University of Waterloo
ECE 657A: Data and Knowledge Modeling and Analysis
Winter 2019
Assignment 1: Data Cleaning and Dimensionality Reduction
Due: February 16th, 2019 11:59pm

Group 36:

Abul Hasan Fahad (20764979)

Syed Anjoom Iqbal (20640625)

Mohamed Abdalla (20800994)

I. Data Cleaning and Preprocessing

Code: “Assignment 1 Q1.ipynb”

After loading ‘Data A’, the existence of missing values, empty features, empty samples, and outliers have been detected (we checked for duplicates but none was found).

To find the outliers the Tukey outlier labeling is applied (using box plot mathematically). If a value lies below the first quartile or above the third quartile by more than 1.5 times the interquartile range, it will be considered an outlier. Next, outliers have been treated as a miss measurement from the specific sensor and therefore changed into missing values.

After that, features and samples with more the 50% missing values have been dropped (7% of the data) as filling them could distort our data and the analysis to be applied on it. Finally, as our data is a time series data set, the remaining missing values have been interpolated to avoid introducing unreal spicks to our time series signals.

As required, the Min-Max and Z-Score Normalizations have applied.

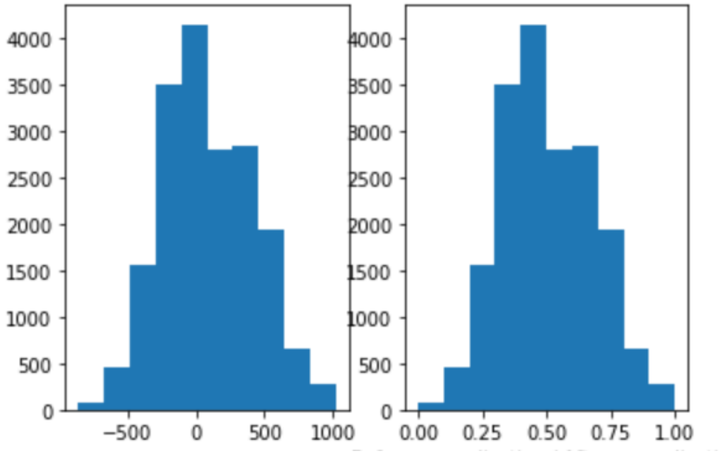


Figure 1 Feature 9 before and after Min-Max normalization

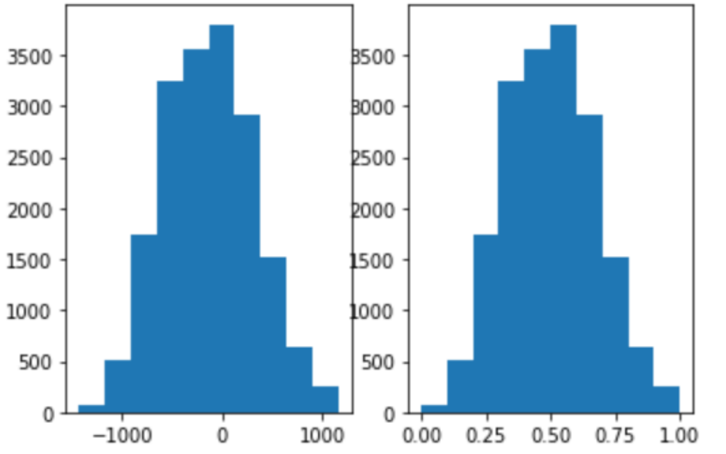


Figure 2 Feature 24 before and after Min-Max normalization

As shown in the Figure 1 and Figure 2, Min-Max normalization normalized the features between 0 and 1. This could be beneficial when different features have different scales and we want them to be treated the same in our machine learning algorithm. However, this normalization has some cons: it can encounter an out-of-bounds error if a future input falls outside the data range (true population max and min should be known to avoid this); it makes the samples that are far in space closer to each other and therefore difficult to separate; and normalization is a bad choice when small differences matters.

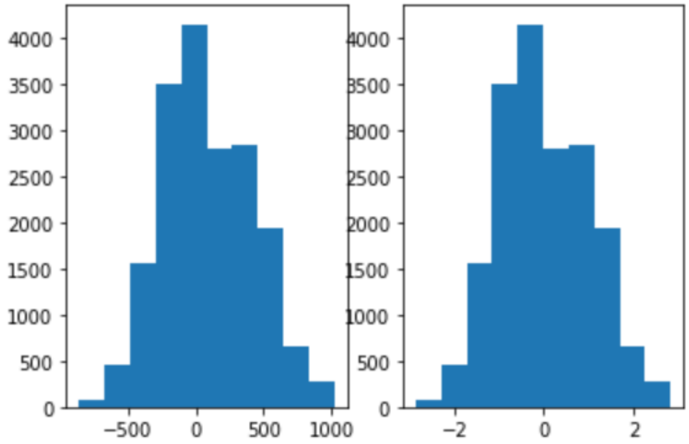


Figure 3 Feature 9 before and after Z-Score normalization

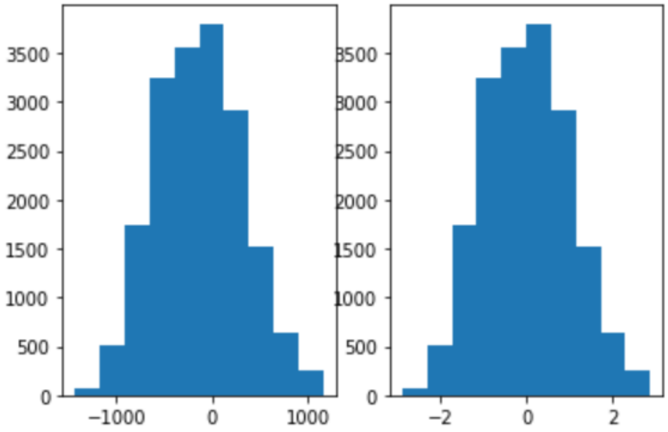


Figure 4 Feature 24 before and after Z-Score normalization

As shown in the Figure 3 and Figure 4, Z-Score normalization normalized the features around their means (with standard deviation=1) where the mean= 0 and samples above the mean are positive and the samples below the mean are negative. This gives your features the aggregate properties of standard normal distribution (mean=0, std=1). However, this normalization has the same cons as Min-Max normalization except that it is useful when the true min and max of an attribute are unknown.

II. Feature Extraction

Code: “Ass1 Q2.ipynb”

First, the eigenvectors and eigenvalues has been computed as required. Next, the Principle Component Analysis (PCA) dimensionality reduction technique has been applied.

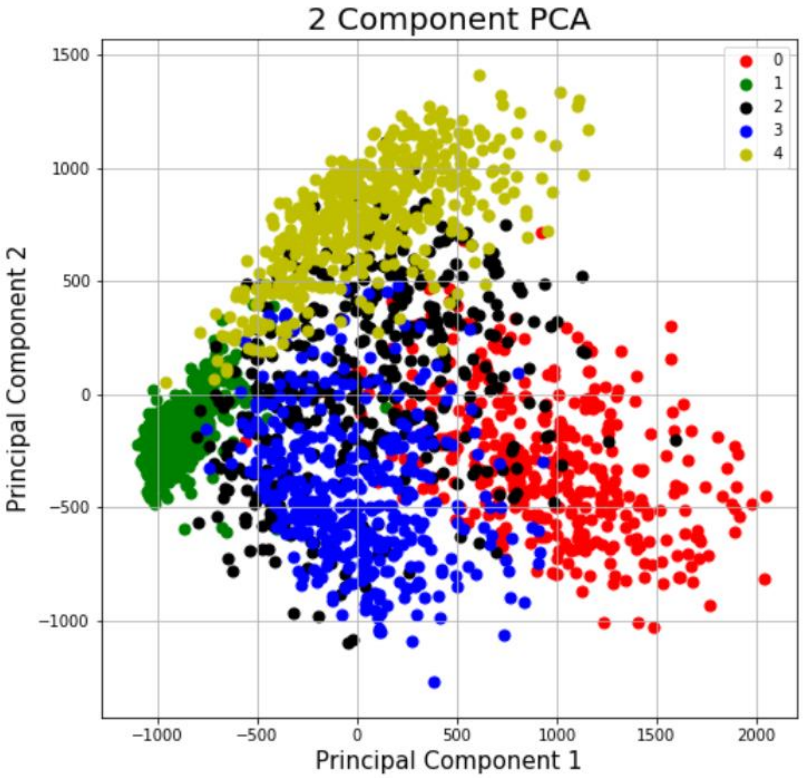


Figure 5 PCA reduced data (1st and 2nd components)

Although the classes overlap they are separated to some extent. The variance retained by using the 1st and 2nd Principal Components is 22% of the total variance in the original data set.

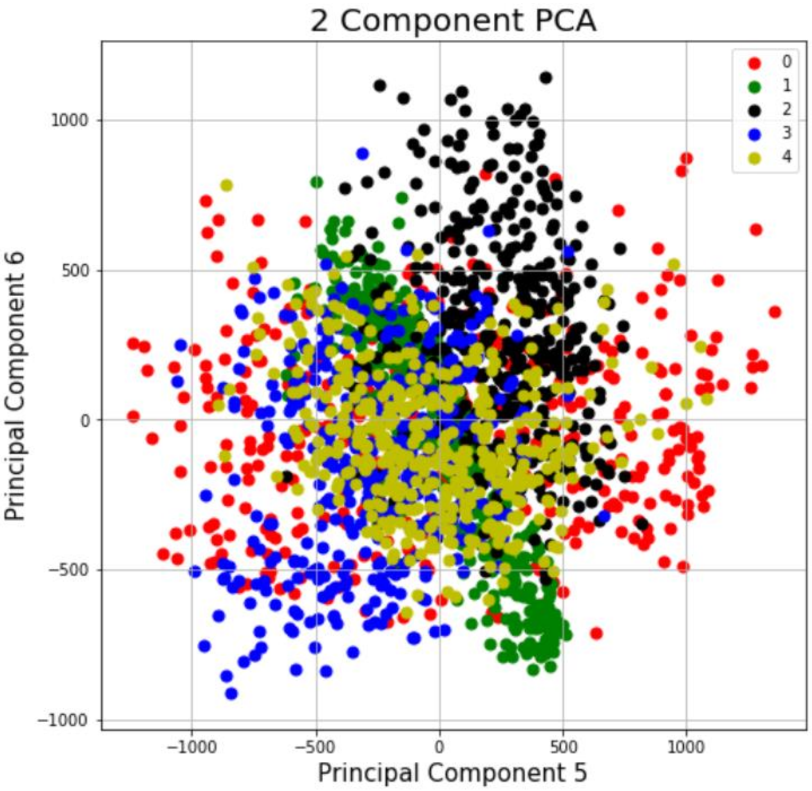


Figure 6 PCA reduced data (5th and 6th components)

Using the 5th and 6th Principal Components to plot the data, one can realize that the variance decreased in comparison to 1st and 2nd Principal Components. In addition, the classes' data points overlapped even more and they are not well separated. The variance retained by using the 5th and 6th Principal Components is 8.9% of the total variance in the original data set.

The Naïve Bayes classifier has been applied to classify 8 sets of PCA reduced data and the classification error has been plot against the retained variance using ‘For loop’.

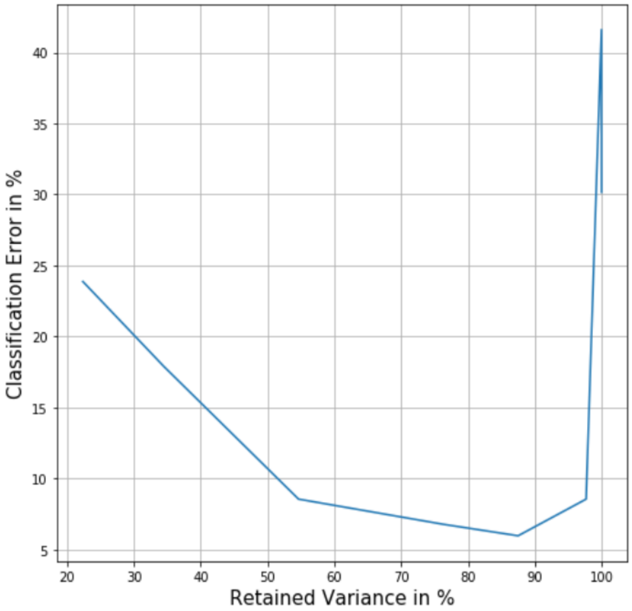


Figure 7 Naive Bayes classifier classification error verses the retained variance for different number of Principle Componets of PCA reduced data.

The graph (Figure 7) shows that the lowest classification error occurs when using 10 to 200 Principal Components with a retained variance of 54.6% and 97.7%. Using more Principal Components resulted in an increase in the error which could be the result of having too many features for our small number of samples. As a common practice, the number of samples should be ten times greater than the number of features (so having up to 200 features for our 2066 samples can be tolerated (classification error of 8.5%).

Finally, the Linear Discriminant Analysis (LDA) has been applied to reduce the dimensionality of our data.

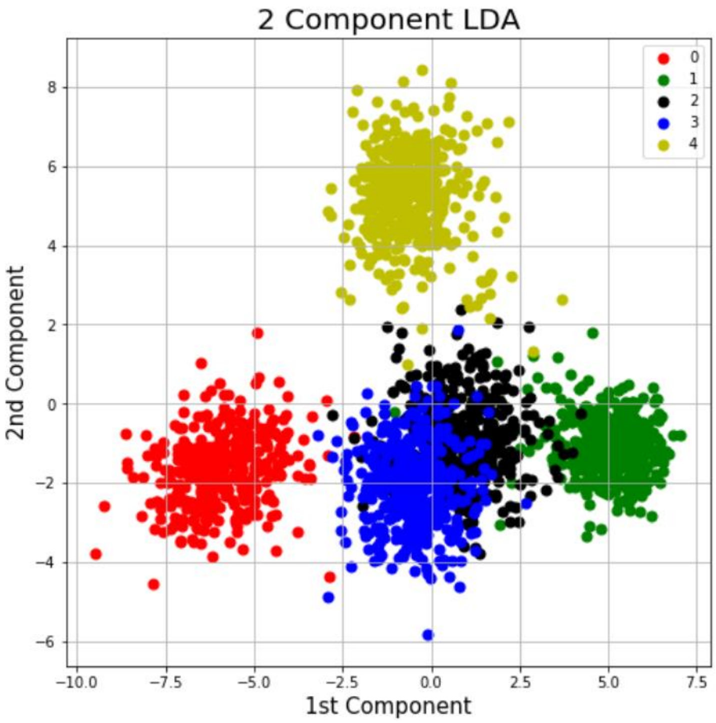


Figure 8 LDA reduced data (1st and 2nd components)

In comparison with PCA, when using LDA the classes separation is significantly apparent and the overlapping between classes has been minimized.

III. Nonlinear Dimensionality Reduction

1. Answer:
   Code: “asgnmnt1_3_1_2.py”
   As instructed, LLE dimension reduction algorithm was applied to the images of digit '3' only. After running the code multiple times, the Digit-3 instances are found to be distributed in the following two fashions over the first two components’ 2D space (Figure 9).

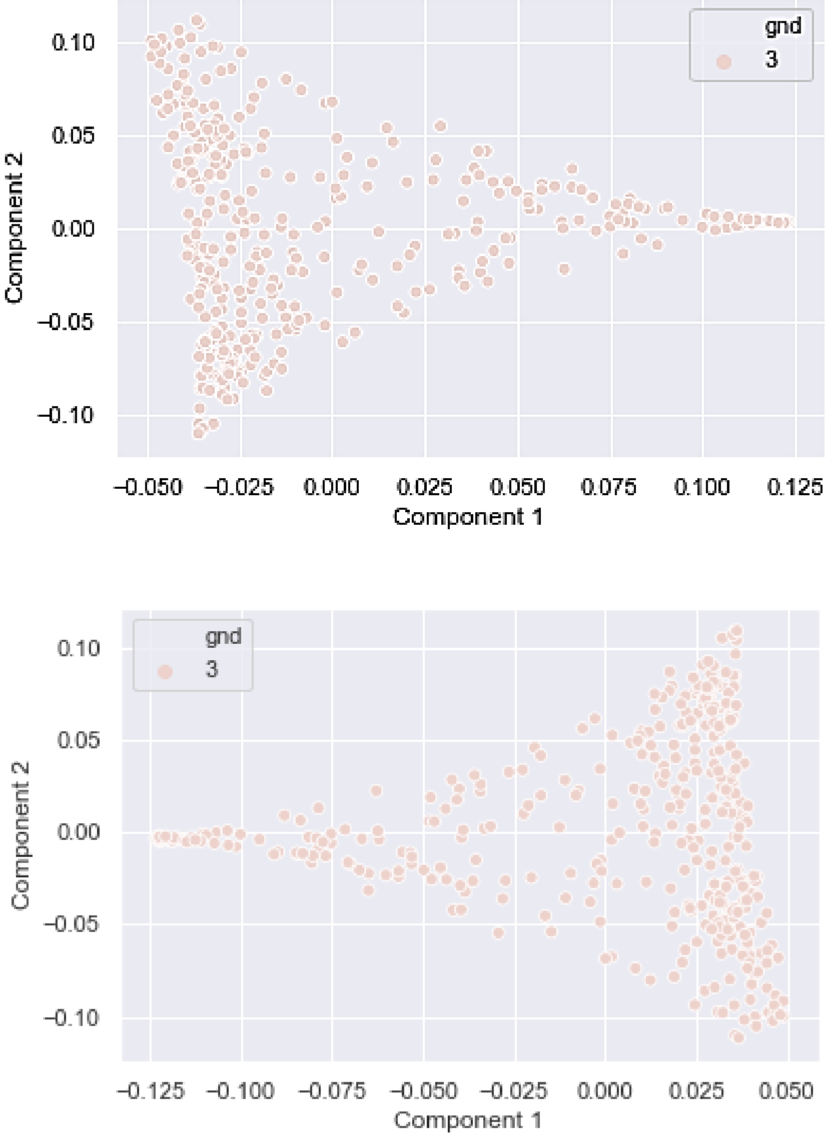


Figure 9: LLE on Digit-3 instances

2. Answer:

Code: “asgnmnt1_3_1_2.py”

ISOMAP algorithm method was applied to the instances of Digit-3. No specific pattern was observed after plotting the first 2 components in 2D plane, as plotted in Figure 10. Hence, ISOMAP didn’t improve the situation and visibly provides poor performance compared to LLE in this case.

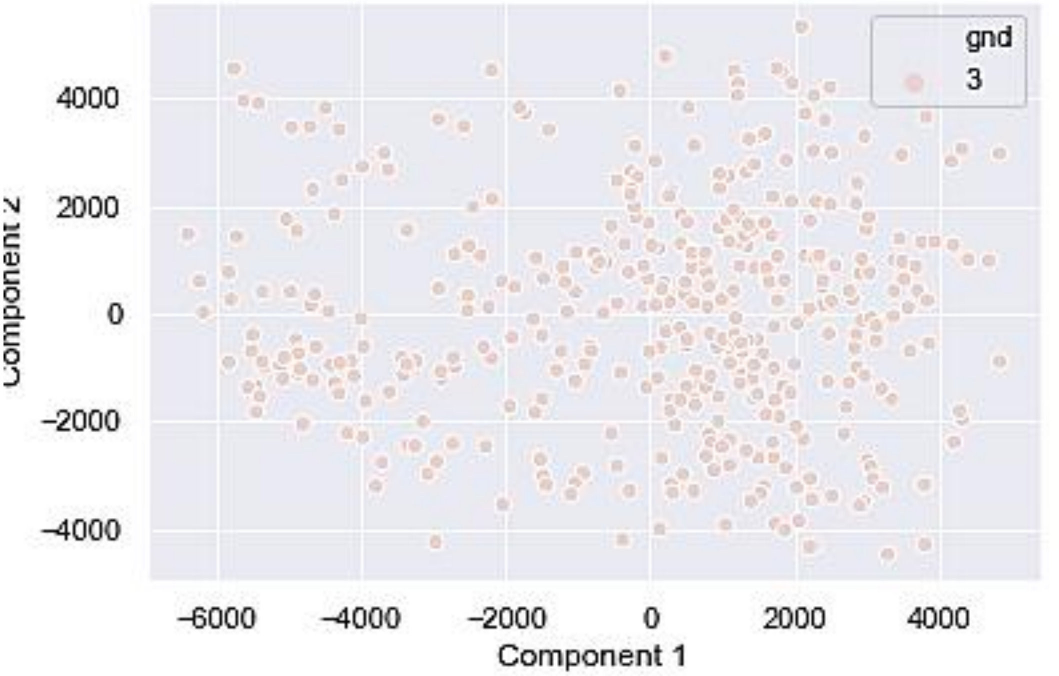


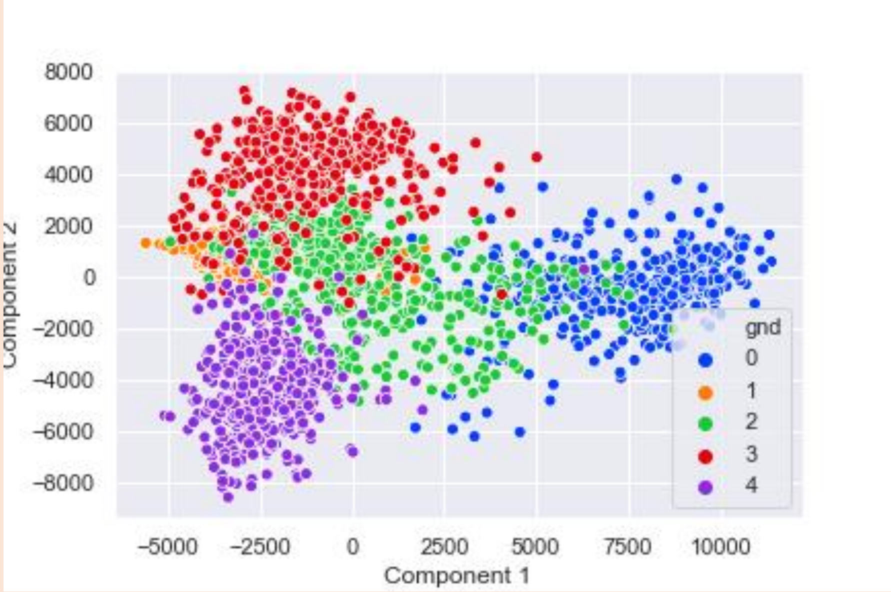
Figure 10: ISOMAP method on Digit-3 instances

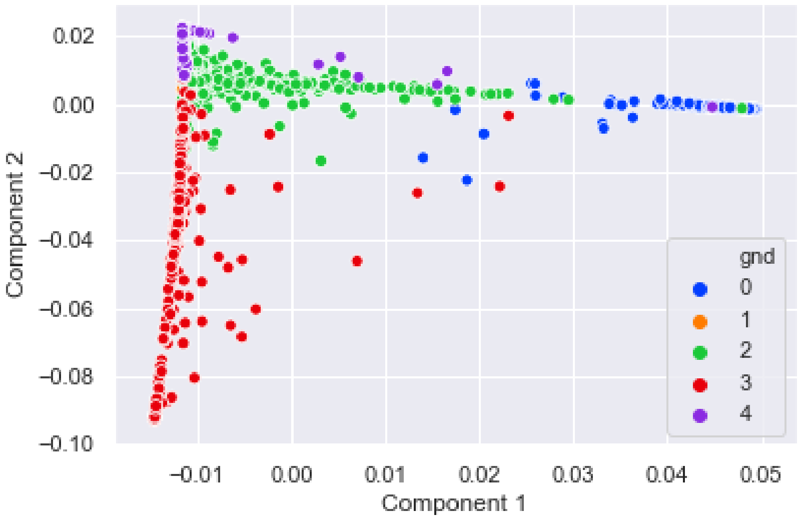
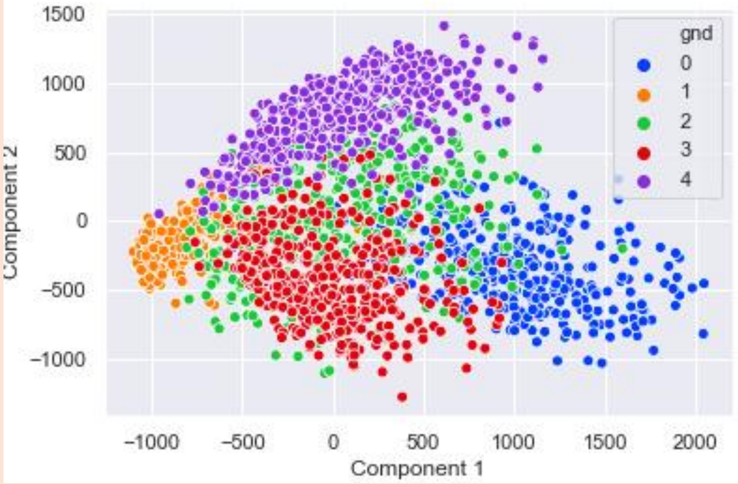
3. Answer:

Code: “assignment1_3_3.py”

- Naive Bayes classifier was applied to classify the dataset based on the projected 4-dimension representations of the LLE, ISOMAP, PCA and LDA.
    - First, whole dataset was dimensionally-reduced using LLE, ISOMAP, PCA and LDA to 4-components. Each resultant dataset was captured and Table-I contains the plot of each case.
    - In case of each new low-dimension dataset, Gaussian NB classifier was trained by randomly selecting 70% of data, and test with remained 30%. A variation on k-fold cross-validation termed as “Repeated Random Test-Train Splits”, was employed. This has the speed of using a train/test split and the reduction in variance in the estimated performance of k-fold cross-validation. Based on the average accuracies, their performance was recorded.
    - Table-II contains the classification report, confusion matrix and accuracy score of each case. The best accuracy was obtained from LDA. The second, third and last would be ISOMAP, LLE and PCA.

Table-I

No.	**Dimensionality Reduction Method**	**Plot of Classes in 2D after Dimension Reduction**
1	ISOMAP	

2	LLE	0.02 0.00 −0.02 −0.04 −0.06 −0.08 −0.10 Component 2 −0.01 0.00 0.01 0.02 0.03 0.04 0.05 Component 1 gnd 0 1 2 3 4
3	PCA	1500 1000 500 0 −500 −1000 Component 2 −1000 -500 0 500 1000 1500 2000 Component 1 gnd 0 1 2 3 4

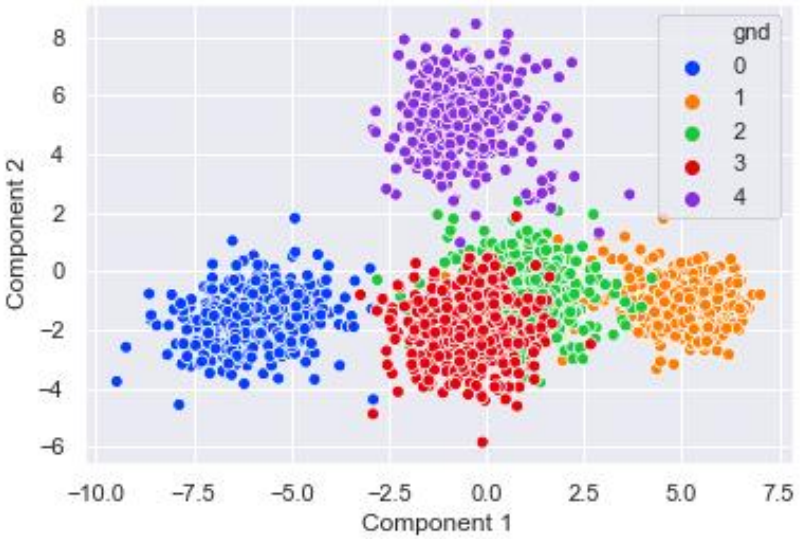
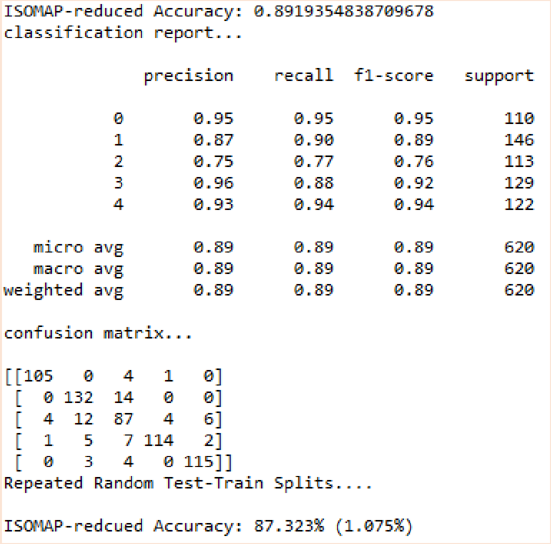
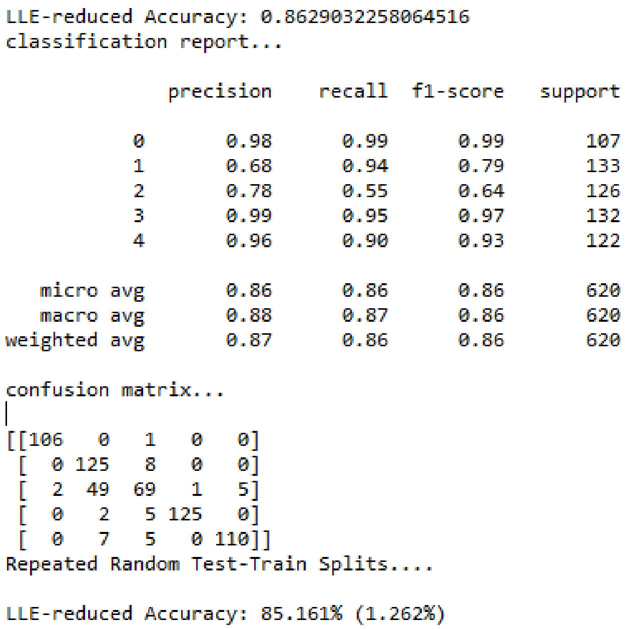
4	LDA	gnd 0 1 2 3 4 Component 2 8 6 4 2 0 -2 -4 -6 −10.0 −7.5 −5.0 −2.5 0.0 2.5 5.0 7.5 Component 1

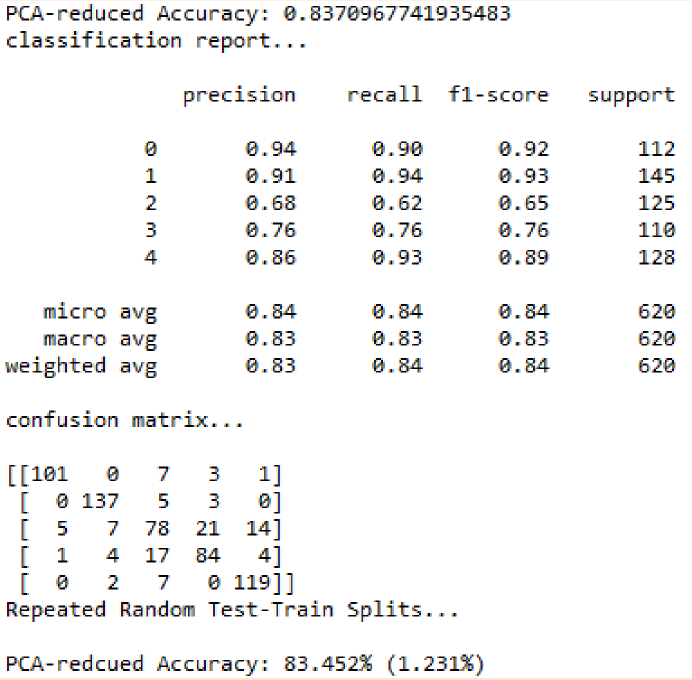
Table-II

Dim. Reduction Method No.	**Method Name**	**Gaussian Naïve-Bayes Classifier Model Classification Performance on Dimension-reduced Dataset**
1	ISOMAP	
2	LLE	

```
ISOMAP-reduced Accuracy: 0.8919354838709678
classification report...

              precision    recall  f1-score   support

           0       0.95      0.95      0.95       110
           1       0.87      0.90      0.89       146
           2       0.75      0.77      0.76       113
           3       0.96      0.88      0.92       129
           4       0.93      0.94      0.94       122

   micro avg       0.89      0.89      0.89       620
   macro avg       0.89      0.89      0.89       620
weighted avg       0.89      0.89      0.89       620

confusion matrix...

[[105   0   4   1   0]
 [  0 132  14   0   0]
 [  4  12  87   4   6]
 [  1   5   7 114   2]
 [  0   3   4   0 115]]
Repeated Random Test-Train Splits....

ISOMAP-redcued Accuracy: 87.323% (1.075%)
```

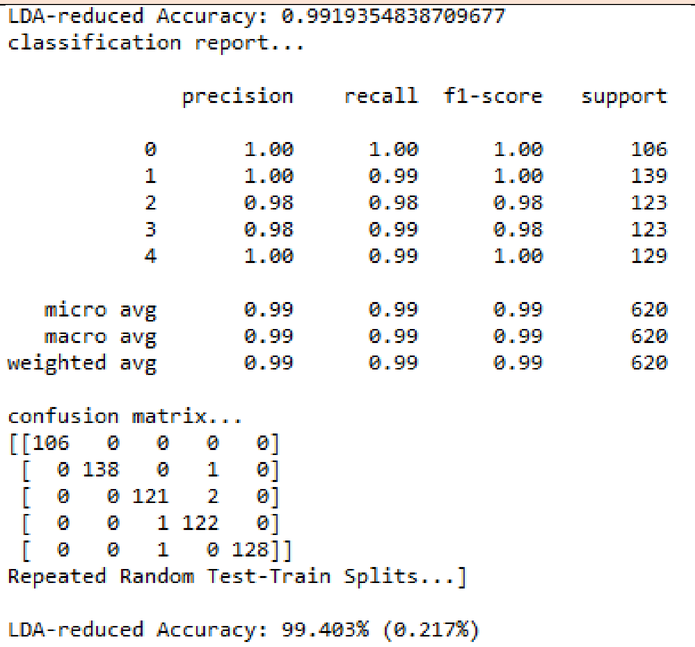
```
LLE-reduced Accuracy: 0.8629032258064516
classification report...

              precision    recall  f1-score   support

           0       0.98      0.99      0.99       107
           1       0.68      0.94      0.79       133
           2       0.78      0.55      0.64       126
           3       0.99      0.95      0.97       132
           4       0.96      0.90      0.93       122

   micro avg       0.86      0.86      0.86       620
   macro avg       0.88      0.87      0.86       620
weighted avg       0.87      0.86      0.86       620

confusion matrix...

[[106   0   1   0   0]
 [  0 125   8   0   0]
 [  2  49  69   1   5]
 [  0   2   5 125   0]
 [  0   7   5   0 110]]
Repeated Random Test-Train Splits....

LLE-reduced Accuracy: 85.161% (1.262%)
```

<table>
<tr>
<td><i>3</i></td>
<td>PCA</td>
<td>



```
PCA-reduced Accuracy: 0.8370967741935483
classification report...

              precision    recall  f1-score   support

           0       0.94      0.90      0.92       112
           1       0.91      0.94      0.93       145
           2       0.68      0.62      0.65       125
           3       0.76      0.76      0.76       110
           4       0.86      0.93      0.89       128

   micro avg       0.84      0.84      0.84       620
   macro avg       0.83      0.83      0.83       620
weighted avg       0.83      0.84      0.84       620

confusion matrix...

[[101   0   7   3   1]
 [  0 137   5   3   0]
 [  5   7  78  21  14]
 [  1   4  17  84   4]
 [  0   2   7   0 119]]
Repeated Random Test-Train Splits...

PCA-redcued Accuracy: 83.452% (1.231%)
```

</td>
</tr>
<tr>
<td><i>4</i></td>
<td>LDA</td>
<td>



```
LDA-reduced Accuracy: 0.9919354838709677
classification report...

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       106
           1       1.00      0.99      1.00       139
           2       0.98      0.98      0.98       123
           3       0.98      0.99      0.98       123
           4       1.00      0.99      1.00       129

   micro avg       0.99      0.99      0.99       620
   macro avg       0.99      0.99      0.99       620
weighted avg       0.99      0.99      0.99       620

confusion matrix...
[[106   0   0   0   0]
 [  0 138   0   1   0]
 [  0   0 121   2   0]
 [  0   0   1 122   0]
 [  0   0   1   0 128]]
Repeated Random Test-Train Splits...]

LDA-reduced Accuracy: 99.403% (0.217%)
```

</td>
</tr>
</table>