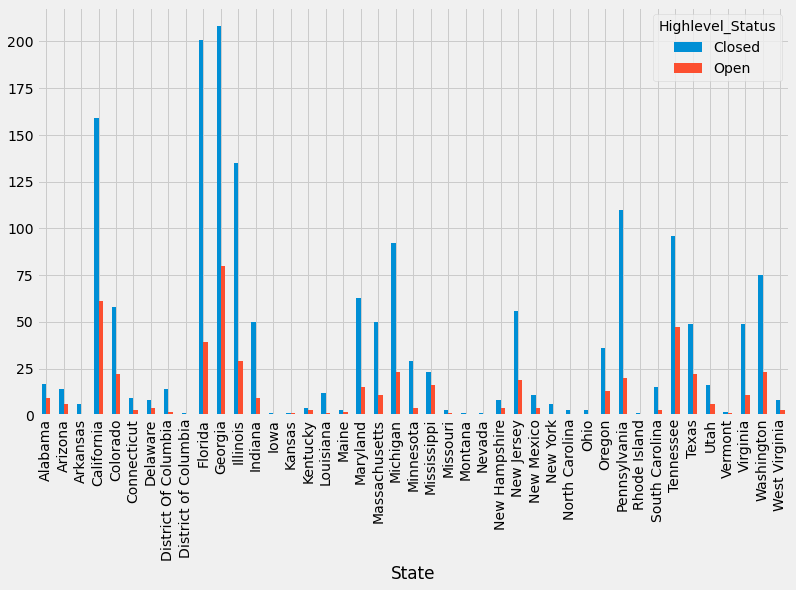
comcast\_df.head(15)

comp\_st=comcast\_df.groupby(['State', 'Highlevel\_Status']).size().unstack().fillna(0)

comp\_st

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

comp\_st.plot(kind="bar", figsize=(12,7))



comcast\_df.groupby(['State']).size().sort\_values(ascending = False).to\_frame().rename({0: "Complaint count"}, axis=1)[:1]

**Conclusion:** "Georgia" state has the maximum complaint count.

* ***Which state has the highest percentage of unresolved complaints?***

un\_cm['Unresolved\_comp\_perc']=un\_cm['Open']/un\_cm['Open'].sum()\*100

un\_cm.sort\_values('Unresolved\_comp\_perc', ascending=False, axis=0)[:1]

**Conclusion** - "Georgia" has the highest 15% of unresolved complaints

**Task - 5** : Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

int\_ccc=comcast\_df.groupby(['Received Via', 'Highlevel\_Status']).size().unstack().fillna(0)

int\_ccc['Resolved']=int\_ccc['Closed']/int\_ccc['Closed'].sum()\*100

int\_ccc['Resolved']

**Conclusion** - 50% of Complaints received via "Costumer Care Call" and 49% of complaints recieved by "Internet" are resolved

