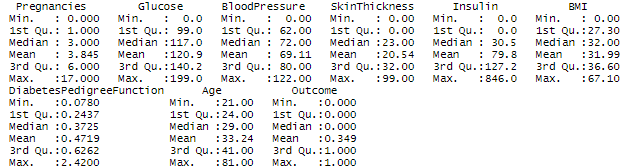
**The Dataset**

The Pima Indians Diabetes Database was used for comparison of NCA and PCA as dimensionality reduction preprocessing techniques. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Thus, it is a binary classification problem with the classes representing the infliction status (positive or negative – represented by 1 and 0 in the dataset) of the patient with respect to diabetes. The predictor variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on for a total of 8 predictor variables. The summary of the dataset is as follows:

****

The presence of 8 features makes it difficult to search for the relevance of the features for classification – thus, we aim to reduce the dimensionality of the problem to classify it easily as well as achieve better visualization of the problem in the transformed space. To that end, we implement the Neighborhood Components Analysis (NCA) and Principal Components Analysis (PCA) dimensionality reduction techniques.

**Tools Used**

We have used R for the assignment. The library(e1071) has been used to perform classification using SVM and library(class) is used to perform classification using KNN.

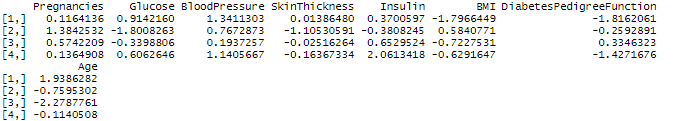
**NCA**

NCA is a supervised learning method for classifying multivariate data into distinct classes according to a given distance metric over the data. Functionally, it serves the same purposes as the K-nearest neighbors algorithm, and makes direct use of a related concept termed stochastic nearest neighbors. Since it is a supervised technique, it uses the dataset itself to learn the distance metric dynamically.

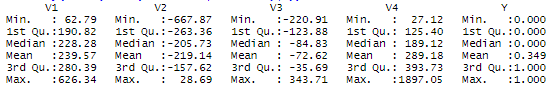
This technique aims at "learning" a distance metric from the dataset (instead of using a predetermined hyperparameter like K in KNN to apply Euclidean distance since it may not be important for classification) by finding a linear transformation of input data such that the average leave-one-out (LOO) classification performance is maximized in the transformed space. This is achieved by iteratively optimizing the performance function (defined by stochastic nearest neighbors chosen each iteration for each point using the probability of selection as softmax of the distances between the points) through gradient descent (regular, conjugate or stochastic) using the given dataset. We have used conjugate gradient descent in our implementation.

For initialization procedures, we first determine the dimensions to reduce to using other procedures on the dataset such as PCA. We find that reducing to 4 dimensions preserves ~99% of the variance in the data as shown in the next section, and testing the classifiers shows that any less dimensions produces a huge amount of false positives which are dangerous for this medical problem. Thus, we choose d=4. To initialize the transformation matrix, we have implemented three selectable procedures – identity matrix, mean adjusted diagonal matrix (from dataset) and random initialization. The results shown are with a seeded random initialization.

After applying the technique, we get the final transformation matrix as:



Note that the scale (chosen by initialization procedures) and direction (learned from the dataset) of the transformation matrix both are important in finding the ideal transformation space for effective class separation. The summary of the transformed dataset is as follows:



**PCA**

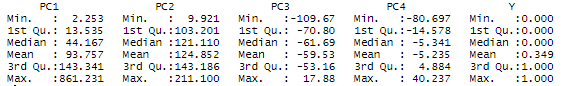
Since PCA is an unsupervised dimensionality reduction technique, we do not use the class labels in the procedure – instead the technique only uses the predictor variables and finds the directions that give, in order, the maximum variance (spread) among the data points. We transform the data by projecting the dataset onto a hyperplane formed by the selected number of directions (in order of maximum spread). The technique, being unsupervised, is only useful for classification if the directions of maximum variance are important for classification.

Applying the technique onto the dataset, we get the following summary with respect to the components:



As visible in the above, we can choose any number of components greater than two to have the transformed space maintain 95%+ of the total variance in the data. However, after testing the dataset with the classifiers based on the number of dimensions, the classifiers are able to perform with a much higher accuracy with d>=4, so we choose d=4 for classification purposes. (as an aside, this result also hints that the directions of maximum variance do not help much in separating the classes)

Using the chosen number of components, we transform the dataset. The summary of the dataset post-transformation is as follows:



**Classification**

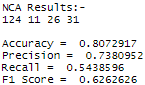
We apply two classification techniques to observe the effect of the dimensionality reduction techniques – Support Vector Machines (SVM) with linear kernel and K-Nearest Neighbors (KNN). We use 75% of the dataset for training (576 data points) and 25% for testing (192 data points). The summary of their results is as follows:

**1. SVM**

The confusion matrix for NCA transformed dataset is as follows:



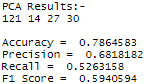
which gives us the following performance statistics:



Accordingly, the confusion matrix for PCA is as follows:



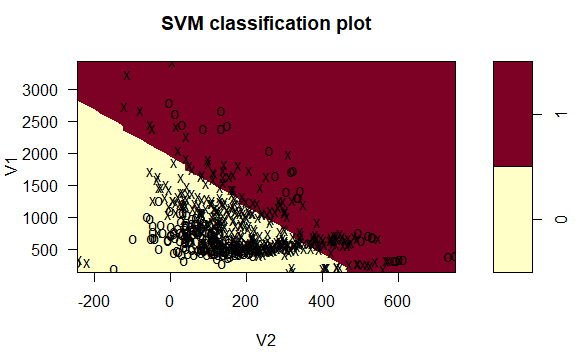
which gives the following performance statistics:

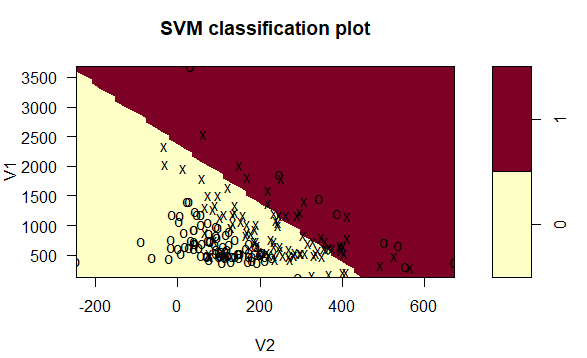


As we can see, the performance of SVM is slightly better on NCA than PCA with the chosen seed for train/test split. However, it is to be noted that with other seeds (or without a predetermined one for true randomness), there are cases where NCA performs similar to or, in some cases, even worse than PCA – indicating that the metric based NCA is unreliable as a supervised dimensionality reduction technique for classification if the classifier is not metric based since the learned metric ends up not being used for classification purposes.

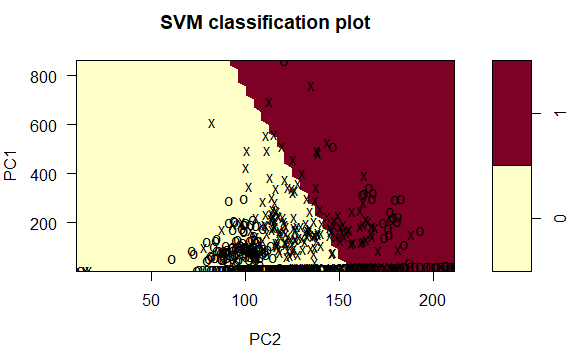
For visualization purposes, we have given the SVM plots between the first two components for the dataset for the two techniques below (first on training set then on testing set).

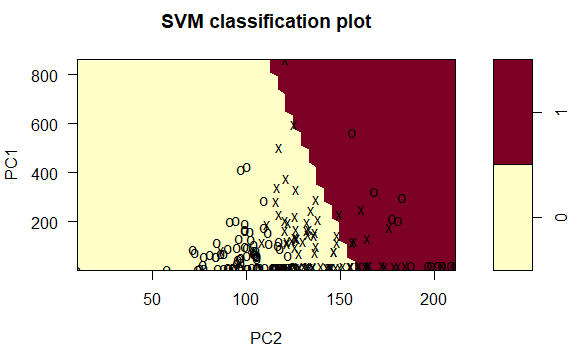
For NCA:





while for PCA:



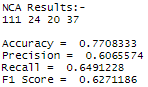


**2. KNN**

The confusion matrix for NCA transformed dataset is as follows:



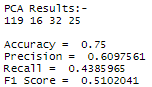
which gives us the following performance statistics:



Accordingly, the confusion matrix for PCA is as follows:



which gives the following performance statistics:



As we can see here, NCA performs much better compared to PCA (as indicated by the F1 Score and Accuracy). Testing with other seeds, the margin of difference between the two can vary, however, NCA almost always performs better than PCA as a pre-processing dimensionality reduction technique for KNN, which is reasonable since KNN is metric based and Euclidean distance in the transformed space is equivalent to the learned metric in NCA for class separation.

Conclusively, it is seen that NCA almost always performs better than PCA as a pre-processing dimensionality reduction technique for metric based classification techniques. For other classification techniques, however, it performs similarly to PCA.