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

```python
df=pd.read_csv('iris.csv')
```

```python
df.head()
```

```
   sepallen  sepalwid  petallen  petalwid  label
0       5.1       3.5       1.4       0.2      0
1       4.9       3.0       1.4       0.2      0
2       4.7       3.2       1.3       0.2      0
3       4.6       3.1       1.5       0.2      0
4       5.0       3.6       1.4       0.2      0
```

```python
df.drop(['label'],axis=1,inplace=True)
```

```python
df.head()
```

```
   sepallen  sepalwid  petallen  petalwid
0       5.1       3.5       1.4       0.2
1       4.9       3.0       1.4       0.2
2       4.7       3.2       1.3       0.2
3       4.6       3.1       1.5       0.2
4       5.0       3.6       1.4       0.2
```

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

```python
km=KMeans(n_clusters=3)
```

```python
km.fit(df)
ym=km.predict(df)
```

```python
km.labels_
```

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 0, 0, 0, 2, 0, 0,
0,
       0, 0, 0, 2, 2, 0, 0, 0, 0, 2, 0, 2, 0, 2, 0, 0, 2, 2, 0, 0, 0,
0,
       0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 2])
```

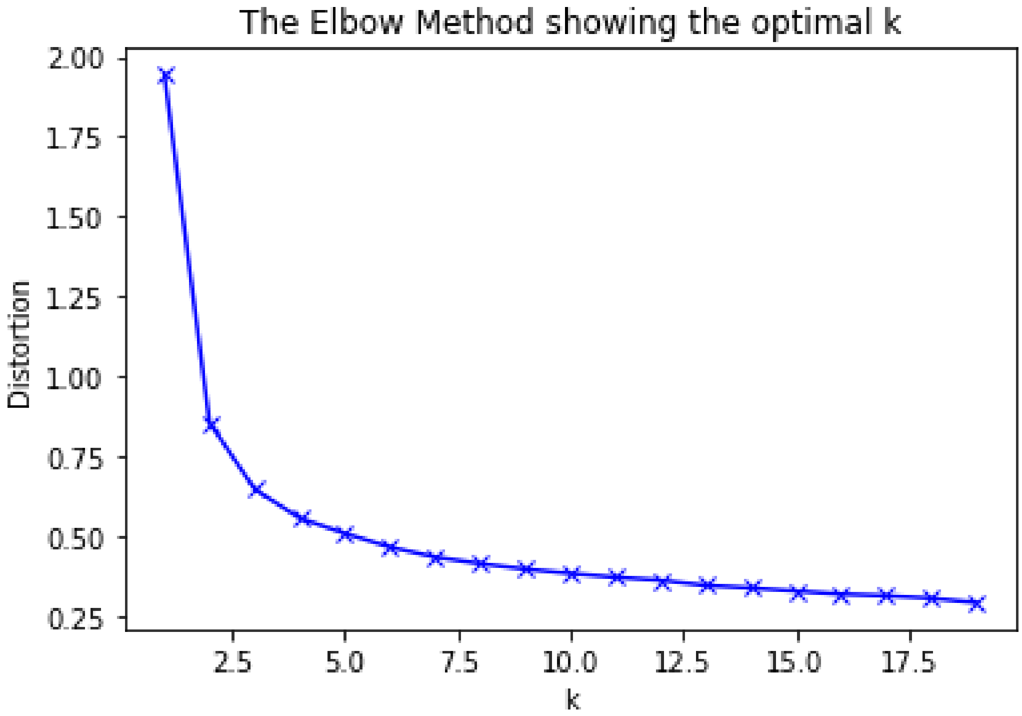
```python
ym
```

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 0, 0, 0, 2, 0, 0,
0,
       0, 0, 0, 2, 2, 0, 0, 0, 0, 2, 0, 2, 0, 2, 0, 0, 2, 2, 0, 0, 0,
0,
       0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 2])
```

```
from scipy.spatial.distance import cdist
distortions = []
K = range(1,20)
for k in K:
    km = KMeans(n_clusters=k).fit(df)
    km.fit(df)
    distortions.append(sum(np.min(cdist(df, km.cluster_centers_,
'euclidean'), axis=1)) / df.shape[0])

# Plot the elbow
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\cluster\
_kmeans.py:1036: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads.
You can avoid it by setting the environment variable
OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\user\anaconda3\lib\site-packages\sklearn\cluster\
_kmeans.py:1036: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads.
You can avoid it by setting the environment variable
OMP_NUM_THREADS=1.
  warnings.warn(
```



```
km=KMeans(n_clusters=3)
km.fit(df)
ym=km.predict(df)
center=km.cluster_centers_
km.labels_
```

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 0, 0, 0, 2, 0, 0,
0,
       0, 0, 0, 2, 2, 0, 0, 0, 0, 2, 0, 2, 0, 2, 0, 0, 2, 2, 0, 0, 0,
0,
       0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 2])
```

```
df.columns
```

```
Index(['sepallen', 'sepalwid', 'petallen', 'petalwid'],
dtype='object')
```

```
tdf=pd.DataFrame(data=center,columns=df.columns)
```

```
tdf
```

```
   sepallen  sepalwid  petallen  petalwid
0  6.850000  3.073684  5.742105  2.071053
1  5.006000  3.428000  1.462000  0.246000
2  5.901613  2.748387  4.393548  1.433871
```

```
com_df=pd.concat([df,tdf])
```

```
from sklearn.preprocessing import StandardScaler
df_norm=StandardScaler().fit_transform(com_df)
```

```
df_norm[:10]
```

```
array([[-0.90389781,  1.02375167, -1.34096881, -1.31681816],
       [-1.14661325, -0.13392882, -1.34096881, -1.31681816],
       [-1.3893287 ,  0.32914338, -1.39778675, -1.31681816],
       [-1.51068642,  0.09760728, -1.28415088, -1.31681816],
       [-1.02525553,  1.25528777, -1.34096881, -1.31681816],
       [-0.53982464,  1.94989607, -1.17051501, -1.05354215],
       [-1.51068642,  0.79221557, -1.34096881, -1.18518015],
       [-1.02525553,  0.79221557, -1.28415088, -1.31681816],
       [-1.75340187, -0.36546492, -1.34096881, -1.31681816],
       [-1.14661325,  0.09760728, -1.28415088, -1.44845616]])
```

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
trans_pc = pca.fit_transform(df_norm)
```

```
trans_pc[:10]
```

```
array([[-2.26957633,  0.47908601],
       [-2.08213805, -0.68122811],
       [-2.36712663, -0.34682497],
       [-2.30168974, -0.60342672],
       [-2.39559757,  0.64702343],
       [-2.08352464,  1.49389065],
       [-2.44866953,  0.04533184],
       [-2.23708554,  0.22101257],
       [-2.33554646, -1.1239887 ],
       [-2.18629012, -0.47502155]])
```

```
trans_pc[:-3].shape
```

```
(150, 2)
```

```
trans_pc[-3:]
```

```
array([[ 1.92577832,  0.6018583 ],
       [-2.22920499,  0.28719207],
       [ 0.61244603, -0.60506777]])
```

```
pc_df=pd.DataFrame(data = trans_pc[:-3], columns = ['pc1', 'pc2'])
```

```
pc_df.shape
```

```
(150, 2)
```

```
km.labels_
```

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 0, 0, 0, 2, 0, 0,
0,
       0, 0, 0, 2, 2, 0, 0, 0, 0, 2, 0, 2, 0, 2, 0, 0, 2, 2, 0, 0, 0,
0,
       0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 2])
```

```
pc_df['Cluster']=km.labels_
```

```
pc_df.head()
```

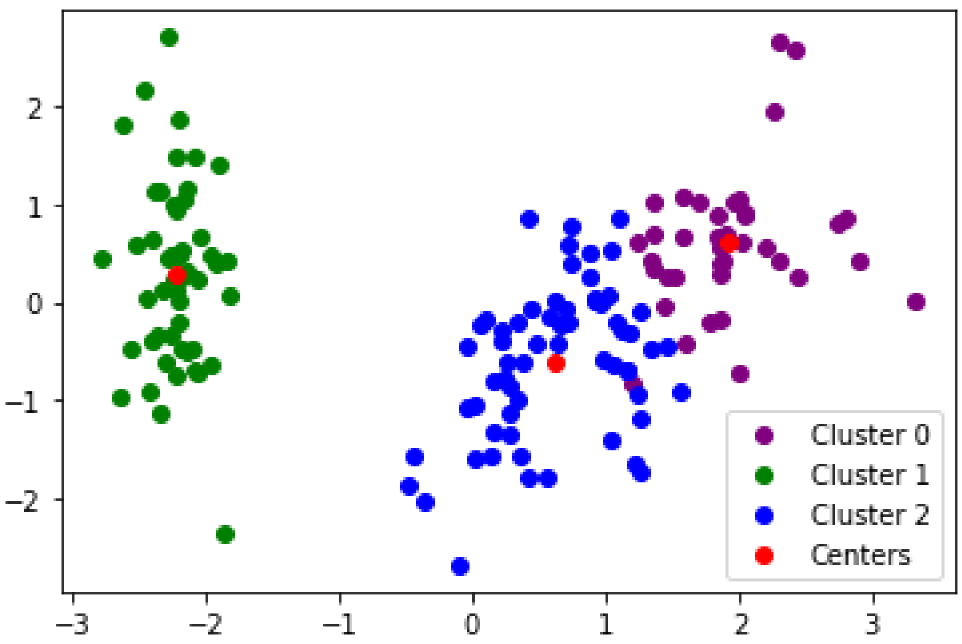
```
        pc1       pc2  Cluster
0 -2.269576  0.479086        1
1 -2.082138 -0.681228        1
2 -2.367127 -0.346825        1
3 -2.301690 -0.603427        1
4 -2.395598  0.647023        1
```

```
first=pc_df[pc_df.Cluster==0]
second=pc_df[pc_df.Cluster==1]
third=pc_df[pc_df.Cluster==2]
```

```
center_df=pd.DataFrame(data = trans_pc[-3:], columns = ['pc1', 'pc2'])
center_df
```

```
        pc1       pc2
0  1.925778  0.601858
1 -2.229205  0.287192
2  0.612446 -0.605068
```

```
plt.scatter(x=first.pc1,y=first.pc2,color='purple',label='Cluster 0')
plt.scatter(x=second.pc1,y=second.pc2,color='green',label='Cluster 1')
plt.scatter(x=third.pc1,y=third.pc2,color='blue',label='Cluster 2')
plt.scatter(x=center_df.pc1,y=center_df.pc2,color='red',label='Centers
')
plt.legend()
plt.show()
```



2
1
0
−1
−2
−3
−2
−1
0
1
2
3
● Cluster 0
● Cluster 1
● Cluster 2
● Centers