05-basic-clustering

April 22, 2025

1 Topic 3: Basic Clustering Exercise

Adding 3 more models into the basic clustering exercise

1.0.1 Import necessary libraries

[1]:
```
from numpy import unique, where
import matplotlib.pyplot as plt
```

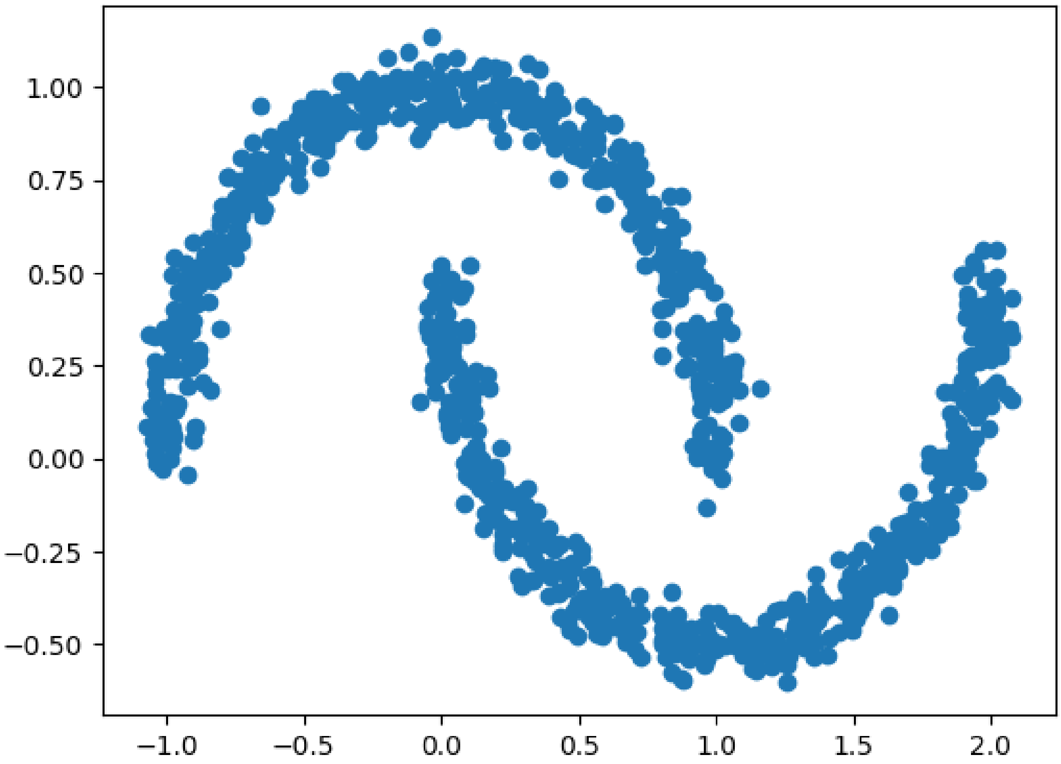
1.0.2 Import and display the dataset

We'll use make_moons dataset, which is a built-in dataset in scikit-learn library.

Can read more about it here.

[2]:
```
from sklearn.datasets import make_moons
X, _ = make_moons(noise=0.05, random_state=0, n_samples=1000)

plt.scatter(X[:,0], X[:,1]) # visualize the dataset
```

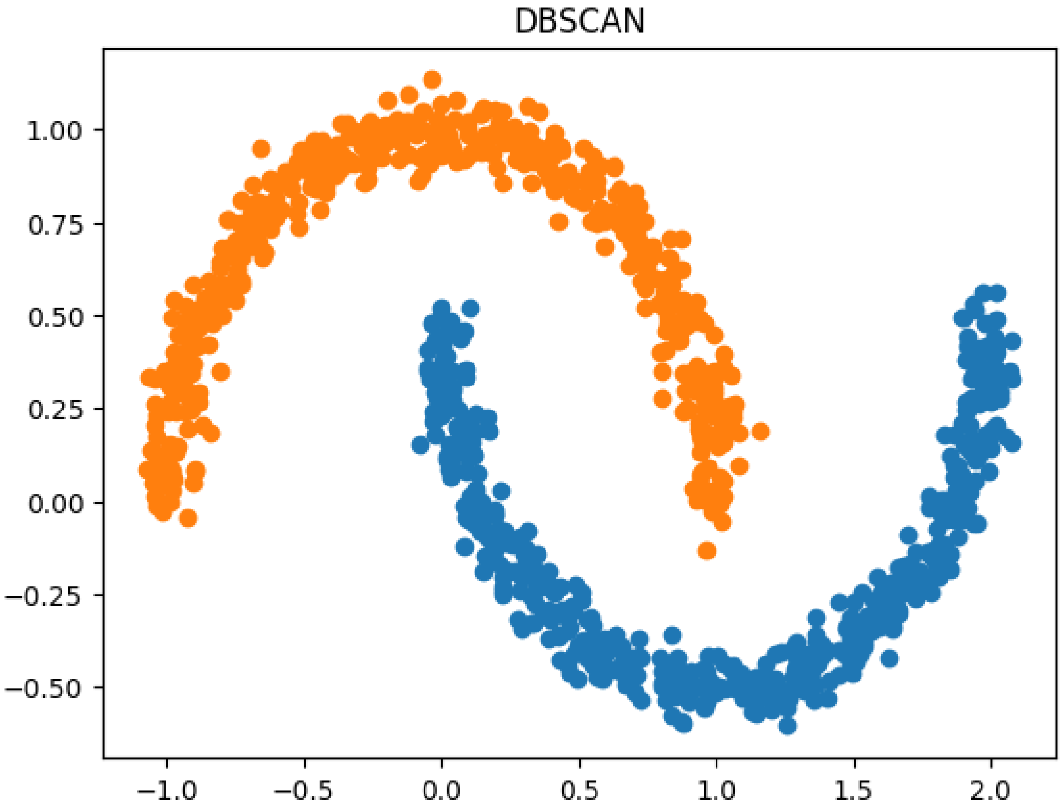
[2]:
```
<matplotlib.collections.PathCollection at 0x1c67abf7a70>
```



Model

DBSCAN

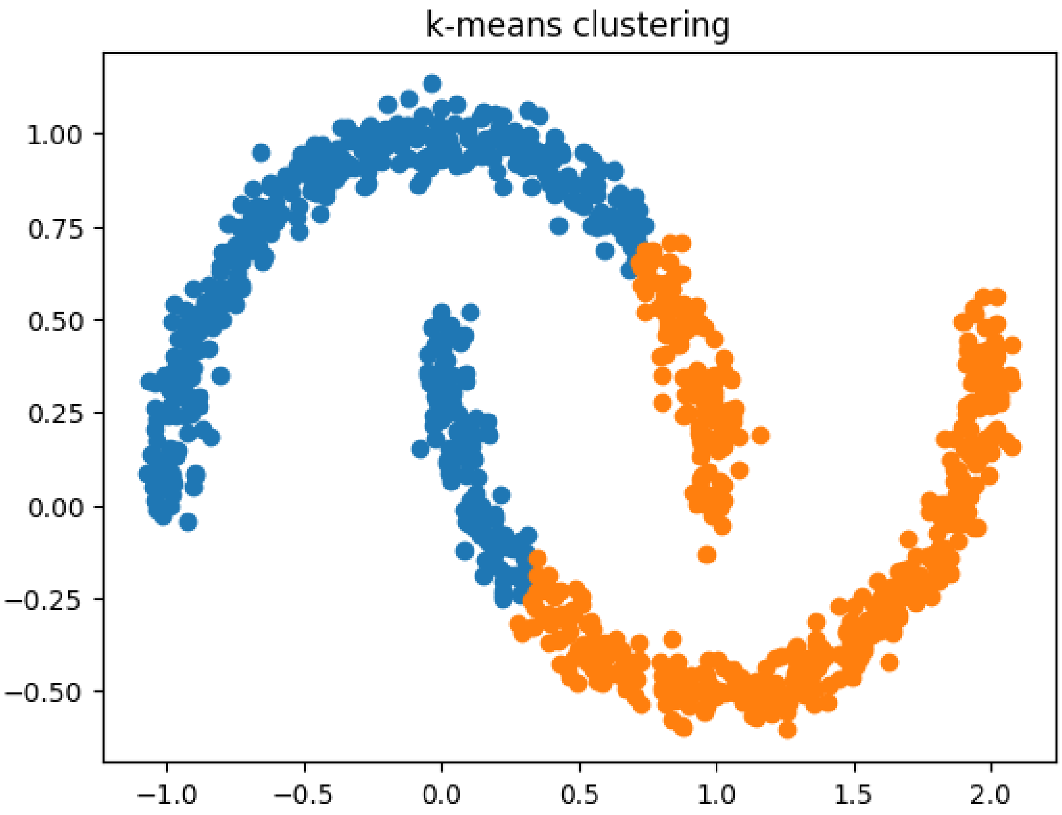
[3]:
```python
from sklearn.cluster import DBSCAN

model = DBSCAN(eps=0.2, min_samples=9)

yhat = model.fit_predict(X)

clusters = unique(yhat)

for cluster in clusters:
    row_ix = where(yhat == cluster)
    plt.scatter(X[row_ix, 0], X[row_ix, 1])
plt.title("DBSCAN")
plt.show()
```



k-Means Clustering

[4]:

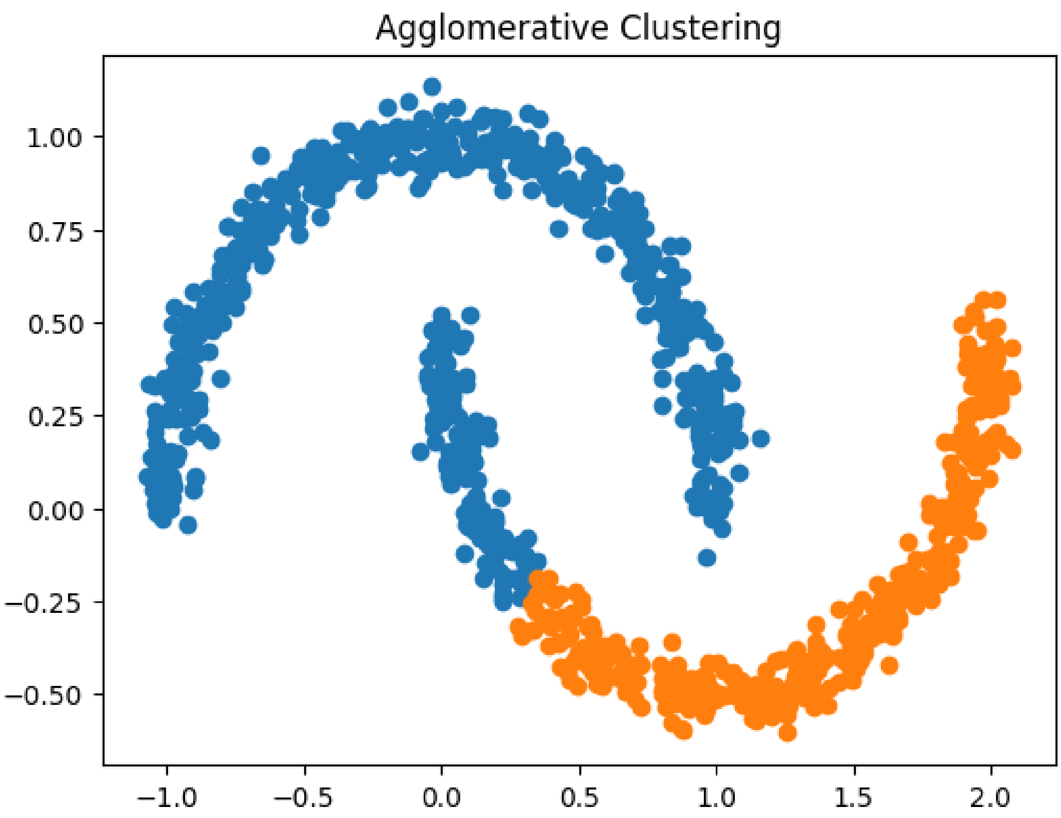
```
from sklearn.cluster import KMeans

model = KMeans(n_clusters=2)

model.fit(X)

yhat = model.predict(X)

clusters = unique(yhat)

for cluster in clusters:
    row_ix = where(yhat == cluster)
    plt.scatter(X[row_ix, 0], X[row_ix, 1])
plt.title("k-means clustering")
plt.show()
```



Hierarchical Clustering

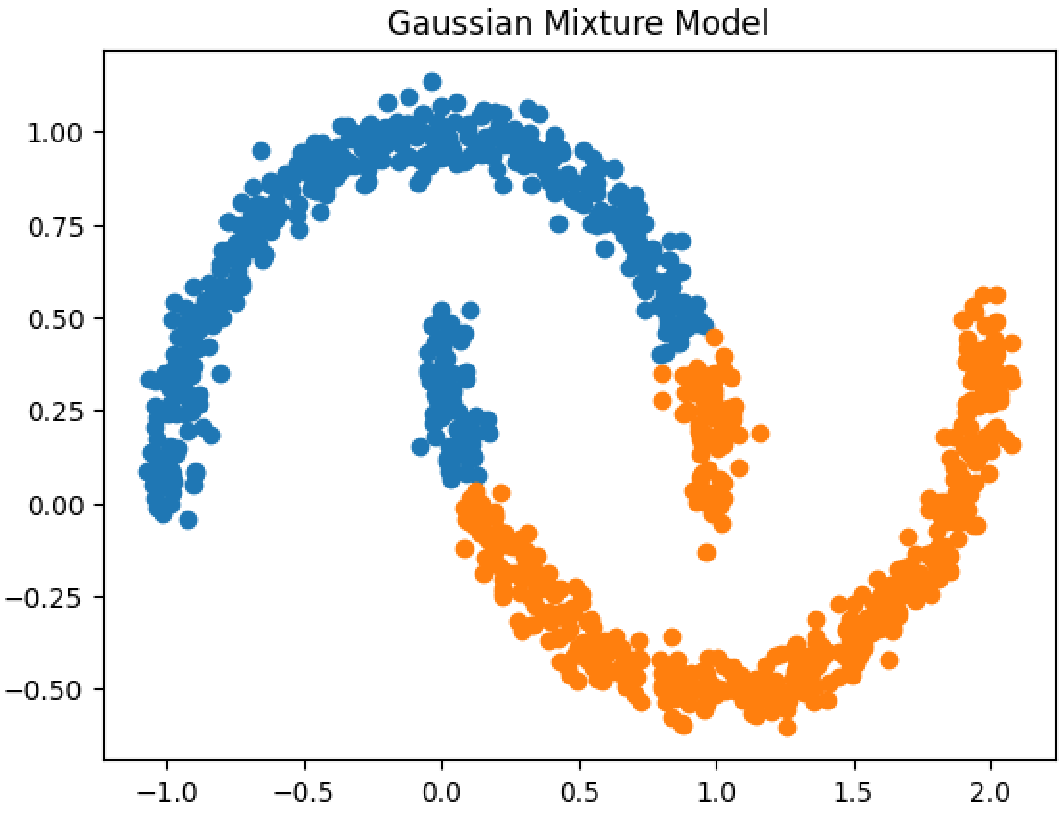
[5]:

```python
from sklearn.cluster import AgglomerativeClustering

model = AgglomerativeClustering(n_clusters=2)
yhat = model.fit_predict(X)

clusters = unique(yhat)
for cluster in clusters:
    row_ix = where(yhat == cluster)
    plt.scatter(X[row_ix, 0], X[row_ix, 1])
plt.title("Agglomerative Clustering")
plt.show()
```



Gaussian Mixture Model (GMM)

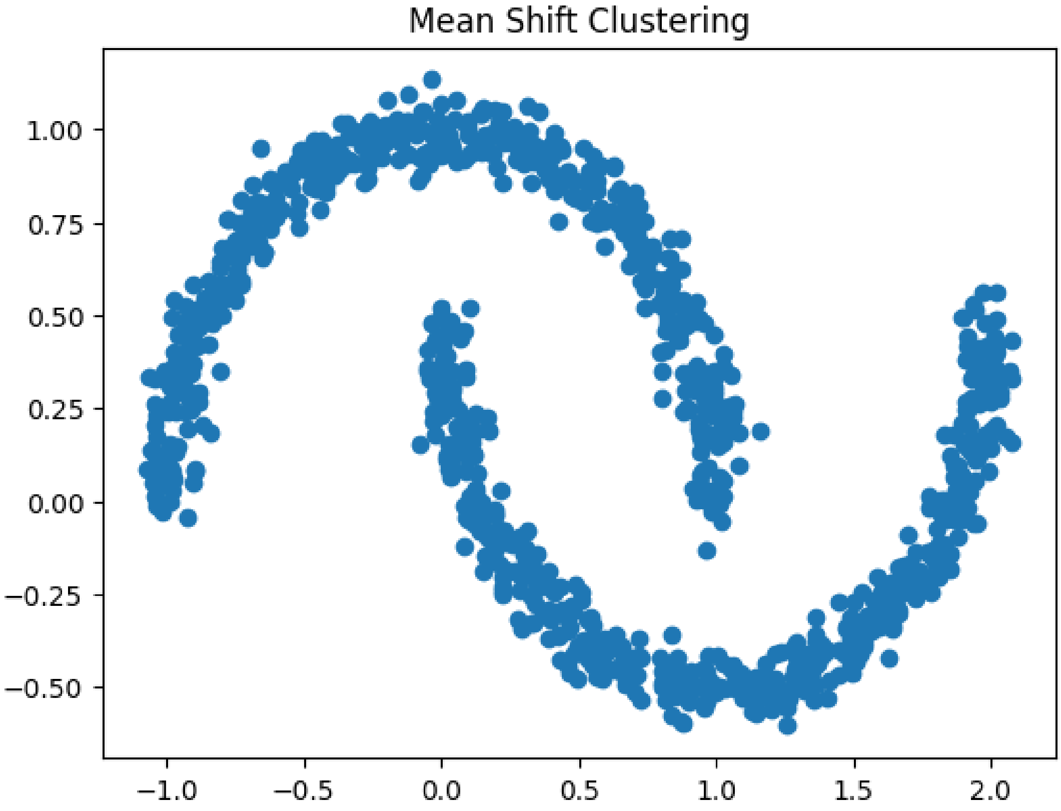
[6]:
```
from sklearn.mixture import GaussianMixture

model = GaussianMixture(n_components=2)
model.fit(X)
yhat = model.predict(X)

clusters = unique(yhat)
for cluster in clusters:
    row_ix = where(yhat == cluster)
    plt.scatter(X[row_ix, 0], X[row_ix, 1])
plt.title("Gaussian Mixture Model")
plt.show()
```



Mean Shift Clustering

[7]:

```
from sklearn.cluster import MeanShift

model = MeanShift()
yhat = model.fit_predict(X)

clusters = unique(yhat)
for cluster in clusters:
    row_ix = where(yhat == cluster)
    plt.scatter(X[row_ix, 0], X[row_ix, 1])
plt.title("Mean Shift Clustering")
plt.show()
```


Mean Shift Clustering
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0