Assignment 5

MET CS 777 - Big Data Analytics
Classification - Logistic Regression, SVM, Dimensionality Selection
(20 points)

GitHub Classroom Invitation Link
`https://classroom.github.com/a/nkBNSm3C`

1 Description

In this assignment, you will implement logistic regression and support vector machines and compare the performance results in terms of accuracy and computation time. At the end of your classification task, you will reduce the feature matrix dimension using random selection and redo the classification tasks.

2 Data

You will be dealing with a data set that consists of around 170,000 text documents (this is 7.6 million lines of text in all), and a test/evaluation data set that consists of 18,700 text documents (almost exactly one million lines of text in all). All but around 6,000 of these text documents are Wikipedia pages; the remaining documents are descriptions of Australian court cases and rulings. At the highest level, your task is to build a classifier that can automatically figure out whether a text document is an Australian court case. We have prepared three data sets for your use.

1. The Training Data Set (1.9 GB of text). This is the set you will use to train your logistic regression model.
2. The Testing Data Set (200 MB of text). This is the set you will use to evaluate your model.
3. The Small Data Set (37.5 MB of text). This is for you to use for training and testing of your model locally, before you try to do anything in the cloud.

	Amzon AWS
Small Training Data Set (37.5 MB of text)	s3://metcs777/SmallTrainingData.txt
Large Training Data Set (1.9 GB of text)	s3://metcs777/TrainingData.txt
Test Data Set (200 MB of text)	s3://metcs777/TestingData.txt

Table 1: Data set on Amazon AWS - URLs

	Google Cloud Storage
Small Training Data Set (37.5 MB of text)	gs://metcs777/SmallTrainingData.txt
Large Training Data Set (1.9 GB of text)	gs://metcs777/TrainingData.txt
Test Data Set (200 MB of text)	gs://metcs777/TestingData.txt

Table 2: Data set on Google Cloud Storage - URLs

Some Data Details to Be Aware Of. You should download and look at the SmallTrainingData.txt file before you begin. You'll see that the contents are sort of a pseudo-XML, where each text document begins with a $< doc\ id = ... >$ tag, and ends with $< /doc >$.

Note that all of the Australia legal cases begin with something like < doc id = "AU1222" ... > that is, the doc id for an Australian legal case always starts with AU. You will be trying to figure out if the document is an Australian legal case by looking only at the contents of the document.

3 Assignment Tasks

3.1 Task 1 : Using Logistic regression model (7 points)

First, you need to write Spark code that uses the spark MLlib or ml library (RDD-Based or Dataframe) to train a logistic regression model on the given data based on term frequency of the top20k (20 thousand) words of the corpus. You should use the class

```
from pyspark.mllib.classification import LogisticRegressionWithLBFGS

or

from pyspark.ml.classification import LogisticRegression
```

Subtasks:

- Set the max number of iterations to 100
- Read the data, prepare the feature vector similar to Assignment 4 and train your model.
- Test the model based on the given train data and print out the F1-measure of the test.
- Print out the F1 measure
- Print out the total time to read the data, train and test the model each of the separately and the total time from start to end.

3.2 Task 2 - Implementing SVM model (7 Points)

In this task you should use Support Vector Machines to classify your data. All of the other parameters are similar to task 2. You are allowed to change SVM model parameters.

Note: Do not use any of the Spark Model libraries (mlib or ml) for this task. This should be your own implementation using numpy and pyspark only.

Subtasks:

- Read the data, prepare the feature vector similar to Assignment 4 and train your model.
- Test the model based on the given train data and print out the F-measure of the test.

- Print out the F1 measure
- Print out the total time to read the data, train and test the model each of the separately and the total time from start to end.

3.3 Task 3 - Weighted Loss Function (6 Points)

Implement the Weighted Loss function for SVM to solve the issues with the unbalanced data set in this assignment. Compute the ratios of Australia Court Cases and Wikipedia Documents, and use it to compute the weighted loss function, and use the gradient of it execute gradient descent.

4 Important Considerations

4.1 Machines to Use

One thing to be aware of is that you can choose virtually any configuration for your EMR cluster - you can choose different numbers of machines, and different configurations of those machines. And each is going to cost you differently!

Pricing information is available at: `http://aws.amazon.com/elasticmapreduce/pricing/`

Since this is real money, it makes sense to develop your code and run your jobs locally, on your laptop, using the small data set. Once things are working, you'll then move to Amazon EMR.

We are going to ask you to run your Spark jobs over the "real" data using 3 machines with **4 cores and 8GB RAM each** as workers. This provides 4 cores per machine (16 cores total) so it is quite a bit of horsepower. On the Google cloud take 4 machines with 4 cores and 8 GB of memory.

As you can see on EC2 Price list , this costs around 50 cents per hour. That is not much, but **IT WILL ADD UP QUICKLY IF YOU FORGET TO SHUT OFF YOUR MACHINES**. Be very careful, and stop your machine as soon as you are done working. You can always come back and start your machine or create a new one easily when you begin your work again. Another thing to be aware of is that Amazon charges you when you move data around. To avoid such charges, do everything in the "N. Virginia" region. That's where data is, and that's where you should put your data and machines.

- You should document your code very well and as much as possible.
- You code should be compilable on a unix-based operating system like Linux or MacOS.

4.2 Academic Misconduct Regarding Programming

In a programming class like our class, there is sometimes a very fine line between "cheating" and acceptable and beneficial interaction between peers. Thus, it is very important that you fully understand what is and what is not allowed in terms of collaboration with your classmates. We want to be 100% precise, so that there can be no confusion.

The rule on collaboration and communication with your classmates is very simple: you cannot transmit or receive code from or to anyone in the class in any way—visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading code to someone) or in any other way we have not yet imagined. Any other collaboration is acceptable.

The rule on collaboration and communication with people who are not your classmates (or your TAs or instructor) is also very simple: it is not allowed in any way, period. This disallows (for example) posting any questions of any nature to programming forums such as **StackOverflow**. As far as going to the web and using Google, we will apply the **"two line rule"**. Go to any web page you like and do any search that you like. But you cannot take more than two lines of code from an external resource and actually include it in your assignment in any form. Note that changing variable names or otherwise transforming or obfuscating code you found on the web does not render the "two line rule" inapplicable. It is still a violation to obtain more than two lines of code from an external resource and turn it in, whatever you do to those two lines after you first obtain them.

Furthermore, you should cite your sources. Add a comment to your code that includes the URL(s) that you consulted when constructing your solution. This turns out to be very helpful when you're looking at something you wrote a while ago and you need to remind yourself what you were thinking.

4.3 Turnin

Create a single document that has results for all three tasks. For each task, copy and paste the result that your last Spark job wrote to Amazon S3. Also for each task, for each Spark job you ran, include a screen shot of the Spark History.

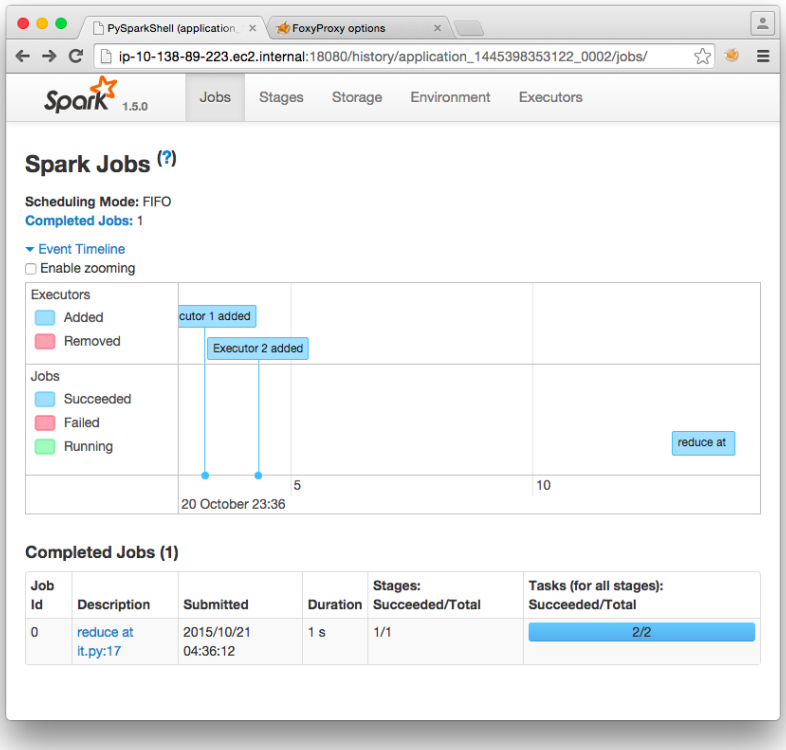

Figure 1: Screenshot of Spark History

Please zip up all of your code and your document (use .zip only, please!), or else attach each piece of code as well as your document to your submission individually.

Please have the latest version of your code on the GitHub. Zip the files from GitHub and submit as your latest version of assignment work to the Blackboard. We will consider the latest version on the Blackboard but it should exactly match your code on the GitHub