COVID-19 Sentiment Analysis Dashboard on Tweets and Trending Hashtags

by

Azizul Qusyairin Bin Azman

17001950

Dissertation submitted in partial fulfilment of

the requirements for the

BACHELOR OF INFORMATION TECHNOLOGY

SEPTEMBER 2021

Universiti Teknologi PETRONAS

32610 Seri Iskandar

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

COVID-19 Sentiment Analysis Dashboard on Tweets and Trending Hashtags

By

Azizul Qusyairin Bin Azman

17001950

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS in partial
fulfilment of the requirement for the
BACHELOR OF INFORMATION
TECHNOLOGY

Approved by,

Dr Ahmad Sobri Hashim
Senior Lecturer
Computer & Information Sciences Department
Faculty of Science & Information Technology
Universiti Teknologi PETRONAS
Email: sobri.hashim@utp.edu.my

Dr. Ahmad Sobri Bin Hashim

UNIVERSITI TEKNOLOGI PETRONAS

SERI ISKANDAR, PERAK

SEPTEMBER 2021

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

AZIZUL QUSYAIRIN BIN AZMAN

Acknowledgement

I am very thankful to Allah the Almighty, the most beneficent, the most merciful and the most gracious, for giving me the strength and compassion to successfully complete this final year project.

First and foremost, I offer my sincerest gratitude to my supervisor, Dr. Ahmad Sobri bin Hashim, for his kind encouragement and support throughout my journey whilst allowing me the opportunity to explore and work in my own way. His invaluable constructive comments and suggestions throughout the project duration have contributed to the success of my final year project.

I wish to express my sincere thanks to Universiti Teknologi PETRONAS (UTP), especially the Computer and Information Sciences Department members and the Faculty of Science and Information Technology for providing the necessary academic and technical support for this project.

I would like to take this opportunity to dedicate my deepest appreciation to my friends and colleagues from Bachelor of Information Technology, class of September 2018, for the supports throughout my journey in UTP. Thank you for being there through thick and thin.

Lastly, I would like to express my deep thanks of gratitude thanks to the most important people in my life, my parents and family for the never-ending love, support, and patience. Thank you for always being there for me and believing in me which makes me become for what I am today. It would not have been possible to complete my study, especially this project, without their assistance and motivational support.

Abstract

Coronavirus disease 2019 (COVID-19) outbreak in Malaysia sparks different sentiment on its vaccination program, government policies and the rising number of cases especially in Twitter platform. It is well-known that Twitter has become an increasingly popular social media platform in expressing opinions, sharing interest and discuss the latest news. Thus, sentiment analytics can examine and classify the views from Twitter users to different polarity and categorization. The existing mechanisms to analyse people's opinions on COVID-19 are not adequate. Analysis on people’s opinions is especially crucial in any COVID-19 policymaking by the government to gradually reduce the impact of the pandemic towards the citizen. This project aims to develop a dashboard that can visualize the sentiment analytics results on COVID-19 related issues analysed from Twitter. The targeted issues that will be analysed are National Vaccination Programme, Movement Control Order (MCO) and the number of daily cases. The analysis comprises two main classification of the result which are polarity and subjectivity using Naïve-Bayes method. The analysis results are visualized in form of pie chart. On this basis, the index results are used to make a brief conclusion on the current sentiment of COVID-19 related issues in Malaysia. The dashboard provides a reliable analysis that reflects Malaysians’ needs in the current situation, thus the government bodies or the publics can plan COVID-19 countermeasure plans accordingly following current issues.

Table of Contents

Table of Figures

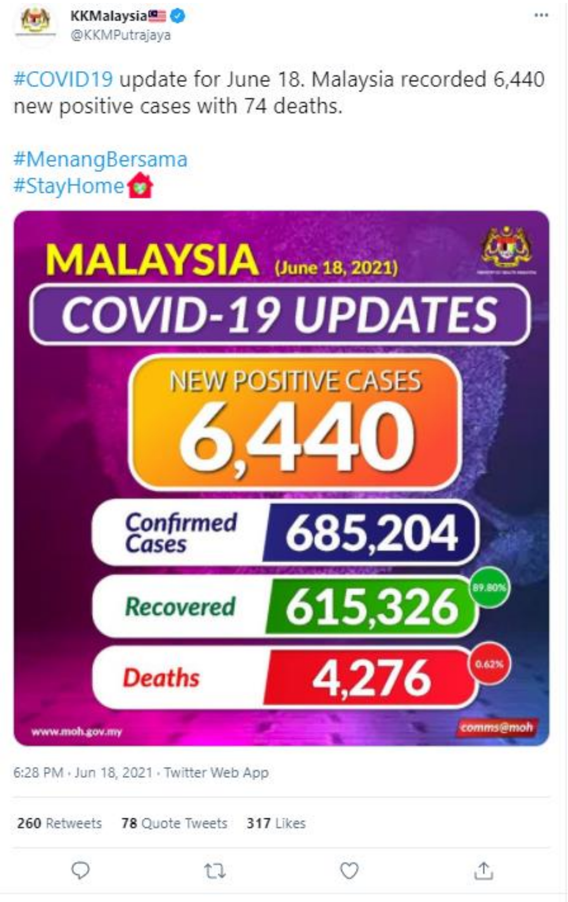
Table of Figures

CHAPTER 1
INTRODUCTION

1.1 Background of Study

COVID-19 issues have been discussed openly and widely in Twitter since the start of the pandemic February last year. Twitter is a famous microblogging website that allows users to read and create millions of short messages under a 140-character limit on any topic. Users who are well-known or prominent tweet their status updates, which are then retweeted, discussed, or reacted to by their followers. In some cases, a tweet from an influential person could affect the cryptocurrency or stock market (Ante, 2021). Thus, opinions on Twitter are crucial in performing sentiment analytics on COVID-19 issues.

In 2016, 26.6 percent of Malaysia's 21.9 million social media users had a Twitter account, and the number is growing in comparison to other social media platforms (Bakar et al., 2018). In addition, some Malaysian government officials and bodies have their own Twitter account such as TS Muhyiddin Yassin, Ministry of International Trade and Industry (MITI), and Ministry of Health (MOH). Government officials and bodies utilize Twitter platform to mitigate latest news on COVID-19. The number of COVID-19 daily cases is the most anticipated news by Malaysian which usually tweeted by MOH official Twitter account such as below:



KKMalaysia 🇲🇾
@KKMPutrajaya

#COVID19 update for June 18. Malaysia recorded 6,440 new positive cases with 74 deaths.

#MenangBersama
#StayHome

MALAYSIA (June 18, 2021)
COVID-19 UPDATES

NEW POSITIVE CASES
6,440

Confirmed Cases	685,204
Recovered	615,326 (89.80%)
Deaths	4,276 (0.62%)

www.moh.gov.my
comms@moh

6:28 PM · Jun 18, 2021 · Twitter Web App

260 Retweets 78 Quote Tweets 317 Likes

Figure 1.1*: KKM Daily COVID-19 Cases Tweet*

The number of COVID-19 daily cases affects financial market, economic activity, household income and government policies (Buckman et al., 2020). Other than government officials, local news also play an important role to distribute the news through their official Twitter account. As observed, Malaysian tends to express their opinions and views on government COVID-19 policies. Government's countermeasure policies are crucial to minimize the impact of the outbreak to the citizen's wellbeing. Thus, making it critical to analyse the sentiments to quickly able to see the wider public opinion on a particular COVID-19 issues especially regarding Movement Control Order (MCO), national vaccination program and the number of daily cases.

The hashtag "#stayhome" has been widely used in various social media including Twitter. The government took an action to provide personal protection equipment for frontliners and also announced PRIHATIN package for Malaysian at the early stage of the pandemic (Shah et al., 2020). Following the government action, non-NGOs started to follow to provide any form of assistance to the people in need. Different sentiments come with the hashtag which brought a big impact in making sure social media users take care of themselves and stay at home.

However, visualizing the most accurate sentiment on the topic is the significant gap on identifying the public opinions. Sentiments cannot be concluded by analysing individual points of view. Every opinion must be carefully analysed and group together with other opinions to accurately perform sentiment analytics. Sentiment analysis can be broken up to two relevant classification which are polarity and subjectivity. Polarity is to determine the orientation of the sentiment whether it is positive, negative, or neutral. Meanwhile, subjectivity is the emotions, attitudes, and feelings of the emotion such as sad, angry, and happy. Therefore, these two elements are used to classify the sentiment analytics result. Dashboard is used to visualize the data provided by the analysis to show a certain key performance index (KPI) in the sentiment.

On the other hand, sentiment analysis is the process of determining a statement's or sentence's sentiment which uses a classification technique to derive opinions and compute a sentiment based on a data. Sentiment is a subjective component towards a topic of interest where machine learning can be applied to formulate features that able to decide for the sentiment it expressed. While in programming model, sentiment analysis uses different models for different types of topics in order to get the most accurate sentiment out of the entities in the topic. For instance, Naïve Bayes sentiment analysis model is the best for large data sets, and it is known to perform better than even the highly sophisticated classification methods (Gupta et al., 2017).

Twitter is home of where datasets can be found easily because it houses people's opinions about a topic of interest. The main actions that can be done in Twitter are tweeting, retweeting which involves tweeting other people's tweet into own profile, replying to tweet, and like a tweet. The content of a tweet affects the retweet and like numbers of the tweet where in some point, popular tweets are displayed in a Topic. Although hashtag plays a huge role in the discovery of a new topic of interest, Twitter Topic is also a famous feature that provides a platform for Twitter user to explore a new topic of interest. Therefore, Twitter is ideal in gathering opinions from different users to generate a collective sentiment value towards a topic.

According to a Twitter statistic, there are 310 million monthly active users on the platform around the world (Ante, 2021). Every second, about 6000 tweets are sent out. This volume of tweets has resulted in consumers consistently sending 4 about 350, 000 tweets per minute through the Internet. As a result, every day, about 500 million tweets are sent out by all users around the world. In just one year, 1.3 billion accounts were established on Twitter. Users tweet about a variety of topics, remarks, opinions, and thoughts, whether positive or negative.

1.2 Problem Statement

The existing mechanisms to analyse people's opinions on COVID-19 are not reliable to allow the government bodies or the publics to gain an accurate overview of the wider public opinion on COVID-19 issues such as National Vaccination Programme, Movement Control Order (MCO), and number of daily cases in Malaysia. The current rising trend of COVID-19 cases creates sub-problems to this issue.

In the early to middle stage of COVID-19 pandemic in Malaysia, it has been revealed that the proportion of households experiencing moderate and severe anxiety was highest among those with an income of less than MYR2000 (Wong & Alias, 2021). However, the statistics do not represent the underlying impact of COVID-19 onto the poverty cases in Malaysia. The government is responsible in reducing the psycho-behavioural and economic impact of the pandemic towards the citizens of Malaysia.

Furthermore, fake news can potentially create a major problem in altering people’s sentiment. The rising cases of fake news distribution through social media platform will affect the overall views on a topic which leads to issues such as harmful home remedies, and anti-vaccine activists. With WhatsApp being the common platform to forward messages among friends and families, a lot of home remedies have been circulated that can be harmful to a person through the platform. For instance, a video of a man drinking bleach which some interpreted that drinking the bleach might help with COVID-19. Other than that, only 45% of Malaysia population registered for national vaccination programme and around 12,000 did not turn up for their vaccination appointment in Kelantan (News Strait Times, 2021). These sentiment issues are the problem in combating the pandemic. Therefore, the government must take people’s views or opinions to plan future initiatives in combating COVID-19.

1.2.1 Objectives

To address the problem statements discussed earlier in 1.2, the objectives of this project are as follows:

1. To develop an algorithm that analyses people's opinions on Twitter by using Naïve-Bayes approach.
2. To conclude the algorithm's results in analysing people's opinion through the sentiment score.
3. To develop a dashboard that visualizes the sentiment analysis result into graph and numerical score.

1.2.2 Project Scope and Limitation

This project aims to perform a sentiment analysis on Malaysian tweets and visualize its results. The targeted issues that will be analysed are National Vaccination Programme, Movement Control Order (MCO) and the number of daily cases.

The sentiment analysis covers two elements which are polarity and subjectivity index. Real-time Twitter API streaming is used to stream the latest tweet to get the most accurate result on the topic. However, only tweets that have *place_id* will be streamed to get only tweets from Malaysia. In this case, there might be a scenario where non-geotagged tweet from Malaysia are not included.

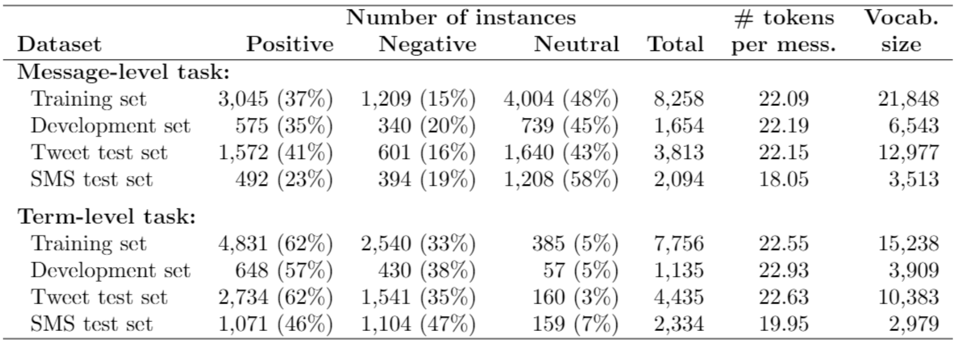
Furthermore, the polarity index is to determine how positive, negative, or neutral are the tweets about. Meanwhile, the subjectivity index is to determine the emotion of the tweets which includes happy, sad, angry and confused. The index will be computed using Naïve-Bayes algorithm. The index results will be visualized in pie chart to make it more understandable. A simple sentence to conclude the result.

CHAPTER 2
LITERATURE REVIEW

2.1 Sentiment Analysis

Sentiment Analysis (SA) is also known as opinion mining, and it is a computer research that uses Natural Language Processing (NLP), text analysis, and computational linguistics to gather general attitudes about a topic based on views. The general approach of sentiment analysis/opinion mining is to extract data, pre-process data, create a dictionary, break the data into chunks, and analyse the data (Bakshi et al., 2016). Natural Language Processing (NLP) is the common technique in performing sentiment analysis which involves in tokenization and processing at machine level (Li, 2020). Sentiment is closely related to attitude of the message towards a specific topic which can be classified by polarity and subjectivity index. In this project, Twitter is source of information. Twitter is a microblogging platform that contains people's public opinions on a topic discussed. Compared to other social media platforms, Twitter discussed an issue more transparent and honest which will enhance the result of the analysis (Alayba, 2020).

Table 1 shows the two-level of sentiment analysis that comprised of training set, development set and two test set (Kiritchenko et al., 2014).



	Number of instances				# tokens	Vocab.
Dataset	Positive	Negative	Neutral	Total	per mess.	size
Message-level task:						
Training set	3,045 (37%)	1,209 (15%)	4,004 (48%)	8,258	22.09	21,848
Development set	575 (35%)	340 (20%)	739 (45%)	1,654	22.19	6,543
Tweet test set	1,572 (41%)	601 (16%)	1,640 (43%)	3,813	22.15	12,977
SMS test set	492 (23%)	394 (19%)	1,208 (58%)	2,094	18.05	3,513
Term-level task:						
Training set	4,831 (62%)	2,540 (33%)	385 (5%)	7,756	22.55	15,238
Development set	648 (57%)	430 (38%)	57 (5%)	1,135	22.93	3,909
Tweet test set	2,734 (62%)	1,541 (35%)	160 (3%)	4,435	22.63	10,383
SMS test set	1,071 (46%)	1,104 (47%)	159 (7%)	2,334	19.95	2,979

Table 2.1*: Two-level of sentiment analysis*

The tweets include both standard English-language vocabulary and Twitter-specific terminology like emoticons, URLs, and inventive spellings. The message-level task portrays the process of detecting the polarity of the message whether it is positive, negative, or neutral. Meanwhile, the term-level task includes the detection of words in different context such as “You are so unpredictable” and “The drama has an unpredictable ending”. The usage of unpredictable can be positive, negative, or neutral. Therefore, other words from the message should be compared in conjunction of the word unpredictable to get a more accurate result.

Twitter is the perfect social media platform to get opinions from Malaysian especially on COVID-19 current issues. Twitter allows retweet, quote retweet and reply feature to express opinions onto another user’s tweet. As the nature of discussion, there might be disagreement among the users which leads to continuous discussion. Furthermore, the tweets can be downloaded through Twitter API which contains a lot of noisy data (Lee et al., 2012). Various research on Twitter data has been done for the past recent years especially for opinion mining research.

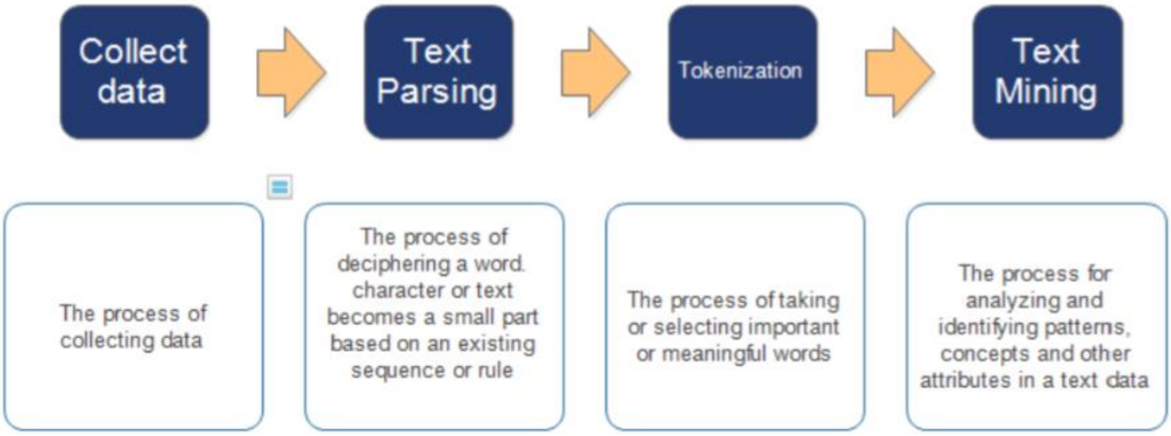

Collect data
Text Parsing
Tokenization
Text Mining
The process of collecting data
The process of deciphering a word. character or text becomes a small part based on an existing sequence or rule
The process of taking or selecting important or meaningful words
The process for analyzing and identifying patterns, concepts and other attributes in a text data

Figure 2.1: Common steps in Naïve-Bayes technique

The most common steps in applying Naïve-Bayes technique which are to collect data and parse the data to get the sentence into sequence. Next, the tokenization process takes place to clean the tweet and extract the features of the message. The last step is to do text mining using Naïve-Bayes method (Wongkar & Angdresey, 2019). Naïve-Bayes method is commonly used in opinion mining, however there is still debate among researchers of its accuracy that sometimes do not reflect to the original message of a sentence. Therefore, some researchers tend to use other methods of opinion mining such as Support Vector Machine (SVM) and K Nearest Neighbour (KNN).

Table 2 shows the comparison between Naïve-Bayes, Decision Tree and KNN in Machine Learning algorithms based on a journal by International Journal of Advanced Computer Science and Applications (Ashari et al., 2013).

***Table 2.2**: Comparison between Naive-Bayes, Decision Tree, and KNN*

Naïve-Bayes	Decision Tree	K-Nearest Neighbour (KNN)
Generative model	Discriminative model	Discriminative model
Parametric	Non-parametric	Non-parametric
Moderate performance	Fastest performance	Slowest performance

Based on the study by Ashari et al. (2013), Naïve-Bayes outperforms decision tree and K-nearest neighbour. Both decision tree and KNN are discriminative model which focuses on finding the decision boundary to separate one class to another. Meanwhile, Naïve-Bayes model is generative which focuses on the distribution of individual classes in a dataset which allow the algorithm to learn the data pattern. Therefore, Naïve-Bayes algorithm is the most suitable in analysing COVID-19 related tweets to enable the machine to learn certain keywords that can have multiple meanings unlike decision tree and KNN which draw the boundary that unable to detect subjective keywords in a tweet.

2.2 COVID-19

COVID-19 is an ongoing outbreak of respiratory disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) contributes more than 192 million cases in the world (Fauci et al., 2020). The pandemic begun to spread rapidly in the year 2020 which the impacts are beyond mortality. Up until the year 2021, country leaders are still executing their COVID-19 recovering plan to curb further outbreak of the virus.

Economy suffers the most in this pandemic due to movement restriction worldwide. People reduced their social consumption (physical interaction) as an effort to avoid COVID-19 infection which largely impacts on GDPs (Wren-Lewis, 2020). This shows that human capital is the most valuable asset that impacted by the pandemic situation. Wren-Lewis (2020) also mentioned the one who mostly unemployed during the pandemic period are the one who was employed especially in hospitality and travel agency. Financial problems on individuals arise at the early to mid-stage of the pandemic. Therefore, Twitter is an important data source to fetch people's emotions and discussions on the current issue (Xue et al., 2020).

Other than that, misinformation is another issue that should be analysed to prevent further the spread of false information among the community especially on Twitter. An exploratory analysis conducted shows that the verified Twitter accounts (organization/celebrity) are also involved in creating or spreading misinformation of COVID-19 (Shahi et al., 2021). Misinformation can create confusion among the Twitter community, and any responsible bodies might not be aware of the false information. Even worse, harmful home remedies claimed to cure COVID-19 are also being spread like wildfire. Therefore, misinformation can be recognized through sentiment analysis by analysing people's views on a specific topic.

2.3 Previous Related Work on COVID-19 Sentiment Analysis

A COVID-19 Twitter sentiment analysis on Florida have been conducted from 22 April 2020 until 28 April 2020 on a total of 26,541 tweets by applying TextBlob library and NRC Lexicon techniques (Li, 2020). The study utilized the two mentioned sentiment analysis method to give different views of sentiment from citizens, governors, and organizations in Florida. The process of sentiment analysis is searching for tweets by using hashtags '#covid-19', '#coronavirus', and '#covid', set the geographical location of the tweets, pre-process tweets to remove useless contents, and perform the sentiment analysis. Although the study did not conclude which method is better and accurate, the study found the percentage of tweets regarding COVID-19 issues on Twitter which are averagely 53.68% on TextBlob and 46.43% on NRC.

Furthermore, a study in applying sentiment analysis on Twitter data worldwide have been conducted by Kurdistan Journal of Applied Research. The tweet data were collected during one of the most spread week of COVID-19 which is from 9 April 2020 until 15 April 2020 (H. Manguri et al., 2020). The collected tweet is a total amount of 530,232 tweets worldwide, only TextBlob Python library of Sentiment Analysis technique is utilized in this study. The two key hashtags in this study are '#coronavirus' and '#COVID-19'. The results shown in this study is neutral polarity for both the key hashtags which is more than 50 percent, meanwhile a large portion of the records were objective approximately around 64 percent.

CHAPTER 3
METHODOLOGY

This project will be developed in Python programming language because it offers wide choice of machine learning libraries to perform sentiment analysis. Meanwhile, the dashboard will be developed by utilizing HTML, CSS and JavaScript to visualize the sentiment analysis results.

Agile methodology is the most suitable project management for this project. The methodology will involve in gathering the requirements using appropriate elicitation techniques. Tasks are broken up into several phases to further improve the current focus feature. As this nature of project, improvement over time is required to make sure the analysis produce an accurate result based on the current trend.

3.1 Agile Phases

This project has two parts which are sentiment analysis and dashboard that is combined in the agile methodology to ease requirement gathering and deployment. Agile methodology is broken up into several phases which can be repetitive throughout the project development as depicted in the figure.

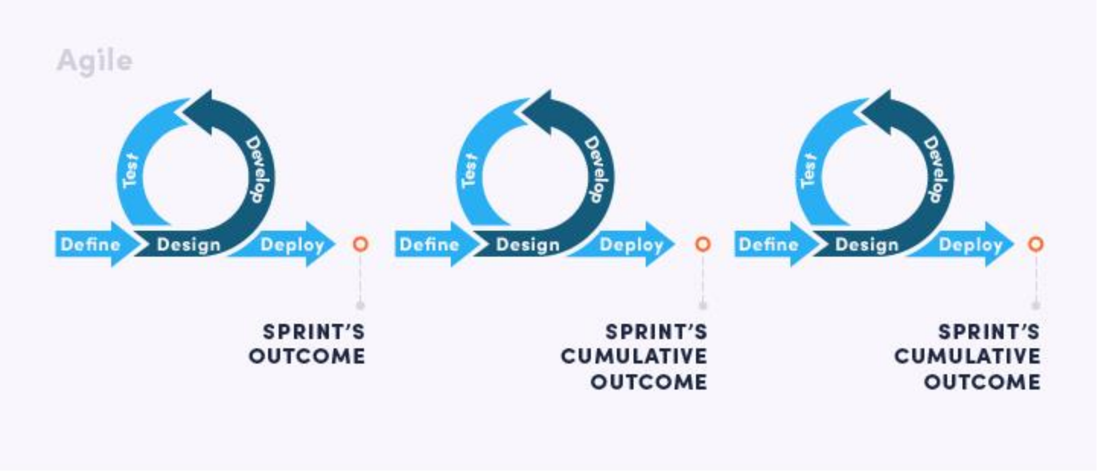


***Figure 3.1**: Agile Phases*

1) PLAN

- Identify the required data to perform sentiment analysis including listing down the characteristics of data.
- Determine which algorithm and method to be used from extracting, preparing, and executing the categorization of data in sentiment analysis process.

2) DESIGN

- Design data flow diagram for dashboard and the backend for the system.
- Design the algorithm of performing sentiment analysis.
- Design the dashboard to represent the data visualization computed from the backend.

3) DEVELOP

- Develop the algorithm of performing sentiment analysis in the backend.
- Develop the dashboard of the sentiment analysis using Anvil web app platform.

4) TEST

- Test the dashboard if the dashboard reflects the correct data generated from the backend.
- Test the algorithm with sets of data.

5) RELEASE

- Deploy the feature planned to the dashboard.
- Keep track the rate of success of the release.

6) FEEDBACK

- Possible improvement on the algorithm to improve the data visualization of the sentiment analysis results.

3.2 Use Case Diagram

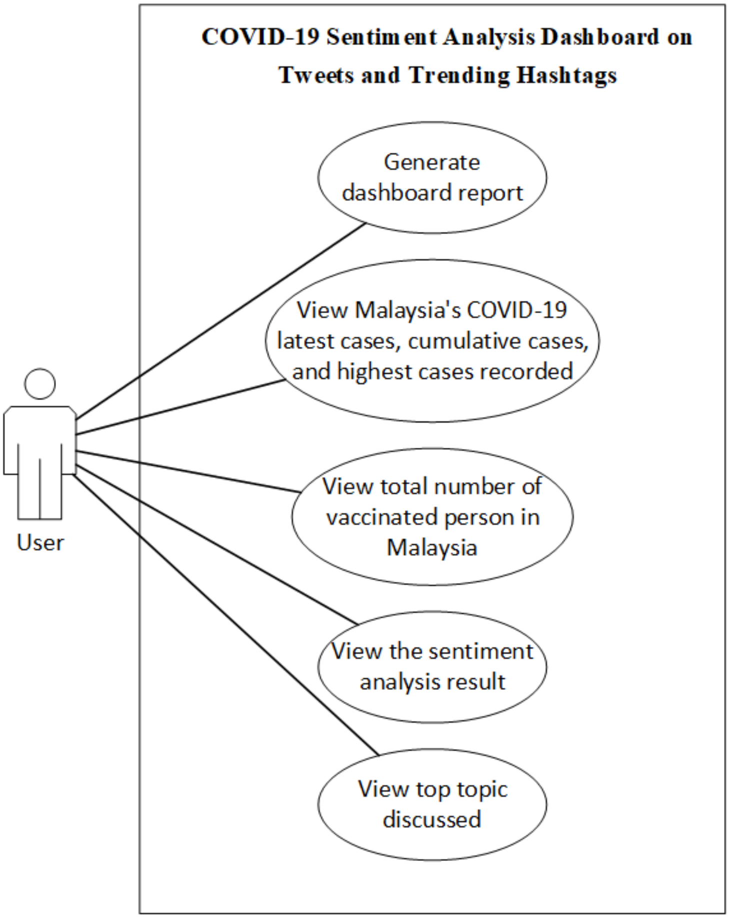

COVID-19 Sentiment Analysis Dashboard on Tweets and Trending Hashtags
Generate dashboard report
View Malaysia's COVID-19 latest cases, cumulative cases, and highest cases recorded
View total number of vaccinated person in Malaysia
View the sentiment analysis result
View top topic discussed
User

***Figure 3.2**: Use Case Diagram*

Figure 3.2 shows the use case diagram for this project. The main actor for this system is a user that able to generate report through a button. The system accepts the click event from the user and call the backend API to return requested information from the front-end. Next, the user is able to view the latest Malaysia's COVID-19 cases, cumulative cases, and highest cases fetched from KKM's COVID-19 open data in Github. Other than that, the total number of vaccinated people in Malaysia is also displayed that are fetched from KKM's COVID-19 vaccination open data. The dashboard showcases top topic discussed to the user through a bar chart and word cloud. Lastly, the user can view the technical-based sentiment analysis result generated through plot graph and bar chart.

3.3 Activity Diagram for Backend

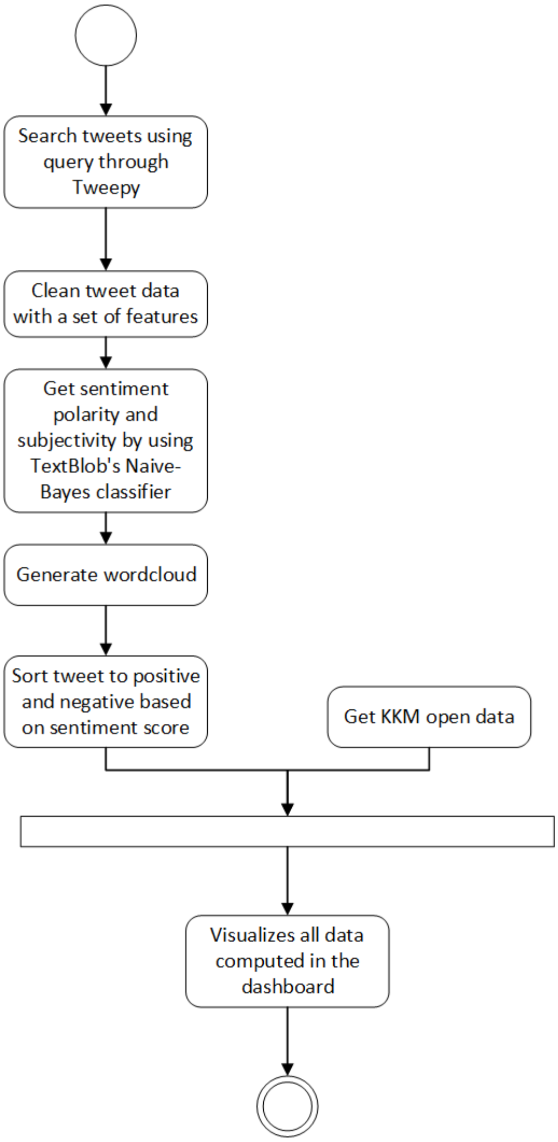

Search tweets using query through Tweepy
Clean tweet data with a set of features
Get sentiment polarity and subjectivity by using TextBlob's Naive-Bayes classifier
Generate wordcloud
Sort tweet to positive and negative based on sentiment score
Get KKM open data
Visualizes all data computed in the dashboard

***Figure 3.3**: Activity Diagram for Backend*

Figure 3.3 shows the activity diagram of this project. The activity diagram shows the activity processes when an event to generate report is triggered. The backend API starts with searching related tweet data through a set of queries specified in the backend. The tweet data is parsed into a data frame for better organization. The tweet data is cleaned through a set of features with regular expressions for a clean sentiment score result. Based on the data frame, the system generates a wordcloud to showcase the top topic discussed. The tweet data is then sorted into positive, negative, or neutral score. Along with getting to sort the tweet data, the system fetches relevant data from KKM open data Github and visualizes all the data into the dashboard.

3.4 Activity Diagram for Dashboard (Web App)

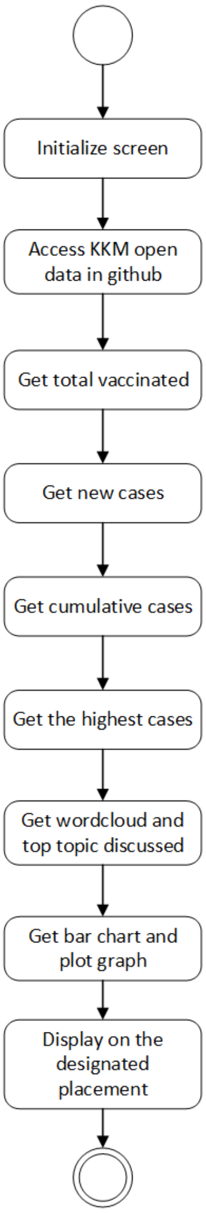

Initialize screen
Access KKM open data in github
Get total vaccinated
Get new cases
Get cumulative cases
Get the highest cases
Get wordcloud and top topic discussed
Get bar chart and plot graph
Display on the designated placement

***Figure 3.4**: Activity Diagram for Dashboard*

Figure 3.4 shows the activity diagram for which the sequence of fetching information from backend. Information are being passed by functions; therefore, the dashboard must sort the sequence efficiently to fetch information in the shortest amount of time. As displayed in the activity diagram above, the system fetches numbers from KKM open data first because it does not include any computation or scientific execution in order to produce the numbers. Next, the top discussed topic is fetched along with the bar chart and plot graph for the results of sentiment analysis.

3.5 Data flow diagram

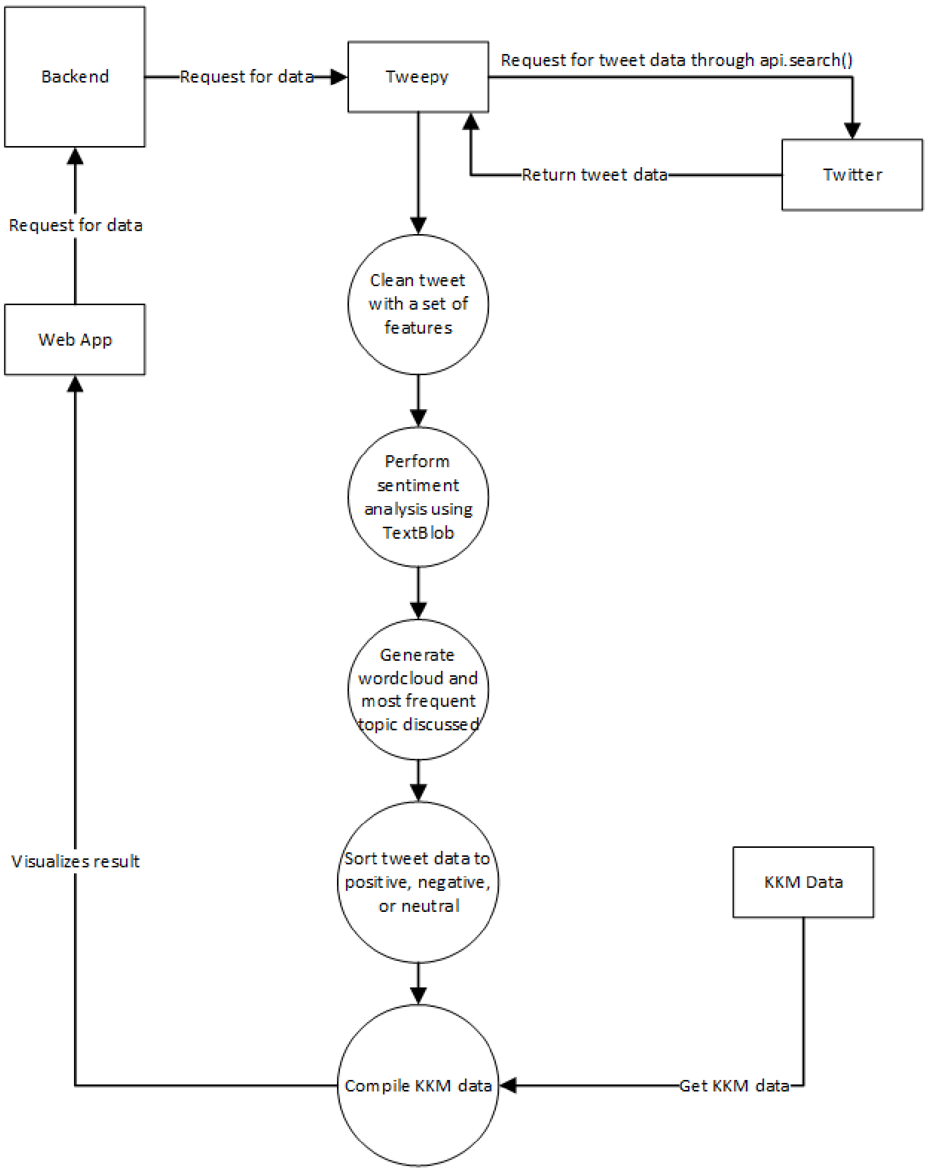

Backend
Request for data
Tweepy
Request for tweet data through api.search()
Twitter
Return tweet data
Request for data
Web App
Clean tweet with a set of features
Perform sentiment analysis using TextBlob
Generate wordcloud and most frequent topic discussed
Visualizes result
Sort tweet data to positive, negative, or neutral
KKM Data
Compile KKM data
Get KKM data

Figure 3.5*: Data flow Diagram*

Figure 3.5 shows the flow of data during the process of sentiment analysis. The process starts on the web app where it requests the backend API for Tweepy to fetch latest tweet from Twitter. Tweepy accesses Twitter API to get the related tweets and the data will passed back to Tweepy. Cleaning phase of the tweet data is processed and performing the sentiment analysis towards the cleaned tweet data using TextBlob is executed. The computed tweet data are sorted into positive, negative, or neutral sentiment. Next, the process of compiling KKM data begins and the information is parsed from backend to the web app referring to Section 3.4.

3.6 Twitter Sentiment Analysis

3.6.1 Twitter for Developers

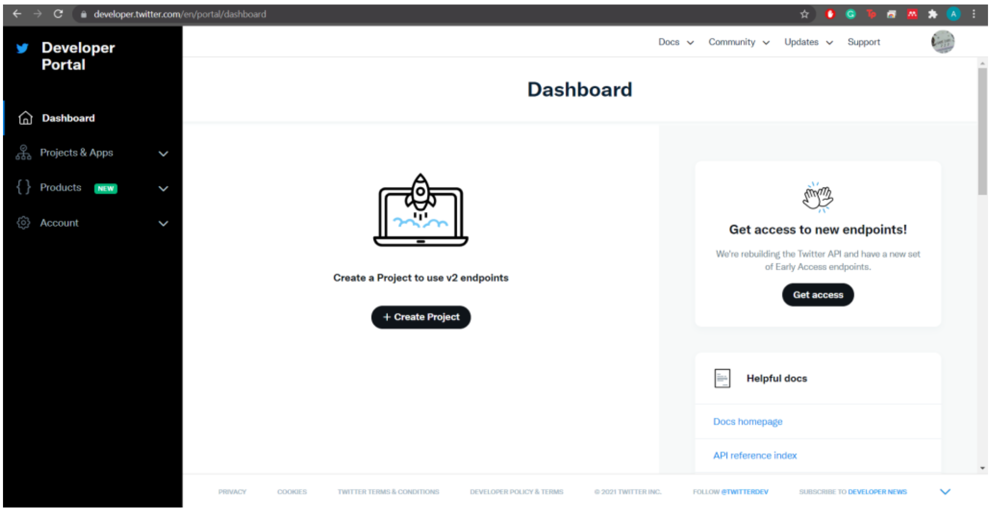
A Twitter Developer account is needed to access Twitter API to perform various types of actions on Twitter. API is an *application programming interface* which is a ‘middleman service’ acting as an intermediary between the developer’s code and Twitter’s database. Every data in Twitter’s database a developer requested, will go through the API before being passed to the developer’s code. However, to obtain a Twitter Developer account, developers are required submit a request by specifying several information to Twitter.

3.6.1.1 Apply for Twitter Developer account

Twitter data is protected under Twitter terms and regulations. Twitter does not approve their data to be sold to any third-party for businesses or profits (Campan et al., 2018). Therefore, Twitter reviews every request for Twitter Developer account carefully before approving it. The process to apply Twitter Developer account is as follows:

1. Go to https://developer.twitter.com
2. Login to the profile to associate the account
3. Categorise the associated account
4. Choose any use case related to the usage of the account
5. Describe what will be built with the account
6. Determine if the product or service is for government use
7. Agree to Twitter’s terms and conditions
8. An email of request will be sent to the user account
9. Wait for 1-2 days approval of the account

After approval of the request, the account can access the developer portal as shown in Figure 8 below.



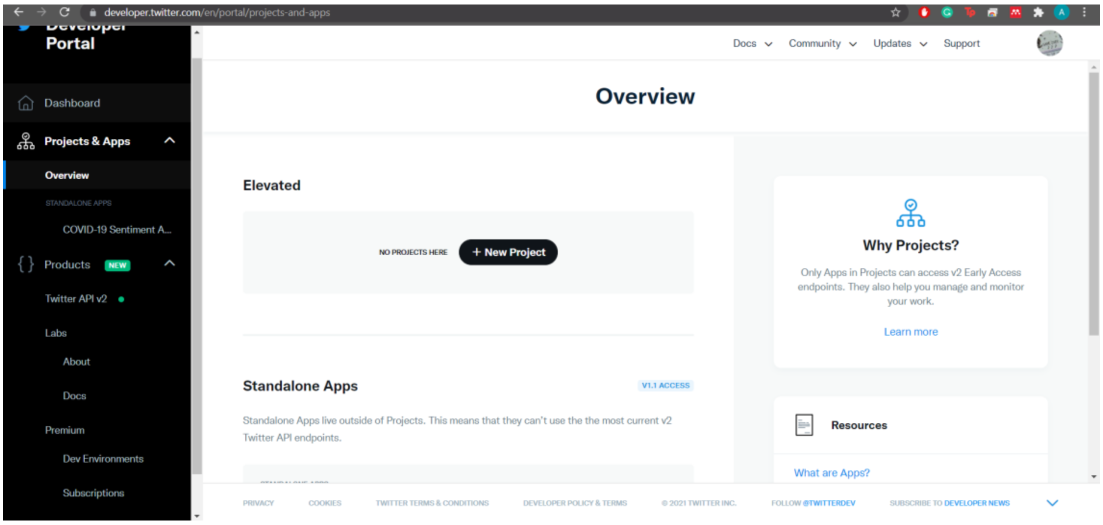
We're rebuilding the Twitter API and have a new set of Early Access endpoints.

Figure 3.6: Twitter Developer Dashboard

3.6.1.2 Create a standalone app in Twitter Developer Dashboard

The next step is to create a standalone app in Twitter Developer Dashboard where developers can obtain their consumer token, consumer secret, access token, and lastly access token secret. The steps are as below:

1. Create a new project



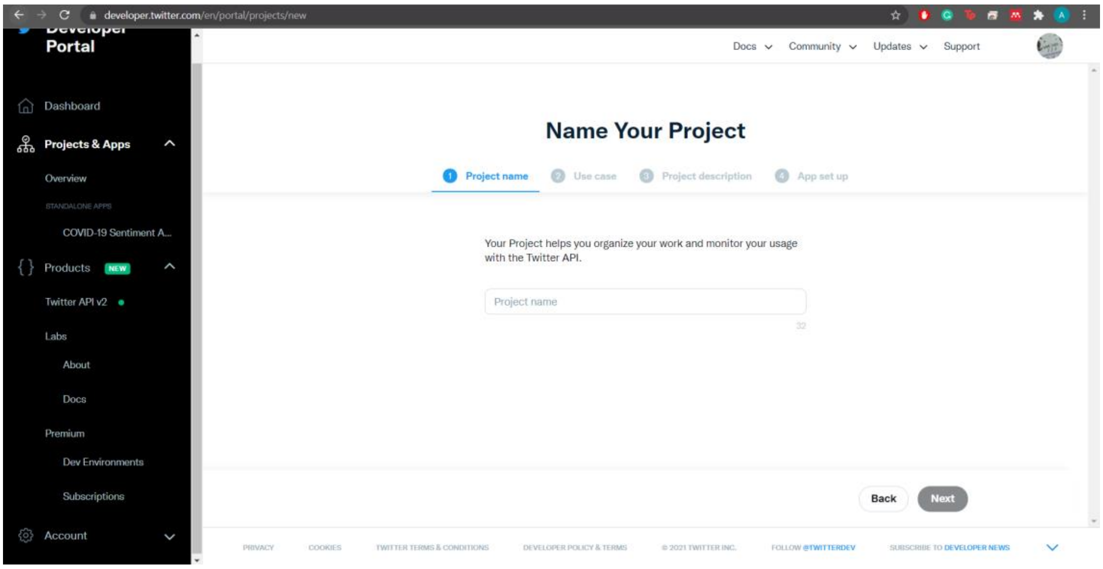
Only Apps in Projects can access v2 Early Access endpoints. They also help you manage and monitor your work.

Standalone Apps live outside of Projects. This means that they can't use the the most current v2 Twitter API endpoints.

Figure 3.7: Create New Project

Developers can go to Projects & Apps section and click the 'New Project' button. This will bring the developer to page in Figure 3.7.

2. Enter project name



Name Your Project

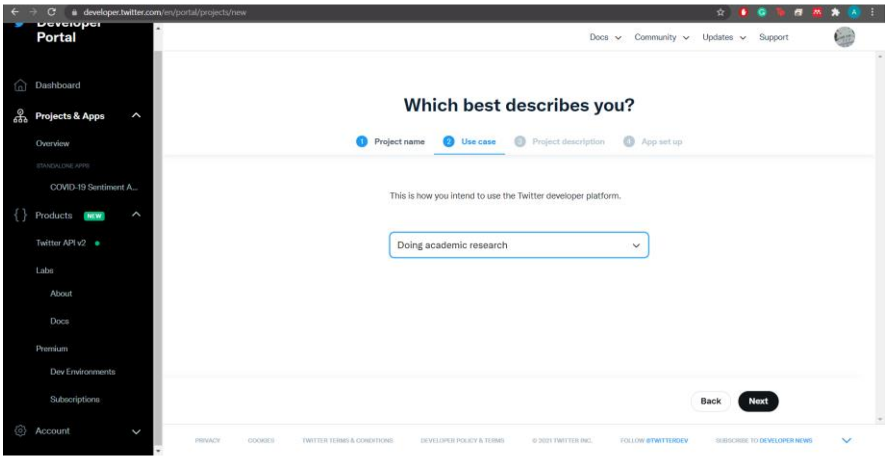
Your Project helps you organize your work and monitor your usage with the Twitter API.

Project name
32

Figure 3.8: Enter project name

The project name entered must be appropriate following Twitter's rules and guidelines. Any offensive or inappropriate name can be subjected to account suspension or removal.

3. Choose use case for the app



Which best describes you?

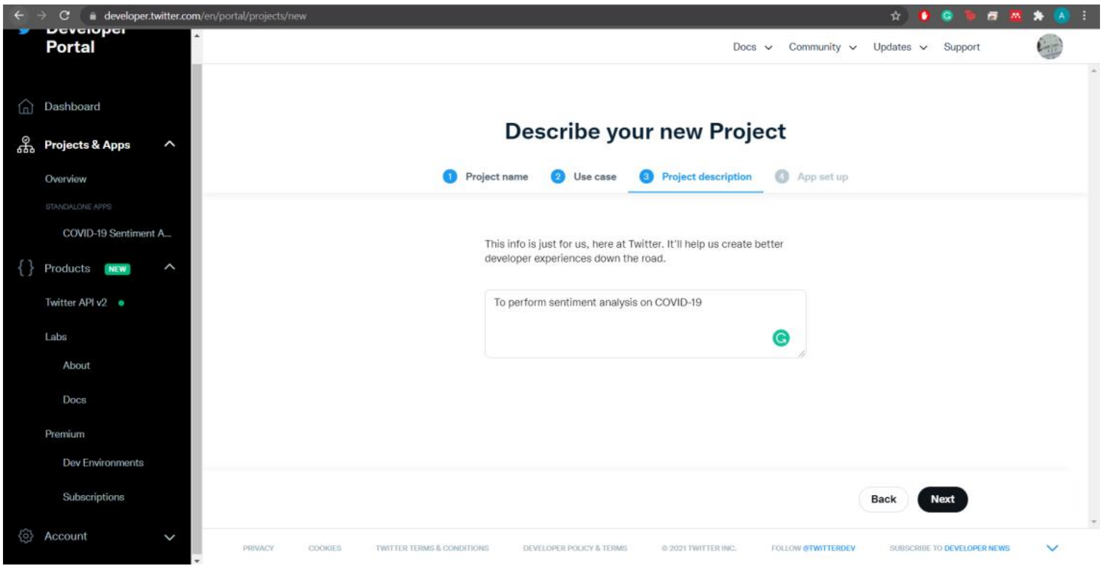
This is how you intend to use the Twitter developer platform.

Doing academic research

Figure 3.9: Pick a Use Case

There are 12 main use cases in the choice provided by Twitter. In this project, the use case will be 'doing academic research' on sentiment analysis. However, it does not affect the developers to access any API endpoint.

4. Describe the project description



Describe your new Project

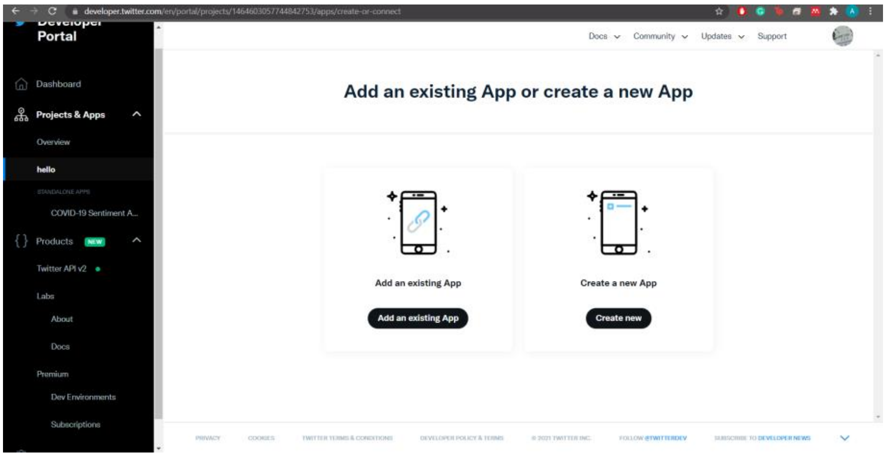
This info is just for us, here at Twitter. It'll help us create better developer experiences down the road.

To perform sentiment analysis on COVID-19

Figure 3.10: Describe the project description

Developers are required to fill in their project details on this section. This section in creating the app is to inform Twitter the uses of it so that it would not be misused for radical purposes. In this project, the app is used to perform sentiment analysis on COVID-19.

5. Choose app option

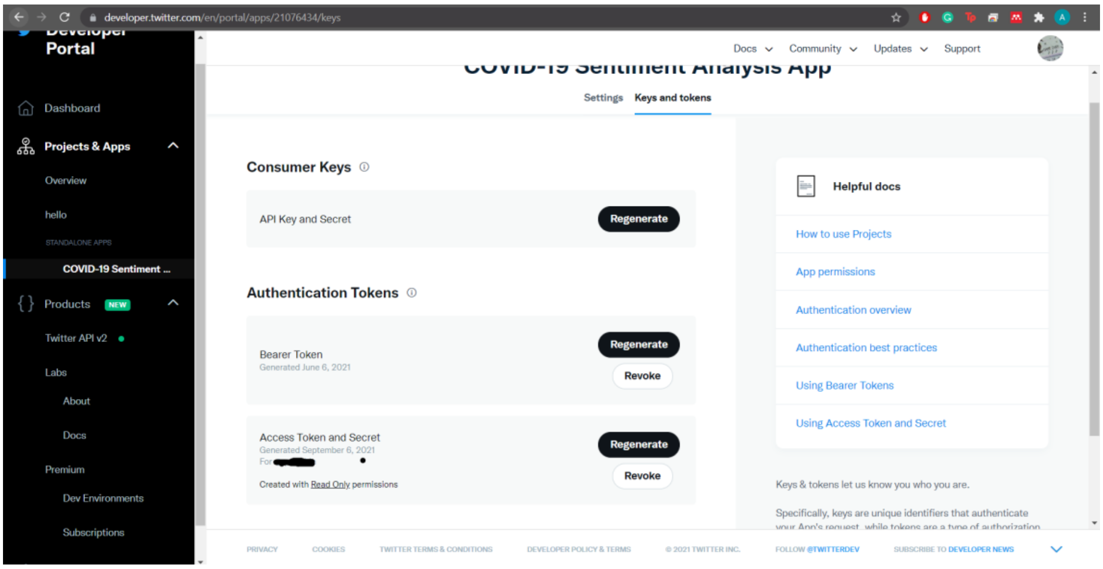


Add an existing App or create a new App

Figure 3.11: Choose an app option

Twitter gives options to add an existing app or create a new app. Adding an existing app will reset the consumer key, consumer secret, access token, and access token secret of the existing app. Meanwhile, creating a new app will give the developer a new set of the stated credentials.

6. Get credentials



COVID-19 Sentiment Analysis App

Consumer Keys

API Key and Secret

Authentication Tokens

Bearer Token
Generated June 6, 2021

Access Token and Secret
Generated September 6, 2021
For [redacted]

Created with Read Only permissions

Helpful docs

- How to use Projects
- App permissions
- Authentication overview
- Authentication best practices
- Using Bearer Tokens
- Using Access Token and Secret

Keys & tokens let us know you who you are.

Specifically, keys are unique identifiers that authenticate your App's request, while tokens are a type of authorization

Figure 3.12: Get credentials

Developers can go to their apps in standalone apps under Projects & Apps section. After clicking the app, all credentials will be under Keys and Tokens. The credentials will be used in the code in the backend.

3.6.2 Programming Language

After setting up Twitter Developer account and obtaining the credentials, the backend can finally connect to Twitter API through Tweepy library. In this project, Python is the programming language, while utilizing **Google Colab** as the platform to code. The justification to choose Python as the programming language is as follows (Brownlee, 2016):

a) Has an extensive selection of libraries and frameworks

- Python is well-known to be the best programming language for machine learning (ML) and artificial intelligence (AI). AI and ML use cases such as data analysis, data visualization, computer vision, and natural processing language have libraries like Numpy, TensorFlow, OpenCV, and NLTK respectively. Therefore, developers can develop a product faster without building an algorithm from scratch. In this project, various libraries are used to perform sentiment analysis.

b) Independent platform

- Python code is supported by various platform including Linux, Windows, and MacOS. An existing Python code can produce standalone programs where the code can be easily distributed and executed in every platform and operating systems without Python interpreter. The cross-environment friendly traits in Python makes data training faster and cheaper. In this project, the Python code is also used in the web app where data from backend is parsed into the web app where Python makes it easier to pass data from an entity to another.

c) Readability

- Python code is easy to read because the syntax is not quite heavy compared to other programming languages. In this project, there are a lot of functions that dependent to each other where a heavy syntax format will be a barrier to an efficient code writing process. Therefore, Python makes it easier for developers to understand the code structure through indentation.

CHAPTER 4
Results and Discussion

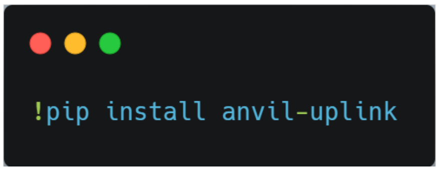
The objective of this chapter is to showcase the finished prototype of the system and the unit testing of important features in the system. Section 4.1 will discuss the algorithm in performing sentiment analysis, Section 4.2 will discuss on the finished prototype of the system and Section 4.3 will discuss on the unit testing result conducted.

4.1 Algorithm

4.1.1 Setting up the environment

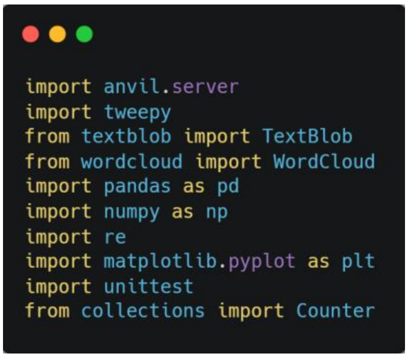
In this section, preparing the data at the backend is the initial process in performing sentiment analysis upon the fetched tweet data. The process of importing necessary libraries and preparing the tweet data will be discussed in this section. The process is explained as follows:

1) Installing anvil libraries for web app



```
!pip install anvil-uplink
```

2) Importing libraries



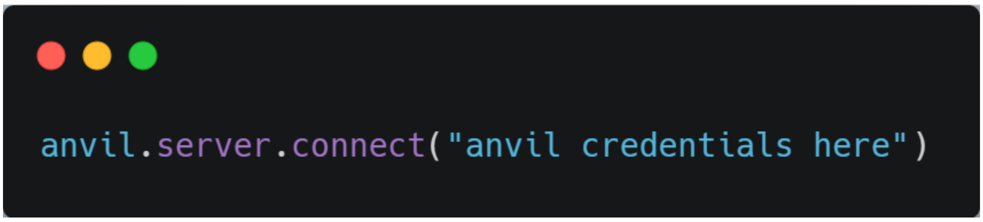
```
import anvil.server
import tweepy
from textblob import TextBlob
from wordcloud import WordCloud
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
import unittest
from collections import Counter
```

Usage of important libraries:

- ❖ ***import anvil.server*** – to use anvil server call function

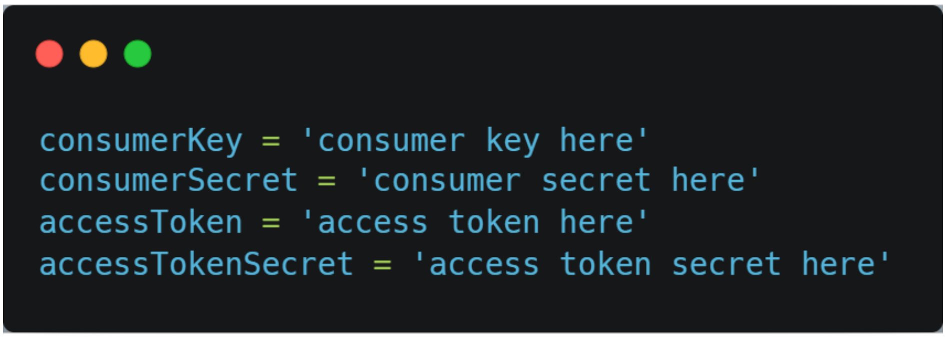
- ***import tweepy*** – provides a convenient access to Twitter API
- ***from textblob import Textblob*** – to get sentiment polarity and subjectivity
- ***import wordcloud import WordCloud*** – generates wordcloud and get the most frequent topic discussed
- ***import matplotlib.pyplot as plt*** – to use plotting functions
- ***import unittest*** – for the use of unit testing the code

3) Connect backend with Anvil Web App



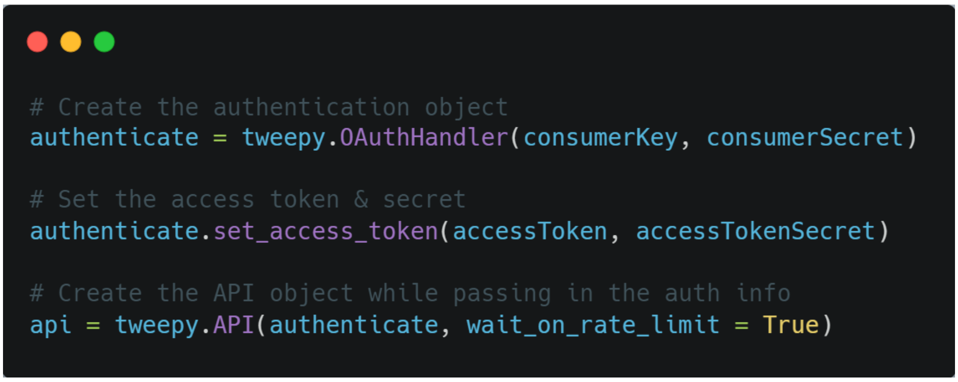
```
anvil.server.connect("anvil credentials here")
```

4) Initialize Twitter credentials



```
consumerKey = 'consumer key here'
consumerSecret = 'consumer secret here'
accessToken = 'access token here'
accessTokenSecret = 'access token secret here'
```

5) Verify backend connection to Twitter API



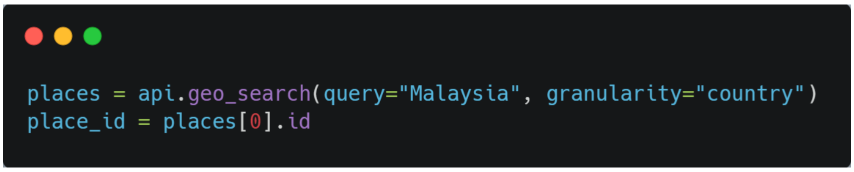
```
# Create the authentication object
authenticate = tweepy.OAuthHandler(consumerKey, consumerSecret)

# Set the access token & secret
authenticate.set_access_token(accessToken, accessTokenSecret)

# Create the API object while passing in the auth info
api = tweepy.API(authenticate, wait_on_rate_limit = True)
```

With these steps, the system is now ready to perform functions from Tweepy or Anvil. Further explanation of function implementation will be further explained in Section 4.1.2.

4.1.2 Preparing the tweet data

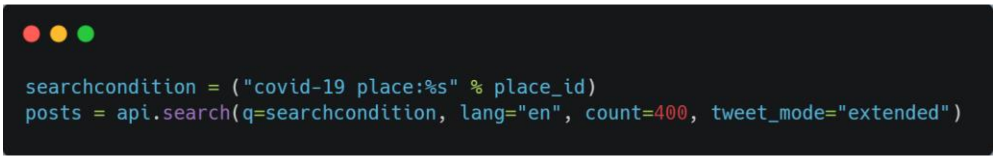
In this section, the method of preparing tweet data before performing sentiment analysis is discussed. A set of raw tweet data is fetched from Tweepy contains a lot of noise affecting the results of sentiment analysis. Therefore, the process of preparing tweet data is explained from fetching tweet data to the cleaning process of the tweet data.

1. Initialize place query to Malaysia



```
places = api.geo_search(query="Malaysia", granularity="country")
place_id = places[0].id
```

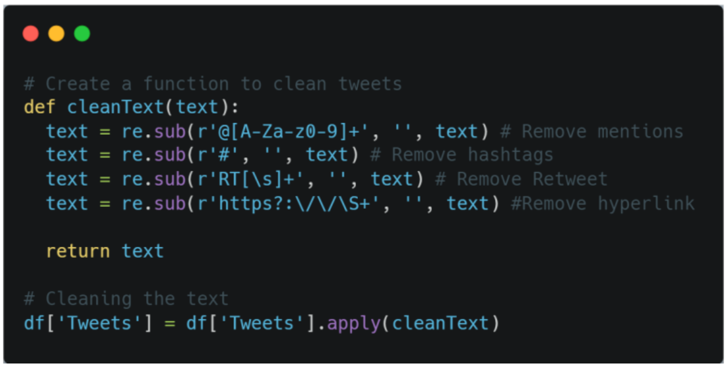
2. Perform search tweet



```
searchcondition = ("covid-19 place:%s" % place_id)
posts = api.search(q=searchcondition, lang="en", count=400, tweet_mode="extended")
```

The search condition is ‘covid-19’ within Malaysian tweet. There are about 400 tweets fetched from this function.

3. Clean the tweet



```
# Create a function to clean tweets
def cleanText(text):
  text = re.sub(r'@[A-Za-z0-9]+', '', text) # Remove mentions
  text = re.sub(r'#', '', text) # Remove hashtags
  text = re.sub(r'RT[\s]+', '', text) # Remove Retweet
  text = re.sub(r'https?:\/\/\S+', '', text) #Remove hyperlink

  return text

# Cleaning the text
df['Tweets'] = df['Tweets'].apply(cleanText)
```

The process of cleaning the tweet involved removing mentions, hashtags, retweet, and hyperlink from the tweet data. The cleaned tweet data is amended in dataframe with ‘Tweets’ column.

Tweet cleaning example

***Table 4.1**: Tweet Example*

ID	Place	Tweet
1	Johor, Malaysia	RT @khairyjamaluddin: 39 out of 46 positive... all asymptotic. Phew! https://t.co/3P6ciuS1LK
2	Sarawak, Malaysia	Lockdown series, Day 1: ensuring safety The video says it all, stay at home enforcement at its full... @michaelang #covid19
3	Malaysia	#nature had asked us to #slowdown #corona #QuaratineLife #QuarantineAndChill @temeabdullah https://t.co/JgJqS3ifjh

Step 1: Remove mentions

***Table 4.2**: Remove mentions*

ID	Tweet
1	RT : 39 out of 46 positive... all asymptotic. Phew! https://t.co/3P6ciuS1LK
2	Lockdown series, Day 1: ensuring safety The video says it all, stay at home enforcement at its full... #covid19
3	#nature had asked us to #slowdown #corona #QuaratineLife #QuarantineAndChill https://t.co/JgJqS3ifjh

Step 2: Remove hashtags

***Table 4.3**: Remove hashtags*

ID	Tweet
1	RT : 39 out of 46 positive... all asymptotic. Phew! https://t.co/3P6ciuS1LK
2	Lockdown series, Day 1: ensuring safety The video says it all, stay at home enforcement at its full...
3	had asked us to https://t.co/JgJqS3ifjh

Step 3: Remove retweet

Table 4.4*: Remove retweets*

ID	Tweet
1	: 39 out of 46 positive... all asymptotic. Phew! https://t.co/3P6ciuS1LK
2	Lockdown series Day 1 ensuring safety The video says it all, stay at home enforcement at its full
3	had asked us to https://t.co/JgJqS3ifjh

Step 3: Remove hyperlinks

Table 4.5*: Remove hyperlinks*

ID	Tweet
1	39 out of 46 positive all asymptotic Phew
2	Lockdown series Day 1 ensuring safety The video says it all, stay at home enforcement at its full
3	had asked us to

Step 4: Classify each word between white spaces

Table 4.6*: Words Classification*

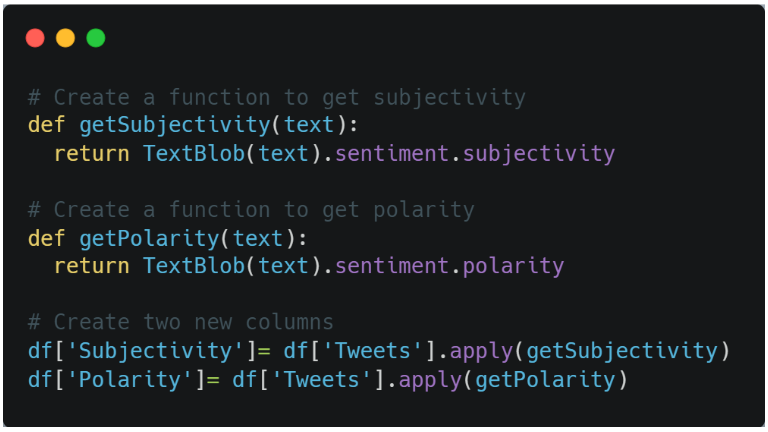
Words	Sentiment
39	Neutral
out	Neutral
46	Neutral
positive	Neutral
all	Neutral
asymptotic	Neutral
Phew	Positive

In this step the word will be stored in TextBlob’s dictionary for further use. Other than that, removing articles can be done manually such as the word ‘the’, ‘a’, and ‘or’.

Features are extracted and training data is also being utilized to get a more accurate result. As been mentioned earlier, Naïve-Bayes algorithm will be applied when doing the analysis. Results are calculated and visualized in the web app.

4.1.3 Preparing the data

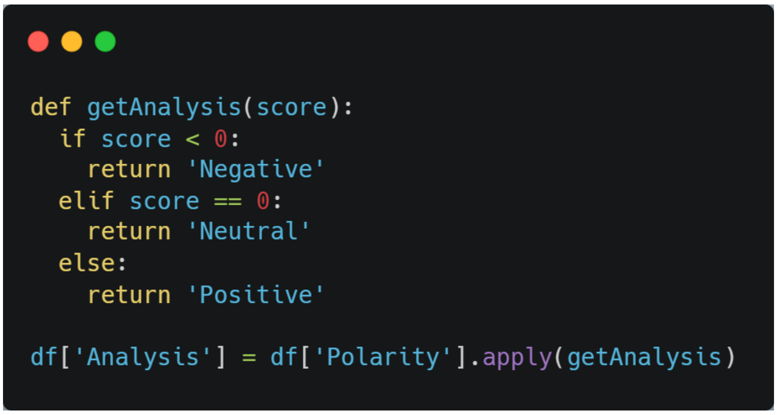
1. Perform sentiment analysis using TextBlob (subjectivity and polarity)



```
# Create a function to get subjectivity
def getSubjectivity(text):
  return TextBlob(text).sentiment.subjectivity

# Create a function to get polarity
def getPolarity(text):
  return TextBlob(text).sentiment.polarity

# Create two new columns
df['Subjectivity']= df['Tweets'].apply(getSubjectivity)
df['Polarity']= df['Tweets'].apply(getPolarity)
```

There two functions in the code snippet above which are *getSubjectivity* and *getPolarity*. There will be two new columns in the existing dataframe after both functions are executed.

2. Sort tweet data based on polarity (positive, negative, or neutral)



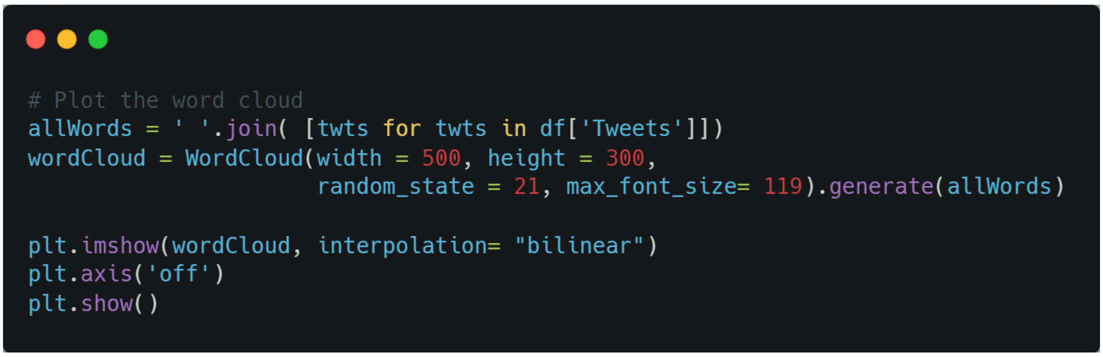
```
def getAnalysis(score):
  if score < 0:
    return 'Negative'
  elif score == 0:
    return 'Neutral'
  else:
    return 'Positive'

df['Analysis'] = df['Polarity'].apply(getAnalysis)
```

Polarity refers to the orientation of the tweet express, therefore it will be either positive, negative, or neutral. If the polarity score less than 0, it will return negative. If the polarity score equals to 0, it will return neutral. Therefore, if the polarity score more than 0, it will return positive.

4.1.4 Preparing the data

Wordcloud is the graphical representation of words weightage to visualize tweet data, in which gains attention because it shows greater distinction to words that appear more often in the source text (MURTHY & SCHOLAR, 2020). With wordcloud, understanding of topic discussed in tweet data is more significant. Topic such as vaccination, booster dose, and healthcare are more noticeable with the generate wordcloud.

1. Generate wordcloud



```
# Plot the word cloud
allWords = ' '.join( [twts for twts in df['Tweets']])
wordCloud = WordCloud(width = 500, height = 300,
                      random_state = 21, max_font_size= 119).generate(allWords)

plt.imshow(wordCloud, interpolation= "bilinear")
plt.axis('off')
plt.show()
```

The code snippet above shows in constructing the wordcloud. The variable *allWords* contains the joined words from dataframe in Tweets column. The width for the wordcloud is 500, height is 300, and the max font size is set to 119. The example of wordcloud generated is as follows:

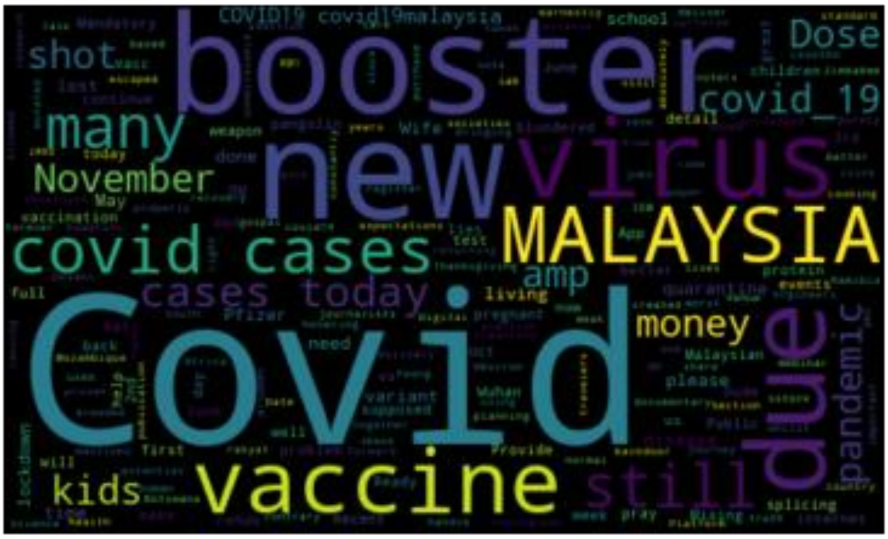
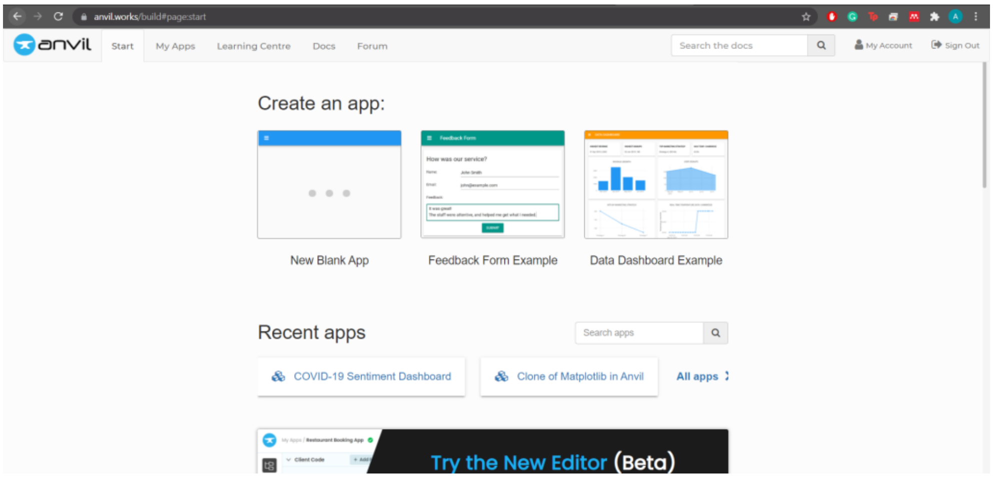


***Figure 3.1**: WordCloud*

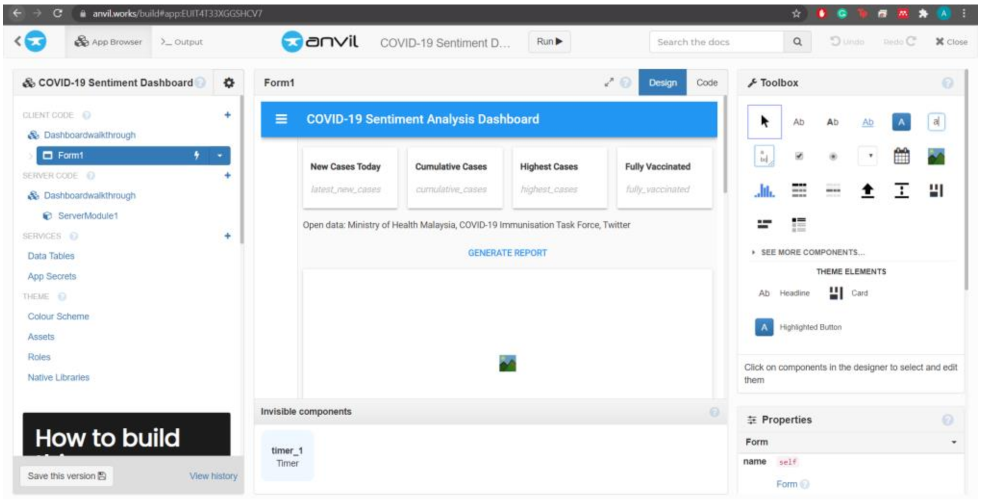
4.1.5 Anvil web app

Anvil is a platform for developers to create a full-stack web app that is written completely in Python. Anvil allows drag and drop for UI organization in their platform which saves developers a lot of time in coding the whole UI. After building the app, developers can deploy the app with one click. In this project, Anvil acts as a dashboard which is used to showcase the sentiment analysis result generated from the backend. Other than that, the numbers from KKM open data are also included in the dashboard. There are several steps must be followed to complete the dashboard.

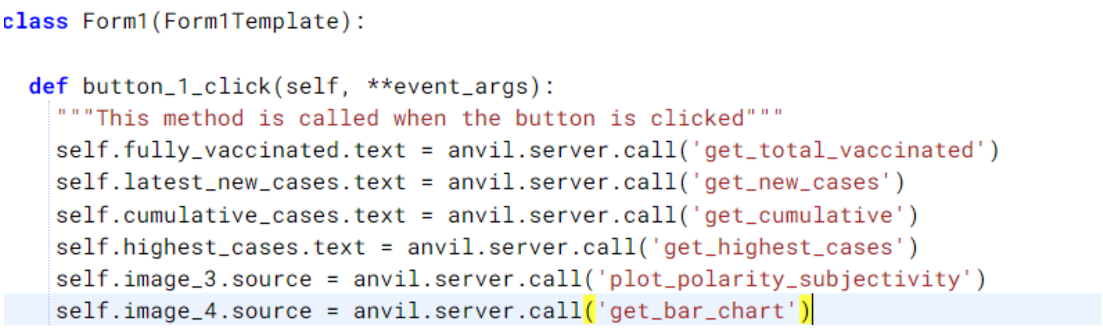
1. Create an Anvil app



2. Design the page



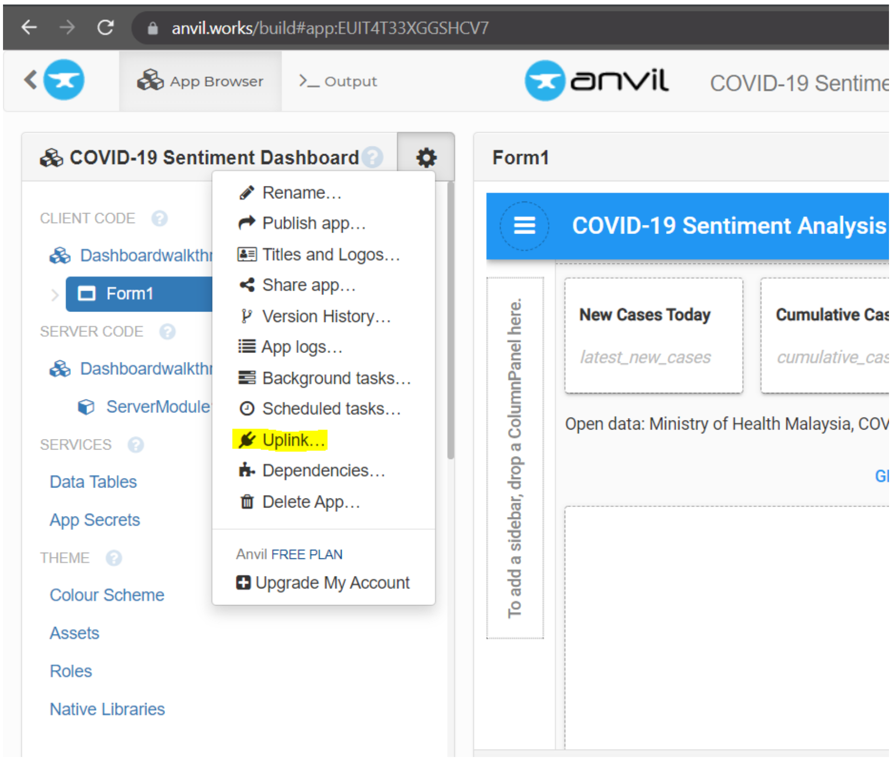
3. Add a button event to generate report



```
class Form1(Form1Template):

  def button_1_click(self, **event_args):
    """This method is called when the button is clicked"""
    self.fully_vaccinated.text = anvil.server.call('get_total_vaccinated')
    self.latest_new_cases.text = anvil.server.call('get_new_cases')
    self.cumulative_cases.text = anvil.server.call('get_cumulative')
    self.highest_cases.text = anvil.server.call('get_highest_cases')
    self.image_3.source = anvil.server.call('plot_polarity_subjectivity')
    self.image_4.source = anvil.server.call('get_bar_chart')
```

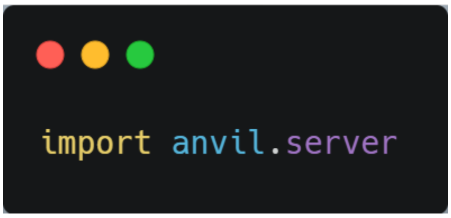
All the actions under *button_1_click* will be executed once the specified button is clicked.

4. Enable the uplink



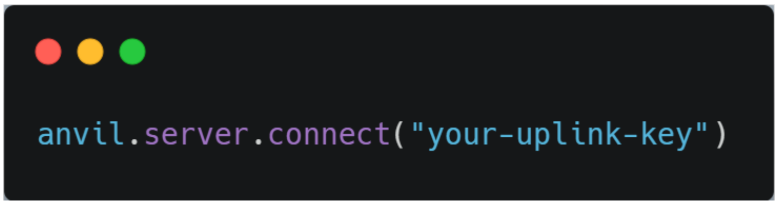
Enabling the uplink allows Google Colab to connect to Anvil through Anvil credentials. Uplink key is provided to be used for connecting other platforms to Anvil.

5. Connecting the script to Google Colab



```
import anvil.server
```

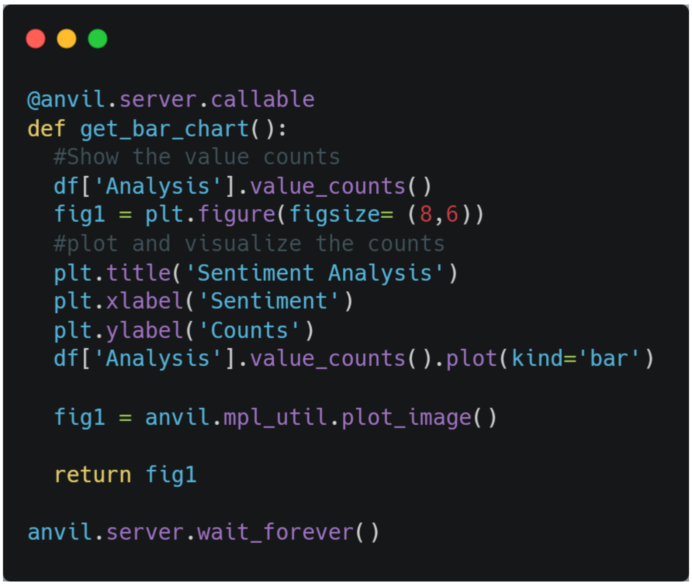
The connection can be started with importing *anvil.server* library.



```
anvil.server.connect("your-uplink-key")
```

Google Colab is connected to Anvil and ready to be executed the code snippet above is applied.

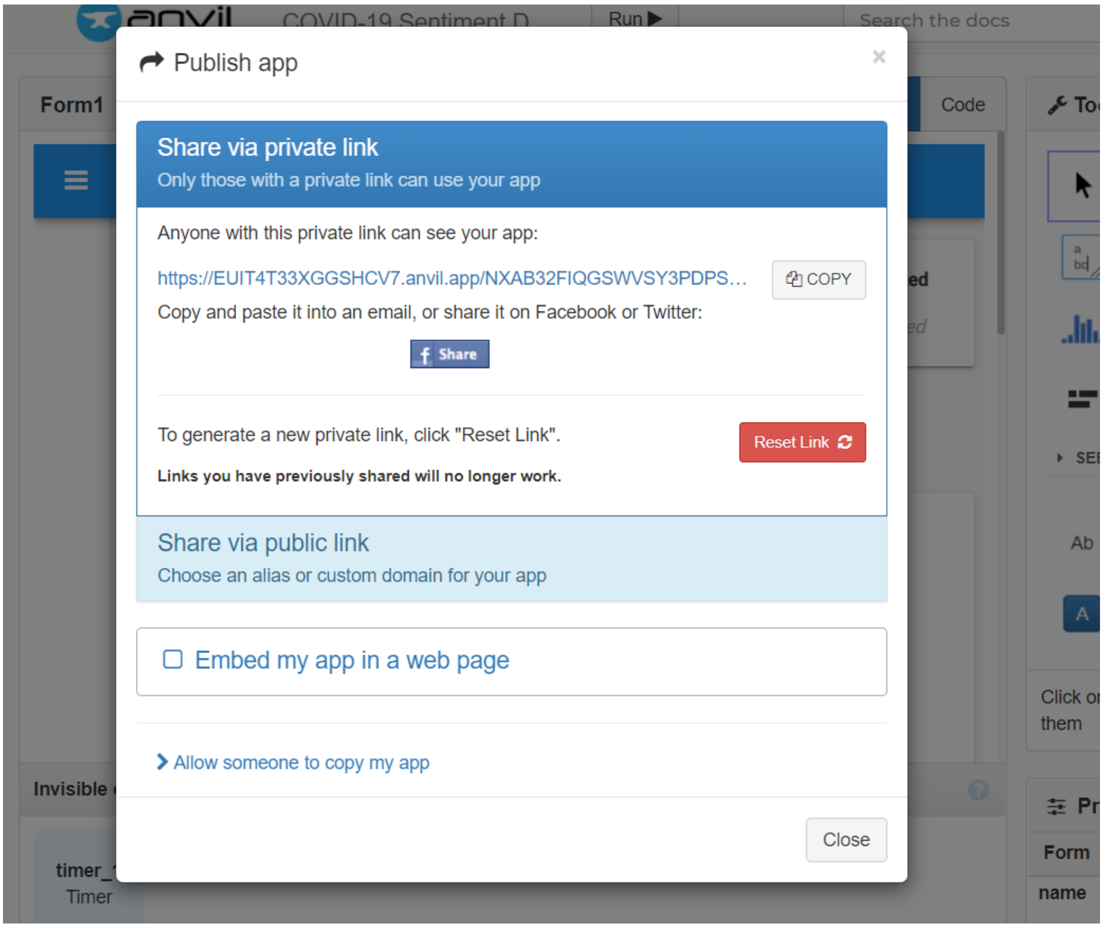
6. Create a callable function



```
@anvil.server.callable
def get_bar_chart():
  #Show the value counts
  df['Analysis'].value_counts()
  fig1 = plt.figure(figsize= (8,6))
  #plot and visualize the counts
  plt.title('Sentiment Analysis')
  plt.xlabel('Sentiment')
  plt.ylabel('Counts')
  df['Analysis'].value_counts().plot(kind='bar')

  fig1 = anvil.mpl_util.plot_image()

  return fig1

anvil.server.wait_forever()
```

The code snippet above shows on how to get the bar chart generated from Google Colab backend to be displayed in Anvil web app. *@anvil.server.callable* is required so that Anvil acknowledge the function can be called. Other than that, *anvil.server.wait_forever()* to keep the backend running and allows Anvil to call the function indefinitely.

7. Publish the app



Publish app

Share via private link

Only those with a private link can use your app

Anyone with this private link can see your app:

https://EUIT4T33XGGSHCV7.anvil.app/NXAB32FIQGSWVSY3PDPS…

Copy and paste it into an email, or share it on Facebook or Twitter:

To generate a new private link, click "Reset Link".

Links you have previously shared will no longer work.

Share via public link

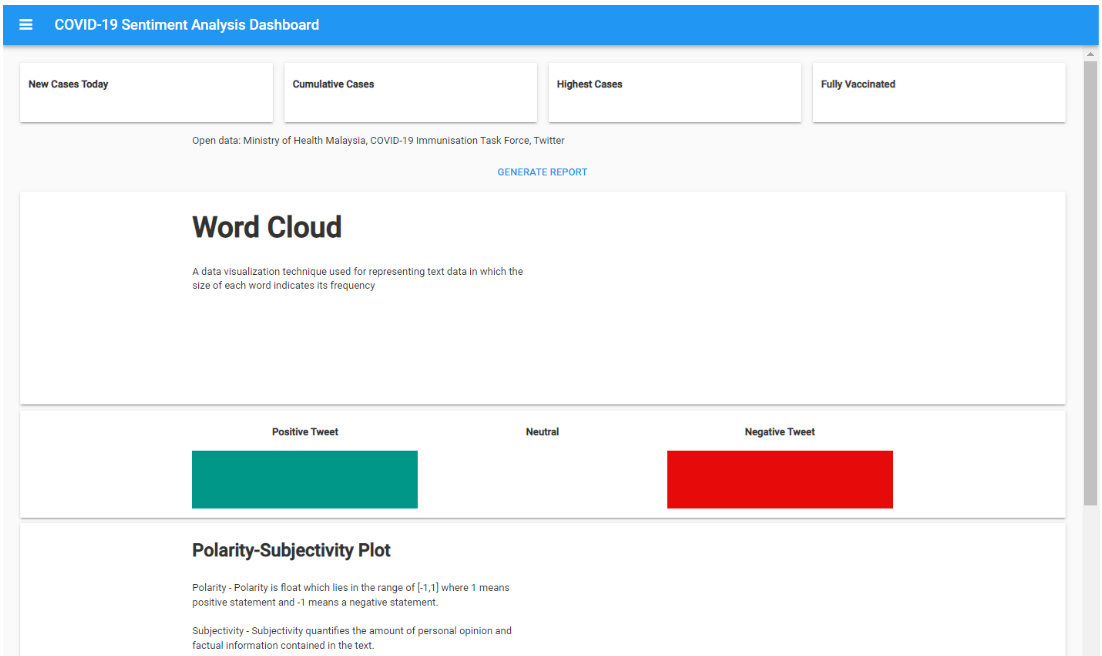
Choose an alias or custom domain for your app

□ Embed my app in a web page

That will be the process of connecting Google Colab backend to Anvil web app in this project.

4.2 Finished prototype

4.2.1 Initial screen



COVID-19 Sentiment Analysis Dashboard

Open data: Ministry of Health Malaysia, COVID-19 Immunisation Task Force, Twitter

Word Cloud

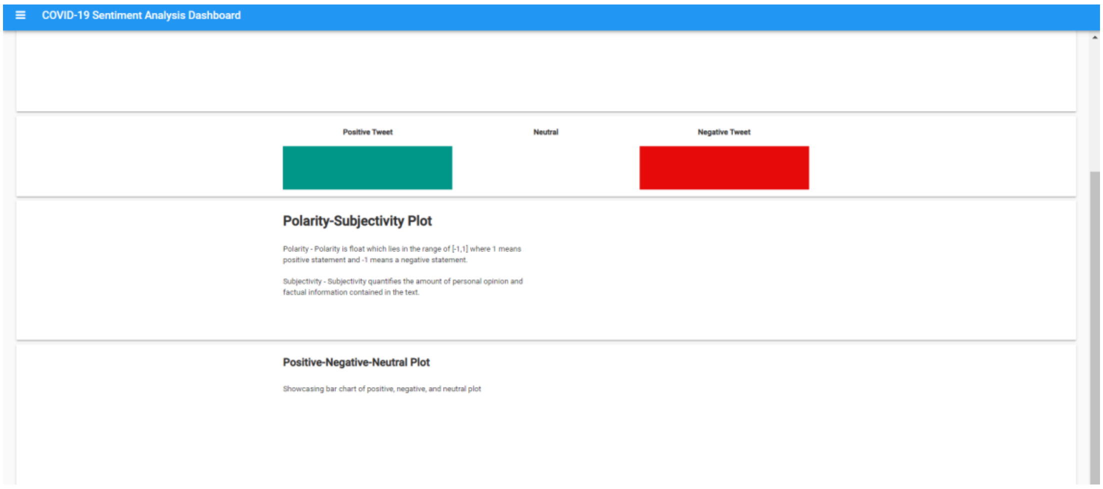
A data visualization technique used for representing text data in which the size of each word indicates its frequency

Polarity-Subjectivity Plot

Polarity - Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement.

Subjectivity - Subjectivity quantifies the amount of personal opinion and factual information contained in the text.

Figure 4.2: Initial Screen



COVID-19 Sentiment Analysis Dashboard

Polarity-Subjectivity Plot

Polarity - Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement.

Subjectivity - Subjectivity quantifies the amount of personal opinion and factual information contained in the text.

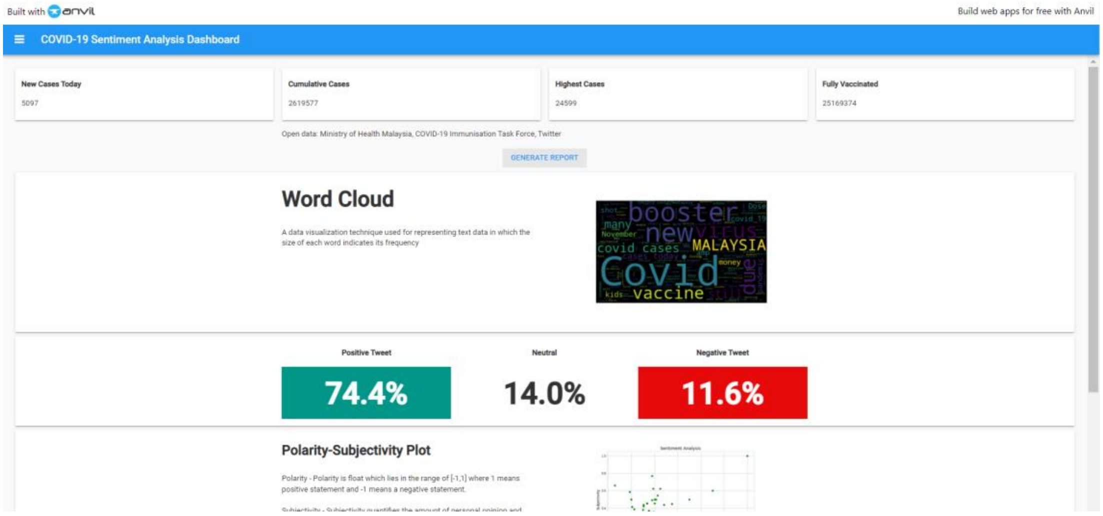
Positive-Negative-Neutral Plot

Showcasing bar chart of positive, negative, and neutral plot

Figure 4.3: Initial Screen

Figure 16 and Figure 17 shows the initial screen of the dashboard in which the user will see right after they opened the dashboard. In the initial screen, there are no values or graphs displayed in any place in the dashboard before clicking the ‘generate report’ button.

4.2.2 Results screen



COVID-19 Sentiment Analysis Dashboard

New Cases Today	Cumulative Cases	Highest Cases	Fully Vaccinated
5097	2619577	24599	25169374

Open data: Ministry of Health Malaysia, COVID-19 Immunisation Task Force, Twitter

Word Cloud

A data visualization technique used for representing text data in which the size of each word indicates its frequency

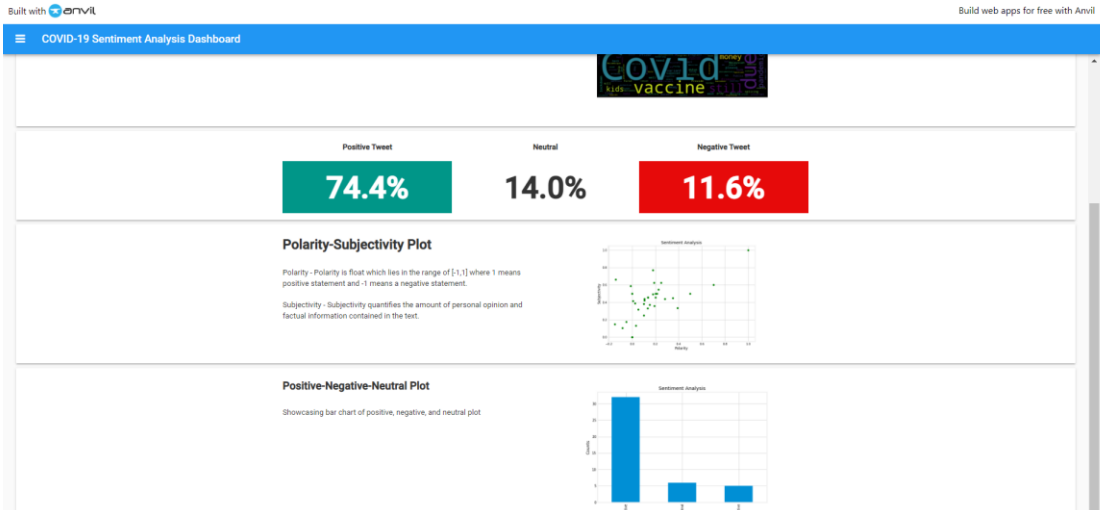
Positive Tweet	Neutral	Negative Tweet
74.4%	14.0%	11.6%

Polarity-Subjectivity Plot

Polarity - Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement.

Subjectivity - Subjectivity quantifies the amount of personal opinion and

Figure 4.4: Results screen



COVID-19 Sentiment Analysis Dashboard

Positive Tweet	Neutral	Negative Tweet
74.4%	14.0%	11.6%

Polarity-Subjectivity Plot

Polarity - Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement.

Subjectivity - Subjectivity quantifies the amount of personal opinion and factual information contained in the text.

Positive-Negative-Neutral Plot

Showcasing bar chart of positive, negative, and neutral plot

Figure 4.5: Results screen

Figure 18 and Figure 19 shows the dashboard after a user clicked on ‘generate report’ button. The dashboard will display KKM open data results, numbers of positive, negative, and neutral tweet, the polarity-subjectivity plot, and the positive-negative-neutral plot.

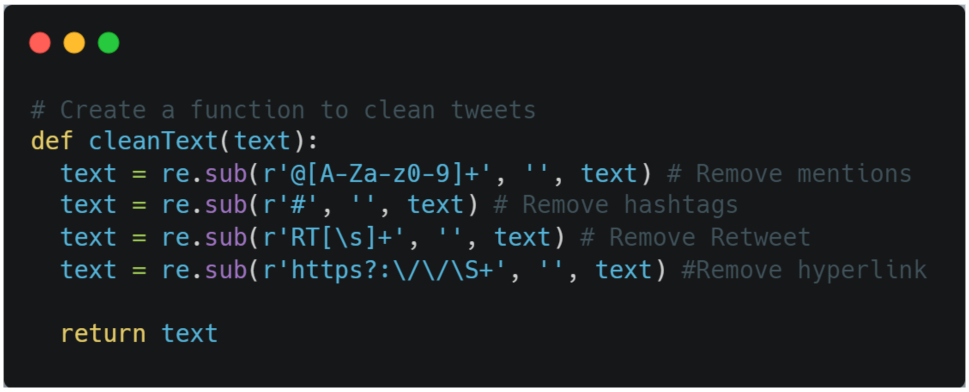
4.3 Unit Testing

Unit testing is used as part of this project because it helps to maintain and change the code if any defects are detected during the process of testing. Unit testing executes features by units to make sure everything can run on its own. Good unit of test cases helps code owner to discover defects and bugs in a particular function. Writing test case for a function will be done only once unless the code structure is changed based on the requirements. The test case is reusable which makes writing the test case takes time, but the time taken to execute the test case will take less amount of time. Therefore, unit testing is the most suitable testing method to be used in this project because if one feature fails, it will affect other feature that involves plotting graphs.

In Python, unit testing can be performed by using *unittest* framework which originally inspired by JUnit. *Unittest* supports test automation, setup sharing, aggregation of test into collections, and independence of the tests from the reporting framework. *Unittest* contains concepts like test fixture, test case, test suite, and test runner to achieve the testing.

4.3.1 List of features to be tested

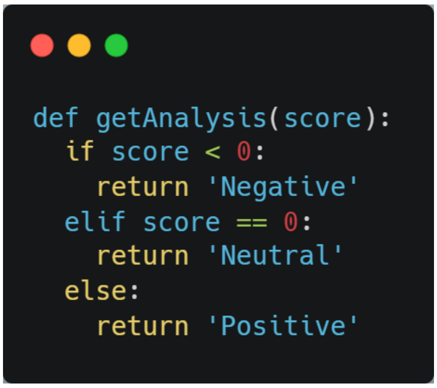
1. Clean Tweets.



```python
# Create a function to clean tweets
def cleanText(text):
  text = re.sub(r'@[A-Za-z0-9]+', '', text) # Remove mentions
  text = re.sub(r'#', '', text) # Remove hashtags
  text = re.sub(r'RT[\s]+', '', text) # Remove Retweet
  text = re.sub(r'https?:\/\/\S+', '', text) #Remove hyperlink

  return text
```

***Figure 4.6**: Clean Tweets*

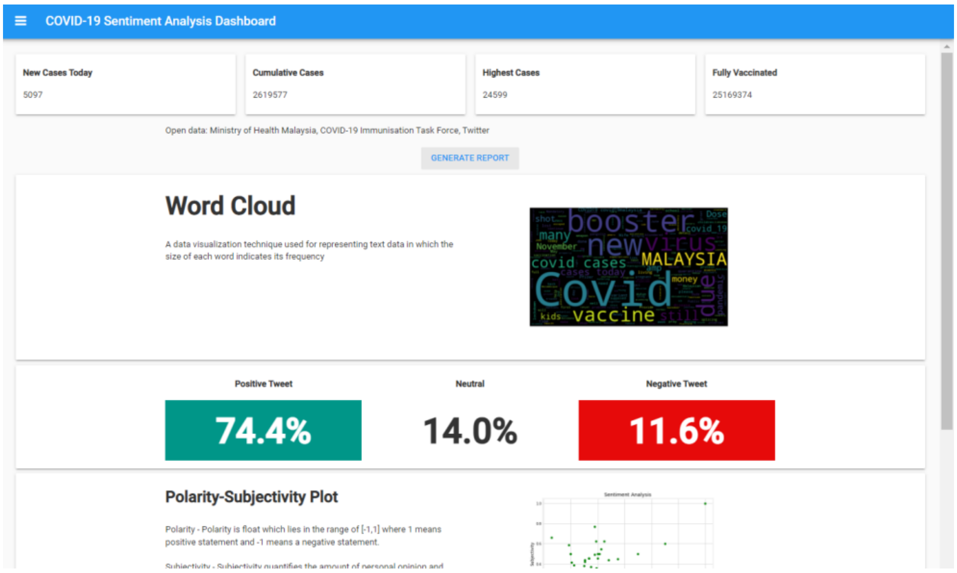
2. Sorting positive and negative sentiment based on sentiment score.



```python
def getAnalysis(score):
  if score < 0:
    return 'Negative'
  elif score == 0:
    return 'Neutral'
  else:
    return 'Positive'
```

Figure 4.7: Sentiment Results

3. Dashboard



COVID-19 Sentiment Analysis Dashboard

New Cases Today	Cumulative Cases	Highest Cases	Fully Vaccinated
5097	2619577	24599	25169374

Open data: Ministry of Health Malaysia, COVID-19 Immunisation Task Force, Twitter

Word Cloud

A data visualization technique used for representing text data in which the size of each word indicates its frequency

Positive Tweet	Neutral	Negative Tweet
74.4%	14.0%	11.6%

Polarity-Subjectivity Plot

Polarity - Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement.

Subjectivity - Subjectivity quantifies the amount of personal opinion and

Figure 4.8: Dashboard

4.3.2 Attributes and Characteristics of Features

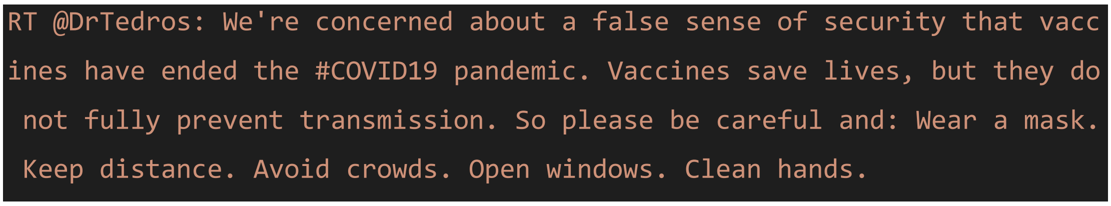
***Table 4.7**: Feature attributes and characteristics*

Feature ID	Feature Name	Attributes	Characteristics
1.0	Clean Tweets	❖ *text* – Tweet data	This feature receives a tweet text data to remove mentions, remove hashtags, remove retweet, and remove hyperlink.
2.0	Sorting positive and negative sentiment based on sentiment score.	❖ *score* – Sentiment score generated	This feature receives sentiment score generated from backend API to sort the tweet data to positive, negative, or neutral.
3.0	Dashboard	❖ *latest_new_cases* ❖ *cumulative_cases* ❖ *highest_cases* ❖ *fully_vaccinated* ❖ *bar_chart* ❖ *get_wordcloud* ❖ *polarity_subjectivity* ❖ *get_positive* ❖ *get_negative* ❖ *get_neutral*	This feature is a dashboard feature that showcases results from backend API to be displayed on the designed interface.

4.3.3 Test Data

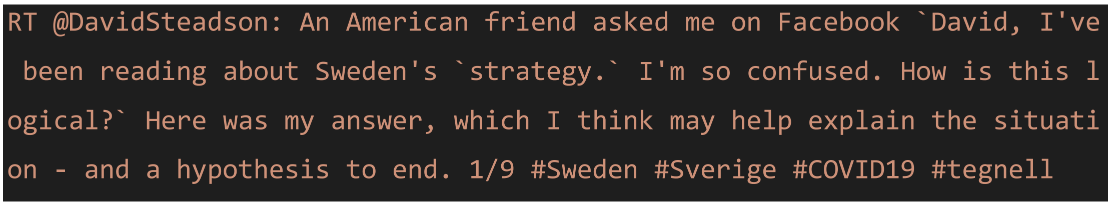
FEATURE ID: 1.0

Test Data 1:



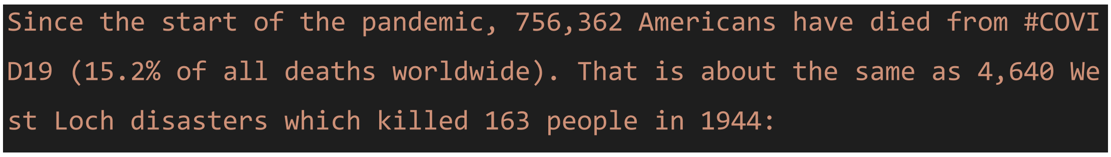
```
RT @DrTedros: We're concerned about a false sense of security that vacc
ines have ended the #COVID19 pandemic. Vaccines save lives, but they do
 not fully prevent transmission. So please be careful and: Wear a mask.
 Keep distance. Avoid crowds. Open windows. Clean hands.
```

Test Data 2:



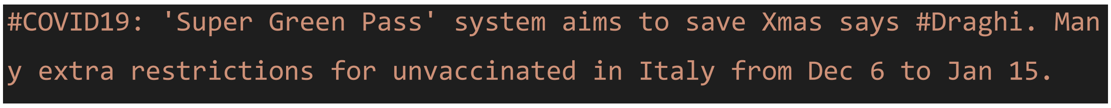
```
RT @DavidSteadson: An American friend asked me on Facebook `David, I've
 been reading about Sweden's `strategy.` I'm so confused. How is this l
ogical?` Here was my answer, which I think may help explain the situati
on - and a hypothesis to end. 1/9 #Sweden #Sverige #COVID19 #tegnell
```

Test Data 3:



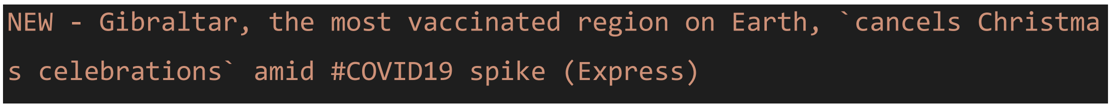
```
Since the start of the pandemic, 756,362 Americans have died from #COVI
D19 (15.2% of all deaths worldwide). That is about the same as 4,640 We
st Loch disasters which killed 163 people in 1944:
```

Test Data 4:



```
#COVID19: 'Super Green Pass' system aims to save Xmas says #Draghi. Man
y extra restrictions for unvaccinated in Italy from Dec 6 to Jan 15.
```

Test Data 5:



```
NEW - Gibraltar, the most vaccinated region on Earth, `cancels Christma
s celebrations` amid #COVID19 spike (Express)
```

FEATURE 2.0

Table 4.8: Feature 2.0 Test Data

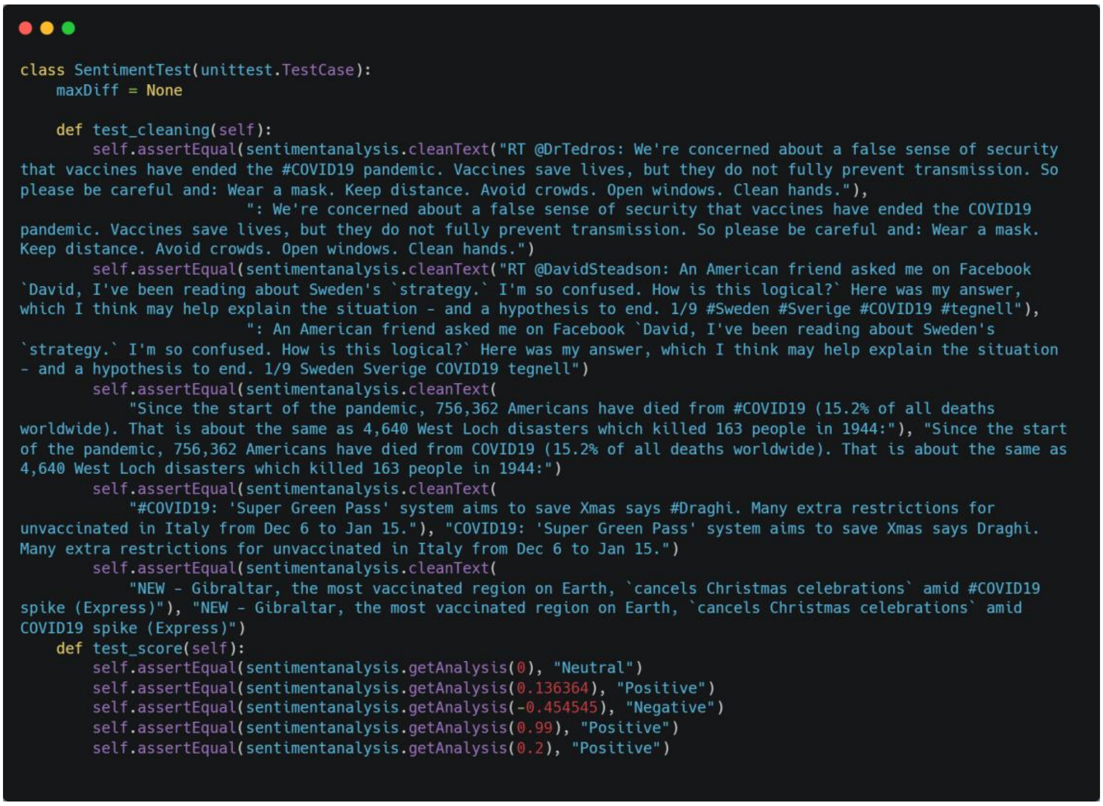
Test Data 1	score ➔ 0
Test Data 2	score ➔ 0.136364
Test Data 3	score ➔ - 0.454545
Test Data 4	score ➔ 0.99
Test Data 5	score ➔ 0.2

FEATURE 3.0

Table 4.9: Feature 3.0 Testing Checklist

Item	Checklist
Able to get COVID-19 latest new cases from KKM open data	❖ Data displayed correctly ❖ Data displayed with a correct font ❖ Data displayed at the designated placement
Able to get COVID-19 cumulative cases from KKM open data	❖ Data displayed correctly ❖ Data displayed with a correct font ❖ Data displayed at the designated placement
Able to get COVID-19 highest cases from KKM open data	❖ Data displayed correctly ❖ Data displayed with a correct font ❖ Data displayed at the designated placement
Able to get fully vaccinated numbers from KKM open data	❖ Data displayed correctly ❖ Data displayed with a correct font ❖ Data displayed at the designated placement
WordCloud	❖ Data displayed correctly ❖ Data displayed at the designated placement
Positive-Negative-Neutral Bar chart	❖ Positive, Negative, and Neutral Data displayed correctly ❖ Data displayed matches the backend result ❖ Data displayed at the designated placement
Plotting of subjectivity and polarity	❖ Data displayed correctly ❖ Data displayed matches the backend result ❖ Data displayed at the designated placement

4.3.4 Test Suites



```
class SentimentTest(unittest.TestCase):
    maxDiff = None

    def test_cleaning(self):
        self.assertEqual(sentimentanalysis.cleanText("RT @DrTedros: We're concerned about a false sense of security
that vaccines have ended the #COVID19 pandemic. Vaccines save lives, but they do not fully prevent transmission. So
please be careful and: Wear a mask. Keep distance. Avoid crowds. Open windows. Clean hands."),
                             ": We're concerned about a false sense of security that vaccines have ended the COVID19
pandemic. Vaccines save lives, but they do not fully prevent transmission. So please be careful and: Wear a mask.
Keep distance. Avoid crowds. Open windows. Clean hands.")
        self.assertEqual(sentimentanalysis.cleanText("RT @DavidSteadson: An American friend asked me on Facebook
`David, I've been reading about Sweden's `strategy.` I'm so confused. How is this logical?` Here was my answer,
which I think may help explain the situation - and a hypothesis to end. 1/9 #Sweden #Sverige #COVID19 #tegnell"),
                             ": An American friend asked me on Facebook `David, I've been reading about Sweden's
`strategy.` I'm so confused. How is this logical?` Here was my answer, which I think may help explain the situation
- and a hypothesis to end. 1/9 Sweden Sverige COVID19 tegnell")
        self.assertEqual(sentimentanalysis.cleanText(
            "Since the start of the pandemic, 756,362 Americans have died from #COVID19 (15.2% of all deaths
worldwide). That is about the same as 4,640 West Loch disasters which killed 163 people in 1944:"), "Since the start
of the pandemic, 756,362 Americans have died from COVID19 (15.2% of all deaths worldwide). That is about the same as
4,640 West Loch disasters which killed 163 people in 1944:")
        self.assertEqual(sentimentanalysis.cleanText(
            "#COVID19: 'Super Green Pass' system aims to save Xmas says #Draghi. Many extra restrictions for
unvaccinated in Italy from Dec 6 to Jan 15."), "COVID19: 'Super Green Pass' system aims to save Xmas says Draghi.
Many extra restrictions for unvaccinated in Italy from Dec 6 to Jan 15.")
        self.assertEqual(sentimentanalysis.cleanText(
            "NEW - Gibraltar, the most vaccinated region on Earth, `cancels Christmas celebrations` amid #COVID19
spike (Express)"), "NEW - Gibraltar, the most vaccinated region on Earth, `cancels Christmas celebrations` amid
COVID19 spike (Express)")
    def test_score(self):
        self.assertEqual(sentimentanalysis.getAnalysis(0), "Neutral")
        self.assertEqual(sentimentanalysis.getAnalysis(0.136364), "Positive")
        self.assertEqual(sentimentanalysis.getAnalysis(-0.454545), "Negative")
        self.assertEqual(sentimentanalysis.getAnalysis(0.99), "Positive")
        self.assertEqual(sentimentanalysis.getAnalysis(0.2), "Positive")
```

***Figure 4.9**: Test Suites*

Test suites is the collection of test cases. Figure 23 shows the test suite of Feature 1.0 and Feature 2.0. The test suite utilizing *unittest* framework from Python to create test cases. The test cases must be aligned with actual functions in the backend. Running this test suite will generate a report of how many tests is being run.

4.3.5 Test Results

FEATURE 1.0

Test Data 1

***Table 4.10**: Test Result 1 (F1)*

Data	RT @DrTedros: We're concerned about a false sense of security that vaccines have ended the #COVID19 pandemic. Vaccines save lives, but they do not fully prevent transmission. So please be careful and: Wear a mask. Keep distance. Avoid crowds. Open windows. Clean hands.
Expected Output	: We're concerned about a false sense of security that vaccines have ended the COVID19 pandemic. Vaccines save lives, but they do not fully prevent transmission. So please be careful and: Wear a mask. Keep distance. Avoid crowds. Open windows. Clean hands.
Actual Output	: We're concerned about a false sense of security that vaccines have ended the COVID19 pandemic. Vaccines save lives, but they do not fully prevent transmission. So please be careful and: Wear a mask. Keep distance. Avoid crowds. Open windows. Clean hands.
Pass/Fail	Pass

Test Data 2

Table 4.11: Test Result 2 (F1)

Data	RT @DavidSteadson: An American friend asked me on Facebook `David, I've been reading about Sweden's `strategy.` I'm so confused. How is this logical?` Here was my answer, which I think may help explain the situation - and a hypothesis to end. 1/9 #Sweden #Sverige #COVID19 #tegnell
Expected Output	: An American friend asked me on Facebook `David, I've been reading about Sweden's `strategy.` I'm so confused. How is this logical?` Here was my answer, which I think may help explain the situation - and a hypothesis to end. 1/9 Sweden Sverige COVID19 tegnell
Actual Output	: An American friend asked me on Facebook `David, I've been reading about Sweden's `strategy.` I'm so confused. How is this logical?` Here was my answer, which I think may help explain the situation - and a hypothesis to end. 1/9 Sweden Sverige COVID19 tegnell
Pass/Fail	Pass

Test Data 3

***Table 4.12**: Test Result 3 (F1)*

Data	Since the start of the pandemic, 756,362 Americans have died from #COVID19 (15.2% of all deaths worldwide). That is about the same as 4,640 West Loch disasters which killed 163 people in 1944:
Expected Output	Since the start of the pandemic, 756,362 Americans have died from COVID19 (15.2% of all deaths worldwide). That is about the same as 4,640 West Loch disasters which killed 163 people in 1944:
Actual Output	Since the start of the pandemic, 756,362 Americans have died from COVID19 (15.2% of all deaths worldwide). That is about the same as 4,640 West Loch disasters which killed 163 people in 1944:
Pass/Fail	Pass

Test Data 4

***Table 4.13**: Test Result 4 (F1)*

Data	#COVID19: 'Super Green Pass' system aims to save Xmas says #Draghi. Many extra restrictions for unvaccinated in Italy from Dec 6 to Jan 15.
Expected Output	COVID19: 'Super Green Pass' system aims to save Xmas says Draghi. Many extra restrictions for unvaccinated in Italy from Dec 6 to Jan 15.
Actual Output	COVID19: 'Super Green Pass' system aims to save Xmas says Draghi. Many extra restrictions for unvaccinated in Italy from Dec 6 to Jan 15.
Pass/Fail	Pass

Test Data 5

***Table 4.14**: Test Result 5 (F1)*

Data	NEW - Gibraltar, the most vaccinated region on Earth, `cancels Christmas celebrations` amid #COVID19 spike (Express)
Expected Output	NEW - Gibraltar, the most vaccinated region on Earth, `cancels Christmas celebrations` amid COVID19 spike (Express)
Actual Output	NEW - Gibraltar, the most vaccinated region on Earth, `cancels Christmas celebrations` amid COVID19 spike (Express)
Pass/Fail	Pass

FEATURE 2.0

Test Data 1

***Table 4.15**: Test Result 1 (F2)*

Data	0
Expected Output	Neutral
Actual Output	Neutral
Pass/Fail	Pass

Test Data 2

***Table 4.16**: Test Result 2 (F2)*

Data	0.136364
Expected Output	Positive
Actual Output	Positive
Pass/Fail	Pass

Test Data 3

***Table 4.17**: Test Result 3 (F2)*

Data	-0.454545

Expected Output	Negative
Actual Output	Negative
Pass/Fail	Pass

Test Data 4

Table 4.18: Test Result 4 (F2)

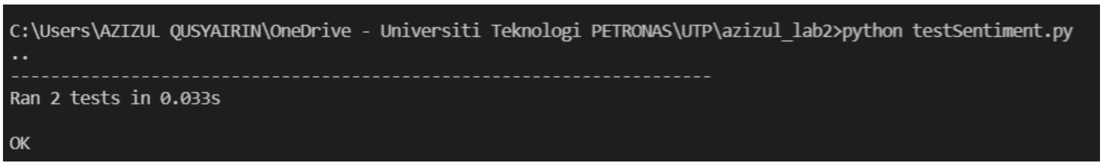
Data	0.99
Expected Output	Positive
Actual Output	Positive
Pass/Fail	Pass

Test Data 5

Table 4.19: Test Result 5 (F2)

Data	0.2
Expected Output	Positive
Actual Output	Positive
Pass/Fail	Pass

Overall Test Results for Feature 1.0 and Feature 2.0

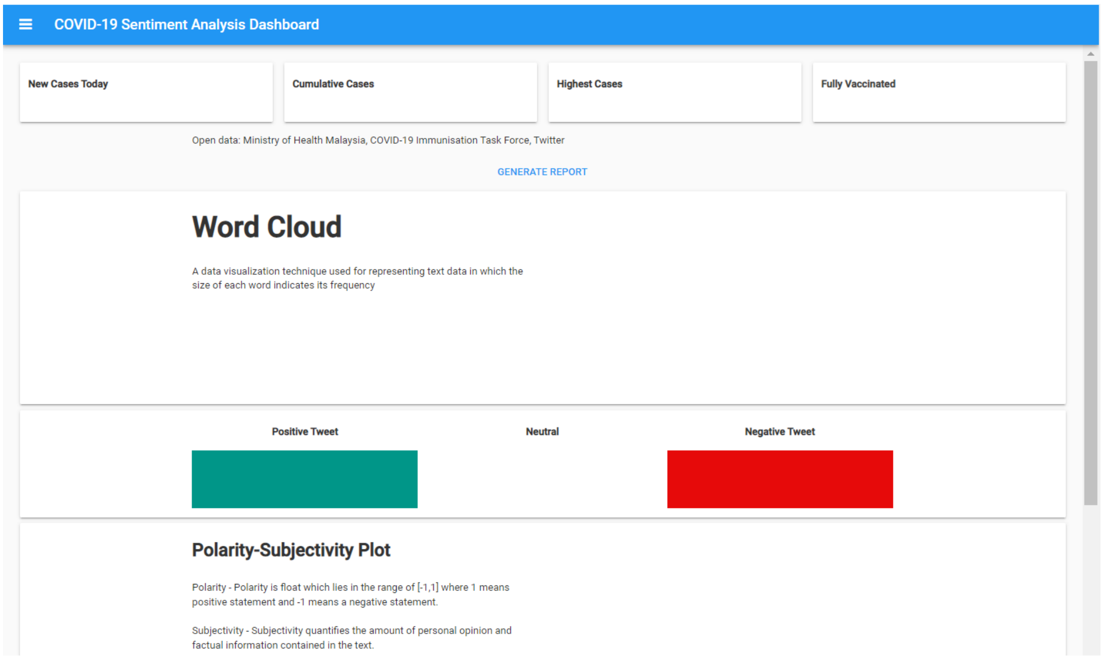


```
C:\Users\AZIZUL QUSYAIRIN\OneDrive - Universiti Teknologi PETRONAS\UTP\azizul_lab2>python testSentiment.py
..
----------------------------------------------------------------------
Ran 2 tests in 0.033s

OK
```

Figure 4.10: Overall test results

The unit testing ran 2 tests in 0.033s in *testSentiment.py* file. Both tests for Feature 1.0 and Feature 2.0 passed the unit test.

FEATURE 3.0



COVID-19 Sentiment Analysis Dashboard

New Cases Today	Cumulative Cases	Highest Cases	Fully Vaccinated

Open data: Ministry of Health Malaysia, COVID-19 Immunisation Task Force, Twitter

Word Cloud

A data visualization technique used for representing text data in which the size of each word indicates its frequency

Positive Tweet	Neutral	Negative Tweet
🟩		🟥

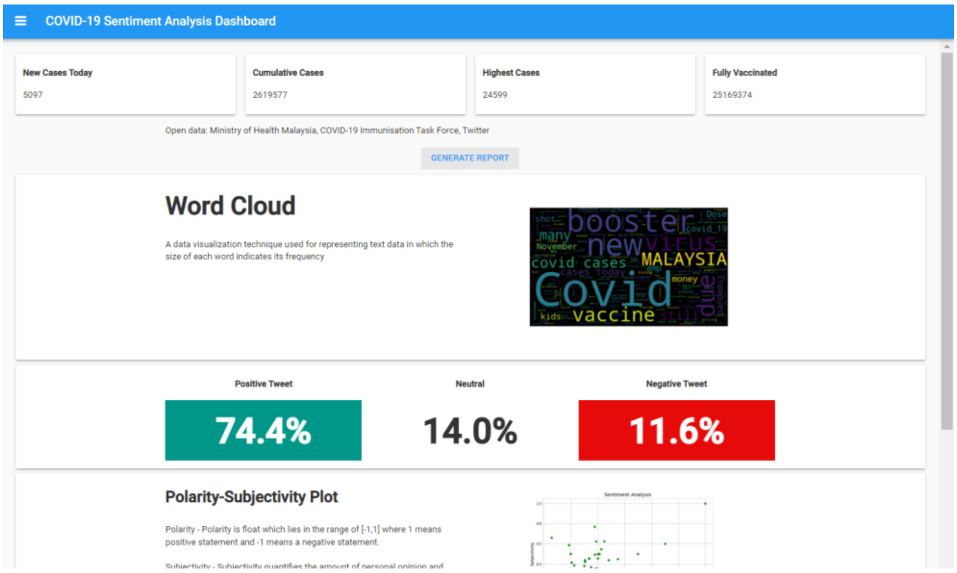
Polarity-Subjectivity Plot

Polarity - Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement.

Subjectivity - Subjectivity quantifies the amount of personal opinion and factual information contained in the text.

Figure 4.11: Dashboard

The initial screen shows the dashboard before user starts to click on Generate Report button that will generate COVID-19 number of new cases, cumulative cases, highest cases, and fully vaccinated. Other than that, the dashboard will also generate a bar chart, a wordcloud, and a plotting graph showing the result of the sentiment analysis.



COVID-19 Sentiment Analysis Dashboard

New Cases Today	Cumulative Cases	Highest Cases	Fully Vaccinated
5097	2619577	24599	25169374

Open data: Ministry of Health Malaysia, COVID-19 Immunisation Task Force, Twitter

Word Cloud

A data visualization technique used for representing text data in which the size of each word indicates its frequency

Positive Tweet	Neutral	Negative Tweet
74.4%	14.0%	11.6%

Polarity-Subjectivity Plot

Polarity - Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement.

Subjectivity - Subjectivity quantifies the amount of personal opinion and

Figure 4.12: Dashboard with Results

The post-screen shows the results when user click on the Generate Report button. The component of latest COVID-19 statistics and the results of the sentiment analysis are displayed.

Able to get COVID-19 latest new cases from KKM open data

Expected output: 5097

Actual output: 5097

Status: Pass

Able to get COVID-19 cumulative cases from KKM open data

Expected output: 2619577

Actual output: 2619577

Status: Pass

Able to get COVID-19 highest cases from KKM open data

Expected output: 24599

Actual output: 24599

Status: Pass

Able to get fully vaccinated numbers from KKM open data

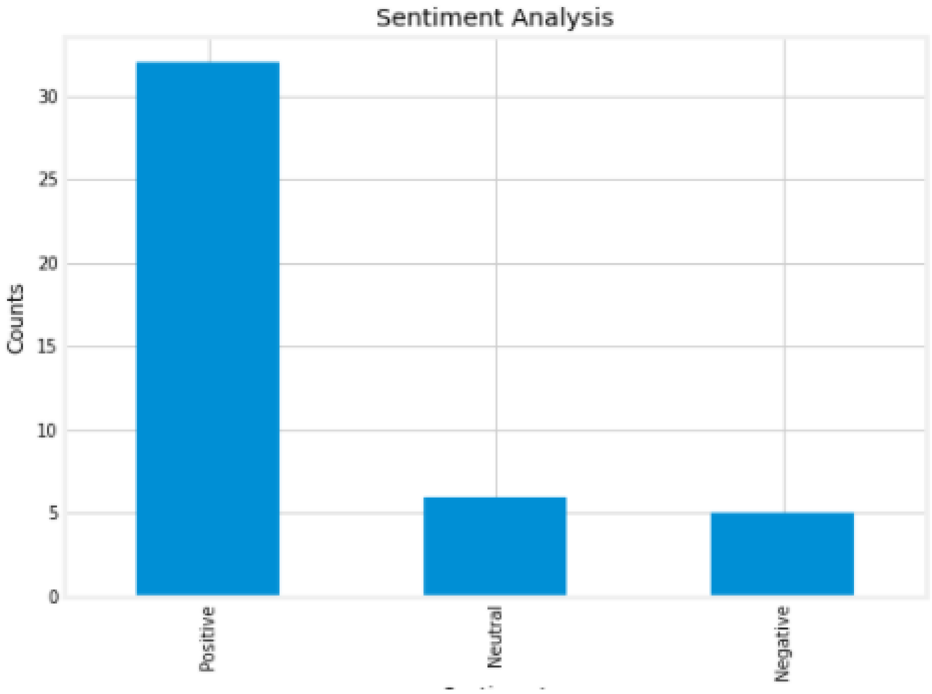
Expected output: 25169374

Actual output: 25169374

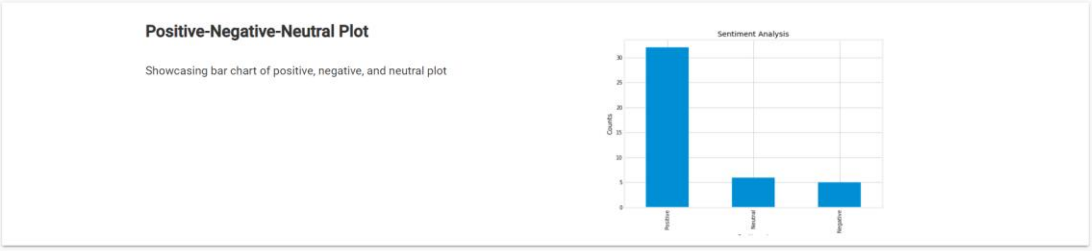
Status: Pass

BAR CHART

Expected Output (data from Backend):



Actual output (dashboard):



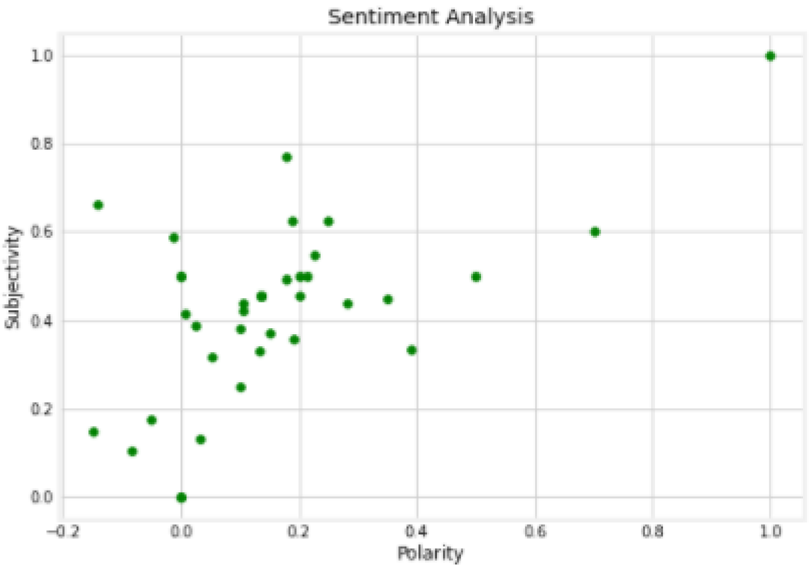
Positive-Negative-Neutral Plot

Showcasing bar chart of positive, negative, and neutral plot

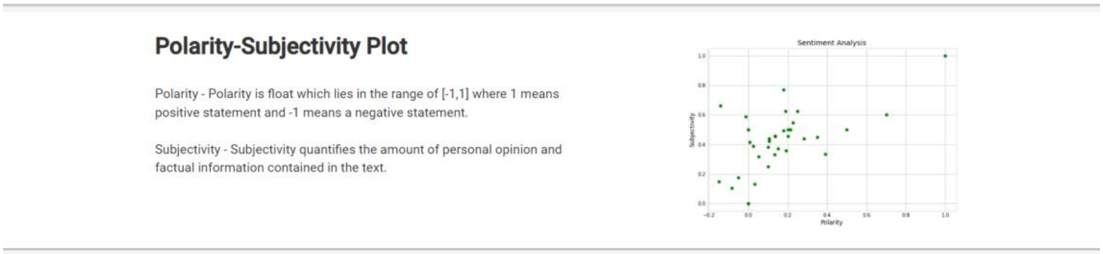
Status: Pass

POLARITY-SUBJECTIVITY PLOT

Expected Output (data from Backend):



Actual output (dashboard):



Polarity-Subjectivity Plot

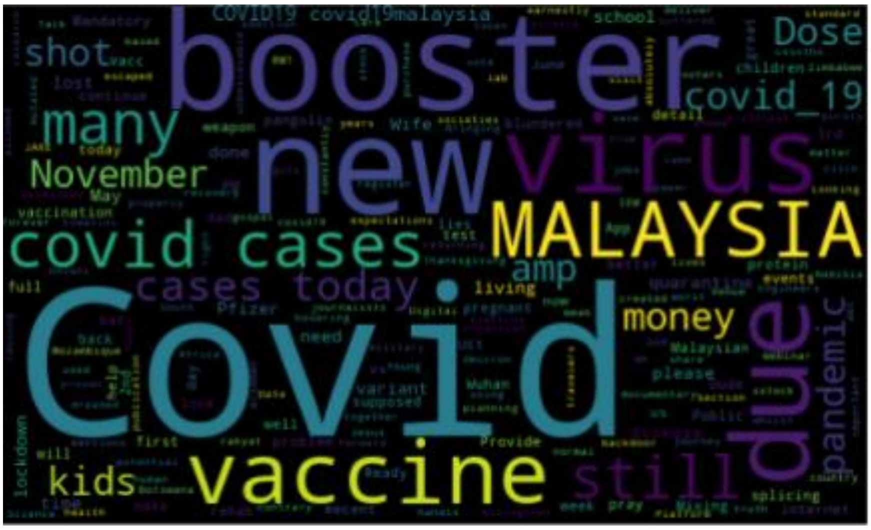
Polarity - Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement.

Subjectivity - Subjectivity quantifies the amount of personal opinion and factual information contained in the text.

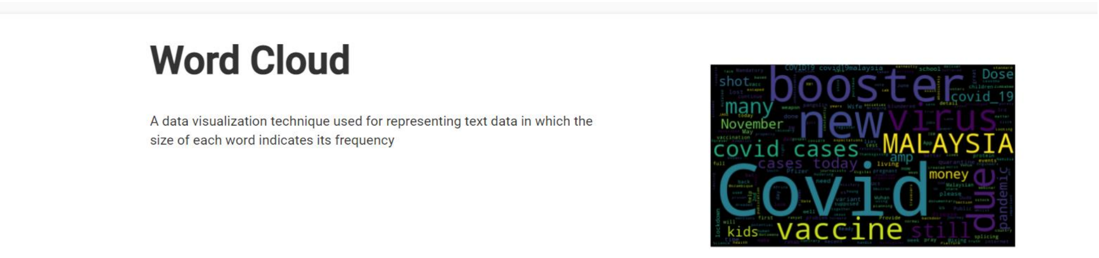
Status: Pass

WORDCLOUD

Expected Output (data from Backend):



Actual output (dashboard):



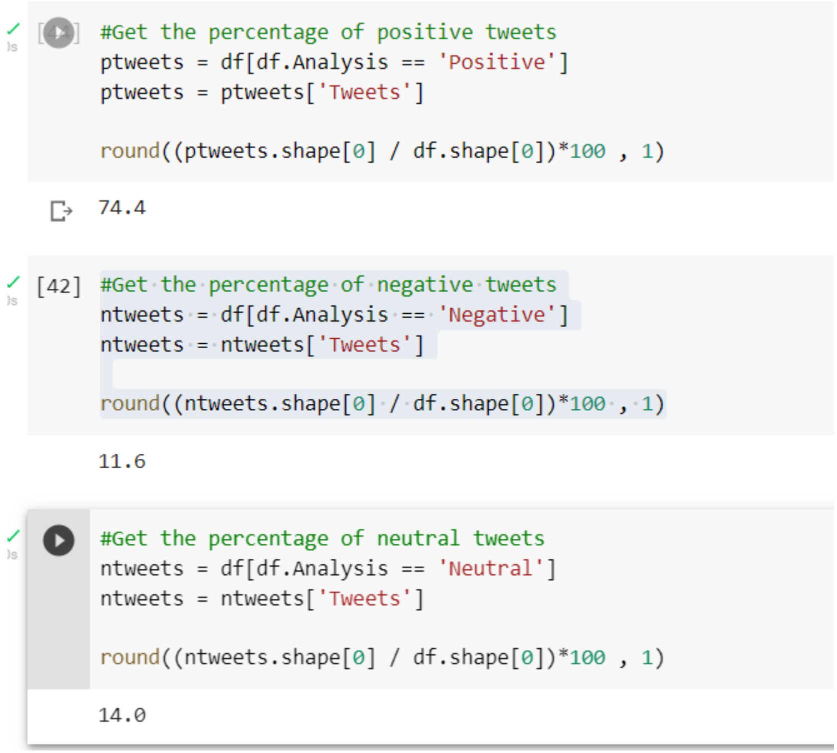
Word Cloud

A data visualization technique used for representing text data in which the size of each word indicates its frequency

Status: Pass

NUMBER OF POSITIVE, NEGATIVE AND NEUTRAL TWEET

Expected Output (data from Backend):



```
#Get the percentage of positive tweets
ptweets = df[df.Analysis == 'Positive']
ptweets = ptweets['Tweets']

round((ptweets.shape[0] / df.shape[0])*100 , 1)
```

```
74.4
```

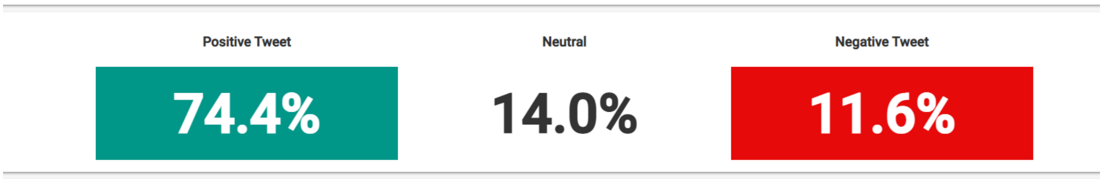
```
[42] #Get the percentage of negative tweets
ntweets = df[df.Analysis == 'Negative']
ntweets = ntweets['Tweets']

round((ntweets.shape[0] / df.shape[0])*100 , 1)
```

```
11.6
```

```
#Get the percentage of neutral tweets
ntweets = df[df.Analysis == 'Neutral']
ntweets = ntweets['Tweets']

round((ntweets.shape[0] / df.shape[0])*100 , 1)
```

```
14.0
```

Actual output (dashboard):



Positive Tweet	Neutral	Negative Tweet
74.4%	14.0%	11.6%

Status: Pass

CHECKLIST

Table 4.20: Checklist Result

Item	Checklist	Pass/Fail
Able to get COVID-19 latest new cases from KKM open data	Data displayed correctly	Pass
	Data displayed with a correct font	Pass
	Data displayed at the designated placement	Pass
Able to get COVID-19 cumulative cases from KKM open data	Data displayed correctly	Pass
	Data displayed with a correct font	Pass
	Data displayed at the designated placement	Pass
Able to get COVID-19 highest cases from KKM open data	Data displayed correctly	Pass
	Data displayed with a correct font	Pass
	Data displayed at the designated placement	Pass
Able to get fully vaccinated numbers from KKM open data	Data displayed correctly	Pass
	Data displayed with a correct font	Pass
	Data displayed at the designated placement	Pass
WordCloud	Data displayed correctly	Pass
	Data displayed at the designated placement	Pass
Positive-Negative-Neutral Bar chart	Positive, Negative, and Neutral Data displayed correctly	Pass
	Data displayed matches the backend result	Pass
	Data displayed at the designated placement	Pass
Plotting of subjectivity and polarity	Data displayed correctly	Pass
	Data displayed matches the backend result	Pass
	Data displayed at the designated placement	Pass

CHAPTER 5: Conclusion and Recommendation

This chapter summarises the findings of the completed research as well as potential future projects. It includes a summary of the main objectives, a research analysis, a study scope, and suggestions for future improvements.

5.1 Research Summary

The sentiment analysis of COVID-19 related issues will be performed by using Naïve-Bayes algorithm that classifies the results to polarity and subjectivity index. National Vaccination Programme, Movement Control Order (MCO) and the number of daily cases are the issues that is analysed in this project. The issues are widely discussed among Twitter users which makes opinion mining more effective. Naïve-Bayes method used is available in the libraries of various programming language. However, conclusions made from the results must be designed from an algorithm to make it understandable instead of showing numbers. All the results of the analysis will be displayed through a dashboard.

5.2 Achievement of Objectives

This research work has fulfilled all objectives as outlined in Chapter 1. The following are the achievements with respect to the objectives of this research work:

1. To develop an algorithm that analyses people's opinions on Twitter by using Naïve-Bayes approach.

The algorithm started with getting tweet data, cleaning tweet data with a set of feature, perform sentiment analysis through TextBlob’s Naïve-Bayes classification, and visualizing the sentiment analysis results on a dashboard.

2. To conclude the algorithm's results in analysing people's opinion through the sentiment score.

An algorithm to sort out the sentiment score to positive, negative, or neutral was developed. Polarity of 0 will be assigned to neutral, polarity less than 0 will be assigned to negative, and polarity more than 0 will be assigned to positive. Other than that, the percentage of tweet data getting a particular sentiment score is also developed to be displayed in the dashboard.

3. To develop a dashboard that visualizes the sentiment analysis result into graph and numerical score.

A dashboard was developed by connecting Google Colab (backend) to Anvil (front-end) to successfully created a responsive app through the button of ‘Generate Report’. Every necessary information on the sentiment analysis results visualized in the dashboard.

5.2 Recommendations

Sentiment analysis field is less popular compared to another Python sub-field. Sentiment analysis is an exciting area because people are widely expressing their emotions and opinions in the social media. It is almost becoming a main platform for everyone to get connected while discovering other people’s opinions. Acknowledging other’s opinions on a particular topic helps in better decision-making in various areas such as government policies, security, and business. After a thorough research on sentiment analysis in this project, it can be concluded that Twitter is the main platform to perform sentiment analysis. The existing algorithm are getting better each day; therefore, the way data is represented in sentiment analysis dashboard might bring sentiment analysis becomes one of the ways to decide major decisions in the future especially in politics. However, the existing algorithm still need an improvement in understanding sarcasm and joke better. With that, the sentiment results towards COVID-19 or any diseases might empower the existing health institutions in Malaysia.

REFERENCES

Alayba, A. M. (2020). *Twitter Sentiment Analysis on Health Services in Arabic By*.

Ante, L. (2021). How Elon Musk’s Twitter Activity Moves Cryptocurrency Markets. *SSRN Electronic Journal*, *16*, 1–13. https://doi.org/10.2139/ssrn.3778844

Ashari, A., Paryudi, I., & Tjoa, A. M. (2013). Performance comparison between Naïve Bayes, decision tree and k-nearest neighbor in searching alternative design in an energy simulation tool. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *4*(11).

Bakar, M. A. A., Ariff, N. M., & Hui, E. X. (2018). Exploratory data analysis of Twitter’s rhythm in Malaysia. *AIP Conference Proceedings*, *2013*(1), 20056.

Bakshi, R. K., Kaur, N., Kaur, R., & Kaur, G. (2016). Opinion mining and sentiment analysis. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 452–455.

Brownlee, J. (2016). Machine learning mastery with python. *Machine Learning Mastery Pty Ltd*, *527*, 100–120.

Buckman, S. R., Shapiro, A. H., Sudhof, M., & Wilson, D. J. (2020). News Sentiment in the Time of COVID-19. *FRBSF Economic Letter*, *08*, 1–5. https://www.frbsf.org/economic-

Campan, A., Atnafu, T., Truta, T. M., & Nolan, J. (2018). Is Data Collection through Twitter Streaming API Useful for Academic Research? *2018 IEEE International Conference on Big Data (Big Data)*, 3638–3643. https://doi.org/10.1109/BigData.2018.8621898

Fauci, A. S., Lane, H. C., & Redfield, R. R. (2020). Covid-19 — Navigating the Uncharted. *New England Journal of Medicine*, *382*(13), 1268–1269. https://doi.org/10.1056/NEJMe2002387

Gupta, B., Negi, M., Vishwakarma, K., Rawat, G., & Badhani, P. (2017). Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python. *International Journal of Computer Applications*, *165*(9), 29–34.

https://doi.org/10.5120/ijca2017914022

H. Manguri, K., N. Ramadhan, R., & R. Mohammed Amin, P. (2020). Twitter Sentiment Analysis on Worldwide COVID-19 Outbreaks. *Kurdistan Journal of Applied Research*, 54–65. https://doi.org/10.24017/covid.8

Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, *50*, 723–762. https://doi.org/10.1613/jair.4272

Lee, S.-W., Song, Y.-I., Lee, J.-T., Han, K.-S., & Rim, H.-C. (2012). A New Generative Opinion Retrieval Model Integrating Multiple Ranking Factors. *J. Intell. Inf. Syst.*, *38*(2), 487–505. https://doi.org/10.1007/s10844-011-0164-5

Li, J. (2020). *TWITTER SENTIMENT ANALYSIS DURING COVID-19 IN FLORIDA by JINGYI LI (Under the Direction of Tianming Liu)*.

MURTHY, K. N., & SCHOLAR, P. G. (2020). WORD CLOUD IN PYTHON. *Complexity International*, *24*(01).

Shah, A. U. M., Safri, S. N. A., Thevadas, R., Noordin, N. K., Rahman, A. A., Sekawi, Z., Ideris, A., & Sultan, M. T. H. (2020). COVID-19 outbreak in Malaysia: Actions taken by the Malaysian government. *International Journal of Infectious Diseases*, *97*, 108–116. https://doi.org/https://doi.org/10.1016/j.ijid.2020.05.093

Shahi, G. K., Dirkson, A., & Majchrzak, T. A. (2021). An exploratory study of COVID-19 misinformation on Twitter. *Online Social Networks and Media*, *22*, 100104. https://doi.org/https://doi.org/10.1016/j.osnem.2020.100104

Wong, L. P., & Alias, H. (2021). Temporal changes in psychobehavioural responses during the early phase of the COVID-19 pandemic in Malaysia. *Journal of Behavioral Medicine*, *44*(1), 18–28. https://doi.org/10.1007/s10865-020-00172-z

Wongkar, M., & Angdresey, A. (2019). Sentiment Analysis Using Naive Bayes Algorithm Of The Data Crawler: Twitter. *2019 Fourth International Conference on Informatics and Computing (ICIC)*, 1–5. https://doi.org/10.1109/ICIC47613.2019.8985884

Wren-Lewis, S. (2020). The economic effects of a pandemic. In *Economics in the Time of COVID-19*. https://voxeu.org/content/economics-time-covid-19

Xue, J., Chen, J., Hu, R., Chen, C., Zheng, C., Su, Y., & Zhu, T. (2020). Twitter Discussions and Emotions About the COVID-19 Pandemic: Machine Learning Approach. *J Med Internet Res*, *22*(11), e20550. https://doi.org/10.2196/20550