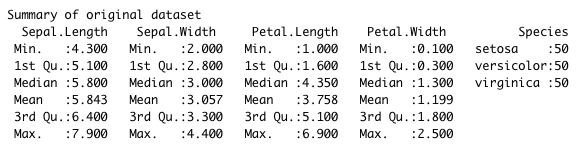
**The Dataset**

The Iris dataset was used for comparison between Fishers Linear Discriminant (FLD) and Principal Component Analysis (PCA). It is a multivariate dataset introduced by Ronald Fisher in 1936. The dataset is based on three different species of the Iris Flower, which form the three different classes in our problem – setosa, versicolor and virginica. There are 50 data points for each of the species forming a total of 150 entries in our dataset. Each data point has four different features namely Sepal length, Sepal width, Petal length and Petal width.

The summary of each of the features is given in the screenshot below.



The four features make it tough to visualize and classify, so our aim is to reduce the number of dimensions to make it easier to visualize and compute while keeping the loss of information as less as possible. For this we implement FLD and PCA and compare the performances of the two on the same Iris dataset. Finally, classification is done in both these techniques using Support Vector Machines (SVM) to see how well and with minimum loss of information was the dimensionality reduced.

**Tools Used**

We used R for the assignment. The library(e1071) has been used to perform classification and library(datasets) is used to import the Iris dataset. The library (MASS) is used for eigen vector and matrix inverse computations.

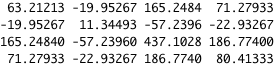
**FLD**

FLD is a supervised dimensionality reduction technique. Our goal is to project the given feature subspace onto a subspace with lesser dimensions without losing too much class-discriminatory information.

The aim of the projection in this technique is to separate the data in a least-squares sense. Since this is a supervised technique, we can make use of the information about which class each datapoint belongs to. We compute a hyperplane and the projection of each datapoint on this hyperplane will be our transformed dataset. The constraint is to ensure that the population of each class is best separated, that is the distance between the projections of each class on this hyperplane must be maximized. For this we consider the scatter within classes and the scatter between classes.

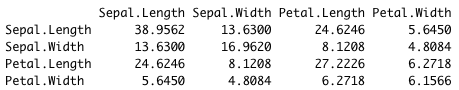
The scatter between-classes must be maximized since we want the projection of each class on the hyperplane to be as far apart as possible. This is also the variance between classes.

The scatter between-class matrix is the summation of products of variances of the mean of each feature from a particular class from the overall mean of that feature. The scatter between-class for the Iris data is as computed below.



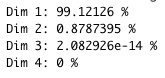
Since our aim is to maximize the distance between points of different classes, we need to cluster the points of the same class closer together. Hence the scatter within-classes, that is the variance within a class must be minimized.

The scatter within-class matrix is the summation of the products of variances of each datapoint from its respective mean for that feature in a particular class. The scatter within-class matrix for the Iris data is as computed below.



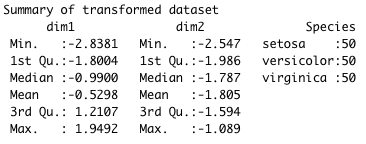
Hence we can say that the ratio of scatter between-classes to scatter within-classes must be maximized.

On differentiating this function to maximize it, we get the condition that our dependency is proportional to the product between the inverse of the scatter within-matrix and the scatter between-matrix. To estimate the linear discriminants, we find the eigen vectors to find the direction in which the maximum variance is shown in the dataset. On calculating this, we get the following contributions of each eigenvector in the variance of the Iris dataset.



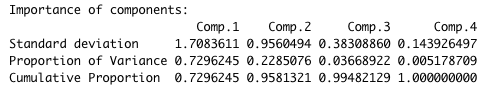
Observing these results, we note that dimension 3 and dimension 4 show a variance of almost zero.  
Hence we can reduce our dataset into two dimensions along the top two eigen vectors.

The summary of each feature in the newly transformed subspace is given below.

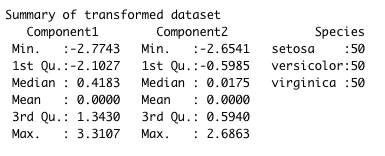


**PCA**

Since PCA is an unsupervised dimensionality reduction reduction technique, we do not make use of the various class labels and these categorical variables are dropped out before we perform PCA.  
Here we do not consider the scatter within or between classes as there is no clear distinction between the classes. To estimate the linear discriminants in this scenario, we standardize our dataset and find the direction in which the entire dataset shows the maximum variance. Eigenvectors are calculated accordingly and we get the following result.

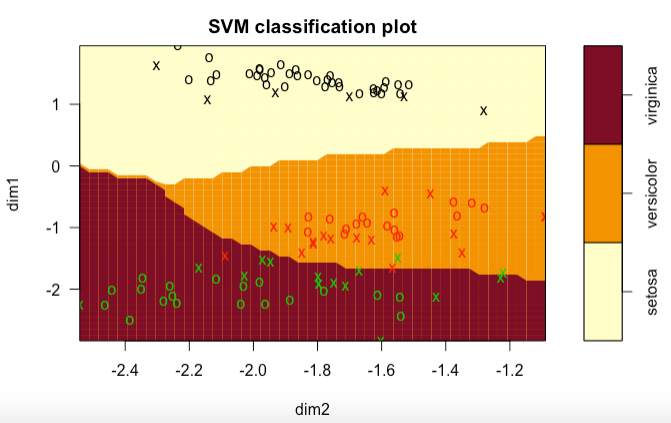


As we can see that the proportion of variance in Comp.1 and Comp2. make up for approximation 96% of the entire dataset, we can project our dataset onto these two dimensions to get the following new dataset.

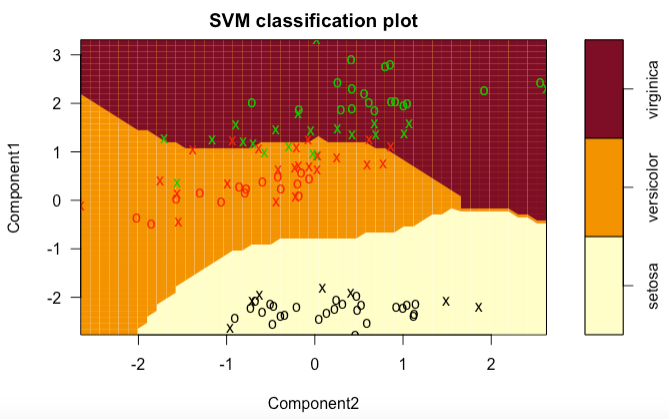


**Classification**

Once we have reduced the Iris dataset into two dimensions, we perform SVM to classify the dataset based on the classes. For this we have divide our dataset into 110 points for training and 40 points for testing.  
Now on comparing the results for FLD and PCA, we get the following results.

This is the SVM classification plot based on the training set for FLD. We are able to achieve a training accuracy of 97.27%.

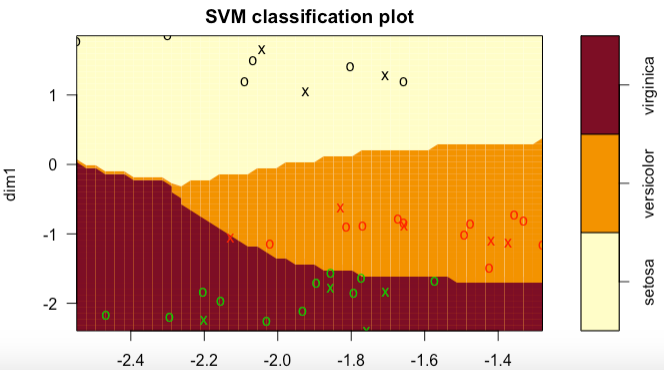
Below is the SVM classification plot based on the training set for PCA. We are able to achieve a training accuracy of 90.91%.

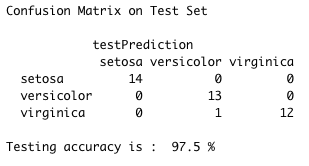


On testing our model over the test data, we can see that the supervised dimensionality reduction technique generalizes better than the non-supervised technique.

The following are the results of classification on the Iris data after FLD is performed.

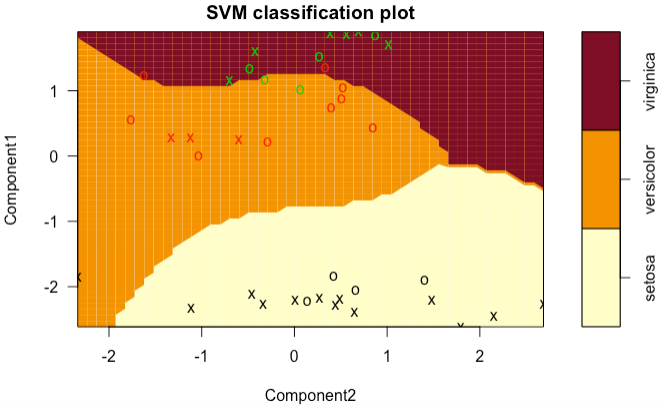
We get a generalization error of 2.5% in this model.

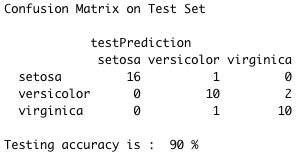




Below are the results of classification on the Iris data after PCA is performed.

We get a generalization error of 10% in this model.





By looking at the confusion matrices above we can observe that the number of true positives based on the testing set are much more in FLD than in PCA, hence we conclude that FLD generalizes better for the Iris dataset.