EXPLORATORY ANALYSIS OF RAINFALL DATA IN INDIA FOR AGRICULTURE

NAALAIYA TIRAN PROJECT BASED LEARNING ON PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

A PROJECT REPORT

DEVI PRIYA S	**611819104009**
SOUMYA PRASAD	**611819104045**
YUVA DHARSHINI K	**611819104058**
MEGA G	**611819106026**

TEAM ID : PNT2022TMID40880

FACULTY MENTORS NAME: C PRAKASH NARAYANAN

INDUSRTY MENTORS NAME: LALITHA GAYATHRI

EVALUATOR NAME : B NEELU

P.S.V. COLLEGE OF ENGINEERING AND TECHNOLOGY

(An ISO 9001:2015 Certified Institution)
(Accredited by NAAC with 'A' Grade)

KRISHNAGIRI-635108

NOVEMBER 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "EXPLORATORY ANALYSIS OF RAINFALL DATA IN INDIA FOR AGRICULTURE" is the bonafide work of "DEVI PRIYA S (611819104009), SOUMYA PRASAD (611819104045), YUVADHARSHINI (611819104058) and MEGHA (611819106026)" who carried out the project work under my supervision.

SIGNATURE

Prof. B. SAKTHIVEL.,M.E.,(Ph.D)

Head of the Department

Dept of CSE,
P.S.V.College of
Engineering & Technology,
Krishnagiri-635 108.

SIGNATURE

Prof. C PRAKASH NARAYANAN

Faculty Mentors,

Dept of CSE,
P.S.V.College of
Engineering & Technology,
Krishnagiri-635 108.

Submitted for the **Naalaiya Tiran Project based learning on professional readiness for Innovation, Employability and Entrepreneurship** held on …….……at P.S.V College of Engineering and Technology, Krishnagiri.

INTERNAL EXAMINAR

EXTERNAL EXAMINAR

TABLE OF CONTENTS

CHAPTER 1

EXPLORATORY ANALYSIS OF RAINFALL DATA IN INDIA FOR AGRICULTURE

1.INTRODUCTION

1.1Project Overview :

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 Regression,Linear Regression,Visualisations etc.

The main aim of objective is to find the

- Rainfall Prediction is the application of science and technology to predict the amount of rainfall over a region.
- It is important to exactly determine the rainfall for effective use of water resources, crop productivity and pre-planning of water structures.

1.2 PURPOSE

Rainfall Prediction Model has a main purpose in prediction of the amount of rain in a specific well or division in advance by using various regression technique and find out which one is best for rainfall prediction.This model also helps the farmer for agriculture to decide the crop, helping the watershed department for water storage and also helps to analyze the ground water level.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing problem :

• 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

•The heavy and irregular rainfall can have many impacts like destruction of crops and farms, damage of property. The user need a basic knowledge to use the existing application.

2.2 References :

1. ADALINE system to weather forecasting, Technical Report Stanford Electron, 1964
2. Lee S. Cho S.& Wong P.M. “Rainfall prediction using artificial neural network. Geog. Inf. Decision Anal. 2, 233–242 1998.
3. Michaelides, S. C., Neocleous, C. C. & Schizas, C. N. “Artificial neural networks and multiple linear regression in estimating missing rainfall data.” In Proceedings of the DSP95 International Conference on Digital Signal Processing, Limassol, Cyprus. pp 668–6731995

2.3 Problem Statement Definition :

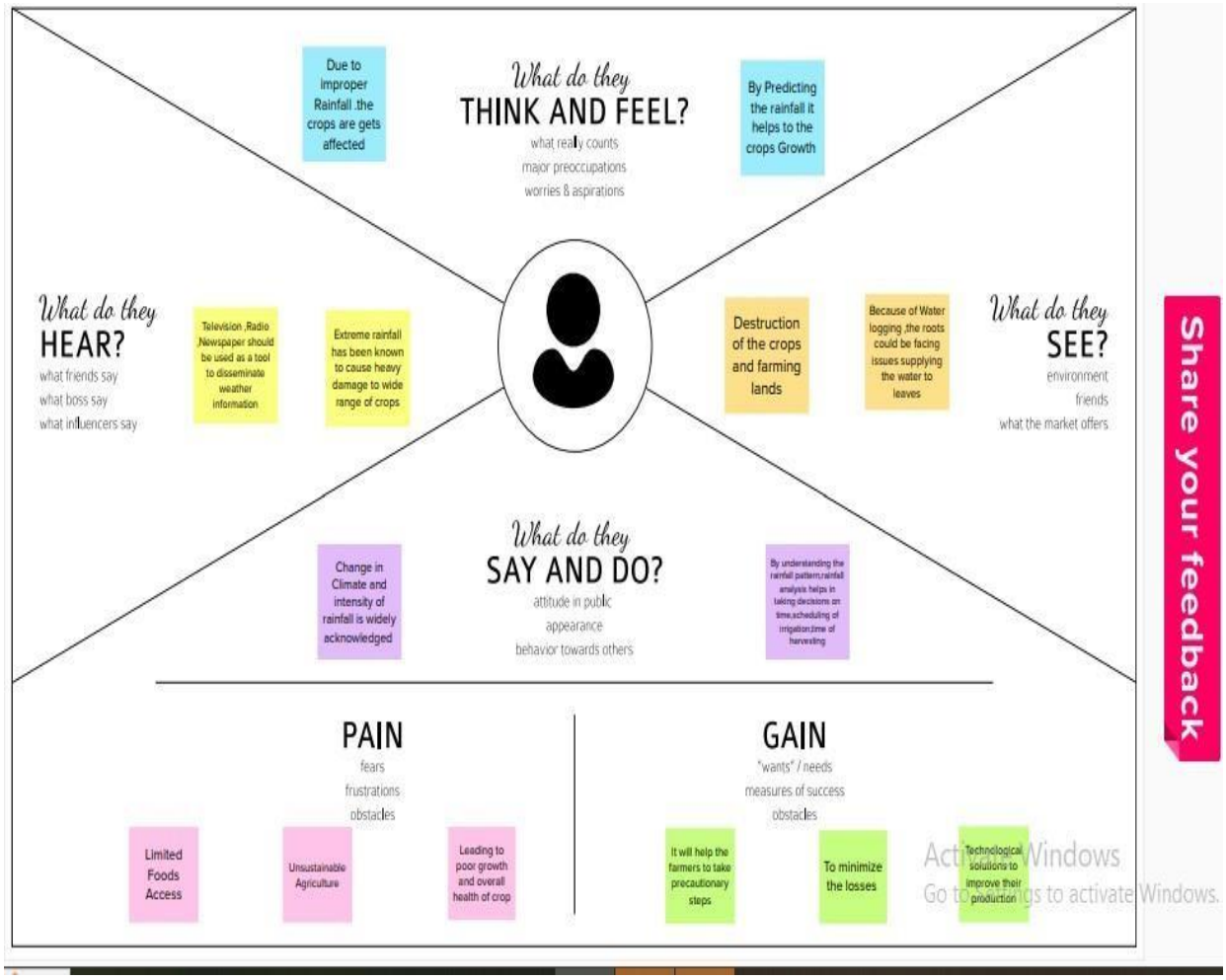
Weather conditions changes then and often. This can lead to Severe threats to all the living beings including human beings. So predicting weather, especially Irregular heavy rainfall, Droughts can cause huge economic losses. This also decreases crop productivity and may lead into Food shortage. Predicting the Rainfall plays a vital role in our life time. Farmers will get benefit due to this and Our country’s GDP will rise. Collection of previous 10 years data may giveus an idea about the pattern of Rainfall.Using all these Datas, Appropriatefarming activities can be performed. Water is the vital mineral for a life. So,these datas can help us in predicting Rainfall during summer days to save water. Agriculture definitely requires gallons of waters.

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1Empathy Map Canvas:

An empathy map is a collaborative visualization used to articulate what we know about a particular type of user. It externalizes knowledge about users in order to create a shared understanding of user needs, and aid in decision makin

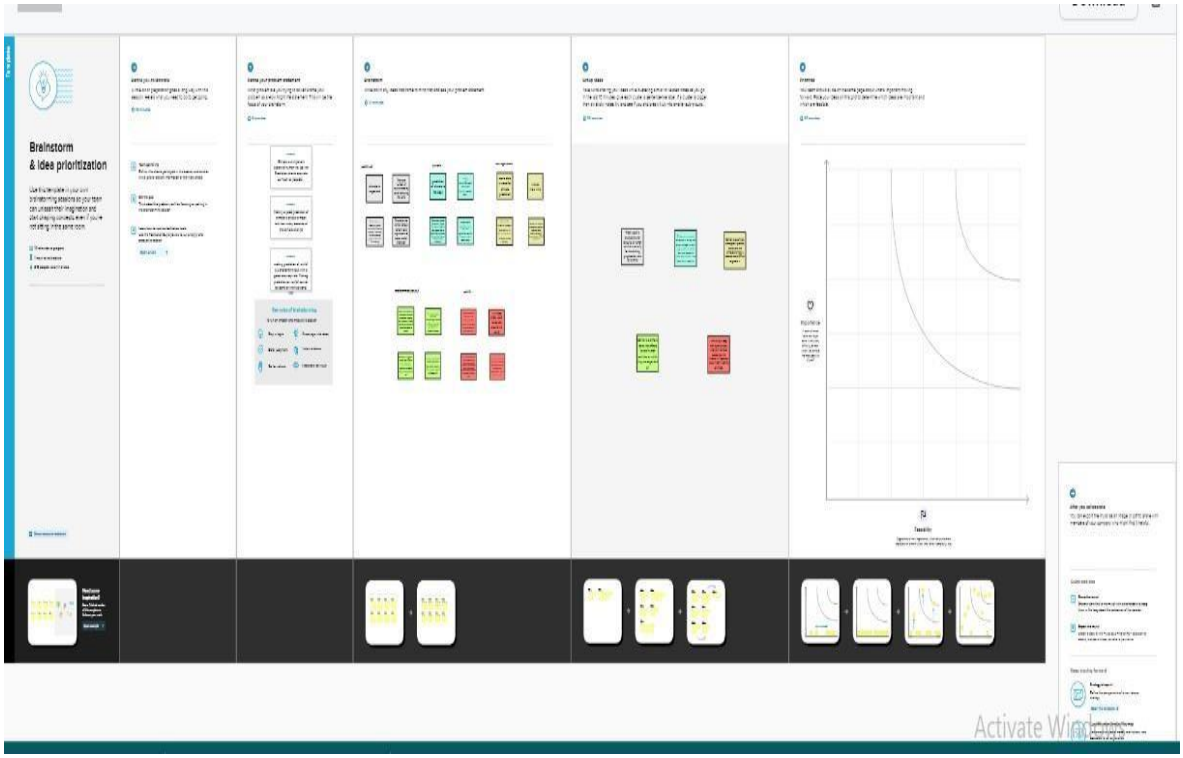


3.2 IDEATION & BRAINSTORMING

Ideation refers to the process of developing and conveying prescriptive ideas to others, typically in a business setting. It describes the sequence of thoughts, from the original concept to implementation.

Ideations can spring forth from past or present knowledge, external influences, opinions, convictions, or principles. Ideation can be expressed in graphical, written, or verbal terms.

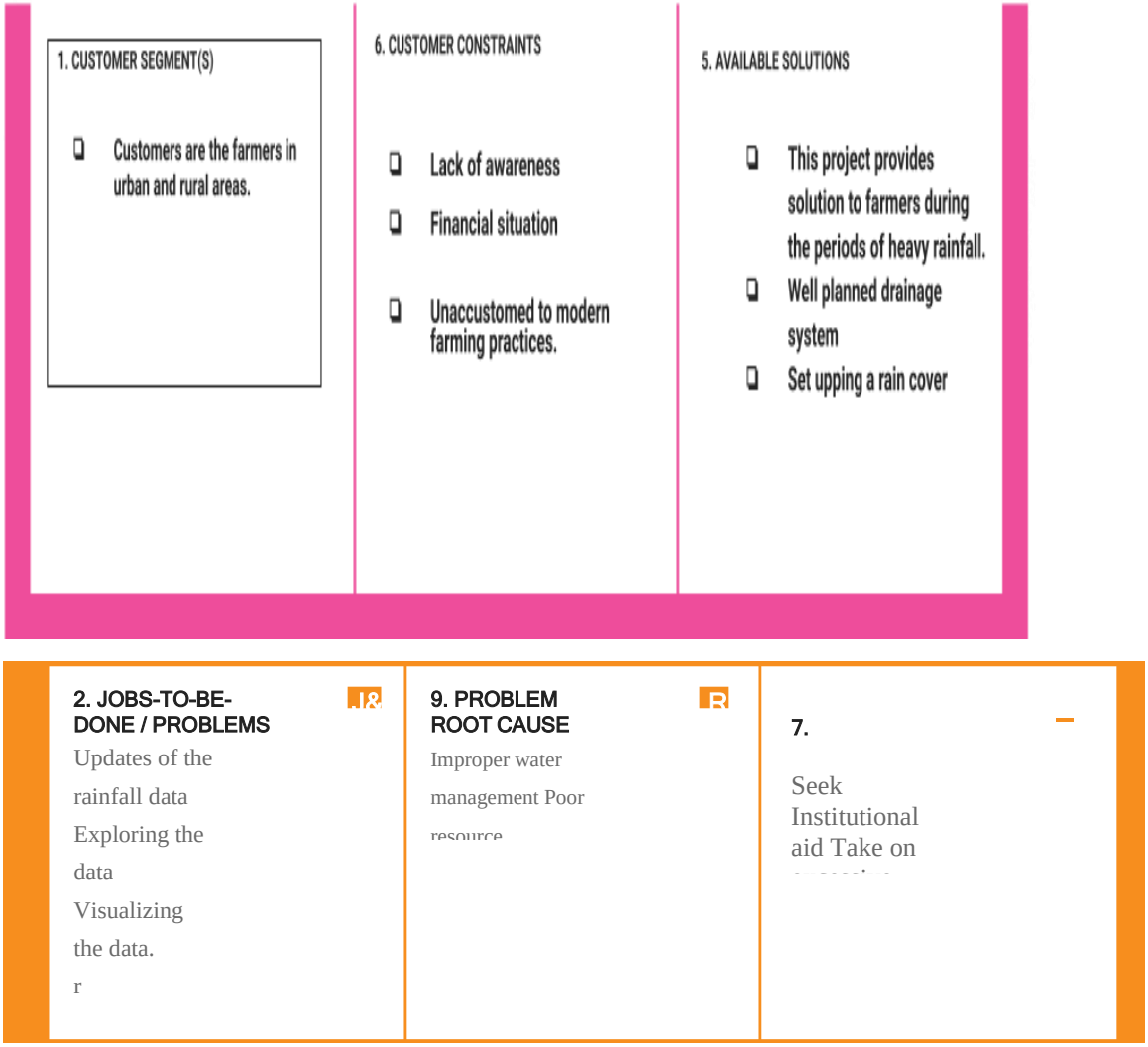
Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1	Problem Statement	Climate is a important aspect of human life. So, the Prediction should accurate as much as possible. In this paper we try to deal with the prediction of the rainfall which is also a major aspect of human life and which provide the major resource of human life which is Fresh Water. • Now climate change is the biggest issue all over the world. Peoples are working on to detect the patterns in climate change as it affects the economy in production to infrastructure.
2	Proposed Solution	Analyzing the previous 10 years data can give us a rough idea about Rainfall pattern. Using Data Science, we can predict the Rainfall up to some good extent.
3	Uniqueness	• This application is useful for the beginners in agriculture. • Seed maturity selection features are available.
4	Social Impact	• Different types of crops can be planted for good health. • Helps in producing healthy crops and good fields.
5	Business Model	This comparative study is conducted concentrating on the following aspects: modeling inputs, Visualizing the data, modeling methods, and pre-processing techniques. The results provide a comparison of various evaluation metrics of these machine learning techniques and their reliability to predict rainfall by analyzing the weather data. We will be using classification algorithms such as Decision tree, Random forest, KNN, and xgboost
6	Scalability	• When we predict rainfall correctly,it helps growth of crop and yielding will be better.

3.4 PROBLEM SOLUTION FIT:



CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Import necessary packages	Import necessary packages Importing packages like NumPy, pandas,seaborn, etc
FR-2	Download and load dataset	Download the dataset Load the Appropriate dataset
FR-3	Pre-processing of data	Making data suitable for building a good model
FR-4	Building Machine learning model	Choose the best algorithm. Check for the best optimised result.
FR-5	Train the data	Train the model using training data.
FR-6	Test the mode	Test the model for the best evaluation and analysing..

4.2 NON FUNCTIONAL REQUIREMENTS

FRNo.	Non-Functional Requirement	Description
NFR-1	Usability	The usability of the website is to make all users willbe satisfied with our requirements of the product. The user should reach the summarized text or resultwith one button press if possible
NFR-2	Security	The security of the project is to develop the website that prevents SQL injection attack, XSS attack and DOS attack
NFR-3	Reliability	The reliability of the system is to make sure the websitedoes not go offline. The users can be reach and use program at any time, so maintenance should not be big issue.
NFR-4	Performance	The performance of the website isto provide data to allusers without

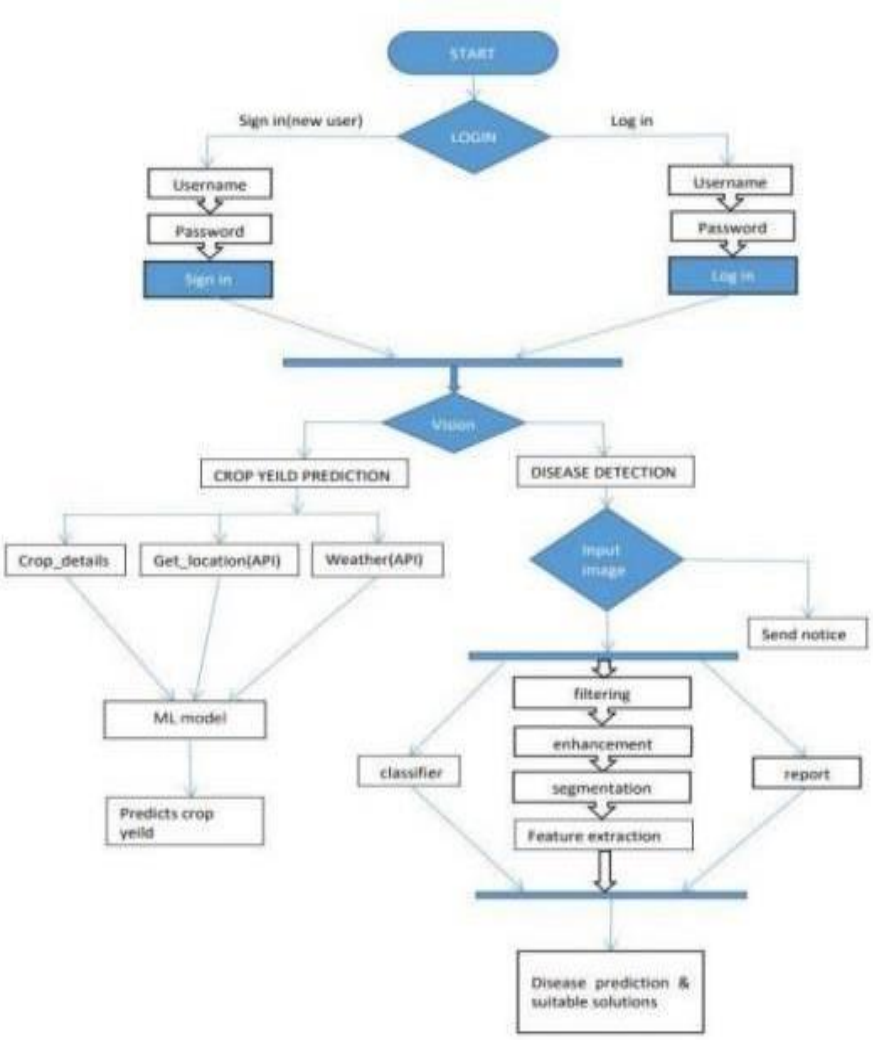
		unnecessary delay and provide 24*7 availability.
NFR-5	Availability	The availability of the website is that the website willbe active on The Internet and people will be able to browse to it.
NFR-6	Scalability	The scalability of the system is we have limited ourproject to Indian cities We have plans to scale it to continent's level in comingupdates.

CHAPTER 5

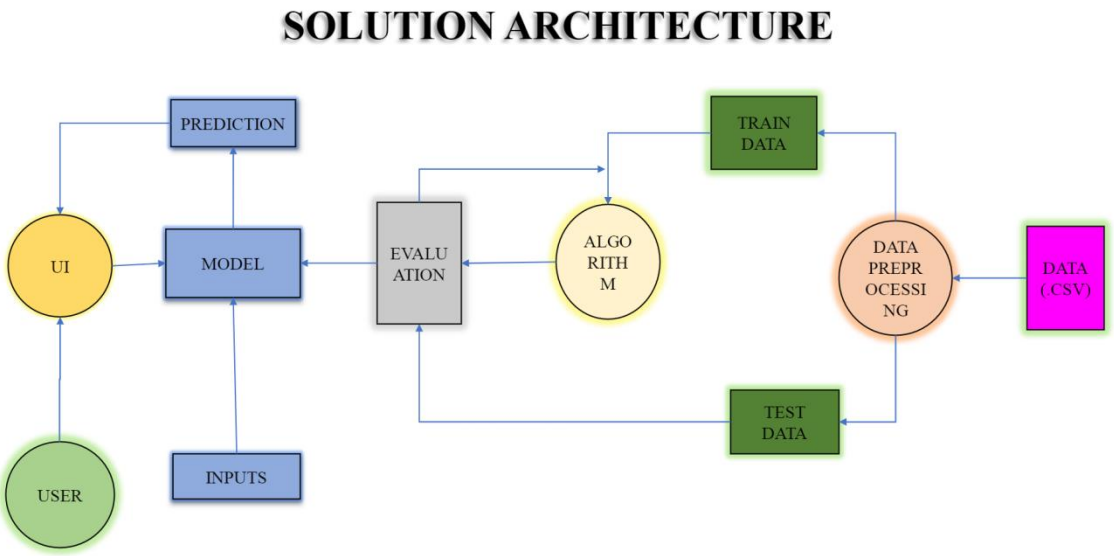
PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

A DataFlowDiagram(DFD)is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system,what changes the information,and where data is stored.



5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	As a user , I can view the details about the page and navigate through the entire pages	I can navigate through the pages.	Medium	Sprint-1
	Prediction	USN-7	User can search for the area / place where the user wants to know the prediction of rainfall .	Searching for the region within INDIA only be accepted	High	Sprint-1
		USN-8	The prediction or analysis for the desired region for the future or past events respectively		High	Sprint-1
		USN-9	User can see the visualization of the rainfall data for the specific region in INDIA for a specified time period.		High	Sprint-1
	News	USN-10	User can view the latest news articles related to agriculture.	I can view the news articles.	Medium	Sprint-2
Customer (Web user)	Support	USN-11	User can ask queries about the system.	I can rectify my doubts	High	Sprint-3
Customer Care Executive		USN-12	The team must analyse all the queries and debug it in the next update		High	Sprint-3

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Core development team	Core function	USN - 14	Design and develop the application in such a way that the best user interface and maintenance should be taken care of		High	Sprint-1
		USN - 1 5.	The website is responsive on all the devices and the screen sizes. User experience should be good irrespective of the devices or platforms		High	Sprint - 1

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Soumya Prasad, Yuva dharshini.
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Devi priya, mega.
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Soumya Prasad, Devi priya, Yuva dharshini
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Yuva dharshini, Mega, soumya Prasad
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Soumya prasad, Devi priya.

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint - 1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint - 2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint - 3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint - 4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

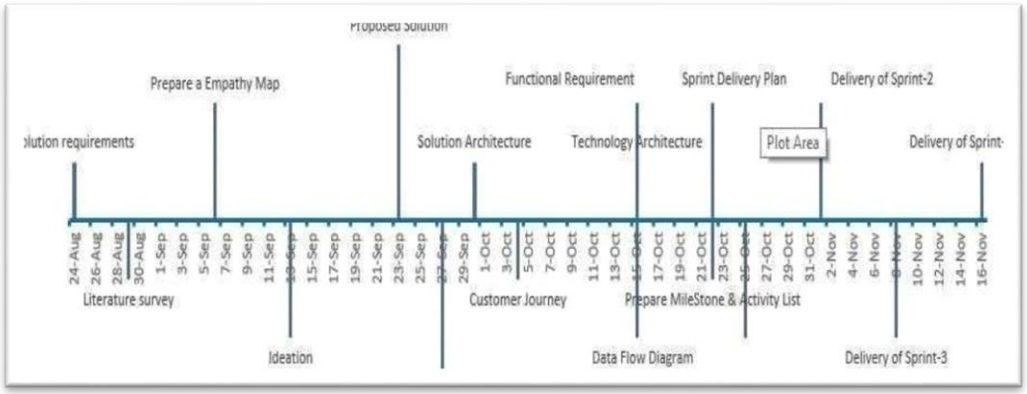
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

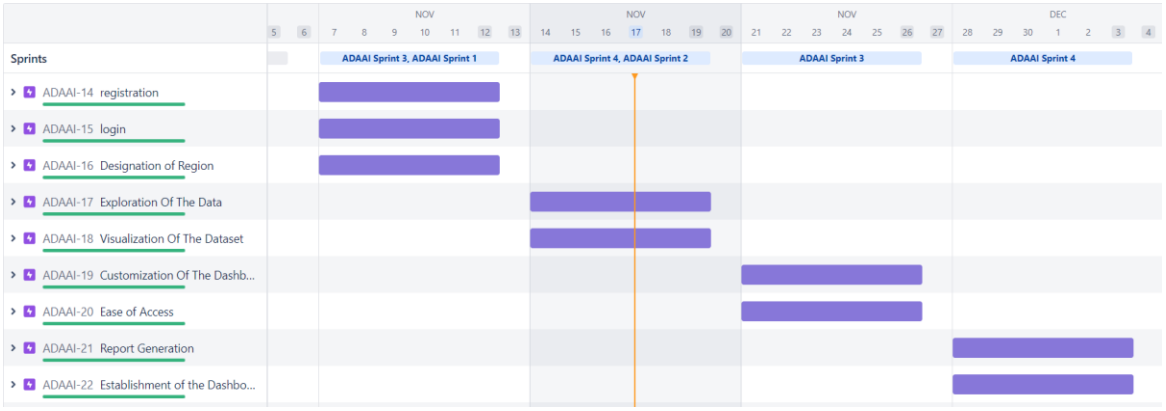
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.2 Sprint Delivery Schedule :

A milestone schedule, or milestone chart, is a timeline that uses milestones to divide a project schedule into major phases. A milestone chart is a way to visualize the most important steps of our project. Each milestone the team achieves brings us closer to completing the project. As a result, milestones provide a sense of accomplishment and show the team how the work they're doing contributes to the overarching project objective.


Proposed Solution
Prepare a Empathy Map
Functional Requirement
Sprint Delivery Plan
Delivery of Sprint-2
klution requirements
Solution Architecture
Technology Architecture
Plot Area
Delivery of Sprint
24-Aug
26-Aug
28-Aug
30-Aug
1-Sep
3-Sep
5-Sep
7-Sep
9-Sep
11-Sep
13-Sep
15-Sep
17-Sep
19-Sep
21-Sep
23-Sep
25-Sep
27-Sep
29-Sep
1-Oct
3-Oct
5-Oct
7-Oct
9-Oct
11-Oct
13-Oct
15-Oct
17-Oct
19-Oct
21-Oct
23-Oct
25-Oct
27-Oct
29-Oct
31-Oct
2-Nov
4-Nov
6-Nov
8-Nov
10-Nov
12-Nov
14-Nov
16-Nov
Literature survey
Customer Journey
Prepare MileStone & Activity List
Ideation
Data Flow Diagram
Delivery of Sprint-3

6.3Reports from JIRA :



CHAPTER 7

WORKING WITH THE DATASETS AND DATA VISUALISATION

7.1 Feature Code

Working With The Dataset :

- Understand the Dataset
- Load the Dataset
- Perform Joins of the Dataset tables

Understanding The Dataset :

The data can be downloaded from the Links :

- weatherAUS
- district-wise-rainfall-normal
- rainfall-in-india-1901-2015

DATASET LINK:

https://www.kaggle.com/datasets/rajanand/rainfall-in-india

Intializing :

For this feature we have made use of Jupyter notebook which uses Python programming language. To use Jupyter Notebook install Anaconda, whichis a desktop graphical user interface (GUI) included in Anaconda® Distribution that allows you to launch applications and manage conda packages, environments, and channels without using command line interface (CLI) commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux. It provides all basic necessary python libraries which are needed for Data Analysis and Visualizations.

Importing libraries

IMPORT NECESSARY LIBRARIES

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re
import os
import collections
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
!pip3 install openpyxl
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: openpyxl in /usr/local/lib/python3.7/dist-packages (3.0.10)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.7/dist-packages (from openpyxl) (1.1.0)
```

Loading of Dataset

2. Exploratory Data Analysis

```
In [2]: df = pd.read_csv("weatherAUS.csv")
        pd.set_option("display.max_columns", None)
        df
```

Out[2]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	WindSp
0	01-12-2008	Albury	13.4	22.9	0.6	NaN	NaN	W	44.0	W	WNW	20.0	
1	02-12-2008	Albury	7.4	25.1	0.0	NaN	NaN	WNW	44.0	NNW	WSW	4.0	
2	03-12-2008	Albury	12.9	25.7	0.0	NaN	NaN	WSW	46.0	W	WSW	19.0	
3	04-12-2008	Albury	9.2	28.0	0.0	NaN	NaN	NE	24.0	SE	E	11.0	
4	05-12-2008	Albury	17.5	32.3	1.0	NaN	NaN	W	41.0	ENE	NW	7.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
145455	21-06-2017	Uluru	2.8	23.4	0.0	NaN	NaN	E	31.0	SE	ENE	13.0	
145456	22-06-2017	Uluru	3.6	25.3	0.0	NaN	NaN	NNW	22.0	SE	N	13.0	
145457	23-06-2017	Uluru	5.4	26.9	0.0	NaN	NaN	N	37.0	SE	WNW	9.0	
145458	24-06-2017	Uluru	7.8	27.0	0.0	NaN	NaN	SE	28.0	SSE	N	13.0	
145459	25-06-2017	Uluru	14.9	NaN	0.0	NaN	NaN	NaN	NaN	ESE	ESE	17.0	

145460 rows × 23 columns

Handling Missing Values

```
In [3]: numerical_feature = [feature for feature in df.columns if df[feature].dtypes != 'O']
        discrete_feature=[feature for feature in numerical_feature if len(df[feature].unique())<25]
        continuous_feature = [feature for feature in numerical_feature if feature not in discrete_feature]
        categorical_feature = [feature for feature in df.columns if feature not in numerical_feature]
        print("Numerical Features Count {}".format(len(numerical_feature)))
        print("Discrete feature Count {}".format(len(discrete_feature)))
        print("Continuous feature Count {}".format(len(continuous_feature)))
        print("Categorical feature Count {}".format(len(categorical_feature)))
```

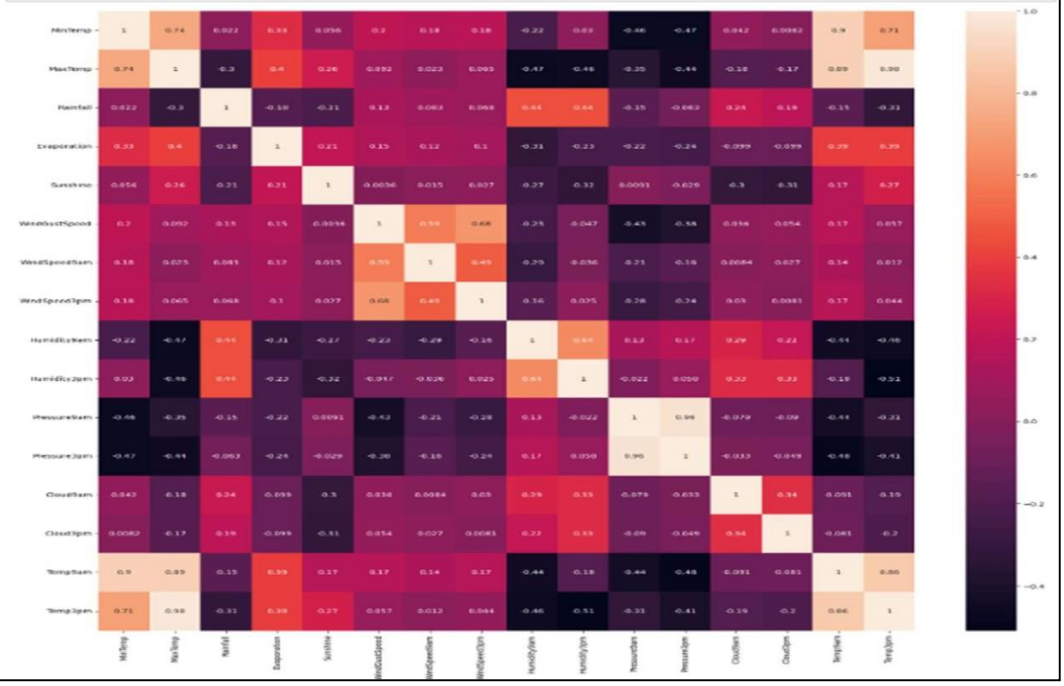
```
Numerical Features Count 16
Discrete feature Count 2
Continuous feature Count 14
Categorical feature Count 7
```

```
In [4]: # Handle Missing Values
        df.isnull().sum()*100/len(df)
```

```
Out[4]: Date             0.000000
        Location         0.000000
        MinTemp          1.020899
        MaxTemp          0.866905
        Rainfall         2.241853
        Evaporation     43.166506
        Sunshine        48.009762
        WindGustDir      7.098859
        WindGustSpeed    7.055548
        WindDir9am       7.263853
        WindDir3pm       2.906641
        WindSpeed9am     1.214767
        WindSpeed3pm     2.105046
        Humidity9am      1.824557
        Humidity3pm      3.098446
        Pressure9am     10.356799
        Pressure3pm     10.331363
        Cloud9am        38.421559
        Cloud3pm        40.807095
        Temp9am          1.214767
        Temp3pm          2.481094
        RainToday        2.241853
        RainTomorrow     2.245978
        dtype: float64
```

Correlation of the data

```
In [9]: corrmat = df.corr(method = "spearman")
        plt.figure(figsize=(20,20))
        #Plot heat map
        g=sns.heatmap(corrmat,annot=True)
```



Null Values

```
In [12]: for feature in continuous_feature:
             if(df[feature].isnull().sum()*100/len(df))>0:
                 df[feature] = df[feature].fillna(df[feature].median())

In [13]: df.isnull().sum()*100/len(df)

Out[13]: Date             0.000000
         Location         0.000000
         MinTemp          0.000000
         MaxTemp          0.000000
         Rainfall         0.000000
         Evaporation      0.000000
         Sunshine         0.000000
         WindGustDir      7.098859
         WindGustSpeed    0.000000
         WindDir9am       7.263853
         WindDir3pm       2.906641
         WindSpeed9am     0.000000
         WindSpeed3pm     0.000000
         Humidity9am      0.000000
         Humidity3pm      0.000000
         Pressure9am      0.000000
         Pressure3pm      0.000000
         Cloud9am         0.000000
         Cloud3pm         0.000000
         Temp9am          0.000000
         Temp3pm          0.000000
         RainToday        2.241853
         RainTomorrow     2.245978
         dtype: float64
```

Splitting of the dataset

3. Splitting Dataset into Independent and Dependent Variables

In [64]:
```
X = df.drop(["RainTomorrow", "Date", "Date_month", "Date_day"], axis=1)
Y = df["RainTomorrow"]
```

4. Feature Scaling

In [65]:
```
scaler = RobustScaler()
X_scaled = scaler.fit_transform(X)
```

5. Splitting The Data Into Train And Test

In [66]:
```
X_train, X_test, y_train, y_test = train_test_split(X_scaled,Y, test_size =0.2, stratify = Y, random_state = 0)
```

In [67]:
```
X_train.shape
X_test.shape
```

Out[67]: (29092, 21)

In [68]:
```
y_train.shape
y_test.shape
```

Out[68]: (29092,)

Model Evaluation

10. Model Evaluation

9]:
```
import sklearn.metrics as metrics
```

Accuracy_score

0]:
```
print(metrics.accuracy_score(y_train,p1))
```

0.9999472546020359

1]:
```
print(metrics.accuracy_score(y_test,p2))
```

0.8567460177924681

Balancing the data

6. Balancing the Data

In [69]:
```
sm=SMOTE(random_state=0)
X_train_res, y_train_res = sm.fit_resample(X_train, y_train)
print("The number of classes before fit {}".format(Counter(y_train)))
print("The number of classes after fit {}".format(Counter(y_train_res)))
```

```
The number of classes before fit Counter({0: 90866, 1: 25502})
The number of classes after fit Counter({0: 90866, 1: 90866})
```

7.2 Feature code

User Interface

Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<title>Weather App using Flask in Python</title>
 <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
      <style> body
      {
  background-image: url('https://www.worldatlas.com/r/w768/upload/7e/2e/5a/untitled-
design-79.jpg');
   background-repeat:    no-repeat;
 background-attachment:     fixed;
 background-size: cover;

      }
</style>
</head>
<body>
      <div class="container">
         <br><br><br>
              <div class="row"><h2 style="color:Blue;">Weather Prediction App</h2></div>
              <br>
              <div class="row">
                    <b style="color:Tomato;">Get weather details of any city around the
world.</b>
              </div>
```

```
<div class="row">
    {% block content %}
        <form action="{{ url_for("index")}}" method="post">
        <div class="form-group">
            <label style="color:Red;" for="Email">Email:</label><br>
        <input type="email" id="Email" name="Email" value="{{Email}}"
placeholder="Email" required><br>
            <label style="color:blue;"
for="cityName"><b>Password:</b></label><br>
        <input type="password" id="password" name="password"
value="{{password}}" placeholder="password" required><br>
        <label for="cityName"><b style="color:Yellow;">City
Name:</b></label><br>
        <input type="text" id="cityName" name="cityName"
value="{{cityName}}" placeholder="City Name" required><br>
            <br>
        <button class="submit">Find</button>
        {% if error is defined and error %}
            <br><br><span class="alert alert-danger">Error: Please enter
valid city name.</span></br>
        {% endif %}
        </div>
    {% endblock %}
    {% if data is defined and data %}
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Country Code</th>
                <th>Coordinate</th>
                <th>temperature</th>
                <th>Pressure</th>
                <th>Humidity</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td class="bg-success">{{ data.sys.country }}</td>
```

```
                    <td class="bg-info">{{data.coord.lon }}
{{data.coord.lat}}</td>
                    <td class="bg-danger">{{data.main.temp }} k</td>
                    <td class="bg-warning">{{data.main.pressure}}</td>
                    <td class="bg-primary">{{data.main.humidity}}</td>
                </tr>
                >
```

App.py

from flask import Flask, request, render_template import
requests
from flask import Flask, request, render_template import
requests

```
app = Flask(_name_)


@app.route('/', methods=["GET", "POST"])
def index():
  weatherData = ''
  error = 0
  cityName = ''
  if request.method == "POST":
    cityName = request.form.get("cityName")
    if cityName:
      weatherApiKey = '3f5d38932ad9ae0caa0302a35fbc8496'
      url = "https://api.openweathermap.org/data/2.5/weather?q=" + cityName + "&appid="
+ weatherApiKey
      weatherData = requests.get(url).json()
    else:
      error = 1
  return render_template('index.html', data=weatherData, cityName=cityName, error=error)


if _name_ == "_main_":
  app.run()

app = Flask(_name_)


@app.route('/', methods=["GET", "POST"])
def index():
  weatherData = ''
  error = 0
  cityName = ''
  if request.method == "POST":
    cityName = request.form.get("cityName")
    if cityName:
      weatherApiKey = '3f5d38932ad9ae0caa0302a35fbc8496'
```

```
        url = "https://api.openweathermap.org/data/2.5/weather?q=" +
cityName + "&appid=" + weatherApiKey
        weatherData =
      requests.get(url).jso
      n() else:
        error = 1
  return render_template('index.html', data=weatherData,
  cityName=cityName, error=error)


if _name_
  ==
  "_main_"
  :
  app.run(
  )
```

CHAPTER 8
TESTING

8.1 TEST CASES

Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Steps To Execute**	**Expected Result**	**Actual Result**	**Status**
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Sign up popup when user clicked on My account button	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Sing up popup displayed or not	Login/Sig nup popup should display	Working as expected	Pass

LoginPage_TC_OO2	UI	dashboard page	verify user is able to see airport report in dashboard page	1.Airstat dashboard will be displayed. 2.Check if each tab can able to access. 3.Click on the required dataset. 4.OBtain the report	required visualisation will be displayed on the dashboard	working as expected	Pass

8.2 USER ACCEPTANCE TESTING

Defect Analysis:

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

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Test Case Analysis :

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

This report shows the number of test cases that have passed, failed, and untested

CHAPTER 9

RESULTS

9.1 PERFORMETRICS

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Paramete r	Values	Screenshot

1.	Metrics	Classification
		
		Model:Confusion
		
		Matrix - Accuracy
		
		Score- Classification Report
			`y_pred = cat.predict(X_test)`
			`print(confusion_matrix(y_test,y_pred))`
			`print(accuracy_score(y_test,y_pred))`
			`print(classification_report(y_test,y_pred))`
			
			`[[21510 1207]`
			` [ 2795 3580]]`
			`0.8624364086346762`
			`precision recall f1-score support`
			
			`0 0.89 0.95 0.91 22717`
			`1 0.75 0.56 0.64 6375`
			
			`accuracy 0.86 29092`
			`macro avg 0.82 0.75 0.78 29092`
			`weighted avg 0.85 0.86 0.85 29092`

2.	Tune the Model	Hyperparameter Tuning –
		
		Validation Method -
			`{'learning_rate': 0.1, 'max_depth': 8}`
			`0.8892227301457538`
			
			`Accuracy:83.11 %`
			`Standard Deviation:17.73 %`

CHAPTER 10

ADVANTAGES & DISADVANTAGES

Advantages :

- Farmers can know when to plant or harvest their crops
- People can choose where and when to take their holidays to take advantages of goodweather
- Surfers known when large waves are expected
- Regions can be evacuated if hurricanes or floods are expected
- Aircraft and shipping rely heavily on accurate weather forecasting
- It will help the farmers to take precautionary steps
- Technological solutions to improve their production

Disadvantages :

- Weather is extremely difficult to forecast correctly
- It is expensive to monitor so many variables from so many sources
- The computers needed to perform the millions of calculations necessary are expensive
- The weather forecasters get blamed if the weather is different from the forecast
- Leading to poor growth and overall health of crop
- Limited Foods Access

CHAPTER 11

CONCLUSION

The weather prediction has become one of the most essential entities now a days. To improve the risk management systems and to know the weather in coming days in an automatic and in scientific way, many models have been emerging to assist in weather Prediction. In this paper, we have seen building a Weather Prediction Web Application from scratch by making use of 6 different ML algorithms namely CatBoost Classifier, RandomForset Classifier, Logistic Regression, GaussianNB, KNN and XGB Classifier.In the result section, the results from the all the six models and its results such as Accuracy, Error rate, mean absolute error, Root mean squared error, Relative squared error, Root relative squared error and time taken to build the model are tabulated. The results show that the CatBoost Classifier and XGB Classifier has output the results of high accuracy than all the other classifiers that were used. When coming to the time taken to build the model, The CatBoost Classifier outperforms all the other classifiers in solving the Problem under scrutiny.

CHAPTER 12

FUTURE SCOPE

In upcoming future updates, the WEATHER FORECASTING application will have additional features suchas:

- Live Location tracking
- News on Live Disasters
- Weather Forecast for next one week
- Will deploy as android app
- Help in predicting which crop will be best suited according to weather conditions

13.APPENDIX :

Source Code:

Ipynb file Link: RAINFALL PREDICTION

UI Link: FILE

GITHUB

DEMO VIDEO

Github repositories :

https://github.com/IBM-EPBLIBM-Project-38886-1660386433