Comparative Analysis of Selected Classification Algorithms on Weather Data

Shakirat Ronke YUSUFF[1] Ibrahim ODETUNDE[2] Jumoke Falilat AJAO[3]

KWASU Journal of Information, Communication and Technology (KJICT.) 2023, 2(1)
© 2023 Faculty of Information and Communication Technology, Kwara State University, Malete, Nigeria.
URL: http://kjict.ng/

1,2,3 Department of Computer Science, Kwara State University, Malete, Nigeria.

Abstract

Numerous human daily activities, such as transportation, farming, and commerce, are weather-dependent. Scientists have used a range of techniques to carry out weather forecasting, some more accurate than others, to predict environmental variables. Critical information about the weather is provided through weather forecasts. Accurate weather forecasting would assist the government in planning to prepare its citizens for the worst, since severe weather annually damages life, property, and many government projects that are typically substantially funded. In this study, a comparative analysis is carried out on three classification algorithms: random forest, decision tree and extra tree classifiers on weather data. Two sets of predictions were performed on weather data: the precipitation type and weather summary. With the prediction of precipitation type, the accuracy scores for both random forest and decision tree were 99.71%, while extra tree classifiers achieved 99.44% accuracy. The confusion matrix shows better prediction of records by random forest and decision tree classifiers, with a score of 99.71% each, while the extra tree classifier performed the least, predicting 99.69% records correctly. Also, the Classification report revealed that random forest had a precision of 99.84%, recall of 98.69% and F1-score of 99.26%. The decision tree classifier had a precision of 99.84%, recall of 98.69% and F1-score of 99.26%, while extra tree classifier had a precision of 99.18%, recall of 99.24% and F1-score of 99.21%. With the second set of predictions using weather summary, random forest had an accuracy score of 66.05%, decision tree had 53.9%, and the extra tree had 64.8% accuracy scores. As for the confusion matrix, random forest performed best with 66.17% records correctly predicted, extra tree classifier with 64.8% correct records and decision tree with 53.6% records. The Classification report showed random forest had a precision of 69.6%, recall of 47.8% and F1-score of 52.2%; decision tree achieved a precision of 17.39%, recall of 39.13% and F1-score of 30.43% while extra tree had a precision of 52.17%, recall of 30.43% and F1-score of 21.74%. In conclusion, the random forest classifier has proven to consistently outperform the other classifiers in this study. Further studies could focus on predicting future weather occurrences.

Keywords: Weather, Classification algorithm, Random Forest, Decision Tree, Extra Tree, Prediction

1. Introduction

One of the most experimental and challenging technological problems in the 20th century is weather forecasting (Abrahamsen et al., 2018). Environmental change has long been a subject of substantial discussion because of the sudden shifts it causes. It is difficult to predict the weather accurately currently because of the limitations on the improved execution of weather forecasting. In meteorology, the practice of weather forecasting is crucial because they are used to protect lives and property. Nonetheless, the ability to make accurate predictions is one of the main challenges faced by meteorologists globally.

Weather forecasts are useful for planning outdoor activities, growing crops, managing time, operating hydropower plants, managing floods, and other human-related activities. Forecasts based on weather, outlook, humidity, and wind are crucial

to farming and, by extension, to traders in product markets. Temperature forecasts are used by utility managers to assess future demands (Abrahamsen et al., 2018). Estimates can be used to plan activities around severe weather as well as to prepare for and survive it because it has a significant negative influence on outdoor activities. Instances of severe weather include heavy rain, snowfall, and wind chill. Without accurate weather forecasts, people could be injured or worse if they were involved in dangerous situations when they are unprepared. The processing capacity of technology has not increased enough to perform weather prediction effectively. Even if the meteorologist was able to construct more sophisticated models that required complex computation and enhanced prediction abilities, thanks to the capabilities of modern computer systems, the accuracy and the timely forecast of weather occurrences remain significant problems. The significance of timely and precise weather forecasting has also grown because of global climate change and the occurrence of some catastrophic weather phenomena. With the Kogi State flooding that occurred in November 2022, Dr. Zakari Usman, the Kogi state Commissioner for Health reported during a press Conference in Lokoja, Nigeria that over 514 villages were impacted, 471,000 people were relocated, 92 medical institutions were destroyed, and 24 people died, having a significant negative impact on the state's health (Omole, 2022). The 2018 flooding in Ogbaru Local Government Area of Anambra State, Nigeria had a record of about 11,180 persons displaced, affecting the 13 communities out of 16 in the LGA, costing farmers their produce and the community residents their property (Nwabueze, 2018). Russia has experienced both the deadliest heat wave and the worst drought in recorded human history. Heavy rains caused a record-breaking natural disaster that affected Australia and Colombia. The World Economic Forum has also raised the alarm, stating that one of the biggest global concerns facing people in the twenty-first century would be climate change. Although there is no way to stop natural disasters, early and accurate forecasts can help to lessen the number of lives lost and other costs, which are the motivations behind the development of weather forecasting models. A more accurate model can predict the event as soon as feasible and with great accuracy. Numerous regional, national, and international organisations throughout the world are working to improve climate forecasts, including the World Meteorological Organisation (WMO), the International Research Institute (IRI), and the World Climate Application and Services Program (WCASP). The primary contributors are those who are dedicated to employing high-quality research to predict environmental changes, protect people and property, and provide decision-makers with accurate, timely, and scientific information.

On weather data, a variety of machine learning techniques have been used to forecast climatic variables like temperature, wind speed, rainfall, and meteorological pollution. Artificial Neural Networks (ANN) (Abhishek et al, 2012), Naive Bayesian Networks (Wu et al, 2012), Support Vector Machines (SVM) (Jayasingh & Mantri, 2020), Fuzzy Logic, Rule-based Techniques, which include Memory-based reasoning Techniques and Genetic Algorithms, are some of the most frequently used machine learning techniques for weather prediction. Despite good performance, some of the shortcomings of these algorithms include complexity, black box nature and time taken in making predictions by the machine, which has made their adoption in the real world slow. It is, therefore, imperative to conduct a comparative analysis of some selected classification algorithms on weather data, which aligns with this study's aim. Overall, this research contributes to the meteorological sector by analysing the performance of Random Forest, Decision Tree and Extra Tree Classifiers in predicting precipitation type and weather summary. It underscores the importance of adopting simple classification algorithms on weather data to enhance prediction accuracy in a timely and less complex manner.

2. Literature Review

In the past ten years, numerous attempts have been made to address the problem of weather forecasting, using statistical modelling, including machine learning systems, with published results. Examples include artificial neural networks, predictive analysis in Apache Hadoop Framework using Naive Bayes Algorithm, and neural network-based algorithms using Back Propagation Neural Network (BPN) and Hopfield Network, Recurrence Neural Network (RNN), Conditional Restricted Boltzmann Machine (CRBM), and Convolutional Network (CN) models.

Sujatha et al. (2024) aimed to design a model using Random Forest and K-Nearest Neighbor algorithms to predict weather events, using Python to reach 100% accuracy in the prediction. At the end of the research work, the random forest algorithm achieved

an accuracy of 86.77, precision of 69.05, F1_score of 62.15, recall of 56.53, and misclassification error of 13.22 compared to k-nearest neighbor with 84.35 and 59.38, 68.67, 57.48 and 16.08, respectively. The random forest algorithm achieved higher accuracy and precision scores, while the K-nearest neighbor algorithm achieved higher F1-score and recall, but at the end of the research work, the goal of the project, which was to achieve a 100% accuracy score, was not achieved.

Xhabafti et al. (2024) performed a comparative study by designing a model based on selected statistical methods and deep learning models for weather prediction based in Albania. The model is made up of some statistical methods, such as Auto Regressive Integrated Moving Average (ARIMA), Error, Trend & Seasonal (ETS) and some deep learning models such as Long Short-Term Memory Network (LSTM) and Deep Forward Neural Network (DFNN). The analysis of the study was divided into three parts, which are the comparison of ARIMA and ETS, LSTM and DFNN, while the third part is the comparison of the best in the first and second parts. As for the result, the Root Mean Square Error (RMSE) was used as the performance metrics. In the first part of the analysis, ARIMA had an RMSE of 38.06956 while ETS with 33.89779. The ETS model performed better with the lower RMSE in the prediction. The second part of the analysis, LSTM had an RMSE of 55.234002, while DFNN with 53.723089. The DFNN outperformed the LSTM based on the RMSE metrics, which is lower. The last part of the analysis sees the ETS model compared with the DFNN model and with the RMSE as the main metric. The ETS model performed better with lower RMSE 33.89779 compared with DFNN which is 53.723089.

Ajina et al. (2023) developed an application (Graphical User Interface) using Python to predict the weather condition of a specific city and compare the data with different cities. The Artificial Neural Network with backpropagation algorithm was used. With this in place, an experiment was done to prove the impact of COVID-19 epidemic on the meteorological forecast accuracy. The data used for the comparison was from March to May 2020 (During COVID-19) and March to May 2015 to 2019 (pre-COVID-19). The experiments were carried out in various cities in India, the Bangalore, Mumbai, Kolkata, Chennai and Delhi. At the end of the experiments, it was discovered that due to the drastic reduction of airplane transportation services and aircraft-related monitoring from March to May 2020, caused by the nationwide lockdown during the COVID-19 period, the predicted accuracy during the COVID-19 period is different from the average accuracy of the pre-COVID-19 period. It is concluded that a global pandemic can affect the result of weather forecasts, which can impact future weather predictions.

AbdulRaheem et al (2022) developed a model to compare three machine learning algorithms: Decision tree (DT), k-nearest neighbor(k-NN) and logistic regression (LR) on weather data with the aid of a web-based application developed using Flask App. At the end of the study, the decision tree algorithm performed best with a 100% accuracy score, outperforming the logistic regression and k-nearest neighbor, which had an accuracy score of 93% and 78%, respectively. In terms of rain prediction, the decision tree achieved an appropriate score of accuracy.

Muthukumar et al. (2021) used a two-stage model capable of learning spatiotemporal trends based on data from ground-based sensors monitoring air pollutants and meteorological features, and remote sensing satellite imagery of air pollution to anticipate PM2.5 atmospheric air pollution in Los Angeles, USA. The resulting model used meteorological and air pollution data from the ground-level sensors and remote sensing satellite imagery from roughly 10 days ago to predict spatiotemporal PM2.5 in terms of micrograms per cubic meter (gm3) at seven sensor locations in Los Angeles County every 46 hours at intervals of approximately 10 days in the future. However, the study was limited to Los Angeles in the United States.

Support vector machine (SVM) and multiple linear regressions were used by Jyothi et al. (2020) to create an efficient rainfall prediction agent model. For this, the decision tree and linear regression algorithms were used with Support Vector Regression (SVR) and implemented in MATLAB. Two sets of experiments were run; the first using SVR to create a time series forecasting agent model and the second utilising the multivariate regression issue to create a visualising agent model. The accuracy score increased by 15.9% because of Support Vector Regression (SVR). Quantitative estimation can therefore be produced by increasing the complexity of time series data for diverse situations of weather patterns with more parameters.

Kashikar et al. (2019) designed a weather

prediction model for Agricultural purposes. The functional regression algorithm and linear regression were both used in the approach, which was carried out using Python, the Pandas, NumPy, and Scikit Learn libraries. The outcome of the experiment is as follows: Explained Variance: 0.94; Mean Absolute Error: 1.07°C; Median Absolute Error: 0.81°C. According to the test results, the system performed in line with the project's intended goals and objectives. The weather prediction model system was able to predict the weather accurately and efficiently.

Abrahamsen et al. **(**2018) used an Artificial Neural Network (ANN) to attempt to predict the temperature. ANN is employed in this process. The temperature is tested in four models, which are the prediction of the temperature 1, 3, 6, and 12 hours in advance. For each model, two experiments were conducted. Only temperature was used as an input to the networks in the first experiment using Auto-Regressive Neural Network (AR-NN). As the prediction period increased from 1 hour to 12 hours, the test error also increased, starting at 0.0101 for the 1-hour model, 0.0318 for the 3-hour model, 0.0608 for the 6-hour model and reaching 0.0894 for the 12-hour model with regression horizons 24, 48, 48 and 169, respectively. In the second experiment, an Autoregressive Neural Network (ARXNN) with exogenous input was created by adding precipitation data to the network. As the prediction period increased from 1 hour to 12 hours, the test error increased as well, starting at 0.0133 for the 1-hour model, 0.0653 for the 3-hour model, 0.0946 for the 6-hour model and reaching 0.0991 for the 12-hour model with regression horizons 24, 48, 48 and 24, respectively. Although there was a slight increase in test error with ARXNN compared to the ARNN, it has been demonstrated that adding precipitation as an input to the ARXNN model slightly enhances forecast accuracy, as the regression of the 12-hour model reduced drastically to 24. Consequently, it might be intriguing to add more input to the model. Studying if adding data from other geographical places can enhance prediction outcomes is of relevance.

Mohammad et al. (2014) aimed to predict weather using MkNN algorithm and compare results with the following 5 algorithms: 1. Linear Regression, a model based on regression, 2. The Support Vector Machine version (SMO reg), 3. A type of artificial neural network, the multilayer perception. 4. RepTree, an approach based on decision trees, 5. CP-KNN, a KNN-based climate prediction model. The results of the study are arranged respectively as (Linear Regression, SMOReg, Multilayer Perception, RepTree, Knn, MkNN) Mean Absolute Error: (2.703, 5.177, 2.766, 2.738, 3.620, 2.080), Root Mean Square Error: (3.226, 6.384, 3.206, 3.144, 3.646, 2.114), Mean Absolute Percentage Error (24.943, 49.202, 23.435, 27.661, 5.679, **3.279**), Mean Squared Error (10.409, 40.751, 10.282, 9.884, 13.294, 4.471). The findings demonstrate that, in comparison to existing data mining prediction strategies, the suggested model yielded remarkably good results. To make the forecast more comprehensive and for a wider area, the model can be expanded to include additional attributes (such as solar radiation, the water cycle, air pressure, and earth rotation, among others). Additionally, it can be used to predict the climate, doing so on a seasonal to inter-annual time frame.

Scientists have tried their best to forecast environmental characteristics using several methods, and some of these methods are more accurate than others. In weather data, a variety of machine learning techniques are used to forecast climatic variables like temperature, wind speed, rainfall, and meteorological pollution. Artificial Neural Networks (ANN), Naive Bayes Networks, Support Vector Machines (SVM), Fuzzy Logic, Rule-based Techniques, which include Memory-based reasoning Techniques and Genetic Algorithms, are some of the most frequently used machine learning techniques for weather prediction.

Some of the shortcomings with these algorithms include the complexity of the algorithm and human error (during calculation process) due to the mathematical nature of the algorithms and time taken in making prediction by the machine.

This research will perform a comparative analysis of the classifiers: Random Forest, Decision Tree and Extra Tree classifiers on the weather data provided to it so as to provide the best of them. The choice of these classifiers is due to the non-complexity (ease of implementation) and non-mathematical nature of the classifiers which does not involve calculation process that could lead to failure due to human error.

A summary of the reviewed literature is presented in Table 1.

Table 1: *Summary of Existing Studies*

AUTHOR	TITLE	METHODOLOGY	RESULTS	LIMITATIONS/ INFERENCE
Sujatha et al. (2024)	Weather Prediction Using Machine Learning Algorithms	Random Forest and K-nearest neighbor algorithm using Python	The Random Forest algorithm performed better with accuracy of 86.77, precision of 69.05, F1_score of 62.15, recall of 56.53, and misclassification error of 13.22 compared to k-nearest neighbor with 84.35 and 59.38, 68.67, 57.48 and 16.08, respectively	Random forest algorithm performed better but was unable to reach the goal of the project which is 100% accuracy score of the weather prediction.
Xhabafti et al. (2024)	A Comparative Study of Statistical and Deep Learning Model-Base weather Prediction in Albania	The model is made up of some statistical methods, such as Auto Regressive Integrated Moving Average (ARIMA), Error, Trend & Seasonal (ETS) and using some deep learning models such as Long Short-Term Memory Network (LSTM) and Deep Forward Neural Network (DFNN	Three parts of the experiments were conducted, which are the comparison of ARIMA and ETS, LSTM and DFNN, while the third part is the comparison of the best in the first and second parts. As for the result, the Root Mean Square Error (RMSE) was used as the performance metrics. In the first part of the analysis, ARIMA had an RMSE of 38.06956 while ETS with 33.89779. The ETS model performed better, having lower RMSE in the prediction. The second part of the analysis, LSTM had an RMSE of 55.234002, while DFNN with 53.723089. The DFNN outperformed the LSTM based on the RMSE metrics, which is lower. The last part of the analysis sees the ETS model compared with the DFNN model and with the RMSE as the main metric, the ETS model performed better with lower RMSE 33.89779 compared with DFNN, which is 53.723089	The experiment is limited to Albania with records from January 1901 to December 2022, the statistical method had a more optimised error compared to the deep learning model.
Ajina et al. (2023)	Prediction of weather forecasting using artificial neural networks	The Artificial Neural Network with back-propagation algorithm was used	It was discovered that due to the drastic reduction of airplane transportation services and aircraft-related monitoring from March to May 2020, caused by the nationwide lockdown during the COVID-19 period, the predicted accuracy during the COVID-19 period from the average accuracy of the pre-COVID-19 period	A global pandemic can affect the accuracy of weather forecasts, which can impact future weather predictions.

AbdulRaheem et al (2022)	Weather prediction performance evaluation on selected machine learning algorithms	The Decision tree (DT), k-nearest neighbor(k-NN) and logistic regression (LR) algorithm were used to develop a model with the aid of a web-based application developed using Flask App	At the end of the study, the decision tree algorithm performed best with a 100% accuracy score, outperforming the logistic regression and k-nearest neighbor, which have an accuracy score of 93% and 78%, respectively	In terms of rain prediction, the decision tree achieved an appropriate score of accuracy
Muthukumar et al (2021)	Predicting PM2.5 atmospheric air pollution using deep learning with meteorological data and ground-based observations, and remote-sensing satellite big data	A two-stage model capable of learning spatiotemporal trends based on remote-sensing satellite imagery of air pollution and data of ground-based sensors monitoring air pollutants and meteorological features.	The resulting model predicts spatiotemporal PM2.5 in terms of micrograms per cubic meter (μgm3) at seven sensor locations in Los Angeles County every 46h at roughly 10 days in the future intervals using meteorological and air pollution data of remote-sensing satellite imagery and ground-level sensors from roughly 10 days in the past.	The model was designed to only work in Los Angeles County, with a concentration on PM2.5 air pollution
Jyothi et al (2020)	Weather Prediction Through Machine Learning	Decision Tree and linear regression algorithms were used with SVM and were implemented in MATLAB.	On the data, two sets of experiments were run, the first using SVR to create a time series forecasting agent model and the second utilising the multivariate regression issue to create a visualising agent model. The Support Vector Regression (SVR) resulted in 15.9% improved accuracy rate	Increasing the Complexity of time series data for various cases of weather patterns with a greater number of parameters can result in quantitative estimation.
Kashikar et al (2019)	Weather Prediction using Scikit-Learn	Algorithm used for this is the linear regression and a variation of functional regression and implemented using Python, Pandas library, NumPy library as well and the Scikit Learn library.	The Explained Variance: 0.94 The Mean Absolute Error: 1.07°C The Median Absolute Error: 0.81°C	Based on the test results, the whole system performed according to the designed aim and objectives of the project. The weather prediction model system was able to forecast weather for next 7 days to 1 month with accuracy and efficiency.

Abrahamsen et al (2018)	Machine Learning in Python for Weather Forecast based on Freely Available Weather Data	The Artificial Neural Network (ANN) is used in this work. Four separate models were trained to predict the temperature 1, 3, 6, and 12 hours ahead. Two experiments were done for each model. In the first experiment, only temperature was used as input to the networks. This constitutes an auto-regressive neural network (AR-NN). In the second experiment, precipitation data was introduced into the network, forming an autoregressive neural network with exogenous input (ARXNN).	As the prediction period increased from 1 hour to 12 hours, the test error also increased, starting at 0.0101 for the 1-hour model, 0.0318 for the 3-hour model, 0.0608 for the 6-hour model and reaching 0.0894 for the 12-hour model with regression horizons 24, 48, 48 and 169 respectively. In the second experiment, an Autoregressive Neural Network (ARXNN) with exogenous input was created by adding precipitation data to the network. As the prediction period increased from 1 hour to 12 hours, the test error increased as well, starting at 0.0133 for the 1-hour model, 0.0653 for the 3-hour model, 0.0946 for the 6-hour model and reaching 0.0991 for the 12-hour model with regression horizons 24, 48, 48 and 24, respectively.	Although there was a slight increase in test error with ARXNN compared to the ARNN but it has been demonstrated that adding precipitation as an input to the ARXNN model slightly enhances forecast accuracy as the regression of 12-hour model reduced drastically to 24. Introducing precipitation as an input in the ARX model was shown to slightly improve the prediction performance. Hence, it may be interesting to extend the model with other inputs. Mainly, it is of interest to study whether the introduction of data from other geographical locations can improve the prediction results.
Mohammad et al (2014)	Weather prediction using classification	The kNN (k-Nearest Neighbour) and MkNN (Modified k-nearest Neighbour) algorithms was used for this with JAVA programming language	The results are arranged respectively as (Linear Regression, SMOReg, MultiLayer Perception, RepTree, Knn, MkNN) • Mean Absolute Error: (2.703, 5.177, 2.766, 2.738, 3.620, 2.080) • Root Mean Square Error: (3.226, 6.384, 3.206, 3.144, 3.646, 2.114) • Mean Absolute Percentage Error (24.943, 49.202, 23.435, 27.661, 5.679, 3.279) • Mean Squared Error (10.409, 40.751, 10.282, 9.884, 13.294, 4.471)	The results show that the proposed model produces very good results compared to other data mining prediction techniques. The model can be extended to incorporate other attributes (such as solar radiation, water cycle, air pressure and earth rotation, etc) to make the forecast more general and for larger areas. It can also be extended to climate prediction, in which it will predict the climate at the seasonal to inter-annual time scale

3. Material and Method

A weather-based dataset is supplied as input to the prediction system, which comprises the three classifiers. The dataset then goes through data preprocessing, which involves data cleaning by removing from the dataset the null values, blank spaces, features that have no impact on the prediction, as well as converting categorical data to numerical data. The preprocessed dataset goes through the process of feature engineering. Here, the "formatted date" feature is used to generate new features of both hour and day using the sine and cosine function as machine learning is used to encode cyclic and periodic patterns and then visualise the correlation of all features in the dataset to see if any can be dropped. The processed dataset is then split into features and labels. Two rounds of experiments were done using 80% of the data for training and 20% for testing. Firstly, the "precipitation type" with only 2 possible outcomes is set as the target variable, the three classifiers perform their operations and evaluations were done and recorded, after which another round of tests was done, setting "summary" with a possibility of 27 outcomes as the target variable.

Algorithms used in this Study

The following classification algorithms were utilised in this study: Random Forest, Decision Tree and Extra Tree Classifiers.

A. Random Forest Classifier (RFC)

Step 1: Select random samples from a given data or training set.

Step 2: This algorithm will construct a decision tree for every training data.

Step 3: Voting will take place by averaging the decision tree.

Step 4: Finally, select the most voted prediction result as the final prediction result.

B. Decision Tree Classifier Algorithm (DTC)

Step 1: Begin the tree with the root node, S, which contains the complete dataset.

Step 2: Find the best attribute in the dataset using Attribute Selection Measure.

Step 3: Divide the S into subsets that contain possible values for the best attributes.

Step 4: Generate the decision tree node, which contains the best attribute.

Step 5: Recursively make new decision trees using the subsets of the dataset created in **step 3.** Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

C. Extra Tree Classifier (ETC)

Extra tree (short for Extremely Randomised Trees) is an ensemble supervised machine learning method that uses decision. This method is similar to random forests but can be faster. Like Random Forest's technique, the extra trees algorithm generates a large number of decision trees, but each tree's sampling is random and without replacement. This generates a dataset with distinct samples for each tree. For each tree, a predetermined number of features is also randomly chosen from the entire set of features. The selection of a splitting value for a feature at random is the most significant and distinctive aspect of additional trees. The algorithm chooses a split value at random rather than figuring out a locally optimal split using Gini or entropy. As a result, the trees are diverse and uncorrelated. (ArcGIS Pro, nd).

The algorithm for implementing the three Models: RFC, DTC, ETC on weather data is presented in Figure 1, while an explanation of the algorithm parameters is given in Table 2:

Algorithm to perform prediction on weather data using selected classifiers

Start

Input*: dataset weather_data*

Output: *model_scores[ ], model_confusion[ ], model_classification[ ],*

Begin

label[1,2]=["precip_type", "summary"]; //setting precip_type and summary as the target variables for each instance of test.

df_temp = weather_data // duplicate a copy of the weather dataset.

df_temp= Preprocess (df_temp) //perform data cleaning on the duplicated dataset and convert categorical data to numerical data.

for i = 1 to 2{

//Begin loop for each test

y= label[i] //target variable i.e label

X= df_temp – y // removing the target variable from the processed data so we are left with the remaining features.

//Creating subsets of training and testing data from X and y using the 80:20 split ratio, 80% for training and 20% for testing.

X_train = Split (X,0.8)

X_test = Split(X,0.2)

y_train = Split(y, 0.8)

y_test = Split(y,0.2)

models ["name", "model"] = ["RFC: RandomForestClassifier( )", "DTC: DecisionTreeClassifier( )", "ETC: ExtraTreeClassifier( )"] //each instance of the models is stored in the respective label RFC, DTC and ETC

for each models {

Train & Test (model) //Train the models and predict the target variable

Print model_scores[name] //generates accuracy scores

Print model_confusion[name] //generates confusion matrix

Print model_classification[name] //generates classification report

}

}

End

Figure 1: Model Implementation

Table 2: *Parameters used in the algorithm to perform prediction on a weather dataset using selected classifiers*

Parameter	Definition
weather_data	Dataset.
model_scores	stores the accuracy scores of each classifier.
model_confusion	stores the confusion matrix data of each classifier.
model_classification	stores the classification reports of each classifier.
df_temp	Duplicated copy of dataset to work on.
Label	An array storing the target variable for each test
I	Counter for each round of test.
X	The list of features for prediction.
Y	The label (target variable) for prediction.
X_train	80% of the dataset of variable X used for training.
y_train	80% of the dataset of variable y used for training.
X_test	20% of the dataset of variable X used for testing.
y_test	20% of the dataset of variable y used for testing.
Models	Array storing the predefined programming language functions of each classifier.
Name	The Classifier label in the array
Model	Each classifier assigned to "name"

Dataset

The Dataset (weather data) for this research was obtained from Kaggle https://www.kaggle.com/datasets/muthuj7/weather-dataset. The dataset spans the years 2006 to 2016. The features of the dataset include: datetime, summary, precipitation type, temperature, apparent temperature, humidity, windspeed, wind-bearing, visibility, loud cover, pressure and daily summary. The dataset contains 19188 records with 12 features.

Data Preprocessing

Whenever data is gathered from several sources, analysis is not practicable because the data was collected in raw form. Data preprocessing is essential in machine learning projects if better results are to be obtained from the model that is being used. Data in such a format is necessary for a particular machine learning model. For instance, because the Random Forest technique does not accept null values, null values from the original raw dataset must be handled before the algorithm can be used.

Pre-processing is the most crucial step in the machine learning work cycle since data provides the foundation of a strong model and an accurate prediction over time. By using variable Python routines, all the blank spaces, illogical entries, and erroneous data were eliminated.

Pseudocode for Data Preprocessing (Data Cleaning)

Data pre-processing is the most crucial step in the machine learning work cycle since data provides the foundation of a strong model and an accurate prediction over time. It is essential in machine learning projects to obtain better results from the algorithm being used. This is because raw data are usually "noisy", containing missing values or inconsistent data in some instances. For instance, because the Random Forest technique does not accept null values, null values from the original raw dataset must be handled before the algorithm can be used. The steps utilised for data preprocessing in this study include:

1. Remove features which do not have any impact on the prediction.

(The value of ***"Loud Cover"*** *feature is the same for all rows, so, it was removed. Also,* ***"daily summary"*** *feature is a long description of the "****Summary****" feature and cannot be used within the algorithms, so it was removed as well. They were all removed using drop() function in Python.*

2. Remove Null Values. *dropna() function in Python was used*

3. Remove blank spaces. *dropna() function in Python was used*

4. Convert Categorical data into numerical.

(The ***"precipitation type"*** *and* ***"summary"*** *features were converted into numerical values). The LabelEncoder class was used from sklearn. preprocessing library in Python*

The assigned values for each label after converting

the "precipitation type (Precip Type)" and "summary" features is shown in Tables 3 and 4, respectively.

Table 3: *Numerical conversion of the* ***"precip type"*** *feature*

Label	Converted to:
Rain	0
Snow	1

Table 3 shows the assigned values for the categorical labels "rain" and "snow", which were converted to 0 and 1 numerical values, respectively.

Table 4: *Numerical conversion of the* ***"summary"*** *feature*

Label	Converted to:
Breezy	0
Breezy and Dry	1
Breezy and Foggy	2
Breezy and Mostly Cloudy	3
Breezy and Overcast	4
Breezy and Partly Cloudy	5
Clear	6
Dangerously Windy and Partly Cloudy	7
Drizzle	8
Dry	9
Dry and Mostly Cloudy	10
Dry and Partly Cloudy	11
Foggy	12
Humid and Mostly Cloudy	13
Humid and Overcast	14
Humid and Partly Cloudy	15
Light Rain	16
Mostly Cloudy	17
Overcast	18
Partly Cloudy	19
Rain	20
Windy	21
Windy and Dry	22
Windy and Foggy	23
Windy and Mostly Cloudy	24
Windy and Overcast	25
Windy and Partly Cloudy	26

Table 4 shows the assigned values for the summary label after conversion to numerical values. The values ranged between 0 and 26, representing values such as "breezy", "breezy and dry", "clear", "dry" and so on.

Feature Engineering

The act of choosing, modifying, and converting unprocessed data into features that can be applied in supervised learning is known as feature engineering. A machine learning technique called feature engineering uses data to generate new variables that are not present in the training set. To streamline and accelerate data transformations while also improving model accuracy, it can generate new features for both supervised and unsupervised learning. In order to remove unneeded features, new features were added, and correlation was examined.

New Features added to the dataset

The feature "Formatted Date" is used to generate four new features which are "hour_sin", "hour_cos", "day_sin", and "day_cos". The sine and cosine functions are used for both hour and day.

Sine and cosine functions are used in machine learning to encode cyclic or periodic patterns in datetime data. By transforming datetime values using sine and cosine functions, we can create continuous representations that capture the periodic nature of phenomena such as time of day, day of the week, and seasonality.

Data Splitting

The data was split into features and class label; X: feature, y: label. The ***"Precip Type"*** was initially used as the target variable for the prediction. At this time, it was difficult to select the best model based on the accuracy score since "***Precip Type"*** only has 2 kinds of data in it, which is *rain* and *snow*. Then, the "***Summary"*** feature was selected as the target variable to get a more precise accuracy score.

Evaluation Metrics

The evaluation metrics used in this study are Accuracy Score, confusion matrix and classification report (which includes precision, recall and F1-score). Table 4 depicts the formula for computing accuracy, precision, recall and F1-score.

Table 5: *Evaluation Metrics*

Measure	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1-Score	$\frac{2 \times Precision \times Recall}{Precision + Recall}$

Table 5 depicts the formulas for evaluating the accuracy, precision, recall and F1-score of the models implemented in this study, where:

• **TP (True Positives):** The number of instances correctly predicted as positive by the model.

• **FN (False Negatives):** The number of instances that were truly positive but incorrectly predicted as negative by the model.

• **FP (False Positives):** The number of instances incorrectly predicted as positive

4. Result and Discussion

This section presents the key findings of the research, highlighting the model's performance across the different evaluation metrics. It also provides an in-depth discussion of the results, giving insights into the implications of the findings.

In evaluating the models, two instances were considered: First, "***Precip Type"*** was used as the target variable and the other instance used "***Summary"*** as the target variable. The results and discussion after training and testing the models are presented in the following sections.

Model Evaluation based on Accuracy

The accuracy score represents the proportion of correct predictions made by the model and is expressed as a percentage. The higher the value, the better the performance of the model.

Accuracy Score when using "precip type" for prediction

The accuracy scores of the models generated using "***Precip type"*** as the target variable are presented in Table 5.

Table 6*: Accuracy scores using "Precip Type" as the target variable.*

Classifier	Accuracy
RFC	0.997134
DTC	0.997134
ETC	0.994424

Table 6 shows that Random Forest Classifier and Decision Tree Classifiers had accuracy scores of 99.7%, while Extra Tree Classifier had 99.4% accuracy score. This implies that all three models accurately predicted the Precip Type, which could have been due to having only two class labels: rain and snow.

Accuracy Score using "summary" for prediction

The accuracy scores of the models generated using "***summary"*** as the target variable are presented in Table 7.

Table 7: *Accuracy Score of the models using Summary as the target variable.*

Classifier	Accuracy
RFC	0.660465
DTC	0.538618
ETC	0.647801

Table 7 shows RFC with an accuracy score of 66% which indicates that the model correctly predicted the weather 66 out of 100 times. DTC has an accuracy score of 53.9% while ETC has a 64.8% accuracy score. From this, it clearly shows that Random Forest Classifier has the best accuracy score for the prediction using summary as target variable.

Comparison of Accuracy Scores

Comparison of the accuracy of the models based on "***Precip Type***" and "***summary***" as target variables is presented in Figure 2.

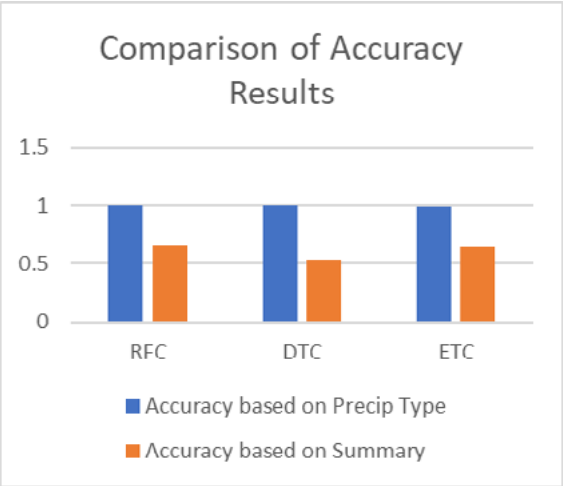


Figure 2: Comparison of Accuracy Scores

Figure 2 shows a comparison of the accuracy of the models when ***"precip type"*** and ***"summary"*** were used for prediction. It was revealed that the accuracy scores when "Precip Type" was used as target variable consistently performed better than when "summary" was used for all three models. This may have been because of the fewer labels for the "Precip Type" (rain and snow). Nonetheless, the accuracy scores based on "summary" performed above average in all three models, which indicates that for more than 50% of the time, the models will accurately predict the weather.

Model Evaluation based on Confusion Matrix

The confusion matrix is a matrix that evaluates the performance of a machine learning model by showing counts of correct and incorrect predictions for each class label. This study also plotted the confusion matrix on a chart for easy representation.

Confusion Matrix using "precip type" for prediction

The confusion matrix for RFC, DTC, and ETC using "Precip Type" as the target is depicted in Figures 3 (a), (b) and (c), respectively.

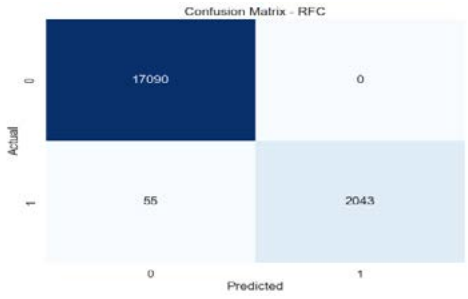


Figure 3(a): *Confusion Matrix for RFC using "Precip Type" as Target variable.*

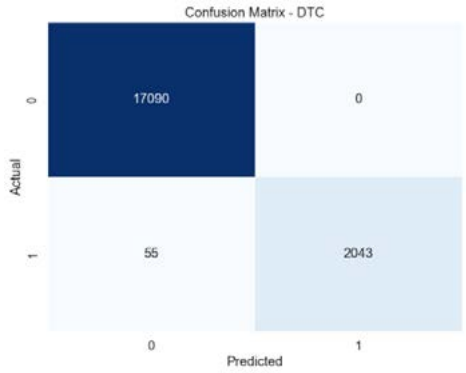


Figure 4: *Confusion Matrix for Decision Tree Classifier using "Precip Type" as the Target variable.*

Figure 3 (a) and (b) shows the confusion matrix for RFC and DTC which revealed that out of the total test record 19188, both models predicted 17090 instances correctly for when the actual value is 0 with no wrong predictions. Also, out of the total test record, both models predicted 2043 records correctly when the actual value is 1 and 55 records incorrectly.

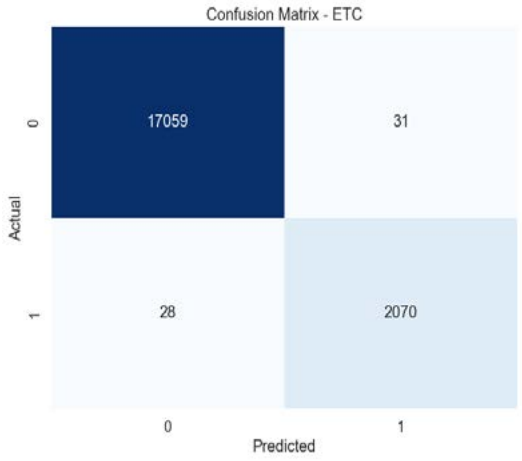


Figure 3(c): *Confusion Matrix for ETC using "Precip Type" as the Target variable.*

Figure 3(c) shows that out of the total test record of 19188, the model predicted 17059 records correctly when the actual value is 0 and predicted 31 records wrongly. Also, out of the total test records, it predicted 2070 records correctly when the actual value is 1 and 28 records incorrectly.

Confusion Matrix using "summary" for prediction

The confusion matrix for RFC, DTC and ETC using "summary" as the target are depicted in Figures 5 (a), 6 (a) and 7 (a), respectively while the percentage distributions of correct and incorrect predictions for each class label are shown in Figures 5 (b), 6(b) and 7 (c) for RFC, DTC and ETC, respectively using bar charts.

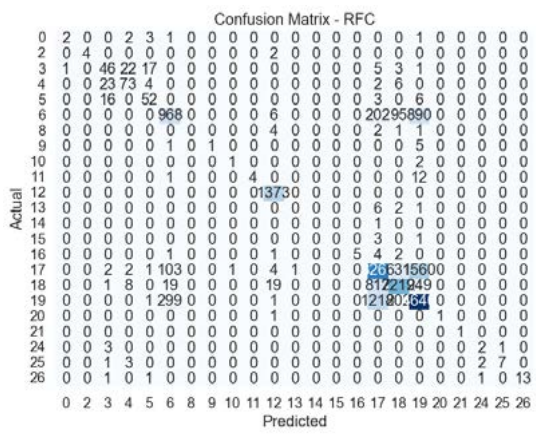


Figure 5 (a): *Confusion Matrix for RFC using "Summary" as the Target variable.*

Figure 5(a) shows the confusion matrix for RFC while the summary of correct and incorrect predictions is depicted on the bar chart in Figure 5(b)

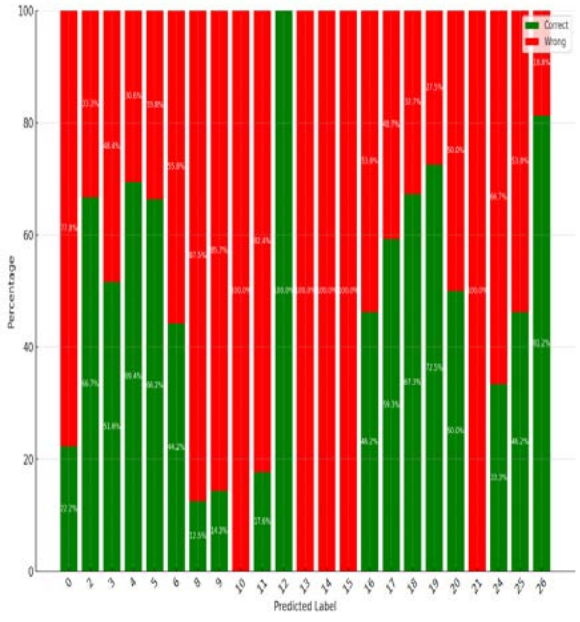


Figure 5 (b): Bar chart showing the percentage of Correct vs incorrect predictions per label based on the confusion matrix for RFC.

In Figure 5(b), the green bars represent the percentage of correct predictions for each label while the red bars show the percentage of incorrect predictions for each label. From the total number of predictions, RFC predicted correctly 12696 times and incorrectly 6492 times, with the model correctly predicting all instances of label 12.

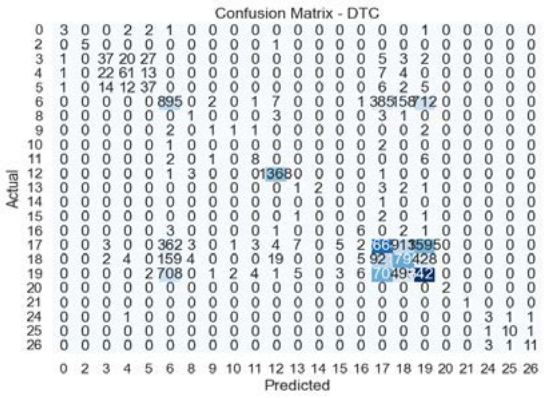


Figure 6 (a) *Confusion Matrix for DTC using "Summary" as the Target variable.*

Figure 6 (a) shows the confusion matrix for DTC, while the summary of correct and incorrect predictions is depicted on the bar chart in Figure 6(b).



Figure 6(b): Bar chart showing the percentage of Correct vs incorrect predictions per label based on the confusion matrix for DTC.

In Figure 6(b), the green bars represent the percentage of correct predictions for each label, while the red bars show the percentage of incorrect predictions for each label of a model. From the total number of predictions, 19188, DTC predicted correctly 10291 times and wrongly 8897 times, with the model correctly predicting label 21, which had just one record.

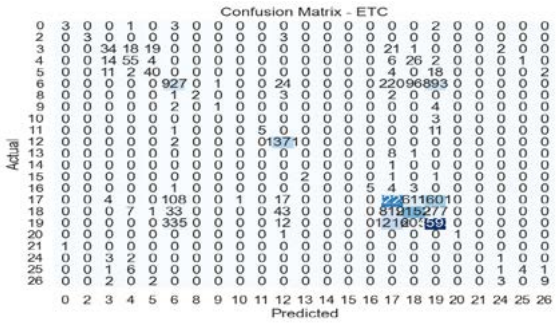


Figure 7 (a): *Confusion Matrix ETC using "Summary" as Target variable.*

Figure 7 (a) shows the confusion matrix for ETC while the summary of correct and incorrect predictions is depicted on the bar chart in Figure 7(b).

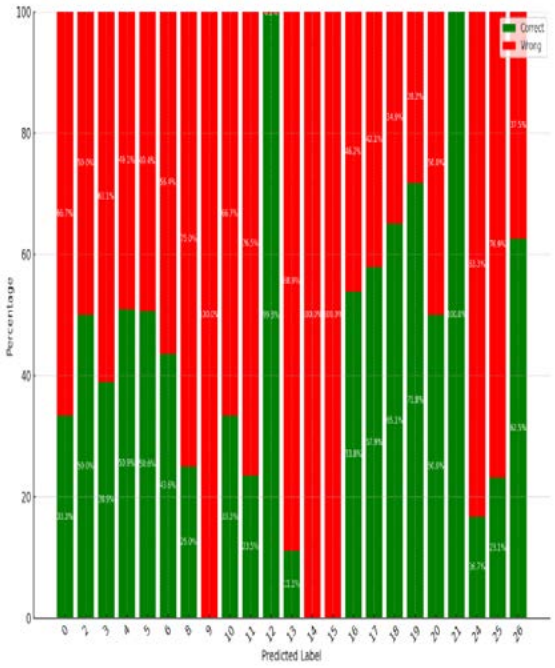


Figure 7(b): *Bar chart showing the percentage of Correct vs incorrect predictions per label based on the confusion matrix for DTC.*

In Figure 7(b), the green bars represent the percentage of correct predictions for each label, while the red bars show the percentage of incorrect predictions for each label of a model. From the total number of predictions, 19188, DTC predicted correctly 12436 times and wrongly 6752 times, with the model correctly predicting label 21, which had just one record.

Model Evaluation based on Classification Reports

In this study, the classification report gives a detailed description of a model's performance in terms of Precision, Recall and F1-Scores.

Classification Report using "precip type" for prediction

The Classification report of the three models: RFC, DTC and ETC, using "precip type" as target variable for the labels "rain" and "snow" is presented in Table 8.

Table 8: *Classification Report using "precip type" as the target variable*

Predicted Label	Metric	RFC	DTC	ETC
0	Precision	0.996792	0.996792	**0.998361**
	Recall	**1.000000**	**1.000000**	0.998186
	F1-Score	**0.998393**	**0.998393**	0.998274
1	Precision	**1.000000**	**1.000000**	0.985245
	Recall	0.973785	0.973785	**0.986654**
	F1-Score	**0.986718**	**0.986718**	0.985949

Table 8 shows the classification report of the three models using "Precip Type" as target variable based on the precision, recall and F1-score metrics. The predicted label 0 (rain) had the best precision of 99.836% when predicted using ETC, which indicates that 99.836% of the predicted values were correct, while RFC and DTC achieved the best precision of 100% respectively for the predicted label 1 (snow). Also, RFC achieved the best recall of 100% for the predicted label 0, which indicates that the model identified all instances correctly with no false negatives, while ETC achieved the best recall of 99.665% for the predicted label 1. The best F1-scores of 99.839% and 98.672% were achieved by RFC and DTC, respectively, for the predicted labels 0 and 1, which further emphasises the effectiveness of the model in predicting precipitation type. Consequently, RFC and DTC had similar performances while ETC came behind the two classifiers, except in the precision value for label 0.

Classification Report using "Summary" for prediction

The classification reports of the three models using summary as the target variable for all the predicted labels (0-26) is presented in Table 9.

Table 9: *Classification Report using "summary" as the target variable*

Predicted Label		RFC	DTC	ETC
0	Precision	0.666667	0.500000	**1.000000**
	Recall	0.222222	**0.333333**	**0.333333**
	F1-Score	0.333333	0.400000	**0.500000**
2	Precision	**1.000000**	**1.000000**	0.750000
	Recall	0.666667	**0.833333**	0.500000
	F1-Score	0.800000	**0.909091**	0.600000
3	Precision	0.521277	0.462500	**0.528571**
	Recall	**0.515789**	0.389474	0.389474
	F1-Score	**0.518519**	0.422857	0.448485
4	Precision	**0.681818**	0.645833	0.617978
	Recall	**0.694444**	0.574074	0.509259
	F1-Score	**0.688073**	0.607843	0.558376
5	Precision	**0.708333**	0.477273	0.619048
	Recall	**0.662338**	0.545455	0.506494
	F1-Score	**0.684564**	0.509091	0.557143
6	Precision	**0.690029**	0.403707	0.656446
	Recall	**0.441925**	0.413235	0.435909
	F1-Score	**0.538787**	0.408415	0.523915
8	Precision	**1.000000**	0.200000	**1.000000**
	Recall	0.125000	**0.250000**	**0.250000**
	F1-Score	0.222222	0.222222	**0.400000**
9	Precision	**1.000000**	0.142857	**1.000000**
	Recall	**0.142857**	**0.142857**	0.000000
	F1-Score	**0.250000**	0.142857	0.000000
10	Precision	0.000000	0.250000	**0.500000**
	Recall	0.000000	**0.333333**	**0.333333**
	F1-Score	**1.000000**	0.285714	0.400000
11	Precision	0.600000	0.470588	**0.800000**
	Recall	0.176471	**0.470588**	0.235294
	F1-Score	0.272727	**0.470588**	0.363636
12	Precision	0.972380	**0.973684**	0.925051
	Recall	**1.000000**	0.997087	0.997815
	F1-Score	**0.985996**	0.985246	0.960056
13	Precision	0.000000	0.000000	**0.250000**
	Recall	0.000000	0.000000	**0.111111**
	F1-Score	**1.000000**	**1.000000**	0.153846
14	Precision	**1.000000**	0.000000	**1.000000**
	Recall	0.000000	0.000000	0.000000
	F1-Score	0.000000	**1.000000**	0.000000
15	Precision	**1.000000**	0.000000	**1.000000**
	Recall	0.000000	0.000000	0.000000
	F1-Score	0.000000	**1.000000**	0.000000
16	Precision	**1.000000**	0.352941	**1.000000**
	Recall	0.461538	0.461538	**0.538462**
	F1-Score	0.631579	0.400000	**0.700000**
17	Precision	**0.590340**	0.470831	0.581769
	Recall	**0.592885**		0.579051
	F1-Score	**0.591610**	0.471254	0.580407
18	Precision	**0.702787**	0.518227	0.696882
	Recall	**0.673469**	0.533313	0.650660
	F1-Score	**0.687816**	0.525662	0.672978
19	Precision	**0.633155**	0.556419	0.621511
	Recall	**0.724729**	0.540324	0.717654
	F1-Score	**0.675854**	0.548253	0.666132
20	Precision	**1.000000**	0.250000	**1.000000**
	Recall	**0.500000**	**0.500000**	**0.500000**
	F1-Score	**0.666667**	0.333333	**0.666667**
21	Precision	**1.000000**	**1.000000**	**1.000000**
	Recall	0.000000	**1.000000**	**1.000000**
	F1-Score	0.000000	**1.000000**	**1.000000**
24	Precision	0.285714	**0.300000**	0.166667
	Recall	0.333333	**0.500000**	0.166667
	F1-Score	0.307692	**0.375000**	0.166667
25	Precision	**0.857143**	0.818182	0.600000
	Recall	0.461538	**0.692308**	0.230769
	F1-Score	0.600000	**0.750000**	0.333333
26	Precision	**0.928571**	0.846154	0.769231
	Recall	**0.812500**	0.687500	0.625000
	F1-Score	**0.866667**	0.758621	0.689655

The classification report depicted in Table 9 shows the precision, recall and F1-scores of the three models using summary as the target variable, where 0 means "Breezy", 1 means "Breezy & Dry" and so on. Using the predicted label 0 as an instance, the first row in Table 10 means the three algorithms are trying to predict the weather as "Breezy". During this prediction, Random Forest Algorithm achieved a precision of 66.7%, recall of 22.2% and F1-score of 33.3%. Decision Tree Classifier obtained a precision of 50%, recall of 33.3% and F1-score of 4%, while the Extra Tree classifier achieved a precision of 100%, recall of 33.3% and F1-score of 50%. During the prediction of the label "Breezy", the Extra Tree classifier performed better. The summary of the number of times each model achieved the best prediction scores for the metrics under consideration is presented in Table 12. The F1-score is used as a determinant of the best-performing model since it represents the harmonic mean of the precision and recall metrics, ensuring the effectiveness of the prediction.

For example, predicted label 0 means "Breezy", 1 means "Breezy & Dry", etc. Now, using predicted label 0 as an instance, the first row in Table 10 means the three algorithms are trying to predict the weather as "Breezy". During this prediction, Random Forest Algorithm has a precision of 66.7%, recall of 22.2% and F1-score of 33.3%. Decision Tree Classifier has a precision of 50%, recall of 33.3% and F1-score of 4% while the Extra Tree classifier has a precision of 100%, recall of 33.3% and F1-score of 50%. During the prediction of the label "Breezy", the Extra Tree classifier performed better.

Table 10 shows the number of times each algorithm predicted better to get the overall best of the three algorithms. Here, the F1-score will be considered as the decision factor since it comprises of both precision and recall.

Table 10: *Number of times each classifier predicts better.*

Predicted Label	RFC	DTC	ETC
0			✓
2		✓	
3	✓		
4	✓		
5	✓		
6	✓		
8			✓
9	✓		
10	✓		
11		✓	
12	✓		
13	✓	✓	
14		✓	
15		✓	
16			✓
17	✓		
18	✓		
19	✓		
20	✓		✓

21		✓	✓
24		✓	
25		✓	
26	✓		
Total	**11**	**6**	**3**

From Table 10, it is shown that RFC performed better at predicting 11 labels. DTC performed well with 6 labels, while ETC performed well at predicting 3 labels. However, the result is inconclusive when 3 labels were predicted.

Therefore, we can conclude that Random Forest Classifier has the best overall performance among the three models since it has the highest number of times it performed better.

Threats to Validity

At the end of the study, it was discovered that the results of the experiments performed are dependent on the number of possible outcomes of the target variables. As was shown when "Precip Type" was used for prediction compared to the use of "summary", much higher accuracy results were obtained.

5. Conclusion and Recommendation

In the realm of machine learning, weather prediction is a key research subject. There are still many mysteries to be solved and many fresh algorithms to be employed on weather data using precipitation type and weather summary. With the prediction of precipitation type, the accuracy scores for both random forest and decision tree were 99.71%, while extra tree classifiers achieved 99.44% accuracy. The confusion matrix shows better prediction of records by random forest and decision tree Classifiers with score of 99.71% each, while extra tree classifier performed the least, predicting 99.69% records correctly. Also, the Classification reports revealed that random forest had a precision of 99.84%, recall of 98.69% and F1-score of 99.26%. The decision tree classifier had a precision of 99.84%, recall of 98.69% and F1-score of 99.26%, while extra tree classifier had a precision of 99.18%, recall of 99.24% and F1-score of 99.21%. With the second set of predictions using weather summary, random forest had an accuracy score of 66.05%, decision tree had 53.9% and the extra tree had 64.8% accuracy scores. As for the confusion matrix, random forest performed best with 66.17% records correctly predicted, extra tree classifier with 64.8% correct records and decision tree with 53.6% records. The Classification report showed random forest had a precision of 69.6%, recall of 47.8% and F1-score of 52.2%, decision tree achieved a precision of 17.39%, recall of 39.13% and F1-score of 30.43%, while extra tree had a precision of 52.17%, recall of 30.43% and F1-score of 21.74%. In conclusion, Random Forest Classifier has proven to consistently outperform the other classifiers in this study. This research work can be reviewed in the future to make the algorithms predict future weather occurrences. Also, other machine learning algorithms and ensemble methods can be tested to verify the claims from the study.

References

AbdulRaheem, M., Awotunde, J. B., Abidemi, E. A., Idowu, D. O., & Adekola, S. O. (2022). Weather prediction performance evaluation on selected machine learning algorithms. *IAES International Journal of Artificial Intelligence (IJ-AI), 11(4),* 1535-1544.

Abhishek, K., Singha, M. P., Ghosh, S., & Anand, A. (2012). Weather forecasting model using artificial neural network. *Procedia Technology*, 4, 311-318

Abrahamsen, E. B., Brastein, O. M., and Lie, B. (2018). Machine Learning in Python for Weather Forecast based on Freely Available Weather Data. *In Proceedings of The 59th Conference on Simulation and Modelling (SIMS 59), 26-28 September 2018, Oslo Metropolitan University, Norway. Linköping Electronic Conference Proceedings 153:24, p. 169-176.* https://doi.org/10.3384/ecp18153169

Abrar, M., Sim, A. T. H., Shah, D., & Khusro, S.

(2014). Weather Prediction Using Classification. *Sci.Int. (Lahore),26(5),2217-2223.*

Ajina, A., Jaya, C. K. G, Bhat, D. N., Saxena, K. (2023). Prediction of weather forecasting using artificial neural networks. *Journal of Applied Research and Technology 21 (2) 205-211*.

ArcGIS PRO(Nd). https://pro.arcgis.com/en/pro-app/latest/tool-reference/geoai/how-extra-tree-classification-and-regression-works.htm

Iseh. A. J., Woma. T. Y. (2013). Weather Forecasting Models, Methods and Applications. *International Journal of Engineering Research & Technology (IJERT),* 2(12), 1945- 1956.

Jayasingh, S. K., & Mantri, J. K. (2020). Weather Predictions using Support Vector Machine and Rough Set. *International Journal of Engineering and Advanced Technology*, 9(2), 5513-5516

Jyothi, M. J. N. Manepalli (2020). Weather Prediction Through Machine Learning. *Complexity International Journal (CIJ),* 24(1).

Kashikar, S., Patil, S., Vedantwar, A., Katpatal, S., & Pillai, S. (2019). Weather Prediction using Scikit-Learn. *International Journal of Computer Science and Engineering,* 7(4), 26-40.

Krishna, G. V. (2015). An Integrated Approach for Weather Forecasting based on Data Mining and Forecasting Analysis. *International Journal of Computer Applications,* 120(11)26-29

Muthukumar, P., Cocom, E., Kabir N., Comer, B., Burga, I., Taub, J., Calvert, C. F., Holm, J., & Pourhomayoun, M. (2021). Predicting PM2.5 atmospheric air pollution using deep learning with meteorological data and ground-based observations and remote-sensing satellite big data. *Air Quality, Atmosphere and Health,* 1-14.

Omole, A. (2022). "Flood Killed 24, displaced 471000 persons, Kogi Govt". *Leadership Newspaper.* Available at https://leadership.ng/flood-killed-24-displaced-471000-persons-kogi-govt/

Sujatha, N., Lokesh, V. S., Valli, L. N., & Mech, M. (2024). Weather prediction using machine learning algorithms. *Journal of Emerging Technologies and Innovative Research (JETIR)* 11(1), 550-555

Wu, S. F., Zhu, J., & Wang, Y. (2012). Weather Forecasting Using Naïve Bayesian. *Advances in Future Computer and Control Systems:* 1, 337–341.

Xhabafti, M., Vika, B., & Sinaj, V. (2024). A Comparative Study of Statistical and Deep Learning Model-Base weather Prediction in Albania. *WSEAS Transactions on Computer Research, 12, 151-160*