Prediction result -report

- ## Clustering algorithm:

Firstly, I import Clustering algorithm from sklearn package.

then A located population column and Victimisations column in new variable using (iloc()) method.

```
X1=data.iloc[:,[3,2]].Values
```

Then import kmeans from sklearn.cluster.

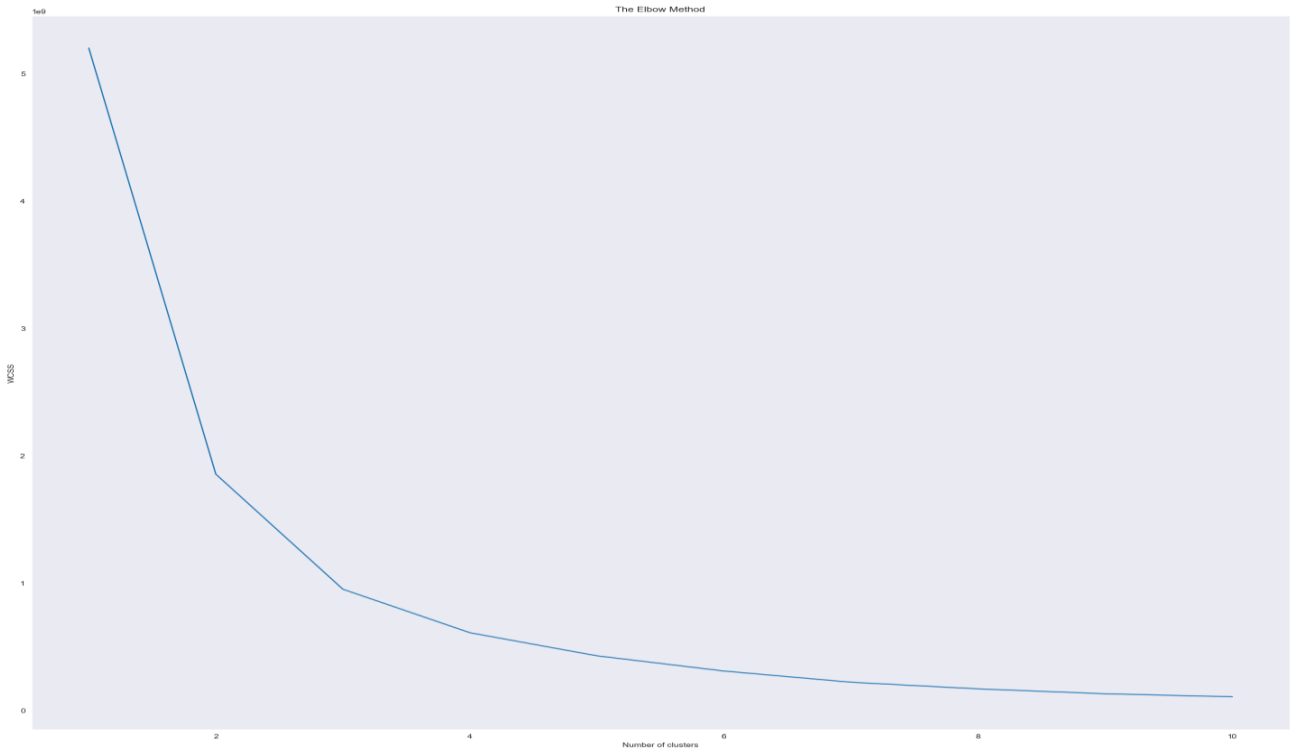
Then calculate the pest number of clusters by using (kmeans() and fit()) methods. (Elbow Method)

```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 100)
    kmeans.fit(X1)
    wcss.append(kmeans.inertia_)
```

Then `visualize the` result by using (plot()) method.

```
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

Prediction result -report



From plot I create four clusters.

Then appley kmeans algorithm by using (KMeans() and fit_predict()) methods.

```
kmeans = KMeans(n_clusters = 4, init = 'k-means++', random_state = 100)
y_kmeans = kmeans.fit_predict(X1)
```

Then `visualize the` result by using scatter plot.

```
plt.scatter(X1[y_kmeans == 0, 0], X1[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X1[y_kmeans == 1, 0], X1[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X1[y_kmeans == 2, 0], X1[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X1[y_kmeans == 3, 0], X1[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 400, c = 'yellow',
label = 'Centroids')
plt.title('Clusters of Victimisations')
plt.xlabel('Population_mid_point_2015')
plt.ylabel('Victimisations_calendar_year_2015')
plt.legend()
plt.show()
```

Prediction result -report