

Ex.No: 14

Date: \_\_\_\_\_

# Implementation of clustering

Technique: K-Means.

Aim:

To implement a K-Means clustering techniques using Python language.

Explanation:

- \* Import KMeans from sklearn.cluster
- \* Assign x and y
- \* Call the function KMeans().
- \* Perform scatter operation and display the output.

Source code:

```
import numpy as np
```

```
import pandas as pd
```

```
from matplotlib import pyplot as plt
```

```
from sklearn.datasets.samples_generator import make_blobs
```

```
from sklearn.cluster import KMeans
```

```
X, y = make_blobs (n_samples=300, centers=4, cluster_std=0.60,  
random_state=0)
```

```
plt.scatter(X[:,0], X[:,1])
```

```
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(X)
```

```
    wcss.append (kmeans.inertia_)
```

```
plt.plot (range(1,11), wcss)
```

```
plt.title ('Elbow Method')
```



```
plt.xlabel('Number of clusters')
```

```
plt.ylabel('Wcss')
```

```
plt.show()
```

```
kmeans = KMeans (n_clusters = 4, init = 'k-means++',
```

```
max_iter = 300, n_init = 10, random_state = 0)
```

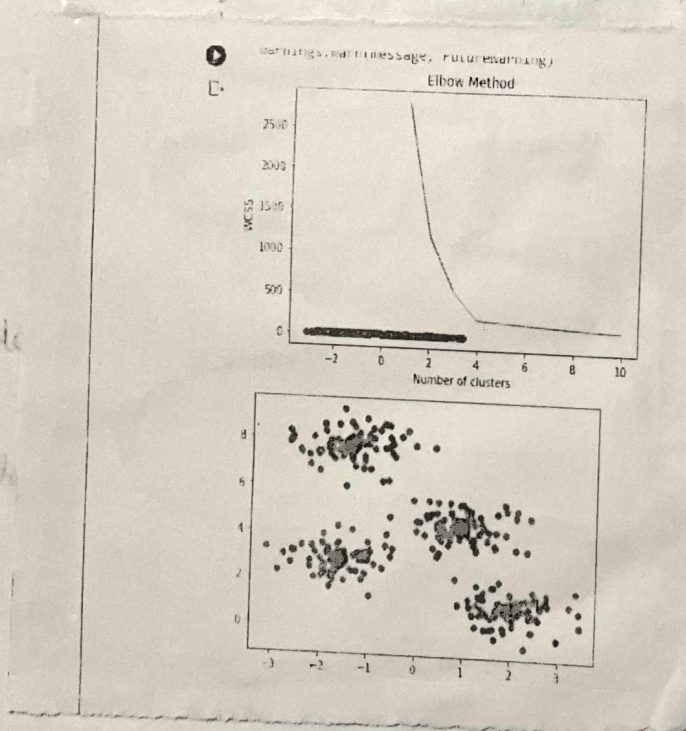
```
pred_y = kmeans.fit_predict(x)
```

```
plt.scatter(x[:, 0], x[:, 1])
```

```
plt.scatter(kmeans.cluster_centers[:, 0], kmeans.cluster_centers[:, 1], s = 300, c = 'red')
```

```
plt.show()
```

Output:



Result:

Thus the result was successfully executed and the output was executed.