EXPLORATORY ANALYSIS OF RAINFALL DATA IN INDIA FOR AGRICULTURE

A PROJECT REPORT

Submitted by

IVARAJ C

KISHORE KUMAR S

MOHAMED SHEHIN S

VISWAJITHSAIRAM S K

in partial fulfilment of the award of the degree of

BACHELOR OF ENGINERERING

in

COMPUTER SCIENCE AND ENGINEERING

RAMCO INSTITUTE OF TECHNOLOGY, RAJAPALAYAM-626 125

DEC 2022

Lalitha Gayathri

Gomathy nayagam

(INDUSTRY MENTOR)

(FACULTY MENTOR)

Project Report Format

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 (Explain the features added in the project along with code)
- 7.1 Feature 1
- 7.2 Feature 2
- 7.3 Database Schema (if Applicable)

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

Source Code GitHub & Project Demo Link

1. INTRODUCTION:

1.1. Project Overview:

With the changes in the climatic conditions and irregular pattern of weather conditions, predicting their occurrence for preventing lifeloss to humankind and environment is an utmost societal needed problem of the society.Drastic changes in climate have occurred over the past yearsand with change in revolution proper preventive measures are needed.Heavy rainfall can lead tofloods.Flash floods are catastrophic.Climate change is increasing the frequency, intensity and magnitude of disasters, leading to a higher number of deaths and injuries, as well as increased property and economic losses. In the past 20 years, 90% of major disasters have been caused by weather-related events such as heatwaves, storms, floods and droughts, according to the UN Officefor Disaster Risk Reduction (UNISDR).Natural disasters are increasing in strength and frequency.Shifting weather patterns make predictions and emergency planning difficult. Hence , we focus onthe effective prediction of the probability of the flood occuring in a particular region and recommending an evacuation area nearby by performing an exploratory analysis of the data collected.

1.2. Purpose

To design a disaster management system by forecasting a flood event to control flood risk by recommending an evacuation area from flood hazard areas which ultimately helps to manage theenvironment and water resource system.This also serves a purpose of the Early warning system bytraining a model and selecting the best prediction algorithm among the classifiers.The occurrence offlash floods can cause catastrophic damage to the society. They first mainly affect the people livingnear to the riverbeds.Evacuating them from the hazard areas and providing them the shelter theyneeded.With the irregular change in climate patterns, it's been difficult to predict the occurrence offloods using traditional methods leading to massive destruction. Thus to cope with flash floods andto handle critical situations new methodologies are invented to overcome such difficulties.Technology has to be more aware to reduce the loss that a flash flood would make. In themodernizing era, it's made even easier to predict the occurrence of floods and recommend nearbyevacuation areas. Hazard areas that are prone to destruction and devastating loss are monitoredregularly and the rainfall readings are collected, integrated from multiple resources, curated, mined,analyzed and prediction is done over patterns.

With the prediction, recommendation areas are listed for the society.Early warning systems are climate change adaptation measures that use integratedcommunication systems to help communities prepare for dangerous climate-related events. An earlywarning system’s success saves lives and jobs, land and infrastructure, and supports long-term sustainability.

2.LITERATURESURVEY:

2.1. Existing Solutions

This paper deals with the idea of predicting floods using the algorithm Artificial neural networks(ANN) and with the support of the Internet of Things. This system looks after the humidity, temperature, pressure, rainfall, and river water level periodically to the temporal correlative information for flood prediction analysis. Flood data is dynamic and non-linear in nature. The sensors read the data and inform the system. With those values, the prediction is doneand the decision is taken on the occurrence of a flood.[1] Precipitation in any form such as snow, rain or hail can affect the routine of the society.Therefore predicting the occurrence of rainfall beforehand and warning the society about the day’s condition can be helpful in a lot more ways.Providing accurate results for forecasting rainfall has been a major issue with the drastic change in climatic conditions.Using a fusion of machine learning techniques can help in providing much more accurate results about the occurrenceof rainfall.Four supervised learning algorithms has been used to get out the accurate results forprediction.The four effective algorithms that results in accurate prediction are decision tree, NaïveBayes, K-nearest neighbors, and support vector machines.The effectiveness of the algorithm ischecked by incorporating the technology known as fuzzy logic.A twelve year historical weatherdata of city lahore is considered for training ,validating and for testing .In such a way that thisfusion model outperformed other existing models.[2] The drastic change in climatic conditions has caused severe impact on the society and environment. A country's economic and financial condition is mainly dependent on the country’s agriculture. Farming and agriculture are considered to be India’s backbone of economic conditions.In such a way any climate change affects the agricultural development which in directly affects the economic and financial conditions of the country.Therefore predicting the occurrence of rainfall is one of the most important aspect for the safety of the society as well as the country with its economic conditions. Loss in agriculture could lead to famine and create a huge economic crisis.Prediction made should be to the point. The traditional methods of predicting rainfall have gone out of control with the drastic change in climatic conditions and development of the country.With the rise in global warming conditions, rough humidity and change in the oceans predicting rainfall with any modest technologies that results in the precise results is an utmost needof the society.Applying machine learning classification algorithms to predict the accurate results ofrainfall has been implemented.UCI repository dataset has been considered for training,validating and testing.[3] With nature being unpredictable the intensity of the rainfall varies according to the climatic conditions and the pressure of the wind. Under such conditions, urban floods can be a great disasterfor society. This paper deals with a classification-based real-time flood prediction model with thesupport of a numerical analysis model based on hydraulic theory and the required machine learning models. The Flood database has been created beforehand with the help of the Environmental Protection Agency-Storm Water Management model and from a twodimensional inundationmodel. Using the Latin hypercube sampling and probabilistic neural network are used forcategorizing the flood depth data into five categories. This machine learning model is constructedto identify the respective cumulative volume if the observed rainfall data is entered. Therefore asystem that's capable of generating a real-time flood map by cumulative volume of each grid to the cumulative volume using linear regression and nonlinear regression. The developed system can predict the rainfall-induced flooding potential in such a way that reduces the risk due to disaster and minimizes damage to health and properties. Therefore a useful disaster management system has been developed for preventing huge losses due to disasters.[4] On a high note, research has been continuously carried out on achieving efficient and accurate prediction technology or systems. With the help of machine learning techniques and algorithms, prediction can be made easy to obtain accurate and earlier results such as making therequired arrangements and evacuating people from the hazard areas. Over the two decades, neuralnetworks have shown an extraordinary outcome in predicting the occurrence of floods with the given rainfall data providing better results and cost-effective solutions. This paper is novel in the way of analyzing databases by Multi-layer perceptron classifier to read data such as dynamicidentification, deficit treatment, data validation, and data cleaning to be carried across the database.Advancements in every note can provide better results based on the preprocessing of data.[5

2.2. References

[1] Swapnil Bande, Virendra V. Shete, “Smart flood disaster prediction system using IoT& neural networks“, 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)

[2] Atta-ur Rahman,Sagheer Abbas, Mohammed Gollapalli, Rashad Ahmed, Shabib Aftab, Munir Ahmad,Muhammad Adnan Khan, Amir Mosav, “Rainfall Prediction System Using Machine Learning Fusion for Smart Cities”, 2022 May National Library of Medicine

[3] Vikas Kumar, Vishal Kumar Yadav, Er. Sandeep Dubey, “Rainfall Prediction using Machine Learning”, Ijraset Journal For Research in Applied Science and Engineering Technology, 2022.

[4] Ho Jun Keum, Kun Yeun Han & Hyun Il Kim, “Real-Time Flood Disaster PredictionSystem by Applying Machine Learning Technique”, KSCE Journal of Civil Engineering 24, 2835-2848(2020)

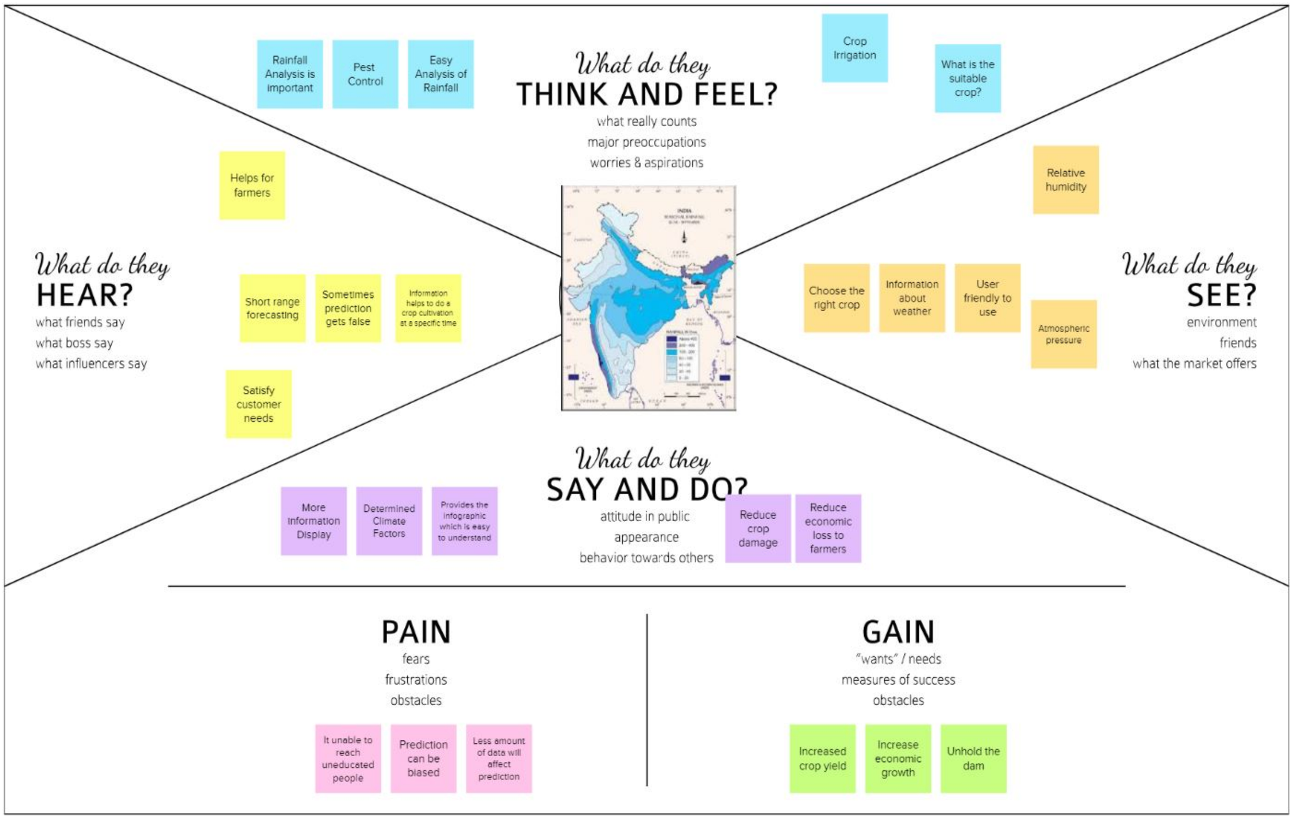
[5] Thegeshwar Sivamoorthy, Asif Mohammed Ansari, Dr. B. Sivakumar, V. Nallarasan,“Flood Prediction Using ML Classification Methods on Rainfall Data”, IJRASET Journal For Research in Applied Science and Engineering Technology

2.3. Problem Statement Definition

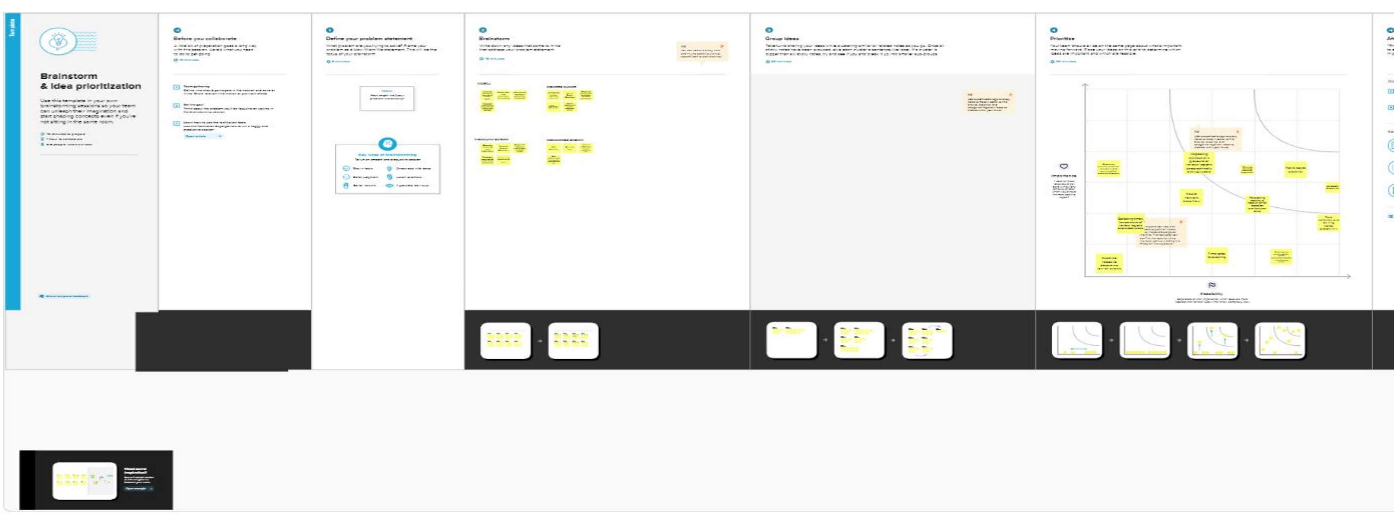
S.No.	Parameter	Description
1		In this paperwe try to deal with the prediction of the rainfall whichis a major aspect of human life Climate is a important resource of human life. So, the Prediction should accurate as muchas possible. Heavy Rainfall may cause huge threat toall living beings, especially in the field of agriculture. Predicting Rainfall is a major task in both summer and Rainy season

3. IDEATION AND PROPOSED SOLUTION:

3.1. Empathy Map



3.2. Ideation and Brainstorming:



3.3. Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem tobesolved)	In this paperwe try to deal with the prediction of the rainfall whichis a major aspect of human life Climate is a important resource of human life. So, the Prediction should accurate as muchas possible. Heavy Rainfall may cause huge threat toall living beings, especially in the field of agriculture. Predicting Rainfall is a major task in both summer and Rainy season
2.	Idea / Solution description	Using Data Science, we could solve this and predict the Rainfallprediction rainfall is a challenging task with a good accuracy rate. Making prediction on rainfall cannot be doneby the traditional way, soscientist is using machine learning and deep learning to find out the pattern for rainfall prediction,so it can generate extra support to maintain the agriculture.
3.	Novelty / Uniqueness	We are not going to use any kind of equipment.Time of prediction is very less and easy withaffordable cost.
4.	Social Impact / CustomerSatisfaction	varies types of healthy crops can beplanted .and also Helps in producing healthy crops to the customers
5.	Business Model (Revenue Model)	This comparative study is conducted concentrating on the following aspects: modelling inputs, Visualizing the data,

		modelling methods, and pre-processing techniques. The results providea comparison
		of variousevaluation metrics of these machinelearning techniques and their reliability to predict rainfall by analyzing theweather data. We will be using classification algorithms such as Decision tree, Randomforest, KNN, and xgboost This could cost really low as a person should develop knowledge in Data scienceand probably a gadget to developthis. However, deploying as an App attached with other facilities may cost anextra charge
6.	Scalability of solution	if we canpredict the rainfall accurately andthen we can help in improvement of crops growth

3.4 Problem Solution fit

Define CS, fit into CC	1. CUSTOMER SEGMENTS CS	6. CUSTOMER CONSTRAINTS CC	5. AVAILABLE SOLUTIONS AS	Explore AS, differentiate
	• Mainly Farmers • Employees/Workers associated with Agricultural activities • Departments of the government or news organisations seeking agricultural rainfall forecasts	• To estimate the duration and volume of rainfall beforehand and take decisions accordingly • To get a prediction with 100% accuracy • Cost factors for applications with high prediction accuracy and value • Limited time to make use of digital devices to get the prediction information • Unstable network connection	• News on weather forecasting from various communication media like radio, news channels, etc. • Announcements from the concerned authorities and notifications from connections [friends and families] on upcoming rainfalls affecting the agriculture	

Focus on J&P, tap into BE, understand RC	2. JOBS-TO-BE-DONE / PROBLEMS J&P	9. PROBLEM ROOT CAUSE RC	7. BEHAVIOUR BE	Focus on J&P, tap into BE, understand RC
	• Get proper analysis from previous data • Achieve correct and accurate predictions • Sudden change in weather and immediate rainfall or showers • Damage to crops due to heavy rainfall	• Irregular rainfall in various regions of India • Drastic variability in climate change • Biodiversity loss	• Take suggestions from concerned authorities, agricultural scientists, and other influencers to make decisions • Take decisions as per previous experiences and self-analysis	

Identify Strong TR & EM	3. TRIGGERS TR	10. OUR SOLUTION SL	8. CHANNELS of BEHAVIOUR CH	Identify Strong TR & EM
	• Current losses and debts • Yearly crop damage due to heavy rainfall • Evolving market competition and change in demand-supply **4. EMOTIONS: BEFORE / AFTER EM** • Before : Paying debts, incurring losses, low crop production • After : Increase in crop production, making effective decisions, experiencing growth and profits	• Region [district or sub-division] based analysis of previous years' rainfall data to get the seasonal patterns with respect to the production of different sorts of crops • Building a low-cost or free ML-based application [consuming low bandwidth] to predict the rainfall of places in India with a high concentration of agricultural activities while taking care of the trends and analysis done already	8.1 ONLINE • Receive early notifications on their digital devices, especially mobiles or smartphones, through SMS or app alerts 8.2 OFFLINE • Community forums, meeting where farmers and other people can share ideas, discuss and decide on crop activities	

4. REQUIREMENT ANALYSIS:

4.1. Functional Requirements

S.No	Component	Description	Technology
1	User Interface	The user interacts with the application through a web UI and a chatbot	HTML,CSS,Python, FLASK
2	Application logic-1	Logic for registration	Python
3	Application logic-2	Logic for login to the application	python

4	Application logic-3	Integration machinelearning model and the webpage	flask
5	Database	Numeric data	Mysql
6	File storage	To store file such as prediction report	Local file system

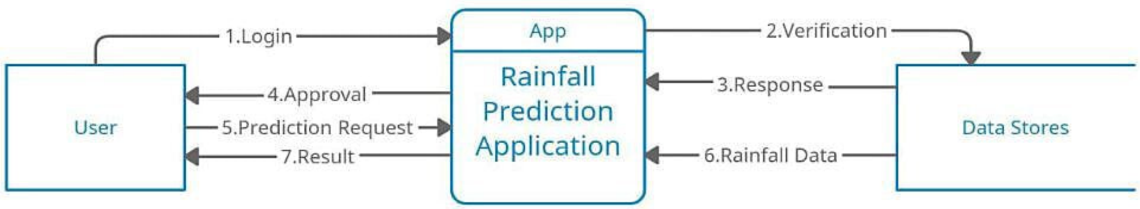
Non Functional Requirements

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask	Micro web framework written in Python
2.	Security Implementations	Basic HTTP authentication, Session based authentication, User Registration, Login Tracking	Flask Security
3.	Scalable Architecture	Size is everything, and Flask's status as a microframework means that you can use it to grow a tech project such as a web app incredibly quickly. Its simplicity of use and few dependencies enable it to run smoothly even as it scales up and up.	Flask
4.	Availability	Higher compatibility with latest technologies and allows customization	Flask

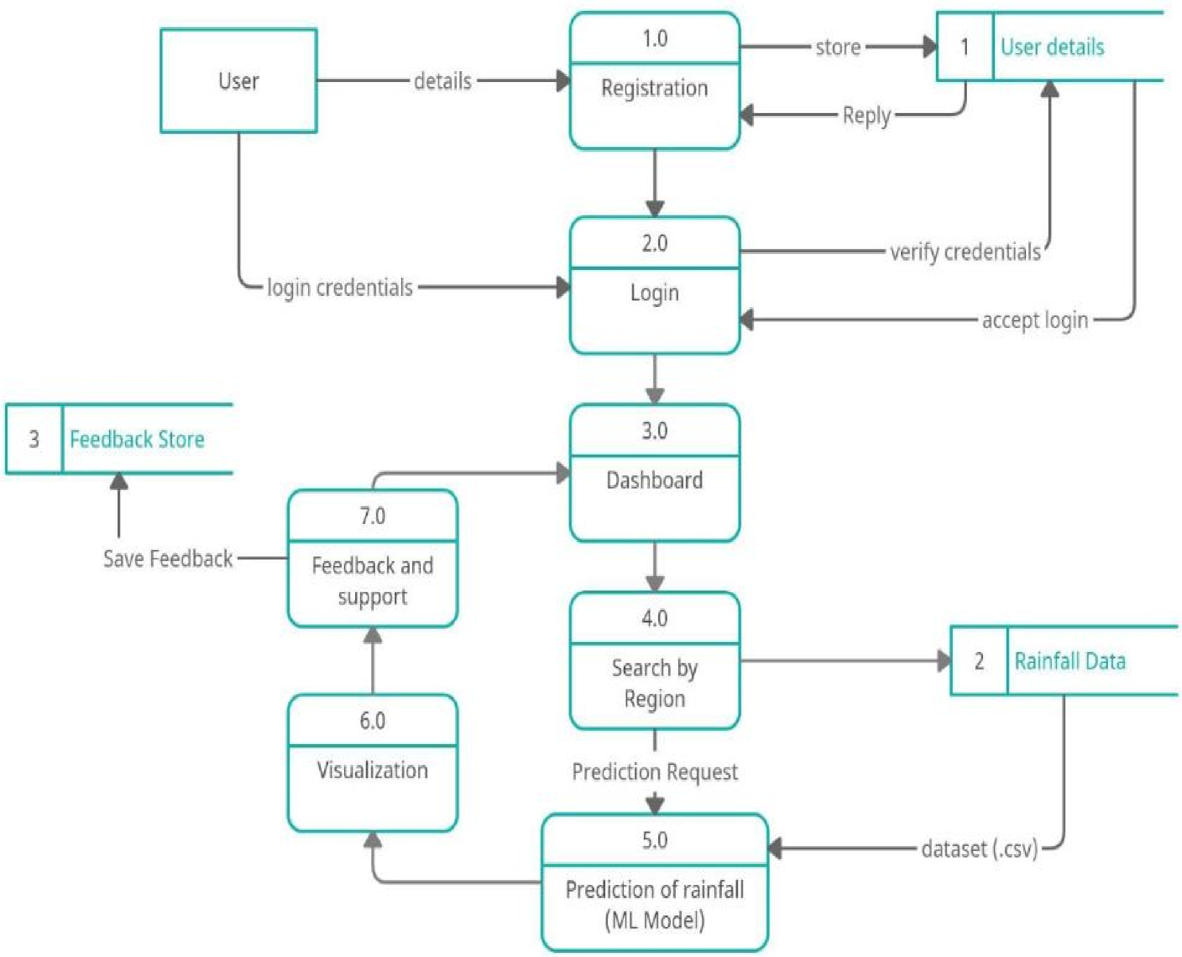
5. PROJECT DESIGN

Data Flow Diagram:

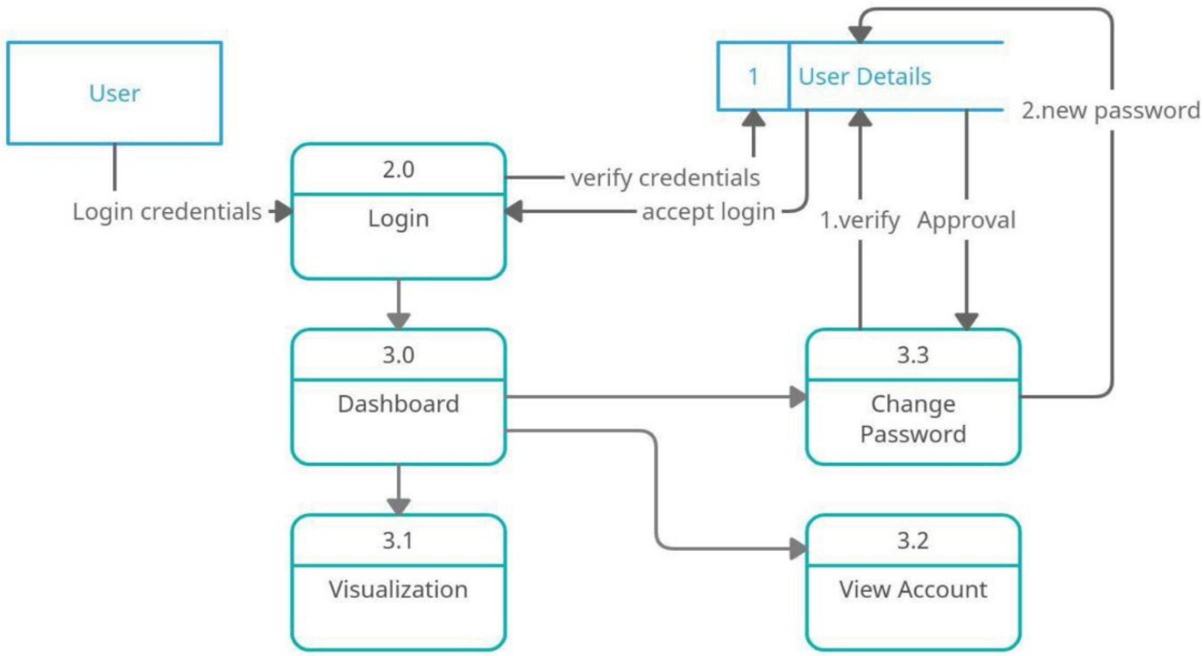
0 LEVEL DIAGRAM



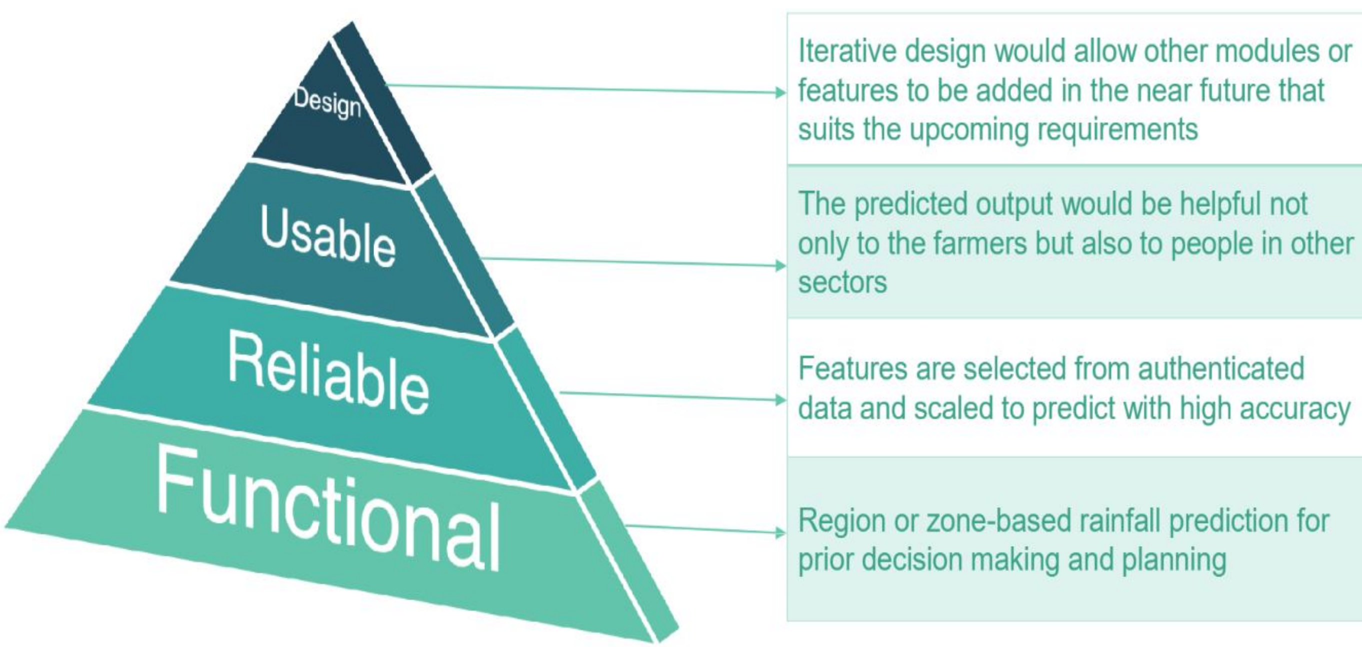
1 LEVEL DIAGRAM

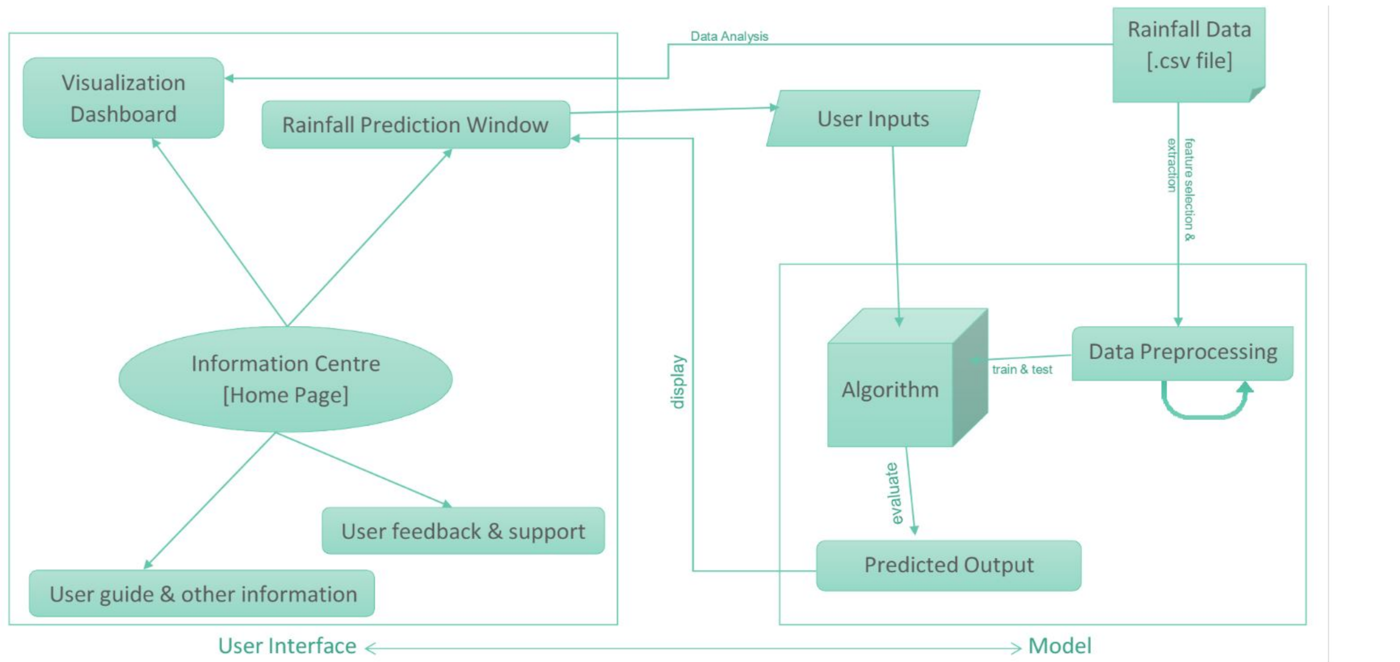


2 LEVEL DIAGRAM



Solution And Technical Architecture:





5.3. User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User in Website	Registration	USN-1	User can register for the application by entering his or her email, password, and confirming the password.	Account specific tasks and actions can be performed	High	Sprint-1
		USN-2	User will receive confirmation email or message once registered for the application	Verify the registered account	High	Sprint-1
		USN-3	Validation of the user can be done directly using email or OTP	Account validated and got access to profile dashboard	Medium	Sprint-1
	Login	USN-4	Enter the username and password to login to the application	Right account credentials should be entered	High	Sprint-1
		USN-5	The existing credentials should be used for login on multiple systems		Medium	Sprint-1
	Dashboard	USN-6	User can search for the region where he/she wants to know the prediction of rainfall	Searching for the region in India will be accepted only	High	Sprint-2
		USN-7	User can view the visualization of the rainfall data for a specific region in India or for a specific time period		Medium	Sprint-2
		USN-8	User can change his/her password and can view the account details and search history	Verification will be required and new password should be entered	High	Sprint-2
		USN-9	The prediction or analysis request can be asked for the desired region for future or past events respectively		High	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		USN-10	User can give the feedback on the accuracy of the prediction and on the user interface		High	Sprint-3
Support Team	Support	USN-11	Responds to user queries via telephone, email etc.	Queries can be raised in situation of doubts	Medium	Sprint-3
		USN-12	The team must analyse all the queries and try to debug and make plans so that such queries wouldn't be raised again		Low	Sprint-3
		USN-13	Organize for a FAQ session where commonly asked doubts can be redressed by the team	The user will get all their doubt clarified	Low	Sprint-3
		USN-14	The team must respond immediately to the queries based on the priority	Queries should get resolved	High	Sprint-3
Core Development Team	Core Function	USN-13	Design, develop the application in such a way that the best user interface and maintenance should be taken care of.	Easy and self-understandable user interface	High	Sprint-4
		USN-14	The website is responsive on all the devices and the screen sizes	User experience should be good irrespective of the devices or platforms	Medium	Sprint-4
		USN-15	The updates should be on time with the solutions of the raised queries	The existing functionalities should not affected by the update	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING:

Planning & Estimation:

Project Tracker, Velocity & Burndown Chart: (4 Marks)

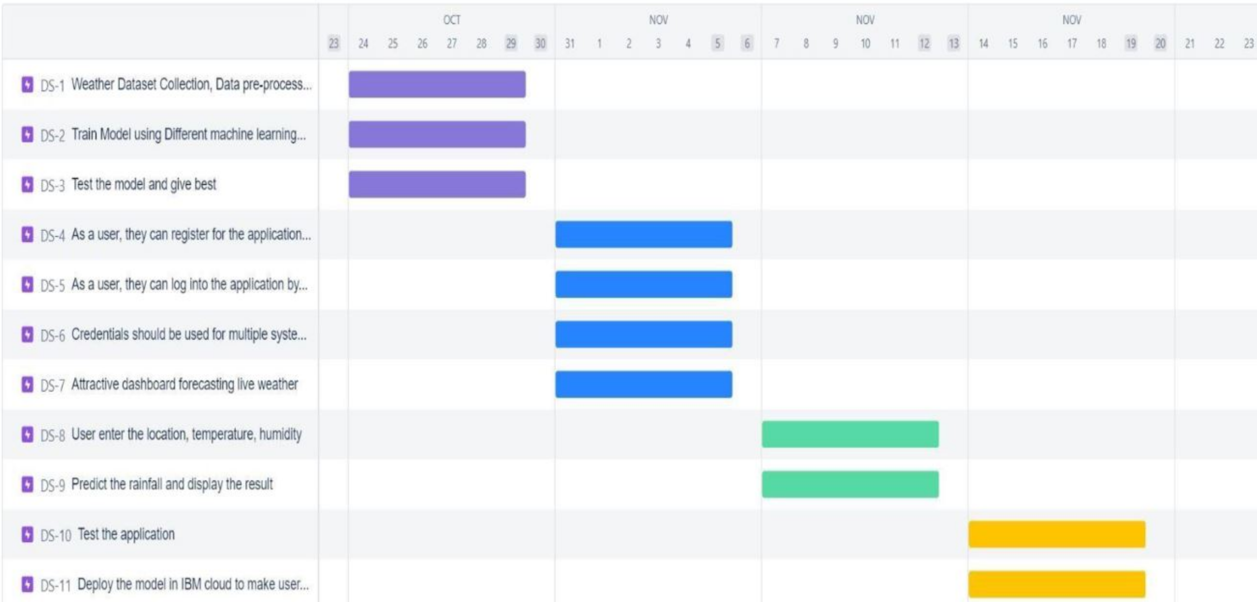
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	31Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-2	20	6 Days	05 Nov 2022	10 Nov 2022	20	10 Nov 2022
Sprint-3	20	6 Days	10 Nov 2022	15 Nov 2022	20	15 Nov 2022
Sprint-4	20	6 Days	15 Nov 2022	21 Nov 2022	20	21 Nov 2022

6.2. Sprint Delivery Schedule:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Rainfall PredictionML Model (Dataset)	USN-1	Weather Dataset Collection, Datapreprocessing, Data Visualization.	5	High	C.Ivaraj , S.Kishore Kumar
Sprint-1		USN-2	Train Model using Different machine learningAlgorithms	5	High	S.K.Viswajithsairam , S.Mohamed Shehin
Sprint-1		USN-3	Test the model and give best	10	High	C.Ivaraj , S.Mohamed Shehin
Sprint-2	Registration	USN-4	As a user, they can register for the applicationthrough Gmail. Password is set up.	5	Medium	S.Kishore Kumar , S.K.Viswajithsairam
Sprint-2	Login	USN-5	As a user, they can log into the application byentering email & password	5	Medium	S.Mohamed Shehin , S.Kishore Kumar
Sprint-2		USN-6	Credentials should be used for multiplesystems and verified	4	Medium	S.K.Viswajithsairam, C.Ivaraj
Sprint-2	Dashboard	USN-7	Attractive dashboard forecasting live weather	6	Low	S.Mohamed Shehin , S.Kishore Kumar
Sprint-3	Rainfall Prediction	USN-8	User enter the location, temperature, humidity	10	High	C.Ivaraj , S.K.Viswajithsairam
Sprint-3		USN-9	Predict the rainfall and display the result	10	High	S.Kishore Kumar, C.Ivaraj

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Testing	USN-10	Test the application	10	High	S.K.Viswajithsairam , S.Kishore Kumar
Sprint-4	Deploy Model	USN-11	Deploy the model in IBM cloud to make userfriendly application	10	High	S.K.Viswajithsairam , S.Mohamed Shehin

6.3. Report



7. CODE AND SOLUTIONING

Feature 1:

```
<Html>
<Head>
<Title>
EDA of Rainfall LOGIN!!
</Title>

<style type=text/css>
body
{
height: 125vh;
margin-top: 20px;
padding: 30px;
font-family: sans-serif;
}
</style>
</Head>
<Body>
<h1 style="color:rgb(216, 64, 64);">
```

```
<center> EXPLORATORY ANALYSIS OF RAIN FALL DATA IN INDIA FOR AGRICULTURE</h1> </cent
<h2 style="color:white;">
<center> <marquee> A Single Gentle Rain Makes The grass Many Shades Greener </marquee></h
<Title>
LOGIN PAGE
</Title>
<center><style type=text/css>
Body {
  font-family: Calibri, Helvetica, sans-serif;
font-size: 190,90;
background-image: url("Capture.jpg.JPG");
background-position: center;
background-repeat: no-repeat;
background-attachment: fixed;
background-size: cover;
}
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-color: white;
}
button {
        background-color: red;
       width: 100%;
        color: rgb(255, 255, 255);
        padding: 15px;
        margin: 10px 18px;
        border: blue;
        cursor: pointer;
         }
 form {
        border: 3px solid #ffffff8a;
        background-color: #ffffff8a;
padding: 10px 18px;
       width:50%;
    margin-left:25%;
    margin-right:25%;
```

```
   color: blue;
   }
 input[type=text], input[type=password] {
       width: auto;
       margin: 8px 0;
       padding: 10px 18px;
       display: inline-block;
       border: 2px blue;
       box-sizing: border-box;
   }
 button:hover {

padding: 10px 18px;
      width:50%;
   margin-left:25%;
   margin-right:25%;
   }
.subbtn
{
      padding: 10px 18px;
      width:50%;
   margin-left:25%;
   margin-right:25%;
}
  .cancelbtn {
      padding: 10px 18px;
      width:50%;
   margin-left:25%;
   margin-right:25%;
}
.regbtn {
       padding: 10px 18px;
      width:50%;
   margin-left:25%;
   margin-right:25%;
}
   }
```

```
.container {
        padding: 25px;
        background-image: url("rain7.jpg");
background-position: center;
background-repeat: no-repeat;
background-attachment: fixed;
background-size: cover;
    }
</style>
</head>    <center><body background="rain7.jpeg"></center>
<center><style type=text/css>
Body {
  font-family: Calibri, Helvetica, sans-serif;
font-size: 1000,1000;
}
}
<style>
</style>
</head>
<body>
    <center> <h1>  LOGIN FORM </h1> </center>
    <form style="margin: auto; width: 220px;">
        <div class="container">
           <h3> <label>Username : </label>
            <input type="text"  name="username" required><br>
            <label>Password : </label>   <h3>
            <input type="password"  name="password" required>  <br>

<button type="button" class="subbtn"id="login">Login</button>
<a href="ibmregister.html">
<a href="./ibmregister.html"><button type="button" class="regbtn"id="register">Register</button><
<button type="button" class="cancelbtn"> Cancel</button>
<br>
            <h5 style="color:blue;">
            <a href="#"> Need Help in Login? </a>
        </div>
    </form>
```

```
</body>
</html>
</p>
</Body>
</Html>
```

Feature 2:

```
import numpy as np
import pickle
#import joblib
#import matplotlib
#import matplotlib.pyplot as plt
#import time
import pandas
#import os
from flask import Flask, request, jsonify, render_template, redirect, url_for



# 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":
        input_feature=[x for x in request.form.values() ]
        features_values=[np.array(input_feature)]
        names = [['Location','MinTemp','MaxTemp','Rainfall','WindGustSpeed',
        'WindSpeed9am','WindSpeed3pm','Humidity9am','Humadity3pm',
        'Pressure9pm','Pressure3am','Temp9pm','Temp3pm','RainyTodaty',
        'WindGustDir','WindDir9pm','WindDir3pm']]
        data = pandas.DataFrame(features_values,columns=names)
        data = scale.fit_transform(data)
        data = pandas.DataFrame(data,columns=names)

        #Get prediction
        prediction = model.predict(data)

    else:
        prediction = ""
```

```
    if prediction == 1:
      return redirect(url_for('chance'))

    elif prediction == 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)
Footer
```

Feature 3 :

```
<html>
    <head>
        <meta charset="utf-8">
        <title>Rainfall Prediction Webpage</title>

        <link rel="stylesheet" href="{{ url_for('static', filename='css/home.css') }}">

            <link rel="preconnect" href="https://fonts.googleapis.com">
            <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
            <link href="https://fonts.googleapis.com/css2?family=Raleway&family=Roboto:wght@100;
rel="stylesheet">
            <style type="text/css"></style>
    </head>

<body>
<header>
    <div class="header1">
        <img src="{{url_for('static', filename='css/logo.png')}}" alt="logo" />
```

```
</div>
  <div class="navbar">
        <ul>

              <div class="nav"><a href="">HOME</a></div>
              <div class="nav"><a href="{{ url_for('predict') }}">PREDICTOR</a></div>
    <div class="nav"><a href="{{ url_for('about') }}">ABOUT</a></div>
              <div class="nav"><a href="{{ url_for('help') }}">HELP</a></div>
              <div class="nav"><a href="{{ url_for('contact') }}">CONTACT</a></div>

        </ul>
</div>
    </header>
    <div>
        <div class="head1">
        Forecast Rainfall
    </div>
    <div class="body1">
        We serve as an early warning system to exactly determine the rainfall for effective us
  crop productivity, and pre-planning of water structures.
    </div>
    </div>
</body>
<footer><p>IBM - Nalaiya Thiran</p></footer>
</html>
```

8. TESTING:

Test Cases

Test case ID	Feature Type	Component	Test Scenario	Prerequisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
HomePage_TC_001	UI	Home Page	Verify all the UI elements in Home page rendered properly	HTML	1. Enter URL and click go 2. Verify all the UI elements displayed or not		All the UI elements rendered properly	Working as expected	Pass
HomePage_TC_002	Functional	Home page	Verify the Data Entry page can be reachable.	HTML, CSS	1. click the predict tab in navigation bar. 2. Verify all the UI elements displayed or not.		User should navigate to Predictor page	Working as expected	Pass

[76]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn import model_selection
from sklearn import metrics
from sklearn import linear_model
from sklearn import ensemble
from sklearn import tree
from sklearn import svm
import xgboost
import warnings
warnings.filterwarnings('ignore')
import collections
```

[49]:
```python
df = pd.read_csv("Dataset.csv")
```

[7]:
```python
df.head()
```

[7]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	...	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm
0	2008-12-01	Delhi	13.4	22.9	0.6	NaN	NaN	W	44.0	W	...	71.0	22.0	1007.7	1007.1
1	2008-12-02	Delhi	7.4	25.1	0.0	NaN	NaN	WNW	44.0	NNW	...	44.0	25.0	1010.6	1007.8
2	2008-12-03	Delhi	12.9	25.7	0.0	NaN	NaN	WSW	46.0	W	...	38.0	30.0	1007.6	1008.7
3	2008-12-04	Delhi	9.2	28.0	0.0	NaN	NaN	NE	24.0	SE	...	45.0	16.0	1017.6	1012.8
4	2008-12-05	Delhi	17.5	32.3	1.0	NaN	NaN	W	41.0	ENE	...	82.0	33.0	1010.8	1006.0

5 rows × 23 columns

[8]:

```
df.columns
```

t[8]:

```
Index(['Date', 'Location', 'MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation',
       'Sunshine', 'WindGustDir', 'WindGustSpeed', 'WindDir9am', 'WindDir3pm',
       'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',
       'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am',
       'Temp3pm', 'RainToday', 'RainTomorrow'],
      dtype='object')
```

[9]:

```
df.shape
```

t[9]:

```
(145460, 23)
```

[10]:

```
df.info()
```

```
RangeIndex: 145460 entries, 0 to 145459
Data columns (total 23 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Date           145460 non-null  object
 1   Location       145460 non-null  object
 2   MinTemp        143975 non-null  float64
 3   MaxTemp        144199 non-null  float64
 4   Rainfall       142199 non-null  float64
 5   Evaporation    82670 non-null   float64
 6   Sunshine       75625 non-null   float64
 7   WindGustDir    135134 non-null  object
 8   WindGustSpeed  135197 non-null  float64
 9   WindDir9am     134894 non-null  object
 10  WindDir3pm     141232 non-null  object
 11  WindSpeed9am   143693 non-null  float64
 12  WindSpeed3pm   142398 non-null  float64
```

```
 13  Humidity9am    142806 non-null  float64
 14  Humidity3pm    140953 non-null  float64
 15  Pressure9am    130395 non-null  float64
 16  Pressure3pm    130432 non-null  float64
 17  Cloud9am       89572 non-null   float64
 18  Cloud3pm       86102 non-null   float64
 19  Temp9am        143693 non-null  float64
 20  Temp3pm        141851 non-null  float64
 21  RainToday      142199 non-null  object
 22  RainTomorrow   142193 non-null  object
dtypes: float64(16), object(7)
memory usage: 25.5+ MB
```

:

```
df.describe()
```

:

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustSpeed	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	I
count	143975.000000	144199.000000	142199.000000	82670.000000	75625.000000	135197.000000	143693.000000	142398.000000	142806.000000	140953.000000	130395.00000	1:
mean	12.194034	23.221348	2.360918	5.468232	7.611178	40.035230	14.043426	18.662657	68.880831	51.539116	1017.64994	
std	6.398495	7.119049	8.478060	4.193704	3.785483	13.607062	8.915375	8.809800	19.029164	20.795902	7.10653	
min	-8.500000	-4.800000	0.000000	0.000000	0.000000	6.000000	0.000000	0.000000	0.000000	0.000000	980.50000	
25%	7.600000	17.900000	0.000000	2.600000	4.800000	31.000000	7.000000	13.000000	57.000000	37.000000	1012.90000	
50%	12.000000	22.600000	0.000000	4.800000	8.400000	39.000000	13.000000	19.000000	70.000000	52.000000	1017.60000	
75%	16.900000	28.200000	0.800000	7.400000	10.600000	48.000000	19.000000	24.000000	83.000000	66.000000	1022.40000	
max	33.900000	48.100000	371.000000	145.000000	14.500000	135.000000	130.000000	87.000000	100.000000	100.000000	1041.00000	

:

```
df.isnull().any()
```

:

```
Date            False
Location        False
MinTemp          True
```

```
MinTemp           True
MaxTemp           True
Rainfall          True
Evaporation       True
Sunshine          True
WindGustDir       True
WindGustSpeed     True
WindDir9am        True
WindDir3pm        True
WindSpeed9am      True
WindSpeed3pm      True
Humidity9am       True
Humidity3pm       True
Pressure9am       True
Pressure3pm       True
Cloud9am          True
Cloud3pm          True
Temp9am           True
Temp3pm           True
RainToday         True
RainTomorrow      True
dtype: bool
```

```python
df[ "MinTemp"].fillna(df["MinTemp"].mean(),inplace=True)
df[ "MaxTemp"].fillna(df["MaxTemp"].mean(),inplace=True)
df[ "Rainfall"].fillna(df["Rainfall"].mean(),inplace=True)
df[ "Evaporation"].fillna(df["Evaporation"].mean(),inplace=True)
df[ "Sunshine"].fillna(df["Sunshine"].mean(),inplace=True)
df[ "Pressure9am"].fillna(df["Pressure9am"].mean(),inplace=True)
df[ "Cloud9am"].fillna(df["Cloud9am"].mean(),inplace=True)
df[ "Cloud3pm"].fillna(df["Cloud3pm"].mean(),inplace=True)
df[ "Temp9am"].fillna(df["Temp9am"].mean(),inplace=True)
df[ "Temp3pm"].fillna(df["Temp3pm"].mean(),inplace=True)
df[ "WindGustSpeed"].fillna(df["WindGustSpeed"].mean(),inplace=True)
df[ "WindSpeed9am"].fillna(df["WindSpeed9am"].mean(),inplace=True)
df[ "WindSpeed3pm"].fillna(df["WindSpeed3pm"].mean(),inplace=True)
df[ "Humidity9am"].fillna(df["Humidity9am"].mean(),inplace=True)
df[ "Humidity3pm"].fillna(df["Humidity3pm"].mean(),inplace=True)
```

```python
print("Unique values in WindGustDir:",df.WindGustDir.unique())
print("Unique values in WindDir9am:",df.WindDir9am.unique())
print("Unique values in WindDir3pm:",df.WindDir3pm.unique())
print("Unique values in RainToday:",df.RainToday.unique())
print("Unique values in RainTomorrow:",df.RainTomorrow.unique())
```

```
Unique values in WindGustDir: ['W' 'WNW' 'WSW' 'NE' 'NNW' 'N' 'NNE' 'SW' nan 'ENE' 'SSE' 'S' 'NW' 'SE'
 'ESE' 'E' 'SSW']
Unique values in WindDir9am: ['W' 'NNW' 'SE' 'ENE' 'SW' 'SSE' 'S' 'NE' nan 'SSW' 'N' 'WSW' 'ESE' 'E'
 'NW' 'WNW' 'NNE']
Unique values in WindDir3pm: ['WNW' 'WSW' 'E' 'NW' 'W' 'SSE' 'ESE' 'ENE' 'NNW' 'SSW' 'SW' 'SE' 'N' 'S'
 'NNE' nan 'NE']
Unique values in RainToday: ['No' 'Yes' nan]
Unique values in RainTomorrow: ['No' 'Yes' nan]
```

```python
df[ "WindGustDir"].fillna(df["WindGustDir"].mode()[0],inplace=True)
df[ "WindDir9am"].fillna(df["WindDir9am"].mode()[0],inplace=True)
df[ "WindDir3pm"].fillna(df["WindDir3pm"].mode()[0],inplace=True)
df[ "Pressure3pm"].fillna(df["Pressure3pm"].mode()[0],inplace=True)
df[ "RainToday"].fillna(df["RainToday"].mode()[0],inplace=True)
df[ "RainTomorrow"].fillna(df["RainTomorrow"].mode()[0],inplace=True)
```

```python
df.isnull().any()
```

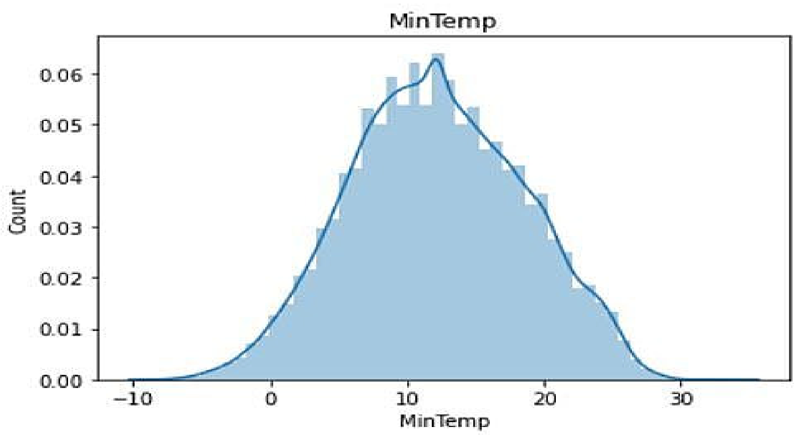
```
Date             False
Location         False
MinTemp          False
MaxTemp          False
Rainfall         False
Evaporation      False
Sunshine         False
WindGustDir      False
WindGustSpeed    False
WindDir9am       False
WindDir3pm       False
WindSpeed9am     False
```

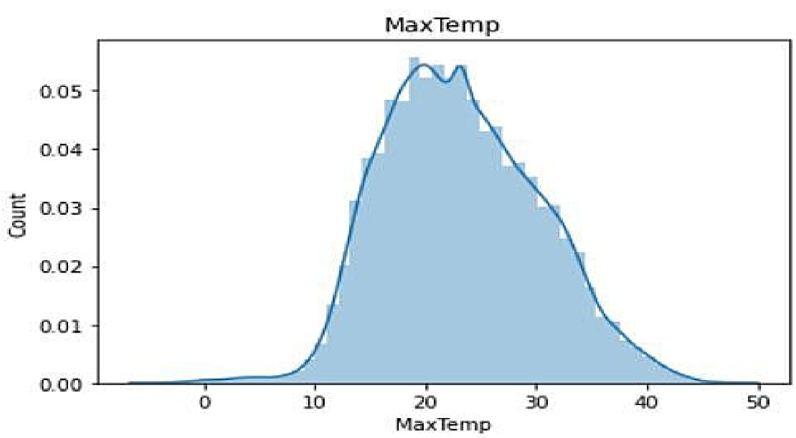
```
WindDir3pm       False
WindSpeed9am     False
WindSpeed3pm     False
Humidity9am      False
Humidity3pm      False
Pressure9am      False
Pressure3pm      False
Cloud9am         False
Cloud3pm         False
Temp9am          False
Temp3pm          False
RainToday        False
RainTomorrow     False
dtype: bool
```

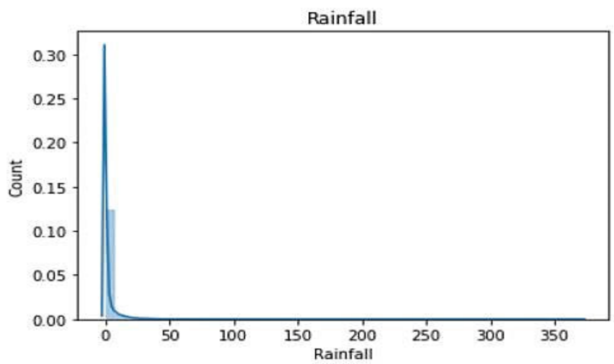
```
numerical_feature = [feature for feature in df.columns if df[feature].dtypes != 'O']
discrete_feature=[feature for feature in numerical_feature if len(df[feature].unique())<25]
continuous_feature = [feature for feature in numerical_feature if feature not in discrete_feature]
categorical_feature = [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)))
```

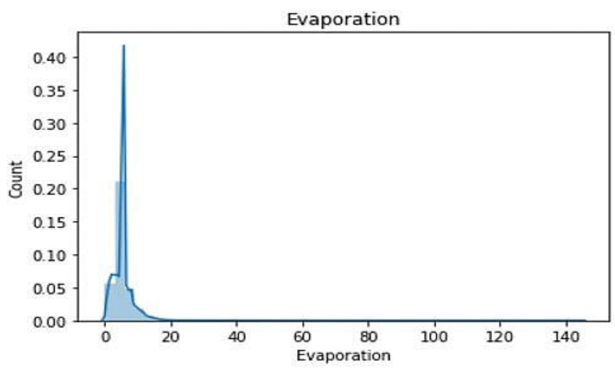
```
Numerical Features Count 16
Discrete feature Count 2
Continuous feature Count 14
Categorical feature Count 7
```

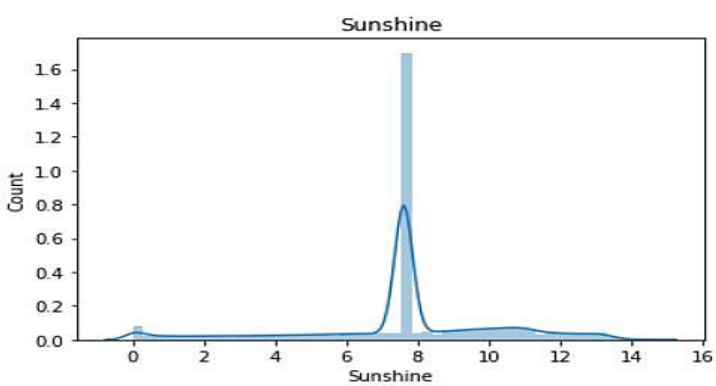
```
for feature in continuous_feature:
    data=df.copy()
    sns.distplot(df[feature])
    plt.xlabel(feature)
    plt.ylabel("Count")
    plt.title(feature)
    plt.figure(figsize=(15,15))
    plt.show()
```

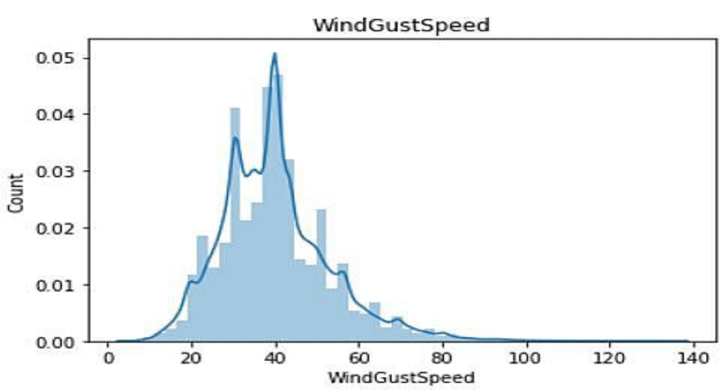


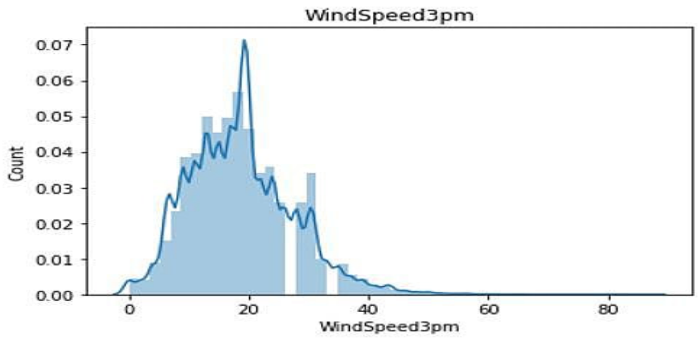


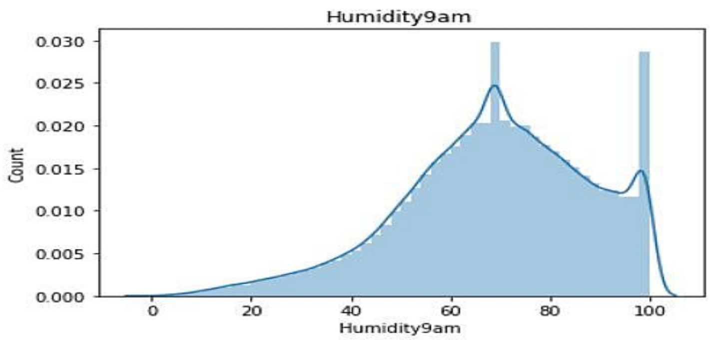

Rainfall
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Count
0
50
100
150
200
250
300
350
Rainfall

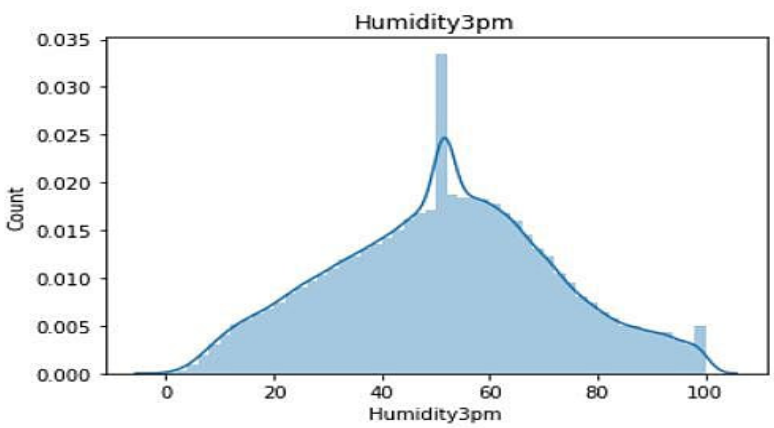

Evaporation
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Count
0
20
40
60
80
100
120
140
Evaporation

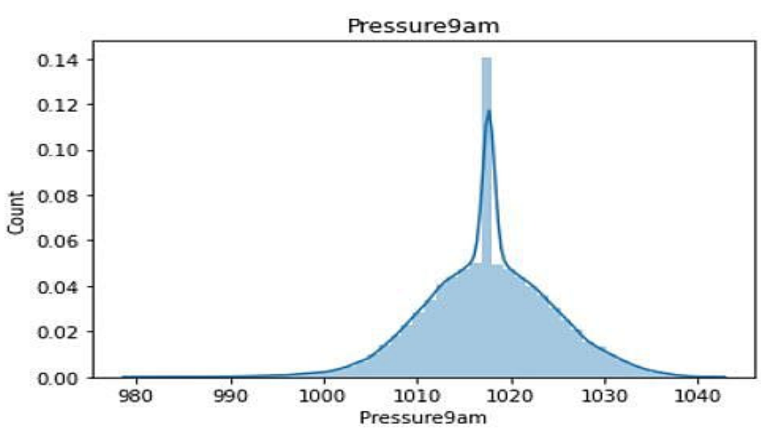

Sunshine
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Count
0
2
4
6
8
10
12
14
16
Sunshine

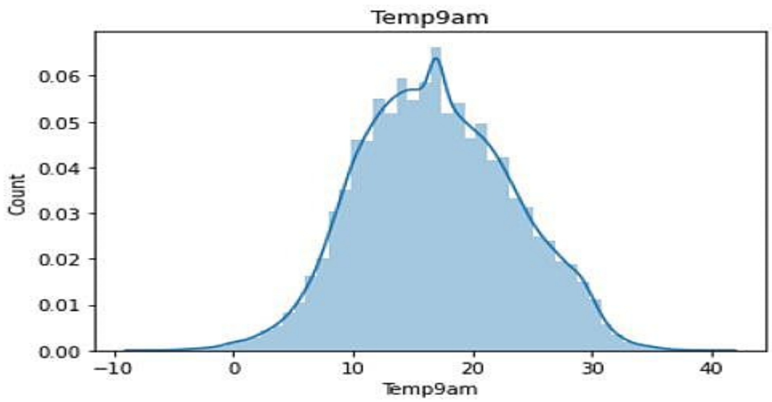

WindGustSpeed
0.05
0.04
0.03
0.02
0.01
0.00
Count
0
20
40
60
80
100
120
140
WindGustSpeed

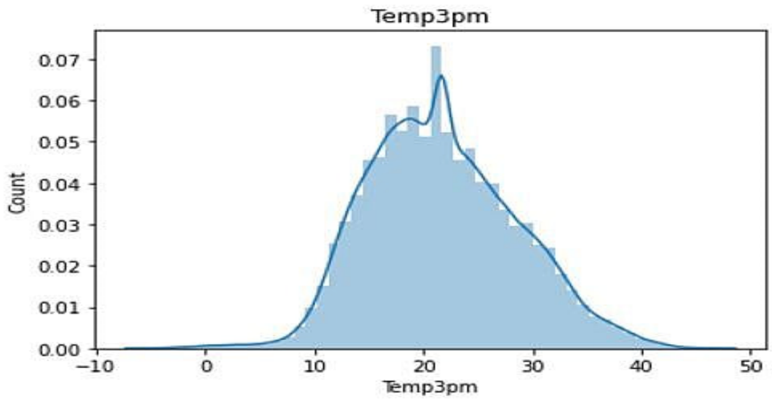

WindSpeed3pm
Count
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
20
40
60
80
WindSpeed3pm


Humidity9am
Count
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
20
40
60
80
100
Humidity9am


Humidity3pm
Count
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
20
40
60
80
100
Humidity3pm

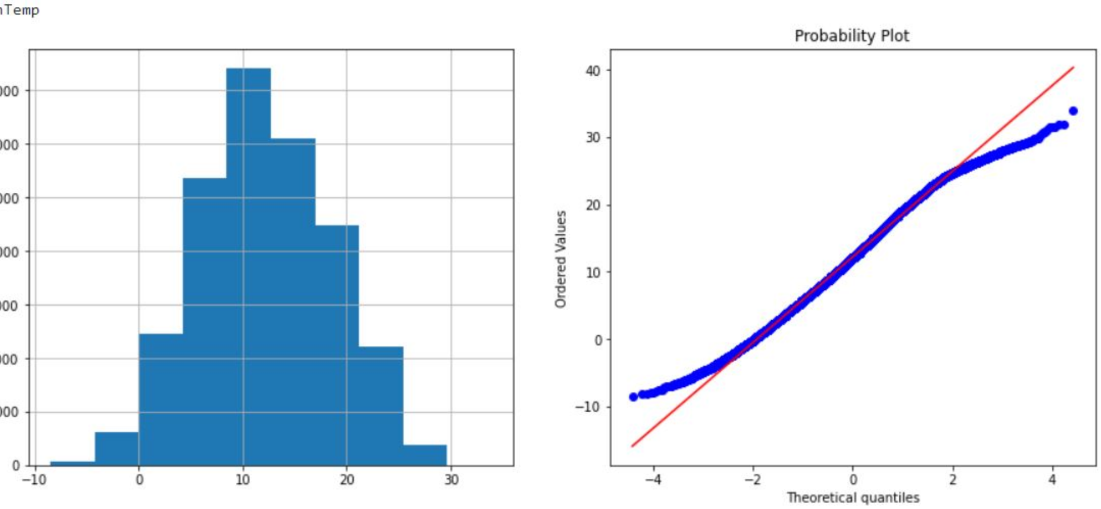

Pressure9am
Count
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
980
990
1000
1010
1020
1030
1040
Pressure9am



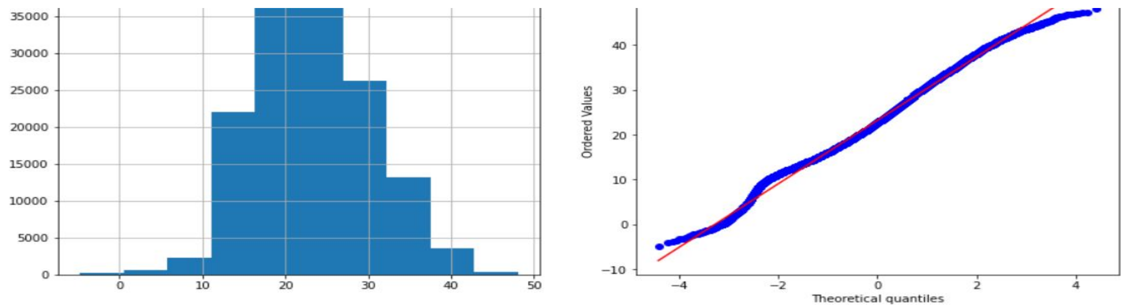



```
import scipy.stats as stats
for feature in continuous_feature:
    print(feature)
    plt.figure(figsize=(15,6))
    plt.subplot(1, 2, 1)
    df[feature].hist()
    plt.subplot(1, 2, 2)
    stats.probplot(df[feature], dist="norm", plot=plt)
    plt.show()
```

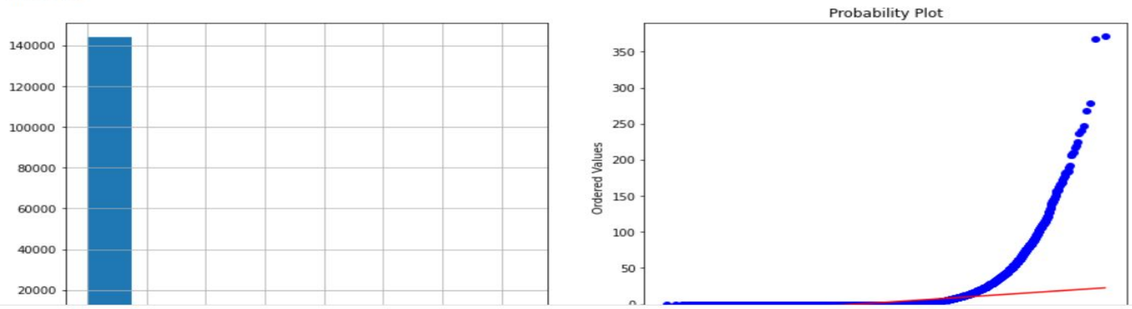
MinTemp

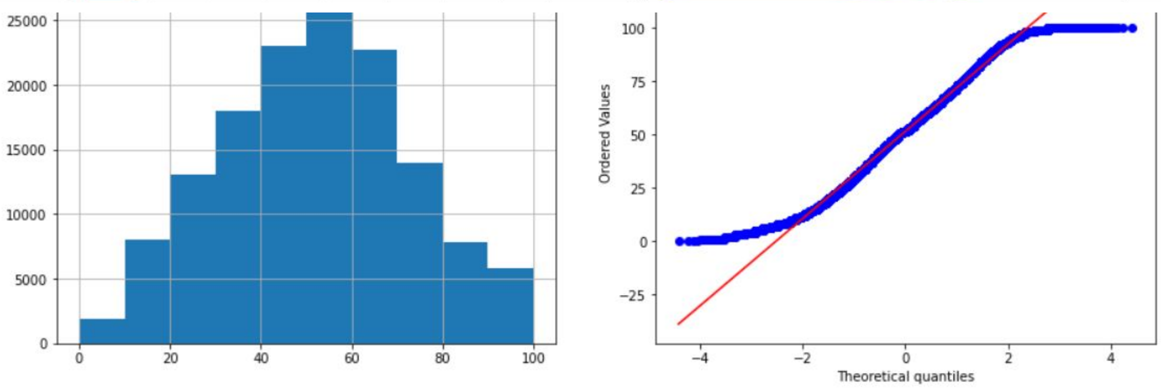


MaxTemp

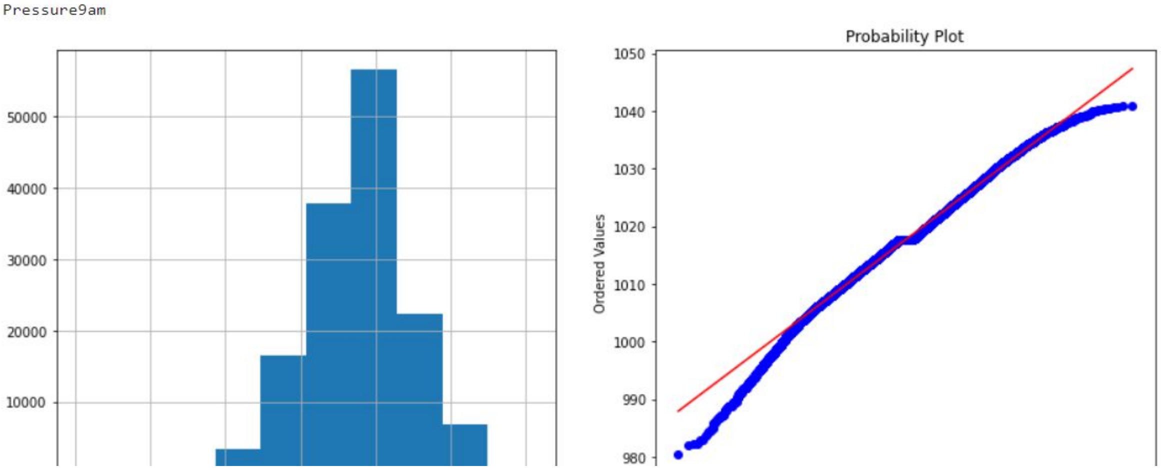


Rainfall





Pressure9am



]:

```python
X = df.iloc[:, :-2].values
print(X)
```

```
[['2008-12-01' 'Delhi' 13.4 ... 4.509930082924903 16.9 21.8]
 ['2008-12-02' 'Delhi' 7.4 ... 4.509930082924903 17.2 24.3]
 ['2008-12-03' 'Delhi' 12.9 ... 2.0 21.0 23.2]
 ...
 ['2017-06-23' 'Uluru' 5.4 ... 4.509930082924903 12.5 26.1]
 ['2017-06-24' 'Uluru' 7.8 ... 2.0 15.1 26.0]
 ['2017-06-25' 'Uluru' 14.9 ... 8.0 15.0 20.9]]
```

]:

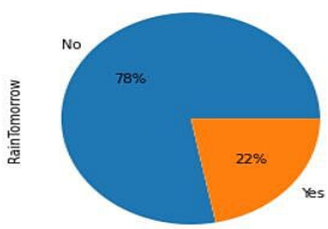
```python
Y = df.iloc[:, -2].values
print(Y)
```

```
['No' 'No' 'No' ... 'No' 'No' 'No']
```

]:

```python
df['RainTomorrow'].value_counts().plot(kind='pie',autopct='%1.0f%%')
```

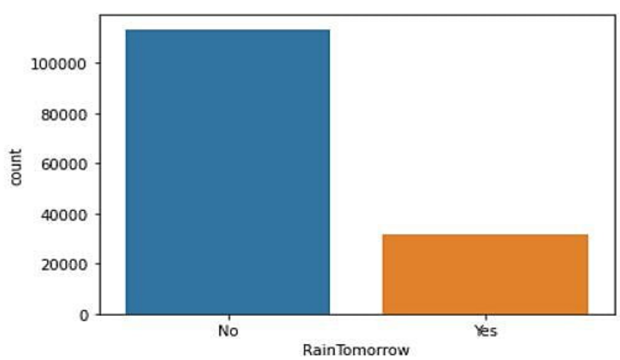
]:



]:

```python
sns.countplot(x = df["RainTomorrow"])
```

23]:



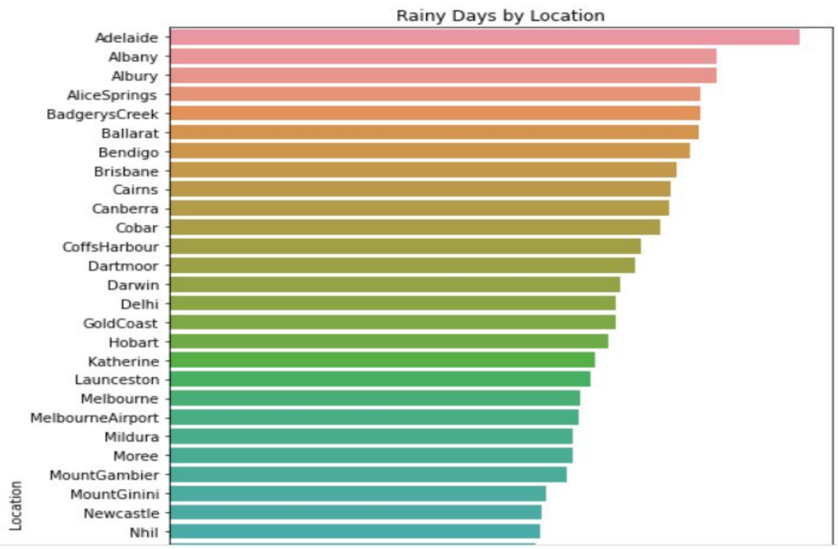
24]:

```python
df.RainToday = df.RainToday.map({'No': 0, 'Yes': 1})
df_rain_by_loc = df.groupby(by='Location').sum()
df_rain_by_loc = df_rain_by_loc[['RainToday']]
df_rain_by_loc.head()
```

24]:

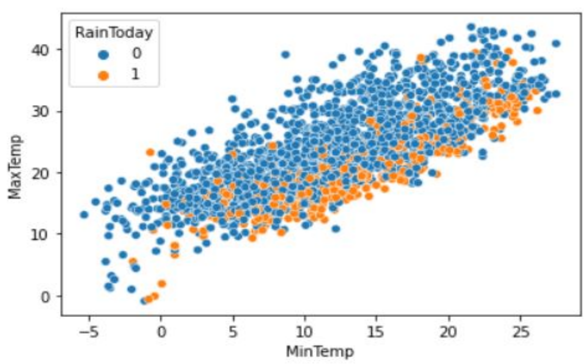
	RainToday
Location	
Adelaide	689
Albany	902
Albury	613
AliceSprings	244
BadgerysCreek	583

5]:
```python
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 12))
sns.barplot(x='RainToday',
            y=df_rain_by_loc.index,
            data=df_rain_by_loc.sort_values('RainToday', ascending=False),
            orient='h'
           )
plt.xlabel('Number of Days')
plt.title('Rainy Days by Location')
plt.tight_layout()
```



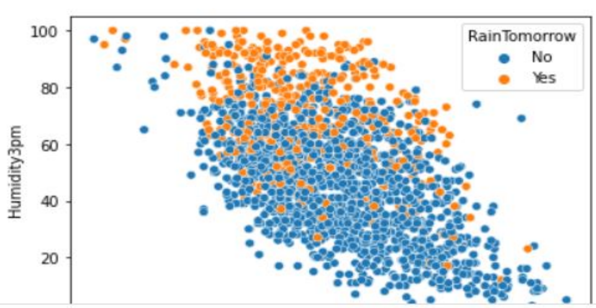
7]:
```python
sns.scatterplot(data=df.sample(2000), x="MinTemp", y="MaxTemp",hue="RainToday")
```

7]:

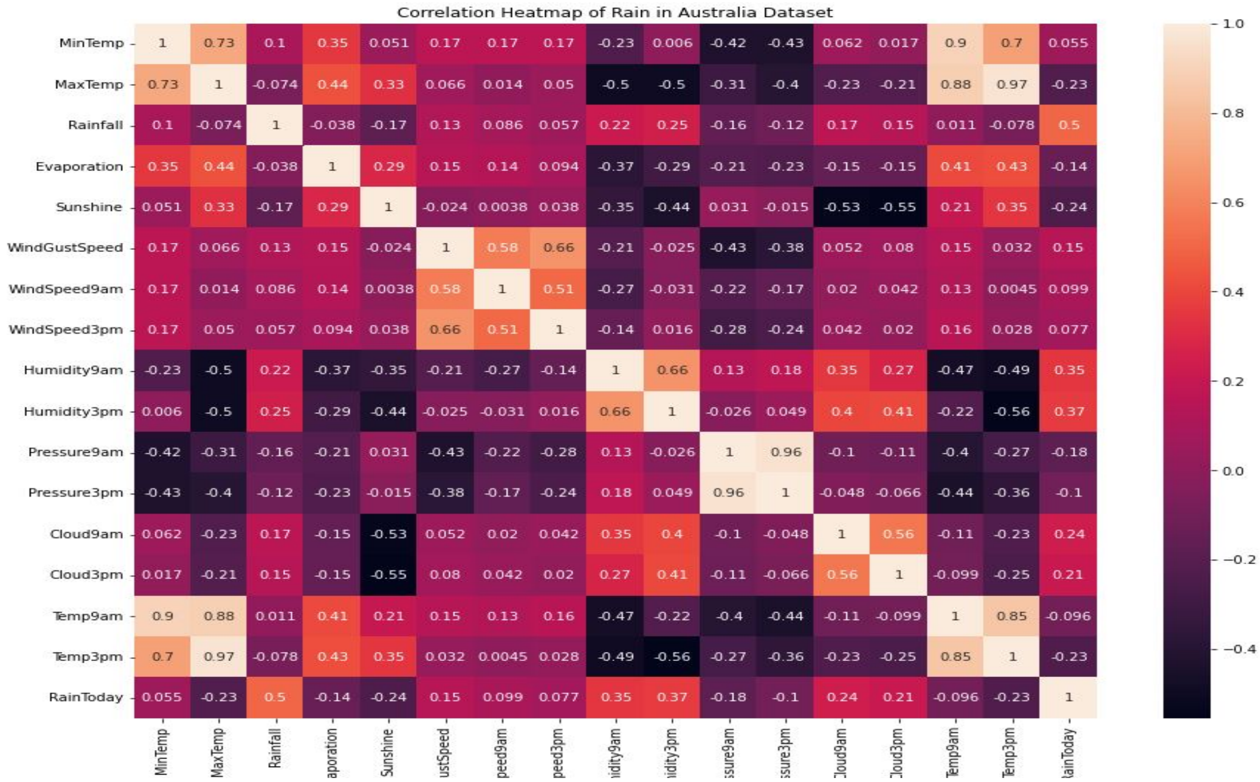


8]:
```python
sns.scatterplot(data=df.sample(2000), x="Temp3pm", y="Humidity3pm",hue="RainTomorrow")
```

8]:



```
cormat = df.corr()
plt.figure(figsize=(16,12))
sns.heatmap(cormat, square=True, annot=True)
plt.title('Correlation Heatmap of Rain in Australia Dataset')
plt.show()
```




```
import pandas as pd
from sklearn.preprocessing import StandardScaler

#Initialise the Scaler
scaler = StandardScaler()
X = df.drop(['Date','Location','WindGustDir', 'WindDir9am', 'WindDir3pm', 'RainToday', 'RainTomorrow'], axis=1)
#To scale data
scaler.fit(X)
```

```
StandardScaler()
```

```
df.head()
```

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	...	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm
0	2008-12-01	Delhi	13.4	22.9	0.6	5.468232	7.611178	W	44.0	W	...	71.0	22.0	1007.7	1007.1
1	2008-12-02	Delhi	7.4	25.1	0.0	5.468232	7.611178	WNW	44.0	NNW	...	44.0	25.0	1010.6	1007.8
2	2008-12-03	Delhi	12.9	25.7	0.0	5.468232	7.611178	WSW	46.0	W	...	38.0	30.0	1007.6	1008.7
3	2008-12-04	Delhi	9.2	28.0	0.0	5.468232	7.611178	NE	24.0	SE	...	45.0	16.0	1017.6	1012.8
4	2008-12-05	Delhi	17.5	32.3	1.0	5.468232	7.611178	W	41.0	ENE	...	82.0	33.0	1010.8	1006.0

5 rows × 23 columns

]:
```
df.columns
```

]:
```
Index(['Date', 'Location', 'MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation',
       'Sunshine', 'WindGustDir', 'WindGustSpeed', 'WindDir9am', 'WindDir3pm',
       'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',
       'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am',
       'Temp3pm', 'RainToday', 'RainTomorrow'],
      dtype='object')
```

]:
```
x = df.drop(["RainTomorrow","Date"], axis = 1)
y = df['RainTomorrow']
```

]:
```
from sklearn import model_selection
```

]:
```
y = df['RainTomorrow']
X = df.drop(['Date','Location','WindGustDir', 'WindDir9am', 'WindDir3pm', 'RainToday', 'RainTomorrow'], axis=1)
x_train, x_test, y_train, y_test = model_selection.train_test_split(X,y,test_size= 0.1, random_state = 0)
```

]:
```
import xgboost
from sklearn import metrics
from sklearn import linear_model
from sklearn import tree
from sklearn.ensemble import RandomForestClassifier
from sklearn import svm
```

]:
```
XGBoost = xgboost.XGBRFClassifier()
XGBoost.fit(x_train,y_train)
```

]:
```
XGBRFClassifier()
```

]:
```
p1 = XGBoost.predict(x_train)
```

]:
```
Randm_Forest = RandomForestClassifier()
Randm_Forest.fit(x_train,y_train)
```

[41]:
```
p2 = Randm_Forest.predict(x_train)
```

[42]:
```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression(solver='lbfgs',class_weight='balanced', max_iter=3000)
lr.fit(x_train,y_train)
```

[42]:
```
LogisticRegression(class_weight='balanced', max_iter=3000)
```

[43]:
```
p3 = lr.predict(x_train)
```

[46]:
```
from sklearn.neighbors import KNeighborsClassifier
kn=KNeighborsClassifier(n_neighbors=3)
kn.fit(x_train,y_train)
```

[46]:
```
KNeighborsClassifier(n_neighbors=3)
```

[47]:
```
p4 = kn.predict(x_train)
```

[50]:
```
from sklearn import tree
dt=tree.DecisionTreeClassifier()
dt.fit(x_train,y_train)
```

[50]:
```
DecisionTreeClassifier()
```

[51]:
```
p5=dt.predict(x_train)
```

[52]:
```
print("XGBoost:", metrics.accuracy_score(y_train,p1))
print("Random_Forest:", metrics.accuracy_score(y_train,p2))
print("Logistic Regression:", metrics.accuracy_score(y_train,p3))
print("K-Nearest neighbors:", metrics.accuracy_score(y_train,p4))
print("Decison Tree:", metrics.accuracy_score(y_train,p5))
```

```
XGBoost: 0.833761095070046
Random_Forest: 0.9999312525780283
Logistic Regression: 0.7800006110881953
```

]:
```python
import pickle
```

]:
```python
pickle.dump(XGBoost,open('XGBoost_rainfall.pkl','wb'))
pickle.dump(Randm_Forest,open('RandomForest_rainfall.pkl','wb'))
pickle.dump(lr,open('LinearRegressor.pkl','wb'))
pickle.dump(kn,open('KNN_rainfall.pkl','wb'))
pickle.dump(dt,open('DecisionTree.pkl','wb'))
```

]:
```python
models = [XGBoost, Randm_Forest, lr, kn, dt]
m = ["XGBoost", "Random_Forest", "Logistic Regression", "K-Nearest neighbors", "Decision Tree"]
```

]:
```python
accuracy = []
accuracy.append(metrics.accuracy_score(y_train,p1))
accuracy.append(metrics.accuracy_score(y_train,p2))
accuracy.append(metrics.accuracy_score(y_train,p3))
accuracy.append(metrics.accuracy_score(y_train,p4))
accuracy.append(metrics.accuracy_score(y_train,p5))
```

]:
```python
d={m[0]:{"Accuracy":accuracy[0]*100},m[1]:{"Accuracy":accuracy[1]*100},m[2]:{"Accuracy":accuracy[2]*100},m[3]:{"Accuracy":accuracy[3]*100}, m[4]:
d=pd.DataFrame(d)
d
```

]:

	XGBoost	Random_Forest	Logistic Regression	K-Nearest neighbors	Decision Tree
Accuracy	83.37611	99.993125	78.000061	89.639764	99.995417

]:
```python
max_accuracy=max(accuracy)
model_m=m[accuracy.index(max_accuracy)]
print(model_m, "has the maximum Training accuracy")
print("Max accuracy ",max_accuracy*100,"%")
```

```
Decision Tree has the maximum Training accuracy
Max accuracy  99.99541683853522 %
```

]:
```python
print(m[accuracy.index(max_accuracy)])
```

In [60]:
```python
model=models[accuracy.index(max_accuracy)]
y_pred=model.predict(x_test)
```

In [61]:
```python
y_predict={"Predicted value":y_pred,"Actual value":y_test}
y_predict=pd.DataFrame(y_predict)
y_predict[:10].style.hide_index()
```

Out[61]:

Predicted value	Actual value
Yes	Yes
No	Yes
No	No
No	No
No	No
No	No
No	No
No	Yes
Yes	Yes
No	No

In [62]:
```python
from sklearn.metrics import classification_report as cr
```

In [63]:
```python
print(cr(y_test, y_pred))
```

```
              precision    recall  f1-score   support

          No       0.86      0.86      0.86     11328
         Yes       0.50      0.52      0.51      3218

    accuracy                           0.78     14546
   macro avg       0.68      0.69      0.68     14546
weighted avg       0.78      0.78      0.78     14546
```

[65]:
```
acc = []
from sklearn.metrics import accuracy_score
for i in models:
  pr=i.predict(x_test)
  acc.append(accuracy_score(y_test, pr))
```

[66]:
```
d={m[0]:{"Accuracy":acc[0]*100},m[1]:{"Accuracy":acc[1]*100},m[2]:{"Accuracy":acc[2]*100},m[3]:{"Accuracy":acc[3]*100}, m[4]:{"Accuracy":acc[4]*100}}
d=pd.DataFrame(d)
d
```

[66]:

	XGBoost	Random_Forest	Logistic Regression	K-Nearest neighbors	Decision Tree
Accuracy	82.875017	85.102434	78.145195	82.380036	78.097071

[67]:
```
max_acc=max(acc)
model_m=m[acc.index(max_acc)]
print(model_m, "has the maximum Testing accuracy")
print("Max accuracy ",max_acc*100,"%")
```

```
Random_Forest has the maximum Testing accuracy
Max accuracy  85.1024336587378 %
```

[68]:
```
print(m[acc.index(max_acc)])
```

```
Random_Forest
```

[69]:
```
model=models[acc.index(max_acc)]
y_pred=model.predict(x_test)
```

[70]:
```
y_predict={"Predicted value":y_pred,"Actual value":y_test}
y_predict=pd.DataFrame(y_predict)
y_predict[:10].style.hide_index()
```

[ ]:

Predicted value	Actual value
No	Yes
No	Yes
No	No
No	No
No	No
No	No
No	No
No	Yes
Yes	Yes
No	No

[ ]:
```
from sklearn.metrics import classification_report as cr
```

[ ]:
```
print(cr(y_test, y_pred))
```

```
              precision    recall  f1-score   support

          No       0.87      0.95      0.91     11328
         Yes       0.75      0.49      0.59      3218

    accuracy                           0.85     14546
   macro avg       0.81      0.72      0.75     14546
weighted avg       0.84      0.85      0.84     14546
```

[ ]:
```
filen = "/content/KNN_rainfall.pkl"
model = pickle.load(open(filen, 'rb'))
```

```
[ ]: MinTemp = float(input())
MaxTemp = float(input())
Rainfall = float(input())
Evaporation = float(input())
Sunshine = float(input())
WindGustSpeed = float(input())
WindSpeed9am = float(input())
WindSpeed3pm = float(input())
Humidity9am = float(input())
Humidity3pm = float(input())
Pressure9am = float(input())
Pressure3pm = float(input())
Cloud9am = float(input())
Cloud3pm = float(input())
Temp9am = float(input())
Temp3pm = float(input())
lis = [[MinTemp, MaxTemp, Rainfall, Evaporation, Sunshine, WindGustSpeed, WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm, Pressure9am, Pressure3

pred = model.predict(lis)
pred
```

UserAcceptanceTesting

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Project Exploratory Analysis of Rainfall data in India for Agriculture at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

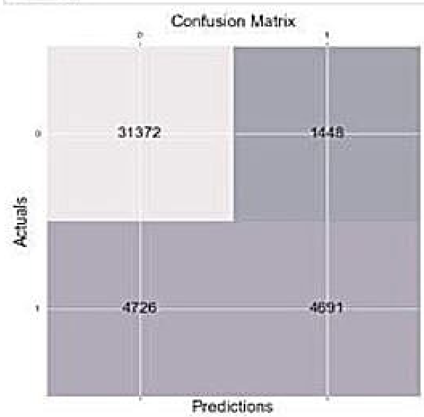
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	0	0	0	0
Duplicate	0	0	0	0	0
External	0	0	0	0	0
Fixed	0	0	0	0	0
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	0	0	0	0	0

3. Test Case Analysis

Section	Total Cases	Not Tested	Fail	Pass
Home Page	2	0	0	2
Predict Page	4	0	0	4

9.RESULTS: Performance Metrics

S.No.	Parameter	Values	Screenshot

1.	Metrics	**Classification Model: Random Forest**
		
		Confusion Matrix –
		[[31372 1448]
		[4726 4691]]
		
		Accuracy Score-
		0.8538248455145963
		
		Classification Report –
		Accuracy: 0.8538248455145963
		Precision: 0.7641309659553673
		Recall: 0.49814165870234683
		F1-score: 0.6031113396760092
			Random forest Confusion matrix
			conf_matrix = metrics.confusion_matrix(y_test,t1)
			
			fig,ax = plt.subplots(figsize=(7.5,7.5))
			ax.matshow(conf_matrix,alpha=0.3)
			for i in range(conf_matrix.shape[0]):
			  for j in range(conf_matrix.shape[1]):
			    ax.text(x=j, y=i, s=conf_matrix[i,j], va ='center', ha='center',size='xx-larg
			plt.xlabel('Predictions',fontsize=18)
			plt.ylabel('Actuals',fontsize=18)
			plt.title('Confusion Matrix',fontsize=18)
			plt.show()
			
			t1 = Rand_forest.predict(X_test_scaled)
			
			print("Rand_forest:",metrics.accuracy_score(y_test,t1))
			Rand_forest: 0.8538248455145963

			print("*"*10, "Classification Report", "*"*10) print("-"*30) print(classification_report(y_test, t1)) print("-"*30)

2	Tunethe Model	Hyperparameter Tunfng & Validartion Method -	Hy p er-parameter Tu nin y fr,o.m skleaM1. e111semble. impor R,:uu:rom ForestRegN!s s.o:r rf = Rando ores:tEtegre ssor(randH_state • 2) frolll pprint import pprint Loo• d pckt'(hm!ter used by out" curre H forest pr n ('Para eters current! use:\n") pprinc(ri.get_parains()) Para eters currently in use: {'bootstra p': Trul!!, ·ccp_alplla': a.e, 'eriter.i,0111 •; 'mise •, •raa _deptch": None, 'r•.:P!_features': 'auto', 'iil:IX_leaf_11odes' one , '111a _sal1ples' : None, 'nrln_ 111ril1:y_dee,rea,se•: 0.0, ·lllin_ ority_split ·: None, '■in_saaples_leaf': 1, 'ntin_s.i pl!!s_split ·•: 2., 'llll n_welght_fractlon_leaf': 8,e, 'n_est!mators' : 100, 'n_jobs' : None, 'oob_scor,e': :False, '·randOIII_S 3 e' : 42, 'verbos ·: e. 'war,111_star ': False}

10.ADVANTAGES&DISADVANTAGES:

Advantages:

- As Weather conditions have been changing for the time being this helps people to know abouttherainfallprediction
- To avoid unnecessary floods by opening dams with the help of rainfall prediction
- Farmers and fisherman will get the most advantage of these rainfall details so that we they canplanaccordingly
- During the monsoon days it helps the government to find the evacuation areas to avoid loss of human life and costly things.

Disadvantages:

- As the data was collected from limited places so it helps only for the people who located in thoseareas.
- In case the data was collected being wrong the algorithm will produce the wrong prediction

● As of now have collecting only a limited number of data set, In feature, we will make the algorithm to work worldwide

11. CONCLUSION:

Floods are the most common natural disasters and have widespread effect flood forecasting is hence an important research area and various possible solutions have been presented in literatureto this end the input data were selected based on a correlation and uncertainty analysis oftherainfall and flood data and a classification based real-time flood prediction model was developed heavy rainfall that may occur in urban areas was analyzed in advance and the expected range of an urban flood was predicted in real time using the proposed model.

12. FUTURE SCOPE:

With the change in climatic conditions and rainfall patterns this can lead to flash floods causing catastrophic damage to the environment.The system can be further enhanced with a flood prediction system along with rainfall prediction. Evacuation areas can be included along with the flood prediction system in such a way that the system recommends the user as well as to the community if there might be an occurrence of flood. A recommendation system integrated with the prediction system shall sound good for society.