**Links:**

**Github:** [CS-643-programming-project-2/CS643-Programming Project-Wine Prediction-Edward-Salib.docx at main · EdwardRiad/CS-643-programming-project-2 (github.com)](https://github.com/EdwardRiad/CS-643-programming-project-2)

**Docker Hub:** [Repositories | Docker Hub](https://hub.docker.com/repositories/edwardsamirriad)

|  |
| --- |
| **S3 bucket**   * I created a S3 bucket and gave it the name: [**progproj2-s3**](https://s3.console.aws.amazon.com/s3/buckets/progproj2-s3?region=us-east-1)   url: s3://progproj2-s3   * Uploaded the training dataset (TrainingDataset.csv) and the validation dataset (ValidationDataset.csv). * Uploaded training.py and predict.py |

|  |
| --- |
| **Cluster**   * In AWS dashboard select EMR * Click on create cluster * I entered ProgProj2 for the cluster name |
| * Selected Spark as Application bundle * Select instance groups under Cluster configuration * select m4.large as EC2 instance type | | |

|  |
| --- |
| * Under Cluster scaling and provisioning option select set cluster size manually and set the number of instances to 4. * Under Security configuration and EC2 key pair select click the EC2 key pair already created else create a new one * Click Create Cluster button. Wait for around 15 minutes for the cluster to start functioning. |

|  |
| --- |
| EC2 instance without Docker  * In AWS dashboard select EC2 * Click Launch instances * Under Application and OS Images (Amazon Machine Image) select Amazon Linux 2 Ami * Under instance time select  t2.micro * Select key pair and create security group then click on launch instance. |

|  |
| --- |
| Installing Spark on EC2 instanceUpdate EC2 using the command  *sudo yum update -y*Install java 8 *sudo apt install openjdk-8-jdk**Check java version java –version* |

|  |
| --- |
| Install python3 sudo apt install python3.8Check python version python3 --version |

|  |
| --- |
| sudo pip install --upgrade pip Install Py4j sudo pip install py4j |

|  |
| --- |
| * Install findspark **sudo pip install findspark** |

|  |
| --- |
| Install Spark and Hadoopwget http://archive.apache.org/dist/spark/spark-3.0.0/spark-3.0.0-bin-hadoop2.7.tgzUnzip the package sudo tar -zxvf spark-3.0.0-bin-hadoop2.7.tgzInstall pandas pip install pandas |

|  |
| --- |
| Install scikit-learn pip install scikit-learn |

|  |
| --- |
| * Running the training model in EC2 * **spark-submit --packages org.apache.hadoop:hadoop-aws:2.7.7 s3://progproj2-s3/training.py** |

|  |
| --- |
| **The model is saved in S3** |

|  |
| --- |
| * Running the validation model in EC2   **spark-submit --packages org.apache.hadoop:hadoop-aws:2.7.7 s3://progproj2-s3/predict.py** |

|  |
| --- |
| **EC2 Instance with docker**     * Install the most recent Docker Community Edition package **sudo yum install docker –y** |

|  |
| --- |
| * Start the Docker service **sudo service docker start** * Add the ec2-user to the docker group so you can execute Docker commands without using sudo **sudo usermod -a -G docker ec2-user** * Verify that the ec2-user can run Docker commands without sudo  **docker --version or docker info** |

|  |
| --- |
| Building Dockerfile  * We will create a directory named MyDockerImages with the command:   **mkdir MyDockerImages**   * Move into that directory and create a new empty file (Dockerfile) in it by typing:   **cd MyDockerImages**  **then touch Dockerfile**   * Open the file with a text editor of your choice. In this example, we opened the file [using Nano](https://phoenixnap.com/kb/use-nano-text-editor-commands-linux#htoc-how-to-open-nano): **nano Dockerfile** * Then, add the following content:   **FROM datamechanics/spark:3.1-latest**  **ENV PYSPARK\_MAJOR\_PYTHON\_VERSION=3**  **RUN pip install --upgrade pip --user \**  **&& pip install scikit-learn \**  **&& pip install pandas \**  **&& pip install pyspark**  **COPY TrainingDataset.csv ~/MyDockerImages/**  **COPY ValidationDataset.csv ~/MyDockerImages/**  **COPY predict.py ~/MyDockerImages/**  **COPY training.py ~/MyDockerImages/**  **CMD python training.py**  **ENTRYPOINT ["python","predict.py"]** |

|  |
| --- |
| * Save and exit the file. * You can check the content of the file by using the cat command: **cat Dockerfile** |

|  |  |
| --- | --- |
| * Name the image as pproj2 **sudo docker build . -f Dockerfile -t cs643-pproject-two** | |
| * **sudo docker run cs643-pproject-two**   + **sudo docker images** | |

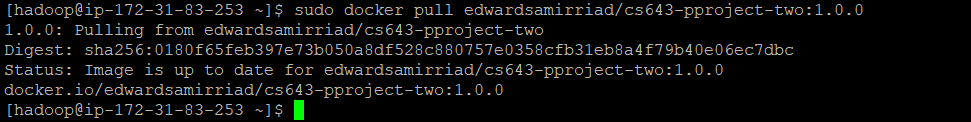
|  |
| --- |
| * push the image into the dockerhub **sudo docker login** |

|  |
| --- |
| * give a tag to image so that version your image to multiple images.   **sudo docker tag cs643-pproject-two edwardsamirriad/cs643-pproject-two:1.0.0**  then **sudo docker images** |

|  |
| --- |
| * pushing your Docker image back to Ec2   **sudo docker push edwardsamirriad/cs643-pproject-two:1.0.0** |

* pushing your Docker image from Ec2

**sudo docker pull edwardsamirriad/cs643-pproject-two:1.0.0**

****

|  |
| --- |
| **Running chatGpt model on EC2**    I generated two codes in ChatGPT one is for training and the second is for prediction   * I gave the training model the name "[chagtGptTraining.py](https://github.com/EdwardRiad/CS-643-programming-project-2/blob/main/chagtGptTraining.py)” and the validation model the name “[chagtGptPrediction.py](https://github.com/EdwardRiad/CS-643-programming-project-2/blob/main/chagtGptPrediction.py)” and then uploaded the two files onto theS3 bucket [**(progproj2-s3**](https://s3.console.aws.amazon.com/s3/buckets/progproj2-s3?region=us-east-1)**)**   Then I did all the steps as in “without docker” section:  **spark-submit --packages org.apache.hadoop:hadoop-aws:2.7.7 s3://progproj2-s3/** **chagtGptPrediction.py**  **and then**  **spark-submit --packages org.apache.hadoop:hadoop-aws:2.7.7 s3://progproj2-s3/** **chagtGptPrediction.py**  **I had some errors with the code so I had to solve them first before applying the previous codes**  **After that I repeated the same steps as in “with docker” section**  **This time I opened a file with nano text editor** Building chatGPT Dockerfile  * We will create a directory named MyDockerImages with the command:   **mkdir ChatGPT-DockerImages**   * Move into that directory and create a new empty file (Dockerfile) in it by typing:   **cd ChatGPT-DockerImages**  **then touch Dockerfile**   * Open the file with a text editor of your choice. In this example, we opened the file [using Nano](https://phoenixnap.com/kb/use-nano-text-editor-commands-linux#htoc-how-to-open-nano): **nano Dockerfile** * Then, add the following content:   **FROM datamechanics/spark:3.1-latest**  **ENV PYSPARK\_MAJOR\_PYTHON\_VERSION=3**  **RUN pip install --upgrade pip --user \**  **&& pip install scikit-learn \**  **&& pip install pandas \**  **&& pip install pyspark**  **COPY TrainingDataset.csv ~/ChatGPT-DockerImages /**  **COPY ValidationDataset.csv ~/ChatGPT-DockerImages /**  **COPY chagtGptPrediction.py ~/ChatGPT-DockerImages /**  **COPY chagtGptTraining.py ~/ChatGPT-DockerImages /**  **CMD python chagtGptTraining.py**  **ENTRYPOINT ["python","** **chagtGptPrediction.py"]**   * Save and exit the file. * You can check the content of the file by using the cat command: **cat Dockerfile**      * **sudo docker build . -f Dockerfile -t cs643-chatgpt-proj**      * + **sudo docker images**     **sudo docker tag cs643-pproject-two edwardsamirriad/cs643-chatgpt-proj:1.0.0**   * pushing your Dockerimage back to Ec2   **sudo docker push edwardsamirriad/cs643-chatgpt-proj:1.0.0**       * pull hing your Dockerimage from Ec2   **sudo docker pull edwardsamirriad/cs643-chatgpt-proj:1.0.0** |