
Climate Change Data Dashboard Documentation

This project is a Streamlit-based dashboard that visualizes climate change data and trains a machine learning model to predict sea surface temperature based on various environmental factors. The dashboard allows users to explore trends in climate data, filter by year range, and generate predictions using a linear regression model.

Table of Contents

1. Setup Instructions
 2. Code Overview
 3. Data Preprocessing
 4. Streamlit App Features
 5. Model Training and Predictions
 6. Dependencies
 7. Running the Application
 8. Troubleshooting
-

1. Setup Instructions

Prerequisites

- Python 3.7 or higher
- A dataset file (`climate_change_dataset.csv`) containing climate-related data

Step-by-Step Setup

1. Install Python:

- Download and install Python from python.org.
- Ensure `pip` is installed (it comes bundled with Python).

2. Create a Virtual Environment (Optional but Recommended):

```
python -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate
```

3. Install Required Libraries: Install the necessary libraries using `pip`:

```
pip install streamlit pandas numpy plotly scikit-learn
```

4. Prepare the Dataset:

- Place the `climate_change_dataset.csv` file in the same directory as the script or update the file path in the code to point to its location.

5. Run the Streamlit App: Execute the following command to start the app:

```
streamlit run app.py
```

- Open the provided URL (e.g., `http://localhost:8501`) in your browser to view the dashboard.
-

2. Code Overview

The code is divided into several sections:

1. Data Loading and Preprocessing:

- Loads the dataset and converts columns to appropriate data types.
- Handles missing values and removes outliers.

2. Streamlit App:

- Provides an interactive interface for users to filter data by year range and visualize trends.

3. Model Training and Predictions:

- Trains a linear regression model to predict sea surface temperature.
- Displays performance metrics and plots actual vs. predicted values.

3. Data Preprocessing

Data Loading

- The dataset is loaded using `pd.read_csv()` .
- Columns are converted to numeric types (`float64` or `Int64`) to ensure compatibility with mathematical operations.

Handling Missing Values

- Missing values in integer-type columns (`Year` , `Month`) are filled using the median (converted to an integer).
- Missing values in floating-point columns are filled using the mean.

Outlier Removal

- Outliers are detected using the Interquartile Range (IQR) method:
 - Lower Bound = $Q1 - 1.5 * IQR$
 - Upper Bound = $Q3 + 1.5 * IQR$
- For integer-type columns, bounds are rounded to the nearest integer before replacing outliers.

4. Streamlit App Features

Interactive Year Range Slider

- Users can select a range of years to filter the dataset.
- The filtered dataset is displayed with information about its shape and missing values.

Visualizations

1. **Average Temperature Over Years:**
 - Line chart showing how average temperature changes over time.
2. **CO2 Concentration Over Years:**
 - Line chart displaying trends in CO2 concentration.
3. **CO2 Concentration vs. Sea Surface Temperature:**
 - Scatter plot highlighting the relationship between CO2 levels and sea surface temperature.

Model Training Button

- When clicked, the app trains a linear regression model and generates predictions for sea surface temperature.
- Displays:
 - Actual vs. Predicted plot
 - Model performance metrics (Mean Squared Error and R^2 Score)

5. Model Training and Predictions

Input Features

The model uses the following features to predict sea surface temperature:

- `CO2_Concentration` (ppm)
- `Solar_Irradiance` (W/m^2)
- `ENSO_Index`
- `Precipitation` (mm)
- `Humidity` (%)

Output Variable

- `Sea_Surface_Temp` ($^{\circ}C$)

Training Process

1. The dataset is split into training (80%) and testing (20%) sets.
2. A linear regression model is trained on the training set.
3. Predictions are made on the test set, and performance metrics are calculated.

Performance Metrics

- **Mean Squared Error (MSE):** Measures the average squared difference between actual and predicted values.
- **R² Score:** Indicates the proportion of variance in the target variable explained by the model.

6. Dependencies

The following Python libraries are required:

Library	Purpose
<code>streamlit</code>	Build the interactive web application
<code>pandas</code>	Data manipulation and preprocessing
<code>numpy</code>	Numerical computations
<code>plotly</code>	Create interactive visualizations
<code>scikit-learn</code>	Train the linear regression model

Install all dependencies using:

```
pip install streamlit pandas numpy plotly scikit-learn
```

7. Running the Application

1. Save the code to a file named `app.py`.
2. Run the Streamlit app:

```
streamlit run app.py
```

3. Open the provided URL in your browser to interact with the dashboard.

8. Troubleshooting

Common Issues and Fixes

1. **File Not Found Error:**
 - Ensure the dataset file (`climate_change_dataset.csv`) is in the correct location.
 - Update the file path in the code if necessary.
2. **TypeError for Integer Columns:**
 - Ensure that integer-type columns (`Year`, `Month`) are handled correctly during preprocessing and outlier removal.
3. **Empty Dataset After Filtering:**
 - Check the year range slider and ensure it matches the available data.
4. **Missing Dependencies:**
 - Install missing libraries using `pip install <library_name>`.

Conclusion

This Climate Change Data Dashboard provides an intuitive way to explore climate trends and train predictive models. By following the setup instructions and understanding the code structure, you can deploy this application locally or on a cloud platform. If you encounter any issues, refer to the troubleshooting section or consult the official documentation of the libraries used.