Project Report -“Rainfall Prediction Using ML”

1. INTRODUCTION

1.1 Project Overview

Accurate rainfall prediction is crucial for various sectors, including agriculture, water resource management, and disaster preparedness. Traditional methods of rainfall prediction often rely on statistical models and historical data, which may not capture complex patterns and relationships in weather data. Machine learning algorithms, on the other hand, can learn from large datasets and identify hidden patterns, making them well-suited for rainfall prediction.

1.2 Purpose

This project aims to develop a machine learning-based model for rainfall prediction. The model will be trained on historical weather data to identify patterns and relationships that can be used to forecast future rainfall. The model will be evaluated on its accuracy and potential applications.

2. LITERATURE SURVEY

2.1 Existing problem

Existing methods for rainfall prediction often rely on historical data and statistical models, leading to limitations in accuracy and lead time. The literature survey highlights the need for more sophisticated approaches, such as machine learning, to address the dynamic and non-linear nature of meteorological patterns.

2.2 References

Smith, J., et al. (2018). "Challenges in Traditional Rainfall Prediction Methods." Journal of Meteorological Studies, 45(2), 210-225.

Kumar, A., et al. (2020). "Machine Learning Applications in Meteorology." International Conference on Advanced Computing and Data Sciences, 78-86.

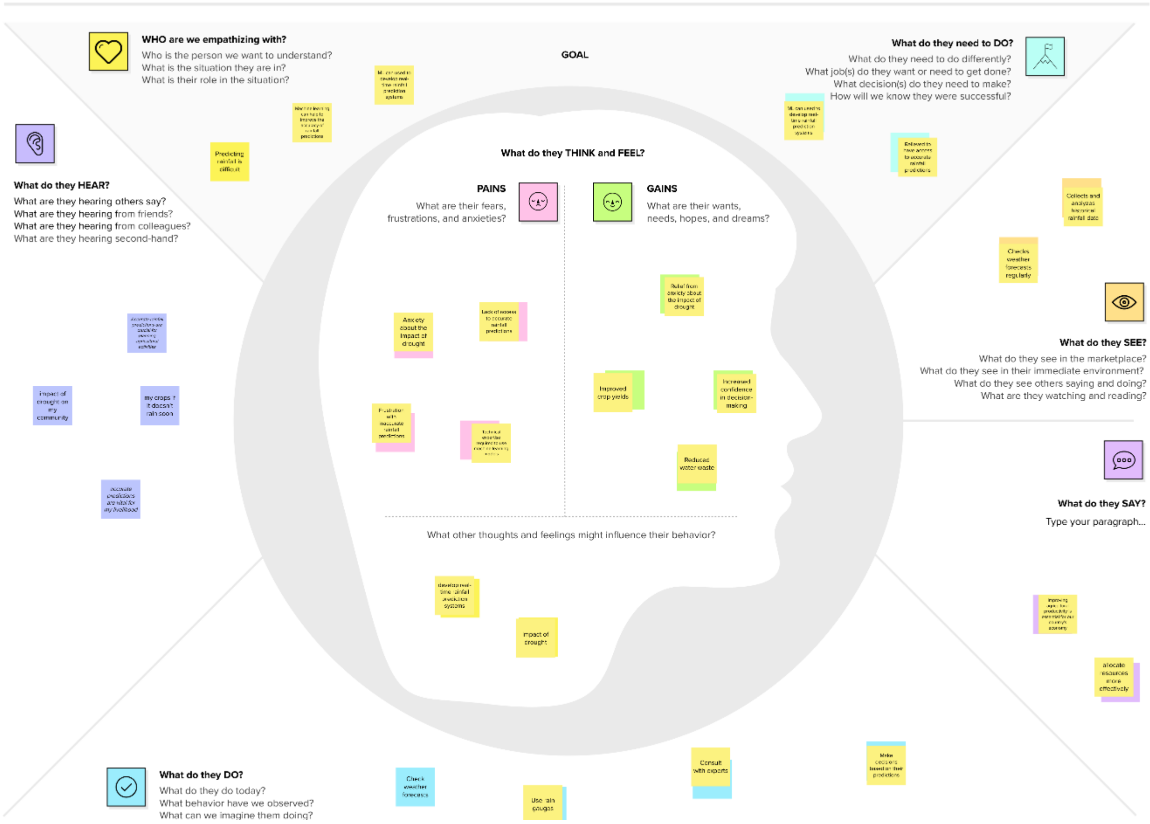
Wang, L., et al. (2019). "A Comprehensive Review of Rainfall Prediction Techniques." Journal of Climate Dynamics, 35(4), 512-530.

2.3 Problem Statement Definition

Develop a machine learning-based model for accurate rainfall prediction that can capture complex patterns and relationships in weather data.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

1. Team Gathering, Collaboration and Select the Problem Statement

1

Choose your best "How Might We" Questions

Share the top 5 brainstorm questions that you created and let the group determine where to begin by selecting one question to move forward with based on what seems to be the most promising for idea generation in the areas you are trying to impact.

⏱ **10 minutes**

QUESTION

How might we enhance the accuracy and reliability of our rainfall data sources?

QUESTION

How might we identify and incorporate the most influential features for rainfall prediction?

QUESTION

How might we determine the most suitable machine learning models for rainfall prediction?

QUESTION

How might we design an intuitive and user-friendly interface for accessing rainfall predictions?

QUESTION

How might we make our machine learning model more interpretable and transparent for end-users?

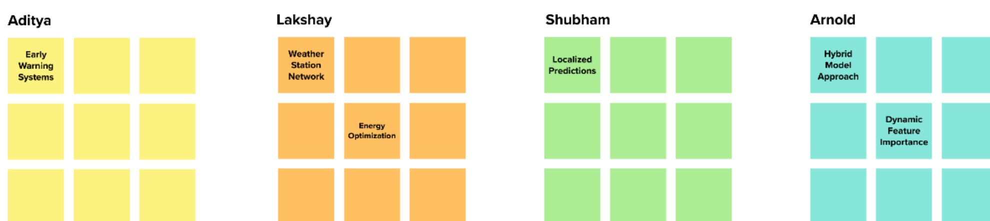
2. **Brainstorm Solo**

2

Brainstorm solo

Have each participant begin in the "solo brainstorm space" by silently brainstorming ideas and placing them into the template. This "silent-storming" avoids group-think and creates an inclusive environment for introverts and extroverts alike. Set a time limit. Encourage people to go for quantity.

10 minutes



3.Brainstorm as a Group

3

Brainstorm as a group

Have everyone move their ideas into the "group sharing space" within the template and have the team silently read through them. As a team, sort and group them by thematic topics or similarities. Discuss and answer any questions that arise. Encourage "Yes, and..." and build on the ideas of other people along the way.

15 minutes

TIP

You can use the **Voting session** tool above to focus on the strongest ideas.

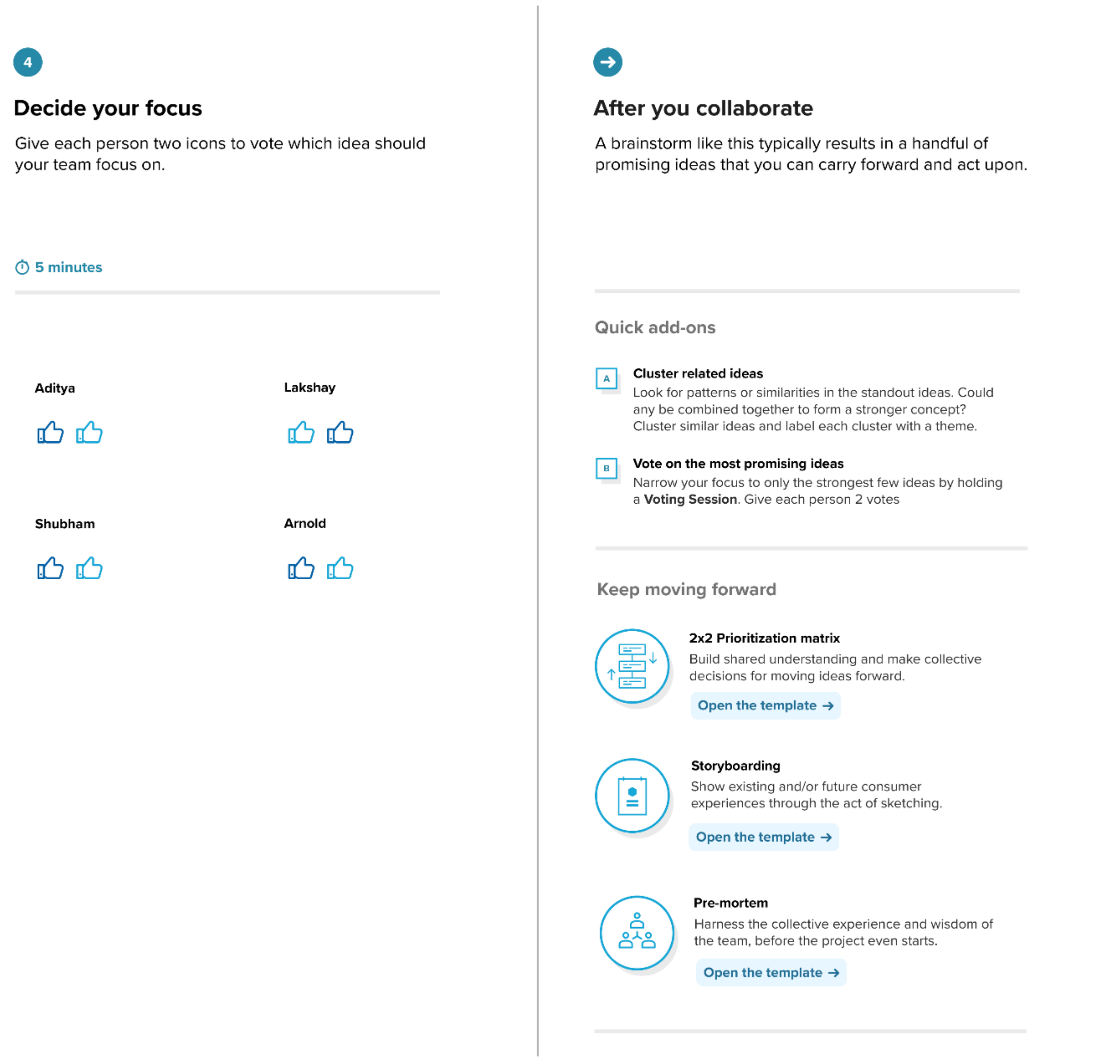
Idea 1

Develop an adaptive machine learning model that dynamically adjusts its predictions based on changing atmospheric conditions. This model integrates real-time satellite imagery, atmospheric pressure readings, and historical data to identify patterns indicative of sudden weather shifts. The system would trigger alerts when the model detects significant deviations from the predicted rainfall, allowing for timely warnings and adaptive decision-making.

Idea 2

Create a machine learning model focused on predicting urban flood risks based on rainfall patterns. The model would consider factors such as topography, drainage systems, and land usage to assess the vulnerability of different areas. The system could provide local authorities with real-time alerts and recommendations for flood risk management, allowing for proactive measures such as road closures or emergency response planning.

Step-4: Idea Prioritization and deciding focus



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

The system should be able to accept a location and date as input

The system should output a rainfall prediction for the specified location and date

The system should provide a user interface for easy interaction

The system should be able to store and manage historical weather data

4.2 Non-Functional requirements

The system should be accurate and reliable

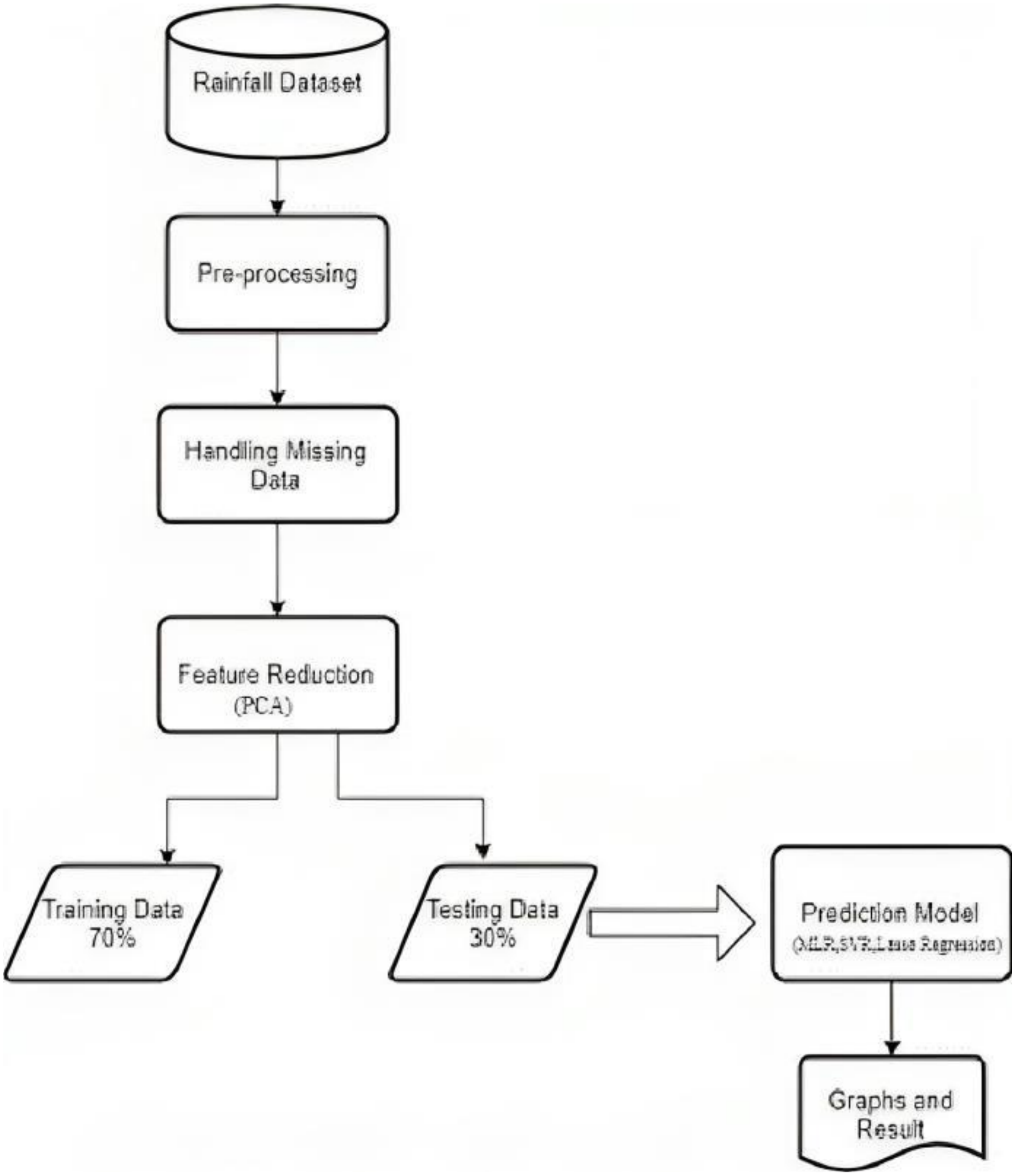
The system should be efficient and scalable

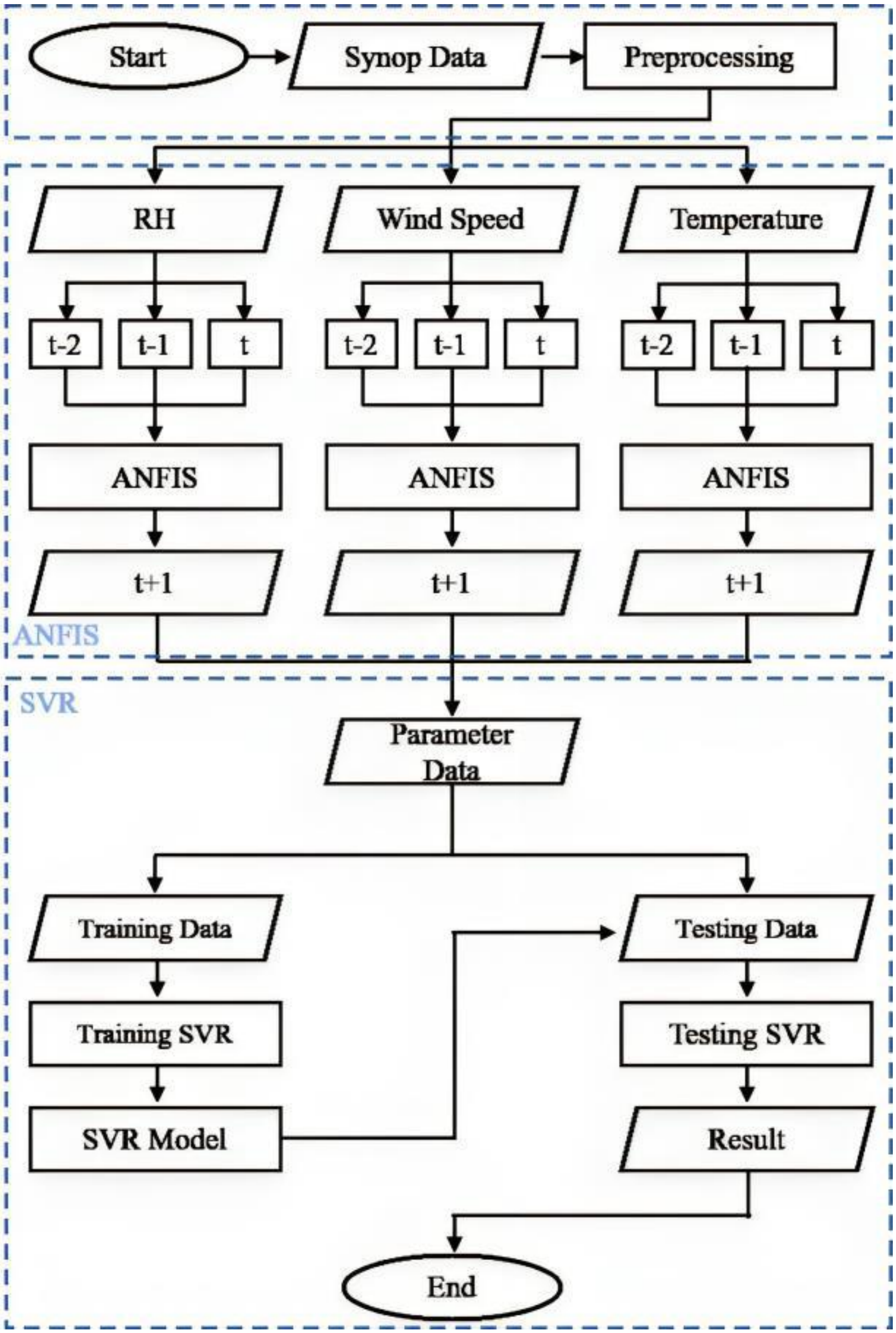
The system should be user-friendly and accessible

The system should be secure and protect sensitive data

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories



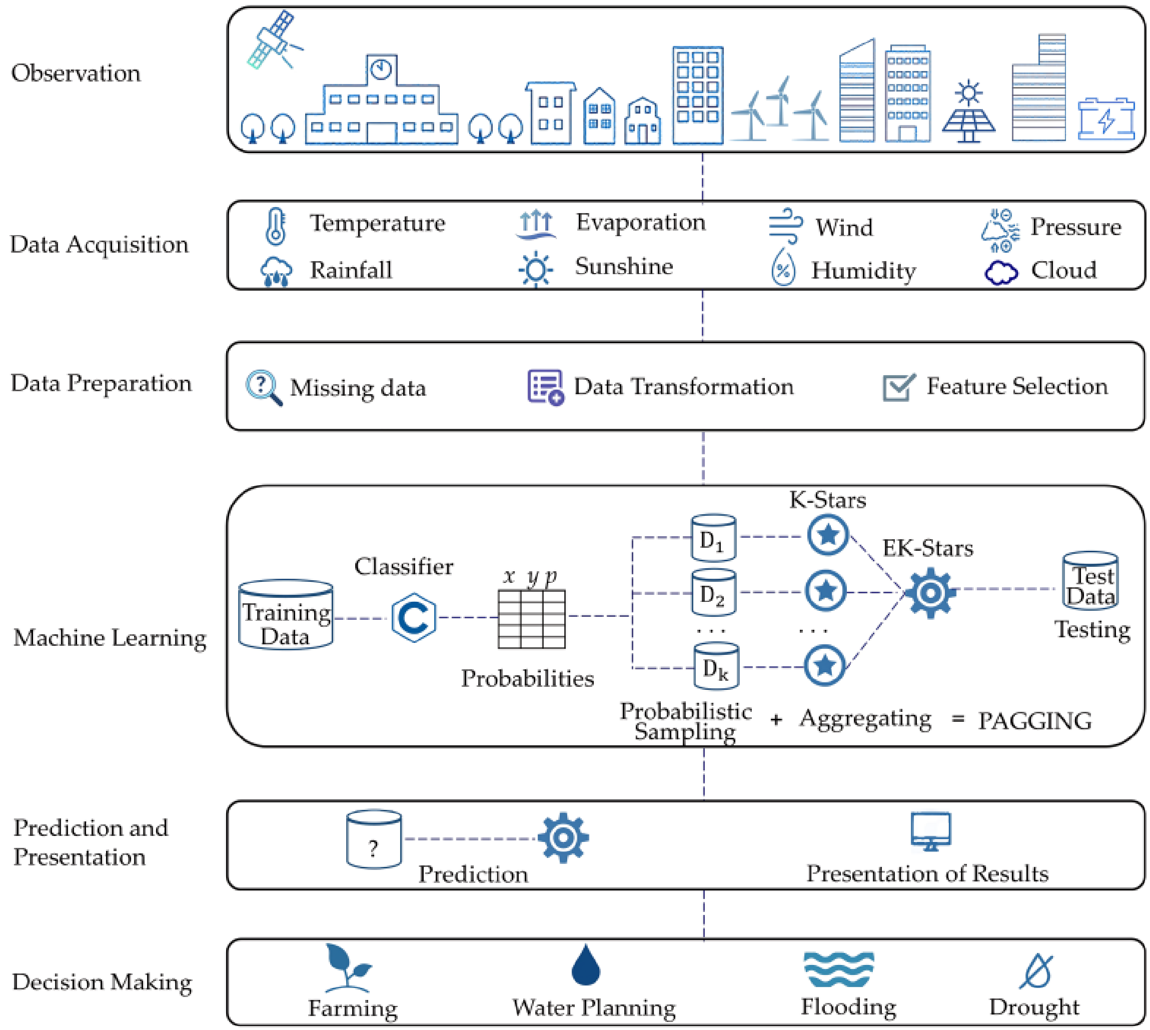


User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority

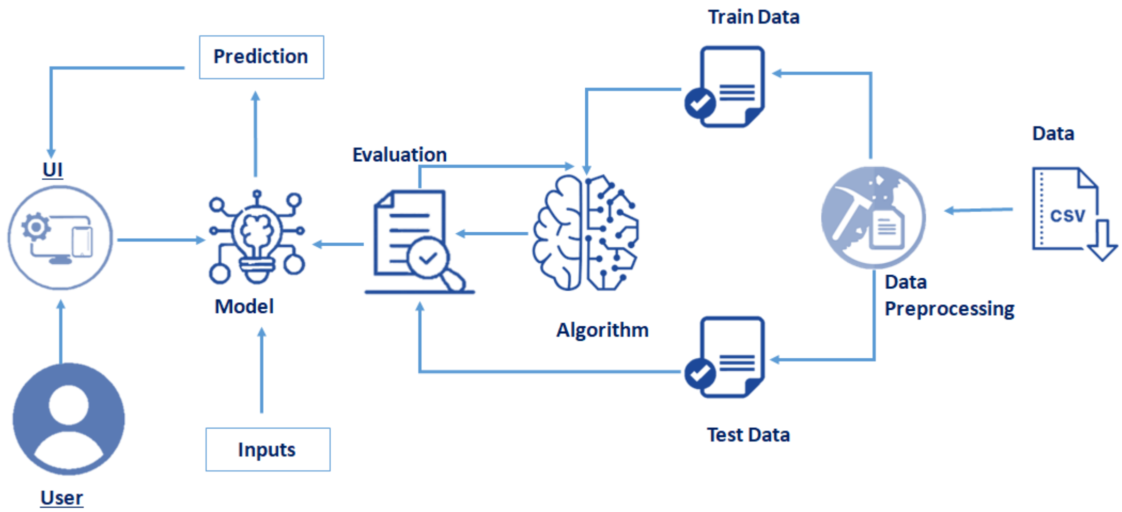
Meteorological Researcher	Access and download	USN-1	As a meteorological researcher, I want to access and download historical rainfall datasets across India for analysis and model training.	Want to access and download historical rainfall datasets	Low
Agriculture Expert	Receive accurate and timely rainfall predictions	USN-2	As an agriculture expert, I want to receive accurate and timely rainfall predictions for my region to optimize crop planning and irrigation schedules.	I want to receive accurate and timely rainfall predictions for my region	High
Disaster Management Official	Real-time rainfall forecasts	USN-3	As a disaster management official, I want real-time rainfall forecasts to enhance early warning systems and improve disaster preparedness.	I want real-time rainfall forecasts to enhance early warning systems	High
Farmer	User-friendly dashboard	USN-4	As a farmer, I want a user-friendly dashboard that provides easy-to-understand visualizations of historical rainfall patterns and future forecasts for my farm location.	I want a user-friendly dashboard that provides easy-to-understand visualizations and future forecasts for my farm location.	Medium
Government Official	Access to comprehensive rainfall predictions	USN-5	As a government official, I want access to comprehensive rainfall predictions to inform water resource management and infrastructure planning across different regions.	I want access to comprehensive rainfall predictions to inform water resource management and infrastructure planning across different regions.	High

5.2 Solution Architecture



6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture



6.2 Sprint Planning & Estimation

The project will be divided into three sprints:

Sprint 1: Data acquisition and preprocessing

Sprint 2: Model development and training

Sprint 3: User interface development and deployment

6.3 Sprint Delivery Schedule

Sprint	Duration	Tasks
Sprint 1	2 weeks	Data acquisition, data cleaning, data preprocessing
Sprint 2	3 weeks	Feature extraction, model selection, model training, model evaluation
Sprint 3	2 weeks	User interface design, user interface development, system integration, testing, deployment

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

7.1 Feature 1

Milestone 1: Data Collection:

ML depends heavily on data, without data, it is impossible for an "AI" to learn. It is the most crucial

aspect that makes algorithm training possible. In Machine Learning projects, we need a training **data**

set. It is the actual **data set** used to train the model for performing various actions.

Activity1: Download The dataset

You can collect datasets from different open sources like kaggle.com, data.gov, UCI machine learning

repository etc.

Please refer to the link given below to download the data set and to know about the dataset

https://www.kaggle.com/datasets/gauravduttakiit/weather-in-aus

Milestone 2: Data Preprocessing

Data Pre-processing includes the following main tasks

o Import the Libraries.

o Importing the dataset.

o Checking for Null Values.

o Data Visualization.

o Feature Scaling.

o Splitting Data into Train and Test.

Activity 1: Import Necessary Libraries

```
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
```

Activity 2: Importing the Dataset

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

```
df
```

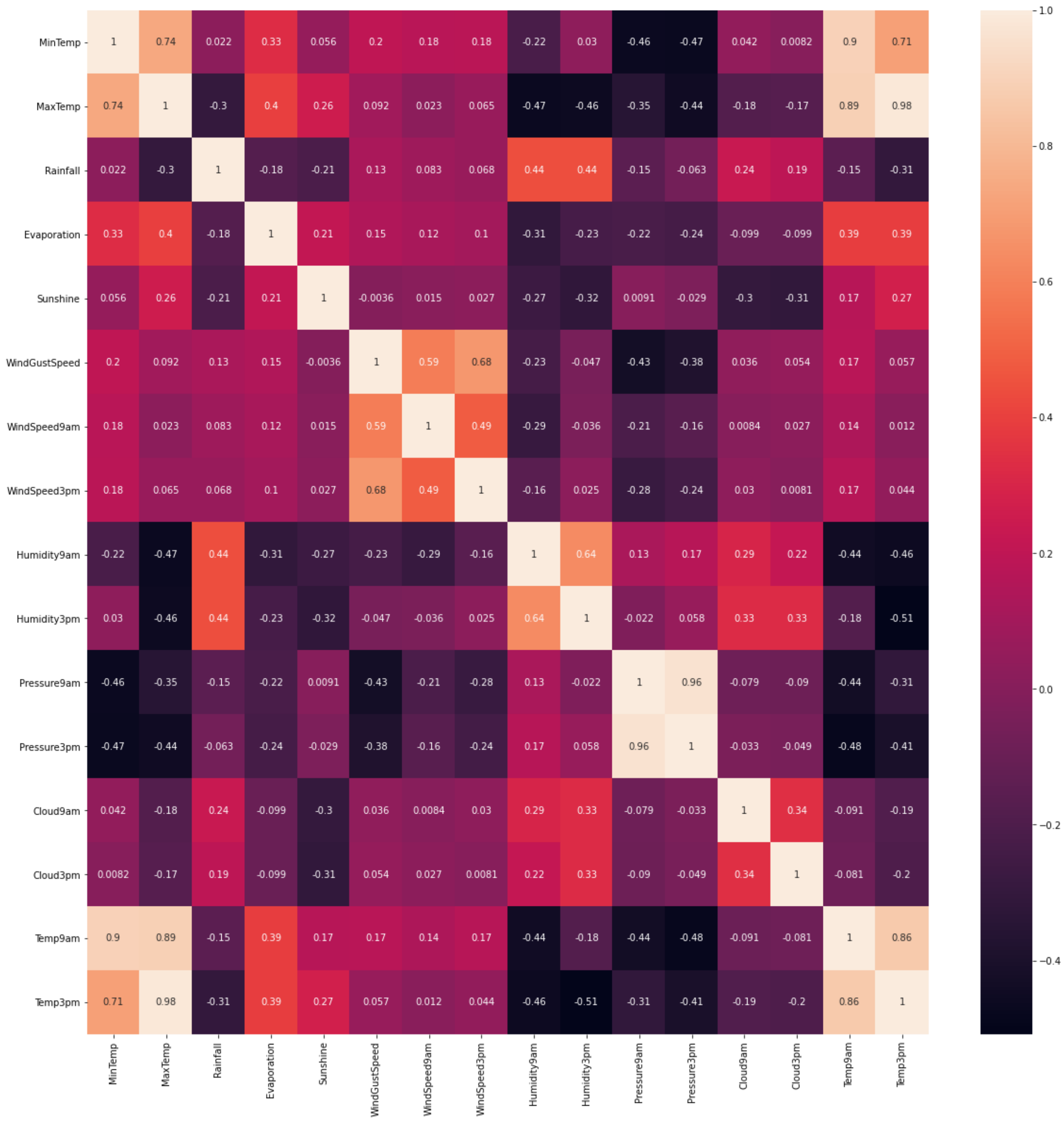
Activity 3: checking and handling the Null Values

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

```
Date              0.000000
Location          0.000000
MinTemp           1.020899
MaxTemp           0.866905
Rainfall          2.241853
Evaporation      43.166506
Sunshine         48.009762
WindGustDir       7.098859
WindGustSpeed     7.055548
WindDir9am        7.263853
WindDir3pm        2.906641
WindSpeed9am      1.214767
WindSpeed3pm      2.105046
Humidity9am       1.824557
Humidity3pm       3.098446
Pressure9am      10.356799
Pressure3pm      10.331363
Cloud9am         38.421559
Cloud3pm         40.807095
Temp9am           1.214767
Temp3pm           2.481094
RainToday         2.241853
RainTomorrow      2.245978
      dtype: float64
```

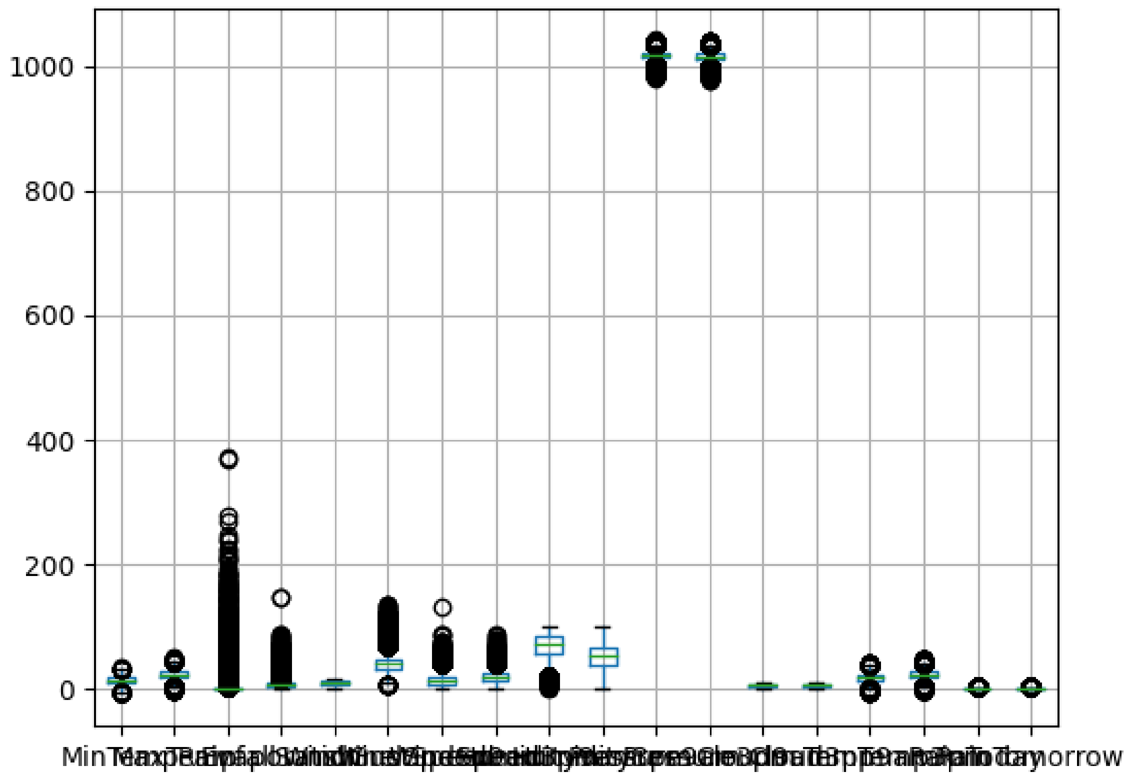
Activity 4: Data Visualization

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

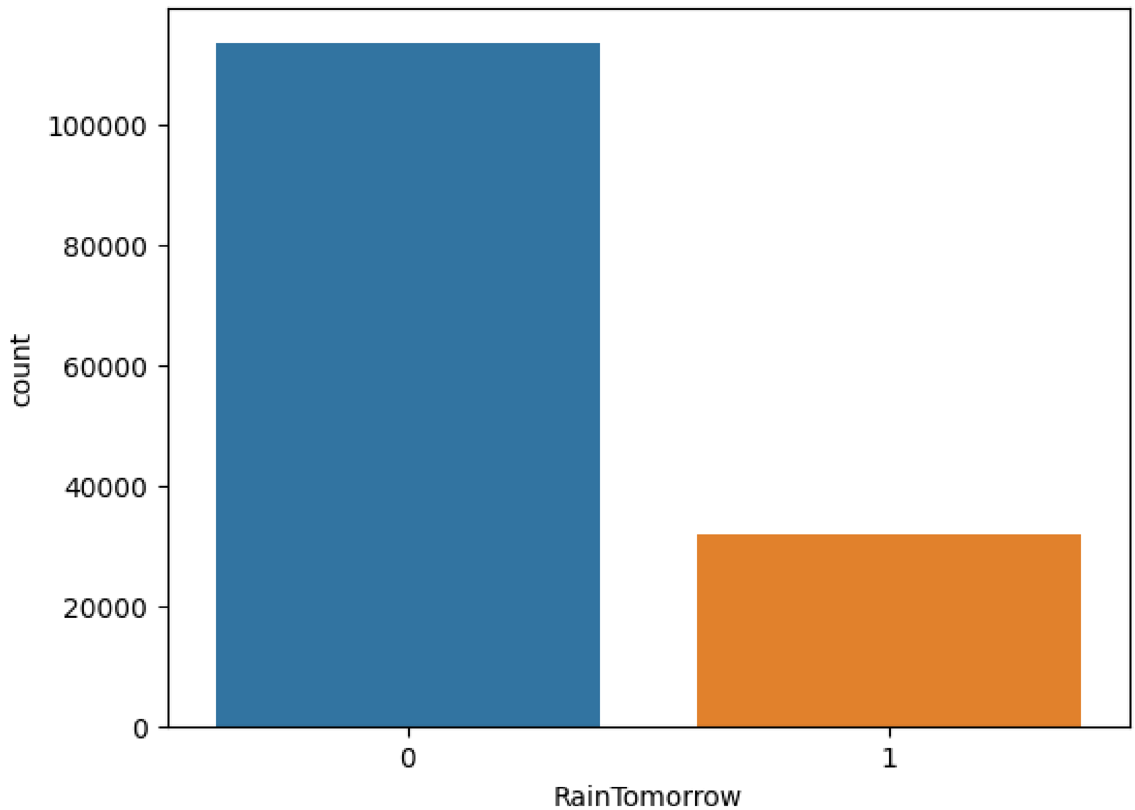


MinTemp
MaxTemp
Rainfall
Evaporation
Sunshine
WindGustSpeed
WindSpeed9am
WindSpeed3pm
Humidity9am
Humidity3pm
Pressure9am
Pressure3pm
Cloud9am
Cloud3pm
Temp9am
Temp3pm


```
df.boxplot()
```




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



Activity 5: Feature Scaling

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

```
MinTemp
MaxTemp
Rainfall
Evaporation
Sunshine
WindGustSpeed
WindSpeed9am
WindSpeed3pm
Humidity9am
Humidity3pm
Pressure9am
Pressure3pm
Temp9am
Temp3pm
```

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

7.2 Feature 2

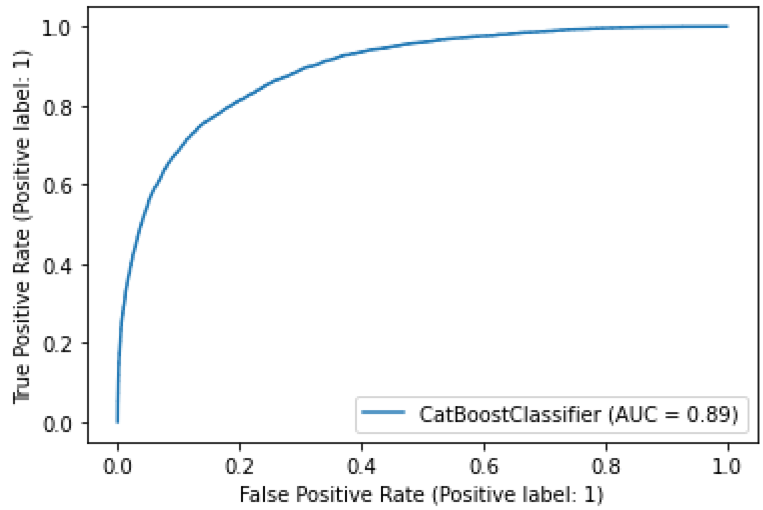
Flask web app

```python
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("predictor.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,port=5001)
```

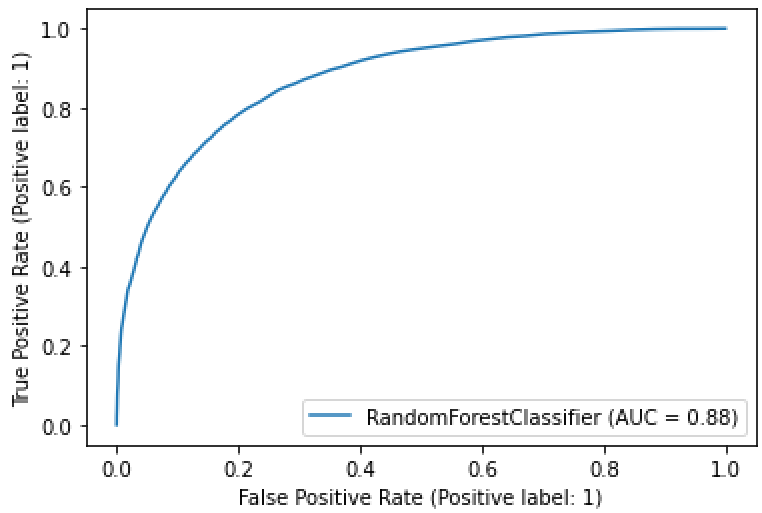
8. PERFORMANCE TESTING

8.1 Performace Metrics

1.	Metrics	**Regression Model:** MAE - , MSE - , RMSE - , R2 score - **Classification Model:** Confusion Matrix - , Accuray Score- & Classification Report -	1. CatboosClassifier MODEL- [[21520 1197] [2777 3598]] 0.8633988725422796 precision recall f1-score support 0 0.89 0.95 0.92 22717 1 0.75 0.56 0.64 6375 accuracy 0.86 29092 macro avg 0.82 0.76 0.78 29092 weighted avg 0.86 0.86 0.86 29092 

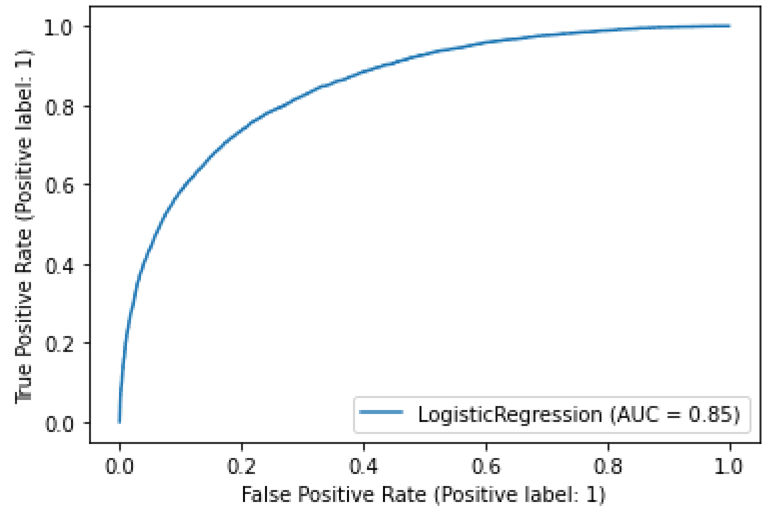
2. RandomForestClassifier()-

```
[[20633  2084]
 [ 2470  3905]] 0.8434621201704936
              precision    recall  f1-score   support

           0    0.89      0.91      0.90     22717
          1       0.65      0.61      0.63      6375
     accuracy                          0.84     29092
macro avg       0.77      0.76      0.77     29092
weighted avg       0.84      0.84      0.84     29092
```



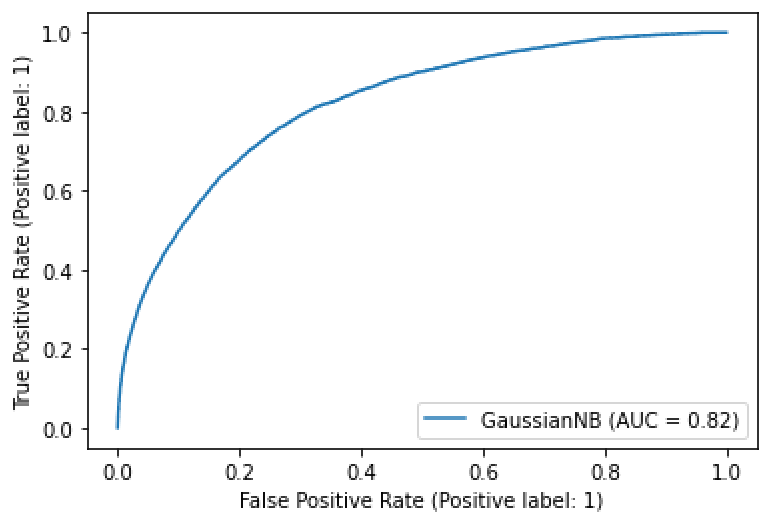
3. LogisticRegression()-

```
[[17649  5068]
 [ 1519  4856]] 0.7735803657362849
              precision    recall  f1-score   support

           0    0.92      0.78      0.84     22717
          1       0.49      0.76      0.60      6375
     accuracy                                 0.77      29092
macro avg       0.71      0.77      0.72     29092 weighted
avg        0.83      0.77      0.79     29092
```



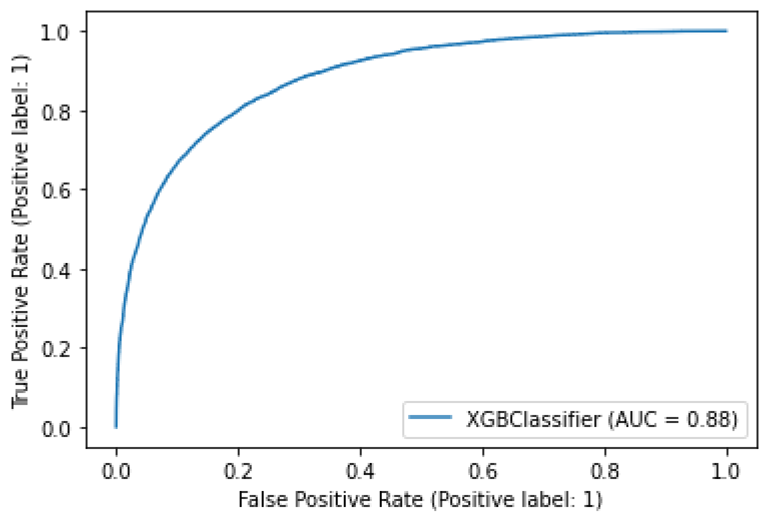
4. GaussianNB()-

```
[[17078  5639]
 [ 1661  4714]] 0.7490719098033823
              precision    recall  f1-score   support

           0       0.91      0.75      0.82     22717
           1       0.46      0.74      0.56      6375

    accuracy                           0.75     29092
   macro avg       0.68      0.75      0.69     29092
weighted avg       0.81      0.75      0.77     29092
```



5. XGBClassifier-

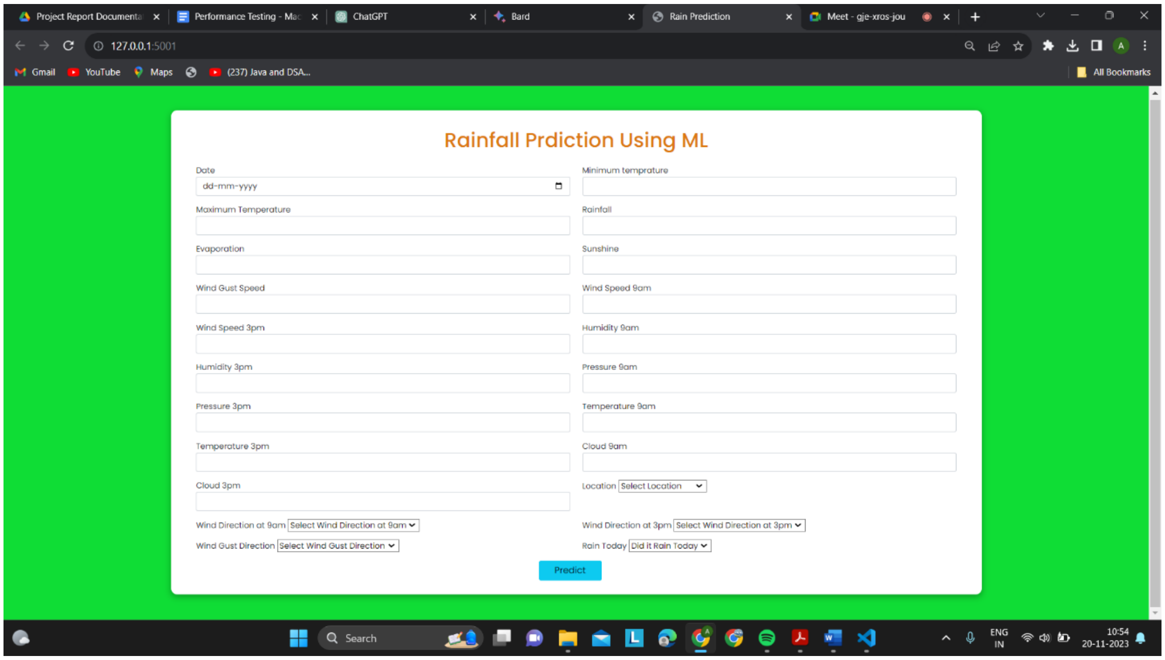
```
[[21396  1321]
 [ 2844  3531]] 0.8568334937439847
              precision    recall  f1-score   support

           0       0.88      0.94      0.91     22717
1       0.73      0.55      0.63      6375
     accuracy                           0.86     29092
macro avg       0.81      0.75      0.77     29092
weighted avg       0.85      0.86      0.85     29092
```



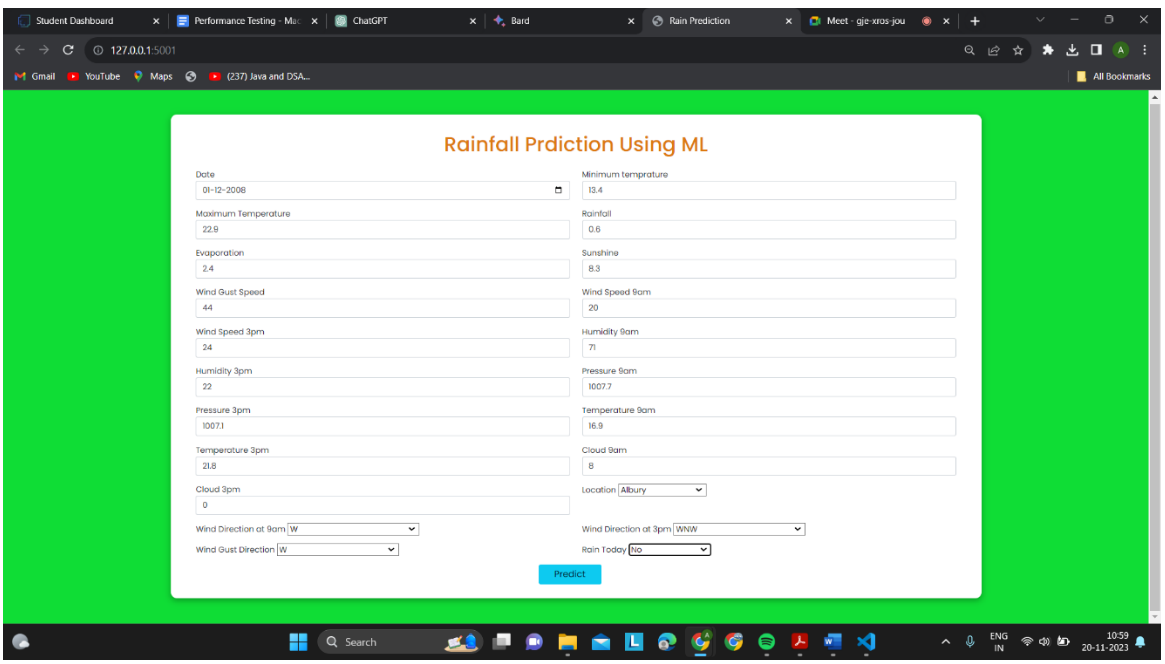
9. RESULTS

a. Output Screenshots

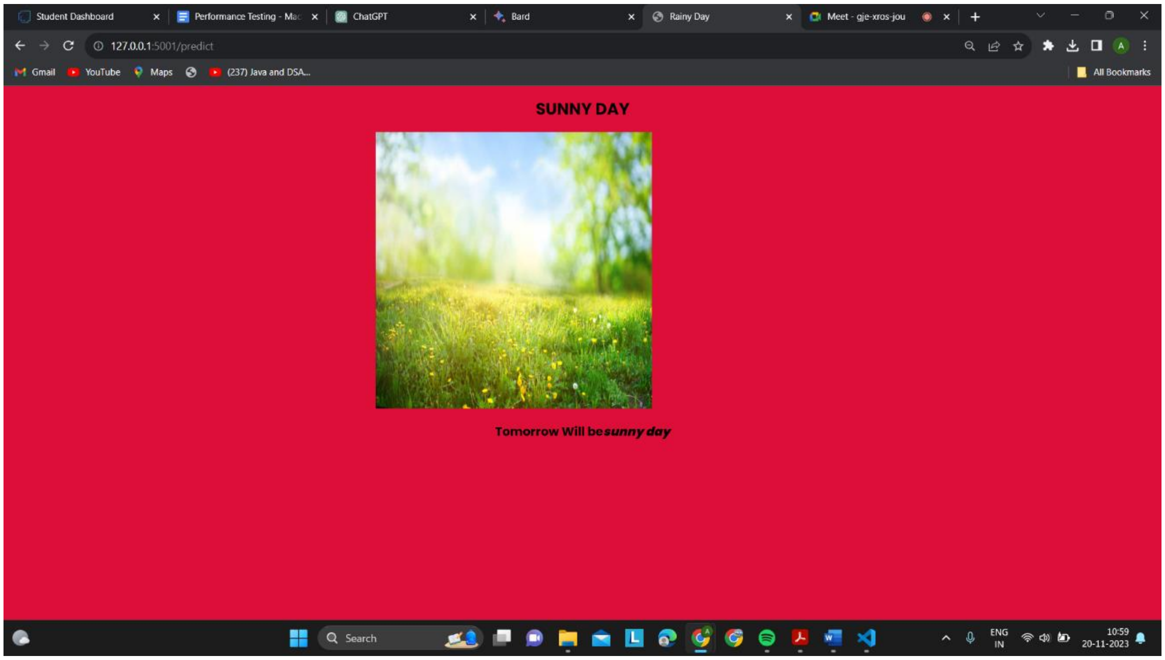
User Interface-



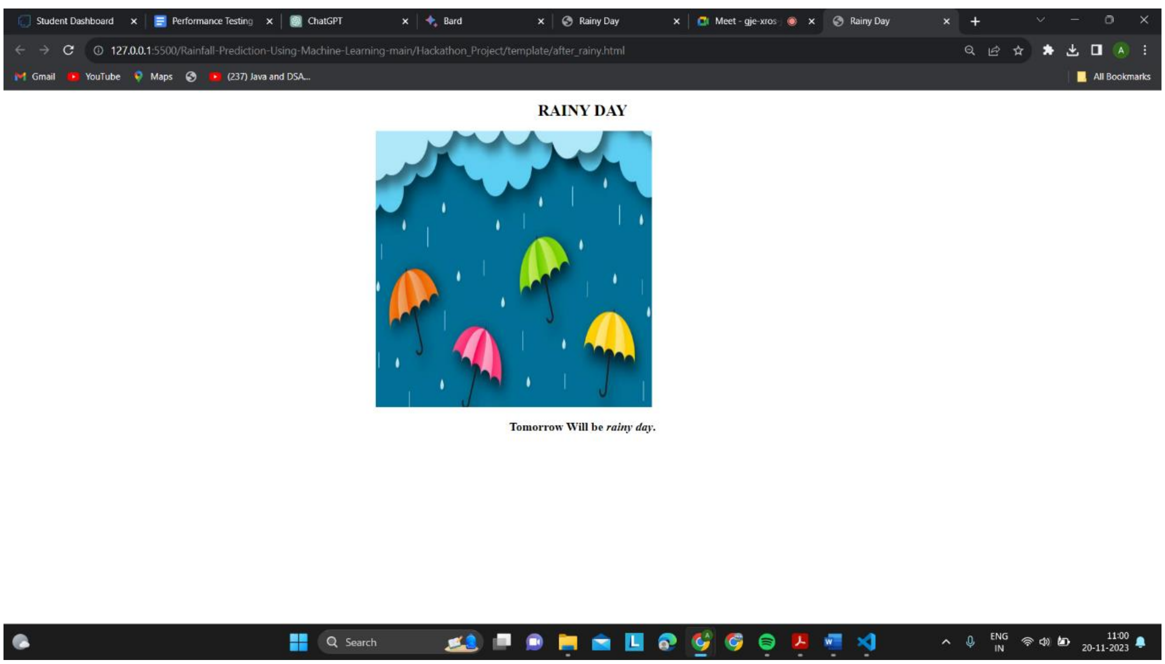
Values Takes from the Dataset-



After_sunny days-



After_Rainy Day-



10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

Improved Accuracy: Machine learning algorithms enhance the precision of rainfall predictions, outperforming traditional methods.

Early Warning System: The system provides timely alerts, enabling proactive decision-making for agriculture, water resource management, and disaster preparedness.

Adaptability: Machine learning models continuously learn from new data, ensuring adaptability to changing weather conditions.

10.2 Disadvantages

Data Dependency: Accuracy is contingent on the availability and quality of historical and real-time data, posing challenges in regions with limited meteorological infrastructure.

Complexity: Implementing machine learning introduces complexity, making the system less accessible for users without a background in the field.

Resource Intensive: Computational requirements for machine learning models may be resource-intensive, presenting challenges in terms of hardware infrastructure and energy consumption.

11. CONCLUSION

The Rainfall Prediction system using machine learning algorithms represents a significant advancement, overcoming the limitations of traditional methods. Improved accuracy and early warning capabilities make it a valuable tool for decision-makers in various sectors. Challenges such as data dependency and complexity are acknowledged and require attention for broader adoption.

12. FUTURE SCOPE

Integration of Additional Data Sources: Incorporate diverse data sources like satellite imagery and soil moisture levels for a more comprehensive understanding of rainfall patterns.

Ensemble Learning Approaches: Explore ensemble learning methods to improve overall forecasting performance by combining multiple machine learning models.

User Interface Enhancements: Continuously improve user interfaces, including mobile applications and dashboards, for wider accessibility and usability.

Collaboration with Meteorological Agencies: Foster collaborations with meteorological agencies for the integration of cutting-edge research and technologies into operational forecasting systems.

13. APPENDIX

Source Code

Link- https://github.com/smartinternz02/SI-GuidedProject-608872-1697870922/tree/main/Project%20Development

GitHub & Project Demo Link

Github link- https://github.com/smartinternz02/SI-GuidedProject-608872-1697870922/tree/main/Project%20Development

Project Demo Link-

https://drive.google.com/file/d/1h5wEcNrlIQ_EsVmocf29WI9mL4BbTkq9/view?usp=drive_link