



Twitter Sentiment Analysis in Hive (Big Data Analytics)

Submitted by

Murtaza Jamali

Project Advisor

Moeed Tariq

Muhammad Fayyaz

December 30, 2023

Dice Analytic

ACKNOWLEDGEMENT

First and foremost, I would like to express our gratitude to Almighty Allah for granting me the strength and conviction to advance our project to its current level of accomplishment. Secondly, this dissertation/monograph would not have materialized without the guidance of our instructor, Moeed Tariq. He has been remarkably supportive, and his feedback has significantly enhanced our work. His direct and active involvement in supervising our project is exemplary. His continuous support, endless provision, and consistent inspiration have become our driving force throughout this project.

I also extend our thanks to him for directing us to other individuals involved in the creation of this project. I am deeply appreciative of all those from whom I have gained substantial knowledge:

Our co-instructor, Muhammad Fayyaz, deserves special gratitude for illuminating the right path, providing encouragement during stressful times, having faith in me when confidence was lacking, and dedicating precious time to explain ideas and various aspects of the project. He has offered valuable suggestions, engaged in

Introduction

Introduction:

The field of Twitter Sentiment Analysis has gained significant prominence in recent years, offering a powerful means to comprehend public opinion and reactions on a global scale. This project delves into the abstraction of Twitter Sentiment Analysis, employing Apache Hive for data processing and analysis, and Power BI for visualization.

Objective:

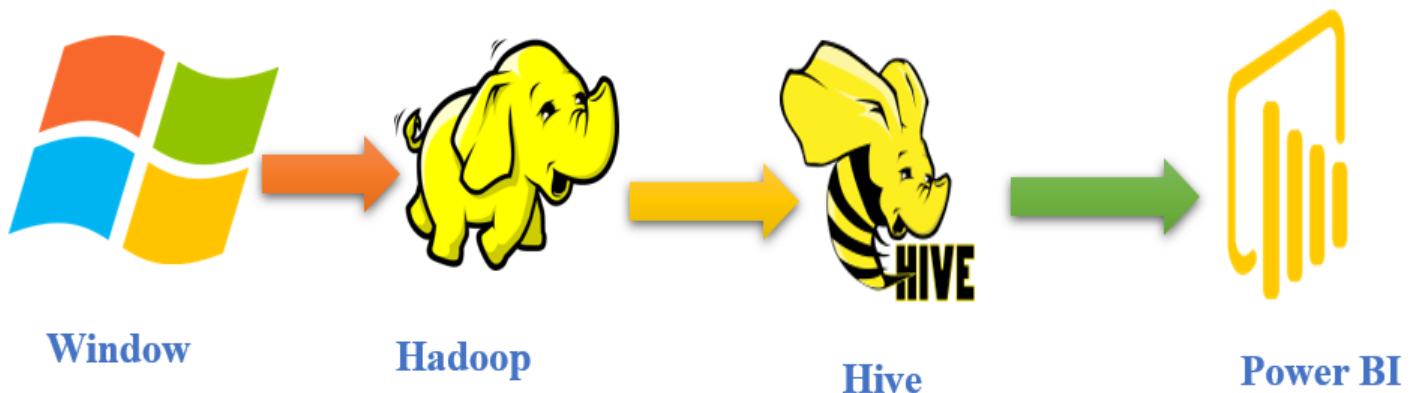
The primary goal of this project is to develop an efficient and scalable solution for Twitter Sentiment Analysis. By utilizing Hive, a data warehousing and SQL-like query language system for Hadoop, in conjunction with Power BI, a robust visualization tool, we aim to provide a comprehensive understanding of sentiments expressed in tweets.

Conclusion:

In summary, this project seeks to offer a robust and scalable solution for Twitter Sentiment Analysis, leveraging the combined capabilities of Apache Hive and Power BI to provide a nuanced understanding of sentiments expressed on the dynamic platform of Twitter.

Work Flow

The workflow ensures a systematic approach to Twitter Sentiment Analysis, integrating the capabilities of Hive for data processing and Power BI for intuitive visualization. It enables users to gain valuable insights from the dynamic landscape of sentiments expressed on Twitter.

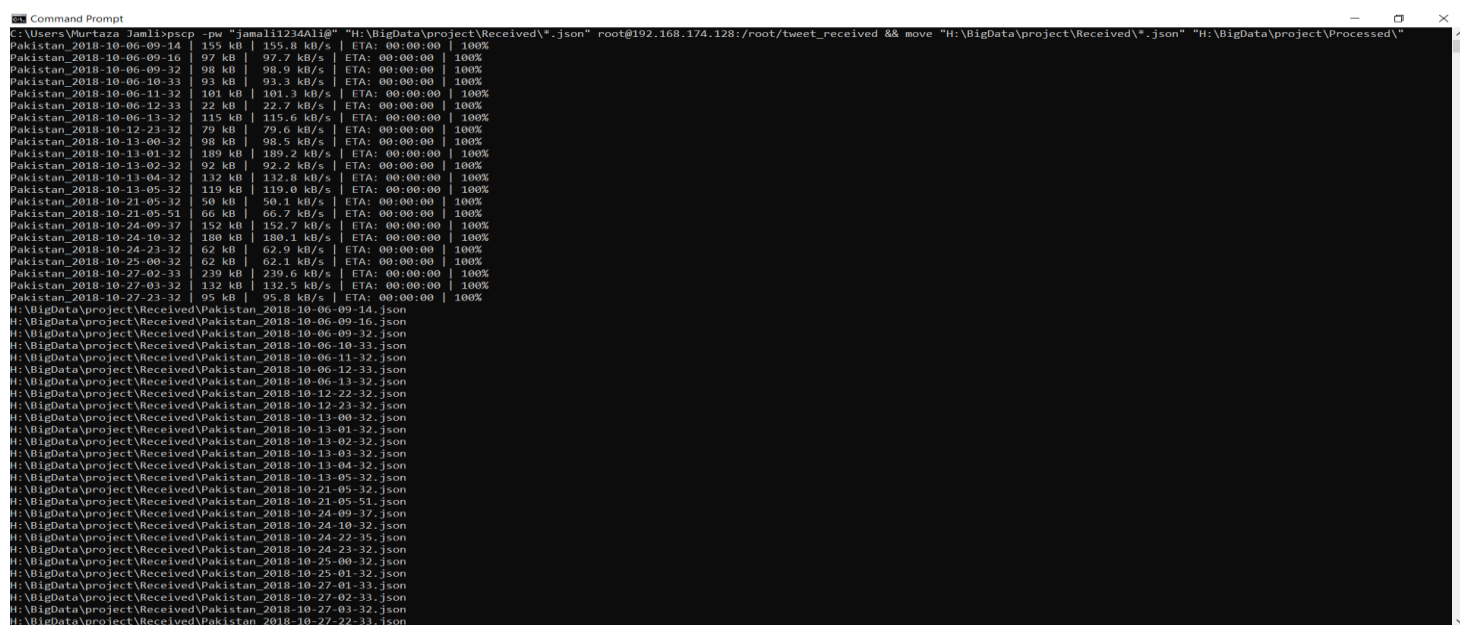


Copy files window to NDFS and Move Files Window Processed folder for backup purpose

This code snippet appears to be a command using the `pscp` utility for secure file transfer. It copies all JSON files from the local directory "H:\BigData\project\Received\" to a remote server with the IP address 192.168.174.128. The destination path on the server is "/root/tweet_received". Additionally, after the files are successfully transferred, the code uses the 'move' command to relocate the same set of JSON files from the local "Received" directory to the "Processed" directory under the same project path. The password for authentication is provided as "jamali1234Ali@" with the `-pw` option in the `pscp` command.

Code:

```
pscp -pw "jamali1234Ali@" "H:\BigData\project\Received\*.json" root@192.168.174.128:/root/tweet_received && move "H:\BigData\project\Received\*.json" "H:\BigData\project\Processed\"
```



```
Command Prompt
C:\Users\Murtaza.Jamali>pscp -pw "jamali1234Ali@" "H:\BigData\project\Received\*.json" root@192.168.174.128:/root/tweet_received && move "H:\BigData\project\Received\*.json" "H:\BigData\project\Processed\"
Pakistan_2018-10-06-09-14 | 155 KB | 155.8 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-06-09-16 | 97 KB | 97.7 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-06-09-32 | 98 KB | 98.9 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-06-10-33 | 93 KB | 93.3 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-06-11-32 | 101 KB | 101.3 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-06-12-32 | 22 KB | 22.7 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-06-13-32 | 115 KB | 115.6 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-12-23-32 | 79 KB | 79.6 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-13-00-32 | 98 KB | 98.5 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-13-01-32 | 189 KB | 189.2 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-13-02-32 | 92 KB | 92.2 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-13-04-32 | 132 KB | 132.8 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-13-05-32 | 119 KB | 119.0 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-21-05-32 | 50 KB | 50.1 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-21-05-51 | 66 KB | 66.7 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-24-09-37 | 152 KB | 152.7 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-24-10-32 | 180 KB | 180.1 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-24-23-32 | 62 KB | 62.9 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-25-00-32 | 62 KB | 62.1 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-27-02-33 | 239 KB | 239.6 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-27-03-32 | 132 KB | 132.5 KB/s | ETA: 00:00:00 | 100%
Pakistan_2018-10-27-23-32 | 95 KB | 95.8 KB/s | ETA: 00:00:00 | 100%
H:\BigData\project\Received\Pakistan_2018-10-06-09-14.json
H:\BigData\project\Received\Pakistan_2018-10-06-09-16.json
H:\BigData\project\Received\Pakistan_2018-10-06-09-32.json
H:\BigData\project\Received\Pakistan_2018-10-06-10-33.json
H:\BigData\project\Received\Pakistan_2018-10-06-11-32.json
H:\BigData\project\Received\Pakistan_2018-10-06-12-32.json
H:\BigData\project\Received\Pakistan_2018-10-06-13-32.json
H:\BigData\project\Received\Pakistan_2018-10-12-23-32.json
H:\BigData\project\Received\Pakistan_2018-10-13-00-32.json
H:\BigData\project\Received\Pakistan_2018-10-13-01-32.json
H:\BigData\project\Received\Pakistan_2018-10-13-02-32.json
H:\BigData\project\Received\Pakistan_2018-10-13-03-32.json
H:\BigData\project\Received\Pakistan_2018-10-13-04-32.json
H:\BigData\project\Received\Pakistan_2018-10-13-05-32.json
H:\BigData\project\Received\Pakistan_2018-10-21-05-32.json
H:\BigData\project\Received\Pakistan_2018-10-21-05-51.json
H:\BigData\project\Received\Pakistan_2018-10-24-09-37.json
H:\BigData\project\Received\Pakistan_2018-10-24-10-32.json
H:\BigData\project\Received\Pakistan_2018-10-24-23-32.json
H:\BigData\project\Received\Pakistan_2018-10-24-23-32.json
H:\BigData\project\Received\Pakistan_2018-10-25-00-32.json
H:\BigData\project\Received\Pakistan_2018-10-25-01-32.json
H:\BigData\project\Received\Pakistan_2018-10-27-01-32.json
H:\BigData\project\Received\Pakistan_2018-10-27-02-33.json
H:\BigData\project\Received\Pakistan_2018-10-27-03-32.json
H:\BigData\project\Received\Pakistan_2018-10-27-23-32.json
```

Automation window to NDFS

The screenshot shows the Windows Task Scheduler interface. A task named 'Movefile_Window_Linux' is highlighted in the task list. The task is configured to run at 11:00 AM every day, repeating every 5 minutes indefinitely. The next run time is 12/29/2023 11:15:00 AM. The last run time was 12/29/2023 11:12:19 AM, and the last run result was 'The operation completed successfully'.

The 'Movefile_Window_Linux Properties (Local Computer)' dialog box is open, showing the 'Actions' tab. The action is 'Start a program' with the command 'H:\projebg\move_file_window_HDFS.bat'. The 'OK' button is highlighted.

The 'move_file_window_HDFS.bat' file is open in Notepad++, showing the following commands:

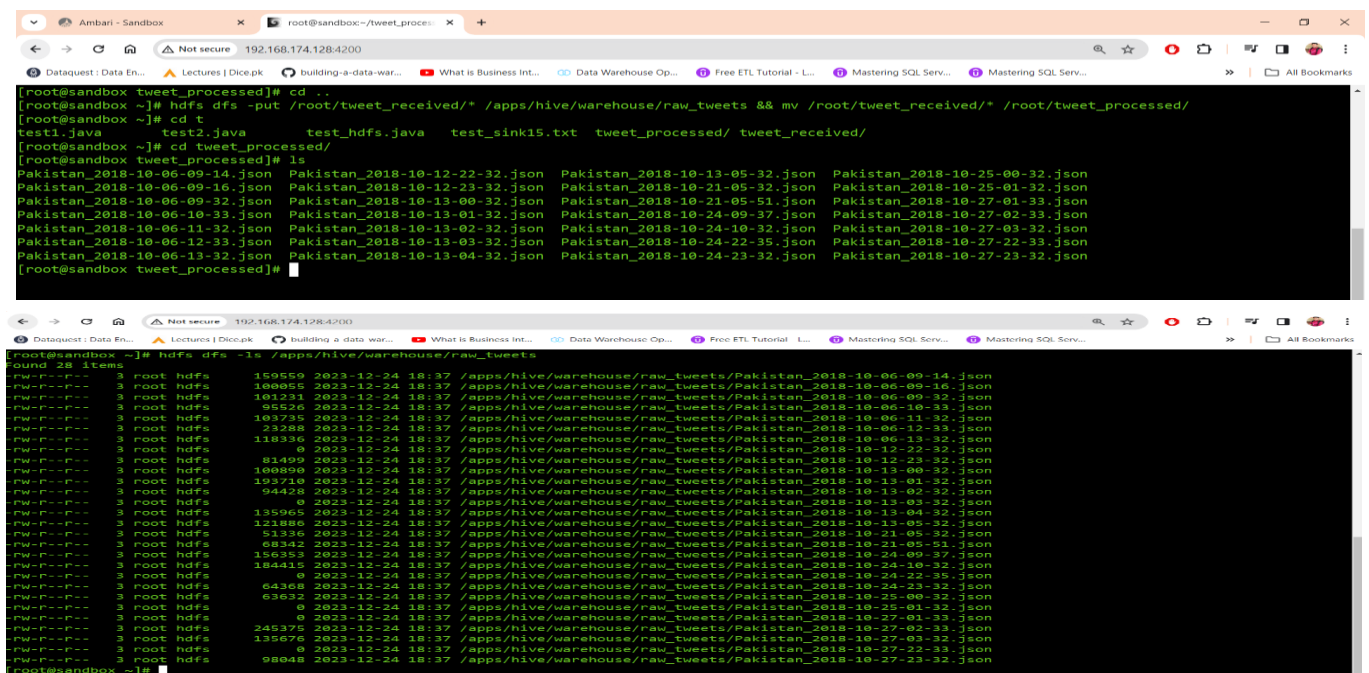
```
1 pscp -pw "jamali1234Ali@" "H:\BigData\project\Received\*.json" root@192.168.174.128:/root/tweet_received &&
2 move "H:\BigData\project\Received\*.json" "H:\BigData\project\Processed\"
```

Copy files NDfs to HDFS and Move Files NDfs tweet_processed folder for backup purpose

This sequence suggests a workflow where raw tweet data is initially uploaded to an HDFS location, likely for processing or storage, and then the local copies are moved to a different directory for further processing or archiving.

Code:

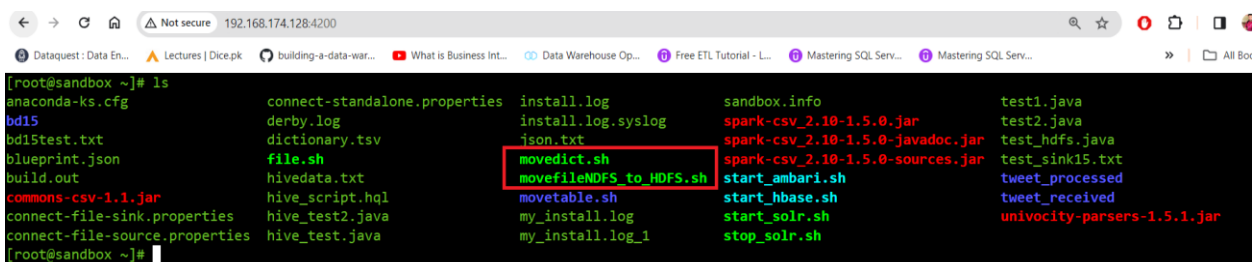
```
hdfs dfs -put /root/tweet_received/* /apps/hive/warehouse/raw_tweets && mv /root/tweet_received/* /root/tweet_processed/
```



```
[root@sandbox tweet_processed]# cd ..
[root@sandbox ~]# hdfs dfs -put /root/tweet_received/* /apps/hive/warehouse/raw_tweets && mv /root/tweet_received/* /root/tweet_processed/
[root@sandbox ~]# cd t
test1.java test2.java test_hdfs.java test_sink15.txt tweet_processed/ tweet_received/
[root@sandbox ~]# cd tweet_processed/
[root@sandbox tweet_processed]# ls
Pakistan_2018-10-06-09-14.json Pakistan_2018-10-12-22-32.json Pakistan_2018-10-13-05-32.json Pakistan_2018-10-25-00-32.json
Pakistan_2018-10-06-09-16.json Pakistan_2018-10-12-23-32.json Pakistan_2018-10-21-05-32.json Pakistan_2018-10-25-01-32.json
Pakistan_2018-10-06-09-32.json Pakistan_2018-10-13-00-32.json Pakistan_2018-10-21-05-51.json Pakistan_2018-10-27-01-33.json
Pakistan_2018-10-06-10-11-32.json Pakistan_2018-10-13-01-32.json Pakistan_2018-10-24-09-37.json Pakistan_2018-10-27-02-33.json
Pakistan_2018-10-06-12-13.json Pakistan_2018-10-13-02-32.json Pakistan_2018-10-24-10-32.json Pakistan_2018-10-27-03-32.json
Pakistan_2018-10-06-12-33.json Pakistan_2018-10-13-03-32.json Pakistan_2018-10-24-22-35.json Pakistan_2018-10-27-22-33.json
Pakistan_2018-10-06-13-32.json Pakistan_2018-10-13-04-32.json Pakistan_2018-10-24-23-32.json Pakistan_2018-10-27-23-32.json
[root@sandbox tweet_processed]#
```

```
[root@sandbox ~]# hdfs dfs -ls /apps/hive/warehouse/raw_tweets
Found 28 items
-rw-r--r-- 3 root hdfs 159559 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-06-09-14.json
-rw-r--r-- 3 root hdfs 100055 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-06-09-16.json
-rw-r--r-- 3 root hdfs 101231 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-06-09-32.json
-rw-r--r-- 3 root hdfs 95526 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-06-10-11-32.json
-rw-r--r-- 3 root hdfs 103735 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-06-11-32.json
-rw-r--r-- 3 root hdfs 23288 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-06-12-13.json
-rw-r--r-- 3 root hdfs 118336 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-06-12-33.json
-rw-r--r-- 3 root hdfs 0 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-12-22-32.json
-rw-r--r-- 3 root hdfs 81499 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-12-23-32.json
-rw-r--r-- 3 root hdfs 100890 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-13-00-32.json
-rw-r--r-- 3 root hdfs 109710 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-13-01-32.json
-rw-r--r-- 3 root hdfs 94428 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-13-02-32.json
-rw-r--r-- 3 root hdfs 0 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-13-03-32.json
-rw-r--r-- 3 root hdfs 135965 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-13-04-32.json
-rw-r--r-- 3 root hdfs 121886 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-13-05-32.json
-rw-r--r-- 3 root hdfs 51336 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-21-05-32.json
-rw-r--r-- 3 root hdfs 88342 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-21-05-51.json
-rw-r--r-- 3 root hdfs 156353 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-24-09-37.json
-rw-r--r-- 3 root hdfs 184415 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-24-10-32.json
-rw-r--r-- 3 root hdfs 0 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-24-22-35.json
-rw-r--r-- 3 root hdfs 64368 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-24-23-32.json
-rw-r--r-- 3 root hdfs 63632 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-25-00-32.json
-rw-r--r-- 3 root hdfs 0 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-25-01-32.json
-rw-r--r-- 3 root hdfs 245375 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-27-01-33.json
-rw-r--r-- 3 root hdfs 135676 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-27-02-33.json
-rw-r--r-- 3 root hdfs 0 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-27-03-32.json
-rw-r--r-- 3 root hdfs 98040 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-27-22-33.json
-rw-r--r-- 3 root hdfs 0 2023-12-24 18:37 /apps/hive/warehouse/raw_tweets/Pakistan_2018-10-27-23-32.json
[root@sandbox ~]#
```

Automation NDfs to HDFS



```
[root@sandbox ~]# ls
anaconda-ks.cfg          connect-standalone.properties  install.log                  sandbox.info                 test1.java
bd15                     derby.log                     install.log.syslog           spark-csv_2.10-1.5.0.jar     test2.java
bd15test.txt             dictionary.tsv                 json.txt                     spark-csv_2.10-1.5.0-javadoc.jar  test_hdfs.java
blueprint.json           file.sh                       movedict.sh                  spark-csv_2.10-1.5.0-sources.jar  test_sink15.txt
build.out                hivedata.txt                 movetable.sh                 start_ambari.sh              tweet_processed
commons-csv-1.1.jar       hive_script.hql              my_install.log               start_hbase.sh               tweet_received
connect-file-sink.properties  hive_test2.java              my_install.log_1            start_solr.sh                 univocity-parsers-1.5.1.jar
connect-file-source.properties  hive_test.java
[root@sandbox ~]#
```

```

[root@sandbox ~]# crontab -l

#copy disctionary file root to hdfs
48 18 * * * /root/movedict.sh

# move files NDFS to HDFS
50 18 * * * /root/movefileNDFS_to_HDFS.sh
[root@sandbox ~]#

```

```

# Move files to HDFS
hdfs dfs -put /root/tweet_received/* /apps/hive/warehouse/raw_tweets

# Move files locally
mv /root/tweet_received/* /root/tweet_processed/

"movefileNDFS_to_HDFS.sh" 13L, 180C

```

Step:1 Transfer Dictionary window to NDFS:

pscp -pw "jamali1234Ali@" "H:\projebg/dictionary.tsv" root@192.168.174.128:/root/

Step:2 NDFS to HDFS

hdfs dfs -put /root/dictionary.tsv /tmp/dictionary/

Automation Dictionary move

```
← → ↻ 🏠 🔒 Not secure 192.168.174.128:4200
🔍 Dataquest : Data En... 📁 Lectures | Dice.pk 🔄 building-a-data-war... 📺 What is Business Int... 🔗 Data Warehouse Op... ⓘ
[root@sandbox ~]#
[root@sandbox ~]#
[root@sandbox ~]# cat movedict.sh
#!/bin/bash

# This script copies a file to HDFS using hdfs dfs -put

# Path to the local file
local_file="/root/dictionary.tsv"

# HDFS destination path
hdfs_destination="/tmp/dictionary/"

# Execute hdfs dfs -put command
hdfs dfs -put "$local_file" "$hdfs_destination"
[root@sandbox ~]#
```

Cron Job Dictionary

```
[root@sandbox ~]# crontab -l

#copy disctionary file root to hdfs
48 18 * * * /root/movedict.sh

# move files NDFS to HDFS
50 18 * * * /root/movefileNDFS_to_HDFS.sh
[root@sandbox ~]#
```


Create Table and Load Raw Tweets

Now we can create the hive table to load tweets in it. In the query below I have only selected some columns from the .json file to show that we can just load the columns we require later in the process. Use the query or modify it to create the table where tweets will be loaded.

Query:

```
CREATE EXTERNAL TABLE IF NOT EXISTS raw_tweets
```

```
(
```

```
  created_at string,
```

```
  id string,
```

```
  id_str string,
```

```
  text string,
```

```
  source string,
```

```
  truncated string,
```

```
  user_tw struct
```

```
<
```

```
  id:string,
```

```
  id_str:string,
```

```
  name:string,
```

```
  screen_name:string,
```

```
  location:string,
```

```
  url:string,
```

```
  description:string,
```

```
  translator_type:string,
```

```
  protected:string,
```

```
  verified:string,
```

```
  followers_count:string,
```

```
  friends_count:string,
```

```
  listed_count:string,
```

```
  favourites_count:string,
```

)

ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'

STORED AS TEXTFILE

LOCATION '/apps/hive/warehouse/raw_tweets/' ;

Hive

Query

Saved Queries

History

UDFs

Upload Table

Database Explorer

project

Search tables...

Databases

bd15

bdtest

default

project

raw_tweets

test

xademo

Query Editor

Worksheet

raw_tweets sample

raw_tweets sample

```

1 CREATE EXTERNAL TABLE IF NOT EXISTS raw_tweets
2 (
3   created_at string,
4   id string,
5   id_str string,
6   text string,
7   source string,
8   truncated string,
9   user_tw_struct
10 )
11 <
12 id:string,
13 id_str:string,
14 name:string,
15 screen_name:string,
16 location:string,
17 url:string,
18 description:string,
19 translator_type:string,
20 protected:string,
21 verified:string,

```

Execute

Explain

Save as...

Kill Session

New Worksheet

Query Process Results (Status: Succeeded)

Save results...

Logs

Results

Filter columns...

previous next

project

Search tables...

Databases

bd15

bdtest

default

project

raw_tweets

test

xademo

Worksheet

raw_tweets sample

raw_tweets sample

```

1 SELECT * FROM raw_tweets LIMIT 100;

```

Execute

Explain

Save as...

Kill Session

New Worksheet

Query Process Results (Status: Succeeded)

Save results...

Logs

Results

Filter columns...

previous next

Create Table and Load Dictionary

Now we create table and Load Dictionary

```
CREATE EXTERNAL TABLE my_dictionary (
```

```
type string,
```

```
length int,
```

```
word string,
```

```
pos string,
```

```
stemmed string,
```

```
polarity string
```

```
)
```

```
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
```

```
STORED AS TEXTFILE;
```

--Following will load data from HDFS directory into Hive table.

```
LOAD DATA INPATH '/tmp/dictionary/dictionary.tsv' INTO TABLE my_dictionary;
```

The screenshot displays a Hive query execution interface. On the left, a sidebar shows a list of databases including 'project', 'bd15', 'bdtest', 'default', 'my_dictionary', 'raw_tweets', 'test', and 'xademo'. The main area shows a worksheet with the following SQL query:

```
1 CREATE EXTERNAL TABLE my_dictionary (  
2 type string,  
3 length int,  
4 word string,  
5 pos string,  
6 stemmed string,  
7 polarity string  
8 )  
9 ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'  
10 STORED AS TEXTFILE;  
11 --Following will load data from HDFS directory into Hive table.  
12 LOAD DATA INPATH '/tmp/dictionary/dictionary.tsv' INTO TABLE my_dictionary;
```

Below the query, the 'Query Process Results (Status: Succeeded)' section is visible. It shows a table with the following columns: my_dictionary.type, my_dictionary.length, my_dictionary.word, my_dictionary.pos, my_dictionary.stemmed, and my_dictionary.polarity. The table contains four rows of data:

my_dictionary.type	my_dictionary.length	my_dictionary.word	my_dictionary.pos	my_dictionary.stemmed	my_dictionary.polarity
weaksubj	1	abandoned	adj	n	neg
weaksubj	1	abandonment	noun	n	neg
weaksubj	1	abandon	verb	y	neg
strongsubj	1	abase	verb	y	neg

Create L1, L2 and L3 views:

To analyze the sentiment of a tweet, we need to break it down into words so that we can match the sentiment of words from dictionary table.

First, we will create Layer 1 view which will extract date elements, convert the text to lowercase and remove any new line characters.

Query:

```
CREATE OR REPLACE VIEW project.Layer1 AS
SELECT
    created_at,
    SUBSTR(created_at, 27, 4) AS years,
    SUBSTR(created_at, 5, 3) AS months,
    SUBSTR(created_at, 9, 2) AS days,
    SUBSTR(created_at, 12, 8) AS times,
    id,
    LOWER(REGEXP_REPLACE(text, '\n', '')) AS text
FROM raw_tweets;
```

The screenshot shows a SQL query execution interface. On the left, a sidebar lists databases: bd15, bdtest, default, project, layer1, my_dictionary, raw_tweets, test, and xademo. The main area displays a query: `SELECT * FROM layer1 LIMIT 100;`. Below the query, there are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'. The 'Query Process Results (Status: Succeeded)' section shows a table with the following columns: layer1.created_at, layer1.years, layer1.months, layer1.days, layer1.times, layer1.id, and layer1.text. The table contains two rows of data.

layer1.created_at	layer1.years	layer1.months	layer1.days	layer1.times	layer1.id	layer1.text
Sat Oct 06 16:14:41 +0000 2018	2018	Oct	06	16:14:41	1048607484695535621	rt @javedmalik: ٻه ڏينهن اڳيتو سماءُ ڦٽا
Sat Oct 06 16:14:54 +0000 2018	2018	Oct	06	16:14:54	1048607539519213568	isabella emb. ci rs.7,950/- to rs. https://t.co/mjbe

Layer 2 view will explode the tweet by ID and separate each word into new line.

Query:

CREATE OR REPLACE VIEW project.Layer2 AS

SELECT

id,

words

FROM

project.Layer1

LATERAL VIEW EXPLODE(SPLIT(text, '\\W+')) text AS words;

The screenshot displays a SQL query editor interface. On the left, a sidebar shows a database schema with tables like 'bd15', 'bdtest', 'default', 'project', 'layer1', 'layer2', 'id', 'words', 'my_dictionary', 'raw_tweets', 'test', and 'demo'. The main editor area shows a query: `1 SELECT * FROM layer2 LIMIT 100;`. Below the editor are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'. The 'Query Process Results' section shows the query succeeded. The 'Results' tab displays a table with two columns: 'layer2.id' and 'layer2.words'. The data is as follows:

layer2.id	layer2.words
1048607484695535621	rt
1048607484695535621	javedmalik
1048607484695535621	
1048607539519213568	isabella
1048607539519213568	emb

Layer 3 view matches each word with the dictionary table picks whether polarity is negative or positive and assigns a value -1 or +1 respectively.

Query:

Create or Replace view project.Layer3 AS

select

id, L2.words,

case d.polarity

when 'negative' then -1

when 'positive' then 1

else 0 end

as polarity

from Layer2 L2 left outer join my_dictionary d

on L2.words=d.word;

The screenshot displays a database management interface. On the left, a sidebar shows a list of databases including 'project', 'bd15', 'bdtest', 'default', 'layer1', 'layer2', 'layer3', 'my_dictionary', 'my_tweets', 'test', and 'xademo'. The main area shows a SQL query editor with the following query: `1 SELECT * FROM layer3 LIMIT 100;`. Below the editor are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'. A green progress bar indicates 100% completion. Below the progress bar, a section titled 'Query Process Results (Status: SUCCEEDED)' shows a table of results. The table has three columns: 'layer3.id', 'layer3.words', and 'layer3.polarity'. The results are as follows:

layer3.id	layer3.words	layer3.polarity
1048607484695535621	rt	0
1048607484695535621	javedmalik	0
1048607484695535621		0
1048607539519213568	isabella	0

Layer 4

Finally, Layer 4 will sum up the polarity against a particular ID which generates the collective sentiment of the tweet with that ID. If you are comfortable with writing complex queries, you can also combine Layer 3 and Layer 4 views and just create one view.

Query:

create or replace view project.sentiment as

select

id,

case

when sum(polarity) > 0 then 'positive'

when sum(polarity) < 0 then 'negative'

else 'neutral' end as sentiment

from layer3 l3 group by id;

The screenshot displays a database management interface. On the left, a sidebar shows a tree of databases including 'project'. The main area is titled 'Worksheet (12)' and contains a SQL query: `1 SELECT * FROM sentiment LIMIT 100;`. Below the query editor are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'. The 'Query Process Results' section shows a status of 'Succeeded' and a table of results with columns 'sentiment.id' and 'sentiment.sentiment'.

sentiment.id	sentiment.sentiment
1048607484695535621	neutral
1048607539519213568	neutral
104860754443392002	positive
1048607553452761088	positive
1048607559937081344	neutral

Now, We have the sentiment of each tweet in *project.sentiment* table/view grouped by each ID. If you wish to combine it with the Raw table or Layer 1 view/table to find the actual content of tweet. You can do so using the following query or create a final table using a CTAS statement.

project

Search tables...

Databases

bd15

bdtest

default

project

layer1

layer2

layer3

my_dictionary

raw_tweets

sentiment

test

xademo

Worksheet (12) x sentiment sample * x

```

1 SELECT
2 L1.*, s.sentiment
3 FROM layer1 L1 LEFT OUTER JOIN sentiment s on L1.id = s.id
4 ;

```

Execute

Explain

Save as...

Kill Session

New Worksheet

100%

Query Process Results (Status: SUCCEEDED)

Save results...

Logs

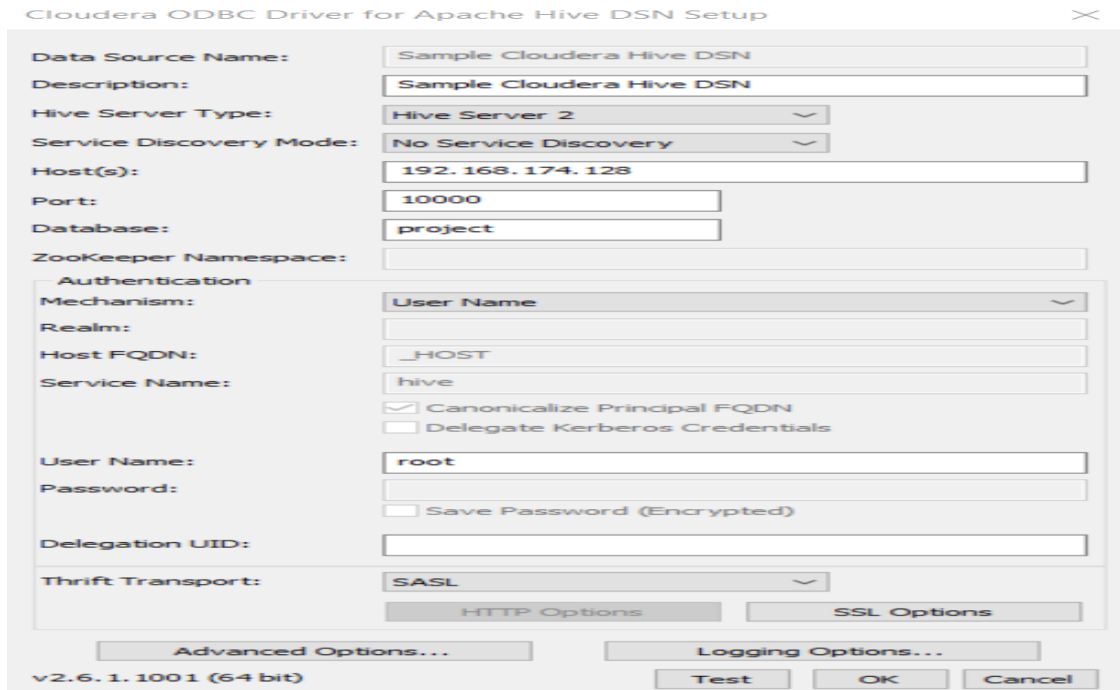
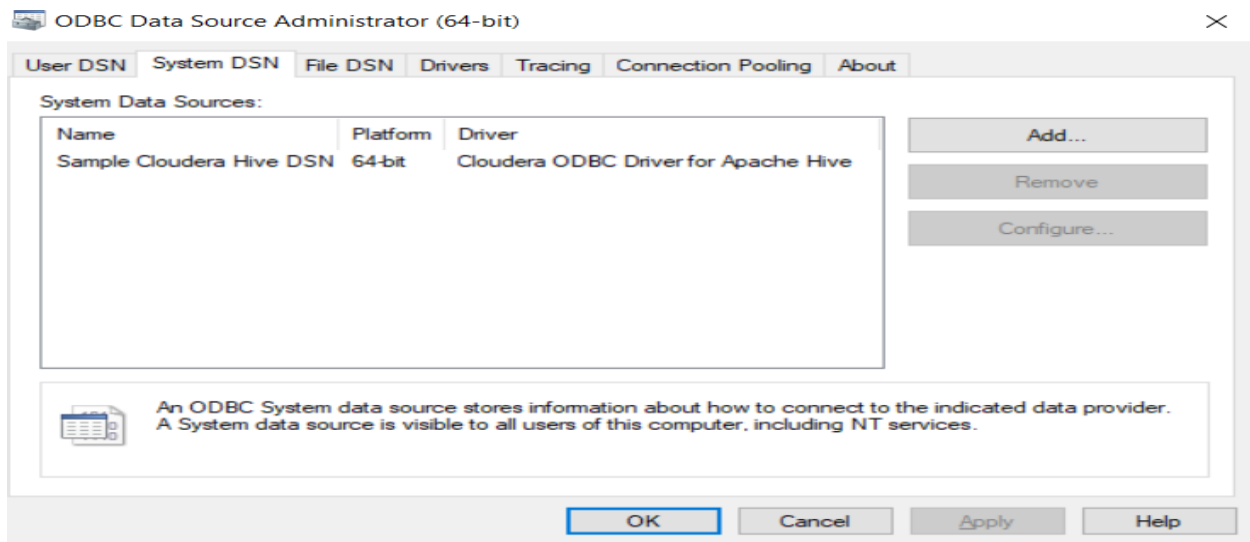
Results

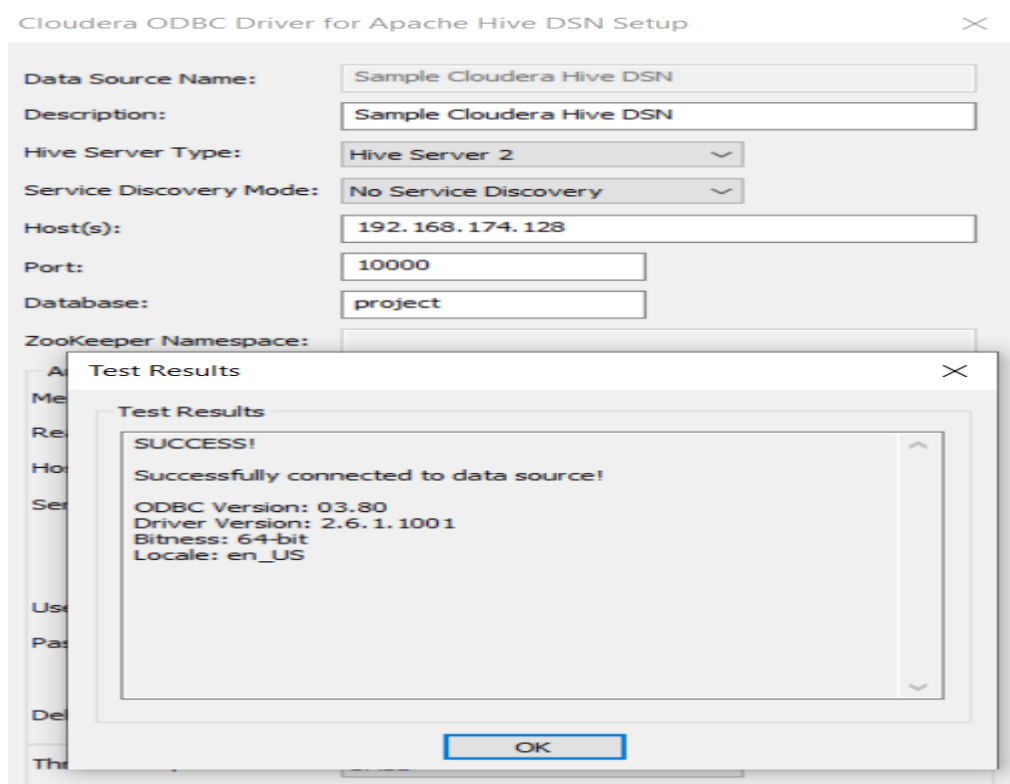
Filter columns...

previous next

l1.created_at	l1.years	l1.months	l1.days	l1.times	l1.id	l1.text
Sat Oct 06 16:14:41 +0000 2018	2018	Oct	06	16:14:41	1048607484695535621	rt @javedmalik: ملک کی بڑی سیاسی جماعت ٹیپار شریف کے خلاف کروالی نے بیت سوالیہ نشا
Sat Oct 06 16:14:54	2018	Oct	06	16:14:54	1048607539519213568	isabella emb. chiffon collection '18 by I rs.7,950/- to rs.8,450/-for order.call/wh

Connect Power BI with Hive:





Navigator

Display Options ▾

- ODBC (dsn=Sample Cloudera Hive DSN) [1]
 - HIVE [6]
 - bd15
 - bdtest
 - default
 - project [6]
 - layer1
 - layer2
 - layer3
 - my_dictionary
 - raw_tweets
 - sentiment
 - test
 - xademo

sentiment

Preview downloaded on Sunday, December 3, 2023

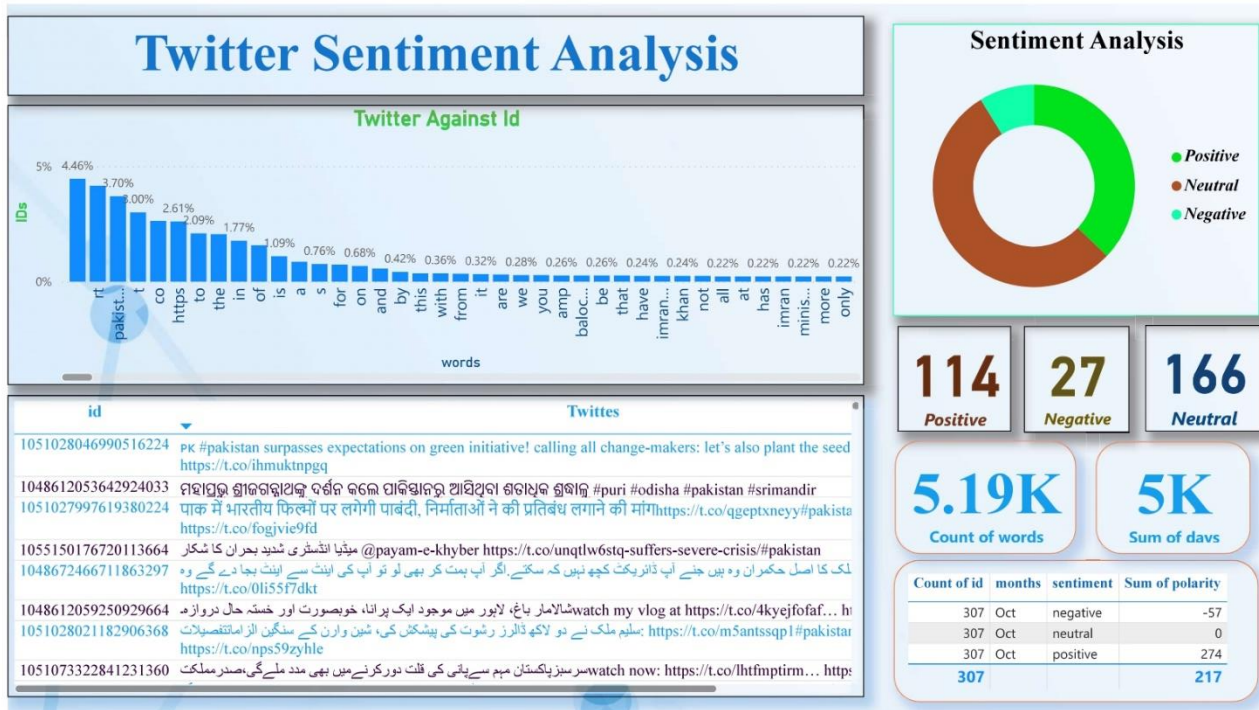
id	sentiment
1048607484695535621	neutral
1048607539519213568	neutral
104860754443392002	positive
1048607553452761088	positive
1048607559937081344	neutral
1048607580023652357	neutral
1048607583051890689	neutral
1048607583672705025	neutral
1048607586193481728	neutral
1048607586780635136	neutral
1048607587816693760	neutral
1048607608993714176	negative
1048607620163162112	neutral
1048607627742199809	neutral
1048607644506836992	positive
1048607646578819073	neutral
1048607647556141056	positive
1048607653713272833	positive
1048607671904079873	neutral
1048607710550405120	neutral
1048607948728090627	positive
1048607952058417152	neutral
1048607963211059201	positive

Select Related Tables

Load

Transform Data

Cancel



Please find All files and Assignments from GitHub:

<https://github.com/murtaza221/Twitter-Sentiment-Analysis-in-Hive-Big-Data-Analytics->