Third Homework Assignment (AWS)

Due Date: **6/12/2019** at **5:00pm**

AWS Information

This homework assignment must be completed using Amazon Web Services. You are not allowed to use Wolf for any part of the assignment. The dataset is available in the Amazon S3 bucket: **s3://homework-trading/full_data.csv**. You must not move the data to an s3 bucket that you own (or any other bucket). The data is very confidential and thus it must not leave this bucket. You are not allowed to create a small subset for debugging and testing, to move it your laptop/mac and perform these tasks. For debugging and testing on your own laptop/mac, you must create a fictitious dataset. The assignment must be done in spark.

In order to gain access to the data file, you must send you canonical id to Tucker on slack in the direct channel (don't make it public since this is your AWS id). After you login, select the pull down menu with your name at the top left of the screen, then choose "My Security Credentials." Your canonical id is under "Account Identifiers."

Submission Guidelines:

Please follow carefully the instructions posted on the course's github page when submitting your solutions (https://github.com/MSIA/bigdatacourse/blob/master/README.md). Failure to follow the instructions will result in lost points.

Deliverables:

1. Your spark source code file: name the file "Exercise3.py/scala/java"
2. Output of your spark job: name the file "Exercise3.txt"
   The output file must include MAPE for each walk forward step. It must also include at the end the average MAPE score, the maximum one, and the minimum one.

Problem

Data: The csv file includes a descriptive header. The data is about trading 'profit' and features of a security (it is not a stock). There is a timestamp, bar number, profit, feature values from val12 to val78, and trade_id. The interpretation of these values is unknown. You should treat them as black-box features and include them in your analysis. All other fields are irrelevant for this homework assignment.

For a trade_id value, you can sort by bar_num (which should also sort the samples by the time stamp). This gives you profits in the correct sequential order, i.e., as they accumulate in time. A sequence is a term relating to all samples aka bars with the same trade id.

Description: The goal of the assignment is to predict the profit at the bar level. You will predict on increments of 10. That is, use bars 1-10 to predict 11-20. You then have access to bars 1-20 (including their profit values) to predict 21-30 (etc).

For example, you have the first 10 bars, and then for bar 11 you have: the time stamp and the feature values. You need to predict the profit for bar 1.

Next, you have the first 10 bars, the feature values and time stamps for bars 11 and 12 (note that you pshould not assume you have the profit for bars 11 and 12), and you need to predict the profit for bar 12. You then continue in this manner until bar 20.

At bar 21, assume that all profits are available for bars 1 to 20. Now you need to make predictions for bars 20-30 without knowing the profits for these bars, etc.

Regarding the evaluation process, the data spans years from 2008 to 2015. You should evaluate your model in the walk forward fashion. Train your model on the first 6 months of data. Then perform inference on the next one month. And then move forward for 6 months. You can think of this as a way to define your train/test split (clearly within train you can further divide to train/validate; this is completely up to you).

Example: train Jan 2008 to Jun 2008, inference on July 2008; train July 2008 to December 2008, inference on January 2009, etc.

You are free to use whatever classification model and also feature engineering is completely up to you. You can either create your own spark EC2 cluster (ill-advised) or use EMR.

By the way, the real-world problem was not about predicting profits, but when to sell/buy which is a much harder nut to crack.