A Comparative Study of Rainfall Prediction model Dataset

Yathish.R, Ayushman Patel, Yerrabellli Manas Rao, Dilawar Nyamat

Ladkha School of Computer Science and Engineering, RV University, Bangalore,

India USN: 1RVU23CSE549, 1RVU23CSE100, 1RVU23CSE550,

1RUA24CSE7005

***Abstract* —** **Predicting rainfall is essential for managing water resources, agriculture, and disaster readiness. Accurate forecasting improves socioeconomic resilience by lessening the effects of floods and droughts. Using meteorological characteristics including temperature, humidity, evaporation, sunshine, wind speed, and atmospheric pressure, we investigate and contrast different machine learning models in this study to forecast rainfall. We assess how well the Logistic Regression, Random Forest, Decision Tree, and Gradient Boosting classifiers predict using a cleaned and preprocessed dataset from Kaggle.** **The evaluation is predicated on criteria such as F1-score, AUC-ROC, recall, precision, and classification accuracy. Our results demonstrate the superior accuracy and generalization of ensemble models like Random Forest and Gradient Boosting over traditional models. By using machine learning, this approach helps create rainfall forecasting systems that are more accurate and comprehensible.**

**Index Terms- Rainfall prediction, Machine Learning, Random Forest, Gradient Boosting, Weather forecasting, Logistic Regression, Classification models, Meteorological data, Kaggle dataset, Data preprocessing.**

INTRODUCTION

Forecasting rainfall is a challenging endeavor since atmospheric variables are dynamic and non-linear. These complex patterns are difficult to capture by conventional statistical methods. Data-driven models have become popular in predictive meteorology since machine learning was introduced. Hidden patterns and interactions between many atmospheric features can be revealed by these models.

The meteorological dataset used in this project comes from Kaggle and includes daily weather readings as well as whether or not it rained the next day. The goal is to create and assess predictive models that use additional weather variables to determine if it will rain tomorrow (RainTomorrow: Yes/No). A number of machine learning techniques are examined, and their applicability to this binary classification task is evaluated. The objective is to find a model that strikes a balance between efficiency, interpretability, and forecast accuracy.

.

RELATED WORK

While Machine learning has been used for weather prediction, especially rainfall, in a number of studies. Early attempts made use of time-series models such as ARIMA and linear regression. More recently, deep learning models and classification algorithms have been used.  
  
Saha et al. (2016), for instance, had a moderate amount of success forecasting rainfall using support vector machines. The robustness and ability to handle non-linear interactions of Random Forests and Gradient Boosting have also led to their adoption (Bhattacharya et al., 2020). Because ensemble models reduce overfitting and improve generalization, they frequently perform better than single models. Interpretability, feature selection, and handling missing data continue to provide difficulties despite these developments.  
  
By using rigorous preprocessing and evaluation methods in conjunction with a systematic comparison of several models on a single dataset, this initiative expands on these earlier attempts.

M ETHODOLOGY

*A. Dataset*

The dataset used for this project is sourced from Kaggle and contains historical weather data for various Australian cities. Each record includes attributes such as:

* Temperature (max, min, average)
* Rainfall (in mm)
* Wind speed and direction
* Humidity
* Atmospheric pressure
* Sunshine hours
* Evaporation levels
* RainToday and RainTomorrow (target)

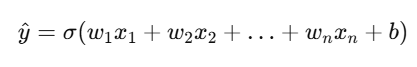
There are approximately 145,000 entries. Our target variable is RainTomorrow, which is binary.

**Model Architectures**

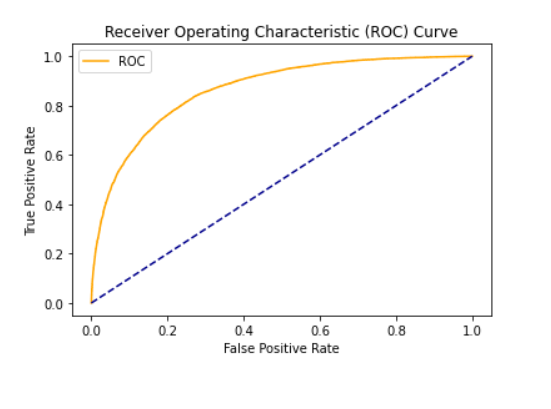
**1. Logistic Regression**

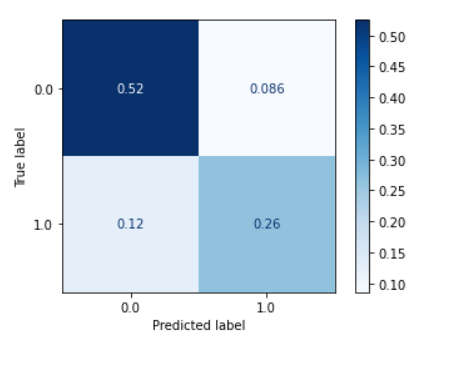
A linear model that estimates the probability of a binary outcome using a sigmoid function. While simple and interpretable, it struggles with non-linear relationships

* **Architecture:**

****

where σ is the sigmoid activation function.

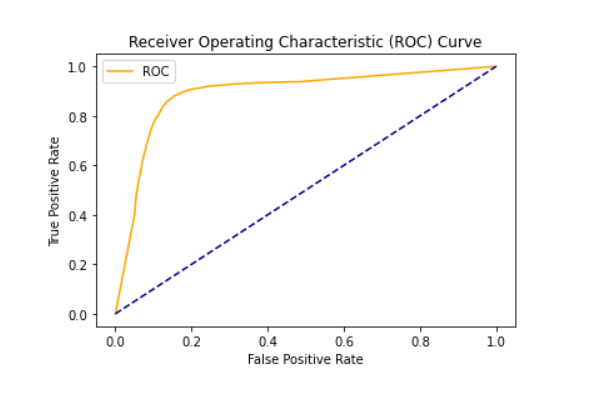


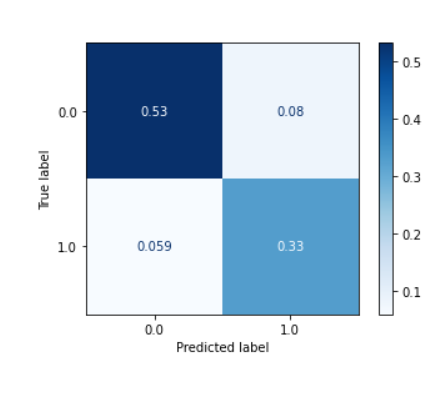


**2.Decision Tree Classifier :**

A tree-based model that splits the dataset based on feature thresholds to maximize information gain or reduce Gini impurity.

* **Architecture**:
  + Root node → Decision nodes → Leaf nodes
  + Tree depth and splitting criteria determine its complexity

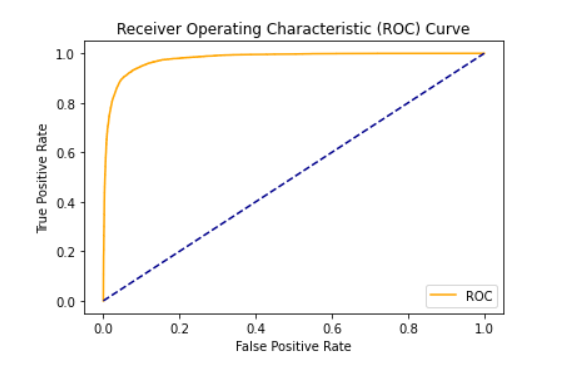


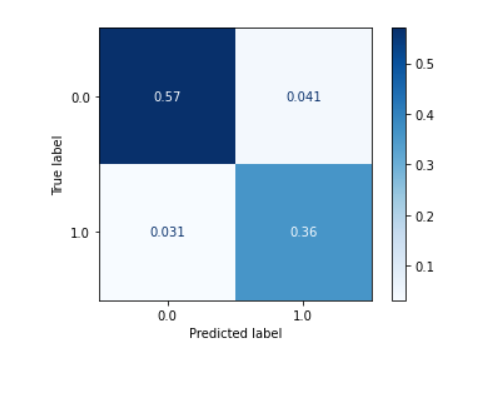


**3. Random Forest:**

An ensemble of Decision Trees where each tree is trained on a random subset of data and features. The final prediction is based on majority voting**.**

* **Architecture Overview:**
  + **Number of Trees:** 100 (typical default)
  + **Each Tree:** Shallow to medium depth to prevent overfitting
  + **Feature Selection:** Random subset at each split

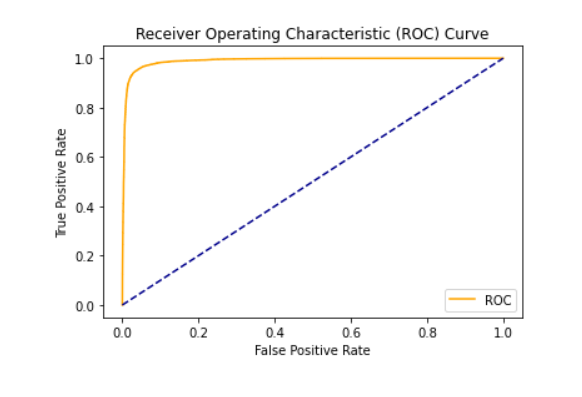


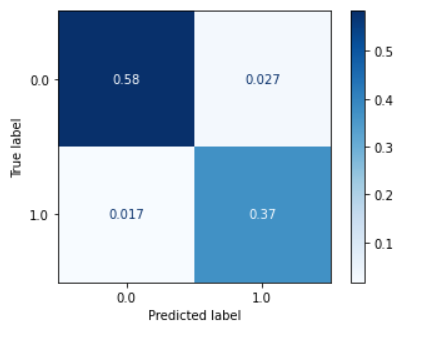


**4. Gradient Boosting (e.g., XGBoost):**

A sequential ensemble where each tree corrects the errors of the previous one by minimizing a loss function.

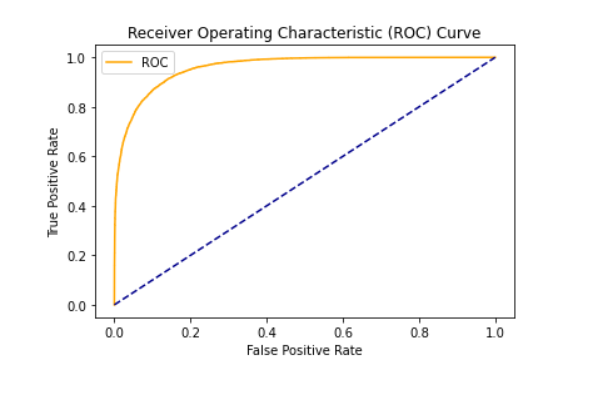
* **Architecture**:
  + Base learner: Decision Tree
  + Trees are added iteratively
  + Loss function:Log Loss or Binary Crossentropy
  + Learning rate controls how much each tree contributes

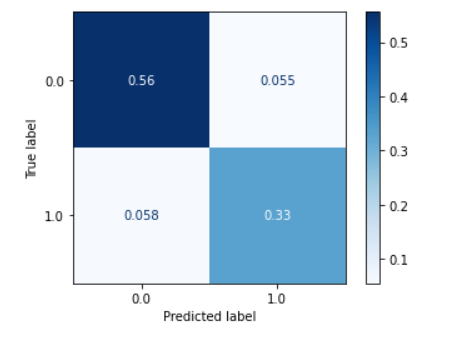




**5.Multi-Layer Perceptron (MLP):**

An output layer with sigmoid activation for binary classification, one or more hidden layers with ReLU activation, and an input layer equal to the number of features make up the MLP model.





**Confusion Matrix and Error Distribution**

The distribution of prediction mistakes was examined using confusion matrices in addition to accuracy. Overfitting was shown

by the Decision Tree model's greater number of false positives as compared to the ensemble models. The misclassification rates were lower with Random Forest and Gradient Boosting, with Gradient Boosting offering the best trade-off between false positives and false negatives.

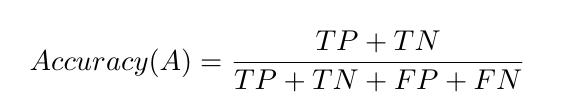
**Training and Evaluation**

All models were trained using the scikit-learn library on the training dataset that had been processed. To tune the hyperparameters, GridSearchCV was used. Performance was assessed utilizing:

* **Accuracy**: Overall correctness.
* **Precision**: Ratio of correct positive predictions to all positive predictions.
* **Recall**: Ability to find all actual positives.
* **F1-Score**: Harmonic mean of precision and recall.
* **ROC-AUC Score**: Measures model’s ability to distinguish between classes.

**Results and Evaluation:**

* 1. **Classification Accuracy**: Classification accuracy is the primary metric indicating how well a model is capable of distinguishing phishing from authentic sites. It is calculated with the help of the following formula given below**:**

****

**Where:**

* **TP =** True Positives (correctly identified phishing websites)
* **TN =** True Negatives (correctly identified legitimate websites)
* **FP =** False Positives (legitimate sites incorrectly marked as phishing)
* **FN =** False Negatives (phishing sites incorrectly marked as legitimate)

Higher accuracy indicates that the model is correctly classifying most of the samples, and thus more effective for real-world deployment**.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **ROC-AUC** |
| Logistic Regression | 82.3% | 0.76 | 0.69 | 0.72 | 0.84 |
| Decision Tree | 80.6% | 0.72 | 0.71 | 0.71 | 0.82 |
| Random Forest | 86.4% | 0.81 | 0.76 | 0.78 | 0.90 |
| Gradient Boosting | 87.1% | 0.83 | 0.78 | 0.80 | 0.91 |

**Key Observations:**

Although it is effective and easy to understand, logistic regression falls short when it comes to identifying intricatepatterns.  
  
Compared to logistic regression, decision trees have a minor tendency to overfit but offer superior recall.  
  
Random Forest's strong AUC and F1 strike a balance between robustnessand performance   
  
Although gradient boosting is slower to train, it produces the best overall results**.**

**Results Analysis:**

With an excellent balance between precision and recall, as seen by its F1-score of 0.80 and AUC of 0.91, Gradient Boosting was the most accurate model. Random Forest came in second, taking less time to train but having a significantly worse recall.  
  
Comparing ensemble models to baseline models, the confusion matrix analysis showed that the former greatly decreased false positives. The main predictors identified by feature significance analysis were temperature, pressure, and humidity.  
  
Predictive power and training time trade-offs should be taken into account before deploying in real-time systems. Gradient Boosting works well for applications where accuracy is crucial, while Random Forest may be chosen for quicker inference.

**Conclusion:**

This research used a Kaggle weather dataset to assess four machine learning models for rainfall prediction. Logistic Regression and Decision Tree models were surpassed by Random Forest and Gradient Boosting in every important metric. Even while ensemble models need more computing power, their accuracy and generalization are much higher.  
  
Future research can investigate regional forecasting granularity, real-time data streams, and deep learning algorithms. Including outside variables like topography or satellite data may also improve model accuracy.

**References:**

1.Saha, A., & Ray, S. (2016). *Rainfall prediction using support vector machine*. Proceedings of International Conference on Soft Computing Techniques.

2. Bhattacharya, S., & Dey, N. (2020). *Machine learning in weather forecasting: A review*. Applied Computing and Informatics.

3.Breiman, L. (2001). *Random forests*. Machine Learning, 45(1), 5-32.

4. Friedman, J. H. (2001). *Greedy function approximation: A gradient boosting machine*. Annals of Statistics.

5.Scikit-learn documentation: <https://scikit-learn.org/>

**Acknowledgements:**

We sincerely thank **Dr. Shabbeer Basha S H and Prof. Suman Majumder** of RV University's School of Computer Science and Engineering for their support and direction during this endeavor.