MACHINE LEARNING

Lab-2 Report

Linear Ridge Regression

Least Square "Linear Regression" is a statistical method to regress the data with dependent variable having continuous values whereas independent variables can have either continuous or categorical values. In other words "Linear Regression" is a method to predict dependent variable (Y) based on values of independent variables (X). It can be used for the cases where we want to predict some continuous quantity. E.g., Predicting traffic in a retail store, predicting a user's dwell time or number of pages visited on Dezyre.com etc.

Q6-We first find the least significant attributes but before that we find the error for training and test dataset before removing those attributes. The fraction size we take here is 0.7.

- lambda value used->
  0.1
- training error before removing the least significant attributes->
  4.6980
- testing error before removing the least significant attributes->
  5.2446
- training error after removing the least significant attributes->
  5.4994
- testing error after removing the least significant attributes->
  5.8377
- Weights before removing the least significant attributes

1. 9.8690
2. 0.1437
3. -0.2552
4. 0.1086
5. 0.0658
6. 0.8883
7. 0.8051
8. 4.1215
9. -4.1937
10. -1.1588
11. 1.0195

- Weights after removing the least significant attributes

1. 9.8690
2. 0.1437
3. -0.2552
4. 0

5. 0
6. 0.8883
7. 0
8. 4.1215
9. -4.1937
10. -1.1588
11. 1.0195

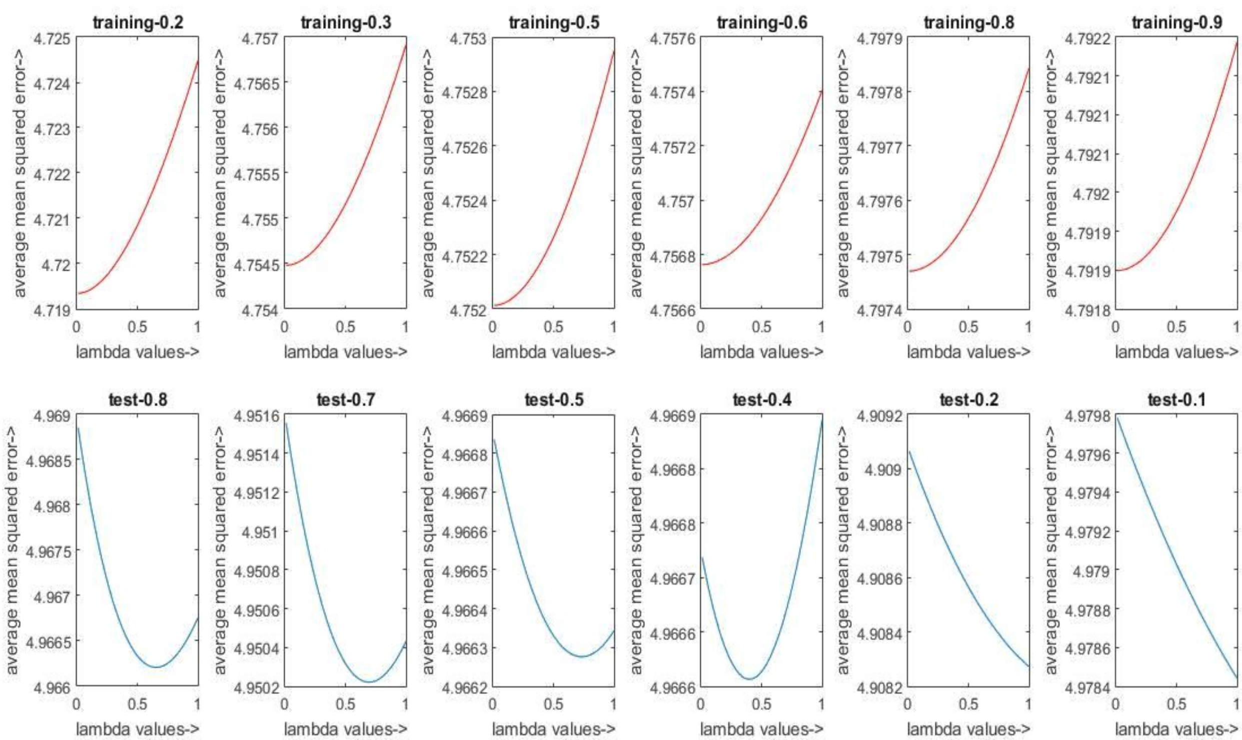
.

Q7-Q1-Does the effect of lambda on error change for different partitions of the data into training and
testing sets?

Yes, effect of lambda on error changes for different partitions of data into training and testing sets.

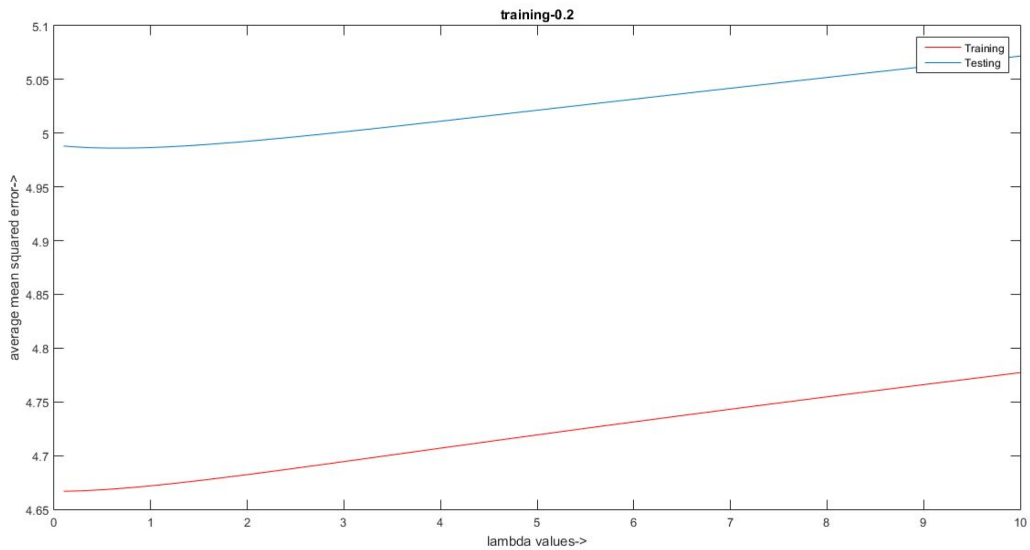
Suppose we take lambda from 0 to 1 taking 100 different values in between.The change is depicted by the following figure.

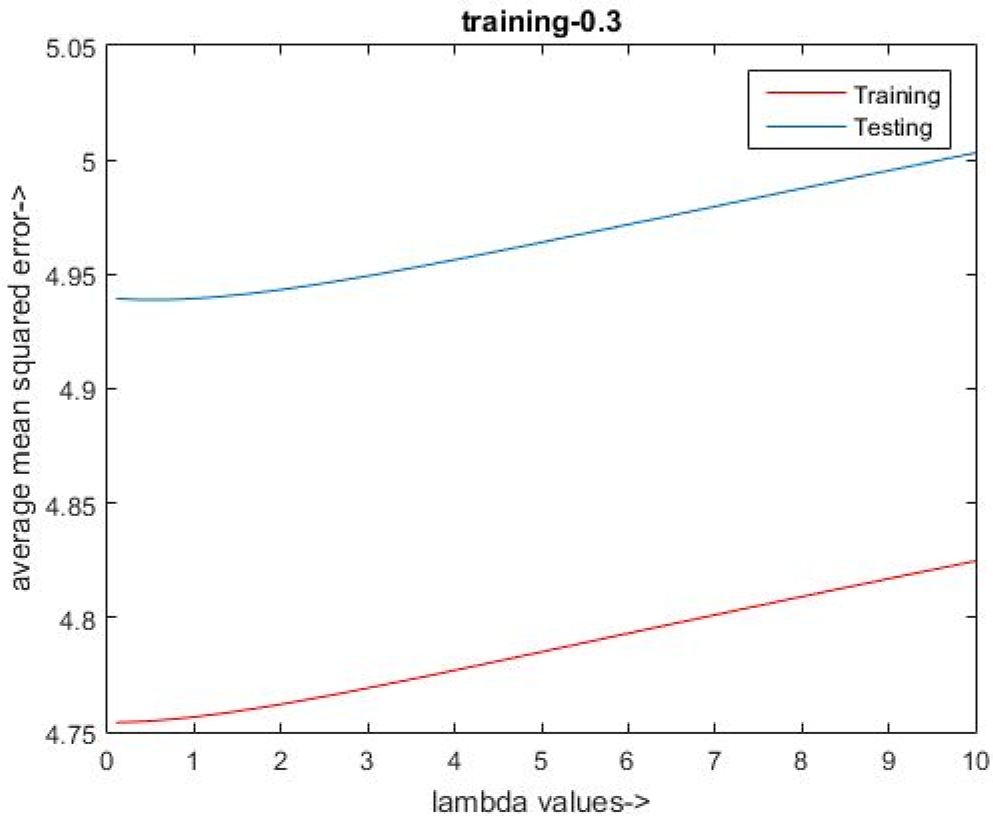
Note:-In the figure training-0.2 implies that we have taken 20% of data for training set.
Similarly for test-0.8 implies that we have taken 80% of data for testing.

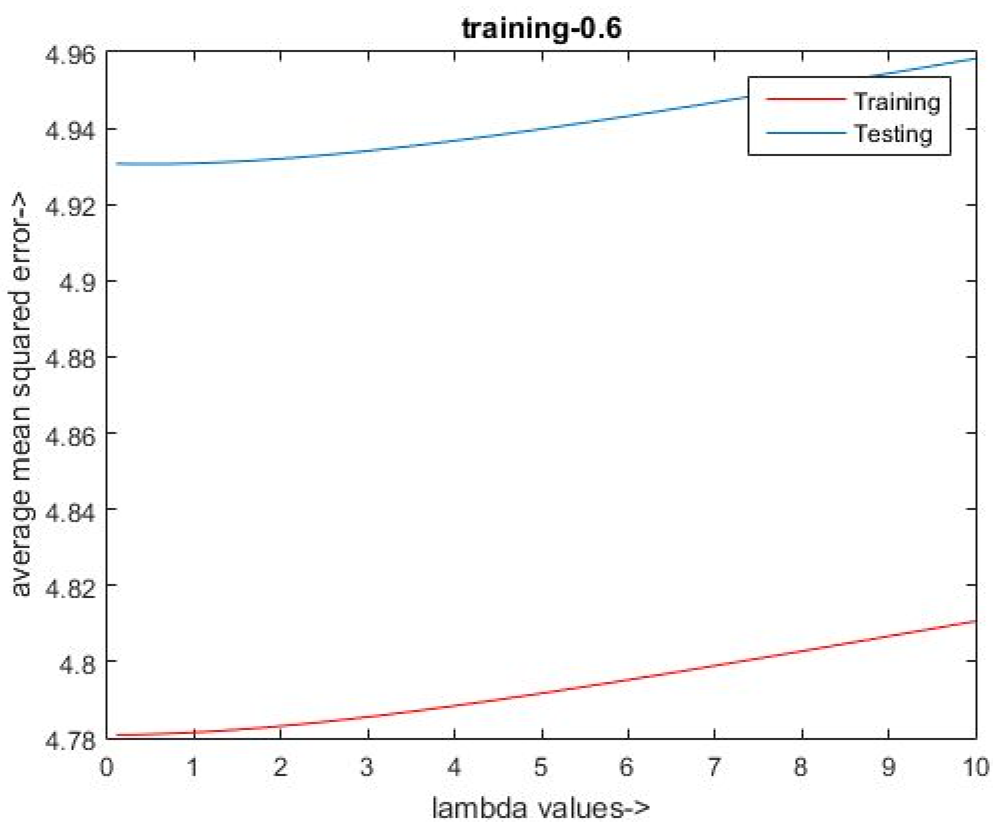


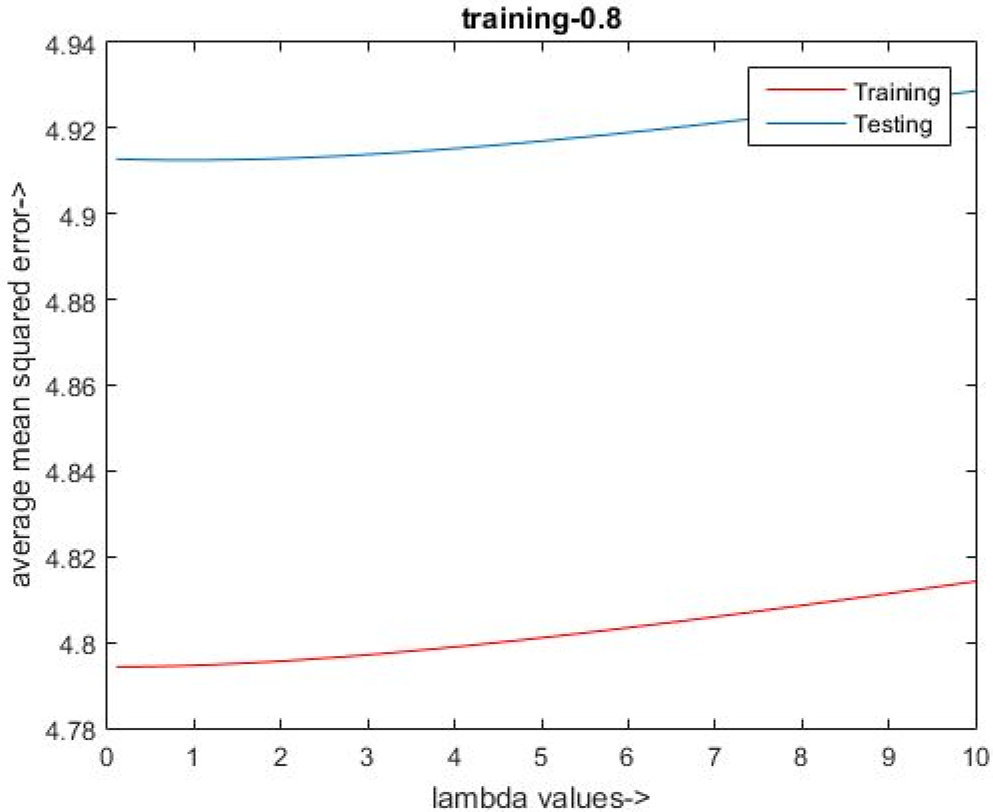
Observation-:We observe that when we vary lambda from 0-1 error corresponding to different lambda values increases in training set but in test set we observe that error first decreases with slight increase at some points.

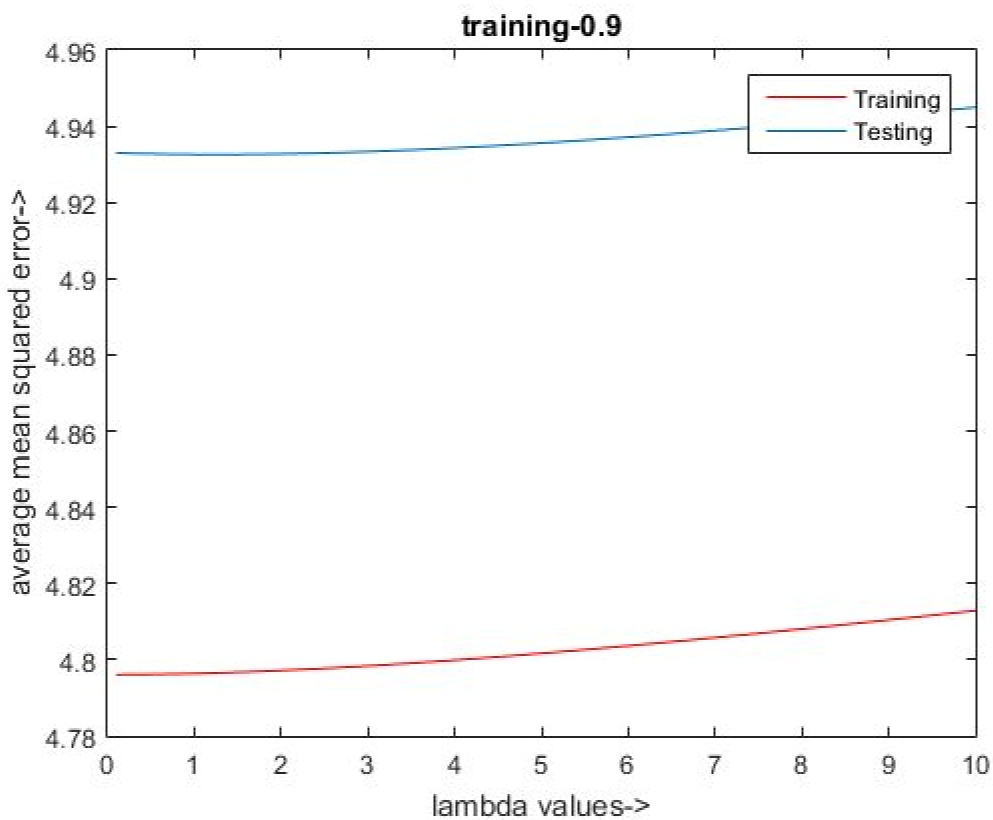
Now we take lambda to vary from 0 to 10 having 100 values in between. The corresponding figures we obtained are shown below




training-0.3
Training
Testing
average mean squared error->
lambda values->
5.05
5
4.95
4.9
4.85
4.8
4.75
0
1
2
3
4
5
6
7
8
9
10

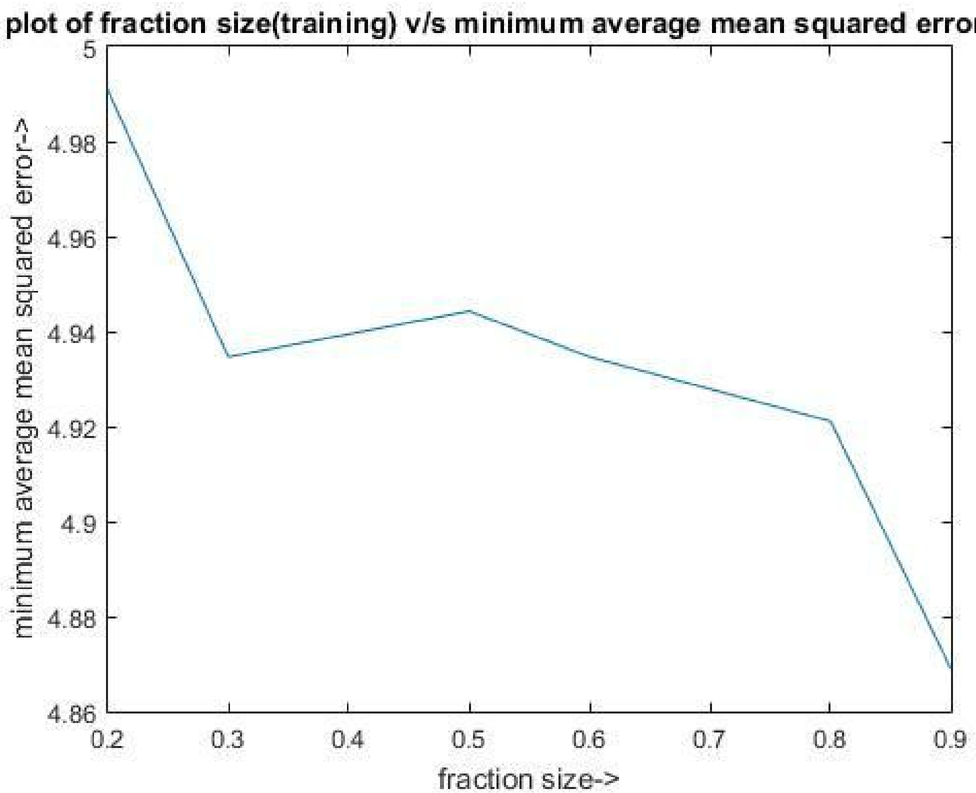

training-0.6
Training
Testing
average mean squared error->
lambda values->
4.96
4.94
4.92
4.9
4.88
4.86
4.84
4.82
4.8
4.78
0
1
2
3
4
5
6
7
8
9
10


training-0.8
Training
Testing
4.94
4.92
4.9
4.88
4.86
4.84
4.82
4.8
4.78
0
1
2
3
4
5
6
7
8
9
10
average mean squared error->
lambda values->

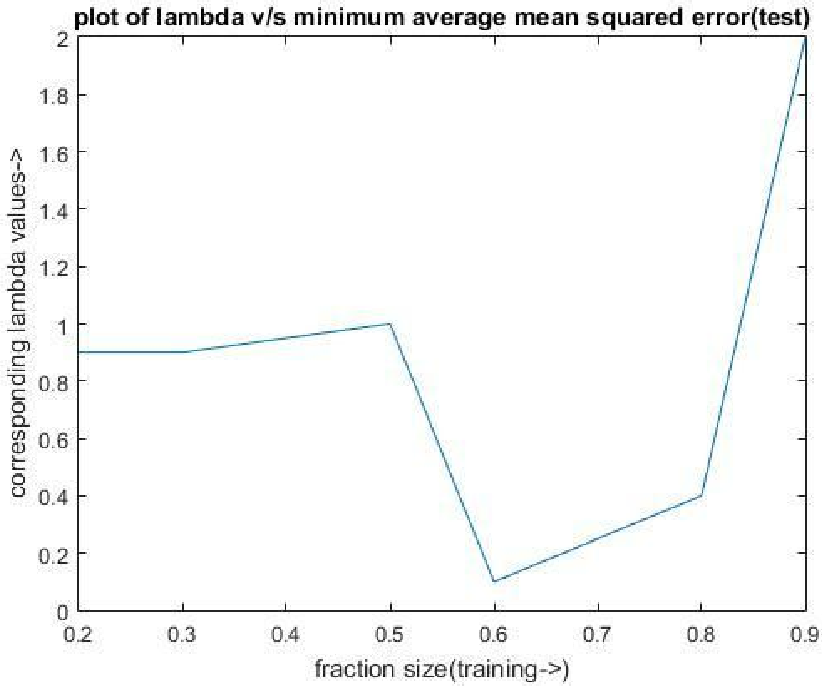

training-0.9
Training
Testing
4.96
4.94
4.92
4.9
4.88
4.86
4.84
4.82
4.8
4.78
0
1
2
3
4
5
6
7
8
9
10
average mean squared error->
lambda values->

Observation-: When we vary lambda from 1 to 100 with 100 values in between we observe that error increases with lambda for both training and testing set. Though this increment is not very much as we can see that only a difference of maximum 0.2 is there between maximum and minimum average mean squared error value for any case of training or test data

For better understanding we have also plot two curves for lambda values of above-:
Curve-1: Minimum average mean squared testing error versus training fraction values



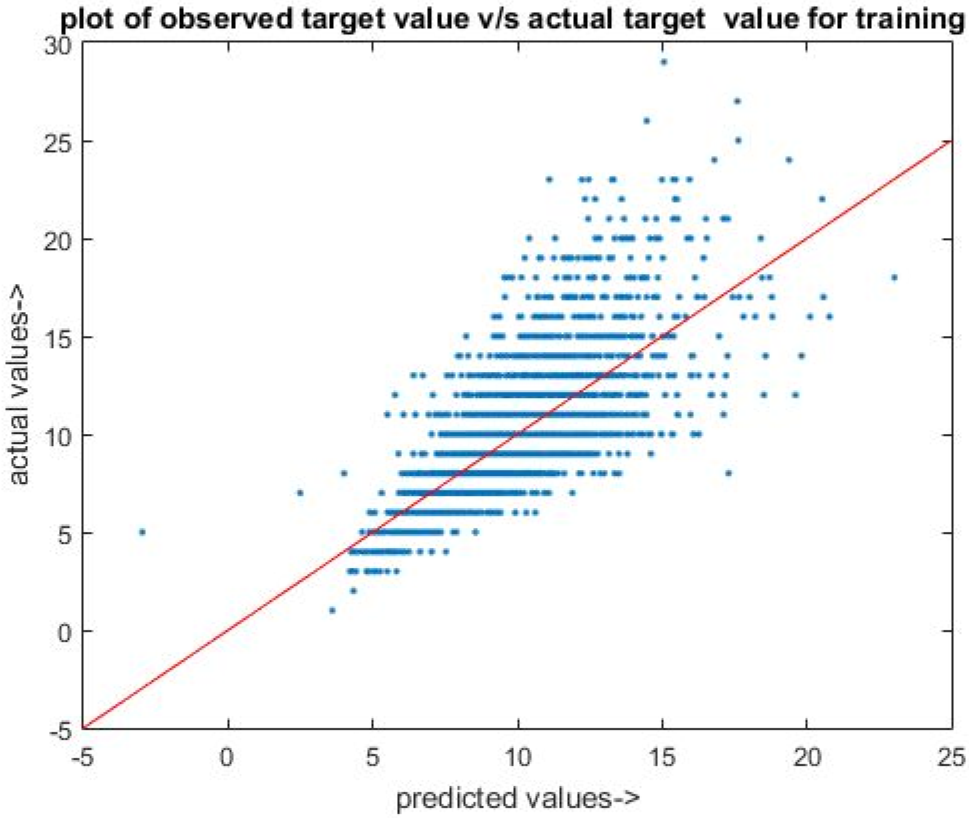
Curve2-: lambda value that produced the minimum average mean squared testing error versus the training set fraction.



Observation-:We observe that minimum average mean squared error decreases as fraction size increases and this has to be true as our model is trained properly as the fraction size for training set increases.

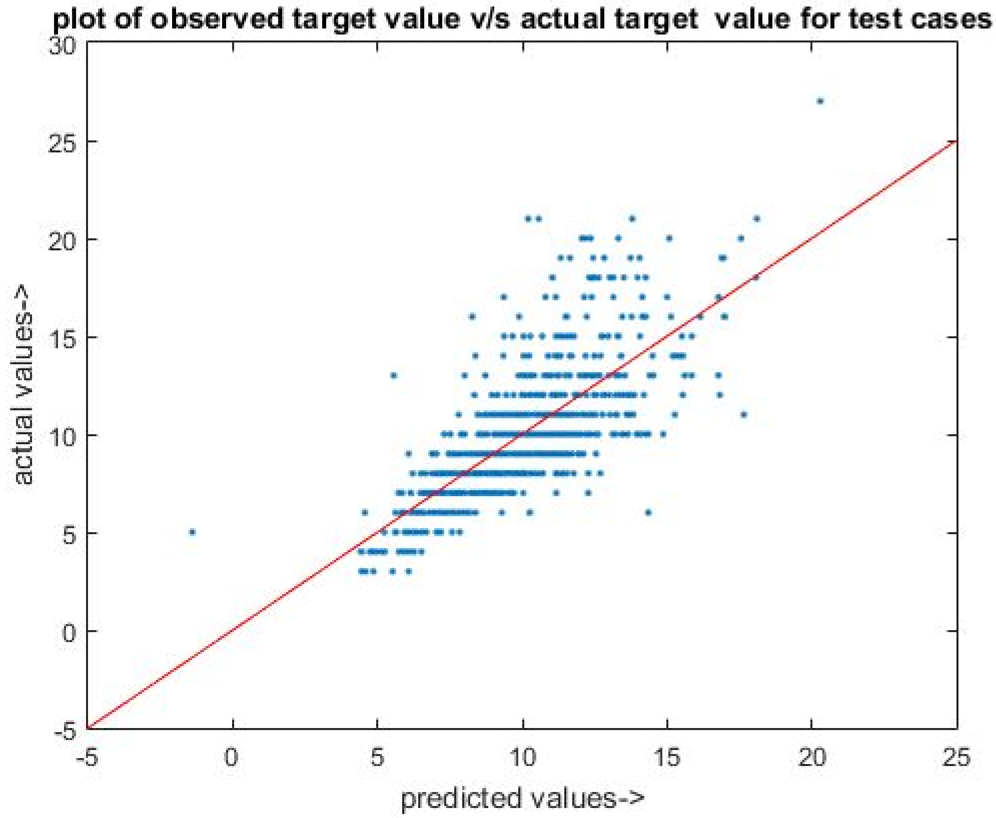
Q7-Q2**-How do we know if we have learned a good model?**

To answer this question we plot the the curve of actual values versus the predicted values.If all the points come close to a 45 degree line we can say that we have learned a good model.The plots for training and test data set are shown below.



Note:In above figure the blue dots represent the points and the red line is a reference line of slope 1 and intercept 0;The above curve is for training set examples.

Observation:-Our model fits correctly to the training set as all the points are very close to the reference line.

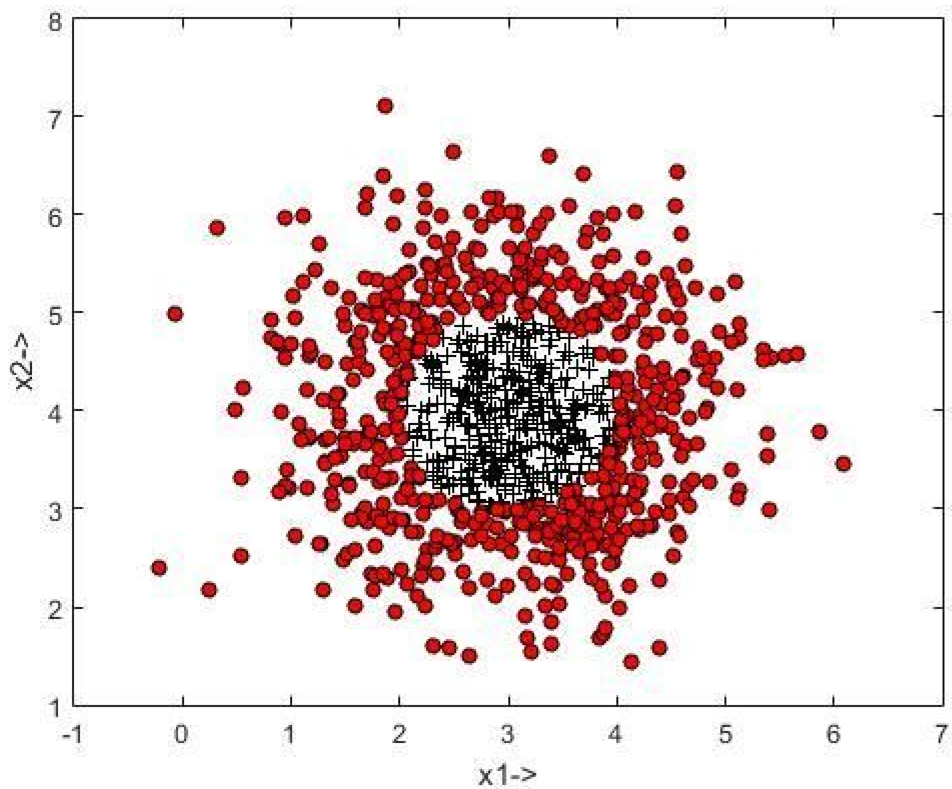


Note:In above figure the blue dots represent the points and the red line is a reference line of slope 1 and intercept 0;The above curve is for training set examples.

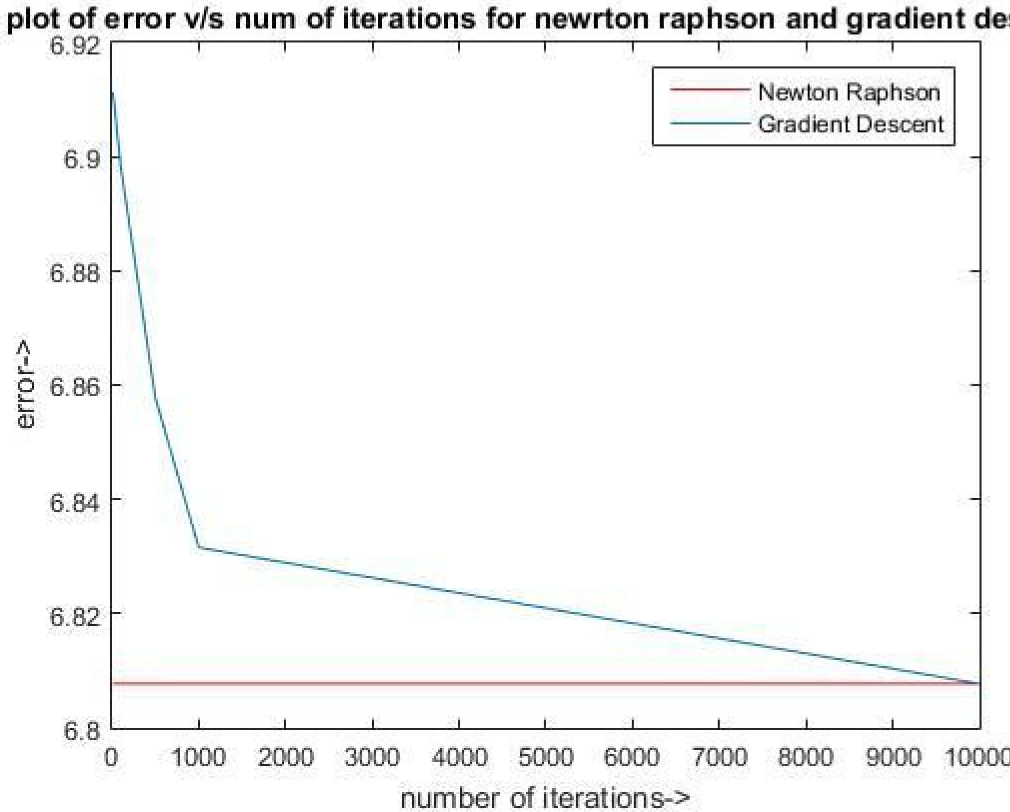
Observation:-Our model fits correctly to the testing set as all the points are very close to the reference line.

Regularized Logistic Regression

In first part of this lab we plot the dataset for two classes. The curve is shown below. In the curve black '+' represent the classes with positive labels and red 'o' represent classes with negative labels.



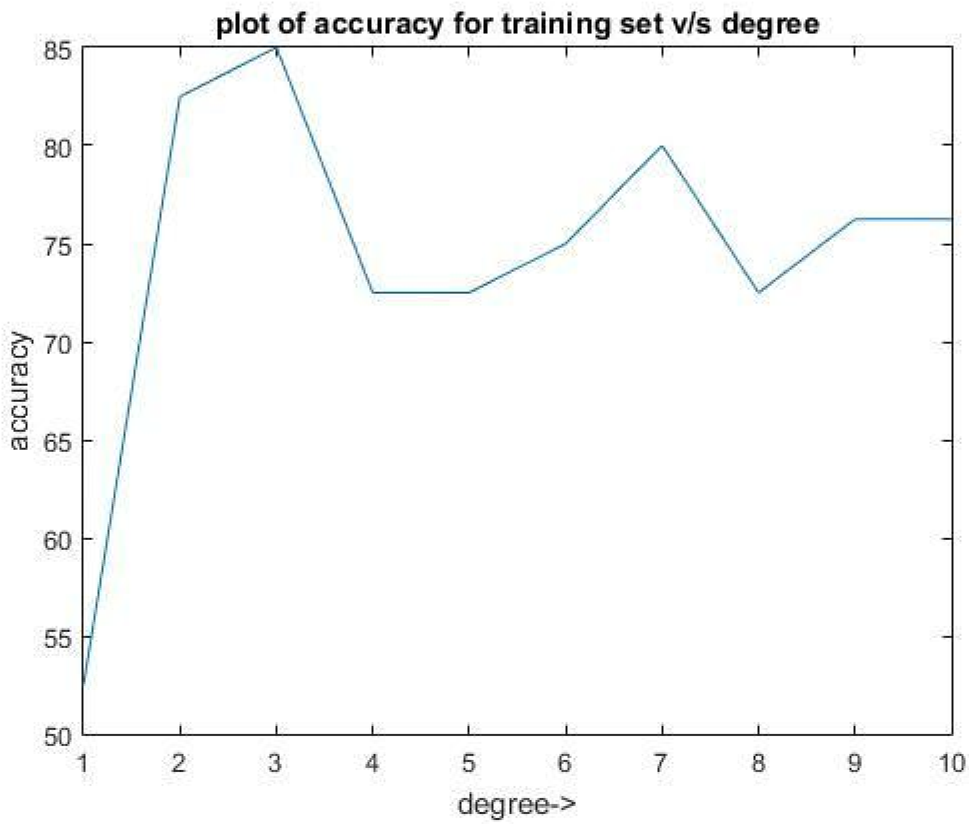
Q2-Now for optimization we implement two optimization techniques namely gradient Descent and Newton Raphson.To know which performs better we plot a curve of error v/s number of iterations The plot we obtain is shown below



Observation-:As the graph suggests that if we take newton raphson method than it gives us less error as compared to gradient descent.Also the graph tells us that after a fixed number of iteration both the method converges to one point i.e. both give us same error after 10000 iterations.Thus for less number of iteration we can say that newton raphson give us more optimized model than gradient descent.This has to be true as newton raphson converges much faster than gradient descent.

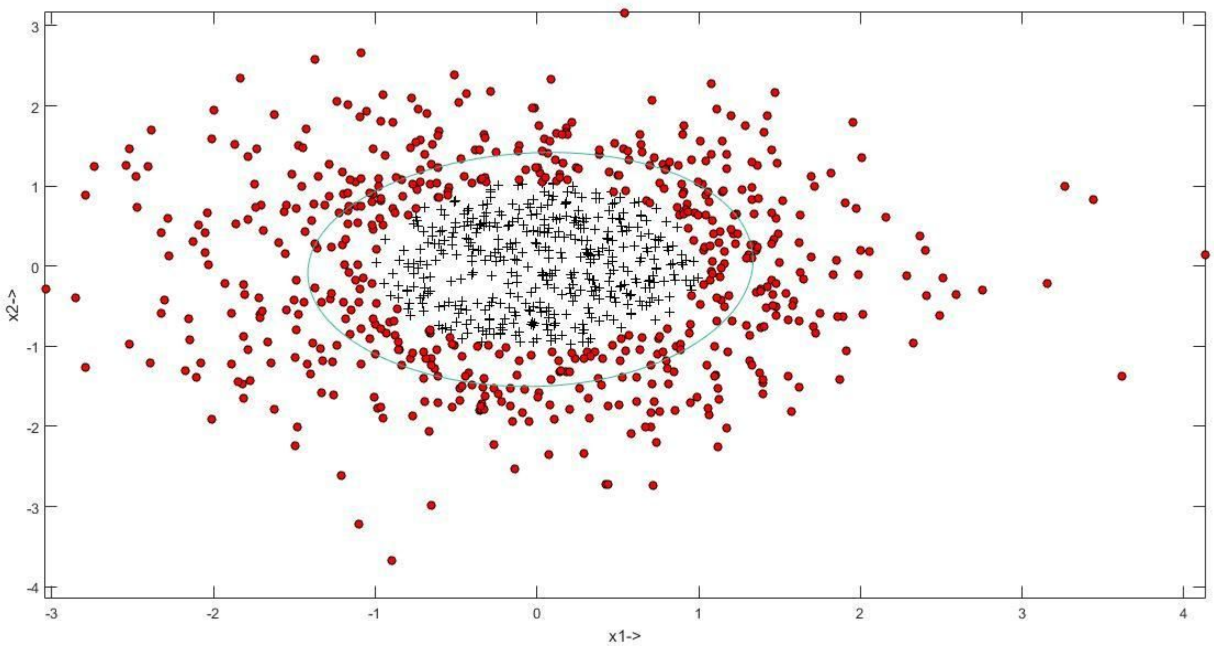
Q3-No as we can see from the curve we plotted for two different classes we can comment that the data has a non- linear decision boundary.

Q4-We implemented the function featuretransform(X,deg) that takes the corresponding input and form a polynomial of maximum degree ‘deg’.To know what is the best degree to pick we plot a graph of accuracy for the training set v/s degree.the graph is shown below.



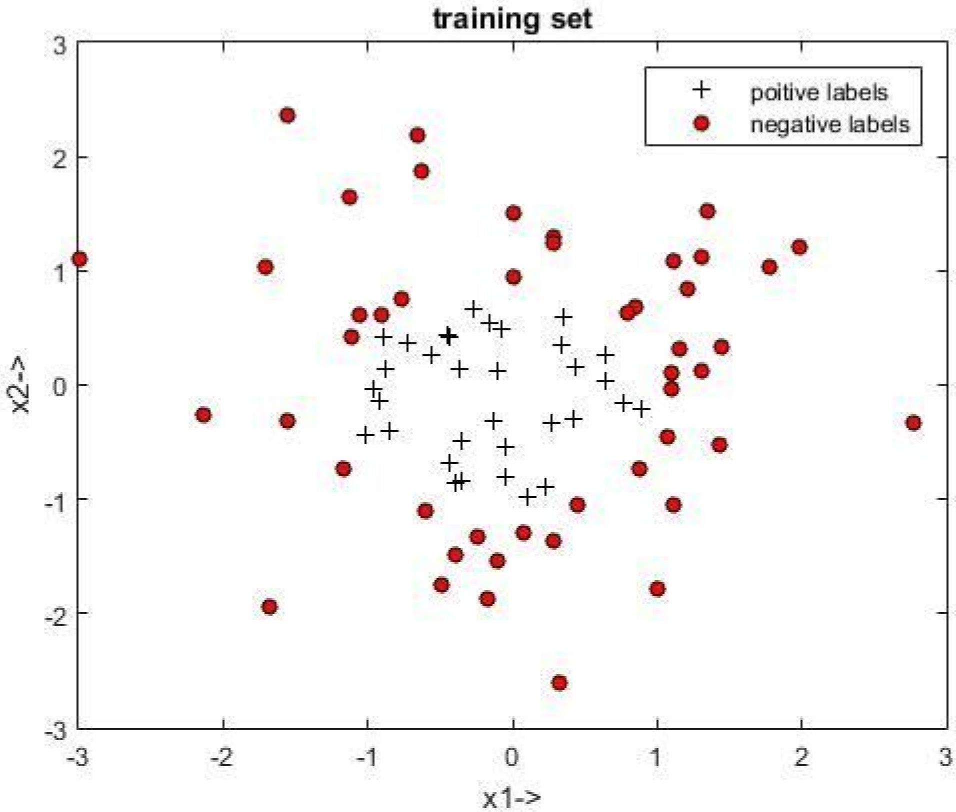
As the plot suggests degree of 2 or 3 is the best to pick for our further analysis.Therefore we pick the degree as 3.

Q5-Now we will plot the non linear decision boundary for the classifier learned in previous step using degree 3.The curve is shown below

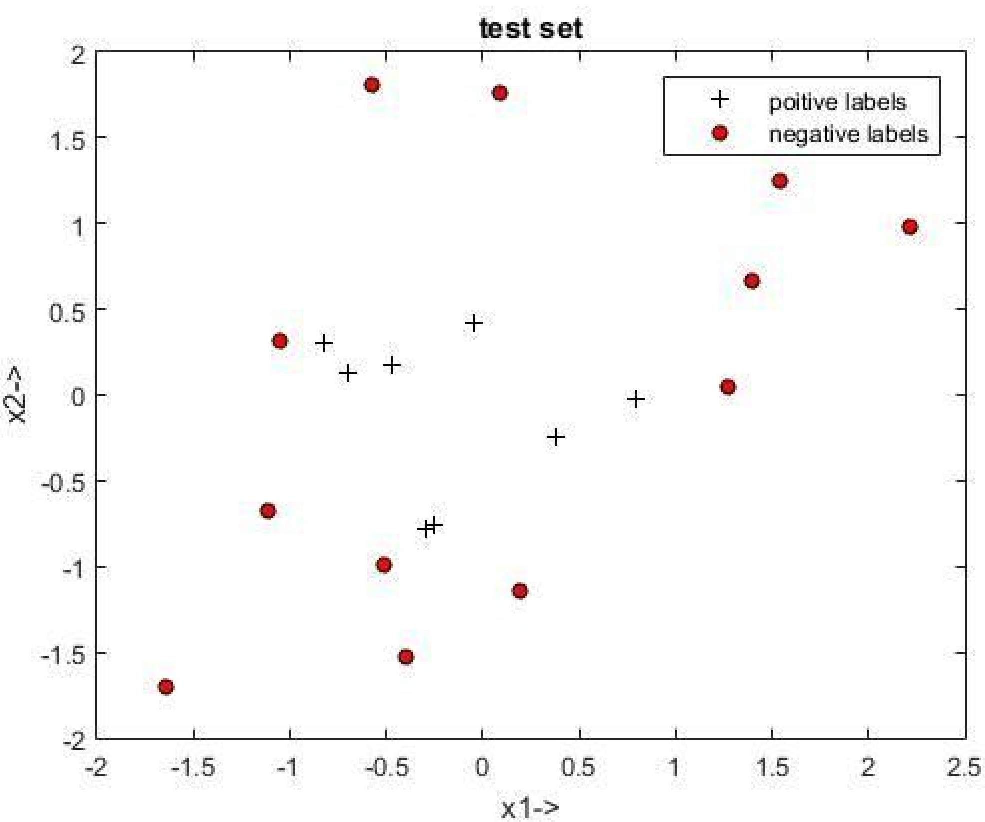


Note:In above figure black ‘+’ represent the classes for positive label and red ‘o’ represent the classes with negative label.The blue contour is our decision boundary.

Q6-To find the value of lambda for corresponding overfitting and underfitting we divide our data set in training and test set and check accuracy for the two.We divide our dataset into 80% training and 20% test.The corresponding plots are shown below.

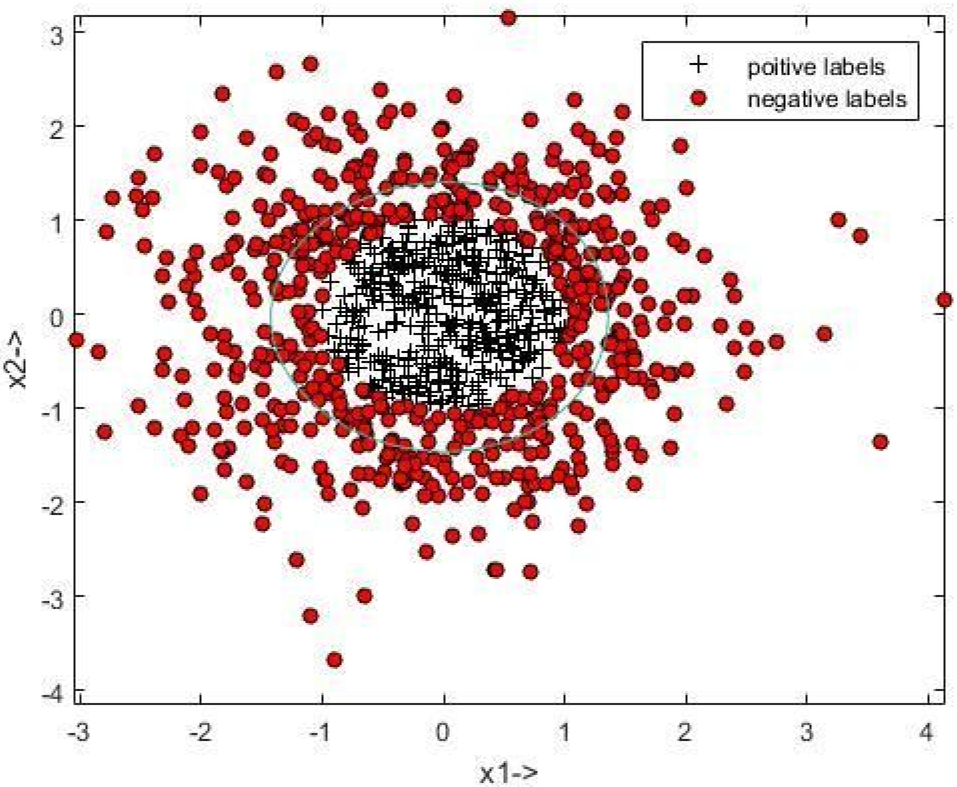


training set
poitive labels
negative labels
x2->
x1->
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3



test set
poitive labels
negative labels
x2->
x1->
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

Case of Overfitting->
Value of lambda=0.001
train_accuracy->100%
Test accuracy->70%



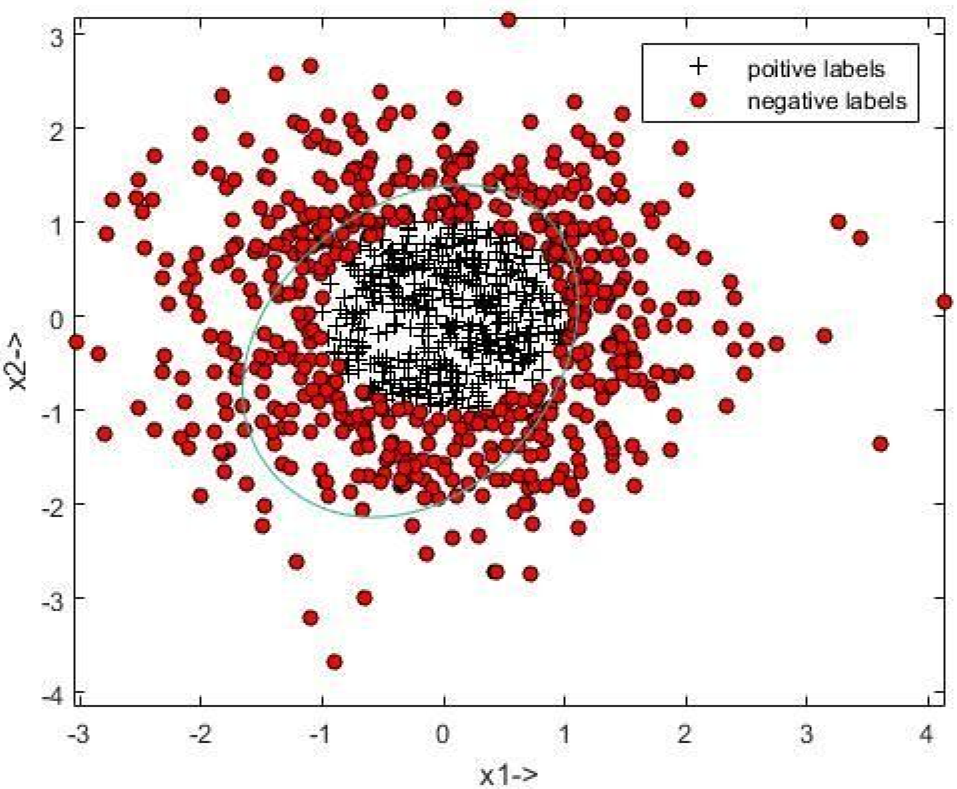
Observation-We observe that the decision boundary almost separates the two classes and also accuracy for training set comes out to be 100% this simply implies that it is a case of overfit model

Case of underfitting->

Value of lambda=120

train_accuracy=42.5

test_accuracy=30



Observation-We observe that the decision boundary is not able to distinguish between the two classes and also accuracy for training set comes out to be 42.5% this simply implies that it is a case of underfit model.