# CCT College Dublin

Assessment Cover Page

To be provided separately as a word doc for students to include with every submission.

|  |  |
| --- | --- |
| **Module Title:** | Advanced Data Analytics  Big Data Storage and Processing |
| **Assessment Title:** | MSC\_DA\_CA2v4 |
| **Lecturer Name:** | David McQuaid  Muhammad Iqbal |
| **Student Full Name:** | Michelle Moran |
| **Student Number:** | SBA22220 |
| **Assessment Due Date:** | 21 May 2023 |
| **Date of Submission:** | 26 May 2023 |

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

|  |  |
| --- | --- |
| Total Document Word Count | 4095 |
| Word Count excluding References, appendices and Cover page | 3510 |

**Github details for virtual machine work =** [**www.github.com/MichelleMoran431/MSC\_DA\_CA2v4A.git**](http://www.github.com/MichelleMoran431/MSC_DA_CA2v4A.git)

**For the windows machine = https://github.com/MichelleMoran431/MSC\_DA\_CA2v4.git**

**1.0 Abstract**

The aim of the assessment is to carryout an analysis of a large dataset gleaned from twitter API. At least a years’ worth of tweet data must be collected on a topic and stored using a HDFS system and utilising a database system on the virtual machine. A comparative study is also required using any benchmarking system between two databases.

The type of analysis is Sentiment analysis involves determining the sentiment of the tweet data, is it positive or negative? To also analyse any change sentiment that occurs over the time that is selected. This will involve various data pre-processing techniques such as tokenization.

The second part of the assessment, following the analysis, a time series forecast of the sentiment at 1 week, 1 month and 3 months going forward and the results of this to be displayed as a dynamic dashboard.

**2.0 Introduction**

This assessment criteria highlights the importance of Advanced data analytics and Big Data storage and Processing. Due to the increasing amount of data generated in today’s digital world, there is an increasing importance on data management including storage. Big Data Management is the organization, administration, and governance of large volumes of both structured and unstructured data.The list of big data technologies that can be deployed, often in combination with one another, includes distributed processing frameworks Hadoop and Spark; stream processing engines; cloud object storage services; cluster management software; NoSQL databases; data lake and data warehouse platforms; and SQL query engines.

According to Martin et al, 2023, twitter is the worlds 7th favourite social media platform and twitter usage is growing 30% faster than Instagram. The most popular topic on twitter is politics and there are at least 500 million tweets sent every day. There is some fast statistics on twitter. The number of tweets is staggering, and this tweet data can be useful if managed and processed correctly. It can prove useful to companies and their brands, provide feedback on topics, also give an understanding of the public opinion, (shepard et al , 2023).

There are many processes in examining big data including tweets which can uncover hidden patterns and correlations. These processes can include data mining, machine learning, natural language processing in particular Sentiment analysis. Sentiment analysis allows people to gauge the overall sentiment or opinion of a large group of people and with time series prediction, can make predictions how people may feel about a certain topic based on previous tweet data.

**3.0 The Twitter Dataset**

Various avenues were examined in the attempt to acquire a 1 years’ worth of tweets on a topic:

1. Using the step-by-step guide reference in the continuous assessment, for getting data from Twitter API in python. I acquired a developer account and requested an academic access. The current level of access was only 30 days’ worth of tweets. With academic access the possible key benefits were to access Twitters real time and historical public data at 10 million tweets/month. However, there was no reply on my request and was unable to use as I need more than 30 days’ worth of tweets.
2. Internet archive site – the twitter Stream Grab, a simple collection of JSON grabbed form the general twitter streams. Its static dataset for years 2011 to 2021. For any chosen year and its associated 12 months files contained a large volume of tweets and presented a lot of challenges which made this option not viable, one main challenge was:
   1. Storage capacity – Using a laptop with limited storage, made downloading the data and storing the data difficult.
   2. Computational resources – the idea of downloading by month and then processing the data was examined by again, I didn’t have the sufficient memory of computational processing capabilities to complete. It took too long to download one months’ worth of data.
3. Kaggle and Data. World websites contain datasets of tweets that are easily accessible and do not have large volumes of data. However, these datasets were already cleaned and processed, and the topic chosen
4. The sentiment 140 is dataset of tweets created by computer science graduate students in Stanford university. They used Twitter API to build this dataset. It contains 1,600,000 tweets, for the year 2019.

The Sentiment140 dataset was the option used for this assessment. It was easily accessed and downloaded and still met the requirements of a 1 years’ worth of tweets and it was unclean.

The following are the 6 fields of the dataset:

1. **Sentiment:** The sentiment label indicating whether the tweet is positive (4) or negative (0).
2. **Tweet ID:** The unique identifier for each tweet.
3. **Date:** The date and time when the tweet was posted.
4. **Query:** The query used to collect the tweets. In the Sentiment140 dataset, this field is empty and can be ignored.
5. **User:** The username of the tweet author.
6. **Text:** The text content of the tweet.

NB: The Sentiment140 training dataset was downloaded and saved as **Tweets.csv** on the windows machine. The dataset was too large to save on the github prior to processing. The Tweets.csv file was then transferred to the virtual machine and uploaded onto the HDFS

**4.0 Utilisation of a Distributed data processing environment (**Belov et al 2021, Lam, et al, 2010)

**NB: Terminal screenshots of all utilisations of the HDFS and HIVE environments can be found in the appendices. I was unable to complete a screencast video demonstrating the above, as the virtual machine was unstable and had a lot of bother with it.**

* 1. **Hadoop DFS (** Zeebaree et al , 2020, Tate et al , 2012)

The dataset was downloaded and saved into the Hadoop distributed data processing environment. Hadoop is an open-source software platform for distributes storage and distributes processing of very large dataset on computer clusters. Due to the size of datasets and their storage needs, it become necessary to partition datasets across several machines and the management of this storage is governed by filesystems such as the Hadoop DFS.

The size of the dataset is 233 MB approx.

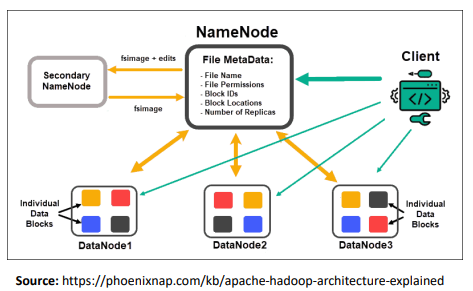


Fig 1.0 Apache Hadoop Architecture diagram

How does it work?

* Hadoop cluster has multiple machines working together to store and process data
* Its core components are NameNodes and Datanodes
* NameNodes is a metadata server and keeps track of the file system namespace, data on the files, directories, and their block locations.
* The Tweets.csv dataset is stored in the HDFS by splitting it into multiple blocks and distributes across DataNodes in the cluster. These dataNodes communicated with the nameNode to report on their status and perform data operations.

The advantages to using a HDFS is that it is a simple file system interface, easy to access and manage data. Users can interaction with the interface using the command-line or higher-level query languages such as Hive or PIG.

It is compatible with other Hadoop Ecosystems such as various data processing frameworks like MapReduce, Apache Spark, and hive.

* 1. **Storage of the source’s dataset into a SQL Hive**

After the Hadoop cluster was set up and running, the Tweets dataset was stored in the csv format in the HDFS. Apache HIVE was used in this assessment as for storing the sources dataset, Tweets.csv.

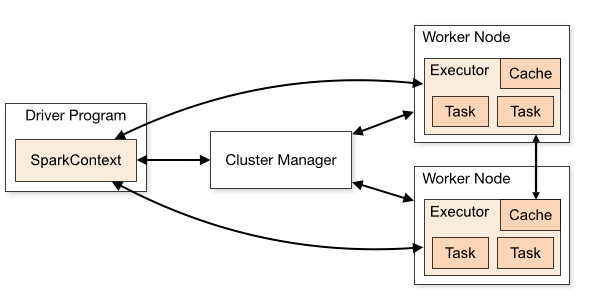
In general, it depends on the specific use cases, what the data requires as to what type of storage is required. Apache HIVE is a SQL-like query infrastructure built on top of Apache Hadoop for querying and analysing large datasets stored in HDF systems. Its many advantages include a SQL-like interface to work with big data. As its built-on top of APACHE HADOOP, it is a highly scalable framework for distributed processing.

It was uploaded from the Hadoop cluster, a *default* database was created, following this a table was created called table hive, detailing the headings along with the datatype required.

The data was uploaded into the table *tablehive*. This took 2 seconds for this. The first 2 lines were checked to make sure it was uploaded. Next was to connect to pyspark to complete processing. However, it didn’t connect and load the table from hive to the notebook.

1. **Processing the dataset (Ref: ProcessedTweetDataframeSpark.iynb)**

**5.1 Apache Spark**



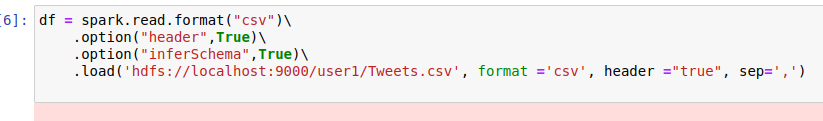
***Fig 2. Spark Architecture***

Apache spark was then used as the processing method to process the data. Apache spark is a general-purpose distributed cluster computing data processing framework like MapReduce which offers powerful abstractions for the processing of a large dataset. It is popular for its ability to keep large working dataset in memory between jobs and this allows it to outperform MapReduce workflow. The programming language used is Python using Pyspark libraries. It is commonly used and tends to be installed on all worker nodes.

The driver program mentioned in the fig 2 above, is the starting point and control centre for the spark application: organising the execution of the job and interacting with the spark cluster using python as the language of choice

**5.2 Data Processing (**Pang et al 2008)

The dataset Tweets.csv was uploaded and a data frame created = Tweets



***Fig 2.0 Notebook copy of the creation of the dataframe Tweets.***

Under Data Preparation the following was checked:

* Columns = added column titles: sentiment/ids/date/flag/user/text
* Remove unnecessary columns: ids/flag/user
* Null Values = None
* Duplication = None
* Case Normalization = completed

The dateframe Tweets1 was filtered for the key topic of interest which was food. The shape of the dataset went from 1600000 to 7721 rows with 3 selected columns.

A new csv file was created for these filtered rows and was saved to the HDFS system, called filteredtweet.csv and this was sent to HDFS.

**6.0 Comparative Analysis of two databases using the benchmarking tool YCSB(** Cooper et al , 2010)

YCSB is an open-source specification and software package for benchmarking No SQL database management solutions relative performance.

Its main function is to provide a standardized framework for benchmarking and comparing the performance of different databases across a range of comment application scenarios.

For this assessment, I used the YCSB Benchmark tutorial 9 for databases Mongodb and MySql and the wordloadaA example. This is another example of loading data into a NoSQL database.

The results were output to two text files: outputMySQL\_WorkLOADA.txt and outputMongodb\_WorkLOADA.txt, see appendix for screenshots.

Results table 2 below shows results under the throughput, latency, and runtimes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** |  | **MongoDB** | **MySQL** |
| Throughout PS |  | 0.0 | 0.0 |
| AverageLatencyMS | CLEANUP | 3357 | 25960.0 |
|  | INSERT | NaN | NaN |
|  | INSERT-FAILED | 3.00E7 | 303232 |
| 95thPercentileLatencyMS | CLEANUP | 3357 | 25967 |
|  | INSERT | 0 | 0 |
|  | INSERT-FAILED | 30097407 | 303359 |
| Runtime(ms) |  | 32030 | 3613 |

***Table 1: Comparison results for MongoDB and MySQL via the benchmarking tool YCSB***

Throughput result for both databases was 1 as there was only one operation that each database handled, where was the uploading of the file to each database. The latency was measured through 3 operations, CLEANUP, INSERT, INSERT-FAILED. The lower the latency the faster the response times. The results for the Average latency were better for the Mongo dB database. As a database, it was quicker at the clean-up, that is restoring to its original state before the next operation, it has better performance and efficiency of data insertion and higher robustness and error handling capabilities as a database system.

**7.0 Sentiment Analysis of one year’s Tweets dataset, (** Bhakuni, M. et al. (2022)

**Part 1: Virtual Machine using pyspark : reference notebook :ProcessedTweetDataFrameSpark.ipynb( ,** Aydin et al , 2023)

The dataset was saved as Tweets.csv file and stored in the virtual machine. Pyspark was used to import from the HDFS directory and to complete some data preparation and processing in Jupyter notebook. Python was the language of choice as its simple and ease of use. Pyspark was chosen as its the python API for Apache spark allowing a high-level interface between a distributed system and python language and libraries.

In the data preparation, this was completed on the full dataset, as I tried to filter at this stage but was unable to perform tasks in the notebook for some unknown reason, so I continued using the full dataset. Column headings were added to the dataset and then the unnecessary columns were removed such as the ids, flag queries, and the user details.

The sentiment polarity was completed, where negative tweets were deemed 0, positive tweets were 4. Sentiment polarity is important as it tells us the overall sentiment conveyed by the tweet text.

The shape of the dataset before filtering by topic was 1600000 rows by 3 columns. There was no null values and one duplicate row which was removed.

Data pre-processing or cleaning is important prior data machine analysis, using tweet text makes training a classifier/machine difficult based on the presence of emoticons, whitespaces, usernames, and links in the text.

Case normalization was completed as a text pre-processing technique used to reduce randomness of the tweet text for later analysis, so all tweet text was converted to lowercase. Trimming was performed to remove leading and trailing whitespace characters from a string tweet text. Examples of whitespace characters are spaces, tabs, newlines, all these do not add to the content of the tweet text.

**5.1 Filtering the Tweets1 dataset for the key topic of interest**

The key topic of interest chosen was the general term - Food. This brought the shape of the data frame from 1600000 rows of data to 7721 rows. The data frame was converted to a new csv file( ref: ProcessFilteredTweets.csv and saved to HDFS system. The rest of the sentiment analysis and time series analysis was performed on the windows system due to issues with the virtual machine freezing.

**Part 2: Windows system: reference: Sentiment Analysis.ipynb**

**6.0 Sentiment Analysis, (**Bhakuni, M. et al. (2022), Al-Khazaleh, M et al , 2023)

Here we will do further text pre-processing using NLTK. NLTK is natural language processing toolkit, it is a part of Artificial Intelligence that deals with the interaction between humans/computers and the natural language. Its goal is to read, determine and understand languages. This toolkit contains libraries and programs for processing the language written in python programming language. We have utilised some of its common algorithms such as tokenizing and Stop words.

In the NLT, the module called stop words contains common stop words such as “the”, “is” and are not deemed important in the whole sense of the sentence. So, by removing them improves the efficiency and accuracy for later NLT tasks.

The variable corpus was created to hold further preprocessed text from the dataframe.

The data was then split into a training and test dataset in an 80:20 ratio. The sentiment columns were 0 value for negative tweets and 4 for positive tweets. Using the encoder, we transformed that in) and 1.

* 1. **Word2Vec (** Aydin et al , 2023)

To gain a better understanding about the words in the tweet text of the dataframe bases on their usage and semantic associations, we looked at Word2Vec NTL algorithm. Another advantage to this is to learn meaningful word representations and how they are used together in sentences. This model was trained using the training dataset created, and the tweet column sentences was split in words and stored in the variable documents.

Three words were looked at “good”, “hate” and “great” in the trained model. A list of their similar words and their similarity scores. The higher the score, the greater the closeness to the chosen word. So, for example the scores for the similar words to hate and great were higher then for “good”. The words in the list were all descriptive of the topic food.

* 1. **Tokenisation**

This is fundamental part of NLT involving breaking the tweets text data into words called tokens. Using the keras library and the text\_to\_word sequence. A word index of length 11132 was created.

Using the pad function, and setting the maximum length to 300, this ensures that the input texts are the same lengths which is required by some machine models such as LSTM.

**7.0 Times Series Analysis(** Joosery et al , 2019)

LSTM model was used for this analysis as it can capture dependencies and patterns over time in sequential tweet data and it can handle variable-length input. It can also provide flexibility for feature engineering using word embedding and word2vec. The other model option ARIMA lacks these mechanisms for feature engineering and is also more suitable for analysing sentiment in aggregrated tweet data over time focusing on the time series pattern rather than the textual content.

For the model, a sequential model (a simple stack of layers) which includes embedding, dropout, LSTM and Dense layers.

Here are the steps involved:

Step 1

- The input to model is 300 words because these are the number features/words that we extracted above from the tokenizer vocabulary

Step 2

- Embeddings provide the presentation of words and their relative meanings. Like in this, we are feeding the limit of maximum words, length of input words and the inputs of previous layer.

Step 3

- LSTM (long short-term memory) save the words and predict the next words based on the previous words. LSTM is a sequence predictor of next coming words.

The Dropout layer is added to prevent overfitting and the next layer is LSTM.

Step 4

- Dense layers reduce the outputs by getting inputs from flatten layer. Dense layers use all the inputs of previous layer neurons and perform calculations and send 256 outputs

Step 5

- Activation function is node that is put at the end of all layers of neural network model or in between neural network layers. Activation function help to decide which neuron should be pass and which neuron should fire. So, activation function of node defines the output of that node given an input or set of inputs.

Step 6

- Dropout layers drop some neurons from previous layers. We apply this to avoid the overfitting problems. In overfitting, model give good accuracy on training time but not good on testing time.

**Model compilation**

2 classes used = "binary\_crossentropy

Optimizer = is a function that used to change the features of neural network such as learning rate (how the model learns with features) to reduce the losses.

Metrics = accuracy, calculate the percentage of correct predictions over all predictions on the validation set. Accuracy is the number of correctly classify tweets from all the tweets of positive and negative.

For example, if the trained model classify the 80 tweets correct and 20 tweets wrong from total of 100 tweets then the accuracy score will be 80%.

Accuracy= Total number of correct predictions/Total number of predictions

**Parameters**

**Batch size** =32 so the model takes 32 number of tweets in each iteration and train them. Batch size is a term used in machine learning and refers to the number of training examples utilized in one iteration.

**Epochs** =30 so the model will train on the data 30 Times. Epoch is a term used in machine learning and indicates the number of passes of the entire training dataset the machine learning algorithm has completed.

NB: These parameters can be changed to get good results from the training model. This is called parameter tuning.

**Results and Conclusions:**

* **Sentiment analysis.ipynb ( windows machine)**

There were equal negative and positive sentiments in the dataset on the topic Food.

Word2Vec was completed on the dataset, however there was a missed opportunity to provide change analysis. If time wasn’t an issue, further analysis of word2vec from different time periods could give insights into sentiment changes on food.

The X training and test data was based on the sentiments and the Y training and testing data was based on the vocabulary list from the tokenization section. An LSTM model was applied, the training accuracy was higher than the validation accuracy which indicates overfitting.

* **LSTM and Dashboard.ipynb**

This notebook shows the initial workings of creating a time series model and a working dashboard. However, the training and testing datasets created were inaccurate in the representing the foodtweets.csv file. I realised I was comparing only the polarity and not the tweets. However, the dashboard did work, see appendix for screenshot.

**Sentiment and LSTM.ipynb**

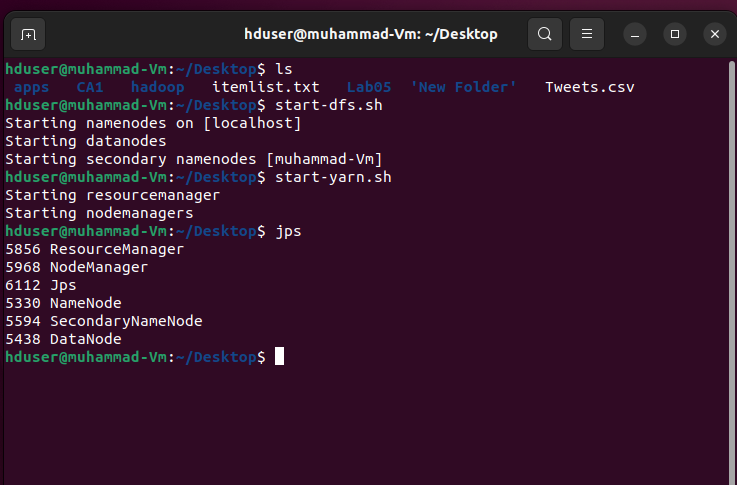
So, using this information, the Sentiment Analysis.ipynb was copied and the LSTM /Dashboard code was added to ensure the correct training and testing datasets were used. Unfortunately, due to time constraints, I had to stop the code as it was taking too long iliterate. This could be due to the large dataset, the batch size and no of epochs rans. However, looking at the accuracies for each of the time periods of 1 week, 1 month and 3 months, the trained model would potentially be able to correctly prediction the sentiment of a tweet.

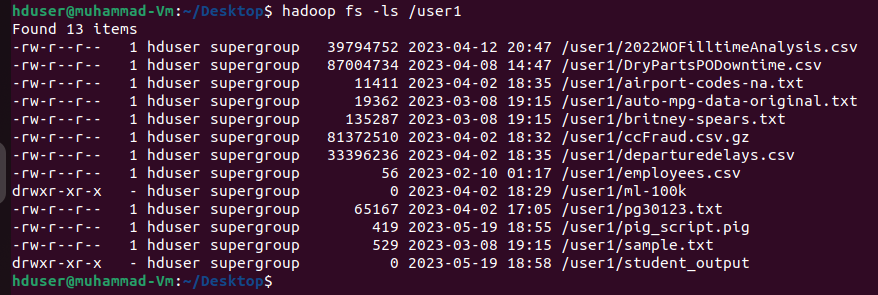
A learning from this assessment is to select a less broad topic, Food covers so much. The code in the notebooks attached worked but due to the parameter selection it took too long, as did the Dashboard, see working dashboard code.ipynb for details.

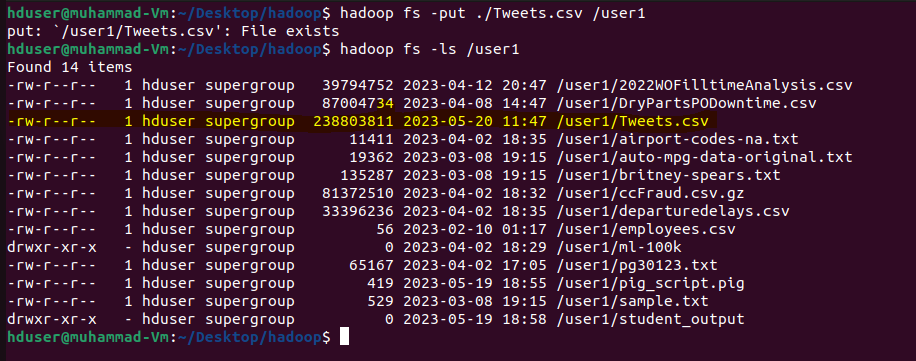
**Appendix:**

**SCREENSHOTS OF VIRTUAL MACHINE OPERATIONS:**

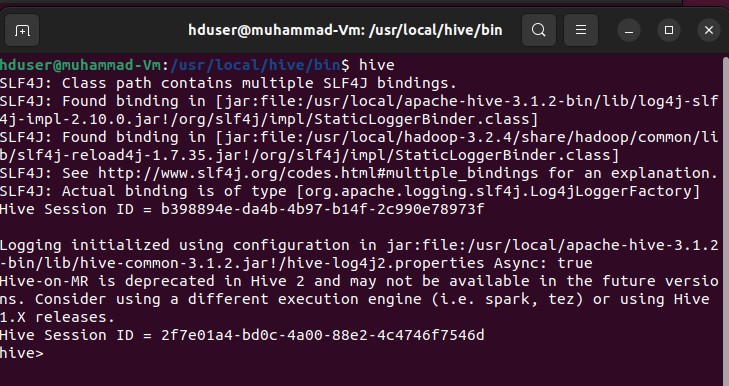
1. **Data Storage and Preparation using a HADOOP and APACHE HIVE**
   1. **Saving the Tweets.csv file onto the HADOOP DFS system**

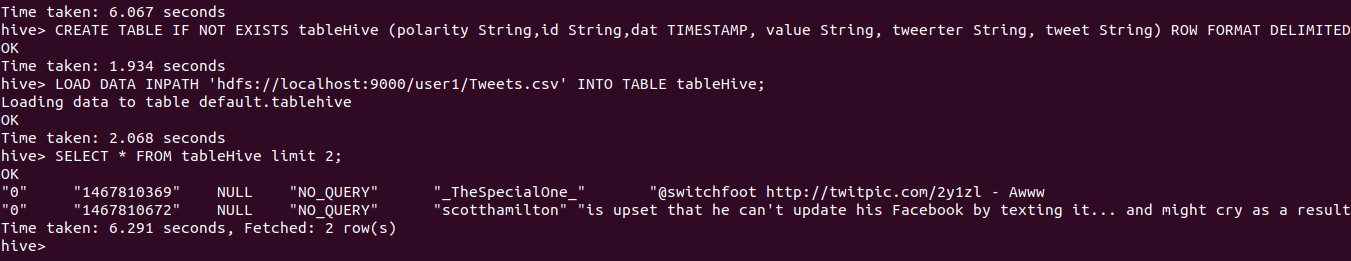






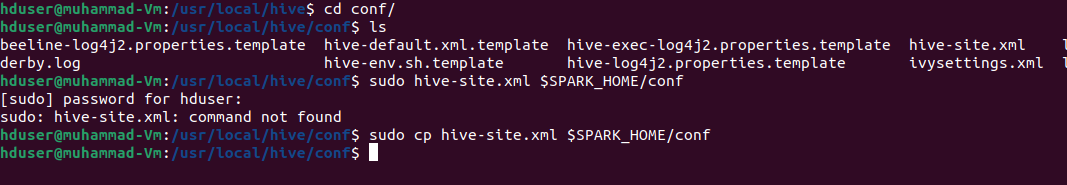
* 1. **Saving the Tweets.csv file from the HADOOP DFS system to the HIVE system**





**1.3 Data processing using pyspark on the hive table created:**

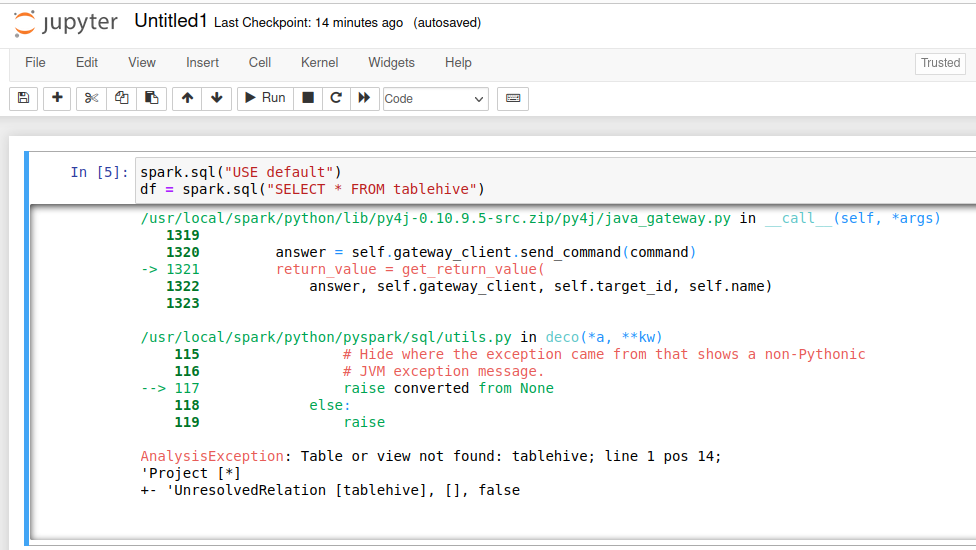
Copying the hive-site.xml from the hive/conf folder to the spark configuration folder:

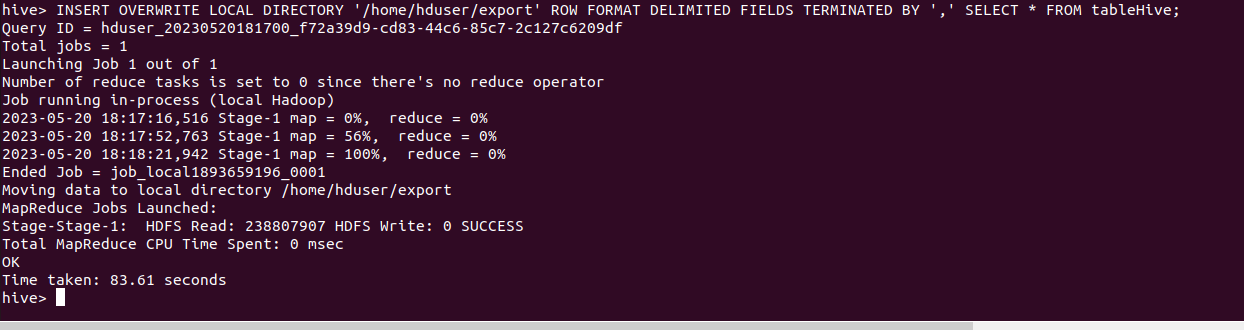


Ref :https://www.projectpro.io/recipes/read-table-of-data-from-hive-database-pyspark

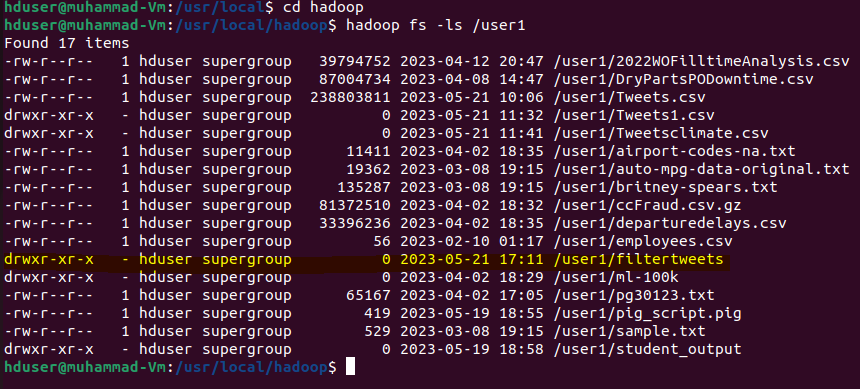
<https://phoenixnap.com/kb/hive-create-external-table>

Trying performing processing using pyspark by calling the table from hive:





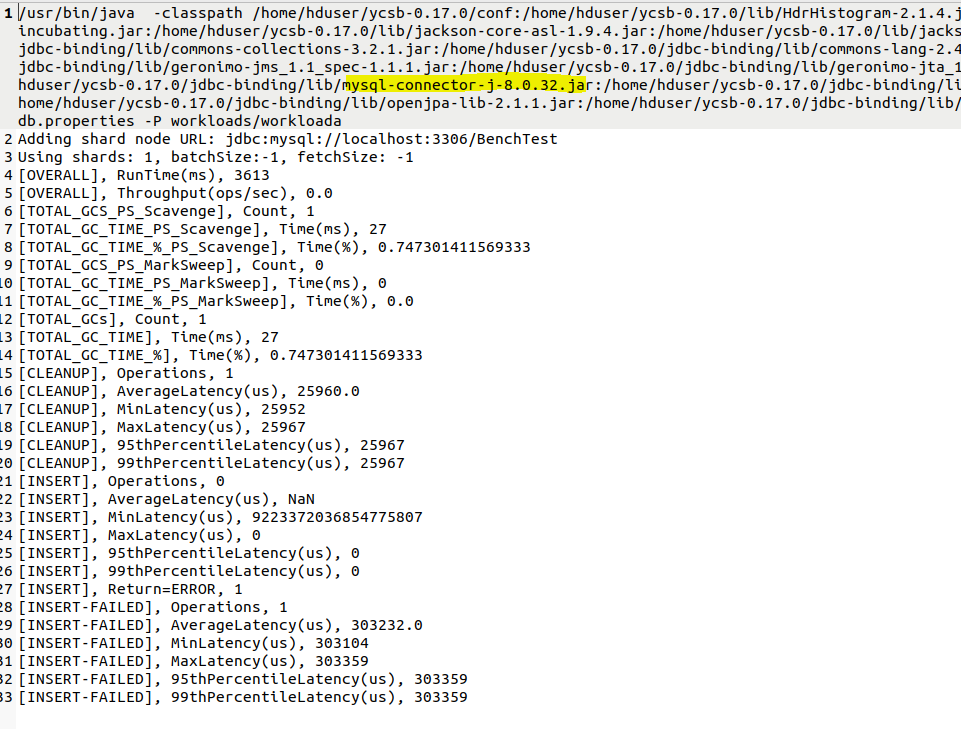
Created a new csv file in pyspark containing the filtered Tweets and saved it into Hadoop and then to the local machine to move to the windows system to complete sentiment analysis.



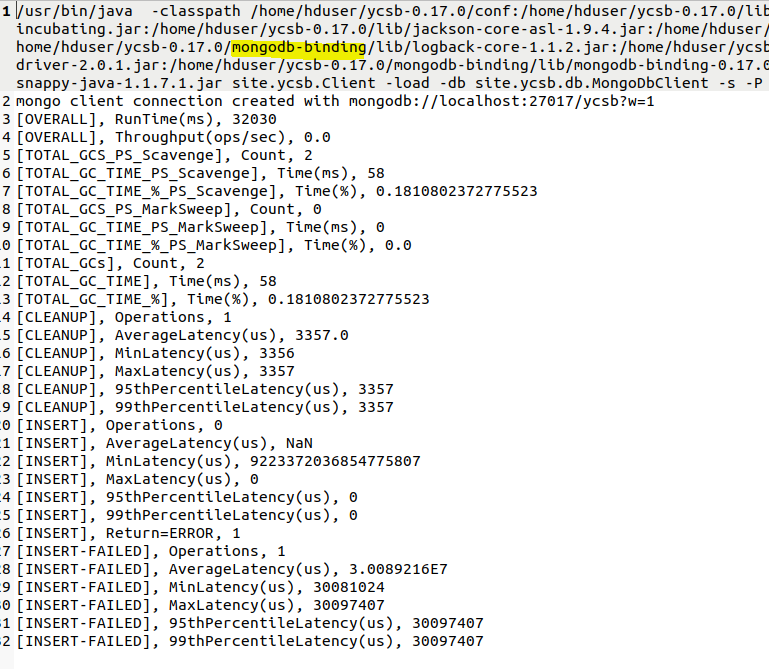
**1.4 Comparative Analysis using the benchmarking tool: YCSB**

Screenshots of the results .txt files:

**outputMySQL\_WorkLOADA.txt**



**outputMongodb\_WorkLOADA.txt**



**References:**

Amazon SimpleDB. http://aws.amazon.com/simpledb/.

Apache Cassandra. http://incubator.apache.org/cassandra/.

Apache CouchDB. http://couchdb.apache.org/.

Apache HBase. <http://hadoop.apache.org/hbase/>

Apache Spark <https://spark.apache.org/docs/latest/cluster-overview.html>

Apache HIVE: https://hive.apache.org/

Pyspark: <https://www.projectpro.io/article/pyspark-learning-spark-with-python/554#:~:text=PySpark%20emphasizes%20in%2Dmemory%20processing,and%20this%20exhibits%20low%20latency.&text=PySpark%20platform%20is%20compatible%20with,Java%2C%20Python%2C%20and%20R>.

[**https://blog.hootsuite.com/twitter-statistics/**](https://blog.hootsuite.com/twitter-statistics/)**, Michelle Martin, 2013**

Bhakuni, M. et al. (2022) ‘Evolution and Evaluation: Sarcasm Analysis for Twitter Data Using Sentiment Analysis’, Journal of Sensors, pp. 1–10. doi:10.1155/2022/6287559.

Belov, V., Tatarintsev, A., Nikulchev, E., 2021. Choosing a data storage format in the apache hadoop system based on experimental evaluation using apache spark. Symmetry 13, 195.

Cooper, B.F., Silberstein, A., Tam, E., Ramakrishnan, R. and Sears, R., 2010, June. Benchmarking cloud serving systems with YCSB. In Proceedings of the 1st ACM symposium on Cloud computing (pp. 143-154).

Joosery, Baleshwarsingh & Deepa, G. (2019). Comparative analysis of time-series forecasting algorithms for stock price prediction. 1-6. 10.1145/3373477.3373699.

http://help.sentiment140.com/for-students/, n.d.

Lam, C., 2010. Hadoop in Action. Manning.

Olson, M., 2010. Hadoop: Scalable, flexible data storage and analysis. IQT Quart 1, 14–18.

Aydin et al , 2023, PySpark Sentiment Analysis with Word2Vec Embedding

Paulpidou et al , 2023 , NLP Beginner - Text Classification using LSTM

Pang, B., Lee, L., 2008. Opinion mining and sentiment analysis. Foundations and Trends® in information retrieval 2, 1–135.

Reyes et al , 2020 , <https://medium.com/@karyrs1506/sentiment-analysis-on-tweets-with-lstm-22e3bbf93a61>

Sanketh et al , 2020 , https://towardsdatascience.com/text-preprocessing-with-nltk-9de5de891658

<https://www.analyticsvidhya.com/blog/2021/06/twitter-sentiment-analysis-a-nlp-use-case-for-beginners/>

https://www.kaggle.com/code/muhammadimran112233/eda-twitter-sentiment-analysis-using-nn

Tate, J., Beck, P., Ibarra, H.H., Kumaravel, S., Miklas, L., 2012. Introduction to storage area networks and system networking. IBM Redbooks.

Twitter Statistics :https://thesocialshepherd.com/ , Jack Shepherd, 2023

White, T., 2012. Hadoop: The Definitive Guide. O’Reilly Media.

Zeebaree, S.R., Shukur, H.M., Haji, L.M., Zebari, R.R., Jacksi, K., Abas, S.M., 2020. Characteristics and analysis of16 hadoop distributed systems. Technology Reports of Kansai University 62, 1555–1564.

**Lecture Notes from Semester 1**

Creating an Interactive Dashboard from Jupyter Notebook with Voila.pdf

Data mining and sentiment anaysis.pdf

Executable Dashboards.pdf

Sentiment Analysis with Python NLP Tutorial.pdf

time-series-forecasting-python-scikitlearn.pdf

Twitter Sentiment Analysis A Case Study for Apparel Brands].pdf