Github link: [EdwardRiad/CS643\_Programming\_Project\_2: NJIT Cloud Computing-CS 643 programming project 2 (github.com)](https://github.com/EdwardRiad/CS643_Programming_Project_2)

Dockerhub link: [edwardsamirriad/cs643-pproj2 general | Docker Hub](https://hub.docker.com/repository/docker/edwardsamirriad/cs643-pproj2/general)

**Project Description:**

**First step create a bucket**

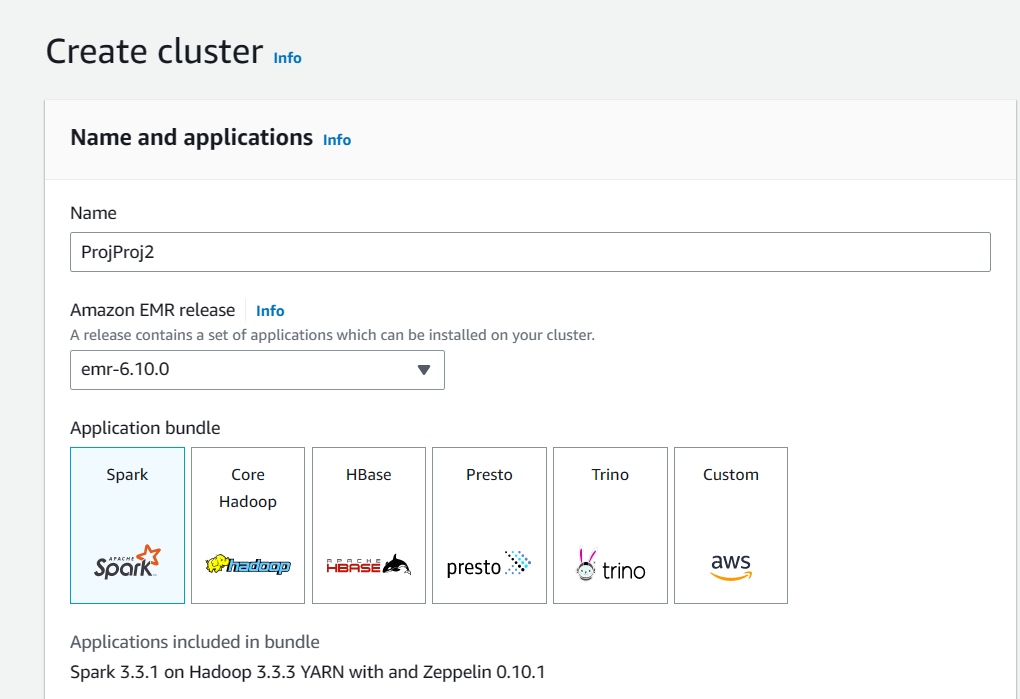
* I created a S3 bucket and gave it the

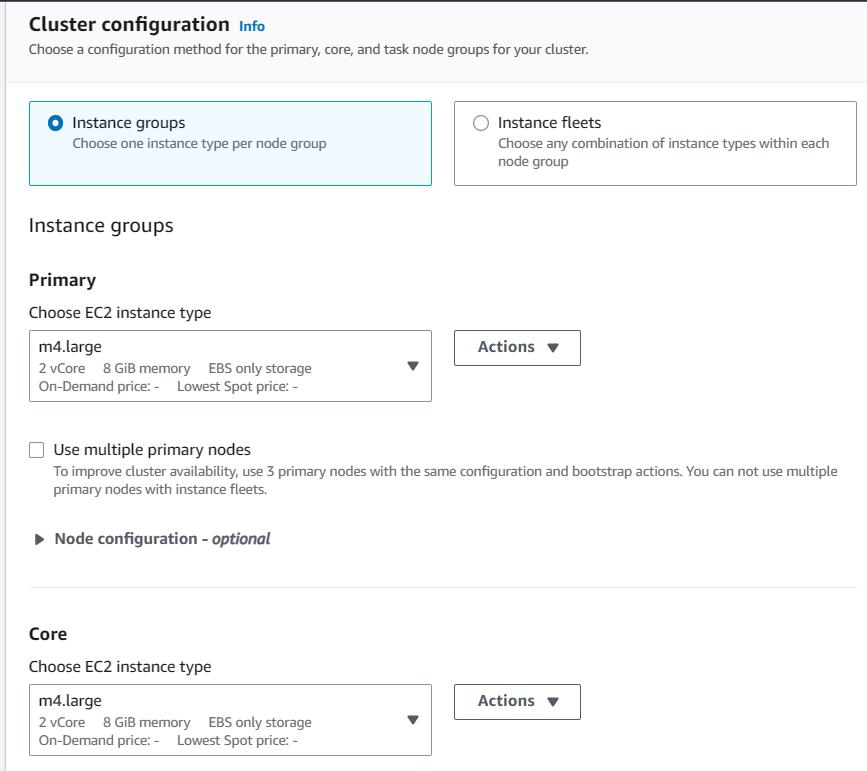
name: [progproj2-s3](https://s3.console.aws.amazon.com/s3/buckets/progproj2-s3?region=us-east-1)

url: s3://progproj2-s3

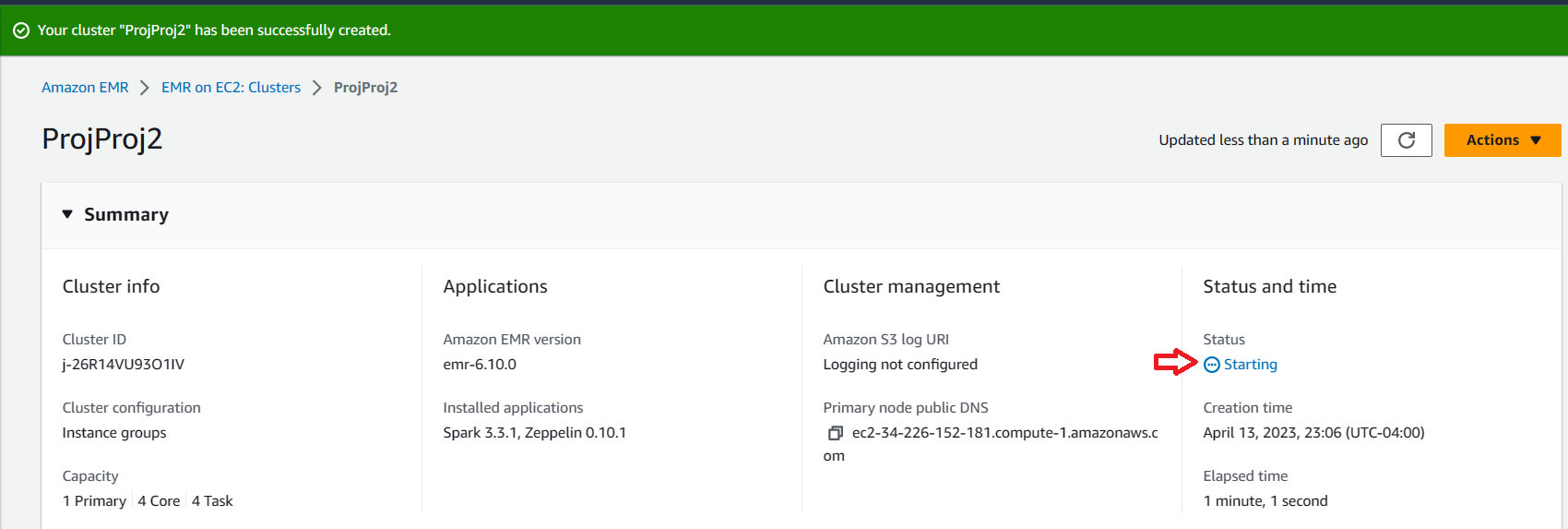
* Upload the training dataset (TrainingDataset.csv) and the validation dataset (ValidationDataset.csv).
* Upload .csv files and training.py and predict.py

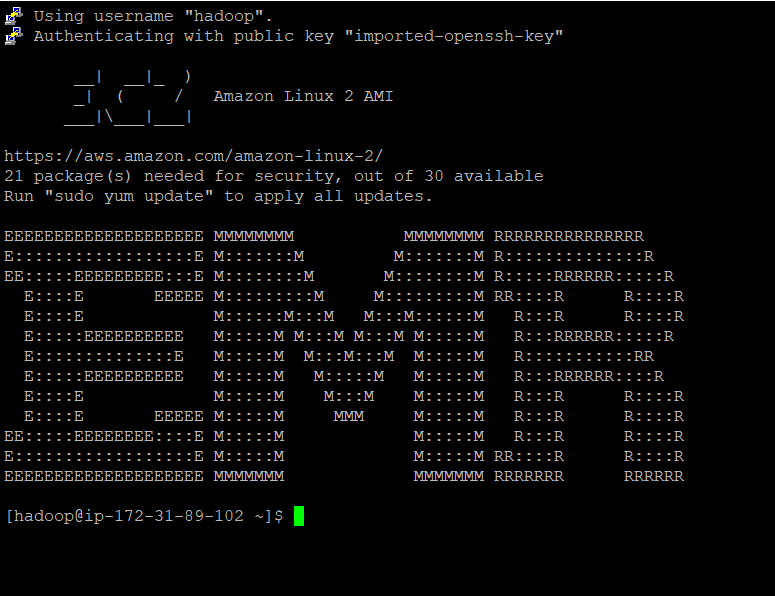
**Second step create a cluster**

* 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.





### Third Step create 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.

### Fourth Step installing Spark on EC2 instance

### Update 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 python3.8

### Check python version python3 --version

### 

### sudo pip install --upgrade pip

### Install Py4j sudo pip install py4j

### 

* Install findspark **sudo pip install findspark**
* ****

### Install Spark and Hadoop

### wget http://archive.apache.org/dist/spark/spark-3.0.0/spark-3.0.0-bin-hadoop2.7.tgz

### Unzip the package sudo tar -zxvf spark-3.0.0-bin-hadoop2.7.tgz

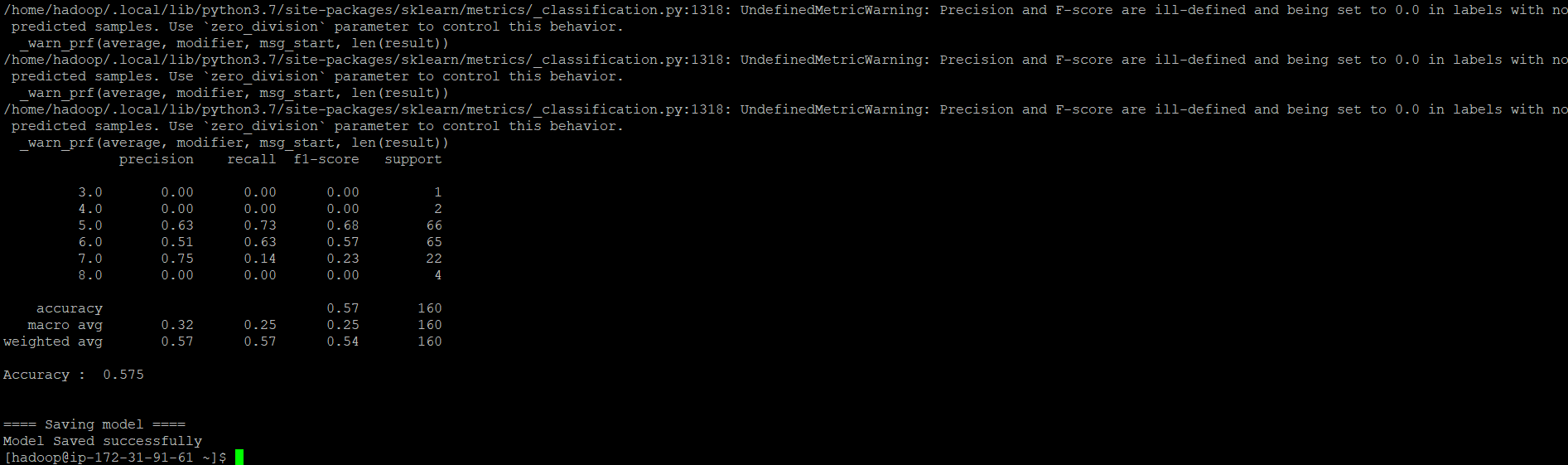
### Install pandas pip install pandas

### 

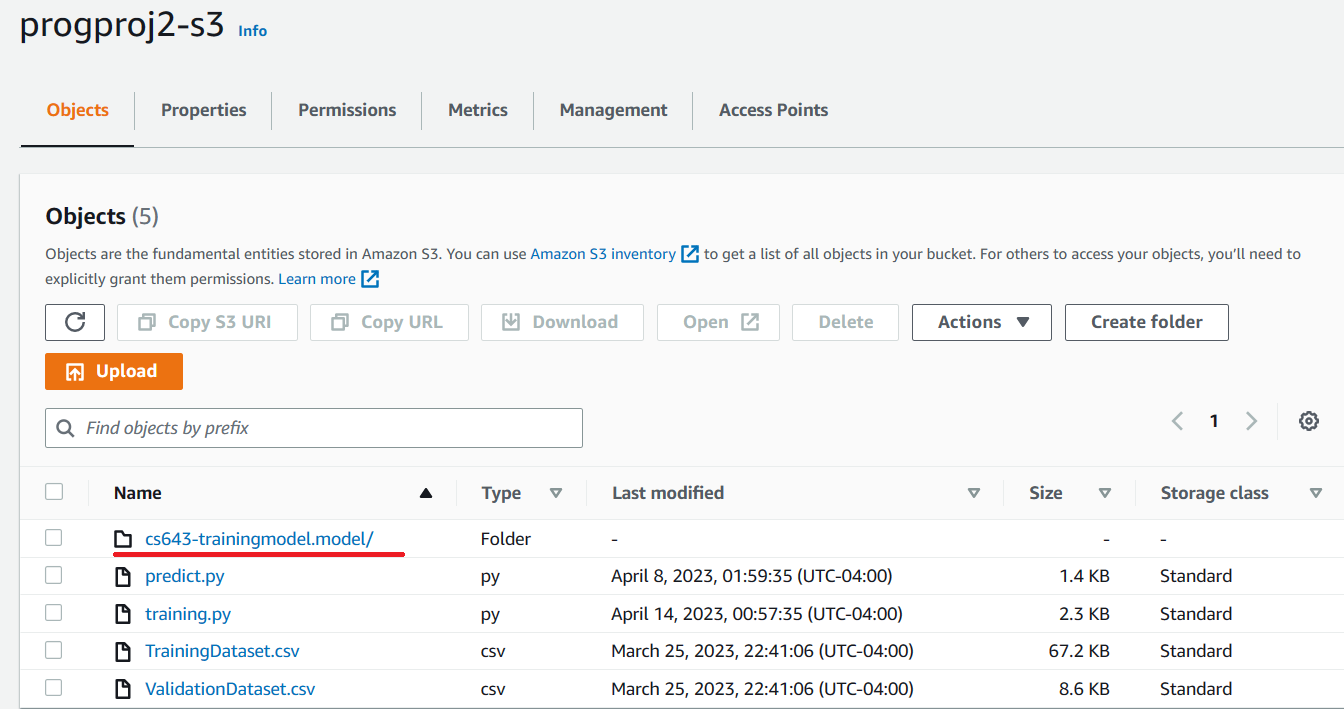
### Install scikit-learn pip install scikit-learn

### 

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

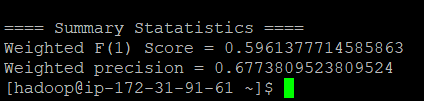
**** ****

**The model is saved in S3**

****

* Running the validation model in EC2

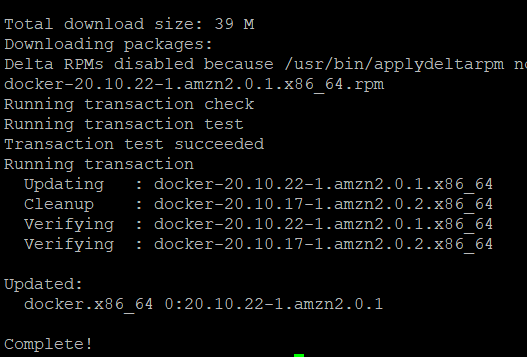
**spark-submit --packages org.apache.hadoop:hadoop-aws:2.7.7 s3://progproj2-s3/predict.py**



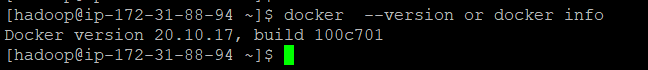
EC2 Instance with docker

First step Installation

* 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**



### Second Step 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 curl https://bootstrap.pypa.io/pip/2.7/get-pip.py -o get-pip.py \

&& python get-pip.py \

&& pip install pyspark \

&& mkdir ~/ MyDockerImages

COPY predicting.py ~/ MyDockerImages /

COPY training.py ~/ MyDockerImages /

COPY TrainingDataset.csv ~/ MyDockerImages /

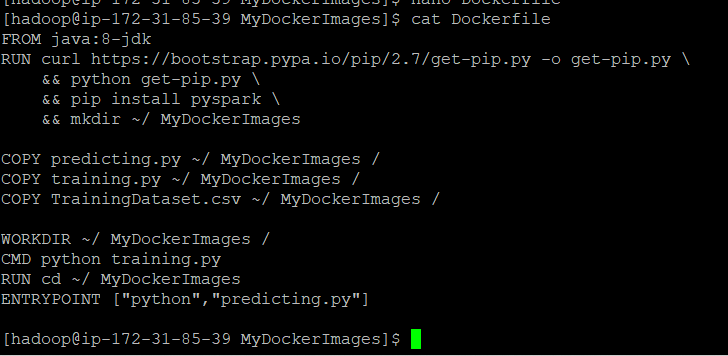
WORKDIR ~/ MyDockerImages /

CMD python training.py

RUN cd ~/cs643

ENTRYPOINT ["python","predicting.py"]

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

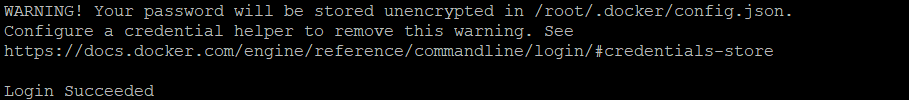


* Name the image as pproj2 **sudo docker build . -f Dockerfile -t cs643-pproj2**
* **run this image sudo docker run –rm –it cs643-pproj2**

**sudo docker images**



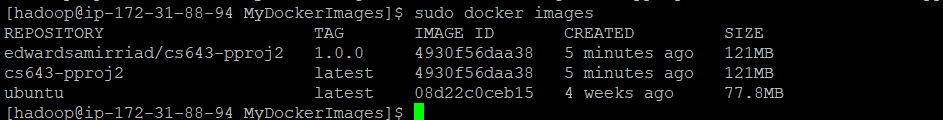
* push the image into the dockerhub **sudo docker login**



* give a tag to image so that version your image to multiple images.

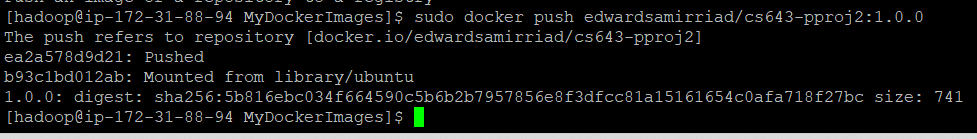
**sudo docker tag cs643-pproj2 edwardsamirriad/cs643-pproj2:1.0.0**

then **sudo docker images**



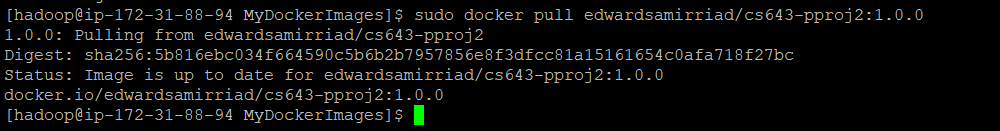
* pushing your Dockerimage back to Ec2

**sudo docker push edwardsamirriad/cs643-pproj2:1.0.0**



* pulling your Dockerimage back to Ec2

**sudo docker pull edwardsamirriad/cs643-pproj2:1.0.0**



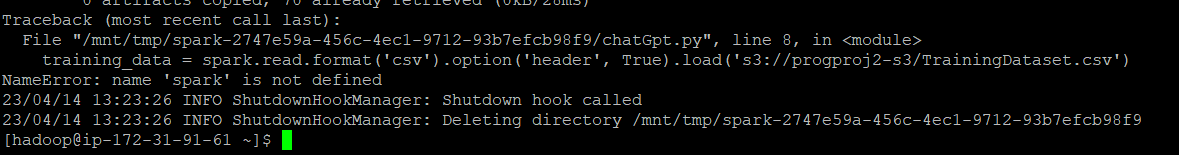
* Running my docker image **sudo docker run -t cs643-pproj2**

**Running chatGpt model on EC2**

**I generated a two codes from chatGPT under the names: chatGptTraining.py and chatGptValidation.py they both are uploaded on Github:** [EdwardRiad/CS643\_Programming\_Project\_2: NJIT Cloud Computing-CS 643 programming project 2 (github.com)](https://github.com/EdwardRiad/CS643_Programming_Project_2)

**spark-submit --packages org.apache.hadoop:hadoop-aws:2.7.7 s3://progproj2-s3/ChatGPTModel.py**

**When I ran the spark package it gave me some errors but I tried to solve them.**

****