Lab on 5 May

KUMAR GAURAV 20122065

Loading Libraries

```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import OrdinalEncoder
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading dataset

```
df = pd.read_csv("/content/drive/MyDrive/Classroom/MSc 2020- Machine Learning MSc Semester
df.head()
```

	State	Year	Data.Population	Data.Rates.Property.All	Data.Rates.Property.Burg
0	Alabama	1960	3266740	1035.4	
1	Alabama	1961	3302000	985.5	
2	Alabama	1962	3358000	1067.0	
3	Alabama	1963	3347000	1150.9	
4	Alabama	1964	3407000	1358.7	

Glimpse of dataset

```
df.shape
```

```
(2751, 21)
```

```
df.columns
```

```
Index(['State', 'Year', 'Data.Population', 'Data.Rates.Property.All',
       'Data.Rates.Property.Burglary', 'Data.Rates.Property.Larceny',
       'Data.Rates.Property.Motor', 'Data.Rates.Violent.All',
       'Data.Rates.Violent.Assault', 'Data.Rates.Violent.Murder',
```

```
       'Data.Rates.Violent.Rape', 'Data.Rates.Violent.Robbery',
       'Data.Totals.Property.All', 'Data.Totals.Property.Burglary',
       'Data.Totals.Property.Larceny', 'Data.Totals.Property.Motor',
       'Data.Totals.Violent.All', 'Data.Totals.Violent.Assault',
       'Data.Totals.Violent.Murder', 'Data.Totals.Violent.Rape',
       'Data.Totals.Violent.Robbery'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2751 entries, 0 to 2750
Data columns (total 21 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   State                           2751 non-null   object 
 1   Year                            2751 non-null   int64  
 2   Data.Population                 2751 non-null   int64  
 3   Data.Rates.Property.All         2751 non-null   float64
 4   Data.Rates.Property.Burglary    2751 non-null   float64
 5   Data.Rates.Property.Larceny     2751 non-null   float64
 6   Data.Rates.Property.Motor       2751 non-null   float64
 7   Data.Rates.Violent.All          2751 non-null   float64
 8   Data.Rates.Violent.Assault      2751 non-null   float64
 9   Data.Rates.Violent.Murder       2751 non-null   float64
 10  Data.Rates.Violent.Rape         2751 non-null   float64
 11  Data.Rates.Violent.Robbery      2751 non-null   float64
 12  Data.Totals.Property.All        2751 non-null   int64  
 13  Data.Totals.Property.Burglary   2751 non-null   int64  
 14  Data.Totals.Property.Larceny    2751 non-null   int64  
 15  Data.Totals.Property.Motor      2751 non-null   int64  
 16  Data.Totals.Violent.All         2751 non-null   int64  
 17  Data.Totals.Violent.Assault     2751 non-null   int64  
 18  Data.Totals.Violent.Murder      2751 non-null   int64  
 19  Data.Totals.Violent.Rape        2751 non-null   int64  
 20  Data.Totals.Violent.Robbery     2751 non-null   int64  
dtypes: float64(9), int64(11), object(1)
memory usage: 451.5+ KB
```

```
df.describe()
```

	Year	Data.Population	Data.Rates.Property.All	Data.Rates.Property.Bu
count	2751.000000	2.751000e+03	2751.000000	2751.(
mean	1986.043621	9.349570e+06	3686.538750	931.2
std	15.279324	3.368126e+07	1427.899631	442.4
min	1960.000000	2.261670e+05	573.100000	182.(
25%	1973.000000	1.208000e+06	2613.400000	592.1
50%	1986.000000	3.282000e+06	3670.000000	870.(
75%	1999.000000	5.840286e+06	4579.150000	1183.8
max	2012.000000	3.139140e+08	9512.100000	2906.7

```
df.State.value_counts()
```

```
Oklahoma               53
Wyoming                53
Nevada                 53
North Dakota           53
Louisiana              53
New Mexico             53
New Jersey             53
Tennessee              53
Idaho                  53
Alaska                 53
Nebraska               53
Ohio                   53
Illinois               53
New Hampshire          53
California             53
Florida                53
Mississippi            53
Colorado               53
Oregon                 53
Connecticut            53
Utah                   53
Missouri               53
West Virginia          53
Indiana                53
South Carolina         53
Hawaii                 53
South Dakota           53
Wisconsin              53
Maine                  53
Minnesota              53
North Carolina         53
Vermont                53
Arizona                53
Pennsylvania           53
Massachusetts          53
Montana                53
District of Columbia   53
Rhode Island           53
United States          53
Kansas                 53
Kentucky               53
Georgia                53
Texas                  53
Washington             53
Delaware               53
Iowa                   53
Maryland               53
Alabama                53
Virginia               53
Arkansas               53
Michigan               53
New York               48
Name: State, dtype: int64
```

```
# separating property related features for visualization
```

```
df_property = df[['State','Year','Data.Rates.Property.All','Data.Rates.Property.Burglary',
```

```
'Data.Totals.Property.Burglary','Data.Totals.Property.Larceny', 'Data.Totals.Prop
```

```
# separating violent related features for visualization

df_violent = df.drop(df_property,axis=1)
df_violent[['State','Year']] = df[['State','Year']]
df_violent.head()
```

	Data.Population	Data.Rates.Violent.All	Data.Rates.Violent.Assault	Data.Rates.V
0	3266740	186.6	138.1	
1	3302000	168.5	128.9	
2	3358000	157.3	119.0	
3	3347000	182.7	142.1	
4	3407000	213.1	163.0	

```
# Encoding `State` variable

ord_enc = OrdinalEncoder()
df["State_enc"] = ord_enc.fit_transform(df[["State"]])

df.drop("State",axis=1,inplace=True)
df.sample(10)
```

	Year	Data.Population	Data.Rates.Property.All	Data.Rates.Property.Burglary
328	1970	3032217	3319.0	1084.2
1277	1965	2321000	807.1	285.5
540	1970	4589575	2577.1	899.9
2174	1966	682000	1430.6	318.9
2110	2008	1050788	2845.0	548.7
614	1991	1135000	5728.6	1234.4
1437	1966	1456000	1861.0	420.1
577	2007	9544750	3889.6	946.0
176	1977	2144000	3018.1	972.6
194	1995	2484000	4137.7	996.9

Visualizations

1. Property related variables

```
plt.figure(figsize=(20,15))
sns.set_theme(style="whitegrid")
sns.barplot("Data.Rates.Property.All","State",data=df_property,orient='h')
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-3-bb143bec5b35> in <module>()
----> 1 plt.figure(figsize=(20,15))
      2 sns.set_theme(style="whitegrid")
      3 sns.barplot("Data.Rates.Property.All","State",data=df_property,orient='h')
      4 plt.show()

NameError: name 'plt' is not defined
```

SEARCH STACK OVERFLOW

Most rate of crimes related to property took place in the District of Columbia followed by Arizona

Least can be seen in West Virginia followed by North and South Dakota, as seen from the above graph

```
print(df.Year.unique())
print(df_property.Year.nunique())
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-4-bf6b5c0988c7> in <module>()
----> 1 print(df.Year.unique())
      2 print(df_property.Year.nunique())

NameError: name 'df' is not defined
```

SEARCH STACK OVERFLOW

```
plt.figure(figsize=(20,15))
sns.set_theme(style="whitegrid")
sns.barplot("Data.Rates.Property.All","Year",data=df_property,orient='h')
plt.show()
```

From the above graph, it is seen that from 1960's there is an increasing trend in the rate of property crimes which gradually increased until 1980's and then it started to decline. In the year 2012, rate is less than 3000.

```
plt.figure(figsize=(20,15))
```

```
sns.heatmap(df.corr(),annot=True)
```

```
plt.figure(figsize=(20,15))
sns.set_theme(style="whitegrid")
sns.barplot("Data.Rates.Property.Burglary","State",data=df_property,orient='h')
plt.show()
```

In Columbia and District of Columbia, the rates of property Burglary is seen the highest

North Dakota followed by South Dakota are states having minimum rates of Burglary

- List item
- List item

```
plt.figure(figsize=(20,15))
sns.set_theme(style="whitegrid")
sns.barplot("Data.Rates.Property.Larceny","State",data=df_property,orient='h')
plt.show()
```

Maximum rate of property Larcency is seen in the case of District of Columbia followed by Arizona

Minimum rate or property Larcency is seen in West Virginia

```
plt.figure(figsize=(20,15))
sns.set_theme(style="whitegrid")
sns.barplot("Data.Rates.Property.Motor","State",data=df_property,orient='h')
plt.show()
```

District of Columbia is seen on the top having maximum rate of motor crimes

least is for South Dakota

```
df_property.columns
```

```
plt.figure(figsize=(20,15))
sns.set_theme(style="whitegrid")
sns.barplot("Data.Totals.Property.All","State",data=df_property,orient='h')
plt.show()
```

```
# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(df)
    # inertia method returns wcss for that model
    wcss.append(kmeans.inertia_)


plt.figure(figsize=(10,5))
sns.lineplot(range(1, 11), wcss,marker='o',color='blue')
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()



# Fitting K-Means to the dataset
kmeans = KMeans(n_clusters = 2, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(df)
y_kmeans


y_k=pd.Series(y_kmeans)


kmeans.cluster_centers_



out=pd.DataFrame(y_kmeans)
pd.concat([df, out],axis=1)


df = pd.concat([df,y_kmeans],axis=1)


plt.figure(figsize = (20,10))
df = np.array(df)
plt.scatter(df[y_kmeans == 0,0],df[y_kmeans == 0,1],s = 50, c = 'green', label = "High inc
plt.scatter(df[y_kmeans == 1,0],df[y_kmeans == 1,1],s = 50, c = 'blue', label = "medium in
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1], s = 100, c = "yello
plt.xlabel("Anual income(k$) -- >")
plt.ylabel("spending score out of 100 -- >")
plt.legend()
plt.show()
```

1s completed at 5:17 PM