Assignment 2

Koustav Das (2015CSB1017)
Naman Goyal (2015CSB1021)

Part 1
Linear Regression

Q6)

a) The most significant attributes are Diameter, Whole Weight and Shucked Weight, Shell Weight.

b) For $\lambda = 0.5$

When we remove some non-significant attributes the train error increases while test error decreases. This shows the model now closely resembles the general trend and less is overfitted according to train data.

<u>Results Over Original Attribute</u>

Train Error = 2.278384, Test Error=2.450312

Weights =

9.8287	(Constant)
0.1583	(Female)
-0.3012	(Infant)
0.1397	(Male)
-0.5095	(Length)
1.5510	(Diameter)
0.4709	(Height)
3.4883	(Whole weight)
-3.8280	(Shucked weight)
-1.1780	(Viscera weight)
1.5082	(Shell weight)

<u>Results Over Pruned Attributes</u>

Train Error = 2.289375, Test Error=2.450010

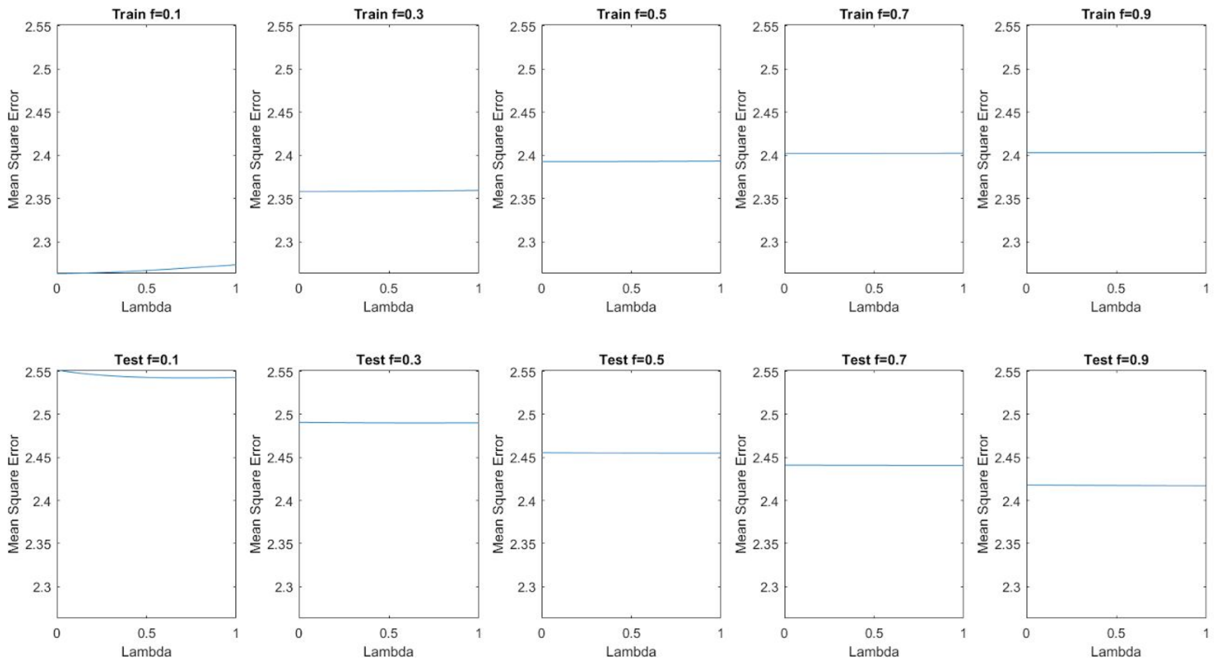
Weight =

9.8287	(Constant)
0	(Female)
-0.3012	(Infant)
0	(Male)

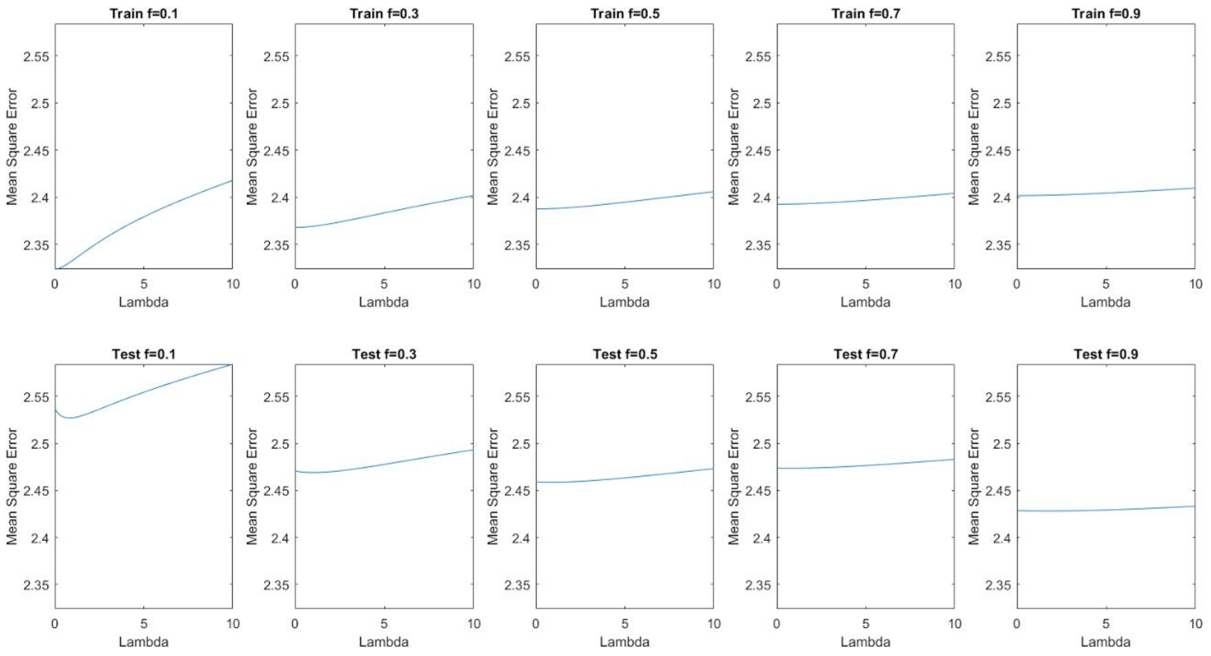
-0.5095	(Length)
1.5510	(Diameter)
0.4709	(Height)
3.4883	(Whole weight)
-3.8280	(Shucked weight)
-1.1780	(Viscera weight)
1.5082	(Shell weight)

Q7 and Q8)

In this question we would like to see the result when we vary λ values over a small range (suppose from 0 to 1) and what happens when we vary λ values over a large range of values.



Varying λ over small range of values



Varying λ over large range of values

a) Does the effect of λ on error change for different partitions of the data into training and testing sets?

Ans: There is an inflection point in graph around λ~ (0,2) for train error. In examples with smaller fraction of training set this point is nearer to zero for smaller fraction while in examples with greater fraction of training set it closer to 2 .

Inflection point in test set is nearly around λ~ (0.5,1) for all fractions.

Before inflection Point

- As lambda increases Train Error decreases slightly for all fractions.
- As lambda increases Test Error decreases significantly for all fractions.

Reason: Before inflection point increase of lambda reduces over fitting by making some less significant attributes contribution closer to zero.

After Inflection Point

- As lambda increases Train Error increases slightly for all fractions.
- As lambda increases Test Error increases significantly for all fractions.

Reason: After inflection point increase of lambda increases underfitting by making many attributes (both significant and non-significant) contribution closer to zero.

b) How do we know if we have learned a good model

We know we have learned good model if Mean squared error for train and test data is less. If error is less our model effectively approximates the actual model closely. Else mean squared error would very large.

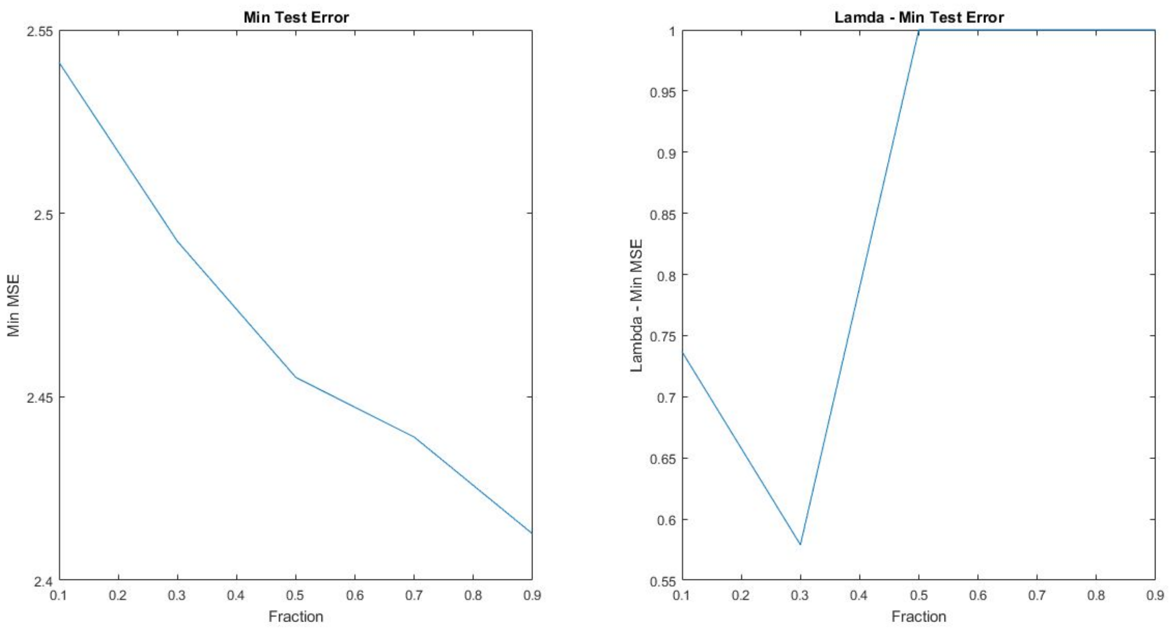
Also mean train error and test error should be have very close values.

If Train Error < Test Error
This shows overfitting

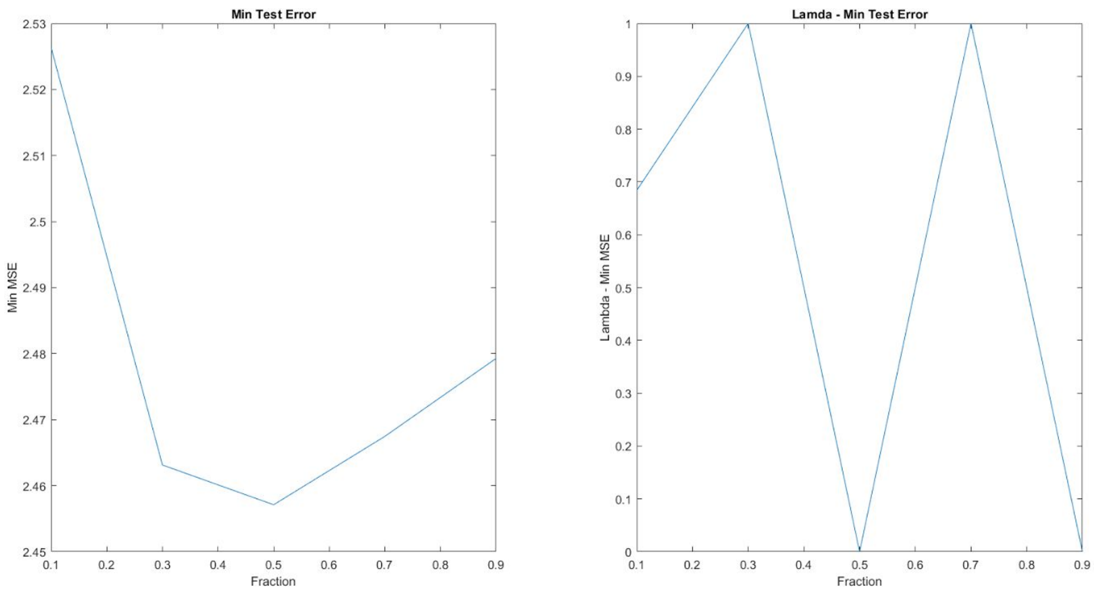
Train Error > Test Error
This shows underfitting

Train Error ~ Test Error
This gives optimal answer i.e. good model

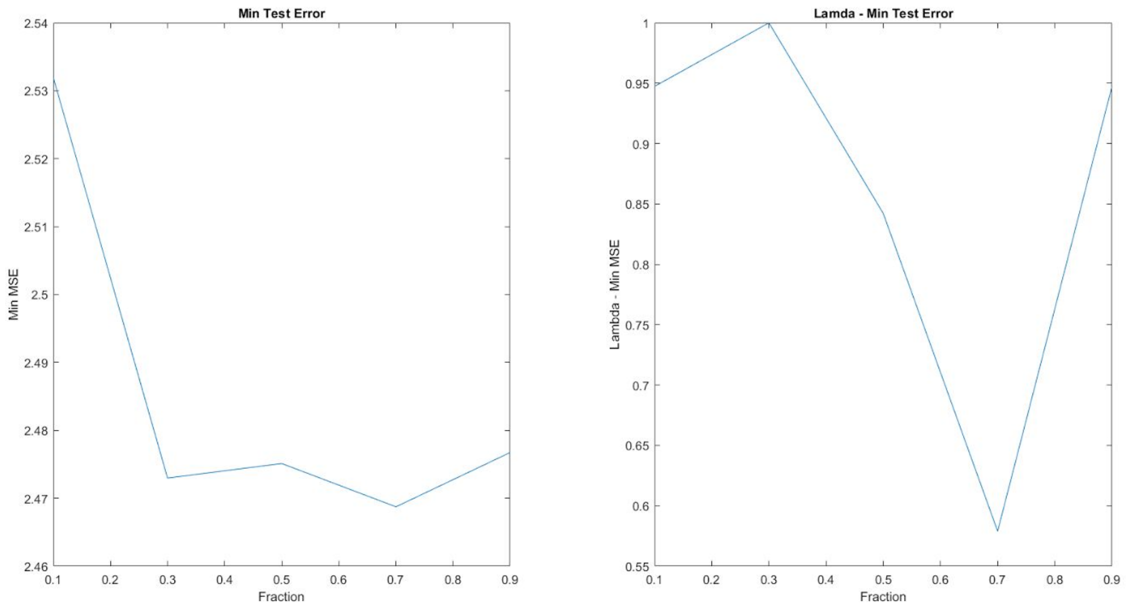
Q9)



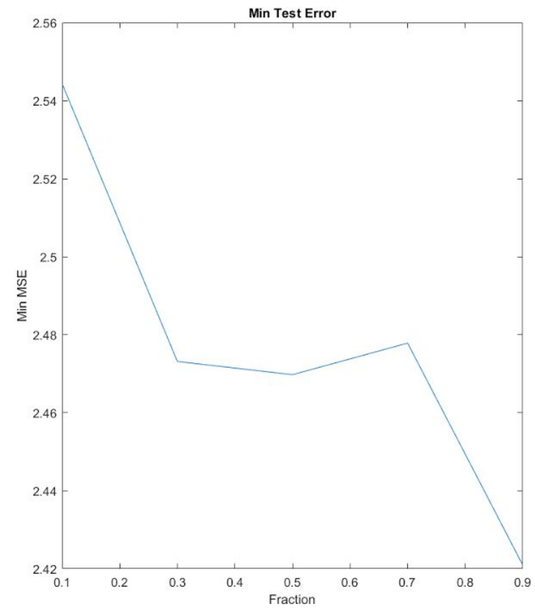
<u>Result on first Run</u>

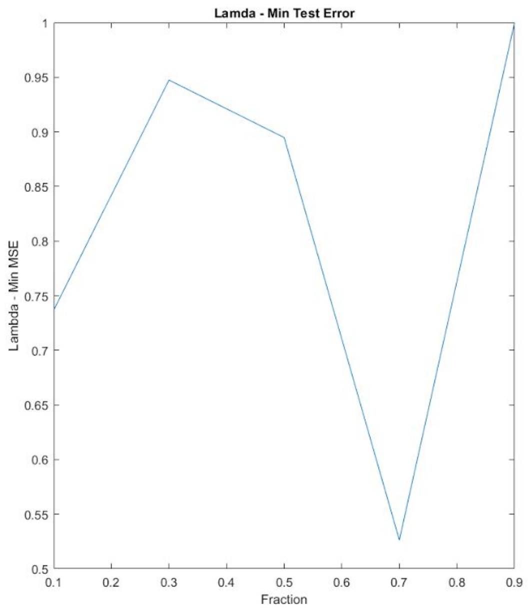


<u>Result on second Run</u>



<u>Result on third Run</u>





<u>Result on fourth Run</u>

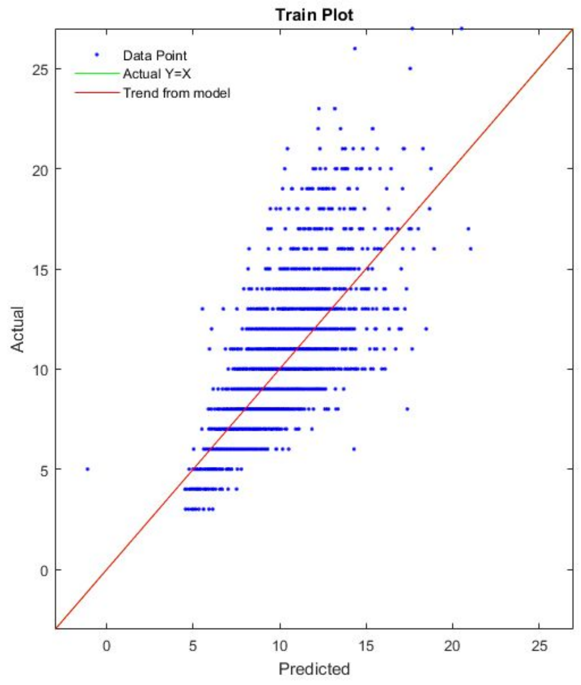
- <u>Observation</u>: Minimum Test Error decreases as fraction of train data is increased

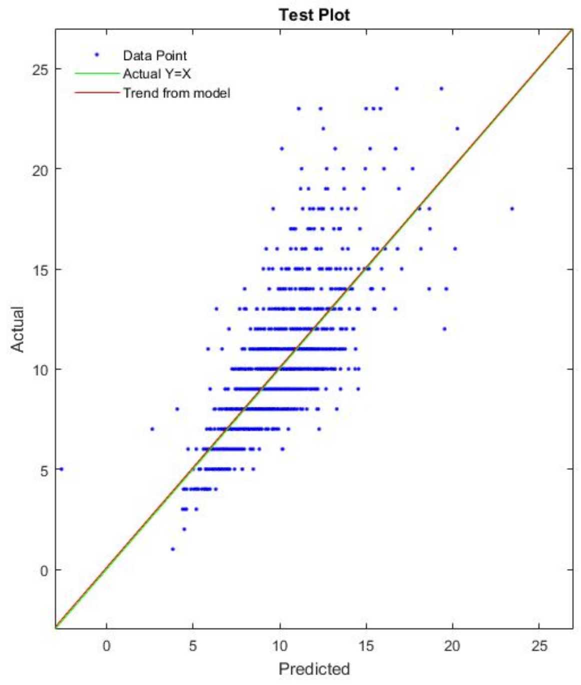
  <u>Reason</u>: As we learn a model on larger train set we learn a better model which follows general trend and is less overfitted to the train set. Since the training set is getting larger it represents the overall instance class in a better way.

- <u>Observation</u>: There is no correlation between λ and fraction i.e. for a fraction the λ to minimum test error changes for every run between $\lambda \sim (0.5,1)$.

  <u>Reason</u>: The value of λ and fractions are independent from each as evident from the graph. This is because the inflection point of test data is distributed over $\lambda \sim (0.5,1)$ for all fractions.

Q10)





<u>Prediction vs Actual</u>

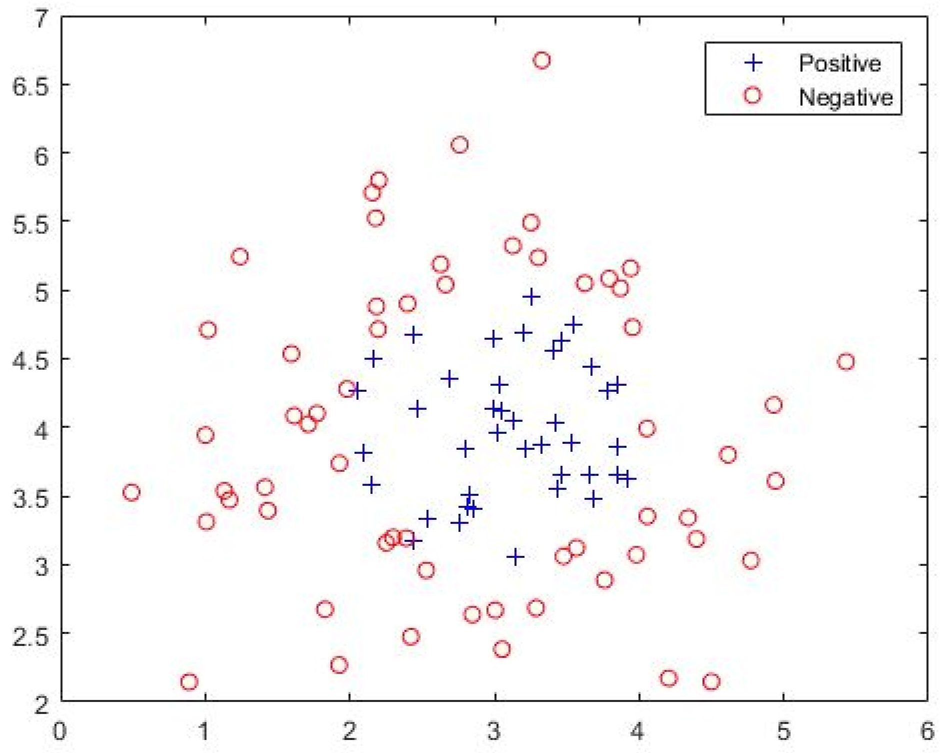
The trend line of model is generated using Polyfit function in matlab which tries to fit a polynomial which in this case is a straight line one over the X and Y values.

Observation:

- For train set, the actual Y=X closely resembles the trend line from learned model showing that weights learn help in classification on training data accurately.
- For test set, the trend line differs from Y=X line very slightly showing the model closely approximates the actual model.

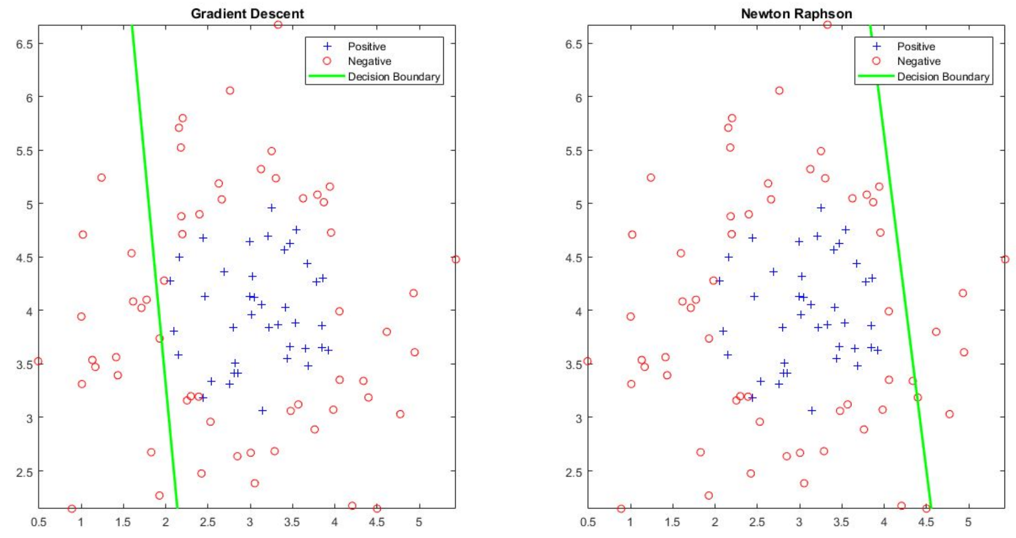
Part2
Logistic Regression

Q1)



The plot for the dataset

Q2.



<u>The decision boundary when we try to fit a straight line as a classifier</u>

Gradient Descent

- Each iteration includes less work so each iteration is fast.
- It has very high no of iterations.
- Its convergence depends on α value.

Newton-Raphson

- Each iteration takes large amount since it involves larger work.
- It converges in very small no of iterations.

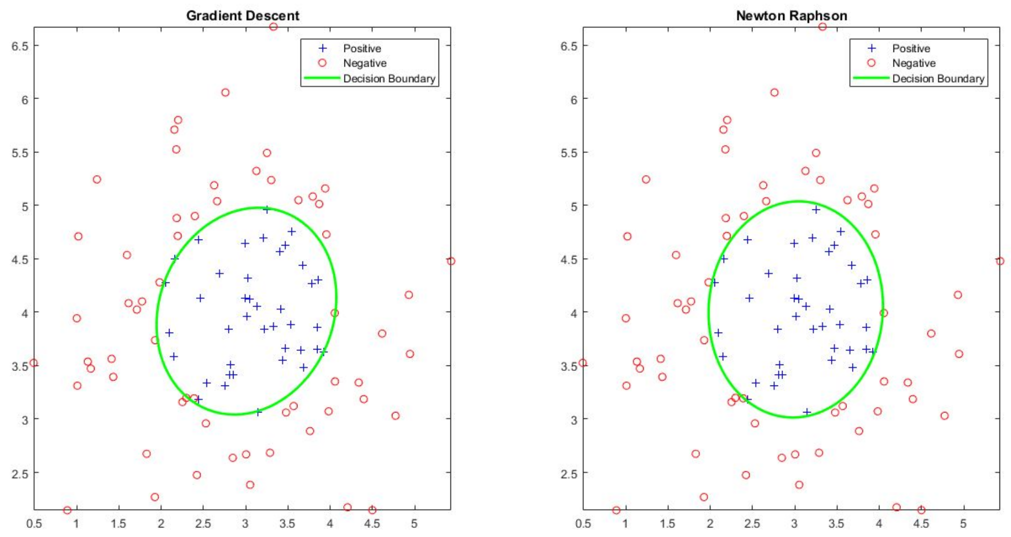
The decision boundary learned by both methods very a lot. One finds a left of data and other on right since data is not linearly separable.

Gradient Descent
No of Iterations :1000000 (keeps oscillating - doesn’t converge)
Accuracy :0.570000

Newton Raphson
No of Iterations :3
Accuracy :0.540000

Q3. The data is NOT linearly separate. Since there exists no line which can classify the positive label on one side and negative on other. Since positive data is surrounded by negative data.

Q4 and Q5 .

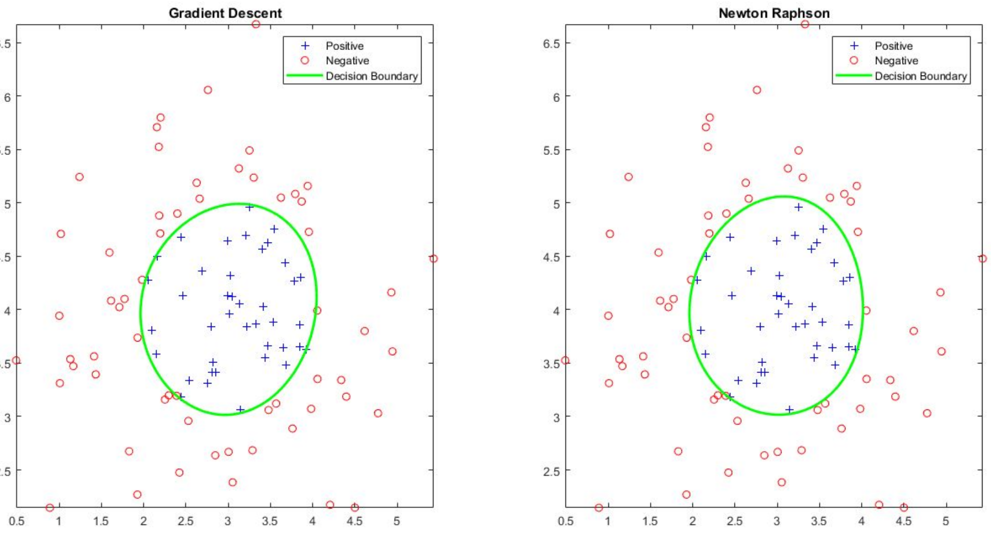


<u>Plot for Degree 2</u>

For degree = 2
Gradient Descent
No of Iterations :857958
Accuracy :0.950000

Newton Raphson
No of Iterations :6
Accuracy :1.000000

The surface that is generated is a circle as the one that we can see in the image.



Plot for degree 3

For degree = 3
Gradient Descent
No of Iterations :637357
Accuracy :1.000000

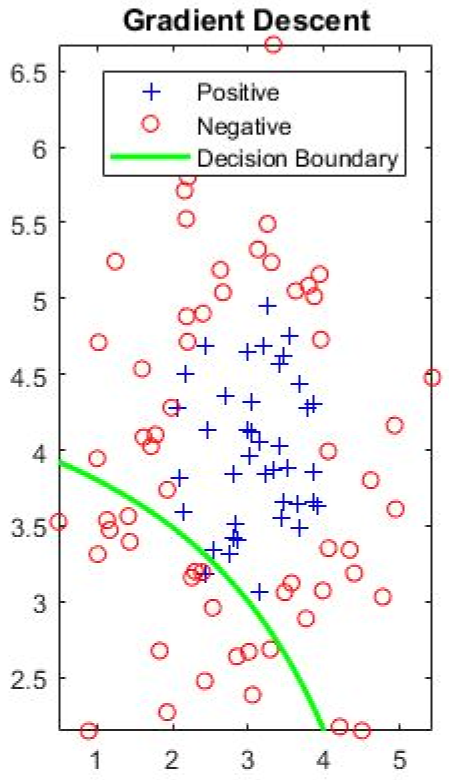
Newton Raphson
No of Iterations :7
Accuracy :1.000000

The surface that is generated in this case is a sphere or an ellipsoid. The diagram is only a contour plot hence we get a circular/elliptical contour.

Q6.

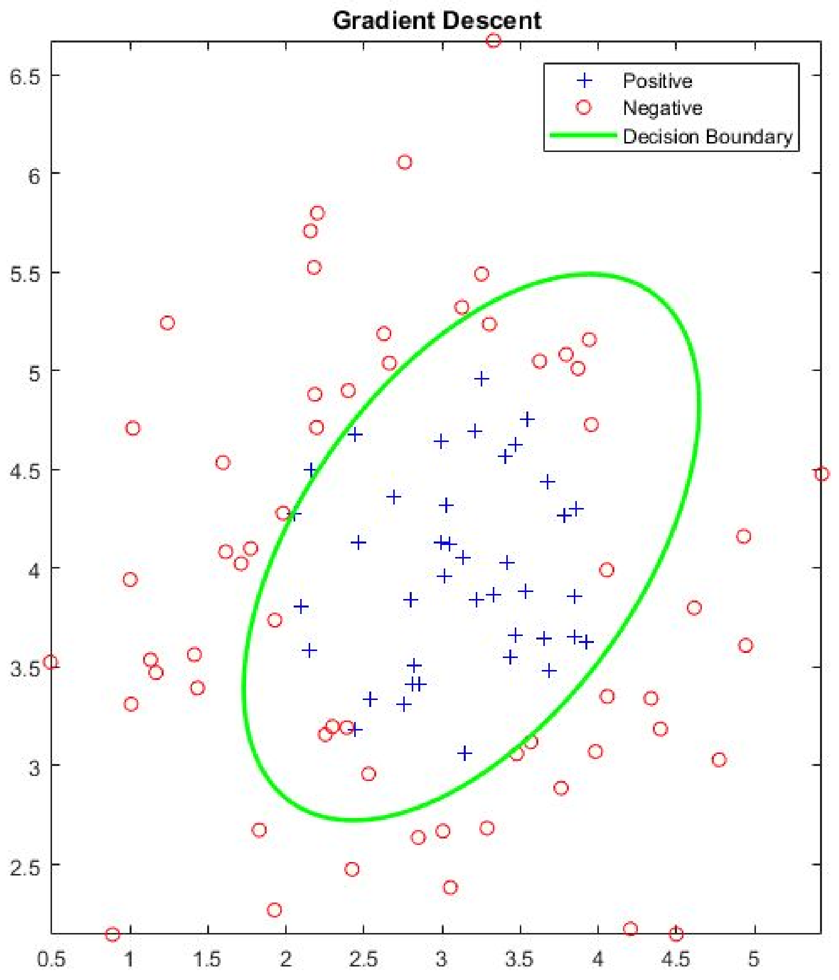
Variation on alpha: When the value of alpha is relatively small the model fits the data well. But as and when we keep increasing the value of alpha beyond some limit the model starts underfitting. The following image shows an example of such a case

Underfitting (λ:100, Degree: 4)



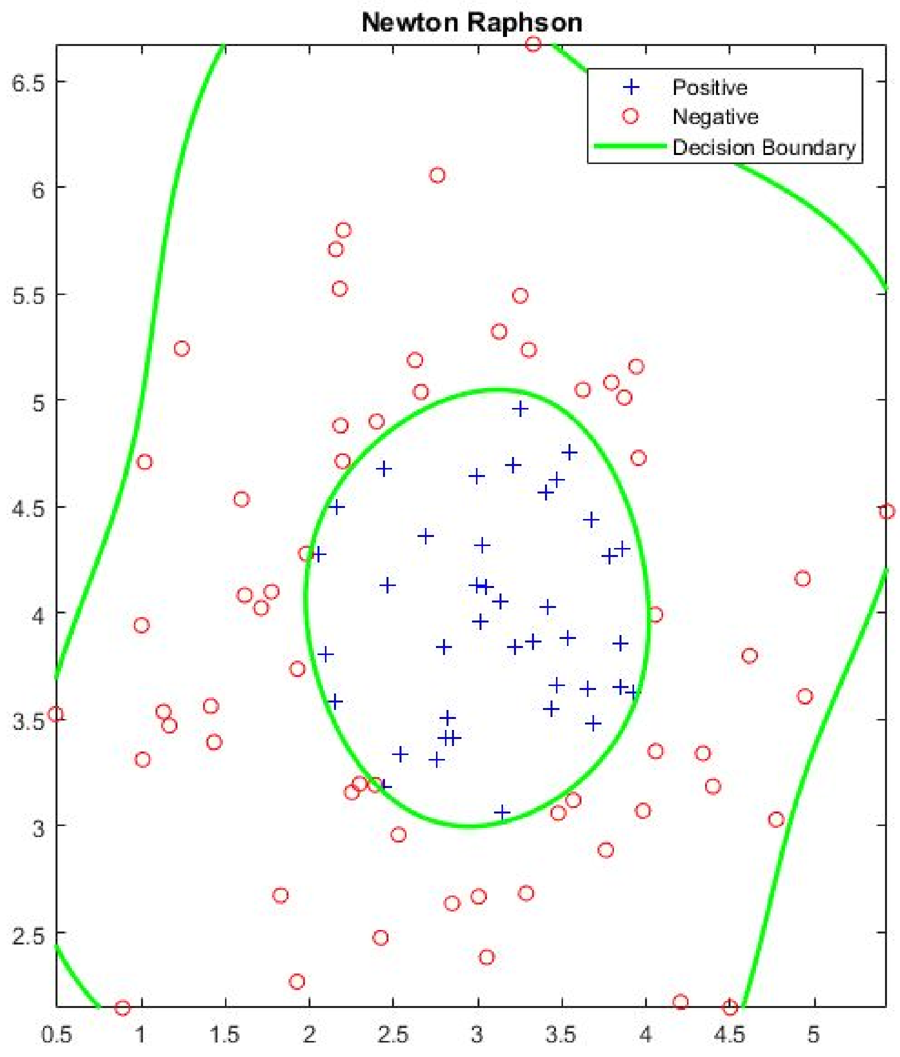
λ:100
Degree:4

Underfitting (λ: 0.002, Degree: 2)



Specification
Method: Gradient Descent
λ: 0.002
Degree: 2
No of Iterations :1000000
Accuracy :0.870000

Overfitting (λ:0, Degree: 4**)**



Specification
Method: Newton Raphson
λ: 0
Degree: 4
No of Iterations :5
Accuracy :1.000000