REPORT

AIM: To identify a dimension reduction technique using various datasets with the intention to identify the most optimal technique that could efficiently embed the high dimensional points in low dimensional space. It also involves developing a new algorithm which can perform equivalently or better than TSNE.

APPROACH 1: USING LSH FOREST WITH STACKED DENOSING AUTOENCODER AND RBM.

To evaluate whether using LSH Forest enhances/degrades the quality of the reconstructed input in comparison to original input. Also, to plot the Bottleneck Neurons of Denoising Autoencoder and RBM and identify whether it correctly generates clusters of different labels of Digits if hash codes are used as input.

In this implementation, Mnist, Iris and Bouston Housing datasets from UCI machine learning repository is taken as input data.

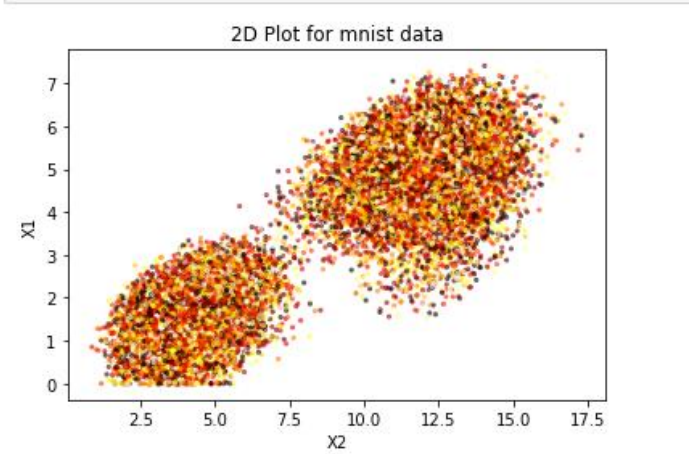
Steps for Implementation:

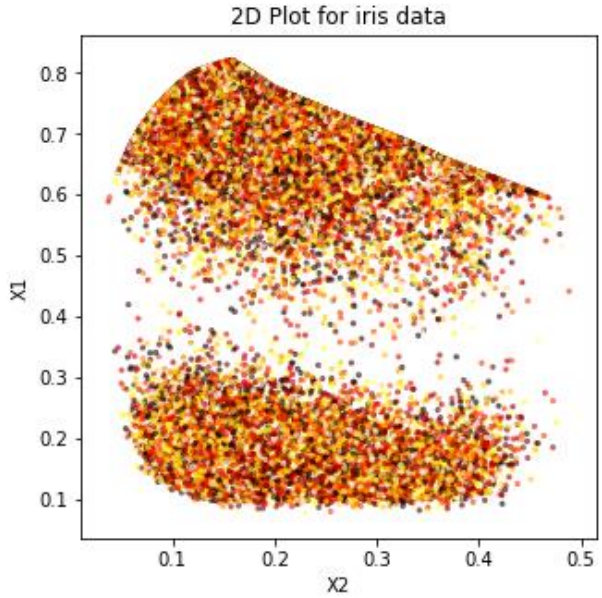
This implementation can be described in the following steps -:

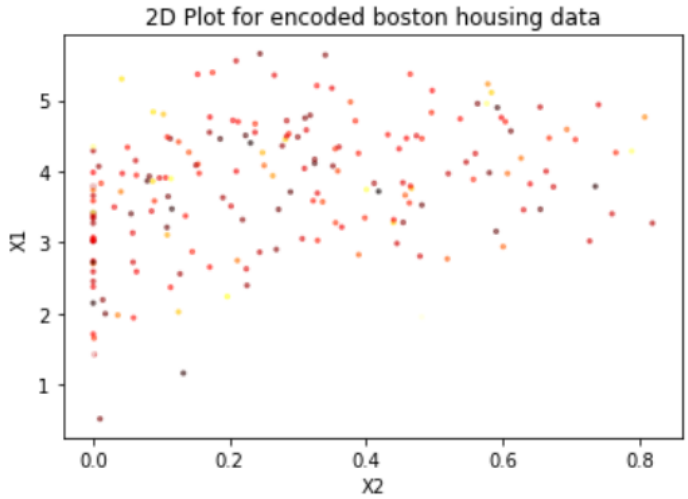
1. Preparing input data in the form of Dataframe from the csv file of training data.
2. Training LSH Forest with n_estimator=1.
3. Extracting Binary Hash Codes from the LSH Forest model.
4. Creating Stacked Denoising Autoencoder using keras.
5. Feeding the obtained Binary Hash Codes as input to the autoencoder.
6. Training the model of created autoencoder on the input.
7. Plotting the bottleneck neurons (encoded and decoded hidden layers).
8. Creating Restricted Botlzmann Machine using Tensorflow.
9. Feeding the obtained Binary Hash Codes as input to the RBM.
10. Training RBM on the input.
11. Plotting the Bottleneck Neurons.

Results:

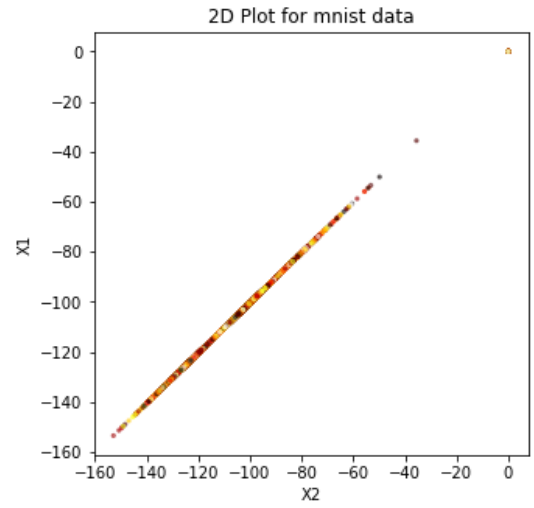
1. With LSH+ Stacked Denoising Autoencoder:

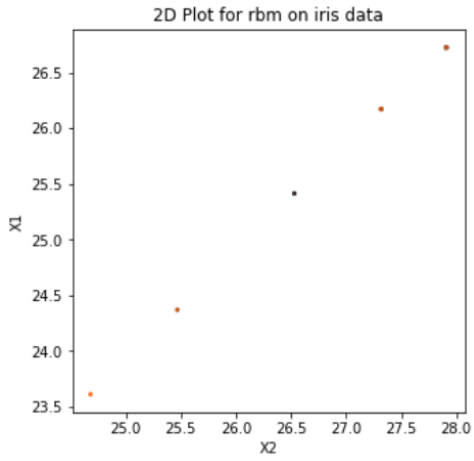


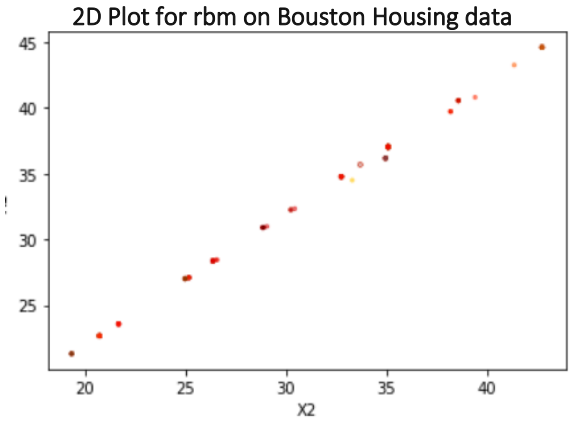




2. With LSH+ RBM:







APPROACH 2: USING LAPLACIAN EIGENMAPS

In this implementation, Gene Expression Cancer RNA Sequence Dataset, Iris and Bouston Housing datasets from UCI machine learning repository is taken as input data.

Steps for Implementation:

This implementation can be described in the following steps -:

1. Preparing input data in the form of Dataframe from the csv file of data.

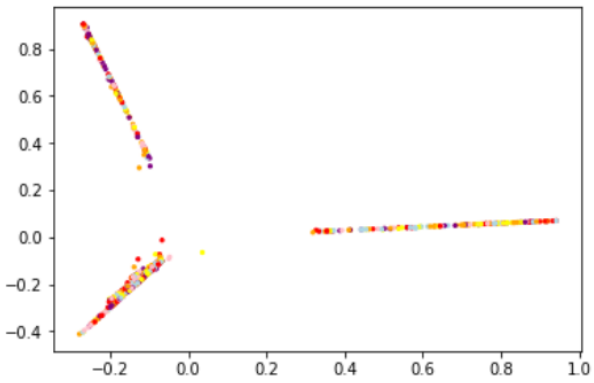
2 Preprocessing input data.

3. Applying Laplacian Eigenmaps in the form of Spectral Embedding using Sklearn Manifold.

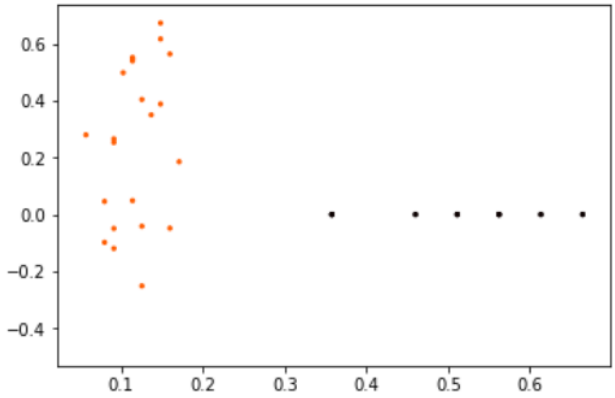
4. Generating plot of the results.

Results:

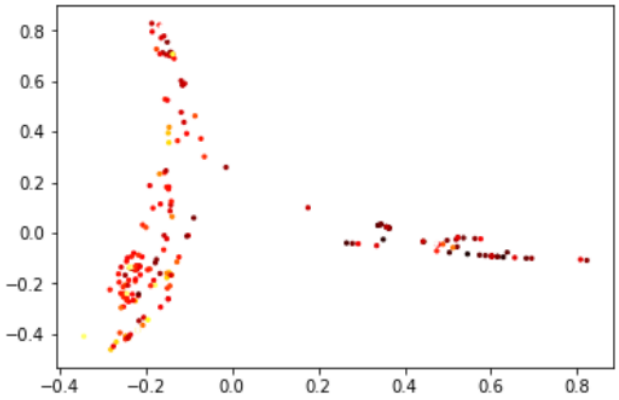
1. 2D Plot for Gene Expression Cancer RNA Sequence Dataset.



2. 2D Plot for IRIS Dataset



3. 2D Plot for Bouston Housing Dataset.



APPROACH 3: KMEANS ON INPUT DATA IN WHICH FEATURE SELECTION IS DONE THROUGH ZSCORE AND FDR

In this implementation, Mnist, Iris and Breast Cancer datasets from UCI machine learning repository is taken as input data.

Steps for Implementation:

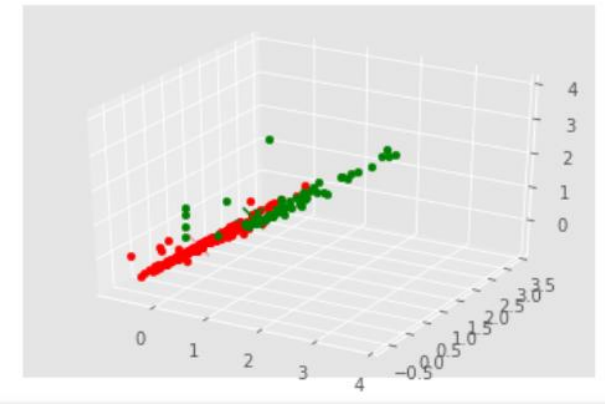
This implementation can be described in the following steps -:

1. Preparing input data in the form of Dataframe from the csv file of training data.
2. Preprocessing of Input Data.
3. Calculating zscore for each feature of each sample in the training set.
4. Consider only values of those features of a sample whose zscore > 0.05 while setting the zscore values < 0.05 to zero.
5. Computing kmeans on this newly created dataset from Step 4.
6. Visualization/Plotting of the clusters in 3D plot.

Results:

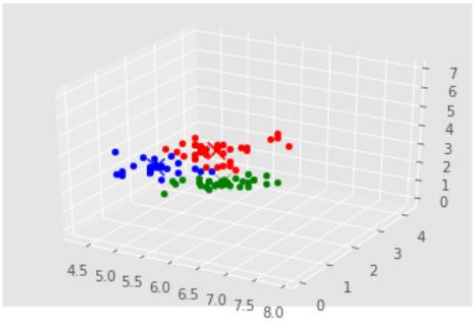
1. 3D Plot for Breast Cancer Dataset.

```
Cluster 0 contains 96 samples
Cluster 1 contains 245 samples
```



2. 3D Plot for Iris Dataset

```
Cluster 0 contains 31 samples
Cluster 1 contains 36 samples
Cluster 2 contains 23 samples
```



3. Clusters of Mnist Dataset with kmeans+TSNE.

```
Cluster 0 contains 3655 samples
Cluster 1 contains 1889 samples
Cluster 2 contains 3162 samples
Cluster 3 contains 2409 samples
Cluster 4 contains 1310 samples
Cluster 5 contains 3888 samples
Cluster 6 contains 2378 samples
Cluster 7 contains 2378 samples
Cluster 8 contains 2785 samples
Cluster 9 contains 1346 samples
```