Project 1

By-Usnik Singh Chawla

Data Loading and Pre-processing

The project is done on two types of data set

1. The first data set is created from DATA. Taking the first two images from each class and using this data for expression classification.
2. The second dataset used is the Pose data set.

For each dataset we flatten out the images to form a vector. After flattening out the images we divided in into test and train data sets. I have used a 80, 20 split for both the classification Task. After splitting the data, we perform dimensionality reduction to reduce the feature size. This was done in two ways.

1.PCA with reduced dimensions of () for TASK 1 and () of () for Task2.

2. MDA with reduced dimensions of () for TASK 1 and () of () for Task2.

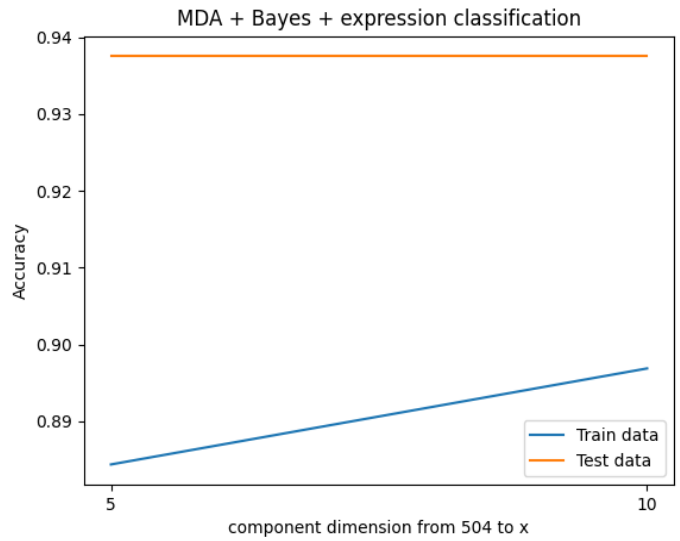
Classification

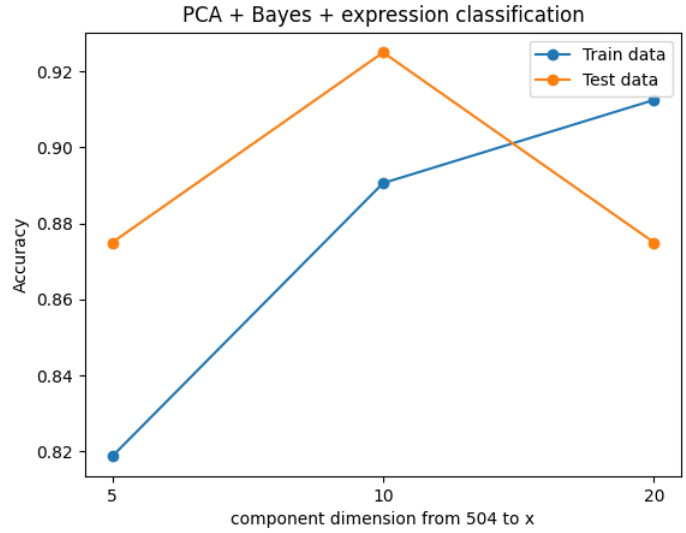
TASK 2 (Expression classification)

BAYES CLASSIFIER

With MDA, Bayes' classifier achieves a maximum of 93.5% test accuracy for components = 5 (and train accuracy of 89.30%). With PCA, the highest test accuracy of 92.5% is observed with components = 10 (85.5% train accuracy).

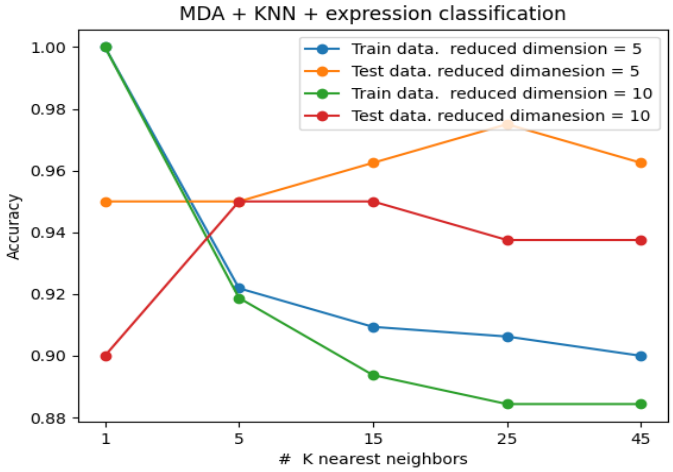
PCA requires much more components (=10) than MDA to achieve similar results.

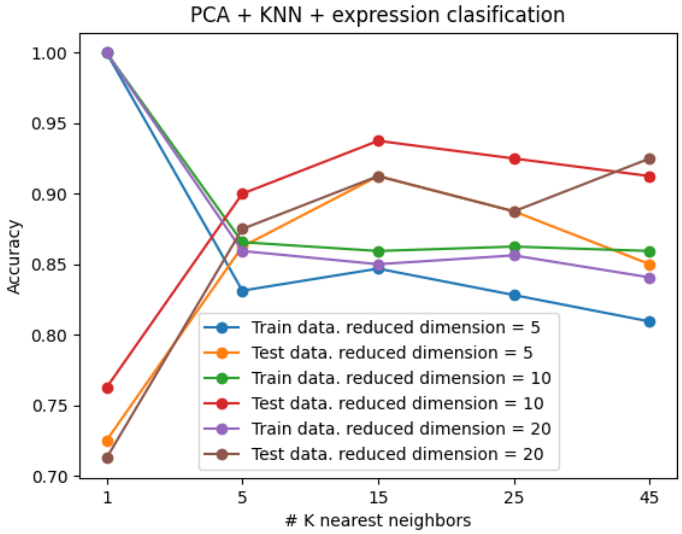




k-NN CLASSIFIER

I use five values of nearest neighbours (nn) 1, 5, 15, 25, and 45. With MDA, k-NN achieves a maximum test accuracy of 98% (nn = 25 and components = 5). With PCA, k-NN achieves a maximum test accuracy of 95% (nn = 15 and components = 10). This shows MDA is more efficient (requires lesser components) than PCA in this task.





Kernel SVM

Kernel SVM The kernel hyperparameters (sigma for RBF and r for polynomial kernel) for SVM is determined using 3-fold Monte Carlo cross-validation. The hyperparameters obtained from cross-validation is given below (test accuracies in parentheses):

MDA with components = 5 and Kernel SVM (RBF): sigma = 7 (96.5)
MDA with components = 10 and Kernel SVM (RBF): sigma = 6 (95)
MDA with components = 5 and Kernel SVM (Poly): r = 2 (77)

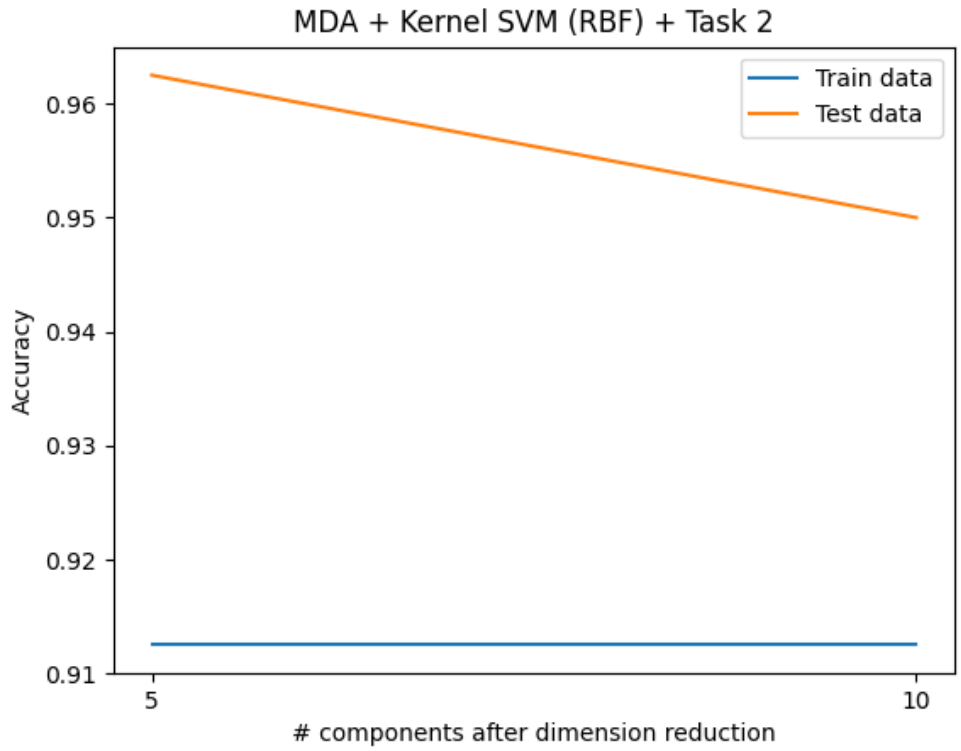
MDA with components = 10 and Kernel SVM (Poly): r = 2 (92)

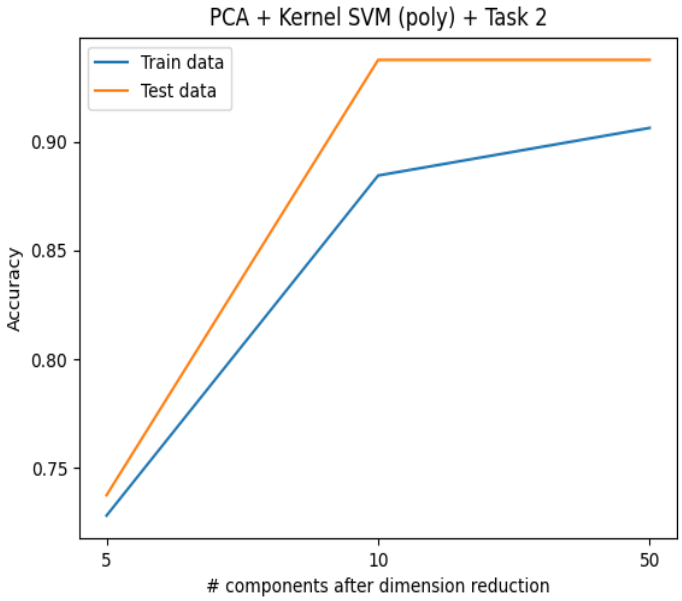
PCA with components = 5 and Kernel SVM (RBF): sigma = 13 (81.0)
PCA with components = 10 and Kernel SVM (RBF): sigma = 11 (90.0)
PCA with components = 50 and Kernel SVM (RBF): sigma = 16 (92.5)
PCA with components = 5 and Kernel SVM (poly): r = 11 (73)

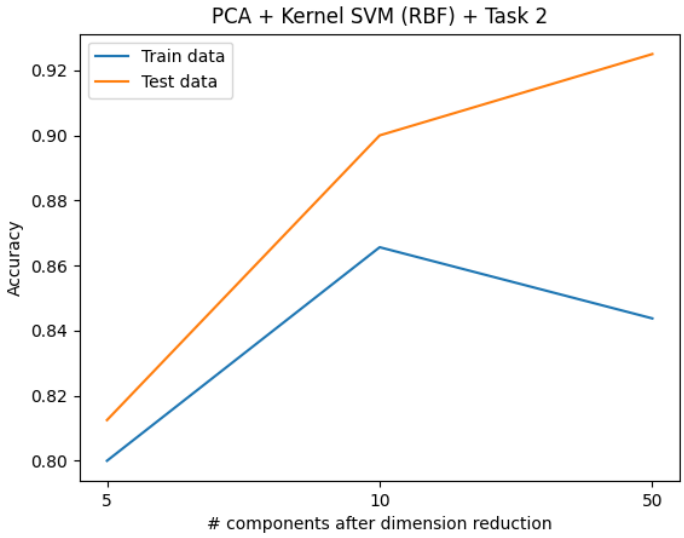
PCA with components = 10 and Kernel SVM (poly): r = 1 (93.75)

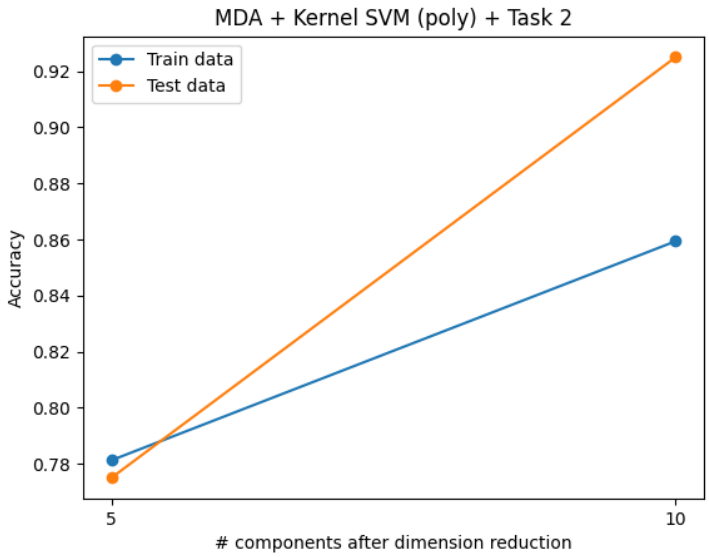
PCA with components = 50 and Kernel SVM (poly): r = 1 (93.75)

Kernel SVM with MDA (RBF) gives the best test set accuracy of 97.5% with components = 5. Kernel SVM with PCA gives the best accuracy of 93.75% with polynomial kernel and components = 10 and 50. Gradient descent is used for training the SVM with learning rate = 0.1 (trained until the error converges)




PCA + Kernel SVM (poly) + Task 2
Train data
Test data
0.90
0.85
0.80
0.75
Accuracy
5
10
50
components after dimension reduction


PCA + Kernel SVM (RBF) + Task 2
Train data
Test data
0.92
0.90
0.88
0.86
0.84
0.82
0.80
Accuracy
5
10
50
components after dimension reduction


MDA + Kernel SVM (poly) + Task 2
Train data
Test data
0.92
0.90
0.88
0.86
0.84
0.82
0.80
0.78
Accuracy
5
10
components after dimension reduction

Boosted SVM

Linear SVM is used as the weak classifier. With MDA, boosting very effective with component =10 as accuracy goes from 77 to almost 97 %. Boosting is not very effective for component =5. Also, boosting stops after two steps for component =5.

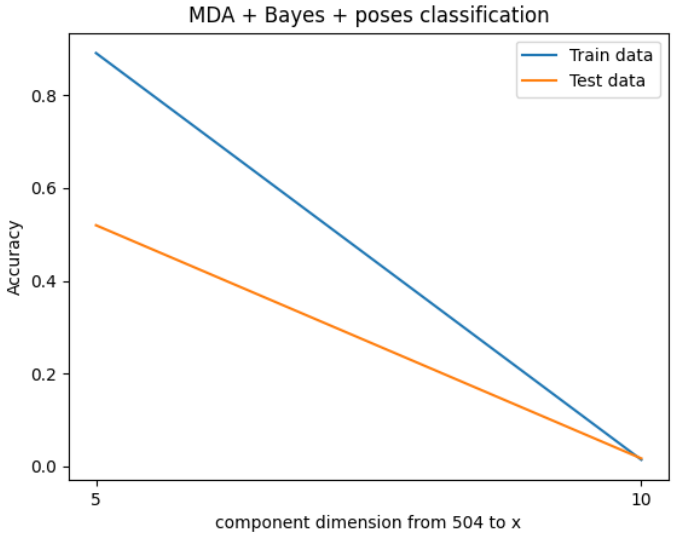
With PCA, boosting is improving the performance. The best test accuracy achieved is 94.5% with components = 10. The test accuracies improve by 9% using boosting for this case. For components = 5, there is an improvement of 18%, from 72% to 90% test accuracy with 4 steps of boosting. In general, the trend of the weights of the adaboost models reduces as the step increases.

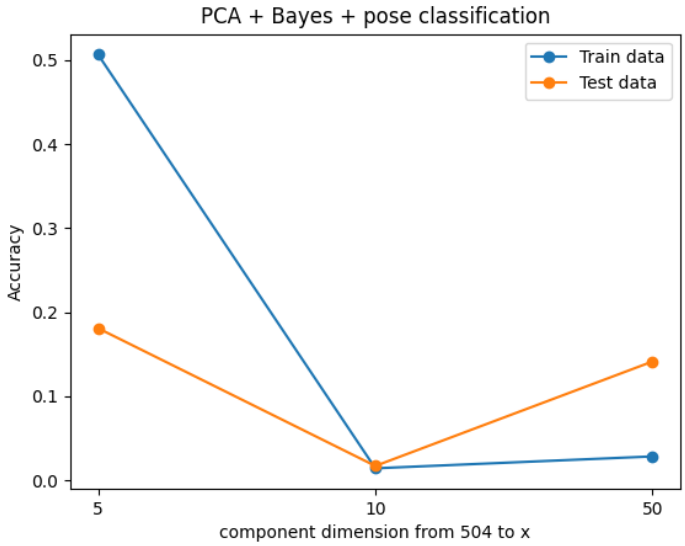
Gradient descent is used for training the SVM with learning rate = 0.1 (trained until the error converges).

TASK 1 (Pose classification)

BAYES CLASSIFIER

With MDA, the results are much more consistent than using PCA. With MDA, Bayes' classifier achieves 50% test accuracy with components = 5 (train accuracy = 90%). With PCA, Baye's classifier achieves a maximum of 19% test accuracy with components = 5. As components increase, accuracy does not improve. This might be since PCA is unsupervised and it is not providing sufficient information for classifying 68 classes.



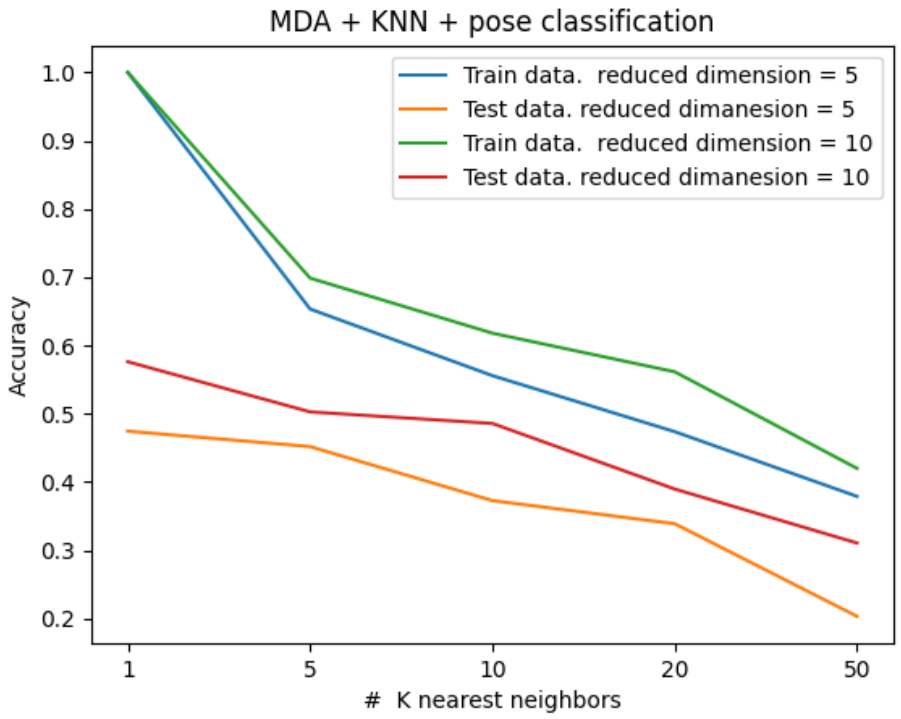


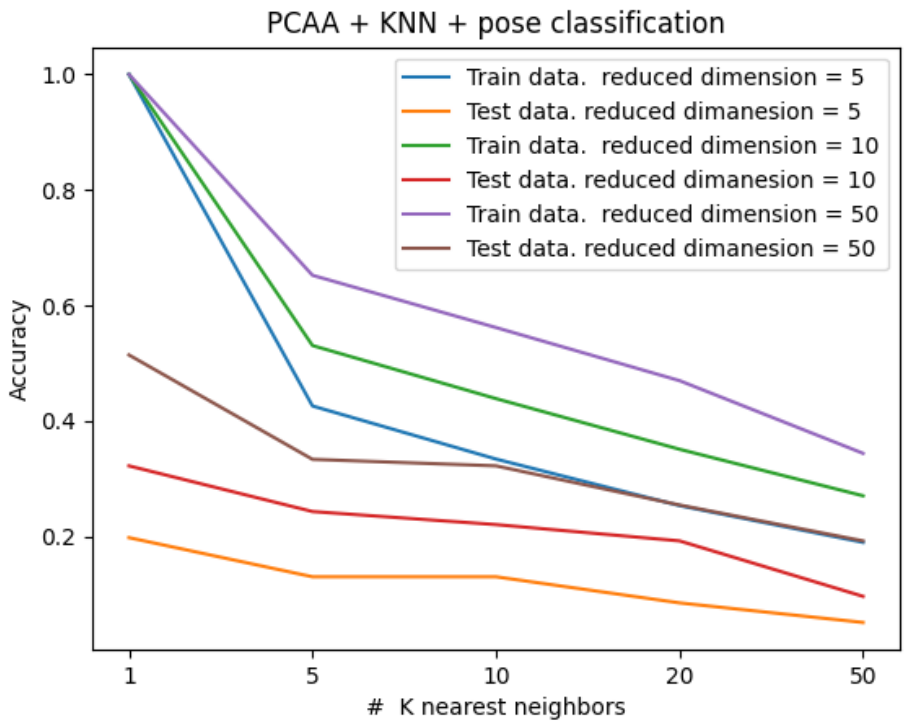
K-NN CLASSIFIER

I try four possibilities of nearest neighbours (nn) 1, 5, 10, and 20.

The best generalization on data is obtained when the classifier uses nn = 1. With MDA, k-NN achieves a maximum test accuracy of 60% (when nn = 1 and components = 10).

With PCA, k-NN achieves a maximum test accuracy of 55.36% (when nn = 1 and components = 50), 37% (when nn = 1 and components = 10).


MDA + KNN + pose classification
Train data. reduced dimension = 5
Test data. reduced dimanesion = 5
Train data. reduced dimension = 10
Test data. reduced dimanesion = 10
Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
1
5
10
20
50
K nearest neighbors


PCAA + KNN + pose classification
Train data. reduced dimension = 5
Test data. reduced dimanesion = 5
Train data. reduced dimension = 10
Test data. reduced dimanesion = 10
Train data. reduced dimension = 50
Test data. reduced dimanesion = 50
Accuracy
1.0
0.8
0.6
0.4
0.2
1
5
10
20
50
K nearest neighbors