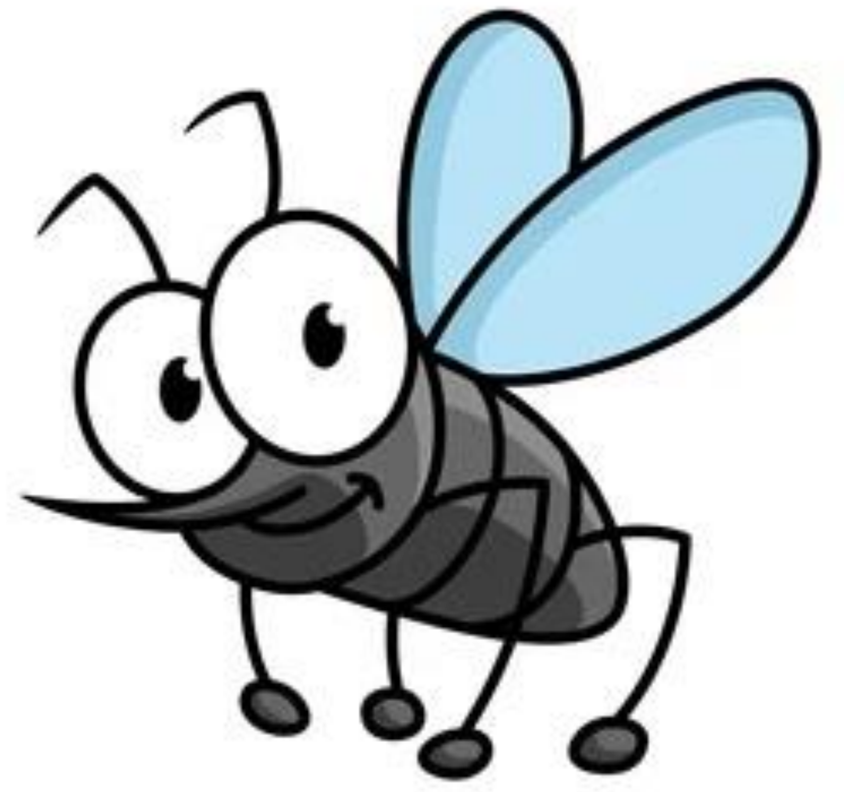


Intelligent Dengue Predictor

Project Report

ISY5001 Intelligent Reasoning Systems: Practice Module

Team Members:

- BAI SIHAI / A0229986E
- LU ZHIPING / A0229959E
- MEN JINLONG / A0229972N
- NG KWEE BOON / A0010308B

Contents

Figures

1 Executive Summary

According to the World Health Organization (WHO), dengue is a mosquito-borne viral infection typically found in tropical and sub-tropical climates worldwide and occurring mostly in urban and semi-urban areas.[1]

Forty percent of the world's population (about 3 billion people) live in areas with dengue. Dengue is often a leading cause of illness in these areas. Each year, up to 400 million people are infected. Approximately 100 million of them will get sick from the infection, and 22,000 will die from severe dengue.[2] 70% of the cases occur in Asia.[1]

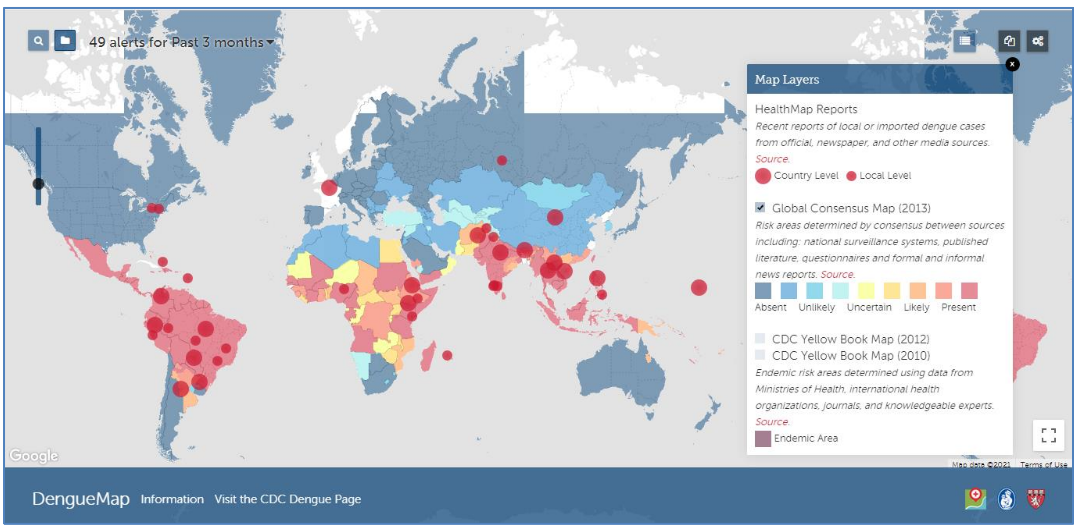


Figure 1. Dengue Map (2013) [3]

Singapore experienced the largest dengue outbreak in 2020. The DenV-3 dengue serotype is likely to be the cause, with 35,315 people infected[4] and 32 deaths[5].

---

[1] World Health Organization - Dengue and severe dengue. https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue (Accessed 28 Apr 2021)
[2] Centers for Disease Control and Prevention (CDC) - About Dengue: What You Need to Know. https://www.cdc.gov/dengue/about/index.html (Accessed 28 Apr 2021)
[3] HealthMap - Dengue Map. https://www.healthmap.org/dengue/en/ (Accessed 28 Apr 2021)
[4] The Straits Times - Dengue still a danger as Aedes mosquito population in Singapore grows even as cases fall. https://www.straitstimes.com/singapore/health/dengue-still-a-danger-as-aedes-mosquito-population-in-singapore-grows-even-as-cases (Accessed 28 Apr 2021)
[5] The Straits Times - Record 32 deaths from dengue last year. https://www.straitstimes.com/singapore/environment/record-32-deaths-from-dengue-last-year (Accessed 28 Apr 2021)

The ability to accurately forecast the number of people in a given area that will be infected with dengue is of great value to public health officials, physicians, and ultimately anyone at risk of infection. For example, Public health officials will be able to deploy preventive measures within potential danger zones with greater accuracy based on forecasted numbers. More efficient deployment of resources can also conserve resources for other needs.

We would like to help in this global mission to prevent dengue by building an Intelligent Dengue Predictor application. The application will provide early and accurate dengue case forecasts to public health officials. This is achieved by applying Machine learning techniques to find the correlations between weather conditions and the resulting dengue cases.

Key facts from World Health Organization[1]

- Dengue is a mosquito-borne viral infection.
- The virus responsible for causing dengue, is called dengue virus (DENV). There are four DENV serotypes, meaning that it is possible to be infected four times.
- Severe dengue is a leading cause of serious illness and death in some Asian and Latin American countries. It requires management by medical professionals.
- While many DENV infections produce only mild illness, DENV can cause an acute flu-like illness. Occasionally this develops into a potentially lethal complication, called severe dengue.
- There is no specific treatment for dengue/severe dengue. Early detection of disease progression associated with severe dengue, and access to proper medical care lowers fatality rates of severe dengue to below 1%.
- Dengue is found in tropical and sub-tropical climates worldwide, mostly in urban and semi-urban areas.
- The global incidence of dengue has grown dramatically in recent decades. About half of the world's population is now at risk. There are an estimated 100-400 million infections each year.
- Dengue prevention and control depends on effective vector control measures.
- Sustained community involvement can improve vector control efforts substantially.

2 Background

2.1 Business Case

The Intelligent Dengue Predictor is a highly viable product. Dengue has huge economic impact and covers large geographic areas worldwide. The Intelligent Dengue Predictor can also be adapted to forecast other diseases and trends.

2.1.1 Huge economic impact

Dengue causes an estimated global economic burden of US$8.9 billion per year, of which 40% was due to productivity costs.[6]

In Singapore, the economic burden is estimated to range between US$0.85 billion and US$1.15 billion per year (from 2000 to 2009). 41 per cent to 58 per cent of the estimated costs are due to direct medical, direct non-medical and indirect costs such as productivity loss. The remainder is the cost of vector control efforts. [7]

Region	Economic burden
Global	US$8.9 billion
Singapore	US$0.85 billion and US$1.15 billion per year (from 2000 to 2009)

Figure 2. Economic burden

Productivity costs are a significant proportion of the estimated economic burden. However, they are not directly comparable across studies, because the methods used to calculate them are often very variable.[6]

[6] BioMed Central - Productivity costs from a dengue episode in Asia: a systematic literature review. https://bmcinfectdis.biomedcentral.com/articles/10.1186/s12879-020-05109-0 (Accessed 28 Apr 2021)

[7] The Straits Times - Act against dengue now with tools that exist. https://www.straitstimes.com/singapore/environment/act-against-dengue-now-with-tools-that-exist (Accessed 28 Apr 2021)

2.1.2 Large geographical spread

Before 1970, only 9 countries had experienced severe dengue epidemics. The disease is now endemic in more than 100 countries in the WHO regions of Africa, the Americas, the Eastern Mediterranean, South-East Asia and the Western Pacific. The America, South-East Asia and Western Pacific regions are the most seriously affected, with Asia representing approximately 70% of the global burden of disease.[1]

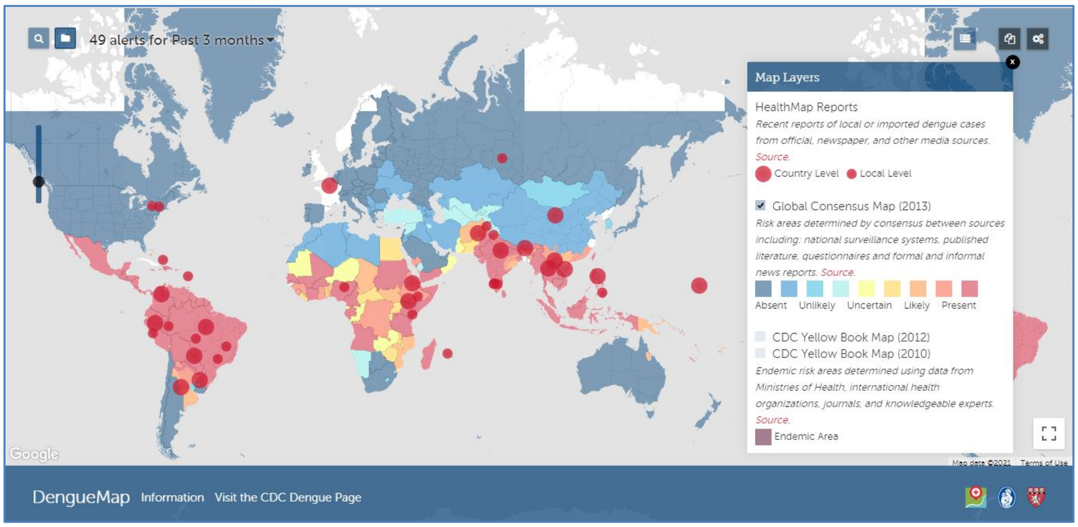


Figure 3. Dengue Map (2013) [3]

Forty percent of the world's population (about 3 billion people) live in areas with dengue. Dengue is often a leading cause of illness in these areas.[2]

Therefore, there are very large geographic areas worldwide, where our application can be deployed.

2.1.3 Severity of dengue disease

Each year, up to 400 million people are infected. Approximately 100 million of them will get sick from the infection, and 22,000 will die from severe dengue.[2]

Singapore experienced the largest dengue outbreak in 2020. The DenV-3 dengue serotype is likely to be the cause, with 35,315 people infected[8] and 32 deaths[9].

Region	Number of cases	Number of deaths
Global	Up to 400 million	22,000
Singapore	35,315 (in 2020)	32 (in 2020)

Figure 4. Number of cases and deaths

Most people infected have mild or no symptoms. About 1 in 4 people infected with dengue will get sick. Mild symptoms of dengue may be confused with other illnesses that cause fever and flu-like illness. Most people will recover after about one week.[10]

The most common symptoms of dengue are fever and one or more of the following symptoms: headache; eye pain (typically behind the eyes); muscle, joint, or bone pain; rash; nausea and vomiting; or unusual bleeding (nose or gum bleed, small red spots under the skin, or unusual bruising).[10]

However, severe dengue can result in shock, internal bleeding, and even death. Severe dengue patients have the following symptoms: Severe stomach pain or vomiting (at least 3 vomiting episodes within 24 hours); bleeding from the nose or gums; vomiting blood or blood in the stool; drowsiness or irritability; pale, cold, or clammy skin; difficulty breathing.[10] **Error! Bookmark not defined.**

---

[8] The Straits Times - Dengue still a danger as Aedes mosquito population in Singapore grows even as cases fall. https://www.straitstimes.com/singapore/health/dengue-still-a-danger-as-aedes-mosquito-population-in-singapore-grows-even-as-cases (Accessed 28 Apr 2021)

[9] The Straits Times - Record 32 deaths from dengue last year. https://www.straitstimes.com/singapore/environment/record-32-deaths-from-dengue-last-year (Accessed 28 Apr 2021)

[10] Centers for Disease Control and Prevention (CDC) - Dengue Virus: What you need to know. https://www.cdc.gov/dengue/resources/factsheets/factsheet_dengue-what-you-need-to-know.pdf (Accessed 28 Apr 2021)

2.1.4 Wide applicability to other diseases

The Intelligent Dengue Predictor application is applicable for many other diseases as detailed in the following sub-sections.

2.1.4.0 Other mosquito-transmitted diseases

Our application is not only applicable to dengue, it can also be used for other mosquito-transmitted diseases such as Zika virus, West Nile virus, Chikungunya virus, and malaria.

Weather conditions affects the breeding rates of mosquitoes, and the subsequent transmission rate of mosquito-transmitted diseases.

Our application analyzes the correlations between weather conditions and the resulting dengue cases. Therefore, it can be used to predict other mosquito-transmitted diseases, by training the machine learning algorithm with the data for other mosquito-transmitted diseases.

2.1.4.1 Other vector-borne diseases

The Intelligent Dengue Predictor can be adapted to predicted other vector-borne diseases, so long as the root conditions are measurable.

Vectors are living organisms that can transmit infectious pathogens between humans, or from animals to humans. Many of these vectors are bloodsucking insects, which ingest disease-producing microorganisms during a blood meal from an infected host (human or animal) and later transmit it into a new host, after the pathogen has replicated. Often, once a vector becomes infectious, they are capable of transmitting the pathogen for the rest of their life during each subsequent bite/blood meal. [11]

Vector-borne diseases are human illnesses caused by parasites, viruses and bacteria that are transmitted by vectors. Every year there are more than 700,000

[11] World Health Organization - Vector-borne diseases. https://www.who.int/news-room/fact-sheets/detail/vector-borne-diseases (Accessed 28 Apr 2021)

deaths from diseases such as malaria, dengue, schistosomiasis, human African trypanosomiasis, leishmaniasis, Chagas disease, yellow fever, Japanese encephalitis and onchocerciasis. [11]

The following table is a non-exhaustive list of vector-borne disease from WHO. [11]

Vector		Disease caused	Type of pathogen
Mosquito	*Aedes*	Chikungunya	Virus
		Dengue	Virus
		Lymphatic filariasis	Parasite
		Rift Valley fever	Virus
		Yellow Fever	Virus
		Zika	Virus
	Anopheles	Lymphatic filariasis	Parasite
		Malaria	Parasite
	Culex	Japanese encephalitis	Virus
		Lymphatic filariasis	Parasite
		West Nile fever	Virus
Aquatic snails		Schistosomiasis (bilharziasis)	Parasite
Blackflies		Onchocerciasis (river blindness)	Parasite
Fleas		Plague (transmitted from rats to humans)	Bacteria
		Tungiasis	Ectoparasite
Lice		Typhus	Bacteria
		Louse-borne relapsing fever	Bacteria
Sandflies		Leishmaniasis	Parasite
		Sandfly fever (phlebotomus fever)	Virus
Ticks		Crimean-Congo haemorrhagic fever	Virus
		Lyme disease	Bacteria
		Relapsing fever (borreliosis)	Bacteria
		Rickettsial diseases (eg: spotted fever and Q fever)	Bacteria
		Tick-borne encephalitis	Virus
		Tularaemia	Bacteria
Triatome bugs		Chagas disease (American trypanosomiasis)	Parasite
Tsetse flies		Sleeping sickness (African trypanosomiasis)	Parasite

Figure 5. Non-exhaustive list of vector-borne disease from the WHO [11]

2.1.5 Wide applicability to other Time-series Prediction Problems

Forecasting the future number of dengue cases is a time-series prediction problem. The machine learning algorithm can be adapted to solve other time-series prediction problems (e.g economy forecasting, store sales forecasting, plant equipment failure forecasting).

2.1.6 Benefits of Intelligent Dengue Predictor

With the Intelligent Dengue Predictor, we expect to reduce dengue cases by 10% through the early deployment of preventive measures in affected areas. This can result in major reduction in dengue cases, deaths and economic burden as detailed in the following table.

Region	Number of cases reduced	Number of deaths prevented	Economic burden relieved
Global	400 million x 10% = 40 million	22,000 x 10% = 2,200	US$8.9 billion x 10% = US$ 890 million
Singapore	35,315 (in 2020) x 10% = 3,532	32 (in 2020) x 10% = 3	Between US$0.85 billion and US$1.15 billion per year (from 2000 to 2009) x 10% = US$85 million to US$115 million

Figure 6. Direct benefits of Intelligent Dengue Predictor

With the reduction in dengue cases, the probability of falling sick or deaths will be lower. Therefore, we expect the quality of life to improve for the people living in dengue areas.

2.2 Research

2.2.1 What is Dengue?

Dengue is a mosquito-borne viral disease that has rapidly spread in all regions of WHO in recent years. Dengue virus is transmitted by female mosquitoes mainly of the species Aedes aegypti and, to a lesser extent, Ae. albopictus. These mosquitoes are also vectors of chikungunya, yellow fever and Zika viruses. Dengue is widespread throughout the tropics, with local variations in risk influenced by rainfall, temperature, relative humidity and unplanned rapid urbanization.[1]

Dengue causes a wide spectrum of disease. This can range from subclinical disease (people may not know they are even infected) to severe flu-like symptoms in those infected. Although less common, some people develop severe dengue, which can be any number of complications associated with severe bleeding, organ impairment and/or plasma leakage. Severe dengue has a higher risk of death when not managed appropriately. Severe dengue was first recognized in the 1950s during dengue epidemics in the Philippines and Thailand. Today, severe dengue affects most Asian and Latin American countries and has become a leading cause of hospitalization and death among children and adults in these regions.[1]

Dengue is caused by a virus of the Flaviviridae family and there are four distinct, but closely related, serotypes of the virus that cause dengue (DENV-1, DENV-2, DENV-3 and DENV-4). Recovery from infection is believed to provide lifelong immunity against that serotype. However, cross-immunity to the other serotypes after recovery is only partial, and temporary. Subsequent infections (secondary infection) by other serotypes increase the risk of developing severe dengue.[1] It is possible to get infected 4 times by the different dengue serotypes.

Dengue has distinct epidemiological patterns, associated with the four serotypes of the virus. These can co-circulate within a region, and indeed many countries are hyper-endemic for all four serotypes. Dengue has an alarming impact on both human health and the global and national economies. DENV is frequently transported from one place to another by infected travellers; when susceptible vectors are present in these new areas, there is the potential for local transmission to be established.[1]

2.2.2 Mosquito life cycle

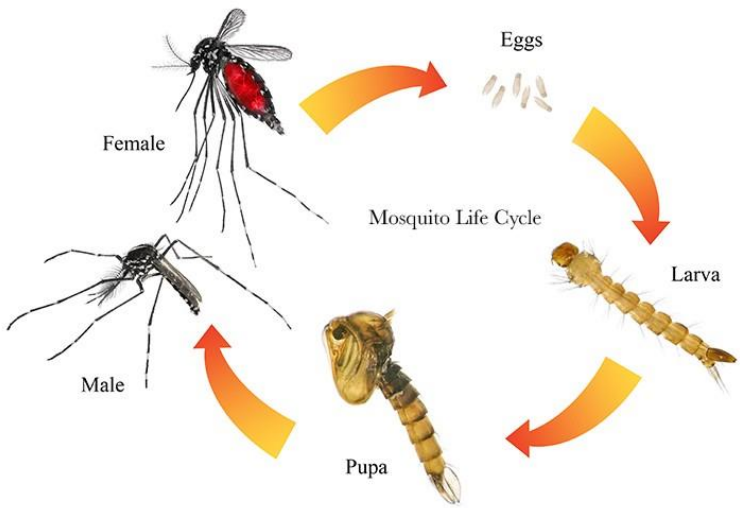


Figure 7. Mosquito Life Cycle

After a female Aedes aegypti feeds on blood, she lays an average of 100 to 200 eggs and can do this up to five times in her lifespan. Eggs are laid on damp surfaces in areas likely to temporarily flood, such as tree holes and man-made containers like barrels, drums, jars, pots, buckets, flower vases, plant saucers, tanks, discarded bottles, tins, tires, water cooler, etc. and a lot more places where rain-water collects or is stored. She lays her eggs separately, unlike most species. Not all eggs are laid at once, but they can be spread out over hours or days, depending on the availability of suitable substrates. Eggs will most often be placed at varying distances above the waterline. Additionally, she will not lay the entire clutch at a single site, but rather spread out the eggs over several sites. The eggs can survive in a dry environment for over a year but when they are in a wet environment they can hatch in as little as two days. The remaining growth stages to adulthood can occur in as little as six days (denguevirusnet.com)[12]

It is important to know the mosquito life-cycle, because there are correlations

[12] Towards Data Science - Using Keras and TensorFlow to Predict Dengue Fever Outbreaks. https://towardsdatascience.com/using-keras-and-tensorflow-to-predict-dengue-fever-outbreaks-99392202bd5c (Accessed 28 Apr 2021)

between weather conditions and the breeding rate of mosquitoes. More mosquitoes will lead to faster transmission rates.

2.2.3 Effects of weather on mosquito breeding rate

In San Juan, reported dengue cases often spike during the later part of the year and clearly some years have significant increases in the number of infections. This is understood to be correlated to increases in the number of mosquitos present. Mosquitos reproduce more readily in warm, humid conditions and when rainfall is plentiful. These conditions create more areas of standing water and thus more locations suitable for laying eggs.[12]

2.2.4 Transmission

2.2.4.0 Mosquito-to-human transmission

The virus is transmitted to humans through the bites of infected female mosquitoes, primarily the Aedes aegypti mosquito. Other species within the Aedes genus can also act as vectors, but their contribution is secondary to Aedes aegypti.

After feeding on an DENV-infected person, the virus replicates in the mosquito midgut, before it disseminates to secondary tissues, including the salivary glands. The time it takes from ingesting the virus to actual transmission to a new host is termed the extrinsic incubation period (EIP). The EIP takes about 8-12 days when the ambient temperature is between 25-28°C. Variations in the extrinsic incubation period are not only influenced by ambient temperature; a number of factors such as the magnitude of daily temperature fluctuations, virus genotype, and initial viral concentration can also alter the time it takes for a mosquito to transmit virus. Once infectious, the mosquito is capable of transmitting virus for the rest of its life.[1]

2.2.4.1 Human-to-mosquito transmission

Mosquitoes can become infected from people who are viremic with DENV. This can be someone who has a symptomatic dengue infection, someone who is yet to have a symptomatic infection (they are pre-symptomatic), but also people who show no signs of illness as well (they are asymptomatic).

Human-to-mosquito transmission can occur up to 2 days before someone shows symptoms of the illness, up to 2 days after the fever has resolved.

Risk of mosquito infection is positively associated with high viremia and high fever in the patient; conversely, high levels of DENV-specific antibodies are associated with a decreased risk of mosquito infection (Nguyen et al 2013 PNAS). Most people are viremic for about 4-5 days, but viremia can last as long as 12 days.[1]

2.2.4.2 Other modes of transmission

The primary mode of transmission of DENV between humans involves mosquito vectors. There is evidence however, of the possibility of maternal transmission (from a pregnant mother to her baby). Vertical transmission rates appear to be low, with the risk of vertical transmission seemingly linked to the timing of the dengue infection during the pregnancy. When a mother does have a DENV infection when she is pregnant, babies may suffer from pre-term birth, low birthweight, and fetal distress. [1]

3 System Design / Model

3.1 System Design

The system architecture for Dengue Cases Prediction is illustrated in diagram below.

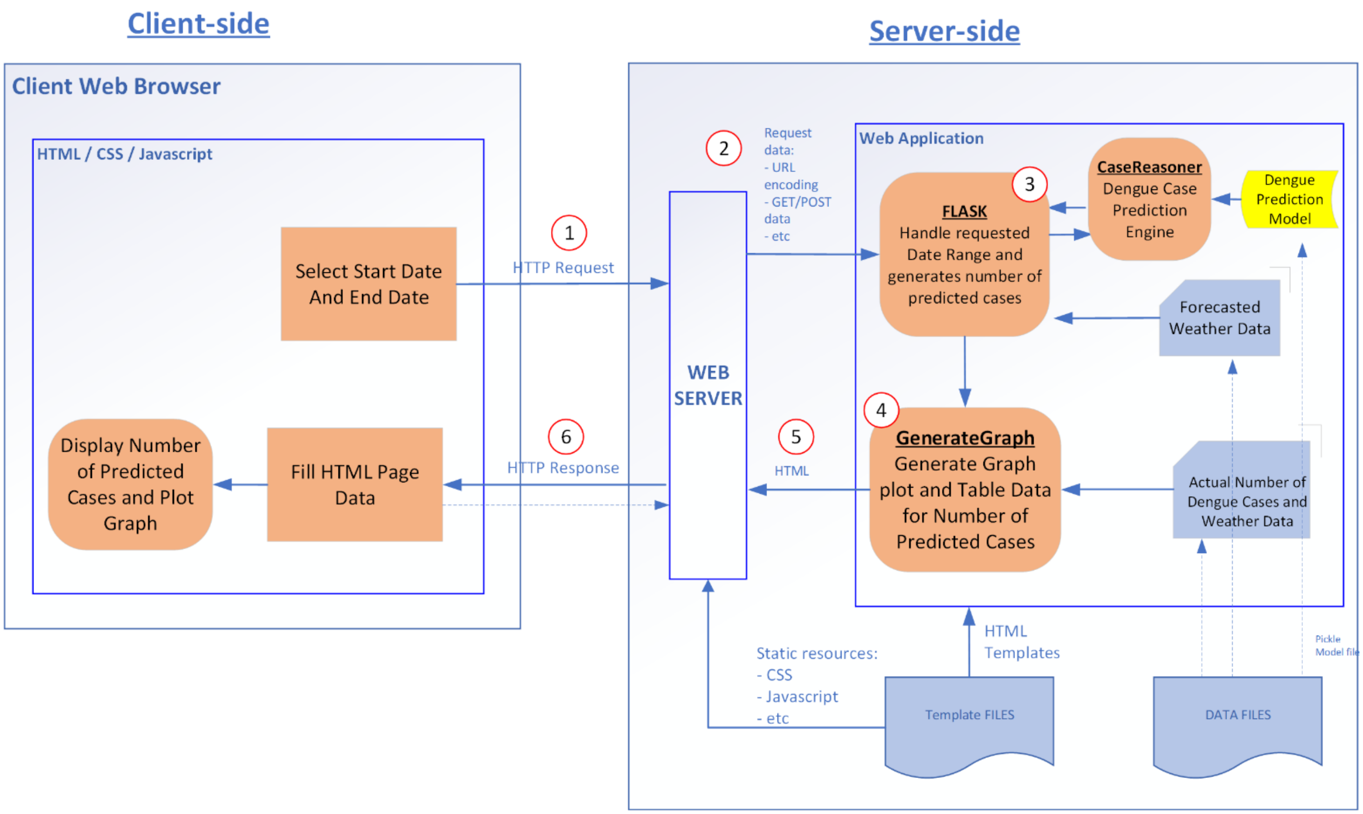


Figure 8: System Architecture

It is designed as a Client-Server based application/architecture and consists of the following:

1) Client Side

a. User can access the URL address of the Dengue Cases Prediction Web page and enter the date range that he/she wants to display the dengue cases prediction.

2) Server Side

a. A web server that hosts a web application (written using Python and FLASK web framework). It handles the requests from the web browser and processed it to generate the information that is required for display in the Web Browser page of the Client.

The diagram shows the main elements of the "Dengue Cases Prediction" website, along with numbered labels for the sequence of operations when the Client accesses the Dengue Cases Prediction information.

The parts of the site that make it dynamic are the Web Application (server-side code that processes HTTP requests and returns HTTP responses), the Dengue Case Prediction Engine and Model, the HTML Templates and the DATA FILES (which contains the information of past actual weather data (Dec 2016 to Nov 2020) and forecasted weather data (Dec 2020 to April 2021) and past actual number of Dengue Cases (Dec 2016 to Nov 2020)).

Note*: The DATA FILES will need to be updated regularly in an OFF-LINE process as forecasted weather data and predicted number of Dengue cases will be replaced with actual weather data and actual number of dengue cases, respectively. This will also require update/tuning to the Dengue Prediction Model in the OFF-LINE process as more data are available.*

Assuming the Client has accessed the URL of the main Dengue Cases Prediction web page. After the Client submits the Date Range with the "future" Start Date and End Date that he/she is interested to predict the Number of Dengue Cases:

1. 1 The Client's Web Browser creates an HTTP GET request to the server using the base URL for the resource (/Range/) and encoding the Start Date and End Date as URL parameters. A GET request is used because the request is only fetching data (not modifying data).

2. 2 The Web Server detects the request and forwards it to the Web Application for processing (the web server determines how to handle different URLs based on pattern matching rules defined in its configuration).

3. 3 The Web Application identifies that the intention of the request is to get the predicted number of dengue cases based on the URL (/Range/) and the Start Date and End Date information from the URL. The Web Application then uses the Start Date and End Date range to determine the forecasted weather data of the date range and supply them to the Dengue Prediction Engine to generate the predicted number of dengue cases for each day of the date range.

4. 4 The Web Application dynamically creates an HTML page by putting the data (from the predicted number of dengue cases for each day of the date range) into placeholders inside the HTML template. This includes the population of the table with Headings (Date, Number of Predicted Cases) and a dynamic Graph Plot (using Bokeh Python package).

5. 5 The Web Application returns the generated HTML to the Client's web browser (via the Web Server), along with an HTTP status code of 200 ("success"). If anything prevents the HTML from being returned, then the Web Application will return another code – for example "404" to indicate the page is unavailable or no display update to imply no predicted data available.

6. 6 The Web Browser will then start to process and display the returned HTML, sending separate requests to get any other CSS or JavaScript files to the Web Server that it references:

- The Web Server loads static files from the file system and returns them to the browser directly.

Below is an interaction view for the sequences described from 1 to 6 above:

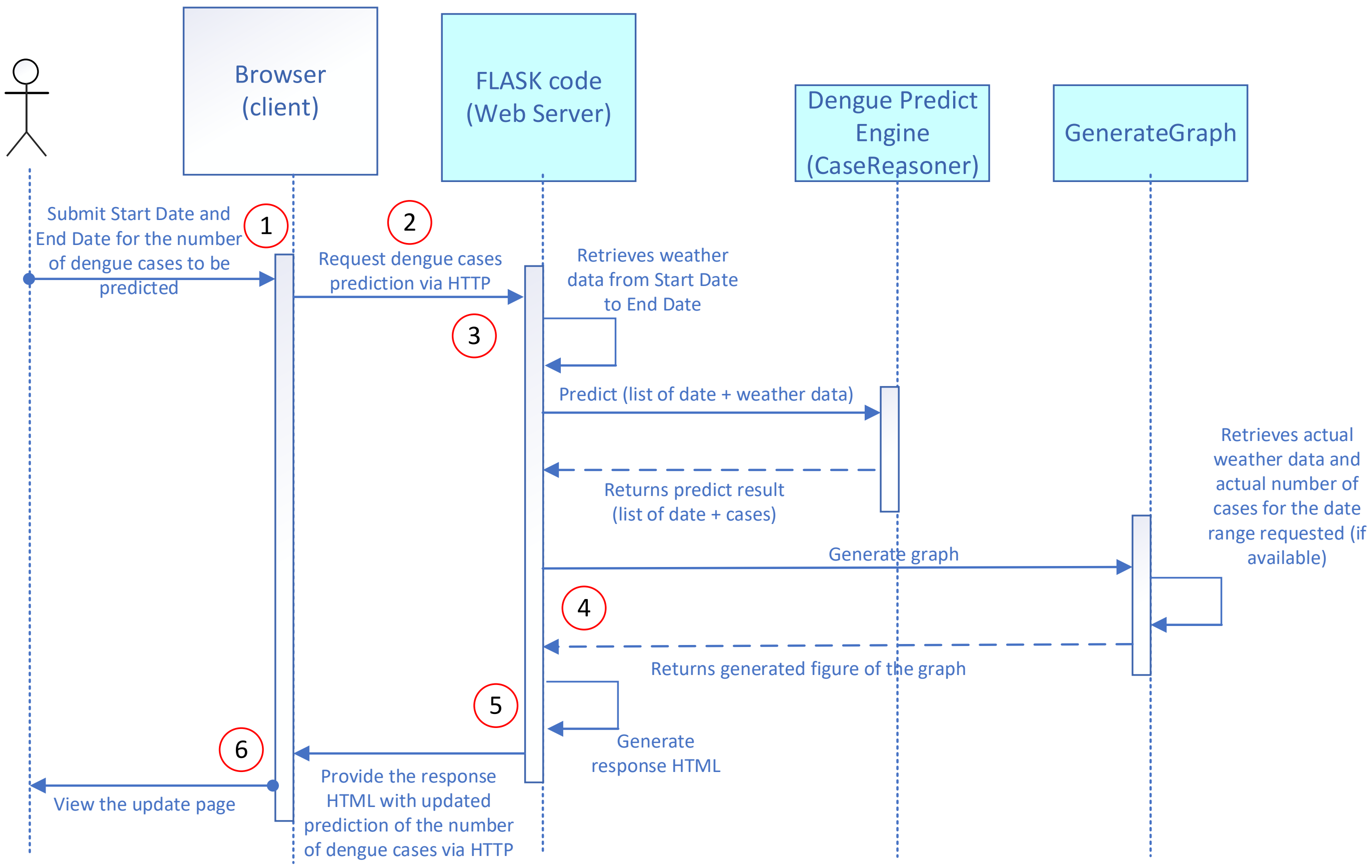


Figure 9: System Interactions Diagram

Note:

Dengue Predict Engine (CaseReasoner) loads a python pickle serialized object "TrainedModel" data. This is the trained machine learning model that is used to predict the number of the dengue cases, using the model that was decided to have the desirable level of accuracy or precision. It allows the web application to re-use the model without having to re-train it all over again every time the application starts.

There will be an OFF-LINE process that will train the model again when new data is available (split the data set, used LightGBM to train the model, tested it, re-trained it, tested and went through all the cycle to enhance it to the desirable level of accuracy and precision) and generate new version of the model pickle file again at a regular interval. This will help to preserve the model that was trained earlier and for comparison of the model's performance.

3.1.1 System Hardware and Setup

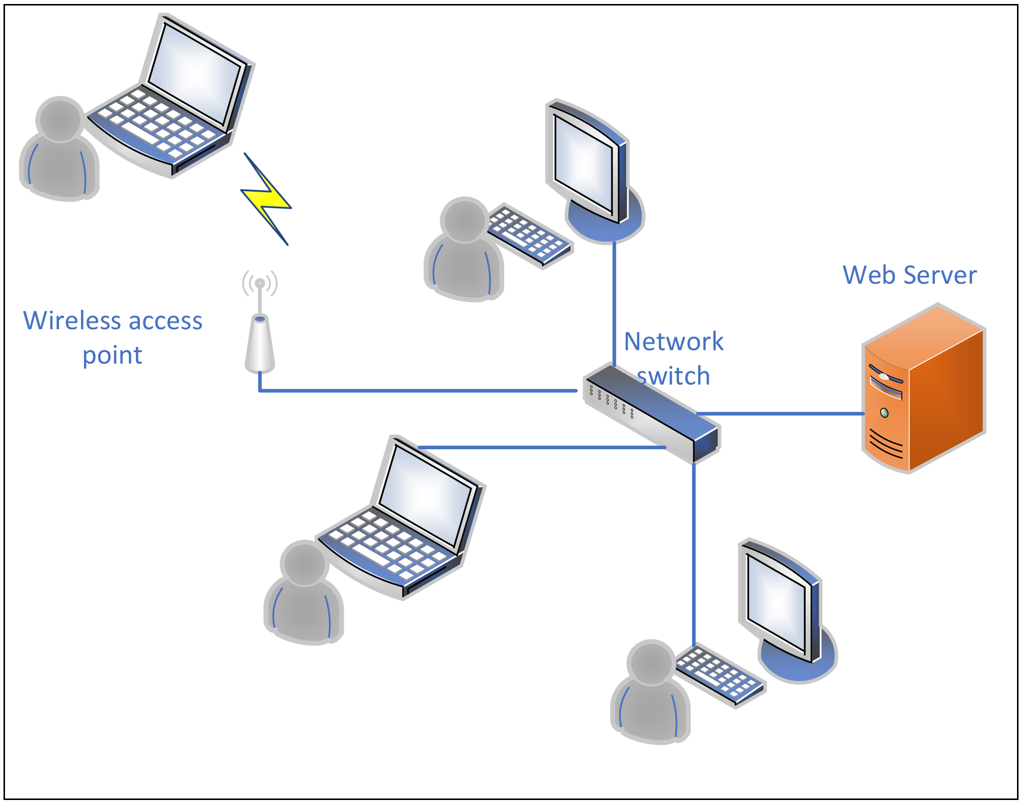


Figure 10: Typical Hardware setup

The system is designed to be setup in a Local Area Network environment.

The following hardware requirements outline the minimum hardware required for any machine that will serve as the Web Server machine, Client's computer, and the configuration of a Local Area Network:

- Web Server Machine:
  - Intel Core i7 2.0 Ghz or higher with minimum 16 GB RAM and 50 GB HDD
- Desktop/Laptop
  - Intel Core i5 1.8 Ghz or higher with minimum 4 GB RAM and 50 GB HDD.
- Local Area Network (To network between the Web Server and the Users' work computer)
  - 100BASE-TX Network Switch or higher
  - Wireless Router (802.11 G/N compatible or higher router)

The web Server and web application will be hosted in a single computer and connected to a switch or wireless access point.

Users within an organization department can access the Dengue Cases Prediction web site via their work desk computer or notebook.

Below is the user view of the Dengue Cases Prediction web site.

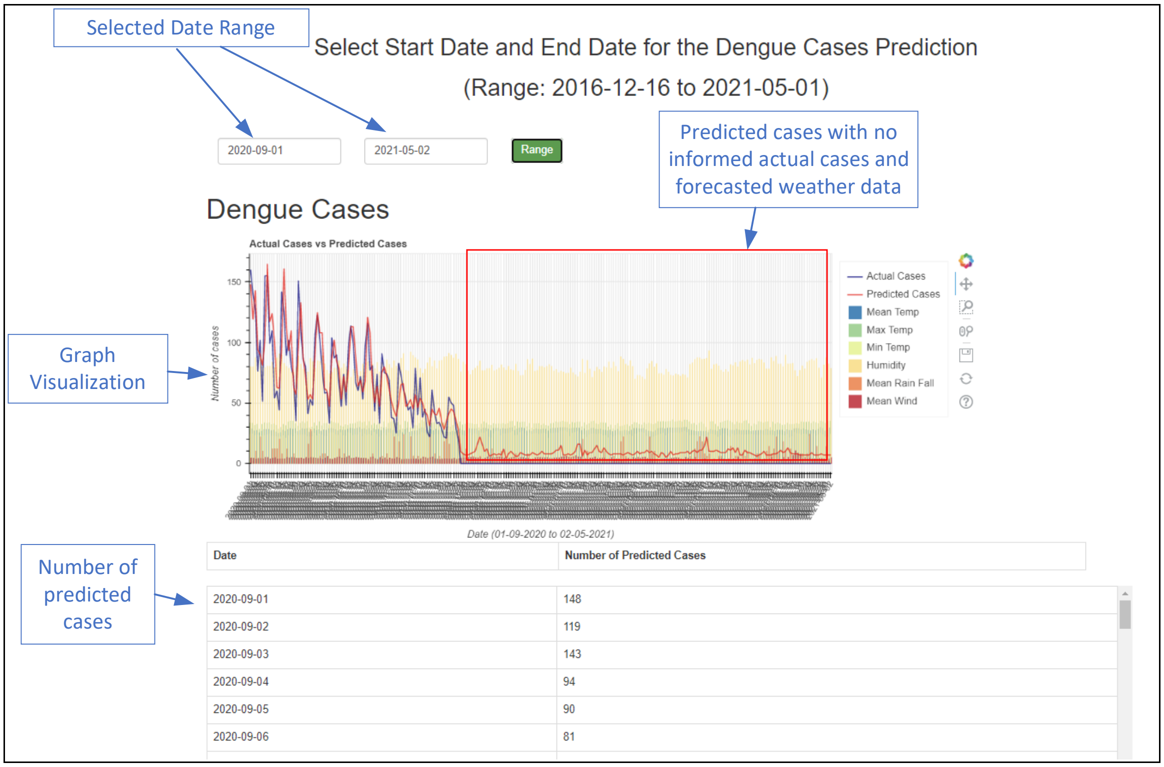


Figure 11: Dengue Cases Prediction web page

Refer to appendix for the user interfaces guide 9.3

3.2 System Development and Implementation Approaches

The development of the Dengue Cases Prediction system adopts existing open-source sub-modules and/or application programming interfaces APIs to minimize "reinventing the wheels". To see the list of open-source sub-modules used, refer to 4.System Development & Implementation Tools.

FLASK(https://en.wikipedia.org/wiki/Flask_(web_framework)):

This system utilizes the FLASK web frameworks as it simplifies the server-side web programming (it makes writing code to handle operations much easier).

One of the most important operations it performs is providing simple mechanisms to map URLs for different resources/pages to specific handler functions. This makes it easier to keep the code associated with each type of resource separate. It also has benefits in terms of maintenance because you can change the URL used to deliver a particular feature in one place, without having to change the handler function.

For example, consider the following FLASK code that maps two URL patterns ('/' and '/range') to two function web pages. The first pattern ensures that an HTTP request with a resource URL of '/' will be passed to a function named index() and '/range' will be passed to a function named range().

For the range() function, it gets the Start Date and End Date and uses them to determine the forecasted weather data of the date range and supply them to the Dengue Prediction Engine (Reasoner) to generate the predicted number of dengue cases for each day of the date range. Then it calls the render_template() function, passing an HTML template and a "context" object defining the information to be included in the template. The render_template() function is a convenience function that generates HTML using a context and an HTML template, and returns it in an HttpResponse object to the Web Browser (via the Web Server).

```
@app.route('/')
def index():
    return render_template('index.html')

@app.route("/range",methods=["POST","GET"])
def range():

    print(request.method)

    if request.method == 'POST':
        From = request.form['From']
        to = request.form['to']
        print(From)
        print(to)

        script, div, predict_cases = createDengueCasesPlot(From=From, to=to)
        return jsonify({'htmlresponse': render_template('response.html', predict_cases=predict_cases,
the_div=div, the_script=script)})
```

Figure 12: Example of FLASK http routing code

3.2.1 Development Project Folder Organization

```
Project Organization
------------
    ├── LICENSE
    ├── Makefile           <- Makefile with commands like `make data` or `make train`
    ├── README.md          <- The top-level README for developers using this project.
    ├── data
    │   ├── external       <- Data from third party sources. (Not used at the moment)
    │   ├── interim        <- Intermediate data that has been transformed. (Not used at the moment)
    │   ├── processed      <- The final, canonical data sets for modeling.
    │   └── raw            <- The original, immutable data dump.
    │
    ├── docs               <- A default Sphinx project; see sphinx-doc.org for details (Not used at the moment)
    │
    ├── models             <- Trained and serialized models, model predictions, or model summaries
    │   └── lightGNM_MAY_2021 <- light GBM serialized models.
    │
    ├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
    │                         the creator's initials, and a short `-` delimited description, e.g.
    │                         `1.0-jqp-initial-data-exploration`.
    │
    ├── references         <- Data dictionaries, manuals, and all other explanatory materials. (Not used at the moment)
    │
    ├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc. (Not used at the moment)
    │   └── figures        <- Generated graphics and figures to be used in reporting (Not used at the moment)
    │
    ├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
    │                         generated with `pip freeze > requirements.txt`
    │
    ├── setup.py           <- makes project pip installable (pip install -e .) so src can be imported
    ├── src                <- Source code for use in this project.
    │   ├── __init__.py    <- Makes src a Python module
    │   │
    │   ├── data           <- Scripts to download or generate data (Not used at the moment)
    │   │   └── make_dataset.py
    │   │
    │   ├── features       <- Scripts to turn raw data into features for modeling (Not used at the moment)
    │   │   └── build_features.py
    │   │
    │   ├── models         <- Scripts to train models and then use trained models to make
    │   │   │                 predictions
    │   │   ├── predict_model.py
    │   │   └── train_model.py
    │   │
    │   ├── visualization  <- Scripts to create exploratory and results oriented visualizations
    │   │   └── generate_graph.py
    │   │
```

```
│   ├ appy.py
│   │
│   └ definitions.py
│
└── tox.ini            <- tox file with settings for running tox; see tox.readthedocs.io (Not used at the moment)
```

3.3 Data Collection

Data on dengue cases for the period 16th December 2016 to 26th November 2020 are collected from National Environment Agency (NEA) website.[13] Dengue cases has been included in the system as one of the surveilled diseases in Singapore. Since dengue fever is a legally national notifiable infectious disease, case data was collected systematically and continuously. Dengue fever cases were diagnosed based on standardized laboratory tests and clinical/ epidemiological investigations.

Number of Reported Cases

24-Apr	25-Apr	26-Apr	27-Apr	28-Apr	29-Apr	30-Apr at 3pm
17	8	16	18	7	9	11

Number of Reported Cases by E-week (from Sun 0000hrs to Sat 2359hrs)

E-week 11 (14-20Mar21)	E-week 12 (21-27Mar21)	E-week 13 (28Mar-03Apr21)	E-week 14 (04-10Apr21)	E-week 15 (11-17Apr21)	E-week 16 (18-24Apr21)	E-week 17 (25-30Apr21 at 3pm)
129	120	113	102	112	131	69

Figure 13: Screenshot from NEA Dengue Cases Website

[13] National Environment Agency Dengue Cases Page. https://www.nea.gov.sg/dengue-zika/dengue/dengue-cases (Accessed 28 Apr 2021)

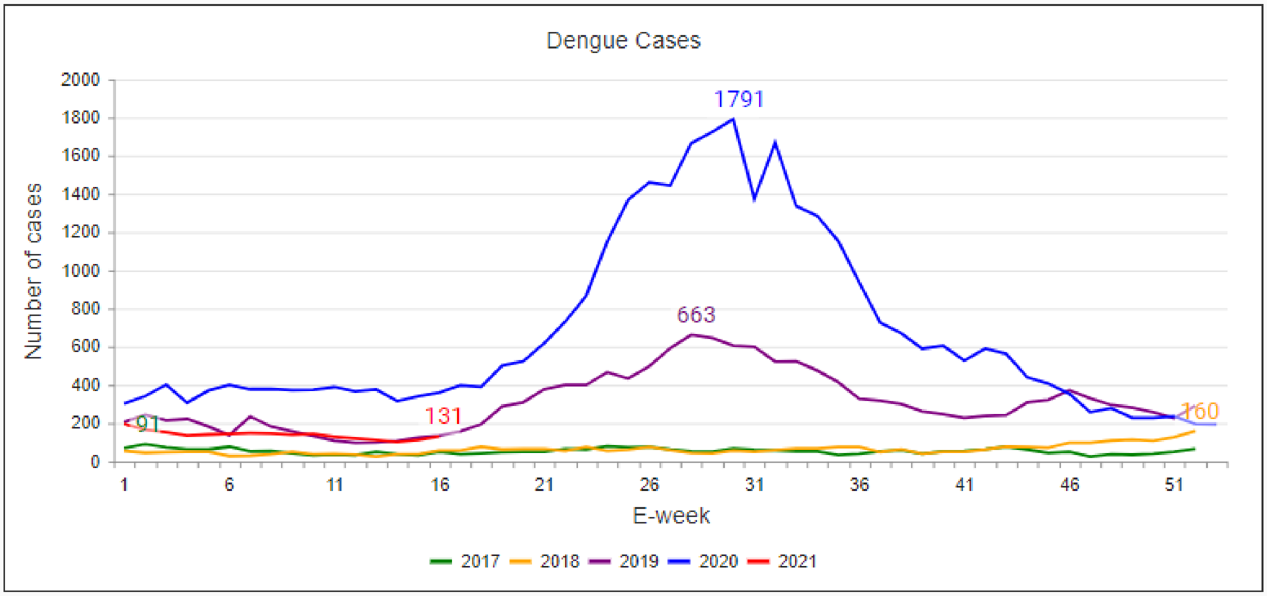


Figure 14: Historical Dengue Cases in Past Few Years.

Weather data in Singapore for the period 16th December 2016 to 26th November 2020 were collected from the data.gov.sg Realtime Weather Readings across Singapore[14].

The data.gov.sg was first launched in 2011 as the government's one-stop portal to its publicly available datasets from 70 public agencies. To date, more than 100 apps have been created using the government's open data.

Weather data is collected from data.gov.sg API including daily average minimum temperature, maximum temperature, minimum temperature, relative humidity, rainfall, and wind speed.

3.4 Data Processing

After data cleaning and filling the missing data, the time series of daily dengue cases were plotted as below figure.

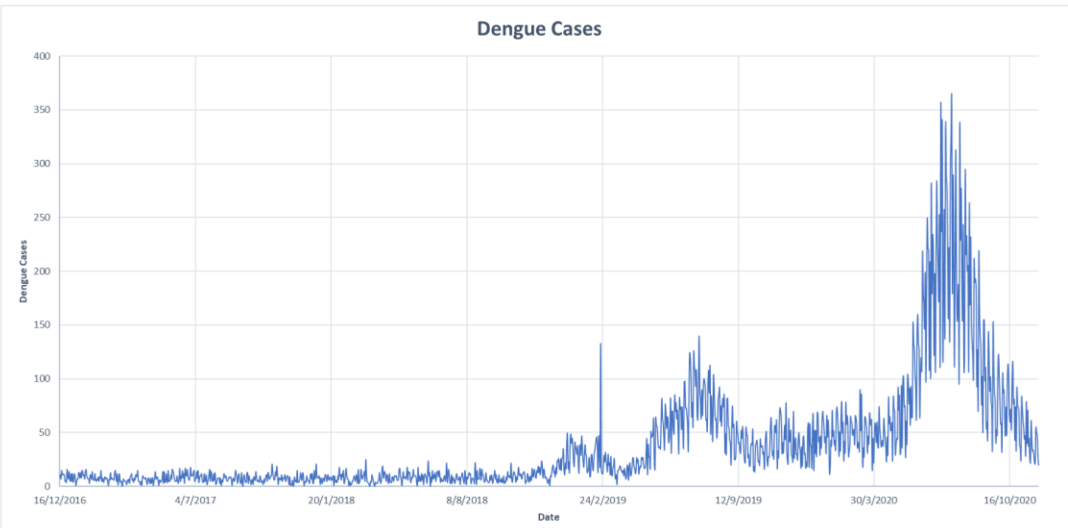


Figure 15: Dengue Cases (From 2016. Dec.16th to 2020. Nov.26th)

To obtain a robust and smooth model, weather data was transformed into new variables using prior moving average of 5days/ 10days / 15days/ 20days and weather information of past few days. Cross-correlations were also conducted between natural logarithm transformed daily dengue case counts in Singapore (natural logarithm transformed case counts = Ln (case counts + 1)) and each new variable (the treated meteorological variable) to optimize the prediction model performance.

[14] Data.gov.sg Realtime Weather Readings across Singapore. https://data.gov.sg/dataset/realtime-weather-readings (Accessed 28 Apr 2021)

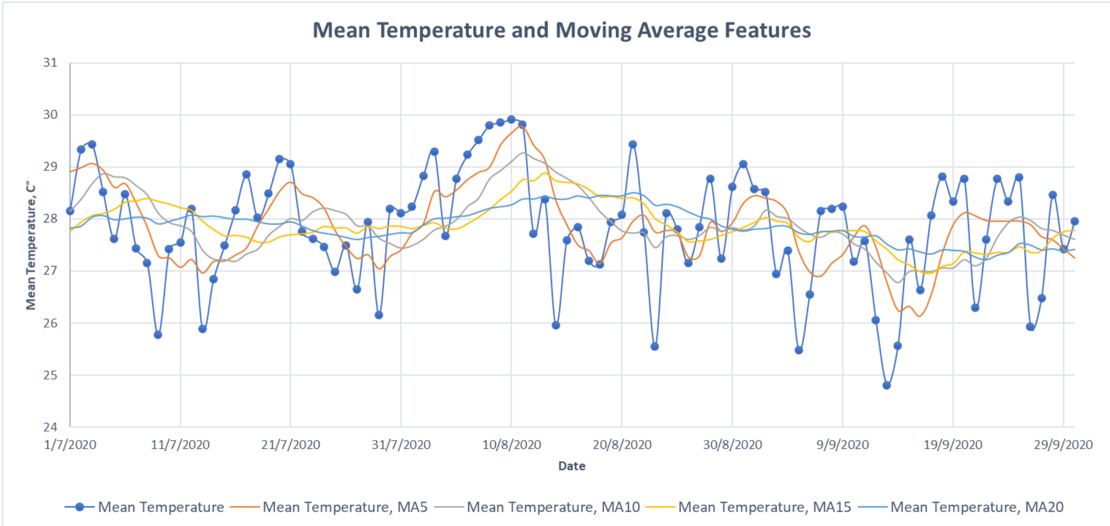


Figure 16: Mean Temperature and Moving Average Features (3rd Quarter of 2020 for Illustration)

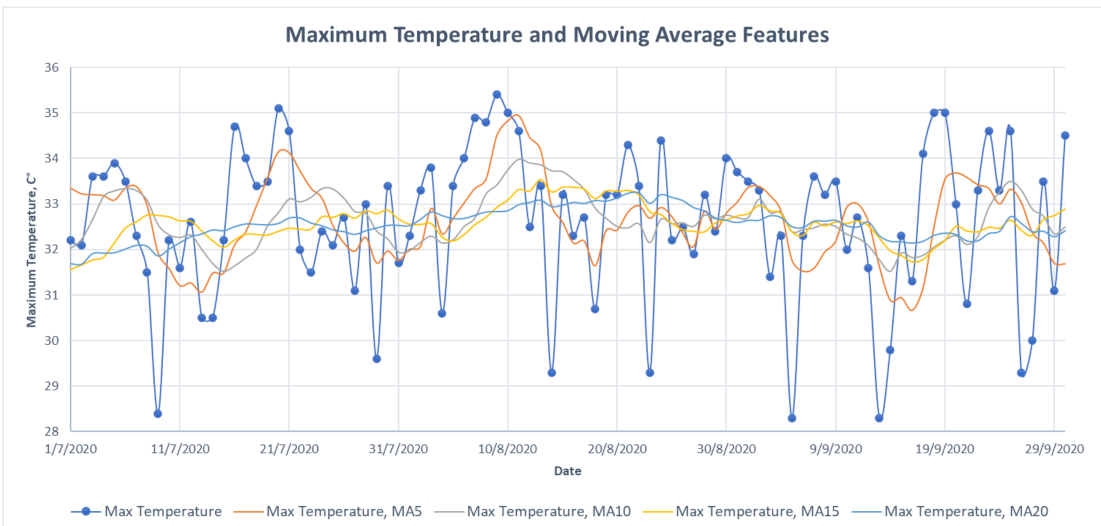


Figure 17: Maximum Temperature and Moving Average Features (3rd Quarter of 2020 for Illustration)

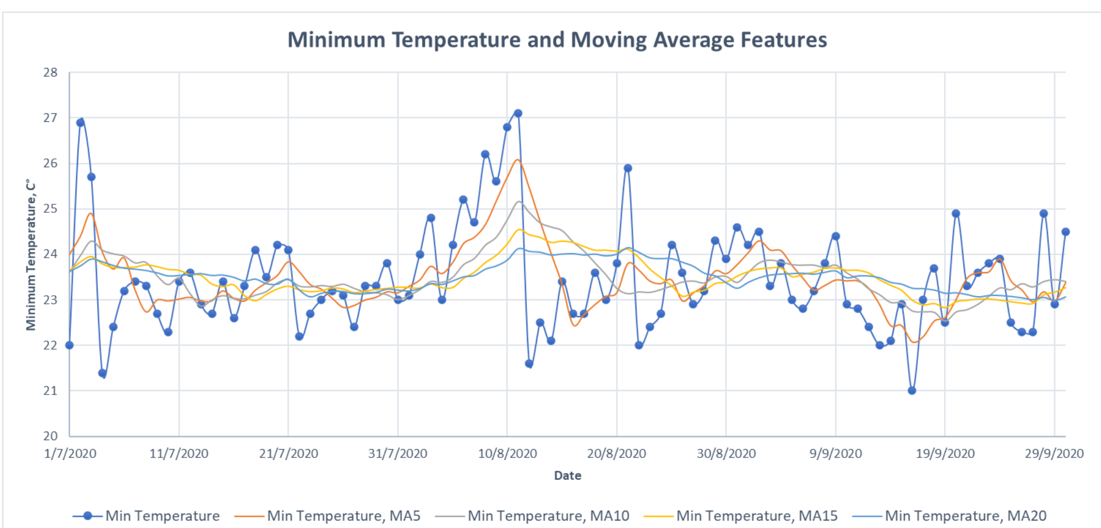


Figure 18: Minimum Temperature and Moving Average Features (3rd Quarter of 2020 for Illustration)

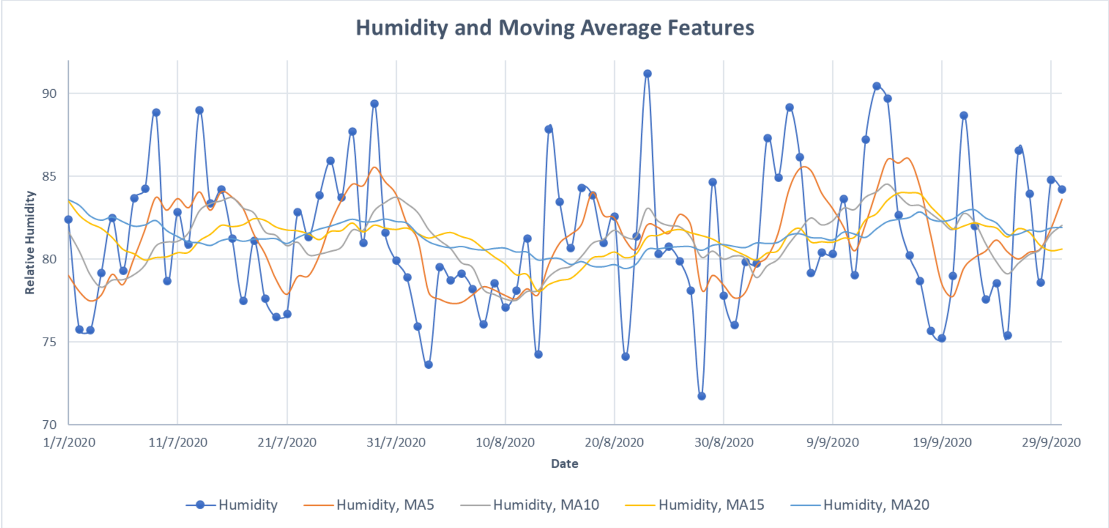


Figure 19: Relative Humidity and Moving Average Features (3rd Quarter of 2020 for Illustration)

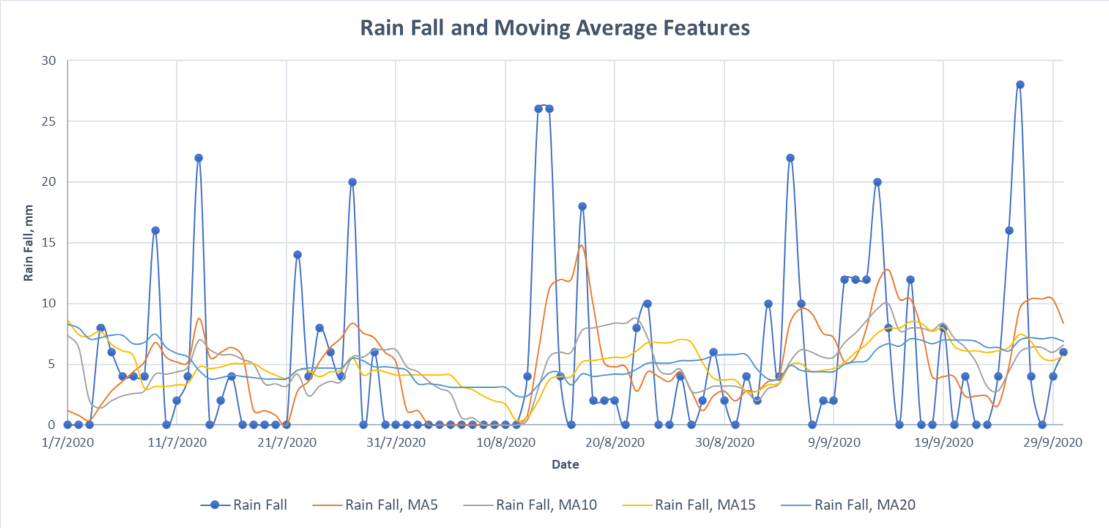


Figure 20: Rain Fall and Moving Average Features (3rd Quarter of 2020 for Illustration)

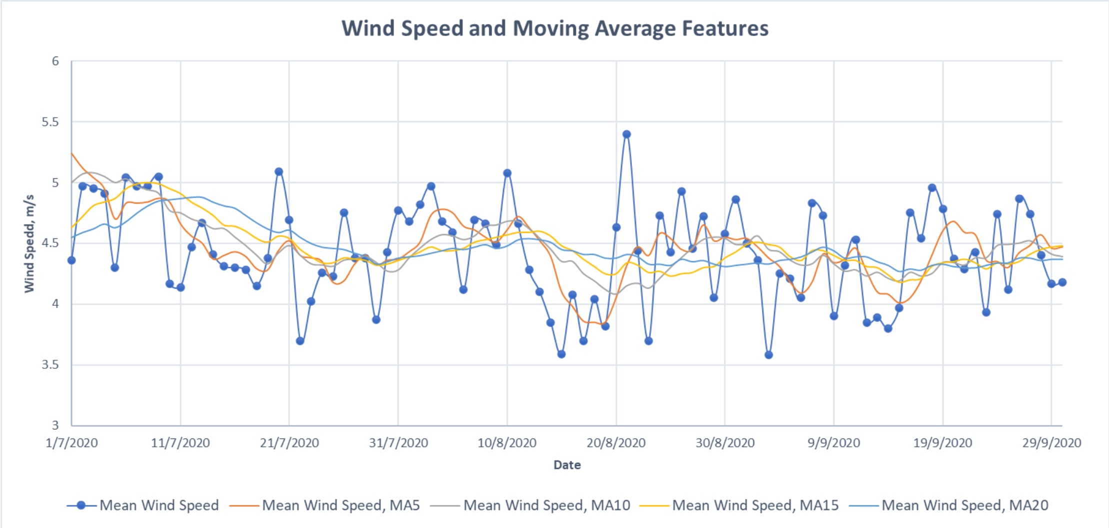


Figure 21: Wind Speed and Moving Average Features (3rd Quarter of 2020 for Illustration)

3.5 Model Selection

Machine Learning Model algorithms selection, due to our actual problem is regression problem, here is the candidate's algorithm that planned to use:

- Linear Regression
- Lasso Regression (aka L1 regularization)
- Ridge Regression (aka L2 regularization)
- ElasticNet: Linear regression with combined L1 and L2 priors as regularizer.
- Support Vector Machine
- Decision Tree
- Neural Network: Multi-layer Perceptron regressor.
- RandomForest Regressor
- GradientBoostingMachine
- XGBoost
- LightGBM

Metric Selection:

- MAE: Mean Absolute Error
- MSE: Mean Square Error
- R^2 Score: (coefficient of determination) regression score function.

Hyper-parameter optimizers:

- GridSearchCV: Exhaustive search over specified parameter values for an estimator.
- RamdonSearchCV: Randomized search on hyper parameters.
- HalvingGridSearchCV: introduced in sklearn. Ver 0.24, much faster turning speed compared to previous.

4 System Development & Implementation Tools

Type	Tool / Version	Information
Programming Language	Python 3.8.5	**Wiki Description**
		Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.[Link]
		Web links
		Wiki Site: Python (programming language) - Wikipedia Official Site: Welcome to Python.org
Python Webapp framework	Flask 1.1.2	**Wiki Description**
		Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.[Link] It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.[Link] Applications that use the Flask framework include Pinterest and LinkedIn.[Link][Link]
		Web links
		Wiki Site: Flask (web framework) - Wikipedia Official Site: Flask Documentation
Graph visualization (Open-Source sub-module)	Bokeh 2.3.1	**Python Wiki Description**
		Bokeh is a BSD licensed, open source Python interactive visualization platform that targets modern web browsers for presentation. Bokeh can create richly interactive visualizations and data applications, as standalone HTML documents, sophisticated server-backed applications, or inline in Jupyter Notebooks.
		Web links
		Python Wiki Site: Bokeh - Python Wiki Official Site: Bokeh 2.3.1 Documentation
Version control system	Git Hub	**Wiki Description**
		GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuous integration and wikis for every project.[Link] Headquartered in California, it has been a subsidiary of Microsoft since 2018.[Link]

Type	Tool / Version	Information
		Web links
		Wiki Site: GitHub - Wikipedia Official Site: GitHub
Open-Source sub-module	Pandas 1.2.3	**Wiki Description**
		pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.[Link] The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.[Link] Its name is a play on the phrase "Python data analysis" itself.[Link] Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010.[Link]
		Web links
		Wiki Site: pandas (software) - Wikipedia Official Site: pandas - Python Data Analysis Library (pydata.org)
Open-Source sub-module	Numpy 1.19.5	**Wiki Description**
		NumPy (pronounced /ˈnʌmpaɪ/ (NUM-py) or sometimes /ˈnʌmpi/[Link][Link] (NUM-pee)) is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.[5] The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.
		Web links
		Wiki Site: NumPy - Wikipedia Official Site: NumPy
Machine Learning Framework (Open-Source sub-module)	Light GBM 3.2.1	**Wiki Description**
		LightGBM, short for Light Gradient Boosting Machine, is a free and open source distributed gradient boosting framework for machine learning originally developed by Microsoft.[Link][Link] It is based on decision tree algorithms and used for ranking, classification and other machine learning tasks. The development focus is on performance and scalability.
		Web links
		Wiki Site: LightGBM - Wikipedia Official Site: LightGBM documentation
Python sub-module	Pathlib 1.0.1	**Wiki Description**
		This module offers classes representing filesystem paths with semantics appropriate for different operating systems. Path classes are divided between pure paths, which provide purely computational operations

Type	Tool / Version	Information
		without I/O, and concrete paths, which inherit from pure paths but also provide I/O operations.
		Web links
		Official Site: pathlib — Object-oriented filesystem paths documentation
Python sub-module	Datetime 4.3	**Wiki Description**
		The datetime module supplies classes for manipulating dates and times. While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation.
		Web links
		Official Site: datetime — Basic date and time types — documentation
Python Project templates	Cookiecutter 1.7.2	**Wiki Description**
		Cookiecutter creates projects from project templates, e.g. Python package projects.
		Web links
		Official Site: Cookiecutter: Better Project Templates — documentation
Python object serialization	Pickle 3.8.5	**Wiki Description**
		The pickle module implements binary protocols for serializing and de-serializing a Python object structure. *"Pickling"* is the process whereby a Python object hierarchy is converted into a byte stream, and *"unpickling"* is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy. Pickling (and unpickling) is alternatively known as "serialization", "marshalling," 1 or "flattening"; however, to avoid confusion, the terms used here are "pickling" and "unpickling".
		Web links
		Official Site: pickle — Python object serialization — documentation

5 System Results

After the process of data pre-processing, feature engineering, model selection and model tuning, the performance of prediction model has been improved a lot from the initial stage. The dataset has been split into training/ test dataset for cross validation. It found the feature input expansion contribute on the model accuracy. Below table shows the comparison between original weather data and expanded feature after considering moving average, past data and cross-correlation dengue cases.

	MAE	
	Training	Test
Original Weather Data Only, 6 features	22.53	30.85
Moving Average and Past Weather Data Added, 55 features	7.85	15.61

Figure 22: Comparison of the Prediction Results with Different Feature Table

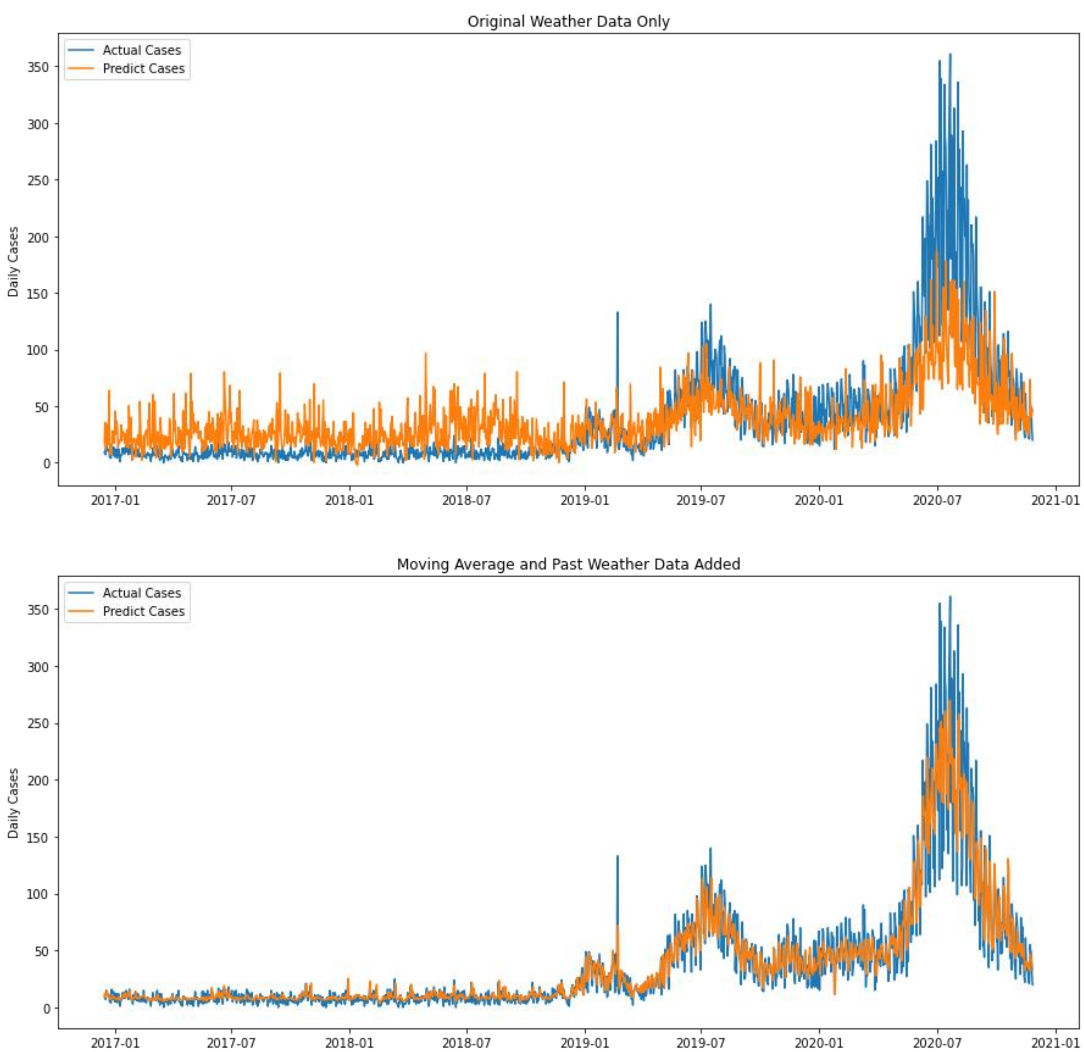


Figure 23: Comparison of the Prediction Results with Different Feature

5.1 Functional Test Cases

5.1.1 Model Testing:

The function of model training and prediction is packed in below python files:

- "train_model.py" should return trained model binary in ".pkl" file, file hash value should be same as expected hash value
- "predict_model.py" should return float number based on the input features vector, for example: input feature vector: [29.24, 34.4, 24.3, 76.2, 0.0, 5.58, 29.02, 33.5, 24.34, 78.36, 1.2, 5.31, 27.98, 31.68, 23.72, 82.42, 8.6, 4.95, 27.76, 31.55, 23.79, 83.69, 8.93, 4.61, 27.89, 31.72, 23.87, 83.36, 8.4, 4.55, 5.36], shall return float number: 101.85.

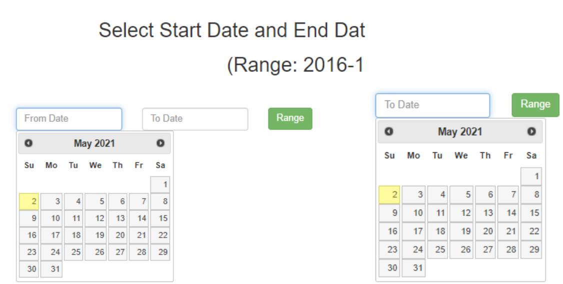
Test No	Test Description	Expected Results	Actual Results
1	Verify that output model hash shall be identical to expected model hash	Hash equal	PASSED
2	Verify that input features vector shall output the expected prediction	Prediction value equal.	PASSED

Figure 24: Model Testing result table

5.1.2 User Interface Testing:

The user interface is a web page and interact with the Python web application via a Web Server.

The functions of the user interface and web application are as follow:

Test No	Test Description	Expected Results	Actual Results
1	Select the From Date and To Date from the web page	Web page allows the selection of From Date and To Date. 	PASSED

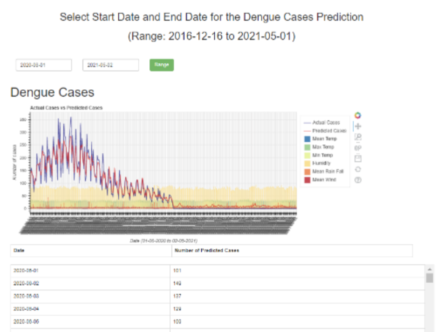
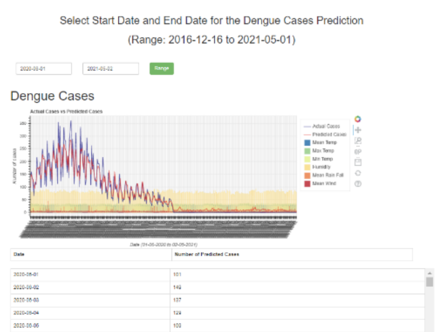
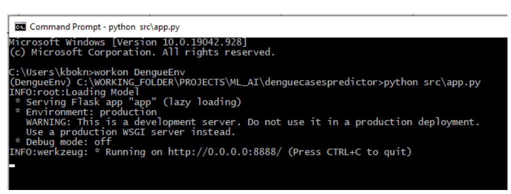
Test No	Test Description	Expected Results	Actual Results
2	Submit the From Date and To Date of the web page and to the web application.	Web page allows the selection of the From Date and To Date and submit it to the web application.  	PASSED
3	Web application processes the submitted date range request and generates the predicted number of dengue cases for the date range requested.	Web page displayed the generated the number of predicted dengue cases.  	PASSED
4	Web application can generate an HTML page and display the result of the generated number of predicted dengue cases in a Table format and a Graph format and display in a Web Browser.	Web page displayed the generated the number of predicted dengue cases.  	PASSED
5	The web application is started within a Web server.	The command prompt window is able to start the web server/application in a python virtual environment:  	PASSED

Figure 25: User Interface and Web application testing result table

6 Conclusion

During trial run, original weather data is selected for the feature input to train the model. It found the feature is not enough to describe the nature of problem. Feature engineering, including moving average and past weather data is combined into the feature input. It shows in section 3, the model prediction accuracy increased with this process. This also shown the correlation between the dengue cases is highly related to the mosquito breeding process. The feature needs to cover the range of weather information to decide whether mosquito is easy to breed during the period.

After expansion on the feature, we also tried various of machine learning algorithms as below:

- Linear Regression
- ElasticNet
- Decision Tree Regressor
- RandomForest Regressor
- GradientBoostingMachine Regressor
- LightGBM

Currently the LightGBM algorithm perform the best, due to LightGBM rich hyperparameters option that can better fitted to the current problem, compared to Scikit-Learn GradientBoostingMachine, and also benefit from the gradient boosting algorithms, which is the parent algorithm for GradientBoostingMachine in sklearn, and XGboost, and LightGBM, it's to predict the residual of the previous prediction instead of predicting the actual value, gave the extra power of LightGBM to the previous algorithms. We have tried the XGBoost, but compare with LightGBM, which can train the model much faster and more efficient, as LightGBM use histogram based and replaces continuous values to discrete bins, we finally choose LightGBM as our final model.

But we also realized that is actually the data define the maximum where the model can go, and the algorithms just trying to approach to this maximum, for machine reasoning, the critical part is how we define our data, and how we select the features that can describe the phenomenon the most.

Limitation

Compare the actual production, our dataset size still small, for the further research, we need much more data to generate the knowledge for reasoning, and to further leverage the power of Gradient Boosting Algorithm, and LightGBM parallel training, 100 times more would be ideal number.

7 Appendix A: Project Proposal

Date of proposal: 1 May 2021
Project Title: ISS Project – Intelligent Dengue Predictor
Sponsor/Client: *(Name, Address, Telephone No. and Contact Name)* Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore NATIONAL UNIVERSITY OF SINGAPORE (NUS) Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg
Background/Aims/Objectives: We would like to help prevent dengue by building an Intelligent Dengue Predictor application. The application will provide early and accurate dengue case forecasts to public health officials. This is achieved by applying Machine learning techniques to find the co-relation between weather conditions and the resulting dengue cases.
Requirements Overview:
Research ability Programming ability System integration ability
Resource Requirements (please list Hardware, Software and any other resources) Hardware Requirements: • Intel Core i7 2.0 Ghz or higher with minimum 16 GB RAM and 50 GB HDD Software Requirements: • Programming IDE: Visual Studio Code or Jupyter Notebook • Programming Language: Python • Webapp framework – Flask • Gradient boosting framework: LightGBM • Graph visualization - Bokeh • Version control system - GitHub

Number of Learner Interns required: (Please specify their tasks if possible)

Four team members

Name	Tasks
Bai Sihai	Project management: • Co-ordinate team efforts • Organize meetings • Provide meeting agenda and meeting minutes Application implementation: • Data Pre-processing • Results visualization module Report: • Project report template design • Research on dengue • Report writing and consolidation Video Presentation: • Slide creation and presentation for business case
Lu Zhiping	Machine Learning Algorithms exploration: • Select the best performing model • Train the model on the full dataset • Write Python module that run the model training, and the `CaseReasoner` to use the trained model, receive the incoming vector, and produce the prediction.
Men Jinlong	Prediction Model Preparation and Validation which including: • Dengue cases collection from NEA website • Weather data collection from Data.gov.sg Realtime Weather Data across Singapore API • Data pre-processing • Feature Engineering • Model cross validation
Ng Kwee Boon	Software integration: • Validate and integrate the software solutions by bringing together the different software sub-modules together and integrate them to a unified single software system. • Collaborates and resolves any interface issues among the sub-modules with all team members. • Check project deliveries from team members for completeness. • Manage the software project repository in GitHub and its layout. • Installation and setup of the development environment and application execution. Application Architecture/Design: • Provides the software structure for the Dengue Cases Prediction web application in Python FLASK. • Test the final application execution.

Methods and Standards:

Procedures	Objective	Key Activities
Requirement Gathering and Analysis	The team should meet with ISS to scope the details of project and ensure the achievement of business objectives.	1. Gather & Analyze Requirements 2. Define internal and External Design. 3. Prioritize & Consolidate Requirements 4. Establish Functional Baseline
Technical Construction	1. To develop the source code in accordance to the design. 2. To perform unit testing to ensure the quality before the components are integrated as a whole project.	1. Setup Development Environment 2. Understand the System Context, Design 3. Perform Coding 4. Conduct Unit Testing
Integration Testing	To ensure interface compatibility and confirm that the integrated system hardware and system software meets requirements and is ready for acceptance testing.	1. Prepare System Test Specifications 2. Prepare for Test Execution 3. Conduct System Integration Testing 4. Evaluate Testing 5. Establish Product Baseline
Acceptance Testing	To obtain ISS user acceptance that the system meets the requirements.	1. Plan for Acceptance Testing 2. Conduct Training for Acceptance Testing 3. Prepare for Acceptance Test Execution 4. ISS Evaluate Testing 5. Obtain Customer Acceptance Sign-off
Delivery	To deploy the system into production (ISS standalone server) environment.	1. Software must be packed by following ISS's standard. 2. Deployment guideline must be provided in ISS production (ISS standalone server) format. 3. Production (ISS standalone server) support and troubleshooting process must be defined.

Table: Methods and Standards

Team Formation & Registration

Team Name: AI Predictors
Project Title (repeated): Intelligent Dengue Predictor
System Name (if decided): Intelligent Dengue Predictor
Team Member 1 Name: Bai Sihai
Team Member 1 Matriculation Number: A0229986E
Team Member 1 Contact (Mobile/Email): 9656 6401 / e0687394@u.nus.edu
Team Member 2 Name: Lu Zhiping
Team Member 2 Matriculation Number: A0229959E
Team Member 2 Contact (Mobile/Email): 96756380 / lu.zhiping@u.nus.edu
Team Member 3 Name: Men Jinlong
Team Member 3 Matriculation Number: A0229972N
Team Member 3 Contact (Mobile/Email): 82369122/ jinlong.men@u.nus.edu
Team Member 4 Name: Ng Kwee Boon
Team Member 4 Matriculation Number: A0010308B
Team Member 4 Contact (Mobile/Email): 9297 5582 / kweeboon.ng@u.nus.edu

For ISS Use Only		
Programme Name:	**Project No:**	**Learner Batch:**
Accepted/Rejected/KIV:		
Learners Assigned:		
Advisor Assigned: Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg		

8 Appendix B: Functionalities to Knowledge/Techniques/Skills Map

Primary App Features/ Design Concepts	Courses/ Concept
System Design	• **Knowledge Acquisition** – domain expert experience • **Knowledge representation** (inference diagram, attribute table) • **Client-Server Web based application design**
Rule-based Engine	• **Machine Inference** • **Rule inference** • **Inference under uncertainty** (certainty factor) • **Gradient Boosting Algorithm**
Webapp frontend	• **System-user dialogue**
Data Collection/ Pre-processing	• **Machine Reasoning** • **Feature Engineering**
Model Selection/ Tuning/ Validation	• **Machine Learning** • **Cognitive System** • **Linear Regression** • **ElasticNet** • **Decision Tree Regressor** • **RandomForest Regressor** • **GradientBoostingMachine Regressor** • **LightGBM**

Table 4: Functionalities to Knowledge/Techniques/Skills Map

9 Appendix C: Installation and User Guide

9.1 Installation

9.1.1 Pre-requisite:

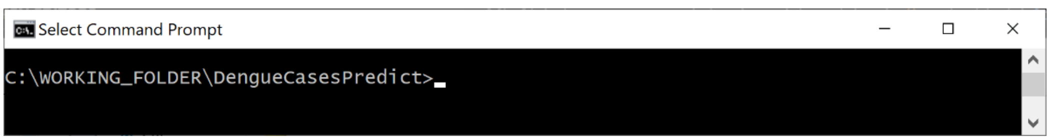
Python Version 3.9.0 must be installed in your computer environment to avoid any potential installation incompatibility of the packages/libraries. Otherwise, go to [Download Python | Python.org] to download Python and install it in your computer.

It is also assumed that the 'pip' package will be automatically installed when you install Python.
(Pip is a Package manager for Python which will be used to load in modules/libraries into the execution environments that will be used by the Dengue Cases Predict web application).

Note: This installation guide is for Windows 10 Operating System environment only.

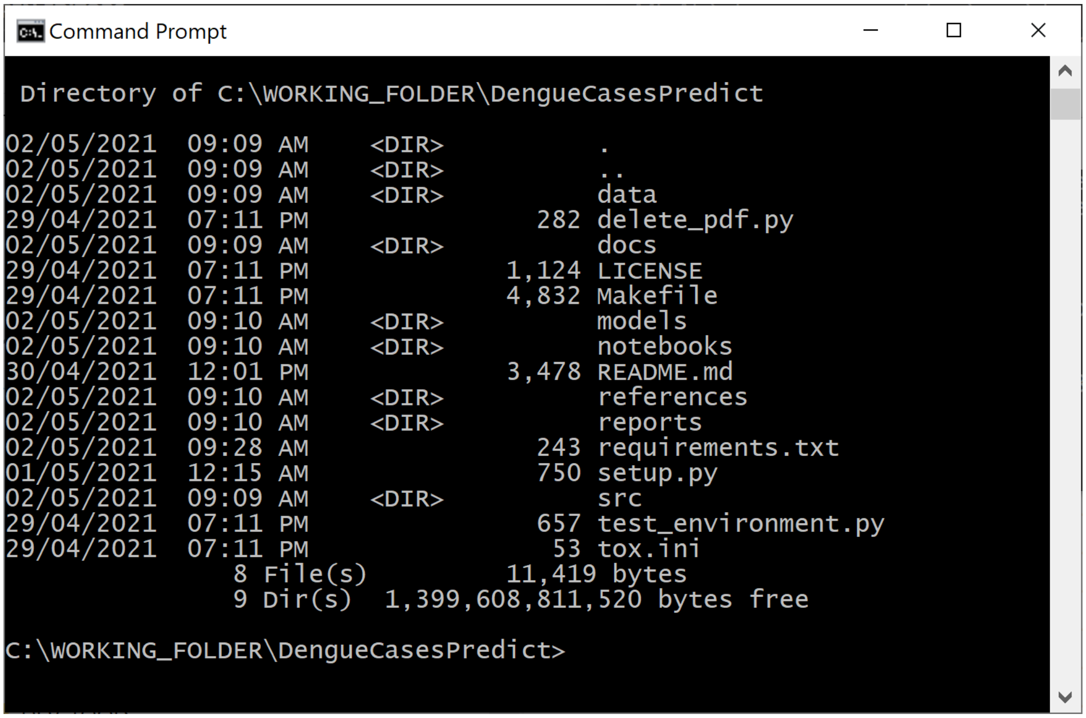
9.1.2 Installation

1. Open a Command Prompt Window and create a Project folder to store the application source code. E.g. DengueCasesPredict.

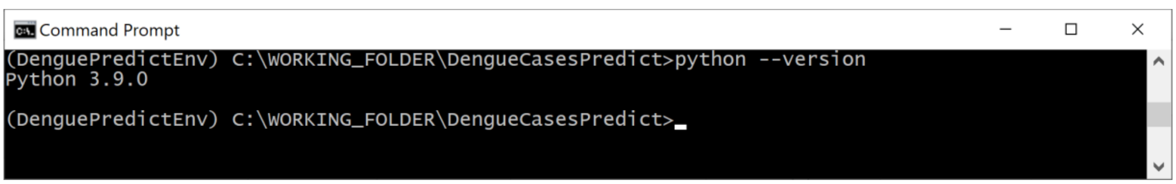


Copy the Project development folders and files into the newly created Project folder 'DengueCasesPredict'.

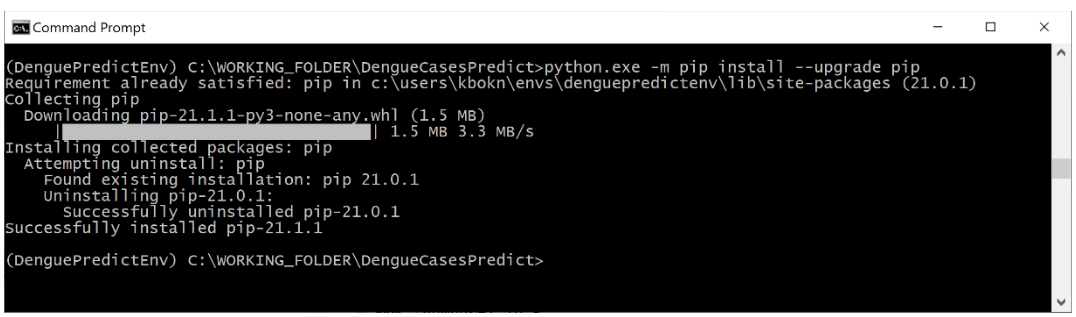
Note:
a. These files are from the project src/dev zip file that was submitted in the project submission.
b. Alternatively, if you clone the project folder from git, you can use the cloned folder for these installation steps also.



Type 'python –version' in the command prompt to check the Python version that it is 3.9.0



Type 'python.exe -m pip install --upgrade pip' in the command prompt to upgrade the pip package for your python.

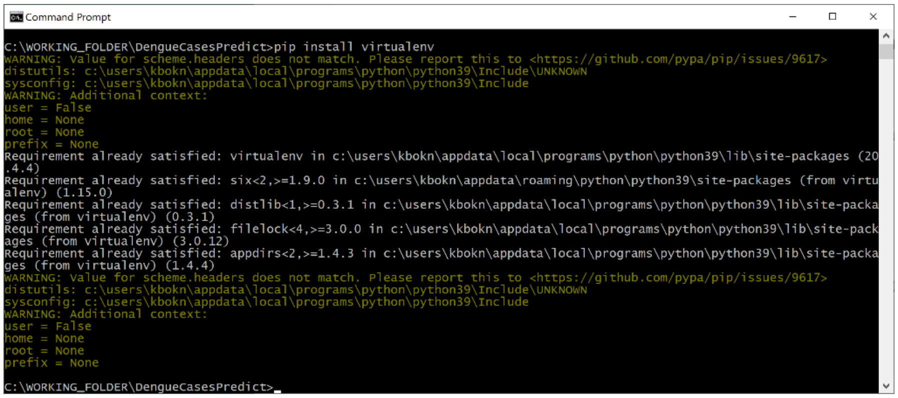


2. Install virtualenv for Python if it is not already installed.

   Type 'pip install virtualenv' into the command prompt window.

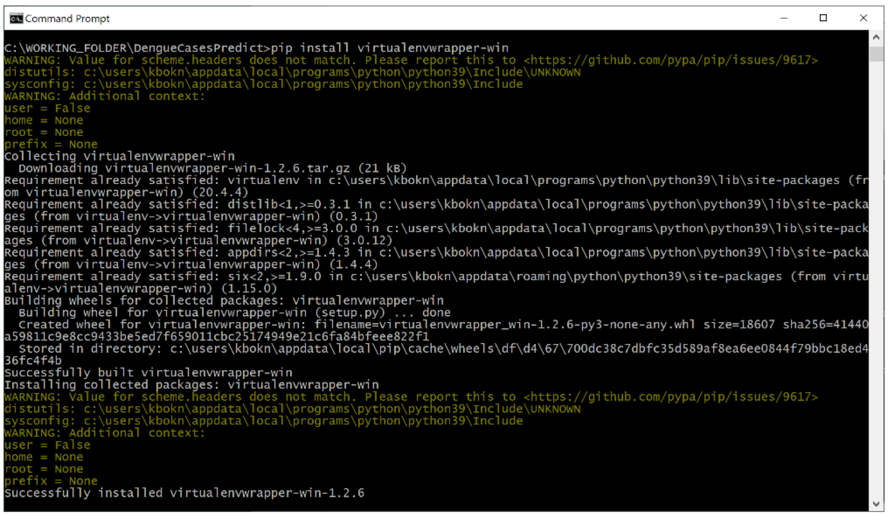
   The virtualenv will be used to create a standalone environments to run the web application in Python.

   If virtualenv was already installed, it will display the following:



3. Install virtualenvwrapper-win:
   Type 'pip install virtualenvwrapper-win' into the command prompt window.

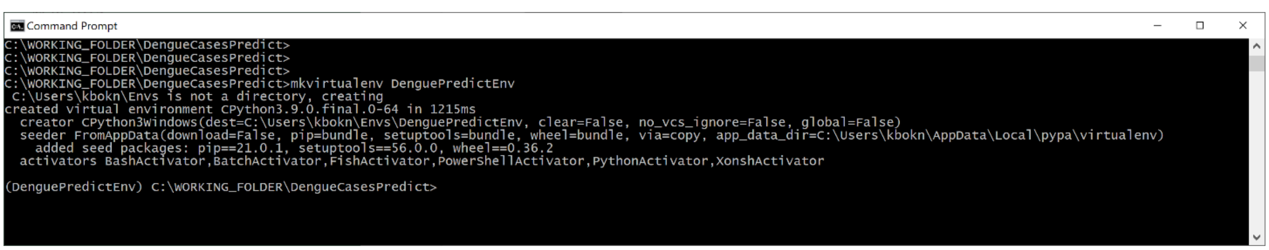
   You will see the following after successful installation of the package.



4. Make a Virtual Environment:

   Type 'mkvirtualenv DenguePredictEnv' into the command prompt window.

   It will create a and activate the Python virtual environment which is indicated with the (DenguePredictEnv) on the left side of the command prompt.



5. Set Project Directory:

   Type 'setprojectdir .\' into the command prompt window.

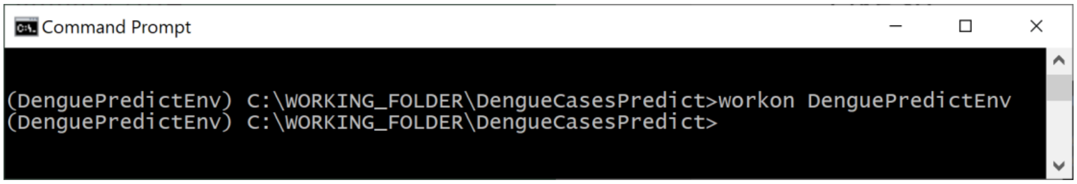
   This is to bind the DenguePredictEnv virtualenv with the current working directory.

Next time when the Python virtual environment is activated again, it will automatically move into this directory.

6. Workon:

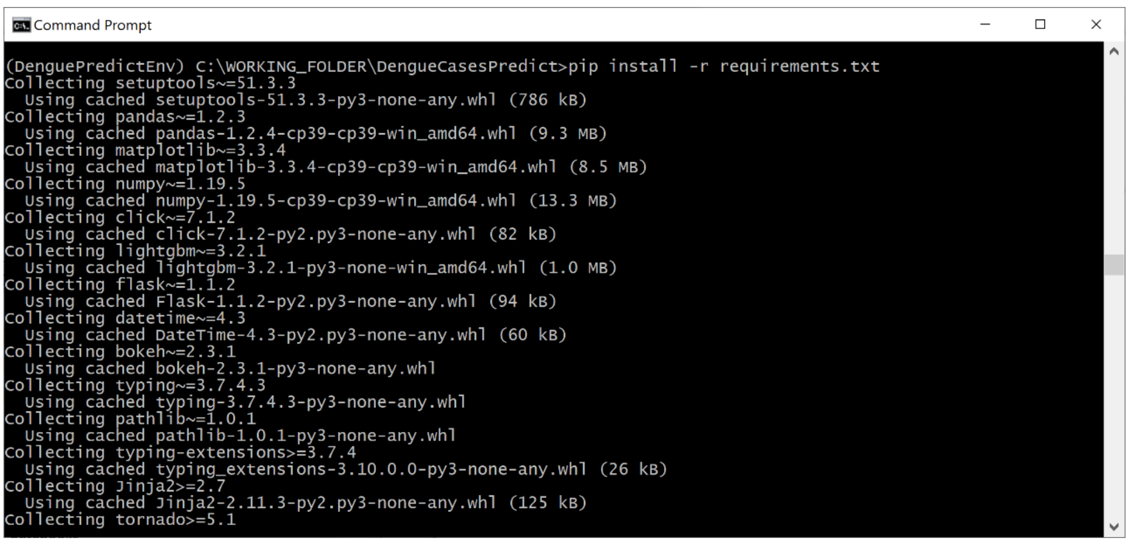
   Type 'workon DenguePredictEnv' into the command prompt window.

   This is to activate the environment and move into the root project folder.



7. Install the Packages/Libraries to be used by the Project:

   Type 'pip install -r requirement.txt' into the command prompt window.



9.2 Running the application on Local Host

1. Go to the Python virtual environment that you had setup if you are not already in your project virtual environment.

   Type 'workon DenguePredictEnv' in the command prompt window

   Note: If you want to know the Python virtual environment setup in your system,

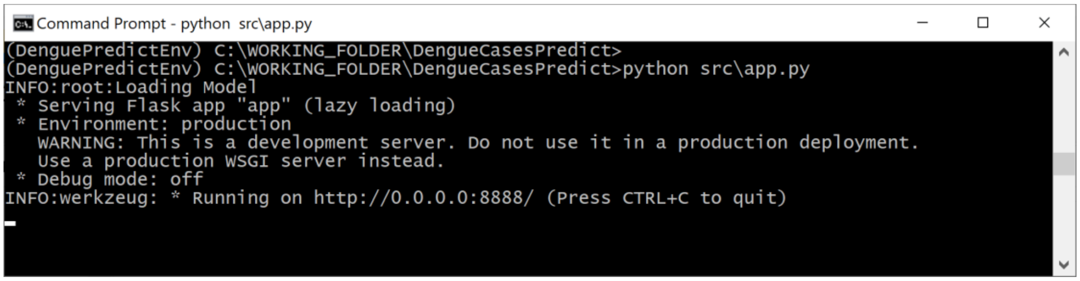
   Type 'lsvirtualenv' in the command prompt window.

   This script (lsvirtualenv) is normally in:
   C:\Users\<your username>\AppData\Local\Programs\Python\Python39\Scripts

2. Run/Start the Web Application for the Dengue Cases Prediction:

   Type 'python src\app.py' into the command prompt window.

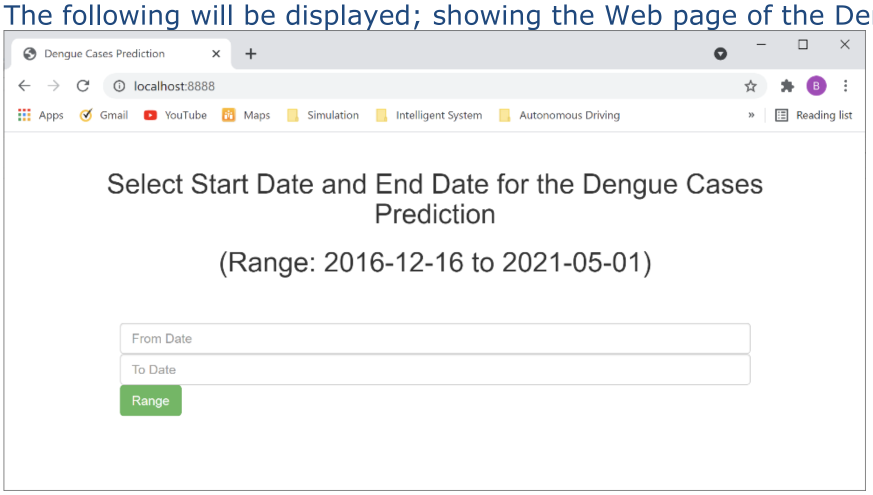
   The following will be displayed, showing that the Web Server is running
   Note: You may have to wait for more than 10 seconds before the application start is completed.



3. Open a Web Browser in your computer:

   Type 'localhost:8888'

   The following will be displayed; showing the Web page of the Dengue Cases Predict:



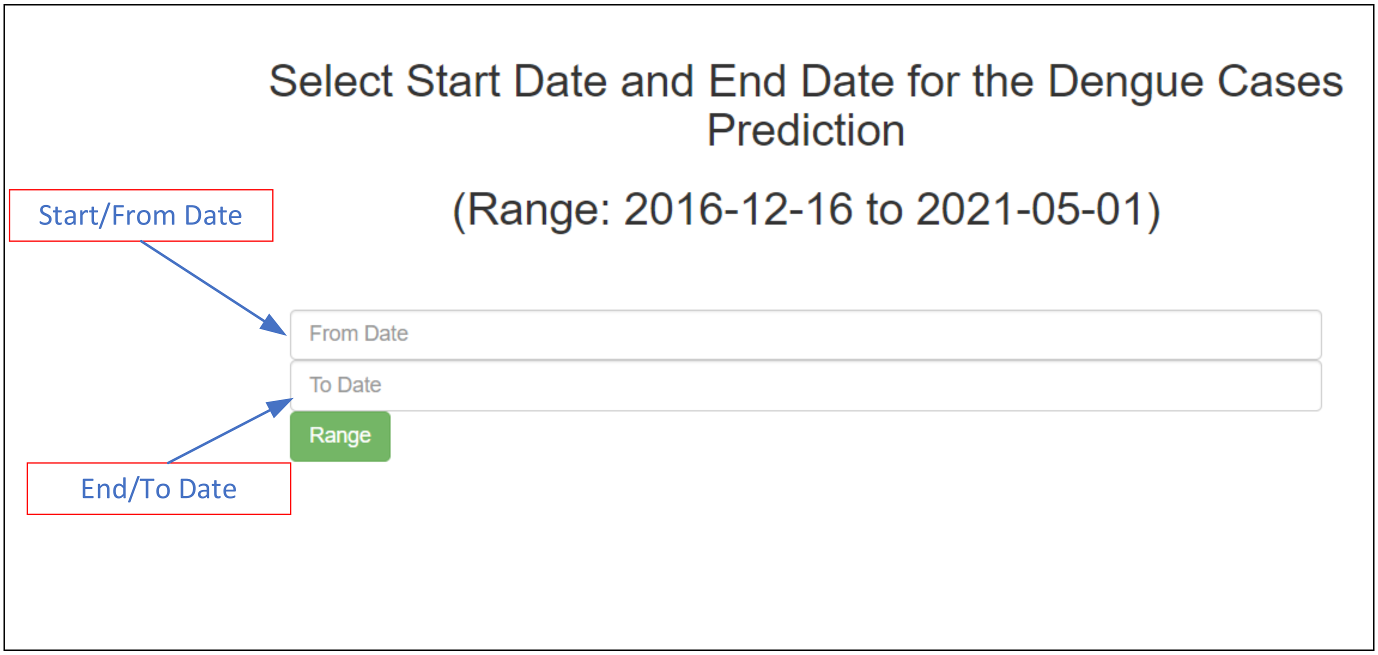
   Refer to the 9.3 User Interface below for the operation.

9.3 User Interface

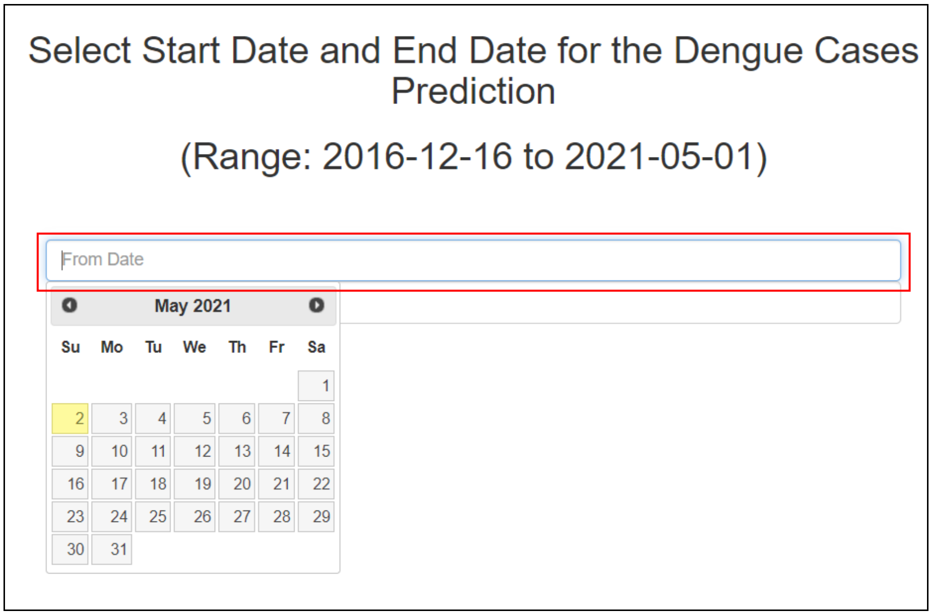
Below are the user interfaces and its operation that the Client will interact with from his/her Web Browser.

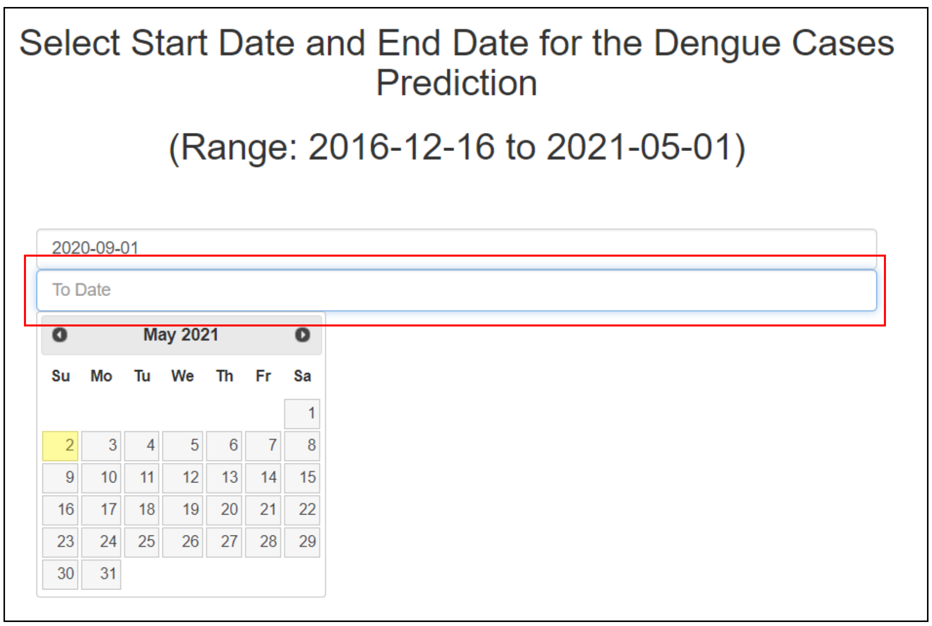
Pre-condition: Web server is started with the FLASK web application.

1. Start page (http://127.0.0.1:8888)

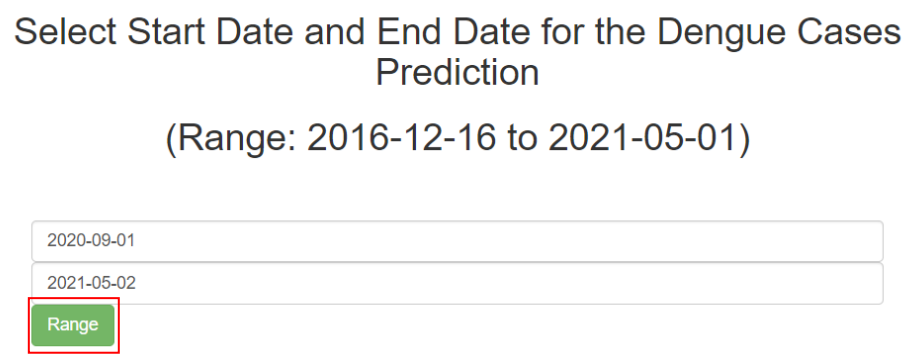


2. Select "From" Date and "To" Date

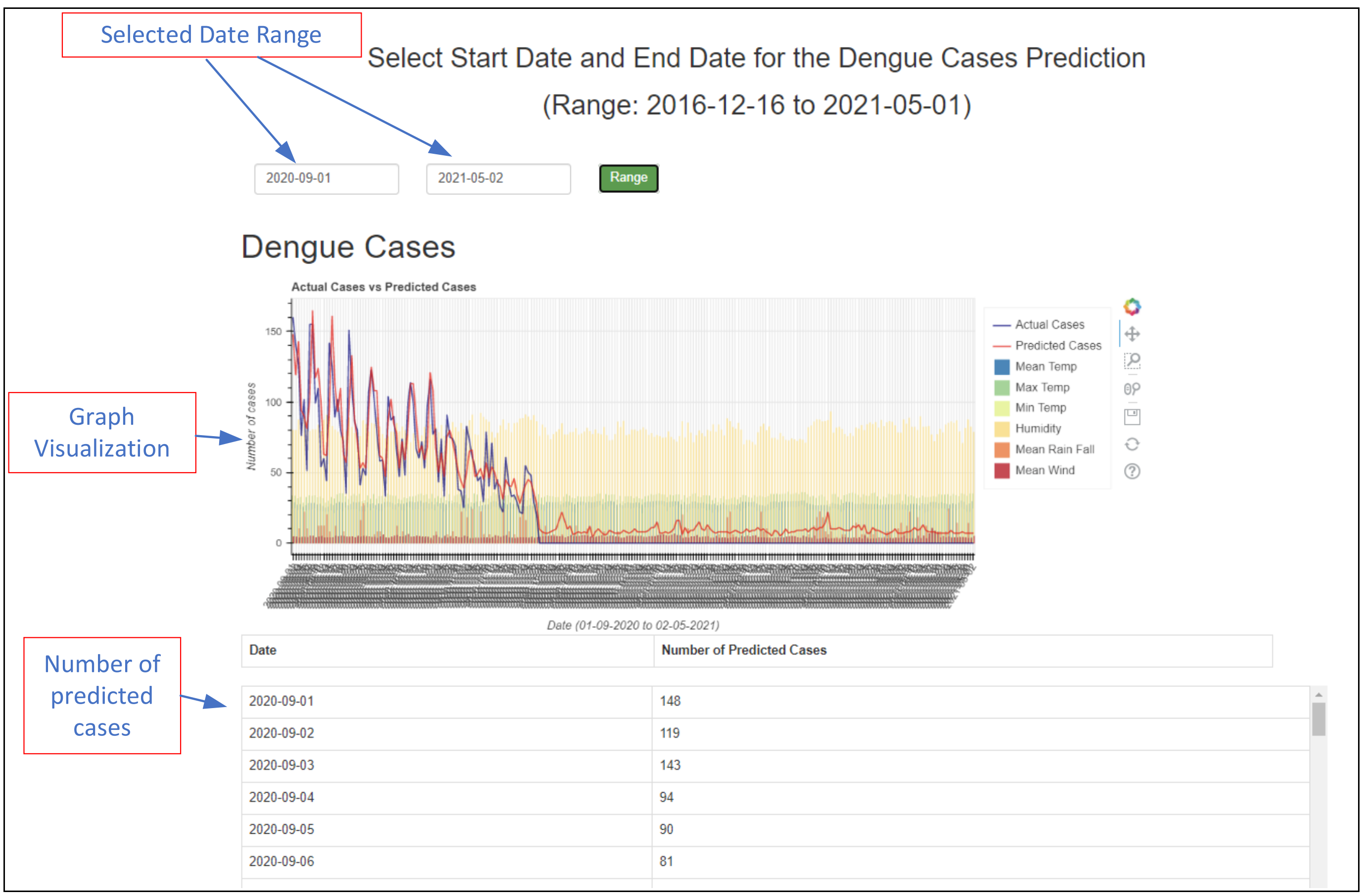




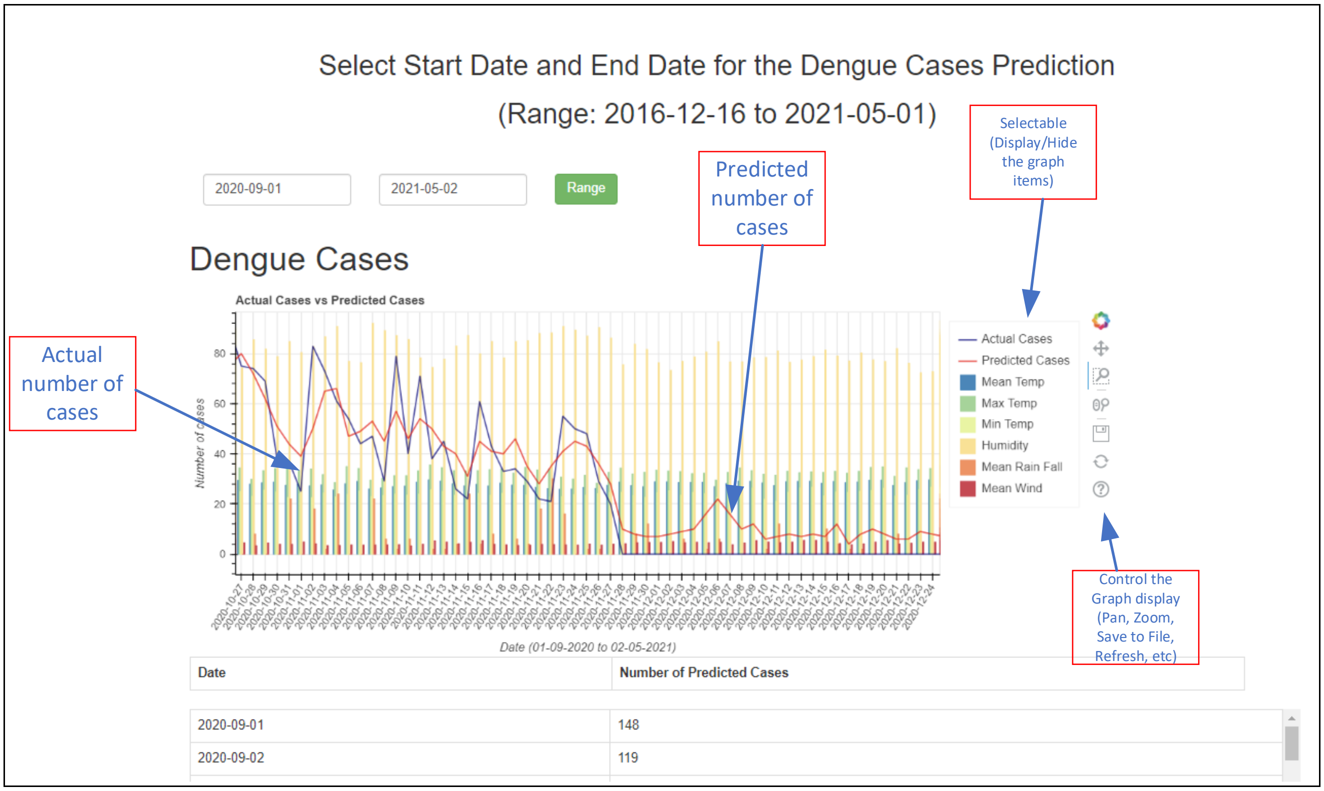
3. Click on "Range" Button to generate the prediction.



4. Result page of the Number of Dengue Cases prediction



5. Zoom in view of the Result page of the Number of Dengue Cases prediction



9.4 Troubleshooting

1. Dates formatting

   Due to the feature and target data are collected from different resources, there is formatting problem in the dataset After we well defined the dataset formatting, we have fixed this problem.

2. Feature Problem

   The original weather data was collected from Data.gov.sg, including daily mean temperature, maximum temperature, minimum temperature, relative humidity, rain fall and wind speed, total 6 features. Some trial runs were made to explore the dengue cases prediction model. We have tried several machine learning algorithms, but the performance is not satisfied. The prediction results were flattened which was not able to catch the outbreak of the dengue cases.

   After some literature review, we found the dengue cases are caused by mosquito bite which highly correlated with the weather condition of past few days. And the more dengue cases in the community, the higher risk of the outbreak on dengue. In these concepts, feature engineering was applied to cover the weather data and dengue cases in the past few days, including moving average weather data, past weather data and cross-correlation dengue cases. After including these features, the prediction model accuracy has increased. The detailed comparison between the different features is presented in Section 3.