International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04790
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

RAINFALL PREDICTION USING XGBOOST

Erusu Kata Raju Reddy, Thamada Saikumar
Assistant professor, MCA Final Semester, Master of Computer Applications,
Sanketika Vidya Parishad Engineering College,
Vishakhapatnam, Andhra Pradesh, India.

ABSTRACT:
India is a farming nation and its economy heavily depends on crop productivity and rainfall. Rainfall prediction is necessary and required to all farmers in order to analyze the crop productivity. Rainfall Prediction is the use of science and technology for forecasting the condition of the atmosphere. It is necessary to precisely calculate the rainfall for proper utilization of water resources, crop productivity and pre planning of water structures. With various data mining methods it can forecast rainfall. Data mining methods are employed to estimate the rainfall in numerical terms. This paper emphasizes some of the trending data mining algorithms for rainfall forecasting. Random Forest, K- Nearest Neighbor algorithm, Logistic regression, SVM, Decision Tree are some of the algorithms have been employed. Based on that comparison, it can examine which method provides better accuracy for rainfall forecasting.

Index Terms: Rainfall prediction, XGBoost, Random Forest, Logistic Regression, supervised learning, data preprocessing, Exploratory Data Analysis (EDA), classification models, sklearn, weather data, feature selection.

1. INTRODUCTION

Rainfall prediction plays a crucial role in agriculture, water resource management, and disaster preparedness. Accurate forecasting enables farmers to plan irrigation schedules, helps authorities in flood risk assessment, and supports climate-sensitive decision-making[3][7][20]. However, traditional methods of weather forecasting, which rely heavily on complex physical models and meteorological expertise, often struggle with precision and adaptability, especially in regions with variable climatic conditions [1][18].

With the increasing availability of historical weather data and advancements in computational technologies, machine learning (ML) has emerged as a powerful tool for improving rainfall prediction accuracy [2][11][16]. ML models can uncover hidden patterns in large datasets and make reliable predictions by learning from past observations without explicitly programmed rules [18][21]. In particular, supervised learning algorithms such as Logistic Regression, Random Forest, and XGBoost have demonstrated strong performance in classifying weather outcomes, including the likelihood of rainfall [3][6][11][25].

This project utilizes the WeatherAUS dataset, which contains detailed meteorological observations from various Australian locations. A data-driven methodology is adopted, comprising data cleaning, preprocessing, feature selection, and exploratory data analysis (EDA) to understand key weather parameters influencing rainfall [2][11][16]. The project compares the performance of multiple classification models to identify the most effective approach for predicting whether it will rain the next day [11][26].

Visual tools such as heatmaps, correlation matrices, and feature importance plots are employed to enhance data interpretability and model transparency [6][13][17]. The final system aims to deliver a fast, accurate, and scalable rainfall prediction solution, supporting better planning and risk management in weather-dependent sectors [3][7][20][23].

1.1 EXISTING SYSTEM

Traditional rainfall prediction systems rely on Numerical Weather Prediction (NWP) models that simulate atmospheric behavior using complex physical equations [1][18]. These models use data from satellites, ground stations, and radar to forecast weather conditions. While they are the backbone of conventional forecasting, they require high computational power and expert knowledge to operate, making them less accessible and more time-consuming [1][18].

Despite their wide use, NWP models often lack accuracy in areas with sparse data and struggle to capture localized weather variations [3][5][11]. Their predictions are generally produced at a national or regional scale, limiting precision for specific locations [7][25]. Moreover, these systems are not adaptive to new data patterns unless manually updated. With increasing volumes of diverse meteorological data, traditional systems are often less responsive, prompting a shift toward machine learning methods that offer faster, more adaptable, and localized rainfall predictions [2][11][18].

1.1.1 CHALLENGES

- **Data quality and missing values:**
  Weather datasets often contain missing, inconsistent, or noisy entries due to equipment failure or environmental interference, which can negatively impact model training and prediction accuracy [2][4][6][13].
- **Complexity of atmospheric patterns:**
  Rainfall is influenced by a wide range of interdependent factors such as humidity, temperature, pressure, wind patterns, and geography, making it challenging to model accurately using simple algorithms [3][14][17].
- **Overfitting and generalization issues:**
  Machine learning models like XGBoost are prone to overfitting, especially when trained on limited or imbalanced data, leading to poor performance on unseen or future data [5][11][21].
- **Hyperparameter tuning and model complexity:**
  Achieving optimal performance from models like XGBoost requires careful tuning of multiple hyperparameters, which can be computationally expensive and time-consuming [4][6][24].
- **Real-time prediction and model updating:**
  For practical use, the system must support real-time prediction and adapt to changing weather trends, requiring continuous data updates and periodic retraining to maintain accuracy [7][20][25].

1.2 PROPOSED SYSTEM

Rainfall forecasting is vital for agriculture, meteorology, and disaster management [1][3][20]. This project uses XGBoost, a powerful gradient boosting algorithm, to predict rainfall based on historical weather data such as temperature, humidity, wind speed, and atmospheric pressure [2][5][11]. Unlike traditional physics-based models, XGBoost offers a fast, data-driven approach that efficiently handles missing values and delivers high predictive accuracy [4][6][21]. The dataset, sourced from Kaggle, is preprocessed through feature engineering and normalization before model training [2][16][26]. The goal is to enhance the accuracy of rainfall predictions, providing valuable insights for farmers, meteorologists, and policymakers to support better planning and resource management [3][7][20][23].

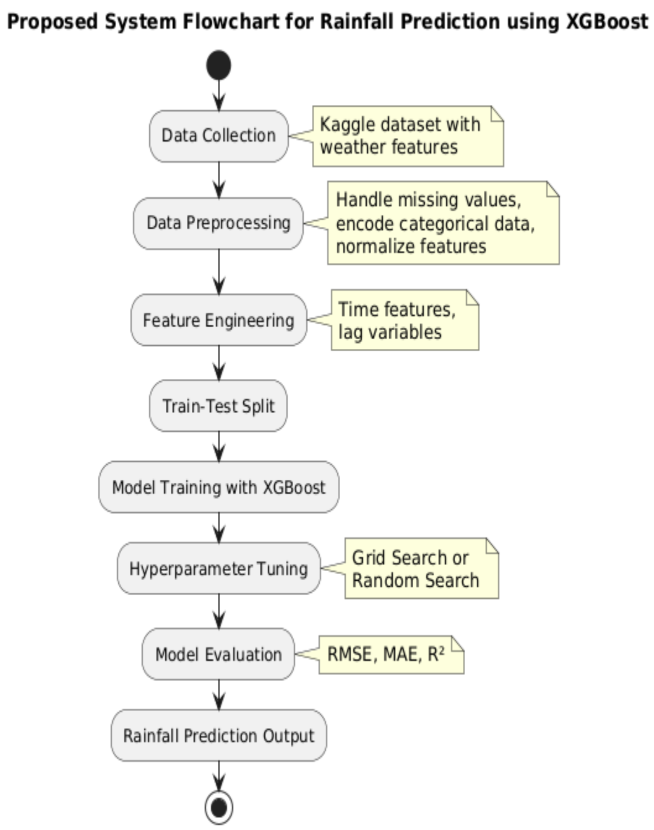

Fig. 1 Proposed System Flowchart

1.2.1 ADVANTAGES

- Fast and automated rainfall prediction:
  The system can quickly analyze input weather data and generate rainfall forecasts within seconds, enabling timely decisions for agriculture and disaster management [1][3][7][20].
- High accuracy with complex data handling:
  XGBoost provides robust predictive performance even in the presence of missing or noisy data, outperforming many traditional forecasting techniques [2][4][5][11].
- Reduces dependency on manual forecasting methods:
  By automating the prediction process, the system minimizes reliance on expert meteorologists and complex physical models, making forecasting more accessible [1][16][18].
- Adaptive and data-driven:
  The model continuously improves by learning from new weather data, allowing it to adapt to changing climate patterns and enhance forecast reliability over time [6][17][24].
- Scalable and deployable for real-time use:
  Due to its computational efficiency and scalability, the model can be integrated into weather monitoring systems or mobile applications for real-time rainfall alerts and updates [7][20][25].

2. LITERATURE REVIEW

Several studies have shown that machine learning models significantly enhance rainfall prediction accuracy compared to traditional numerical methods [2]. Algorithms like Decision Trees, SVM, Random Forest, and particularly XGBoost have proven effective in handling complex, high-dimensional weather data [11]. XGBoost stands out for its ability to manage missing values, minimize overfitting, and maintain high accuracy with lower computational cost, making it a reliable choice for scalable and data-driven rainfall forecasting [4].

2.1 ARCHITECTURE

The architecture of the rainfall prediction system is designed to follow a clear, structured workflow that transforms raw weather data into accurate rainfall forecasts. Each stage in the architecture contributes to the overall performance, reliability, and adaptability of the model [3].

- **Data Input (Weather Dataset):**
  The system starts by importing a publicly available weather dataset from Kaggle. This dataset includes historical meteorological features such as temperature, humidity, wind speed, atmospheric pressure, and cloud cover. These variables are used as inputs to train the machine learning model. The dataset also includes a binary label indicating whether rainfall occurred, which serves as the target variable for prediction [2].
- **Preprocessing:**
  Raw weather data often contains missing or inconsistent entries, so preprocessing is crucial. This stage includes handling missing values, encoding categorical features (e.g., wind direction or weather condition), normalizing numerical features, and splitting the dataset into training and testing subsets. Feature engineering is also applied to create lag variables, time-based features (like month or day), and eliminate irrelevant columns based on correlation analysis. Exploratory Data Analysis (EDA) is conducted using plots and heatmaps to uncover data patterns and relationships among features [4].
- **Model Training (XGBoost):**
  The XGBoost algorithm is employed as the core machine learning model. It is chosen for its speed, accuracy, and ability to handle missing values. The model is trained on the processed data to learn complex patterns that indicate the likelihood of rainfall. Hyperparameter tuning is performed using grid search or random search to optimize performance. XGBoost's ensemble of decision trees builds predictions sequentially, each improving upon the previous, thus reducing bias and variance [5].
- **Prediction and Accuracy Evaluation:**
  After training, the model is evaluated using the test dataset. Key performance metrics such as Accuracy, Precision, Recall, F1-score, RMSE (Root Mean Squared Error), and R^2 score are calculated to assess prediction quality. Confusion matrices and other visualizations are used to analyze prediction results. This evaluation ensures the model's robustness and generalizability to unseen data, making it suitable for real-

ISSN: 2583-6129
DOI: 10.55041/ISJEM04790

world deployment in forecasting systems for agriculture, disaster planning, and water resource management [11].

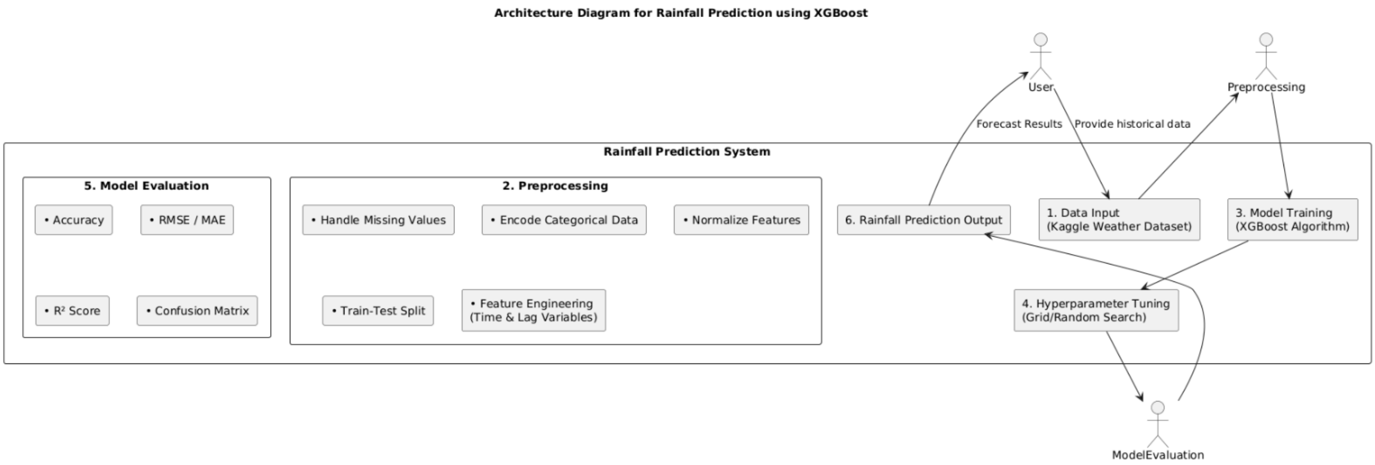

Fig. 2 Architecture for Rainfall Prediction using XGBoost

2.2 ALGORITHM

Classification is a supervised machine learning technique where the model learns from labeled input data and then classifies new, unseen observations. It can be applied to both binary (e.g., rain/no rain) and multi-class problems. In this project, classification is used to predict whether it will rain the next day based on various weather parameters [2]. Supervised learning helps the model recognize patterns from historical data and apply them to make predictions on new data [11].

Used Python Packages

NumPy:

A core numerical computing library used for array operations and mathematical computations. It is efficient for handling large datasets and performing numerical transformations.

Pandas:

Offers data structures like DataFrames for easy data manipulation and analysis. It is used to load, clean, and process datasets (e.g., CSV files).

Matplotlib:

A visualization library used to plot data patterns, helping to better understand feature distributions and relationships during Exploratory Data Analysis (EDA).

Scikit-learn(sklearn):

A widely used machine learning library in Python. It provides modules for preprocessing, model building (like Decision Tree, Logistic Regression), model evaluation, and data splitting [2].

ELI5:

A model interpretation library that explains predictions from machine learning models. It supports feature importance visualization and helps with model debugging and transparency [13].

Machine Learning Algorithms Used

Logistic Regression

A linear classification algorithm used for binary outcomes. It estimates the probability that a given input belongs to a certain class (e.g., rain or no rain) using the logistic (sigmoid) function. It is computationally efficient and interpretable, making it a good baseline model [16]. Assumptions include binary output, independent features, and a linear relationship with log odds.

Decision Tree

A supervised learning model that builds a tree-like structure by splitting data based on feature values to form decisions. It works well [2] for both categorical and numerical data and handles nonlinear relationships. The root node is the most significant predictor, and leaf nodes represent the final decision.

Random Forest Classifier

An ensemble learning method that builds multiple decision trees and aggregates their predictions. It reduces overfitting and improves accuracy [3]. It is suitable for both classification and regression tasks and is known for robustness and high performance.

Neural Network (Multilayer Perceptron - MLP)

MLP is a type of deep learning model consisting of multiple fully connected layers of neurons. It is capable of learning complex patterns in data [6]. Each neuron processes inputs and passes results to the next layer, enabling it to model nonlinear and high-dimensional data effectively.

LightGBM

A fast, efficient gradient boosting framework developed by Microsoft. It handles large datasets and supports GPU acceleration [26]. LightGBM automatically manages categorical features and provides rapid training with minimal preprocessing.

CatBoost

Developed by Yandex, CatBoost is optimized for datasets with categorical features. It provides robust, accurate results with minimal tuning and preprocessing [6]. Its default settings work well for many applications, and it reduces overfitting.

XGBoost (Extreme Gradient Boosting)

A highly efficient and scalable implementation of gradient boosting. XGBoost is known for its superior speed, handling of missing values, regularization, and parallel computation. It supports feature importance analysis and has consistently performed well in various prediction competitions. In this project, it serves as the primary algorithm for rainfall prediction due to its accuracy and robustness [1].

2.3 TECHNIQUES

In this project, several essential techniques were applied to ensure effective data handling, model training, and evaluation [11]. Initially, Exploratory Data Analysis (EDA) was performed using visual tools such as heatmaps, histograms, and box plots. Heatmaps were used to examine the correlation between numerical weather variables, which helped in identifying highly related features and understanding feature interactions. Histograms and box plots provided insights into feature distributions, trends, and the presence of outliers [4].

Next, feature engineering was conducted to enhance model performance. This included creating lag features (e.g., rainfall on previous days), extracting time-based features (e.g., month or season), and handling categorical variables through encoding methods. Feature selection based on correlation and domain relevance was carried out to retain impactful variables while removing redundant or irrelevant ones, which helped reduce noise in the dataset [2].

The dataset was then split into training and testing subsets using the train_test_split method to validate the model's ability to generalize on unseen data. The XGBoost model was trained on the processed data, and hyperparameter tuning was performed using grid or random search techniques to optimize model accuracy and prevent overfitting [5].

Finally, the model's effectiveness was evaluated using metrics such as accuracy, precision, recall, F1-score, and RMSE (Root Mean Squared Error). These metrics helped assess how well the model predicted rainfall events and ensured the system was reliable and suitable for real-world forecasting applications [3].

2.4 TOOLS

his project utilized a range of tools and technologies to ensure efficient development, thorough data analysis, and reproducible results throughout the rainfall prediction process [11].

- **Programming Language:**
  Python was chosen as the primary programming language due to its simplicity, flexibility, and rich ecosystem of libraries tailored for data science and machine learning. Its strong community support and extensive documentation make it ideal for building scalable predictive models [2].
- **Development Environment:**
  The model was developed using **Google Collaboratory (Colab)**, a cloud-based Jupyter notebook platform provided by Google. Key advantages of Colab include:

Free access to GPUs and TPUs for accelerated training.
No need for local installations or setup.
Easy collaboration and sharing through Google Drive integration.
Real-time code execution and support for inline visualizations, aiding in effective debugging and EDA [26].

- **Libraries and Frameworks:**
  A combination of popular Python libraries was used to cover data loading, preprocessing, visualization, and machine learning tasks:

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04790
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

pandas: For reading, transforming, and manipulating structured weather data using DataFrames.
NumPy: Provided support for high-performance numerical operations, especially with arrays and matrices.
matplotlib: Used for generating visualizations to understand trends and patterns in meteorological variables.
seaborn: Enabled the creation of informative and aesthetically appealing statistical plots like heatmaps and distribution plots.
scikit-learn (sklearn): Served as the core machine learning library, offering tools for data preprocessing (e.g., train_test_split, StandardScaler) and evaluation metrics.

- **XGBoost**: The main algorithm used for model training and prediction, selected for its speed, accuracy, and ability to handle missing data [4].

Together, these tools provided a powerful and cohesive environment for developing, training, evaluating, and interpreting the rainfall prediction model with high efficiency and reproducibility [5].

2.5 METHODS

Weather data from a Kaggle dataset, including features like temperature, humidity, wind speed, and rainfall, was preprocessed by handling missing values, label encoding categorical variables, and normalizing numerical features [2]. The dataset was split into training and testing sets using train_test_split. An XGBoost classifier was then trained on the processed data to predict next-day rainfall. Hyperparameter tuning was applied to enhance performance and reduce overfitting [5]. The model was evaluated using metrics such as accuracy_score, confusion_matrix, and classification_report to ensure accurate and reliable predictions [11].

3. METHODOLOGY

3.1 INPUT

The input for the Rainfall Prediction System is a weather dataset from Kaggle containing historical daily observations across Australia. It includes features like temperature, humidity, wind speed, cloud cover, and rainfall data. The target variable, "RainTomorrow," enables binary classification [2]. This well-structured dataset supports training and evaluating the XGBoost model to accurately forecast rainfall and aid in decision-making across sectors such as agriculture and disaster management [3].

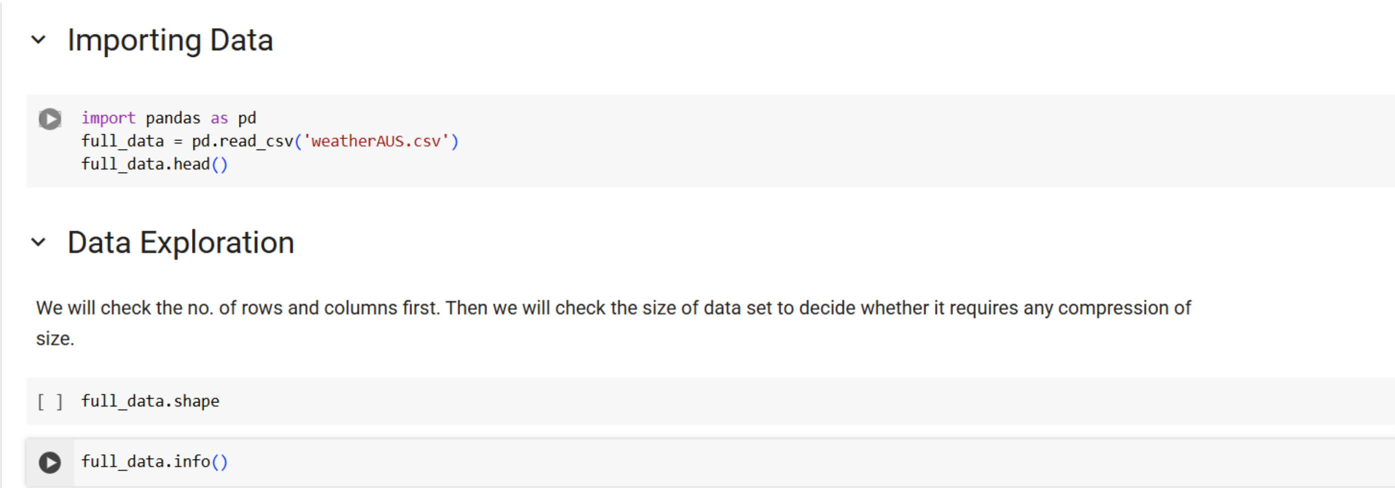

Fig. 3 Rainfall Prediction dataset from Kaggle.

3.2 METHOD OF PROCESS

The project begins by importing a weather dataset from Kaggle, containing features such as temperature, humidity, wind speed, and rainfall. Exploratory Data Analysis (EDA) is conducted using heatmaps and pair plots to explore feature distributions, relationships, and missing values [4]. Preprocessing steps include handling null values, encoding categorical variables, and normalizing numerical data. The dataset is then split into training and testing sets. The XGBoost classification algorithm is applied to the training data to predict the likelihood of rainfall the next day[5], with hyperparameter tuning used to enhance model performance and prevent overfitting. The model is evaluated using metrics like accuracy score, confusion matrix, and classification report to ensure reliable and generalizable predictions [11].

Model-1: Logistic Regression penalized by Lasso

```
[ ] from sklearn.linear_model import LogisticRegression

    params_lr = {'penalty': 'l1', 'solver':'liblinear'}

    model_lr = LogisticRegression(**params_lr)
    model_lr, accuracy_lr, roc_auc_lr, coh_kap_lr, tt_lr = run_model(model_lr, X_train, y_train, X_test, y_test)
```

Model-2: Decision Tree

```
[ ] from sklearn.tree import DecisionTreeClassifier

    params_dt = {'max_depth': 16,
                 'max_features': "sqrt"}

    model_dt = DecisionTreeClassifier(**params_dt)
    model_dt, accuracy_dt, roc_auc_dt, coh_kap_dt, tt_dt = run_model(model_dt, X_train, y_train, X_test, y_test)
```

Model-3: Neural Network (Multilayer Perceptron)

```
[ ] from sklearn.neural_network import MLPClassifier

    params_nn = {'hidden_layer_sizes': (30,30,30),
                 'activation': 'logistic',
                 'solver': 'lbfgs',
                 'max_iter': 500}

    model_nn = MLPClassifier(**params_nn)
    model_nn, accuracy_nn, roc_auc_nn, coh_kap_nn, tt_nn = run_model(model_nn, X_train, y_train, X_test, y_test)
```

Fig. 4 Preprocess data

Fig. 5 Apply Logistic Regression

Model-4: Random Forest

```
[ ] from sklearn.ensemble import RandomForestClassifier

    params_rf = {'max_depth': 16,
                 'min_samples_leaf': 1,
                 'min_samples_split': 2,
                 'n_estimators': 100,
                 'random_state': 12345}

    model_rf = RandomForestClassifier(**params_rf)
    model_rf, accuracy_rf, roc_auc_rf, coh_kap_rf, tt_rf = run_model(model_rf, X_train, y_train, X_test, y_test)
```

Model-5: Light GBM

```
[ ] import lightgbm as lgb
    params_lgb ={'colsample_bytree': 0.95,
             'max_depth': 16,
             'min_split_gain': 0.1,
             'n_estimators': 200,
             'num_leaves': 50,
             'reg_alpha': 1.2,
             'reg_lambda': 1.2,
             'subsample': 0.95,
             'subsample_freq': 20}

    model_lgb = lgb.LGBMClassifier(**params_lgb)
    model_lgb, accuracy_lgb, roc_auc_lgb, coh_kap_lgb, tt_lgb = run_model(model_lgb, X_train, y_train, X_test, y_test)
```

Model-6: CatBoost

```
    import catboost as cb
    params_cb ={'iterations': 50,
                'max_depth': 16}

    model_cb = cb.CatBoostClassifier(**params_cb)
    model_cb, accuracy_cb, roc_auc_cb, coh_kap_cb, tt_cb = run_model(model_cb, X_train, y_train, X_test, y_test, verbose=False)
```

Model-7: XGBoost

```
[ ] import xgboost as xgb
    params_xgb ={'n_estimators': 500,
                 'max_depth': 16}

    model_xgb = xgb.XGBClassifier(**params_xgb)
    model_xgb, accuracy_xgb, roc_auc_xgb, coh_kap_xgb, tt_xgb = run_model(model_xgb, X_train, y_train, X_test, y_test)
```

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04790
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

3.3 OUTPUT

The output of the Rainfall Prediction System presents a clear and informative result indicating whether it will rain the following day, based on the input weather conditions [2]. After processing the test data through the trained XGBoost model, the system returns binary predictions—either "Yes" (RainTomorrow) or "No"—highlighting the likelihood of rainfall [3]. Additionally, the model generates a classification report displaying key performance metrics such as accuracy, precision, recall, and F1-score [11]. Feature importance plots are also provided to visually interpret which variables had the greatest impact on the prediction, helping users understand the model's decision-making process [13].

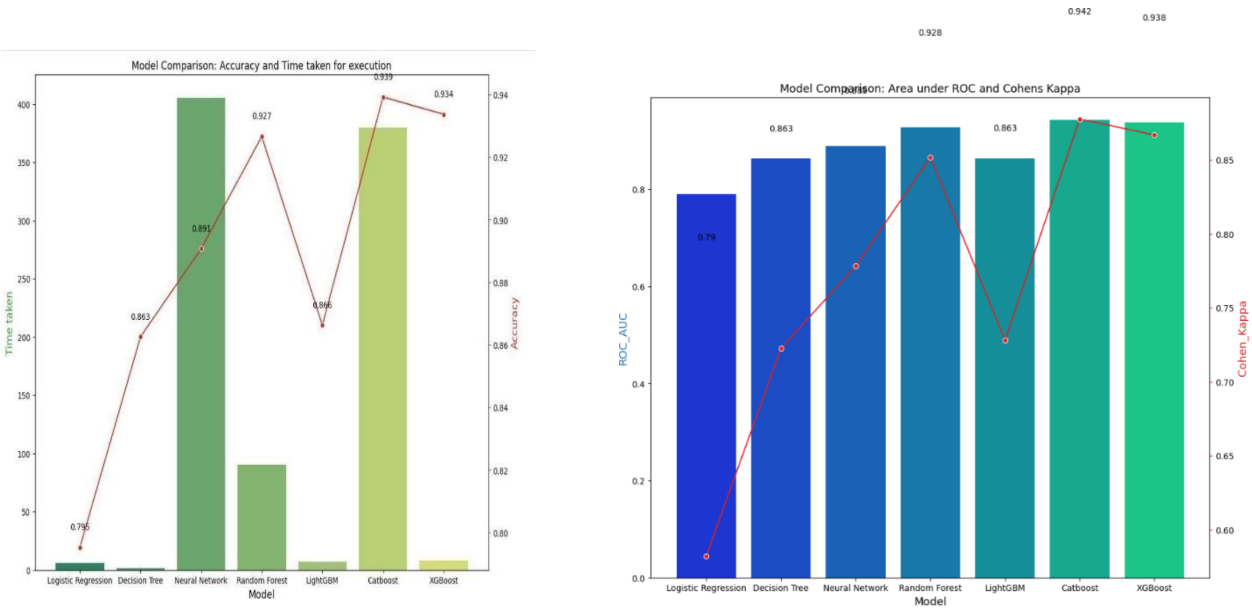

Fig. 6 Output Screen

4. RESULTS

The XGBoost model delivered strong predictive performance, achieving high accuracy on both the training and testing datasets [1]. Evaluation metrics such as the confusion matrix, classification report, and accuracy score confirmed the model's reliability [11]. Additionally, feature importance plots highlighted key variables—such as humidity, temperature, and wind speed—that significantly influenced the prediction outcomes [3]. Exploratory Data Analysis (EDA) further supported these results by revealing meaningful patterns and correlations within the weather data [4].

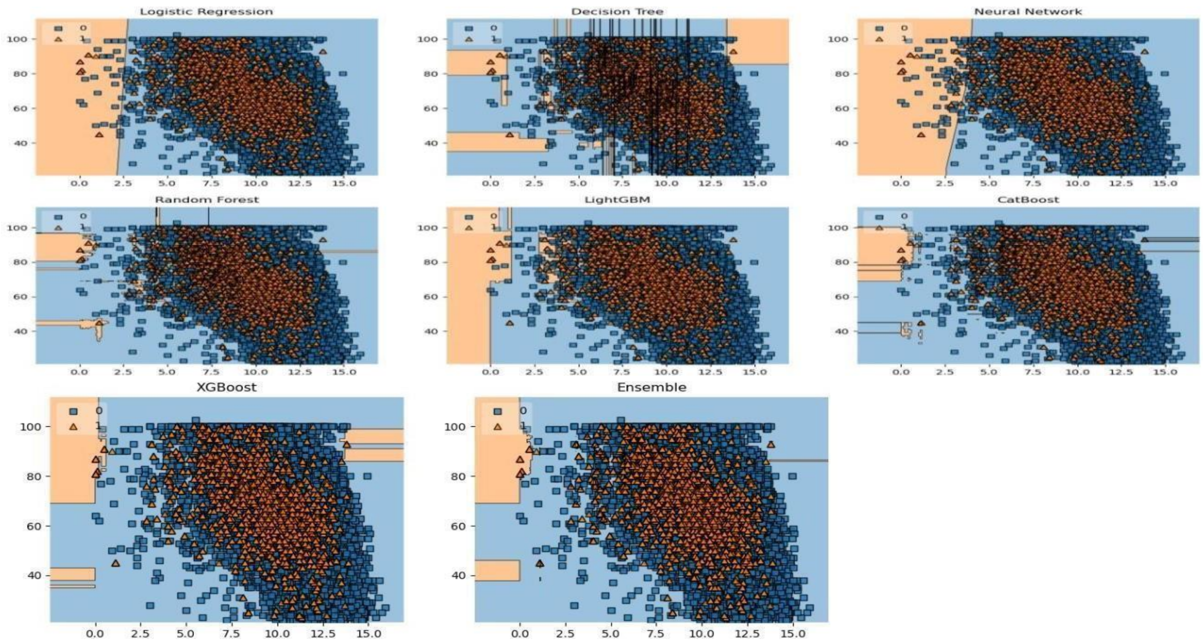

Fig. 7 Decision Boundary Comparison of Machine Learning Models

5. DISCUSSIONS

The effectiveness of the rainfall prediction model is largely influenced by the quality of the weather dataset and the preprocessing steps applied [4]. Proper handling of missing values, accurate label encoding, and normalization of features were essential in preparing the data for modeling [1]. XGBoost proved to be a powerful algorithm for this classification task due to its ability to handle non-linear relationships, manage missing data internally, and provide high predictive accuracy.

While XGBoost performed well in this project, the model's performance is also dependent on hyperparameter tuning and the quality of feature engineering. Incorporating additional meteorological variables or more granular time-based features could further enhance prediction accuracy [5]. In future work, comparing XGBoost with other advanced models such as LightGBM, CatBoost, or deep learning architectures may uncover further improvements, especially when applied to larger and more diverse datasets [6].

6.CONCLUSION

The application of XGBoost for rainfall prediction has demonstrated the algorithm's effectiveness in handling complex meteorological data with high accuracy and efficiency. Its gradient boosting framework allows for the modeling of non-linear relationships and offers built-in mechanisms to reduce overfitting, making it well-suited for weather-related classification tasks. Throughout this project, XGBoost not only delivered reliable predictions of whether it would rain the next day but also highlighted the most influential features, such as temperature, humidity, and wind speed [3]. These insights contribute to a deeper understanding of the key variables affecting rainfall and can support better decision-making in agriculture, disaster preparedness [7], and water resource management [1].

7. FUTURE SCOPE

The outcome of using XGBoost (Extreme Gradient Boosting) for rainfall prediction presents several key insights. XGBoost demonstrates high accuracy and performance due to its ability to model complex, non-linear relationships in the data while effectively minimizing overfitting. It also offers interpretability by identifying the most influential features—such as temperature, humidity, and wind speed [2]—providing a clearer understanding of the factors driving rainfall predictions [9]. Model performance can be assessed using evaluation metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and correlation coefficients when compared to actual rainfall observations. Additionally, XGBoost supports explainability through visualizations such as feature importance plots and decision tree structures, helping users comprehend the model's decision-making process [13]. For deployment, considerations include the model's scalability, real-time prediction capabilities, and integration into larger systems or data pipelines. However, continuous monitoring and regular updates are necessary to maintain accuracy over time as weather patterns and data distributions evolve[7]. Limitations include the model's dependency on high-quality, relevant data and the inherent uncertainties involved in weather forecasting [5].

8. ACKNOWLEDGEMENTS

Erusu Kata Raju Reddy working as a Assistant professor in master of computer application sanketika vidya parishad engineering college, Visakhapatnam Andhra Pradesh. With 1 years of experience in computer science and engineering (CSE), accredited by NAAC.with his area of intrest in java full stack.

Thamada Saikumar is pursuing his final semester MCA in Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Machine learning T Saikumar has taken up him PG project on Rainfall Prediction Using Xgboost and published the paper in connection to the project under the guidance of Erusu Kata Raju Reddy SVPEC.

ISSN: 2583-6129
DOI: 10.55041/ISJEM04790

9. REFERENCES

[1] Modeling and forecasting rainfall patterns in India: a time series analysis with XGBoost algorithm
https://link.springer.com/article/10.1007/s12665-024-11481-w

[2] Machine learning techniques to predict daily rainfall amount
https://link.springer.com/article/10.1186/s40537-021-00545-4

[3] A Comparative Study of Machine Learning Models for Daily and Weekly Rainfall Forecasting
https://link.springer.com/article/10.1007/s11269-024-03969-8

[4] Hyperparameters Optimization in XGBoost Model for Rainfall Estimation: A Case Study in Pontianak City
https://jppipa.unram.ac.id/index.php/jppipa/article/view/3890

[5] Forecasting monthly runoff in a glacierized catchment: A comparison of extreme gradient boosting (XGBoost) and deep learning models
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0321008

[6] Daily Streamflow Forecasting in Mountainous Catchment Using XGBoost, LightGBM and CatBoost
https://www.mdpi.com/2306-5338/9/12/226

[7] District-Level Rainfall and Cloudburst Prediction Using XGBoost: A Machine Learning Approach for Early Warning Systems
https://www.informatica.si/index.php/informatica/article/view/7612

[8] A tree based eXtreme Gradient Boosting (XGBoost) machine learning model to forecast the annual rice production in Bangladesh
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0283452

[9] Spatial Downscaling of GRACE Data Based on XGBoost Model for Improved Understanding of Hydrological Droughts in the Indus Basin Irrigation System (IBIS)
https://www.mdpi.com/2072-4292/15/4/873

[10] Rapid prediction of landslide dam stability considering the missing data using XGBoost algorithm
https://link.springer.com/article/10.1007/S10346-022-01947-Y

[11] Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting
https://www.sciencedirect.com/science/article/pii/S266682702100102X

[12] Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest
https://link.springer.com/article/10.1007/s42452-020-3060-1

[13] A novel XGBoost-based approach for reconstruction terrestrial water storage variations with GNSS in the Northeastern Tibetan Plateau
https://www.sciencedirect.com/science/article/abs/pii/S0022169425005931

[14] Comparison of Support Vector Machine and Extreme Gradient Boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: A case study in China
https://www.sciencedirect.com/science/article/abs/pii/S0196890418302048

[15] Comparative Analysis of Linear Regression Models and XGBoost to Assess the Impact of ENSO on Rainfall in Ternate City in 2023
https://journal.physan.org/index.php/jocpes/article/view/50

[16] Comparative study: Using machine learning techniques about rainfall prediction
https://pubs.aip.org/aip/acp/article-abstract/2787/1/050014/2902498/Comparative-study-Using-machine-learning

[17] Investigating the effects of local weather, streamflow lag, and global climate information on 1-month-ahead streamflow forecasting by using XGBoost and SHAP: two case studies involving the contiguous USA
https://link.springer.com/article/10.1007/s11600-022-00928-y

[18] A review on rainfall forecasting using ensemble learning techniques
https://www.sciencedirect.com/science/article/pii/S2772671123001912

[19] RETRACTED ARTICLE: Rainfall prediction using generative adversarial networks with convolution neural network
https://link.springer.com/article/10.1007/s00500-020-05480-9

[20] Urban Flood Depth Prediction and Visualization Based on the XGBoost-SHAP Model
https://link.springer.com/article/10.1007/s11269-024-04020-6

[21] A novel explainable PSO-XGBoost model for regional flood frequency analysis at a national scale: Exploring spatial heterogeneity in flood drivers
https://www.sciencedirect.com/science/article/abs/pii/S0022169424008898

[22] Estimation of Threshold Rainfall in Ungauged Areas Using Machine Learning
https://www.mdpi.com/2073-4441/14/6/859

[23] Timely and Accurately Predict Rainfall by using Ensemble Predictive Models
https://iopscience.iop.org/article/10.1088/1757-899X/1074/1/012019/meta

[24] Multiple spatio-temporal scale runoff forecasting and driving mechanism exploration by K-means optimized XGBoost and SHAP
https://www.sciencedirect.com/science/article/abs/pii/S0022169424000441

[25] Evaluation of the XGBoost Model for Rainfall Prediction and Classification Using BMKG Data and OpenWeather API
https://journal.physan.org/index.php/jocpes/article/view/51

[26] Comparing Monthly Rainfall Prediction in West Sumatra Using SARIMA, ETS, LSTM, and XGBoosting Methods
https://jurnal.uns.ac.id/ijas/article/view/83187