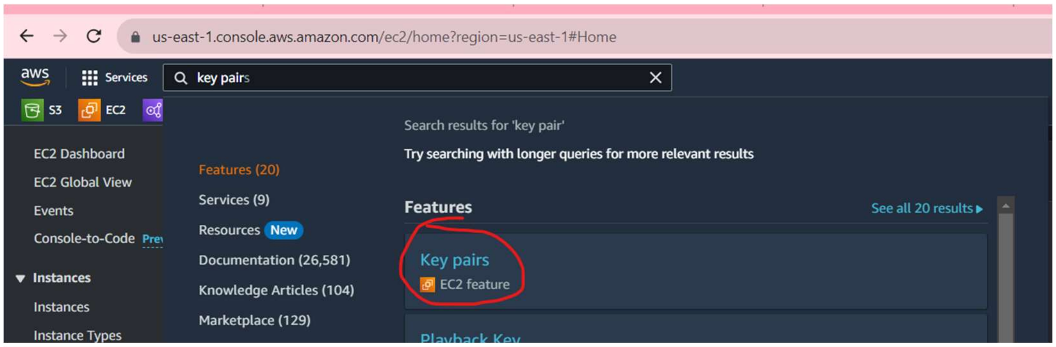
SAI KARTHIK KASUMURTHY (sk3374@njit.edu)

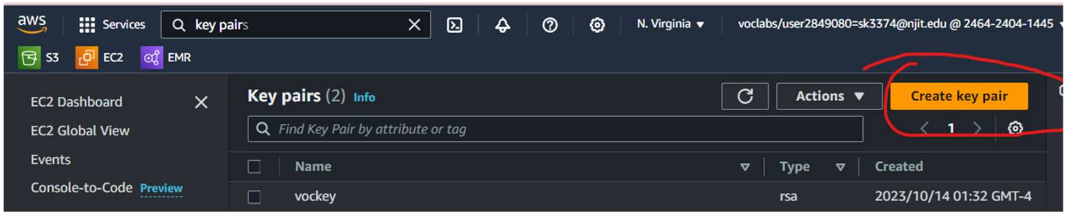
Programming Assignment 2

GitHub Link: https://github.com/karthik984/WinePredictionAnalysis
Docker Image Link: https://hub.docker.com/r/karthikkk999/wine-prediction

Step-by-step process on how to set-up the cloud environment, run the model training and the application prediction:

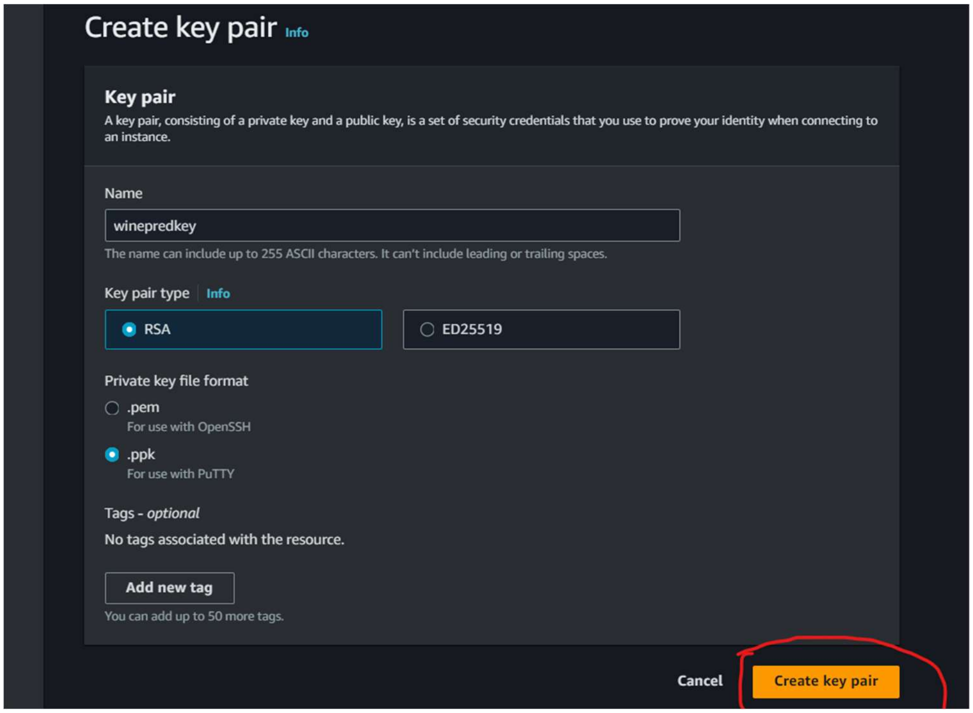
1. **Create a Key Pair:**
   a. Go to EC2 feature > "Key Pairs" and click on "Create key pair" as





□	Name	Type	Created
□	vockey	rsa	2023/10/14 01:32 GMT-4

   b. Give it a name and select ".pem" if you have MAC OS or ".ppk" if you have Windows and click on "Create Key pair"



A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name
winepredkey
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type Info
● RSA
○ ED25519

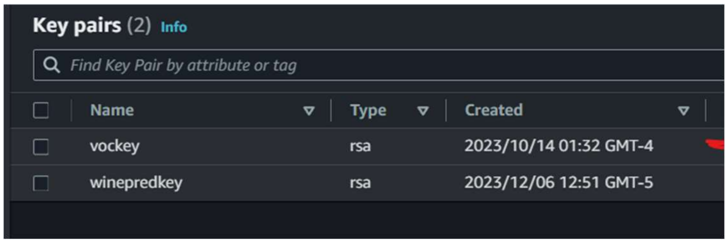
Private key file format
○ .pem
For use with OpenSSH
● .ppk
For use with PuTTY

Tags - *optional*
No tags associated with the resource.

You can add up to 50 more tags.

c. This will also download the ".ppk" or ".pem" file to your local machine. Make sure you save it. You will require it later.

d. Once Created it should appear in your key pairs list:



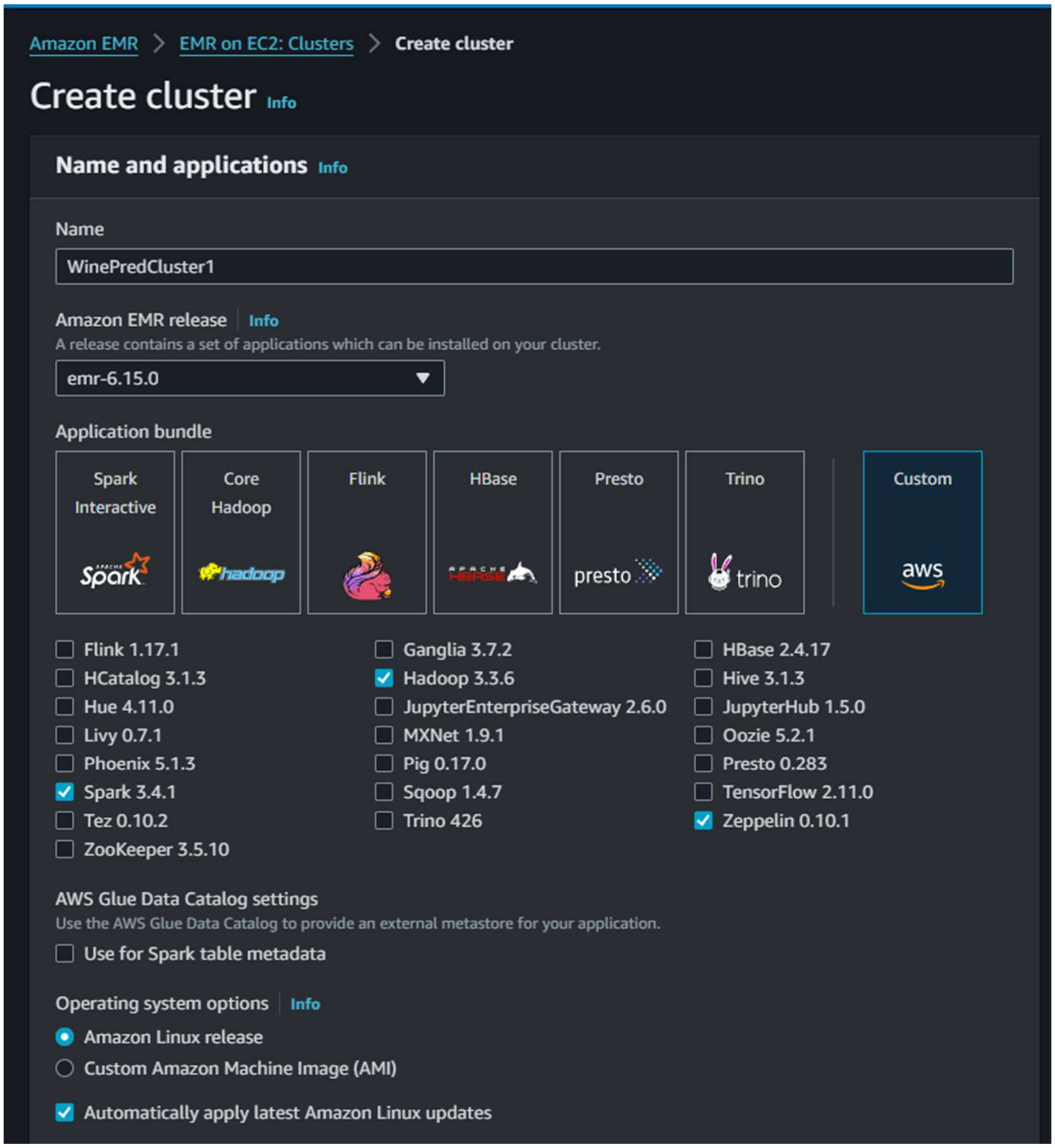
Key pairs (2) Info

Find Key Pair by attribute or tag

□	Name	Type	Created
□	vockey	rsa	2023/10/14 01:32 GMT-4
□	winepredkey	rsa	2023/12/06 12:51 GMT-5

2. **Create EMR Cluster:**

a. Go to "Amazon EMR > EMR on EC: Clusters > Create cluster" and give it a name and select the latest emr version. Also check the Hadoop, Spark , Zeppelin which are required by our code to train the model



Amazon EMR > EMR on EC2: Clusters > Create cluster

Create cluster Info

Name and applications Info

Name

WinePredCluster1

Amazon EMR release Info

A release contains a set of applications which can be installed on your cluster.

emr-6.15.0 ▼

Application bundle

□ Flink 1.17.1
□ HCatalog 3.1.3
□ Hue 4.11.0
□ Livy 0.7.1
□ Phoenix 5.1.3
☑ Spark 3.4.1
□ Tez 0.10.2
□ ZooKeeper 3.5.10
□ Ganglia 3.7.2
☑ Hadoop 3.3.6
□ JupyterEnterpriseGateway 2.6.0
□ MXNet 1.9.1
□ Pig 0.17.0
□ Sqoop 1.4.7
□ Trino 426
□ HBase 2.4.17
□ Hive 3.1.3
□ JupyterHub 1.5.0
□ Oozie 5.2.1
□ Presto 0.283
□ TensorFlow 2.11.0
☑ Zeppelin 0.10.1

AWS Glue Data Catalog settings

Use the AWS Glue Data Catalog to provide an external metastore for your application.

□ Use for Spark table metadata

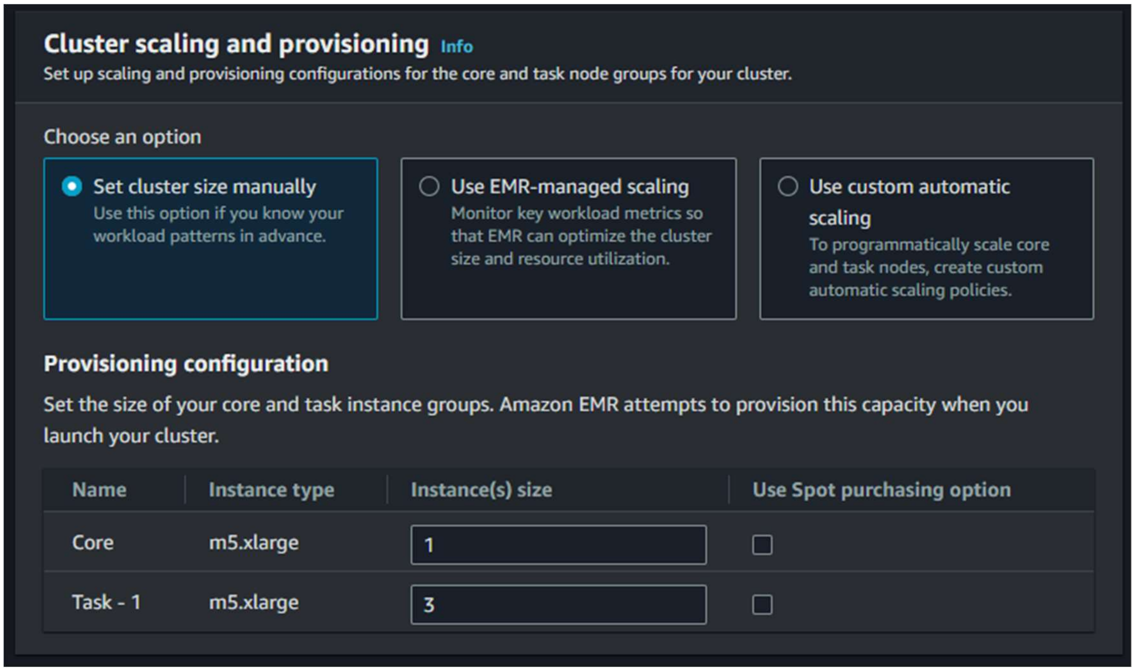
Operating system options Info

● Amazon Linux release
○ Custom Amazon Machine Image (AMI)

☑ Automatically apply latest Amazon Linux updates

b. Keep Cluster Configuration as it is and scroll until "Cluster scaling and provisioning".

c. In “Cluster scaling and provisioning” select 1 Core Instance and 3 Task – 1 instances. (Total 4)



Cluster scaling and provisioning Info

Set up scaling and provisioning configurations for the core and task node groups for your cluster.

Choose an option

● Set cluster size manually
Use this option if you know your workload patterns in advance.

○ Use EMR-managed scaling
Monitor key workload metrics so that EMR can optimize the cluster size and resource utilization.

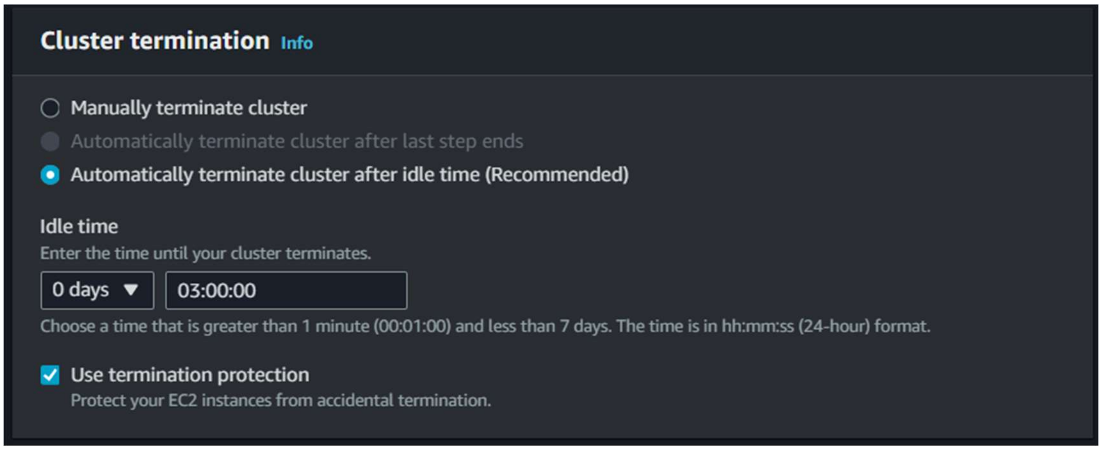
○ Use custom automatic scaling
To programmatically scale core and task nodes, create custom automatic scaling policies.

Provisioning configuration

Set the size of your core and task instance groups. Amazon EMR attempts to provision this capacity when you launch your cluster.

Name	Instance type	Instance(s) size	Use Spot purchasing option
Core	m5.xlarge	1	□
Task - 1	m5.xlarge	3	□

d. Select the Cluster termination as per your convenience. I had selected 3 hours



Cluster termination Info

○ Manually terminate cluster
○ Automatically terminate cluster after last step ends
● Automatically terminate cluster after idle time (Recommended)

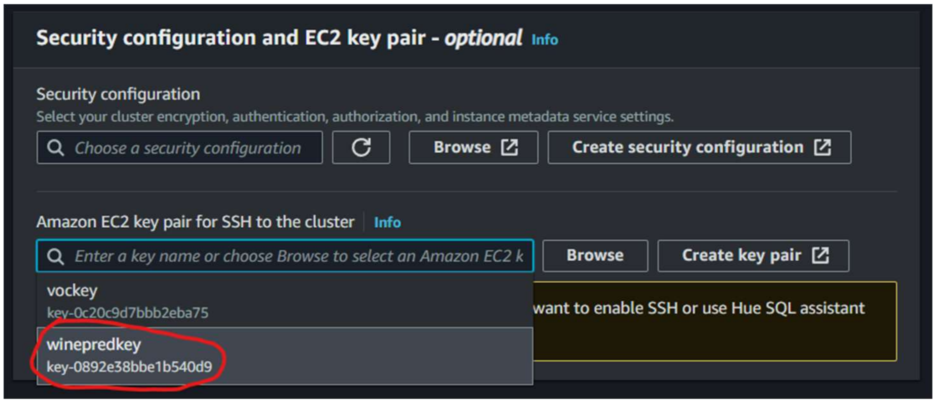
Idle time
Enter the time until your cluster terminates.

0 days ▼ 03:00:00

Choose a time that is greater than 1 minute (00:01:00) and less than 7 days. The time is in hh:mm:ss (24-hour) format.

☑ Use termination protection
Protect your EC2 instances from accidental termination.

e. Keep rest as it is and scroll down to “Security configuration and “EC2 key pair”. Select the key that you have created in Step 1.



Security configuration and EC2 key pair - *optional* Info

Security configuration
Select your cluster encryption, authentication, authorization, and instance metadata service settings.

Choose a security configuration
Browse
Create security configuration

Amazon EC2 key pair for SSH to the cluster Info

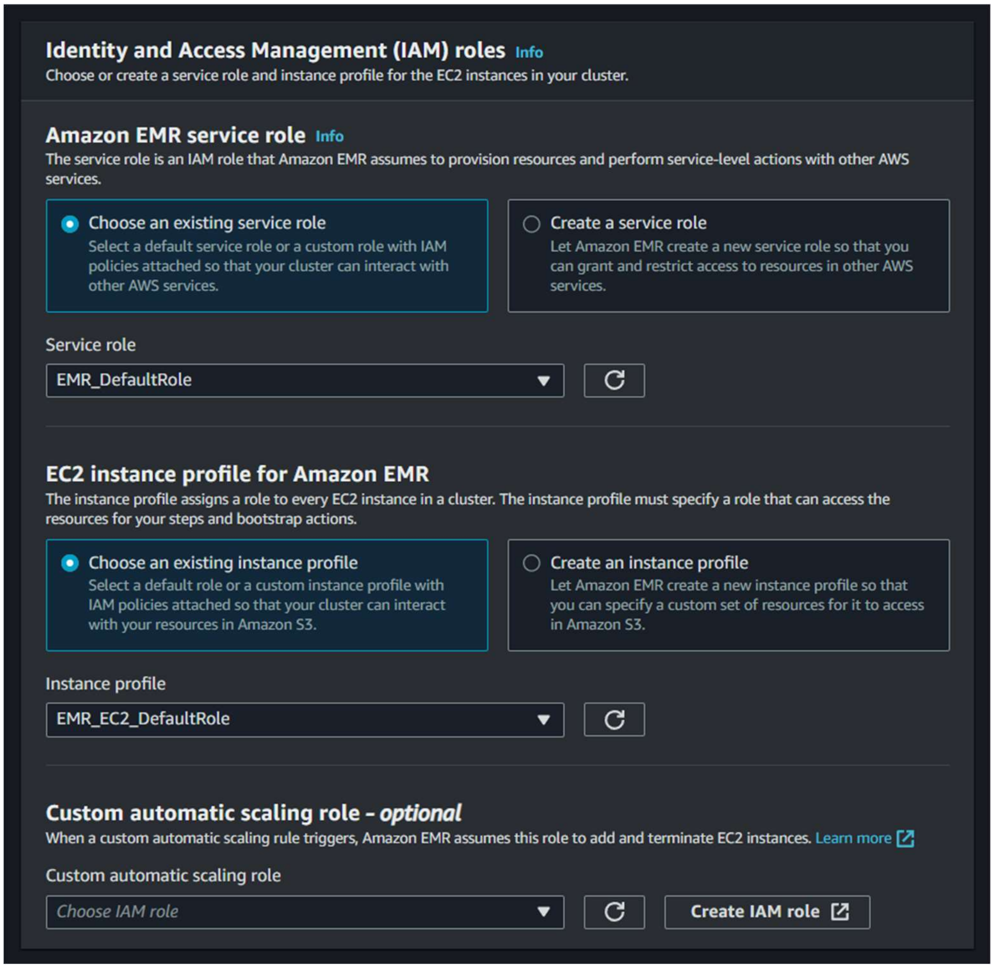
Enter a key name or choose Browse to select an Amazon EC2 k
Browse
Create key pair

vockey
key-0c20c9d7bbb2eba75

winepredkey
key-0892e38bbe1b540d9

want to enable SSH or use Hue SQL assistant

f. Select the EMR_DefaultRole as the service role and EMR_EC2_DefaultRole as the instance profile and the rest as it is.



Identity and Access Management (IAM) roles Info

Choose or create a service role and instance profile for the EC2 instances in your cluster.

Amazon EMR service role Info

The service role is an IAM role that Amazon EMR assumes to provision resources and perform service-level actions with other AWS services.

● Choose an existing service role
Select a default service role or a custom role with IAM policies attached so that your cluster can interact with other AWS services.

○ Create a service role
Let Amazon EMR create a new service role so that you can grant and restrict access to resources in other AWS services.

Service role: EMR_DefaultRole

EC2 instance profile for Amazon EMR

The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.

● Choose an existing instance profile
Select a default role or a custom instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.

○ Create an instance profile
Let Amazon EMR create a new instance profile so that you can specify a custom set of resources for it to access in Amazon S3.

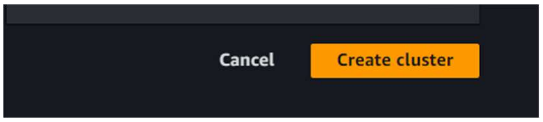
Instance profile: EMR_EC2_DefaultRole

Custom automatic scaling role - *optional*

When a custom automatic scaling rule triggers, Amazon EMR assumes this role to add and terminate EC2 instances. Learn more

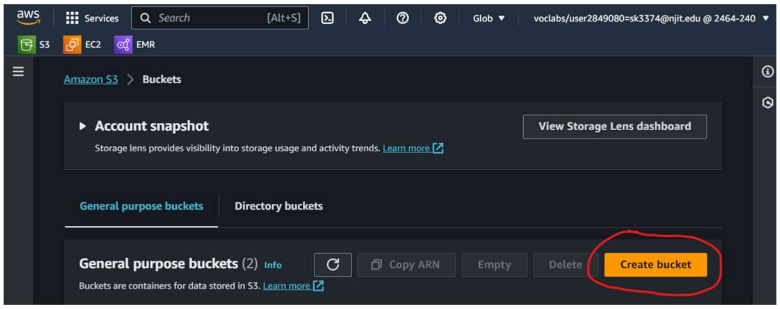
Custom automatic scaling role: *Choose IAM role*

g. Verify the summary and create the cluster



3. Creating S3 Bucket:

a. Go Amazon S3 > Buckets and create an S3 bucket with a name as shown below



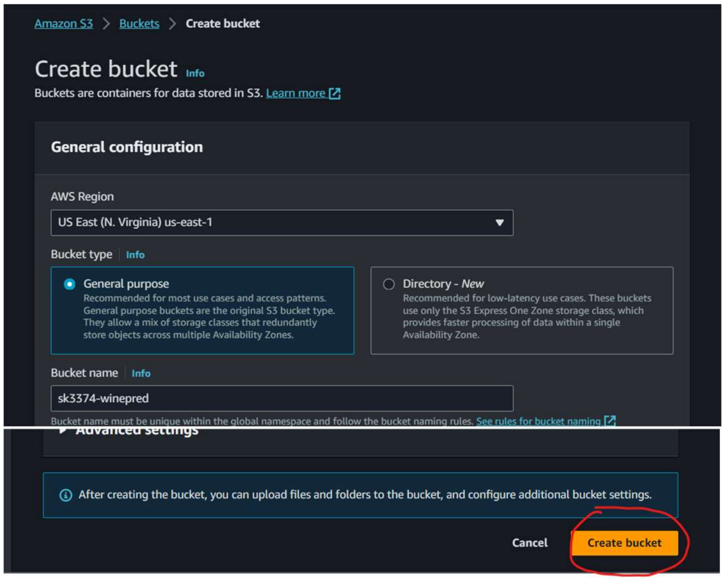
Account snapshot

Storage lens provides visibility into storage usage and activity trends. Learn more

General purpose buckets (2) Info

Buckets are containers for data stored in S3. Learn more

b. Select the AWS region, give a name to the S3 bucket, keep the rest settings as it is



Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

AWS Region: US East (N. Virginia) us-east-1

Bucket type Info

● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

○ Directory - *New*
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

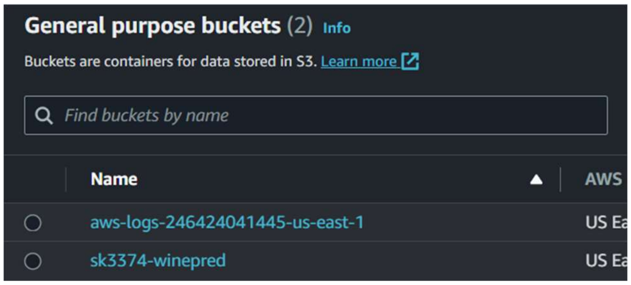
Bucket name Info: sk3374-winepred

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

▸ Advanced settings

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

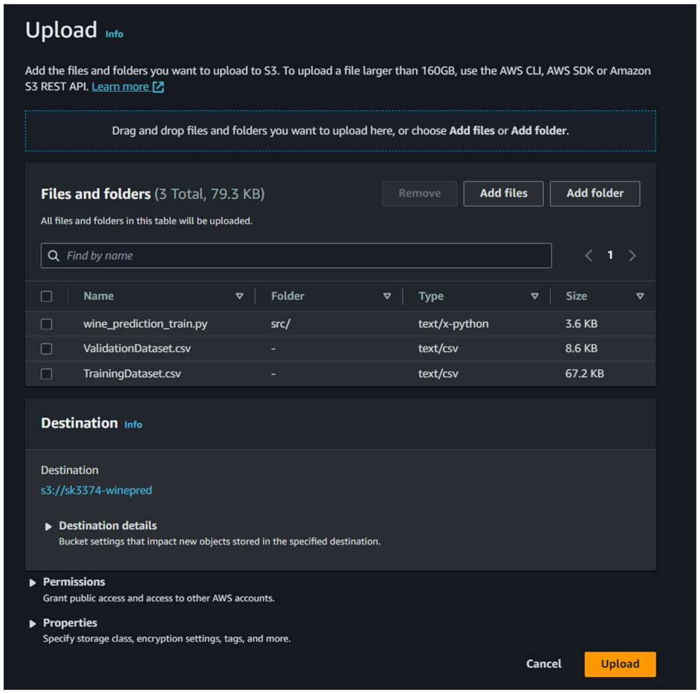
c. Once created it should appear in the list of s3 buckets


General purpose buckets (2) Info

Buckets are containers for data stored in S3. Learn more

Name	AWS
○ aws-logs-246424041445-us-east-1	US Ea
○ sk3374-winepred	US Ea

d. Upload the TrainingDataset.csv and ValidationDataset.csv and the src/wine_prediction_train.py train and


Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (3 Total, 79.3 KB)

All files and folders in this table will be uploaded.

Name	Folder	Type	Size
wine_prediction_train.py	src/	text/x-python	3.6 KB
ValidationDataset.csv	-	text/csv	8.6 KB
TrainingDataset.csv	-	text/csv	67.2 KB

Destination Info

Destination: s3://sk3374-winepred

▸ Destination details
Bucket settings that impact new objects stored in the specified destination.

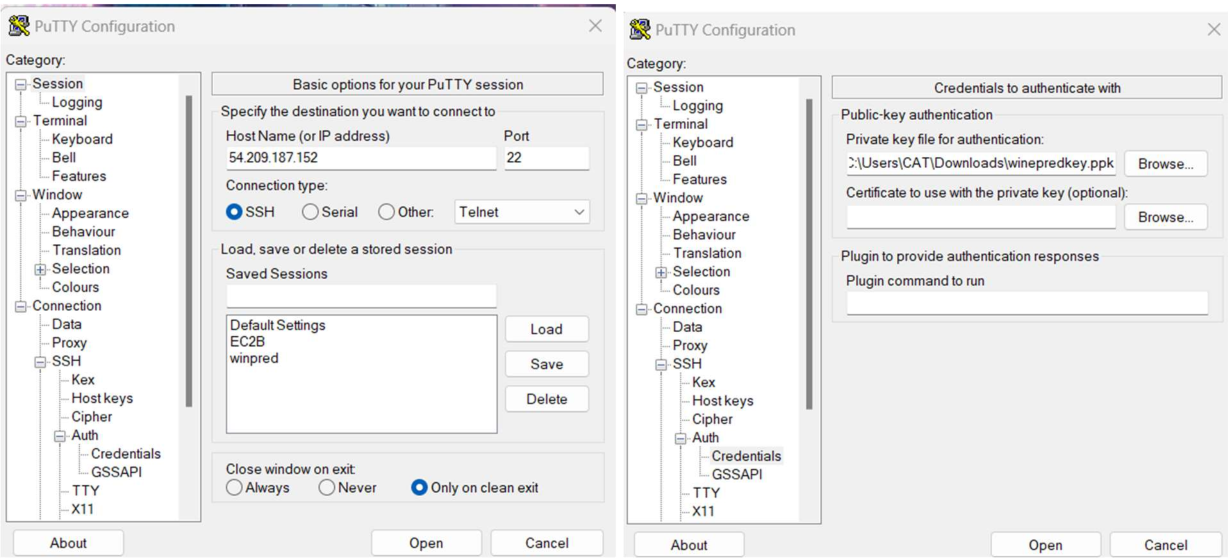
▸ Permissions
Grant public access and access to other AWS accounts.

▸ Properties
Specify storage class, encryption settings, tags, and more.

e. The folder “src/” would have the code that trains and creates the model.

4. Parallelly Training the Model in the EMR Cluster

a. Connect to the EMR EC2 Master node through putty using the “winepred.ppk” file from your local as shown below:



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 54.209.187.152

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings
EC2B
winpred

Close window on exit: ○ Always ○ Never ● Only on clean exit

Credentials to authenticate with

Public-key authentication

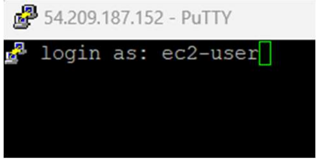
Private key file for authentication: :\Users\CAT\Downloads\winepredkey.ppk

Certificate to use with the private key (optional):

Plugin to provide authentication responses

Plugin command to run

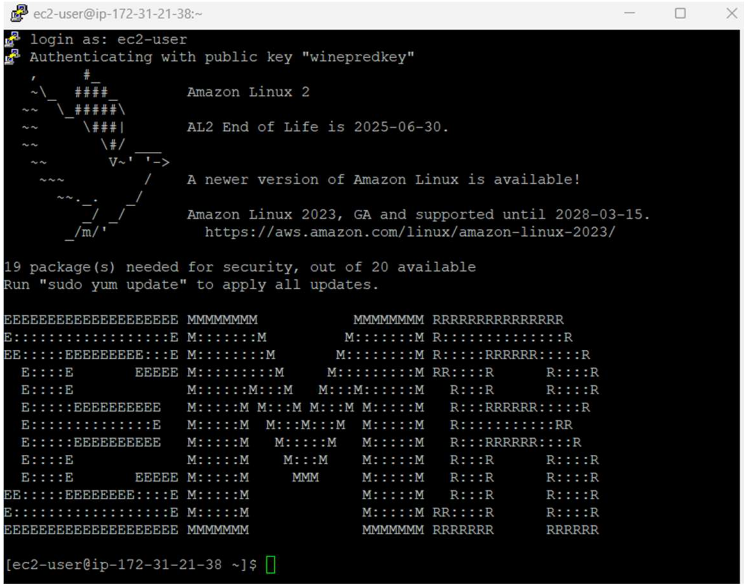
b. Use ec2-user as the login user



54.209.187.152 - PuTTY

```
login as: ec2-user
```

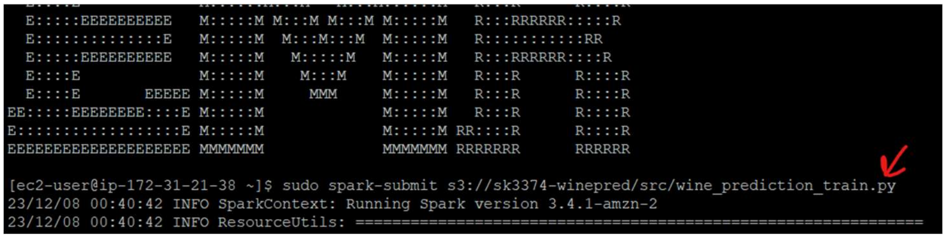
c. After successful you should see something like this



ec2-user@ip-172-31-21-38:~

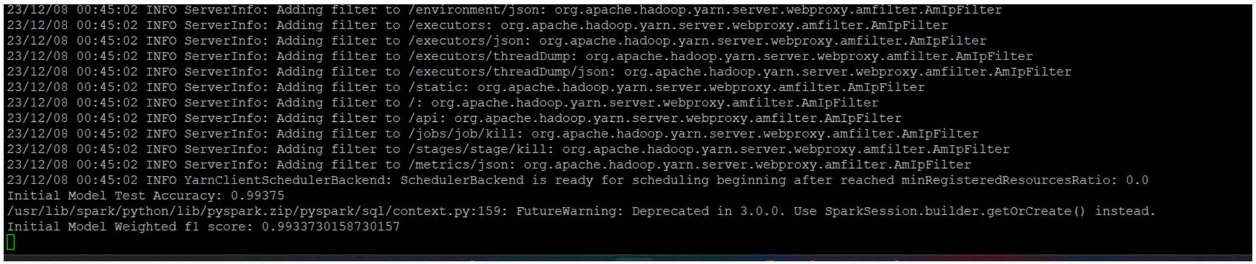
```
login as: ec2-user
Authenticating with public key "winepredkey"
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

19 package(s) needed for security, out of 20 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[ec2-user@ip-172-31-21-38 ~]$ 
```

d. Now run the command “sudo spark-submit s3://sk3374-winepred/src/wine_prediction_train.py” to start and train the model

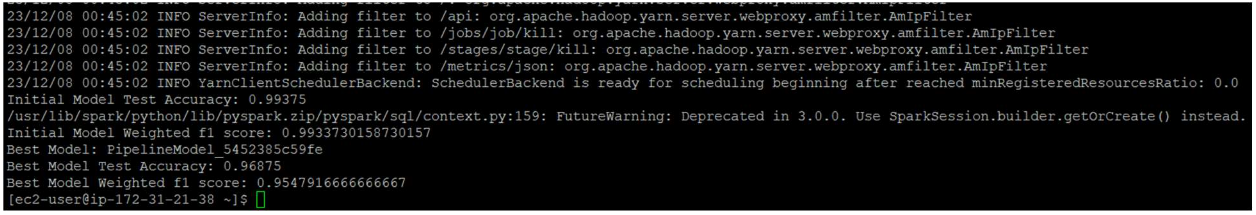


```
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M    R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M    R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M    R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M    R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M    R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M    R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[ec2-user@ip-172-31-21-38 ~]$ sudo spark-submit s3://sk3374-winepred/src/wine_prediction_train.py
23/12/08 00:40:42 INFO SparkContext: Running Spark version 3.4.1-amzn-2
23/12/08 00:40:42 INFO ResourceUtils: ==============================================================
```

e. You will see something like this it the code is running and the model is getting trained:

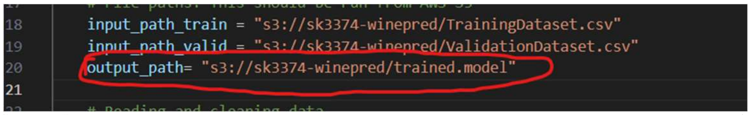


```
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /environment/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /executors: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /executors/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /executors/threadDump: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /executors/threadDump/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /static: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /api: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /jobs/job/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /stages/stage/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /metrics/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.0
Initial Model Test Accuracy: 0.99375
/usr/lib/spark/python/lib/pyspark.zip/pyspark/sql/context.py:159: FutureWarning: Deprecated in 3.0.0. Use SparkSession.builder.getOrCreate() instead.
Initial Model Weighted f1 score: 0.9933730158730157
```

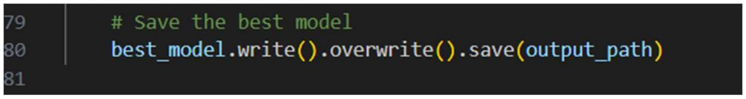


```
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /api: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /jobs/job/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /stages/stage/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO ServerInfo: Adding filter to /metrics/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
23/12/08 00:45:02 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.0
Initial Model Test Accuracy: 0.99375
/usr/lib/spark/python/lib/pyspark.zip/pyspark/sql/context.py:159: FutureWarning: Deprecated in 3.0.0. Use SparkSession.builder.getOrCreate() instead.
Initial Model Weighted f1 score: 0.9933730158730157
Best Model: PipelineModel_5452385c59fe
Best Model Test Accuracy: 0.96875
Best Model Weighted f1 score: 0.9547916666666667
[ec2-user@ip-172-31-21-38 ~]$
```

f. Once the code completed executed it should create a model and store in in the AWS S3 Bucket that you had created “s3://sk3374-winepred” earlier as per the login in the code:

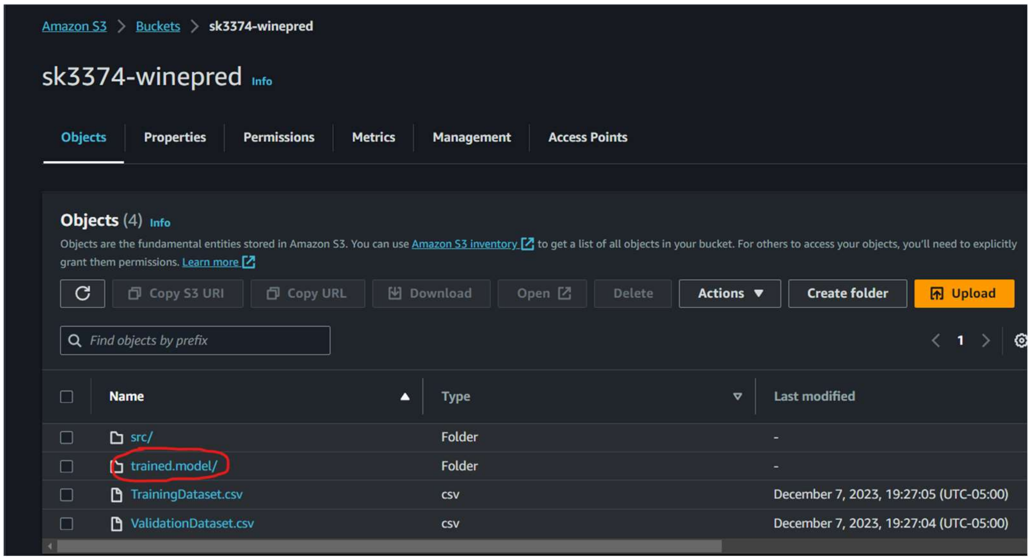


```
    input_path_train = "s3://sk3374-winepred/TrainingDataset.csv"
    input_path_valid = "s3://sk3374-winepred/ValidationDataset.csv"
    output_path= "s3://sk3374-winepred/trained.model"
```



```
    # Save the best model
    best_model.write().overwrite().save(output_path)
```

g. As shown below, the model would be created in the s3 bucket in the folder “trained.model/”



Amazon S3 > Buckets > sk3374-winepred

sk3374-winepred Info

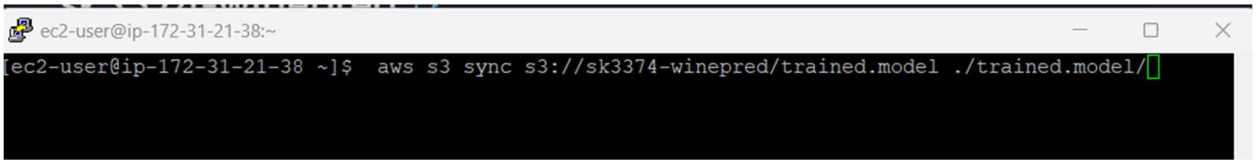
Objects (4) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified
src/	Folder	-
trained.model/	Folder	-
TrainingDataset.csv	csv	December 7, 2023, 19:27:05 (UTC-05:00)
ValidationDataset.csv	csv	December 7, 2023, 19:27:04 (UTC-05:00)

5. Testing the Trained Model on EC2 Instance and Locally

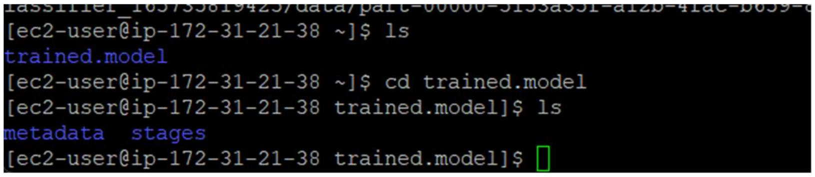
a. Now can you can get the model from your s3 bucket to your EC2 instance node using the below command "aws s3 sync s3://sk3374-winepred/trained.model ./trained.model/"



ec2-user@ip-172-31-21-38:~

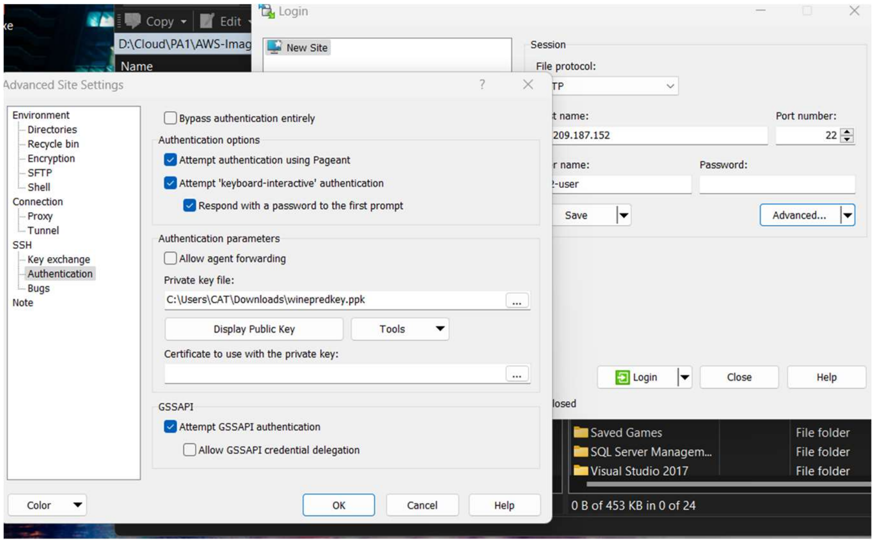
```
[ec2-user@ip-172-31-21-38 ~]$  aws s3 sync s3://sk3374-winepred/trained.model ./trained.model/
```

Now when you do ls you can see the downloaded model into your instance



```
lassifier_165735819423/data/part-00000-5153a551-a12b-41ac-b659-
[ec2-user@ip-172-31-21-38 ~]$ ls
trained.model
[ec2-user@ip-172-31-21-38 ~]$ cd trained.model
[ec2-user@ip-172-31-21-38 trained.model]$ ls
metadata  stages
[ec2-user@ip-172-31-21-38 trained.model]$
```

b. Now using WinSCP connect to the EC2 instances to test the trained model using a script. Create a testdata.csv using the previous training data or the validation data just to test the model



□ Bypass authentication entirely

Authentication options

☑ Attempt authentication using Pageant

☑ Attempt 'keyboard-interactive' authentication

☑ Respond with a password to the first prompt

Authentication parameters

□ Allow agent forwarding

Private key file: C:\Users\CAT\Downloads\winepredkey.ppk

Certificate to use with the private key:

GSSAPI

☑ Attempt GSSAPI authentication

□ Allow GSSAPI credential delegation

File protocol: TP

t name: 209.187.152

Port number: 22

r name: -user

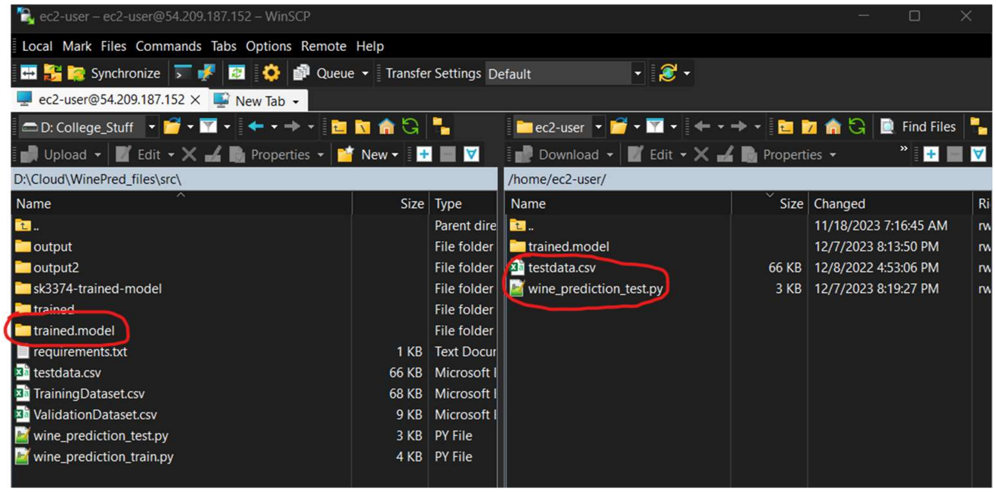
Password:

losed

Saved Games	File folder
SQL Server Managem...	File folder
Visual Studio 2017	File folder

0 B of 453 KB in 0 of 24

c. Now download the trained.model folder from the EC2 instance to your local and transfer the testdata.csv and wine_prediction_test.py (the script to test the accuracy of the trained model) to the EC2 instance



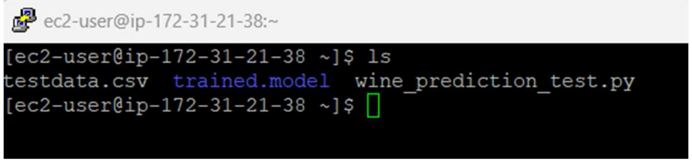
D:\Cloud\WinePred_files\src\

Name	Size	Type
..		Parent dire
output		File folder
output2		File folder
sk3374-trained-model		File folder
trained		File folder
trained.model		File folder
requirements.txt	1 KB	Text Docu
testdata.csv	66 KB	Microsoft
TrainingDataset.csv	68 KB	Microsoft
ValidationDataset.csv	9 KB	Microsoft
wine_prediction_test.py	3 KB	PY File
wine_prediction_train.py	4 KB	PY File

/home/ec2-user/

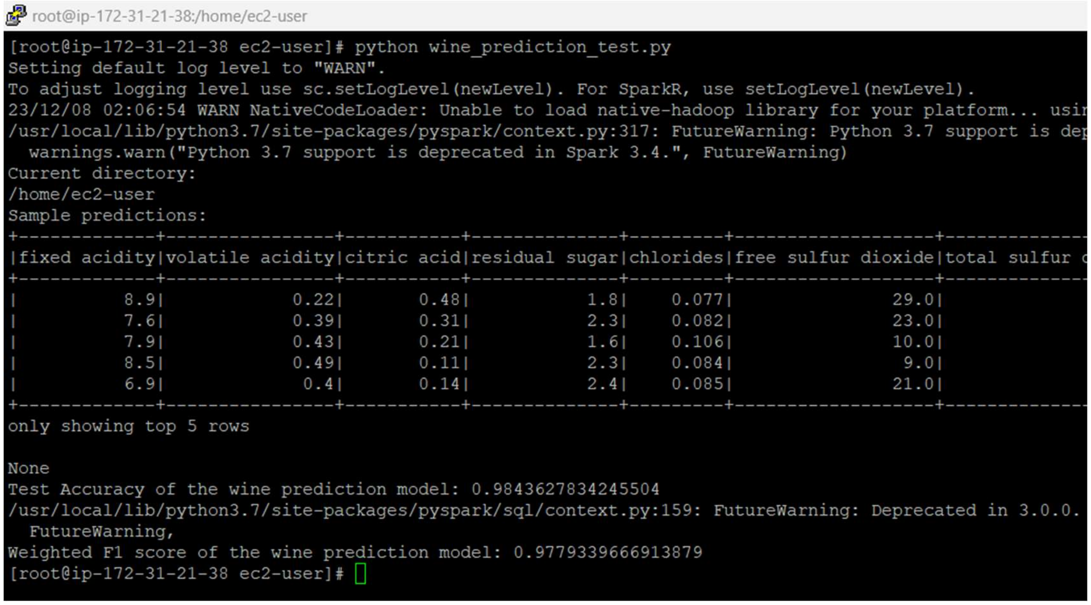
Name	Size	Changed	Ri
..		11/18/2023 7:16:45 AM	rw
trained.model		12/7/2023 8:13:50 PM	rw
testdata.csv	66 KB	12/8/2022 4:53:06 PM	rw
wine_prediction_test.py	3 KB	12/7/2023 8:19:27 PM	rw

d. One done make sure all the files needed for testing the trained model are on your EC2 instance.


ec2-user@ip-172-31-21-38:~

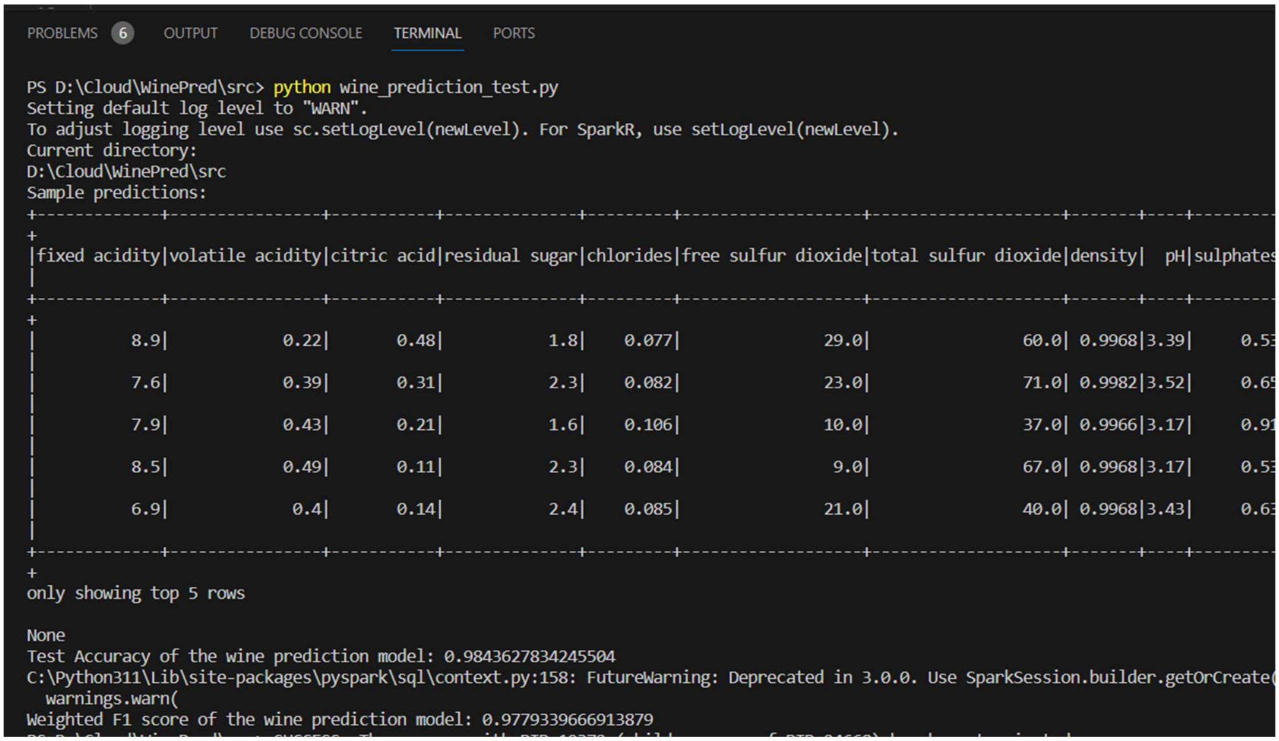
```
[ec2-user@ip-172-31-21-38 ~]$ ls
testdata.csv  trained.model  wine_prediction_test.py
[ec2-user@ip-172-31-21-38 ~]$ 
```

e. Now go ahead and run the wine_prediction_test.py using the below command
python wine_prediction_test.py


root@ip-172-31-21-38:/home/ec2-user

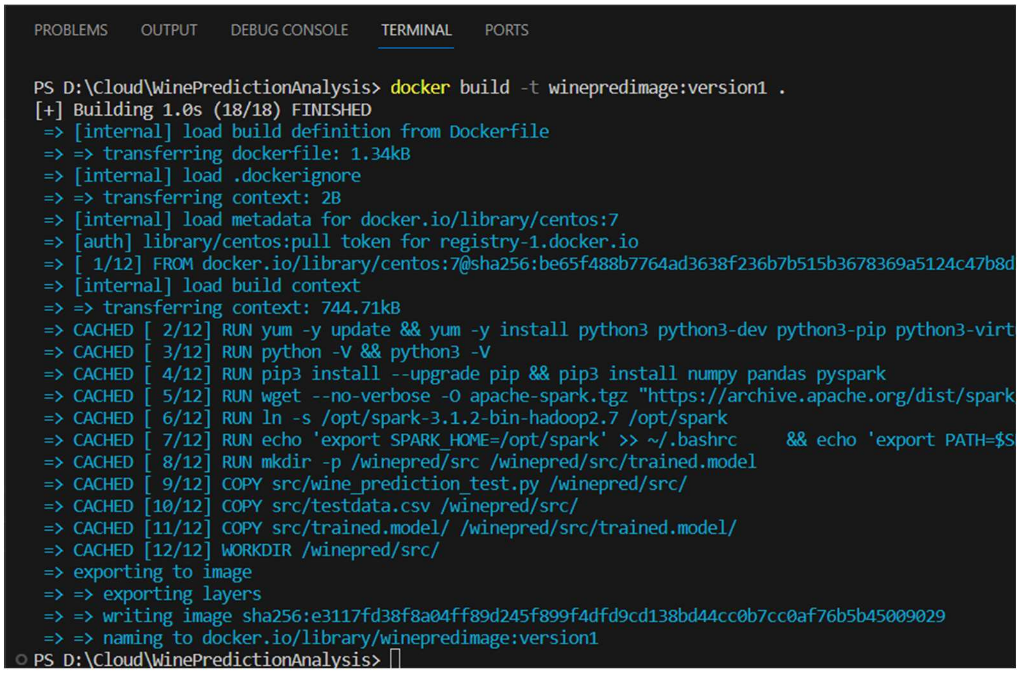
```
[root@ip-172-31-21-38 ec2-user]# python wine_prediction_test.py
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/12/08 02:06:54 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... usin
/usr/local/lib/python3.7/site-packages/pyspark/context.py:317: FutureWarning: Python 3.7 support is dep
  warnings.warn("Python 3.7 support is deprecated in Spark 3.4.", FutureWarning)
Current directory:
/home/ec2-user
Sample predictions:
+-------------+----------------+-----------+--------------+---------+-------------------+-------------
|fixed acidity|volatile acidity|citric acid|residual sugar|chlorides|free sulfur dioxide|total sulfur d
+-------------+----------------+-----------+--------------+---------+-------------------+-------------
|          8.9|            0.22|       0.48|           1.8|    0.077|               29.0|
|          7.6|            0.39|       0.31|           2.3|    0.082|               23.0|
|          7.9|            0.43|       0.21|           1.6|    0.106|               10.0|
|          8.5|            0.49|       0.11|           2.3|    0.084|                9.0|
|          6.9|             0.4|       0.14|           2.4|    0.085|               21.0|
+-------------+----------------+-----------+--------------+---------+-------------------+-------------
only showing top 5 rows

None
Test Accuracy of the wine prediction model: 0.9843627834245504
/usr/local/lib/python3.7/site-packages/pyspark/sql/context.py:159: FutureWarning: Deprecated in 3.0.0.
  FutureWarning,
Weighted F1 score of the wine prediction model: 0.9779339666913879
[root@ip-172-31-21-38 ec2-user]# 
```

f. Also you can run the wine_prediction_test.py in your local once you get the trained.model from the S3 as shown below



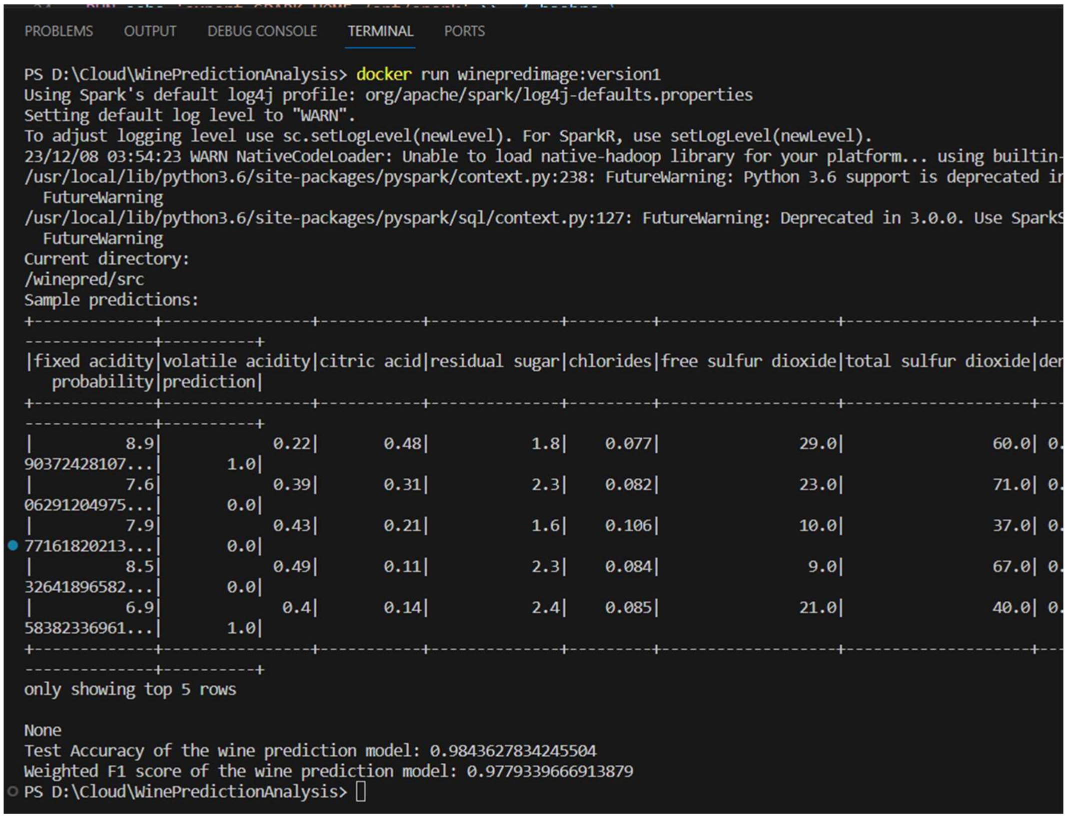
```
PS D:\Cloud\WinePred\src> python wine_prediction_test.py
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Current directory:
D:\Cloud\WinePred\src
Sample predictions:
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+-------+----+--------
+
|fixed acidity|volatile acidity|citric acid|residual sugar|chlorides|free sulfur dioxide|total sulfur dioxide|density|  pH|sulphates
|
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+-------+----+--------
+
|          8.9|            0.22|       0.48|           1.8|    0.077|               29.0|                60.0| 0.9968|3.39|     0.53
|
|          7.6|            0.39|       0.31|           2.3|    0.082|               23.0|                71.0| 0.9982|3.52|     0.65
|
|          7.9|            0.43|       0.21|           1.6|    0.106|               10.0|                37.0| 0.9966|3.17|     0.91
|
|          8.5|            0.49|       0.11|           2.3|    0.084|                9.0|                67.0| 0.9968|3.17|     0.53
|
|          6.9|             0.4|       0.14|           2.4|    0.085|               21.0|                40.0| 0.9968|3.43|     0.63
|
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+-------+----+--------
+
only showing top 5 rows

None
Test Accuracy of the wine prediction model: 0.9843627834245504
C:\Python311\Lib\site-packages\pyspark\sql\context.py:158: FutureWarning: Deprecated in 3.0.0. Use SparkSession.builder.getOrCreate(
  warnings.warn(
Weighted F1 score of the wine prediction model: 0.9779339666913879
```

6. **Create a Docker Image and Push it to DockerHub**
    a. Now create a Dockerfile with all the required configurations to run the wine_prediction_test.py in your repo
    b. Build an image using Dockerfile. Below is the command that is used create and build a docker image.
    Command: “docker build -t winepredimage:version1 .”



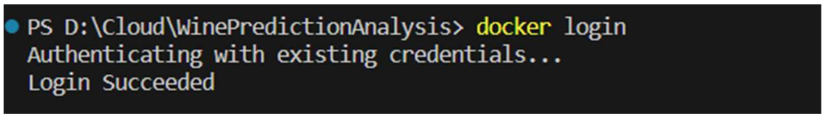
```
PS D:\Cloud\WinePredictionAnalysis> docker build -t winepredimage:version1 .
[+] Building 1.0s (18/18) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 1.34kB
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/centos:7
 => [auth] library/centos:pull token for registry-1.docker.io
 => [ 1/12] FROM docker.io/library/centos:7@sha256:be65f488b7764ad3638f236b7b515b3678369a5124c47b8d
 => [internal] load build context
 => => transferring context: 744.71kB
 => CACHED [ 2/12] RUN yum -y update && yum -y install python3 python3-dev python3-pip python3-virt
 => CACHED [ 3/12] RUN python -V && python3 -V
 => CACHED [ 4/12] RUN pip3 install --upgrade pip && pip3 install numpy pandas pyspark
 => CACHED [ 5/12] RUN wget --no-verbose -O apache-spark.tgz "https://archive.apache.org/dist/spark
 => CACHED [ 6/12] RUN ln -s /opt/spark-3.1.2-bin-hadoop2.7 /opt/spark
 => CACHED [ 7/12] RUN echo 'export SPARK_HOME=/opt/spark' >> ~/.bashrc     && echo 'export PATH=$S
 => CACHED [ 8/12] RUN mkdir -p /winepred/src /winepred/src/trained.model
 => CACHED [ 9/12] COPY src/wine_prediction_test.py /winepred/src/
 => CACHED [10/12] COPY src/testdata.csv /winepred/src/
 => CACHED [11/12] COPY src/trained.model/ /winepred/src/trained.model/
 => CACHED [12/12] WORKDIR /winepred/src/
 => exporting to image
 => => exporting layers
 => => writing image sha256:e3117fd38f8a04ff89d245f899f4dfd9cd138bd44cc0b7cc0af76b5b45009029
 => => naming to docker.io/library/winepredimage:version1
○ PS D:\Cloud\WinePredictionAnalysis> 
```

    c. Once built, test it on your local using the docker run command as shown below
    Command: “docker run winepredimage:version1” (This command is only for running on local)



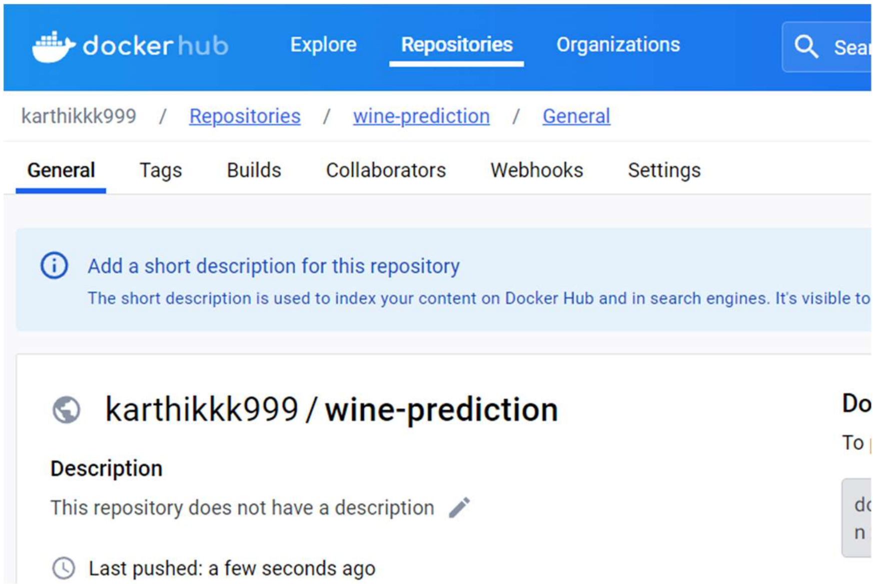
```
PS D:\Cloud\WinePredictionAnalysis> docker run winepredimage:version1
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/12/08 03:54:23 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-
/usr/local/lib/python3.6/site-packages/pyspark/context.py:238: FutureWarning: Python 3.6 support is deprecated in
  FutureWarning
/usr/local/lib/python3.6/site-packages/pyspark/sql/context.py:127: FutureWarning: Deprecated in 3.0.0. Use SparkS
  FutureWarning
Current directory:
/winepred/src
Sample predictions:
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+---
--------------+----------+
|fixed acidity|volatile acidity|citric acid|residual sugar|chlorides|free sulfur dioxide|total sulfur dioxide|den
   probability|prediction|
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+---
--------------+----------+
|          8.9|            0.22|       0.48|           1.8|    0.077|               29.0|                60.0| 0.
90372428107...|       1.0|
|          7.6|            0.39|       0.31|           2.3|    0.082|               23.0|                71.0| 0.
06291204975...|       0.0|
|          7.9|            0.43|       0.21|           1.6|    0.106|               10.0|                37.0| 0.
● 77161820213...|       0.0|
|          8.5|            0.49|       0.11|           2.3|    0.084|                9.0|                67.0| 0.
32641896582...|       0.0|
|          6.9|             0.4|       0.14|           2.4|    0.085|               21.0|                40.0| 0.
58382336961...|       1.0|
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+---
--------------+----------+
only showing top 5 rows

None
Test Accuracy of the wine prediction model: 0.9843627834245504
Weighted F1 score of the wine prediction model: 0.9779339666913879
○ PS D:\Cloud\WinePredictionAnalysis> 
```

d. Now use docker login to login to your DockerHub account. (I have already logged in and that's why it said Login Succeeded)



```
PS D:\Cloud\WinePredictionAnalysis> docker login
Authenticating with existing credentials...
Login Succeeded
```

e. Now go to DockerHub on a web browser and create a repository called wine-prediction



Add a short description for this repository

The short description is used to index your content on Docker Hub and in search engines. It's visible to

karthikkk999 / wine-prediction

Description

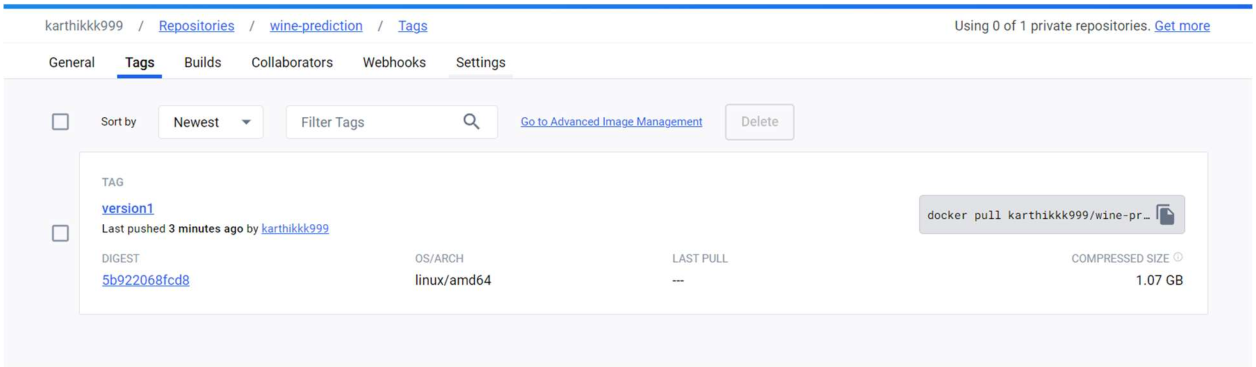
This repository does not have a description

Last pushed: a few seconds ago

f. Now tag the local image to your repository that you have created using the below command "docker tag winepredimage:version1 karthikkk999/wine-prediction:version1"

g. And push the docker image to the DockerHub using the command "docker push karthikkk999/wine-prediction:version1"

h. Once successfully pushed, You can see this on your DockerHub account.



karthikkk999 / Repositories / wine-prediction / Tags
General
Tags
Builds
Collaborators
Webhooks
Settings
Sort by
Newest
Filter Tags
Go to Advanced Image Management
Delete

Using 0 of 1 private repositories. Get more

TAG

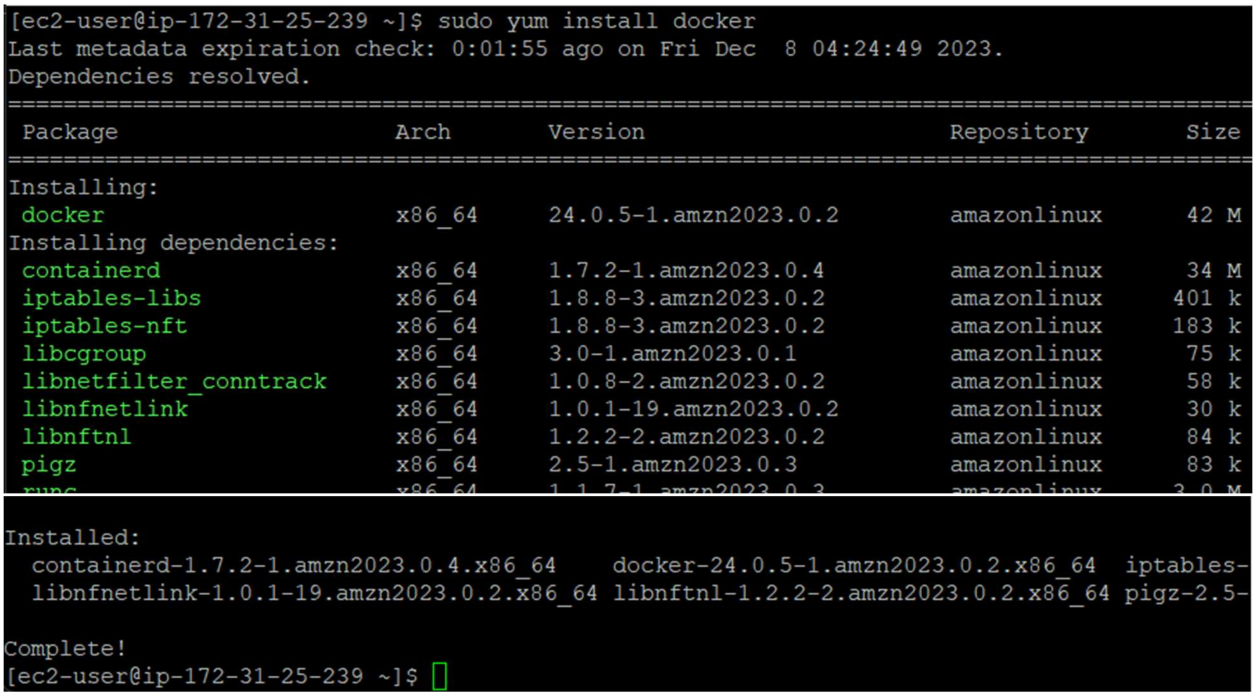
version1

Last pushed **3 minutes ago** by karthikkk999

`docker pull karthikkk999/wine-pr…`

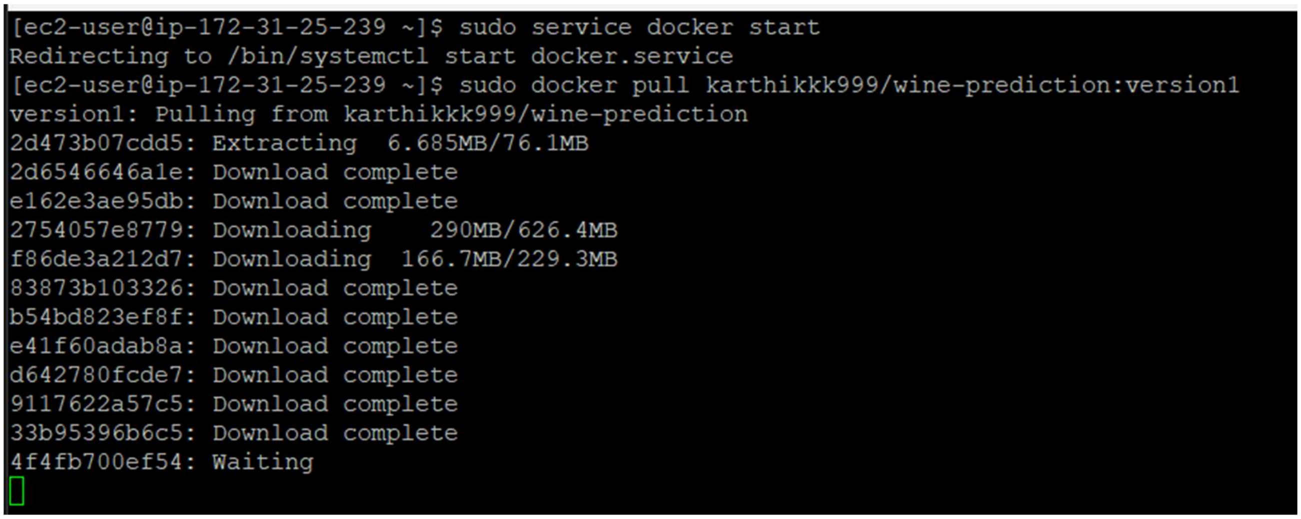
DIGEST	OS/ARCH	LAST PULL	COMPRESSED SIZE
5b922068fcd8	linux/amd64	---	1.07 GB

7. **Pulling the Docker Image and Running it on an EC2 Instance**
   a. SSH into any of the EC2 instances that you want to run the docker image and install docker
      “sudo yum install docker”



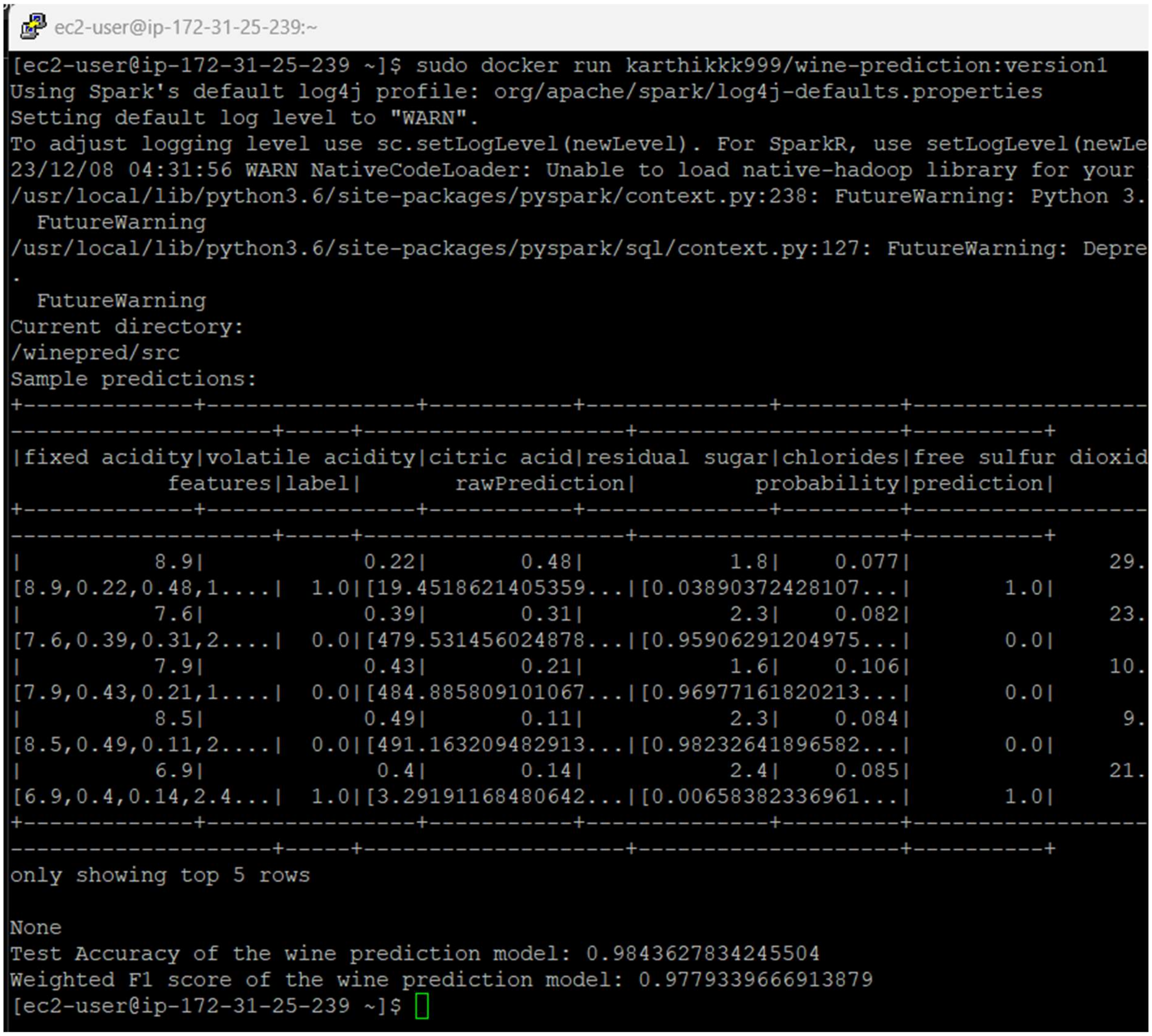
```
[ec2-user@ip-172-31-25-239 ~]$ sudo yum install docker
Last metadata expiration check: 0:01:55 ago on Fri Dec  8 04:24:49 2023.
Dependencies resolved.
==========================================================================================
 Package                    Arch       Version                     Repository        Size
==========================================================================================
Installing:
 docker                     x86_64     24.0.5-1.amzn2023.0.2       amazonlinux       42 M
Installing dependencies:
 containerd                 x86_64     1.7.2-1.amzn2023.0.4        amazonlinux       34 M
 iptables-libs              x86_64     1.8.8-3.amzn2023.0.2        amazonlinux      401 k
 iptables-nft               x86_64     1.8.8-3.amzn2023.0.2        amazonlinux      183 k
 libcgroup                  x86_64     3.0-1.amzn2023.0.1          amazonlinux       75 k
 libnetfilter_conntrack     x86_64     1.0.8-2.amzn2023.0.2        amazonlinux       58 k
 libnfnetlink               x86_64     1.0.1-19.amzn2023.0.2       amazonlinux       30 k
 libnftnl                   x86_64     1.2.2-2.amzn2023.0.2        amazonlinux       84 k
 pigz                       x86_64     2.5-1.amzn2023.0.3          amazonlinux       83 k
 runc                       x86_64     1.1.7-1.amzn2023.0.3        amazonlinux      3.0 M

Installed:
  containerd-1.7.2-1.amzn2023.0.4.x86_64    docker-24.0.5-1.amzn2023.0.2.x86_64  iptables-
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64 libnftnl-1.2.2-2.amzn2023.0.2.x86_64 pigz-2.5-

Complete!
[ec2-user@ip-172-31-25-239 ~]$ 
```

   b. Start docker and pull the wine-prediction image that you have built
      “sudo service docker start”
      “sudo docker pull karthikkk999/wine-prediction:version1”



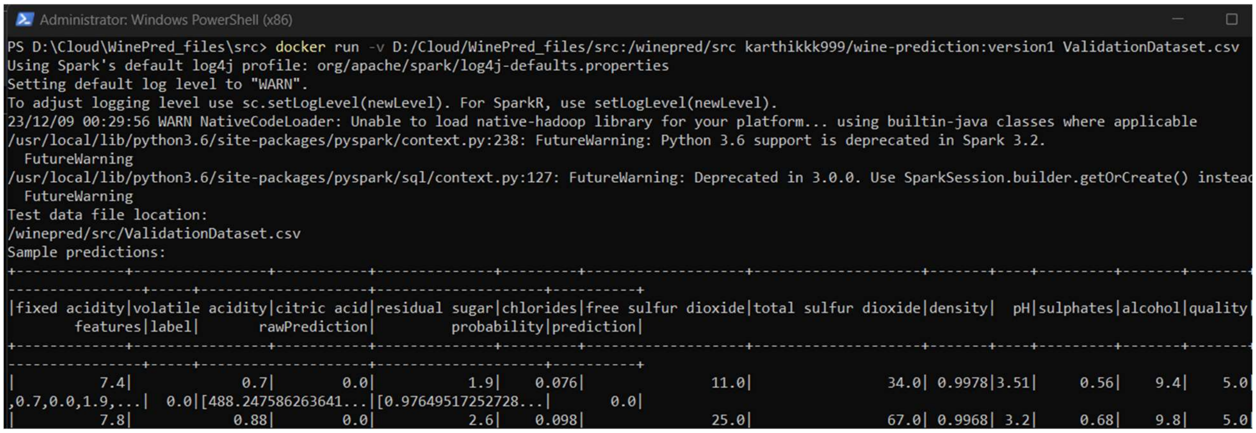
```
[ec2-user@ip-172-31-25-239 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-172-31-25-239 ~]$ sudo docker pull karthikkk999/wine-prediction:version1
version1: Pulling from karthikkk999/wine-prediction
2d473b07cdd5: Extracting  6.685MB/76.1MB
2d6546646a1e: Download complete
e162e3ae95db: Download complete
2754057e8779: Downloading    290MB/626.4MB
f86de3a212d7: Downloading  166.7MB/229.3MB
83873b103326: Download complete
b54bd823ef8f: Download complete
e41f60adab8a: Download complete
d642780fcde7: Download complete
9117622a57c5: Download complete
33b95396b6c5: Download complete
4f4fb700ef54: Waiting
```

c. Once the docker image is pulled run it using the docker run command
“sudo docker run karthikkk999/wine-prediction:version1”



```
[ec2-user@ip-172-31-25-239 ~]$ sudo docker run karthikkk999/wine-prediction:version1
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLe
23/12/08 04:31:56 WARN NativeCodeLoader: Unable to load native-hadoop library for your
/usr/local/lib/python3.6/site-packages/pyspark/context.py:238: FutureWarning: Python 3.
  FutureWarning
/usr/local/lib/python3.6/site-packages/pyspark/sql/context.py:127: FutureWarning: Depre
.
  FutureWarning
Current directory:
/winepred/src
Sample predictions:
+-------------+----------------+-----------+--------------+---------+------------------
--------------------+-----+--------------------+--------------------+----------+
|fixed acidity|volatile acidity|citric acid|residual sugar|chlorides|free sulfur dioxid
            features|label|       rawPrediction|         probability|prediction|
+-------------+----------------+-----------+--------------+---------+------------------
--------------------+-----+--------------------+--------------------+----------+
|          8.9|            0.22|       0.48|           1.8|    0.077|                29.
[8.9,0.22,0.48,1....|  1.0|[19.4518621405359...|[0.03890372428107...|       1.0|
|          7.6|            0.39|       0.31|           2.3|    0.082|                23.
[7.6,0.39,0.31,2....|  0.0|[479.531456024878...|[0.95906291204975...|       0.0|
|          7.9|            0.43|       0.21|           1.6|    0.106|                10.
[7.9,0.43,0.21,1....|  0.0|[484.885809101067...|[0.96977161820213...|       0.0|
|          8.5|            0.49|       0.11|           2.3|    0.084|                 9.
[8.5,0.49,0.11,2....|  0.0|[491.163209482913...|[0.98232641896582...|       0.0|
|          6.9|             0.4|       0.14|           2.4|    0.085|                21.
[6.9,0.4,0.14,2.4...|  1.0|[3.29191168480642...|[0.00658382336961...|       1.0|
+-------------+----------------+-----------+--------------+---------+------------------
--------------------+-----+--------------------+--------------------+----------+
only showing top 5 rows

None
Test Accuracy of the wine prediction model: 0.9843627834245504
Weighted F1 score of the wine prediction model: 0.9779339666913879
[ec2-user@ip-172-31-25-239 ~]$
```

d. If you want to specify a custom dataset file that you want, you can do that by persisting volume outside the container by using a flag -v as shown below
“docker run -v D:/Cloud/WinePred_files/src:/winepred/src karthikkk999/wine-prediction:version1 ValidationDataset.csv”



```
PS D:\Cloud\WinePred_files\src> docker run -v D:/Cloud/WinePred_files/src:/winepred/src karthikkk999/wine-prediction:version1 ValidationDataset.csv
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/12/09 00:29:56 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
/usr/local/lib/python3.6/site-packages/pyspark/context.py:238: FutureWarning: Python 3.6 support is deprecated in Spark 3.2.
  FutureWarning
/usr/local/lib/python3.6/site-packages/pyspark/sql/context.py:127: FutureWarning: Deprecated in 3.0.0. Use SparkSession.builder.getOrCreate() instead
  FutureWarning
Test data file location:
/winepred/src/ValidationDataset.csv
Sample predictions:
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+-------+----+---------+-------+-------+
----------------+-----+--------------------+--------------------+----------+
|fixed acidity|volatile acidity|citric acid|residual sugar|chlorides|free sulfur dioxide|total sulfur dioxide|density|  pH|sulphates|alcohol|quality|
        features|label|       rawPrediction|         probability|prediction|
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+-------+----+---------+-------+-------+
----------------+-----+--------------------+--------------------+----------+
|          7.4|             0.7|        0.0|           1.9|    0.076|               11.0|                34.0| 0.9978|3.51|     0.56|    9.4|    5.0|
,0.7,0.0,1.9,...|  0.0|[488.247586263641...|[0.97649517252728...|       0.0|
|          7.8|            0.88|        0.0|           2.6|    0.098|               25.0|                67.0| 0.9968| 3.2|     0.68|    9.8|    5.0|
```