MACHINE LEARNING FINAL PROJECT REPORT

FRUITS CLASSIFICATION

TEAM MEMBERS:
TARUN PUNHANI: TXP190029
KIRAN DEAVARAJ RAJ: KXR190038

PROBLEM STATEMENT

We have a dataset containing images of fruits and vegetables. We are performing multiclass image classification using different Scikit Learn Classification Algorithms and analyzing and comparing the results.

Algorithms used:

Decision Tree, Bagging, Boosting, SVM, KNN, Artificial Neural Network

DATASET CONSIDERED

We have chosen a Kaggle dataset containing images of various fruits and vegetables. We will be applying the classification models on part of this dataset.

Original Dataset properties

Total number of images: 90483.

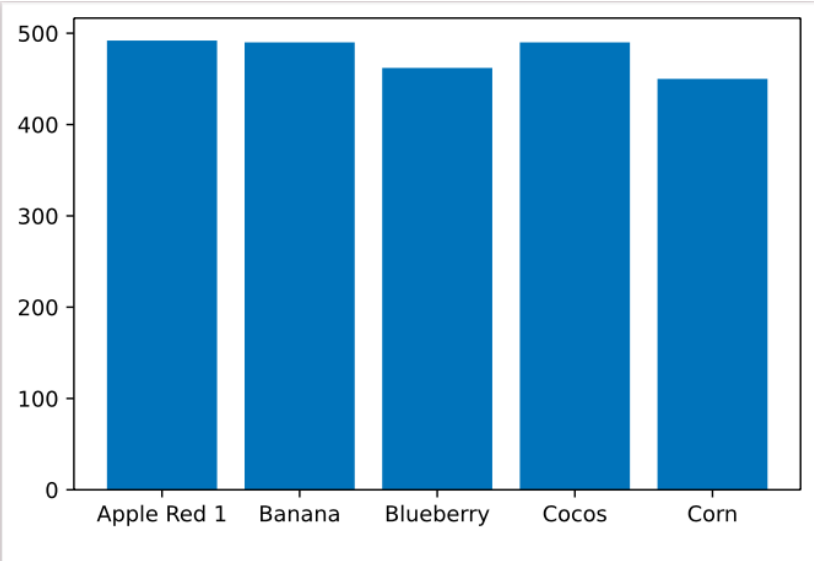
Training set size: 67692 images (one fruit or vegetable per image).

Test set size: 22688 images (one fruit or vegetable per image).

Number of classes: 131 (fruits and vegetables).

Image size: 50x50 pixels.

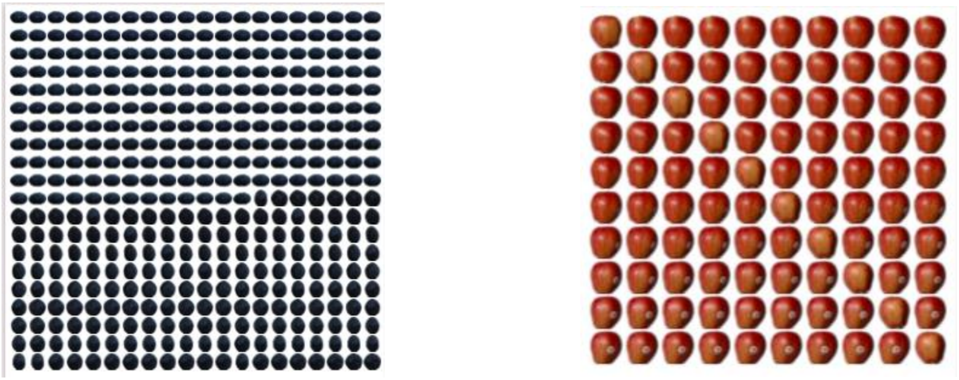
Total number of images considered in out project: 2384 for 5 classes



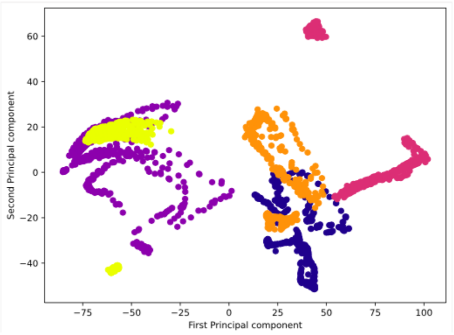
Data Pre-Processing

Since the dataset contains Images, we have to pre-process and convert it to a form suitable for applying our models on. We have used skimage library to read and preprocess images.

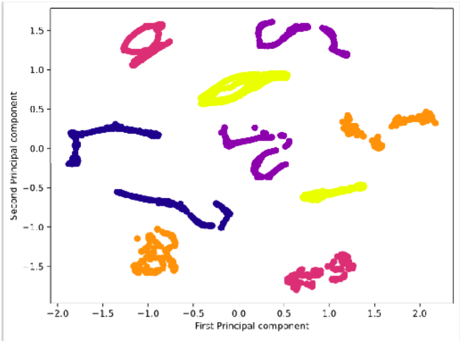
- We first read the images from the dataset using imread function of skimage.
- We then resize each image into a standard 50x50 dimension.
- We flatten the image into single dimensional array. Since its and RGB image, each image feature vector will have all three channels.
- Therefore, each image will have 3x50x50 pixels or columns in the array.
- We apply the classification models on the flattened array representation of each image.
- We can see that TSNE shows better classification than PCA



Images after Flattening



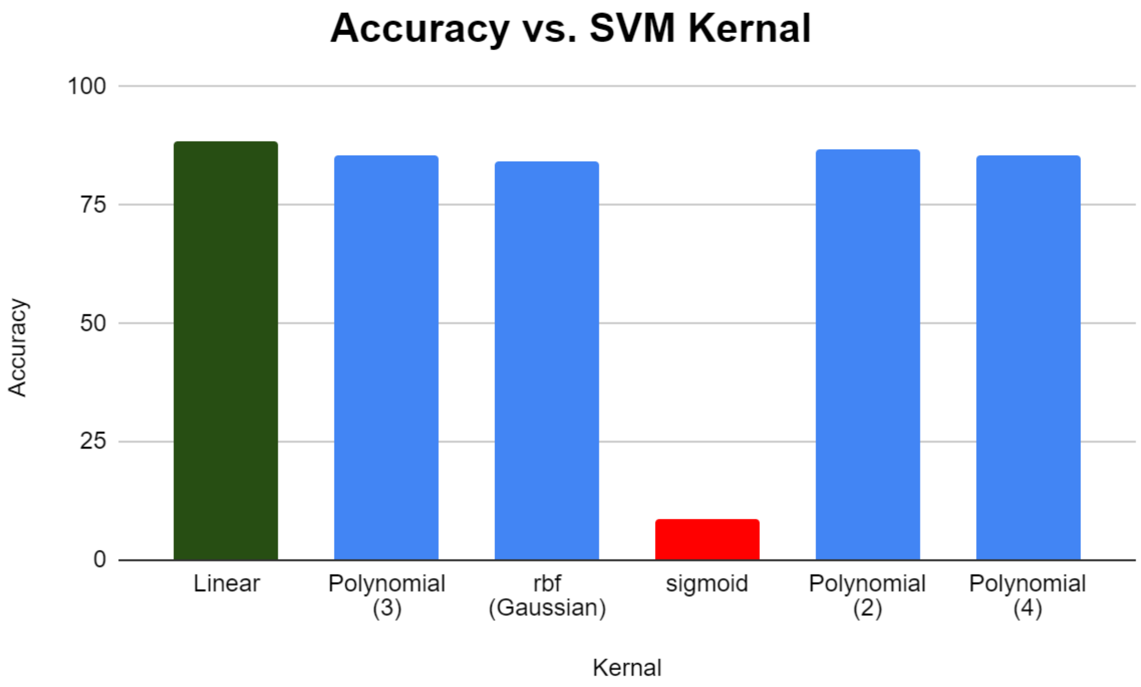
After Applying PCA



After Applying TSNE

Support Vector Machines

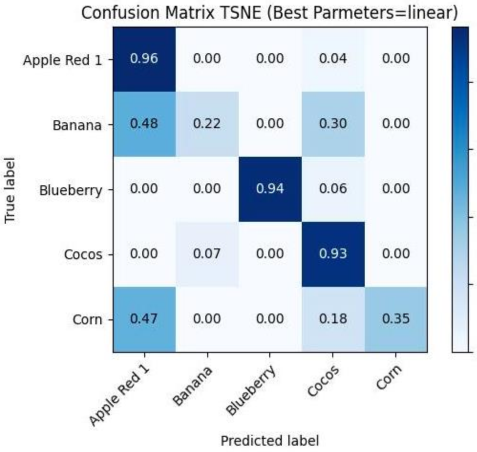
We have used different SVM kernels and compared their accuracy and out of all Linear kernel gives the best performance and all of others are almost similar except Sigmoid. The sigmoid kernel is worst among all is because it only works with binary classifiers.

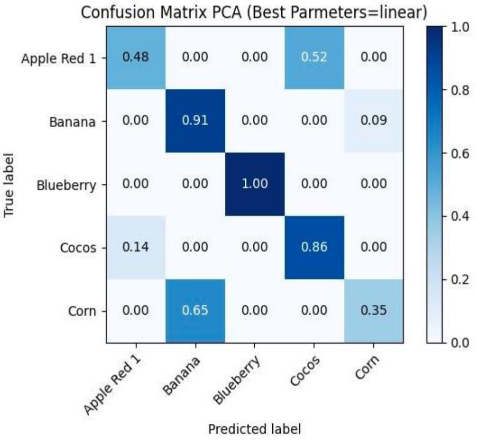


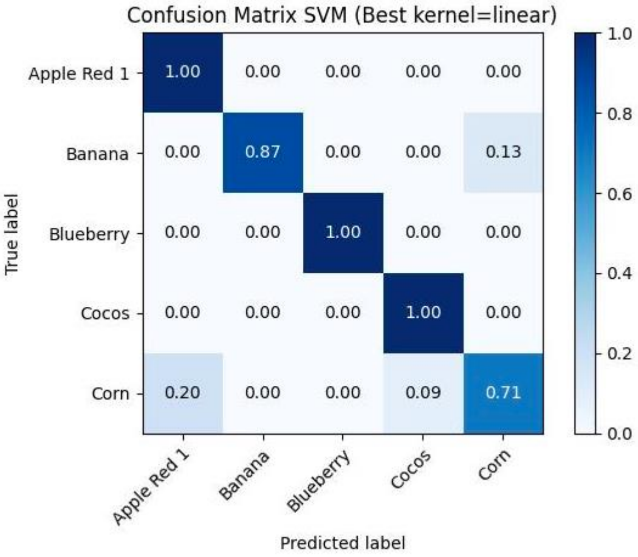
Confusion Matrices

We ran SVM on best parameters after using GridSearchCV and captured confusion matrices for train test split of 75% to 25%.

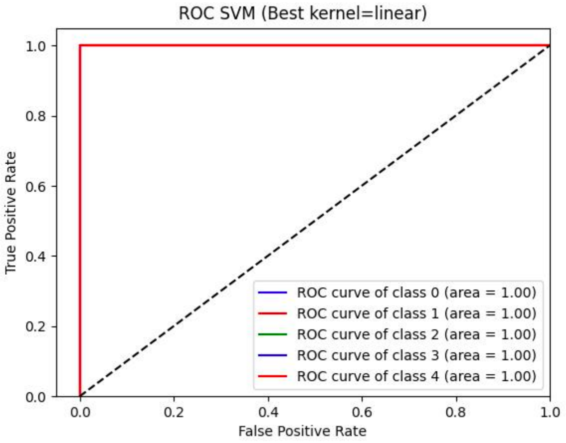
Accuracy without TSNE and PCA = 91.75%
Accuracy with PCA(50) -> TSNE(2) = 72.48%
Accuracy with only PCA = 71.64%

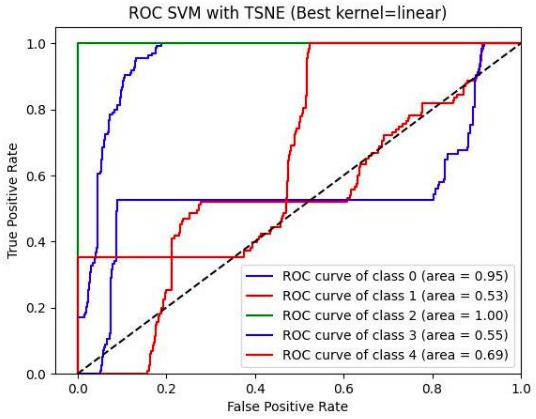


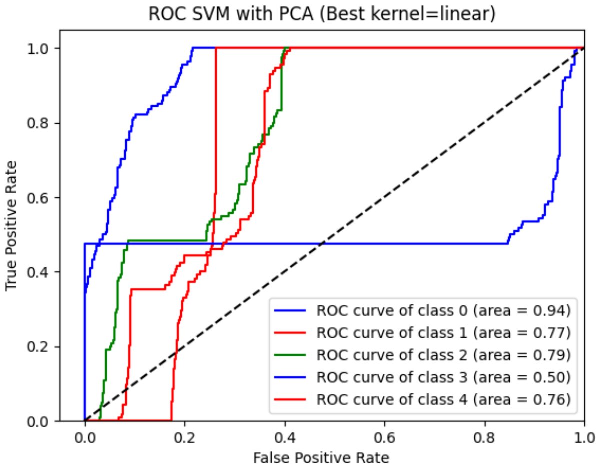




SVM ROC Curves



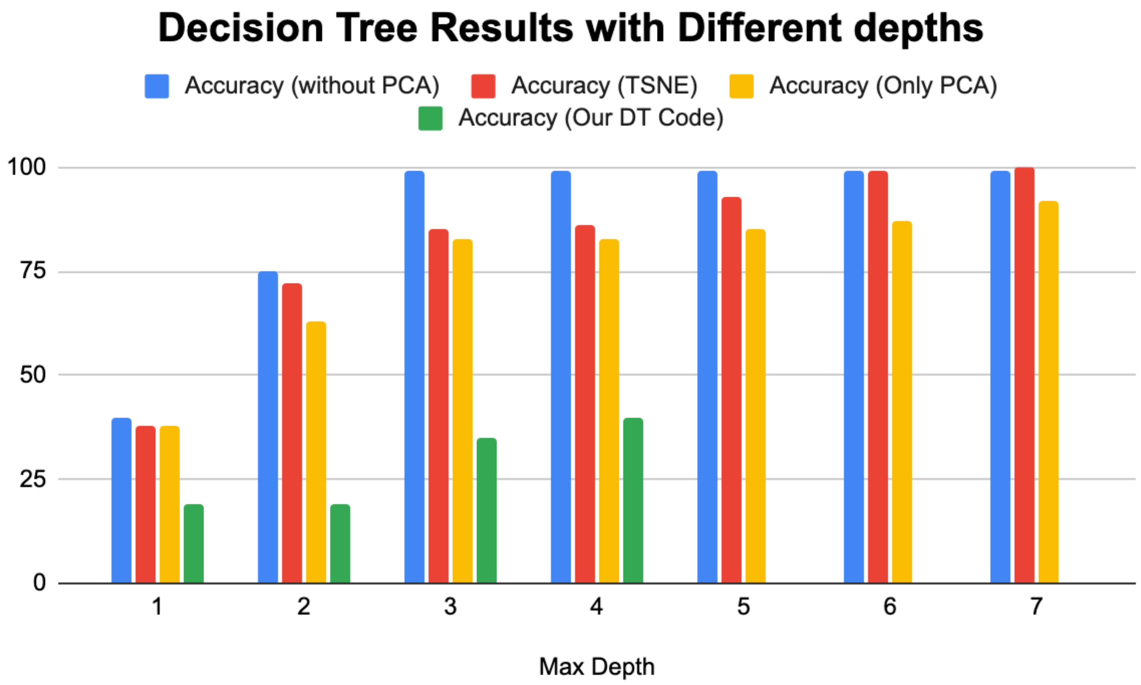




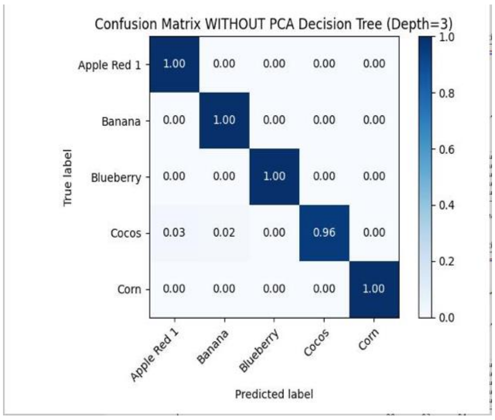
DECISION TREE CLASSIFIER

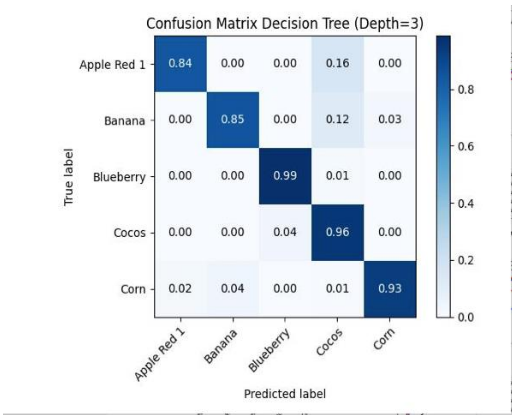
We have applied decision tree on various depths and compared their performances. The best depth for decision tree is 3.

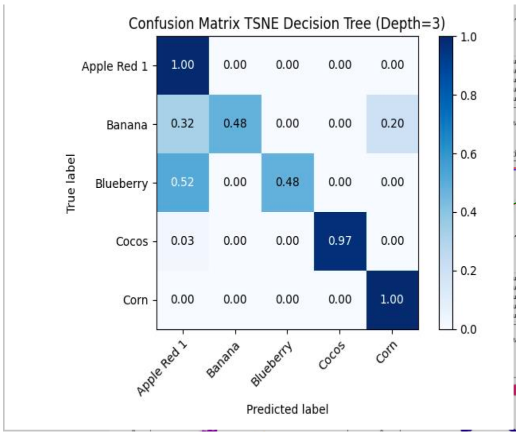
Accuracy without train and test split = 91.25%
Accuracy without TSNE and PCA = 99.16%
Accuracy with PCA(50) -> TSNE(2) = 72.65%
Accuracy with only PCA = 83.22%

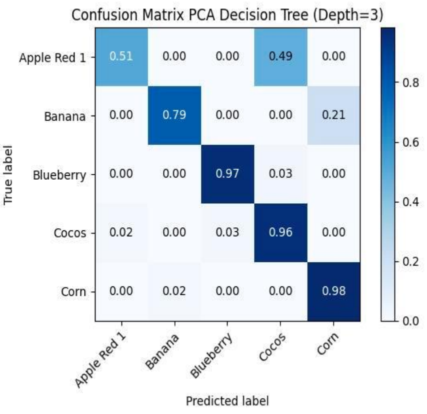


Confusion Matrices

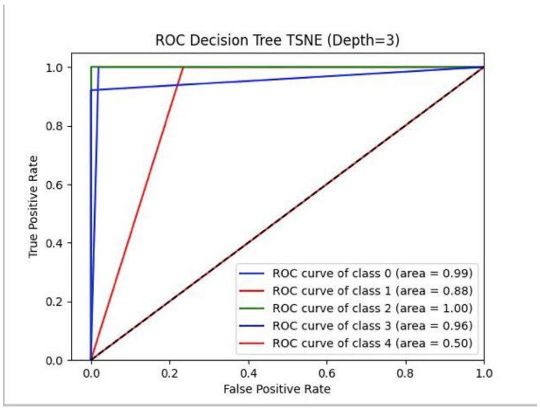


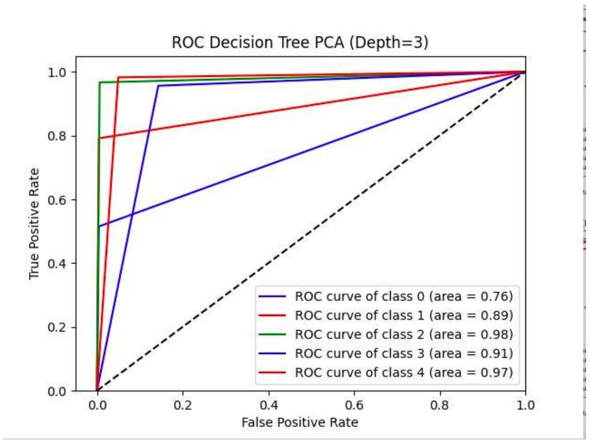


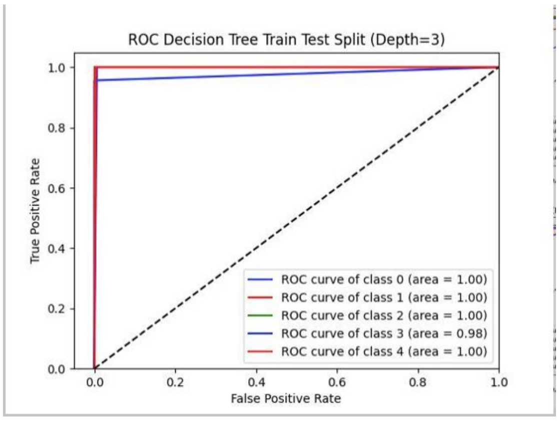


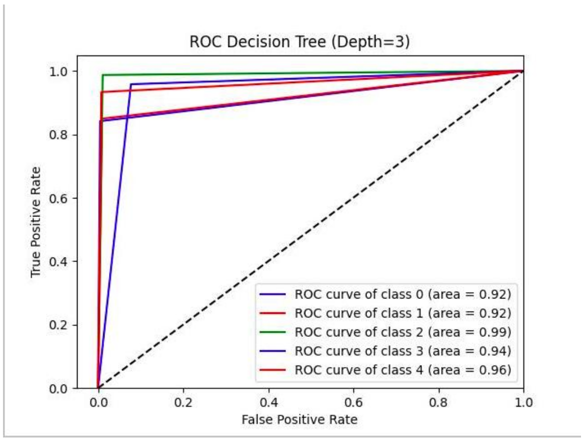


Decision Trees ROC Curves







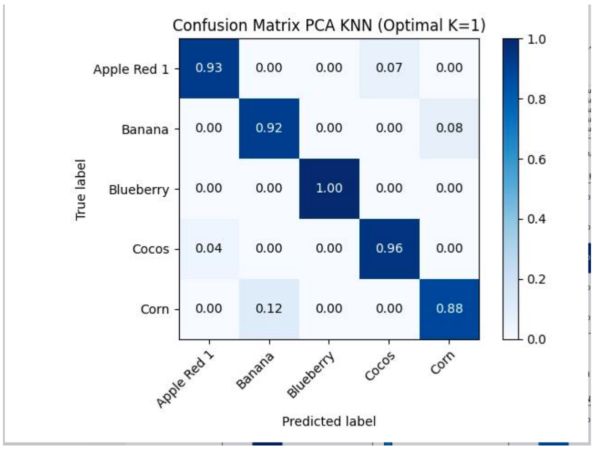


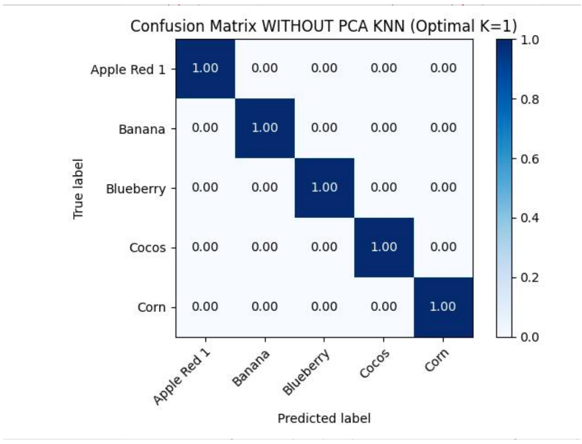
K-NEAREST NEIGHBORS

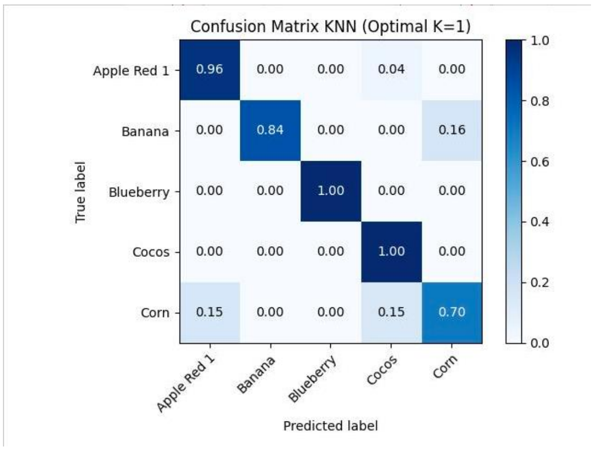
We have applied 10 fold cross validation to find out the best value of k which comes out to be ‘1’. The reason could be because of the similarity of the training and test data.

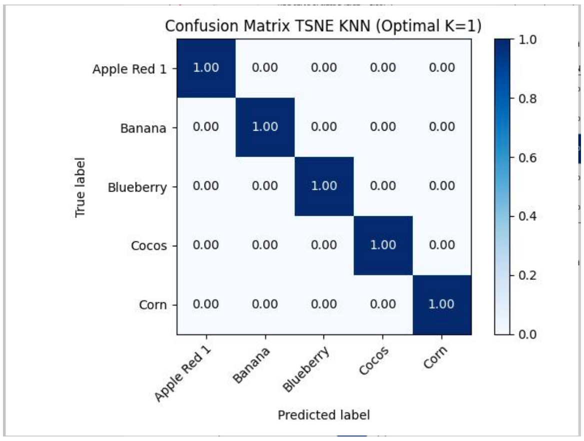
Accuracy without train and test split = 90.25%
Accuracy without TSNE and PCA = 93.96%
Accuracy with PCA(50) -> TSNE(2) = 100%
Accuracy with only PCA = 100%

Confusion Matrices

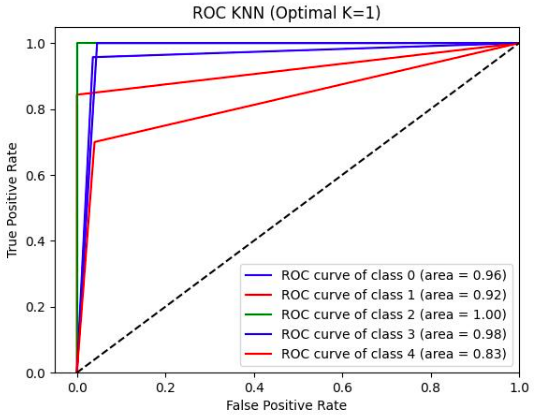


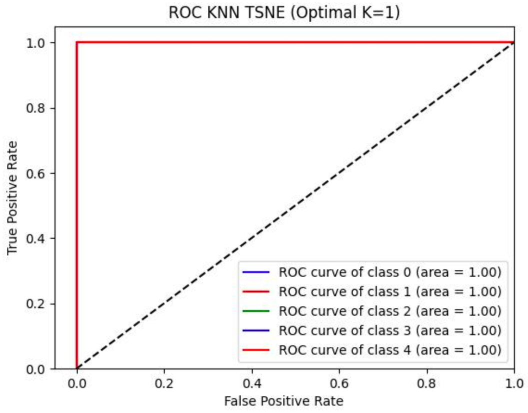


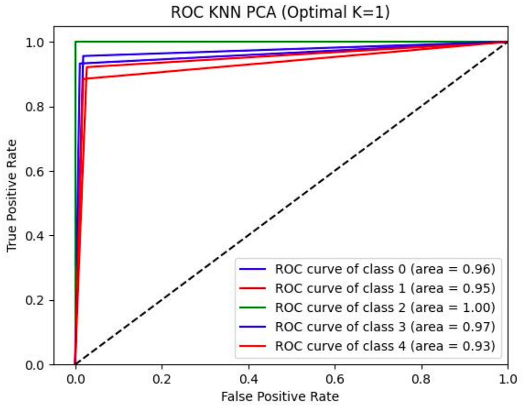


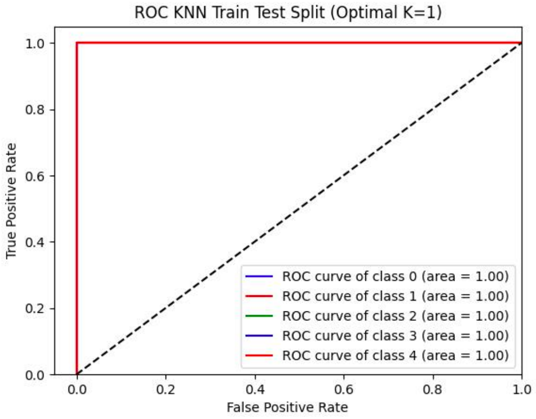


KNN ROC Curves

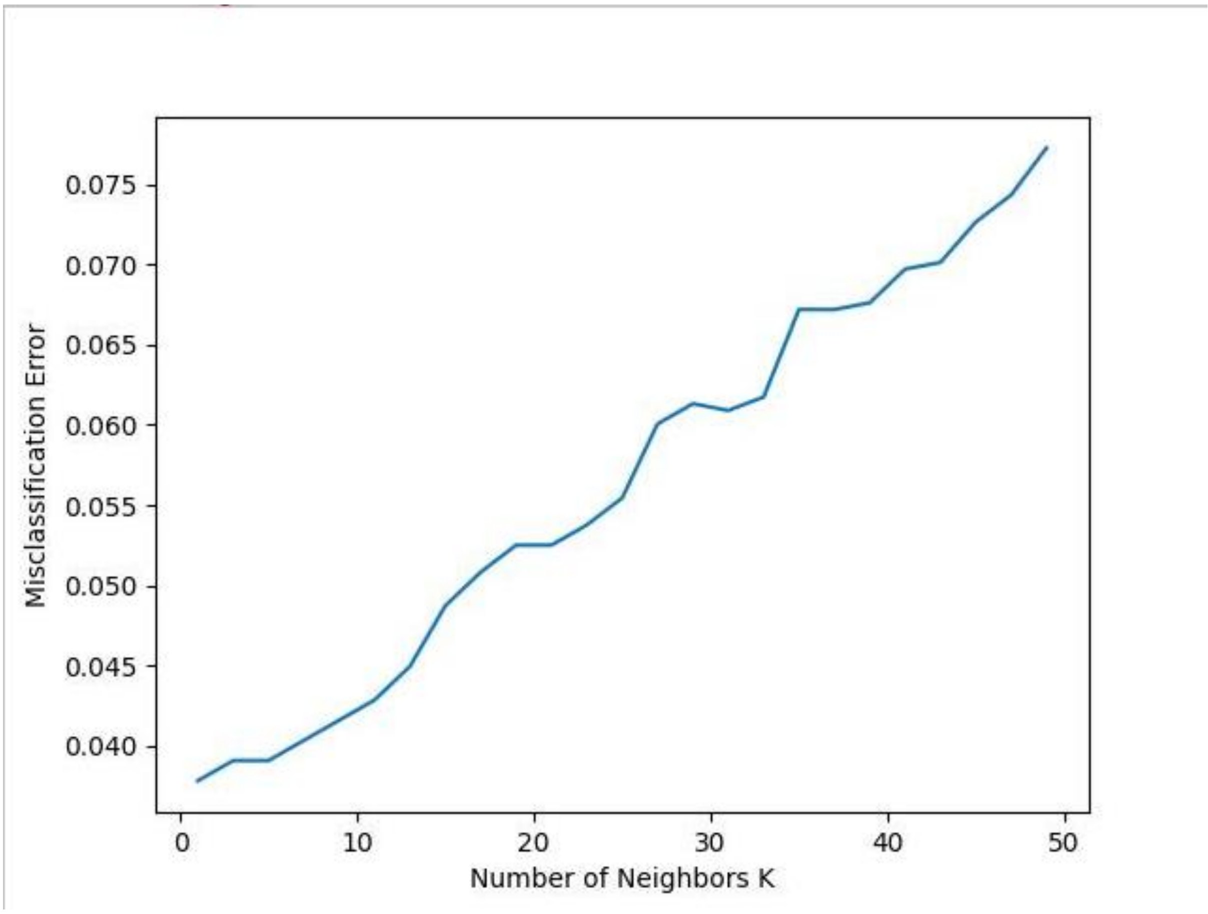








KNN for Different k values

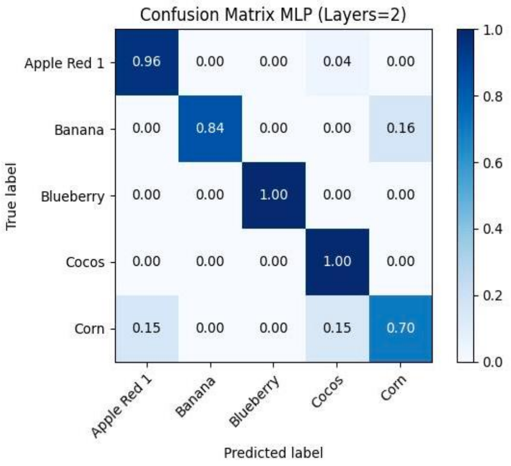


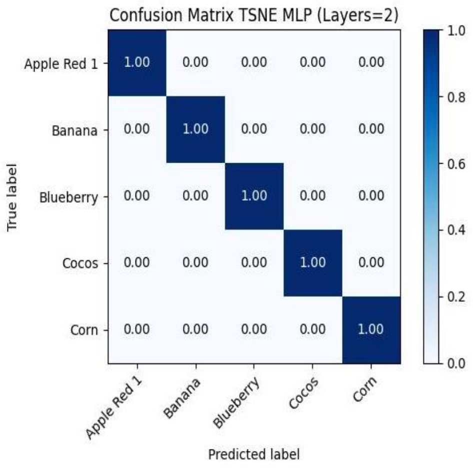
ARTIFICIAL NEURAL NETWORK

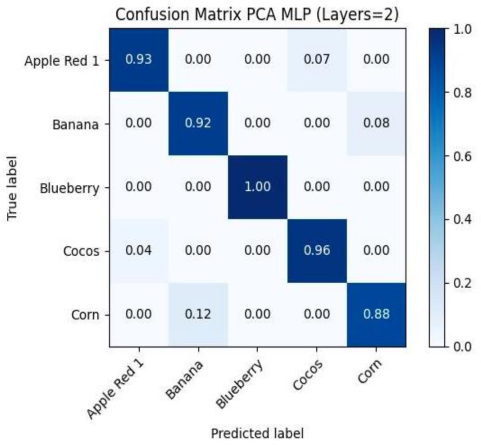
We have used 2 hidden layers for capturing the performance of Multilayer Perceptron.

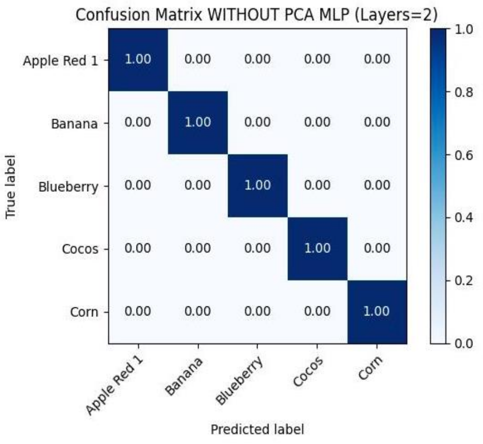
Accuracy without train and test split = 90.25%
Accuracy without TSNE and PCA = 99.8%
Accuracy with PCA(50) -> TSNE(2) = 93.65%
Accuracy with only PCA = 97.36%

Confusion Matrices



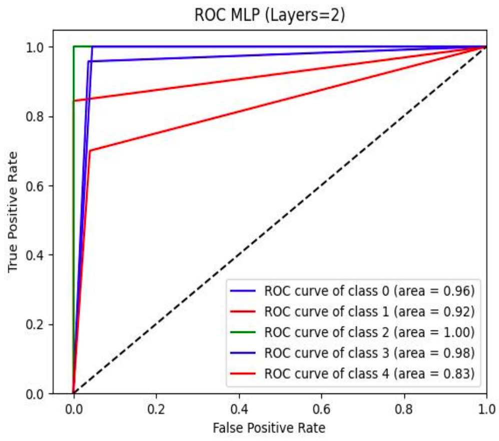


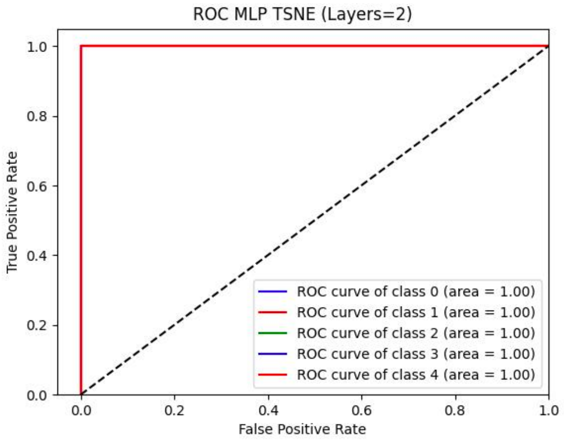


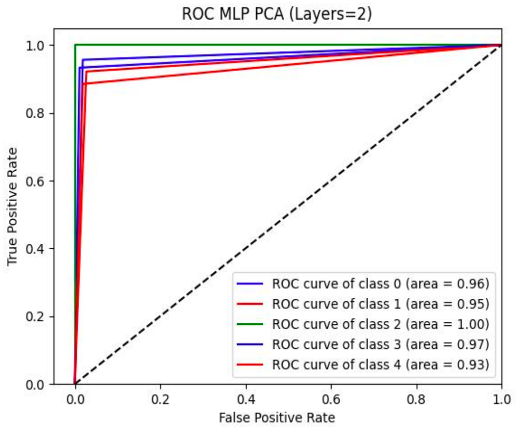


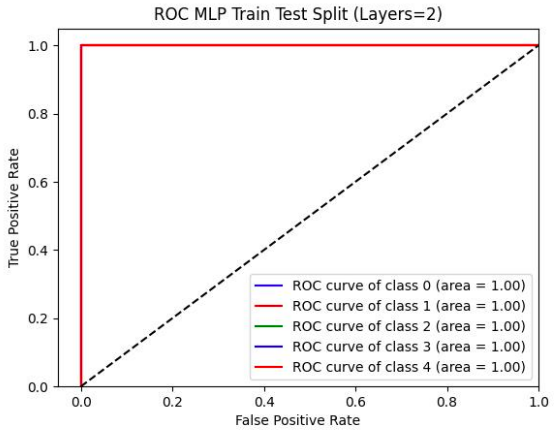
Artificial Neural Network

ROC curves







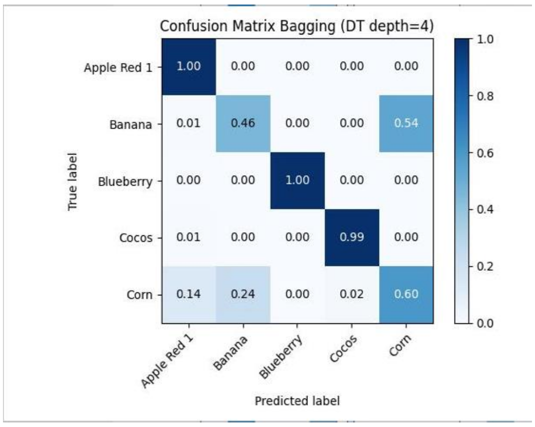


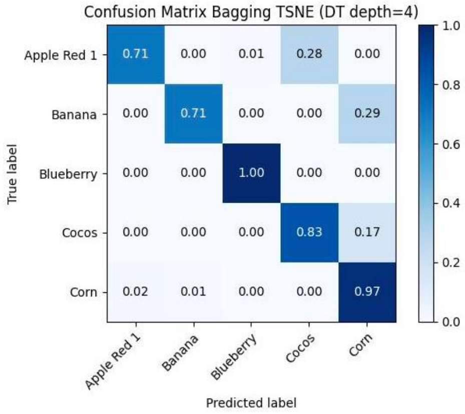
Bagging

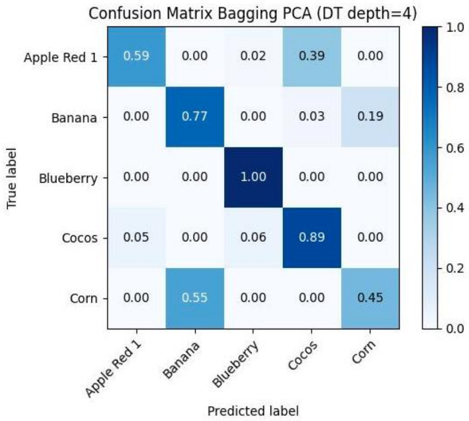
We ran bagging algorithm for Decision Tree at various depths on different values of n_estimators. The best depth comes out to be 4 for 15 estimators.

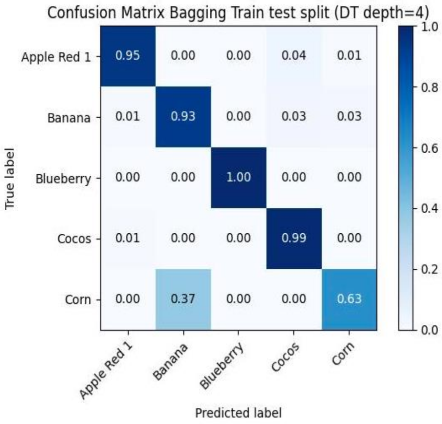
Accuracy without train and test split = 81.12%
Accuracy without TSNE and PCA = 80.87%
Accuracy with PCA(50) -> TSNE(2) = 90.27%
Accuracy with only PCA = 73.83%

Confusion Matrices







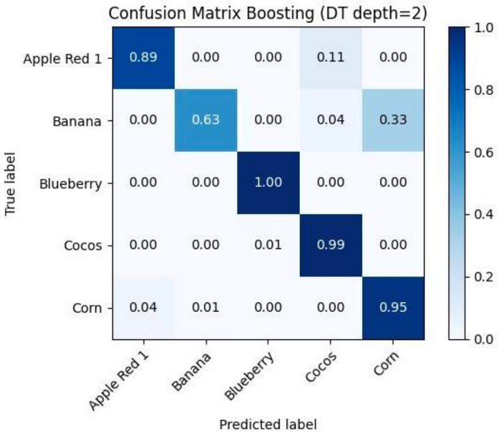


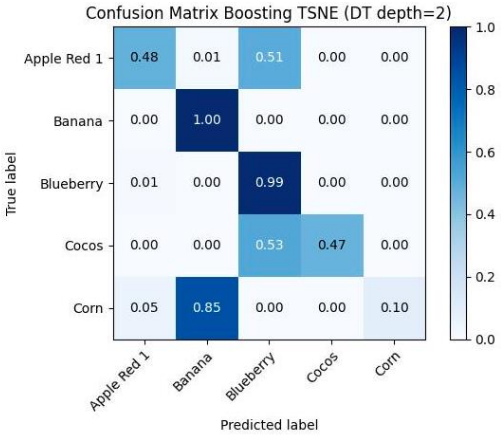
Boosting

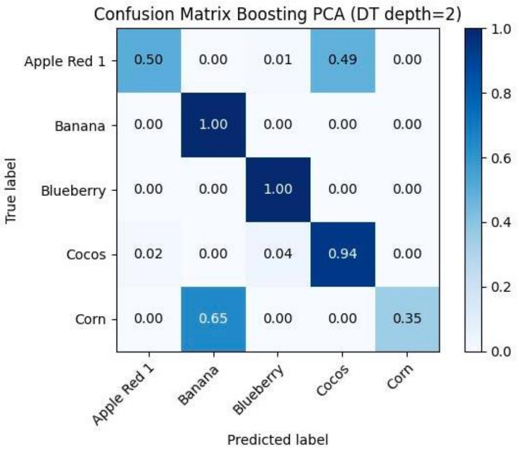
We ran boosting algorithm for Decision Tree at various depths on different values of n_estimators. The best depth comes out to be 2 for 15 estimators.

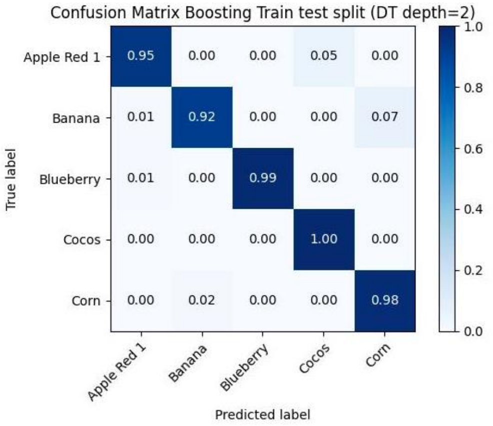
Accuracy without train and test split = 96.12%
Accuracy wth original data = 89%
Accuracy with PCA(50) -> TSNE(2) = 90.27%
Accuracy with only PCA = 73.83%

Confusion Matrices









Results of All Algorithms

Algorithm	Accuracy
Decision Tree - Scikit - d=3	82.4
Bagging - Scikit - d=4	95.6
Boosting - Scikit - d=2	87.6
SVM – Linear – C=1	88.4
SVM – Gaussian – C=10	88.0
KNN - k=1	85.6
Artificial Neural Network - HL=3	94.4

Result Analysis

1. Out of all algorithms, Bagging and ANN gives the best results if we use image data with all of it’s features i.e., 30k. This is because bagging decreases variance and retain the bias and our dataset contains less number of images due to which creating bootstraps and applying different estimators helped in reducing the misclassification.

2. KNN is the best when PCA or TSNE is applied is because it works on Euclidian distances and with PCA and TSNE we have easily segregated the data.

3. We were not expecting Decision tree to reach the accuracy of ANN but because of the limited training data with bagging and boosting even Decision tree can reach ANN accuracy.

4. Because of similarity of training and test data, most of the algorithms reached more than 95% accuracy when PCA and TSNE is applied.