

**Ahsanullah University of Science and Technology**

**Department of Computer Science and Engineering**

**CSE 4108**

**Artificial Intelligence Lab**

**Project Name: Weather Forecast Prediction**

Submitted To

**Md. Siam Ansary**

Lecturer, CSE, AUST

**Md. Ashek Seum**

Lecturer, CSE, AUST

Submitted By

**Inzamum Ul Haque**

17.02.04.087

**Description of the Problem**

Artificial Intelligence is developing rapidly at all sectors of the industry and with the use of this we can solve or predict many aspects for a given dataset by using AI models. In this project, I have tried to predict the weather forecast for a given dataset. Its mainly a classification problem with predicting whether it is rain or foggy or thunderstorm for a given area. Here I have tried to predict this with some features.

**Dataset**

Mainly I have used Boston Weather Dataset as classification dataset for the prediction model. The dataset has a total of 19 columns and 366 rows including the target column.

The target column is the Events column which we have to predict using the features as the rest of the columns. The events column has 8 variants of predictive answers which are 1. Fog, 2. Fog, Rain, 3. Fog, Rain, Thunderstorm, 4. No rain, 5. Rain, 6. Rain, Snow, 7. Rain, Thunderstorm, 8. Thunderstorm. These answers should predict the weather forecast for the day.

Rest of the remaining columns (except the date column) are used as features to predict the answers. For this, I have taken Temperature, Dew Point, Humidity, Sea Level Pressure, Wind Gust and their respective high, low and average values. These features are independent as we have taken it for X and the events column is the dependent column as Y.

**Model Description**

For predictive modeling of the features, I have used 2 classification algorithms**. Random Forest Classifier and Decision Tree**.

**Random forest** is a supervised learning algorithm. It can be used for both classification and regression. It is also the most flexible and easy to use algorithm. Random forest classifier selects decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of feature importance.

**Decision tree** is a flowchart like tree structure where an internal node represents feature, the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value and partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps us in decision making. It’s visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.

There are some comparisons regarding Random Forest and Decision Tree algorithm.

**Random Forests vs Decision Trees**

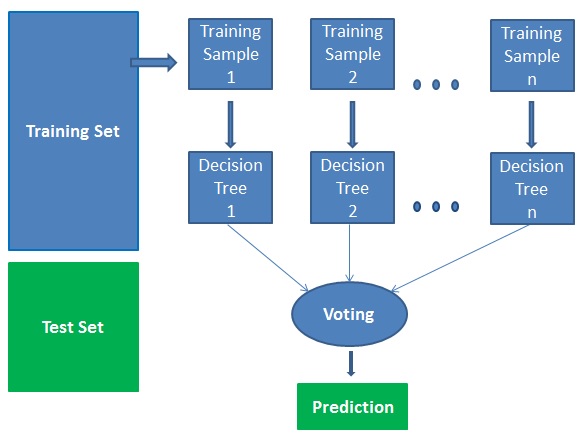
1. Random forest is a set of multiple decision trees.
2. Deep decision trees may suffer from overfitting, but random forest prevents overfitting by creating trees on random subsets.
3. Decision Trees are computationally faster.
4. Random forest is difficult to interpret, while a decision tree is easily interpretable and can be converted to rules.

**Performance Comparison**

Let’s look at the Random Forest Classifier first.

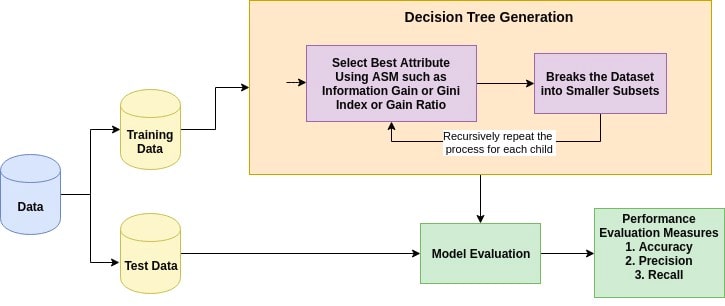
First, we separated the dependent and the independent columns for X and Y and took all the rows for selection. We train the dataset for X\_train and Y\_train and then test the dataset for X\_test and Y\_test splitting the values. We set the test size for 0.6 meaning it will take 60% of the total dataset.

Here the X\_train set takes some random data samples for training as it has to predict the model based on some features, construct a decision tree and predict the results from there. We can set the value for constructing maximum number of trees for the model using n\_estimators as parameter, setting it a value of 100. It means it can construct up to 100 levels of node for taking decision. We can increase the level of nodes of the tree to 150 or 200 to increase the accuracy of Random Forest, but it will increase only small amount of accuracy, rather it will increase the computing time of the algorithm. That’s why I have taken n\_estimators as 100.



**Fig: Working structure of Random Forest**

Decision tree algorithm shares a decision-making logic. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions.



**Fig: Working Structure of Decision Tree**

Here we construct the same splitting of data for decision tree as we did for Random Forest. We take the test size for 0.6 and pass it to the Decision tree classifier function for modeling and predicting.

**Metric Score Comparison**

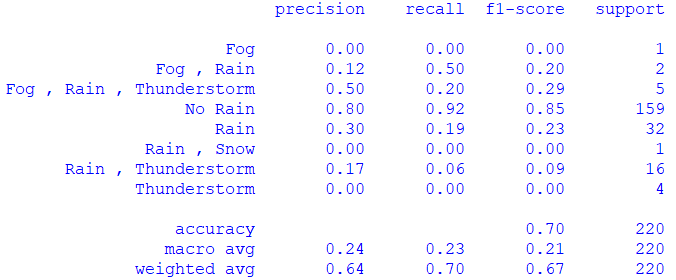
After modeling and predicting the data, the model gives us a metric or performance score about how well our algorithm has performed. Here we have used 4 performance metrics. Accuracy, Mean Absolute Error, Mean Squared Error and Root Mean Squared Error. The two model gives us different metric performance scores.

|  |  |  |
| --- | --- | --- |
|  | **Random Forest** | **Decision Tree** |
| **Accuracy** | 70.45 | 59.54 |
| **MAE (Mean Absolute Error)** | 62.27 | 86.36 |
| **MSE (Mean Squared Error)** | 1.7136 | 2.2091 |
| **RMSE (Root Mean Squared Error)** | 1.3091 | 1.4863 |

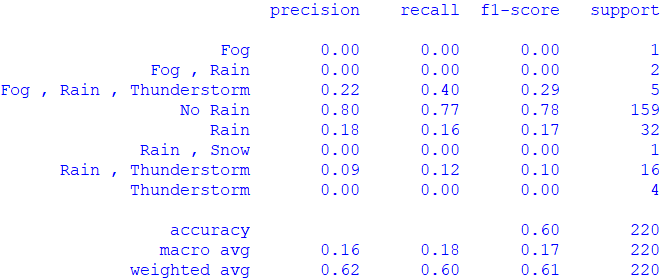
The table shows a brief comparison of how the 2 models are performing with the same dataset and their accuracy score as well as their errors. We can see the accuracy Random Forest is pretty much higher than Decision Tree, that can tell Random Forest is performing better than Decision tree. We can also see that the MAE, MSE and RMSE of Random Forest is also lower than Decision Tree, which means the error percentage of Decision Tree is higher than Random Forest. Random Forest can predict our data 70.45% whereas Decision Tree can predict 59.34%.

We can also show the classification report of each classification algorithm for each classification result. The classification report is the performance evaluation metric

to show the precision, recall, F1 score and support of the trained model.



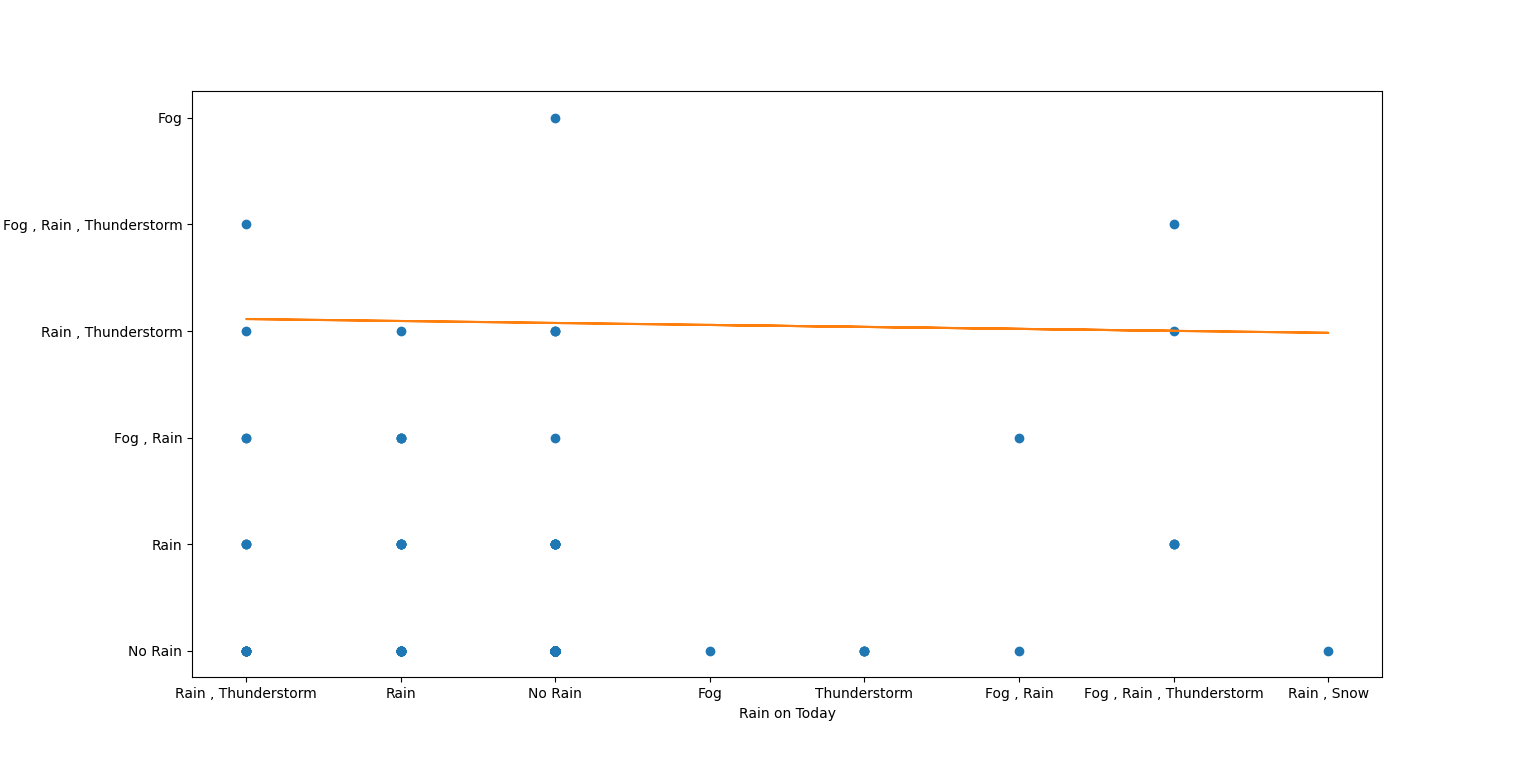
**Fig 1: Classification Report for Random Forest Classifier**



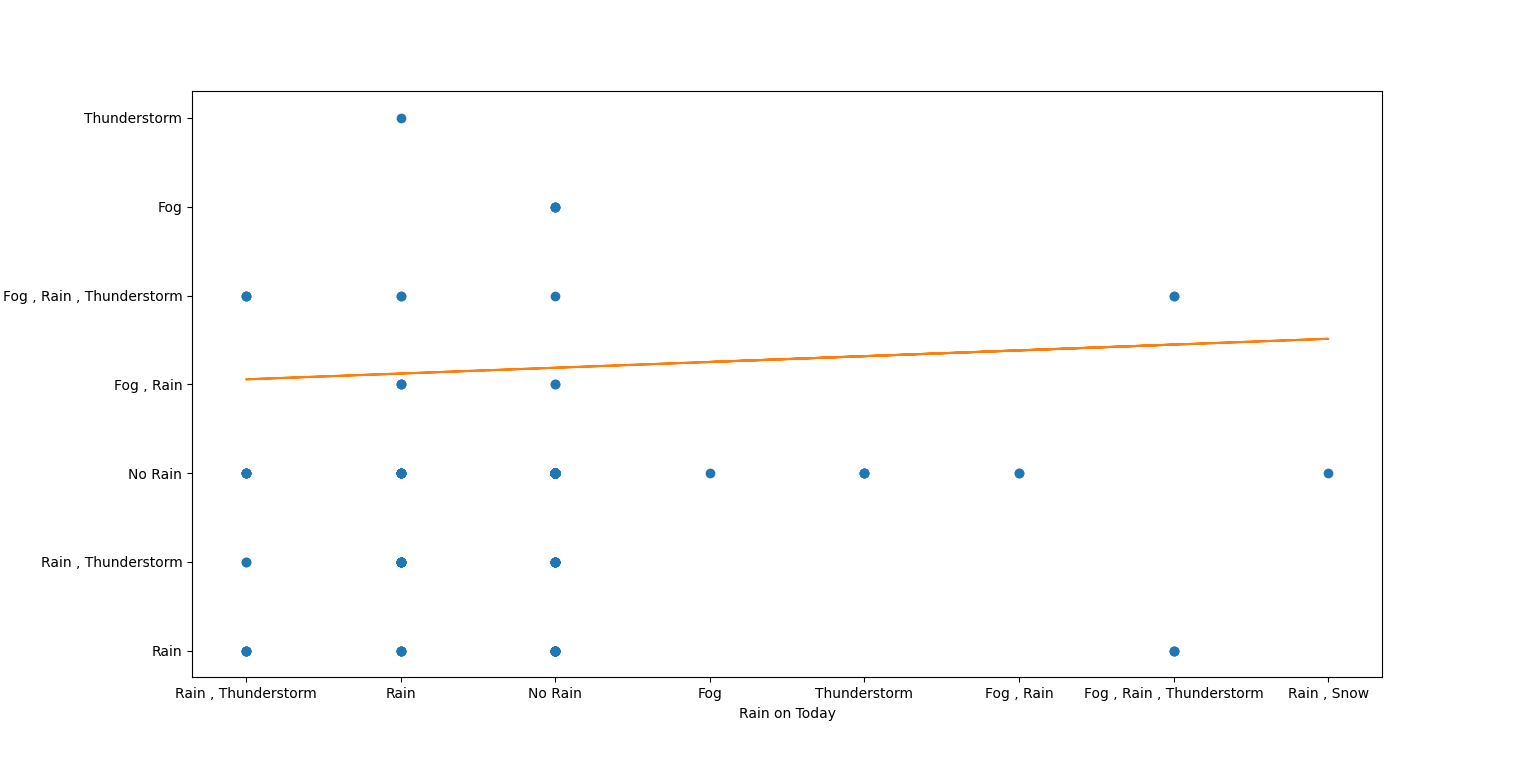
**Fig 2: Classification Report for Decision Tree**

**Graphical Comparison**

There is a final output file of Random Forest Classifier and Decision Tree to compare our target data and predicted data. Here is a graphical representation of how our predicted data differ from our actual data.



**Fig: Actual vs Predicted for Random Forest**



**Fig: Actual vs Predicted for Decision Tree**

**Discussion**

After analyzing the whole dataset, we can come to a conclusion that the Random Forest model is working much better than Decision Tree.

After working with the Random Forest and Decision Tree, I tried different things by increasing and decreasing the test size of the data. Each time I increase the test size, accuracy decreases. So, I took a decent test size for both of the model to better compare the results.

Random Forest works pretty smooth and compute better results for datasets. But the computation of Decision Tree is much faster than Random Forest. The accuracy of the Random Forest is 70%, that means the model is predicting 70% of the data correct, which is an ideal value. On the other hand, Decision Tree is giving an accuracy of 59% means the model is predicting 59% of the data correct. Moreover, the error rate of Decision Tree is considerably much higher than Random Forest. That’s why it’s giving 40% of the value false.

So, in compare to these 2 model for predictive dataset, we can say that Random Forest model is best suited than Decision Tree for this dataset.