COMCAST TELECOM CONSUMER COMPLAINTS ANALYSIS

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.

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 Analysis Task

To perform these tasks, you can use any of the different Python libraries such as NumPy, SciPy, Pandas, scikit-learn, matplotlib, and BeautifulSoup.

- 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.

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 complaints

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

The analysis results to be provided with insights wherever applicable.

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

In [3]:
```
df = pd.read_csv('C:/Users/admin/Documents/Data Science with python projects/Comc
```

In [4]:
```
df.head()
```

Out[4]:

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

In [5]:
```
df[df.isnull()].count()
```

Out[5]:
```
Customer Complaint             0
Date                           0
Date_month_year                0
Time                           0
Received Via                   0
City                           0
State                          0
Zip code                       0
Status                         0
Filing on Behalf of Someone    0
dtype: int64
```

In [6]:
```
df.describe(include='all')
```

Out[6]:

	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Statu
count	2224	2224	2224	2224	2224	2224	2224	2224.000000	222
unique	1841	91	91	2190	2	928	43	NaN	
top	Comcast	24-06-15	24-Jun-15	2:13:31 PM	Customer Care Call	Atlanta	Georgia	NaN	Solve
freq	83	218	218	2	1119	63	288	NaN	97
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	47994.393435	Na
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	28885.279427	Na
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1075.000000	Na
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	30056.500000	Na
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	37211.000000	Na
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	77058.750000	Na
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	99223.000000	Na

EDA and Cleanup the data set

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2224 entries, 250635 to 363614
Data columns (total 10 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   Customer Complaint           2224 non-null   object
 1   Date                         2224 non-null   object
 2   Date_month_year              2224 non-null   object
 3   Time                         2224 non-null   object
 4   Received Via                 2224 non-null   object
 5   City                         2224 non-null   object
 6   State                        2224 non-null   object
 7   Zip code                     2224 non-null   int64 
 8   Status                       2224 non-null   object
 9   Filing on Behalf of Someone  2224 non-null   object
dtypes: int64(1), object(9)
memory usage: 191.1+ KB
```

In [8]:
```
df['Date_month_year'] = pd.to_datetime(df['Date_month_year'])
df['Created_Month'] =  df['Date_month_year'].apply(lambda x: x.month)
df['Created_Day'] = df['Date_month_year'].apply(lambda x: x.day)
df['Created_Day of Week'] = df['Date_month_year'].apply(lambda x: x.dayofweek)
```

In [9]:
```
dmap = {0:'Mon',1:'Tue',2:'Wed',3:'Thur',4:'Fri',5:'Sat',6:'Sun'}
df['Created_Day of Week']=df['Created_Day of Week'].map(dmap)
df.head(5)
```

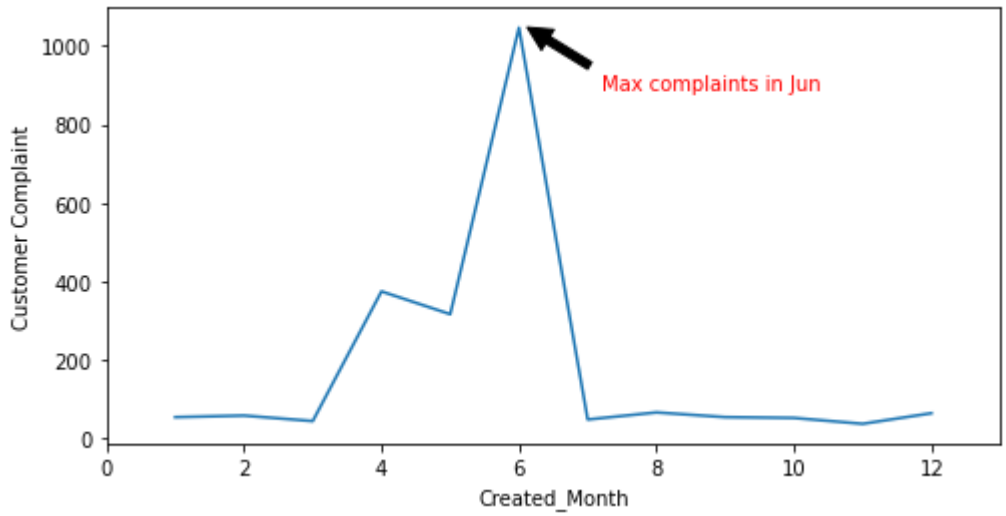
Out[9]:

Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Statu
250635	Comcast Cable Internet Speeds	22-04-15	2015-04-22	3:53:50 PM	Customer Care Call	Abingdon	Maryland	21009	Close
223441	Payment disappear - service got disconnected	04-08-15	2015-08-04	10:22:56 AM	Internet	Acworth	Georgia	30102	Close
242732	Speed and Service	18-04-15	2015-04-18	9:55:47 AM	Internet	Acworth	Georgia	30101	Close
277946	Comcast Imposed a New Usage Cap of 300GB that ...	05-07-15	2015-07-05	11:59:35 AM	Internet	Acworth	Georgia	30101	Ope
307175	Comcast not working and no service to boot	26-05-15	2015-05-26	1:25:26 PM	Internet	Acworth	Georgia	30101	Solve

number of complaints monthly

In [10]:
```
plt.figure(figsize=(8,4))
bymonth = df.groupby('Created_Month').count().reset_index()
lp = sns.lineplot(x='Created_Month', y= 'Customer Complaint', data = bymonth, sor
ax = lp.axes
ax.set_xlim(0,13)
ax.annotate('Max complaints in Jun', color='red',
            xy=(6, 1060), xycoords='data',
            xytext=(0.8, 0.85), textcoords='axes fraction',
            arrowprops=dict(facecolor='black', shrink=0.1),
            horizontalalignment='right', verticalalignment='top')
```

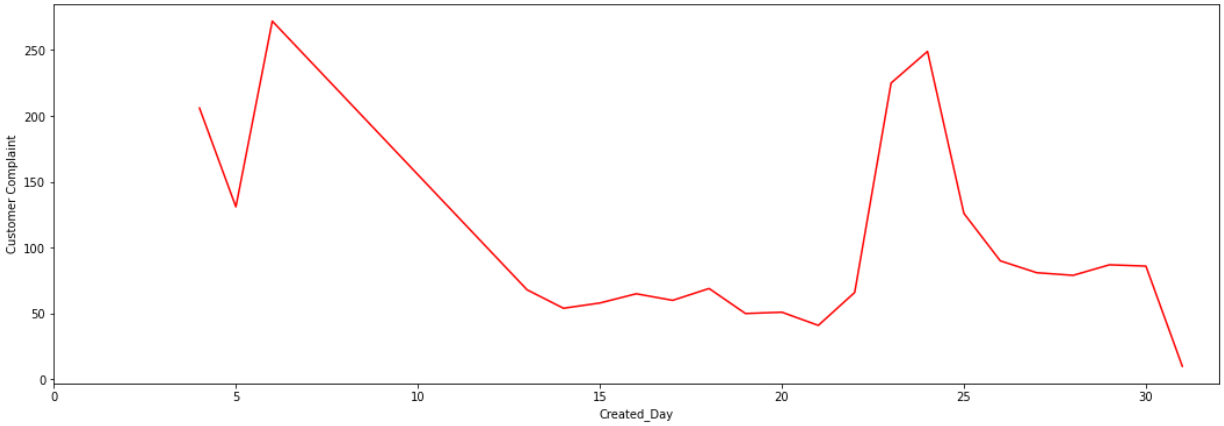
Out[10]: Text(0.8, 0.85, 'Max complaints in Jun')



number of complaints Daily

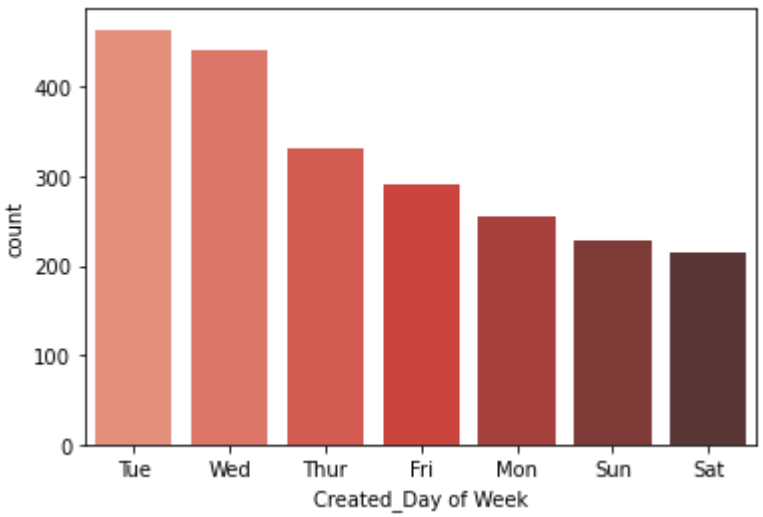
In [11]:
```
plt.figure(figsize=(18,6))
byday = df.groupby('Created_Day').count().reset_index()
lp = sns.lineplot(x='Created_Day', y= 'Customer Complaint', data = byday, sort=Fa
ax = lp.axes
ax.set_xlim(0,32)
```

Out[11]: (0.0, 32.0)



In [12]:
```
#number of complaints based on created day of the week
sns.countplot(x='Created_Day of Week', data = df, order=df['Created_Day of Week']
#More number of complaints on Tuesday and wednesday
```

Out[12]: <AxesSubplot:xlabel='Created_Day of Week', ylabel='count'>



TASK 2 - Provide a table with the frequency of complaint types.

In [13]:
```
df['Customer Complaint'] = df['Customer Complaint'].str.title()
CT_freq = df['Customer Complaint'].value_counts()
CT_freq
```

Out[13]:
```
Comcast                                            102
Comcast Data Cap                                    30
Comcast Internet                                    29
Comcast Data Caps                                   21
Comcast Billing                                     18
                                                  ...
Comcast- Internet                                    1
Data Capping And Lack Of Options In Tucson Az        1
Comcast Lied About Pricing And Installation          1
Comcast Not Honoring Agreement                       1
Misleading Sales Practice And Advertising            1
Name: Customer Complaint, Length: 1740, dtype: int64
```

In [14]:
```
import nltk
%pip install wordcloud
```

```
Collecting wordcloud
  Downloading wordcloud-1.8.1-cp38-cp38-win_amd64.whl (155 kB)
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-
packages (from wordcloud) (3.3.2)
Requirement already satisfied: numpy>=1.6.1 in c:\programdata\anaconda3\lib\sit
e-packages (from wordcloud) (1.19.2)
Requirement already satisfied: pillow in c:\programdata\anaconda3\lib\site-pack
ages (from wordcloud) (8.0.1)
Requirement already satisfied: cycler>=0.10 in c:\programdata\anaconda3\lib\sit
e-packages (from matplotlib->wordcloud) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\programdata\anaconda3\li
b\site-packages (from matplotlib->wordcloud) (1.3.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\p
rogramdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.4.7)
Requirement already satisfied: certifi>=2020.06.20 in c:\programdata\anaconda3
\lib\site-packages (from matplotlib->wordcloud) (2020.6.20)
Requirement already satisfied: python-dateutil>=2.1 in c:\programdata\anaconda3
\lib\site-packages (from matplotlib->wordcloud) (2.8.1)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-package
s (from cycler>=0.10->matplotlib->wordcloud) (1.15.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.8.1
Note: you may need to restart the kernel to use updated packages.
```

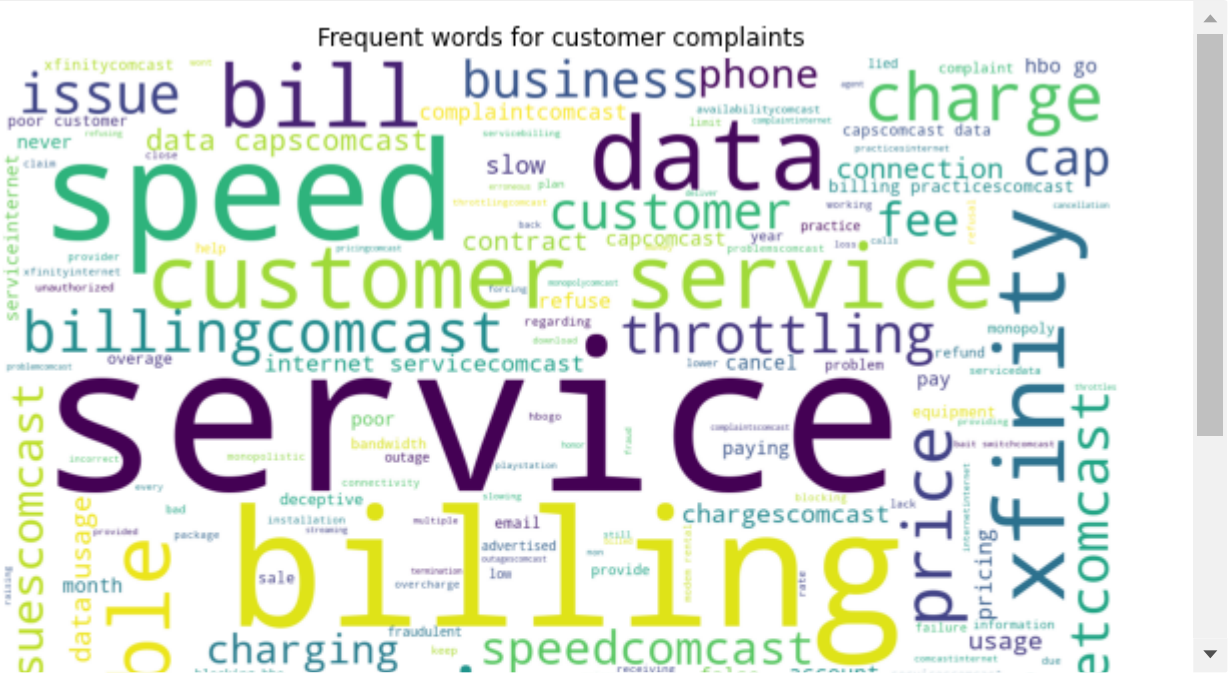
In [15]:
```
from wordcloud import WordCloud, STOPWORDS
common_complaints = df['Customer Complaint'].dropna().tolist()
common_complaints =''.join(common_complaints).lower()

list_stops = ('Comcast','Now','Company','Day','Someone','Thing','Also','Got','Way

for word in list_stops:
    STOPWORDS.add(word)
```

In [16]:
```
wordcloud = WordCloud(stopwords=STOPWORDS,
                      background_color='white',
                      width=1200,
                      height=1000).generate(common_complaints)
```

In [16]:
```
plt.figure( figsize=(10,12) )
plt.imshow(wordcloud)
plt.title('Frequent words for customer complaints')
plt.axis('off')
plt.show()
#Internet complaints are Maximum
```



In [28]:
```
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
import string
nltk.download("stopwords")
stop = set(stopwords.words('english'))
exclude = set(string.punctuation)
lemma = WordNetLemmatizer()
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

In [21]:
```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\wordnet.zip.
```

Out[21]:
```
True
```

In [29]:
```
def clean(doc):
    stop_free = " ".join([i for i in doc.lower().split() if i not in stop])
    punc_free = "".join([ch for ch in stop_free if ch not in exclude])
    normalized = " ".join(lemma.lemmatize(word) for word in punc_free.split())
    return normalized
```

In [30]:
```
doc_complete = df['Customer Complaint'].tolist()
doc_clean = [clean(doc).split() for doc in doc_complete]
```

In [35]:
```
pip install gensim==4.0.1
```

```
Collecting gensim==4.0.1
  Downloading gensim-4.0.1-cp38-cp38-win_amd64.whl (23.9 MB)
Requirement already satisfied: Cython==0.29.21 in c:\programdata\anaconda3\lib
\site-packages (from gensim==4.0.1) (0.29.21)
Collecting smart-open>=1.8.1
  Downloading smart_open-5.1.0-py3-none-any.whl (57 kB)
Requirement already satisfied: numpy>=1.11.3 in c:\programdata\anaconda3\lib\si
te-packages (from gensim==4.0.1) (1.19.2)
Requirement already satisfied: scipy>=0.18.1 in c:\programdata\anaconda3\lib\si
te-packages (from gensim==4.0.1) (1.5.2)
Installing collected packages: smart-open, gensim
Successfully installed gensim-4.0.1 smart-open-5.1.0
Note: you may need to restart the kernel to use updated packages.
```

In [36]:
```
import gensim
from gensim import corpora
```

```
C:\ProgramData\Anaconda3\lib\site-packages\gensim\similarities\__init__.py:15:
UserWarning: The gensim.similarities.levenshtein submodule is disabled, because
the optional Levenshtein package <https://pypi.org/project/python-Levenshtein/>
is unavailable. Install Levenhstein (e.g. `pip install python-Levenshtein`) to
suppress this warning.
  warnings.warn(msg)
```

In [37]:
```
dictionary = corpora.Dictionary(doc_clean)
dictionary
```

Out[37]:
```
<gensim.corpora.dictionary.Dictionary at 0x25ae2b8a370>
```

In [44]:
```
doc_term_matrix = [dictionary.doc2bow(doc) for doc in doc_clean]
doc_term_matrix
```

```
 [(259, 1), (447, 1), (448, 1), (449, 1)],
 [(1, 1), (155, 1)],
 [(199, 1), (252, 1), (432, 1), (450, 1), (451, 1)],
 [(1, 1), (57, 1), (384, 1)],
 [(1, 1), (2, 1), (225, 1), (452, 1)],
 [(95, 1), (103, 1), (225, 1), (453, 1), (454, 1)],
 [(1, 1), (455, 1), (456, 1), (457, 1), (458, 1), (459, 1)],
 [(38, 1), (112, 1)],
 [(1, 1), (57, 1)],
 [(1, 1)],
 [(2, 1), (3, 1), (299, 1)],
 [(1, 1), (52, 1), (115, 1), (460, 1)],
 [(1, 1), (82, 1), (86, 1)],
 [(1, 1), (2, 1)],
 [(1, 1)],
 [(1, 1), (8, 1), (72, 1), (210, 1), (461, 1), (462, 1)],
 [(1, 1), (463, 1), (464, 1)],
 [(1, 1), (2, 1)],
 [(8, 1), (66, 1), (199, 1), (465, 1), (466, 1), (467, 1)],
 [(2, 1), (8, 1), (38, 1), (57, 1), (72, 1), (97, 1), (210, 1)],
```

In [45]:
```
from gensim.models import LdaModel
```

In [46]:
```
num_topic = 9
ldamodel = LdaModel(doc_term_matrix,num_topics=num_topic,id2word = dictionary,pas
```

In [47]:
```
topics = ldamodel.show_topics()
for topic in topics:
    print(topic)
    print()
```

```
(0, '0.203*"comcast" + 0.120*"complaint" + 0.046*"bill" + 0.033*"charged" + 0.0
26*"without" + 0.022*"lack" + 0.018*"credit" + 0.017*"phone" + 0.016*"signal" +
0.014*"option"')

(1, '0.226*"service" + 0.157*"comcast" + 0.078*"internet" + 0.040*"customer" +
0.022*"poor" + 0.013*"terrible" + 0.011*"2" + 0.011*"problem" + 0.008*"broadban
d" + 0.008*"misleading"')

(2, '0.050*"comcast" + 0.049*"get" + 0.046*"service" + 0.030*"unreliable" + 0.0
28*"email" + 0.020*"pay" + 0.019*"refusal" + 0.019*"10" + 0.018*"disconnection"
+ 0.017*"improper"')

(3, '0.117*"speed" + 0.089*"comcast" + 0.062*"charge" + 0.062*"internet" + 0.02
6*"paying" + 0.022*"service" + 0.020*"fee" + 0.018*"without" + 0.018*"promised"
+ 0.015*"cramming"')

(4, '0.135*"comcast" + 0.132*"data" + 0.107*"cap" + 0.059*"issue" + 0.044*"inte
rnet" + 0.029*"cable" + 0.025*"usage" + 0.023*"throttling" + 0.016*"bill" + 0.0
15*"xfinity"')

(5, '0.057*"false" + 0.045*"switch" + 0.041*"contract" + 0.040*"account" + 0.03
6*"payment" + 0.032*"advertising" + 0.030*"bait" + 0.023*"comcast" + 0.019*"mon
th" + 0.016*"check"')

(6, '0.191*"internet" + 0.126*"comcast" + 0.047*"slow" + 0.033*"speed" + 0.026
*"xfinity" + 0.025*"connection" + 0.023*"deceptive" + 0.018*"monopoly" + 0.018
*"business" + 0.016*"high"')

(7, '0.211*"billing" + 0.089*"comcast" + 0.079*"practice" + 0.063*"unfair" + 0.
048*"pricing" + 0.028*"comcastxfinity" + 0.025*"day" + 0.021*"monopolistic" +
0.020*"back" + 0.020*"show"')

(8, '0.074*"comcast" + 0.060*"billing" + 0.034*"service" + 0.030*"help" + 0.027
*"failure" + 0.025*"price" + 0.024*"year" + 0.022*"incorrect" + 0.021*"refund"
+ 0.020*"contract"')
```

In [48]:
```
word_dict = {}
for i in range(num_topic):
    words = ldamodel.show_topic(i,topn = 20)
    word_dict['Topic '+"{}".format(i)]=[i[0] for i in words]
```

In [49]:
```
pd.DataFrame(word_dict)
```

Out[49]:

	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	
0	comcast	service	comcast	speed	comcast	false	internet	
1	complaint	comcast	get	comcast	data	switch	comcast	(
2	bill	internet	service	charge	cap	contract	slow	
3	charged	customer	unreliable	internet	issue	account	speed	
4	without	poor	email	paying	internet	payment	xfinity	
5	lack	terrible	pay	service	cable	advertising	connection	comca
6	credit	2	refusal	fee	usage	bait	deceptive	
7	phone	problem	10	without	throttling	comcast	monopoly	mon
8	signal	broadband	disconnection	promised	bill	month	business	
9	option	misleading	improper	cramming	xfinity	check	high	
10	rate	please	inability	installation	connectivity	continues	home	appo
11	throttled	bad	area	modem	limit	extortion	price	
12	change	horrible	streaming	12	billed	att	charge	ag
13	outage	price	inconsistent	shitty	monthly	monopolistic	intermittent	co
14	notice	quality	paid	low	xfinitycomcast	egregious	service	
15	request	xfinity	power	throttling	increased	person	sale	
16	isp	access	neighborhood	unauthorized	several	advertisingbait	bill	
17	transfer	overcharge	monopolist	equipment	mb	breach	plan	
18	consent	claim	transferred	ps4	charging	bullying	loss	r
19	higher	hbo	subsequent	hbogo	overage	terminating	outage	

TASK 3 - 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.

In [63]:
```
df['Highlevel_Status'] = ["Open" if Status=="Open" or Status=="Pending" else "Clo
```

In [64]:
```
df['Highlevel_Status'].unique()
```

Out[64]:
```
array(['Closed', 'Open'], dtype=object)
```

TASK 4 - Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3.

In [65]:
```
df['State'] = df['State'].str.title()
st_cmp = df.groupby(['State','Highlevel_Status']).size().unstack().fillna(0)
```

In [66]:
```
st_cmp
```

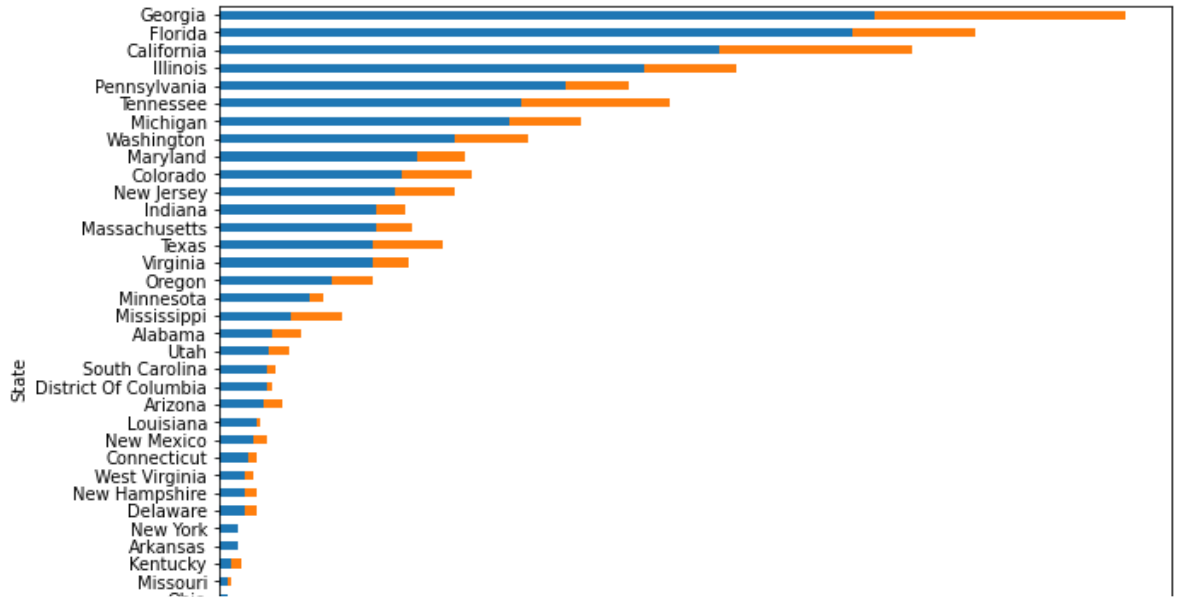
Out[66]:

Highlevel_Status	Closed	Open
State		
Alabama	17.0	9.0
Arizona	14.0	6.0
Arkansas	6.0	0.0
California	159.0	61.0
Colorado	58.0	22.0
Connecticut	9.0	3.0
Delaware	8.0	4.0
District Of Columbia	15.0	2.0
Florida	201.0	39.0
Georgia	208.0	80.0
Illinois	135.0	29.0
Indiana	50.0	9.0
Iowa	1.0	0.0
Kansas	1.0	1.0
Kentucky	4.0	3.0
Louisiana	12.0	1.0
Maine	3.0	2.0
Maryland	63.0	15.0
Massachusetts	50.0	11.0
Michigan	92.0	23.0
Minnesota	29.0	4.0
Mississippi	23.0	16.0
Missouri	3.0	1.0
Montana	1.0	0.0
Nevada	1.0	0.0
New Hampshire	8.0	4.0
New Jersey	56.0	19.0
New Mexico	11.0	4.0
New York	6.0	0.0
North Carolina	3.0	0.0
Ohio	3.0	0.0
Oregon	36.0	13.0
Pennsylvania	110.0	20.0

Highlevel_Status	Closed	Open
State		
Rhode Island	1.0	0.0
South Carolina	15.0	3.0
Tennessee	96.0	47.0
Texas	49.0	22.0
Utah	16.0	6.0
Vermont	2.0	1.0
Virginia	49.0	11.0
Washington	75.0	23.0
West Virginia	8.0	3.0

In [35]:
```
st_cmp.sort_values('Closed',axis = 0,ascending=True).plot(kind="barh", figsize=(1
```

Out[35]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7113f85978>
```



TASK 5 - Which state has the maximum complaints Which state has the highest percentage of unresolved complaints

In [67]:
```
df.groupby(["State"]).size().sort_values(ascending=False).to_frame().rename({0: '
#Georgia has highest complaints
```

Out[67]:

	Complaint count
State	
Georgia	288

In [68]:
```
CT = df.groupby(["State","Highlevel_Status"]).size().unstack().fillna(0)
CT.sort_values('Closed',axis = 0,ascending=False)[:1]
```

Out[68]:

Highlevel_Status	Closed	Open
State		
Georgia	208.0	80.0

In [69]:
```
#highest percentage of unresolved complaints
CT['Resolved_cmp_prct'] = CT['Closed']/CT['Closed'].sum()*100
CT['Unresolved_cmp_prct'] = CT['Open']/CT['Open'].sum()*100
```

In [70]:
```
CT.sort_values('Unresolved_cmp_prct',axis = 0,ascending=False)[:1]
#Georgia state has highest Unresolved complaints when compared to other states
```

Out[70]:

Highlevel_Status	Closed	Open	Resolved_cmp_prct	Unresolved_cmp_prct
State				
Georgia	208.0	80.0	12.18512	15.473888

TASK 6 --- Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

In [71]:
```
cr = df.groupby(['Received Via','Highlevel_Status']).size().unstack().fillna(0)
cr['resolved'] = cr['Closed']/cr['Closed'].sum()*100
cr['resolved']
```

Out[71]:
```
Received Via
Customer Care Call    50.615114
Internet              49.384886
Name: resolved, dtype: float64
```

In [41]:
```
#df["item"].value_counts().nlargest(n=1).values[0]
```