

Comcast Telecom Consumer Complaints

Table of Contents

<i>Introduction</i>	<i>- 2 -</i>
<i>Problem Statement.....</i>	<i>- 2 -</i>
<i>Overview</i>	<i>- 2 -</i>
<i>Phases of Project Work.....</i>	<i>- 3 -</i>
<i>Data Dictionary</i>	<i>- 3 -</i>
<i>Source Code.....</i>	<i>- 3 -</i>
<i>Conclusion</i>	<i>- 7 -</i>

Introduction

This is the project report for the Project 3: Comcast Telecom Consumer Complaint” present as part to the “Data Science with Python” course of Simplilearn.

- Project submitted by : Krishnaveni Rajan
- Email Id: krishnaveni07.rajan@gmail.com
- Submission Date: 03-May-2021
- Language Used : Python

Problem Statement

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. Using the dataset, help to pin down what is wrong with Comcast's customer service with the following information

1. Provide the trend chart for the number of complaints at monthly and daily granularity levels.
2. Provide a table with the frequency of complaint types.
3. Which complaint types are maximum i.e., around internet, network issues, or across any other domains.
4. 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?”
5. Which state has the highest percentage of unresolved complaints
6. Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

Overview

The analysis request by the Comcast Telecom is to provide the insights of the complaints based on various factor with the data provided for the year of 2015. This is requires the “**Descriptive Analytics**” to be done to analyse the data from the historic data and

provide the insights about what has happened in the past. This information of Descriptive Analytics could be used the company to come-up with the corrective actions, improvements required and identify the gap in the existing process.

Phases of Project Work

The project work has been done using the following phases

1. Understanding the Problem statement
2. Review the Dataset to understand the data provided
3. Identify the columns in the Dataset that requires Data Wrangling
4. Break the analysis required for the solution into smaller chunks
 - a. Come up with the step-by-step activity to be done (without starting the coding)
5. Write the Python code
 - a. Execute the code and get results
 - b. Ensure there is no errors
6. Complete the project and submit for Grading.

Data Dictionary

Data dictionary of the Comcast Telecom Complaints data.csv is as follows

- 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 : Confirm if the ticket is filled by the customer directly or being filed on on-behalf of others

Source Code

Source code of the project is as follows

```

#import libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import rlcompleter, readline
readline.parse_and_bind('tab:complete')

import sys

# Read the csv file from the local storage and load as dataframe. Validate the total number of rows, columns and
view the 5 rows of the data

Comcast = pd.read_csv('/Users/renu/Learnings/Data Science With Python/Projects/Comcast Telecom Consumer
Complaints/Comcast_telecom_complaints_data.csv')

count_row = Comcast.shape[0] # Gives number of rows
count_col = Comcast.shape[1]

print("\nTotal Columns =", count_col, "\nTotal Records =", count_row, "\n\n")
print("View of 5 rows of the data frame is as follows:\n")
Comcast.head(5)


#Trend Chart for the number of Complaints in the Monthly and Daily Granularity levels.
# Add 2 new columns named Month and Date to generate the trend graph
Comcast['Months']=pd.to_datetime(Comcast['Date_month_year']).dt.month_name()
Comcast['Dates']=pd.to_datetime(Comcast['Date_month_year']).dt.day


#Group the Complaints based on date and represent in the bar chart
Comcast.groupby(['Dates'])['Customer Complaint'].count().plot(kind='bar', color='red')

plt.title('Daily Trend of Complaints')
plt.ylabel('Complaints Count')
plt.xlabel('Complaint raised Date')
plt.show()


#Group the Complaints based on motnh and represent in the bar chart
Comcast.groupby(['Months'])['Customer Complaint'].count().plot(kind='bar',color='orange')

plt.title('Monthly Trend of Complaints')
plt.ylabel('Complaints Count')
plt.xlabel('Complaint raised Month')

```

```

plt.show()

# Table from the Comcast Complaint based on the Frequency of Complaints
print("Frequency of Complaints on Comcast Telecome is as follows:")
Comcast['Customer Complaint'].value_counts().to_frame().reset_index()

# Top 5 of Complaints by "Maximum complaints based on Complaint type or Domain
print("Top 5: Maximum complaints based on Complaint type or Domain:\n")
Comcast['Customer Complaint'].value_counts().head(5)

# Create a new categorical variable with value as Open and Closed. Open & Pending is to be categorized as Open.
Closed & Solved is to be categorized as Closed.
Comcast['Status']=Comcast['Status'].apply(lambda x: 'Open' if ((x=='Open') | (x=='Pending')) else 'Closed' )
print("Total Complaints based on Status is as follows", Comcast['Status'].value_counts().to_frame())
# Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3.

print("\t INSIGHTS FOR THE STATE WISE COMPLAINTS IN OPEN STATUS AND RESOLVED TICKET BY
RECEIVED TYPE FOR Q3 of 2015 \n")

# Add the Quarter Column based on the Date in the Comcast
Comcast['Quarter']=pd.to_datetime(Comcast['Date_month_year']).dt.quarter

# Filter and create the DataFrame for Q3
Comcast_Q3 = Comcast[Comcast.Quarter.eq(3)]

# Group the State Wise data for the Status 'Open' and 'Closed'
Open=Comcast_Q3[Comcast_Q3['Status']=='Open'].groupby(['State'])['Status'].count().to_frame().reset_index()
Closed=Comcast_Q3[Comcast_Q3['Status']=='Closed'].groupby(['State'])['Status'].count().to_frame().reset_index()

# Plot the Stacked Graph for the State wise Open and Closed Complaints for Q3
fig=plt.figure(figsize=(10,10))
plt.barh(Closed.State, Closed.Status,color='green')
plt.barh(Open.State, Open.Status,color='red')
plt.ylabel("State Name", size=15)
plt.xlabel("Complaint Count")

```

```

plt.legend(["Closed", "Open"])
plt.title("State Wise Open and Closed Complaints in Q3")
plt.show()

# Finding State with the Maximum number of Open Compliants using max and iloc function
np.where(Open.Status==Open.Status.max())
print("STATE WITH MAXIMUM OPEN TICKETS \n")
Open.iloc[8]

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

round((Comcast_Q3[Comcast_Q3['Status']=='Closed'].groupby('Status')['Received
Via'].value_counts(normalize=True)*100),2)

print("\n\t INSIGHTS FOR THE STATE WISE COMPLAINTS IN OPEN STATUS AND RESOLVED TICKET BY
RECEIVED TYPE FOR THE YEAR 2015 \n")

# Group the State Wise data for the Status 'Open' and 'Closed'
Open=Comcast[Comcast['Status']=='Open'].groupby(['State'])['Status'].count().to_frame().reset_index()
Closed=Comcast[Comcast['Status']=='Closed'].groupby(['State'])['Status'].count().to_frame().reset_index()

# Plot the Stacked Graph for the State wise Open and Closed Complaints for Q3
fig=plt.figure(figsize=(10,10))
plt.barh(Closed.State, Closed.Status,color='green')
plt.barh(Open.State, Open.Status,color='orange')
plt.ylabel("State Name", size=15)
plt.xlabel("Complaint Count")
plt.legend(["Closed", "Open"])
plt.title("State Wise Open and Closed Complaints")
plt.show()

# Finding State with the Maximum number of Open Compliants using max and iloc function
np.where(Open.Status==Open.Status.max())

```

```

print("STATE WITH MAXIMUM OPEN TICKETS \n")

Open.iloc[8]

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

print("\nPERCENTAGE OF COMPLAINTS RESOLVED RECIEVED THROUGH INTERNET & CUSTOMER
CARE CALLS\n")

round(((Comcast[Comcast['Status']=='Closed'].groupby('Status')['Received
Via'].value_counts(normalize=True)*100),2)

```

Conclusion

With the data analysis done on the dataset the ‘Descriptive Analytics’ inference is as follows

1. Total tickets raised in the year 2015 was 2224. In the year of 2015
2. On 6th Date there was maximum number of complaints in the Daily granularity.
3. June motnh had the maximum number of complaints in the Monthly granularity.
4. The Top 5 category of the complaint type and their frequency was as follows
 - a. Comcast 83
 - b. Comcast Internet 18
 - c. Comcast Data Cap 17
 - d. comcast 13
 - e. Data Caps 11
5. In-sights for Q3 of 2015
 - a. **Utah** had the maximum number tickets in “Open” Status which counted to 1.
 - b. Out of the total 2224 tickets raised, “Resolved %” of tickets in Q3 was as follows
 - i. Customer Care was 56.41%
 - ii. Internet was 43.59%
6. In-sights for the year 2015
 - a. **Georgia** had the maximum number tickets in “Open” Status which counted to 80.
 - b. Out of the total 2224 tickets raised, “Resolved %” of tickets in Q3 was as follows
 - i. Customer Care was 50.62%
 - ii. Internet was 49.38%

NOTE: The Write-up and Snapshots of the result are shared in separate documents.