Machine Learning
ETE-Practical

Course Name: Machine Learning

Course Code : BCSE3093

Program : B.Tech

Year/Semester : Third / 5th

Given BY-
Pratyush Deka Sir,

Submitted By-
Rajat Ranjan
18SCSE1010008

Problem-Write a program to demonstrate the working of the K-means Clustering. Use an appropriate data set the implementation.

Theory

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.
Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes.
A cluster refers to a collection of data points aggregated together because of certain similarities.

Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares.

In other words, the K-means algorithm identifies *k* number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. The *'means'* in the K-means refers to averaging of the data; that is, finding the centroid.

How the K-means algorithm works

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids.

It halts creating and optimizing clusters when either:

The centroids have stabilized — there is no change in their values because the clustering has been successful.

The defined number of iterations has been achieved.

CODE

In [1]:

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [4]:

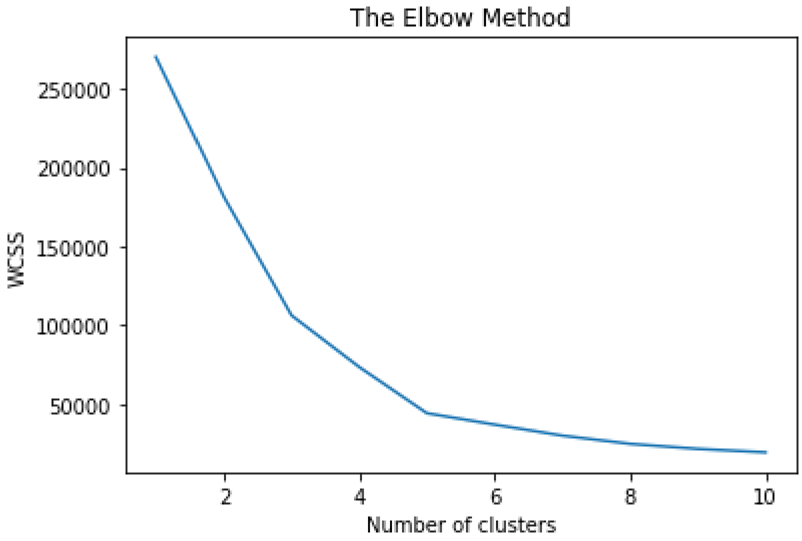
```
#Importing the mall dataset with pandas

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

In [5]:

```
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,
      kmeans.fit(X)
      wcss.append(kmeans.inertia_)
```

In [6]:

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



In [9]:

```
kmeans=KMeans(n_clusters= 5, init = 'k-means++', max_iter = 300, n_init = 10, rando
```

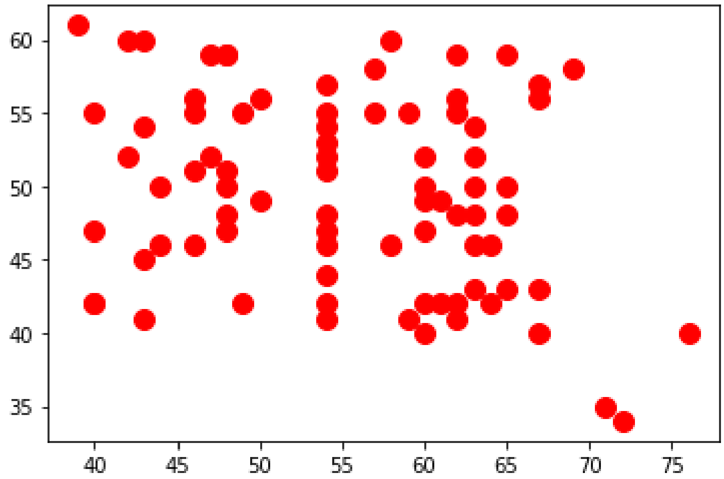
```
y_kmeans = kmeans.fit_predict(X)
```

In [13]:

```
# Visualising the clusters

plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0,1],s = 100, c='red', label ='Clus
```

Out[15]:

```
<matplotlib.collections.PathCollection at 0x7efe308afdf0>
```

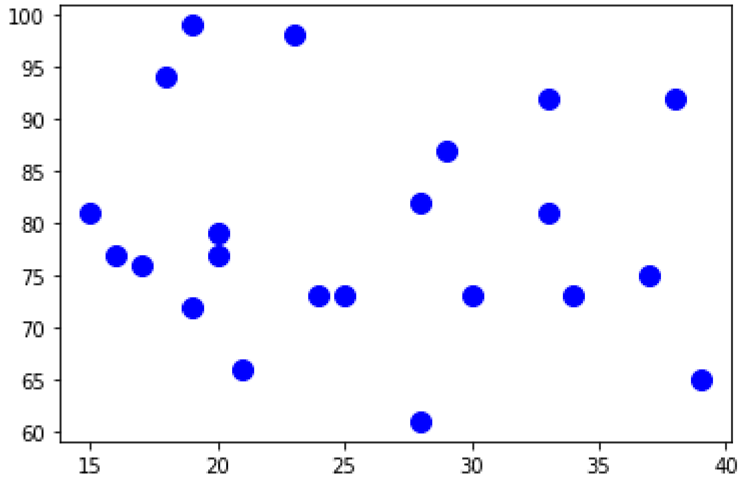


In [16]:

```
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1,1],s = 100, c='blue', label ='Clu
```

Out[16]:

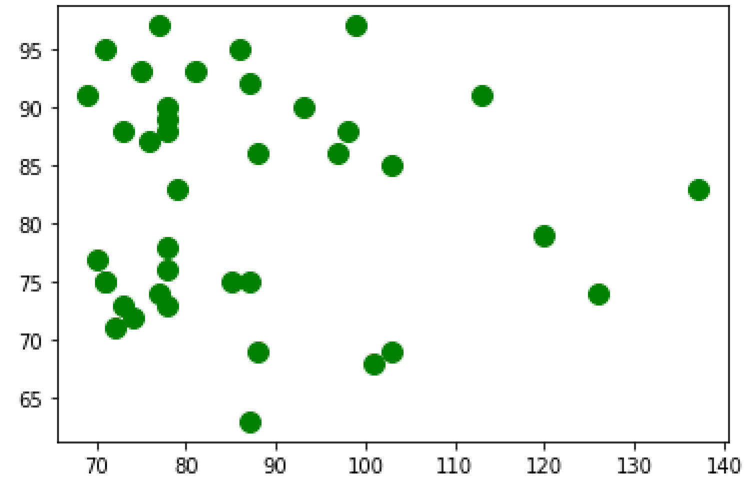
```
<matplotlib.collections.PathCollection at 0x7efe3081d9d0>
```



```
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2,1],s = 100, c='green', label ='Cl
```

Out[17]:

```
<matplotlib.collections.PathCollection at 0x7efe307f7f40>
```

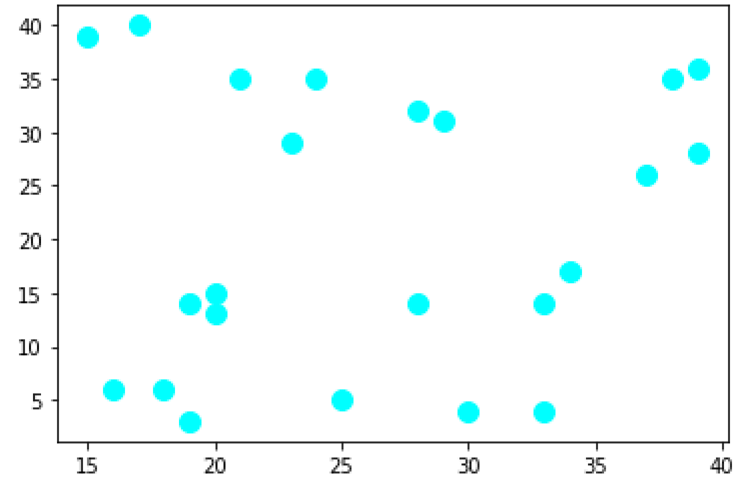


In [18]:

```
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3,1],s = 100, c='cyan', label ='Clu
```

Out[18]:

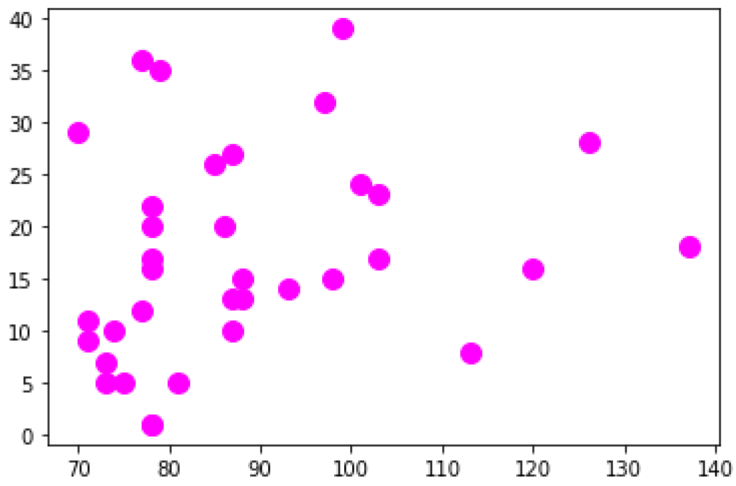
```
<matplotlib.collections.PathCollection at 0x7efe307576a0>
```



```
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4,1],s = 100, c='magenta', label ='
```

Out[19]:

```
<matplotlib.collections.PathCollection at 0x7efe307337f0>
```

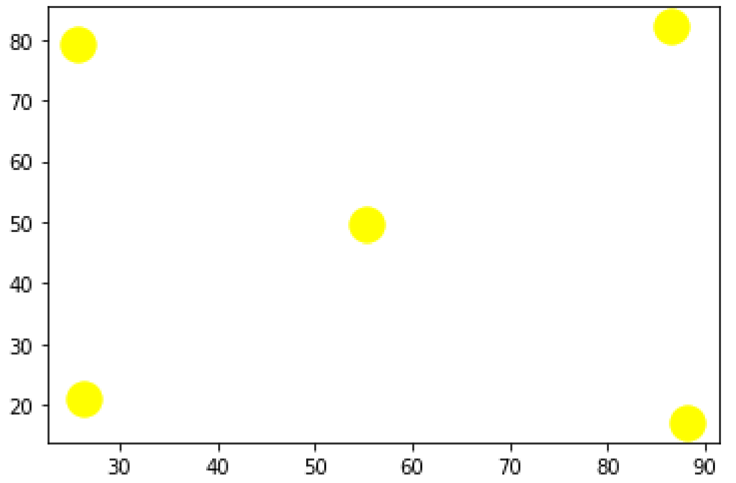


In [20]:

```
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], s = 300, c
```

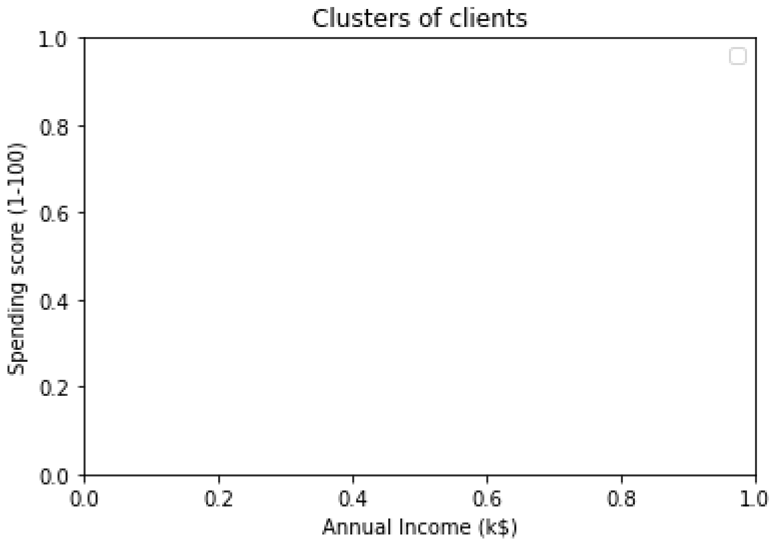
Out[20]:

```
<matplotlib.collections.PathCollection at 0x7efe306929d0>
```



```
plt.title('Clusters of clients')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending score (1-100)')
plt.legend()
plt.show()
```

```
No handles with labels found to put in legend.
```



In [ ]:

In [ ]: