Enhancing classification performance of binary class imbalanced Data for Weather Forecasting using Machine Learning Classifiers

Rahul Gupta
Netaji Subhas University of Technology
Anil Kumar Yadav (anilei007@gmail.com)
National Institute of Technology Hamirpur
SK Jha
Netaji Subhas University of Technology
Pawan Kumar Pathak
Banasthali University

Research Article

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

Posted Date: April 25th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1546284/v1

Enhancing classification performance of binary class imbalanced Data for Weather Forecasting using Machine Learning Classifiers

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

[1]Department of Electrical Engineering, Netaji Subhas University of Technology, Dwarka, New Delhi, 110078, India.
E-mail: rahul.ee20@nsut.ac.in
[2]Department of Electrical Engineering, National Institute of Technology Hamirpur, Hamirpur (H.P.) 177005, India.
E-mail: anilkyadav@nith.ac.in, anilei007@gmail.com (*Corresponding Author)
[3]Department of ICE, Netaji Subhas University of Technology, Dwarka, New Delhi, 110078, India. E-mail: skjha@nsut.ac.in
[4]School of Automation, Banasthali Vidyapith, Rajasthan, 304022, India. E-mails: ppathak999@gmail.com

Abstract: Drastic change in climatic conditions is a very big and challenging task for people around the globe. Most of the biological, constructional, transportation and agricultural sectors get affected due to uneven weather conditions, i.e. flood, rainfall, drought, etc. As part of the weather system, rainfall being most prominent phenomena, its rate is treated as one of the most important variables. Meteorological scientists try to identify the parameters of the atmosphere such as temperature, sunshine, cloudiness and humidity of the earth by applying conventional techniques and developing a prediction model. These days, Machine Learning (ML) techniques are more evolving and give more accurate results than the traditional approaches. ML is a subset of artificial intelligence (AI) which is used in this paper for predicting the next day's rainfall from the past 10 year's weather dataset of Australia. 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 of the next day. 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 evaluation metrics like Area under the Receiver Operating Characteristic (AUROC) curve, recall, accuracy, precision, and Cohen kappa are used to check the performance of ML algorithms.

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

1. Introduction

The prediction of weather is an indispensable requirement because it plays an essential role in agriculture, transportation, industrial, commercial, and tourism sectors etc. Flood, precipitation, evapotranspiration, thunderstorm, hurricane, typhoons, lightning, and fog are the most salient atmospheric events for the sustainable weather forecast system. Changing climatic conditions each day has motivated the researchers and scientists to predict the next day's rainfall, which has a major impact on society. One of the most challenging solutions for preventing agricultural and financial losses is weather forecasting. Weather forecasting started in the late nineteenth century and progress in climate prediction and weather forecast operations is delineated in [1, 2]. In 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 rapidly changing behavior of weather patterns worldwide, many researchers and scientists are trying to find a different way to forecast by using features such as temperature, humidity, pressure, wind speed, wind direction etc. [6]. Traditionally, forecasting has been a human effort, but today, it is dominated by computational efforts 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 monsoon, unable to predict smart interpretation and analysis of data in Numerical Weather Prediction. In real-world applications such as medical diagnosis, speech & pattern recognition, natural language processing, and in some renewable energy applications like solar irradiation, bioenergy, wind speed prediction, etc. ML algorithms use computational techniques to learn information from past data and extract useful data to improve performance [9, 10, 11]. This study aims to address the rain uncertainty problem to predict whether the rain will be happening the next day or not based on Australian rainfall dataset obtained from Kaggle [26] over 10 years by using ML techniques by creating a model for accurate predictions. This work explores and compares the performance of different ML methods, including LR, DT, RF, LGBM, CB, and XGB, to predict Australian weather uncertainty**. 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. A dist, box, and violin plots are

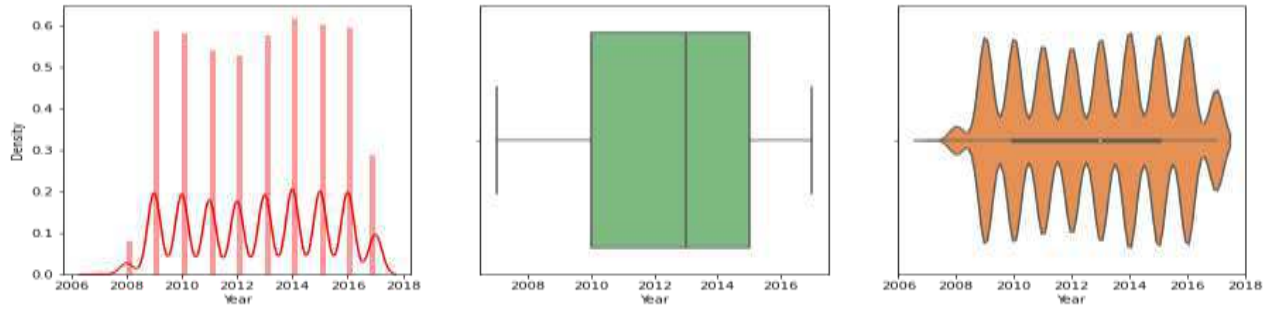
used for data visualization and exploratory data analysis as shown in **Fig. 2**.

Fig. 1 Rainfall location of Australia

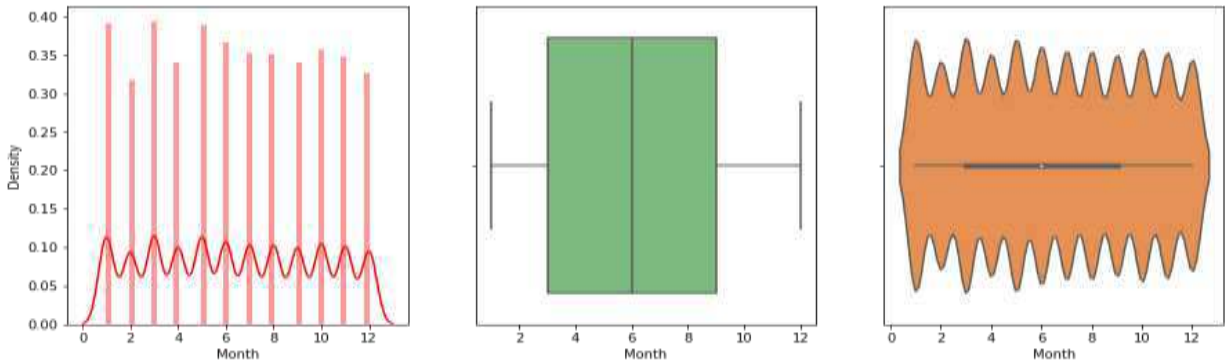
Dist plot shows the data distribution of a variable against the density distribution, box plot represents a measure of how well distributed the data is in a dataset, and violin plot shows the probability density of data. Rainfall density distribution data is plotted on yearly, monthly, and daily basis as shown in Fig. 2 (a), (b) and (c), respectively. **Fig. 3** represents the stages of implementing a machine learning model for weather prediction.

2. Literature Review

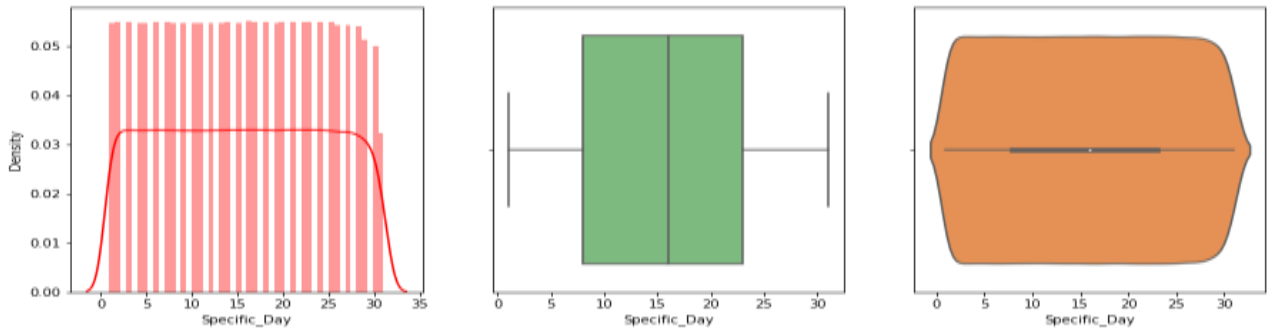
It is well documented that many publications on rainfall prediction consider models that can perform classification and prediction based on different parameters of atmospheric conditions. Furthermore, the meteorologist currently utilizes mathematics to simulate for predicting climate, while the new techniques give an easy solution that includes artificial intelligence, machine learning, deep learning, and reinforced learning-based techniques. In previous years, numerous methods were often used to predict the weather uncertainty such as Multiple Linear Regression, Naïve Bayes, Support Vector Machine, artificial neural network, and K-nearest neighbor models.



(a)



(b)



(c)

Fig. 2 Dist, box and violin plot of (a) Yearly, (b) Monthly, and (c) Daily rainfall data

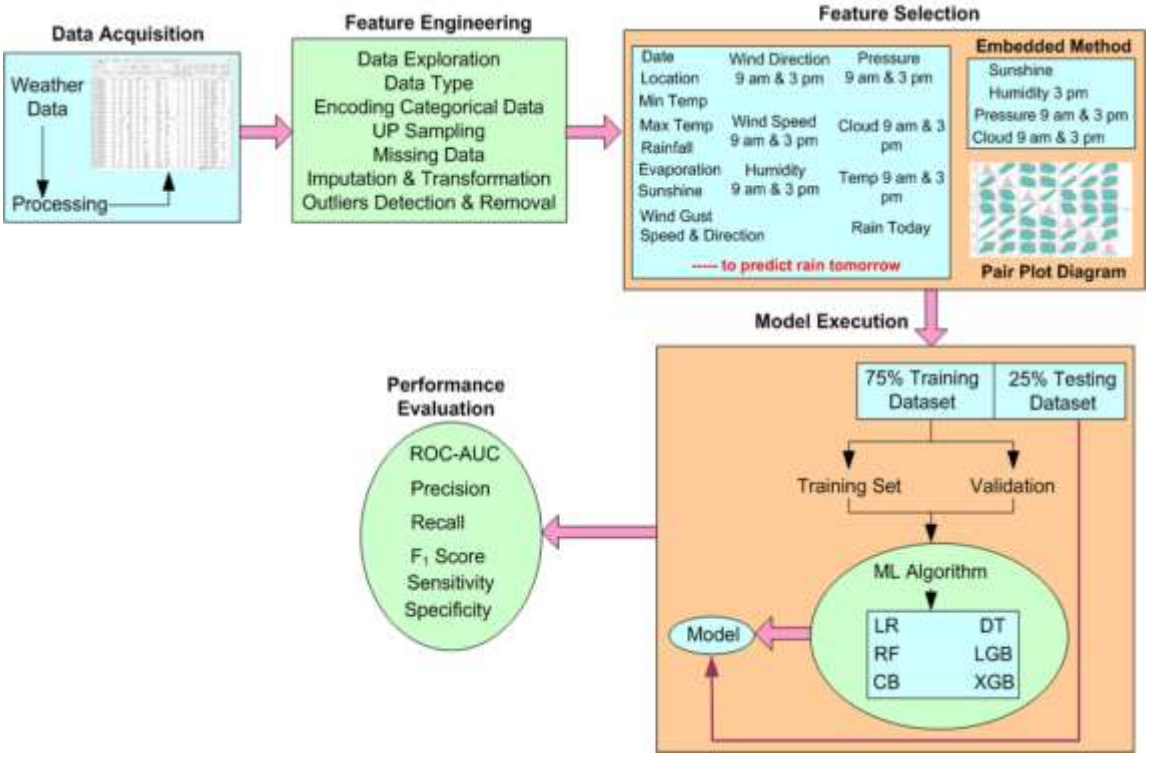


Fig. 3 Stages of ML Model implementation

Since the ML model can do feature extraction as well as feature selection from the previous data and forecast the future outcomes depending on the weather parameters. Some of the published papers based on ML have been examined and summarized as follows. Potonick et al. [12] showed the effects of an autoregressive artificial neural network with exogenous input and extreme learning machine models for predicting temperature in buildings. They used the python software package for developing the model using Tensor Flow. The variety of ML methods for classification tasks has been studied over the years related to the efficiency, accuracy, and performance of the algorithms while dealing with a huge number of predictor features [14]. Dueben et al. [13] examined a multilayer perception (MLP) brain neural network built and evaluated on meteorological data, thus showing that it can provide more realistic predictions than classical climate models. Campo-Avilla et al. [15] reviewed an extensive article on the use of data mining methods and compared the results from a different algorithm for applications of global solar radiation prediction. Kannan et al. [16] designed a forecast model based on multiple linear regression techniques to collect three months' weather data and got the 52% accuracy. Nikam and Meshram [17] proposed a rainfall prediction model in India and collected data from Indian Meteorological Department based on a Bayesian prediction classifier with an accuracy of 81.66%. Rashid et al. [18] introduced a machine learning techniques for the detection of learning style in e-learning system using classification algorithm. Bagirov et al. [19] introduced forecasting of monthly rainfall in Victoria based on Cluster wise linear regression approach. Kusaik et al. [20] proposed data mining approaches to predict the amount of rainfall using radar reflectivity data. Radhika and Shashi [21] worked on MLP with a back propagation algorithm to predict maximum temperature based on the present temperature data. Esteves et al. [22] and Cakir et al. [23] proposed a back propagation feed-forward neural network for the prediction of weather uncertainty. Scher and Messori [24] introduced a machine learning technique for weather prediction. In a recently published paper, Ali et al. [25] presented a new gradient boosting technique to predict regression datasets. Declercq et al. [27] introduced the novel ML techniques for the prediction of renewable energy.

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. Novel techniques of ML are used to forecast rainfall in terms of accuracy and other different evaluation metrics. 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. The stages of ML models implementation are displayed in **Fig.3.**

3.1 Data Acquisition

Based on everyday observations and data availability of different Australian weather stations, we have used the binary

classification dataset collected by the Kaggle competitive website Platform [26]. It contains the dataset (rows*columns) consisting of 145460 records and 23 features. Online access to Australian meteorological data is considered a good opportunity for conducting weather forecasting-related research work. Ten years of data for Australian weather stations over the period from 2007 to 2017 is obtained for the prediction 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 Speed9am (WS9), Wind Speed3pm (WS3), Humidity9 am (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 features or 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 given in **Table 1**. The following dataset contains float and object values. **Fig. 4 (a)** and **(b)** represent bar and pie plots respectively, in which 69.6% of data contains a float value, and 30.4% contain an object value. A dataset is divided into two sets: 75% is used for training, and 25% is used for testing. To obtain the aim of this work, twenty-two columns are selected as input data for binary classification models, and one column is for the target variable.

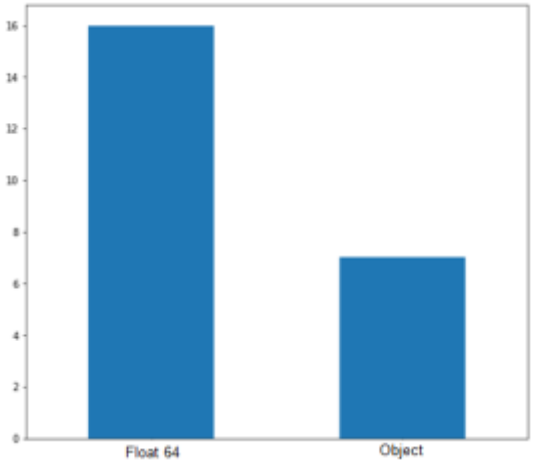
Table 1: Description of Australia weather stations features dataset [26]

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.

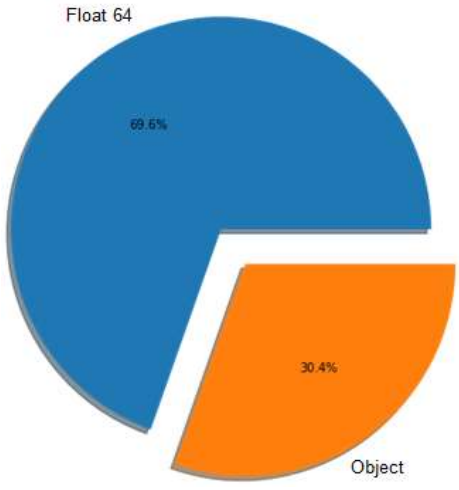
The entire data set contains two class labels in the weather classification task, namely Yes or No for Rain Today & Rain Tomorrow. Furthermore, the dimension of the whole dataset is divided into training and testing data (109095 for training and 36365 testing). The dataset in MS Excel File is to be imported from the PC or Google drive in a python software package.

3.2 Feature Engineering

For better model compatibility with the dataset, features and samples need to be preprocessed before training. After collection of data, 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.



(a)



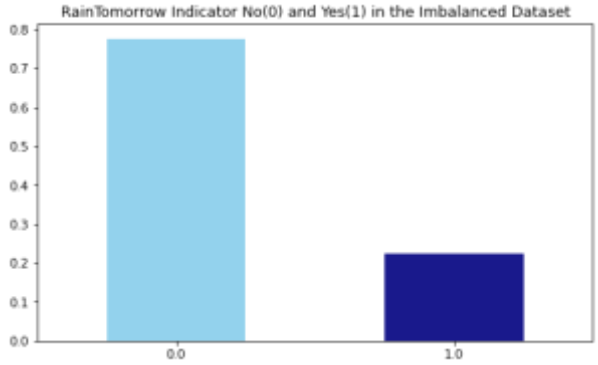
(b)

Fig. 4 (a) Bar, and (b) Pie plot of percentage distribution of data

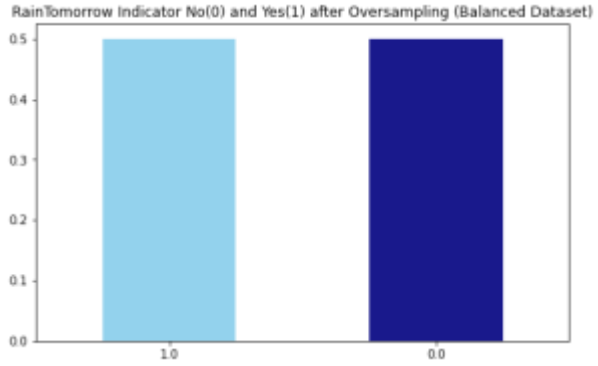
3.2.1 Handling the missing values

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. **Fig. 5(a)** 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 predictor variables also contain the null/missing variables, as shown in **Table 2**. But 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 order to bring data in machine-readable form, a label encoder is used for converting the labels into a numeric number.



(a)



(b)

Fig. 5 Data set (a) Before Sampling, and (b) After Sampling

Table 2: Feature contains missing value in percentage

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: Features contains more number of null values

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

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 and Q3+1.5*IQR
4. Finally the data points greater than the upper limits or less than the lower limits are outliers.

Table 4: Features contains outliers

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		

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 & 1. It is also called a min-max scaling, which is described as follows:

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

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 [28]. 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, 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 is 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.**

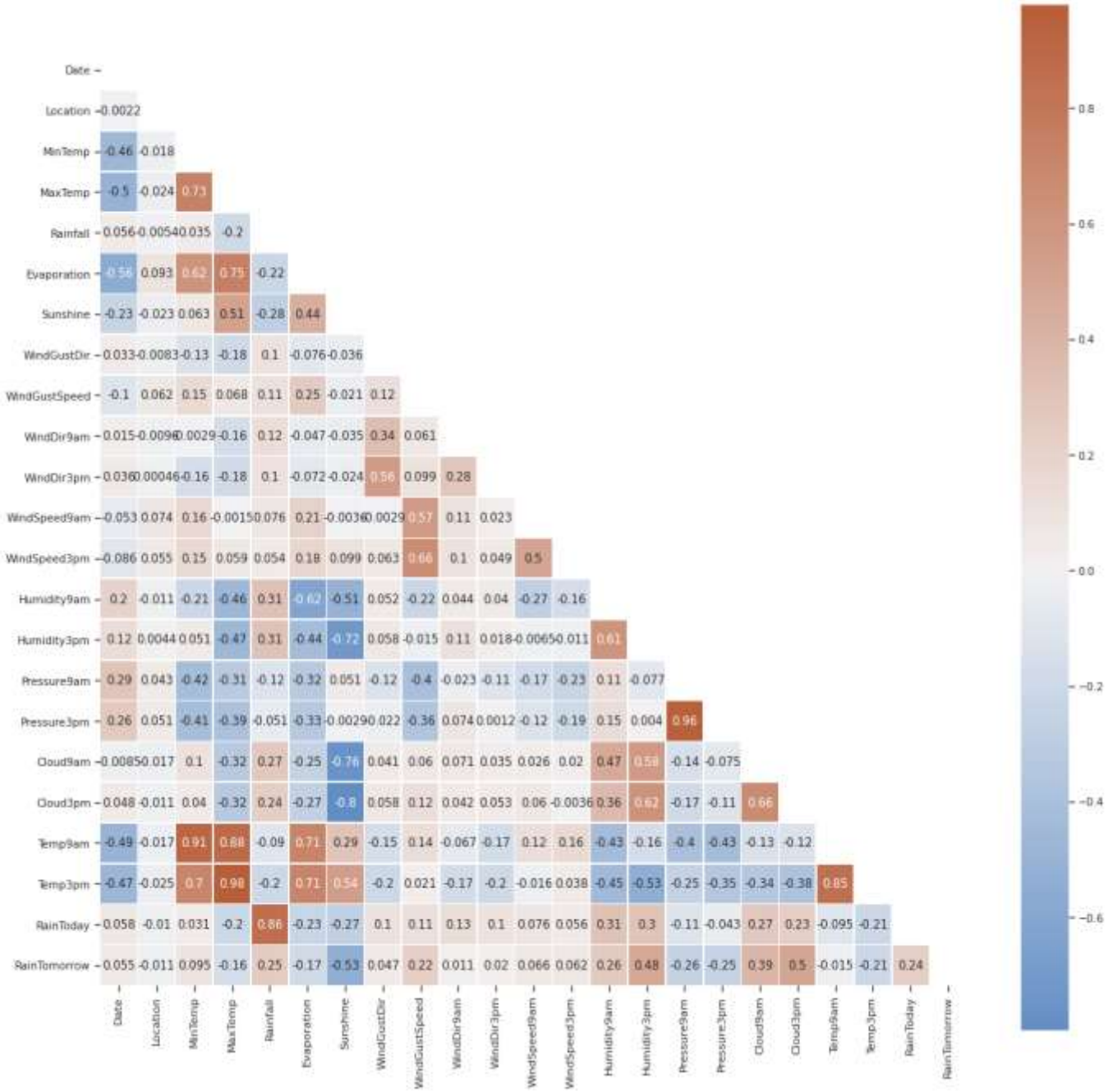


Fig. 6 Pearson correlation heat map between different features in the dataset

But 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.

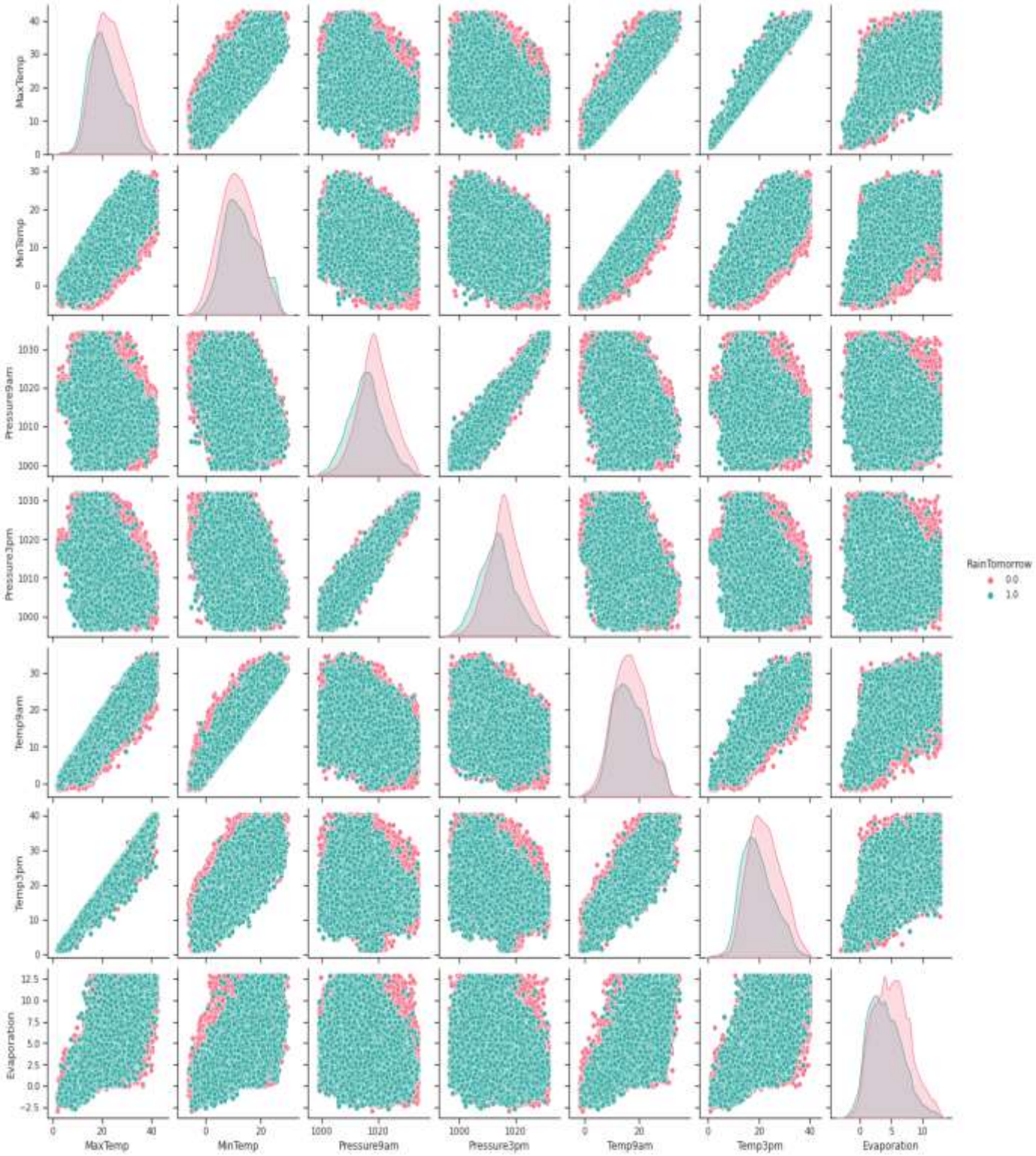


Fig. 7 Pair Wise Plot showing the distribution of each feature based on the other feature

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.

$$G = \sum_{c=1}^{c} \hat{p}_{jc}\left(1 - \hat{p}_{jc}\right) \quad (2)$$

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 [29]. 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 $y = 0$ [30]. Logistic regression is an approach to fitting models using logistic functions [31]. In the present study, since the dependent variable was discrete, so the sigmoid function is given as follows:

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

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:

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

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 outputs 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 [32]. 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, the three boosting technique such as LGB, CB, and XGB are proposed for binary classification problem, 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, 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 [33]. Comparing the analog's 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 ICU utilization, and enhances accuracy. **Fig. 8** represents the process of LGBM mechanism.

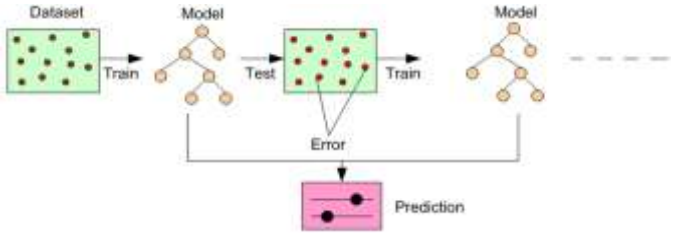


Fig. 8 Process of LGBM mechanism

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[34].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 [36,37]. 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 XGB mechanism.

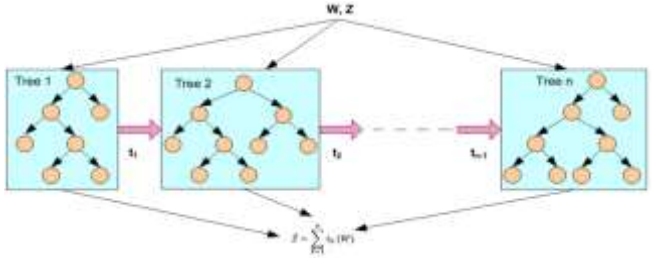


Fig. 9 Process of XGB mechanism

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).

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

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:

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

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:

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

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:

$$\text{F1 Score} = \frac{2}{\frac{1}{Recall} * \frac{1}{Precision}} \quad (8)$$

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

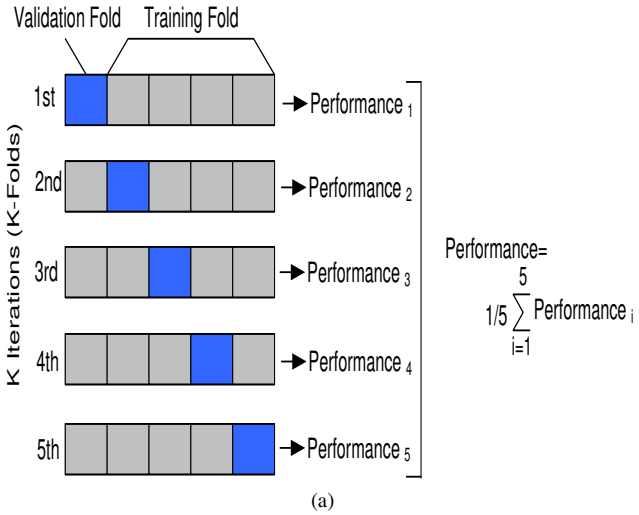
It summarizes the ratio of correctly predicted negative results divided by the total number of true negative results and false-positive results.

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 [35]. Here 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.

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

A schematic representation of the 5th, 7th and 10th fold cross-validation is shown in Fig. 10 (a), (b), and (c), respectively.



(a)

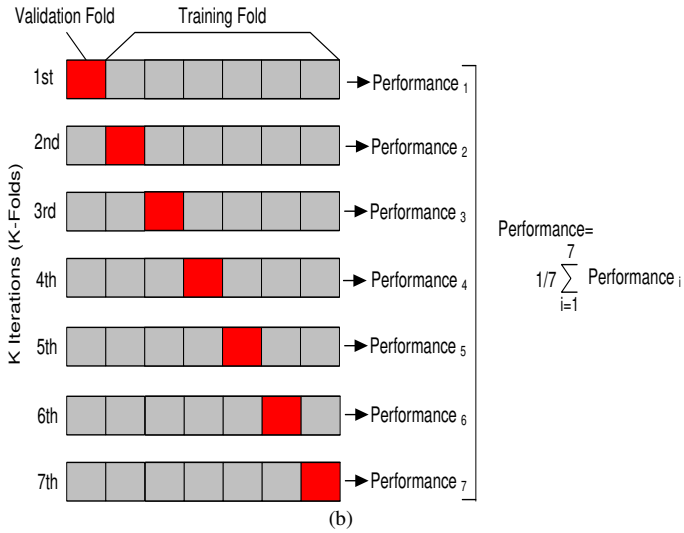


Fig. 10 (a) 5th, (b) 7th and, (c) 10th fold cross validation

6. Experimental Results and Discussion

This section describes the experimental results and comparison with other ML classifiers and the related works. The model assumptions must also be reviewed to ensure accuracy, and the necessary corrections must be made. In addition, important feature selection and correlation are used to reduce the number of factors that are taken into consideration for accelerating the prediction process. The ML models applied to the dataset observe the competency of the algorithms for 'k-fold' cross-validation. The 75% of the dataset is used for the training purpose of each ML algorithm and 25% of the testing dataset is used to check the model efficiency. For an Australian weather data set [26], 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

This 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. $k = 5, k = 7, \text{and } k = 10$ are used for training and testing datasets for all the models as depicted in **Table 5**. The comprehensive view of the k-fold (i.e. $k = 5, 7, 10$ for the performance of this model is represented in **Fig. 10.** From the above three selected k-fold values, the most accurate prediction was found when the proposed dataset is performed with CV=10 in which data is folded into ten training and one testing pair. It has been seen that 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.

Table 5: Performance of 'k-fold' CV

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

6.2 Hyperparameter Tuning Results

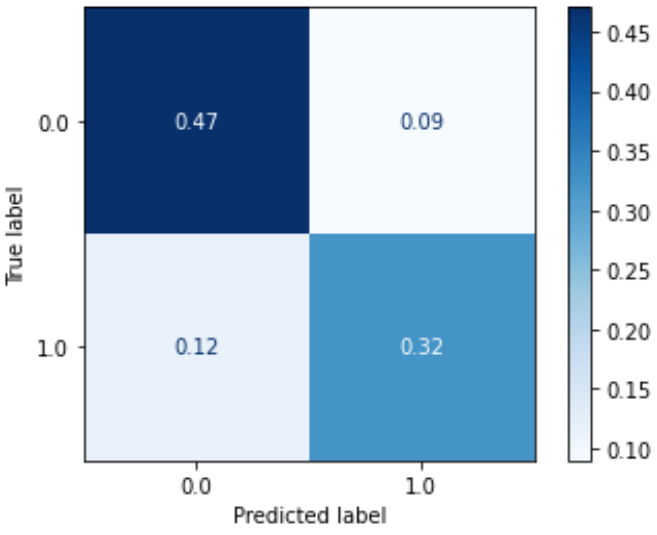
Before training the ML models, a set of optimal parameters are selected using the grid search method from Scikit learn library for each model and obtained the results as shown in **Table 6**. The XGB model gives 95.2% accuracy as compared to other ML models, by selecting the following parameters: n_estimators=500, 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** shows the confusion matrix 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.

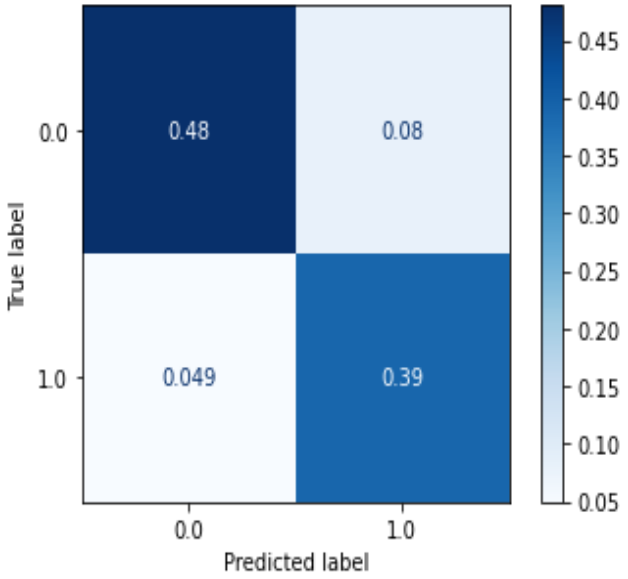
Table 6: Optimal parameters of all ML models

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

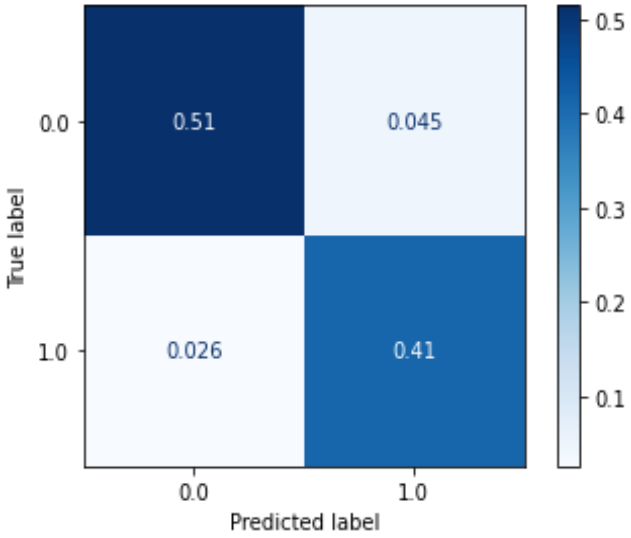
However, TP represents the actual positive value and the model predicted a positive value, FP represents the actual negative value and the model predicted a positive value, TN represents the actual negative value and the model predicted a negative value, and FN represents the actual positive value but the model predicted a negative value. It shows that the XGB model gives the good predicted outcomes for the positive class and negative class both.



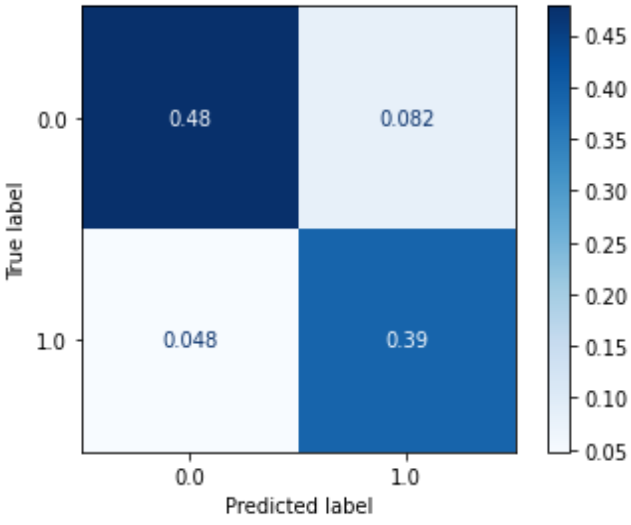
(a)



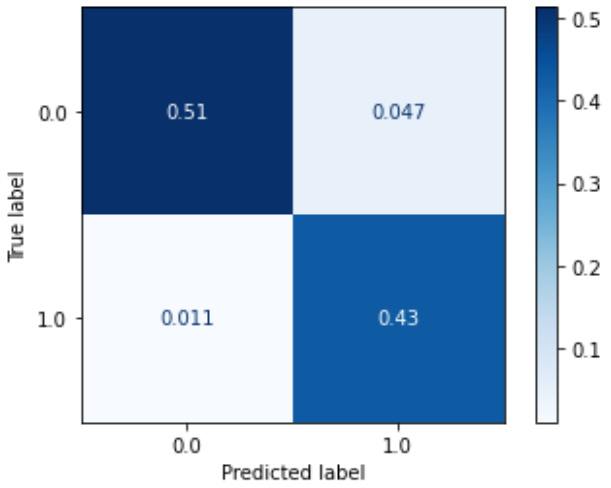
(b)



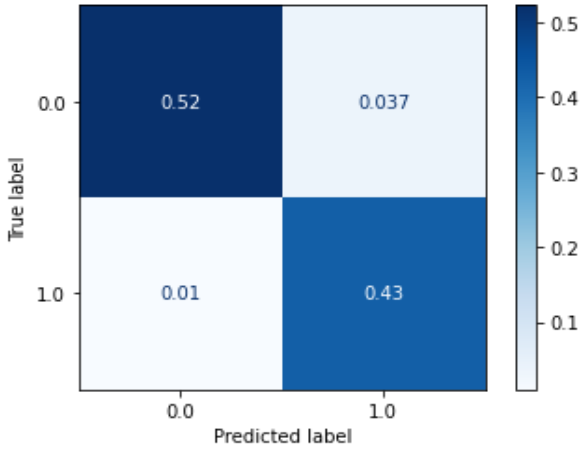
(c)



(d)



(e)



(f)

Fig. 11 Confusion Matrix of (a) LR, (b) DT, (c) RF, (d) LGB, (e) CB, and (f) XGB

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 evaluation metrics are used to check the performance of the classification dataset. 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. For the imbalanced dataset used in this study, more importance is given to F1 Score than accuracy to measure the best-performing algorithm. It can be checked 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. **Table 7** represents the classification report of all proposed ML models.

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 in Fig. 12 indicates that the AUC value of the XGB classifier is 0.99, which is very close to one. Therefore, XGB gives a better performance as compared to other implemented ML models.

Table 7: Classification Report of the all proposed ML models

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

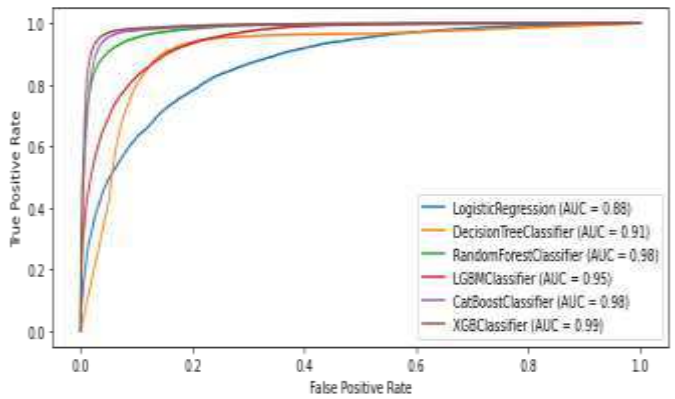


Fig. 12 ROC-AUC curve of implemented classifiers

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. 0.99, which gives a good result, and the LR shows the lowest AUC value of 0.88. 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 0.21to 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.

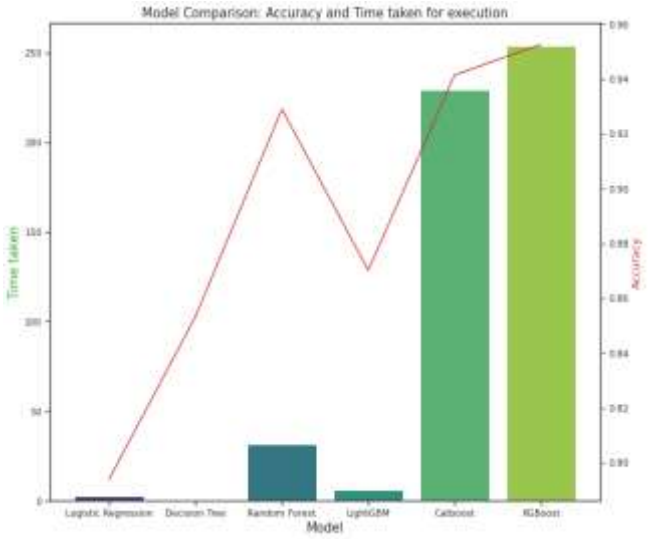


Fig. 13 Comparison of All Models in terms of time taken and accuracy

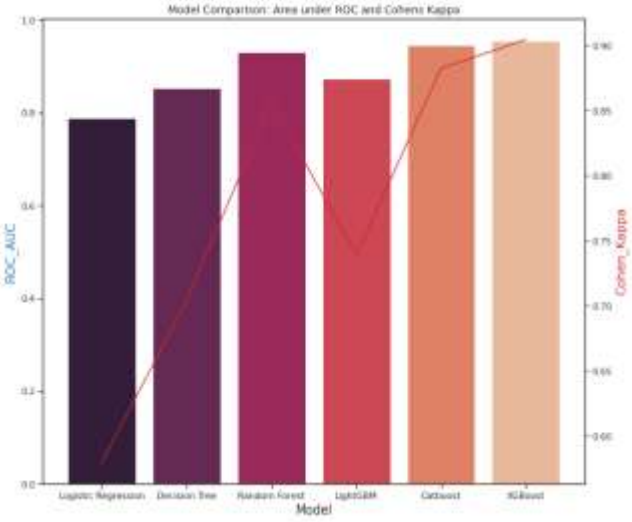


Fig. 14 Comparison in terms of ROC-AUC and Cohen kappa Score

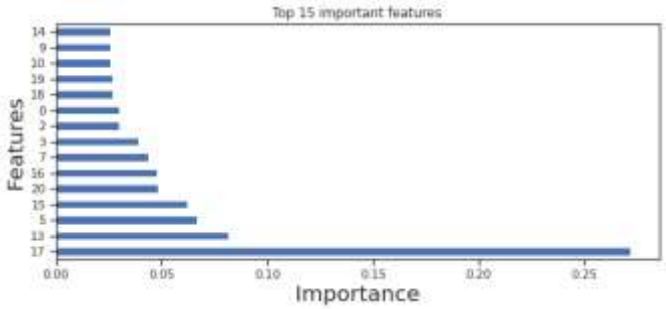


Fig. 15 Top 15 Features of XGB Model

Table 8: Feature Score of XGB classifier

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

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 paper, a different stage of end to end ML life cycle has been envisaged, and the performance of all ML classifiers is observed meticulously. A comparative study is done of all the ML classifiers, and conscientious endeavour has been made to reveal the appropriate methodology of fitting data into the models for getting accurate target variables related to the weather forecast. This paper presents a novel predictive ML algorithm, namely XGB for binary classification imbalanced datasets, which supports the popular Scikit-learn package of Python.

The experiments are performed for the proposed dataset based on two imbalanced classification labels, and performance is then observed by evaluation metrics. Based on highly imbalanced data of the Australian weather dataset, XGB is designed to improve the performance of traditional models to detect rain on the following day. It is observed that data preprocessing steps greatly improve the performance of the models. Using a 10-fold cross-validation test, the results showed that the proposed novel model outperformed other ensemble learning classifiers. This study can be harnessed for a

variety of real-life applications such as image recognition, speech recognition, medical diagnosis etc. In future, this novel method can be used in conjunction with a regression model to improve the reliability of climatic predictions, and to find out the other weather parameters. Moreover, this novel package of ML may be applied for renewable energy applications such as forecasting of solar radiation, wind speed, solar power, and wind energy etc. This may also be used for smart grid infrastructure planning. Most prominently, this may be used for ecological balance, which in turn may save the advanced form of human civilization from extinction.

Author Contribution: Rahul Gupta (conceptualization, methodology, formal analysis, data collection and preparation validation, original draft writing and visualization), Anil Kumar Yadav (writing, review and editing, supervision), SK Jha (review and editing, supervision), and Pawan Kumar Pathak (writing, review and editing and visualization).

Funding: Not applicable.

Data Availability: The data used in this study is available from the Kaggle Website provided https://www.kaggle.com/jsphyg/weather-dataset-rattle-package.

Code Availability: Not applicable.

Declarations

Conflicts of interest: The authors declare that they have no conflict of interest.

Ethics Approval: Not applicable

Consent to participate: Not applicable

Consent for publication: Not applicable

References

[1] Xiao, Z., Liu, B., Liu, H., & Zhang, D. (2012). Progress in climate prediction and weather forecast operations in China. Advances in Atmospheric Sciences, 29(5), 943-957.

[2] Bengtsson, L. (1980). The weather forecast. Pure and applied geophysics, 119(3), 515-537.

[3] Jain, H., & Jain, R. (2017). Big data in weather forecasting: Applications and challenges. In IEEE International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), 138-142.

[4] Rodríguez-Mazahua, L., Rodríguez-Enríquez, C. A., Sánchez-Cervantes, J. L., Cervantes, J., García-Alcaraz, J. L., & Alor-Hernández, G. (2016). A general perspective of Big Data: applications, tools, challenges and trends. The Journal of Supercomputing, 72(8), 3073-3113.

[5] Talia, D. (2013). Clouds for scalable big data analytics. Computer, 46(05), 98-101

[6] Li, K., & Liu, Y. S. (2005). A rough set based fuzzy neural network algorithm for weather prediction. In IEEE international conference on machine learning and cybernetics, 1888-1892.

[7] Hewage, P., Trovati, M., Pereira, E., & Behera, A. (2021). Deep learning-based effective fine-grained weather forecasting model. Pattern Analysis and Applications, 24(1), 343-366.

[8] Haupt, S. E., Cowie, J., Linden, S., McCandless, T., Kosovic, B., & Alessandrini, S. (2018). Machine learning for applied weather prediction. In IEEE 14th International Conference on e-Science (e-Science), 276-277.

[9] Pathak, P. K., Yadav, A. K., & Alvi, P. A. (2021). A state-of-the-art review on shading mitigation techniques in solar photovoltaics via meta-heuristic approach. Neural Computing and Applications, 1-39.

[10] Pathak, P. K., & Yadav, A. K. (2019). Design of battery charging circuit through intelligent MPPT using SPV system. Solar Energy, 178, 79-89.

[11] Jebli, I., Belouadha, F. Z., Kabbaj, M. I., & Tilioua, A. (2021). Prediction of solar energy guided by pearson correlation using machine learning. Energy, 224, 120109.

[12] Potočnik, P., Vidrih, B., Kitanovski, A., & Govekar, E. (2019). Neural network, ARX, and extreme learning machine models for the short-term prediction of temperature in buildings. In Building Simulation, Tsinghua University Press, 12 (6), 1077-1093.

[13] Dueben, P. D., & Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. Geoscientific Model Development, 11(10), 3999-4009.

[14] Wang, F., Zhen, Z., Wang, B., & Mi, Z. (2018). Comparative study on KNN and SVM based weather classification models for day ahead short term solar PV power forecasting. Applied Sciences, 8(1), 28.

[15] Del Campo-Ávila, J., Takilalte, A., Bifet, A., & Mora-López, L. (2021). Binding data mining and expert knowledge for one-day-ahead prediction of hourly global solar radiation. Expert Systems with Applications, 167, 114147.

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

[17] Nikam, V. B., & Meshram, B. B. (2013). Modeling rainfall prediction using data mining method: A Bayesian approach. In Fifth IEEE International Conference on Computational Intelligence, Modelling and Simulation, 132-136.

[18] Rasheed, F., & Wahid, A. (2021). Learning style detection in E-learning systems using machine learning techniques. Expert Systems with Applications, 174, 114774.

[19] Bagirov, A. M., Mahmood, A., & Barton, A. (2017). Prediction of monthly rainfall in Victoria, Australia: Clusterwise linear regression approach. Atmospheric research, 188, 20-29.

[20] Kusiak, A., Wei, X., Verma, A. P., & Roz, E. (2012). Modeling and prediction of rainfall using radar reflectivity data: A data-mining approach. IEEE Transactions on Geoscience and Remote Sensing, 51(4), 2337-2342.

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

[22] Esteves, J. T., de Souza Rolim, G., & Ferraudo, A. S. (2019). Rainfall prediction methodology with binary multilayer perceptron neural networks. Climate Dynamics, 52(3), 2319-2331.

[23] Cakir, S., Kadioglu, M., & Cubukcu, N. (2013). Multischeme ensemble forecasting of surface temperature using neural network over Turkey. Theoretical and applied climatology, 111(3), 703-711.

[24] Scher, S., & Messori, G. (2018). Predicting weather forecast uncertainty with machine learning. Quarterly Journal of the Royal Meteorological Society, 144(717), 2830-2841.

[25] Ali Shehadeh, A., Alshboul, O., Al Mamlook, R. E., & Hamedat, O. (2021). 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, 103827.

[26] https://www.kaggle.com/jsphyg/weather-dataset-rattle-package

[27] De Clercq, D., Jalota, D., Shang, R., Ni, K., Zhang, Z., Khan, A., & Yuan, K. (2019). Machine learning powered software for accurate prediction of biogas production: A case study on industrial-scale Chinese production data. Journal of Cleaner Production, 218, 390-399.

[28] Vergara, J. R., & Estévez, P. A. (2014). A review of feature selection methods based on mutual information. Neural computing and applications, 24(1), 175-186.

[29] Yang, Y., & Loog, M. (2018). A benchmark and comparison of active learning for logistic regression. Pattern Recognition, 83, 401-415.

[30] Ekström, M., Esseen, P. A., Westerlund, B., Grafström, A., Jonsson, B. G., & Ståhl, G. (2018). Logistic regression for clustered data from environmental monitoring programs. Ecological Informatics, 43, 165-173.

[31] Huang, T., Li, B., Shen, D., Cao, J., & Mao, B. (2017). Analysis of the grain loss in harvest based on logistic regression. Procedia Computer Science, 122, 698-705.

[32] Genuer, R., Poggi, J. M., Tuleau-Malot, C., & Villa-Vialaneix, N. (2017). Random forests for big data. Big Data Research, 9, 28-46.

[33] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30, 3146-3154

[34] Hussain, S., Mustafa, M. W., Jumani, T. A., Baloch, S. K., Alotaibi, H., Khan, I., & Khan, A. (2021). A novel feature engineered-CatBoost-based supervised machine learning framework for electricity theft detection. Energy Reports, 7, 4425-4436.

[35] Fushiki, T. (2011). Estimation of prediction error by using K-fold cross-validation. Statistics and Computing, 21(2), 137-146.

[36] Tao, H., Awadh, S. M., Salih, S. Q., Shafik, S. S., & Yaseen, Z. M. (2021). Integration of extreme gradient boosting feature selection approach with machine learning models: application of weather relative humidity prediction. Neural Computing and Applications, 1-19.

[37] Asselman, A., Khaldi, M., & Aammou, S. (2021). Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. Interactive Learning Environments, 1-20.