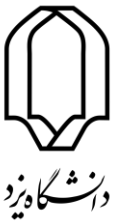
In the name of Allah



Computer Engineering Faculty of Yazd University
Artificial Intelligence

Machine Learning Course

Homework 1 - Part 2

(Logistic Regression)

Instructor: Dr. Yazdian

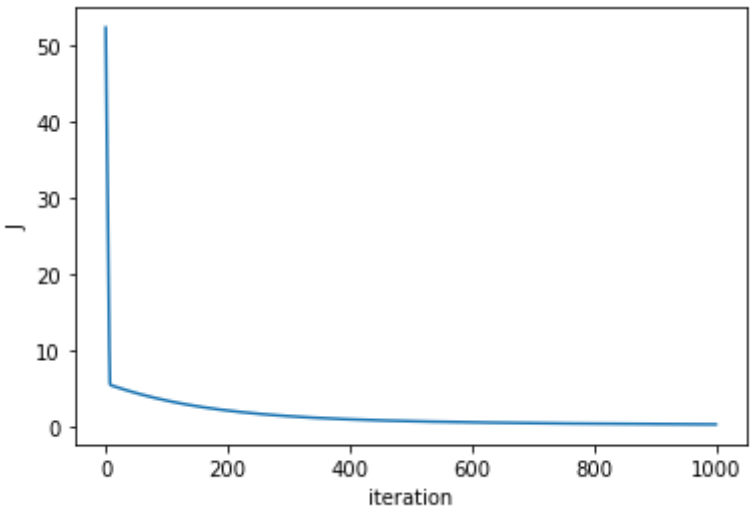
Shiva Zeymaran

Student ID: 40109434

Fall 2022

C. Logistic Regression

1. First we read the text file and try to convert it into a data-frame of pandas to work with the data easier. The first seven columns are features in our data and the last one is the label. Then we should shuffle the data because the labels are in order.
   When we consider only two labels (regions 1 and 3), we will have a Binary Classifier. In total, we have 1000 samples with regions 1 or 3. To make it easier, we consider region 1 as 0 and region 3 as 1 for labels.
   Splitting data into test and train sets is done by using the train_test_split method of sklearn library. Which we will have 800 samples for training set and 200 for test set according to the question.
   Then we model the Binary Classifier using Gradient Descent for Logistic Regression. Considering the learning rate of 0.001 and after 1000 iterations, the final cost will be 0.25. The plot of cost vs. iterations is as follows:



   Then we test the model on our test-data using the calculated weights. Note that it's important to change the value calculated from the softmax function for predicting the right answers in the last step:

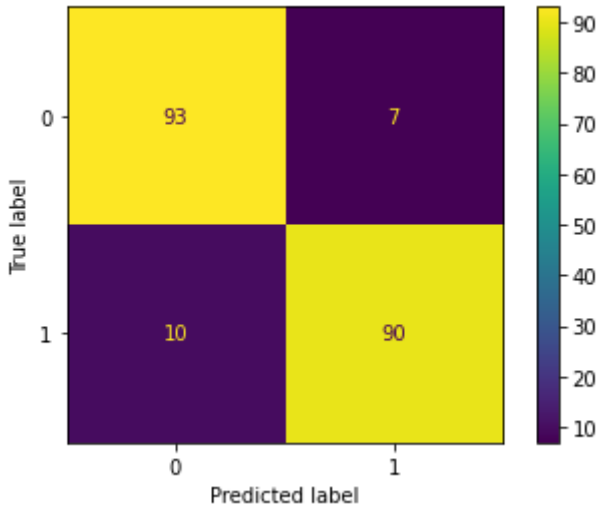
```
y_pred = 1 / (1 + np.exp(-1*(X_test@W)))

preds = [1 if p>=0.5 else 0 for p in y_pred]
```

To calculate the Precision and Confusion Matrix, we used the corresponding functions in sklearn library of python. The final result is:

Precision = 0.93

Confusion Matrix:



2. Now we add L2-norm Regularization to the previous model. There are the results for Precision by using different values of regularize factor:

   Note: To have better comparison with the results of last part, here, we use the same values of learning rate and iteration numbers. Also, we use the same initial values for thetas; this will help to only consider the effect of regularize factor's value.

Regularize Factor	*Precision*
0	0.89
1	0.91
2	0.92
5	0.94

It seems that by increasing the Regularize factor, the Precision will also increase.

3. In this part, we will have a Multi-class Classifier by considering all 4 labels we have in the original data. Therefore, we load the data again and split it to test and train sets.
   Before defining our model, we should change the shape of our labels. For example, if we have a sample with the label = 3, we should reshape its label into 4 (number of classes) values using One-hot Encoding. After that the label of this sample will be 0010. The 1 digit in the third position means that the original label was 3. We should perform this on all samples.

   Note that the initial value of weights should be an array of shape (number of features*number of classes).

   Considering 0.001 for learning rate and 1000 iterations, then we start learning the classifier on our train data. The results of testing our model on test set are as follows:

   Precision = 0.90

   Confusion Matrix: