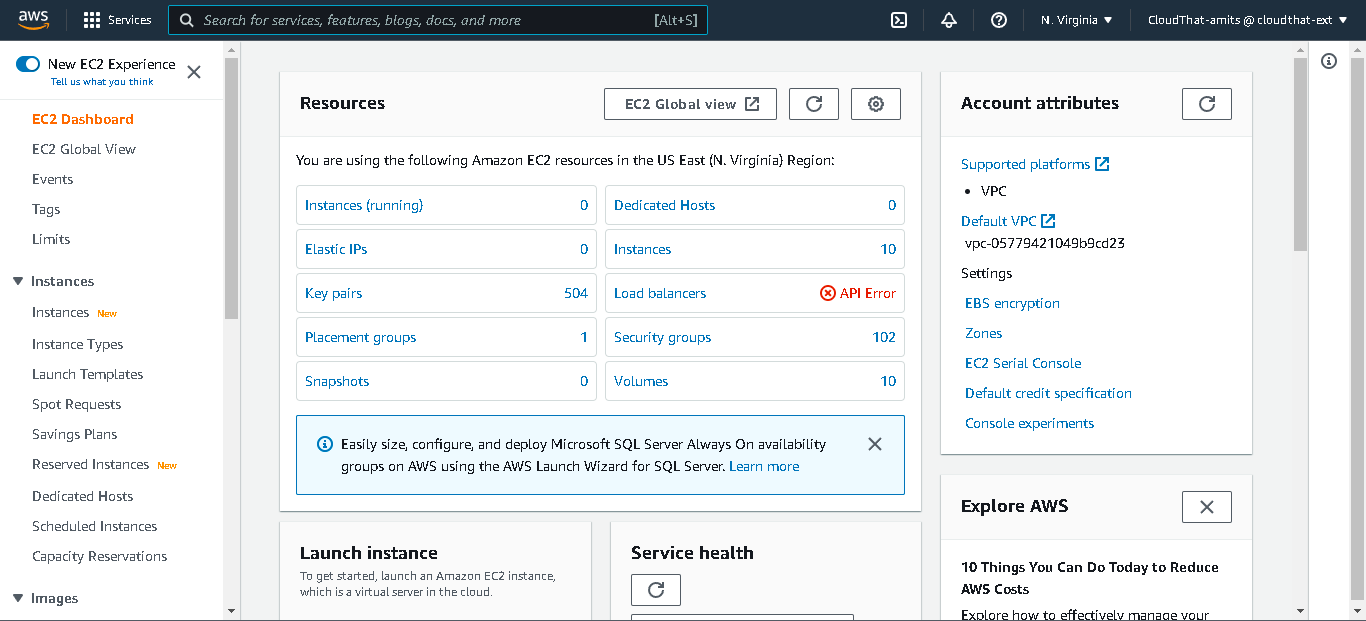
**Install Spark on Linux 20.04LTS**

Step 1: Use the URL <https://aws.amazon.com/console/> shared with you for Login to aws console. After successful login you will get AWS Management Console Page as follows

Graphical user interface, text, email, website

Description automatically generated

Step 2: Click on EC2 link in AWS services. It will redirect to you on EC2 Dashboard



Step 3: Click on Instances (Running Tab) from Resources as shown in above step

Step 4: It will take you to Instances Page as follows

Graphical user interface, application, Word

Description automatically generated

Step 5: It will take you to Launch an Instance page. Here you must provide following info

* Name and Tags: <yourfirstname>spark

Graphical user interface, text, application

Description automatically generated

* Application and OS images : Select Ubuntu OS from tiles and Ubuntu 20.04LTS AMI from dropdown as follows

Graphical user interface, text, application, email

Description automatically generated

* Instance type: select t2.xlarge from dropdown

Graphical user interface, application

Description automatically generated

* Key Pair (login)
  + Either select from list – if you already created keypair

Click on down arrow in keypair search bar

Select your keypar

Graphical user interface, application

Description automatically generated

2

1

* + Else you can create key pair and use it

Graphical user interface, text, application, email

Description automatically generated

Click on create new key pair with name <yourfirstname>spark and click on create KeyPair tab. It will download one file <yourfirstname>spark.pem.

Graphical user interface, text, application, email

Description automatically generated

2

1

After creation of new key pair its name appear in KeyPairName Textbox.

* Network Setting
  + Click on Edit Tab

Graphical user interface, text, application

Description automatically generated

* + It will open edit configuration page in same dialogue. Then click on radio button in front of *Select Existing group security group* and thenclick on down-arrow from *Common Security Group* textbox. Select “Launch-wizard-10”

Graphical user interface, application

Description automatically generated

3

2

1

* Configure Storage:- make value 30 [where value 8 is there]

Graphical user interface, text, application, email

Description automatically generated

2

1

Click on Launch Instance.

Step 6: Click on View all Instance tab

Graphical user interface, text, application, email, Teams

Description automatically generated

Step 7: It will redirect to Instances Page. Scroll down to locate your Instance. Under Instance check you can see state running but EC2 instance is not initialized and pass status check. You can see that under status check, So wait for 2-3min to initialize EC2 instance. Once Done Click on Instance id at right hand side of Instance Name.

Graphical user interface, application

Description automatically generated



Step 8: Note Down Public and Private address. Use copy icon from page to copy address

Graphical user interface, text, application, email

Description automatically generated

Step 9: Click on Connect tab

Graphical user interface, text, application, email

Description automatically generated

Step 10: It will take you to Connect to instance page. Then Click Connect [Default username is ubuntu don’t change it]

Graphical user interface, text, application

Description automatically generated

Step 11: After successful connect you will be getting Comand Line Interface like below

Text

Description automatically generated

Step 11: Configure Spark – Copy following command one after one then click on CLI and press ctrl+v

sudo apt update

sudo apt install software-properties-common

sudo apt install -y python3.8

sudo apt install -y pip

pip install virtualenv

sudo apt install python3-pip python3-dev

sudo -H pip3 install --upgrade pip

sudo -H pip3 install virtualenv

sudo apt install -y openjdk-8-jdk

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

wget https://downloads.apache.org/spark/spark-3.1.2/spark-3.1.2-bin-hadoop3.2.tgz

sudo mkdir /opt/spark

sudo tar -xf spark\*.tgz -C /opt/spark --strip-component 1

sudo chmod -R 777 /opt/spark

echo "export SPARK\_HOME=/opt/spark" >> ~/.bashrc

echo "export PATH=$PATH:$SPARK\_HOME/bin:$SPARK\_HOME/sbin" >> ~/.bashrc

echo "export PYSPARK\_PYTHON=/usr/bin/python3" >> ~/.bashrc

source ~/.bashrc

mkdir ~/my\_ML\_Projects

cd ~/my\_ML\_Projects

virtualenv ml\_project\_env

source ml\_project\_env/bin/activate

pip install jupyter

pip install pyspark

pip install findspark

pip install matplotlib

pip install seaborn

pip install sklearn

pip install pandas

pip install numpy

Jupyter notebook –ip=\*

Text

Description automatically generated

Open new browsing session

Enter <public ip>:8888 in address bar and hit enter

It will open Jupyter authentication section – get token from cli

Text

Description automatically generated

Paste it in token box and press ok

It will take you to jupyter file tree page

Graphical user interface, text, application, email

Description automatically generated

Click on New and select Terminal.

New browsing tab for Jupyter terminal is opened

Text

Description automatically generated

And enter command “git clone <https://github.com/siyad-CT/HPE_ML_WS>”

Once cloning done. Close terminal.

Now you can see folder named “HPE\_ML\_WS” is now listed

Graphical user interface, application, email

Description automatically generated

Step 12: Check Installation

Click on HPE\_ML\_WS

Click on Lab 2

Click on “[Word Count Example PySpark.ipynb](http://3.93.194.162:8888/notebooks/HPE_ML_WS/Lab%202/Word%20Count%20Example%20PySpark.ipynb)” – new notebook session is opened in browser

Graphical user interface, text, application, email

Description automatically generated

Click on Cell from Menu. Click on Run All.

It will execute all cells successfully with no error and list the word and its frequency in txt file.

This validates installation is successful.

**Open Jupyter Notebook**

Step 1: Login to Console

Step 2: Click on EC2. It will redirect to EC2 Dashboard. Click on Instances (Running)

Graphical user interface, application, email

Description automatically generated

Step 3: It shows you running instances. Scroll vertically and locate your instance with name *“<yourfirstname>spark”* click on Instance ID

Step 4: Click on Connect

Graphical user interface, text, application, email

Description automatically generated

Again, Click on Connect

Graphical user interface, text, application, email

Description automatically generated

It open CLI

Text

Description automatically generated

Enter and execute following command

cd ~/my\_ML\_Projects

virtualenv ml\_project\_env

source ml\_project\_env/bin/activate

Insert command

jupyter notebook –ip=\*

Open new browsing session

Enter <public ip>:8888 in address bar and hit enter

It will open Jupyter authentication section – get token from cli

Text

Description automatically generated

Paste it in token box and press ok

It will take you to jupyter file tree page

Graphical user interface, text, application, email

Description automatically generated

Click on New and select Terminal.

New browsing tab for Jupyter terminal is opened

Text

Description automatically generated

And enter command “git clone <https://github.com/siyad-CT/HPE_ML_WS>”

Once cloning done. Close terminal.

Now you can see folder named “HPE\_ML\_WS” is now listed

Graphical user interface, application, email

Description automatically generated

Step 12: Check Installation

Click on HPE\_ML\_WS

**Create Spark Cluster with Elephas**

Step 1. Connect to Instance <your first name>spark

Once cli is opened

Then enter following command

cd ~/my\_ML\_Projects

virtualenv ml\_project\_env

source ml\_project\_env/bin/activate

pip install elephas

*This instance will serve as master.*

Step 2: Launch EC2 instance with name <yourname>sparkw1

Then install spark on it using procedure mentioned in Appendix “Install Spark on Ubuntu20.04LTS”

*This will serve as your Worker 1*

In CLI

Enter command

Deactivate – (if virtualenv is active else skip)

pip install elephas

Step 3: Launch EC2 instance with name <yourname>sparkw1

Then install spark on it using procedure mentioned in Appendix “Install Spark on Ubuntu20.04LTS”

*This will serve as your Worker 2*

In CLI

Enter command

Deactivate – (if virtualenv is active else skip)

pip install elephas

Step 4: Launch EC2 instance with name <yourname>sparkw1

Then install spark on it using procedure mentioned in Appendix “Install Spark on Ubuntu20.04LTS”

*This will serve as your Worker 3*

In CLI

Enter command

Deactivate – (if virtualenv is active else skip)

pip install elephas

Step 5: Prepare table and note following

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Name** | **Role** | **Public IP** | **Private IP** |
| <yourname>spark | master | <public ip> | <private ip> |
| <yourname>sparkw1 | Worker #1 | <public ip> | <private ip> |
| <yourname>sparkw2 | Worker #2 | <public ip> | <private ip> |
| <yourname>sparkw3 | Worker #3 | <public ip> | <private ip> |

Note: put appropriate values at place holder

Step 6: On Master

Open CLI (Connect with EC2 Instance)

Enter Following Command one by one and execute

sudo apt install openssh-server openssh-client

cd ~/.ssh

ssh-keygen -t rsa -P ""

sudo service ssh restart

It will create one file id\_rsa.pub

Open it to view its content

cat id\_rsa.pub

Text

Description automatically generated

it will display key content. Copy it and paste it in notepad but make sure it should ne one line string only. if not appear as one line string after pasting in notepad, then make formatting to make it one line content.

Step 7: Connect to <your first name>sparkw1. After successful connection it will open CLI.

Enter following command one by one and execute

cd ~/.ssh

nano authorisedkey

Text

Description automatically generated

Take cursor on next line and copy key from notepad that you have maintained in step 6, paste using ctrl+v

Press ctrl+x and hit enter.

Step 8: Connect to <your first name>sparkw2. After successful connection it will open CLI.

Enter following command one by one and execute

cd ~/.ssh

nano authorisedkey

Text

Description automatically generated

Take cursor on next line and copy key from notepad that you have maintained in step 6, paste using ctrl+v

Press ctrl+x and hit enter.

Step 9: Connect to <your first name>sparkw3. After successful connection it will open CLI.

Enter following command one by one and execute

cd ~/.ssh

nano authorisedkey

Text

Description automatically generated

Take cursor on next line and copy key from notepad that you have maintained in step 6, paste using ctrl+v

Press ctrl+x and hit enter.

Step 10: check connection from master to each worker1

For worker 1

Enter command in cli and execute

ssh -i ~/.ssh/id\_rsa ubuntu@<public ip of worker 1>

it will give your prompt with ubuntu@ip-<private ip worker 1>

if yes connection is secured and successful

enter following command to get master prompt

exit

Step 11: repeat step 10, for check connection from master to each worker2

Step 12: repeat step 10, for check connection from master to each worker3

Step 13: On Master CLI

Enter Following command and execute one by one

cd /opt/spark/conf/nd

cp spark-env.sh.template spark-env.sh

nano spark-env.sh

It will open file in writable mode. Press down arrow to go to end of file. Then copy following lines and paste (please put private ip in placeholder before copy and paste)

export SPARK\_MASTER\_HOST=<private ip master>

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export PYSPARK\_PYTHON=python3

(It will add three new lines one after one. If not make formatting.)

Then press CTRL+x

Enter Y

Hit enter

It saves content and prompt of master appeared again.

Step 14: On Master CLI

Enter Following command and execute one by one

cp workers.template slaves

nano slaves

It will open file in writable mode. Press down arrow to go to end of file. Then copy following lines and paste (please put private ip in placeholder before copy and paste)

<private ip address worker 1>

<private ip address worker 2>

<private ip address worker 3>

(It will add three new lines one after one. If not make formatting.)

Then press CTRL+x

Enter Y

Hit enter

It saves content and prompt of master appeared again.

Step 15: On Master CLI

Enter Following command and execute one by one

cp spark-defaults.conf.template spark-defaults.conf

nano spark-defaults.conf

It will open file in writable mode. Press down arrow to go to end of file. Then copy following lines and paste

spark.driver.memory 5g

spark.driver.maxResultSize 2g

spark.executor.memory 5g

(It will add three new lines one after one. If not make formatting.)

Then press CTRL+x

Enter Y

Hit enter

It saves content and prompt of master appeared again.

Step 16: On Master CLI

Enter Following command and execute one by one

sh /opt/spark/sbin/start-all.sh

it will start spark cluster.

Step execute whenever you wish to stop spark Cluster

sh /opt/spark/sbin/stop-all.sh

**Install determined Cluster**

Step 1: Use the URL <https://aws.amazon.com/console/> shared with you for Login to aws console. After successful login you will get AWS Management Console Page as follows

Graphical user interface, text, email, website

Description automatically generated

Step 2: In Search bar enter keyword cloud formation, it will open navigation menu, select Cloud formation from there

1

A screenshot of a computer

Description automatically generated

2

Step 3: It will take you to cloud formation page. Click on create and then select “With new resources (standard)” option

Graphical user interface, application

Description automatically generated

Step 4: It will open create stack page. Enter url

<https://hp-training-resources.s3.amazonaws.com/determined.yaml>

Click Next

Graphical user interface, text, application

Description automatically generated

Step 5: Enter Stack name as <your first name>detcl and DBPassword as postgress as follows

Graphical user interface, text, application, email

Description automatically generated

Then select KeyPair - <your first name>spark

Graphical user interface, text, application, email

Description automatically generated

Step 6: it will take you to Configure Stack option. Click Next

Step 7: It will take you to Review page and display some warning. Ignore it

Scroll at end of page

Check the box infront of **I acknowledge that AWS CloudFormation might create IAM resources.**

**Click on Create Stack tab**

Step 8: it will display stack creation is in progress

Graphical user interface, text, application

Description automatically generated

Step 9: Once it completes (it will take 5-7min)

Graphical user interface, text, application, email

Description automatically generated

Then go to ec2 instance dashboard and you can see one instance with stack name is running

Step 10: Connect to EC2 Instance with name det-master-<yourfirst name>detcl

Wait till prompt appears

Then execute following command one by one

sudo apt-get update

sudo apt install python3-pip

pip install determined

sudo apt install -y git

git clone https://github.com/determined-ai/determined.git

cd determined/examples/computer\_vision/cifar\_tf\_keras

nano distributed.yml

It will open distributed.yml file , do following changes (instead 16 write 2)

slots per trial 16 - slots per trial 2

ctrl+x

hit enter

Prompt is appeared again

Enter following command to submit experiment run

det experiment create distributed.yaml .

Text

Description automatically generated

open browser new tab enter <public ip of det-master>:8080 and hit enter

It will open Determined Wb UI Asking for username and password

Put username as admin and keep password blank press sign in.

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Click on Experiment to see dashboard.

Graphical user interface, application

Description automatically generated

Yet Cluster is not up.

Once cluster is set up you can see Experiment in execution.

Lab-0 GUIDE

Name: To Implement Word count using spark on given txt file ‘words.txt’

Aim: We are going to familiarize on how to use the Jupyter notebook with PySpark with the help of word count example.

Input: text file

Process: Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 0/Word Count Example PySpark.ipynb

Step3. Importing Libraries

Step4. Create Spark Context

Step5. Read text file

Step6. Perform word counting operation

Step7. Stop Spark Context

Output: list of distinct word with there no of occurrence

Lab-1(a) GUIDE

Name: Learning python Pandas essentials

Aim: To learn how to use PANDAS to create, get, prepare, and analyse the data

Input: NA

Process: Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 1/PythonPandasEssentials.ipynb

Step3. Importing Libraries

Step4. Create data

Step5. Get Data

Step6. Prepare Data

Step7. Analyse data by finding max value

Output: learn about Pandas essentials

Lab-1(b) GUIDE

Name: Learning python Sklearn libraries

Aim: To learn how to use sklearn library to use prebuild model

Input: Iris dataset (Imported in program)

Process: Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 1/PythonsklearnEssentials.ipynb

Step3. Importing Libraries

Step4. Do pre-processing

Step5. Do Binarization

Step6. Analyse results

Output: learn about how to use sklearn prebuild model

Lab-2 GUIDE

Name: Feature Engineering using PySpark

Aim: To learn how to perform feature engineering using PySpark

Input: Titanic.csv

Process:

Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 2/ FeatureEngineeringusingPySpark.ipynb

Step3. Importing Libraries

Step4. Read Data

Step5. Do Basic feature Engineering i.e.check for null values and check what each column is describing about dataset

Step6. Perform Normalization and standardization

Step7. Analyse data by finding max value

Step 8. Convert python Data frame into spark Data frames

Step 9. Stop Spark Session

Output: learn about feature engineering and conversion to spark dataframes

Lab-3 GUIDE

Name: Encoding Categorical Features in Spark

Aim: To learn how to use Encoding Categorical Features in Spark

Input: Iris dataset (Imported in program)

Process: Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 1/Encoding Categorical Features in Spark.ipynb

Step3. Importing Libraries

Step4. Initiate Spark Session

Step5. Do pre-processing

Step6. Do Indexing using String Indexer

Step7. Apply one-hot encoding

Step8. Play with example

Step8. Stop Spark Session

Output: learn about how to use Encoding Categorical Features in Spark

Lab-4 GUIDE

Name: Classification using Logistic Regression

Aim: To learn how to perform Classification using Logistic Regression

Input: Diabetes.csv and Iris.csv

Process:

Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 4/ Train and Evaluate Classification Model.ipynb

Step3. Importing Libraries

Step4. Read Data

Step5. Do Basic feature Engineering i.e., check for null values and check what each column is describing about dataset

Step6. Split dataset into test and train dataset

Step7. Train Logistic model

Step 8. Test the model and evaluate using evaluation matrix

Step 9. Apply multiclass classification using Random Forest Classifier on Iris dataset

Step 10. Train and test the model performance

Step 11. Stop Spark Session

Output: learn how to do classification and Multiclass classification

Lab-5 GUIDE

Name: Train and Evaluate Regression Models

Aim: To learn how to perform training and evaluating Regression model

Input: Real estate.csv

Process:

Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 5/ Train and Evaluate Regression Models.ipynb

Step3. Importing Libraries

Step4. Read Data

Step5. Data Pre-processing and exploration

Step6. Split dataset into test and train dataset

Step7. Train Linear Regression model

Step 8. Test the model and evaluate using evaluation matrix

Step 9. Apply Random Forest Regression on Real estate dataset

Step 10. Train and test the model performance

Step 9. Stop Spark Session

Output: learn how to perform Regression on given dataset.

Lab-6 GUIDE

Name: Train and Evaluate Clustering Models

Aim: To learn how to perform training and evaluating Clustering model

Input: retail\_loyalty\_rfm.csv

Process:

Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 6/ Train and Evaluate Clustering Models.ipynb

Step3. Importing Libraries

Step4. Read Data

Step5. Vectorization and Scaler

Step6. Applying K-Means Clustering

Step7. Visualizing the Results and cluster

Step 8. 3D Visualization of clustering

Step 9. Applying Bisecting KMeans Clustering

Step 10. Visualize results

Step 11. Stop Spark Session

Output: learn how to perform Clustering on given dataset.

Lab-7 GUIDE

Name: Hyperparameter Tunning

Aim: To learn how to do hyperparameter tunning

Input: toyota.csv

Process:

Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 7/ Hyperparameter Tuning.ipynb

Step3. Importing Libraries

Step4. Read Data

Step5. Create spark session

Step6. Apply required feature engineering

Step7. Tune hyper parameter in Linear Regression Model by specifying the input column, output column, and the maximum number of iterations.

Step 8. Define the Parameter Grid for the Linear Regression Model

Step 9. Evaluate model performance

Step 10. Repeat hyperparameter tunning for random forest regressor

Step 11. Evaluate model performance

Step 12. Stop Spark Session

Output: learn how to perform hyperparameter tunning to enhance model performance

Lab-8 GUIDE

Name: Creating a Neural Network in Spark

Aim: To learn how to Create a Neural Network in Spark

Input: NA

Process:

Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 8/ Creating a Neural Network in Spark.ipynb

Step3. Importing Libraries

Step4. Create spark session

Step5. Create Dataframe

Step6. Apply required feature engineering

Step7.Visualize the dataset.

Step 8. Apply normalization

Step 9. Define sigmoid function

Step 10. Check and visualize sigmoid vs sigmoid deviate

Step 11. Evaluate model performance

Step 12. Stop Spark Session

Output: learn how to create Neural Network in spark.

Lab-9 GUIDE

Name: Creating a Convolutional Neural Network

Aim: To learn how to Create a Convolutional Neural Network

Input: MNIST dataset

Process:

Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 9/ Convolutional Neural Network.ipynb

Step3. Importing Libraries

Step4. Create spark session

Step5. Import dataset and split it into train and test dataset

Step6. Apply required feature engineering

Step7.Visualize the dataset.

Step 8. Train CNN model to classify images in MNIST dataset

Step 9. Test CNN model to classify images in MNIST dataset

Step 10. Evaluate model performance

Step 11. Stop Spark Session

Output: learn how to create Convolutional Neural Network in spark.

Lab-10 GUIDE

Name: Predicting Apple Stock Market Cost with LSTM

Aim: To learn how to Predict Apple Stock Market Cost with LSTM

Input: AAPL.csv

Process:

Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 10/ Predict Apple Stock Market Cost with LSTM.ipynb

Step3. Importing Libraries

Step4. Create spark session

Step5. Load dataset and split it into train and test dataset

Step6. Apply required feature engineering

Step7.Visualize the dataset.

Step 8. Train LSTM model

Step 9. Test LSTM model

Step 10. Evaluate model performance

Step 11. Stop Spark Session

Output: learn how to Predict Apple Stock Market Cost with LSTM

Lab-11 GUIDE

Name: Creating a Movie Recommendation Engine using deep Neural Network with Keras

Aim: To learn how to Create a Movie Recommendation Engine with Keras

Input: 'movies.csv', 'links.csv', 'README.txt', 'tags.csv', 'ratings.csv'

Process:

Step1. Open Jupyter Notebook

Step2. Open HP-AI-ML/Lab 11/ Creating a Movie Recommendation Engine with Keras.ipynb

Step3. Importing Libraries

Step4. Create spark session

Step5. Load dataset and split it into train and test dataset

Step6. Apply required feature engineering

Step7.Visualize the dataset.

Step 8. Train Keras model with 20 epochs

Step 9. Evaluate model performance

Step 10. Stop Spark Session

Output: learn how to Predict Creating a Movie Recommendation Engine with Keras

Lab-12 GUIDE

Name: Distributed Deep Neural Network with Spark

Aim: To implement distributed deep neural network using elephas on Spark - Executing MNIST classifier on Standalone Cluster

Input: const.yaml, distributed.yaml

Process:

Step1. Set up elephas Cluster

Step 2. Start cluster using master

Step 3. Open Jupyter Notebook on master

Step 4. Open HP-AI-ML/Lab 12/ MNIST\_using\_spark\_cluster.ipynb

Step 5. Initialise sparksession

Step 6. Import the necessary tensorflow and keras libraries

Step 7. Load dataset and split it into train and test dataset define get data function, get training function and validation of data

Step 8. Define Model and Train it

Step7. Evaluate model performance

Step 8. Stop Spark Session

Output: learn how to use Keras `Sequence`s and multiprocessing /multithreading for Keras models

Lab-13 GUIDE

Name: Distributed Deep Neural Network with Deterimened.ai

Aim: To implement distributed deep neural network using determined.ai cluster

Set-Up : Setup determined.ai cluster for distributed processing. Follow installation guide.