# DBSCAN_Clustering

December 14, 2021

## 1 Clustering Data With DBSCAN On Python

### 1.1 Please cluster the following data with DBSCAN Algorithm

### 1.2 [[3, 2, 1], [5, 5, 6], [4, 5, 5],[3, 3, 2], [7, 6, 6], [5, 5,4], [1, 0 ,1],[7, 8, 7]]

[ ]:

```
from matplotlib import pyplot as plt
from sklearn.cluster import DBSCAN
import pandas as pd

# Use Pandas lib for Create Dataframe
MyData = pd.DataFrame([[1, 0, 1],
                       [7, 8, 7],
                       [3, 3, 2],
                       [5, 5, 6],
                       [3, 2, 1],
                       [2, 2, 2],
                       [5, 5, 4],
                       [7, 6 ,6],
                       [4, 5, 5]],columns=['F1', 'F2',␣
 ↪'F3'],index={'X1','X2','X3','X4','X5','X6','X7','X8','X9'})
print('Show Our Data')
MyData
```

```
Show Our Data
```

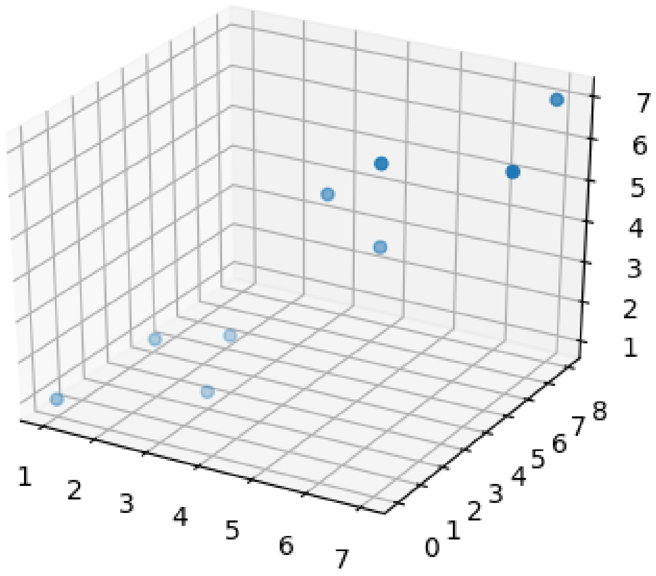
[ ]:

```
    F1  F2  F3
X8   1   0   1
X9   7   8   7
X4   3   3   2
X2   5   5   6
X1   3   2   1
X7   2   2   2
X6   5   5   4
X5   7   6   6
X3   4   5   5
```

## 2 Plotting Data

[ ]:
```
fig = plt.figure(figsize=(5, 4), dpi=100)
ax = plt.axes(projection='3d')
ax = plt.axes(projection='3d')
ax.scatter3D(MyData['F1'], MyData['F2'], MyData['F3'])
```

[ ]:
```
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x7f7454ef2650>
```



**eps = The maximum distance between two samples for one to be considered as in the neighborhood of the other.**

**min_samples = The number of samples (or total weight) in a neighborhood for a point to be considered as a core point**

**eps = 3 && min_samples = 3**

[ ]:
```
clustering = DBSCAN(eps=3, min_samples=3).fit(MyData)
MyData['clusts'] = clustering.labels_
print('Show Clustered Data')
MyData
```

```
Show Clustered Data
```

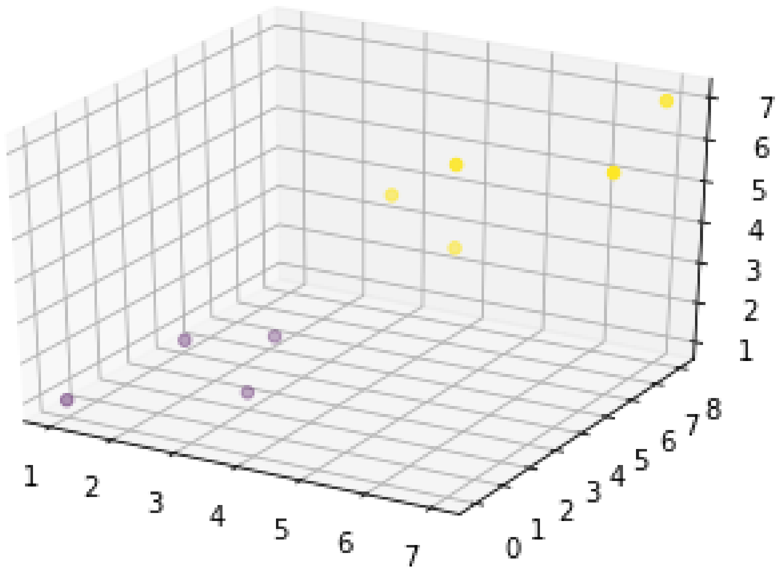
[ ]:
```
    F1  F2  F3  clusts
X8   1   0   1       0
X9   7   8   7       1
X4   3   3   2       0
```

```
X2   5   5   6        1
X1   3   2   1        0
X7   2   2   2        0
X6   5   5   4        1
X5   7   6   6        1
X3   4   5   5        1
```

[ ]:

```
# Use Matplotlib For plotting Clustered data
fig = plt.figure()
ax = plt.axes(projection='3d')
ax = plt.axes(projection='3d')
ax.scatter3D(MyData['F1'], MyData['F2'], MyData['F3'], c=MyData['clusts'])
print ('Plotting Clustered Data')
```

```
Plotting Clustered Data
```



[ ]:

```
print('Show Sorted by Clustered Data Label')
MyData.sort_values(by=['clusts'])
```

```
Show Sorted by Clustered Data Label
```

[ ]:

```
    F1  F2  F3  clusts
X8   1   0   1       0
X4   3   3   2       0
X1   3   2   1       0
X7   2   2   2       0
X9   7   8   7       1
```

```
X2   5   5   6         1
X6   5   5   4         1
X5   7   6   6         1
X3   4   5   5         1
```

## 3 Result is $=>$ C1=X1,X4,X7,X8 & C2=X5,X3,X6,X2,X9