Data604_final_project

Chih-Hua Chang UID: 118143422

Data Introduction

In this project, I use fruits 360 from Kaggle to test the accuracies of KNN classifier with different dimensionality reduction techniques applied on the dataset. I choose three different types of fruits, including apple red 1, banana red, and cherry 2 from the dataset for my following research. As the three classes have similar color or shape, it will be more difficult to distinguish by camera eyes.

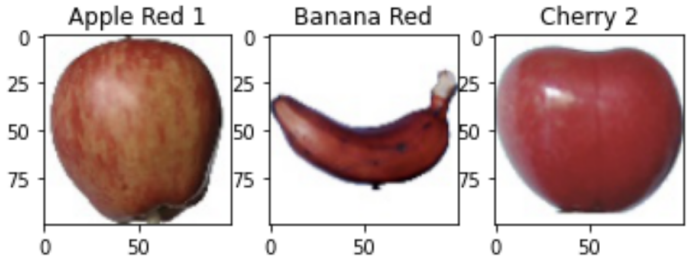


Figure 1: The first images of the three classes

Each jpeg image is 100*100 pixels, and the original shape of an image array is 100*100*3 (data type = unit8). I decided to use the mean of the three matrix, which represents the rbg colors, so the shape of the image is 100*100.

Select the Optimal Training Set Size

The “apple red 1” dataset contains 656 instances, the “banana red” dataset contains 656 instances, and the “cherry 2” dataset contains 984 instances. To decide how many training data to use, I tested the accuracies of KNN (k=20) classifier with different training set size. The training set contains N instances for each of the three classes. It can be seen in figure 2 that when N increases to 300, the accuracy

remain stable at around 98% afterward. Thus, I decide to select N = 300, which is 300*3 = 900 instances in total, as my training set size.

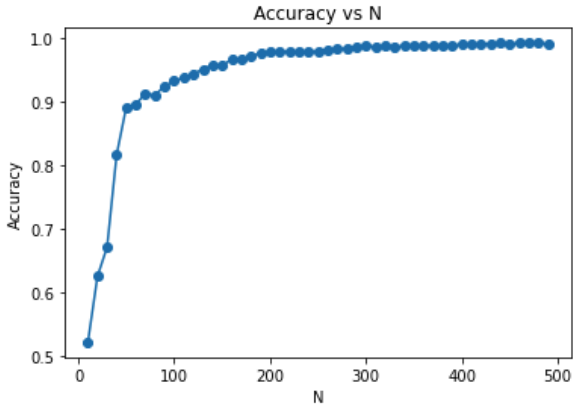


Figure 2: Relationship between accuracy and N

The consuming time for the classifier is about 0.44 seconds. (Figure 2)

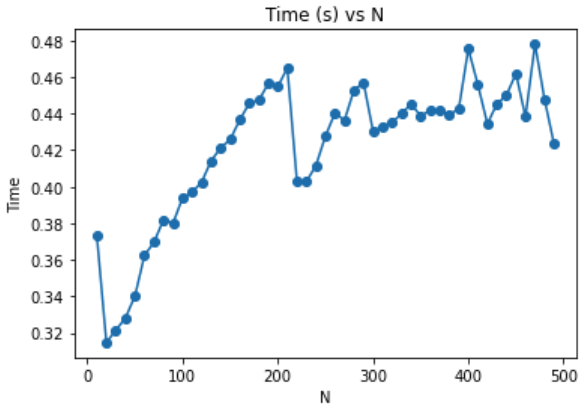


Figure 3: Relationship between consuming time and N

Now, the training set includes 900 instances (300 instances for each class). For the validation set, I select 300 instances in total (100 instances for each class), and for the test set, I select 768 instances (256 instances for each class).

Apply KNN Classifier on Raw Data

First, select the optimal k for the KNN classifier. Figure 4 shows that the accuracy of the validation set is 100% when k < 9 and decrease gradually when k gets bigger. In this case, any k that is lower than 9 is an optimal parameter and I apply k = 5 on my testing set.

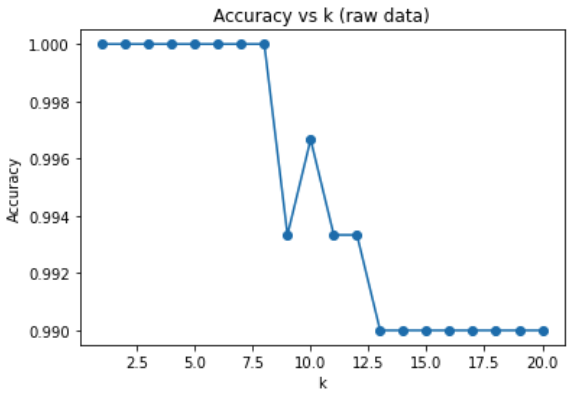


Figure 4: Raw data acccuracy with respect to different k in KNN classifier

With k = 5 applied on the testing set, the global accuracy is 100%. The accuracies of the three classes are all 100%. Overall, the KNN classifier with k = 5 performs excellent on the raw data. (Figure 5)

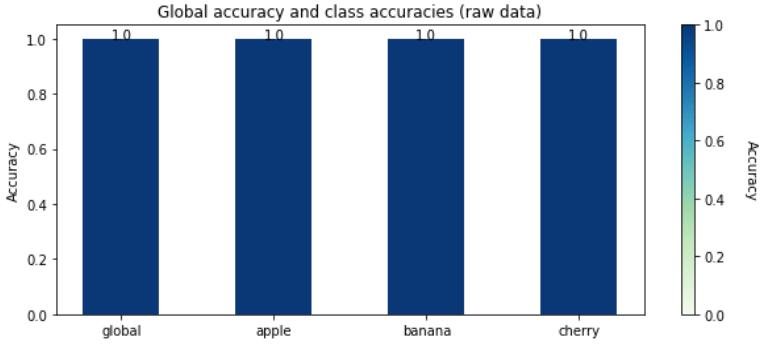


Figure 5: Raw data accuracies with KNN classifier

Figure 6 shows the confusion matrix of the KNN classifier for raw data. As precision is defined as TP / (TP+FP), the precisions for all three fruits are 1. As recall is refine as TP / (TP+FN), the recall for all three fruits are 1.

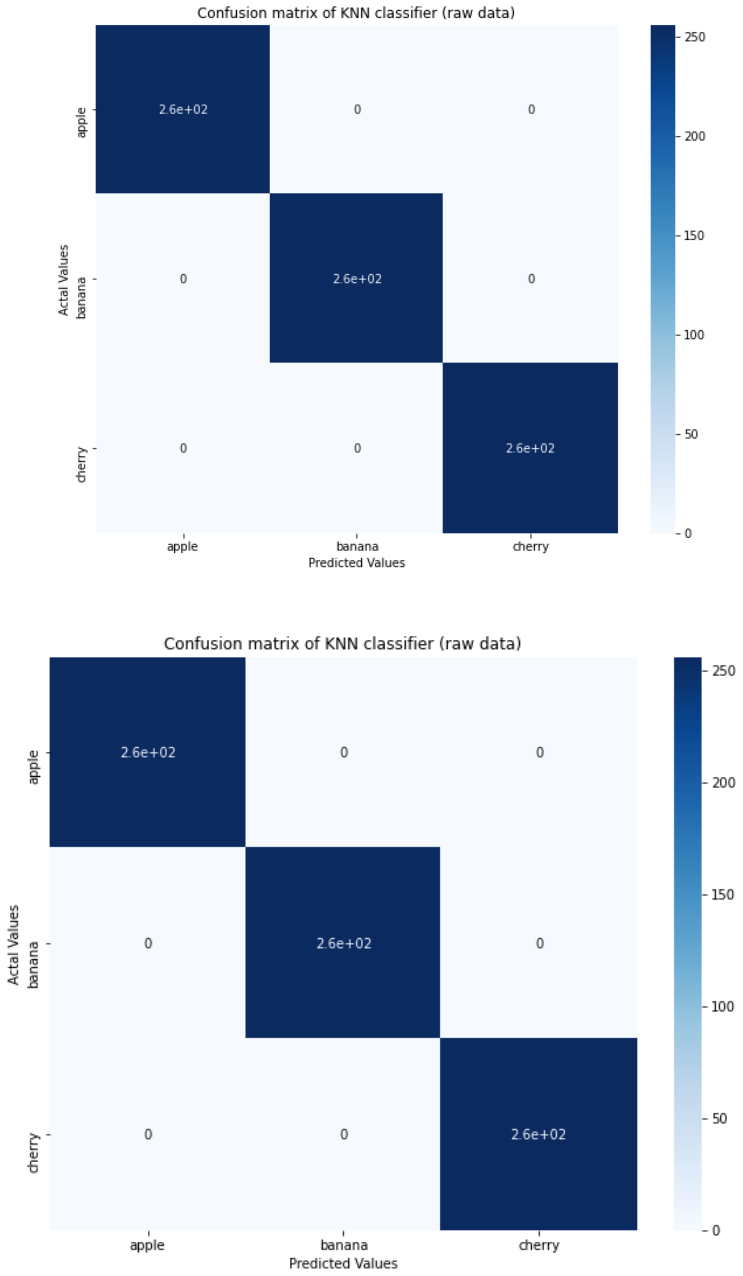


Figure 6: Confusion matrix of KNN classifier for raw data

PCA Introduction

PCA is a dimensionality reduction method that project the data points on a new basis and preserve as much information as possible. The eigenvalues, which is the variances, of the data's eigenvectors show the size of the information which the data contain in the different proportion of feature formulas. The larger the eigenvalue is, the more information it contains. Projecting the data point to the first few components

will reduce the dimensionality of the raw data but still preserve enough information to do the further analysis.

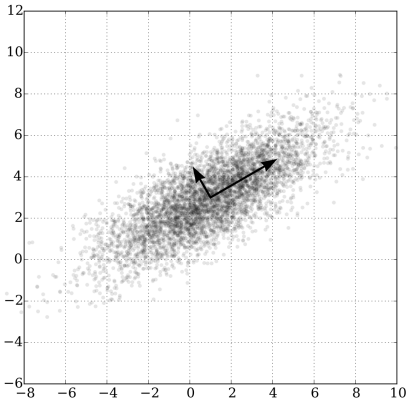
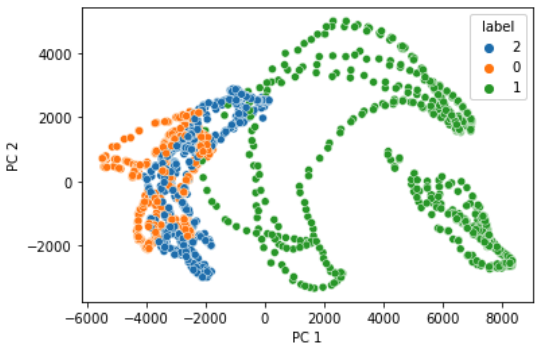


Figure 7: The first two eigenvectors that contains the most information in the data.

Apply KNN Classifier on PCA Transformed Data

First, take a look at how the dataset represents with only two principal component remain.

Figure 8 shows that with two principal components representing the dataset, apple and cherry mixed together at a large degree.



class 0: apple; class 1: banana; class 2: cherry

Figure 8: PCA transformed data on two dimesional subspace

By observing the cumulative variance and the numbers of components (eigenvector) used (Figure 9), we can see that 90% of the information (variance) is represented by about 20 components. Thus, I use 20 components for the PCA dimensionality reduction of the raw data.

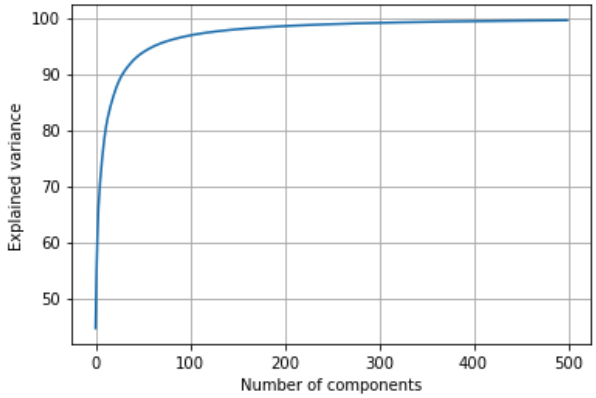


Figure 9: Cumalative variance with different numbers of components selected

Figure 10 indicates that with k = 1 to k = 12, the accuracy of the validation set is 100%. As k increases, the accuracy drops. Thus, k that is lower than 12 is an optimal parameter, and I decide to use k = 7 on the KNN classifier.

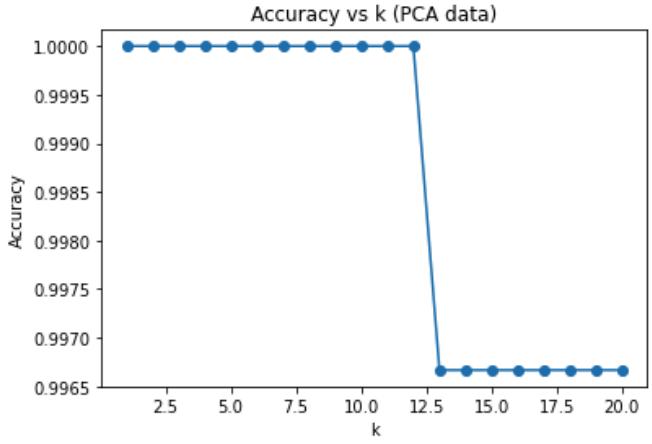


Figure 10: PCA transformed dataset acccuracy with respect to different k in KNN classifier

With k = 7, the global accuracy of the testing set is 99.9%. The accuracies of apple and cherry are both 100%, and the accuracy of banana is 99.6%. The KNN classifier with k = 7 performs excellent on the PCA transformed data. (Figure 11)

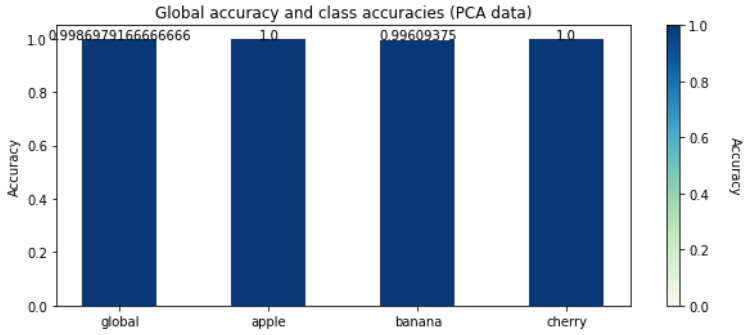


Figure 11: PCA transformed data accuracies with KNN classifier

Figure 12 shows the confusion matrix of KNN classifier for PCA transformed data. The precision of apple, banana, and cherry are 0.996, 1, and 1 The recall of apple, banana, and cherry are 1, 0.996, and 1.

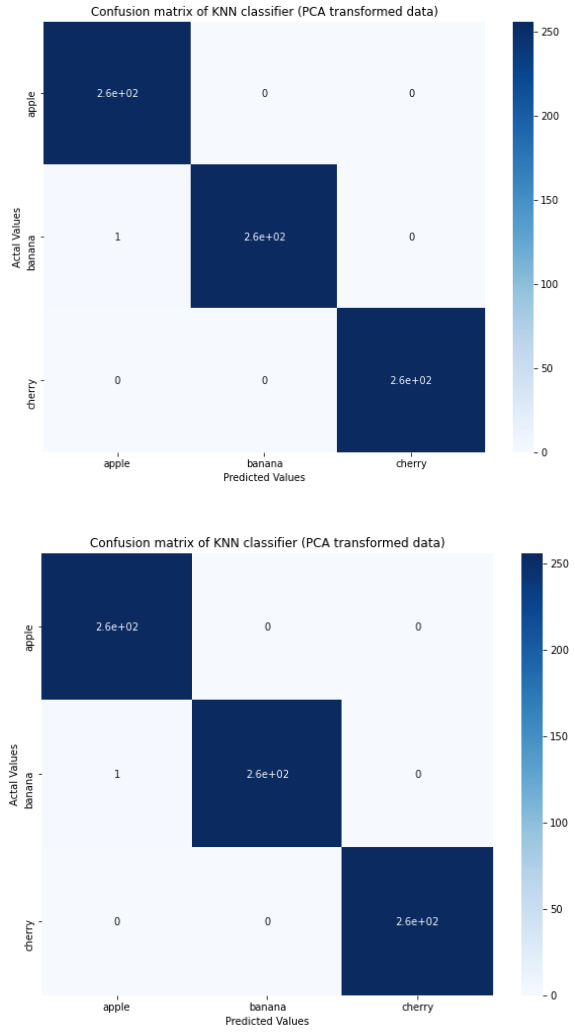


Figure 12: Confusion matrix of KNN classifier for PCA transformed data

Kernel PCA (kPCA) Introduction

kPCA is an extension of the conventional PCA with a kernel trick. The kernel trick is a calculation cost saving and non-trivial process that does not require calculating the actual covariance, so it would be easier to construct a hyperplane that separate the data into different clusters. Noted that data that is not linearly separable in n<d dimensional space would almost always be separable in n≥d dimensional space.

Thus, when the data that is not linearly separable and not in a manifold, it has a high possibility to be separated by kPCA.

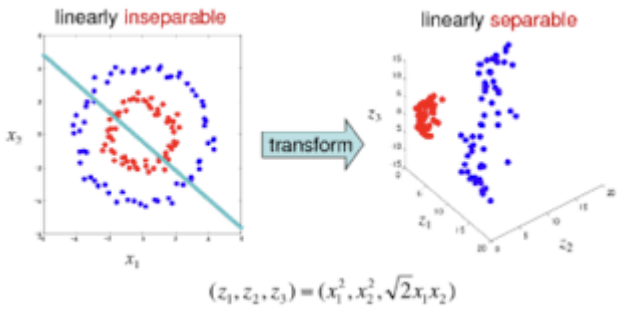


Figure 13: linearly inseparable data transformed to linearly separable data by kPCA

Apply KNN Classifier on kPCA Transformed Data

By projecting the data to a two-dimensional subspace by kPCA, the three classes are mixed and is difficult to distinguish between classes. (Figure 14)

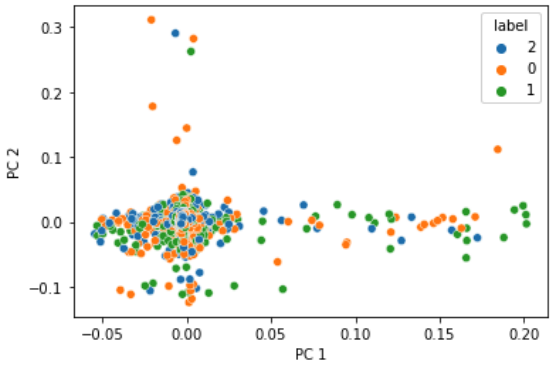


Figure 14: kPCA transformed data on two dimesional subspace

Same as the PCA transformed data, the first 20 components contains 90% of the information. Apply kernel PCA with n_components equals to 20 and see which k performs the best in KNN classifier.

Figure 15 shows that the kernel PCA transformed validation set performs undesirably with the highest accuracy only reach to 36% when k = 20.

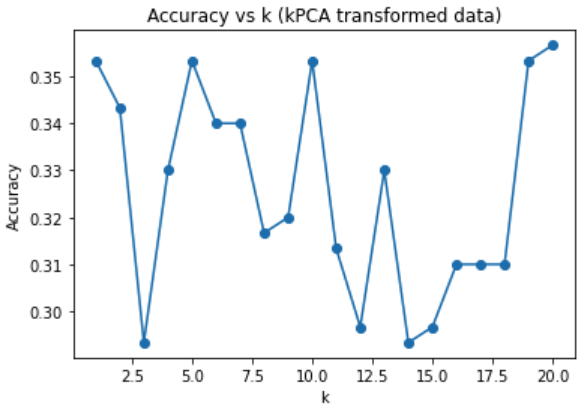


Figure 15: kPCA transformed data acccuracy with respect to different k in knn classifier

With k = 20, the global accuracy of the test set is 32.3%. The accuracy of apple is 39.5%, the accuracy of banana is 34.4%, and the accuracy of cherry is 23.0%. All classes performs undesirably in the kPCA transformed data, and among them, apple performs the best and banana performs the worst. (Figure 16)

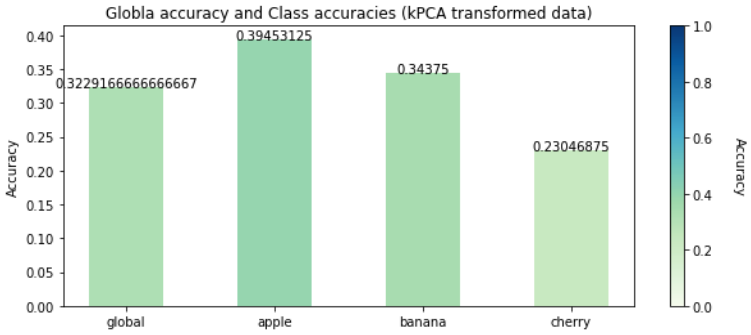


Figure 16: kPCA transformed data accuracies with KNN classifier

Figure 17 shows the confusion matrix of KNN classifier for kPCA transformed data. The precision of apple, banana, and cherry are 0.33, 0.33, and 0.30. The recall of apple, banana, cherry are 0.39, 0.34, and 0.23.

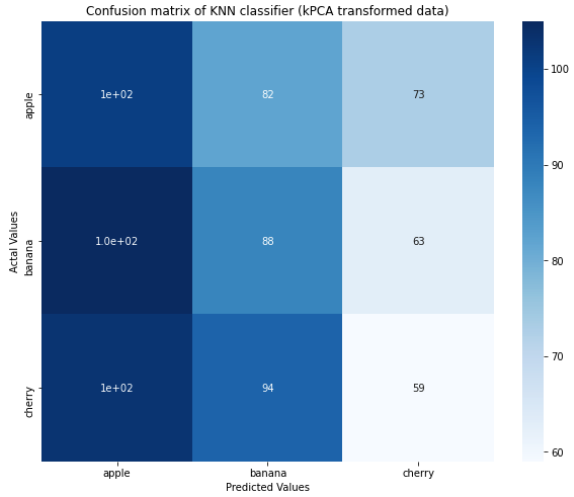


Figure 17: Confusion matrix of KNN classifier for kPCA transformed data

Laplacian Eigenmaps (LE) Introduction

The Laplacian Eigenmaps is based on the intrinsic value of the manifold and is insensitive to outliers and noise. The graph Laplacian applied in the algorithm is used for clustering and partition problems. It is often been used to project manifold data points on a lower dimensional space by building a neighborhood graph, connecting only nearby points, and applying MDS. Noted that Laplacian Eigenmaps focus on preserving the local geometry, which means that nearby points in the original space remain nearby in the reduced place, but the actual distance will change.

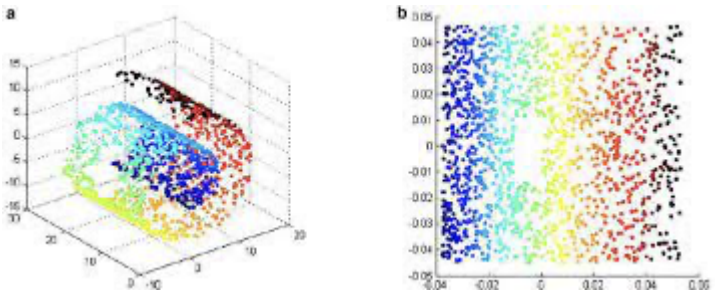


Figure 18: Manifold data transformed to linearny separable data by LE

Apply KNN Classifier on LE Transformed Data

The intrinsic value calculated by maximum likelihood estimator is 2.4, so I decide to select n = 2 as the parameter of LE dimensionality reduction technique.

Figure 19 shows that the LE transformed data representing in two-dimension subspace. We can see that banana is well separated from apple and cherry, which is better than kernel PCA. However, apple and cherry still mixed together.

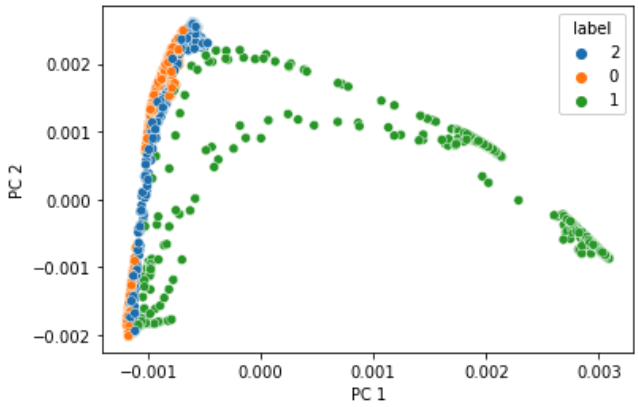


Figure 19: LE transformed data on two dimesional subspace

When k = 2, the accuracy of the validation set is about 94%. However, as k increase, the accuracy decline gradually. Thus, the optimal k is 2. (Figure 20)

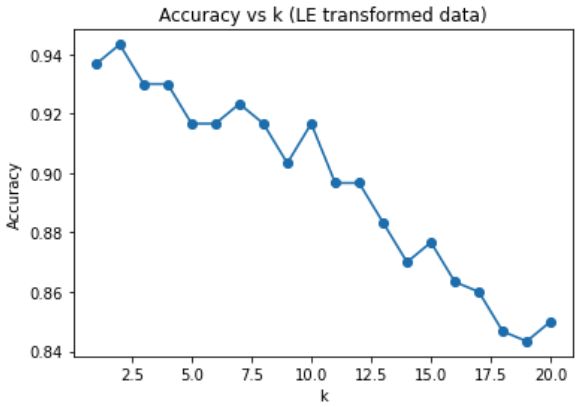


Figure 20: LE transformed data acccuracy with respect to different k in knn classifier

Figure 21 shows that banana has the highest accuracy among the three classes. The global accuracy is 92.3% and all classes performs well in the LE transformed data.

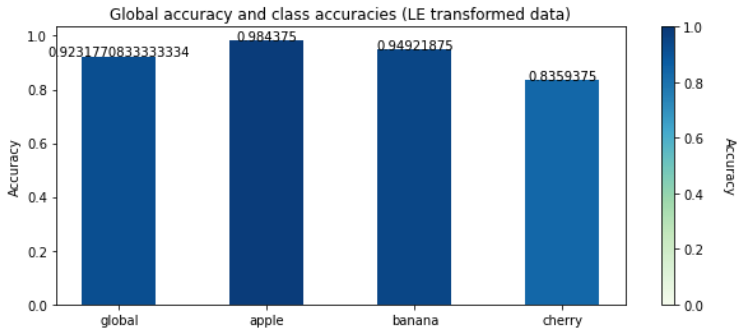
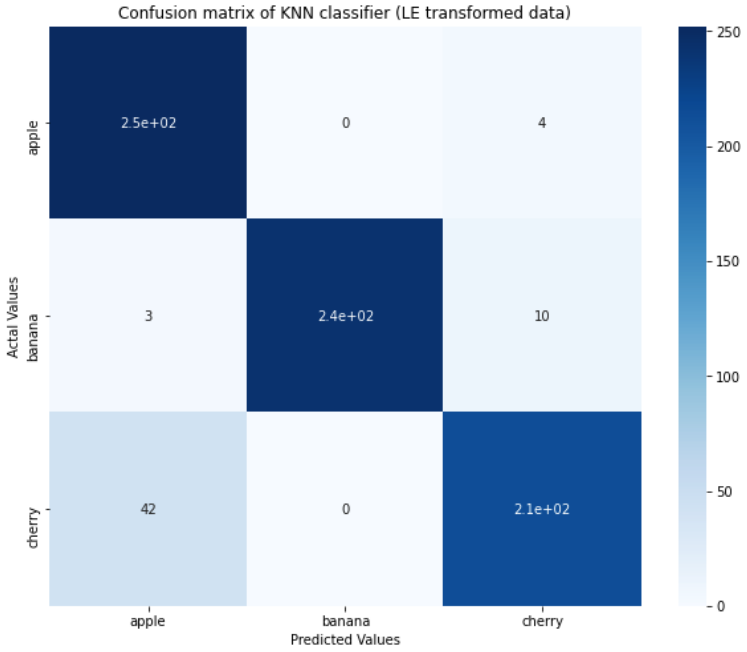


Figure 21: LE transformed data accuracies with KNN classifier

Figure 22 shows the confusion matrix of KNN classifier for LE transformed data. The precision of apple, banana, and cherry are 0.84, 1, and 0.94. The recall of apple, banana, and cherry are 0.98, 0.95, and 0.84.



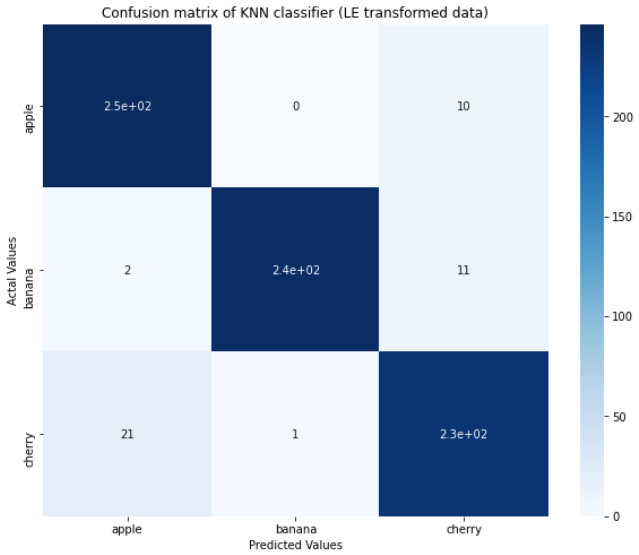


Figure 22: Confusion matrix of KNN classifier for LE transformed data

Conclusion

In conclusion, the raw dataset performs the best among the four dataset and receives a 99.9% global accuracy, which make sense because every information is preserved.

With PCA applied to the dataset, the dataset still performs excellent with 20 principal components preserved and receives an accuracy similar to the raw dataset.

With Laplacian Eigenmaps applied to the dataset, the dataset performs well in two-dimension subspace and receives an accuracy higher than 90%.

Finally, with kPCA applied to the dataset, the dataset performs the worst and receive an accuracy lower than 33%, which is the probability of guessing without any information.

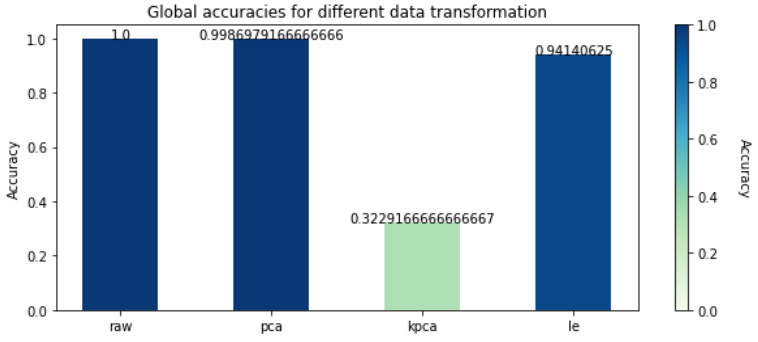


Figure 23: Global accuracies for the four datasets

References

Kaggle Fruit 360 Data

https://www.kaggle.com/datasets/moltean/fruits

PCA

https://towardsdatascience.com/principal-component-analysis-part-1-the-different-formulations-6508f63a5553

https://towardsdatascience.com/image-compression-using-principal-component-analysis-pca-253f26740a9f

Kernel PCA

https://en.wikipedia.org/wiki/Principal_component_analysis

https://towardsdatascience.com/kernel-pca-vs-pca-vs-ica-in-tensorflow-sklearn-60e17eb15a64

https://en.wikipedia.org/wiki/Kernel_principal_component_analysis

https://cmdlinetips.com/2021/02/kernal-pca-with-python/

PCA vs kPCA

https://nirpyresearch.com/pca-kernel-pca-explained/

Dimensionality estimators

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.1327&rep=rep1&type=pdf

https://scikit-dimension.readthedocs.io/en/latest/api.html

LE

https://juanitorduz.github.io/laplacian_eigenmaps_dim_red/

https://scikit-learn.org/stable/modules/generated/sklearn.manifold.SpectralEmbedding.html

https://www.sjsu.edu/faculty/guangliang.chen/Math253S20/lec12laplacian.p