Experiment 5: Clustering

Aim:
To use clustering on a dataset and test the accuracy of the model.

Problem Statement:
Choose a classification dataset of your choice from any of the following Repository Links, download it:

1. Kaggle: https://www.kaggle.com/
2. UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php

Perform Linear Regression on the chosen dataset.

Your notebook should contain:

1. Basic EDA

[***Hint***: Follow the steps in Titanic notebook uploaded on moodle under Expt 3 reference material]

Tool/Language:
Programming language: Python
Libraries: numpy, pandas, sklearn, matplotlib, seaborn

Code with visualisation graphs:

1) **Dataset Chosen:** Worldwide Food/Feed Production & Distribution.
2) **Dataset Description:** This dataset was meticulously gathered, organized and published by the Food and Agriculture Organization of the United Nations.
3) **Code:**

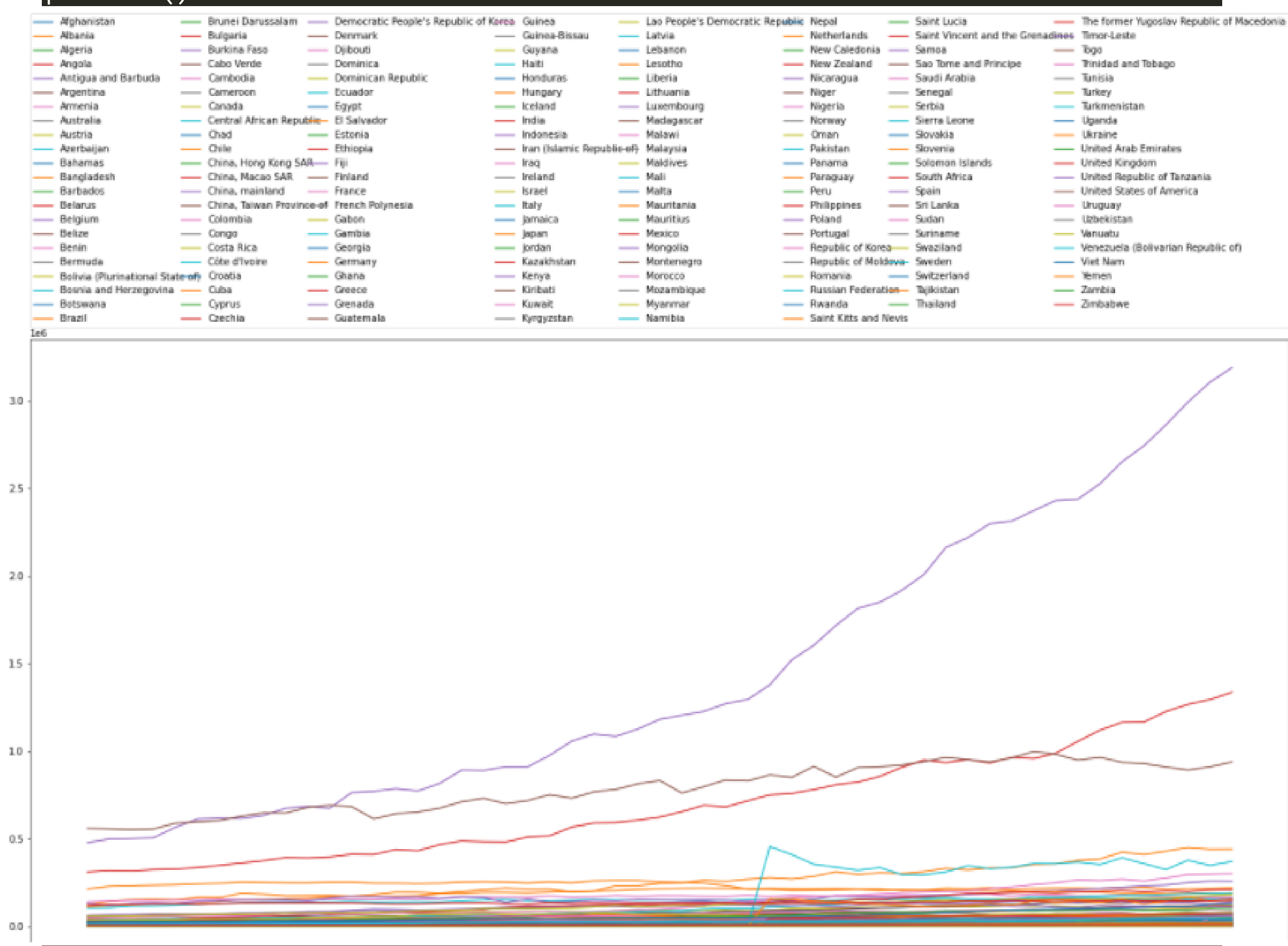
```
from google.colab import files
uploaded = files.upload()

import io
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Reading the dataset
df = pd.read_csv(io.BytesIO(uploaded['FAO.csv']), encoding = "ISO-8859-1")
df.head()
```

	Area Abbreviation	Area Code	Area	Item Code	Item	Element Code	Element	Unit	latitude	longitude	Y1961	Y1962	Y1963	Y1964	Y1965	Y1966	Y1967	Y1968	Y1969	Y1970	Y1971
0	AFG	2	Afghanistan	2511	Wheat and products	5142	Food	1000 tonnes	33.94	67.71	1928.0	1904.0	1666.0	1950.0	2001.0	1808.0	2053.0	2045.0	2154.0	1819.0	1963.0
1	AFG	2	Afghanistan	2805	Rice (Milled Equivalent)	5142	Food	1000 tonnes	33.94	67.71	183.0	183.0	182.0	220.0	220.0	195.0	231.0	235.0	238.0	213.0	205.0
2	AFG	2	Afghanistan	2513	Barley and products	5521	Feed	1000 tonnes	33.94	67.71	76.0	76.0	76.0	76.0	76.0	75.0	71.0	72.0	73.0	74.0	71.0
3	AFG	2	Afghanistan	2513	Barley and products	5142	Food	1000 tonnes	33.94	67.71	237.0	237.0	237.0	238.0	238.0	237.0	225.0	227.0	230.0	234.0	223.0
4	AFG	2	Afghanistan	2514	Maize and products	5521	Feed	1000 tonnes	33.94	67.71	210.0	210.0	214.0	216.0	216.0	216.0	235.0	232.0	236.0	200.0	201.0

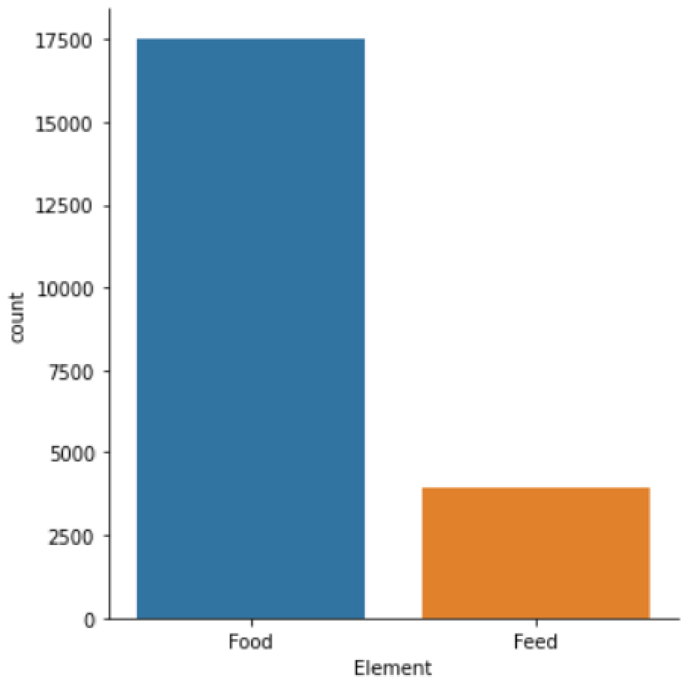
```
df.shape
```

```
(21477, 63)
```

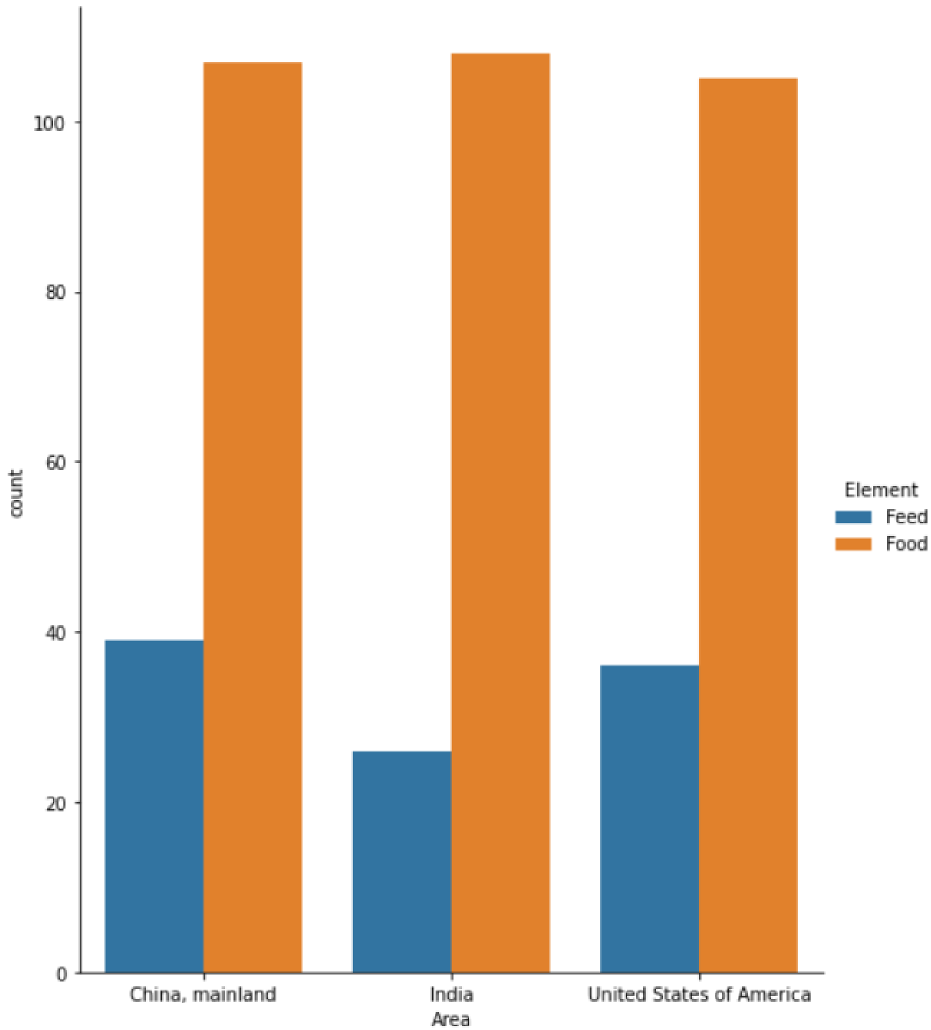
```
area_list = list(df['Area'].unique())
year_list = list(df.iloc[:,10:].columns)

plt.figure(figsize=(24,12))
for ar in area_list:
    yearly_produce = []
    for yr in year_list:
        yearly_produce.append(df[yr][df['Area'] == ar].sum())
    plt.plot(yearly_produce, label=ar)
plt.xticks(np.arange(53), tuple(year_list), rotation=60)
plt.legend(bbox_to_anchor=(0., 1.02, 1., .102), loc=3, ncol=8, mode="expand", bor
deraxespad=0.)
plt.show()
```



```
sns.catplot(x="Element", data=df, kind="count")
plt.show()
```




```
sns.catplot(x="Area", data=df[(df['Area'] == "India") | (df['Area'] == "China, ma
inland") | (df['Area'] == "United States of America")], kind="count", hue="Elemen
t", height=8, aspect=.8)
```




```python
new_df_dict = {}
for ar in area_list:
    yearly_produce = []
    for yr in year_list:
        yearly_produce.append(df[yr][df['Area']==ar].sum())
    new_df_dict[ar] = yearly_produce
new_df = pd.DataFrame(new_df_dict)
new_df.head()
```

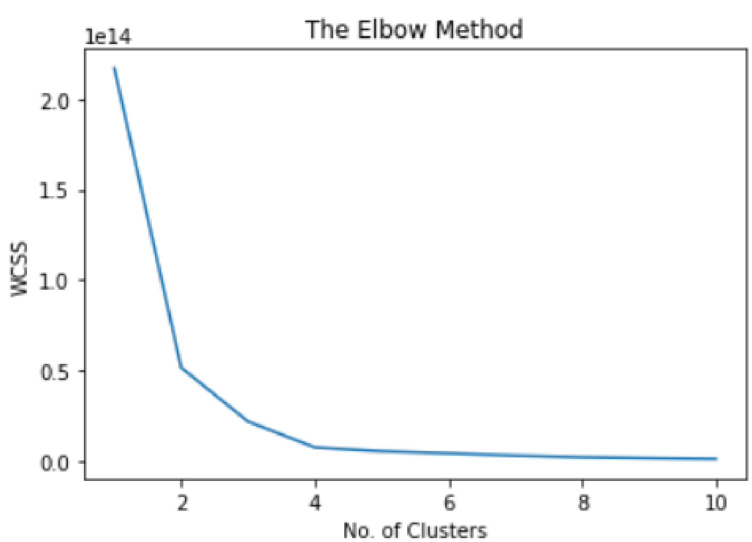
	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	Azerbaijan	Bahamas	Bangladesh	Barbados	Belarus	Belgium	Belize	Benin	Bermuda
0	9481.0	1706.0	7488.0	4834.0	92.0	43402.0	0.0	25795.0	22542.0	0.0	138.0	29451.0	244.0	0.0	0.0	91.0	2270.0	67.0
1	9414.0	1749.0	7235.0	4775.0	94.0	40784.0	0.0	27618.0	22627.0	0.0	142.0	29975.0	252.0	0.0	0.0	106.0	2247.0	68.0
2	9194.0	1767.0	6861.0	5240.0	105.0	40219.0	0.0	28902.0	23637.0	0.0	152.0	31446.0	264.0	0.0	0.0	103.0	2209.0	70.0
3	10170.0	1889.0	7255.0	5286.0	95.0	41638.0	0.0	29107.0	24099.0	0.0	167.0	32434.0	254.0	0.0	0.0	104.0	2287.0	72.0
4	10473.0	1884.0	7509.0	5527.0	84.0	44936.0	0.0	28961.0	22664.0	0.0	173.0	33108.0	253.0	0.0	0.0	104.0	2484.0	73.0

5 rows × 174 columns

```
new_df = pd.DataFrame.transpose(new_df)
new_df.columns = year_list

new_df.head()
```

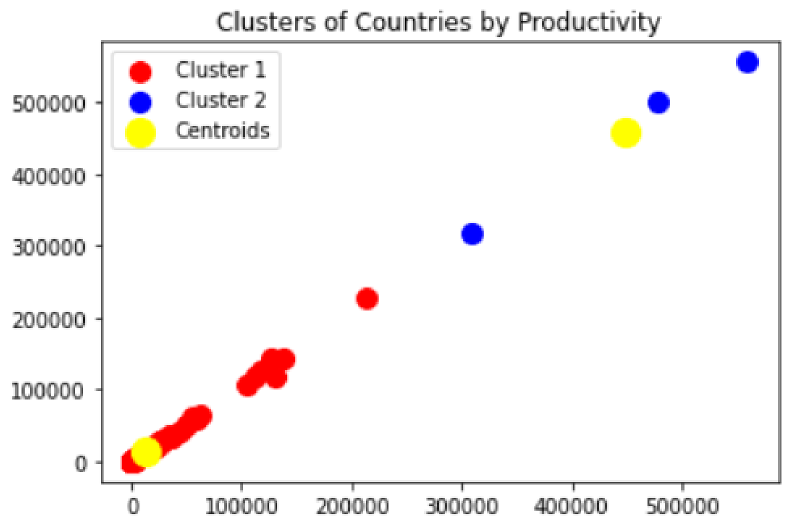
	Y1961	Y1962	Y1963	Y1964	Y1965	Y1966	Y1967	Y1968	Y1969	Y1970	Y1971	Y1972	Y1973	Y1974	Y1975	Y1976	Y1977	Y1978	Y1979	Y1980
Afghanistan	9481.0	9414.0	9194.0	10170.0	10473.0	10169.0	11289.0	11508.0	11815.0	10454.0	10433.0	11121.0	11759.0	12017.0	12348.0	13090.0	11274.0	12218.0	12150.0	11810.0
Albania	1706.0	1749.0	1767.0	1889.0	1884.0	1995.0	2046.0	2169.0	2230.0	2395.0	2376.0	2478.0	2575.0	2728.0	2822.0	3097.0	3258.0	3377.0	3352.0	3324.0
Algeria	7488.0	7235.0	6861.0	7255.0	7509.0	7536.0	7986.0	8839.0	9003.0	9355.0	9891.0	10711.0	11085.0	12418.0	14042.0	14248.0	15162.0	16214.0	17745.0	19205.0
Angola	4834.0	4775.0	5240.0	5286.0	5527.0	5677.0	5833.0	5685.0	6219.0	6460.0	6603.0	6499.0	6639.0	6526.0	6211.0	6413.0	6645.0	6923.0	6844.0	6906.0
Antigua and Barbuda	92.0	94.0	105.0	95.0	84.0	73.0	64.0	59.0	68.0	77.0	85.0	57.0	58.0	56.0	59.0	55.0	53.0	57.0	61.0	76.0

```
from sklearn.cluster import KMeans
wcss = []
for i in range(1,11):
    kmeans = KMeans(n_clusters=i,init='k-
means++',max_iter=300,n_init=10,random_state=0)
    kmeans.fit(new_df)
    wcss.append(kmeans.inertia_)
plt.plot(range(1,11),wcss)
plt.title('The Elbow Method')
plt.xlabel('No. of Clusters')
plt.ylabel('WCSS')
plt.show()
```




```
kmeans = KMeans(n_clusters=2,init='k-
means++',max_iter=300,n_init=10,random_state=0)
```

```
y_kmeans = kmeans.fit_predict(new_df)

X = new_df.values

plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0,1],s=100,c='red',label='Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1,1],s=100,c='blue',label='Cluster 2')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],s=200,c='yellow',label='Centroids')
plt.title('Clusters of Countries by Productivity')
plt.legend()
plt.show()
```



Conclusion:

The model works pretty good with only 2 clusters. This is because the data is primarily divided into upper half and lower half. The upper half is dominated by three countries – USA, India and China whereas the lower half is filled with the remaining countries.