Assignment 3
Dimensionality Reduction And SVMs

Ujjwal Yadav
2018CSB1127

Abstract

This Assignment deals with processing a high dimensional data to reduce its dimensions ,so that it becomes easy for us to visualise and draw inferences from our data. Also dimensionality reduction reduction helps in eliminating redundant features and help us to get the most important features.

Here we employ Principal Component Analysis , learn about eigenfaces and eigenspace , employ Nearsest Neighbour Classifier , tSNE Algorithm , LDA Algorithm , SVM Classifier with Linear and Radial Basis Functions as Kernels to classify the Fisher Iris Dataset.

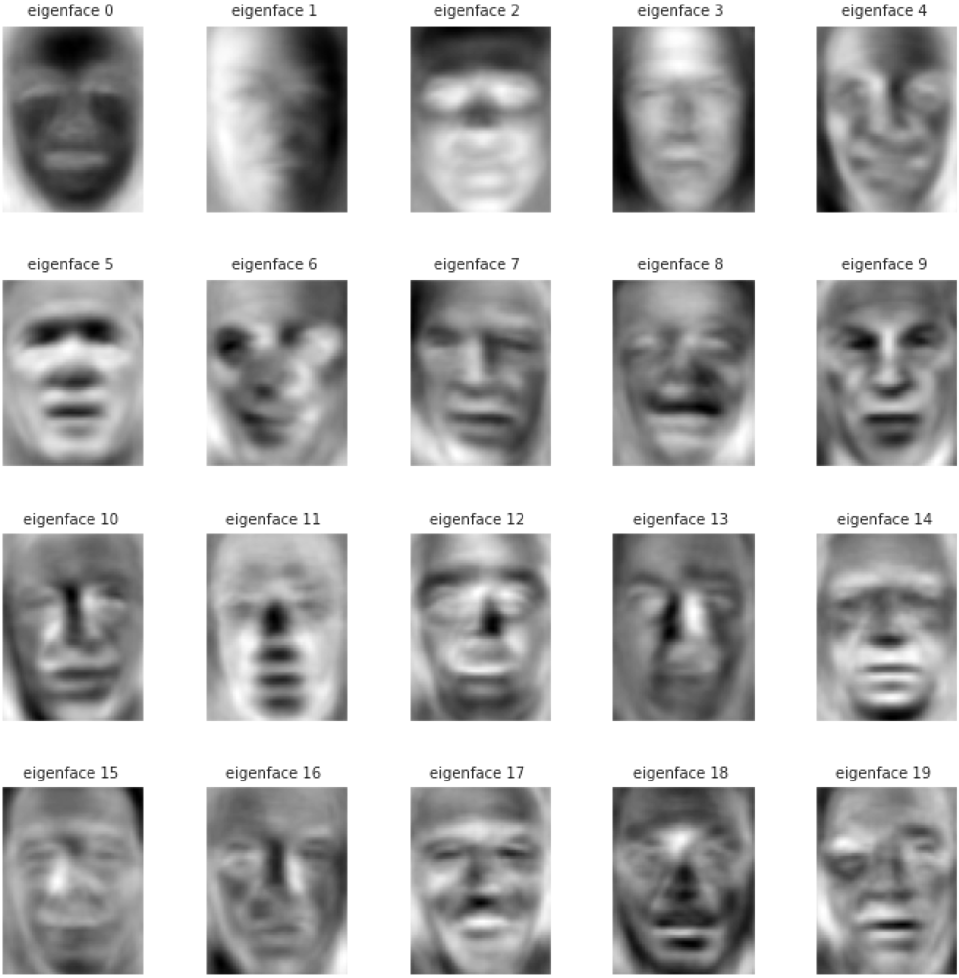
1 Introduction

Here we use two Datasets -
1) We use the a subset of "Labeled Faces in the Wild" dataset with a minimum of 100 faces per individual. The dataset we use contains multiple images of 5 persons. Each image is a 250*250 jpg image cropped , enlarged and scaled to uniform size to capture more of the head region.
2) The second Dataset we use is The Fisher Iris dataset which contains four features: SEPAL LENGTH,SEPAL WIDTH,PETAL LENGTH and PETAL WIDTH.There are 150 rows of data and there are 3 labels of flowers into which the data is classified.

1.1 TASK 1

Here we deal with the subset of Labelled Faces in the Wild Dataset. First we employ the Principal Component Analysis on the dataset to find the eigenvectors which explain the maximum variance of the data. Each 62*47 pixel image of dataset can be viewed as 2914 dimensional vector. There are multiple images in our dataset. Thus we employ the PCA with the number of components set to 100.PCA finds out the 100 eigen vectors which correspond to the direction of 100 directions which explain the maximum variance of the image dataset.
Subtask 1 and 4
We then transform every image of our dataset to this 100 dimensional space known as eigenspace and project every image onto the 100 eigen vectors. We plot the first 20 eigenfaces ,which are obtained by plotting each datapoint onto the 100 dimensional space and

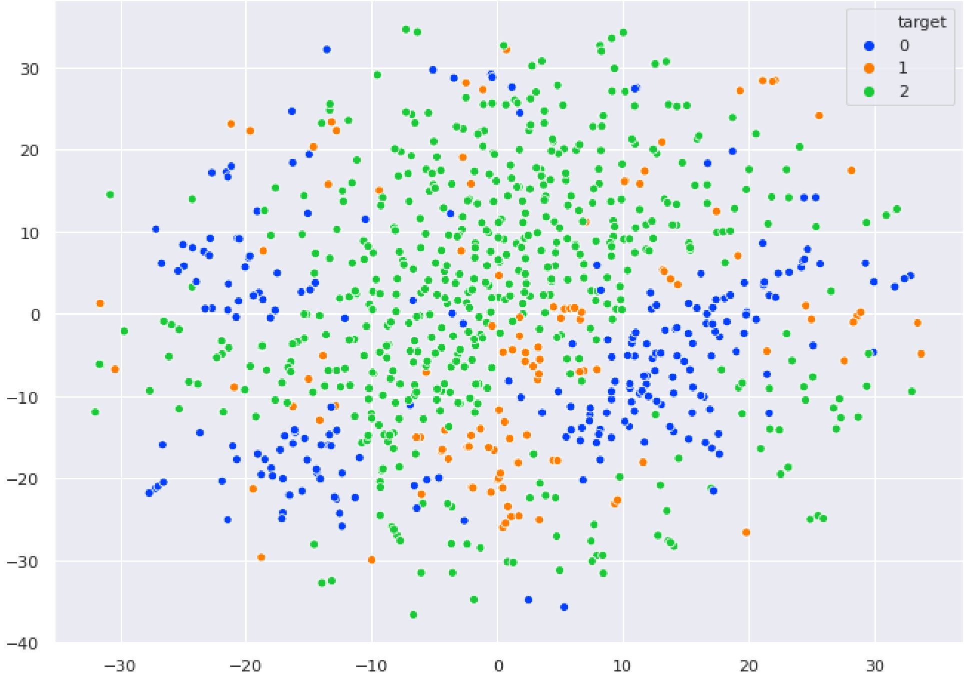
obtaining the projected images from the first 20 components of PCA or the first 20 Eigen-Vectors.
->The First 20 Eigen Faces are as shown in the figure:



Observations:
-> The PCA components capture the direction of maximum variance.
->The images of a particular eigenface has some Bright or White region , which are regions of maximum variance along that eigen vector, while the black shadow region indicates the region that has least variance along that eigen vector direction.
->Like in eigenface 2 , the eyes are very bright which indicates that there is maximun variance around the eye region along that eigen vector.
->Likewise along eigen vector 1 ,the forehead is very black indicating that the variance in forehead is almost negligible along eigen vector 1.

Subtask 2

We choose 3 personalities from the Transformed PCA dataset i.e , we take all images with targets marked as 0 ,1 and 2 corresponding to the following persons - i)Colin Powell ii)Donald Rumsfeld iii)George W Bush .We then plot all these datapoints of the 100 dimensional space to a 2-D Space with the help of tSNE algorithm.tSNE helps us to cluster points in a low dimensional space by measuring similarity between them in the high dimensional space. tSNE helps us in better visualisation of points by projecting the similar points as cluster in a low dimensional space as visualisation in high dimensional space is very difficult.tSNE assumes a gaussian distribution around every point in the high dimensional space and assigns probability densities to every point under that gaussian and renormalize for all the points. Then it similarly assigns the probabilities for different points in the low dimensional space using the Cauchy function . At last , tSNE tries to minimize the difference between the probabilities of the two dimensional space using the Kullback-Liebler divergence (KL Divergence) and Gradient Descent Algorithm.



Observations:

Our tSNE plot separates the data points into many clusters.Each cluster contains data points corresponding to images from a same person.

There is not a single cluster for each person ,i.e, the cluster contains data points of closely related images of a same person and the all the images from a same person does not fall into a single cluster , but into a number of distributed clusters, each with local similarities.

For example if the facial expressions of a person are very different in two images , then the two images of that person fall into different clusters. But as the facial expressions and characteristics of a same person will not differ much in most of the images , therefore 3-4 clusters one for each personality contains most of the data points corresponding to that person's image.

In the figure a most of the images from class 2 person form a big cluster in the middle of the plot , which signifies that the most of the images from class 2 person are closely related to each other and share almost similar range features values among them.

There is also a blue cluster with significant number of samples from class 0 indicating good amount of similarity between the different images from person 0, while the class 1 samples are scattered all around indicating the images from the same person vary widely from one another maybe in facial expressions and other features.

Subtask 3

We employ a Nearest Neighbour Classifier to train our model using the training dataset and then predict for the test dataset to classify the test dataset into different labels for different personalities.The algorithm behind Nearest Neighbour Classifier is very simple . We compare each data point in the test image with every data point in the training set and assign the label to the test data point same as the data point in the Training data which is closest to the the given instance for all the features that we are taking into consideration.

The Classification Report for Nearest Neighbour Classifier is:

	precision	recall	f1-score	support
Colin Powell	0.61	0.67	0.64	64
Donald Rumsfeld	0.38	0.41	0.39	29
George W Bush	0.76	0.79	0.77	161
Gerhard Schroeder	0.65	0.30	0.41	37
Tony Blair	0.48	0.51	0.50	51
accuracy			0.64	342
macro avg	0.57	0.54	0.54	342
weighted avg	0.64	0.64	0.63	342

Observations:

->The accuracy for Nearest Neighbour classifier is around 64%.

-> The persons for which Precision and Recall are high as compared to others like for George W Bush indicate that they are well distinguishable from others and share less similarities with other.

->While for people like Donald Rumsfeld , who have less precision and recall indicate that they have many features similar to the other people and are less different from others.

Subtask 5

-> Here we do not fix the number of PCA components but we fix the amount of variance of the data that we want our PCA algorithm to capture.

-> In our case we fix the variance to be equal to 80%.

->On finding the number of components that the PCA use to capture the 80% data , we find that it use **31** components to capture 80% variance of the Data.

->We again apply the Nearest Neighbour Classifier on this dataset which captures 80% variance of data and has data transformed to 31 features.

-> The accuracy is also somewhat low because the 100 components of PCA explain around 93-94% of the variation in data , and do not account for the remaining 6-7% of variation in data.

-> Classification Report of this model is :

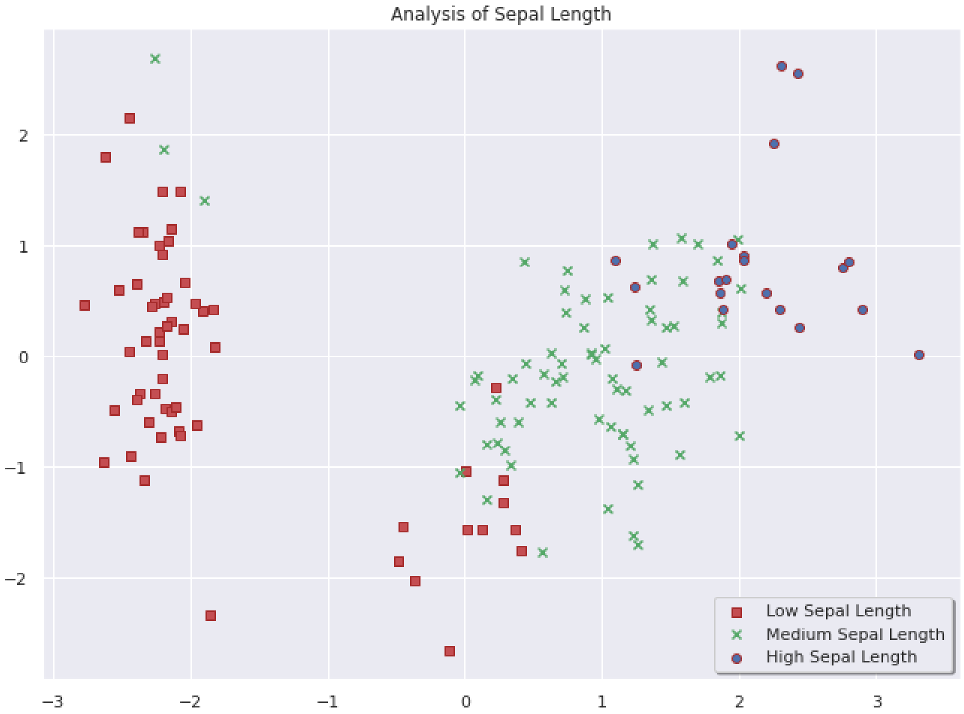
	precision	recall	f1-score	support
Colin Powell	0.49	0.66	0.56	64
Donald Rumsfeld	0.39	0.41	0.40	29
George W Bush	0.73	0.71	0.72	161
Gerhard Schroeder	0.56	0.27	0.36	37
Tony Blair	0.46	0.45	0.46	51
accuracy			0.59	342
macro avg	0.52	0.50	0.50	342
weighted avg	0.59	0.59	0.58	342

-> Some observations regarding Classification Report of PCA with 100 components and classificatio report of PCA which covers about 80% of variance of data are as follows:
-> The accuracy of PCA with 100 components is more than the PCA which explains around 80% of Variance. For first model it is around 65% , while for the second model it is around 60% .This is because variance explained by 100 pca components is around 93% which is much more than 80% explained by the other model. Thus 100 PCA components explain and predict the data much more accurately than 31 PCA components which explain around 80% of the variance in the data.

1.2 Task 2

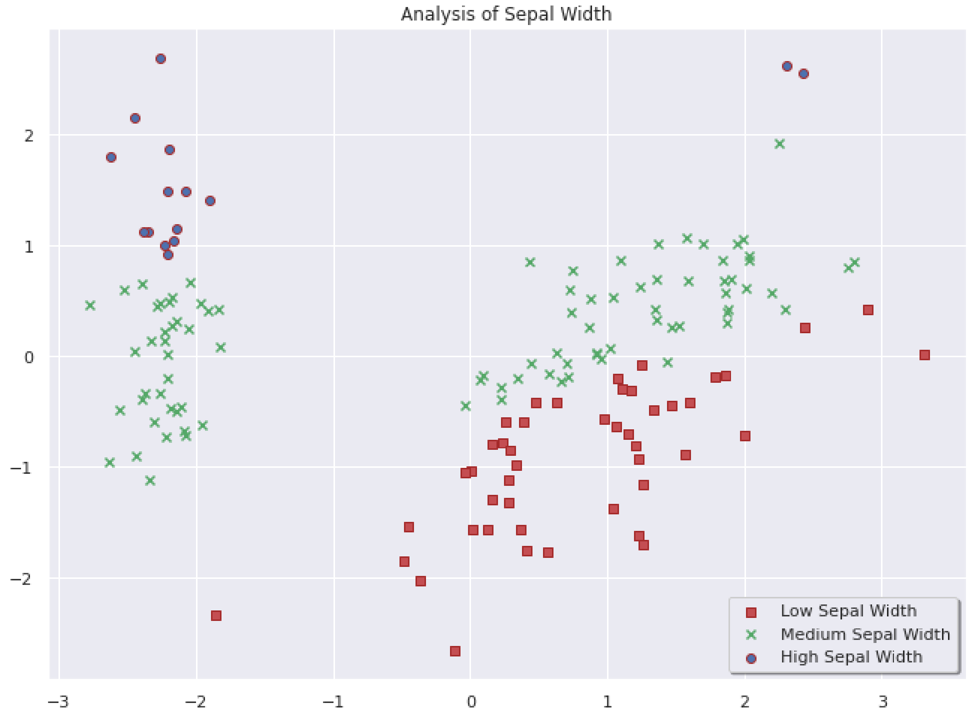
Subtask 1

In this task we deal with the Fisher Iris dataset . The Fisher Iris dataset contains 150 samples of data with each data point being classified into one of the 3 flower types. First we employ the PCA on the dataset to reduce it from 4 dimensions to 2 dimensions of maximum variance. We then plot the data onto a 2D space by dividing each of the features into three ranges like low, medium and high and highlighting the distribution of the different ranges point with the help of different colors.We do it for every feature and infer our observations along both the Principal Component 1 and Principal Component 2.
->The distribution of sepal length along the two eigenvectors is highlighted by the following figure:

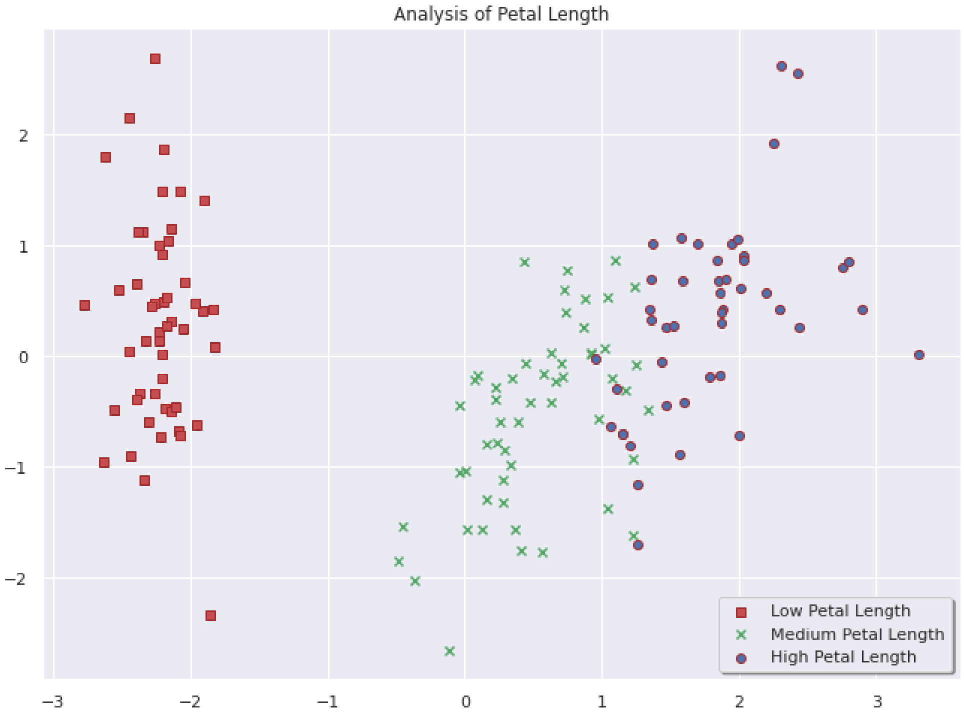


-> High Values along PCA Component 1 reflects high Sepal Length as is reflected by the plot.
-> Thus as Coordinate along PCA Component 1 increases , it implies increased Sepal Length.
-> Along the PCA Component 2 ,most of the data points are concentrated in the middle range of the PCA Component 2 , irrespective of the Sepal Length.While low values along PCA comp2 in general implies low to medium Sepal Length , while high values along the Comp2 reflect mostly the data points with either low or high Sepal Length.
->But most of the data points , irrespective of the Sepal Length lie in the mid range along the PCA Comp2 .

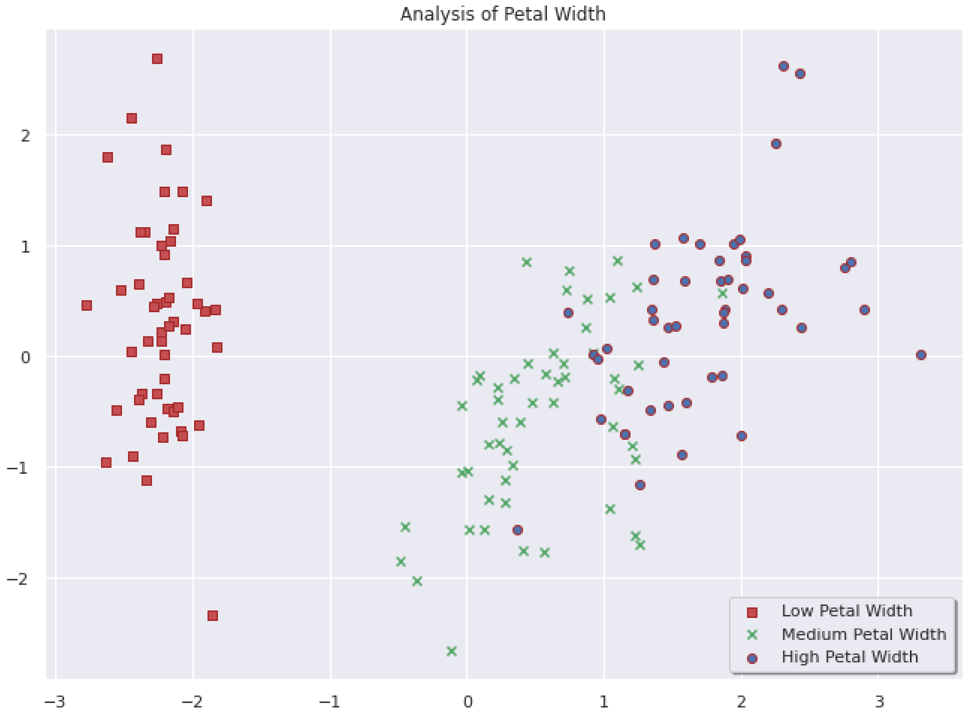
->The distribution of sepal width along the two eigen vectors is highlighted by the following figure:



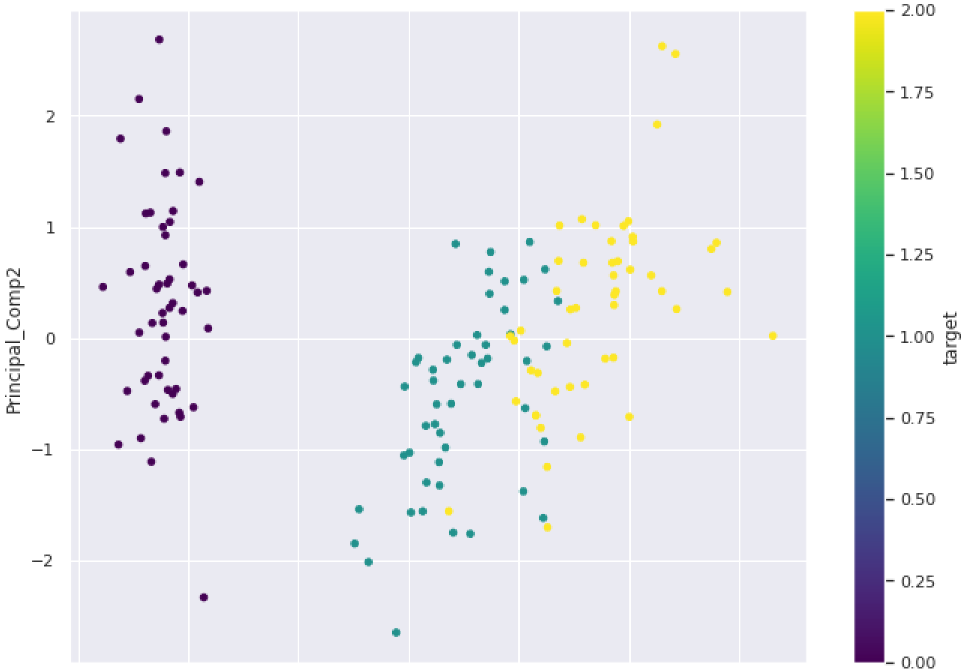
->Lower values along PCA Comp1 in general implies High to Medium Sepal Width , while Medium Coordinates along Comp1 reflect Medium to Low Sepal Width.
-> High Comp1 values reflect data points with Medium to Low Sepal Width
-> The Distribution of Sepal Width along PCA Comp2 is more easy to infer.As PCA comp2 values of data points increases , the Sepal Width in general increases with increase in Comp2 values.
->Thus High Values along eigen direction -2 corresponds to high Sepal Width. ->The distribution of petal length along the two eigen vectors is highlighted by the following figure:



->High values along Eigen Direction 1 implies high values of Petal Length of Data points , which can be inferred easily from the graph.
->Most of data points irrespective of the Petal Length lie in mid region of Eigen Direction 2, i.e most of the data points lie within the medium eigen Direction 2 values.
->Points lying within the Low eigen direction 2 range mostly correspond to Medium Petal Length , while high values along eigen direction 2 corresponds to mostly Low Petal Length.
->The distribution of petal width along the two eigen vectors is highlighted by the following figure:



-> The distribution of Petal Width along Eigen Direction 1 and 2 is almost same as that of Petal Length.
-> As we move higher along Eigen Direction 1 the Petal Width increases .
-> While most of the data points lie within the medium range of the Values along Eigen Direction 2.Small Values along Eigen Direction 2 mostly corresponds to Medium Petal Width ,while high values corresponds to Low Petal Width.
-> The plot of the data points along with labels along the Principal Components is shown below:



Subtask 2

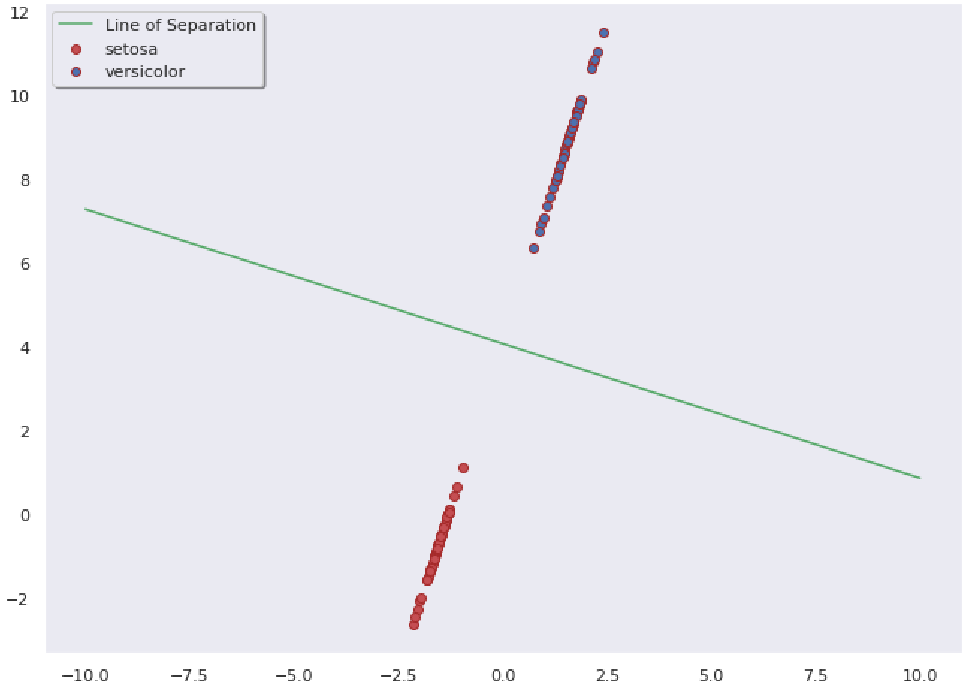
Here we learn about the LDA algorithm. LDA stands for Linear Discriminant Analysis. LDA is a dimension reduction technique which is used to find the new axis to project data in such a way so as to minimise the within class variance and to maximize the Interclass Means , so that we can easily cluster the data points from different classes into unique clusters.

In this subtask, we select the two classes at a time from the 3 classes of the Iris Dataset. Now LDA will project the chosen datapoints onto a line (1-D line) as the number of classes we took at a time are just two and LDA generally projects the 'C' classes data onto a C-1 Dimensional Space in a way to maximize the distance between the means of the 2 classes on the projected line and to minimise the Intra Class Variance.The 3 plots are shown below along with the line of Separation which separates the data into 2 parts either class data points lying on either side of the line.

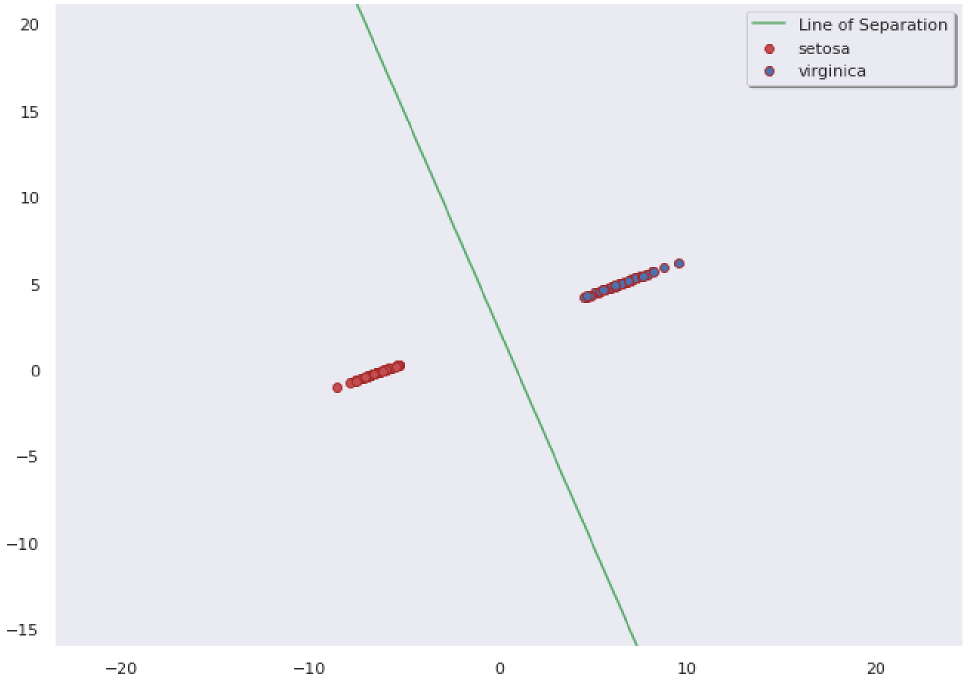
LDA generally works well for data which has Gaussian distribution for every class data points with minimum overlap between any two Gaussians from different classes. If the data of two or more classes is overlapping then LDA may fail to find trhe optimal solution.

-> The three plots of the 3 pair of classes taken 2 at a time , with the points projected on a 1-D line along with the line of separation are shown below:

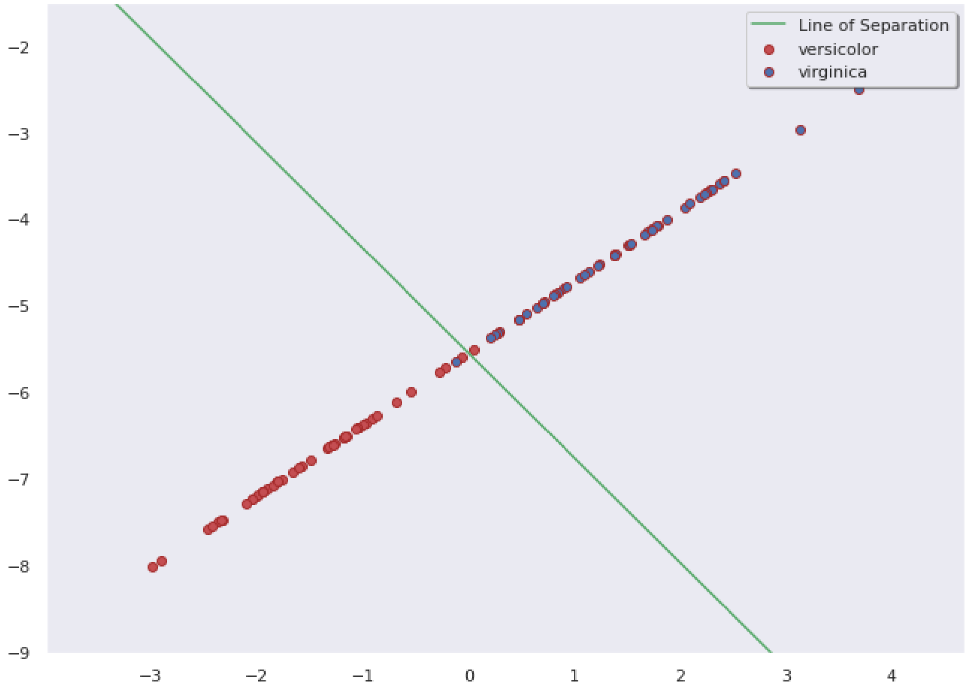
-> The plot for Class 0 and Class 1 is:



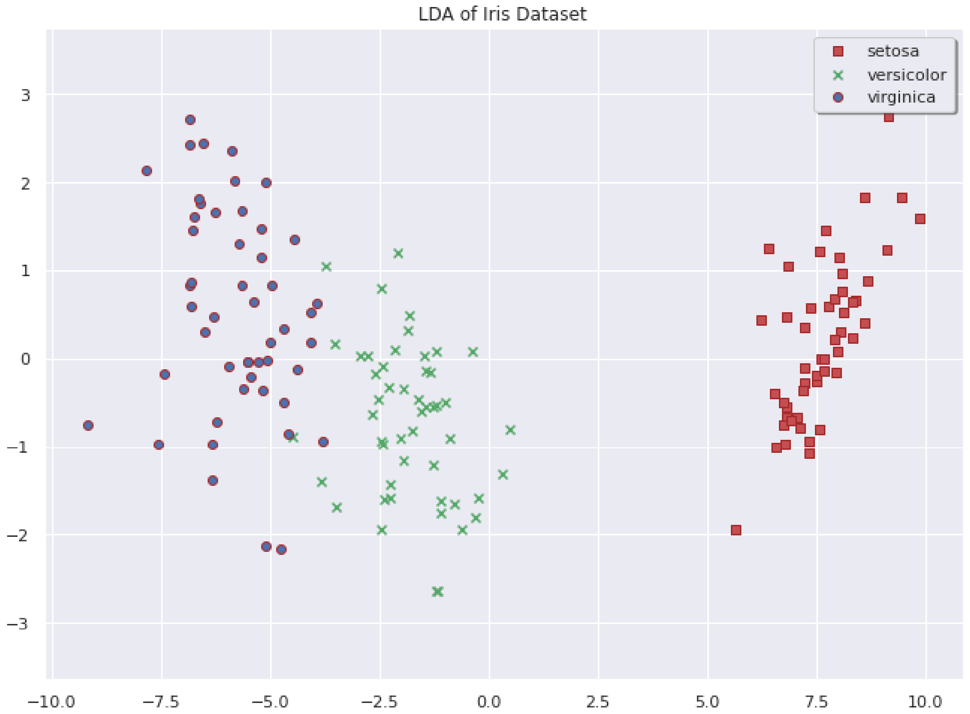
-> The plot for Class 0 and Class 2 is:



-> The plot for Class 1 and Class 2 is:



-> Finally the plot for 3 class iris data projected on the 2-D space to maximize inter-class scatter while minimizing intra class variance is plotted by applying Linear Discriminant Analysis on the Fisher Iris dataset.The LDA finds the suitable direction which maximizes the inter-class scatter and minimizes the intra- class variance.
-> That suitable direction is made the X-axis while the direction orthogonal or perpendicular to that direction is made the Y-axis. Then we project the data points along those two directions and plot them on a 2-D plot.
-> The figure showing the above described LDA plot on Iris dataset is:



Observations:
->We see that the LDA finds an appropriate direction such that the samples of the two classes when projected onto that direction ensure separate clusters , while maximising inter-class scatter and minimizing intra-class variance.
->We also observe that the separating line will be perpendicular to direction of projection of points.

Subtask 3

Here we project the Fisher Iris Dataset onto a 2-D and 3-D space by using the tSNE algorithm . tSNE algoritm as explained above transforms a higher Dimensional data into a lower dimensional data by preserving local similarities. Here we analyse the tSNE performances for various **Metrices like 'Euclidean Distance' , 'Manhattan Distance' , 'Minkowski Distance (p)',etc**
We observe the clustering for each plot with the help of two different types of Distance Metrices and infer our obsevations by comparing the effectiveness and preciseness of the model . First let us introduce ourselves to different Distance Metrices :
-> The two metrices we used are : 1) **Euclidian Metrice** and 2)**Mahalanobis metrice**
Euclidean Metrice:
Euclidean Distance is the ordinary straight line distance between two points in the Euclidean Space.
Let us consider two points X and Y in an Euclidean Space with n dimensions.Then the dis-

tance between them is given by the formula:

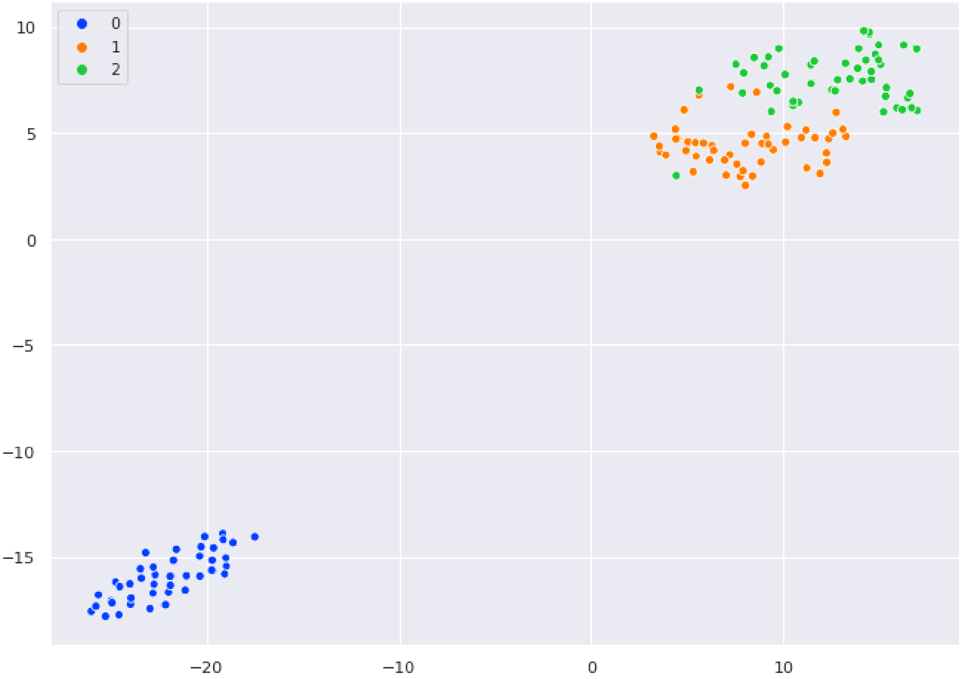
$$\sqrt[2]{\sum_{i=1}^{n}((x_i)^2-(y_i)^2)} \tag{1}$$

Mahalanobis metrice

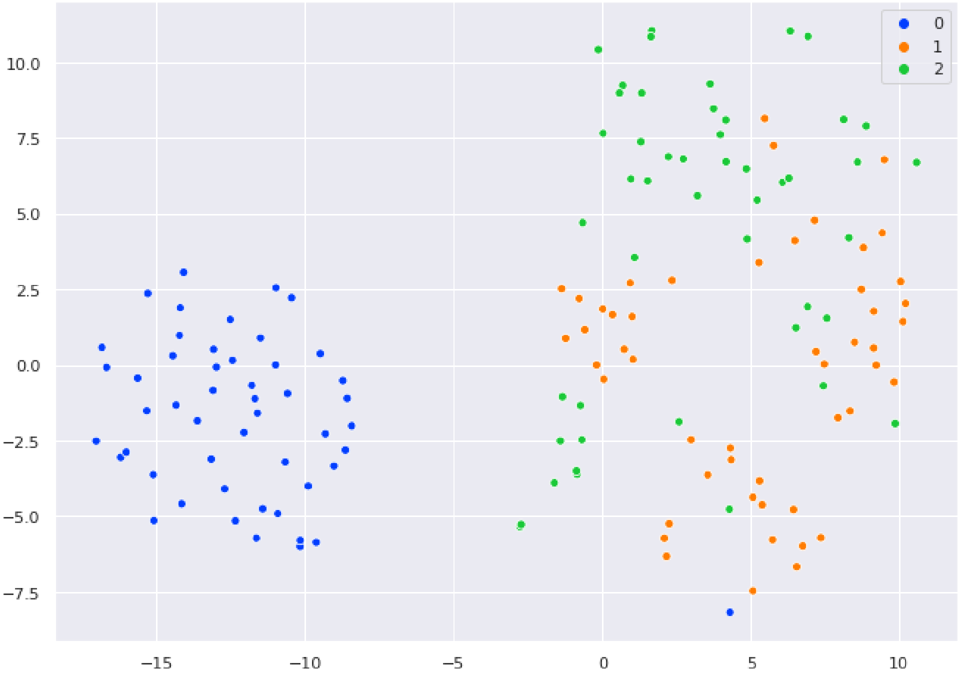
It is the distance between a Vector X and a Vector Y in multidimensional space ,with let us say n dimensions, with the covariance matrix of the distribution given by S.It measures sort of dissimilarity between the vectors X and Vector Y in the same space with S being the Covariance Matrix of the distribution.It's formula is given as:

$$d(X^{->},Y^{->}) = \sqrt[2]{(X^{->}-Y^{->})^T S^{-1}(X^{->}-Y^{->})} \tag{2}$$

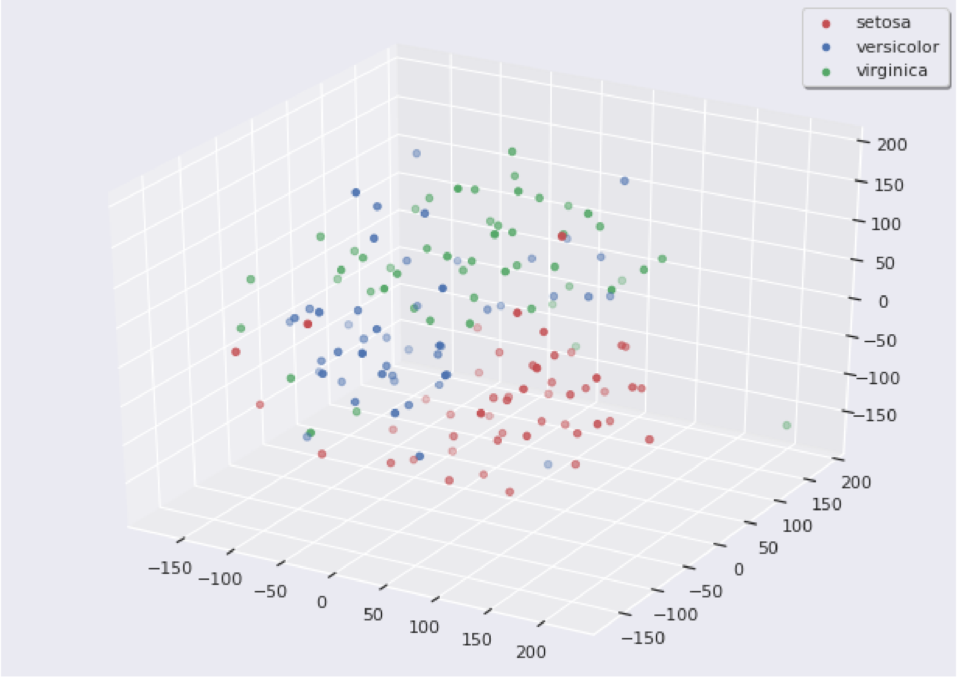
-> 2-D tSNE plot for Iris dataset with metric parameter as **'Euclidian'** is shown below:



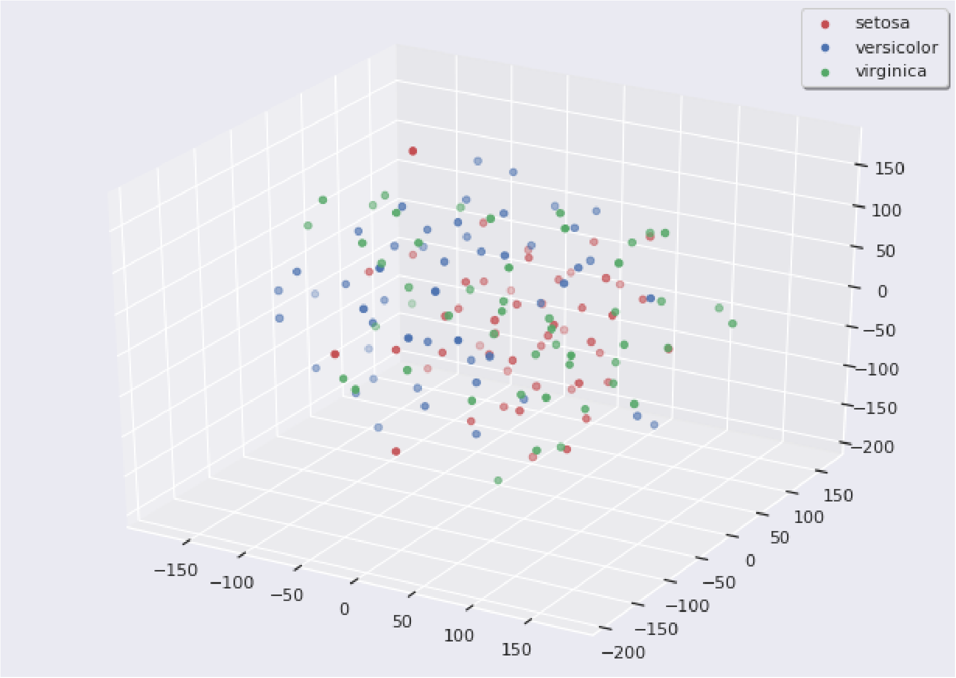
-> 2-D tSNE plot for Iris dataset with metric parameter as **'Mahalanobis'** is shown below:



-> 3-D tSNE plot for Iris Dataset with metric parameter as **'Euclidian'** is shown below:



-> 3-D tSNE plot for Iris dataset with metric parameter as **'Mahalanobis'** is shown below:



Observations:

-> Using Euclidean Metric , forms three distict and separable clusters both in 2-D and 3-D plots as can be seen by tSNE plots.tSNE is able to cluster three classes distinctly by using Euclidean Distance as metric parameter , thus Euclidean Distance preserves local similarities in the data points belonging to each of the three types of flowers.This is because as most of the samples belonging to a same class have values of the different features in the same range.
->While using the Mahalanobis metric as distance , we tend to find the class which is different from the other two classes. That is we tend to find that how far away or different is a set of points from the rest of the distribution.
->In our plot of 2-D and 3-D we find that the class 0 is quite different from the class 1 and class 2 as can be seen from the 2-D and 3-D plot of the samples using tSNE.tSNE preserves local similarity , and the given plot using Mahalanobis metric goes onto show that the features values of class 1 and class 2 samples differ from one another , but do not differ to that extent as class 0 samples differ from the rest of the class samples.

1.3 Task 3

Subtask 1 and Subtask 2

SVM stands for Support Vector Machine.SVM is a supervised learning algorithm , which helps us in classification of two or more class dataset by drawing decision boundaries in a n-dimensional space , where n is the number of features of the dataset. It finds an optimal Hyperplane which separates the two classes in an effective manner. SVM has a number of parameters like ’Kernels’ , ’C’ , and ’Gamma’ , which decide the model of SVM we would want to apply on our dataset.
Kernels basically determine the type of Decision Boundeary that we want to employ to sep-

arate different classes .Kernels are of different types such as 'Linear', 'Radial Basis Function(RBF)' , 'Poly' , etc.

Linear SVM's employ a Linear Decision Boundary to segregate the data points of different claases , while RBF employ a Radial Function Boundary to separate the classes . Radial Function between two points x , y in a m-dimensional space is given as K(X,Y)=exp(-(g)||X-Y||$\hat{2}$)

C value is an important feature of SVM Model , which is a measure of trade-off between the accuracy with which we classify the Training Data and the Maximum Margin that we want between the nearest data points of different classes.

If we use a high value of C , we make the decision boundary in such a way that classifies the training data points very correctly.In this case the model is sensitive to the Outliers and the more accuracy in classifying is achieved by decreasing the Margins to an extent possible.The model tries to classify the outliers correctly in this case.

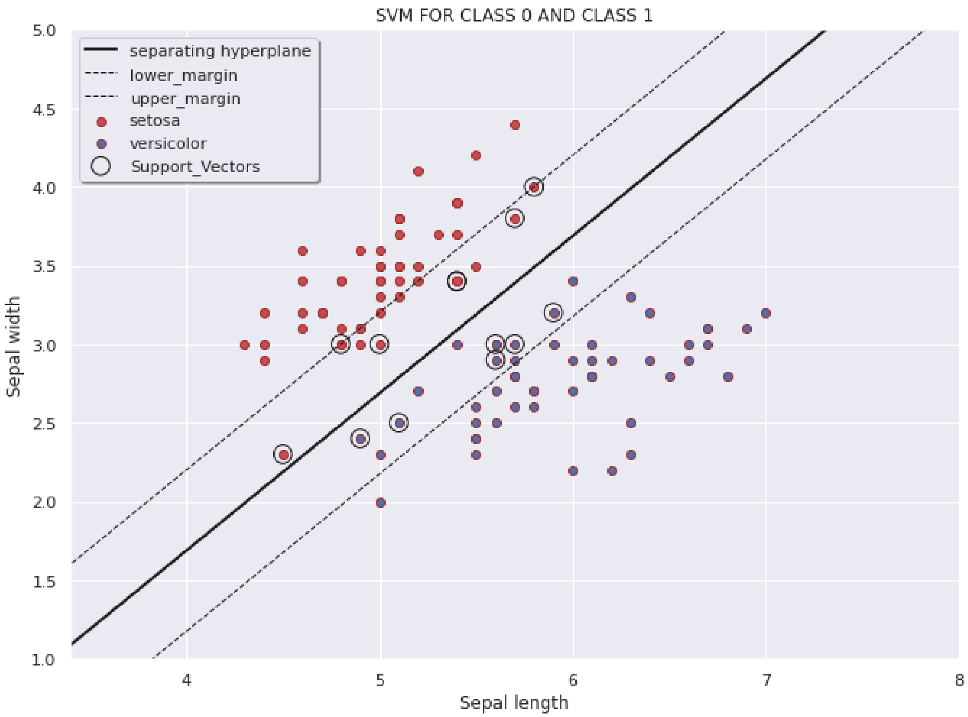
While low values of C implies that the model will try to ignore the outliers to an extent as possible to maximise the Marghiin Width.In this case the decision boundary is drawn as to maximise the Margins of the separating Hyperplane by ignoring the outliers to an extent as possible.

-> The optimal values of C is determined by cross validation of the model on the Test dataset , whichever value of C gives the best results on the Test Dataset , that value is used to train the SVM Model. Thus the variation in test dataset determines an optimal values of C. -> The svm model is trained using 'Linear Kernel' on Iris dataset with features reduced to Sepal Length and Sepal Width with 3 pairs of classes , taking two classes at atime.

-> We plot the Separating Linear Hyperplane, show the Margins , highlight the Support Vectors and plot a scatter plot of the points in the transformed dataset.

-> We also obtain the Classification report for the SVM Model by taking 3 different values of Parameter 'C'.

Plot for Class 0 and Class 1



The Classification Report for Class 0 and Class 1 data using Linear Kernel is:

Value of C=1

	precision	recall	f1-score	support
Setosa	1.00	1.00	1.00	19
Versicolor	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Value of C=0.001

	precision	recall	f1-score	support
Setosa	0.00	0.00	0.00	19
Versicolor	0.37	1.00	0.54	11
accuracy			0.37	30
macro avg	0.18	0.50	0.27	30
weighted avg	0.13	0.37	0.20	30

Value of C=1000

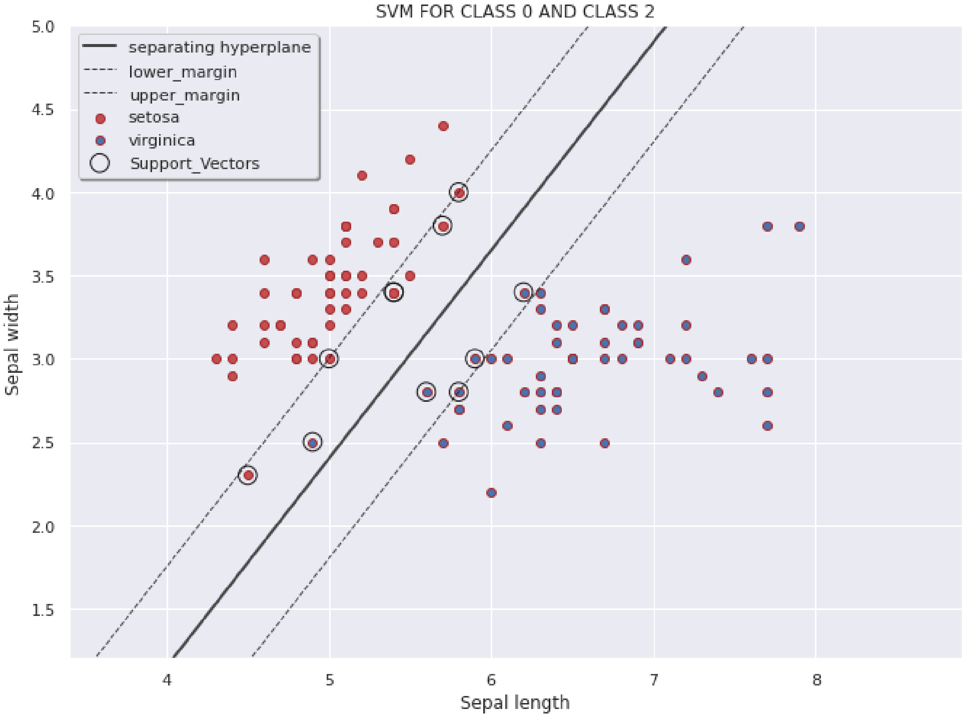
	precision	recall	f1-score	support
Setosa	1.00	1.00	1.00	19
Versicolor	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Observations

-> The accuracy and other parameters like Precision and Recall are 1 for both classes in case

of C Values of 1 and 1000 , While for C=0.001 , the accuracy score decrease drastically to 0.37 , which indicates that most of the Test data points of both classes are sort of outliers with respect to the Training data points of both classes in case of Lower C value of 0.001.
->With such a low value of C ,the svm model tries to maximise the margins and it tolerates wrong classification of training data points in bid for Large Margins.
->But for C values of 1 and 1000 the accuracy is more important than maximisation of Margins for the svm model.Thus in that case the outliers are not ignored to that extent and the accuracy in classification is given importance.
->Thus the two classes data points are not separated into two distinct and far off clusters along the features of Sepal Length and Sepal Width , because if that would have been the case than for low C values the accuracy would have been quite high which is not the case here.

Plot for Class 0 and Class 2



The Classification Report for Class 0 and Class 2 data using Linear Kernel is:
Value of C=1

	precision	recall	f1-score	support
Setosa	1.00	1.00	1.00	19
Virginica	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Value of C=0.001

	precision	recall	f1-score	support
Setosa	0.00	0.00	0.00	19
Virginica	0.37	1.00	0.54	11
accuracy			0.37	30
macro avg	0.18	0.50	0.27	30
weighted avg	0.13	0.37	0.20	30

Value of C=1000

	precision	recall	f1-score	support
Setosa	1.00	1.00	1.00	19
Virginica	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

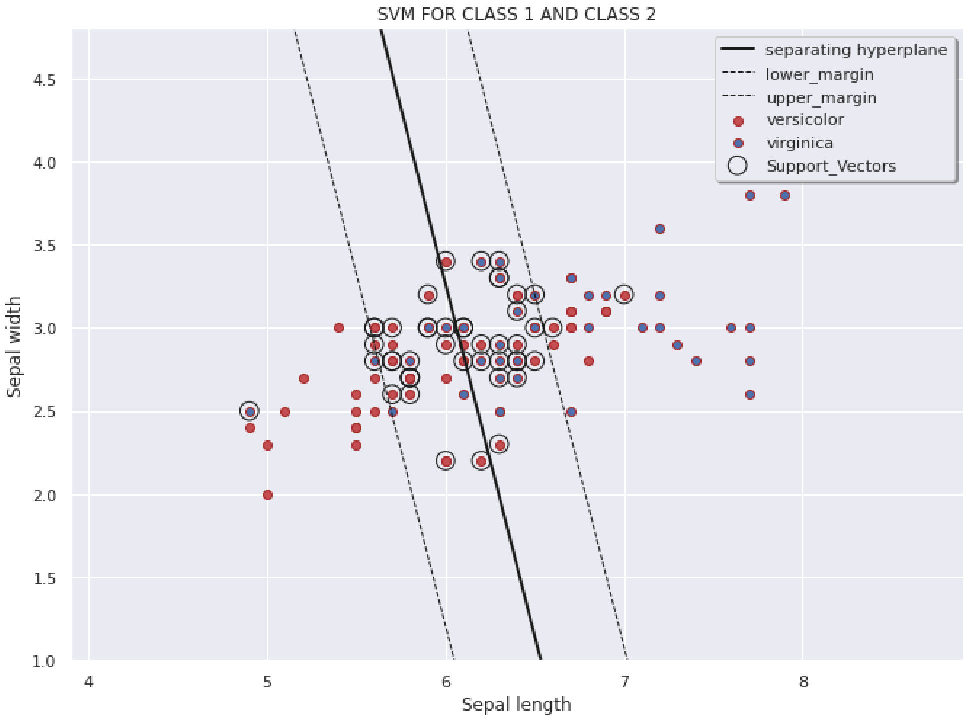
Observations

->The Classifaction Reports for class 0 and class 2 flowers is same as that of Class 0 and Class 1.

->The observations remain same that for very low value of C the accuracy is quite low and foe high values of c like for 1 and 1000 the accuracy is 1.

->Thus the two classes data points are not separated into two distinct and far off clusters along the features of Sepal Length and Sepal Width , because if that would have been the case than for low C values the accuracy would have been quite high which is not the case here.

Plot for Class 1 and Class 2



The Classification Report for Class 1 and Class 2 data using Linear Kernel is:

Value of C=1

	precision	recall	f1-score	support
Versicolor	0.83	0.53	0.65	19
Virginica	0.50	0.82	0.62	11
accuracy			0.63	30
macro avg	0.67	0.67	0.63	30
weighted avg	0.71	0.63	0.64	30

Value of C=0.001

	precision	recall	f1-score	support
Versicolor	0.00	0.00	0.00	19
Virginica	0.37	1.00	0.54	11
accuracy			0.37	30
macro avg	0.18	0.50	0.27	30
weighted avg	0.13	0.37	0.20	30

Value of C=1000

	precision	recall	f1-score	support
Versicolor	0.82	0.47	0.60	19
Virginica	0.47	0.82	0.60	11
accuracy			0.60	30
macro avg	0.65	0.65	0.60	30
weighted avg	0.69	0.60	0.60	30

Observations

->The accuracy for the SVM classifier with value of C =1 is around 63% , for C=1000 is around 60% and for C=0.001 it is around 37%.

->This means that for large values of C also the accuracy is around 60% which means with linear boundary , it is very difficult to separate the two classes very accurately and also there are no unique clusters for the data points of two classes.

->All the Test data points of both the classes do not lie within the boundaries formed by the Training data points of the respective classes , therefore the accuracy is low for all the values of C.

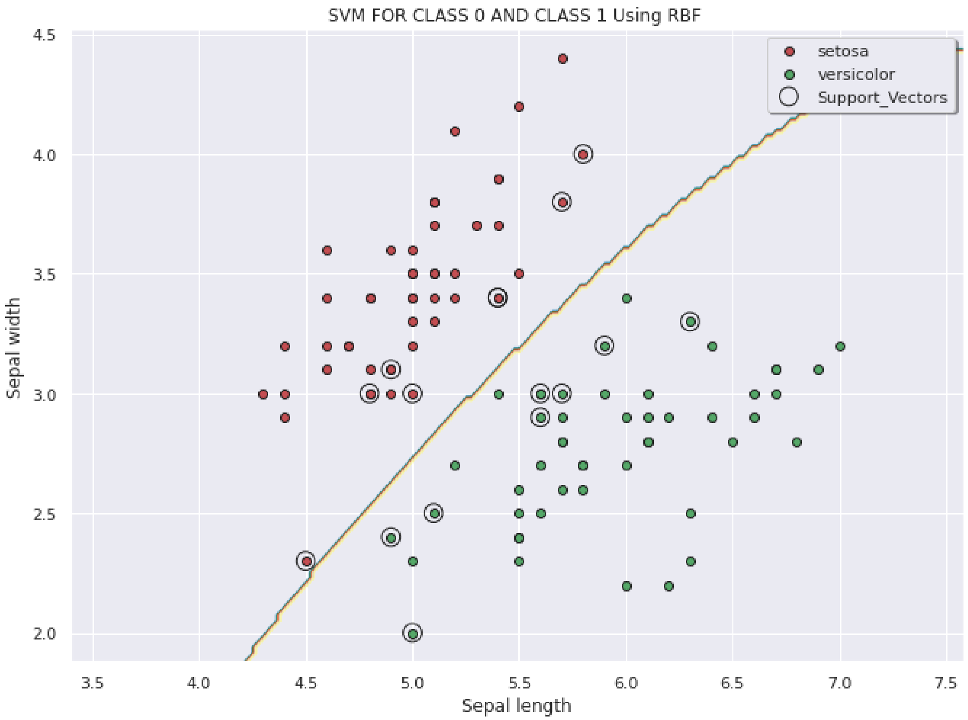
->It is maximum for C=1 as medium value of C trades off most suitabely between the Margin Width and the Accuracy in Classification of Training data points, i.e neither it minimizes margins to achieve maximum possible accuracy in classification of Training Data points(like with C=1000) , nor it maximizes margins ignoring the accuracy in classification of Training Data points(like with C=0.001).It maintains an intermediate stand.

Subtask 3

-> Here we use the rbf Kernel instead of Linear Kernel.

-> Similary to Subtask 1 and Subtask 2 , we plot the Separating Hyperplane (margin not asked in this part) , highlight the Support Vectors and plot the data points.

Plot for Class 0 and Class 1



The Classification Report for Class 0 and Class 1 data using RBF Kernel is:

Value of C=1

	precision	recall	f1-score	support
Setosa	1.00	1.00	1.00	19
Versicolor	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Value of C=0.001

	precision	recall	f1-score	support
Setosa	0.00	0.00	0.00	19
Versicolor	0.37	1.00	0.54	11
accuracy			0.37	30
macro avg	0.18	0.50	0.27	30
weighted avg	0.13	0.37	0.20	30

Value of C=1000

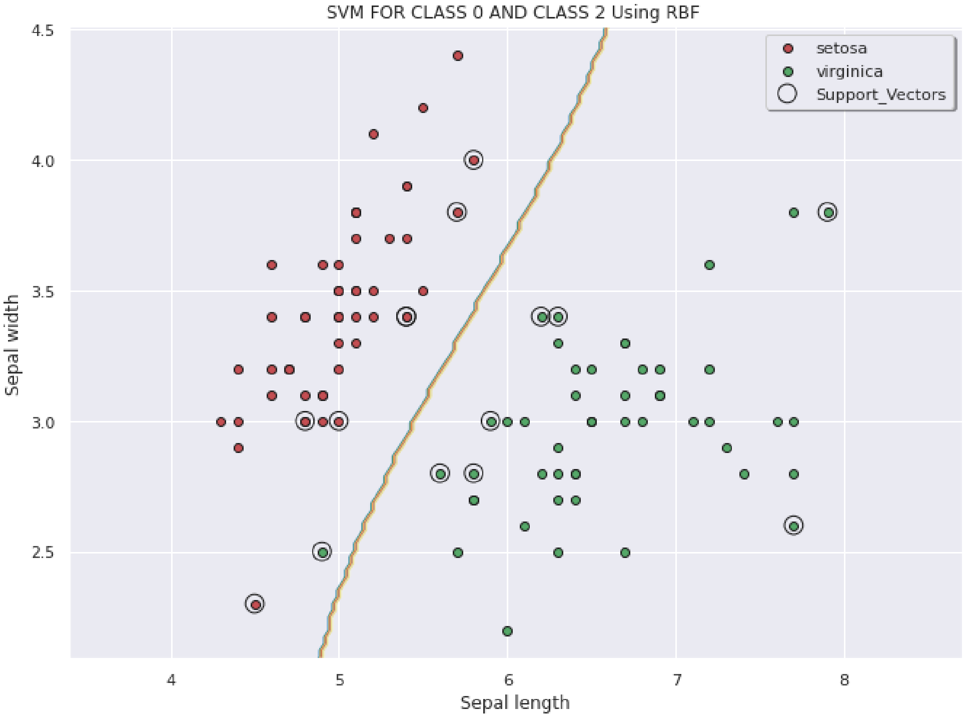
	precision	recall	f1-score	support
Setosa	1.00	1.00	1.00	19
Versicolor	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Observations

-> For values of C=1 and C=1000 ,the accuracy in classification of test data points is 100% , which means that the test data points are separated by the RBF Hyperplane for both values

of C.
->For C=1 , the width of Hyperplane Margins is much more than that with C value equal to 1000,where the accuracy of Training data points classification is more important. This tells us that the test data points of both classes are not outliers with respect to the training data points of respective classes.
->For C=0.001 ,the width of the Hyperplane is maximized as possible ignoring the Training data points classification accuracy to a large extent.This may help in right classification of a small number data points which are very corrupt data points ,but for most of the Test data points , the prediction is wrong.

Plot for Class 0 and Class 2



The Classification Report for Class 0 and Class 2 data using RBF Kernel is:

Value of C=1

	precision	recall	f1-score	support
Setosa	1.00	1.00	1.00	19
Virginica	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Value of C=0.001

	precision	recall	f1-score	support
Setosa	0.00	0.00	0.00	19
Virginica	0.37	1.00	0.54	11
accuracy			0.37	30
macro avg	0.18	0.50	0.27	30
weighted avg	0.13	0.37	0.20	30

Value of C=1000

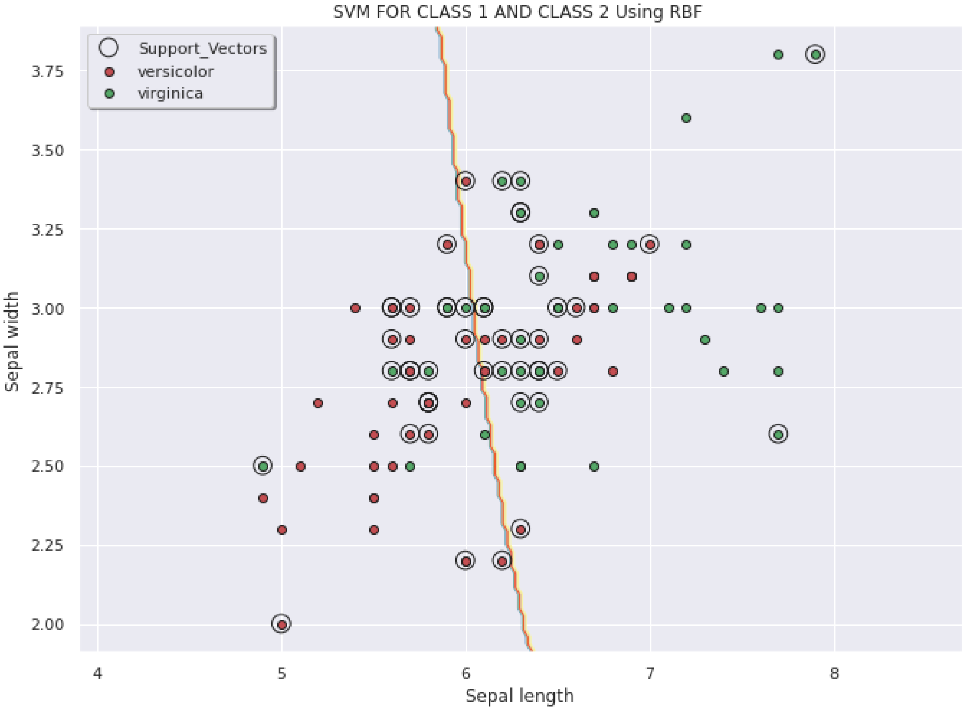
	precision	recall	f1-score	support
Setosa	1.00	1.00	1.00	19
Virginica	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Observations

->In this case also the observations are quite similar to those of class 0 and class 1 .For C=1 and C=1000 ,the accuracy is 1.00 , while for C=0.001 , the accuracy is 0.37.

->The precision and Recall for 'Setosa' class flower is 0 for C=0.001.This tells us that all the test data points of 'Setosa' are mispredicted ,which implies that the Setosa test data points mostly lie in the region of the 'Virginica' class Training Data points.

Plot for Class 1 and Class 2



The Classification Report for Class 1 and Class 2 data using RBF Kernel is:

Value of C=1

	precision	recall	f1-score	support
Versicolor	0.82	0.47	0.60	19
Virginica	0.47	0.82	0.60	11
accuracy			0.60	30
macro avg	0.65	0.65	0.60	30
weighted avg	0.69	0.60	0.60	30

Value of C=0.001

	precision	recall	f1-score	support
Versicolor	0.00	0.00	0.00	19
Virginica	0.37	1.00	0.54	11
accuracy			0.37	30
macro avg	0.18	0.50	0.27	30
weighted avg	0.13	0.37	0.20	30

Value of C=1000

	precision	recall	f1-score	support
Versicolor	0.78	0.37	0.50	19
Virginica	0.43	0.82	0.56	11
accuracy			0.53	30
macro avg	0.60	0.59	0.53	30
weighted avg	0.65	0.53	0.52	30

Observations

-> The Accuracy for C=1 is 60% , while for C=1000 , it is around 53% and for C=0.001 , it is around 37%.

->The RBF Kernel too is not able to separate the test data points into different regions , that is why the accuracy of prediction is low. This is because there are not well defined clusters of the two classes.The two classes data points are somewhat mixed.

->It is maximum for C=1 as medium value of C trades off most suitably between the Margin Width and the Accuracy in Classification of Training data points, i.e neither it minimizes margins to achieve maximum possible accuracy in classification of Training Data points(like with C=1000) , nor it maximizes margins ignoring the accuracy in classification of Training Data points(like with C=0.001).It maintains an intermediate stand.

1.4 Conclusion and Learnings

->We first deal with the Labelled Faces in the Wild Dataset.This consists of 62*47 pixel images.We employ PCA on the dataset of images setting the number of PCAComponents to 100.PCA extracts 100 features of maximum variance out of the 2914 features of the image dataset.

->We then project the transformed image data onto this 100 dimensional space and obtain the projections along each of the eigen vectors . We plot the first 20 EigenFaces , ie we plot the projectiions along the 20 eigen vectors of maximum variance.

->We also employ tSNE algorithm on a dataset consisting of images of any three personalities and observe the clusters formed in the plot.

->We employ the Nearest Neighbour Classifier on the transformed dataset (100 dimensional transformed dataset), tarin it and test it using the test dataset and report its classification report.

->Instead of Hardcoding the number of PCA components , we choose the minimum number of components explaining 80% variance of the dataset and apply the Nearest Neighbour Classifier on that dataset obtained by transforming original dataset into 31 components as 31 explain 80% of the variance.

->We then use the Iris Dataset and apply PCA on it reducing the number of Dimensions to 2.We explain the relation between the 4 features of the original dataset and the 2 PCA eigenvectors by plotting appropriate graph.

->We then apply the LDA on the 2 class transformed Iris dataset for every pair of classes and plot the Projected points along the LDA direction that maximise the inter class scatter while minimsing Intra class variance.We also draw the separating line which is perpendicular to the direction of projected points.We then again employ LDA on the original three class Iris dataset and report the observations.

->We then project the Iris Dataset into a 2-D and 3-D plane by employing tSNE algorithm using two different metric parameter.This plot helps us in visualising similarities and differences between any two pairs of classes out of the 3 classes.
->We then learn about SVM algorithm. We take Sepal Length and Sepal Width as two features in the transformed dataset and apply SVM algorithm on the 3 pairs of classes , two taken at one time, using Linear and RBF Kernels .
->We plot the Separating Hyperplane , and margins(in case of Linear Kernels only) and highlight the Support Vectors.
->We also vary the Value of C parameter and report the classification reports for different values of C.