Basic & Application of Machine Learning

Homework 2 – Logistic Regression



Student Name : Jason Melvin Tedjokusumo

Student ID : 32205061

Major : SW 융합모바일시스템공학과

Professor : 이규행교수님

Submission Date : October 9th, 2022

* Program Description

This assignment's primary goal is to use logistic regression to code a binary classifier. For a dataset with two features, denoted by x1 and x2, and each data point belonging to class 0 or 1, denoted by y, we are provided two files (train.csv and test.csv). The goal of this program is first, to train a classifier using train.csv file; and second, to classify the data in test.csv file using the trained model.

Task 1: Train a classifier with train.csv

The most important code lines for this are the logistic regression function, where we pass the ‘x1’, ‘x2’, and ‘y’ data values to the function. Since this is logistic regression, we are using the sigmoid function for our prediction. First, we initialize the parameters and the learning rate, then we go to the iteration to find the parameters and the loss/cost. In the loop, we obtain ‘y\_pred’ by using sigmoid function and we calculate the loss. Next, we calculate the partial derivatives of the cost function for each parameter, here since we are dealing with 2 features, that means we will have 3 parameters, for ‘b0’, ‘b1’, and ‘b2’. After that, just like gradient descent, we update the parameters in each calculation. Then, we will get the desired values for the parameters, and the cost.

Task 2: Classify the data in test.csv using the trained model

In here, we are using test data set to test our trained model in the task 1. So, we just use our model to predict the value of ‘y\_pred’ by giving the ‘x1\_test’ and ‘x2\_test’ and the final parameters to the sigmoid function. We can also get the loss for test data set. However, the most important part here is the threshold in which we want the predicted value to be 1 or 0. If the prediction is greater than 0.5, we count it as 1, and 0 otherwise. Then, we can calculate the accuracy of our prediction, by comparing the predicted y values with the actual y values divided by the total of y.

* How to Run

I use Google Colabs on a web browser to run the program. First, open Google Colabs, when prompted to open files, you can first upload the files to google drive then open it from google drive section, or you can upload it from local drive immediately.

Graphical user interface, application

Description automatically generated

After that, you need to upload the data files too, which are “hw2\_train.csv” and “hw2\_test.csv” files, by using the upload button just like in the picture.

A screenshot of a computer

Description automatically generated with medium confidence

Finally, you can run the program by pressing “Ctrl + f9” or just click run on each bar from top to bottom in orderly.

* Snapshots of Progress

This is task 1.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

* Results

1. Parameters of classifier

|  |  |  |  |
| --- | --- | --- | --- |
|  | b0 | b1 | b2 |
| Task1 | 1.92 | -3.30 | -0.98 |

1. Total cost for each task

|  |  |
| --- | --- |
|  | Cost |
| Task 1 | 47.40 |
| Task 2 | 13.51 |

1. Plot of the classification results

Chart, scatter chart

Description automatically generated

1. Accuracy: 96.4%

* Conclusion

I’ve tried to predict using LogisticRegression() function provided by the sklearn on the bottom of my code just to see and compare my result with the logistic regression from sklearn. I try to get closer to the accuracy from sklearn, by changing the learning rate. The data below is from my own trial:

1. Learning rate = 0.001, accuracy = 93.2%
2. Learning rate = 0.01, accuracy = 95.4%
3. Learning rate = 0.1, accuracy = 96.4%

In summary, by increasing the learning rate, we get higher and higher accuracy. For reference, I tried 0.5, and the accuracy is exactly the same with the LogisticRegression() function from sklearn, which is 96.6%