Architecture Design

Climate Change Analysis and Forecasting

Document Version Control

Date	Version	Description	Author
05/01/2022	1	Initial Doc	Sachin Sarkar
07/01/2022	2	Final Update	Sachin Sarkar

Contents

Table of Contents

Introduction

What is Architecture Design Document?

Any software needs the architectural design to represent the design of the software. IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” The software that is built for computer-based systems can exhibit one of these many architectures.

Each style will describe a system category that consists of:

- A set of components (eg: a database, computational modules) that will perform a function required by the system.
- The set of connectors will help in coordination, communication, and cooperation between the components.
- Conditions that how components can be integrated to form the system.
- Semantic models help the designer to understand the overall properties of the system.

Scope

Architecture Design Document (ADD) is an architectural design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the design principles may be defined during requirement analysis and then refined during architectural design work.

Product Perspective :

This Climate Change Analysis and Forecasting Application is a Web Application powered by Streamlit, Plotly and a Deep Learning based Time Series Forecasting Model to analyze, visualize and forecast average temperatures per month.

Problem Statement :

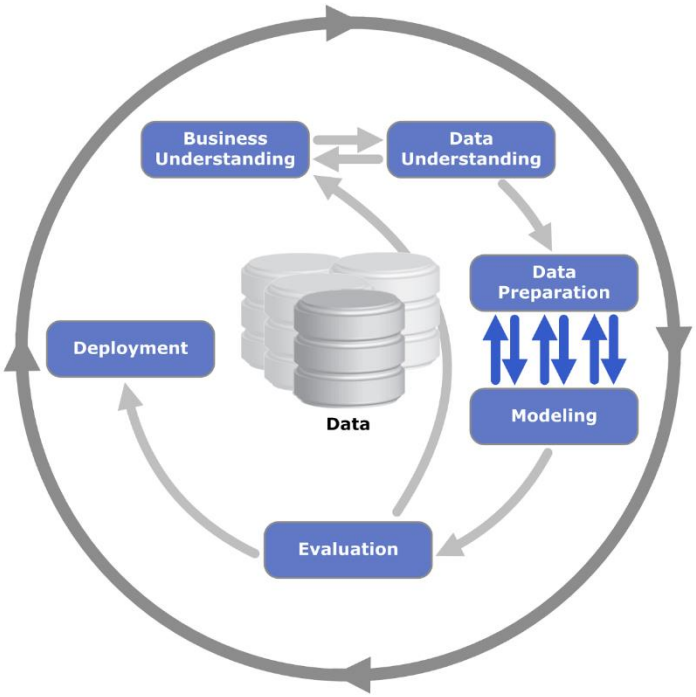
Will analyze the change in temperatures across globe from the 17th century till now and build a multivariate deep learning-based time series model to forecast the U.S. Average temperature. Predictive models attempt at forecasting future value based on historical data. The main objective here is -

1. Analyse the changes in climate across the globe.
2. Alert if any unusual climate change happen.
3. Maintain a database to store each and every data.

Proposed Solution :

The solution here proposed is a Multi Paged Web Application through which user can input or set Country, State, Time or Year Range and based on that data (may historical or future) of average temperatures per month, statistical description like mean, median, max, min, count etc. This data also downloadable on a button click. An interactive Trend Plot to view trend based on moving average , an interactive seasonal bar plot to view monthly average temperatures and an interactive autocorrelation plot to show correlation upto 100 previous data. Except this all user also can send a feedback or message to development team.

Architecture



Data Used :

For this project we used 3 csv file and 1 json. This are –

1) **GlobalTemperatures.csv :** For global average land temperatures.
2) **GlobalLandTemperaturesByCountry.csv :** For average temperatures group by countries.
3) **GlobalLandTemperaturesByState.csv :** For average temperatures group by states.
4) **Countries.geo.json :** Fo geographical locations of countries.

Data used in this project already discussed before. This data was collected from http://berkeleyearth.org/data/

Exploratory Data Analysis :

Analyze the change in temperatures across globe from the 17th century till now.

This Exploratory Data Analysis contains,

1) Load and Show Dataset
2) Missing Values Imputation
3) Lag Plot and Analysis
4) ACF or Autocorrelation plots and Analysis
5) Trend Plot and Analysis
6) Seasonal Plot and Analysis
7) Chloropleth Map of Average Temperature by Countries and Analysis

Data Modelling :

Select, train and Validate a best Time Series Model to forecast Average Temperatures per month for next years.

This part contains,

1) Import All Required Libraries
2) Load and Show Dataset
3) Missing Values Imputation
4) Data Stationarity Check
5) Time Series Modeling : Neural Prophet
7) Preparing dataset for NeuralProphet Training
8) Spliting Dataset for training and validation
9) Model Training
10) Model Training History and Visualization
11) Model performence analysis on Validation Data
12) Demo forecasting for next 2 years

Data Stationarity Check

Strategy : Augmented Dickey Fuller test (ADF Test)

Time Series Modeling : Neural Prophet

What is Neural Prophet ?

NeuralProphet is an upgraded version of Prophet that is built using PyTorch and uses deep learning models such as AR-Net for time-series forecasting. The main benefit of using NeuralProphet is that it features a simple API inspired by Prophet, but gives you access to more sophisticated deep learning models for time-series forecasting.

Data Validation :

Testing model performance on validation data. Validation metrices used –

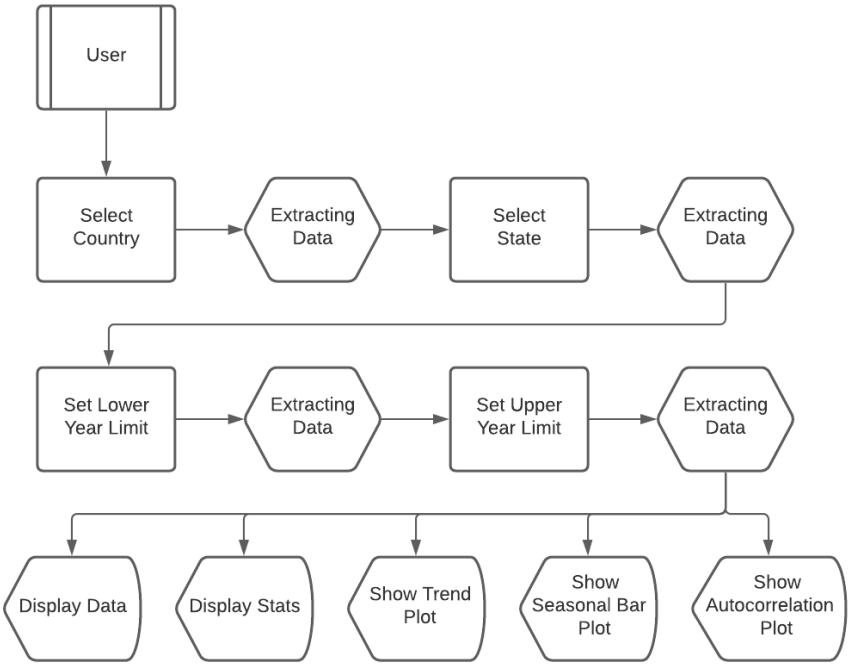
RMSE and R2 Score.

Web Application Development :

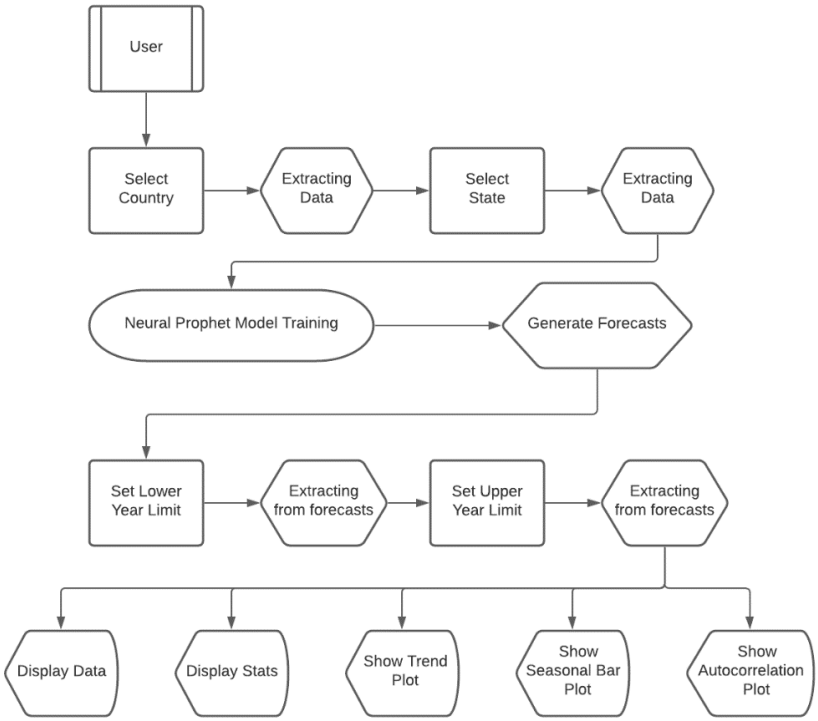
As per the proposed solution is a Multi Paged Web Application. To develop this frontend application, Streamlit is used powered by HTML / CSS. The whole coding for this frontend development is done on Spyder IDE.

Here 5 different web pages developed inside the application. This are –

- Documentation
- Historical Data and Plotting
- Future Data and Plotting
- Feedback Us
- About Us



Architecture for Historical Data and Plotting



Architecture for Historical Data and Plotting

Deployment :

Streamlit Cloud used for deployment this application.

When you work on an app in Streamlit Cloud—be it a new model, data analysis, or idea—you're just a few clicks away from securely sharing it and collaborating on it with your team.