ds-practical-8

October 29, 2024

0.1 DS PRACTICAL 8

0.2 Performing Clustering using Scikit-learn for a given problem statement

---

0.2.1 X contain training data samples

[73]:
```
import numpy as np

X=[[3,280],[5,320],[8,80],[2,225],[5,80],[6,265],[1,71],[5,235],[7,98],[4,300],[2,210]]
Xarr = np.array(X)
```

0.2.2 Setting the number of clusters for data

[74]:
```
from sklearn.cluster import KMeans

kmeans=KMeans(n_clusters=4)
kmeans=kmeans.fit(X)  # Fitting the input data
kmeans
```

[74]:
```
KMeans(n_clusters=4)
```

0.2.3 Getting the Co-ordinates of Centroids

[75]:
```
centroids = kmeans.cluster_centers_
print("\nCluster centers are :")

for i, center in enumerate(centroids):
    print(f"Cluster {i+1} center: [{center[0]:.2f}, {center[1]:.2f}]")
```

```
Cluster centers are :
Cluster 1 center: [4.50, 272.50]
Cluster 2 center: [5.25, 82.25]
```

```
Cluster 3 center: [3.00, 223.33]
Cluster 4 center: [4.50, 310.00]
```

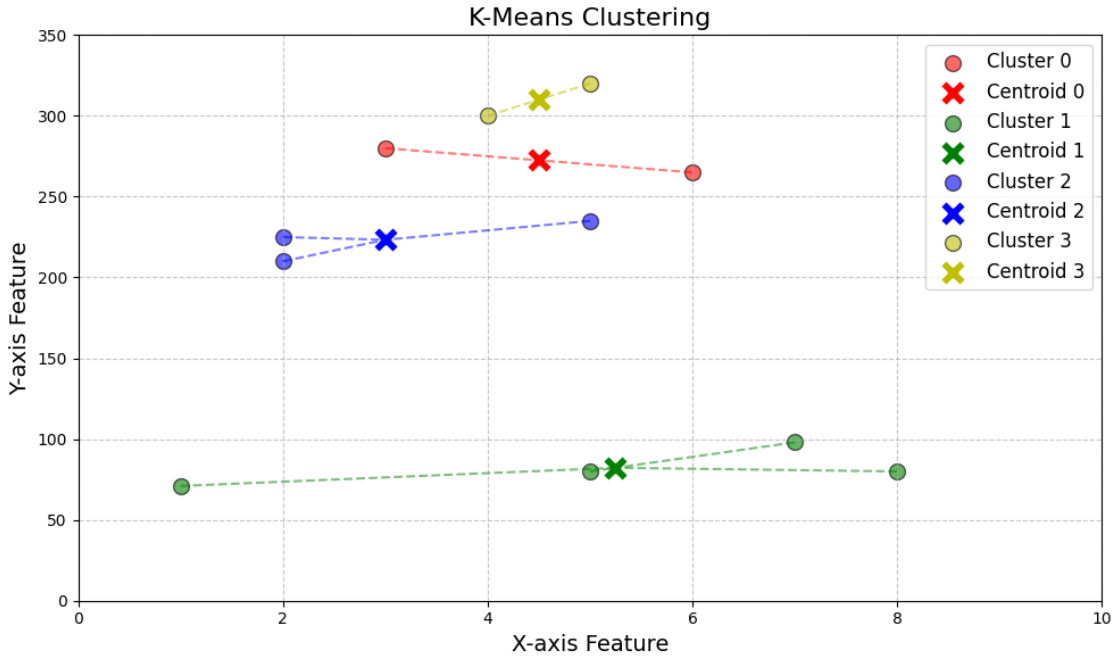
0.2.4 Predicting the Cluster Tables for each data value

[76]:
```
labels=kmeans.predict(X)
print("\nCluster labels are :\n",labels)
```

```
Cluster labels are :
 [0 3 1 2 1 0 1 2 1 3 2]
```

0.2.5 Plotting the Clusters

[77]:
```
import matplotlib.pyplot as plt

colors=['r','g','b','y']

fig, ax = plt.subplots(figsize=(10, 6))

for i in range(4):
    points = np.array([Xarr[j] for j in range(len(Xarr)) if labels[j] == i])
    ax.scatter(points[:, 0], points[:, 1], s=100, c=colors[i],
               alpha=0.6, edgecolors='k', label=f"Cluster {i}")

    ax.scatter(centroids[i, 0], centroids[i, 1], marker='X', s=300,
               c=colors[i], edgecolor='white', linewidth=2, label=f"Centroid␣
 ↪{i}")

    for point in points:
        ax.plot([point[0], centroids[i, 0]], [point[1], centroids[i, 1]],
                color=colors[i], linestyle='--', alpha=0.5)

ax.set_title("K-Means Clustering", fontsize=16)
ax.set_xlabel("X-axis Feature", fontsize=14)
ax.set_ylabel("Y-axis Feature", fontsize=14)
ax.legend(fontsize=12)
ax.grid(True, linestyle='--', alpha=0.7)
ax.set_xlim(0, 10)
ax.set_ylim(0, 350)

plt.tight_layout()
plt.show()
```



0.2.6 Example

Predicting Class of a Input Pattern of [4 , 200]

Expected Output : Pattern belongs to Cluster [2]. i.e Blue

[78]:
```
xnew=[[4,200]]
print("Actual Output :-")
print("\nPattern ",xnew," belongs to Cluster ",kmeans.predict(xnew))
```

```
Actual Output :-

Pattern  [[4, 200]]  belongs to Cluster  [2]
```