Progress so far

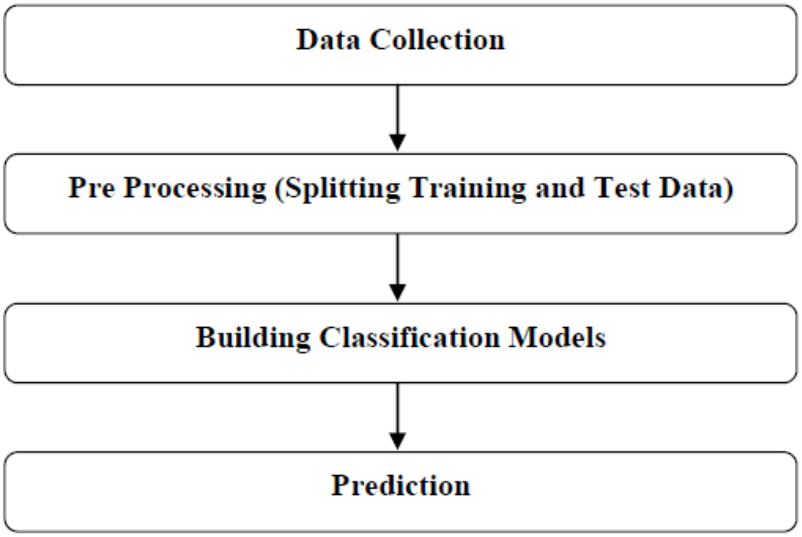


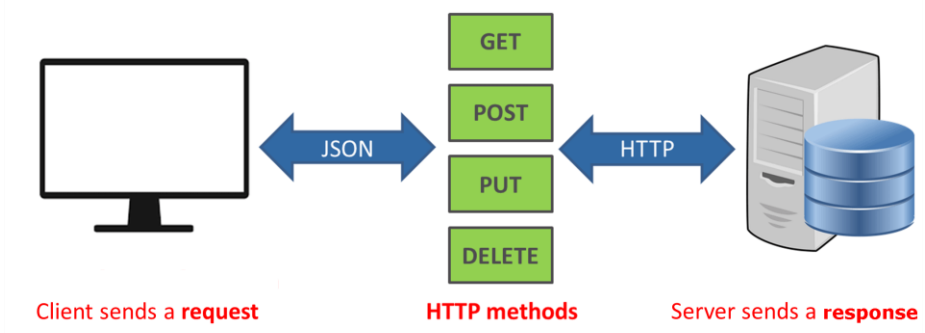
Figure 5.1: Process Flow Diagram

As per the flow of the project the very first step in a machine learning project is to collect the required data from a credible source.

For our project we have chosen data.gov.in which is an open government data provider who have many repositories related to many fields like GDP growth rate, air quality, foreign direct investment etc. For our use case we needed annual rainfall data of subdivisions of India.

Data Collection:

We are using RESTful API service to request data from the repository. This uses HTTP protocols to send and receive information from a web service. Request is made to the resource's Uniform Resource Identifier (URI). The GET request will send the request and return a response with the payload as the rainfall data in a JSON format. JSON stands for JavaScript Object Notation which is the most common and the standard response type used by the RESTful API services.



Here we are the client sending the request to the server. The URI of the server which we request the data is: https://api.data.gov.in/resource/8e0bd482-4aba-4d99-9cb9-ff124f6f1c2f?api-key={api_key}&format=json&offset=0

Here the api_key is the private key which is used to authenticate the user.

The returned payload includes a lot of metadata which includes title, created date, updated date, the organization that provided this data (Ministry of Earth Sciences, India Meteorological Department in our case) etc. The dataset includes rainfall data of various subdivisions of India from the year 1901 to 2017. These information is not helpful for us in the context of this project so we filter it out.

The data we received will have the following fields which we are going to focus on: subdivision, year, rainfall for each month of the year (in mm), annual rainfall for each subdivision and year (in mm).

The subdivisions included in the data are:

Andaman & Nicobar Islands, Arunachal Pradesh, Assam & Meghalaya, Bihar, Chhattisgarh, Coastal Andhra Pradesh, Coastal Karnataka, East Madhya Pradesh, East Rajasthan, East Uttar Pradesh, Gangetic West Bengal, Gujarat Region, Haryana Delhi & Chandigarh, Himachal Pradesh, Jammu & Kashmir, Jharkhand, Kerala, Konkan & Goa, Lakshadweep, Madhya Maharashtra, Matathwada, Naga Mani Mizo Tripura, North Interior Karnataka, Orissa, Punjab, Rayalseema, Saurashtra & Kutch, South Interior Karnataka, Sub Himalayan West Bengal & Sikkim, Tamil Nadu, Telangana, Uttarakhand, Vidarbha, West Madhya Pradesh, West Rajasthan, West Uttar Pradesh.

Pre processing:

Raw data received cannot be directly used by the machine learning models because they may contain many anomalies which will affect the performance, accuracy and the robustness of the models. So we need to clean the data. We also need to convert the data from JSON format to CSV so that pandas can use the data effectively in the data frames that we it's going to create. We only use the subdivision, year, rainfall for each month of the year (in mm), annual rainfall for each subdivision and year (in mm) when extracting data from the JSON file to CSV file.

The CSV file is then imported to the project using Numpy. Pandas is used to remove the anomalies from the data. The parameters that we considered for cleaning up the data is, any missing field values, any duplicate values, any null values.

We later split the data into training and testing sets which are in the ratio of 80:20 where 80% of the data is used for training the model and 20% of the data is used for testing. After this the data is cleaned and split we use algorithms to predict the probability of the occurrence of flood in a given subdivision. Time series data can be analysed by regression algorithms which we are using.

We use graphs to represent the data and give more context to the dataset. This will help the end user better understand what they are looking and to make sense of the data.

The algorithms then output their predictions for the given data with the accuracy (in percentage). The output is a boolean value which is true or false, where true being the chances of flooding based on the past rainfall data is high and false being the chances of flooding based on the past rainfall data is low. We then display these results to the user.

The limitation of this approach is that it is just one dimensional. Flood can occur because of many reasons, not just rainfall alone. So we cannot conclude that the results from this project is the accurate representation of the real world. Predicting weather is a very difficult job, even with all the variables and data we cannot predict the weather with high accuracy.

We will try to bring in other dimension to the project by including other datasets like river flow rate and discharge etc.