

## Exp-14 Implementation of Clustering

Techniques: K-means

### Aim:

To implement a K-means clustering technique using Python language.

### Explanation

- \* Import K-means from sklearn.cluster
- \* Assign X and y.
- \* Call the function k\_means()
- \* Perform scatter operation and display the

### Code:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.datasets import make_blobs

X, y = make_blobs(n_samples=200, centers=4,
                  cluster_std=0.80, random_state=0)

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

for i in range(1, 11):

k\_means = KMeans(n\_clusters=i, init='k-means++', max\_iter=300, n\_init=10, random\_state=0)

k\_means.fit(X)

WCSS.append(k\_means.inertia\_)

plt.plot(range(1, 11), WCSS)

plt.title('Elbow Method')

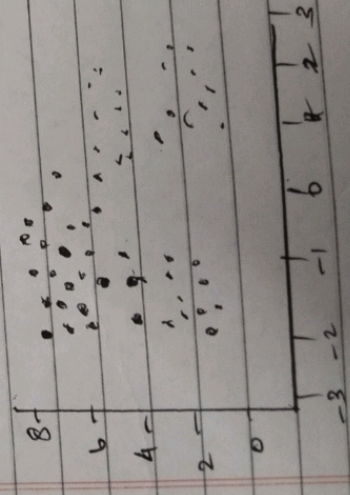
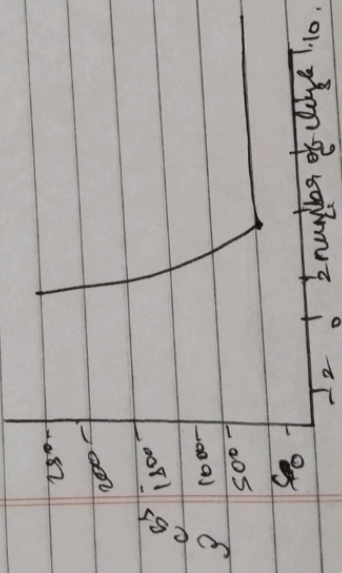


plt.xlabel('Number of clusters')  
plt.ylabel('wcss')  
plt.show()

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()

o/p

Elbow method



Result:  
Thus the result was successfully  
executed and the output was displayed