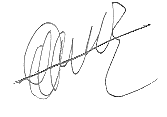
**CCT College Dublin**

**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(s):** | *David McQuaid*  *Muhammad Iqbal* |
| **Student Full Name:** | Candan Bayar |
| **Student Number:** | 2022352 |
| **Assessment Due Date:** | *26/05/2023* |
| **Date of Submission:** | *26/05/2023* |

****

**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. |

**TABLE OF CONTENTS**

**TABLE OF CONTENTS2**

**LIST OF FIGURES3**

**Introduction4**

**Methodlogy5**

Overview 5

Mysql vs SQLite 5-8

Pyspark8-11

Word Cloud 11-12

Sentiment Analysis12-13

Time Series Analysis13-15

K-means Clustering 16

LSTM(Single and BiLSTM)……………………………………………………………………………………………17-20

**Dynamic Dashboard22-23**

**Conclusion24-25**

**References26**

**LIST OF FIGURES**

Figure1: Mysql Sign in6

Figure 2: Mysql show dataset 6

Figure 3: SQLite create table 7

Figure 4: SQLite show dataset8

Figure 5: Create Pyspark App on Jupyter Notebook 9

Figure 6: Pyspark show data 9

Figure 7: print schema 10

Figure 8: type of dataset and sum null values. 11

Figure 9: word cloud 12

Figure 10: sentiment analysis 13

Figure 11: time series hour and count tweet 13

Figure 12: hour and tweet ……………………………………………………………………………………………………..………..14

Figure 13: daily, monthly and count tweet 15

Figure 14: k-means clustering 16

Figure 15: Single LSTM model accuracy 17

Figure 16: Single LSTM model loss 18

Figure 17: Single LSTM model evaluation 18

Figure 18: BiLSTM model accuracy 19

Figure 19: BiLSTM model loss 20

Figure 20: BiLSTM model evaluation 21

Figure 21: Correlation Matrix Heat Map21

Figure 22: Dynamic Dashboard 23-24

**INTRODUCTION**

Wordle application is very popular today. A word is shared every day, hidden by boxes and people make predictions. Words are written with Wordle, the correct letters in the right place light green, the orange letters are in the word but are in the wrong place, the wrong letters light up red. Wordle, which is a worldwide application, serves people in many countries and with many language options. Wordle, which also operates on Twitter, shares daily words on social media and people write comments, predict true word. In this study, twitter data written in wordle was analyzed and machine learning models were applied. Twitter data is for 2022, all data for 1 year have been collected and analyzed. Twitter data consists of date, id, tweet id and tweet texts. In addition, there are symbols in the form of square boxes in twitter texts. These data were collected from kaggle, archive.org for the year 2022. Data types are csv, json type, but they are recorded as sql in the database. Two database tools were used in this data analysis. They are Mysql and SQLite. Comparisons were made on both tools, and the advantages and disadvantages were shared. The data saved in the database were combined to analyze json and csv data and unnecessary data was cleaned from the dataset. Then, the data from the database was exported as csv and uploaded to the jupyter notebook. It was also integrated into pyspark, which is a big data tool, and the dataset was started to be analyzed. Data were analyzed using Natural Language Processing methods. The NLP methods used are as follows; Sentiment Analysis, Time Series Analysis, Kmeans Clustering, LSTM. These model results were written in the report.

**METHODLOGY**

**Overview**

I did not collect the data on Twitter through the API because Twitter's policies on API changed after Elon Musk bought Twitter. Elon Musk followed a policy that cares about privacy. In addition, along with the monetization policy from each service, Twitter also paid the API service. It has a free version, but you can only analyze daily and weekly data, but you have to pay a fee to analyze annual data. The data of 2022 were downloaded to the computer from the Archive.org website. However, it was difficult to select topics on archive.org, so the topics were searched on kaggle and it was decided to analyze the "Wordle" dataset. The reason for choosing this dataset was a clearer subject than other topics, so this dataset was chosen. The Kaggle dataset has a dataset of approximately 200 mb. The data of the year 2022 in json type with a size of 300 mb was downloaded from archive.org. After the data was downloaded, it was uploaded to Mysql and SQLite databases and converted to sql data type.

**Mysql**

Mysql database is used and recommended by many companies because Mysql is secure, you need to enter a password when logging in. If you enter the wrong password, you cannot use the system. In addition, with Mysql, you can not only work with your own local data, but also access and use data on a remote host. Mysql, which is among the most preferred tools for professional work, allows working with big data. I first uploaded the Twitter data to Mysql, combined the json and csv data and created a table.

**A screenshot of a login page

Description automatically generated with medium confidence**

Figure 1: Mysql sign in

I wrote the SQL query (**SELECT \* FROM new\_schema ‘tweet’**) to show all the data as shown in the picture in mysql database. In this way, I saved all the data in the database and showed the tables.

**A screenshot of a computer

Description automatically generated**

Figure 2: Mysql show dataset

**SQLite**

SQLite is a tool suitable for beginners to use SQL. People who have no knowledge of SQL and do not know how to create a database start using SQLite because it is the easiest tool to learn.

You can create a database with SQLite as in the picture.

A screenshot of a computer

Description automatically generated

Figure 3: SQLite create table

You can import data in csv and json data types more easily than Mysql, while Mysql's structure has difficulty in recognizing some data types, you can easily integrate and use it in SQLite. However, it is not secure because it does not require any password to login to the application, so it is not recommended for professional users, but personal users. In addition, while displaying the data, Mysql did not recognize the shape data, but SQLite recognized and displayed this data as well.

The image below shows the boxes under the text data.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 4: SQLite show dataset

In addition, you can download the dataset in any type you want while exporting. For example, you can convert a csv type dataset to sql data type and download it.

**Pyspark**

I have uploaded the Dataset to Jupyter Notebook. I analyzed the dataset using Pyspark. Pyspark is a tool for analyzing big data. Using Pyspark, it can analyze and model big data. PySpark combines the learnability and ease of use of Python with the power of Apache Spark to enable the processing and analysis of data of any size used with Python.

First I downloaded Pyspark, imported the library and then created an application called Pyspark "wordle\_tweet".

A screenshot of a computer program

Description automatically generated with medium confidence

Figure 5: Create Pyspark App on Jupyter Notebook

I uploaded the dataset to Pyspark and performed the operations related to the dataset. I wrote the codes to get various information about the dataset.

A screenshot of a computer

Description automatically generated

Figure 6: Pyspark show data

I wrote code the type and schema of the dataset.

A screenshot of a computer code

Description automatically generated with medium confidence

Figure 7: print schema

Then I made the libraries ready for import and analysis in order to model the dataset.

In order to analyze the dataset well, it is necessary to learn more about the dataset, so I made various queries about the dataset. These queries checked the null values in the dataset. Fortunately, there is no null value in the data.

The absence of null values in the dataset is a big advantage because if you delete too much data while cleaning the dataset, you may face some problems while modeling. Then I learned the row and column counts and learned more about the dataset.

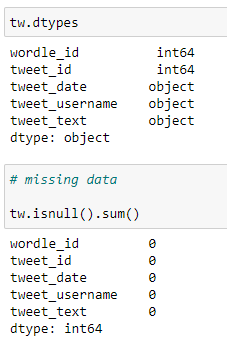


Figure 8: type of dataset and sum null values.

**Word Cloud**

Next step, the collected data was cleaned, I removed the data because it would contribute to better analysis by removing the dirty data from the data set. Cleaning the data is the first step in processing the data. Following that, the "word cloud" methodology was used to visually depict the most commonly used words based on their frequency of usage. The most frequently used word is coronavirus, Wordle and this result is obvious. It is very normal to have these outputs, but in addition to this, other posts people share are related to the "today", "one word".

A picture containing text, screenshot, font, number

Description automatically generated

Figure 9: word cloud

Next stage, I use Tokenization, the text into split markers and words. For example: 'wordle', 'got', 'lol'.

**Sentiment Analysis**

Sentiment analysis is a Natural Language Processing technique used to determine the emotional tone or polarity of text data, such as a product review, social media post, or news article. In this study, the mood of Wordle Twitter data was analyzed and visualized. Thanks to these sentiment analyses, I made inferences by determining what emotions people are experiencing regarding wordle. I showed the analyzes as percentages and numbers in the table. Values greater than zero are "positive", values less than zero are "negative" and values that do not meet these two conditions are "neutral".

According to the results, sentiment analysis has the highest rate of "neutral". The biggest reason for this is because people usually write words and don't write any comments that would include an emotion, it is not known what emotion they are writing in. Positive and negative values have the same ratio.

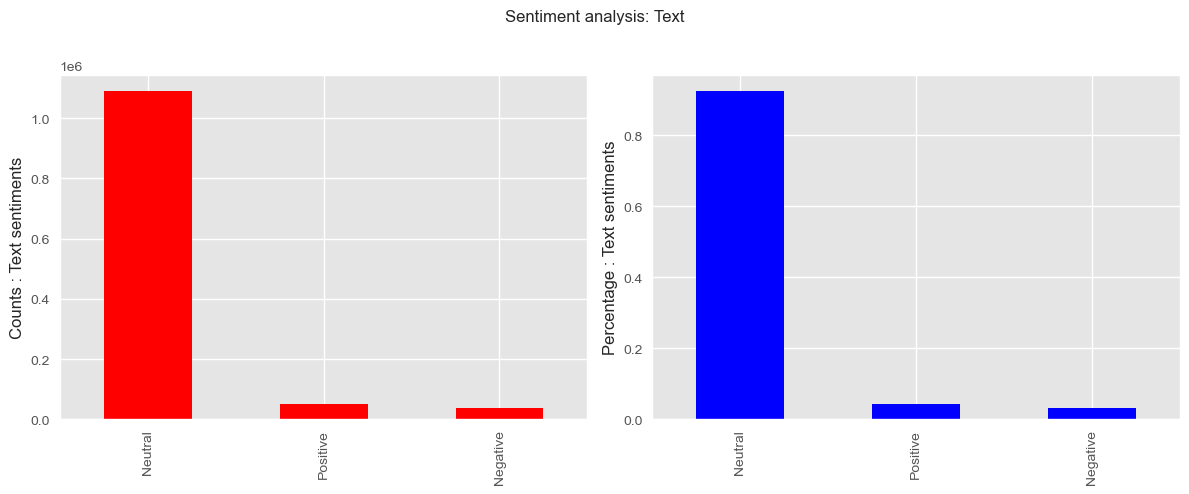


Figure 10: sentiment analysis

**Time Series Analysis**

Time Series Analysis, as the name suggests, allows modeling between values over time. Modeling is done with these two main factors.

According to the chart below, it shows how many tweets are shared per hour. The average time of tweets shared is around 20 and around.

A picture containing diagram, screenshot, line, plot

Description automatically generated

Figure 11: time series hour and count tweet

In this graph, it shows the number of tweets shares to the media every hour in a day. According to the graph, the number of tweets shared between 17 and 22 hours increased. One of the biggest reasons for this is that people share more when they are at home after working hours, that is, their twitter usage rates are the same. Also, since wordle is aware of this situation, it shares between these hours, so the participation will be higher.

A picture containing diagram, screenshot, plot, text

Description automatically generated

Figure 12: hour and tweet

This chart shows daily and monthly tweets together. While people shared about wordle in January and February, this rate decreased in other months.

A picture containing screenshot, plot, text, line

Description automatically generated

Figure 13: daily, monthly and count tweet

**K-means Clustering**

Generally, clustering is used to detect structures in a data . Since clustering is unsupervised, it works on data with no outcome (tweet text) variable as well as that which no relationship between the observations is known. Clustering is useful in creating generalizations about groups in the data. During model selection, a accuracy score was used to determine the model with the optimal separation degree among the clusters.

A part of it is shown in the image below. Its clusters are between 210 and 517, and these are 0,1 and 2 .

A screen shot of a computer

Description automatically generated with medium confidence

Figure 14: k-means

**LSTM**

LSTM is best suited for text data because it does not encounter overfit problems while learning the dataset and is best suited to give model-appropriate results. In this dataset, I used and analyzed 2 types of LSTM methods, and also visualized them in graphs. The LSTM models used are Single LSTM and Bidirectional LSTM.

**Single LSTM**

I wrote a function and added graphics in order to create the model with Single LSTM. While creating the model, I divided the dataset into 2 as train and test. In the train, the function I wrote is analyzed. In the test score, my train model should be tested with my real dataset. According to the graph, I showed the accuracy score of the model as train and test score.

It was concluded that there is a difference between the Train score and the test scores, that is, the dataset does not match the test set very well and the written function needs to be improved.

A picture containing text, line, screenshot, plot

Description automatically generated

Figure 15: Single LSTM model accuracy

I also visualized the loss values of the model in graphic. The model does not have much loss value.

A graph with a red line

Description automatically generated with low confidence

Figure 16: Single LSTM model loss

After this stage, I wrote a function to evaluate the model. According to the model score, the accuracy score is the most important score, and this score was taken as a basis while evaluating. According to the model score, 0.94 is a very high score, and it is observed that the model is successful.

A picture containing text, screenshot, font, number

Description automatically generated

Figure 17: Single LSTM model evaluation

**Bidirectional LSTM**

The next model is Bidirectional LSTM. Bidirectional LSTM is a model that processes data in both directions, from the beginning to the end of the array. BiLSTM is two LSTM networks stacked on top of each other, that is, it processes all data bidirectionally from the beginning of an input forward and then backwards from the end. The biggest reason I chose this model is that it is a more advanced model than Single LSTM. In BiLSTM, as in Single LSTM, there are two types of evaluation as accuracy score and loss score.

The accuracy score was evaluated in two ways as train and test. As can be seen in the graph, there is no harmony between each other.

**A picture containing text, plot, diagram, line

Description automatically generated**

Figure 18: BiLSTM model accuracy

As seen in the graph, the loss values of the model are slightly compatible with each other, because it has been observed that the test value decreases as the train value decreases.

A picture containing plot, diagram, text, screenshot

Description automatically generated

Figure 19: BiLSTM model loss

It is difficult to make accurate predictions based on graphics, so it is necessary to write a function to evaluate the model. When we write functions, we can make inferences about the accuracy score more clearly. For this reason, I wrote a function and got the following values. Accuracy score is a criterion for evaluation. According to the evaluation, the accuracy score is 0.94, so the model is successful. Although 0.95 and above is considered a successful score in general, this score may not be a problem, even if it is sometimes lower than the dataset.

A picture containing text, screenshot, font, number

Description automatically generated

Figure 20: BiLSTM model evaluation

**Confusion Matrix**

The confusion matrix is an evaluation of the harmony between values between 0 and 1. It is visualized with the Heat Map and evaluated over it.

In the graph below, it is observed that the confusion matrix averages between 0.2 and 0.8 and there is more agreement at neutral values.

**A screenshot of a screen

Description automatically generated with low confidence**

Figure 21: Correlation Matrix Heat Map

**Dynamic Dashboard**

Because of the format of the dataset, it is difficult and laborious to prepare a dashboard because it is difficult to visualize a text-type dataset. The function was written to prepare the dashboard and the appropriate visualizations were displayed on the dynamic dashboard. It was observed that some visualizations were meaningful and some were meaningless. These meaningless visualizations were excluded from the evaluation and dashboard outputs were obtained with meaningful visualizations. Many methods and tools can be used while preparing the Dashboard. The most frequently used tools are Power BI and Tableau, but since the code needs to be written with Jupyter Notebook in this study, it is not possible to write it in this way. If the dashboard was prepared with other techniques, more accurate results could be obtained and better visualized.

Some visualizations were included in the Dashboard, these are as follows;

* Scatter Plot
* Pie Chart
* Bar Plot
* Violin Plot
* Box Plot
* Distribution Plot
* Histogram
* Correlation Plot

A dynamic dashboard was prepared using these visualization methods and the results were shared. To run all visualizations, there are libraries to be downloaded in the code file, it is not possible to view the dashboard without downloading them. Next, you have to import the libraries and load the dataset. You can write the appropriate functions for the dataset and view the dashboard and compare the values within itself. By comparing the values in each row and column, you can make all the comparisons and see the correct results. If you do not compare the correct values, it will not output correctly and the images on the dashboard will not output correctly. You must use interrelated methods by trying the appropriate visualizations for the dataset. You can take a look at some examples below. They were prepared using interrelated methods. You can make visualizations and make accurate results by selecting rows and columns similar to this one. In addition, it is a very suitable technique to distinguish the differences between columns and rows in graphs with different color codes. In this way, do not experience any difficulties while making comparisons.

A picture containing text, screenshot, line, plot

Description automatically generatedA screenshot of a graph

Description automatically generated with medium confidence

A screenshot of a computer screen

Description automatically generated with low confidence

Figure 22: Dynamic Dashboard

**CONCLUSION**

As a result, the Wordle dataset is a very large dataset. The large dataset was used with the Pyspark tool, and then the model was analyzed with too many dimensions with the appropriate methods applied for large data. The dataset downloaded from archive.org and kaggle was uploaded to Mysql and SQLite database and the dataset in Json format was converted to SQL format. First, I uploaded the dataset to Jupyter notebook. In order to keep the SQL, Excel type dataset in one type, I uploaded it to the system and attitude in Excel format. After getting basic information about the data Set, I used the WordCloud method and had it analyze which words were used more. Then I did a sentiment analysis, that is, I learned how people shared their tweets and the result of this. People generally tweet in a neutral mood. For this situation, because wordle's content is to know the word, people only write words and make guesses, that is, they do not write long articles, which is reflected in the emotions as neutral. After these calculations, I went to the modeling stage and preferred K-means Clustering, Single LSTM, BiLSTM, which I think are the most suitable models for the dataset. It was difficult to make an accurate prediction according to the K-means Clustering algorithm, but it is generally evaluated between 210 and 517, between 0-2. While modeling LSTM, I made 2 types of evaluation. 2 types of Single LSTM and BiLSTM were evaluated and the results were visualized in the graph. Since the models could not be evaluated correctly on the graph, the function was written to evaluate the model, that is, it was evaluated based on the "accuracy score".It has benefited from many sources and similar studies have been looked at and appropriate models have been applied. This is the main reason why the applied models are successful. It is shown that the model is modeled successfully and the applied methods are correct. Detailed information has been shared in the relevant sections, models with an average accuracy of 0.94 show a successful result. Successful results can be obtained by writing similar algorithms in future studies. After that, the dynamic dashboard was prepared, so that the visualizations related to the dataset were analyzed in more detail and inferences were made. It was difficult to prepare a dynamic dashboard for this dataset, because the dataset is generally a text-type dataset, and since it is difficult to visualize it, some difficulties were encountered while preparing the dashboard.

As a result, various analyzes were made and the dataset was analyzed in detail and the results were shared. It is difficult to analyze in large datasets, it is necessary to have a powerful operating system. During these analyses, the model was expected to be trained for a long time during the modeling phase and the results were written accordingly. Different results could have been obtained with different methods, but I did not need it because the methods I applied yielded correct results.

**REFERENCES**

[1] <https://archive.org/details/archiveteam-twitter-stream-2022-11>

[2] <https://www.kaggle.com/datasets/benhamner/wordle-tweets>

[3] <https://towardsdatascience.com/hacking-wordle-f759c53319d0>

[4] <https://towardsdatascience.com/finding-the-best-wordle-opener-with-machine-learning-ce81331c5759>

[5] <https://digitalprojectstudio.wordpress.com/2015/05/28/visualizing-twitter-status-data-with-wordle/>

[6] <https://realpython.com/k-means-clustering-python/>

[7] <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

[8] <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

[9] <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>

[10] <https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/>

[11] <https://www.aiplusinfo.com/blog/introduction-to-long-short-term-memory-ltsm/>

[12] <https://www.lexalytics.com/blog/machine-learning-natural-language-processing/>

[13] <https://www.analyticssteps.com/blogs/top-nlp-algorithms>

[14] <https://www.kdnuggets.com/2021/02/6-nlp-techniques.html>

[15] <https://www.projectpro.io/article/machine-learning-vs-nlp/493>

[16] <https://www.expert.ai/blog/natural-language-processing-algorithms/>

[17] <https://www.marketingaiinstitute.com/blog/7-key-differences-between-nlp-and-machine-learning-and-why-you-should-learn-both>

[18] <https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/>

[19] <https://medium.com/analytics-vidhya/natural-language-processing-from-basics-to-using-rnn-and-lstm-ef6779e4ae66>