Exploratory Analysis of Rainfall Data in India for Agriculture

IBM-Project-23275-1659875881

Project Report

Team Members: Sai Spoorti N, Sakthi G, Shivani R, Shobana G

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
7. **CODING & SOLUTIONING**
8. **TESTING**
   - 8.1 Test Cases
   - 8.2 User Acceptance Testing
9. **RESULTS**
   - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

Source Code

GitHub Link

Chapter 1: INTRODUCTION

1.1 Project Overview

Rainfall has been a primary concern. The weather conditions have been changing for the time being. Rainfall forecasting is essential. Otherwise, it may lead to many disasters. Irregularly heavy rainfall may lead to the destruction of crops and heavy floods that can cause harm to human life. It is important to know exactly how much rain falls so that water resources can be used well, crops can grow well, and water structures can be planned ahead of time.

This comparative study focuses on modeling inputs, making the data visible, modelling methods, and pre-processing techniques. The results compare various evaluation metrics of these machine learning techniques and their reliability in predicting rainfall by analyzing weather data.

We will use classification algorithms such as decision trees, random forests, KNN, and boost. We will train and test the algorithms with the data. The best model is chosen, and the dataset is saved in PKL format. Once the model is saved, we integrate it with the Flask application and deploy it at IBM.

1.2 Purpose

Agriculture is the backbone of the Indian economy. For agriculture, the most important thing is the water source, i.e., rainfall. Farmers can protect their crops from rain if they know how much rain is going to fall ahead of time, which is why farmers try to predict how much rain will fall. crops from rain. So, in order to forecast the rainfall as accurately as possible, exploration and analysis of data on rainfall over various regions of India are especially important, especially when modeling areas where agricultural work has been done. coFuture rainfall predictions for those regions were made using various machine learning techniques such as XGBoost classifiers, SVM classifiers, decision trees, Naive Bayes classifiers, logistic regression, and so on.er, Logistic regression etc.

Chapter 2: LITERATURE SURVEY

2.1 References

	Paper	Year	Citation	Methodologies used
	Improved Regional Frequency Analysis of rainfall data	2022	Philomène Le Gall, Anne-Catherine Favre, Philippe Naveau, Clémentine Prieur, Improved Regional Frequency Analysis of rainfall data, Weather and Climate Extremes, Volume 36, 2022, 100456, ISSN 2212-0947, https://doi.org/10.1016/j.wace.2022.100456.	In this paper, they propose three models based on the clustering outputs. A comparison is made between their local, semi-regional and regional models.

	Probability Analysis for Prediction of Rainfall for Tamil Nadu Agricultural University, Coimbatore	2022	M. Nagarajan, & K. Ramaswamy. (2022). Probability Analysis for Prediction of Rainfall for Tamil Nadu Agricultural University, Coimbatore. Journal of Soil and Water Conservation, 5(2). Retrieved from https://epubs.icar.org.in/index.php/JSWC/article/view/126389	The rainfall data were analyzed by fitting different probability distributions like Normal, Log-normal, Gumbel, Log-Pearson Type Il, Gamma and Exponential about minimum D-Index.
	Analysis of a Long-Term IMD Gridded Rainfall Data for Dry Period in Meghalaya	2022	Phawa, R., Kusre, B.C. & Gupta, S. Analysis of a Long-Term IMD Gridded Rainfall Data for Dry Period in Meghalaya. J Indian Soc Remote Sens 50, 1959–1977 (2022). https://doi.org/10.1007/s12524-022-01575-y	Using IMD grid data, researchers in Meghalaya looked at how rainfall changes over space and time. The spatial and temporal variation of rainfall was studied for Meghalaya using IMD gridded data. Standard Precipitation Index (SPI) was also calculated to examine rainfall variation during dry and wet periods.

	Implementing a novel deep learning technique for rainfall forecasting via climatic variables: An approach via hierarchical clustering analysis	2022	Shah Fahad, Fang Su, Sufyan Ullah Khan, Muhammad Rashid Naeem, Kailei Wei, Implementing a novel deep learning technique for rainfall forecasting via climatic variables: An approach via hierarchical clustering analysis, Science of The Total Environment, Volume 854, 2023, 158760, ISSN 0048-9697, https://doi.org/10.1016/j.scitotenv.2022.158760.	It is a deep forecasting model based on an optimized (Gated Recurrent Unit) GRU neural network to predict rainfall in Pakistan based on 30 years of climate data from 1991 to 2020.
	Innovative and polygonal trend analyzes applications for rainfall data	2021	Şan, M., Akçay, F., Linh, N.T.T. et al. Innovative and polygonal trend analyzes applications for rainfall data in Vietnam. Theor Appl Climatol 144, 809–822 (2021). https://doi.org/10.1007/s00704-021-03574-4	In this paper, they compared IPTA, and ITA with the Significance Test and Mann-Kendall (MK) methods.

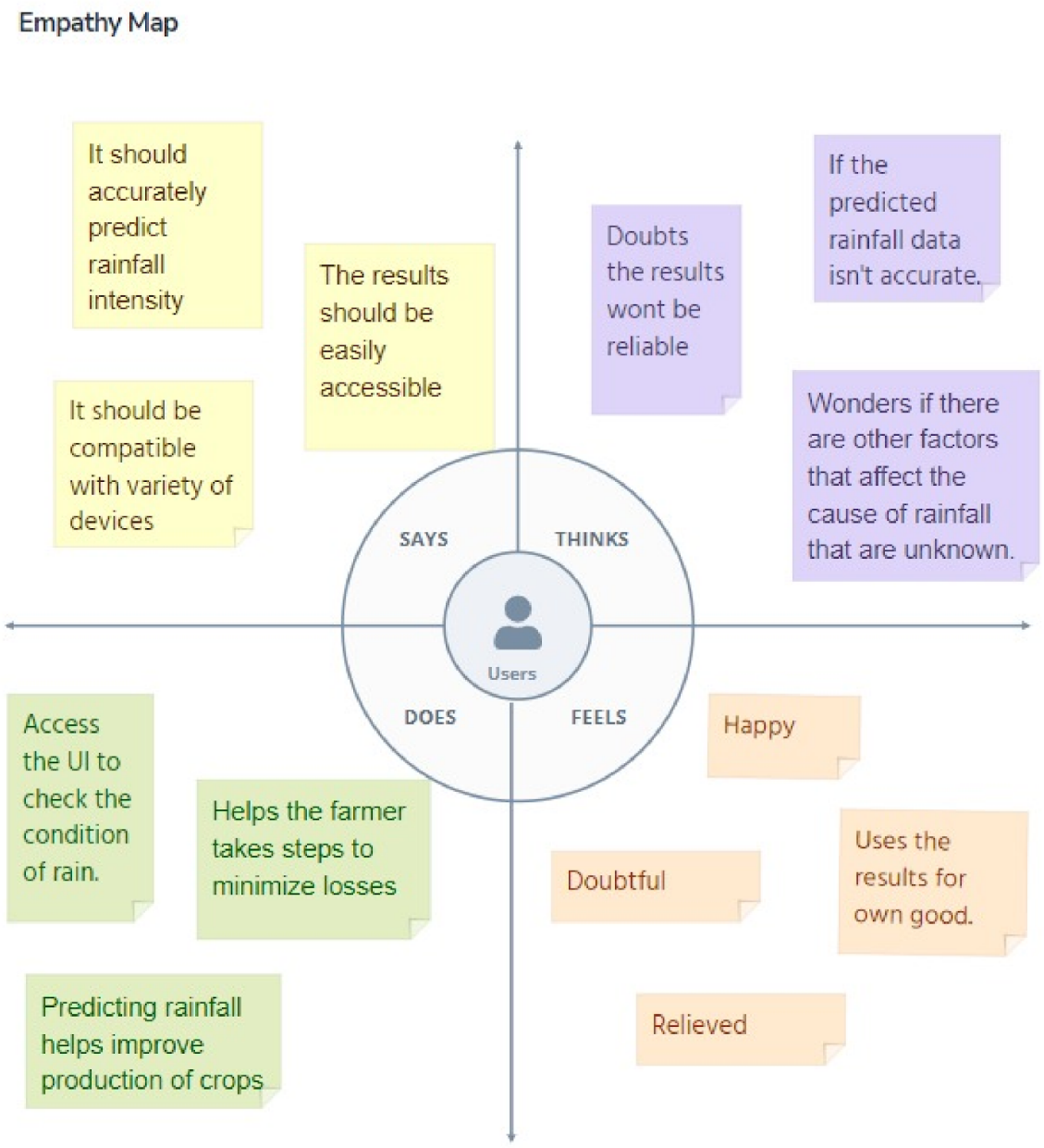
2.2 Problem Statement Definition

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Farmer	find rainfall status for suitable crop production	unable to get reliable source	rainfall information is not accurate	doubtful
PS-2	Vendor	sell raw materials	do not have enough supply	unpredictable rainfall patterns	disappointed

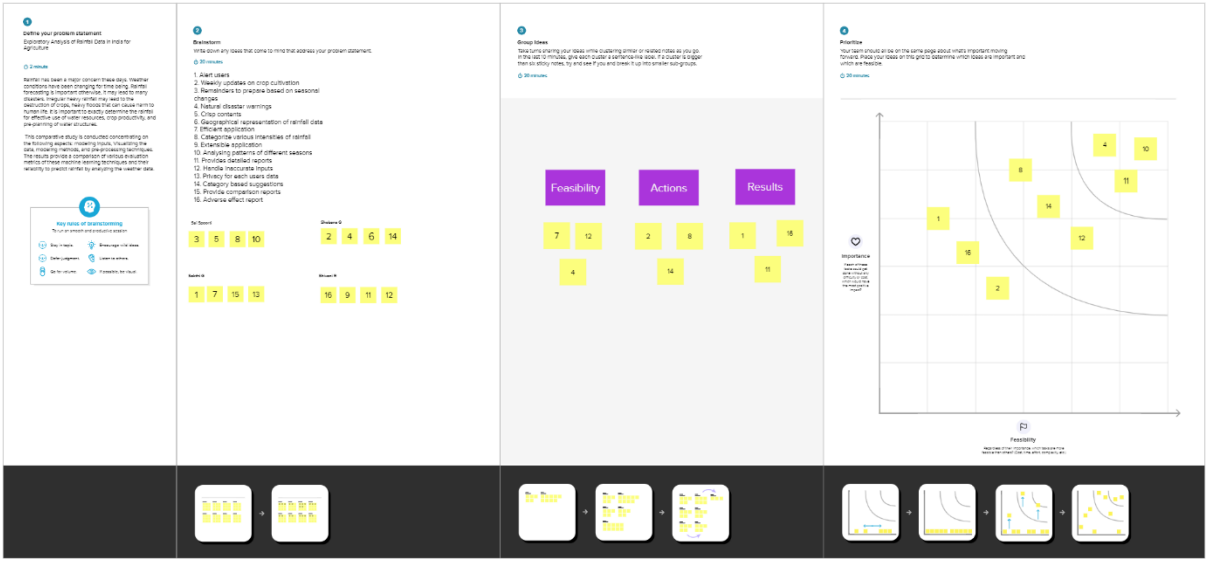
Chapter 3: IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. Creating an effective solution requires understanding the problem and the person experiencing it. Creating the map helps participants consider things from the user's perspective, along with their goals and challenges.



3.2 Ideation & Brainstorming



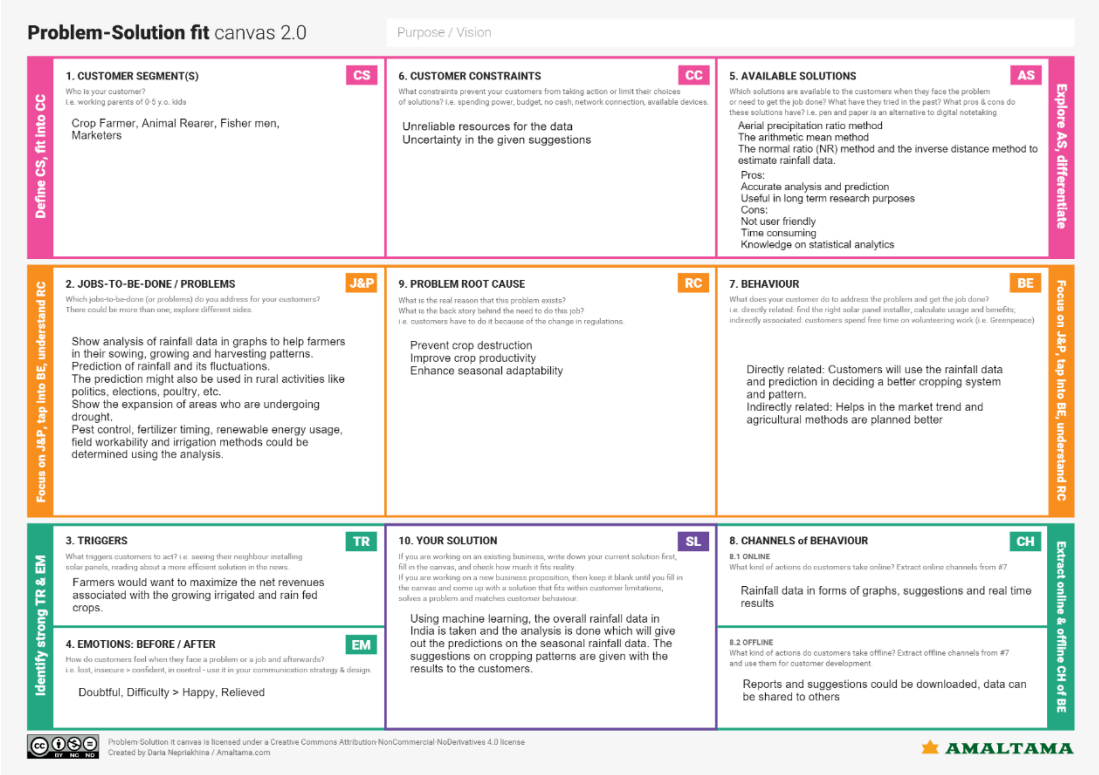
3.3 Proposed Solution

	Parameter	Description
	Problem Statement (Problem to be solved)	Rainfall has been a primary concern these days. The weather conditions have been changing for the time being. Rainfall forecasting is essential. Otherwise, it may lead to many disasters regularly ular heavy rainfall may lead to the destruction of crops and heavy floods that can cause harm to human life. It is essential to precisely determine the rainfall effectively using water resources, crop productivity, and preplanning of water structures.

	IdPredictionslution Description:	We will use classification algorithms such as decision trees, random forests, KNN, and boost. We will train and test the algorithms with the data. The best model is chosen, and the dataset is saved in PKL format. After saving the model, we integrate the Flask application and deploy the model in IBM.
	Novelty / Uniqueness	The novelty of the present study is that the model is not concentrated on one or more states or India overall. Instead, the current research considers predicting and forecasting rainfall for thirty-four meteorological subdivisions in India. As a result, the study will be highly beneficial, as it worked on a micro level in India. Also, its importance directly impacts the agriculture of the regions and how it could affect the other facets as well.
	Social Impact / Customer Satisfaction	India is an agricultural country, and a good monsoon will keep the secondary agricultural market steady. The economic growth of each year depends on the amount and duration of monsoon rai. A. A deficient monsoon can lead to the destruction of some crops, which may rein thelin thein scarcity of some agricultural products, which can cause food insecurity, security,curity and public unrest. In our analysis, we are trying to predict rainfall in India over the years, by months and by different suband use the predictions to help

		enhance productivity.
	Business Model (Revenue Model)	Parametric modelling with components for seasonal variation is used to represent the data. It detects changes in the rainfall process. The user will see the properties of trends and changes over time. Also, certain information, like wet days, the expected amount of rain, spell lengths, and extreme events, is delivered to the users on a subscription basis.
	Scalability of the Solution	One of the climate variables that has been studied a lot for a long time is the data on rainfall. It has a wide range of applications that also shed light on sowing data, facilitate policy decisions regarding cropping patterns, construct roads, and provide drinking water to urban and rural areas. This analysis could also be expanded into an application that displays plots and graphical data to the user.

Problem-Solution Fit



Chapter 4: REQUIREMENT ANALYSIS

4.1 Functional requirement

	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
	User Input	Registration through Form Registration through Gmail
	User Confirmation	Confirmation via Email Confirmation via OTP
	Prediction	Predicting the rainfall pattern

	Suggestions and analysis	Generate Suggestion Reports Provide visualization

4.2 Non-Functional requirements

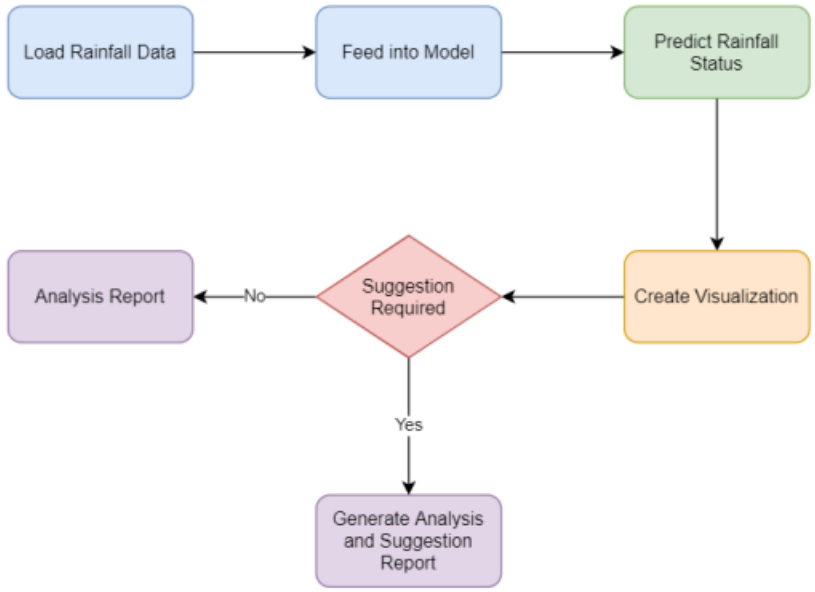
	Non-Functional Requirement	Description
	Usability	User-friendly
	Security	User’s requirement for planning is not shared
	Reliability	Better accuracy for predicting the rainfall
	Performance	Provide rainfall status and give suggestions immediately
	Availability	Available to the logged in users

	Scalability	Can be used for all agricultural activities including cultivation planning, irrigation, etc

Chapter 5: PROJECT DESIGN

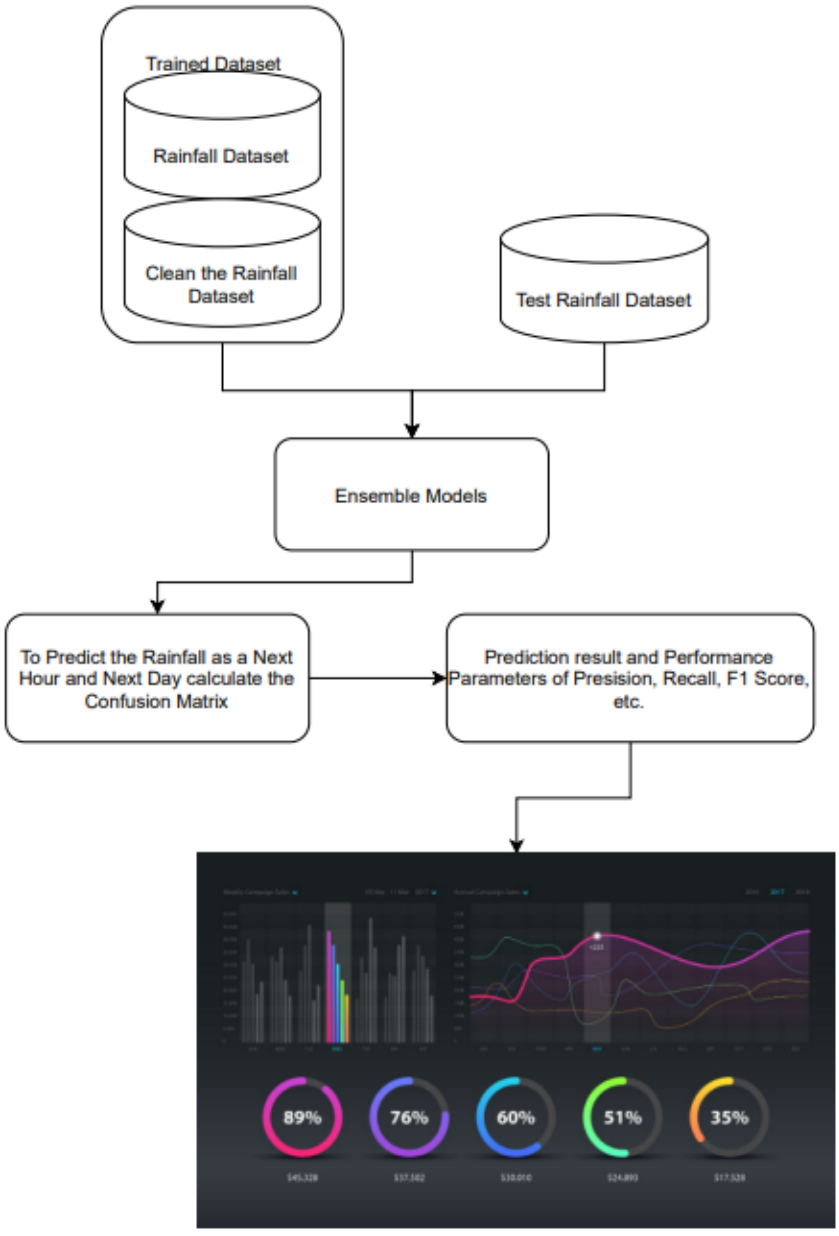
5.1 Data Flow Diagrams

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 requirements graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

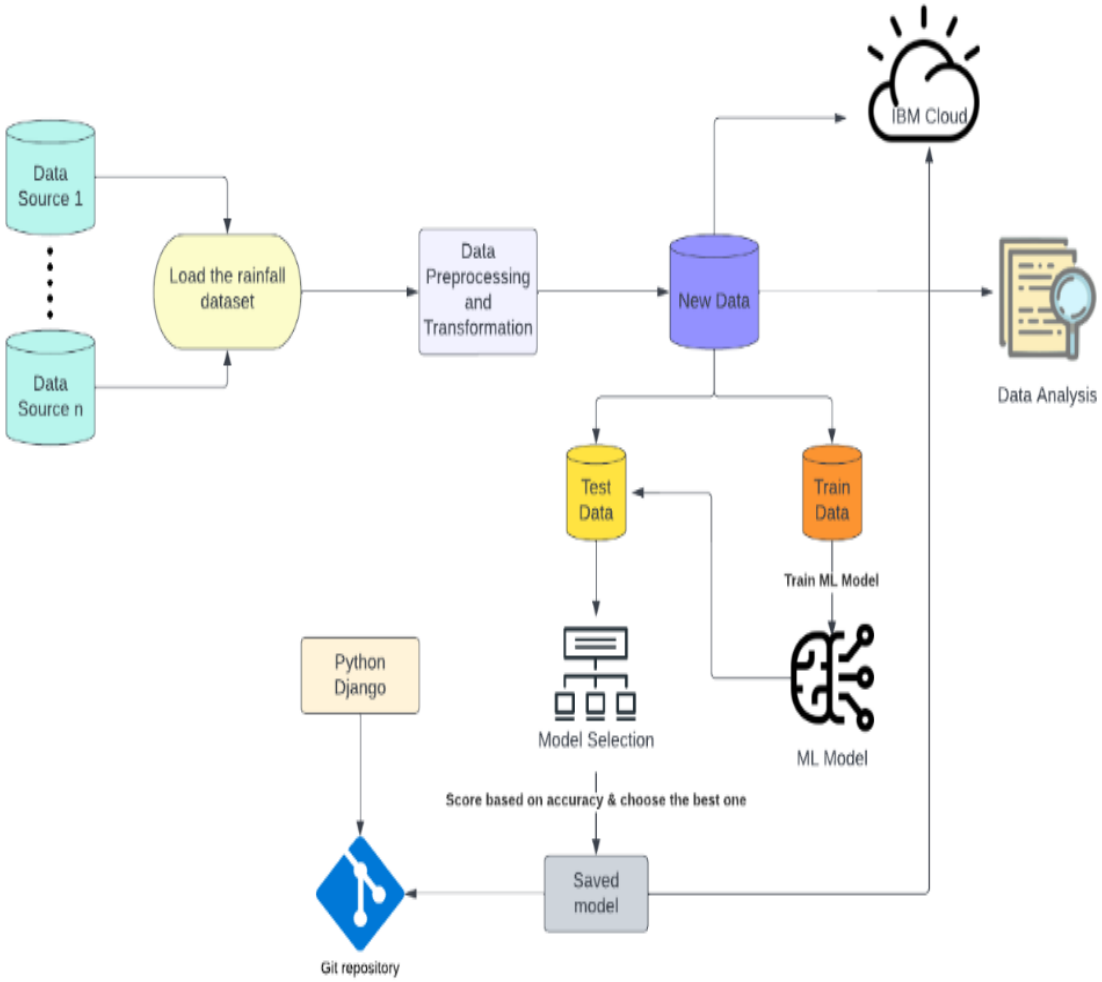


5.2 Solution & Technical Architecture

Solution Architecture:



TECHNICAL ARCHITECTURE:



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Farmer)	Registration	USN-1	As a user, I can sign up by entering my email address, and password, and then confirm my password.	I can create my account	High	Sprint-1
		USN-2	As a user, I can register through Gmail	I can create my account through Gmail	Medium	Sprint-1
	Login	USN-3	As a user, I can log in by entering email & password	I can access my account	High	Sprint-1
	Prediction	USN-4	As a user, I can access the rainfall status in a specific location and season	The prediction of rainfall data can be viewed	High	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Visualization	USN-5	As a user, I can view visualized data based on the predictions and analysis	I can see predicted data in graphical and plot form	High	Sprint-2
	Suggestion	USN-6	As a user, I can see the patterns and appropriate suggestions	The suggestion report can be used according to my requirements on crop production	High	Sprint-3

Chapter 6: PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation

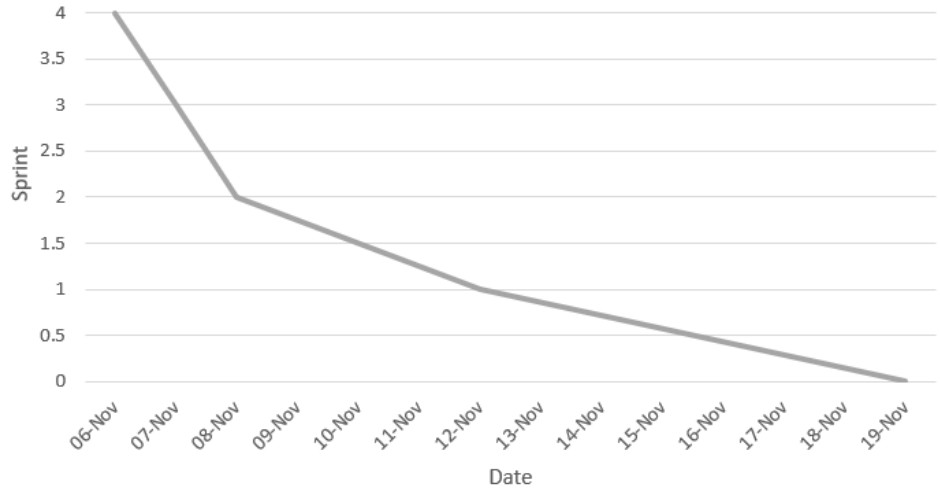
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Working with Dataset	USN-1	To download and import the dataset along with necessary libraries	5	High	Sai Spoorti N
		USN-2	To analyze the data and handle missing data	5	Low	Shivani R
		USN-3	To perform data visualization and find dependent and independent features	5	Medium	Shobana G
		USN-4	To perform feature scaling and split dataset into train and test	5	High	Sakthi G
Sprint-2	Model Decision	USN-5	To train and test different models and find the accuracy	10	Medium	Sakthi G Shobana G
		USN-6	To perform model evaluation based on different evaluation metrics and saving the model	10	Medium	Sai Spoorti N Shivani R
Sprint-3	Building web application	USN-7	Creating a web page for content display	10	High	Sai Spoorti N Sakthi G
		USN-8	Creating python script for prediction and rendering to web page	10	High	Shivani R Shobana G
Sprint-4	Export Application	USN-9	Run and export the application	20	High	Sai Spoorti N, Sakthi G, Shivani R, Shobana G

6.2 Sprint Delivery Schedule

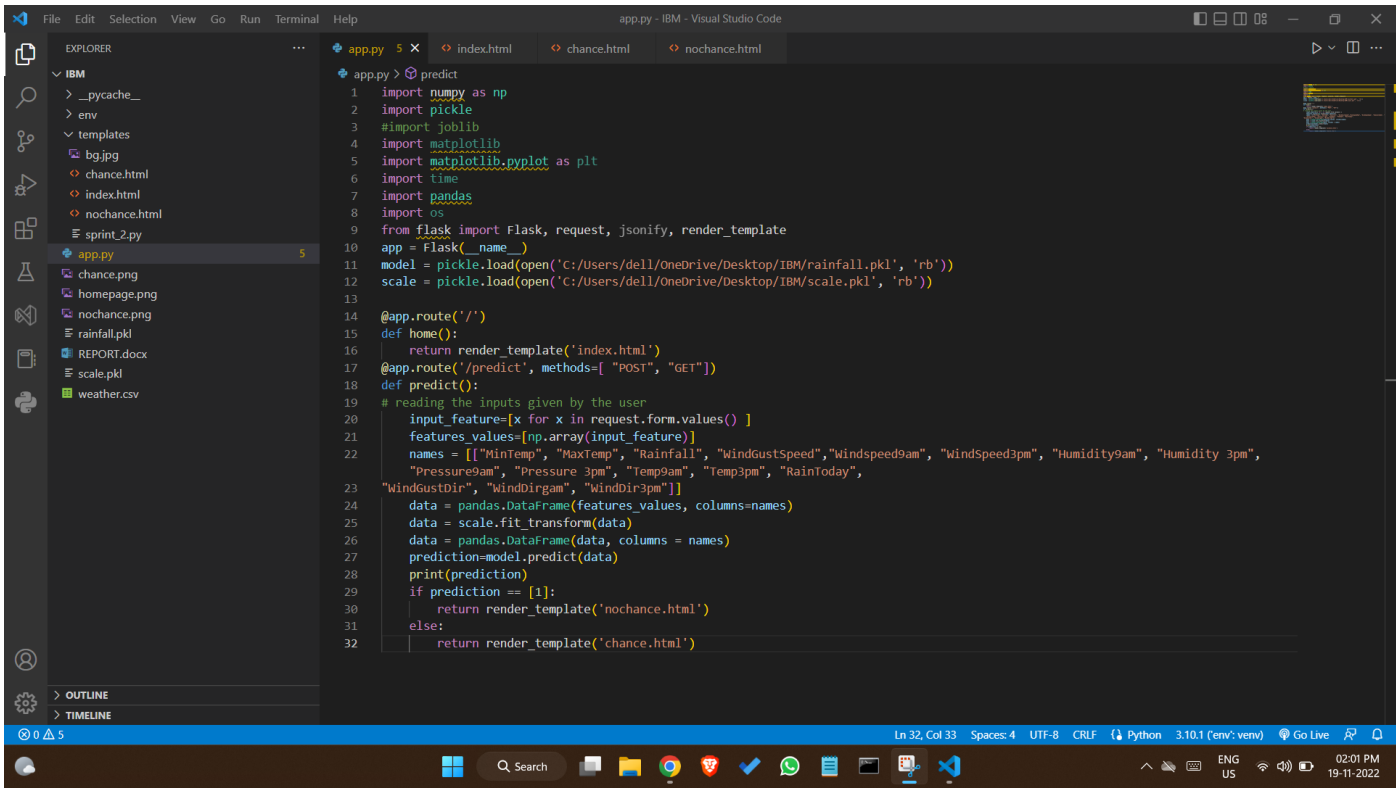
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	5 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

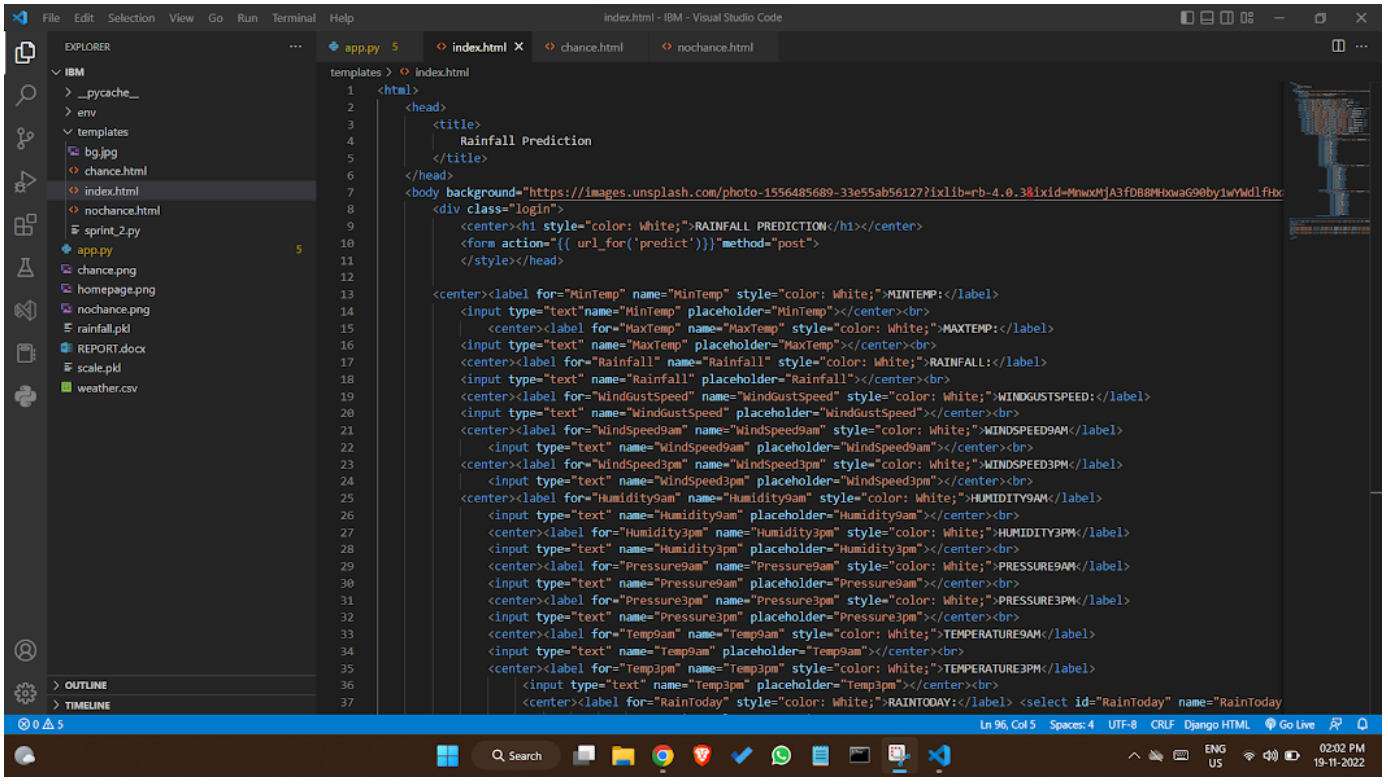
Velocity:

Imagine we have a 10-day sprint duration. The velocity of the team is 20 points per sprint. Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

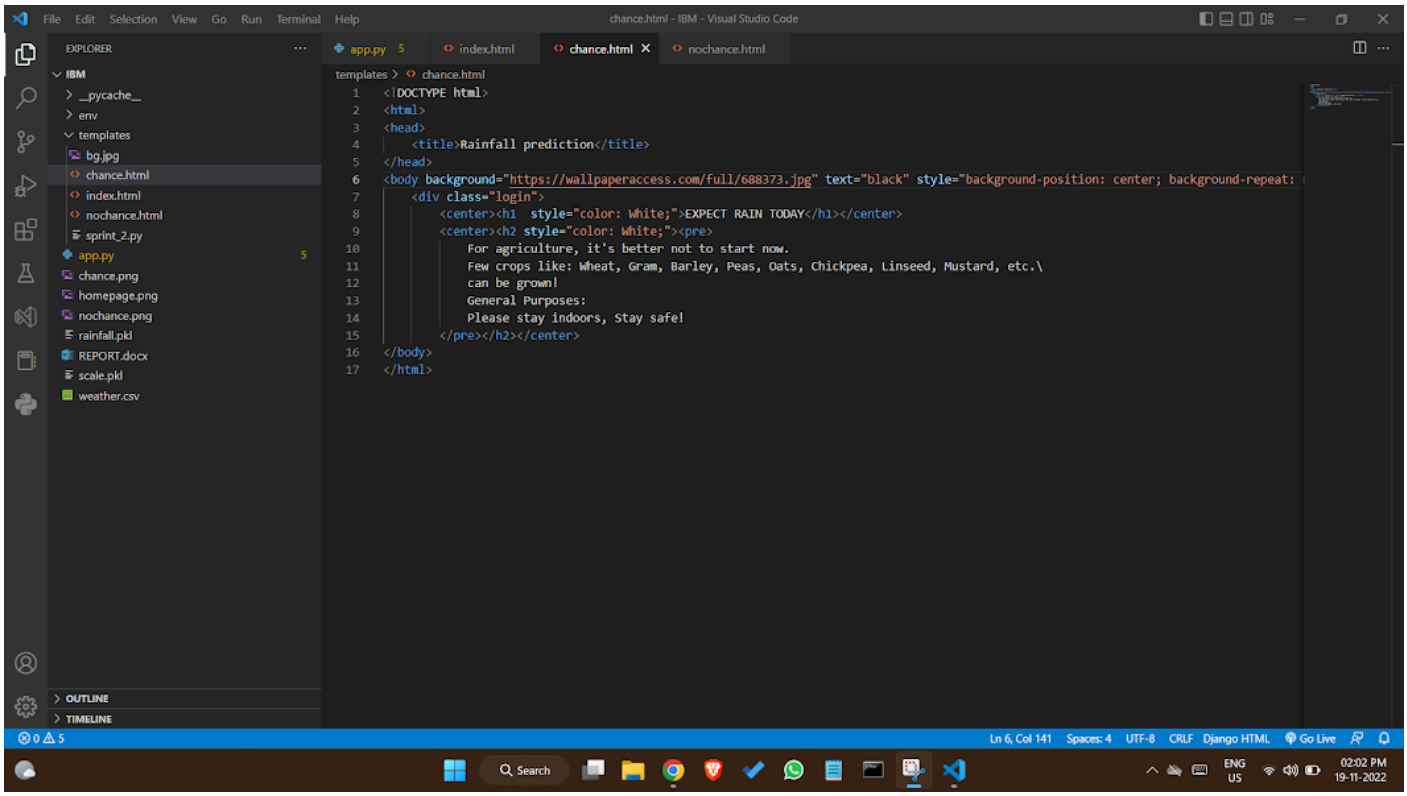


Chapter 7: CODING & SOLUTIONING

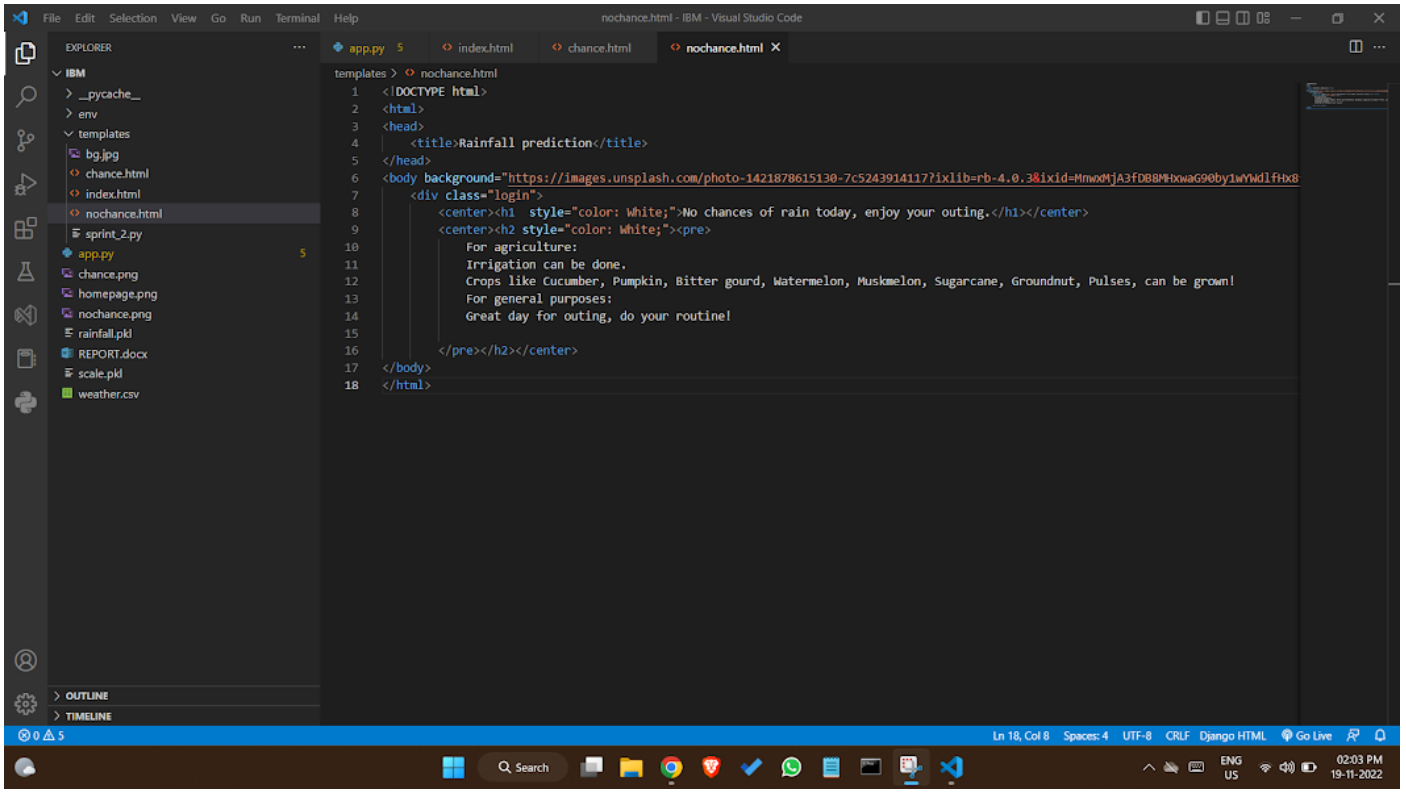

File Edit Selection View Go Run Terminal Help
app.py - IBM - Visual Studio Code
EXPLORER
IBM
__pycache__
env
templates
bg.jpg
chance.html
index.html
nochance.html
sprint_2.py
app.py 5
chance.png
homepage.png
nochance.png
rainfall.pkl
REPORT.docx
scale.pkl
weather.csv
app.py 5
index.html
chance.html
nochance.html
app.py > predict
1 import numpy as np
2 import pickle
3 #import joblib
4 import matplotlib
5 import matplotlib.pyplot as plt
6 import time
7 import pandas
8 import os
9 from flask import Flask, request, jsonify, render_template
10 app = Flask(__name__)
11 model = pickle.load(open('C:/Users/dell/OneDrive/Desktop/IBM/rainfall.pkl', 'rb'))
12 scale = pickle.load(open('C:/Users/dell/OneDrive/Desktop/IBM/scale.pkl', 'rb'))
13
14 @app.route('/')
15 def home():
16 return render_template('index.html')
17 @app.route('/predict', methods=["POST", "GET"])
18 def predict():
19 # reading the inputs given by the user
20 input_feature=[x for x in request.form.values()]
21 features_values=[np.array(input_feature)]
22 names = [["MinTemp", "MaxTemp", "Rainfall", "WindGustSpeed","Windspeed9am", "WindSpeed3pm", "Humidity9am", "Humidity 3pm",
"Pressure9am", "Pressure 3pm", "Temp9am", "Temp3pm", "RainToday",
"WindGustDir", "WindDirgam", "WindDir3pm"]]
23 data = pandas.DataFrame(features_values, columns=names)
24 data = scale.fit_transform(data)
25 data = pandas.DataFrame(data, columns = names)
26 prediction=model.predict(data)
27 print(prediction)
28 if prediction == [1]:
29 return render_template('nochance.html')
30 else:
31 return render_template('chance.html')
32
OUTLINE
TIMELINE
Ln 32, Col 33 Spaces: 4 UTF-8 CRLF Python 3.10.1 ('env': venv) Go Live
Search
ENG US
02:01 PM 19-11-2022


File Edit Selection View Go Run Terminal Help
index.html - IBM - Visual Studio Code
EXPLORER
IBM
templates
bg.jpg
chance.html
index.html
nochance.html
sprint_2.py
app.py 5
chance.png
homepage.png
nochance.png
rainfall.pkl
REPORT.docx
scale.pkl
weather.csv
app.py 5
index.html
chance.html
nochance.html
templates > index.html
<html>
<head>
<title>
Rainfall Prediction
</title>
</head>
<body background="https://images.unsplash.com/photo-1556485689-33e55ab56127?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx

<h1 style="color: White;">RAINFALL PREDICTION</h1>
<form action="{{ url_for('predict')}}"method="post">
</style></head>
<label for="MinTemp" name="MinTemp" style="color: White;">MINTEMP:</label>
<input type="text"name="MinTemp" placeholder="MinTemp">
<label for="MaxTemp" name="MaxTemp" style="color: White;">MAXTEMP:</label>
<input type="text" name="MaxTemp" placeholder="MaxTemp">
<label for="Rainfall" name="Rainfall" style="color: White;">RAINFALL:</label>
<input type="text" name="Rainfall" placeholder="Rainfall">
<label for="WindGustSpeed" name="WindGustSpeed" style="color: White;">WINDGUSTSPEED:</label>
<input type="text" name="WindGustSpeed" placeholder="WindGustSpeed">
<label for="WindSpeed9am" name="WindSpeed9am" style="color: White;">WINDSPEED9AM</label>
<input type="text" name="WindSpeed9am" placeholder="WindSpeed9am">
<label for="WindSpeed3pm" name="WindSpeed3pm" style="color: White;">WINDSPEED3PM</label>
<input type="text" name="WindSpeed3pm" placeholder="WindSpeed3pm">
<label for="Humidity9am" name="Humidity9am" style="color: White;">HUMIDITY9AM</label>
<input type="text" name="Humidity9am" placeholder="Humidity9am">
<label for="Humidity3pm" name="Humidity3pm" style="color: White;">HUMIDITY3PM</label>
<input type="text" name="Humidity3pm" placeholder="Humidity3pm">
<label for="Pressure9am" name="Pressure9am" style="color: White;">PRESSURE9AM</label>
<input type="text" name="Pressure9am" placeholder="Pressure9am">
<label for="Pressure3pm" name="Pressure3pm" style="color: White;">PRESSURE3PM</label>
<input type="text" name="Pressure3pm" placeholder="Pressure3pm">
<label for="Temp9am" name="Temp9am" style="color: White;">TEMPERATURE9AM</label>
<input type="text" name="Temp9am" placeholder="Temp9am">
<label for="Temp3pm" name="Temp3pm" style="color: White;">TEMPERATURE3PM</label>
<input type="text" name="Temp3pm" placeholder="Temp3pm">
<label for="RainToday" style="color: White;">RAINTODAY:</label> <select id="RainToday" name="RainToday
OUTLINE
TIMELINE
Ln 96, Col 5 Spaces: 4 UTF-8 CRLF Django HTML Go Live
Search
ENG US
02:02 PM 19-11-2022


File Edit Selection View Go Run Terminal Help
chance.html - IBM - Visual Studio Code
EXPLORER
IBM
__pycache__
env
templates
bg.jpg
chance.html
index.html
nochance.html
sprint_2.py
app.py 5
chance.png
homepage.png
nochance.png
rainfall.pkl
REPORT.docx
scale.pkl
weather.csv
app.py 5
index.html
chance.html
nochance.html
templates > chance.html
<!DOCTYPE html>
<html>
<head>
<title>Rainfall prediction</title>
</head>
<body background="https://wallpaperaccess.com/full/688373.jpg" text="black" style="background-position: center; background-repeat:

<h1 style="color: White;">EXPECT RAIN TODAY</h1>
<h2 style="color: White;"><pre>
For agriculture, it's better not to start now.
Few crops like: Wheat, Gram, Barley, Peas, Oats, Chickpea, Linseed, Mustard, etc.\
can be grown!
General Purposes:
Please stay indoors, Stay safe!
</pre></h2>
</body>
</html>
OUTLINE
TIMELINE
Ln 6, Col 141 Spaces: 4 UTF-8 CRLF Django HTML Go Live
Search
ENG US
02:02 PM 19-11-2022


File Edit Selection View Go Run Terminal Help
nochance.html - IBM - Visual Studio Code
EXPLORER
IBM
__pycache__
env
templates
bg.jpg
chance.html
index.html
nochance.html
sprint_2.py
app.py 5
chance.png
homepage.png
nochance.png
rainfall.pkl
REPORT.docx
scale.pkl
weather.csv
app.py 5
index.html
chance.html
nochance.html
templates > nochance.html
<!DOCTYPE html>
<html>
<head>
<title>Rainfall prediction</title>
</head>
<body background="https://images.unsplash.com/photo-1421878615130-7c5243914117?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8

<h1 style="color: White;">No chances of rain today, enjoy your outing.</h1>
<h2 style="color: White;"><pre>
For agriculture:
Irrigation can be done.
Crops like Cucumber, Pumpkin, Bitter gourd, Watermelon, Muskmelon, Sugarcane, Groundnut, Pulses, can be grown!
For general purposes:
Great day for outing, do your routine!
</pre></h2>
</body>
</html>
OUTLINE
TIMELINE
Ln 18, Col 8 Spaces: 4 UTF-8 CRLF Django HTML Go Live
Search
ENG US
02:03 PM 19-11-2022

Chapter 8: TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Pre requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
HomePage_TC_OO1	UI	Home Page	User is able to see the form along with UI elements		Enter URL and click go	http://127.0.0.1:5000/	Homepage should be displayed	Working as expected	Pass
Prediction_TC_OO2	Functional	Home Page	User fills the form to view the prediction		1.Enter URL and click go 2.Fill the form 3.Click Predict	[15.9,21.7,2.2,31,15,13,89,91,1010.5,1004.2,15.9,17]	Page showing "Expect Rain Today" along with necessary suggestions	Working as expected	Pass
PredictionPage_TC_OO3	Functional	Home page	User fills the form to view the prediction		1.Enter URL and click go 2.Fill the form 3.Click Predict	[13.4,22.9,0.6,44,20,24,71,22,1007.7,1007.1,16.9,21.8]	Page showing "No Chances of Rain Today" along with necessary suggestions	Working as expected	Pass

8.2 User Acceptance Testing

1. Purpose of Document

The goal of this document is to give a short summary of the Exploratory Analysis of Rainfall Data in India for Agriculture project's test coverage and open issues 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	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

Test Case Analysis

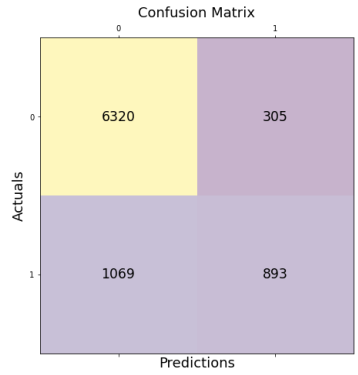
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
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

Chapter 9: RESULTS

9.1 Performance Metrics

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
	Metrics	**Classification Model:** ● Confusion Matrix ● Accuray Score- 0.999796203563526 ● Classification Report	

			Accuracy: 0.9997962035635263 Precession: 0.8003672494261763 Recall: 0.7045550362548805 F1-score: 0.7335725305435974 print(classification_report(y_test, p1)) precision recall f1-score support 0 0.87 0.96 0.91 22067 1 0.77 0.51 0.61 6372 accuracy 0.85 28439 macro avg 0.82 0.73 0.76 28439 weighted avg 0.85 0.85 0.84 28439

Chapter 10: ADVANTAGES & DISADVANTAGES

Advantages:

The proposed solution gives farmers accurate results and good advice on how to make their crops more productive.

Providing this type of assistance is of great help for farmers to plan accordingly and reduce crop loss

Disadvantages:

Results are not capable of providing preventive measures as the scope of the prediction is limited to days rather than months

Suggestions can be tailored to the season but are difficult to customize extensively.

Chapter 11: CONCLUSION

The results show that India has two main rainfall seasons: the southwest monsoon (advancing monsoon) and the northeast monsoon (retreating monsoon). The advancing monsoon contributes almost 80% of the rainfall. Southwest and northeast India receive most of the rainfall during the advancing monsoon. During the retreating monsoon, the Andaman and Nicobar Islands, Kerala, and Tamil Nadu receive more rainfall as compared to other subdivisions.

The trend analysis of annual rainfall considering India as a whole shows a decreasing trend; however, when the trend is analyzed for all subdivisions individually, we can see some divisions showing an increasing trend and some showing a

decreasing trend. It showed that it is important to study subdivisions for better forecasting.

Chapter 12: FUTURE SCOPE

The future enhancements of this project could include a real-time prediction of rainfall integrated with the Internet of Things architecture that could tell about the weather instantly and quickly. This project could also be extended to different modules other than the farmer, and people could gain information from it.

Chapter 13: APPENDIX

13.1 Source Code

index.html

```
<html>
  <head>
    <title>
      Rainfall Prediction
    </title>
  </head>
  <body
background="https://images.unsplash.com/photo-1556485689-33e55ab56127?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=870&q=80" text="black"
style="background-position: center; background-repeat: no-repeat; background-size: cover;">
    <div class="login">
      <center><h1 style="color: White;">RAINFALL PREDICTION</h1></center>
      <form action="{{ url_for('predict')}}"method="post">
      </style></head>

    <center><label for="MinTemp" name="MinTemp" style="color: White;">MINTEMP:</label>
      <input type="text"name="MinTemp" placeholder="MinTemp"></center><br>
        <center><label for="MaxTemp" name="MaxTemp" style="color: White;">MAXTEMP:</label>
      <input type="text" name="MaxTemp" placeholder="MaxTemp"></center><br>
      <center><label for="Rainfall" name="Rainfall" style="color: White;">RAINFALL:</label>
      <input type="text" name="Rainfall" placeholder="Rainfall"></center><br>
      <center><label for="WindGustSpeed" name="WindGustSpeed" style="color:
White;">WINDGUSTSPEED:</label>
      <input type="text" name="WindGustSpeed" placeholder="WindGustSpeed"></center><br>
      <center><label for="WindSpeed9am" name="WindSpeed9am" style="color:
White;">WINDSPEED9AM</label>
```

```
            <input type="text" name="WindSpeed9am" placeholder="WindSpeed9am"></center><br>
        <center><label for="WindSpeed3pm" name="WindSpeed3pm" style="color:
White;">WINDSPEED3PM</label>
            <input type="text" name="WindSpeed3pm" placeholder="WindSpeed3pm"></center><br>
        <center><label for="Humidity9am" name="Humidity9am" style="color:
White;">HUMIDITY9AM</label>
            <input type="text" name="Humidity9am" placeholder="Humidity9am"></center><br>
            <center><label for="Humidity3pm" name="Humidity3pm" style="color:
White;">HUMIDITY3PM</label>
            <input type="text" name="Humidity3pm" placeholder="Humidity3pm"></center><br>
            <center><label for="Pressure9am" name="Pressure9am" style="color:
White;">PRESSURE9AM</label>
                        <input type="text" name="Pressure9am" placeholder="Pressure9am"></center><br>
            <center><label for="Pressure3pm" name="Pressure3pm" style="color:
White;">PRESSURE3PM</label>
                        <input type="text" name="Pressure3pm" placeholder="Pressure3pm"></center><br>
            <center><label for="Temp9am" name="Temp9am" style="color:
White;">TEMPERATURE9AM</label>
                        <input type="text" name="Temp9am" placeholder="Temp9am"></center><br>
            <center><label for="Temp3pm" name="Temp3pm" style="color:
White;">TEMPERATURE3PM</label>
                <input type="text" name="Temp3pm" placeholder="Temp3pm"></center><br>
                <center><label for="RainToday" style="color: White;">RAINTODAY:</label> <select
id="RainToday" name="RainToday">
                                <option value=0>No<option>
                                <option value=1>Yes<option>
                                </select>  </center><br>
                                <center><label for="WindGustDir" style="color:
White;">WINDGUSTDIR:</label> <select id="WindGustDir" name="WindGustDir">
                                <option value=13>W<option>
                                <option value=14>WNW<option>
                                <option value=15>WSW<option>
                                <option value=4>NE<option>
                                <option value=6>NNW<option>
                                <option value=3>N<option>
                                <option value=5>NNE<option>
                                <option value=12>SW<option>
                                <option value=1>ENE<option>
                                <option value=10>SSE<option>
```

<option value=8>S<option>
<option value=7>NW<option>
<option value=9>SE<option>
<option value=2>ESE<option>
<option value=0>E<option>
<option value=11>SSW<option>
</select>
<label for="WindDir9am" style="color: White;">WINDDIR9AM:</label>
<select id="WindDir9am" name="WindDir9am">
<option value=13>W<option>
<option value=14>WNW<option>
<option value=15>WSW<option>
<option value=4>NE<option>
<option value=6>NNW<option>
<option value=3>N<option>
<option value=5>NNE<option>
<option value=12>SW<option>
<option value=1>ENE<option>
<option value=10>SSE<option>
<option value=8>S<option>
<option value=7>NW<option>
<option value=9>SE<option>
<option value=2>ESE<option>
<option value=0>E<option>
<option value=11>SSW<option>
</select>
<label for="WindDir3pm" style="color: White;">WINDDIR3PM:</label>
<select id="WindDir3pm" name="WindDir3pm">
<option value=13>W<option>
<option value=14>WNW<option>
<option value=15>WSW<option>
<option value=4>NE<option>
<option value=6>NNW<option>
<option value=3>N<option>
<option value=5>NNE<option>
<option value=12>SW<option>
<option value=1>ENE<option>
<option value=10>SSE<option>

```
                                        <option value=8>S<option>
                                        <option value=7>NW<option>
                                        <option value=9>SE<option>
                                        <option value=2>ESE<option>
                                        <option value=0>E<option>
                                        <option value=11>SSW<option>
                                </select>  </center><br>

<br>

<center><button type="submit" class="btn btn-primary btn-block btn-large"
style="height:30px;width:200px">PREDICT</button></center>
</form> <br>
{{ prediction_text }}
<br> <br>
<img src=" {{url_3}}" alt="Submit Form" height="180" width="233"
onerror="this.style.display='none'"/>
<img src=" {{url_1}}" alt="Submit Form" height="180" width="233"
onerror="this.style.display='none'"/>
<img src="{{url_4}}" alt="Submit Form" height="180" width="233"
onerror="this.style.display='none'"/>
<br> <br>
<img src="{{ur1_2}}" alt="Submit Form" height="150" width="711"
onerror="this.style.display='none'"/>
</div>

   </body>
</html>
```

Chance.html

```
<!DOCTYPE html>

<html>

<head>

   <title>Rainfall prediction</title>

</head>
```

```
<body background="https://wallpaperaccess.com/full/688373.jpg" text="black" style="background-position: center; background-repeat: no-repeat">

   <div class="login">

      <center><h1  style="color: White;">EXPECT RAIN TODAY</h1></center>

      <center><h2 style="color: White;"><pre>

         For agriculture, it's better not to start now.

         Few crops like: Wheat, Gram, Barley, Peas, Oats, Chickpea, Linseed, Mustard, etc.\

         can be grown!

         General Purposes:

         Please stay indoors, Stay safe!

      </pre></h2></center>

</body>

</html>
```

nochance.html

```
<!DOCTYPE html>

<html>

<head>

   <title>Rainfall prediction</title>

</head>

<body
background="https://images.unsplash.com/photo-1421878615130-7c5243914117?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=869&q=80"
text="black" style="background-position: center; background-repeat: no-repeat;  background-size: cover;">

   <div class="login">

      <center><h1  style="color: White;">No chances of rain today, enjoy your outing.</h1></center>

      <center><h2 style="color: White;"><pre>

         For agriculture:

         Irrigation can be done.

         Crops like Cucumber, Pumpkin, Bitter gourd, Watermelon, Muskmelon, Sugarcane, Groundnut, Pulses, can be grown!

         For general purposes:

         Great day for outing, do your routine!



      </pre></h2></center>

</body>

</html>
```

app.py

```
import numpy as np

import pickle

#import joblib

import matplotlib

import matplotlib.pyplot as plt

import time

import pandas

import os

from flask import Flask, request, jsonify, render_template

app = Flask(_name_)

model = pickle.load(open('C:/Users/dell/OneDrive/Desktop/IBM/rainfall.pkl', 'rb'))

scale = pickle.load(open('C:/Users/dell/OneDrive/Desktop/IBM/scale.pkl', 'rb'))


@app.route('/')

def home():

    return render_template('index.html')

@app.route('/predict', methods=[ "POST", "GET"])

def predict():

# reading the inputs given by the user

    input_feature=[x for x in request.form.values() ]

    features_values=[np.array(input_feature)]

    names = [["MinTemp", "MaxTemp", "Rainfall", "WindGustSpeed","Windspeed9am", "WindSpeed3pm",
"Humidity9am", "Humidity 3pm", "Pressure9am", "Pressure 3pm", "Temp9am", "Temp3pm", "RainToday",

"WindGustDir", "WindDirgam", "WindDir3pm"]]
```

```
    data = pandas.DataFrame(features_values, columns=names)
    data = scale.fit_transform(data)
    data = pandas.DataFrame(data, columns = names)
    prediction=model.predict(data)
    print(prediction)
    if prediction == [1]:
        return render_template('nochance.html')
    else:
        return render_template('chance.html')
```

13.2 GitHub Link

GitHub link:

https://github.com/IBM-EPBL/IBM-Project-23275-1659875881