

**EXPLOTARY ANALYSIS OF RAINFALL DATA IN INDIA FOR  
AGRICULTURE**

**BACHELOR OF ENGINEERING  
IN  
COMPUTER SCIENCE AND ENGINEERING  
submitted by**

**TEAM ID: PNT2022TMID04534**

<b>NITHIESH B</b>	<b>- 737819CSR125</b>
<b>NANDHA KISHORE K C</b>	<b>- 737819CSR115</b>
<b>JAYARAM S</b>	<b>- 737819CSR071</b>
<b>KARMUKIL D</b>	<b>- 737819CSR077</b>

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
KONGU ENGINEERING COLLEGE**

## **1.INTRODUCTION**

- 1.1 Project Overview
- 1.2 Purpose

## **2.LITERATURE SURVEY**

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

## **3.IDEATION & PROPOSED SOLUTION**

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

## **4.REQUIREMENT ANALYSIS**

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

## **5.PROJECT DESIGN**

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

## **6.PROJECT PLANNING & SCHEDULING**

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

## **7. CODING & SOLUTIONING**

- 7.1 Feature 1
- 7.2 Feature 2

## **8. TESTING**

8.1 Test Cases

8.2 User Acceptance Testing

## **9. RESULTS**

9.1 Performance Metrics

## **10.ADVANTAGES & DISADVANTAGES**

## **11.CONCLUSION**

## **12.FUTURE SCOPE**

## **13.APPENDIX**

# **1 INTRODUCTION**

## **1.1 Project Overview**

India is an agricultural country and secondary agro based market. Rainfall prediction model plays a vital role in prediction of the rainfall in a specific region and by using various technique like regression and find the best which is more suitable rainfall prediction. This model help the agriculture to improve and help the farmer to decide the crop to be cultivated.

India is well known for agriculture and market place. It has steady monsoon. Economic growth of the country depends on the amount of the country depends on the amount of more or less which lead to scarcity of agriproducts which causes food demand and food waste . In our analysis we are trying to understand the behaviour of rainfall in India over the years by months and different subdivisions.

## **1.2 Purpose**

- Rainfall Prediction model help us to identify the rainfall amount in the particular region.
- This rainfall prediction help us to identify the amount of rainfall to utilize the water resources for the agriculture to produce the agri-products according to data.

## **2.Literature survey:**

### **2.1:Existing problem:**

Rainfall is the important to human life. So the prediction accuracy is must more. In this paper we try to predict the rainfall. Rainfall forecasting has gained utmost research relevance in recent times due to its complexities and persistent applications such as flood forecasting and monitoring of pollutant concentration levels, among others. Existing models use complex statistical models that are often too costly, both computationally and budgetary, or are not applied to downstream applications. Therefore, approaches that machine learning algorithm use in conjunction with time-series data are being explored as an alternative to overcome these drawbacks. To this end, this study presents a comparative analysis using simplified rainfall estimation models based on conventional Machine Learning algorithms and Deep Learning architectures that are efficient for these downstream applications. Models based on LSTM Stacked-LSTM, Bidirectional-LSTM Networks, XGBoost, and an ensemble of Gradient Boosting Regressor, super vector machine and an Extra-trees Regressor were compared in the task of forecasting hourly rainfall volumes using time-series data. Climate data from 2000 to 2020 from five major cities in the United Kingdom were used. The evaluation metrics of Loss, Root Mean Squared, mean absolute error, Error, and Root Mean Squared Logarithmic Error were used to evaluate the models' performance. Results show that a Bidirectional-LSTM Network can be used as a rainfall forecast model with comparable performance to Stacked-LSTM Networks. Among all the models tested, the Stacked-LSTM Network with two hidden layers and the Bidirectional-LSTM Network performed best. This suggests that models based on LSTM-Networks with fewer hidden layers perform better for this approach; denoting its ability to be applied as an approach for budget-wise rainfall forecast applications.

## 2.2 References

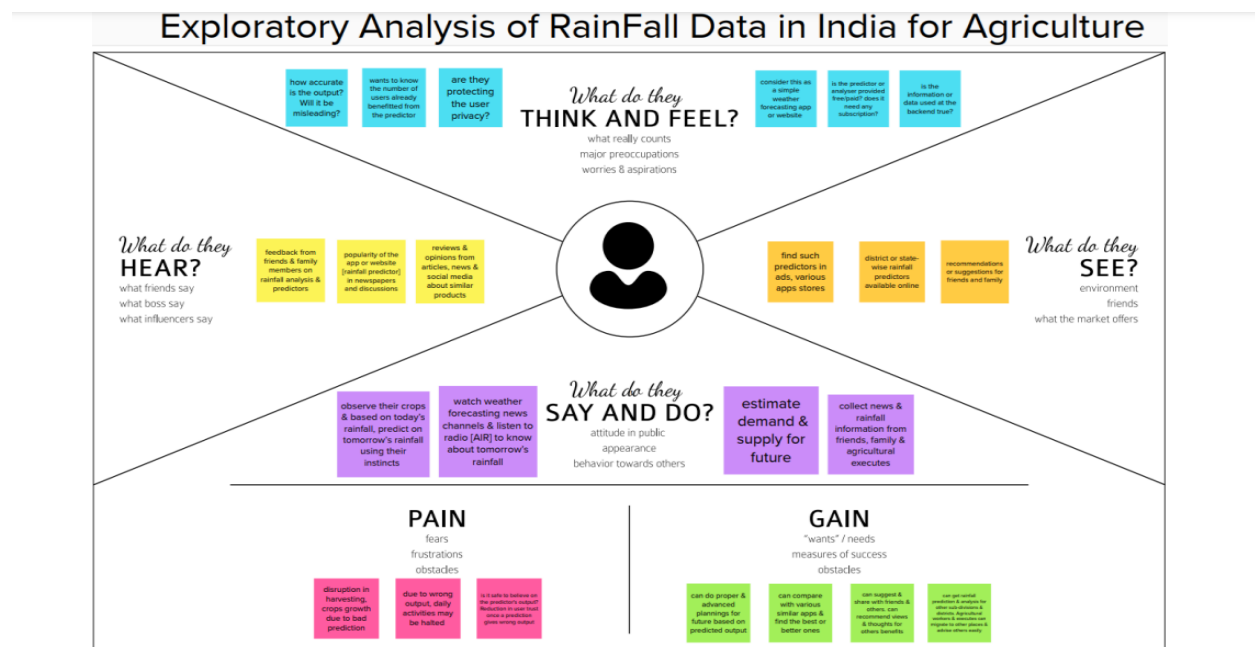
1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. URL <http://tensorflow.org/>, Software available from tensorflow.org.
- 2.. [2] P. Shirisha, K.V. Reddy, D. Pratap, Real-time flow forecasting in a watershed using rainfall forecasting model and updating model, *Water Resour. Manag.* 33 (14) (2019) 4799–4820
3. B. Wang, B. Xiang, J. Li, et al., Rethinking Indian monsoon rainfall prediction in the context of recent global warming, *Nat. Commun.* 6 (2015).
4. R.T. Kabo-Bah, C. Diji, K. Nokoe, Y. Mulugetta, D. Obeng-Ofori, K. Akpoti, et al., Multiyear rainfall and temperature trends in the volta river basin and their potential impact on hydropower generation in Ghana, *Climate* 4 (4) (2016) 49

## 2.3 Problem statement:

Rainfall is the major problem in the agriculture and the economy of the country. Rainfall is more in some areas and less in some areas. Due to this there is scarcity of agri products may occur. India is an agricultural country and secondary agro based market will be steady with a good monsoon. The economic growth of each year depends on the amount of duration of monsoon rain, bad monsoon can lead to destruction of some crops, which may result in scarcity of some agricultural products which in turn can cause food inflation, insecurity and public unrest. In our analysis we are trying to understand the behavior of rainfall in India over the years, by months and different subdivisions. Rainfall occurring at the end of the monsoon season provides stored soil moisture and sometimes irrigation water for the rabi crop, which is sown in the post-monsoon season. The summer monsoon therefore is responsible for both Kharif and Rabi crop production over India. Most of the Indian agricultural land is irrigated by the southwest monsoon. Crops such as wheat, rice, pulses, which are a staple in Indian diets, need heavy rainfalls to grow. Rubber trees in the southern region require heavy and regular rain with high temperatures. So, it is important to analyze the rainfall data for agriculture in India

## 3 IDEATION & PROPOSED SOLUTION:

### 3.1 EMPATHY MAP CANVAS:



## 3.2 IDEATION & BRAINSTORMING:

### 1 Define your problem statement

What problem are you trying to solve? Frame your problem as a clear, brief, life statement. This will be the focus of your brainstorm.

10 minutes

### Problem Statement

Agriculture is the backbone of the Indian economy. For agriculture, the most important thing is water source, i.e. rainfall. The prediction of the amount of rainfall gives alertness to farmers by knowing early they can protect their crops from rain. So, it is important to predict the rainfall accurately as much as possible. Exploration and analysis of data on rainfall over various regions of India and especially the regions where agricultural works have been done persistently in a wide range. With the help of analysis and the resultant data, future rainfall prediction for those regions using various machine learning techniques such as K-Nearest Classifier, SVM Classifier, Decision tree, Naive Bayes Classifier, Logistic regression etc.

### 2 Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

#### Nithiesh B

- Inspecting atmospheric pressure and various regions geographically distinguishable
- Forecasting the amount of rainfall based on the day or hour of rainfall
- Ensure data accuracy and adequacy for prediction

#### Jayaram S

- Collecting strong temperature of various regions & sub-divisions
- Looking into the sky early to observe the clouds density
- Logistic Regression to predict tomorrow's rainfall (yes or no)
- Naive Bayes Classifier
- Ensuring proper weather forecast performance
- XGBoost Classifier

#### Common Ideas Shared

- Random Forest Classifier
- Analysis & Prediction based on the day or hour of rainfall
- Linear Regression on rainfall trends (based on features)
- K-Nearest Neighbours
- Examining trends & improvements in rain-related effects
- Time Series Forecasting

#### Nandha Kishore KC

- Observing & noting down daily temperatures
- Watching or analysing the rainfall trends season wise
- Neural Network Classifiers
- Forecasting months to receive rainfall based on previous years' data
- Decision Tree

#### Karmukil D

- SVM Classifier
- Data collection & learning the trends present in it

### 3 Group Ideas

Start brainstorming your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is larger than six sticky notes, try and split it up into smaller sub-groups.

20 minutes

#### Data Mining Ideas

- Observing & noting down daily temperatures
- Proctoring wind-speed direction wise
- Data collection & learning the trends present in it
- Ensure data correctness & adequacy for prediction
- Comparing present weather forecast performance

#### Agricultural Intention (Farmers usually do so)

- Forecasting months to receive rainfall based on previous years' data
- Looking into the sky daily to observe the clouds density
- Analysing & Proctoring notes based on observed, monthly or annual crop yields
- Examining trends & improvements to view rainfall effects

#### Pattern Recognition

- Data collection & learning the trends present in it
- Forecasting months to receive rainfall based on previous years' data
- Inspecting atmospheric pressure of various regions geographically distinguishable
- Collecting strong temperature of various regions & sub-divisions
- Watching or analysing the rainfall trends season wise

#### AI Approaches

##### ML Algorithms

- Support Vector Classifier
- Naive Bayes Classifier
- Logistic Regression Classifier
- K-Nearest Neighbours
- Decision Tree
- Time Series Forecasting
- Random Forest Classifier
- SVM Classifier

##### Deep Learning Techniques

- Neural Network Classifier

### 4 Prioritize

You have visual of all on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

## 3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	Predicting Rainfall in India for agriculture
2.	Idea / Solution description	Using Exploratory Data Analysis
3.	Novelty / Uniqueness	Best fit machine learning algorithm is usedalong with data analysis
4.	Social Impact / Customer Satisfaction	Customer would come to know about the details like amountof rainfall, rainfall start time etc...
5.	BusinessModel (Revenue Model)	Subscription Revenue model
6.	Scalability of the Solution	Wide Scalability



## 3.4 PROBLEM SOLUTION FIT

Exploratory Analysis of Rainfall Data in India For Agriculture

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> Who is your customer? i.e. working parents of 0-5 yr. kids People who are the part of farming and directly dependent on rainfall, especially farmers	<b>6. CUSTOMER CONSTRAINTS</b> What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices. Lack of network and technology Low Budget	<b>5. AVAILABLE SOLUTIONS</b> Which solutions are available to the customers when they face the problem? or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking predicting rainfall by cloud movements in the sky based on cloud types Pros: It worked and rainfall is predicted correctly. Cons: Often inaccurate and rainfall occur in different place	Explore AS, differential
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. Uncertainty of rainfall in different places in various seasons.	<b>9. PROBLEM ROOT CAUSE</b> What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations The main cause of problem is physical factors like atmospheric pressure, humidity, heat intensity etc. These factors are responsible for changes in climate, weather and seasons.	<b>7. BEHAVIOUR</b> What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace) Finding the most appropriate weather from media Using weather application in smartphones.	
Focus on J&P, tap into BE, understand RC	<b>3. TRIGGERS</b> What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news. Seeing false news in news channel about the weather in their area.	<b>10. YOUR SOLUTION</b> If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.	<b>8. CHANNELS of BEHAVIOUR</b> <b>8.1 ONLINE</b> What kind of actions do customers take online? Extract online channels from #7 <b>8.2 OFFLINE</b> What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.	Focus on J&P, tap into BE, understand RC
<b>4. EMOTIONS: BEFORE / AFTER</b> How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design Lack of confidence, hopelessness.				

## 4 REQUIREMENT ANALYSIS:

### 4.1 Functional requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	Import necessary packages	Importing packages like NumPy,pandas etc.
FR-2	Download and load dataset	Download the dataset.Load the appropriate dataset for the model.
FR-3	Preprocessing	Making a suitable data for building the model.
FR-4	Built Machinelearning model	Choose the best and optimised algorithm.
FR-5	Train the data	Train the model withthe help of training data

FR-6	Test the model	Test the model for best performance ,optimisation,evaluation and analysis.
------	----------------	--

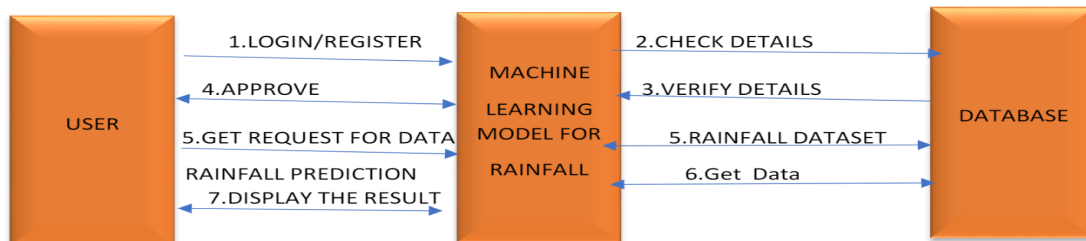
## 4.2 NonFunctional Requirement:

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The application can be used by the customers or persons easily.
NFR-2	Security	Security is provided .So it is safe to use the application and no problems will occur.
NFR-3	Reliability	Supporting the application.Detect the problems and rectify it.
NFR-4	Performance	This modelwill provide outputwith high accuracy,.
NFR-5	Availability	Any person can use this application at any timeand at anywhere.
NFR-6	Scalability	It provide betterautomation,enhanced modulation andcost effectiveness.

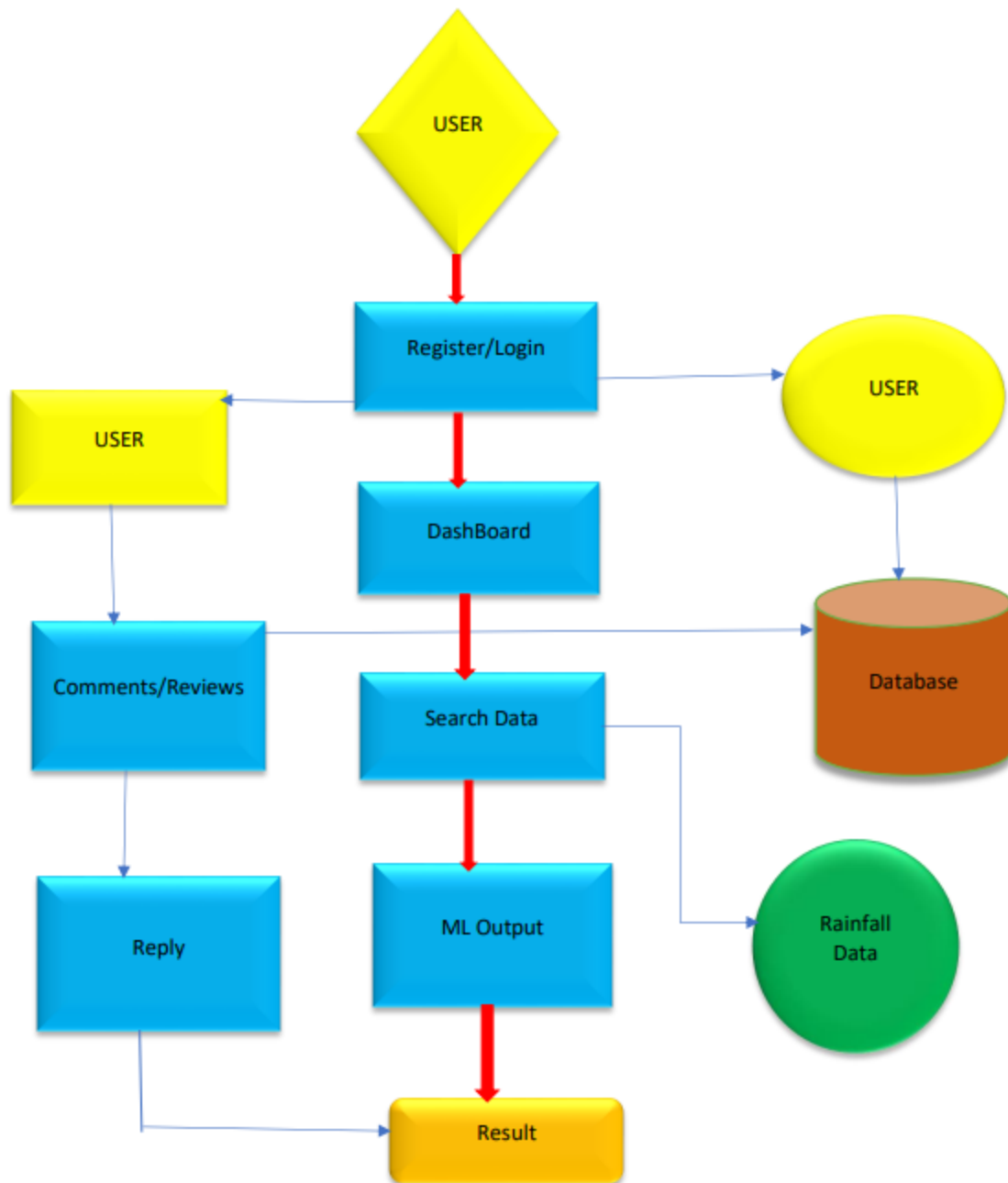
## PROJECT DESIGN:

### 5.1 Data Flow Diagrams:

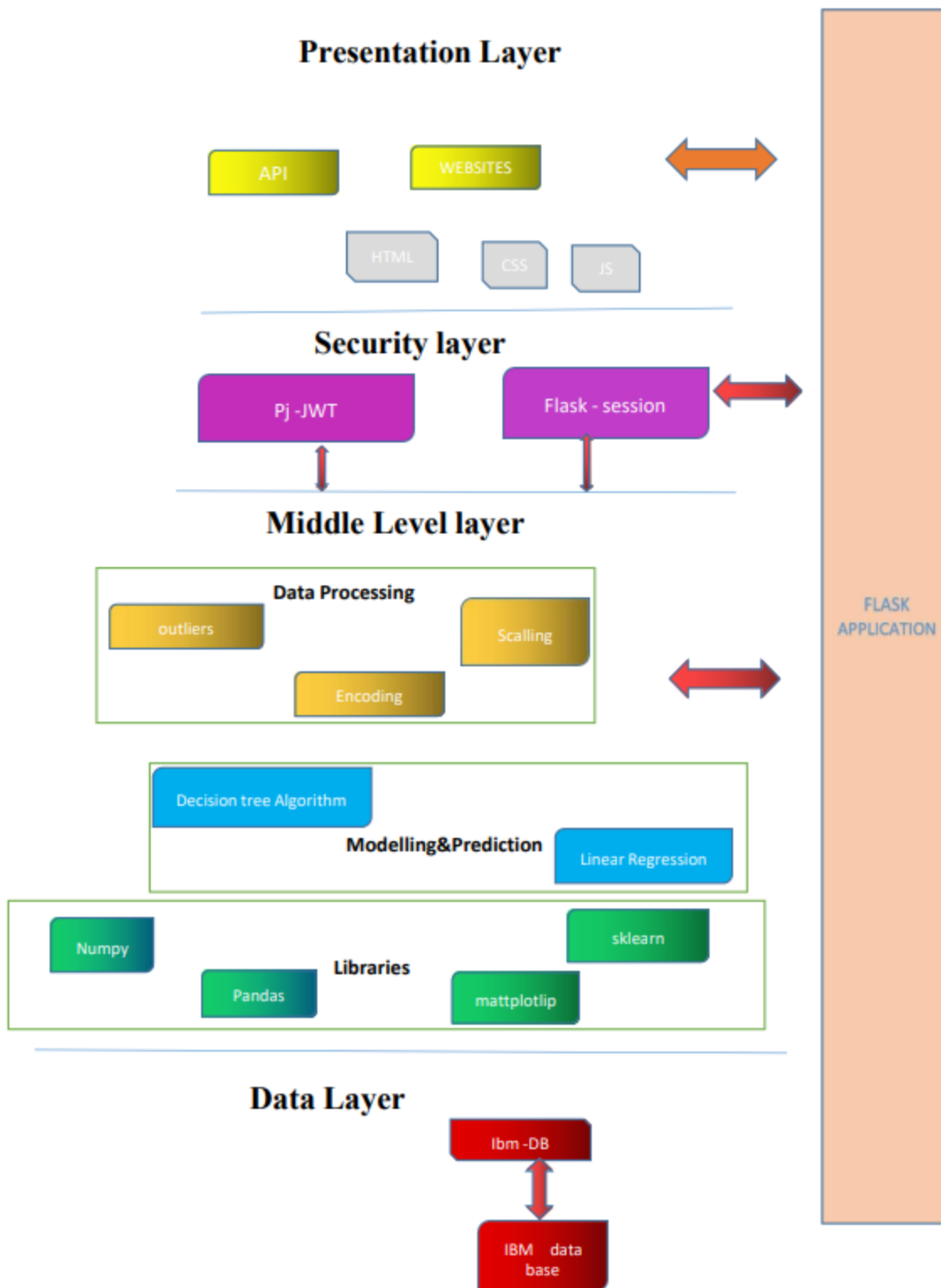
O-LEVEL ARCHITECTURE:



## 2-LEVEL ARCHITECTURE



## 5.2 Solution & Technical Architecture:



### 5.3 User Stories:

User Type	Functional Requirement(Epic)	User Story Number	User Story/ Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	User can register for the application by entering the email, password and other details.	I can access my account/ dashboard	High	Sprint-1
		USN-2	User receive the conformation mail.	I can receive confirmation email & click confirm	High	Sprint-2
	Login	USN-3	User can login through the registered email and password.	User can access the dashboard	High	Sprint-1
		USN-4	User can change the password and other things	New password is entered	High	Sprint-1
	Dashboard	USN-5	User can view the details about the page.		Medium	Sprint-1
	Data Prediction	USN-6	User can search the area /place where user want to get the data.	India regions	High	Sprint-2
		USN-7	User can see the visualization of the rainfall		High	Sprint-2

		USN-8	The prediction for the region for the future or past events respectively.		High	Sprint-2
	Information	USN-9	User can get the extra information about the agriculture	Can view the information	Medium	Sprint-1
Customer Support		USN-10	User can ask any queries related to agriculture.	Clarify doubts	High	Sprint-3
		USN-11	The incharge can analyse and rectify the mistakes		High	Sprint-3
		USN-12	Organize query session		Medium	Sprint-4
Development team	Core	USN-13	Develop the application that the user interface and maintenance should be taken care		High	Sprint-4
		USN-14	The website is responsive		High	Sprint-4
		USN-15	The data should be updated frequently.	The existing system should not be affected	High	Sprint-4

## 6.PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

Sprint	Functional Requirement(Epic)	User StoryNumber	User Story/Task	Story Points	Priority	Team Members
Sprint1	Registration	USN-1	User can register for the application by entering the email, password and other details.	5	High	Nandha Kishore, Jayaram, Nithiesh,
Sprint1		USN-2	User receive the conformation mail.	4	High	Nandha Kishore, Nithiesh, Karmukil
Sprint1	Login	USN-3	User can login through the registered email and password.	5	High	Nandha Kishore, Jayaram, Nithiesh, Karmukil
Sprint1		USN-4	User can change the password and other things	6	High	Jayaram, Nithiesh, Karmukil
Sprint2	Dashboard	USN-5	User can view the details about the page.	4	Medium	Nandha Kishore, Jayaram, Karmukil
Sprint2	Data Prediction	USN-6	User can search the area / place where user want to get the data.	5	High	Nandha Kishore, Jayaram, Nithiesh, Karmukil
Sprint2		USN-7	User can see the visualization of the rainfall	6	High	Nandha Kishore, Nithiesh, Karmukil
Sprint2		USN-8	The prediction for the region for the future or past events respectively.	5	High	Nandha Kishore, Jayaram, Nithiesh
Sprint3	Information	USN-9	User can get the extra information about the agriculture	6	Medium	Jayaram, Nithiesh, Karmukil
Sprint4		USN-10	User can ask any queries related to agriculture.	6	High	Nandha Kishore, Jayaram, Karmukil

Sprint3		USN-11	The incharge can analyse and rectify the mistakes	5	High	Nandha Kishore, Jayaram, Nithiesh, Karmukil
Sprint3		USN-12	Organize query session	3	Medium	Nandha Kishore, Nithiesh, Karmukil
Sprint4	Core	USN-13	Develop the application that the user interface and maintainance should be taken care	6	High	Nandha Kishore, Jayaram, Nithiesh, Karmukil
Sprint4		USN-14	The website is responsive	6	High	Nandha Kishore, Jayaram, Nithiesh
Sprint4		USN-15	The data should be updated frequently.	8	High	Nandha Kishore, Jayaram, Nithiesh, Karmukil

## 6.2 Sprint Delivery Schedule

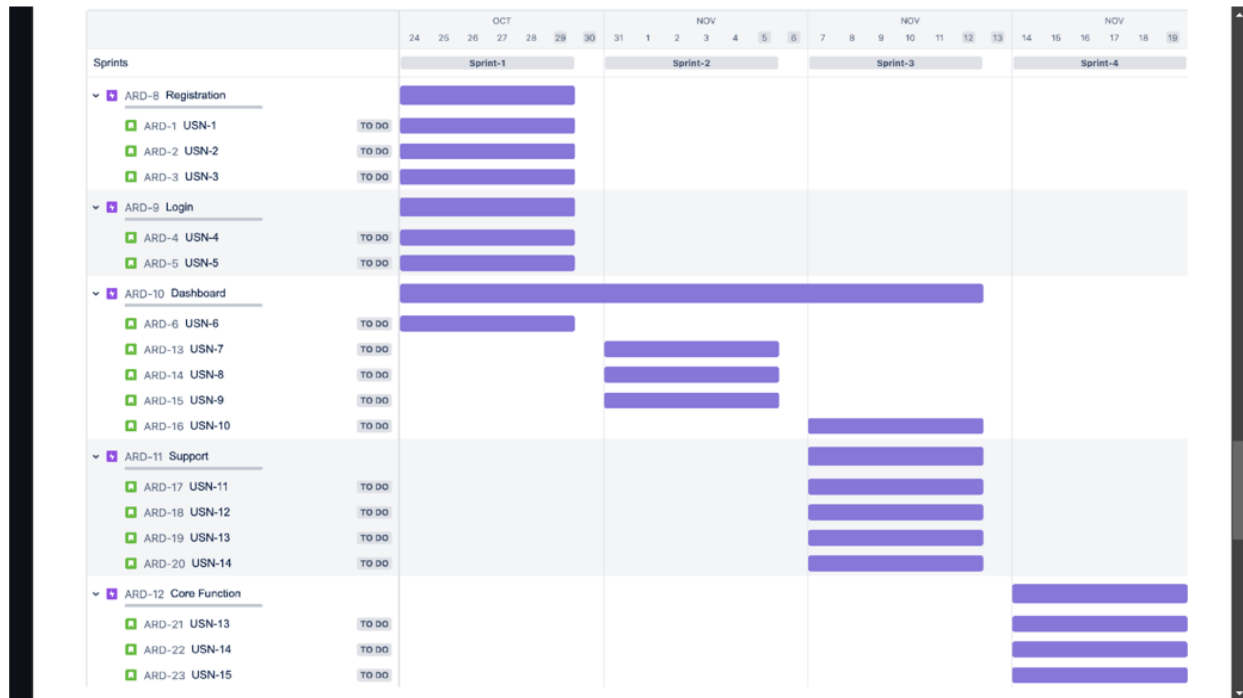
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date	Story Points Completed	Sprint Release Date
Sprint-1	20	6 Days	25 Oct 2022	30 Oct 2022	20	30 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	5 Nov 2022	20	5 Nov 2022
Sprint-3	20	6 Days	6 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	13 Nov 2022	19 Nov 2022	20	19 Nov 2022



## 6.3 Reports from JIRA

### RoadMap & Timeline

Tool Used: JIRA Software



## 7.CODING AND SOLUTION:

### 7.1 Feature 1

```
DataPreprocessing

Importing necessary libraries

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn import tree
import warnings

Import the dataset

[ ] rain_df= pd.read_csv('/content/rainfall_in_india_1901-2015.csv')
rain_df.head()
```

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6	3373.2	136.3	560.3	1696.3	980.3
1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5	3520.7	159.8	458.3	2185.9	716.7
2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225.0	2957.4	156.7	236.1	1874.0	690.6
3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40.1	3079.6	24.1	508.9	1977.6	571.0
4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344.7	2566.7	1.3	309.7	1624.9	630.8

```
[ ] rain_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4116 entries, 0 to 4115
Data columns (total 14 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   SUBDIVISION      4116 non-null   object
1   YEAR             4116 non-null   int64
2   JAN              4112 non-null   float64
3   FEB              4113 non-null   float64
4   MAR              4110 non-null   float64
5   APR              4112 non-null   float64
6   MAY              4113 non-null   float64
7   JUN              4111 non-null   float64
8   JUL              4109 non-null   float64
9   AUG              4112 non-null   float64
10  SEP              4110 non-null   float64
11  OCT              4109 non-null   float64
12  NOV              4105 non-null   float64
13  DEC              4106 non-null   float64
dtypes: float64(12), int64(1), object(1)
memory usage: 450.3+ KB

[ ] rain_df.isnull().sum()
```

```
[ ] rain_df=rain_df[["SUBDIVISION","YEAR","JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC"]]
rain_df.head()
```

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6
1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5
2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225.0
3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40.1
4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344.7

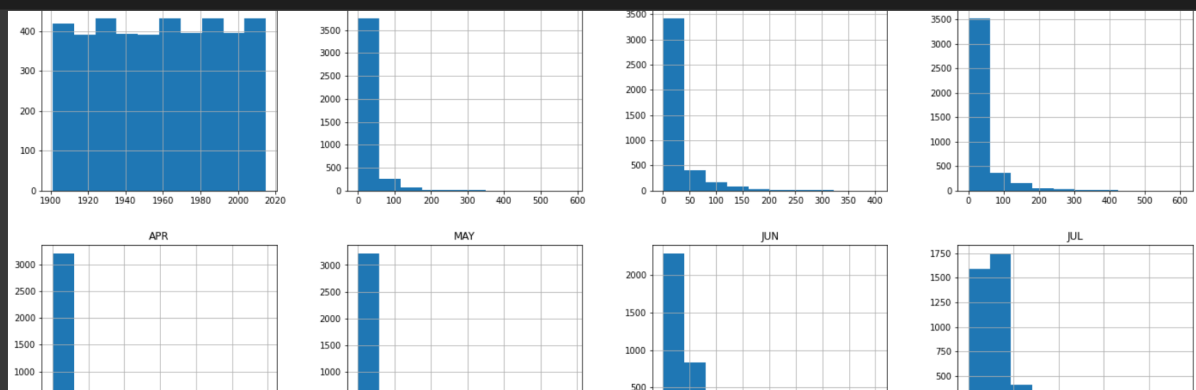
```
[ ] df1=pd.read_csv('/content/District_Rainfall.csv')
```

```
[ ] df1.isnull().any()
```

```
STATE/UT      False
DISTRICT      False
JAN           False
FEB           False
MAR           False
APR           False
MAY           False
JUN           False
JUL           False
AUG           False
SEP           False
OCT           False
NOV           False
DEC           False
ANNUAL        False
JAN+FEB       False
MAM           False
JJAS          False
OND           False
dtype: bool
```

Data visualisation ,dependent & independent,train & test

```
[ ] rain_df.hist(figsize=(24,24));
```



```
[ ] rain_df.SUBDIVISION.unique()
```

```
array(['ANDAMAN & NICOBAR ISLANDS', 'ARUNACHAL PRADESH',  
      'ASSAM & MEGHALAYA', 'NAGA MANI MIZO TRIPURA',  
      'SUB HIMALAYAN WEST BENGAL & SIKKIM', 'GANGETIC WEST BENGAL',  
      'ORISSA', 'JHARKHAND', 'BIHAR', 'EAST UTTAR PRADESH',  
      'WEST UTTAR PRADESH', 'UTTARAKHAND', 'HARYANA DELHI & CHANDIGARH',  
      'PUNJAB', 'HIMACHAL PRADESH', 'JAMMU & KASHMIR', 'WEST RAJASTHAN',  
      'EAST RAJASTHAN', 'WEST MADHYA PRADESH', 'EAST MADHYA PRADESH',  
      'GUJARAT REGION', 'SAURASHTRA & KUTCH', 'KONKAN & GOA',  
      'MADHYA MAHARASHTRA', 'MATATHWADA', 'VIDARBHA', 'CHHATTISGARH',  
      'COASTAL ANDHRA PRADESH', 'TELANGANA', 'RAYALSEEMA', 'TAMIL NADU',  
      'COASTAL KARNATAKA', 'NORTH INTERIOR KARNATAKA',  
      'SOUTH INTERIOR KARNATAKA', 'KERALA', 'LAKSHADWEEP'], dtype=object)
```

```
[ ] rain_df.describe()
```

```
[ ] from sklearn.preprocessing import LabelEncoder  
  
lab = LabelEncoder()  
rain_df.SUBDIVISION = lab.fit_transform(rain_df.SUBDIVISION)  
  
rain_df.head()
```

	SUBDIVISION	YEAR	MONTH	MAX_TEMP	MIN_TEMP	MEAN_TEMP	PRECEPTIONS	PRESSURE	WIND_SPEED	RAINFALL
0	0	1981	1	10.68	23.80	17.21	20.71	94.20	7.29	11.1
1	0	1981	2	12.99	27.48	20.21	8.59	94.10	7.15	27.8
2	0	1981	3	16.50	30.73	23.59	34.10	93.97	7.05	1.7
3	0	1981	4	21.06	35.07	28.04	35.36	93.78	9.35	94.1
4	0	1981	5	23.54	36.59	30.04	64.94	93.66	8.31	270.8

```
[ ] rain_df.SUBDIVISION.unique()
```

```
array([ 0,  1,  2, 21, 28, 10, 23, 15,  3,  9, 35, 31, 12, 24, 13, 14, 34,  
       8, 33,  7, 11, 26, 17, 19, 20, 32,  4,  5, 30, 25, 29,  6, 22, 27,  
      16, 18])
```

```
[ ] feature=rain_df[["SUBDIVISION","MONTH","MAX_TEMP","MIN_TEMP","MEAN_TEMP","PRECEPTIONS","PRESSURE","WIND_SPEED"]]
    target=rain_df["RAINFALL"]
```

## Training & Testing

```
[ ] acc=[]
    model=[]
```

```
[ ]
```

```
[ ] from sklearn.model_selection import train_test_split
    X_train,X_test,y_train,y_test=train_test_split(feature,target,test_size=0.2,random_state=2)
```

```
[ ] X_train
```

```
[ ]
```

13896	27	1	10.50	23.81	17.13	17.29	93.98	7.57
6637	14	2	14.53	29.01	21.75	6.71	94.03	7.86
2575	23	8	22.82	30.28	26.52	241.93	93.62	8.38
7336	8	5	22.84	35.88	29.34	39.16	93.74	10.20

12096 rows × 8 columns

```
[ ] y_train
```

14492	517.6
6598	21.8
1554	301.5
8237	93.5
12007	266.8
...	...
11798	4.7
13896	2.1
6637	117.9
2575	460.9
7336	14.2

Name: RAINFALL, Length: 12096, dtype: float64

```
[ ] X_test
```

```
[ ] y_test
```

12224	130.9
559	294.3
10735	286.2
13117	0.0
3670	2.9
...	...
946	16.2
12736	58.3
6162	141.3
12651	43.9
1404	11.1

Name: RAINFALL, Length: 3024, dtype: float64

## Build Model-Model Evaluation

```
[ ] from sklearn.tree import DecisionTreeRegressor
    from sklearn.metrics import roc_auc_score,classification_report,mean_squared_error,r2_score

[ ] # create a regressor object
    dtregressor = DecisionTreeRegressor(random_state = 0)

    # fit the regressor with X and Y data
    dtregressor.fit(X_train, y_train)

    DecisionTreeRegressor(random_state=0)

[ ] # predicting with regression model with X and Y
    y_train_pred=dtregressor.predict(X_train)
    y_test_pred=dtregressor.predict(X_test)

[ ] #Mean Squared Error and r2 Score
    print("MSE",mean_squared_error(y_train,y_train_pred),mean_squared_error(y_test,y_test_pred))
    print((r2_score(y_train,y_train_pred),(r2_score(y_test_pred,y_test))))

MSE 7.164917947869181e-37 7408.483804291254
(1.0, 0.7196069044190031)
```

```
[ ] model.append('Decision Tree')
    acc.append(dtregressor.score(X_test,y_test))
    print(dtregressor.score(X_test,y_test))

0.7207121752928417
```

## Random Forest

```
[ ] from sklearn.ensemble import RandomForestRegressor

[ ] # create a regressor object
    forest=RandomForestRegressor()

    # fit the regressor with X and Y data
    forest.fit(X_train,y_train)

    RandomForestRegressor()

[ ] # fit the regressor with X and Y data
    forest.fit(X_train,y_train)

    RandomForestRegressor()

[ ] # predicting with regression model with X and Y
    y_train_pred=forest.predict(X_train)
    y_test_pred=forest.predict(X_test)

[ ] print("MSE",mean_squared_error(y_train,y_train_pred),mean_squared_error(y_test,y_test_pred))
    print((r2_score(y_train,y_train_pred),(r2_score(y_test_pred,y_test))))

MSE 601.6440267475264 4323.741234140449
(0.9795898417476755, 0.8154716738237888)

[ ] model.append('Random Forest')
    acc.append(forest.score(X_test,y_test))
    print(forest.score(X_test,y_test))

0.8370019675037605
```

```
XGBoost

[ ] from xgboost import XGBRegressor

[ ] # create a regressor object
xgb = XGBRegressor()

# fit the regressor with X and Y data
xgb.fit(X_train,y_train)

[05:20:31] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
XGBRegressor()

[ ] # predicting with regression model with X and Y
y_train_pred=xgb.predict(X_train)
y_test_pred=xgb.predict(X_test)

[ ] print("MSE",mean_squared_error(y_train,y_train_pred),mean_squared_error(y_test,y_test_pred))
print((r2_score(y_train,y_train_pred),(r2_score(y_test_pred,y_test))))

MSE 6771.934017080991 6879.805083100889
(0.770269064732319, 0.6064046953302488)
```

```
MSE 6771.934017080991 6879.805083100889
(0.770269064732319, 0.6064046953302488)

[ ] model.append('XGB Boost')
acc.append(xgb.score(X_test,y_test))
print(xgb.score(X_test,y_test))

0.7406425056965733

Save the model

[ ] import pickle
file_name='Model/rainfall_pred.pkl'
pk1=open(file_name,'wb')
pickle.dump(forest,pk1)
pk1.close()
```

## 7.2 Feature 2

**Web code:**

**Index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<script

  src="https://kit.fontawesome.com/64d58efce2.js"

  crossorigin="anonymous"

></script>

<link rel="stylesheet" href="style.css" />

<link rel="stylesheet" href="path/to/font-awesome/css/font-awesome.min.css">

  <title>Exploratory Analysis of Rain Fall Data in India for Agriculture Login
Page</title>

</head>

<body>

  <div class="container">

    <div class="signin-signup">

      <form action="#" class="sign-in-form">

        <h2 class="title">Sign in</h2>

        <div class="input-field">

          <i class="fas fa-user"></i>

          <input type="text" placeholder="Username" />

        </div>

        <div class="input-field">

          <i class="fas fa-lock"></i>
```



```
<input type="password" placeholder="Password" />

</div>

<input type="submit" value="Login" class="btn solid" />

<p class="social-text">Or Sign in with social platforms</p>

<div class="social-media">

  <a href="#" class="social-icon">

    <i class="fab fa-facebook-f"></i>

  </a>

  <a href="#" class="social-icon">

    <i class="fab fa-twitter"></i>

  </a>

  <a href="#" class="social-icon">

    <i class="fab fa-google"></i>

  </a>

  <a href="#" class="social-icon">

    <i class="fab fa-linkedin-in"></i>

  </a>

</div>

</form>

<form action="#" class="sign-up-form">

  <h2 class="title">Sign up</h2>
```

```
<div class="input-field">

  <i class="fas fa-user"></i>

  <input type="text" placeholder="Username" />

</div>

<div class="input-field">

  <i class="fas fa-envelope"></i>

  <input type="email" placeholder="Email" />

</div>

<div class="input-field">

  <i class="fas fa-lock"></i>

  <input type="password" placeholder="Password" />

</div>

<input type="submit" class="btn" value="Sign up" />

<p class="social-text">Or Sign up with social platforms</p>

<div class="social-media">

  <a href="#" class="social-icon">

    <i class="fab fa-facebook-f"></i>

  </a>

  <a href="#" class="social-icon">

    <i class="fab fa-twitter"></i>

  </a>
```

```
<a href="#" class="social-icon">
```

```
<i class="fab fa-google"></i>
```

```
</a>
```

```
<a href="#" class="social-icon">
```

```
<i class="fab fa-linkedin-in"></i>
```

```
</a>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
<div class="panels-container">
```

```
<div class="panel left-panel">
```

```
<div class="content">
```

```
<h3><b style="font-size: 25px;" >Exploratory Analysis of Rain Fall Data  
in India for Agriculture</b></h3><br>
```

```
<h4>Welcomes you!!</h4>
```

```
<p> </p>
```

```
<button class="btn transparent" id="sign-up-btn">
```

```
Sign up
```

```
</button>
```

```
</div>
```

```

```

</div>

<div class="panel right-panel">

<div class="content">

<h1 style="font-size: 50px;">One of us ?</h1>

<p >

Join with us and grow with us...

</p>

<button class="btn transparent" id="sign-in-btn">

Sign in

</button>

</div>



</div>

</div>

</div>

</body>

<!-- loader part -->

<!--div class="loader-container">



</div-->

```
<script src="app.js"></script>

<script src="mongo.js"></script>

</body>

</html>
```

## **APP.JS**

```
const sign_in_btn = document.querySelector("#sign-in-btn");
const sign_up_btn = document.querySelector("#sign-up-btn");
const container = document.querySelector(".container");

sign_up_btn.addEventListener("click", () => {

  container.classList.add("sign-up-mode");

});

sign_in_btn.addEventListener("click", () => {

  container.classList.remove("sign-up-mode");

});
```

## **DISPLAY.html**

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Home</title>

</head>

<body>  <h1>you are successfully registered</h1>

</body>

</html>
```

### **Style.css:**

```
@import
url("https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;60
0;700;800&display=swap");

* {

  margin: 0;

  padding: 0;

  box-sizing: border-box;

}

body,

input { font-family: "Poppins", sans-serif;

}.container {

  position: relative;

  width: 100%;

  background-color: #fff;

  min-height: 100vh;
```

```
overflow: hidden;

}

.signin-signup {
  position: absolute;
  top: 50%;
  transform: translate(-50%, -50%);
  left: 75%;
  width: 50%;
  transition: 1s 0.7s ease-in-out;
  display: grid;
  grid-template-columns: 1fr;
  z-index: 5;
}

form {
  display: flex;
  align-items: center;
  justify-content: center;
  flex-direction: column;
  padding: 0rem 5rem;
  transition: all 0.2s 0.7s;
  overflow: hidden;
```

```
    grid-column: 1 / 2;
    grid-row: 1 / 2;
}

form.sign-up-form {
    opacity: 0;
    z-index: 1;
}

form.sign-in-form {
    z-index: 2;
}

.title {
    font-size: 2.2rem;
    color: #444;
    margin-bottom: 10px;
}

.input-field {
    max-width: 380px;
    width: 100%;
    background-color: #f0f0f0;
    margin: 10px 0;
    height: 55px;
```



```
border-radius: 55px;

display: grid;

grid-template-columns: 15% 85%;

padding: 0 0.4rem;

position: relative;
}

.input-field i {

text-align: center;

line-height: 55px;

color: #acacac;

transition: 0.5s;

font-size: 1.1rem;
}.input-field input {

background: none;

outline: none;

border: none;

line-height: 1;

font-weight: 600;

font-size: 1.1rem;

color: #333;
}
```

```
.input-field input::placeholder {  
  color: #aaa;  
  font-weight: 500;  
}
```

```
.social-text {  
  padding: 0.7rem 0;  
  font-size: 1rem;  
}
```

```
.social-media {  
  display: flex;  
  justify-content: center;  
}
```

```
.social-icon {  
  height: 46px;  
  width: 46px;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  margin: 0 0.45rem;  
  color: #333;  
  border-radius: 50%;
```

```
border: 1px solid #333;

text-decoration: none;

font-size: 1.1rem;

transition: 0.3s;

}

.social-icon:hover {

    color: #4481eb;

    border-color: #4481eb;

}.btn {

    width: 150px;

    background-color: #5995fd;

    border: none;

    outline: none;

    height: 49px;

    border-radius: 49px;

    color: #fff;

    text-transform: uppercase;

    font-weight: 600;

    margin: 10px 0;

    cursor: pointer;

    transition: 0.5s;}
```

```
.btn:hover {  
    background-color: #4d84e2;  
}  
  
.panels-container {  
    position: absolute;  
    height: 100%;  
    width: 100%;  
    top: 0;  
    left: 0;  
    display: grid;  
    grid-template-columns: repeat(2, 1fr);  
}
```

```
.container:before {  
    content: "";  
    position: absolute;  
    height: 2000px;  
    width: 2000px;  
    top: -10%;  
    right: 48%;  
    transform: translateY(-50%);
```

```
background-image: linear-gradient(-45deg, #4481eb 0%, #04befe 100%);  
transition: 1.8s ease-in-out;  
border-radius: 50%;  
z-index: 6;  
}
```

```
.image {  
width: 100%;  
transition: transform 1.1s ease-in-out;  
transition-delay: 0.4s;  
}
```

```
.panel {  
display: flex;  
flex-direction: column;  
align-items: flex-end;  
justify-content: space-around;  
text-align: center;  
z-index: 6;  
}
```

```
.left-panel {  
  pointer-events: all;  
  padding: 3rem 17% 2rem 12%;  
}
```

```
.right-panel {  
  pointer-events: none;  
  padding: 3rem 12% 2rem 17%;  
}
```

```
.panel .content {  
  color: #fff;  
  transition: transform 0.9s ease-in-out;  
  transition-delay: 0.6s;  
}
```

```
.panel h3 {  
  font-weight: 600;  
  line-height: 1;  
  font-size: 1.5rem;  
}
```

```
.panel p {  
    font-size: 0.95rem;  
    padding: 0.7rem 0;  
}
```

```
.btn.transparent {  
    margin: 0;  
    background: none;  
    border: 2px solid #fff;  
    width: 130px;  
    height: 41px;  
    font-weight: 600;  
    font-size: 0.8rem;  
}
```

```
.right-panel .image,  
.right-panel .content {  
    transform: translateX(800px);  
}
```

```
/* ANIMATION */
```

```
.container.sign-up-mode:before {  
    transform: translate(100%, -50%);  
    right: 52%;  
}
```

```
.container.sign-up-mode .left-panel .image,  
.container.sign-up-mode .left-panel .content {  
    transform: translateX(-800px);  
}
```

```
.container.sign-up-mode .signin-signup {  
    left: 25%;  
}
```

```
.container.sign-up-mode form.sign-up-form {  
    opacity: 1;  
    z-index: 2;  
}
```



```
.container.sign-up-mode form.sign-in-form {  
  opacity: 0;  
  z-index: 1;  
}
```

```
.container.sign-up-mode .right-panel .image,  
.container.sign-up-mode .right-panel .content {  
  transform: translateX(0%);  
}
```

```
.container.sign-up-mode .left-panel {  
  pointer-events: none;  
}
```

```
.container.sign-up-mode .right-panel {  
  pointer-events: all;  
}
```

```
@media (max-width: 870px) {  
  .container {  
    min-height: 800px;  
  }  
}
```

```
    height: 100vh;  
}
```

```
.signin-signup {  
    width: 100%;  
    top: 95%;  
    transform: translate(-50%, -100%);  
    transition: 1s 0.8s ease-in-out;  
}
```

```
.signin-signup,  
.container.sign-up-mode .signin-signup {  
    left: 50%;  
}
```

```
.panels-container {  
    grid-template-columns: 1fr;  
    grid-template-rows: 1fr 2fr 1fr;  
}
```

```
.panel {  
    flex-direction: row;
```

```
justify-content: space-around;  
align-items: center;  
padding: 2.5rem 8%;  
grid-column: 1 / 2;  
}
```

```
.right-panel {  
  grid-row: 3 / 4;  
}
```

```
.left-panel {  
  grid-row: 1 / 2;  
}
```

```
.image {  
  width: 200px;  
  transition: transform 0.9s ease-in-out;  
  transition-delay: 0.6s;  
}
```

```
.panel .content {
```

```
padding-right: 15%;  
  
transition: transform 0.9s ease-in-out;  
  
transition-delay: 0.8s;  
}
```

```
.panel h3 {  
    font-size: 1.2rem;  
}
```

```
.panel p {  
    font-size: 0.7rem;  
    padding: 0.5rem 0;  
}
```

```
.btn.transparent {  
    width: 110px;  
    height: 35px;  
    font-size: 0.7rem;  
}
```

```
.container:before {
```

```
width: 1500px;
height: 1500px;
transform: translateX(-50%);
left: 30%;
bottom: 68%;
right: initial;
top: initial;
transition: 2s ease-in-out;
}
```

```
.container.sign-up-mode:before {
  transform: translate(-50%, 100%);
  bottom: 32%;
  right: initial;
}
```

```
.container.sign-up-mode .left-panel .image,
.container.sign-up-mode .left-panel .content {
  transform: translateY(-300px);
}
```

```
.container.sign-up-mode .right-panel .image,  
.container.sign-up-mode .right-panel .content {  
  transform: translateY(0px);  
}
```

```
.right-panel .image,  
.right-panel .content {  
  transform: translateY(300px);  
}
```

```
.container.sign-up-mode .signin-signup {  
  top: 5%;  
  transform: translate(-50%, 0);  
}  
}
```

mango.js:

```
const mongoose=require("mongoose");  
const ejs=require("ejs");  
const express=require("express");  
const bp = require("body-parser");  
const app = express();
```

```

const
db="mongodb+srv://<jayaram>:<jayaramdharani>@cluster0.bpvhs.mongodb.net/d
atabase?retryWrites=true&w=majority";
const connectp={
  useNewUrlParser:true,
  useUnifiedTopology: true
};
mongoose.connect(db,connectp).then(()=>{console.info("connected to the DB");})
.catch((e)=>{
  console.log("error:",e);
});
javascript.js:
var pastSearches = []
var lat
var lon

//Setting Date
var date = (moment().format("L"))

//Calling functions to retrieve local storage on page load , if there is any

```

```
retrieveDailyWeatherStorage()
```

```
retrieveListStorage()
```

```
//Function to retrieve local storage for list and append it to the page
```

```
function retrieveListStorage() {  
    if (localStorage.getItem("pastSearches") !== null) {  
        var prevSearches = []  
        prevSearches = localStorage.getItem("pastSearches").split(",").reverse()  
        for (i = 0; i < prevSearches.length && i < 5; i++) {  
            pastSearches.push(prevSearches[i])  
        }  
        for (i = 0; i < 5 && i < prevSearches.length; i++) {  
            let searchItem = $("- ").attr("id", "listItem")  
            searchItem.text(prevSearches[i])  
            $(".list").append(searchItem)  
        }  
    }  
}

```

```
//Retrieving local storage for weather all weather information, create the elements  
and append them on the page
```



```

function retrieveDailyWeatherStorage() {

    var localWeatherDiv = $("<div>")

    //If statement to determine if the correct items are in local storage

        if    (localStorage.getItem("InitialCity")    !==    null    &&
localStorage.getItem("initialWeather")            !==    null    &&
localStorage.getItem("initialTempFixed")           !==    null    &&
localStorage.getItem("initialHumidity")             !==    null    &&
localStorage.getItem("initialWindSpeed")           !==    null    &&
localStorage.getItem("initalUvIndex") !== null) {

        //City name

        var LocalCityDiv = $("<div>").addClass("bigger")
        var LocalCity = localStorage.getItem("InitialCity")

        //Weather animation

        var localWeather = localStorage.getItem("initialWeather")
        var localWeatherAnimation = $("<span>")

        // applying class to weather animation span

        if (localWeather === "Clouds") {

            localWeatherAnimation.addClass("fas fa-cloud fa-2x")

```

```
} else if (localWeather === "Rain") {  
    localWeatherAnimation.addClass("fas fa-cloud-rain fa-2x")  
} else if (localWeather === "Clear") {  
    localWeatherAnimation.addClass("far fa-sun fa-2x")  
} else if (localWeather === "Snow") {  
    localWeatherAnimation.addClass("fas fa-snowflake fa-2x")  
} else if (localWeather === "Mist") {  
    localWeatherAnimation.addClass("fas fa-cloud-rain fa-2x")  
} else if (localWeather === "Haze") {  
    localWeatherAnimation.addClass("fas fa-smog fa-2x")  
}
```

```
LocalCityDiv.text(LocalCity + " " + date + " ");
```

```
LocalCityDiv.append(localWeatherAnimation)
```

```
//temperature
```

```
var localTempF = localStorage.getItem("initialTempFixed")
```

```
var LocalTempDiv = $("<div>")
```

```
LocalTempDiv.text("Temperature : " + localTempF +  
"F").addClass("tempClass")
```

```
// Humidity

var localHumidity = localStorage.getItem("initialHumidity")

var localHumidityDiv = $("<div>")

        localHumidityDiv.text("Humidity : " + localHumidity + "
%").addClass("humidity")
```

```
// Wind Speed

var localWindSpeed = localStorage.getItem("initialWindSpeed")

var localWindSpeedDiv = $("<div>")

        localWindSpeedDiv.text("Wind Speed : " + localWindSpeed + "
MPH").addClass("windSpeed")
```

```
//UV Index

var localUvIndexDiv = $("<div>")

var localUvIndex = localStorage.getItem("initialUvIndex")

var localUvIndexSpan = $("<span>")

localUvIndexDiv.text("UV Index: ").addClass("UvIndex")

localUvIndexDiv.append(localUvIndexSpan)

localUvIndexSpan.text(localUvIndex)
```

```
//setting the color of uv index span based on uv value
```

```

    if (localUvIndex <= 3) {
        localUvIndexSpan.addClass("low")
    } else if (localUvIndex > 7.01) {
        localUvIndexSpan.addClass("high")
    } else if (3.1 <= localUvIndex <= 6.9) {
        localUvIndexSpan.addClass("medium")
    }

    //Appending to page

    $(localWeatherDiv).append(LocalCityDiv, LocalTempDiv, localHumidityDiv,
localWindSpeedDiv, localUvIndexDiv)

    $("#dayForecast").append(localWeatherDiv)
}

// Getting local storage for 5 day divs

//For Day One
var localNextDayDiv = $("<div>")

        if    (localStorage.getItem("nextDayDate")    !==    null    &&
localStorage.getItem("weatherNextDay")    !==    null    &&
localStorage.getItem("fNextDay")    !==    null    &&

```

```
localStorage.getItem("nextDayHumidity") !== null) {

    // Get Date

    var localNextDayDate = localStorage.getItem("nextDayDate")

    console.log(localNextDayDate)

    var LocalNextDayDateDiv = $("

").addClass("head")

    LocalNextDayDateDiv.text(localNextDayDate)


    //Get weather

    var LocalWeatherNextDay = localStorage.getItem("weatherNextDay")

    var LocalWeatherNextDayDiv = $("

")


    // If statement for weather

    if (LocalWeatherNextDay === "Clouds") {

        LocalWeatherNextDayDiv.addClass("fas fa-cloud fa-2x")

    } else if (LocalWeatherNextDay === "Rain") {

        LocalWeatherNextDayDiv.addClass("fas fa-cloud-rain fa-2x")

    } else if (LocalWeatherNextDay === "Clear") {

        LocalWeatherNextDayDiv.addClass("far fa-sun fa-2x")

    } else if (LocalWeatherNextDay === "Snow") {

        LocalWeatherNextDayDiv.addClass("fas fa-snowflake fa-2x")


```

```

    } else if (LocalWeatherNextDay === "Mist") {
        LocalWeatherNextDayDiv.addClass("fas fa-cloud-rain fa-2x")
    } else if (LocalWeatherNextDay === "Smog") {
        LocalWeatherNextDayDiv.addClass("fas fa-smog fa-2x")
    }

    // Temperature

    var localFNextDay = localStorage.getItem("fNextDay")

    console.log(localFNextDay)

    var LocalNextDayTempCDiv = $("<div>")

        LocalNextDayTempCDiv.text("Temp:  " + localFNextDay + "
°F").addClass("tempClass")

    // Humidity

    var LocalNextDayHumidity = localStorage.getItem("nextDayHumidity")

    var LocalNextDayHumidityDiv = $("<div>")

        LocalNextDayHumidityDiv.text("Humidity: " + LocalNextDayHumidity +
"%").addClass("humidity")

    localNextDayDiv.append(LocalNextDayDateDiv, LocalWeatherNextDayDiv,
LocalNextDayTempCDiv, LocalNextDayHumidityDiv)

    $("#nextDay").append(localNextDayDiv)

```

```
}
```

```
//For day two
```

```
var localDayTwoDiv = $("<div>")
```

```
        if    (localStorage.getItem("dayTwoDate")    !==    null    &&  
localStorage.getItem("dayTwoWeather")    !==    null    &&  
localStorage.getItem("dayTwoFixedTemp")    !==    null    &&  
localStorage.getItem("dayTwoHumidity") !== null) {
```

```
    // Get Date
```

```
    var localDayTwoDate = localStorage.getItem("dayTwoDate")
```

```
    var LocalDayTwoDateDiv = $("<div>").addClass("head")
```

```
    LocalDayTwoDateDiv.text(localDayTwoDate)
```

```
    //Get weather
```

```
    var LocalWeatherDayTwo = localStorage.getItem("dayTwoWeather")
```

```
    var LocalWeatherDayTwoDiv = $("<div>")
```

```
    // If statement for weather
```

```
    if (LocalWeatherDayTwo === "Clouds") {
```

```
        LocalWeatherDayTwoDiv.addClass("fas fa-cloud fa-2x")
```

```

} else if (LocalWeatherDayTwo === "Rain") {
    LocalWeatherDayTwoDiv.addClass("fas fa-cloud-rain fa-2x")
} else if (LocalWeatherDayTwo === "Clear") {
    LocalWeatherDayTwoDiv.addClass("far fa-sun fa-2x")
} else if (LocalWeatherDayTwo === "Snow") {
    LocalWeatherDayTwoDiv.addClass("fas fa-snowflake fa-2x")
} else if (LocalWeatherDayTwo === "Mist") {
    LocalWeatherDayTwoDiv.addClass("fas fa-cloud-rain fa-2x")
} else if (LocalWeatherDayTwo === "Haze") {
    LocalWeatherDayTwoDiv.addClass("fas fa-smog fa-2x")
}

```

```

// Temperature

```

```

var localFDayTwo = localStorage.getItem("dayTwoFixedTemp")

```

```

var LocalDayTwoTempDiv = $("<div>")

```

```

    LocalDayTwoTempDiv.text("Temp:    " + localFDayTwo + "
°F").addClass("tempClass")

```

```

// Humidity

```

```

var LocalDayTwoHumidity = localStorage.getItem("dayTwoHumidity")

```

```

var LocalDayTwoHumidityDiv = $("<div>")

```



```
LocalDayTwoHumidityDiv.text("Humidity: " + LocalDayTwoHumidity +
"%").addClass("humidity")
```

```
// Appending to page
```

```
localDayTwoDiv.append(LocalDayTwoDateDiv, LocalWeatherDayTwoDiv,
LocalDayTwoTempDiv, LocalDayTwoHumidityDiv)

$("#dayTwo").append(localDayTwoDiv)

}
```

```
//For day three
```

```
var localDayThreeDiv = $("<div>")

if (localStorage.getItem("dayThreeDate") !== null &&
localStorage.getItem("dayThreeWeather") !== null &&
localStorage.getItem("dayThreeTempFixed") !== null &&
localStorage.getItem("dayThreeHumidity") !== null) {
```

```
// Get Date
```

```
var localDayThreeDate = localStorage.getItem("dayThreeDate")

var LocalDayThreeDateDiv = $("<div>").addClass("head")

LocalDayThreeDateDiv.text(localDayThreeDate)
```

```
//Get weather

var LocalWeatherDayThree = localStorage.getItem("dayThreeWeather")

var LocalWeatherDayThreeDiv = $("<div>")


// If statement for weather

if (LocalWeatherDayThree === "Clouds") {

    LocalWeatherDayThreeDiv.addClass("fas fa-cloud fa-2x")

} else if (LocalWeatherDayThree === "Rain") {

    LocalWeatherDayThreeDiv.addClass("fas fa-cloud-rain fa-2x")

} else if (LocalWeatherDayThree === "Clear") {

    LocalWeatherDayThreeDiv.addClass("far fa-sun fa-2x")

} else if (LocalWeatherDayThree === "Snow") {

    LocalWeatherDayThreeDiv.addClass("fas fa-snowflake fa-2x")

} else if (LocalWeatherDayThree === "Mist") {

    LocalWeatherDayThreeDiv.addClass("fas fa-cloud-rain fa-2x")

} else if (LocalWeatherDayThree === "Haze") {

    LocalWeatherDayThreeDiv.addClass("fas fa-smog fa-2x")

}


// Temperature

var localFDayThree = localStorage.getItem("dayThreeTempFixed")
```

```

var LocalDayThreeTempDiv = $("<div>")

    LocalDayThreeTempDiv.text("Temp:  " + localFDayThree + "
°F").addClass("tempClass")

// Humidity

var LocalDayThreeHumidity = localStorage.getItem("dayThreeHumidity")
var LocalDayThreeHumidityDiv = $("<div>")

    LocalDayThreeHumidityDiv.text("Humidity: " + LocalDayThreeHumidity +
"%").addClass("humidity")

// Appending to page

        localDayThreeDiv.append(LocalDayThreeDateDiv,
LocalWeatherDayThreeDiv,                LocalDayThreeTempDiv,
LocalDayThreeHumidityDiv)

    $("#dayThree").append(localDayThreeDiv)
}

//For day four

var localDayFourDiv = $("<div>")

        if    (localStorage.getItem("dayFourDate")    !==    null    &&
localStorage.getItem("dayFourWeather")                !==    null    &&

```

```
localStorage.getItem("dayFourFixedTemp") !== null &&  
localStorage.getItem("dayFourHumidity") !== null) {
```

```
    // Get Date
```

```
    var localDayFourDate = localStorage.getItem("dayFourDate")
```

```
    var LocalDayFourDateDiv = $("

").addClass("head")


```

```
    LocalDayFourDateDiv.text(localDayFourDate)
```

```
    //Get weather
```

```
    var LocalWeatherDayFour = localStorage.getItem("dayFourWeather")
```

```
    var LocalWeatherDayFourDiv = $("

")


```

```
    // If statement for weather
```

```
    if (LocalWeatherDayFour === "Clouds") {
```

```
        LocalWeatherDayFourDiv.addClass("fas fa-cloud fa-2x")
```

```
    } else if (LocalWeatherDayFour === "Rain") {
```

```
        LocalWeatherDayFourDiv.addClass("fas fa-cloud-rain fa-2x")
```

```
    } else if (LocalWeatherDayFour === "Clear") {
```

```
        LocalWeatherDayFourDiv.addClass("far fa-sun fa-2x")
```

```
    } else if (LocalWeatherDayFour === "Snow") {
```

```
        LocalWeatherDayFourDiv.addClass("fas fa-snowflake fa-2x")
```

```

    } else if (LocalWeatherDayFour === "Mist") {
        LocalWeatherDayFourDiv.addClass("fas fa-cloud-rain fa-2x")
    } else if (LocalWeatherDayFour === "Haze") {
        LocalWeatherDayFourDiv.addClass("fas fa-smog fa-2x")
    }

    // Temperature
    var localFDayFour = localStorage.getItem("dayFourFixedTemp")
    var LocalDayFourTempDiv = $("<div>")
        LocalDayFourTempDiv.text("Temp:  " + localFDayFour + "
°F").addClass("tempClass")

    // Humidity
    var LocalDayFourHumidity = localStorage.getItem("dayFourHumidity")
    var LocalDayFourHumidityDiv = $("<div>")
        LocalDayFourHumidityDiv.text("Humidity: " + LocalDayFourHumidity +
"%").addClass("humidity")

    // Appending to page
    localDayFourDiv.append(LocalDayFourDateDiv, LocalWeatherDayFourDiv,
LocalDayFourTempDiv, LocalDayFourHumidityDiv)

```

```
$("#dayFour").append(localDayFourDiv)
}
```

```
//For day five
```

```
var localDayFiveDiv = $("<div>")
    if (localStorage.getItem("dayFiveDate") !== null &&
localStorage.getItem("dayFiveWeather") !== null &&
localStorage.getItem("dayFiveFixedTemp") !== null &&
localStorage.getItem("dayFiveHumidity") !== null) {
```

```
// Get Date
```

```
var localDayFiveDate = localStorage.getItem("dayFiveDate")
```

```
var LocalDayFiveDateDiv = $("<div>").addClass("head")
```

```
LocalDayFiveDateDiv.text(localDayFiveDate)
```

```
//Get weather
```

```
var LocalWeatherDayFive = localStorage.getItem("dayFiveWeather")
```

```
var LocalWeatherDayFiveDiv = $("<div>")
```

```
// If statement for weather
```

```
if (LocalWeatherDayFive === "Clouds") {
```

```

        LocalWeatherDayFiveDiv.addClass("fas fa-cloud fa-2x")
    } else if (LocalWeatherDayFive === "Rain") {
        LocalWeatherDayFiveDiv.addClass("fas fa-cloud-rain fa-2x")
    } else if (LocalWeatherDayFive === "Clear") {
        LocalWeatherDayFiveDiv.addClass("far fa-sun fa-2x")
    } else if (LocalWeatherDayFive === "Snow") {
        LocalWeatherDayFiveDiv.addClass("fas fa-snowflake fa-2x")
    } else if (LocalWeatherDayFive === "Mist") {
        LocalWeatherDayFiveDiv.addClass("fas fa-cloud-rain fa-2x")
    } else if (LocalWeatherDayFive === "Haze") {
        LocalWeatherDayFiveDiv.addClass("fas fa-smog fa-2x")
    }

// Temperature
var localFDayFive = localStorage.getItem("dayFiveFixedTemp")
var LocalDayFiveTempDiv = $("<div>")

        LocalDayFiveTempDiv.text("Temp:    " + localFDayFive + "
°F").addClass("tempClass")

// Humidity
var LocalDayFiveHumidity = localStorage.getItem("dayFiveHumidity")

```

```
var LocalDayFiveHumidityDiv = $("<div>")

    LocalDayFiveHumidityDiv.text("Humidity: " + LocalDayFiveHumidity +
"%").addClass("humidity")

// Appending to page

    localDayFiveDiv.append(LocalDayFiveDateDiv, LocalWeatherDayFiveDiv,
LocalDayFiveTempDiv, LocalDayFiveHumidityDiv)

    $("#dayFive").append(localDayFiveDiv)
}
}

//On button click call function callCity
$("#add-city").on("click", function () {
    event.preventDefault()

    //setting location
    var location = $("#city-input").val().trim();
    callCity(location)

})
```



```
//Main function of application

function callCity(location) {

    //emptying day forecast div
    $("#dayForecast").empty()

    //making location input capitalized
    var firstLetter = location.charAt(0).toUpperCase()
    var restWord = location.slice(1)
    let locationC = (firstLetter + restWord)

    //pushing location to pastSearches array
    pastSearches.push(locationC)
    localStorage.setItem("pastSearches", pastSearches)
    $(".list").empty()

    // for loop for appending list items
    for (i = 0; i < pastSearches.length; i++) {
        let searchItem = $("- ").attr("id", "listItem")
        searchItem.text(pastSearches[i])
        $(".list").prepend(searchItem)
    }
}

```

```
}
```

```
//Api call for initial day
```

```
var APIKey = "20c488e0a9aff750eabd58301c43b3ce"
```

```
    var queryURL = "https://api.openweathermap.org/data/2.5/weather?q=" +  
location + "&appid=" + APIKey;
```

```
$.ajax({
```

```
    url: queryURL,
```

```
    method: "GET"
```

```
}).then(function (response) {
```

```
    var lat = response.coord.lat
```

```
    var lon = response.coord.lon
```

```
//Api call for uv index
```

```
var APIKey = "20c488e0a9aff750eabd58301c43b3ce"
```

```
var queryURL =
```

```
`https://api.openweathermap.org/data/2.5/uv?appid=${APIKey}&lat=${lat}&lon=  
${lon}`
```

```
console.log(queryURL)
```

```
$.ajax({
```

```
url: queryURL,  
method: "GET"  
}).then(function (uVResponse) {  
  
    //creating initial div and city div , and setting their content to local storage  
    var weatherDiv = $("<div>")  
    var cityDiv = $("<div>").addClass("bigger")  
    var city = (response.name)  
    localStorage.setItem("InitialCity", city)  
  
    //creating span for font awesome emoji  
    var weather = (response.weather[0].main)  
    var weatherAnimation = $("<span>")  
    localStorage.setItem("initialWeather", weather)  
  
    // applying class to weather animation span  
    if (weather === "Clouds") {  
        weatherAnimation.addClass("fas fa-cloud fa-2x")  
    } else if (weather === "Rain") {  
        weatherAnimation.addClass("fas fa-cloud-rain fa-2x")  
    } else if (weather === "Clear") {
```

```
        weatherAnimation.addClass("far fa-sun fa-2x")
    } else if (weather === "Snow") {
        weatherAnimation.addClass("fas fa-snowflake fa-2x")
    } else if (weather === "Mist") {
        weatherAnimation.addClass("fas fa-cloud-rain fa-2x")
    } else if (weather === "Haze") {
        weatherAnimation.addClass("fas fa-smog fa-2x")
    }
    cityDiv.text(city + " " + date + " ")
    cityDiv.append(weatherAnimation)
```

```
//temperature
```

```
var k = (response.main.temp)
var f = (k - 273.5) * 1.80 + 32
var tempDiv = $("<div>")
tempDiv.text("Temperature : " + f.toFixed(2) + "F").addClass("tempClass")
localStorage.setItem("initialTempFixed", f.toFixed(2))
```

```
// Humidity
```

```
var humidity = (response.main.humidity)
var humidityDiv = $("<div>")
```

```
humidityDiv.text("Humidity : " + humidity + " %").addClass("humidity")

localStorage.setItem("initialHumidity", humidity)


// Wind Speed

var windSpeed = (response.wind.speed)

var windSpeedDiv = $("<div>")

        windSpeedDiv.text("Wind Speed : " + windSpeed + "
MPH").addClass("windSpeed")

localStorage.setItem("initialWindSpeed", windSpeed)


//UV Index

var uvIndexDiv = $("<div>")

var uvIndex = uVResponse.value

uvIndexDiv.text("UV Index: ").addClass("UvIndex")

var uvIndexSpan = $("<span>")

uvIndexSpan.text(uvIndex)

uvIndexDiv.append(uvIndexSpan)

localStorage.setItem("initalUvIndex", uvIndex)


//Setting color for uv index span based on number value

if (uvIndex <= 3) {
```

```
        uvIndexSpan.addClass("low")
    } else if (uvIndex > 7.01) {
        uvIndexSpan.addClass("high")
    } else if (3.1 <= uvIndex <= 6.9) {
        uvIndexSpan.addClass("medium")
    }
```

```
//Appending to page
```

```
        $(weatherDiv).append(cityDiv, tempDiv, humidityDiv, windSpeedDiv,
uvIndexDiv)

        $("#dayForecast").append(weatherDiv)
    })
})
```

```
// 5 day api call
```

```
var APIKey = "20c488e0a9aff750eabd58301c43b3ce"
```

```
    var queryURL = "https://api.openweathermap.org/data/2.5/forecast?q=" +
location + "&appid=" + APIKey
```

```
    console.log(queryURL)
```

```
$.ajax({
```

```
    url: queryURL,
```

```
method: "GET"

}).then(function (response) {

    // Emptying divs when new city is called

    $("#nextDay").empty()

    $("#dayTwo").empty()

    $("#dayThree").empty()

    $("#dayFour").empty()

    $("#dayFive").empty()


    //Next Day

    var nextDayDiv = $("<div>")


    //Date

    var nextDayDate = moment().add(1, 'days').format("L")

    localStorage.setItem("nextDayDate", nextDayDate)

    var nextDayDateDiv = $("<div>").addClass("head")

    nextDayDateDiv.text(nextDayDate)


    //Set weather

    var weatherNextDay = (response.list[4].weather[0].main)
```

```
localStorage.setItem("weatherNextDay", weatherNextDay)
```

```
var weatherNextDayDiv = $("
```

```
// If statement for weatherDiv
```

```
if (weatherNextDay === "Clouds") {
```

```
    weatherNextDayDiv.addClass("fas fa-cloud fa-2x")
```

```
} else if (weatherNextDay === "Rain") {
```

```
    weatherNextDayDiv.addClass("fas fa-cloud-rain fa-2x")
```

```
} else if (weatherNextDay === "Clear") {
```

```
    weatherNextDayDiv.addClass("far fa-sun fa-2x")
```

```
} else if (weatherNextDay === "Snow") {
```

```
    weatherNextDayDiv.addClass("fas fa-snowflake fa-2x")
```

```
} else if (weatherNextDay === "Mist") {
```

```
    weatherNextDayDiv.addClass("fas fa-cloud-rain fa-2x")
```

```
} else if (weatherNextDay === "Haze") {
```

```
    weatherNextDayDiv.addClass("fas fa-smog fa-2x")
```

```
}
```

```
// Temperature
```

```
var nextDayTempC = (response.list[4].main.temp)
```

```
var fNextDay = (nextDayTempC - 273.5) * 1.80 + 32
```



```
localStorage.setItem("fNextDay", fNextDay.toFixed(2))

var nextDayTempCDiv = $("<div>")

    nextDayTempCDiv.text("Temp:  " + fNextDay.toFixed(2) + "
°F").addClass("tempClass")


// Humidity

var nextDayHumidity = (response.list[4].main.humidity)

localStorage.setItem("nextDayHumidity", nextDayHumidity)

var nextDayHumidityDiv = $("<div>")

    nextDayHumidityDiv.text("Humidity:  " + nextDayHumidity +
"%").addClass("humidity")


//Appending to page

    nextDayDiv.append(nextDayDateDiv,  weatherNextDayDiv,
nextDayTempCDiv, nextDayHumidityDiv)

$("#nextDay").append(nextDayDiv)


// Day 2

var dayTwoDiv = $("<div>")


//Date
```

```
var dayTwoDate = moment().add(2, 'days').format("L")
```

```
var dayTwoDateDiv = $("

").addClass("head")


```

```
localStorage.setItem("dayTwoDate", dayTwoDate)
```

```
dayTwoDateDiv.text(dayTwoDate)
```

```
//Set weather
```

```
var dayTwoWeather = (response.list[12].weather[0].main)
```

```
var dayTwoWeatherDiv = $("

")


```

```
localStorage.setItem("dayTwoWeather", dayTwoWeather)
```

```
//If statement for weatherDiv
```

```
if (dayTwoWeather === "Clouds") {
```

```
    dayTwoWeatherDiv.addClass("fas fa-cloud fa-2x")
```

```
} else if (dayTwoWeather === "Rain") {
```

```
    dayTwoWeatherDiv.addClass("fas fa-cloud-rain fa-2x")
```

```
} else if (dayTwoWeather === "Clear") {
```

```
    dayTwoWeatherDiv.addClass("far fa-sun fa-2x")
```

```
} else if (dayTwoWeather === "Snow") {
```

```
    dayTwoWeatherDiv.addClass("fas fa-snowflake fa-2x")
```

```
} else if (dayTwoWeather === "Mist") {
```

```
    dayTwoWeatherDiv.addClass("fas fa-cloud-rain fa-2x")
```

```

    } else if (dayTwoWeather === "Haze") {
        dayTwoWeatherDiv.addClass("fas fa-smog fa-2x")
    }

    // Temperature
    var dayTwoTempC = (response.list[12].main.temp)
    var fTwo = (dayTwoTempC - 273.5) * 1.80 + 32
    var dayTwoTempCDiv = $("<div>")
    var dayTwoFixedTemp = fTwo.toFixed(2)
    localStorage.setItem("dayTwoFixedTemp", dayTwoFixedTemp)

    dayTwoTempCDiv.text("Temp:    " + dayTwoFixedTemp + "
°F").addClass("tempClass")

    // Humidity
    var dayTwoHumidity = (response.list[12].main.humidity)
    var dayTwoHumidityDiv = $("<div>")

    dayTwoHumidityDiv.text("Humidity:    " + dayTwoHumidity +
"%").addClass("humidity")

    localStorage.setItem("dayTwoHumidity", dayTwoHumidity)

    //Appending to page

```

```
    dayTwoDiv.append(dayTwoDateDiv, dayTwoWeatherDiv, dayTwoTempCDiv,  
dayTwoHumidityDiv)
```

```
$("#dayTwo").append(dayTwoDiv)
```

```
// Day 3
```

```
var dayThreeDiv = $("<div>")
```

```
//Date
```

```
var dayThreeDate = moment().add(3, 'days').format("L")
```

```
var dayThreeDateDiv = $("<div>").addClass("head")
```

```
dayThreeDateDiv.text(dayThreeDate)
```

```
localStorage.setItem("dayThreeDate", dayThreeDate)
```

```
//Set weather
```

```
var dayThreeWeather = (response.list[20].weather[0].main)
```

```
var dayThreeWeatherDiv = $("<div>")
```

```
localStorage.setItem("dayThreeWeather", dayThreeWeather)
```

```
//If statement for weatherDiv
```

```
if (dayThreeWeather === "Clouds") {
```

```
    dayThreeWeatherDiv.addClass("fas fa-cloud fa-2x")
```

```
} else if (dayThreeWeather === "Rain") {  
    dayThreeWeatherDiv.addClass("fas fa-cloud-rain fa-2x")  
} else if (dayThreeWeather === "Clear") {  
    dayThreeWeatherDiv.addClass("far fa-sun fa-2x")  
} else if (dayThreeWeather === "Snow") {  
    dayThreeWeatherDiv.addClass("fas fa-snowflake fa-2x")  
} else if (dayThreeWeather === "Mist") {  
    dayThreeWeatherDiv.addClass("fas fa-cloud-rain fa-2x")  
} else if (dayThreeWeather === "Haze") {  
    dayThreeWeatherDiv.addClass("fas fa-smog fa-2x")  
}
```

```
// Temperature
```

```
var dayThreeTempC = (response.list[20].main.temp)
```

```
var fThree = (dayThreeTempC - 273.5) * 1.80 + 32
```

```
var dayThreeTempCDiv = $("
```

```
    dayThreeTempCDiv.text("Temp:    " + fThree.toFixed(2) + "  
°F").addClass("tempClass")
```

```
localStorage.setItem("dayThreeTempFixed", fThree.toFixed(2))
```

```
// Humidity
```

```
var dayThreeHumidity = (response.list[20].main.humidity)

var dayThreeHumidityDiv = $("<div>")

    dayThreeHumidityDiv.text("Humidity:  " + dayThreeHumidity +
"%").addClass("humidity")

localStorage.setItem("dayThreeHumidity", dayThreeHumidity)


//Appending to page

    dayThreeDiv.append(dayThreeDateDiv,  dayThreeWeatherDiv,
dayThreeTempCDiv, dayThreeHumidityDiv)

$("#dayThree").append(dayThreeDiv)


// Day 4

var dayFourDiv = $("<div>")


//date

var dayFourDate = moment().add(4, 'days').format("L")
var dayFourDateDiv = $("<div>").addClass("head")
dayFourDateDiv.text(dayFourDate)
localStorage.setItem("dayFourDate", dayFourDate)
```

```
//Set weather

var dayFourWeather = (response.list[28].weather[0].main)

var dayFourWeatherDiv = $("<div>")

localStorage.setItem("dayFourWeather", dayFourWeather)


// If statement for weatherDiv

if (dayFourWeather === "Clouds") {

    dayFourWeatherDiv.addClass("fas fa-cloud fa-2x fa-2x")

} else if (dayFourWeather === "Rain") {

    dayFourWeatherDiv.addClass("fas fa-cloud-rain fa-2x fa-2x")

} else if (dayFourWeather === "Clear") {

    dayFourWeatherDiv.addClass("far fa-sun fa-2x fa-2x")

} else if (dayFourWeather === "Snow") {

    dayFourWeatherDiv.addClass("fas fa-snowflake fa-2x fa-2x")

} else if (dayFourWeather === "Mist") {

    dayFourWeatherDiv.addClass("fas fa-cloud-rain fa-2x")

} else if (dayFourWeather === "Haze") {

    dayFourWeatherDiv.addClass("fas fa-smog fa-2x")

}


// Temperature
```

```

var dayFourTempC = (response.list[28].main.temp)

var fFour = (dayFourTempC - 273.5) * 1.80 + 32

var dayFourTempCDiv = $("<div>")

        dayFourTempCDiv.text("Temp:    " + fFour.toFixed(2) + "
°F").addClass("tempClass")

localStorage.setItem("dayFourFixedTemp", fFour.toFixed(2))


// Humidity

var dayFourHumidity = (response.list[28].main.humidity)

var dayFourHumidityDiv = $("<div>")

        dayFourHumidityDiv.text("Humidity:    " + dayFourHumidity +
"%").addClass("humidity")

localStorage.setItem("dayFourHumidity", dayFourHumidity)


// Appending to page

        dayFourDiv.append(dayFourDateDiv,    dayFourWeatherDiv,
dayFourTempCDiv, dayFourHumidityDiv)

$("#dayFour").append(dayFourDiv)


// Day 5

var dayFiveDiv = $("<div>")

```



```
//date

var dayFiveDate = moment().add(5, 'days').format("L")
var dayFiveDateDiv = $("

").addClass("head")
dayFiveDateDiv.text(dayFiveDate)
localStorage.setItem("dayFiveDate", dayFiveDate)


```

```
//Set weather

var dayFiveWeather = (response.list[36].weather[0].main)
var dayFiveWeatherDiv = $("

")
localStorage.setItem("dayFiveWeather", dayFiveWeather)


```

```
//If Statement for Weather

if (dayFiveWeather === "Clouds") {
    dayFiveWeatherDiv.addClass("fas fa-cloud fa-2x")
} else if (dayFiveWeather === "Rain") {
    dayFiveWeatherDiv.addClass("fas fa-cloud-rain fa-2x")
} else if (dayFiveWeather === "Clear") {
    dayFiveWeatherDiv.addClass("far fa-sun fa-2x")
} else if (dayFiveWeather === "Snow") {
```

```

        dayFiveWeatherDiv.addClass("fas fa-snowflake fa-2x")
    } else if (dayFiveWeather === "Mist") {
        dayFiveWeatherDiv.addClass("fas fa-cloud-rain fa-2x")
    } else if (dayFiveWeather === "Haze") {
        dayFiveWeatherDiv.addClass("fas fa-smog fa-2x")
    }

// Temperature
var dayFiveTempC = (response.list[36].main.temp)
var fFive = (dayFiveTempC - 273.5) * 1.80 + 32
var dayFiveTempCDiv = $("<div>")

        dayFiveTempCDiv.text("Temp:    " + fFive.toFixed(2) + "
°F").addClass("tempClass")

localStorage.setItem("dayFiveFixedTemp", fFive.toFixed(2))

// Humidity
var dayFiveHumidity = (response.list[36].main.humidity)
var dayFiveHumidityDiv = $("<div>")

        dayFiveHumidityDiv.text("Humidity:    " + dayFiveHumidity +
"%").addClass("humidity")

localStorage.setItem("dayFiveHumidity", dayFiveHumidity)

```

```
        //Appending to page
        dayFiveDiv.append(dayFiveDateDiv, dayFiveWeatherDiv,
dayFiveTempCDiv, dayFiveHumidityDiv)
        $("#dayFive").append(dayFiveDiv)
    })
}
```

```
// On click event for list
```

```
$(document).on("click","li", function () {
    event.preventDefault()
    console.log("clicked")

    var listLocation = $(this).text()
    console.log("TCL: listLocation", listLocation)

    let location = listLocation
    callCity(location)
})
```

home.css

```
body{  
    background-image: url('drops.jpg');  
    background-size:cover;  
}  
  
#title{  
    margin-right: 11vw;  
    width: 5vw;  
}  
  
.nav{  
    margin:0 auto;  
    float: left;  
    list-style-type:none;  
    width: 11vw ;  
    height: 2vw;  
    background-color: rgb(10, 73, 112,0.7);  
    color: white;  
    padding: 0.5%;  
}
```

```
header{  
    display: block ;  
    width: 100%;  
    height: 7vh;  
  
}  
  
a{  
    text-decoration: none;  
}  
  
a:link{  
    color: inherit;  
}  
  
a:link:hover{  
    color: inherit;  
}  
  
a:visited{  
    color: inherit;  
}  
  
a:visited:hover{
```

```
    color: inherit;
}

.nav:hover,.nav:active{

    background-color: #d3d3d3;


    color: black;
}

.head1{


    color: rgb(10, 73, 112);
    font-family: 'Century Gothic';
    font-size: 60px;
    width: 40%;
    margin-left: 30%;
    text-align: center;
    margin-top: 2vh;
    padding: 5px;
}

.body1{

    background-color:rgb(255, 255, 255,0.8);
    background-origin: padding-box;
```

```
    color: #3d3d3d;

    font-size: x-large;

    width: 75%;

    margin-left: 12.5%;

    margin-top: 10vh;

    text-align: center;

    padding: 35px;

    background-blend-mode: lighten;

}

.navbar{

    margin-left: 20%;

}

*{

    font-family: 'Raleway', sans-serif;

    /*font-family: 'Roboto', sans-serif;*/

}

footer {
```

```
display: flex;
justify-content: center;
padding: 5px;
margin-top: 43vh;
background-color:rgb(10, 73, 112);
color: #fff;
}
```

```
.header1 img {
float: left;
width: 100px;
height: 100px;
background: #555;
margin-left: 5%;
}
```

```
app1.py
```

```
from pyexpat import features, model
```

```
import numpy
```

```
import pickle
```

```
from sklearn.preprocessing import LabelEncoder
```

```
import pandas as pd
```



```
from flask import Flask, request, jsonify, render_template, redirect, url_for
import requests
import json
```

```
# NOTE: you must manually set API_KEY below using information retrieved
from your IBM Cloud account.
```

```
API_KEY = "PQBr9MBF7mFuSh2VVLfOE-liIA04VH-h5VEk8EfjFIuw"
```

```
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-
type:apikey'})
```

```
mltoken = token_response.json()["access_token"]
```

```
print("ML Token",mltoken)
```

```
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
```

```
# Declare a Flask app
```

```
app = Flask(__name__,template_folder='template')
```

```
model = pickle.load(open("rainfall.pkl",'rb'))
```

```
scale = pickle.load(open("scale.pkl", 'rb'))
```

```
@app.route('/')
```

```
def home():
```

```
    return render_template("home.html")
```

```
@app.route('/chance/', methods=['GET', 'POST'])
```

```
def chance():
```

```
    return render_template("chance.html")
```

```
@app.route('/nochance/', methods=['GET', 'POST'])
```

```
def nochance():
```

```
    return render_template("noChance.html")
```

```
@app.route('/help/')
```

```
def help():
```

```
    return render_template("help.html")
```

```
@app.route('/contact/')
```

```
def contact():
```

```
return render_template("contact.html")
```

```
@app.route('/about/')
```

```
def about():
```

```
    return render_template("about.html")
```

```
@app.route('/predict',methods=['POST','GET'])
```

```
def predict():
```

```
    res = " "
```

```
    # If a form is submitted
```

```
    if request.method == "POST":
```

```
        Location = request.form.get('Location')
```

```
        MinTemp = request.form['MinTemp']
```

```
        MaxTemp = request.form['MaxTemp']
```

```
        Rainfall = request.form['Rainfall']
```

```
        WindGustSpeed = request.form['WindGustSpeed']
```

```
        WindSpeed9am = request.form['WindSpeed9am']
```

```
        WindSpeed3pm = request.form['WindSpeed3pm']
```

```
        Humidity9am = request.form['Humidity9am']
```

```
        Humidity3pm = request.form['Humidity3pm']
```

```
        Pressure9am = request.form['Pressure9am']
```

```

Pressure3pm = request.form['Pressure3pm']
Temp9am = request.form['Temp9am']
Temp3pm = request.form['Temp3pm']
RainToday = request.form.get('RainToday')
WindGustDir = request.form.get('WindGustDir')
WindDir9am = request.form.get('WindDir9am')
WindDir3pm = request.form.get('WindDir3pm')

```

```

new_row =
{'Location':Location,'MinTemp':MinTemp,'MaxTemp':MaxTemp,'Rainfall':Rainfall
,'WindGustSpeed':WindGustSpeed,'WindSpeed9am':WindSpeed9am,'WindSpeed3
pm':WindSpeed3pm,'Humidity9am':Humidity9am,'Humidity3pm':Humidity3pm,'P
ressure9am':Pressure9am,'Pressure3pm':Pressure3pm,'Temp9am':Temp9am,'Temp3
pm':Temp3pm,'RainToday':RainToday,'WindGustDir':WindGustDir,'WindDir9am':
WindDir9am,'WindDir3pm':WindDir3pm}

print(new_row)

```

```

new_df =
pd.DataFrame(columns=['Location','MinTemp','MaxTemp','Rainfall','WindGustSpe
ed','WindSpeed9am','WindSpeed3pm','Humidity9am','Humidity3pm','Pressure9am'
,'Pressure3pm','Temp9am','Temp3pm','RainToday','WindGustDir','WindDir9am','W
indDir3pm'])

```

```

new_df = new_df.append(new_row,ignore_index=True)

labeled =

new_df[['Location','MinTemp','MaxTemp','Rainfall','WindGustSpeed','WindSpeed9
am','WindSpeed3pm','Humidity9am','Humidity3pm','Pressure9am','Pressure3pm','T
emp9am','Temp3pm','RainToday','WindGustDir','WindDir9am','WindDir3pm']]

X = labeled.values

print(X)

payload_scoring = {"input_data": [{"field":
[['Location','MinTemp','MaxTemp','Rainfall','WindGustSpeed','WindSpeed9am','Wi
ndSpeed3pm','Humidity9am','Humidity3pm','Pressure9am','Pressure3pm','Temp9a
m','Temp3pm','RainyToday','WindGustDir','WindDir9am','WindDir3pm']],
"values": X.tolist()}}}

response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/73230b85-51ea-45db-baa7-
e86b5d528fbe/predictions?version=2022-11-14',
json=payload_scoring,headers={'Authorization': 'Bearer ' + mltoken})

print("Scoring response")

predictions = response_scoring.json()

print(predictions)

output = predictions['predictions'][0]['values'][0][0]

```

```
    print(output)
```

```
else:
```

```
    output = ""
```

```
if output == 1:
```

```
    return redirect(url_for('chance'))
```

```
elif output == 0:
```

```
    return redirect(url_for('nochance'))
```

```
return render_template("index.html", output = res)
```

```
#Running the app
```

```
if __name__ == "__main__":
```

```
    app.run(debug = True,host='0.0.0.0',port=80)
```

## 8.TESTING:

### 8.1 Test cases

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Executed By
LoginPage_TC_001	UI	Home Page	Verify user is login by entering email,password.and confirming password.	1.Enter URL and click go 2.Enter the email id, password and confirm password. 3.click the login button.	<a href="https://rainfalldata.w3spaces.com">https://rainfalldata.w3spaces.com</a>	Login/ registering for the application	Working as expected	Pass	Malhusudhan
LoginPage_TC_002	UI	Home Page	Verify the can access the dashboard with the LinkedIn login.	1. Enter the URL and click enter 2. enter the valid mail id in the Email text box. 3. enter the valid password in the password text box. 4. click on the join now button in linked in.	<a href="https://rainfalldata.w3spaces.com/">https://rainfalldata.w3spaces.com/</a>	Application should show below UI elements: a. email text box b. password text box c. join now button d. shows the dashboard page	Working as expected	pass	Vishnudev
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials and get the confirmation mail.	1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login and get mail.	Username: ibmmsec@gmail.com password: Testing123	Application should send the confirmation mail	Working as expected	Pass	Mohammedasath
Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Executed By
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Valid credentials	1.Enter URL( <a href="https://shopenzer.com/">https://shopenzer.com/</a> ) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: ibmmsec@gmail.com password: Testing123	User should navigate to the home page.	Working as expected	Pass	Mohamed Abhuthahir Khan
LoginPage_TC_005	Functional	Login page	Verify user is able to log into application with Invalid credentials	1.Enter URL( <a href="https://shopenzer.com/">https://shopenzer.com/</a> ) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: chalam@gmail.com password: Testing123678686786876876	Application should show 'Incorrect email or password' validation message.	Working as expected	pass	Malhusudhan
LoginPage_TC_006	Functional	Login page	Verify user is able to log into application with Invalid credentials	1.Enter URL( <a href="https://shopenzer.com/">https://shopenzer.com/</a> ) and click go 2.Click on My Account dropdown button 3.Enter Invalid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: ibmmsec@gmail.com password: Testing654	Application should show 'Incorrect email or password' validation message.	Working as expected	pass	Vishnudev

### 8.2 USER ACCEPTANCE TESTING:

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

## 9. RESULTS

### 9.1 Performance Metrics:

For rainfall prediction model, applying different machine learning algorithm to train and test the model, the xgboost machine learning algorithm shows the higher accuracy score compare to other machine learning algorithm like decison tree, Naive Bayes, Radom Forest,etc and saving the xgboost model as pickle for future use For sowing crop recommendation model, applying different machine learning algorithm to train and test the model, the Random Forest Classification machine learning algorithm shows the higher accuracy score compare to other machine learning algorithm like decison tree, Naive Bayes, xgboost,etc and saving the random forest model for future use Then use this saved model, using flask create a responsive web page to access the prediction and test the flask with postman appication with multiple user, it show good testing results



## **10.ADVANTAGES AND DISADVANTAGES:**

### **10.1 Advantages:**

- \* Farmers can analyse the rainfall amount. According to that the farmer can cultivate.

- \* Farmers obtain more benefit without any loss in the crop cultivation.

- \* Farmers can obtain more benefit.

- \* Water resources can be managed efficiently by using rainfall prediction system.

### **10.2 Disadvantages:**

- \* Some times the prediction may inaccurate.

- \* Sudden changes in may climate may lead to more rainfall which lead to loss in the crop.

## **11.CONCLUSION:**

The overall aim is to define various ML techniques that are useful in predicting rainfall. The goal of this research is to design accurate and efficient model by applying lesser number of attributes and tests. Firstly, the data is pre-processed and then it is used in the model. K-Nearest N and Random Forest classifier with approximately 88% are the most efficient classification algorithms. However, Decision Tree classifier gives the least accuracy. We can further expand this research covering other ML techniques such as time series, clustering and association rules and other ensemble techniques. Taking into consideration the limitations of this study, there is a need to build more complex and combination of models to get higher accuracy for rainfall prediction system. Study can

also be formulated using greater articulate monitoring for particular area and create this kind of model for enormous dataset so that calculation rate can be increased with better precision and with more accuracy.

## **12.FUTURE SCOPE:**

We can able to update the data Rainfall is the major problem in the agriculture and the economy of the country. Rainfall is more in some areas and less in some areas. Due to this there is economic loss. In future we can improve the accuracy and make the application more friendly to the users. Add extra features in application to get more data. Train and test with more attributes for the accuracy. We can implement the AI and IOT technologies in it.

## **13.APPENDIX:**

### **Sample code:**

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
<script
```

```
src="https://kit.fontawesome.com/64d58efce2.js"
```

```
crossorigin="anonymous"
```

```
></script>
```

```
<link rel="stylesheet" href="style.css" />
```

```
<link rel="stylesheet" href="path/to/font-awesome/css/font-awesome.min.css">
```

```
<title>Exploratory Analysis of Rain Fall Data in India for Agriculture Login  
Page</title>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<div class="signin-signup">
```

```
<form action="#" class="sign-in-form">
```

```
<h2 class="title">Sign in</h2>
```

```
<div class="input-field">
```

```
<i class="fas fa-user"></i>
```

```
<input type="text" placeholder="Username" />
```

```
</div>
```

```
<div class="input-field">
```

```
<i class="fas fa-lock"></i>
```

```
<input type="password" placeholder="Password" />
```

```
</div>
```

```
<input type="submit" value="Login" class="btn solid" />
```

```
<p class="social-text">Or Sign in with social platforms</p>
```

```
<div class="social-media">
```

```
<a href="#" class="social-icon">
```

```
<i class="fab fa-facebook-f"></i>

</a>

<a href="#" class="social-icon">

  <i class="fab fa-twitter"></i>

</a>

<a href="#" class="social-icon">

  <i class="fab fa-google"></i>

</a>

<a href="#" class="social-icon">

  <i class="fab fa-linkedin-in"></i>

</a>

</div>

</form>

<form action="#" class="sign-up-form">

  <h2 class="title">Sign up</h2>

  <div class="input-field">

    <i class="fas fa-user"></i>

    <input type="text" placeholder="Username" />

  </div>

  <div class="input-field">

    <i class="fas fa-envelope"></i>
```

```
<input type="email" placeholder="Email" />

</div>

<div class="input-field">

  <i class="fas fa-lock"></i>

  <input type="password" placeholder="Password" />

</div>

<input type="submit" class="btn" value="Sign up" />

<p class="social-text">Or Sign up with social platforms</p>

<div class="social-media">

  <a href="#" class="social-icon">

    <i class="fab fa-facebook-f"></i>

  </a>

  <a href="#" class="social-icon">

    <i class="fab fa-twitter"></i>

  </a>

  <a href="#" class="social-icon">

    <i class="fab fa-google"></i>

  </a>

  <a href="#" class="social-icon">

    <i class="fab fa-linkedin-in"></i>

  </a>
```

</div>

</form>

</div>

<div class="panels-container">

<div class="panel left-panel">

<div class="content">

<h3><b style="font-size: 25px;" >Exploratory Analysis of Rain Fall Data  
in India for Agriculture</b></h3><br>

<h4>Welcomes you!!</h4>

<p> </p>

<button class="btn transparent" id="sign-up-btn">

Sign up

</button>

</div>



</div>

<div class="panel right-panel">

<div class="content">

<h1 style="font-size: 50px;">One of us ?</h1>

<p >

Join with us and grow with us...

```
</p>

<button class="btn transparent" id="sign-in-btn">

  Sign in

</button>

</div>



</div>

</div>

</div>

</body>

<!-- loader part -->

<!--div class="loader-container">

</div-->

  <script src="app.js"></script>

</body>

</html>
```

GIT HUB LINK:

<https://github.com/IBM-EPBL/IBM-Project-33341-1660218815.git>

DEMO LINK:

<https://drive.google.com/file/d/1gShc4w8fERxKviMCqPmFZDGJVzO8QyEw/view?usp=sharing>