Professional Readiness For Innovative, Employability And Entrepreneurship
HX 8001

PROJECT TITLE:
Exploratory Analysis of Rainfall Data in India for Agriculture
DOMAIN: Applied Data Science

TEAM MEMBERS:
V.LOGESH (Team Leader)
P.MANOJKUMAR
S.ASHWIN
T.KEERTHIDHARAN

TEAM ID: PNT2022TMID28004
DEPARTMENT: INFORMATION TECHNOLOGY
COLLEGE: MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE, CHENNAI-600024.

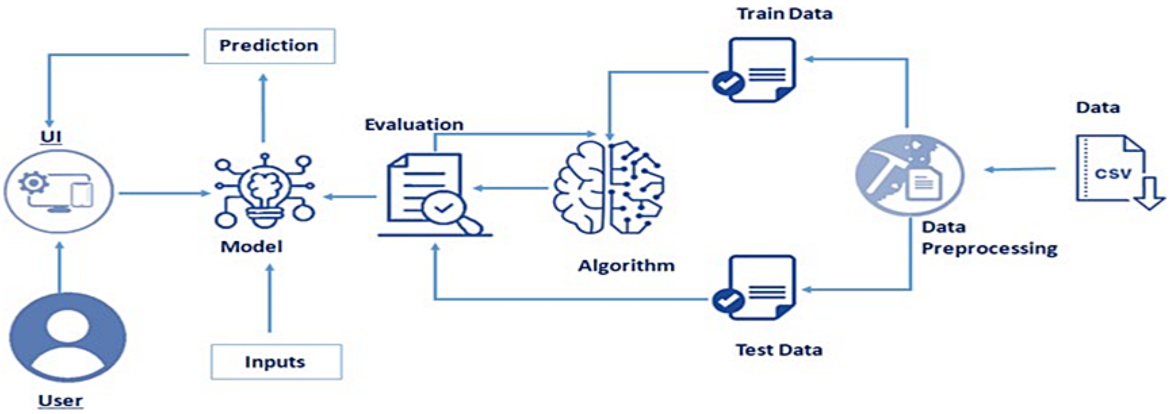
INSTITUTION MENTOR: K.GAYATHRI

INDUSTRIAL MENTOR: LALITHA GAYATHRI

1.INTRODUCTION

1.1 Project Overview

Rainfall has been a majorconcern these days. Weather conditions have been changingfortime being. Rainfallforecasting is importantotherwise, it may lead to many disasters. Irregularheavy rainfallmay lead to the destruction of crops, heavy floods that can cause harm to humanlife. It is important to exactly determine the rainfall for effective use of water resources, cropproductivity, 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 machinelearning techniques andtheir reliability to predict rainfall by analyzing the weather data.



1.2 Purpose

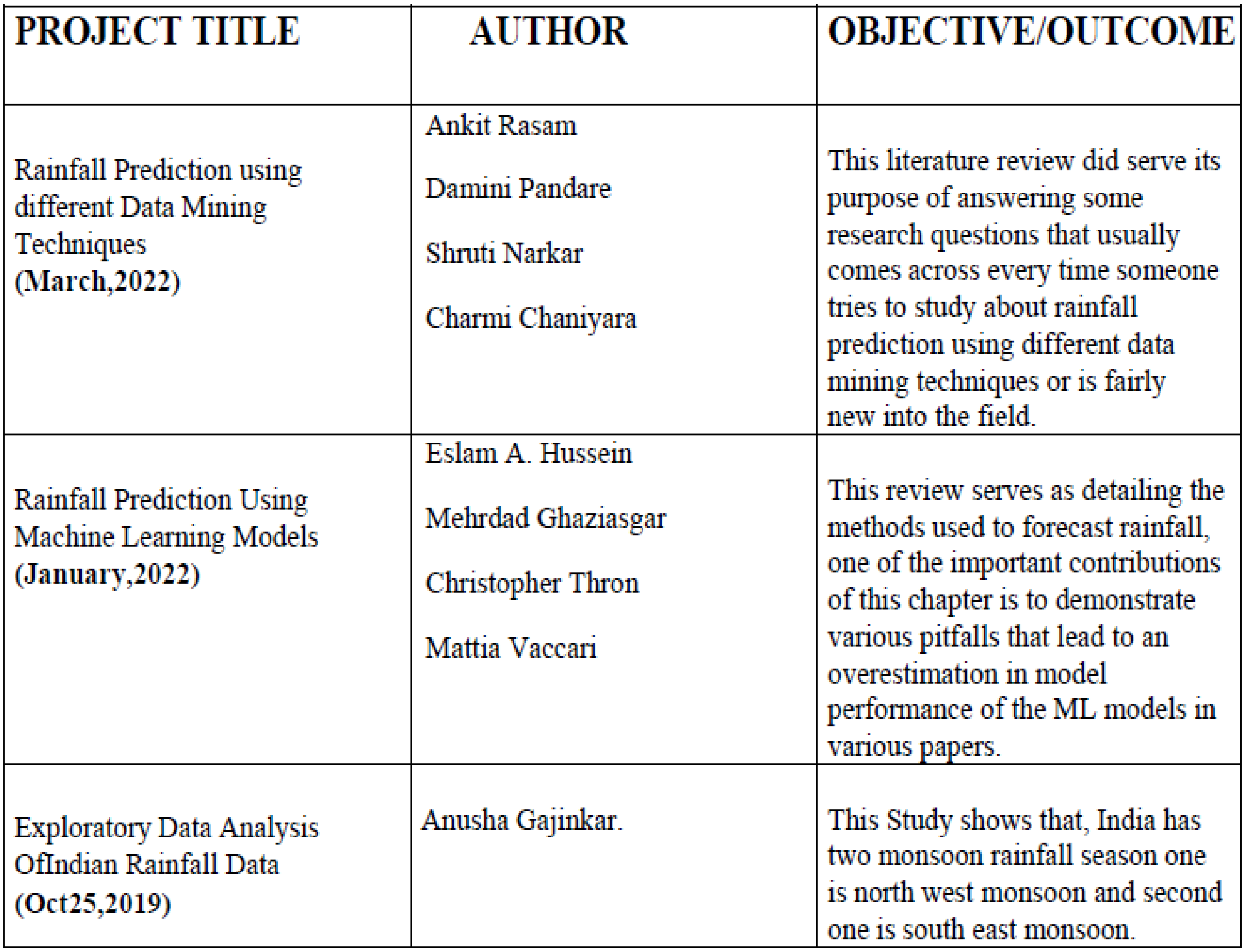
The main purpose of our project is to detect the rainfall detection in Agricultural area inIndia with a help of machine learning. To design a disaster management system by forecasting a flood event to control flood risk by recommending an evacuation area fromflood hazard areas which ultimately helps to manage theenvironment and water resource system.This also serves a purpose of the Early warning system by training a model and selecting the best prediction algorithm among the classifiers.The occurrence of flash floods can cause catastrophic damage to the society. They first mainly affect the people living nearto the riverbeds.Evacuating them from the hazard areas and providing them the shelter they needed.With the irregular change in climate patterns, it's been difficult to predict the occurrence of floods using traditional methods leading to massive destruction. Thus to copewith flash floods and to handle critical situations new methodologies are invented to overcome such difficulties.

2. LITERATURE SURVEY

2.1 Existing problem

A bad rainfall prediction can affect the agriculture mostly framers as their whole crop is depend on the rainfall and agriculture is always an important part of every economy. So, makingan accurate prediction of the rainfall somewhat good.Now climate change is the biggest issue all over the world. Peoples are working on to detect the patterns in climate change as it affects the economy in production to infrastructure. Soas in rainfall also making prediction of rainfall is a challenging task with a good accuracy rate. Making prediction on rainfall cannot be done by the traditional way, so scientist is using machine learning and deep learning to find out the pattern for rainfall prediction.

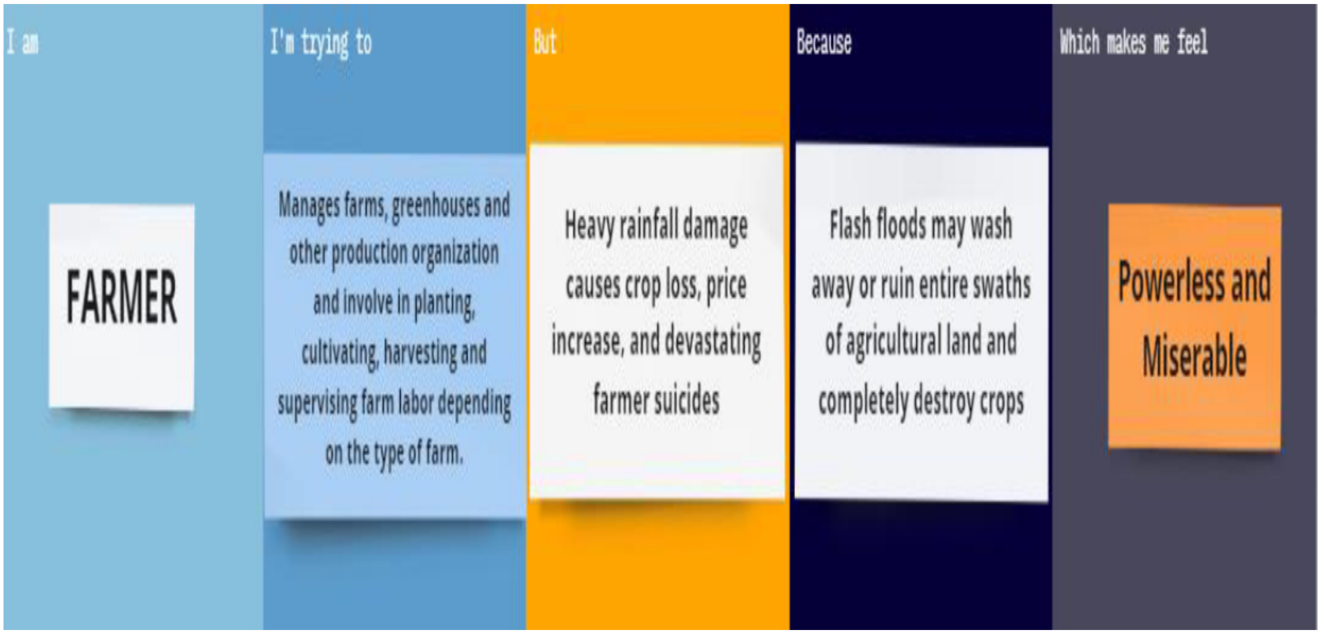
2.2 Refrences



PROJECT TITLE	AUTHOR	OBJECTIVE/OUTCOME
Rainfall Prediction using different Data Mining Techniques **(March,2022)**	Ankit Rasam Damini Pandare Shruti Narkar Charmi Chaniyara	This literature review did serve its purpose of answering some research questions that usually comes across every time someone tries to study about rainfall prediction using different data mining techniques or is fairly new into the field.
Rainfall Prediction Using Machine Learning Models **(January,2022)**	Eslam A. Hussein Mehrdad Ghaziasgar Christopher Thron Mattia Vaccari	This review serves as detailing the methods used to forecast rainfall, one of the important contributions of this chapter is to demonstrate various pitfalls that lead to an overestimation in model performance of the ML models in various papers.
Exploratory Data Analysis OfIndian Rainfall Data **(Oct25,2019)**	Anusha Gajinkar.	This Study shows that, India has two monsoon rainfall season one is north west monsoon and second one is south east monsoon.

2.3 Problem Statement Definition

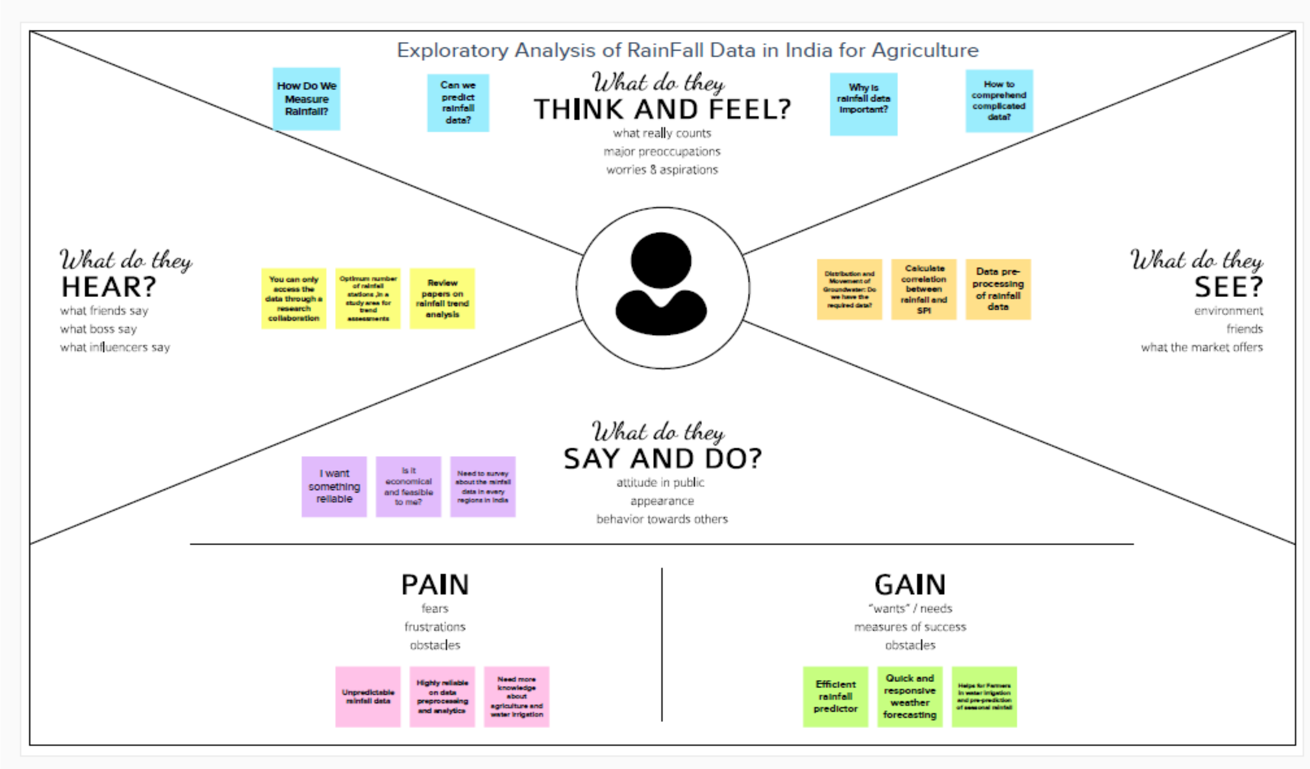
Weather conditions changes then and often. This can lead to Severe threats to all the living beings including human beings. So predicting weather, especially Irregular heavy rainfall, Droughts can cause huge economic losses. This also decreases crop productivity and may lead into Food shortage. Predicting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



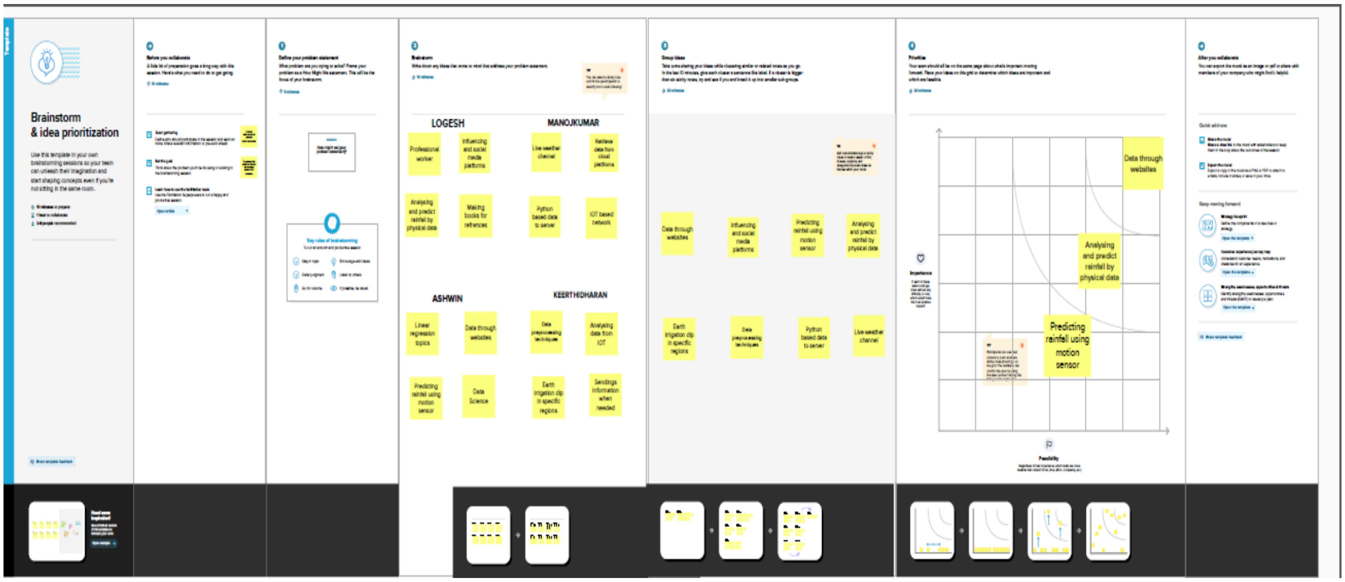
I am	I'm trying to	But	Because	Which makes me feel
FARMER	Manages farms, greenhouses and other production organization and involve in planting, cultivating, harvesting and supervising farm labor depending on the type of farm.	Heavy rainfall damage causes crop loss, price increase, and devastating farmer suicides	Flash floods may wash away or ruin entire swaths of agricultural land and completely destroy crops	Powerless and Miserable

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas


Exploratory Analysis of RainFall Data in India for Agriculture
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
How Do We Measure Rainfall?
Can we predict rainfall data?
Why is rainfall data important?
How to comprehend complicated data?
What do they
HEAR?
what friends say
what boss say
what influencers say
You can only access the data through a research collaboration
Optimum number of rainfall stations ,in a study area for trend assessments
Review papers on rainfall trend analysis
What do they
SEE?
environment
friends
what the market offers
Distribution and Movement of Groundwater: Do we have the required data?
Calculate correlation between rainfall and SPI
Data pre-processing of rainfall data
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
I want something reliable
Is it economical and feasible to me?
Need to survey about the rainfall data in every regions in India
PAIN
fears
frustrations
obstacles
Unpredictable rainfall data
Highly reliable on data preprocessing and analytics
Need more knowledge about agriculture and water irrigation
GAIN
"wants" / needs
measures of success
obstacles
Efficient rainfall predictor
Quick and responsive weather forecasting
Helps for Farmers in water irrigation and pre-prediction of seasonal rainfall

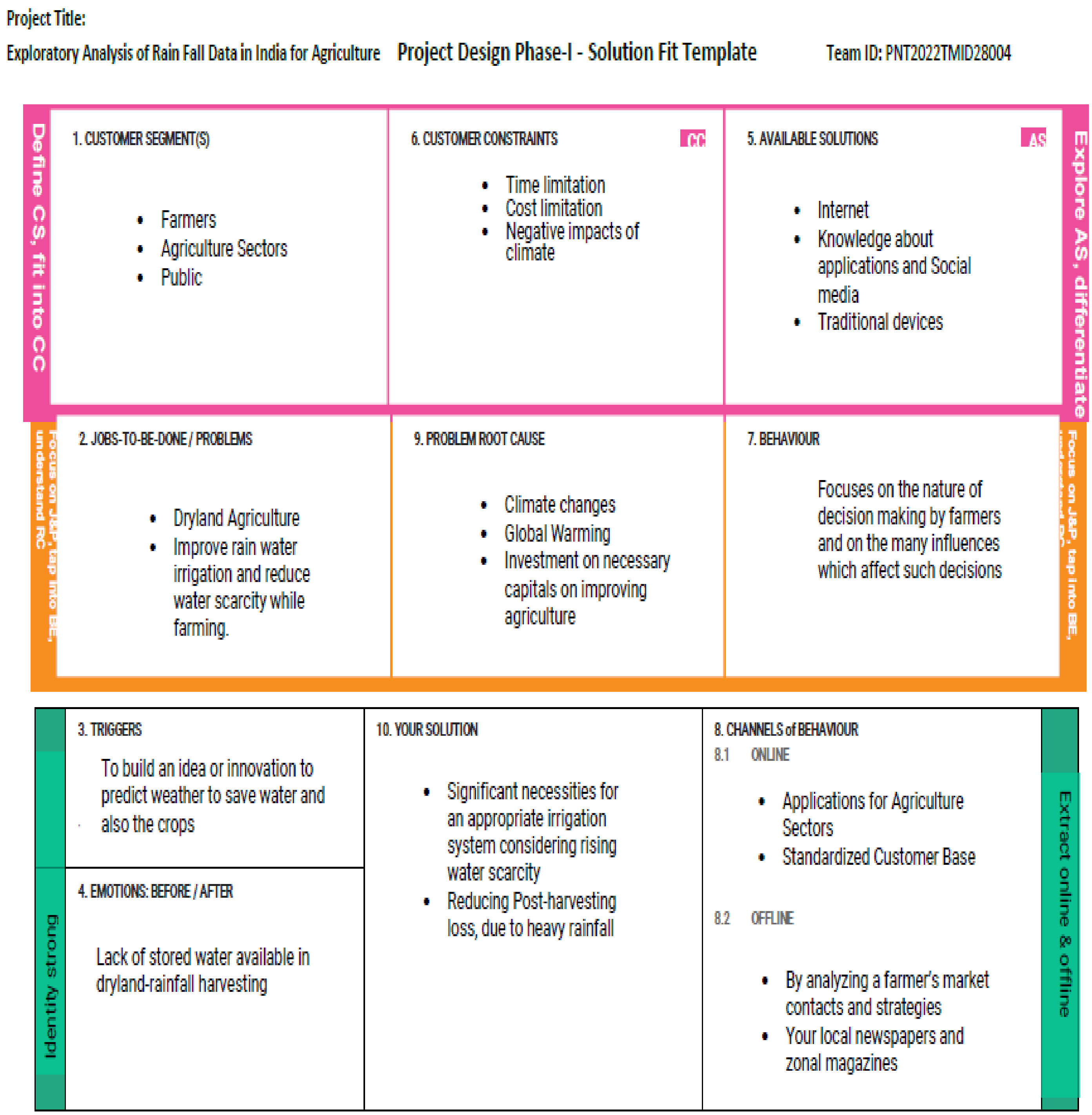
3.2 Ideation and Brainstorming


Brainstorm
& idea prioritization
LOGESH
Professional worker
Influencing and social media platforms
Analysing and predict rainfall by physical data
Making books for refrences
MANOJKUMAR
Live weather channel
Retrieve data from cloud platforms
Python based data to server
IOT based network
ASHWIN
Linear regression topics
Data through websites
Predicting rainfall using motion sensor
Data Science
KEERTHIDHARAN
Data preprocessing techniques
Analysing data from IOT
Earth irrigation dip in specific regions
Sendings information when needed
Data through websites
Influencing and social media platforms
Predicting rainfall using motion sensor
Analysing and predict rainfall by physical data
Earth irrigation dip in specific regions
Data preprocessing techniques
Python based data to server
Live weather channel
Importance
Feasibility
Data through websites
Analysing and predict rainfall by physical data
Predicting rainfall using motion sensor

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Heavy Rainfall may cause huge threat to all living beings, especially in the field of Agriculture. Droughts could do the same too. It may destroy the crops and cause huge loss to Farmers and dependent field workers. Predicting Rainfall is a major task in bothsummer and Rainy season.
2.	Idea / Solution description	Analysing the previous 10 years data's cangive us a rough idea about Rainfall pattern. UsingData Science, we could solve this and predict the Rainfall up tosome good extent.
3.	Novelty / Uniqueness	AI, IOT and so many other fields may require different sensors. We are not going to use any kind of equipment. Time of prediction is very less and easywith affordable cost.
4.	Social Impact/ Customer Satisfaction	Farmers (theysave crops and money), Vegetable sellers(they knows about vegetable stocks and its emergency)
5.	Business Model (Revenue Model)	This could cost really low as a person should develop knowledge in Datascience and probably a gadget to developthis.However, deploying as an App attached with otherfacilities may costan extra charge.
6.	Scalability of the Solution	Farmers, Vegetable sellers, Citizens

3.4 Problem Solution Fit



Project Title:

Exploratory Analysis of Rain Fall Data in India for Agriculture

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID28004

1. CUSTOMER SEGMENT(S)

- Farmers
- Agriculture Sectors
- Public

6. CUSTOMER CONSTRAINTS

- Time limitation
- Cost limitation
- Negative impacts of climate

5. AVAILABLE SOLUTIONS

- Internet
- Knowledge about applications and Social media
- Traditional devices

2. JOBS-TO-BE-DONE / PROBLEMS

- Dryland Agriculture
- Improve rain water irrigation and reduce water scarcity while farming.

9. PROBLEM ROOT CAUSE

- Climate changes
- Global Warming
- Investment on necessary capitals on improving agriculture

7. BEHAVIOUR

Focuses on the nature of decision making by farmers and on the many influences which affect such decisions

3. TRIGGERS

To build an idea or innovation to predict weather to save water and also the crops

4. EMOTIONS: BEFORE / AFTER

Lack of stored water available in dryland-rainfall harvesting

10. YOUR SOLUTION

- Significant necessities for an appropriate irrigation system considering rising water scarcity
- Reducing Post-harvesting loss, due to heavy rainfall

8. CHANNELS of BEHAVIOUR

8.1 ONLINE

- Applications for Agriculture Sectors
- Standardized Customer Base

8.2 OFFLINE

- By analyzing a farmer's market contacts and strategies
- Your local newspapers and zonal magazines

Define CS, fit into CC
CC
AS
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC
Identify strong
Extract online & offline

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	User Registration	Registration through Form Registration through Gmail Registration throughLinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Reliability	Theprediction will be provided by the systemerror-free.
FR-4	Performance	The expected output will be produces immediatelyto the user withoutmuch delay.
NFR-5	Availability	The system wouldbe available 24/7
NFR-6	Scalability	Thesystem would be available on web applicationand any user can login and use it without anydisruptions.

4.2 Non-Functional requirements

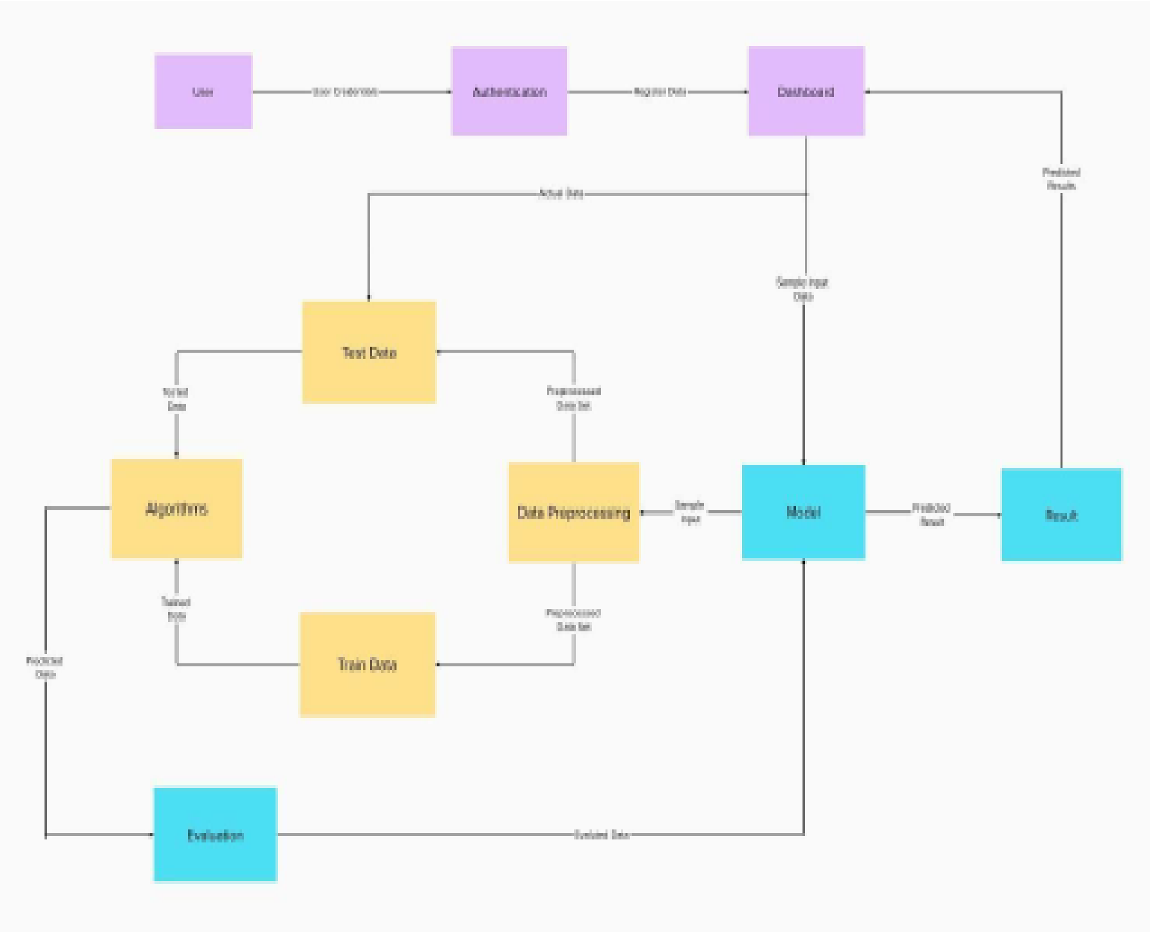
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Can be used anywhere(remote villages to metropolitan cities), anybody (kids to old age)
NFR-2	**Security**	Security is given over the model, so the user can usethis with full trust. However, there are no personaldetails required to use this.
NFR-3	**Reliability**	Goodconnectivity and a supporting devicecanprovide goodresults upto an extent.
NFR-4	**Performance**	Thismodel can give a high accuracy prediction.

NFR-5	**Availability**	Anyperson can use this and this is an open-sourcemodel.

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM:

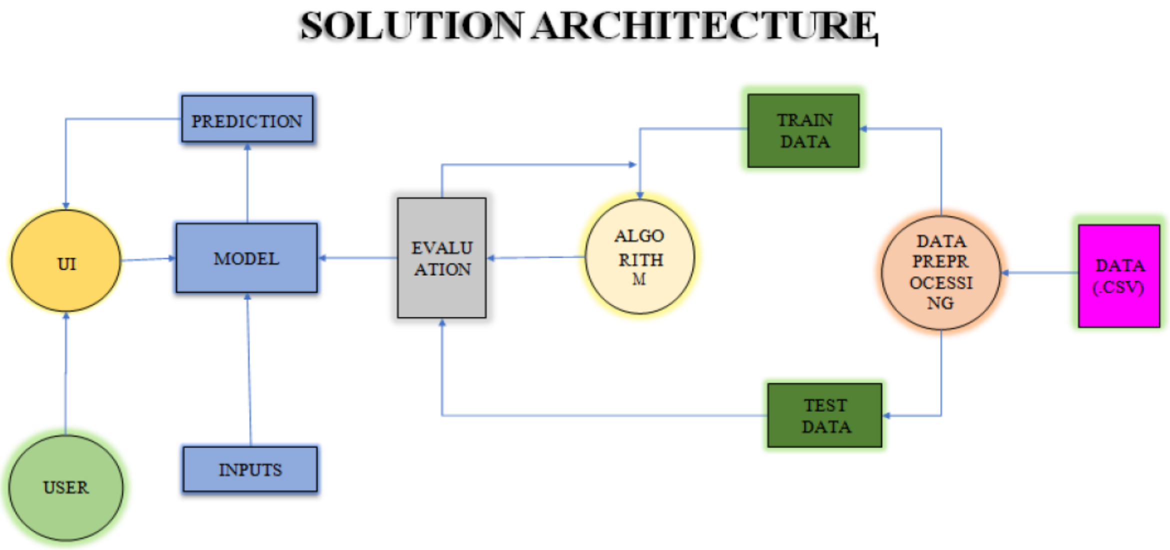


5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Solution Architecture:

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

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



Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2Technology architecture associates application components from application architecturewith technology components representing software and hardware components. Its components are generally acquired in the marketplace and can be assembled and configured to constitute the enterprise’s technological infrastructure.

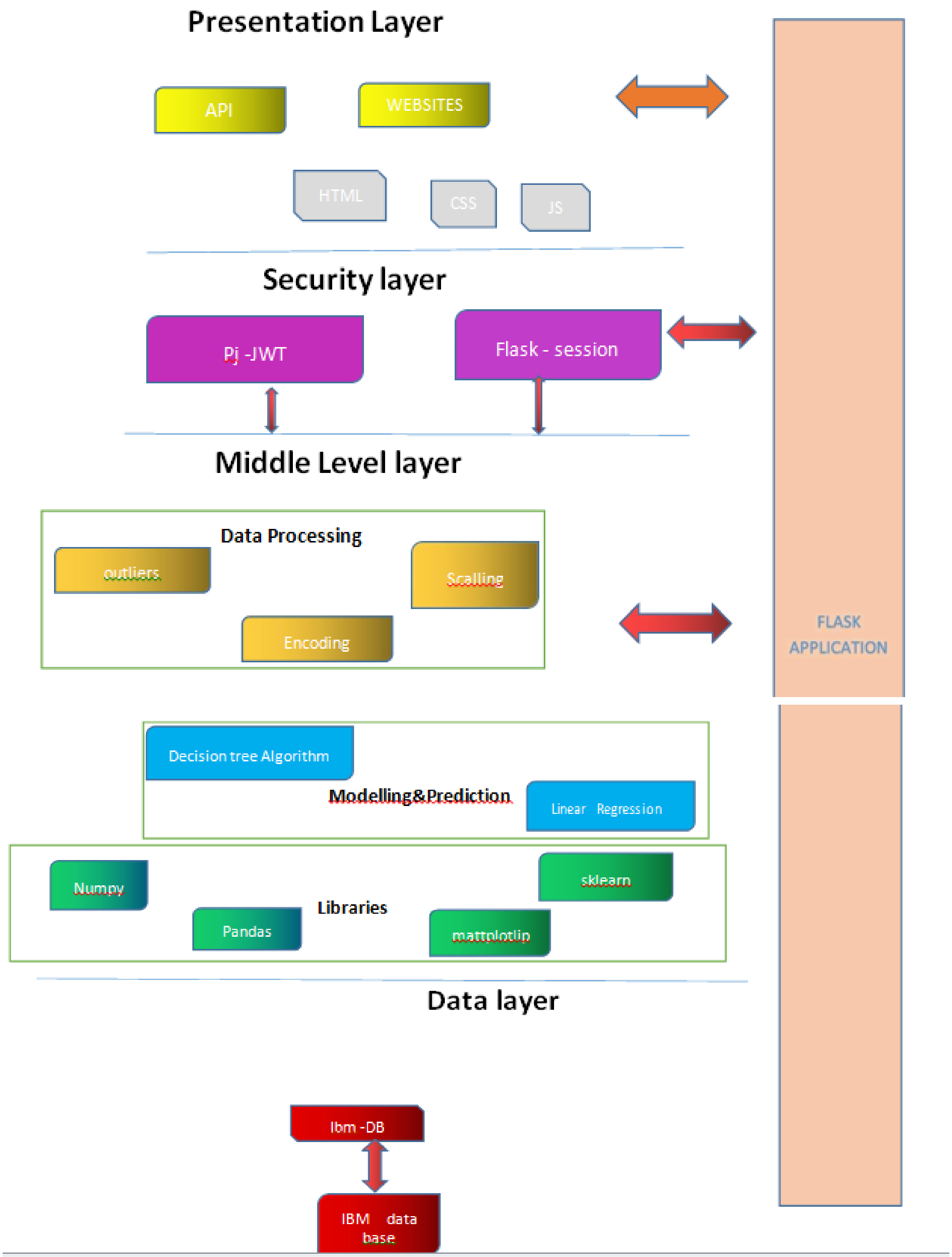
Table-1 : Components & Technologies:

S.NO	COMPONENTS	DESCRIPTION	TECHNOLOGY
1.	User interface	To anticipate thedata for rainfall,the user engageswith the prediction modelviaa website.	HTML, CSS, JavaScript
2.	Cloud Database	The model receives information froman IBM cloud database.	IBM Cloud DB, ibm_db(pythonpackage)
3.	APL	used to expandservice to additional applications	Flask Application
4.	JWT&Sessions	Is employed to extend serviceto more applications	PyJWT**,** Flask Application
5.	Machine Learning Model	This model wascreated to forecast rainfallusing machine learning	Sklearn, Algorithms - DT & MLR

Table-2:Application Characteristics:

S.NO	CHARCTERITICS	DESCRIPTION	TECHNOLOGY
1.	Open-Source Frameworks	Backend Framework, CSSStyling framework, RelationalData base	PyJWT, Flask, IBM Cloud DB
2.	Security Implementations	Request authentication using JWT Tokens	HS-256, Encryptions, SSL Certs
3.	Scalable Architecture	Support for Multiple Sampleprediction using Excel File	File Pandas, Numpy
4.	Availability	Availability is increased by Distributed Servers in CloudVPS	IBM Cloud Hosting
5.	Performance	The applicationis expected to handle multiple predictions per second	Load Balancers, Distributed ServerS

Technical Architecture :



5.3 User Stories

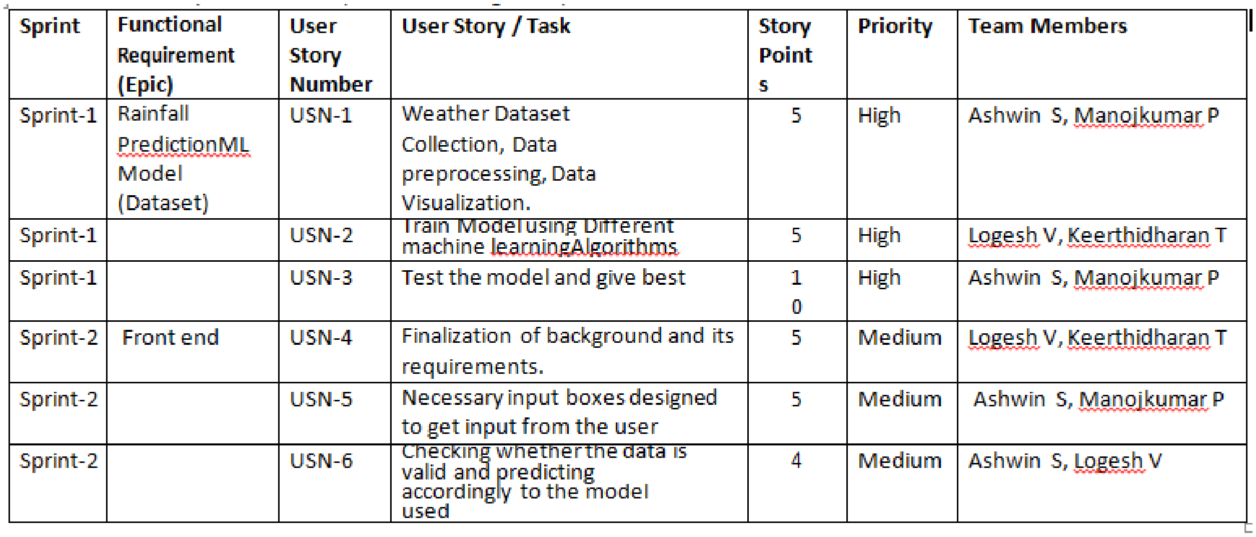
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration/ Login	USN-1	As a user, I can register or login to create a dashboard for my processing	I can access my account/ dashboard	High	Sprint-1
	Dashboard	USN-2	Once I enter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

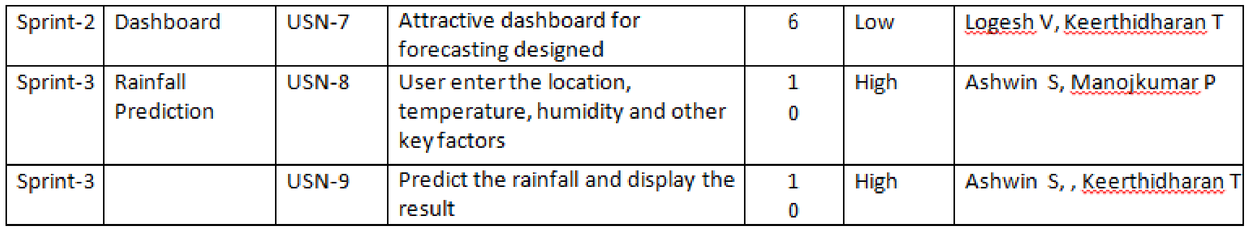
6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Sprint Delivery Schedule

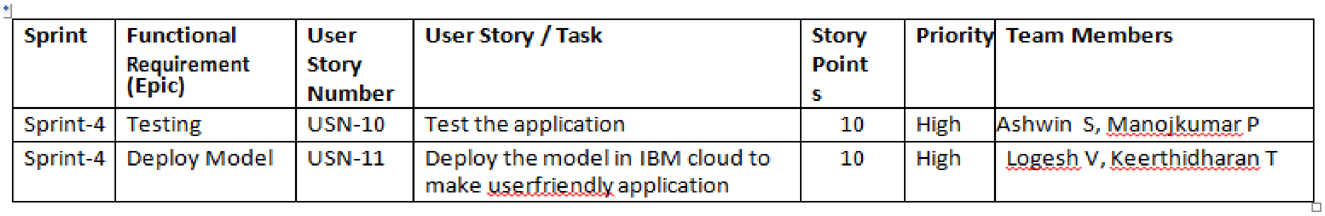
Product Backlog, Sprint Schedule, and Estimation

Use the below template to create product backlog and sprint schedule



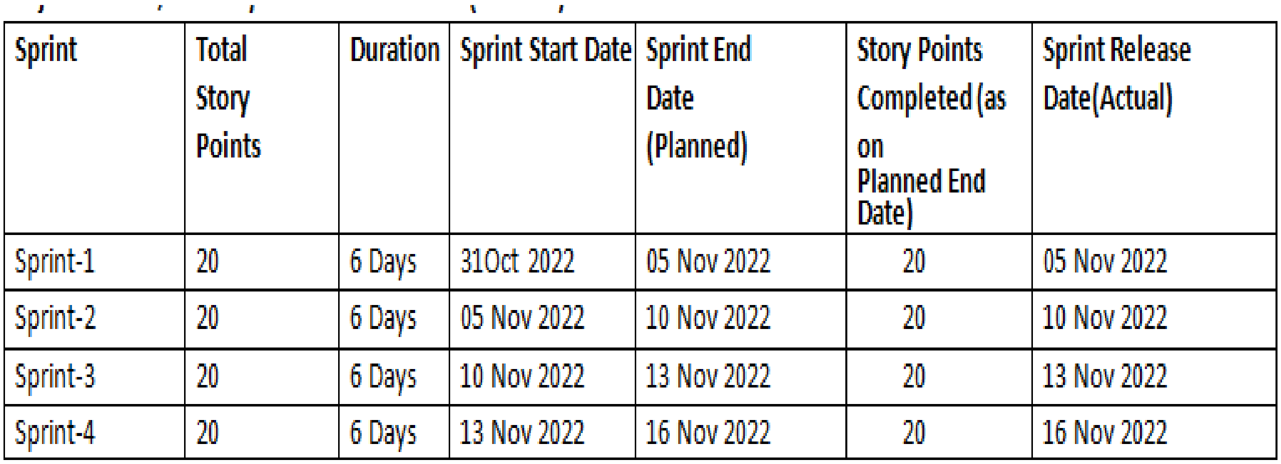


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Point s	Priority	Team Members
Sprint-1	Rainfall PredictionML Model (Dataset)	USN-1	Weather Dataset Collection, Data preprocessing, Data Visualization.	5	High	Ashwin S, Manojkumar P
Sprint-1		USN-2	Train ModelusIng Different machine learningAlgorithms	5	High	Logesh V, Keerthidharan T
Sprint-1		USN-3	Test the model and give best	1 0	High	Ashwin S, Manojkumar P
Sprint-2	Front end	USN-4	Finalization of background and its requirements.	5	Medium	Logesh V, Keerthidharan T
Sprint-2		USN-5	Necessary input boxes designed to get input from the user	5	Medium	Ashwin S, Manojkumar P
Sprint-2		USN-6	Checking whether the data is valid and predicting accordingly to the model used	4	Medium	Ashwin S, Logesh V
Sprint-2	Dashboard	USN-7	Attractive dashboard for forecasting designed	6	Low	Logesh V, Keerthidharan T
Sprint-3	Rainfall Prediction	USN-8	User enter the location, temperature, humidity and other key factors	1 0	High	Ashwin S, Manojkumar P
Sprint-3		USN-9	Predict the rainfall and display the result	1 0	High	Ashwin S, , Keerthidharan T



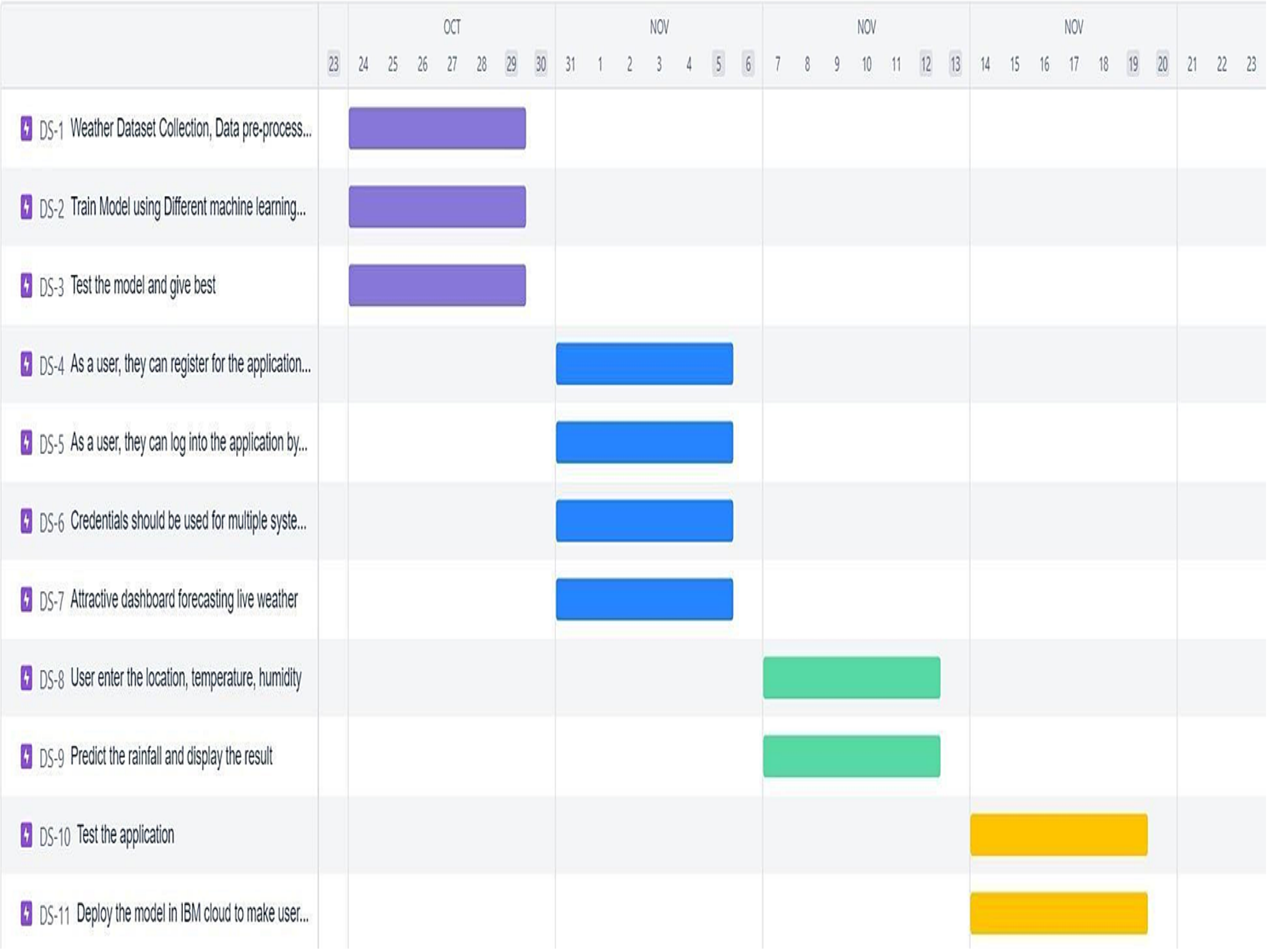
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Point s	Priority	Team Members
Sprint-4	Testing	USN-10	Test the application	10	High	Ashwin S, Manojkumar P
Sprint-4	Deploy Model	USN-11	Deploy the model in IBM cloud to make userfriendly application	10	High	Logesh V, Keerthidharan T

Project Tracker, Velocity & Burndown Chart:



Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	31Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-2	20	6 Days	05 Nov 2022	10 Nov 2022	20	10 Nov 2022
Sprint-3	20	6 Days	10 Nov 2022	13 Nov 2022	20	13 Nov 2022
Sprint-4	20	6 Days	13 Nov 2022	16 Nov 2022	20	16 Nov 2022

6.2 Reports from JIRA



DS-1 Weather Dataset Collection, Data pre-process...

DS-2 Train Model using Different machine learning...

DS-3 Test the model and give best

DS-4 As a user, they can register for the application...

DS-5 As a user, they can log into the application by...

DS-6 Credentials should be used for multiple syste...

DS-7 Attractive dashboard forecasting live weather

DS-8 User enter the location, temperature, humidity

DS-9 Predict the rainfall and display the result

DS-10 Test the application

DS-11 Deploy the model in IBM cloud to make user...

7.CODING AND SOLUTIONING

Feature 1: To retrieve information from IBM cloud account using API Key

```
import requests import json
API_KEY = "PQBr9MBF7mFuSh2VVLfOE-liIA04VH-h5VEk8EfjFIuw"

token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})

mltoken = token_response.json()["access_token"] print("ML Token",mltoken)
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken
```

Feature 2: To get predictions when the user enters the values and connecting to the deployed ML model using scoring end point

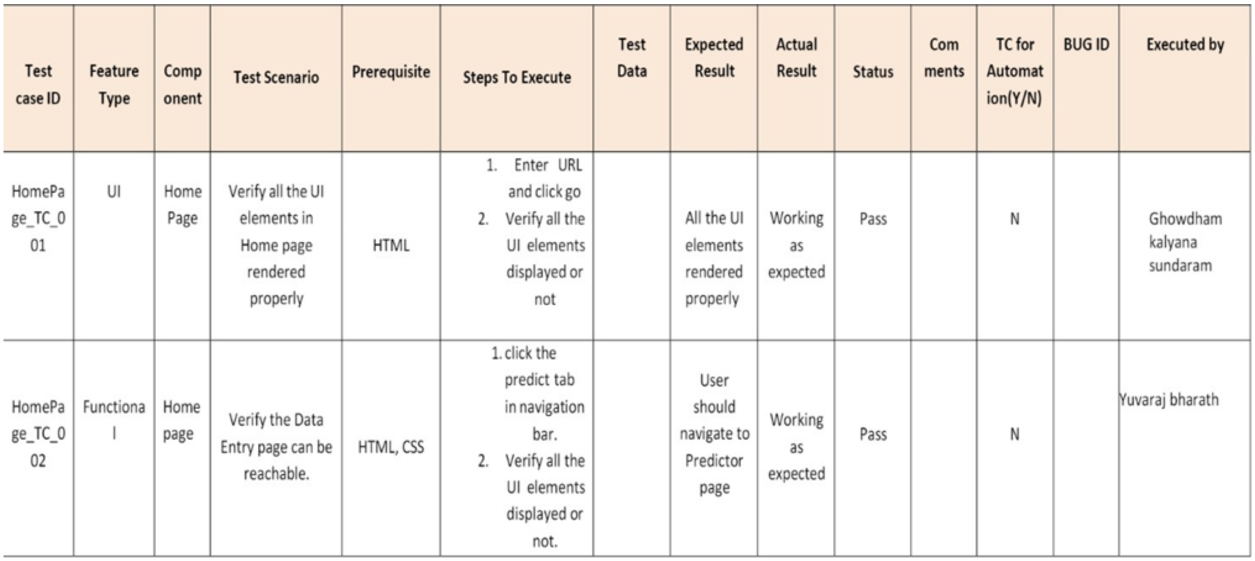
```
def predict():

res = " "

# If a form is submitted

if request.method == "POST":

Location = request.form.get('Location') MinTemp = request.form['MinTemp'] MaxTemp =
request.form['MaxTemp'] Rainfall = request.form['Rainfall']
WindGustSpeed = request.form['WindGustSpeed'] WindSpeed9am =
request.form['WindSpeed9am'] WindSpeed3pm = request.form['WindSpeed3pm'] Humidity9am
= request.form['Humidity9am'] Humidity3pm = request.form['Humidity3pm'] Pressure9am =
request.form['Pressure9am'] Pressure3pm = request.form['Pressure3pm'] Temp9am =
request.form['Temp9am']
Temp3pm = request.form['Temp3pm'] RainToday = request.form.get('RainToday') WindGustDir
= request.form.get('WindGustDir') WindDir9am = request.form.get('WindDir9am') WindDir3pm
= request.form.get('WindDir3pm')
new_row =
{'Location':Location,'MinTemp':MinTemp,'MaxTemp':MaxTemp,'Rainfall':Rainfall,'WindGustSp
eed':WindGustSpeed,'WindSpeed9am':WindSpeed9am,'WindSpeed3pm':WindSpeed3pm,'Hu
midity9am':Humidity9am,'Humidity3pm':Humidity3pm,'Pressure9am':Pressure9am,'Pressure

3pm':Pressure3pm,'Temp9am':Temp9am,'Temp3pm':Temp3pm,'RainToday':RainToday,'WindG
ustDir':WindGustDir,'WindDir9am':WindDir9am,'WindDir3pm':WindDir3pm}
print(new_row) new_df =
pd.DataFrame(columns=['Location','MinTemp','MaxTemp','Rainfall','WindGustSpeed','WindSpe
ed9am','WindSpeed3pm','Humidity9am','Humidity3pm','Pressure9am','Pressure3pm','Temp9a
m','Temp3pm','RainToday','WindGustDir','WindDir9am','WindDir3pm'])
new_df = new_df.append(new_row,ignore_index=True) labeled =
new_df[['Location','MinTemp','MaxTemp','Rainfall','WindGustSpeed','WindSpeed9am','WindSp
eed3pm','Humidity9am','Humidity3pm','Pressure9am','Pressure3pm','Temp9am','Temp3pm','R
ainToday','WindGustDir','WindDir9am','WindDir3pm']]

X = labeled.values print(X)
payload_scoring = {"input_data": [{"field":
[['Location','MinTemp','MaxTemp','Rainfall','WindGustSpeed','WindSpeed9am','WindSpeed3pm
','Humidity9am','Humidity3pm','Pressure9am','Pressure3pm','Temp9am','Temp3pm','RainyTod
ay','WindGustDir','WindDir9am','WindDir3pm']], "values": X.tolist()}]}

response_scoring =
requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/73230b85-51ea-45d b-
baa7-e86b5d528fbe/predictions?version=2022-11-14',
json=payload_scoring,headers={'Authorization': 'Bearer ' + mltoken})

print("Scoring response") predictions = response_scoring.json() print(predictions)
output = predictions['predictions'][0]['values'][0][0] print(output)
else:

output = "" if output == 1:
return redirect(url_for('chance')) elif output == 0:

return redirect(url_for('nochance'))

return render_template("index.html", output = res)
```

Feature 3 : To navigate between pages

```
<div class="navbar">

<ul>

<div class="nav"><a href="">HOME</a></div>

<div class="nav"><a href="{{ url_for('predict') }}">PREDICTOR</a></div>

<div class="nav"><a href="{{ url_for('help') }}">HELP</a></div>

<div class="nav"><a href="{{ url_for('contact') }}">CONTACT</a></div>

</ul>

</div>
```

8.TESTING

8.1 Test Cases:



Test case ID	Feature Type	Component	Test Scenario	Prerequisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed by
HomePage_TC_001	UI	Home Page	Verify all the UI elements in Home page rendered properly	HTML	1. Enter URL and click go 2. Verify all the UI elements displayed or not		All the UI elements rendered properly	Working as expected	Pass		N		Ghowdham kalyana sundaram
HomePage_TC_002	Functional	Home page	Verify the Data Entry page can be reachable.	HTML, CSS	1. click the predict tab in navigation bar. 2. Verify all the UI elements displayed or not.		User should navigate to Predictor page	Working as expected	Pass		N		Yuvaraj bharath

Predict_Page_TC_003	UI	Predict Page	Verify all the UI elements in Predict page rendered properly	HTML,CSS	1. Enter URL and click go 2. Verify all the UI elements displayed or not		All the UI elements rendered properly	Working as expected	Pass		N		Madhu singh
PredictPage_TC_004	Functional	Predict Page	Enter all the values and verify the prediction	Flask	1. Enter URL and click go 2. Enter the values for 17 attributes 3. Click Predict	NewCastle 13.4 22.6 0.6 44 21 24 70 78 1007.7 1007.1 34 32 Yes WSW NNW ESE	Redirect to corresponding html page (chance/no chance)	Working as expected	Pass		N		Raksith.R
OutputPage_TC_005	Functional	Chance Page	Verify whether it is redirected to chance page		1. Enter URL and click go 2. Enter the values and click predict button	Prediction = 1	Redirect to chance page	Working as expected	Pass		N		Nixon christhuraj
					3. If prediction equals one, chance page is displayed.								
OutputPage_TC_006	Functional	No chance Page	Verify whether it is redirected to no chance page		1. Enter URL and click go 2. Enter the values and click predict button 3. If prediction equals zero, no chance page is displayed.	Prediction = 0	Redirect to no chance page	Working as expected	Pass		N		Ghowdham kalyana sundaram

S.NO	Test Scenerios
1	Verify all the UI elements in Home page rendered properly.
2	Verify the Data Entry page can be reachable.
3	Verify all the UI elements in Predict page rendered properly
4	Enter all the values and verify the prediction
5	Verify whether it is redirected to chance page
6	Verify whether it is redirected to no chance page

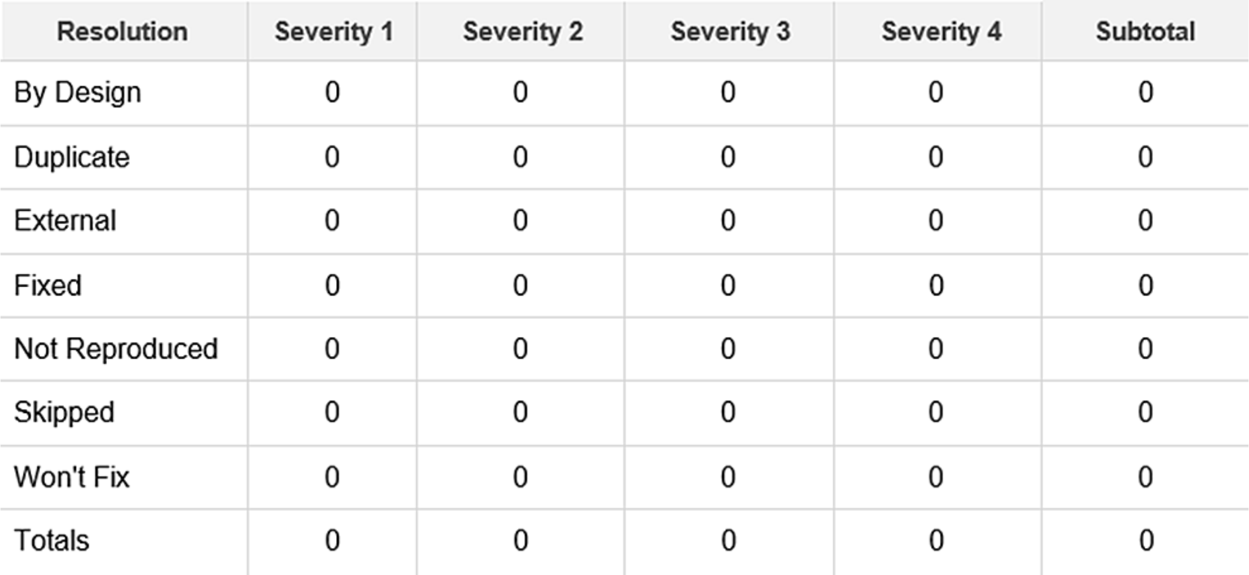
8.2 USER ACCEPTANCE TESTING

1.Purpose of Document

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

2.Defect Analysis

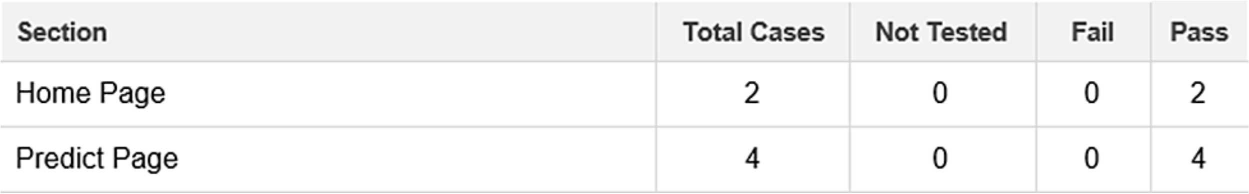
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved



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

3.Test Case Analysis

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

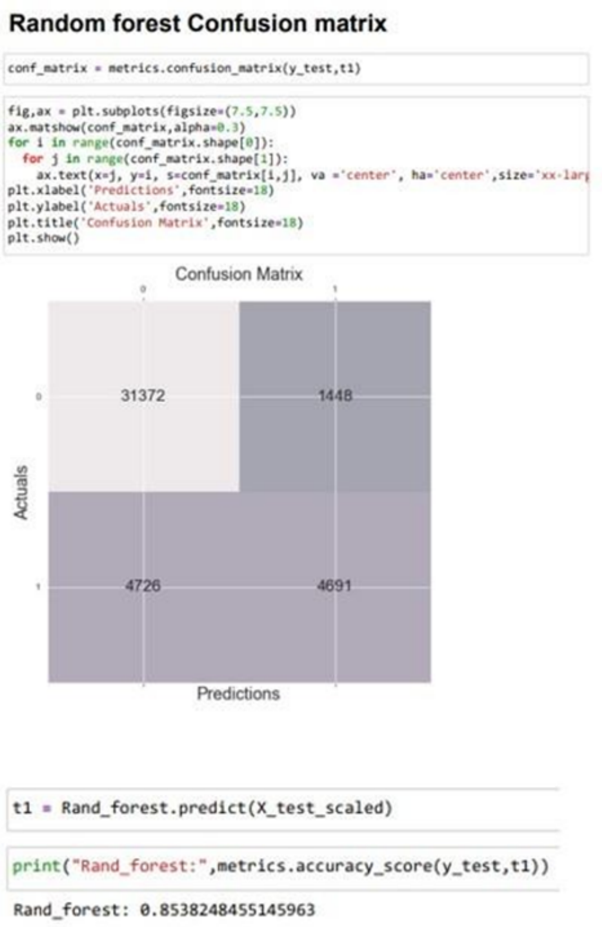


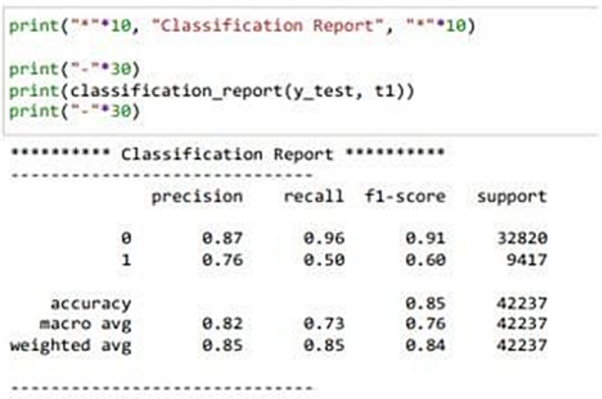
Section	Total Cases	Not Tested	Fail	Pass
Home Page	2	0	0	2
Predict Page	4	0	0	4

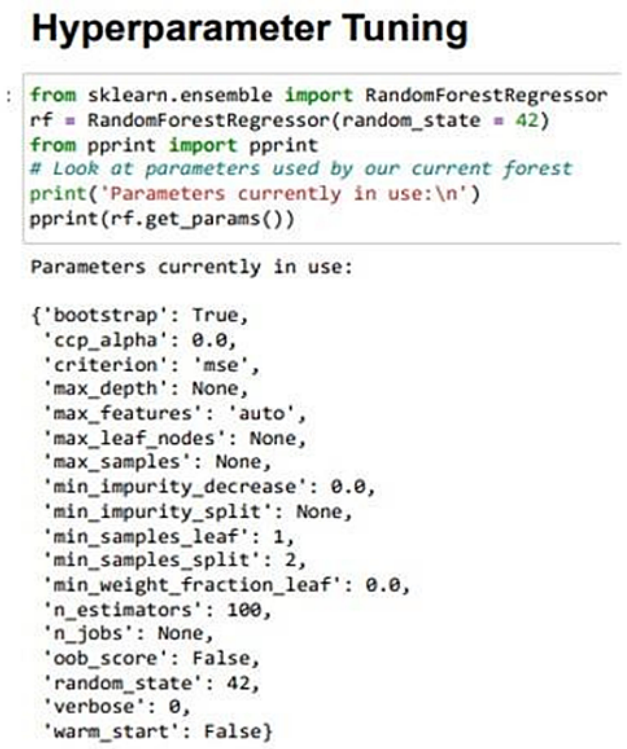
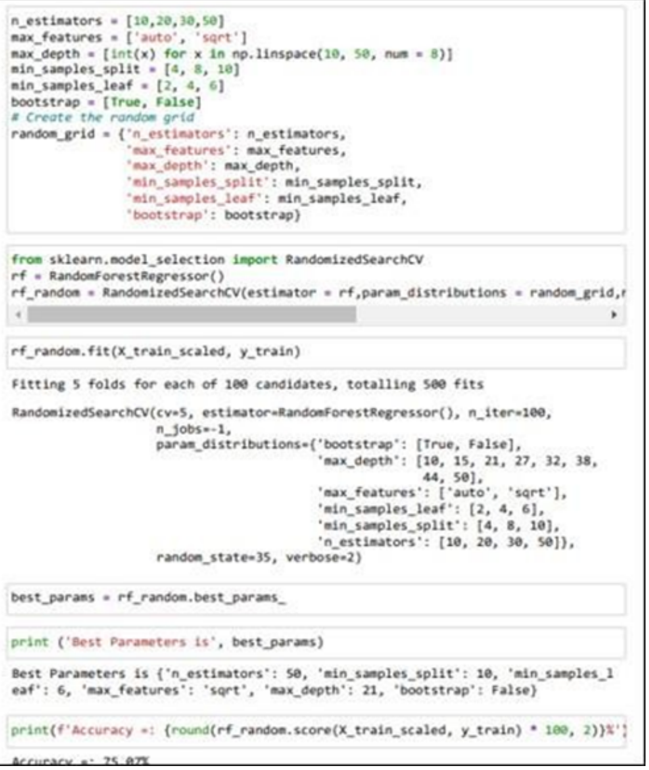
9.RESULTS

9.1 Performance Metrics

S.No.	Parameter	Values	Screenshot

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

			
			print("*"*10, "Classification Report", "*"*10)
			
			print("-"*30)
			print(classification_report(y_test, t1))
			print("-"*30)
			
			********** Classification Report **********
			------------------------------ 	
			              precision recall f1-score support
			
			           0 0.87 0.96 0.91 32820
			           1 0.76 0.50 0.60 9417
			
			    accuracy 0.85 42237
			   macro avg 0.82 0.73 0.76 42237
			weighted avg 0.85 0.85 0.84 42237
			
			------------------------------ 	

2.	Tune the Model	Hyperparameter Tuning & Validation Method - RandomizedSearchCV	
			

Hyperparameter Tuning

```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(random_state = 42)
from pprint import pprint
# Look at parameters used by our current forest
print('Parameters currently in use:\n')
pprint(rf.get_params())
```

```
Parameters currently in use:

{'bootstrap': True,
 'ccp_alpha': 0.0,
 'criterion': 'mse',
 'max_depth': None,
 'max_features': 'auto',
 'max_leaf_nodes': None,
 'max_samples': None,
 'min_impurity_decrease': 0.0,
 'min_impurity_split': None,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'n_estimators': 100,
 'n_jobs': None,
 'oob_score': False,
 'random_state': 42,
 'verbose': 0,
 'warm_start': False}
```

```
n_estimators = [10,20,30,50]
max_features = ['auto', 'sqrt']
max_depth = [int(x) for x in np.linspace(10, 50, num = 8)]
min_samples_split = [4, 8, 10]
min_samples_leaf = [2, 4, 6]
bootstrap = [True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
```

```
from sklearn.model_selection import RandomizedSearchCV
rf = RandomForestRegressor()
rf_random = RandomizedSearchCV(estimator = rf,param_distributions = random_grid,r
```

```
rf_random.fit(X_train_scaled, y_train)
```

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits

RandomizedSearchCV(cv=5, estimator=RandomForestRegressor(), n_iter=100,
                   n_jobs=-1,
                   param_distributions={'bootstrap': [True, False],
                                        'max_depth': [10, 15, 21, 27, 32, 38,
                                                      44, 50],
                                        'max_features': ['auto', 'sqrt'],
                                        'min_samples_leaf': [2, 4, 6],
                                        'min_samples_split': [4, 8, 10],
                                        'n_estimators': [10, 20, 30, 50]},
                   random_state=35, verbose=2)
```

```
best_params = rf_random.best_params_
```

```
print ('Best Parameters is', best_params)
```

```
Best Parameters is {'n_estimators': 50, 'min_samples_split': 10, 'min_samples_l
eaf': 6, 'max_features': 'sqrt', 'max_depth': 21, 'bootstrap': False}
```

```
print(f'Accuracy =: {round(rf_random.score(X_train_scaled, y_train) * 100, 2)}%'
```

```
Accuracy =: 75.07%
```

10. ADVANTAGES & DISADVANTAGES:

10.1 Advantages:

●As Weather conditions have been changing for the time being this helps people to know about the rainfall prediction

●To avoid unnecessary floods by opening dams with the help of rainfall prediction

●Farmers and fisherman will get the most advantage of these rainfall details so that we they can plan accordingly

●During the monsoon days it helps the government to find the evacuation areas to avoid loss of human life and costly things

10.2 Disadvantages:

●As the data was collected from limited places so it helps only for the people who located in those areas.

●In case the data was collected being wrong the algorithm will produce the wrong prediction

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

11. CONCLUSION:

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

12. FUTURE SCOPE:

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

13. APPENDIX

13.1 Source Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
import scipy.stats as stats
from sklearn.model_selection import train_test_split
from collections import Counter
from imblearn.over_sampling import SMOTE
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
from sklearn import metrics
from sklearn.ensemble import RandomForestClassifier
from catboost import CatBoostClassifier
from xgboost import XGBClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
import joblib

df = pd.read_csv("weatherAUS.csv")
pd.set_option("display.max_columns", None)
df

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

# Handle Missing Values
```

```
df.isnull().sum()*100/len(df)
```

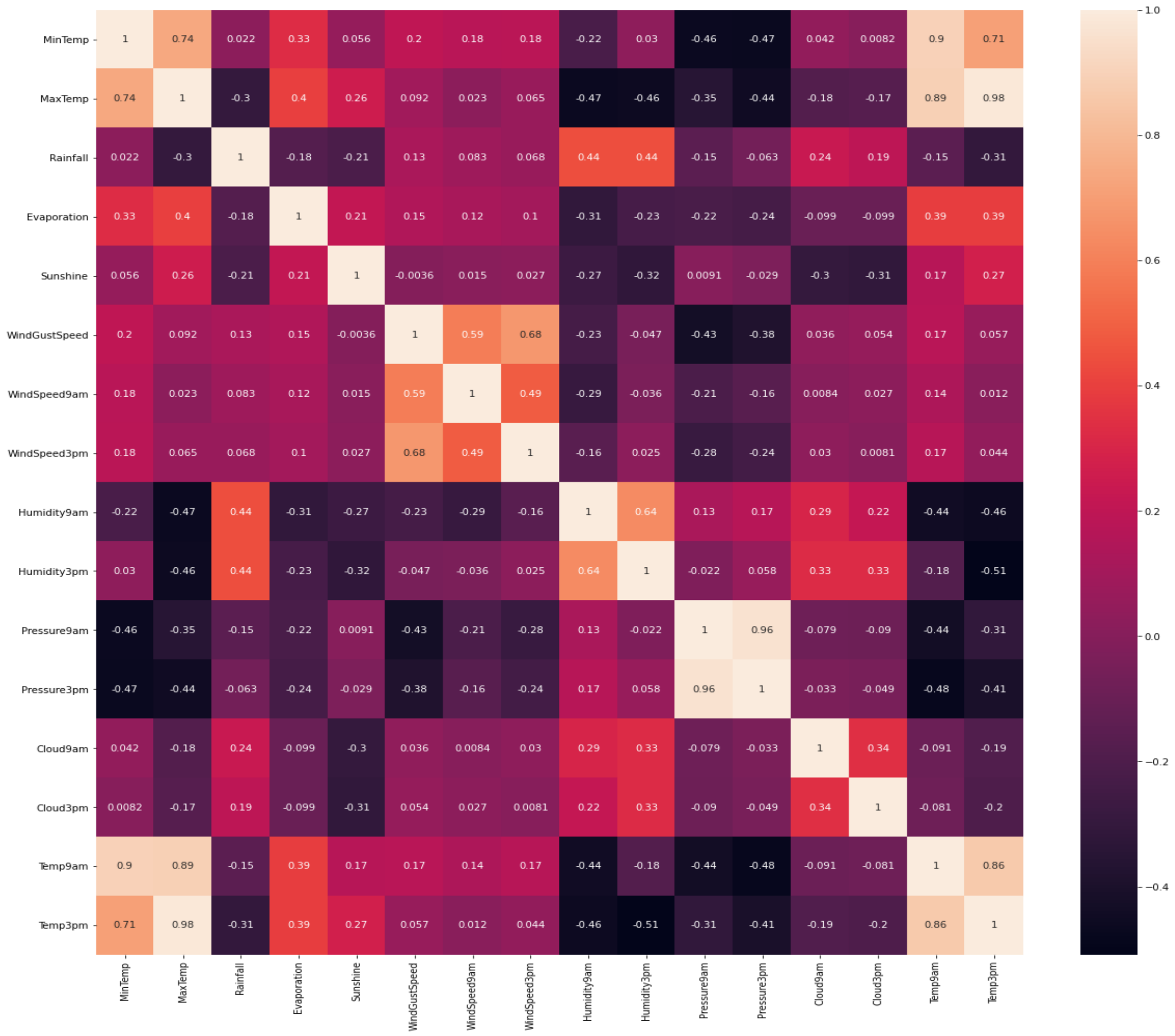
```
print(numerical_feature)
```

```
def randomsampleimputation(df, variable):
    df[variable]=df[variable]
    random_sample=df[variable].dropna().sample(df[variable].isnull().sum(),random_state=0)
    random_sample.index=df[df[variable].isnull()].index
    df.loc[df[variable].isnull(),variable]=random_sample
```

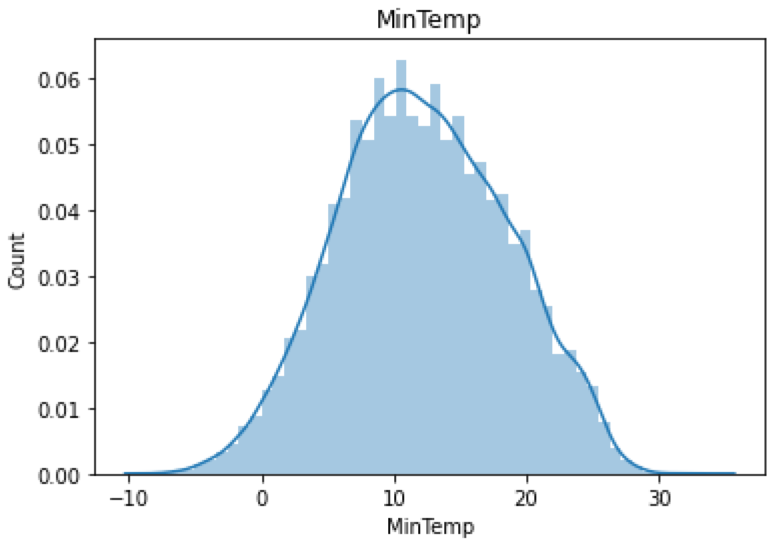
```
randomsampleimputation(df, "Cloud9am")
randomsampleimputation(df, "Cloud3pm")
randomsampleimputation(df, "Evaporation")
randomsampleimputation(df, "Sunshine")
```

```
df
```

```
corrmat = df.corr(method = "spearman")
plt.figure(figsize=(20,20))
#plot heat map
g=sns.heatmap(corrmat,annot=True)
```

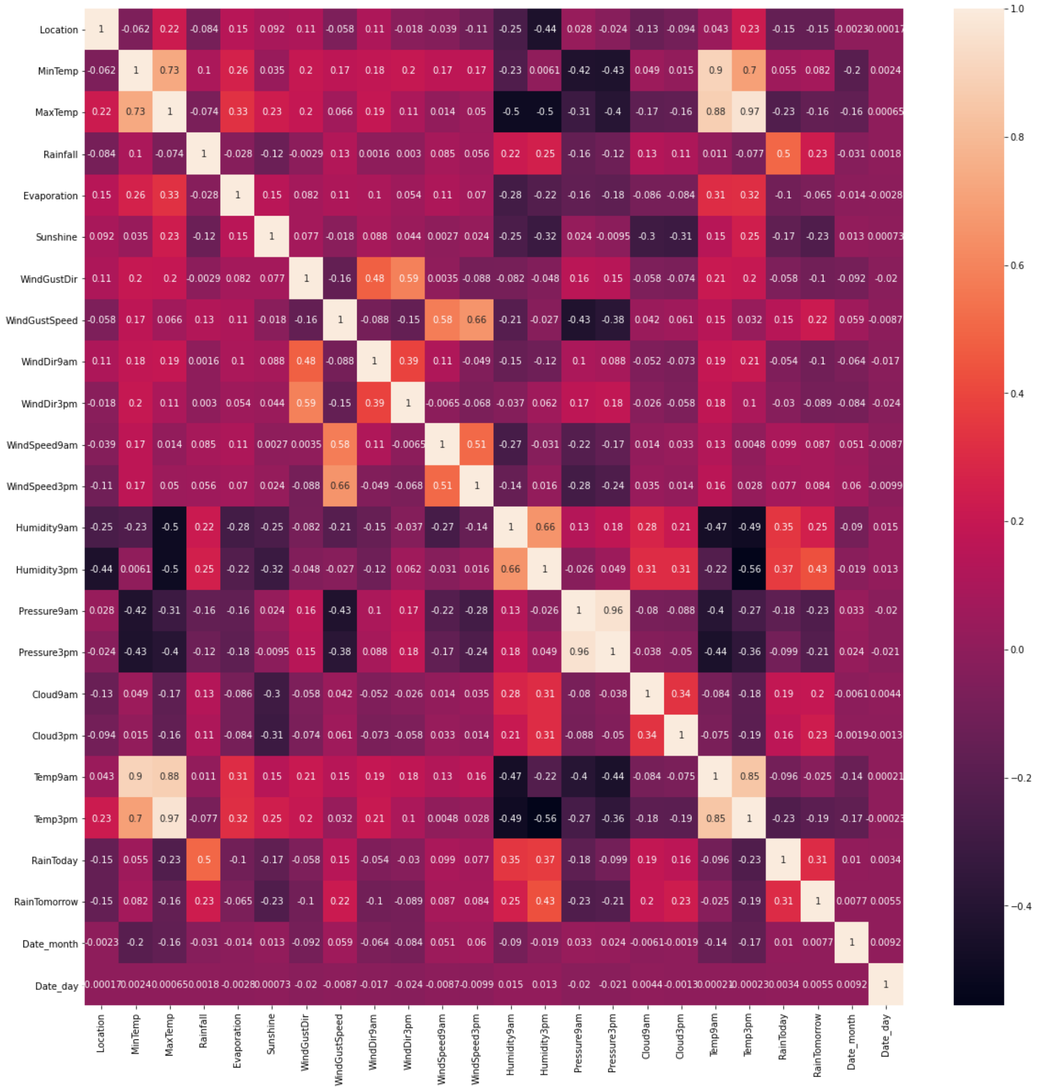


```
for feature in continuous_feature:
    data=df.copy()
    sns.distplot(df[feature])
    plt.xlabel(feature)
    plt.ylabel("Count")
    plt.title(feature)
    plt.figure(figsize=(15,15))
    plt.show()
```



```
#A for loop is used to plot a boxplot for all the continuous features to see the outliers
for feature in continuous_feature:
    data=df.copy()
    sns.boxplot(data[feature])
    plt.title(feature)
    plt.figure(figsize=(15,15))

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

df.isnull().sum()*100/len(df)

discrete_feature

def mode_nan(df,variable):
    mode=df[variable].value_counts().index[0]
    df[variable].fillna(mode,inplace=True)
mode_nan(df,"Cloud9am")
mode_nan(df,"Cloud3pm")

df["RainToday"] = pd.get_dummies(df["RainToday"], drop_first = True)
df["RainTomorrow"] = pd.get_dummies(df["RainTomorrow"], drop_first = True)
df
```

```
for feature in categorical_feature:
    print(feature, (df.groupby([feature])["RainTomorrow"].mean().sort_values(ascending = 
False)).index)

windgustdir = {'NNW':0, 'NW':1, 'WNW':2, 'N':3, 'W':4, 'WSW':5, 'NNE':6, 'S':7, 'SSW':8, 
'SW':9, 'SSE':10,
        'NE':11, 'SE':12, 'ESE':13, 'ENE':14, 'E':15}
winddir9am = {'NNW':0, 'N':1, 'NW':2, 'NNE':3, 'WNW':4, 'W':5, 'WSW':6, 'SW':7, 'SSW':8, 
'NE':9, 'S':10,
        'SSE':11, 'ENE':12, 'SE':13, 'ESE':14, 'E':15}
winddir3pm = {'NW':0, 'NNW':1, 'N':2, 'WNW':3, 'W':4, 'NNE':5, 'WSW':6, 'SSW':7, 'S':8, 
'SW':9, 'SE':10,
        'NE':11, 'SSE':12, 'ENE':13, 'E':14, 'ESE':15}
df["WindGustDir"] = df["WindGustDir"].map(windgustdir)
df["WindDir9am"] = df["WindDir9am"].map(winddir9am)
df["WindDir3pm"] = df["WindDir3pm"].map(winddir3pm)

df["WindGustDir"] = df["WindGustDir"].fillna(df["WindGustDir"].value_counts().index[0])
df["WindDir9am"] = df["WindDir9am"].fillna(df["WindDir9am"].value_counts().index[0])
df["WindDir3pm"] = df["WindDir3pm"].fillna(df["WindDir3pm"].value_counts().index[0])

df.isnull().sum()*100/len(df)

df1 = df.groupby(["Location"])["RainTomorrow"].value_counts().sort_values().unstack()

df1

df1[1].sort_values(ascending = False)

df1[1].sort_values(ascending = False).index

len(df1[1].sort_values(ascending = False).index)

location = {'Portland':1, 'Cairns':2, 'Walpole':3, 'Dartmoor':4, 'MountGambier':5,
        'NorfolkIsland':6, 'Albany':7, 'Witchcliffe':8, 'CoffsHarbour':9, 'Sydney':10,
        'Darwin':11, 'MountGinini':12, 'NorahHead':13, 'Ballarat':14, 'GoldCoast':15,
    'SydneyAirport':16, 'Hobart':17, 'Watsonia':18, 'Newcastle':19, 'Wollongong':20,
    'Brisbane':21, 'Williamtown':22, 'Launceston':23, 'Adelaide':24, 'MelbourneAirport':25,
    'Perth':26, 'Sale':27, 'Melbourne':28, 'Canberra':29, 'Albury':30, 'Penrith':31,
    'Nuriootpa':32, 'BadgerysCreek':33, 'Tuggeranong':34, 'PerthAirport':35, 'Bendigo':36,
    'Richmond':37, 'WaggaWagga':38, 'Townsville':39, 'PearceRAAF':40, 'SalmonGums':41,
    'Moree':42, 'Cobar':43, 'Mildura':44, 'Katherine':45, 'AliceSprings':46, 'Nhil':47,
    'Woomera':48, 'Uluru':49}
df["Location"] = df["Location"].map(location)

df["Date"] = pd.to_datetime(df["Date"], format = "%Y-%m-%dT", errors = "coerce")

df["Date_month"] = df["Date"].dt.month
df["Date_day"] = df["Date"].dt.day

df

corrmat = df.corr()
plt.figure(figsize=(20,20))
#plot heat map
g=sns.heatmap(corrmat,annot=True)
```



```
sns.countplot(df["RainTomorrow"])
```

```
df
```

```
for feature in continuous_feature:
    data=df.copy()
    sns.boxplot(data[feature])
    plt.title(feature)
    plt.figure(figsize=(15,15))
```

```
for feature in continuous_feature:
    print(feature)
```

```
IQR=df.MinTemp.quantile(0.75)-df.MinTemp.quantile(0.25)
lower_bridge=df.MinTemp.quantile(0.25)-(IQR*1.5)
upper_bridge=df.MinTemp.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)

df.loc[df['MinTemp']>=30.45,'MinTemp']=30.45
df.loc[df['MinTemp']<=-5.95,'MinTemp']=-5.95

IQR=df.MaxTemp.quantile(0.75)-df.MaxTemp.quantile(0.25)
lower_bridge=df.MaxTemp.quantile(0.25)-(IQR*1.5)
upper_bridge=df.MaxTemp.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)

df.loc[df['MaxTemp']>=43.5,'MaxTemp']=43.5
df.loc[df['MaxTemp']<=2.7,'MaxTemp']=2.7

IQR=df.Rainfall.quantile(0.75)-df.Rainfall.quantile(0.25)
lower_bridge=df.Rainfall.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Rainfall.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)

df.loc[df['Rainfall']>=1.5,'Rainfall']=1.5
df.loc[df['Rainfall']<=-0.89,'Rainfall']=-0.89

IQR=df.Evaporation.quantile(0.75)-df.Evaporation.quantile(0.25)
lower_bridge=df.Evaporation.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Evaporation.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)

df.loc[df['Evaporation']>=14.6,'Evaporation']=14.6
df.loc[df['Evaporation']<=-4.6,'Evaporation']=-4.6

IQR=df.WindGustSpeed.quantile(0.75)-df.WindGustSpeed.quantile(0.25)
lower_bridge=df.WindGustSpeed.quantile(0.25)-(IQR*1.5)
upper_bridge=df.WindGustSpeed.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
```

```
df.loc[df['WindGustSpeed']>=68.5,'WindGustSpeed']=68.5
df.loc[df['WindGustSpeed']<=8.5,'WindGustSpeed']=8.5

IQR=df.WindSpeed9am.quantile(0.75)-df.WindSpeed9am.quantile(0.25)
lower_bridge=df.WindSpeed9am.quantile(0.25)-(IQR*1.5)
upper_bridge=df.WindSpeed9am.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)

df.loc[df['WindSpeed9am']>=37,'WindSpeed9am']=37
df.loc[df['WindSpeed9am']<=-11,'WindSpeed9am']=-11

IQR=df.WindSpeed3pm.quantile(0.75)-df.WindSpeed3pm.quantile(0.25)
lower_bridge=df.WindSpeed3pm.quantile(0.25)-(IQR*1.5)
upper_bridge=df.WindSpeed3pm.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)

df.loc[df['WindSpeed3pm']>40.5,'WindSpeed3pm']=40.5
df.loc[df['WindSpeed3pm']<=-3.5,'WindSpeed3pm']=-3.5

IQR=df.Humidity9am.quantile(0.75)-df.Humidity9am.quantile(0.25)
lower_bridge=df.Humidity9am.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Humidity9am.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)

df.loc[df['Humidity9am']>=122,'Humidity9am']=122
df.loc[df['Humidity9am']<=18,'Humidity9am']=18

IQR=df.Pressure9am.quantile(0.75)-df.Pressure9am.quantile(0.25)
lower_bridge=df.Pressure9am.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Pressure9am.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)

df.loc[df['Pressure9am']>=1034.25,'Pressure9am']=1034.25
df.loc[df['Pressure9am']<=1001.05,'Pressure9am']=1001.05

IQR=df.Pressure3pm.quantile(0.75)-df.Pressure3pm.quantile(0.25)
```

```
lower_bridge=df.Pressure3pm.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Pressure3pm.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)

df.loc[df['Pressure3pm']>=1031.85,'Pressure3pm']=1031.85
df.loc[df['Pressure3pm']<=998.65,'Pressure3pm']=998.65

IQR=df.Temp9am.quantile(0.75)-df.Temp9am.quantile(0.25)
lower_bridge=df.Temp9am.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Temp9am.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)

df.loc[df['Temp9am']>=35.3,'Temp9am']=35.3
df.loc[df['Temp9am']<=-1.49,'Temp9am']=-1.49

IQR=df.Temp3pm.quantile(0.75)-df.Temp3pm.quantile(0.25)
lower_bridge=df.Temp3pm.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Temp3pm.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)

df.loc[df['Temp3pm']>=40.45,'Temp3pm']=40.45
df.loc[df['Temp3pm']<=2.45,'Temp3pm']=2.45

for feature in continuous_feature:
    data=df.copy()
    sns.boxplot(data[feature])
    plt.title(feature)
    plt.figure(figsize=(15,15))

def qq_plots(df, variable):
    plt.figure(figsize=(15,6))
    plt.subplot(1, 2, 1)
    df[variable].hist()
    plt.subplot(1, 2, 2)
    stats.probplot(df[variable], dist="norm", plot=plt)
    plt.show()
```

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

df.to_csv("preprocessed_1.csv", index=False)

X = df.drop(["RainTomorrow", "Date"], axis=1)
Y = df["RainTomorrow"]

# scaler = RobustScaler()
# X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X,Y, test_size =0.2, stratify = Y, random_state = 0)

y_train

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

cat = CatBoostClassifier(iterations=2000, eval_metric = "AUC")
cat.fit(X_train_res, y_train_res)

y_pred = cat.predict(X_test)
print(confusion_matrix(y_test,y_pred))
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))

metrics.plot_roc_curve(cat, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred, average=None)
```

```
rf=RandomForestClassifier()
rf.fit(X_train_res,y_train_res)

y_pred1 = rf.predict(X_test)
print(confusion_matrix(y_test,y_pred1))
print(accuracy_score(y_test,y_pred1))
print(classification_report(y_test,y_pred1))

metrics.plot_roc_curve(rf, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred1, average=None)

logreg = LogisticRegression()
logreg.fit(X_train_res, y_train_res)

y_pred2 = logreg.predict(X_test)
print(confusion_matrix(y_test,y_pred2))
print(accuracy_score(y_test,y_pred2))
print(classification_report(y_test,y_pred2))

metrics.plot_roc_curve(logreg, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred2, average=None)

gnb = GaussianNB()
gnb.fit(X_train_res, y_train_res)

y_pred3 = gnb.predict(X_test)
print(confusion_matrix(y_test,y_pred3))
print(accuracy_score(y_test,y_pred3))
print(classification_report(y_test,y_pred3))

metrics.plot_roc_curve(gnb, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred3, average=None)

knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train_res, y_train_res)
```

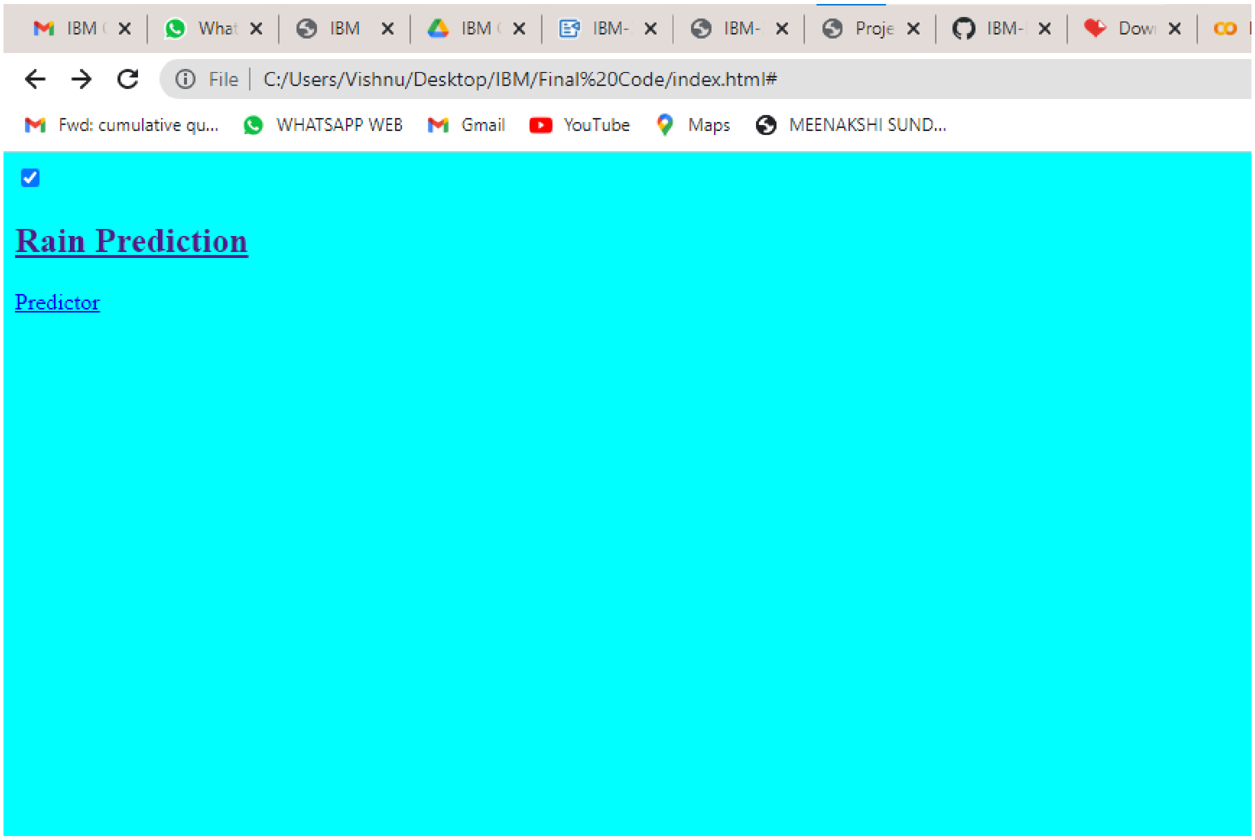
```
y_pred4 = knn.predict(X_test)
print(confusion_matrix(y_test,y_pred4))
print(accuracy_score(y_test,y_pred4))
print(classification_report(y_test,y_pred4))

metrics.plot_roc_curve(knn, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred4, average=None)

xgb = XGBClassifier()
xgb.fit(X_train_res, y_train_res)

y_pred6 = xgb.predict(X_test)
print(confusion_matrix(y_test,y_pred6))
print(accuracy_score(y_test,y_pred6))
print(classification_report(y_test,y_pred6))

metrics.plot_roc_curve(xgb, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred6, average=None)

svc = SVC()
svc.fit(X_train_res, y_train_res)

y_pred5 = svc.predict(X_test)
print(confusion_matrix(y_test,y_pred5))
print(accuracy_score(y_test,y_pred5))
print(classification_report(y_test,y_pred5))

metrics.plot_roc_curve(svc, X_test, y_test)
metrics.roc_auc_score(y_test, y_pred5, average=None)

# joblib.dump(rf, "rf.pkl")
# joblib.dump(cat, "cat.pkl")
# joblib.dump(logreg, "logreg.pkl")
# joblib.dump(gnb, "gnb.pkl")
# joblib.dump(knn, "knn.pkl")
joblib.dump(svc, "svc.pkl")
joblib.dump(xgb, "xgb.pkl")
```

HTML Codes

index.html:

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Rain Prediction</title>
    <link rel="stylesheet" href={{url_for('static',filename='style1.css')}}>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.14.0/css/all.min.css">
  </head>
  <body style="background-color:aqua;" >

    <section>
      <input type="checkbox" id="check">
      <header>
        </div>
        <h2><a href="#" class="logo">Rain Prediction</a></h2>
        <div class="navigation">

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

      </header><div class="content" style="margin-top: 8%;">

      </div>
</section>
  </body>
</html>
```

OUTPUT:



☑

Rain Prediction

Predictor

predictor.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;400;500;600;700;800;900&
display=swap" rel="stylesheet">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"
crossorigin="anonymous">
    <link rel="stylesheet" href={{url_for('static',filename='predictor.css')}}>
    <title>Rain Prediction</title>
```

```
</head>
<body>
    <section id="prediction-form">
        <form class="form" action="/predict", method="POST">
            <h1 class="my-3 text-center">Predictor</h1>
            <div class="row">
                <div class="col-md-6 my-2">
                    <div class="md-form">
                        <label for="date" class="date">Date</label>
                        <input type="date" class="form-control" id="date" name="date">
                    </div>
                </div>
                <div class="col-md-6 my-2">
                    <div class="md-form">
                        <label for="mintemp" class="mintemp"> Minimum temprature</label>
                        <input type="text" class="form-control" id="mintemp" name="mintemp">
                    </div>
                </div>
                <div class="col-md-6 my-2">
                    <div class="md-form">
                        <label for="maxtemp" class="maxtemp">Maximum Temperature</label>
                        <input type="text" class="form-control" id="maxtemp" name="maxtemp">
                    </div>
                </div>
                <div class="col-md-6 my-2">
                    <div class="md-form">
                        <label for="rainfall" class="rainfall">Rainfall</label>
                        <input type="text" class="form-control" id="rainfall" name="rainfall">
                    </div>
                </div>
                <div class="col-md-6 my-2">
                    <div class="md-form">
                        <label for="evaporation" class="evaporation">Evaporation</label>
                        <input type="text" class="form-control" id="evaporation" name="evaporation">
                    </div>
                </div>
                <div class="col-md-6 my-2">
```

```
                    <div class="md-form">
                        <label for="sunshine" class="sunshine">Sunshine</label>
                        <input type="text" class="form-control" id="sunshine" name="sunshine">
                    </div>
                </div>
                <div class="col-md-6 my-2">
                    <div class="md-form">
                        <label for="windgustspeed" class="windgustspeed">Wind Gust Speed</label>
                        <input type="text" class="form-control" id="windgustspeed"
name="windgustspeed">
                    </div>
                </div>
                <div class="col-md-6 my-2">
                    <div class="md-form">
                        <label for="windspeed9am" class="windspeed9am">Wind Speed 9am</label>
                        <input type="text" class="form-control" id="windspeed9am"
name="windspeed9am">
                    </div>
                </div>
                <div class="col-md-6 my-2">
                    <div class="md-form">
                        <label for="windspeed3pm" class="windspeed3pm">Wind Speed 3pm</label>
                        <input type="text" class="form-control" id="windspeed3pm"
name="windspeed3pm">
                    </div>
                </div>
                <div class="col-md-6 my-2">
                    <div class="md-form">
                        <label for="humidity9am" class="humidity9am">Humidity 9am</label>
                        <input type="text" class="form-control" id="humidity9am"
name="humidity9am">
                    </div>
                </div>
                <div class="col-md-6 my-2">
                    <div class="md-form">
                        <label for="humidity3pm" class="humidity3pm">Humidity 3pm</label>
                        <input type="text" class="form-control" id="humidity3pm"
```

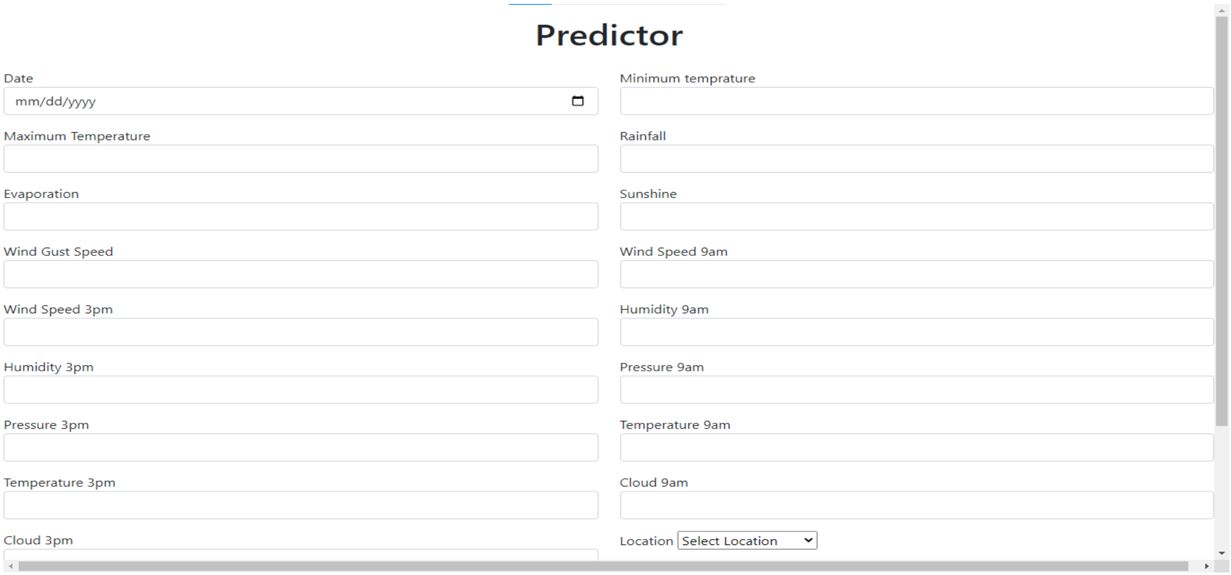
```
name="humidity3pm">
          </div>
        </div>
        <div class="col-md-6 my-2">
          <div class="md-form">
            <label for="pressure9am" class="pressure9am">Pressure 9am</label>
            <input type="text" class="form-control" id="pressure9am"
name="pressure9am">
          </div>
        </div>
        <div class="col-md-6 my-2">
          <div class="md-form">
            <label for="pressure3pm" class="pressure3pm">Pressure 3pm</label>
            <input type="text" class="form-control" id="pressure3pm"
name="pressure3pm">
          </div>
        </div>
        <div class="col-md-6 my-2">
          <div class="md-form">
            <label for="temp9am" class="temp9am">Temperature 9am</label>
            <input type="text" class="form-control" id="temp9am" name="temp9am">
          </div>
        </div>
        <div class="col-md-6 my-2">
          <div class="md-form">
            <label for="temp3pm" class=temp3pm">Temperature 3pm</label>
            <input type="text" class="form-control" id="temp3pm" name="temp3pm">
          </div>
        </div>
        <div class="col-md-6 my-2">
          <div class="md-form">
            <label for="cloud9am" class="cloud9am">Cloud 9am</label>
            <input type="text" class="form-control" id="cloud9am" name="cloud9am">
          </div>
        </div>
        <div class="col-md-6 my-2">
          <div class="md-form">
        <label for="cloud3pm" class="cloud3pm">Cloud 3pm</label>
        <input type="text" class="form-control" id="cloud3pm" name="cloud3pm">
    </div>
</div>
<div class="col-md-6 my-2">
    <div class="md-form">
        <label for="location" class="location" name="location">Location</label>
        <select class="location" id="location" name="location" aria-label="Location">
            <option selected>Select Location</option>
            <option value= 24>Adelaide</option>
            <option value= 7>Albany</option>
            <option value= 30>Albury</option>
            <option value= 46>AliceSprings</option>
            <option value= 33>BadgerysCreek</option>
            <option value= 14>Ballarat</option>
            <option value= 36>Bendigo</option>
            <option value= 21>Brisbane</option>
            <option value= 2>Cairns</option>
            <option value= 43>Cobar</option>
            <option value= 9>CoffsHarbour</option>
            <option value= 4>Dartmoor</option>
            <option value= 11>Darwin</option>
            <option value= 15>GoldCoast</option>
            <option value= 17>Hobart</option>
            <option value= 45>Katherine</option>
            <option value= 23>Launceston</option>
            <option value= 28>Melbourne</option>
            <option value= 25>Melbourne Airport</option>
            <option value= 44>Mildura</option>
            <option value= 42>Moree</option>
            <option value= 5>MountGambier</option>
            <option value= 12>MountGinini</option>
            <option value= 19>Newcastle       </option>
            <option value= 47>Nhil</option>
            <option value= 13>NorahHead</option>
            <option value= 6>NorfolkIsland</option>
            <option value= 32>Nuriootpa</option>
```

```
                            <option value= 40>PearceRAAF</option>
                            <option value= 31>Penrith</option>
                            <option value= 26>Perth</option>
                            <option value= 35>Perth Airport</option>
                            <option value= 1>Portland</option>
                            <option value= 37>Richmond</option>
                            <option value= 27>Sale</option>
                            <option value= 41>Salmon Gums</option>
                            <option value= 10>Sydney</option>
                            <option value= 16>Sydney Airport</option>
                            <option value= 39>Townsville</option>
                            <option value= 34>Tuggeranong</option>
                            <option value= 49>Uluru</option>
                            <option value= 38>WaggaWagga</option>
                            <option value= 3>Walpole</option>
                            <option value= 18>Watsonia</option>
                            <option value= 22>William Town</option>
                            <option value= 8>Witchcliffe</option>
                            <option value= 20>Wollongong</option>
                            <option value= 48>Woomera</option>
                        </select>
                    </div>
                </div>
                <div class="col-md-6 my-2">
                    <div class="md-form">
                        <label for="winddir9am" class="winddir9am" name = "winddir9am">Wind Direction at 9am</label>
                        <select class="winddir9am" id="winddir9am" name="winddir9am" aria-label="Wind Direction 9am">
                            <option selected>Select Wind Direction at 9am</option>
                            <option value= 1>N</option>
                            <option value= 5>W</option>
                            <option value= 10>S</option>
                            <option value= 15>E</option>
                            <option value= 2>NW</option>
                            <option value= 9>NE</option>
                            <option value= 7>SW</option>
                            <option value= 13>SE</option>
                            <option value= 0>NNW</option>
                            <option value= 3>NNE</option>
                            <option value= 8>SSW</option>
                            <option value= 11>SSE</option>
                            <option value= 4>WNW</option>
                            <option value= 6>WSW</option>
                            <option value= 12>ENE</option>
                            <option value= 14>ESE</option>
                        </select>
                    </div>
                </div>
                <div class="col-md-6 my-2">
                    <div class="md-form">
                        <label for="winddir3pm" class="winddir3pm" name = "winddir3pm">Wind Direction at 3pm</label>
                        <select class="winddir3pm" id="winddir3pm" name = "winddir3pm" aria-label="Wind Direction at 3pm">
                            <option selected>Select Wind Direction at 3pm</option>
                            <option value= 2>N</option>
                            <option value= 4>W</option>
                            <option value= 8>S</option>
                            <option value= 14>E</option>
                            <option value= 0>NW</option>
                            <option value= 11>NE</option>
                            <option value= 9>SW</option>
                            <option value= 10>SE</option>
                            <option value= 1>NNW</option>
                            <option value= 5>NNE</option>
                            <option value= 7>SSW</option>
                            <option value= 12>SSE</option>
                            <option value= 3>WNW</option>
                            <option value= 6>WSW</option>
                            <option value= 13>ENE</option>
                            <option value= 15>ESE</option>
                        </select>
                    </div>
```

```
        </div>
        <div class="col-md-6 my-2">
          <div class="md-form">
            <label for="windgustdir" class="windgustdir" name = "windgustdir">Wind Gust Direction</label>
            <select class="windgustdir" id="windgustdir" name = "windgustdir" aria-label="Wind Gust Direction">
              <option selected>Select Wind Gust Direction</option>
              <option value= 3>N</option>
              <option value= 4>W</option>
              <option value= 7>S</option>
              <option value= 15>E</option>
              <option value= 1>NW</option>
              <option value= 11>NE</option>
              <option value= 9>SW</option>
              <option value= 12>SE</option>
              <option value= 0>NNW</option>
              <option value= 6>NNE</option>
              <option value= 8>SSW</option>
              <option value= 10>SSE</option>
              <option value= 2>WNW</option>
              <option value= 5>WSW</option>
              <option value= 14>ENE</option>
              <option value= 13>ESE</option>
            </select>
          </div>
        </div>
        <div class="col-md-6 my-2">
          <div class="md-form">
            <label for="raintoday" class="raintoday" name="raintoday">Rain Today</label>
            <select class="raintoday" id="raintoday" name="raintoday" aria-label="Rain Today">
              <option selected>Did it Rain Today</option>
              <option value= 1>Yes</option>
              <option value= 0>No</option>
            </select>
          </div>
```

```
                </div>
                <div class="col-md-6 my-2 d-flex align-items-end justify-content-around">
                    <button type="submit" class="btn btn-info button" style="margin-left:
100%;">Predict</button>
                </div>
            </div>
        </form>
    </section>
    <div>
        <h1><center> {{ prediction }} </center></h1>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0"
crossorigin="anonymous"></script>
</body>
</html>
```

OUTPUT:



Predictor

Date
mm/dd/yyyy

Minimum temprature

Maximum Temperature

Rainfall

Evaporation

Sunshine

Wind Gust Speed

Wind Speed 9am

Wind Speed 3pm

Humidity 9am

Humidity 3pm

Pressure 9am

Pressure 3pm

Temperature 9am

Temperature 3pm

Cloud 9am

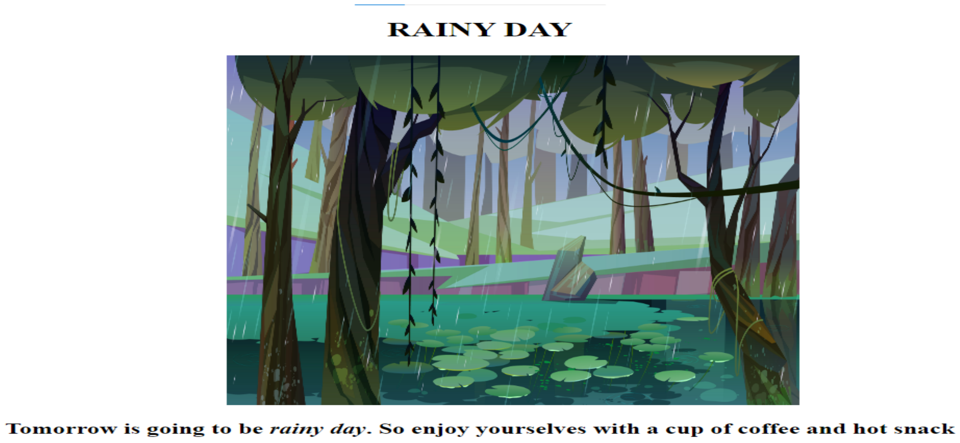
Cloud 3pm

Location Select Location

after_rainy.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;400;500;600;700;800;900&display=swap" rel="stylesheet">
    <link rel="stylesheet" href={{url_for('static',filename='after_rainy.css')}}>
    <title>Rainy Day</title>
</head>
<body>
    <h1 style="text-align: center; font-size: 3 rem; font-weight: bolder">SUNNY DAY</h1>
    <div class="rainyimg">
        <img src="../static/sunny.jpg" alt="Vasanth" style="height: 550px; width: 550px; margin-left: 32%">
    </div>
    <div>
        <h2><center> Tomorrow is going to be <span style="font-style: italic; font-weight: bolder;">sunny day</span>. So enjoy yourselves
        with a cool milkshake and icecream </center></h2>
   </div>
</body>
</html>
```

OUTPUT:



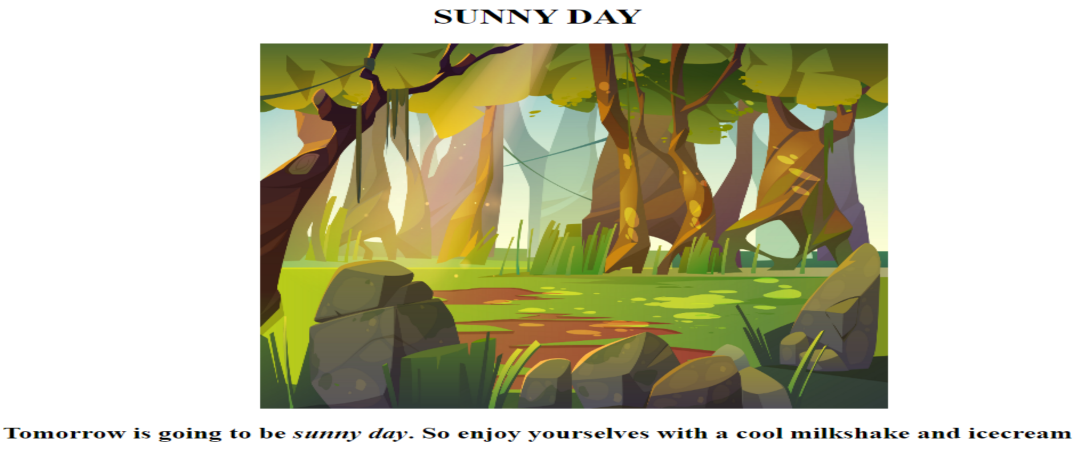
RAINY DAY

Tomorrow is going to be *rainy day*. So enjoy yourselves with a cup of coffee and hot snack

after_sunny.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;400;500;600;700;800;900&
display=swap" rel="stylesheet">
    <link rel="stylesheet" href={{url_for('static',filename='after_rainy.css')}}>
    <title>Rainy Day</title>
</head>
<body>
    <h1 style="text-align: center; font-size: 3 rem; font-weight: bolder">SUNNY DAY</h1>
    <div class="rainyimg">
        <img src="../static/sunny.jpg" alt="Vasanth" style="height: 550px; width: 550px; margin-
left: 32%">
    </div>
    <div>
        <h2><center> Tomorrow is going to be <span style="font-style: italic; font-weight:
bolder;">sunny day</span>. So enjoy yourselves
        with a cool milkshake and icecream </center></h2>
   </div>
</body>
</html>
```

OUTPUT:



SUNNY DAY

Tomorrow is going to be *sunny day*. So enjoy yourselves with a cool milkshake and icecream

Integrate flask with scoring end point

app.py:

```
from flask import Flask,render_template,url_for,request,jsonify
from flask_cors import cross_origin
import pandas as pd
import numpy as np
import datetime
import pickle



app = Flask(__name__, template_folder="template")
model = pickle.load(open("./models/cat.pkl", "rb"))
print("Model Loaded")

@app.route("/",methods=['GET'])
@cross_origin()
def home():
	return render_template("index.html")

@app.route("/predict",methods=['GET', 'POST'])
@cross_origin()
def predict():
	if request.method == "POST":
		# DATE
		date = request.form['date']
		day = float(pd.to_datetime(date, format="%Y-%m-%dT").day)
		month = float(pd.to_datetime(date, format="%Y-%m-%dT").month)
		# MinTemp
		minTemp = float(request.form['mintemp'])
		# MaxTemp
		maxTemp = float(request.form['maxtemp'])
		# Rainfall
		rainfall = float(request.form['rainfall'])
		# Evaporation
		evaporation = float(request.form['evaporation'])
```

```
# Sunshine
sunshine = float(request.form['sunshine'])
# Wind Gust Speed
windGustSpeed = float(request.form['windgustspeed'])
# Wind Speed 9am
windSpeed9am = float(request.form['windspeed9am'])
# Wind Speed 3pm
windSpeed3pm = float(request.form['windspeed3pm'])
# Humidity 9am
humidity9am = float(request.form['humidity9am'])
# Humidity 3pm
humidity3pm = float(request.form['humidity3pm'])
# Pressure 9am
pressure9am = float(request.form['pressure9am'])
# Pressure 3pm
pressure3pm = float(request.form['pressure3pm'])
# Temperature 9am
temp9am = float(request.form['temp9am'])
# Temperature 3pm
temp3pm = float(request.form['temp3pm'])
# Cloud 9am
cloud9am = float(request.form['cloud9am'])
# Cloud 3pm
cloud3pm = float(request.form['cloud3pm'])
# Cloud 3pm
location = float(request.form['location'])
# Wind Dir 9am
winddDir9am = float(request.form['winddir9am'])
# Wind Dir 3pm
winddDir3pm = float(request.form['winddir3pm'])
# Wind Gust Dir
windGustDir = float(request.form['windgustdir'])
# Rain Today
rainToday = float(request.form['raintoday'])

input_lst = [location , minTemp , maxTemp , rainfall , evaporation , sunshine ,
                                windGustDir , windGustSpeed , winddDir9am , winddDir3pm ,
windSpeed9am , windSpeed3pm ,
                                                    humidity9am , humidity3pm , pressure9am , pressure3pm ,
cloud9am , cloud3pm , temp9am , temp3pm ,
                                                    rainToday , month , day]
            pred = model.predict(input_lst)
            output = pred
            if output == 0:
                        return render_template("after_sunny.html")
            else:
                        return render_template("after_rainy.html")
          return render_template("predictor.html")

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

13.2 GitHub

GitHub Link: **https://github.com/IBM-EPBL/IBM-Project-30189-1660141562**

Project Demo Link: **https://drive.google.com/file/d/1N5y2siPI_LqQxmV-DeBBTmR1TowoGwKB/view?usp=sharing**