ASSIGNMENT 4 - MACHINE LEARNING

Kartic Choubey - IIT2018181

1(a).

DATA RETRIEVING AND DIVIDING INTO TRAINING AND TESTING

In this , I divide my dataset into 70% training and 30% testing randomly by using train_test_split . As the dataset contains 100 rows, so 70 rows were there in the Training Set and 30 rows were there in the Testing set. Then in my trainingX and testingX , I use slicing to take the first two columns as my ‘X’ and the last column will be my ‘Y’ for trainingY and testingY. Then I concatenate the column with 1 as the value in column (as X0 = 1) .

WITHOUT FEATURE SCALING

I use batch gradient descent , stochastic gradient descent , and mini batch gradient descent algorithm with epochs = 2000 , learning rate = 0.00001 and batch size = 10. In all the cases , my accuracy is 60 % . That means out of 30 predictions , 18 were correct.

WITH FEATURE SCALING

I use min max scaling for feature scaling . Min max scaling will bring my values in the range [0,1] according to the formula

$$Xscaled = (X - min(X)) / (max(X) - min(X))$$

I use batch gradient descent , stochastic gradient descent , and mini batch gradient descent algorithm with epochs = 2000 , learning rate = 0.00001 and batch size = 10. In all the cases , my accuracy is 60 % . That means out of 30 predictions , 18 were correct.

1(b).

ADDING FEATURES

Initially we have two features in the csv. Let's name it X1 & X2 . Now we are adding 6 new features . Those features will be:

$X1^2$, $X2^2$, X1*X2 , $X1*X2^2$, $X1^2 * X2$, $X1^2 * X2^2$

So now we have total 8 features .

I use batch gradient descent , stochastic gradient descent , and mini batch gradient descent algorithm with feature scaled values with epochs = 2000 , learning rate = 0.00001 and batch size = 10. **(Repeating 1(a))**
In all the cases , my accuracy is 60 % . That means out of 30 predictions , 18 were correct.

1(c).

WITH REGULARIZATION

I chose my hyper parameter(lambda) as 100000. In all three algorithms, I just changed one line to make it regularized .

In normal algorithm : **theta = theta - learningRate * gradFactor**
In regularized algorithm :
theta = (1 - (lambda * learningRate)/Number of Samples) - learningRate* gradFactor

I use batch gradient descent , stochastic gradient descent , and mini batch gradient descent algorithm with feature scaled values with epochs = 2000 , learning rate = 0.000000001 and batch size = 10.
In all the cases , my accuracy is 60 % . That means out of 30 predictions , 18 were correct.

Also I **repeated 1(b)** and then use regularized algorithms to find Theta and then accuracy.
In all the cases , my accuracy is 60 % . That means out of 30 predictions , 18 were correct.

2.

DATA RETRIEVING AND DIVIDING INTO TRAINING AND TESTING

In this , I divide my dataset into 70% training and 30% testing randomly by using train_test_split . As the dataset contains 302 rows, so 291 rows were there in the Training Set and 91 rows were there in the Testing set. Then in my trainingX and testingX , I drop the last column and then the remaining columns will be my ‘X’ and the last column will be my ‘Y’ for trainingY and testingY. Then I concatenate the column with 1 as the value in column (as X0 = 1) in trainingX and testingX.

PREDICTING CLASSIFIER

I use batch gradient descent algorithm with epochs = 2000 and learning rate = 0.0001 to predict my classifier and my classifier gives 75.82% accuracy . That means out of 91 predictions, 69 predictions were correct.

PERFORMANCE ANALYSIS

I use a confusion matrix to do the performance analysis . My performance matrix is:

| 44 , 14 |
| 08 , 25 |

By analysing confusion matrix, we can conclude that:

1. There are two predicting classes as our matrix is 2*2. Those Classes are "Yes" & "No"
2. The Classifier made a total of 44 + 14 + 8 + 25 = 91 predictions.
3. Out of those 91 cases,the classifier predicted "Yes" 14 + 25 = 39 times and "No" 44 + 8 = 52 times.
4. In reality, 33 patients in the sample have a heart disease and 58 do not.
5. True Negative(TN) = 44
   False Positive(FP) = 14
   False Negative(FN) = 8
   True Positive(TP) = 25
6. Accuracy = (TP + TN)/Total = (44 + 25)/91 = 0.758