GitHub Link: <https://github.com/mt475/Programming-Assignment-2>

Docker Link: <https://hub.docker.com/r/mtahir3/docker-spark>

The above-mentioned GitHub repo has 2 projects (PredictOnly & TrainPredict) and a docker file. For both Java programs to run in IDE

we have to specify spark master url by -Dspark.master=url or localhost

1. **TrainPredict Project**: It has 2 main files, Training.Java and TrainingAndPrediction.Java. Based on the requirement we can either train, predict in one single application or save the trained model and load it in different project to predict.
   1. Training.Java: It takes 2 arguments file name and path to save the model. It trains the model and saves it for later use. This model is then loaded by PredictOnly.
   2. TrainingAndPrediction.Java: It takes 2 arguments training and prediction file location. It trains, predicts, and outputs the performance of the model. It is a single file that trains a model and predicts and outputs the performance.
2. **PredictOnly**: It takes 2 arguments, file location for prediction and model location. It loads the model from the provided location. It does prediction and outputs

the performance.

**-------------------- To run on EC2 Spark cluster -----------------**

1- copy the jar (TrainPredict-0.0.1-SNAPSHOT.jar & PredictOnly-0.0.1-SNAPSHOT.jar ) and csv files (TrainingDataset.csv & ValidationDataset.csv ) to EC2 by following commands.Here is the command pattern

scp -i <location of key> <location of file local> <user>@<public address of EC2>:<path on server>

**For example**:

scp -i labsuser.pem TrainPredict-0.0.1-SNAPSHOT.jar ec2-user@ec2-44-204-43-7.compute-1.amazonaws.com:~/.

scp -i labsuser.pem PredictOnly-0.0.1-SNAPSHOT.jar ec2-user@ec2-44-204-43-7.compute-1.amazonaws.com:~/.

2- SSH to EC2 instance. here is the command pattern

ssh -i <location of key> <user>@public address of ec2

**For example:**

ssh -i "labsuser.pem" ec2-user@ec2-44-204-43-7.compute-1.amazonaws.com

3- Configure AWS credentials on current EC2. We need this to login copy files to Spark cluster

4- Make sure flintrock is configured on the current EC2 instance, where files were copied. Please follow instructions provided by Professor.

5- Now copy Jars and CSV files to configured Spark cluster. Use the following flintrock command pattern

flintrock copy-file <cluster-name> <file location on ec2> <location on Spark Cluster>

**Example**

flintrock copy-file PA2 ./TrainPredict-0.0.1-SNAPSHOT.jar /home/ec2-user/

flintrock copy-file PA2 ./PredictOnly-0.0.1-SNAPSHOT.jar /home/ec2-user/

flintrock copy-file PA2 ./TrainingDataset.csv /home/ec2-user/

flintrock copy-file PA2 ./ValidationDataset.csv /home/ec2-user/

6- now login to flintrock by following command

flintrock login <cluster-name>

**Example**

flintrock login PA2

7- Confirm we got all the files in Spark master node by typing "ls".

8- Now run the jar by following command.

spark/bin/spark-submit --class com.PA2\_part1.TrainingAndPrediction --master spark://ec2-3-237-192-164.compute-1.amazonaws.com:7077 TrainPredict-0.0.1-SNAPSHOT.jar TrainingDataset.csv ValidationDataset.csv

**---------- DOCKER --------**

https://hub.docker.com/r/mtahir3/docker-spark

1- make sure docker is installed properly.

2- run the following command.

docker pull mtahir3/docker-spark:1.0.0

3- now run the downloaded image.

docker run -it mtahir3/docker-spark:1.0.0 /bin/bash

4- run the program with following command.

/opt/spark/bin/spark-submit --class com.PA2\_part1.TrainingAndPrediction --master local TrainPredict-0.0.1-SNAPSHOT.jar TrainingDataset.csv ValidationDataset.csv

**-------- local -----**

to run program in local IDE we have to pass following arguments in addition to spark.master url (-Dspark.master=local) in arguments.

1- TrainPredict Project --> Training.Java --> < Training file name> <model save location path>

--> TrainingAndPrediction.Java --> <Training CSV file name and location> <Prediction CSV file-name and location>

2- PredictOnly project --> PredictOnly.Java --> <prediction file name location> < model location>