Comcast_Customer_Complaints

November 24, 2023

[1]:
```
#import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[ ]:
```
1. Import data into Python environment.
```

[2]:
```
df= pd.read_csv('Comcast_telecom_complaints_data.csv')
```

[3]:
```
df.head()
```

[3]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone
0	250635	Comcast Cable Internet Speeds	22-04-15	22-Apr-15	3:53:50 PM	Customer Care Call	Abingdon	Maryland	21009	Closed	No
1	223441	Payment disappear - service got disconnected	04-08-15	04-Aug-15	10:22:56 AM	Internet	Acworth	Georgia	30102	Closed	No
2	242732	Speed and Service	18-04-15	18-Apr-15	9:55:47 AM	Internet	Acworth	Georgia	30101	Closed	Yes
3	277946	Comcast Imposed a New Usage Cap of 300GB that …	05-07-15	05-Jul-15	11:59:35 AM	Internet	Acworth	Georgia	30101	Open	Yes
4	307175	Comcast not working and no service to boot	26-05-15	26-May-15	1:25:26 PM	Internet	Acworth	Georgia	30101	Solved	No

[4]:
```
df.shape
```

[4]:
```
(2224, 11)
```

[5]:
```
df.columns
```

[5]:
```
Index(['Ticket #', 'Customer Complaint', 'Date', 'Date_month_year', 'Time',
       'Received Via', 'City', 'State', 'Zip code', 'Status',
       'Filing on Behalf of Someone'],
      dtype='object')
```

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2224 entries, 0 to 2223
Data columns (total 11 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   Ticket #                     2224 non-null   object
 1   Customer Complaint           2224 non-null   object
 2   Date                         2224 non-null   object
 3   Date_month_year              2224 non-null   object
 4   Time                         2224 non-null   object
 5   Received Via                 2224 non-null   object
 6   City                         2224 non-null   object
 7   State                        2224 non-null   object
 8   Zip code                     2224 non-null   int64 
 9   Status                       2224 non-null   object
 10  Filing on Behalf of Someone  2224 non-null   object
dtypes: int64(1), object(10)
memory usage: 191.2+ KB
```

[23]:
```
# Check for duplicate entries in the 'Customer Complaint' column
duplicate_complaints = df[df.duplicated(subset='Customer Complaint')]

# Display the duplicate entries
print("Duplicate Complaints:")
print(duplicate_complaints[['Customer Complaint']])
```

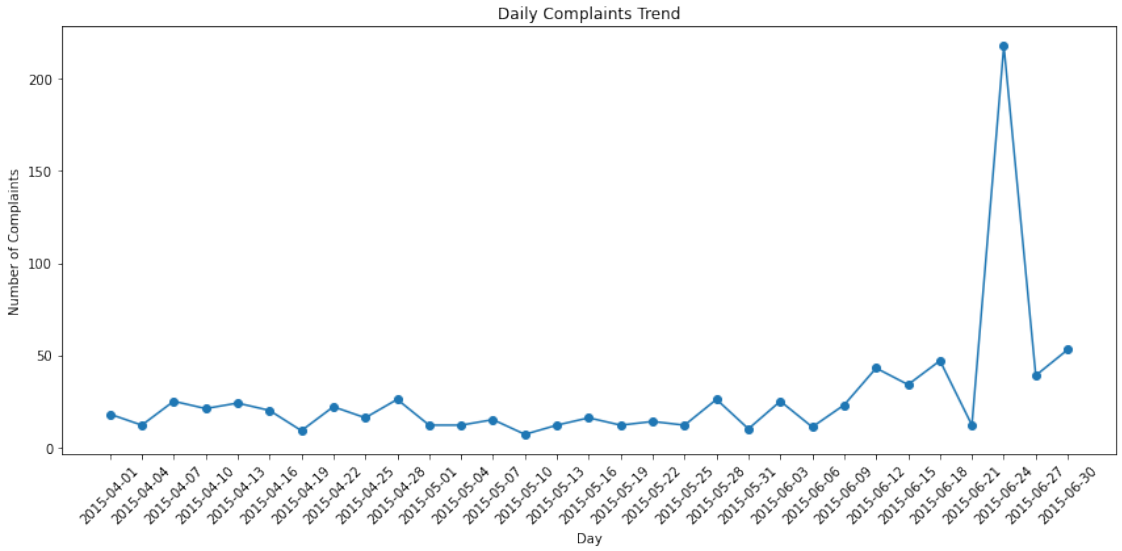
```
Duplicate Complaints:
     Customer Complaint
37               Comcast
47               Comcast
61               Comcast
91               Comcast
98               Comcast
…                    …
2173      Internet speed
2180             Comcast
2189             Comcast
2207    Comcast Internet
2211             Comcast
```

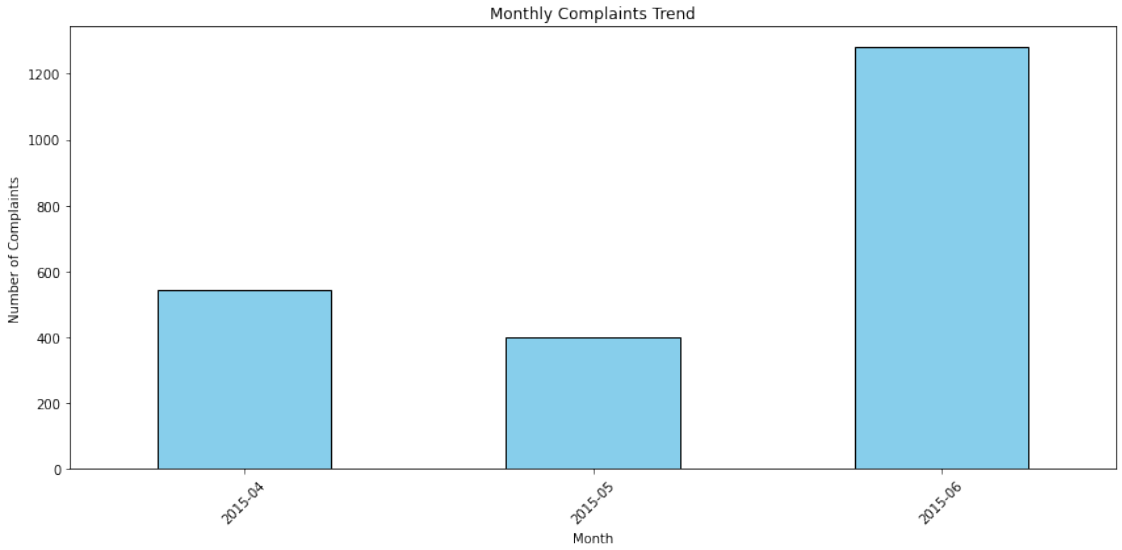
```
[383 rows x 1 columns]
```

[ ]:
```
1.1 Provide the trend chart for the number of complaints at monthly and daily granularity levels.
```

[11]:
```
# Load the CSV file into a DataFrame
df = pd.read_csv('Comcast_telecom_complaints_data.csv')

# Convert the 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Create a new column for the day
df['Day'] = df['Date'].dt.to_period('D')

# Group by day to get the number of complaints
daily_complaints = df.groupby('Day').size()

# Plot the trend chart for daily complaints with improved readability
plt.figure(figsize=(12, 6))
n = 3  # Plot every 7th day for better readability
plt.plot(daily_complaints.index.astype(str)[::n], daily_complaints.values[::n], marker='o')
plt.title('Daily Complaints Trend')
plt.xlabel('Day')
plt.ylabel('Number of Complaints')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()


# Convert the 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Create a new column for the month
df['Month'] = df['Date'].dt.to_period('M')

# Group by month to get the number of complaints
monthly_complaints = df.groupby('Month').size()

# Plot the trend chart for monthly complaints using a column chart
plt.figure(figsize=(12, 6))
monthly_complaints.plot(kind='bar', color='skyblue', edgecolor='black')
plt.title('Monthly Complaints Trend')
plt.xlabel('Month')
plt.ylabel('Number of Complaints')
plt.xticks(rotation=45)
```

```
plt.tight_layout()
plt.show()
```





[30]:
```
# 1.2 Provide a table with the frequency of complaint types.
# Display the frequency of complaint types in a table
complaint_types_frequency = df['Customer Complaint'].value_counts().
 ↪reset_index()
complaint_types_frequency.columns = ['Complaint Type', 'Frequency']

# Print the table
```

```
print(complaint_types_frequency)
```

```
                                    Complaint Type  Frequency
0                                          Comcast         83
1                                  Comcast Internet         18
2                                  Comcast Data Cap         17
3                                          comcast         13
4                                   Comcast Billing         11
…                                               …        …
1836  Improper Billing and non resolution of issues          1
1837                                Deceptive trade          1
1838                            intermittent internet        1
1839          Internet Speed on Wireless Connection          1
1840           Comcast, Ypsilanti MI Internet Speed          1

[1841 rows x 2 columns]
```

[28]:
```
# 2. Which complaint types are maximum i.e., around internet, network issues,␣
 ↪or across any other domains.

# Group by 'Customer Complaint' to aggregate frequencies
complaints_frequency = df.groupby('Customer Complaint').size().
 ↪reset_index(name='Frequency')

# Group by 'Complaint Type' to sum frequencies
aggregated_complaints = complaints_frequency.groupby('Customer␣
 ↪Complaint')['Frequency'].sum().reset_index()

# Sort the DataFrame by 'Frequency' in descending order
sorted_complaints = aggregated_complaints.sort_values(by='Frequency',␣
 ↪ascending=False)

# Display the top complaint types and their frequencies
print("Top Complaint Types:")
print(sorted_complaints.head(10))
```

```
Top Complaint Types:
            Customer Complaint  Frequency
160                    Comcast         83
331          Comcast Internet          18
266          Comcast Data Cap          17
1631                   comcast         13
208           Comcast Billing          11
273         Comcast Data Caps          11
906                  Data Caps          11
1530  Unfair Billing Practices           9
569          Comcast data cap           8
```

```
575          Comcast data caps          8
```

[ ]:
```
2.1 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.
```

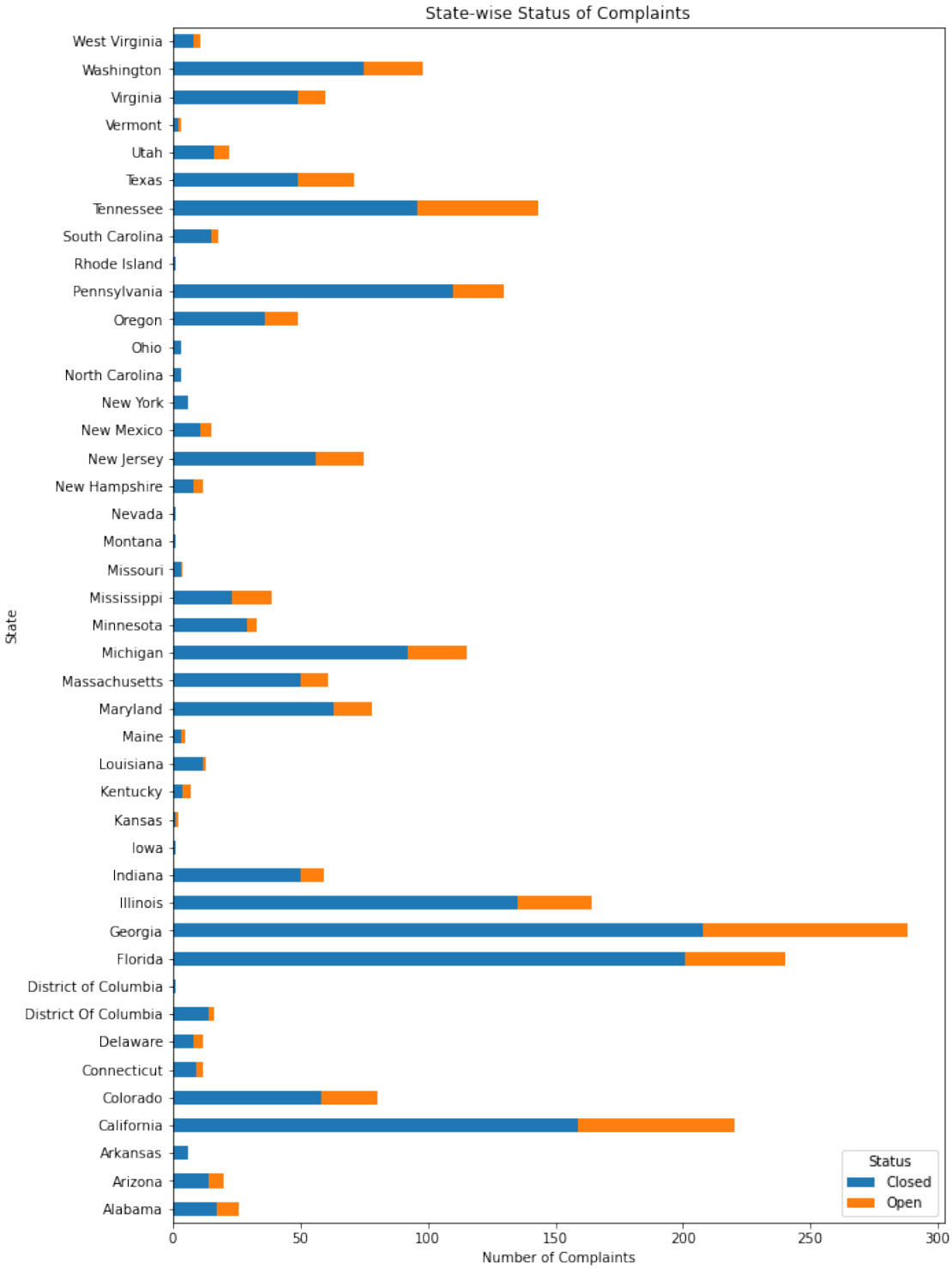
[31]:
```
# Create a new categorical variable based on the specified conditions
df['Status'] = np.where((df['Status'] == 'Open') | (df['Status'] == 'Pending'),␣
↪'Open', 'Closed')

# Display the updated DataFrame with the new 'Status' column
print(df.head())
```

```
  Ticket #                                  Customer Complaint       Date  \
0   250635                       Comcast Cable Internet Speeds 2015-04-22
1   223441       Payment disappear - service got disconnected 2015-04-08
2   242732                                    Speed and Service 2015-04-18
3   277946  Comcast Imposed a New Usage Cap of 300GB that … 2015-05-07
4   307175         Comcast not working and no service to boot 2015-05-26

  Date_month_year         Time        Received Via      City     State  \
0       22-Apr-15   3:53:50 PM  Customer Care Call  Abingdon  Maryland
1       04-Aug-15  10:22:56 AM            Internet   Acworth   Georgia
2       18-Apr-15   9:55:47 AM            Internet   Acworth   Georgia
3       05-Jul-15  11:59:35 AM            Internet   Acworth   Georgia
4       26-May-15   1:25:26 PM            Internet   Acworth   Georgia

   Zip code  Status Filing on Behalf of Someone         Day    Month
0     21009  Closed                           No  2015-04-22  2015-04
1     30102  Closed                           No  2015-04-08  2015-04
2     30101  Closed                          Yes  2015-04-18  2015-04
3     30101    Open                          Yes  2015-05-07  2015-05
4     30101  Closed                           No  2015-05-26  2015-05
```

[ ]:
```
2.2 Provide state wise status of complaints in a stacked bar chart. Use the␣
↪categorized variable from Q3.
```

[38]:
```
# Group by 'State' and 'Status' to get the counts
state_status_counts = df.groupby(['State', 'Status']).size().unstack().fillna(0)

# Increase the length of the chart for better visualization
fig, ax = plt.subplots(figsize=(12, 16))

# Plot the horizontal stacked bar chart for all states
state_status_counts.plot(kind='barh', stacked=True, ax=ax)

plt.title('State-wise Status of Complaints')
```

```
plt.xlabel('Number of Complaints')
plt.ylabel('State')
plt.legend(title='Status', loc='lower right')

# Adjust layout to prevent clipping of the xlabel
plt.subplots_adjust(left=0.25, right=0.9)

plt.show()
```



[ ]: 2.3 Provide insights on: Which state has the maximum complaints.

[39]:
```
# Find the state with the maximum complaints
max_complaints_state = df.groupby(['State', 'Status']).size().unstack().
 ↪fillna(0).sum(axis=1).idxmax()

print(f"The state with the maximum complaints is: {max_complaints_state}")
```

```
The state with the maximum complaints is: Georgia
```

[40]:
```
# Find the state with the maximum complaints
max_complaints_state = df['State'].value_counts().idxmax()

print(f"The state with the maximum complaints is: {max_complaints_state}")
```

```
The state with the maximum complaints is: Georgia
```

[ ]:
```
The state with Maximum Complaints from categorized variable from Q3 and Maximum␣
 ↪Complaints throughout is Georgia.
```

[ ]:
```
2.4 Which state has the highest percentage of unresolved complaints?
```

[41]:
```
# Group by 'State' and 'Status' to get the counts
state_status_counts = df.groupby(['State', 'Status']).size().unstack().fillna(0)

# Calculate the percentage of unresolved complaints for each state
state_status_counts['Unresolved Percentage'] = (state_status_counts['Open'] /␣
 ↪state_status_counts.sum(axis=1)) * 100

# Find the state with the highest percentage of unresolved complaints
max_unresolved_state = state_status_counts['Unresolved Percentage'].idxmax()

print(f"The state with the highest percentage of unresolved complaints is:␣
 ↪{max_unresolved_state}")
```

```
The state with the highest percentage of unresolved complaints is: Kansas
```

[ ]:
```
2.5 Provide the percentage of complaints resolved till date, which were␣
 ↪received through the Internet and customer care calls.
```

[43]:
```
# Filter complaints received through Internet and customer care calls
filtered_df = df[(df['Received Via'] == 'Internet') | (df['Received Via'] ==␣
 ↪'Customer Care Call')]

# Group by 'Status' to get the counts
resolved_counts = filtered_df['Status'].value_counts()

# Calculate the percentage of complaints resolved till date
percentage_resolved = (resolved_counts['Closed'] / filtered_df.shape[0]) * 100
```

```
print(f"The percentage of complaints resolved till date for Internet and␣
  ↪Customer Care calls is: {percentage_resolved:.2f}%")
```

```
The percentage of complaints resolved till date for Internet and Customer Care
calls is: 76.75%
```

[ ]: