K-Means Clustering

Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

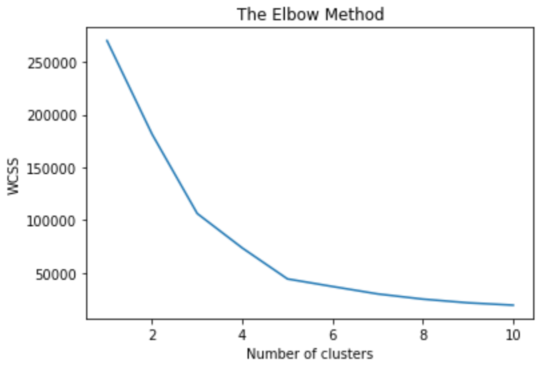
Importing the dataset

```
dataset = pd.read_csv('Mall_Customers.csv')
X = dataset.iloc[:, [3, 4]].values
X
```

```
           [ 60,  42],
           [ 60,  52],
           [ 60,  47],
           [ 60,  50],
           [ 61,  42],
           [ 61,  49],
           [ 62,  41],
           [ 62,  48],
           [ 62,  59],
           [ 62,  55],
           [ 62,  56],
           [ 62,  42],
           [ 63,  50],
           [ 63,  46],
           [ 63,  43],
           [ 63,  48],
           [ 63,  52],
           [ 63,  54],
           [ 64,  42],
           [ 64,  46],
           [ 65,  48],
           [ 65,  50],
           [ 65,  43],
           [ 65,  59],
           [ 67,  43],
           [ 67,  57],
           [ 67,  56],
           [ 67,  40],
           [ 69,  58],
           [ 69,  91],
           [ 70,  29],
           [ 70,  77],
           [ 71,  35],
           [ 71,  95],
           [ 71,  11],
           [ 71,  75],
           [ 71,   9],
           [ 71,  75],
           [ 72,  34],
           [ 72,  71],
           [ 73,   5],
           [ 73,  88],
           [ 73,   7],
           [ 73,  73],
           [ 74,  10],
           [ 74,  72],
           [ 75,   5],
           [ 75,  93],
           [ 76,  40],
           [ 76,  87],
           [ 77,  12],
           [ 77,  97],
           [ 77,  36],
           [ 77,  74],
           [ 78,  22],
           [ 78,  90],
           [ 78,  17],
           [ 78,  88],
           [ 78,  20],
```

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

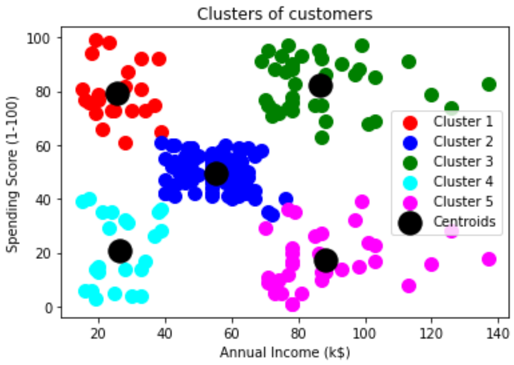


Training the K-Means model on the dataset

```
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(X)
```

Visualising the clusters

```
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(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'black', label = 'Centroids')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```


Clusters of customers
Spending Score (1-100)
Annual Income (k$)
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Centroids