**EMPIRICAL SOFTWARE ENGINEERING**

**Laboratory File**

**Subject Code:  SE-302a**

A picture containing logo

Description automatically generated

**TOPIC**

RAINFALL PREDICTION

**Submitted To:**

Dept of Software Engineering, DTU

**Submitted By:**

A (2K20/SE/A)

B (2K20/SE/B)

C (2K20/SE/C)

**DELHI TECHNOLOGICAL UNIVERSITY**

**(Formerly DCE)**

**Main Bawana Road, Delhi-110042**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **S No** | **EXPERIMENT** | **DATE** | **SIGN** |
| **1.** | Collection of Empirical Studies | 23/01/2023 |  |
| **2.** | Identify Research gaps. Collect Data sets from open-source repositories. | 06/02/2023 |  |
| **3.** | Write a Program to perform Exploratory analysis of the dataset. | 13/02/2023 |  |
| **4.** | Write a Program to perform various feature techniques for collected dataset. | 20/02/2023 |  |
| **5.** | Develop a Machine learning Model for selected topics. | 27/02/2023 |  |
| **6.** | State the hypothesis for rainfall prediction using logistic regression | 13/02/2023 |  |

**EXPERIMENT – 1**

**Aim**: Collection of Empirical Studies for “Rainfall Prediction”.

The collection of empirical studies in Empirical Software Engineering is important for advancing the field, improving software development, validating theories, providing a better understanding of software development, and supporting evidence-based decision-making. Empirical studies provide evidence-based insights and understanding of software development processes, practices, and techniques, leading to better decision-making, reduced risks, and improved software quality.

For the topic “Rainfall Prediction”, we have selected 9 research papers belonging to various conferences and journals around the world.

A tabular list of these Papers with their conference/journal name is given below:

|  |  |  |
| --- | --- | --- |
| S No | Research Paper Name | Conference/Journal Name |
| 1. | Rainfall Prediction Using Machine Learning & Deep Learning Techniques | [2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)](https://ieeexplore.ieee.org/xpl/conhome/9145513/proceeding) |
| 2. | Rainfall Prediction using Machine Learning and Neural Network | International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878 (Online) |
| 3. | Rainfall Prediction using Machine Learning | International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; |
| 4. | Efficient Rainfall Prediction and Analysis using Machine Learning Techniques | Turkish Journal of Computer and Mathematics Education |
| 5. | A STUDY ON RAINFALL PREDICTION TECHNIQUES | International Journal of Scientific Research in Engineering and Management (IJSREM) |
| 6. | Rainfall prediction through TRMM dataset using machine learning model | [2020 8th International Conference on Information and Communication Technology (ICoICT)](https://ieeexplore.ieee.org/xpl/conhome/9162155/proceeding) |
| 7. | Rainfall Prediction: A Deep Learning Approach | Springer International Publishing Switzerland 2016 |
| 8. | A Survey of Rainfall Prediction Using Deep Learning | [2021 3rd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)](https://ieeexplore.ieee.org/xpl/conhome/9664652/proceeding) |
| 9. | Rainfall Prediction based on Ensemble Model | International Journal of Innovative Research in Science, Engineering and Technology |

**EXPERIMENT - 2**

**Aim**: Identify Research gaps. Collect Data sets from open-source repositories.

|  |  |  |
| --- | --- | --- |
| S No. | Research Paper Name | Research Gap |
| 1. | Rainfall Prediction Using Machine Learning & Deep Learning Techniques | The results intend that in terms of MSE and RMSE, our proposed architecture outperforms remaining approaches. The accuracy can be measured by the MSE and RMSE comparing with the other models. In circumstances of water resource and management, human being life and the climate they possess, precipitation prediction is of huge importance. Wrong or unfinished estimation issues can be faced because the measurement of precipitation is influenced by spatial and local change and property. |
| 2. | Rainfall Prediction using Machine Learning and Neural Network | The current approach for rainfall prediction fails in most of the complex cases, as it is unable to predict the hidden layers present, which is yet to be recognized for performing the precise prediction |
| 3. | Rainfall Prediction using Machine Learning | Decision Tree classifier gives the least accuracy with 73%. We can further expand this research covering other ML techniques such as time series, clustering and association rules and other ensemble techniques. Taking into consideration the limitations of this study, there is a need to build more complex and combination of models to get higher accuracy for rainfall prediction system. |
| 4. | Efficient Rainfall Prediction and Analysis using Machine Learning Techniques | The existing methods for rainfall forecasting fail in the most complicated situations because they cannot forecast the hidden patterns present, which are yet to be understood to perform an accurate prediction. Further predictions can be carried out by evaluating additional classification methods and climate characteristics on various weather dates |
| 5. | A STUDY ON RAINFALL PREDICTION TECHNIQUES | We prescribe tenderfoots to utilize straight and RBF piece for direct and non-straight relationship individually. We see that SVR is better than MLR as an expectation strategy. MLR can't catch the non-linearity in a data set and SVR winds up helpful in such circumstances.  We prescribe tenderfoots to utilize straight and RBF piece for direct and non-straight relationship individually. We see that SVR is better than MLR as an expectation strategy. MLR can't catch the non-linearity in a data set and SVR winds up helpful in such circumstances. We additionally process |
| 6. | Rainfall prediction through TRMM dataset using machine learning model | [2020 8th International Conference on Information and Communication Technology (ICoICT)](https://ieeexplore.ieee.org/xpl/conhome/9162155/proceeding) |
| 7. | Rainfall Prediction: A Deep Learning Approach | The paper did not provide an explanation of how the deep learning models arrived at their predictions, and there may be a need for further research on interpretable deep learning methods. |
| 8. | A Survey of Rainfall Prediction Using Deep Learning | hyperparameters of the networks  are searched using the trial and error method, and few papers  used optimization algorithms to find the optimal  hyperparameter. Research can be focused in this direction to  implement a hybrid model of optimization algorithms and  deep learning methods  hyperparameters of the networks are searched using the trial and error method, and few papers used optimization algorithms to find the optimal hyperparameter. Research can be focused in this direction to implement a hybrid model of optimization algorithms and deep learning methods. Due lack of paper that consider all the popular deep learning models with similar dataset and metrics, it is difficult to see which model has the best performance |
| 9. | Rainfall Prediction based on Ensemble Model | Research paper could not define that changing the technique yields the result or by increasing the input data set for the same technique results change in the findings. Importance of rainfall prediction cannot be over emphasized |

.

**EXPERIMENT – 3**

**Aim**: Perform Exploratory Data Analysis on the Dataset.

**Overview of the dataset**:

The following dataset is from Location “Albury” from the year 2008, month December, day first i.e. 01-12-2008. Dataset is contained with more than 30 fields which are all taken on different days.

Dataset contains various data field such as Location, Minimum Temperature, Maximum Temperature, Rainfall per centi-meter, Evaporation, Sunshine, Wind, Humidity, Pressure, Cloud, Temperature, Rain Today, Rain Tomorrow, and the Risk Parameter.

Wind readings are divided into WindGustDir, WindGustSpeed. Which is also divided into two parts as follows: WindGustDir at 9am , 3pm. And WindGustSpeed at 9am, 3pm.

Wind Gust is a term used when the maximum speed exceeds the average speed by 10 to 15 knots, the terms gusts is used for departure of 15 to 25 knots etc.

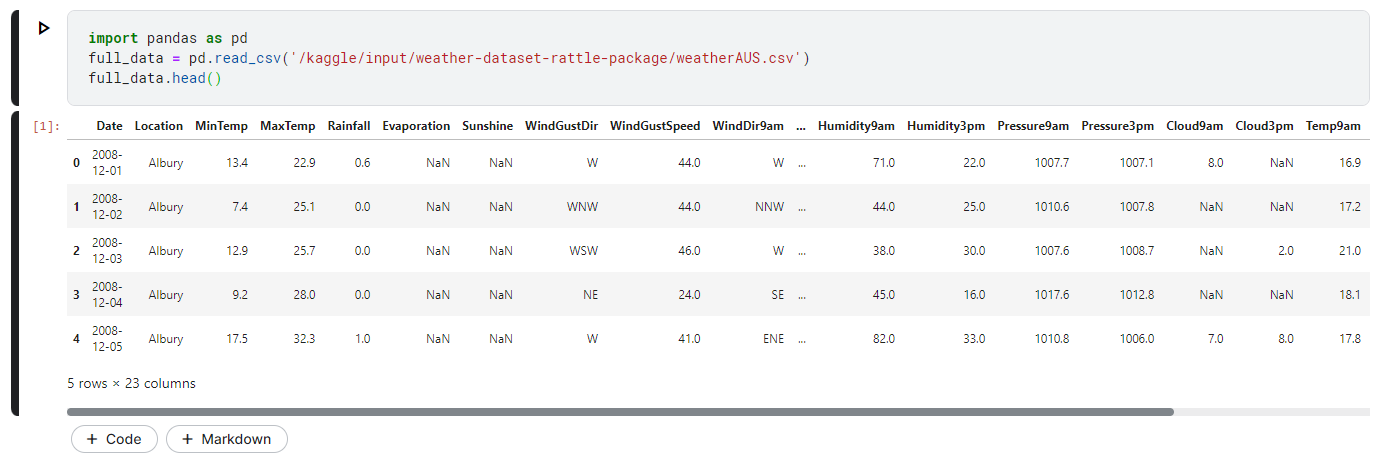
Pressure and Cloud at Different time is calculated such as at 9am, 3pm.

Based on the values of all the above mentioned fields Risk\_MM parameter is determined.

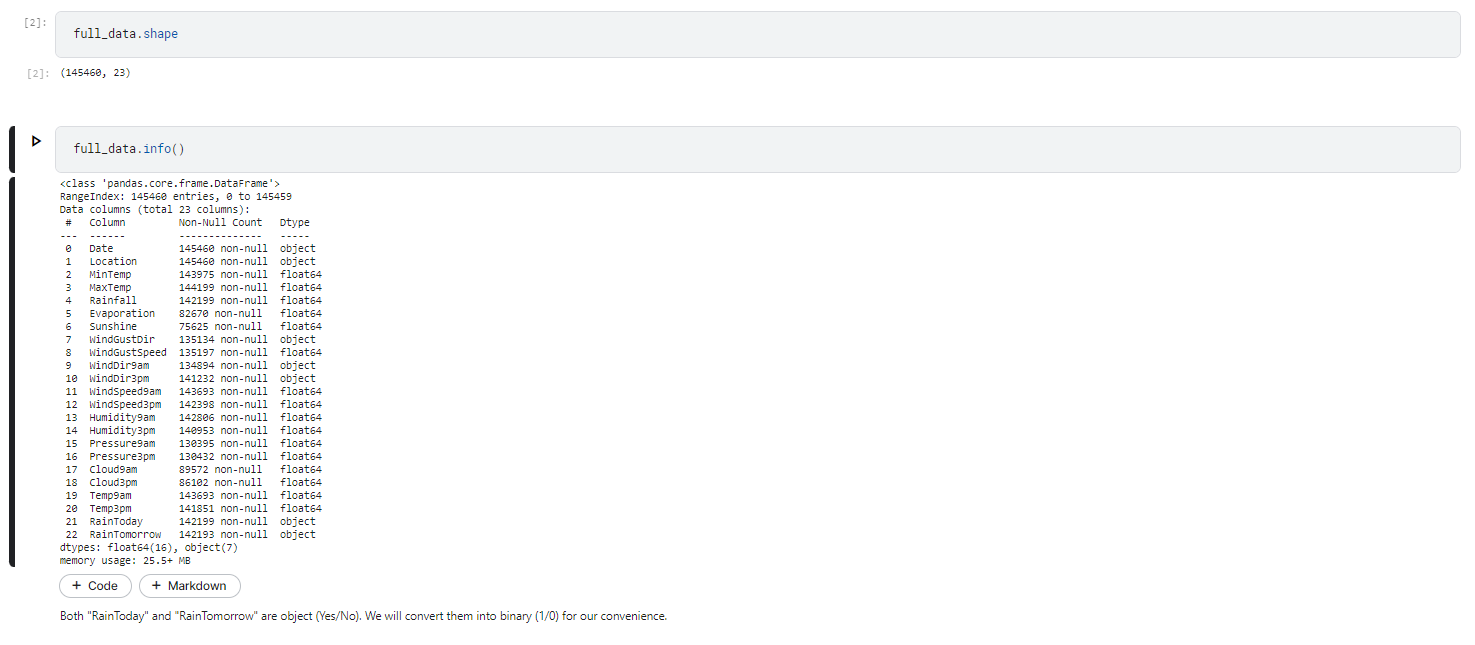
Rain Today and Rain Tomorrow is predicted/determined on the basis of Pressure, Humidity, and Cloud Formation.

Each window of reading is a datapoint of 24 features.

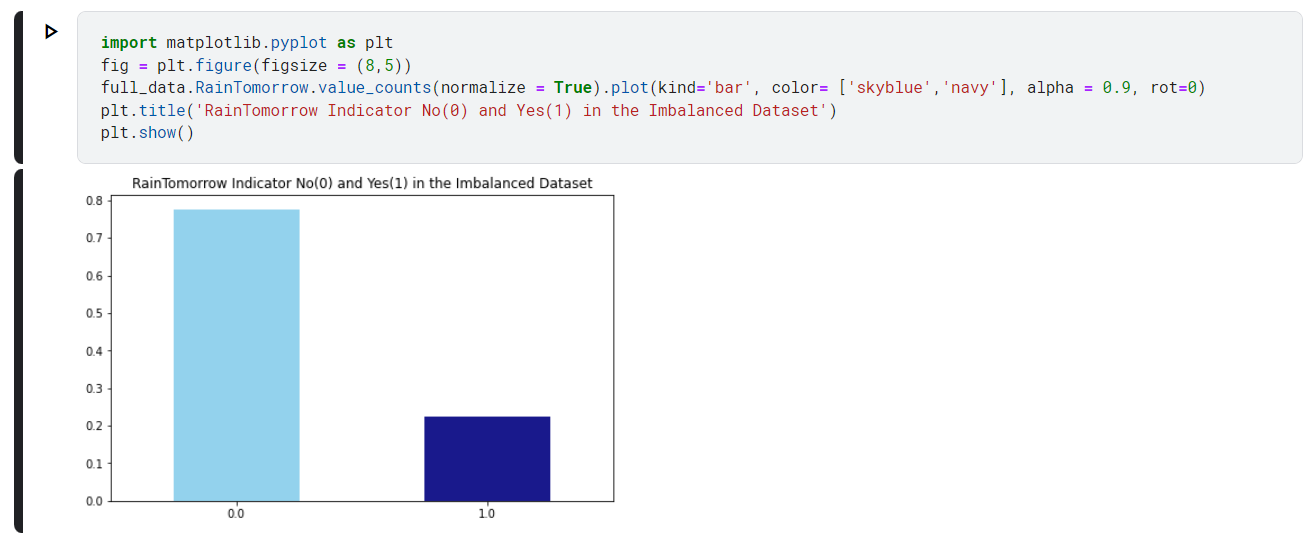
**Obtaining The Train Data.**



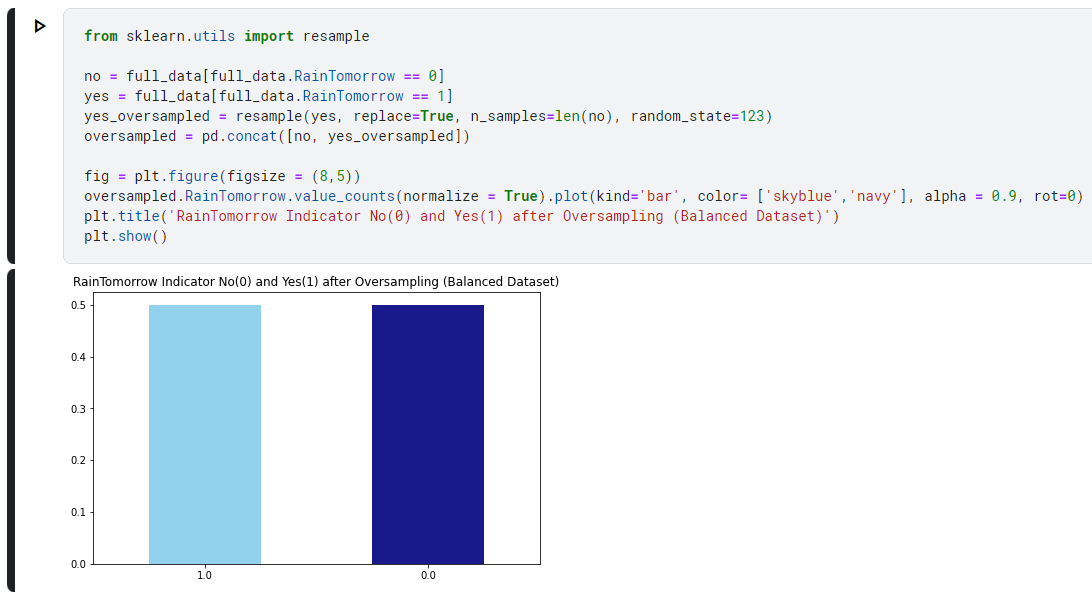
**Data Exploration**: checking the size of data to decide whether it requires any compression of size.

Checking whether the dataset is Imbalanced or Balanced. If the data is imbalanced, converting under sample majority or will oversample minority to balance it. ****

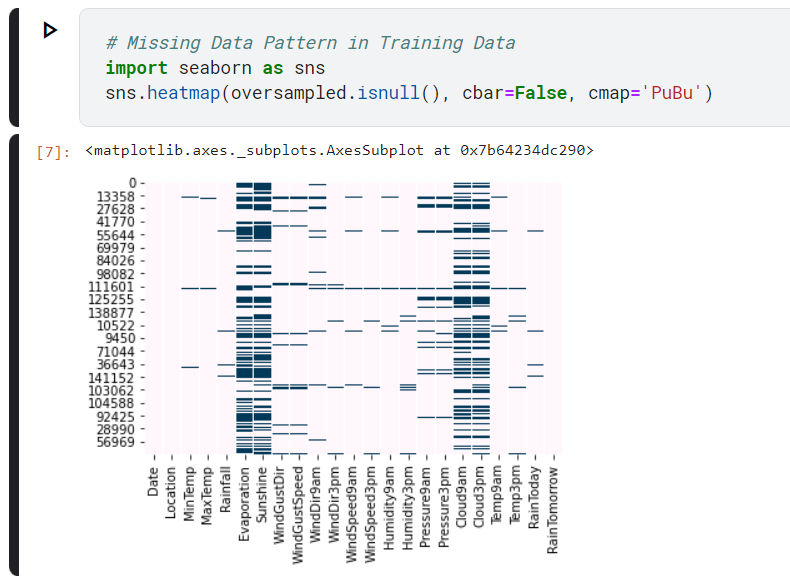
**Observation**: we noticed that presence of "0" and "1" are almost in the ratio 78:22. So there is a class imbalance and we have to handle it. For tackling class imbalance, we will use **oversampling of minority class** here. Since the size of the data set is quite small, under sampling of majority class would not make much sense here.

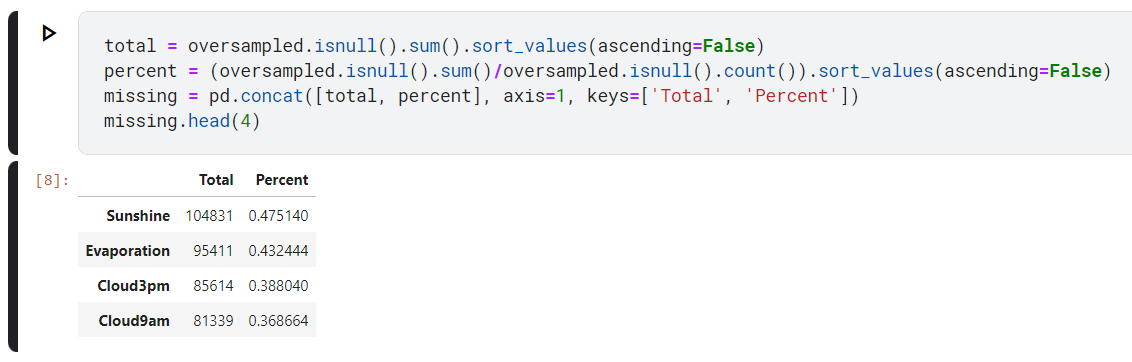


**Handling Class Imbalance**



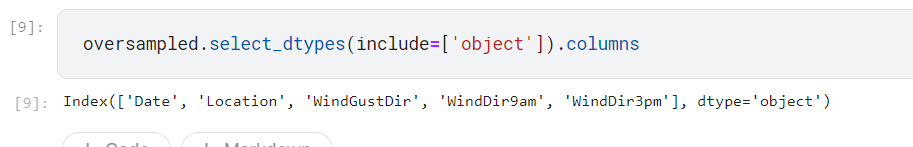
Checking for the missing data pattern in the dataset.



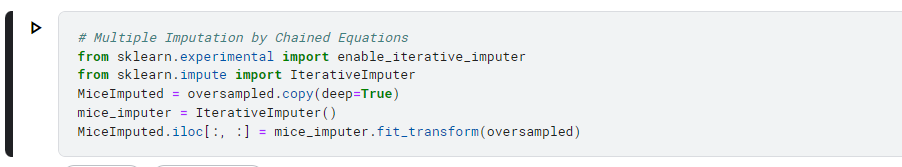


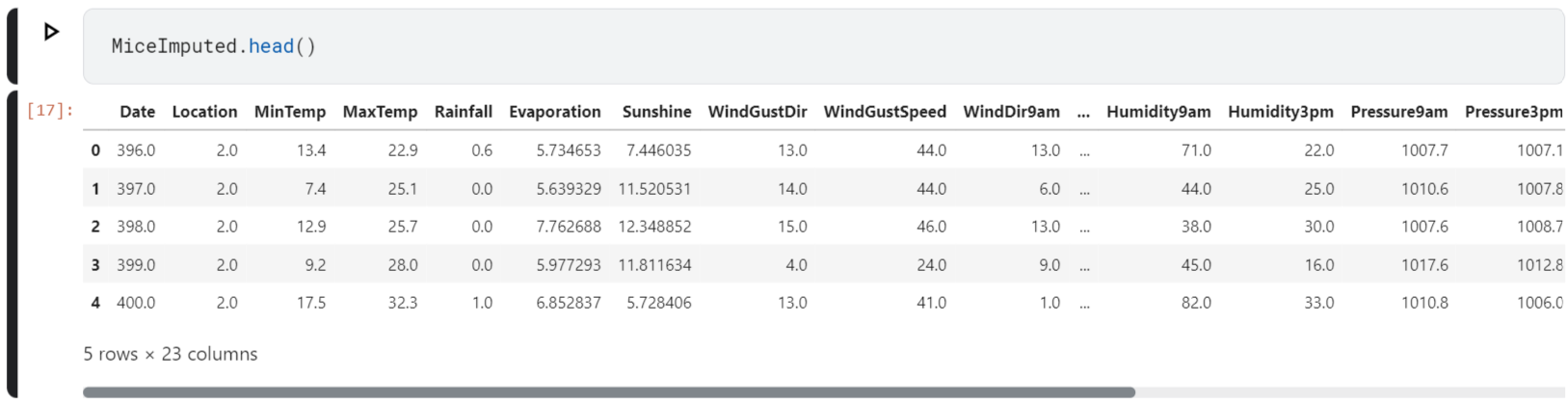
**Observation**: Visibly, 'Evaporation','Sunshine','Cloud9am','Cloud3pm' are the features having high missing percentage.

Imputation and Transformation

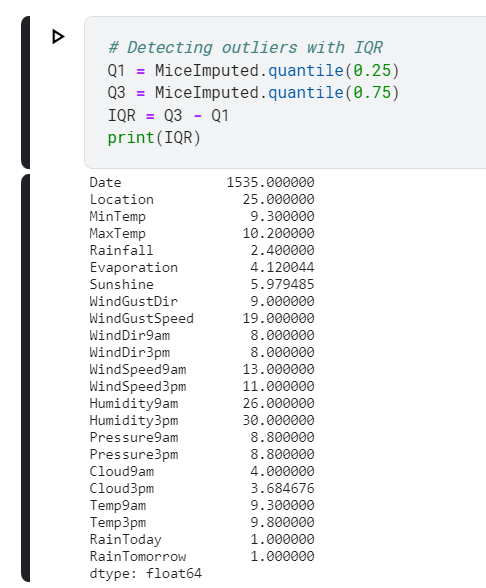
****







Detecting and discarding the outliers from dataset on the Inter-Quartile Range



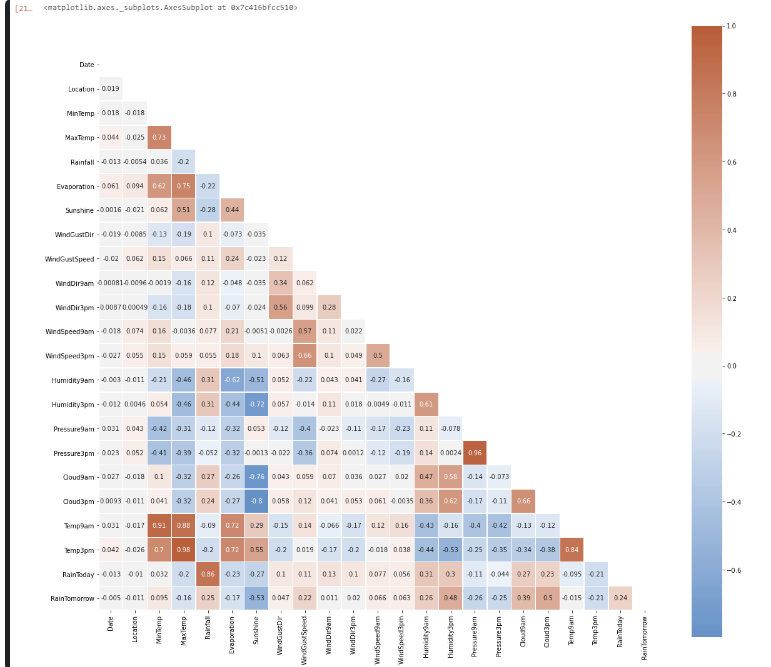


We observe that the original dataset was having the shape (87927, 24). After running outlier-removal code snippet, the dataset is now having the shape (86065, 24).

So, the dataset is now

free of 1862 outliers. We will now check for multi-collinearity i.e. whether any feature is highly correlated with another.





**Our Observation**:

The following pairs of features are having high correlation between them:

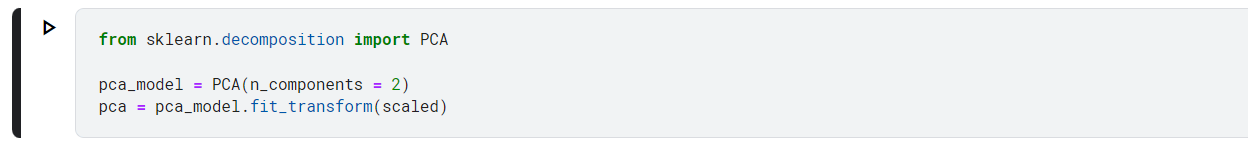
* MaxTemp and MinTemp
* Pressure9am and Pressure3pm
* Temp9am and Temp3pm
* Evaporation and MaxTemp
* MaxTemp and Temp3pm But in no case, the correlation value is equal to a perfect "1". So we are not discarding any feature.

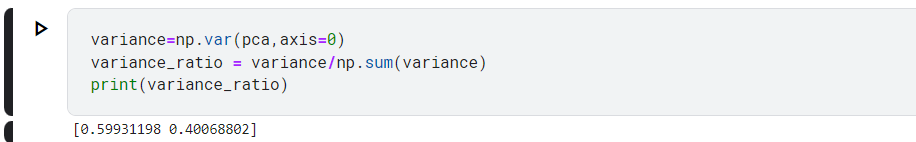
**EXPERIMENT - 4**

**Aim:** Perform Feature Reduction Techniques on the Collected Dataset.

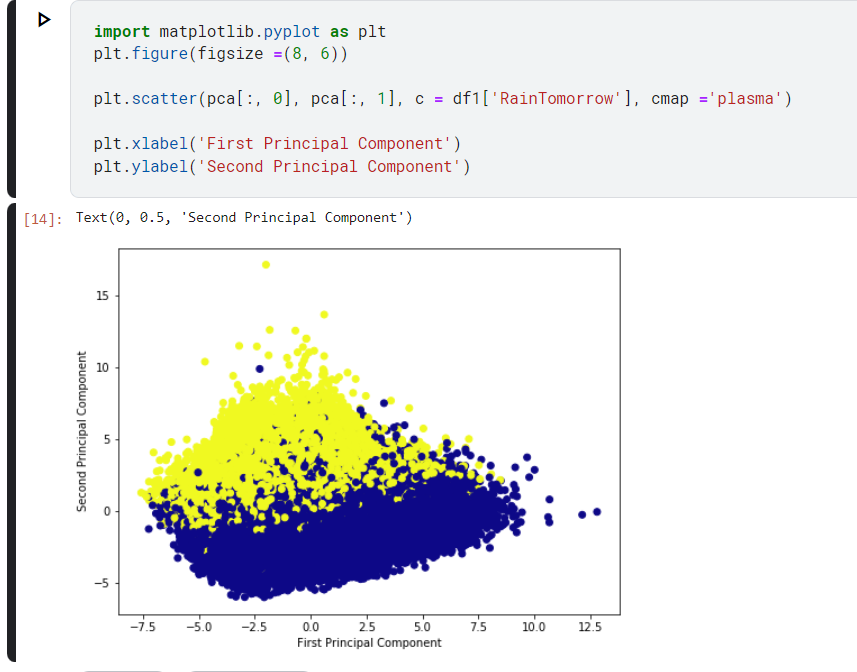
**Principal Component Analysis:**

PCA is a statistical technique that simplifies a high-dimensional dataset by identifying its most important patterns. It does so by finding the principal components, which are linear combinations of the original features that explain the most variance in the data. These principal components are calculated by finding the eigenvectors and eigenvalues of the dataset's covariance matrix. The dataset can then be projected onto a lower-dimensional space by selecting only the most important principal components. This resulting dataset will have fewer dimensions but still retain most of the original variance.





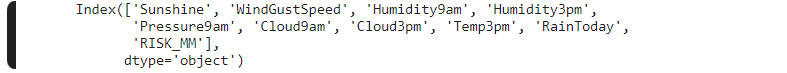
**Finding corelation between both the components**



**Information Gain Feature Evaluation**

Information Gain is a feature evaluation metric that measures the amount of information gained about the target variable by knowing the value of a particular feature. It calculates the difference between the entropy of the target variable before and after splitting the data based on a particular feature. Features with higher Information Gain are considered more important, as they provide more information about the target variable. Information Gain is commonly used in decision tree algorithms to determine the order in which features are split to maximize the information gained at each step.



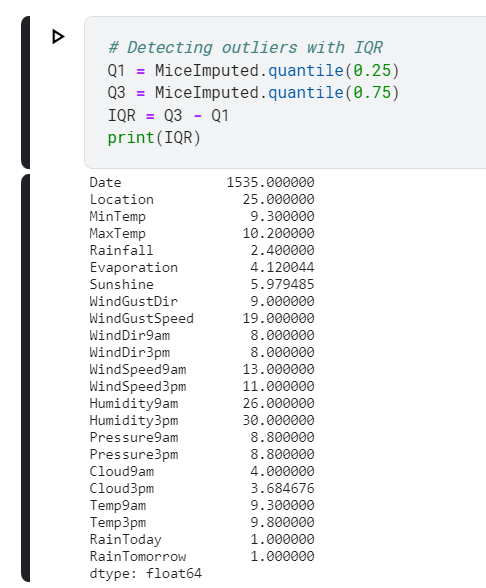


**Relief Attribute Feature Evaluation**

Relief attribute feature evaluation is a type of feature selection method that evaluates the relevance of features by computing the difference between the feature values of the nearest instances with different class labels. This method is useful for datasets with noisy or redundant features. The algorithm works by iteratively updating feature weights based on the misclassification rate. The weights are then used to rank the features, with higher weights indicating more relevance. ReliefF is an extension of the original Relief algorithm that computes the weights using a weighted average over multiple random feature subsets. This approach helps to reduce the impact of noisy or irrelevant features on the ranking.

**Correlation Based Feature Evaluation**

Correlation-Based Feature Evaluation is a method for selecting features that are highly correlated with the target variable. It involves calculating the correlation coefficient between each feature and the target variable, and selecting features with high correlation coefficients. Features with low correlation coefficients are discarded. The correlation coefficient measures the strength and direction of the linear relationship between two variables. A high correlation coefficient indicates a strong linear relationship, while a low correlation coefficient indicates a weak or no linear relationship. This method is simple and effective in selecting relevant features, but it may not capture non-linear relationships between features and the target variable. It is also sensitive to outliers and multicollinearity between features.



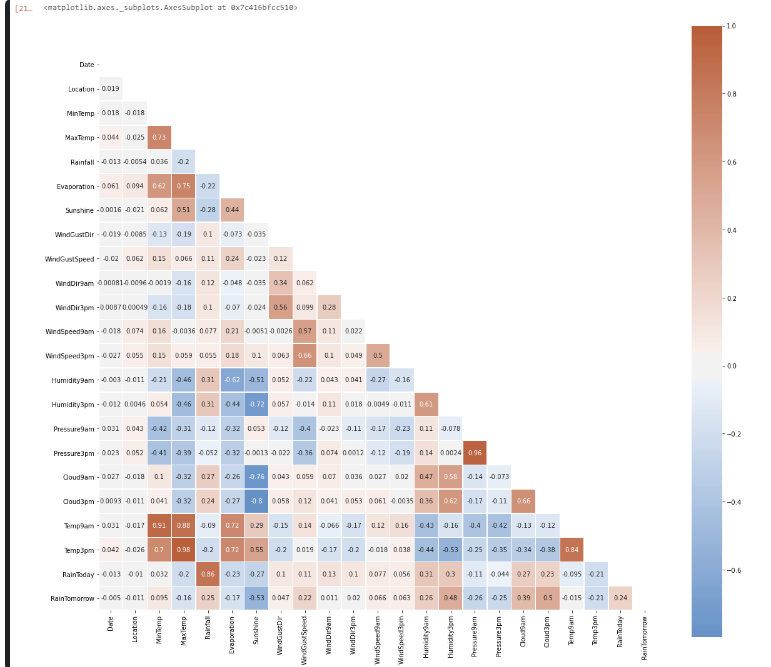


We observe that the original dataset was having the shape (87927, 24). After running outlier-removal code snippet, the dataset is now having the shape (86065, 24).

So, the dataset is now

free of 1862 outliers. We will now check for multi-collinearity i.e. whether any feature is highly correlated with another.

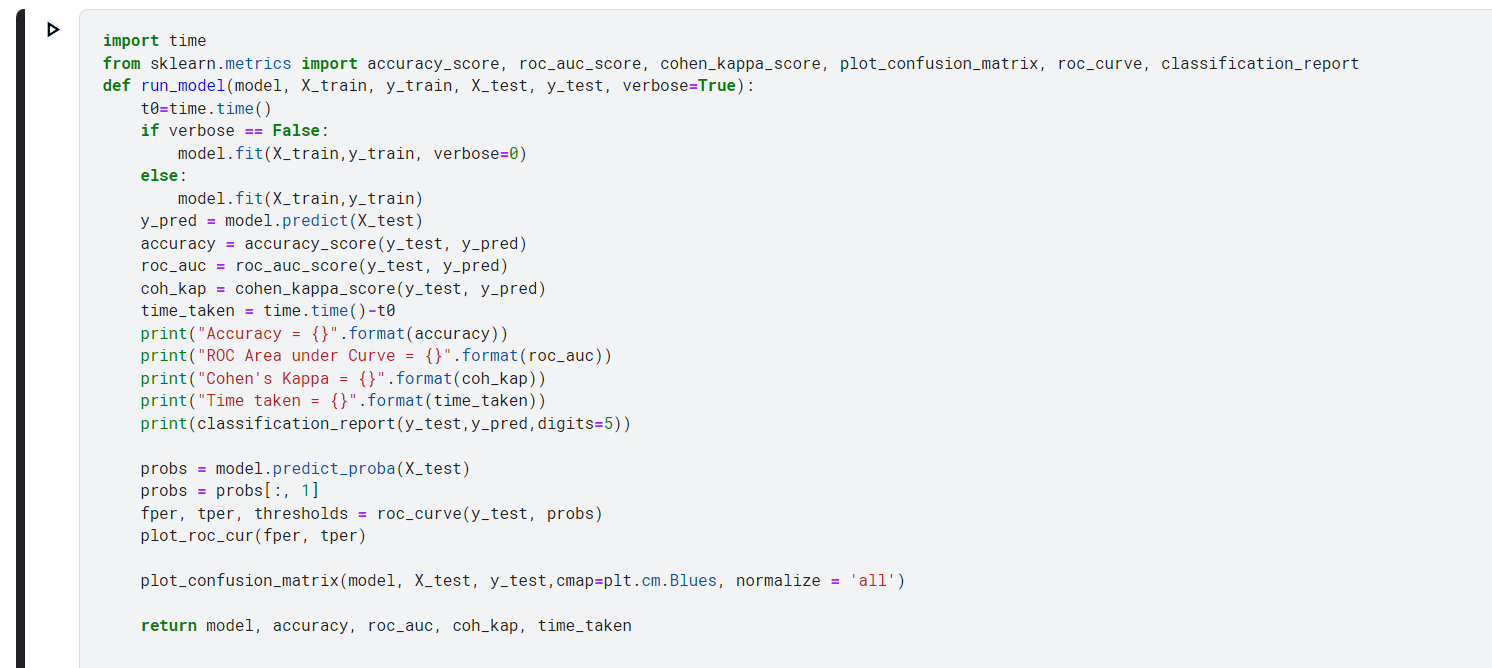




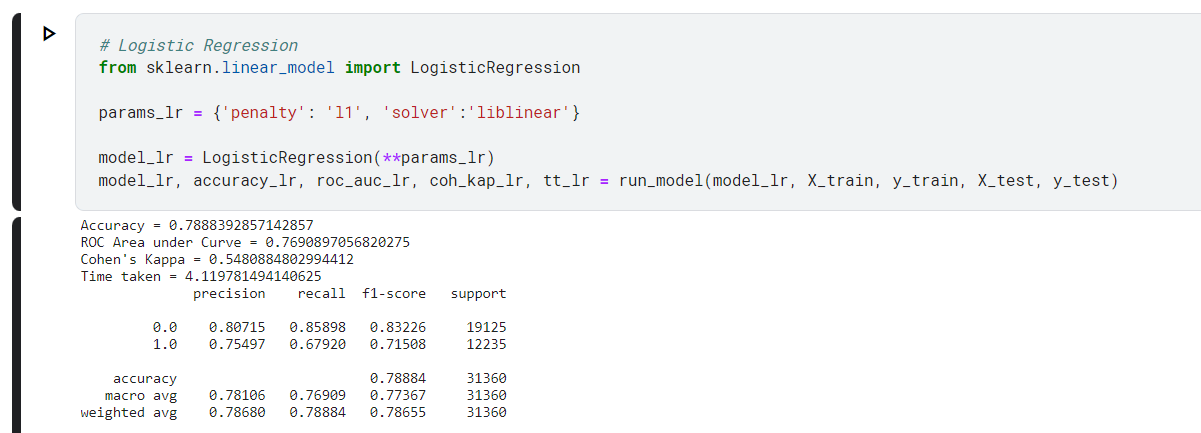
**EXPERIMENT – 5**

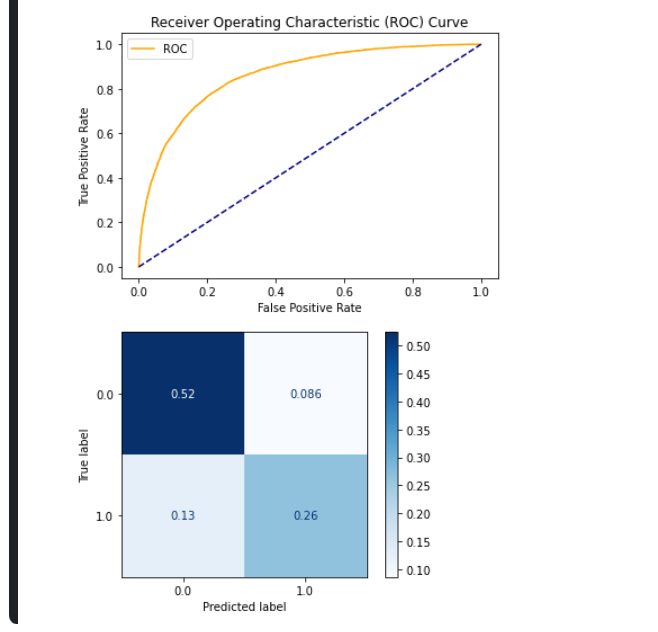
**Aim:** Develop a machine learning model for the selected topic (minimum 10 dataset and 10 techniques).





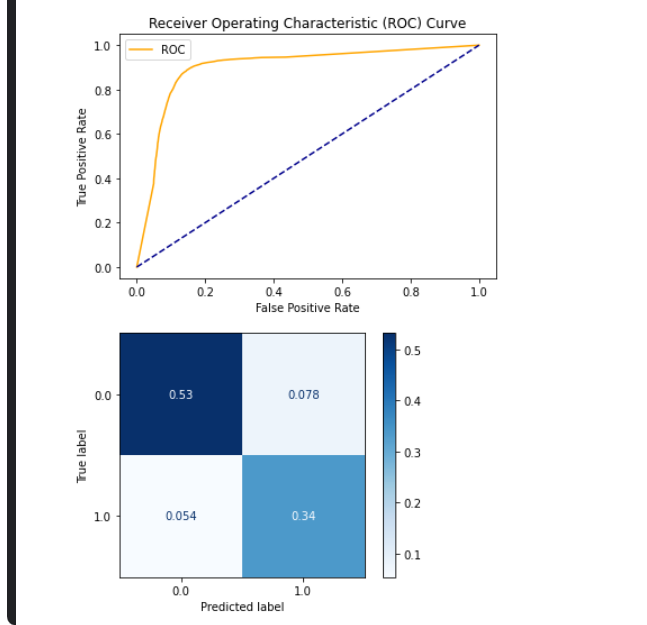
**Performing Logistic Regression**



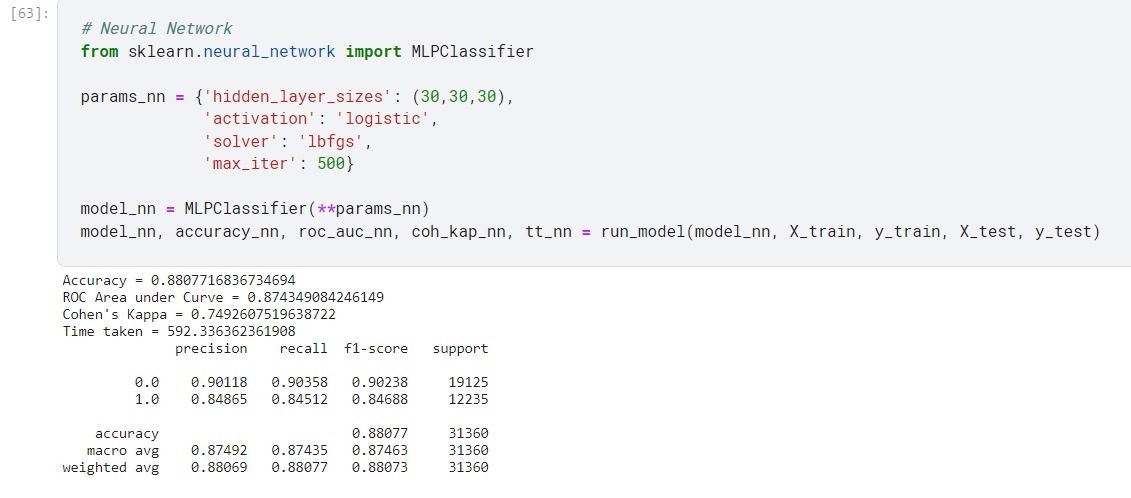


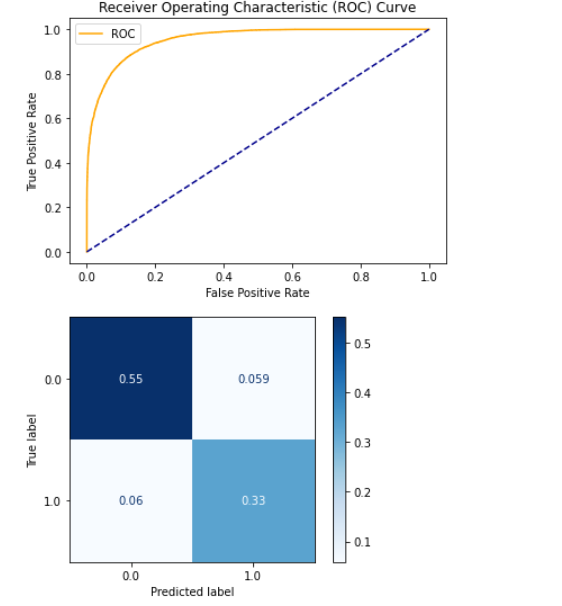
**Performing Decision Tree**



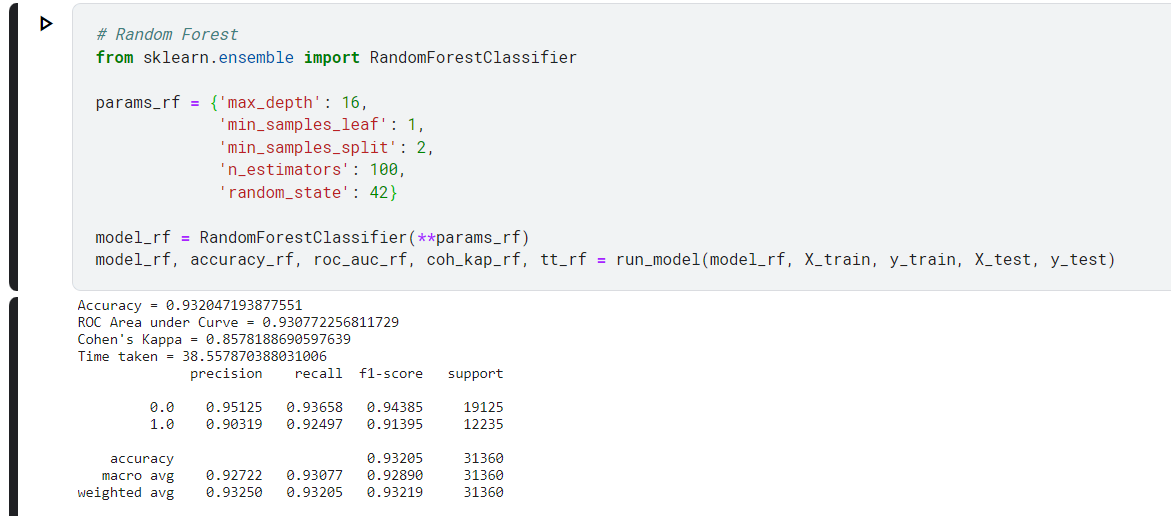


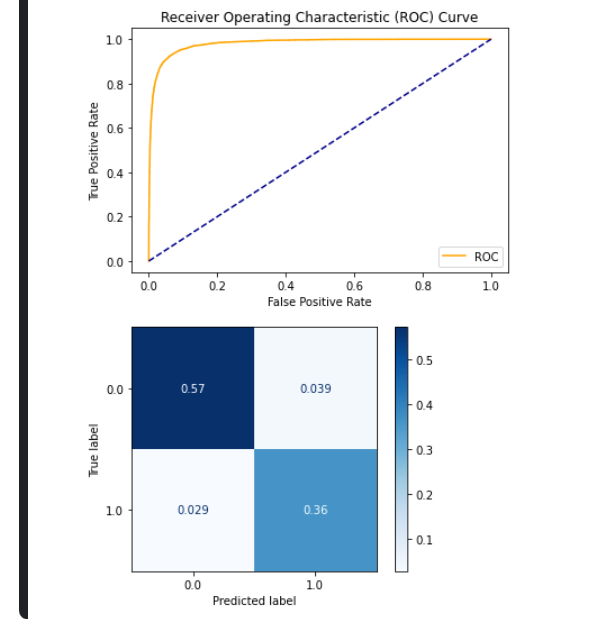
**Performing Neural Network**



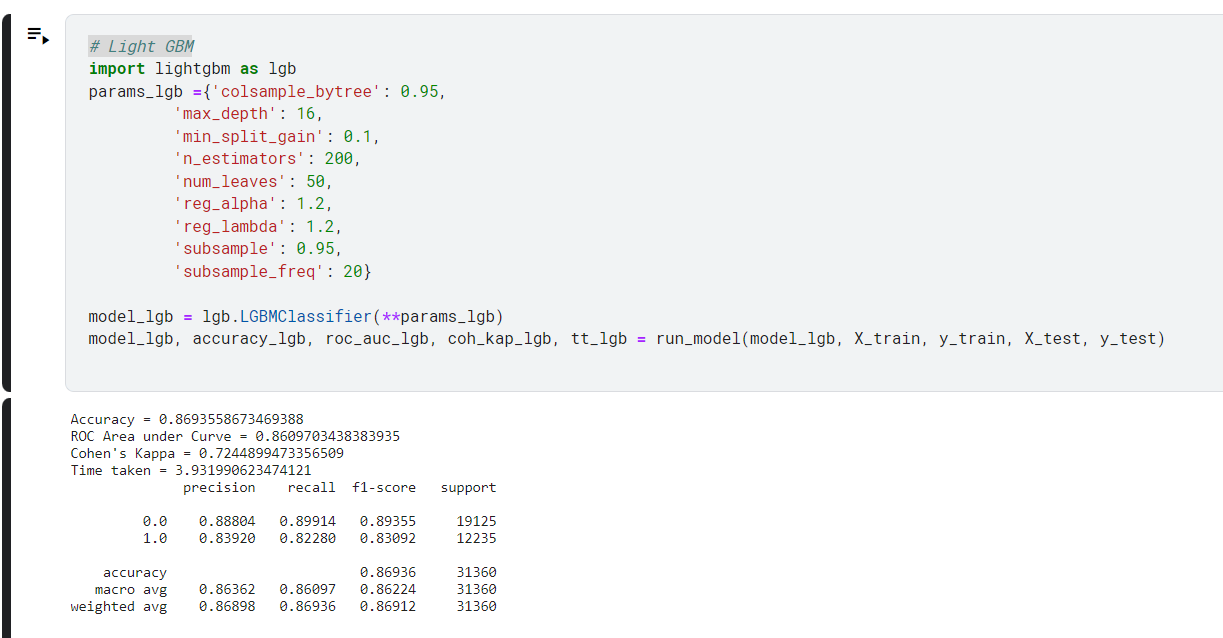


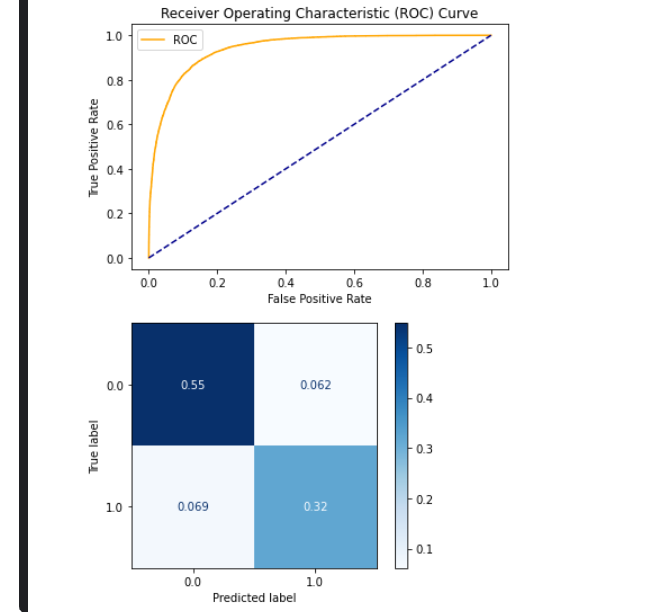
**Perfroming Random Forest**





**Performing Light GBM**



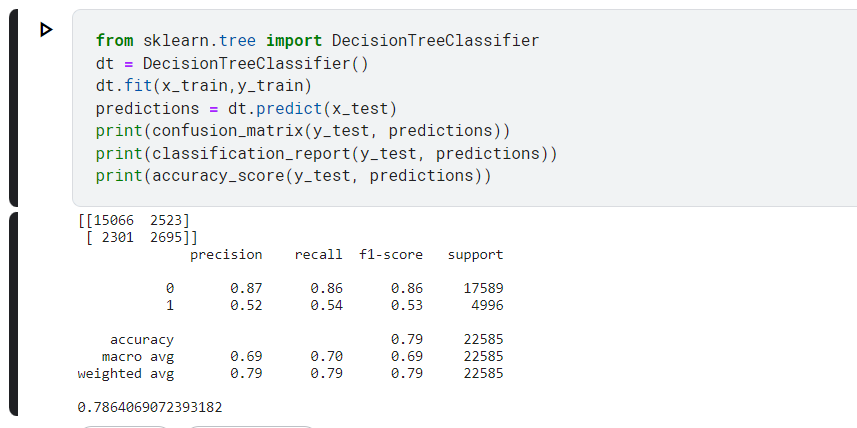


**DATASET 2** –

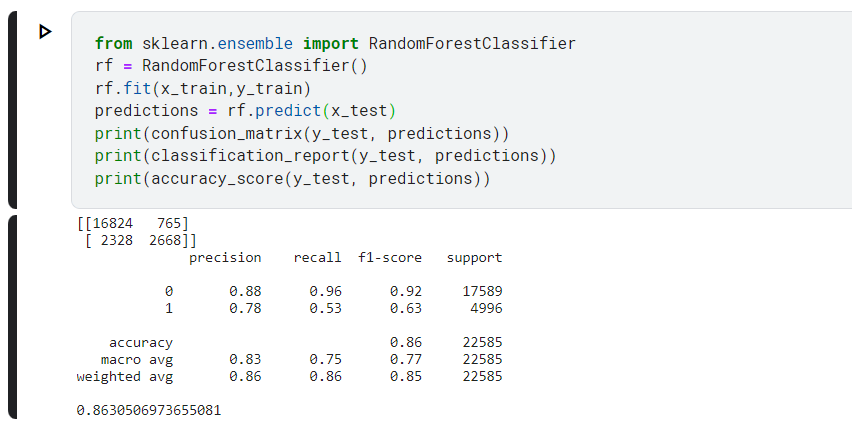
**Performing Logistic Regression**



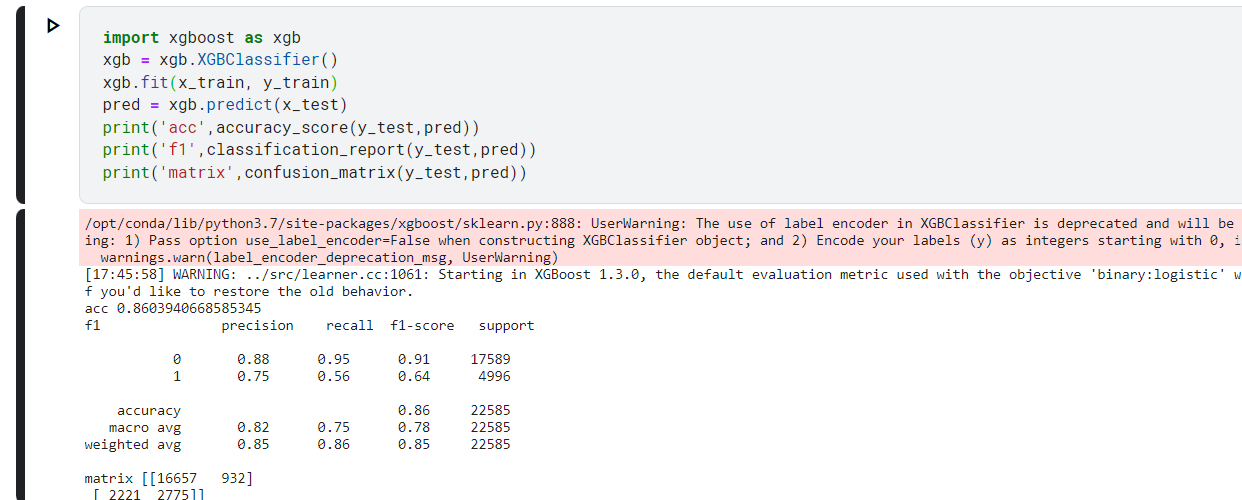
**Performing Decision Tree**



**Performing Random Forest Classifier**



**Performing XGBoost Classifier**



**EXPERIMENT 6**

**Aim - State the hypothesis for rainfall prediction using logistic regression**

The hypothesis for rainfall prediction using logistic regression could be stated as follows:

The hypothesis for rainfall prediction using logistic regression is that there exists a linear relationship between the input variables and the probability of rainfall occurrence. The logistic regression model assumes that the log odds of rainfall occurrence can be modeled as a linear function of the input variables, and that the relationship between the input variables and the log odds of rainfall occurrence is characterized by the logistic function. The model further assumes that the errors in the model are normally distributed, and that the input variables are not highly correlated with each other. By estimating the coefficients of the model from a training dataset, the model can be used to predict the probability of rainfall occurrence given a set of input variables for a new data point. The hypothesis is tested by evaluating the goodness-of-fit of the model and assessing its predictive accuracy on a validation dataset.

**State the hypothesis for rainfall prediction using Decision Tree**

The hypothesis for rainfall prediction using Decision Tree could be stated as follows:

The hypothesis for rainfall prediction using Decision Tree is that by recursively partitioning the input space based on the values of the input variables, it is possible to create a set of rules that can accurately predict the occurrence of rainfall. The decision tree algorithm builds a tree-like model in which each internal node represents a test on an input variable, and each leaf node represents a predicted class (rain or no rain). The hypothesis is that the decision tree model will be able to capture complex interactions and nonlinearities in the input variables that are relevant for rainfall prediction. The model can also handle missing data and irrelevant variables by choosing the most informative variables for splitting. The model's accuracy is evaluated using a validation dataset, and the hypothesis is supported if the model achieves high accuracy and generalizes well to new data. However, the hypothesis is also susceptible to overfitting, especially when the tree is allowed to grow too deep or when the input variables are highly correlated. In such cases, pruning techniques can be used to reduce the complexity of the tree and improve its generalization performance.

**State the hypothesis for rainfall prediction using Neural Network**

The hypothesis for rainfall prediction using Neural Network could be stated as follows:

By using a deep learning neural network model trained on high-resolution satellite imagery, we can accurately predict short-term (e.g., hourly) rainfall patterns for specific geographical locations. This will be particularly useful for weather forecasting and emergency response planning. By analyzing historical rainfall data and land-use information using a convolutional neural network, we can predict the impact of deforestation and other land-use changes on future rainfall patterns. This will help policymakers and conservationists to make informed decisions about land management practices that could affect local and regional weather patterns That is through a combination of weather sensor data and machine learning algorithms, we can develop a neural network model that can predict long-term (e.g., seasonal) rainfall patterns in a given region. This will provide valuable insights for farmers, water resource managers, and other stakeholders who need to plan for droughts and other weather-related risks

**State the hypothesis for rainfall prediction using Random Forest**

The hypothesis for rainfall prediction using Random Forest could be stated as follows:

A possible hypothesis for rainfall prediction using a random forest model is: "If we train a random forest model using historical weather data, such as temperature, humidity, wind speed, and atmospheric pressure, in combination with topographic and land cover data, we can accurately predict rainfall patterns for a given region. By using an ensemble of decision trees that are able to handle non-linear and interactive relationships between these variables, we can create a robust predictive model for rainfall. This could provide valuable insights for various applications, including agriculture, water management, and natural disaster preparedness."

In other words, the hypothesis assumes that the occurrence of rainfall can be predicted by constructing a random forest model that consists of multiple decision trees, each trained on a subset of the available features and observations. The random forest algorithm uses a suitable dataset of historical weather observations and rainfall data to learn the optimal set of decision rules that can accurately predict the occurrence of rainfall. The goal is to use this random forest model to predict the occurrence of rainfall in real-time based on the current weather conditions. The hypothesis further assumes that the random forest model will be able to capture complex interactions between the predictor variables and provide better predictive accuracy compared to a single decision tree.

**State the hypothesis for rainfall prediction using Light GBM**

The hypothesis for rainfall prediction using Light GBM could be stated as follows:

A possible hypothesis for rainfall prediction using Light GBM (Gradient Boosting Machine) is: "If we train a Light GBM model using historical weather data, including factors such as temperature, humidity, wind speed, atmospheric pressure, and topographic and land cover data, we can develop an accurate predictive model for rainfall patterns. By using a gradient boosting algorithm that iteratively improves the model's performance by adding weak learners, we can create a powerful machine learning tool for predicting rainfall. This could have important applications in agriculture, hydrology, and disaster management." validity of this statement will depend on the quality of the data, the rigor of the analysis, and the accuracy of the predictions. It will be important to validate the model using independent observations and to account for factors such as overfitting, bias, and the potential for spurious correlations. Nonetheless, a well-designed Light GBM model could provide a powerful tool for predicting rainfall patterns and their impacts on society and the environment.

**Analysis Plan for rainfall prediction using logistic regression:**

1. Data collection and preprocessing:

* Collect historical weather data for the region of interest, including variables such as temperature, humidity, wind speed, atmospheric pressure, and precipitation.
* Preprocess the data by checking for missing values, outliers, and anomalies.
* Merge or join datasets if necessary to include additional variables, such as topographic or land cover data.

1. Data exploration:

* Perform descriptive statistics and data visualization to identify trends, patterns, and correlations in the data.
* Check for multicollinearity and other potential issues that may affect the logistic regression model.

1. Model building:

* Split the data into training and testing sets to evaluate the performance of the model.
* Use logistic regression to build a model to predict rainfall, with precipitation as the dependent variable and the other weather variables as independent variables.
* Choose a suitable method to handle categorical variables, such as one-hot encoding or dummy variables.
* Use techniques such as forward or backward selection to select the most significant independent variables for the model.

1. Model evaluation:

* Evaluate the performance of the model using metrics such as accuracy, sensitivity, specificity, and AUC-ROC curve.
* Use cross-validation or bootstrap methods to estimate the generalization error of the model.
* Compare the performance of the logistic regression model with other machine learning models, such as decision trees, random forests, or gradient boosting machines.

1. Model interpretation:

* Interpret the coefficients of the logistic regression model to gain insights into the factors that are most influential in driving rainfall patterns.
* Check for the significance of the coefficients and the goodness of fit of the model.
* Use visualization tools, such as heat maps or scatter plots, to explore the relationships between the independent variables and rainfall.

1. Reporting:

* Present the results of the analysis in a clear and concise manner, using tables, charts, and graphs where appropriate.
* Summarize the key findings of the analysis and discuss their implications for the region of interest.
* Discuss limitations of the analysis and future directions for research.

Top of Form

**Formulate an Analysis Plan for rainfall prediction using decision tree**

1. Data collection: Gather historical rainfall data from reliable sources. This data should include information on the amount of rainfall, date and time of occurrence, geographical location, and other relevant factors that may influence rainfall.
2. Data preprocessing: Clean the data by removing any missing or duplicate values, and transforming the data into a format suitable for analysis. This may include aggregating the data into monthly or yearly time intervals, converting categorical variables into numeric values, and normalizing the data if necessary.
3. Feature selection: Identify the key features that may influence rainfall using domain knowledge and statistical techniques such as correlation analysis. This may include factors such as temperature, humidity, wind speed, pressure, and geographical location.
4. Decision tree model selection: Choose an appropriate decision tree algorithm such as ID3, C4.5, CART, or Random Forest based on the problem requirements, data characteristics, and available computational resources.
5. Model training: Divide the data into training and testing sets and use the training set to train the decision tree model. The model should be tuned using hyperparameter optimization techniques to ensure optimal performance.
6. Model evaluation: Evaluate the performance of the decision tree model on the testing set using appropriate metrics such as accuracy, precision, recall, F1 score, and AUC-ROC. This will help to determine if the model is overfitting or underfitting the data.
7. Model deployment: Once the model is trained and tested, deploy it in a production environment for real-time rainfall prediction. The model should be regularly updated with new data to ensure its accuracy and reliability.
8. Model interpretation: Finally, interpret the decision tree model to gain insights into the factors that are most important in predicting rainfall. This can help to inform future research and decision-making in areas such as agriculture, water management, and climate change.

Top of Form

**Formulate an Analysis Plan for rainfall prediction using Neural network**

1. Data Collection: Collect historical rainfall data for the region of interest. The dataset should include at least a few years worth of daily rainfall measurements.
2. Data Preparation: Preprocess the data to ensure that it is ready for analysis. This includes steps such as checking for missing or erroneous values, normalizing the data to a consistent scale, and splitting the dataset into training and testing sets.
3. Model Selection: Choose an appropriate neural network architecture for the rainfall prediction task. This may involve experimenting with different network topologies, activation functions, and other hyperparameters.
4. Model Training: Train the neural network using the training data. This involves feeding the network input data (e.g. previous day's rainfall measurements) and adjusting the weights of the network to minimize the difference between the predicted and actual rainfall values.
5. Model Evaluation: Evaluate the performance of the trained neural network using the testing data. This involves calculating metrics such as mean absolute error (MAE) or root mean squared error (RMSE) to assess how accurately the network is able to predict rainfall.
6. Model Optimization: Based on the evaluation results, make any necessary adjustments to the model architecture or hyperparameters to improve its performance.
7. Predictions: Once the model has been optimized, it can be used to predict future rainfall values based on new input data. It is important to monitor the performance of the model over time and make adjustments as necessary to ensure that it continues to provide accurate predictions.
8. Reporting: Finally, summarize the results of the analysis and provide recommendations based on the findings. This may include suggestions for future data collection or model refinement.

Top of Form

**Formulate an Analysis Plan for rainfall prediction using Random Forest**

a. Define the problem: The first step is to clearly define the problem and the research question you want to answer. For example, you may want to predict the amount of rainfall based on various meteorological variables such as temperature, humidity, wind speed, etc., using a random forest model.

b. Collect data: The next step is to collect data on the meteorological variables and rainfall from reliable sources such as weather stations, satellite imagery, or climate databases. Ensure that the data is complete, accurate, and representative of the study area.

c. Explore the data: Before applying random forest analysis, it is important to explore and visualize the data to gain insights and check for any anomalies or outliers. Use summary statistics, histograms, scatter plots, and correlation matrices to understand the distribution and relationships between variables.

d. Prepare the data: Once you have explored the data, you need to prepare it for random forest analysis. This involves cleaning, transforming, and scaling the data. For example, you may need to remove missing values, standardize the variables, or convert categorical variables into binary dummy variables.

e. Split the data: The next step is to split the data into training and testing sets. The training set will be used to build the random forest model, while the testing set will be used to evaluate the model's performance on new data.

f. Build the model: The next step is to build the random forest model using the prepared training data. Use techniques such as bagging, feature subsetting, and decision tree ensembles to create a set of individual decision trees that work together to make predictions. Experiment with different hyperparameters such as the number of trees, the maximum depth of each tree, or the minimum number of samples per leaf.

g. Evaluate the model: After building the model, you need to evaluate its performance using various metrics such as accuracy, precision, recall, F1-score, or AUC-ROC. These metrics will help you assess how well the model fits the data and how accurate its predictions are.

h. Test the model: Once you have evaluated the model using the training data, you need to test its performance on the testing data that the model has not seen before. Use the same evaluation metrics as in step 7 to test the model and avoid overfitting.

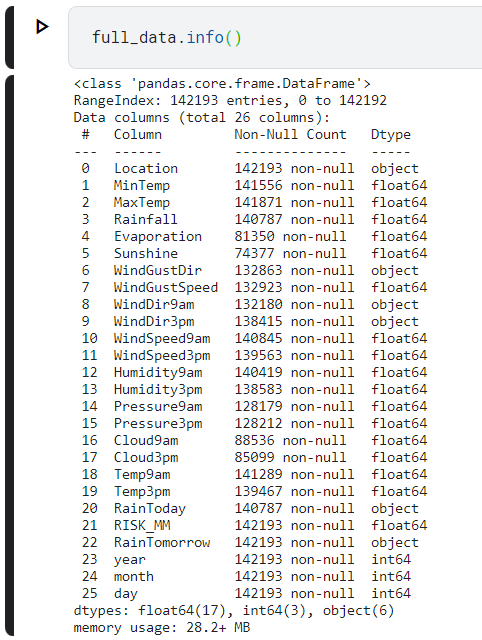
i. Interpret the results: Finally, you need to interpret the results of the random forest model in the context of the research question. Identify the most important predictor variables and their relationships, and explain how they influence the amount of rainfall. Also, discuss the limitations and assumptions of the random forest model and suggest future research directions

**Formulate an Analysis Plan for rainfall prediction using Light GBM**

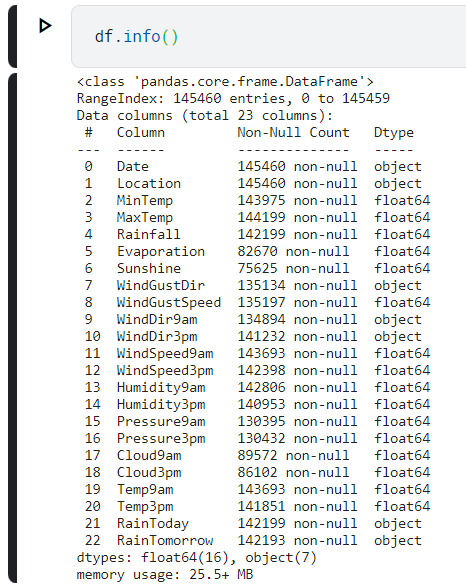
1. Data Preparation: Gather and preprocess the rainfall dataset, ensuring it's clean and formatted in a way that can be ingested by the Light GBM algorithm. Split the data into training, validation, and testing sets. Normalize the data to improve model performance.
2. Feature Engineering: Create new features from existing data to improve model accuracy, such as extracting time-series features, creating lagged variables, and aggregating data by geographical regions.
3. Model Training: Train a Light GBM model on the training set, tuning hyperparameters like learning rate, number of trees, depth of trees, and early stopping to prevent overfitting.
4. Model Validation: Evaluate the model's performance on the validation set, using metrics like mean squared error (MSE) and mean absolute error (MAE). Adjust the model as needed and retrain it on the entire training set.
5. Model Testing: Test the final model on the testing set to estimate its real-world performance.
6. Model Interpretation: Use feature importance scores to understand which features contribute most to the model's predictions.
7. Deployment: Deploy the model in production and monitor its performance over time, using additional data to retrain and improve the model as needed.

**Analyse the sample data**

**Australia - Alubury Weather Dataset**



**Australia Dataset**

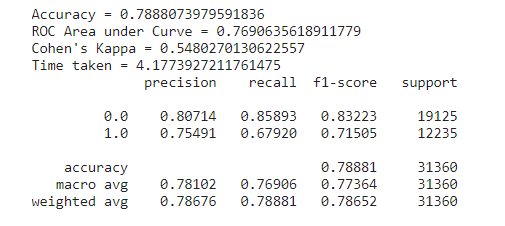


**Interpret the results**

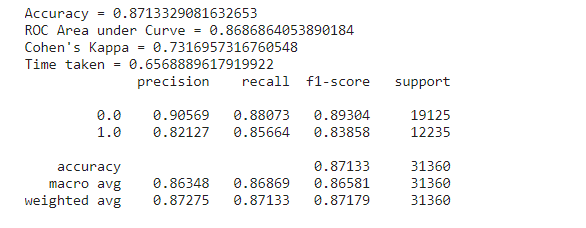
We have used 2 datasets – the Australia-Alubury Weather Dataset and the Australia Weather dataset as training data for our rainfall prediction algorithms.

**Australia-Alubury Weather Dataset**

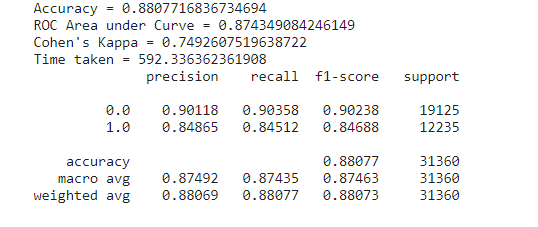
**Logistic Regression Confusion Matrix –**



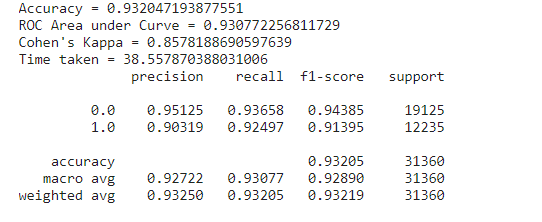
**Decision Tree – Confusion Matrix –**



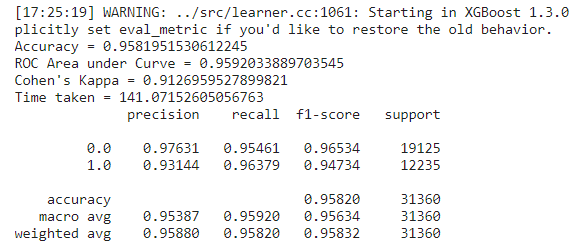
**Neural Network – Confusion Matrix –**

****

**Random Forest – Confusion Matrix**

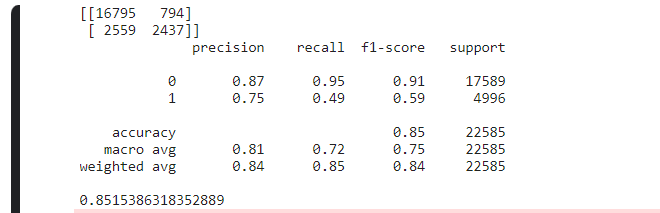


**XGBoost – confusion Matrix –**

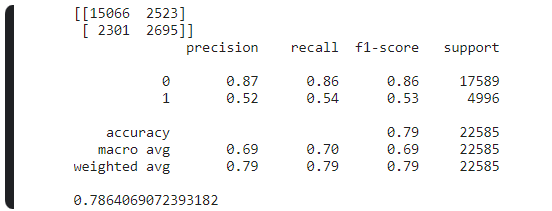


**Australia Dataset**

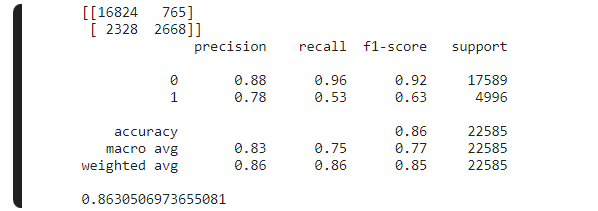
**Logistic Regression Confusion Matrix –**



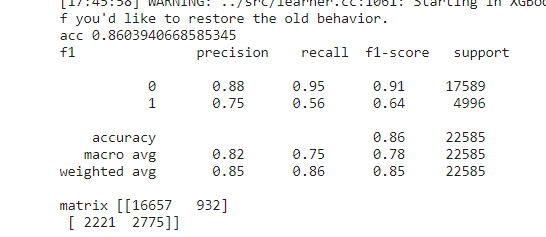
**Decision Tree Confusion Matrix –**



**Random Forest Confusion Matrix –**



**XGBoost Confusion Matrix –**



**Estimate type 1 and type 2 errors**

**Type 1 – False Positive Type 2 – False Negative**

**Alubury Weather Dataset** **Australia Weather Dataset** 1. Logistic Regression 1. Logistic Regression Type 1 - 898 Type 2- 2549 Type 1 - 857 Type 2 2454 2. Neural Network 2. Neural Network Type 1 – 923 Type 2- 2529 Type 1 - 916 Type 2 1029 3. Decision Tree 3. Decision Tree Type 1 - 2547 Type 2- 2281 Type 1 - 467 Type 2 2114 4. Random Forest 4. Random Forest Type 1 - 897 Type 2 -2321 Type 1 - 736 Type 2 2023 5. Xgboost 5. Xgboost

Type 1 - 916 Type 2- 2029 Type 1 - 876 Type 2 1329