Sentiment analysis and forecast

1. Data storage – tweets data set

* Comparing loading data set to Cassandra, MongoDB and MySQL

1. Data manipulation and saving the results

* Cassandra & Pyspark
* MongoDB – sample aggregation

1. Data storage comparison (workloads)

- MongoDB

- MySQL

●           Utilisation of a distributed data processing environment (e.g., Hadoop Map-reduce or Spark), for some part of the analysis.

●           Source dataset(s) can be stored into an appropriate SQL/ NoSQL database(s) prior to processing by MapReduce / Spark (HBase / HIVE / Spark SQL /Cassandra / MongoDB / etc.) The data can be populated into the NoSQL database using an appropriate tool (Hadoop/ Spark etc.)

**●           Post Map-reduce processing dataset(s) can be stored into an appropriate NoSQL database(s) (Follow a similar choice as in the previous step)**

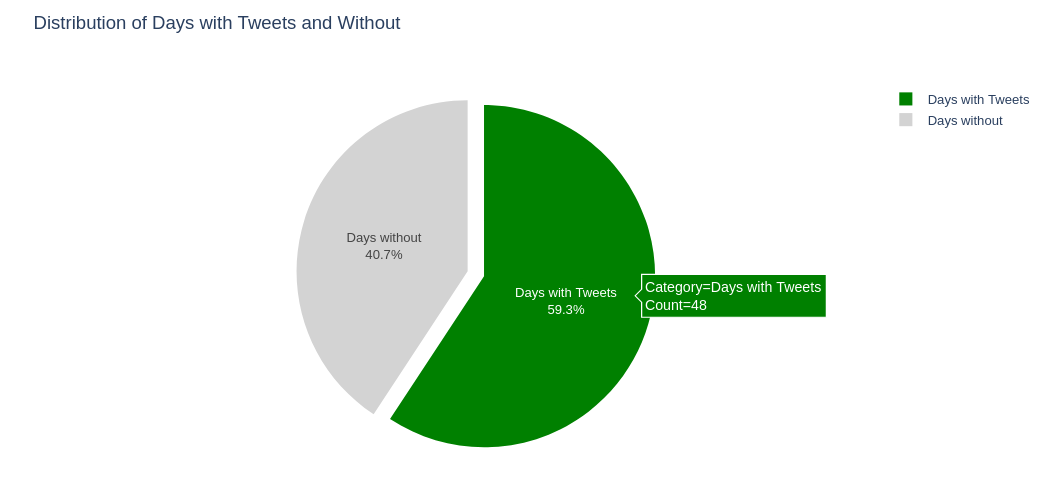
●           Store the data and then follow-up analysis on the output data. It can be extracted from the NoSQL database into another format, using an appropriate tool, if necessary (e.g. extract to CSV to import into R/ Python etc.).

●           Devise and implement a test strategy in order to perform a comparative analysis of the capabilities of any two databases (MySQL, MongoDB, Cassandra, HBase and CouchDB) in terms of the performance. You should record a set of appropriate metrics and perform a quantitative analysis for comparison purposes between the two chosen database systems.

1. EDA (sentiment data set)

The sentiment Forecasting process started with Explanatory Data analysis of the sentiment data set. EDA results show that 1,600,000 entries across a period of 80 days, ranging from the oldest date, 2009-04-06, to the newest date, 2009-06-25. The dataset displays variations in tweet counts per day, with some days witnessing a high volume of tweets, such as 2009-05-17, with 40,154 tweets, while others have significantly fewer, like 2009-04-06, with 3,360 tweets, or even zero.

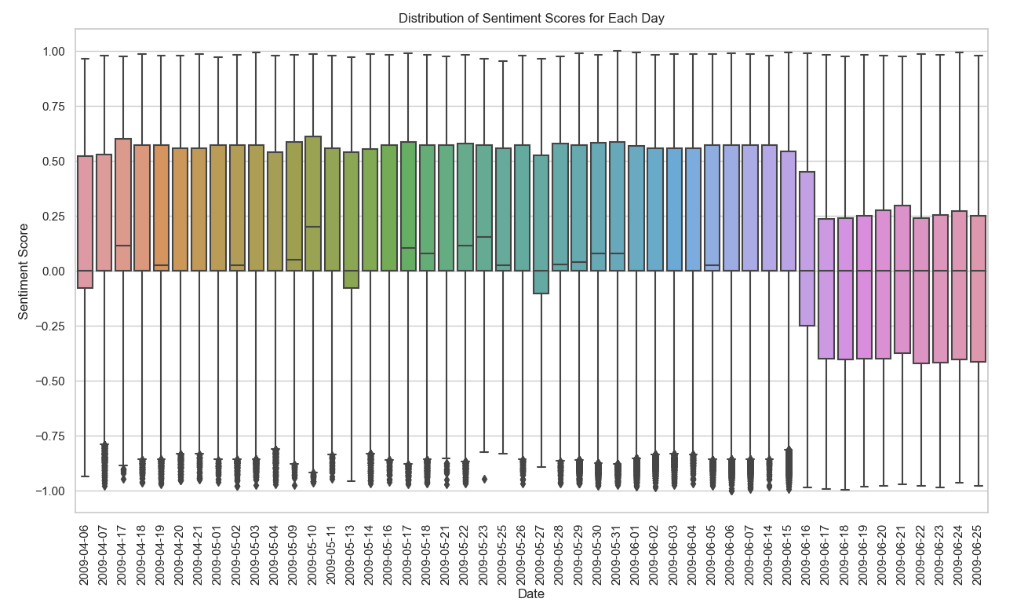
An analysis of missing dates in the dataset indicates that out of the 48 days covered, 33 days were absent. The ratio of days with values to days with null values is approximately 1.45 which will require further investigation and handling before proceeding with creating a forecast.



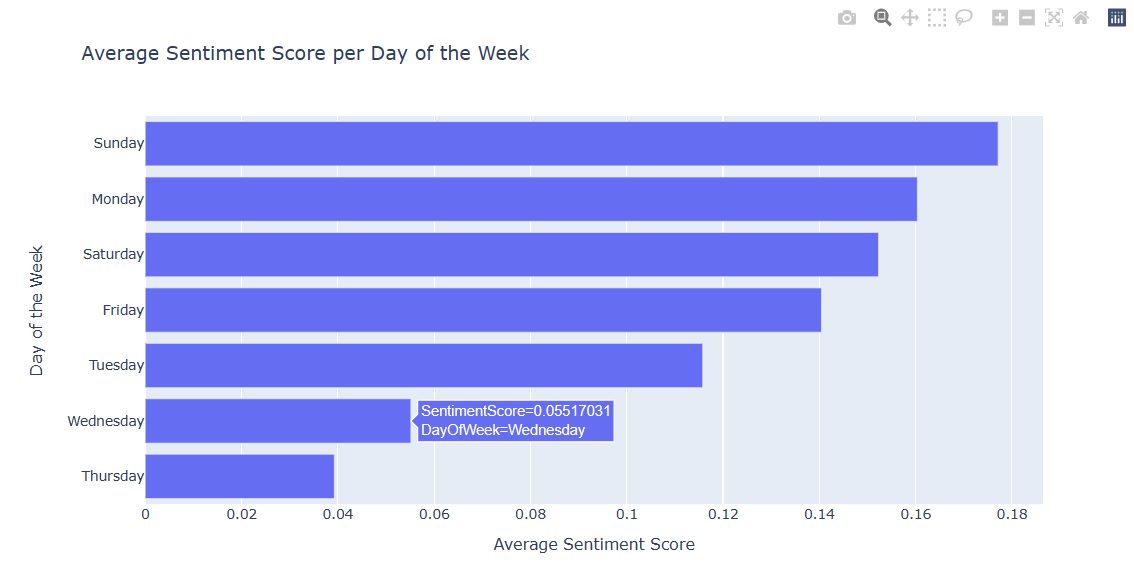
1. Sentiment analysis

* Distribution
* Average (including before and after 2009-06-17), Tkinter
* Sentiment per day

The sentiment analysis of the dataset provides valuable insights into the distribution of sentiment scores, average sentiments, and other relevant information. The sentiment scores show a very wide range, with the highest sentiment score at 0.9987 and the lowest sentiment score at -0.9985. This wide distribution of sentiment scores suggests a diverse range of opinions and emotions expressed in the data. The overall average sentiment across the dataset is approximately 0.13, indicating a slightly positive sentiment on average.

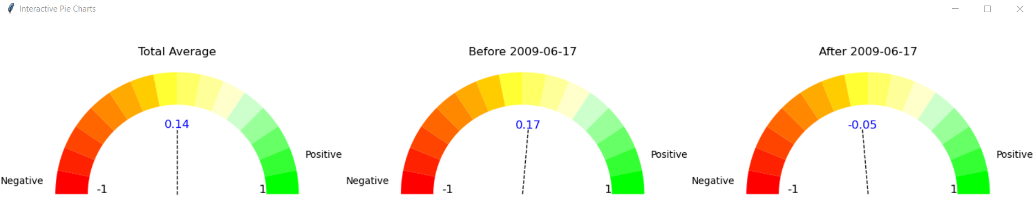


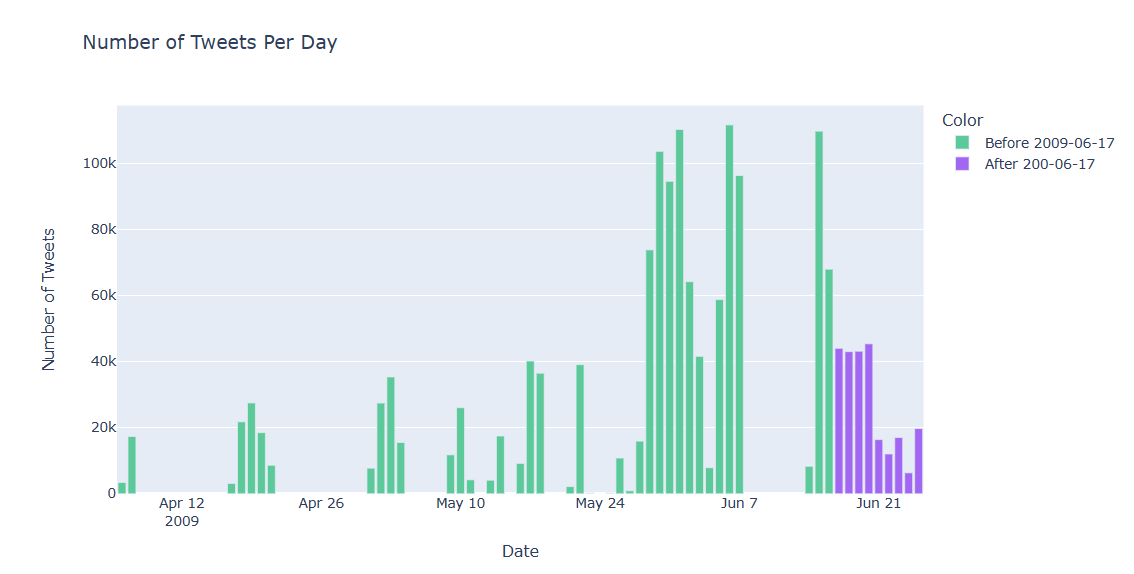
Additionally, the distribution of sentiment scores across different days of the week provides interesting insights. The average sentiment scores for each day of the week are presented below:



This could suggest that people are more prone to write negative tweets on Wednesday and Thursday. However, it is important to note that this observation may not be entirely accurate, as the dataset contains many missing data points.

Further examination of the dataset shows that, before June 17, 2009, there are approximately 18,793 tweets per day on average, with a positive sentiment score of 0.17. However, after that date, the average number of tweets per day increases to 30,860, accompanied by a shift towards a negative sentiment score of -0.05.





Lastly, 3.51% of the entire dataset, equivalent to 56,092 rows, has sentiment scores lower than -0.7 and the majority of them are after 2009-06-17. This subset of data could indicate the presence of a specific factor that is causing strongly negative sentiments within the dataset.

This transition in sentiment potentially signifies a shift in the sentiment expressed in the data around that date. Notably, the most significant drop in sentiment occurs on Wednesdays and Thursdays which would explain average sentiment distribution over days of the week.

Considering the insights obtained from the sentiment analysis, there is a need to delve into the content of the tweets. This investigation is driven by the necessity to search for potential reasons or factors that could explain the observed shifts in sentiment. Understanding these underlying factors is necessary, as they may have a lasting impact on the long-term sentiment forecast.

1. Data context

* Stop words
* Finding topics within the data set
* Analysing the results and conclusion

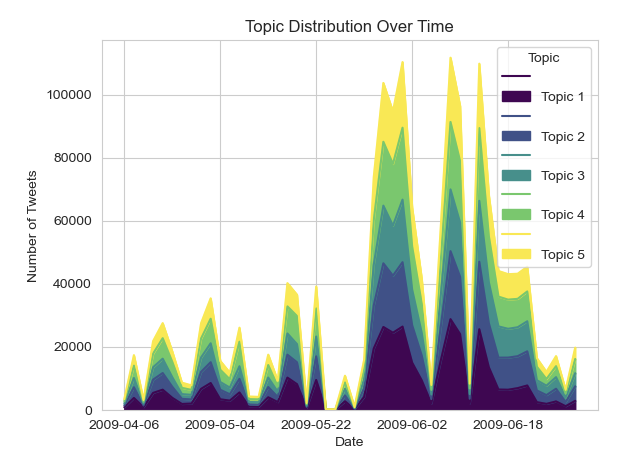
The analysis of tweet content begins with the utilization of the original tweet dataset. A series of data preprocessing steps were performed to refine and extract valuable information from the dataset.

Text data undergoes text preprocessing steps using the NLTK library. This includes tokenization, lowercasing, and the removal of stop words to prepare the tweet text for further analysis.

Afterward, a Document-Term Matrix (DTM) is constructed, which quantifies the frequency of words in the tweet text. The DTM is created using CountVectorizer, considering a maximum of 1000 features.

Next, Latent Dirichlet Allocation (LDA), a topic modeling technique, is applied to identify latent topics within the tweet content. This method helps categorize tweets into meaningful topics by analysing the word distribution within each topic. The top 10 words associated with each topic are extracted but unfortunately, none of them can present a strong meaning or connection to the drop of sentiment.

The initial quality of the topics appeared to be insufficient in providing any meaningful insights. Even with comprehensive analysis, it is evident that the topics are relatively evenly distributed across the entire dataset. The results do not uncover dominant themes that could potentially explain the rapid shifts in sentiment.



1. Sentiment Forecast

* Handling missing data
* Seasonal decomposition
* Stationarity check
* Checking Arima, ETS, and Rolling method and selecting the best for the forecast

To conduct sentiment forecasting, a thorough data check was initially performed to ensure data integrity, focusing on the presence of missing data points. The goal was to initially work with a dataset that had the least amount of missing data. The dataset exhibited 22 missing values before 2009-05-09 and 11 missing values after this date. This led to the selection of the dataset 'after 2009-05-09' for examination of data trend, seasonality, and residuals.

A graph of different types of data

Description automatically generated with medium confidence

The results obtained from the additive decomposition of the 'AverageSentimentScore' time series shows evidence that the 'trend' component displays a notable persistence and consistent trajectory until 2009-06-17, suggesting the presence of a strong underlying sentiment trend.

A graph showing different colored lines

Description automatically generated

Furthermore, the 'seas' component exhibits relatively minor variations. These illustrate that the seasonal component carries a moderate influence on sentiment trends, but it does not overshadow the dominant impact of the 'trend.'

On the other hand, changes in sentiment scores may be influenced by a large amount of missing data that was filled in. These occasional shifts in sentiment, suggest that not all the ups and downs in sentiment can be explained by the main trend and seasonal patterns. It is possible that when the missing data was filled, it made the sentiment data look smoother and caused these small changes. So, some of the changes in sentiment might be due to the missing data and how they were filled in.

1. Dashboard

* Putting the most suitable charts and forecast into a dashboard