COLLEGE OF COMPUTING AND INFORMATICS UNIVERSITI TENAGA NASIONAL

ADVANCEMENTS IN MONITORING & FORECASTING FUTURE COVID-19 OUTBREAKS USING PROPHET AND HOLT'S WINTER MODEL

AYMAN FIKRY BIN ASMAJUDA

2025

ADVANCEMENTS IN MONITORING & FORECASTING FUTURE COVID-19 OUTBREAKS USING PROPHET AND HOLT'S WINTER MODEL

by

AYMAN FIKRY BIN ASMAJUDA

Project Supervisor: Ms. Nur Laila Bte Ab Ghani

PROJECT REPORT SUBMITTED IN PARTUAL FULFILLMENT OF THE REQUIREMENTS FOR THE BACHELOR OF INFORMATION TECHNOLOGY (INFORMATION SYSTEMS) (HONS.)

COLLEGE OF COMPUTING AND INFORMATICS

UNIVERSITI TENAGA NASIONAL

2025

DECLARATION

I hereby declare that this final year project is my original work except for quotations and citations have been duly acknowledged. I also declare that it has not been previously and is not concurrently submitted for any degree program at Universiti Tenaga Nasional or at any other institutions. This final year project may be made available within the university library and may be borrowed, consulted, copied or reproduced in accordance with the provision of the UNITEN Library Regulations from time to time made by the Library Committee.

.....

Name: Ayman Fikry bin Asmajuda

Student ID: IS01081779

Date: 3 February 2025

APPROVAL PAGE

TITLE: ADVANCEMENTS IN MONITORING & FORECASTING FUTURE COVID-19 OUTBREAKS USING PROPHET AND HOLT'S WINTER MODEL

AUTHOR: AYMAN FIKRY BIN ASMAJUDA

The undersigned certify that the above candidate has fulfilled the condition of the Final Year Project in partial fulfilment for the Bachelor of Information Technology (Information Systems) (Hons.)

SUPERVISOR:

Signature: ……..

Name: Ms. Nur Laila Bte Ab Ghani

Date: 3 February 2025

EXECUTIVE SUMMARY

A widespread of disease outbreak in various countries from all over the globe would eventually lead to a global pandemic such as the recent COVID-19 pandemic which affected countless lives worldwide. Malaysia was one of the few countries that was severely affected by the epidemic at first, but over the years, cases have slowly declined due to the necessary efforts and attempts made by the Malaysian Ministry of Health (MOH) to fully control the transmission of COVID-19. Despite the descend of COVID-19 transmission rates, sudden COVID-19 outbreaks still emerge from time to time in many parts of Malaysia. The difficulty to appropriately adapt in these sudden outbreaks, lack of predictive capabilities in the current dashboard system and understanding the various circumstances of the project that may affect the underlying results are some of the problems identified for the project. All in all, the main purpose of this project is to monitor the daily progression of COVID-19 and predict sudden COVID-19 outbreaks throughout the daily COVID-19 cases in Malaysia.

The project will proceed accordingly based on the Cross Industry Standard Process for Data Mining methodology, or some might call as the CRISP-DM methodology. Only four of the six phases of CRISP-DM will be implemented as to accommodate with the project timeline, which includes the business understanding phase, data understanding phase, data preparation phase and finally the modelling phase.

Through the successful development of a COVID-19 dashboard that is equipped with the best predictive analytics model in its arsenal, Malaysian health officials would be able to predict COVID-19 outbreaks in a much more informed and efficient manner through the latest COVID-19 cases. This would further demonstrate on the potential growth and progression of the COVID-19 dashboard in Malaysia to maintain public awareness on the trends of COVID-19 that is happening in Malaysia. As a result, this visualization tool would be beneficial in identifying the best course of action to overcome the emergence of sudden COVID-19 outbreaks while also uncovering the hidden patterns and trends that may lead to the sudden outbreaks.

TABLE OF CONTENTS

LIST OF TABLES

Table No Page

LIST OF FIGURES

LIST OF ABBREVIATIONS/GLOSSARY OF TERMS

COVID-19	Coronavirus Virus Disease 2019
MOH	Ministry of Health
CRISP-DM	Cross-Industry Standard Process For Data Mining
WHO	World Health Organization
SARS-CoV-2	Severe Acute Respiratory Syndrome Coronavirus 2
PHEIC	Public Health Emergency Of International Concern
PUI	Person Under Investigation
ML	Machine Learning
STD	Sexual Transmitted Disease
KCDC	Korea's Center For Disease Control
MAPE	Mean Absolute Percentage Error
LR	Linear Regression
HW	Holt's Winter
MAD	Mean Absolute Deviation
PR	Polynomial Regression
SVM	Support Vector Machine
MLP	Multilayer Perceptron
PMP	Polynomial Multilayer Perceptron
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
LSTM	Long Short-Term Memory network
RF	Random Forest

ARIMA	Autoregressive Integrated Moving Average
GBR	Gradient Boosting Regression
KNN	K-Nearest Neighbour
DT	Decision Tree
SARIMA	Seasonal Autoregressive Integrated Moving Average
API	Application Programming Interface
EWMA	Exponentially Weighted Moving Average
FB	Facebook
EDA	Exploratory Data Analysis
CSV	Comma-Separated Values
MCO	Movement Control Order
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
R^2	R Squared

CHAPTER 1

INTRODUCTION

1.0 Overview

Chapter 1 introduces to the project with six main subtopics which consists of project background, problem statement, objectives, project scope, expected outcome of the project and the project timeline.

1.1 Project Background

1.1.1 Introduction to Outbreaks

According to Houlihan (2019), an outbreak can refer to a situation when there are more disease-related cases than expected in a particular location for an extended period of time. In other words, outbreaks usually lead to an increased number of cases for quite some time from a particular kind of disease or illness. It may come in a variety of forms, from minor instances such as small clusters to something much more serious such as a global pandemic that spread the disease in an extensive manner to multiple nations and countries worldwide. These outbreaks differ in terms of its source of origin. The World Health Organization (n.d.) stated that infectious disease that are transmitted from close contact with another person, animal or insects could contribute to the rise of outbreaks. Generally, actions made by humans would almost often play a part in promoting the transmission of such diseases. In a way, it is safe to assume that the growth of outbreaks are highly dependent on a specific catalyst which could make matters worse if left unresolved without any contingency plans. Not to mention that outbreaks are also unpredictable as it could occur to anyone, anytime and anywhere. This means that one could not simply expect the unexpected since outbreaks may come and go whenever it pleases. One type of outbreak that made headline news all over the world and considered to be a worldwide historical event is the COVID-19 pandemic.

1.1.2 COVID-19 Origin

COVID-19 is known throughout the entire world as one of the deadliest and life-threatening diseases out there due to its exponential growth leading to a global pandemic which affected numerous people from all walks of life. The term COVID-19 derives from the Coronavirus Disease that started in the year 2019, as announced by the World Health Organization (2020). Cennimo (2024) defined COVID-19 as a highly infectious and contagious disease that comes from the novel coronavirus, SARS-CoV-2, which is an acronym for severe acute respiratory syndrome coronavirus 2. The renown disease was initially discovered in Wuhan City of Hubei Province, Central China on 31st December 2019. The first cases of COVID-19 were linked to Wuhan city's Huanan seafood market since there have been rumors on social media about the wild exotic animals being sold there which include bats, snakes, pangolins and many more. Through multiple investigations, it was assumed that the virus was originated from bats as there were high similarities, but it is still unknown on which intermediate species that enabled the virus to spread to humans (Singhal, 2020). Since then, the COVID-19 disease has spread throughout numerous countries and nations worldwide with many identified cases and fatalities that comes with it. It is to the point that the disease has been officially declared as a public health emergency of international concern (PHEIC) by WHO for the dangers it could pose to the whole world (Li et al., 2020).

1.1.3 COVID-19 Symptoms

COVID-19 could infect practically anyone from all ages, from people who are incredibly ill to healthy individuals. It is considered to be highly contagious as it could spread easily through droplets and particles from an infected person when they either breathe, talk, or sneeze (CDC, 2023). An unaware individual might end up breathing in these particles without noticing it which could cause them to be infected as well. Sometimes these infected droplets or particles might also contaminate surfaces which could boost the risk of someone being infected by the virus when they accidentally touch these surfaces and immediately put their hands on their face. Those who are exposed to the virus might suffer from mild to severe symptoms such as fatigue, headaches, difficulty breathing, high fevers and other possible symptoms. In worst cases, it may even lead to one’s untimely and unforeseen death. People

with disabilities or underlying illnesses might be more prone to these kinds of symptoms compared to others (CDC, 2024). There are also instances where people who are infected with COVID-19 that do not show any signs of weaknesses or symptomatic problems as their immune system might developed the necessary countermeasure to fight against the infectious disease. Painter (2023) reported that in some cases, there are also several people who have been reinfected by COVID-19 after just recovered from the illness, though the symptoms are usually mild and only in rare circumstances that it will become much severe. Nonetheless, the symptoms can differ from person to person and as time goes on, new variants emerged with newer symptoms which might lead to the sudden COVID-19 outbreaks especially in developing countries like Malaysia.

1.1.4 COVID-19 Transmission in Malaysia

Initially, it was reported by Hashim et al. (2021) that the first transmitted COVID-19 disease in Malaysia was on 25th January 2020, whereby the disease was contracted by three individuals. The three individuals were identified through screening and monitoring upon receiving news from the Singapore Ministry of Health of 8 people that had close contact with confirmed cases individuals went to Johor, Malaysia. These three individual cases can be considered as imported cases which means that their illness was obtained from outside of Malaysia. The individuals who were infected had to undergo containment and monitoring procedures to avoid the disease from spreading to the citizens of Malaysia. Up until 15th February, the cases slightly increased around 22, that had accumulated from 12 Person Under Investigation (PUI), 8 that had close contact with people of confirmed cases and the other two were Malaysians that got evacuated as part of a humanitarian support mission from Wuhan, China. After a while, the first wave of individuals recovered successfully with no new cases appearing for the next 11 days. Not long after that, on 27th February, comes a second wave of outbreak that grew the total number of new cases due to a few people that had close contact with others when travelling internationally for various meetings and events. Moving on to 10th March, the total number of cases increased to 129. The 129 cases were related to Person Under Investigation (PUI), people of close contact with those infected, and the humanitarian support mission evacuees. Then, came the Sri Petaling Tabligh incident which resulted in a widespread of infection throughout all states in Malaysia.

1.1.5 Tabligh Gathering Incident

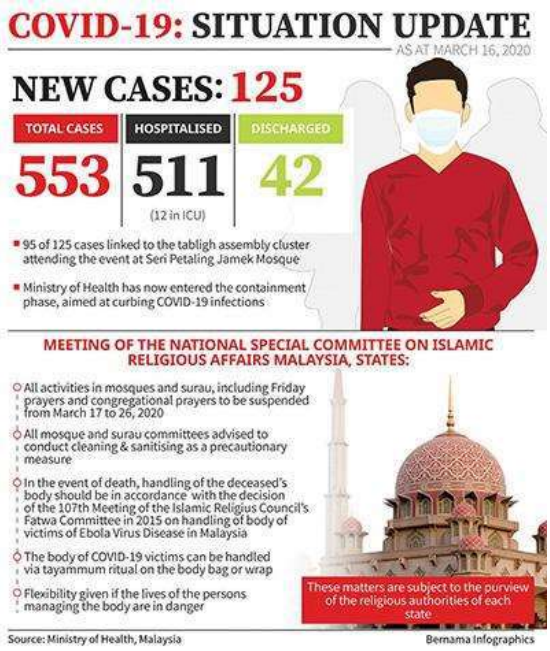

Figure I - Situation Update on COVID-19 On 16th March 2020

The figure I above shows the updated situation of COVID-19 on 16th March 2020 after the tabligh gathering incident that was held at Sri Petaling Jamek Mosque. At the time, it was known as the largest COVID-19 outbreak infection in Malaysia. Due to the abovementioned incident, many locals across the states of Malaysia had to go tested for risk assessment. During the tabligh gathering at the Sri Petaling Mosque, over 16, 000 people from various countries attended the gathering which might of cause a few of the locals to came in close contact with someone infected (Kaur, 2020). To summarize, there were a total of 553 cases recorded. 125 of the cases were considered newer ones with 95 of the cases related with the tabligh gathering. Among the 553 cases, a whopping 511 people were hospitalized while 12 had to undergo Intensive Care Unit (ICU). Fortunately, 42 of them were able to be discharged. To mitigate the spread of COVID-19 from becoming worse, the Malaysian Ministry of Health (MOH) declared Malaysia to be entering in the final late containment phase by commencing the Malaysian Movement Control Order (MCO). This ensures that the act of social distancing will be maintained throughout the daily lives of all Malaysians. According to (Hashim et al., 2021), daily cases began to surge up to 900 confirmed cases by 19th March, ranking Malaysia as the 4th country with the highest number of cases in Asia while also becoming the first in Southeast Asia.

1.1.6 COVID-19 Post Pandemic in Malaysia

The fluctuating numbers of COVID-19 outbreaks in Malaysia started to decline over the course of time. With the implementation of several health and safety regulations such as the public adherence to lockdowns, compliance with the act of social distancing and wearing masks, successful vaccinations initiatives made by health officials, cases related to COVID-19 significantly declined along with the trend of the disease in all parts of the world. Likewise, the mortality rate as well as the frequency of hospital admissions also drastically reduced as many people have developed an immunity for such disease, hence the disease could not reproduce and transmit to people that easily anymore. Even though the widespread threat of COVID-19 is gradually fading, there are still seasonal factors that must be taken into consideration as cases could spike up in certain instances throughout the year. As a result of these seasonal patterns and trends, sudden COVID-19 outbreaks could still be appearing at any point of time in Malaysia.

1.1.7 Recent COVID-19 Outbreak in Malaysia

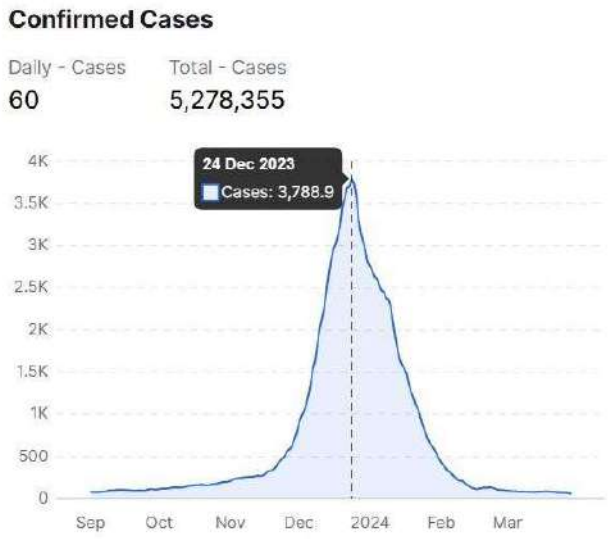

Figure II - COVID-19 Confirmed Cases Statistics from September 2023 to March 2024

Figure II shows the number of confirmed cases in Malaysia from September 2023 till March 2024. As indicated in the figure above, the number of confirmed cases suddenly escalated to almost 3800 confirmed cases on 24th December 2023. Though, after a while the frequency continue to decline into almost nothing. This shows the seasonality factor in play which causes the short term outbreak lasting during the Christmas holidays as most Malaysians would travel abroad while at the same time foreigners would also visit Malaysia in this time of year for a way to spend their holidays. Hence, sudden outbreaks may still happen from time to time depending on the time year and appropriate measures should be taken in preparing for unforeseen COVID-19 outbreaks in Malaysia.

1.2 Problem Statements

One of the main issues is that various countries from all over the world including Malaysia have difficulties in adapting to the sudden COVID-19 outbreaks. COVID-19 outbreaks may occur in the most unlikely and unexpected times. It is quite hard to truly determine the main causes of these outbreaks as it revolves around various factors that may directly or indirectly affect the rate of COVID-19 outbreaks. Oftentimes, countries would devise countermeasures for the COVID-19 outbreaks only when the outbreaks have spiked up out of the blue which by then, its already too late. Valuable healthcare and medical resources are wasted at unnecessary time and places whilst others have limited resources when they are needed the most. Not to mention, businesses and corporations suffer from financial loss and bankrupt as economic crisis comes abrupt. This clearly shows the need of an appropriate modeling approach to predict COVID-19 outbreaks before it comes to worse, in hopes to be fully prepared in the likelihood of an outbreak.

Besides that, the absence of predictive analytics makes it much more challenging to predict these sudden COVID-19 outbreaks. There have been many attempts made by several individuals and organizations to effectively track and monitor the state of affairs regarding COVID-19 all around Malaysia through the use of data visualization tools namely dashboards. On a similar note, the Malaysian Ministry Of Health(MOH) has also introduced a dashboard called COVIDNOW, that gives a summary on the daily progression of COVID-19 pandemic

in Malaysia. Despite the overwhelming number of dashboards, none of them seem to implement a predictive analytics model which could be significant in forecasting for potential COVID-19 outbreaks. As it stands now, Malaysia's current COVID-19 dashboards lacks the predictive capabilities to plan against unforeseen outbreaks. This could be detrimental to the country's welfare which limits the potential for growth in gaining resourceful insights to resolve the COVID-19 outbreaks throughout Malaysia.

Furthermore, considering the accuracy, precision, and performance in predicting these COVID-19 outbreaks, the results may vary from one predictive model to another due to the various circumstances throughout the project. Depending on the specific goals and objectives of the project, geographical locations, complexity of the data, features selection, scalability of forecasting and many other aspects, the output will without doubt differ from each other which may result in unsatisfactory and bias judgements, predictions and justifications. For that reason, it is best to test out with different approaches and machine learning algorithms to compare the end results accordingly so a suitable predictive model may be identified for the project.

1.3 Project Objectives

The main aim of this project is to predict COVID-19 outbreaks through the surrounding cases in Malaysia using the ideal machine learning model in hopes to assist health officials and government to plan against future COVID-19 outbreaks. The objectives of the project are as follows:

a) To determine the ideal Machine Learning(ML) Model that could predict future COVID-19 outbreaks
b) To develop a predictive analytics model using the identified Machine Learning(ML) methods to predict future COVID-19 outbreaks effectively and accurately
c) To create a dashboard that could visualize the current COVID-19 trends and patterns to gain informative insights

1.4 Project Scopes

The scope of this project revolves around the COVID-19 cases all over Malaysia. An open data catalogue will be utilized from the Ministry Of Health (MoH) Malaysia website. The aforementioned website is known as one of Malaysia's official data portals that stores and procures latest data that contains various COVID-19 information in Malaysia. Two datasets will be used for this project which comprised of historical and current data related to COVID-19 cases in Malaysia. They consist of daily COVID-19 cases from all over the 13 states and 3 federal territories in Malaysia that covers the span of 5 years from the year 2020 up until the present time. Other than the different states, one of the datasets classified the cases into specific age groups that ranged from age 0 to 80 years old and above, whereas the other one classified the cases into various types. These datasets include several key features namely numerical features, temporal feature, and categorical feature. Numerical features represent the different types of cases such as new cases, imported, cases, recovered cases, active cases, and cluster cases and also the cases related to specific age groups. While temporal features refer to the specific dates by days of such cases. The categorical feature, on the other hand, emphasize on the 13 states, 3 federal territories and Malaysia as a country. As of 1st February 2025 the number of rows for both datasets are approximately 31,000 rows respectively, indicating a low-dimensional data structure.

1.5 Expected Outcomes

The expected outcome of the project is the development of a data visualization dashboard that monitors the current state of the COVID-19 cases in Malaysia which will present the information in visual graphs that are both comprehensive and interactive. Additionally, a predictive model will also be implemented to predict and forecasts future COVID-19 outbreaks through the records of daily COVID-19 cases with ideal precision, accuracy, and performance. Essentially, it will aid the Malaysian Ministry Of Health (MOH) and health institutions all over Malaysia to make informed decisions from the patterns and insights gathered in hopes to plan against potential COVID-19 outbreaks.

1.6 Project Timeline

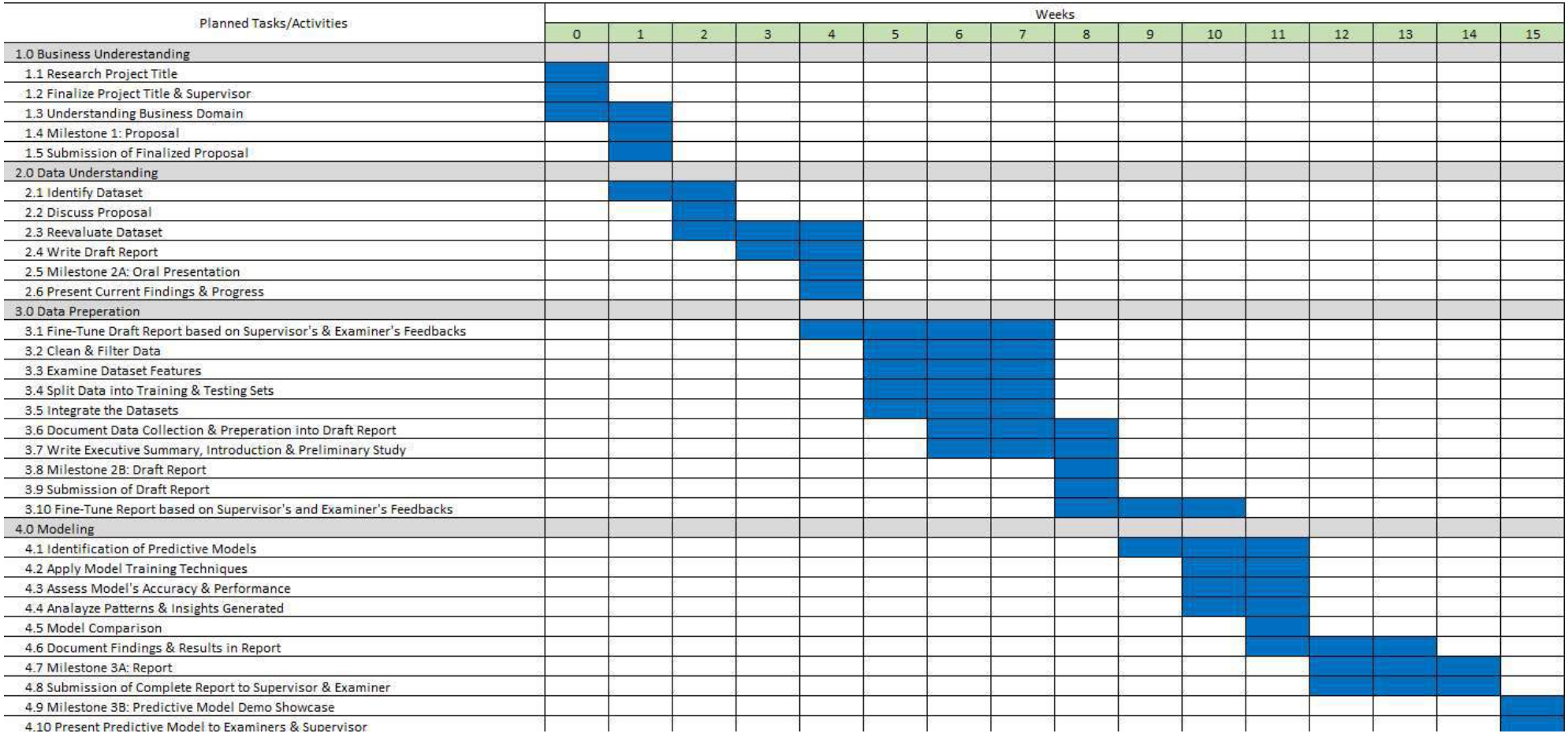

Figure III - Project Timeline in accordance with CRISP-DM Methodology

1.6.1 Project Timeline Explanation

The project timeline above presents on the planned tasks and activities which will be done during the 15-week period starting from week 0 till week 15. The project timeline includes the first four phases of the CRISP-DM methodology as the four main stages of the project which are business understanding, data understanding, data preparation and last but not least, modeling phase. The four main stages can be summarized as below:

Main Stages	Brief Explanation
1.0 Business Understanding	The initial stage whereby tasks mostly focus on understanding business domain, topic of research, nature and scope of the project
2.0 Data Understanding	The second stage whereby tasks are related to finding, evaluating, understanding and proposing datasets with supervisor and examiner
3.0 Data Preparation	The third stage whereby the identified dataset will be prepared accordingly to be utilized for next coming, modeling stage
4.0 Modeling	The fourth stage whereby the selected ML models will be created, tested, analyzed, compared for the ideal predictive model selection

Table 1 - The First Four stages of CRISP-DM Methodology

Furthermore, the project timeline also consists of significant dates namely milestones which helps to track and evaluate the progress of the students throughout Project 1. The important milestones are as follows:

Milestones	Week
Milestone 1: Proposal	Week 1
Milestone 2A: Oral Presentation	Week 4
Milestone 2B: Draft Report	Week 7
Milestone 3A: Report Submission	Week 12
Milestone 3B: Dashboard/Predictive Model Demo	Week 15

Table 2 - Project Milestones Timeline

1.7 Chapter Summary

In summary, Chapter 1 sets forth as the foundation for all of the following chapters ahead. It provides the general concept and idea related to the project. First off, the background of the project elaborated on the definition of outbreaks, the origins of COVID-19, the COVID-19 disease transmission, Malaysia's COVID-19 historical timeline, the peak of COVID-19 outbreaks in Malaysia and finally concluded on the recent sudden COVID-19 outbreak in Malaysia due to seasonality. Along the way, the related problems are identified which includes the challenge in adapting to sudden COVID-19 outbreaks, absence of a predictive analytics in the context of Malaysia and the varied circumstances and factors of the project which might affect the accuracy, precision and performance in forecasting outbreaks. Moving on to the project objectives whereby the aim and objectives of the project are mentioned as to set the goals and standards that are needed to be achieved for the project. Next is the scope of the project that is focus on entirely in Malaysia and the use of an open dataset from the Malaysian Ministry of Health (MOH) website. After that, is the expected outcomes of the project which is the successful development of a dashboard with predictive analytics capabilities in hopes to assist in the health and safety movement in Malaysia. Last but not least, shows the project timeline which is expected to be followed accordingly to the predetermined schedule. The following chapter 2 will discuss more on research evaluation and analysis to help guide the course of the project.

CHAPTER 2

PRELIMINARY STUDY

2.0 Overview

Chapter 2 discusses on the literature review of past projects and research which covers four three key subtopics comprising of an introduction to health disease, review of related works and also the review of machine learning methods.

2.1 Introduction To Health Disease

An infected person is a term to define someone who has contract an infectious disease within them while also capable of spreading their disease to other people. As stated by Mayo Clinic (2024), an infection occurres when harmful organisms that might cause diseases goes inside the human body and starts to reproduce to increase their numbers. This in turn would trigger the immune system to fight back against these infections by producing antibodies and specific cells to eradicate the infection from the body for good. Failure to do so would result in the growth of a disease as the cells in the body are heavily damaged and eventually lead to symptoms that may vary in its severity. Diseases as a whole, is a well-known topic and most people no strangers to it. A disease could be categorized into two different types which are infectious disease and non-infectious disease. Some might also refer to them as communicable and non-communicable disease respectively.

2.1.1 Infectious vs Non-Infectious Disease

Infectious diseases derive from the fact that the type of disease or illness is contagious and can be transmitted from external conditions into our human bodies. According to Mayo Clinic (2022), methods of transmission may vary as some may contract the disease through other people or animals, while others may be infected due to either eating or drinking contaminants in their food or beverage. Not to mention, infectious disease often causes a diverse range of signs or symptoms ranging from bad to worst conditions. Whereas non-infectious diseases indicate that it does not rely on foreign organisms as such it could it not be transmitted by other living things (Cleveland Clinic, 2022). One could say that the main difference between them is the presence of pathogens and deadly organisms such as bacteria, viruses, fungi etc. This is due to the fact that infectious diseases are highly dependent on these pathogens to effectively spread from person to person, whereas non-infectious disease do not depend on these pathogens but instead are caused by various other factors and circumstances (ChildFund Australia, 2024). The COVID-19 disease can be classified as an infectious disease as it is highly contagious and also originate from the novel coronavirus, SARS-CoV-2. On the other hand, non-infectious disease is mainly caused by genetical factors, surrounding environment, physiological factors and lifestyle behaviors as reported by the World Health Organization (2023). Both infectious and non-infectious diseases are known to cause many problems and issues towards the health and welfare of people from all walks of life, with some even leading to unfortunate deaths. Before the COVID-19 pandemic, non-infectious diseases or non-communicable diseases were one of the most common causes of death in comparison to infectious diseases that were initially the leading causes of death in the past (Dattani et al., 2019).

2.1.2 Leading Causes of Death

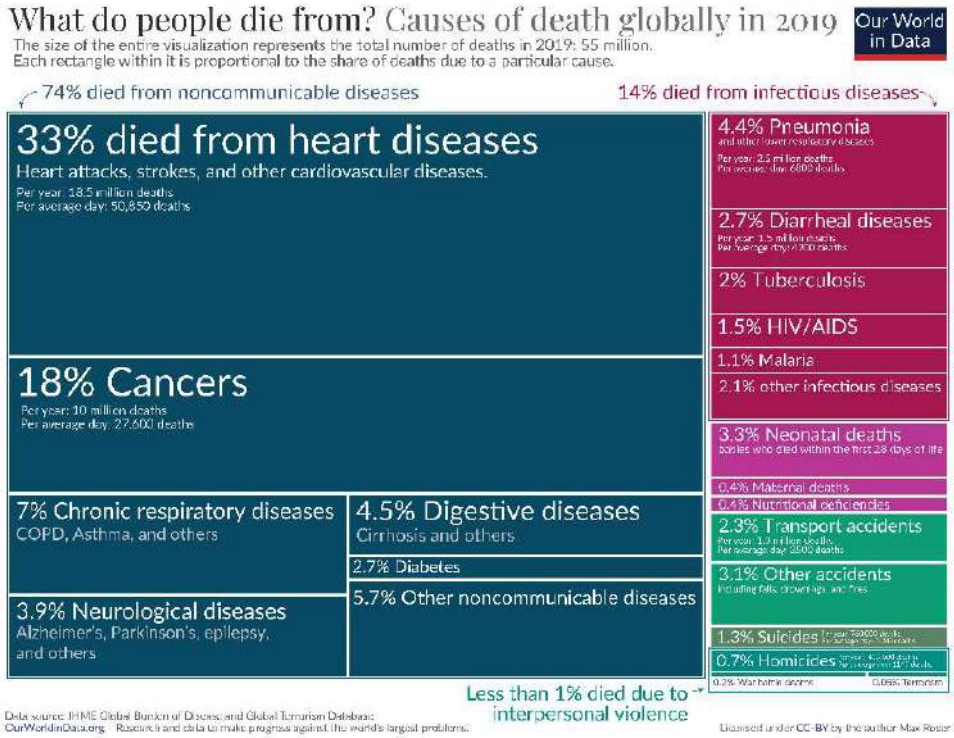

Figure IV - Causes of Deaths Visualization in 2019 before Covid-19 Pandemic

The figure IV above presents on the past information on the global causes of death in the year 2019. As a whole, there are approximately 55 million total number of deaths that is caused by the sum of both non-communicable or non-infectious diseases and infectious diseases. Non-communicable diseases comprise of a whopping value of 74% of the total deaths, whereas only 14% died from infectious diseases. Heart attacks, strokes, and other cardiovascular disease represents the classification of heart diseases with the highest percentage amongst the non-communicable diseases which is 33%. On the other hand, pneumonia and other lower respiratory diseases resulted in 4.4% of the total 14% which is the highest for infectious diseases. This clearly indicates that before the era of COVID-19 originated, non-communicable diseases were much more serious when compared to infectious diseases as infectious diseases were not much of a threat since various countries and nations worldwide could contain and control them without much trouble.

2.1.3 COVID-19 Pandemic

With the birth of COVID-19, a new global pandemic arises and statistically wise the infectious disease has taken the world by storm, changing the tide of the current causes of death for years to come. As COVID-19 cases began to arise one after another, it opens the door for other infectious diseases to take charge as well since the strains of COVID-19 left a huge impact in the healthcare system, limiting the required resources, services and surveillance for other diseases. Though to put in another perspective, the growth of COVID-19 also sheds a new light towards a healthier and proactive lifestyle that encourages people to maintain social distancing, wear masks, regularly exercise and have frequent health checkups which in the long term will lessen the transmission rates of other infectious diseases in the process. That being said, infectious diseases are a much more concerning matter compared to non-infectious disease due to its nature of causing sudden outbreaks which could lead to a widespread pandemic that may wreak havoc to the daily lives of many people, businesses, societies and even countries around the world.

2.1.4 Factors Affecting Disease Outbreaks

Disease outbreaks are somewhat of a mystery because of its tendency to either pop up out of nowhere or just disappear without a trace. This shows that there are various factors and circumstances that come into play that might affect the severity of the outbreaks that emerged. Identifying the particular factors may be the key to unveil the underlying truths behind the outbreaks on its potential occurrences, to the extent of predicting whether the outbreaks will result in either an increase or decrease number of cases over time. Through extensive research, various sources have indicated that health preventive measures and initiatives, the country's stage of development, human behaviors and seasonality are a few of the major factors that may affect the rate for future potential outbreaks. The aforementioned factors are explained further below:

2.1.5 Health Preventive Measures and Initiatives

First and foremost, concerning health preventive measures and initiatives, compliance with appropriate acts of conduct such as a movement control order, that limits people's movement from going places to places or maintaining social distancing, which will prevent close contact with infected individuals, would contribute towards mitigating the risk of potential outbreaks. Even simple actions like consistently wearing masks, washing one's hand, maintaining good hygiene or regular health screening could have huge impact to the health and welfare of a society. In addition, countries that advocate for the use of vaccinations as a disease preventive measure often fare well during widespread outbreaks. Vaccines aids in producing antigens which will be crucial in protecting and safeguarding our immune system against unknown diseases (World Heath Organization, 2020). In most countries, government and health officials would offer two or three-dose vaccination campaigns to their citizens, regardless of their background, age or gender, in hopes to control the spread of disease from causing a massive outbreak within the population. A study that was conducted by Moghadas et al. (2020) on the impact of vaccines in regards to COVID-19 outbreaks in the United States, shows that there is an overall reduction in attack rate in populations that were vaccinated. Even though, there were a few people that could not be vaccinated due to specific health reasons and complications, the disease could be mitigated as the community as a whole were immunized by the COVID-19 outbreak. This is known as herd immunity as both vaccinated and non-vaccinated parties, live amongst one another which makes it much more difficult for the disease to spread even to the non-vaccinated since they are protected by those around them who have the immunity to fight against the disease. Therefore, proper health preventive measures and initiatives will without doubt be a significant factor in reducing the transmission of disease and decrease the likelihood of potential outbreaks from occurring.

2.1.6 Country's Stage of Development

Moving on, is regarding the country's stage of development which consists of three main categories such as developed countries, developing countries and least developed countries. Depending on the distinct development stage, the spread of diseases may vary, with some are more in control of their dire situation while others suffer catastrophic loses due to their inadequate resources and infrastructures. ChildFund Australia (2024) stated that the outbreaks are much more severe when there are insufficient infrastructures. This is due to the fact that countries that are in either the developing or least developing stage could not cope with the abundance number of cases and provide optimal services which would result in drastic outbreaks that may take many innocent lives in the process. On a similar note, Helou et al. (2021) reported on the dreadful living situation in Lebanon, that comprise of unsanitized water, detrimental health conditions and low herd immunity among the local and migrated population which resulted in the trasmission of multiple diseases. This clearly shows the state of least developed countries such as Lebanon that had to succumb to their current way of living in dealing with outbreaks without any sort of help. On the contrary, developed or better developing countries could stand their ground in a much more favourable situation against the outbreaks as they have the necessary monetary resources, health care services, state-of-the-art tecnhological assistance and many more in dealing with potential outbreaks. By inference, countries that are not quite yet developed, tend to operate badly in confronting outbreaks when compared to countries that are in a higher development stage. As such, every country, be it developed, developing, or least developed countries should give a hand to one another in providing the necessities to countries that are in the lower spectrum so the risk of widespread outbreaks could be reduced slowly but surely.

2.1.7 Human Behaviors

Besides that, human behaviors also play a vital role in affecting the course of outbreaks. Everyone in this world lives in a way that is different than others especially in terms of one's own lifestyle. Our daily lifestyles are shaped in a variety of manner from how we are taught

and educated by our parents since childhood to the memorable experiences and environments that we grew up that define us today. It comes to no surprise that those who live in terrible environments or have poor upbringing may often act in peculiar ways and carry out bad habits such as drinking alcohol, taking drugs and performing other horrendous acts. These habits would heighten the risk of someone to be infected by infectious diseases since their bad lifestyle, will consequently lead to a detrimental state of health and wellbeing. To put in a perspective, if for instance not just one but many people led a similar kind of lifestyle, this could allow the pathogens to easily spread from people to people due to their weak immune system, thereby contribute to the growth of outbreaks in the long run. According to Church (2004), humans naturally have the urge to practice in sexual behaviors particularly in countries that would consider the abovementioned behavior as a norm such as the acceptance of prostitution, hence promoting the transmission of Sexual Transmitted Diseases (STD) among the population. As a result, cases of STDs would surge surrounding the public, thus affecting the widespread of outbreaks to occur at an irregular rate.

2.1.8 Seasonality

The final factor to discuss on is regarding seasonality factors that may steer the direction on the rate of sudden, unexpected outbreaks. Seasonality refers to specific patterns or similar occurrences that may happen within a time period. These patterns may exhibit daily, weekly, monthly or even yearly depending on the range of time intervals. In the case of identifying outbreaks, it is best to analyze them annually since there will be a longer period of time to observe and analyze any insights or trends that can be gathered throughout the years, with many cases related to infectious disease tend to appear in particular seasons of the year. Nath et al. (2021) conducted a research on how the roles of extreme climate change have an effect towards the COVID-19 outbreaks around the world. In summary, Nath et al. mentioned that due to the COVID-19 adaptability in diverse weather condition and various temperatures, daily cases fluctuates from time to time and still remains active. The disease speedy adaptibility allows it to even survive in extreme weather conditions such as cold temperatures below -15^{o} celcius and warm temperatures up to 25^{o} celcius. This means that the COVID-19 transmission rate would be just as effective in countries as cold as Russia or countries with high humidity such

as Brazil. Alternatively, apart from the effects of climate change in regards to the seasonality of outbreaks, another research that carried out by Quinn, et al., (2024), shows results of seasonal patterns on specific times of the year aligning with COVID-19 outbreaks in Denmark as shown on Figure V below.

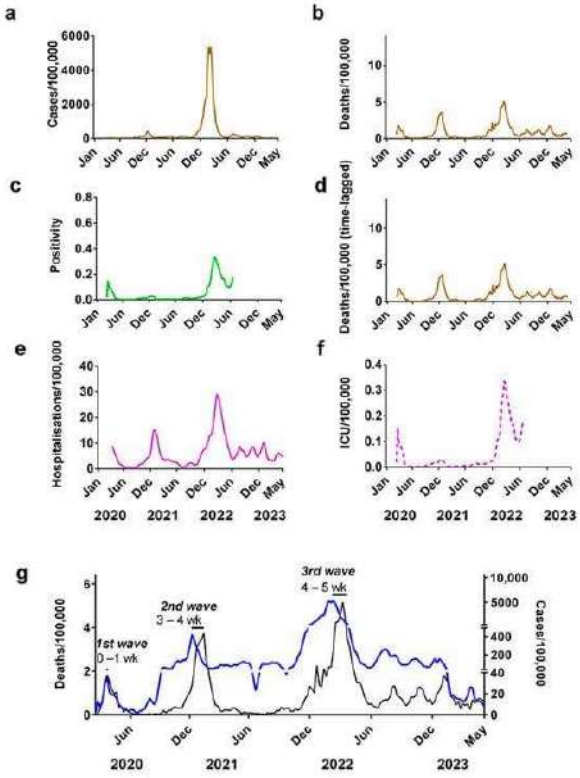

Figure V - Graph Representation of the COVID-19 pandemic in Denmark

The figure V above shows a few graphs that represents cases, positivity, hospitalizations and deaths per 100,00 people related to COVID-19 outbreaks from the year 2020 to 2023 in Denmark. Whenever, it reached the end of the year namely during the holidays, the results suddenly skyrocketed which indicate the emergence of COVID-19 outbreaks. As shown in the graph representation of deaths per 100,000 people, there are 3 waves of outbreak that occurred in Denmark during the fixed time period. It will start to spike up when December has arrived and then gradually decreases up until June. This may because of international travels as many people especially foreigners will travel to various countries to celebrate the holidays. As a result, it will increase the risk of close contact with those who are infected which indicate the sudden outbreak that occurred according to the seasonal patterns.

2.2 Review Of Related Work

A total of six journals have been identified to be reviewed and summarized based on their key features and unique findings. Each of them differs in their chosen titles, specific aims and objectives, project scope, data sources gathered, machine learning models selected and findings. On top of that, a few of the journals also have their own notable similarities and differences which further highlights on the diversity of the creative works.

A study was conducted by Shahir Asfahan et al. (2020) to predict the cumulative numbers of COVID-19 cases in South Korea using a simple open source automated machine learning algorithm. The data was sourced from the Korea's center for disease control (KCDC). The PROPET model was the machine leanrning model that was utlized for their project. In the journal, the forecasts that was produced by PROPHET model were represented with a confidence level of 95% by range of upper and lower bounds for each of the predictions. Not to mention, the values that were forecasted were well-defined when compared with the actual numbers. It was mentioned that the predicted and observed values have a difference ranging from 4.08% to 12.77%. When the accuracy of the results were evaluated through a Mean Absolute Percentage Error (MAPE) metric, it was shown that for one week the MAPE index was 7.42% which means that the PROPHET model was a highly accurate forecasting model based on the journal. They also made remarks on how the model was given training data and managed to learn from it to automatically determine the changes in trend and from there produce estimations of the predictions which was a significant factor contributing to their success. Throughout the predictions, any sort of inaccuracies were gradually decreasing when it has reached the middle threshold but started to increase during the end of predictions set of time period. This is due to the fact that machine leanring models such as PROPHET model are consistently hungry and in need of data to continue performing accurate predictions for a longer time period.

Aside from that, another work is regarding a research that was carried out in Malaysia by Hasri et al. (2021), with the aim of predicting the daily cases of COVID-19 in Malaysia using Linear Regression and Holt's Winter model. The data source was obtained through the Malaysian Minitstry of Health (MOH). As mentioned above, the machine learning models that were used are Linear Regression and Holt's Winter model. Both models accuracy of prediction will be compared after predictive time period. As a result, both model generated really good performance overall, with Linear Regression approximately valued at 82% accuracy, while the Holt's Winter model was valued at 89% accuarcy. By going through the evaluation metrics of Mean Absolute Deviation (MAD) and Mean Absolute Percentage Error (MAPE) to determine the accuracy of the predictions of both model, it was concluded that Holt's Winter model outperforms the Linear Regression model as it has lower values in MAD and MAPE which indicated a better predictive model. Thus, making Holt's Winter model the best model out of the two because of its higher accuracy of predicting daily cases of COVID-19 in Malaysia. Based on the above work's findings, this shows the relevance and advantage of implementing Holt's Winter model for this project.

Furthermore, another journal primarily focuses on cases in Bangladesh and Worldwide that was experimented by Satu, et al. (2021). The dataset was acquired through the public Application Programming Interface(API) from github. There a total of six machine learning models that were implemented for forecasting in the project which includes Linear Regression, Polynomial Regression, Support Vector Machine, Multilayer Perceptron, Polynomial Multilayer Perceptron and the previous journal choice of model, the PROPHET model. The models were selected to analyze their capabilities for short term forecasting. The models' accuracy of prediction were also evaluated using Root Mean Square Error (RMSE), Mean Absolute Error(MAE) and R^2 Squared. The end results shows that when comparing the models in the context of Bangladesh as well as worldwide, the majority of the models produced larger error rates than others while only the PROPHET model produced the lowest rate and highest R^2 Squared value. In addition, Satu, et al. (2021) also added on that if there were various time intervals taken as influence or factors to the project, most of the time, PROPHET model will triumph over other models as it presented much accurate results of prediction in comparison to other models when forecasting infectious and fatality cases. This clearly indicates the competency and efficiency of using models such as the PROPHET model in predicting Covid-19 outbreaks.

Moving on to the next journal review, which utilize forecasting techniques using Long Short Term Memory (LSTM) and Deep learning model with the aim of predicting the number of COVID-19 cases in Canada using LSTM networks. The source of the dataset was collected from the Johns Hopkins University & Canadian Health authority. The results of the journal as reported by Chimmula et al. (2020) shows that in regards to short-term prediction, the Root Mean Squared Error (RMSE) values at 34.83 with accuracy of the model at a whopping value of 93%. Whereas, when applying the same concept for long term predictions, the RMSE resulted in 45.70 and accuracy of 92.67%. This means that LSTM and related deep learning models prove to be a significant predictive model in capturing the dynamics of transmission rate of COVID-19 in Canada because of its higher flexibility, adaptability, performance-wise, scalability and many other aspects compared with typical machine learning algorithms. Conventional statistical models, on the other hand, may have difficulties to choose the best parameters and variables which may lead to uncertainty (Chimmula et al., 2020). In comparison, LSTM networks deal with real time data and does not assume when selecting the hyperparamters, which indicates its superiority over statistical methods. However, in the case of the project, predicting sudden COVID-19 outbreaks in Malaysia, it can prove to be challenging due to the nature and complexities of the deep learning field. Without the necessary expertise and adequate knowledge in such field, the project would definitely end up as a failure.

Apart from that is another related work that was conducted by Kumar et al. (2021), on predicting total cases and death of COVID-19 in numerous countries which consist of the United States, India, Brazil and Russia using several machine learning methods. The dataset for the project was obtained through the World in Data by University of Oxford. Four models were selected for the perspective of statiscal modelling which comprise of Linear Regression, Random Forest, Autoregressive Integrated Moving Average (ARIMA) AND Long Short Term Memory Networks (LSTM). Findings of the journal shows that ARIMA provided the highest accuracy of prediction of test data when compared to the other three models. On the other hand, LSTM generated the highest accuracy of prediction when performing for death foreasting in comparison to the other models. This indicated that depending on the objective of either forecasting number of cases or predicting number of deaths, both ARIMA and LSTM establish their dominance in their respective categories over the other two models.

Ultimately, the final journal depicts a study on predicting COVID-19 trend in various countries using multiple different machine learning models while under the influence of numerous circumstances. The study was conducted by Saba et al. (2021), and the data source is from CSSEGISandData on Github. The models that were performed on are such as Random Forest, Polynomial Regression, Support Vector Regression, Gradient Boosting Regression, K-Nearest Neighbour, Decision Tree, Seasonal Autoregressive Integrated Moving Average (SARIMA), ARIMA and Holt's Winter. Despite using various ML models to perform the forecasting tasks, the results varied from countries to countries. This is because there are many underlying factors and circumstances in place such as the different type of lockdowns, the variety of evaluation metrics used to validate the accuracy of prediction for each model, the particular geographical locations, the diverse structure of each country and the specific forecasting tasks which is through either confirmed cases or deaths. This entails that one cannot assume which model is the best in cases where many factors and various circumstances are in play. Therefore, to determine which model is suitable for which specific set of criteria and requirements, it is best to experiment with various models to identify which of them will produce the highest accuracy, precision and performance in the particular prediction tasks

In summary, the main similarities between the journals are the aim to predict COVID-19 outbreaks through the number of COVID-19 cases in their respective countries of choice. Besides that, for every journal, the results of the model predictions were evaluated and validated through the use of evaluation metrics to assess their performance and accuracy of the predictions. On the other hand, the main differences between the journals are on the prediction goals whereby some of the journals would only forecast with a machine learning model or modelling technique while others compare the results of numerous models at the same time to analyze the pros and cons of each selected predictive model. Finally one of the notable differences is on the respective forecasting tasks of each journal because all of them have a different goal in mind that they would want to accomplish through their own distinct projects.

2.2.1 Summarized Table Of Literature Review

Citation & Title	Research Aim	Data Source	ML Model/Technique	Findings
Asfahan et al. (2020), Using a simple open-source automated machine learning algorithm to forecast COVID-19 spread: A modelling study	To predict the cumulative numbers of COVID 19 cases in South Korea using a simple open source automated machine learning algorithm	Korea's center for disease control (KCDC)	PROPHET Model	Well-defined predicted values when compared to actual values with difference ranging from 4.08% to 12.77%, MAPE index indicate 7.42%, hence Prophet model is highly accurate
Hasri et al. (2021), Linear Regression and Holt's Winter Algorithm in Forecasting Daily Coronavirus Disease 2019 Cases in Malaysia: Preliminary Study	To predict the daily cases of COVID 19 in Malaysia using Linear Regression and Holt's Winter model	Ministry of Health(MOH) Malaysia	Linear Regression & Holt's Winter	Holt's Winter model outperforms the Linear Regression model as it has lower values in MAD and MAPE index, though both Holt's Winter and Linear Regression produce good accuarcy results with value of 89% and 82% respectively
Satu et al. (2021), COVID-19: Update, Forecast and Assistant - An Interactive Web Portal to Provide Real-Time Information and Forecast COVID-19 Cases in Bangladesh	To predict COVID 19 cases in Bangladesh and Worldwide using multiple regression models	Public Application Programming Interface(API): https://github.com/ahmedsadman/covid19-bd	Linear Regression(LR), Polynomial Regression(PR), Support Vector Machine(SVM), Multilayer Perceptron(MLP), Polynomial Multilayer Perceptron(PMP) & PROPHET model	Prophet model produced the lowest error rate & highest R^2 Squared value which equals to 1, in comparison to the other utilized models which produced large error rates, thus Prophet model triumphs over other models when forecasting infectious and fatality cases

Chimmula et al. (2020), Time series forecasting of COVID-19 transmission in Canada using LSTM networks	To predict the number of COVID 19 cases in Canada using LSTM networks	Johns Hopkins University & Canadian Health authority	Long Short Term Memory (LSTM) & Deep Learning	LSTM presented high values in accuracy of prediction for both short-term and long-term prediction, hence is it highly capable for forecasting tasks in comparison to statistical models
Kumar et al. (2020), Predictive Analytics of COVID-19 Pandemic: Statistical Modelling Perspective	To predict the total cases and deaths of COVID 19 in the USA, India, Brazil and Russia using several machine learning methods	World in Data by University of Oxford	Linear Regression(LR), Random Forest(RF), Autoregressive Integrated Moving Average (ARIMA) & Long Short-term Memory networks (LSTMs)	ARIMA provided highest accuracy prediction of test data, whereas LSTM provide very high accuracy when performing in death forecasting task, indicating their suitability during prediction depending on the objectives
Saba et al. (2021), Machine learning techniques to detect and forecast the daily total COVID-19 infected and deaths cases under different lockdown types	To predict COVID-19 trend in selected countries using several machine learning models	CSSEGISandData : https://github.com /CSSEGISandDat a?tab=repositories	Random Forest(RF), Polynomial Regression(PR), Support Vector Regression(SVR), Gradient Boosting Regression(GBR), K-Nearest Neighbours(KNN), Decision Tree(DN), Seasonal Autoregressive Integrated Moving Average(SARIMA), Autoregressive Integrated Moving Average(ARIMA) & Holt's Winter	Most of the models provided the best accuracy depending on the specific country, features(confirmed cases or deaths) and lockdown types(partial, complete or herd), therefore it is impossible to choose the ideal predictive model due to the varied circumstances that may affect its accuracy of prediction

Table 3 - Summarized table for Review for Literature Review

2.3 Review of Machine Learning Methods

Based on the review of related works from past journals and research that aligns with the intention of predicting and forecasting COVID-19 outbreaks, it is clear to see that there are countless numbers of predictive analytics methodologies to choose from for the development of a comprehensive and informative COVID-19 dashboard with predictive capabilities. These includes the various types of machine learning algorithms, statistical models, and forecasting techniques that would be the ideal predictive model to achieve the goal, objectives and scope of the project. Four of the mentioned models in the review of related works section are chosen based on its relevance and suitability according to the project. The following highlights both the strengths and weaknesses of each approach and model in comparison to each other:

2.3.1 Autoregressive Integrated Moving Average (ARIMA)

ARIMA is the abbreviation for an Autoregressive Integrated Moving Average model that is primarily focuses on statistical analysis. It is suitable for time-series forecasting by utilizing the temporal features such as time series data in the dataset for future predictions. Basically, how this model works is by predicting through the past values to gain insights for better future predictive values. Hayes (2024) noted that the term ARIMA itself can be split into three distinct categories which are Autoregression (AR), Integrated (I) and Moving Average (MA).

Autoregression (AR) represents on how the model regresses its own past values to form future values for prediction. In other words, the variable's current value changes depending on its prior values. This indicates on the everchanging motion of the variable due to its historical values and from there, what cause the change in the variable could be identified. The following equation below represents the model in accordance with the parameter p:

$$m_t = 0 + 1m_t\text{-}1 + 2m_t\text{-}2 + 3m_t\text{-}3 + \ldots + pm_t\text{-}p$$

From the above equation, p refers to the lag order which determines the frequency of lag observations for the model. m_t refers to the observed value for the m variable as per the time which is t. As presented in the equation, mt is highly dependent on the historical values, with p periods that comes along with it. p can be understood as the specific weight that signifies for every past value for m, following the coefficients 0,1,2 etc. During the training of model, the coefficients are discovered for which the model analyzes the relationship between the current value which is m and its respective past values. Common techniques used are autocorrelation and partial autocorrelation as stated by Bajaj (2023).

Next is Integrated (I), which denotes on the differences between each observation and previous observation. Hayes (2024) mentioned that it can be called as the degree of differencing. This model component functions to make the data stationary, by finding out the any stationary attributes and then compare them with a number of diferencing factors which can be represented as the parameter d. Some of the stationary attributes includes the mean and variance which do not changes much over time.

Finally is the Moving Average (MA), which generally focuses on the correlation between an observation and residual errors through the Moving Average model that is performed on lagged and past observations. In other words, the MA model itself utilizes on not only the past values but also taking into consideration on the values of past errors during prediction (Bajaj, 2023). It can be represented in an equation with a parameter q as shown below:

$$\mathbf{m_t = 0 + 1e_t\text{-}1 + 2e_t\text{-}2 + 3e_t\text{-}3 + \ldots + qe_t\text{-}q}$$

The above equation is known as MA(q) model. The e_t can be considered as the errors made in past predictions, while q notes on periods from long ago which also acts as the coefficients for the respective weights of errors. Additionally, q can also be classifed as an order of moving average (Hayes, 2024). Basically, the main goal of this model is to search and gather

short-term fluctuations and random shocks to see how it affects the future values and from there, could be utilized to alter the future values accordingly.

Strengths

1. Ideal for time series analysis projects as it could seize on the temporal features and patterns for both linear and non-linear realtionships.
2. The model provides clear understanding on the specific components such as Autoregressive, Integrated and Moving Average models that do not require much effort to interpret.

Weaknesses

1. Incapable of working with seasonality factors as it is not recommneded for to take in such factors into consideration which may result in loss of crucial information due to the lack of context.
2. Despite the model's clear interpretability, at the end of the day it proves to be a challenge in selecting the ideal type of model from the range of selections to choose from due to the complex model orders.

2.3.2 Seasonal Autoregressive Integrated Moving Average (SARIMA)

The Seasonal Autoregressive Integrated Moving Average or what some would call SARIMA, works in an almost similar manner to the traditional ARIMA model but with the catch of seasonality coming into play. With seasonality being added to the order of the standard Autoregressive (AR), Integrated (I), Moving Average (MA) model order, the SARIMA model will become much more robust than usual (Bajaj, 2023). In addition to that, data that contains seasonal patterns

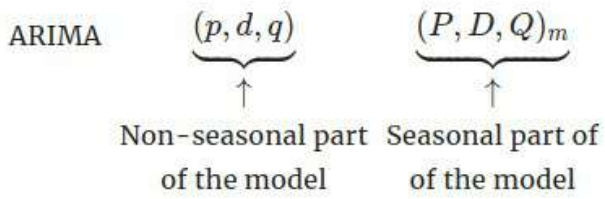

Figure VI - SARIMA model representation

The figure VI above presents on the SARIMA model The m above in Figure VI refers to the numbers of observation for every year. Uppercase notations will applied to the seasonal parts of the model, whereas lowercase notations uses the non-seasonal parts of the model.

Strengths

1. Capable of forecasting time series analysis with strong seasonal patterns and trends which makes it more flexible in understanding underlying patterns and gaining meaningful insights from the data.
2. Compared to the conventional ARIMA model, SARIMA could predict better in terms of higher accuracy, precision and performance wise due to differences of seasonality.
3. It incorporates ways of automation in selecting the ideal order for both seasonal and non-seasonal parts, allowing it to abandon the manual selection method.

Weaknesses

1. With the presence of seasonality, the model itself will become much more complex to handle as there are many more factors to consider when predicting.
2. Requires substantial computational resources as there are numerous seasonality components to consider which also leads to lengthy and time-consuming training times.
3. It would need a consistent and adequate amount of past historical data to predict with inclusion of seasonal patterns, thus not compatible with time series analysis projects that are shorter in its time frame.

2.3.3 Holt's Winter (HW)

The origin of this model comes from its creators Charles Holt and Peter Winters naming the model after them. It follows a Triple Exponential Smoothing technique that is a renowned method for time series prediction analysis with the presentation of trend and seasonality. The term exponential smoothing demonstrates the application of an exponentially weighted moving average(EWMA) to make the time series much smoother. It utilizes past values which will be used to make predictions on current and future values (SolarWinds, 2019). The triple exponential smoothing consists of three main components which are level (l_t), trend(b_t) and seasonality(s_t). Below gives more explanation on the respective components and the distinct exponential smoothing techniques equation:

$$s_t=\alpha x_t+(1-\alpha)s_t-1$$

The above equation represents a simple exponential smoothing technique that only accommodates for trend. Charles Holt decided to change the abovementioned equation to support linear trends as well. Therefore, the new equation would be named Holt's exponential smoothing.

$$st=\alpha xt+(1-\alpha)(st-1+bt-1)$$

$$bt=\beta(st-st-1)+(1-\beta)bt-1$$

The Holt's exponential smoothing introduces new terms such as levels and trends in the mix. Compared to simple exponential smoothing, this one involves two exponential weighted moving average (EWMA) which are x_t for the smoothed values and also the slope or trend(b_t). Another name for it is called double exponential smoothing. As a result, it could lead to better accuracy prediction and forecasts as it allows the model to take into consideration of the linear patterns and trends in the data. Afterwards, one of Holt's students, Peter Winters improvised the equation by also proposing a new factor called seasonality in the equation.

$$st=\alpha xt+(1-\alpha)(st-1+bt-1)$$

$$bt=\beta(st-st-1)+(1-\beta)bt-1$$

$$s_t=\gamma(x_t-l_t-p)+(1-\gamma)s_t-p$$

The equation above shows the current Holt's Winter model which revolves around 3 exponential smoothing naming it as the Triple Exponential Smoothing model. With the new formula, the model can forecast current or future values with the aspects of the levels, trends and seasonality as significant key factors to improve the accuracy of prediction. There are a few parameters involved in the Holt's Winter model such as the observed value (X_t), the current level (l_t), the current trend (b_t) and also the smoothing parameter (γ) which creates the designated formula for seasonal components (s_t). Other parameters include the t for the time, p for the seasonal period and the first two smoothing parameters (α and β). Hence, the model would operate effectively as intended in identifying seasonal fluctuations to forecast the time series data accordingly.

Strengths

1. The Holt's Winter model would be able to handle the complexities in the data as it relies on both the trend and seasonality making it ideal for predicting the time series data.
2. Adaptable in various circumstances such as managing seasonal fluctuations and outliers in the data due to its high flexibility.
3. High interpretability as it could forecast in the context of trends and seasonality as well, making the predictions easier to be understood by others.

Weaknesses

1. This model needs high computational resources as it is dealing with multiple seasonal patterns and trends which would be time consuming and intensive in regards to long term prediction.
2. It does not perform well with data that are constantly changing over time as well as insufficient historical data which might affect the accuracy of prediction.

2.3.4 PROPHET Model

The Prophet model or some would call FB Prophet is an open source tool that was developed by Facebook to effectively perform time series analysis using time series data and temporal features. Just like Holt's Winter, it involves both trend and seasonality, with the addition of the context of holidays in its predictions. Trend refers to when the data is more inclined to either increase or decrease over some time, while removing any seasonal variations. The seasonality attribute represents the recurring variations that might happen for a short, predetermined time period as explained by Rahulhegde (2024). He also mentioned that basically the PROPHET model is equivalent to a generalized additive model. The model's formula or equation can be seen from below:

$$y(t) = g(t) + s(t) + h(t) + e(t)$$

From the formula, g(t) represents the trend, s(t) represents the seasonality, h(t) represents the prediction value that is resulted from the involvement of holidays, e(t) represents the error term and lastly is y(t) which represents the forecast itself. Through the equation and the utilization of the PROPHET model, one could easily automate various calculations that is primarily mathematical.

Strengths

1. The model will perform exceptionally well in the context of holidays which would be convenient for users to make any changes towards crucial events in the data, thus allowing for higher accuracy of prediction.
2. It is highly adaptable in the sense that it could handle both linear and non-liner types of trends making it suitable for projects that deal with varied circumstances.
3. It is easy to use, understand and interpret the forecasts due to its user friendly features.

Weaknesses

1. It does not perform well in situations whereby the time series data have unpredictable and peculiar patterns as in complex non-linear trends residing in the data.
2. It lacks the capability to deal with external factors which may affect the prediction since it mainly prioritize on internal conditions rather than external ones during modelling.
3. Just like many other of the previously mentioned models, it requires huge computational resources for larger datasets which may take a long time to train the data.

2.3.5 Comparison Table Of Machine Learning Methods

Reviewed ML Models	Brief Explanation	Strengths	Weaknesses
Autoregressive Integrated Moving Average (ARIMA)	A statistical model that primarily focuses on time-series forecasting by utilizing past values to gain insights for better prediction values	1. Suitable for time-series analysis projects 2. High interpretability	1. Incapable of handling seasonality aspects 2. Challenging to decide on the ideal order of models
Seasonal Autoregressive Integrated Moving Average (SARIMA)	A modified model of the traditional ARIMA model, that takes seasonality patterns into consideration	1. Flexible and adaptable in time-series forecasting 2. Able to handle seasonality patterns 3. Automatic method of ideal model order selection	1. Complex of seasonality 2. Requires high computational resources 3. Not suitable for short term prediction
Holt's Winter (HW)	A time-series model that follows a Triple Exponential Smoothing technique which emphasize on levels, trends and seasonality attributes	1. Able to handle complexities in the data 2. Highly adaptable 3. Highly interpretability	1. Requires high computational resources 2. Does not perform well with constantly changing data
PROPHET Model	An open source tool that is developed by Facebook to perform time series analysis with the aspects of trend, seasonality and the context of holidays in its prediction	1. Suitable for context of holidays and crucial events 2. Highly adaptable 3. Convenient and easy to use	1. Does not perform well with non-linear trends 2. Prioritize only internally 3. Requires high computational resources

Table 4 - Summarized Table of Predictive Model Approaches

2.3.6 Ideal Model Selection

Based on the wide variety of models to choose from, it is clear to see that every model has its own strengths and weaknesses that defines each of the model's distinctiveness. Hence, the best way to go forward with the project is to compare the models in terms of their accuracy, precision and overall performance to determine the ideal predictive analytics model for the project. Even though each of the aforementioned models primarily functions as a time series analysis model, the two models that seems to stand out the most are the Holt's Winter Model and the PROPHET model. Both of the models are chosen amongst the four models due to the fact that they are easier to work with in accordance with the specific criteria and requirements of the project. On top of that, both models could handle the concept of seasonality patterns and fluctuations of data without too many complexities when delving into forecasting. Additionally, both Holt's Winter and PROPHET model do not have many specific components to keep in touch when compared to the ARIMA and SARIMA models. Hence, the best course of action would be to compare the two models, Holt's Winter and PROPHET model during the modeling phase for the succession of predicting sudden COVID-19 outbreaks in Malaysia.

2.4 Chapter Summary

To recap, Chapter 2 basically started off with the introduction to health diseases which includes infectious and non-infectious diseases while also noting the factors that may affect the transmission rate leading to outbreak according to various sources with similar mentioned factors. Afterwards, the review of related works discussed on the six journals with their own unique sets of goals and objectives, data sources used, machine learning algorithms that were performed and selected as the best model based on each journal findings respectively. Finally, Chapter 2 concludes on the predictive analytics approaches which explained in a detail and concise manner of five chose models based on the review of related works previously. The models are compared with one another in terms of their functionalities, equations/formulas applied, strengths and weaknesses. In the end, the Holt's Winter and PROPHET are chosen as comparison models to be evaluated for their accuracy, precision and performance in forecasting the sudden COVID-19 outbreaks in Malaysia so that a champion model can be declared as the ideal predictive analytics model for the project. Moving forward, Chapter 3 will analyze and document on data collection and preparation while fixating towards Exploratory Data Analysis (EDA).

CHAPTER 3

DATA COLLECTION AND PREPARATION

3.0 Overview

Chapter 3 will elaborate on the data collection and preparation for this project which further demonstrates on the data sources to be utilized, and the necessary data preparation techniques implemented which ultimately leads to data exploration concerning analysis of the datasets.

3.1 Data Sources

The data source of this project will be gathered from the Malaysian Ministry of Health (MoH) website called KKMNOW. Two key datasets will be incorporated in this project, each with the purpose of descriptive and predictive analytics. One of the datasets as shown in Table 5, will be primarily tasked in predictive analytics which aligns with the project objective of forecasting sudden COVID-19 outbreaks in Malaysia. On the other hand. both of the abovementioned datasets as presented in Table 5 and Table 6, can be used in descriptive analytics mainly for Exploratory Data Analysis (EDA) to gain informative insights through the COVID-19 trends and patterns. These datasets are accessible through downloading the CSV files containing the specific dataset from KKMNOW's open data catalogue. The data catalogue can be described as an open data portal that stores an abundance of health-related data from the Ministry of Health that offer limitless access to the public for their own particular uses. While both of the datasets basically provide crucial information on the daily COVID-19 cases in Malaysia from the year 2020 up until the present time, these datasets may differ in terms of their primary uses, columns and in-depth focus point.

The first dataset which will be implemented for descriptive and predictive analytics purposes, presents on the daily COVID-19 cases categorized by different dates, states and type of clusters associated with the number of cases. The contents of the first dataset are explained in Table 5 below:

Features	Column Name	Detail
Temporal Feature	Date	Represents the specific data according to YYYY-MM-DD format
Categorical Feature	State	Represent the specific state and federal territories in Malaysia or Malaysia as a whole
Numerical Feature	cases_new	Represents the number of new COVID-19 cases that are reported in the 24 hours since last report
	cases_import	Represents the number of new COVID-19 cases that are imported from infected individuals outside of Malaysia
	cases_recovered	Represents the number of COVID-19 cases that have recovered as reported in the 24 hours since the last report
	cases_active	Represents the number of active COVID-19 cases that are neither recovered nor died
	cases_cluster	Represents the number of cases related to clusters which can be used to calculate the number of sporadic cases

Table 5 – Tabulated Daily COVID-19 Cases by State and type of cases

The second dataset which will be applied for descriptive analytics purposes only, exhibit on the daily COVID-19 cases sorted into different dates, states, human life stages and specific age groups relating to the number of cases. The contents of the second dataset are explained in Table 6 below:

Features	Column Name	Detail
Temporal Feature	Date	Represents the specific data according to YYYY-MM-DD format
Categorical Feature	State	Represent the specific state and federal territories in Malaysia or Malaysia as a whole

Numerical Feature	cases_child	Represents the number of new COVID-19 cases for people aged 11 years old and below
	cases_adolescent	Represents the number of new COVID-19 cases for people aged 12 to 15 years old
	cases_adult	Represents the number of new COVID-19 cases for people aged 18 to 59 years old
	cases_elderly	Represents the number of new COVID-19 cases for people aged 60 years old and above
	Cases age 0-4	Represents the number of new COVID-19 cases for people aged 0 to 4 years old
	Cases aged 5-11	Represents the number of new COVID-19 cases for people aged 5 to 11 years old
	Cases aged 12-17	Represents the number of new COVID-19 cases for people aged 12 to 17 years old
	Cases aged 18-29	Represents the number of new COVID-19 cases for people aged 18 to 29 years old
	Cases aged 30-39	Represents the number of new COVID-19 cases for people aged 30 to 39 years old
	Cases aged 40-49	Represents the number of new COVID-19 cases for people aged 40 to 49 years old
	Cases aged 50-59	Represents the number of new COVID-19 cases for people aged 50 to 59 years old
	Cases aged 60-69	Represents the number of new COVID-19 cases for people aged 60 to 69 years old
	Cases aged 70-79	Represents the number of new COVID-19 cases for people aged 70 to 79 years old
	Cases aged 80+	Represents the number of new COVID-19 cases for people aged 80 years old and above

Table 6 – Tabulated Daily COVID-19 Cases by State, Life Stages and Age Groups

From the above tables 5 and 6, it can be concluded that both datasets have similar temporal and categorical features which are date and state respectively. Whereas the clear discrepancy between the two datasets lies in their numerical features, where one emphasizes more towards different type of cases, while the other vary in cases of human life stages and specific age groups. As of 1st February 2025, the number of rows for both datasets are about a total 31,000 respectively.

The cases_new column of the dataset in table 5 can be represented as the sum of the number of cases in cases_child, cases_adolescent, cases_adult and cases_elderly columns of the dataset in table 6. In other words, both of the datasets correlate with one another. Although there are correlations, the datasets could not integrate due to reporting lag. When comparing the results of both datasets, the number of cases in cases_new column seems to be slightly higher than the sum of cases in the cases_child, cases_adolescent, cases_adult and cases_elderly columns of dataset in table 6 mentioned above. This could mean that the dataset in table 5 recorded cases based on the day of reporting which would also capture cases that were from earlier dates. Whereas the dataset in table 6 is based on immediate reporting which only includes cases that were tested and reported on the same day. Hence, cases recorded in the dataset of table 5 may contain instances reported on a certain day but tested on an earlier date, therefore for some dates, the number of cases may be slightly higher than those of the dataset in table 6.

3.2 Data Preparation

Initially, the datasets will be collected from the KMMNOW website through the process of downloading the CSV files of each dataset. After downloading, the datasets will be uploaded to JupyterNotebook for cleaning and further visualization for descriptive analytics purposes uses. Python will be used as the main programming language for the abovementioned software. The first step for data preparation in this project is to import python libraries and packages in the script.

```
In [1]: # import packages
        import plotly.express as px
        import plotly.graph_objects as go
        import plotly.figure_factory as ff
        from plotly.subplots import make_subplots
        from datetime import datetime


        !pip install folium
        import folium

        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt


        %matplotlib inline

        import math
        import random
        from datetime import timedelta

        import warnings
        warnings.filterwarnings('ignore')
        import plotly as py
        py.offline.init_notebook_mode(connected = True)
```

Figure VII - Import Python Libraries & Packages

Figure VII shows the packages and libraries that are necessary for this project as they provide easy access to modules that would help in accomplishing certain tasks quicker and display the visuals for analytics. There are various charts and graphs to choose from which comprise of scatterplots, line charts, pie charts, tree maps and many more. Secondly is to read both of the datasets as mentioned in table 5 and table 6.

In [2]:
```
# read dataset
df = pd.read_csv('covid_case_type.csv')
casetype_all_states = df[['date', 'state', 'cases_new','cases_import']]
df2 = pd.read_csv('covid_case_age.csv')
caseage_all_states = df2[['cases_child','cases_adolescent','cases_adult',

# print casetype dataset
casetype_all_states
```

Out[2]:

	date	state	cases_new	cases_import
0	25/1/2020	Malaysia	4	4
1	26/1/2020	Malaysia	0	0
2	27/1/2020	Malaysia	0	0
3	28/1/2020	Malaysia	0	0
4	29/1/2020	Malaysia	3	3
...	...	...	...	...
26906	21/5/2024	W.P. Putrajaya	3	0
26907	22/5/2024	W.P. Putrajaya	5	0
26908	23/5/2024	W.P. Putrajaya	6	0
26909	24/5/2024	W.P. Putrajaya	4	0
26910	25/5/2024	W.P. Putrajaya	5	0

26911 rows × 4 columns

Figure VIII - Read Case Type and Age Group Dataset

The Figure VIII above shows the read dataset process for both of the datasets. Read both of the datasets and declare them into their respective variables so they could be used elsewhere for other steps onwards. Not to mention, both of them should be specified on which columns to be used as not all of the columns have significance for these descriptive tasks. The dataset in table 5 will be declared as casetype_all_states while the dataset in table 6 will be declared as caseage_all_states. After declaration, it is a good practice to print the dataset to make the results are correct, consistent and standardized. Since the datasets are free from any sorts of errors, missing values and inconsistencies, there is no need to go through a data cleaning process. Though, as shown in figure VII, the dataset seems to be lacking in appropriate features/columns which could detrimentally affect the descriptive analytics process. Hence, in this scenario it is best to add on new features which are suitable and relevant towards the objective of this project.

The columns that will be added are called epidemic_week, public_holiday, mco and last but not least monsoon_season. Each of the columns play a significant role for this project. The first one is epidemic_week which is also known as epi week for some people. This term is typically used for disease outbreaks as it tracks the epidemiological activities of a disease in a seven-day time period.

```
# Function to get the start date of the epidemic for each year
def get_epidemic_start_date(year):
    # Assuming the epidemic starts on January 25 each year
    return datetime(year, 1, 25)

# Function to convert date to epidemic week, restarting each year
def get_epidemic_week(date):
    epidemic_start_date = get_epidemic_start_date(date.year)
    if date < epidemic_start_date:
        return 1
    days_since_start = (date - epidemic_start_date).days
    epidemic_week = days_since_start // 7 + 1
    return epidemic_week

# Apply the function to the 'date' column to get the epidemic week
casetype_all_states['epidemic_week'] = casetype_all_states['date'].apply(get_epidemic_week)

casetype_all_states
```

	date	state	cases_new	cases_import	epidemic_week
0	2020-01-25	Malaysia	4	4	1
1	2020-01-26	Malaysia	0	0	1
2	2020-01-27	Malaysia	0	0	1
3	2020-01-28	Malaysia	0	0	1
4	2020-01-29	Malaysia	3	3	1
...	...	...	...	...	...
26906	2024-05-21	W.P. Putrajaya	3	0	17
26907	2024-05-22	W.P. Putrajaya	5	0	17
26908	2024-05-23	W.P. Putrajaya	6	0	18
26909	2024-05-24	W.P. Putrajaya	4	0	18

Figure IX - Function to add Epidemic Week

The above figure IX presents on the function to add the new column called epidemic week. It starts off with a function to get the start day for the beginning of the epidemic week and then the function to get the value for the epidemic week which is based on the date of days for the casetype_all_states dataset. When printing, the results display the new column and appropriate value given to each of the rows. The epidemic week will not be used in this descriptive analysis as it is much more appropriate for predicting and forecasting tasks later on.

Next is the public_ holiday column which essentially indicate whether the given date is a holiday or not.

```
# list out all holidays in malaysia 2020
chinese_ny = ('25/1/2020', '26/1/2020')
labour_day = ('1/5/2020',)
wesak = ('7/5/2020',)
aidilfitri = ('24/5/2020', '25/5/2020')
ydpa_bday = ('8/6/2020',)
aidiladha = ('31/7/2020',)
maal_hijrah = ('20/8/2020',)
national_day = ('31/8/2020',)
malaysia_day = ('16/9/2020',)
maulidur_rasul = ('29/10/2020',)
deepavali = ('14/11/2020',)
christmas = ('25/12/2020',)

public_hol20 = (chinese_ny,labour_day,wesak,aidilfitri,ydpa_bday,aidiladha,maal_hijrah,national_day,malaysia_day,maulidur_rasul,c

# Flatten the list of holiday dates
public_hol20_flat = [date for sublist in public_hol20 for date in sublist]

# Convert the date column to datetime format
casetype_all_states['date'] = pd.to_datetime(casetype_all_states['date'], format='%d/%m/%Y')
```

Figure X - List out all holidays in Malaysia

Based on figure X, it is important to initially manually set the dates which consist of the public holidays all around Malaysia from year 2020 to year 2024 so it could later be applied in a function.

```
def public_holiday(dates):
    if dates in public_hol20_dates:
        return 'yes'
    if dates in public_hol21_dates:
        return 'yes'
    if dates in public_hol22_dates:
        return 'yes'
    if dates in public_hol23_dates:
        return 'yes'
    if dates in public_hol24_dates:
        return 'yes'
    return 'no'

casetype_all_states['public_holiday'] = casetype_all_states['date'].apply(public_holiday)

#casetype_all_states['date'] = casetype_all_states['date'].dt.strftime('%d/%m/%Y')
casetype_all_states
```

	date	state	cases_new	cases_import	epidemic_week	public_holiday
0	2020-01-25	Malaysia	4	4	1	yes
1	2020-01-26	Malaysia	0	0	1	yes
2	2020-01-27	Malaysia	0	0	1	no
3	2020-01-28	Malaysia	0	0	1	no
4	2020-01-29	Malaysia	3	3	1	no
...	...	...	...	...	...	...
26906	2024-05-21	W.P. Putrajaya	3	0	17	no
26907	2024-05-22	W.P. Putrajaya	5	0	17	yes
26908	2024-05-23	W.P. Putrajaya	6	0	18	no
26909	2024-05-24	W.P. Putrajaya	4	0	18	no
26910	2024-05-25	W.P. Putrajaya	5	0	18	no

Figure XI - Function to add Public Holiday

As presented in figure XI, the function called public_holiday will be used to determine whether it is a public holiday in Malaysia or not by indicating “yes” for holiday and “no” for not a public holiday. It will be calculated from the year 2020 up until the present year which is year 2024. Moving on, is to add on the mco column which stands for Movement Control Order in Malaysia.

```
start_mco = datetime.strptime('2020-03-18', '%Y-%m-%d')
end_mco = datetime.strptime('2022-01-03', '%Y-%m-%d')

def dates_of_mco(dates):
    if (dates>=start_mco and dates<=end_mco):
        return 'yes'
    return 'no'

casetype_all_states['mco'] = casetype_all_states['date'].apply(dates_of_mco)
casetype_all_states
```

	date	state	cases_new	cases_import	epidemic_week	public_holiday	mco
0	2020-01-25	Malaysia	4	4	1	yes	no
1	2020-01-26	Malaysia	0	0	1	yes	no
2	2020-01-27	Malaysia	0	0	1	no	no
3	2020-01-28	Malaysia	0	0	1	no	no
4	2020-01-29	Malaysia	3	3	1	no	no
...	...	...	...	...	...	...	...
26906	2024-05-21	W.P. Putrajaya	3	0	17	no	no
26907	2024-05-22	W.P. Putrajaya	5	0	17	yes	no
26908	2024-05-23	W.P. Putrajaya	6	0	18	no	no
26909	2024-05-24	W.P. Putrajaya	4	0	18	no	no
26910	2024-05-25	W.P. Putrajaya	5	0	18	no	no

26911 rows × 7 columns

Figure XII - Function to add MCO column

As shown in figure XII, the start date and end date of the MCO will be declared first so it could be validated by the dates_of_mco function to determine whether or not the date is an MCO or not with values of either “yes” or “no”. Finally is the last column to add on which is the monsoon_season column. This column basically will indicate the monsoon season for each date of days.

```python
# Function to determine the monsoon season
def monsoon_season(date):
    month = date.month
    if month in [11, 12, 1, 2, 3]:
        return 'Northeast'
    elif month in [5, 6, 7, 8, 9]:
        return 'Southwest'
    else:
        return 'Inter-Monsoon'

# Create a new column 'Monsoon Season' based on the monsoon season
casetype_all_states['monsoon_season'] = casetype_all_states['date'].apply(monsoon_season)
casetype_all_states
```

	date	state	cases_new	cases_import	epidemic_week	public_holiday	mco	monsoon_season
0	2020-01-25	Malaysia	4	4	1	yes	no	Northeast
1	2020-01-26	Malaysia	0	0	1	yes	no	Northeast
2	2020-01-27	Malaysia	0	0	1	no	no	Northeast
3	2020-01-28	Malaysia	0	0	1	no	no	Northeast
4	2020-01-29	Malaysia	3	3	1	no	no	Northeast
...	...	...	...	...	...	...	...	...
26906	2024-05-21	W.P. Putrajaya	3	0	17	no	no	Southwest
26907	2024-05-22	W.P. Putrajaya	5	0	17	yes	no	Southwest
26908	2024-05-23	W.P. Putrajaya	6	0	18	no	no	Southwest
26909	2024-05-24	W.P. Putrajaya	4	0	18	no	no	Southwest
26910	2024-05-25	W.P. Putrajaya	5	0	18	no	no	Southwest

26911 rows × 8 columns

Figure XIII - Function to add Monsoon Season column

The above figure XIII displays the function for monsoon_season which will help in identifying the related seasons such as the Northeast, Southwest and Inter-Monsoon seasons in Malaysia. The table as shown in the figure showcases the final result of the data preparation process with 8 different columns which are significant for the analytics tasks in data exploration.

3.3 Data Exploration

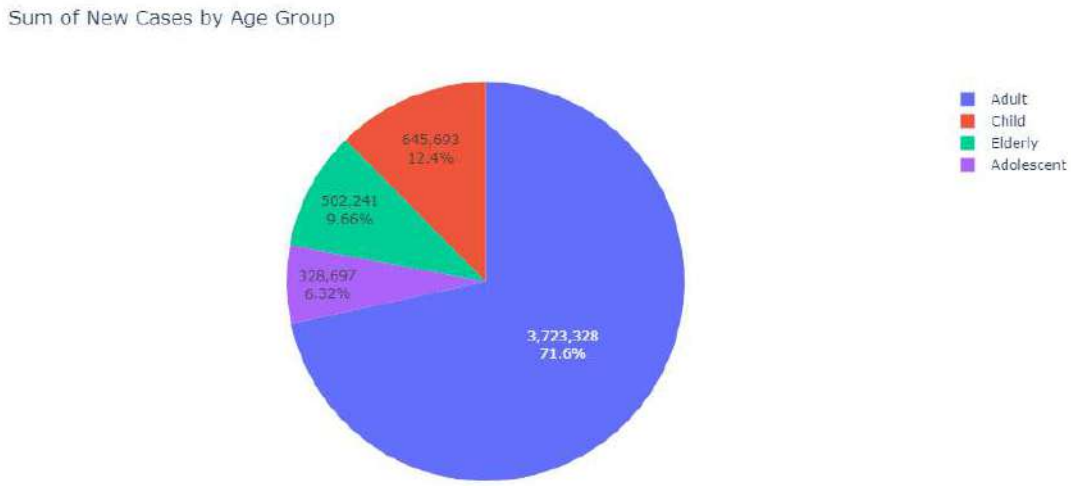

Figure XIV - Sum of new cases by Age group (Pie Chart)

As shown in the pie chart in figure XIV above, it represents the sum of new cases by age group. The pie chart itself is coloured according to the legend on the left side which indicate the specific groups of each different colours. It is clear to see that adult sections seems to have taken the majority of the pie chart with a whopping value of 3,723,328 sum of new cases which covers 71.6% of the entire pie chart. This comes to no surprise as adults are usually out and about in their daily lives, meeting and interacting with numerous people from all walks of life which might have contributed to the high number of COVID-19 cases particularly for adults when compared to the other age groups. Hence, from this pie chart we can inference whereby that adults are much more prone to being infected from the COVID-19 disease due to their own behaviours and lifestyles as they have to travel to many places for various reasons such as work related, family related etc.

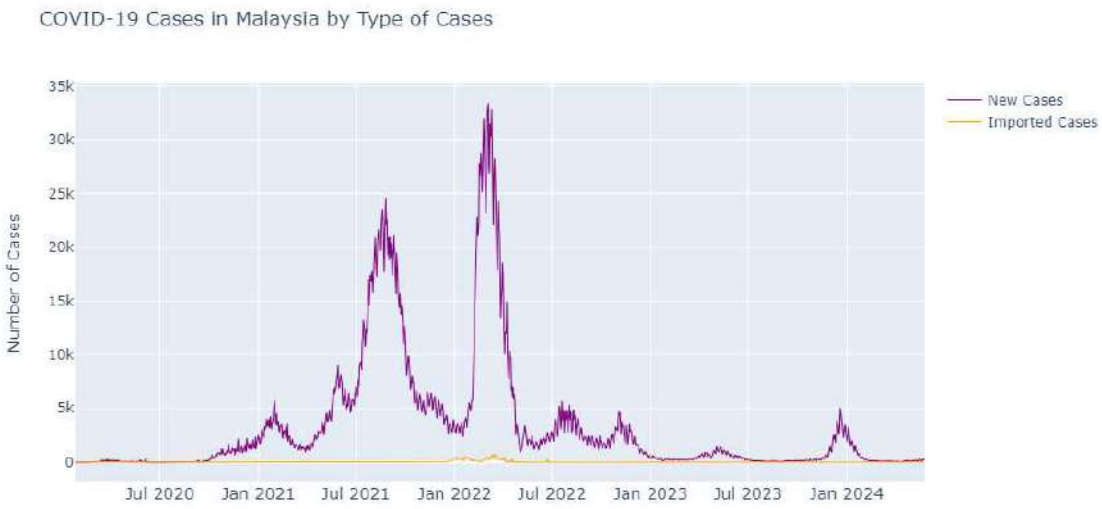

Figure XV - Covid 19 number of cases by type of case (Line Chart)

The above figure XV shows a line chart that represents the Covid 19 cases in Malaysia by the specific types of cases as recorded for each day. These cases are categorized into two which are new cases coloured in purple and imported cases coloured in orange. From the figure, we can see that initially most of the new cases comes from the imported cases which strongly indicates the beginning of the COVID 19 transmission from the other countries to the citizens of Malaysia. As time goes on, the new cases started to spike up from time to time while the imported cases seems to remain almost constant or barely noticeable in its changes. This concludes that eventhough the imported cases from other countires may have slightly contributed to the transmission rate of COVID 19 at first, it does not signify its cruciality as a main factor of how the new cases might appear from time to time and the emergence of sudden outbreaks. Thefore, there is low possibility that the apparent high cases are because of COVID 19 disease transmission from outside of Malaysia.

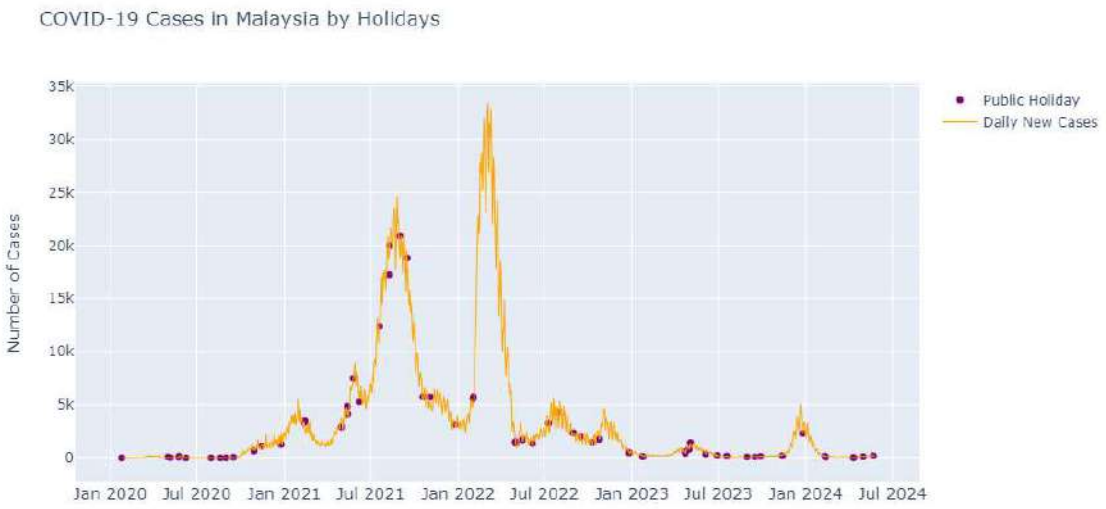

Figure XVI - Covid 19 number of cases by Holidays (Line Chart + Scatter Plot)

The above figure XVI showw a combination of a line chart and scatter plot that represents the Covid 19 cases in Malaysia by Holidays. The purple coloured plots indicate the presence of public holdiay for that particular day, whereas the orange coloured line depicts the daily new cases of Covid 19 in Malaysia. Just from observing the graph itself, it is obvious to see that the public holidays might of played a crucial role in the sudden spike of new cases which indicate a sudden Covid 19 outbreak appearing. Particularly in the months between July 2021 and January 2022 where there it is at the end of the year whereby most people would go travelling and have their own vacation. Thus, contributing to the sudden high number of daily cases which means that public holidays in Malaysia could be one of the key factors affecting rate of transmission of Covid 19 disease outbreaks.

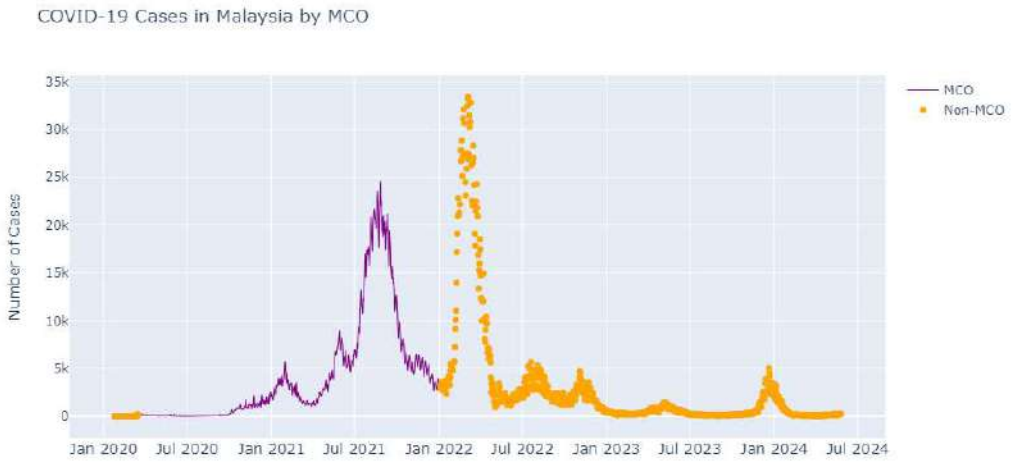

Figure XVII - Covid 19 number of cases by MCO (Line Chart + Scatter Plot)

The above figure XVII also showcases a mixture of a line chart and scatter plot that depicts the Covid 19 cases in Malaysia by MCO. Two things to take into consideration are purple coloured plots for MCO and the orange coloured lines for Non-MCO. This means that in some days the act of the Malaysia Movement Control Order have taken place whereas the other days are consist of Non MCO's days. We could observe whereby at first when the MCO is conducted, the number of new cases seems to be in control and not change in a drastic way. Closing on January 2021, cases have suddenly begun to rise up until July 2021 since during that time, the necessary safety measures and protocols were not yet implemented as some people were still eager to move around as they do not take the issue with COVID-19 issue seriously. Not to mention, people were still unware on how to adapt in the new environment whereby they had to quarantine themselves in order control the transmission of the disease. Afterwards, the number of new cases started to gradually decrease up until January 2022 as stricter rules and MCO regulations were implemented for the public as an effort by the Malaysian government to advert the crisis. Then, the MCO stopped completely as the situation seems to be in control which caused the massive spike of number of new cases to appear as many people are able to travel to various places and meet their family, relatives and even loved ones. As time goes on, the cases have begun to dropped at a sigficant rate as most people are immuned to the virus since the vaccines are widely distributed to various parts of Malaysia making it much more difficult for the disease to infect others. Hence, this entails that the presence of MCO does play a key factor in affecting the rate of outbreaks.

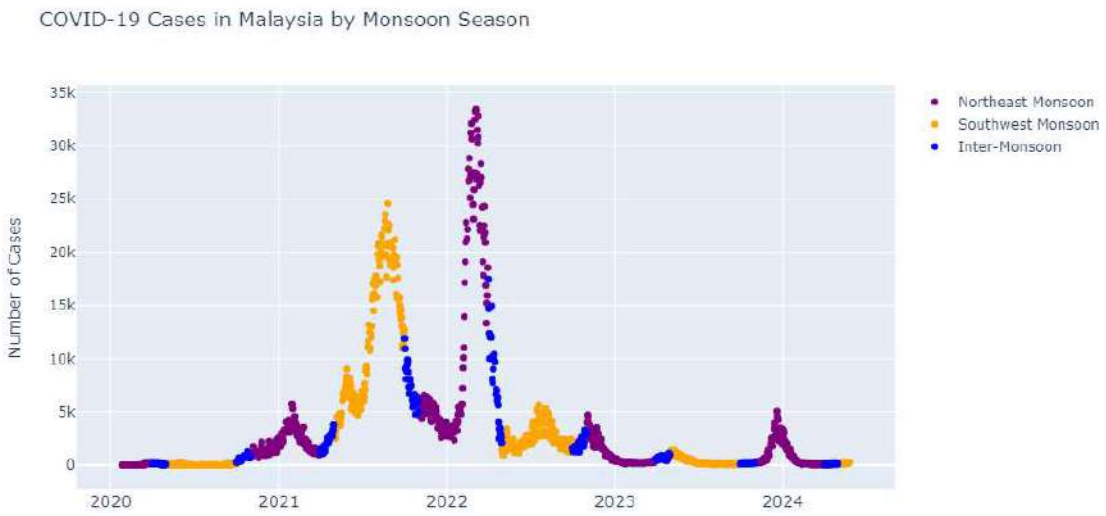

Figure XVIII - Covid 19 number of cases by Monsoon season (Scatter Plot)

Figure XVIII above displays a scatter plot which represents the Covid 19 cases in Malaysia by Monsoon Season. In Malaysia, there are 3 specific monsoon seasons which includes Northeast, Southwest and Inter-Monsoon. These monsoon seasons vary in their timing and climate change. Just from the graph, we could see that both the purple coloured Northeast Monsoon and orange coloured Southwest Monsoon highly affect the number of new cases which clearly indicates the two massive sudden spikes of cases. Whereas, the Inter-Monsoon does not contribute much to the rise of these cases as usually it stands in the middle of these two main Monsoon seasons. This signify the Covid 19 disease capability to adapt in extreme weathers and conditions as during the Northeast Monsoon, many parts of Malaysia are showered with heavy rain downpour while during the Southwest Monsoon, most states in Malaysia will experince less rain which leads to hot weather and drier conditions. This means that the scope of weather change will affect how Covid 19 reacts in specific situation which may lead to a higher number of new cases and sudden outbreak to appear from time to time.

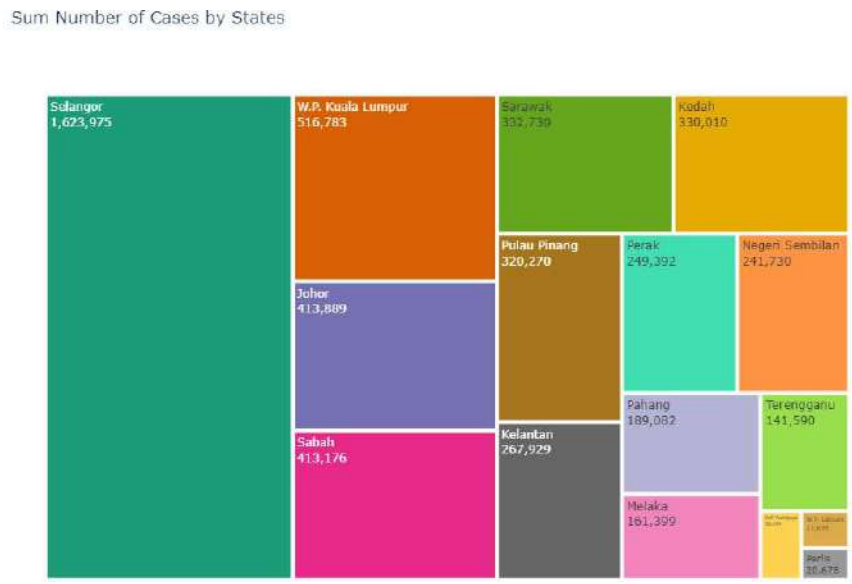

Figure XIX - Covid 19 sum of new cases by States (Treemap)

Figure XIX above showcases a Treemap that represents the Sum number of Covid 19 cases by state. This treemap depicts each state with its respective sum of new cases and colours. As the state with the largest value, Selangor's sort of box area is much larger compared to the other states as shown from the figure above. The reason why Selangor is the largest and has the highest value by a longshot is due to its high density of population. When we look at the three biggest area, Selangor, Kuala Lumpur and Johor, all three of them are highly populated, not to mention since they are consisting of wide area of cities, the weather would usually be extremely hot due to the numerous building all over the state and lack of greenery. This concludes that the population and development of a state may contribute to the high number of new cases of emerging.

3.4 Chapter Summary

In a nutshell, Chapter 3 had emphasized on the two datasets used for this project and the steps towards preparing the data for data exploration which uncovers the into an in-depth analytical perspective. Chapter 4 onwards will continue with model development and evaluation to determine the ideal machine learning algorithm for this project.

CHAPTER 4

MODEL DEVELOPMENT AND EVALUATION

4.0 Overview

Chapter 4 will focus more on the model development and evaluation phase which at first explains on the step-by-step process of data preprocessing and preparation for the use of modelling. In this case, modelling involves utilizing the selected machine learning algorithms based on the decisions made from Chapter 2 for forecasting COVID-19 cases according to two different forecasting goals namely short and long terms forecasting respectively. This will later on resume with the model's evaluation and comparison to determine the ideal predictive model for both short- and long-term forecasting goals in hopes to be fully incorporated to a well-defined dashboard.

4.1 Data Preprocessing and Preparation

Figure XX – Initial Outlook Files/Folders in Project Folder

For this project, the main focus is on the FYP2 folder which consists of an Icon folder, a Model_Development jupyter notebook file and a Covid19_Dashboard tableau workbook file as shown in the initial outlook on Figure XX above. As this chapter 4 will delve into the model development and evaluation stage, the Model_Development.ipynb file will serves as the appropriate file to start with. As or right now, there is only 3 files/folders but later on the final outlook of the FYP2 folder will change accordingly.

```
# Install and import pandas for parquet files
!pip install pandas fastparquet
import pandas as pd

# Install folium for data visualization
!pip install folium
import folium

# Import numpy for numerical and mathematical functions
import numpy as np
import math

# Import for date and time handling
from datetime import datetime
from datetime import timedelta

# Import for RNG and selecting random values
import random

# Import matplotlib's pyplot module for interactive visualizations
import matplotlib.pyplot as plt

# Command to display matplotlib plots directly to notebook as inline images
%matplotlib inline

# Import Plotly and enable offline mode
import plotly as py
py.offline.init_notebook_mode(connected = True)

# Import Plotly express for interactive plots
import plotly.express as px

# Import Plotly low lvl API for control over chart customization
import plotly.graph_objects as go

# Import Plotly figure factory module for complex visualization (Eg: Heatmap)
import plotly.figure_factory as ff

# Import make_subplots function for complex layout with multiple charts in one figure
from plotly.subplots import make_subplots

# Import for warning control by suppressing non-critical warnings
import warnings
warnings.filterwarnings('ignore')
```

Figure XXI - Import Library and Packages

To start things off, open the Model_Development.ipynb file by double clicking on it. All in all, the codes involved in Chapter 4 differs from Chapter 3's code as it consists of newer features and functionalities to accommodate with the goal of building a predictive model for a comprehensive dashboard. As shown in Figure XXI, when working with python codes it is important to import and install all of the necessary libraries and packages beforehand. Each imports play a massive role as without them some of the features and function inside Jupyter Notebook might not work as intended.

Dataset 1: Type of Cases by State

```
URL_DATA_CASES = 'https://storage.data.gov.my/healthcare/covid_cases.parquet'

df = pd.read_parquet(URL_DATA_CASES)
if 'date' in df.columns: df['date'] = pd.to_datetime(df['date'])

df
```

	date	state	cases_new	cases_import	cases_recovered	cases_active	cases_cluster
0	2020-01-25	Malaysia	4	4	0	4	0
1	2020-01-26	Malaysia	0	0	0	4	0
2	2020-01-27	Malaysia	0	0	0	4	0
3	2020-01-28	Malaysia	0	0	0	4	0
4	2020-01-29	Malaysia	3	3	0	7	0
...	...	...	...	...	...	...	...
31071	2025-01-21	W.P. Putrajaya	5	0	0	249	0
31072	2025-01-22	W.P. Putrajaya	1	0	5	245	0
31073	2025-01-23	W.P. Putrajaya	0	0	0	245	0
31074	2025-01-24	W.P. Putrajaya	3	0	0	248	0
31075	2025-01-25	W.P. Putrajaya	1	0	5	244	0

31076 rows × 7 columns

Figure XXII - Extract Dataset 1 from KKMNOW

The next step is to extract the first dataset by reading through parquet, which allows easy access on the latest dataset from the website KKMNOW. Read the dataset from the URL link and insert it into a dataframe such as df. As a good practice, one should print out the dataframe to check the dataset whether its features, records and overall content is the correct one for the project.

Dataset 2: Age Group by State

```
URL_DATA_AGE = 'https://storage.data.gov.my/healthcare/covid_cases_age.parquet'

df2 = pd.read_parquet(URL_DATA_AGE)
if 'date' in df2.columns: df2['date'] = pd.to_datetime(df2['date'])

df2
```

	date	state	cases_child	cases_adolescent	cases_adult	cases_elderly	cases_0_4	cases_5_11	cases_12_17	cases_18_29	cases_30_39	cases_40_49	cases_50_59
0	2020-01-25	Malaysia	0	0	1	0	0	0	0	0	0	1	0
1	2020-01-26	Malaysia	0	0	0	0	0	0	0	0	0	0	0
2	2020-01-27	Malaysia	0	0	0	0	0	0	0	0	0	0	0
3	2020-01-28	Malaysia	0	0	0	0	0	0	0	0	0	0	0
4	2020-01-29	Malaysia	1	0	2	0	1	0	0	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
31071	2025-01-21	W.P. Putrajaya	0	0	4	1	0	0	0	1	3	0	0
31072	2025-01-22	W.P. Putrajaya	0	0	1	0	0	0	0	0	0	1	0
31073	2025-01-23	W.P. Putrajaya	0	0	0	0	0	0	0	0	0	0	0
31074	2025-01-24	W.P. Putrajaya	0	0	2	1	0	0	0	2	0	0	0
31075	2025-01-25	W.P. Putrajaya	0	0	1	0	0	0	0	0	1	0	0

31076 rows × 16 columns

Figure XXIII - Extract Dataset 2 from KKMNOW

Likewise, also extract the second dataset by the same method and insert it into a dataframe called df2 as shown in Figure XXIII. Basically, when a user press run button on jupyter notebook the code will extract the necessary datasets directly through the given link. This ensures that the dataset consists of the latest records and remove the need to download the datasets everytime. As such, it can be considered as an auto update data extraction feature.

```
df = df[['date', 'state', 'cases_new','cases_import','cases_recovered','cases_active']]
df.rename(columns={'cases_new': 'new_cases'},  inplace=True)
df.rename(columns={'cases_import': 'imported_cases'}, inplace=True)
df.rename(columns={'cases_recovered': 'recovered_cases'}, inplace=True)
df.rename(columns={'cases_active': 'active_cases'}, inplace=True)

columns_to_add = ['cases_child','cases_adolescent','cases_adult','cases_elderly']
selected_columns = df2[columns_to_add]
df = pd.concat([df, selected_columns], axis=1)

df.rename(columns={'cases_child': 'child_cases'},  inplace=True)
df.rename(columns={'cases_adolescent': 'adolescent_cases'}, inplace=True)
df.rename(columns={'cases_adult': 'adult_cases'}, inplace=True)
df.rename(columns={'cases_elderly': 'elderly_cases'}, inplace=True)
df
```

	date	state	new_cases	imported_cases	recovered_cases	active_cases	child_cases	adolescent_cases	adult_cases	elderly_cases
0	2020-01-25	Malaysia	4	4	0	4	0	0	1	0
1	2020-01-26	Malaysia	0	0	0	4	0	0	0	0
2	2020-01-27	Malaysia	0	0	0	4	0	0	0	0
3	2020-01-28	Malaysia	0	0	0	4	0	0	0	0
4	2020-01-29	Malaysia	3	3	0	7	1	0	2	0
...	...	...	...	...	...	...	...	...	...	...
31071	2025-01-21	W.P. Putrajaya	5	0	0	249	0	0	4	1
31072	2025-01-22	W.P. Putrajaya	1	0	5	245	0	0	1	0
31073	2025-01-23	W.P. Putrajaya	0	0	0	245	0	0	0	0
31074	2025-01-24	W.P. Putrajaya	3	0	0	248	0	0	2	1
31075	2025-01-25	W.P. Putrajaya	1	0	5	244	0	0	1	0

31076 rows × 10 columns

Figure XXIV - Select and Rename Columns for Dataset Preparation

After that, we move to the data preparation and its components which ensures the dataset to fully ready before applying or feeding it to the machine learning algorithms. One of the first steps is to adjust the dataframes as we want to focus on just a single dataframe rather than using both of the extracted datasets. As shown in Figure XXIV, this involves selecting the suitable features and columns which are relevant for this project, renaming these columns to allow easy standardized identification and consistency and lastly integrating all of them into a single dataframe which contains the integrated datasets of both the first and second datasets.

```python
# Function to get the start date of the epidemic for each year
def get_epidemic_start_date(year):
    if year == 2020:
        return datetime(year, 1, 25)
    else:
        return datetime(year, 1, 1)

# Function to convert date to epidemic week, restarting each year
def get_epidemic_week(date):
    epidemic_start_date = get_epidemic_start_date(date.year)
    if date < epidemic_start_date:
        return 1
    days_since_start = (date - epidemic_start_date).days
    epidemic_week = days_since_start // 7 + 1
    return epidemic_week

# Apply the function to the 'date' column to get the epidemic week
df['epidemic_week'] = df['date'].apply(get_epidemic_week)

# Create new year column
df['year'] = df['date'].dt.year

# Function to calculate the start date of each epidemic week
def get_week_start(year, epidemic_week):
    if year == 2020:
        first_day_of_year = datetime(year, 1, 25)
        start_date = first_day_of_year + timedelta(weeks=epidemic_week - 1)
        return start_date
    else:
        first_day_of_year = datetime(year, 1, 1)
        start_date = first_day_of_year + timedelta(weeks=epidemic_week - 1)
        return start_date

# Apply the function to calculate the start date for each row
df['start_date'] = df.apply(lambda row: get_week_start(row['year'], row['epidemic_week']), axis=1)

# Modify start date to datetime format
df['start_date'] = df['start_date'].dt.strftime('%Y-%m-%d')

# Rearrange the columns
df = df[['date', 'year','epidemic_week', 'start_date','state','new_cases','imported_cases','recovered_cases','active_cases','child_cases','adolescent_cas
df
```

Figure XXV - Add Year, Epidemic Week, and Start Date Columns

Moving on to the next step for data preparation is adding new columns and features which could prove useful in effectively visualizing the data into appropriate graphs and also for extra features for the models as in some ways it may or may not improve the performance of the predictive models. As shown in Figure XXV, three columns will be added to the dataset which are year, epidemic week and start date respectively. The functions involved helps to automate the process of classifying the appropriate values for each record with the specific dates in mind.

	date	year	epidemic_week	start_date	state	new_cases	imported_cases	recovered_cases	active_cases	child_cases	adolescent_cases	adult_cases	elderly_c
0	2020-01-25	2020	1	2020-01-25	Malaysia	4	4	0	4	0	0	1	
1	2020-01-26	2020	1	2020-01-25	Malaysia	0	0	0	4	0	0	0	
2	2020-01-27	2020	1	2020-01-25	Malaysia	0	0	0	4	0	0	0	
3	2020-01-28	2020	1	2020-01-25	Malaysia	0	0	0	4	0	0	0	
4	2020-01-29	2020	1	2020-01-25	Malaysia	3	3	0	7	1	0	2	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
31071	2025-01-21	2025	3	2025-01-15	W.P. Putrajaya	5	0	0	249	0	0	4	
31072	2025-01-22	2025	4	2025-01-22	W.P. Putrajaya	1	0	5	245	0	0	1	
31073	2025-01-23	2025	4	2025-01-22	W.P. Putrajaya	0	0	0	245	0	0	0	
31074	2025-01-24	2025	4	2025-01-22	W.P. Putrajaya	3	0	0	248	0	0	2	
31075	2025-01-25	2025	4	2025-01-22	W.P. Putrajaya	1	0	5	244	0	0	1	

31076 rows × 13 columns

Figure XXVI - Output of Year, Epidemic Week and Start Date Columns

As shown in Figure XXVI, the additional columns are automatically added to each record from earliest to the latest with varied values. The epidemic week column is necessary to indicate the start of the COVID-19 cases and ends by the end of the year which is either 52 or 53 epidemic weeks in a year total depending on the differing number of days in a year. As such it will start at 1 and will reset back to epidemic week 1 when it reaches a new year. The year and start date columns are just referenced for the epidemic weeks to allow easier traceback.

```
# Initialize start and end date of MCO into datetime format
start_mco = datetime.strptime('2020-03-18', '%Y-%m-%d')
end_mco = datetime.strptime('2021-12-31', '%Y-%m-%d')

# Function to get the exact dates for MCO
def dates_of_mco(dates):
    if (dates >= start_mco and dates <= end_mco):
        return 'Yes'
    return 'No'

# Apply the function to get dates of MCO for each row
df['mco'] = df['date'].apply(dates_of_mco)

# Filter date for MCO periods only
df.query('mco == "Yes"')
```

ate	year	epidemic_week	start_date	state	new_cases	imported_cases	recovered_cases	active_cases	child_cases	adolescent_cases	adult_cases	elderly_cases	mco
20--18	2020	8	2020-03-14	Malaysia	117	8	11	728	4	8	78	26	Yes
20--19	2020	8	2020-03-14	Malaysia	110	5	15	823	6	7	77	20	Yes
20--20	2020	8	2020-03-14	Malaysia	130	6	12	940	2	7	96	22	Yes
20--21	2020	9	2020-03-21	Malaysia	153	11	27	1062	7	4	106	33	Yes
20--22	2020	9	2020-03-21	Malaysia	123	21	25	1156	2	6	95	20	Yes
...	...	...	...	...	...	...	...	...	...	...	...	...	...
21--27	2021	52	2021-12-24	W.P. Putrajaya	11	1	32	407	2	0	8	1	Yes
21--28	2021	52	2021-12-24	W.P. Putrajaya	24	1	21	410	2	2	19	1	Yes
21--29	2021	52	2021-12-24	W.P. Putrajaya	22	3	20	412	4	1	16	1	Yes
21--30	2021	52	2021-12-24	W.P. Putrajaya	25	4	28	409	2	1	20	2	Yes
21--31	2021	53	2021-12-31	W.P. Putrajaya	24	2	26	407	4	1	19	0	Yes

Figure XXVII - Add MCO Column

Apart from that, another column to include is the MCO column. When the system/code runs, the function in Figure XXVII above will automatically provide values on whether dates are either “Yes” for MCO or “No” for non MCO for each date records. This feature is helpful in determining the difference in cases during mid pandemic and post pandemic just to see whether there is any significant impact. Additionally, it may also be used as an additional regressor for the model’s performance.

```python
# Function to determine the monsoon season
def monsoon_season(date):
    month = date.month
    if month in [11, 12, 1, 2, 3]:
        return 'Northeast'
    elif month in [5, 6, 7, 8, 9]:
        return 'Southwest'
    else:
        return 'Inter-Monsoon'

# Create a new column 'Monsoon Season' based on the monsoon season
df['monsoon_season'] = df['date'].apply(monsoon_season)
df
```

ic_week	start_date	state	new_cases	imported_cases	recovered_cases	active_cases	child_cases	adolescent_cases	adult_cases	elderly_cases	mco	monsoon_season
1	2020-01-25	Malaysia	4	4	0	4	0	0	1	0	No	Northeast
1	2020-01-25	Malaysia	0	0	0	4	0	0	0	0	No	Northeast
1	2020-01-25	Malaysia	0	0	0	4	0	0	0	0	No	Northeast
1	2020-01-25	Malaysia	0	0	0	4	0	0	0	0	No	Northeast
1	2020-01-25	Malaysia	3	3	0	7	1	0	2	0	No	Northeast
...	...	...	...	...	...	...	...	...	...	...	...	...
3	2025-01-15	W.P. Putrajaya	5	0	0	249	0	0	4	1	No	Northeast
4	2025-01-22	W.P. Putrajaya	1	0	5	245	0	0	1	0	No	Northeast
4	2025-01-22	W.P. Putrajaya	0	0	0	245	0	0	0	0	No	Northeast
4	2025-01-22	W.P. Putrajaya	3	0	0	248	0	0	2	1	No	Northeast
4	2025-01-22	W.P. Putrajaya	1	0	5	244	0	0	1	0	No	Northeast

Figure XXVIII - Add Monsoon Season column

Aside from that, another feature to add is for Monsoon Season. Its values consist of Northeast, Southwest and Inter-Monsoon. As shown in Figure XXVII, the monsoon season column is added into the dataset for each record and the values are appropriately given based on the months, which helps to indicate the current type of Monsoon season during the specific date or time period.

```
# Define holiday dates as sets of datetime objects
cny = {pd.to_datetime(date, format='%d/%m/%Y') for date in ('25/1/2020', '26/1/2020', '12/2/2021', '13/2/2021', '1/2/2022', '2/2/2022', '23/1/2023', '24
labour_day = {pd.to_datetime(date, format='%d/%m/%Y') for date in ('1/5/2020', '1/5/2021', '1/5/2022', '1/5/2023', '1/5/2024','1/5/2025')}
wesak = {pd.to_datetime(date, format='%d/%m/%Y') for date in ('7/5/2020', '26/5/2021', '16/5/2022', '4/5/2023', '22/5/2024','12/5/2025')}
aidilfitri = {pd.to_datetime(date, format='%d/%m/%Y') for date in ('24/5/2020', '25/5/2020', '13/5/2021', '14/5/2021', '1/5/2022', '2/5/2022', '22/4/202
ydpa_bday = {pd.to_datetime(date, format='%d/%m/%Y') for date in ('8/6/2020', '7/6/2021', '6/6/2022', '5/6/2023', '3/6/2024','2/6/2025')}
aidiladha = {pd.to_datetime(date, format='%d/%m/%Y') for date in ('31/7/2020', '20/7/2021', '10/7/2022', '29/6/2023', '17/6/2024','6/6/2025','7/6/2025')
maal_hijrah = {pd.to_datetime(date, format='%d/%m/%Y') for date in ('20/8/2020', '9/8/2021', '10/8/2021', '30/7/2022', '19/7/2023', '7/7/2024','26/6/202
national_day = {pd.to_datetime(date, format='%d/%m/%Y') for date in ('31/8/2020', '31/8/2021', '31/8/2022', '31/8/2023', '31/8/2024','31/8/2025')}
malaysia_day = {pd.to_datetime(date, format='%d/%m/%Y') for date in ('16/9/2020', '16/9/2021', '16/9/2022', '16/9/2023', '16/9/2024','16/9/2025')}
maulidur_rasul = {pd.to_datetime(date, format='%d/%m/%Y') for date in ('29/10/2020', '19/10/2021', '9/10/2022', '28/9/2023', '16/9/2024','4/9/2025')}
deepavali = {pd.to_datetime(date, format='%d/%m/%Y') for date in ('14/11/2020', '4/11/2021', '24/10/2022', '13/11/2023', '31/10/2024','20/10/2025')}
christmas = {pd.to_datetime(date, format='%d/%m/%Y') for date in ('25/12/2020', '25/12/2021', '25/12/2022', '25/12/2023', '25/12/2024','25/12/2025')}

# Convert the date column to a datetime format
df['date'] = pd.to_datetime(df['date'], format='%d/%m/%Y')

# Function to determine public holiday type based on date
def holiday_type(dates):
    if dates in cny:
        return 'Chinese New Year'
    elif dates in labour_day:
        return 'Labour Day'
    elif dates in wesak:
        return 'Wesak Day'
    elif dates in aidilfitri:
        return 'Eid Al-Fitr'
    elif dates in ydpa_bday:
        return 'YDPA Birthday'
    elif dates in aidiladha:
        return 'Eid al-Adha'
    elif dates in maal_hijrah:
        return 'Maal Hijrah'
    elif dates in national_day:
        return 'National Day'
    elif dates in malaysia_day:
        return 'Malaysia Day'
    elif dates in maulidur_rasul:
        return 'Maulidur Rasul'
    elif dates in deepavali:
        return 'Deepavali'
    elif dates in christmas:
        return 'Christmas Day'
    else:
        return 'None'

# Apply the function
df['public_holiday'] = df['date'].apply(holiday_type)
df
```

Figure XXIX - Add Public Holiday column

Besides that, another crucial column to add is Public Holiday as shown in Figure XXIX. Though the data has to be manually added through hard coding as there is no external data source for it, but each dates have their own specific public holiday names as its values which helps to determine how different public holidays affect the overall COVID-19 cases. As of right now, the available public holiday is from 2020 to present year, 2025, hence public holidays for 2026 onwards might need human intervention for updates on the future dates inside the dataset. This feature is important because it could provide context to one of the models which is Prophet, that has its own holidays context for impact on the overall accuracy of prediction.

ate	state	new_cases	imported_cases	recovered_cases	active_cases	child_cases	adolescent_cases	adult_cases	elderly_cases	mco	monsoon_season	public_holiday
01-25	Malaysia	4	4	0	4	0	0	1	0	No	Northeast	Chinese New Year
01-25	Malaysia	0	0	0	4	0	0	0	0	No	Northeast	Chinese New Year
01-25	Malaysia	0	0	0	4	0	0	0	0	No	Northeast	None
01-25	Malaysia	0	0	0	4	0	0	0	0	No	Northeast	None
01-25	Malaysia	3	3	0	7	1	0	2	0	No	Northeast	None
...	...	...	...	...	...	...	...	...	...	...	...	...
01-15	W.P. Putrajaya	5	0	0	249	0	0	4	1	No	Northeast	None
01-22	W.P. Putrajaya	1	0	5	245	0	0	1	0	No	Northeast	None
01-22	W.P. Putrajaya	0	0	0	245	0	0	0	0	No	Northeast	None
01-22	W.P. Putrajaya	3	0	0	248	0	0	2	1	No	Northeast	None
01-22	W.P. Putrajaya	1	0	5	244	0	0	1	0	No	Northeast	None

Figure XXX - Output of Public Holiday Column

As shown in Figure XXX, the public holiday column has been successfully added with dates that has a public holiday is valued with the specific name of the holiday, whereas, dates without any public holiday is just labeled as “None” for easy identification.

```
# Define School holiday periods for multiple years with formatted dates
mid_year = set(pd.date_range(start='2020-05-23', end='2020-06-07').strftime('%d/%m/%Y').to_list() +
               pd.date_range(start='2021-05-29', end='2021-06-13').strftime('%d/%m/%Y').to_list() +
               pd.date_range(start='2023-04-22', end='2023-04-30').strftime('%d/%m/%Y').to_list())

term1 = set(pd.date_range(start='2020-03-14', end='2010-03-22').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2021-03-27', end='2011-04-04').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2022-06-04', end='2012-06-12').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2023-05-27', end='2013-06-04').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2024-05-25', end='2014-06-02').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2025-05-29', end='2015-06-09').strftime('%d/%m/%Y').to_list())

term2 = set(pd.date_range(start='2020-08-20', end='2020-08-24').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2021-07-17', end='2021-07-25').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2022-09-03', end='2022-09-11').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2023-08-26', end='2023-09-03').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2024-09-14', end='2024-09-22').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2025-09-13', end='2015-09-21').strftime('%d/%m/%Y').to_list())

term3 = set(pd.date_range(start='2021-09-11', end='2011-09-19').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2022-12-10', end='2012-12-31').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2023-12-16', end='2014-01-01').strftime('%d/%m/%Y').to_list() +
            pd.date_range(start='2024-12-21', end='2024-12-29').strftime('%d/%m/%Y').to_list())

year_end = set(pd.date_range(start='2020-12-19', end='2020-12-31').strftime('%d/%m/%Y').to_list() +
               pd.date_range(start='2021-12-11', end='2021-12-31').strftime('%d/%m/%Y').to_list() +
               pd.date_range(start='2023-02-18', end='2023-03-12').strftime('%d/%m/%Y').to_list() +
               pd.date_range(start='2024-02-10', end='2024-03-10').strftime('%d/%m/%Y').to_list() +
               pd.date_range(start='2025-01-18', end='2025-02-16').strftime('%d/%m/%Y').to_list())

# Function to determine school holiday type based on date
def scholiday_type(date):
    # Convert date to string format '%d/%m/%Y' for comparison
    date_str = date.strftime('%d/%m/%Y')
    if date_str in mid_year:
        return 'Mid-Year'
    elif date_str in term1:
        return 'Term 1'
    elif date_str in term2:
        return 'Term 2'
    elif date_str in term3:
        return 'Term 3'
    elif date_str in year_end:
        return 'Year-End'
    else:
        return 'None'

# Apply the function
df['school_holiday'] = df['date'].apply(scholiday_type)
df
```

Figure XXXI - Add School Holiday Column

To end the data preprocessing and preparation stage is by adding a similar feature just like public holiday which is the school holiday column as shown in Figure XXXI. The way it differs from one another is that public holidays involve all parties from children up to the elderly people, while school holiday only affects those who are still considered as students that are studying in the various forms of education, be it primary, secondary, or tertiary education infrastructures.

new_cases	imported_cases	recovered_cases	active_cases	child_cases	adolescent_cases	adult_cases	elderly_cases	mco	monsoon_season	public_holiday	school_holiday
4	4	0	4	0	0	1	0	No	Northeast	Chinese New Year	None
0	0	0	4	0	0	0	0	No	Northeast	Chinese New Year	None
0	0	0	4	0	0	0	0	No	Northeast	None	None
0	0	0	4	0	0	0	0	No	Northeast	None	None
3	3	0	7	1	0	2	0	No	Northeast	None	None
...	...	...	...	...	...	...	...	...	...	...	...
5	0	0	249	0	0	4	1	No	Northeast	None	Year-End
1	0	5	245	0	0	1	0	No	Northeast	None	Year-End
0	0	0	245	0	0	0	0	No	Northeast	None	Year-End
3	0	0	248	0	0	2	1	No	Northeast	None	Year-End
1	0	5	244	0	0	1	0	No	Northeast	None	Year-End

Figure XXXII - Output of School Holiday Column

As shown in Figure XXXII, similarly to the public holiday column, values of school holiday features have its distinct names to classify the different types of school holiday during each particular date, on the other hand, where when there is none, it will just be labeled as "None" to indicate the absence of any sort of school holiday for that date.

4.2 Short-Term Forecasting Model Development

After all data preparation steps are completed, now it's time for the model development and evaluation stage. This project features two forecasting goals which are short-term and long-term forecasting. Both Prophet and Holt's Winter model will be utilized in both forecasting goals and evaluated based on their prediction's accuracy and performance.

4.2.1 Prophet Model

```
!pip install prophet
from prophet import Prophet
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from dateutil.relativedelta import relativedelta
```

Figure XXXIII - Install Prophet Model libraries and packages

First and foremost is on the short-term forecasting model development with Prophet model. For Prophet, it is necessary to install the libraries and packages that are related to Prophet so its predefined features and functionalities could be utilized to the fullest as shown in Figure XXXIII.

```
holidays = pd.DataFrame({
    'holiday': [
        'CNY 2020', 'CNY 2021', 'CNY 2022', 'CNY 2023', 'CNY 2024', 'CNY 2025',
        'LD 2020', 'LD 2021', 'LD 2022', 'LD 2023', 'LD 2024', 'LD 2025',
        'WESAK 2020', 'WESAK 2021', 'WESAK 2022', 'WESAK 2023', 'WESAK 2024', 'WESAK 2025',
        'AF 2020', 'AF 2021', 'AF 2022', 'AF 2023', 'AF 2024', 'AF 2025',
        'YB 2020', 'YB 2021', 'YB 2022', 'YB 2023', 'YB 2024', 'YB 2025',
        'AD 2020', 'AD 2021', 'AD 2022', 'AD 2023', 'AD 2024', 'AD 2025',
        'MH 2020', 'MH 2021', 'MH 2022', 'MH 2023', 'MH 2024', 'MH 2025',
        'ND 2020', 'ND 2021', 'ND 2022', 'ND 2023', 'ND 2024', 'ND 2025',
        'MD 2020', 'MD 2021', 'MD 2022', 'MD 2023', 'MD 2024', 'MD 2025',
        'MR 2020', 'MR 2021', 'MR 2022', 'MR 2023', 'MR 2024', 'MR 2025',
        'DV 2020', 'DV 2021', 'DV 2022', 'DV 2023', 'DV 2024', 'DV 2025',
        'XMAS 2020', 'XMAS 2021', 'XMAS 2022', 'XMAS 2023', 'XMAS 2024', 'XMAS 2025'
    ],
    'ds': pd.to_datetime([
        '25/1/2020', '12/2/2021', '1/2/2022', '23/1/2023', '10/2/2024', '30/1/2025',
        '1/5/2020', '1/5/2021', '1/5/2022', '1/5/2023', '1/5/2024','1/5/2025',
        '7/5/2020', '26/5/2021', '16/5/2022', '4/5/2023', '22/5/2024','12/5/2025',
        '24/5/2020', '13/5/2021', '1/5/2022', '22/4/2023', '10/4/2024', '31/3/2025',
        '8/6/2020', '7/6/2021', '6/6/2022', '5/6/2023', '3/6/2024','2/6/2025',
        '31/7/2020', '20/7/2021', '10/7/2022', '29/6/2023', '17/6/2024','6/6/2025',
        '20/8/2020', '9/8/2021', '30/7/2022', '19/7/2023', '7/7/2024','26/6/2025',
        '31/8/2020', '31/8/2021', '31/8/2022', '31/8/2023', '31/8/2024','31/8/2025',
        '16/9/2020', '16/9/2021', '16/9/2022', '16/9/2023', '16/9/2024','16/9/2025',
        '29/10/2020', '19/10/2021', '9/10/2022', '28/9/2023', '16/9/2024','4/9/2025',
        '14/11/2020', '4/11/2021', '24/10/2022', '13/11/2023', '31/10/2024','20/10/2025',
        '25/12/2020', '25/12/2021', '25/12/2022', '25/12/2023', '25/12/2024','25/12/2025'
    ]),
    'lower_window': [0] * 72,  # Start on the holiday date
    'upper_window': [1,1,1,1,2,1,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,(
})
```

Figure XXXIV - Define Holiday context

Then, we should define the holiday context with holidays in Malaysia from the start of COVID-19 which is in 2020 up until 2025's holidays. Based on the figure XXXIV above, the holidays are defined into a data frame with 'holiday' as the type of holiday, 'ds' as the date and lower and upper window as the start of holiday date and effect of the holiday respectively.

```
# Filter for Malaysia's data
prophet_data = df[df['state'] == 'Malaysia']
prophet_data = prophet_data[['date', 'new_cases', 'mco', 'monsoon_season', 'public_holiday', 'school_holiday']]

# Rename columns as required by Prophet
prophet_data.rename(columns={"date": "ds", "new_cases": "y"}, inplace=True)

# Ensure the 'ds' column is in datetime format
prophet_data['ds'] = pd.to_datetime(prophet_data['ds'])
```

Figure XXXV - Prepare dataset for Prophet

After that, as shown in Figure XXXV, we need to prepare the dataset primarily for the use of Prophet model by filtering the data to only Malaysia without including its states,

renaming the date as ‘ds’ column and new cases as ‘y’ column as this is a requirement for Prophet to function accordingly. Then, just ensure the ‘ds’ column is in the right datetime format.

```
# One-hot encode the 'monsoon_season' column
prophet_data['Northeast'] = (prophet_data['monsoon_season'] == 'Northeast').astype(int)
prophet_data['Southwest'] = (prophet_data['monsoon_season'] == 'Southwest').astype(int)
prophet_data['Inter-Monsoon'] = (prophet_data['monsoon_season'] == 'Inter-Monsoon').astype(int)
```

Figure XXXVI - Add on Monsoon Season as additional regressor

Afterwards, we can also add on additional regressors such incorporating the Monsoon season column to further help in providing more depth into the forecasting later on as shown in Figure XXXVI. The term one hot encode basically means to sort each Monsoon season by themselves in its own dataframe to ensure clarity.

```
# Define the end date as the maximum date in the dataset
end_date = prophet_data['ds'].max()

# Calculate the start date as 5 months before the end date
start_date = end_date - relativedelta(months=5)

# Filter the data based on the calculated date range
prophet_data = prophet_data[(prophet_data['ds'] >= start_date) & (prophet_data['ds'] <= end_date)]
```

Figure XXXVII - Define date range for training and testing

Moving on, we should define the date range to suit the goal of this project which in this case is for short-term forecasting as illustrated in Figure XXXVII. Thus, the appropriate date range is set to during the 5 months before the latest date in the dataset and as such the start date is automatically selected.

```
model = Prophet(
    seasonality_mode='multiplicative',
    holidays=holidays,
    seasonality_prior_scale=10.0,  # Increase seasonality flexibility
    holidays_prior_scale=5.0,      # Increase holiday effect
    changepoint_prior_scale=0.5    # Allow larger changes in trend
)

# Add all three monsoon season columns as regressors
model.add_regressor('Northeast')
model.add_regressor('Southwest')
model.add_regressor('Inter-Monsoon')

# Fit the model
model.fit(prophet_data)

# Create a dataframe for future dates
future = model.make_future_dataframe(periods=30)

future['Northeast'] = prophet_data['Northeast'].iloc[-1]  # Set the last known value for future
future['Southwest'] = prophet_data['Southwest'].iloc[-1]
future['Inter-Monsoon'] = prophet_data['Inter-Monsoon'].iloc[-1]
```

Figure XXXVIII - Build Prophet Model along with holiday and monsoon season regressors

Next as displayed in Figure XXXVIII, it is very crucial to build the Prophet model by fitting it with the prepared dataset and appropriate parameters as this could drastically affect the output even if for the slightest mistake. On the same note, we could also include the additional regressor of Monsoon Seasons and feed it to the model to provide more context along with the predefined holidays. Once the model is created, then we could create a data frame for future dates which in this case it will predict 30 days into the future as the periods is set to 30.

```
# Make predictions
prof_forecast = model.predict(future)

# Post-process predictions
prof_forecast['yhat'] = prof_forecast['yhat'].round().clip(lower=0)
prof_forecast['yhat_lower'] = prof_forecast['yhat_lower'].round().clip(lower=0)
prof_forecast['yhat_upper'] = prof_forecast['yhat_upper'].round().clip(lower=0)

# Display predictions
print(prof_forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail())
```

Figure XXXIX - Define the prediction values

Afterwards, we could define the prediction values as illustrated in Figure XXXIX, by ensuring suitable columns are chosen for the forecasted values. Not to mention, the values are also rounded off so it will be in whole numbers and clipped to the lowest as 0 so there cannot be negative values as it does not make sense to have COVID-19 cases with values of either in decimals or negative. From the above figure, 'ds' represents the date column, 'yhat' represents the predicted values column, and 'yhat_lower' and 'yhat_upper' defines the upper and lower boundary values respectively.

```
# Length of the dataset
data_length = len(prophet_data)

# Define the split point (e.g., 80-20 split, so 80% for training and 20% for testing)
split_index = int(0.8 * data_length)  # 80% for training

# Split the data into train and test sets
train_data = prophet_data[:split_index]  # Use the first 80% of the data for training
test_data = prophet_data[split_index:]   # Use the last 20% of the data for testing

# Merge actual test data with predictions
results = test_data[['ds', 'y']].merge(prof_forecast[['ds', 'yhat']], on='ds')
```

Figure XL - Split Dataset for Prophet with 80/20 split

As we need to later on test the accuracy of the predicted values, it is a good practice to always split the dataset accordingly into training and testing set as displayed in Figure XL. To put it in simple terms, training set can be described as the data that will be provided to appropriately train the model so it will be capable of predicting values for the testing set without any help nor answers given directly. Common splitting method is done by either 70/30 or 80/20 split but for this project all models and forecasting goals will only utilize the standard 80/20 split so the model could learn more so it would be more knowledgeable during testing.

```
# Calculate evaluation metrics
prof_mae = mean_absolute_error(results['y'], results['yhat'])
prof_rmse = np.sqrt(mean_squared_error(results['y'], results['yhat']))
prof_r2 = r2_score(results['y'], results['yhat'])

# Calculate MAPE (handle division by zero for cases where y is zero)
results['abs_percentage_error'] = np.abs((results['y'] - results['yhat']) / results['y'].replace(0, np.nan))
prof_mape = results['abs_percentage_error'].mean() * 100

# Create a dictionary for metrics
metrics = {
    "Evaluation Metrics |": ["Mean Absolute Error (MAE) |", "Mean Absolute Percentage Error (MAPE) |", "Root Mean Squared Error (RMSE) |", "R-squared (
    "Value ": [prof_mae, prof_mape, prof_rmse, prof_r2]
}

# Convert the dictionary to a pandas DataFrame
metrics_df = pd.DataFrame(metrics)

# Add two blank lines before printing the table
print("\n")
print(metrics_df.to_string(index=False, float_format="{:.4f}".format))
print("\n")
```

Figure XLI - Define the evaluation metrics for testing Prophet predictions

Next is to define the evaluation metrics that will be utilized for evaluating the predicted values with actual values over the testing dataset as shown in Figure XLI. In this case, the evaluation metrics that would be used are MAE, MAPE, RMSE, and R^2 which will later on be explained in a much more detailed manner in section 4.4 Model Evaluation and Comparison.

```
# Plot the forecast
model.plot(prof_forecast)
plt.show()

# Plot the forecast components (trend, seasonality, etc.)
model.plot_components(prof_forecast)
plt.show()

# Plot actual vs predicted values (only for the test data)
plt.figure(figsize=(10, 6))
plt.plot(results['ds'], results['y'], label='Actual', marker='o')
plt.plot(results['ds'], results['yhat'], label='Predicted', marker='x')
plt.xlabel('Date')
plt.ylabel('New Cases')
plt.title('Actual vs Predicted Cases (Test Data)')
plt.legend()
plt.grid()
plt.show()
```

Figure XLII - Plot graph with Prophet components

Finally, just plot suitable graphs by incorporating the Prophet's built in plotting features and components. As shown in Figure XLII, the code will display a set of graphs/charts that provide significant insight individually.

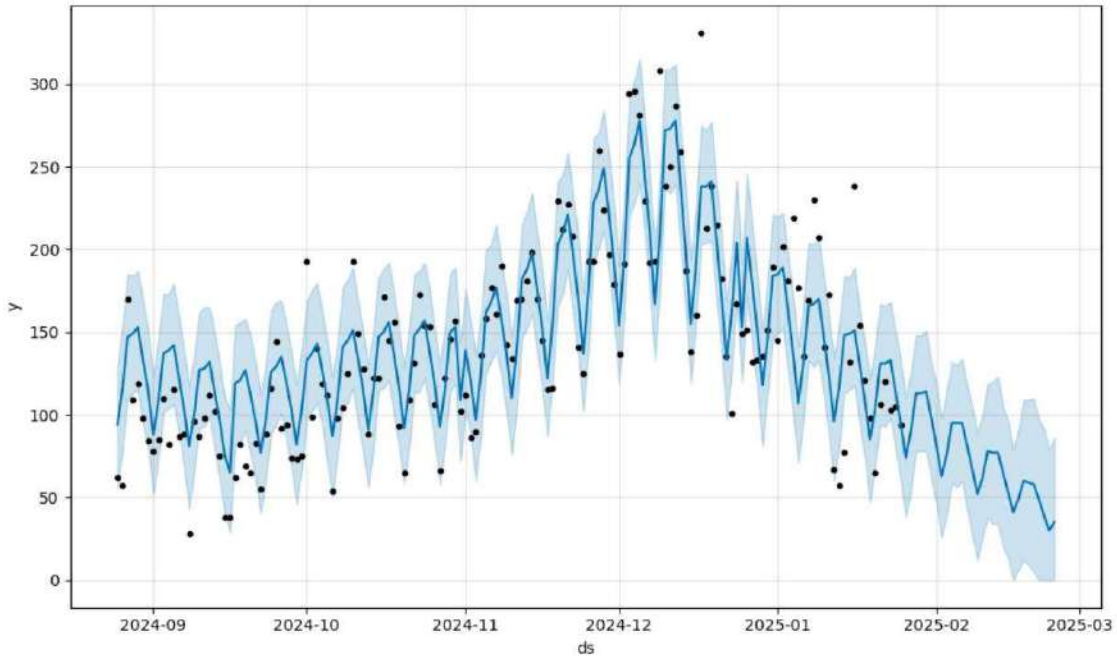

Figure XLIII – Prophet's Graph with forecasted values in 5 months duration

The first graph will feature the forecasted values using Prophet from the training data set period to the testing data set which is about 5 months duration and ultimately end with the future 30 days prediction in advance. As shown in Figure XLIII, the dark circles represent the actual values, the blue-coloured line represents the predicted values while the shaded blue areas are the upper and lower boundary values as a range for the predicted values.

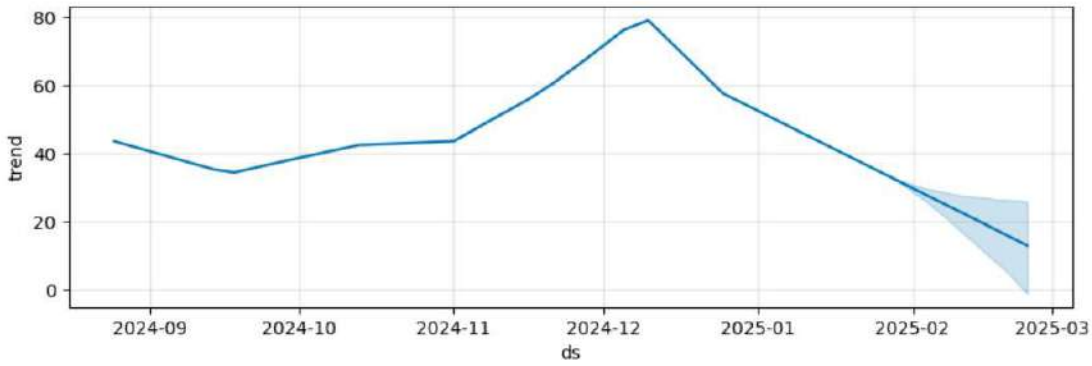

Figure XLIV – Prophet's Trend Plot in 5 months duration

The second graph as illustrated in Figure XLIV, is the trend plot of the forecasted cases which basically entails the overall trend during the 5 months duration which is shown to be quite similar with the actual values, hence it's a good indication. The blue shaded areas for this graph on the hand, means the predicted area with its values supposedly should only be around the area.

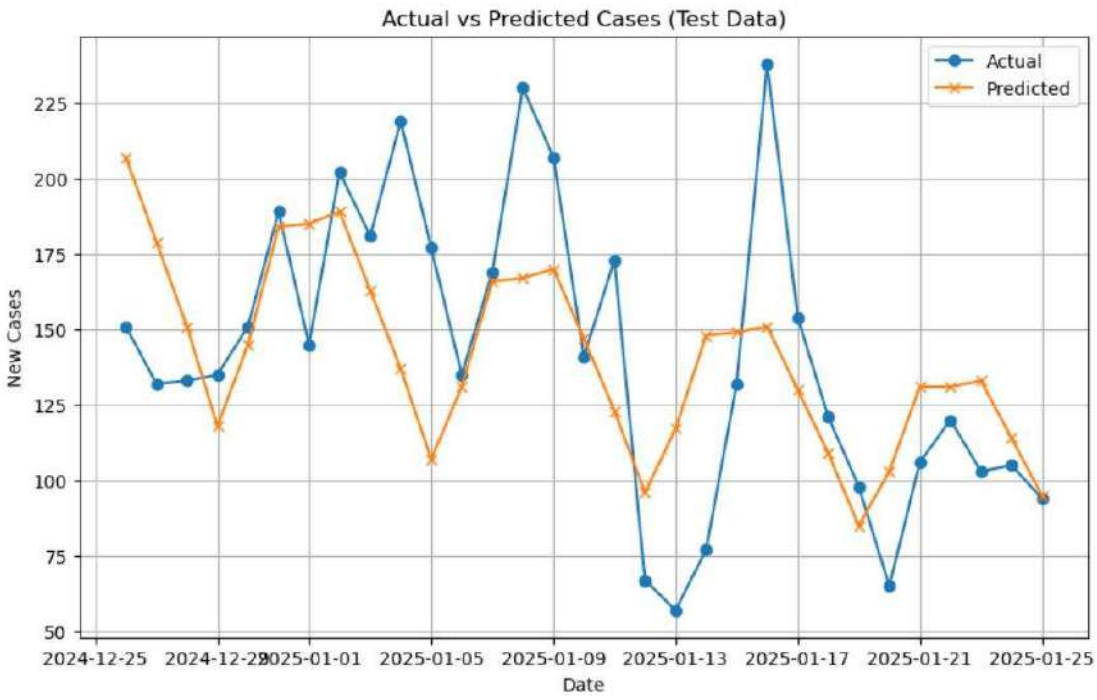

Figure XLV - Actual vs Predicted values of 1 month testing data

To end it off with the Prophet model is the final graph, Figure XLV, which shows the actual vs predicted values when compared in the test data which is 20% of the overall dataset, equivalent to 1 months of data for testing set. As we can see, since the actual values are fluctuating in an inconsistent manner, Prophet's predicts the values to maintain a consistent trend just to balance it out which indicates a lazy model of some sort.

4.2.2 Holt's Winter Model

```
!pip install pandas statsmodels matplotlib scikit-learn
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from math import sqrt
from sklearn.metrics import mean_absolute_error, r2_score
```

Figure XLVI - Install Holt's Winter model libraries and packages

Now moving on to Holt's Winter model, we should first install the necessary libraries and packages for Holt's winter to function accordingly just as the same when using Prophet as shown in Figure XLVI.

```
# Filter for Malaysia's data
hw_data = df[df['state'] == 'Malaysia']
hw_data = hw_data[['date', 'new_cases']]

# Define the end date as the maximum date in the dataset
end_date = hw_data['date'].max()
# Calculate the start date as 5 months before the end date
start_date = end_date - relativedelta(months=5)

# Filter the data based on the calculated date range
hw_data = hw_data[(hw_data['date'] >= start_date) & (hw_data['date'] <= end_date)]

# Ensure 'date' is parsed correctly and set as index
hw_data['date'] = pd.to_datetime(hw_data['date'])
hw_data.set_index('date', inplace=True)
hw_data = hw_data.asfreq('D')
```

Figure XLVII - Prepare Dataset for Holt's Winter model

Similarly to Prophet as well, it must have its own prepared dataset that fills in Holt's Winter requirements such as the frequency to be set as 'D' which means daily as we are predicting the values for every date. As featured in the above Figure XLVII, filtering the dataset to Malaysia only, defining the start and end date as 5 months before the latest data record to have appropriate date range are some examples of preparing the dataset for Holt's Winter.

```
# Split data into 80% training and 20% test
train_size = int(len(hw_data) * 0.8)
train, test = hw_data[:train_size], hw_data[train_size:]
```

Figure XLVIII - Split dataset for HW with 80/20 split

Moving on, we should split the dataset into training and testing set with an 80/20 ratio split. As shown in the Figure XLVIII above, the code will allow an automated way of splitting the data as per our requirements.

```
# Fit Holt-Winters Exponential Smoothing Model to training data
model = ExponentialSmoothing(train['new_cases'],
                             trend='mul',  # Options: 'add' or 'mul' for additive or multiplicative trend
                             seasonal='mul',  # Options: 'add' or 'mul' for seasonal effect
                             seasonal_periods=12)  # Set this to the number of periods in a season (e.g., 12 months)
fitted_model = model.fit()

# Forecast on the test data
forecast_steps = len(test)  # Forecast the same number of steps as the length of the test set
hw_forecast = fitted_model.forecast(steps=forecast_steps)

# Define the number of future steps you want to forecast (e.g., 30 days)
future_steps = 30  # Forecast the next 30 days
future_forecast = fitted_model.forecast(steps=future_steps)
```

Figure XLIX - Build Holt's Winter model along with its future steps to forecast

Afterwards, we should build the Holt's winter model by defining its trend, seasonal and periods parameters as shown in Figure XLIX. The seasonal period is set to 12 as for 12 months in a season while the values for trend and seasonal could be either additive or multiplicative. In this case, multiplicative is chosen because the fluctuating data of COVID-19 cases is at times higher than usual so multiplicative would be the best choice as it suitable for exponential growth, whereas additive is more towards steady growth over time. Then, we would define the future steps which basically means the number of future dates we want to predict on which is set to 30 days.

```
# Generate future dates for plotting (continuing from the last date of the current dataset)
last_date = hw_data.index[-1]  # Get the last date in the original data
future_dates = pd.date_range(start=last_date, periods=future_steps + 1, freq='D')[1:]  # Get future dates

# Adjust forecasted values to avoid negative values and decimals
hw_forecast = np.maximum(np.round(hw_forecast), 0)  # Round and set negative forecasts to zero
future_forecast = np.maximum(np.round(future_forecast), 0)  # Round and set negative forecasts to zero
```

Figure L - Define forecasted dates and values

Moving forward to Figure L, is to generate the future dates for forecasting the values with an appropriate date range and standardized values. Based on the above, this would enable forecasted values to remain as whole and positive values.

```
# Calculate performance metrics
hw_mae = mean_absolute_error(test['new_cases'], hw_forecast)

# MAPE Calculation
hw_mape = np.mean(np.abs((test['new_cases'] - hw_forecast) / test['new_cases'])) * 100  # MAPE in percentage

hw_rmse = sqrt(mean_squared_error(test['new_cases'], hw_forecast))  # RMSE calculation
hw_r2 = r2_score(test['new_cases'], hw_forecast)

# Create a dictionary for metrics
metrics = {
    "Evaluation Metrics |": ["Mean Absolute Error (MAE) |", "Mean Absolute Percentage Error (MAPE) |", "Root Mean Squared Error (RMSE) |", "R-squared (R²
    "Value ": [hw_mae, hw_mape, hw_rmse, hw_r2]
}

# Convert the dictionary to a pandas DataFrame
metrics_df = pd.DataFrame(metrics)

# Add two blank lines before printing the table
print("\n")
print(metrics_df.to_string(index=False, float_format="{:.4f}".format))
print("\n")
```

Figure LI - Define the evaluation metrics for HW model

Since we are splitting the dataset into training and testing set, it would be wise to define the evaluation metrics for Holt's Winter model which includes MAE, MAPE, RMSE and R^2 which is similar to prophet's so later on both could be compared without any bias or unfairness as presented in Figure LI.

```
# Plot the actual vs predicted values (including both training and test data, plus future predictions)
plt.figure(figsize=(12,6))
plt.plot(hw_data.index[:train_size], train['new_cases'], label='Training Data')
plt.plot(hw_data.index[train_size:], test['new_cases'], label='Actual Test Data', color='green')
plt.plot(hw_data.index[train_size:], hw_forecast, label='Forecasted Test Data', color='red')
plt.plot(future_dates, future_forecast, label='Forecasted Future Data', color='blue', linestyle='dashed')

plt.legend(loc='best')
plt.title('Holt-Winters Forecast of Daily COVID-19 Cases in Malaysia')
plt.xlabel('Date')
plt.ylabel('Confirmed Cases')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```

Figure LII - Plot the graph to compare actual and forecasted values

Finally, for the Holt's Winter model development stage is to plot the graphs and figures to showcase the results and compare the values to see the disparity between each of them as illustrated in Figure LII.

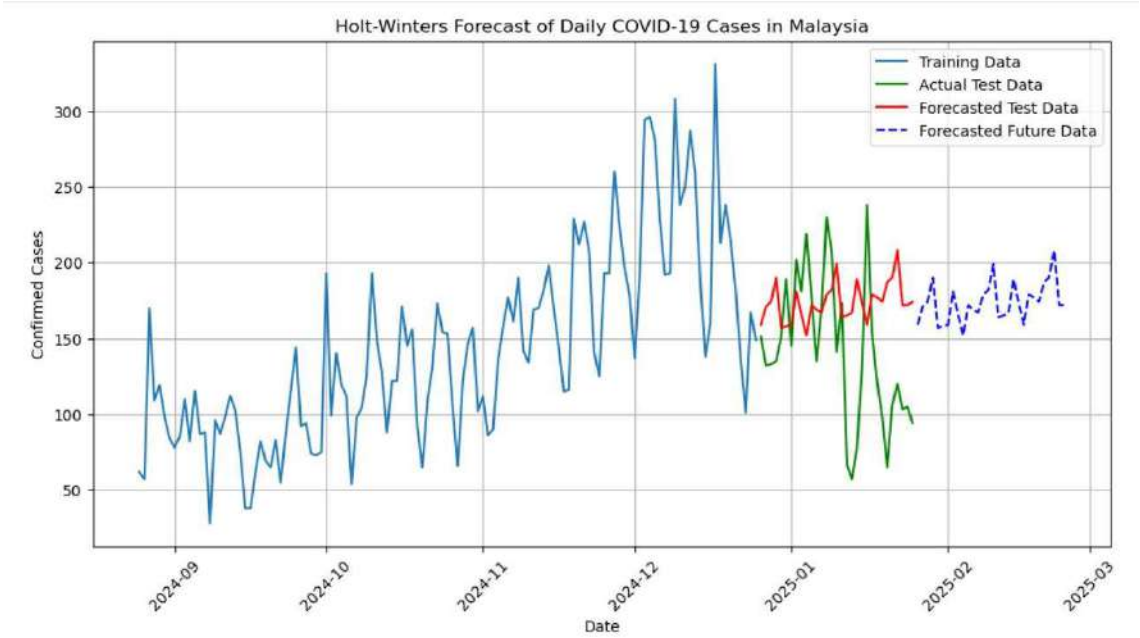

Figure LIII - Holt's Winter graph plot in 1 month duration

Only for Holt's Winter, the important graph to look at is the comparison of values between the actual and forecasted values, along with the distinct forecasted future data. The main difference when presenting visualizations between Prophet and Holt's Winter is that Prophet would also predict for both training and testing data while Holt's Winter would not even try to predict the training dataset as its only copies back from the original values. As we can see from the Figure LIII above, initially the forecasted test data values seems to be almost similar in terms of trend with the actual test values but eventually when the actual values drastically decrease over time, the model just assumes that it should follow the trend which shows its weaknesses for non-linear trends.

4.3 Long-Term Forecasting Model Development

Besides short-term forecasting, the other forecasting goal is emphasize more towards long-term forecasting which uses the entire dataset as a whole without leaving any date record behind.

4.3.1 Prophet Model

```
# Define the start and end date for the range
start_date = prophet_data['ds'].min()
end_date = prophet_data['ds'].max()

# Filter the data based on the date range
prophet_data = prophet_data[(prophet_data['ds'] >= start_date) & (prophet_data['ds'] <= end_date)]
```

Figure LIV - Reset the dataset date range for Prophet

For the most part, the code for Prophet model's long term forecasting model development is the same as the ones in the short-term forecasting goal but its only difference as shown in Figure LIV, lies within the date range of dataset. The start and end date are set to min and max date respectively, which means it will take the earlier date record and the latest date record for its definition. As such from 2020 to 2025 it will be around 5 years for the overall dataset with 80% of the data being training set which is 4 years while the other 20% of the data will be in the testing set which is approximately 30 days or 1 month to be exact.

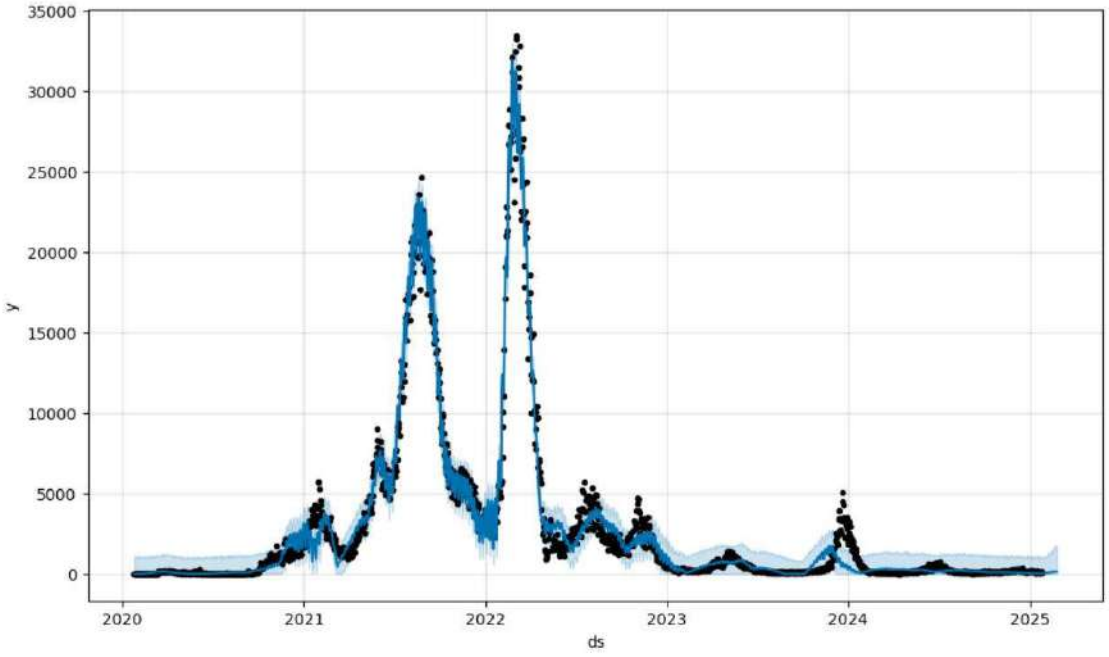

Figure LV – Prophet's graph with forecasted values in 5 years duration

For the graphs in long-term forecasting as shown in Figure LV, Prophet's will feature all of the 5 years duration of the dataset, and it shows an uncanny similarity for the initial 3 highest peak cases when compared actual vs predicted. The issues might lie on the data being overfitted, hence the values are almost similar.

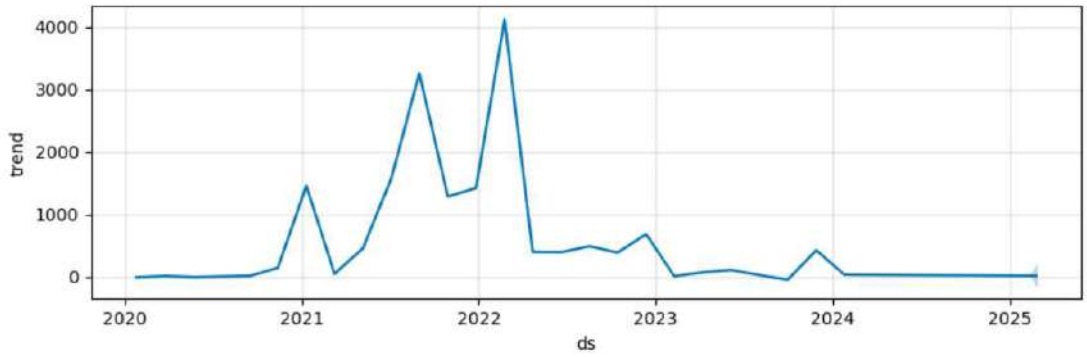

Figure LVI – Prophet's Trend Plot in 5 years duration

Besides that, for the trend plot as shown in Figure LVI, there is a resemblance of between the trend of cases with the actual values, though during the spike cases just before 2024 started, the spike of cases is predicted a bit earlier when compared with the actual cases. This means that the test data is in motion as the values are being tested, hence not quite similar with the actual ones.

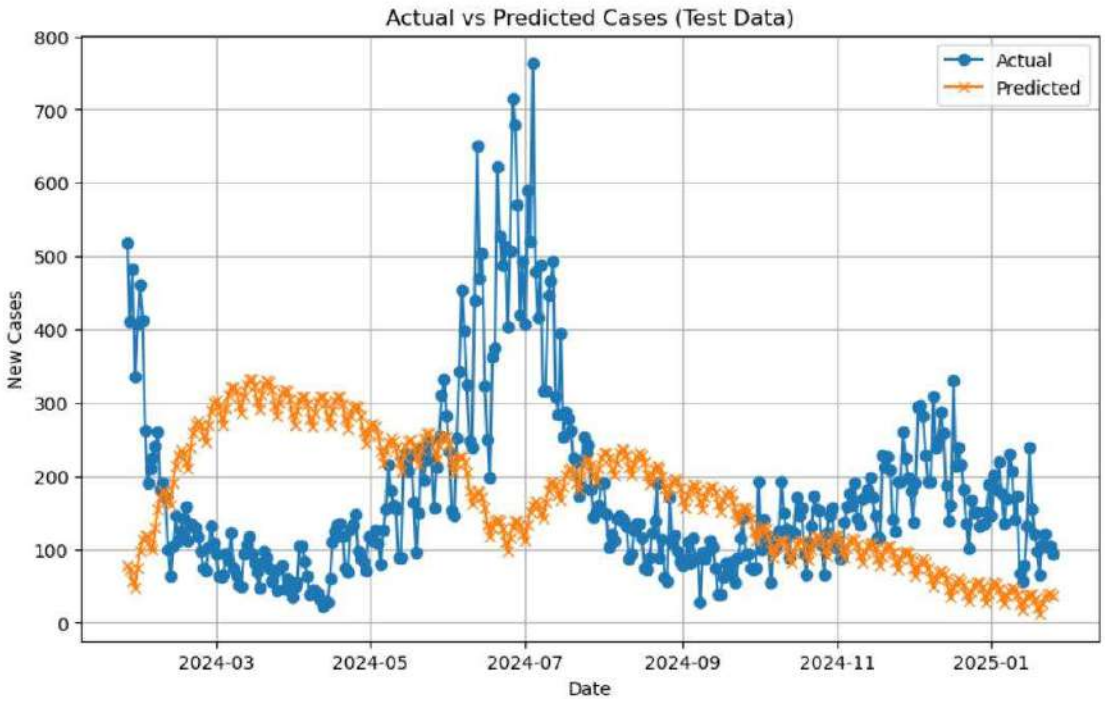

Figure LVII - Actual vs Predicted values of 1 year testing data

At the end of it, is the actual vs predicted values whereby we could see that during the testing set, it shows that the predicted values are way off when compared with the actual values as presented in Figure LVII. This indicates that this model may not quite suitable for this forecasting goal in general.

4.2.2 Holt's Winter Model

```
# Define the start and end date for the range
start_date = hw_data['date'].min()
end_date = hw_data['date'].max()

# Filter the data based on the calculated date range
hw_data = hw_data[(hw_data['date'] >= start_date) & (hw_data['date'] <= end_date)]
```

Figure LVIII - Reset the dataset date range for HW

Now for Holt's Winter model, for long-term forecasting, code-wise it is alike with the short-term forecasting but same like the Prophet's is the definition of start and end date to accommodate all data from the original dataset which is for the duration of 5 years as illustrated in Figure LVII.

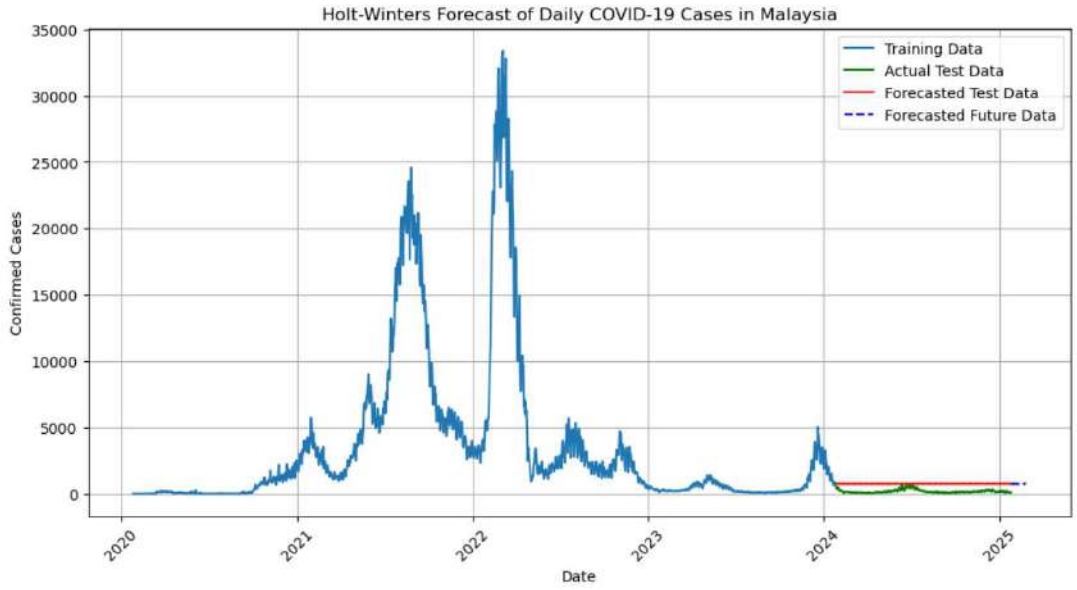

Figure LIX - Holt's Winter graph plot in 1 year duration

Last but not least, as illustrated in Figure LIX, the graph plot for Holt's winter for testing data of 1 year duration is shown to be incredibly bad as the forecasted values just stayed in the same values without any change in trend as it does not even try to forecast accordingly with the actual values.

4.4 Model Evaluation and Comparison

To effectively evaluate both models, Prophet and Holt's Winter, proper evaluation metrics must be applied on the testing set of the data. By comparing the predicted values with the actual values beforehand, it would definitely help in determining the predictive model's capabilities in forecasting cases for future dates. All in all, four evaluation metrics are used to measure the accuracy and performance of both models which consists of Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE) and R-Squared (R^2).

4.4.1 Mean Absolute Error (MAE)

For regression models particularly, MAE measures the average absolute difference between the predicted and actual values. This can be beneficial for comprehending error

magnitude without taking direction into account since it handles all mistakes equally without squaring them. It is highly interpretable, not easily affected by outliers and basically much simpler to calculate when compared with the other evaluation metrics (Ahmed, 2023). It can be calculated with the formula below:

$$\mathbf{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Based on the above formula, n represents the number of data points, while y_i and x_i represents the predictive and actual values respectively. Ideals scores for MAE would be values closer to 0, thus indicating a highly accurate model the closer the MAE score is to 0.

4.4.2 Mean Absolute Percentage Error (MAPE)

According to Roberts (2023), MAPE is almost similar to MAE whereby one could consider it as a percentage-based counterpart of the metric, in which it measures the average absolute percentage difference between the predicted and actual values. To put it simply, it calculates how far off predictions are on average, or the average magnitude of error a model produces. The formula is as follows:

$$M = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

Based on the formula above, n represents how many summation itterations that occurred, on the other hand A_t and F_t represents the actual and predicted values respectively. Just like MAE, MAPE also follows a similar scoring pattern for which lower values is equivalent to a better and accurate model while higher values describes an inaccurate model.

4.4.3 Root Mean Squared Error (RMSE)

RMSE is a crucial evaluation metric for machine learning and statistics in general where it quantifies the average number of errors to effectively assess the prediction accuracy as mentioned by Padhma (2024). Some might even call it as a Root Mean Square Deviation

(RMSD) metric. It is computed by calculating the square root of the squared errors and averaging them. Details of its formula is shown below:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}},$$

In this case, N represents the number of data points, y_i is the actual values and $\hat{}y_i$ is the predicted values. Similarly as the above mentioned metrics, an ideal model performance is shown by lower RMSE values, but more disparity towards the actual values is suggested from higher values. In contrast, RMSE can be influenced by other characteristics or features that may play a role during prediction which further evaluates the model fit.

4.4.4 R-Squared (R^2)

From the understanding of Fernando (2024), R^2 which ranges from 0 to 1, measures the degree to which independent factors account for variance in a dependent variable. In other words, from the range of 0 to 1, a perfect model is shown to have a value equal to 1 while a weaker predictive model is shown closer to 0. For this project, R_2 is significant, but it should still be interpreted in conjunction with other metrics to prevent drawing false conclusions, particularly when overfitting is present. The formula of R_2 derives from below:

$$R^2 = 1 - \frac{RSS}{TSS}$$

As shown in the abovementioned formula, RSS may be interpreted as unexplained variation or Residual's Sum of Square (RSS), while TSS could be described as simply total variation or Total Sum of Squares (TSS). To summarize, R^2 effectively explains the changes in

the dependent variable using the independent variables for which score closer to 1 is ideal whereas closer to 0 is consider unreasonable.

4.4.5 Short-Term Forecasting

Evaluation Metrics	Value
Mean Absolute Error (MAE)	31.0323
Mean Absolute Percentage Error (MAPE)	24.7899
Root Mean Squared Error (RMSE)	39.7719
R-squared (R^2)	0.3035

Figure LX - Prophet's Evaluation Metrics for short-term

Evaluation Metrics	Value
Mean Absolute Error (MAE)	51.2258
Mean Absolute Percentage Error (MAPE)	49.7001
Root Mean Squared Error (RMSE)	61.5750
R-squared (R^2)	-0.6694

Figure LXI - Holt's Winter Evaluation Metrics for short-term

According to the result for the short-term forecasting model, Prophet's evaluation metric as shown in figure LX is consider ideal for this project as it has lower MAE, MAPE and RMSE values while also having a higher score for R^2 as well when compared to Holt's Winter evaluation metrics score in figure LXI. Even though, the values for both models are not highly accurate in the sense of good prediction models for this project, Prophet's values would still be consider reasonable as the data itself contains various fluctuations which entirely oppose to the criteria of both models as they are not well suited to handle non-linear data and follow along certain trends without much consideration.

4.4.6 Full-Term Forecasting

Evaluation Metrics	Value
Mean Absolute Error (MAE)	134.8415
Mean Absolute Percentage Error (MAPE)	116.0965
Root Mean Squared Error (RMSE)	170.3894
R-squared (R^2)	-0.7969

Figure LXII - Prophet's Evaluation Metrics for long-term

Evaluation Metrics	Value
Mean Absolute Error (MAE)	566.6858
Mean Absolute Percentage Error (MAPE)	535.0372
Root Mean Squared Error (RMSE)	580.8874
R-squared (R^2)	-19.8849

Figure LXIII - Holt's Winter Evaluation Metrics for long-term

According to the result for the long-term forecasting model, Prophet's evaluation metric as shown in figure LXII is championed for this project as it has lower MAE, MAPE and RMSE values while also having a higher score for R^2 as well when compared to Holt's Winter evaluation metrics score in figure LXIII. In this scenario where both models tried to predict for longer periods, results have shown to be somewhat unpleasant as the fluctuations and variance in the dataset may have had heavily influced the overall predictions for this foreasting goal. Nonetheless, when comparing both models, Prophet is chose as the best amongst the two for the prediction model.

4.5 Short-Term Forecasting Model Selection

```
prophet_model = 0
holtswinter_model = 0

if prof_mae < hw_mae:
    prophet_model += 1
else:
    holtswinter_model += 1

if prof_mape < hw_mape:
    prophet_model += 1
else:
    holtswinter_model += 1

if prof_rmse < hw_rmse:
    prophet_model += 1
else:
    holtswinter_model += 1

if prof_r2 > hw_r2:
    prophet_model += 1
else:
    holtswinter_model += 1
```

Figure LXIV - Auto comparison of Model's Evaluation Metrics for short-term

At the end of the model development and evaluation stage, there is an automated model comparison which basically compares the evaluation metrics of both Prophet and Holt's Winter to assess the better model for the purpose of short-term forecasting as shown in Figure LXIV.

```
if prophet_model > holtswinter_model:
    # Create a new dataframe with date, actual cases, and predicted cases
    df3 = results[['ds', 'y', 'yhat']].copy()
    df4 = prof_forecast[['ds','yhat']].copy()

    # Rename the columns for better clarity
    df3.rename(columns={
        'ds': 'date',
        'y': 'actual_cases',
        'yhat': 'predicted_cases'
    }, inplace=True)

    df4.rename(columns={
        'ds': 'date',
        'yhat': 'future_cases'
    }, inplace=True)

    # Display the dataframe
    print("\nBest Model is Prophet!\n")
    print("Actual Cases vs Predicted Cases(Test Data):")
    print(df3)
    print("\nPredicted and Future Cases Over Time:")
    print(df4)
else:
   # Create a DataFrame for the test data with actual and predicted values
    df3 = pd.DataFrame({
        "date": test.index,  # Dates from the test set
        "actual_cases": test['new_cases'].values,  # Actual cases from the test set
        "predicted_cases": hw_forecast  # Forecasted cases for the test set
    }).reset_index(drop=True)  # Reset index to avoid duplication in the 'date' column

    # Create a DataFrame for future forecasted values (only date and predicted values)
    df4 = pd.DataFrame({
        "date": future_dates,  # Unique future dates
        "future_cases": future_forecast  # Forecasted cases for the future
    }).reset_index(drop=True)  # Reset index to avoid duplication in the 'date' column

    # Display the first few rows of each DataFrame to confirm
    print("\nBest Model is Holt's Winter!\n")
    print("Actual Cases vs Predicted Cases(Test Data):")
    print(df3)
    print("\nFuture Forecast Cases:")
    print(df4)
```

Figure LXV - Save datasets based on best model for short-term

As presented in Figure LXV, after the comparing process is completed, the predicted values and forecasted values for the best model will be saved into relevant dataframes called df3 and df4.

```
Best Model is Prophet!

Actual Cases vs Predicted Cases(Test Data):
         date  actual_cases  predicted_cases
0  2024-12-26           151            207.0
1  2024-12-27           132            179.0
2  2024-12-28           133            151.0
3  2024-12-29           135            118.0
4  2024-12-30           151            145.0
5  2024-12-31           189            184.0
6  2025-01-01           145            185.0
7  2025-01-02           202            189.0
8  2025-01-03           181            163.0
9  2025-01-04           219            137.0
10 2025-01-05           177            107.0
11 2025-01-06           135            131.0
12 2025-01-07           169            166.0
13 2025-01-08           230            167.0
14 2025-01-09           207            170.0
15 2025-01-10           141            147.0
16 2025-01-11           173            123.0
17 2025-01-12            67             96.0
18 2025-01-13            57            117.0
19 2025-01-14            77            148.0
20 2025-01-15           132            149.0
21 2025-01-16           238            151.0
22 2025-01-17           154            130.0
23 2025-01-18           121            109.0
24 2025-01-19            98             85.0
25 2025-01-20            65            103.0
26 2025-01-21           106            131.0
27 2025-01-22           120            131.0
28 2025-01-23           103            133.0
29 2025-01-24           105            114.0
30 2025-01-25            94             95.0

Predicted and Future Cases Over Time:
          date  future_cases
0   2024-08-25          94.0
1   2024-08-26         116.0
2   2024-08-27         147.0
3   2024-08-28         149.0
4   2024-08-29         153.0
..         ...           ...
179 2025-02-20          58.0
180 2025-02-21          49.0
181 2025-02-22          39.0
182 2025-02-23          30.0
183 2025-02-24          35.0

[184 rows x 2 columns]
```

Figure LXVI - Result for comparison of short-term forecasting models

According to the results in Figure LXVI, Prophet model is the ideal model based on its performance when comparing its test data applied with evaluation metrics scoring. This means that Prophet model will be the chosen predictive model for short-term forecasting inside the dashboard.

4.6 Long-Term Forecasting Model Selection

```
if prophet_model > holtswinter_model:
    # Create a new dataframe with date, actual cases, and predicted cases
    df5 = results[['ds', 'y', 'yhat']].copy()
    df6 = prof_forecast[['ds','yhat']].copy()

    # Rename the columns for better clarity
    df5.rename(columns={
        'ds': 'date',
        'y': 'actual_cases',
        'yhat': 'predicted_cases'
    }, inplace=True)

    df6.rename(columns={
        'ds': 'date',
        'yhat': 'future_cases'
    }, inplace=True)

    # Display the dataframe
    print("\nBest Model is Prophet!\n")
    print("Actual Cases vs Predicted Cases(Test Data):")
    print(df5)
    print("\nPredicted and Future Cases Over Time:")
    print(df6)
else:
   # Create a DataFrame for the test data with actual and predicted values
    df5 = pd.DataFrame({
        "date": test.index,  # Dates from the test set
        "actual_cases": test['new_cases'].values,  # Actual cases from the test set
        "predicted_cases": hw_forecast  # Forecasted cases for the test set
    }).reset_index(drop=True)  # Reset index to avoid duplication in the 'date' column

    # Create a DataFrame for future forecasted values (only date and predicted values)
    df6 = pd.DataFrame({
        "date": future_dates,  # Unique future dates
        "future_cases": future_forecast  # Forecasted cases for the future
    }).reset_index(drop=True)  # Reset index to avoid duplication in the 'date' column

    # Display the first few rows of each DataFrame to confirm
    print("\nBest Model is Holt's Winter!\n")
    print("Actual Cases vs Predicted Cases(Test Data):")
    print(df5)
    print("\nFuture Forecast Cases:")
    print(df6)
```

Figure LXVII - Save datasets based on best model for long-term

On the other hand, for long-term forecasting, the coding section is just the same with short-term forecasting, but the best model's predicted values are saved into different dataframes called df5 and df6 accordingly as indicated in Figure LXVII.

```
Best Model is Prophet!

Actual Cases vs Predicted Cases(Test Data):
          date  actual_cases  predicted_cases
0   2024-01-26           518             78.0
1   2024-01-27           411             72.0
2   2024-01-28           482             56.0
3   2024-01-29           335             48.0
4   2024-01-30           407             75.0
..         ...           ...              ...
361 2025-01-21           106             25.0
362 2025-01-22           120             34.0
363 2025-01-23           103             39.0
364 2025-01-24           105             40.0
365 2025-01-25            94             36.0

[366 rows x 3 columns]

Predicted and Future Cases Over Time:
          date  future_cases
0    2020-01-25           0.0
1    2020-01-26           0.0
2    2020-01-27           0.0
3    2020-01-28           0.0
4    2020-01-29           0.0
...         ...           ...
1853 2025-02-20         141.0
1854 2025-02-21         143.0
1855 2025-02-22         140.0
1856 2025-02-23         132.0
1857 2025-02-24         127.0

[1858 rows x 2 columns]
```

Figure LXVIII - Result for comparison of long-term forecasting models

According to the results in Figure LXVIII, Prophet model is the ideal model based on its performance when comparing its test data applied with evaluation metrics scoring. This means that Prophet model will also be the chosen predictive model for long-term forecasting inside the dashboard. All in all, this further emphasize on Prophet model's capabilities to effectively predict values in an accurate and reliable manner.

4.7 Chapter Summary

In summary Chapter 4 explained on the steps for the whole Model Development and Evaluation phase which involves the data preprocessing and preparation, developing models for both short-term and long-term forecasting, evaluating the performance of the models with 4 appropriate evaluation metrics, and conducting model comparison. At the end of it, the ideal model for the project was determined which is Prophet model for both short-term and long-term forecasting purposes.

CHAPTER 5

DASHBOARD DESIGN AND DEVELOPMENT

5.0 Overview

This project will ultimately conclude with Chapter 5 as the final stage on Dashboard Design and Development which further describes on the software requirements needed for this project, the process moving from Model Development to Dashboard Development and the finalized dashboard design and features.

5.1 Software Requirements

For the success of this project, the first required software is Jupyter Notebook. It is an open-sourced web tool that can be downloaded for free. In essence, it enables users to make and share notebooks including code, text, and other data. Currently, it is quite a popular and renowned software for the purpose of data science and analytics. In the case of my project, I have used Jupyter Notebook with python as its primary programming language to build and develop the predictive model and generate the datasets for the use of a visualization tool such as a dashboard.

The second required software would be Tableau. It is a very useful visualization and analytical tool that allows users to analyze data and solve problems particularly for many businesses and large corporations alike. Not to mention, it could also be beneficial for businesses by providing them the competitive edge of gathering insights in hopes to make better decisions through the use of data. In the case of my project, I have used Tableau to present my data in graphical formats in appealing dashboards.

5.2 Jupyter Notebook

```
import os

# Save updated CSV
df.to_csv(os.path.join(os.getcwd(), 'covid19_processed_data.csv'), index=False)
df3.to_csv(os.path.join(os.getcwd(), 'covid19_shortterm_predicted_data.csv'), index=False)
df4.to_csv(os.path.join(os.getcwd(), 'covid19_shortterm_future_data.csv'), index=False)
df5.to_csv(os.path.join(os.getcwd(), 'covid19_longterm_predicted_data.csv'), index=False)
df6.to_csv(os.path.join(os.getcwd(), 'covid19_longterm_future_data.csv'), index=False)
print("The Covid 19 Datasets have been updated.")
```

```
The Covid 19 Datasets have been updated.
```

Figure LXIX - Auto-save and generated datasets

After the models have been compared and evaluated, an ideal model will emerge, and its predictive values will saved into specific dataframes. As shown in Figure LXIX, the main dataset, df and the other datasets related to prediction model and its values are saved into csv files with its own specific file names. Basically it will auto generate and create new datasets, but if the datasets had been created before, then it will simply update them accordingly.

```
# Open Tableau workbook
workbook_path = os.path.join(os.getcwd(), "Covid19_Dashboard.twb")

if os.path.exists(workbook_path):
    os.startfile(workbook_path)
    print(f"Tableau workbook Covid19_Dashboard.twb opened.")
else:
    print(f"Workbook Covid19_Dashboard.twb not found!")
```

```
Tableau workbook Covid19_Dashboard.twb opened.
```

Figure LXX - Auto Open the Tableau workbook dashboard

Then, the Covid19_Dashboard.twb file will also automatically opened up as shown in Figure LXX. This means that the tableau workbook which contains the dashboards for this project will open by itself once the datasets have been created and saved.

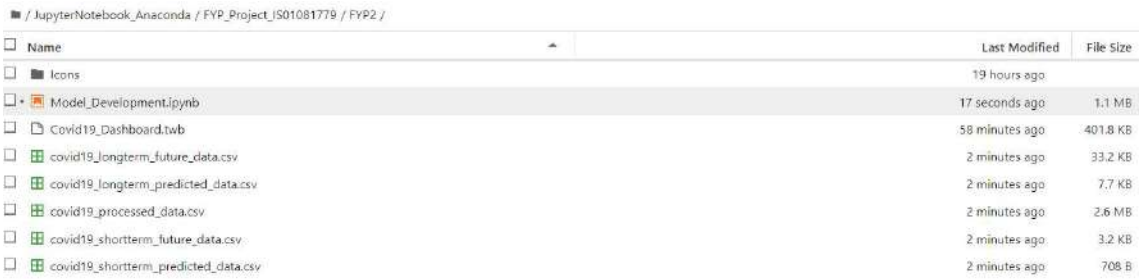

Figure LXXI - Final Outlook Files/Folders in Project Folder

When we look at back into the main FYP2 folder, as depicted in the Figure LXXI, there are five additional csv files that have just been created as the one of the final outputs for the Model_Development.ipynb file. These five datasets will be used for the development of dashboards in Tableau.

5.3 Tableau Software

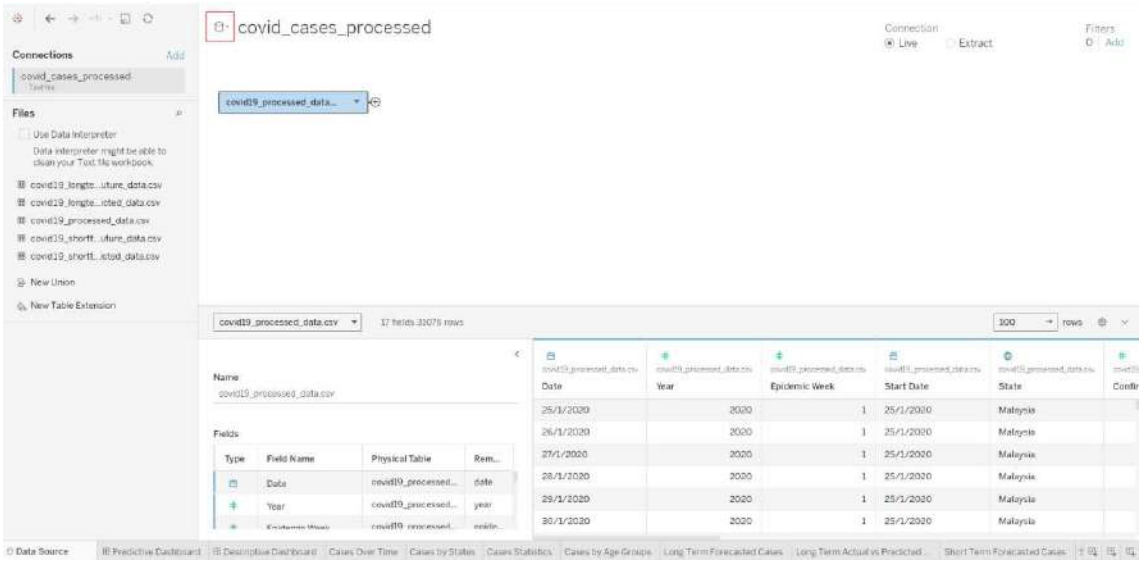

Figure LXXII - Data Source View in Tableau Workbook

Once the Covid19_Dashboard has finished loading up, we will be directed to the data source tab where we can the datasets from the earlier FYP2 project folder are already connected. In this tab, we can view all of the connections from the datasets involved along with their features, rows and values in an organized and neat tabulated formats. As highlighted in the red box in Figure LXXII, we can press that dropdown button to check the other datasets that are connected with the tableau workbook as well.

Figure LXXIII - Different tabs in Tableau Workbook

As presented in Figure LXXIII, there are many tabs in this tableau workbook which consists of a data source, two dashboard namely predictive dashboard and descriptive dashboard and the other worksheets which contains certain graphs.

5.4 Predictive Dashboard

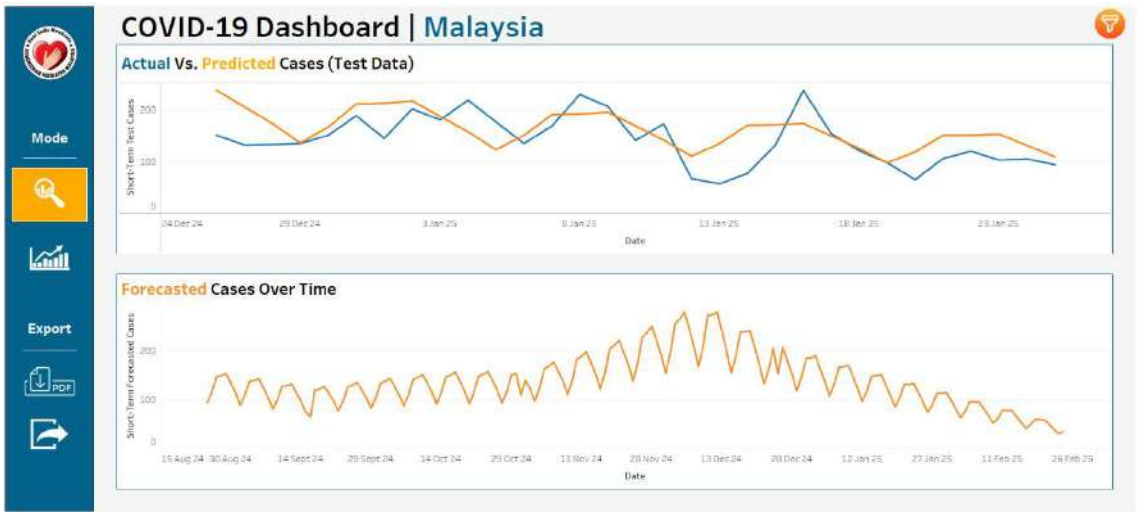

Figure LXXIV - Predictive Dashboard Short-Term Forecasting View

When we click on the predictive dashboard tab, we will be redirected to the actual dashboard itself. The Figure LXXIV above shows the view of the predictive dashboard, with two main graphs for Actual vs Predicted Cases based on test data and a forecasted cases over time graph. The Actual vs Predicted Cases graph depicts the comparison of values in the test data which is about 1 month duration. While, in the forecasted cases over time graph, the date of duration is about 5 months which includes both the training and testing data for predicted values only along with 30 days in advance of future cases. At the moment, these two graphs are by default set as short-term forecasting graphs. On the left side, we could see a blue navigation bar which features two modes, predictive dashboard and descriptive dashboard. Additionally, there is also an export feature where we could export the dashboard into pdf or image formats. When we clicked on the orange filter button on the top left corner, a filter bar will pop up at the side of the dashboard.

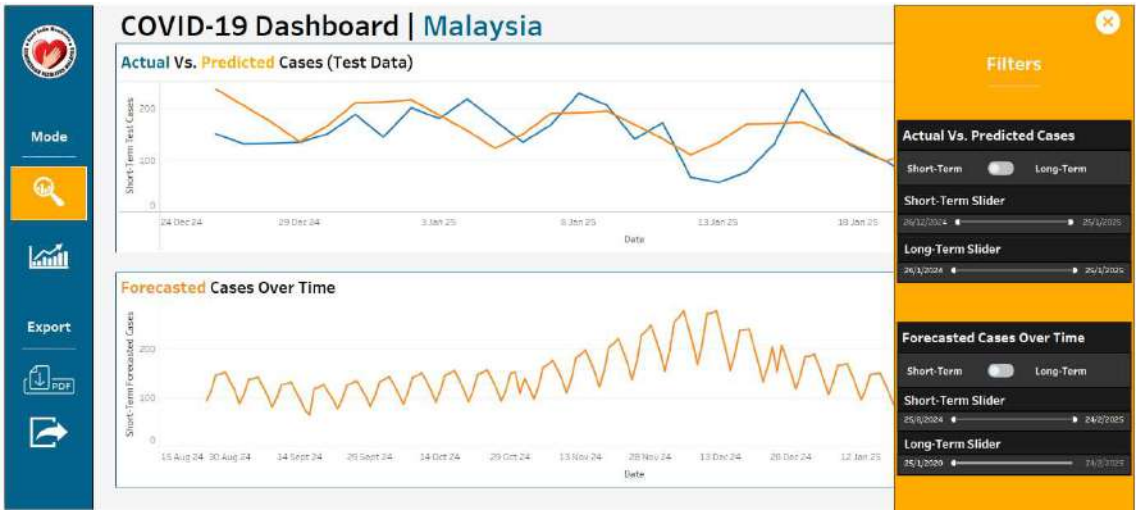

Figure LXXV - Predictive Dashboard Open Filter Bar

Figure LXXV above shows the filter bar popped up at the side of the dashboard overlapping on top of the graphs. Both of the graphs in this predictive dashboard can be filtered according to one's own preference. For instance, there are toggle buttons which would change the view of the graph from short-term to long-term. Not to mention, there is also sliders to control the date range of both the short-term and long-term of each individual graphs.

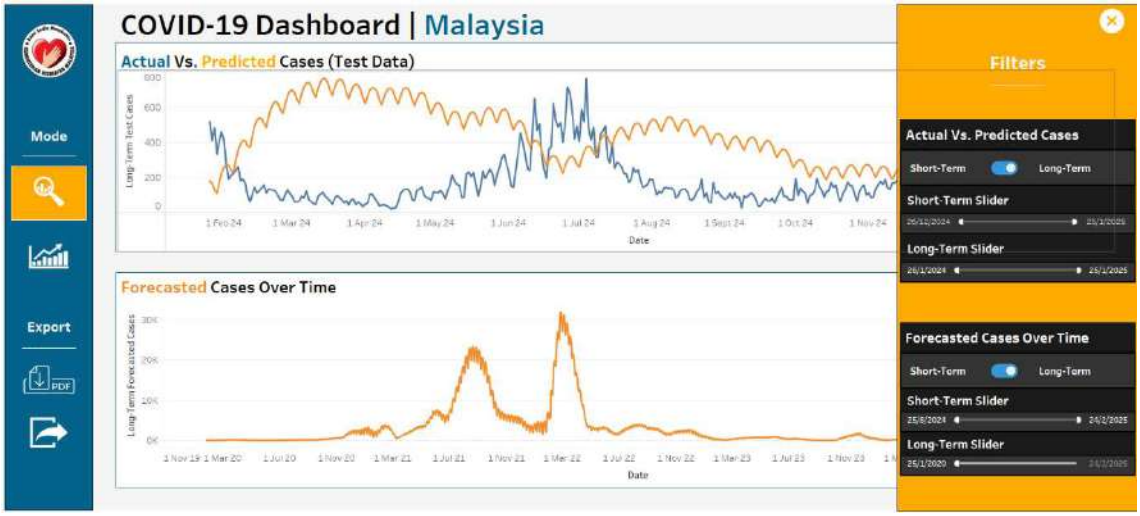

Figure LXXVI - Predictive Dashboard Long-Term Forecasting Graphs

Figure LXXVI shows that when clicking the toggle buttons, both of the graphs will immediately change according to the specified criteria of forecasting terms which in this

scenario is long-term as the duration for the upper graph changed to 1 year while the duration for the lower graph changed to 5 years duration.

Figure LXXVII - Predictive Dashboard Individual Graph's sliders

On the other hand, Figure LXXVII shows that when dragging the sliders, the specific graph based on its particular forecasting terms will change according to the defined date range so we could focus on specific dates or time periods. Moving forward, when we click on the descriptive dashboard mode which is located at the navigation bar under the orange box with magnifying glass, a different dashboard will appear.

5.5 Descriptive Dashboard

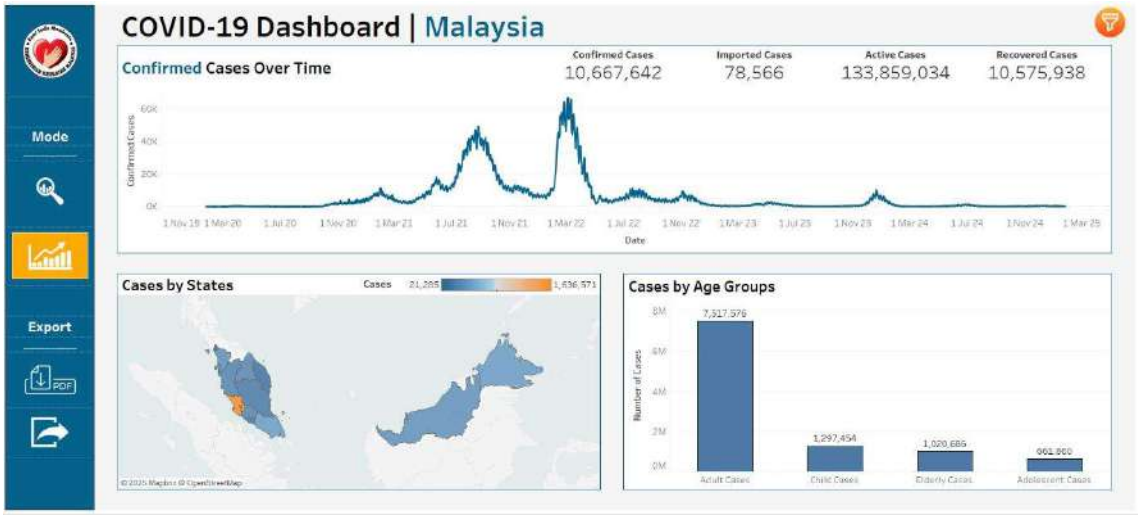

Figure LXXVIII - Descriptive Dashboard View

The above Figure LXXVIII presents on the descriptive dashboard view. The first graph to look at is the Confirmed Cases Over Time graph which just shows the daily COVID-19 cases from the earliest date record to the latest date of record. Along with it, there are also statistics related to COVID-19 cases in terms of the total Confirmed Cases, Imported Cases, Active Cases and Recovered Cases as of the current date. To put it simply, confirmed cases represents the cumulative value of new COVID-19 cases that emerged, imported cases represents cases that were imported from other countries to Malaysia, Active Cases represents the current cases that are still at going and the recovered cases represents the total number of recoveries from COVID-19 cases. The second graph is the Cases by States which showcase worldmap of Malaysia with its respective states. Each states have different colours, indicating the number of cases for each of them where the orange ones are higher in cases while the blue ones are lower in cases. The third graph is on the Cases by Age Groups which basically displays the parties affected by the COVID-19 cases with adult cases accounts for the highest leading number of COVID-19 cases at the moment. Other than that, design-wise the descriptive dashboard is almost similar with the predictive dashboard with the exception of the filters that are applied.

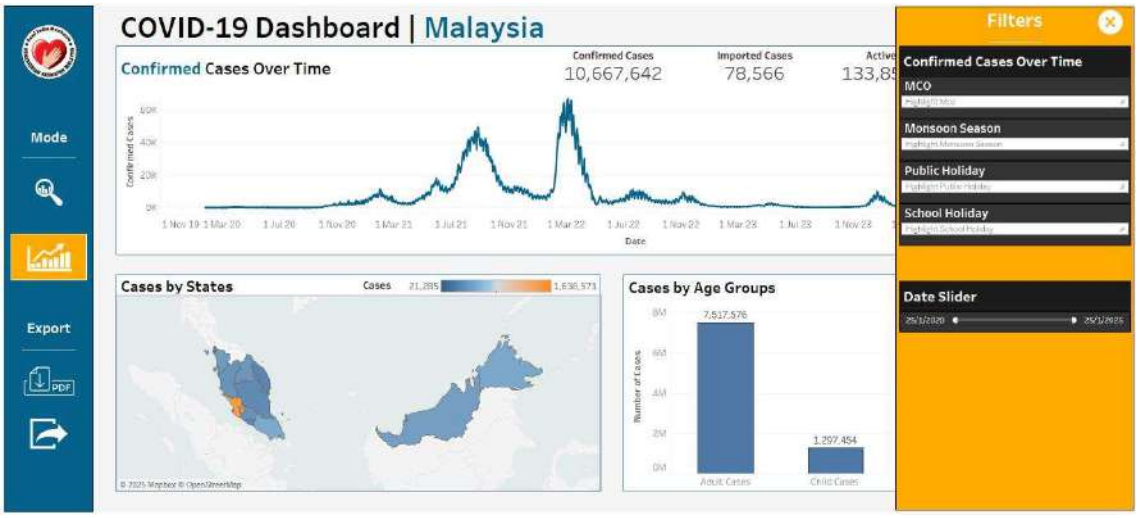

Figure LXXIX - Descriptive Dashboard Open Filter Bar

Just like the predictive dashboard, when we click on the orange filter at the top left corner, a filter bar will pop up at the side but with different set of filters as shown in the figure LXXIX above. For the Confirmed Cases Over Time graph, crucial features in the datasets such as MCO, Monsoon Season, Public Holiday and School Holiday could be highlighted

respectively on the graph. On top of that, there is also a date slider can influence all of the graphs and statistics in this dashboard.

Figure LXXX - Descriptive Dashboard Highlight Features

Figure LXXX shows the highlighted feature of Southwest monsoon season, where the graph will reduce the opacity for other monsoon seasons that are not involved only focusing on the selected monsoon season.

Figure LXXXI - Descriptive Dashboard Date Slider

Figure LXXXI depicts how the date slider when dragged to a specific date range could influence the total number for each statistics, the duration and trend of the Confirmed Cases

Over Time graph, colours on the Cases by States map and the values of the Cases by Age Groups graph.

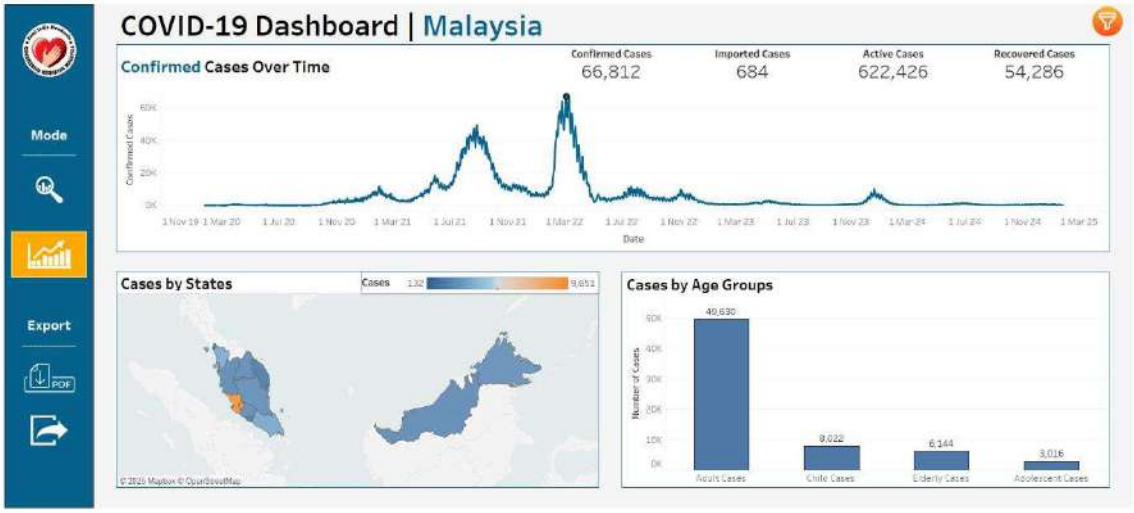

Figure LXXXII - Descriptive Dashboard Filter Cases by Specific Date

Other than that, there are also other filters besides the ones in the filter bar. As displayed on Figure LXXXII, when we click on any specific date on the Confirmed Cases Over Time graph, the values and statistics for the other graphs would also change accordingly. This is because the Confirmed Cases Over Time graph itself acts as a filter.

Figure LXXXIII - Descriptive Dashboard Filter Cases by States

The final filter is for the Cases by States map, which when clicked at any of the states, it will focus on cases particularly on that selected state only rather than cases in Malaysia altogether.

5.6 Chapter Summary

Chapter 5 marks the end of the CRISP-DM methodology and its designated phases. To summarize, Chapter 5 discusses on the specific software requirements to be used for the succession of this project which are Jupyter Notebook with the coding section for Model Development and Tableau Software with its seamless visualization capabilities of displaying key graphs and insights into respective predictive and descriptive dashboards. Apart from that, there is also the continuous explanation on Jupyter Notebook's expected outcomes which consists of five CSV dataset files and the opening of a tableau workbook. In regards to the Tableau workbook, further elaboration and justification for the data sources and dashboard choices are discussed to ensure clarity and understanding of its features and the important functionalities implemented for a user.

CHAPTER 6

CONCLUSION

6.0 Project 1 Outcome

In summary, this Final Year Project 1 is concluded with the successful analysis of descriptive analytics with six statistics that clearly indicates the factors that may affect the rate of sudden COVID-19 outbreaks. Essentially, this project has certainly been a learning experience to delve further into the field of analytics and understand the domain of interest with much more detail. Final Year Project 2 will move towards to the implementation of the modeling phase as we start to develop the dashboard to present and interpret the visuals while also building predictive models that could withstand various forecasting tasks with high accuracy, precision and performance.

6.1 Project 2 Outcome

In a nutshell, this Final Year Project 2 comes at end with the achievement of developing a predictive model that is fairly accurate and decent performance-wise in both short-term and long-term forecasting goals. On top of that, the successful development of a comprehensive and visual appealing dashboards which clearly captures the essence of the dataset transforming it into meaningful insights. One of the striking features of the dashboard when comparing it with the current COVID-19 dashboard in Malaysia is the implementation of varied analytical modes which includes a predictive analytics and descriptive analytics view inside their own distinct dashboards. In terms of predictive analytics, the ability to forecast COVID-19 cases up to 30 days into the future will be beneficial for Malaysian Ministry of Health (MOH). Whereas in regards to descriptive analytics, the dashboard also includes additional features such as MCO, Monsoon season, Public holiday and School holiday which are highlighted and

emphasized towards each date since 2020 up to the present. Thus, this project seems to be doing well as it manages to achieve its targeted aim and objectives to a great extent.

6.2 Project Strengths

One of the main strengths of this project revolves around its numerous auto features utilized in many sections and parts of the system/dashboard. First and foremost, is on the Jupyter Notebook's function to auto extract datasets from the website containing the data sources for this project without any intermissions from manually downloading the datasets every time which all in all save's times and effort. Secondly, is on the function to auto add columns/features into the main dataset from start date to end which is just simply efficient because it enhances the quality, clarity and provide additional context to certain data or insights. Thirdly, is on the feature to auto save the output datasets for the use of the dashboard based on the results of the best model when comparing between the two machine learning algorithms. Last but not least, is a simple auto feature to auto open the Tableau workbook once the required datasets have been generated and connected accordingly with the dashboards.

Another strength to look at is the implementation of the different types of analytical modes and goals These analytical modes and goals include a predictive dashboard that could also display the visualizations for both short-term and long-term forecasting and also a detailed and thorough descriptive dashboard that involves various features to provide better insight in hopes to view the data in a broader perspective and also conduct in-depth analysis.

6.3 Project Limitations/Weaknesses

With all of the strengths, benefits and competencies towards the project, there are also its limitations and weaknesses that occurred throughout the project. Firstly, due to the lack of available external datasets for the features, some of them I had to manually insert them with appropriate coding. For instance, since there are no reliable weather features, I had to settle with monsoon season as an alternative by doing an educated assumption for which certain months will result in the specific monsoon season. Other examples would be for the both the public holiday and school holidays features as I had to manually add on the particular holiday

based on its differing dates from 2020 to 2025, hence holidays for 2026 onwards would need some human intervention in manually adding the values for the future dates.

Despite its numerous automatic features and functionalities, the system which consists of the Jupyter Notebook file and Tableau Workbook, might not be portable in a way where users could easily use it. Due to the specific nature of Tableau which by default will set up the connection to the datasets according to their absolute path. This means that other users might have to reconnect the datasets in the dashboard by themselves as Tableau may not find where the datasets are placed as it follows my PC's file directory path. Hence, it might result in unnecessary errors to appear when users decide to test the system in their own PCs.

The last limitation would be due to my own inexperience with the selected machine learning algorithms to properly predict the values in the ideal and best manner. Both Prophet model and Holt's Winter model despite being easy to utilize, would still require some extensive knowledge and expertise in to fully utilized them to their utmost potential. As a result, the accuracy of the predictions might not achieve the expected result of a highly accurate predictive model as reference to its decent but not quite good enough evaluation metrics scores.

6.4 Suggestions for Future Improvements

My first suggestion would be to improve the user friendliness aspects of the system whereby users could easily access the dashboard with just a single click just like a fully functional application or a website rather than by going through the backend in Jupyter Notebook.

My second suggestion is to find effective ways to increase the accuracy of prediction such as by properly utilizing the parameters to the full extent and incorporating additional features and regressors into the mix in hopes to achieve the ideal evaluation metrics score that indicates an almost perfect predictive model.

My third and final suggestion is to search for free and easily accessible datasets which could become additional features for the main dataset without having to rely much on human intervention but rather automatically generated features. This would ensure the dataset would still be relevant for future uses as a catalyst in predicting various other infectious diseases in Malaysia.

7.0 REFERENCES

Ahmed, M. W. (2023, August 24). *Understanding Mean Absolute Error (MAE) in Regression: A Practical Guide*. Retrieved from Medium: https://medium.com/@m.waqar.ahmed/understanding-mean-absolute-error-mae-in-regression-a-practical-guide-26e80ebb97df

Bajaj, A. (2023, August 18). *ARIMA & SARIMA: Real-World Time Series Forecasting*. Retrieved from neptune.ai: https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide

CDC. (2023, July 10). *About COVID-19*. Retrieved from Centers for Disease Control and Prevention: https://www.cdc.gov/coronavirus/2019-ncov/your-health/about-covid-19.html

CDC. (2024, March 15). *Symptoms of COVID-19*. Retrieved from Centers for Disease Control and Prevention: https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html

Cennimo, D. J. (2024, April 1). *Coronavirus Disease 2019 (COVID-19)*. Retrieved from Medscape: https://emedicine.medscape.com/article/2500114-overview?form=fpf

ChildFund Australia. (2024, March 13). *What is the difference between an infectious and non-infectious disease?* Retrieved from ChildFund Australia: https://www.childfund.org.au/stories/what-is-the-difference-between-an-infectious-and-non-infectious-disease/#:~:text=What%20is%20a%20non-infectious,%2C%20malnutrition%2C%20environment%20and%20lifestyle.

Chimmula, V. R., & Zhang, L. (2020). Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos, Solitons & Fractals*.

Cleveland Clinic. (2022, June 6). *Infectious Diseases*. Retrieved from Cleveland Clinic: https://my.clevelandclinic.org/health/diseases/17724-infectious-diseases

Dattani, S., Spooner, F., Ritchie, H., & Roser, M. (2019). *Causes of Death*. Retrieved from Our World in Data: https://ourworldindata.org/causes-of-death#:~:text=In%20the%20past%2C%20infectious%20diseases,common%20causes%20of%20death%20globally.

Fernando, J. (2024, November 13). *R-Squared: Definition, Calculation, and Interpretation*. Retrieved from Investopedia: https://www.investopedia.com/terms/r/r-squared.asp

Hashim, J. H., Mohammad Adam Adman, Zailina Hashim, Mohd Firdaus Mohd Radi, & Soo Chen Kwan. (2021). COVID-19 Epidemic in Malaysia: Epidemic Progression, Challenges, and Response. *Frontiers in public health*, 9, 560592.

Hasri, H., Aris , S. M., & Ahmad, R. (2021). Linear Regression and Holt's Winter Algorithm in Forecasting Daily Coronavirus Disease 2019 Cases in Malaysia: Preliminary Study. *2021 IEEE National Biomedical Engineering Conference (NBEC)*, 157-160.

Hayes, A. (2024, April 5). *Autoregressive Integrated Moving Average (ARIMA) Prediction Model*. Retrieved from Investopedia: https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

Helou, M. V. (2021). Factors influencing the occurrence of infectious disease outbreaks in Lebanon since the Syrian crisis. *Pathogens and Global Health*, 13-21.

Houlihan, C. F. (2019). Outbreak science: recent progress in the detection and response to outbreaks of infectious diseases. *Clinical medicine (London, England)*, 140–144. Retrieved from https://doi.org/10.7861/clinmedicine.19-2-140

Kaur, D. (2020, March 17). *125 new Covid-19 cases — 95 linked to tabligh gathering*. Retrieved from The Malayisan Reserve: https://themalaysianreserve.com/2020/03/17/125-new-covid-19-cases-95-linked-to-tabligh-gathering/

Kumar, S. L., M, V. R., & L, J. A. (2021). Predictive Analytics of COVID-19 Pandemic: Statistical Modelling Perspective. *Walailak Journal of Science and Technology (WJST)*.

L., C. D. (2004). Major factors affecting the emergence and re-emergence of infectious diseases. *Clinics in laboratory medicine*, 559-586.

Li, H., Shang-Ming Liu, Xiao-Hua Yu, Shi-Lin Tang, & Chao-Ke Tang. (2020). Coronavirus disease 2019 (COVID-19): current status and future perspectives. *International journal of antimicrobial agents*, 55(5).

Mayo Clinic. (2022, February 18). *Infectious diseases*. Retrieved from Mayo Clinic: https://www.mayoclinic.org/diseases-conditions/infectious-diseases/symptoms-causes/syc-20351173

Mayo Clinic. (2024, March 5). *Germs: Understand and protect against bacteria, viruses and infections*. Retrieved from Mayo Clinic: https://www.mayoclinic.org/diseases-conditions/infectious-diseases/in-depth/germs/art-20045289#:~:text=Understanding%20infection%20versus%20disease&text=Infection%2C%20often%20the%20first%20step,symptoms%20of%20an%20illness%20appear.

Moghadas, S. M. (2020). The impact of vaccination on COVID-19 outbreaks in the United States. *medRxiv*.

Nath, D., Sasikumar, K., Chen, W., & Nath, R. (2021). Factors Affecting COVID-19 Outbreaks across the Globe: Role of Extreme Climate Change. *Sustainability 2021*.

Padhma. (2024, November 21). *A Comprehensive Introduction to Evaluating Regression Models*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/#h-mean-squared-error-mse

Painter, K. (2023, November 21). *Can You Get COVID Twice?* Retrieved from WebMD: https://www.webmd.com/covid/can-you-get-covid-twice

Quinn, G. A., Connolly, M., Fenton, N. E., Hatfill, S. J., Hynds, P., ÓhAiseadha, C., . . . Connolly, R. (2024). Influence of Seasonality and Public-Health Interventions on the COVID-19 Pandemic in Northern Europe. *Journal of Clinical Medicine*.

Rahulhegde. (2024, Jnauary 31). *Time Series Analysis using Facebook Prophet*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/time-series-analysis-using-facebook-prophet/

Roberts, A. (2023, February 2). *Mean Absolute Percentage Error (MAPE): What You Need To Know* . Retrieved from Arize: https://arize.com/blog-course/mean-absolute-percentage-error-mape-what-you-need-to-know/

Saba, T., Abunadi, I., Shahzad, M. N., & Khan, A. R. (2021). Machine learning techniques to detect and forecast the daily total COVID-19 infected and deaths cases under different lockdown types. *Microscopy research and technique*, 1462–1474.

Satu, M. S., Rahman, M. K., Rony, M. A., Shovon, A. R., Adnan, M. J., Howlader, K. C., & Kaiser, M. S. (2021). COVID-19: Update, Forecast and Assistant - An Interactive Web Portal to Provide Real-Time Information and Forecast COVID-19 Cases in Bangladesh. *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, 456-460.

Shahir Asfahan, M. G. (2020). Using a simple open-source automated machine learning . *Via Medica*, 400-405.

Singhal, T. (2020). A Review of Coronavirus Disease-2019 (COVID-19). *The Indian Journal of Pediatrics*, pg 281–286.

SolarWinds. (2019, December 15). *Holt-Winters Forecasting and Exponential Smoothing Simplified*. Retrieved from orangematter: https://orangematter.solarwinds.com/2019/12/15/holt-winters-forecasting-simplified/

World Health Organization (WHO). (2020, February 11). *Naming the coronavirus disease (COVID-19) and the virus that causes it*. Retrieved from World Health Organization: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it

World Health Organization (WHO). (2023, September 16). *Noncommunicable diseases*. Retrieved from World Health Organization (WHO): https://www.who.int/news-room/fact-sheets/detail/noncommunicable-diseases

World Health Organization (WHO). (n.d.). *Disease outbreaks*. Retrieved from World Health Organization (WHO): https://www.emro.who.int/health-topics/disease-outbreaks/index.html

World Heath Organization (WHO). (2020, December 8). *How do vaccines work?* Retrieved from World Heath Organization (WHO): https://www.who.int/news-room/feature-stories/detail/how-do-vaccines-work#:~:text=Vaccines%20contain%20weakened%20or%20inactive,rather%20than%20the%20antigen%20itself.