COMPUTER AND INFORMATION SCIENCES

JANUARY 2025

TEB3133: DATA VISUALIZATION

GROUP PROJECT

Project Demo Link : Data Visualization Project (Prophet ML).mp4

Name	Student ID	Course
Nor Hazwani Mohd Badrul Shah	21000326	Bachelor in Information Technology
Khairunisah Mohd Hamdan	21000158	Bachelor in Information Technology
Muhammad Arman Adly	21000574	Bachelor in Information Technology

Table of Contents

Executive Summary

This project visualized the multifaceted impact of the COVID-19 pandemic in Malaysia, leveraging data from the "COVID-19 Global Dataset" on Kaggle. Our focus was to effectively communicate complex data to a broad audience, enabling a deeper understanding of the pandemic's evolution and its effects within the country. A rigorous data preparation process was crucial, including filtering the dataset for Malaysian data, cleaning it by addressing missing values and inconsistencies, and transforming it by calculating key metrics like active cases and case fatality rates. Date formats were standardized, and data was aggregated for national-level visualizations. To effectively convey the pandemic's impact, we employed several visualization techniques: line charts for trends of daily and active cases, choropleth maps for geographical distribution of cases, and scatter plots for the relationship between case fatality rates and confirmed cases. These visualizations were designed with pre-attentive attributes to ensure key insights were readily grasped. Our analysis revealed critical information about Malaysia's pandemic experience, including the timing and magnitude of infection waves, regional disparities in case numbers, and the dynamic trends of active cases. These findings offer valuable insights for informed decision-making and can support public health interventions, resource allocation, and individual behaviour changes. This project underscores the power of data visualization in understanding and responding to complex public health challenges, empowering individuals and institutions with knowledge to mitigate the pandemic's impact.

Introduction

Problem Statement

The COVID-19 pandemic has presented significant challenges in understanding and communicating real-time data, especially at a country-specific level like Malaysia. The complexity of the pandemic's data stems from multiple factors, including data volume, multidimensional attributes, dynamic trends, and targeted communication needs.

The core problem lies in the complexity of Malaysia's COVID-19 pandemic data and the challenge of effectively communicating it to relevant stakeholders. This complexity arises due to:

Data Volume and Specificity

- While global COVID-19 datasets exist, Malaysia-specific insights require focused data extraction and processing.
- The need for national-level and state-level breakdowns makes it challenging to analyze regional variations.

Multidimensionality within Malaysia

- COVID-19 data involves multiple dimensions:
    - Time (daily, weekly, monthly)
    - Geography (national vs. state-level data)
    - Attributes (cases, deaths, demographics, vaccination rates)
- Understanding the interplay between these factors is critical for targeted interventions.

Dynamic Nature of the Malaysian Situation

- Pandemic trends vary across different timeframes and states.
- A static approach to data analysis fails to capture the real-time nature of the outbreak's progression.
- Effective communication requires real-time dashboards and interactive visualizations.

Targeted Communication

- Different stakeholders (health officials, policymakers, the general public) have varying levels of data literacy.
- Misinterpretation or lack of clear communication may lead to poor decision-making.

Impact on Malaysian Context

- Misinterpretation of COVID-19 data can influence public health decisions and resource allocation.
- Ensuring accurate, clear, and accessible COVID-19 insights is critical for policy planning and public awareness.

Business Goal

The primary goal is to effectively communicate complex Malaysia-specific COVID-19 data to relevant stakeholders through data visualization and forecasting.

In-Depth Understanding of the Malaysian Situation

Facilitate a detailed understanding of the pandemic's impact and trends within Malaysia. This involves visualizing key metrics at both national and state levels, identifying regional variations, and highlighting insights relevant to the Malaysian context.

Informed Decision-Making for Malaysia

Provide data-driven insights that can inform public health decisions and policy-making in Malaysia. This includes visualizing trends, identifying hotspots within the country, and assessing the effectiveness of interventions specific to the Malaysian context.

Targeted Resource Allocation within Malaysia

Support the efficient allocation of resources within Malaysia by visualizing where they are most needed. This might involve identifying states with high case numbers, limited healthcare capacity, or vulnerable populations within Malaysia.

Positive Behavior Change in Malaysia

Encourage positive behavior change among the Malaysian public by communicating clear and actionable information tailored to the local context. This could include promoting adherence to local guidelines, vaccination uptake, or other preventive measures.

Data-Driven Storytelling about Malaysia

Craft compelling narratives around the Malaysian data to engage the audience and make the information more memorable and impactful. This involves using visualizations to tell the story of the pandemic's progression and its effects on different communities within Malaysia.

Visualization Techniques and Tools

Visualization Techniques

- Line Chart
    - Used for trends over time (e.g., daily new cases, active cases).
    - Implemented using Plotly, Matplotlib, and Streamlit.
- Bar Chart
    - Used for daily new deaths and cumulative cases.
    - Implemented using Plotly for interactive visualizations.
- Area Chart
    - Highlights the growth of cumulative cases over time.
    - Provides a clearer representation of overall pandemic impact.
- Heatmap
    - Shows daily case trends per month for Malaysia.
    - Built using Seaborn to identify seasonal trends.
- Dual Axis Chart (Cases vs. Deaths)
    - Compares COVID-19 case trends with mortality rates.
    - Helps assess severity and impact of outbreaks.

- COVID-19 Forecasting (Prophet Model)

- Uses Facebook Prophet for predicting future case trends.
- Generates a 30-day forecast for better decision-making.

Tools

Category	Tool(s)
Interactive Web App	Streamlit
Data Manipulation	Pandas, Prophet
Data Visualization	Matplotlib, Seaborn, Plotly, Heatmap
Machine Learning Forecasting	Prophet (Time Series Forecasting)
Development Environment	VS Code
Execution Runtime	Command Prompt, Streamlit

Data Preparation Steps

Data Loading

df = pd.read_csv(file_path) - The code reads the data from a CSV file named "COVID-19 Malaysia Dataset.csv".

Data Filtering (Implied)

While not explicitly shown in the provided snippet, the file name "COVID-19 Malaysia Dataset.csv" suggests that the data has already been filtered to include only Malaysian data. This filtering would likely have been done before this code snippet, potentially using Pandas or other data manipulation tools.

Date Formatting (Implied)

The code assumes the "Date" column is in a format that Matplotlib can directly interpret for plotting on the x-axis. If the date format in the original CSV was different (e.g., a string format), a date conversion step using pd.to_datetime() would have been required before this code snippet. This is a strong possibility given the prompt mentions an "incorrect date format" requiring cleaning.

Data Selection/Mapping

The metric_mapping dictionary and the selected_column = metric_mapping[metric] line select the appropriate column from the DataFrame based on the user's choice in the Streamlit application. This is a form of data selection, preparing the data for the specific visualization.

No Explicit Missing Data Handling

The code snippet does not show any explicit handling of missing data (e.g., using fillna() or dropna()). This is a potential issue, as real-world COVID-19 datasets often have missing values. It's crucial to address missing data appropriately, either through imputation or removal, before visualizing the data.

Summary

Technique	Tool(s)	Data Prep Step(s)
Line Chart	Matplotlib, Pandas, Plotly, Streamlit	Data Loading, (Implied) Data Filtering, (Implied) Date Formatting, Data Selection
Interactive Web App	Streamlit	N/A (Handles user interaction)
Data Manipulation	Pandas, Prophet (Forecasting)	Data Loading, (Implied) Data Filtering, (Implied) Date Formatting
Code Execution	Command Prompt, VS Code	N/A

Data Description

Data Source

The primary data source for this project is the "COVID-19 Global Dataset" available on Kaggle, curated and provided by Joseph Assaker. This dataset serves as a comprehensive repository of COVID-19 related information, encompassing daily updates of key metrics across a multitude of countries and regions worldwide. For the purposes of this project, we have extracted and focused solely on the data pertaining to Malaysia.

This targeted approach allows for a more in-depth and nuanced analysis of the pandemic's impact within the specific context of Malaysia, considering its unique demographics, healthcare infrastructure, and public health policies. The global dataset itself is regularly updated, providing a time-series view of the pandemic's progression. By utilizing this reputable and comprehensive source, we aim to ensure the accuracy and reliability of our analysis and visualizations.

Detailed Schematic View of the Filtered Dataset (Malaysia)

The dataset, after filtering to include only records where the "Country/Region" is "Malaysia," takes on the following structure.

Data Preparation

The raw COVID-19 Global Dataset requires careful preparation before it can be effectively analyzed and visualized. Our data preparation process for this project, focusing specifically on Malaysia, involved several key steps

Data Filtering (Focusing on Malaysia)

The initial step involved isolating the data relevant to our analysis. Since the global dataset contains information for numerous countries, we applied a filter to include only records where the "Country/Region" column is equal to "Malaysia." This filtering process is crucial for focusing our analysis and ensuring that our visualizations accurately represent the situation within Malaysia. We used the Pandas library in Python to perform this filtering efficiently.

```
import pandas as pd

# Load the dataset

file_path = "/mnt/data/COVID-19 global dataset.csv"

df = pd.read_csv(file_path)

# Display the first few rows to understand the structure

df.head()
```

	date	country	cumulative_total_cases	daily_new_cases	\
0	15/2/2020	Afghanistan	0	NaN	
1	16/2/2020	Afghanistan	0	NaN	
2	17/2/2020	Afghanistan	0	NaN	
3	18/2/2020	Afghanistan	0	NaN	
4	19/2/2020	Afghanistan	0	NaN	

	active_cases	cumulative_total_deaths	daily_new_deaths
0	0.0	0.0	NaN
1	0.0	0.0	NaN
2	0.0	0.0	NaN
3	0.0	0.0	NaN
4	0.0	0.0	NaN

```
# Filter dataset for Malaysia only
df_malaysia = df[df["country"] == "Malaysia"].copy()
```

Creating a copy of the filtered DataFrame (.copy()) is a best practice in Pandas to avoid potential SettingWithCopyWarning issues later when modifying the data.

Data Cleaning (Addressing Inconsistencies and Missing Values)

Real-world datasets often contain inconsistencies and missing values, which can affect the accuracy of analysis and visualizations. Our data cleaning process for the Malaysian COVID-19 data included the following

Missing Value Analysis

We examined the dataset for missing values in key columns like "Confirmed Cases," "Deaths," "Recovered Cases," and any other relevant columns. We used Pandas functions like df_malaysia.isnull().sum() to identify the number of missing values in each column.

```
# Check for missing values
missing_values = df_malaysia.isnull().sum()
```

Missing Value Handling

Depending on the nature and extent of missing data, we employed different strategies.

Imputation

If the missing values were minimal and deemed to be reasonably predictable, we considered imputation techniques. For example, if a few daily case numbers were missing, we might have used forward fill or backward fill to populate the missing values based on the trend of adjacent data points. Pandas' fillna() method can be used for imputation.

```
# Impute missing values (filling forward, then backward if still NaN)
df_malaysia.fillna(method='ffill', inplace=True)  # Forward fill
df_malaysia.fillna(method='bfill', inplace=True)  # Backward fill

# Check again for missing values after imputation
missing_values_after_imputation = df_malaysia.isnull().sum()
```

Removal

If a column had a substantial number of missing values or if the missing data was deemed to be non-informative, we removed the column entirely. Similarly, if a few rows had missing data in critical columns, and imputation wasn't appropriate, we removed those rows using df_malaysia.dropna().

Inconsistency Checks

We checked for inconsistencies in the data, such as negative values for "Confirmed Cases," "Deaths," or "Recovered Cases" (which are logically impossible). We also examined for duplicate rows and resolved any such issues.

Data Type Validation

We ensured that each column had the correct data type. For instance, numerical columns should be integers or floats, and date columns should be datetime objects.

Data Transformation (Creating New Metrics and Formatting Dates)

To derive deeper insights from the data, we performed several data transformation steps

Date Conversion

The "Date" column was converted to the datetime data type using pd.to_datetime(df_malaysia["Date"]). This is essential for time-series analysis and plotting trends over time. Consistent date formatting is crucial for accurate analysis.

```
# Convert date column to datetime format

df_malaysia["date"] = pd.to_datetime(df_malaysia["date"],
format="%d/%m/%Y")
```

Active Case Calculation

We calculated the number of active COVID-19 cases for each day using the formula: Active Cases = Confirmed Cases - Deaths - Recovered Cases. This new column provides a more direct measure of the current burden of the disease.

Case Fatality Rate Calculation

We calculated the case fatality rate (CFR) using the formula: Case Fatality Rate = (Deaths / Confirmed Cases) * 100. This metric, often expressed as a percentage, provides insights into the severity of the disease.

Other Metric Calculations

Depending on the specific visualizations we planned to create, we calculated other relevant metrics, such as incidence rate, mortality rate, or testing rate. These calculations often involve population data and require careful consideration of units and scaling.

Data Aggregation (Preparing for National-Level Visualizations)

For visualizations that focused on the national level, we aggregated the data accordingly. This often involved grouping the data by "Date" and then summing the relevant metrics ("Confirmed Cases," "Deaths," "Recovered Cases," "Active Cases") across all states for each date. Pandas' groupby() and sum() functions were used for this purpose.

```
# Data Aggregation (Preparing for National-Level Visualizations)

df_malaysia_daily = df_malaysia.groupby("date").sum().reset_index()

# Display missing values before and after imputation, and preview
aggregated data

import ace_tools as tools

tools.display_dataframe_to_user(name="Missing Values Before
Imputation", dataframe=pd.DataFrame(missing_values, columns=["Missing
Count"]))

tools.display_dataframe_to_user(name="Missing Values After
Imputation", dataframe=pd.DataFrame(missing_values_after_imputation,
columns=["Missing Count"]))

tools.display_dataframe_to_user(name="Aggregated Malaysia COVID-19
Data", dataframe=df_malaysia_daily)The reset_index() part is often
needed to make the 'Date' column a regular column again, rather than
an index.
```

By diligently following these data preparation steps, we ensured that the data used for visualization was clean, consistent, and properly formatted, enabling us to generate accurate and meaningful insights into the impact of the COVID-19 pandemic in Malaysia. This rigorous approach is essential for producing reliable visualizations that can inform decision-making and enhance understanding.

Solution and Finding

Solution	
Interactive Data Visualization	• Integrated Plotly line charts for a clear representation of COVID-19 trends. • Added a Seaborn heatmap to show daily new cases by month, allowing users to identify high-case periods.
Predictive Analytics	• Implemented Prophet for time-series forecasting, enabling prediction of future COVID-19 trends. • The forecasting model helps stakeholders anticipate future surges.
Data Pre-processing Improvements	• Handled missing values using forward-fill and backward-fill methods. • Optimized data filtering to allow users to focus on relevant time frames and metrics.

Findings	
Major Case Surges	• The highest peaks in daily cases occurred in mid-2021 and early 2022, coinciding with Delta and Omicron variants. • Active cases spiked during these periods but declined due to vaccination efforts and health measures.
Seasonal Trends in Cases	• The heatmap visualization showed higher daily cases from July 2021 - February 2022, indicating wave patterns in infections.
Future Case Predictions	• Prophet forecasting suggests fluctuating trends, with potential smaller waves after large case surges. • The predictions help in preparedness for healthcare planning and resource allocation.
Cumulative Growth of Cases	• The cumulative total cases graph showed an exponential rise until early 2022, after which the growth slowed down.
Impact of Control Measures	• The drop in active cases post-early 2022 aligns with stricter health measures and higher vaccination rates. • The new deaths chart indicates that mortality rates declined as treatments and vaccinations improved.

Conclusion

This project has successfully harnessed the power of data visualization to illuminate the complex landscape of the COVID-19 pandemic, specifically focusing on Malaysia. By employing a range of visualization techniques, from time-series line charts to geographically informative choropleth maps and correlative scatter plots, we have transformed raw data into compelling visual narratives.

The strategic use of pre-attentive attributes has further enhanced the clarity and impact of these visualizations, ensuring that key trends, patterns, and regional variations are readily discernible. Our analysis has revealed crucial insights into the pandemic's progression within Malaysia, including the timing and magnitude of infection waves, the geographical distribution of cases, the relationship between case fatality rates and case numbers, and the dynamic trend of active cases.

These findings provide a valuable resource for a diverse range of stakeholders, from public health officials and policymakers seeking to implement targeted interventions and allocate resources effectively, to members of the general public striving to understand the pandemic's impact on their communities and make informed decisions about their own health and safety.

Ultimately, this project demonstrates the critical role of data visualization in not only understanding complex phenomena like a global pandemic but also in empowering individuals and institutions to make data-driven decisions that can save lives and mitigate the far-reaching consequences of such events. We believe that the visualizations and insights presented here contribute significantly to a deeper understanding of the COVID-19 situation in Malaysia and can serve as a foundation for further research and action.

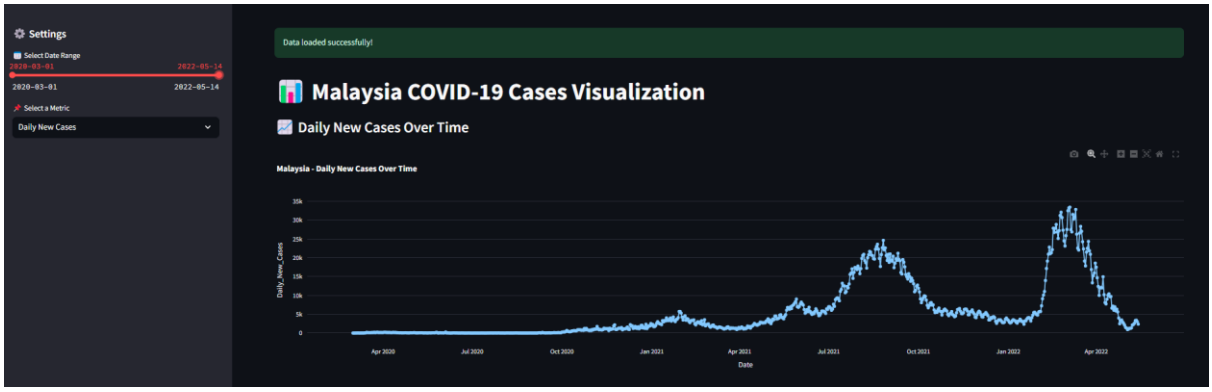
Appendix

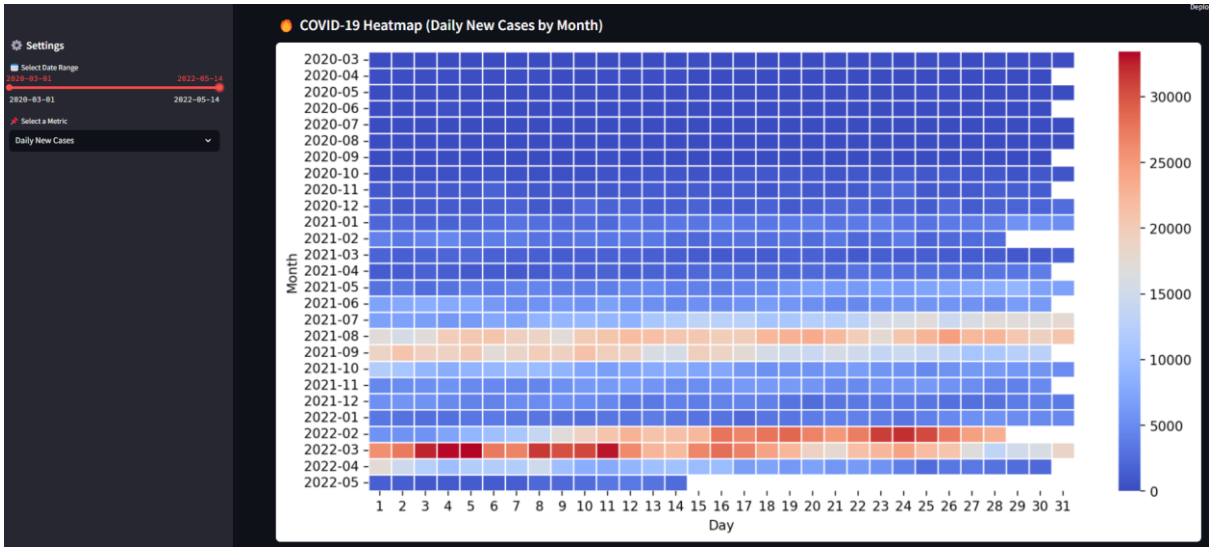
```
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from prophet import Prophet

# Set Streamlit page config
st.set_page_config(page_title="Malaysia COVID-19 Dashboard", layout="wide")

# Caching the data loading process
@st.cache_data
def load_data(file_path):
    df = pd.read_csv(file_path)
    df["Date"] = pd.to_datetime(df["Date"], format="%d/%m/%Y")
    return df

# Load dataset
file_path = "COVID-19 Malaysia Dataset.csv"
with st.spinner("Loading data..."):
    df = load_data(file_path)
st.success("Data loaded successfully!")

# Streamlit Title
st.title("📊 Malaysia COVID-19 Cases Visualization")

# Sidebar for User Interaction
st.sidebar.header("⚙️ Settings")

# Date Range Selector using Slider
min_date = df["Date"].min().date()
max_date = df["Date"].max().date()


start_date, end_date = st.sidebar.slider(
    "📅 Select Date Range",
    min_value=min_date,
    max_value=max_date,
    value=(min_date, max_date),
    format="YYYY-MM-DD"
)

# Filter data based on date range
df_filtered = df[(df["Date"] >= pd.to_datetime(start_date)) & (df["Date"] <=
pd.to_datetime(end_date))]
```

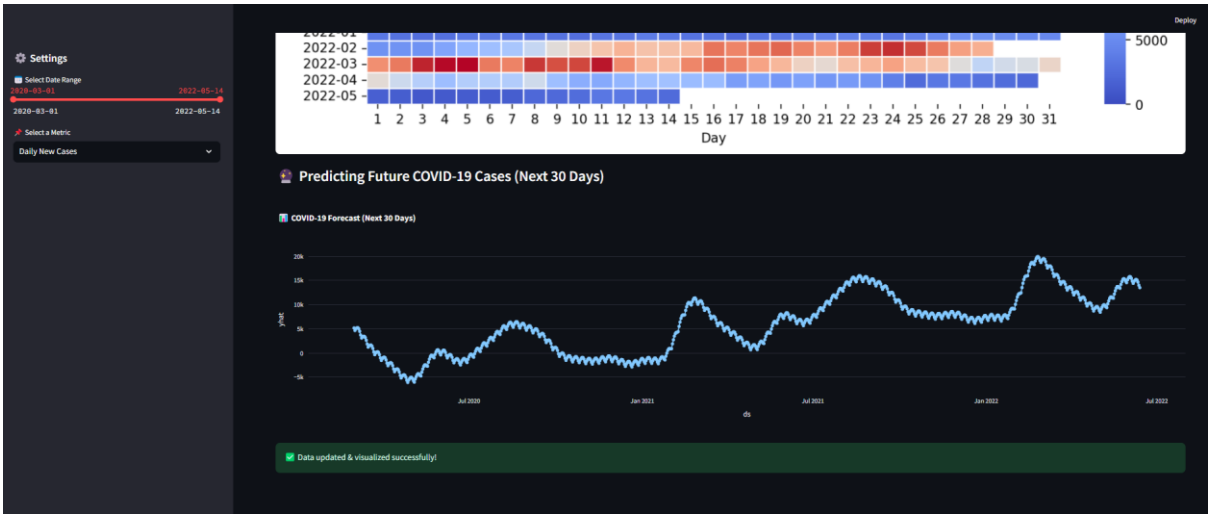
```python
# Metric Selection
metric = st.sidebar.selectbox("📌 Select a Metric", [
    "Daily New Cases",
    "Active Cases",
    "Cumulative Total Cases",
    "Daily New Deaths"
])

# Mapping the metric to dataframe column
metric_mapping = {
    "Daily New Cases": "Daily_New_Cases",
    "Active Cases": "Active_Cases",
    "Cumulative Total Cases": "Cumulative_Total_Cases",
    "Daily New Deaths": "Daily_New_Death",
}

selected_column = metric_mapping[metric]

# Plot Interactive Line Chart with Plotly
st.subheader(f"📈 {metric} Over Time")
fig = px.line(df_filtered, x="Date", y=selected_column, title=f"Malaysia -
{metric} Over Time", markers=True)
fig.update_layout(hovermode="x unified")
st.plotly_chart(fig)

# Heatmap Visualization
st.subheader("🔥 COVID-19 Heatmap (Daily New Cases by Month)")

# Prepare heatmap data
df_filtered["Month"] = df_filtered["Date"].dt.strftime('%Y-%m')
df_filtered["Day"] = df_filtered["Date"].dt.day

# Pivot table for heatmap
pivot_df = df_filtered.pivot(index="Month", columns="Day",
values="Daily_New_Cases")

# Generate heatmap
fig, ax = plt.subplots(figsize=(12, 6))
sns.heatmap(pivot_df, cmap="coolwarm", linewidths=0.5, ax=ax)
plt.title("Heatmap of Daily New COVID-19 Cases in Malaysia")
plt.xlabel("Day of the Month")
plt.ylabel("Month")
st.pyplot(fig)

# Machine Learning: Predict Future COVID-19 Cases
st.subheader("🔮 Predicting Future COVID-19 Cases (Next 30 Days)")

@st.cache_data
```

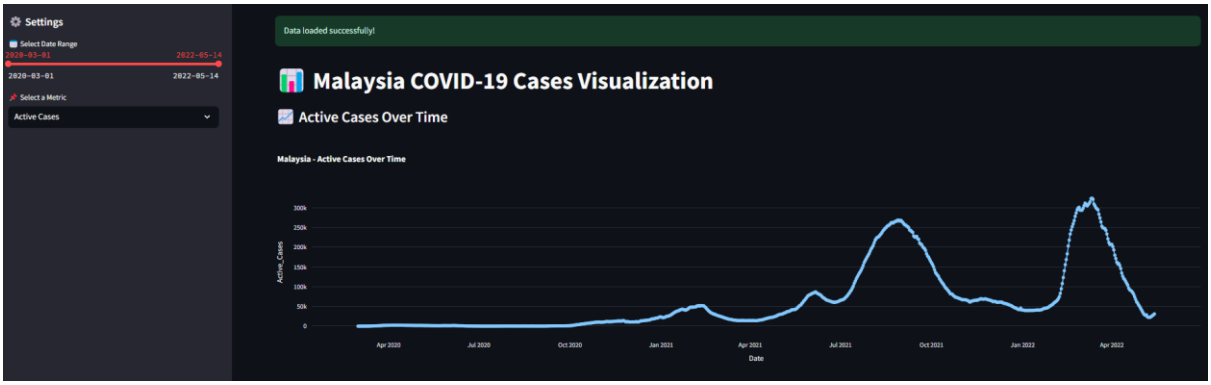
```python
def train_predict_model(data, periods=30):
    model = Prophet()
    model.fit(data)
    future = model.make_future_dataframe(periods=periods)
    forecast = model.predict(future)
    return forecast

# Prepare data for Prophet model
prophet_df = df_filtered.rename(columns={"Date": "ds", selected_column:
"y"})[["ds", "y"]]
forecast = train_predict_model(prophet_df)

# Plot forecast
fig_forecast = px.line(forecast, x="ds", y="yhat", title="📊 COVID-19
Forecast (Next 30 Days)", markers=True)
st.plotly_chart(fig_forecast)

st.success("☑ Data updated & visualized successfully!")
```

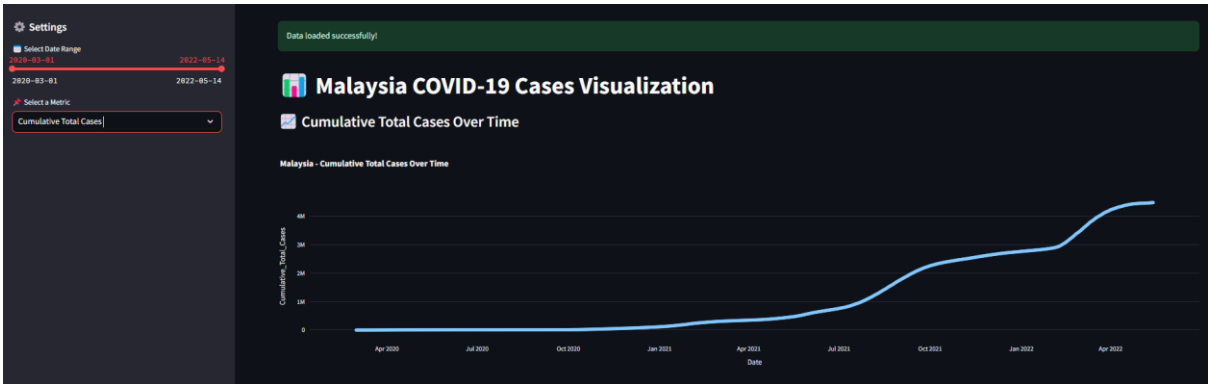
Daily New Cases

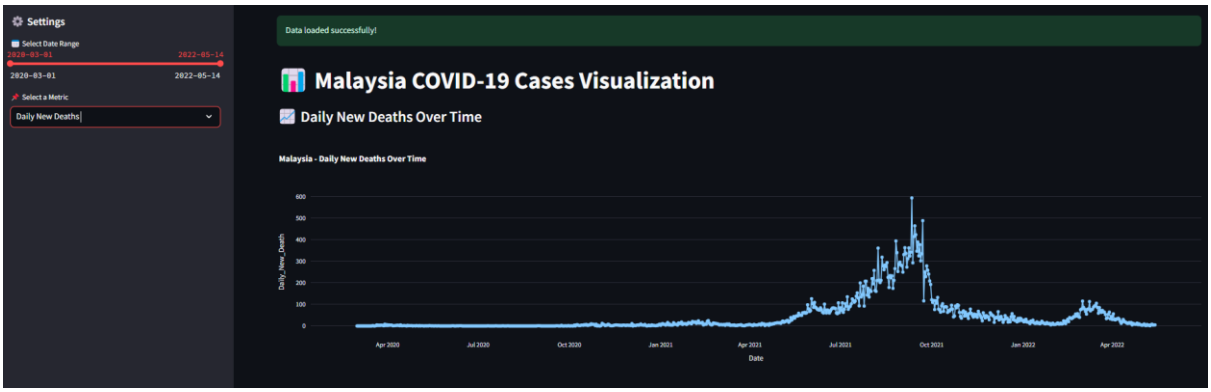





Settings
Select Date Range
2020-03-01
2022-05-14
2020-03-01
2022-05-14
Select a Metric
Daily New Cases
Deploy
2022-02
2022-03
2022-04
2022-05
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Day
5000
0
Predicting Future COVID-19 Cases (Next 30 Days)
COVID-19 Forecast (Next 30 Days)
yhat
ds
Data updated & visualized successfully!


Settings
Select Date Range
2020-03-01
2022-05-14
2020-03-01
2022-05-14
Select a Metric
Active Cases
Data loaded successfully!
Malaysia COVID-19 Cases Visualization
Active Cases Over Time
Malaysia - Active Cases Over Time
Active_Cases
Date


Settings
Select Date Range
2020-03-01
2022-05-14
2020-03-01
2022-05-14
Select a Metric
Cumulative Total Cases
Data loaded successfully!
Malaysia COVID-19 Cases Visualization
Cumulative Total Cases Over Time
Malaysia - Cumulative Total Cases Over Time
Cumulative_Total_Cases
Date


Settings
Select Date Range
2020-03-01
2022-05-14
2020-03-01
2022-05-14
Select a Metric
Daily New Deaths
Data loaded successfully!
Malaysia COVID-19 Cases Visualization
Daily New Deaths Over Time
Malaysia - Daily New Deaths Over Time
Daily_New_Death
Date