5e-DBSCAN

November 8, 2024

1 DBSCAN

1.1 Imports

[ ]:
```
import numpy as np
import pandas as pd
import math
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.cluster import DBSCAN
from sklearn.datasets import make_blobs
```

1.2 Blob Example

[ ]:
```
X, y_true = make_blobs(n_samples=300, centers=4,cluster_std=0.60,␣
 ↪random_state=0)
plt.scatter(X[:, 0], X[:, 1], s=50)
;
```

1.2.1 K-Means

[ ]:
```
kmeans = KMeans(n_clusters=4, n_init=10).fit(X)
```

[ ]:
```
plt.figure(figsize = (7,7))
sns.scatterplot(x = X[:,0], y = X[:,1], hue = kmeans.labels_) ;
```

1.2.2 DBSCAN

[ ]:
```
dbscan=DBSCAN(eps=.8,min_samples=9)
dbscan.fit(X)
;
```

[ ]:
```
plt.figure(figsize = (7,7))
sns.scatterplot(x = X[:,0], y = X[:,1], hue=dbscan.labels_) ;
```

1.3 Non-Blob Example

1.3.1 Create Random Data

[ ]:
```
np.random.seed(100)

# Function for creating datapoints in the form of a circle
def PointsInCircum(r,n=100):
    '''This does math stuff'''
    return [ (math.cos(2*math.pi/n*x)*r+np.random.normal(-30,30),
              math.sin(2*math.pi/n*x)*r+np.random.normal(-30,30))
                for x in range(1,n+1)
            ]
```

[ ]:
```
# Creating data points in the form of a circle
dfs = [ pd.DataFrame(PointsInCircum(500,1000)) ]
dfs[0].shape
```

[ ]:
```
# Add another circle inside
dfs += [ pd.DataFrame( PointsInCircum(300,700) ) ]
dfs[1].shape
```

[ ]:
```
# Adding noise to the dataset
dfs += [ pd.DataFrame( ( np.random.randint(-600,600), np.random.
 ↪randint(-600,600) ) for i in range(300) ) ]
dfs[2].shape
```

[ ]:
```
# Combine data sets
df = pd.concat( dfs )
df.shape
```

[ ]:
```
# Plotting data
plt.figure(figsize=(8,8))
plt.scatter(df[0],df[1],s=15,color='grey')
plt.xlabel('Feature 1',fontsize=14)
plt.ylabel('Feature 2',fontsize=14)
plt.show() ;
```

1.3.2 K-means

[ ]:
```
kmeans=KMeans(n_clusters=2, random_state=42, n_init=10).fit(df)
```

[ ]:
```
plt.figure(figsize = (7,7))
sns.scatterplot(x = df[0], y = df[1], hue=kmeans.labels_) ;
```

1.3.3 DBSCAN

[ ]:
```
dbscan=DBSCAN(eps=40, min_samples=7)
dbscan.fit(df) ;
```

[ ]:
```
plt.figure(figsize = (7,7))
sns.scatterplot(x = df[0], y = df[1], hue=dbscan.labels_) ;
```

[ ]:
```

```