Source Code

Cd

cd Desktop

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read\_csv("Comcast\_telecom\_complaints\_data.csv")

df.head(5)

df["date\_index"] = df["Date\_month\_year"] + " " + df["Time"]

df["date\_index"] = pd.to\_datetime(df["date\_index"])

df["Date\_month\_year"] = pd.to\_datetime(df["Date\_month\_year"])

df.dtypes

df.head()

df = df.set\_index(df["date\_index"])

df.head(5)

df["Date\_month\_year"].value\_counts()

df["Date\_month\_year"].value\_counts().plot();

f = df.groupby(pd.Grouper(freq="M")).size()

f.head()

#monthly chart

df.groupby(pd.Grouper(freq="M")).size().plot()

f = df.groupby(pd.Grouper(freq="D")).size()

f.head()

#Daily chart

df.groupby(pd.Grouper(freq="D")).size().plot()

# frequency of complaint types.

df.groupby(["Customer Complaint"]).size().sort\_values(ascending=False).to\_frame().reset\_index().rename({0: "Count"}, axis=1)

# maximum

df.groupby(["Customer Complaint"]).size().sort\_values(ascending=False).to\_frame().reset\_index().rename({0: "Count"}, axis=1).max()

df["newStatus"] = ["Open" if Status=="Open" or Status=="Pending" else "Closed" for Status in df["Status"]]

df.head()

df.groupby(["State"]).size().sort\_values(ascending=False).to\_frame().reset\_index().rename({0: "Count"}, axis=1)

Status\_complaints = df.groupby(["State","newStatus"]).size().unstack().fillna(0)

Status\_complaints

Status\_complaints.plot(kind="barh", figsize=(30,50), stacked=True)

df.groupby(["State"]).size().sort\_values(ascending=False).to\_frame().reset\_index().rename({0: "Count"}, axis=1).max()

df.groupby(["State","newStatus"]).size().unstack().fillna(0).max()

pip install wordcloud

pip install nltk

from nltk.corpus import stopwords

from nltk.stem.wordnet import WordNetLemmatizer

import string

import nltk

nltk.download('stopwords')

stop = set(stopwords.words('english'))

exclude = set(string.punctuation)

lemma = WordNetLemmatizer()

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])

normalised = " ".join(lemma.lemmatize(word) for word in punc\_free.split())

return normalized

import nltk

nltk.download('wordnet')

doc\_complete = df["Customer Complaint"].tolist()

doc\_clean = [clean(doc).split() for doc in doc\_complete]

pip install genism

import gensim

from gensim import corpora

dictionary = corpora.Dictionary(doc\_clean)

print(dictionary)

doc\_term\_matrix = [dictionary.doc2bow(doc) for doc in doc\_clean]

doc\_term\_matrix

from gensim.models import LdaModel

Num\_Topic = 9

ldamodel = LdaModel(doc\_term\_matrix, num\_topics= Num\_Topic, id2word= dictionary, passes= 30)

topics = ldamodel.show\_topics()

for topic in topics:

print(topic)

print()

word\_dict = {}

for i in range(Num\_Topic):

words = ldamodel.show\_topic(i,topn =30)

word\_dict["Topic # " + "{}".format(i)] = [i[0] for i in words]

pd.DataFrame(word\_dict)