**Text

Description automatically generated with low confidenceDocument type**

CONFIDENTIAL | AUTHORIZED HPE PARTNER USE ONLY

# Section 1: Lab Environment Setup

## 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

Graphical user interface, text, email

Description automatically generated

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

1. Name and Tags: <yourfirstname>spark

Graphical user interface, text, application

Description automatically generated

1. 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

1. Instance type: select t2.xlarge from dropdown.

Graphical user interface, application

Description automatically generated

1. 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

1. Else you can create key pair and use it.

Graphical user interface, text, application, email

Description automatically generated

1. 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

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

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

Graphical user interface, text, application, email

Description automatically generated

2

1

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 addresses. 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.

# Section 2

## Appendix 1 Open Jupyter Notebook

Step 1: Login to the 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 opens 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

## Appendix 2: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

* Thisinstance will serve as a 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*
  1. 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 the 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 the 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 be 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 the 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 the following commands 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 a successful connection, it will open CLI.

* Enter the following commands one by one and execute

cd ~/.ssh

nano authorisedkey

Text

Description automatically generated

* Take the cursor on the 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 the 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 the 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 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 the 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 the 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

## Appendix 3: 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)

1. slots per trial 16 - slots per trial 2
2. ctrl+x
3. 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.

# Section 3:

## 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 0/Word Count Example PySpark.ipynb
3. Step3. Importing Libraries
4. Step4. Create Spark Context
5. Step5. Read text file
6. Step6. Perform word counting operation
7. 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 1/PythonPandasEssentials.ipynb
3. Step3. Importing Libraries
4. Step4. Create data
5. Step5. Get Data
6. Step6. Prepare Data
7. 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 1/PythonsklearnEssentials.ipynb
3. Step3. Importing Libraries
4. Step4. Do pre-processing
5. Step5. Do Binarization
6. 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 2/ FeatureEngineeringusingPySpark.ipynb
3. Step3. Importing Libraries
4. Step4. Read Data
5. Step5. Do Basic feature Engineering i.e.check for null values and check what each column is describing about dataset
6. Step6. Perform Normalization and standardization
7. Step7. Analyse data by finding max value
8. Step 8. Convert python Data frame into spark Data frames
9. 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 1/Encoding Categorical Features in Spark.ipynb
3. Step3. Importing Libraries
4. Step4. Initiate Spark Session
5. Step5. Do pre-processing
6. Step6. Do Indexing using String Indexer
7. Step7. Apply one-hot encoding
8. Step8. Play with example
9. 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 4/ Train and Evaluate Classification Model.ipynb
3. Step3. Importing Libraries
4. Step4. Read Data
5. Step5. Do Basic feature Engineering i.e., check for null values and check what each column is describing about dataset
6. Step6. Split dataset into test and train dataset
7. Step7. Train Logistic model
8. Step 8. Test the model and evaluate using evaluation matrix
9. Step 9. Apply multiclass classification using Random Forest Classifier on Iris dataset
10. Step 10. Train and test the model performance
11. 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 5/ Train and Evaluate Regression Models.ipynb
3. Step3. Importing Libraries
4. Step4. Read Data
5. Step5. Data Pre-processing and exploration
6. Step6. Split dataset into test and train dataset
7. Step7. Train Linear Regression model
8. Step 8. Test the model and evaluate using evaluation matrix
9. Step 9. Apply Random Forest Regression on Real estate dataset
10. Step 10. Train and test the model performance
11. 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 6/ Train and Evaluate Clustering Models.ipynb
3. Step3. Importing Libraries
4. Step4. Read Data
5. Step5. Vectorization and Scaler
6. Step6. Applying K-Means Clustering
7. Step7. Visualizing the Results and cluster
8. Step 8. 3D Visualization of clustering
9. Step 9. Applying Bisecting KMeans Clustering
10. Step 10. Visualize results
11. 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 7/ Hyperparameter Tuning.ipynb
3. Step3. Importing Libraries
4. Step4. Read Data
5. Step5. Create spark session
6. Step6. Apply required feature engineering
7. Step7. Tune hyper parameter in Linear Regression Model by specifying the input column, output column, and the maximum number of iterations.
8. Step 8. Define the Parameter Grid for the Linear Regression Model
9. Step 9. Evaluate model performance
10. Step 10. Repeat hyperparameter tunning for random forest regressor
11. Step 11. Evaluate model performance
12. 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 8/ Creating a Neural Network in Spark.ipynb
3. Step3. Importing Libraries
4. Step4. Create spark session
5. Step5. Create Dataframe
6. Step6. Apply required feature engineering
7. Step7.Visualize the dataset.
8. Step 8. Apply normalization
9. Step 9. Define sigmoid function
10. Step 10. Check and visualize sigmoid vs sigmoid deviate
11. Step 11. Evaluate model performance
12. 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 9/ Convolutional Neural Network.ipynb
3. Step3. Importing Libraries
4. Step4. Create spark session
5. Step5. Import dataset and split it into train and test dataset
6. Step6. Apply required feature engineering
7. Step7.Visualize the dataset.
8. Step 8. Train CNN model to classify images in MNIST dataset
9. Step 9. Test CNN model to classify images in MNIST dataset
10. Step 10. Evaluate model performance
11. 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 10/ Predict Apple Stock Market Cost with LSTM.ipynb
3. Step3. Importing Libraries
4. Step4. Create spark session
5. Step5. Load dataset and split it into train and test dataset
6. Step6. Apply required feature engineering
7. Step7.Visualize the dataset.
8. Step 8. Train LSTM model
9. Step 9. Test LSTM model
10. Step 10. Evaluate model performance
11. 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:

1. Step1. Open Jupyter Notebook
2. Step2. Open HP-AI-ML/Lab 11/ Creating a Movie Recommendation Engine with Keras.ipynb
3. Step3. Importing Libraries
4. Step4. Create spark session
5. Step5. Load dataset and split it into train and test dataset
6. Step6. Apply required feature engineering
7. Step7.Visualize the dataset.
8. Step 8. Train Keras model with 20 epochs
9. Step 9. Evaluate model performance
10. 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:

1. Step1. Set up elephas Cluster
2. Step 2. Start cluster using master
3. Step 3. Open Jupyter Notebook on master
4. Step 4. Open HP-AI-ML/Lab 12/ MNIST\_using\_spark\_cluster.ipynb
5. Step 5. Initialise sparksession
6. Step 6. Import the necessary tensorflow and keras libraries
7. Step 7. Load dataset and split it into train and test dataset define get data function, get training function and validation of data
8. Step 8. Define Model and Train it
9. Step7. Evaluate model performance
10. 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.