**README:**

Develop the spark application code for the training model and prediction model

Create a EMR Cluster by following the below steps

**Create EMR Cluster:**

* Login to the AWS Console using AWS Learner Lab
* Search for EMR and navigate to the EMR page
* Click on the Create Cluster button
* In the Create Cluster page, provide the following information and leave the default value for other fields
  + Name - <Name of your cluster>
  + Cluster Logs – Select a S3 bucket and add /logs to ensure that the logs are added to the new directory. [Note: Create a new S3 bucket prior to adding the Cluster Logs]
  + Provide the cluster size
  + Select the Amazon EC2 Key pair in the Security configuration and EC2 key pair section which would enable us to login to the EMR Master [Rules for Security group should be modified to allow SSH connection]
  + Select the service role – EMR\_DefaultRole for Amazon EMR Service Role
  + Select the instance profile – EMR\_EC2\_DefaultRole in the EC2 instance profile for Amazon EMR section
* Click on the Create Cluster button
* Navigate back to the EMR on EC2: Clusters page
* The status of the EMR Cluster created should be Starting and should be changing to Waiting so that the steps can be added and have the Spark application run on the cluster.

**Adding a Step to run the Spark Application:**

To run the spark application on the EMR Cluster, follow the below steps

* Find the EMR Cluster created in the EMR on EC2: Clusters page and click on the Cluster ID hyperlink which would take us to the Cluster page
* Click on the Steps tab
* Click on Add Step button
* Steps Configuration
  + Select Spark application for the Type field
  + Provide a name for the step
  + For the application location, select the spark application code file from the S3 bucket [Note: The files needed to be moved to the S3 bucket prior to executing this step]
  + Click on Add
* Once the step is added, the spark application will start running.
* If the application fails, the cluster will get terminated and saves the logs to the s3 bucket path provided during Cluster Creation
* If the application runs successfully, the cluster will get terminated depending on the Idle time entered in the Cluster creation process.

**Training Model Code Considerations:**

* The training dataset will be retrieved from an s3 bucket.
* CrossValidator will be utilized to identify the best model, which will be saved back to the s3 bucket.

**Prediction Model Code Considerations:**

* The application will be containerized using Docker, and a Dockerfile will be created to include the source code and dataset.
* During the execution of the application using "docker run" command, the dataset will be provided as an argument.
* The best model, saved in the s3 bucket during training, will be accessed and loaded.
* Once the model is loaded, predictions will be executed on the validation dataset.

**Executing the Prediction Model:**

1. Create a new EC2 instance (ensure that the default size of the instance is changed from 8 GB to 24 GB). The size was changed so that the docker build is running successfully importing the required images to run the spark application.
2. Login to the EC2 instance using SSH
3. Execute the below commands in the same order to install the docker in the ec2 instance and run the docker commands

sudo apt-get update

curl -fsSL https://get.docker.com -o get-docker.sh

sudo sh get-docker.sh

sudo usermod -aG docker ubuntu [this command is executed to allow the user ubuntu to run the docker commands]

1. Create a new directory in ec2 instance
2. Copy the docker file to the new directory created in the above step using SFTP.
3. Execute the following command to build the docker image

docker build -t cs643\_assignment2 -f DockerFile .

wineprediction is the name of the docker image

DockerFile is the name of the docker file created. If the default file name is used, -f DockerFile can be removed from the command.

1. Execute the following command to run the spark application using docker

docker run –env-file aws\_credentials.env cs643\_assignment2 spark-submit wineprediction\_app.py ValidationDataset.csv

**Assumption for this command:** The spark application code and the dataset will be copied to the docker app which will be mentioned in the docker file.

Screenshots for the following items are added to the document – **<<Screenshots.docx>>**

* Adding EMR Cluster
* Adding a step to the EMR Cluster and executing the Spark application
* Docker Execution – docker build & docker run commands

**Challenges Encountered – Running Prediction Model**

While attempting to copy data from the S3 bucket to either the EC2 instance or the Docker container, we encountered difficulties due to restrictions imposed by the bucket policies and IAM user permissions.

**Attempted Solution**

To address the issue, we made modifications to the S3 bucket's restrictions by disabling the "Block Public Access" option. Additionally, we implemented a public policy that allows copying or accessing the best model identified by CrossValidator in the Spark Application. The policy is as follows:

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "AllowAllS3ActionsInUserFolder",

"Effect": "Allow",

"Principal": {

"AWS": "\*",

"Service": "s3.amazonaws.com"

},

"Action": "s3:\*",

"Resource": [

"arn:aws:s3:::cs643-programming-assignment-2",

"arn:aws:s3:::cs643-programming-assignment-2/\*"

]

}

]

}

However, despite implementing the above solution, we were still unable to access the S3 bucket using our code.

**Workaround:**

As a workaround, we manually downloaded the best model from the S3 bucket to a local folder. Subsequently, we copied the model to the directory on the EC2 instance where we executed Docker commands. To accommodate these changes, we made necessary modifications to the application code for the prediction model to load the best model successfully.