Archishman Ghosh and Shanmukha Priya

**Climate Change Analysis Project Documentation**

**Project Overview**

This project aims to analyze the relationship between CO₂ emissions and global temperature anomalies. The goal is to explore how temperature rise and CO₂ emissions have evolved over time, build predictive models for CO₂ emissions and temperature anomalies, and visualize future trends.

**Table of Contents**

1. [Setup Instructions](https://chatgpt.com/c/679b066b-32d4-800c-94b0-cf18323a81dc#setup-instructions)
2. [Analysis Process](https://chatgpt.com/c/679b066b-32d4-800c-94b0-cf18323a81dc#analysis-process)
3. [Data Sources](https://chatgpt.com/c/679b066b-32d4-800c-94b0-cf18323a81dc#data-sources)
4. [Preprocessing Steps](https://chatgpt.com/c/679b066b-32d4-800c-94b0-cf18323a81dc#preprocessing-steps)
5. [Modeling & Methods](https://chatgpt.com/c/679b066b-32d4-800c-94b0-cf18323a81dc#modeling-methods)
6. [API Documentation](https://chatgpt.com/c/679b066b-32d4-800c-94b0-cf18323a81dc#api-documentation)
7. [Testing & Validation](https://chatgpt.com/c/679b066b-32d4-800c-94b0-cf18323a81dc#testing-validation)
8. [GitHub Repository Documentation](https://chatgpt.com/c/679b066b-32d4-800c-94b0-cf18323a81dc#github-repository-documentation)

**1. Setup Instructions**

**1.1 Prerequisites**

**Before starting the project, you'll need to have the following tools and technologies installed on your system:**

**1.1.1 Python 3.x**

* **Python is the programming language used to develop and run the analysis scripts.**
* **This project is compatible with Python 3.6 or later. Python 2 is not recommended, as it has reached the end of its life and is no longer maintained.**

**Installation Instructions:**

* **Windows/Mac/Linux:** 
  + **Download and install Python from the official website:** [**Python.org**](https://www.python.org/downloads/)
  + **Follow the installation instructions for your operating system.**
  + **Ensure that the option "Add Python to PATH" is checked during installation.**

**Verify Installation:**

**After installation, verify that Python is installed correctly by running the following command in the terminal/command prompt:**

**python --version**

**1.1.2 Jupyter Notebook**

* **Jupyter Notebook is an open-source web application that allows you to create and share live code, equations, visualizations, and narrative text.**
* **We use Jupyter Notebook for executing the analysis code and displaying results interactively.**

**Installation Instructions:**

**You can install Jupyter using pip (Python's package manager) by running:**

**pip install jupyter**

**Verify Installation:**

**After installing Jupyter, you can verify that it is working by running:**

**jupyter --version**

**1.1.3 Git**

* **Git is a version control system that helps you track changes in your code over time and collaborate with other developers. It is used to manage and maintain the project's code repository.**

**Installation Instructions:**

* **Windows: Download and install Git from** [**Git-scm**](https://git-scm.com/downloads)**.**
* **Mac/Linux: Git is typically pre-installed on most systems. To check if Git is installed, run the following command:**

**git --version**

**If Git is not installed, you can install it via brew (on macOS) or apt (on Linux):**

**# macOS**

**brew install git**

**# Linux (Ubuntu)**

**sudo apt-get install git**

**1.2 Installing Dependencies**

**This project relies on several Python libraries for data processing, modeling, and visualization. These libraries are listed in the requirements.txt file, and you can install them using the pip command.**

**1.2.1 Install Dependencies via requirements.txt**

1. **Clone the repository to your local machine:**
2. **git clone https://github.com/your-username/climate-analysis-notebook.git**
3. **Navigate to the project directory:**
4. **cd climate-analysis-notebook**
5. **Install all the required dependencies:**
6. **pip install -r requirements.txt**

**1.2.2 List of Required Packages:**

**The main Python libraries used in this project include:**

* **Pandas: For data manipulation and analysis. It provides data structures like DataFrames, which are crucial for processing large datasets.**
* **pip install pandas**
* **NumPy: For numerical operations and handling arrays and matrices. It is often used in conjunction with Pandas for advanced data analysis.**
* **pip install numpy**
* **Matplotlib: For creating static, animated, and interactive visualizations. We use it to plot temperature anomalies, CO₂ emissions, and prediction graphs.**
* **pip install matplotlib**
* **Scikit-learn: For machine learning, including model creation, training, and evaluation. It includes tools for regression, classification, and clustering.**
* **pip install scikit-learn**
* **Seaborn (optional): For statistical data visualization, providing a high-level interface to Matplotlib.**
* **pip install seaborn**

**1.2.3 Install Jupyter Extensions (Optional)**

**If you're using Jupyter Notebooks for your analysis, you might want to install Jupyter extensions like jupyterlab for enhanced notebook functionality:**

**pip install jupyterlab**

**1.3 Running the Project**

**1.3.1 Running Jupyter Notebook**

**Once all the dependencies are installed, you can open the project notebook. In the terminal/command prompt, navigate to the project directory and run:**

**jupyter notebook**

**This will open a browser window where you can view and interact with the Jupyter Notebook file (climate\_analysis.ipynb).**

**1.3.2 Running the Analysis**

* **Open the notebook (climate\_analysis.ipynb) in Jupyter Notebook.**
* **Execute each code cell one by one to run the analysis. Make sure the cells are run in order, as each cell might depend on the outputs from the previous ones.**

**1.4 Project Folder Structure**

**Once you've cloned the repository, the folder structure should look something like this:**

**climate-analysis-notebook/**

**│**

**├── climate\_analysis.ipynb # Main Jupyter Notebook with analysis code**

**├── requirements.txt # List of dependencies to install**

**├── README.md # Project documentation**

**└── data/ # Directory for datasets (if applicable)**

* **climate\_analysis.ipynb: This file contains the entire analysis process, including code for data loading, processing, model training, predictions, and visualizations.**
* **requirements.txt: A plain text file containing all the dependencies necessary for running the project.**
* **README.md: A markdown file that provides an overview of the project and any additional setup instructions.**
* **data/ (optional): If the project includes datasets, they will be located here.**

**Conclusion**

**By following these setup instructions, you should be able to successfully install the necessary tools, download the dependencies, and run the climate change analysis project locally.**

**If you encounter any issues or need further assistance, feel free to check the README.md file for additional information or reach out for support.**

**2. Analysis Process**

**The analysis process involves a series of steps designed to understand the relationship between CO₂ emissions and global temperature anomalies over time. The steps include loading data, cleaning and preprocessing it, exploring the data for patterns, building models, and making predictions. Let's go through the process in detail:**

**2.1 Data Sources**

**The primary data sources used in this analysis are historical global temperature anomaly data and CO₂ emission data. Both datasets come from publicly available scientific sources and can be accessed from trusted research institutions like NASA and NOAA.**

**2.1.1 Global Temperature Anomaly Data**

**The temperature anomaly data used in this project is from NASA's GISS (Goddard Institute for Space Studies) Global Surface Temperature dataset. This dataset records monthly anomalies in temperature relative to a baseline period.**

* **Dataset Source:** [**NASA GISS Surface Temperature Data**](https://data.giss.nasa.gov/gistemp/)
* **File Format: CSV file format, where each row represents the year and the average temperature anomaly for each month.**

**2.1.2 CO₂ Emission Data**

**The CO₂ emissions data comes from a combination of multiple scientific sources, including reports from NOAA, the Global Carbon Project, and the United Nations. This dataset includes historical global CO₂ emissions, typically measured in gigatonnes (Gt).**

* **Dataset Source:** [**Global Carbon Project**](https://www.globalcarbonproject.org/)
* **File Format: CSV or similar formats with annual emission records.**

**Both of these datasets are critical for understanding the global climate change phenomenon. The temperature anomaly data reveals how the Earth's temperature has changed over time, while the CO₂ data provides insight into the amount of carbon dioxide released into the atmosphere.**

**2.2 Data Preprocessing**

**Data preprocessing is a critical step in any data analysis project. The goal is to clean, structure, and transform the raw data into a format suitable for analysis and model training. Below are the key preprocessing steps taken in this project:**

**2.2.1 Data Import and Inspection**

**First, we load the datasets into the Python environment using the Pandas library. Pandas makes it easy to load, explore, and manipulate large datasets.**

* **Loading Data:**

**python**

**CopyEdit**

**import pandas as pd**

**# Load temperature anomaly data**

**temperature\_df = pd.read\_csv('global\_temperature\_anomaly.csv')**

**# Load CO2 emissions data**

**co2\_df = pd.read\_csv('co2\_emissions.csv')**

**Once the data is loaded, we inspect it to ensure that it has been loaded correctly and that it conforms to the expected structure.**

**python**

**CopyEdit**

**print(temperature\_df.head()) # Show the first few rows of the temperature dataset**

**print(co2\_df.head()) # Show the first few rows of the CO2 dataset**

**2.2.2 Missing Data Handling**

**Real-world datasets often have missing or incomplete data. This can lead to inaccurate results if not properly handled. In this project, we used mean imputation to handle missing values in the datasets.**

**python**

**CopyEdit**

**from sklearn.impute import SimpleImputer**

**# Handle missing values by replacing them with the mean of the column**

**imputer = SimpleImputer(strategy='mean')**

**temperature\_df = imputer.fit\_transform(temperature\_df)**

**co2\_df = imputer.fit\_transform(co2\_df)**

**2.2.3 Data Transformation**

**Data transformation may be required to align the two datasets in terms of time and structure. Since the datasets may have different time intervals or units, we might need to:**

* **Normalize both datasets by aligning them to a common timeframe (e.g., year-based analysis).**
* **Ensure both datasets are in the same format (e.g., same column names, similar structures).**

**For example:**

**python**

**CopyEdit**

**# Ensure both datasets have 'Year' column**

**temperature\_df['Year'] = temperature\_df['Year'].astype(int)**

**co2\_df['Year'] = co2\_df['Year'].astype(int)**

**2.2.4 Merging Datasets**

**Once the data is cleaned and transformed, the next step is to merge the two datasets based on the common 'Year' column. This allows us to analyze the relationship between temperature anomalies and CO₂ emissions over the same time period.**

**python**

**CopyEdit**

**# Merge the two datasets on the 'Year' column**

**merged\_df = pd.merge(temperature\_df, co2\_df, on='Year')**

**2.3 Exploratory Data Analysis (EDA)**

**Exploratory Data Analysis (EDA) is a crucial step to understand the characteristics and patterns of the data before building models. It involves plotting, visualizing, and summarizing the data to identify trends, outliers, and correlations.**

**2.3.1 Visualizing Temperature Anomalies and CO₂ Emissions**

**We used Matplotlib and Seaborn to visualize the trends and patterns in the data.**

**python**

**CopyEdit**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**# Plot temperature anomalies over the years**

**plt.figure(figsize=(10, 5))**

**plt.plot(merged\_df['Year'], merged\_df['Temperature\_Anomaly'], label='Temperature Anomaly')**

**plt.title('Global Temperature Anomalies Over Time')**

**plt.xlabel('Year')**

**plt.ylabel('Temperature Anomaly (°C)')**

**plt.legend()**

**plt.grid(True)**

**plt.show()**

**# Plot CO2 emissions over the years**

**plt.figure(figsize=(10, 5))**

**plt.plot(merged\_df['Year'], merged\_df['CO2'], label='CO2 Emissions', color='green')**

**plt.title('Global CO2 Emissions Over Time')**

**plt.xlabel('Year')**

**plt.ylabel('CO2 Emissions (Gt)')**

**plt.legend()**

**plt.grid(True)**

**plt.show()**

**These plots help to visualize the trends over time and can serve as a foundation for further analysis.**

**2.3.2 Correlation Analysis**

**One important aspect of EDA is understanding the relationship between the two key variables: temperature anomalies and CO₂ emissions. We calculate the correlation coefficient to understand how strongly the two variables are related.**

**python**

**CopyEdit**

**# Calculate correlation between temperature anomaly and CO2 emissions**

**correlation = merged\_df[['Temperature\_Anomaly', 'CO2']].corr()**

**print(correlation)**

**2.4 Model Building**

**After conducting exploratory data analysis, the next step is to build a model that can predict temperature anomalies based on CO₂ emissions (or vice versa). We used linear regression to model the relationship between CO₂ emissions and temperature anomalies.**

**2.4.1 Preparing Data for Modeling**

**Before building the model, we split the data into features (independent variables) and targets (dependent variables). In this case, CO₂ emissions will serve as the feature and temperature anomaly as the target.**

**python**

**CopyEdit**

**from sklearn.model\_selection import train\_test\_split**

**# Define features (X) and target (y)**

**X = merged\_df[['CO2']] # CO2 as the feature**

**y = merged\_df['Temperature\_Anomaly'] # Temperature Anomaly as the target**

**# Split the data into training and testing sets (80% training, 20% testing)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**2.4.2 Linear Regression Model**

**We used scikit-learn's LinearRegression model to build the prediction model. Linear regression is a simple yet powerful technique for predicting continuous variables.**

**python**

**CopyEdit**

**from sklearn.linear\_model import LinearRegression**

**# Initialize the model**

**model = LinearRegression()**

**# Train the model**

**model.fit(X\_train, y\_train)**

**# Predict on the test set**

**y\_pred = model.predict(X\_test)**

**2.4.3 Model Evaluation**

**We evaluated the performance of the model using mean squared error (MSE) and R-squared (R²) to assess how well the model is fitting the data.**

**python**

**CopyEdit**

**from sklearn.metrics import mean\_squared\_error, r2\_score**

**# Evaluate the model's performance**

**mse = mean\_squared\_error(y\_test, y\_pred)**

**r2 = r2\_score(y\_test, y\_pred)**

**print(f'Mean Squared Error: {mse}')**

**print(f'R² Score: {r2}')**

**2.5 Future Predictions**

**Once the model is trained and evaluated, we can use it to make predictions for future years. We create a new dataset of future CO₂ emissions and use the model to predict temperature anomalies for those years.**

**python**

**CopyEdit**

**# Example of predicting future temperature anomalies**

**future\_co2 = pd.DataFrame({'CO2': [400, 420, 440, 460, 480]}) # Example future CO2 emissions**

**future\_temperature\_anomalies = model.predict(future\_co2)**

**print(future\_temperature\_anomalies)**

**This process allows us to make data-driven predictions about future temperature anomalies based on projected CO₂ emissions.**

**2.6 Conclusion of Analysis**

**In summary, the analysis process involves importing, cleaning, transforming, and merging the data, followed by exploratory data analysis (EDA) to understand the relationships and patterns. After building a predictive model using linear regression, we validated its performance and made predictions for future scenarios.**

**3. 3. Data Sources**

**The success of any data science project hinges on the quality and relevance of the data used. In this project, the main goal was to analyze the relationship between CO₂ emissions and global temperature anomalies. To achieve this, we utilized two primary datasets: one for global temperature anomalies and one for CO₂ emissions. Below, we’ll explore both of these data sources in detail.**

**3.1 Global Temperature Anomaly Data**

**The global temperature anomaly data is crucial to understand how Earth's surface temperature has changed over time relative to a baseline period. This data provides insights into the overall warming trend and is essential for analyzing climate change.**

**3.1.1 Source of Global Temperature Data**

* **Source: The primary source of the temperature anomaly data is NASA's GISS Surface Temperature Analysis (GISTEMP).**
* **Link: The data can be accessed at** [**NASA GISS GISTEMP**](https://data.giss.nasa.gov/gistemp/)**.**
* **Description: The GISTEMP dataset is a widely used resource that records global surface temperatures, adjusted for known biases such as changes in instrumentation or station placement. It provides monthly and annual temperature anomalies, which represent how much the observed temperatures differ from a long-term average (baseline period).**

**3.1.2 Data Structure**

**The dataset is structured with one row per year (or month in some versions), and each column represents a different variable. In the version we used, it contains:**

* **Year: The year of the temperature record.**
* **Temperature Anomalies: The anomalies in temperature relative to a reference period, typically expressed in degrees Celsius.**

**Example of the first few rows in the dataset:**

**mathematica**

**CopyEdit**

**Year Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec J-D D-N**

**1880 -0.20 -0.26 -0.09 -0.17 -0.10 -0.22 -0.21 -0.11 -0.16 -0.23 -0.23 -0.19 -0.18 \*\*\***

**1881 -0.20 -0.16 0.02 0.03 0.06 -0.19 0.00 -0.05 -0.16 -0.22 -0.19 -0.08 -0.10 -0.11**

**1882 0.15 0.13 0.04 -0.17 -0.14 -0.23 -0.17 -0.08 -0.15 -0.24 -0.17 -0.37 -0.12 -0.09**

**3.1.3 Purpose in Analysis**

**The temperature anomaly data is used to track changes in the Earth's surface temperature over time. These anomalies help to quantify global warming by comparing the observed temperatures to a reference baseline, which provides an understanding of whether the Earth is warming or cooling over time. This data is critical to the project as we are analyzing the correlation between CO₂ emissions and temperature anomalies.**

**3.1.4 Data Challenges**

**Some challenges with the temperature anomaly data include:**

* **Missing values: Like most real-world data, some parts of the temperature anomaly dataset may have missing or incomplete data.**
* **Time Granularity: Temperature data may be available monthly or annually, and for consistency, we use annual averages for analysis.**

**To ensure we have a continuous dataset, any missing values need to be handled carefully (for example, by using imputation techniques).**

**3.2 CO₂ Emission Data**

**Carbon dioxide (CO₂) emissions are one of the major contributors to climate change. The CO₂ emissions dataset provides information about the amount of CO₂ emitted globally each year. This dataset is critical in understanding how human activity, particularly industrialization and deforestation, has affected the planet’s climate over time.**

**3.2.1 Source of CO₂ Emissions Data**

* **Source: The primary source for the CO₂ emissions data is the Global Carbon Project, which provides detailed records of global CO₂ emissions from fossil fuels, land use, and other anthropogenic sources.**
* **Link: The data can be accessed at** [**Global Carbon Project**](https://www.globalcarbonproject.org/)**.**
* **Description: The Global Carbon Project provides annual data on CO₂ emissions and breakdowns by sector, country, and type of emission (e.g., fossil fuels vs. land use). The dataset offers a global view of the carbon dioxide released each year and is widely regarded as an authoritative source.**

**3.2.2 Data Structure**

**The CO₂ emissions dataset includes several important features, such as:**

* **Year: The year of the emission record.**
* **CO₂ Emissions: The total CO₂ emissions in gigatonnes (Gt) released into the atmosphere globally in that year.**
* **Breakdowns by Sector (optional): Emissions from different sectors such as energy, transportation, and industry.**

**Example of the first few rows in the CO₂ emissions dataset:**

**yaml**

**CopyEdit**

**Year CO2 Emissions (Gt)**

**1880 0.50**

**1881 0.52**

**1882 0.55**

**1883 0.59**

**1884 0.63**

**3.2.3 Purpose in Analysis**

**The CO₂ emissions data is used as the independent variable in our analysis. By analyzing the trends and patterns of CO₂ emissions over time, we aim to understand their impact on global temperature anomalies. This data serves as an indicator of human-induced climate change and helps explain why temperature anomalies are rising globally.**

**3.2.4 Data Challenges**

**The challenges related to the CO₂ emissions dataset include:**

* **Data Consistency: Different organizations and governments might report CO₂ emissions differently. It is important to ensure that the data we use is consistent and comparable.**
* **Missing Data: Similar to the temperature data, CO₂ emissions data may also contain missing values, which need to be handled appropriately.**

**3.3 Data Collection and Storage**

**Both datasets were collected from their respective sources (NASA for temperature anomalies and the Global Carbon Project for CO₂ emissions). The data was downloaded in CSV format and stored in the project’s directory.**

* **Storage Format: CSV files for both datasets, making it easy to import and manipulate them in Python using libraries like Pandas.**
* **File Paths: The files are stored locally or on the Kaggle dataset platform, with file paths specified when loading the data into the environment.**

**3.4 Data Integration**

**Once the datasets were acquired, they were integrated into a single unified dataset based on the common Year column. This step allows for the analysis of both CO₂ emissions and temperature anomalies over the same time period, ensuring that the correlation analysis is valid.**

**Example of Data Integration:**

**python**

**CopyEdit**

**# Merge the temperature anomaly and CO2 emissions data on 'Year'**

**merged\_df = pd.merge(temperature\_df, co2\_df, on='Year')**

**By merging the datasets on the Year column, we ensure that the corresponding temperature anomaly and CO₂ emission values are aligned correctly, making it possible to analyze their relationship.**

**3.5 Data Quality and Validation**

**Ensuring data quality is crucial for the success of the analysis. Both datasets were carefully examined for missing values, duplicates, or inconsistencies before proceeding with any data exploration or modeling.**

* **Missing Data: If missing data was found, we used techniques like mean imputation or interpolation to fill in the gaps.**
* **Data Validation: The data was validated by checking for unusual or outlier values, ensuring that each dataset accurately reflected the scientific measurements it was derived from.**

**3.6 Conclusion of Data Sources**

**The temperature anomaly data and CO₂ emissions data are both essential for understanding the key drivers of global climate change. These datasets offer a comprehensive view of temperature trends and human-induced carbon emissions, making them ideal for exploring the correlation between CO₂ levels and temperature anomalies.**

**By sourcing data from well-established and reputable institutions like NASA and the Global Carbon Project, we ensure that the analysis is based on accurate and reliable information.**

**Preprocessing Steps**

1. **Load Dataset**: The dataset was loaded into a Pandas DataFrame for easier manipulation.
2. **Handling Missing Data**: Missing data in the temperature anomalies and CO₂ emissions columns was imputed using the mean value of the respective columns.
3. **Data Selection**: We selected only the Year, Month, CO2, and Temperature\_Anomaly columns for the analysis.
4. **Feature Engineering**: Features include Year, Month, and CO2 as predictors for temperature anomaly, and the year and month to predict CO₂ emissions.

**Modeling & Methods**

We implemented linear regression models to predict both temperature anomalies and CO₂ emissions based on historical data. The following steps were taken:

1. **Data Split**: The data was split into training and test sets (80% training, 20% test) using train\_test\_split from sklearn.
2. **Linear Regression**: Two models were created using LinearRegression from sklearn:
   * One for predicting temperature anomalies (model\_temp).
   * One for predicting CO₂ emissions (model\_co2).
3. **Evaluation**: The models were evaluated using mean\_squared\_error and r2\_score.
4. **Future Predictions**: We forecasted future CO₂ emissions over the next 5 years by extending the dataset with new data points and using the trained model to predict CO₂ levels.

**Data Visualizations**

* **Temperature Anomalies**: A scatter plot was used to visualize the relationship between predicted and actual temperature anomalies.
* **CO₂ Emissions**: A line plot was used to visualize the historical CO₂ emissions, and future predictions were plotted to show expected trends.

**API Documentation**

This project does not directly interface with external APIs or cloud services, so no specific API documentation is required.

**Testing & Validation**

**1. Model Validation**

The models were validated using the following steps:

* **Cross-validation**: We used cross-validation to ensure the models performed well on unseen data.
* **Metrics**: We evaluated the models using:
  + **Mean Squared Error (MSE)**: To measure the error between actual and predicted values.
  + **R² Score**: To assess how well the models fit the data.

**2. Future Prediction Validation**

To validate the future predictions, we checked the stability of the model over different periods and ensured that it was not overfitting to the training data.

**GitHub Repository Documentation**

**1. Repository Structure**

The repository contains the following files and folders:

* **climate\_analysis.ipynb**: The Jupyter notebook containing the project code and analysis.
* **requirements.txt**: A text file listing all the required dependencies for the project.
* **README.md**: This documentation file.

**2. Setup Instructions**

* Clone the repository:
* git clone https://github.com/your-username/climate-analysis-notebook.git
* Navigate into the project directory and install dependencies:
* cd climate-analysis-notebook
* pip install -r requirements.txt

**3. Usage Guidelines**

To run the project, open the Jupyter notebook:

jupyter notebook climate\_analysis.ipynb

The notebook is well-commented, and you can run the cells in order to replicate the analysis.

**Conclusion**

This project provides valuable insights into the relationship between CO₂ emissions and global temperature anomalies, leveraging machine learning for predictions and visualizations. By following the provided setup instructions and running the code in the notebook, users can replicate and expand upon this analysis.