Project Report

1. INTRODUCTION

1.1 Project Overview

This Project **"EXPLORATORY ANALYSIS ON RAINFALL DATA IN INDIA FOR AGRICULTURE"** relys on helping our farmers and agriculturalist in India to predict the rainfall rate for a month in their respective state.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'll be using various regression techniques to predict the rainfall in India for a month in particular state with the usage of rainfall dataset and crop recommendation dataset from kaggle website. The farmers or the end users can easily access our site by providing the inputs as details of month number and the state name of India.

1.2 Purpose

Rainfall prediction is playing an increasingly instrumental role in the evolving field of precision agriculture, a farming practice that emphasizes accuracy and control when it comes to the growing of crops. An essential aspect of this approach is the use of information technology, which includes weather prediction and other items.The ultimate purpose of precision agriculture is to maximize growth efficiency at the individual seed and plant level.

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. And so we used Random forest regressor for predicting the rainfall rate and given as inout to GaussianNB claasification method for predicting the best crop to grow in the considered month.

2. LITERATURE SURVEY

2.1 Existing problem

In today's time the problems faced by the farmers is that the selection of appropriate crop for agriculture. There are several factors that affect crop yields, including rain, temperature, and soil. Crop forecasting helps farmers choose the right crops to plant to maximize yields. With the help of data mining techniques, it was even possible to make accurate predictions of the yield by taking environmental parameters into account. During this work, the classifiers used support vector engines and region-wise processing. Crop forecasts are completed taking into account parameters such as rainfall, minimum and maximum temperatures, soil type, moisture and soil, pH and layer price. The knowledge is collected from the agricultural website of geographical area. the knowledge is split into 9 agricultural zones. The employee interface is designed to allow farmers to enter given information to predict their harvest. Soil classification is necessary for farmers to recognize soil types and to cultivate crops according to soil type.

2.2 References

Project Title	Author	Objective/Outcome
A Novel Study of Rainfall in the Indian States and Predictive Analysis using Machine Learning Algorithms	Nikhil Tiwari, AnmolSingh	*) This research proposes a study and analysis of rainfall in the Indian states using ML algorithms.The model proposed is Neural Networks. *) The attempt was made to improve the previous results by ensembleML algorithms.

Rainfall Prediction usingNeural Network	Dr. Maulika S. Patel, VidhiRajvir, Devanshi Shukla	*) Rainfall prediction plays an important role in agricultural In this work,multilayered neural network with Back- propagation learning algorithm is used. *) It have configured Feed forward and cascade network with 1000 epoch-and achieved 82% and 81% accuracy respectively.
Heuristic prediction of rainfall using machine learning techniques	Chandrasegar Thirumalai,M Lakshmi Deepak.	*) This paper discusses the rate of rainfall in previous years according to various crops seasons and predicts the rainfall in future seasons. *) The paper also uses linear regression method in metrics.

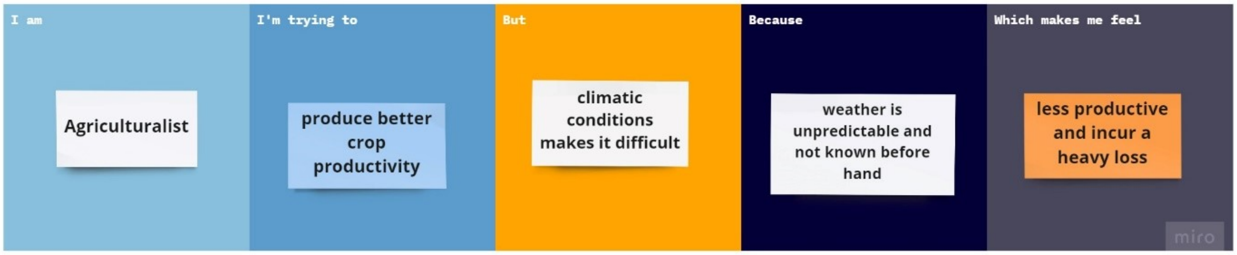
2.3 Problem Statement Definition

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:

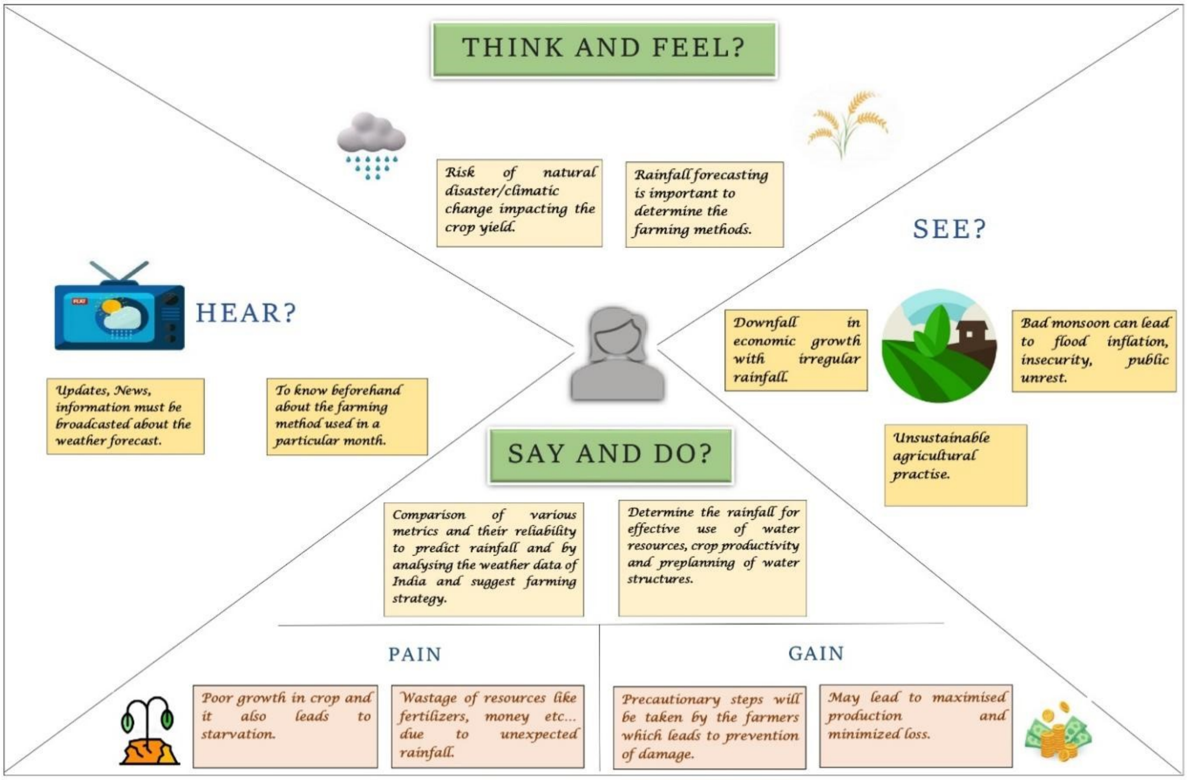


I am	I'm trying to	But	Because	Which makes me feel
Agriculturalist	produce better crop productivity	climatic conditions makes it difficult	weather is unpredictable and not known before hand	less productive and incur a heavy loss

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

EMPATHY MAP



THINK AND FEEL?

Risk of natural disaster/climatic change impacting the crop yield.

Rainfall forecasting is important to determine the farming methods.

HEAR?

Updates, News, information must be broadcasted about the weather forecast.

To know beforehand about the farming method used in a particular month.

SEE?

Downfall in economic growth with irregular rainfall.

Bad monsoon can lead to flood inflation, insecurity, public unrest.

Unsustainable agricultural practise.

SAY AND DO?

Comparison of various metrics and their reliability to predict rainfall and by analysing the weather data of India and suggest farming strategy.

Determine the rainfall for effective use of water resources, crop productivity and preplanning of water structures.

PAIN

Poor growth in crop and it also leads to starvation.

Wastage of resources like fertilizers, money etc... due to unexpected rainfall.

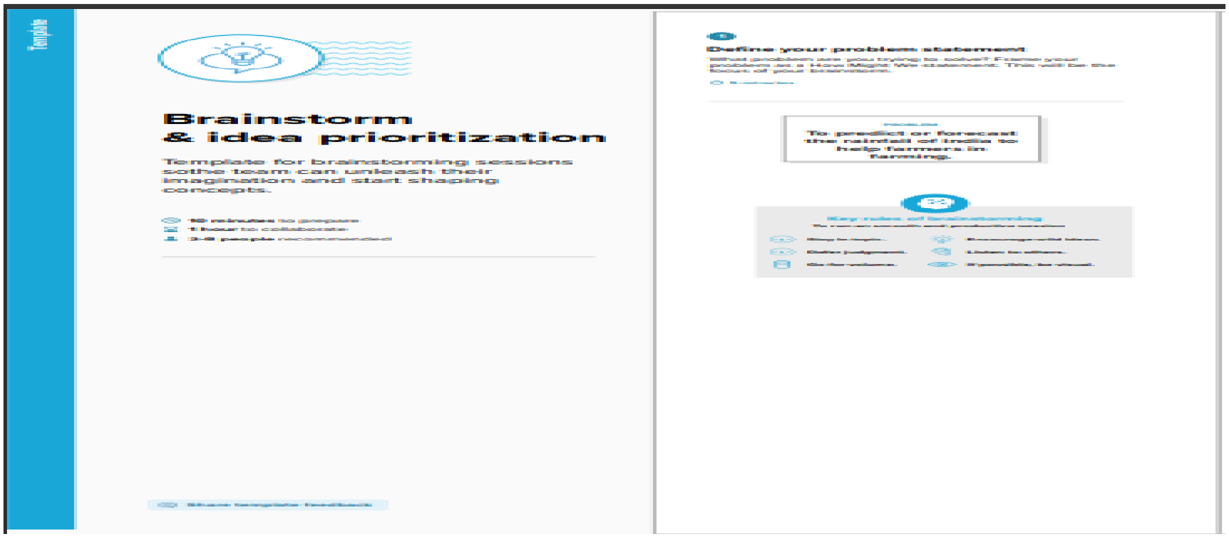
GAIN

Precautionary steps will be taken by the farmers which leads to prevention of damage.

May lead to maximised production and minimized loss.

3.2 Ideation & Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Template

Brainstorm & idea prioritization

Template for brainstorming sessions sothe team can unleash their imagination and start shaping concepts.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

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.

5 minutes

PROBLEM

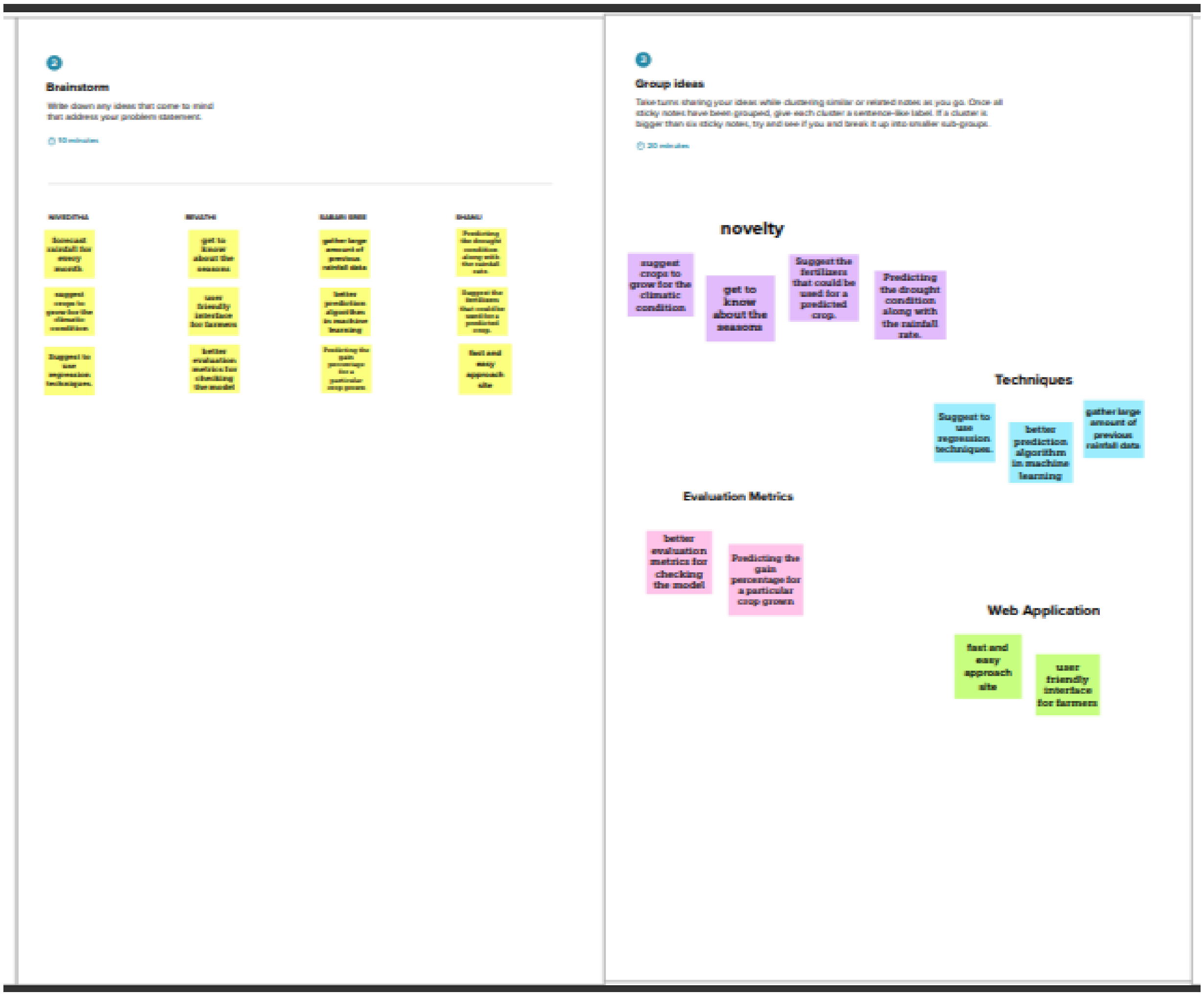
To predict or forecast the rainfall of India to help farmers in farming.

Key rules of brainstorming

To run an smooth and productive session

Stay in topic. Encourage wild ideas.
Defer judgment. Listen to others.
Go for volume. If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

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

10 minutes

NIVEDITHA	RIVATHI	SABARI SREE	DHANU
forecast rainfall for every month	get to know about the seasons	gather large amount of previous rainfall data	Predicting the drought condition along with the rainfall rate.
suggest crops to grow for the climatic condition.	user friendly interface for farmers	better prediction algorithm in machine learning	Suggest the fertilizers that could be used for a predicted crop.
Suggest to use regression techniques.	better evaluation metrics for checking the model	Predicting the gain percentage for a particular crop grown	fast and easy approach site

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

novelty

- suggest crops to grow for the climatic condition.
- get to know about the seasons
- Suggest the fertilizers that could be used for a predicted crop.
- Predicting the drought condition along with the rainfall rate.

Techniques

- Suggest to use regression techniques.
- better prediction algorithm in machine learning
- gather large amount of previous rainfall data

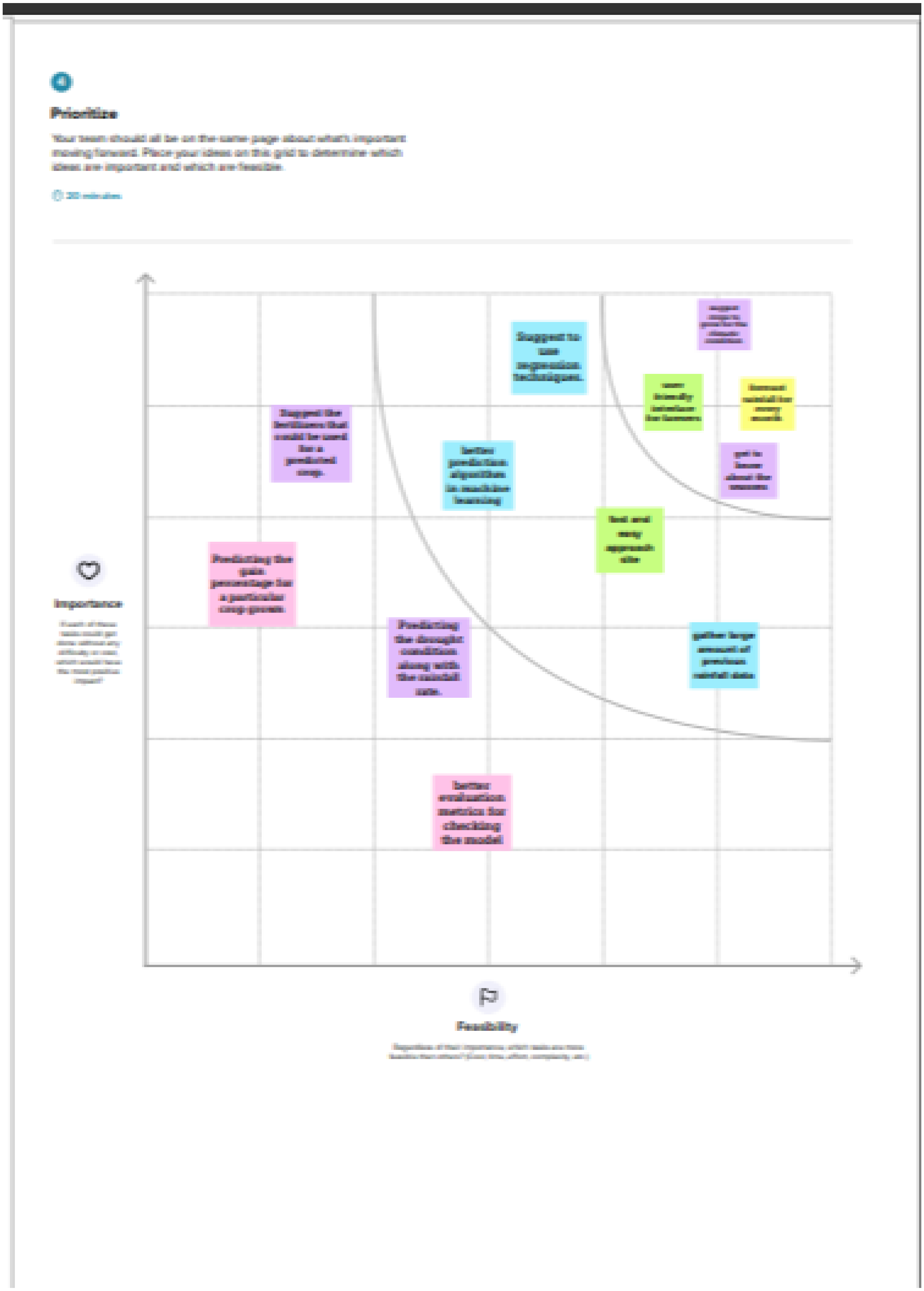
Evaluation Metrics

- better evaluation metrics for checking the model
- Predicting the gain percentage for a particular crop grown

Web Application

- fast and easy approach site
- user friendly interface for farmers

Step-3: Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

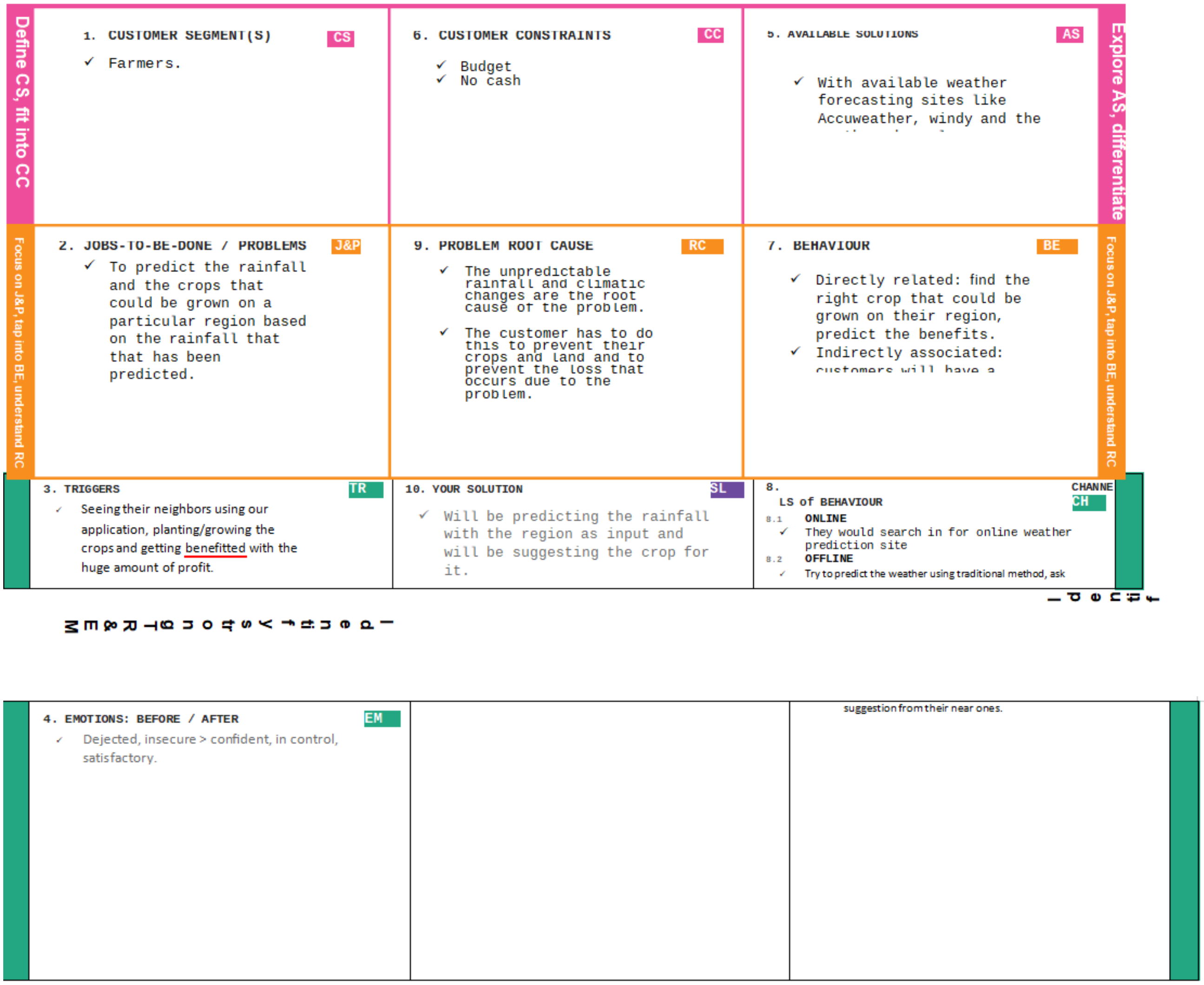
suggest crops to grow for the climatic condition
Suggest to use regression techniques.
user friendly interface for farmers
forecast rainfall for every month
Suggest the fertilizers that could be used for a predicted crop.
better prediction algorithm in machine learning
get to know about the seasons
fast and easy approach site
Predicting the gain percentage for a particular crop grown.
Predicting the drought condition along with the rainfall rate.
gather large amount of previous rainfall data
better evaluation metrics for checking the model

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

1. 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

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

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

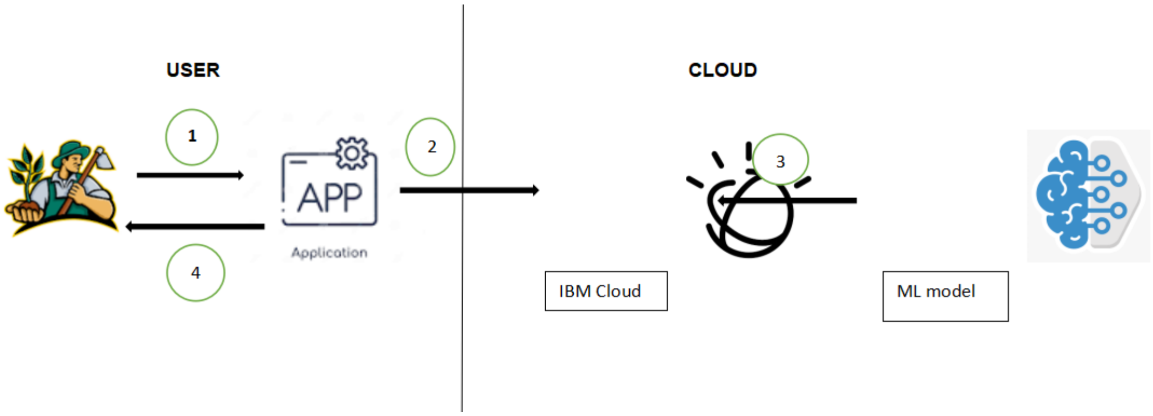
4.2 Non-Functional requirements

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

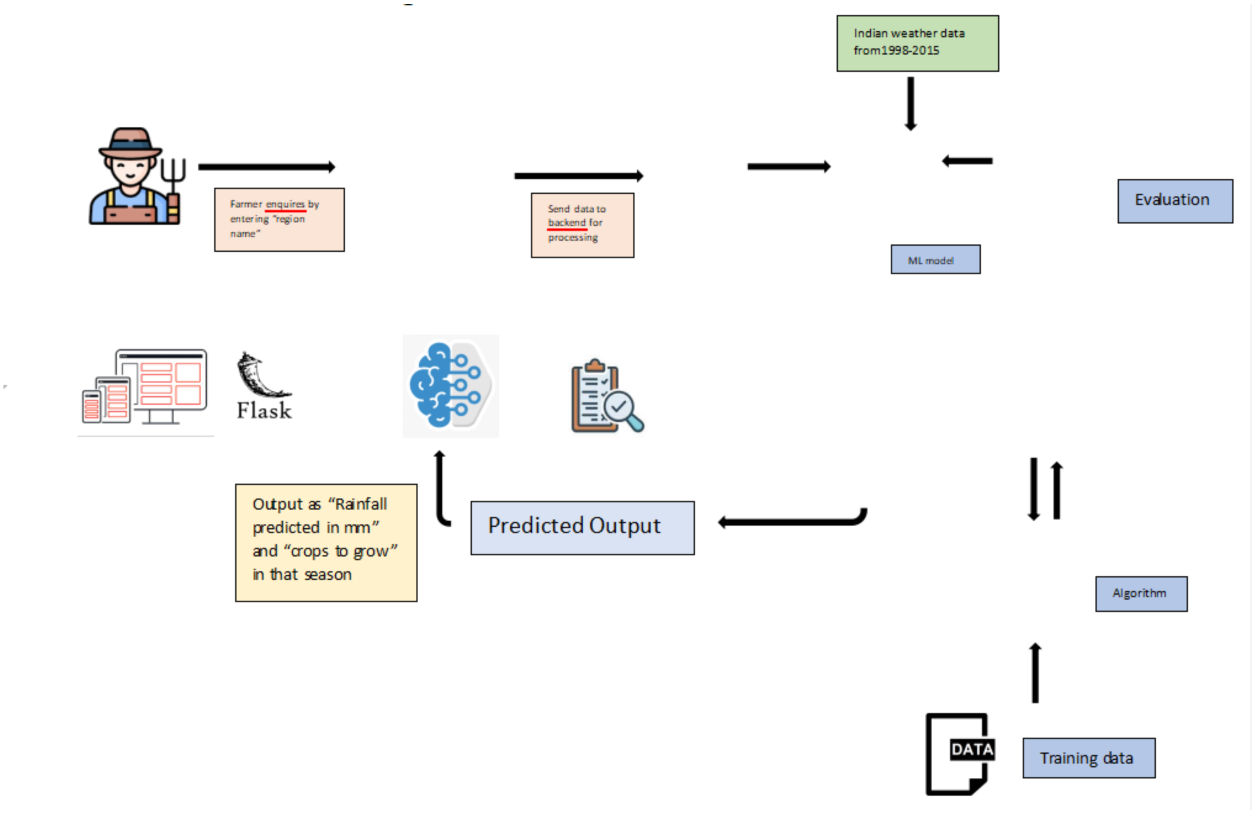
5.1 Data Flow Diagrams



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. User enters the region into the web application to predict the rainfall rateThe input data is sent to the cloud The Machine learning model deployed in the cloud predicts the rainfall And finally the predicted rainfall rate in mm and crops are suggested as output.Solution & Technical Architecture

5.2 Solution and technical Architecture

Solution Architecture


Indian weather data from1998-2015
Farmer enquires by entering "region name"
Send data to backend for processing
ML model
Evaluation
Flask
Output as "Rainfall predicted in mm" and "crops to grow" in that season
Predicted Output
Algorithm
DATA
Training data

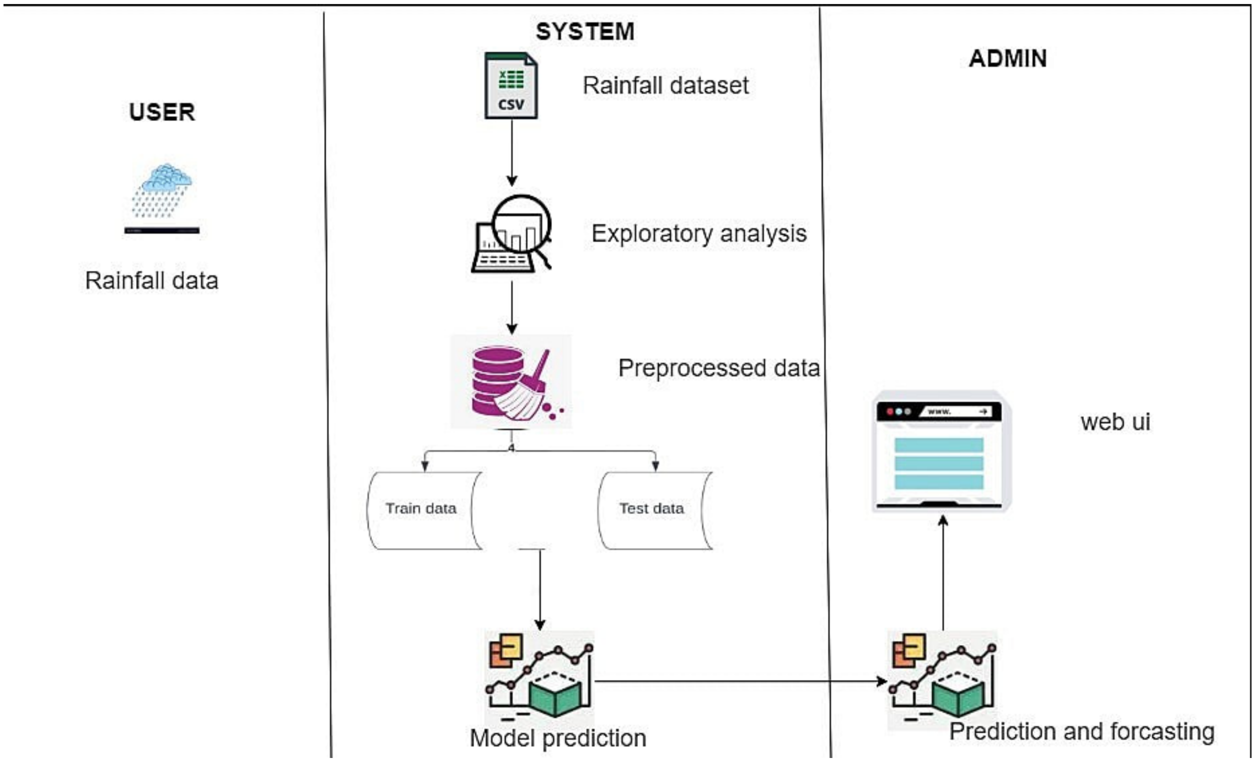


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user interacts with the application through a webUI and a chatbot	HTML, CSS, python,flask
2.	Application Logic-1	Logic for a Registration	Python
3.	Application Logic-2	Logic for a loginthe application	Python
4.	Application Logic-3	Integrating machine learning model and the webpage	Flask
5.	Database	Numeric data	MySQL

6.	File Storage	To storefiles such as prediction report	Local Filesystem
7.	External API-1	Allows developers access to critical forecasts, alerts, and observations, along with other weather data.	IBM WeatherAPI, etc.
8.	Machine Learning Model	Predictive modeling is a statistical technique using machine learning and datamining to predictand forecast likely future outcomes with the aid of historical and existing data	Predictive Model,etc.
9.	Infrastructure (Server)	Application Deployment on Local System Local Server Configuration: built-in flask web server	Flask web server

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask	Micro webframework written in Python
2.	Security Implementations	Basic HTTP authentication, Session based authentication, User Registration, Login Tracking	Flask security
3.	Scalable Architecture	Size is everything, and Flask’s statusas a microframework meansthat you can use it to grow a tech project such as a web app	

		incredibly quickly. .	
		Its simplicity of use and few dependenciesenable it to run smoothlyeven as it scales up and up.	
4.	Availability	Higher compatibility with latest technologies and allows customization	Flask
5.	Performance	Integrated supportfor unit testing. RESTful requestdispatching. Uses Jinja templating. Support for secure cookies(client side sessions) 100% WSGI 1.0 compliant.	Flask

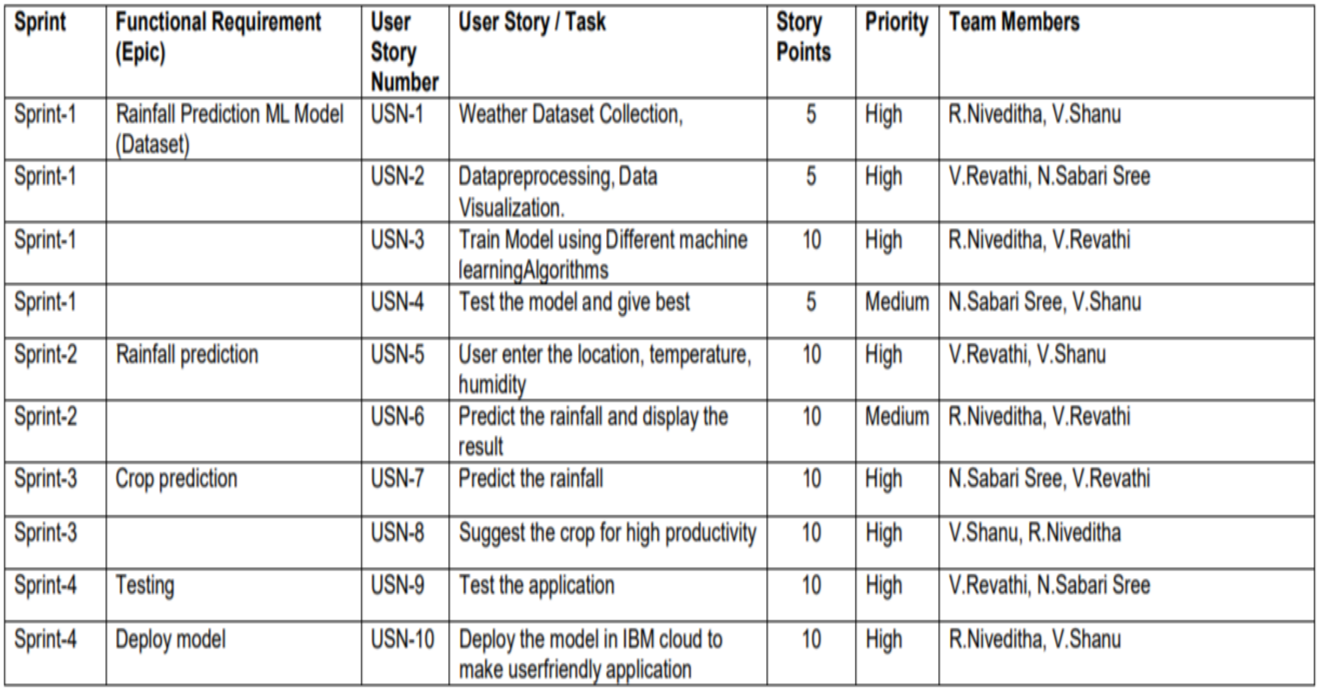
User Stories

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		As a customer,	I can view the		

	Crop	USN-2	With the predicted rainfall I can view the suggested crops for higher productivity	suggested crops with the predicted rainfall	High	Sprint-2

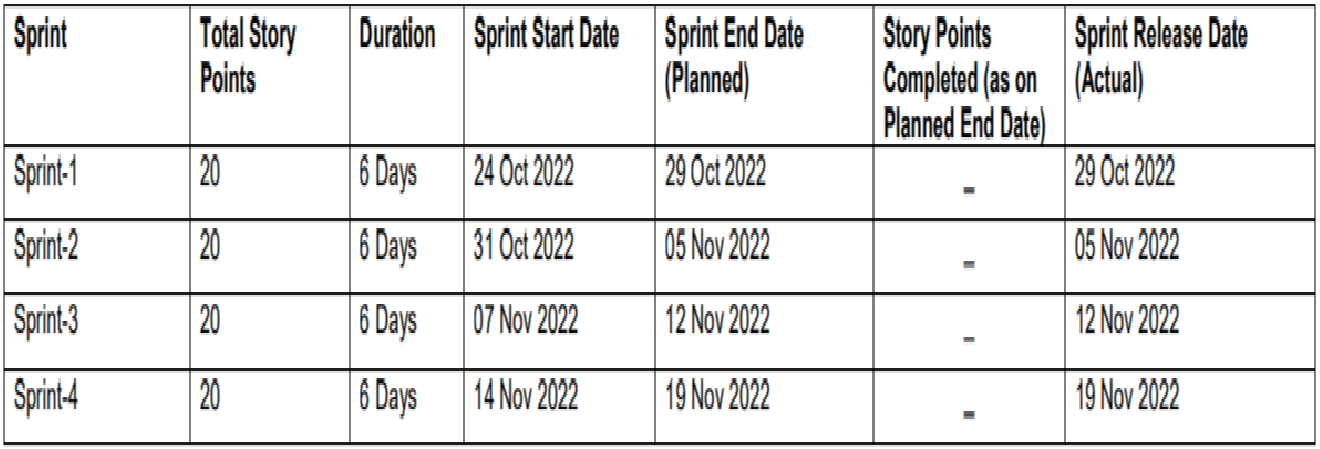
PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation



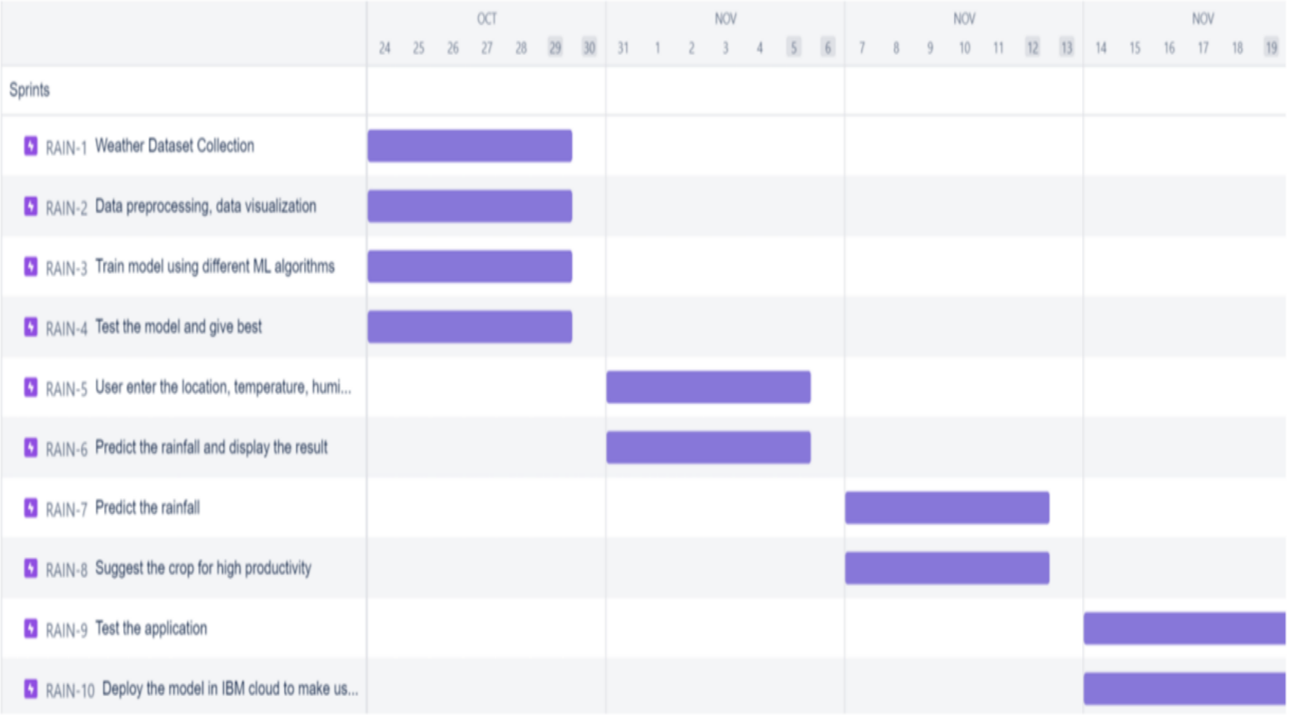
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Rainfall Prediction ML Model (Dataset)	USN-1	Weather Dataset Collection,	5	High	R.Niveditha, V.Shanu
Sprint-1		USN-2	Datapreprocessing, Data Visualization.	5	High	V.Revathi, N.Sabari Sree
Sprint-1		USN-3	Train Model using Different machine learningAlgorithms	10	High	R.Niveditha, V.Revathi
Sprint-1		USN-4	Test the model and give best	5	Medium	N.Sabari Sree, V.Shanu
Sprint-2	Rainfall prediction	USN-5	User enter the location, temperature, humidity	10	High	V.Revathi, V.Shanu
Sprint-2		USN-6	Predict the rainfall and display the result	10	Medium	R.Niveditha, V.Revathi
Sprint-3	Crop prediction	USN-7	Predict the rainfall	10	High	N.Sabari Sree, V.Revathi
Sprint-3		USN-8	Suggest the crop for high productivity	10	High	V.Shanu, R.Niveditha
Sprint-4	Testing	USN-9	Test the application	10	High	V.Revathi, N.Sabari Sree
Sprint-4	Deploy model	USN-10	Deploy the model in IBM cloud to make userfriendly application	10	High	R.Niveditha, V.Shanu

Sprint Delivery Schedule



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

Reports from JIRA



Sprints

RAIN-1 Weather Dataset Collection

RAIN-2 Data preprocessing, data visualization

RAIN-3 Train model using different ML algorithms

RAIN-4 Test the model and give best

RAIN-5 User enter the location, temperature, humi...

RAIN-6 Predict the rainfall and display the result

RAIN-7 Predict the rainfall

RAIN-8 Suggest the crop for high productivity

RAIN-9 Test the application

RAIN-10 Deploy the model in IBM cloud to make us...

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

Feature 1

For predicting the rainfall rate for 5 different states.

```
import numpy as np
import pandas as pd
import pickle
from sklearn import metrics

data = pd.read_csv(r"C:/Users/NIVEDITHA/Downloads/rainfall.csv")
# data.head()

data = data.fillna(data.mean())

group =
data.groupby('SUBDIVISION')['YEAR','JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','
SEP','OCT','NOV','DEC']
dt=group.get_group(('TAMIL NADU'))
# data.head()

df=dt.melt(['YEAR']).reset_index()
# df.head()

df= df[['YEAR','variable','value']].reset_index().sort_values(by=['YEAR','index'])
# df.head()

df.columns=['Index','Year','Month','Avg_Rainfall']
Month_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,
   'OCT':10,'NOV':11,'DEC':12}
df['Month']=df['Month'].map(Month_map)
# df.head(12)

df.drop(columns="Index",inplace=True)

X=np.asanyarray(df[['Month']]).astype('int')
y=np.asanyarray(df['Avg_Rainfall']).astype('int')
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=10)

from sklearn.ensemble import RandomForestRegressor
random_forest_model = RandomForestRegressor(max_depth=100,
max_features='sqrt', min_samples_leaf=4,
                    min_samples_split=10, n_estimators=800)
random_forest_model.fit(X_train, y_train)

#-------------------------------WEST BENGAL--------------------------------#


dt1=group.get_group(('WEST BENGAL'))
# data.head()

df1=dt1.melt(['YEAR']).reset_index()
# df.head()

df1= df1[['YEAR','variable','value']].reset_index().sort_values(by=['YEAR','index'])
# df.head()

df1.columns=['Index','Year','Month','Avg_Rainfall']
Month_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,
  'OCT':10,'NOV':11,'DEC':12}
df1['Month']=df1['Month'].map(Month_map)
# df.head(12)

df1.drop(columns="Index",inplace=True)

X1=np.asanyarray(df1[['Month']]).astype('int')
y1=np.asanyarray(df1['Avg_Rainfall']).astype('int')

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X1, y1, test_size=0.3,
random_state=10)

random_forest_model1 = RandomForestRegressor(max_depth=100,
max_features='sqrt', min_samples_leaf=4,
                    min_samples_split=10, n_estimators=800)
random_forest_model1.fit(X_train, y_train)
#y_predict = random_forest_model.predict(X_test)

#print('MAE:', metrics.mean_absolute_error(y_test,y_predict))
# print('MSE:', metrics.mean_squared_error(y_test, y_predict))

#print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_predict)))

#------------------------------------ORISSA------------------------------------

dt2=group.get_group(('ORISSA'))
# data.head()

df2=dt2.melt(['YEAR']).reset_index()
# df.head()

df2= df2[['YEAR','variable','value']].reset_index().sort_values(by=['YEAR','index'])
# df.head()

df2.columns=['Index','Year','Month','Avg_Rainfall']
Month_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,
  'OCT':10,'NOV':11,'DEC':12}
df2['Month']=df2['Month'].map(Month_map)
# df.head(12)

df2.drop(columns="Index",inplace=True)

X2=np.asanyarray(df2[['Month']]).astype('int')
y2=np.asanyarray(df2['Avg_Rainfall']).astype('int')

X_train, X_test, y_train, y_test = train_test_split(X2, y2, test_size=0.3,
random_state=10)

random_forest_model2 = RandomForestRegressor(max_depth=100,
max_features='sqrt', min_samples_leaf=4,
               min_samples_split=10, n_estimators=800)
random_forest_model2.fit(X_train, y_train)
#--------------------------------------------PUNJAB-----------------------------------------------

#group3=
data.groupby('SUBDIVISION')['YEAR','JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','
SEP','OCT','NOV','DEC']
dt3=group.get_group(("PUNJAB"))
# data.head()

df3=dt3.melt(['YEAR']).reset_index()
# df.head()

df3= df3[['YEAR','variable','value']].reset_index().sort_values(by=['YEAR','index'])
# df.head()

df3.columns=['Index','Year','Month','Avg_Rainfall']
Month_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,
  'OCT':10,'NOV':11,'DEC':12}
df3['Month']=df3['Month'].map(Month_map)
# df.head(12)

df3.drop(columns="Index",inplace=True)

X3=np.asanyarray(df3[['Month']]).astype('int')
y3=np.asanyarray(df3['Avg_Rainfall']).astype('int')

X_train, X_test, y_train, y_test = train_test_split(X3, y3, test_size=0.3,
random_state=10)


random_forest_model3 = RandomForestRegressor(max_depth=100,
max_features='sqrt', min_samples_leaf=4,
               min_samples_split=10, n_estimators=800)
random_forest_model3.fit(X_train, y_train)
#---------------------------------------------------UTTARAKHAND---------------------
```

```
dt4=group.get_group((‘UTTARAKHAND'))
# data.head()

df4=dt4.melt(['YEAR']).reset_index()
# df.head()

df4= df4[['YEAR','variable','value']].reset_index().sort_values(by=['YEAR','index'])
# df.head()

df4.columns=['Index','Year','Month','Avg_Rainfall']
Month_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,
   'OCT':10,'NOV':11,'DEC':12}
df4['Month']=df4['Month'].map(Month_map)
# df.head(12)

df4.drop(columns="Index",inplace=True)

X4=np.asanyarray(df4[['Month']]).astype(‘int')
y4=np.asanyarray(df4['Avg_Rainfall']).astype(‘int')

X_train, X_test, y_train, y_test = train_test_split(X4,y4, test_size=0.3, random_state=10)

random_forest_model4 = RandomForestRegressor(max_depth=100,
max_features='sqrt', min_samples_leaf=4,
                      min_samples_split=10, n_estimators=800)
random_forest_model4.fit(X_train, y_train)
#-----------------------------------JAMMU & KASHMIR------------------------------------

dt5=group.get_group((‘JAMMU & KASHMIR'))
# data.head()

df5=dt5.melt(['YEAR']).reset_index()
# df.head()

df5= df5[['YEAR','variable','value']].reset_index().sort_values(by=['YEAR','index'])
# df.head()
```

```
df5.columns=['Index','Year','Month','Avg_Rainfall']
Month_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,
   'OCT':10,'NOV':11,'DEC':12}
df5['Month']=df5['Month'].map(Month_map)
# df.head(12)

df5.drop(columns="Index",inplace=True)

X5=np.asanyarray(df5[['Month']]).astype('int')
y5=np.asanyarray(df5['Avg_Rainfall']).astype('int')

X_train, X_test, y_train, y_test = train_test_split(X5, y5, test_size=0.3,
random_state=10)


random_forest_model5 = RandomForestRegressor(max_depth=100,
max_features='sqrt', min_samples_leaf=4,
                     min_samples_split=10, n_estimators=800)
random_forest_model5.fit(X_train, y_train)

#--------------------------------------------------------------------------------
file = open("model.pkl","wb")
file1=open("model1.pkl","wb")
pickle.dump(random_forest_model,file)
pickle.dump(random_forest_model1,file1)
file.close()
file1.close()
file2 = open("model2.pkl","wb")
file3=open("model3.pkl","wb")
pickle.dump(random_forest_model2,file2)
pickle.dump(random_forest_model3,file3)
file2.close()
file3.close()
file4 = open("model4.pkl","wb")
file5=open("model5.pkl","wb")
pickle.dump(random_forest_model4,file4)
pickle.dump(random_forest_model5,file5)
```

```
file4.close()
file5.close()
# print(y_predict)
```

Feature 2

For suggesting the better yielding crop with the rainfall predicted for a particular state.

```
from flask import render_template,Flask,request
import pickle

from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
dt = pd.read_csv(r"C:/Users/NIVEDITHA/Downloads/Crop_recommendation.csv")

# Create feature and target arrays
train=dt['rainfall']
target=dt['label']
train=np.array(train)
target=np.array(target)
# Split into training and test set
X_train, X_test, y_train, y_test = train_test_split(
            train,target, test_size = 0.3, random_state=1)

knn = GaussianNB()

knn.fit(X_train.reshape(-1,1), y_train)
pred=knn.predict(X_test.reshape(-1,1))
print(accuracy_score(y_test,pred))
```

```
appl=Flask(__name__)
file=open("model.pkl","rb")
file1=open("model1.pkl","rb")
file2=open("model2.pkl","rb")
file3=open("model3.pkl","rb")
file4=open("model4.pkl","rb")
file5=open("model5.pkl","rb")
random_Forest=pickle.load(file)
file.close()
random_Forest1=pickle.load(file1)
file1.close()
random_Forest2=pickle.load(file2)
file2.close()
random_Forest3=pickle.load(file3)
file3.close()
random_Forest4=pickle.load(file4)
file4.close()
random_Forest5=pickle.load(file5)
file5.close()

#random_Forest=pickle.load(file)
#file.close()



@appl.route("/", methods=["GET","POST"])
def home():
    if request.method=="POST":
        myDict = request.form
        Month = int(myDict["Month"])
        state= (myDict["state"])
        pred = [Month]
        #stateCall(state)
        #res=random_Forest.predict([pred])[0]
        if(state=="TAMILNADU"):
            res=random_Forest.predict([pred])[0]
        elif state=="WEST BENGAL":
            res=random_Forest1.predict([pred])[0]
        elif(state=="ORISSA"):
            res=random_Forest2.predict([pred])[0]
        elif(state=="PUNJAB"):
            res=random_Forest3.predict([pred])[0]
        elif(state=="UTTARAKHAND"):
            res=random_Forest4.predict([pred])[0]
        else:
            res=random_Forest5.predict([pred])[0]
        res=round(res,2)
        ans=knn.predict([[res]])[0]
        return render_template('result.html',Month=Month,state=state,res=res,ans=ans)
    return render_template('index.html')

if __name__ == "__main__":
    appl.run(debug=True)
```

8. TESTING

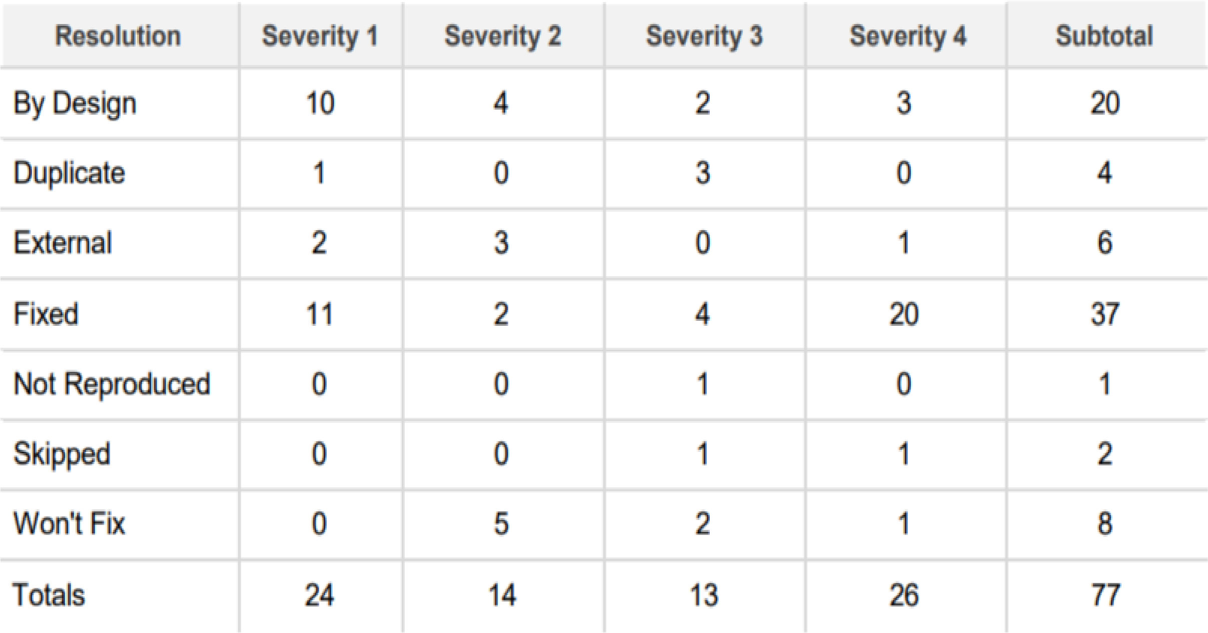
8.1 Test Cases

INPUT	PREDICTED OUTPUT
MONTH :12 STATE:WEST BENGAL	The Rainfall rate is 6.42 mm Can plant papaya for good yield
MONTH:9 STATE:JAMMU & KASHMIR	The Rainfall rate is 85.23 mm Can plant chickpea for good yield
MONTH:2 STATE:TAMILNADU	The Rainfall rate is 11.17 mm Can plant papaya for good yield

MONTH:1 STATE:ASSAM	The Rainfall rate is 100.92 mm Can plant banana for good yield

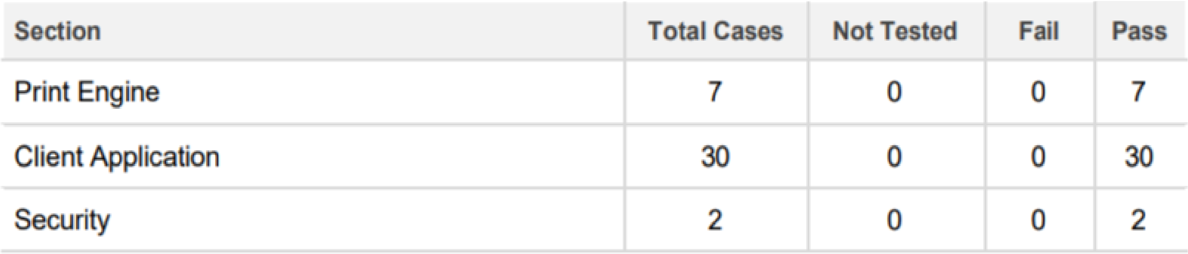
8.2 User Acceptance Testing

Defect Analysis

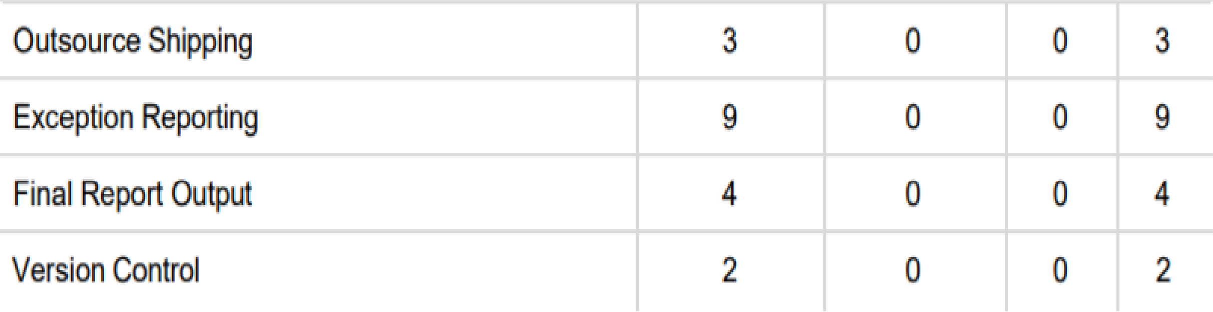


Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

TestCase analysis



Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	30	0	0	30
Security	2	0	0	2



Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9.RESULTS

9.1.Performance Metrics

```
y_test_predict=rfmodel.predict(X_test)
print("-------Test Data--------")
print('MAE:', metrics.mean_absolute_error(y_test, y_test_predict))
print('MSE:', metrics.mean_squared_error(y_test, y_test_predict))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_test_predict)))
```

```
-------TestData--------
MAE:82.07252495976634
MSE:20286.85437242721
RMSE: 142.4319289079075
```

10.ADVANTAGES

1. Predicting the rainfall for a month can help them plan for the many day-to-day decisions. These decisions include crop irrigation, time to fertilize, and what days are suitable for working in the field. The decisions that farmers make will result in a profitable crop or failure.
2. Crop suggested can help farmers to acheive a better yield and high business-profit.
3. Weather forecast helps in controlling the pests and other crop diseases to spread over the field. Weather factors can influence crop-destroying pests.

DISADVANTAGES

1. Only predicted for certain states and not for all ,due to unavailability of data's.
2. Provides information only for a month and not for individual days .

3. Provides only what crop to plant for the particular month in a state.

11.CONCLUSION

Sometimes farmers in India occur loss due to false predictions of weather. Now that the technology is developed and special weather forecasting mechanisms are available, the farmers can get all the updates are on a smartphone. Education towards that is, of course, an important thing but most of the farmer population at this stage knows the basics which make it easy for them to use the features. Most field crops are dependent solely upon weather to provide life-sustaining water and energy. Livestock are also dependent upon weather for their comfort and food supplies.Our machine learning model can thus, help all the farmers and agriculturalists to predict or forecast the rainfall for a particular month with the Rainfall datset and further gets to know about the crop to grow for getting high yield and good profit with the crop recommendation dataset.

12.FUTURE SCOPE

In future ,the forecasting of weather and precise details about the crops can be recommended.Unlike our weather prediction model, forecast systems that use machine learning are not constrained by the physical laws that govern the atmosphere. So it’s possible that they could produce unrealistic results – for example, forecasting temperature extremes beyond the bounds of nature. And it is unclear how they will perform during highly unusual or unprecedented weather phenomena. So techinques can be used to improve the accuracy by using large amount of weather data.

13.APPENDIX

Source Code

WEBPAGE CREATION CODE

index.html //The home page to get input from end-users

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
<linkrel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
```

```
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <title>Rainfall-Prediction</title>
</head>
<style>
body{
 background-image: url('https://i.gifer.com/A4XX.gif') ;
 background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
  }
  </style>
  <body>
  <br><br><br><br>
 <div class="container">
    <h1 class="text-center m-3 badge-dark text-wrap">
        Rainfall Prediction for India
      </h1>
        <div class="card container" style="width: 65%; ">

            <div class="card-body">

                <form action="/" method="post">
                    <div class="form-group">
                      <label for="formGroupExampleInput1">MONTH</label>
                      <input
                        type="text"
                        class="form-control"
                        id="formGroupExampleInput1"
                        name="Month"
                        required
                      />
                    </div>
                    <div class="form-group">
                      <label for="formGroupExampleInput2">STATE</label>
                      <input
                        type="text"
                        class="form-control"
                        id="formGroupExampleInput2"
                        name="state"
                        required
                      />
                    </div>
                        <center><button type="submit" class="btn btn-dark">CHECK</button></center>
                  </form>
            </div>
        </div>
    </div>

</body>
</html>
```

result.html //The output is displayed in this page

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <title>Rainfall-Prediction</title>
</head>
<style>
body{
 background-image: url('https://i.gifer.com/A4XX.gif') ;
 background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
  }
  </style>
<body >
<br><br><br><br>
```

```
  <div class="container" >
  <h1 class="text-center m-3 badge-dark text-wrap">Rainfall Prediction</h1>
    <div class="card container" style="width: 50%;">
      <div class="card-body" >
          <form action="/" method="post">
            <div class="form-group">
             <label for="formGroupExampleInput1">Month</label>
             <input
              type="text"
              class="form-control"
              id="formGroupExampleInput1"
              name="Month"
              placeholder="{{Month}}"
              required
             />
            </div>
            <div class="form-group">
             <label for="formGroupExampleInput2">state</label>
             <input
              type="text"
              class="form-control"
              id="formGroupExampleInput2"
              name="Year"
              placeholder="{{state}}"
              required
             />
            </div>
            <h2 class="text-center  text-wrap" >The Rainfall rate is {{res}} mm </h2>
            <h2 class="text-center  text-wrap">Can plant {{ans}} for good yield</h2>

           </form>
          </div>

                      <center><a href="/"><button type="submit" class="btn btn-dark">Back</button></a></center>
          </div>
    </div>

</body>
</html>
```

GitHub link:
https://github.com/IBM-EPBL/IBM-Project-2784-1658482679.git
Project Demo Link:
https://drive.google.com/file/d/1eEb7PMwI7nGDLYJSqb4r18khoa11uti7/view?usp=share_link