PREDICTIVE MODELLING OF GROUNDWATER FLUORIDE CONTAMINATION

USING MACHINE LEARNING MODELS

MINOR PROJECT – I

SUBMITTED BY –

HARSHIT SHARMA (9919103016)
SIDDHARTH SINGH (9919103029)
PRANAT JAIN (9919103017)
YASHA JAFRI (9919103009)

UNDER THE SUPERVISION OF –

DR. SHIKHA K MEHTA

DEPARTMENT OF CS/IT
JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

DECEMBER 2021

ACKNOWLEDGEMENT

We would like to place on record our deep sense of gratitude to Dr. Shikha K Mehta, Associate Professor, Jaypee Institute of Information Technology, India for her generous guidance, help and useful suggestions.

We express our sincere gratitude to Joel Podgorski, EAWAG Swiss Federal Institute of Aquatic Science and Technology, for his guidance and encouragement in the course of present work.

We also wish to extend our heartfelt gratitude to our batchmates for their insightful comments and constructive suggestions to improve the quality of this project work. Lastly, we would like to thank our parents to be the constant source of motivation which we needed during the course of this project.

Harshit Sharma

Harshit Sharma (9919103016)

Siddharth

Siddharth Singh (9919103029)

Pranat. Jain

Pranat Jain (9919103017)

Yasha Jafri.

Yasha Jafri (9919103009)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and beliefs, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: Noida, Uttar Pradesh

Date: 1st December, 2021

Harshit Sharma
9919103016

Siddharth Singh
9919103029

Pranat Jain
9919103017

Yasha Jafri
9919103009

CERTIFICATE

This is to certify that the work titled “Predictive Modelling of Groundwater Fluoride Contamination using Machine Learning Models” submitted by Harshit Sharma, Siddharth Singh, Pranat Jain and Yasha Jafri of B.Tech of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Dr. Shikha K Mehta

(Associate Professor, Jaypee Institute of Information Technology)

1st December, 2021

ABSTRACT

India is a country which heavily relies on Groundwater for its Development and Economic growth. But, with the ever-increasing rate of Population, the daily requirement of Groundwater is also rising every day. Among all such challenges, purity of Groundwater becomes a concern.

Fluoride, among many other contaminants, is one of the most harmful Groundwater Contaminants found in every habitable continent. WHO has specified a concentration range of 0.5-1.0 mg/L beneficial for human health, with an exception of up to 1.5 mg/L.

This project studies the various factors responsible for the Fluoride Contamination in Groundwater and tries to predict the concentrations of Fluoride. We have used various Machine Learning Models like Random Forest Regressor, Random Forest Classifier, XGBoost Regressor and XGBoost Classifier.

The accuracy achieved from these models range from 84.5% to 85%, where Random Forest Model produced the best results among all models. For Regression, XGBoost Model has better accuracy, whereas for Classification, Random Forest Model has better accuracy.

The Models can be used for mitigation of measures required to deal with in areas of High Fluoride Contamination in Groundwater.

TABLE OF CONTENTS

LIST OF TABLES

INTRODUCTION

India is a country which is home to different landforms. It is the home to World's highest Rainfall region (Mawsynram). It is the home to the dry deserts of Thar Desert. It is home to the huge Mountains of Himalayas and even one of the longest shorelines in a country.

India's land is mostly covered by Arid and Semi-arid regions. These regions have very less rainfall. Thus, the country relies heavily on groundwater to support the growing population and rising economy. According to various researches, it is estimated that Groundwater amounts to 60% of the Irrigation water and 85% of the Household water usage. Additionally, satellite observations have indicated the Overexploitation of Groundwater.

Under these circumstances, it is very important to maintain the quality of Groundwater. Groundwater is free of microbial contaminants, but the contaminants like Fluoride, Arsenic, Calcium, etc. are of major concern. Out of all these, Fluoride has been considered important for human health, but also dangerous if consumed in higher concentrations.

WHO has advised concentration levels of 0.5-1.0 mg/L as safe, with exceptions up to 1.5 mg/L. India maintains a level of 1.5 mg/L, but many regions in India, as suggested by studies in various parts of India show high levels of Fluoride Contamination levels. Some of these regions have even reported an average level of above 3.0 mg/L. High intake of Fluoride for longer periods can lead to many health problems like Dental and Skeletal Fluorosis. India wants to reach an average level of 1.0 mg/L considering the amount of water the populations have to consume due to the hot & humid climate of the country.

BACKGROUND STUDY

HOW DOES FLUORIDE REACH GROUNDWATER?

The release of fluoride to groundwater is dependent on chemical and physical processes that take place between the groundwater and its geological environment. Fluorite (CaF_2) is the predominant mineral that controls the dissolved fluoride concentration in the groundwater. Thus fluoride-rich groundwaters are often associated with low calcium concentrations. This is associated with rocks with low calcium content, or high pH conditions where sodium bicarbonate dominates the groundwater composition. A part from the groundwater chemistry, hydrological properties (e.g., residence time) as well as climatic conditions (e.g., evapotranspiration, precipitation) and soil conditions (e.g., pH, soil type) have an influence on fluoride concentration. Hence, the spatial and temporal heterogeneities of fluoride concentrations in groundwater are particularly large.

The alkaline conditions produce a conductive environment for dissolving F-- bearing minerals more effectively, causing an accumulation of F- in groundwater. This is supported by a positive correlation of F- with pH as well as with HCO3 - + CO3 2- Further support is provided by a positive chloralkaline index, an oversaturation with respect to CaCO3, and a negative correlation between the well depth and indicating effects of ion exchange, evaporation, and solubility and/or leaching, respectively. All these factors regulate the concentration of F- in the groundwater. On the other hand, a positive correlation between K and F and also between SO_4 and F suggest the effects of human land use activities which provide the additional concentration of F in the groundwater.

$$CO_2 + H_2O \rightleftharpoons H_2CO_3.$$

$$H_2CO_3 \rightleftharpoons H^+ + HCO_3^-.$$

$$HCO_3^- \rightleftharpoons H^+ + CO_3^{2-}.$$

$$CaF_2 + HCO_3^- \rightleftharpoons CaCO_3 + 2F^- + H^+$$

STATISTICAL MODELING OF GLOBAL GEOGENIC FLUORIDE CONTAMINATION IN GROUNDWATERS

In this study, it provides a global probability map that indicates the risk of fluoride contamination in groundwaters. It is a global overview of potentially fluoride-rich groundwaters by modelling fluoride concentration. A large database of worldwide fluoride concentrations as well as available information on related environmental factors such as soil properties, geological settings, and climatic and topographical information on a global scale have all been used in the model. The modelling approach combines geochemical knowledge with statistical methods to devise a rule-based statistical procedure, which divides the world into 8 different "process regions". For each region a separate predictive model was constructed. The end result was a global probability map of fluoride concentration in the groundwater.

A statistical model to predict fluoride concentration was developed for each region by splitting the fluoride data into two subsets using stratified random sampling. The subset with 80% of the data was used for model development and the remaining was set aside for model validation. Stepwise regression was then used to identify the significant variables for model development. This was followed by the application of the Adaptive Neuro-Fuzzy Inference System (ANFIS) to identify the best predictive model. Mean error (ME), mean square error (MSE), and R^2 statistics were used to evaluate the models. ANFIS models were then linked to a Latin hypercube sampling method to propagate the uncertainty of the parameters on the results. These parameters included: centres of different fuzzy classes of variables and the coefficients of the multiple linear regression model. This allowed the prediction of the cumulative distribution function and, subsequently, the determination of the probability of fluoride concentrations exceeding a specific threshold (1.5 mgL^{-1}) for each pixel.

GIS (GLOBAL INFORMATION SCHEMA)

A geographic information system (GIS) is a conceptualized framework that provides the ability to capture and analyse spatial and geographic data. GIS applications are computer-based tools that allow the user to create interactive queries, store and edit spatial and non-spatial data, analyse spatial information output, and visually share the results of these operations by presenting them as maps.

A geographic information system (GIS) is a system that creates, manages, analyses, and maps all types of data. GIS connects data to a map, integrating location data (where things are) with all types of descriptive information (what things are like there). This provides a foundation for mapping and analysis that is used in science and almost every industry. GIS helps users understand patterns, relationships, and geographic context. The benefits include improved communication and efficiency as well as better management and decision making.

Geographic information systems are utilized in multiple technologies, processes, techniques and methods. They are attached to various operations and numerous applications, that relate to: engineering, planning, management, transport/logistics, insurance, telecommunications, and business. For this reason, GIS and location intelligence applications are at the foundation of location-enabled services, that rely on geographic analysis and visualization. GIS provides the capability to relate previously unrelated information, through the use of location as the key-variable.

RANDOM FOREST ALGORITHM

Random forest is a kind of Machine Learning Algorithm that is used in Classification and Regression problems. Random forest algorithm takes majority number of votes by building decision trees on different samples. Random forest can handle dataset which contain continuous and categorical variables in the case of regression and classification respectively.

Random Forests are a multipurpose tool, applicable to both regression and classification problems, including multiclass classification. They give an internal estimate of generalization error so cross validation is unnecessary. They can be tuned, but often work quite well with default tuning parameters. Variable importance measures are available, which can be used for variable selection. Random Forests produce proximities, which can be used to impute missing values. Proximities can also provide a wealth of information by enabling novel visualizations of the data. Random Forests have been successfully used for a wide variety of applications and enjoy considerable popularity in several disciplines.

Ensemble means combining multiple models together. Thus, with the help of ensemble we can easily make predictions with the help of collection of models or not just from a single one. Ensemble uses two types of models:

1. Bagging - Bagging makes a different training subset from a sample data using replacement and our final output is totally based upon majority voting or averaging. For example, Random Forest algorithm.
2. Boosting - Boosting refers to a family of algorithms which converts weak learner to strong learners by creating sequential models. For example, ADA BOOST, XG BOOST.

Features of Random Forest Algorithm are –

1. Diversity - All attributes or variables are not considered while making an individual tree, each tree is different.
2. Immune to dimensionality - Since each tree does not have every feature, space is reduced.
3. Parallelization - Each tree is created independently out of different data and attributes.
4. Stability - The output is based on majority voting or averaging and not some other method.

XGBOOST ALGORITHM

XGBoost stands for "Extreme Gradient Boosting". XGBoost is an advanced gradient library designed to be highly functional, flexible and portable. It uses machine learning algorithms under the Gradient Boosting framework. It provides consistent tree growth to solve many data science problems in a fast and well-thought-out way.

Boosting is an integrated learning method for building a strong separator from a few weak class actors in a series. Developing algorithms play an important role in dealing with biased trade and diversity. Unlike bagging algorithms, which control only the top variations in the model, boosting controls both aspects (bias and variability) and is considered continuous.

REQUIREMENT ANALYSIS

The requirements for this project are:

1) Python 3.x
   Python is a very varied tool used for Predictive Analysis, due to the availability of various modules which ease the implementation of the required analyses.

2) Python Modules –
   i) NumPy – to work with large arrays of data
   ii) Pandas – to create and manage data frames containing data
   iii) Scikit-learn – provides tools for ML Models and metrics related to them
   iv) RasterIo – extracting data from Raster files like .tiff images

3) Microsoft Excel
   It is a very useful software which provides for mechanism and tools o store data. Unlike SQL or similar DBMS, it is easier to operate upon.

4) Google Colab (For Implementation purposes)
   It is an Online tool to work on our projects without the need to install any modules on our own environment. It has several advantages over the traditional ways to create a Virtual Environment, where several users ran into problems.

DETAILED DESIGN

1) The Predictor Variables' Dataset has to be collected from the various sources, cited in different research papers. These datasets are available in different formats. They have to be collected in the 30 arc seconds (30", or 1-km resolution) format.

2) Extract the data out of these files and organize them in an Excel file. This part of the project can be done using the RasterIo and Pandas modules of Python. As the data is to be collected in Excel files, thus Microsoft Excel becomes important to the project.

3) After collecting and organizing the data, implement Machine Learning Models on the data and make predictions. The ML Models to be implemented are Random Forest and XGBoost. These both models provide Regressor as well as Classifier Models.

4) Calculate the Prediction Accuracy of all these models and compare them with each other. The Scikit-learn module provides for built-in function to test the accuracy of these models.

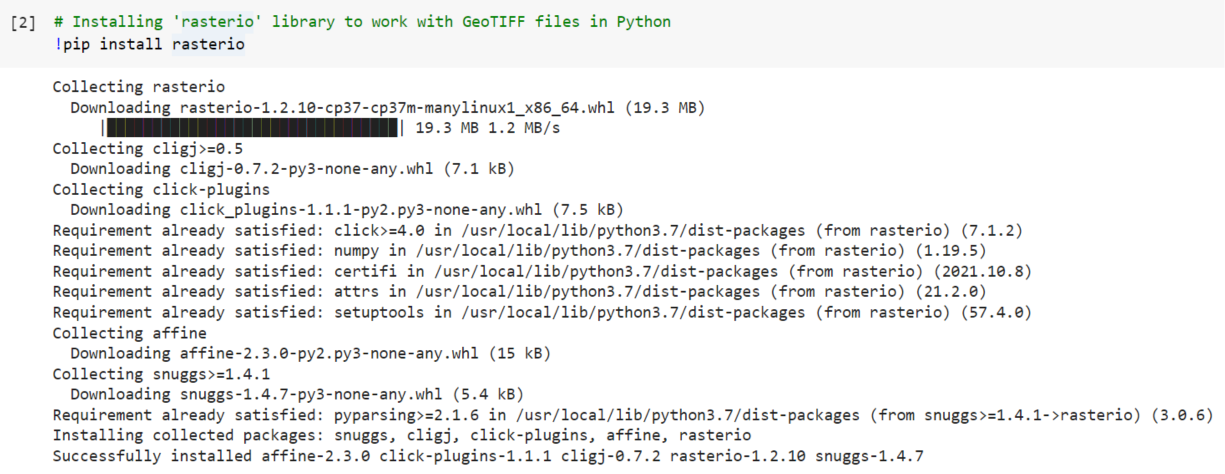
IMPLEMENTATION

PREDICTOR VARIABLES' DATASET

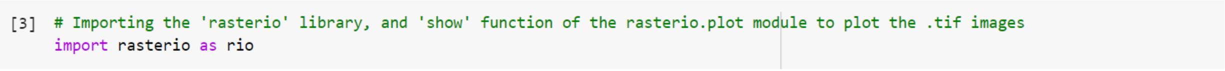
According the Research Papers studied, there are several variables that can affect the prediction of Fluoride levels in Groundwater like Evapotranspiration, Temperature, Soil pH, Rock Formation, etc. The most prominent of these variables are – Potential & Actual Evapotranspiration, Aridity, Calcisols, Gypsiols, Rock Formation, Slope, Soil pH, Precipitation, Clay Fraction, Sand & Silt Fraction, and Cropland.

Out of the prominent variables, the Variables considered under this project are – Actual Evapotranspiration, Potential Evapotranspiration, Aridity, Soil pH, Slope (0-60%) and Precipitation. Other variables were disregarded (either due to low Computation power of the System or unavailability of correct resolution which could not be managed to be resized or changed).

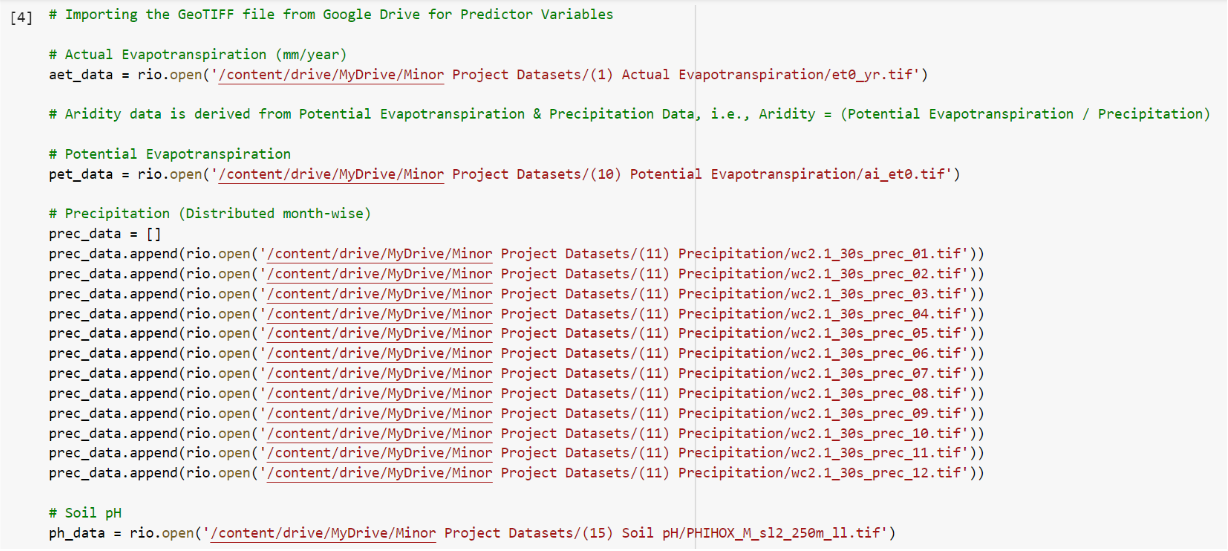
The RasterIo implementation of the Dataset Collection & Extraction is given below:



```
[2] # Installing 'rasterio' library to work with GeoTIFF files in Python
    !pip install rasterio

    Collecting rasterio
      Downloading rasterio-1.2.10-cp37-cp37m-manylinux1_x86_64.whl (19.3 MB)
         |████████████████████████████████| 19.3 MB 1.2 MB/s
    Collecting cligj>=0.5
      Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
    Collecting click-plugins
      Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
    Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.7/dist-packages (from rasterio) (7.1.2)
    Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from rasterio) (1.19.5)
    Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from rasterio) (2021.10.8)
    Requirement already satisfied: attrs in /usr/local/lib/python3.7/dist-packages (from rasterio) (21.2.0)
    Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from rasterio) (57.4.0)
    Collecting affine
      Downloading affine-2.3.0-py2.py3-none-any.whl (15 kB)
    Collecting snuggs>=1.4.1
      Downloading snuggs-1.4.7-py3-none-any.whl (5.4 kB)
    Requirement already satisfied: pyparsing>=2.1.6 in /usr/local/lib/python3.7/dist-packages (from snuggs>=1.4.1->rasterio) (3.0.6)
    Installing collected packages: snuggs, cligj, click-plugins, affine, rasterio
    Successfully installed affine-2.3.0 click-plugins-1.1.1 cligj-0.7.2 rasterio-1.2.10 snuggs-1.4.7
```

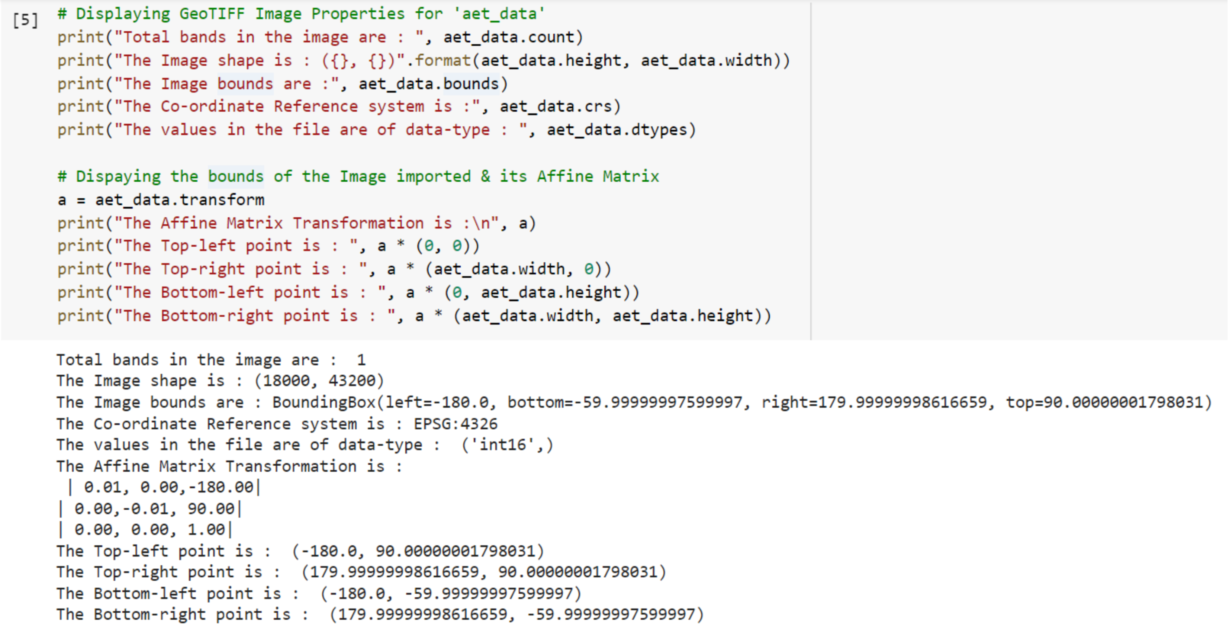


```
[3] # Importing the 'rasterio' library, and 'show' function of the rasterio.plot module to plot the .tif images
    import rasterio as rio
```



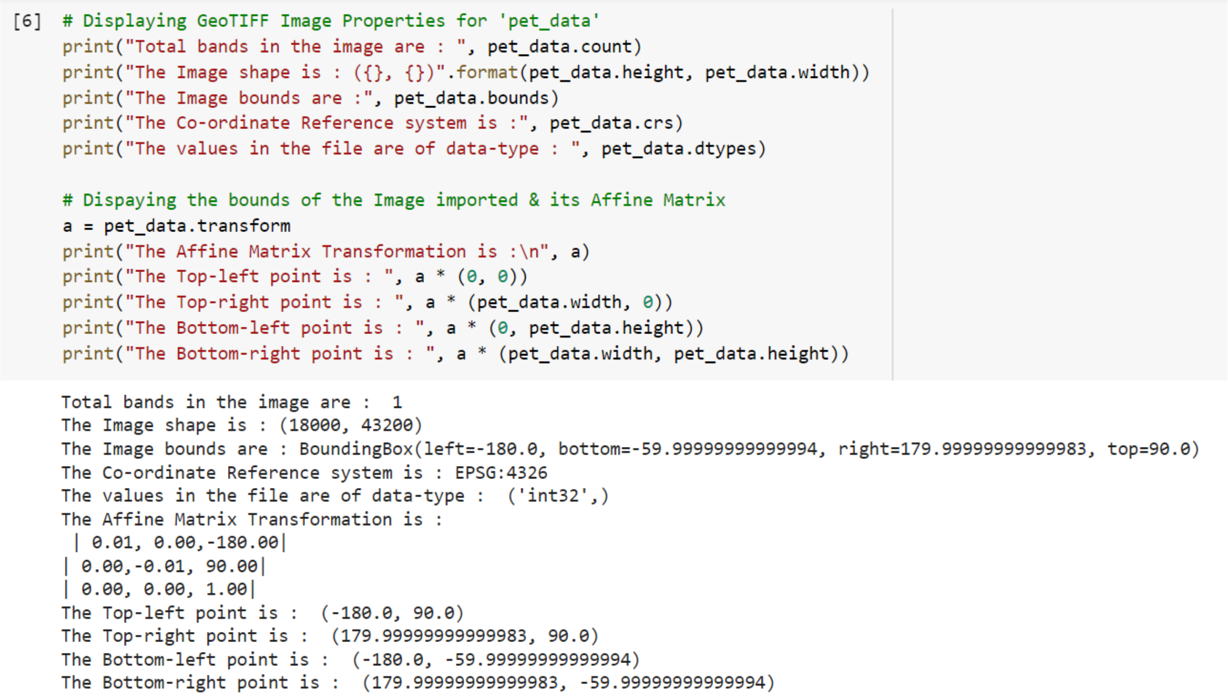
```
[4] # Importing the GeoTIFF file from Google Drive for Predictor Variables

    # Actual Evapotranspiration (mm/year)
    aet_data = rio.open('/content/drive/MyDrive/Minor Project Datasets/(1) Actual Evapotranspiration/et0_yr.tif')

    # Aridity data is derived from Potential Evapotranspiration & Precipitation Data, i.e., Aridity = (Potential Evapotranspiration / Precipitation)

    # Potential Evapotranspiration
    pet_data = rio.open('/content/drive/MyDrive/Minor Project Datasets/(10) Potential Evapotranspiration/ai_et0.tif')

    # Precipitation (Distributed month-wise)
    prec_data = []
    prec_data.append(rio.open('/content/drive/MyDrive/Minor Project Datasets/(11) Precipitation/wc2.1_30s_prec_01.tif'))
    prec_data.append(rio.open('/content/drive/MyDrive/Minor Project Datasets/(11) Precipitation/wc2.1_30s_prec_02.tif'))
    prec_data.append(rio.open('/content/drive/MyDrive/Minor Project Datasets/(11) Precipitation/wc2.1_30s_prec_03.tif'))
    prec_data.append(rio.open('/content/drive/MyDrive/Minor Project Datasets/(11) Precipitation/wc2.1_30s_prec_04.tif'))
    prec_data.append(rio.open('/content/drive/MyDrive/Minor Project Datasets/(11) Precipitation/wc2.1_30s_prec_05.tif'))
    prec_data.append(rio.open('/content/drive/MyDrive/Minor Project Datasets/(11) Precipitation/wc2.1_30s_prec_06.tif'))
    prec_data.append(rio.open('/content/drive/MyDrive/Minor Project Datasets/(11) Precipitation/wc2.1_30s_prec_07.tif'))
    prec_data.append(rio.open('/content/drive/MyDrive/Minor Project Datasets/(11) Precipitation/wc2.1_30s_prec_08.tif'))
    prec_data.append(rio.open('/content/drive/MyDrive/Minor Project Datasets/(11) Precipitation/wc2.1_30s_prec_09.tif'))
    prec_data.append(rio.open('/content/drive/MyDrive/Minor Project Datasets/(11) Precipitation/wc2.1_30s_prec_10.tif'))
    prec_data.append(rio.open('/content/drive/MyDrive/Minor Project Datasets/(11) Precipitation/wc2.1_30s_prec_11.tif'))
    prec_data.append(rio.open('/content/drive/MyDrive/Minor Project Datasets/(11) Precipitation/wc2.1_30s_prec_12.tif'))

    # Soil pH
    ph_data = rio.open('/content/drive/MyDrive/Minor Project Datasets/(15) Soil pH/PHIHOX_M_sl2_250m_ll.tif')
```

The Properties of each and every Dataset were read so that appropriate operations can be applied on them in the future as follows:



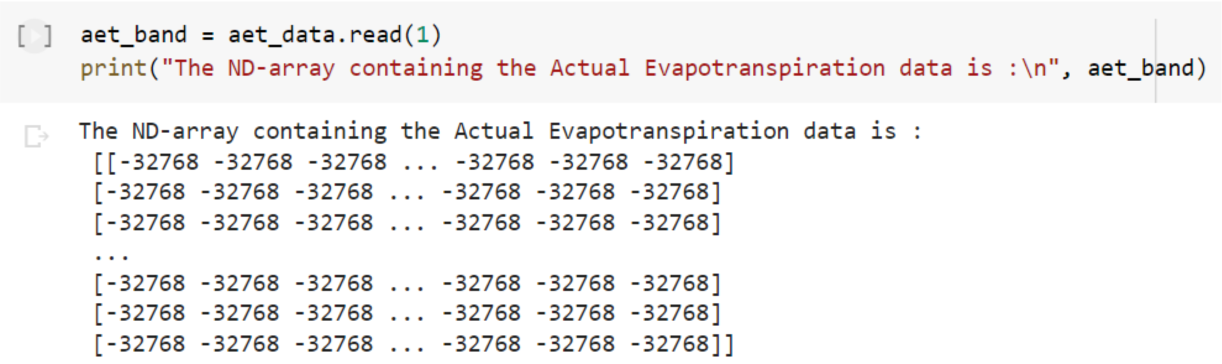
```
[5]  # Displaying GeoTIFF Image Properties for 'aet_data'
     print("Total bands in the image are : ", aet_data.count)
     print("The Image shape is : ({}, {})".format(aet_data.height, aet_data.width))
     print("The Image bounds are :", aet_data.bounds)
     print("The Co-ordinate Reference system is :", aet_data.crs)
     print("The values in the file are of data-type : ", aet_data.dtypes)

     # Dispaying the bounds of the Image imported & its Affine Matrix
     a = aet_data.transform
     print("The Affine Matrix Transformation is :\n", a)
     print("The Top-left point is : ", a * (0, 0))
     print("The Top-right point is : ", a * (aet_data.width, 0))
     print("The Bottom-left point is : ", a * (0, aet_data.height))
     print("The Bottom-right point is : ", a * (aet_data.width, aet_data.height))
```

```
Total bands in the image are :  1
The Image shape is : (18000, 43200)
The Image bounds are : BoundingBox(left=-180.0, bottom=-59.99999997599997, right=179.99999998616659, top=90.00000001798031)
The Co-ordinate Reference system is : EPSG:4326
The values in the file are of data-type :  ('int16',)
The Affine Matrix Transformation is :
 | 0.01, 0.00,-180.00|
| 0.00,-0.01, 90.00|
| 0.00, 0.00, 1.00|
The Top-left point is :  (-180.0, 90.00000001798031)
The Top-right point is :  (179.99999998616659, 90.00000001798031)
The Bottom-left point is :  (-180.0, -59.99999997599997)
The Bottom-right point is :  (179.99999998616659, -59.99999997599997)
```



```
[6]  # Displaying GeoTIFF Image Properties for 'pet_data'
     print("Total bands in the image are : ", pet_data.count)
     print("The Image shape is : ({}, {})".format(pet_data.height, pet_data.width))
     print("The Image bounds are :", pet_data.bounds)
     print("The Co-ordinate Reference system is :", pet_data.crs)
     print("The values in the file are of data-type : ", pet_data.dtypes)

     # Dispaying the bounds of the Image imported & its Affine Matrix
     a = pet_data.transform
     print("The Affine Matrix Transformation is :\n", a)
     print("The Top-left point is : ", a * (0, 0))
     print("The Top-right point is : ", a * (pet_data.width, 0))
     print("The Bottom-left point is : ", a * (0, pet_data.height))
     print("The Bottom-right point is : ", a * (pet_data.width, pet_data.height))
```

```
Total bands in the image are :  1
The Image shape is : (18000, 43200)
The Image bounds are : BoundingBox(left=-180.0, bottom=-59.99999999999994, right=179.99999999999983, top=90.0)
The Co-ordinate Reference system is : EPSG:4326
The values in the file are of data-type :  ('int32',)
The Affine Matrix Transformation is :
 | 0.01, 0.00,-180.00|
| 0.00,-0.01, 90.00|
| 0.00, 0.00, 1.00|
The Top-left point is :  (-180.0, 90.0)
The Top-right point is :  (179.99999999999983, 90.0)
The Bottom-left point is :  (-180.0, -59.99999999999994)
The Bottom-right point is :  (179.99999999999983, -59.99999999999994)
```



```
[ ]  aet_band = aet_data.read(1)
     print("The ND-array containing the Actual Evapotranspiration data is :\n", aet_band)
```

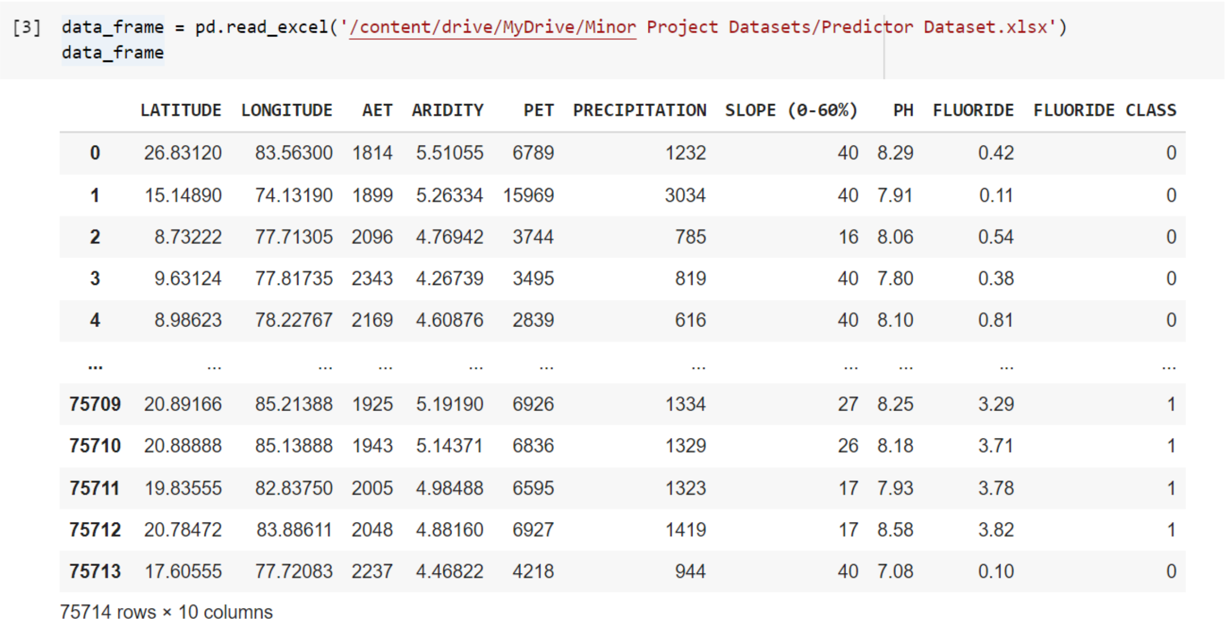
```
The ND-array containing the Actual Evapotranspiration data is :
 [[-32768 -32768 -32768 ... -32768 -32768 -32768]
 [-32768 -32768 -32768 ... -32768 -32768 -32768]
 [-32768 -32768 -32768 ... -32768 -32768 -32768]
 ...
 [-32768 -32768 -32768 ... -32768 -32768 -32768]
 [-32768 -32768 -32768 ... -32768 -32768 -32768]
 [-32768 -32768 -32768 ... -32768 -32768 -32768]]
```

Similarly, all the data had been collected and stored in Excel Files, where further Incomplete Data was filtered out for better predictions using Excel Functions.

DATA PROCESSING, PREPARATION & LOADING

One of the major issues faced during Data Preprocessing in Excel was the conversion of Geographical Coordinates so that it could be fed to the Model. The Latitude and Longitude values were given as xx°yy'zz". This form had to be converted to xx.yyyyy, i.e., degrees to decimals. Excel operations like 'Convert Text to Table' were used. Mathematical operations like xx + yy/60 + zz/3600 were used to finally convert the coordinate values in degrees to decimal.

After the Dataset was prepared, it had to be tested by applying Machine Learning Models. The Machine Learning Models used in this project are – Random Forest Algorithm and XGBoost Algorithm. These have been discussed earlier in this project. An Instance of the Dataset collected and Organised is given below:

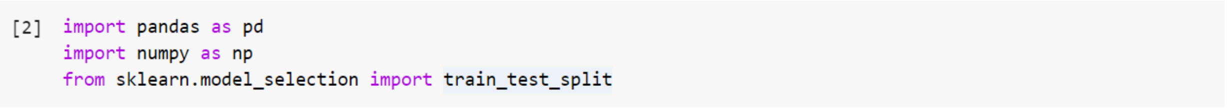


```
[3] data_frame = pd.read_excel('/content/drive/MyDrive/Minor Project Datasets/Predictor Dataset.xlsx')
    data_frame
```

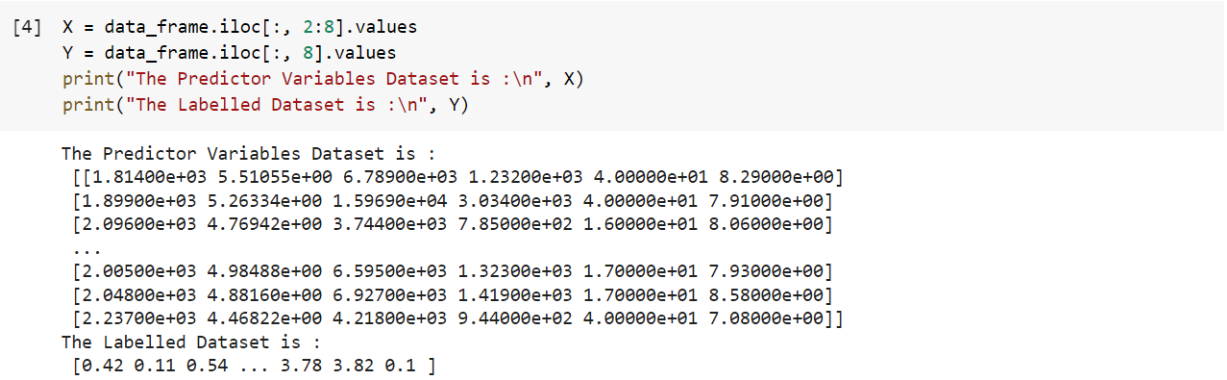
	LATITUDE	LONGITUDE	AET	ARIDITY	PET	PRECIPITATION	SLOPE (0-60%)	PH	FLUORIDE	FLUORIDE CLASS
0	26.83120	83.56300	1814	5.51055	6789	1232	40	8.29	0.42	0
1	15.14890	74.13190	1899	5.26334	15969	3034	40	7.91	0.11	0
2	8.73222	77.71305	2096	4.76942	3744	785	16	8.06	0.54	0
3	9.63124	77.81735	2343	4.26739	3495	819	40	7.80	0.38	0
4	8.98623	78.22767	2169	4.60876	2839	616	40	8.10	0.81	0
...	...	...	...	...	...	...	...	...	...	...
75709	20.89166	85.21388	1925	5.19190	6926	1334	27	8.25	3.29	1
75710	20.88888	85.13888	1943	5.14371	6836	1329	26	8.18	3.71	1
75711	19.83555	82.83750	2005	4.98488	6595	1323	17	7.93	3.78	1
75712	20.78472	83.88611	2048	4.88160	6927	1419	17	8.58	3.82	1
75713	17.60555	77.72083	2237	4.46822	4218	944	40	7.08	0.10	0

75714 rows × 10 columns

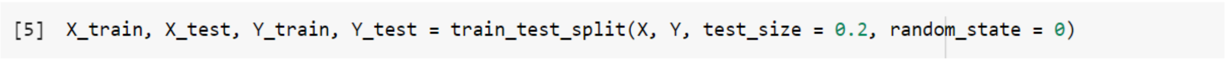
In order to implement the model, we need to divide our data into Training and Testing Data. This is the most fundamental step before training a model in Supervised Learning. The implementation of the same for Regression is given below:



```
[2] import pandas as pd
    import numpy as np
    from sklearn.model_selection import train_test_split
```

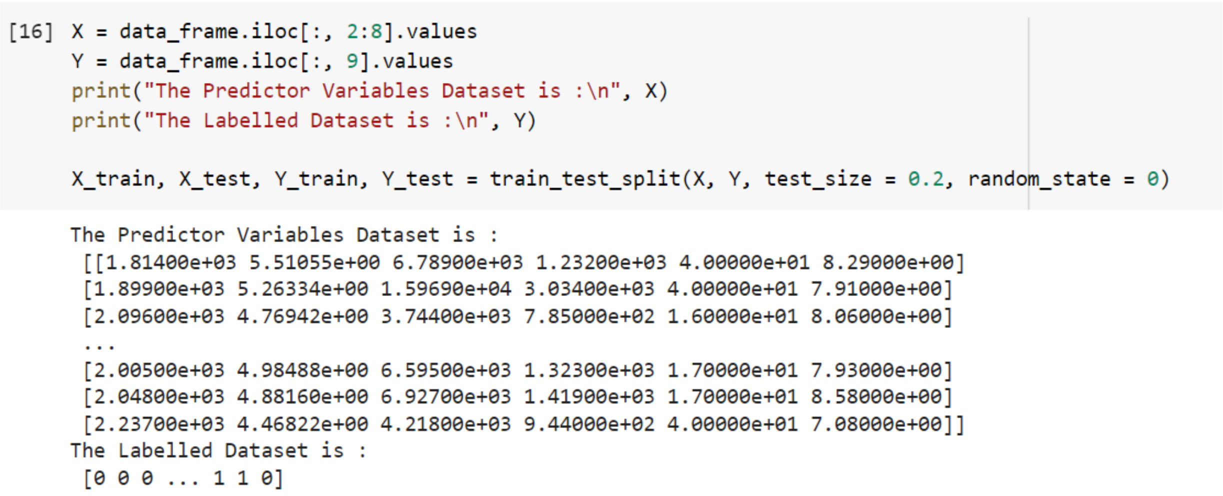


```
[4] X = data_frame.iloc[:, 2:8].values
    Y = data_frame.iloc[:, 8].values
    print("The Predictor Variables Dataset is :\n", X)
    print("The Labelled Dataset is :\n", Y)

    The Predictor Variables Dataset is :
     [[1.81400e+03 5.51055e+00 6.78900e+03 1.23200e+03 4.00000e+01 8.29000e+00]
     [1.89900e+03 5.26334e+00 1.59690e+04 3.03400e+03 4.00000e+01 7.91000e+00]
     [2.09600e+03 4.76942e+00 3.74400e+03 7.85000e+02 1.60000e+01 8.06000e+00]
     ...
     [2.00500e+03 4.98488e+00 6.59500e+03 1.32300e+03 1.70000e+01 7.93000e+00]
     [2.04800e+03 4.88160e+00 6.92700e+03 1.41900e+03 1.70000e+01 8.58000e+00]
     [2.23700e+03 4.46822e+00 4.21800e+03 9.44000e+02 4.00000e+01 7.08000e+00]]
    The Labelled Dataset is :
     [0.42 0.11 0.54 ... 3.78 3.82 0.1 ]
```



```
[5] X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 0)
```

The implementation of the same for Classification is given below:

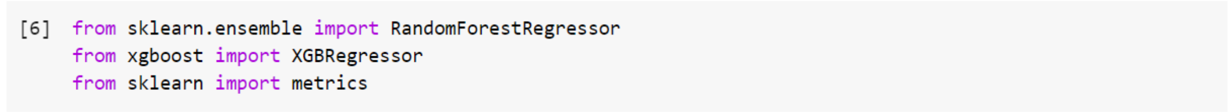


```
[16] X = data_frame.iloc[:, 2:8].values
     Y = data_frame.iloc[:, 9].values
     print("The Predictor Variables Dataset is :\n", X)
     print("The Labelled Dataset is :\n", Y)

     X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 0)
```

```
The Predictor Variables Dataset is :
 [[1.81400e+03 5.51055e+00 6.78900e+03 1.23200e+03 4.00000e+01 8.29000e+00]
 [1.89900e+03 5.26334e+00 1.59690e+04 3.03400e+03 4.00000e+01 7.91000e+00]
 [2.09600e+03 4.76942e+00 3.74400e+03 7.85000e+02 1.60000e+01 8.06000e+00]
 ...
 [2.00500e+03 4.98488e+00 6.59500e+03 1.32300e+03 1.70000e+01 7.93000e+00]
 [2.04800e+03 4.88160e+00 6.92700e+03 1.41900e+03 1.70000e+01 8.58000e+00]
 [2.23700e+03 4.46822e+00 4.21800e+03 9.44000e+02 4.00000e+01 7.08000e+00]]
The Labelled Dataset is :
 [0 0 0 ... 1 1 0]
```

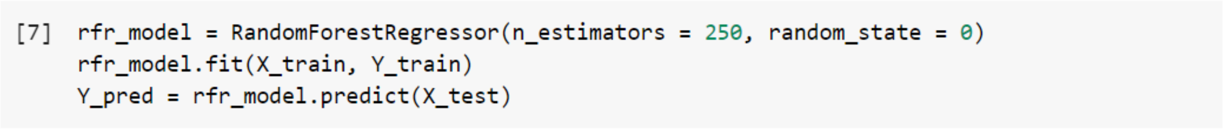
RANDOM FOREST & XGBOOST REGRESSION

The Random Forest & XGBoost Regression Implementations are depicted below:

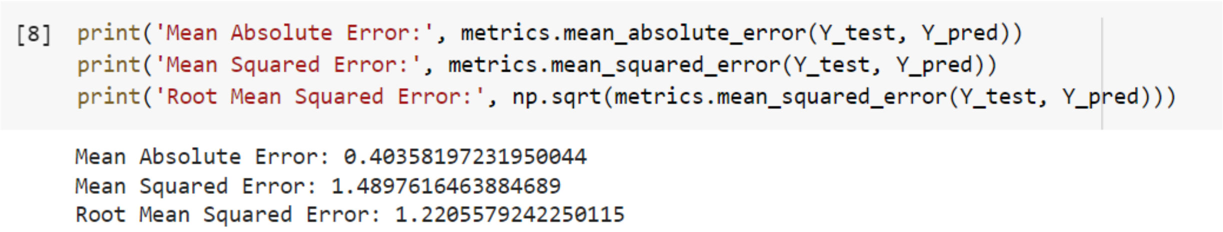


```
[6]  from sklearn.ensemble import RandomForestRegressor
     from xgboost import XGBRegressor
     from sklearn import metrics
```

Random Forest Model with 250 Decision Trees:



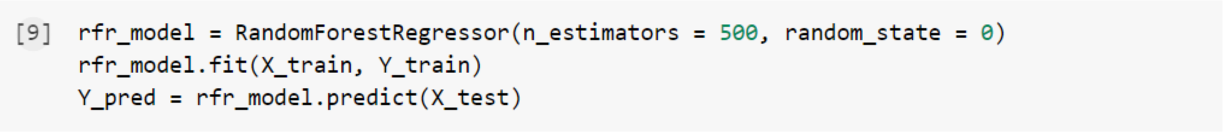
```
[7]  rfr_model = RandomForestRegressor(n_estimators = 250, random_state = 0)
     rfr_model.fit(X_train, Y_train)
     Y_pred = rfr_model.predict(X_test)
```



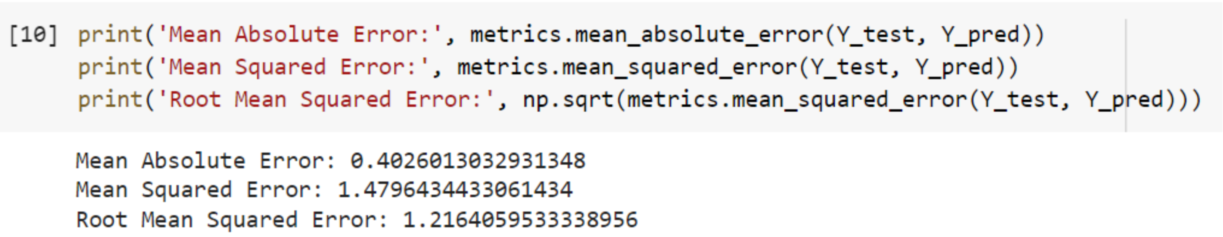
```
[8]  print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, Y_pred))
     print('Mean Squared Error:', metrics.mean_squared_error(Y_test, Y_pred))
     print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))
```

```
Mean Absolute Error: 0.40358197231950044
Mean Squared Error: 1.4897616463884689
Root Mean Squared Error: 1.2205579242250115
```

Random Forest Model with 500 Decision Trees:



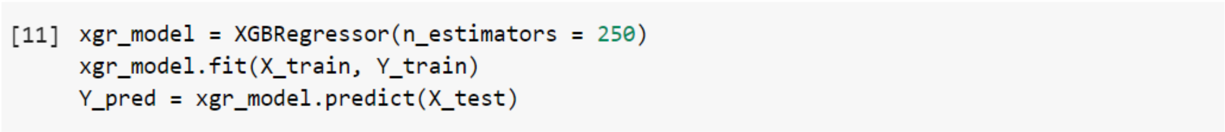
```
[9]  rfr_model = RandomForestRegressor(n_estimators = 500, random_state = 0)
     rfr_model.fit(X_train, Y_train)
     Y_pred = rfr_model.predict(X_test)
```



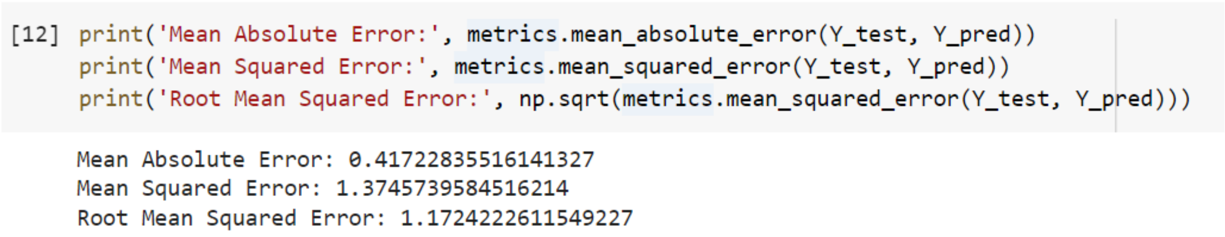
```
[10] print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, Y_pred))
     print('Mean Squared Error:', metrics.mean_squared_error(Y_test, Y_pred))
     print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))
```

```
Mean Absolute Error: 0.4026013032931348
Mean Squared Error: 1.4796434433061434
Root Mean Squared Error: 1.2164059533338956
```

XGBoost Model with 250 Decision Trees:

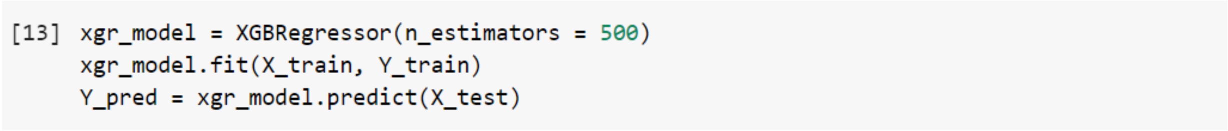


```
[11] xgr_model = XGBRegressor(n_estimators = 250)
     xgr_model.fit(X_train, Y_train)
     Y_pred = xgr_model.predict(X_test)
```

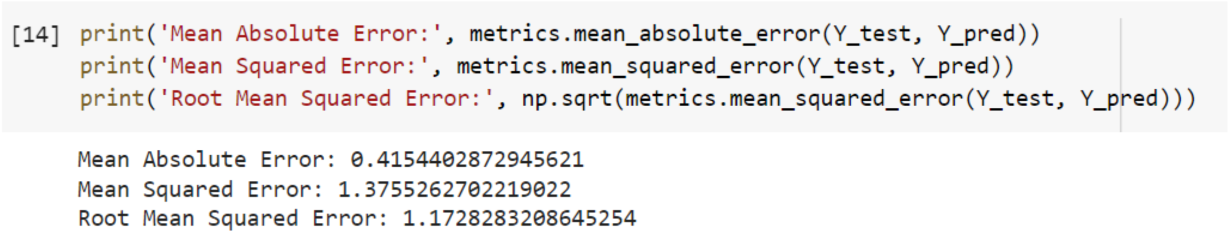


```
[12] print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, Y_pred))
     print('Mean Squared Error:', metrics.mean_squared_error(Y_test, Y_pred))
     print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))

     Mean Absolute Error: 0.41722835516141327
     Mean Squared Error: 1.3745739584516214
     Root Mean Squared Error: 1.1724222611549227
```

XGBoost Model with 500 Decision Trees:



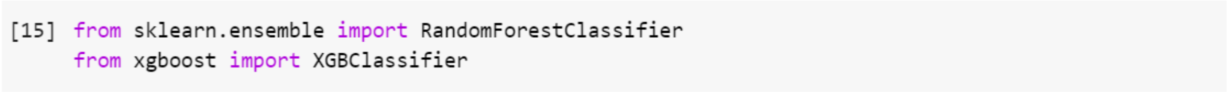
```
[13] xgr_model = XGBRegressor(n_estimators = 500)
     xgr_model.fit(X_train, Y_train)
     Y_pred = xgr_model.predict(X_test)
```



```
[14] print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, Y_pred))
     print('Mean Squared Error:', metrics.mean_squared_error(Y_test, Y_pred))
     print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))

     Mean Absolute Error: 0.4154402872945621
     Mean Squared Error: 1.3755262702219022
     Root Mean Squared Error: 1.1728283208645254
```

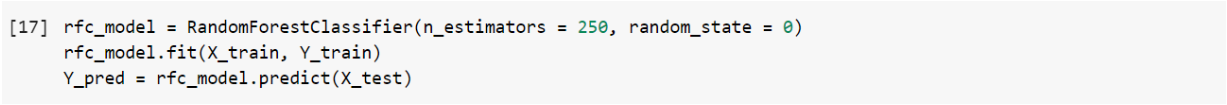
RANDOM FOREST & XGBOOST CLASSIFICATION

The Random Forest & XGBoost Regression Implementations are depicted below:

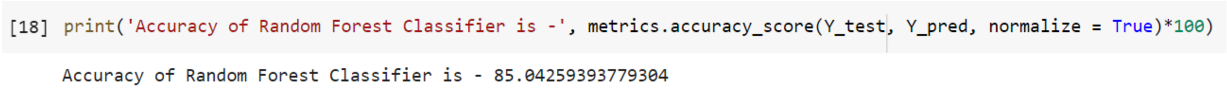


```
[15] from sklearn.ensemble import RandomForestClassifier
     from xgboost import XGBClassifier
```

Random Forest Model with 250 Decision Trees:

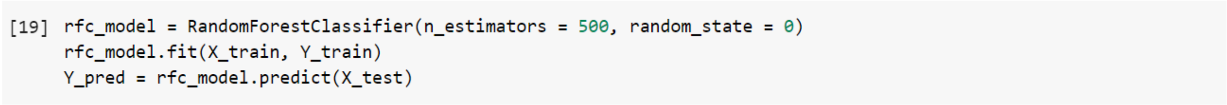


```
[17] rfc_model = RandomForestClassifier(n_estimators = 250, random_state = 0)
     rfc_model.fit(X_train, Y_train)
     Y_pred = rfc_model.predict(X_test)
```

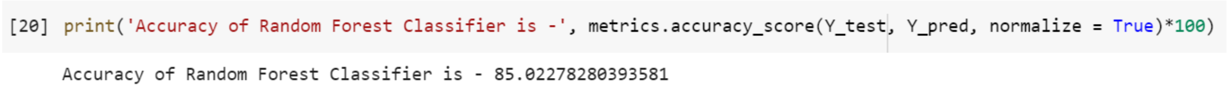


```
[18] print('Accuracy of Random Forest Classifier is -', metrics.accuracy_score(Y_test, Y_pred, normalize = True)*100)

     Accuracy of Random Forest Classifier is - 85.04259393779304
```

Random Forest Model with 500 Decision Trees:

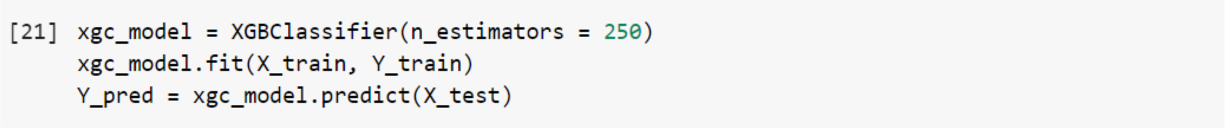


```
[19] rfc_model = RandomForestClassifier(n_estimators = 500, random_state = 0)
     rfc_model.fit(X_train, Y_train)
     Y_pred = rfc_model.predict(X_test)
```

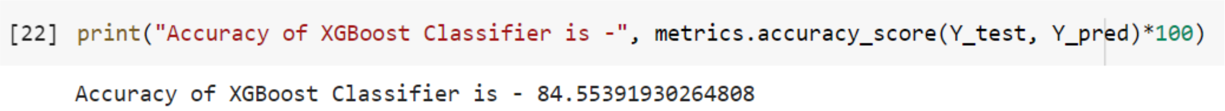


```
[20] print('Accuracy of Random Forest Classifier is -', metrics.accuracy_score(Y_test, Y_pred, normalize = True)*100)

     Accuracy of Random Forest Classifier is - 85.02278280393581
```

XGBoost Model with 250 Decision Trees:

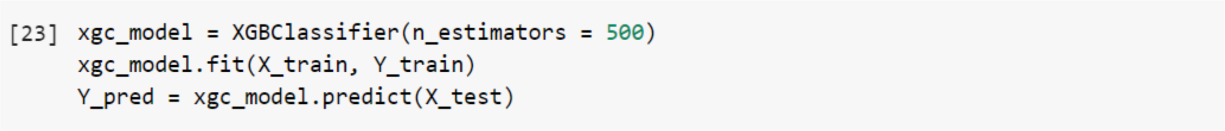


```
[21] xgc_model = XGBClassifier(n_estimators = 250)
     xgc_model.fit(X_train, Y_train)
     Y_pred = xgc_model.predict(X_test)
```

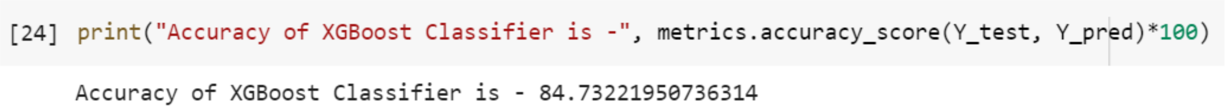


```
[22] print("Accuracy of XGBoost Classifier is -", metrics.accuracy_score(Y_test, Y_pred)*100)

     Accuracy of XGBoost Classifier is - 84.55391930264808
```

XGBoost Model with 500 Decision Trees:



```
[23] xgc_model = XGBClassifier(n_estimators = 500)
     xgc_model.fit(X_train, Y_train)
     Y_pred = xgc_model.predict(X_test)
```



```
[24] print("Accuracy of XGBoost Classifier is -", metrics.accuracy_score(Y_test, Y_pred)*100)

     Accuracy of XGBoost Classifier is - 84.73221950736314
```

EXPERIMENTAL RESULTS & ANALYSIS

The Statistics of the 4 Regression Models have been depicted in the table below:

PREDICTION PARAMETER	RANDOM FOREST MODEL - 1	XGBOOST MODEL - 1
Mean Absolute Error	0.40358	0.41723
Mean Squared Error	1.48976	1.37457
Root Mean Squared Error	1.22056	1.17242

Table 1: Prediction Statistics of Random Forest & XGBoost Regression Model when ‘n_estimators = 250’

PREDICTION PARAMETER	RANDOM FOREST MODEL - 2	XGBOOST MODEL - 2
Mean Absolute Error	0.40260	0.41544
Mean Squared Error	1.47964	1.37553
Root Mean Squared Error	1.21641	1.17283

Table 2: Prediction Statistics of Random Forest & XGBoost Regression Model when ‘n_estimators = 500’

The Statistics of the 4 Classification Models have been depicted in the table below:

PREDICTION PARAMETER	RANDOM FOREST MODEL - 1	XGBOOST MODEL - 1
Accuracy	85.04259	84.55392

Table 3: Accuracy of Random Forest & XGBoost Classification Model when ‘n_estimators = 250’

PREDICTION PARAMETER	RANDOM FOREST MODEL - 2	XGBOOST MODEL - 2
Accuracy	85.02278	84.73222

Table 4: Accuracy of Random Forest & XGBoost Classification Model when ‘n_estimators = 500’

For Regression Models, XGBoost Models perform better compared to Random Forest Model producing less Mean Squared Error and Root Mean Squared Error. For Classification Models, Random Forest Models perform better compared to XGBoost Models producing better accuracy by nearly 0.3-0.5%.

The difference in the values of accuracies suggest that both the models perform nearly equally well. The Model performance, when compared to each other, might go up and down, but will possibly remain very close in accuracy.

CONCLUSION

Although, there are several predictor variables that can enhance the accuracy of predictions in Fluoride Contamination, the six predictors, namely – Actual Evapotranspiration, Potential Evapotranspiration, Soil pH, Slope (0-60% in Majority area of the land), Aridity and Precipitation can easily predict the levels of Fluoride.

It can also be concluded that Random Forest Modelling and XGBoost Modelling, both results to nearly equal. An algorithm might perform better for Regression, while the another might perform better for Classification. These both can be used interchangeably without hesitation.

FUTURE SCOPE

Groundwater is excessively used in India and Fluoride contamination in Groundwater is a growing concern these days. It is important to pin-point these locations of high risk and provide remedial measures, in order to free the population of the hazardous effects of diseases like Fluorosis. Also, India has set a target to reach the National average of 1.0 mg/L fluoride in Groundwater from the existing 1.5 mg/L mark. So, this project has a societal benefit added to it.

Our Project helps us to achieve both these goals. This project can also be expanded to many other Groundwater contaminants like Arsenic & Nitrate which in-turn can help to provide quality water to every household in India. This project can be envisioned to improve the current accuracy of predictions.

REFERENCES

- Joel E. Podgorski, Pawan Labhasetwar, Dipankar Saha, and Michael Berg. Environmental Science & Technology **2 018** *5 2* (17), 9889-9898. DOI: 10.1021/acs.est.8b01679
- WHO Expert Committee on Oral Health Status Fluoride Use, Fluorides and Oral Health: Report of the WHO Expert Committee on Oral Health Status and Fluoride Use; World Health Organization: 1994; Vol. 846.
- World Health Organization Guidelines for Drinking-Water Quality, 4th ed.; World Health Organization: 2011.
- Saha, D.; Shekhar, S.; Ali, S.; Vittala, S. S.; Raju, N. J. Recent Hydrogeological Research in India. Proc. Indian Natl. Sci. Acad., Part A 2016, 82 (3), 787−803.
- Edmunds, W. M.; Smedley, P. L. Fluoride in natural waters. In Essentials of Medical Geology; Springer: Dordrecht, 2013; pp 311−336.
- Ali, S.; Thakur, S. K.; Sarkar, A.; Shekhar, S. Worldwide contamination of water by fluoride. Environ. Chem. Lett. 2016, 14 (3), 291−315.
- Amini, M.; Mueller, K.; Abbaspour, K. C.; Rosenberg, T.; Afyuni, M.; Møller, K. N.; Sarr, M.; Johnson, C. A. Statistical modeling of global geogenic fluoride contamination in groundwaters. Environ. Sci. Technol. 2008, 42 (10), 3662−3668.
- Shortt, H.; Pandit, C.; Raghavachari, R. S. T. Endemic fluorosis in the Nellore district of South India. Indian Med. Gazette 1937, 72 (7), 396.
- Singh, C. K.; Kumari, R.; Singh, N.; Mallick, J.; Mukherjee, S. Fluoride enrichment in aquifers of the Thar Desert: controlling factors and its geochemical modelling. Hydrological Processes 2013, 27 (17), 2462−2474.
- Amini, M.; Abbaspour, K. C.; Berg, M.; Winkel, L.; Hug, S. J.; Hoehn, E.; Yang, H.; Johnson, C. A. Statistical modeling of global geogenic arsenic contamination in groundwater. Environ. Sci. Technol. 2008, 42 (10), 3669−3675.
- Kumar, S.; Venkatesh, A.; Singh, R.; Udayabhanu, G.; Saha, D. Geochemical signatures and isotopic systematics constraining dynamics of fluoride contamination in groundwater across Jamui district, Indo-Gangetic alluvial plains. Chemosphere 2018, 205, 493− 505.
- Singh, C. K.; Mukherjee, S. Aqueous geochemistry of fluoride enriched groundwater in arid part of Western India. Environ. Sci. Pollut. Res. 2015, 22 (4), 2668−2678.
- Raju, N. J.; Dey, S.; Gossel, W.; Wycisk, P. Fluoride hazard and assessment of groundwater quality in the semi-arid Upper Panda River basin, Sonbhadra district, Uttar Pradesh, India. Hydrol. Sci. J. 2012, 57 (7), 1433−1452.
- Raju, N. J. Prevalence of fluorosis in the fluoride enriched groundwater in semi-arid parts of eastern India: Geochemistry and health implications. Quaternary International 2017, 443, 265−278.
- Reddy, A.; Reddy, D.; Rao, P.; Prasad, K. M. Hydrogeochemical characterization of fluoride rich groundwater of Wailpalli watershed, Nalgonda District, Andhra Pradesh, India. Environ. Monit. Assess. 2010, 171 (1−4), 561−577.
- Rao, N. S.; Subrahmanyam, A.; Rao, G. B. Fluoride-bearing groundwater in Gummanampadu sub-basin, Guntur district, Andhra Pradesh, India. Environ. Earth Sci. 2013, 70 (2), 575−586.
- Padhi, S.; Muralidharan, D. Fluoride occurrence and mobilization in geo-environment of semi-arid Granite watershed in southern peninsular India. Environ. Earth Sci. 2012, 66 (2), 471−479.
- https://towardsdatascience.com/understanding-random-forest-58381e0602d2
- https://builtin.com/data-science/random-forest-algorithm
- https://towardsdatascience.com/multivariate-logistic-regression-in-python-7c6255a286ec
- www.machinelearningmastery.com