**ADVANCED PYTHON AI AND ML TOOLS (AML 2203)**

**Topic: Rain Prediction Analysis**

**Final Project**

**Submitted to:**

**Vahid Hadavi, PhD, P. Eng.**

**Submitted by:**

Abhinav Duvvuru - C0810417

Sandeep Reddy Alla - C0824160

Sivapriya Tatireddy - C0817271

Uma Maheswararao Gutta - C0814571

**December 7, 2021**

Table of Contents

[1. Introduction: 3](#_Toc89643017)

[2. Data Description: 3](#_Toc89643018)

[Figure 1. Importing required libraries 3](#_Toc89643019)

[3. Data Preprocessing: 4](#_Toc89643020)

[3.1. Loading Dataset: 4](#_Toc89643021)

[Figure 2. Reading dataset 4](#_Toc89643022)

[Figure 3. Shape of dataset 4](#_Toc89643023)

[Figure 4. Summary of dataset 4](#_Toc89643024)

[Figure 5. Descriptive statistics of dataset 5](#_Toc89643025)

[3.2. Handling Missing Values: 5](#_Toc89643026)

[Figure 6. Missing values of attributes 5](#_Toc89643027)

[Figure 7. Dropping attributes 6](#_Toc89643028)

[4. Exploratory Data Analysis: 6](#_Toc89643029)

[4.1. Exploring Target Variable: 6](#_Toc89643030)

[Figure 8. Plot of target variable “Rain Tomorrow” 6](#_Toc89643031)

[4.2. Correlation Between Numeric Attributes: 7](#_Toc89643032)

[Figure 9. Heatmap of attributes 7](#_Toc89643033)

[4.3. Analysing Categorical Data: 7](#_Toc89643034)

[Figure 10. Categorical data 8](#_Toc89643035)

[Figure 11. One hot encoding of attributes 8](#_Toc89643036)

[5. Model Building and Evaluation: 8](#_Toc89643037)

[Figure 12. Decision tree classifier 8](#_Toc89643038)

[Figure 13. Classification report and accuracy of Decision Tree Classifier 9](#_Toc89643039)

[Figure 14. Accuracy of Random Forest Classifier 9](#_Toc89643040)

[6. Hyper Parameter Tunning: 9](#_Toc89643041)

[Figure 15. Hyper parameter tunning 10](#_Toc89643042)

[Figure 16. Accuracy score with varying max\_depth parameter 10](#_Toc89643043)

[7. Conclusion: 11](#_Toc89643044)

[8. References: 11](#_Toc89643045)

# 1. Introduction:

Rainfall prediction has gained most research relevance in recent times. In this project we use machine learning algorithms to predict whether or not it will rain tomorrow in Australia for that we first pre-process the data and handle any missing values and train the data to perform Decision Tree classifier and Random Forest classifier with the help of scikit learn and tuned the hyper parameters with grid search and will conclude with which model performs the best with good accuracy.

# 2. Data Description:

In this report, we consider the data from the Kaggle, and it contains almost 10 years of daily weather observations from many locations across Australia. Below are the basic details about dataset:

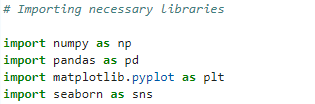
Number of attributes: 23

Number of instances: 145460

Number of independent variables: 22

Number of dependent variable/Target variable: 1

As a first step of data analysis, we imported all the necessary libraries as shown in Figure 1



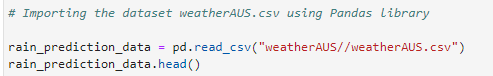
### Figure 1. Importing required libraries

# 3. Data Preprocessing:

We perform data pre-processing to make sure our data is clean without any missing values and noise so that we can feed to the machine learning model. Below are the steps followed:

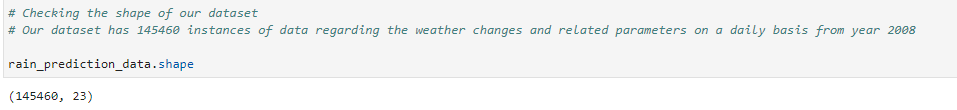
## 3.1. Loading Dataset:

Dataset is loaded using pandas read\_csv method as shown in below Figure 2.



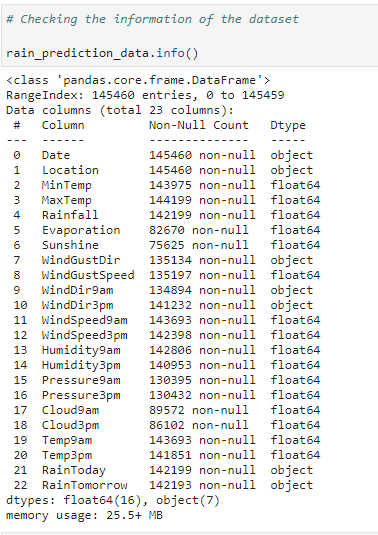
### Figure 2. Reading dataset

The dataset has around 145,000 instances and 23 attributes as shown in Figure 3.



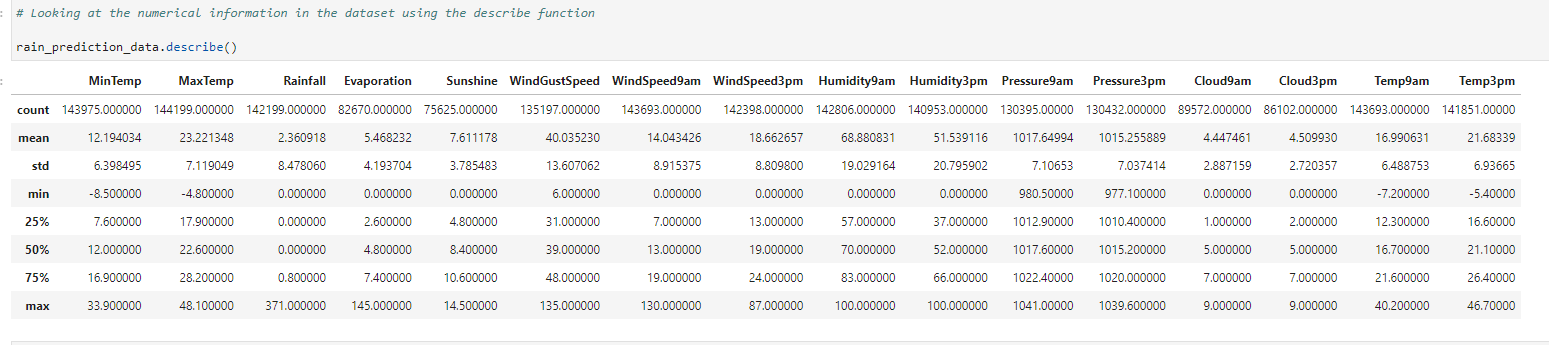
### Figure 3. Shape of dataset

The info() method in pandas gives the information about the dataset, from the below Figure 4, our data consists of two types: float64 and object



### Figure 4. Summary of dataset

The descriptive statistics of the dataset like mean, count, max, standard deviation, etc. can be obtained by using describe() method (Figure 5)

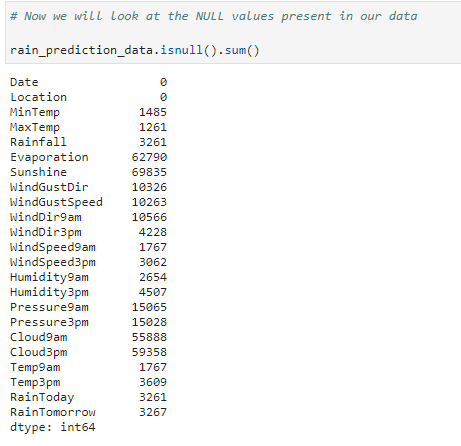


### Figure 5. Descriptive statistics of dataset

There is total 16 numerical and 7 categorical attributes.

## 3.2. Handling Missing Values:

Total number of missing values in our dataset can be identified by using isnull().sum() method as shown in Figure 6.



### Figure 6. Missing values of attributes

From the above, we can see that almost all the columns of our data have null values. Only the Date and Location columns have No NULL values.

We can also notice that columns like "Evaporation", "Sunshine", "Cloud9am" and "Cloud3pm" have nearly half the data as Null. We can drop those columns from our data for further analysis.

Imputation of missing data is not possible for those columns as NULL values are dominating them. Dropping the columns "Evaporation", "Sunshine", "Cloud9am" and "Cloud3pm" from our dataset



### Figure 7. Dropping attributes

# 4. Exploratory Data Analysis:

Exploratory Data Analysis is used to analyse and visualize the data and find trends and patterns and relationships in data, which helps in making better decisions.

## 4.1. Exploring Target Variable:

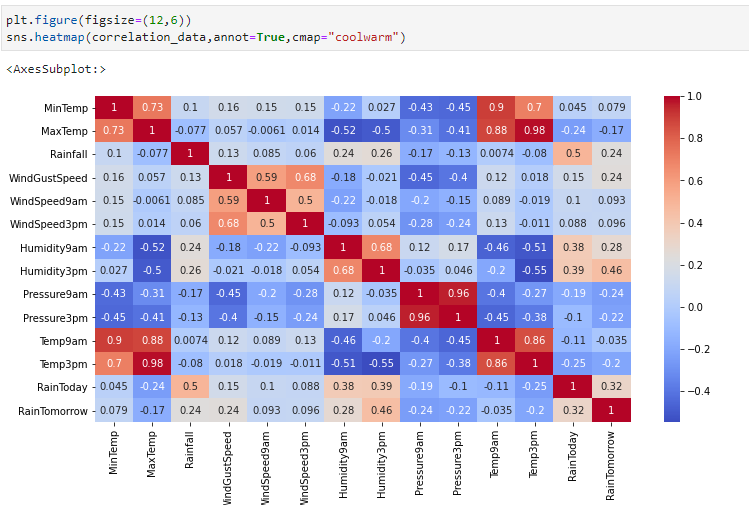
The target variable “Rain Tomorrow” has majority of ‘No’ values. If the data is imbalanced then it might decrease the performance of the model. It doesn’t make any sense when we try to balance the target variable, because the truthfulness of data might decrease. So, let’s keep it as it is (Figure 8)

### Figure 8. Plot of target variable “Rain Tomorrow”

## 4.2. Correlation Between Numeric Attributes:

## 

Correlation helps to measure the strength of relationship between features both positively and negatively. corr() method in pandas is used to find correlation among attributes as shown in Figure 9.



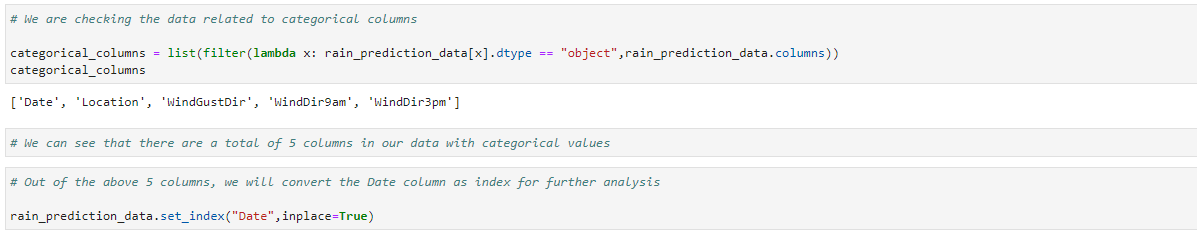
### Figure 9. Heatmap of attributes

We can see from the above heatmap that the correlation between “RainToday” and “Rain Tomorrow” is 0.32. This indicates that there are 32% probability or chance of raining tomorrow if there is rain today.

We can also observe that the parameters like MinTemp, WindSpeed9am, WindSpeed3pm, Temp9am columns are very less correlated to our RainTomorrow attributes.

## 4.3. Analysing Categorical Data:

From the below Figure 10, there are total 5 categorical attributes. The Date column is set as index for further analysis.



### Figure 10. Categorical data

Most of the machine learning algorithms cannot handle categorical data. Hence, we need to convert these categorical data into numerical data. For that we use one hot encoding technique as shown in Figure 11.



### Figure 11. One hot encoding of attributes

To normalize the features, we used standard scaler scaling technique.

# 

# 5. Model Building and Evaluation:

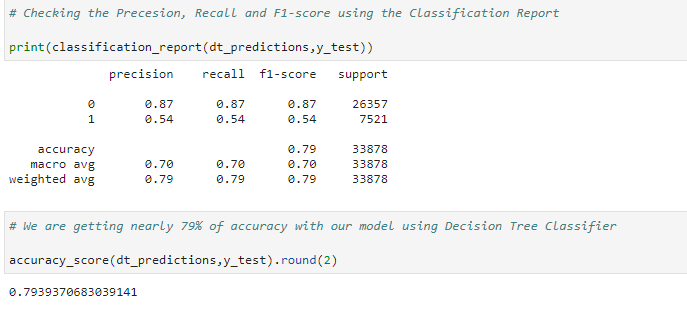
The entire dataset is split into test and train data with test data as 30% of whole data. And then applied Decision Tree Classifier to build our model as shown in Figure 12.

Graphical user interface, text, application, email

Description automatically generated

### Figure 12. Decision tree classifier

The classification\_report package from sklearn metrics gives information about F-1 score, precision and recall (Figure 13)



### Figure 13. Classification report and accuracy of Decision Tree Classifier

As the Accuracy is 79%, we applied Random Forest Classifier to check for better accuracy.

Graphical user interface, text

Description automatically generated

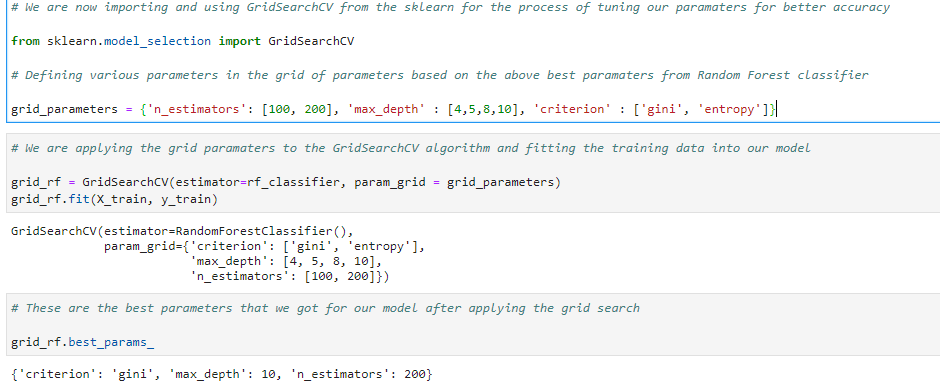
### Figure 14. Accuracy of Random Forest Classifier

We are getting nearly 86% of accuracy with our model using Random Forest Classifier

# 6. Hyper Parameter Tunning:

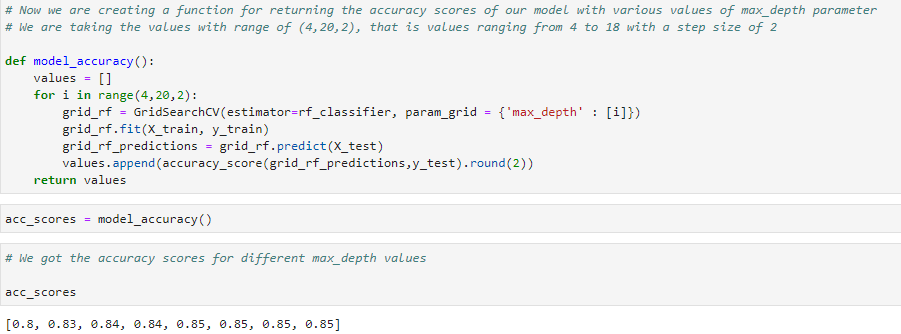
To increase our model performance, we performed hyperparameter tunning technique. Finding the optimal hyperparameters might help us to achieve best-performing model.

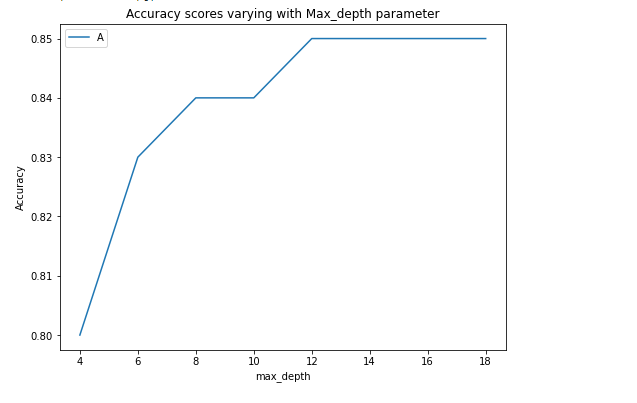
Here we are using GridSearchCV from the sklearn for the process of tuning our paramaters for better accuracy as shown in Figure 15.



### Figure 15. Hyper parameter tunning

Let us compare the varying accuracy scores with “max\_depth” parameters.





### Figure 16. Accuracy score with varying max\_depth parameter

It is evident from the above graph (Figure 16) that the accuracy is stable after having the “max\_depth” parameter as 12 and so on. We are getting 85-86 % of accuracy for our model using the Random Forest Classifier.

# 7. Conclusion:

From the above results performed on various classifiers, it is clear that the Random Forest Classifier (86% Accuracy) is performing much better than the Decison Tree Classifier (79% Accuracy). In addition to that, we can also see that using the hyperparameter tuning, our model is stable and is clearly getting accuracy of 85-86% with various parameters.

The final accuracy of our model is 85.7% which is nearly 86%. This is very good accuracy, but our model needs to be improved in terms of predicting the rainfall. It is performing very good in terms of predicting that there is going to be no Rain tomorrow with an F1-Score of 91% but in the instances of Rain tomorrow our model is not reaching expectations with F1-Score of only 61%.

In conclusion, overall, our model is performing very well with an accuracy of 86% but needs to improve in predicting the possible Raining instances.

# 8. References:

Daily Weather Observations (n.d.), Bureau of Meteorology Australian Government. Retrieved from <http://www.bom.gov.au/climate/dwo/>

Climate Data Online (n.d.), Bureau of Meteorology Australian Government. Retrieved from <http://www.bom.gov.au/climate/data/>

Jeremy Jordan (2017). Hyperparameter tuning for machine learning models. Retrieved from <https://www.jeremyjordan.me/hyperparameter-tuning/>

Random Forest Classifier (n.d.), scikit learn. Retrieved form <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>