**Software Requirements Specification**

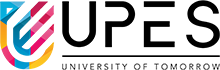
For

AdaptiPlan: Intelligent Scenario Modelling for Climate Change Mitigation using Computational Statistical Model

10th October, 2024

Prepared by

|  |  |  |
| --- | --- | --- |
| **Specialization** | **SAP ID** | **Name** |
| AIML HONS. | 500090912 | CHARU GUPTA |
| AIML HONS. | 500094127 | LAKSHAY AGARWAL |



Artificial Intelligence Cluster

School of Computer Science

UPES Dehradun – 248007, Uttarakhand

Table of Contents

|  |  |  |
| --- | --- | --- |
| **Topic** | | **Page No** |
|  | |  |
| 1 | Introduction | 3 |
|  | 1.1 Purpose of the Project | 3 |
|  | 1.2 Target Beneficiary | 3 |
|  | 1.3 Project Scope | 3 |
|  | 1.4 References | 4 |
| 2 | Project Description | 5 |
|  | 2.1 Reference Algorithm | 5 |
|  | 2.2 Data/ Data structure | 7 |
|  | 2.3 SWOT Analysis | 7 |
|  | 2.4 Project Features | 7 |
|  | 2.5 User Classes and Characteristics | 8 |
|  | 2.6 Design and Implementation Constraints | 9 |
|  | 2.7 Design diagrams | 10 |
|  | 2.8 Assumption and Dependencies | 12 |
| 3 | System Requirements | 12 |
|  | 3.1 User Interface | 12 |
|  | 3.2 Software Interface | 12 |
|  | 3.3 Database Interface | 12 |
|  | 3.4 Protocols | 12 |
| 4 | Non-functional Requirements | 12 |
|  | 4.1 Performance requirements | 12 |
|  | 4.2 Security requirements | 12 |
|  | 4.3 Software Quality Attributes | 13 |
| Appendix A: Glossary | | 14 |

1. INTRODUCTION:

Purpose of the project:

Climate change poses significant risks to ecosystems, economies, and societies globally. Accurate forecasting of climate variables such as temperature, precipitation, and extreme weather events is crucial for developing effective mitigation strategies. Traditional climate models, while robust, often require substantial computational resources and may not capture localized patterns effectively.

The primary purpose of the "AdaptiPlan" project is to develop a predictive tool that helps model future climate change scenarios using a combination of advanced machine learning and statistical methods, namely Long Short-Term Memory (LSTM), Autoregressive Integrated Moving Average (ARIMA), and Monte Carlo Simulation.

The goal is to determine the most accurate and reliable model for predicting future environmental conditions, and integrate this model into a web-based application to assist policymakers and industries in decision-making for climate adaptation and mitigation.

Target Beneficiary

The target beneficiaries of this project are:

* **Policymakers**: To enable informed decisions based on data-driven climate predictions and manage risks related to climate change.
* **Industries**: Particularly agriculture, insurance, and energy sectors, which rely heavily on climate conditions for operational efficiency and future planning.
* **Environmental Researchers**: To offer accurate tools for simulating and analyzing environmental conditions and trends.
* **Communities**: To improve resilience to future climate events by enabling better preparedness and adaptation strategies.

Project Scope

* Development of a Predictive Climate Scenario Modeling System: Create a reliable system capable of predicting climate change impacts using the ERA5 dataset and integrating advanced computational statistical models.
* Integration of Advanced Machine Learning and Statistical Techniques: Utilize LSTM, ARIMA, and Monte Carlo Simulation to enhance the accuracy and reliability of climate scenario predictions.
* Data Collection and Preprocessing: Implement data acquisition strategies for the ERA5 dataset using the CDS API, followed by preprocessing techniques to ensure data quality and model compatibility.
* Feature Extraction and Time Series Forecasting: Employ time series analysis and feature extraction techniques to uncover patterns in historical climate data, improving the adaptability and performance of predictive models.
* Comparative Analysis of Predictive Models: Conduct a comprehensive evaluation of LSTM, ARIMA, and Monte Carlo Simulation to identify the most effective model for climate forecasting, and integrate the best-performing model into a web-based application.

References

[1] Xu, J., Wang, Z., Li, X., Li, Z., & Li, Z. (2024). Prediction of Daily Climate Using Long Short-Term Memory (LSTM) Model. International Journal of Innovative Science and Research Technology, 83-90.

[2] Lai, Y., & Dzombak, D. A. (2020). Use of the autoregressive integrated moving average (ARIMA) model to forecast near-term regional temperature and precipitation. Weather and forecasting, 35(3), 959-976.

[3] Bonate, P. L. (2001). A brief introduction to Monte Carlo simulation. Clinical pharmacokinetics, 40, 15-22.

[4] Vo, T. Q., Kim, S. H., Nguyen, D. H., & Bae, D. H. (2023). LSTM-CM: a hybrid approach for natural drought prediction based on deep learning and climate models. *Stochastic Environmental Research and Risk Assessment*, *37*(6), 2035-2051.

[5] Dimri, T., Ahmad, S., & Sharif, M. (2020). Time series analysis of climate variables using seasonal ARIMA approach. *Journal of Earth System Science*, *129*, 1-16.

[6] New, M., & Hulme, M. (2000). Representing uncertainty in climate change scenarios: a Monte-Carlo approach. *Integrated assessment*, *1*(3), 203-213.

1. PROJECT DESCRIPTION:

Reference Algorithm

* LSTM (Long Short-Term Memory): A deep learning model designed to handle time-series data by capturing both long-term and short-term dependencies. It's well-suited for modeling climate patterns due to its ability to learn complex temporal relationships.

Input:

Input sequence X = [x₁, x₂, ..., xₜ] where xₜ ∈ ℝⁿ at time step t, T is the total length of the sequence.

Initial hidden state h₀, and cell state c₀.

Model parameters: input weight matrices Wᵢ, Wf, Wₒ, Wc, recurrent weight matrices Uᵢ, Uf, Uₒ, Uc, bias vectors bᵢ, bf, bₒ, bc.

Output:

Output sequence H = [h₁, h₂, ..., hₜ], where each hₜ ∈ ℝᵐ, and final hidden and cell states hₜ, cₜ.

Step 1: Initialization

Initialize the hidden state h₀ and cell state c₀ to zero vectors: h₀ = 0, c₀ = 0.

Step 2: Iterate through the sequence

For each time step t = 1, 2, ..., T, repeat the following steps:

Step 2.1: Forget gate computation

Compute the forget gate vector fₜ, which decides what information from the previous cell state to forget:

fₜ = σ(Wf xₜ + Uf hₜ₋₁ + bf)

Step 2.2: Input gate computation

Compute the input gate vector iₜ, which decides what new information to store in the cell state:

iₜ = σ(Wᵢ xₜ + Uᵢ hₜ₋₁ + bᵢ)

Step 2.3: Candidate cell state

Compute the candidate cell state c̃ₜ, which is the potential update for the cell state:

c̃ₜ = tanh(Wcxₜ + Uc hₜ₋₁ + bc)

Step 2.4: Update cell state

Update the cell state cₜ using the forget gate and the input gate:

cₜ = fₜ ⊙ cₜ₋₁ + iₜ ⊙ c̃ₜ

Step 2.5: Output gate computation

Compute the output gate vector oₜ, which decides the next hidden state:

oₜ = σ(Wₒ xₜ + Uₒ hₜ₋₁ + bₒ)

Step 2.6: Update hidden state

Update the hidden state hₜ based on the cell state and the output gate:

hₜ = oₜ ⊙ tanh(cₜ)

Step 3: Final Output

After iterating through all time steps t = 1, 2, ..., T, the final hidden state hₜ and cell state cₜ are obtained. The output sequence H = [h₁, h₂, ..., hₜ] represents the final hidden states at each time step.

Step 4: Prediction

For tasks like sequence prediction, apply a fully connected layer to the final hidden state hₜ (or each hₜ) to map it to the desired output dimension.

* ARIMA (Autoregressive Integrated Moving Average): A classical statistical model used for time-series forecasting. ARIMA captures trends and patterns over time, making it suitable for forecasting climate data with temporal correlations.

Input:

Time series data X = [x₁, x₂, ..., xₜ] where xₜ is the value at time step t.  
Parameters p, d, and q:  
 - p: Order of the autoregressive (AR) model.  
 - d: Degree of differencing.  
 - q: Order of the moving average (MA) model.

Output:

Forecasted values ˆxₜ₊₁, ˆxₜ₊₂, ...

Step 1: Differencing

Apply differencing d times to the time series X to make it stationary:  
Yₜ = Δ^d Xₜ  
where Δ represents differencing.

Step 2: AR Model

Fit an AR model of order p:  
Yₜ = φ₁ Yₜ₋₁ + φ₂ Yₜ₋₂ + ... + φₚ Yₜ₋ₚ + εₜ  
where εₜ is the error term.

Step 3: MA Model

Fit an MA model of order q:  
εₜ = θ₁ εₜ₋₁ + θ₂ εₜ₋₂ + ... + θ\_q εₜ₋\_q + Zₜ  
where Zₜ is white noise.

Step 4: Combining AR and MA

Combine the AR and MA components to form the ARIMA model:  
ˆxₜ = φ₁ ˆxₜ₋₁ + ... + φₚ ˆxₜ₋ₚ + θ₁ εₜ₋₁ + ... + θ\_q εₜ₋\_q

Step 5: Forecasting

Use the fitted ARIMA model to forecast future values.

* Monte Carlo Simulation: A probabilistic model that simulates a wide range of possible outcomes by incorporating random inputs. It’s useful for assessing uncertainty in climate predictions by running multiple scenarios.

Input:

Probability distribution P of possible outcomes.  
Number of simulations N.

Output:

Estimated outcome based on the average of simulations.

Step 1: Initialize

Set the number of simulations N and define the probability distribution P.

Step 2: Simulate

For each simulation i = 1, 2, ..., N:  
 - Draw a random sample xᵢ from the distribution P.  
 - Compute the outcome yᵢ based on xᵢ.

Step 3: Aggregate Results

Calculate the mean of all simulated outcomes:  
ˆy = (1/N) Σᵢⁿ=₁ yᵢ

Step 4: Output

Return the estimated outcome ˆy.

Dataset

The ERA5 dataset, which provides hourly estimates of climate variables such as temperature, humidity, wind speed, and precipitation for several decades.

The data is structured in a time-series format, where each entry corresponds to a specific climate variable at a specific time (e.g., temperature on January 1st, 2020, at 12:00 pm).

SWOT Analysis

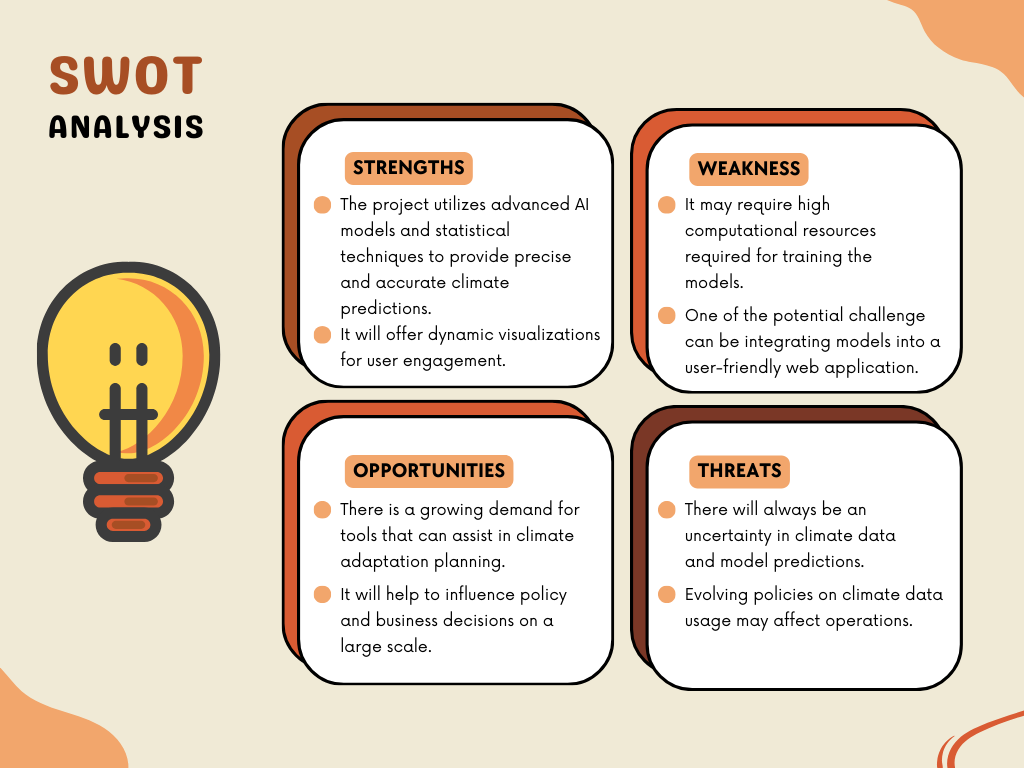


Fig 2.1. SWOT

Project Features

* Predictive Climate Scenario Modeling:

Objective: Create a predictive model to simulate future climate scenarios.

Key Components: Develop predictive models using LSTM, ARIMA, and Monte Carlo Simulation to forecast long-term climate trends such as temperature, precipitation, and other environmental factors. This will enable the exploration of potential future climate conditions based on historical climate data.

* Comparative Analysis of Predictive Algorithms:

Objective: Compare the performance of multiple time-series forecasting methods.

Key Components: Perform a detailed analysis of three major models (LSTM, ARIMA, Monte Carlo Simulation). Each algorithm will be evaluated based on metrics like accuracy, speed, and adaptability to the dataset, ensuring the best model is selected for implementation.

* Data Processing and Feature Engineering:

Objective: Enhance the predictive capability of the models by preprocessing the dataset effectively.

Key Components: Implement data preprocessing techniques to clean and transform the ERA5 dataset. This includes handling missing values, normalizing climate variables, and generating relevant features that improve model performance.

* Web Application Development for Interactive Scenario Exploration:

Objective: Build an accessible platform for users to simulate climate scenarios.

Key Components: Develop a web-based interface where users can input various climate variables or choose from predefined scenarios. The system will then simulate future climate impacts based on the chosen model, offering insights for risk assessment and adaptation strategies.

User Classes and Characteristics

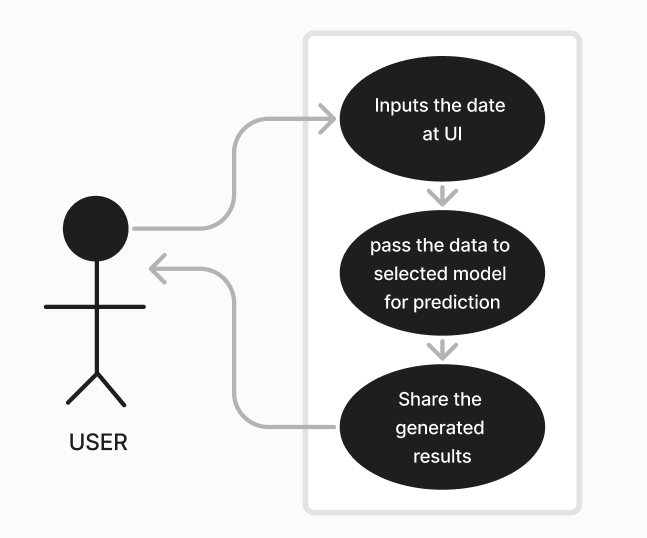


Fig 2.2. UML

It can be used for the following user classes:

* Researchers
* Policymakers

Design and Implementation Constraints

1. Design Constraints:

* Hardware Limitations:

The design of the AdaptiPlan system must consider the hardware limitations of the environment where it will be deployed. This includes memory, processing power, and storage limitations, especially when handling large climate datasets like ERA5. The system must be efficient enough to run on cloud servers or local systems with moderate configurations.

* Software Compatibility:

AdaptiPlan must ensure compatibility with the libraries, APIs, and platforms used in climate data analysis and machine learning. It should integrate seamlessly with climate data sources and support commonly used software stacks such as TensorFlow, Keras, or PyTorch for model training and deployment.

* Scalability Considerations:

The system should be designed with scalability in mind. As the dataset grows or more users start using the platform, the system should efficiently handle larger datasets and increased processing demands without significant performance degradation. Scalability considerations will also affect how the web application manages multiple simulations concurrently and how it stores historical climate predictions.

* Accuracy and Precision Requirements:

The system must meet strict accuracy standards for predicting climate scenarios, as inaccuracies could lead to flawed adaptation strategies. Therefore, precision in model design, dataset usage, and post-processing is paramount, and the system design should ensure high-quality results.

1. Implementation Constraints:

* Technology Stack: The technology stack for AdaptiPlan will likely be constrained by the need to work with specific machine learning frameworks e.g., TensorFlow, Keras and platforms that can handle large-scale data and complex computations. The choice of technologies will also depend on the expertise and availability of resources for implementing machine learning models and web applications.
* Algorithm Complexity: The implementation of LSTM, ARIMA, and Monte Carlo simulation models can be computationally intensive, especially with large datasets. Resource constraints such as CPU or GPU availability and memory limits may require optimizing the algorithms to ensure they run efficiently without sacrificing prediction accuracy.
* Data Availability and Quality: The success of the models depends on the availability and quality of climate data, such as the ERA5 dataset. Limitations in data granularity, missing values, or data biases could impact the model’s ability to generalize well to different scenarios. Extensive data preprocessing and augmentation might be necessary to mitigate these issues and improve model robustness.
* Testing and Validation: AdaptiPlan’s models must undergo rigorous testing and validation to ensure their predictions are reliable. Constraints might arise from limited access to real-world validation datasets or a lack of benchmarking tools for climate scenario modeling. Additionally, the web application must be tested to handle different user inputs, and real-time simulations under various conditions must be validated to ensure accuracy and responsiveness.
* Regulatory and Compliance Issues: The project may face constraints related to compliance with environmental data regulations or security protocols for handling large datasets. The system must ensure data privacy and adhere to global environmental data management standards where applicable.

Design Diagram

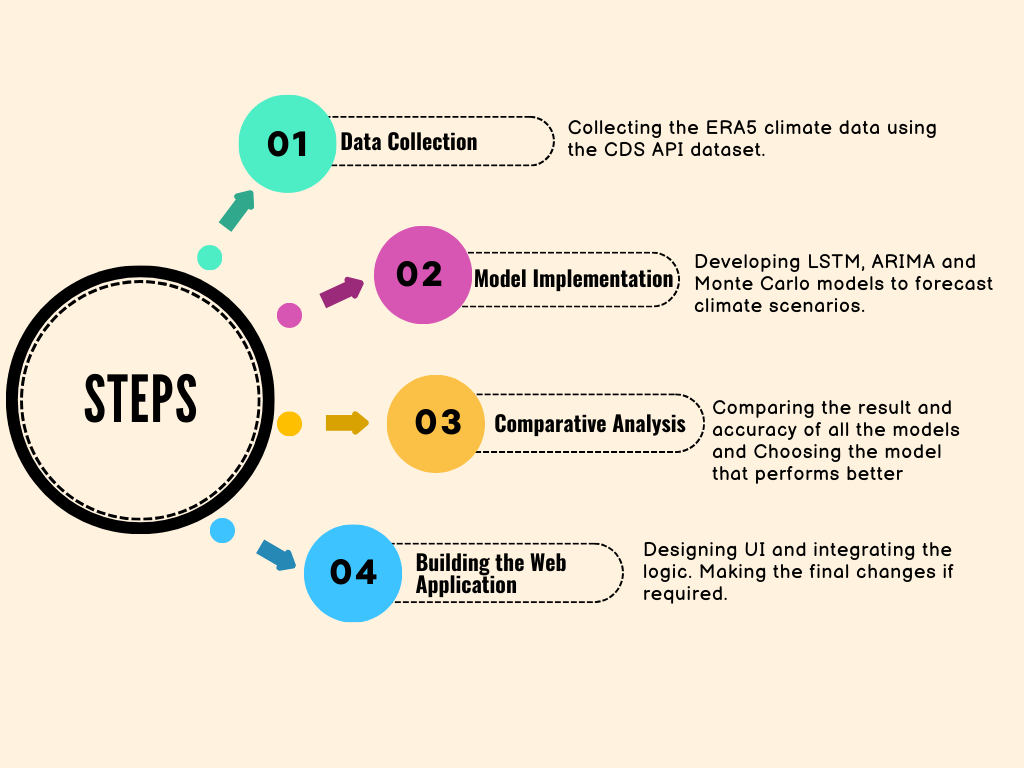


Fig 2.3. Workflow

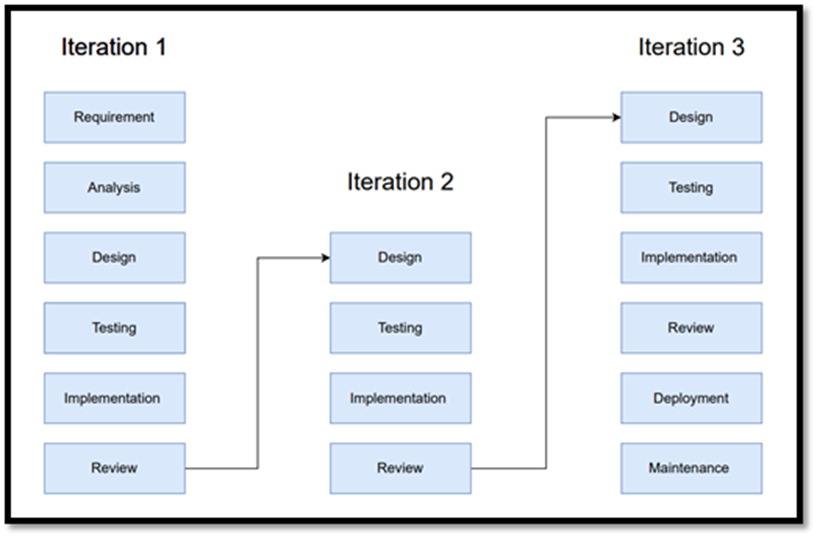


Fig 2.4. Iterative Model

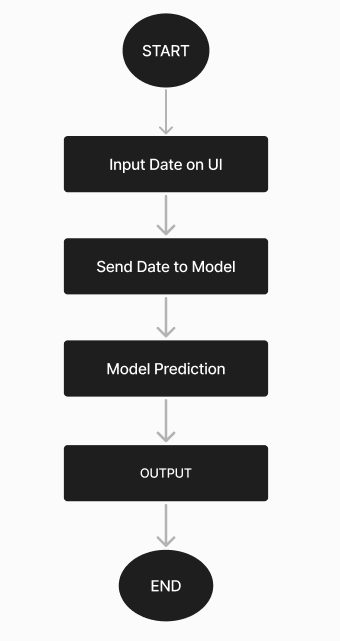


Fig 2.5. Flowchart

Assumption and Dependencies

* Assumes availability of historical climate data from the ERA5 dataset.
* Depends on machine learning frameworks like TensorFlow for LSTM, Statsmodels for ARIMA.
* Relies on APIs for fetching real-time climatic data.

1. SYSTEM REQUIREMENTS:

User Interface

The user interface (UI) of the web application will be clean and intuitive, allowing users to:

* Input desired parameters for climate scenario simulation
* Get the output based on the predictive model.

Software Interface

* Backend: The web application backend will be built using Python, leveraging framework Flask
* Frontend: HTML, CSS, and JavaScript will be used for the front-end interface.
* Integration: Machine learning models will be integrated via RESTful APIs.

Protocols

* HTTP/HTTPS: Used for communication between users and the web application.
* API Protocols: For fetching and interacting with external data sources like the ERA5 dataset (using the CDS API).

1. NON-FUNCTIONAL REQUIREMENTS:

Performance Requirements

* Efficiency: The system should be able to handle large datasets without performance bottlenecks.
* Response Time: Predictions and simulations should be generated within a reasonable timeframe, considering the computational complexity.
* Scalability: The system should be scalable to handle an increasing number of users or larger datasets in the future.

Security Requirements

Data Protection: All user inputs and results must be protected using secure encryption protocols.

Software Quality Attributes

* Reliability: The system must provide consistent, accurate predictions across different scenarios.
* Maintainability: The system should be modular and easy to update as new data or improved algorithms become available.
* Usability: The interface should be user-friendly, even for non-technical users.
* Portability: The application should be deployable across different platforms, such as cloud environments.

1. APPENDIX A: GLOSSARY

* LSTM (Long Short-Term Memory)- A type of recurrent neural network (RNN) designed to process and predict time series data by capturing long-term dependencies and preventing issues like vanishing gradients. It is highly effective for tasks such as climate forecasting, where relationships over time are crucial.
* ARIMA (Autoregressive Integrated Moving Average)- A statistical time series forecasting model that combines autoregression, differencing (to make data stationary), and moving averages to predict future points in a series. It is widely used for climate data forecasting due to its simplicity and effectiveness in handling linear trends.
* Monte Carlo Simulation- A stochastic or random method that uses repeated random sampling to calculate probable outcomes and model uncertainty in a system. In the context of climate prediction, Monte Carlo simulation helps in understanding the range of possible future climate scenarios by accounting for uncertainties in model inputs.
* ERA5 Dataset- A global atmospheric reanalysis dataset produced by the European Centre for Medium-Range Weather Forecasts. It provides hourly estimates of various atmospheric, land, and oceanic climate variables, making it a critical resource for modeling and forecasting climate changes.
* Time Series Data- A sequence of data points recorded at successive time intervals. In climate science, time series data are used to track changes in environmental factors like temperature, precipitation, and humidity over time.
* CDS API (Climate Data Store API)- An API that provides access to climate datasets, including the ERA5 dataset. The API allows users to programmatically download large climate datasets for research, analysis, and modeling.
* Stochastic Modeling- A type of modeling that incorporates random variables and probabilistic outcomes to simulate uncertainty in a system. Monte Carlo simulation is an example of a stochastic model, which will be used to assess future climate risks.