Big Data – Case Study

India Demonetization 2016: Tweet Sentiment Analysis

Objective

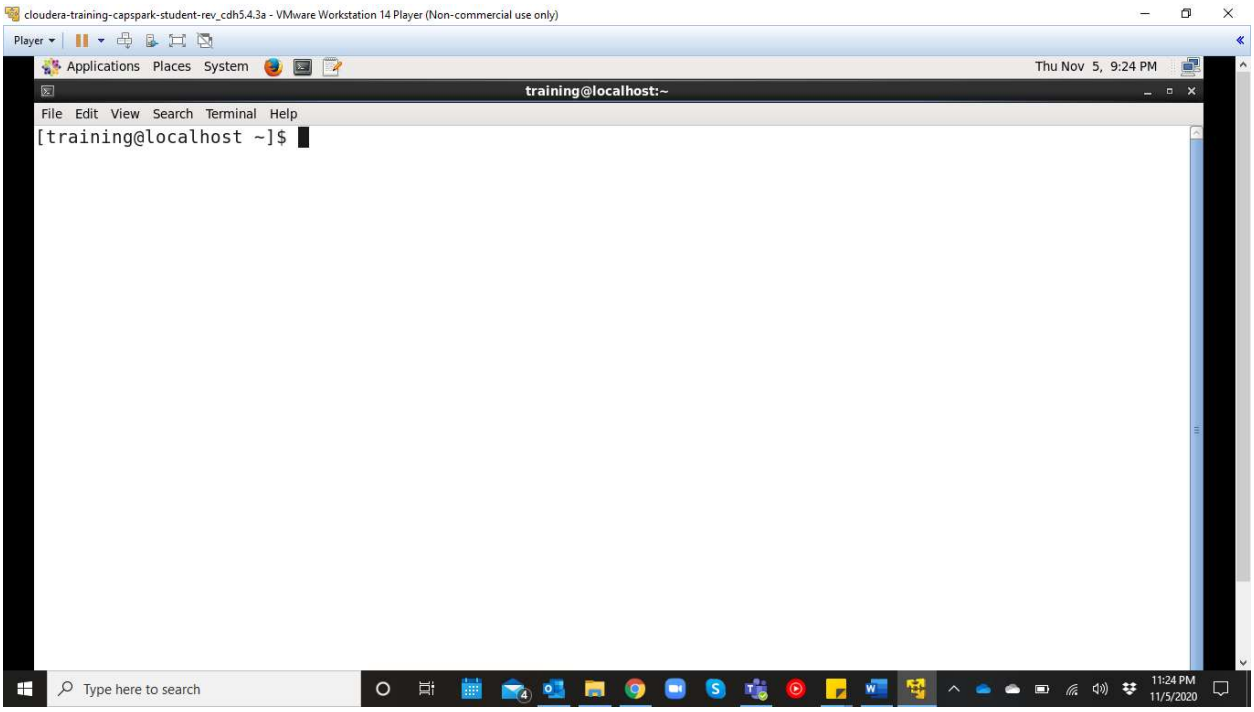
The objective of this case study is to analyze sentiments from twitter of a highly spoken-of event that took place on 8th November 2016 in India. To combat the black money problem in the country, the government decided to demonetize two bills of the highest denominations. We will analyze on how the users reacted to the event using a dictionary and further on, we will visualize the results obtained.

Table of Contents

Part 1: Setup

In this section we will set up the files and data that is required to progress throughout this case study.

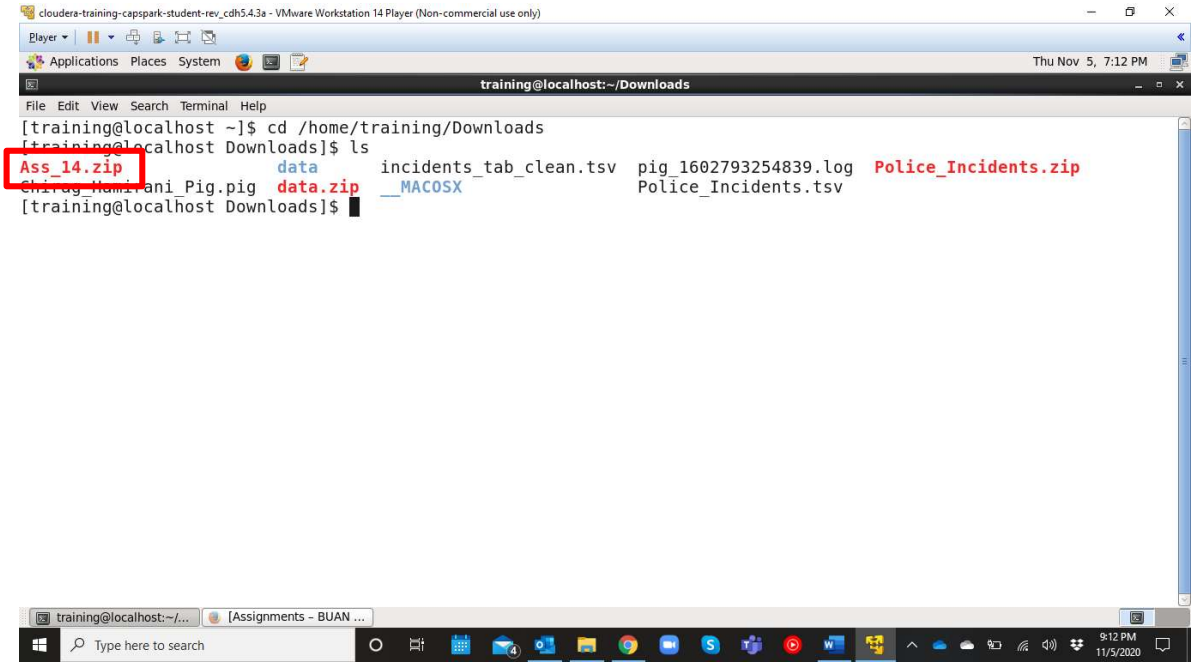
Step-1: Open the terminal application on the Cloudera VM desktop.



Step-2: Download the 'CS01_Dat_Dict_Files.zip' file into the downloads folder **IN** the Cloudera VM using Mozilla Firefox.

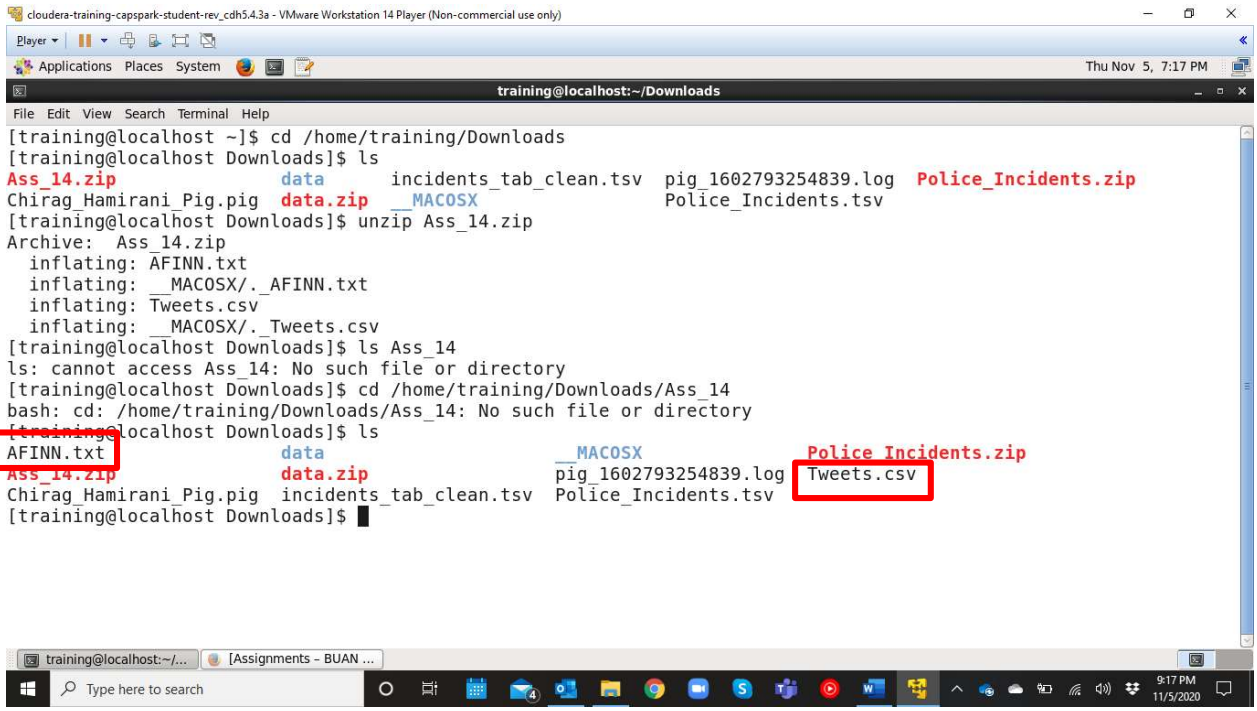
Step-3: Execute the following command:

```
cd/home/training/Downloads
ls
```



Step-4: Execute the following commands at the shell prompt:

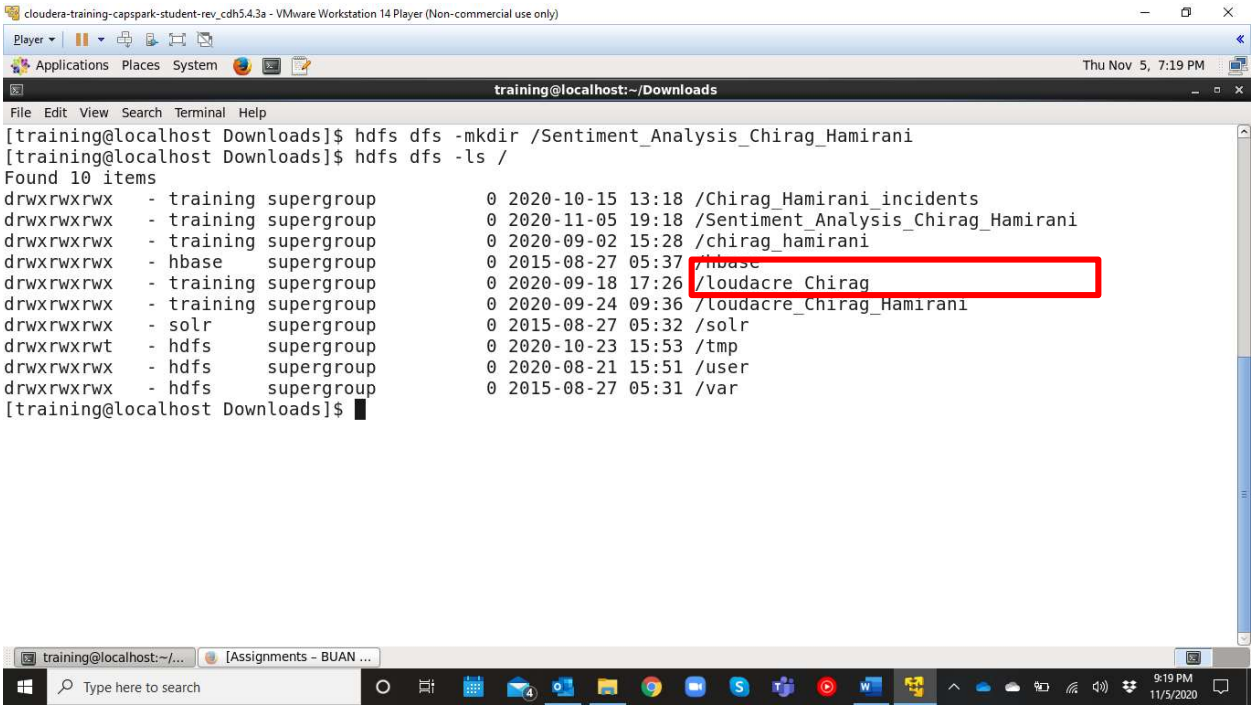
```
unzip CS01_Dat_Dict_Files.zip
ls CS01_Dat_Dict_Files
```



Explanation: The “unzip” command unzips the zip file.

Step-5: Execute the following command at the shell prompt:

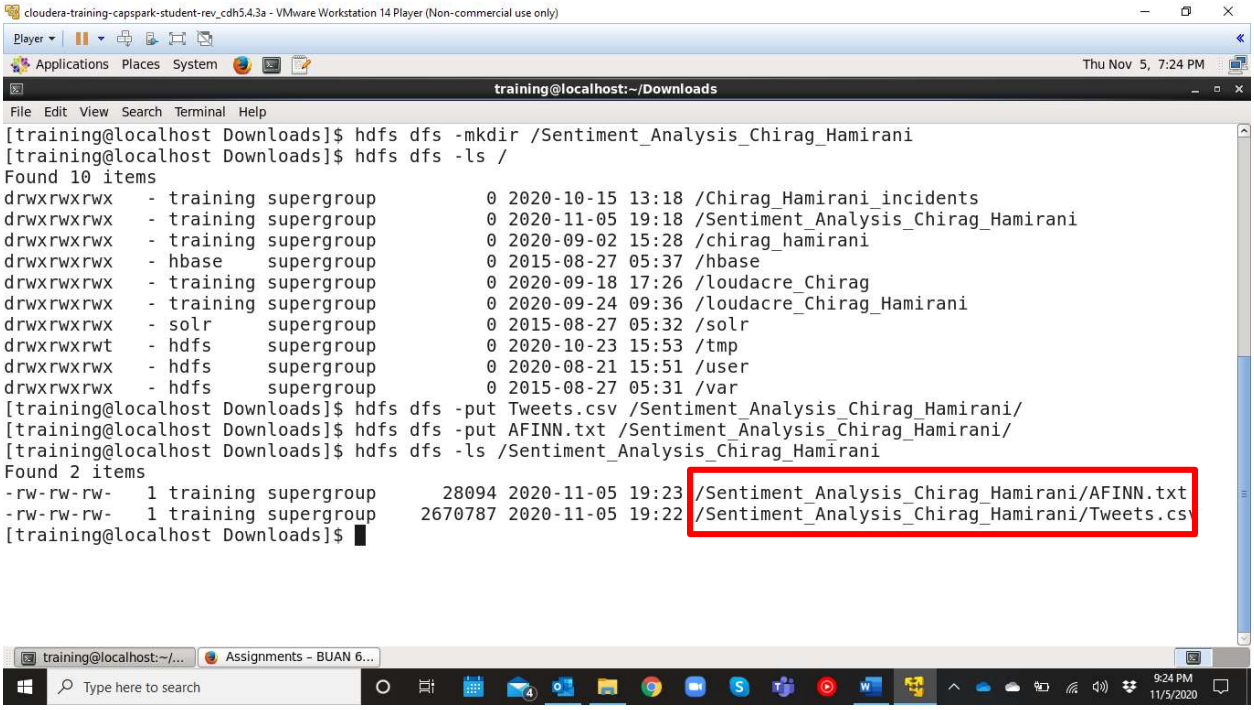
```
hdfs dfs -mkdir /Sentiment_Analysis_Chirag_Hamirani
hdfs dfs -ls /
```



Explanation: We are creating a directory at the root directory for this assignment. We will be using this directory throughout the assignment.

Step-6: Execute the following 2 commands at the shell prompt

```
hdfs dfs -put Tweet.csv /Sentiment_Analysis_Chirag_Hamirani/
hdfs dfs -put AFINN.txt /Sentiment_Analysis_Chirag_Hamirani/
hdfs dfs -ls /Sentiment_Analysis_Chirag_Hamirani/
```



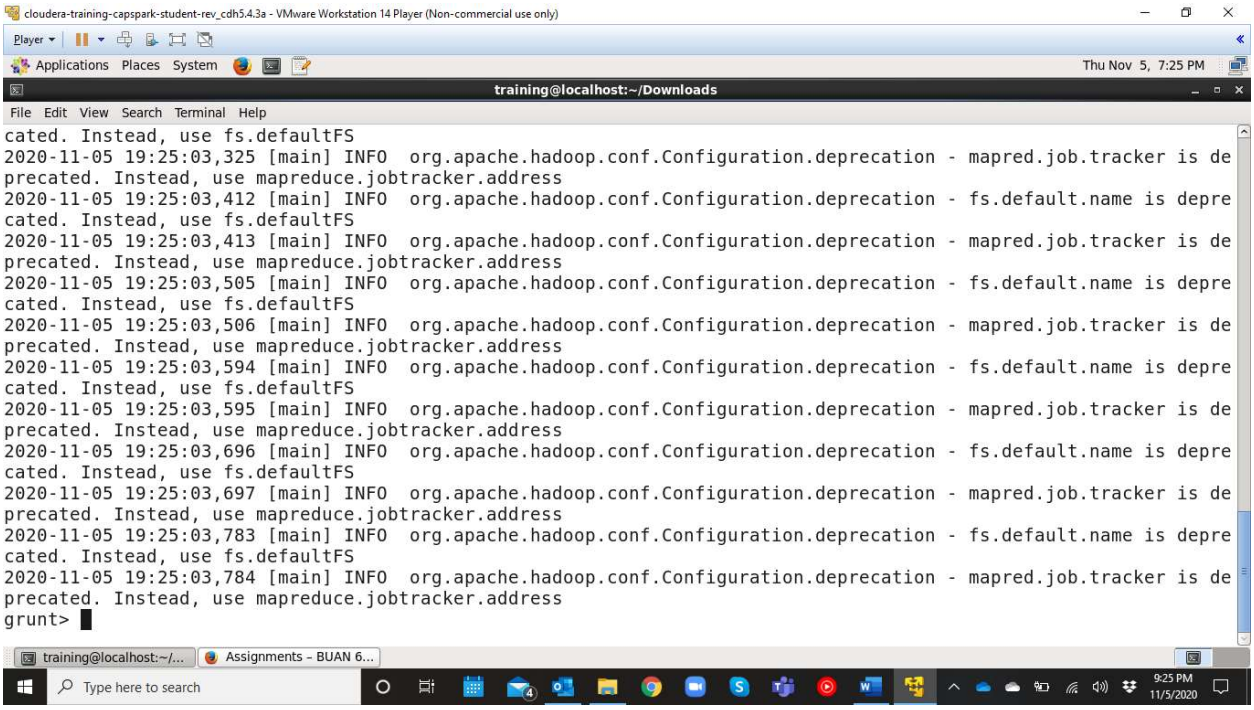
Explanation: We are copying the data files required for this exercise into the directory that we created in Step 5.

Part-2: Sentiment Analysis

In this section we will perform a sentiment analysis on tweets regarding Demonetization in India.

Step-7: Execute the following command at the shell prompt:

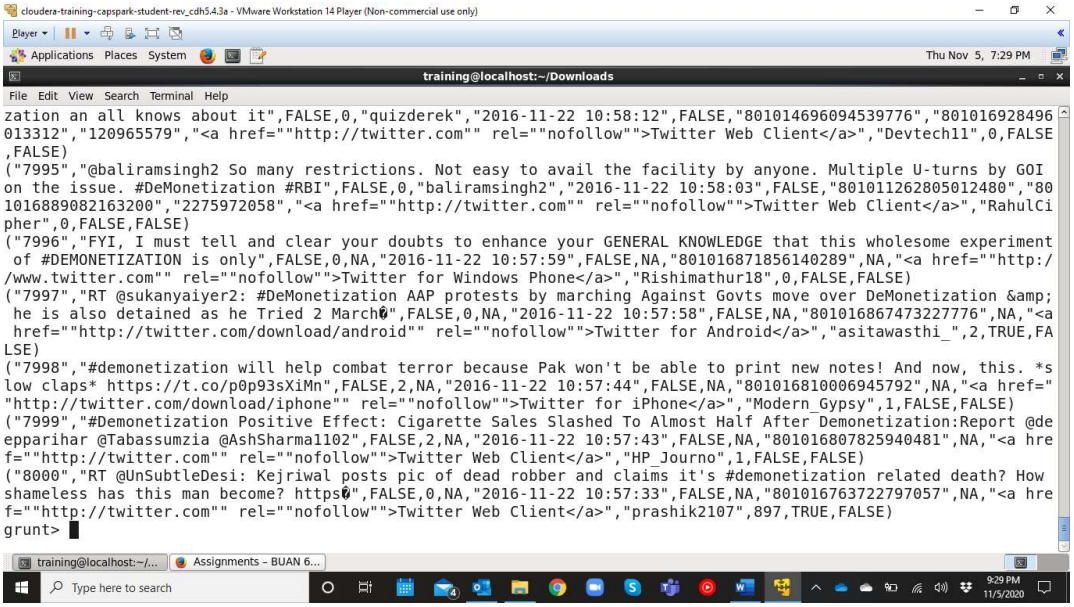
```
pig
```



Explanation: The above command invokes the grunt shell. The grunt shell is an interactive shell to execute pig commands & scripts. The command above can be followed by various additional configurations but for our use case the stand grunt shell will suffice.

Step-8: Execute the following command at the grunt shell prompt:

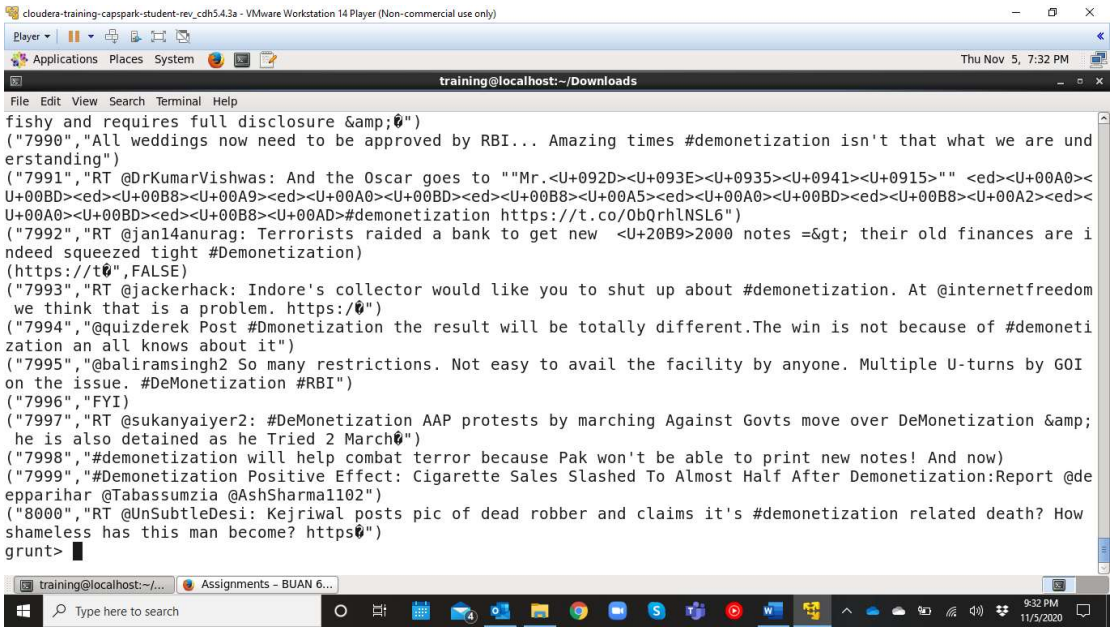
```
Demonit_Tweets = LOAD '/Sentiment_Analysis_Chirag_Hamirani/Tweets.csv'
Using PigStorage(',');
dump Demonit_Tweets;
```



Explanation: The dump command displays the data for the variable specified in the grunt shell.

Step-9: Execute the following command at the grunt shell prompt:

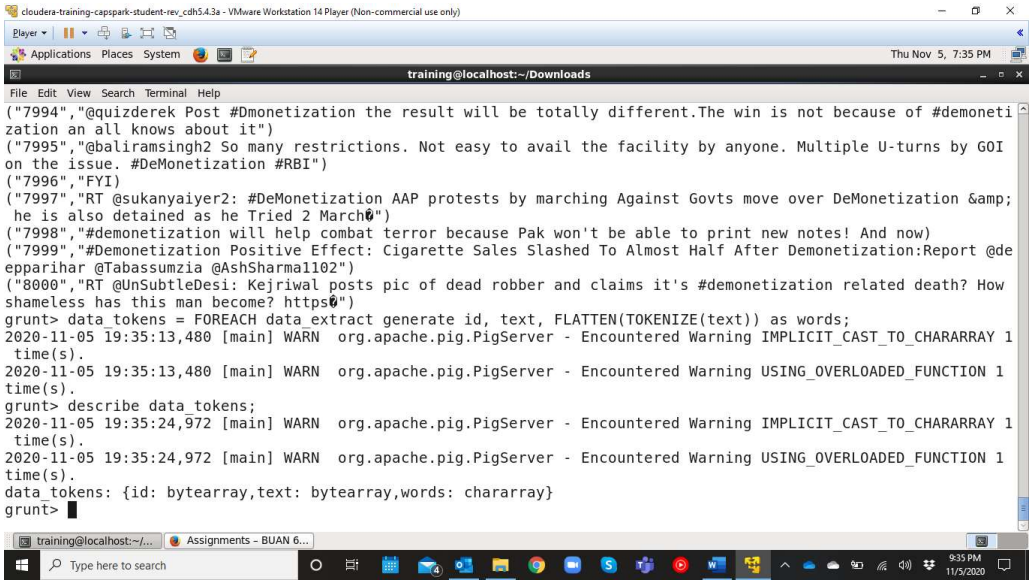
```
data_extract = FOREACH Demonit_Tweets GENERATE $0 as id, $1 as text;
dump data_extract;
```



Explanation: We are defining and collecting the first two data elements in the csv file and dumping the data at the grunt shell.

Step-10: Execute the following command at the grunt shell prompt:

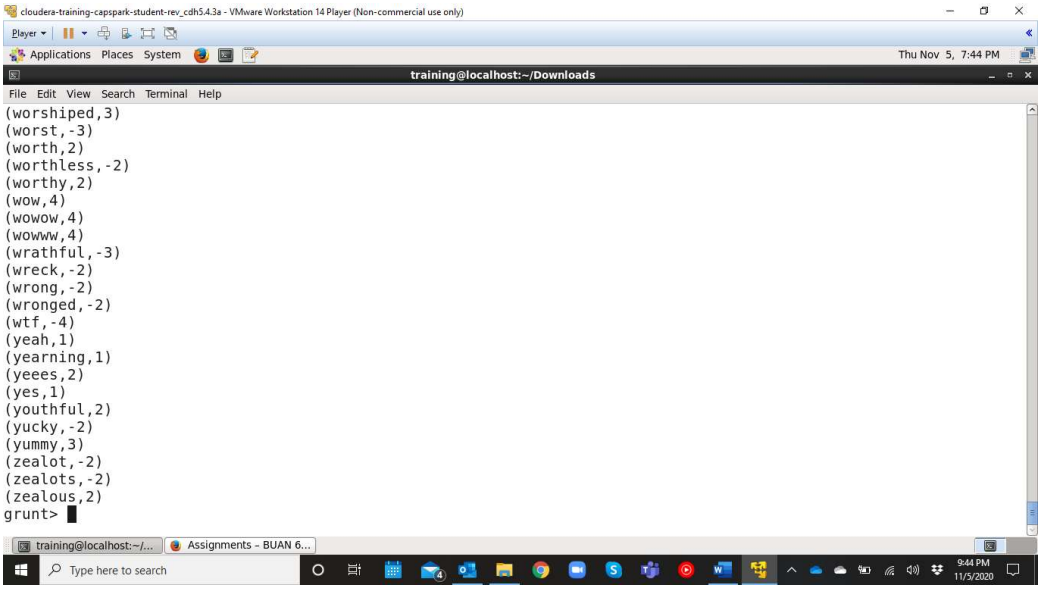
```
data_tokens = FOREACH data_extract generate id, text, FLATTEN(TOKENIZE(TEXT)) as words;
describe data_token;
```



Explanation: We are defining a third element by breaking down the text information in the variable 'text' into words. The describe operation shows us the variable names and types of the data in the variable specified.

Step-11: Execute the following command at the grunt shell prompt:

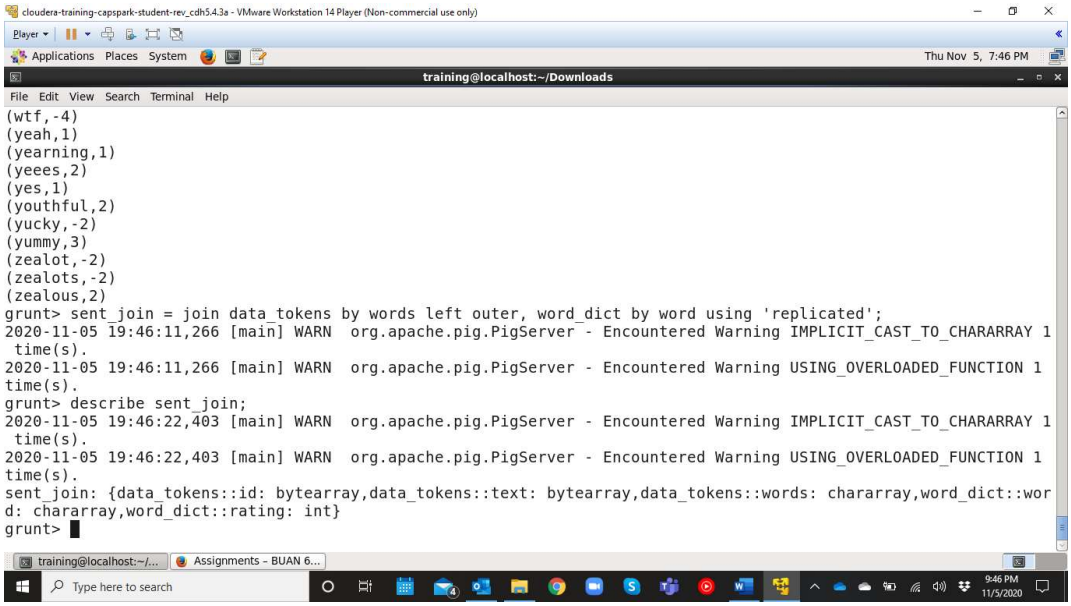
```
word_dict = LOAD '/Sentiment_Analysis_Chirag_Hamirani/AFINN.txt'
USING PigStorage('\t') As(word:chararray, rating:int);
dump word_dict
```



Explanation: AFINN dictionary is a dictionary consisting of 2500 words that are rated on a scale from -5 to +5 depending on the sentiment the word conveys. We will be using this dictionary to perform the sentiment analysis on our tweet data.

Step-12: Execute the following command at the grunt shell prompt:

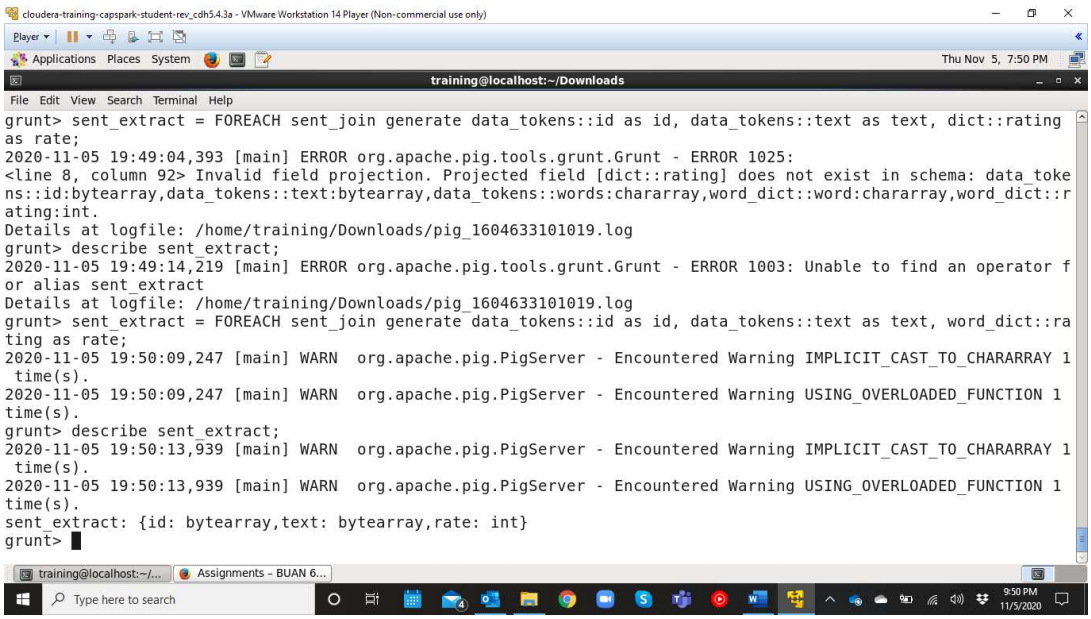
```
sent_join = join data_tokens by words left outer, word_dict by word using 'replicated';
describe sent_join;
```



Explanation: We are joining the sentiments values from the AFINN dictionary with the words in the text from each tweet.

Step-13: Execute the following command at the grunt shell prompt:

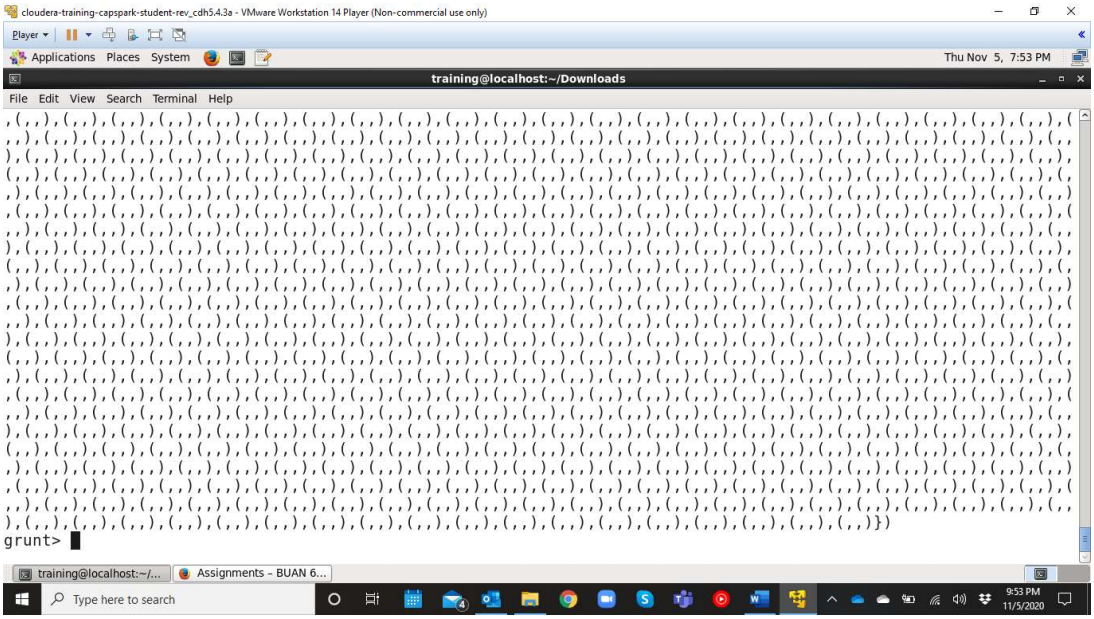
```
sent_extract = FOREACH sent_join generate data_tokens::id as id, data_tokens::text as text, word_dict::rating as rate;
describe sent_extract;
```



Explanation: We are now extracting the data from the variable in step 12 that we need to complete our sentiment analysis.

Step-14: Execute the following command at the grunt shell prompt:

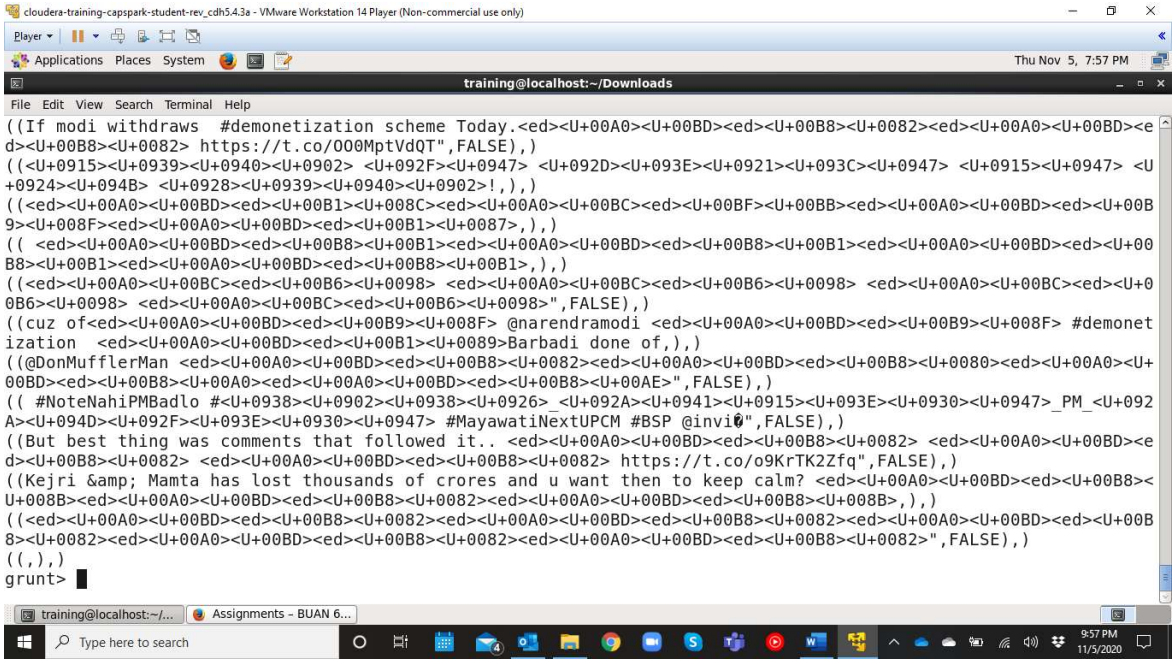
```
word_grouping = group sent_extract by (id,text);
dump word_grouping;
```



Explanation: We are grouping each record by its id and text.

Step-15: Execute the following command at the grunt shell prompt:

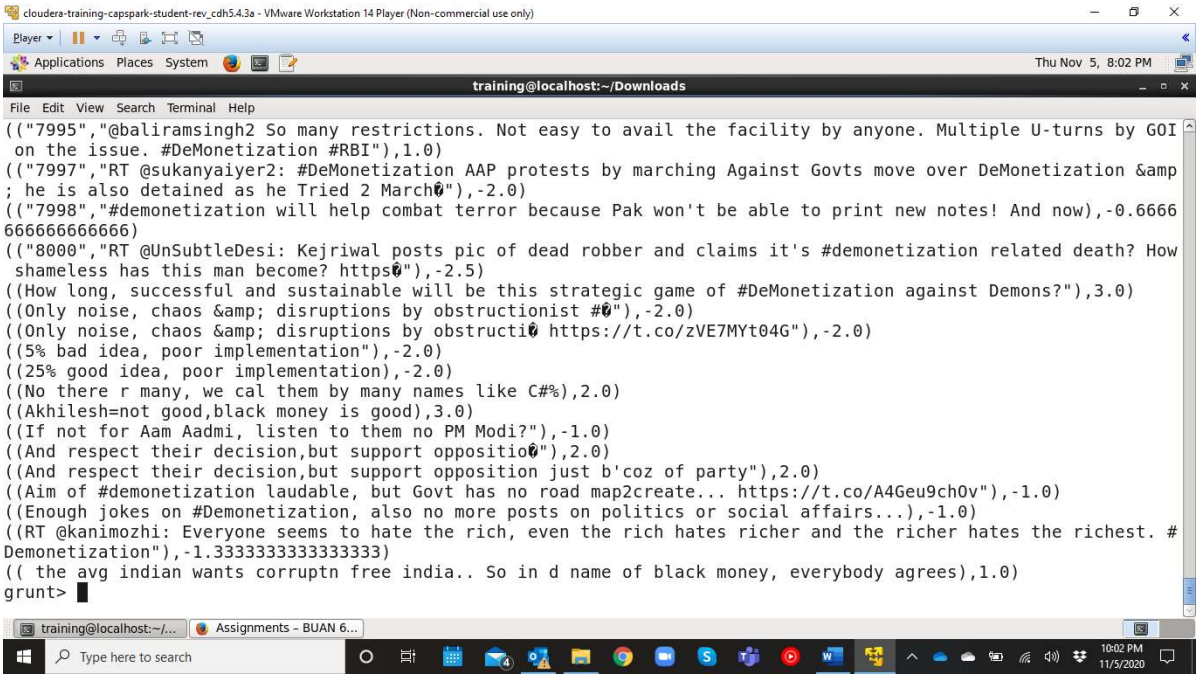
```
average_rating = FOREACH word_grouping generate group, AVG(sent_extract.rate) as tweet_rate;
dump average_rating;
```



Explanation: We are averaging the rating for all the words in each tweet and storing the result in average_rating.

Step-16: Execute the following command at the grunt shell prompt:

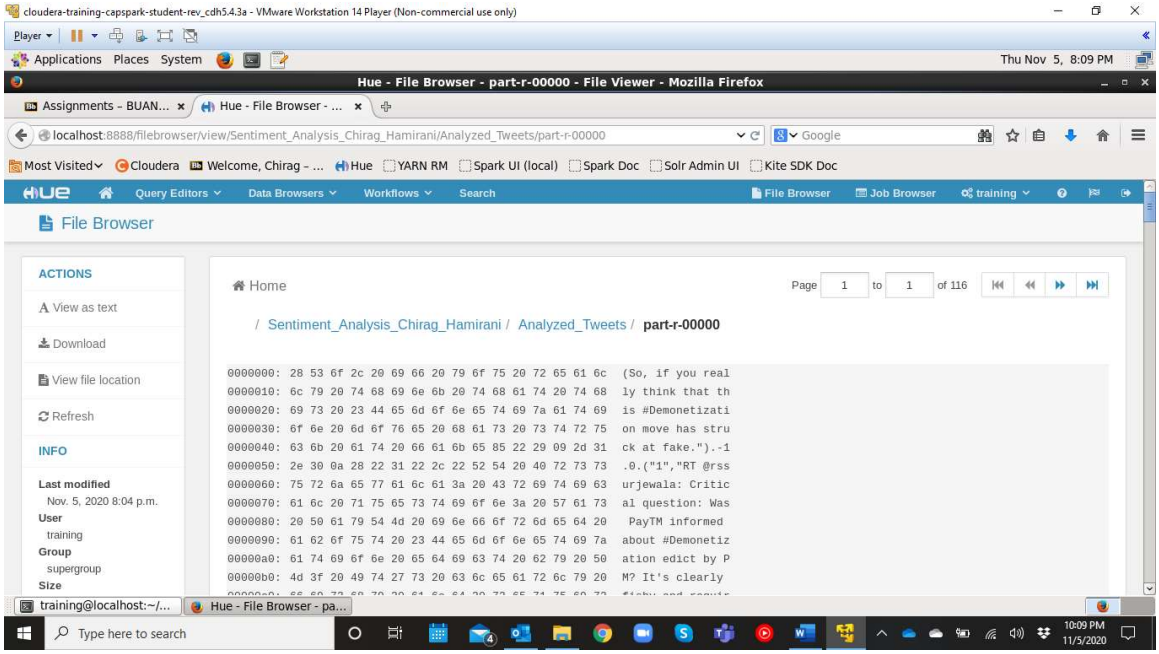
```
final_tweets = filter average_rating by tweet_rate >= -5;
dump final_tweets;
```



Explanation: We are filtering all the tweets for those tweets that have an average rating >= -5.

Step-17: Execute the following command at the grunt shell prompt:

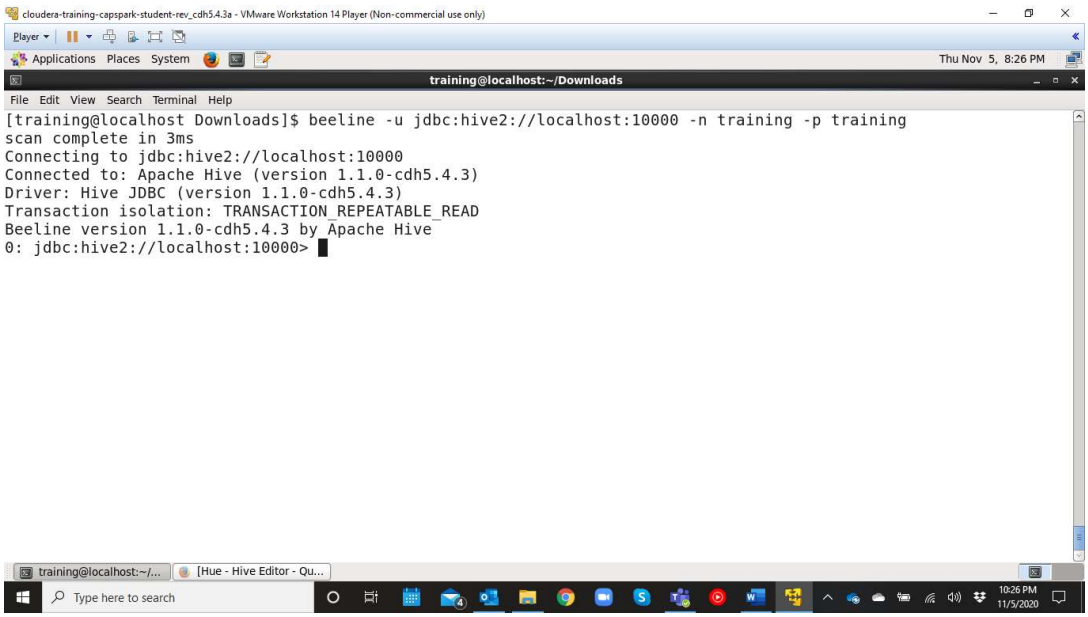
```
STORE final_tweets INTO
'/Sentiment_Analysis_Chirag_Hamirani/Analyzed_Tweets';
```



Explanation: We are viewing the output of the Analyzed_Tweets file in HUE.

Step-18: Execute the following command at the shell prompt:

```
sudo service zookeeper-server start
sudo service hive-server2 start
sudo service hue restart
beeline -u jdbc:hive2://localhost:10000 -n training -p training
```

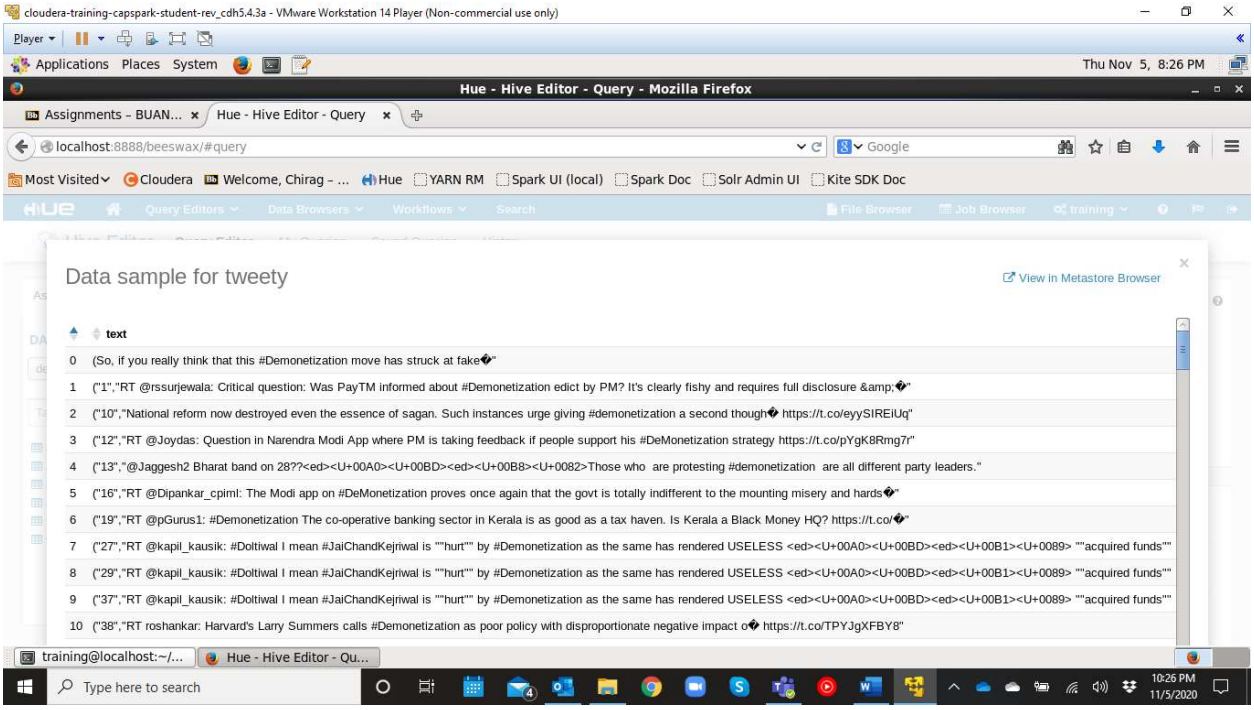


Explanation: We are starting up beeline, to execute the hive commands. Beeline shell is an interactive shell based on the SQLLine utility.

Step-19: Execute the following command at the beeline shell:

```
CREATE EXTERNAL TABLE tweety (text STRING, tweet_rating DOUBLE) ROW FORMAT DELIMITED FIELDS TERMINATED BY ')' LOCATION
'Sentiment_Analysis_Chirag_Hamirani/Analyzed_Tweets';
```

Open the Hive Query Editor and click on the refresh button in the database panel, then click on the 'Preview Sample Data' button next to the tweety table. Take a screenshot of the sample data and paste it below.

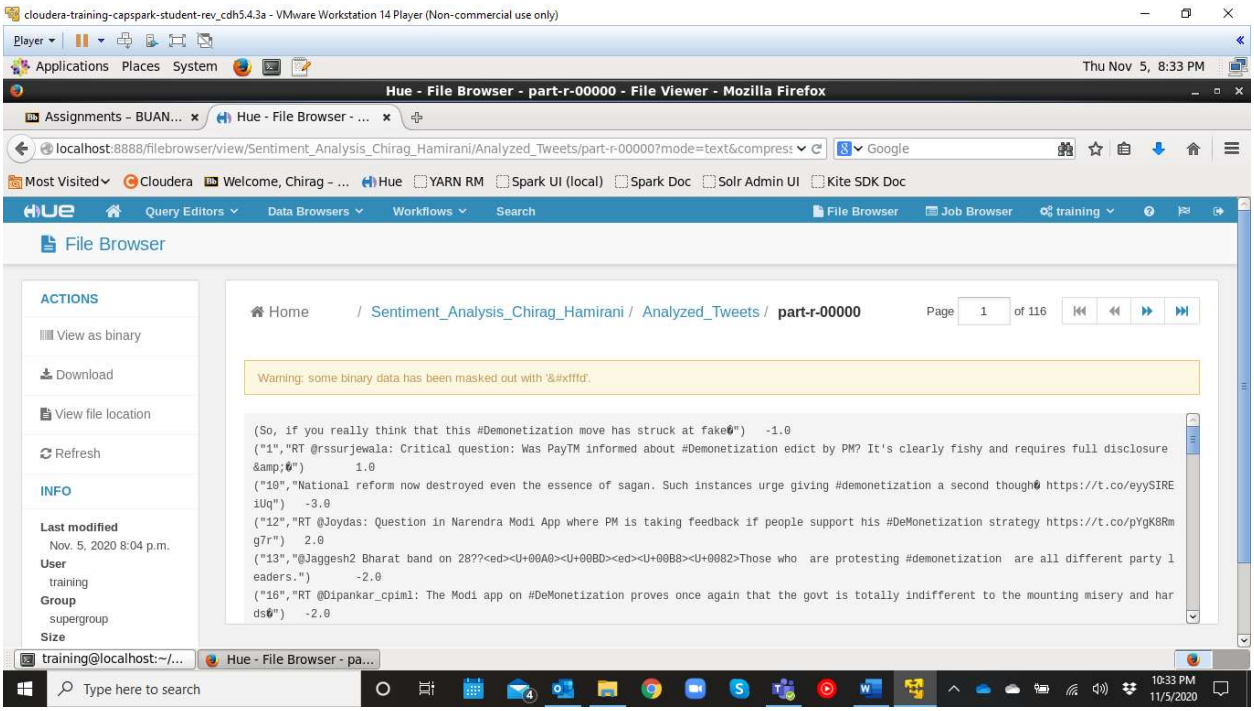


Explanation: We are creating a hive table to structure our unstructured data.

Step-20: Execute the following command at the beeline shell:

```
INSERT OVERWRITE LOCAL DIRECTORY 'Sentiment_Analysis_Chirag_Hamirani/Analyzed_Tweets' ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' select * from tweety;
```

Open Hue and view the file we just created, click on "View as text" on the left under ACTIONS. Take a screenshot of the contents and paste it below.



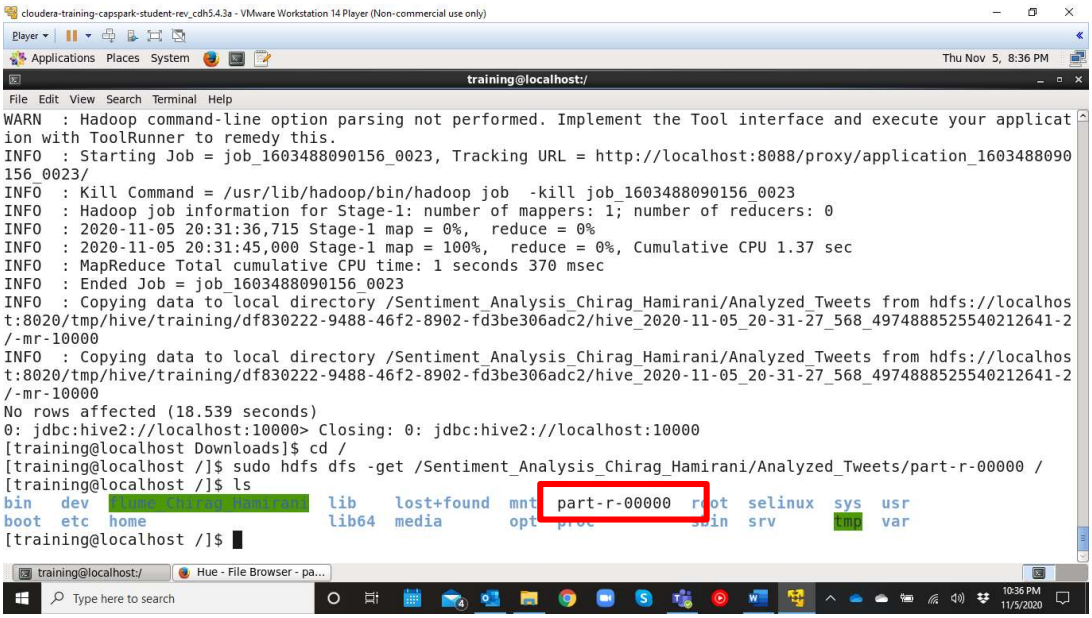
Explanation: We are now creating a tsv file from the table that we created in Step-20.

Part-3: Visualization of Sentiments

In this section we will visualize the results of our sentiment analysis using Jupyter notebooks.

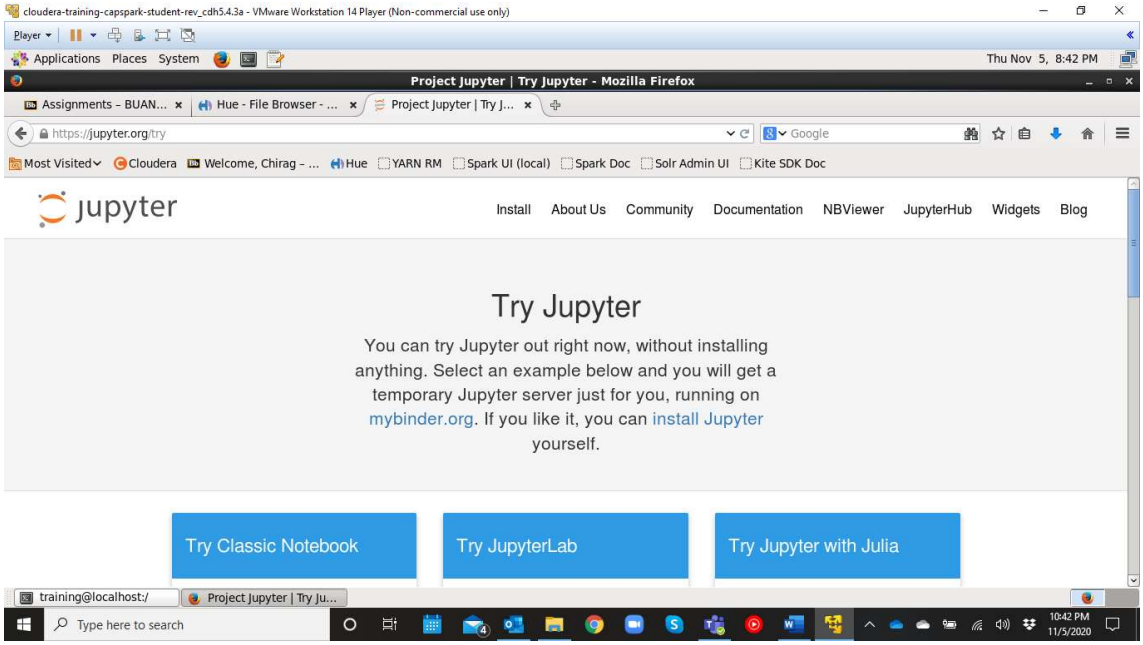
Step-21: Exit Beeline then execute the following command at the shell prompt:

```
cd/
sudo hdfs dfs -get /Sentiment_Analysis_Chirag_Hamirani/Analyzed_Tweets/part-r-00000 /
ls
```

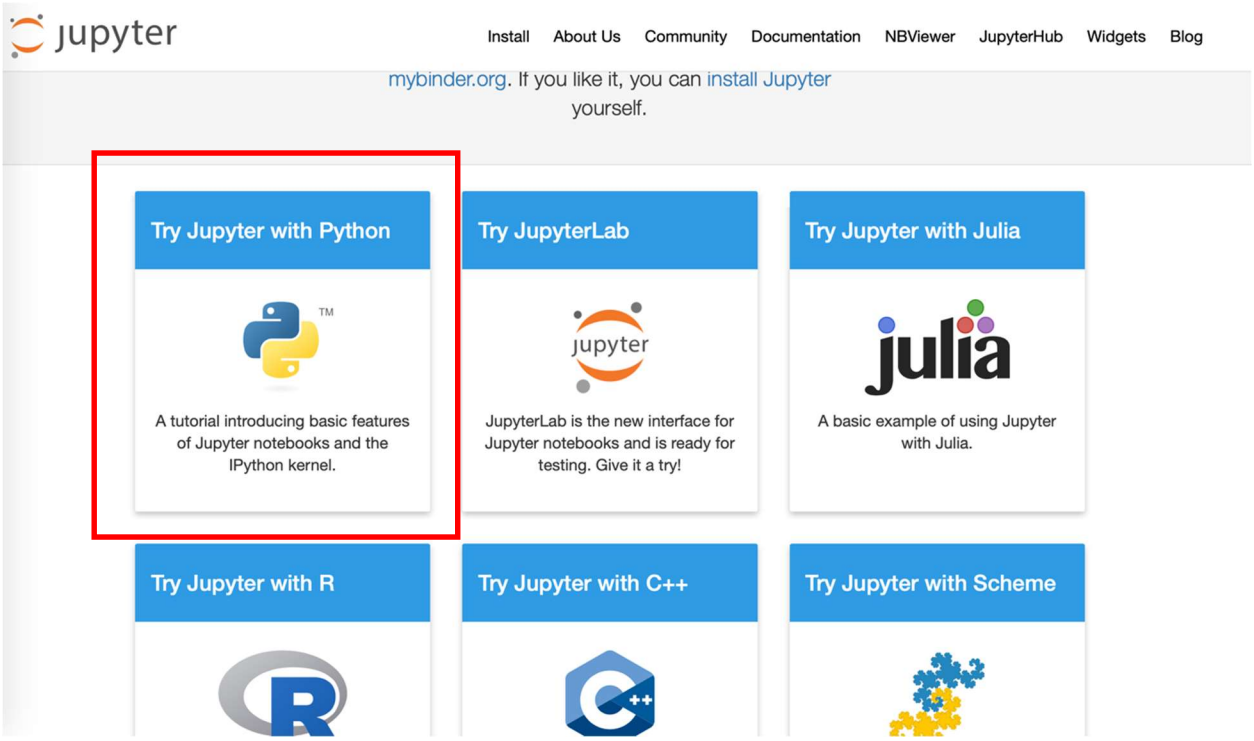


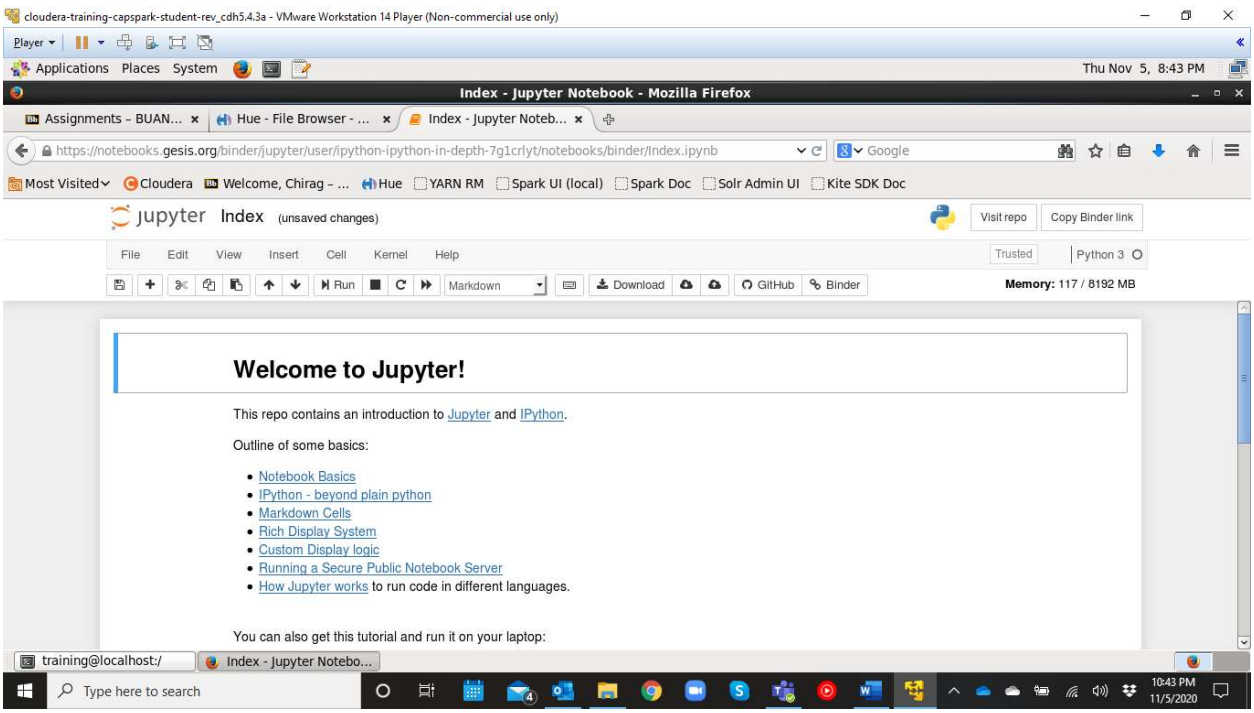
Explanation: We are copying the results of our sentiment analysis back on the local file system to visualize.

Step-22: Using Firefox, browse to the following address: https://jupyter.org/try

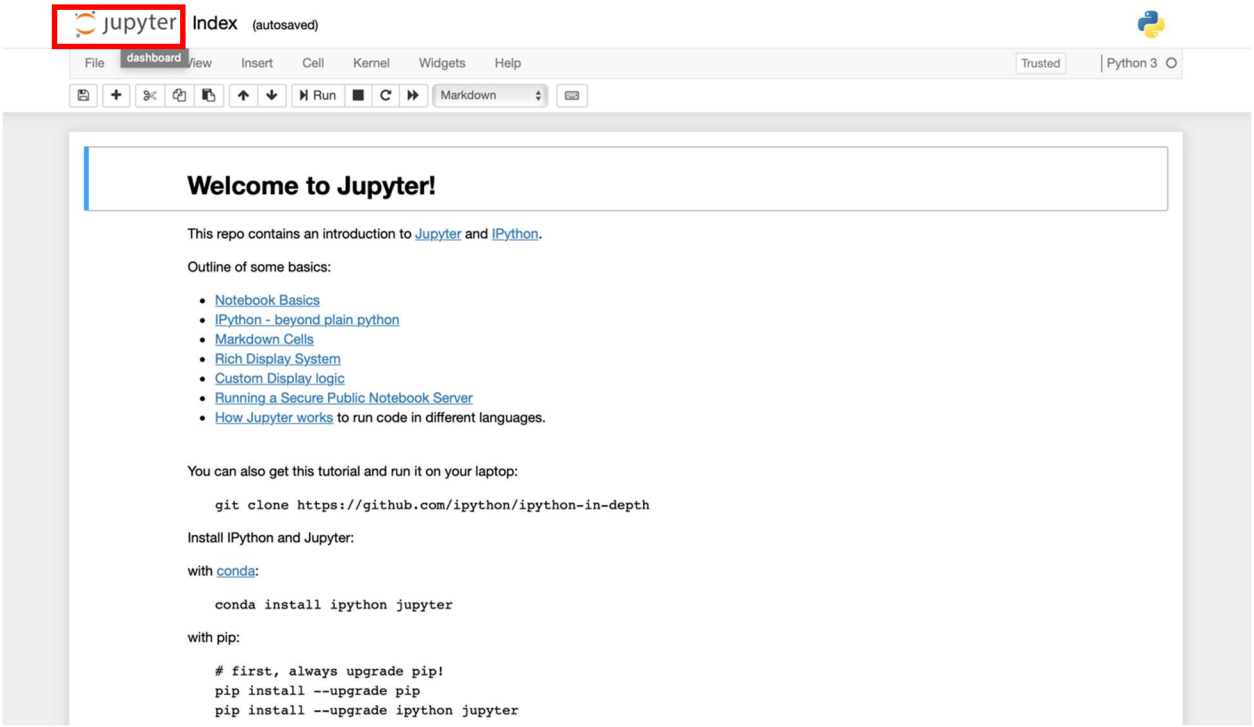


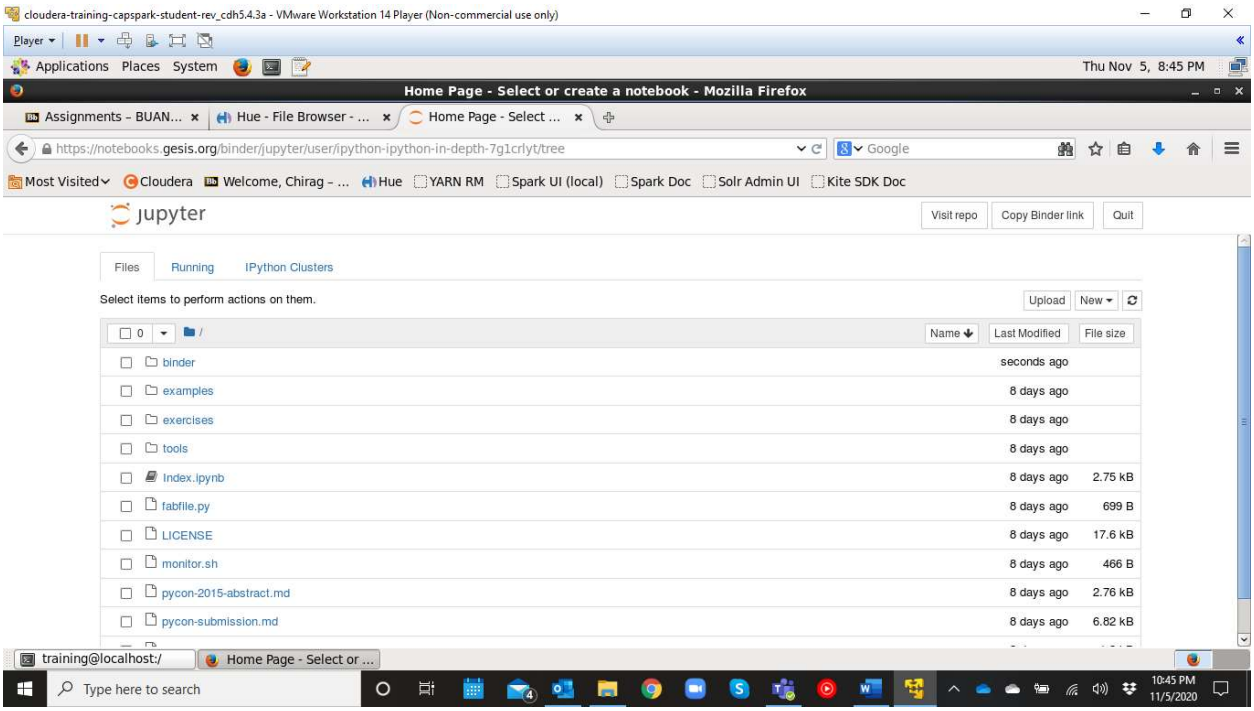
Step-23: On the Jupyter homepage click on the “Try Jupyter with Python” icon that is highlighted in the screenshot below:



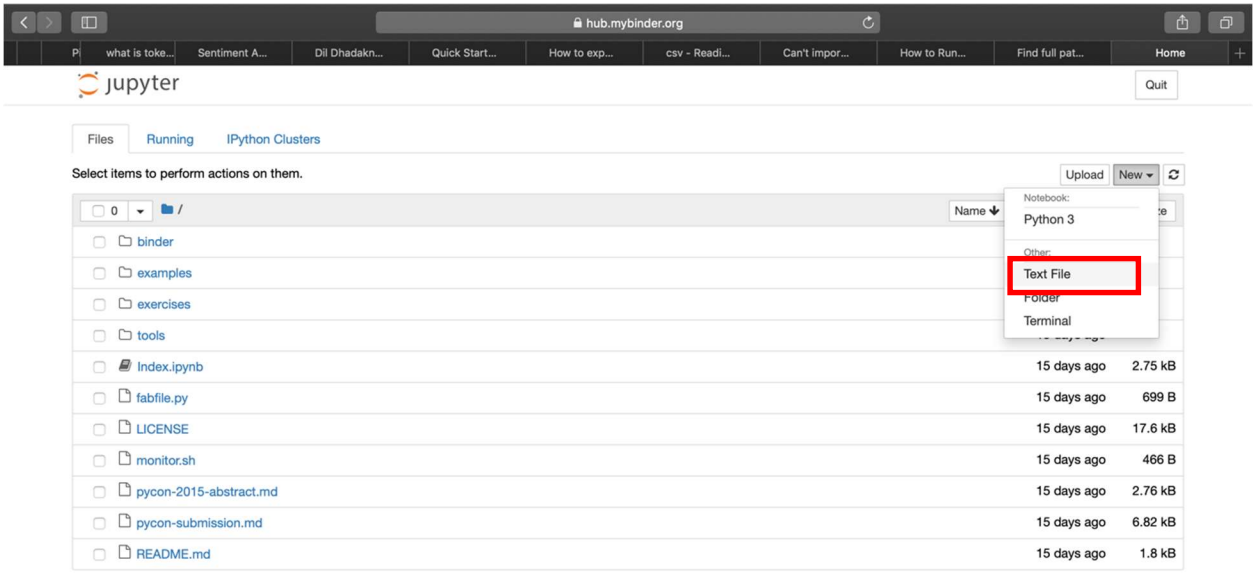


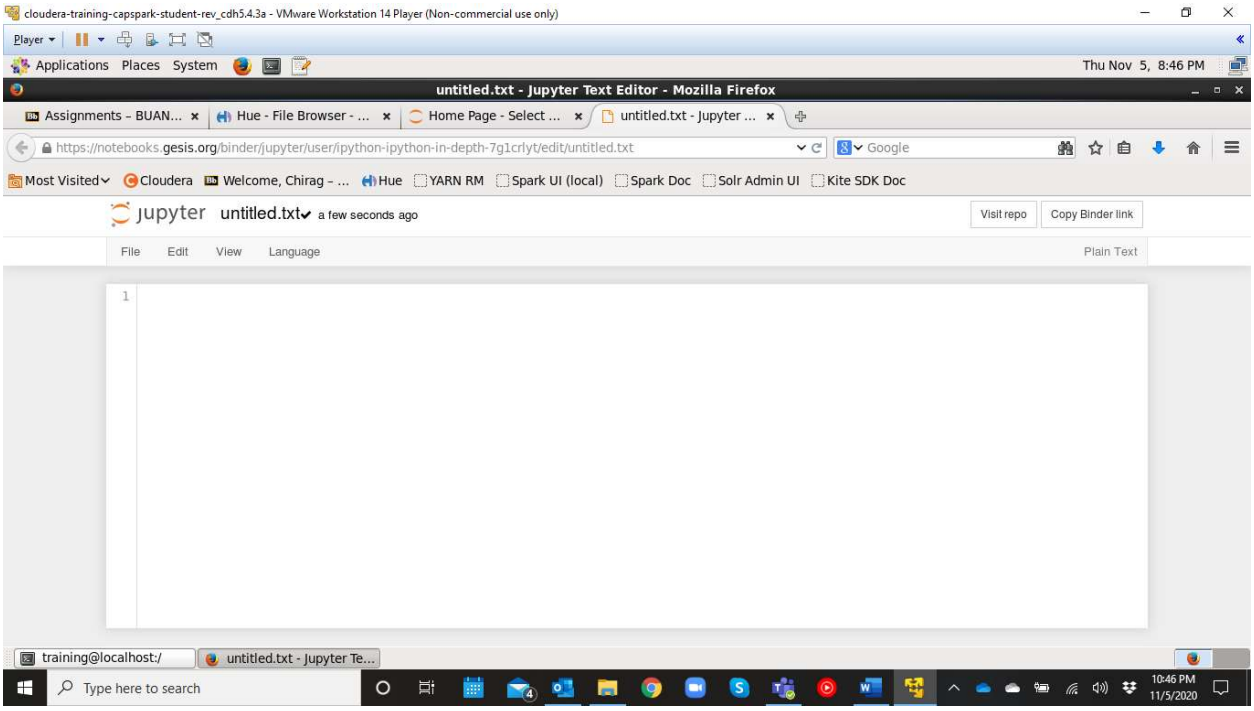
Step-24: Now click on the homepage button highlighted in the screenshot below.



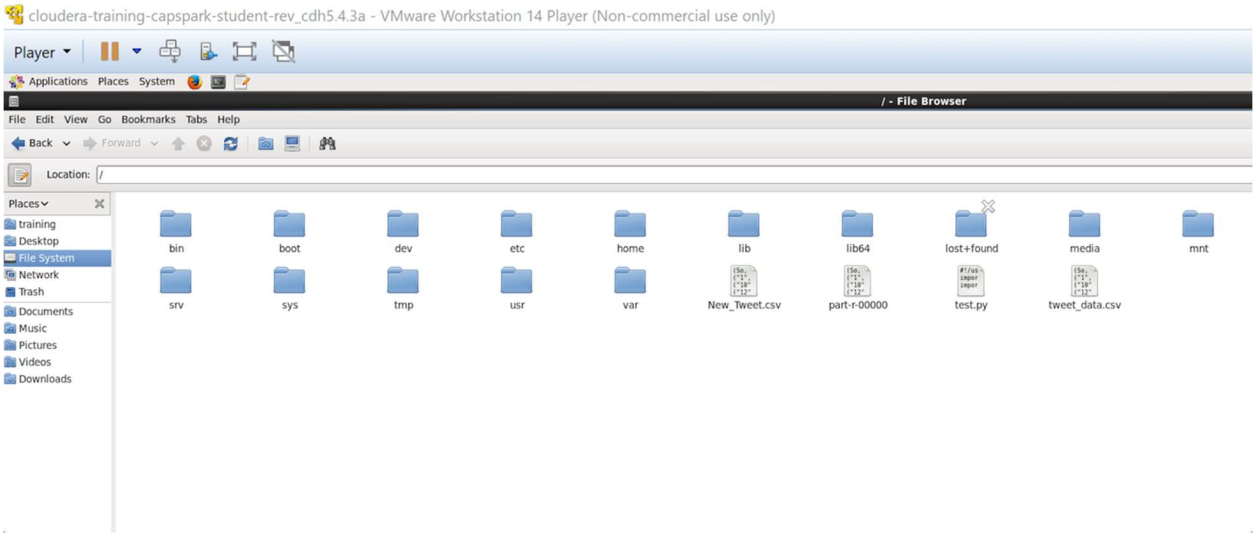


Step-25: Now click on the “Text File” button highlighted in the screenshot below.



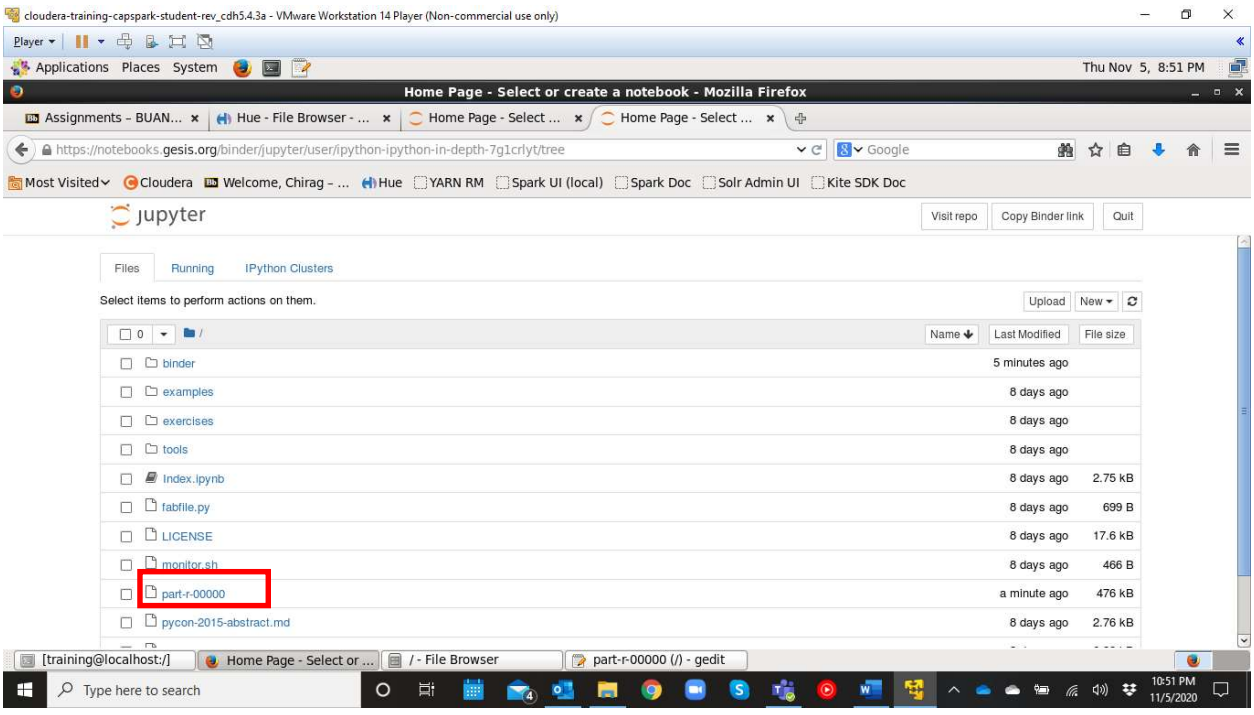


Step-26: Now open the file containing the calculated sentiments in the computer application. After opening the computer application click on the filesystem folder highlighted in the screenshot below.

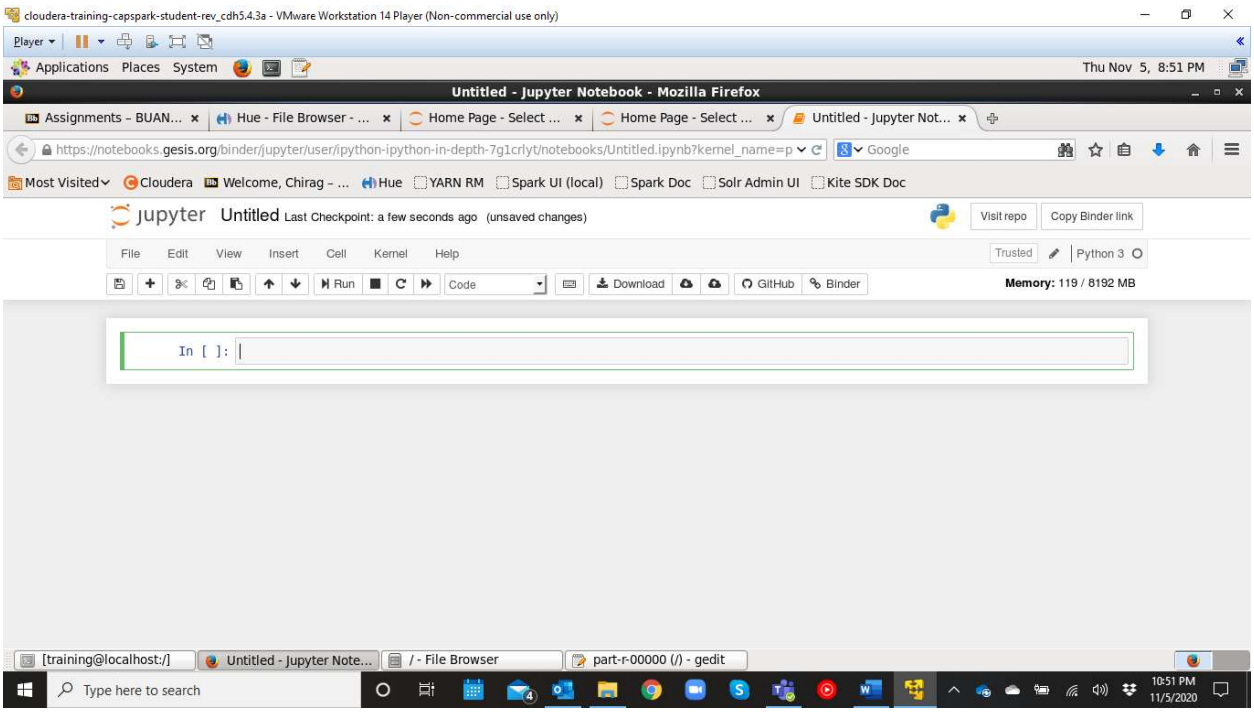


Open the part-r-00000 and copy the contents and paste it in the browser running Jupyter we opened in Step-25. Save the file as “part-r-00000”. Force save the file.

Click on the Jupyter icon like in Step 24 and highlight the new file you just created.



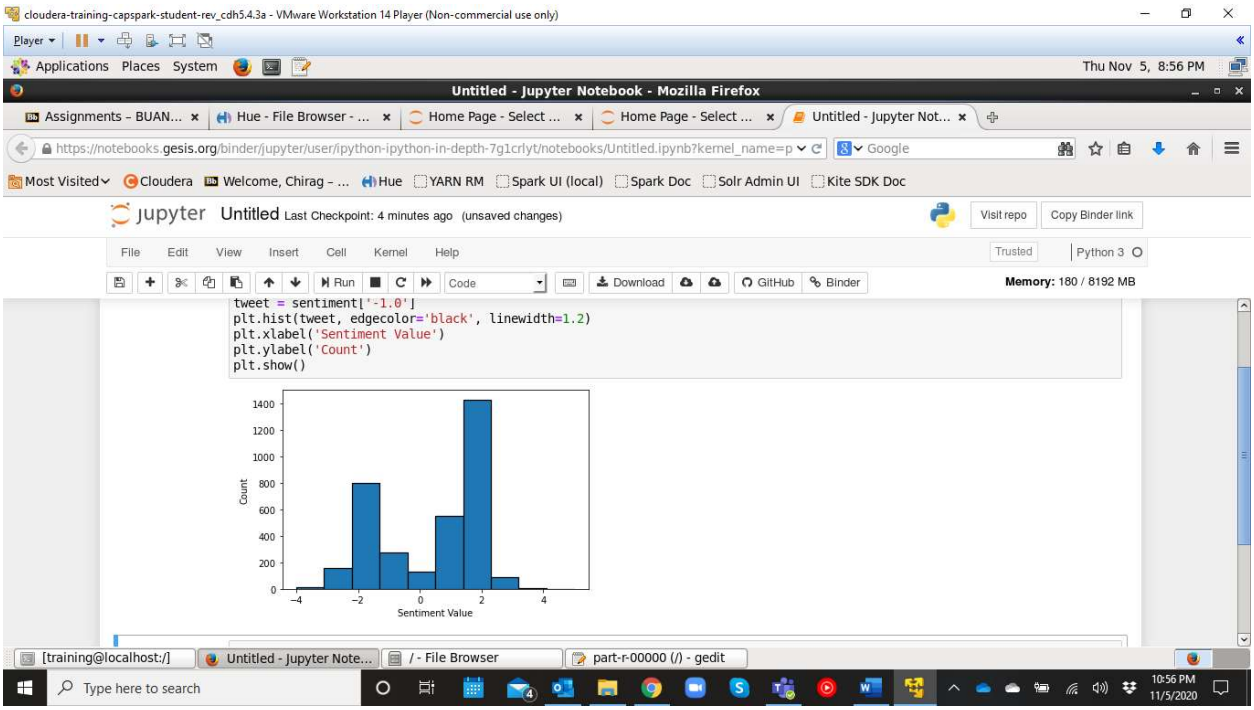
Step-27: Now create a Python 3 Jupyter Notebook file by clicking on the “Python 3” button highlighted in the screenshot below.



Step-28: Now type the code in the screenshot below at the prompt (after In [ ]:) then hit Run.

```
import matplotlib.pyplot as plt
import pandas as pd
sentiment = pd.read csv('part-r-00000', delimiter = '\t')
tweet = sentiment['-1.0']
plt.hist(tweet, edgecolor='black', linewidth=1.2)
plt.xlabel('Sentiment Value')
plt.ylabel('Count')
plt.show()
```

Conclusion



The tweets seem to be divided in two normal distribution groups on both sides, positive and negative. Tweets with highest frequency are moderate negative (-2) and moderate positive (+2). However, the distribution clearly shows that more tweets are assessed as positive. Note that number of tweets with positive 1 and 2 are almost double compared to tweets with negative 1 and 2 respectively. This clearly shows that most people (out of given sample) are in favor of demonetization despite certain unrest.