

Wine Quality Prediction Tool User Guidance

Fan Yu

Github link for the code and dockerfile

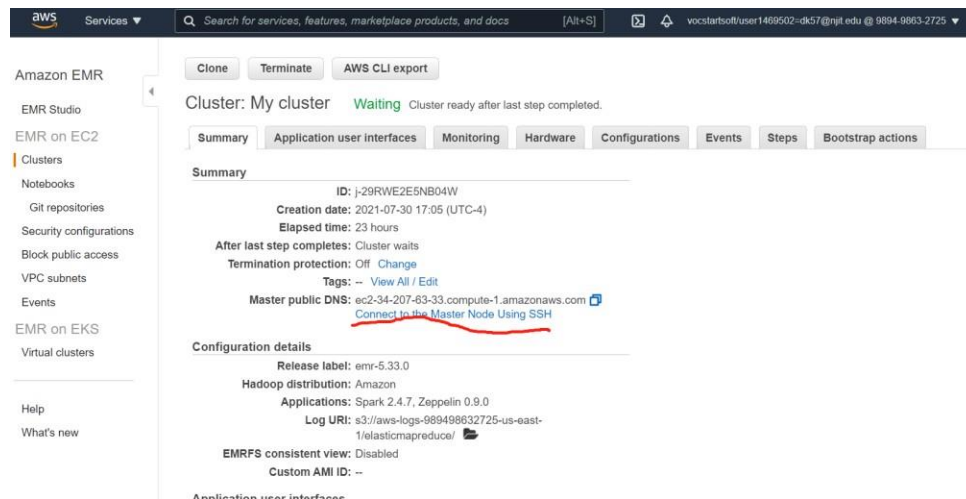
<https://github.com/kakatoto1/CS643.git>

DockerHub link for the docker image

<https://hub.docker.com/repository/docker/konaer/cs643>

Part1. Spark, AWS emr cluster training

1. In AWS, build EMR cluster to run spark, open ssh connection in the cluster in order to connect locally



2. Use ssh connection to upload training.py and training data, from the connection, we could see that we have 3 works to do the computing work for the program.

```
ubuntu@Master: ~$ start-workers.sh
172.31.89.84: org.apache.spark.deploy.worker.Worker running as process 2847. Stop it first.
172.31.92.99: starting org.apache.spark.deploy.worker.Worker, logging to /home/ubuntu/cs643/spark-3.1.2-
bin-hadoop3.2/logs/spark-ubuntu-org.apache.spark.deploy.worker.Worker-1-Master.out
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /home/ubuntu/cs643/spark-3.1.2-bin-
hadoop3.2/logs/spark-ubuntu-org.apache.spark.deploy.worker.Worker-1-Master.out
ubuntu@Master: ~$ jps
2296 Worker
3382 Jps
2490 Master
3275 Worker
ubuntu@Master: ~$
```

3. Run training.py, we have the trained_model file build, for us to run prediction latterly. From the running, we could also see that the F-measure for our algorithm is around 0.59.

Part2. Run predicting.py with our trained model and the data in your hands.

A. With docker

1. Please prepare your testdata.csv file to a specific directory that you would like to run.
2. Use following commend line to pull docker image to the same directory of your data file.
\$docker pull konaer/cs643:Dejing
3. Use the following commend line to run docker and test result.
\$docker run konaer/cs643:Dejing <yourTestFileName.csv>
4. Here's a result example, with ValidationDatas.csv as test data

```

hadoop@ip-172-31-60-95:~/cs643
21/07/31 02:54:40 INFO MapOutputTrackerMasterEndpoint: Asked to send map output locations for shuffle 3 to 172.31.4
:51690
21/07/31 02:54:40 INFO TaskSetManager: Finished task 0.0 in stage 126.0 (TID 125) in 18 ms on ip-172-31-49-119.ec2.
nal (executor 1) (1/1)
21/07/31 02:54:40 INFO YarnScheduler: Removed TaskSet 126.0, whose tasks have all completed, from pool
21/07/31 02:54:40 INFO DAGScheduler: ResultStage 126 (collectAsMap at MulticlassMetrics.scala:53) finished in 0.024
21/07/31 02:54:40 INFO DAGScheduler: Job 122 finished: collectAsMap at MulticlassMetrics.scala:53, took 0.114820 s
F-measure: 0.5904050519731796
21/07/31 02:54:40 INFO SparkContext: Invoking stop() from shutdown hook
21/07/31 02:54:40 INFO SparkUI: Stopped Spark web UI at http://ip-172-31-60-95.ec2.internal:4040
21/07/31 02:54:40 INFO YarnClientSchedulerBackend: Interrupting monitor thread
21/07/31 02:54:40 INFO YarnClientSchedulerBackend: Shutting down all executors
21/07/31 02:54:40 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
21/07/31 02:54:40 INFO SchedulerExtensionServices: Stopping SchedulerExtensionServices
(serviceOption=None,
services=List(),
started=false)
21/07/31 02:54:40 INFO YarnClientSchedulerBackend: Stopped
21/07/31 02:54:40 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
21/07/31 02:54:40 INFO MemoryStore: MemoryStore cleared
21/07/31 02:54:40 INFO BlockManager: BlockManager stopped
21/07/31 02:54:40 INFO BlockManagerMaster: BlockManagerMaster stopped
21/07/31 02:54:40 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
21/07/31 02:54:40 INFO SparkContext: Successfully stopped SparkContext
21/07/31 02:54:40 INFO ShutdownHookManager: Shutdown hook called
21/07/31 02:54:40 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-38a9db92-1163-49b3-88bb-e37110f8e8c3/
21/07/31 02:54:40 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-38a9db92-1163-49b3-88bb-e37110f8e8c3/

```

B. Without Docker

If you would like to run without docker, you need to follow steps below.

1. Install Java 8 or later
2. Install python 3.6+
3. Install PyNumpy
4. Install Apache Spark from official web or use pip pyspark to install pyspark
5. From github, pull training.py , predicting.py, TrainingDataset.csv to a specific directory
6. Run [python training.py TrainingDataset.csv] first, so that trained_model file could generated.
7. Run [python predicting.py yourTestData.csv], so that you could see prediction result and F measure.