**CCT College Dublin Continuous Assessment**

**CCT College Dublin**

**Assessment Cover Page**

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

|  |  |
| --- | --- |
| **Module Title:** | BIG DATA PROCESSING AND STORAGE |
| **Assessment Title:** | LSTM AND AUTOREGRESSIVE TIME SERIES FORECASTING OF SENTIMENTS AT DAY 1,3 AND 7 GOING FOWARD |
| **Lecturer Name:** | DAVID MCQUAID  MOHAMMED IQBAL |
| **Student Full Name:** | DIANA FLORA NAMAEMBA |
| **Student Number:** | 2023385 |
| **Assessment Due Date:** | 19TH MAY 2024 |
| **Date of Submission:** | 18TH MAY 2024 |

**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. |

Contents

[INTRODUCTION 4](#_Toc166745588)

[Method 4](#_Toc166745589)

[DATA 4](#_Toc166745590)

[BIG DATA PROCESSING AND STORAGE 5](#_Toc166745591)

[PROJECT TWEETS DATA PROCESSING USING APACHE SPARK AND STORAGE USING MONGODB 5](#_Toc166745592)

[YCSB FOR COMPARATIVE ANALYSIS OF MYSQL AND MONGODB 6](#_Toc166745593)

[**Figure 1:The flow chart below shows the process of data Processing and storage** 6](#_Toc166745597)

[**Figure 2: Graph of RunTime comparison of MySQL vs MongoDB against other comparative parameters** 8](#_Toc166745598)

[**Figure 3: Graph of Throughput comparison of MySQL vs MongoDB against other comparative parameters** 9](#_Toc166745599)

[**Figure 4: Graph of Average Latency comparison of MySQL vs MongoDB against other comparative parameters** 9](#_Toc166745600)

[**Figure 5:Graph of RunTime comparison of MySQL vs MongoDB against other comparative parameters** 10](#_Toc166745601)

[**Figure 6:Graph of Throughput comparison of MySQL vs MongoDB against other comparative parameters** 10](#_Toc166745602)

[**Figure 7:Graph of Average Latency comparison of MySQL vs MongoDB against other comparative parameters** 11](#_Toc166745603)

[**Figure 8:Graph of RunTime comparison of MySQL vs MongoDB against other comparative parameters** 12](#_Toc166745604)

[**Figure 9:Graph of Throughput comparison of MySQL vs MongoDB against other comparative parameters** 12](#_Toc166745605)

[**Figure 10:Graph of Average Latency comparison of MySQL vs MongoDB against other comparative parameters** 13](#_Toc166745606)

[Table 1: Table on the comparison parameters considered for YCSB workbench evaluation 7](#_Toc166745607)

**ACRONYMS**

# INTRODUCTION

Social networking sites connect people in the world, allowing them to share pictures, content, videos and share their first-hand opinions on various issues. Big data analytics techniques are highly applied in the social networks because they are characterized by the 5V (Velocity, Volume, Veracity, Value AND Variety) s of big data.(Bazzaz Abkenar *et al.*, 2021). Some examples of social networking sites include Twitter, Facebook etc. Due to social media providing a big source of data, there has been an increase in application of machine learning, deep learning and time series techniques to better understand various problems. A lot of these analysis has been done mostly on twitter data. Twitter which has over 313 million monthly active users and which in a day over 500 million tweets are made, is one of the most utilised social media platforms when it comes to data.(Jianqiang and Xiaolin, 2017)

Some key techniques employed for understanding social media data include, sentiment analysis, times series analysis

# METHOD

## DATA

The dataset used for this project was project Tweets dataset, a csv dataset with 1,600,000 observations and five features extracted using the twitter api. It contains the following 5 fields:

* ids: The id of the tweet (eg. 4587)
* date: the date of the tweet (eg. Sat May 16 23:58:44 UTC 2009)
* flag: The query (eg. lyx). If there is no query, then this value is NO\_QUERY.
* user: the user that tweeted (eg. bobthebuilder)
* text: the text of the tweet (eg. Lyx is cool)

## BIG DATA PROCESSING AND STORAGE

### PROJECT TWEETS DATA PROCESSING USING APACHE SPARK AND STORAGE USING MONGODB

Big data processing is techniques utilised to access large scale data and extract meaningful information from them for decision making.(Mehdipour, Noori and Javadi, 2016), while big data storage are storage technologies that are not relational database systems that can be able to address the Volume, variety and velocity challenges of data.(Strohbach *et al.*, 2016). There are different big data storage and processing technologies available. Processing technologies include Hadoop Map-Reduce or Apache Spark, etc. While storage include either SQL or NoSQL databases such as HBase, HIVE, Spark SQL, Cassandra, MongoDB. Etc. MongoDB is a NoSQL database that is document oriented. It was preferred to other NoSQL databases because It is more useful for tweet analysis and other applications. It stores data in JSON format make it easy to analyse data.(Krishnan and Elayidom, 2016).

For this project Apache Spark was used for Preparing and Processing the Project Tweets Data, while MongoDB and spark SQL were used to Populate, Store and save Processed Data.Apache spark is a big data processing platform that has capabilities for both batch and stream processing. Spark was preferred over mapreduce because it outperfoms MapReduce in terms of performance.(Shaikh *et al.*, 2019). The Data was processed under batch processing, that is the its was a one time operation and after processing the output data was stored. Processing the data using Apache spark from MongoDB included: -

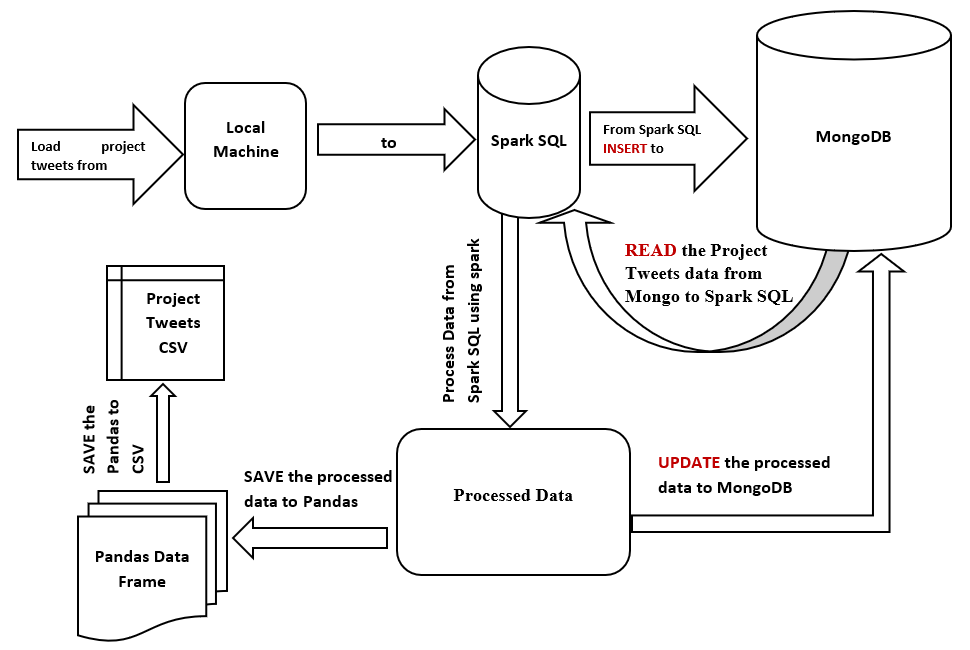
1. Performing CRUD (Create, Read, Update, Delete) operations that are possible for data stored in MongoDB. The CRUD operations are categorized into Read operations and Write Operations. The Project Tweets data was first loaded to Spark SQL from the Local machine. The CRUD operations were then performed as follows:
2. **Read operations** – These involved reading/loading the project tweets data stored in MongoDB into Spark SQL.
3. **Write Operations**- These operations involved creating/inserting, modifying and deleting data in MongoDB. Data was inserted from Spark SQL, deleting some parameters during processing and finally updating the processed data into MongoDB.
4. **Processing the data**

This involved Exploratory Data Analysis (EDA**).** EDA was performed to better understand the dataset, its patterns and characteristics. It involved checking for duplicates and missing data/dates in the dataset. The time components that are days and months were also extracted.

From EDA: -

* There were missing data points.
* There were duplicates

**Figure 1:The flow chart below shows the process of data Processing and storage**

****

The rationale behind choosing these two technologies was the need for efficient data processing and storage solutions that are capable of handling big data. Both Apache spark and MongoDB were chosen because of their performance capabilities, scalability, flexibility and suitability when it comes to handling tweets datasets, so as to ensure effective data manipulation, analysis and storage. Python was chosen as the programming language due to its capabilities and compatibility to effectively integrate with Apache spark.

### YCSB FOR COMPARATIVE ANALYSIS OF MYSQL AND MONGODB

There are various test strategies that can be implemented in order to perform comparative analysis of the capabilities of various big databases. One of the most common is Yahoo cloud service benchmark client (YCSB). YCSB is an open-source license tool used to benchmark new cloud database systems. Through YCSB one can be able to benchmark multiple systems and compare them by creating “workloads”. (Gaikwad and Goje, 2015). A YCSB Comparative analysis was conducted to compare Capabilities of MySQL and MongoDB.

The comparison involved comparing the two databases performance based on record counts 1,000, 10,000 and 100,000. Further the read and update proportions were also compared for different proportions and lastly the request distributions compared were Zipfian and uniform distribution for workload A, B and C, as shown in table below.

Table 1: Table on the comparison parameters considered for YCSB workbench evaluation

|  |  |  |
| --- | --- | --- |
| **Comparison Parameters** | **MYSQL** | **MONGODB** |
| Record Counts | 1,000, 10,000 and 100,000 | |
| Request distribution | Zipfian and Uniform | |
| Workload A – Update heavy | * Read: 50% and Update: 50% * Read: 70% and Update: 30% | |
| Workload B- Read heavy | * Read: 95% and Update: 5% * Read: 70% and Update: 30/5 | |
| Workload C- Read only | * Read: 100% and Update:0% * Read: 70% and Update: 30% | |

The metrics considered for quantitative analyses were:

1. ***Average Latency*** which is the average time that a database takes to read or write data. A lower value of the average latency indicated faster response time hence better performance by the database.
2. ***Throughput***is the rate at which a database processes a given number of operations within a given time period. A higher throughput value indicates the database has a better performance when handling workloads.
3. ***Runtime*** is the total time that is taken to execute a particular workload/ a set of operations. When the runtime is longer a database is able to perform a more comprehensive performance evaluation and stress testing of the database.

#### WORKLOAD A

##### Runtime

From the graph below, the runtime of MongoDB is lower than MySQL under all comparative parameters. In uniform distribution the runtime values are higher for both databases than for Zipfian distribution. The uniform distribution allows the databases to perform more comprehensive evaluations and stress testing. The runtime for the proportion 50/50 is higher than 70/30 for both distributions for both databases.

**Figure 2: Graph of RunTime comparison of MySQL vs MongoDB against other comparative parameters**

##### Throughput

Under Zipfian distribution MongoDB had higher values of throughput compared to MySQL. The values of throughput increase as the record count increases. While for MySQL the values are very low. The read/update value 50/50 has higher values of throughput compared to 70/30 proportion. Therefore, MongoDB had better performance when handling workload A compared to MySQL. The throughput is increasing linearly indicating good scalability for MongoDB.

**Figure 3: Graph of Throughput comparison of MySQL vs MongoDB against other comparative parameters**

##### Average Latency

MongoDB has lower average latency in both uniform and Zipfian distribution as compared to MySQL. The uniform distribution has lower average latency compared to Zipfian in MongoDB and the latency decreases with increase in read count. MongoDB has lower values of average latency, this indicated faster response time hence better performance by the database, hence a scalable system.

**Figure 4: Graph of Average Latency comparison of MySQL vs MongoDB against other comparative parameters**

#### WORKLOAD B

##### Runtime

From the graph below, the runtime of MongoDB is lower than MySQL under all comparative parameters. In uniform distribution the runtime values are higher for MySQL when the read count is 100,000 for both 95/5 and 70/30.

**Figure 5:Graph of RunTime comparison of MySQL vs MongoDB against other comparative parameters**

##### Throughput

Under Zipfian distribution MongoDB had higher values of throughput compared to MySQL. The values of throughput increase as the record count increases. While for MySQL the values are very low. The read/update value fluctuate between the two proportions. Therefore, MongoDB had better performance when handling workload B compared to MySQL. The throughput is increasing linearly indicating good scalability for MongoDB.

**Figure 6:Graph of Throughput comparison of MySQL vs MongoDB against other comparative parameters**

##### Average Latency

MongoDB has lower average latency in both uniform and Zipfian distribution as compared to MySQL. The uniform distribution has lower average latency compared to Zipfian in MongoDB and the latency decreases with increase in read count. MongoDB has lower values of average latency, this indicated faster response time hence better performance by the database, hence a scalable system

**Figure 7:Graph of Average Latency comparison of MySQL vs MongoDB against other comparative parameters**

#### WORKLOAD C

##### Runtime

From the graph below, the runtime of MongoDB is lower than MySQL under all comparative parameters. In uniform distribution the runtime values are higher for MySQL when the read count is 100,000 for both 100/0 and 70/30.

**Figure 8:Graph of RunTime comparison of MySQL vs MongoDB against other comparative parameters**

##### Throughput

Under Zipfian distribution MongoDB had higher values of throughput compared to MySQL. The values of throughput increase as the record count increases. While for MySQL the values were very low. The read/update value fluctuate between the two proportions. Therefore, MongoDB had better performance when handling workload C compared to MySQL. The throughput was increasing linearly indicating good scalability for MongoDB.

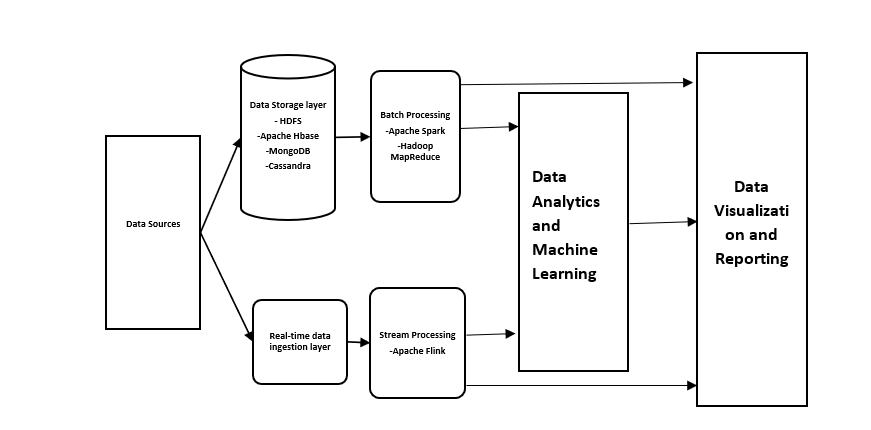
**Figure 9:Graph of Throughput comparison of MySQL vs MongoDB against other comparative parameters**

##### Average Latency

MongoDB had lower average latency in both uniform and Zipfian distribution as compared to MySQL. The uniform distribution had lower average latency compared to Zipfian in MongoDB and the latency decreased with increase in read count. MongoDB had lower values of average latency, this indicated faster response time hence better performance by the database, hence a scalable system

**Figure 10:Graph of Average Latency comparison of MySQL vs MongoDB against other comparative parameters**

### THE ARCHITECTURE FOR PROCESSING BIG DATA

****

**Figure 11:The architecture for Processing Big Data**

**Data sources**

The data sources could include: Real time data, Application generated data, Static (web server log files) and Application data (connection data). Social media data is a good source of data. The data could be structured, unstructured or both. The data can then be ingested either as batch or real-time.

**Data Storage Layer**

These is the layer where data is ingested. The data can be structured or unstructured, hence the reason the data storages are either NoSQL or Relational Databases. Structured data is stored in relational databases such as sparkSQL. While unstructured data is stored in NoSQL Databases such as HBase, MongoDB, Cassandra etc.

**Real Time data ingestion layer**

Allows for categorizing data so that the data can be smoothly transitioned into a deeper layer of the environment where the data is stored in real time for stream processing.

**Batch processing Layer**

This involves running long batch jobs to ensure the data is useful for analysis. The source files are read, processed and the output written to a new file. Hadoop is good for this.

**Data Analytics and Machine Learning store**

The processed data can now be utilised for analysis and machine learning. E.g Apache Hive

**Data Visualization and Reporting**

After analysis has been done, the findings can be visualized in dashboards etc or summarized in reports.

## ADVANCED DATA ANALYTICS

### A discussion of the rationale, evaluation, and justification for the choices you have made in terms of EDA, data wrangling, machine learning models and algorithms that you have implemented.[0-40]

**EXPLORATORY DATA ANALYSIS**

The primary aim of the EDA was to examine the data’s distribution, outliers, and any anomalies that would be used to generate specific hypotheses for testing and to assist in pattern recognition. (*Secondary Analysis of Electronic Health Records*, 2016). EDA was performed in three stages, exploring the entire data. Exploring the date variable and then exploring the text variable.

1. In the whole dataset wo main EDA processes that were beneficial for this data were: -

a) Checking for any missing data:Missing data have major effects on conclusions made from the data. Therefore, identifying them is crucial for handling problems they cause.(Dettori and Norvell, 2018). The data had 80 missing dates.

Checking for duplicates:Some impacts of duplicates include; the generation of erroneous observations, generation of more repeated observations, loss of observations, and incorrect statistics. (Cheng, no date). The duplicates were determined by checking three variables together, that is, user, date and text. A total of 3738 duplicates were found.

1. In the date variable EDA included, checking the start and last date, and missing dates. The start date was 7/4/2009 and the last date was 25/4/2009. The data had a total of 774,363 unique dates.
2. In the text variable EDA included: -

* Counting number words in the text. The largest text had 110 words.
* Counting the number of characters in the text. The maximum number of characters is 374.
* The average word count in the text
* The number of stop words. The maximum number of stop words is 25
* The number of hashtags. The maximum number is 24
* The number of @. The maximum number is 13.
* The number of upper cases. The maximum number is 40

This EDA guided the data cleaning and preprocessing of the text variable.

**DATA PREPARATION AND CLEANING**

Data cleaning organizes data, making it ready for analysis. It helps identify and remove inconsistencies and errors in data, improving the data quality.(Ridzuan and Wan Zainon, 2019).

The Data cleaning steps included:

**Step 1: Handling missing data:** Handling missing data ensured the data was reliable, meaningful in analysis, and unbiased(Kang, 2013). The listwise deletion method was used to handle missing data in both datasets. Other techniques would alter the shape of the distribution.(Kang, 2013). The missing dates were handled using time series technique for handling missing dates called linear interpolation. interpolation was done by considering daily data. In linear interpolation method values between two known data points are estimated. The method was preferred over the other techniques because it assumes there is a relationship between a range of data points, which is the case in tweets.(*Preprocessing and Data Exploration for Time Series — Handling Missing Values | by Data Science Wizards | Medium*, 2023)

**Step 2: Removing features that were not used:** Removing irrelevant features helps overcome the curse of dimensionality and reduce overfitting problems.(Afshar and Usefi, 2022). Variable flag, user and ids were not used in the analysis.

**Step 3: Removing duplicate Observations:** Duplicate observations were dropped because they could result in incorrect statistics. These were 3738 observations.

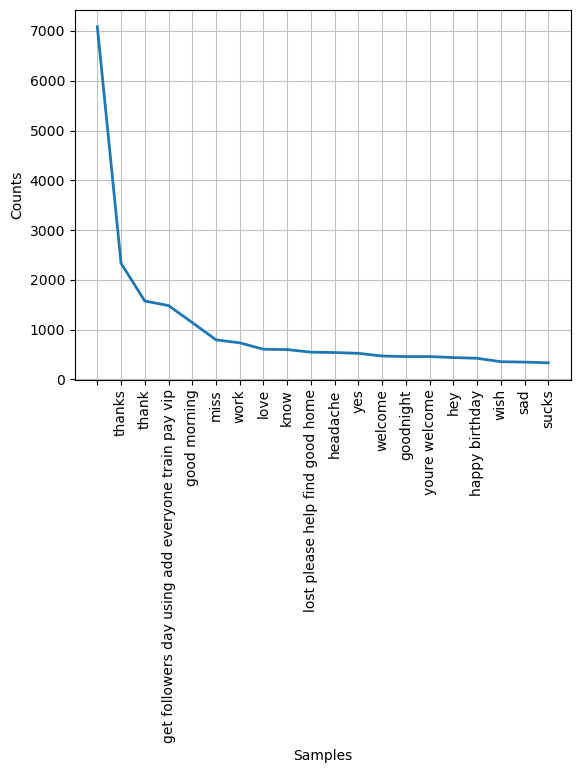
**Step 4: Tweets Processing**

After EDA, the identified problems were addressed during preprocessing. Preprocessing text is important since it helps to remove noise from text and reduce inconsistencies to ensure the data can be used for sentiment analysis of mining text.(Samuels and Mcgonical, 2019). Preprocessing involved this step: -

* **Text normalization** -which is the process of trying to reduce randomness in a text, by trying to make is closer to or even standard.(*Text Normalization for Natural Language Processing (NLP) | by Diego Lopez Yse | Towards Data Science*, 2021). It involved: -
* Remove special characters
* Change the upper cases to lower cases
* Remove numbers/integers
* Remove punctuations
* Remove white space
* Remove URLS/links
* Remove the username
* **Tokenization-** Tokenization is breaking the text into tokens. Tokens could be either words, symbols, phrases, or even the whole sentence Tokenization used was the word tokenization.
* **Remove stop words**- The number of stop words was counted for each text and then removed from the texts.
* **Lemmatization -** is the process of finding the root of a word rather than the stem. (S *et al.*, 2020) . Lemmatization was applied to the comment variable to obtain the root of the word. Stemming was done but the output did not convey any meaningful information, so lemmatization was done because the root of the words was more meaningful. Below Is the process of text processing: -

**After text processing a frequency distribution plot was made for the tokenized text. Below is the graph of the frequency distribution.**

Figure 12:frequency distribution plot of tokenized text.



**DATA VISUALIZATION OF THE DATE VARIABLE**

Figure 13: Graph on the count of each unique day

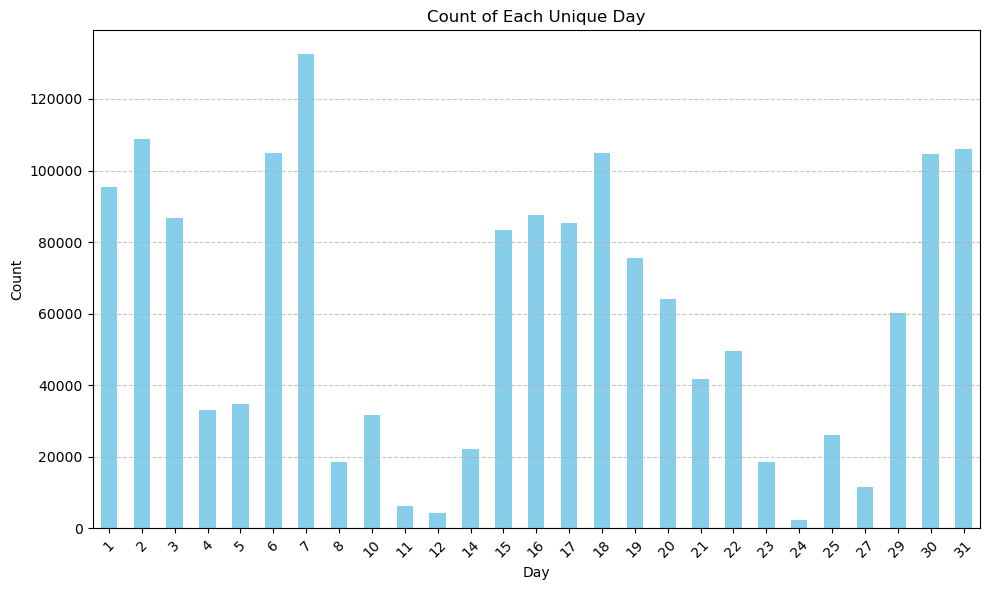


Figure 14:Histogram of the Date variable

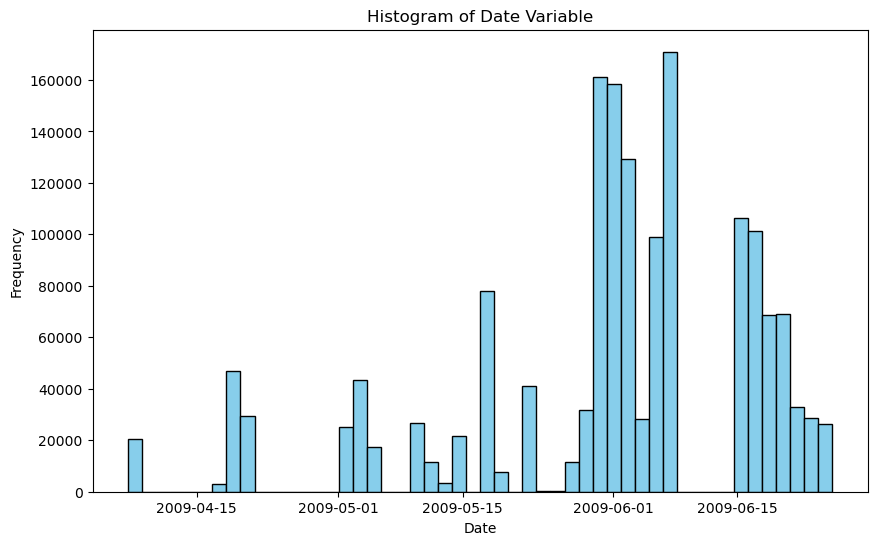
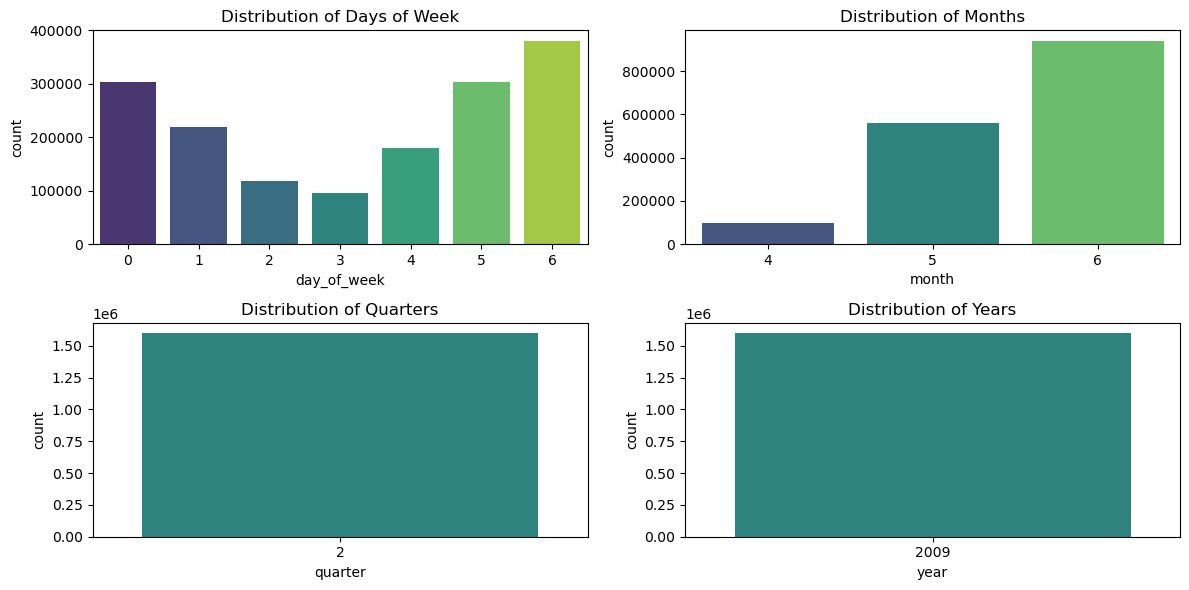


Figure 15: Distribution of the Different time components in the date variable



From the graphs above, The date variable has only 3 months, April, May and June.

**SENTIMENT EXTRACTION AND ANALYSIS**

After text processing, the sentiments of the text were extracted. Two methods were explored, text blob and Vader sentiment.

**Text blob sentiment** –

### Evaluation and justification of the hyperparameter tuning techniques that you have used [0-20]

### Your analysis of any change sentiment that occurs and your forecast of the sentiment at 1 day, 3 days and 7 days going forward[0-20]

### Presentation of results by making appropriate use of figures along with caption, tables, etc and your dashboard for your forecast, Discuss Tufts Principles in relation to your Dashboard .[0-20]

### Provide evidence and justification of your choice of sentiment extraction techniques.

### Explore at least 2 methods of time-series forecasting including at least 1 Neural Network and 1 autoregressive model (ARIMA, SARIMA etc…) . (Hint: that this is a Short time series, How are you going to handle this?)

### Evidence and justify your choices for your final analysis and include your forecasts at 1 day, 3 days and 7 days going forward.

### Your dashboard must be dynamic and interactive. Include your design rationale expressing Tufts principles.

#### Use Tufte Principle

According to Edward Tufte, there are 6 principles a visualization should strive toward, that is comparison rather than description, high resolution and utilization of classic designs, Content focus, concepts proven by time, and integrity. (Globus, 2014). From machine learning, UK untransformed Label encoded data was the best in making predictions about the number of passengers for some models not all while untransformed Label encoded data was the best in making predictions about the number of passengers for Ireland data.

**Dashboard**

The tufte principle was utilized to create a dashboard that communicates findings to different Air transportation stakeholders that is passengers, the Data Science team, and aircraft companies.

Information on the comparison of the best models was plotted using a bar graph and the most and least common aircraft were also plotted for national and International Transport. Passengers are required to know the most common and least common aircraft to help in managing their travels. A bar plot was used because the data was categorical. Findings from Machine learning were relevant to the data science team since they could know which model is better when making predictions about air passengers’ numbers. Aircraft companies need to know which aircraft is common to ensure targeted Ref Jupyter Notebook