CS 4265 Homework #4

Write two Scala programs in Spark that performs linear regression on COVID-19 vaccination data. The first one will use Spark ML library and the second one will compute $\hat{\beta}_0$ and $\hat{\beta}_1$ in your code. There are two machine learning libraries (spark.ml and spark.mllib). In this homework, we will use spark.ml package. Below is an excerpt from Spark's documentation about the difference of the two machine learning libraries:

As of Spark 2.0, the RDD-based APIs in the `spark.mllib` package have entered maintenance mode. The primary Machine Learning API for Spark is now the DataFrame-based API in the `spark.ml` package.

What are the implications?

- MLlib will still support the RDD-based API in `spark.mllib` with bug fixes.
- MLlib will not add new features to the RDD-based API.
- In the Spark 2.x releases, MLlib will add features to the DataFrames-based API to reach feature parity with the RDD-based API.

Why is MLlib switching to the DataFrame-based API?

- DataFrames provide a more user-friendly API than RDDs. The many benefits of DataFrames include Spark Datasources, SQL/DataFrame queries, Tungsten and Catalyst optimizations, and uniform APIs across languages.
- The DataFrame-based API for MLlib provides a uniform API across ML algorithms and across multiple languages.
- DataFrames facilitate practical ML Pipelines, particularly feature transformations. See the Pipelines guide for details.

Dataset: country_vaccinations.csv, country_vaccinations.txt (Download from D2L software repository)

Input: USA data with date and people_vaccinated fields.

Output: linear regression models with MSE, $\hat{\beta}_0$ and $\hat{\beta}_1$.

Hints:

1. Try the Spark sample linear regression program at https://spark.apache.org/docs/latest/ml-classification-regression.html#linear-regression. Note that the dataset format is "libsvm" but the vaccination data is CSV. The libsvm format is a label, followed by an index (starting from 1) with a colon and a feature value. For example, "2.34 1:5.2 2:3.4" will specify a sample with a label 2.34 and two features 5.2 and 3.4.
2. The LinearRegression library works on datasets and dataframe views. So you have to convert data from CSV to a dataset with required columns "label" and "features" where

the features is a vector. You may use spark.read.csv() or sc.textFile() to load the CSV dataset. However, either one you will have to write your own parser for mapping.

3. Write Scala code to compute MSE, $\hat{\beta}_0$ and $\hat{\beta}_1$. Recall that we need to compute some summations before we compute MSE, $\hat{\beta}_0$ and $\hat{\beta}_1$. For example:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} X_i Y_i - \frac{\left(\sum_{i=1}^{n} X_i\right)\left(\sum_{i=1}^{n} Y_i\right)}{n}}{\sum_{i=1}^{n} X_i^2 - \frac{\left(\sum_{i=1}^{n} X_i\right)^2}{n}} = \frac{218 - \frac{(32)(24)}{4}}{296 - \frac{(32)^2}{4}} = 0.65$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X} = 6 - (0.65)(8) = 0.80$$

4. There are multiple ways to compute summations. The easiest way is reduction. For example, given x an array of doubles, x.foldLeft(0)(_+_) will sum all numbers in x. To compute sum of squares, simply map x to x^2 and reduce, such as x.map(Math.pow(_, 2.0)).foldLeft(0)(_+_). For the sum of x times y, you can zip them together and multiply each pairs accordingly such as x.zip(y).map{case (a, b) => a*b}.foldLeft(0)(_+_). You may use this technique to compute MSE as well.

What to submit?

Take a screenshot of your program outputs that contain MSE, $\hat{\beta}_0$ and $\hat{\beta}_1$ for both programs, and turn it in with your Scala programs in D2L.