**Wine Quality Prediction on AWS**

**Introduction**

The goal of this project is to develop a machine learning (ML) model for predicting wine quality using Apache Spark on AWS EMR. The assignment involves parallel training on a Spark cluster managed by EMR, single-node predictions, Docker containerization for deployment, and sharing the code repository on GitHub for collaboration and evaluation.  
  
Link to GitHub: [Wine\_Quality\_Prediction\_inAWS](https://github.com/pruthvidholakia/Wine_Quality_Prediction_inAWS)  
Link to Docker: [wine-quality-prediction](https://hub.docker.com/r/pruthvidholkia/wine-quality-prediction)

**Tools Used**

1. AWS EMR: Managed Hadoop framework to set up and run Apache Spark clusters for distributed computing.
2. Apache Spark: Framework for distributed data processing and machine learning with MLlib. Used for model training and prediction tasks.
3. Python as Programming language for implementing training and prediction code
4. Sparks MLlib: Machine learning library for classification, regression, and evaluation metrics.
5. Library for implementing machine learning models like RandomForestClassifier.
6. Docker: Containerization platform to package and deploy the prediction application.
7. AWS S3: Cloud storage for datasets and the trained model pipeline.
8. GitHub: Repository for storing and sharing project code, scripts, and documentation.

**Infrastructure Setup**

1. **Create s3 Bucket**

Create an S3 bucket in our aws cloud to store training/predicting script, dataset and our model after getting trained.

* S3 bucket as aws: pa2winequalitybucket

1. **Create Key-pair for our EMR cluster**

* EC2 > Network > Key-pairs
* Create key as: - pa2\_pruthvidholkia\_WineQuality.pem

1. **Now let’s create an EMR cluster as per project requirement for parallel job**

* Name: **my\_cluster\_wineQuality\_predict\_18**
* Create Cluster > Application bundle custom (Hadoop, spark) > OS (Amazon Linux)
* Cluster Config > Core and Task(m5.xlarge) > Cluster scaling and provisioning (set task to 3)
* Networking > subnet >

browse and select different region if cluster gets terminated because of instance type (m5).

* Security Config > Amazon EC2 keypair for ssh > pa2\_pruthvidholkia\_WineQuality
* IAM Role > EMR\_DefaultRole
* A screenshot of a computer

  Description automatically generatedEC2 Instance > EMR\_DefaultRole

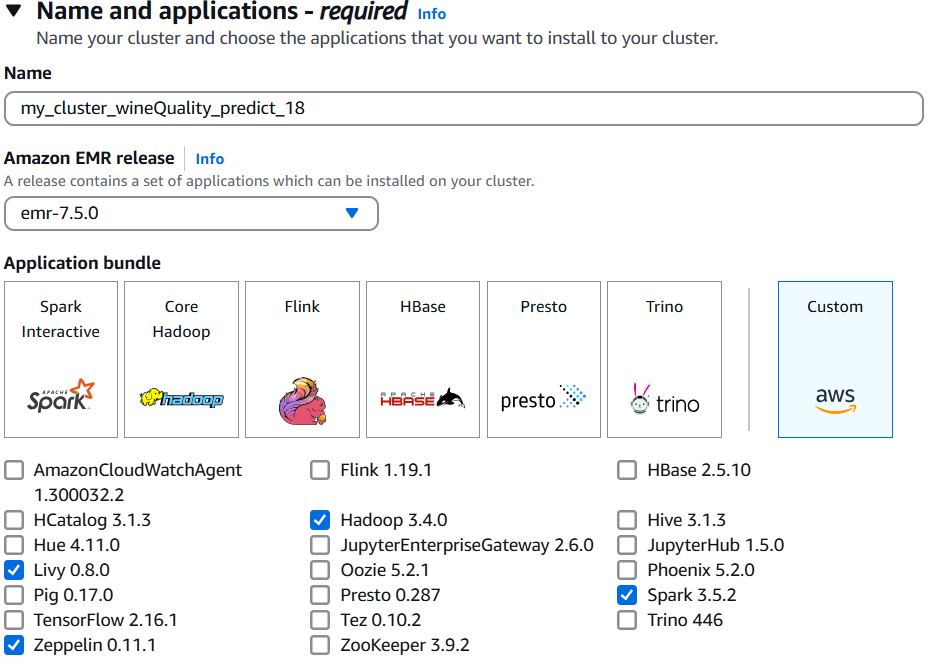


Figure 1

Figure 2

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Figure 3

Figure 4

1. **Parallel Model Training on EMR and Predicting**

After creating cluster upload your training.py and predicting.py and both the datasets to your s3 because now we will run our cluster and train our model parallelly in 4 instances.

* ssh your EMR cluster with PuTTy:

A screenshot of a computer

Description automatically generatedpass your pa2\_pruthvidholkia\_WineQuality.pem key to PuTTy   
pass your public DNS in Hostname or IP   
[hadoop@ec2-18-205-20-68.compute-1.amazonaws.com](mailto:hadoop@ec2-18-205-20-68.compute-1.amazonaws.com)

A screen shot of a computer

Description automatically generated

Figure 5

Figure 6

* Install numpy, imbalanced-learn and pandas in your EMR

pip install numpy

pip install imbalanced-learn

pip install pandas

* Run this cmd to execute train\_wineQuality.py which performs data preprocessing, training model and saving it in same s3 location:  
   spark-submit s3://pa2winequalitybucket/train\_wineQuality.py
* A black screen with white text

  Description automatically generatedAfter a while your model will get traind and get saved to you s3 bucket which you will pass in train file

1. **Now Let’s test our model on validation dataset**

Figure 7

* Pass this cmd in your Putty terminal as you did for training part:  
  spark-submit s3://pa2winequalitybucket/predict\_wineQuality.py

Once the predict\_wineQuality runs successfully in your EMR you could see **f1 score** as below:

A black screen with white text

Description automatically generated

Figure 8

* This prediction will run on single ec2 instance the master node or master ec2 as per project requirement and how we achieve this in predict file you should see this:

A red box with a black background

Description automatically generated

Figure 9

* Since the script explicitly specifies local[\*], even if the EC2 instance is part of a cluster, this job won't utilize other nodes in the cluster.

1. **Docker Containerization (Running application with Docker)**

Objective: Package the prediction application for deployment.

Steps:

* Load all files and train model folder in you EMR.  
  aws s3 cp s3://pa2winequalitybucket/train\_wineQuality.py ./train\_wineQuality.py

aws s3 cp s3://pa2winequalitybucket/predict\_wineQuality.py ./predict\_wineQuality.py

aws s3 cp s3://pa2winequalitybucket/requirements.txt ./requirements.txt

aws s3 cp s3://pa2winequalitybucket/TrainingDataset.csv ./TrainingDataset.csv

aws s3 cp s3://pa2winequalitybucket/ValidationDataset.csv ./ValidationDataset.csv

aws s3 cp s3://pa2winequalitybucket/optimized\_wine\_model\_rf.model ./optimized\_wine\_model\_rf.model –recursive

* Created a Dockerfile with the following components:

Spark setup.  
 Python dependencies.  
 Train and prediction script (train\_wineQuality.py, predict\_wineQuality.py).

* Create the image called wine-quality-prediction  
  sudo docker build -t wine-quality-prediction .
* Login to docker with sudo login docker
* sudo docker tag wine-quality-prediction pruthvidholkia/wine-quality-prediction:latest
* Push: sudo docker push pruthvidholkia/wine-quality-prediction:latest

A screenshot of a computer

Description automatically generated

Figure 10