Machine Learning Approach For Predicting Rainfall

Project Document

Team Details
Member-1: Anirudh Sriram
Member-2: Siddhi Marri

CONTENTS

INTRODUCTION

Project Overview

In India, a vast and agriculturally dependent nation, the significance of accurate rainfall prediction cannot be overstated. Given its expansive geography, rainfall emerges as a pivotal seasonal event, crucial for the agricultural sector that forms the backbone of the country's economy. While the India Meteorological Department (IMD) plays a vital role in forecasting, the challenge lies in improving prediction accuracy beyond the initial 2-3 days, as an extended timeframe is essential for effective planning and preparedness in a nation deeply intertwined with the rhythms of the monsoon. This project plans on using Deep Learning techniques to try and predict rainfall[to be edited based on what we actually do]

Purpose

This project aims to revolutionize rainfall prediction in India, catering to a diverse audience that includes farmers, construction site workers, and the general populace. By enhancing the accuracy and extending the forecast period, the project directly addresses the pressing need for better pre planning and damage mitigation strategies. For farmers, this means optimizing planting schedules and resource allocation, while construction site workers can plan tasks more efficiently, reducing the risk of weather-related disruptions. Moreover, the broader community gains the ability to take proactive measures for their health and physical safety, such as preparing for floods or avoiding outdoor activities during periods of heavy rainfall. In essence, the project seeks to empower individuals across various sectors, fostering resilience in the face of India's climatic variability.

LITERATURE REVIEW

Existing Problem

The concern surrounding rainfall in India is akin to a chameleon, constantly changing with the weather itself. The monsoon has become more erratic and unpredictable, bringing extreme rainfall on the one hand and sudden drought on the other. India is in a tropical zone. It is surrounded by sea and oceans and the weather is quite warm, humidity increases with high temperatures. Tropical weather is less predictable than middle altitudes like Europe, and the US. This highlights the pivotal need for precise rainfall predictions for the judicious utilization of water resources, optimization of crop yields, and prudent planning of water-related infrastructure. In the past few decades, the accuracy of track and intensity forecast of tropical cyclones has improved considerably, but the estimation and forecast of associated rainfall is still a challenge. Accurate rainfall estimation at the time of landfall is crucial as it enables disaster management agencies to plan for disaster management strategies in the affected areas

References

Ankur A, Busireddy NKR, Osuri KK, Niyogi D (2020) On the relationship between intensity changes and rainfall distribution in tropical cyclones over the North Indian Ocean. Int J Climatol 40:2015–2025
https://www.indiatoday.in/science/story/why-imd-can-t-predict-weather-like-us-europe-what-are-the-roadblocks-1976001-2022-07-15

Problem Statement Definition

This project endeavors to revolutionize rainfall prediction in India, a nation deeply intertwined with agriculture and the monsoon's rhythms. While the India Meteorological Department shoulders forecasting responsibilities, the pressing challenge lies in surpassing the current 2-3 day prediction limits. Deep learning techniques offer a new paradigm. The project's purpose extends beyond meteorology to benefit a wide range of stakeholders. It aims to enhance rainfall prediction accuracy and extend forecasting horizons. This advancement empowers farmers to optimize planting and resource allocation, enabling bountiful harvests. Construction workers benefit from efficient task scheduling, reducing weather-related disruptions. The broader community gains the ability to proactively safeguard their health and well-being during inclement weather, fostering resilience across diverse sectors in the face of India's capricious climate.

IDEATION AND PROPOSED SOLUTION

Empathy Map Canvas

Considering the details and demands of our project, an empathy map was created to unravel the specific needs and perspectives of our users in the context of rainfall prediction. The following details provides a concise exploration of the map, shedding light on the considerations that shape our commitment to a user-centric approach.

1. Target Audience: Farmers, Construction site workers and the general public
2. Pains: Potential property and life loss due to unprecedented rain, public hygiene and sanitation issues, road congestions and blockage, and irregular crop yields
3. Gain from the project: thoroughly considered travel plans, operational safety, preparedness and well planned crop management in cases of farmers
4. What the customers hear: failed crop management, road accidents due to unforeseen rain and dangers of working on a construction site.
5. What they might do: Avoid leaving their house, seek for sources which provide better prediction of rainfall, implement damage mitigation strategies, try studying the rainfall pattern on their own.
6. What they demand: better rainfall predicting sources
7. Actions influenced by other feelings and thoughts: Give on their work or task commitment due to safety concerns

The presented empathy map delves into the distinct requirements of our user base—farmers, construction workers, and the general public. Informed by these insights and as discussed, our primary objective will be to ensure our model is user-centric and easily accessible to the broader public.

Ideation and Brainstorming

During our brainstorming session, individual ideation yielded diverse suggestions, leading to two primary categories: "Access to the Public" and "Model Enhancement." Ideas encompassed a mobile application for localized rainfall predictions, a rainwater harvesting system, and a community resource center for farmers (grouped under access), while model-focused concepts included continuous learning integration and leveraging local knowledge.
The list of points is given below:

1. Mobile Application that uses Machine Learning to predict rainfall in a particular area and time
2. Rainwater Harvesting and Management System for effective management of rainwater
3. A community resource center equipped with rainfall data, enabling farmers to access vital information
4. Try to integrate continuous learning into our model to make rainfall prediction more accurate
5. Incorporate local knowledge to enhance prediction
6. Create a web Application for the general public to access

Prioritizing by feasibility and importance, points 1 (mobile app) and 6 (web app) were deemed high, emphasizing accessibility to the public. In addition, despite lower feasibility, point 4 (continuous learning) was assigned high priority, underscoring its strategic significance in model enhancement.

REQUIREMENT ANALYSIS

Functional Requirements

1. Data Collection and prepocessing: collection hourly and daily weather data from weatherapi, at regular intervals. Cleaning data and feature selection will be done in preprocessing stage.
2. Model Selection and training: multiple regression and classification models like RandomForest, XGBoost etc. will be used to train.
3. Model Serialization: the model will be saved as a pickle file in order to reuse it in the flask file for the user interface.
4. Web Application Development: Develop a web application using Flask. Create an intuitive user interface that allows users to input their location. The application should handle user requests and display predictions
5. Prediction Display

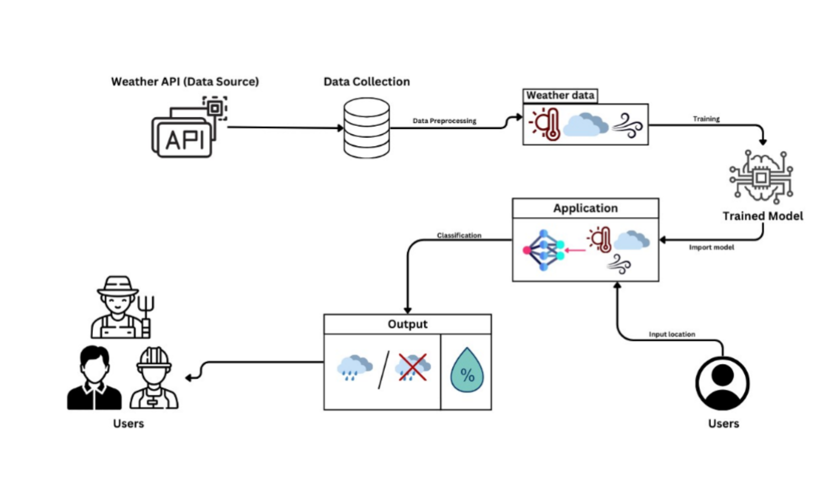
Non-Functional Requirements

1. Performance
2. Scalability
3. Reliability:
4. User Interface Usability
5. Compatibility

PROJECT DESIGN

Data Flow Diagram and User Stories

A Data Flow Diagram (DFD) is a graphical representation that depicts the flow of data within a system. It shows how information enters and exits the system, the changes to the information, and where it is stored. DFDs are used to visualize, understand, and simplify complex systems.

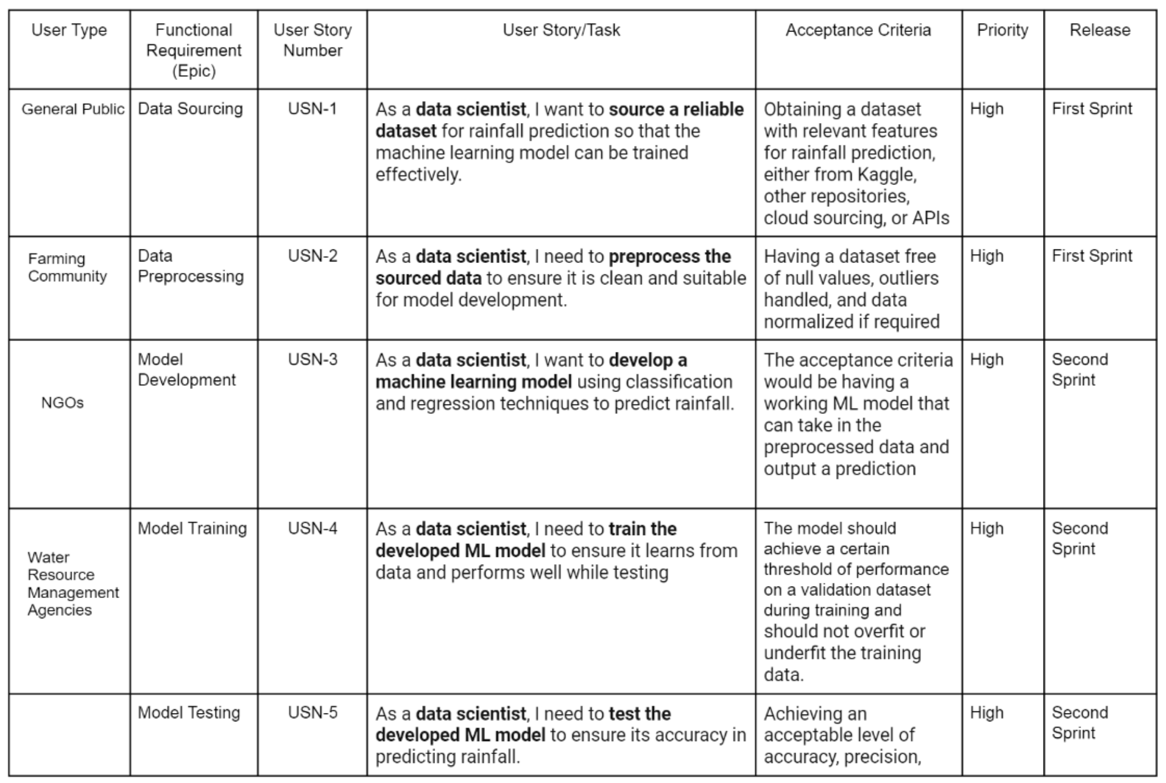


The above Data Flow Diagram (DFD) represents a rainfall prediction system. It depicts how information enters and exits the system, the changes to the information, and where it is stored. The process begins with the Weather API Data Source, which is an external entity that provides the raw weather data. This data could include parameters like temperature, humidity, wind speed, etc. This raw data is then fed into the Data Processing process. This could involve cleaning the data, handling missing values, and transforming the data into a suitable format for further processing.

The cleaned and transformed data, referred to as Weather Data, is then used to develop a model that has been trained on historical weather data to make rainfall predictions. Finally, the Output is presented to the Users. They could be end-users looking for weather forecasts, other systems that use these predictions for further processing, or even researchers studying weather patterns.

User Stories

A User Story is a tool used in Agile software development to capture a description of a software feature from an end-user perspective. It helps the team to understand the requirements and what the user expects from the system.



User Type	Functional Requirement (Epic)	User Story Number	User Story/Task	Acceptance Criteria	Priority	Release
General Public	Data Sourcing	USN-1	As a **data scientist**, I want to **source a reliable dataset** for rainfall prediction so that the machine learning model can be trained effectively.	Obtaining a dataset with relevant features for rainfall prediction, either from Kaggle, other repositories, cloud sourcing, or APIs	High	First Sprint
Farming Community	Data Preprocessing	USN-2	As a **data scientist**, I need to **preprocess the sourced data** to ensure it is clean and suitable for model development.	Having a dataset free of null values, outliers handled, and data normalized if required	High	First Sprint
NGOs	Model Development	USN-3	As a **data scientist**, I want to **develop a machine learning model** using classification and regression techniques to predict rainfall.	The acceptance criteria would be having a working ML model that can take in the preprocessed data and output a prediction	High	Second Sprint
Water Resource Management Agencies	Model Training	USN-4	As a **data scientist**, I need to **train the developed ML model** to ensure it learns from data and performs well while testing	The model should achieve a certain threshold of performance on a validation dataset during training and should not overfit or underfit the training data.	High	Second Sprint
	Model Testing	USN-5	As a **data scientist**, I need to **test the developed ML model** to ensure its accuracy in predicting rainfall.	Achieving an acceptable level of accuracy, precision,	High	Second Sprint

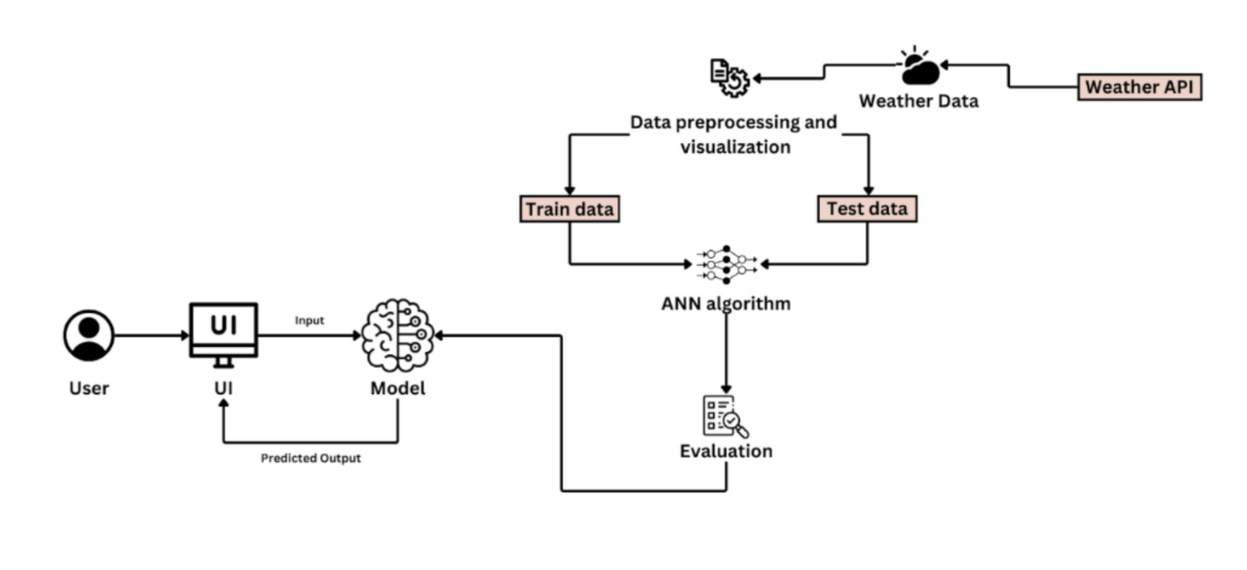
In the context of ‘Rainfall Prediction using Machine Learning’, user stories help define the roles and expectations of different stakeholders involved in the project. For instance, one of the user stories could be: “As a Data Scientist, I want to process the raw data efficiently, so that I can ensure a reliable prediction model.” This user story helps the team understand that the data scientist wants to process the raw data efficiently to ensure a reliable prediction model.

Similarly, a user story for the public could be: “As a member of the public, I want to access accurate and timely rainfall predictions through a user-friendly web app, so that I can plan my activities accordingly.” This user story emphasizes the need for accuracy, timeliness, and user-friendliness in the rainfall prediction system from the perspective of the end-user.

Each user story is associated with specific tasks, acceptance criteria, priority, and release information. These details help in planning and executing the project effectively while ensuring that all user requirements are met.

Solution Architecture

Solution architecture is a detailed description of an enterprise’s software, hardware, and network setup. It outlines how these components interact to achieve specific business goals or requirements.



The Rainfall Prediction Solution Architecture encompasses data gathering via API calls or existing datasets. Data preprocessing ensures data quality and feature extraction for modeling. Machine learning models using techniques including regression, classification, and ANN, are developed and continuously refined. Real-time rainfall predictions are generated for specific timeframes and regions, with ongoing analysis and adaptation to changing conditions. A continuous learning loop incorporates user feedback and validation data to enhance prediction accuracy, benefiting water resource management, agriculture, and disaster preparedness.

PROJECT PLANNING AND SCHEDULING

Technical Architecture

Technical architecture refers to the high-level design and structure of a system or project, detailing the arrangement of components, data flow, and the technologies used to achieve specific goals. It defines how different parts of a system interact and how data is processed, stored, and transmitted.

For our rainfall prediction project, the technical architecture can be outlined as follows:

- **Data Collection Layer:**
    - Interfaces to gather historical weather data from meteorological agencies, socioeconomic data, and indigenous knowledge from local communities.
- **Data Preprocessing Layer:**
    - Data cleaning and transformation processes to handle missing values, outliers, and inconsistencies.
    - Feature extraction and engineering to create meaningful variables for modeling.
    - Data storage and indexing for efficient access.
- **Machine Learning Layer:**
    - Development of machine learning models, including regression and classification techniques, for rainfall prediction.
    - Ensemble learning methods to combine and refine the models.
- **Prediction Engine:**
    - The prediction engine that takes input parameters and uses the trained models to generate real-time rainfall forecasts.
    - Incorporation of uncertainty estimation for prediction reliability.

Future Scope

- **Deployment and Integration:**
    - Deployment on cloud infrastructure or dedicated servers for scalability and accessibility.
    - Integration points for users, including web interfaces, APIs, and data access mechanisms.

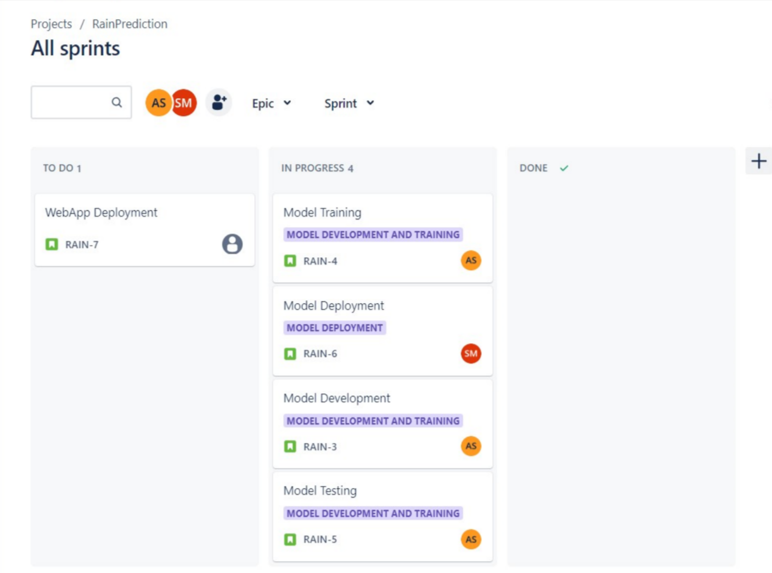
Sprint Plannning and Estimation

Sprint Planning is an event in Scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. It involves the entire team and is done in collaboration with the whole scrum team. The product owner describes the objective (or goal) of the sprint and what backlog items contribute to that goal. The development team then plans the work necessary to deliver the sprint goal.

It involves defining the ML tasks that need to be accomplished in the sprint. This includes tasks like data collection, data cleaning, feature selection, model training, testing, etc. The tasks are usually divided into three categories-'To Do, 'In Progress', and 'Done'

Sprint Estimation is the process of predicting the amount of work that can be completed in a sprint. It is a crucial part of sprint planning. The team needs to define what can or cannot be done in the sprint. It helps in evaluating the time and effort needed to complete work items in the product backlog.

For instance:



To Do: These are the tasks that need to be worked on. In the image, there is one task in this category: "WebApp Deployment". This task needs to be planned for the upcoming sprint.

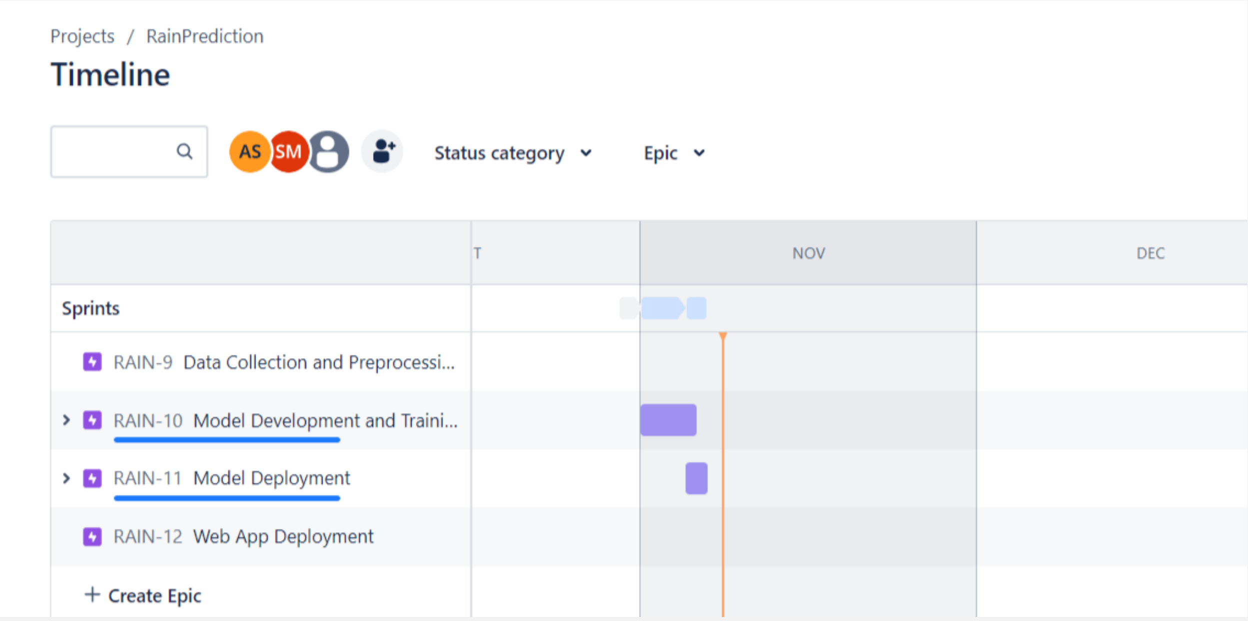
In Progress: These are the tasks that are currently being worked on. In the image, there are three tasks in this category: "Model Development and Training", "Model Deployment", and "Model Testing and Training". The progress of these tasks needs to be monitored and any blockers need to be addressed.

Done: These are the tasks that have been completed. In the image, there is one task in this category: "Epic Sprint 4". The completion of these tasks indicates the progress made in the previous sprints.

Sprint Delivery Schedule

A 'Sprint Delivery Schedule' is a crucial part of Agile project management. It outlines the timeline for the completion of tasks or user stories during each phase of the Agile Process. Its represented by a ***timeline*** on which tasks move from inception to completion. It helps in organizing the team's efforts and ensures that the project stays on track. It demands meticulous planning and regular updates to reflect the actual progress of the project.

While its important to stick to the schedule as much as possible, its equally essential to be able to adapt to unforseen challenges



Projects / RainPrediction

Timeline

Sprints

RAIN-9 Data Collection and Preprocessi...

RAIN-10 Model Development and Traini...

RAIN-11 Model Deployment

RAIN-12 Web App Deployment

CODING AND SOLUTIONING

Data pre-processing and visualization

The Rainfall Prediction project combines data preprocessing, machine learning, and a user-friendly web application to provide users with real-time rainfall predictions based on weather attributes. This comprehensive documentation outlines the key components and functionalities of the project

1. **Data Import and Attribute Categorization**
   - The project starts with importing a weather dataset from the file `weatherAUS.csv`.
   - Attributes in the dataset are categorized into numerical and categorical types for further analysis.
2. **Exploratory Data Analysis (EDA)**
   - To gain insights into the dataset, various data visualization techniques were applied.
   - Visualization tools such as `Matplotlib` and `Seaborn` were used to create visualizations.
3. **Handling Missing Data**
   - Missing data in the dataset was addressed through imputation techniques.
   - For attributes with normal distributions (e.g., `MinTemp`, `MaxTemp`, `Temp9am`, and `Temp3pm`), missing values were filled with the mean.
   - For other attributes, missing values were imputed using the median.
4. **Label Encoding for Categorical Attributes**
   - Categorical attributes such as `RainTomorrow` and `RainToday` were encoded using `LabelEncoder` to convert them into numerical format.
5. **Feature Engineering**
   - Additional features were engineered, including temporal attributes like month and day from the `Date` column.
6. **Outlier Handling**
   - Outliers were identified and treated using the Interquartile Range (IQR) method to maintain data integrity.
7. **Correlation Analysis**
   - A correlation analysis was performed to understand relationships between attributes using a heatmap.

Model Building

1. **Feature Selection**
   - The model was built using selected features, including `Location`, `MaxTemp`, `Sunshine`, `WindGustSpeed`, `Humidity9am`, `Pressure3pm`, `Cloud9am`, and `Cloud3pm`.
2. **Data Imbalance Handling**
   - Given data imbalance, the Random Oversampling technique was applied to balance the dataset.
3. **Classification Models**
   - Two classification models, Logistic Regression and XGBoost, were employed to predict rainfall.
4. **Evaluation**
   - Model performance was evaluated using metrics such as ROC-AUC, and classification reports were generated for both training and validation datasets.
5. **Neural Network Model**
   - A neural network model was developed using TensorFlow and Keras for predictive modeling.

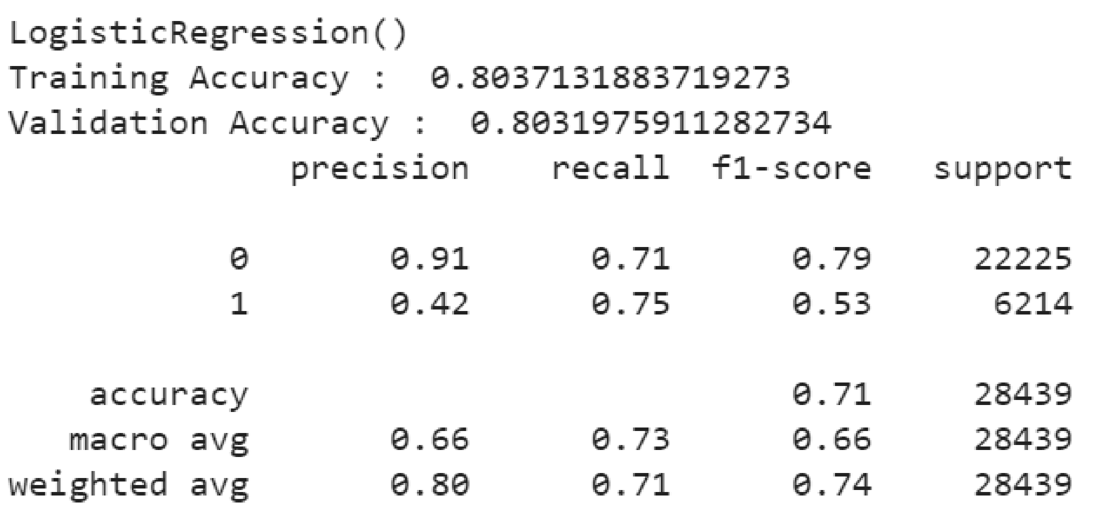
User Interface

1. **`index.html`**
   - This HTML template serves as the main user interface for the Rainfall Predictor application.
   - It includes an input form for users to enter weather attributes.
   - JavaScript populates the "Location" dropdown with city options and assigns numeric values to cities for the machine learning model.
   - After submission, the prediction result is displayed along with a corresponding message.
2. **`norain.html`**
   - Displayed when the application predicts "No Rain."
   - It provides a cheerful message and an image indicating a rain-free day.
3. **`rain.html`**
   - Displayed when the application predicts "Rain."
   - Users are informed about the chance of rain and are advised to carry an umbrella.
4. **`app.py`**
   - A Flask-based web application that integrates the machine learning model with the HTML templates.
   - It handles user interactions and predictions.
   - Key functionalities include:
     - Handling user input and predicting rainfall based on chosen weather attributes.
     - Redirecting to the appropriate HTML template depending on the prediction outcome.
   - The application can be run using the command "app.run(debug=True" to make it accessible via a web browser.

The Rainfall Prediction project combines machine learning and a user-friendly web application to provide users with real-time rainfall predictions based on weather attributes. This documentation provides a comprehensive overview of the project's components, functionalities, and data preprocessing steps.
For a more detailed and step-by-step explanation refer to "ProjectManual.pdf" in project development folder.

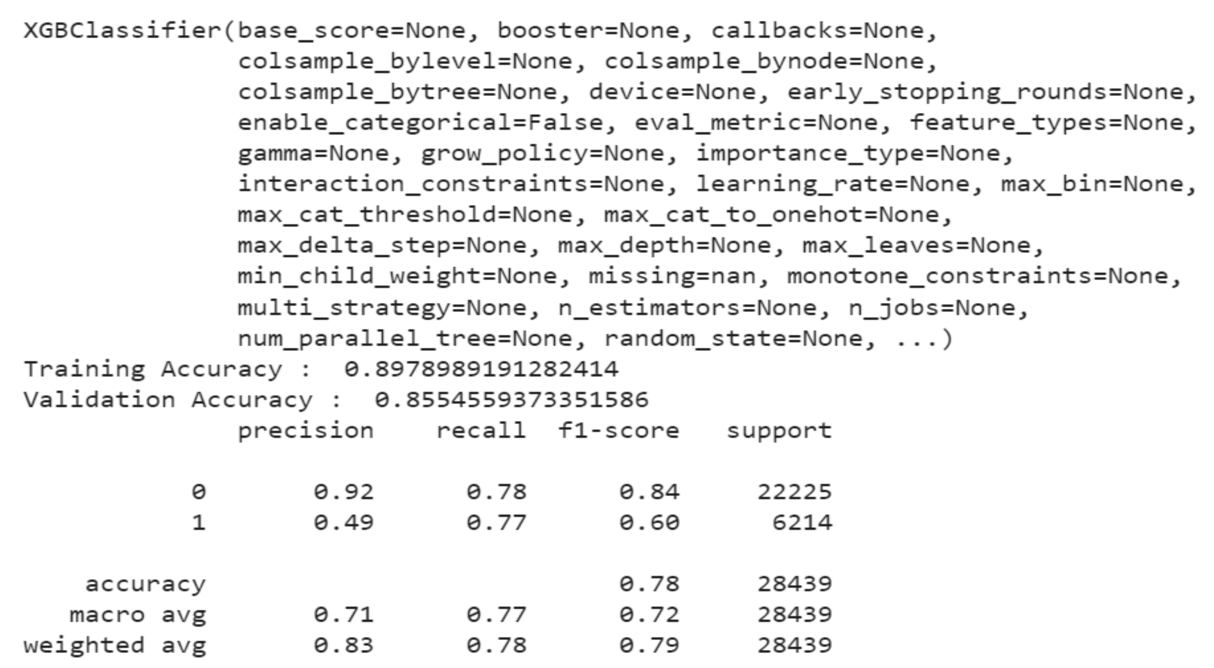
PERFORMANCE TESTING

Model 1: Logistic regression



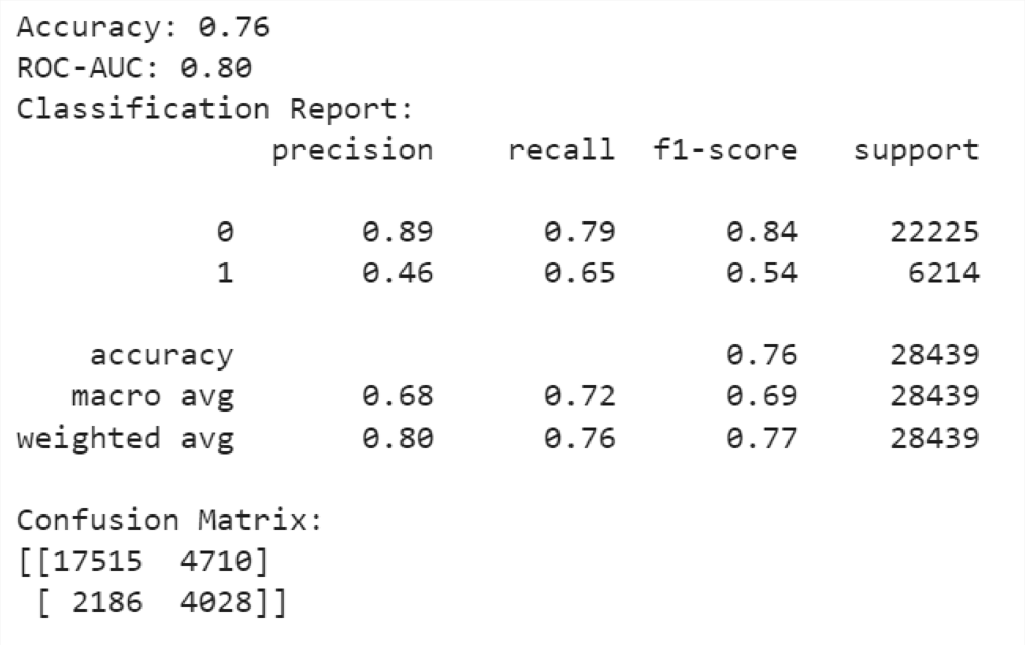
```
LogisticRegression()
Training Accuracy :  0.8037131883719273
Validation Accuracy :  0.8031975911282734
              precision    recall  f1-score   support

           0       0.91      0.71      0.79     22225
           1       0.42      0.75      0.53      6214

    accuracy                           0.71     28439
   macro avg       0.66      0.73      0.66     28439
weighted avg       0.80      0.71      0.74     28439
```

Model 2: XGBoost



```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=None,
              num_parallel_tree=None, random_state=None, ...)
Training Accuracy :  0.8978989191282414
Validation Accuracy :  0.8554559373351586
              precision    recall  f1-score   support

           0       0.92      0.78      0.84     22225
           1       0.49      0.77      0.60      6214

    accuracy                           0.78     28439
   macro avg       0.71      0.77      0.72     28439
weighted avg       0.83      0.78      0.79     28439
```

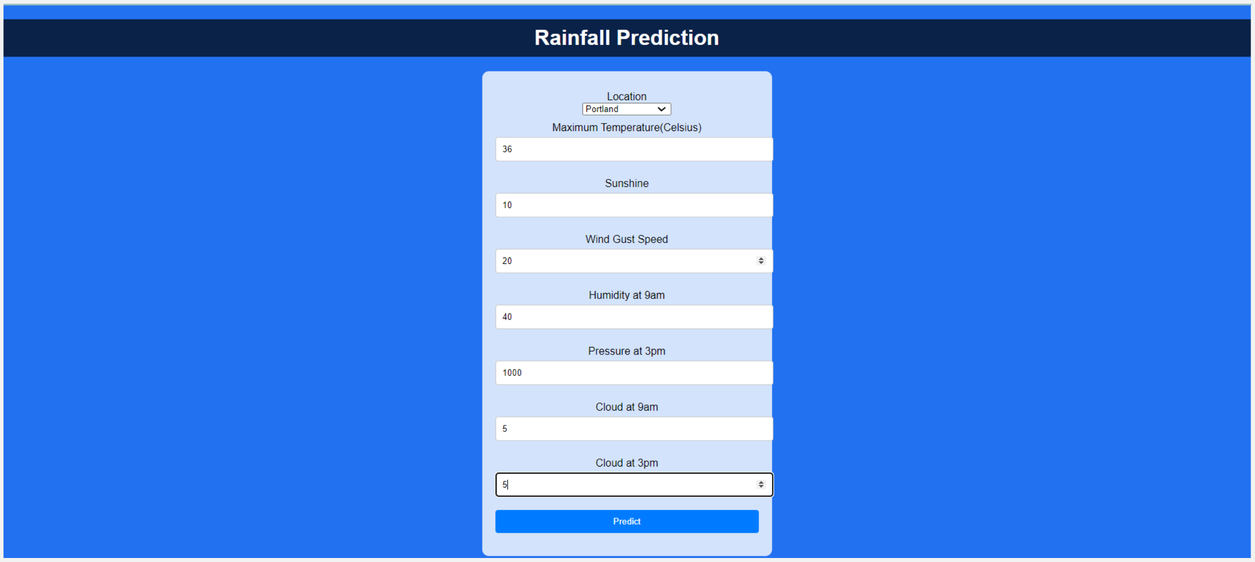
Model 3: Artifical Neural Networks



```
Accuracy: 0.76
ROC-AUC: 0.80
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.79      0.84     22225
           1       0.46      0.65      0.54      6214

    accuracy                           0.76     28439
   macro avg       0.68      0.72      0.69     28439
weighted avg       0.80      0.76      0.77     28439

Confusion Matrix:
[[17515  4710]
 [ 2186  4028]]
```

RESULTS

Input 1:



Rainfall Prediction

Location: Portland

Maximum Temperature(Celsius): 36

Sunshine: 10

Wind Gust Speed: 20

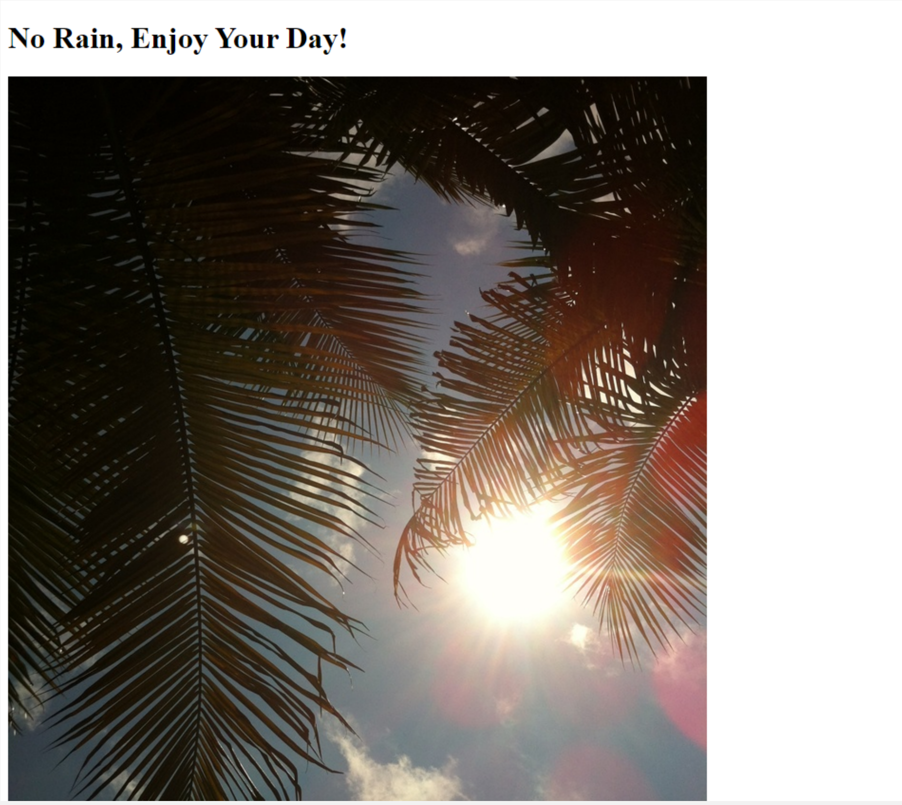
Humidity at 9am: 40

Pressure at 3pm: 1000

Cloud at 9am: 5

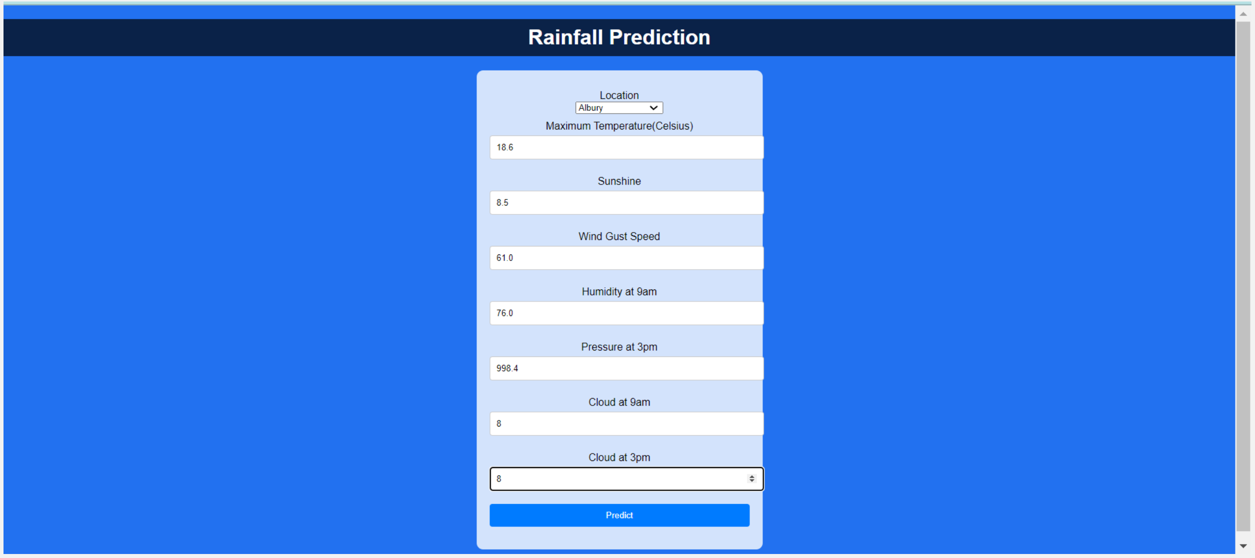
Cloud at 3pm: 5

Output:



No Rain, Enjoy Your Day!

Input 2:



Rainfall Prediction

Location
Albury

Maximum Temperature(Celsius)
18.6

Sunshine
8.5

Wind Gust Speed
61.0

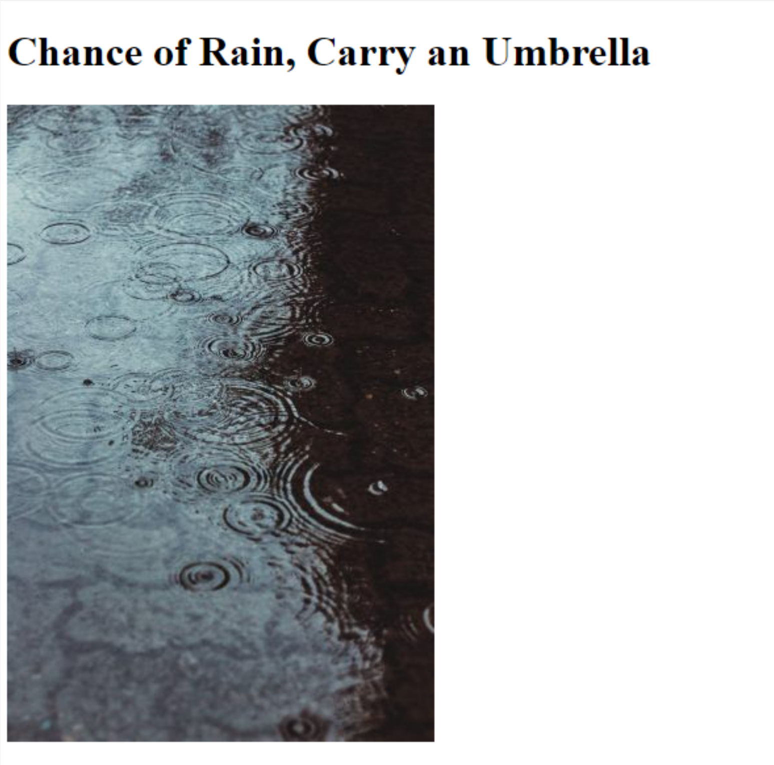
Humidity at 9am
76.0

Pressure at 3pm
998.4

Cloud at 9am
8

Cloud at 3pm
8

Output:



Chance of Rain, Carry an Umbrella

ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- Handling imbalanced data using RandomOverSampling increased the accuracy of the prediction as the model got enough instances of both classes to learn from
- User Accessibility: the webApp has no restriction as to who can access and does not require any authentication, hence, making it easier to use and understand
- Scalability: use of Flask has made it suitable to handle growing user bases
- Versatility: due to the dataset chosen, dynamic city mapping was possible which increased its relevance.
- XGBoost offers superior predictive performance compared to Logistic Regression and Artificial Neural Networks for complex, non-linear data patterns, thanks to its ensemble-based boosting technique, which minimizes both bias and variance in the model's predictions.

DISADVANTAGES:

- Use of static dataset decreases the accuracy as the dataset becomes old and does not necessarily depict the current patterns very well.
- This project is only trained on Australia's dataset and might not recognize the rainfall pattern in other geographical locations

CONCLUSION

In summary, the project adopts a model selection approach to address the challenge of rainfall prediction. By evaluating various machine learning models, including Logistic Regression, XGBoost, and a Neural Network, the project aims to identify the most accurate and reliable model for deployment.The web application's user-friendly interface, dynamic city mapping, and data visualizations enhance user accessibility and interpretability, making it a practical tool for decision-making in various fields, such as agriculture, event planning, and outdoor activities.

This model selection strategy ensures that the deployed model offers the highest predictive accuracy among the alternatives, providing users with dependable rainfall forecasts. It acknowledges the need for flexibility in model selection, considering that no single machine learning algorithm is universally superior for all types of data.While the project presents a robust solution, it is essential to remain vigilant about data quality and model maintenance to sustain the accuracy and relevance of the deployed model.

In conclusion, this project demonstrates a pragmatic approach to rainfall prediction, prioritizing predictive accuracy through a thoughtful model selection process, user accessibility, and interpretability, with an awareness of potential challenges related to data quality and ongoing maintenance.

FUTURE SCOPE

The project has achieved accurate rainfall prediction based on user input and established a straightforward user interface. However, there are several areas where further enhancements can be implemented. Following are areas of improvement:

- Use of a dynamic dataset via APIs from trusted weather information providing websites
- Use of transfer learning, so that the model doesn't need to completely re-train everytime new set of data is fetched
- Since rainfall is a time dependent factor, building a Recurrent Neural Network, with a sliding window to maintain the relevance of data, could improve the accuracy
- Integrate more cities from around the world to make it available to a greater set of users
- Deploy it on a free/paid cloud platform so that the users can access it anytime they want to and the speed of the web application is also faster

APPENDIX

GitHub Repo Link-
https://github.com/smartinternz02/SI-GuidedProject-594699-1697537002.git

Project Demo(Video)
https://drive.google.com/file/d/1fNJ2nFLfUzHBHbFT2Gb_3QRRZGx_kSlw/view?usp=sharing