

## Comcast Telecom Consumer Complaints

**Business Scenario:** 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.

1. Import data into Python environment.

**Code:**

```
#import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from datetime import date
#read in the data set
df = pd.read_csv('Comcast_telecom_complaints_data.csv')
```

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

**Code:**

```
df["Date_month_year"].value_counts().plot();
#Number of customer complaints by month
sns.countplot(x='month', data=df);
#Number of customer complaints by day
dims = (12, 6)
fig, ax = plt.subplots(figsize=dims)
sns.countplot(x='day', data=df);
```

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

**Code:**

```
#To see the most common complaint type, create a Word Cloud to see the most
common words within the complaints
from wordcloud import WordCloud

# Start with one review
text = df['Customer Complaint'][0]
```

```

# Create and generate a word cloud image
wordcloud = WordCloud().generate(text)

# Display the generated image
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()

# Create stopwords list:
stopwords.update(["Comcast", "not",
                  "working", "Xfinity", "Service", "Complaint", "Customer"])

# Generate a word cloud image
wordcloud = WordCloud(stopwords=stopwords, max_font_size=50, max_words=100,
                      background_color="white").generate(text)

# Set figure size
plt.figure(figsize=(20,10))

# Display the generated image
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()

```

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.

**Code:**

```

#Create a new categorical variable with value as Open and Closed
Status_open_closed = []

for Status in df["Status"]:
    if Status == 'Open' or Status == 'Pending':
        Status_open_closed.append('Open')
    else:
        Status_open_closed.append('Closed')

df['Status_open_closed'] = Status_open_closed
df.head()

```

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

**Code:**

```
#Graph of statewise status of complaints
Status_complaints.plot(kind="barh", figsize=(30,50), stacked=True);
plt.legend(fontsize=30)
```

6. Find out which state has the maximum complaints

**Code:**

```
df_State = df.groupby(['State']).size().to_frame().reset_index().rename({0: "Count"},
axis=1)
df_State.sort_values(by=['Count'], ascending=False).head()
#Look for the state with the maximum number of complaints
df.State.max()
```

7. Find out which state has the highest percentage of unresolved complaints

**Code:**

```
Status_complaints['Total'] = Status_complaints['Open'] + Status_complaints['Closed']
Status_complaints['Percent Unresolved'] = (Status_complaints['Open'] /
Status_complaints['Total'])
#Look for the State with the highest percentage of unresolved complaints
Status_complaints.loc[Status_complaints['Percent Unresolved'] ==
Status_complaints['Percent Unresolved'].max()]
```

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

**Code:**

```
df = df.rename({"Received Via": "Received_Via"}, axis=1)
Status_complaints_Received =
df.groupby(["Received_Via", "Status_open_closed"]).size().unstack()
Status_complaints_Received
#Determine the percent resolved complaints from Customer Care Call and Internet
Status_complaints_Received['Total'] = Status_complaints_Received['Open'] +
Status_complaints_Received['Closed']
Status_complaints_Received['Percent Resolved'] =
(Status_complaints_Received['Closed'] / Status_complaints_Received['Total'])
```