

# CS643 Programming Assignment 2

## Cloud Computing

Sathvik Reddy Thogaru

**My GitHub repository link:** <https://github.com/sathvikreddy25/winePredict>

**Docker link:** <https://hub.docker.com/repository/docker/sathvikreddy968/predictingwine/general>

### Objectives

- to use Apache Spark to train an ML model in parallel on multiple EC2 instances
- to use Spark's MLlib to develop and use an ML model in the cloud
- to use Docker to create a container for your ML model to simplify model deployment

### STEPS ON HOW TO SET THE ENVIRONMENT FOR MODEL CREATION AND TRAINING USING AWS EMR.

1. Login into AWS Account.
2. Search for EMR in service > Open EMR.
3. Click on the Create Cluster Button.
4. Add Cluster Name. Here as we are using Spark we will Select Spark in Software Configuration. We need to run our program in 4 parallel EC2 instances so I created 5 instances 1 master and 4 slave. Select the keypair from the existing or you can also create one at that time and store at your local machine.
5. Click the button below to create the cluster.
6. After that your cluster will be in the Starting Stage and you will see the screen as below.
7. Click ssh to get login details for Master node of our EMR cluster > Save it.
8. In the security and access summary click on the security group for master to change inbound rules for traffic as below.

## Setup

First we have to create EMR cluster using AWS management console  
We should have 4 EC2 instances and using Spark for the ML application




In the 4 Ec2 instances we have one master node and 3 core nodes

After setting up our cluster we need to create S3 bucket which acts as storage and helps to fetch the required data faster and send output to that bucket

One bucket is created for the logs of the cluster and the other for storing our data

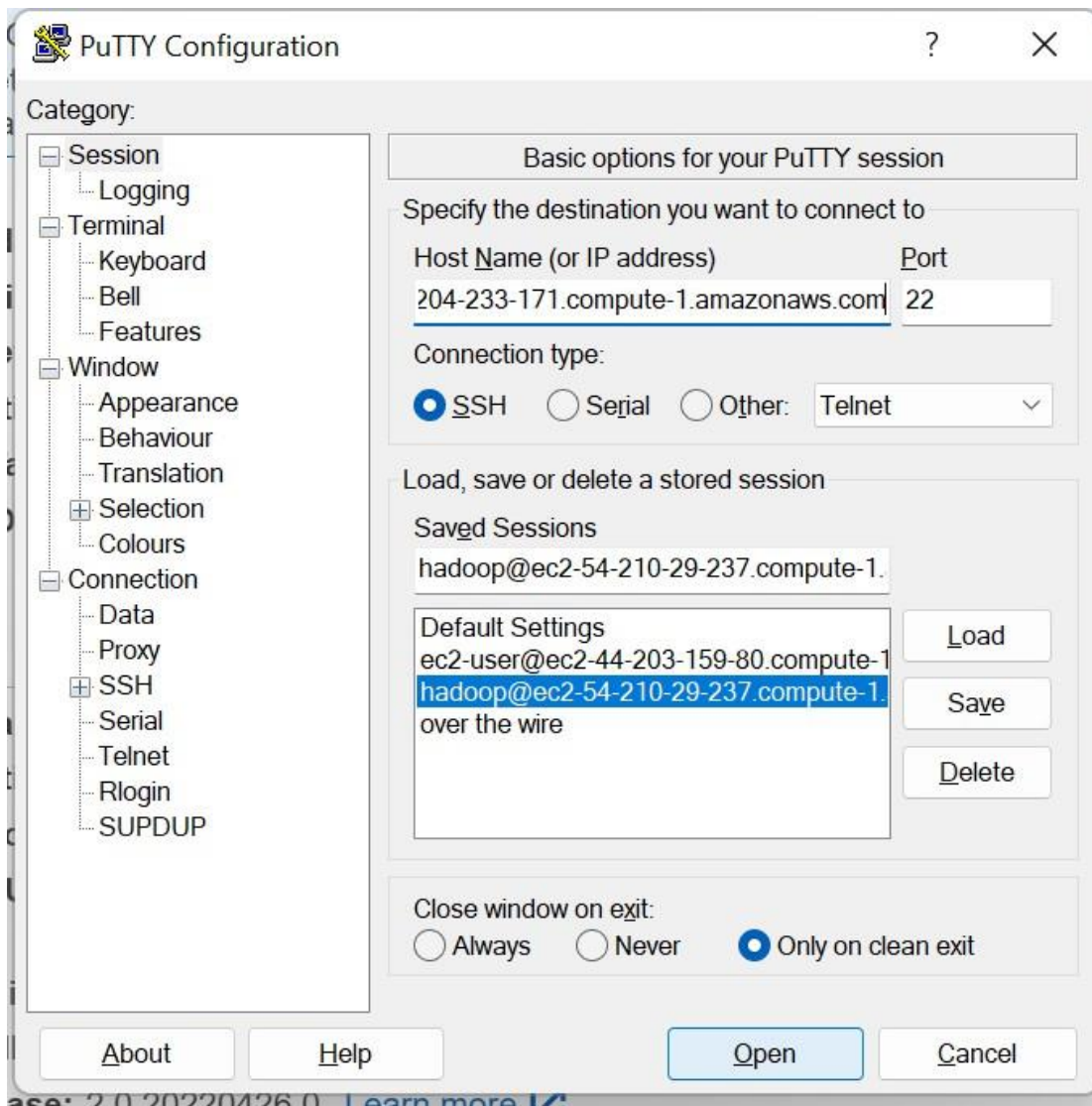
**Master: Running 1 m5.xlarge**

**Core: Running 3 m5.xlarge**

General purpose buckets (2) <a href="#">Info</a> <span>All AWS Regions</span>				
	 Copy ARN	Empty	Delete	Create bucket
Buckets are containers for data stored in S3.				
<input type="text" value="Find buckets by name"/> < 1 > 				
	Name ▲	AWS Region ▼	IAM Access Analyzer	Creation date ▼
<input type="radio"/>	<a href="#">aws-logs-us-east-1-cloud-computing</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	April 28, 2024, 23:33:57 (UTC+05:30)
<input type="radio"/>	<a href="#">my-bucket-source-cs643</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	April 28, 2024, 23:35:03 (UTC+05:30)

## Model training and Application Prediction

Now connect to the master instance using any SSH client . I am using Putty to connect to the master node. Then using WinSCP copy the spark application to run on the cluster



In the home directory i have my training.py file which i am using to train my model using the TrainingDataset.csv

By using the command 'spark-submit training.py' we can run the application on multiple clusters with the help of serialize

The s3 bucket URI is given in the document to fetch the files from the S3 bucket and for saving the model

I have also used ValidationDataset.csv to adjust my hyperparameters and tune it to highest score possible



```

C:\Users\pranavtalanki\Desktop\predictingwine>docker images
REPOSITORY      TAG         IMAGE ID      CREATED       SIZE
quelpred        latest      fb3d5b1d2a6a  3 minutes ago 1.41GB
predictingwine   latest      fb3d5b1d2a6a  3 minutes ago 1.41GB

C:\Users\pranavtalanki\Desktop\predictingwine>docker delete rmi quelpred
docker: 'delete' is not a docker command.
See 'docker --help'

C:\Users\pranavtalanki\Desktop\predictingwine>docker rmi quelpred:latest
Untagged: quelpred:latest

C:\Users\pranavtalanki\Desktop\predictingwine>docker images
REPOSITORY      TAG         IMAGE ID      CREATED       SIZE
predictingwine   latest      fb3d5b1d2a6a  4 minutes ago 1.41GB

C:\Users\pranavtalanki\Desktop\predictingwine>docker push sathvikreddy968/predictingwine:latest
The push refers to repository [docker.io/sathvikreddy968/predictingwine]
An image does not exist locally with the tag: sathvikreddy968/predictingwine

C:\Users\pranavtalanki\Desktop\predictingwine>docker push sathvikreddy968/predictingwine:tagname
The push refers to repository [docker.io/sathvikreddy968/predictingwine]
An image does not exist locally with the tag: sathvikreddy968/predictingwine

C:\Users\pranavtalanki\Desktop\predictingwine>docker tag predictingwine:latest

C:\Users\pranavtalanki\Desktop\predictingwine>docker tag predictingwine:latest sathvikreddy968/predictingwine:latest

C:\Users\pranavtalanki\Desktop\predictingwine>docker push sathvikreddy968/predictingwine:tagname
The push refers to repository [docker.io/sathvikreddy968/predictingwine]
tag does not exist: sathvikreddy968/predictingwine:tagname

```

```

C:\Users\pranavtalanki\Desktop\predictingwine>docker push sathvikreddy968/predictingwine:tagname
The push refers to repository [docker.io/sathvikreddy968/predictingwine]
tag does not exist: sathvikreddy968/predictingwine:tagname

C:\Users\pranavtalanki\Desktop\predictingwine>docker push sathvikreddy968/predictingwine:latest
The push refers to repository [docker.io/sathvikreddy968/predictingwine]
5cbe602fbd8b: Pushed
04a7a8f27da1: Pushed
3a62265466ea: Pushed
728c8a35de40: Pushed
0b2604e31ab9: Pushed
4e8e8ab1de30: Pushed
b66078cf4b41: Mounted from library/openjdk
cd5a0a9f1e01: Mounted from library/openjdk
eafe6e032dbd: Mounted from library/openjdk
92a4e8a3140f: Mounted from library/openjdk
latest: digest: sha256:03c90c527207238da66f0a4dd0b31a273dbe69f5c8da4adb396c811729f8e42e size: 2418

```

## Steps to run the model prediction without Docker

1. Create an EC2 instance.
2. Ssh into your EC2 instance.
3. Configure Spark in to your instance.
4. Pull the GitHub repository in your instance.
5. Create a Directory named predictingwine by running the below command and put your test data set into that directory. Command : `mkdir predictingwine`
6. Now for the model prediction run the below command. Command : `spark-submit --packages org.apache.hadoop:hadoop-aws:2.7.7 test.py`

## Steps to run the model prediction with Docker

1. Create an EC2 instance.
2. Ssh into your EC2 instance.
3. Install the docker into your EC2 instance
4. Add your test data file in your EC2 instance.
5. Run the Command : `docker pull sathvikreddy968/predictingwine:latest`
6. By running the above command the user will get the Docker image.

7. Now run the below command in the directory where your test data file is there.
8. Run the Command : `sudo docker run -it -v "$(pwd)":/sathvikreddy968/predictingwine:latest`
9. This will give the output of Model Accuracy and F1 score as shown below.

## Results

Code tested ValidationDataset.csv and the scores are as below

The F1- score we are seeing is the highest score achieved using RandomForestTrees We can also see scores of other models like LogisticRegression, NaiveBayes, gradientBoostedTrees which are comparatively lowers than RandomForestTrees

```

model accuracy 0.40625
LR F1 Score = 0.53125
F1- score: 0.75
[[10  2  0  0]
 [ 2 10  1  0]
 [ 1  2  3  0]
 [ 0  0  0  1]]

```

	precision	recall	f1-score	support
5.0	0.77	0.83	0.80	12
6.0	0.71	0.77	0.74	13
7.0	0.75	0.50	0.60	6
8.0	1.00	1.00	1.00	1
accuracy			0.75	32
macro avg	0.81	0.78	0.79	32
weighted avg	0.75	0.75	0.74	32

```

Accuracy 0.75

```

Now we have to create Docker container and run our application on it using the docker file created

We use following commands to run docker container using the docker image  
`docker build -t docker-ml-model -f Dockerfile .`  
`docker run docker-ml-model`

After creating our docker successfully we can run our prediction test