EXPLORATORY ANALYSIS OF RAINFALL DATA IN INDIA FOR AGRICULTURE

A mini project report

TEAM ID :PNT2022TMID08185

Submitted by

1.SONIYA A [AC19UEC121]
2.SANJUKTHAA B [AC19UEC110]
3.SHIVA RAMYA M [AC19UEC114]
4.SUJEETHA BAI K [AC19UEC127]

in partial fulfilment for the award of the degree
of
BACHELOR OF ENGINEERING
in
ELECTRONICS AND COMMUNICATION ENGINEERING

ADHIYAMAAN COLLEGE OF ENGINEERING[AUTONOMOUS]
Dr.M.G.R NAGAR,HOSUR - 635130,TAMIL NADU

ANNA UNIVERSITY::CHENNAI 600025

NOVEMBER 2022

CONTENT

1)INTRODUCTION

Rainfall prediction is important as heavy rainfall can lead to many disasters. The prediction helps people to take preventive measures and moreover the prediction should be accurate.There are two types of prediction: short term rainfall prediction and long term rainfall. Prediction, mostly short term prediction, can give us accurate results. The main challenge is to build a model for long term rainfall prediction.Heavy precipitation prediction could be a major drawback for the earth science department because it is closely associated with the economy and lifetime of humans. It’s a cause for natural disasters like flood and drought that is a square measure encountered by individuals across the world each year. Accuracy of rainfall statements has nice importance for countries like India whose economy is basically dependent on agriculture. The dynamic nature of the atmosphere, applied mathematics techniques fail to provide sensible accuracy for precipitation statements

1.1PROJECT OVERVIEW:

This starter kit is a simple application to demonstrate the Speech to Text service. After creating the app, use the getting started material to learn about how to use the service. Included is a zile you can download with everything you need to easily deploy the project locally and to the IBM Cloud.

1.2 PURPOSE

India is an agricultural country and the secondary agro based market will be steady with a good monsoon. The economic growth of each year depends on the amount of duration of monsoon rain, bad monsoon can lead to destruction of some crops, which may result in scarcity of some agricultural products which in turn can cause food inflation, insecurity and public unrest. In our analysis we are trying to understand the behaviour of rainfall in India over the years, by months and different subdivisions.

Flask is a web framework. The use of flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

The flask application which helps the farmers to protect their crops from rainfall and this app predicts whether it rains or not in all over India.

2)LITERATURE SURVEY

Agriculture in india "gamble with monsoon", short term growth rate is linked to annual rainfall, good monsoon ensure high agricultural growth rate during end of the year.In india several rainfall zones(36 rainfall zones),the mean annual rainfall varies as low as 200mm to as high as 11,000mm and large downward deviation in one low rainfall region(north western part of India) and small upward deviation of high rainfall region (north eastern part of India). The average rainfall in country level and divisional level are linked between two variables i.e rainfall and annual agricultural output growth. The high percentage increase of rainfall in previous year results in high agricultural growth and high percentage decline of rainfall results low growth and often resulting in negative growth.Rainfall analysis suggests that annual agricultural growth performance of country depends on two factors:

1) percentage difference in average rainfall of previous year and
2) percentage increase in gross irrigated area of previous year.
3) to access the impact of climate change on rainfall pattern across India

In India,the decreasing frequency of light to moderate rain events have more than offset the increasing frequency of heavy and very heavy rain events to reduce the overall summer monsoonal rainfall in Central India.A spatial and temporal incoherent feature of rainfall necessitates the need to examine its changing pattern because rainfall is one of the most important parameters that influence the agriculture of a region for food production,thus affecting the socio-economic status of resource poor marginal and small farmers.Hence,the following specific objectives

1)to examine the long term trend of temperature across India
2)to identify the year of climate change across India

Mountains are the source of high quality water.They can be said as sentinels to climate change" as they show more dynamic changes than in plains.In four zones of western Ghats there is a variation in rainfall in different seasons and zones,that is the distribution pattern is different.As the regional heterogeneity in Indian monsoon has become more prominent in recent years,the rainfall variation study needs to be carried out in regional scale.The heavy rainfall spells are mostly common during monsoon season in SW peninsular region.The spatial autocorrelation of mean rainfall is examined by univariate Local Moran's (LISA) index at 5% significance.The rainfall variation in study case is clustered into regions ,indicates its association with regional parameters.

2.1.EXISTING PROBLEMS

1)Indian agriculture depends heavily on rainfall. It not only influences agricultural production but also affects the prices of all agricultural commodities.

2)Rainfall is an exogenous variable which is beyond farmers' control. The outcome of rainfall fluctuation is quite natural. It has been observed that fluctuation in rainfall brings about fluctuation in output leading to price changes.

3)Considering the importance of rainfall in determining agricultural production and prices, the study has attempted to forecast monthly rainfall in India with the help of time series analysis using monthly rainfall data. Both linear and non-linear models have been used. The value of diagnostic checking parameters (MAE, MSE, RMSE) is lower in a non-linear model compared to a linear one.

4)The non-linear model - Artificial Neural Network (ANN) has been chosen instead of linear models, namely, simple seasonal exponential smoothing and Seasonal Auto-Regressive Integrated Moving Average to forecast rainfall. This will help to identify the proper cropping pattern.

Demerits

1)Underwatering "starves" the plant of water, which can lead to crop death or low yield.

2)Recent extreme weather events are destroying standing crops as well as causing other impacts such as climate-triggered pestilence.

3)If crops are too wet, they could also start to mould or catch a fungus.

4)The soil can also start to collect bacteria, mould and fungus which can then be absorbed by the plant

2.2.REFERENCES

1) Analysis of historical changes in rainfall in the Indian Himalayas,Ashok Basistha,D.S.Arya, N.k.Goel ,2002.

2) Trends in the rainfall pattern over India, P.Guhathakurta, M.Rajeevan , 2008.

3) Trend and spectral analysis of rainfall over India during 1901–2000 ,Manish .K.joshi , A.C.pandey, 2011.

4) Trends in the annual extreme rainfall events of 1 to 3 days duration over India, P.Rakhecha , M.soman.

5) Impact of climate change on extreme rainfall events and flood risk in India, P.Guhathakurta , O.sreejith ,P.Menon.

6)Dholakia, R. H. and A. A. Sapre (2013), Inter-sectoral Terms of Trade and Aggregate
Supply Response in Gujarat and Indian Agriculture, Working Paper No. 2013-07-02, Indian Institute of Management, Ahmedabad, India.

7)Jagadeesan, S. and M. D. Kumar (2015), The Sardar Sarovar Project: Assessing Economic And Social Impacts, SAGE Publishing India, New Delhi.

8)Kannan, E. and S. Sundaram (2011), Analysis of Trends in India's Agricultural Growth,
Working Paper # 276, Institute of Social and Economic Change, Bangalore.

9)Kumar, M. Dinesh, A. Narayanamoorthy, O. P. Singh, M. V. K. Sivamohan, M. Sharma
and Nitin Bassi (2010), Gujarat's Agricultural Growth Story: Exploding Some Myths, Occasional Paper # 2, Institute for Resource Analysis and Policy, Hyderabad, India.

10)Kumar, M.D. and C. J. Perry (2019), "What can Explain Groundwater Rejuvenation in
Gujarat in Recent Years?", International Journal of Water Resources Development, Vol. 35, No. 5, pp.891-906.

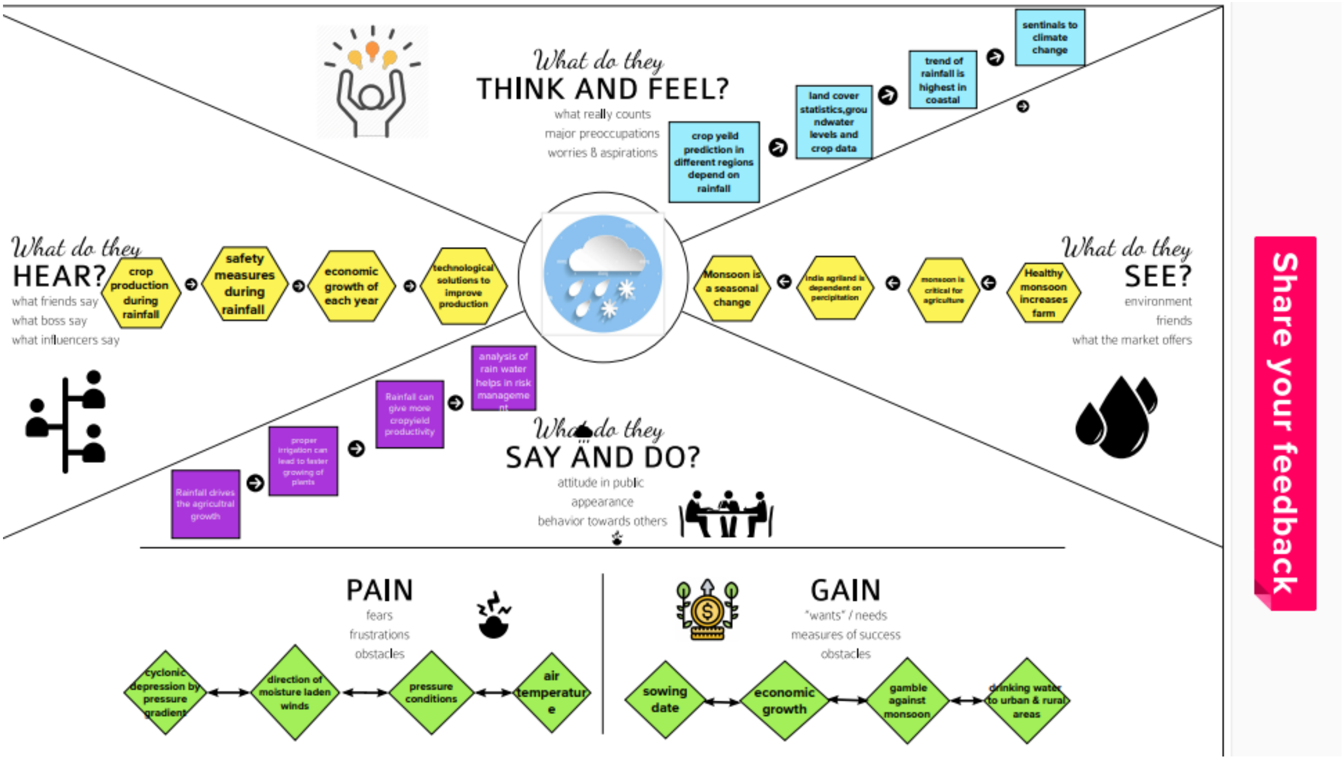
11)Shah, T., A. Gulati, Hemant P., G. Shreedhar and R. C. Jain (2009), "Secret of Gujarat's Agrarian Miracle after 2000", Economic and Political Weekly, Vol. 44, No. 52, pp.45-55.

2.3 PROBLEM STATEMENT DEFINITION

How can we predict the rainfall and the weather? Which method is best for crop yield?.

3)IDEAS OF PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



What do they THINK AND FEEL?

what really counts
major preoccupations
worries & aspirations

crop yeild prediction in different regions depend on rainfall → land cover statistics,groundwater levels and crop data → trend of rainfall is highest in coastal → sentinals to climate change

What do they HEAR?

what friends say
what boss say
what influencers say

crop production during rainfall → safety measures during rainfall → economic growth of each year → technological solutions to improve production

What do they SEE?

environment
friends
what the market offers

Healthy monsoon increases farm → monsoon is critical for agriculture → india agriland is dependent on percipitation → Monsoon is a seasonal change

What do they SAY AND DO?

attitude in public
appearance
behavior towards others

Rainfall drives the agricultral growth → proper irrigation can lead to faster growing of plants → Rainfall can give more cropyield productivity → analysis of rain water helps in risk management

PAIN

fears
frustrations
obstacles

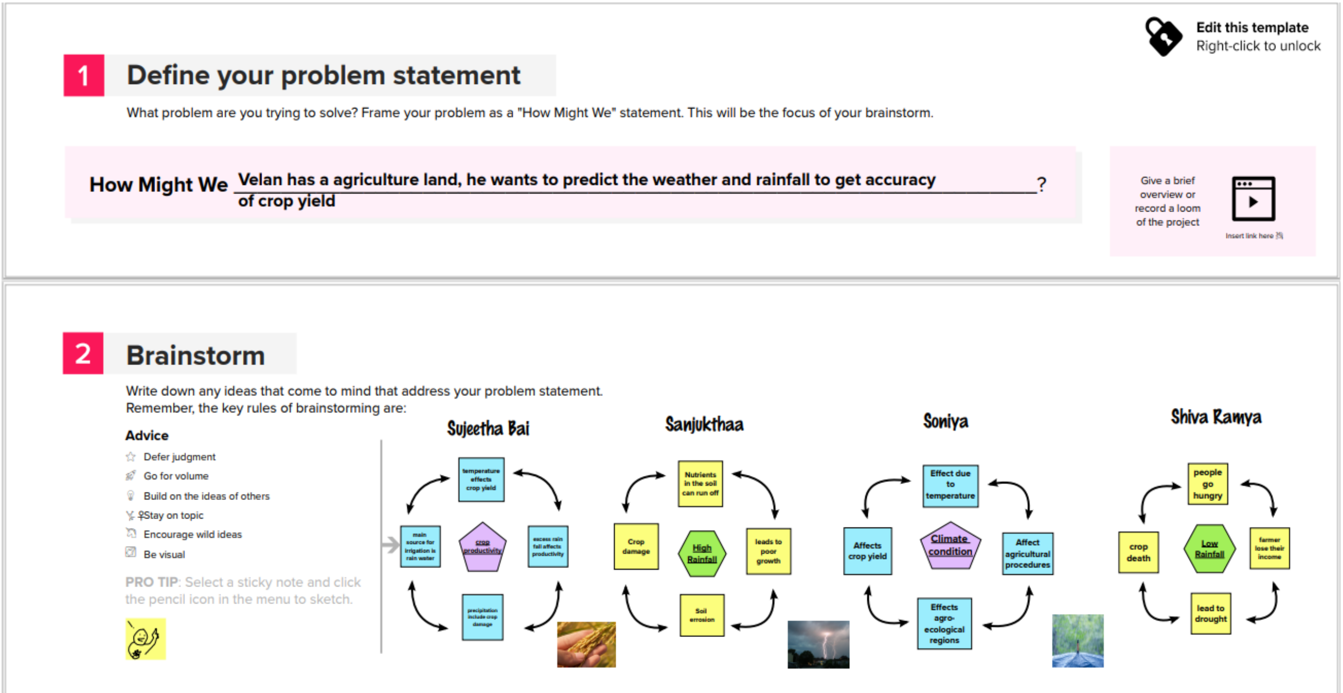
cyclonic depression by pressure gradient ↔ direction of moisture laden winds ↔ pressure conditions ↔ air temperature

GAIN

"wants" / needs
measures of success
obstacles

sowing date ↔ economic growth ↔ gamble against monsoon ↔ drinking water to urban & rural areas

3.2 IDEATION AND BRAINSTORMING



Edit this template
Right-click to unlock

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a "How Might We" statement. This will be the focus of your brainstorm.

How Might We Velan has a agriculture land, he wants to predict the weather and rainfall to get accuracy of crop yield ?

Give a brief overview or record a loom of the project

Insert link here

2 Brainstorm

Write down any ideas that come to mind that address your problem statement.
Remember, the key rules of brainstorming are:

Advice

- Defer judgment
- Go for volume
- Build on the ideas of others
- Stay on topic
- Encourage wild ideas
- Be visual

PRO TIP: Select a sticky note and click the pencil icon in the menu to sketch.

Sujeetha Bai

crop productivity: temperature effects crop yield; excess rain fall affects productivity; precipitation include crop damage; main source for irrigation is rain water

Sanjukthaa

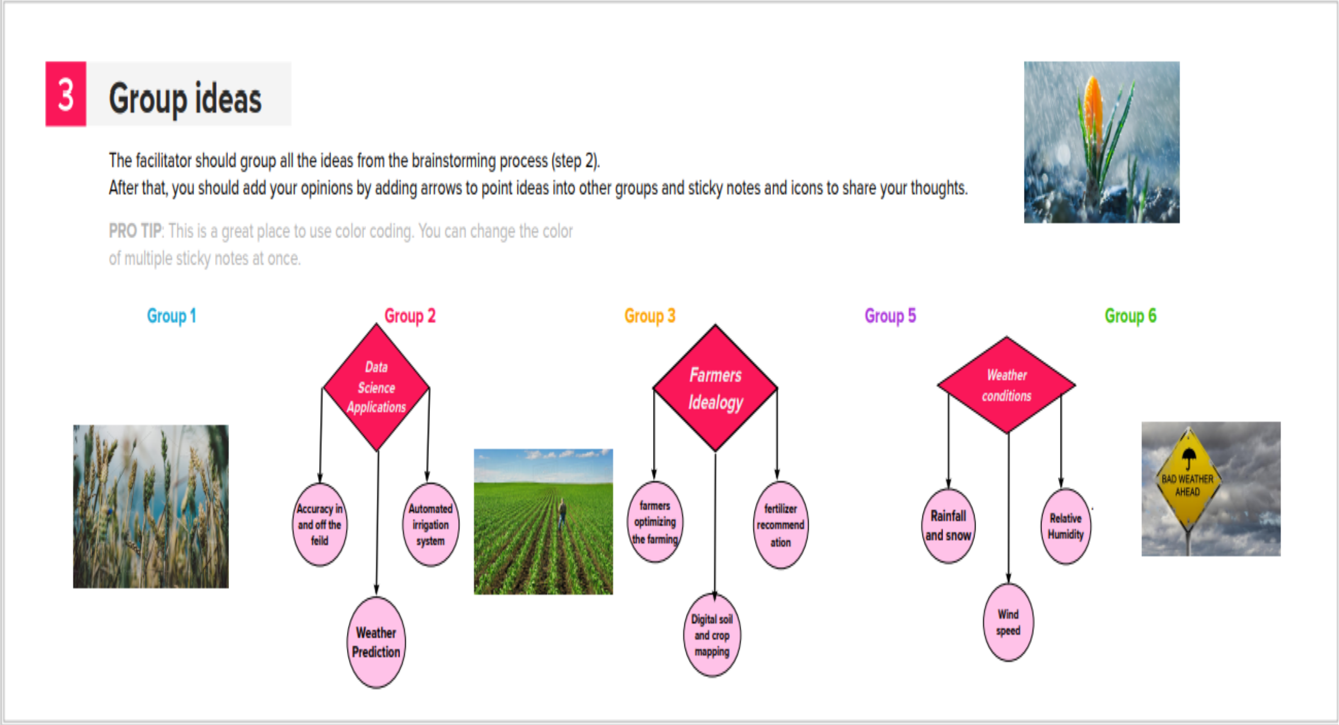
High Rainfall: Nutrients in the soil can run off; leads to poor growth; Soil errosion; Crop damage

Soniya

Climate condition: Effect due to temperature; Affect agricultural procedures; Effects agro-ecological regions; Affects crop yield

Shiva Ramya

Low Rainfall: people go hungry; farmer lose their income; lead to drought; crop death



3 Group ideas

The facilitator should group all the ideas from the brainstorming process (step 2).
After that, you should add your opinions by adding arrows to point ideas into other groups and sticky notes and icons to share your thoughts.

PRO TIP: This is a great place to use color coding. You can change the color of multiple sticky notes at once.

Group 1
Group 2
Data Science Applications
Accuracy in and off the feild
Automated irrigation system
Weather Prediction
Group 3
Farmers Idealogy
farmers optimizing the farming
fertilizer recommend ation
Digital soil and crop mapping
Group 5
Weather conditions
Rainfall and snow
Relative Humidity
Wind speed
Group 6
BAD WEATHER AHEAD

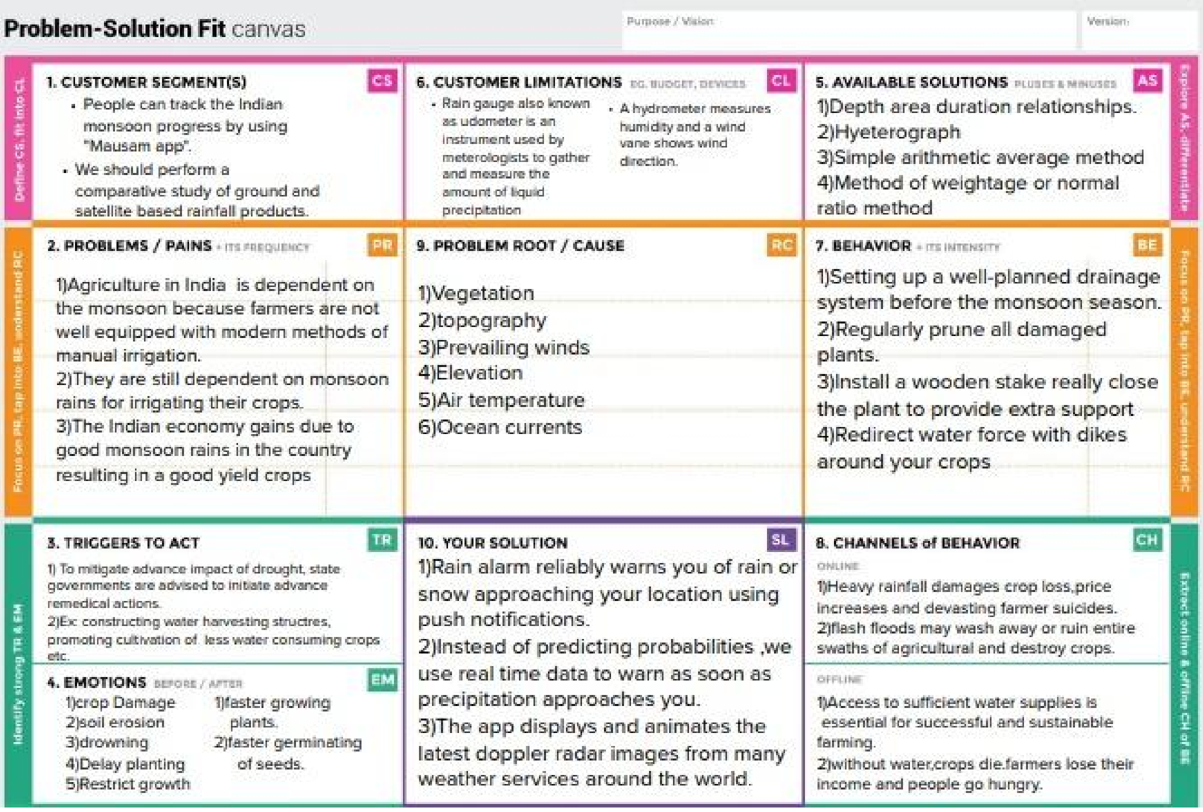
3.3.PROPOSED SOLUTION

The project aims at building an application that provides information about the Crop production during rainfall and to check the weather condition in INDIA to help the farmers to safeguard their crops during rainfall.

1.Problem statement(problem to be solved)	How can we predict the Rainfall and the weather? Which method is best for crop yield? Ensures as a life saving device for crop production which helps the farmers to protect crops from rainfall and checks the weather conditions to predict crops
2.Idea / Solution description	Ensures as a life saving device for crop production which helps the farmers to protect crops from rainfall and checks the weather conditions to predict crops.
3.Novelty /Uniqueness	1)Safeguard the crops from rainfall 2)Analyse the weather condition

	3)Measuring the rainfall data
4.Social Impact / Customer Satisfaction	It will help the farmers to take precautionary steps to minimize the losses and consider technological solutions to improve production
5.Business Model (financial Benefit)	1)Collaboration in agriculture-sector 2)Providing technological solutions 3)Offer exciting on crops in agriculture
6.Scalability of Solution	1)Analyse the climate data from meteorological stations and from food and Agricultural organisations. 2)Estimating climate change impacts on Indian agriculture

3.4 **PROPOSED SOLUTION FIT**



Purpose / Vision
Version:
Define CS, fit into CL
Focus on PR, tap into BE, understand RC
Identify strong TR & EM
Explore AS, differentiate
Focus on PR, tap into BE, understand RC
Extract online & offline CH of BE
CS
CL
AS
PR
RC
BE
TR
EM
SL
CH

Problem-Solution Fit canvas

1. CUSTOMER SEGMENT(S)

- People can track the Indian monsoon progress by using "Mausam app".
- We should perform a comparative study of ground and satellite based rainfall products.

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES

- Rain gauge also known as udometer is an instrument used by meterologists to gather and measure the amount of liquid precipitation
- A hydrometer measures humidity and a wind vane shows wind direction.

5. AVAILABLE SOLUTIONS PLUSES & MINUSES

1)Depth area duration relationships.
2)Hyeterograph
3)Simple arithmetic average method
4)Method of weightage or normal ratio method

2. PROBLEMS / PAINS + ITS FREQUENCY

1)Agriculture in India is dependent on the monsoon because farmers are not well equipped with modern methods of manual irrigation.
2)They are still dependent on monsoon rains for irrigating their crops.
3)The Indian economy gains due to good monsoon rains in the country resulting in a good yield crops

9. PROBLEM ROOT / CAUSE

1)Vegetation
2)topography
3)Prevailing winds
4)Elevation
5)Air temperature
6)Ocean currents

7. BEHAVIOR + ITS INTENSITY

1)Setting up a well-planned drainage system before the monsoon season.
2)Regularly prune all damaged plants.
3)Install a wooden stake really close the plant to provide extra support
4)Redirect water force with dikes around your crops

3. TRIGGERS TO ACT

1) To mitigate advance impact of drought, state governments are advised to initiate advance remedical actions.
2)Ex: constructing water harvesting structres, promoting cultivation of less water consuming crops etc.

4. EMOTIONS BEFORE / AFTER

1)crop Damage	1)faster growing plants.
2)soil erosion	2)faster germinating of seeds.
3)drowning	
4)Delay planting	
5)Restrict growth	

10. YOUR SOLUTION

1)Rain alarm reliably warns you of rain or snow approaching your location using push notifications.
2)Instead of predicting probabilities ,we use real time data to warn as soon as precipitation approaches you.
3)The app displays and animates the latest doppler radar images from many weather services around the world.

8. CHANNELS of BEHAVIOR

ONLINE

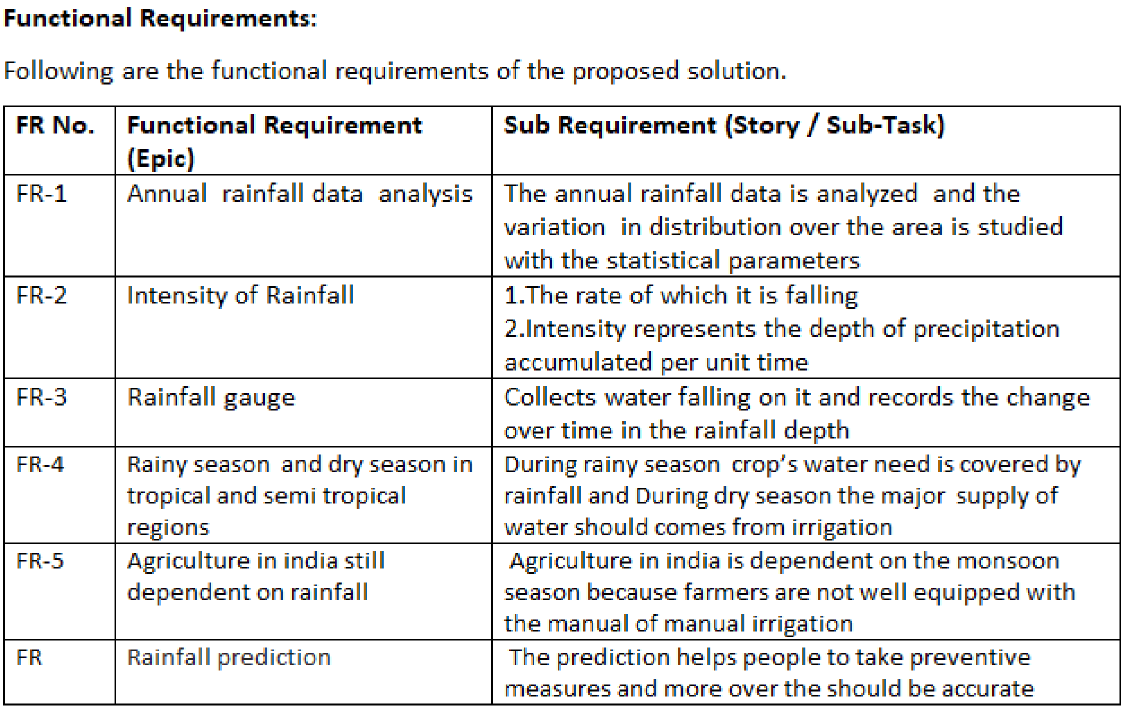
1)Heavy rainfall damages crop loss,price increases and devasting farmer suicides.
2)flash floods may wash away or ruin entire swaths of agricultural and destroy crops.

OFFLINE

1)Access to sufficient water supplies is essential for successful and sustainable farming.
2)without water,crops die.farmers lose their income and people go hungry.

4)REQUIREMENT ANALYSIS

4.1Functional requirement



Functional Requirements:

Following are the functional requirements of the proposed solution.

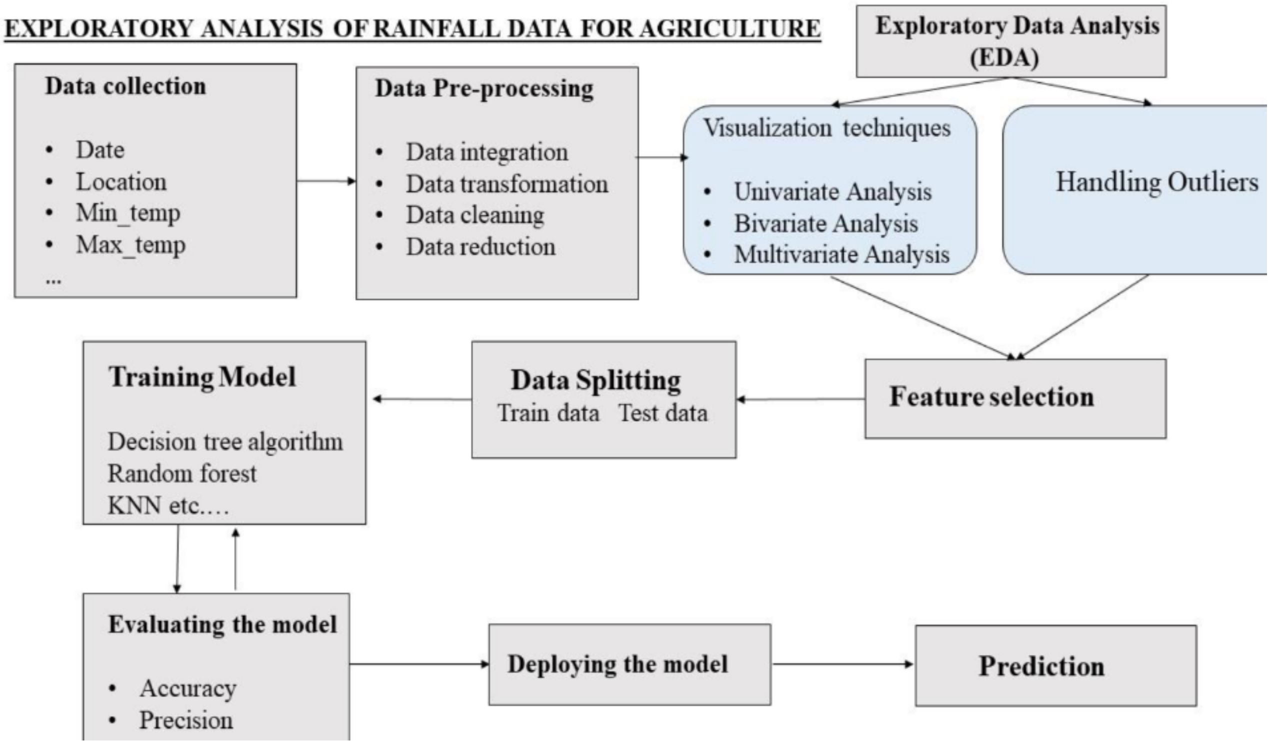
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Annual rainfall data analysis	The annual rainfall data is analyzed and the variation in distribution over the area is studied with the statistical parameters
FR-2	Intensity of Rainfall	1.The rate of which it is falling 2.Intensity represents the depth of precipitation accumulated per unit time
FR-3	Rainfall gauge	Collects water falling on it and records the change over time in the rainfall depth
FR-4	Rainy season and dry season in tropical and semi tropical regions	During rainy season crop's water need is covered by rainfall and During dry season the major supply of water should comes from irrigation
FR-5	Agriculture in india still dependent on rainfall	Agriculture in india is dependent on the monsoon season because farmers are not well equipped with the manual of manual irrigation
FR	Rainfall prediction	The prediction helps people to take preventive measures and more over the should be accurate

4.2Non-Functional requirements

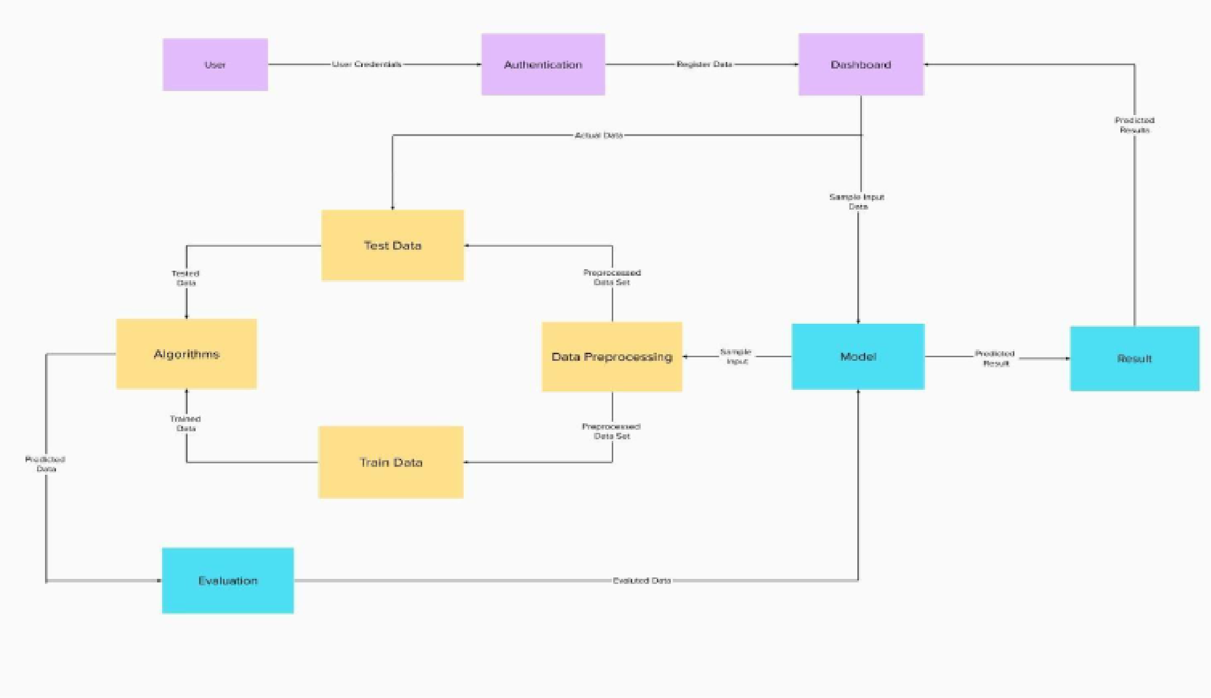
FR NO	Non-Functional Requirements	DESCRIPTION
NFR-1	usability	The system should be easy to install and simple to use
NFR-2	security	The system should authenticate users
NFR-3	Reliability	The system should perform the intended tasks for specific
NFR-4	performance	The system should perform the intended tasks for specific
NFR-5	Availability	The system should be available all the time when required
NFR-6	scalability	The system must be scalable enough to support 1,000,000 visits

5)PROJECT DESIGN

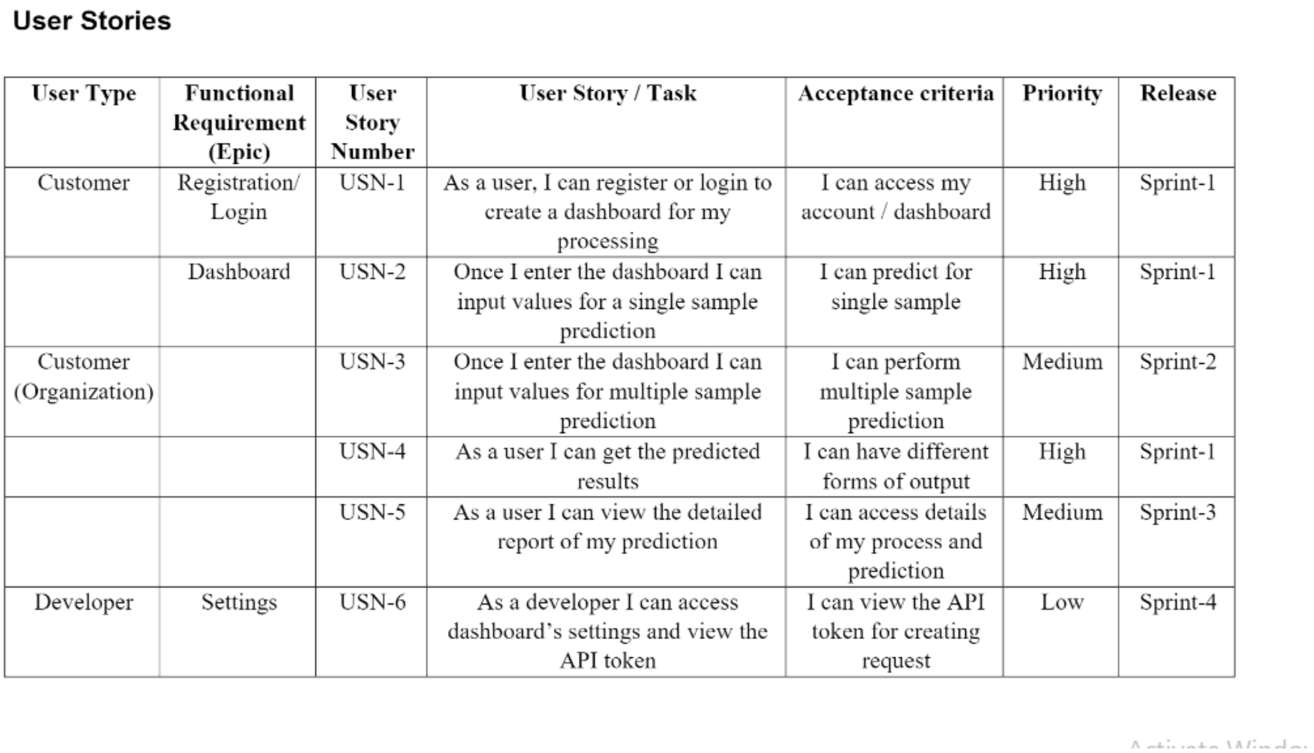
5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3 USER STORIES

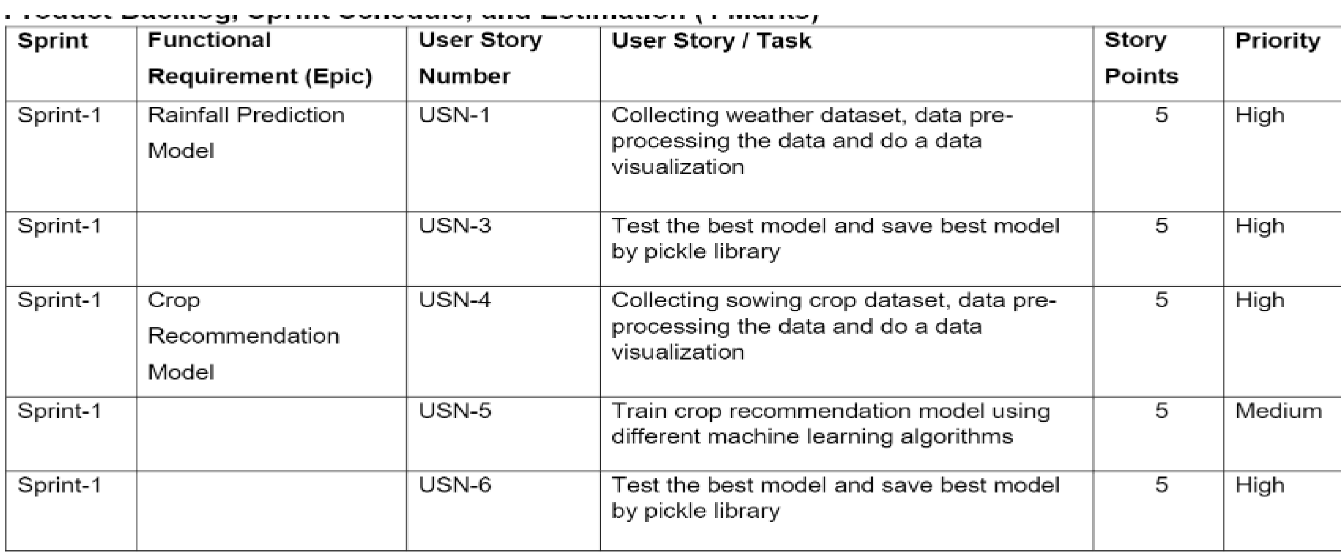


User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration/ Login	USN-1	As a user, I can register or login to create a dashboard for my processing	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-2	Once I enter the dashboard I can input values for a single sample prediction	I can predict for single sample	High	Sprint-1
Customer (Organization)		USN-3	Once I enter the dashboard I can input values for multiple sample prediction	I can perform multiple sample prediction	Medium	Sprint-2
		USN-4	As a user I can get the predicted results	I can have different forms of output	High	Sprint-1
		USN-5	As a user I can view the detailed report of my prediction	I can access details of my process and prediction	Medium	Sprint-3
Developer	Settings	USN-6	As a developer I can access dashboard's settings and view the API token	I can view the API token for creating request	Low	Sprint-4

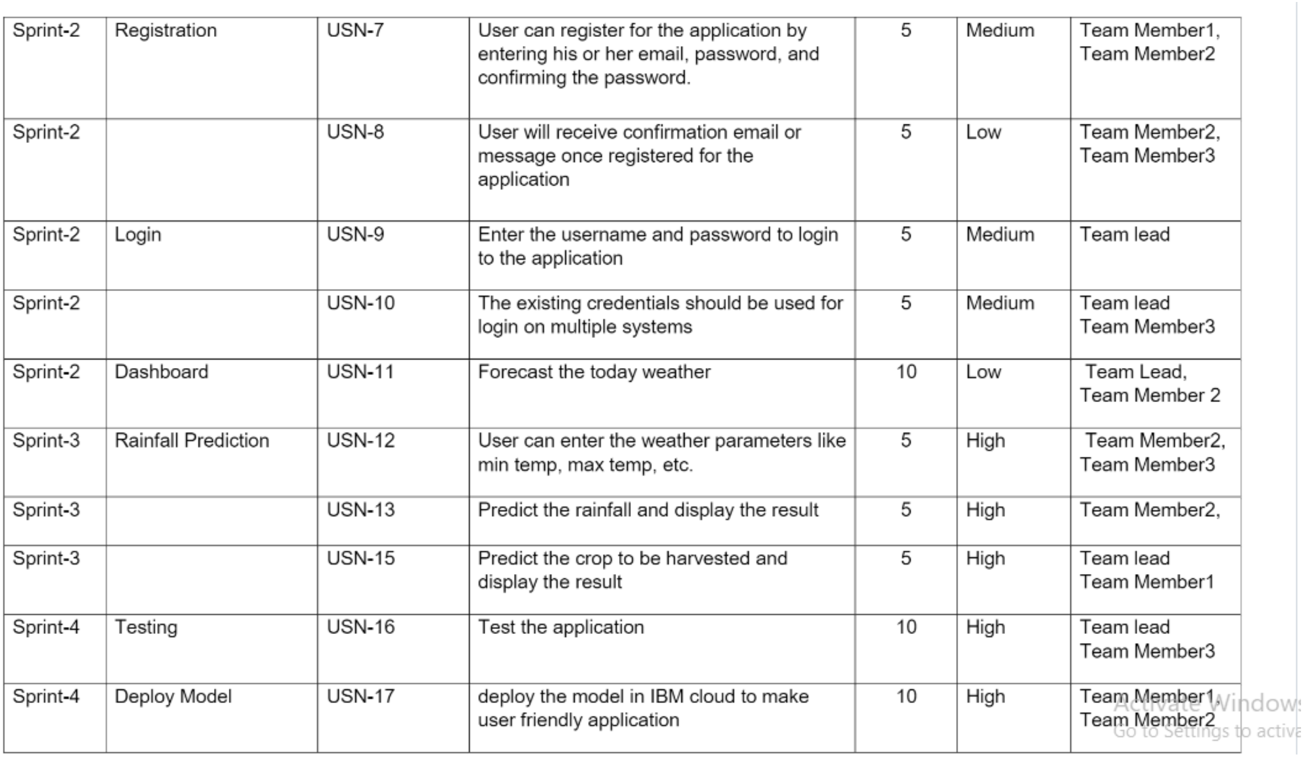
6)PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION



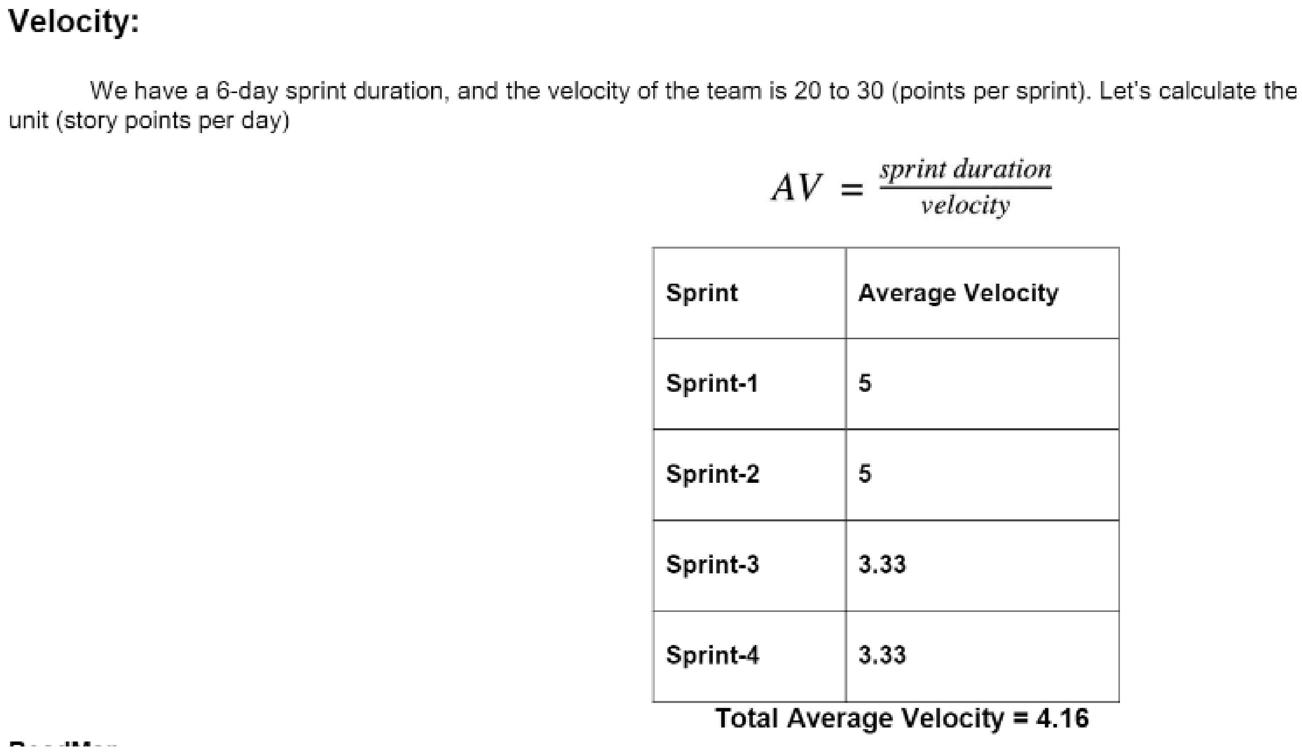
Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Rainfall Prediction Model	USN-1	Collecting weather dataset, data pre-processing the data and do a data visualization	5	High
Sprint-1		USN-3	Test the best model and save best model by pickle library	5	High
Sprint-1	Crop Recommendation Model	USN-4	Collecting sowing crop dataset, data pre-processing the data and do a data visualization	5	High
Sprint-1		USN-5	Train crop recommendation model using different machine learning algorithms	5	Medium
Sprint-1		USN-6	Test the best model and save best model by pickle library	5	High



Sprint-2	Registration	USN-7	User can register for the application by entering his or her email, password, and confirming the password.	5	Medium	Team Member1, Team Member2
Sprint-2		USN-8	User will receive confirmation email or message once registered for the application	5	Low	Team Member2, Team Member3
Sprint-2	Login	USN-9	Enter the username and password to login to the application	5	Medium	Team lead
Sprint-2		USN-10	The existing credentials should be used for login on multiple systems	5	Medium	Team lead Team Member3
Sprint-2	Dashboard	USN-11	Forecast the today weather	10	Low	Team Lead, Team Member 2
Sprint-3	Rainfall Prediction	USN-12	User can enter the weather parameters like min temp, max temp, etc.	5	High	Team Member2, Team Member3
Sprint-3		USN-13	Predict the rainfall and display the result	5	High	Team Member2,
Sprint-3		USN-15	Predict the crop to be harvested and display the result	5	High	Team lead Team Member1
Sprint-4	Testing	USN-16	Test the application	10	High	Team lead Team Member3
Sprint-4	Deploy Model	USN-17	deploy the model in IBM cloud to make user friendly application	10	High	Team Member1, Team Member2

6.2 SPRINT DELIVERY SCHEDULE



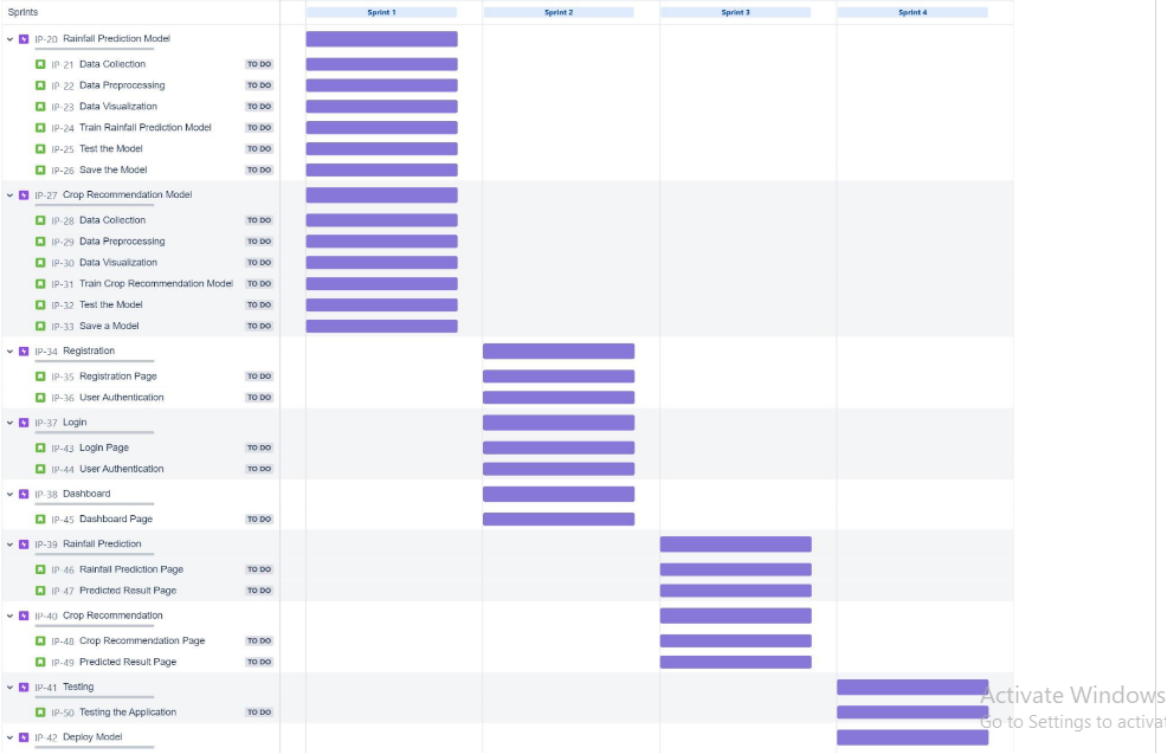
Velocity:

We have a 6-day sprint duration, and the velocity of the team is 20 to 30 (points per sprint). Let's calculate the unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity}$$

Sprint	**Average Velocity**
Sprint-1	**5**
Sprint-2	**5**
Sprint-3	**3.33**
Sprint-4	**3.33**

Total Average Velocity = 4.16



Sprints		Sprint 1	Sprint 2	Sprint 3	Sprint 4
IP-20 Rainfall Prediction Model		▬			
IP-21 Data Collection	TO DO	▬			
IP-22 Data Preprocessing	TO DO	▬			
IP-23 Data Visualization	TO DO	▬			
IP-24 Train Rainfall Prediction Model	TO DO	▬			
IP-25 Test the Model	TO DO	▬			
IP-26 Save the Model	TO DO	▬			
IP-27 Crop Recommendation Model		▬			
IP-28 Data Collection	TO DO	▬			
IP-29 Data Preprocessing	TO DO	▬			
IP-30 Data Visualization	TO DO	▬			
IP-31 Train Crop Recommendation Model	TO DO	▬			
IP-32 Test the Model	TO DO	▬			
IP-33 Save a Model	TO DO	▬			
IP-34 Registration			▬		
IP-35 Registration Page	TO DO		▬		
IP-36 User Authentication	TO DO		▬		
IP-37 Login			▬		
IP-43 Login Page	TO DO		▬		
IP-44 User Authentication	TO DO		▬		
IP-38 Dashboard			▬		
IP-45 Dashboard Page	TO DO		▬		
IP-39 Rainfall Prediction				▬	
IP-46 Rainfall Prediction Page	TO DO			▬	
IP-47 Predicted Result Page	TO DO			▬	
IP-40 Crop Recommendation				▬	
IP-48 Crop Recommendation Page	TO DO			▬	
IP-49 Predicted Result Page	TO DO			▬	
IP-41 Testing					▬
IP-50 Testing the Application	TO DO				▬
IP-42 Deploy Model					▬

6.3 REPORTS FROM JIRA

Contents

1. General 1

Published reports are the primary visible output of the Hydrological Information System. They have several purposes

v to provide information on availability of data for use in planning and design. Rainfall data are used for a variety of purposes and are required at a range of time scales. Real time rainfall data are required for flood forecasting and hydropower and reservoir operation. Summaries of storm rainfall event data are required for assessment of the severity of events at weekly or monthly time scales. Rainfall bulletins for agricultural and irrigation operation are needed at similar time scales. However, the HIS will data at yearly or longer reporting frequency and will not engage in shorter term operation reports. Although the same data may be used for such reports they will not be the direct concern of the HIS.

v to advertise the work of the HIS and its capability and to create interest and awareness amongst potential users.With the availability of data on magnetic media it is conceivable that all requests for data could be met by a direct and specific response to data requests.

2. Yearly reports 2

The annual report provides a summary of the rainfall pattern for the report year in terms of
distribution of rainfall in time and space and makes comparisons with long term statistics.
Details of the observational network and data availability are included. A summary of the the hydrological impact of rainfall is provided with particular reference to floods and droughts.
The following are typical contents of the annual report:
(a) Introduction
(b) The Observational Network
v maps v listings

(c) A descriptive account of rainfall occurrence during the report year
(d) Thematic maps of monthly, seasonal and annual rainfall
(e) Graphical and mapped comparisons with average patterns
(f) Basic rainfall statistics
(g) Description and statistical summaries of major storms
(h) Data validation and quality

3. Periodic reports - long term statistics 5

Long term point and areal statistics are important for planning, management and design of
water resources systems. They also play an important role in validation and analysis. These
statistics must be updated regularly and an interval of 10 years is recommended. The following will be typical contents of such reports.

- Introduction
- Data availability - maps and tabulations
- Descriptive account of annual rainfall since last report
- Thematic maps of mean monthly and seasonal rainfall
- Basic rainfall statistics - monthly and annual means, maxima and minima

v for the standard climatic normal period (1961-90) where available
v for the updated decade
v for the available period of record

- Additional point rainfall statistics for example, daily maximum rainfall, persistence of dry or wet spells during the monsoon, dates of onset or termination of the monsoon.
- Additional areal mean rainfall statistics for administrative or drainage areas for periods of a month or year

4. Periodic reports on unusual rainfall

Special reports should also be prepared on the occurrence of unusual rainfall events. As these will also have unusual hydrological consequences, the reports will normally be combined with reports of the resulting streamflow and flooding within the affected area. The rainfall component of such reports will include the following

- tabulations of hourly or daily point rainfall within the affected area
- isohyetal maps of total storm rainfall
- hyetograph plots of rainfall time distribution based on recording rain gauges
- assessment of event return periods for selected durations based on point rainfall
- Areal storm rainfall totals over affected basins.

7)CODING AND SOLUTIONING

7.1FEATURE-1

Import Packages and Connect to Drive

```
!python --version
 Python 3.9.13
```

```
 import os
 os.getcwd()
'/home/wsuser/work'
import os
import IPython.display as ipd
import matplotlib.pyplot as plt#graph plotting
import numpy as np#Array or Multi dimi. arrays
import pandas as pd#Reading files and sving it as tablur format
import time
from collections import Counter, OrderedDict#[1,2,3,3]
import cv2#opencv - python - Image processing- short form -cv2
#from google.colab.patches import cv2_imshow#Is used for printing the images
```

```
import random#Generating random variables or values
#import seaborn as sns#graph plotting
import json
import joblib
import pickle
from ibm_watson_machine_learning import APIClient
import sklearn
print('The scikit-learn version is {}.'.format(sklearn.__version__))
```

```
The scikit-learn version is 1.0.2.
```

```
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.utils import class_weight
from sklearn.utils.class_weight import compute_class_weight
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from sklearn.linear_model import Ridge, Lasso, ElasticNet
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.exceptions import NotFittedError
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import SGDClassifier
from sklearn import metrics
```

Data Loading and Preprocessing

```
import pandas as pd

from botocore.client import Config

import ibm_boto3
def __iter__(self): return 0
# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='geDl8EdyQxhVAqykYTv3jPIjXaicgiET0P6yZji-ubLg',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.eu.cloud-object-storage.appdomain.cloud')
bucket = 'rainfall-donotdelete-pr-eoii6hshulobg1'
object_key = 'rainfall.csv'
body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )
df = pd.read_csv(body)
Df
df = df.dropna()
df
for i in df.columns:
  print(i, df[i].dtype)
  print(df[i].value_counts())
print("################")
```

[ ]

```
cols = ['RainToday', 'RainTomorrow', 'WindDir3pm', 'WindDir9am', 'WindGustDir', 'Location']
```

[ ]

```
le = LabelEncoder()

for col in cols:

  print(col)

  df[col] = le.fit_transform(df[col])

  le_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))

  print(le_name_mapping)

  print("################")
```

```
RainToday
{'No': 0, 'Yes': 1}
################
RainTomorrow
{'No': 0, 'Yes': 1}
################
WindDir3pm
{'E': 0, 'ENE': 1, 'ESE': 2, 'N': 3, 'NE': 4, 'NNE': 5, 'NNW': 6, 'NW': 7, 'S': 8, 'SE': 9,
'SSE': 10, 'SSW': 11, 'SW': 12, 'W': 13, 'WNW': 14, 'WSW': 15}
################
WindDir9am
{'E': 0, 'ENE': 1, 'ESE': 2, 'N': 3, 'NE': 4, 'NNE': 5, 'NNW': 6, 'NW': 7, 'S': 8, 'SE': 9,
'SSE': 10, 'SSW': 11, 'SW': 12, 'W': 13, 'WNW': 14, 'WSW': 15}
################
WindGustDir
{'E': 0, 'ENE': 1, 'ESE': 2, 'N': 3, 'NE': 4, 'NNE': 5, 'NNW': 6, 'NW': 7, 'S': 8, 'SE': 9,
'SSE': 10, 'SSW': 11, 'SW': 12, 'W': 13, 'WNW': 14, 'WSW': 15}
```

```
#################
Location
{'AliceSprings': 0, 'Brisbane': 1, 'Cairns': 2, 'Canberra': 3, 'Cobar': 4, 'CoffsHarbour': 5, 'Darwin': 6, 'Hobart': 7, 'Melbourne': 8, 'MelbourneAirport': 9, 'Mildura': 10, 'Moree': 11, 'MountGambier': 12, 'NorfolkIsland': 13, 'Nuriootpa': 14, 'Perth': 15, 'PerthAirport': 16, 'Portland': 17, 'Sale': 18, 'Sydney': 19, 'SydneyAirport': 20, 'Townsville': 21, 'WaggaWagga': 22, 'Watsonia': 23, 'Williamtown': 24, 'Woomera': 25}
#################
```

[ ]

```
df
```

[ ]

```
for col in df.columns:
  print(col, df[col].dtype)
```

[ ]

```
df['Date'] = pd.to_datetime(df['Date'],
 format = '%Y-%m-%d',
 errors = 'coerce')
df
```

[ ]

```
for col in df.columns:
  print(col, df[col].dtype)
```

```
Date datetime64[ns]
Location int64
MinTemp float64
MaxTemp float64
Rainfall float64
Evaporation float64
```

```
Sunshine float64
WindGustDir int64
WindGustSpeed float64
WindDir9am int64
WindDir3pm int64
WindSpeed9am float64
WindSpeed3pm float64
Humidity9am float64
Humidity3pm float64
Pressure9am float64
Pressure3pm float64
Cloud9am float64
Cloud3pm float64
Temp9am float64
Temp3pm float64
RainToday int64
RISK_MM float64
RainTomorrow int64
```

[ ]

```
df['year'] = df['Date'].dt.year

df['month'] = df['Date'].dt.month

df['day'] = df['Date'].dt.day

df
```

[ ]

```
del df['Date']
```

[ ]

```
df
```

[ ]

```
for col in df.columns:
    print(col, df[col].dtype)
```

```
Location int64
MinTemp float64
MaxTemp float64
Rainfall float64
Evaporation float64
Sunshine float64
WindGustSpeed float64
WindDir9am int64
WindDir3pm int64
WindSpeed3pm float64
Humidity9am float64
Humidity3pm float64
Pressure9am float64
Pressure3pm float64
Cloud9am float64
Cloud3pm float64
Temp9am float64
Temp3pm float64
RainTomorrow int64
year int64
month int64
day int64
```

[ ]

```
X = df.drop('RainTomorrow', axis=1)

Y = df['RainTomorrow']
```

[ ]

```
X
```

[ ]

```
np.max(df['Temp3pm'].values), np.min(df['Temp3pm'].values)
```

```
(46.1, 3.7)
```

[ ]

```
np.max(df['MinTemp'].values), np.min(df['MinTemp'].values)
```

```
(31.4, -6.7)
```

[ ]

```
Y
```

```
5939      0
5940      0
5942      0
5943      0
5944      0
         ..
139108    0
139109    0
139110    0
139111    0
139112    0
Name: RainTomorrow, Length: 56420, dtype: int64
```

[ ]

```
Y.value_counts()
```

```
0    43993
1    12427
Name: RainTomorrow, dtype: int64
```

[ ]

```
X.shape, Y.shape
```

```
((56420, 25), (56420,))
```

Normalization

[ ]

```
scaler = StandardScaler()

scaler.fit(X)

X = scaler.transform(X)

X
```

```
array([[-1.18184371,  0.69120848,  1.57529783, ..., -1.31421102,
        -1.57185009, -1.67651525],
       [-1.18184371,  0.76913098,  0.67150378, ..., -1.31421102,
        -1.57185009, -1.5626518 ],
       [-1.18184371,  0.92497598,  1.91960032, ..., -1.31421102,
        -1.57185009, -1.33492489],
       ...,
       [-0.90791487,  1.12757448,  1.23099533, ...,  1.94982075,
        -0.1231427 ,  0.71461729],
       [-0.90791487,  0.94056048,  1.08753596, ...,  1.94982075,
        -0.1231427 ,  0.82848075],
       [-0.90791487,  1.04965198,  1.07319002, ...,  1.94982075,
        -0.1231427 ,  0.9423442 ]])
```

[ ]

```
f = open("scaler.pkl", "wb")

pickle.dump(scaler, f)

f.close()
```

[ ]

```
np.random.seed(2)

x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.25,)

print(x_train.shape)

print(x_test.shape)
print(y_train.shape)

print(y_test.shape)
```

```
(42315, 25)
(14105, 25)
(42315,)
(14105,)
```

FEATURE-2

Model Training

[ ]

```
def get_score(name, model, X_train, X_test, y_train, y_test):

model.fit(X_train, y_train)

    pred = model.predict(X_test)

    print(model)

    print()

   score = metrics.accuracy_score(y_test,pred)

    cm = metrics.confusion_matrix(y_test,pred, labels = [0,1])

    print(cm)

    print()

    print("Training Accuracy : ",model.score(X_train, y_train))

    print()

    print("Testing Accuracy : ",model.score(X_test, y_test))

    print()
   joblib.dump(model, str(name)+str(model.score(X_test, y_test))+".joblib")
   print("################")
   return(model, model.score(X_test, y_test))
  lr, result = get_score("lr", LogisticRegression(solver='liblinear',multi_class='ovr',
                           class_weight='balanced'),
        x_train, x_test, y_train, y_test)
#svm, result = get_score("svm", SVC(gamma='auto', probability=True,
class_weight='balanced'),
 #       x_train, x_test, y_train, y_test)
rf, result = get_score("rf", RandomForestClassifier(n_estimators=69,
class_weight='balanced'),
        x_train, x_test, y_train, y_test)
knn, result = get_score("knn", KNeighborsClassifier(n_neighbors = 2,),
        x_train, x_test, y_train, y_test)
sgd, result = get_score("sgd", SGDClassifier(alpha=0.0001, average=False,
class_weight='balanced',
           early_stopping=False, epsilon=0.1, eta0=0.0, fit_intercept=True,
           l1_ratio=0.15, learning_rate='optimal', loss="modified_huber",
           max_iter=50, n_iter_no_change=5, n_jobs=None, penalty='none',
           power_t=0.5, random_state=None, shuffle=True, tol=0.001,
                 validation_fraction=0.1, verbose=0, warm_start=False), x_train, x_test,
 y_train, y_test)
```

```
  LogisticRegression(class_weight='balanced', multi_class='ovr',

                   solver='liblinear')

[[10940    82]
 [    1  3082]]

Training Accuracy :  0.9946591043365237

Testing Accuracy :  0.9941155618574974

################
RandomForestClassifier(class_weight='balanced', n_estimators=69)

[[11022     0]
 [    0  3083]]

Training Accuracy :  1.0

Testing Accuracy :  1.0

################
KNeighborsClassifier(n_neighbors=2)

[[10781   241]
 [ 1766  1317]]

Training Accuracy :  0.9148056244830438

Testing Accuracy :  0.8577100319035803

################
SGDClassifier(class_weight='balanced', loss='modified_huber', max_iter=50,
              penalty='none')

[[11021     1]
 [    0  3083]]
```

```
Training Accuracy :  1.0

Testing Accuracy :  0.9999291031549096

################
```

[ ]

[ ]

```
os.getcwd(), os.listdir()
```

```
('/home/wsuser/work', ['rf1.0.joblib', 'scaler.pkl', '.virtual_documents'])
```

[ ]

```
cos_client.upload_file(Filename='scaler.pkl',Bucket=bucket,Key='scaler.pkl')
```

[ ]

```
cos_client.upload_file(Filename='rf1.0.joblib',Bucket=bucket,Key='rf1.1.joblib')
```

[ ]

```
dirs = dir(cos_client)

','.join(dirs)
```

[ ]

```
help(cos_client.upload_file)
```

```
Help on method upload_file in module ibm_boto3.s3.inject:

upload_file(Filename,  Bucket,  Key,  ExtraArgs=None,  Callback=None,  Config=None)
method of ibm_botocore.client.S3 instance
    Upload a file to an S3 object.
```

Inference

[ ]

```
  rf = joblib.load("rf1.0.joblib")

bucket = 'rainfall-donotdelete-pr-eoii6hshulobg1'

  object_key = 'rainfall.csv'

  body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']

# add missing __iter__ method, so pandas accepts body as file-like object

  if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df = pd.read_csv(body)

  df

  df = df.dropna()

  df

 for i in df.columns:

    print(i, df[i].dtype)

    print(df[i].value_counts())

    print("################")

   cols = ['RainToday', 'RainTomorrow', 'WindDir3pm', 'WindDir9am', 'WindGustDir', 'Location']

  le = LabelEncoder()

  for col in cols:

    print(col)

    df[col] = le.fit_transform(df[col])
    le_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))

    print(le_name_mapping)

    print("################")

  df

  for col in df.columns:

    print(col, df[col].dtype)

df['Date'] = pd.to_datetime(df['Date'],

   format = '%Y-%m-%d',

   errors = 'coerce')

  Df

  for col in df.columns:

    print(col, df[col].dtype)

 df['year'] = df['Date'].dt.year

  df['month'] = df['Date'].dt.month

  df['day'] = df['Date'].dt.day

  df

  del df['Date']

  df

  for col in df.columns:

    print(col, df[col].dtype)
```

```
X = df.drop('RainTomorrow', axis=1)

Y = df['RainTomorrow']
```

[ ]

```
t = X.iloc[0].values

t
```

```
array([4.0000e+00, 1.7900e+01, 3.5200e+01, 0.0000e+00, 1.2000e+01,
       1.2300e+01, 1.1000e+01, 4.8000e+01, 1.0000e+00, 1.2000e+01,
       6.0000e+00, 2.0000e+01, 2.0000e+01, 1.3000e+01, 1.0063e+03,
       1.0044e+03, 2.0000e+00, 5.0000e+00, 2.6600e+01, 3.3400e+01,
       0.0000e+00, 0.0000e+00, 2.0090e+03, 1.0000e+00, 1.0000e+00])
```

[ ]

```
#RainTomorrow

#{'No': 0, 'Yes': 1}

Y.iloc[0]
```

```
0
```

[ ]

```
#lr_val = lr.predict_proba(scaled_X)

#svm_val = svm.predict_proba(scaled_X)

rf_val = rf.predict_proba(scaled_X)

#knn_val = knn.predict_proba(scaled_X)

#sgd_val = sgd.predict_proba(scaled_X)
```

[ ]

```
print(rf_val)
```

```
[[1. 0.]]
```

[ ]

```
#RainTomorrow

#{'No': 0, 'Yes': 1}
```

[ ]

```
#RainTomorrow

#{'No': 0, 'Yes': 1}

Y.iloc[0]

scaled_X = scaler.transform([t])

scaled_X

#lr_val = lr.predict_proba(scaled_X)

#svm_val = svm.predict_proba(scaled_X)

rf_val = rf.predict_proba(scaled_X)

#knn_val = knn.predict_proba(scaled_X)

#sgd_val = sgd.predict_proba(scaled_X)
```

[ ]

```
a = (lr_val+svm_val+rf_val+knn_val+sgd_val)
```

[ ]

```
cols  =  ['RainToday',  'RainTomorrow',  'WindDir3pm',  'WindDir9am',  'WindGustDir',
'Location']
```

[ ]

```
le = LabelEncoder()

for col in cols:

  print(col)

  df[col] = le.fit_transform(df[col])

  le_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))

  print(le_name_mapping)

  print("#################")
```

```
RainToday
{'No': 0, 'Yes': 1}
#################
RainTomorrow
{'No': 0, 'Yes': 1}
#################
WindDir3pm
{'E': 0, 'ENE': 1, 'ESE': 2, 'N': 3, 'NE': 4, 'NNE': 5, 'NNW': 6, 'NW': 7, 'S': 8, 'SE': 9,
'SSE': 10, 'SSW': 11, 'SW': 12, 'W': 13, 'WNW': 14, 'WSW': 15}
#################
WindDir9am
{'E': 0, 'ENE': 1, 'ESE': 2, 'N': 3, 'NE': 4, 'NNE': 5, 'NNW': 6, 'NW': 7, 'S': 8, 'SE': 9,
'SSE': 10, 'SSW': 11, 'SW': 12, 'W': 13, 'WNW': 14, 'WSW': 15}
#################
WindGustDir
{'E': 0, 'ENE': 1, 'ESE': 2, 'N': 3, 'NE': 4, 'NNE': 5, 'NNW': 6, 'NW': 7, 'S': 8, 'SE': 9,
'SSE': 10, 'SSW': 11, 'SW': 12, 'W': 13, 'WNW': 14, 'WSW': 15}
#################
Location
{'AliceSprings': 0, 'Brisbane': 1, 'Cairns': 2, 'Canberra': 3, 'Cobar': 4, 'CoffsHarbour': 5,
'Darwin': 6, 'Hobart': 7, 'Melbourne': 8, 'MelbourneAirport': 9, 'Mildura': 10, 'Moree': 11,
'MountGambier': 12, 'NorfolkIsland': 13, 'Nuriootpa': 14, 'Perth': 15, 'PerthAirport': 16,
'Portland': 17, 'Sale': 18, 'Sydney': 19, 'SydneyAirport': 20, 'Townsville': 21,
'WaggaWagga': 22, 'Watsonia': 23, 'Williamtown': 24, 'Woomera': 25}
```

```
################
/tmp/wsuser/ipykernel_164/3716255542.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

[ ]

```
for col in df.columns:

  print(col, df[col].dtype)
```

[ ]

```
df['Date'] = pd.to_datetime(df['Date'],

 format = '%Y-%m-%d',

 errors = 'coerce')

df
```

[ ]

```
for col in df.columns:

  print(col, df[col].dtype)
```

```
Date datetime64[ns]
Location int64
MinTemp float64
MaxTemp float64
Rainfall float64
Evaporation float64
Sunshine float64
WindGustDir int64
WindGustSpeed float64
WindDir9am int64
WindDir3pm int64
WindSpeed9am float64
WindSpeed3pm float64
```

```
Humidity9am float64
Humidity3pm float64
Pressure9am float64
Pressure3pm float64
Cloud9am float64
Cloud3pm float64
Temp9am float64
Temp3pm float64
RainToday int64
RISK_MM float64
RainTomorrow int64
```

[ ]

```
df['year'] = df['Date'].dt.year

df['month'] = df['Date'].dt.month

df['day'] = df['Date'].dt.day

df
```

[ ]

```
del df['Date']
```

[ ]

```
Df
```

[ ]

```
for col in df.columns:

  print(col, df[col].dtype)
```

[ ]

```
X = df.drop('RainTomorrow', axis=1)
```

```
Y = df['RainTomorrow']
```

[ ]

```
X
```

[ ]

```
np.max(df['Temp3pm'].values), np.min(df['Temp3pm'].values)
```

```
(46.1, 3.7)
```

[ ]

```
np.max(df['MinTemp'].values), np.min(df['MinTemp'].values)
```

```
(31.4, -6.7)
```

[ ]

```
Y
```

```
5939      0
5940      0
5942      0
5943      0
5944      0
         ..
139108    0
139109    0
139110    0
139111    0
139112    0
Name: RainTomorrow, Length: 56420, dtype: int64
```

[ ]

```
Y.value_counts()
```

```
0    43993
1    12427
```

```
Name: RainTomorrow, dtype: int64
```

[ ]

```
X.shape, Y.shape
```

```
((56420, 25), (56420,))
```

Normalization

[ ]

```
scaler = StandardScaler()

scaler.fit(X)

X = scaler.transform(X)

X
```

```
array([[-1.18184371,  0.69120848,  1.57529783, ..., -1.31421102,
        -1.57185009, -1.67651525],
       [-1.18184371,  0.76913098,  0.67150378, ..., -1.31421102,
        -1.57185009, -1.5626518 ],
       [-1.18184371,  0.92497598,  1.91960032, ..., -1.31421102,
        -1.57185009, -1.33492489],
       ...,
       [-0.90791487,  1.12757448,  1.23099533, ...,  1.94982075,
        -0.1231427 ,  0.71461729],
       [-0.90791487,  0.94056048,  1.08753596, ...,  1.94982075,
        -0.1231427 ,  0.82848075],
       [-0.90791487,  1.04965198,  1.07319002, ...,  1.94982075,
        -0.1231427 ,  0.9423442 ]])
```

[ ]

```
f = open("scaler.pkl", "wb")

pickle.dump(scaler, f)

f.close()
```

```
 [ ]

  np.random.seed(2)

  x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.25,)

  print(x_train.shape)

  print(x_test.shape)

  print(y_train.shape)

  print(y_test.shape)
```

```
(42315, 25)
(14105, 25)
(42315,)
(14105,)
```

<u>Inference</u>

```
 [ ]

  rf = joblib.load("rf1.0.joblib")

bucket = 'rainfall-donotdelete-pr-eoii6hshulobg1'

  object_key = 'rainfall.csv'

  body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']

# add missing __iter__ method, so pandas accepts body as file-like object

  if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df = pd.read_csv(body)

  df
 df = df.dropna()

  df

  for i in df.columns:

    print(i, df[i].dtype)

    print(df[i].value_counts())

    print("################")

  cols = ['RainToday', 'RainTomorrow', 'WindDir3pm', 'WindDir9am', 'WindGustDir',
  'Location']

  le = LabelEncoder()

  for col in cols:

    print(col)

    df[col] = le.fit_transform(df[col])

    le_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))

    print(le_name_mapping)

    print("################")

Df

  for col in df.columns:

    print(col, df[col].dtype)

df['Date'] = pd.to_datetime(df['Date'],

   format = '%Y-%m-%d',
   errors = 'coerce')

  df

for col in df.columns:

    print(col, df[col].dtype)

  df['year'] = df['Date'].dt.year

  df['month'] = df['Date'].dt.month

  df['day'] = df['Date'].dt.day

  Df

  del df['Date']

  df

  for col in df.columns:

    print(col, df[col].dtype)

   X = df.drop('RainTomorrow', axis=1)

  Y = df['RainTomorrow']

[ ]

  t = X.iloc[0].values

  t
```

```
array([4.0000e+00, 1.7900e+01, 3.5200e+01, 0.0000e+00, 1.2000e+01,
       1.2300e+01, 1.1000e+01, 4.8000e+01, 1.0000e+00, 1.2000e+01,
       6.0000e+00, 2.0000e+01, 2.0000e+01, 1.3000e+01, 1.0063e+03,
       1.0044e+03, 2.0000e+00, 5.0000e+00, 2.6600e+01, 3.3400e+01,
        0.0000e+00, 0.0000e+00, 2.0090e+03, 1.0000e+00, 1.0000e+00])
```

[ ]

```
#RainTomorrow

#{'No': 0, 'Yes': 1}

Y.iloc[0]
```

```
0
```

[ ]

```
scaled_X = scaler.transform([t])

scaled_X
```

[ ]

```
type(rf)
```

[ ]

```
#lr_val = lr.predict_proba(scaled_X)

#svm_val = svm.predict_proba(scaled_X)

rf_val = rf.predict_proba(scaled_X)

#knn_val = knn.predict_proba(scaled_X)

#sgd_val = sgd.predict_proba(scaled_X)
```

[ ]

```
print(rf_val)
```

```
[[1. 0.]]
```

[ ]

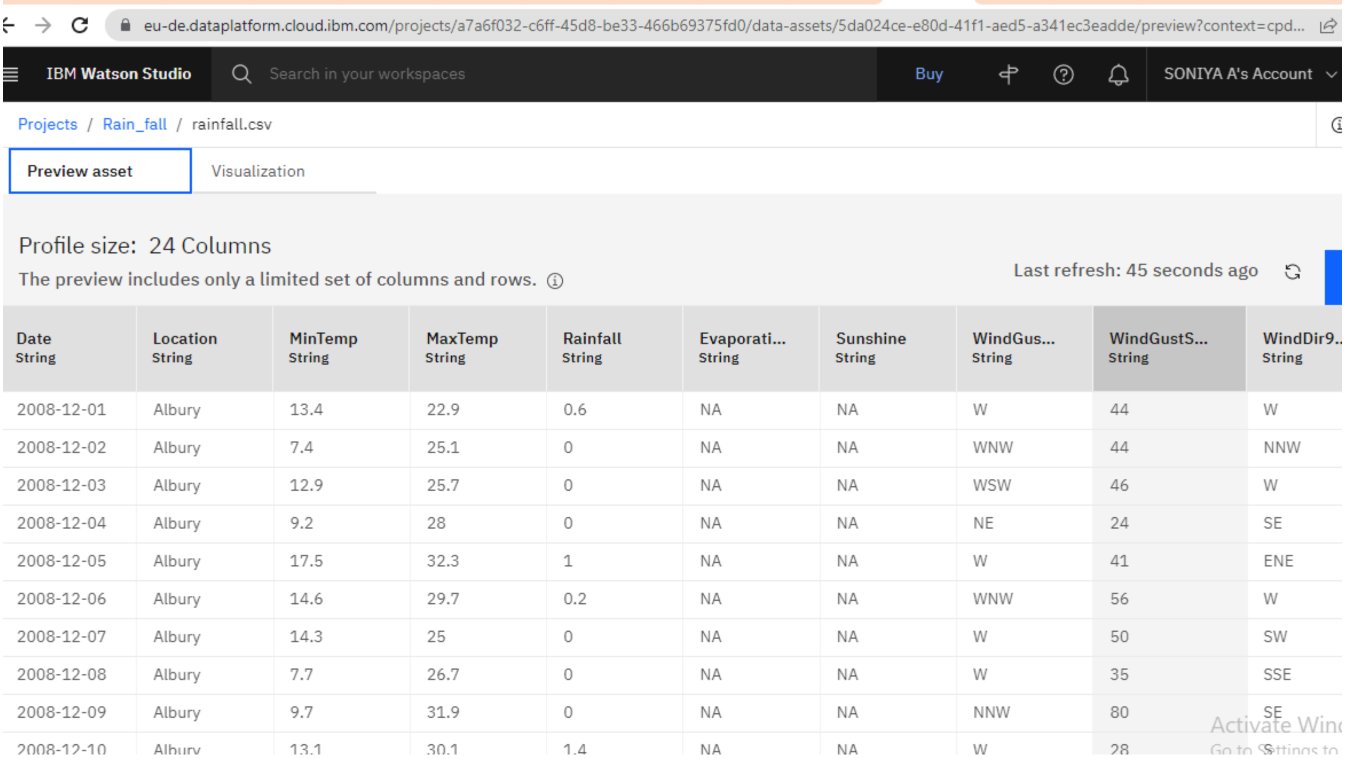
```
#RainTomorrow
```

```
 #{'No': 0, 'Yes': 1}

[ ]

 #RainTomorrow

 #{'No': 0, 'Yes': 1}

 Y.iloc[0]

 scaled_X = scaler.transform([t])

 scaled_X

 #lr_val = lr.predict_proba(scaled_X)

 #svm_val = svm.predict_proba(scaled_X)

 rf_val = rf.predict_proba(scaled_X)

 #knn_val = knn.predict_proba(scaled_X)

 #sgd_val = sgd.predict_proba(scaled_X)

[ ]

 a = (lr_val+svm_val+rf_val+knn_val+sgd_val)/5

[ ]

 a
```

8.TESTING

Test cases help guide the tester through a sequence of steps to validate whether a software application is free of bugs, and working as required by the end-user. Learning how to write test cases for software requires basic writing skills, attention to detail, and a good understanding of the application under test (AUT).

8.1 TEST CASES

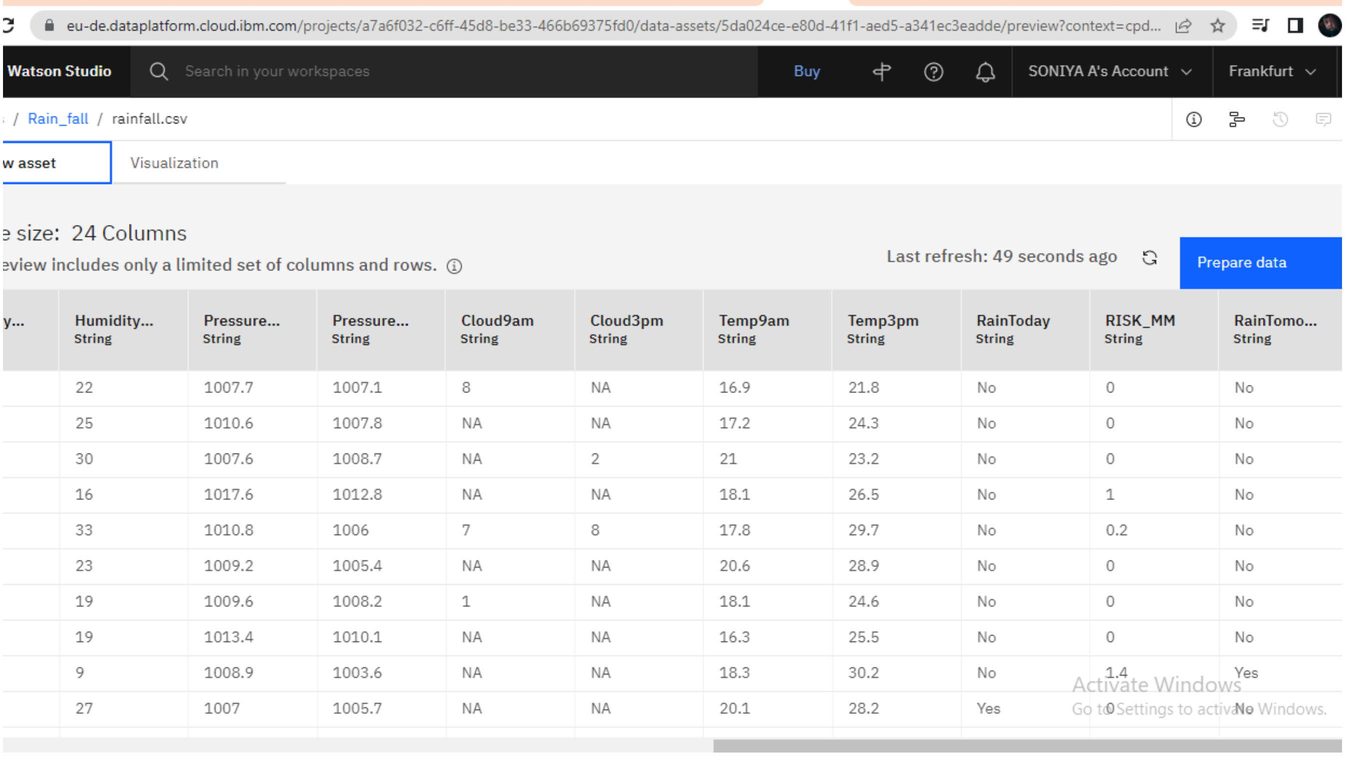

eu-de.dataplatform.cloud.ibm.com/projects/a7a6f032-c6ff-45d8-be33-466b69375fd0/data-assets/5da024ce-e80d-41f1-aed5-a341ec3eadde/preview?context=cpd...
IBM Watson Studio
Search in your workspaces
Buy
SONIYA A's Account
Projects / Rain_fall / rainfall.csv
Preview asset
Visualization

Profile size: 24 Columns

The preview includes only a limited set of columns and rows.

Last refresh: 45 seconds ago

Date String	Location String	MinTemp String	MaxTemp String	Rainfall String	Evaporati... String	Sunshine String	WindGus... String	WindGustS... String	WindDir9.. String
2008-12-01	Albury	13.4	22.9	0.6	NA	NA	W	44	W
2008-12-02	Albury	7.4	25.1	0	NA	NA	WNW	44	NNW
2008-12-03	Albury	12.9	25.7	0	NA	NA	WSW	46	W
2008-12-04	Albury	9.2	28	0	NA	NA	NE	24	SE
2008-12-05	Albury	17.5	32.3	1	NA	NA	W	41	ENE
2008-12-06	Albury	14.6	29.7	0.2	NA	NA	WNW	56	W
2008-12-07	Albury	14.3	25	0	NA	NA	W	50	SW
2008-12-08	Albury	7.7	26.7	0	NA	NA	W	35	SSE
2008-12-09	Albury	9.7	31.9	0	NA	NA	NNW	80	SE
2008-12-10	Albury	13.1	30.1	1.4	NA	NA	W	28	


eu-de.dataplatform.cloud.ibm.com/projects/a7a6f032-c6ff-45d8-be33-466b69375fd0/data-assets/5da024ce-e80d-41f1-aed5-a341ec3eadde/preview?context=cpd...
Watson Studio
Search in your workspaces
Buy
SONIYA A's Account
Frankfurt
/ Rain_fall / rainfall.csv
w asset
Visualization
Prepare data

e size: 24 Columns

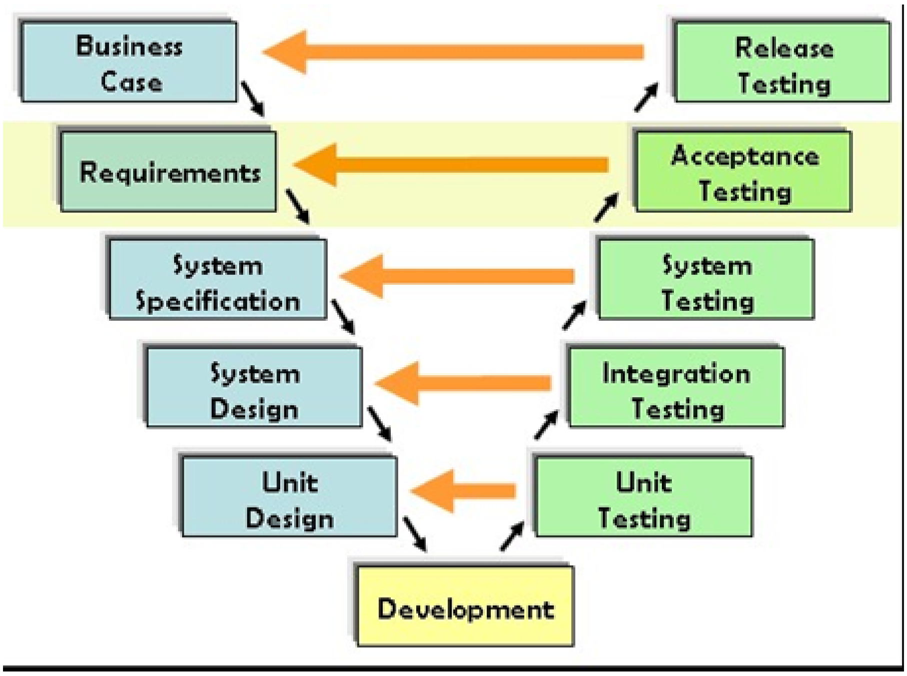
eview includes only a limited set of columns and rows.

Last refresh: 49 seconds ago

y...	Humidity... String	Pressure... String	Pressure... String	Cloud9am String	Cloud3pm String	Temp9am String	Temp3pm String	RainToday String	RISK_MM String	RainTomo... String
	22	1007.7	1007.1	8	NA	16.9	21.8	No	0	No
	25	1010.6	1007.8	NA	NA	17.2	24.3	No	0	No
	30	1007.6	1008.7	NA	2	21	23.2	No	0	No
	16	1017.6	1012.8	NA	NA	18.1	26.5	No	1	No
	33	1010.8	1006	7	8	17.8	29.7	No	0.2	No
	23	1009.2	1005.4	NA	NA	20.6	28.9	No	0	No
	19	1009.6	1008.2	1	NA	18.1	24.6	No	0	No
	19	1013.4	1010.1	NA	NA	16.3	25.5	No	0	No
	9	1008.9	1003.6	NA	NA	18.3	30.2	No	1.4	Yes
	27	1007	1005.7	NA	NA	20.1	28.2	Yes	0	No

8.2 USER ACCEPTANCE TESTING

UAT consists, in practice, of people from the target audience using the application. The defects they find are then reported and fixed. This scenario is what most closely resembles "the real world." The process allows users to "get their hands dirty" with the application. They can see if things work as intended**.**



The main purpose of UAT is to validate end-to-end business flow. It does not focus on cosmetic errors, spelling mistakes, or system testing. User Acceptance Testing is carried out in a separate testing environment with a production-like data setup. It is a kind of black box testing where two or more end-users will be involved.

9.RESULTS

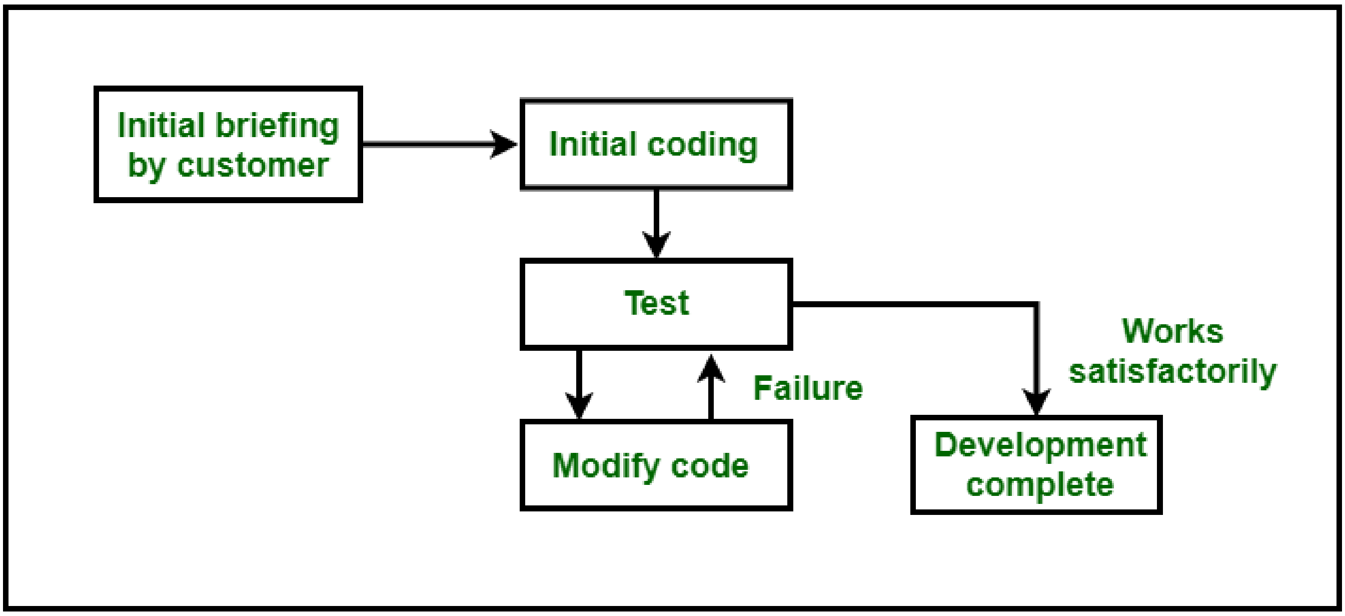
Let's break down the output.

Precision — What percent of your predictions were correct?
Recall — What percent of the positive cases did you catch?
F1-score — The weighted average of Precision and
Recall support —The number of occurrences of each given class.

9.1 PERFORMANCE METRICS

TP - True Positive | FP - False Positive
FN - False Negative | TN - True NegativePrecision = TP/(FP+TP) = 3/(3+4+2) = 3/9 = 0.33 Recall = TP/(TP+FN) = 3/(3+1+0) = 3/4 = 0.75F1-score = 2 * (Recall * Precision) / (Recall + Precision)
= 2 * 0.33 * 0.75 / (0.33 + 0.75)
= 0.495 / 1.08

= 0.458 .



10.ADVANTAGES & DISADVANTAGES

Advantages:

1) It gives us valuable insights into the data.

2)It helps us with feature selection (i.e using PCA)

3)Visualisation is an effective way of detection.Discover errors , outliers, and missing values in the data.

Disadvantages

1)Lack of standardised analysis

2)Small sample population and outdated information that can adversely affect the authenticity of information

3)While explanatory research does help you solidify your theories and hypotheses, it usually lacks conclusive results.

11)CONCLUSION

This Paper has presented a supervised rainfall learning model which used machine learning algorithms to classify rainfall data. We used different machine learning algorithms to check the accuracy of rainfall prediction. We have compared SVM, Random Forest, Naive Bayes and MLP (Multilayer perceptron) classifiers. From the above figure 3 we can conclude that Random forest is the Machine learning algorithm which is suitable for rainfall prediction in India.

Currently machine learning is used in no. industries. As the data increases the complexity of that data will increase and for that we are using machines for the better understanding of that data. In Weather predictions it's pretty helpful with a good accuracy score and in rainfall also it gives pretty good predictions. In future we are planning to increase our work in Storm predictions and Crop prediction with the rainfall prediction.

12)FUTURE SCOPE

To understand the reasons why you should use this framework to create your app or website, you need to know the origin of Flask – Python.

It’s a system using which you can create your apps and website faster and easier. It can expedite your development process by providing code for all types of methods like the file system and database communication.

Based on Python, it has been considered as a fantastic programming language for quick and top-notch app development.

Many popular sites have been created by python including Google, Yahoo, YouTube, Instagram, Quora, Dropbox, and many more. It’s also a wide range of extensions which help to integrate it all well.

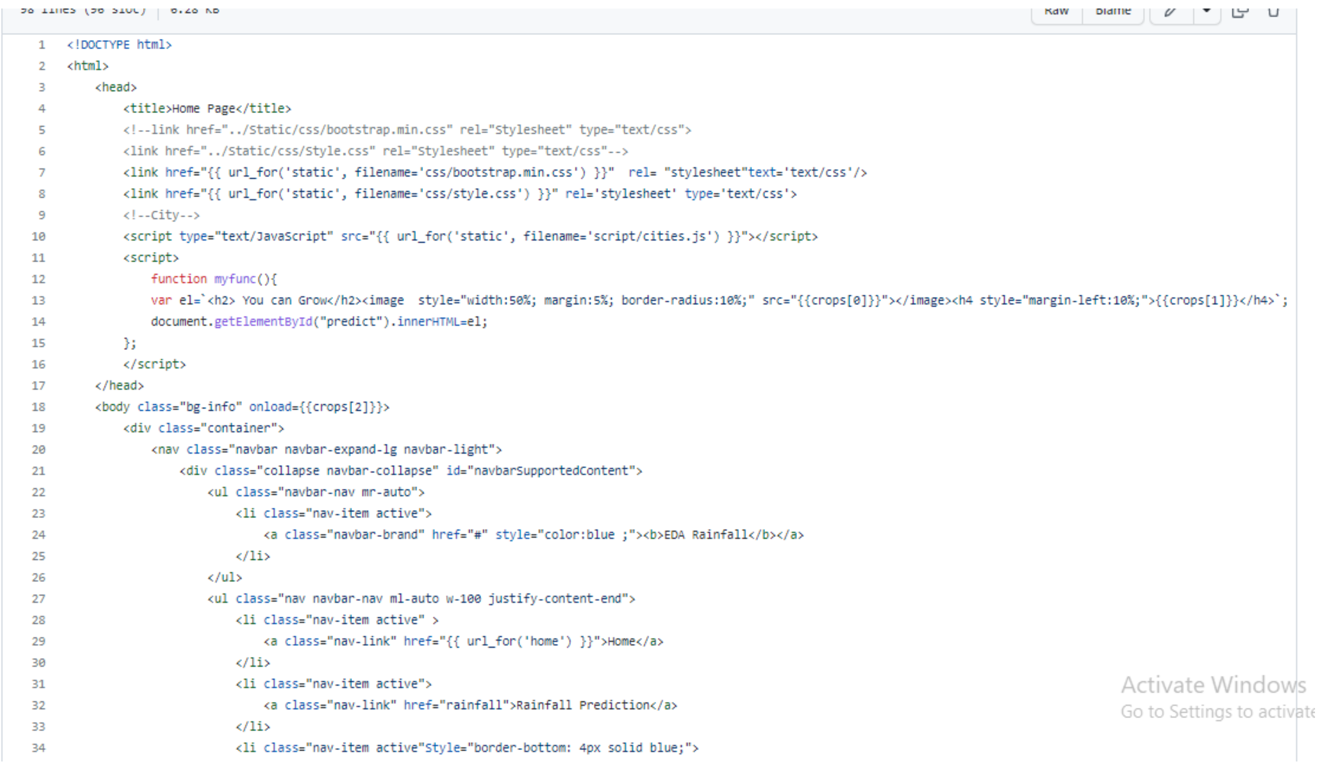
These flask applications are mainly made for farmers, which helps the farmers to know the weather conditions due to heavy rainfall and it is cost free.

13)APPENDIX

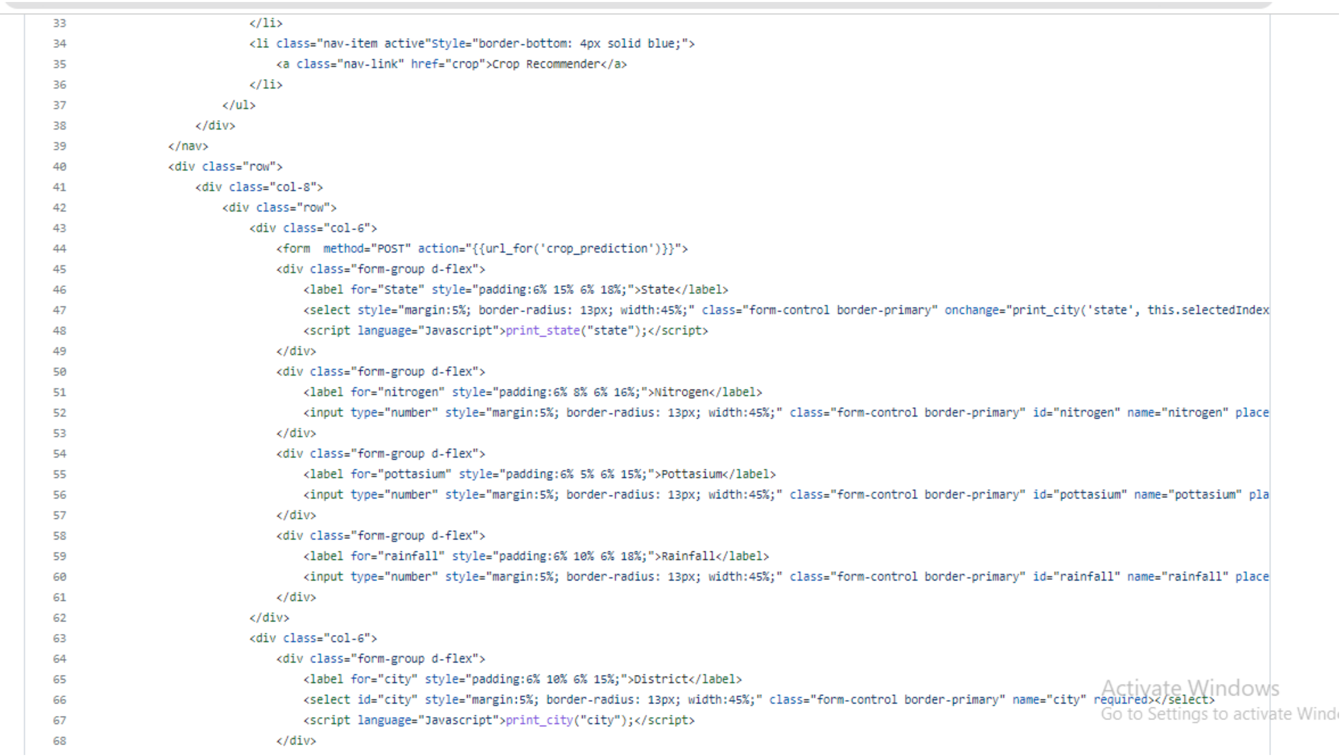
SOURCE CODE

```
from flask import Flask,render_template,request
from Static.utils.Geoloc import geolo
from Static.utils  import Home
app=Flask(__name__)

@app.route("/")
def login():
    return render_template("Login.html")

@app.route("/signin")
def signin():
    return render_template("Signup.html")

@app.route('/login', methods=['POST','GET'])
def welcome():
    email=""
    password=""
    if request.method=='POST':
        email=str(request.form['email'])
        password=str(request.form['password'])
    if(Home.logins(email,password)):
        return render_template("Welcome.html")
    else:
        return render_template("login.html",wrong="myFunction()")

@app.route('/signins', methods=['POST','GET'])
def signins():
    email=""
    password=""
    name=""
    mobile=""
    if request.method=='POST':
        name=str(request.form['name'])
        email=str(request.form['email'])
        mobile=str(request.form['mobile'])
        password=str(request.form['password'])
        print(name,email,mobile,password)
    Home.signups(email,password,name,mobile)
    return render_template("login.html")


@app.route('/home', methods=['POST','GET'])
def home():
    city=""
    if request.method=='POST':
        lat=float(request.form['lat'])
        lon=float(request.form['lon'])
        city=geolo(lat,lon)
        print(city)
    return render_template("Home.html",city=city)


if __name__ == '__main__':
   app.run(debug = True)
```

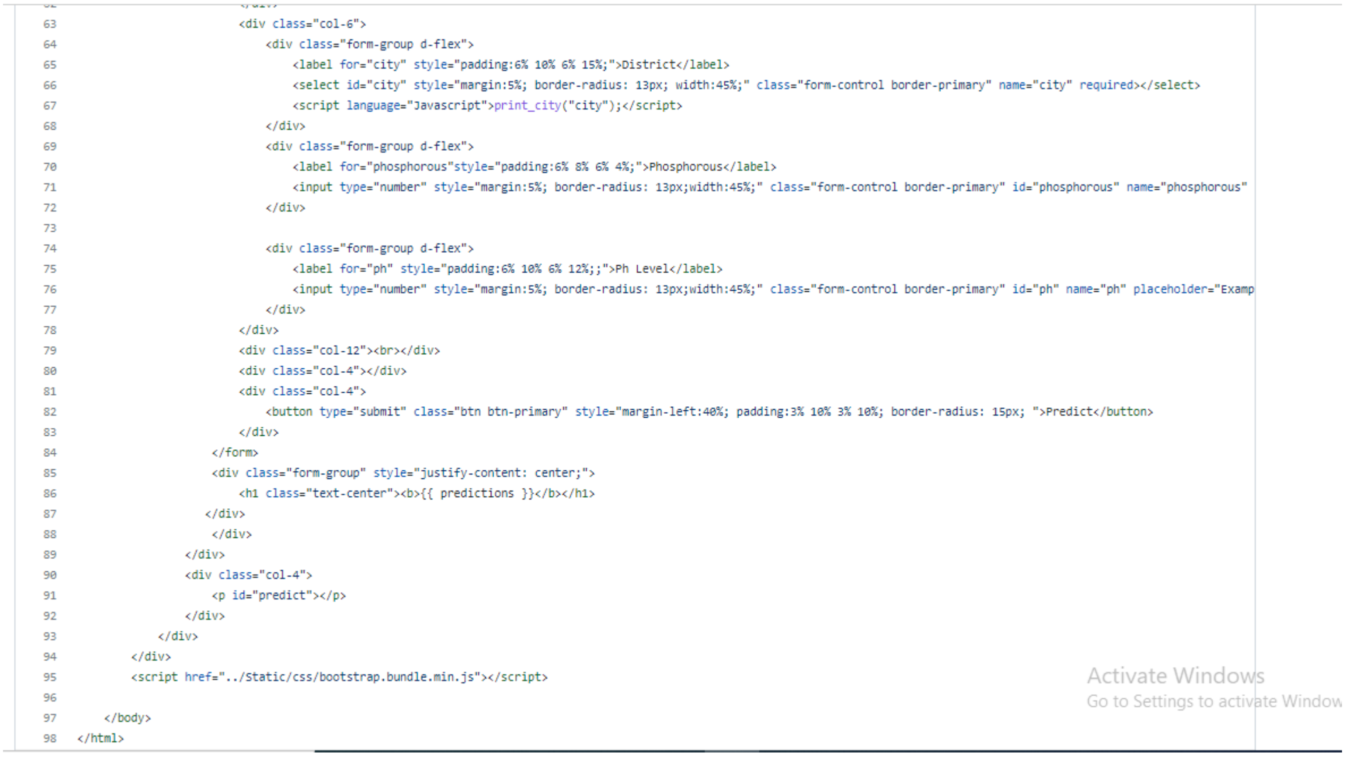
HTML CODE



```html
<!DOCTYPE html>
<html>
    <head>
        <title>Home Page</title>
        <!--link href="../Static/css/bootstrap.min.css" rel="Stylesheet" type="text/css">
        <link href="../Static/css/Style.css" rel="Stylesheet" type="text/css"-->
        <link href="{{ url_for('static', filename='css/bootstrap.min.css') }}"  rel= "stylesheet"text='text/css'/>
        <link href="{{ url_for('static', filename='css/style.css') }}" rel='stylesheet' type='text/css'>
        <!--City-->
        <script type="text/JavaScript" src="{{ url_for('static', filename='script/cities.js') }}"></script>
        <script>
            function myfunc(){
            var el=`<h2> You can Grow</h2><image  style="width:50%; margin:5%; border-radius:10%;" src="{{crops[0]}}"></image><h4 style="margin-left:10%;">{{crops[1]}}</h4>`;
            document.getElementById("predict").innerHTML=el;
        };
        </script>
    </head>
    <body class="bg-info" onload={{crops[2]}}>
        <div class="container">
            <nav class="navbar navbar-expand-lg navbar-light">
                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <ul class="navbar-nav mr-auto">
                        <li class="nav-item active">
                            <a class="navbar-brand" href="#" style="color:blue ;"><b>EDA Rainfall</b></a>
                        </li>
                    </ul>
                    <ul class="nav navbar-nav ml-auto w-100 justify-content-end">
                        <li class="nav-item active" >
                            <a class="nav-link" href="{{ url_for('home') }}">Home</a>
                        </li>
                        <li class="nav-item active">
                            <a class="nav-link" href="rainfall">Rainfall Prediction</a>
                        </li>
                        <li class="nav-item active"Style="border-bottom: 4px solid blue;">
```



```
                        </li>
                        <li class="nav-item active"Style="border-bottom: 4px solid blue;">
                            <a class="nav-link" href="crop">Crop Recommender</a>
                        </li>
                    </ul>
                </div>
            </nav>
            <div class="row">
                <div class="col-8">
                    <div class="row">
                        <div class="col-6">
                            <form  method="POST" action="{{url_for('crop_prediction')}}">
                            <div class="form-group d-flex">
                                <label for="State" style="padding:6% 15% 6% 18%;">State</label>
                                <select style="margin:5%; border-radius: 13px; width:45%;" class="form-control border-primary" onchange="print_city('state', this.selectedIndex
                                <script language="Javascript">print_state("state");</script>
                            </div>
                            <div class="form-group d-flex">
                                <label for="nitrogen" style="padding:6% 8% 6% 16%;">Nitrogen</label>
                                <input type="number" style="margin:5%; border-radius: 13px; width:45%;" class="form-control border-primary" id="nitrogen" name="nitrogen" place
                            </div>
                            <div class="form-group d-flex">
                                <label for="pottasium" style="padding:6% 5% 6% 15%;">Pottasium</label>
                                <input type="number" style="margin:5%; border-radius: 13px; width:45%;" class="form-control border-primary" id="pottasium" name="pottasium" pla
                            </div>
                            <div class="form-group d-flex">
                                <label for="rainfall" style="padding:6% 10% 6% 18%;">Rainfall</label>
                                <input type="number" style="margin:5%; border-radius: 13px; width:45%;" class="form-control border-primary" id="rainfall" name="rainfall" place
                            </div>
                        </div>
                        <div class="col-6">
                            <div class="form-group d-flex">
                                <label for="city" style="padding:6% 10% 6% 15%;">District</label>
                                <select id="city" style="margin:5%; border-radius: 13px; width:45%;" class="form-control border-primary" name="city" required></select>
                                <script language="Javascript">print_city("city");</script>
                            </div>
```



```
                        <div class="col-6">
                            <div class="form-group d-flex">
                                <label for="city" style="padding:6% 10% 6% 15%;">District</label>
                                <select id="city" style="margin:5%; border-radius: 13px; width:45%;" class="form-control border-primary" name="city" required></select>
                                <script language="Javascript">print_city("city");</script>
                            </div>
                            <div class="form-group d-flex">
                                <label for="phosphorous"style="padding:6% 8% 6% 4%;">Phosphorous</label>
                                <input type="number" style="margin:5%; border-radius: 13px;width:45%;" class="form-control border-primary" id="phosphorous" name="phosphorous"
                            </div>

                            <div class="form-group d-flex">
                                <label for="ph" style="padding:6% 10% 6% 12%;;">Ph Level</label>
                                <input type="number" style="margin:5%; border-radius: 13px;width:45%;" class="form-control border-primary" id="ph" name="ph" placeholder="Examp
                            </div>
                        </div>
                        <div class="col-12"><br></div>
                        <div class="col-4"></div>
                        <div class="col-4">
                            <button type="submit" class="btn btn-primary" style="margin-left:40%; padding:3% 10% 3% 10%; border-radius: 15px; ">Predict</button>
                        </div>
                    </form>
                    <div class="form-group" style="justify-content: center;">
                        <h1 class="text-center"><b>{{ predictions }}</b></h1>
                   </div>
                    </div>
                </div>
                <div class="col-4">
                    <p id="predict"></p>
                </div>
            </div>
        </div>
        <script href="../Static/css/bootstrap.bundle.min.js"></script>

    </body>
</html>
```

GITHUB AND PROJECT DEMO LINK

GITHUB

IBM-EPBL/IBM-Project-17869-1659676805

PROJECT DEMO LINK

https://drive.google.com/file/d/18ghTrMqXJGuIamFE6Mc86eh0uCl7CY7r/view?usp=drivesdk