EXPLORATORY ANALYSIS OF RAIN FALL DATA IN INDIA FOR AGRICULTURE

18PF15- PROFESSIONAL READINESS FOR INNOVATION,EMPLOYABILITY AND ENTREPRENEURSHIP

REPORT

Submitted by

Akalya S	717819P102
Narmadha K	717819P119
Narmatha P	717819P120
Pradhyusha S	717819P123

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

KARPAGAM
COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
Accredited by NAAC with 'A' grade
Autonomous | Affiliated to Anna University
(An ISO 9001:2015 and ISO 14001:2015 Certified Institution)

ANNA UNIVERSITY: CHENNAI - 600 025

NOVEMBER 2022

1. INTRODUCTION

1.1 Project Overview

Rainfall has been a major concern these days. Weather conditions have been changing for time being. Rainfall forecasting is important otherwise, it may lead to many disasters. Irregular heavy rainfall may lead to the destruction of crops, heavy floods that can cause harm to human life. It is important to exactly determine the rainfall for effective use of water resources, crop productivity, and pre-planning of water structures.

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. We will train and test the data with these algorithms. From this best model is selected and saved in pkl format. Once the model is saved, we integrate it with flask application and also deploy the model in IBM.

1.2 Purpose

Rainfall forecasting is very important because heavy and irregular rainfall can have many impacts like destruction of crops and farms, damage of property so a better forecasting model is essential for an early warning that can minimize risks to life and property and also managing the agricultural farms in better way. This prediction mainly helps farmers and also water resources can be utilized efficiently. Rainfall prediction is a challenging task and the results should be accurate. There are many hardware devices for predicting rainfall by using the weather conditions like temperature, humidity, pressure. These traditional methods cannot work in an efficient way so by using machine learning techniques we can produce accurate results. We can just do it by having the historical data analysis of rainfall and can predict the rainfall for future seasons. We can apply many techniques like classification, regression according to the requirements and also we can calculate the error between the actual and prediction and also the accuracy. Different techniques produce different accuracies so it is important to choose the right algorithm and model it according to the requirements.

2. LITERATURE SURVEY

2.1 Existing problem

The existing system uses previous data and predicts the weather which may contain null values or incorrect values which causes the output with less accuracy or error

2.2 References

1. Amanatidis, G.T., A.G. Paliatsos, C.C. Repapis and J.G. Bartzis, 1993. “Decreasing precipitation trend in the Marathon area”, Greece. Int. J. Climatol., pp: 191-201.
DOI: 10.1002/joc.3370130205

2. Aronica, G., M. Cannarozzo and L. Noto, 2002. “Investigating the changes in extreme rainfall series recorded in an urbanised area”. Water Sci. Technol., pp: 49-54.
PMID: 11888183

3. Aung, T., A. Singh and U. Prasad, 2009. “Sea level threat in Tuvalu”. Am. J. Applied Sci., pp: 1169-1174.
DOI: 10.3844/ajassp.2009.1169.1174

4. Behringer, J., R. Buerki and J. Fuhrer, 2000. “Participatory integrated assessment of adaptation to climate change in Alpine tourism and mountain agriculture”. Integrated Assess., pp: 331-338.
DOI: 10.1023/A:1018940901744

5. Bonaccorso, B., A. Cancelliere and G. Rossi, 2005. “Detecting trends of extreme rainfall series in Sicily”. Advan. Geosci., pp: 7-11.
DOI: 10.5194/ADGEO-2-7-2005

6. Brunetti, M., L. Buffoni, M. Maugeri and T. Nanni, 2000.” Precipitation intensity trends in Northern Italy”. Int. J. Climatol., pp: 1017-1031.
DOI: 10.1002/1097-0088(200007)20:9<1017::AID-JOC515>3.0.CO;2-S

2.3 Problem Statement Definition

Customer Problem Statement :

To forecast the rainfall in India for agriculturist as the weather conditions have been changing for time being. Rainfall forecasting is important otherwise, it may lead to many disasters. Irregular heavy rainfall may lead to the destruction of crops, heavy floods that can cause harm to human life and so it is important to exactly determine the rainfall for effective use of water resources, crop productivity, and pre-planning of water structures.

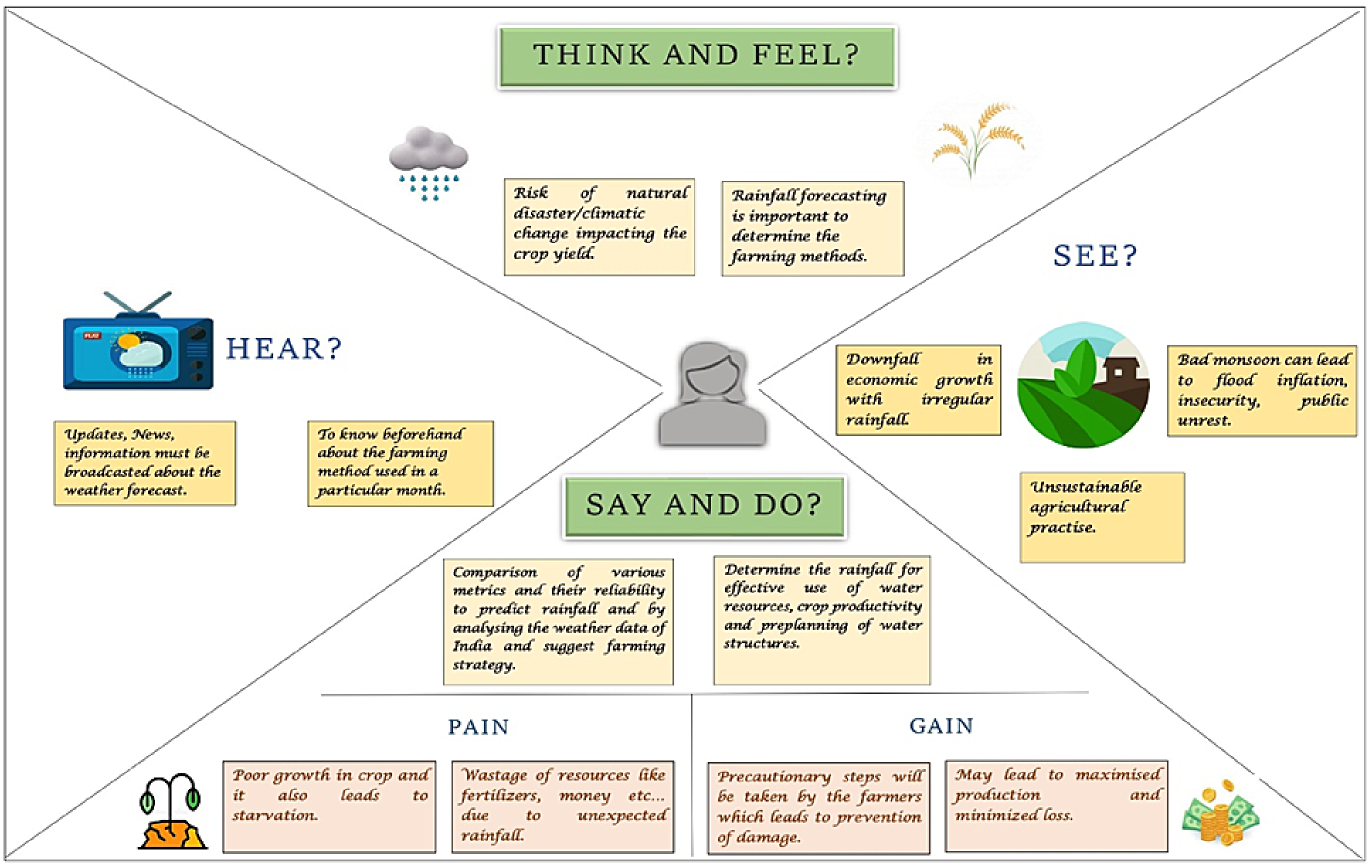
I am	Agriculturists or farmers in India
I'm trying to	Cultivate different crops with their respective seasons and weather conditions.
But	Bad weather may lead to many disasters like irregular heavy rainfall or no rainfall may lead to the destruction of crops, heavy floods that can cause harm to human life.
Because	The weather or rainfall is not forecasted.
Which makes	Crops destruction, Rising demands for food, Biodiversity loss, reduce farm productivity, floods or drought occurrence and many.

Example:

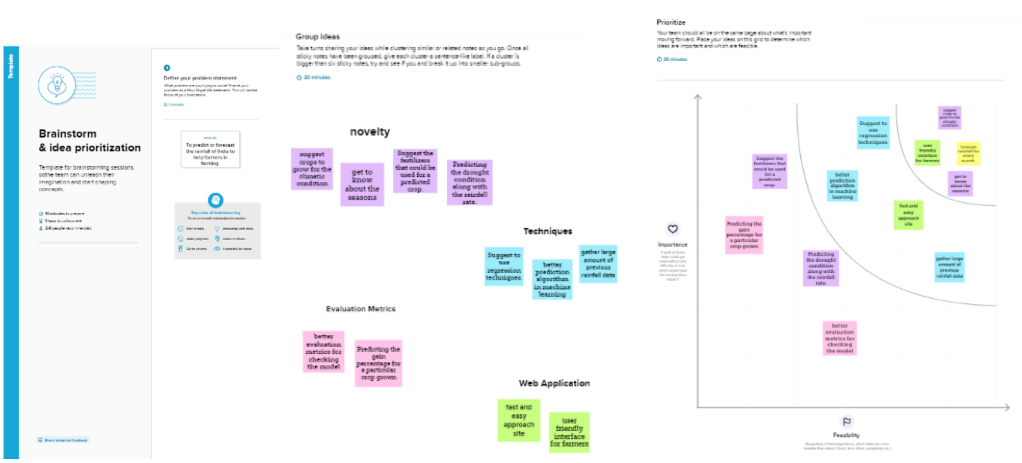
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

EMPATHY MAP



3.2 Ideation & Brainstorming



3.3 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	To exactly determine the rainfall in India for effective use of water resources, crop productivity, and pre-planning of water structures.
2.	Idea / Solution description	The idea is to predict the rainfall in a region by analyzing the weather data in India from 1998-2015.
3.	Novelty / Uniqueness	Here with the given region we suggest the agriculturalist for the best yielding crops to grow.
4.	Social Impact / Customer Satisfaction	Rainfall has been a major concern these days. Weather conditions have been changing for time being. Rainfall forecasting is important otherwise, it may lead to many disasters. Irregular heavy rainfall may lead to the destruction of crops, heavy floods that can cause harm to human life.
5.	Business Model (Revenue Model)	The target of this project is to forecast the rainfall in India and the target user is the farmers. It provides a better service in predicting the rainfall and the crops to grow for better yield.
6.	Scalability of the Solution	It aims to acquire a better scalability with highest accuracy achieved in prediction, user friendly interface, good number of web app users, without investing lot of resources and money, ease of use and works with better functionality.

3.4 Problem Solution fit

Define CS, fit into CC

CUSTOMER SEGMENT(S) CS

- ✔ Farmers.

6. CUSTOMER CONSTRAINTS CC

- ✔ Budget
- ✔ No cash

5. AVAILABLE SOLUTIONS AS

- ✔ With available weather forecasting sites like Accuweather, windy and the weather channel

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- ✔ To predict the rainfall and the crops that could be grown on a particular region based on the rainfall that that has been predicted.

9. PROBLEM ROOT CAUSE RC

- ✔ The unpredictable rainfall and climatic changes are the root cause of the problem.
- ✔ The customer has to do this to prevent their crops and land and to prevent the loss that occurs due to the problem.

7. BEHAVIOUR BE

- ✔ Directly related: find the right crop that could be grown on their region, predict the benefits.
- ✔ Indirectly associated: customers will have a relaxation and inner peace.

Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR

- ✓ Seeing their neighbors using our application, planting/growing the crops and getting benefitted with the huge amount of profit.

10. YOUR SOLUTION SL

- ✓ Will be predicting the rainfall with the region as input and will be suggesting the crop for it.

8.CHANNELS of BEHAVIOUR CH

8.1 ONLINE

- ✓ They would search in for online weather prediction site

8.2 OFFLINE

- ✓ Try to predict the weather using traditional method, ask suggestion from their near ones.

4. EMOTIONS: BEFORE / AFTER EM

- ✓ Dejected, insecure > confident, in control, satisfactory.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Check weather	Customer can check the rainfall by giving region and month as inputs.
FR-2	Suggest crop	With the predicted rainfall, the crops would be suggested to the farmers for higher productivity

4.2 Non-Functional requirements

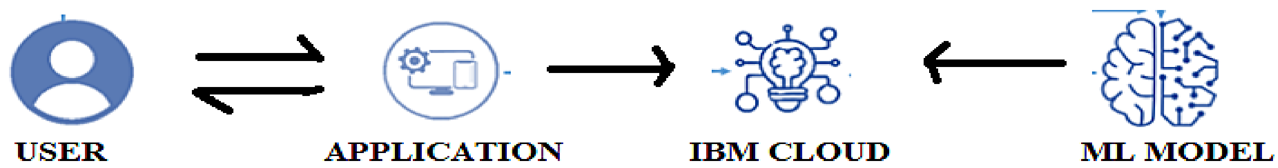
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The farmers and other people can easily use the application and it is user friendly no prior knowledge is required for using it.
NFR-2	**Security**	All data will be protected against malware attacks.
NFR-3	**Reliability**	The system will provide the prediction without any errors.
NFR-4	**Performance**	The expected output will be produces immediately to the user without much delay.
NFR-5	**Availability**	The system would be available 24/7
NFR-6	**Scalability**	The system would be available on web application and any user can login and use it without any disruptions.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (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.



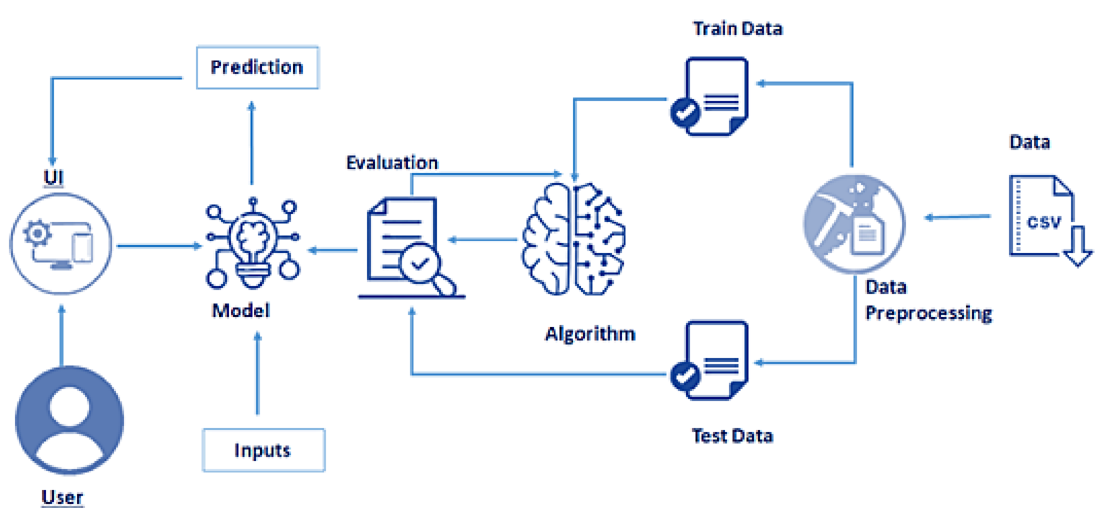
1. User enters the region into the web application to predict the rainfall rate
2. The input data is sent to the cloud
3. The Machine learning model deployed in the cloud predicts the rainfall
4. And finally the predicted rainfall rate in mm and crops are suggested as output.

5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (web user)	Check Weather	USN-1	As a customer, I can check the rainfall by giving the region as input.	I can view the predicted rainfall status by entering information	High	Sprint-2
	Suggested Crop	USN-2	As a customer, With the predicted rainfall I can view the suggested crops for higher productivity	I can view the suggested crops with the predicted rainfall	High	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

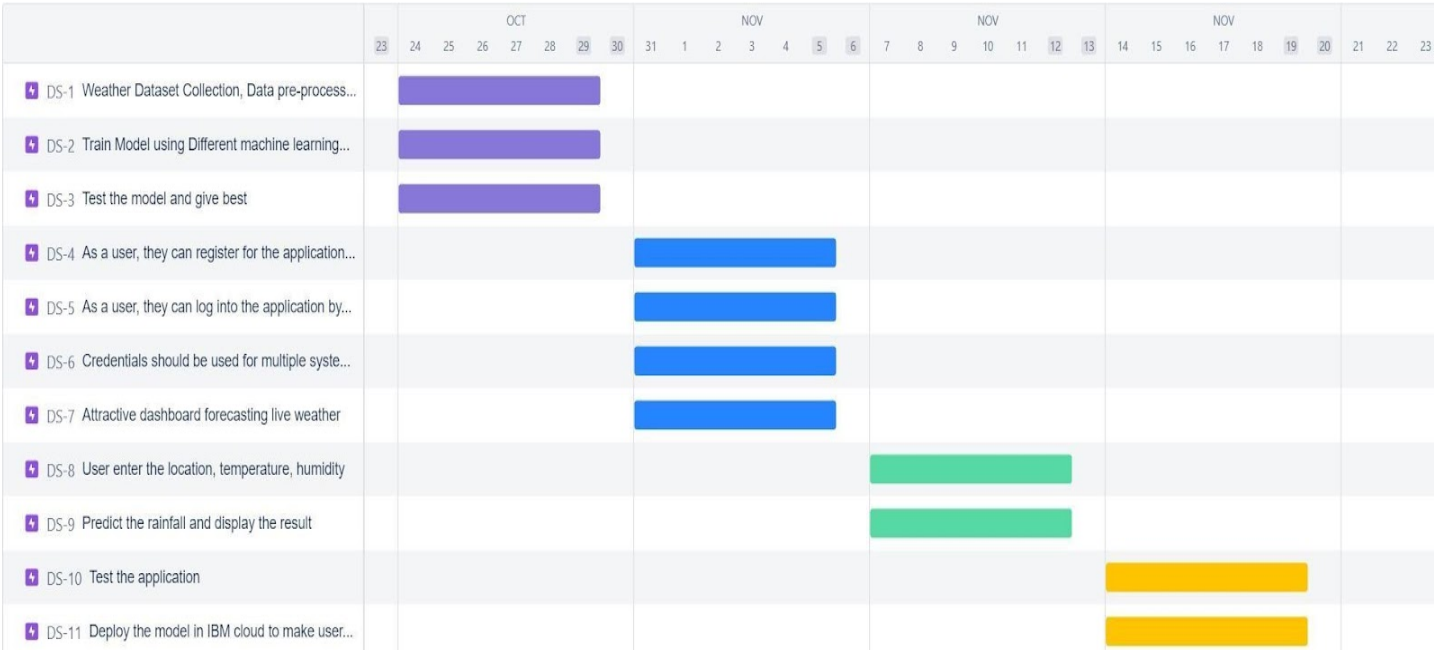
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Nov 2022	20	15 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	15 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	15 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	15 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 Prediction ML Model (Dataset)	USN-1	Weather Dataset Collection, Data preprocessing, Data Visualization.	5	High	Akalya S, Narmadha K, Narmatha P, Pradhyusha S
Sprint-1		USN-2	Train Model using Different machine learning Algorithms	5	High	Akalya S, Narmadha K, Narmatha P, Pradhyusha S
Sprint-1		USN-3	Test the model and give best	10	High	Akalya S, Narmadha K, Narmatha P, Pradhyusha S
Sprint-2	Registration	USN-4	As a user, they can register for the application through Gmail. Password is set up.	5	Medium	Akalya S, Narmadha K, Narmatha P, Pradhyusha S

Sprint-2	Login	USN-5	As a user, they can log into the application by entering email & password	4	Medium	Akalya S, Narmadha K, Narmatha P, Pradhyusha S
Sprint-2		USN-6	Credentials should be used for multiple systems and verified	4	Medium	Akalya S, Narmadha K, Narmatha P, Pradhyusha S
Sprint-2	Dashboard	USN-7	Attractive dashboard forecasting live weather	6	Low	Akalya S, Narmadha K, Narmatha P, Pradhyusha S
Sprint-3	Rainfall Prediction	USN-8	User enter the location, temperature, humidity	10	High	Akalya S, Narmadha K, Narmatha P, Pradhyusha S
Sprint-3		USN-9	Predict the rainfall and display the result	10	High	Akalya S, Narmadha K, Narmatha P, Pradhyusha S
Sprint-4	Testing	USN-10	Test the application	10	High	Akalya S, Narmadha K, Narmatha P, Pradhyusha S
Sprint-4	Deploy Model	USN-11	Deploy the model in IBM cloud to make user friendly application	10	High	Akalya S, Narmadha K, Narmatha P, Pradhyusha S

6.3 Reports from JIRA



7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

```
<!DOCTYPE html>

<html lang="en" dir="ltr">

  <head>

    <meta charset="utf-8">

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

    <title>Rainy Brain</title>

    <link rel="stylesheet" href={{url_for('static',filename='style1.css')}}>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.14.0/css/all.min.css">

  </head>

  <body background="https://wallpaperaccess.com/full/701614.jpg">

    <section>

      <input type="checkbox" id="check">

      <header>

        </div>

        <h2><a href="#" class="logo">Weather Forcast</a></h2>

        <div class="navigation">

          <a href="/predict">Predictor</a>

        </div>

        <label for="check">

        <i class="fas fa-bars menu-btn"></i>

        <i class="fas fa-times close-btn"></i>

        </label>

      </header>
```

```
      <div class="content" style="margin-top: 8%;">
        <div class="info">
          <h2><br><span></span></h2>
          <p> </p>
        </div>
      </div>
    </section>
  </body>
</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">
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;400;500;600;700;800;900&display=swap" rel="stylesheet">
    <link rel="stylesheet" href={{url_for('static',filename='after_rainy.css')}}>
    <title>Rainy Day</title>
</head>
<body background=../static/sunny.png">
    <div class="rainyimg">
        <img src="../static/sunny.png"   style="height: 550px; width: 850px; margin-left: 22%">
    </div>
</body>
</html>
```

7.2 Feature 2

```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">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;400;500;600;700;800;900&display=swap" rel="stylesheet">

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

    <title>Rainy Day</title>

</head>

<body>

    <div class="rainyimg">

        <img src="../static/rainy.png" style="height: 550px; width: 850px; margin-left: 22%">

    </div>

</body>
</html>
```

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
import scipy.stats as stats
from sklearn.model_selection import train_test_split
from collections import Counter
from imblearn.over_sampling import SMOTE
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
from sklearn import metrics
from sklearn.ensemble import RandomForestClassifier
```

```
from catboost import CatBoostClassifier
from xgboost import XGBClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
import joblib
df = pd.read_csv("weatherAUS.csv")
pd.set_option("display.max_columns", None)
df
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)))
# Handle Missing Values
df.isnull().sum()*100/len(df)
print(numerical_feature)
def randomsampleimputation(df, variable):
    df[variable]=df[variable]
    random_sample=df[variable].dropna().sample(df[variable].isnull().sum(),random_state=0)
    random_sample.index=df[df[variable].isnull()].index
    df.loc[df[variable].isnull(),variable]=random_sample
randomsampleimputation(df, "Cloud9am")
randomsampleimputation(df, "Cloud3pm")
randomsampleimputation(df, "Evaporation")
randomsampleimputation(df, "Sunshine")
df
corrmat = df.corr(method = "spearman")
plt.figure(figsize=(20,20))
#plot heat map
g=sns.heatmap(corrmat,annot=True)
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()
```

```
#A for loop is used to plot a boxplot for all the continuous features to see the outliers
for feature in continuous_feature:
    data=df.copy()
    sns.boxplot(data[feature])
    plt.title(feature)
    plt.figure(figsize=(15,15))
for feature in continuous_feature:
    if(df[feature].isnull().sum()*100/len(df))>0:
        df[feature] = df[feature].fillna(df[feature].median())
df.isnull().sum()*100/len(df)
discrete_feature
def mode_nan(df,variable):
    mode=df[variable].value_counts().index[0]
    df[variable].fillna(mode,inplace=True)
mode_nan(df,"Cloud9am")
mode_nan(df,"Cloud3pm")
df["RainToday"] = pd.get_dummies(df["RainToday"], drop_first = True)
df["RainTomorrow"] = pd.get_dummies(df["RainTomorrow"], drop_first = True)
df
for feature in categorical_feature:
    print(feature, (df.groupby([feature])["RainTomorrow"].mean().sort_values(ascending = False)).index)
windgustdir = {'NNW':0, 'NW':1, 'WNW':2, 'N':3, 'W':4, 'WSW':5, 'NNE':6, 'S':7, 'SSW':8, 'SW':9, 'SSE':10,
      'NE':11, 'SE':12, 'ESE':13, 'ENE':14, 'E':15}
winddir9am = {'NNW':0, 'N':1, 'NW':2, 'NNE':3, 'WNW':4, 'W':5, 'WSW':6, 'SW':7, 'SSW':8, 'NE':9, 'S':10,
      'SSE':11, 'ENE':12, 'SE':13, 'ESE':14, 'E':15}
winddir3pm = {'NW':0, 'NNW':1, 'N':2, 'WNW':3, 'W':4, 'NNE':5, 'WSW':6, 'SSW':7, 'S':8, 'SW':9, 'SE':10,
      'NE':11, 'SSE':12, 'ENE':13, 'E':14, 'ESE':15}
df["WindGustDir"] = df["WindGustDir"].map(windgustdir)
df["WindDir9am"] = df["WindDir9am"].map(winddir9am)
df["WindDir3pm"] = df["WindDir3pm"].map(winddir3pm)
df["WindGustDir"] = df["WindGustDir"].fillna(df["WindGustDir"].value_counts().index[0])
df["WindDir9am"] = df["WindDir9am"].fillna(df["WindDir9am"].value_counts().index[0])
df["WindDir3pm"] = df["WindDir3pm"].fillna(df["WindDir3pm"].value_counts().index[0])
df.isnull().sum()*100/len(df)
df1 = df.groupby(["Location"])["RainTomorrow"].value_counts().sort_values().unstack()
df1
df1[1].sort_values(ascending = False)
df1[1].sort_values(ascending = False).index
len(df1[1].sort_values(ascending = False).index)
location = {'Portland':1, 'Cairns':2, 'Walpole':3, 'Dartmoor':4, 'MountGambier':5,
      'NorfolkIsland':6, 'Albany':7, 'Witchcliffe':8, 'CoffsHarbour':9, 'Sydney':10,
      'Darwin':11, 'MountGinini':12, 'NorahHead':13, 'Ballarat':14, 'GoldCoast':15,
      'SydneyAirport':16, 'Hobart':17, 'Watsonia':18, 'Newcastle':19, 'Wollongong':20,
```

```
    'Brisbane':21, 'Williamtown':22, 'Launceston':23, 'Adelaide':24, 'MelbourneAirport':25,
    'Perth':26, 'Sale':27, 'Melbourne':28, 'Canberra':29, 'Albury':30, 'Penrith':31,
    'Nuriootpa':32, 'BadgerysCreek':33, 'Tuggeranong':34, 'PerthAirport':35, 'Bendigo':36,
    'Richmond':37, 'WaggaWagga':38, 'Townsville':39, 'PearceRAAF':40, 'SalmonGums':41,
    'Moree':42, 'Cobar':43, 'Mildura':44, 'Katherine':45, 'AliceSprings':46, 'Nhil':47,
    'Woomera':48, 'Uluru':49}
df["Location"] = df["Location"].map(location)
df["Date"] = pd.to_datetime(df["Date"], format = "%Y-%m-%dT", errors = "coerce")
df["Date_month"] = df["Date"].dt.month
df["Date_day"] = df["Date"].dt.day
df
corrmat = df.corr()
plt.figure(figsize=(20,20))
#plot heat map
g=sns.heatmap(corrmat,annot=True)
sns.countplot(df["RainTomorrow"])
df
for feature in continuous_feature:
  data=df.copy()
  sns.boxplot(data[feature])
  plt.title(feature)
  plt.figure(figsize=(15,15))
for feature in continuous_feature:
  print(feature)
IQR=df.MinTemp.quantile(0.75)-df.MinTemp.quantile(0.25)
lower_bridge=df.MinTemp.quantile(0.25)-(IQR*1.5)
upper_bridge=df.MinTemp.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
df.loc[df['MinTemp']>=30.45,'MinTemp']=30.45
df.loc[df['MinTemp']<=-5.95,'MinTemp']=-5.95
IQR=df.MaxTemp.quantile(0.75)-df.MaxTemp.quantile(0.25)
lower_bridge=df.MaxTemp.quantile(0.25)-(IQR*1.5)
upper_bridge=df.MaxTemp.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
df.loc[df['MaxTemp']>=43.5,'MaxTemp']=43.5
df.loc[df['MaxTemp']<=2.7,'MaxTemp']=2.7
IQR=df.Rainfall.quantile(0.75)-df.Rainfall.quantile(0.25)
lower_bridge=df.Rainfall.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Rainfall.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
df.loc[df['Rainfall']>=1.5,'Rainfall']=1.5
df.loc[df['Rainfall']<=-0.89,'Rainfall']=-0.89
IQR=df.Evaporation.quantile(0.75)-df.Evaporation.quantile(0.25)
```

```
lower_bridge=df.Evaporation.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Evaporation.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
df.loc[df['Evaporation']>=14.6,'Evaporation']=14.6
df.loc[df['Evaporation']<=-4.6,'Evaporation']=-4.6
IQR=df.WindGustSpeed.quantile(0.75)-df.WindGustSpeed.quantile(0.25)
lower_bridge=df.WindGustSpeed.quantile(0.25)-(IQR*1.5)
upper_bridge=df.WindGustSpeed.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
df.loc[df['WindGustSpeed']>=68.5,'WindGustSpeed']=68.5
df.loc[df['WindGustSpeed']<=8.5,'WindGustSpeed']=8.5
IQR=df.WindSpeed9am.quantile(0.75)-df.WindSpeed9am.quantile(0.25)
lower_bridge=df.WindSpeed9am.quantile(0.25)-(IQR*1.5)
upper_bridge=df.WindSpeed9am.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
df.loc[df['WindSpeed9am']>=37,'WindSpeed9am']=37
df.loc[df['WindSpeed9am']<=-11,'WindSpeed9am']=-11
IQR=df.WindSpeed3pm.quantile(0.75)-df.WindSpeed3pm.quantile(0.25)
lower_bridge=df.WindSpeed3pm.quantile(0.25)-(IQR*1.5)
upper_bridge=df.WindSpeed3pm.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
df.loc[df['WindSpeed3pm']>40.5,'WindSpeed3pm']=40.5
df.loc[df['WindSpeed3pm']<=-3.5,'WindSpeed3pm']=-3.5
IQR=df.Humidity9am.quantile(0.75)-df.Humidity9am.quantile(0.25)
lower_bridge=df.Humidity9am.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Humidity9am.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
df.loc[df['Humidity9am']>=122,'Humidity9am']=122
df.loc[df['Humidity9am']<=18,'Humidity9am']=18
IQR=df.Pressure9am.quantile(0.75)-df.Pressure9am.quantile(0.25)
lower_bridge=df.Pressure9am.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Pressure9am.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
df.loc[df['Pressure9am']>=1034.25,'Pressure9am']=1034.25
df.loc[df['Pressure9am']<=1001.05,'Pressure9am']=1001.05
IQR=df.Pressure3pm.quantile(0.75)-df.Pressure3pm.quantile(0.25)
lower_bridge=df.Pressure3pm.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Pressure3pm.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
df.loc[df['Pressure3pm']>=1031.85,'Pressure3pm']=1031.85
df.loc[df['Pressure3pm']<=998.65,'Pressure3pm']=998.65
IQR=df.Temp9am.quantile(0.75)-df.Temp9am.quantile(0.25)
lower_bridge=df.Temp9am.quantile(0.25)-(IQR*1.5)
```

```
upper_bridge=df.Temp9am.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
df.loc[df['Temp9am']>=35.3,'Temp9am']=35.3
df.loc[df['Temp9am']<=-1.49,'Temp9am']=-1.49
IQR=df.Temp3pm.quantile(0.75)-df.Temp3pm.quantile(0.25)
lower_bridge=df.Temp3pm.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Temp3pm.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
df.loc[df['Temp3pm']>=40.45,'Temp3pm']=40.45
df.loc[df['Temp3pm']<=2.45,'Temp3pm']=2.45
for feature in continuous_feature:
   data=df.copy()
   sns.boxplot(data[feature])
   plt.title(feature)
   plt.figure(figsize=(15,15))
def qq_plots(df, variable):
   plt.figure(figsize=(15,6))
   plt.subplot(1, 2, 1)
   df[variable].hist()
   plt.subplot(1, 2, 2)
   stats.probplot(df[variable], dist="norm", plot=plt)
   plt.show()
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()
df.to_csv("preprocessed_1.csv", index=False)
X = df.drop(["RainTomorrow", "Date"], axis=1)
Y = df["RainTomorrow"]
# scaler = RobustScaler()
# X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X,Y, test_size =0.2, stratify = Y, random_state = 0)
y_train
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)))
cat = CatBoostClassifier(iterations=2000, eval_metric = "AUC")
cat.fit(X_train_res, y_train_res)
```

```
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))
metrics.plot_roc_curve(cat, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred, average=None)
rf=RandomForestClassifier()
rf.fit(X_train_res,y_train_res)
y_pred1 = rf.predict(X_test)
print(confusion_matrix(y_test,y_pred1))
print(accuracy_score(y_test,y_pred1))
print(classification_report(y_test,y_pred1))
metrics.plot_roc_curve(rf, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred1, average=None)
logreg = LogisticRegression()
logreg.fit(X_train_res, y_train_res)
y_pred2 = logreg.predict(X_test)
print(confusion_matrix(y_test,y_pred2))
print(accuracy_score(y_test,y_pred2))
print(classification_report(y_test,y_pred2))
metrics.plot_roc_curve(logreg, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred2, average=None)
gnb = GaussianNB()
gnb.fit(X_train_res, y_train_res)
y_pred3 = gnb.predict(X_test)
print(confusion_matrix(y_test,y_pred3))
print(accuracy_score(y_test,y_pred3))
print(classification_report(y_test,y_pred3))
metrics.plot_roc_curve(gnb, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred3, average=None)
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train_res, y_train_res)
y_pred4 = knn.predict(X_test)
print(confusion_matrix(y_test,y_pred4))
print(accuracy_score(y_test,y_pred4))
print(classification_report(y_test,y_pred4))
metrics.plot_roc_curve(knn, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred4, average=None)
xgb = XGBClassifier()
xgb.fit(X_train_res, y_train_res)
y_pred6 = xgb.predict(X_test)
print(confusion_matrix(y_test,y_pred6))
print(accuracy_score(y_test,y_pred6))
```

```
print(classification_report(y_test,y_pred6))
metrics.plot_roc_curve(xgb, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred6, average=None)
svc = SVC()
svc.fit(X_train_res, y_train_res)
y_pred5 = svc.predict(X_test)
print(confusion_matrix(y_test,y_pred5))
print(accuracy_score(y_test,y_pred5))
print(classification_report(y_test,y_pred5))
metrics.plot_roc_curve(svc, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred5, average=None)
# joblib.dump(rf, "rf.pkl")
# joblib.dump(cat, "cat.pkl")
# joblib.dump(logreg, "logreg.pkl")
# joblib.dump(gnb, "gnb.pkl")
# joblib.dump(knn, "knn.pkl")
joblib.dump(svc, "svc.pkl")
joblib.dump(xgb, "xgb.pkl")
```

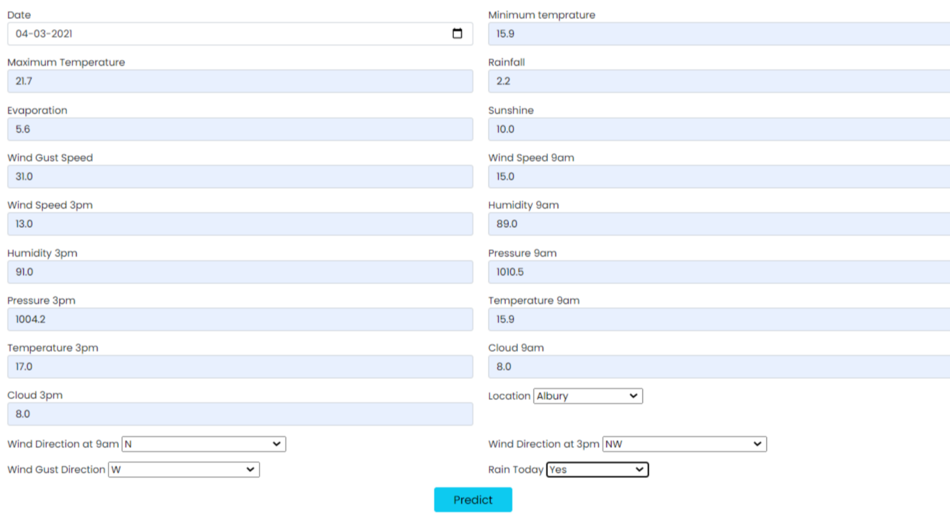
8. TESTING

8.1 Test Cases

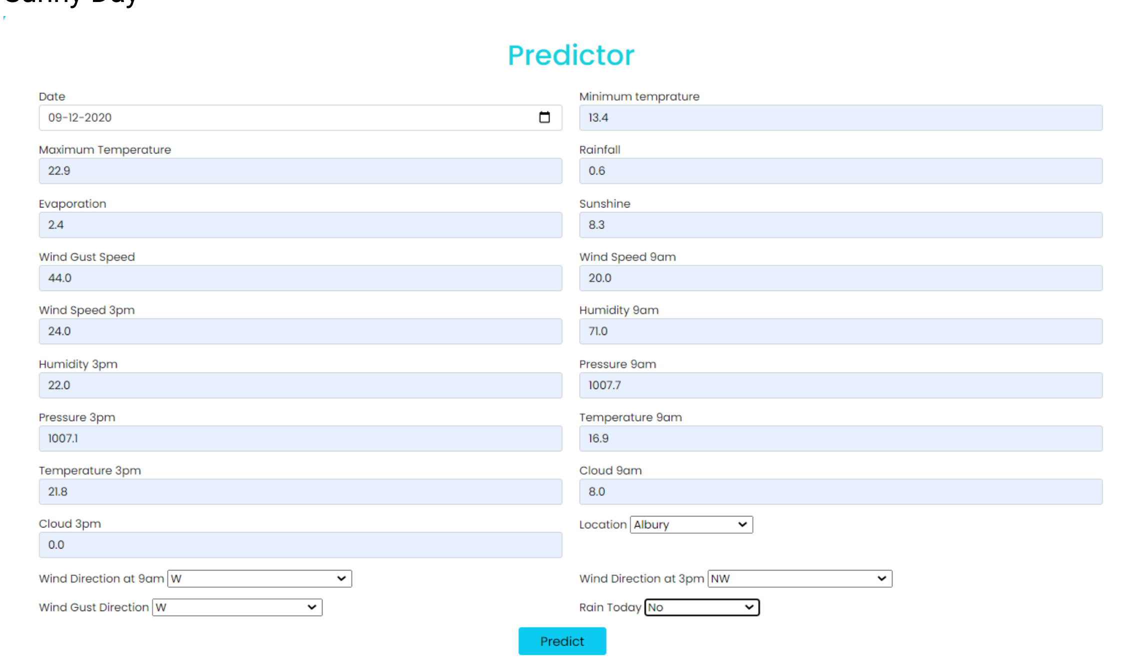
The correlation of wind speed between humidity and temperature was -0.256 and 0.278, which is low correlate. That must be other active factors that influence the wind speed and contribute to the windstorm event. Wind speed, humidity and temperature during the windstorm event on 11 February 2017 was analyzed. During the windstorm event, the wind speed blows up to 15.7 m/s while the humidity reading decrease to 68.4 % and the temperature was 30.9 °C.When the wind speed reading is high, the temperature reading also increases and the humidity reading will go down and vice versa and has caused the windstorm event to happen.

8.2 User Acceptance Testing

Rainy Day:



Sunny Day



9. RESULTS

9.1 Performance Metrics

	precision	recall	f1-score	support
0	0.89	0.95	0.92	22717
1	0.75	0.56	0.64	6375
accuracy			0.86	29092
macro avg	0.82	0.76	0.78	29092
weighted avg	0.86	0.86	0.86	29092

10. ADVANTAGES & DISADVANTAGES

- The weather application enables you make better preparation for the day in relation to giving accurate daily weather forecast.
- Measure for performance of models is taken by three criteria namely frequency of success, mean relative error, and range of relative error.
- This project was decided to be done in three months and was thought to be feasible one.
- To collect yearly and monthly rainfall dataset is difficult.

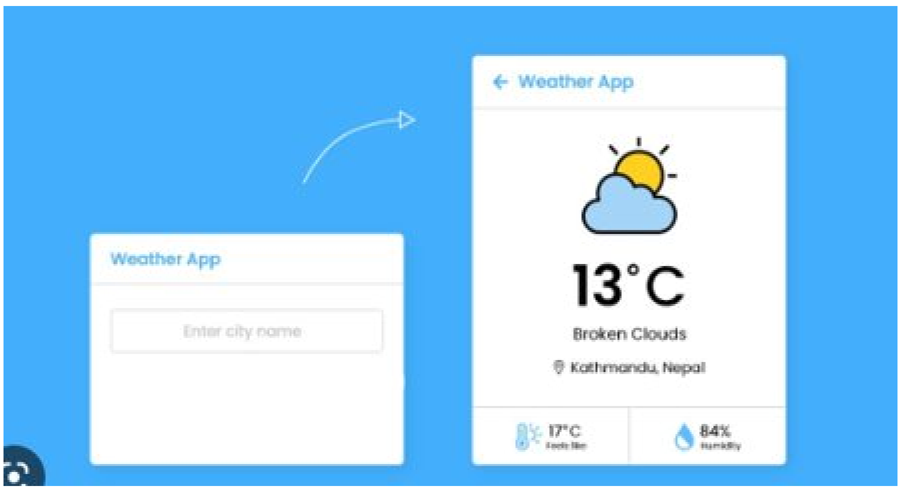
11. CONCLUSION

The results show that India has two main rainfall season: one is southwest monsoon (advancing monsoon) and other is Northeast monsoon(retreating monsoon). Advancing monsoon contributes almost 80% of the rainfall. Southwest and Northeast part of India receives most of the rainfall during the advancing monsoon. During the retreating monsoon, Andaman & Nicobar Islands, Kerala, Tamil Nadu receive more rainfall as compared to other subdivisions. The trend analysis of Annual rainfall considering India as whole show decreasing trend however when trend is analysed for all subdivision individually can see some division showing increasing trend and some showing decreasing trend. It showed that is is import to study subdivision for better forecasting. It considered Tamil Nadu as one of the subdivisions to do further analysis. It receives more rainfall during October and November because of retreating monsoon. Since there are only a few months when the Tamil Nadu gets rains and its location at tropical results in high temperature which in turn results in water scarcity problem. Also because of its geographic location near it is hit sometimes by the cyclones formed in Indian Oceans which results in extreme storms and non normal rainfall.

In an interview, Mrutyunjay Mohapatra, the director general of the IMD, explained how climate change is increasing number of days with heavy rainfall. The season started with 33% deficit rainfall but is ending with 10% higher than normal rainfall, with heavy spells of rain resulting in devastating floods in many states. It said that the number of heavy rainfall days was increasing because of climate change, which was making predictions more difficult. Also, this year, the monsoon in India withdrew 40 days later than normal.

12. FUTURE SCOPE

To find current weather ,we going to use API request call to get the weather of the city entered by the client



13. APPENDIX

1. Amanatidis, G.T., A.G. Paliatsos, C.C. Repapis and J.G. Bartzis, 1993. "Decreasing precipitation trend in the Marathon area", Greece. Int. J. Climatol., pp:191-201.
DOI: 10.1002/joc.3370130205

2. Aronica, G., M. Cannarozzo and L. Noto, 2002. "Investigating the changes in extreme rainfall series recorded in an urbanised area". Water Sci. Technol., pp: 49-54.
PMID: 11888183

3. Aung, T., A. Singh and U. Prasad, 2009. "Sea levelthreat in Tuvalu".
Am. J. Applied Sci., pp: 1169-1174.
DOI: 10.3844/ajassp.2009.1169.1174

4. Behringer, J., R. Buerki and J. Fuhrer, 2000. "Participatory integrated assessment of

adaptation to climate change in Alpine tourism and mountain agriculture”. Integrated Assess., pp: 331-338.
DOI: 10.1023/A:1018940901744

5. Bonaccorso, B., A. Cancelliere and G. Rossi, 2005. “Detecting trends of extreme rainfall series in Sicily”. Advan. Geosci., pp: 7-11.
DOI: 10.5194/ADGEO-2-7-2005