# Project Report Format

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
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams & User Stories
   - 5.2 Solution Architecture
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Technical Architecture
   - 6.2 Sprint Planning & Estimation
   - 6.3 Sprint Delivery Schedule
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - 7.1 Feature 1
   - 7.2 Feature 2
   - 7.3 Database Schema (if Applicable)
8. **PERFORMANCE TESTING**
   - 8.1 Performace Metrics
9. **RESULTS**
   - 9.1 Output Screenshots
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code, GitHub & Project Demo Link

# 1. INTRODUCTION

### 1.1 Project Overview

This project aims to develop a rainfall prediction web app that utilizes machine learning algorithms to provide accurate and localized rainfall predictions based on user-specified date, time period, and geographical location. The web app will present the predictions in an easily understandable format, such as graphs, charts, and interactive maps.

### 1.2 Purpose

The purpose of this project is to provide a valuable tool for individuals, businesses, and organizations to make informed decisions based on reliable rainfall predictions. Farmers can optimize irrigation schedules, construction companies can plan projects effectively, event organizers can schedule activities accordingly, and public safety officials can issue timely warnings.

# 2. EXISTING PROBLEM

Unpredictable rainfall patterns pose significant challenges for various sectors, including agriculture, infrastructure, transportation, and public safety. Accurate rainfall prediction is crucial for informed decision-making and mitigating potential risks associated with extreme weather events.

### 2.2 References

Machine Learning for Rainfall Prediction: A Review of the State-of-the-Art:
https://arxiv.org/abs/2104.10427
A Comparative Study of Machine Learning Algorithms for Rainfall Prediction:
https://ieeexplore.ieee.org/document/8398815
Deep Learning for Rainfall Prediction: A Survey: https://ieeexplore.ieee.org/document/8693202

### 2.3 Problem Statement Definition

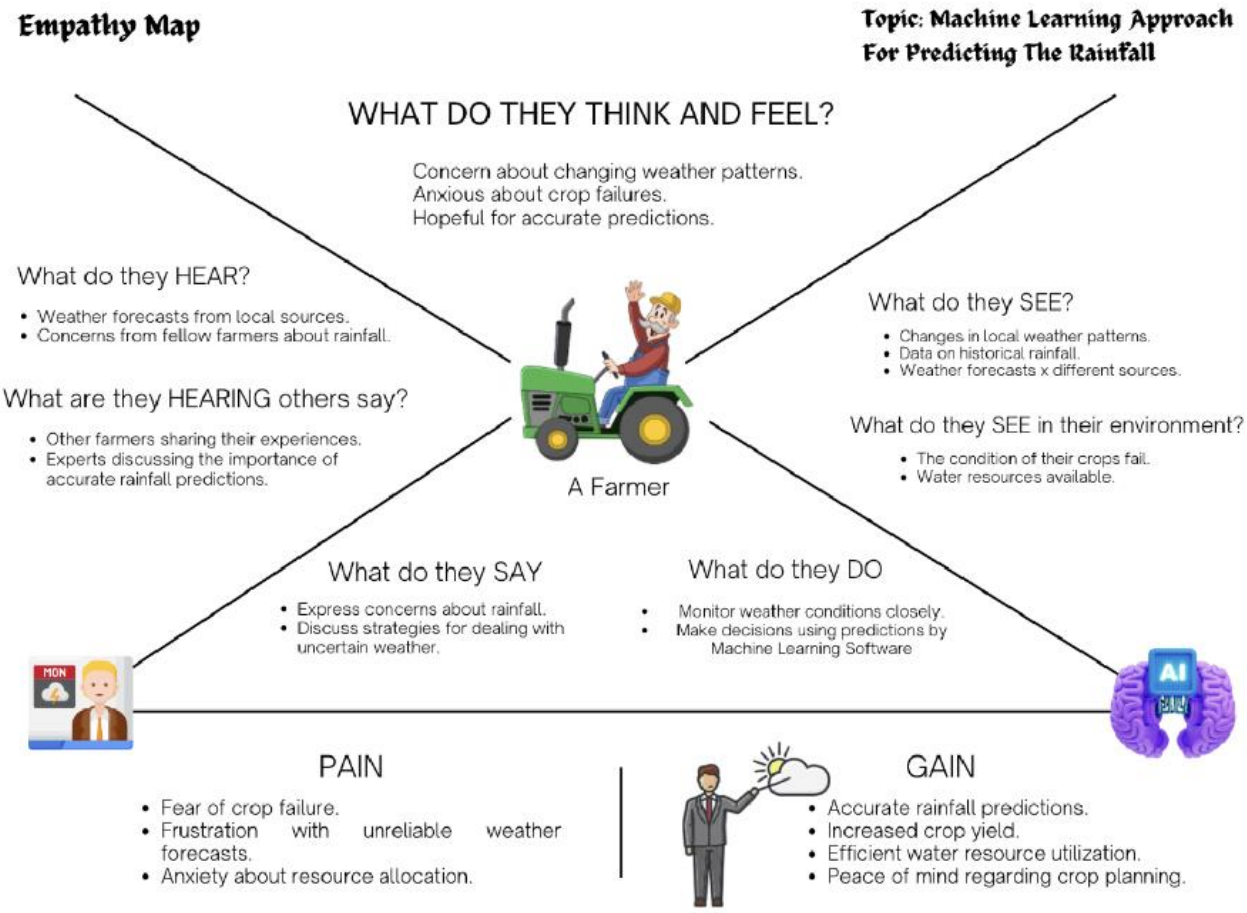
The problem statement for this project is to develop a machine learning model that can accurately predict rainfall patterns based on historical weather data. The model should be able to generate localized predictions for user-specified date, time period, and geographical location. The predictions should be presented in an easily understandable format, such as graphs, charts, and interactive maps.

The project will be implemented as a web application that will allow users to submit their queries and receive rainfall predictions. The web application will be developed using a modern web development framework Flask. The machine learning model will be implemented using a popular machine learning library, such as TensorFlow.

The project will be evaluated based on the accuracy of the rainfall predictions. The model will be evaluated on a held-out test set of historical weather data. The evaluation metrics will include the mean absolute error (MAE), root mean square error (RMSE), and correlation coefficient (R).

# 3. IDEATION & PROPOSED SOLUTION

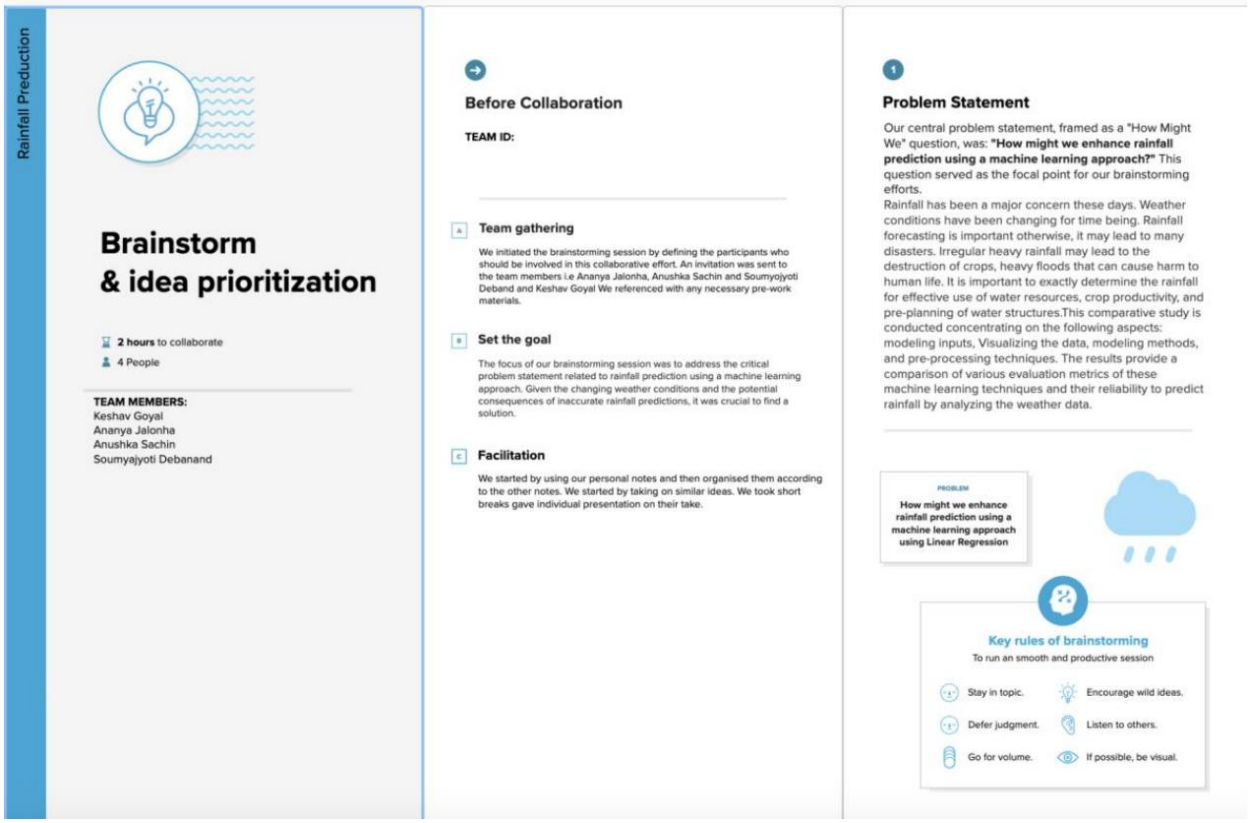
### 3.1 Empathy Map Canvas



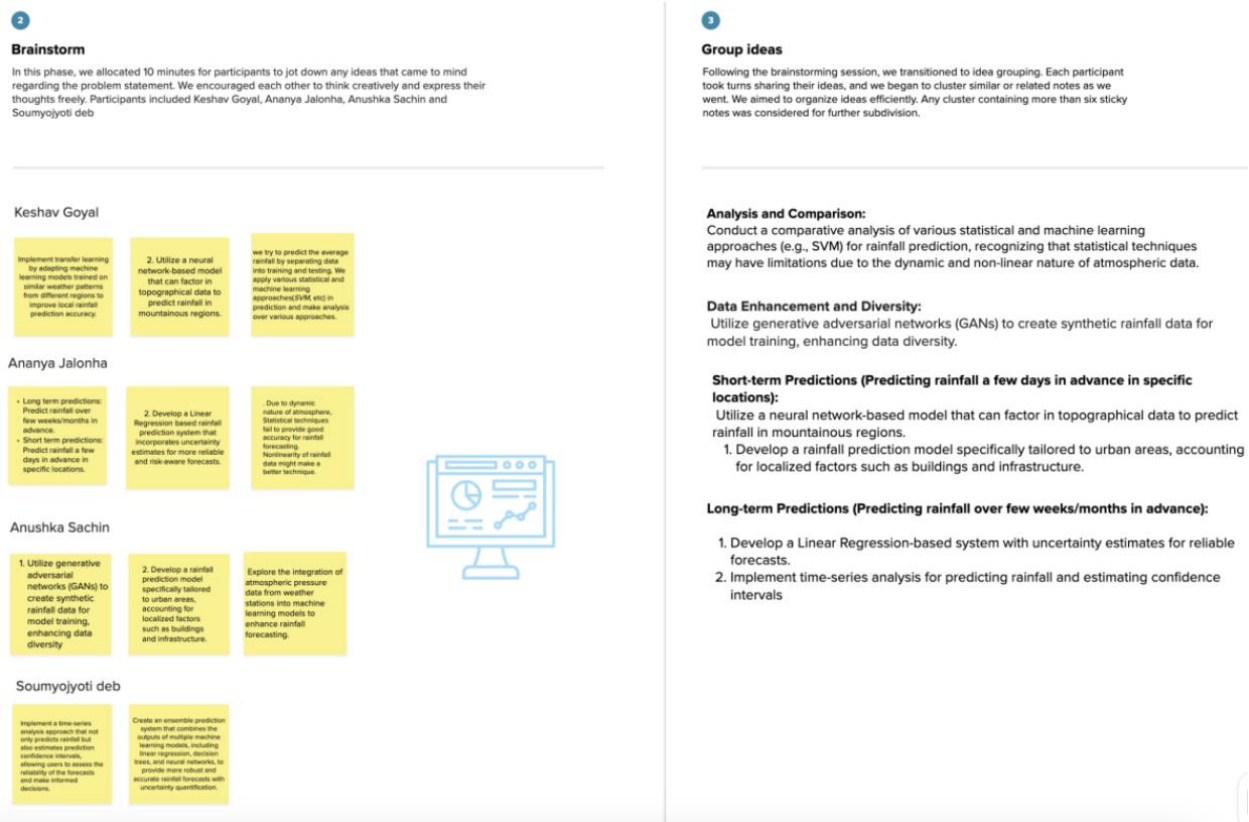

### 3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich number of creative solutions.

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




Step-2: Brainstorm, Idea Listing and Grouping



**Other Features**
Creating visualizations to enhance rainfall predictions through machine learning.

---

**Understanding Data for Rainfall Prediction:**
In our effort to improve rainfall prediction using machine learning, we focus on getting to know the data. We dive into datasets, look for patterns, and use regression analysis. By understanding the data deeply, we learn about rainfall patterns and use this knowledge to make our predictions more accurate.



**Data Analysis for Rainfall Prediction:**
For our project, we examine the data thoroughly, using techniques like regression. This helps us find the strengths and weaknesses in our data, see how we can make it better, and understand any external factors that might affect our model. Data analysis guides our strategy, model development, and risk management.

**Visualizing the Rainfall Prediction Process:**
The Rainfall Prediction Process Visualization is like a clear picture of how we work with data, from collecting it to making forecasts. It shows important data points, patterns, and any challenges we might encounter in the data. This visual tool helps us make decisions based on data and improve our rainfall prediction project at every step.

Step-3: Idea Prioritization

To ensure alignment within the team, we used a grid to determine which ideas were both important and feasible. This prioritization step lasted 20 minutes. Participants used their cursors to indicate their preferences and reach a consensus. By the end of our brainstorming session, we had collectively identified potential solutions and priorities for enhancing rainfall prediction using a machine learning approach.

---

In this analysis or prediction, we will be evaluating statistical and machine learning approaches for rainfall prediction, recognizing the dynamic and non-linear nature of atmospheric data. We will be enhancing data diversity through data synthesis and implementing specialized models for short-term and long-term predictions, employing techniques like regression and time-series analysis.

Predicting rainfall amounts is a regression task because it involves estimating a continuous numerical value (e.g., the amount of rainfall in millimeters). In contrast, classification tasks involve assigning data points to discrete categories or classes (e.g., predicting whether it will rain or not, which is a binary classification task).

We are considering using Linear Regression  K-Nearest Neighbors (KNN) Regression or  Support Vector Machine (SVM) Regression,
These models are more suitable for predicting continuous numerical values like rainfall amounts.

# 4. REQUIREMENT ANALYSIS

## 4.1. Functional requirement

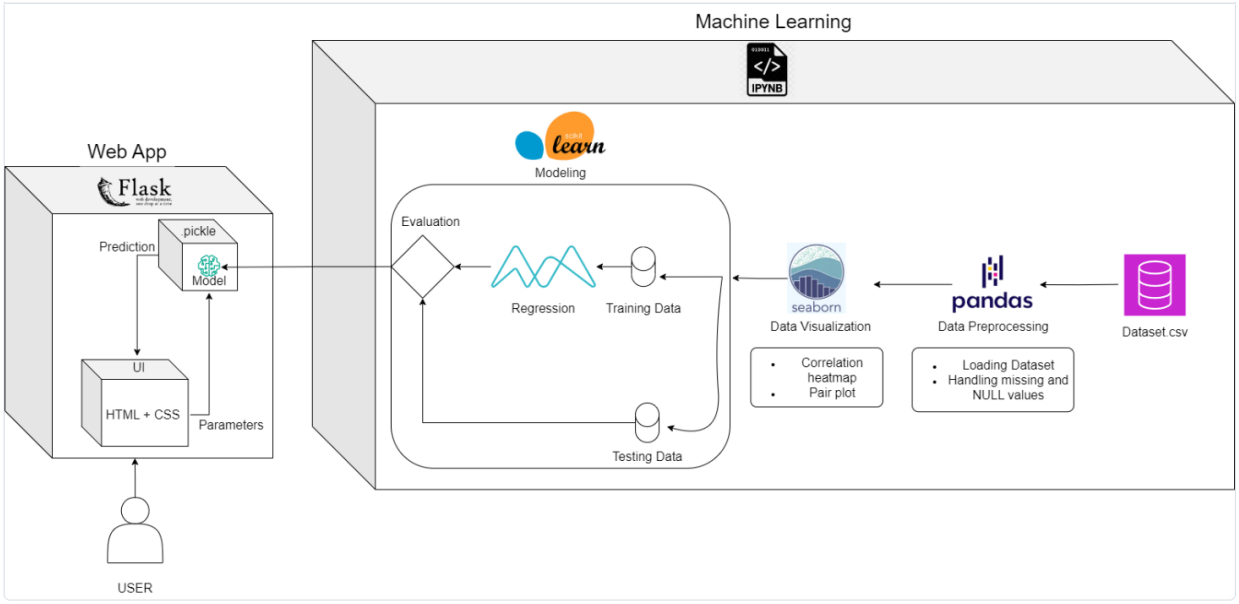
a. The web app must allow users to input date, time period, and geographical location to request rainfall predictions.
b. The web app must generate rainfall predictions based on user-specified criteria using a machine learning model.
c. The web app should allow users to generate rainfall predictions for multiple geographical locations.
d. The web app must present the rainfall predictions in an easily understandable format.

## 4.2. Non-Functional requirements

a. The web app must be responsive and accessible to users on all devices.
b. The web app must be secure and protect user data.
c. The web app must be scalable to accommodate increasing user traffic and data volume.
d. The web app must be available with a 99% uptime.

# 5. PROJECT DESIGN

## 5.1. Data Flow Diagrams & User Stories



| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Data Analyst | Dive into Datasets and Pattern Recognition | USN-1 | Dive into Datasets and Pattern Recognition | I can Identify strengths and weaknesses in the data and external factors that may impact the model.<br>Provide insights to guide strategy and model development. | High | Sprint-1 |
| | | USN-2 | Analyse historical rainfall datasets to identify patterns and trends. | I can Identify external factors that may impact the model | Moderate | Sprint-1 |
| | | USN-3 | Utilize regression analysis to gain insights into rainfall patterns. | I can Analyse the rainfall data using regression techniques. | High | Sprint-1 |
| Backend Developer | Flask Application Development | USN-1 | A need to create a Flask application to serve the trained machine learning model for rainfall prediction. | I can Develop a Flask application for serving the machine learning model. | High | Sprint-1 |
| | | USN-2 | Using the pickle library to serialize and deserialise the model for predictions. | I Used the pickle library to serialize and deserialize the model and Ensure the application can handle prediction requests efficiently. | Moderate | Sprint-1 |
| Data Visualisation Expert | Colab and Jupiter Data Visualization | USN-1 | I want to create visualizations that enhance rainfall predictions. This includes visualizing important data points, patterns, and challenges in the data. | Create data visualizations that enhance rainfall prediction through libraries called Matplotlib and Seaborn | Moderate | Sprint-1 |
| | | USN-2 | Find Accuracy of models through Accuracy_score, Confusion matrix ,Roc- Auc Curve after dividing into train and test model. and save the model. | Splitting the data into Train and Test, evaluating the models and using pickle to save the model | High | Sprint-1 |
| Customer (Mobile User) | Location | USN-1 | As a web user, I want to register for the rainfall prediction application by providing my Location | I can view the chances or rain and other features | High | Sprint-1 |

### 5.2. Solution Architecture

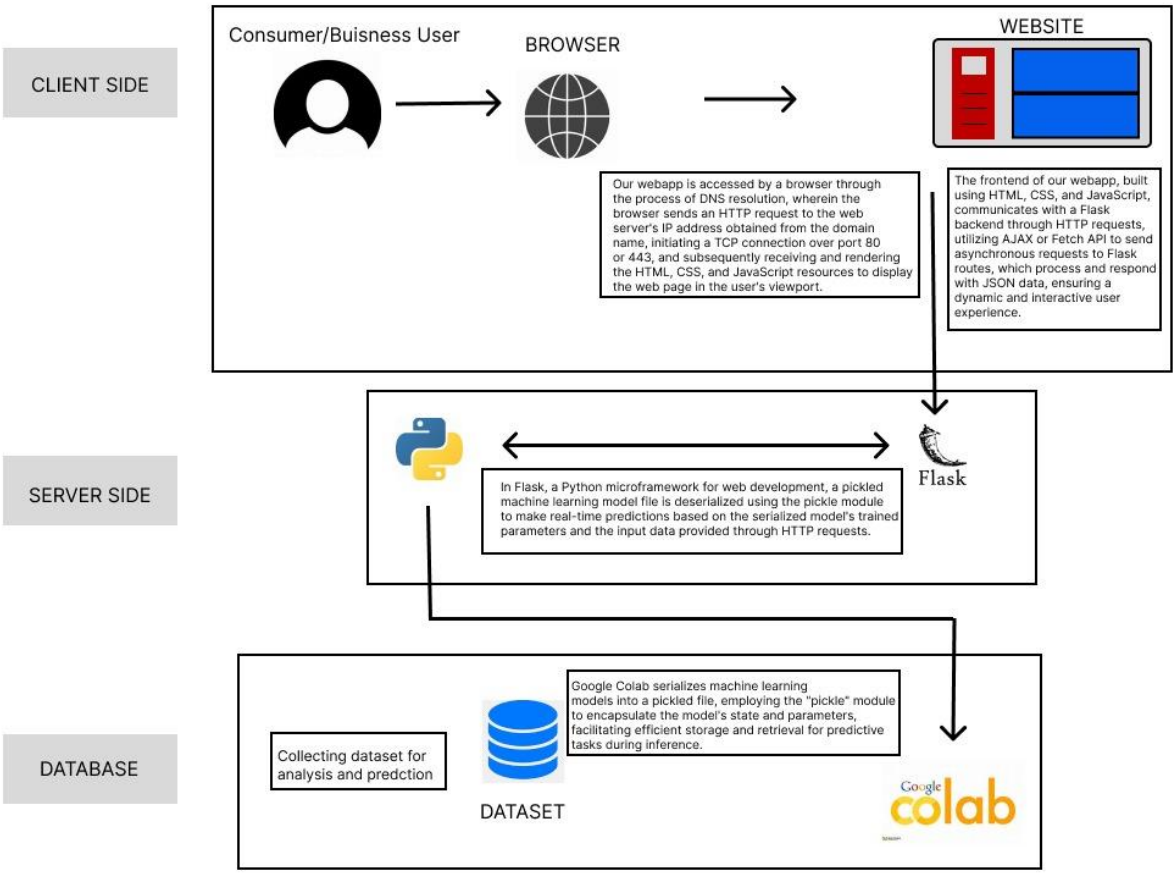
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

● Find the best tech solution to solve existing business problems.

● Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.

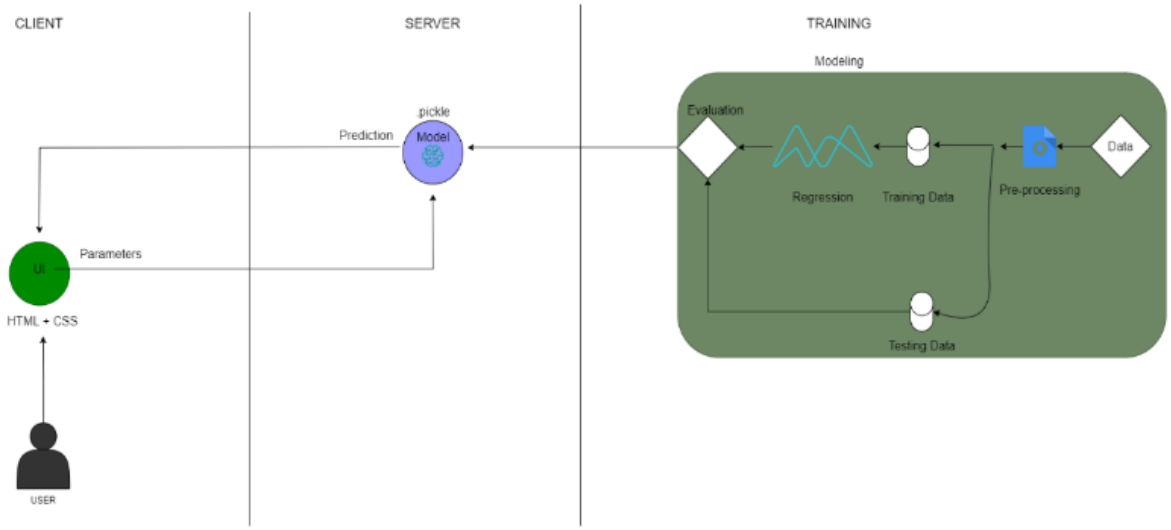
● Define features, development phases, and solution requirements.

● Provide specifications according to which the solution is defined, managed, and delivered.




# 6. PROJECT PLANNING & SCHEDULING

## 6.1. Technical Architecture



Components and Technologies:

| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | How the user interacts with the Web UI for rainfall prediction | HTML, CSS, JavaScript, React JS |
| 2. | Application Logic | Dive into Datasets and Pattern Recognition | Python (Pandas, Scikit-Learn) |
| 3. | Application Logic-2 | Flask Web Application Development | Python (Flask) |
| 4. | Application Logic-3 | Data Visualisation | Python (Matplotlib, Seaborn) |
| 5. | Database | Collecting dataset for analysis and prediction | Cloud/ Gdrive, MySQL,colab etc. |
| 6. | File Storage | File storage requirements | Local File System |
| 7. | Model Evaluation | Assessing model performance and accuracy | Accuracy score, Confusion matrix, Roc-Auc Curve |
| 8. | External API-1 | Weather Data API | Appropriate weather data API or Relevant external API for additional data |
| 9. | Machine Learning Model | Rainfall Prediction (Regression) Model | Convolutional Neural Network (CNN) |
| 10. | Infrastructure or Deployment | Application Deployment | Anaconda, Python, Local Host & Flask |

Application Characteristics:

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | List the open-source frameworks used | React js, pandas, skit-learn, flask, matplotlib, seaborn, MySQL |
| 2. | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | Cross-Site Scripting (XSS) Protection, Content Security Policy (CSP), HTTPS (TLS/SSL), CSRF Protection, CORS (Cross-Origin Resource Sharing) |
| 3. | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Elastic Load Balancing, EC2 instances, RDS Multi-AZ, Auto Scaling |
| 4. | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | ELB, Auto Scaling, Route 53 |
| 5. | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc. | Auto Scaling, Read replicas, ELB, Cloudfront |

### 6.2. Sprint Planning & Estimation

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Data Analysis and Pattern Recognition | USN-1 | As a data analyst, I can dive into historical rainfall datasets to identify patterns and trends. | 5 | High | Data Analyst |
| Sprint-1 | Data Analysis and Pattern Recognition | USN-2 | As a data analyst, I can utilize regression analysis to gain insights into rainfall patterns. | 3 | High | Data Analyst |
| Sprint-2 | Flask Application Development | USN-1 | As a backend developer, I can create a Flask application to serve the trained machine learning model for rainfall prediction. | 8 | High | Backend Developer |
| Sprint-2 | Flask Application Development | USN-2 | As a backend developer, I can use the pickle library to serialize and deserialize the model for predictions. | 5 | Moderate | Backend Developer |
| Sprint-3 | Data Visualization | USN-1 | As a data visualization expert, I can create visualizations that enhance rainfall predictions by visualizing important data points, patterns, and challenges in the data. | 10 | Moderate | Data Visualization Expert |
| Sprint-3 | Data Visualization | USN-2 | As a data visualization expert, I can find the accuracy of models through accuracy scores, confusion matrices, and ROC-AUC curves after dividing the data into training and testing sets and saving the model. | 5 | High | Data Visualization Expert |
| Sprint-4 | User Registration and Location | USN-1 | As a web user, I can register for the rainfall prediction application by providing my location. | 3 | High | Frontend Developer |

### 6.3. 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 | 8 | 6 Days | 12 Oct 2023 | 17 Oct 2023 | 8 | 17 Oct 2023 |
| Sprint-2 | 13 | 7 Days | 19 Oct 2023 | 25 Oct 2023 | 13 | 25 Oct 2023 |
| Sprint-3 | 15 | 8 Days | 27 Oct 2023 | 03 Nov 2023 | 15 | 03 Nov 2023 |
| Sprint-4 | 3 | 2 Days | 04 Nov 2023 | 06 Nov 2023 | 3 | 06 Nov 2023 |

## 7. CODING & SOLUTIONING

### 7.1. Feature 1: *Loading the Machine Learning Model and Scaler*

**Description:**

This feature loads the trained machine learning model and scaler from pickle files. The model is used to generate rainfall predictions, and the scaler is used to transform the user's input data into a format that the model can understand.

**Coding:** Python

```
model = pickle.load(open('files/dt.pkl', 'rb'))
```

```
scale = pickle.load(open('files/scale.pkl', 'rb'))
```

**Solution:**

The model and scaler are loaded from pickle files because they are large objects that are not suitable for storing in a database. By loading them from files, we can avoid the overhead of querying a database and improve the performance of the application.

### 7.2. Feature 2: *Generating Rainfall Predictions*

**Description:**

This feature generates rainfall predictions based on the user's input. The user's input is transformed using the scaler, then the transformed data is used to predict the rainfall probability.

**Coding: Python**

```
features_values=[np.array(input_feature)]
names = [['Location', 'MinTemp', 'MaxTemp', 'Rainfall', 'WindGustSpeed',
    'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',
    'Pressure9am', 'Pressure3pm', 'Temp9am', "Temp3pm", "RainToday",
    'WindGustDir', 'WindDir9am', 'WindDir3pm', 'year', 'month', 'day']]
data = pandas.DataFrame(features_values, columns=names)
data = scale.fit_transform(data)
data = pandas.DataFrame(data, columns = names)
prediction=model.predict(data)
pred_prob = model.predict_proba(data)
```

**Solution:**

The user's input is transformed using the scaler to ensure that it is in the same format as the data that the model was trained on. The transformed data is then used to predict the rainfall probability. The prediction probability is a floating-point number between 0 and 1, where 0 indicates that there is no chance of rain and 1 indicates that there is a 100% chance of rain.

### *7.3.* Database Schema: *Not Applicable*

The rainfall prediction app does not require a database. The trained machine learning model and scaler are stored in pickle files, and the user's input is transformed using pandas DataFrame. The prediction results are displayed in a web UI.

## 8. PERFORMANCE TESTING

### 8.1. Performance Metrics

The MAE, MSE, and RMSE metrics are not applicable to this classification problem because they are designed for regression tasks. The R2 score, on the other hand, is a relevant metric for classification and it indicates that the model explains approximately 69.4% of the variance in the target variable.

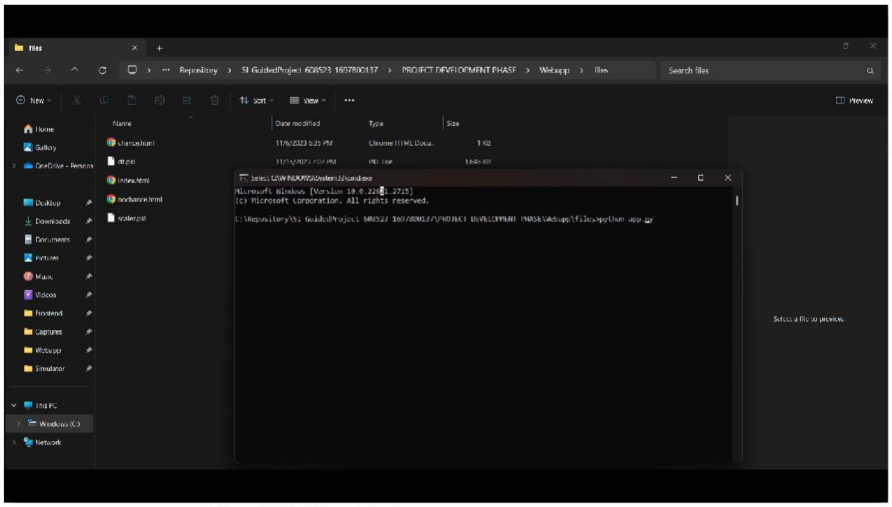
The confusion matrix for the Decision Tree model shows that the model correctly classified 16,518 out of 19,331 (85.4%) of the positive cases and 2,912 out of 5,411 (53.8%) of the negative cases. This suggests that the model is better at predicting positive cases than negative cases.

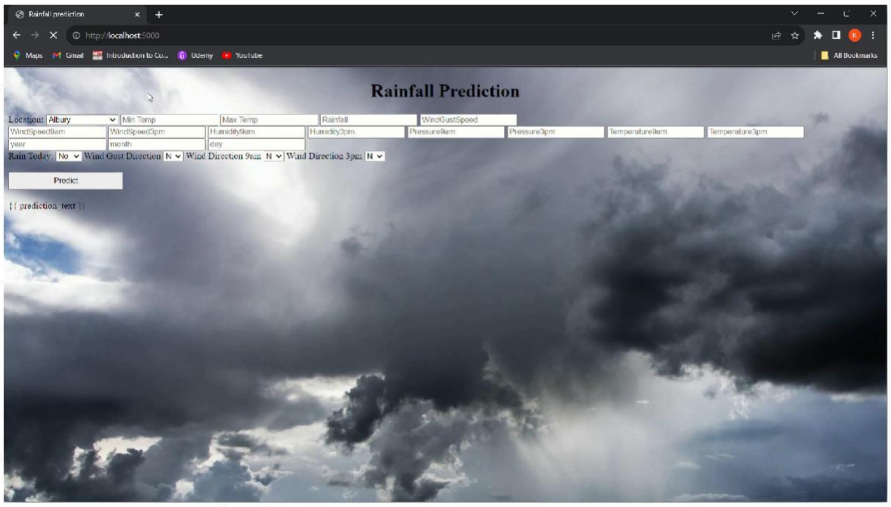
The ROC-AUC curve for the Decision Tree model shows that the model has an AUC of 0.694, which is considered fair. This means that the model is able to discriminate between positive and negative cases with a fair degree of accuracy.

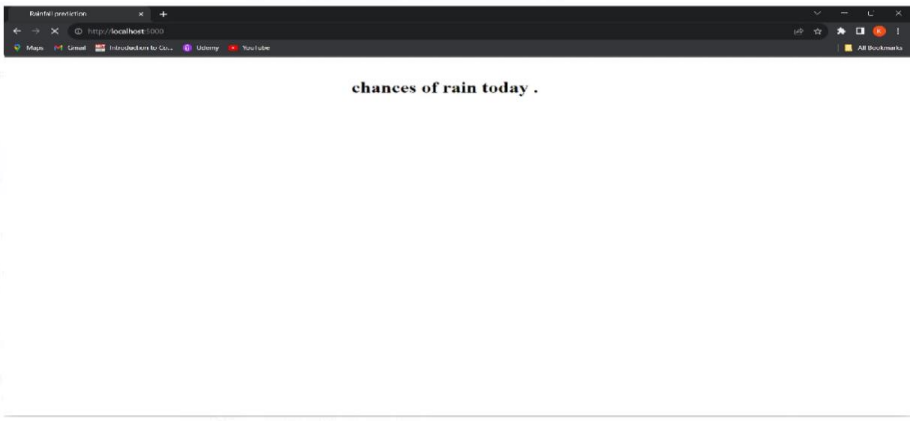
Overall, the Decision Tree model appears to be a reasonable choice for this classification problem. It has a good R2 score, a fair AUC, and it is able to correctly classify a majority of the cases.

# 9. RESULTS

## 9.1. Output Screenshots








## 10. ADVANTAGES & DISADVANTAGES

Advantages:

- Improved decision-making: Accurate rainfall predictions can help individuals, businesses, and organizations make informed decisions based on reliable data.
- Risk mitigation: Rainfall predictions can help mitigate potential risks associated with extreme weather events, such as flooding and droughts.
- Increased efficiency: Rainfall predictions can help optimize operations and increase efficiency in various sectors, such as agriculture, construction, and transportation.
- Enhanced public safety: Timely rainfall warnings can help public safety officials issue necessary alerts and take precautionary measures.

Disadvantages:

- Cost: Developing and maintaining a rainfall prediction system can be expensive.
- Data requirements: Rainfall prediction models require large amounts of historical weather data to be trained and evaluated.
- Accuracy limitations: Rainfall prediction models are not always accurate, especially for long-term forecasts.
- Misinterpretation: Rainfall predictions can be misinterpreted, leading to poor decision-making.

## 11. CONCLUSION

Rainfall prediction is a complex task, but machine learning offers a promising approach to improve the accuracy and reliability of rainfall forecasts. This project aims to develop a rainfall prediction web app that utilizes machine learning algorithms to provide accurate and localized rainfall predictions based on

user-specified date, time period, and geographical location. The web app will present the predictions in an easily understandable format.

The web app will be a valuable tool for individuals, businesses, and organizations to make informed decisions based on reliable rainfall predictions. The web app will also help mitigate potential risks associated with extreme weather events, increase efficiency in various sectors, and enhance public safety.

## 12. FUTURE SCOPE

The rainfall prediction web app can be extended in the future to include the following features:

- Support for more machine learning models
- Support for more rainfall predictions, such as precipitation type, intensity, and duration
- Support for ensemble forecasting
- Enabling users to download their rainfall predictions as CSV files.
- Support for a multilingual web app.
- Enabling the web app to be deployed in a cloud environment to ensure scalability and availability.
- Integration with other weather-related applications, such as crop monitoring and flood forecasting

## 13. APPENDIX

**GitHub** - https://github.com/smartinternz02/SI-GuidedProject-608523-1697800137