k-means-clustering-fl

April 11, 2025

[31]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import load_breast_cancer
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
```

[32]:
```
# Load the Wisconsin Breast Cancer dataset
data = load_breast_cancer()
df = pd.DataFrame(data.data, columns=data.feature_names)
```

[33]:
```
df.head
```

[33]:
```
<bound method NDFrame.head of      mean radius  mean texture  mean perimeter
mean area  mean smoothness  \
0          17.99         10.38          122.80     1001.0          0.11840
1          20.57         17.77          132.90     1326.0          0.08474
2          19.69         21.25          130.00     1203.0          0.10960
3          11.42         20.38           77.58      386.1          0.14250
4          20.29         14.34          135.10     1297.0          0.10030
..           …             …              …         …               …
564        21.56         22.39          142.00     1479.0          0.11100
565        20.13         28.25          131.20     1261.0          0.09780
566        16.60         28.08          108.30      858.1          0.08455
567        20.60         29.33          140.10     1265.0          0.11780
568         7.76         24.54           47.92      181.0          0.05263

     mean compactness  mean concavity  mean concave points  mean symmetry  \
0             0.27760         0.30010              0.14710         0.2419
1             0.07864         0.08690              0.07017         0.1812
2             0.15990         0.19740              0.12790         0.2069
3             0.28390         0.24140              0.10520         0.2597
4             0.13280         0.19800              0.10430         0.1809
..                …             …                    …             …
564           0.11590         0.24390              0.13890         0.1726
565           0.10340         0.14400              0.09791         0.1752
```

566	0.10230	0.09251	0.05302	0.1590
567	0.27700	0.35140	0.15200	0.2397
568	0.04362	0.00000	0.00000	0.1587

	mean fractal dimension	…	worst radius	worst texture	\
0	0.07871	…	25.380	17.33	
1	0.05667	…	24.990	23.41	
2	0.05999	…	23.570	25.53	
3	0.09744	…	14.910	26.50	
4	0.05883	…	22.540	16.67	
..	…	…	…	…	
564	0.05623	…	25.450	26.40	
565	0.05533	…	23.690	38.25	
566	0.05648	…	18.980	34.12	
567	0.07016	…	25.740	39.42	
568	0.05884	…	9.456	30.37	

	worst perimeter	worst area	worst smoothness	worst compactness	\
0	184.60	2019.0	0.16220	0.66560	
1	158.80	1956.0	0.12380	0.18660	
2	152.50	1709.0	0.14440	0.42450	
3	98.87	567.7	0.20980	0.86630	
4	152.20	1575.0	0.13740	0.20500	
..	…	…	…	…	
564	166.10	2027.0	0.14100	0.21130	
565	155.00	1731.0	0.11660	0.19220	
566	126.70	1124.0	0.11390	0.30940	
567	184.60	1821.0	0.16500	0.86810	
568	59.16	268.6	0.08996	0.06444	

	worst concavity	worst concave points	worst symmetry	\
0	0.7119	0.2654	0.4601	
1	0.2416	0.1860	0.2750	
2	0.4504	0.2430	0.3613	
3	0.6869	0.2575	0.6638	
4	0.4000	0.1625	0.2364	
..	…	…	…	
564	0.4107	0.2216	0.2060	
565	0.3215	0.1628	0.2572	
566	0.3403	0.1418	0.2218	
567	0.9387	0.2650	0.4087	
568	0.0000	0.0000	0.2871	

	worst fractal dimension
0	0.11890
1	0.08902
2	0.08758

```
3                    0.17300
4                    0.07678
..                       …
564                  0.07115
565                  0.06637
566                  0.07820
567                  0.12400
568                  0.07039

[569 rows x 30 columns]>
```

[23]:
```
df.keys()
```

[23]:
```
Index(['mean radius', 'mean texture', 'mean perimeter', 'mean area',
       'mean smoothness', 'mean compactness', 'mean concavity',
       'mean concave points', 'mean symmetry', 'mean fractal dimension',
       'radius error', 'texture error', 'perimeter error', 'area error',
       'smoothness error', 'compactness error', 'concavity error',
       'concave points error', 'symmetry error', 'fractal dimension error',
       'worst radius', 'worst texture', 'worst perimeter', 'worst area',
       'worst smoothness', 'worst compactness', 'worst concavity',
       'worst concave points', 'worst symmetry', 'worst fractal dimension'],
      dtype='object')
```

[24]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 30 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   mean radius              569 non-null    float64
 1   mean texture             569 non-null    float64
 2   mean perimeter           569 non-null    float64
 3   mean area                569 non-null    float64
 4   mean smoothness          569 non-null    float64
 5   mean compactness         569 non-null    float64
 6   mean concavity           569 non-null    float64
 7   mean concave points      569 non-null    float64
 8   mean symmetry            569 non-null    float64
 9   mean fractal dimension   569 non-null    float64
 10  radius error             569 non-null    float64
 11  texture error            569 non-null    float64
 12  perimeter error          569 non-null    float64
 13  area error               569 non-null    float64
 14  smoothness error         569 non-null    float64
 15  compactness error        569 non-null    float64
```

```
 16  concavity error          569 non-null    float64
 17  concave points error     569 non-null    float64
 18  symmetry error           569 non-null    float64
 19  fractal dimension error  569 non-null    float64
 20  worst radius             569 non-null    float64
 21  worst texture            569 non-null    float64
 22  worst perimeter          569 non-null    float64
 23  worst area               569 non-null    float64
 24  worst smoothness         569 non-null    float64
 25  worst compactness        569 non-null    float64
 26  worst concavity          569 non-null    float64
 27  worst concave points     569 non-null    float64
 28  worst symmetry           569 non-null    float64
 29  worst fractal dimension  569 non-null    float64
dtypes: float64(30)
memory usage: 133.5 KB
```

[25]:
```
df.isnull().sum()
```

[25]:
```
mean radius                0
mean texture               0
mean perimeter             0
mean area                  0
mean smoothness            0
mean compactness           0
mean concavity             0
mean concave points        0
mean symmetry              0
mean fractal dimension     0
radius error               0
texture error              0
perimeter error            0
area error                 0
smoothness error           0
compactness error          0
concavity error            0
concave points error       0
symmetry error             0
fractal dimension error    0
worst radius               0
worst texture              0
worst perimeter            0
worst area                 0
worst smoothness           0
worst compactness          0
worst concavity            0
worst concave points       0
```

```
worst symmetry             0
worst fractal dimension    0
dtype: int64
```

[26]:
```
df.duplicated().sum()
```

[26]: 0

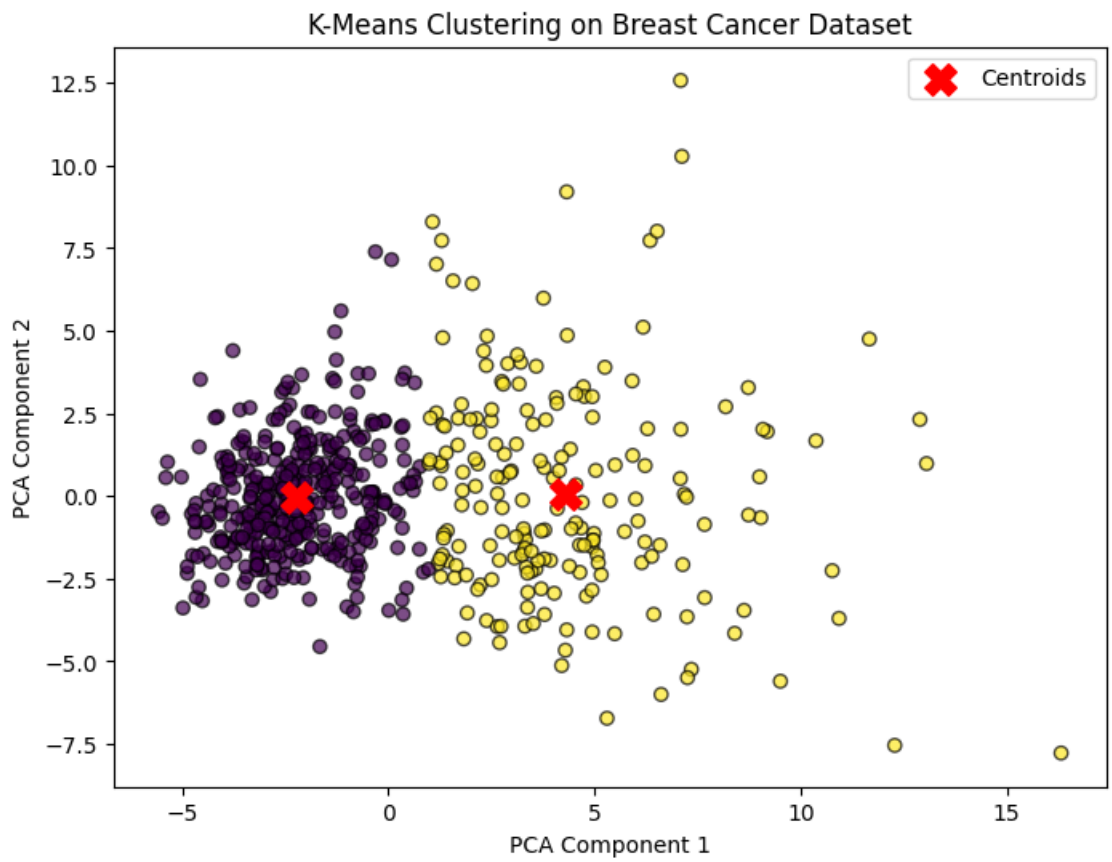
[27]:
```
# Standardizing the features for better clustering performance
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df)
```

[28]:
```
# Apply K-Means Clustering
num_clusters = 2  # We expect two clusters (malignant & benign)
kmeans = KMeans(n_clusters=num_clusters, random_state=42, n_init=10)
df['Cluster'] = kmeans.fit_predict(X_scaled)
```

[29]:
```
# Reduce dimensionality to 2D using PCA for visualization
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
df['PCA1'] = X_pca[:, 0]
df['PCA2'] = X_pca[:, 1]
```

[30]:
```
# Transform centroids into PCA space
centroids_pca = pca.transform(kmeans.cluster_centers_)

# Plot clustering results
plt.figure(figsize=(8, 6))
plt.scatter(df['PCA1'], df['PCA2'], c=df['Cluster'], cmap='viridis', alpha=0.7,␣
 ↪edgecolors='k')
plt.scatter(centroids_pca[:, 0], centroids_pca[:, 1], s=200, c='red',␣
 ↪marker='X', label='Centroids')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.title('K-Means Clustering on Breast Cancer Dataset')
plt.legend()
plt.show()
```


K-Means Clustering on Breast Cancer Dataset
Centroids
PCA Component 2
PCA Component 1
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−5
0
5
10
15

[ ]:

[ ]: