CMPS 290 C W-18 Mini Project

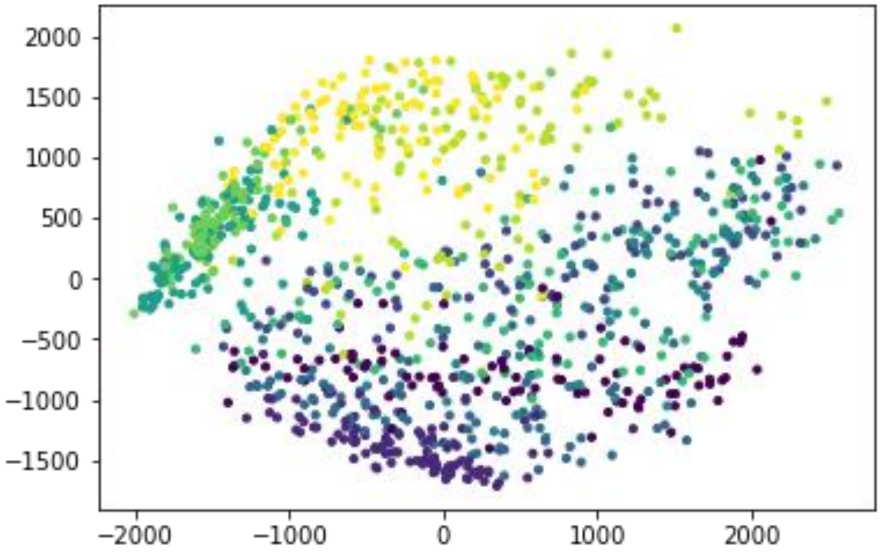
Arghyadeep Giri
Student ID:1566630 argiri@ucsc.edu

- **Dataset Used:** Fashion dataset from Kaggle which consists of 70000 images of different fashion apparels. There are 10 classes and labelled from 0-9. The images are 28*28 in dimension.

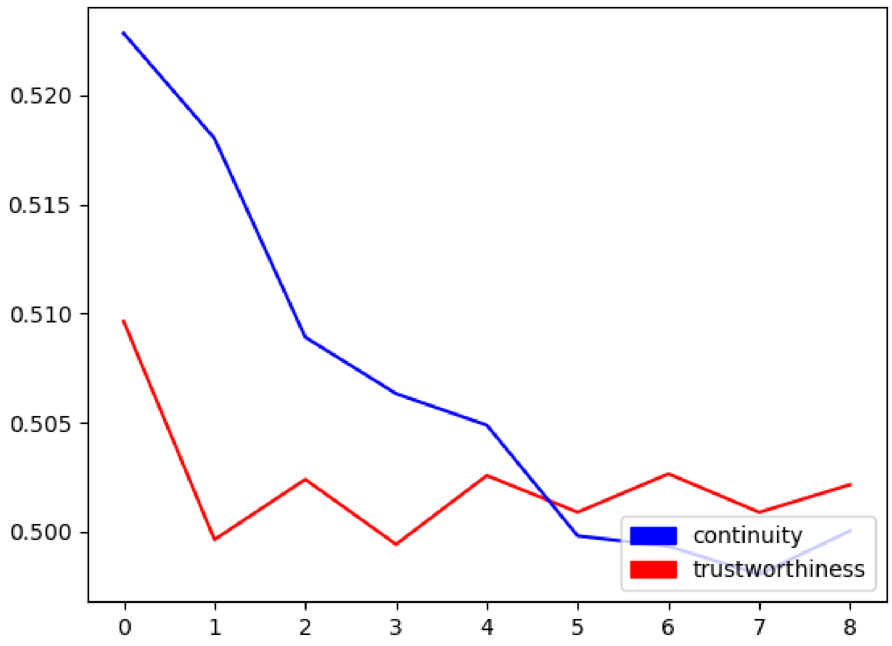
- **Methods Implemented**:

PCA

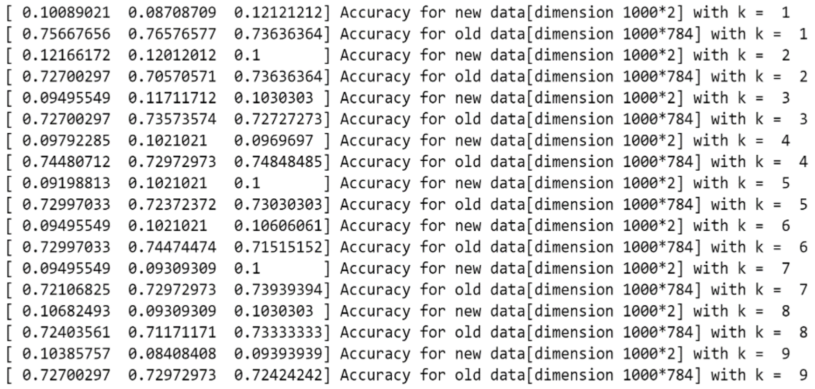
Used the Scikit-learn PCA library to implement. Dataset was shuffled, and 1000 points were used to reduce dimension to 2 components. The results are as follows:



Trustworthy Continuity results:



Nearest neighbor accuracy results (since data is labelled):

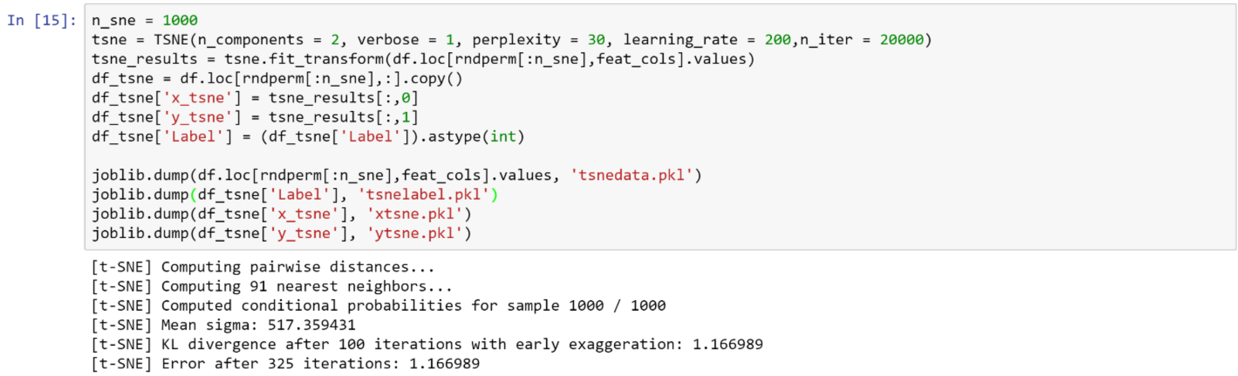


```
[ 0.10089021  0.08708709  0.12121212] Accuracy for new data[dimension 1000*2] with k =  1
[ 0.75667656  0.76576577  0.73636364] Accuracy for old data[dimension 1000*784] with k =  1
[ 0.12166172  0.12012012  0.1       ] Accuracy for new data[dimension 1000*2] with k =  2
[ 0.72700297  0.70570571  0.73636364] Accuracy for old data[dimension 1000*784] with k =  2
[ 0.09495549  0.11711712  0.1030303 ] Accuracy for new data[dimension 1000*2] with k =  3
[ 0.72700297  0.73573574  0.72727273] Accuracy for old data[dimension 1000*784] with k =  3
[ 0.09792285  0.1021021   0.0969697 ] Accuracy for new data[dimension 1000*2] with k =  4
[ 0.74480712  0.72972973  0.74848485] Accuracy for old data[dimension 1000*784] with k =  4
[ 0.09198813  0.1021021   0.1       ] Accuracy for new data[dimension 1000*2] with k =  5
[ 0.72997033  0.72372372  0.73030303] Accuracy for old data[dimension 1000*784] with k =  5
[ 0.09495549  0.1021021   0.10606061] Accuracy for new data[dimension 1000*2] with k =  6
[ 0.72997033  0.74474474  0.71515152] Accuracy for old data[dimension 1000*784] with k =  6
[ 0.09495549  0.09309309  0.1       ] Accuracy for new data[dimension 1000*2] with k =  7
[ 0.72106825  0.72972973  0.73939394] Accuracy for old data[dimension 1000*784] with k =  7
[ 0.10682493  0.09309309  0.1030303 ] Accuracy for new data[dimension 1000*2] with k =  8
[ 0.72403561  0.71171171  0.73333333] Accuracy for old data[dimension 1000*784] with k =  8
[ 0.10385757  0.08408408  0.09393939] Accuracy for new data[dimension 1000*2] with k =  9
[ 0.72700297  0.72972973  0.72424242] Accuracy for old data[dimension 1000*784] with k =  9
```

t-SNE

Used the Scikit-learn tsne library to implement. Dataset was shuffled, and 1000 points were used to reduce dimension to 2 components.

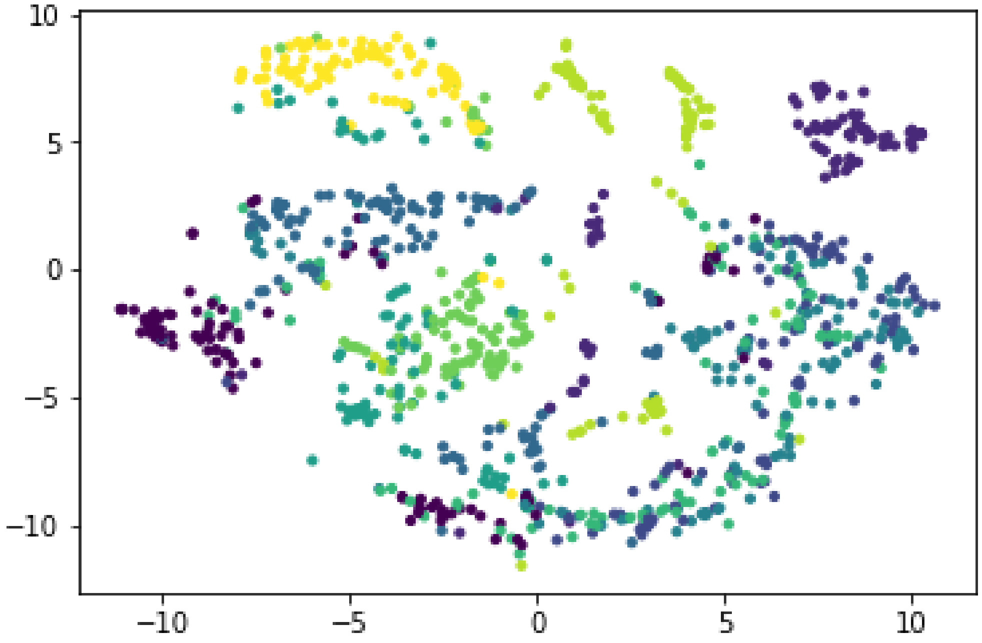
TSNE hyperparameters used:
n_components = 2, verbose = 1, perplexity = 30, learning_rate = 200, n_iter = 20000
The results are as follows:



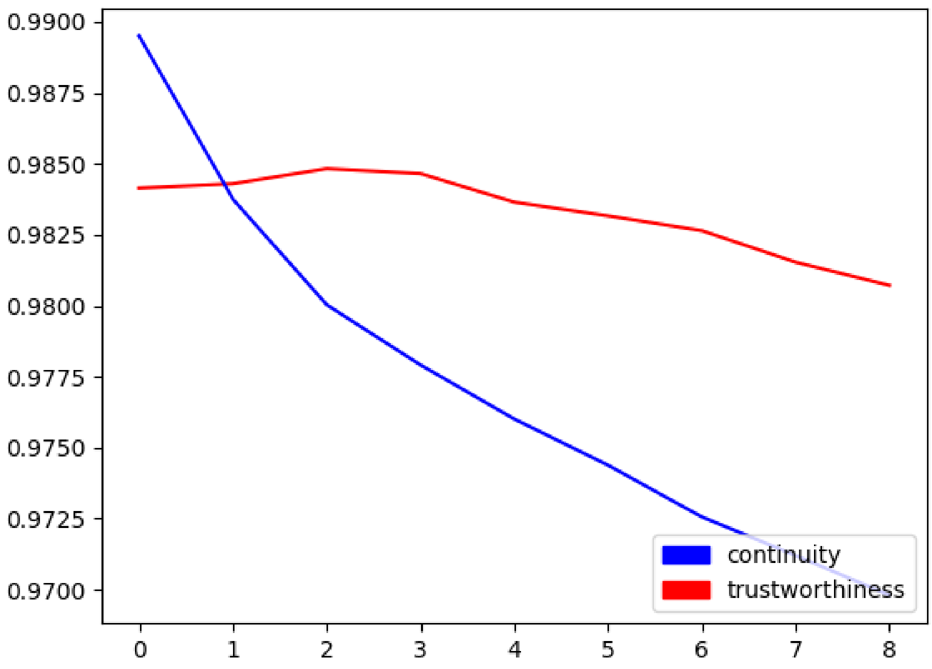
```
In [15]: n_sne = 1000
         tsne = TSNE(n_components = 2, verbose = 1, perplexity = 30, learning_rate = 200,n_iter = 20000)
         tsne_results = tsne.fit_transform(df.loc[rndperm[:n_sne],feat_cols].values)
         df_tsne = df.loc[rndperm[:n_sne],:].copy()
         df_tsne['x_tsne'] = tsne_results[:,0]
         df_tsne['y_tsne'] = tsne_results[:,1]
         df_tsne['Label'] = (df_tsne['Label']).astype(int)

         joblib.dump(df.loc[rndperm[:n_sne],feat_cols].values, 'tsnedata.pkl')
         joblib.dump(df_tsne['Label'], 'tsnelabel.pkl')
         joblib.dump(df_tsne['x_tsne'], 'xtsne.pkl')
         joblib.dump(df_tsne['y_tsne'], 'ytsne.pkl')
```

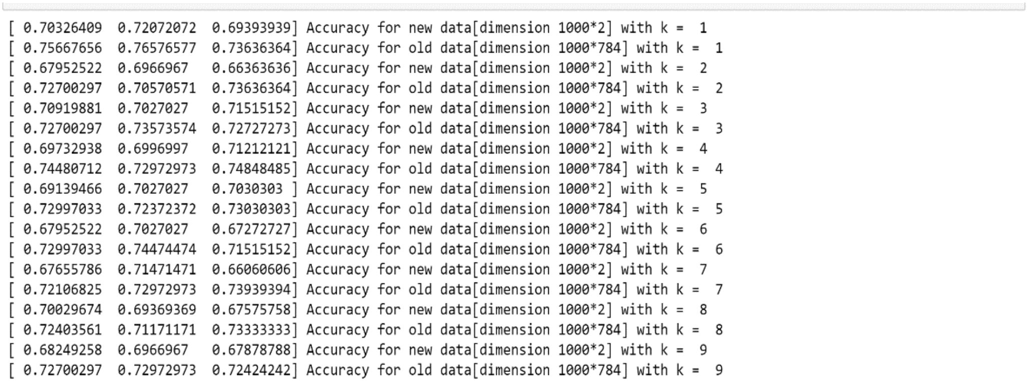
```
[t-SNE] Computing pairwise distances...
[t-SNE] Computing 91 nearest neighbors...
[t-SNE] Computed conditional probabilities for sample 1000 / 1000
[t-SNE] Mean sigma: 517.359431
[t-SNE] KL divergence after 100 iterations with early exaggeration: 1.166989
[t-SNE] Error after 325 iterations: 1.166989
```



Trustworthy Continuity results:



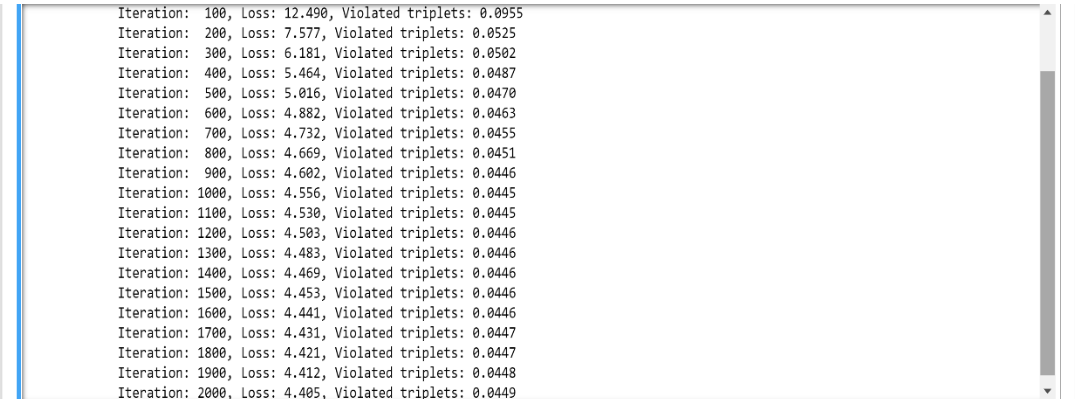
Nearest neighbor accuracy results (since data is labelled):


```
[ 0.70326409  0.72072072  0.69393939] Accuracy for new data[dimension 1000*2] with k =  1
[ 0.75667656  0.76576577  0.73636364] Accuracy for old data[dimension 1000*784] with k =  1
[ 0.67952522  0.6966967   0.66363636] Accuracy for new data[dimension 1000*2] with k =  2
[ 0.72700297  0.70570571  0.73636364] Accuracy for old data[dimension 1000*784] with k =  2
[ 0.70919881  0.7027027   0.71515152] Accuracy for new data[dimension 1000*2] with k =  3
[ 0.72700297  0.73573574  0.72727273] Accuracy for old data[dimension 1000*784] with k =  3
[ 0.69732938  0.6996997   0.71212121] Accuracy for new data[dimension 1000*2] with k =  4
[ 0.74480712  0.72972973  0.74848485] Accuracy for old data[dimension 1000*784] with k =  4
[ 0.69139466  0.7027027   0.7030303 ] Accuracy for new data[dimension 1000*2] with k =  5
[ 0.72997033  0.72372372  0.73030303] Accuracy for old data[dimension 1000*784] with k =  5
[ 0.67952522  0.7027027   0.67272727] Accuracy for new data[dimension 1000*2] with k =  6
[ 0.72997033  0.74474474  0.71515152] Accuracy for old data[dimension 1000*784] with k =  6
[ 0.67655786  0.71471471  0.66060606] Accuracy for new data[dimension 1000*2] with k =  7
[ 0.72106825  0.72972973  0.73939394] Accuracy for old data[dimension 1000*784] with k =  7
[ 0.70029674  0.69369369  0.67575758] Accuracy for new data[dimension 1000*2] with k =  8
[ 0.72403561  0.71171171  0.73333333] Accuracy for old data[dimension 1000*784] with k =  8
[ 0.68249258  0.6966967   0.67878788] Accuracy for new data[dimension 1000*2] with k =  9
[ 0.72700297  0.72972973  0.72424242] Accuracy for old data[dimension 1000*784] with k =  9
```

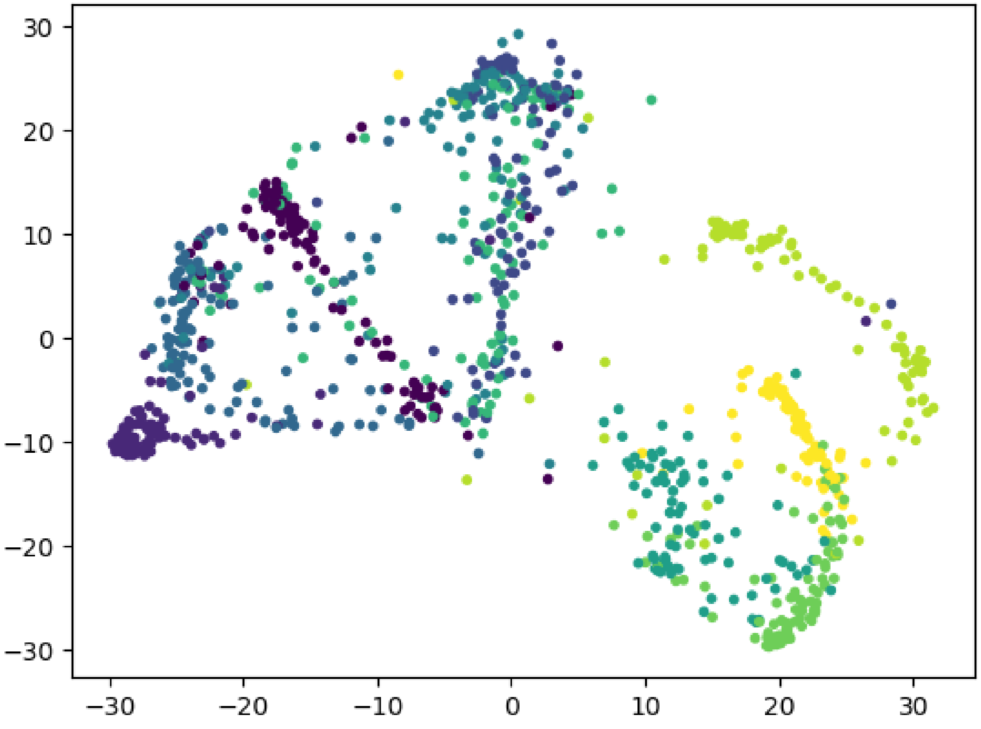
TriMap

Modified the code provided with miniproject pdf to find results. Data used was shuffled 1000 points from the original dataset.

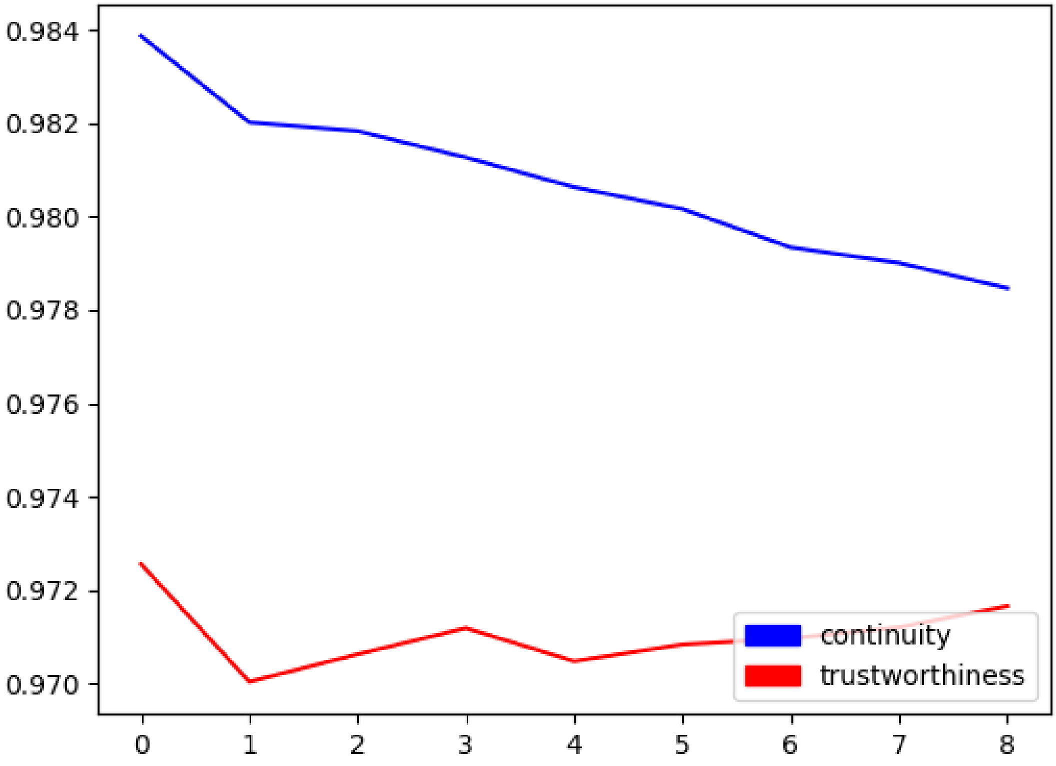
The results are as follows:



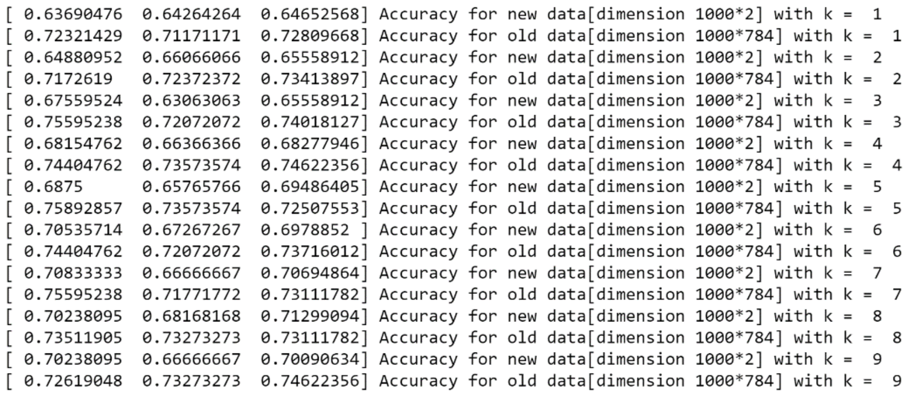
```
Iteration:  100, Loss: 12.490, Violated triplets: 0.0955
Iteration:  200, Loss: 7.577, Violated triplets: 0.0525
Iteration:  300, Loss: 6.181, Violated triplets: 0.0502
Iteration:  400, Loss: 5.464, Violated triplets: 0.0487
Iteration:  500, Loss: 5.016, Violated triplets: 0.0470
Iteration:  600, Loss: 4.882, Violated triplets: 0.0463
Iteration:  700, Loss: 4.732, Violated triplets: 0.0455
Iteration:  800, Loss: 4.669, Violated triplets: 0.0451
Iteration:  900, Loss: 4.602, Violated triplets: 0.0446
Iteration: 1000, Loss: 4.556, Violated triplets: 0.0445
Iteration: 1100, Loss: 4.530, Violated triplets: 0.0445
Iteration: 1200, Loss: 4.503, Violated triplets: 0.0446
Iteration: 1300, Loss: 4.483, Violated triplets: 0.0446
Iteration: 1400, Loss: 4.469, Violated triplets: 0.0446
Iteration: 1500, Loss: 4.453, Violated triplets: 0.0446
Iteration: 1600, Loss: 4.441, Violated triplets: 0.0446
Iteration: 1700, Loss: 4.431, Violated triplets: 0.0447
Iteration: 1800, Loss: 4.421, Violated triplets: 0.0447
Iteration: 1900, Loss: 4.412, Violated triplets: 0.0448
Iteration: 2000, Loss: 4.405, Violated triplets: 0.0449
```



Trustworthy Continuity results:



Nearest neighbor accuracy results (since data is labelled):



```
[ 0.63690476  0.64264264  0.64652568] Accuracy for new data[dimension 1000*2] with k =  1
[ 0.72321429  0.71171171  0.72809668] Accuracy for old data[dimension 1000*784] with k =  1
[ 0.64880952  0.66066066  0.65558912] Accuracy for new data[dimension 1000*2] with k =  2
[ 0.7172619   0.72372372  0.73413897] Accuracy for old data[dimension 1000*784] with k =  2
[ 0.67559524  0.63063063  0.65558912] Accuracy for new data[dimension 1000*2] with k =  3
[ 0.75595238  0.72072072  0.74018127] Accuracy for old data[dimension 1000*784] with k =  3
[ 0.68154762  0.66366366  0.68277946] Accuracy for new data[dimension 1000*2] with k =  4
[ 0.74404762  0.73573574  0.74622356] Accuracy for old data[dimension 1000*784] with k =  4
[ 0.6875      0.65765766  0.69486405] Accuracy for new data[dimension 1000*2] with k =  5
[ 0.75892857  0.73573574  0.72507553] Accuracy for old data[dimension 1000*784] with k =  5
[ 0.70535714  0.67267267  0.6978852 ] Accuracy for new data[dimension 1000*2] with k =  6
[ 0.74404762  0.72072072  0.73716012] Accuracy for old data[dimension 1000*784] with k =  6
[ 0.70833333  0.66666667  0.70694864] Accuracy for new data[dimension 1000*2] with k =  7
[ 0.75595238  0.71771772  0.73111782] Accuracy for old data[dimension 1000*784] with k =  7
[ 0.70238095  0.68168168  0.71299094] Accuracy for new data[dimension 1000*2] with k =  8
[ 0.73511905  0.73273273  0.73111782] Accuracy for old data[dimension 1000*784] with k =  8
[ 0.70238095  0.66666667  0.70090634] Accuracy for new data[dimension 1000*2] with k =  9
[ 0.72619048  0.73273273  0.74622356] Accuracy for old data[dimension 1000*784] with k =  9
```

Conclusion:

The tests were done with 1000 labelled data points and probably more data would have generated better accuracy. The results are nevertheless self-evident with PCA with the worst performance amongst the dimension reduction techniques used in this project. According to performance measures both t-SNE and trimap performed well but t-SNE performed better with the provided dataset by a slight margin. For any of the accuracy measures the values get lower with increasing values of k. Also continuity values are most of the times higher compared to that of trustworthiness values provided any value of k. The data point representations are colored according to the 10 classes in the dataset.