Data Analytics with Python

Certification Project



**Comcast Telecom Consumer Complaints**

**Objective:**  Use data to understand what factors that affect the Comcast telecom company services.

And find what type of complaints are registered and from which state we are getting maximum complaints and what are complaint types , status of complaints in each state.

**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

**Domain:** General

**Analysis to be done:**

* Import data into Python environment.
* Provide the trend chart for the number of complaints at monthly and daily granularity levels.
* Provide a table with the frequency of complaint types.

1. 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.
* Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provide insights on:
  + Which state has the maximum complaints
  + Which state has the highest percentage of unresolved complaint
* Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

**Content:** Dataset: Comcast\_Telecom\_Complaints\_data.csv

**Data Dictionary:**

* Ticket #: Ticket number assigned to each complaint
* Customer Complaint: Description of complaint
* Date: Date of complaint
* Time: Time of complaint
* Received Via: Mode of communication of the complaint
* City: Customer city
* State: Customer state
* Zipcode: Customer zip
* Status: Status of complaint
* Filing on behalf of someone

**Steps to perform:**

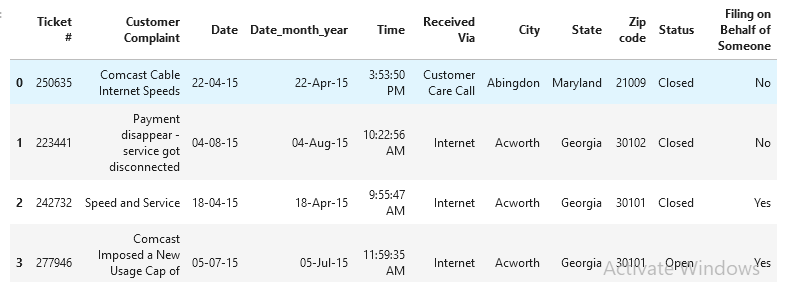
As the first step, Load the file into your Jupiter notebook and look at the null values in the file. A sanity check, to ensure that you have clean records and the data is good to go ahead, is very important. And perform further analysis and provide insights where required.

**Solution**

1. **Import data into Python environment**

df=pd.read\_csv("/Users/admin/Downloads/Comcast\_telecom\_complaints\_data.csv")

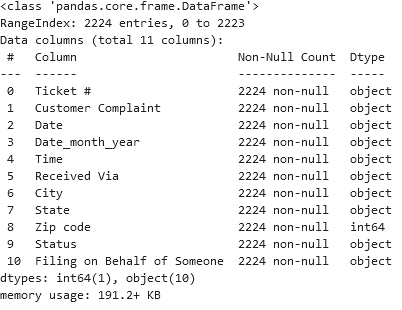
df.head()





* **Check the information of all columns by info()**

df.info()

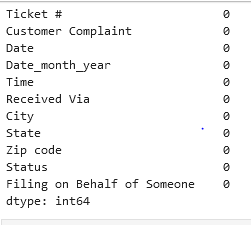
****

In This We can Check data types of all columns.we checked of there is any null values present by looking at non null value count. If every column having same count of non null values then there is no null values.

* **Check for count null values in each column, another way by isnull()**

df.isnull().sum()

Here all values are “0” ,so there is no null values .we can do analysis on this dataframe.



1. **Provide the trend chart for the number of complaints at monthly** **levels.**

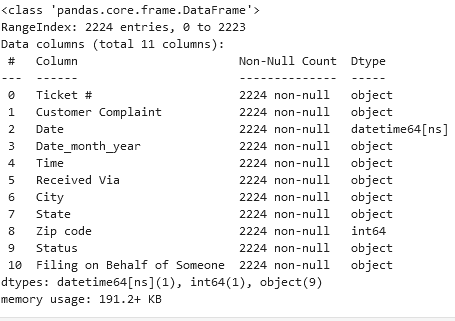
In this dataset there is no month column so first we convert the datatype of “Date” column to datetime datatype ,and create a month column to dataframe by extracting month from “Date” column.

* **Convert the datatype**

df['Date']=pd.to\_datetime(df['Date'])

* **Checking the data type of Date after conversion.**

df.info()



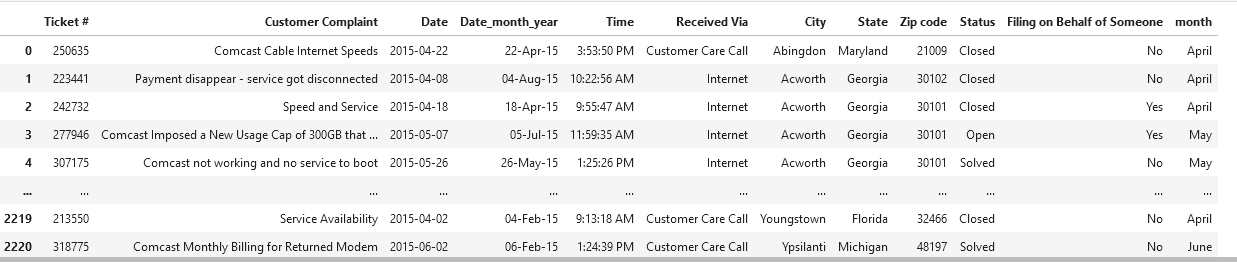
Here the datatype of “Date” column is changed to date time.

* **Creating a new month column.**

df['month']=df['Date'].dt.month\_name()

* **Checking for df again.**

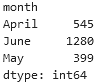
df

****

Here we can see new month column.

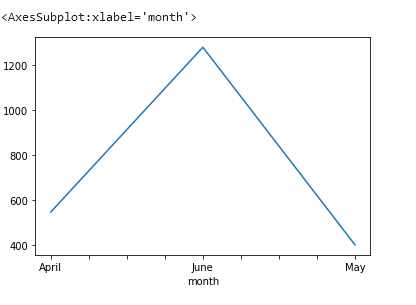
* **Apply group by to month column and calculating no of complaints for each month**

df.groupby('month').size()



* **Ploting trend chat for no of complaints for each month**

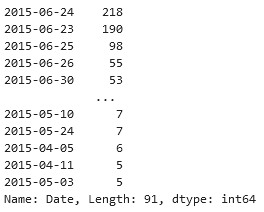
df.groupby('month').size().plot(kind='line')

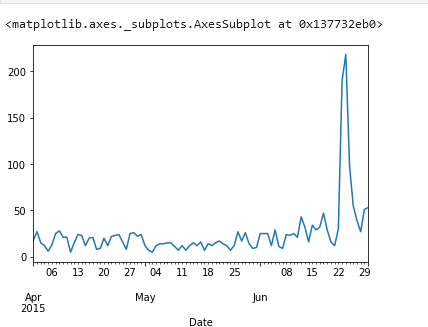


1. **Plotting line chart to see trend in no of complaints for each day**

* **Calculating no of complaints for each day**

df['Date'].value\_counts()



****

Here in Date column we have only 3 months .for 3 month, for each day no of complaints

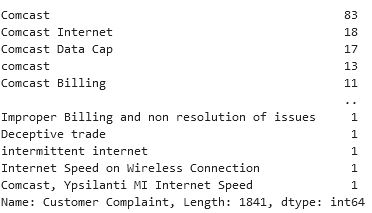
are in this trend.

1. **Provide a table with the frequency of complaint types.**

Here 'Customer Complaint' is the complaints type column

* **Checking count for each 'Customer Complaint' column value.**

df['Customer Complaint'].value\_counts()



Here uniqe values are not identified correctly so use NLP(Natural Language processing ) or other method to find correct unique values, and provide the frequency of complaint types

table.

* **Another way to find complaint types.**

First convert the 'Customer Complaint' to lower case by lower(),and then search for same type of word that represents the same type of compliaints, and add all complaints.repeat this process and find complaint types from that column and provide a table.

* **Converting column to lower case.**

df['Customer Complaint']=df['Customer Complaint'].str.lower()

* **For internet complaint type.**

internet\_issues1=df[df['Customer Complaint'].str.contains('internet')].count()

internet\_issues2=df[df['Customer Complaint'].str.contains('data')].count()

internet\_issues3=df[df['Customer Complaint'].str.contains('network')].count()

total\_internet=internet\_issues1+internet\_issues2+internet\_issues3

* **For Billing complaint type.**

billing\_issues1=df[df['Customer Complaint'].str.contains('billing')].count()

billing\_issues2=df[df['Customer Complaint'].str.contains('charges')].count()

billing\_issues3=df[df['Customer Complaint'].str.contains('bill')].count()

total\_biling=billing\_issues1+billing\_issues2+billing\_issues3

* **For Other complaint type.**

other\_issues=2224-(total\_biling+total\_internet)

* **Frequency table of complaint types.**

total\_internet['Ticket #']

753

total\_biling['Ticket #']

754

other\_issues['Ticket #']

717

frequency\_table\_of\_complainttypes=pd.DataFrame({"internet issues":[total\_internet['Ticket #']],"Billing\_issues":[total\_biling['Ticket #']],"Other\_issues":[other\_issues["Ticket #"]]})

frequency\_table\_of\_complainttypes

****

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

**other domains.**

Here “other issues” are high among all complaint types.

1. **Create a new categorical variable with value as Open and Closed. Open & Pending**

**to be categorized as Open and Closed & Solved is to be categorized as Closed.**

* **Here first checking the unique values of present “Status” column.we have 4 values**

df['Status'].unique()

output: array(['Closed', 'Open', 'Solved', 'Pending'],

dtype=object)

* **Creating a new\_Status column Converting 4 unique values to 2 unique values by applying if\_else condition**

df['new\_status']=["Open" if x=="Open" or x=="Pending" else

"Closed" for x in df['Status']]

* **Here we are checking for the new column unique values**

df['new\_status'].unique()

output: array(['Closed', 'Open'], dtype=object)

1. **Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provide insights on:**
   * **Which state has the maximum complaints**

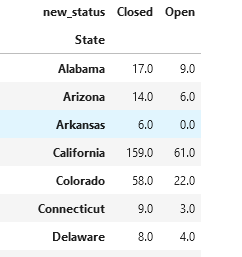
Grouping the “state” and “new\_status “ column for calculating no of Open

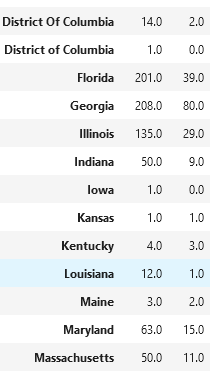
and Closed complaints in each state,and filling the null values with”0”

state\_complain=df.groupby(['State','new\_status']).size().unstack().fillna(0)

state\_complain

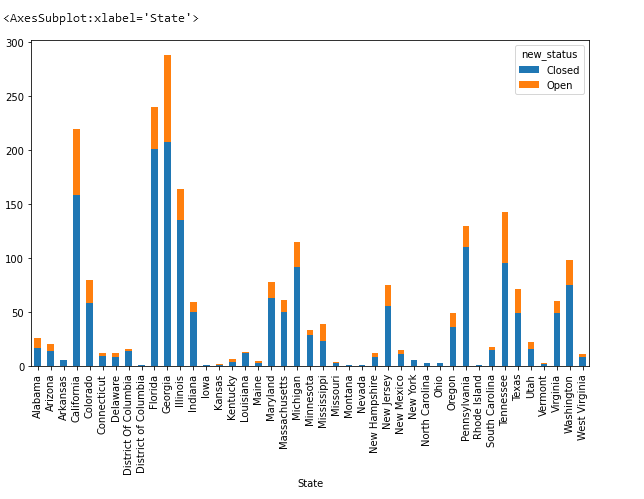
Checking for grouped data.



****

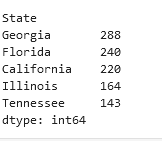
* **Stacked Bar Chart for no of open and closed complaints in each state**

state\_complain.plot.bar(stacked=True,figsize=(10,10))



Here we are sorting the values in descending order to find state with maxiun no of complaints

df.groupby('State').size().sort\_values(ascending=False)[:5]

****

Here we can observe that the state with maximum complaints is Georgia with 288

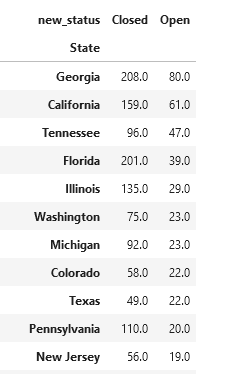
complaints

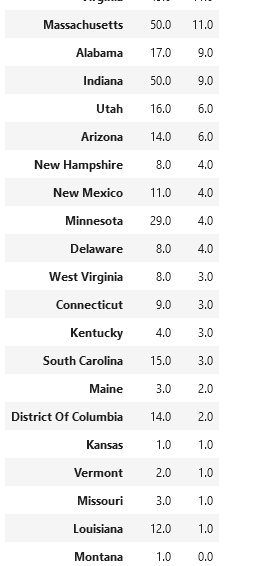
1. **Which state has the highest percentage of unresolved complaint**

Here unresolved data is Open complaints.To calculate the highest percentage of unresolved data group data in 2 columns and unstuck them to get only open complaints data for each state

unresolved\_data=df.groupby(['State','new\_status']).size().unstack().fillna(0).

sort\_values(by='Open',ascending=False)





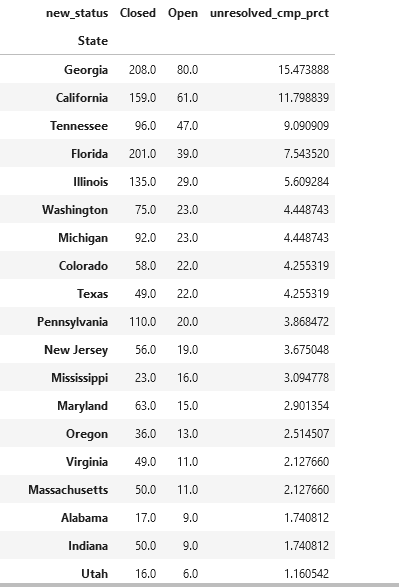
Now we can calculate the percentage by this formula.

unresolved\_data['unresolved\_cmp\_prct']=unresolved\_data['Open']

/unresolved\_data['Open'].sum()\*100

unresolved\_data

Now checking the unresolved percentage of each state.Here the values are in descending orderb so we can observe that the state with maximum unresolved percentage is Georgia with 15.473888.

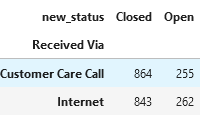


1. **Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.**

Here resolved data is Closed complaints.To calculate the highest percentage of resolved data group data in 2 columns 'Received Via' and 'new\_status' and unstuck them to get only Closed complaints data for each state.

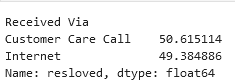
resolved\_data=df.groupby(['Received Via','new\_status']).size().unstack()

resolved\_data



resolved\_data['resloved']=resolved\_data['Closed']/resolved\_data['Closed'].sum()\*100

resolved\_data['resloved']



Here we can observe that the complaints resolved , received via Customre care call are 50.615114 percentage , received via internet are 49.384886 percentage.