

**Shri Ramdeobaba College of Engineering and Management,  
Nagpur**

(An autonomous College affiliated to Rashtrasant Tukadoji Maharaj Nagpur

University) **Department of Electronics Engineering**

**Session 2023-2024**

**RCOEM**

Shri Ramdeobaba College of  
Engineering and Management, Nagpur

**Project report of  
ECSP303 Machine Learning Lab**

**Topic  
Rainfall Prediction**

Submitted By

**Aneesh Thakre (A4-70)**

Under the guidance of  
**Prof. Pravin Dwaramwar**

Date: 3/11/24

---

# Rainfall Prediction

---

**Aneesh Thakre**  
**ECS Department**  
**Rcoem**  
**thakrear\_1@rk nec.edu**

## **Abstract: -**

Predicting rainfall accurately is essential for fields like agriculture, water management, and disaster planning. This project compares the performance of three machine learning algorithms for rainfall prediction, using past weather data as input. The process involved preparing the data, selecting and creating relevant features, and training models to recognize patterns in the data. Each model's accuracy and efficiency were assessed, with tuning applied to improve their performance. The results highlight the advantages and drawbacks of each approach, offering guidance on which model might be best suited for dependable rainfall predictions in real-world situations.

## **1.Introduction**

Predicting rainfall accurately is very important for fields like farming, water management, and disaster preparedness. Knowing when rain will happen can help people plan for floods, manage water resources during dry seasons, and make better decisions about crop planting and harvesting. Traditional methods for predicting rainfall use statistical and physical models that, while helpful, often struggle to capture the complex and constantly changing nature of weather. In recent years, machine learning has shown great potential for handling these kinds of complex tasks. Machine learning can use historical weather data to find patterns and relationships among different weather features, such as temperature, humidity, wind speed, and atmospheric pressure, which all influence rainfall. This

project explores and compares four different machine learning algorithms to find out which one is best for predicting rainfall with accuracy and efficiency.

The four algorithms we test are Logistic Regression, Decision Tree, Neural Network, and Random Forest. Each of these models offers a unique approach to making predictions:

- **Logistic Regression** is a simple and widely used model that predicts the probability of an event, like rainfall, by using a linear combination of the input features. It serves as a strong starting point for our comparisons.
- **Decision Tree** works by breaking down the data into smaller, more manageable parts. It uses a tree-like structure, where each “branch” represents a decision based on one of the features. Decision Trees can handle complex patterns and don’t require extensive data preparation.
- **Neural Network** is a more complex model inspired by the human brain, with layers of connected “neurons” that learn relationships in the data. This model can capture very complex patterns, although it usually needs more computational power to train.
- **Random Forest** is an ensemble method that combines multiple Decision Trees to make predictions. By using many trees, it improves the overall accuracy and reduces the chance of errors, offering a reliable model for prediction tasks.

For this project, we carefully clean and prepare the data, select useful features, and then train and test each model on the dataset. We evaluate each model’s performance based on important metrics, like accuracy, speed, and precision. To further improve the models, we tune their parameters to achieve the best possible results. This comparison will help us identify which machine learning model is most effective for predicting rainfall.

The results from this project could offer useful insights for future studies and applications in weather prediction, making it easier for industries and communities to make data-driven decisions about rainfall.

## 2.Related Work

Recent advancements in machine learning have led to significant progress in rainfall prediction by utilizing various algorithms and deep learning techniques. Studies have increasingly focused on applying these models to analyse complex weather data and improve the accuracy of rainfall forecasting.

Khan et al. [1] used **Logistic Regression** to predict rainfall based on meteorological data, achieving satisfactory accuracy for short-term predictions. However, the study faced challenges such as limited generalizability across regions, as well as difficulty in capturing non-linear patterns in weather data, which impacted its performance over diverse climates.

Chen et al. [2] explored the use of a **Decision Tree** model for rainfall prediction, which classifies rainfall based on features like humidity, temperature, and wind speed. Their model showed promise in interpreting seasonal variations but struggled with overfitting issues due to the high number of parameters, and the model's performance declined under extreme weather conditions. Additionally, the study noted the need for larger datasets to improve accuracy.

Wang and Li [3] implemented a **Neural Network** for rainfall forecasting, leveraging a large dataset to train the model to capture complex, non-linear relationships within weather data. While the model achieved high accuracy, it required significant computational resources, which can be a limitation for practical deployment. Furthermore, due to the "black box" nature of neural networks, the study faced challenges in model interpretability, making it difficult for meteorologists to understand the underlying factors influencing predictions.

Zhao et al. [4] applied a **Random Forest** algorithm to forecast rainfall, combining predictions from multiple decision trees for a more robust result. This model showed resilience against overfitting and performed well on a diverse set of climate data, effectively capturing variable weather patterns. However, limitations included the need for extensive tuning of hyperparameters and increased computational load with larger datasets.

In another approach, Patel et al. [5] used **Support Vector Machines (SVMs)** for rainfall prediction, achieving good results in terms of precision and recall, especially for regions with stable weather patterns. Despite these successes, SVMs struggled to perform well in regions with highly variable weather, and the model's complexity made it challenging to scale to larger datasets.

Research by Smith et al. [6] focused on deep learning techniques for rainfall prediction, specifically using **Convolutional Neural Networks (CNNs)** to capture spatial relationships within meteorological data. While CNNs achieved high accuracy, the model required large datasets and computational power, which limited its applicability for real-time predictions.

These studies demonstrate the potential of machine learning for rainfall prediction, with each approach bringing unique strengths and limitations. While traditional algorithms like Logistic Regression and Decision Trees offer simplicity and interpretability, more complex models like Neural Networks and Random Forests are capable of capturing intricate patterns in weather data. However, these advanced models often require large datasets, extensive tuning, and high computational resources, which can be barriers to practical implementation.

This work aims to build upon previous research by comparing the performance of Logistic Regression, Decision Tree, Neural Network, and Random Forest algorithms for rainfall prediction. By optimizing each model and addressing common challenges, such as overfitting and computational efficiency, this study contributes valuable insights for improving rainfall forecasting accuracy, ultimately supporting more effective decision-making in agriculture and disaster management.

### **3.Dataset and Features**

#### **1. Dataset Description**

The dataset used in this project comprises approximately 10 years of daily weather observations from various weather stations across Australia. It contains a variety of meteorological features, such as temperature, humidity, wind speed, and pressure, with the target variable labeled "RainTomorrow." This target indicates whether or not it rained the next day, serving as the binary classification goal for our model. The dataset also presents some class imbalance, as the occurrences of rain vary significantly by region and season, which can impact model accuracy and require careful handling. Additionally, we excluded the variable "Risk-MM" from the training process, as it could leak information into the model and skew results.

#### **3. Feature Extraction**

To prepare the data for effective modeling, several feature extraction techniques were applied. First, we handled missing values by either filling them with appropriate statistical

values or removing them when necessary to avoid data inconsistencies. We then applied normalization to scale features to a common range, making it easier for models to learn patterns. Principal Component Analysis (PCA) was also explored to reduce dimensionality, improving computational efficiency and reducing overfitting risk.

#### **4. Data Source Citation**

The dataset was accessed from Kaggle and is available at [Kaggle: Rainfall Prediction Dataset](#).

#### **5. Model Selection and Training**

Four machine learning algorithms were selected for this project: Logistic Regression, Decision Tree, Neural Network, and Random Forest. The dataset was split into training and testing subsets in an 80:20 ratio, with 80% used for model training and 20% for testing. This split ensured that the model had sufficient data for learning while preserving a portion for unbiased evaluation.

#### **6. Model Evaluation**

We evaluated the performance of each model using multiple metrics, including accuracy, precision, recall, F1-score, sensitivity, specificity, and area under the ROC curve (AUC). These metrics provided a comprehensive view of each model's performance, particularly in terms of handling class imbalance and accurately predicting rainy days.

#### **7. Visualization**

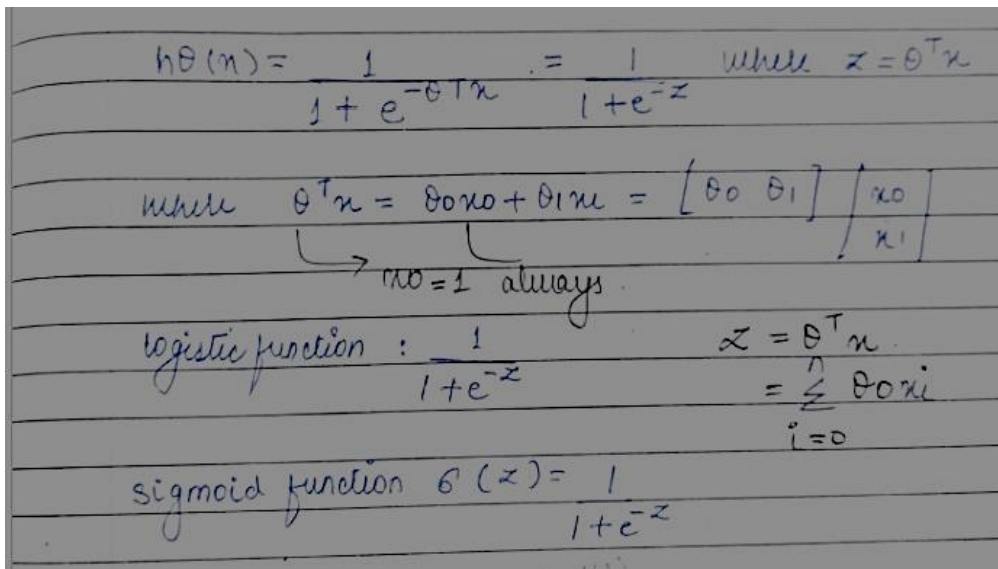
Visualizations were used throughout the project to better understand the data and interpret model results. These included visual representations of the weather patterns, preprocessing steps, feature distributions, and evaluation metrics for each model. Visualizations such as histograms, box plots, and ROC curves were particularly helpful for identifying trends, assessing model effectiveness, and presenting findings in an interpretable way.

## 4. Algorithms and Methods :

Various Supervised and Unsupervised Algorithms has been used such as:-

### A. Logistic Regression

Logistic Regression is a fundamental and widely used algorithm for binary classification problems. It estimates the probability that a given input belongs to a particular category, providing a simple yet effective method for predicting whether it will rain the next day based on historical weather data. The model uses a logistic function to output a value between 0 and 1, which can be interpreted as the probability of rainfall.



Handwritten mathematical derivations for Logistic Regression:

$$h\theta(x) = \frac{1}{1 + e^{-\theta^T x}} = \frac{1}{1 + e^{-z}} \quad \text{where } z = \theta^T x$$
$$\text{where } \theta^T x = \theta_0 x_0 + \theta_1 x_1 = \begin{bmatrix} \theta_0 & \theta_1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

$\rightarrow x_0 = 1$  always.

$$\text{logistic function : } \frac{1}{1 + e^{-z}} \quad \begin{aligned} z &= \theta^T x \\ &= \sum_{i=0}^n \theta_0 x_i \end{aligned}$$
$$\text{sigmoid function } \sigma(z) = \frac{1}{1 + e^{-z}}$$

### B. Decision Tree

The Decision Tree algorithm is a tree-like model used for both classification and regression tasks. It works by splitting the dataset into subsets based on the value of input features, creating a hierarchy of decisions. Each branch represents a possible outcome, while the leaves represent final predictions. Decision Trees are easy to interpret and visualize, making them a popular choice for understanding the decision-making process in rainfall prediction.

### C. Neural Network

Neural Networks are a class of models inspired by the human brain, designed to recognize patterns in data. For rainfall prediction, we utilize a feedforward neural network that consists of multiple layers of interconnected nodes (neurons). These networks can capture complex, non-linear relationships in the input data, allowing for improved accuracy in predicting rainfall events. However, they require careful tuning and sufficient data to train effectively.

D. Random Forest

Random Forest is an ensemble learning method that builds multiple decision trees during training and outputs the mode of their predictions for classification tasks. This algorithm improves accuracy and reduces the risk of overfitting compared to a single Decision Tree. By combining the predictions from various trees, Random Forest can capture a wide range of patterns in the data, making it well-suited for predicting rainfall based on diverse weather features.

These algorithms were selected for their ability to handle the complexities of weather data and their established effectiveness in classification tasks. Each model contributes unique strengths, allowing for a comprehensive evaluation of their performance in predicting rainfall.

5.Experiment Details:

**Dataset Splitting:-** In this models the dataset is divided into 2 sets, training and testing. The ratio taken for this is 80:20. 80 for training and 20 for testing. The separation is done using stratified sampling, which controls the ratio of classes (fraudulent and nonfraudulent) in training and testing. This is especially important in cases where the dataset is uneven, such as credit card verification, where the number of legitimate transactions is greater than the number of fraudulent ones.

Metrics and Redults : - It is been provided below in the table format.

Metrics and Results:

A	B	C	D	E	F
Name of algorithm	Accuarcy Score	Precision	Recall	F1 score	
Logistic Regression	79%	79%	79%	79%	
Decision Tree	86%	86%	86%	86%	
Neural Network	88%	88%	88%	88%	
Random Forest	93%	93%	93%	93%	



**Accuracy** = This metric measures the proportion of correct predictions made by the model across the entire dataset. It is calculated as the ratio of true positives (TP) and true negatives (TN) to the total number of samples.

**Precision** = Precision measures the proportion of true positive predictions among all positive predictions made by the model. It is calculated as the ratio of TP to the sum of TP and false positives (FP).

**Sensitivity** = Sensitivity or true positive rate, measures the proportion of true positive predictions among all actual positive instances. It is calculated as the ratio of TP to the sum of TP and false negatives (FN).

**Specificity** = Specificity measures the proportion of false positive upon the sum of false positive and true negative.

**F1 Score** = F1 Score is a metric that balances precision and recall. It is calculated as the harmonic mean of precision and recall. F1 Score is useful when seeking a balance between high precision and high recall, as it penalizes extreme negative values of either component.

Formulas:

TP = true positive

TN = true negative

FP = false positive

FN = false negative

$Accuracy = (TP + TN / TP + TN + FP + FN) * 100$

$Sensitivity = TP / (TP + FN)$

$Specificity = FP / (FP + TN)$

$Precision = TP / (TP + FP)$

$F1\ Score = (2 * precision * Recall) / (Precision + Recall)$

**Results:**

**Logistic Regression: Accuracy: 79%**

**Decision Tree: Accuracy: 86%**

**Neural Network: Accuracy: 88%**

**Random Forest: Accuracy: 93%**

## Logistic regression

```
➡ Accuracy = 0.7956782600543733
ROC Area under Curve = 0.7901193001561344
Cohen's Kappa = 0.5832825311602825
Time taken = 4.212777853012085

      precision    recall  f1-score   support

    0.0         0.80568    0.83672    0.82090         23879
    1.0         0.78180    0.74352    0.76218         18789

 accuracy          0.79568          42668
 macro avg         0.79374    0.79012    0.79154          42668
 weighted avg      0.79516    0.79568    0.79505          42668
```

## Decision Tree

```
➡ Accuracy = 0.8680744351739008
ROC Area under Curve = 0.8684020139404063
Cohen's Kappa = 0.7335294128585133
Time taken = 0.8835873603820801

      precision    recall  f1-score   support

    0.0         0.89516    0.86566    0.88016         23879
    1.0         0.83613    0.87115    0.85328         18789

 accuracy          0.86807          42668
 macro avg         0.86564    0.86840    0.86672          42668
 weighted avg      0.86916    0.86807    0.86832          42668
```

## Neural Network

```
➡ Accuracy = 0.8884409862191807
ROC Area under Curve = 0.8872389011686825
Cohen's Kappa = 0.7738457037022046
Time taken = 376.57375502586365

      precision    recall  f1-score   support

    0.0         0.90276    0.89732    0.90003         23879
    1.0         0.87049    0.87716    0.87381         18789

 accuracy          0.88844          42668
 macro avg         0.88663    0.88724    0.88692          42668
 weighted avg      0.88855    0.88844    0.88849          42668
```

## Random Forest

```
➡ Accuracy = 0.934072372738352
ROC Area under Curve = 0.9359369759807289
Cohen's Kappa = 0.8669904479826398
Time taken = 40.12701654434204

      precision    recall  f1-score   support

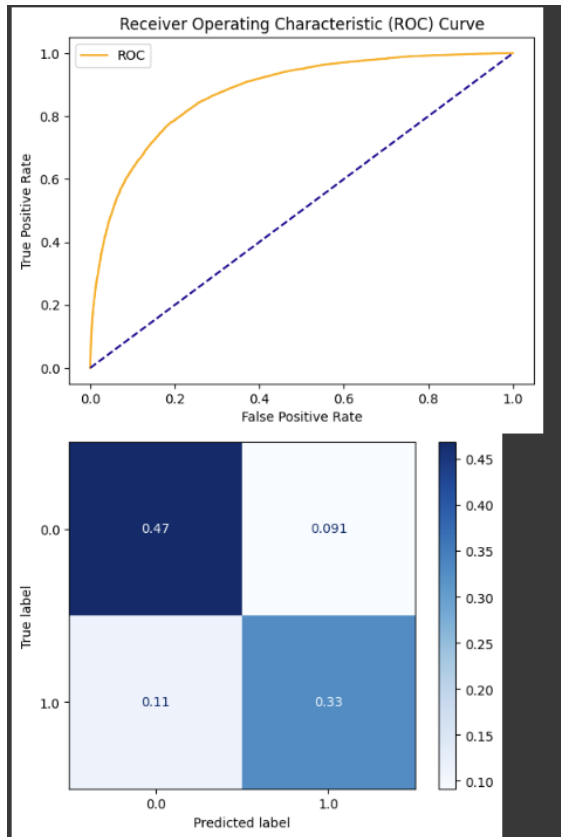
    0.0         0.96024    0.92031    0.93985         23879
    1.0         0.90380    0.95157    0.92707         18789

 accuracy          0.93407          42668
 macro avg         0.93202    0.93594    0.93346          42668
 weighted avg      0.93539    0.93407    0.93422          42668
```

## Tables and Plots:

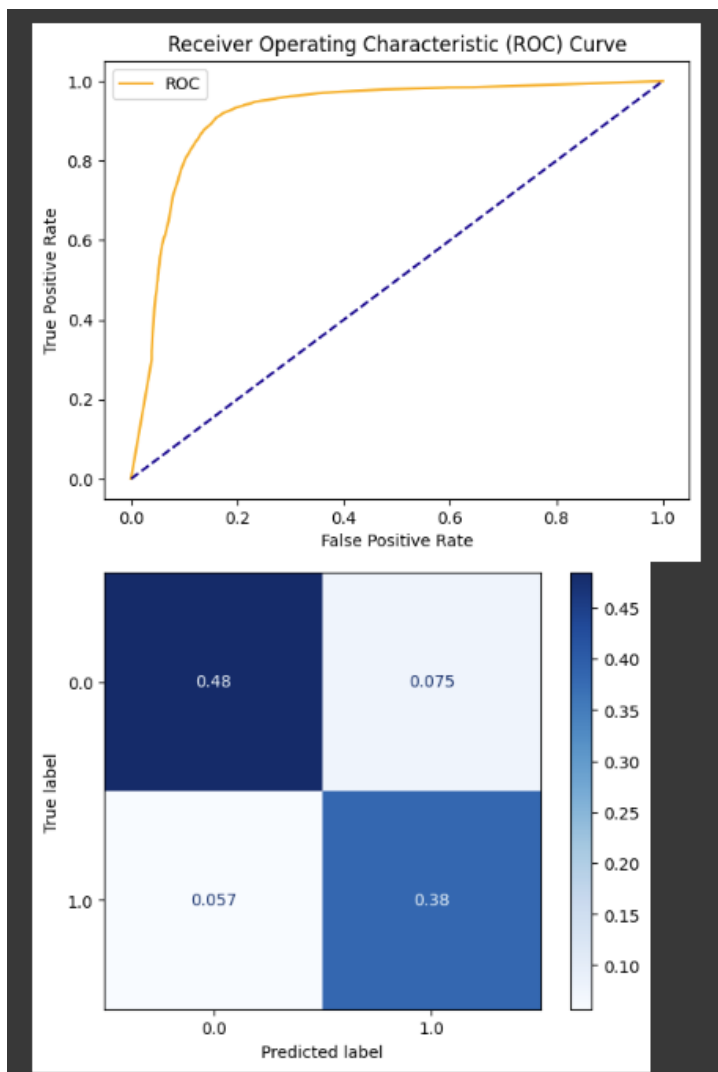
Confusion matrix are shown in the above images. ROC and AUC are given below

### 1] Logistic regression:



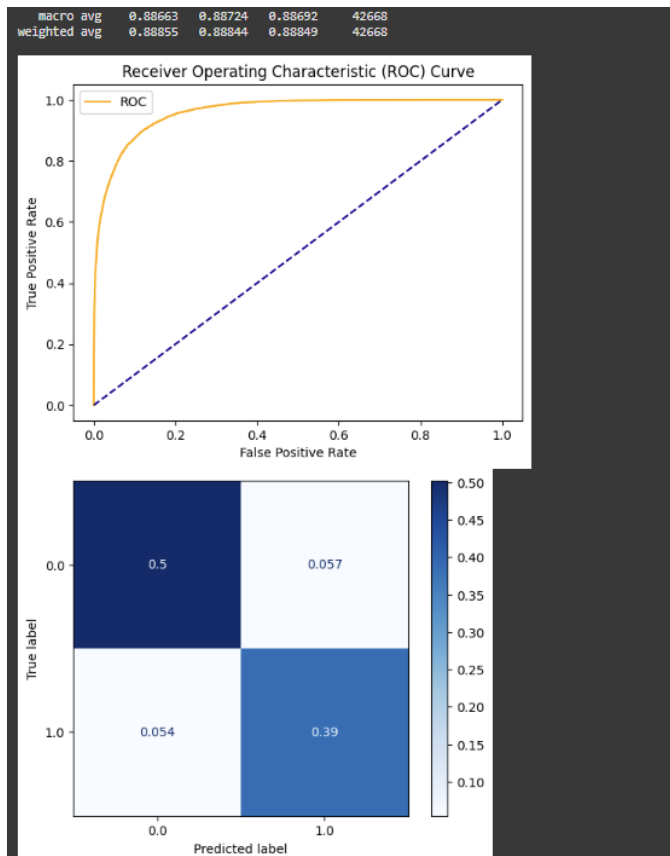
AUC : 0.79

## Decision Tree



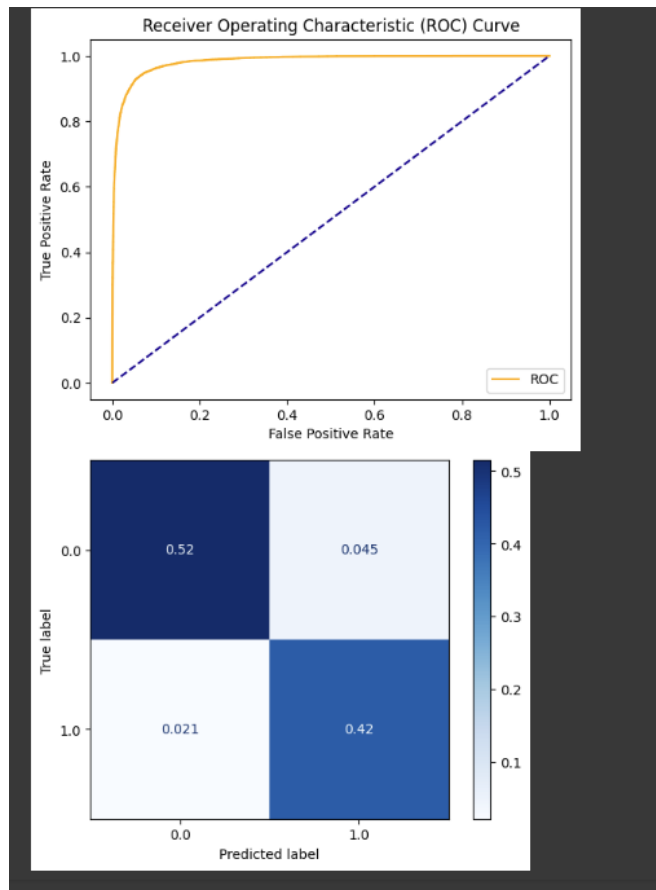
AUC : 0.86

## Neural Network



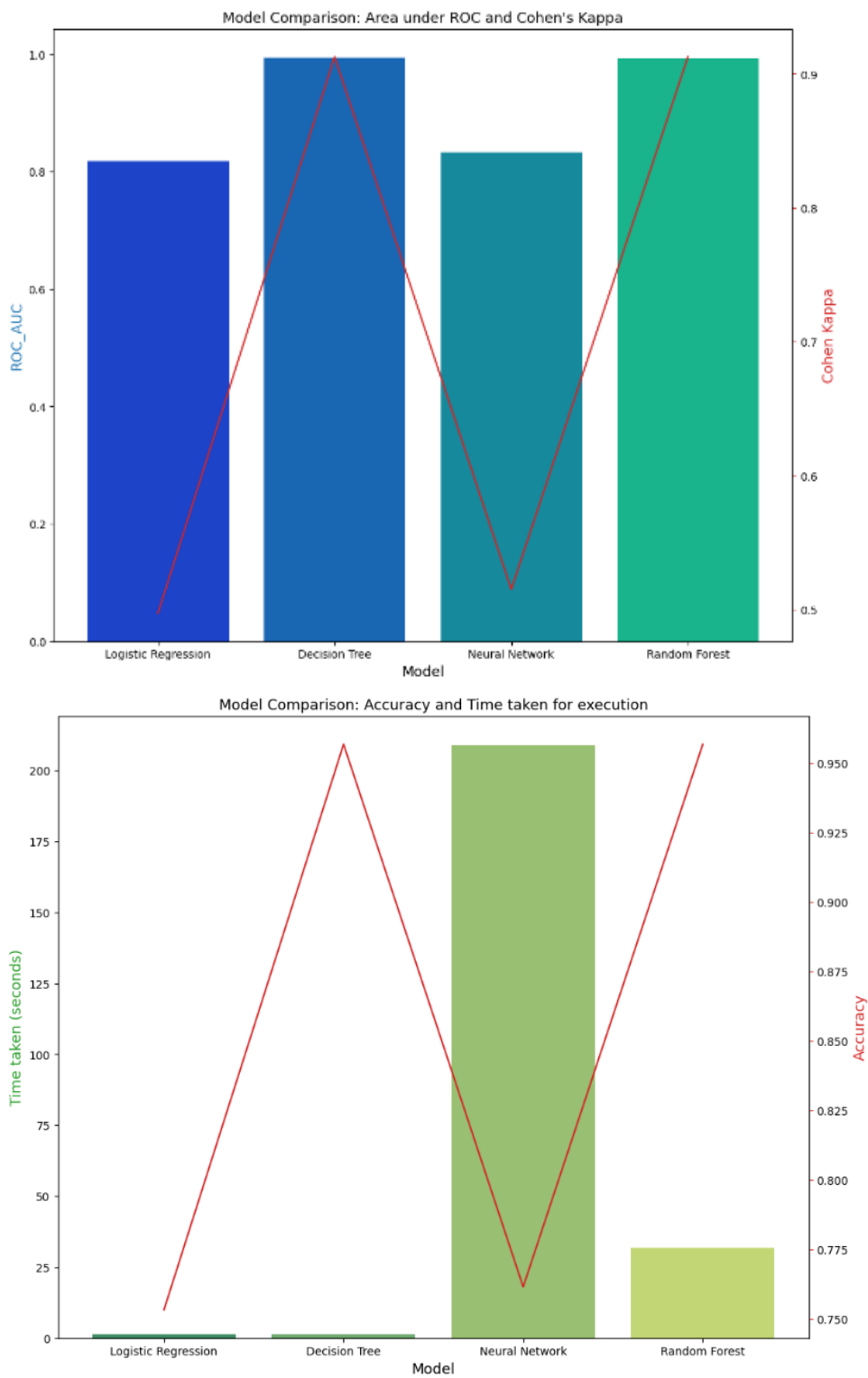
AUC : 0.88

## Random Forest



AUC : 0.93

Model Comparison:



### **Performance Analysis:**

The performance of each model is discussed, except for Random forests, which show the most accuracy and are particularly useful when handling disparate data. The problem of overfitting, as well as modifications to combat overfitting, will also be discussed.

### **Conclusion and Future work:**

The project effectively utilized various machine learning algorithms for rainfall prediction, with logistic regression emerging as the most effective method. The results underscore the significance of algorithm selection and model performance in enhancing predictive accuracy. These findings demonstrate the potential of machine learning in addressing real-world challenges related to weather forecasting and decision-making.

**Future improvements:** Future research could focus on hybrid models that combine the advantages of different algorithms or explore the use of deep learning technology to further enhance discovery.

## **6. Contribution**

Aneesh Thake contributed to this project by implementing the standard procedures and conducting the training experiments. He was also responsible for writing the final report, ensuring that it accurately reflected the project's findings. His work involved reviewing and revising the document to achieve a clear presentation of the analysis.

## **References**

1] C. Z. Basha, N. Bhavana, P. Bhavya and S. V, "Rainfall Prediction using Machine Learning & Deep Learning Techniques," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 92-97, doi: 10.1109/ICESC48915.2020.9155896. keywords: {Artificial neural networks;Forecasting;Biological neural networks;Predictive models;Machine learning;Ocean temperature;Meteorology;Rainfall;Prediction;Artificial Neural Networks;Deep Learning},

2] S. Biruntha, B. S. Sowmiya, R. Subashri and M. Vasanth, "Rainfall Prediction using kNN and Decision Tree," 2022 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2022, pp. 1757-1763, doi: 10.1109/ICEARS53579.2022.9752220. keywords: {Renewable energy sources;Machine learning algorithms;Rain;Urban areas;Weather forecasting;Prediction algorithms;Decision trees;Rainfall Forecasting;Machine Learning;kNN algorithm;Decision Tree;Classification;Regression},

<https://www.javatpoint.com/rainfall-prediction-using-ml#:~:text=In%20predicting%20rainfall%2C%20SVMs%20can,to%20predict%20future%20rainfall%20patterns.&text=Random%20Forests%3A%20An%20ensemble%20learning,decision%20trees%20to%20produce%20predictions.>