```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.cluster import DBSCAN
import pandas as pd
```

```
# Load the IRIS dataset
iris = load_iris()
X = iris.data
X
```

```
            [5.8, 2.6, 4. , 1.2],
            [5. , 2.3, 3.3, 1. ],
            [5.6, 2.7, 4.2, 1.3],
            [5.7, 3. , 4.2, 1.2],
            [5.7, 2.9, 4.2, 1.3],
            [6.2, 2.9, 4.3, 1.3],
            [5.1, 2.5, 3. , 1.1],
            [5.7, 2.8, 4.1, 1.3],
            [6.3, 3.3, 6. , 2.5],
            [5.8, 2.7, 5.1, 1.9],
            [7.1, 3. , 5.9, 2.1],
            [6.3, 2.9, 5.6, 1.8],
            [6.5, 3. , 5.8, 2.2],
            [7.6, 3. , 6.6, 2.1],
            [4.9, 2.5, 4.5, 1.7],
            [7.3, 2.9, 6.3, 1.8],
            [6.7, 2.5, 5.8, 1.8],
            [7.2, 3.6, 6.1, 2.5],
            [6.5, 3.2, 5.1, 2. ],
            [6.4, 2.7, 5.3, 1.9],
            [6.8, 3. , 5.5, 2.1],
            [5.7, 2.5, 5. , 2. ],
            [5.8, 2.8, 5.1, 2.4],
            [6.4, 3.2, 5.3, 2.3],
            [6.5, 3. , 5.5, 1.8],
            [7.7, 3.8, 6.7, 2.2],
            [7.7, 2.6, 6.9, 2.3],
            [6. , 2.2, 5. , 1.5],
            [6.9, 3.2, 5.7, 2.3],
            [5.6, 2.8, 4.9, 2. ],
            [7.7, 2.8, 6.7, 2. ],
            [6.3, 2.7, 4.9, 1.8],
            [6.7, 3.3, 5.7, 2.1],
            [7.2, 3.2, 6. , 1.8],
            [6.2, 2.8, 4.8, 1.8],
            [6.1, 3. , 4.9, 1.8],
            [6.4, 2.8, 5.6, 2.1],
            [7.2, 3. , 5.8, 1.6],
            [7.4, 2.8, 6.1, 1.9],
            [7.9, 3.8, 6.4, 2. ],
            [6.4, 2.8, 5.6, 2.2],
            [6.3, 2.8, 5.1, 1.5],
            [6.1, 2.6, 5.6, 1.4],
            [7.7, 3. , 6.1, 2.3],
            [6.3, 3.4, 5.6, 2.4],
            [6.4, 3.1, 5.5, 1.8],
            [6. , 3. , 4.8, 1.8],
            [6.9, 3.1, 5.4, 2.1],
            [6.7, 3.1, 5.6, 2.4],
            [6.9, 3.1, 5.1, 2.3],
            [5.8, 2.7, 5.1, 1.9],
            [6.8, 3.2, 5.9, 2.3],
            [6.7, 3.3, 5.7, 2.5],
            [6.7, 3. , 5.2, 2.3],
            [6.3, 2.5, 5. , 1.9],
            [6.5, 3. , 5.2, 2. ],
            [6.2, 3.4, 5.4, 2.3],
            [5.9, 3. , 5.1, 1.8]])
```

```
# Fit DBSCAN model for hard clustering
dbscan_hard = DBSCAN(eps=0.5, min_samples=5)
y_hard = dbscan_hard.fit_predict(X)
```

```
# Fit DBSCAN model for soft clustering
dbscan_soft = DBSCAN(eps=0.5, min_samples=5)
dbscan_soft.fit(X)
```

```
▾ DBSCAN
DBSCAN()
```

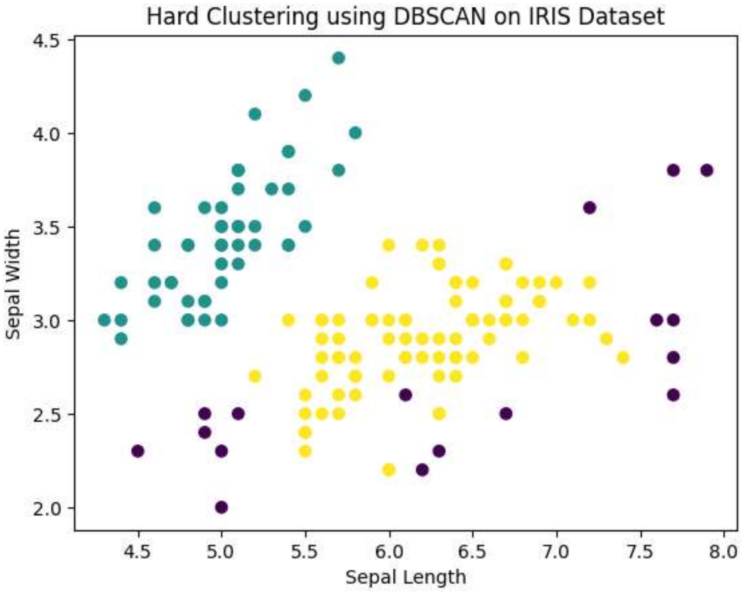
```
soft_labels = dbscan_soft.labels_
print(soft_labels)
```

```
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0
  0  0  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  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]
```

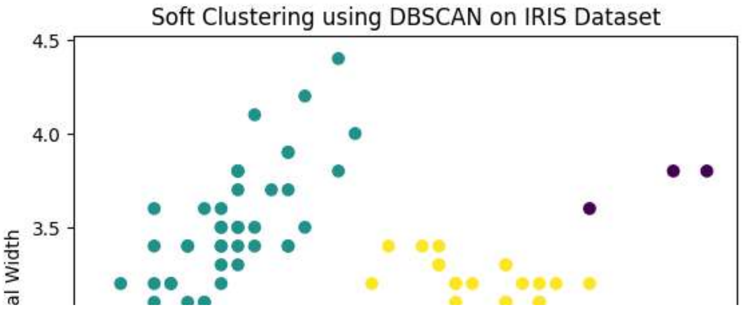
```
soft_scores = dbscan_soft.core_sample_indices_
print(soft_scores)
```

```
[  0   1   2   3   4   5   6   7   8   9  10  11  12  13  16  17  19  20
  21  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39
  40  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  58  61
  63  65  66  67  69  70  71  72  73  74  75  76  77  78  79  80  81  82
  83  84  85  86  88  89  90  91  92  94  95  96  97  99 101 102 103 104
 110 111 112 115 116 120 121 123 124 125 126 127 128 132 133 136 137 138
 139 140 142 143 144 145 146 147 149]
```

```
# Plot hard clustering results
plt.scatter(X[:, 0], X[:, 1], c=y_hard)
plt.title("Hard Clustering using DBSCAN on IRIS Dataset")
plt.xlabel("Sepal Length")
plt.ylabel("Sepal Width")
plt.show()
```



```
# Plot soft clustering results
plt.scatter(X[:, 0], X[:, 1], c=soft_labels, cmap='viridis')
plt.title("Soft Clustering using DBSCAN on IRIS Dataset")
plt.xlabel("Sepal Length")
plt.ylabel("Sepal Width")
plt.show()
```



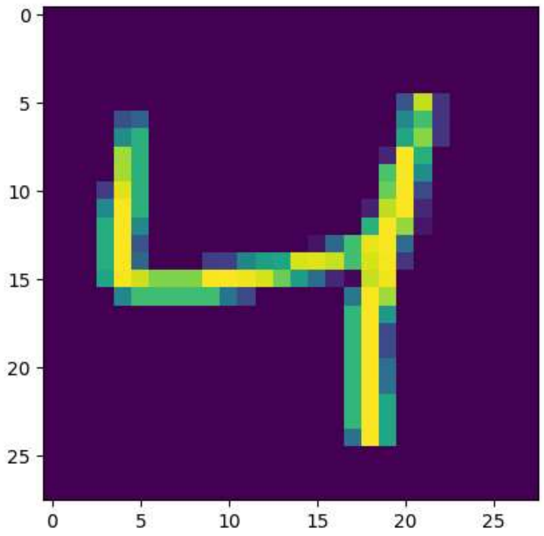
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_openml
from sklearn.decomposition import PCA
from sklearn.cluster import DBSCAN
```

```
# Load the MNIST dataset
mnist = fetch_openml('mnist_784', version=1)
X = mnist.data
y = mnist.target.astype(int)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/datasets/_openml.py:968: FutureWarning: The default value of `parser` will change from `
  warn(
```

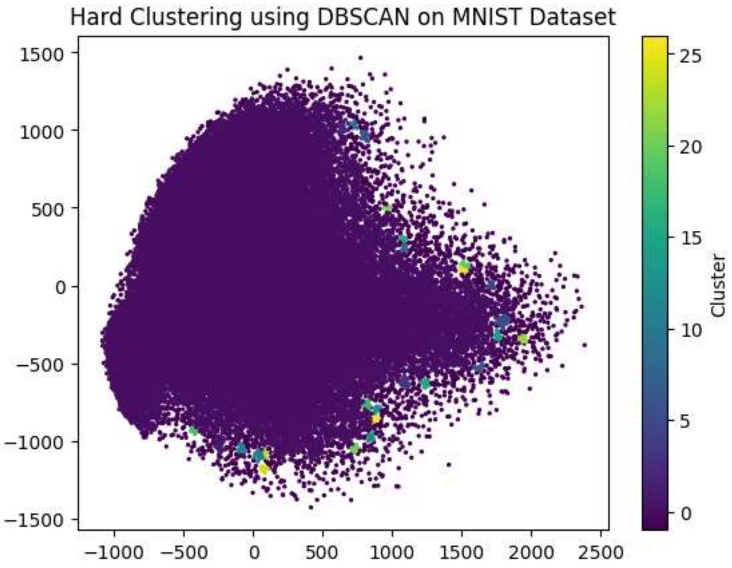
```
sample_a = X.loc[[2]].to_numpy()
sample_b = X.loc[[25]].to_numpy()
sample_a_img = sample_a.reshape(28, 28)
sample_b_img = sample_b.reshape(28, 28)
plt.imshow(sample_a_img)
```

```
<matplotlib.image.AxesImage at 0x7f5f3536bf10>
```

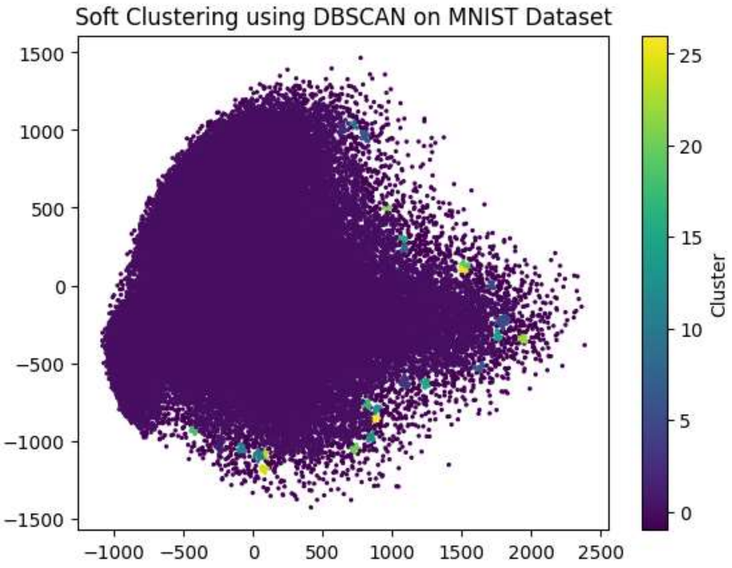


```
# Reduce dimensionality using PCA
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)


# Fit DBSCAN model for hard clustering
dbscan_hard = DBSCAN(eps=30, min_samples=10)
y_hard = dbscan_hard.fit_predict(X_pca)


# Fit DBSCAN model for soft clustering
dbscan_soft = DBSCAN(eps=30, min_samples=10)
dbscan_soft.fit(X_pca)
soft_labels = dbscan_soft.labels_
soft_scores = dbscan_soft.core_sample_indices_
```

```
# Plot hard clustering results
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y_hard, cmap='viridis', s=2)
plt.colorbar(label='Cluster')
plt.title("Hard Clustering using DBSCAN on MNIST Dataset")
plt.show()
```



```
# Plot soft clustering results
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=soft_labels, cmap='viridis', s=2)
plt.colorbar(label='Cluster')
plt.title("Soft Clustering using DBSCAN on MNIST Dataset")
plt.show()
```



● ×