School of Computing

CS4225/CS5425
Big Data Systems for Data Science

PROJECT REPORT

Analysis of public sentiment from social media to assist policy-makers on COVID related policies

Name	**Student ID**
Akhil Venkateswaran Lakshminarayanan	A0228512L
Ankireddy Monica Aiswarya	A0228502M
Kwek Kee En	A0189845H
Lau Wen Hao	A0121528M
Niranjana Anand Unnithan	A0228601M
Ong Fang See Christopher	A0211004J

Table of Contents

1 Introduction

The COVID pandemic, in its severity, has exposed the weakness in policy making under a crisis. In the past, policy decisions on health issues were often well-surveyed across the population to understand the impacts of how a policy affects the public positively. However, the ongoing COVID pandemic has not afforded policymakers the luxury of time to assess the implications of a health policy decision and to arrive at a consensus with the general public. In some countries where public trust and political legitimacy are lacking, the implementation of strict COVID rules can exacerbate political discontent against an authoritarian government perceived to exert more control over the state. As a result, socio-political tensions can quickly spill over into mass street protests, leading to a congregation of a large number of people and facilitating the spread of COVID. Thus far, policy responsiveness towards COVID has been a simple case of bringing down as many COVID cases in the shortest amount of time, excluding the impacts of the psychological effect strict COVID restriction can have on the population due to social isolation.

On this front, the team recognizes the potential of social platforms that can be utilised to perform sentiment analysis to uncover insights on how the public perceives the current COVID situation and is of the view that such data can be used to furnish policymakers with the information necessary to implement the appropriate COVID policy with regards to public sentiments.

2 Literature Review

Twitter is one of the most dominant microblogging social media platforms on the web, with a large user base due to its more liberal privacy settings. This platform is used by people primarily to express their opinions on current affairs, promote products, news reporting and so on. Using the hashtag feature, one can find people who share similar interests or track thousands of tweets filtered by a topic of their choice. This is especially beneficial in the field of market research, where product companies can use tweets to monitor customers' feedback before launching to gauge the profitability of a product. They can further use this data to make improvements to their product based on negative feedback. A widely used NLP technique - Sentiment Analysis, is used to classify the tone of a tweet into positive, negative, or neutral. This opinion mining method has been leveraged across several domains. In the hospitality and tourism industry, there was a study conducted using Las Vegas resorts related tweets to understand the public buzz across different firms. This will enable hospitality operators to narrow down on the areas to improve on while compared against other firms [1]. In the healthcare industry, with the onset of the COVID pandemic, a real-time system was developed to learn about people’s sentiment on the pandemic using Twitter streaming data [2][3]. Furthermore, when the vaccines were launched initially, tweets were used in Iran to understand the citizens' judgement on being administered vaccines. An upward trend in negative sentiment in the coming months was observed. This data can help the respective authorities to promote positive tweets about vaccination so as to encourage the citizens to get vaccinated [4].

As the pandemic has impacted on a global scale, every country devised different measures after assessing the situation in their country. Prior to COVID, surveys were used as a major source of data to gauge the process of political decision making. But with the uncertainty of trends in COVID situation, surveys are no longer a viable option as they are time-consuming. Thus, sentiment analysis of social media has gained popularity in this aspect. To evaluate the causal relation of how people’s sentiment influences policy decision making, there was a study conducted in the US by collecting sentiment data from tweets, COVID numbers, policies introduced and discovering which factors contribute the most to policy changes [5].

Our research is headed in the reverse causal direction (ie) how policy changes impact people’s sentiment. We aim to study the trend of public opinion for a fixed time period before and after the introduction of a policy. This study will aid the policy makers in adopting policies that will be in the best of people’s interests.

3 Methodology and Experimentation

3.1 Dataset Description

3.1.1 Training Dataset

The team obtained a large set of labelled Twitter data (~24GB) which described the latent topic, sentiment and emotion attributes of COVID related tweets made throughout the pandemic. The below describes the attributes provided in the dataset.

Nominal		Continuous		Categorical		
Tweet ID	User ID	Timestamp	Emotion intensities	Keyword	Country	Sentiment

Figure 1: Datatypes in the labelled training dataset

However, the dataset required preprocessing before it was used for model training. The dataset consists of tweet ID instead of the actual tweet text. As such, the team extracted tweet ID within the dataset on the Twitter platform and hydrated it with the actual tweet text. The below describe the preparation step used to hydrate the dataset.

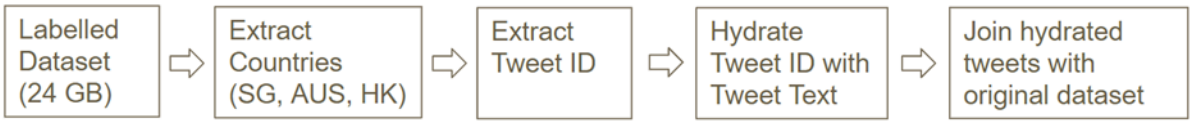

Figure 2: Hydrating the dataset with tweet text

3.1.2 Streaming Dataset

The proposed streaming dataset consists of two sources: Twitter and Reddit.

However, during implementation of the model, it was observed that the number of COVID related posts on Reddit have been dwindling and no longer provide sufficiently large data to achieve a reasonable estimate of the model performance or prediction. In hindsight, this development is expected as many countries have begun to lift COVID related restrictions and embraced a post-pandemic future hence public interest in the pandemic is expected to reduce substantially over time.

Therefore, the team decided to drop streaming data from Reddit and instead focus only on streaming tweets from Twitter. The tweets were obtained through the Twitter streaming API and were filtered with specific pandemic keywords “covid”, “wuhan”, “nCov’, and “corona” that were also used in the original dataset. The tweets were extracted for Singapore, Hong Kong, Australia and the USA.

3.1.3 Policies Dataset

COVID related policies data across 3 countries (Singapore, Hong Kong and Australia) were gathered and compiled into a timeline to form a meaningful comparison of sentiment between different periods or policies.

COVID related policies for the USA were not collected as it was tedious to collect COVID-related policies for the USA, given the large number of states and non-uniformity of the policies implemented across states.

The sentiment trends around an implemented policy will be being discussed under the result section of this report.

3.1.4 4V Data Challenges

Challenges	Justification
Volume	After the dataset was hydrated with tweet texts for Singapore, Hong Kong, Australia and USA, the dataset size was significantly large (~ 14GB) and the dataset required to be split and processed in batches.
Variety	Data sources from Twitter and Reddit were initially proposed to be used. The reason for dropping the reddit data is as covered in the above section.
Velocity	COVID related tweets are constantly being pulled into the implemented data pipeline through the Twitter streaming API.
Veracity	A tweet is generally an expression of an individual disposition towards a certain topic. The aggregation of results from a large group of tweets can provide an overall general sentiment trend of the public and reduce the risk of the result being affected by individual biases.

Table 1: 4V Data Challenges

3.2 Overall Architecture

In the subsequent sections, each part of the architecture will be described in detail.

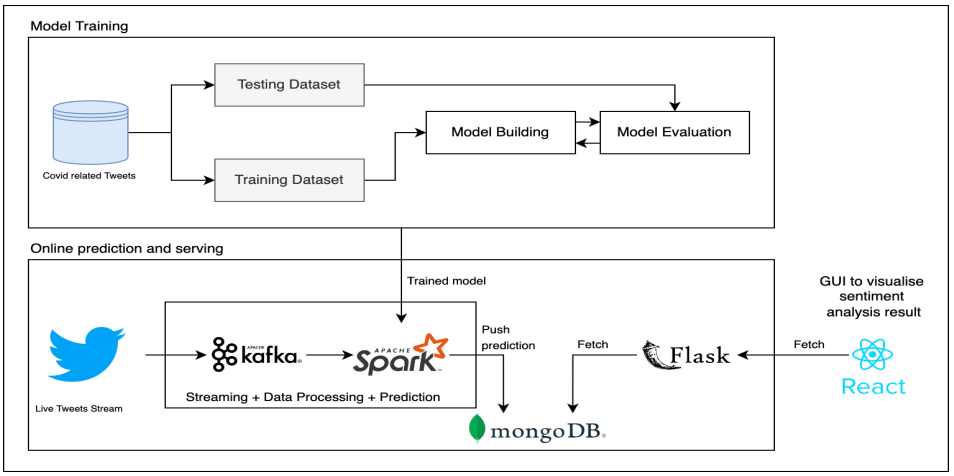

Figure 3: Architecture Diagram

3.3. Training Architecture

3.3.1. Dataset Cleaning and Preprocessing

Raw tweets without preprocessing were highly unstructured and contained redundant information. Hence, the team adopted the following steps for cleaning tweets and preprocessing:

3.3.1.1 Dataset Cleaning

- Removal of fields with null values for text
- Removal of hashtags, hyperlinks, handle references, punctuations, and other irrelevant information that does not provide meaningful context for sentiment analysis
- Dropping all columns except 'text' and 'sentiment' from the dataset
- It was observed that classes such as 'very positive' and 'very negative' are underrepresented. Hence, we combined 'very positive' class with 'positive class' and 'very negative' class with 'negative class'. We also dropped the neutral class as it had less number of samples.

- Undersampling: It was observed that the number of samples belonging to the 'negative class' is much greater than the number of samples belonging to the 'positive class'. We performed undersampling of the 'negative class' to make the class distribution balanced and prevent the model from being biassed.

3.3.1.2 Text Preprocessing

A text preprocessing pipeline was set up using SparkNLP to convert the tweets to a form that the model accepts. Figure 2 shows the steps that were followed in the pipeline. DocumentAssembler() is the entry point to SparkNLP annotators, which transforms the raw data to document type. A SentenceDetector() is used to find the sentence bounds in the document, followed by a Tokenizer() that creates tokens from sentences. Stop words are removed with the help of StopWordCleaner(). The tokens thus obtained are normalised and converted to TF- IDF format using Finisher(), HashingTF() and IDF(). The TF-IDF form of the input text is used to train the model.

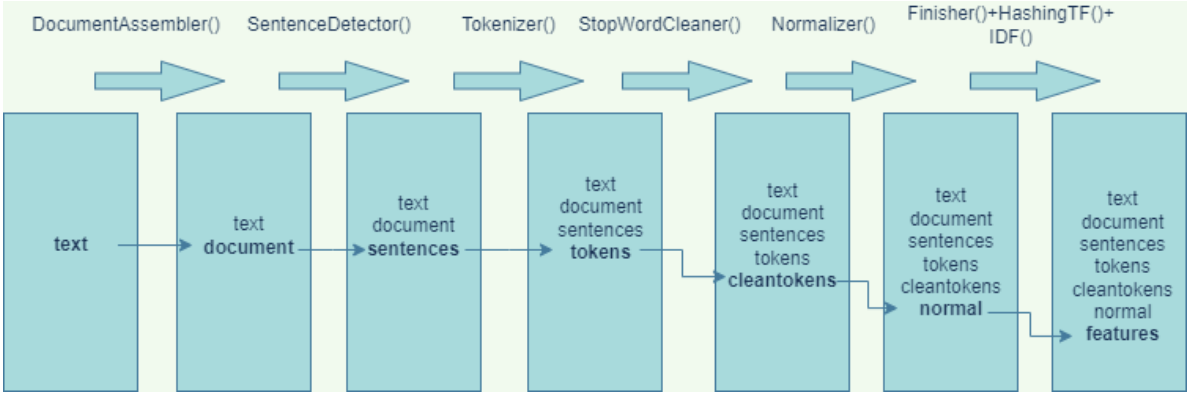

Figure 4: SparkNLP Pipeline for text preprocessing

3.3.2 Exploratory Data Analysis

The figures below (Figure 5 and Figure 6) show the top 30 most frequent words in the tweets in Hong Kong and Singapore. It can be observed that there are several similar words that are related to COVID being commonly used in both countries.

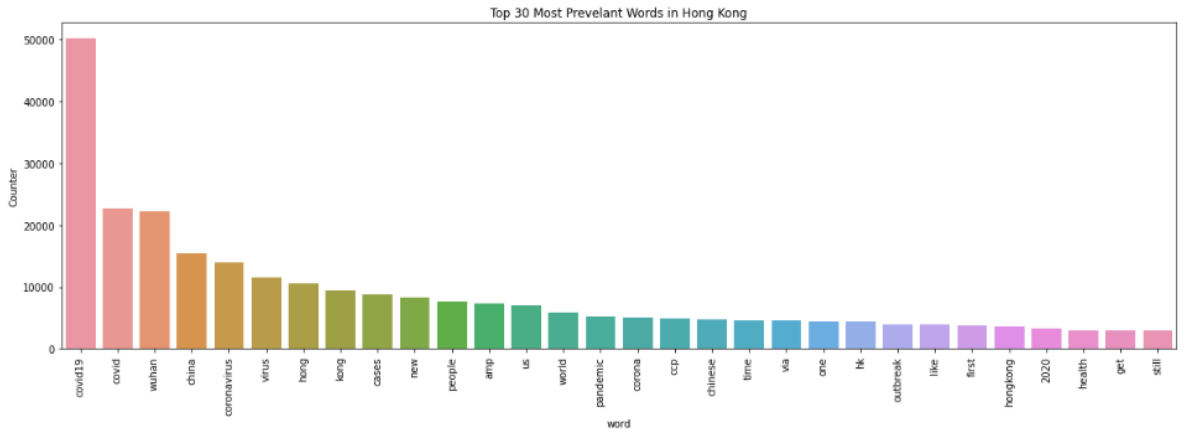

Figure 5: Top 30 most prevalent words in Hong Kong Dataset

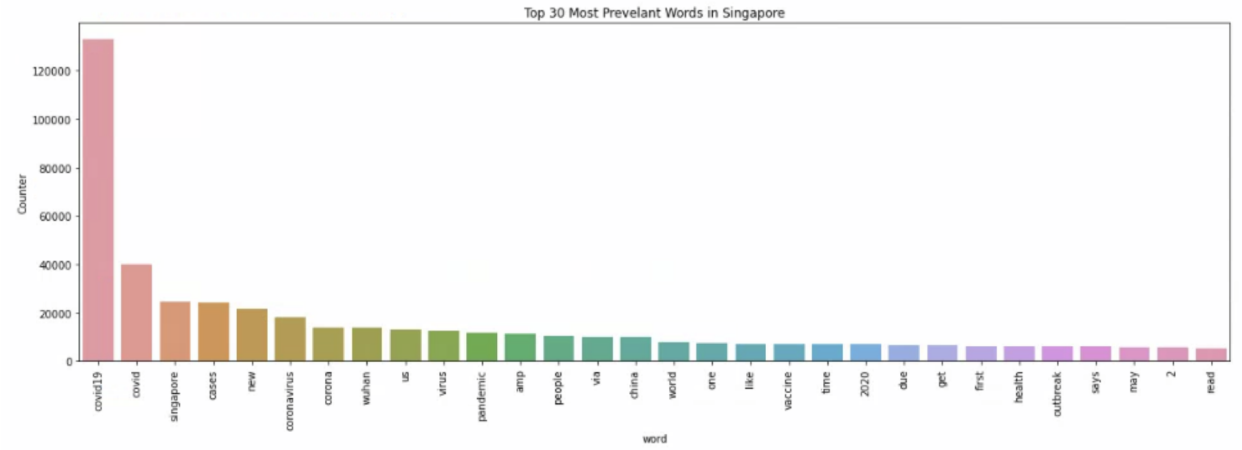

Figure 6: Top 30 most prevalent words in Singapore dataset

3.3.3 Models

Different models were trained for each of the countries to avoid linguistic biases. The models were created for Hong Kong, Singapore and Australia. The team also experimented with different machine learning and deep learning models as well as feature engineering techniques to identify the best methodology.

3.3.3.1 SVM

Support Vector Machine models were created for each of the countries using default parameters. The input to the SVM is the tweets converted to tf-idf form obtained from the SparkNLP pipeline. The models were trained to classify tweets as positive and negative based on the underlying sentiment. Table 1 depicts the evaluation metrics for the country-specific SVM models.

Dataset	Precision*	Recall*	F1 Score*	Accuracy
Hong Kong	0.83	0.83	0.83	0.827
Singapore	0.86	0.84	0.85	0.857
Australia	0.85	0.85	0.85	0.851

Table 2: Evaluation Metrics for SVM model

*macro-averaged

3.3.3.2 Word2Vec with LSTM

- Tweets were first cleaned by removing hashtags, ampersands, quotation marks, and other such symbols
- Using NLTK library, the words were tokenized, stopwords were removed and lemmatized
- Using a pre-trained word2vec model, a representation in tensor form was obtained for each tweet
- LSTM network was chosen for training and prediction as it is a recurrent neural network that performs well in terms of memory. It has the capacity to hold patterns of long sequences in its memory
- The input to the LSTM network is the 3-Dimensional Word2Vec tensor
- The outputs are the 2 classes: Positive and Negative

Drawbacks

- The accuracy of the model on the HK dataset was only 50%.

- High-dimensionality of the Word2Vec tensor size causes “Out of memory error”

3.3.3.3 BERT/Glove/ELMO with DLClassifier

- Standard tweet cleaning and preprocessing steps described in section 3.3.1 were followed.
- Pretrained embeddings such as BERT, Glove and ELMO were used to represent the words instead of tf-idf vectorizer. The embeddings are able to capture the contextual information present in the text
- Deep Learning Classifier provided by SparkNLP was used to train the data and tweets were classified to Positive and Negative classes based on sentiments

Drawbacks

- The high dimensionality of Language Embeddings made it difficult to train datasets of large size leading to “Out of memory error” and long training time.

3.3.4 Overall Models Comparison

SoC cluster was used to train all the models as there was a requirement for large RAM and GPU. As summarised in Table 3, SVM was found to be giving the best accuracy and system performance; SVM was chosen to train on each country’s dataset.

Metrics/Model	SVM with TF-IDF	BERT	ELMo	Word2Vec with LSTM
Accuracy	83%	80%	82%	50%
Speed of training	Fastest	Slow	Slowest	Relatively fast
Limitations	Not suitable for large dataset - No significant increase in accuracy for larger dataset	Out of memory error for large dataset on cluster	Out of memory error for large dataset on cluster	- Poor accuracy - High dimension Word2vec tensor

Table 3: Overall Models Comparison

3.4 Serving Architecture

The serving architecture after the model is trained. As observed from Figure 3, there are several key components in the proposed architecture. All these are running in Docker containers and are orchestrated with Docker Compose locally for ease of development during the prototyping phase.

3.4.1 Kafka Streaming

Firstly, Apache Kafka was used to set up the streaming platform for this project. Apache Kafka is a distributed event store and a stream processing platform. There are 4 main parts in a Kafka system:

1. Broker - Handles all the requests from the clients and keeps the data replicated within the cluster.
2. Zookeeper - A highly reliable distributed configuration server that provides configuration information, naming, synchronisation and group services. Kafka uses it to keep states of the cluster, info about the broker, the topics and the users.
3. Producer - The component that sends the messages to the broker. Producers create the messages and send them to specific topics in the broker, which are like categories.
4. Consumer - The component that consumes messages sent by the producers. There can be more than one consumer consuming from a topic, which is called a consumer group.

As of last year, Apache Kafka has replaced Zookeeper with a self-managed quorum that implements the Raft algorithm that is seen in most distributed systems. However we still chose to set up Kafka with the Zookeeper approach as this new feature is still in early access and there are more abundant resources for setting up Kafka with Zookeeper.

3.4.2 Kafka Producer

For this project, the team applied for the Twitter Developer API access, which grants elevated access to use the Twitter APIs at a higher capacity. The producer is implemented in Python and tweepy Python library was used to set up a stream for the live tweets. These tweets were filtered according to the 4 keywords that were used in the original dataset and are also filtered by location using the streaming rules provided by the Twitter API. The producer then sends these tweets to the appropriate Kafka topics. 3 topics were set up, one for each of the countries to investigate, namely Singapore, Hong Kong and Australia.

3.4.3 Kafka Consumer

There were 3 consumer groups that were set up to consume from the 3 topics. During startup, the consumers will load the model corresponding to the country they are assigned to consume from. When a tweet is available in the Kafka broker, the consumer will consume the tweet, preprocess it with the SparkNLP pipeline mentioned earlier, and lastly perform the sentiment prediction using the model. The results will then be inserted into the database of choice, which is MongoDB.

3.4.4 Database

MongoDB is a NoSQL document-oriented database. NoSQL database was chosen by the team as there were no complex relationships in the data used [8] and further MongoDB was used as it is fast and built to scale up quickly. 2 databases were set up, named *tweets* and *sentiments*. The *tweets* database contains the raw tweets and the sentiment predicted for those tweets. A Cron job was set up to run everyday to aggregate the sentiments (distribution of positive and negative sentiments) for the past day and this result will be inserted into the *sentiments* database.

For each database, 3 collections were created, one for each country. These collections are indexed by the date as all the queries involve the date, and doing this greatly speeds up the queries.

3.4.5 Backend Server

The backend server was implemented using the micro web framework Flask which was written in python. The backend consists of 4 main APIs regarding sentiment analysis.

1. GET /sentiment/{country}?start={YYYY-MM-DD}&end={YYYY-MM-DD}
   Returns the aggregated sentiments for each day in the queried country and date range, for the GUI to plot the time series graph.
2. GET /sentiment/{country}/past-24h
3. GET /sentiment/{country}/past-7-days
4. GET /sentiment/{country}/past-30-days
   Returns the aggregated sentiments for the past 24 hours, past 7 days and past 30 days for the queried country which is updated while live tweets are being streamed in.

3.4.6 System Analysis

Every component chosen in this architecture is able to scale out horizontally and independently as usage and traffic increases and is also platform agnostic, meaning it can run on any platform. Multiple Zookeepers and Kafka brokers can run in a cluster mode. There can be multiple consumers per consumer group consuming from the same topic and the messages can be set to be consumed only once and these states will be maintained in the brokers to know which message the consumer group should consume next. MongoDB can run with a Replica Set (minimum of 3 nodes) that maintains the same data set to ensure data redundancy and high availability of data. Read and write operations can

be distributed across all the instances. The backend API server is stateless so running multiple replicas of the same backend is also possible.

The ideal deployment is in a cluster with a scaling policy that triggers when certain specified CPU and memory thresholds are reached. A time based scheduled scaling policy can also be used in parallel as there are noticeably fewer tweets during midnight as compared to in the day. An example of this can be done in a Kubernetes cluster. In the event that data volume scales up 10 times, this system architecture will still be able to handle the load as these components are able to scale out independently of each other. Scaling out also provides high availability while increasing the fault tolerance.

One exception to this scalability is the Kafka Producer, which the team thinks is the only bottleneck in the system. There's only one Twitter developer account with one set of API keys to set up the Twitter stream. Hence, the team can only set up one stream. Even if multiple accounts are created, there is a limitation of one stream per account and there is no mechanism to coordinate multiple streams from the Twitter APIs, i.e. there will be duplicated data from the different streams over the same filters. So assuming that there is one stream per filter condition (4 keywords and 3 countries), there can only be a maximum of 12 streams. These streams will also not have an even distribution of traffic and they can only scale vertically instead of horizontally.

One more limitation in addition to that is even with elevated access granted by Twitter, the team is limited to 2 million tweets in a month at a rate of 50 tweets per second. So assuming that there is no lack of Twitter traffic (there are at least 50 tweets available per second), the quota provided will be used for the month after slightly over 11 hours.

3.5 Visualisation

The front-end was developed using ReactJs, which is a javascript library for building user interfaces. ReactJs uses virtual-dom, which is a lightweight representation of the real dom in the memory. Since the front end regularly fetches the live stream data from the backend and updates the UI asynchronously, ReactJs was chosen because it is efficient and faster than other available UI libraries.

3.5.1 Sentiment for the past 24hr, 7 days and 30 days

The front-end UI shows the percentage of positive and negative sentiment among people in the selected country for the past 24 hours, 7 days and 30 days. The real time tweets from twitter are streamed and stored in the back-end, which are then fetched using the react axios library asynchronously and updated in the UI.

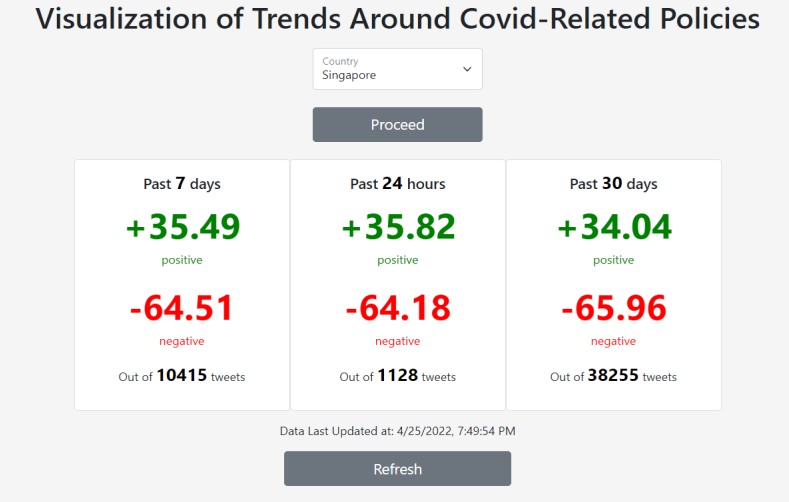

Figure 7: Percentage of Twitter sentiment score for the past 24hrs/7days/30days in Singapore

3.5.2 Sentiment over a date-range

The UI also shows the trends of sentiment among the people of the selected country over a date-range, in the form of a time-series graph. The blue line denotes the percentage of negative sentiment score and green line denotes the percentage of positive sentiment score. The red line denotes the dates when important policy announcements regarding COVID were announced by the government. The dates were identified to analyse the trend of sentiment a few days before and after the decisions regarding COVID were taken.

Figure 8: Time series of tweets sentiment in Australia from 01-01-2021 to 25-08-2021

4. Discussion of Results

4.1 Policy which showed an upward trend in positive sentiment:

Figure 9: Time series of tweets sentiment in Singapore weeks before and after restrictions were lifted on 06-08-2021

An upward trend can be observed (Figure 9) in positive sentiment over a period of 2 months - one month prior to the announcement of policy and one month after the policy announcement. This shows that a policy-maker can draw meaningful conclusions about the impact a policy has had on people based on the sentiment.

4.2 Policy which showed a downward trend in positive sentiment:

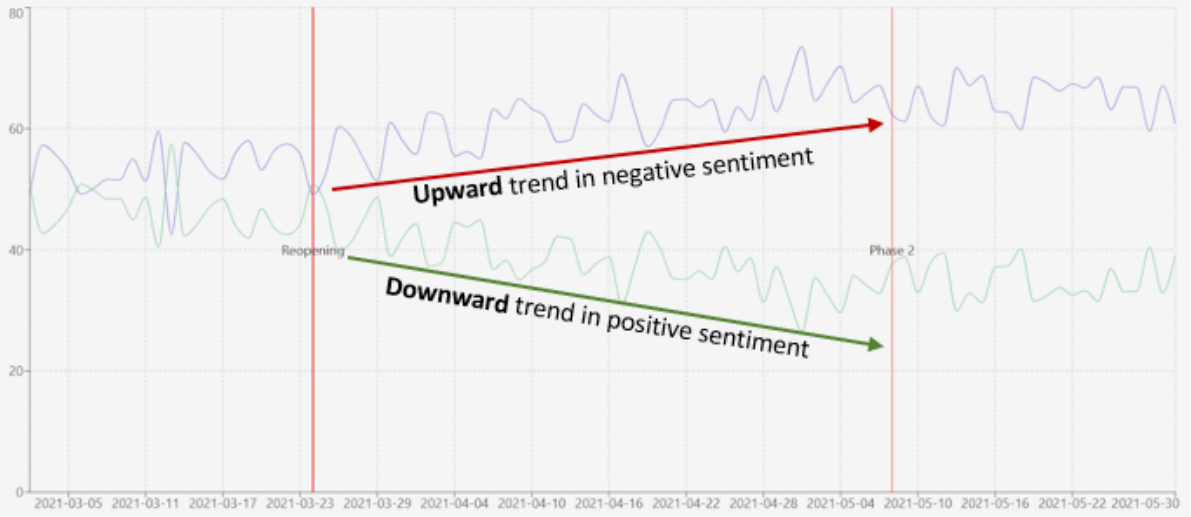

Figure 10: Time series of tweets sentiment in Singapore weeks before and after borders were reopened (on 24-03-2021) and return to Phase 2 (on 16-05-2021)

It can be observed (Figure 10) that there's a gradual decrease in positive sentiment between the days of the 2 policy announcements - "Reopening" and "Phase 2" were made. This could perhaps be correlated to a plausible surge in COVID cases as the borders were reopened, and hence leading to onset of Phase 2.

4.3 Problems Encountered and Lessons Learnt

1. Resource limitation
    a. Out of memory error on cluster for training on > 3 GB dataset. Unable to train the whole USA dataset (~14GB).
    b. SoC cluster was not able to connect to an external Kafka hence we're not able to utilise the cluster resource for stream processing.

2. Twitter API limits were quickly reached even with elevated access granted by Twitter as discussed earlier in 3.4.6.
    a. The team can only selectively bring up the cluster to work within the API limits.
    b. The team learnt that for any product utilising Twitter as a data source, the amount of tweets we can obtain is only sufficient for a proof of concept. To build a full product, we have to purchase higher limits.

3. While analysing the results to observe trends before and after a policy was implemented, the team realised data related to the number of COVID cases could have been included for cross-referencing and drawing significant conclusions. Inclusion of COVID daily cases number for analysis in addition to policy-related data will be considered in the future.

4. Currently, the trends in sentiment were manually analysed based on the visualisations using the GUI, however, this may not be entirely accurate. In the future, we can implement statistical analysis to provide more insights and numerically draw significant conclusions for each policy.

5. Personal Contributions

Team member	Contributions (%)
Akhil Venkateswaran Lakshminarayanan	12
Ankireddy Monica Aiswarya	15
Kwek Kee En	12
Lau Wen Hao	30
Niranjana Anand Unnithan	15
Ong Fang See Christopher	16

6. Project Summary

Although there have been studies regarding the public sentiments predicted using tweets related to COVID, this project takes a fresh angle on analysing the public sentiments on COVID-related policies using social media. Given the nature of social media, the data may be consequently inaccurate of the public sentiments thus cleaning the data was crucial for the team. Substantial static Twitter dataset was used to train different machine learning models, and identify the best performing model that was used for real-time prediction on streamed tweets.

The temporal variations in public sentiments towards updates in COVID-related policies were analysed. The key changes in policies affecting the positive and negative sentiments on social media were mapped. This can help identify the policies which affected more negative sentiments for policymakers to study for potential interventions to allay the underlying public fears and concerns.

From our analysis of Figures 9 and 10, we conclude that we can observe that policy announcements can cause a discernible change in public sentiment. The change in trend can serve as an additional metric to assist policy makers in determining the general sentiment of the public. Tracking the affective states of citizens, especially during periods of disruptive events such as natural disaster or pandemic can be challenging due to the unpredictability and volatility of these crises. However, through this project, the team have delivered on a data-pipeline that shows that through the aggregation of expression on social media platforms, it is possible to monitor the general sentiment of the public. This data can then be used by the policy makers to help craft more effective health strategies that are able to garner more public support and have a higher chance of being implemented successfully.

7. References

[1] Kahlil Philander, YunYing Zhong, Twitter sentiment analysis: Capturing sentiment from integrated resort tweets, International Journal of Hospitality Management, Volume 55, 2016, Pages 16-24, ISSN 0278-4319, https://doi.org/10.1016/j.ijhm.2016.02.001.

[2] Zhang, Xiongwei & Saleh, Hager & Younis, Eman & Sahal, Radhya & Ali, Abdelmgeid. (2020). Predicting Coronavirus Pandemic in Real-Time Using Machine Learning and Big Data Streaming System. Complexity. 2020. 1-10. 10.1155/2020/6688912.

[3] László Nemes & Attila Kiss (2021) Social media sentiment analysis based on COVID-19, Journal of Information and Telecommunication, 5:1, 1-15, DOI: 10.1080/24751839.2020.1790793

[4] Zahra Bokaee Nezhad, Mohammad Ali Deihimi, Twitter sentiment analysis from Iran about COVID 19 vaccine, Diabetes & Metabolic Syndrome: Clinical Research & Reviews, Volume 16, Issue 1, 2022, 102367, ISSN 1871-4021, https://doi.org/10.1016/j.dsx.2021.102367.

[5] Zhijing Jin, Zeyu Peng, Tejas Vaidhya, Bernhard Schoelkopf, and Rada Mihalcea. 2021. Mining the Cause of Political Decision-Making from Social Media: A Case Study of COVID-19 Policies across the US States. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 288–301, Punta Cana, Dominican Republic. Association for Computational Linguistics.

[6] Gupta, Raj, Vishwanath, Ajay, and Yang, Yinping. COVID-19 Twitter Dataset with Latent Topics, Sentiments and Emotions Attributes. Ann Arbor, MI: Inter-university Consortium for Political and Social Research [distributor], 2021-11-04. https://doi.org/10.3886/E120321V11

[7] Kalron, A. (2020, January 16). How do Hadoop and Spark Stack Up? Logz.Io. https://logz.io/blog/hadoop-vs-spark/

[8] Chan, M. (2022, January 14). *SQL vs. NoSQL – what's the best option for your database needs?* Thorn Technologies. https://www.thorntech.com/sql-vs-nosql/

8. Workload

Team member	List of tasks
Akhil Venkateswaran Lakshminarayanan	- Analysis of Twitter and Reddit streams - Filtering tweets based on user location using tweepy - Initial investigation to read Twitter streams - Integration of backend and UI - Report Writing (Section 3.5)

Ankireddy Monica Aiswarya	- Tweets preprocessing and cleaning
	- Modelling
	  - Feature engineering
	  - EDA with Visualisation
	  - Implementing Word2Vec with LSTM
	  - Implementing TF-IDF with SVM
	- Exploring of GUI
	- Report writing (Section 2, 3.3.2, 3.3.3.2, 3.3.4)

Kwek Kee En	- Tweets preprocessing and cleaning - Modelling

	- TF-IDF with SVM on US tweets - Training different models and collecting experimental results - GUI prototype using plotly - Report Writing (Section 1, 2)
Lau Wen Hao	- Architecture design - Developed streaming component (producer) based on Akhil's investigation - Integrating preprocessing pipeline and models into consumer - Developed API server - Database design with MongoDB - Writing Dockerfiles for each custom component: API server, producer, consumer - Setting up the serving architecture with Docker compose - Wireframes for user interface - Report Writing (Section 3.4)
Niranjana Anand Unnithan	- Text Preprocessing and Cleaning - Modelling - Feature engineering - Bert, ELMO and Glove embeddings with DLClassifier - Setting up the cluster environment for training models - Training different models for each country - Generating predictions for test data - Report Writing (Section 3.3.1, 3.3.3.1, 3.3.3.3)
Ong Fang See Christopher	- Data exploratory on labelled dataset - Data collection - Hydrated and processed dataset in batches due to large dataset size - Developed Front-End GUI using ReactJS - Report writing (Section 3.1)

A Appendix

The code implementing the architecture described in this project can be found at https://github.com/wenhaohaoo/cs5425-project