MACHINE LEARNING
(Programming Assignment 2)

Shubhangi Ghosh
EE15B129
Department of Electrical Engineering

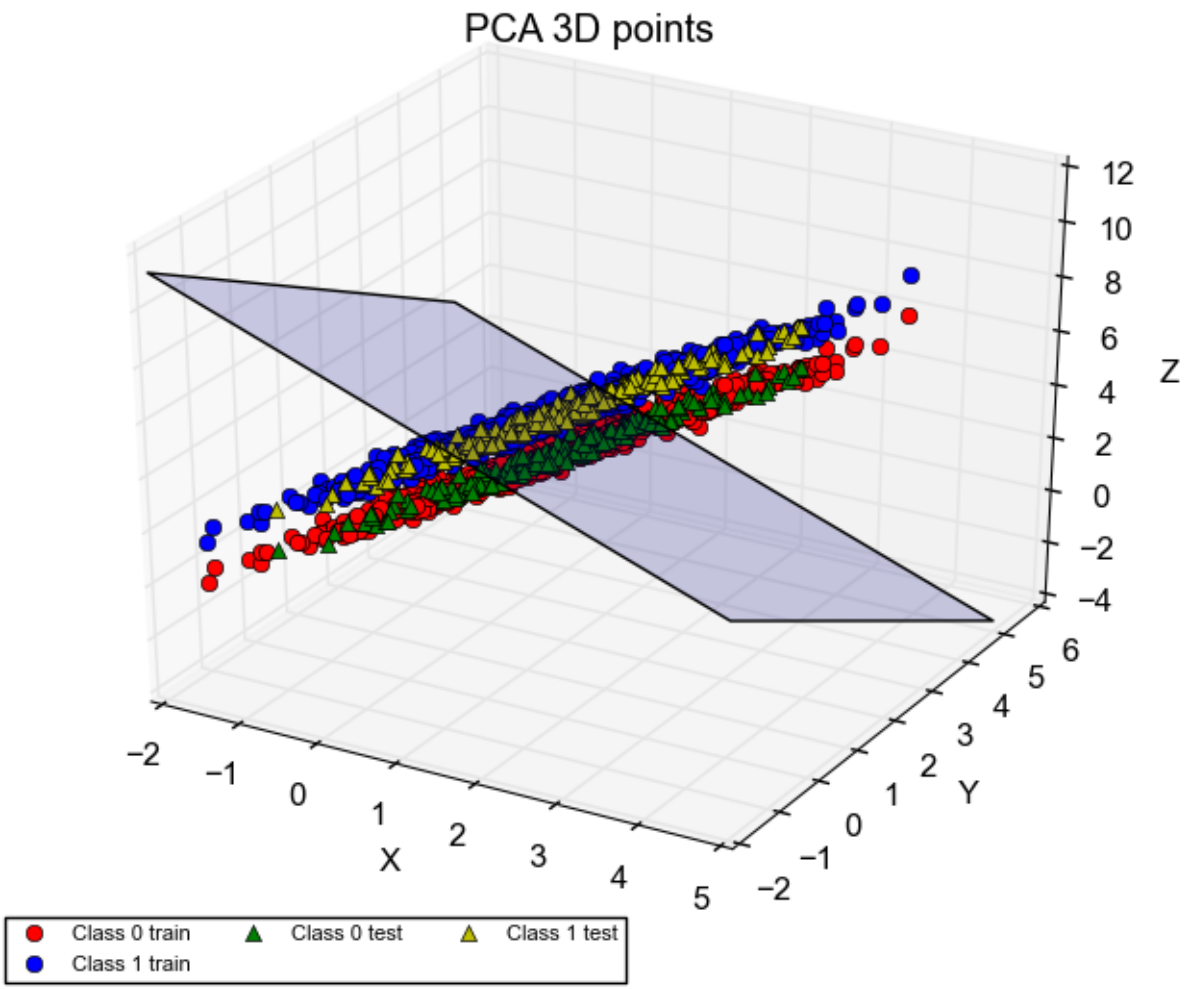
October 10, 2017

1 QUESTION - 1 PCA(PRINCIPAL COMPONENTS ANALYSIS):

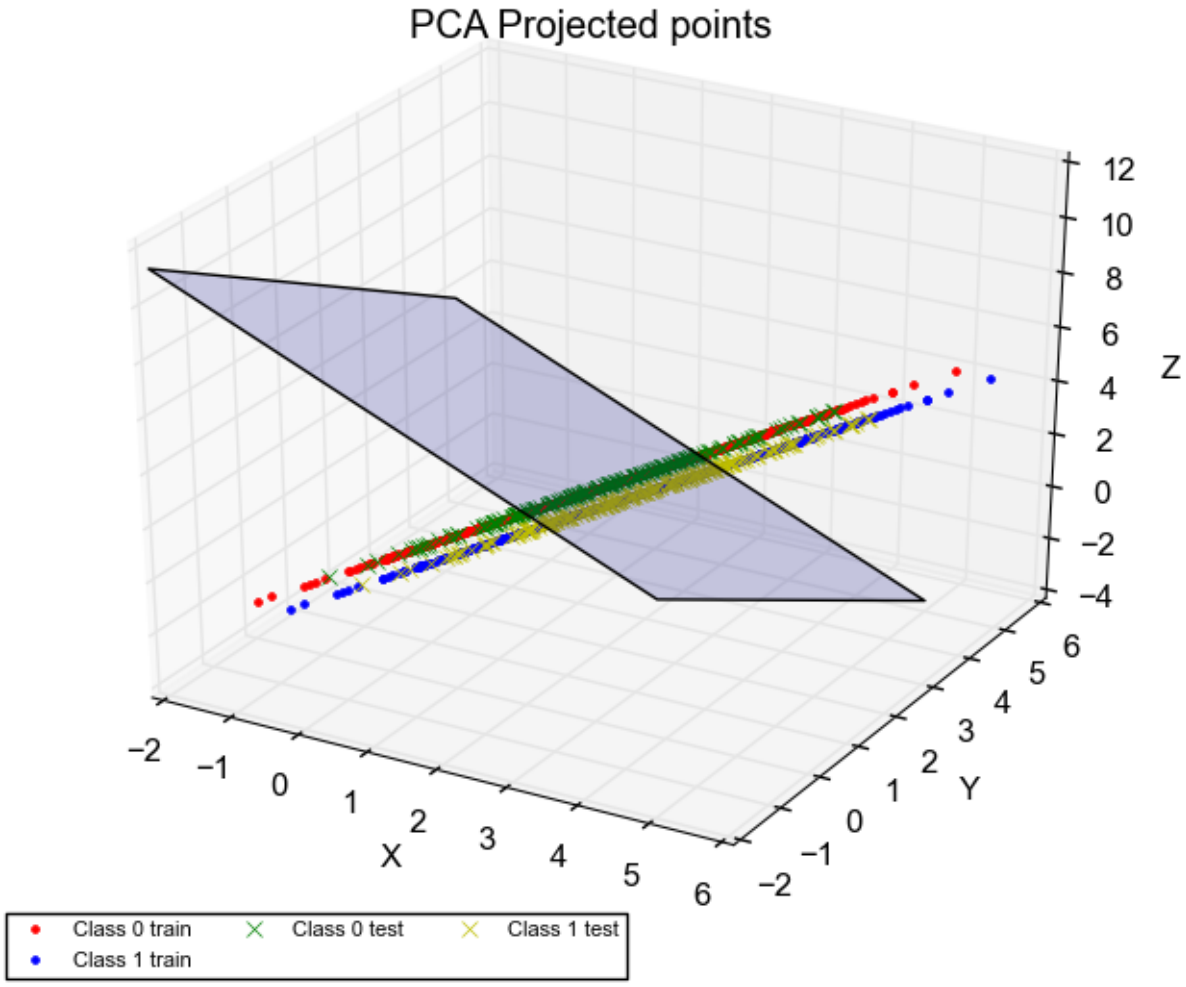
1.1 CLASSIFICATION MEASURES:

Class	Precision	Recall	F-Measure
Class 0	59%	59%	59%
Class 1	59%	58%	59%
Average	59%	58.5%	59%

1.2 3-D PLOT OF DATASET AND DECISION BOUNDARY:



1.3 3-D PLOT OF DATASET IN PROJECTED SPACE AND DECISION BOUNDARY:



1.4 INFERENCES:

1. PCA projects datapoints in the direction of maximum variance of all datapoints in the dataset.
2. Data should be standardised before PCA since it is a variance maximising exercise, and not standardising may give wrong features.
3. PCA also chooses dimensions which have less correlation among them. Here we have chosen the first PCA direction which has the maximum variance.
4. The PCA direction was found to be along $-0.59\hat{x} + -0.59\hat{y} + -0.54\hat{z}$.
5. Linear Regression was performed on Class points projected along this direction. This wasn't very helpful different class points in this case weren't separable in PCA direction.
6. The decision hypeplane was plotted with the PCA direction as its normal.
7. The intercept of the hyperplane from the origin was calculated using coefficients and intercepts obtained from discriminator functions of LR.
8. $$intercept = \frac{intercept\,of\,Class\,2 - intercept\,of\,Class\,1}{coeff\,of\,Class\,2 - coeff\,of\,Class\,1}$$

$$= \frac{0.233 - 0.766}{-0.070 - 0.070}$$
$$= 3.807$$

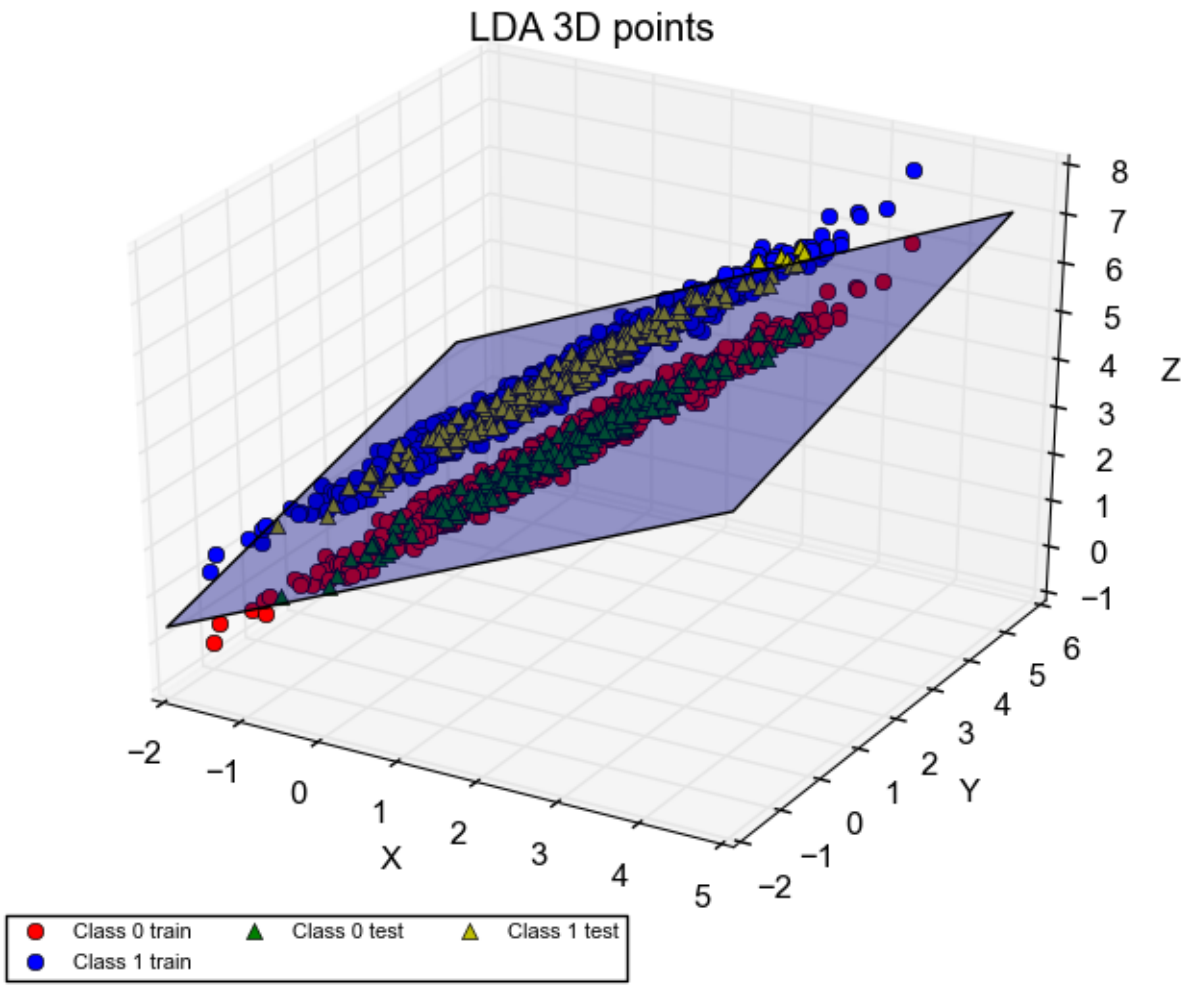
9. The equation of the hyperplane was therefore -0.59x -0.59y -0.54z + 3.807 = 0.

2 QUESTION - 2: LDA : LINEAR DISCRIMINANT ANALYSIS

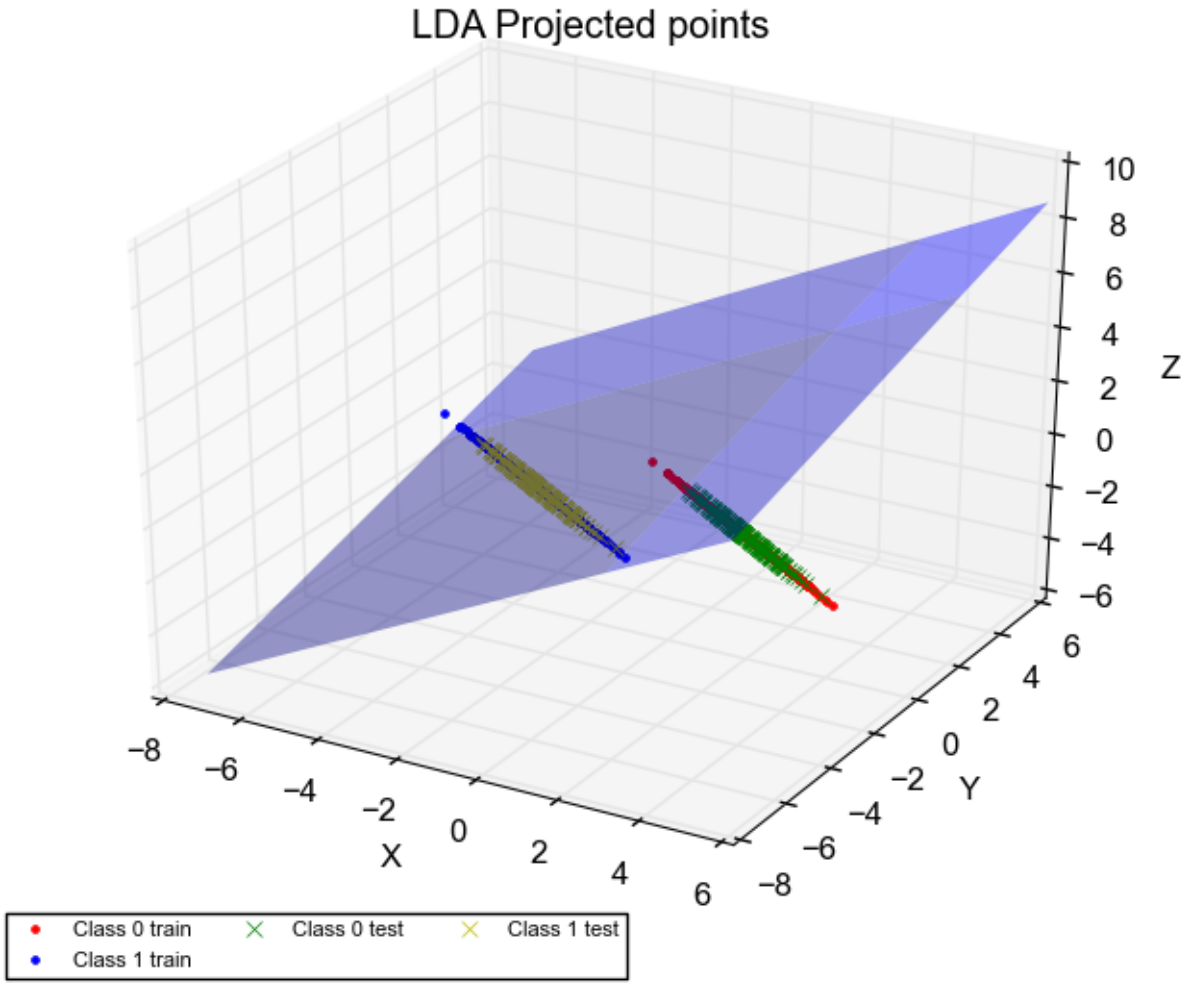
2.1 CLASSIFICATION MEASURES:

Class	Precision	Recall	F-Measure
Class 0	100%	100%	100%
Class 1	100%	100%	100%
Average	100%	100%	100%

2.2 3-D PLOT OF DATASET AND DECISION BOUNDARY:



2.3 3-D PLOT OF DATASET IN PROJECTED SPACE AND DECISION BOUNDARY:



From plots we see that Class 0(red and green) points lie below the separating hyperplane and Class 1 points(blue and yellow) lie above the separating hyperplane.

2.4 INFERENCES:

1. LDA chooses directions in the projected space such that:
   (a) Between-class variance is maximised.
   (b) Within-class variance is minimised.
2. This interpretation can be drawn from Fischer's LDA.
3. From LDA function attributes we find the LDA projected direction to be along $-41.20\hat{x} + -16.89\hat{y} + 56.17\hat{z}$.
4. From LDA function attributes, the LDA hyperplane is found to be along -41.20x -16.89y +56.17z - 135.07 = 0.
5. The PCA direction has been written in unit vector format. When we find the dot product between the LDA direction and the PCA direction, we find it to be almost equal to 0. Hence, these directions are perpendicular to each other.

6. In the LDA direction, we see that datapoints from different classes are actually separable, since LDA maximises between-class variance.

7. PCA just maximises the variance of all datapoints in its dominant direction without taking into account their class. So, we may or may not end up with a direction where between class variance is high, i.e. classes are separable.

8. In this case with PCA, we end up with a direction along which between class variance is very low. Hence LDA is the clear choice for this question.

9. As we see, LDA 100% precision, recall and F-measure for both classes.

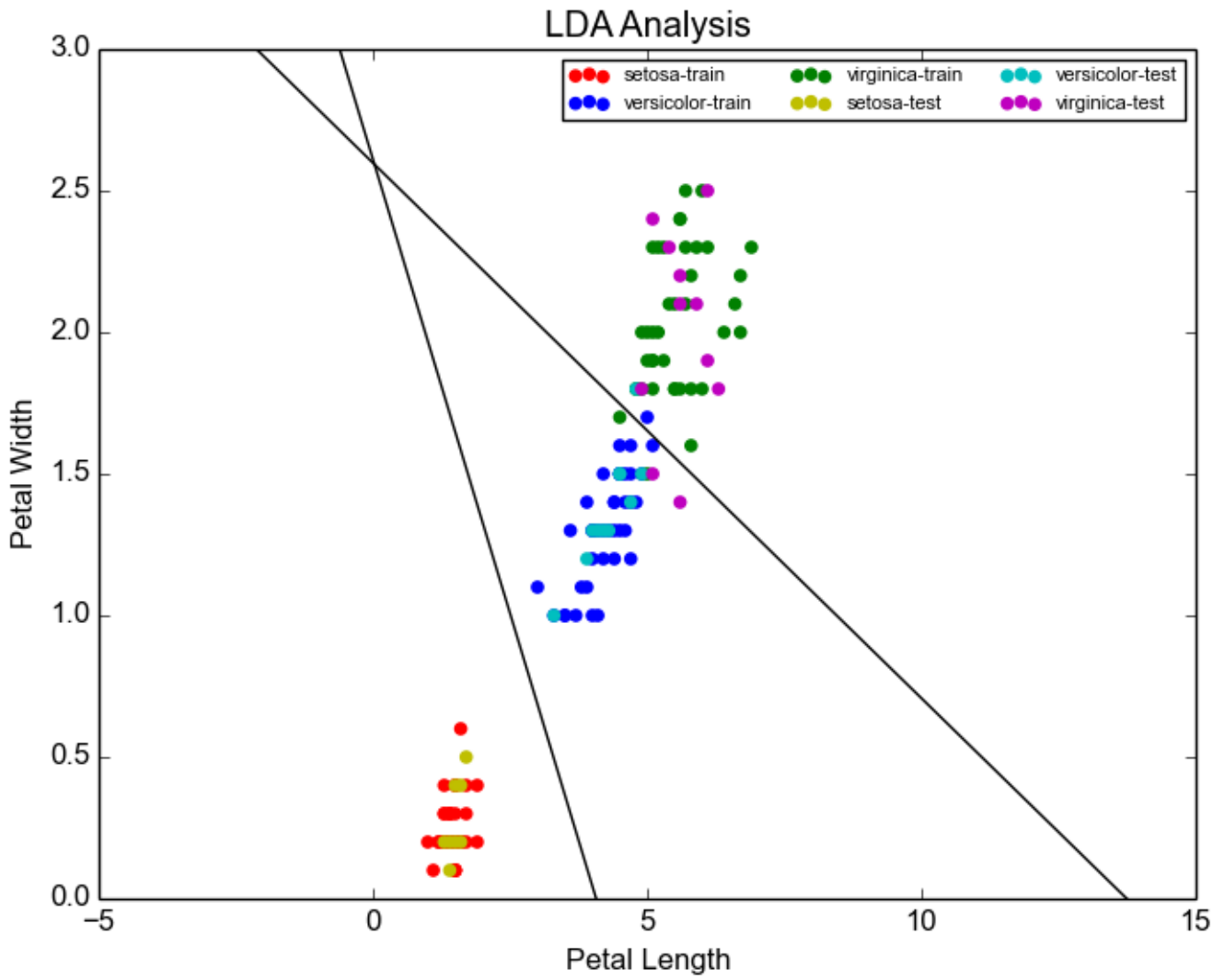
3 QUESTION 3:

3.1 LDA: LINEAR DISCRIMINANT ANALYSIS:

3.1.1 CLASSIFICATION REPORT:

CLASS	PRECISION	RECALL	F-MEASURE
Iris-setosa	100%	100%	100%
Iris-versicolor	83%	91%	87%
Iris-Virginica	90%	82%	86%
Average	90%	90%	90%

3.1.2 DECISION BOUNDARY:



3.1.3 NOTE:

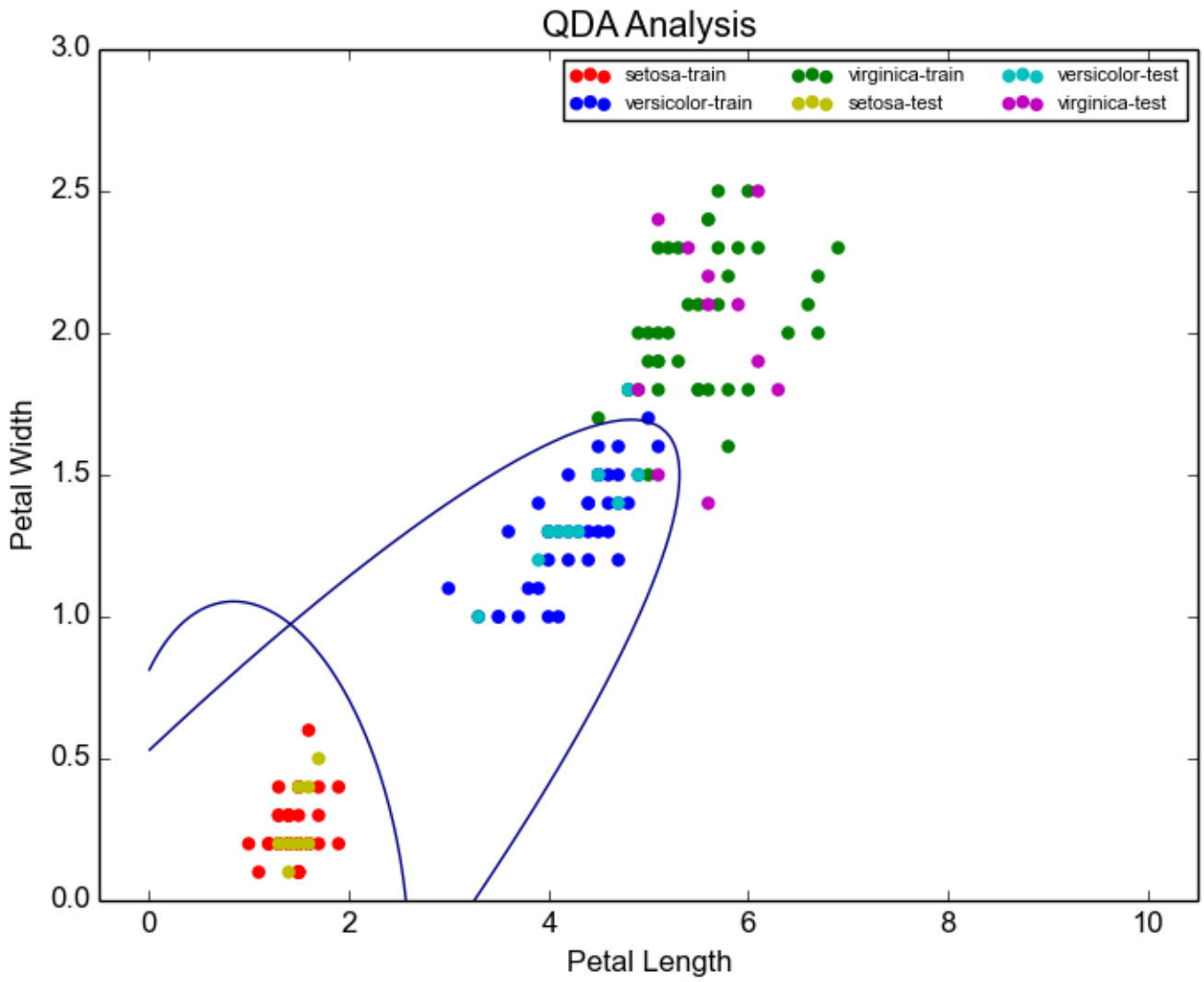
1. Decision boundary has been plotted similarly as in Question 2.
2. Decision boundary between Class a and Class b: $\delta_a - \delta_b = 0.$, where δ_a and δ_b are discriminant functions for Classes a and b.
3. Decision boundary between: setosa and versicolor: 10.64x + 16.59y - 43.32 = 0
   versicolor and virginica: 3.02x + 15.99y - 41.54 = 0.

3.2 QDA: QUADRATIC DISCRIMINANT ANALYSIS:

3.2.1 CLASSIFICATION REPORT:

CLASS	PRECISION	RECALL	F-MEASURE
Iris-setosa	100%	100%	100%
Iris-versicolor	91%	91%	91%
Iris-Virginica	91%	91%	91%
Average	93%	93%	93%

3.2.2 DECISION BOUNDARY:



3.2.3 NOTE:

To plot decision boundaries:

1. Decision boundary between Class a and Class b: $\delta_a - \delta_b = 0.$, where δ_a and δ_b are discriminant functions for Classes a and b.

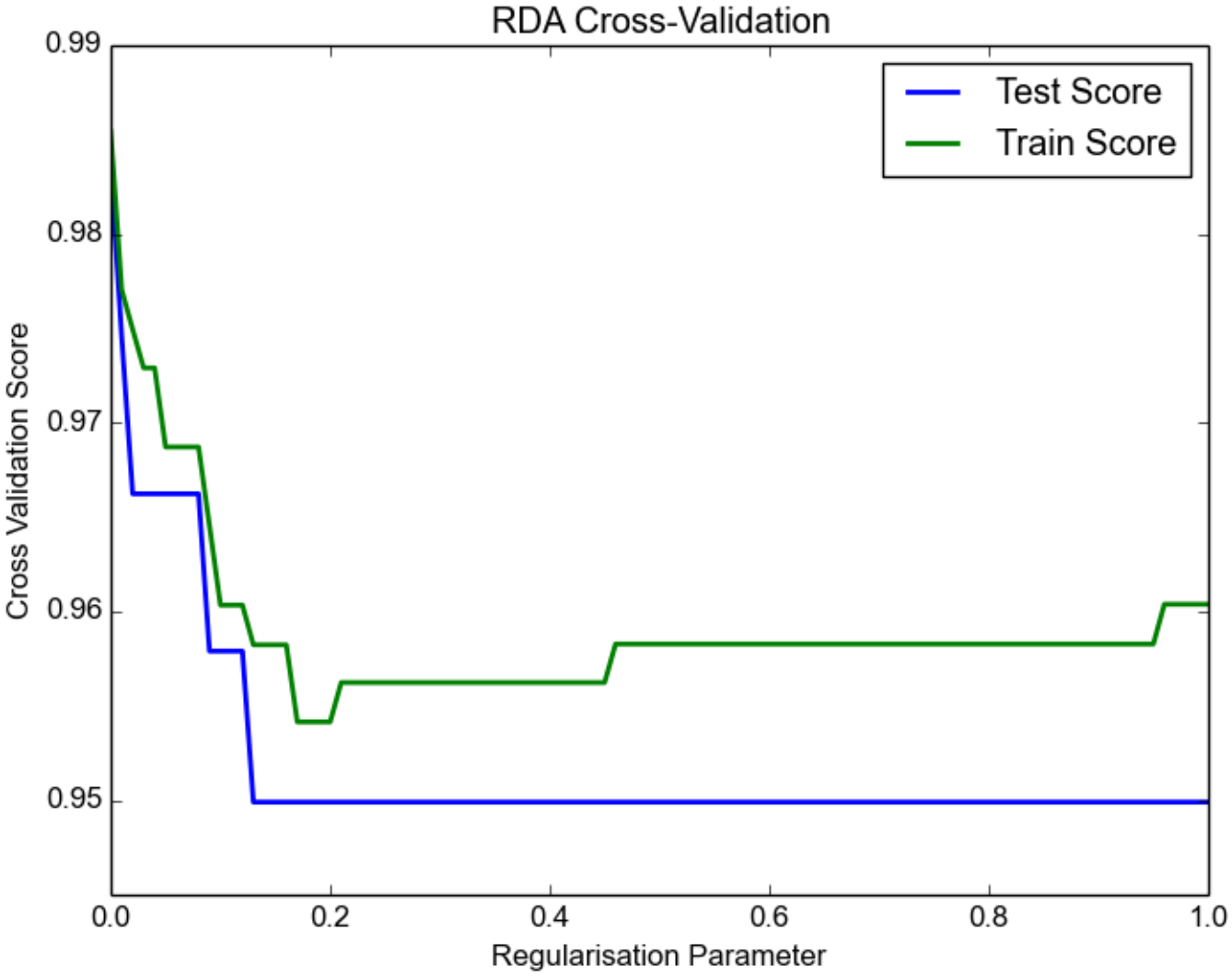
3.3 RDA: REGULARISED DISCRIMINANT ANALYSIS:

3.3.1 CLASSIFICATION REPORT:

CLASS	PRECISION	RECALL	F-MEASURE
Iris-setosa	100%	100%	100%
Iris-versicolor	91%	91%	91%
Iris-Virginica	91%	91%	91%
Average	93%	93%	93%

3.3.2 CHOOSING REGULARISATION PARAMETER USING CROSS-VALIDATION:

1. Test score has been maximised by performing 5-fold cross-validation on the dataset.



2. Regularisation parameter, $\alpha = 0$, indicates LDA, and $\alpha = 1$, approaches RDA.

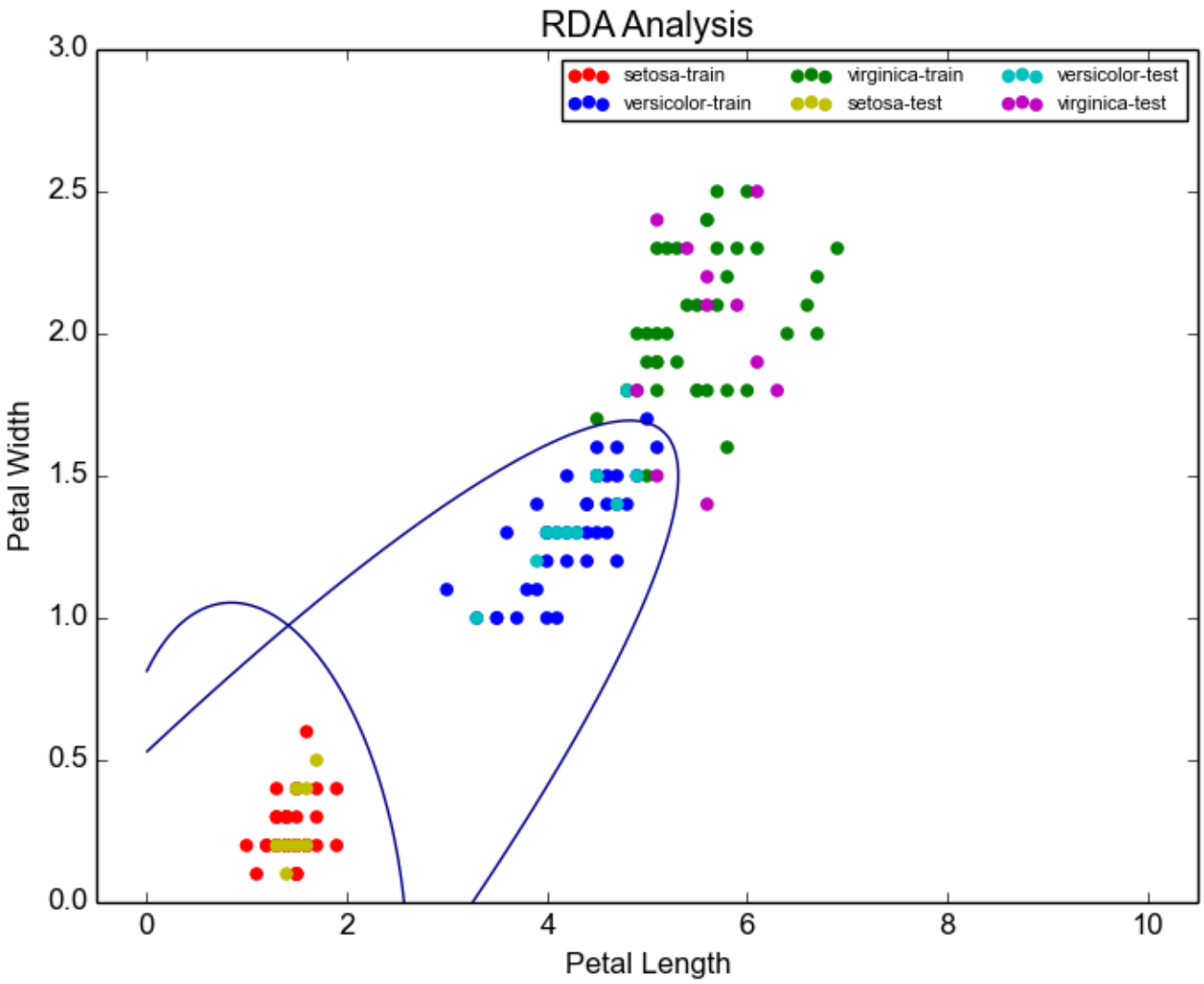
3. We see that test score(accuracy) sharply decreases from $\alpha = 0$, to $\alpha = 0.2$, and then rises in small steps at $\alpha = 0.23$, $\alpha = 0.46$, $\alpha = 0.96$.

4. Thus, we find LDA seems to be a better choice for this case than QDA or RDA.

5. I have choses $\alpha = 0.6$, to illustrate RDA since it lies on the second plateau and test score is kind of improved.

6. I haven't chosen $\alpha = 0$, which should ideally be chosen, since this performance has already been illustrated in the LDA Analysis, and not chosen $\alpha = 0.96$ for higher test score since that approaches QDA, which has already been illustrated.

3.3.3 DECISION BOUNDARY:



3.3.4 NOTE:

To plot decision boundaries:

Decision boundary between Class a and Class b: $\delta_a - \delta_b = 0.$, where δ_a and δ_b are discriminant functions for Classes a and b.

3.3.5 INFERENCE:

1. Here, on plotting the dataset, we realise that the dataset is highly linear.
2. If there are very few training observations(like 120 in this case), and high correlation between attributes(mentioned on data website that there is high correlation between petal length and petal width), minimising variance between classifiers becomes crucial.
3. Thus, LDA has the higher edge in this case, since LDA has fewer parameters, one common covariance matrix, unlike QDA which has a separate covariance matrix for each class, thereby has lesser variance.
4. However, if the uniform covariance criterion is highly off, which doesn't seem to be the case here(degree of spread of data points in each class seems similar from plot), QDA has the higher edge.

5. From the above points, we can say that LDA has the higher edge in this case and for most other simple datasets.

6. When the dataset has more no. of datapoints, with attributes which have low correlation, classes with non-uniform covariance, LDA may suffer higher bias and QDA maybe a more suitable classifier.

7. But to also minimise variance of the classifier and overfitting on training set, some kind of trade-off between LDA and QDA can be established through a regularisation parameter, as in RDA. where RDA minimises the weightage to the extra parameters which introduce high variance and gives some weightage to the common covariance matrix.

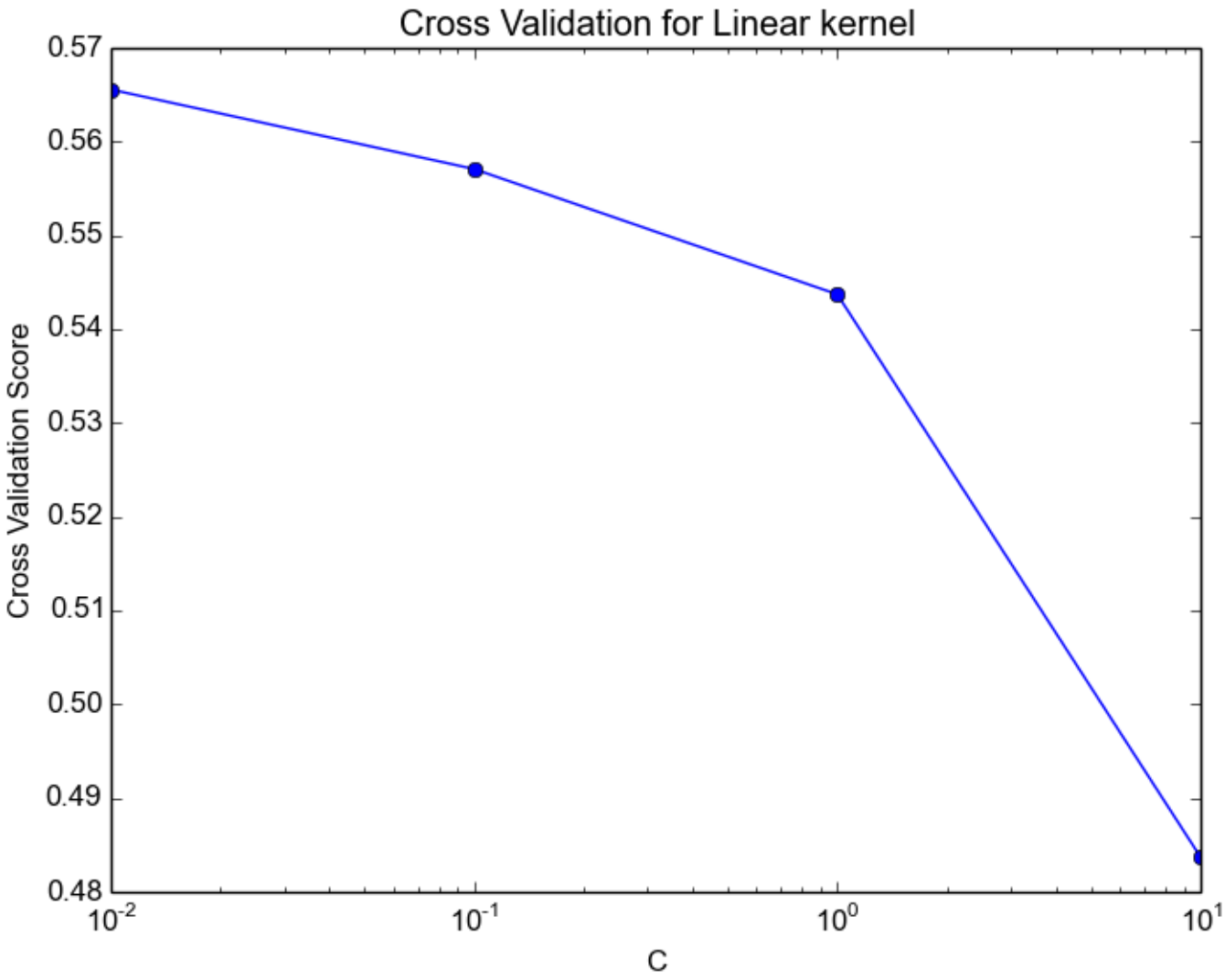
$$\hat{\Sigma_k} = \alpha\hat{\Sigma_k} + (1-\alpha)\hat{\Sigma}$$

8. But here this is completely unnecessary as LDA is a very suitable classifier for this dataset.

4 QUESTION 4: SVM(SUPPORT VECTOR MACHINES) KERNELS:

Models have been saved in .pkl files.

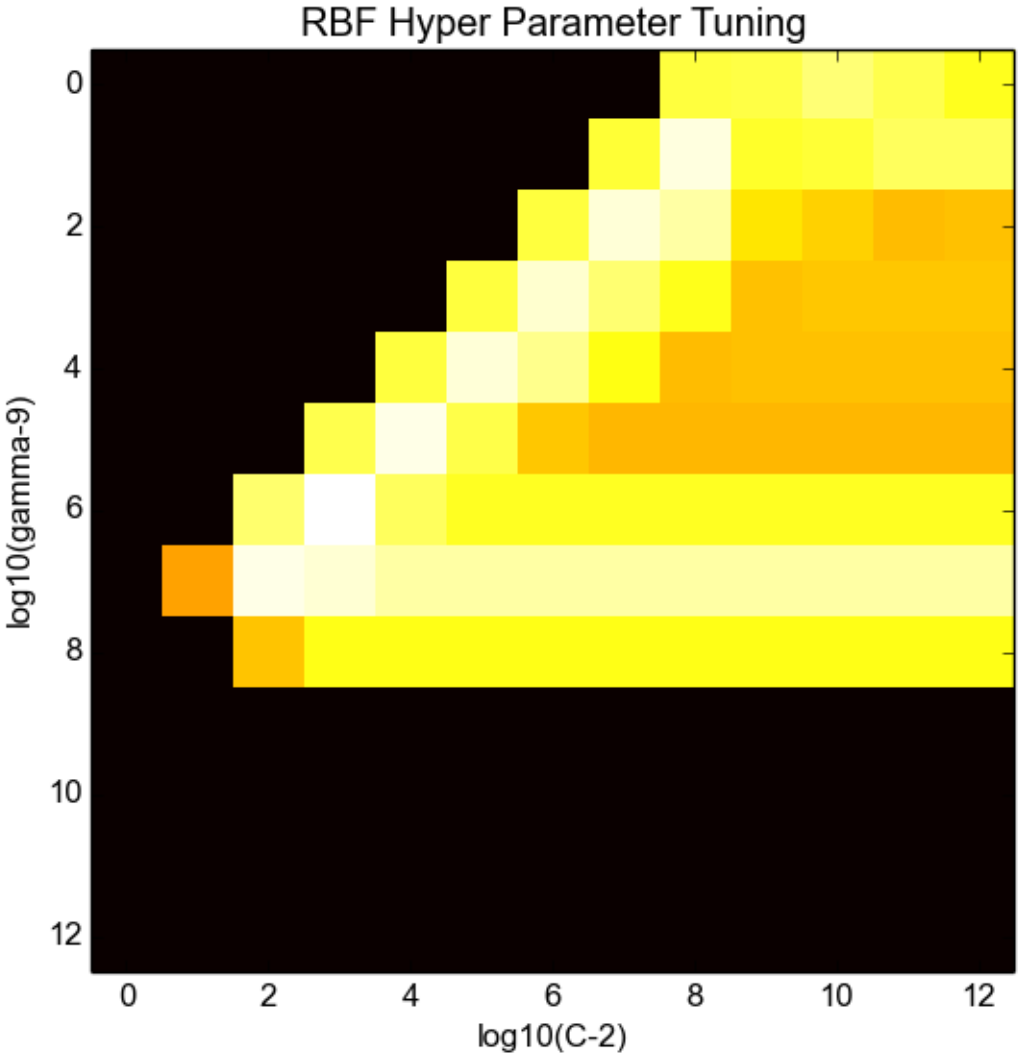
4.1 LINEAR KERNEL:



4.1.1 INFERENCE:

1. C is a measure of error margin.
2. So, for linear kernel we find best fit for smaller error margin.
3. C=0.01, Best fit CV(Cross Validation) Score: 56.557%.

4.2 RBF/GAUSSIAN KERNEL:

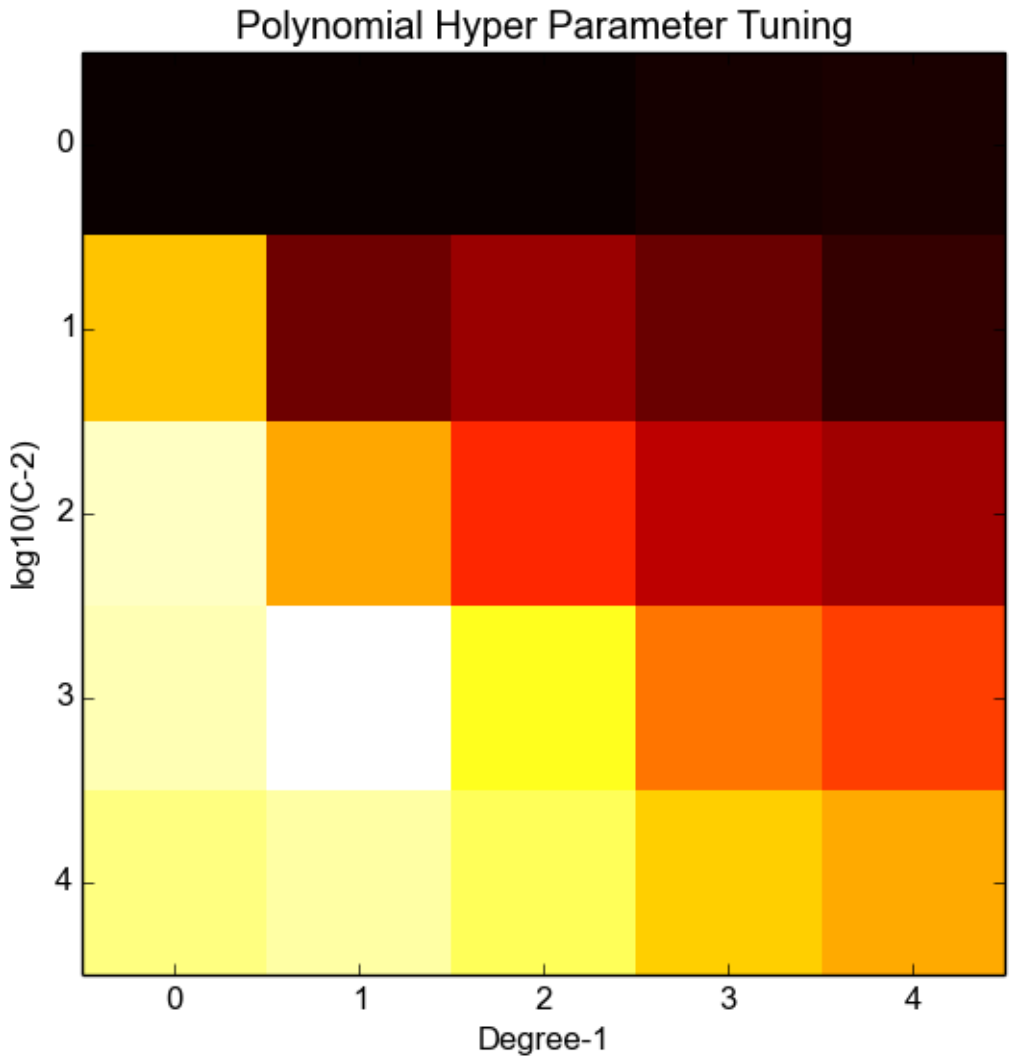


4.2.1 INFERENCE:

1. The squares are black where CV score is minimum, white where maximum.
2. Both axes are in logarithmic scale x–> log10(C-2), y–> log10(C-9).
3. Gamma inversely models radius of influence of support vectors.
4. For a very large gamma, area of influence of SV(Support vectors) is too small, and no amount of regularisation with C can prevent overfitting.
5. If gamma is too small, model cannot estimate the distribution of the data well.
6. So we find our white spots along a diagonal.
7. C still captures error margin.

8. C = 10.0, gamma = 0.01, CV score = 59.445%.

4.3 POLYNOMIAL KERNEL:



4.3.1 BEST FIT PARAMETERS:

C= 10.0 Degree =2, CV score = 58.146%.

4.4 SIGMOIDAL KERNEL:

No plot because more than two hyperparameters.

4.4.1 BEST FIT PARAMETERS:

C =0.1, gamma =0.1 coeff0 =0.0
coeff 0 and gamma are relative weightages to the inner product and constant term.

5 QUESTION 5: DECISION TREES:

5.1 DECISION TREE WITHOUT REDUCED ERROR PRUNING:

5.1.1 VISUALISED TREE:

J48 pruned tree ——————

```
odor = a: e (400.0)
odor = l: e (400.0)
odor = c: p (192.0)
odor = y: p (576.0)
odor = f: p (2160.0)
odor = m: p (36.0)
odor = n | spore-print-color = k: e (1296.0)
                | spore-print-color = n: e (1344.0)
                | spore-print-color = b: e (48.0)
                | spore-print-color = h: e (48.0)
                | spore-print-color = r: p (72.0)
                | spore-print-color = o: e (48.0)
                | spore-print-color = u: e (0.0)
                | spore-print-color = w | | gill-size = b: e (528.0)
                                                | | gill-size = n | | | gill-spacing = c: p (32.0)
                                                                        | | | gill-spacing = w
                                                | | | | population = a: e (0.0)
                                                | | | | population = c: p (16.0)
                                                | | | | population = n: e (0.0)
                                                | | | | population = s: e (0.0)
                                                | | | | population = v: e (48.0)
                                                | | | | population = y: e (0.0)
                                                                        | | | gill-spacing = d: p (0.0)
                | spore-print-color = y: e (48.0)
odor = p: p (256.0)
odor = s: p (576.0)
```

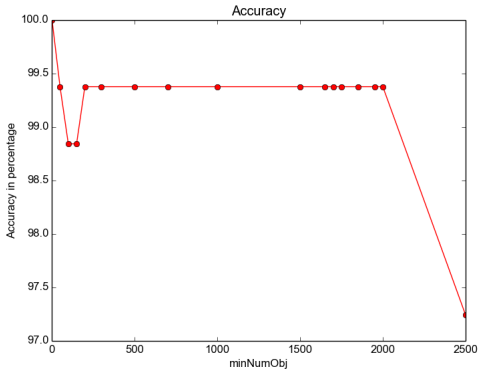
5.1.2 CLASSIFICATION MEASURES:

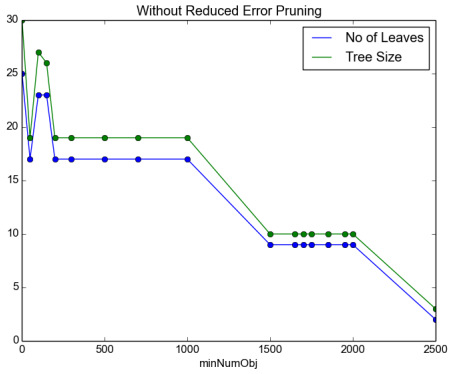
Class	Precision	Recall	F-Measure
Poisonous	100%	100%	100%
Edible	100%	100%	100%
Average	100%	100%	100%

Accuracy: 100%

5.2 DECISION TREE WITHOUT REDUCED ERROR PRUNING:

5.2.1 VARYING minNumObj





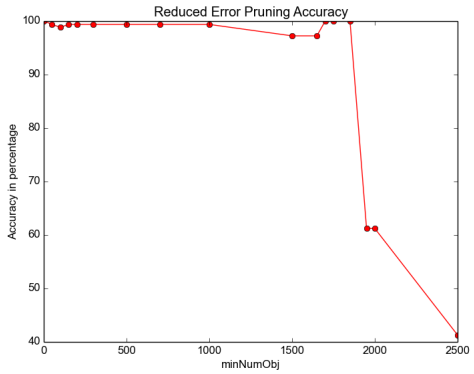
5.2.2 INFERENCE:

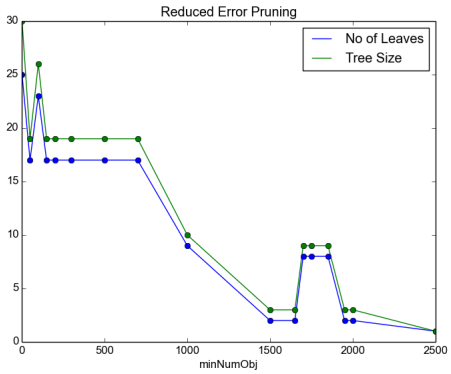
1. By using varying minNumObj(minimum no. of datapoints in a leaf), as a stopping criterion, we see that we have effectively reduced the size of the tree a lot, while keeping accuracy close to 100%.

2. Also, if minNumObj is too small, in our case 2 datapoints per leaf, or if we reduce it to a single datapoint per leaf, while we may reduce training error to zero, we may have cases where test error is pretty high.

5.3 DECISION TREE WITH REDUCED ERROR PRUNING:

5.3.1 EFFECT OF minNumObj WITH REDUCED ERROR PRUNING:





5.3.2 INFERENCE:

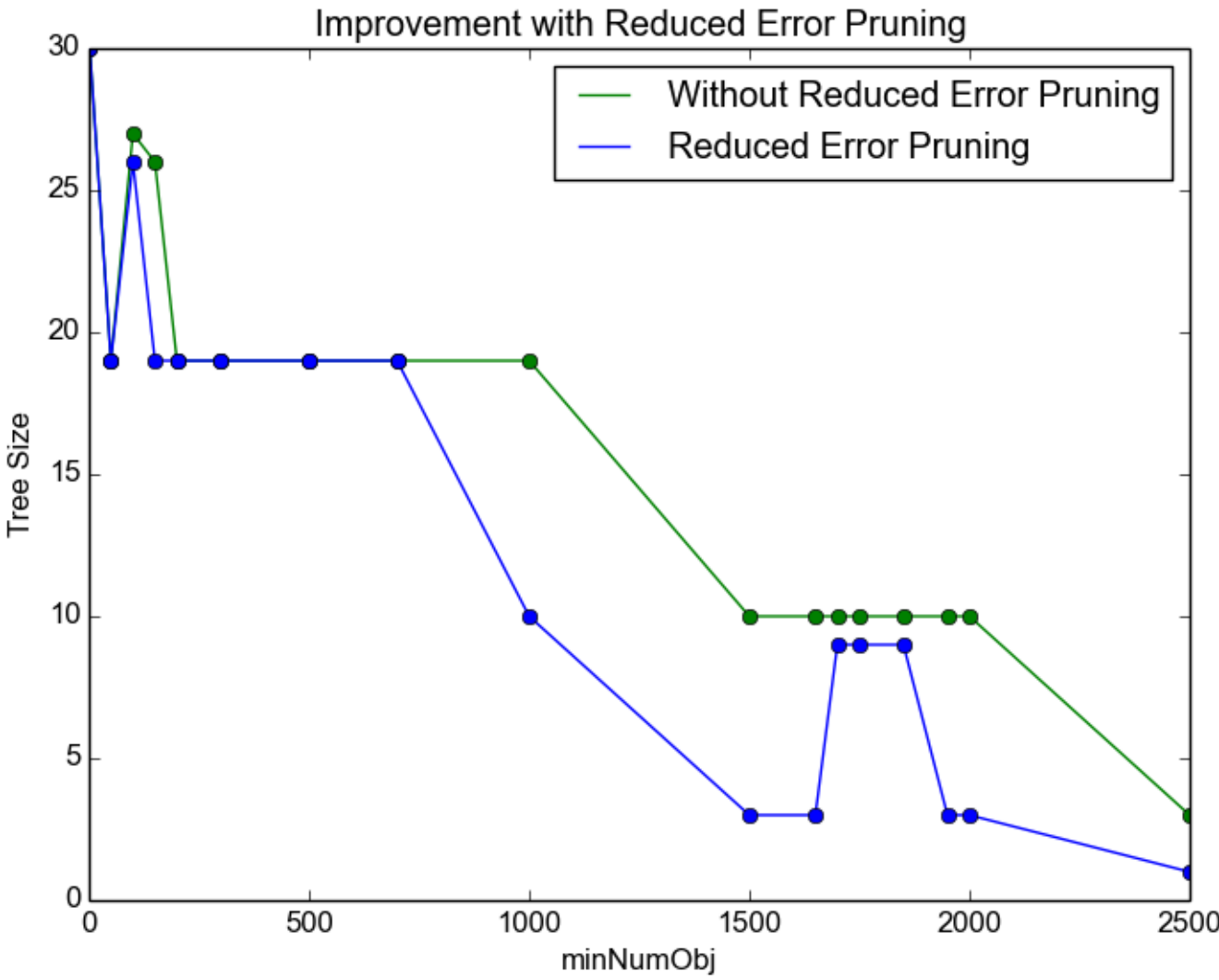
1. From the two plots of Accuracy and size of tree, we notice that upto minNumObj = around 1900, size and number of leaves of the tree can be significantly reduced, while still maintaining a near 100% accuracy.

2. In reduced error pruning, we first grow a tree of a very large size, and very small leaf size(say, minNumObj =2).

3. Then, we prune branches such that classification error is reduced or kept the same.

4. But, in our case we are also fixing minNumObj for the pruned tree. So, to meet the requirements of minNumObj, we may end up pruning a branch which leads to very high classification error or very poor accuracy.

5. But by comparing our current graphs with the graphs for without reduced error pruning we see that in some cases we have reduced the tree size very much, compared to before, which is what we desire.



6. But reduced error pruning also increases the variance of the model a lot depending on the validation set chsen. And trees already suffer from high variance. So reduced error pruning may not be the most desirable pruning technique. Other techniques like cost complexity pruning may be preferred.

7. So, reduced error pruning is very unpredictable or unreliable. A case-wise analysis needs to be done to obtain best tree.

5.4 DECISION TREE WITH BEST PERFORMANCE

Performance considered w.r.t size of tree and accuracy. Two trees seems to be very interpretable and give good accuracy.

5.4.1 TREE 1

minNumObj = 1650
Reduced Error Pruning = True
Accuracy = 97.242%
F1 Score = 97.6%

J48 pruned tree ——————

gill-size = b: e (5612.0/1692.0)
gill-size = n: p (2512.0/288.0)

5.4.2 TREE 2

This has more no. of leaves but still splits on only one feature and has higher accuracy(100%).
minNumObj = 1750
Reduced Error Pruning = True
Accuracy = 100%
F1 Score = 100%

J48 pruned tree ——————
ring-type = c: e (0.0)
ring-type = e: p (1860.0/674.0)
ring-type = f: e (40.0)
ring-type = l: p (850.0)
ring-type = n: p (21.0)
ring-type = p: e (2645.0/554.0)
ring-type = s: e (0.0)
ring-type = z: e (0.0)

5.5 MOST IMPORTANT FEATURES:

From all the trees visualised, the most important features seem to be:

1. ring-type
2. gill-size

Other important features from full grown tree:

1. odor
2. spore-print-color
3. gill-size
4. population