



# **Analyzing Climate Trends and Predicting Extreme Weather Events in Sri Lanka**

# Introduction

**This presentation will cover the analysis of climate trends and the prediction of extreme weather events in Sri Lanka. The primary purpose of this research is to enhance the understanding of how climate change impacts this region and to develop accurate predictive models that can assist in disaster preparedness and mitigation strategies. The objectives include identifying the main drivers of climate change, analyzing how these changes manifest across different regions and seasons, and employing advanced methodologies to predict extreme weather events.**

# Background of the Study

**Sri Lanka has experienced significant climate variability in recent decades, characterized by irregular rainfall patterns, rising temperatures, and an increased frequency of extreme weather events such as floods, droughts, and cyclones. This section provides the necessary context by discussing the historical climate trends and the developments that have led to these changes. Factors such as natural climate variability and anthropogenic activities, including greenhouse gas emissions and deforestation, are highlighted to provide a comprehensive understanding of the current climate scenario in Sri Lanka.**

# Literature Review

Topic Title	Author(s)	Problem	Source
Application of Neural Networks in Weather Forecasting	Gyanesh Shrivastava, Sanjeev Karmakar and Manoj Kumar Kowar	Using Bayesian model averaging for forecast ensemble calibration	<a href="#"><u>International Journal of Computer Applications, August 2012</u></a>
Applying Neural Network Models to Prediction and Data Analysis in Meteorology and Oceanography	Hsieh & Tang	Prediction and data analysis using neural network models	<a href="#"><u>Applying Neural Network Models to Prediction and Data Analysis in Meteorology and Oceanography, 01 Sep 1998</u></a>
Using Bayesian Model Averaging to Calibrate Forecast Ensembles	Raftery et al.	Using Bayesian model averaging for forecast ensemble calibration	<a href="#"><u>Using Bayesian Model Averaging to Calibrate Forecast Ensembles, vNovember 4, 2003</u></a>
Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting	Xingjian <i>SHI</i> , <i>Zhourong Chen</i> , <i>Hao Wang</i>	Using convolutional LSTM networks for precipitation nowcasting, Precipitation nowcasting using machine learning	<a href="#"><u>Advances in Neural Information Processing Systems 28 (NIPS 2015)</u></a>

# Cont.....

Using Bayesian Model Averaging to Calibrate Forecast Ensembles	Adrian E. Raftery, Tilmann Gneiting and Fadoua Balabdaoui	forecast ensemble calibration.	<a href="#"><u>Ethical considerations in machine learning, 01 May 2005</u></a>
Classification, Seasonality and Persistence of Low-Frequency Atmospheric Circulation Patterns	Anthony G. Barnston and Robert E. Livezey	Classification and analysis of low-frequency atmospheric patterns	<a href="#"><u>Classification, Seasonality and Persistence of Low-Frequency Atmospheric Circulation Patterns, 01 Jun 1987</u></a>
The Quiet Revolution of Numerical Weather Prediction	Peter Bauer, Alan Thorpe and Gilbert Brunet	Advances in numerical weather prediction	<a href="#"><u>quiet revolution of numerical weather prediction, 02 September 2015</u></a>
Model Cards for Model Reporting	Simone Wu and Margaret Mitchell	Ethical considerations in machine learning	<a href="#"><u>Model Cards for Model Reporting, January 2019</u></a>



# **Research Objectives/Questions**

- 1. How can the implementation of machine learning algorithms be optimized for scalability and efficiency in the context of operational weather forecasting systems?**
- 2. What strategies and techniques can be employed to enhance the computational performance and scalability of machine learning models, particularly when deployed in resource-constrained environments?**
- 3. To what extent could machine learning models improve temperature and rainfall predictions?**

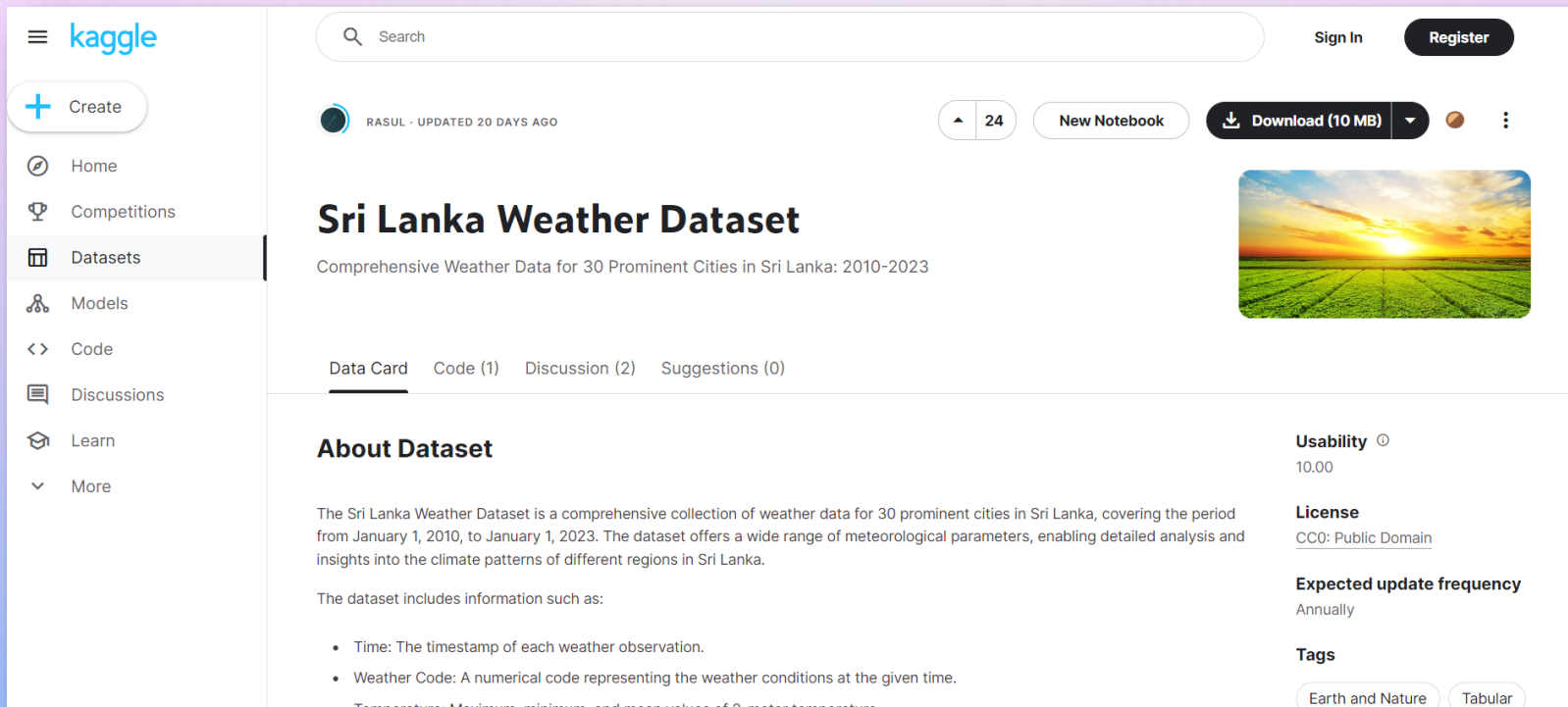
# Methodology

- **Research Design**

The study systematically developed and evaluated machine learning models for weather prediction using Python and sci-kit-learn. Key steps included preprocessing raw weather data to handle missing values and normalize features, selecting linear regression models for their effectiveness, training models on split data sets, and evaluating them using mean squared error (MSE). Models were validated with real-world data to ensure robustness and generalizability.

# • Data Collection Methods

Historical weather data was sourced from Kaggle. Python libraries like pandas and NumPy were used for data extraction and preprocessing.



The screenshot shows the Kaggle interface for the 'Sri Lanka Weather Dataset'. The left sidebar contains navigation links: Home, Competitions, Datasets (selected), Models, Code, Discussions, Learn, and More. The main content area displays the dataset title 'Sri Lanka Weather Dataset' with a subtitle 'Comprehensive Weather Data for 30 Prominent Cities in Sri Lanka: 2010-2023'. A user profile 'RASUL · UPDATED 20 DAYS AGO' is shown. Action buttons include 'New Notebook', 'Download (10 MB)', and a '24' badge. A landscape image of a sunset over a field is featured. Below the title, tabs for 'Data Card', 'Code (1)', 'Discussion (2)', and 'Suggestions (0)' are visible. The 'About Dataset' section describes the data as a comprehensive collection for 30 cities from 2010 to 2023. It lists included information: Time, Weather Code, and Temperature (Maximum, minimum, and mean values of 2-meter temperature). On the right, metadata includes 'Usability 10.00', 'License CC0: Public Domain', 'Expected update frequency Annually', and 'Tags Earth and Nature, Tabular'.

**Sri Lanka Weather Dataset**  
Comprehensive Weather Data for 30 Prominent Cities in Sri Lanka: 2010-2023

**About Dataset**

The Sri Lanka Weather Dataset is a comprehensive collection of weather data for 30 prominent cities in Sri Lanka, covering the period from January 1, 2010, to January 1, 2023. The dataset offers a wide range of meteorological parameters, enabling detailed analysis and insights into the climate patterns of different regions in Sri Lanka.

The dataset includes information such as:

- Time: The timestamp of each weather observation.
- Weather Code: A numerical code representing the weather conditions at the given time.
- Temperature: Maximum, minimum, and mean values of 2-meter temperature.

**Usability** 10.00

**License**  
CC0: Public Domain

**Expected update frequency**  
Annually

**Tags**  
Earth and Nature Tabular



# • Data Preprocessing

- **Selection of Relevant Features:** Only the columns relevant to the prediction tasks were retained. These included 'apparent\_temperature\_mean', 'rain\_sum', 'temperature\_2m\_mean', and 'weathercode'.

## ✓ 2. Data Preprocessing

Keep only the selected columns

```
[ ] data = data[['apparent_temperature_mean', 'rain_sum', 'temperature_2m_mean', 'weathercode']]
```

- **Handling Missing Values:** Any missing values within these selected columns were addressed by removing the incomplete records to maintain data integrity and prevent biases in the model training process.

Handle missing values if any

```
[ ] data = data.dropna()
```

# • Data Preprocessing

- **Correlation Analysis:** A correlation matrix was created to understand the relationships between the variables. This analysis helped in identifying multicollinearity and ensuring the selected features contributed uniquely to the model's predictions

```
# 4. Display Heatmap of Correlation Matrix
plt.figure(figsize=(10, 6))
correlation_matrix = data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix Heatmap')
plt.show()

# 5. Display Correlation Coefficients
print("Correlation Matrix:")
print(correlation_matrix)
```

```
Correlation Matrix:
              apparent_temperature_mean  rain_sum  \
apparent_temperature_mean              1.000000 -0.091989
rain_sum                          -0.091989  1.000000
temperature_2m_mean                  0.888324 -0.234696
weathercode                         0.099155  0.395169

              temperature_2m_mean  weathercode
apparent_temperature_mean      0.888324    0.099155
rain_sum                      -0.234696    0.395169
temperature_2m_mean            1.000000   -0.150776
weathercode                   -0.150776    1.000000
```

# • Model Selection and Training

- **Splitting the Data:** to effectively evaluate the performance of the weather prediction models, the dataset was divided into training and testing sets.

Separate features and target variables

```
[ ] X = data[['apparent_temperature_mean', 'rain_sum']] # Features
    y_temp = data['temperature_2m_mean'] # Target variable for temperature prediction
    y_weathercode = data['weathercode'] # Target variable for weather code prediction
    y_rain = data['rain_sum'] # Target variable for rain prediction
```

Split data into training and testing sets

```
[ ] X_train, X_test, y_temp_train, y_temp_test = train_test_split(X, y_temp, test_size=0.2, random_state=42)
    _, _, y_weathercode_train, y_weathercode_test = train_test_split(X, y_weathercode, test_size=0.2, random_state=42)
    _, _, y_rain_train, y_rain_test = train_test_split(X, y_rain, test_size=0.2, random_state=42)
```

# • Model Selection and Training

- **Training the Models:** the next step involved training separate Linear Regression models to predict temperature, weather code, and rain sum. Linear Regression was chosen for its simplicity and efficiency in modeling linear relationships.

## ✓ 3. Choose Models

```
[ ] temp_model = LinearRegression()  
    weathercode_model = LinearRegression()  
    rain_model = LinearRegression()
```

## ✓ 4. Train the Models

```
[ ] temp_model.fit(X_train, y_temp_train)  
    weathercode_model.fit(X_train, y_weathercode_train)  
    rain_model.fit(X_train, y_rain_train)
```



▼ LinearRegression

LinearRegression()



# • Model Selection and Training

- **Model Evaluation:** Model evaluation is a critical step in the machine learning workflow to ensure that the developed models perform well not only on the training data but also on unseen test data. This involves assessing the models' predictive accuracy and generalizability using various evaluation metrics

## ✓ 5. Evaluate the Models

```
[ ]
y_temp_pred = temp_model.predict(X_test)
temp_mse = mean_squared_error(y_temp_test, y_temp_pred)
print("Temperature Mean Squared Error:", temp_mse)

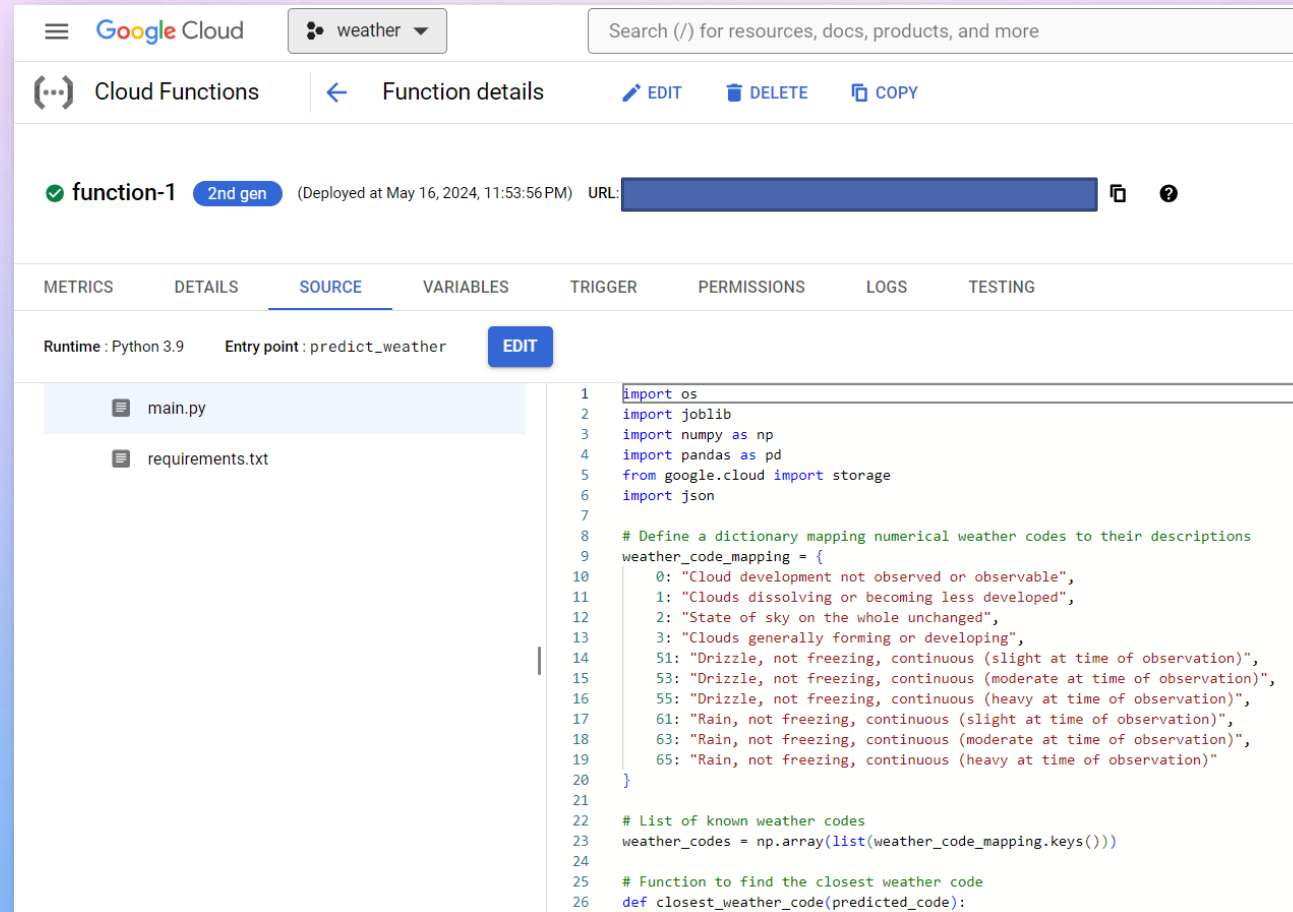
y_weathercode_pred = weathercode_model.predict(X_test)
weathercode_mse = mean_squared_error(y_weathercode_test, y_weathercode_pred)
print("Weather Code Mean Squared Error:", weathercode_mse)

y_rain_pred = rain_model.predict(X_test)
rain_mse = mean_squared_error(y_rain_test, y_rain_pred)
print("Rain Mean Squared Error:", rain_mse)
```

```
⇒ Temperature Mean Squared Error: 0.6294819522850056
Weather Code Mean Squared Error: 379.97785175752966
Rain Mean Squared Error: 4.3244541176454146e-29
```

# • Cloud Function Development

- **Writing the Cloud Function Code:** the cloud function is responsible for loading the trained machine learning models from Google Cloud Storage, processing incoming data, making predictions, and returning the results.



The screenshot displays the Google Cloud Functions console interface. At the top, the Google Cloud logo and a search bar are visible. Below the navigation bar, the 'Function details' page for 'function-1' is shown, indicating it is the '2nd gen' and was deployed on May 16, 2024, at 11:53:56 PM. The 'SOURCE' tab is selected, showing the function's runtime as Python 3.9 and its entry point as 'predict\_weather'. The source code editor displays the following Python code:

```
1 import os
2 import joblib
3 import numpy as np
4 import pandas as pd
5 from google.cloud import storage
6 import json
7
8 # Define a dictionary mapping numerical weather codes to their descriptions
9 weather_code_mapping = {
10     0: "Cloud development not observed or observable",
11     1: "Clouds dissolving or becoming less developed",
12     2: "State of sky on the whole unchanged",
13     3: "Clouds generally forming or developing",
14     51: "Drizzle, not freezing, continuous (slight at time of observation)",
15     53: "Drizzle, not freezing, continuous (moderate at time of observation)",
16     55: "Drizzle, not freezing, continuous (heavy at time of observation)",
17     61: "Rain, not freezing, continuous (slight at time of observation)",
18     63: "Rain, not freezing, continuous (moderate at time of observation)",
19     65: "Rain, not freezing, continuous (heavy at time of observation)"
20 }
21
22 # List of known weather codes
23 weather_codes = np.array(list(weather_code_mapping.keys()))
24
25 # Function to find the closest weather code
26 def closest_weather_code(predicted_code):
```

# • Google Blob Storage Deployment

- Google Cloud Storage (GCS) plays a crucial role in this project for storing and managing various artifacts and resources associated with the machine learning models and their deployment.

The screenshot displays the Google Cloud Storage console interface. At the top, the Google Cloud logo and a search bar are visible. The left sidebar contains navigation links for Cloud Storage, Buckets, Monitoring, and Settings. The main content area shows the 'Bucket details' for a bucket named 'storage-'. The bucket's location is 'us (multiple regions in United States)', storage class is 'Standard', public access is 'Not public', and protection is 'Soft delete'. Below the bucket details, there are tabs for OBJECTS, CONFIGURATION, PERMISSION, PROTECTION, LIFECYCLE, OBSERVABILITY, and INVENTORY REPORTS. The 'OBJECTS' tab is active, showing a 'Folder browser' view. The folder browser shows a hierarchy with 'storage-' as the parent folder and a subfolder. To the right of the folder browser, there are buttons for 'UPLOAD FILES', 'UPLOAD FOLDER', 'CREATE FOLDER', 'TRANSFER DATA', 'DOWNLOAD', and 'DELETE'. Below these buttons, there is a filter section with a dropdown for 'Filter by name prefix only' and a 'Filter' button. The object listing table shows three files: 'rain\_model.pkl', 'temperature\_model.pkl', and 'weathercode\_model.pkl', all with a size of 920 B and type 'application/octet-stream'.

Name	Size	Type
<a href="#">rain_model.pkl</a>	920 B	application/octet-stream
<a href="#">temperature_model.pkl</a>	920 B	application/octet-stream
<a href="#">weathercode_model.pkl</a>	920 B	application/octet-stream

# • Testing and Validation

- **Testing the weather prediction model:** The prepared sample inputs are fed into the trained machine-learning model to generate predictions. The model's output is compared against expected outcomes based on ground truth data or domain knowledge. Testing helps identify any discrepancies or inaccuracies in the model's predictions.

```
# Function to predict with the loaded models
def predict_weather(apparent_temperature_mean, rain_sum):
    # Create a DataFrame for the input features
    input_data = pd.DataFrame({
        'apparent_temperature_mean': [apparent_temperature_mean],
        'rain_sum': [rain_sum]
    })

    # Make predictions using the loaded models
    predicted_temperature = loaded_temp_model.predict(input_data)[0]
    predicted_weathercode = loaded_weathercode_model.predict(input_data)[0]
    predicted_rain = loaded_rain_model.predict(input_data)[0]

    # Find the closest weather code
    closest_code = closest_weather_code(predicted_weathercode)
    predicted_weather_description = weather_code_mapping.get(closest_code, "Unknown weather code")

    # Print the predictions
    print("Predicted Temperature:", predicted_temperature)
    print("Predicted Numerical Weather Code:", predicted_weathercode)
    print("Assigned Weather Code Description:", predicted_weather_description)
    print("Predicted Rain:", predicted_rain)

    return predicted_temperature, closest_code, predicted_rain, predicted_weather_description

# Example usage of the prediction function
apparent_temperature_mean = 38
rain_sum = 5

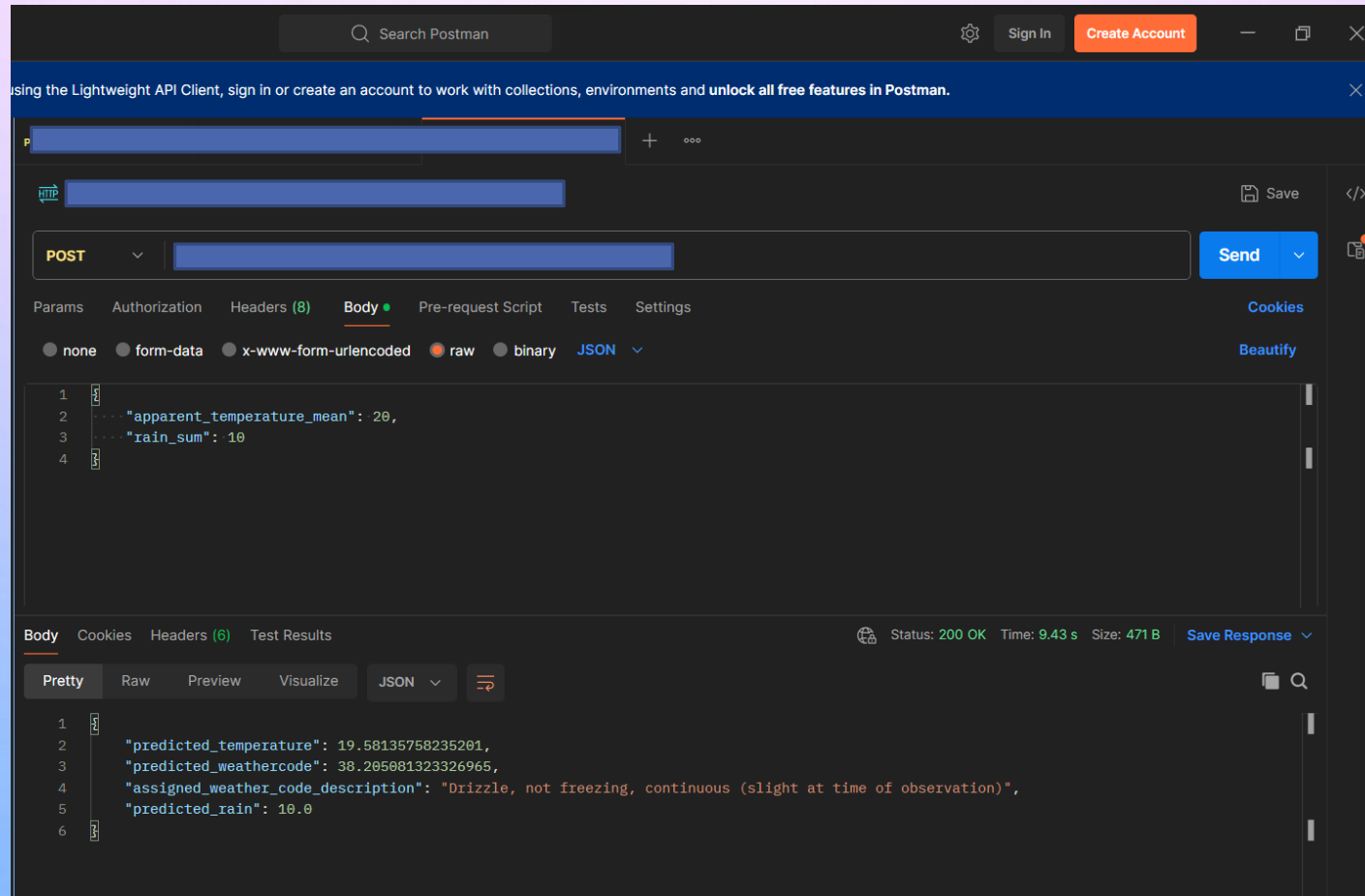
predicted_temperature, closest_code, predicted_rain, weather_description = predict_weather(apparent_temperature_mean, rain_sum)
```

Predicted Temperature: 31.119851939504276  
Predicted Numerical Weather Code: 54.60256579947055  
Assigned Weather Code Description: Drizzle, not freezing, continuous (heavy at time of observation)  
Predicted Rain: 5.000000000000002



# • Testing and Validation

- **Testing the Cloud Function:** cloud function is crucial to ensure that it operates correctly and returns accurate predictions. The primary tool used for this purpose is Postman, a popular API client that allows for the testing of HTTP requests



# Results

**The implementation of machine learning models for weather prediction was extended to a practical deployment, utilizing both a web interface and cloud-based AI model hosting. The web interface was developed using PHP and HTML and deployed on a free domain and hosting service provided by InfinityFree(<https://www.infinityfree.com>). Additionally, the machine learning model was deployed using Google Cloud Platform (GCP), specifically through Google Cloud Functions, to ensure scalable and efficient model inference.**

# Weather Prediction

Today's Temperature in Celsius(°C)

Today's Rain in millimeters (mm)

Predict

## Weather Prediction

Today's Temperature in Celsius(°C)

26

Today's Rain in millimeters (mm)

50

Predict

### Prediction Results

**Predicted Temperature:** 23°C

**Predicted Weather Code:** 80

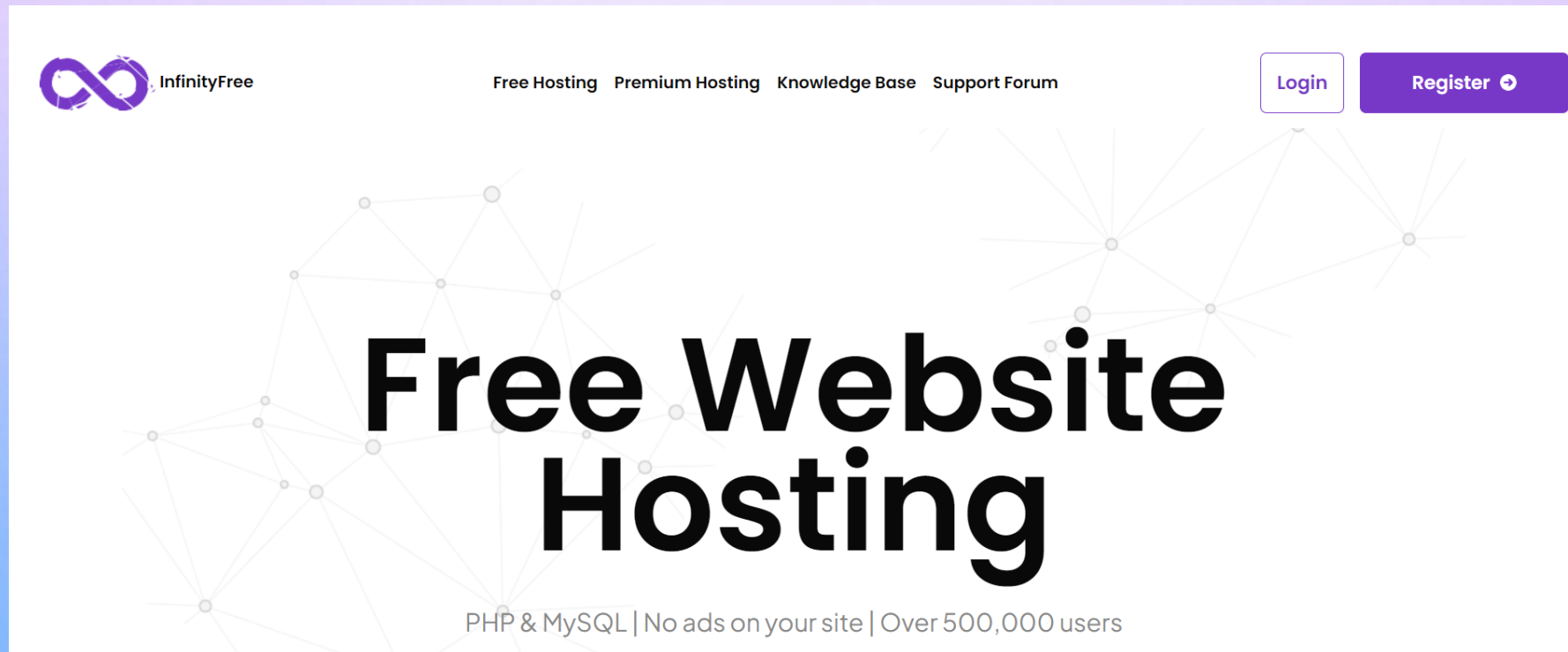
**Weather Description:** Rain, not freezing, continuous (heavy at time of observation)

**Predicted Rain:** 51mm

- **Hosting and domain name**

InfinityFree was chosen as the hosting provider due to its free services and robust infrastructure, offering unlimited disk space and bandwidth for the project, free subdomains, custom domain support, MySQL database support, and PHP/HTML compatibility.

<http://weather-check.rf.gd/>





- # Research Paper publication

By publishing on ResearchGate, the research was positioned to contribute to the broader scientific discourse on the application of machine learning in meteorology, offering valuable insights and methodologies that can be utilized and built upon by other researchers and practitioners in the field.

[https://www.researchgate.net/publication/380695196\\_Analyzing\\_Climate\\_Trends\\_and\\_Predicting\\_Extreme\\_Weather\\_Events\\_in\\_Sri\\_Lanka](https://www.researchgate.net/publication/380695196_Analyzing_Climate_Trends_and_Predicting_Extreme_Weather_Events_in_Sri_Lanka)

The screenshot shows the ResearchGate website interface. At the top, there's a navigation bar with 'ResearchGate' logo, 'Home', 'Questions', 'Jobs', a search bar, and user profile icons. The main content area displays the title 'Analyzing Climate Trends and Predicting Extreme Weather Events in Sri Lanka' with a 'May 2024' date. To the right of the title, there are statistics: 'Research Interest Score' (indicated by a bar), 'Citations' (0), 'Recommendations' (0), and 'Reads' (5). Below the title, there's a blue progress bar. At the bottom, there's a 'Research Spotlight Beta' section with the text 'Want to get 4x more reads of your article?'.

researchgate.net/publication/380695196\_Analyzing\_Climate\_Trends\_and\_Predicting\_Extreme\_Weather\_Events\_in\_Sri\_Lanka

ResearchGate Home Questions Jobs Search for research, journals, people, etc. Add new

Article Private full-text

**Analyzing Climate Trends and Predicting Extreme Weather Events in Sri Lanka**

May 2024

Research Interest Score --

Citations 0

Recommendations 0

Reads ⓘ 5

[Learn about stats on ResearchGate](#)

Overview Stats Comments Citations References (3) Share More

Research Spotlight Beta

Want to get 4x more reads of your article?

# Conclusion

**This study demonstrates the significant advancements in weather prediction through machine learning, utilizing Python and sci-kit-learn to enhance forecast accuracy and efficiency. Neural networks, feature engineering, and ensemble learning have shown great promise in capturing complex meteorological patterns. Despite these advancements, challenges such as model generalizability, interpretability, and ethical considerations remain. Future research should focus on improving the robustness and transparency of these models, integrating traditional numerical methods with machine learning to achieve more reliable and equitable weather forecasting systems.**



**THANK YOU**