# Libraries

import pandas as pd

from pandas import Series, DataFrame

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('darkgrid')

# Visualization Libraries

import matplotlib

%matplotlib inline

figsize = (10,5)

import seaborn as sns

import datetime

import plotly.graph\_objects as go

import plotly.express as px

import folium

from folium import plugins

%matplotlib inline

import folium

import calendar

from datetime import datetime

csv1 = pd.read\_csv("/content/2020-01-gwent-street.csv")

csv2 = pd.read\_csv("/content/2020-02-gwent-street.csv")

csv3 = pd.read\_csv("/content/2020-03-gwent-street.csv")

csv4 = pd.read\_csv("/content/2020-04-gwent-street.csv")

csv5 = pd.read\_csv("/content/2020-05-gwent-street.csv")

csv6 = pd.read\_csv("/content/2020-06-gwent-street.csv")

csv7 = pd.read\_csv("/content/2020-07-gwent-street.csv")

csv8 = pd.read\_csv("/content/2020-08-gwent-street.csv")

csv9 = pd.read\_csv("/content/2020-09-gwent-street.csv")

csv10 = pd.read\_csv("/content/2020-10-gwent-street.csv")

csv11 = pd.read\_csv("/content/2020-11-gwent-street.csv")

csv12 = pd.read\_csv("/content/2020-12-gwent-street.csv")

csv13 = pd.read\_csv("/content/2021-01-gwent-street.csv")

csv14 = pd.read\_csv("/content/2021-02-gwent-street.csv")

csv15 = pd.read\_csv("/content/2021-03-gwent-street.csv")

csv16 = pd.read\_csv("/content/2021-04-gwent-street.csv")

csv17 = pd.read\_csv("/content/2021-05-gwent-street.csv")

csv18 = pd.read\_csv("/content/2021-06-gwent-street.csv")

csv19 = pd.read\_csv("/content/2021-07-gwent-street.csv")

csv20 = pd.read\_csv("/content/2021-08-gwent-street.csv")

csv21 = pd.read\_csv("/content/2021-09-gwent-street.csv")

csv22 = pd.read\_csv("/content/2021-10-gwent-street.csv")

csv23 = pd.read\_csv("/content/2021-11-gwent-street.csv")

csv24 = pd.read\_csv("/content/2021-12-gwent-street.csv")

csv25 = pd.read\_csv("/content/2022-01-gwent-street.csv")

csv26 = pd.read\_csv("/content/2022-02-gwent-street.csv")

csv27 = pd.read\_csv("/content/2022-03-gwent-street.csv")

csv28 = pd.read\_csv("/content/2022-04-gwent-street.csv")

csv29 = pd.read\_csv("/content/2022-05-gwent-street.csv")

csv30 = pd.read\_csv("/content/2022-06-gwent-street.csv")

csv31 = pd.read\_csv("/content/2022-07-gwent-street.csv")

csv32 = pd.read\_csv("/content/2022-08-gwent-street.csv")

csv33 = pd.read\_csv("/content/2022-09-gwent-street.csv")

csv34 = pd.read\_csv("/content/2022-10-gwent-street.csv")

csv35 = pd.read\_csv("/content/2021-01-metropolitan-street.csv")

csv36 = pd.read\_csv("/content/2021-02-metropolitan-street.csv")

csv37 = pd.read\_csv("/content/2021-03-metropolitan-street.csv")

csv38 = pd.read\_csv("/content/2021-04-metropolitan-street.csv")

csv39 = pd.read\_csv("/content/2021-05-metropolitan-street.csv")

csv40 = pd.read\_csv("/content/2021-06-metropolitan-street.csv")

csv41 = pd.read\_csv("/content/2021-07-metropolitan-street.csv")

csv42 = pd.read\_csv("/content/2021-08-metropolitan-street.csv")

csv43= pd.read\_csv("/content/2021-09-metropolitan-street.csv")

csv44 = pd.read\_csv("/content/2021-10-metropolitan-street.csv")

csv45 = pd.read\_csv("/content/2021-11-metropolitan-street.csv")

csv45 = pd.read\_csv("/content/2021-11-metropolitan-street.csv")

csv46 = pd.read\_csv("/content/2021-12-metropolitan-street.csv")

df\_metro = concate\_data\_metro = pd.concat([csv35,csv36,csv37,csv38,csv39,csv40,csv41,csv42,csv43,csv44,csv45,csv46], ignore\_index=True)

df\_gwent = concate\_data = pd.concat([csv1,csv2,csv3,csv4,csv5,csv6,csv7,csv8,csv9,csv10,csv11,csv12,csv13,csv14,csv15,csv16,csv17,csv18,csv19,csv20,csv21,csv22,csv23,csv24,csv25,csv26,csv27,csv28,csv29,csv30,csv31,csv32,csv33,csv34], ignore\_index=True)

df\_2021 = concate\_data1 = pd.concat([csv13,csv14,csv15,csv16,csv17,csv18,csv19,csv20,csv21,csv22,csv23,csv24], ignore\_index=True)

df\_lockdown = concate\_lock = pd.concat([csv13,csv14,csv15,csv16,csv17,csv18,csv19,csv20,csv21,csv22], ignore\_index=True)

df\_2022 = concate\_data3 = pd.concat([csv25,csv26,csv27,csv28,csv29,csv30,csv31,csv32,csv33,csv34], ignore\_index=True)

df\_2022.shape

df\_2021.shape

df\_lockdown.shape

df\_gwent.shape

df\_metro.shape

year = df\_2021["Month"].values

year = [my\_str.split("-")[0] for my\_str in year]

df\_2021["year"] = year

df\_2021['year'] = pd.DatetimeIndex(df\_2021['year']).year

year = df\_2022["Month"].values

year = [my\_str.split("-")[0] for my\_str in year]

df\_2022["year"] = year

df\_2022 ['year'] = pd.DatetimeIndex(df\_2022['year']).year

year = df\_lockdown["Month"].values

year = [my\_str.split("-")[0] for my\_str in year]

df\_lockdown["year"] = year

df\_lockdown['year'] = pd.DatetimeIndex(df\_lockdown['year']).year

year = df\_metro["Month"].values

year = [my\_str.split("-")[0] for my\_str in year]

df\_metro["year"] = year

df\_metro['year'] = pd.DatetimeIndex(df\_metro['year']).year

year = df\_gwent["Month"].values

year = [my\_str.split("-")[0] for my\_str in year]

df\_gwent["year"] = year

df\_gwent['year'] = pd.DatetimeIndex(df\_gwent['year']).year

df\_gwent

df\_gwent['Month'] = pd.to\_datetime(df\_gwent['Month']).dt.strftime(('%B'))

df\_2021['Month'] = pd.to\_datetime(df\_2021['Month']).dt.strftime('%B')

df\_2022['Month'] = pd.to\_datetime(df\_2022['Month']).dt.strftime('%B')

df\_metro['Month'] = pd.to\_datetime(df\_metro['Month']).dt.strftime('%B')

df\_lockdown['Month'] = pd.to\_datetime(df\_lockdown['Month']).dt.strftime(('%B'))

df\_gwent= df\_gwent.rename(columns={"Crime ID":"crime\_id","Crime type":"crime\_type","Reported by":"reported\_by","Falls within":"falls\_within","LSOA name":"LSOA.name","Last outcome category":"last\_outcome\_category"})

df\_gwent

df\_2021 = df\_2021.rename(columns={"Crime ID":"crime\_id","Crime type":"crime\_type","LSOA name":"LSOA.name","Last outcome category":"last\_outcome\_category"})

df\_2022 = df\_2022.rename(columns={"Crime ID":"crime\_id","Crime type":"crime\_type","LSOA name":"LSOA.name","Last outcome category":"last\_outcome\_category"})

df\_lockdown = df\_lockdown.rename(columns={"Crime ID":"crime\_id","Crime type":"crime\_type","LSOA name":"LSOA.name","Last outcome category":"last\_outcome\_category"})

df\_metro = df\_metro.rename(columns={"Crime ID":"crime\_id","Crime type":"crime\_type","LSOA name":"LSOA.name","Last outcome category":"last\_outcome\_category"})

df\_gwent.dtypes

df\_2021.dtypes

df\_gwent.tail()

df\_lockdown.tail()

df\_gwent.info()

df\_gwent.nunique()

df\_gwent.isnull().sum()

df\_2021.isnull().sum()

df\_2022.isnull().sum()

df\_lockdown.isnull().sum()

df\_metro.isnull().sum()

import matplotlib

matplotlib.rcParams['figure.figsize'] = (10,6)

sns.heatmap(df\_gwent.isnull(),yticklabels = False, cbar = False , cmap = 'viridis')

plt.title("Missing null values")

missing\_percentages = (df\_gwent.isna().sum().sort\_values(ascending=False)/(len(df\_gwent)))\*100

missing\_percentages

missing\_percentages = missing\_percentages[missing\_percentages != 0]

import matplotlib

matplotlib.rcParams['figure.figsize'] = (10,6)

missing\_percentages.plot(kind = 'barh')

plt.title("Missig percentage of null values")

data\_gwent = df\_gwent.drop(df\_gwent.columns[[0, 2, 3, 7, 8, 10,11]],axis = 1)

data\_gwent

data\_2021 = df\_2021.drop(df\_2021.columns[[0, 2, 3, 7, 8, 10,11]],axis = 1)

data\_2021

Data\_2021= data\_2021.fillna(0)

Data\_2021

data\_2022 = df\_2022.drop(df\_2022.columns[[0, 2, 3, 7, 8, 10,11]],axis = 1)

data\_lockdown = df\_lockdown.drop(df\_lockdown.columns[[0, 2, 3, 7, 8, 10, 11]],axis = 1)

data\_metro = df\_metro.drop(df\_metro.columns[[0, 7, 8, 10,11]],axis = 1)

df\_Lockdown = data\_lockdown.fillna(0)

data\_Gw=data\_gwent.fillna(0)

data\_Gw

Data\_2022=data\_2022.fillna(0)

Data\_2022

Data\_metro = data\_metro.fillna(0)

Data\_metro

data\_Gw.describe()

Data\_metro.describe()

data\_Gw['crime\_type'].value\_counts()

data\_Gw['crime\_type'].value\_counts()/(len(data\_Gw))\*100

Data\_2021.groupby(['year','crime\_type']).size()

Data\_metro.groupby(['year','crime\_type']).size()

crimes\_percentages\_2021 = Data\_2021['crime\_type'].value\_counts()/(len(Data\_2021))\*100

crimes\_percentages\_2021

crimes\_percentages\_metro = Data\_metro['crime\_type'].value\_counts()/(len(Data\_metro))\*100

crimes\_percentages\_metro

crimes\_percentages\_2021 = crimes\_percentages\_2021[crimes\_percentages\_2021 != 0]

import matplotlib

matplotlib.rcParams['figure.figsize'] = (10,6)

crimes\_percentages\_2021.plot(kind = 'barh')

plt.title("crimes\_percentages\_2021")

crimes\_percentages\_metro = crimes\_percentages\_metro[crimes\_percentages\_metro != 0]

import matplotlib

matplotlib.rcParams['figure.figsize'] = (10,6)

crimes\_percentages\_metro.plot(kind = 'barh')

plt.title("crimes\_percentages\_metro")

Data\_2022.groupby(['year','crime\_type']).size()

df\_Lockdown.groupby(['year','crime\_type']).size()

data\_Gw.dtypes

data\_Gw.count()

Data\_2021.count()

data\_Gw.isnull().sum()

data\_Gw.describe(include='object')

data\_Gw.hist(figsize=(10,10), xrot=45)

for column in data\_Gw.select\_dtypes(include='object'):

    if data\_Gw[column].nunique() < 20:

        sns.countplot(y=column, data=data\_Gw)

        plt.show()

for column in data\_Gw.select\_dtypes(include='object'):

    if data\_Gw[column].nunique() < 20:

        display(data\_Gw.groupby(column).mean())

data\_Gw.groupby(['year','crime\_type']).size()

 [data\_Gw.groupby(['year','crime\_type']).size()/(len(data\_Gw))\*100][:50]

sns.countplot(x='year',hue='crime\_type',data=data\_Gw)

sns.set(rc={'figure.figsize':(15,6)})

plt.title("crimes by the years")

plt.show()

pd.value\_counts(data\_Gw['year'])

data1 = data\_Gw.groupby(['Location','crime\_type']).size()[:30]

import matplotlib

matplotlib.rcParams['figure.figsize'] = (10,6)

data1.plot(kind = 'barh')

plt.title("crime\_type by location")

data\_Gw.groupby('Month').crime\_type.size()

data\_Gw.groupby('crime\_type').Month.value\_counts()[:120]

import matplotlib.pyplot as plt

# function to add value labels

def addlabels(x,y):

    for i in range(len(x)):

        plt.text(i, y[i], y[i], ha = 'center',

                 Bbox = dict(facecolor = 'pink', alpha =.8))

if \_\_name\_\_ == '\_\_main\_\_':

    # creating data on which bar chart will be plot

  x = ['January','February','March','April','May','June','July','August','September','October','November','December']

y = [14947,14116,15790,19265,17860,16348,16381,15591,15364,15339,10064,8947]

plt.figure(figsize = (12,7))

# making the bar chart on the data

plt.bar(x, y)

# calling the function to add value labels

addlabels(x, y)

plt.title("Crimes over month")

# giving X and Y labels

plt.xlabel("Months")

plt.ylabel("Crimes")

# visualizing the plot

plt.show()

data\_Gw.Month

data\_Gw.Latitude

data\_Gw.Longitude

sample\_df = data\_Gw.sample(int(0.1 \* len(data\_Gw)))

sns.scatterplot(x= sample\_df.Longitude, y= sample\_df.Latitude, size =0.001)

Latitude,Longitude = data\_Gw.Latitude[0], data\_Gw.Longitude[0]

Latitude,Longitude

crime\_type\_grp = Data\_2021.groupby(['crime\_type'])

df8=crime\_type\_grp.get\_group('Robbery')

df8

filt=df8['crime\_type'] == 'Robbery'

df8.loc[filt]['Month'].value\_counts()

# plot line graph

plt.title ("Robbery by Month in 2021")

x = ['January','February','March','April','May','June','July','August','September','October','November','December']

y = [21,21,16,29,24,24,18,21,23,31,14,23]

sns.lineplot(x , y)

plt.show()

crime\_type\_grp['Month'].value\_counts().loc['Anti-social behaviour']

# plot line graph

plt.title ("Anti-social behaviour by Month in 2021")

x = ['January','February','March','April','May','June','July','August','September','October','November','December']

y = [2273,2100,2260,1878,1362,1391,1348,1193,1004,944,1072,749]

sns.lineplot(x , y)

plt.show()

df10 = data\_Gw[data\_Gw['Month'].isin(['June','July','August'])]

df10

df21 = data\_Gw[data\_Gw['Month'].isin(['April','May','July'])]

df21

dfa =data\_Gw [data\_Gw['Month'].isin(['April'])]

dfa

crime\_type\_grp3 = dfa.groupby(['crime\_type'])

breakdown\_crime\_gwent = crime\_type\_grp3['Month'].value\_counts().loc['Anti-social behaviour'].sum()

breakdown\_crime\_gwent

df11=data\_Gw[data\_Gw['Month'].isin(['December','January','February'])]

df11

crime\_type\_grp1 = df10.groupby(['crime\_type'])

df22 = df21.groupby('crime\_type').Month.value\_counts()

df22

crime\_type\_grp2 = df11.groupby(['crime\_type'])

summer\_anti = crime\_type\_grp1['Month'].value\_counts().loc['Anti-social behaviour'].sum()

summer\_anti

winter\_anti = crime\_type\_grp2['Month'].value\_counts().loc['Anti-social behaviour'].sum()

winter\_anti

summer\_vio = crime\_type\_grp1['Month'].value\_counts().loc['Violence and sexual offences'].sum()

summer\_vio

winter\_vio = crime\_type\_grp2['Month'].value\_counts().loc['Violence and sexual offences'].sum()

winter\_vio

summer\_pub = crime\_type\_grp1['Month'].value\_counts().loc['Public order'].sum()

summer\_pub

winter\_pub = crime\_type\_grp2['Month'].value\_counts().loc['Public order'].sum()

winter\_pub

summer\_vehi = crime\_type\_grp1['Month'].value\_counts().loc['Vehicle crime'].sum()

summer\_vehi

winter\_vehi = crime\_type\_grp2['Month'].value\_counts().loc['Vehicle crime'].sum()

winter\_vehi

summer\_oth = crime\_type\_grp1['Month'].value\_counts().loc['Other theft'].sum()

summer\_oth

winter\_oth = crime\_type\_grp2['Month'].value\_counts().loc['Other theft'].sum()

winter\_oth

summer\_drugs = crime\_type\_grp1['Month'].value\_counts().loc['Drugs'].sum()

summer\_drugs

winter\_drugs = crime\_type\_grp2['Month'].value\_counts().loc['Drugs'].sum()

winter\_drugs

summer\_shopli = crime\_type\_grp1['Month'].value\_counts().loc['Shoplifting'].sum()

summer\_shopli

winter\_shopli = crime\_type\_grp2['Month'].value\_counts().loc['Shoplifting'].sum()

winter\_shopli

summer\_Robbery = crime\_type\_grp1['Month'].value\_counts().loc['Robbery'].sum()

summer\_Robbery

winter\_Robbery = crime\_type\_grp2['Month'].value\_counts().loc['Robbery'].sum()

winter\_Robbery

labels = ['Violence and sexual offences', 'Anti\_social behaviour', 'Public order', 'Other theft', 'Vehicle crime','Shoplifting','Drugs','Robbery']

summer\_total = [16837, 12150, 7602, 2757, 1775, 1730, 1173, 212]

winter\_total = [13396, 9749, 4841, 2224, 1655, 1615, 1069, 173]

x = np.arange(len(labels))  # the label locations

width = 0.25  # the width of the bars

fig, ax = plt.subplots()

rects1 = ax.bar(x - width/2, summer\_total, width, label='summer')

rects2 = ax.bar(x + width/2, winter\_total, width, label='winter')

# Add some text for labels, title and custom x-axis tick labels, etc.

ax.set\_ylabel('crimes')

ax.set\_title('crimes by summer and winter')

ax.set\_xticks(x)

ax.set\_xticklabels(labels)

ax.legend()

def autolabel(rects):

    """Attach a text label above each bar in \*rects\*, displaying its height."""

    for rect in rects:

        height = rect.get\_height()

        ax.annotate('{}'.format(height),

                    xy=(rect.get\_x() + rect.get\_width() / 3, height),

                    xytext=(0, 5),  # 3 points vertical offset

                    textcoords="offset points",

                    ha='center', va='bottom')

autolabel(rects1)

autolabel(rects2)

fig.tight\_layout()

plt.show()

list(zip(list(data\_Gw.Latitude), list(data\_Gw.Longitude)))

import folium

from folium.plugins import HeatMap

import numpy as np

sample\_data\_Gw = data\_Gw.sample(int(0.01\*len(data\_Gw)))

lat\_lon\_pairs = list(zip(list(sample\_data\_Gw.Latitude), list(sample\_data\_Gw.Longitude)))

map = folium.Map()

HeatMap (lat\_lon\_pairs).add\_to(map)

map

Data\_2022.groupby(['year','crime\_type']).size()

df\_Lockdown.groupby(['year','crime\_type']).size()

import plotly.graph\_objects as go

from plotly.subplots import make\_subplots

labels = ["Anti-social behaviour","Violence and sexual offences","Public order","Other theft","Vehicle crime","Burglary","Shoplifting","Drugs"]

fig = make\_subplots(1, 2, specs=[[{'type':'domain'}, {'type':'domain'}]],

                    subplot\_titles=['Crimes after lockdown', 'Crimes during lockdown'])

fig.add\_trace(go.Pie(labels=labels, values= [9450,20584,8707,3199,2005,1600,2070,1104]

, scalegroup='one',

                     name="Crimes after lockdown-2022"), 1, 1)

fig.add\_trace(go.Pie(labels=labels, values= [15753,16749,6699,2758,1876,1563,1790,1286]

, scalegroup='one',

                     name="Crimes during lockdown-2021"), 1, 2)

fig.update\_layout(title\_text='Crimes during and after lockdown')

fig.show()

Data\_2021.groupby(['year','crime\_type']).size()

import plotly.graph\_objects as go

from plotly.subplots import make\_subplots

labels = ["Anti-social behaviour","Violence and sexual offences","Public order","Other theft","Vehicle crime","Burglary","Shoplifting","Drugs"]

fig = make\_subplots(1, 2, specs=[[{'type':'domain'}, {'type':'domain'}]],

                    subplot\_titles=['crimes by metropoliton area in 2021','crimes by Gwent area in 2021'])

fig.add\_trace(go.Pie(labels=labels, values = [301494,244927,58714,89488,100188,53388,33047,42819,22452]

, scalegroup='one',

                     name="crimes by Metropoliton area in 2021"), 1, 1)

fig.add\_trace(go.Pie(labels=labels, values= [17574,20553,8112,3399,2274,1915,2179,1494,265]

, scalegroup='one',

                     name="crimes by Gwent area in 2021"), 1, 2)

fig.update\_layout(title\_text='Crimes by Metropoliton and Gwent area in 2021 ')

fig.show()

Data\_metro.groupby(['year','crime\_type']).size()

import matplotlib.pyplot as plt

# function to add value labels

def addlabels(x,y):

    for i in range(len(x)):

        plt.text(i, y[i], y[i], ha = 'center',

                 Bbox = dict(facecolor = 'red', alpha =.8))

if \_\_name\_\_ == '\_\_main\_\_':

    # creating data on which bar chart will be plot

       x =['Anti-social behaviour','Violence and sexual offences','Public order','Other theft','Vehicle crime','Burglary','Shoplifting','Drugs','Robbery','Theft from the person','Criminal damage and arson']

y = [301494,244927,58714,89488,100188,53388,33047,42819,2252,37591,52684]

plt.figure(figsize = (15,9))

# making the bar chart on the data

plt.bar(x, y)

# calling the function to add value labels

addlabels(x, y)

plt.title("Crimes in Metropoliton area")

# giving X and Y labels

plt.xlabel("crime\_type")

plt.ylabel("Count")

# visualizing the plot

plt.show()

df\_Metro = Data\_metro.groupby(['crime\_type','Month']).size()[:50]

df\_Metro

sns.countplot(x='Month',hue='crime\_type',data=Data\_metro)

sns.set(rc={'figure.figsize':(20,15)})

plt.title("crimes by Month")

plt.show()

data\_Gw\_ct = pd.crosstab(data\_Gw.crime\_type, data\_Gw.year, margins=True)

data\_Gw\_ct.head()

#include years from 2020 to 2022

df25 = data\_Gw[data\_Gw.year >=2020]

df26 = data\_Gw[data\_Gw.year <=2022]

#Transpoded crosstab & find correlation

data\_Gw\_ct =  pd.crosstab(df25.crime\_type, df26.year, margins=False).reindex()

data\_Gw\_ct = data\_Gw\_ct.T

type\_corr = data\_Gw\_ct.corr()

type\_corr

#Correlation heatmap

plt.figure(figsize=(10,7))

ax = sns.heatmap(type\_corr, annot=True) #notation: "annot" not "annote"

bottom, top = ax.get\_ylim()

ax.set\_ylim(bottom + 0.5, top - 0.5)

plt.figure(figsize=(16, 6))

heatmap = sns.heatmap(type\_corr, vmin=-1, vmax=1, annot=True, cmap='BrBG')

heatmap.set\_title('Correlation Heatmap', fontdict={'fontsize':18}, pad=12);

# save heatmap as .png file

# dpi - sets the resolution of the saved image in dots/inches

# bbox\_inches - when set to 'tight' - does not allow the labels to be cropped

plt.savefig('heatmap.png', dpi=300, bbox\_inches='tight')

top\_crimes = Data\_metro.groupby(['crime\_type'])['Month'].size()

top\_crimes = pd.DataFrame(top\_crimes).nlargest(10,'Month').reset\_index()

top\_crimes = list(top\_crimes['crime\_type'])

df22 = df21.groupby('crime\_type').Month.value\_counts()

df22

data\_Gw\_loc\_ct = pd.crosstab(data\_Gw.Location, data\_Gw.year, margins=True)

data\_Gw\_loc\_ct.sort\_values("All", axis = 0, ascending = False,

                 inplace = True)

data\_Gw\_loc\_ct.head(6)

data\_Gw\_loc\_ct = pd.crosstab(data\_Gw.Location, data\_Gw.year, margins=True)

data\_Gw\_loc\_ct.sort\_values("All", axis = 0, ascending = False,

                 inplace = True)

data\_Gw\_loc\_ct.tail(6)

#group data by Neighbourhood and count

data\_Gw\_gp = data\_Gw.groupby(['Month','Location']).count()[['crime\_type']]

data\_Gw\_gp= data\_Gw\_gp.reset\_index()

data\_Gw\_gp.head(7)

from sklearn import preprocessing

lab=preprocessing.LabelEncoder()

data\_Gw['crime\_type']=lab.fit\_transform(data\_Gw['crime\_type'])

from sklearn.ensemble import ExtraTreesClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.cluster import KMeans

kmeans = KMeans(n\_clusters=14)

kmeans.fit(data\_Gw.iloc[:,4:])

kmeans.cluster\_centers\_

labels = kmeans.labels\_

labels

import numpy as np

unique , counts = np.unique(kmeans.labels\_, return\_counts=True)

dict\_data = dict(zip(unique, counts))

dict\_data

data\_Gw["cluster"] = kmeans.labels\_