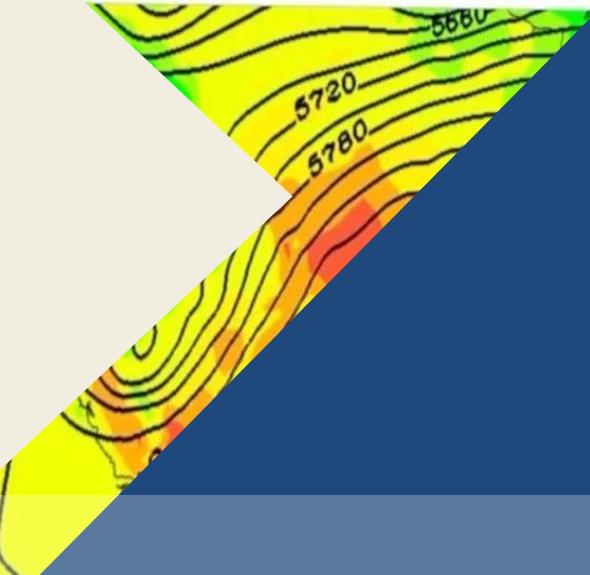


# Advancing Heat Resilience

# Heat Wave

Group ID - 2023-24-114



# Meet Our Team



IT20142728  
Ranawaka.T.D



IT20145552  
Dissanayaka. D. M. S. M



IT20652500  
B.N.S. Gunadasa



IT20145553  
T.M.S.B. Thelyvill

# Introduction.

## Why HeatWave resilience is important?

- **Public health protection**  
Even WHO has declared set of instructions to follow during heatwave conditions.
- **Provide Agricultural Resilience**  
forecast soil moisture and recommend suitable crops which helps to farmers to mitigate the impact .

# Introduction.

## Why HeatGuard is important?

- **Early warning system**

Most people are not aware whether they are in a heatwave condition. This will help to reduce series health impacts

- **Public health protection**

Make people aware about heatwaves which brings significant risk to public health such as heat stroke, dehydration and heat exhaustion.

- **Provide Agricultural Resilience**

forecast soil moisture and recommend suitable crops which helps to farmers to mitigate the impact .

# Introduction.

Through Our App,

- We do risk communication to the users and asses their functionalities and engagement with exposure to heat.
- The background research begins with examines the various impacts on different sectors such as public health , agricultural , environmental etc...

# Research Problems.

- How the data driven decision making can apply to predict heatwaves.
- How to forecast soil moisture and predict crops for mitigate the impacts for yields.
- How to apply data driven decision making to mitigate health risks.

# Background Research

- **Heatwave Phenomenon:** Defines heatwaves, explores their effects on health, agriculture, infrastructure, and the environment.
- **Existing Research and Solutions:** Reviews previous studies on heatwave prediction, monitoring, and mitigation, identifying gaps and issues.
- **Mobile Applications for Heatwave Resilience:** Investigates the role of mobile apps in heatwave resilience, assessing features and user involvement strategies.

# Data Collection

Google Earth Engine

Search places and datasets...

Scripts Docs Assets

Filter scripts... NEW

Owner (1)  
Writer  
Reader  
Archive  
Examples

UntitledFile

Imports (2 entries)

```
> var imageVisParam: ssm from 0 to 28
> var imageVisParam2: ssm from 0 to 28

1 // Define countries boundary
2 var Countries = ee.FeatureCollection('USDOIS/LSIB_SIMPLE/2017');
3 var roi = Countries.filter(ee.Filter.eq('country_na', 'Sri Lanka'));
4 Map.addLayer(roi, {}, "Sri Lanka");
5 Map.centerObject(roi);
6
7 // List of years and months
8 var years = ee.List.sequence(2017, 2022); // Adjust as needed
9 var months = ee.List.sequence(1, 12);
10
11 // Load SMAP Data
12 var coll = ee.ImageCollection('NASA_USDA/HSL/SMAP10KM_soil_moisture');
13
14 // Filter data for the years 2017 to 2022
15 coll = coll.filterDate('2017-01-01', '2022-12-31');
16
17 // Check if the image collection contains valid images
18 if (coll.size().getInfo() === 0) {
19   print('No valid images found in the specified time period.');
20 } else {
```

Inspector Console Tasks

Search or cancel multiple tasks in the Task Manager

UNSUBMITTED TASKS

Soil\_Moisture\_Sri\_Lanka\_2017\_to\_2022 RUN

SUBMITTED TASKS

Soil\_Moisture\_Sri\_Lanka\_2017\_to\_2022 ✓ 2m  
Soil\_Moisture\_Sri\_Lanka\_2017 ✓ <1m  
Soil\_Moisture\_Sri\_Lanka\_2017 ✓ <1m  
Soil\_Moisture\_Sri\_Lanka\_2017 ✓ <1m  
Soil\_Moisture\_Sri\_Lanka\_2017 ✓ <1m  
Soil\_Moisture\_Sri\_Lanka3 ▲ <1m  
Soil\_Moisture\_Sri\_Lanka3 ▲ <1m  
Soil\_Moisture\_Sri\_Lanka3 ▲ <1m

Laccadive Sea

Layers Map Satellite

Keyboard shortcuts | Map data ©2024 | 50 km | Terms

# Data Collection

[kaggle](#)

[Create](#)

[Home](#)

[Competitions](#)

[Datasets](#)

[Models](#)

[Code](#)

[Discussions](#)

[Learn](#)

[More](#)

[View Active Events](#)

Search

ATHARVA INGLE - UPDATED 3 YEARS AGO

352

New Notebook

Download (65 kB)

Crop Recommendation Dataset

Maximize agricultural yield by recommending appropriate crops

Data Card    Code (135)    Discussion (9)    Suggestions (0)

About Dataset

**Context**

Precision agriculture is in trend nowadays. It helps the farmers to get informed decision about the farming strategy. Here, I present you a dataset which would allow the users to build a predictive model to recommend the most suitable crops to grow in a particular farm based on various parameters.

**Context**

This dataset was build by augmenting datasets of rainfall, climate and fertilizer data available for India.

**Data fields**

- N - ratio of Nitrogen content in soil

**Usability** 5.88

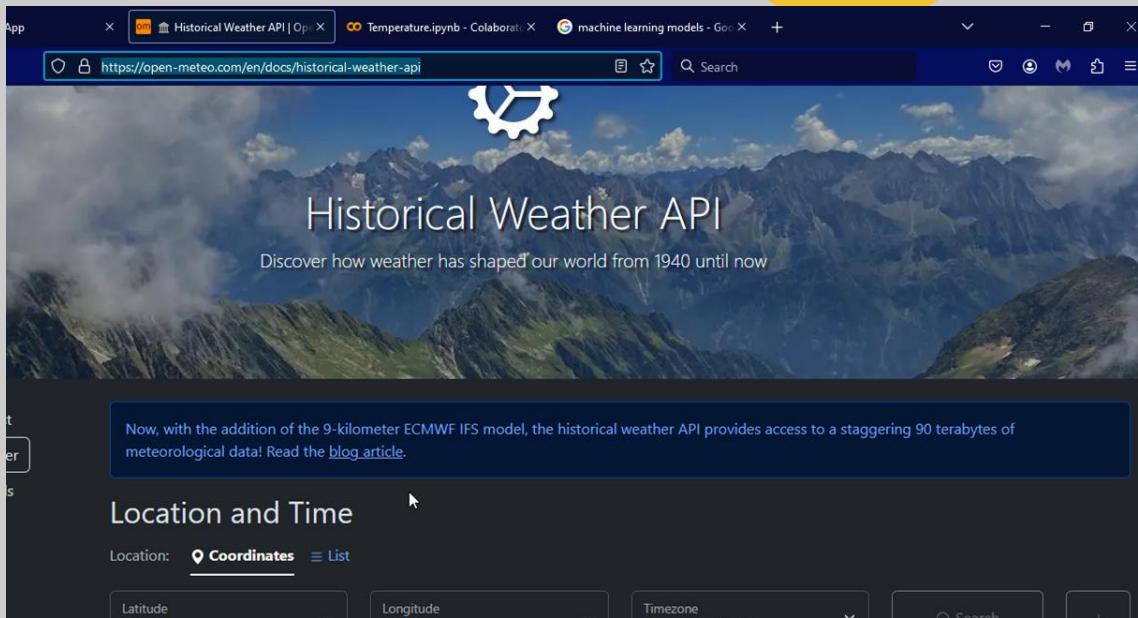
**License** Unknown

**Expected update frequency** Not specified

**Tags**

Tabular   Agriculture  
Recommender Systems

# Data Collection





**Ranawaka . T . D**  
**IT 20142728**

Specialization - Information Technology

# Research Question

- How to forecast soil moisture accurately ?
- How can machine learning models improve their accuracy and dependability of predicting crops according to soil moisture levels ?
- How to implement a system that cultivators can monitor soil moisture for their crops ?

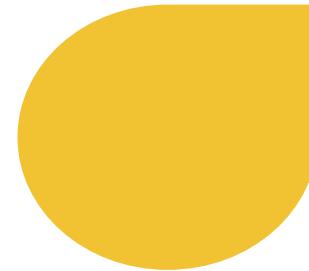


# Research Gap



Features	HeatAlert+	Crisis24 Horizon Mbile	HeatGuard
Recommend Crops	✗	✗	✓
Soil moisture Forecasting	✗	✗	✓
Support for Sri Lanka	✗	✓	✓

# Novelty



- Integrating Soil moisture forecasting with crop prediction.
- There isn't a mobile app that supports for Sri Lankan geography accurately.

# Main Objective



- Create a system for monitoring soil moisture accurately and real time to effectively manage water requirements.

## Sub Objectives

- Use machine learning models to predict crops according to soil moisture levels.

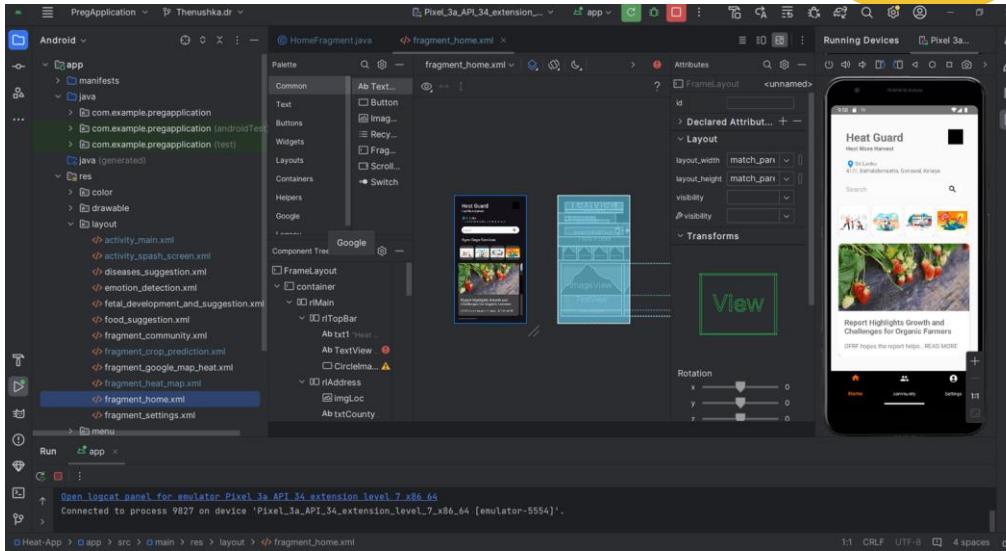
# Tools and Algorithms

# Technologies

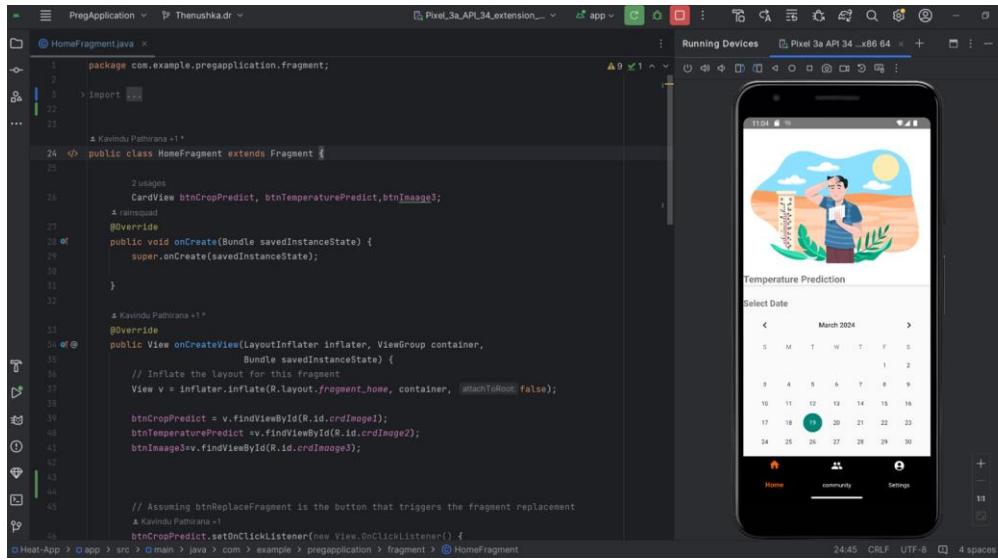
- **Algorithm :** RandomForestClassifier
- **Develop Mobile Application :**  
Android Studio
- **Model Trainer:** PyCharm
- **Languages:** Python, Java



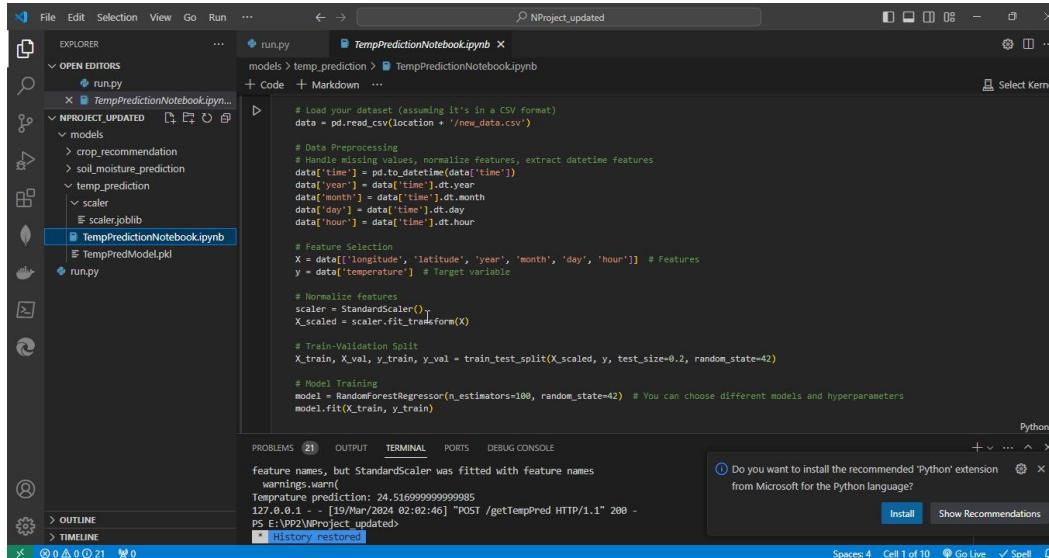
# Project Evidence



# Completion of the project



# Completion of the project



The screenshot shows a Jupyter Notebook interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Title Bar:** run.py TempPredictionNotebook.ipynb
- Explorer:** OPEN EDITORS, NPROJECT\_UPDATED, models, crop\_recommendation, soil\_moisture\_prediction, temp\_prediction, scaler, TempPredictionNotebook.ipynb, TempPredModel.pkl, run.py
- Code Cell:** Python code for temperature prediction, including data loading, preprocessing, feature selection, normalization, and model training.
- Output Cell:** Shows a warning about feature names and a log entry for a POST request to /getTempPred.
- Bottom Status Bar:** Spaces: 4, Cell 1 of 10, Go Live, Spell

# Current Progress

- Collecting the necessary dataset.
- Preparing the environment in the machine.
- Completed machine learning models.
- Completed the main UI.

# What's to be done

- Integrate components.
- Connecting IoT device for data input.



**Dissanayaka. D. M. S. M.**  
**IT 20145552**

Specialization - Information Technology

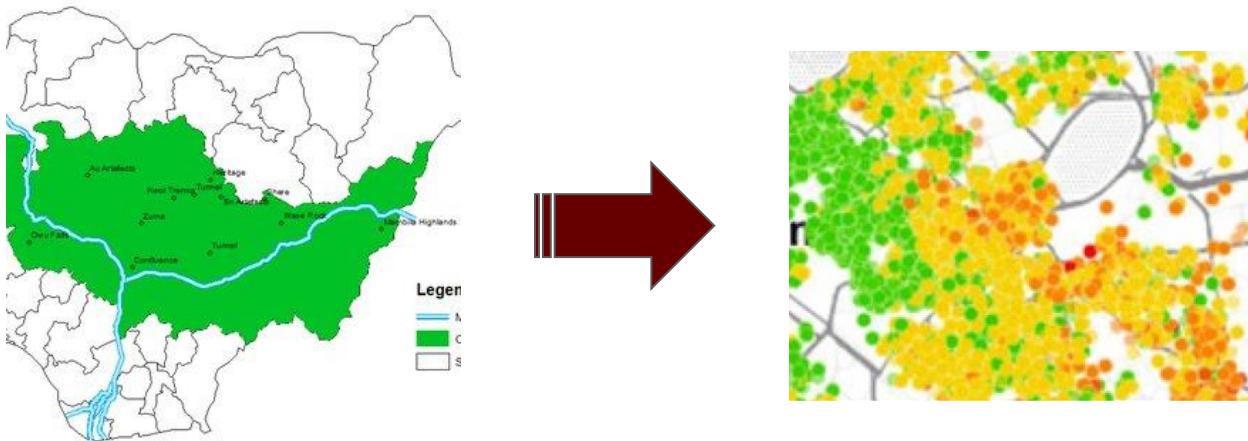
# Research Question

- What are the key elements contributing to vulnerability to heat-related risks in different **geographical areas?**
- How can machine learning algorithms be effectively utilized to process large datasets related to **heatwaves?**
- What are the most suitable data sources for developing a **heatwaves mapping** system?

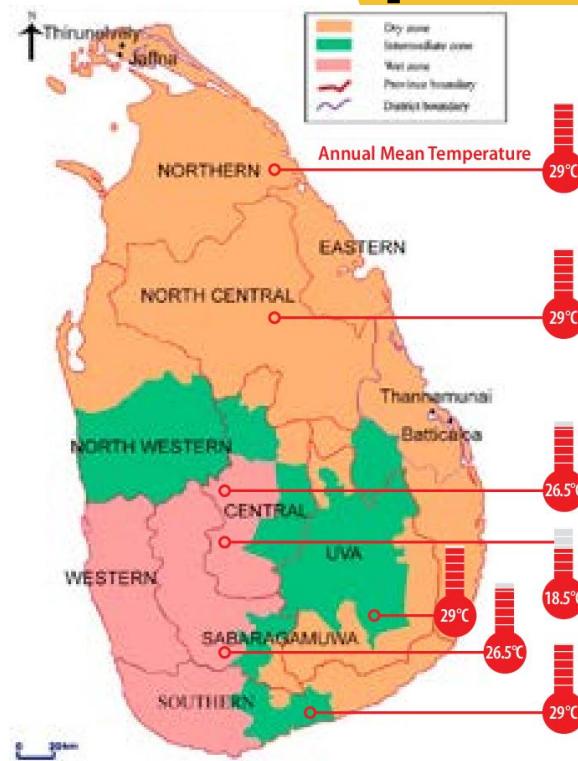


# Research Gap

Inability to obtain information on heat wave resolution and the inability to obtain information simultaneously as a **single point of location** rather than Near the Location



# Research Gap



# Research Gap

Features	HeatAlert+	Crisis24 Horizon
Prediction Crops (Recommend)	✓	✗
Suggest Irrigation levels	✓	✗
Heat Wave Prediction	✓	✗
Support for Sri Lanka	✓	✗

# Novelty

- Current solutions struggle to correctly map Sri Lanka's varied geography, including its coastlines, mountains, and cities.
- Current evaluations often overlook income and healthcare access, making vulnerability assessments too general and not community specific
- The system works on mobile phones, allowing people share data and get alerts.

# Main Objective

- Develop machine learning algorithms capable of accurately analyzing heat waves datasets to identify localized vulnerability patterns to heat waves related risks in Sri Lanka



# Technologies Tools and Algorithms

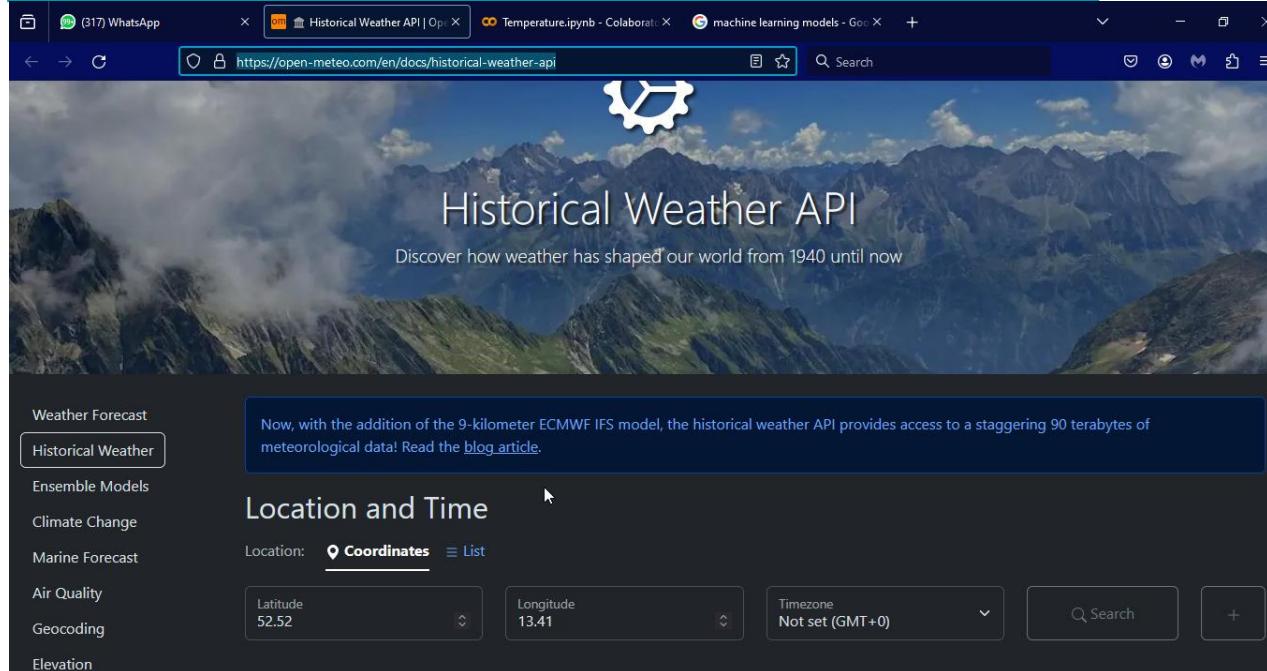
- **Algorithm :**RandomForestRegressor
- **Develop Mobile Application :**  
Android Studio
- **Model Trainer:** PyCharm
- **Languages:** Python, Java



# Project Evidence

- Dataset given from that link:

<https://open-meteo.com/en/docs/historical-weather-api>



The screenshot shows a web browser window with the URL <https://open-meteo.com/en/docs/historical-weather-api> in the address bar. The page features a large background image of a mountain range under a cloudy sky. At the top center is a white gear icon. Below it, the text "Historical Weather API" is displayed in a large, serif font. Underneath, a subtitle reads "Discover how weather has shaped our world from 1940 until now". On the left side, there is a sidebar with several menu items: "Weather Forecast", "Historical Weather" (which is currently selected and highlighted in a box), "Ensemble Models", "Climate Change", "Marine Forecast", "Air Quality", "Geocoding", and "Elevation". The main content area contains a dark blue callout box with the text: "Now, with the addition of the 9-kilometer ECMWF IFS model, the historical weather API provides access to a staggering 90 terabytes of meteorological data! Read the [blog article](#)". Below this, there is a "Location and Time" section with input fields for "Latitude" (52.52), "Longitude" (13.41), and "Timezone" (Not set (GMT+0)). There are also "Search" and "+" buttons.

# Project Evidence

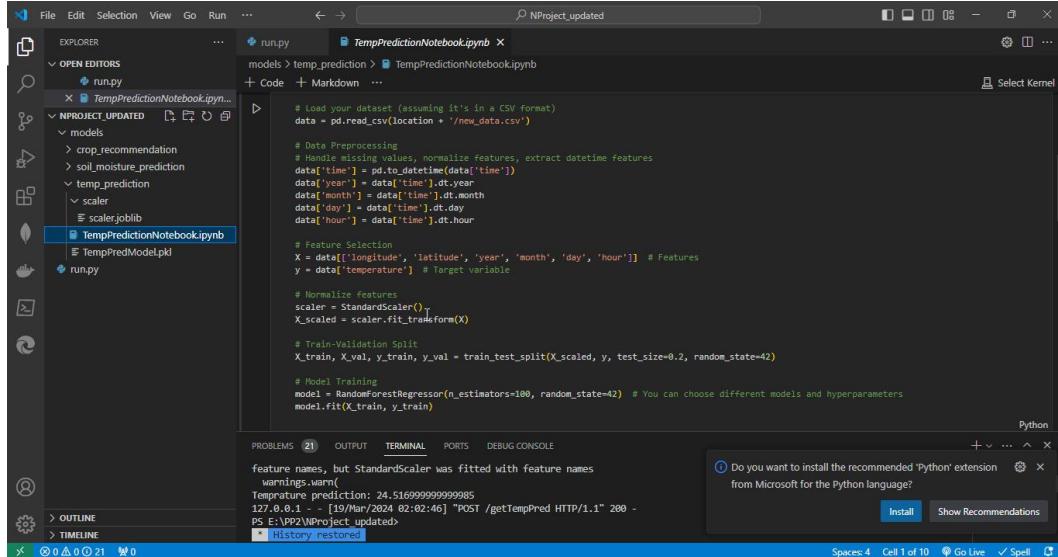
- Dataset:

The screenshot shows a Microsoft Excel spreadsheet titled "new\_data.csv". The data consists of 22 rows and 6 columns. The columns are labeled A through F. The first row contains column headers: longitude, latitude, time, temperature, relative\_humidity\_2m (%), and rain (mm). The subsequent rows provide data points for these variables over time. Row 9 is highlighted with a green border, and the cell containing "63" is also highlighted with a green border, indicating it is selected.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	longitude	latitude	time	temperature	relative_humidity_2m (%)	rain (mm)											
2	80.44992	7.4868193	2010-01-01T00:00	21.6	92	0											
3	80.44992	7.4868193	2010-01-01T01:00	21.7	91	0											
4	80.44992	7.4868193	2010-01-01T02:00	22.6	89	0											
5	80.44992	7.4868193	2010-01-01T03:00	24.4	82	0											
6	80.44992	7.4868193	2010-01-01T04:00	25.8	74	0											
7	80.44992	7.4868193	2010-01-01T05:00	26.7	68	0											
8	80.44992	7.4868193	2010-01-01T06:00	27.4	65	0											
9	80.44992	7.4868193	2010-01-01T07:00	27.9	63	0											
10	80.44992	7.4868193	2010-01-01T08:00	27.8	64	0											
11	80.44992	7.4868193	2010-01-01T09:00	28.1	63	0											
12	80.44992	7.4868193	2010-01-01T10:00	27.6	65	0											
13	80.44992	7.4868193	2010-01-01T11:00	27	68	0											
14	80.44992	7.4868193	2010-01-01T12:00	26.1	71	0											
15	80.44992	7.4868193	2010-01-01T13:00	25.4	73	0											
16	80.44992	7.4868193	2010-01-01T14:00	24.6	77	0											
17	80.44992	7.4868193	2010-01-01T15:00	24.2	80	0											
18	80.44992	7.4868193	2010-01-01T16:00	23.8	82	0											
19	80.44992	7.4868193	2010-01-01T17:00	23.6	83	0											
20	80.44992	7.4868193	2010-01-01T18:00	23.2	86	0											
21	80.44992	7.4868193	2010-01-01T19:00	23	85	0											
22	80.44992	7.4868193	2010-01-01T20:00	22.5	87	0											

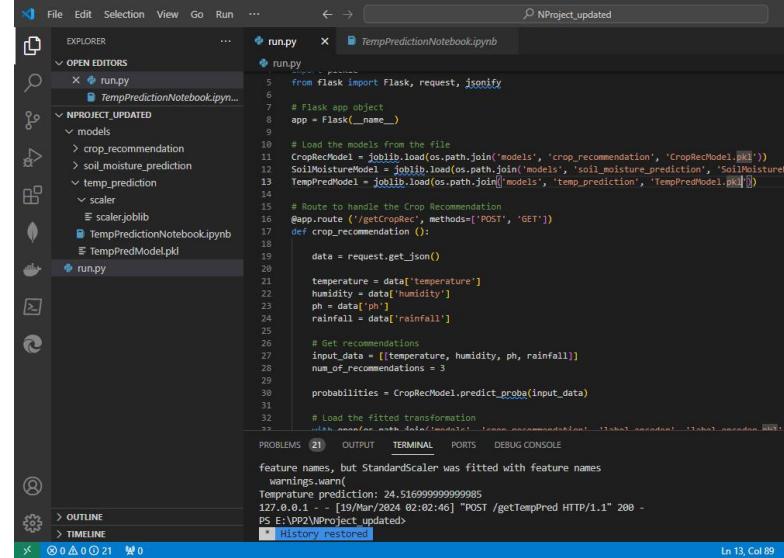
# Completion of the project

## CODE EVIDENCE



The screenshot shows a Jupyter Notebook interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Title Bar:** NProject\_updated
- Explorer:** OPEN EDITORS (run.py, TempPredictionNotebook.ipynb), NPPROJECT\_UPDATED (models, crop\_recommendation, soil\_moisture\_prediction, temp\_prediction, scaler, scaler.joblib, TempPredictionNotebook.ipynb, TempPredModel.pkl), run.py.
- Code Cell:** Contains Python code for data preprocessing, feature selection, model training, and validation.
- Output Cell:** Shows a warning message about StandardScaler fitting feature names.
- Terminal:** Displays command-line output from a POST request to /getTempPred.
- Bottom:** Includes tabs for PROBLEMS, OUTPUT, TERMINAL, PORTS, DEBUG CONSOLE, and various status indicators.

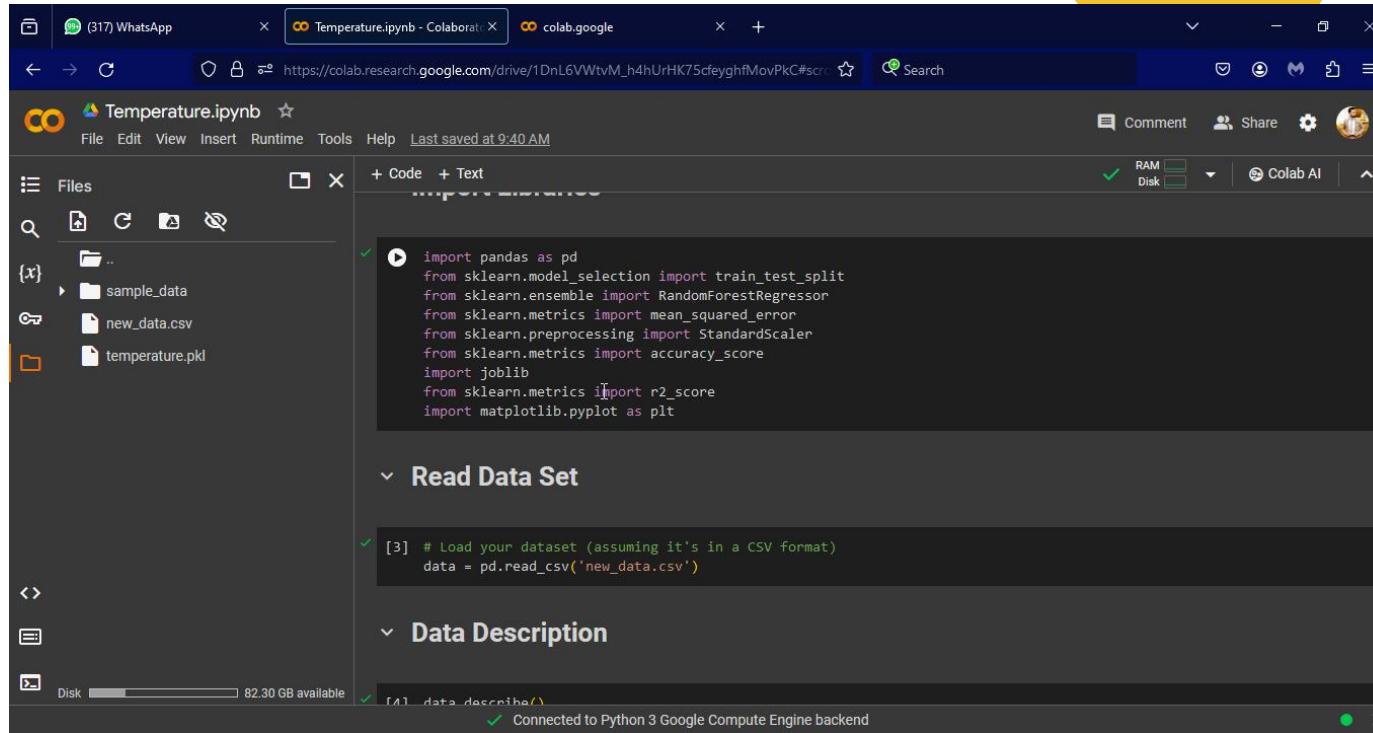


The screenshot shows a Jupyter Notebook interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Title Bar:** NProject\_updated
- Explorer:** OPEN EDITORS (run.py, TempPredictionNotebook.ipynb), NPPROJECT\_UPDATED (models, crop\_recommendation, soil\_moisture\_prediction, temp\_prediction, scaler, scaler.joblib, TempPredictionNotebook.ipynb, TempPredModel.pkl), run.py.
- Code Cell:** Contains Python code for loading models, handling API requests, and generating crop recommendations based on input data.
- Output Cell:** Shows a warning message about StandardScaler fitting feature names.
- Terminal:** Displays command-line output from a POST request to /getCropRec.
- Bottom:** Includes tabs for PROBLEMS, OUTPUT, TERMINAL, PORTS, DEBUG CONSOLE, and various status indicators.

# Completion of the project

## MODEL RUNNING



The screenshot shows a Google Colab notebook titled "Temperature.ipynb". The notebook interface includes a top bar with tabs for WhatsApp, Temperature.ipynb - Collaborator, and colab.google, along with standard browser controls. Below the top bar is a toolbar with File, Edit, View, Insert, Runtime, Tools, Help, and a status message "Last saved at 9:40 AM". On the left is a sidebar with "Files" containing "sample\_data", "new\_data.csv", and "temperature.pkl". The main workspace contains code cells and sections:

- A code cell with imports:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
import joblib
from sklearn.metrics import r2_score
import matplotlib.pyplot as plt
```
- A section titled "Read Data Set" with a code cell:

```
[3] # Load your dataset (assuming it's in a CSV format)
data = pd.read_csv('new_data.csv')
```
- A section titled "Data Description" with a code cell:

```
[4] data.describe()
```

At the bottom, there are disk usage indicators: "Disk 82.30 GB available" and "Connected to Python 3 Google Compute Engine backend".

# Completion of the project

## MODEL RUNNING

The screenshot shows a Google Colab interface with a Jupyter notebook titled "Temperature.ipynb". The code cell [4] contains the command `data.describe()`, which outputs a descriptive statistics table for the dataset. The table includes columns for longitude, latitude, temperature, relative\_humidity\_2m (%), and rain (mm), with rows for count, mean, std, min, 25%, 50%, 75%, and max.

	longitude	latitude	temperature	relative_humidity_2m (%)	rain (mm)
count	177398.000000	177398.000000	177398.000000	177398.000000	177398.000000
mean	81.072362	7.366794	26.55520	79.671935	0.198046
std	0.618114	0.169435	2.96076	13.822294	0.832390
min	80.449920	6.151143	16.60000	15.000000	0.000000
25%	80.449920	7.275923	24.30000	72.000000	0.000000
50%	81.127815	7.275923	26.10000	83.000000	0.000000
75%	81.693474	7.486819	28.30000	90.000000	0.000000
max	81.693474	7.486819	39.90000	100.000000	35.300000

Below the code cell, there is a message: "Start coding or generate with AI." At the bottom of the notebook, it says "Connected to Python 3 Google Compute Engine backend".

# Completion of the project

## MODEL RUNNING



A screenshot of a Google Colab notebook titled "Temperature.ipynb". The code cell contains the following Python code:

```
45 plt.scatter(data['longitude'], data['temperature'], label='longitude vs. temperature')
plt.xlabel('longitude')
plt.ylabel('temperature')
plt.legend()
plt.title('Scatter Plot')
plt.show()
```

The resulting scatter plot shows a positive correlation between longitude and temperature. The x-axis is labeled "longitude" and ranges from approximately 20 to 40. The y-axis is labeled "temperature" and ranges from 20 to 40. A single data point is visible at approximately (35, 33).

Files in the sidebar: sample\_data, new\_data.csv, temperature.pkl.



A screenshot of a Google Colab notebook titled "Temperature.ipynb". The code cell contains the following Python code:

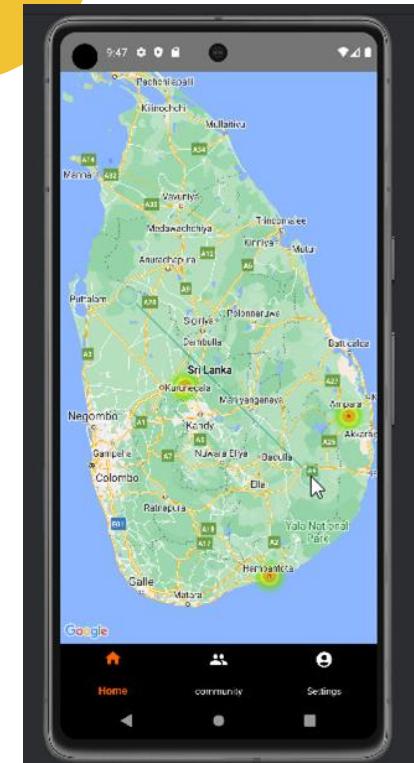
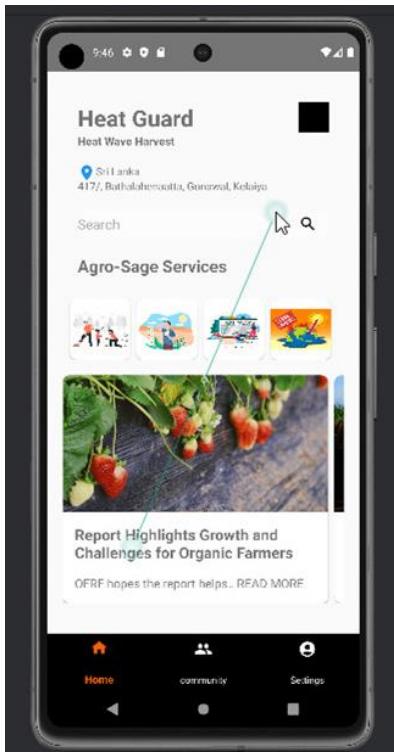
```
2s plt.scatter(data['latitude'], data['temperature'], label='latitude vs. temperature')
plt.xlabel('latitude')
plt.ylabel('temperature')
plt.legend()
plt.title('Scatter Plot')
plt.show()
```

The resulting scatter plot shows a positive correlation between latitude and temperature. The x-axis is labeled "latitude" and ranges from approximately 20 to 40. The y-axis is labeled "temperature" and ranges from 20 to 40. Two data points are visible at approximately (30, 32) and (35, 33).

Files in the sidebar: sample\_data, new\_data.csv, temperature.pkl.

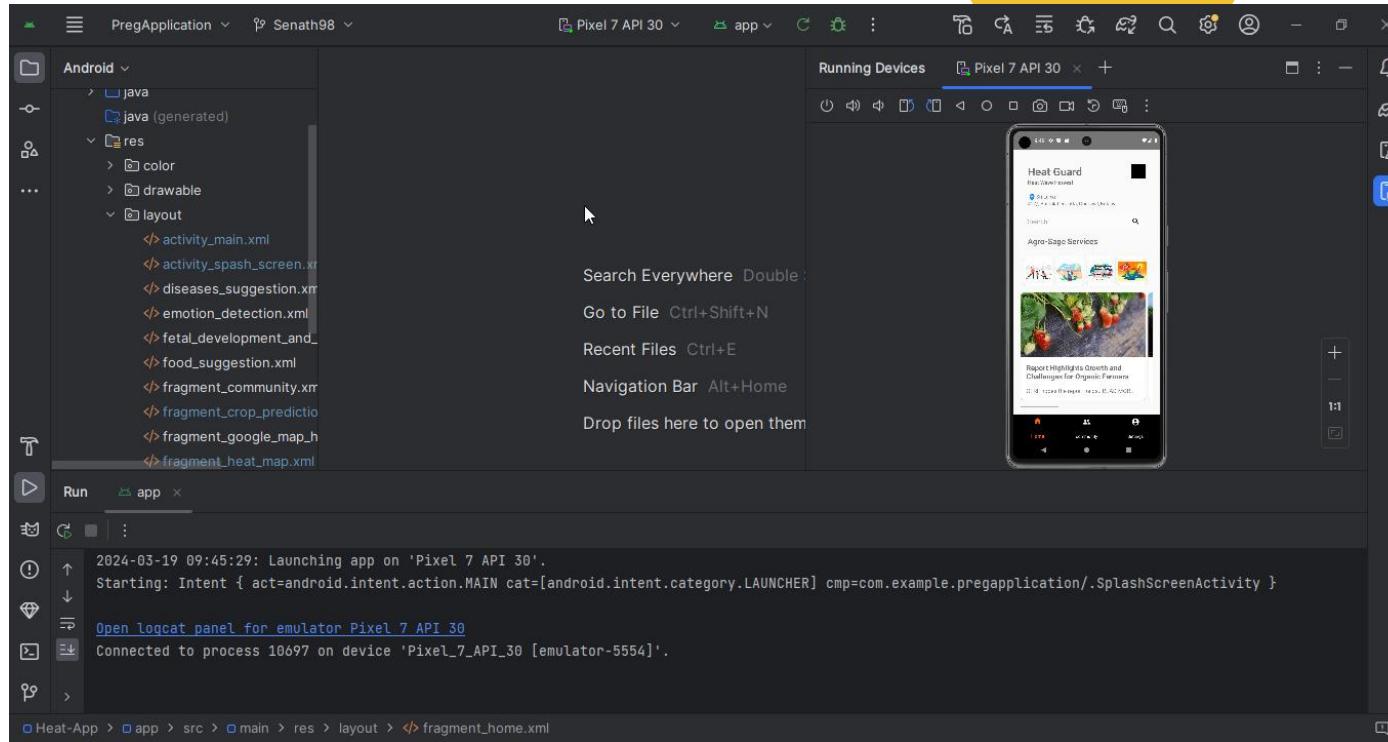
# Completion of the project

# UI EVIDENCE



# Completion of the project

## UI EVIDENCE



# Current Progress

- Identified key vulnerability elements relevant to the Sri Lankan context.
- Completed the initial design of the User Interface (UI)
- The notification system should be developed.

# What's to be done

- Finished develop the application.
- Ensure smooth communication between frontend and backend.
- Test the integrated system for functionality and performance.

# Risk Mitigation

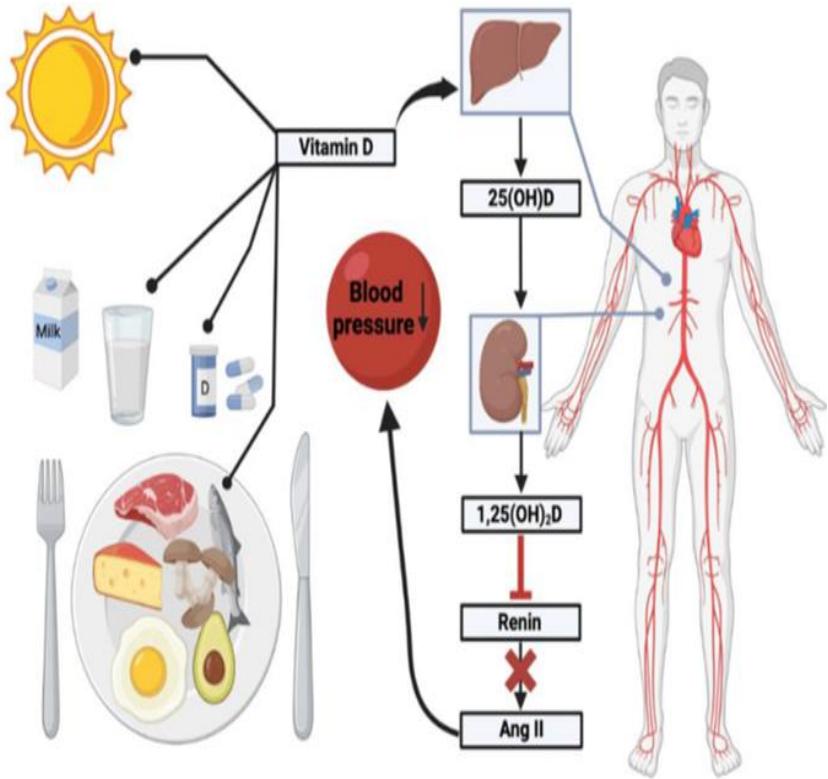
- Prioritize vital features to manage resource constraints effectively.
- Gather user feedback early on to streamline user interface design and reduce complexity.
- Implement robust security actions to protect user data from potential breaches.
- Maintain open communication with users
- Monitor and adjust project plans to mitigate taking off risks effectively.



**Gunadasa .B.N.S**  
**IT 20652500**

Specialization - Information Technology

# Research Introduction



This research paper aims to understand the effects of diuretics and beta-blockers medication on outdoor workers during heat waves. It seeks to develop a system that can improve efficiency and safety by identifying their health risks and recommending preventative measures. The study is crucial as heatwaves are becoming more frequent due to climate change, and outdoor workers are particularly vulnerable to heat exhaustion and dehydration. The paper suggests a system that can help workers manage their medication usage and work periods based on their impact.

# Main Objective

Introduce the research aim: to develop a recommendation system for diuretics and beta-blockers medication during heat waves.

Emphasize the system's goal: to offer personalized guidance on water intake and work timing.

## Specific Objectives

- ★ Collect future heatwave forecast data and user data, such as weight, medication use, and water levels.
- ★ Develop a method to estimate the water level in the human body.
- ★ Recommend how much water users should drink within a given time period, based on the impact of heatwaves, medication, and biodata.
- ★ Identify and analyze the most suitable times for users to work, based on heatwaves and the impact of diuretics and medication.

# About Medicine



Dr. Wasantha P.K. Yamasinghe  
MBBS (SRI LANKA)  
Reg No: 12122

MEDI CARE CENTER,  
KOTAWATUNA,

Beta blocker drugs and  
diuretics are used in cardio-  
vascular diseases and hypertension  
in my clinical practice.



# Technologies, Tools and Algorithms



- Algorithm: **NLP**
- Develop Mobile application: **Android Studio**
- k-means clustering
- LDA topic modeling
- Model Trainer : **PyCharm**
- Languages : **Python,Java**

# Current Progress

- Collect future heatwave forecast data and user data.
- Completed the main UI.
- Outdoor workers can enter their information and complete the forme.

# Project Evidence

## Dataset for progress evaluation

P1	B	C	D	E	F	G	H	I	J	K	L	M
1	Date of Birth	Gender	Current Weight	Health Condition	Allergies	General Health	Diuretics	Medications	Diuretics Name	Beta-Blockers I	Beta-Blockers II	Other Medications
2	27/8/1968	Female	61	Diabetes	Yes	Fair	No	Med_B 20mg	Yes	None	None	Yes
3	1/7/1971	Other	112	Hypertension;H	No	Poor	No	None	No	Med_Z 10mg	Med_P 5mg	Yes
4	24/11/1962	Prefer not to say	99	Diabetes;Kidney	No	Very Good	Yes	None	No	Med_Z 10mg	None	No
5	16/10/1976	Other	108		Yes	Good	No	Med_A 50mg	No	None	Med_Q 25mg	No
6	7/2/1971	Female	120		Yes	Poor	No	Med_C 10mg	Yes	Med_X 50mg	None	Yes
7	16/2/1980	Male	55	Hypertension	Yes	Good	Yes	None	Yes	Med_Z 10mg	None	Yes
8	18/1/1962	Male	78	None;Heart Fai	Yes	Excellent	No	Med_B 20mg	Yes	Med_Y 20mg	Med_P 5mg	Yes
9	13/3/1978	Prefer not to say	88	Kidney Disorde	Yes	Fair	Yes	Med_B 20mg	No	Med_Y 20mg	Med_P 5mg	No
10	8/6/1970	Female	99	None;Hyperten	No	Poor	No	Med_A 50mg	Yes	None	Med_Q 25mg	Yes
11	1/6/1993	Male	76	Heart Failure	No	Very Good	Yes	Med_B 20mg	Yes	Med_Y 20mg	Med_R 15mg	No
12	4/8/1975	Prefer not to say	72	None;Heart Fai	No	Fair	No	Med_B 20mg	No	None	Med_P 5mg	Yes
13	20/11/1970	Male	88	Diabetes	Yes	Good	Yes	Med_B 20mg	No	None	Med_P 5mg	No
14	8/8/1961	Female	118	Diabetes	No	Excellent	No	Med_A 50mg	Yes	Med_Y 20mg	Med_Q 25mg	Yes
15	9/7/1973	Prefer not to say	76	Hypertension;H	No	Excellent	No	Med_B 20mg	No	Med_Y 20mg	Med_P 5mg	No
16	17/7/1973	Prefer not to say	119	Kidney Disorde	No	Very Good	No	Med_B 20mg	No	Med_X 50mg	Med_P 5mg	Yes
17	9/1/1975	Other	47	Kidney Disorde	Yes	Excellent	Yes	None	No	Med_Z 10mg	None	Yes
18	17/4/1984	Male	60	Heart Failure;H	Yes	Good	Yes	Med_A 50mg	No	Med_Z 10mg	Med_Q 25mg	No
19	8/12/1993	Other	120	Heart Failure;H	No	Excellent	No	Med_C 10mg	No	Med_X 50mg	Med_P 5mg	No
20	21/7/1983	Male	99	Hypertension	Yes	Fair	No	Med_A 50mg	Yes	Med_Z 10mg	Med_Q 25mg	Yes
21	28/1/1973	Prefer not to say	92	Hypertension	Yes	Good	No	Med_C 10mg	No	Med_Y 20mg	None	No
22	12/4/1967	Other	69	None;Diabetes	Yes	Very Good	Yes	Med_B 20mg	Yes	Med_Y 20mg	None	Yes
23	8/6/1977	Prefer not to say	73	Heart Failure;D	No	Good	Yes	None	No	Med_Z 10mg	Med_P 5mg	No
24	2/8/1991	Prefer not to say	103	Kidney Disorde	No	Good	Yes	Med_B 20mg	Yes	Med_X 50mg	Med_R 15mg	Yes
25	24/6/1977	Female	119	Diabetes	Yes	Good	No	None	Yes	Med_X 50mg	Med_R 15mg	No
26	22/7/1988	Other	96	Kidney Disorde	Yes	Fair	No	Med_C 10mg	Yes	Med_X 50mg	Med_R 15mg	No
27	3/1/1983	Female	53		Yes	Excellent	No	None	Yes	Med_Z 10mg	None	Yes

# Completion of the project

The screenshot shows a Jupyter Notebook interface with two open cells. The top cell contains a Pandas DataFrame with various health-related columns. The bottom cell contains Python code for data preprocessing, including scaling features and applying K-Means clustering to predict body water levels.

```
Index(['Current Weight (kg)', 'Gender_Female', 'Gender_Male', 'Gender_Other', 'Gender_Prefer not to say', 'Diuretics Medication_No', 'Diuretics Medication_Yes', 'Beta-Blockers Medication_No', 'Beta-Blockers Medication_Yes', 'Medical Records Consent_No', 'Medical Records Consent_Yes', 'Typical Workday in Heatwave_Topic_0', 'Typical Workday in Heatwave_Topic_1', 'Typical Workday in Heatwave_Topic_2', 'Determine Water Intake_Topic_0', 'Determine Water Intake_Topic_1', 'Determine Water Intake_Topic_2', 'Medication Effect on Hydration_Topic_0', 'Medication Effect on Hydration_Topic_1', 'Medication Effect on Hydration_Topic_2', 'Medication Effect on Hydration_Topic_3', 'Feel When Dehydrated_Topic_0', 'Feel When Dehydrated_Topic_1', 'Feel When Dehydrated_Topic_2', 'Severe Dehydration Experience_Topic_0', 'Severe Dehydration Experience_Topic_1', 'Severe Dehydration Experience_Topic_2', 'Hydration Habits_Topic_0', 'Hydration Habits_Topic_1', 'Hydration Habits_Topic_2', 'Tools for Monitoring Hydration_Topic_0', 'Tools for Monitoring Hydration_Topic_1', 'Tools for Monitoring Hydration_Topic_2'], dtype='object')

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Apply K-Means clustering to the scaled features
kmeans_combined = KMeans(n_clusters=3, random_state=42)
kmeans_combined.fit(X_scaled) # Use X_scaled directly

# Assign the clusters as the predicted body water levels with combined features
df['Predicted Body Water Level Combined'] = kmeans_combined.labels_

# Check the distribution of predicted body water levels with combined features
```

The screenshot shows a Jupyter Notebook interface with two open cells. The top cell contains a Pandas DataFrame with weather data. The bottom cell contains Python code defining a recommendation function based on alert level and body water level, and then applying this function to the dataset.

```
# Define the recommendation function
def recommend_work_time(alert_level, body_water_level):
    if alert_level in ["Low", "Moderate"]:
        if body_water_level == 1: # Low body water level
            return "Avoid peak heat; Stay hydrated"
        else: # Medium or high body water level
            return "Suitable with caution; Stay hydrated"
    elif alert_level in ["High", "Extreme"]:
        return "Not suitable; Avoid strenuous activities"
    else:
        return "Check local advisories"

# Apply the recommendation function to the heatwave dataset
# For demonstration, using a placeholder for medium body water level (0) for all
forecast_df['General Recommendation'] = forecast_df['Alert Level'].apply(
    lambda x: recommend_work_time(x, 0) # Using 0 as a placeholder for a medium
)

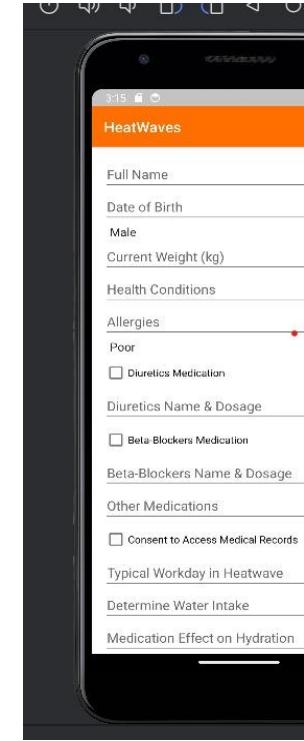
# Display the first few rows of the updated dataset
print(forecast_df.head())
```

	Date	Temperature (°C)	Humidity (%)	Heat Index (°C)	Alert Level
0	2024-01-01	37	95	39.594176	High
1	2024-01-02	28	58	29.534050	Low
2	2024-01-03	38	85	40.887715	High
3	2024-01-04	39	63	43.797167	High

# Completion of the project

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "Android". The "app" module contains "src" > "main" > "java" > "com" > "example" > "heatwaves" > "MainActivity.java".
- MainActivity.java Code:** The main editor window displays the Java code for MainActivity. The code includes imports, a class definition, an onCreate method, and a button click listener.
- Running Device:** The right side shows a virtual device for a "Pixel 3a API 34" running the app. The app's UI has an orange header with the title "HeatWaves" and a white content area with a "FORECAST" button.
- Status Bar:** At the bottom, a message indicates: "Install successfully finished in 842 ms."



# Risk Mitigation

- Prioritize vital features to manage resource constraints effectively.
- Gather user feedback early on to streamline user interface design and reduce complexity.
- Implement robust security actions to protect user data from potential breaches.
- Maintain open communication with users
- Monitor and adjust project plans to mitigate taking off risks effectively.

# Best Practices

- Prioritize vital features to manage resource constraints effectively.
- Gather user feedback early on to streamline user interface design and reduce complexity.
- Implement robust security actions to protect user data from potential breaches.
- Maintain open communication with users
- Monitor and adjust project plans to mitigate taking off risks effectively.

# Commercialization

- Prioritize vital features to manage resource constraints effectively.
- Gather user feedback early on to streamline user interface design and reduce complexity.
- Implement robust security actions to protect user data from potential breaches.
- Maintain open communication with users
- Monitor and adjust project plans to mitigate taking off risks effectively.

*Thank You*