Assignment -3

Report

Manav Mago(2019EEB1274)

Abstract

This report walks us through the application of the dimensionality reduction techniques like the PCA, LDA , t-sNE and the maximal margin classfier(SVM) on two datasets which are Labelled Faces In Wild(LFW) and the Fischer Iris data .

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

Introduction and Thought flow

In order to analyse the images in the LFW dataset we reduce the feature space to 100 dimensional feature space using PCA without loss of much information.Since any image can be represented using the basis vectors of certain weight and those weights can calculated using the techniques like PCA, the images can be visualized using these basis vectors .Moreover the classification algorithms such as the k-nearest neighbor has also be applied on the results of the superivised learning algorithm PCA to check whether the basis vectors(eigen faces) are accurate enough to classify images or not.

1.1 Transforming each face image to a 100-D vector then visualizing the projected faces by applying t-SNE

Out of the five personalities in the dataset Colin Powell, Donald Rumsfield ,George W bush have been taken for visualization using t-SNE(by projecting it to 2-dimensions)

The 2-dimensional t-sNE plot for the above mentioned three personalities looks like

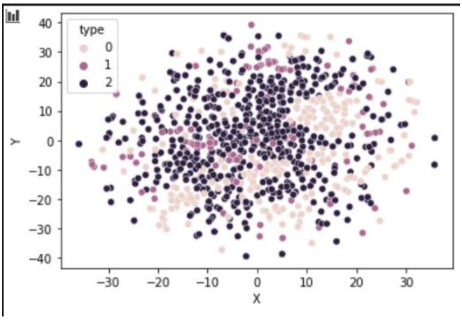


Fig 1: Three personalities visualized in 2-dimension by applying t-SNE

Observations :-

1) Prominent Clusters can be observed for George W Bush and smaller clusters can be observed for Colin Powell while there is no dominant presence of clusters for the Donald Rumsfield.

2) Secondly the presence of clusters even in the lower dimensional feature space is evident from classification report results of KNN performed with 100 components as the higher precison value comes out in the case of George W Bush and Colin powell.

1.2 Applying k-Nearest Neighbour Classifier (KNN)

K-Nearest Neighbor also known as KNN is a supervised learning algorithm that can be used for regression as well as classification problems. But KNN is widely used for classification problems. KNN works on a principle assuming every data point falling near to each other is falling in the same class. That means similar things are near to each other .Over here for applying the k value is taken to be 1.

0.5906432748538012

	precision	recall	f1-score	support
Colin Powel	0.52	0.62	0.57	69
Donald Rumsfield	0.37	0.54	0.43	28
George W Bush	0.78	0.65	0.71	171
Gerhard Schroeder	0.31	0.50	0.39	22
Tony Blair	0.52	0.42	0.47	52
accuracy			0.59	342
macro avg	0.50	0.55	0.51	342
weighted avg	0.63	0.59	0.60	342

Fig 2:- Classification report of K nearest neighbours applied on images in eigen space with components=100

Observations :-

The accuracy comes out to be 59.06% when we reduce the feature space into 100 components

1)It can be observed in the classes 2 and 4 that they have low precision values which indicates many misclassifications i.e many other face images were wrongly classified as them than their true class. This might have occurred due to high intra class scatter and lack of clustering .For eg Lack of clustering for class 2 can be seen in the t-sNE plot as well.

2) Precision in case of class 3 is very high which indicates better clustering of data.

3) It can be observed that the accuracy is better in case when 100 components are used this is because we have more no of eigen orthonormal basis for correct face recognition and hence more accuracy is observed which is evident form the classification report as well.

1.3 First 20 Eigen faces

Fig 3:- First 20 eigen faces

The given figure represents the top 20 eigen faces found after the reducing the dimensionality of the given dataset .Though these are not complete images but they are helpful in full reconstruction of images as they act as eigen orthonormal basis for the face images. The weighed combination of them is further used in reconstruction.

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

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

```
32
100 eigen vector explain :  0.9225893102702685  variance
Eigen vectors required to explain 80 % variance : 32
```

Fig 4:- screenshot of output of python code showing the required no of eigen vector to explain 80% variance.

Applying K- Nearest neighbour classifier to images after they are projected to 32-Dimensional space.

```
0.5467836257309941
                   precision    recall  f1-score   support

      Colin Powel       0.63      0.53      0.57        80
  Donald Rumsfield       0.32      0.37      0.34        38
     George W Bush       0.68      0.68      0.68       151
 Gerhard Schroeder       0.25      0.43      0.32        21
       Tony Blair       0.43      0.37      0.40        52

         accuracy                           0.55       342
        macro avg       0.46      0.47      0.46       342
     weighted avg       0.56      0.55      0.55       342
```

Fig 6:- Classification report of K nearest neighbours applied on images in eigen space with components=32

Observations:-

The Accuracy comes out to be 54.67% when we reduce the components to 32 to explain the 80% variance.

1)Accuracy and F1 score are less in comparison to the report generated after applying KNN on images projected on 100-D space because we have lesser no of eigen orthonormal basis for correct face prediction in case of 32 components and as the reconstruction error will increase ,thereby giving lesser similiarity between the original and reconstructed image and thus less accuracy.

2)Again, we can observe that the precision in case of class 2,4 is bad because there is no similiarity between the neighbouring data which results in poor classification

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 features (Sepal length , Sepal Width ,Petal length, Petal width)and three classes namely:-

a) IRIS-SETOSA(class-0)

b) IRIS-VERSICOLOR(class-1)

c) IRIS-VIRGINICA(class-2)

2.1 Explaining Variance along pca1 and pca2 after employing PCA to reduce the dimensionality of fisher iris dataset to 2

array([0.92461872, 0.05306648])

Fig 6:- Data from pyton code ouput which explains variance ratio along two dimensions

- First eigen vector explains 92.46% variance whereas second eigen vector vexplains almost 5.3% variance of total data .

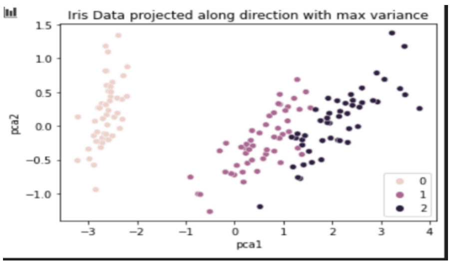
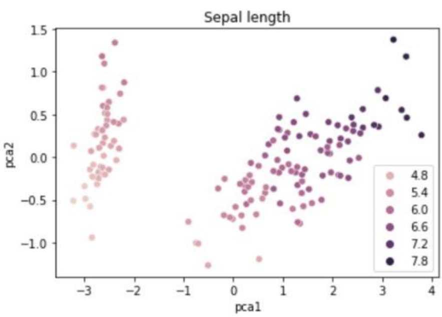


Fig 7:- Plot of the iris data along the different eigen vectors

To explain the variance along the different eigen vectors ,the plot can be coloured using the different features which are sepal length ,sepal width, petal length and petal width where the intensity of these colours vary from cremish to purple color where lesser color intensity refers to lesser values and higher intensity refers to higer values.



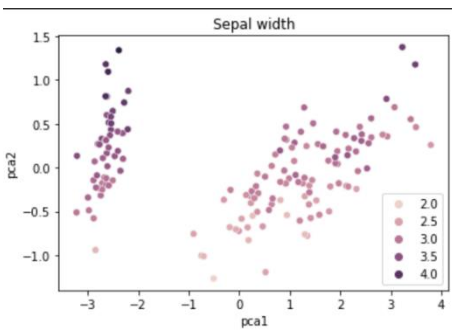
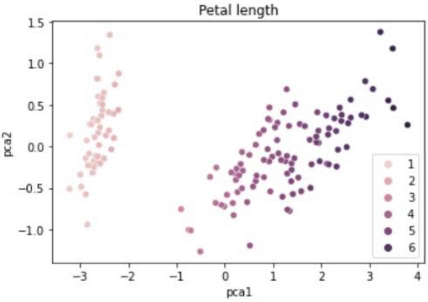


Fig 8:- Plot coloured using sepal length

Fig 9:- Plot coloured using sepal width



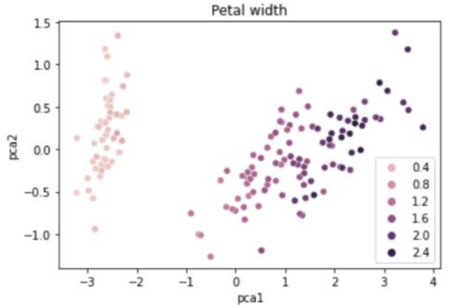


Fig 10:- Plot coloured using petal length

Fig 11:- Plot coloured using petal width

Observations :-

For Eigen direction-1(pca1)

1)High values along eigen direction-1(pca1) correspond to high values of sepal length and low values of sepal width.
2) Low values along eigen direction-1(pca1) correspond to low values of sepal length and high values of sepal width
3)High values along eigen direction-1(pca1) correspond to high values of petal length and high values of petal width.
4)Low values along eigen direction-1(pca1) correspond to low values of petal length and low values of petal width.

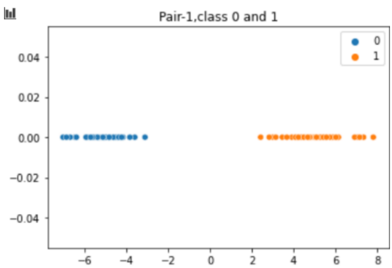
Eigen direction-2(pca2)

1)High values along eigen-direction-2(pca2) correspond to high value of sepal length and high value of sepal width.
2)Low values along eigen-direction-2(pca2) correspond to low value of sepal length and low value of sepal width.

2.1 LDA on Fischer Iris Data

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 plots can be observed in which decision boundary can be visualized passing through zero on x-axis and parallel to y-axis.



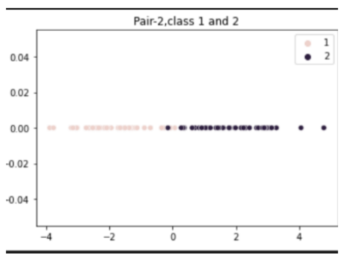


Fig-12:-LDA in pair 1 **Fig-13:-LDA in pair 2**

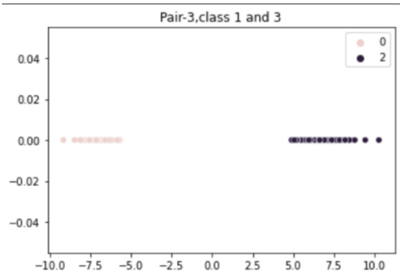


Fig-13:-LDA in pair 3

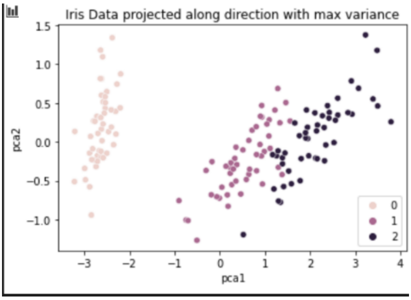


Fig-14:-4-dimensional data projected On 2-dimensional space

Observations:-

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) There are a lot of misclassifications for pair-2 which can be seen in the plot and we also conclude that they are not linearly seperable unlike the pair-1 and pair-3 ,whatever the decision boundary be.

2.2 t-SNE on 4D Iris dataset by using two different measures of ’metric’ parameter

t-sNE is very usefull algorithm for visualizing the data with higher dimensions and somewhat more effective than the PCA.
So mostly after applying t-SNE information is lost however information pertaining to nearest neighbours and similiarity of the data points with multiple features remains intact.

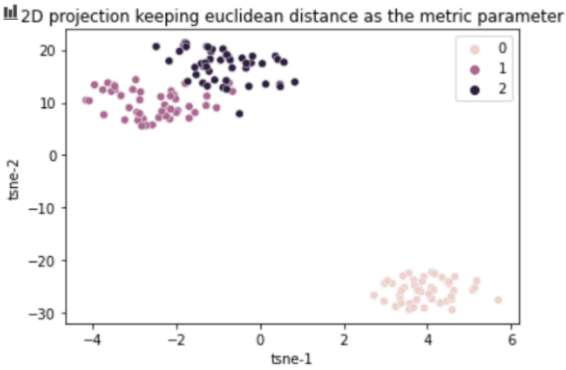


Fig-16:- 2D projection keeping euclidean distance as the metric parameter

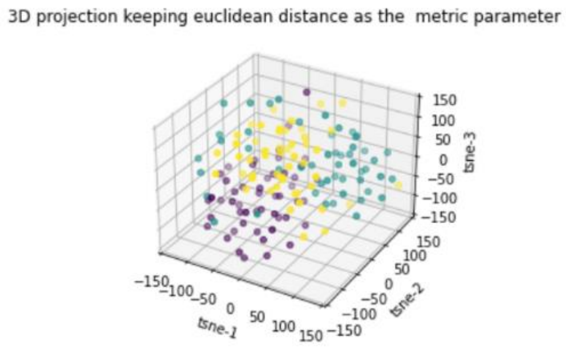


Fig-17:- 3D projection keeping euclidean distance as the metric parameter

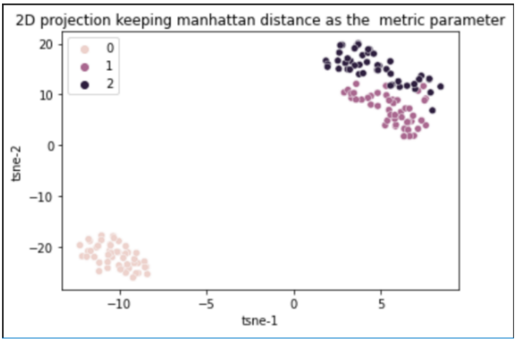


Fig-18:- 2D projection keeping manhattan distance as the metric parameter

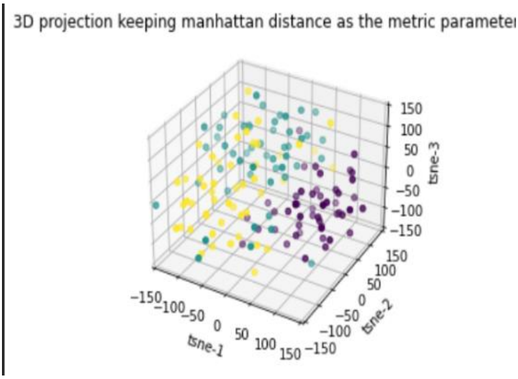


Fig-19:- 3D projection keeping manhattan distance as the metric parameter

Observations:-

1) As we can see , when we change the metric parameter (from euclidean to manhattan) in 3-d plots the clustering becomes better as the green colored cluster comes together which might be due to the change in distance calculation approach.

2) Furthermore in case of the 2-d plots the clusters remain intact as we change the metric Parameter from(Euclidean to manhattan)but their orientation changes as the we change the distance calculation approach.

3. Task-3 Data Classification with linear and non-linear SVM

"Support vector machines" or SVM is supervised learning algorithm which is mostly used in the classification problems .It performs classification by finding the suitable hyperplane that differentiates the classes very well. The support vectors are simply the points which lie along the margins.

Sepal length and Sepal width feature pair is considered and thus the 4-dimensional data is converted to 2-dimensional for this task. Also in this task SVM is used for classifying the data of three pairs of classes.

3.1 Classification of different pairs using linear SVM

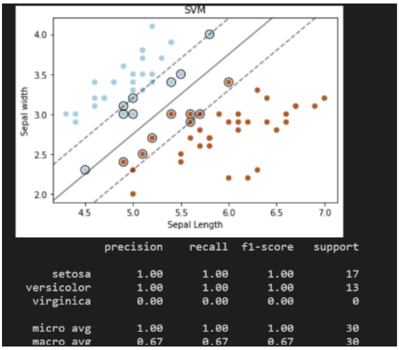


Fig-20:- Max-margin hyperplane and classification report(pair-1)

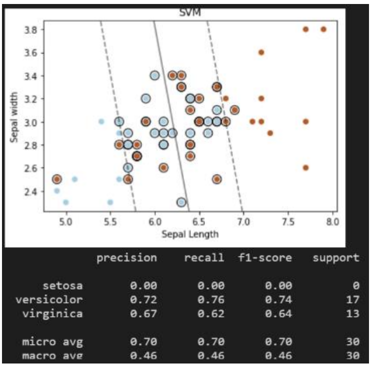
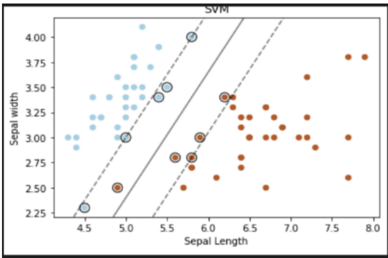


Fig-21:- Max-margin hyperplane and classification report(pair-2)



	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	17
versicolor	0.00	0.00	0.00	0
virginica	1.00	1.00	1.00	13
micro avg	1.00	1.00	1.00	30
macro avg	0.67	0.67	0.67	30

Fig-22:- Max-margin hyperplane and classification report(pair-3)

Observations:

1)By observing SVM's of pair **-**1 and pair**-**3 we see that the corresponding classes are linearly seperable and this is also evident from the F1**-**scores as all of the relevant members are retrieved and all those retrieved belong to the correct class

2)As seen earlier as well the pair**-**2 is not linearly seperable and thus SVM is not performing well on these classes(class**-**2 and 3) and we get low F1**-**score than the pair**-**1 and pair**-**3 scores.

3.2 Comparing the performance of SVM based on different values of C

Since classification report for c=1 has been shown in the earlier part therefore classification report for c=0.001 and c=1000 has been shown in this part.

```
cr(x_sp1,y_sp1,0.001,'linear')
```

	precision	recall	f1-score	support
setosa	0.00	0.00	0.00	17
versicolor	0.43	1.00	0.60	13
virginica	0.00	0.00	0.00	0
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-23:-Classification report of pair1 for c=0.001

```
cr(x_sp1,y_sp1,1000,'linear')
```

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	17
versicolor	1.00	1.00	1.00	13
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 for c=1000

```
cr(x_sp2,y_sp2,0.001,'linear')
```

	precision	recall	f1-score	support
setosa	0.00	0.00	0.00	0
versicolor	0.00	0.00	0.00	17
virginica	0.43	1.00	0.60	13
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 for c=0.001

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	17
versicolor	1.00	1.00	1.00	13
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-26:- Classification report of pair2 for c=1000

```
cr(x_sp3,y_sp3,0.001,'linear')
```

	precision	recall	f1-score	sup
setosa	0.00	0.00	0.00	
versicolor	0.00	0.00	0.00	
virginica	0.43	1.00	0.60	
micro avg	0.43	0.43	0.43	
macro avg	0.14	0.33	0.20	
weighted avg	0.19	0.43	0.26	

```
cr(x_sp3,y_sp3,1000,'linear')
```

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	17
versicolor	0.00	0.00	0.00	0
virginica	1.00	1.00	1.00	13
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-27:-Classification report of pair3 for c=0.001

Fig-28:- Classification report of pair3 for c=1000

Observations:-

For pair-1 a) For c value=0.001,the margin becomes too much such that the f1 score corresponding to the specie setosa comes out to be zero which indicates that the SVM is not able to detect or classify the setosa specie and also performs very bad in classifying the versicolor specie as well.
b)For c value=1000, classification report similar to c=1 is generated the only difference being the thickness of the margin which becomes thin which can be observed in the plot.

For pair-2 a) For c value=0.001,the margin becomes too much such that the f1 score corresponding to the specie versicolor comes out to be zero which indicates that the SVM is not able to detect or classify the versicolor specie and also performs very bad in classifying the virginica specie as well.
b)For c value=1000, classification report similar to c=1 is generated the only difference being the thickness of the margin which becomes thin which can be observed in the plot.

For pair-3 a) For c value=0.001,setosa again has a zero score
b)For c value=1000, classification report similar to c=1 is generated the only difference being the thickness of the margin which becomes thin which can be observed in the plot.

Thus the final conclusion is that at smaller values of c SVM might ignore a particular class of data and at higher values it may overfit the data as the margin becomes smaller and the points come closer to each other.

3.3 Using RBF (Non Linear)as kernel

```
cr(x_sp1,y_sp1,0.001,'rbf')
              precision    recall  f1-score   support

      setosa       0.00      0.00      0.00        17
  versicolor       0.43      1.00      0.60        13
   virginica       0.00      0.00      0.00         0

   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-29:-Classification report of pair1 for c=0.001

```
cr(x_sp1,y_sp1,1,'rbf')
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        17
  versicolor       1.00      1.00      1.00        13
   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-30:- Classification report of pair1 for c=1

```
cr(x_sp1,y_sp1,1,'rbf')
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        17
  versicolor       1.00      1.00      1.00        13
   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-31:- Classification report of pair1 for c=1000

```
cr(x_sp2,y_sp2,.001,'rbf')
              precision    recall  f1-score   support

      setosa       0.00      0.00      0.00         0
  versicolor       0.00      0.00      0.00        17
   virginica       0.43      1.00      0.60        13

   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-32:-Classification report of pair2 for c=0.001

```
cr(x_sp2,y_sp2,1,'rbf')
              precision    recall  f1-score   support

      setosa       0.00      0.00      0.00         0
  versicolor       0.65      0.76      0.70        17
   virginica       0.60      0.46      0.52        13

   micro avg       0.63      0.63      0.63        30
   macro avg       0.42      0.41      0.41        30
weighted avg       0.63      0.63      0.62        30
```

Fig-33:- Classification report of pair2 for c=1

```
cr(x_sp2,y_sp2,1000,'rbf')
              precision    recall  f1-score   support

      setosa       0.00      0.00      0.00         0
  versicolor       0.65      0.65      0.65        17
   virginica       0.54      0.54      0.54        13

   micro avg       0.60      0.60      0.60        30
   macro avg       0.40      0.40      0.40        30
weighted avg       0.60      0.60      0.60        30
```

Fig-34:-Classification report of pair2 for c=1000

```
cr(x_sp3,y_sp3,.001,'rbf')

              precision    recall  f1-score   support

      setosa       0.00      0.00      0.00        17
  versicolor       0.00      0.00      0.00         0
   virginica       0.43      1.00      0.60        13

   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-35:-Classification report of pair3 for c=0.001

```
cr(x_sp3,y_sp3,1,'rbf')

              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        17
  versicolor       0.00      0.00      0.00         0
   virginica       1.00      1.00      1.00        13

   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-36:- Classification report of pair3 for c=1

```
cr(x_sp3,y_sp3,1000,'rbf')

              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        17
  versicolor       0.00      0.00      0.00         0
   virginica       1.00      1.00      1.00        13

   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-37:- Classification report of pair3 for c=1000

Observations:-

For pair-1 a)For c value=1 and 1000 the classes are linearly seperable hence there is no difference in the classification report which can be seen form the above reports as the only difference occurs in the decision boundary. For c=0.001 the non linear kernel in this case adds some complexity due to which underfitting is prevented.

For pair-2 a)Besides performing good on the linearly seperable classes rbf performs good in classifying the non seperable data and thus gives better F1 scores than the linear kernel It can be seen that the highest F1 score comes out in the case of c=1.

For pair-3 a)Again since the classes in this pair are linearly seperable the RBF kernel gives similar classification report as given by the linear kernel since the difference occurs only in the decision boundary.

4.Conclusion and Learnings

During the course of this assignment I got hands on experience on coding different dimension reduction techniques and their application in better visualisation and understanding of data. Such as the usage of PCA in reducing features which can be used to get rid of unwanted and redundant features for enhancing the predictability of our model. The usefull insight in the creating the SVM classifier which can be very helpful visual tool for classifying linearly seperable and non-linearly sperable using the linear and non linear kernels respectively was great.

5. References

[1] https://scikit-learn.org/stable/auto_examples/svm/plot_separating_hyperplane.html

[2] https://www.machinecurve.com/index.php/2020/05/05/how-to-visualize-support-vectors-of-your-svm-classifier/

[3] http://scikit-learn.org/stable/auto_examples/applications/plot_face_recognition.html