import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from matplotlib import style

style.use('fivethirtyeight')

import pandas as pd

date\_fields = [['Date', 'Time']]

comcast\_file=r"C:\Users\hii\Desktop\datascience\Comcast\_telecom\_complaints\_data.csv"

comcast\_df = pd.read\_csv(comcast\_file, parse\_dates = date\_fields, dayfirst = True, keep\_date\_col = True)

**Analysis Task**

**Provide the trend chart for the number of complaints at monthly and daily granularity levels converting date month year column to date object**

comcast\_df.Date\_month\_year = pd.to\_datetime(comcast\_df.Date\_month\_year)

print(comcast\_df.Date\_month\_year)

comcast\_df.Time = pd.to\_datetime(comcast\_df.Time,format='%I:%M:%S %p')

print(comcast\_df.Time)

**The data set is for year 2015**

year = pd.Series([val.year for val in comcast\_df.Date\_month\_year])

year.value\_counts()

**Monthly Trend Chart**

**June month has maximum complaints**

comcast\_df['Month'] = comcast\_df.Date\_month\_year.dt.month

comcast\_df['Month']

comcast\_df.groupby('Month').count()['Ticket #'].plot(kind='bar')

**Daily basis trend chart**

**24 June as maximum complaint**

comcast\_df['Customer Complaint'].groupby(comcast\_df.Date).count().plot(kind='bar',figsize=(22,6))

**Frequency of Complaints**

**large no of complaints are registered in the name of Comcast which cannot be determined. The second maximum no of complaints are in the name of Internet**

comcast\_df['Customer Complaint'].value\_counts()

**New categorical variable with value as Open and Closed**

**Data categorised as Open for all open and pending status**

**Data categorised as Closed for all closed and solved status**

comcast\_df["OC\_Status"] = comcast\_df["Status"].apply(lambda x: "Open" if(x == "Open" or x == "Pending") else "Closed")

print(comcast\_df["OC\_Status"])

comcast\_df["OC\_Status"].unique()

**State wise status of complaints in a stacked bar chart**

**Georgia has maximum complaints**

import seaborn as sns

sns.displot(comcast\_df, x='State', hue='Status', multiple='stack',height=14, aspect=.8)

plt.xticks(rotation=90)

plt.show()

**Highest percentage of unresolved complaints**

**KANSAS has maximum unresolved cases**

**All the opened and pending status are unresolved**

OpenCloseStatus = comcast\_df.groupby(['State',"OC\_Status"])["OC\_Status"].count().unstack().fillna(0)

OpenClose\_df=pd.DataFrame(OpenCloseStatus)

print(OpenClose\_df)

OpenClose\_df["Open Percantage"] = (OpenClose\_df.Open/(OpenClose\_df.Open + OpenClose\_df.Closed)).mul(100).round(1)

OpenClose\_df["Open Percantage"]

OpenClose\_df["Closed Percantage"] = (OpenClose\_df.Closed/(OpenClose\_df.Open + OpenClose\_df.Closed)).mul(100).round(1)

OpenClose\_df["Closed Percantage"]

OpenClose\_df.drop(["Closed"], axis=1, inplace=True)

OpenClose\_df.drop(["Open"], axis=1, inplace=True)

fig, ax = plt.subplots(figsize=(15,4))

plt.xticks(rotation=90)

ax.bar(OpenClose\_df.index, OpenClose\_df["Open Percantage"], color="C0")

**Percentage of complaints resolved till date, which were received through the Internet and customer care calls**

**Closed Percentage of complaints received through call = 77.21%**

**Closed Percentage of complaints received through Internet are 76.29%**

Received\_df=pd.crosstab(comcast\_df["Received Via"],comcast\_df["OC\_Status"]).apply(lambda x: (x\*100/x.sum()).round(2),axis=1)

Received\_df

Received\_df.plot.bar(stacked=True,figsize=(10,4))

plt.legend(title='Call Status')

plt.xticks(rotation=0)

plt.show()