Comparative analysis of advanced machine learning classifiers based on feature engineering framework for weather prediction

[*]Rahul Gupta[1,2], Anil Kumar Yadav[3], SK Jha[4], Pawan Kumar Pathak[5]

[1] Department of Electrical Engineering, Netaji Subhas University of Technology, Dwarka, New Delhi, 110078, India. **e-mail**: rahul.ee20@nsut.ac.in (*Corresponding Author) **Contact no**: +91-8979566530

[2] Department of Electrical and Electronics Engineering, G L Bajaj Institute of Technology and Management, Greater Noida, Uttar Pradesh, 201310, India. **e-mail**: rahul.gupta@glbitm.ac.in

[3] Department of Instrumentation & Control Engineering, Dr B R Ambedkar National Institute of Technology Jalandhar, Punjab-144008, India. **e-mail**: anilei007@gmail.com

[4] Department of ICE, Netaji Subhas University of Technology, Dwarka, New Delhi, 110078, India. **e-mail:** skjha@nsut.ac.in

[5] School of Automation, Banasthali Vidyapith, Rajasthan, 304022, India. **e-mail**: ppathak999@gmail.com

Abstract. Significant climatic change is a really difficult task that affects people all across the world. Rainfall is considered one of the most significant phenomena in the weather system, and its rate is one of the most crucial variables. To develop a prediction model by standard approaches, meteorological experts attempt to detect the atmospheric attributes such as sunlight, temperature, humidity and cloudiness etc. Machine Learning (ML) techniques are recently more evolved which provides results that are more satisfactory than those of traditional methods and are simple to use. This paper presents the ML classifiers such as Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Light Gradient Boost Machine (LGBM), Cat Boost (CB), and Extreme Gradient Boost (XGB) to predict the rainfall using feature engineering framework. The Area under the Receiver Operating Characteristic (AUROC) curve and the other statistical indicators such as recall, accuracy, precision, and Cohen kappa are employed to predict and compare the success rate of the above-mentioned approaches. The validation results of the models in terms of AUROC values are XGB (0.94) > CB (0.93) > LGBM (0.87) >RF (0.93) >DT (0.88) > LR (0.78). Conclusively, the XGB model outperforms the other models in terms of statistical parameters.

Keywords: Binary Classification, Hyper Parameter tuning, Machine Learning, XGB classifier, Weather Forecasting.

1. Introduction

The majority of industries such as biological, constructional, transportation, and agricultural are impacted by unfavourable weather conditions such as flood, rainfall, drought, etc., making weather forecasting a necessary requirement. One of the most challenging solutions for preventing agricultural and financial losses is weather forecasting. Weather forecasting started in the late nineteenth century and subsequent progress in weather forecast operations is delineated in [1, 2]. In the olden days, meteorologists used to estimate weather parameters based on their expertise, but now the process involves applying technology and data [3]. Conventional data management methods have not been proven efficient or effective for handling big data [4, 5]. As a matter of fact, unpredictable weather patterns throughout the world necessitated many scientists and researchers to develop a new technique of forecasting by using different atmospheric attributes such as humidity, pressure, temperature and wind velocity etc. [6]. Traditionally, forecasting was done by human effort but today, it is dominated by prodigious computational methods that require the use of high-quality equipment [7, 8]. Despite using advanced techniques for data acclimatization by the use of satellite knowledge and supercomputers, prognosticator is still perplexed by the vagaries of the monsoon which makes the smart interpretation and analysis of data difficult. In real-world applications such as medical diagnosis, speech & pattern recognition, natural language processing, and in some renewable energy applications such as solar irradiation, bioenergy and wind speed prediction machine learning (ML) algorithms utilize computational methods to obtain desired information from historical data and extract relevant features to enhance the prediction output [9-10].

Unpredictable change in weather conditions every day has inspired the scientists and researchers to estimate the following day's rainfall having higher societal impact. On a day to day basis, people often use weather forecasts to decide what to wear and what not on a particular day. As the outdoor activities are badly affected by heavy rain, snow and the chilling wind, forecasts can be used to chalk out the different plan activities beforehand during these events. The unpredictability of weather impacts crucial industries such as agriculture, construction, and transportation, necessitating accurate forecasting to prevent significant losses. Traditional methods have proven insufficient for managing large datasets, prompting the development of advanced ML techniques that leverage atmospheric attributes to enhance prediction accuracy and minimize errors [11]. For predicting the weather in an efficient way and to overcome all the above-mentioned problems, a weather

forecasting model using ML techniques is proposed. The main benefit of using these techniques is that the prediction errors are minimized, thereby giving a predicted value very close to the actual value.

Therefore, the major contributions of this research framework are as follows.

1) This paper introduces Extreme Gradient Boost (XGB), a tree-based ensemble approach to assessing weather uncertainty, and provides a comprehensive understanding of the advantages and disadvantages of various ML techniques, including Logistic Regression (LR), Decision Trees (DT), Random Forest (RF), Light GBM (LGBM), and Cat Boost (CB).
2) Conduct an analysis that compares and contrasts multiple ML models, focusing on their performance in uncertainty estimation and weather forecasting.
3) Emphasizes the effectiveness of the proposed XGB ensemble approach in improving weather forecasting accuracy and reliability.

The paper's organization is as follows: Section 1 describes the introductory portion related to weather uncertainty. Section 2 discusses the recent literature studies for the prediction of rainfall. Section 3 provides the introduction to the dataset employed and features engineering steps followed in this work. The basic principles of the models employed and give an overview of the ML algorithm, the methodology adopted for predictive performance computation is presented in Section 4. Section 5 describes the statistical parameters for the evaluation of prediction performance employed in this work. Section 6 consists of the final results and discussions of the hybrid algorithm performance evaluation in conjunction with other reference models. Finally, Section 7 includes the conclusion and future aspects of the current study.

2. Literature Review

It is well documented fact that many publications on rainfall prediction consider models that can perform classification and prediction based on different parameters of atmospheric conditions [12-21]. Traditionally the meteorologists utilize mathematics for simulation and prediction of climate, while the modern techniques such as Artificial Intelligence, ML, Deep Learning, and Reinforced Learning give an easy solution. Formerly, various methods such as Multiple Linear Regression (MLR), Artificial Neural Network (ANN), Support Vector Machine (SVM), Naïve Bayes and k-Nearest Neighbor (KNN) models were generally used for prediction of weather uncertainty. As the ML algorithm can perform variable selection as well as variable extraction from the past data and estimate the future results based on climatic parameters, several research studies based on ML model have been investigated and enumerated as below.

The authors showed the effects of an autoregressive models with exogenous inputs, artificial neural networks and extreme learning machine models for predicting temperature in buildings. They found that the ANN model trained by regularization consistently outperforms the other models [22-23]. For the classification problem, the authors applied different ML algorithms such as KNN and SVM to classify the local weather types over the past years, and the performance of these algorithms is checked by the statistical indicators such as precision, recall and sensitivity etc. while dealing with a large number of predictor variables. They found from the simulation result that the SVM performs well for the small sample scale, and for the large sample scale, KNN achieved higher accuracy [24]. In the next study, the authors examined a Multilayer Perceptron (MLP) brain neural network built and evaluated on a meteorological dataset showing that it can provide more realistic predictions than classical climate models. The authors reviewed an extensive article on the use of data mining methods and compared the results from different ML algorithms for applications of global solar radiation prediction. They found that the relative root mean squared error of the prediction model is reduced by 20% as compared to other models [25]. In the next study, the authors collected the three months' weather data and devised a rainfall forecast model dependent on Multiple Linear Regressor (MLR) method achieving the 52% accuracy [26]. The authors proposed a rainfall prediction model in India and collected weather data from Indian Meteorological Department having 36 attributes in which 7 attributes were found relevant after applying the data preprocessing technique. They used a Bayesian classifier for rainfall prediction and got an accuracy of 81.66% [27]. The authors introduced the ML techniques for the detection of learning styles in e-learning system using classification algorithms such as DT, SVM, KNN, Naive Bayes, Linear Discriminant Analysis, RF, and LR [28]. The authors introduced a cluster-wise linear regression approach which is a combination of clustering and regression techniques for the prediction of monthly rainfall in Victoria. The prediction performance is compared with MLR, ANN and SVM algorithms. They found that the cluster wise regression techniques out-performs as compared to other models [29]. The authors proposed data mining approaches to predict the amount of rainfall based on radar reflectivity and tipping bucket data in a watershed basin at Oxford and Lowa City [30]. In the next study the authors worked on MLP trained with a back propagation algorithm and compared with SVM to predict maximum temperature based on the present temperature data. They found that the prediction performance of SVM performs consistently better than others [31]. The author proposed a back propagation feed-forward neural network trained on the past weather dataset for the prediction of weather uncertainty [32, 33]. In this study the authors introduced a deep learning algorithm using artificial convolutional neural networks to predict weather forecast uncertainty. Despite having lower skill than ensemble models, it is computationally efficient and outperforms other methods

[34]. In this article, the authors presented three ML-based techniques such as modified DT, LGBM, and XGB, for the prediction in different applications. They found that the modified decision tree demonstrates a high potential in terms of accuracy [35]. The authors introduced different ML techniques, out of which the KNN achieved 87% accuracy for the prediction of biogas production [36].

From the aforementioned survey, it is found that most of the traditional and ML approaches used by the authors got a low accuracy due to use of small dataset size and lesser number of training samples and features. To overcome this limitation, larger training set samples and more number of features are used. The main contribution of this paper is to preprocess the several predictors, identify the pertinent features, perform feature engineering steps such as outlier detection, feature scaling, and find missing data etc. for weather forecast. This paper presents XGB, a tree-based ensemble approach for assessing weather uncertainty. It provides a thorough grasp of the advantages and disadvantages of several ML techniques, such as LR, DT, RF, LGBM, and CB, by contrasting this approach with them. The analysis compares and contrasts various models, emphasizing the effectiveness of the suggested XGB ensemble approach for uncertainty estimation and weather forecasting. **Fig. 1** shows the rainfall location of Australia latitude 25^0 South and longitude 135^0 East.

Seaborn having an inbuilt library in Python, is used for plotting the data and displaying the variation. Rainfall density distribution data is plotted on yearly, monthly, and daily basis as shown in Fig. 2 (a), (b) and (c), respectively. A dist, box, and violin plots are used for data visualization and exploratory data analysis as shown in **Fig. 2**. Dist plot shows the data distribution of a target variable against the density distribution, box plot represents a measure of how well distributed the data is in a dataset, and violin plot provides a richer visualization of rainfall data distribution.

3. Methodology

Essentially, this research aims to create an ML-based forecast model that overcomes existing model limitations and any bias inherent to existing models. This study follows numerous steps to create an operational tool for weather forecasting. The steps include comprehensive data acquisition, feature engineering/data transformation, selection, model execution, and evaluation. **Fig. 3** represents the stages of implementing an ML model for weather prediction.

3.1 Data Exploration

This research work aims to address the rain uncertainty task to estimate whether the rain will be happening the next day or not. The dataset is obtained from Kaggle [37] over 10 years by and creating a ML model for accurate estimations. It contains the dataset (rows*columns) comprising of 145460 samples and 23 features. Ten years of data for Australian weather stations over the period from 2007 to 2017 is taken for the estimation of target variable. The data set consists of 23 features, out of which the numerical features are Date (DT), Min Temperature (MINT), Max Temperature (MAXT), Rainfall (RAFL), Evaporation (EVPN), Sunshine (SS), Wind Speed 9am (WS9), Wind speed 3pm (WS3), Humidity 9am (HM9), Humidity3pm (HM3), Pressure9am (PR9), Pressure3pm (PR3), Cloud9am (CLD9), Cloud3pm (CLD3), Temp 9 am (TEMP9), Temp 3 pm (TEMP3), Wind Gust Speed (WGS), and categorical variables are Locations (LOC), Wind Gust Dir (WGD), Wind Dir 9 am (WD9), Wind Dir 3 pm (WD3), Rain Today (RTDY) & Rain Tomorrow (RTMORO) as described in **Table 1**. A dataset is divided into two sets: 75% (109095 records) is used for training, and 25 % (36365 records) is used for testing. The Python software package having an inbuilt library like Pandas, Numpy, Scikitlearn, and Matplotlib is extensively used for data management, mathematical computation, ML modeling, and visualization tools, respectively. This is followed by sequential stages of data visualization, training, testing, modeling, and cross-validation. The data set contains two class labels namely Yes or No for Rain Today & Rain Tomorrow.

3.2 Feature Engineering

For better model compatibility with the dataset, features and samples need to be preprocessed before training. After collection of dataset, following steps are to be taken to enable the variables to mainly include missing data, outlier detection, and feature scaling. ML models are then trained for the binary classification of the target variable rain tomorrow. The dataset used in this work contains float and object values. It can be seen from **Fig. 4(a)** that the float values are more than the object values for the considered dataset. 69.6% of the whole dataset contains a float value and 30.4% contains an object value, as shown by the pie plot in **Fig. 4(b).**

3.2.1 Handling the missing values

Missing values are those that are not observed in a dataset. It can be a single value missing in a single row or misplaced in an entire row value. It can occur in both continuous and categorical variables. The **Fig. 5 (a)** of this work represents the distribution of the target variable (0.0 indicates no possibility of rain and 1.0 indicates the possibility of rain). This shows that the data set is imbalanced, and the random oversampling method has been used for the unbalanced minority target instance.

It can be done by increasing new samples or repeating some previous samples. After applying this method, the minority target is balanced, as shown in **Fig. 5 (b).** The features used in this study contain the null/missing variables as indicated in **Table 2**.

Some of the features like sunshine, evaporation, cloud9am, cloud3pm are indicated in **Table 3,** which contains more null values. So, another approach of data imputation technique is used in which an educated guess is made about its actual value by looking at the other data samples. Missing data imputation is done by replacing all the numerical data with the mean and the categorical data with mode, respectively. After that, encoding of the categorical data is done by using a label encoder in this work. In order to bring data in machine-readable form, a label encoder is used to convert the labels into a numeric number.

3.2.2 Handling Outliers

An outlier is legitimate data that lies outside of the expected range or far away from the mean or median in a given sample. In this study, outliers are detected using the Interquartile Range (IQR) as shown in **Table 4,** and are then removed to get the final working dataset. The steps for finding outliers are as follows:

1. Import the necessary libraries and take the data in ascending order.
2. Calculate Q1, Q2 & Q3 and IQR
   - Q1:25 percentile of the given dataset
   - Q2:50 percentile of the given dataset
   - Q3:75 percentile of the given dataset.

   IQR can be found out using: IQR= Q3-Q1
3. Find the lower and upper limits using Q1-1.5*IQR & Q3+1.5*IQR
4. Finally the data points greater than the upper limits or less than the lower limits are outliers.

3.2.3 Feature Scaling

This study used the min-max normalization technique or a scaling method in which the independent features or variables are shifted or rescaled between 0 and 1. It is also called a min-max scaling, which is described as follows:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \qquad (1)$$

where, X_{max} and $X_{\min}$ are the maximum and the minimum values of the features. If $X = X_{\min}$ then $X' = 0$, if $X = X_{\max}$ then $X' = 1$, and if $X_{min} < X < X_{max}$ then $0 < X < 1$.

3.3 Feature Selection

An attribute that can be predicted by using variables that contain significant information is called a Feature. Feature selection reduces the number of input variables that are vital to predicting the outcome [38]. Transformations are needed for some attributes to be fitted into the model input or to increase the analytical accuracy. A correlation analysis is conducted using the most influential attributes determined by feature engineering. To obtain the correlations between the pair of highly interrelated features, a Pearson Correlation heatmap is drawn. Rain tomorrow as the target value is to be considered. Pearson coefficient is a measure of attributes, and its value near +1 indicates a strong positive correlation between two attributes. Alternatively, correlation values closer to -1 indicate a strong negative correlation. It was observed from the correlation matrix that max temp and min temp, pressure9am and pressure3pm, temp9am and temp3pm, evaporation and max temp, max temp and temp3pm are highly correlated with a correlation coefficient value of 0.73, 0.96, 0.85, 0.75, and 0.98 respectively as shown in **Fig. 6.**

However, it is observed that the correlation coefficient value of the features is not perfectly equal to one, thereby not removing any multicollinearity. However, we can delve deeper into the pairwise correlation between these highly correlated characteristics by examining the following pair diagram as depicted in **Fig. 7**. Each paired plot clearly shows distinct clusters of Rain Tomorrow's "yes" and "no" clusters. There is very minimal overlap between them. The histogram plot diagonally describes the probability distribution of each weather factor. The interrelationship between the predictor variables in the upper and lower triangle of the pair plot represents the scatter plot, and each feature follows a normal distribution. The purpose of this pair plot is to visualize how one feature changes over time in relation to all other features. After that, the embedded method is used to describe the most relevant features. It helps better understand the solved problems and sometimes leads to model improvements by employing the feature selection. In general, embedded methods work more efficiently than wrapper methods since they do not require the user to retrain every feature being observed.

To evaluate the influential features, an embedded method (combined strengths of filtering and wrapping) uses feature performance as an evaluation standard. It integrates a feature selection step into the training process (i.e. both the selection of

features and the training process are performed simultaneously). The extracted features obtained from the embedded method are sunshine, humidity3pm, pressure9am, pressure3pm, cloud9am and cloud3pm.

3.4 Feature importance

A feature importance score is one of the results of executing the algorithm. In this study, feature importance is computed using the Gini index method by determining the mean decrease for each feature. It is evaluated by using (2) below.

$$G = \sum_{c=1}^{c} P_{jc}\left(1 - P_{jc}\right) \qquad (2)$$

where, $\hat{p}_{jc}$ represents the proportion of the samples in the j[th] region that belong to class 'c' for a particular node.

4. ML Models Implementation

Forecasting data interpretation is an essential step in supervised ML. Based on relevant values of independent variables, the method learns how to map input data records to specific dependent output variables. New Prediction algorithms must be used to obtain the most accurate results since they are capable of dealing with complex variables and variables that are interconnected. This paper implements six ML prediction algorithms such as LR, DT, RF, LGBM, CB, and XGB. A recent and efficient ML-based forecast algorithm is XGB, LGBM, and CB. An extensive review of multiple ML forecast algorithms has led to the selection of scalable, flexible, accurate, and relatively fast XGB and LGBM algorithms to achieve in-depth model formalization and proper control of over-fitting. To control the efficiency of machine learning procedures, this phase is crucial to demonstrate the implemented model's response to new data being handled for the first time. A detailed illustration and means of implementation are provided in the next section to illustrate the characteristics of each technique.

4.1 Model Selection

4.1.1 Logistic Regression Classifier

Logistic regression is a method of categorizing data suitable for situations in which the dependent variable *'y'* has to belong to a certain category, or it has two possible values, either zero or one [39]. Thus, the dependent variable has the Bernoulli distribution or two-point distribution. It comes under the special type of linear regression when the dependent variable is in the form of a classification problem. In simple words, the independent variables are not in the form of the binary outcome, so the logit functions are used to fit the data in the form of [0, 1] and this is known as a logistic regression classifier. Log odds refer to the ratio between the likelihood of $y = 1$ and the likelihood of $y = 0$ [40]. LR is an approach to fitting models using logistic functions [41]. In the present study, since the dependent variable was discrete, so the sigmoid function is given as follows:

$$g(y) = \frac{1}{1 + e^{-y}} \qquad (3)$$

where 'y' is the input variable, and $g(y)$ is its outcome. It can be understood simply by finding the α arameters that will give the best fits. The parameter w is a linear combination of multiple independent variables, which is expressed as follows:

$$w = \alpha_0 + \alpha_1 z_1 + \alpha_2 z_2 + \ldots\ldots\ldots \alpha_{22} z_{22} \qquad (4)$$

where, α_0 is the intercept of the regression line and z_i is the coefficients of the independent variables($i = 1 \ldots\ldots\ldots 22$), multiplied by each predictor variable.

4.1.2 Decision Tree Classifier

This algorithm is used for both classification and regression problems. It is a condition-based classifier that works on the '*if/else'* statement. It will execute the *if* statement when the condition is true otherwise, for another condition *'else'* statement is used to fit into a programmable structure. It gives a various output when decisions are to be made. The procedure of the decision tree is as follows:

1. Gather a weather dataset containing several predictor features and a target feature.
2. Splitting the target feature along with the values of predictor features at the time of training the decision tree model.
3. Measure the information gained during the training process.
4. Train the model continuously until the process is completed.
5. After this process, leaf nodes are created, which represent the classifier predictions.

4.1.3 Random Forest Classifier

It is the most powerful non-parametric statistical supervised learning technique, which is mainly used for binary class or multiclass classification datasets. It is a group of many single decision trees for getting better results and accuracy. The

practical advantage of a random forest classifier is that it is able to tune the ML model with minimal parameters with high performance [42]. The steps which are followed in this are as follows:

1. First, select the n number of random samples from the weather dataset.
2. For each random sample, individual decision trees are to be formed.
3. After this step, the outcome will be generated by each decision tree.
4. Finally, the voting will be performed for each decision tree, and the majority of the voting in the individual decision tree is to be considered as a final forecasting result.

4.1.4 Boosting

Boosting is a powerful technique that selects the exact classification or regression from the different numbers of incorrect classification. In this, some of the data is to be extracted from the datasets and given to the base learners, which is created sequentially. The base learners generate the model, which has to be trained. It is an iterative process used to correct the sample errors obtained in the previous base learner models. The iteration process will continue until the correct classification is achieved. At each instance, the base learner model is modified by different sample data. It will continue until the correct classification is achieved. The boosting model will reduce the bias error and create an accurate prediction model. The stepwise procedure of boosting method is as follows.

Step 1: Divide the original dataset into m number of sub-sample.

Step 2: Create a base learners model for training.

Step 3: Build a decision tree (including predictor and categorical features) for each base learner.

Step 4: Generate outcomes from different base learner's models for prediction of each testing data set.

Step 5: Concatenate and develop a final prediction result.

In this paper, three boosting techniques such as LGB, CB, and XGB are proposed for binary classification problems, which are described as follows.

A. Light Gradient Boosted Machine (LGBM) Classifier

Due to its exceptional competence, exactness in the classification of dataset problems, and regression capabilities, as well as its short processing time, the LGBM becomes an ideal solution. LGBM classifier is an open-source framework based on decision trees that provides an efficient and effective implementation. It applies leaf-wise tree growth, is used for ranking, classification, and other tasks. It has been designed as a hybrid technique, combining two novel sampling and classification methods, i.e. gradient-based one-side sampling and exclusive feature bundling [43]. Comparing the analog methods with such combined features, the processes of data scanning, sampling, clustering, and classification are performed over a short period of time with greater accuracy. LGBM becomes an excellent choice when memory requirement, processing time, and arithmetic speeds are taken into account. It accelerates the training process, improves efficiency, optimizes memory, does efficient improved computational utilization (ICU), and enhances accuracy. **Fig. 8** represents the process of LGBM mechanism.

B. Cat Boost (CB) Classifier

This algorithm is also based on gradient boosting and works on the decision trees. It converts categorical values into numbers using statistics on combination of categorical features and a combination of categorical and numerical features [44]. To get the feature importance, it simply takes the difference between the metric obtained using the model in normal scenario. It does not require the use of preprocessing data which can take more amount of time in a typical data science model building process.

C. Extreme Gradient Boost (XGB) Classifier

Recent advances in ML have led to the development of XGB, which has been widely used in several fields. The well-organized, portable, and flexible approach will be suitable for a wide variety of purposes [45, 46]. XGB is a variation of gradient boosting that implements an innovative tree search technique. With such unique capabilities, this classifier can be easily utilized to generate forecasting models when regression and classification methods for the target dataset are incorporated. The XGB library is also used for processing large datasets with a considerable number of attributes and classifications. Also, this algorithm offers efficient and reproducible solutions for the optimization of new problems, especially when balancing efficiency and accuracy. LGBM is similar to XGB, however, it uses level-wise tree growth. Because of this, XGB is considerably slower than LGBM. Due to its level-wise tree growth, XGB is also quite memory-intensive. Even with these drawbacks, XGB is an outstanding and state-of-the-art gradient-boosting algorithm. The XGB library allows it to be implemented in the Python framework. To increase the computational speed, it supports both the central processing unit and the graphics processing unit. It currently doesn't have an embedded feature to handle categorical attributes, so pre-processing the data is necessary. **Fig. 9** represents the process of the XGB mechanism. The Pseudo code of gradient-based boosting techniques is as follows:

1. Input $D = \{(x_1, y_1), (x_2, y_2), \ldots\ldots\ldots\ldots (x_n, y_n)\}, L(y, O(x))$
2. Begin
3. Initialize for the classification problem $F_x(x) = argmin \sum_{i=1}^{n} L(y_i, v)$
4. For $m = 1: M$
5. $r_{jm} = -\left[\frac{\partial L((y_i, F(x_i))}{\partial F(x_i)}\right]$
6. Train weak learner $C_m(x)$ on training data.
7. Calculate w: $w_m = argmin \sum_{i=1}^{n} L(y_i, O_{m-1}(x_i) + wC_m(x_i))$
8. Update: $O_m(x) = O_{m-1}(x) + w_m C_m(x)$
9. End for
10. End
11. Output: $O_m(x)$

5. Evaluation metrics for ML Classification Models

For handling the imbalanced classification dataset, various parameters are used to determine the efficiency and performance. LR classifier is imported from *'sklearn.linear_model'* package, RF and LGBM classifiers are imported from *'sklearn.ensemble'* package, and XGB classifier is imported from *'xgboost'* package. The performance of the binary classification model is measured by evaluation metrics, which indicates the matching of the predicted labels and real labels.

5.1 Accuracy

It is the ratio of number of correct predictions to the total number of input samples. It is evaluated using (5).

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (5)$$

where, $\boldsymbol{TP}$ is true positive, $\boldsymbol{FP}$ is false positive, $\boldsymbol{TN}$ is true negative and $\boldsymbol{FN}$ is false negative.

5.2 AUROC

It is used for dichotomous classification problem. The classifier is able to separate FP rate and TP rate if $AUC = 1$, the classifier is able to separate more number of TP and TN than FP and FN if $0.5 < \text{AUC} < 1$, and the classifier is not able to separate between FP rate and TP rate if $\text{AUC} = 0.5$.

5.3 Precision

It summarizes the ratio of correctly predicted positive results and the total predicted positive results by the classifier. It is defined as follows:

$$\text{Pr}ecision = \frac{TP}{TP + FP} \quad (6)$$

5.4 Recall or Sensitivity

It summarizes the ratio of correctly predicted positive results and the total number of true positive results and false negative results. It is defined as follows:

$$\text{Re}call = \frac{TP}{TP + FN} \quad (7)$$

5.5 F- Measure or F1 Score

It is defined as the harmonic mean between the precision and recall. The range of F1 score varies from 0 to 1, which is defined as follows:

$$F1Score = \frac{2}{\frac{1}{\text{Re}call} + \frac{1}{\text{Pr}ecision}} \quad (8)$$

5.6 Cohen kappa

It is a metric used to assess the agreement between two raters, and is defined as follows.

$$k = \frac{P(A) - P(DA)}{1 + P(DA)} \quad (9)$$

where, P (A) is the probability of agreement, and P (DA) is the probability of disagreement.

5.7 Confusion Matrix

In this matrix, target values are compared with their predicted values based on the machine learning algorithm. With it, it can be seen as to how well the classification model is performing and where are all its errors coming from. There are four main categories: TP, where the actual output is YES; TN, where the actual output is NO; FP, where the actual output is NO; FN, where the actual output is YES.

5.8 Specificity

It summarizes the ratio of correctly predicted negative results divided by the total number of true negative results and false-positive results.

5.9 'k-fold' Cross Validation

This step paves the way for the successful implementation of the internal verification process, where cross-validation is employed. In a random way, each dataset is divided into k folds, with k-1 fold used for training and k folds used for testing [47]. The k-fold approach is used to train the model, in which training and testing folds are divided evenly into k sets of 5, 7 & 10 folds, respectively. The 'k-fold' performance can be calculated using (10) below.

$$'k-fold'\ performance = \frac{1}{k}\sum_{i=1}^{k} Performance \qquad (10)$$

The comprehensive view of the k-fold (i.e. $k = 5, 7, 10$) for the performance of the models are represented in **Fig. 10.**

6. Experimental results and discussion

A comparative analysis of experimental results of different ML classifiers, such as LR, DT, RF, LGBM, CB, and XGB is included in this section. The ML models applied to the dataset observe the competency of the algorithms for 'k-fold' cross-validation. 75% of the total dataset is divided for training purposes, and the remaining 25% is used for the testing dataset. For an Australian weather data set, the k-fold cross-validation and hyperparameter tuning methods for each ML model are used to analyze the results of an experiment.

6.1 'k-fold' Cross Validation results

The k-fold cross-validation (CV) involves dividing the dataset into k-fold, out of which k-1 fold is used for training, and the resulting model is validated on the remaining part of the data. In the analysis of the k-fold cross-validation, the iterative process was performed by changing the value of k from 1 to 10. Randomly selected k-fold values i.e. $k = 5, k = 7,$ and $k = 10$ are used for training and testing datasets for all the models as depicted in **Table 5**. It can be observed that the accuracy of the classifiers such as LR, DT, RF, LGBM, CB and XGB increases as the k-fold values increases. From the above three selected k-fold values, the most accurate prediction is found when the proposed dataset is performed with CV=10 pair for all the models but the best prediction accuracy is achieved from an XGB model for a CV=10, with the highest accuracy of 95.31%, followed sequentially by CB having accuracy of 94.32%, RF having accuracy of 92.73%, LGBM having accuracy of 88.39%, DT having accuracy of 87.24%, LR having accuracy of 79.58%. The proposed algorithm finds accurate results when a tenfold CV is used.

6.2 Hyper parameter tuning results

Before training the ML models, a set of optimal parameters are selected using the grid search method from Scikit learn library for each model. The hyper-parameters optimized by selecting different values and the optimal parameters thus obtained by the considered models are indicated in **Table 6**. The XGB model gives 95.3% accuracy as compared to other ML models, by selecting the following parameters: n_estimators=500, max_depth=16, and learning_rate=1. The tenfold cross-validation method gives better results for the XGB model in terms of accuracy as compared to the hyperparameter tuning method. **Fig. 11 (a)** to (f) show the confusion matrix evaluated using LR, DT, RF, LGB, CB, and XGB models respectively, which summarizes the performance of weather prediction binary classification problem. In the LR classifier TP=0.47, FP=0.09, TN=0.12, FN=0.32, DT classifier TP=0.48, FP=0.08, TN=0.049, FN= 0.39, RF classifier TP=0.51, FP=0.045, TN=0.026, FN=0.41, LGBM classifier TP=0.48, FP=0.082, TN=0.048, FN= 0.39, CB classifier TP= 0.51, FP=0.047, TN= 0.011, FN=0.43 and XGB classifier TP=0.52, FP=0.037, TN=0.01, FN= 0.43. When TP represents the actual positive value the model predicts a positive value, when FP represents the actual negative value the model predicts a positive value, when TN represents the actual negative value the model predicts a negative value, when FN represents the actual positive value the model predicts a negative value. It can be observed from **Fig. 11** that the XGB model gives the good predicted outcomes for the positive class and negative class both.

6.3 Comparative analysis of designed classifiers

For comparing the XGB classifier with other classifier models such as LR, DT, RF, and gradient boosting classifier such that CB, LGBM, some of the statistical indicators are used to evaluate the prediction performance. This study is judged by the five significant parameters such as F1 score, recall, precision, AUC-ROC, and Cohen kappa. The above parameters are calculated on a testing dataset for checking the validity of proposed ML models. **Table 7** represents the classification report of all proposed ML models. For the imbalanced dataset used in this study, more importance is given to F1 score than accuracy to measure the best-performing algorithm.

According to **Table 7,** the F1 score of the ML models are arranged in descending order which are as follows: XGB (0.95218) > CB(0.94104) > RF(0.92824) > LGB(0.86925) > DT (0.85171) >LR (0.78946). It can be seen that for the XGB model, the F1 score is 0.95218, which shows a good result as compared to other models. Precision, recall, and specificity is other important metrics to check the proposed binary classification models performance. These metrics provide better insight into the prediction uncertainty and are more useful for accuracy measurement.

Fig. 12 represents the combined ROC curve of all ML models, which is another significant parameter to evaluate the performance of the classification model. The area under the receiver operating characteristics gives a good result when it is nearer to one. The graphical representation of the ROC curve as depicted in **Fig. 12** indicates that the AUC value of the XGB classifier is least i.e. 0.99, which is very close to unity unlike other models such as LR, DT, RF, LGBM and CB. Therefore, XGB gives a better performance as compared to other implemented ML models.

6.4 Model comparison

6.4.1 In terms of accuracy and time

To compare the models in terms of accuracy as shown in **Fig. 13**, the XGB classifier gives better accuracy, i.e. 95.2% representing a better binary classification. To evaluate the accuracy of gradient boosting algorithms for the proposed weather dataset [26], the experiments are performed on laptop Intel Core i5, 1.1 GHz, 8GB RAM with Windows 10. The execution time taken by XGB model is 254.02 sec which is more in comparison to other implemented ML algorithms. The least execution time is taken by DT which is 0.53 sec out of all the implemented models, as given in Table 7.

6.4.2 In terms of area under the curve and Cohen kappa

The bar chart comparison of proposed implemented ML models in terms of AUC and Cohen kappa score is shown in **Fig. 14**. The experimental result shows that the XGB obtained the highest value of AUC, i.e. 0.945, which gives a good result, and the LR shows the lowest AUC value of 0.78. The Cohen kappa score is said to be in almost perfect agreement if it lies between 0.81 to 1.00, substantial if it lies between 0.61 to 0.80, moderate if it lies between 0.41 to 0.60, fair if it lies between 0.21 to 0.40, none to slight if it lies between 0.01 to 0.20 and values less than and equal to zero indicates no agreement. The experimental results of ML models show that the XGB classifier has a perfect Cohen kappa score, i.e. 0.90 and the lowest for LR, i.e. 0.57, which shows a moderate agreement. The 15 most important features for XGB model is shown in **Fig. 15.** The most influential feature of XGB model is feature code no 17, i.e. cloud9am which is more responsible for the target variable, i.e. rain tomorrow. The feature score of all 15 features is given in **Table 8.** The highest feature score of cloud9am is 0.27189 as compared to other features.

7. Conclusion

Amidst prevailing uncertainty and lack of appropriate correlation among features in Australian weather dataset, rigorous dataset analysis is conducted through pattern observance which helped in extracting and selecting important features for better prediction of binary classification model. In this work, different steps of ML life cycle have been described along with the performance of various ML classifiers such as LR, DT, RF, CB, LGBM and XGB for the prediction of rainfall. A comparative study of all ML classifiers is performed to explain the techniques of fitting data into the ML algorithms for getting exact dependent features.

The experiments are conducted for the Australian weather dataset comprising two imbalanced binary classification categories and subsequently prediction performance of ML models is observed by statistical indicators. For the detection of rain on the following day, XGB algorithm outperforms the others in terms of AUC values. It is observed that data preprocessing steps greatly improve the performance of the models. This study can be harnessed for the various real-life applications such as image recognition, speech recognition, medical diagnosis etc. In future, this method can be used in conjunction with a regression model to improve the reliability of climatic predictions. Moreover, the ML models may be applied for renewable energy applications such as forecasting of solar radiation, wind speed, solar power, and wind energy etc. Most prominently, this may be used for ecological balance, which in turn may save the advanced form of human civilization from extinction.

References

[1] Xiao, Z., Liu, B., Liu, H., et al. “Progress in climate prediction and weather forecast operations in China”, *Advances in Atmospheric Sciences*, **29**(5), pp. 943-957 (2012). https://doi.org/10.1007/s00376-012-1194-9

[2] Jha, A., Goel, V., Kumar, M., et al. “An Efficient and Interpretable Stacked Model for Wind Speed Estimation Based on Ensemble Learning Algorithms”, *Energy Technology*, **12**(6) pp. 2301188 (2024). https://doi.org/10.1002/ente.202301188

[3] Gupta, R., Yadav, A. K., et al. “Prediction of Global Horizontal Irradiance Using an Explainable Data Driven Machine Learning Algorithms”, *Electric Power Components and Systems*, pp. 1-18 (2024). https://doi.org/10.1080/15325008.2024.2310771

[4] Rodríguez-Mazahua, L., Rodríguez-Enríquez, C. A., Sánchez-Cervantes, et al. “A general perspective of Big Data: applications, tools, challenges and trends”, *The Journal of Supercomputing*, **72**(8), pp.3073-3113 (2016). https://doi.org/10.1007/s11227-015-1501-1

[5] Gupta, R., Yadav, A. K., et al. “Harnessing the power of hybrid deep learning algorithm for the estimation of global horizontal irradiance”, *Science of The Total Environment*, **943**, pp.173958, (2024). https://doi.org/10.1016/j.scitotenv.2024.173958

[6] Li, K., & Liu, Y. S. "A rough set based fuzzy neural network algorithm for weather prediction". In *IEEE international conference on machine learning and cybernetics*, pp.1888-1892 (2005). https://doi.org/10.1109/ICMLC.2005.1527253

[7] Hewage, P., Trovati, M., Pereira, E., et al. "Deep learning-based effective fine-grained weather forecasting model", *Pattern Analysis and Applications*, **24**(1), pp. 343-366 (2021). https://doi.org/10.1007/s10044-020-00898-1

[8] Gupta, R., Yadav, A. K., Jha, S. K., et al. "A robust regressor model for estimating solar radiation using an ensemble stacking approach based on machine learning", *International Journal of Green Energy*, **21** (8), pp.1-21 (2023). DOI: 10.1080/15435075.2023.2276152.

[9] Singh, S. K., Jha, S. K., & Gupta, R. "Enhancing the accuracy of wind speed estimation model using an efficient hybrid deep learning algorithm". *Sustainable Energy Technologies and Assessments*, **61**, pp.103603 (2024). https://doi.org/10.1016/j.seta.2023.103603

[10] Pathak, P. K., & Yadav, A. K. "Design of battery charging circuit through intelligent MPPT using SPV system". *Solar Energy*, **178**, pp.79-89 (2019). https://doi.org/10.1016/j.solener.2018.12.018

[11] Jebli, I., Belouadha, F. Z., Kabbaj, et al. A. "Prediction of solar energy guided by pearson correlation using machine learning", *Energy*, 224, pp. 120109 (2021). https://doi.org/10.1016/j.energy.2021.120109

[12] Singh, S. K., Jha, S. K., & Gupta, R. "Comparative Analysis Between Bi-LSTM and Uni-LSTM Algorithms for Wind Speed Estimation", In 2023 *7th International Conference on Computer Applications in Electrical Engineering-Recent Advances* (CERA), pp. 1-6. IEEE, (2023). https://doi.org/10.1109/CERA59325.2023.10455462

[13] Gupta, R., Yadav, A. K., et al. "Predicting global horizontal irradiance of north central region of India via machine learning regressor algorithms", *Engineering Applications of Artificial Intelligence*, **133**, pp. 108426, (2024). https://doi.org/10.1016/j.engappai.2024.108426

[14] Markuna, S., Kumar, P., Ali, R., et al. "Application of innovative machine learning techniques for long-term rainfall prediction". *Pure and Applied Geophysics*, **180** (1), pp. 335-363 (2023). https://doi.org/10.1007/s00024-022-03189-4

[15] Zhang, X., Chen, H., Wen, Y.,et al. "A new rainfall prediction model based on ICEEMDAN-WSD-BiLSTM and ESN". *Environmental Science and Pollution Research*, **30** (18), pp.53381-53396 (2023). https://doi.org/10.1007/s11356-023-25906-9

[16] Rao, J., Wu, T., Garfinkel, C. I., et al. "Impact of the initial stratospheric polar vortex state on East Asian spring rainfall prediction in seasonal forecast models", *Climate Dynamics*, **60** (11-12), pp.4111-4131 (2023). https://doi.org/10.1007/s00382-022-06551-3

[17] Abebe, W. T., & Endalie, D. "Artificial intelligence models for prediction of monthly rainfall without climatic data for meteorological stations in Ethiopia". *Journal of Big Data*, **10** (1), 2. (2023). https://doi.org/10.1186/s40537-022-00683-3

[18] Hussein, E. A., Ghaziasgar, M., Thron, C., et al. "Rainfall Prediction Using Machine Learning Models: Literature Survey". *Artificial Intelligence for Data Science in Theory and Practice*, pp.75-108 (2022). https://doi.org/10.1007/978-3-030-92245-0_4

[19] Diez-Sierra, J., & Del Jesus, M. "Long-term rainfall prediction using atmospheric synoptic patterns in semi-arid climates with statistical and machine learning methods". *Journal of Hydrology*, **586**, pp.124789, (2020). https://doi.org/10.1016/j.jhydrol.2020.124789

[20] Bansal, K., Tripathi, A. K., Pandey, A. C., et al. "RfGanNet: An efficient rainfall prediction method for India and its clustered regions using RfGan and deep convolutional neural networks". *Expert Systems with Applications*, 235, pp. 121191 (2023). https://doi.org/10.1016/j.eswa.2023.121191

[21] Wu, Z., Zhou, Y., Wang, H., et al. "Depth prediction of urban flood under different rainfall return periods based on deep learning and data warehouse", *Science of the Total Environment*, **716**, pp.137077 (2020). https://doi.org/10.1016/j.scitotenv.2020.137077

[22] Potočnik, P., Vidrih, B., Kitanovski, A.,et al. "Neural network, ARX, and extreme learning machine models for the short-term prediction of temperature in buildings". *In Building Simulation, Tsinghua University Press*, **12**, pp.1077-1093 (2019). https://doi.org/10.1007/s12273-019-0548-y

[23] Dueben, P. D., & Bauer, P. "Challenges and design choices for global weather and climate models based on machine learning", *Geoscientific Model Development*, **11**(10), pp.3999-4009. https://doi.org/10.5194/gmd-11-3999-2018, (2018).

[24] Wang, F., Zhen, Z., Wang, B., et al. "Comparative study on KNN and SVM based weather classification models for day ahead short term solar PV power forecasting", *Applied Sciences*, **8**(1), pp.28, (2017). https://doi.org/10.3390/app8010028

[25] Del Campo-Ávila, J., Takilalte, A. et al. "Binding data mining and expert knowledge for one-day-ahead prediction of hourly global solar radiation", *Expert Systems with Applications*, **167**, pp.114147 (2021). https://doi.org/10.3390/app8010028

[26] Kannan, M., Prabhakaran, S., & Ramachandran, P. "Rainfall forecasting using data mining technique", (2010).

[27] Nikam, V. B., & Meshram, B. B. "Modeling rainfall prediction using data mining method: A Bayesian approach", In 2013 *Fifth International Conference on Computational Intelligence, Modelling and Simulation*, pp. 132-136 (2013). https://doi.org/10.1109/CIMSim.2013.29

[28] Rasheed, F., & Wahid, A. "Learning style detection in E-learning systems using machine learning techniques". *Expert Systems with Applications*, **174**, pp. 114774 (2021). https://doi.org/10.1016/j.eswa.2021.114774

[29] Bagirov, A. M., Mahmood, A., & Barton, A. "Prediction of monthly rainfall in Victoria, Australia: Clusterwise linear regression approach". *Atmospheric research*, **188**, pp.20-29 (2017). https://doi.org/10.1016/j.atmosres.2017.01.003

[30] Kusiak, A., Wei, X., Verma, A. P., et al. "Modeling and prediction of rainfall using radar reflectivity data: A data-mining approach". *IEEE Transactions on Geoscience and Remote Sensing*, **51**(4), pp. 2337-2342 (2012). https://doi.org/10.1109/TGRS.2012.2210429

[31] Radhika, Y., & Shashi, M. "Atmospheric temperature prediction using support vector machines". *International journal of computer theory and engineering*, **1**(1), pp.55 (2009).

[32] Esteves, J. T., de Souza Rolim, G., et al. "Rainfall prediction methodology with binary multilayer perceptron neural networks". *Climate Dynamics*, **52**, pp.2319-2331 (2019). https://doi.org/10.1007/s00382-018-4252-x

[33] Cakir, S., Kadioglu, M., & Cubukcu, N. "Multischeme ensemble forecasting of surface temperature using neural network over Turkey", *Theoretical and applied climatology*, **111**, pp.703-711 (2013). https://doi.org/10.1007/s00704-012-0703-1

[34] Scher, S., & Messori, G. "Predicting weather forecast uncertainty with machine learning", *Quarterly Journal of the Royal Meteorological Society*, **144** (717), pp.2830-2841 (2018). https://doi.org/10.1002/qj.3410

[35] Shehadeh, A., Alshboul, O., Al Mamlook, R. E., et al. "Machine learning models for predicting the residual value of heavy construction equipment: An evaluation of modified decision tree, LightGBM, and XGBoost regression", *Automation in Construction*, **129**, pp.103827 (2021).https://doi.org/10.1016/j.autcon.2021.103827

[36] De Clercq, D., Jalota, D., Shang, R., et al. "Machine learning powered software for accurate prediction of biogas production: A case study on industrial-scale Chinese production data". *Journal of cleaner production*, **218**, pp.390-399 (2019). https://doi.org/10.1016/j.jclepro.2019.01.031

[37] https://www.kaggle.com/jsphyg/weather-dataset-rattle-package
[38] Vergara, J. R., & Estévez, P. A. "A review of feature selection methods based on mutual information". *Neural computing and applications*, **24**(1), pp.175-186 (2014). https://doi.org/10.1007/s00521-013-1368-0
[39] Yang, Y., & Loog, M. "A benchmark and comparison of active learning for logistic regression". *Pattern Recognition*, 83, pp.401-415 (2018). https://doi.org/10.1016/j.patcog.2018.06.004
[40] Ekström, M., Esseen, P. A., Westerlund, B.,et al. "Logistic regression for clustered data from environmental monitoring programs". *Ecological Informatics*, **43**, pp.165-173 (2018). https://doi.org/10.1016/j.ecoinf.2017.10.006
[41] Huang, T., Li, B., Shen, D., et al. "Analysis of the grain loss in harvest based on logistic regression". *Procedia Computer Science*, **122**, pp. 698-705 (2017). https://doi.org/10.1016/j.procs.2017.11.426
[42] Genuer, R., Poggi, J. M., Tuleau-Malot, C., et al. "Random forests for big data". *Big Data Research*, **9**, pp.28-46 (2017). https://doi.org/10.1016/j.bdr.2017.07.003
[43] Ke, G., Meng, Q., Finley, T., et al. "Lightgbm: A highly efficient gradient boosting decision tree". Advances in neural information processing systems, **30**, pp.3146-3154 (2017).
[44] Hussain, S., Mustafa, M. W., Jumani, T. A., et al. "A novel feature engineered-CatBoost-based supervised machine learning framework for electricity theft detection", Energy Reports, **7**, pp.4425-4436 (2021). https://doi.org/10.1016/j.egyr.2021.07.008
[45] Tao, H., Awadh, S. M., Salih, S. Q., et al. "Integration of extreme gradient boosting feature selection approach with machine learning models: application of weather relative humidity prediction", *Neural Computing and Applications*, **34**, pp. 515–533(2022). https://doi.org/10.1007/s00521-021-06362-3
[46] Asselman, A., Khaldi, M., & Aammou, S. "Enhancing the prediction of student performance based on the machine learning XGBoost algorithm", *Interactive Learning Environments*, **31** (6), pp.3360-3379 (2023). https://doi.org/10.1080/10494820.2021.1928235.
[47] Fushiki, T. "Estimation of prediction error by using K-fold cross-validation", *Statistics and Computing*, **21**(2), pp.137-146 (2011). https://doi.org/10.1007/s11222-009-9153-8

Figure Captions

Fig. 1 Rainfall location of Australia
Fig. 2 Dist, box and violin plot of (a) Yearly, (b) Monthly, and (c) Daily rainfall data
Fig. 3 Stages of the proposed model implementation
Fig. 4 (a) Bar, and (b) Pie plot of percentage distribution of data
Fig. 5 Data set (a) Before Sampling, and (b) After Sampling
Fig. 6 Pearson correlation heat map between different features in the dataset
Fig. 7 Pair wise plot showing the distribution of each feature based on the other feature
Fig. 8 Process of LGBM mechanism
Fig. 9 Process of XGB mechanism
Fig. 10 (a) 5^{th}, (b) 7^{th} and, (c) 10^{th} fold cross validation
Fig. 11 Confusion Matrix of (a) LR, (b) DT, (c) RF, (d) LGB, (e) CB, and (f) XGB
Fig. 12 ROC-AUC curve of implemented classifiers
Fig. 13 Comparison of All Models in terms of time taken and accuracy
Fig. 14 Comparison in terms of ROC-AUC and Cohen kappa Score
Fig. 15 Top 15 Features of XGB Model

Table Captions

Table 1: Description of dataset
Table 2: Feature contains missing value in percentage
Table 3: Features contains more number of null values
Table 4: Features contain outliers
Table 5: Performance of 'k-fold' CV
Table 6: Optimal parameters selected by ML models
Table 7: Classification report of the proposed ML models
Table 8: Feature score of XGB classifier

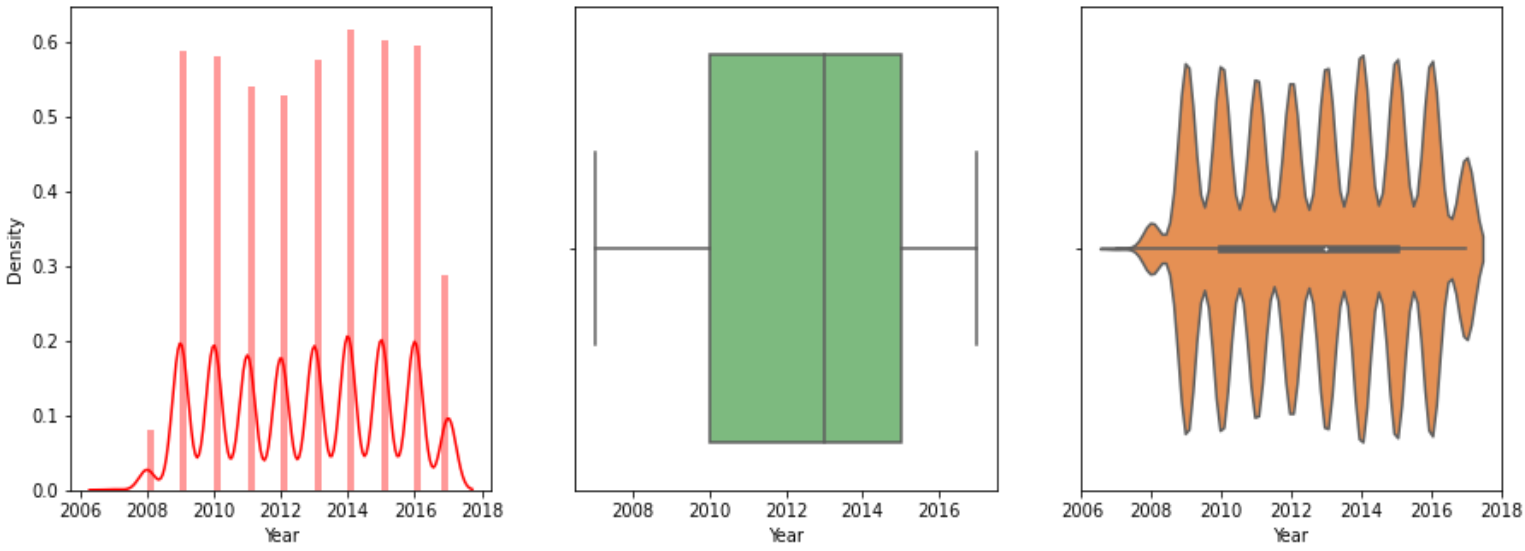
Figures

Fig. 1

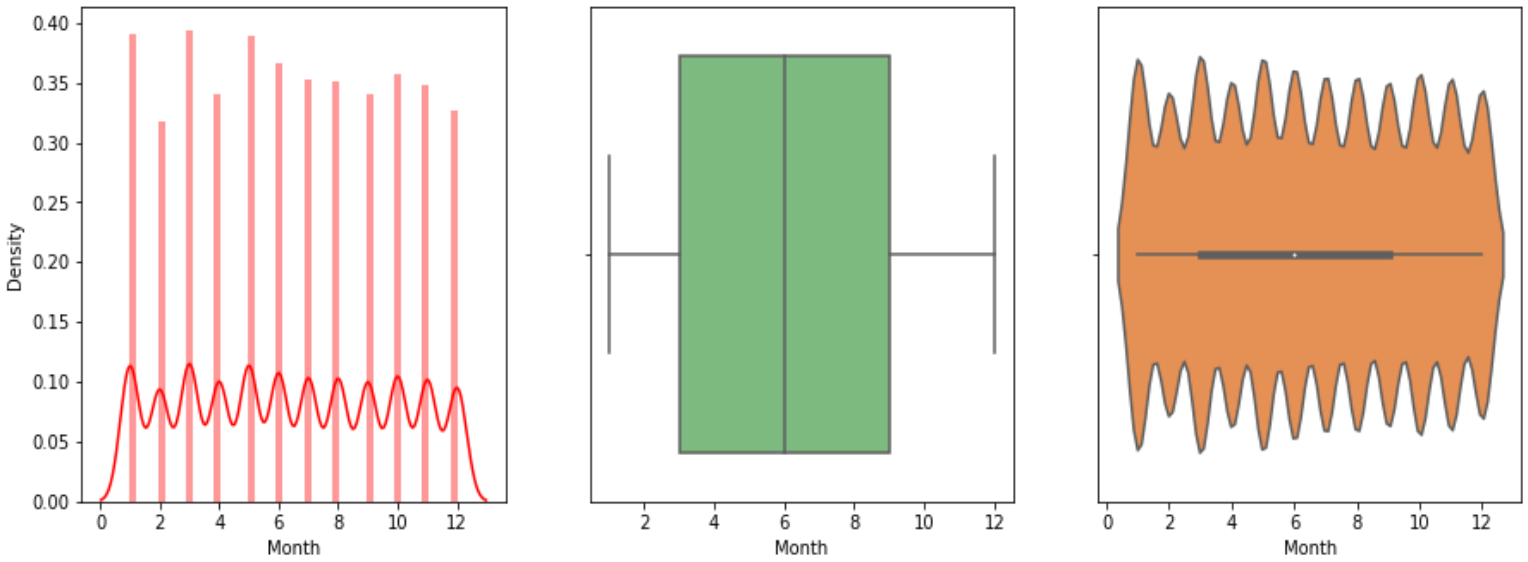

Timor Sea
Solomon Sea
Coral Sea
Australia
Australia
Great
Australian
Bight
Tasman Sea

Fig. 2

(a)


Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2006
2008
2010
2012
2014
2016
2018
Year
2008
2010
2012
2014
2016
Year
2006
2008
2010
2012
2014
2016
2018
Year

(b)


Density
Month
Month
Month

(c)

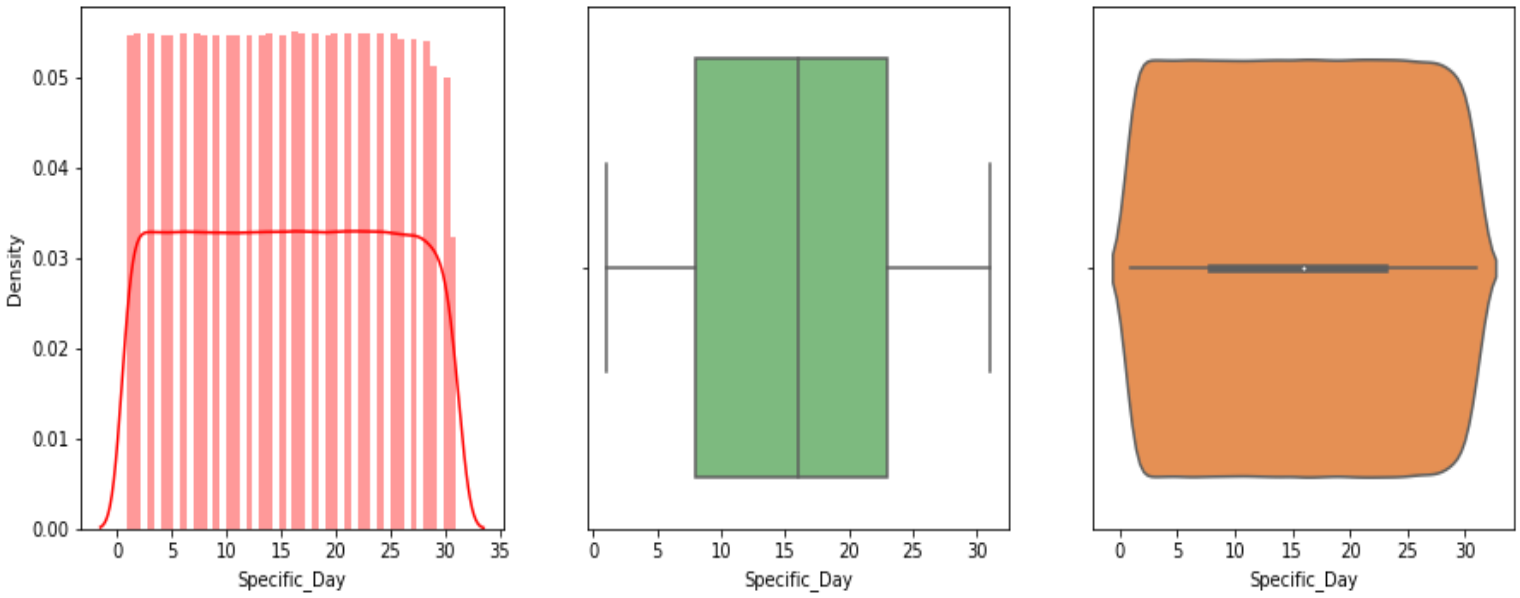

Density
Specific_Day
Specific_Day
Specific_Day

Fig. 3

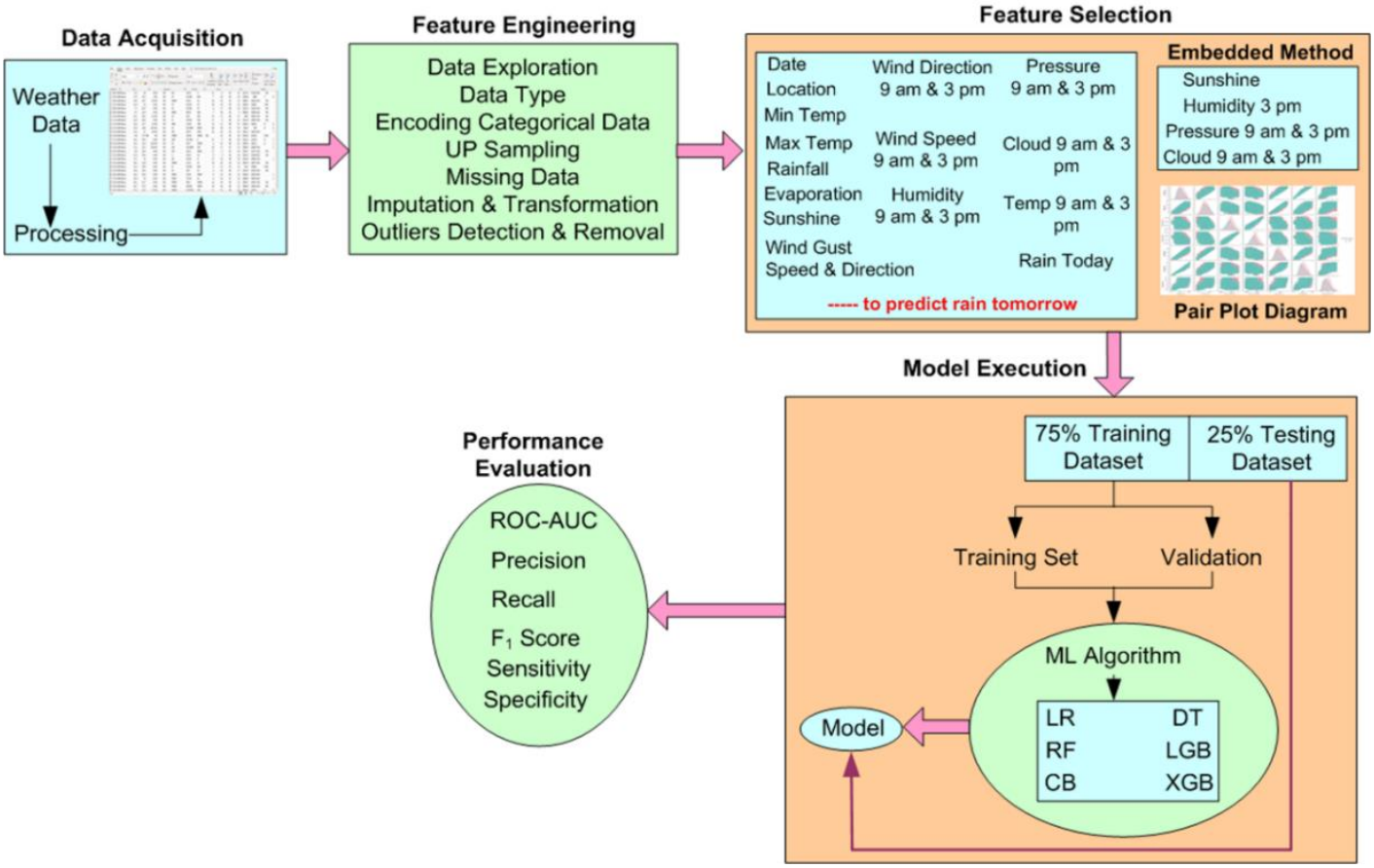
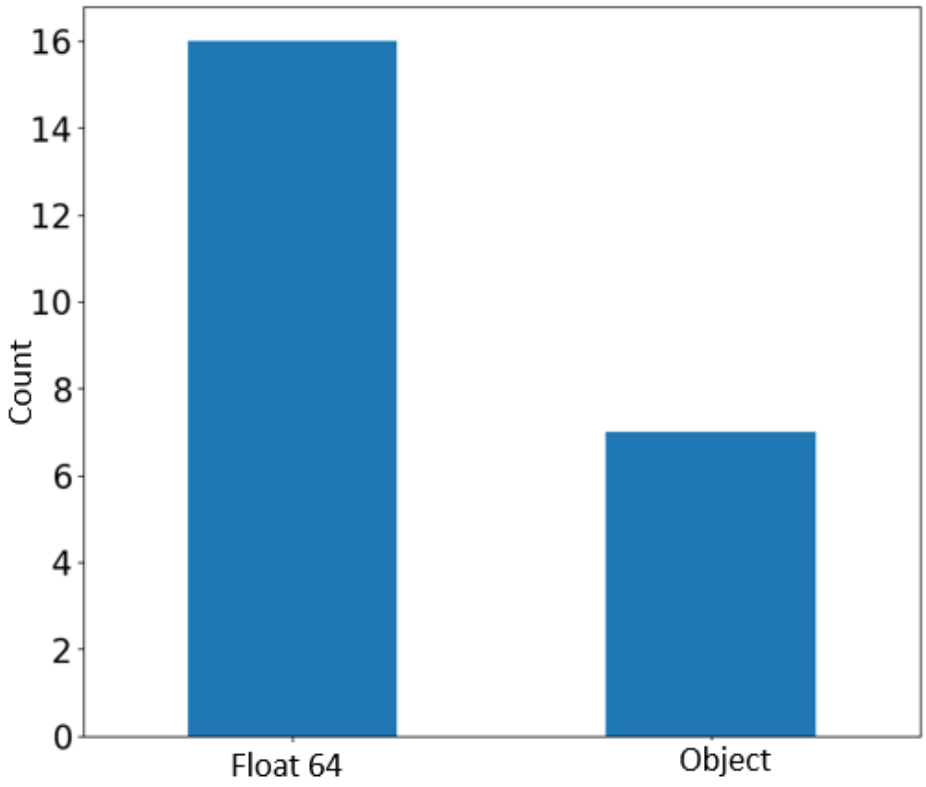

Data Acquisition
Weather Data
Processing
Feature Engineering
Data Exploration
Data Type
Encoding Categorical Data
UP Sampling
Missing Data
Imputation & Transformation
Outliers Detection & Removal
Feature Selection
Date
Location
Min Temp
Max Temp
Rainfall
Evaporation
Sunshine
Wind Gust Speed & Direction
Wind Direction 9 am & 3 pm
Wind Speed 9 am & 3 pm
Humidity 9 am & 3 pm
Pressure 9 am & 3 pm
Cloud 9 am & 3 pm
Temp 9 am & 3 pm
Rain Today
----- to predict rain tomorrow
Embedded Method
Sunshine
Humidity 3 pm
Pressure 9 am & 3 pm
Cloud 9 am & 3 pm
Pair Plot Diagram
Model Execution
75% Training Dataset
25% Testing Dataset
Training Set
Validation
ML Algorithm
LR
DT
RF
LGB
CB
XGB
Model
Performance Evaluation
ROC-AUC
Precision
Recall
F1 Score
Sensitivity
Specificity

Fig. 4

(a)


Count
0
2
4
6
8
10
12
14
16
Float 64
Object

(b)

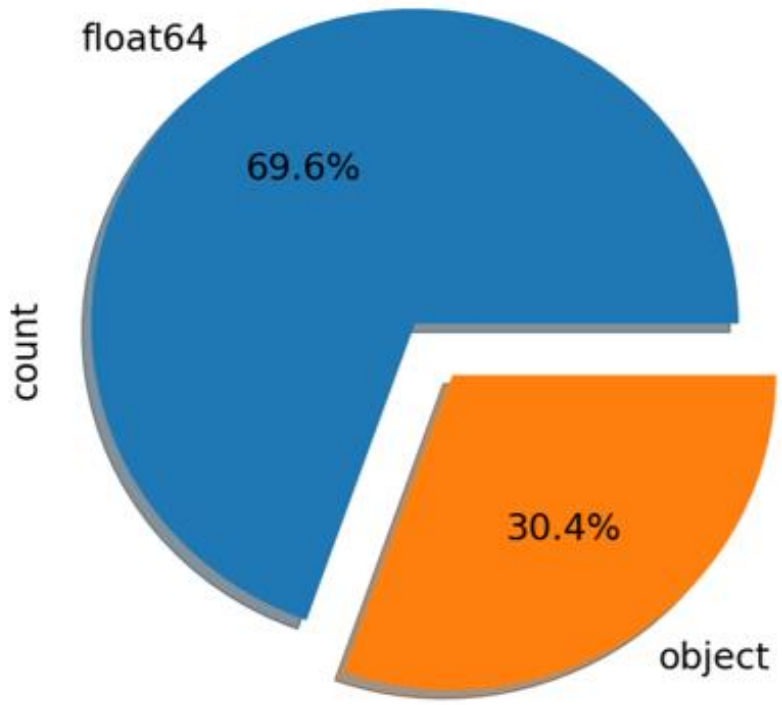
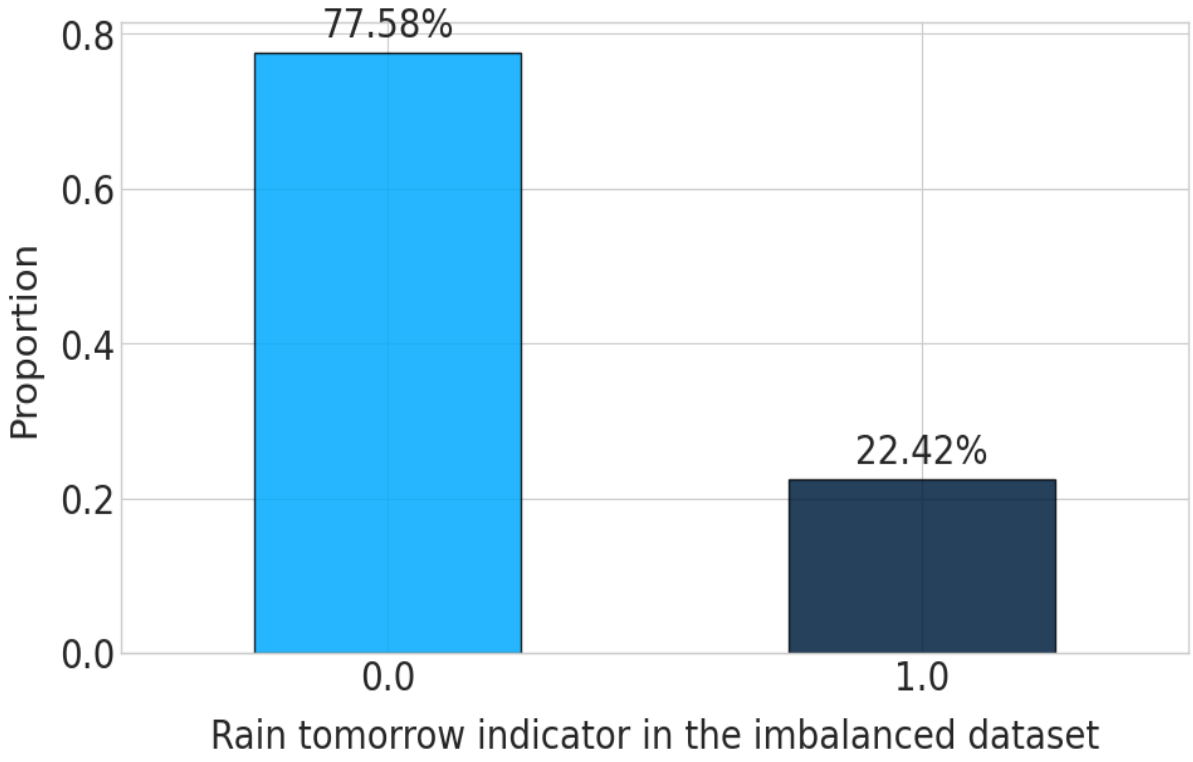

float64
69.6%
count
30.4%
object

Fig. 5

(a)


77.58%
0.8
0.6
0.4
0.2
0.0
Proportion
22.42%
0.0
1.0
Rain tomorrow indicator in the imbalanced dataset

(b)

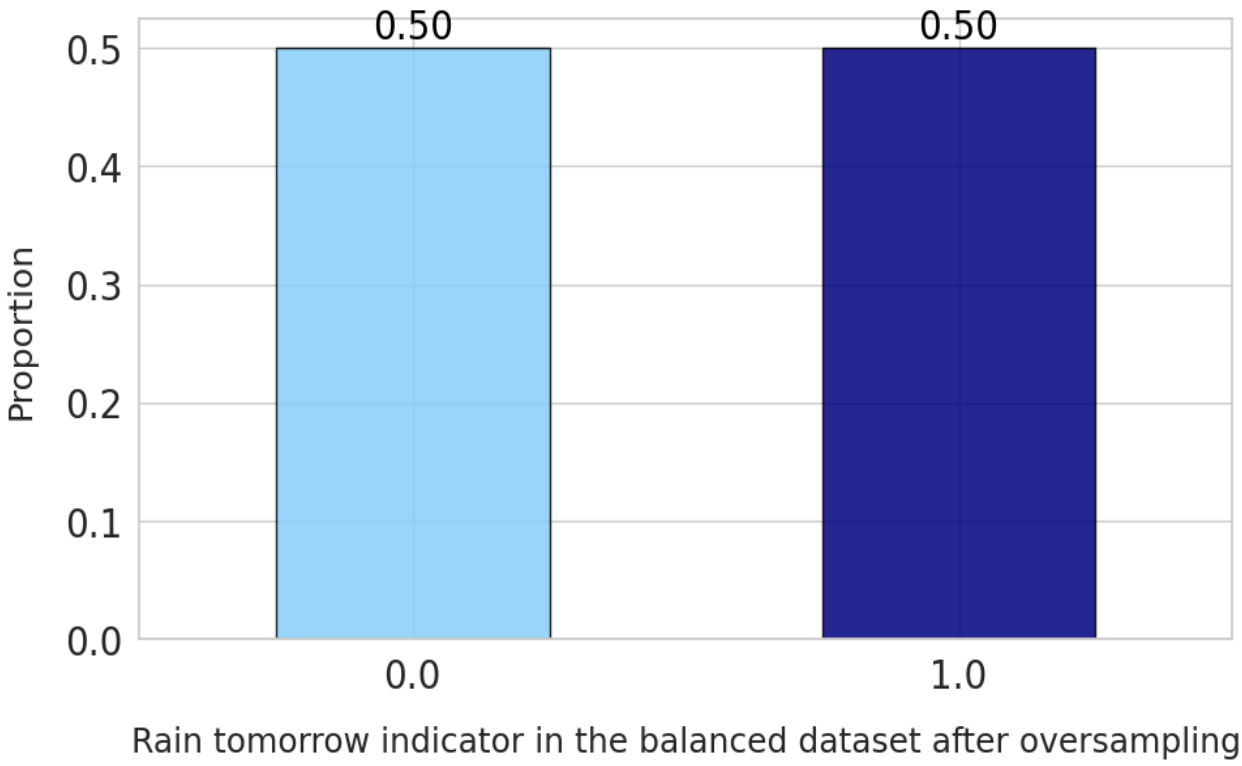

0.50
0.50
0.5
0.4
0.3
0.2
0.1
0.0
Proportion
0.0
1.0
Rain tomorrow indicator in the balanced dataset after oversampling

Fig. 6

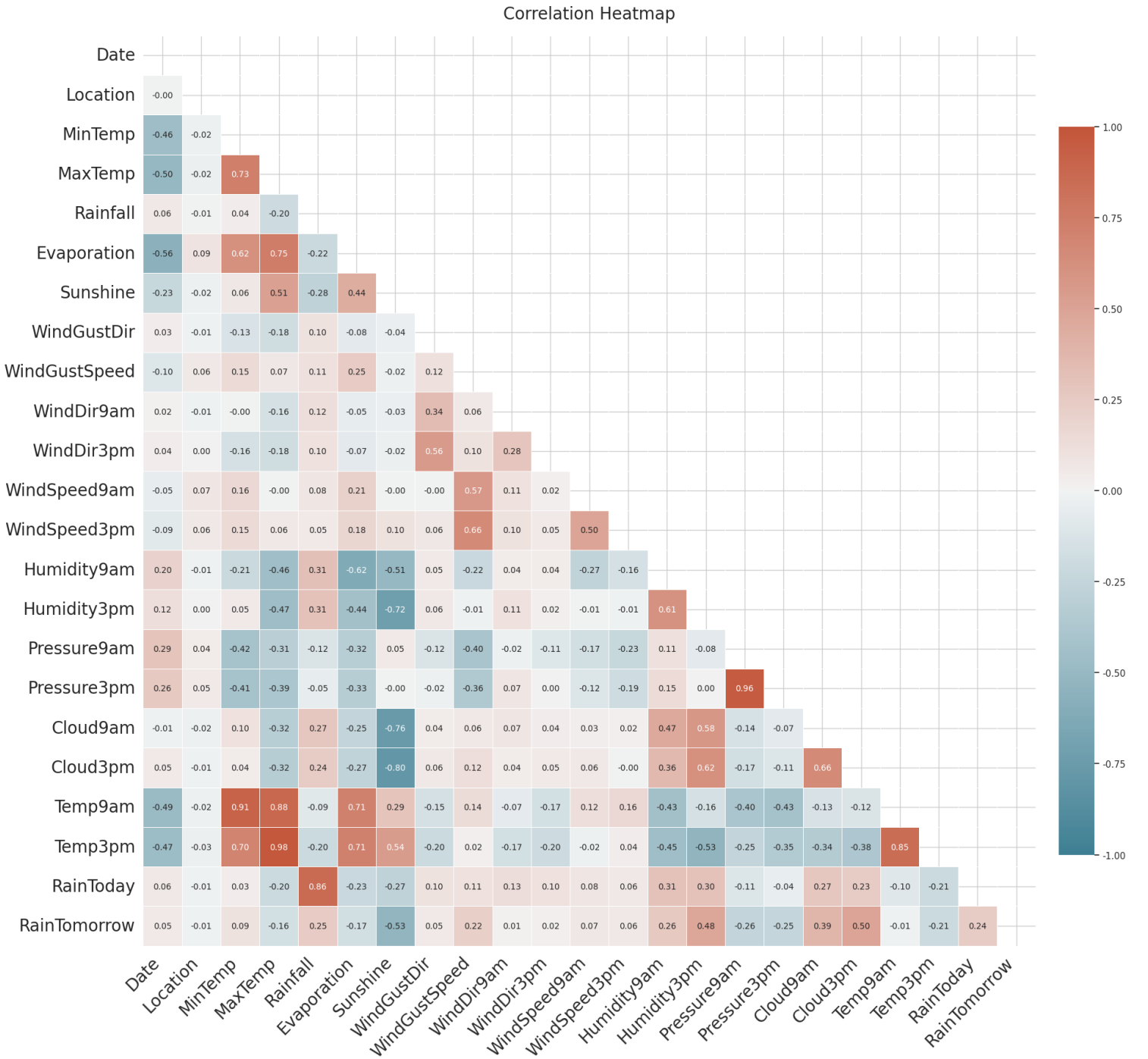

Correlation Heatmap
Date
Location
MinTemp
MaxTemp
Rainfall
Evaporation
Sunshine
WindGustDir
WindGustSpeed
WindDir9am
WindDir3pm
WindSpeed9am
WindSpeed3pm
Humidity9am
Humidity3pm
Pressure9am
Pressure3pm
Cloud9am
Cloud3pm
Temp9am
Temp3pm
RainToday
RainTomorrow
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
-1.00

Fig. 7

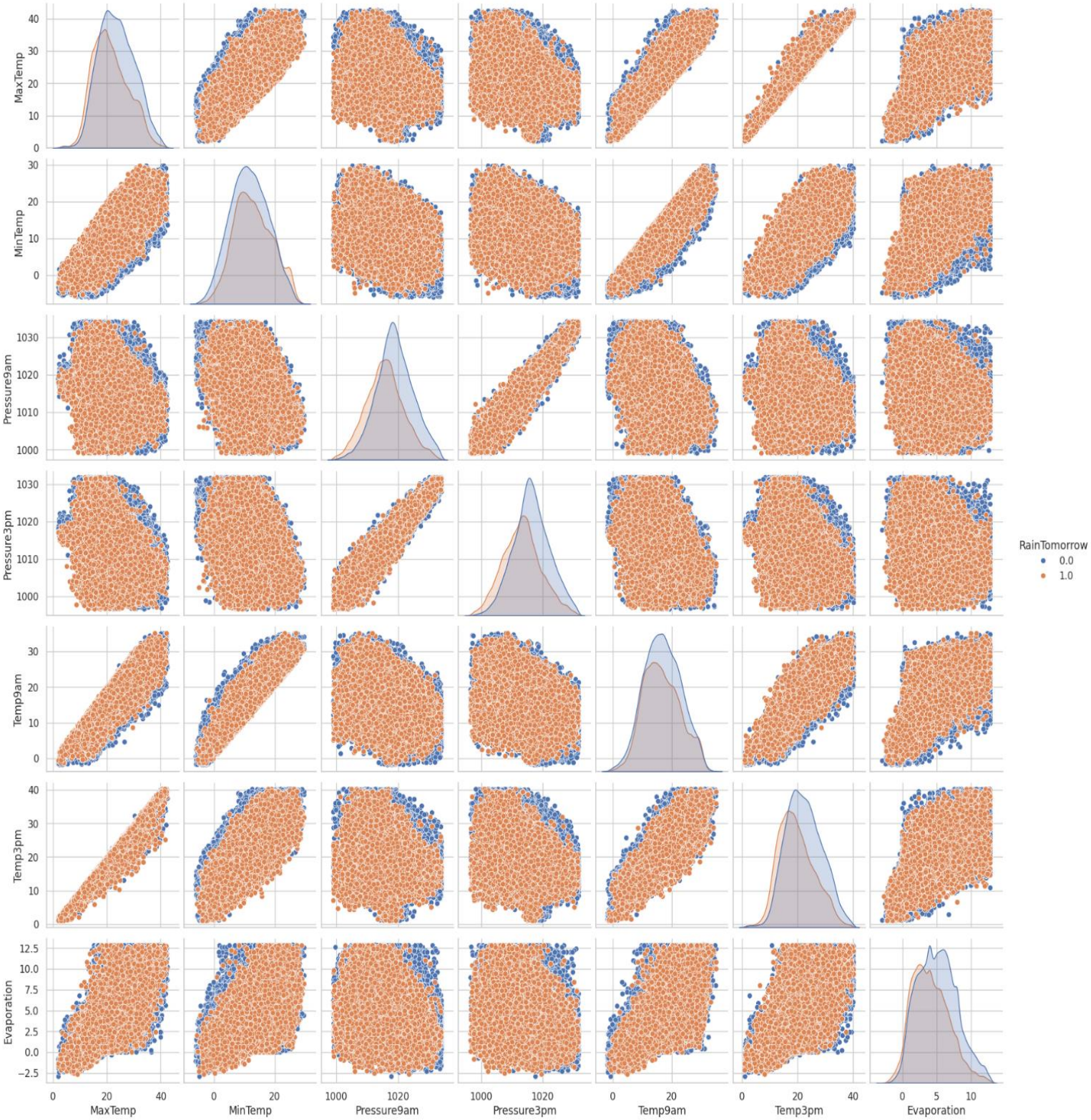

MaxTemp
MinTemp
Pressure9am
Pressure3pm
Temp9am
Temp3pm
Evaporation
RainTomorrow
0.0
1.0

Fig. 8

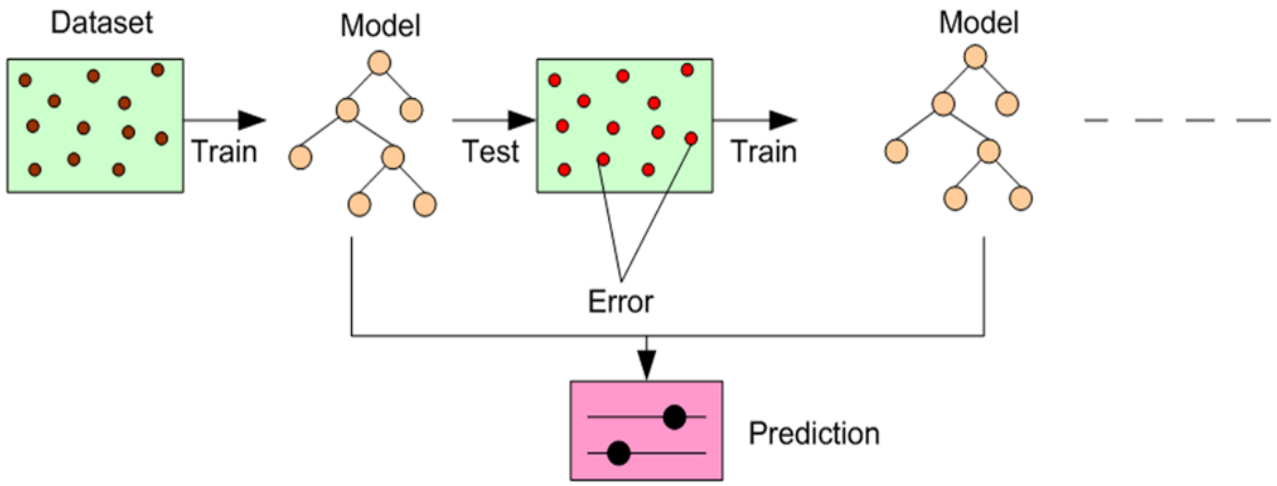


Fig. 9

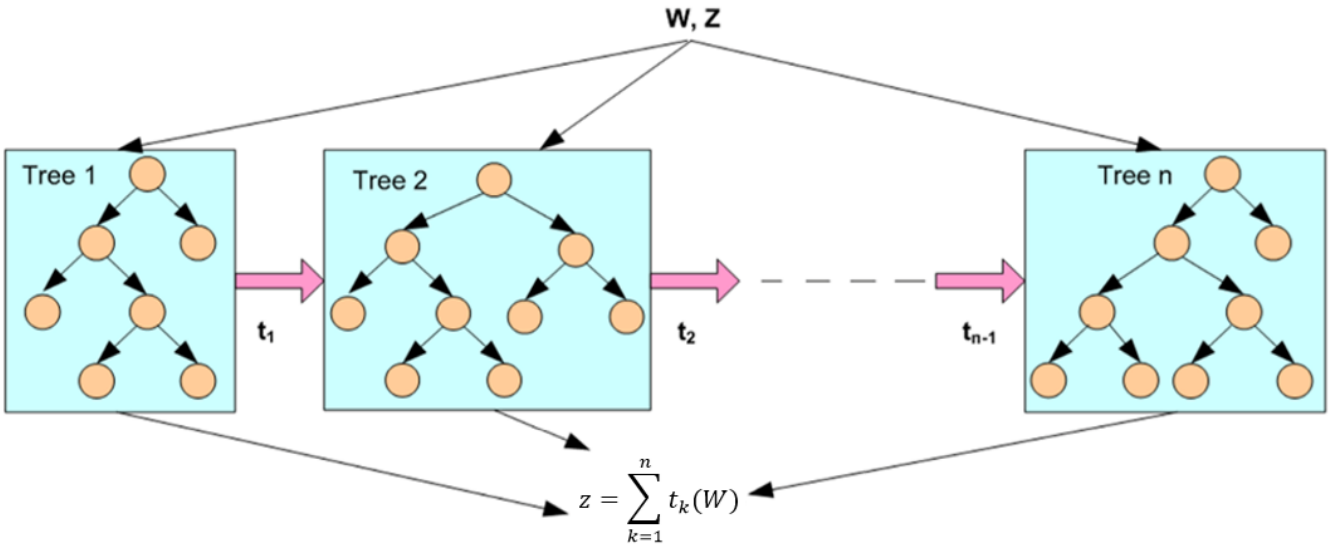
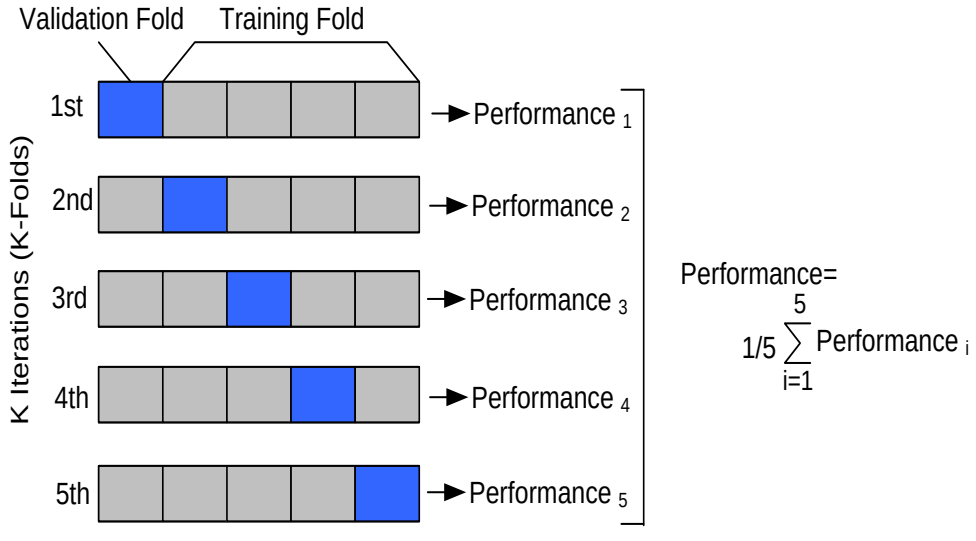


Fig. 10
(a)



(b)


Validation Fold
Training Fold
1st
2nd
3rd
4th
5th
6th
7th
K Iterations (K-Folds)
Performance 1
Performance 2
Performance 3
Performance 4
Performance 5
Performance 6
Performance 7
Performance=
1/7 ∑ Performance i
i=1
7

(c)

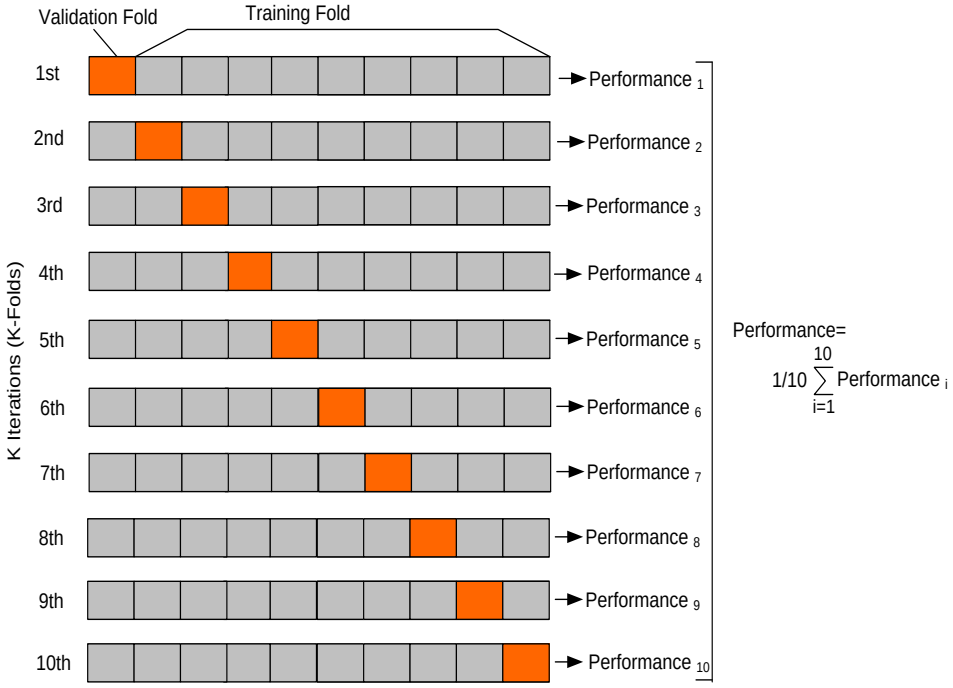
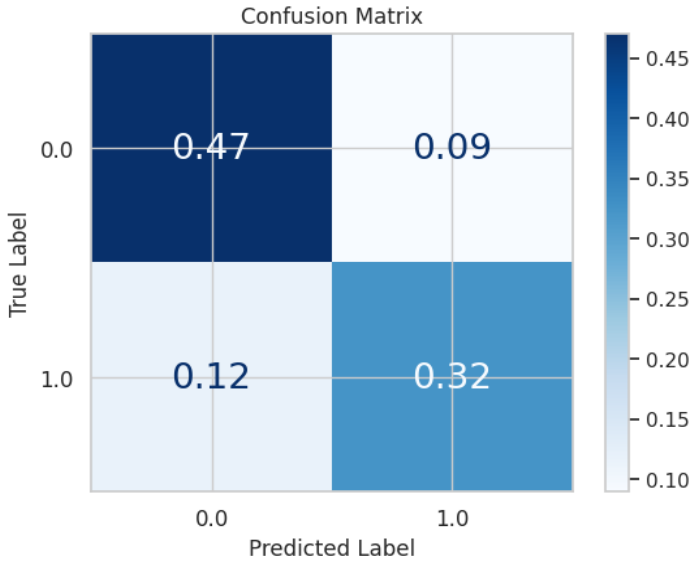
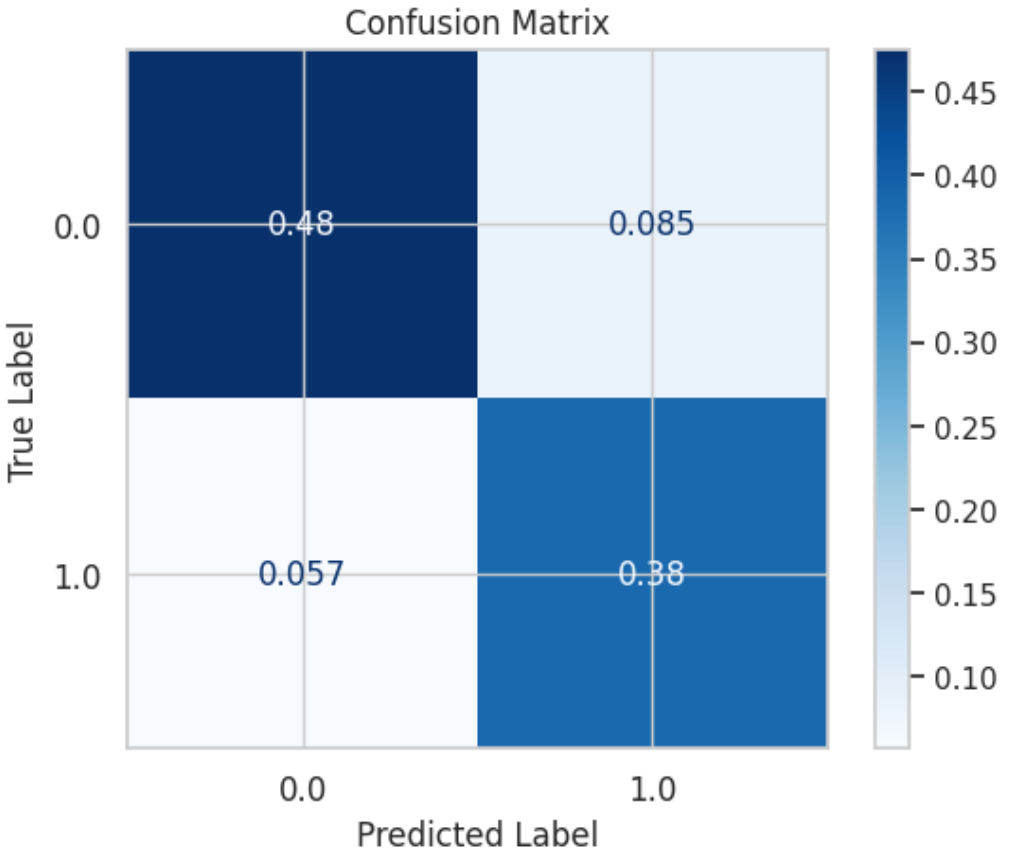

Validation Fold
Training Fold
1st
2nd
3rd
4th
5th
6th
7th
8th
9th
10th
K Iterations (K-Folds)
Performance 1
Performance 2
Performance 3
Performance 4
Performance 5
Performance 6
Performance 7
Performance 8
Performance 9
Performance 10
Performance=
1/10 ∑ Performance i
i=1
10

Fig. 11

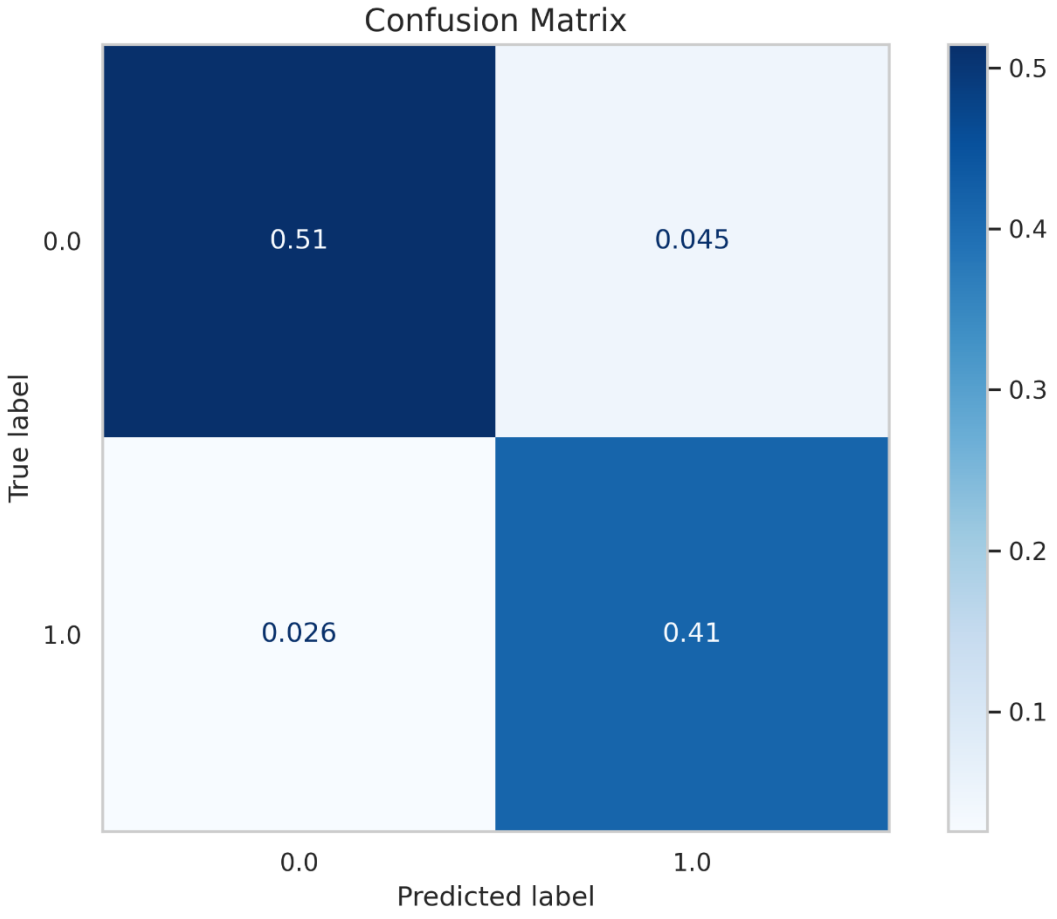
(a)


Confusion Matrix
0.47
0.09
0.12
0.32
0.0
1.0
True Label
0.0
1.0
Predicted Label
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10

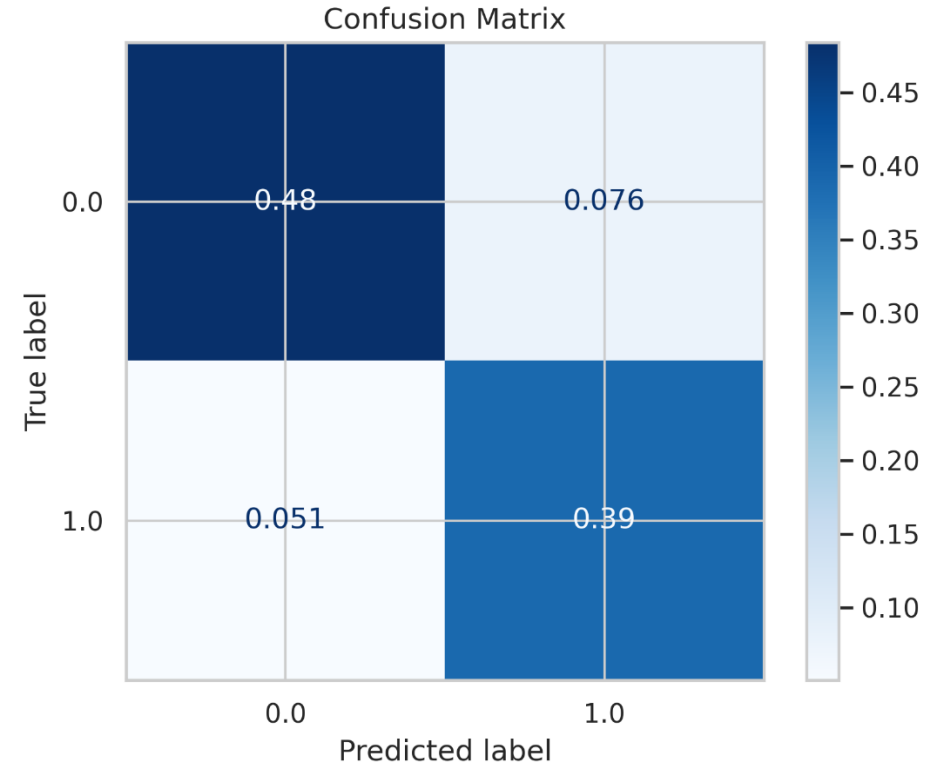
(b)


Confusion Matrix
0.48
0.085
0.057
0.38
0.0
1.0
True Label
0.0
1.0
Predicted Label
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10

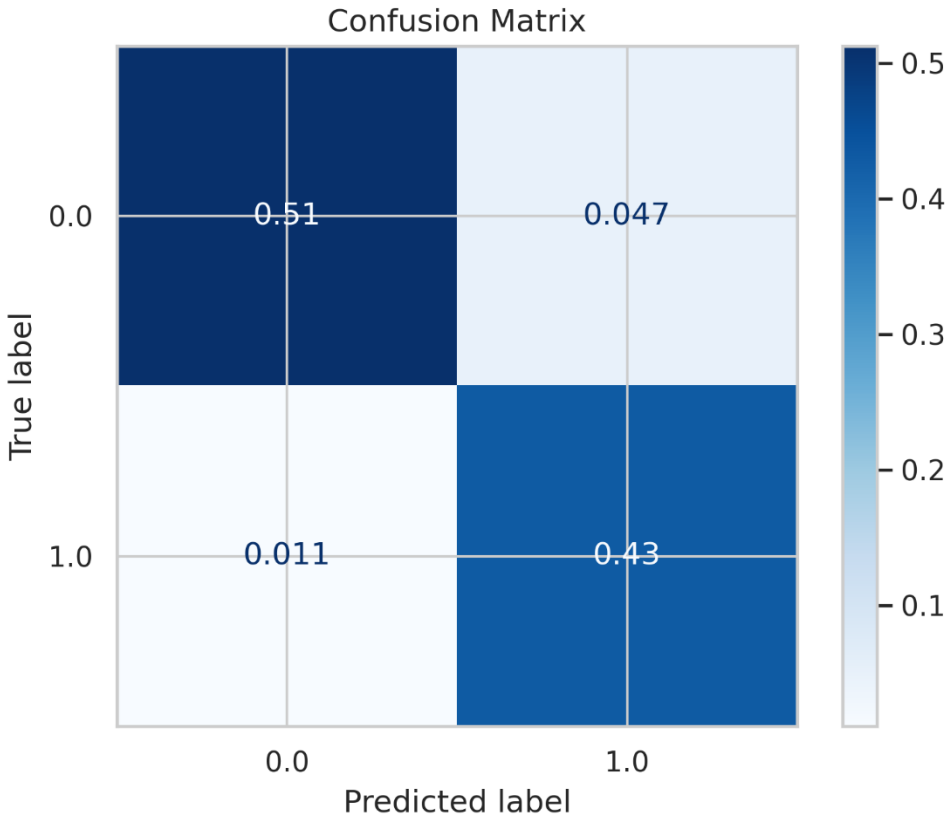
(c)


Confusion Matrix
0.51
0.045
0.026
0.41
0.0
1.0
True label
0.0
1.0
Predicted label
0.5
0.4
0.3
0.2
0.1

(d)


Confusion Matrix
0.48
0.076
0.051
0.39
0.0
1.0
True label
0.0
1.0
Predicted label
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10

(e)


Confusion Matrix
0.51
0.047
0.011
0.43
0.0
1.0
True label
0.0
1.0
Predicted label
0.5
0.4
0.3
0.2
0.1

(f)

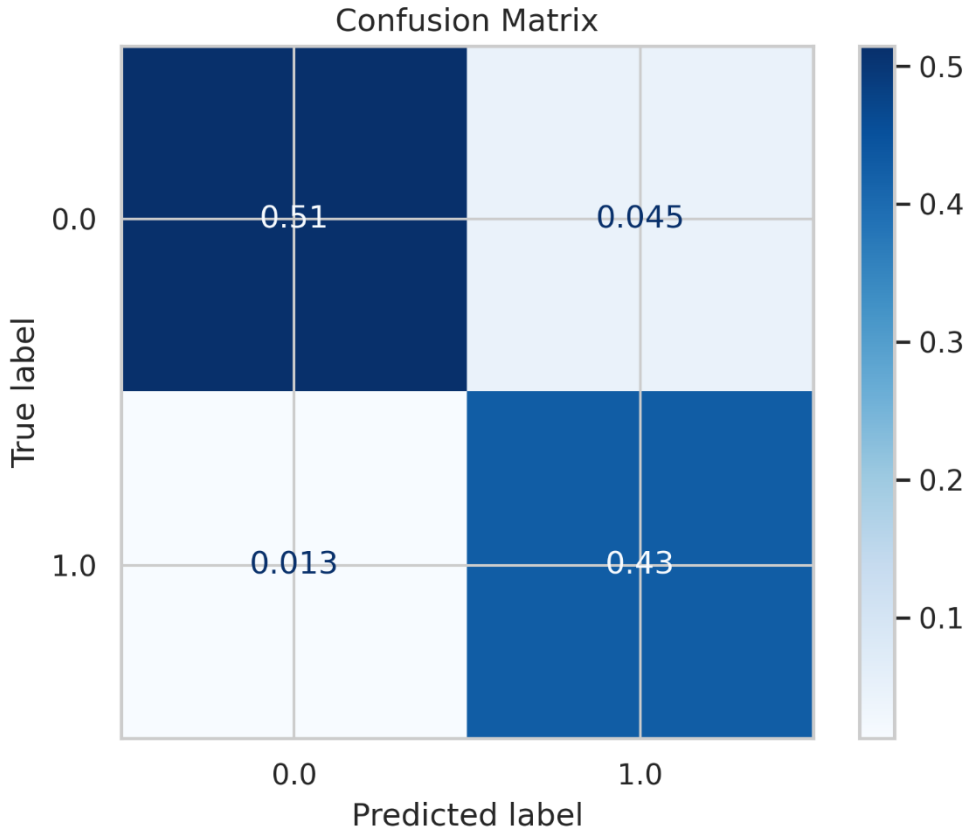


Fig. 12

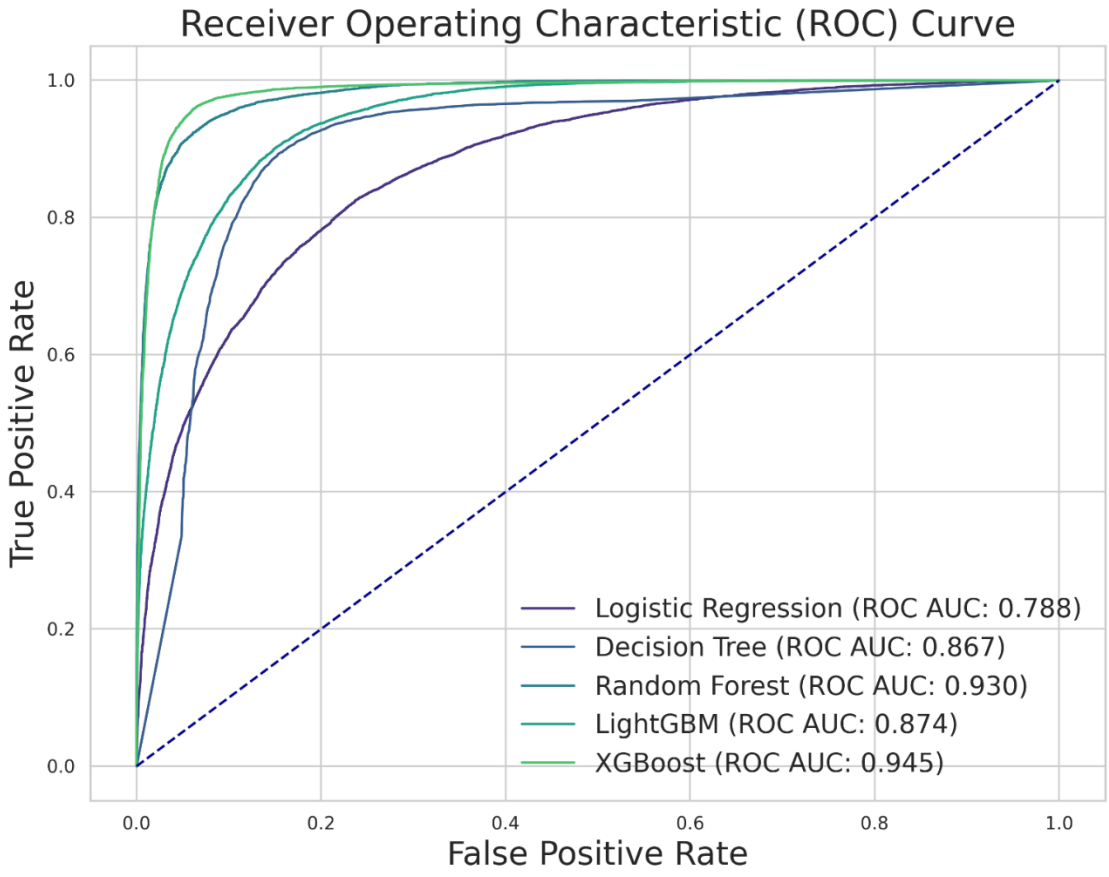


Fig. 13

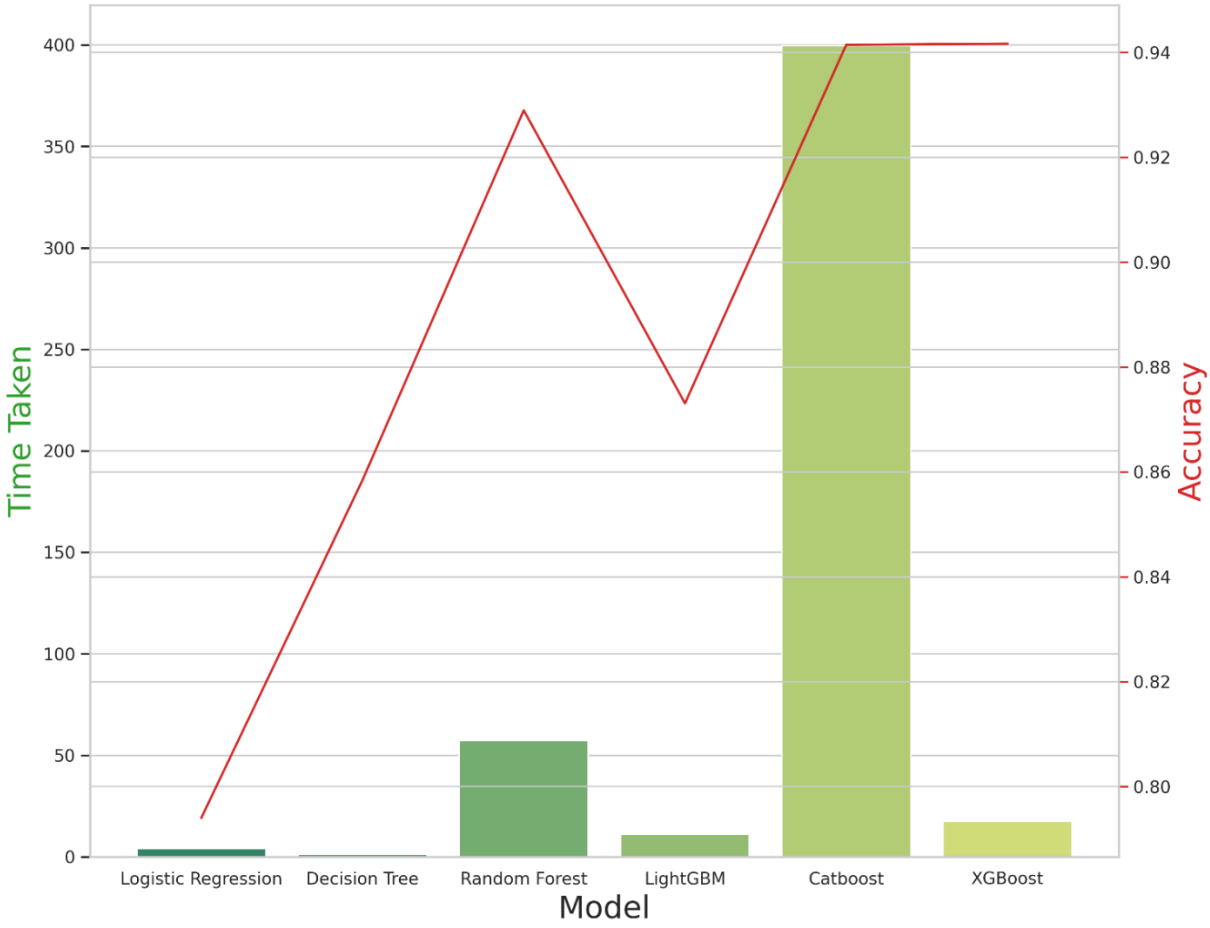

Time Taken
Accuracy
Model
Logistic Regression
Decision Tree
Random Forest
LightGBM
Catboost
XGBoost
0
50
100
150
200
250
300
350
400
0.80
0.82
0.84
0.86
0.88
0.90
0.92
0.94

Fig. 14

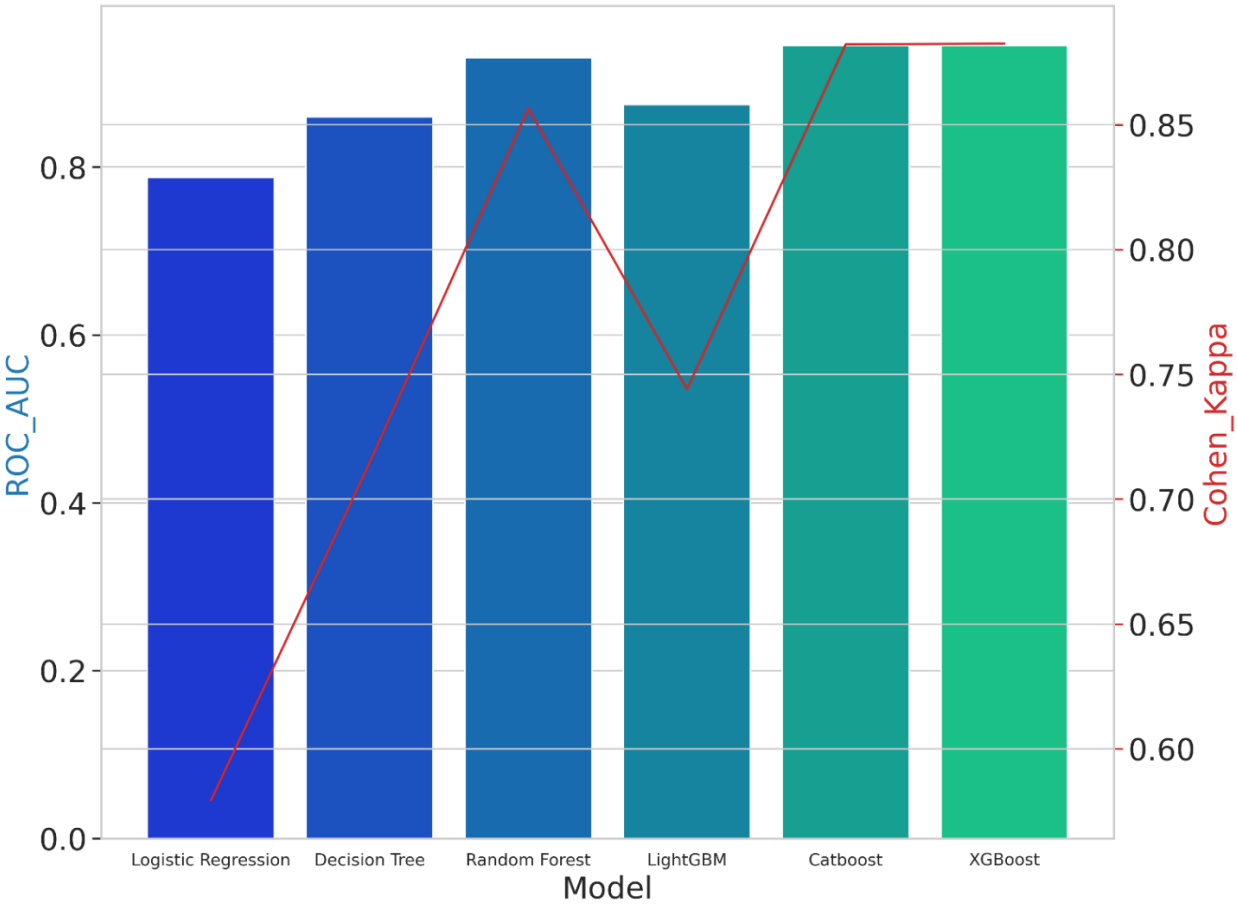
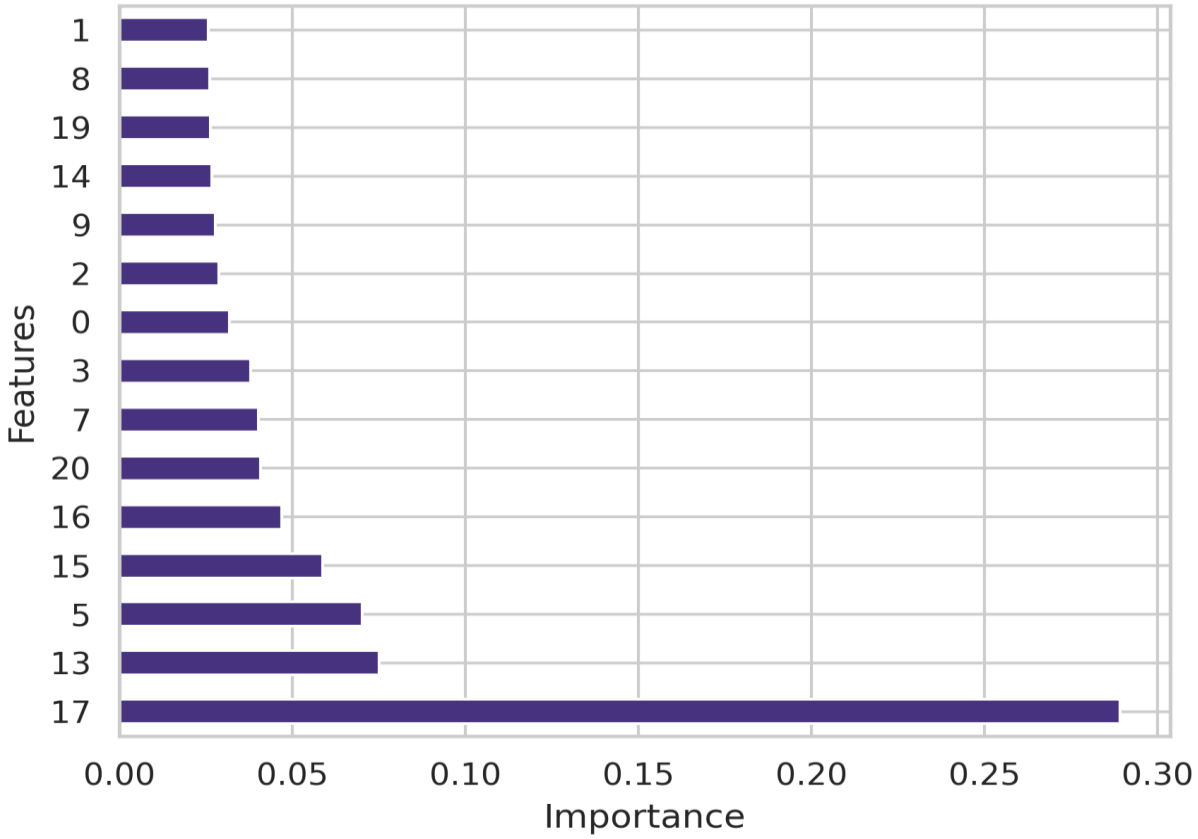

ROC_AUC
Cohen_Kappa
Model
Logistic Regression
Decision Tree
Random Forest
LightGBM
Catboost
XGBoost
0.0
0.2
0.4
0.6
0.8
0.60
0.65
0.70
0.75
0.80
0.85

Fig. 15



Tables

Table 1

Symbol	Feature Type	Description
DT	Predictor	Observation date
LOC	Predictor	Weather station location denotation
MINT	Predictor	Lowest temp (in degree celsius)
MAXT	Predictor	Highest temperature (in degrees celsius)
RAFL	Predictor	Daily recorded Rainfall (in mm)
EVPN	Predictor	Recorded Class A pan evaporation (in mm)
SS	Predictor	Daily record of bright sunshine (in hours)
WSD	Predictor	Recorded strong wind gust direction
WGS	Predictor	Recorded strong wind gust speed (in km/h)
WD9	Predictor	Recorded wind direction at 9am
WD3	Predictor	Recorded wind direction at 9am
WS9	Predictor	Recorded wind speed at 9am (in km/hr)
WS3	Predictor	Recorded wind speed at 3pm (in km/hr)
HM9	Predictor	% Recorded Humidity at 9am
HM3	Predictor	% Recorded Humidity at 3pm
PR9	Predictor	Recorded Atmospheric pressure (in hpa) reduced to mean sea level at 9am
PR3	Predictor	Recorded Atmospheric pressure (hpa) reduced to mean sea level at 3pm
CLD9	Predictor	Recorded Fraction of sky obscured by cloud at 9am
CLD3	Predictor	Recorded Fraction of sky obscured by cloud at 3pm.
TEMP9	Predictor	Observed Temperature (in degrees C) at 9am
TEMP3	Predictor	Observed Temperature (degrees C) at 3pm
RTDY	Predictor	1 if precipitation exceeds 1mm, otherwise 0
RTMORO	Target / Response	The target variable. The rain will happen next day or not.

Table 2

Features	Feature Code	Missing Values	% Missing	Data type
Date	0	0	0.000000	object
Location	1	0	0.000000	object
Min Temp	2	1485	1.020899	Float 64
Max Temp	3	1261	0.866905	Float 64
Rainfall	4	3261	2.241853	Float 64
Evaporation	5	62790	43.166506	Float 64
Sunshine	6	69835	48.009762	Float 64
Wind Gust Dir	7	10326	7.098859	object
Wind Gust Speed	8	10263	7.055548	Float 64
Wind Dir 9am	9	10566	7.263853	object
Wind Dir 3 pm	10	4228	2.906641	object
Wind Speed 9 am	11	1767	1.214767	Float 64
Wind Speed 3 pm	12	3062	2.105046	Float 64
Humidity 9 am	13	2654	1.824557	Float 64
Humidity 3 pm	14	4507	3.098446	Float 64
Pressure 9am	15	15065	10.356799	Float 64
Pressure 3pm	16	15028	10.331363	Float 64
Cloud 9 am	17	55888	38.421559	Float 64
Cloud 3 pm	18	59358	40.807095	Float 64
Temp 9 am	19	1767	1.214767	Float 64
Temp 3 pm	20	3609	2.481094	Float 64
Rain Today	21	3261	2.241853	Float 64
Rain Tomorrow	22	3267	2.245978	Float 64

Table 3

Feature	**Total Value**	**% Null value**
Sunshine	69835	48.009762
Evaporation	62790	43.166506
Cloud 3pm	59358	40.807095
Cloud 9am	55888	38.421559

Table 4

Feature	Outliers	Feature	Outliers
Date	1714	Wind Speed 3 pm	11.0
Location	25	Humidity 9 am	26.0
Min Temp	9.3	Humidity 3 pm	30.0
Max Temp	10.2	Pressure 9am	8.799071
Rainfall	2.4	Pressure 3pm	8.80
Evaporation	4.2	Cloud 9 am	4.0
Sunshine	5.998532	Cloud 3 pm	3.669761
Wind Gust Dir	9.0	Temp 9 am	9.30
Wind Gust Speed	19.0	Temp 3 pm	9.80
Wind Dir 9am	8.0	Rain Today	1.0
Wind Dir 3 pm	8.0	Rain Tomorrow	1.0
Wind Speed 9 am	13.0		

Table 5

Classifier	Cross Validation Score	Accuracy
Logistic Regression	CV=5	0.795755
	CV=7	0.795789
	CV=10	0.795820
Decision Tree	CV=5	0.863186
	CV=7	0.870939
	CV=10	0.872491
Random Forest	CV=5	0.922565
	CV=7	0.925263
	CV=10	0.927354
LGBM	CV=5	0.882818
	CV=7	0.882966
	CV=10	0.883917
Cat Boost	CV=5	0.936986
	CV=7	0.941182
	CV=10	0.943210
XG Boost	CV=5	0.947453
	CV=7	0.950066
	CV=10	0.953186

Table 6

Model Name	Parameter	Optimal Value
LR	penalty	l1
	solver	liblinear
	C	100000000.0
	fit _intercept	True
	max_ iter	50
	random _state	42
DT	max_depth	16
	max_features	sqrt
RF	max_depth	16
	min_samples_leaf	1
	min_samples_split	2
	n_estimators	100
	random_state	12345
LGBM	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
CB	iterations	50
	max_depth	16
XGB	n_estimators	500
	max_depth	16
	learning_rate	1

Table 7

Classifier	Accuracy	Precision	Sensitivity	F1 Score	AUC	Cohen Kappa	Time taken	Specificity
LR	0.7940192339190865	0.79239	0.78772	0.78946	0.88	0.5792306427955354	2.7556116580963135	0.780
DT	0.9289608535929055	0.85082	0.85294	0.85171	0.91	0.7034693379968793	0.53745436668396	0.829
RF	0.9273541439304577	0.92688	0.93035	0.92824	0.98	0.8565310505764735	31.702563047409058	0.901
LGB	0.8702295434869083	0.86809	0.87248	0.86925	0.95	0.7388132209224268	6.24181866645813	0.826
CB	0.9414558813206355	0.93959	0.94505	0.94104	0.98	0.8822386899208776	229.19057393074036	0.901
XGB	0.9525937712052788	0.95058	0.95524	0.95218	0.99	0.9044324140265683	254.02965235710144	0.920

Table 8

Top 15 Features	Feature Code	Feature Score
Date	0	0.02958
Min Temp	2	0.02967
Max Temp	3	0.03912
Evaporation	5	0.06682
Wind Gust Dir	7	0.04342
Wind dir 9am	9	0.02577
Wind Dir 3 pm	10	0.02591
Humidity 9 am	13	0.08171
Humidity 3 pm	14	0.02572
Pressure 9am	15	0.06182
Pressure 3pm	16	0.04794
Cloud 9 am	**17**	**0.27189**
Cloud 3 pm	18	0.02695
Temp 9 am	19	0.02693
Temp 3 pm	20	0.04850

Biographies

Rahul Gupta has 12 years of research and academic experience and pursuing his Ph.D. at Netaji Subhas University of Technology, New Delhi. He is GATE-qualified and serves as a Teaching-cum-Research Fellow (TRF) in the Department of Electrical Engineering at Netaji Subhas University of Technology. He completed his B.Tech. in Electrical and Electronics Engineering from Abdul Kalam Technical University, Lucknow and M.Tech. in Electrical Engineering from Uttarakhand Technical University, Dehradun in 2016. His research interests are in renewable energy, machine learning, and deep learning.

Anil Kumar Yadav received his Ph.D. degree in Instrumentation and Control Engineering from the University of Delhi, Delhi, India in 2017. He is an Assistant Professor in the Department of Instrumentation and Control Engineering, Dr B R Ambedkar National Institute of Technology Jalandhar, Punjab, India. He has 14 years of teaching and research experience and published more than 85 research papers in Journals and Conferences of repute. His research interests include renewable energy, AI Techniques, electric vehicle, and nonlinear and intelligent control.

SK Jha received B. Sc. Engineering degree in Electrical Engineering from Bhagalpur College of Engineering, Bhagalpur with Distinction marks and M.E. degree in Electrical with specialization in Control & Instrumentation from Delhi College of Engineering (DCE) [Presently Delhi Technological University (DTU)], Delhi University. He was awarded Ph. D degree from University of Delhi, New Delhi. Currently he is Professor in the Department of Instrumentation & Control Engineering at Netaji Subhas University of Technology (NSUT) [formerly Netaji Subhas Institute of Technology (NSIT)/Delhi Institute of Technology (DIT)], New Delhi. His teaching and research interests include optimal control, robust control, sustainable energy, bio-inspired control, and electric drives etc. He has published/presented good number of papers in international and national journal, conferences.

Pawan Kumar Pathak received the Ph.D. degree in Electrical Engineering in 2021. He is currently working as an Assistant Professor in the School of Automation at Banasthali Vidyapith (Rajasthan, India). He has more than 8-years of teaching and research experience and published more than 35 research papers in Journals and Conferences of repute. His research interests include renewable energy, load frequency control, battery charger, electric vehicles, cyber-physical power systems, intelligent control, and meta-heuristics.