

CSE 3025 – LARGE SCALE DATA PROCESSING  
PROJECT REVIEW 1

PROJECT TITLE – **FEATURE EXTRACTION FROM IMAGES USING BIGDL**

TEAM MEMBERS -

17BCE1019 – DHRUBANKA DUTTA  
17BCE1026 – ARNAV TRIPATHY  
17BCE1221 – AMAN SHAH  
17BCE1250 – SHIKAR BHARDWAJ

**Hadoop** - Write about hadoop and installation screenshots

Apache Hadoop is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model.

Installation -

To check, use command - `$java -version`

```
aman@ubuntu:~$ java -version
openjdk version "11.0.4" 2019-07-16
OpenJDK Runtime Environment (build 11.0.4+11-post-Ubuntu-1u
buntu218.04.3)
OpenJDK 64-Bit Server VM (build 11.0.4+11-post-Ubuntu-1ubun
tu218.04.3, mixed mode, sharing)
aman@ubuntu:~$
```

- 1) Download Hadoop file using terminal or from website
- 2) Use command `$ tar xzf hadoop-2.7.3.tar.gz` to extract file

```
aman@ubuntu:~$ su
Password:
root@ubuntu:/home/aman# tar xzf hadoop-2.7.3.tar.gz
root@ubuntu:/home/aman# mv hadoop-2.7.3 hadoop/
root@ubuntu:/home/aman# exit
exit
```

- 3) Install Hadoop in Pseudo Distributed Mode
- 4) Move Hadoop folder to `usr/local/hadoop`
- 5) Set Hadoop Environment variables by adding commands to `~/.bashrc` file

6) Use `$ gedit ~/.bashrc` to edit the file

```
aman@ubuntu:~$ gedit ~/.bashrc
```

7) Add the following commands

```
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_INSTALL=$HADOOP_HOME
```

8) Apply changes using `$ source ~/.bashrc`

9) Check Hadoop Installation

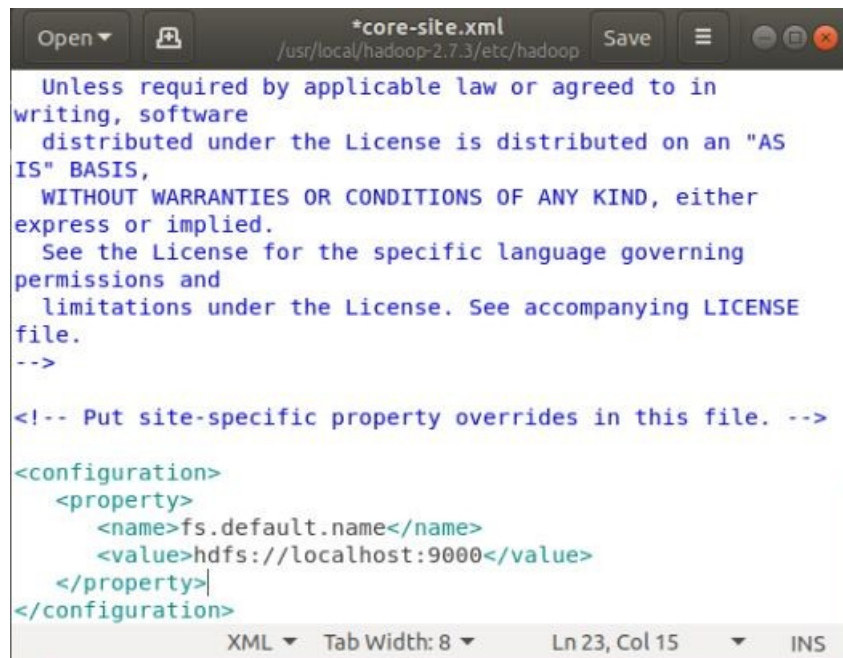
```
aman@ubuntu:/usr/local/jdk-12.0.2$ hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb
92be5982de4719c1c8af91ccff
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /usr/local/hadoop-2.7.3/share/hadoop/common/hadoop-c
ommon-2.7.3.jar
```

10) Use `$ cd $HADOOP_HOME/etc/hadoop`

11) Edit the Hadoop-env.sh file and replace JAVA\_HOME value by export

12) `JAVA_HOME=/usr/local/jdk12.0.2`

13) Edit the core-site.xml

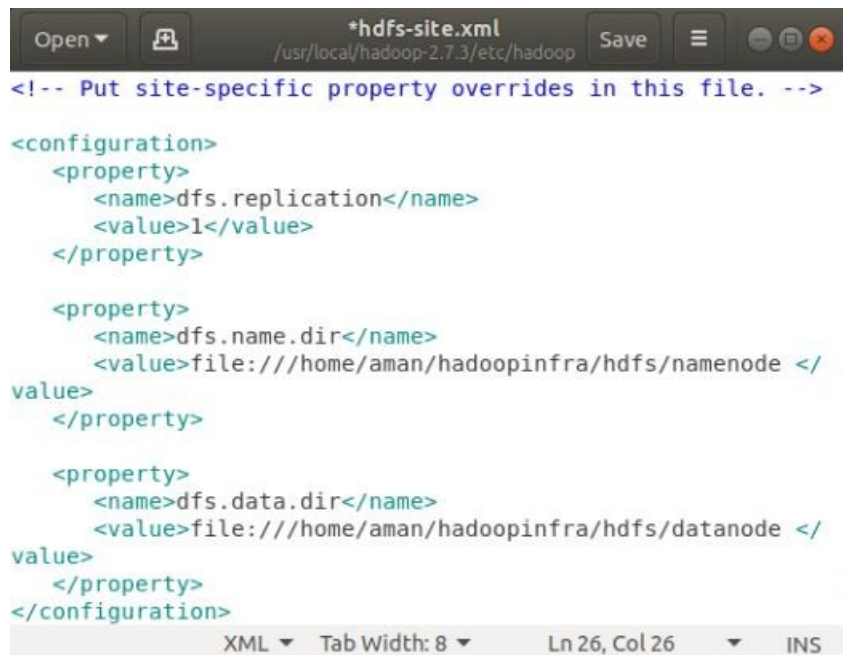


```
Open ▾  *core-site.xml  Save  /usr/local/hadoop-2.7.3/etc/hadoop
Unless required by applicable law or agreed to in
writing, software
distributed under the License is distributed on an "AS
IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied.
See the License for the specific language governing
permissions and
limitations under the License. See accompanying LICENSE
file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
XML ▾  Tab Width: 8 ▾  Ln 23, Col 15  ▾  INS
```

14) Edit hdfs-site.xml



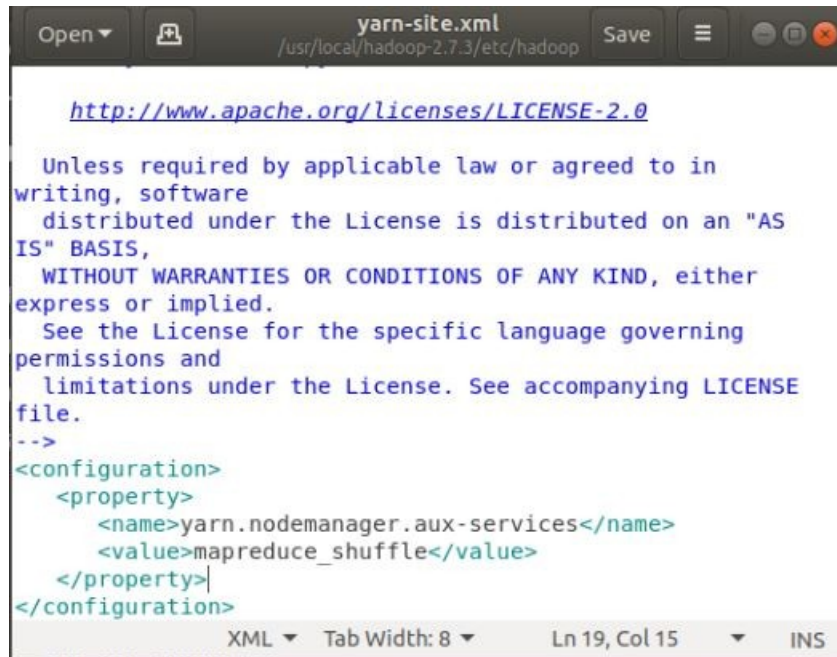
```
Open ▾  *hdfs-site.xml  Save  /usr/local/hadoop-2.7.3/etc/hadoop
<!-- Put site-specific property overrides in this file. -->


<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>

  <property>
    <name>dfs.name.dir</name>
    <value>file:///home/aman/hadoopinfra/hdfs/namenode </
value>
  </property>

  <property>
    <name>dfs.data.dir</name>
    <value>file:///home/aman/hadoopinfra/hdfs/datanode </
value>
  </property>
</configuration>
XML ▾  Tab Width: 8 ▾  Ln 26, Col 26  ▾  INS
```

## 15) Edit yarn-site.xml

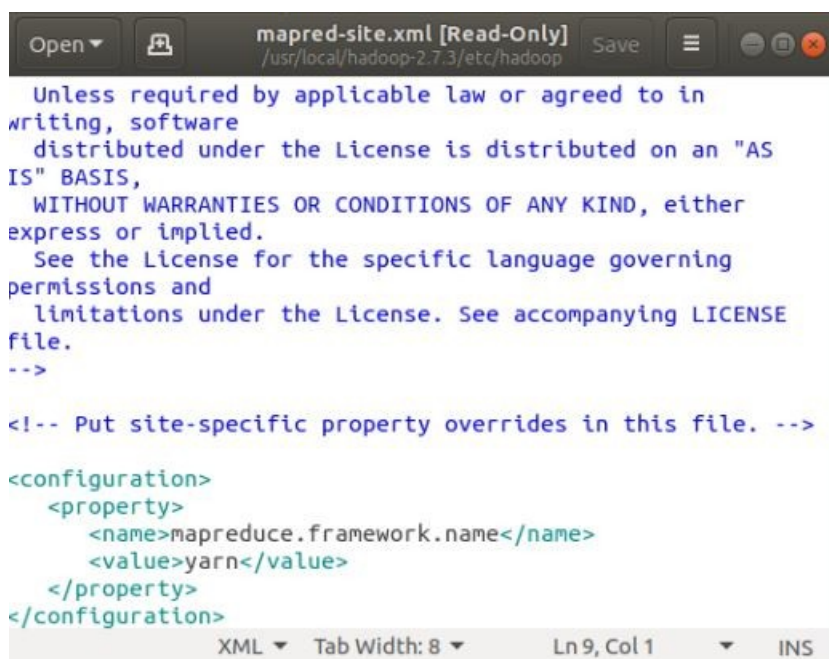
A screenshot of an IDE window titled 'yarn-site.xml' showing the file path '/usr/local/hadoop-2.7.3/etc/hadoop'. The editor contains XML content with a license notice and a configuration block. The status bar at the bottom indicates 'XML', 'Tab Width: 8', 'Ln 19, Col 15', and 'INS' mode.


```
Open ▾  yarn-site.xml /usr/local/hadoop-2.7.3/etc/hadoop Save ≡ ◀ ▶ 🔍 ✖
```

```
http://www.apache.org/licenses/LICENSE-2.0  
  
Unless required by applicable law or agreed to in  
writing, software  
distributed under the License is distributed on an "AS  
IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express or implied.  
See the License for the specific language governing  
permissions and  
limitations under the License. See accompanying LICENSE  
file.  
-->  
<configuration>  
  <property>  
    <name>yarn.nodemanager.aux-services</name>  
    <value>mapreduce_shuffle</value>  
  </property>  
</configuration>
```

XML ▾ Tab Width: 8 ▾ Ln 19, Col 15 ▾ INS

## 16) Edit Mapred-site.xml

A screenshot of an IDE window titled 'mapred-site.xml [Read-Only]' showing the file path '/usr/local/hadoop-2.7.3/etc/hadoop'. The editor contains XML content with a license notice, a comment, and a configuration block. The status bar at the bottom indicates 'XML', 'Tab Width: 8', 'Ln 9, Col 1', and 'INS' mode.

```
Open ▾  mapred-site.xml [Read-Only] /usr/local/hadoop-2.7.3/etc/hadoop Save ≡ ◀ ▶ 🔍 ✖
```

```
Unless required by applicable law or agreed to in  
writing, software  
distributed under the License is distributed on an "AS  
IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express or implied.  
See the License for the specific language governing  
permissions and  
limitations under the License. See accompanying LICENSE  
file.  
-->  
  
<!-- Put site-specific property overrides in this file. -->  
  
<configuration>  
  <property>  
    <name>mapreduce.framework.name</name>  
    <value>yarn</value>  
  </property>  
</configuration>
```

XML ▾ Tab Width: 8 ▾ Ln 9, Col 1 ▾ INS

17) Setup namenode using \$ hdfs namenode -format

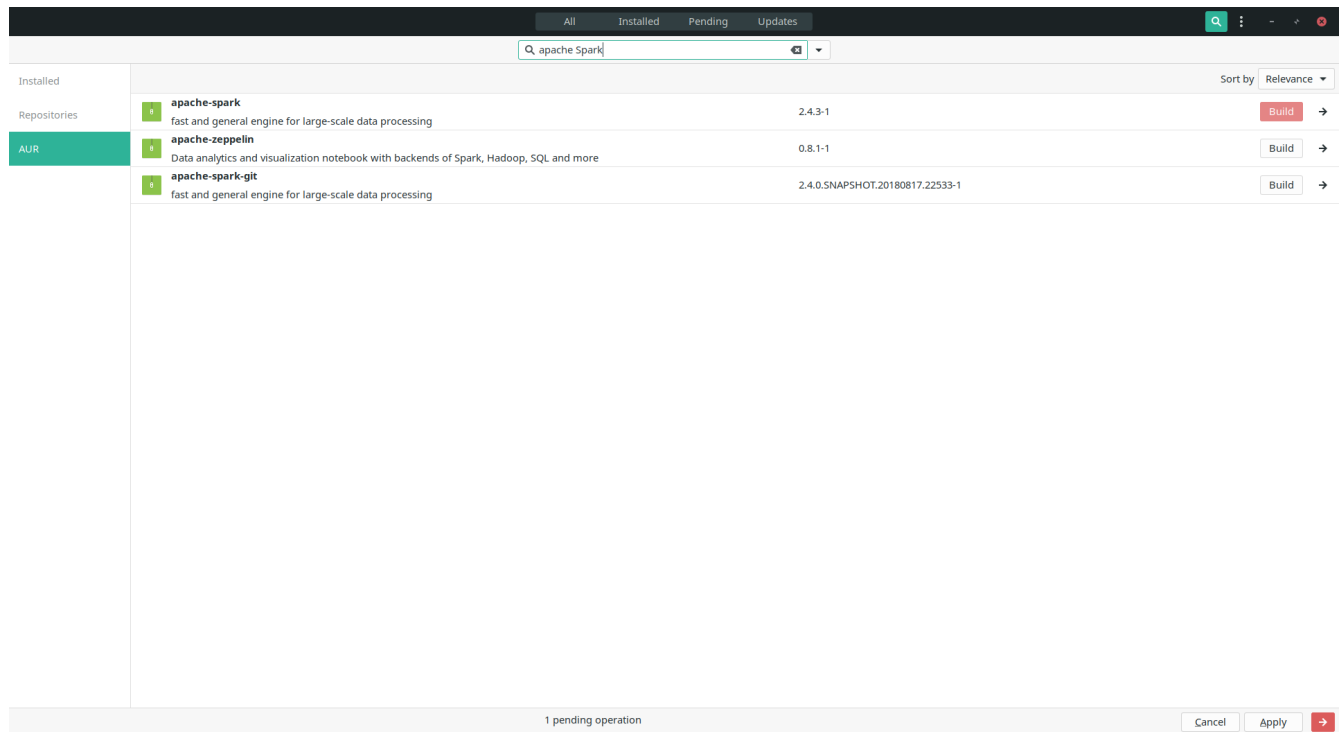
```
aman@ubuntu:/usr/local/hadoop-2.7.3/etc/hadoop$ cd ~
aman@ubuntu:~$ hdfs namenode -format
19/09/26 11:07:58 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = ubuntu/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.7.3
STARTUP_MSG:   config = /usr/local/hadoop-2.7.3/etc/hadoop/hadoop-namenode.xml
*****/
19/09/26 11:08:03 INFO common.Storage: Storage directory /home/aman/hadoopinfra
/hdfs/namenode has been successfully formatted.
19/09/26 11:08:03 INFO namenode.FSImageFormatProtobuf: Saving image file /home/
aman/hadoopinfra/hdfs/namenode/current/fsimage.ckpt_00000000000000000000 using n
o compression
19/09/26 11:08:03 INFO namenode.FSImageFormatProtobuf: Image file /home/aman/ha
doopinfra/hdfs/namenode/current/fsimage.ckpt_00000000000000000000 of size 351 by
tes saved in 0 seconds.
19/09/26 11:08:03 INFO namenode.NNStorageRetentionManager: Going to retain 1 im
ages with txid >= 0
19/09/26 11:08:03 INFO util.ExitUtil: Exiting with status 0
19/09/26 11:08:03 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at ubuntu/127.0.1.1
*****/
ShowApplications
aman@ubuntu:~$
```

## Apache Spark –

Apache Spark is an open-source distributed general-purpose cluster-computing framework. Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. Apache Spark has as its architectural foundation the resilient distributed dataset (RDD), a read-only multiset of data items distributed over a cluster of machines, that is maintained in a fault-tolerant way. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework.

## Installation -

In manjaro arch-linux, apache spark is directly available as a package in the package manager.





```
Building apache-spark...
==> Making package: apache-spark 2.4.3-1 (Wednesday 25 September 2019 04:16:35 PM IST)
==> Checking runtime dependencies...
==> Checking buildtime dependencies...
==> Retrieving sources...
-> Found spark-2.4.3-bin-hadoop2.7.tgz
-> Found apache-spark-master.service
-> Found apache-spark-slave@.service
-> Found spark-env.sh
-> Found spark-daemon-run.sh
-> Found run-master.sh
-> Found run-slave.sh
==> Validating source files with shasums...
spark-2.4.3-bin-hadoop2.7.tgz ... Passed
apache-spark-master.service ... Passed
apache-spark-slave@.service ... Passed
spark-env.sh ... Passed
spark-daemon-run.sh ... Passed
run-master.sh ... Passed
run-slave.sh ... Passed
==> Removing existing $srcdir/ directory...
==> Extracting sources...
-> Extracting spark-2.4.3-bin-hadoop2.7.tgz with bsdtar
==> Starting prepare()...
==> Entering fakeroot environment...
==> Starting package()...
==> Tidying install...
-> Removing libtool files...
-> Purging unwanted files...
-> Removing static library files...
-> Stripping unneeded symbols from binaries and libraries...
-> Compressing man and info pages...
==> Checking for packaging issues...
==> Creating package "apache-spark"...
-> Generating .PKGINFO file...
-> Generating .BUILDINFO file...
-> Adding install file...
-> Generating .MTREE file...
-> Compressing package...
==> Leaving fakeroot environment.
==> Finished making: apache-spark 2.4.3-1 (Wednesday 25 September 2019 04:19:33 PM IST)
==> Cleaning up...

Resolving dependencies...
Checking inter-conflicts...
Running post-transaction hooks...
Checking keyring...
Checking integrity...
Loading packages files...
Checking file conflicts...
Checking available disk space...
Installing apache-spark (2.4.3-1)...
Running post-transaction hooks...
Reloading system manager configuration...
```

```
dhrubanka@dhrubanka-pc:~$ spark-shell
/usr/local/hadoop/bin/hadoop
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/apache-spark/jars/slf4j-log4j12-1.7.16.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
19/09/25 16:20:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://localhost:4040
Spark context available as 'sc' (master = local[*], app id = local-1569408662285).
Spark session available as 'spark'.
Welcome to

  ____      _
 / ___|  __| | | |
 \___ \  / _ \ |_| |
  ___) |/ ___ \  __/
 |_____| \___ \___|_|

 version 2.4.3

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_222)
Type in expressions to have them evaluated.
Type :help for more information.

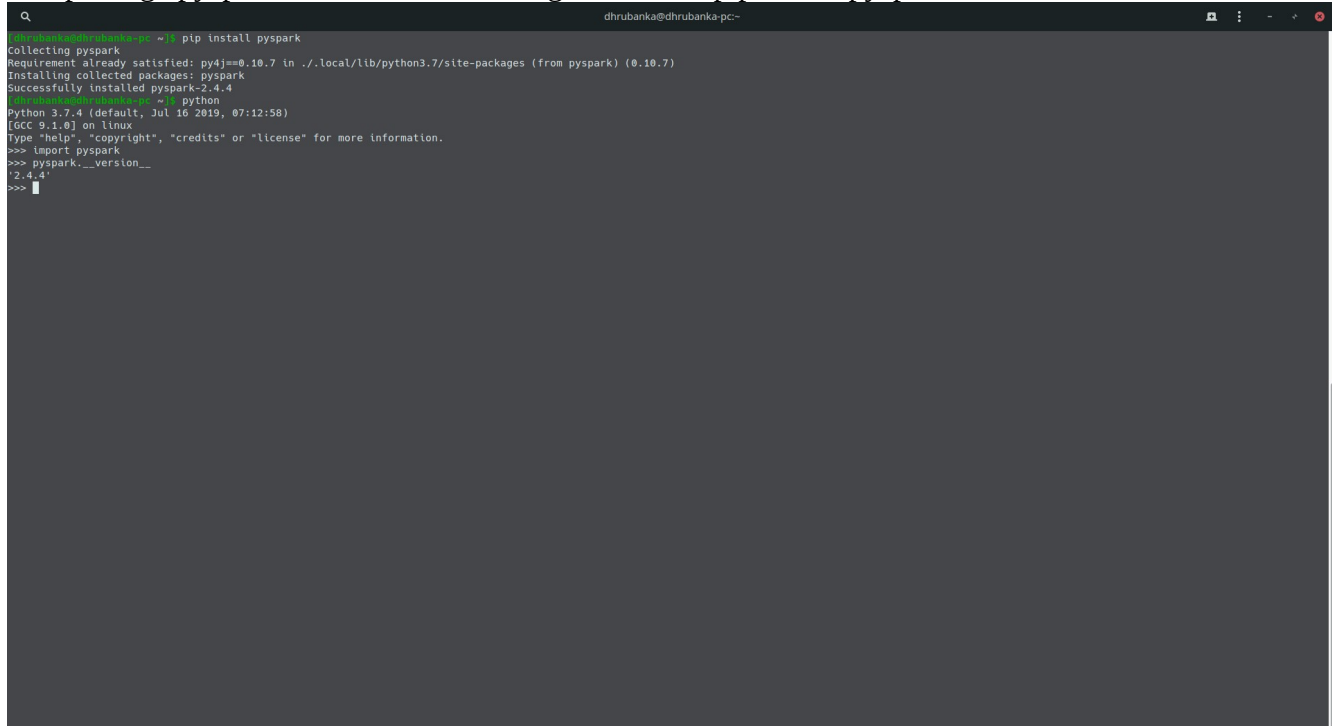
scala>
```

## PySpark -

PySpark is a python API for spark released by Apache Spark community to support python with Spark. Using PySpark, one can easily integrate and work with RDD in python programming language too. There are numerous features that make PySpark such an amazing framework when it comes to working with huge datasets. Whether it is to perform computations on large data sets or to just analyze them, Data engineers are turning to this tool.

## Installation -

The package pyspark can be installed using command, `pip install pyspark --user`.



```
dhrubanka@dhrubanka-pc:~$ pip install pyspark
Collecting pyspark
  Requirement already satisfied: py4j==0.10.7 in ./local/lib/python3.7/site-packages (from pyspark) (0.10.7)
Installing collected packages: pyspark
Successfully installed pyspark-2.4.4

dhrubanka@dhrubanka-pc:~$ python
Python 3.7.4 (default, Jul 16 2019, 07:12:58)
[GCC 9.1.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import pyspark
>>> pyspark.__version__
'2.4.4'
>>>
```



## BigDL -

BigDL is a distributed deep learning framework for Apache Spark, created by Jason Dai at Intel. igDL is a distributed deep learning library for Apache Spark; with BigDL, users can write their deep learning applications as standard Spark programs, which can directly run on top of existing Spark or Hadoop clusters

## Installation -

The package bigdl can be installed using command, `pip install bigdl --user`.

```
dhrubanka@dhrubanka-pc- ~$ pip install BigDL
Collecting BigDL
  Using cached https://files.pythonhosted.org/packages/40/ca/aa8071309d68e88879cd4552ef8a18a577341fc30e2e6fd0760854fb5c6b/BigDL-0.9.0-py2.py3-none-manylinux1_x86_64.whl
Requirement already satisfied: numpy>=1.7 in /usr/lib/python3.7/site-packages (from BigDL) (1.17.0)
Requirement already satisfied: six>=1.10.0 in /usr/lib/python3.7/site-packages (from BigDL) (1.12.0)
Requirement already satisfied: pyspark>=2.2 in ./local/lib/python3.7/site-packages (from BigDL) (2.4.4)
Requirement already satisfied: py4j==0.10.7 in ./local/lib/python3.7/site-packages (from pyspark>=2.2->BigDL) (0.10.7)
Installing collected packages: BigDL
Successfully installed BigDL-0.9.0

dhrubanka@dhrubanka-pc- ~$ python
Python 3.7.4 (default, Jul 16 2019, 07:12:58)
[GCC 9.1.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import bigdl
/home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/util/engine.py:41: UserWarning: Find both SPARK_HOME and pyspark. You may need to check whether they match with each other. SPARK_HOME environment variable is set to: /opt/apache-spark, and pyspark is found in: /home/dhrubanka/.local/lib/python3.7/site-packages/pyspark/__init__.py. If they are unmatched, please use one source only to avoid conflict. For example, you can unset SPARK_HOME and use pyspark only.
  warnings.warn(warning_msg)
Prepending /home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/share/conf/spark-bigdl.conf to sys.path
>>> bigdl
<module 'bigdl' from '/home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/__init__.py'>
>>>
```

## Utils -

## Installation -

The package utils can be installed using command, `pip install utils --user`.

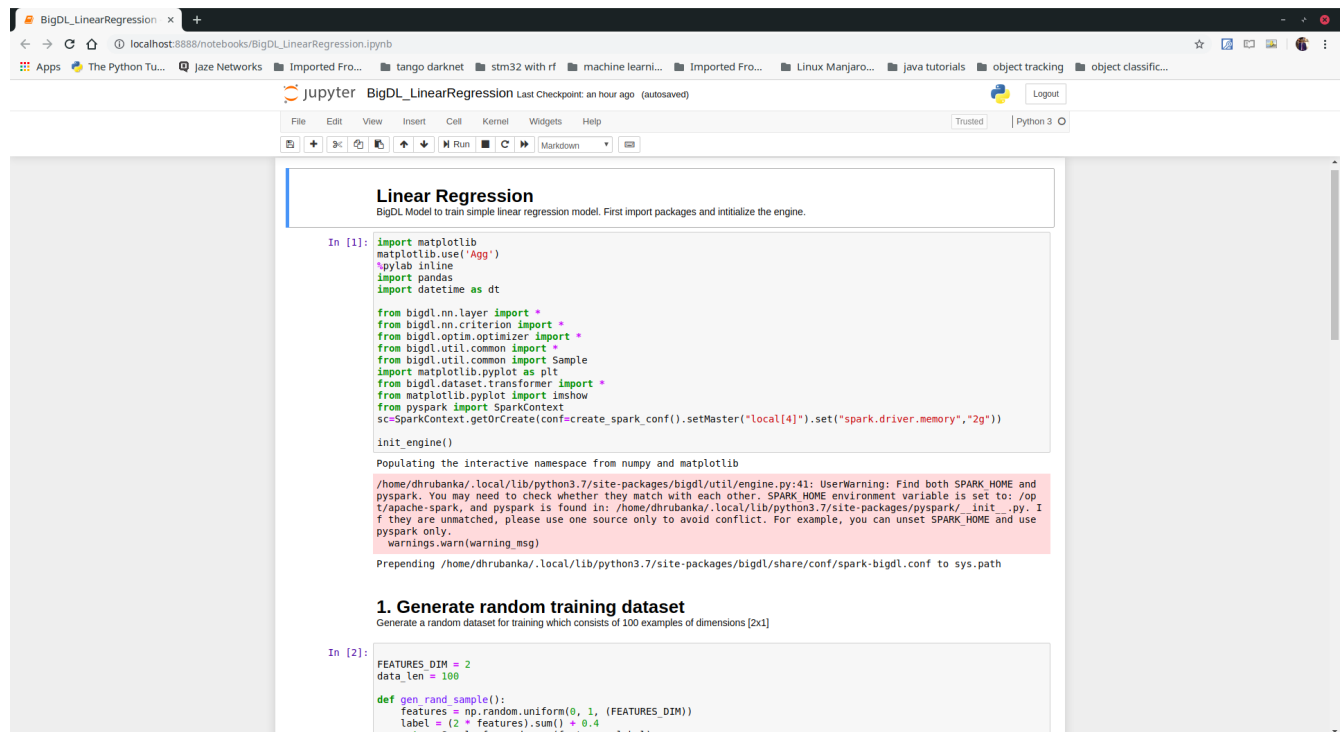
```
dhrubanka@dhrubanka-pc- ~$ pip install git+git://github.com/micahhausler/python3-utils.git --user
Collecting git+git://github.com/micahhausler/python3-utils.git
  Cloning git://github.com/micahhausler/python3-utils.git to /tmp/pip-req-build-wkdtaffu
  Running command git clone -q git://github.com/micahhausler/python3-utils.git /tmp/pip-req-build-wkdtaffu
Requirement already satisfied: six>=1.8.0 in /usr/lib/python3.7/site-packages (from python3-utils==0.4.0) (1.12.0)
Building wheels for collected packages: python3-utils
  Building wheel for python3-utils (setup.py) ... done
  Created wheel for python3-utils: filename=python3_utils-0.4.0-py2.py3-none-any.whl size=5146 sha256=bfb599258b71aa546a68d631406291a2bda6606b0925859f80503b4aa01b6716
  Stored in directory: /tmp/pip-ephem-wheel-cache-7ric2wyl/wheels/05/63/29/1184b904a921546121132296aa09061f1f7384ea9bafc9b97f
Successfully built python3-utils
Installing collected packages: python3-utils
Successfully installed python3-utils-0.4.0

dhrubanka@dhrubanka-pc- ~$ python
Python 3.7.4 (default, Jul 16 2019, 07:12:58)
[GCC 9.1.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import python3_utils
>>> python3_utils.__version__
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: module 'python3_utils' has no attribute '__version__'
>>>
```

```
dhrubanka@dhrubanka-pc:~$ pip install utils
Collecting utils
  Downloading https://files.pythonhosted.org/packages/9b/de/9ffaf89be661b32die0cff85e1af5e4fc2d08c47498975e94aca219aed4/utils-0.9.0-py3-none-any.whl
Installing collected packages: utils
Successfully installed utils-0.9.0
dhrubanka@dhrubanka-pc:~$ python
Python 3.7.4 (default, Jul 16 2019, 07:12:58)
[GCC 9.1.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import utils
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'utils'
>>> import utils
>>> utils
<module 'utils' from '/home/dhrubanka/.local/lib/python3.7/site-packages/utils/__init__.py'>
>>> utils.__version__
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: module 'utils' has no attribute '__version__'
>>>
```

## Example Code :

This is an example code to run linear regression on the BigDL platform.



The screenshot shows a Jupyter Notebook interface with the title 'BigDL\_LinearRegression'. The notebook is running on a local host at localhost:8888. The code in the first cell is as follows:

```
In [1]: import matplotlib
matplotlib.use('Agg')
%pylab inline
import pandas
import datetime as dt

from bigdl.nn.layer import *
from bigdl.nn.criterion import *
from bigdl.optim.optimizer import *
from bigdl.util.common import *
from bigdl.util.common import Sample
import matplotlib.pyplot as plt
from bigdl.dataset.transformer import *
from matplotlib.pyplot import imshow
from pyspark import SparkContext
sc=SparkContext.getOrCreate(conf=create_spark_conf().setMaster("local[4]").set("spark.driver.memory","2g"))

init_engine()

Populating the interactive namespace from numpy and matplotlib

/home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/util/engine.py:41: UserWarning: Find both SPARK_HOME and
pyspark. You may need to check whether they match with each other. SPARK_HOME environment variable is set to: /op
t/apache-spark, and pyspark is found in: /home/dhrubanka/.local/lib/python3.7/site-packages/pyspark/.init_.py. I
f they are unmatched, please use one source only to avoid conflict. For example, you can unset SPARK_HOME and use
pyspark only.
warnings.warn(warning_msg)

Prepending /home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/share/conf/spark-bigdl.conf to sys.path
```

The notebook then displays the first section of the code:

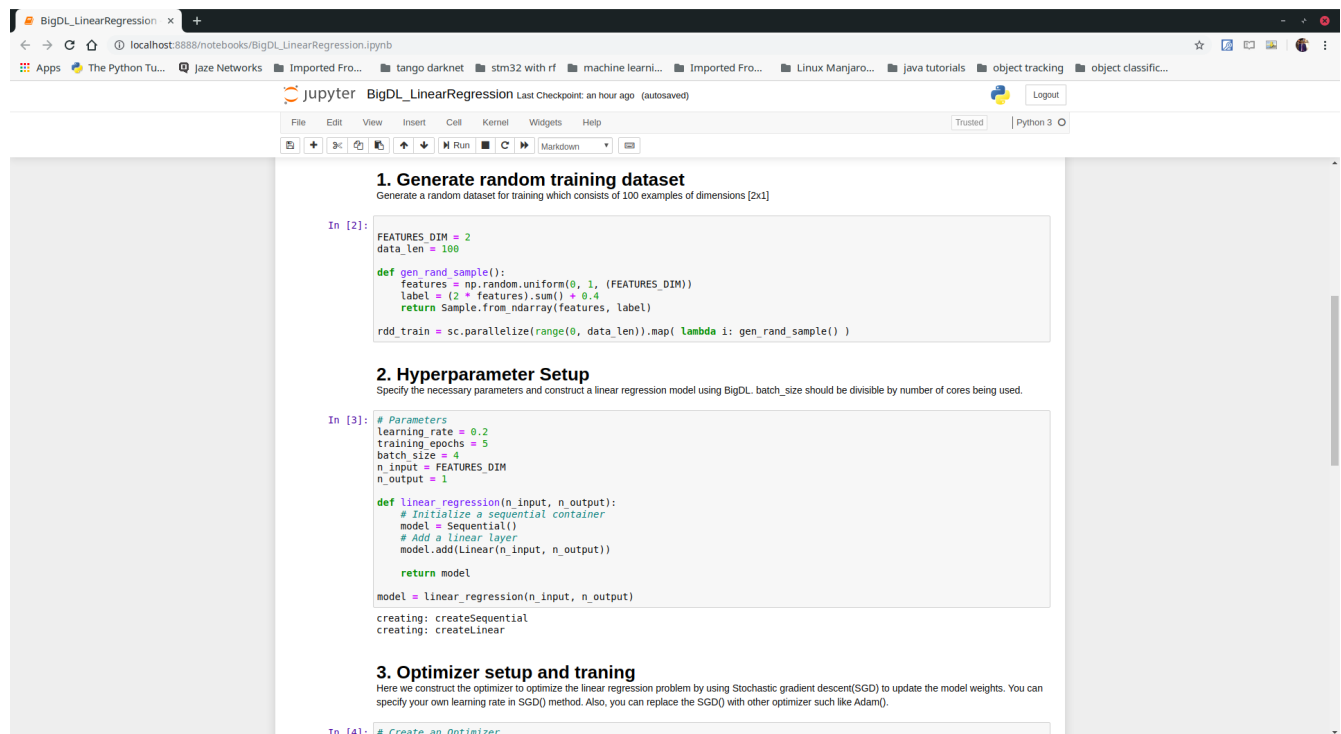
### 1. Generate random training dataset

Generate a random dataset for training which consists of 100 examples of dimensions [2x1]

```
In [2]: FEATURES_DIM = 2
data_len = 100

def gen_rand_sample():
    features = np.random.uniform(0, 1, (FEATURES_DIM))
    label = (2 * features).sum() + 0.4
    return Sample.from_ndarray(features, label)

rdd_train = sc.parallelize(range(0, data_len)).map(lambda i: gen_rand_sample())
```



The screenshot shows the continuation of the Jupyter Notebook. The code in the second cell is as follows:

```
In [2]: FEATURES_DIM = 2
data_len = 100

def gen_rand_sample():
    features = np.random.uniform(0, 1, (FEATURES_DIM))
    label = (2 * features).sum() + 0.4
    return Sample.from_ndarray(features, label)

rdd_train = sc.parallelize(range(0, data_len)).map(lambda i: gen_rand_sample())
```

### 2. Hyperparameter Setup

Specify the necessary parameters and construct a linear regression model using BigDL. batch\_size should be divisible by number of cores being used.

```
In [3]: # Parameters
learning_rate = 0.2
training_epochs = 5
batch_size = 4
n_input = FEATURES_DIM
n_output = 1

def linear_regression(n_input, n_output):
    # Initialize a sequential container
    model = Sequential()
    # Add a linear layer
    model.add(Linear(n_input, n_output))
    return model

model = linear_regression(n_input, n_output)

creating: createSequential
creating: createLinear
```

### 3. Optimizer setup and training

Here we construct the optimizer to optimize the linear regression problem by using Stochastic gradient descent(SGD) to update the model weights. You can specify your own learning rate in SGD() method. Also, you can replace the SGD() with other optimizer such like Adam().

```
In [4]: # Create an Optimizer
```

BigDL\_LinearRegression x +

localhost:8888/notebooks/BigDL\_LinearRegression.ipynb

Apps The Python Tu... Jaze Networks Imported Fro... tango darknet stm32 with rf machine learni... Imported Fro... Linux Manjaro... java tutorials object tracking object classific...

Jupyter BigDL\_LinearRegression Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

### 3. Optimizer setup and training

Here we construct the optimizer to optimize the linear regression problem by using Stochastic gradient descent(SGD) to update the model weights. You can specify your own learning rate in SGD() method. Also, you can replace the SGD() with other optimizer such like Adam().

```
In [4]: # Create an Optimizer
optimizer = Optimizer(
    model=model,
    training_rdd=rdd_train,
    criterion=MSECriterion(),
    optim_method=SGD(learning_rate=learning_rate),
    end_trigger=MaxEpoch(training_epochs),
    batch_size=batch_size)

creating: createMSECriterion
creating: createDefault
creating: createSGD
creating: createMaxEpoch
creating: createDistriOptimizer
```

```
In [5]: # Start to train
trained_model = optimizer.optimize()
```

### 4. Prediction on training data

```
In [6]: # Print the first five predicted results of training data.
predict_result = trained_model.predict(rdd_train)
p = predict_result.take(5)

print("predict predict: \n")
for i in p:
    print(str(i) + "\n")

predict predict:

[3.369568]

[4.1339116]

[2.540867]

[3.773341]

[1.9410255]
```

BigDL\_LinearRegression x +

localhost:8888/notebooks/BigDL\_LinearRegression.ipynb

Apps The Python Tu... Jaze Networks Imported Fro... tango darknet stm32 with rf machine learni... Imported Fro... Linux Manjaro... java tutorials object tracking object classific...

Jupyter BigDL\_LinearRegression Last Checkpoint: an hour ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
p = predict_result.take(5)

print("predict predict: \n")
for i in p:
    print(str(i) + "\n")

predict predict:

[3.369568]

[4.1339116]

[2.540867]

[3.773341]

[1.9410255]
```

### 5. Model evaluation on random test data

```
In [10]: def test_predict(trained_model):
    np.random.seed(100)
    total_length = 10
    features = np.random.uniform(0, 1, (total_length, 2))
    label = (features).sum() + 0.4
    predict_data = sc.parallelize(range(0, total_length)).map(
        lambda i: Sample.from_ndarray(features[i], label))

    predict_result = trained_model.predict(predict_data)
    p = predict_result.take(6)
    ground_label = np.array([[0.47596836], [-0.37598032], [-0.00492062],
                             [-0.5986958], [-0.12307882], [-0.77987401]], dtype="float32")
    mse = ((p - ground_label) ** 2).mean()
    print(mse)

test_predict(trained_model)

8.158284
```