**Alex Shepherd**

**PART 1: NO DOCKER**

First I used Spark 3.0.0 on my Mac to develop the scripts. Then I used an EMR clusterlaunched with Spark 2.4.5 and four instances. In the security groups for master I kept port 22 open and available to all IP addresses.

You log into the cluster using:

ssh -i [something].pem hadoop@[whatever].amazonaws.com

To transfer the files onto the master I used scp. There are two files, wineTasting.py and wineTesting.py! The first does the parallel training. It takes one command line argument that is the file path. To store the datasets, first I tried local storage on the EMR master but this did not work. Then I put the dataset files on S3 in a folder called pa2. This was successful.

To run the training script I typed:

spark-submit wineTasting.py s3://pa2/TrainingDataset.csv

To do testing locally on the master node I typed:

spark-submit --master local[\*] wineTesting.py s3://pa2/ValidationDataset.csv

The training and testing code is here:

<https://github.com/alexgshepherd/spark> This code only performs a linear regression and the fit is best measured by r^2.

**PART 2: DOCKER**

My Docker repo is at <https://hub.docker.com/repository/docker/alexgshepherd/test>

You should be able to run my container for model testing using the command:

docker run alexgshepherd/test

This is either locally or on AWS. You do not need any parameters because I packed all files into the container, although a parameter should not hurt. But on AWS I ran into some obstacles. First I had an error that the Docker daemon was not running so I started it with the command

sudo dockerd&

Then I got another error saying that the permission was denied while trying to connect to the Daemon socket. This was corrected using the command:

sudo usermod -aG docker hadoop

where “hadoop” is the user. And for some reason I had to log out and back in again. Then the docker run command worked!

I recommend using

docker run alexgshepherd/test 2> log.txt

to redirect all log4j output away so you only see console output, for clarity.