Assignment 3
Machine Learning using Spark and Scala

COMP4434

The Hong Kong Polytechnic University

1 Introduction

In this assignment, you are required to

- implement in Spark a multi-class logistic regression classifier using the **one-vs-all approach**, and
- train the classifier based on the example data and calculate the F_1 score of the classifier based on the testing data.

You shall use the following basic linear hypothesis:

$$h_\theta^{(i)}(x) = g(\theta_0^{(i)} + \theta_1^{(i)} x_1 + \theta_2^{(i)} x_2 + \cdots + \theta_n^{(i)} x_n)$$

i.e., you do not need to create extra features.

2 The Details

- You are given two folders */src* and */dat*. Folder */src* contains two scala files: *Main.scala* and *OneVsAllLogisticRegression.scala*.
- You are given two datasets under */dat*. Each dataset i contains three files:
  - dataset$i$_training.txt contains the training samples.
  - dataset$i$_testing.txt contains the testing data.
    These two files are in `LIBSVM` format (cf. lab material: *MLlib Basics*): `label, index1:value1 index2:value2 ...`
    Here, `label` is the classification label (i.e., with values $0, 1, \ldots, C-1$, where C is the number of classes).
  - dataset$i$_expected.txt contains the prediction result on the testing data if your implementation is correct. Specifically, the first line is the F_1 score and the remaining lines are predicted labels for testing points.

Note that we will use **some other datasets**, in addition to the given ones, for grading.

- Work on *OneVsAllLogisticRegression.scala*.
- In *OneVsAllLogisticRegression.scala*, it contains **three** empty functions for you to implement: `transform()`, `predict()`, and `calF1Score()`. These functions will be called by *Main.scala*, the driver program. Hence, **do NOT modify the signatures** of these three functions as well; Otherwise the driver program cannot compile and you will receive 0 mark. Other than these three functions, you are allowed to **add** new variables or functions to class `OneVsAllLogisticRegression`.
- To implement `calF1Score()` function, you are **not allowed** to use `MulticlassMetrics` class and invoke its APIs to return the metric directly.

3 What to submit?

Submit only OneVsAllLogisticRegression.scala (Don't change the file name or you will get 0 marks). Other files will be ignored.

Your submitted file should be free of compilation error and the program should terminate within 10 minutes. Otherwise you will receive 0 points.

4 What we do when we are grading

1. The data folder `dat` will be placed under /home/bigdata/Programs/spark/
2. Put some other datasets in the */dat* folder.
3. Export all codes as a `jar` file and execute it with the following command (as we do in labs; **notice under which directory we execute the command**):

   ```
   bigdata@bigdata-VirtualBox:~/Programs/spark$ bin/spark-submit
   --class "assignment3.Main" --master
   spark://localhost:7077 /path/to/your/jar
   ```

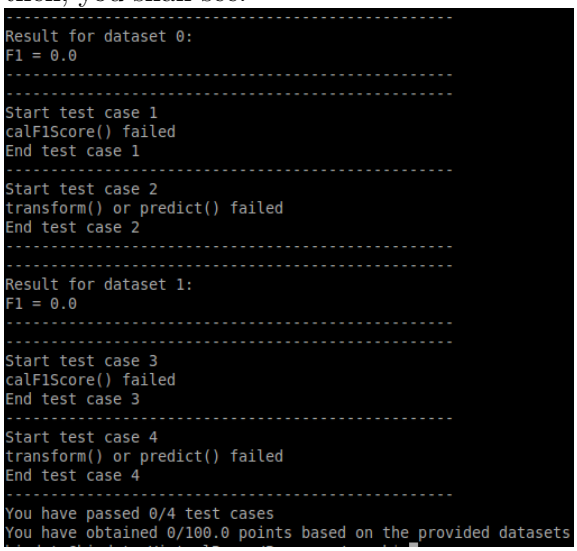
4. Obtain your assignment 3's score based on the output of Main.scala

5 Before vs. After you do the assignment

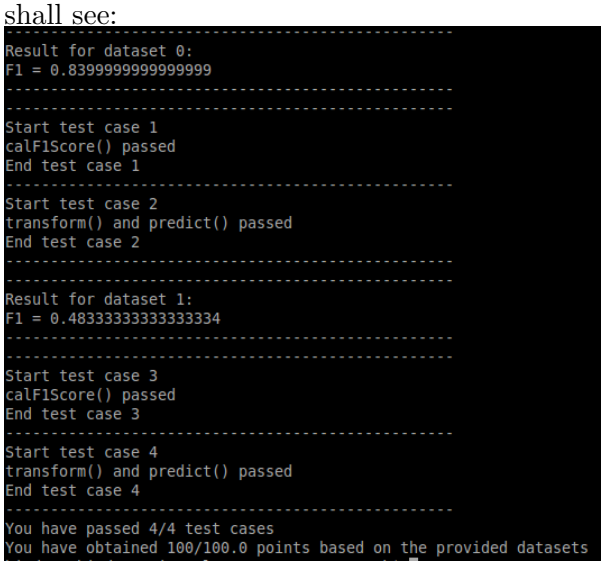
Before you do the assignment, if you get everything ready and

```
bin/spark-submit --class "assignment3.Main" --master
spark://localhost:7077 /path/to/your/jar
```

then, you shall see:



After you finish the assignment correctly and repeat above command, you shall see:



6 Grading rubrics

The driver program, Main.scala, contains a grading script with test cases. You can test whether your implementation is correct based on the given datasets. We will feed in 4 test cases using some other datasets when grading. Passing each test case will give you 25 marks.

7 Deadline

Deadline is 28 Mar, 2016, 11:59am.

Late Penalty: your score = your score $*(100 - 20x)\%$, where x is the number of days late.

8 Plagiarism

Your source code will be subjected to plagiarism check. Plagiarism cases will be strictly handled according to the University's regulation. So please don't risk doing that.

9 Question?

Please send email to Wenjian XU (cswxu@comp.polyu.edu.hk)