Assignment#2 - Unsupervised Learning

이름 : 전예찬

학번 : 20194902

소속 : 소프트웨어대학 소프트웨어학부

1. 테스트 환경

운영체제	Windows 11 23H Education edition
개발환경	Juypter Notebook (local)
Kernel	Python 3 (ipykernel)

2. 실험 준비

a. Dataset

- Fashion MNIST data (테스트 dataset 만 사용)
    - **images**

shape	(10000, 784)
data type	float32 array

- - **labels**

shape	(10000,)
data type	array

- images, labels 전처리
    - images to pandas.DataFrame (변수 이름 : X_)

0	1	2	3	4	5	6
0	0.0	0.0	0.0	0.000000	0.0	0.000000
1	0.0	0.0	0.0	0.000000	0.0	0.000000
2	0.0	0.0	0.0	0.000000	0.0	0.000000
3	0.0	0.0	0.0	0.000000	0.0	0.000000
4	0.0	0.0	0.0	0.007843	0.0	0.003922
...	...	...	...	...	...	...
9995	0.0	0.0	0.0	0.000000	0.0	0.000000
9996	0.0	0.0	0.0	0.000000	0.0	0.000000
9997	0.0	0.0	0.0	0.000000	0.0	0.000000
9998	0.0	0.0	0.0	0.000000	0.0	0.000000
9999	0.0	0.0	0.0	0.000000	0.0	0.000000

10000 rows × 784 columns

- labels to pandas.Series (변수 이름 : y_)

```
0       9
1       2
2       1
3       1
4       6
       ..
9995    9
9996    1
9997    8
9998    1
9999    5
Length: 10000, dtype: int64
```

b. Experimental Setup

Clustering Algorithm

- **KMeans**
    - **구현체** : sklearn.cluster.KMeans 클래스
    - **Hyper Paramter** :
        - n_cluster : 10
        - 나머지 파라미터는 모두 default parameter set 사용

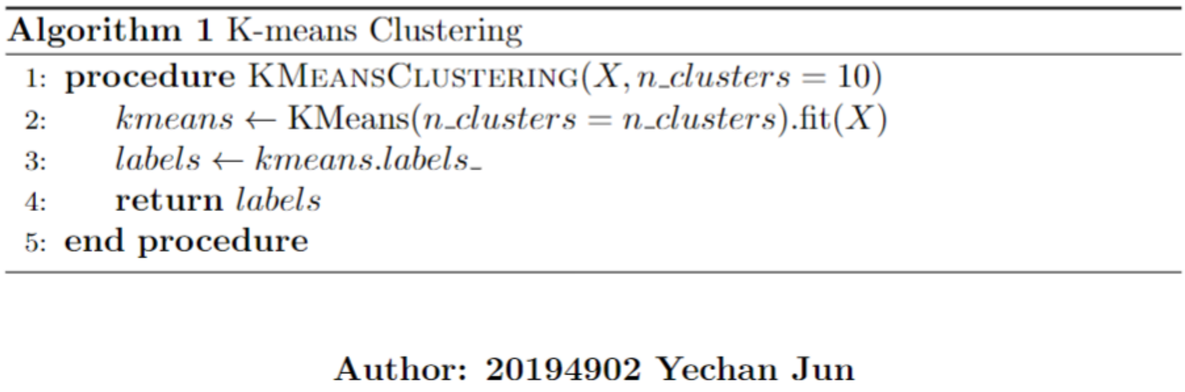
- **DBSCAN**
    - **구현체** : sklearn.cluster.DBSCAN
    - **Hyper Parameter** :
        - eps (epsilon) : parameter search 알고리즘을 통해 구한 최적 epsilon 값
        - min_samples : parameter search 알고리즘을 통해 구한 최적 min_samples 값
        - n_jobs : -1 (all processor)
        - 나머지 파라미터는 모두 default parameter set 사용

c. Pseudo code of Algorithms

아래에 이번 과제에서 사용한 알고리즘들을 pseudo code형태로 나열.

- **Helper algorithms for main learning process**

[KMeans clustering algorithm]

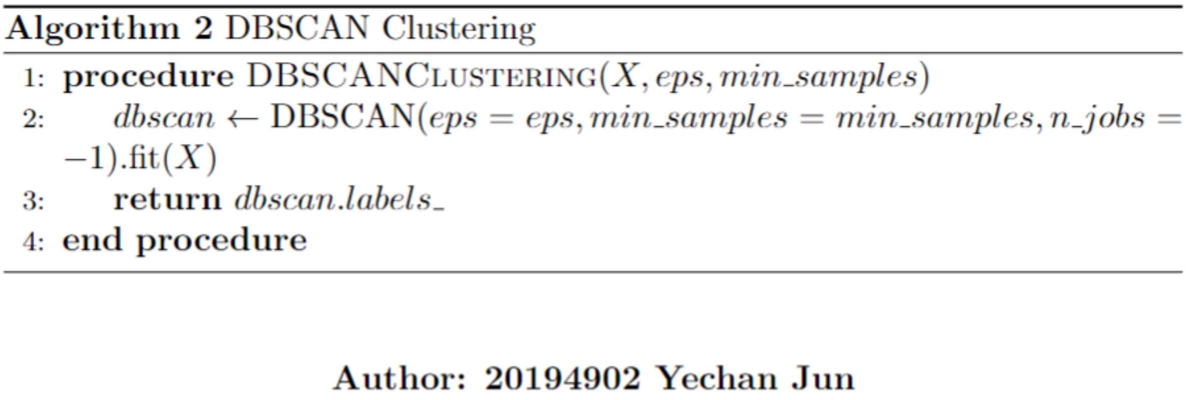


Algorithm 1 K-means Clustering

```
procedure KMeansClustering($X, n\_clusters = 10$)
    $kmeans \leftarrow$ KMeans($n\_clusters = n\_clusters$).fit($X$)
    $labels \leftarrow kmeans.labels\_$
    return $labels$
end procedure
```

Author: 20194902 Yechan Jun

[DBSCAN clustering algorithm]

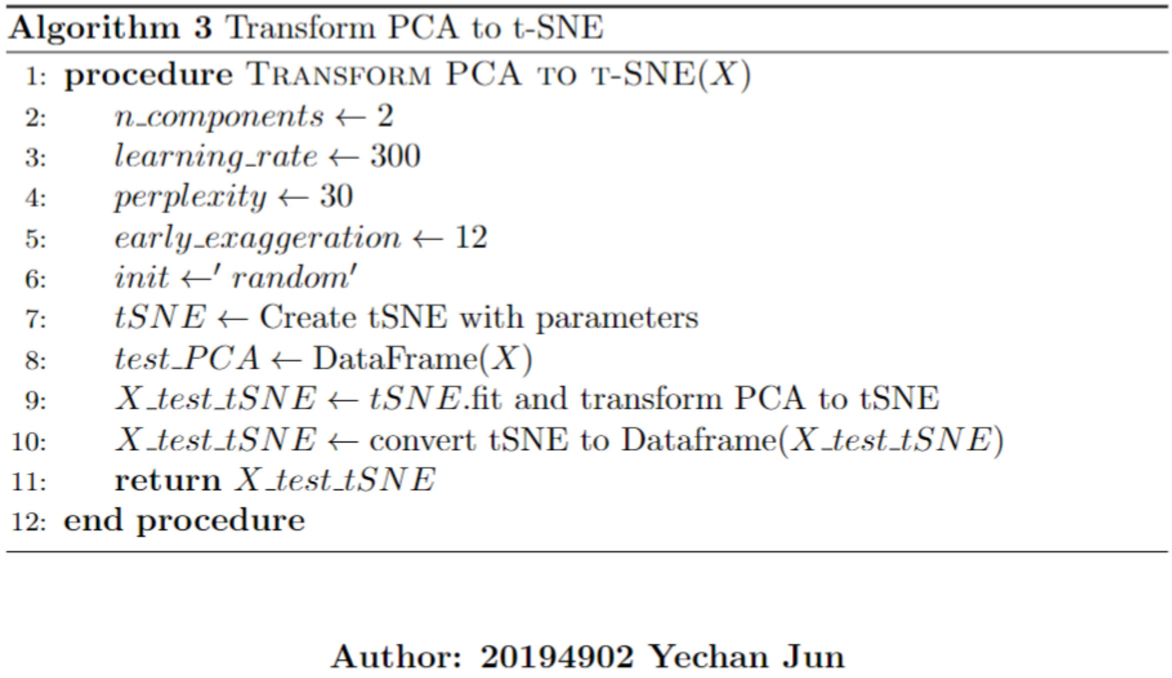


Algorithm 2 DBSCAN Clustering

```
procedure DBSCANClustering($X, eps, min\_samples$)
    $dbscan \leftarrow$ DBSCAN($eps = eps, min\_samples = min\_samples, n\_jobs = -1$).fit($X$)
    return $dbscan.labels\_$
end procedure
```

Author: 20194902 Yechan Jun

[Transform from PCA dataset to t-SNE]

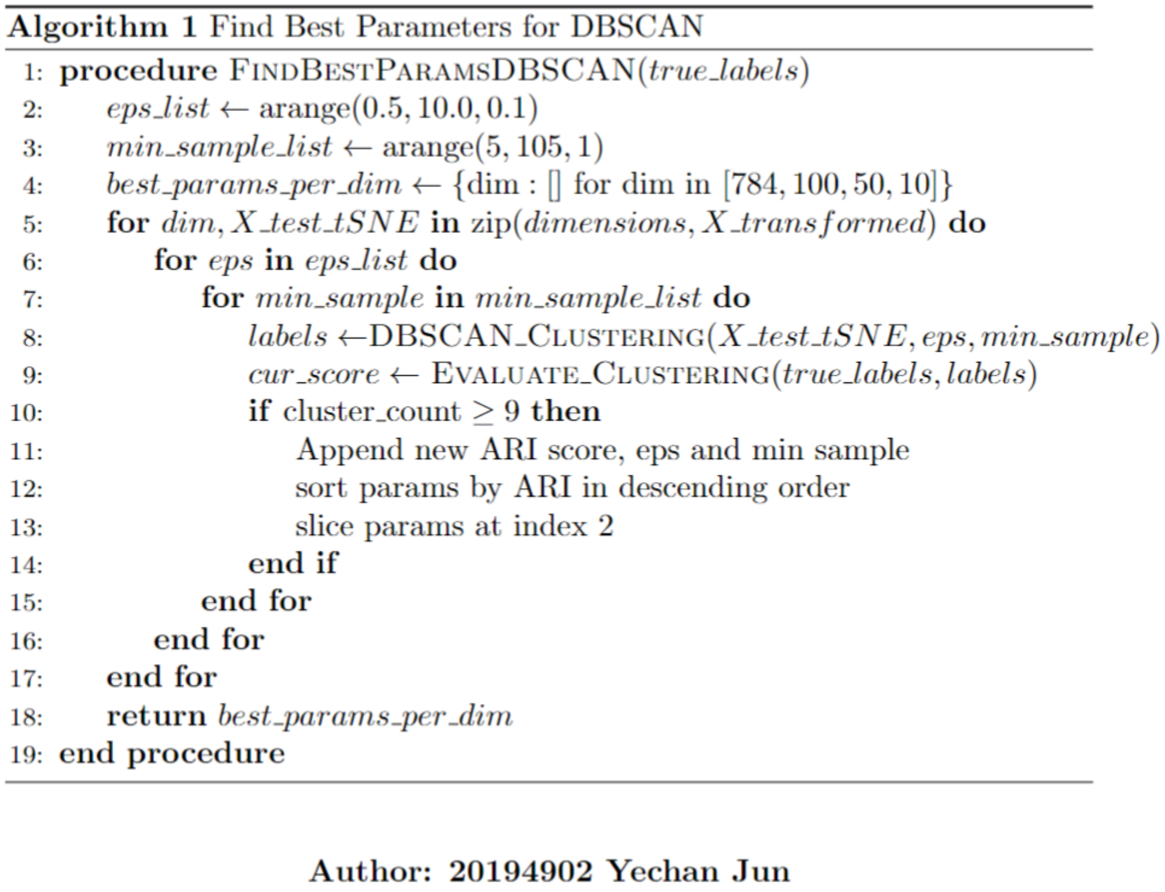


Algorithm 3 Transform PCA to t-SNE

```
procedure Transform PCA to t-SNE($X$)
    $n\_components \leftarrow 2$
    $learning\_rate \leftarrow 300$
    $perplexity \leftarrow 30$
    $early\_exaggeration \leftarrow 12$
    $init \leftarrow' random'$
    $tSNE \leftarrow$ Create tSNE with parameters
    $test\_PCA \leftarrow$ DataFrame($X$)
    $X\_test\_tSNE \leftarrow tSNE$.fit and transform PCA to tSNE
    $X\_test\_tSNE \leftarrow$ convert tSNE to Dataframe($X\_test\_tSNE$)
    return $X\_test\_tSNE$
end procedure
```

Author: 20194902 Yechan Jun

[Hyper parameter search for DBSCAN]



Algorithm 1 Find Best Parameters for DBSCAN

```
procedure FindBestParamsDBSCAN($true\_labels$)
    $eps\_list \leftarrow$ arange(0.5, 10.0, 0.1)
    $min\_sample\_list \leftarrow$ arange(5, 105, 1)
    $best\_params\_per\_dim \leftarrow$ {dim : [] for dim in [784, 100, 50, 10]}
    for $dim, X\_test\_tSNE$ in zip($dimensions, X\_transformed$) do
        for $eps$ in $eps\_list$ do
            for $min\_sample$ in $min\_sample\_list$ do
                $labels \leftarrow$ DBSCAN_Clustering($X\_test\_tSNE, eps, min\_sample$)
                $cur\_score \leftarrow$ Evaluate_Clustering($true\_labels, labels$)
                if cluster_count $\geq 9$ then
                    Append new ARI score, eps and min sample
                    sort params by ARI in descending order
                    slice params at index 2
                end if
            end for
        end for
    end for
    return $best\_params\_per\_dim$
end procedure
```

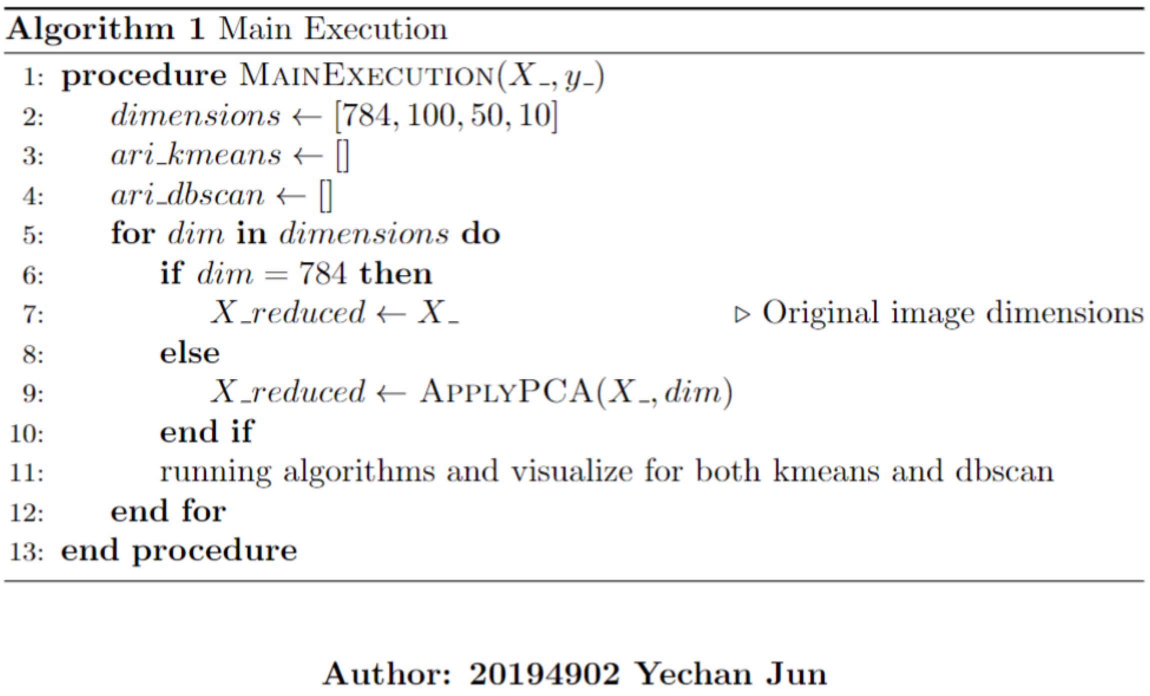
Author: 20194902 Yechan Jun

3. 실험 진행

아래의 pseudo code를 따라,

- 각 차원 784, 100, 54, 10에 대해
    - PCA 적용
    - tSNE 변형
    - 변형된 데이터셋에 대해 Kmeans 와 dbscan 클러스터링 알고리즘으로 학습후, ARI score계산 및 시각 (dbscan은 parameter search로 얻은 최적 params중 임의로 선택하여 통일 적용)

a. learning process



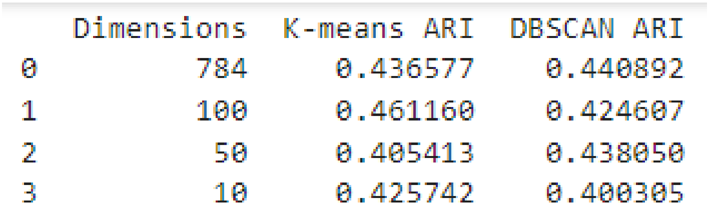
Algorithm 1 Main Execution

```
procedure MainExecution($X\_, y\_$)
    $dimensions \leftarrow [784, 100, 50, 10]$
    $ari\_kmeans \leftarrow []$
    $ari\_dbscan \leftarrow []$
    for $dim$ in $dimensions$ do
        if $dim = 784$ then
            $X\_reduced \leftarrow X\_$                ▷ Original image dimensions
        else
            $X\_reduced \leftarrow$ ApplyPCA($X\_, dim$)
        end if
        running algorithms and visualize for both kmeans and dbscan
    end for
end procedure
```

Author: 20194902 Yechan Jun

4. 실험 결과

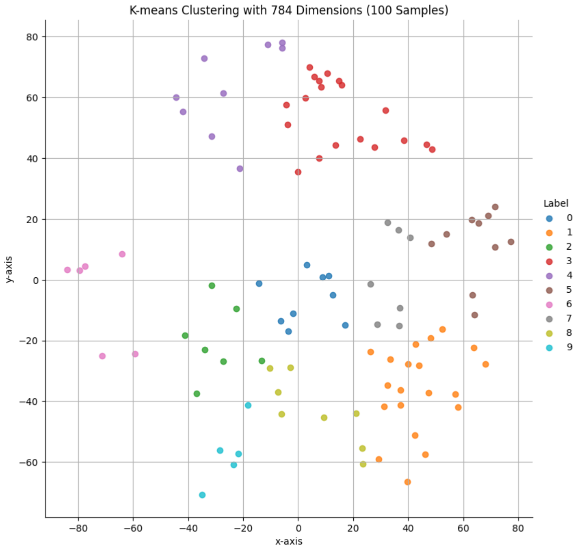
a. ARI scores for both of KMeans and DBSCAN

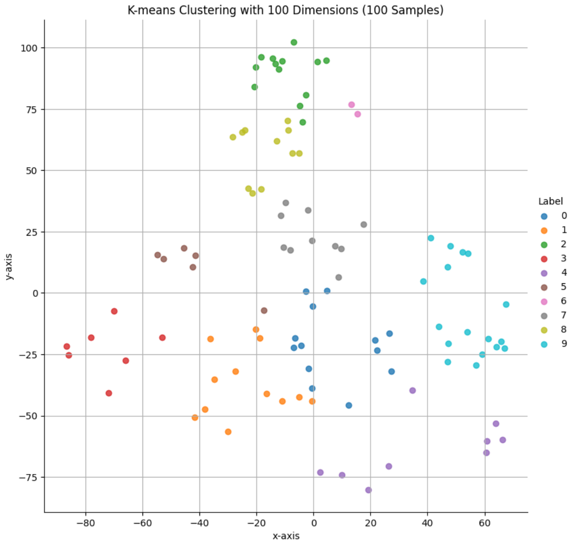


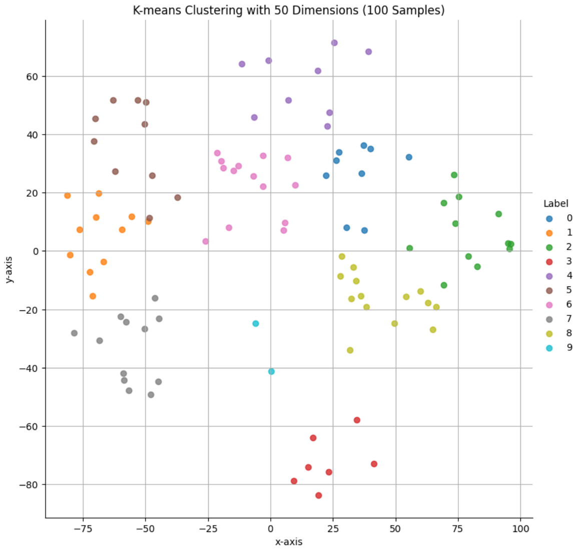
```
   Dimensions  K-means ARI  DBSCAN ARI
0         784     0.436577    0.440892
1         100     0.461160    0.424607
2          50     0.405413    0.438050
3          10     0.425742    0.400305
```

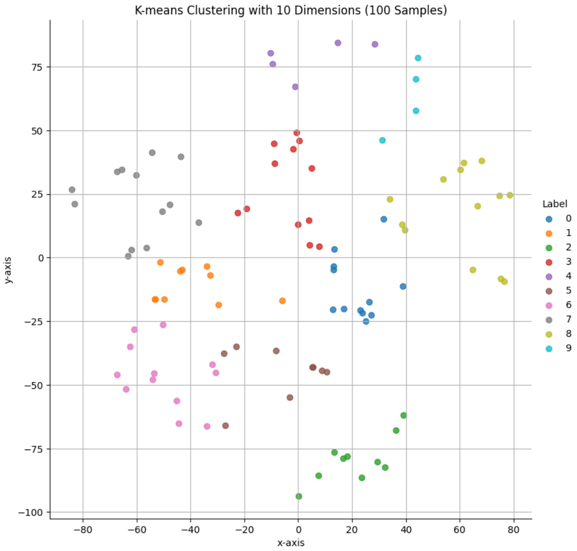
b. KMeans visualizatoin

Kmeans cluster 100 samples plotted

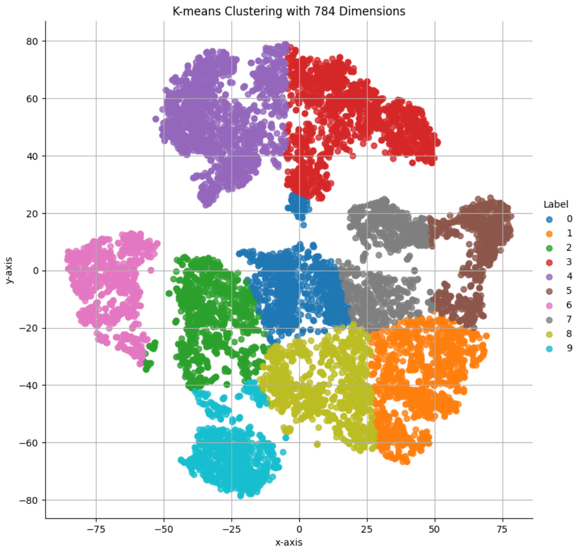


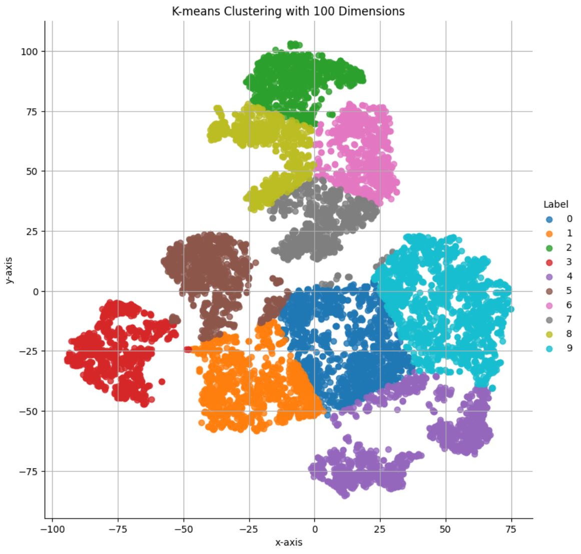


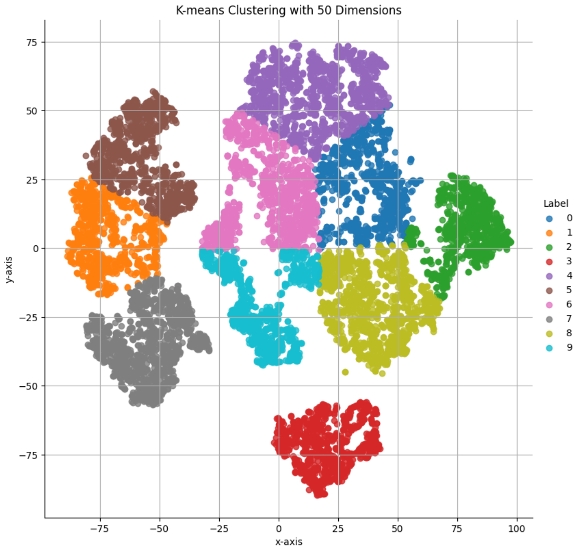


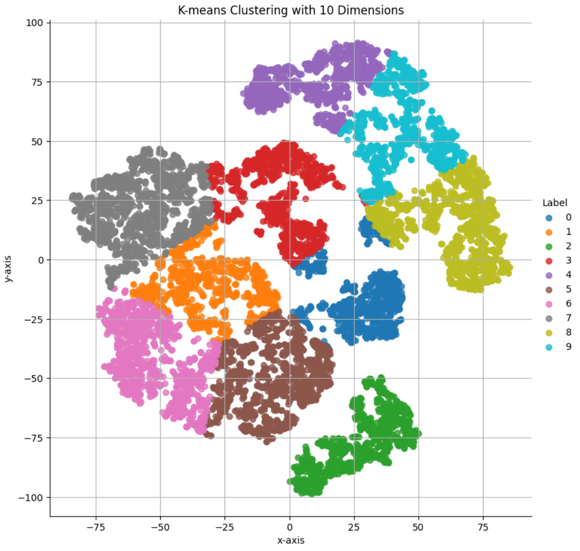


KMeans cluster Without sampling



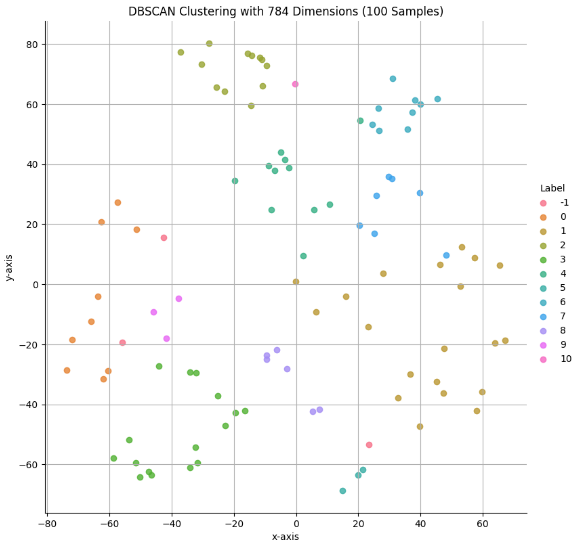


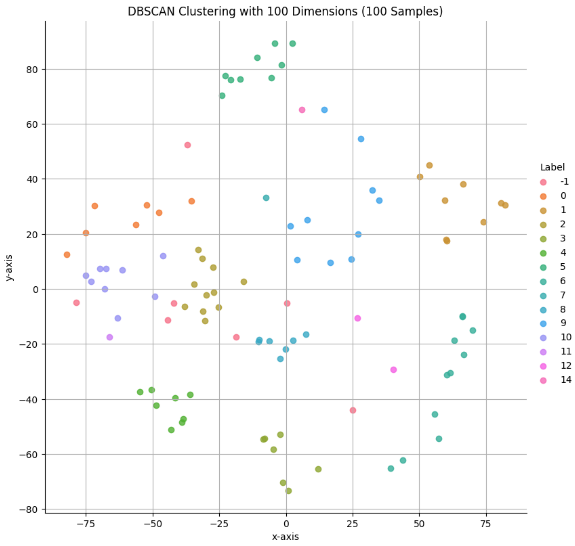


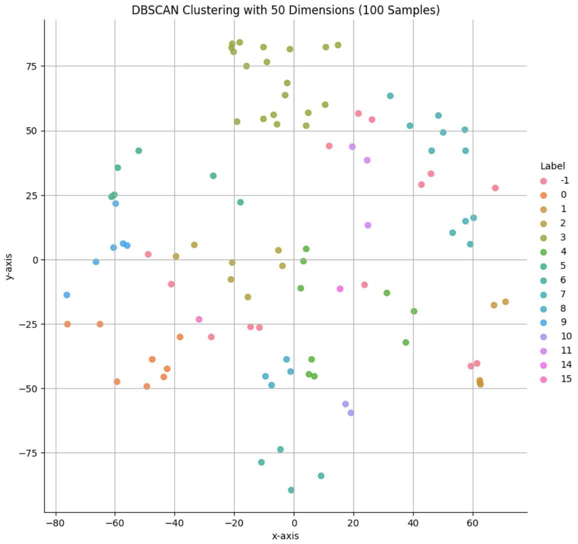


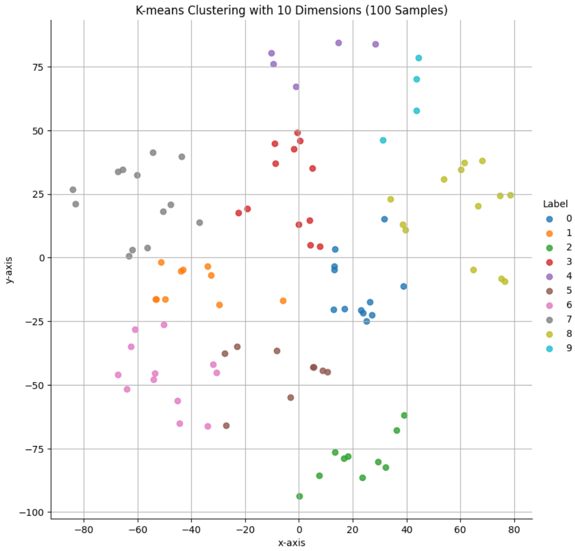
c. DBSCAN visualization (plotted 100 samples)

DBSCAN 100 samples plotted

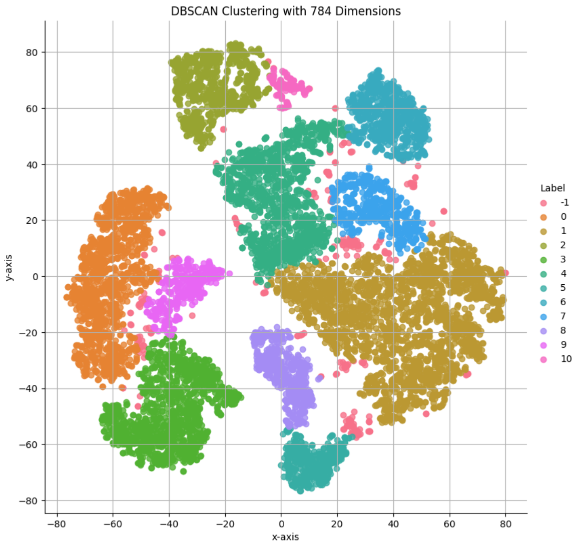




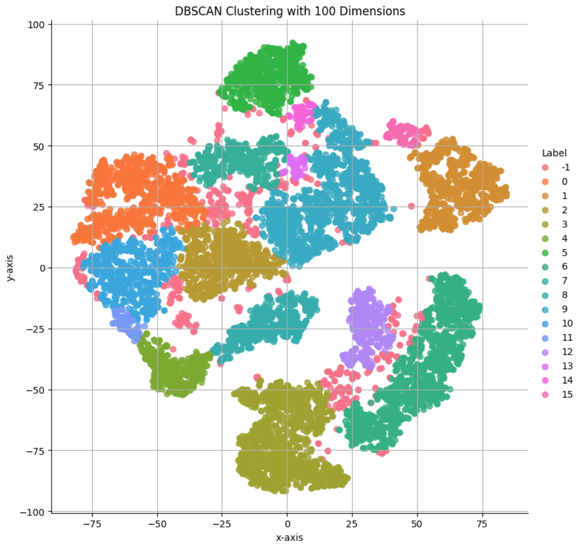




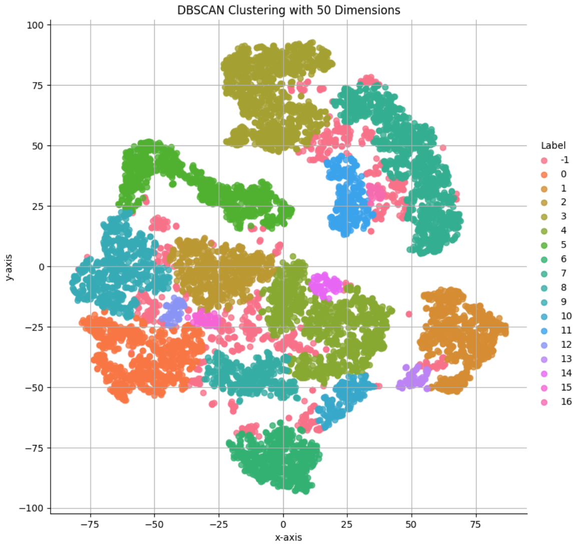
DBSCAN Without sampling



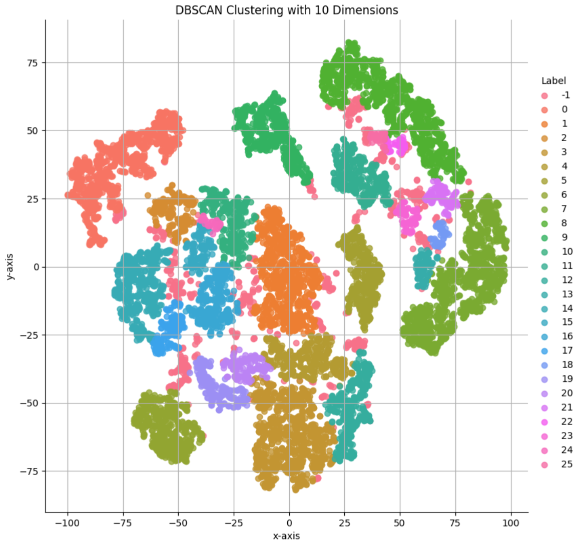
DBSCAN Clustering with 784 Dimensions



DBSCAN Clustering with 100 Dimensions



DBSCAN Clustering with 50 Dimensions



DBSCAN Clustering with 10 Dimensions