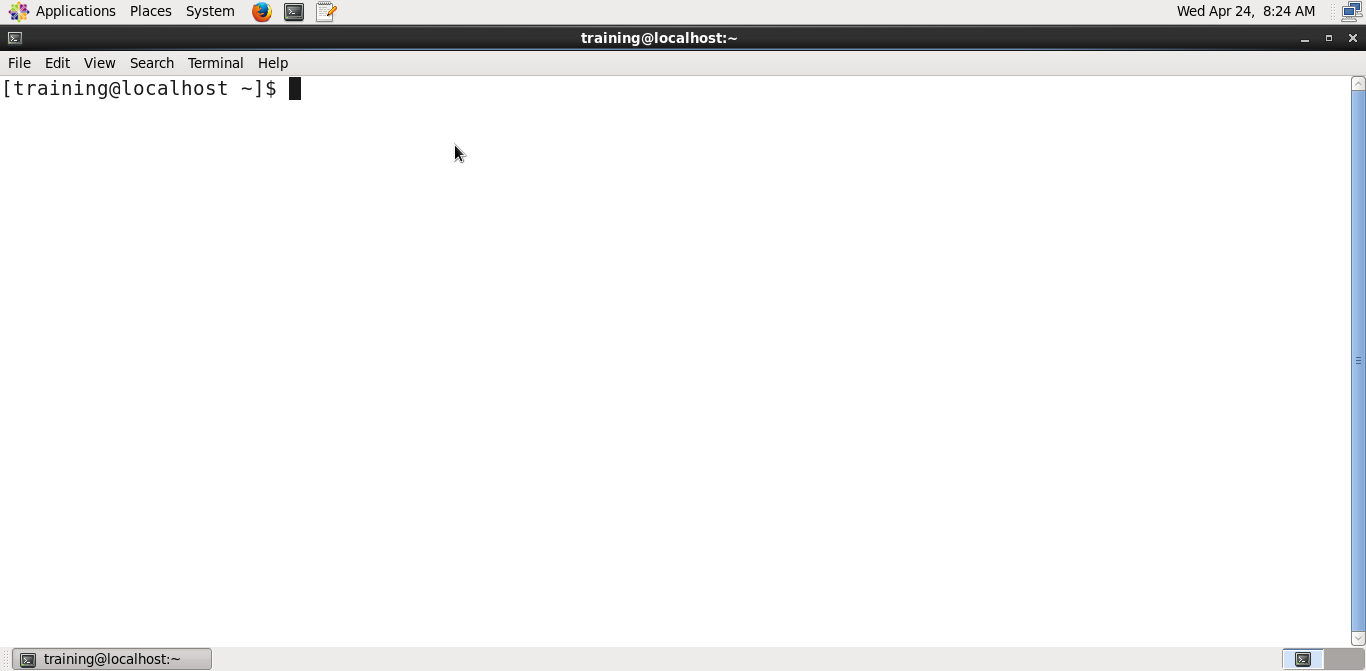
Big Data – Case Study

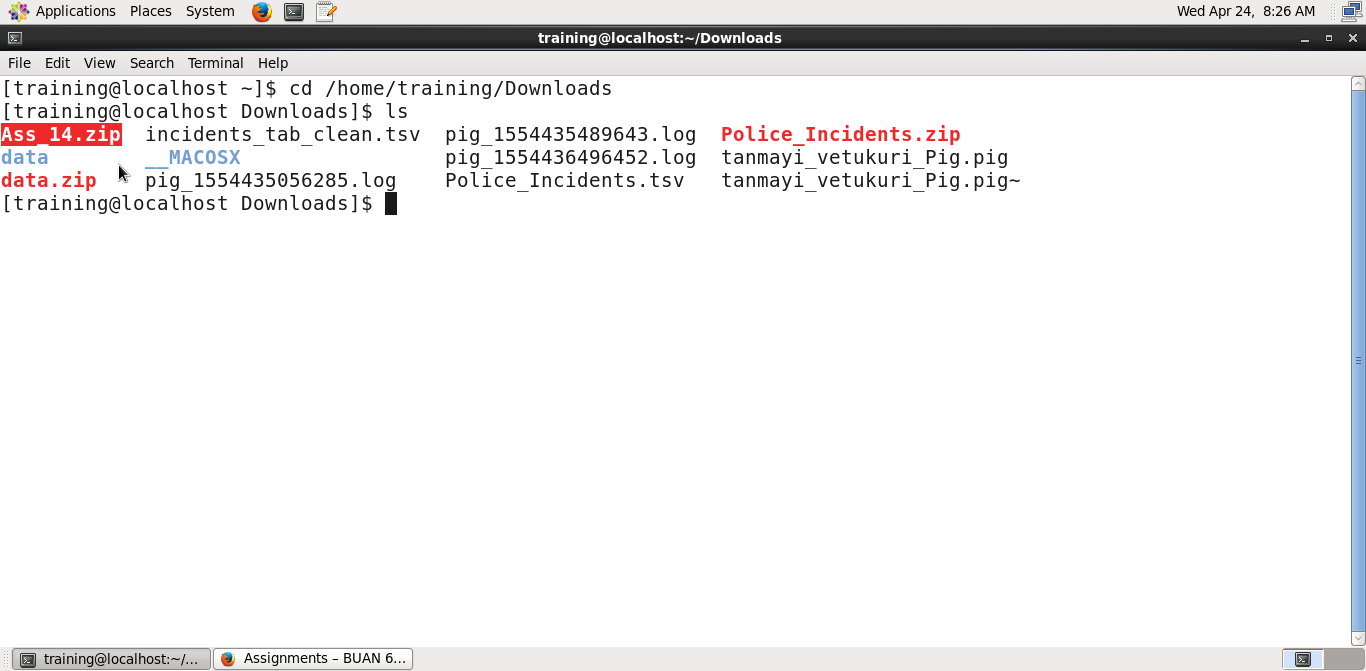
Sentiment Analysis on tweets regarding demonetization in India

Framework: Hadoop – HDFS, PIG, HIVE Programming Language: Python

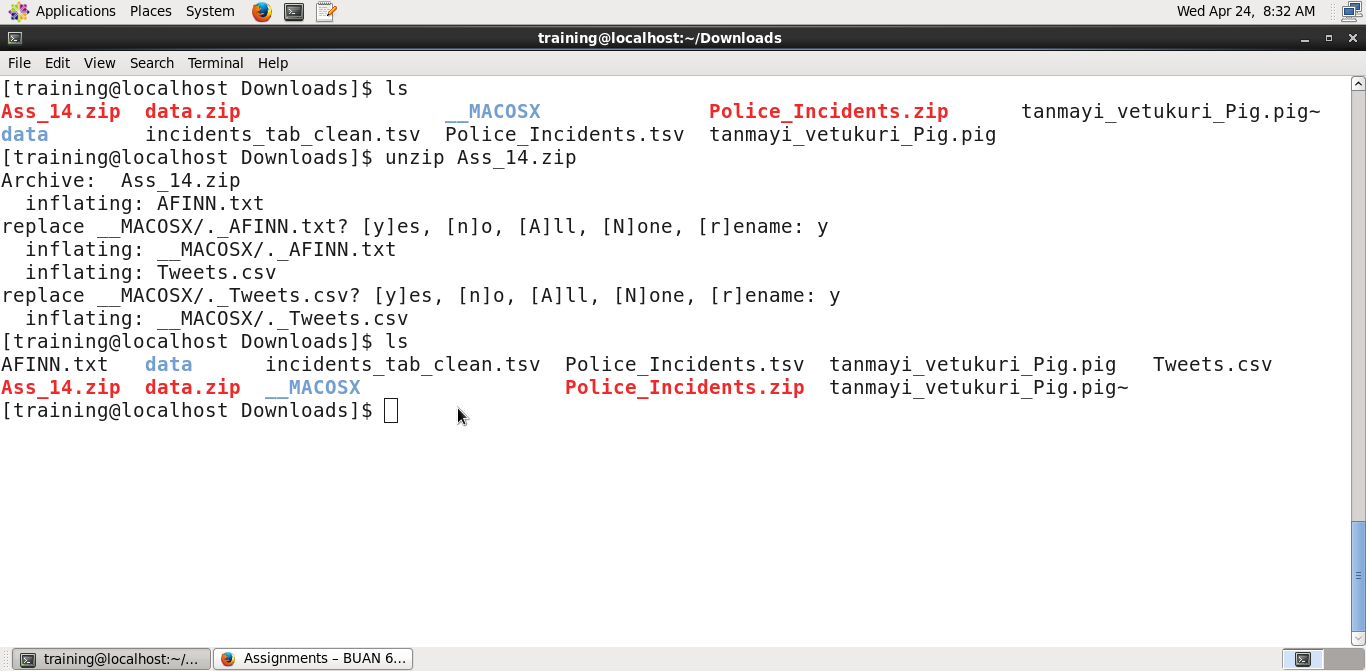
Step 1: Opened the terminal application on the Cloudera VM desktop



Step 2: Downloaded the required file to analyze tweets

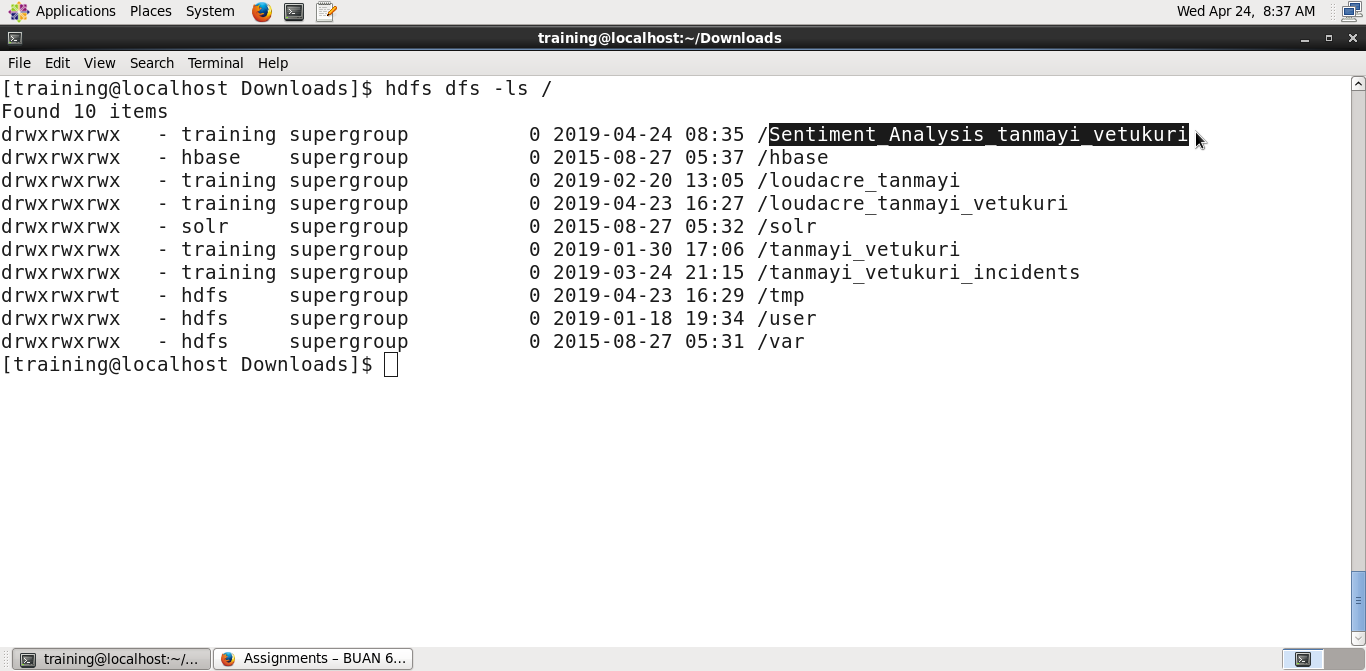


Step 3:

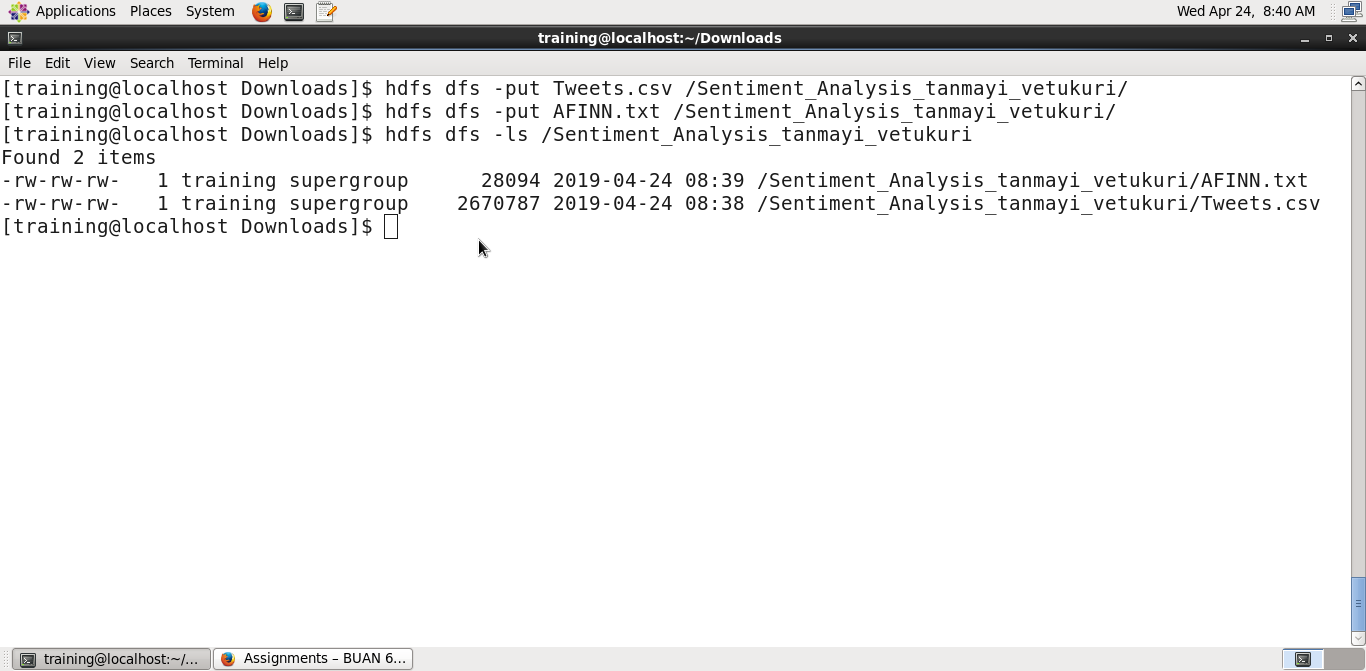




Step 4: Created directory to be used for case study and moved files with tweets into this directory

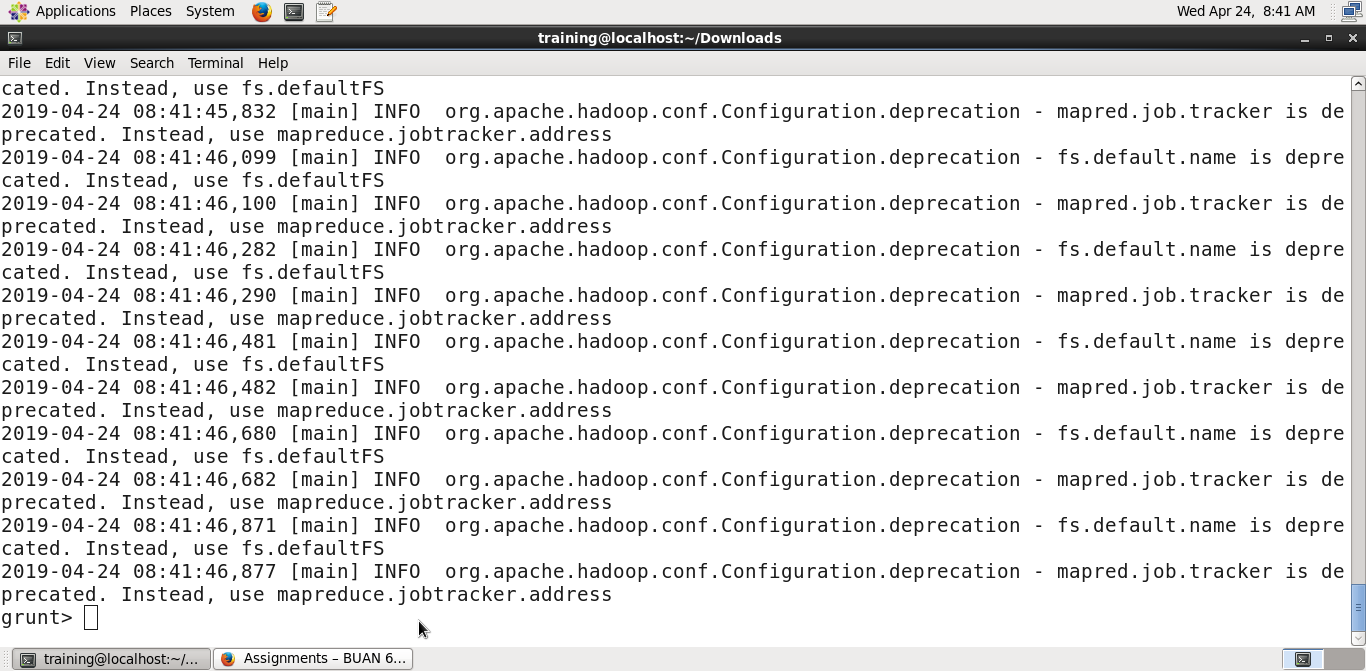


Step 5:

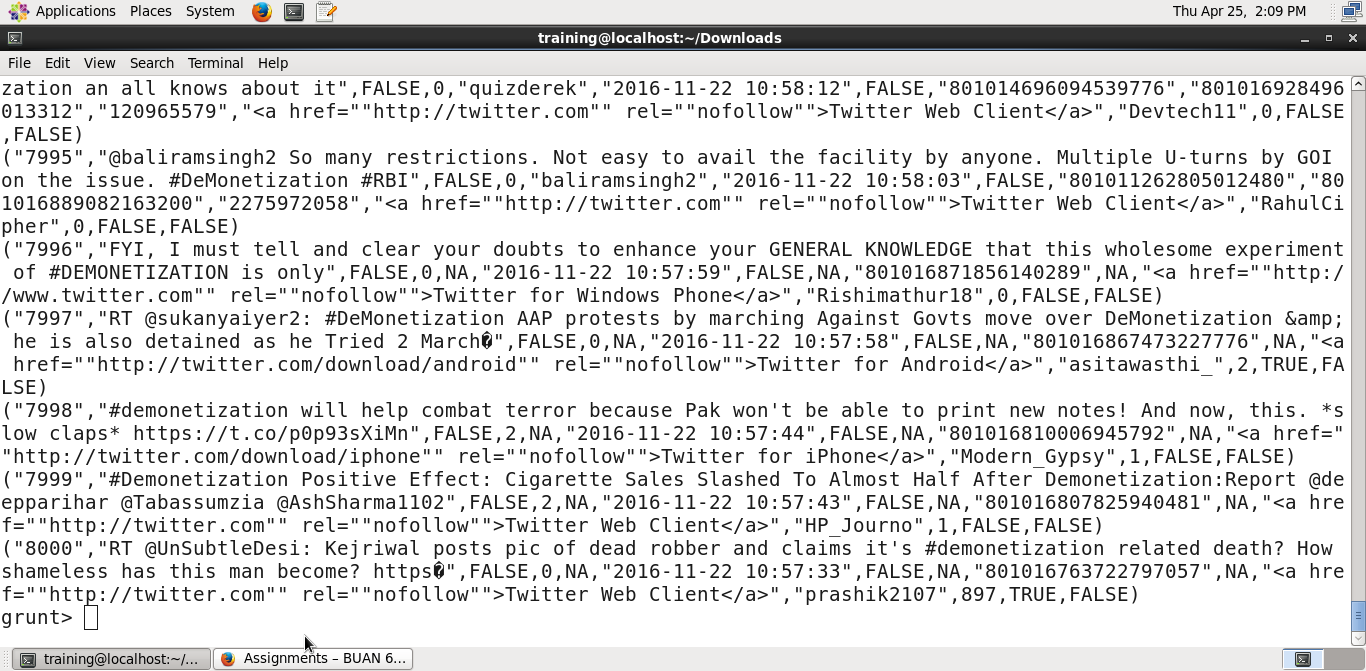




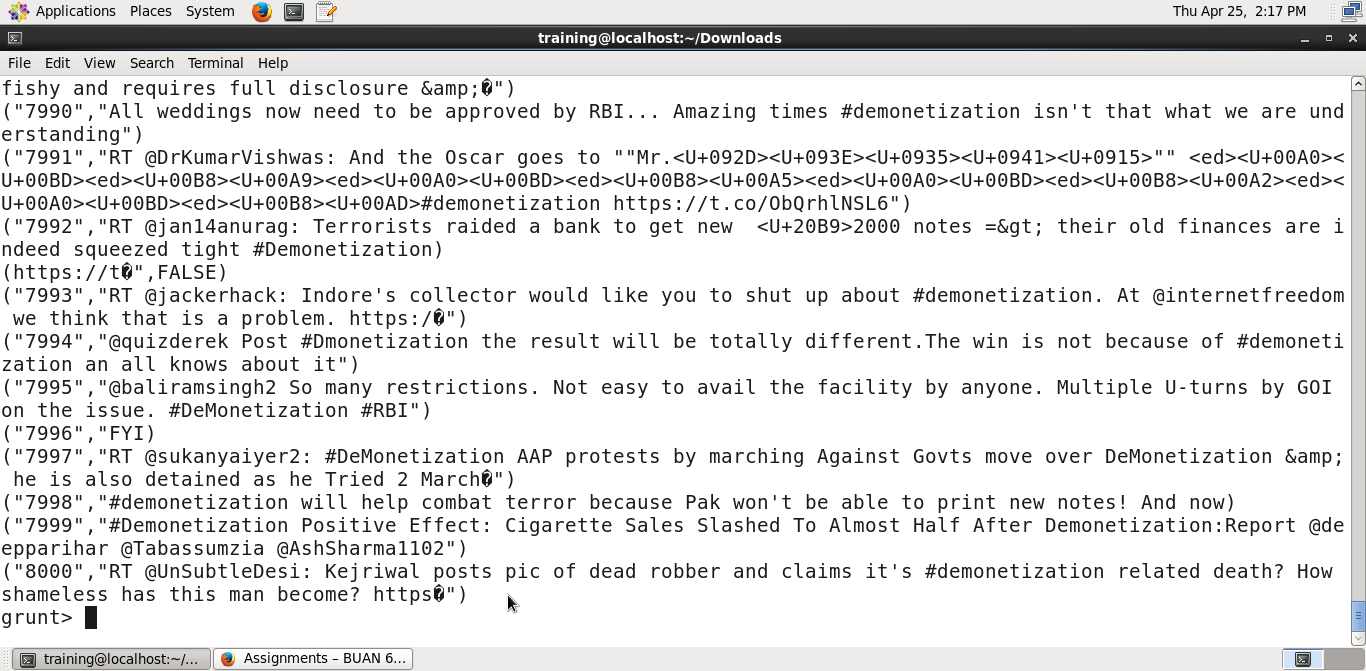
Step 6: Executed PIG command to open grunt shell



Step 7: Loaded Tweets.csv file to Demonit\_Tweets and displaying data in Demonit\_Tweets

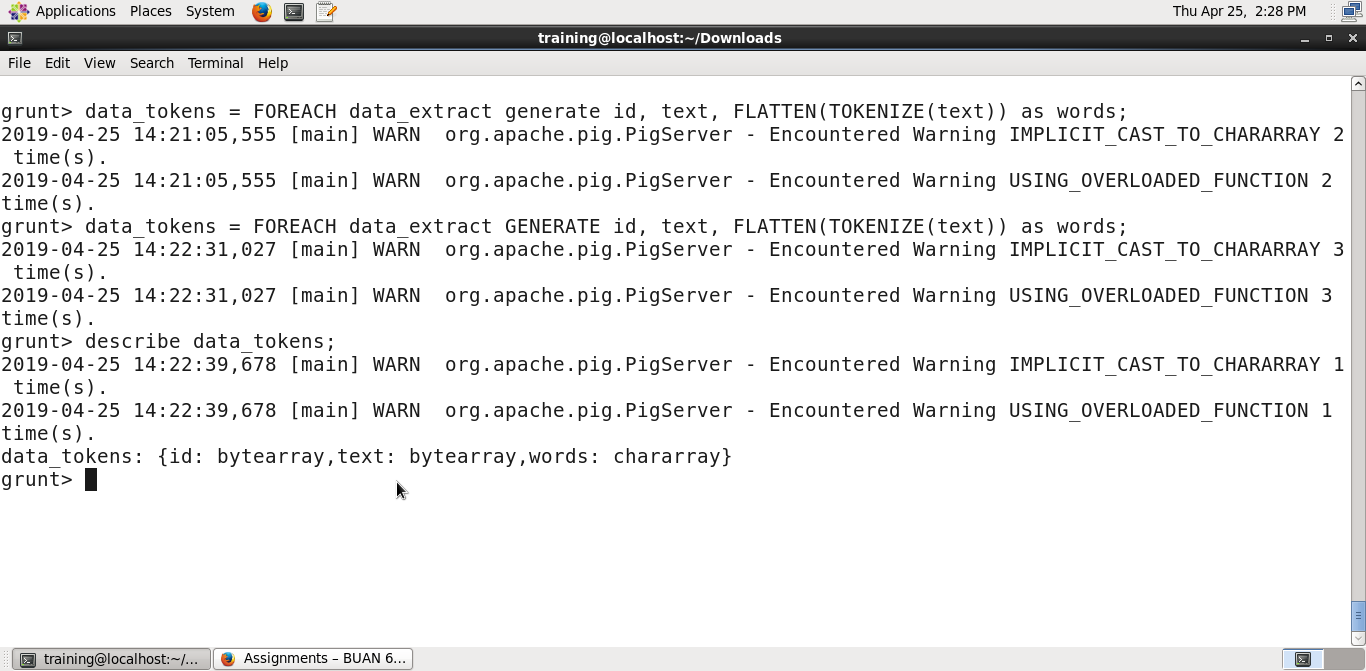


Step 8: Defining and collecting the first two data elements in the csv file and dumping the data at the grunt shell. Command used -> data\_extract = FOREACH Demonit\_Tweets GENERATE $0 as id, $1 as text;



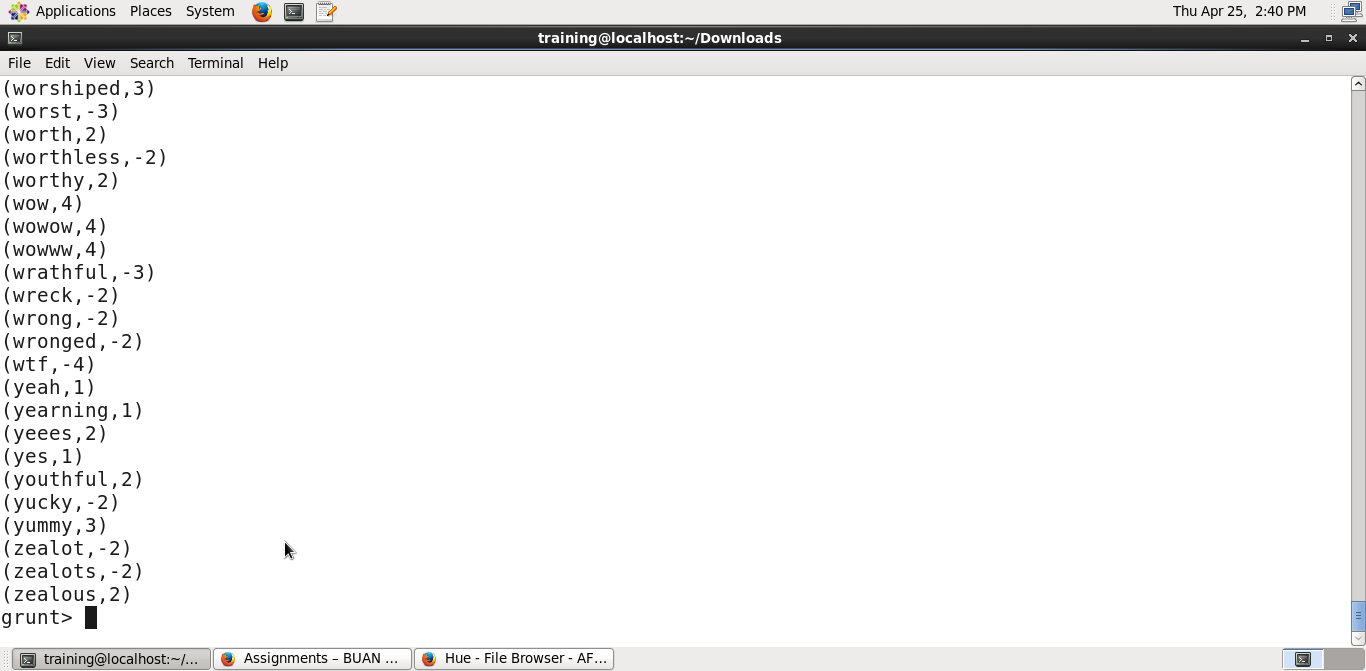
Step 9: Defining a third element by breaking down the text information in the variable ‘text’ into words

Code -> data\_tokens = FOREACH data\_extract generate id, text, FLATTEN(TOKENIZE(text)) as words;



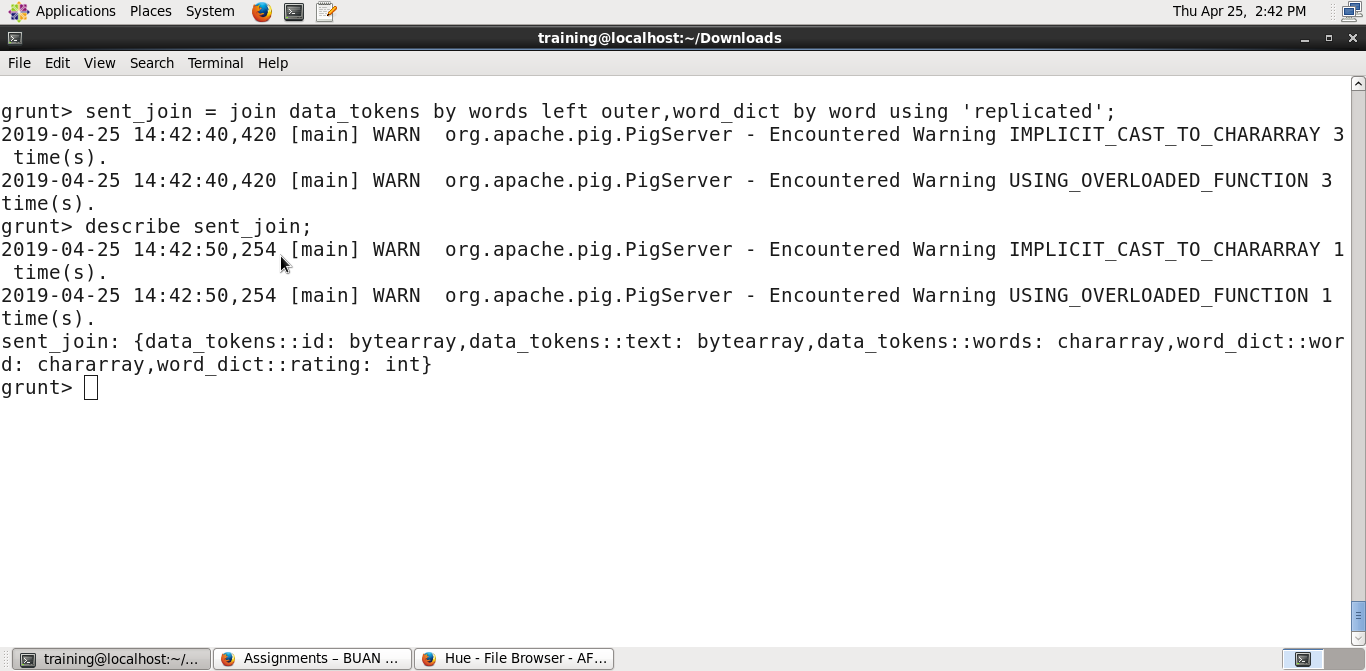
Step 10: AFINN dictionary is a dictionary consisting of 2500 words that are rated on a scale from -5 to +5 depending on the sentiment the word conveys. Used this dictionary to perform the sentiment analysis on tweet data.

Code -> word\_dict = LOAD ‘/Sentiment\_Analysis\_tanmayi\_vetukuri/AFFIN.txt’ USING PigStorage(‘\t’) AS(word:chararray, rating:int);

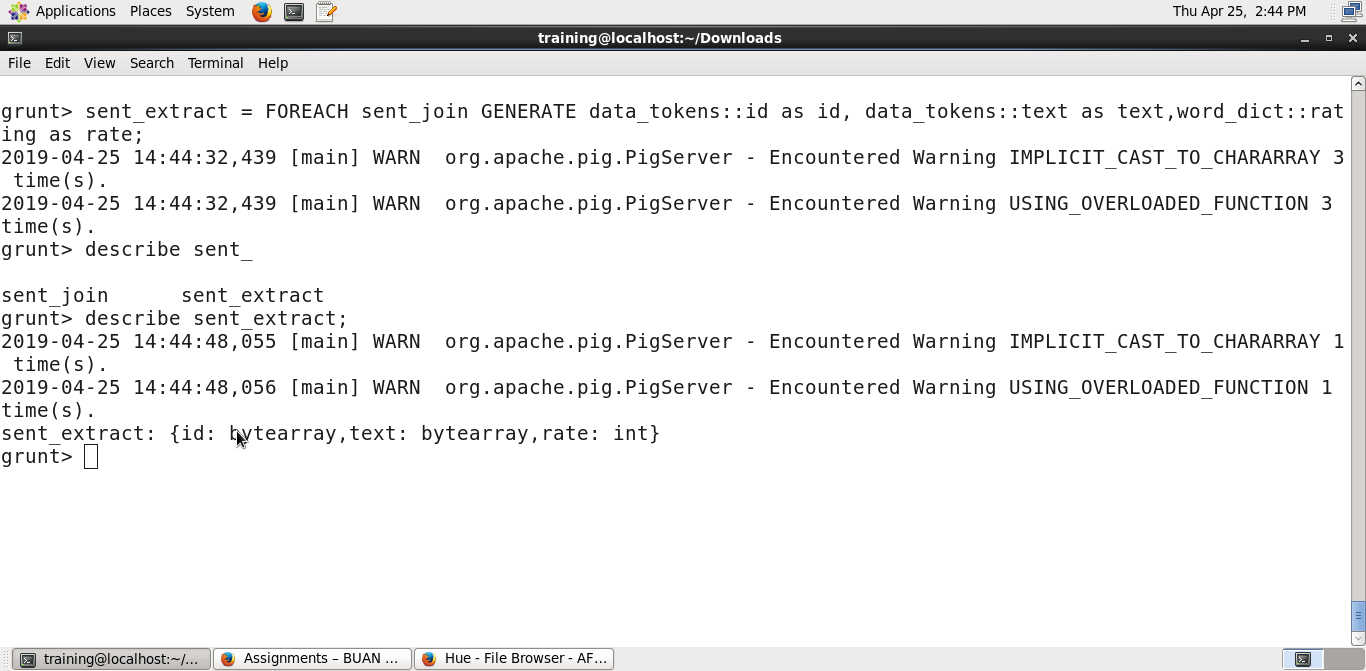


Step 11: Joined sentiments values from the AFINN dictionary with the words in the text from each tweet.

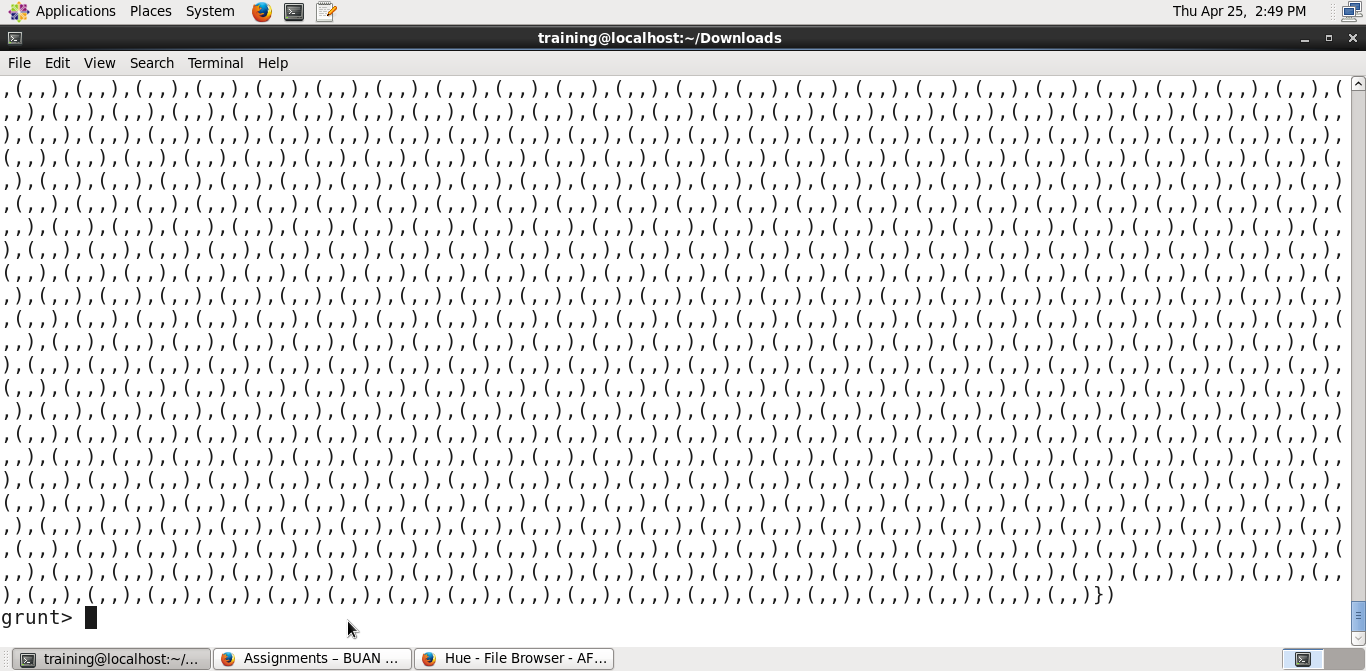
Code -> sent\_join = join data\_tokens by words left outer, word\_dict by word using ‘replicated’;



Step 12: Extracted data from the variable in step 12 i.e. sent\_join that is needed to complete sentiment analysis.

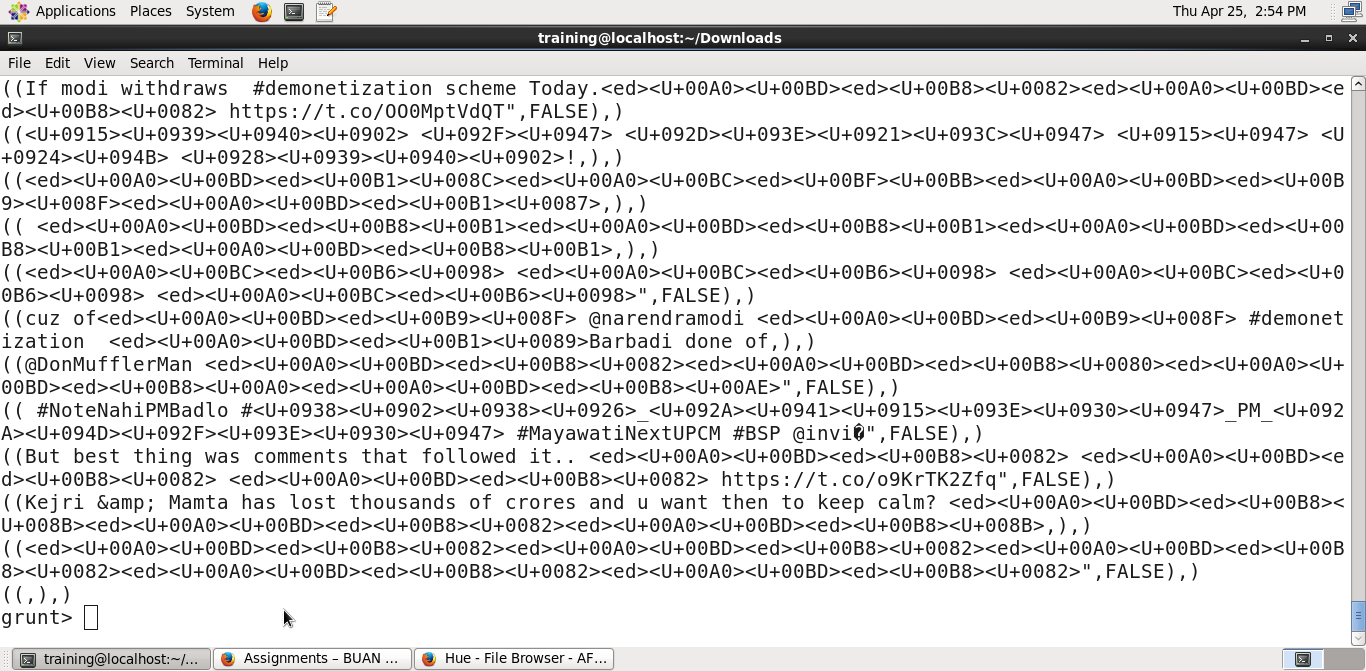


Step 13: Grouped each record by its id and text



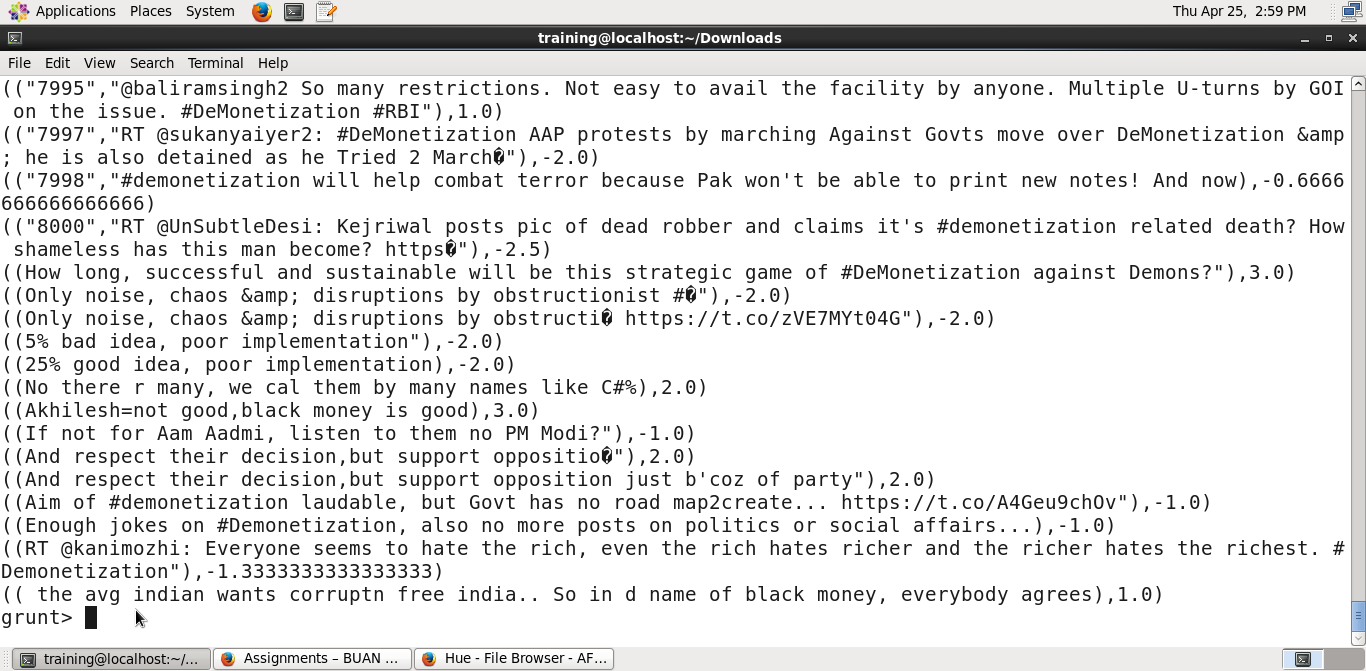
Step 14: Averaging the rating for all the words in each tweet and storing the result in average\_rating.

Code -> average\_rating = FOREACH word\_grouping generate group, AVG(sent\_extract.rate) as tweet\_rate



Step 15: Filtered all the tweets for those tweets that have an average rating >= -5.

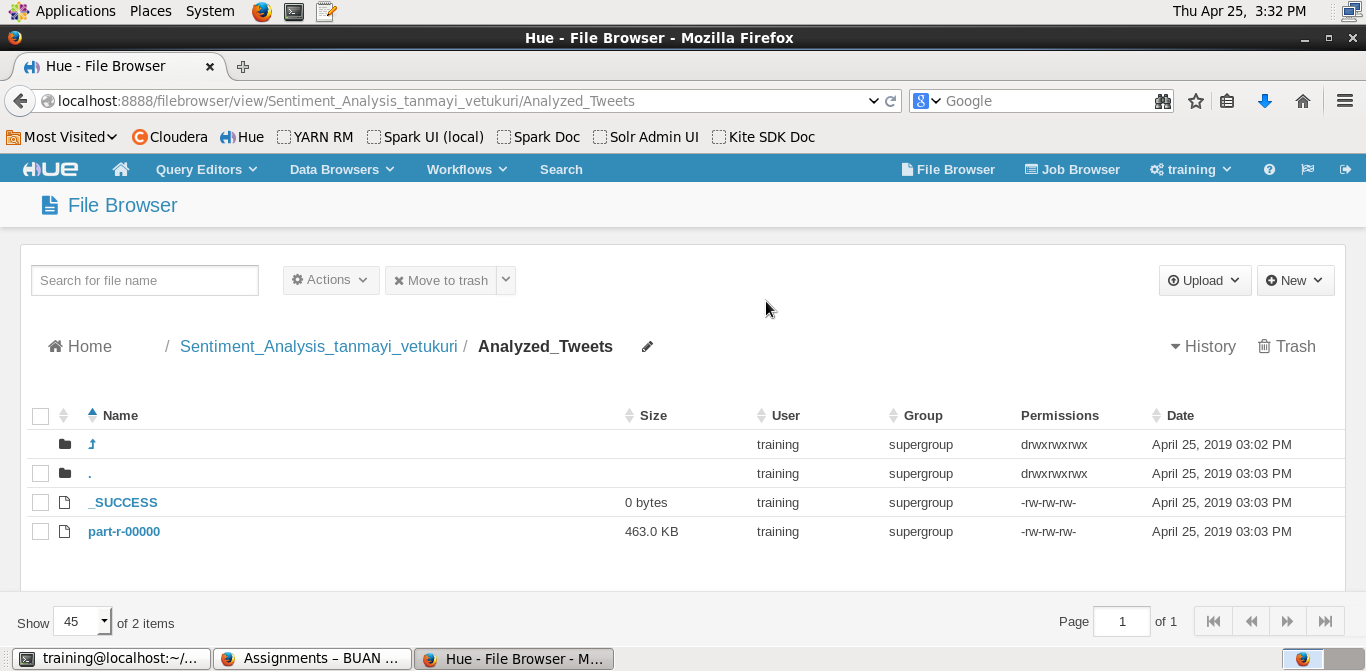
Code -> final\_tweets = filter average\_rating by tweet\_rate >= -5;

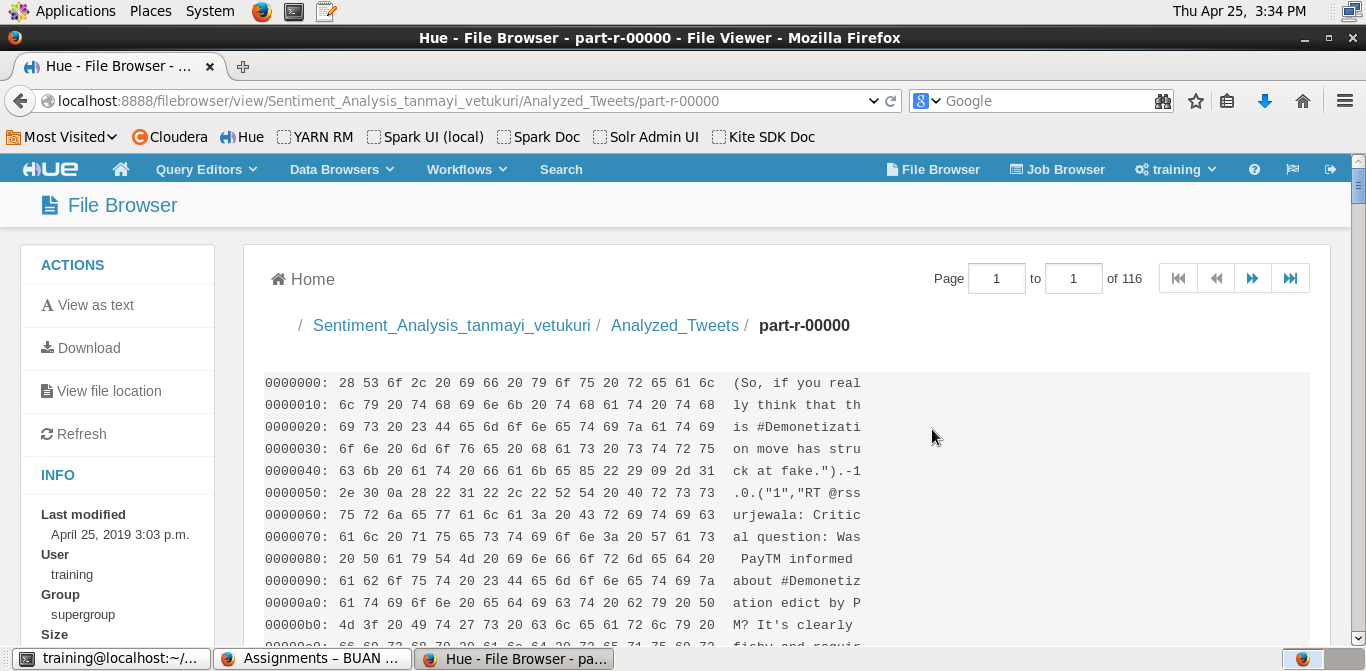


Step 16: Stored final\_tweets into Sentiment\_Analysis\_tanmayi\_vetukuri/Analyzed\_Tweets

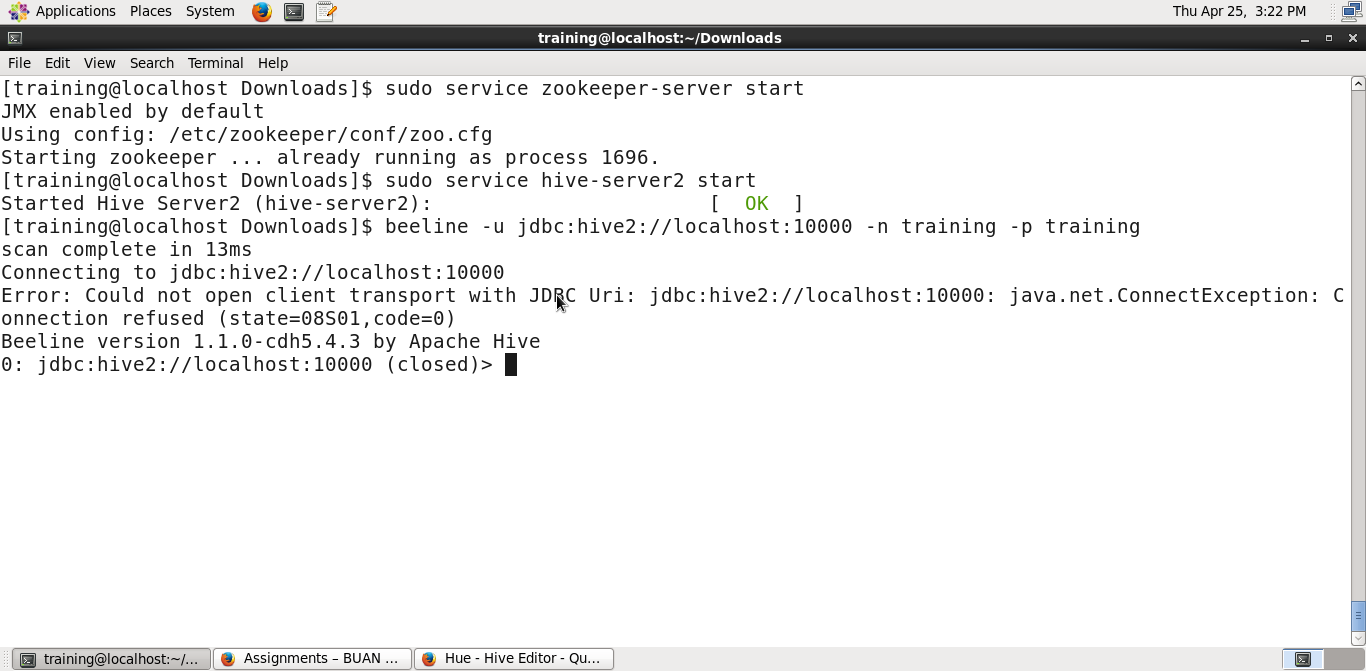
Code -> STORE final\_tweets INTO ‘/Sentiment\_Analysis\_tanmayi\_vetukuri/Analyzed\_Tweets’;

Viewing the output of the Analyzed\_Tweets file in HUE.



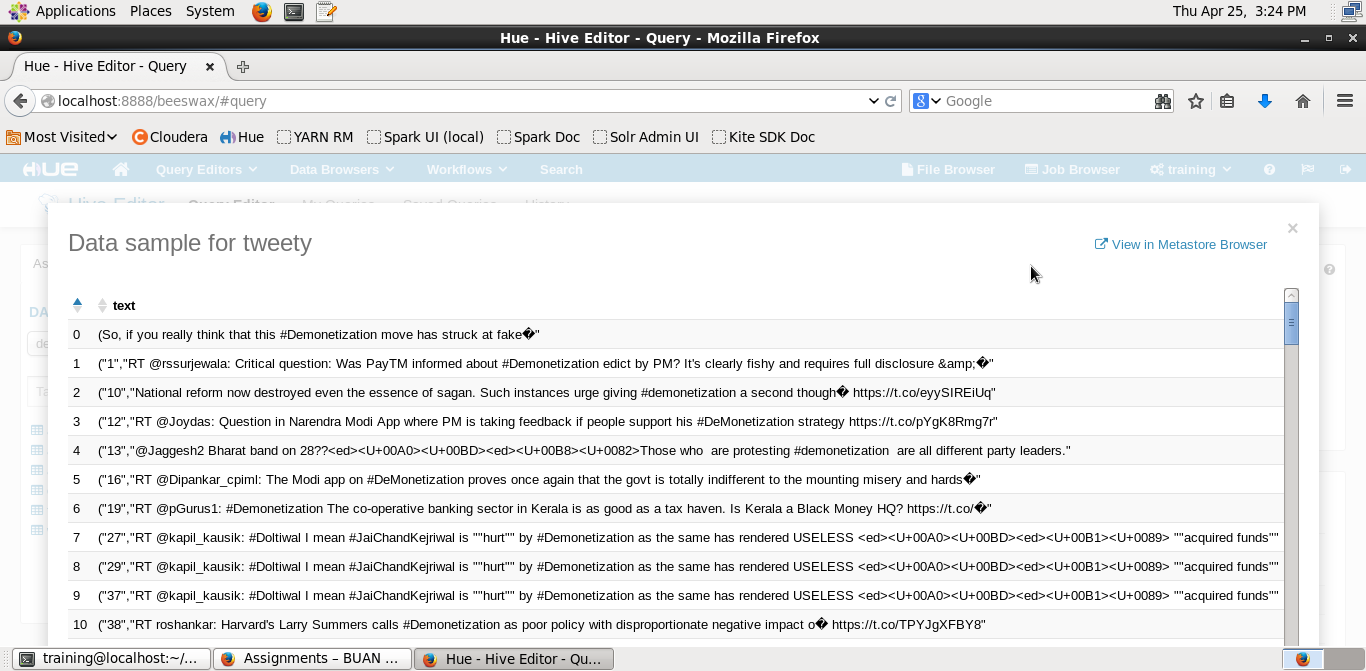


Step 17: Started beeline to execute the hive commands

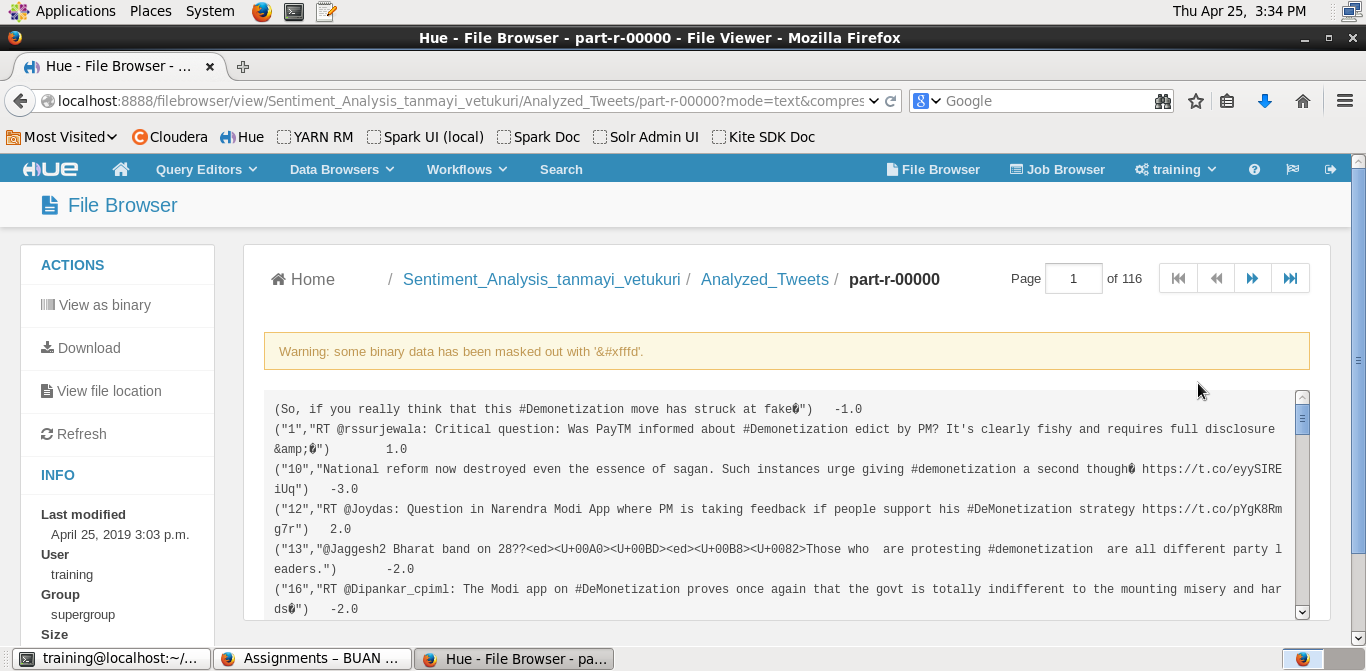


Step 18: Created a hive table to structure the unstructured data

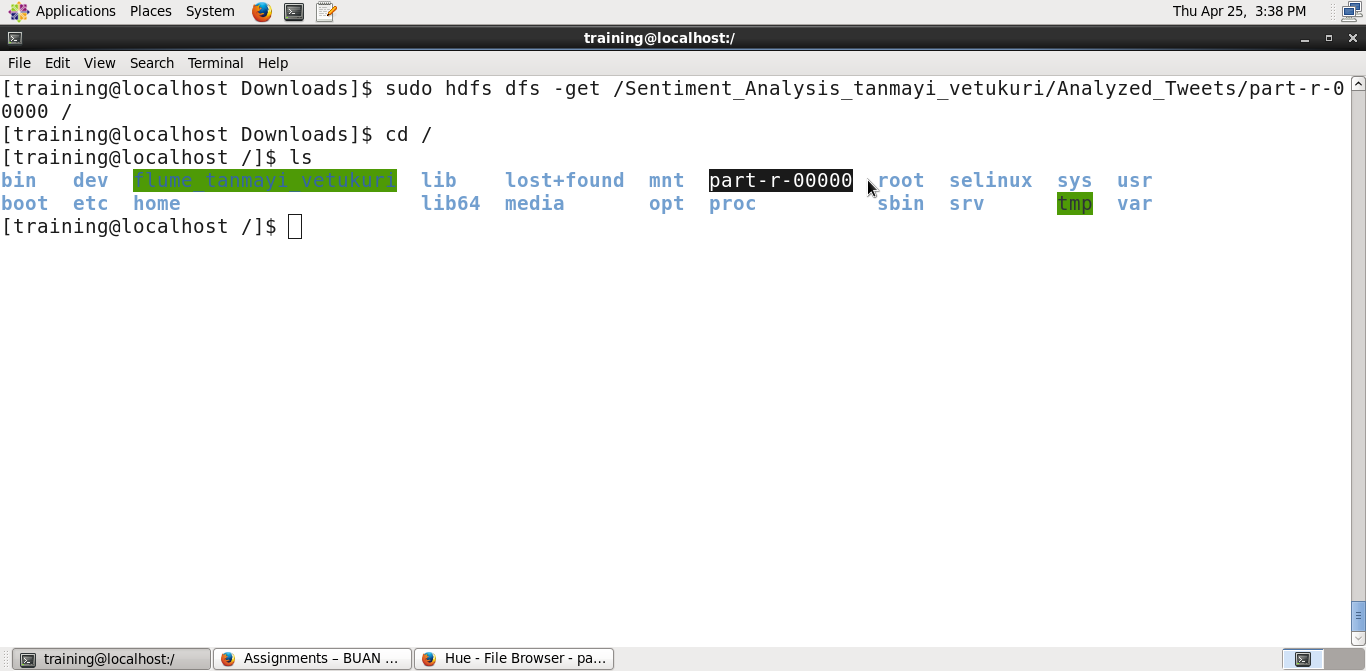
Query -> CREATE EXTERNAL TABLE tweety (text STRING, tweet\_rating DOUBLE) ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘)’ LOCATION ‘/Sentiment\_Analysis\_tanmayi\_vetukuri/Analyzed\_Tweets’;



Step 19: Created a .csv file from the table created in Step-20



Step 20:



Step 21: Created a graph using Python to understand the impact of sentiment score.

Code ->

import matplotlib.pyplot as plt

import pandas as pd

sentiment = pd.read\_csv(‘part-r-00000’, demiliter = ‘\t’)

tweet = sentiment[‘-1,0’]

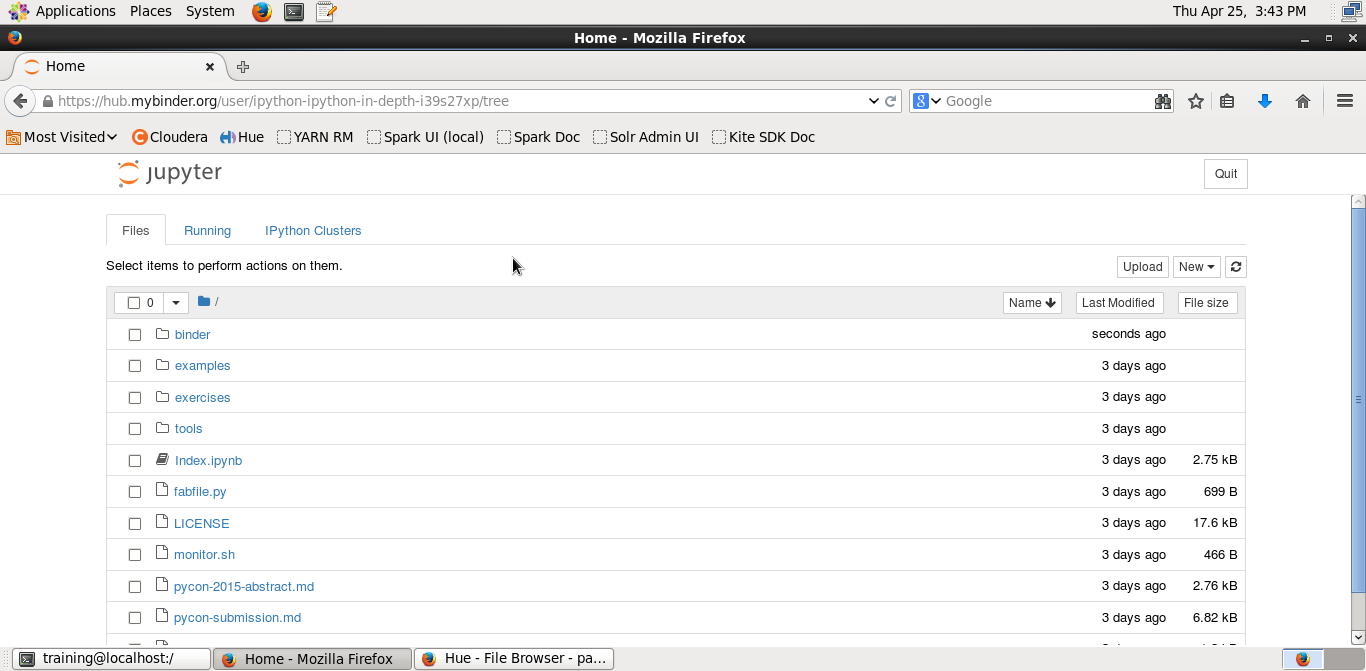
plt.hist(tweet, edgecolor=’black’, linewidth = 1.2)

plt.xlabel(‘Sentiment Value’)

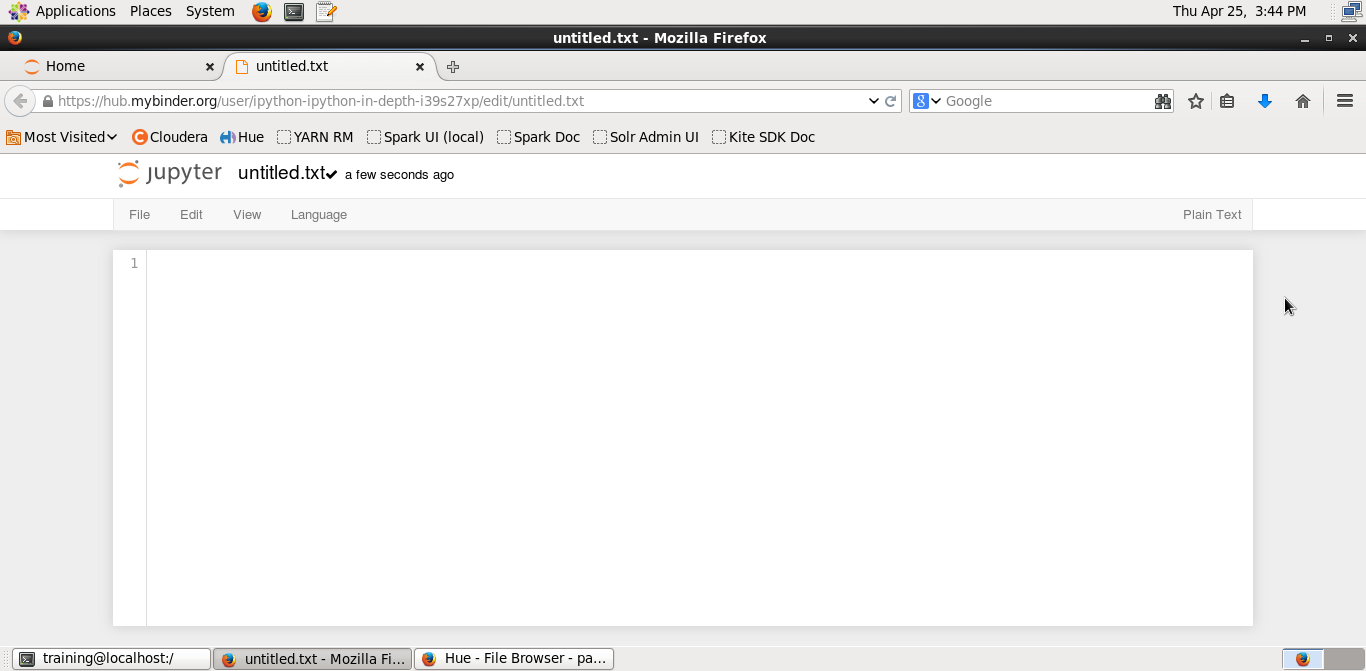
plt.ylabel(‘Count’)

plt.show()

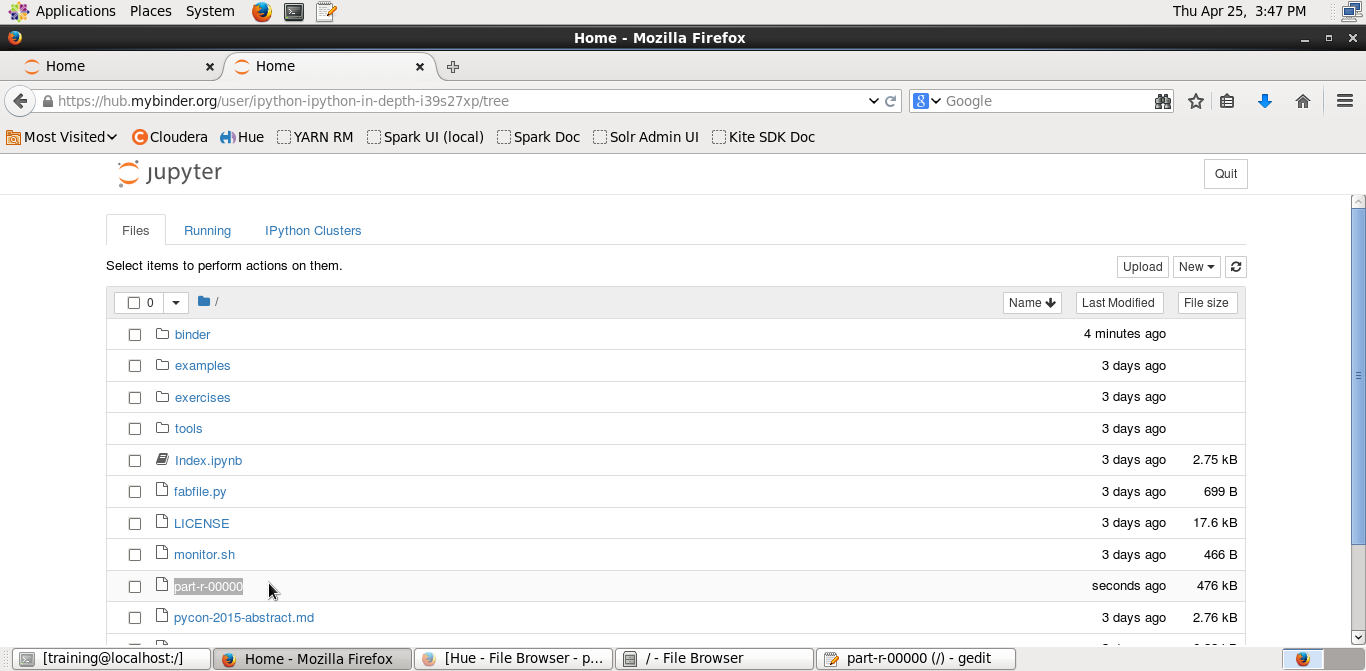
Step 22:



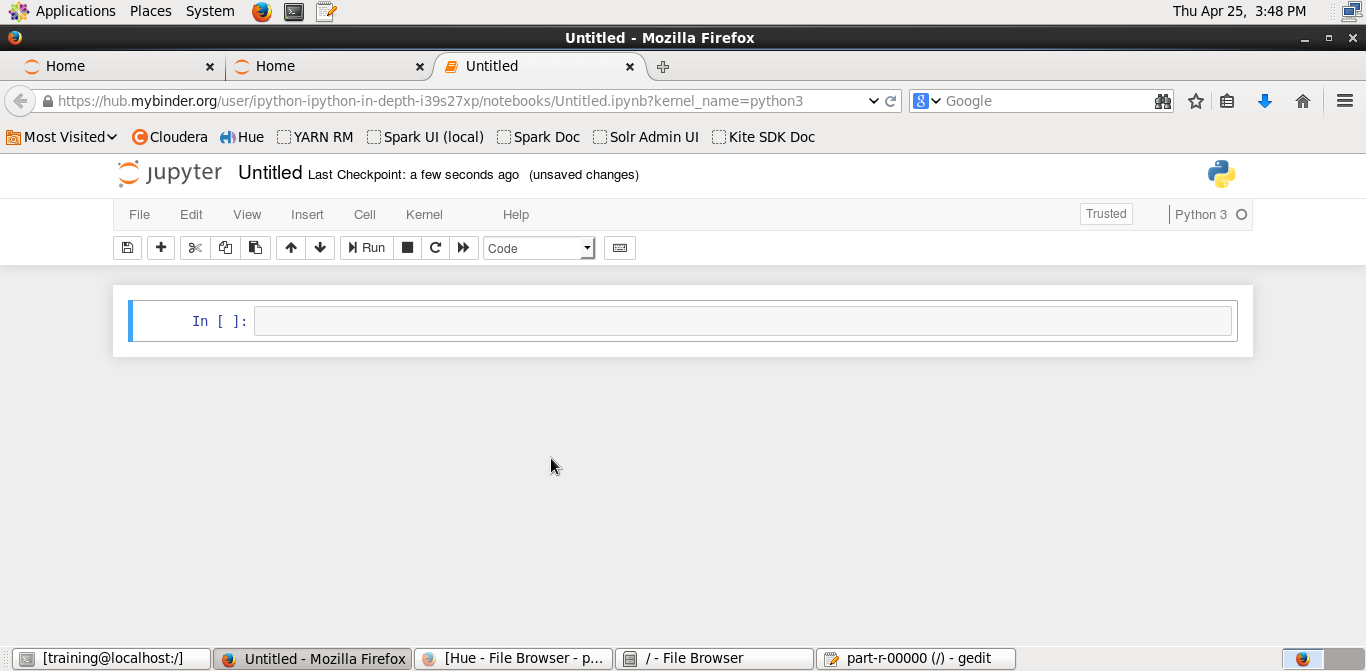
Step 23:



Step 24:



Step 25:



Step 26: From the graph we can say that there are more values for positive sentiment value

