

Dimensionality Reduction and SVM's

Harshit Sakhuja
2018eeb1044

Indian Institute Of Technology - Ropar

Abstract

This report consists of various inferences gathered after implementing **dimensionality reduction techniques** like Principal Component analysis(PCA) , Linear Discriminant Analysis(LDA) , t-SNE(t-distributed stochastic neighbor estimation) and a maximal margin classifier (**Support Vector Machine** - SVM) on datasets like Labelled Faces In Wild(LFW) and the infamous Fischer Iris data .

1 Task-1 :- Principal component analysis and eigen faces for Face recognition

Overview :-

Applying a k nearest neighbour classifier(knn) after projecting the face images in LFW-dataset to eigen face space. Any face image can literally be decomposed as weighted summation of the basis vectors(eigen faces), and thus each element of the feature(in our case we are reducing to 100 dimensional feature space) we'll extract represents the weight of the corresponding basis vector(eigen face). This of course implies that it should be possible to visualise the basis vectors as meaningful images(top 20 eigen faces shown in 1.3).

Approach:-

Assuming that most face images lie on a low-dimensional subspace determined by the first 100($100 < d$) directions of maximum variance.(where d is the dimension of original feature space)

- Used PCA to determine the vectors u_1, \dots, u_{100} that span that subspace: $x = w_1 u_1 + w_2 u_2 + \dots + w_{100} u_{100}$
- Represent each face using its "face space" coordinates (w_1, \dots, w_{100})
- Perform nearest-neighbor recognition in "face space". Changed the number of principal components and compared the results .

1.1 Transforming each face image to a 100-D vector then visualizing the projected faces(Donald Rumsfeld, Gerhard Schroeder and Tony Blair) by applying t-SNE

Number of PCA components set to be 100.

Training and test set images were projected to the 100-D eigen face space .All 100-D vectors were resized to (10,10) to visualize them as images to find if they make any sense .
Inference :- These are just weights corresponding to each eigen face and are not meaningful images as such .

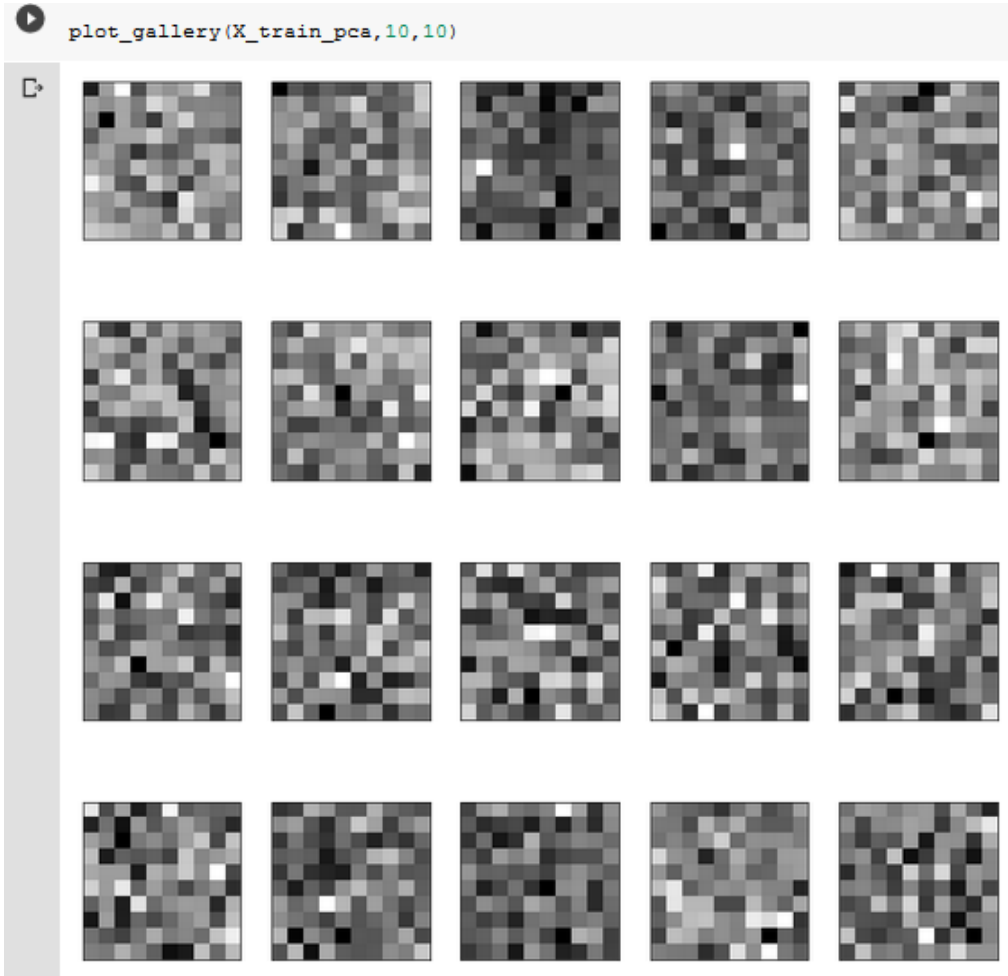


Fig 1: Face images projected to eigen face space (These are not eigen faces)

Downloaded dataset has only 5 classes as a limit of 100 faces per person was set. Out of them Donald Rumsfeld, Gerhard Schroeder and Tony Blair (as they have almost same number of images) have been chosen for visualizing using t-SNE (by reducing to 3 dimensions).

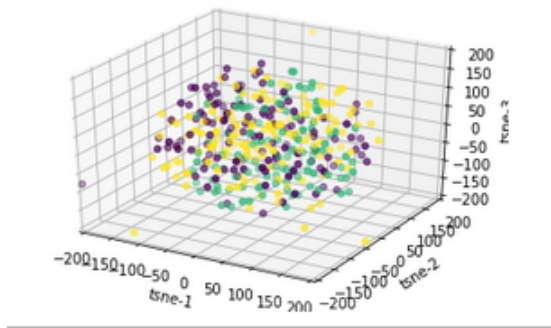
```
236 121 530 109 144
```

```
['Colin Powell' 'Donald Rumsfeld' 'George W Bush' 'Gerhard Schroeder'  
 'Tony Blair']
```

Fig 2:Images per class)

First row consists of number of images per class.
 Second row consists of class names .

This is how the plot looks like

**Fig 3: Three particular classes visualized in 3 dimension by applying t-SNE**

Inferences :-

- 1:) As t-SNE preserves clustering if any while projecting to lower dimensional space , we can infer that there were no class specific clusters in eigen space (100-D space) for the three classes namely Donald Rumsfeld, Gerhard Schroeder and Tony Blair .
- 2:) Maybe there were no class specific clusters in the original feature space or maybe there were class specific clusters but when those images were projected to eigen space (100-D), the class-separability just vanished . So no comment can be made regarding this with certainty .
- 3:) But since we will be applying KNN in the eigen space where there are no class specific clusters atleast for these three classes we will not be able to achieve good results.
- 4:) If there were clusters present , then KNN would have performed well as there would have been sufficient number of neighbours to vote for the correct class but since there are no clusters .

1.2 Applying Nearest Neighbour Classifier (KNN)

Classification Report for k=1 :-

0.5964912280701754					
	precision	recall	f1-score	support	
0	0.55	0.51	0.53	67	
1	0.55	0.44	0.49	39	
2	0.69	0.75	0.72	158	
3	0.44	0.52	0.48	29	
4	0.44	0.41	0.43	49	
accuracy			0.60	342	
macro avg		0.54	0.52	0.53	342
weighted avg		0.59	0.60	0.59	342
0 Colin Powell					
1 Donald Rumsfeld					
2 George W Bush					
3 Gerhard Schroeder					
4 Tony Blair					

Fig 4:- Classification report of K nearest neighbours applied on images in eigen space(K=1)

Inferences :-

Accuracy -60% (This is when svdsolver parameter was kept at auto state when that parameter is changed to randomized state accuracy increase to 72%)

F1 score (macro average) - 0.53

F1 score (weighted average) - 0.59

1:) As we can see classes 1,3 and 4 (which were visualized using t-SNE) have low precision values implying that many other face images were classified as them rather than their true class, this can be because of the high intra class scatter of these classes and no class specific clusters in either of the class.

2:) Since support value for class -2 i.e George Bush is very high thus accuracy is highly influenced by performance of the classifier on that class , so a better metric to use will be weighted average of f1 score over all classes (which is almost same to accuracy in this case)

3:) Also KNN is a weak classifier as the measure used to convey similarity between two images is distance(Euclidean or Manhattan) which has it's own pitfalls.

4:) In this case direction of maximum variance is not the one along which class separability remains intact . As a result nearest neighbour classifier doesn't work so well , had we tried fischer faces (ie using directions along which there is max class separability) instead of eigen faces then nearest neighbour classifier would have performed better.

1.3 Top 20 Eigen faces

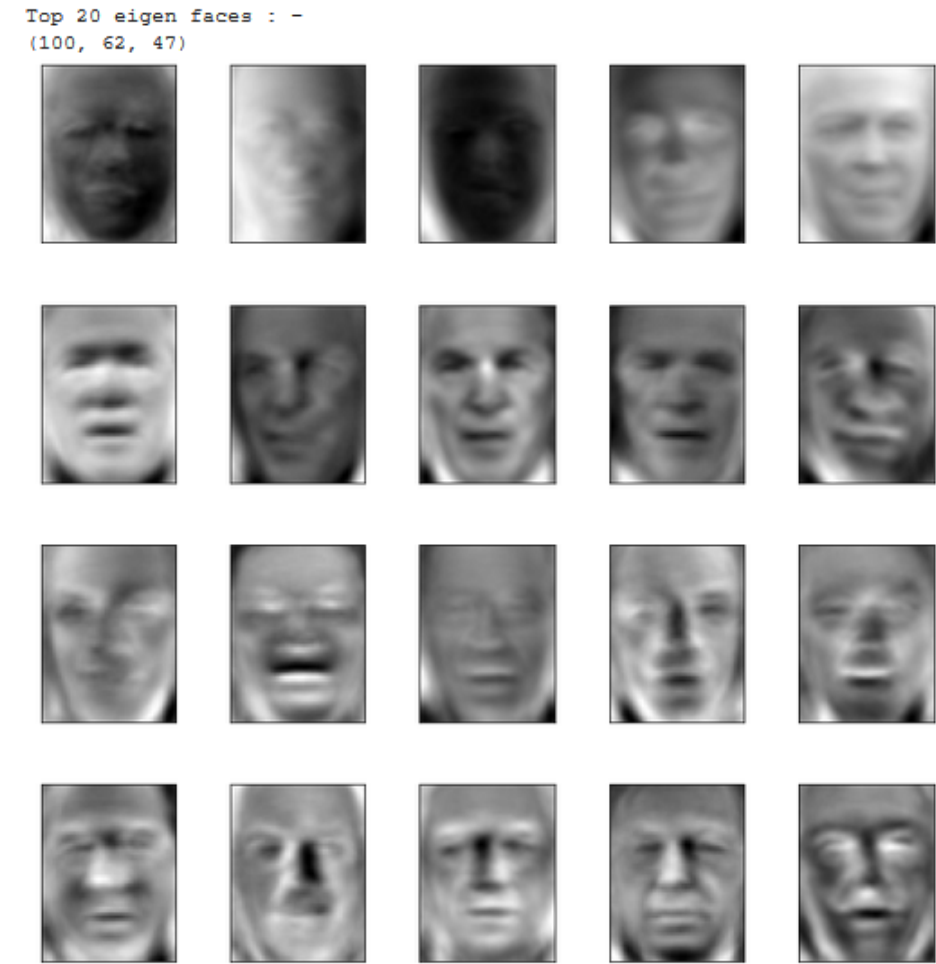


Fig 4:- Top 20 eigen faces

These are image visualizations of the top 20 eigen vectors(eigen vectors corresponding to top 20 eigen values) out of 100 eigen vectors that we found using eigen decomposition . As we can see these are somewhat meaningful which also makes sense as at the end they are eigen orthonormal basis for face images ie face images can be reconstructed by weighted combination of them .

1.4 Projecting face images to a set of eigen faces which cumulatively explain 80% of the total variance in training data

```
BY CHOOSING 100 eigen vectors(pc's/eigen directions/eigen faces) I am able to explain
0.9185228192363866 % of total variance
No. of eigen vectors required to explain 80 % variance : - 32
Exact percentage of variance explained by 32 :- 0.8011994296684861
```

Fig 5:- Ouput snippet from python notebook illustrating variance explained by different number of eigen vectors(eigen faces)

32 eigen vectors or eigen faces are required to explain around 80% variance in training data.

Applying Nearest neighbour classifier to images after they are projected to 32-D space

Classification Report(k=1) :-

0.5584795321637427					
	precision	recall	f1-score	support	
0	0.59	0.56	0.57	77	
1	0.54	0.39	0.45	51	
2	0.65	0.70	0.67	148	
3	0.25	0.43	0.32	21	
4	0.42	0.36	0.39	45	
accuracy			0.56	342	
macro avg		0.49	0.48	342	
weighted avg		0.57	0.56	342	
0 Colin Powell					
1 Donald Rumsfeld					
2 George W Bush					
3 Gerhard Schroeder					
4 Tony Blair					

Fig 6:- Classification report when images were projected to 32-D eigen face orthonormal basis

Observations and Inferences:-

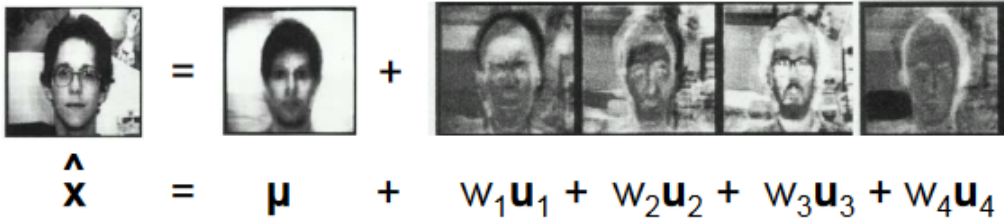
Accuracy:- 0.56
F1 score (macro avg):-0.48
F1 score(weighted avg):-0.56

1.)Accuracy and F1 score are less in comparison to the report generated after applying knn on images projected on 100-D space .100 eigen faces were explaining a total of 90%

variance whereas 32 eigen faces are explaining 80% variance thus this subspace is providing a weaker representation of our images.

In terms of reconstruction:-

• Reconstruction:



$$\hat{x} = \mu + w_1 u_1 + w_2 u_2 + w_3 u_3 + w_4 u_4$$

Reconstruction demonstration

As we reduce the number of eigen vectors , the reconstruction error increases that is the similarity between original and reconstructed image will be less indicating a weaker representation.

2:) In this report also class 1,3 and 4 have low F1 scores.

3:) The most probable reason as to why knn is not working well could be the fact that we are projecting on direction of maximum variance (eigen vectors/eigen faces) which are not always good for classification.

Principal components are not able to perfectly describe data which has a non gaussian distribution so may be our original data didn't had a gaussian distribution.

Knn would have worked better on data that was projected on fischer faces rather than eigen faces.

2 Task 2:- Dimensionality Reduction and Visualization with PCA,LDA and t-SNE on Fischer Iris data

Fischer iris data has 150 samples each having 4 dimensions (Sepal length,Sepal Width,Petal length,Petal width)

2.1 Explaining Variance along pc1 and pc2 after employing PCA to reduce the dimensionality of fisher iris dataset to 2

Standardizing data before applying PCA

The general method of calculation is to determine the distribution mean and standard deviation for each feature. Next we subtract the mean from each feature. Then we divide the values (mean is already subtracted) of each feature by its standard deviation.

```
array([0.72962445, 0.22850762])
```

Explained variance ratio along two dimensions

First eigen vector explains 73% variance whereas second eigen vector explains almost 23% variance of total data .

```
pcafi.components_  
  
array([[ 0.52106591, -0.26934744,  0.5804131 ,  0.56485654],  
       [ 0.37741762,  0.92329566,  0.02449161,  0.06694199]])
```

Contribution of different features in formation of principal components

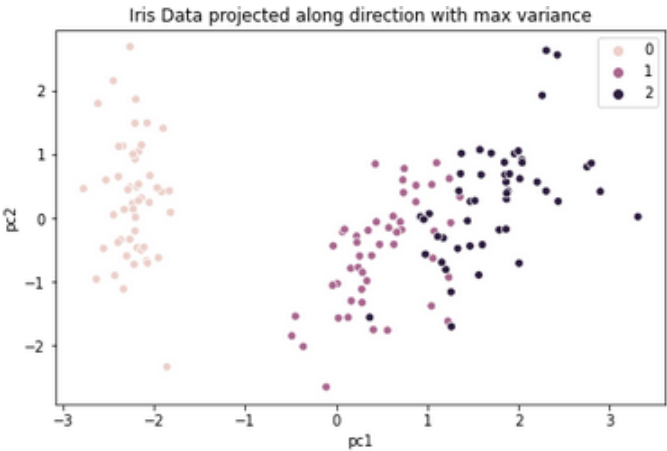


Fig-7:- Iris data projected to 2D space coloured according to species

To explain variance along different eigen vectors , the same plot will be coloured on the basis of Sepal Length,Sepal Width,Petal Length,Petal Width respectively .

Creamish color - refers to low values , **Purple color** refers to medium values ,**Dark blue** color refers to high values.(which is also mentioned in the bottom right corner of the image)

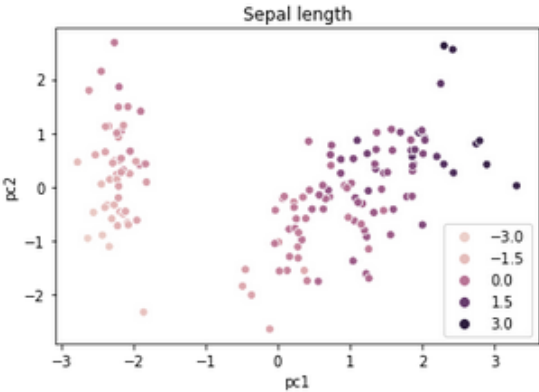


Fig-8:-Coloring by Sepal length

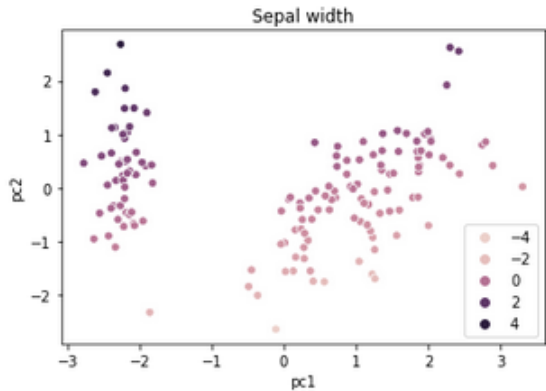


Fig-9:-Coloring by Sepal width

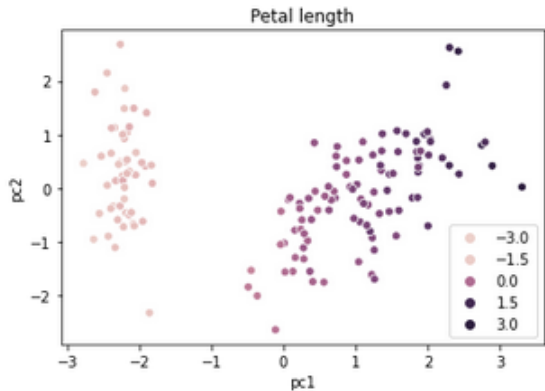


Fig-10:-Coloring by Petal length

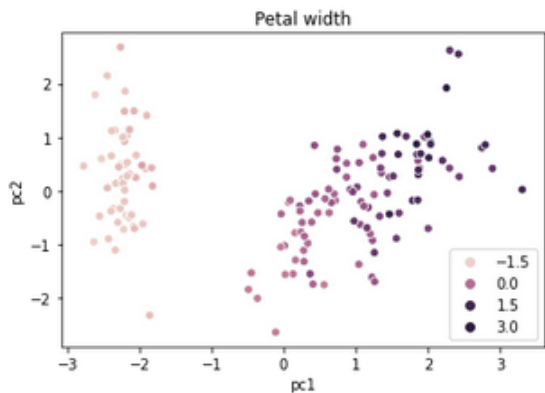


Fig-11:-Coloring by Petal width

Inferences :-

Although low, medium and high ranges of original features are highlighted in above plot. Inferences are drawn for low and high values of the new features only.

Eigen direction-1(pc1)

1:) High values along eigen direction-1(pc1) correspond to high values of sepal length and low values of sepal width. (according to Fig-8 and Fig-9).

2:) High values along eigen direction-1(pc1) correspond to high values of petal length and high values of petal width. (according to Fig-10 and Fig-11)

3:) Low values along eigen direction-1(pc1) correspond to low values of sepal length and high values of sepal width. (according to Fig-8 and Fig-9).

4:) Low values along eigen direction-1(pc1) correspond to low values of petal length and low values of petal width. (according to Fig-10 and Fig-11)

Eigen direction-2(pc2)

1:) High values along eigen-direction-2(pc2) correspond to high value of sepal length and high value of sepal width.

2:) Low values along eigen-direction-2(pc2) correspond to low value of sepal length and low value of sepal width.

2.2 LDA on Fischer Iris Data

Class 0 :- Setosa, Class 1:-Versicolor, Class 3:- Virginica

Pair-1 consists of classes 0 and 1, Pair-2 consists of classes 1 and 2 whereas Pair-3 consists of classes 0 and 2.

After projecting this two class data present in 4 dimensions to a single dimension following is the decision boundary and 1D data points.

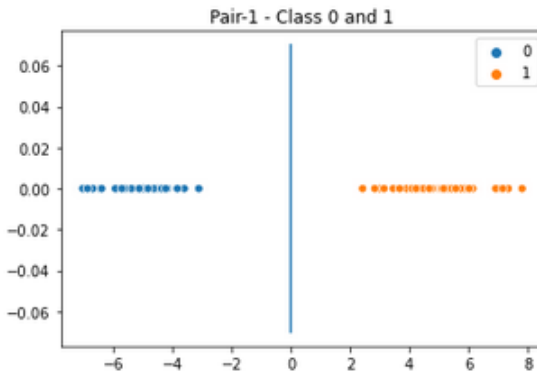


Fig-12:-Pair-1

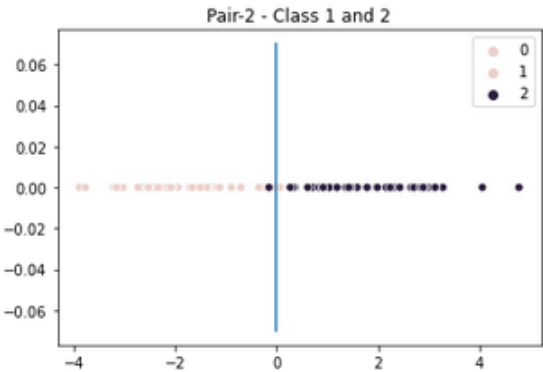


Fig-13:-Pair-2

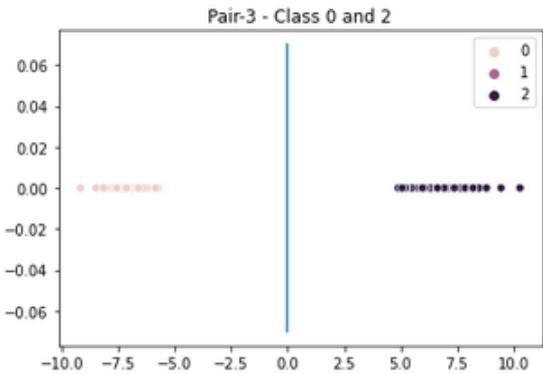


Fig-14:-Pair-3

Inferences:-

1:)Pair-1 and Pair-3 are linearly separable , decision boundary in case of lda passes through mean of projected centres which may or may not be the max margin decision boundary i.e optimal decision boundary which is able to generalize well on unseen data.
 If both classes have equal variance then the decision boundary will have max margin otherwise we will have a suboptimal decision boundary.

2:) Pair-2 is not linearly separable thus whatever be the decision boundary there will always be misclassifications on training data points.

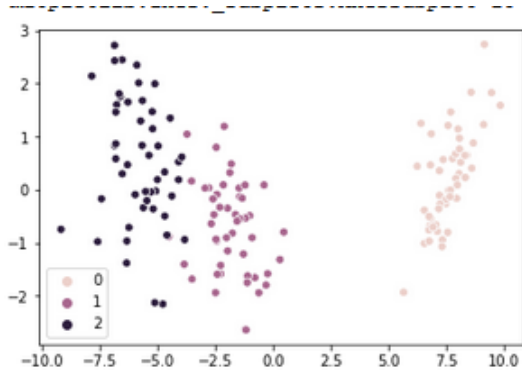


Fig-15:- Three Class data(4 dimensional) projected to 2D space

2.3 t-SNE on 4D Iris dataset by using two different values of 'metric' parameter

So mostly after applying t-SNE , difference information and density information is lost however nearest neighbours information remains intact. Thus, no interpretations can be made based on distance between two clusters or density of a cluster . However if there is a cluster in the projected space this implies there should be similar structure in the high dimensional space , however if actual data is sparse then there will be no clusters in the projected space.

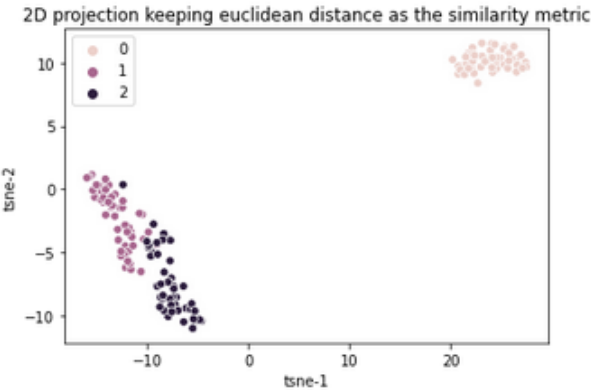


Fig-16:- 2D projection keeping euclidean distance as the similarity metric

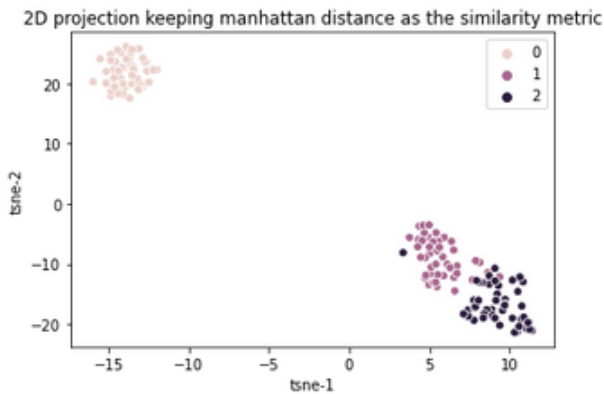


Fig-17:- 2D projection keeping manhattan distance as the similarity metric

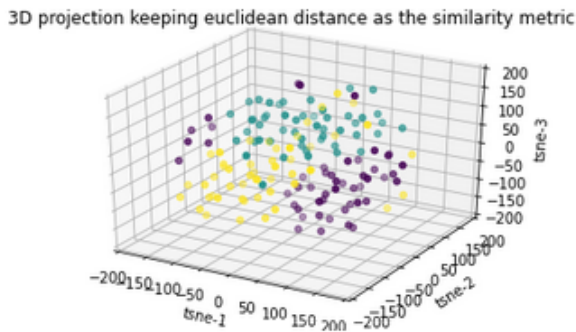


Fig-18:- 3D projection keeping euclidean distance as the similarity metric

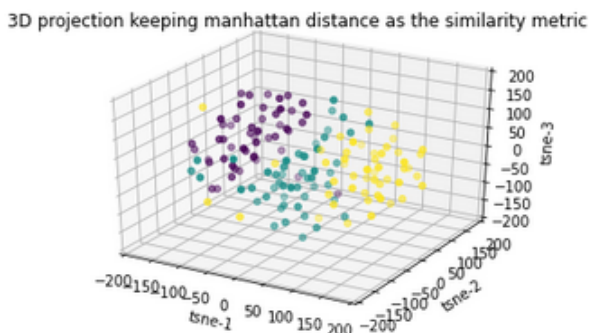


Fig-19:- 3D projection keeping manhattan distance as the similarity metric

Inference:-
1:) As we can see , when we change the similarty metric(from euclidean to manhattan) the orientation of clusters changes but I am not sure about how much is the influence of metric change.
As t-SNE being a probabilistic approach even if we keep the same metric each run of the

also gives different orientation of the clusters.

2:)Also Iris data belonged to a space where euclidean distance gives the knowledge of clusters if any , if our data belonged to non-euclidean space then euclidean metric would't have recognised clusters in the high D space ,if any.

3 Data Classification with linear and non-linear SVM

Converted the 4D Iris data to 2D Iris data by considering the following feature pair :- Sepal length and Sepal Width .
Pair 1 contains classes -> setosa,versicolor

3.1 Classifying all the three pair of classes with linear SVM

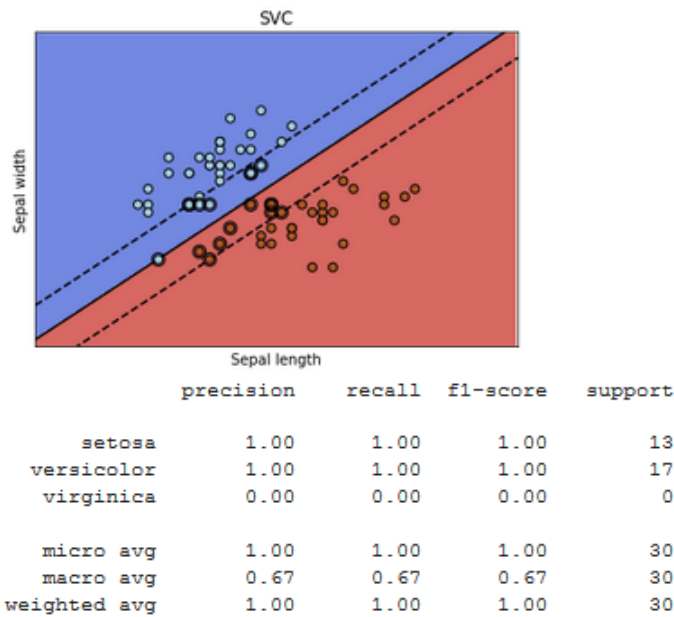


Fig-20:- Pair-1 Max-margin hyperplane with highlighted support vectors along with the classification report

Pair 2 contains classes -> versicolor,virginica

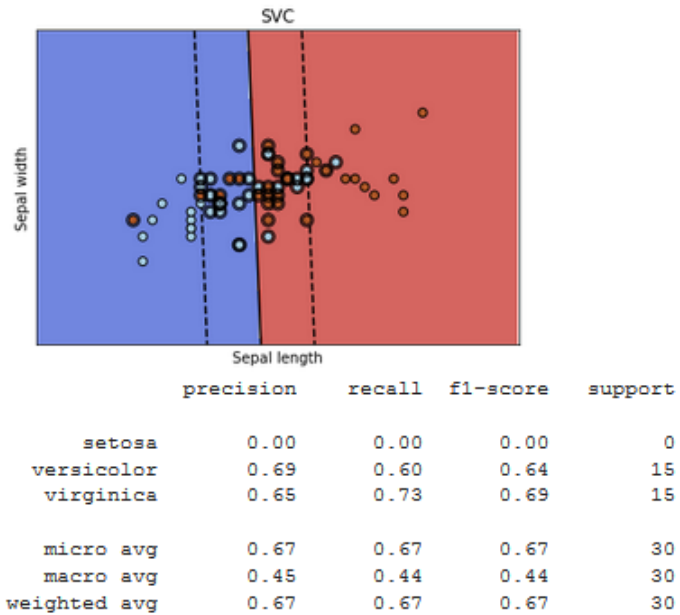


Fig-21:- Pair-2 Max-margin hyperplane with highlighted support vectors along with the classification report

pair 3 contains classes -> setosa,virginica

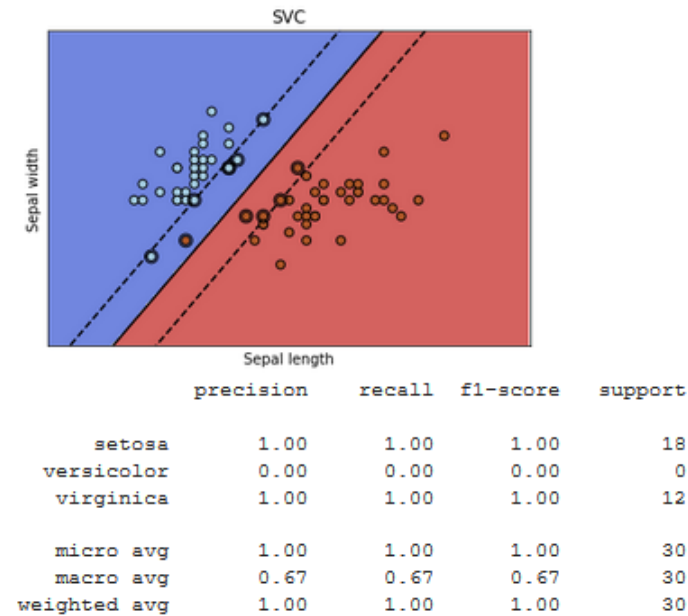


Fig-22:- Pair-3 Max-margin hyperplane with highlighted support vectors along with the classification report

Inferences:-

1:)Pair 1 and Pair 2 are linearly separable as a result linear SVM performed well on both of them , in both the classification reports F1 score corresponding to the classes present in test set (as per pair) is 1.

2:)Support in both the cases add up to 30 (test set size) but there is slight class imbalance in test set . macro avg in both the cases is giving value 0.67 which is because of the fact that only two classes are present in test set at a time.

3:)As we already saw pair-2 is not linearly separable as a result linear SVM is not performing well in this case. Accuracy is around 0.67 and we have similar f1 scores for both the classes.

Inference related to margins and support vectors are written in section 3.2

3.2 Comparing the performance of SVM based on different values of C(Regularization parameter)

	precision	recall	f1-score	support
setosa	0.47	1.00	0.64	14
versicolor	0.00	0.00	0.00	16
virginica	0.00	0.00	0.00	0
micro avg	0.47	0.47	0.47	30
macro avg	0.16	0.33	0.21	30
weighted avg	0.22	0.47	0.30	30

Fig 23:- Classification report of pair1(setosa,versicolor) for c=0.001

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	14
versicolor	1.00	1.00	1.00	16
virginica	0.00	0.00	0.00	0
micro avg	1.00	1.00	1.00	30
macro avg	0.67	0.67	0.67	30
weighted avg	1.00	1.00	1.00	30

Fig 24:- Classification report of pair1(setosa,versicolor) for c=1000

Inference :-

Small value of C leads to big margin

High value of C leads to small margin

1:)SVM tries to have a hyperplane which generalizes well i.e it's performance on testing (unseen) data is best for that purpose high margin classifiers are preferred. With high margin there may be a comparatively higher misclassification rate on training data but it is expected that such hyperplane will have comparataively lower misclassification on testing data.

2:)Previously we plotted the classification report for default c i.e 1 where accuracy was 100% .

3:)For c=0.01 The thing to note is f1 score corresponding to versicolor is 0 which implies **none of the versicolor was detected** which also reflects in the weighted average F1 =0.30 . Thus for such a low value of c , margin becomes too much which ultimately hampers the performance of the classifier .

For c=1000 , classification report is exactly similar to C=1 , so no inferences can be made based on that , however plot suggests that margin became really thin .

	precision	recall	f1-score	support
setosa	0.00	0.00	0.00	0
versicolor	0.43	1.00	0.60	13
virginica	0.00	0.00	0.00	17
micro avg	0.43	0.43	0.43	30
macro avg	0.14	0.33	0.20	30
weighted avg	0.19	0.43	0.26	30

Fig 25:- Classification report of pair2(virginica,versicolor) for c=0.001

	precision	recall	f1-score	support
setosa	0.00	0.00	0.00	0
versicolor	0.75	0.63	0.69	19
virginica	0.50	0.64	0.56	11
micro avg	0.63	0.63	0.63	30
macro avg	0.42	0.42	0.42	30
weighted avg	0.66	0.63	0.64	30

Fig 26:- Classification report of pair2(virginica,versicolor) for c=1000

Inference :-

1:) Again in the case of 0.001 the margin becomes so big that it's not able to classify virginica leading to (0) F1 score for that class .

2:) In case of c=1000 weighted average value of f1 is slightly less than that in case of c=1 , thus we can infer that margin would have decreased significantly which would have caused comparatively poor performance of classifier.

	precision	recall	f1-score	support
setosa	0.37	1.00	0.54	11
versicolor	0.00	0.00	0.00	0
virginica	0.00	0.00	0.00	19
micro avg	0.37	0.37	0.37	30
macro avg	0.12	0.33	0.18	30
weighted avg	0.13	0.37	0.20	30

Fig 27:- Classification report of pair3(virginica,setosa) for c=0.001

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	16
versicolor	0.00	0.00	0.00	0
virginica	1.00	1.00	1.00	14
micro avg	1.00	1.00	1.00	30
macro avg	0.67	0.67	0.67	30
weighted avg	1.00	1.00	1.00	30

Fig 28:- Classification report of pair3(virginica,setosa) for c=1000

Inference :-
1:) For c=0.001 This time virginica had (0) F1 score implying very big margins.
2:) For c=1000 Exactly similar classification report is there as compared to c=1 do no inference can be made on it's basis .However, definitely margins would have decreased

So we saw that both very big and very small values of C have the potential to screw up our classifier because really large margins can completely lead to ignoring a class as we saw and small margins are in a sense more probable to overfit .
Thus the best practice is to run a search for c , to find best possible values for your validation set.

3.3 Repeating (3.1) employing an RBF instead of a linear kernel

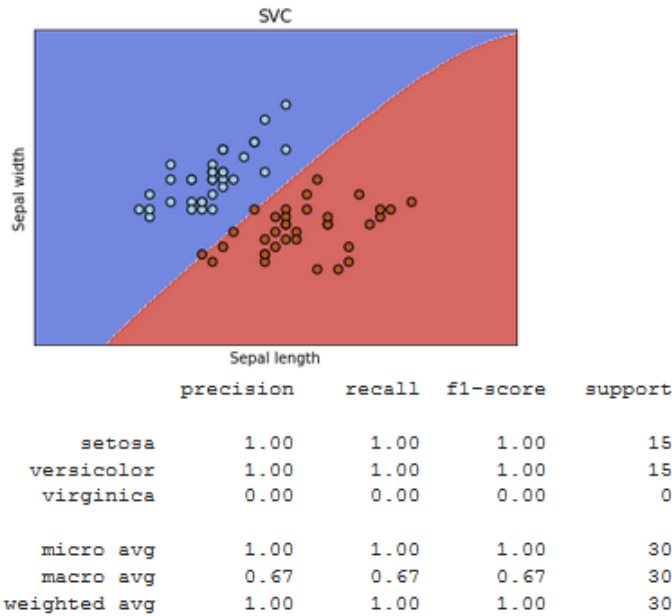


Fig 29:- Classification report of pair1(versicolor,setosa) for c=0.001

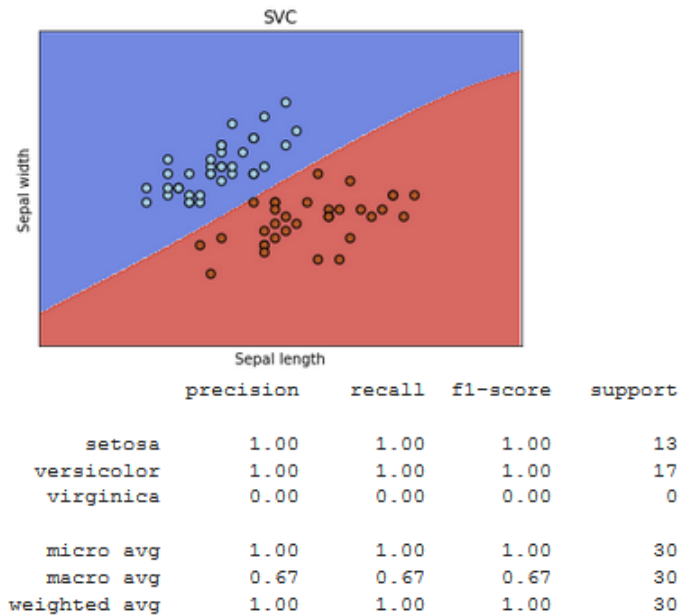


Fig 30:- Classification report of pair1(versicolor,setosa) for c=1

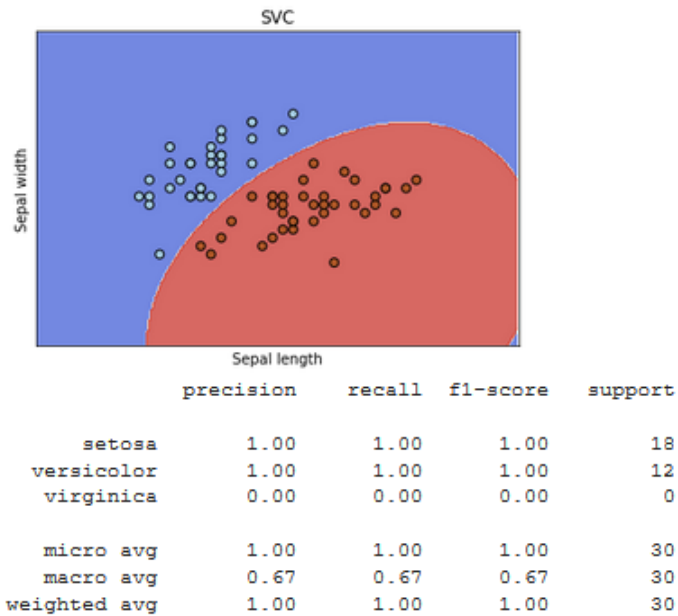


Fig 31:- Classification report of pair1(versicolor,setosa) for c=1000

Inference:-
1:)As these pairs were already linear separable for $c=1$ and $c=1000$ there is no difference in the classification report as such only the shape of decision boundary is different.

But earlier for $c=0$ when we had a linear kernel we had a very simple hypothesis(decision boundary) a perfect example of underfitting as a result of which one class was completely unrecognized but here the non-linear kernel added suitable amount of complexity to our hypothesis thus preventing it from underfitting .

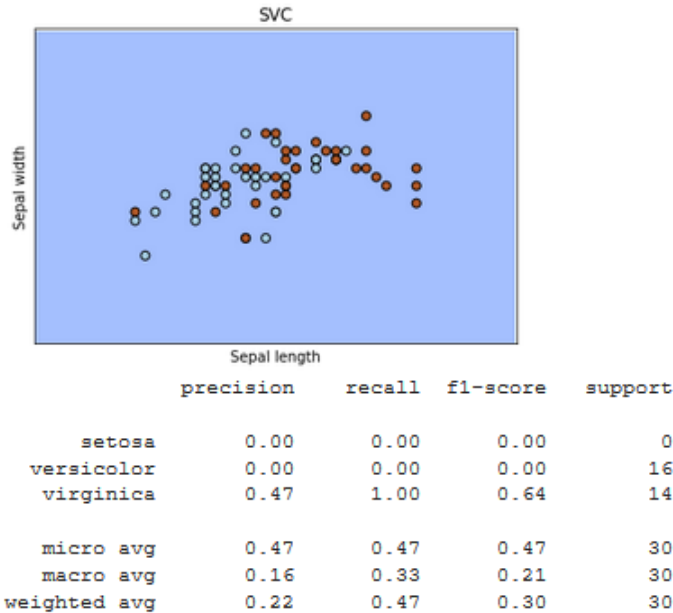


Fig 32:- Classification report of pair2(versicolor,virginica) for $c=0.001$

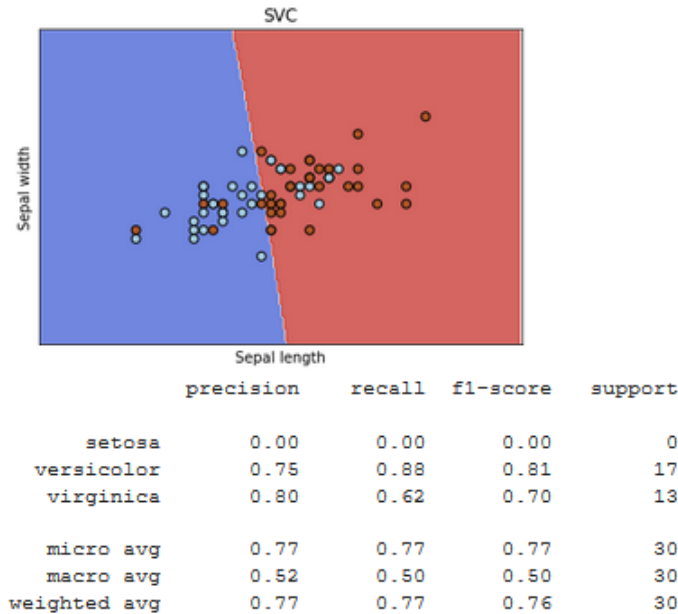


Fig 33:- Classification report of pair2(versicolor,virginica) for c=1

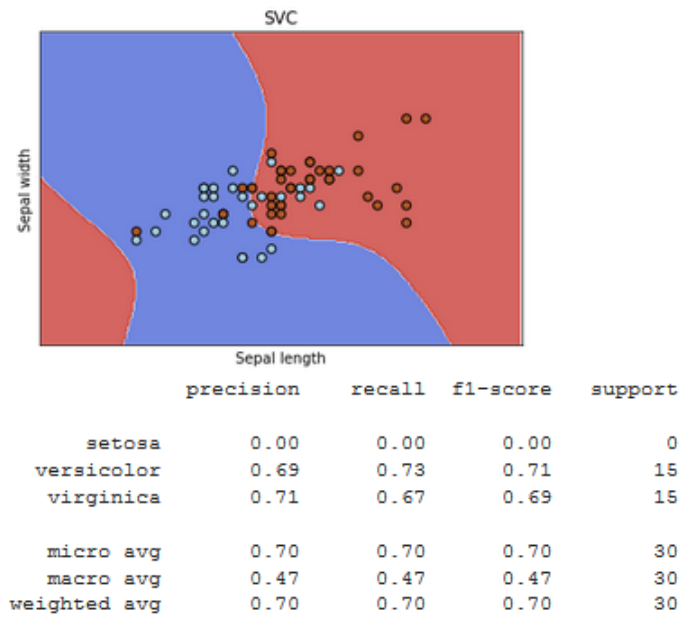


Fig 34:- Classification report of pair2(versicolor,virginica) for c=1000

Inference:-

1:) This pair was not linearly separable so introducing a non-linear kernel hepled it a lot for all values of c as compared to having a linear kernel.(increased the weighed avg F1 score) For the non-linear kernel if we compare all the values of c, max f1 score is for c=1 as for c=0.001 margins become too huge(too worried about testing data) and for c=1000 margins become ver less(less worrying about testing data). So a in between value (c=1 was best).

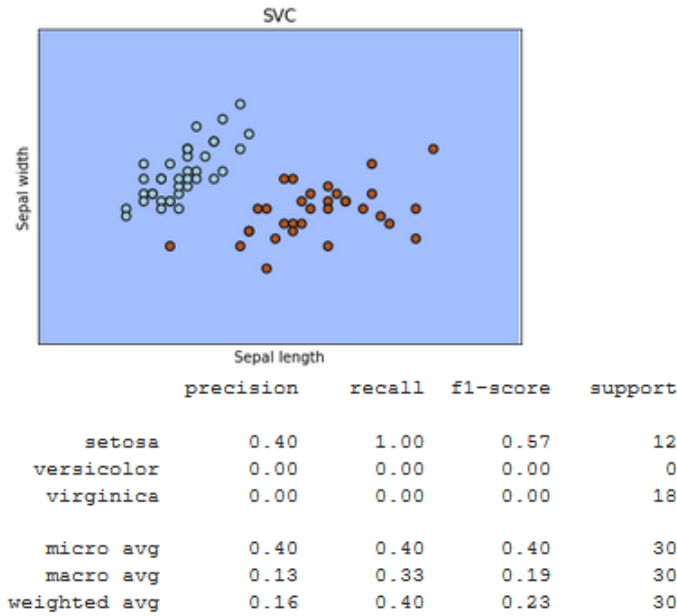


Fig 35:- Classification report of pair3(virginica,setosa) for c=0.001

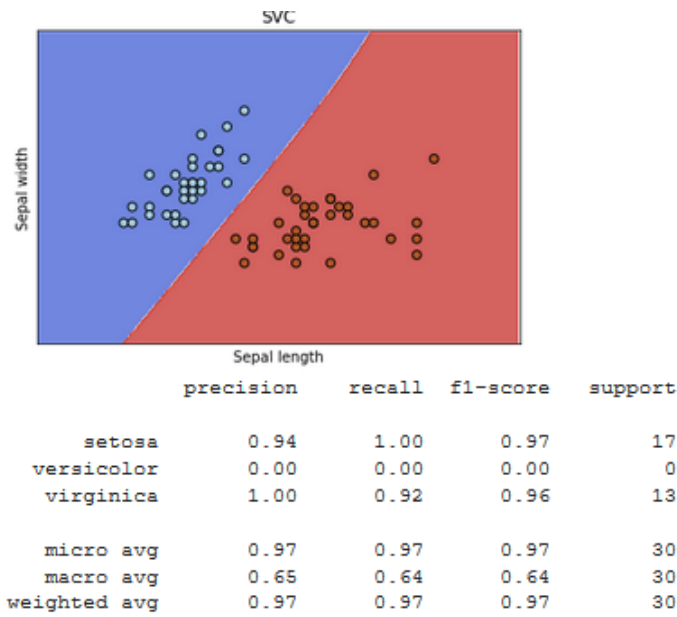


Fig 36:- Classification report of pair3(virginica,setosa) for c=1

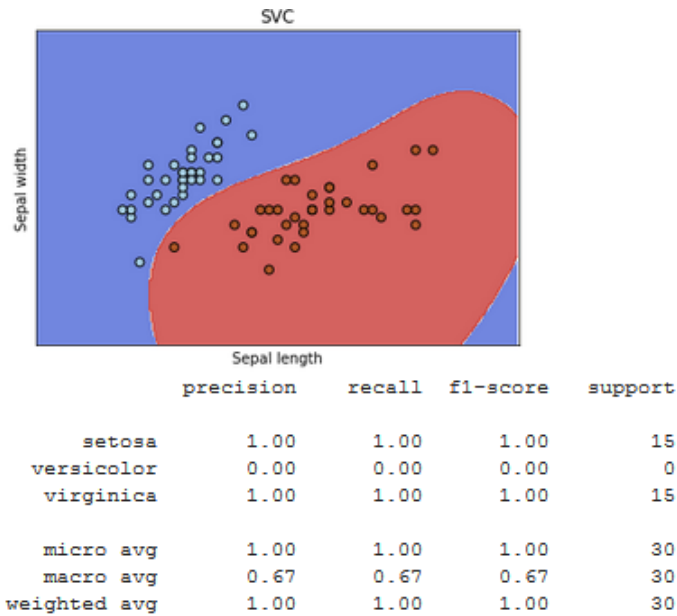


Fig 37:- Classification report of pair3(virginica,setosa) for c=1000

Inference:-

1:)In this case c=1000 performed best as compared to c=1 or c=0.001 .

Here also c=0.001 i.e having a very big margin which implies that model is thinking too much about test set and not fitting to its capacity to the training set(underfitting) hampers the performance of the model.

For c=1 and c=1000 accuracy is different in the non linear kernel case in contrast to the previous case but if we look at the support values , in c=1000 case test set is balanced whereas for c=1 there is slight imbalance.