**Rainfall Weather Forecasting**



Weather forecasting is the application of science and technology to predict the conditions of the atmosphere for a given location and time. Weather forecasts are made by collecting quantitative data about the current state of the atmosphere at a given place and using meteorology to project how the atmosphere will change.

We have a Rain Dataset which is used to predict whether or not it will rain tomorrow. The Dataset contains about 10 years of daily weather observations of different locations in Australia.

1. **Project Description**

Design a predictive model with the use of machine learning algorithms to forecast whether or not it will rain tomorrow.

**Dataset Description:**

The description of the dataset gives important details about the data, such as the units of measurement, any unique considerations or restrictions, and the format and meaning of each variable. Prior to conducting any analysis, make sure you have a complete knowledge of the context and structure of the data by carefully reading the dataset description. Throughout the analysis process, having this understanding facilitates accurate interpretation of the results and well-informed decision-making. It also aids in locating any biases, mistakes, or inconsistencies in the data that can affect the results of the research. Analysts can select relevant analytical tools, deal with missing or inaccurate data effectively, and steer clear of misinterpretations that could result in poor decisions or findings by being familiar with the dataset description.

Number of columns: 23

Date - The date of observation

Location - The common name of the location of the weather station

MinTemp - The minimum temperature in degrees Celsius

MaxTemp - The maximum temperature in degrees Celsius

Rainfall - The amount of rainfall recorded for the day in mm

Evaporation - The so-called Class A pan evaporation (mm) in the 24 hours to 9am

Sunshine - The number of hours of bright sunshine in the day.

WindGustDir- The direction of the strongest wind gust in the 24 hours to midnight

WindGustSpeed -The speed (km/h) of the strongest wind gust in the 24 hours to midnight

WindDir9am -Direction of the wind at 9am

WindDir3pm -Direction of the wind at 3pm

WindSpeed9am -Wind speed (km/hr) averaged over 10 minutes prior to 9am

WindSpeed3pm -Wind speed (km/hr) averaged over 10 minutes prior to 3pm

Humidity9am -Humidity (percent) at 9am

Humidity3pm -Humidity (percent) at 3pm

Pressure9am -Atmospheric pressure (hpa) reduced to mean sea level at 9am

Pressure3pm -Atmospheric pressure (hpa) reduced to mean sea level at 3pm

Cloud9am - Fraction of sky obscured by cloud at 9am.

Cloud3pm -Fraction of sky obscured by cloud

Temp9am-Temperature (degrees C) at 9am

Temp3pm -Temperature (degrees C) at 3pm

RainToday -Boolean: 1 if precipitation (mm) in the 24 hours to 9am exceeds 1mm, otherwise 0

RainTomorrow -The amount of next day rain in mm. Used to create response variable. A kind of measure of the "risk".

**Dataset Link**

<https://raw.githubusercontent.com/dsrscientist/dataset3/main/weatherAUS.csv>

1. **Data Analysis**

Data analysis is the process of discovering useful information by cleaning of data, transformation of data and making it ready for modelling of dataset. Before analysis we need to read the dataset, which we have done my importing dataset using the pandas library. A number of procedures are included in data analysis with the goal of deriving significant insights from data. The first step is to obtain the dataset, after which it is cleaned, transformed, and examined to determine its features and structure. To maintain data integrity, errors, outliers, and missing numbers are dealt with during cleaning. Through variable reshaping, aggregation, or encoding, transformation readies the data for analysis. Visualizing and summarizing the data in order to spot trends, patterns, and correlations between variables is known as exploratory data analysis, or EDA. After that, the data is subjected to statistical and machine learning techniques in order to create descriptive or predictive models. In order to assist in decision-making, the analysis's findings are finally evaluated and shared with the relevant parties. Technical know-how, domain expertise, and critical thinking are all necessary for effective data analysis in order to derive useful insights that spur innovation and corporate value. It provides a framework for evidence-based problem-solving, process optimization, and decision-making in a variety of fields and sectors.

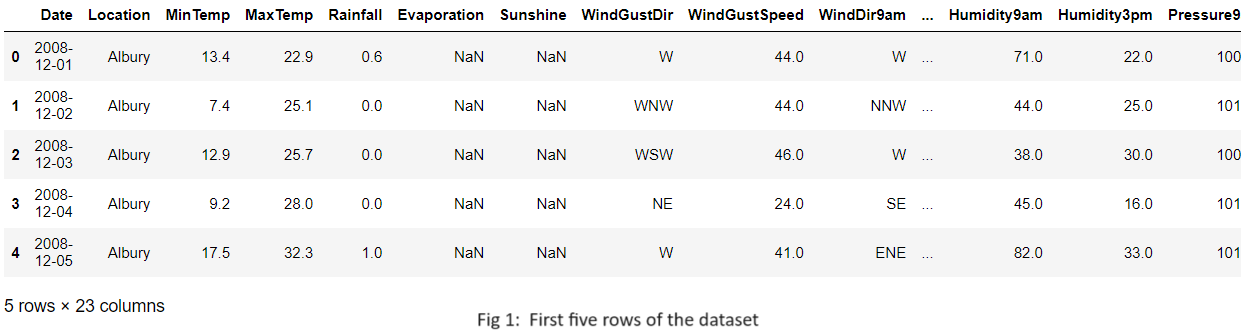
# Read the dataset

import pandas as pd

data = pd.read\_csv('weatherAUS.csv')

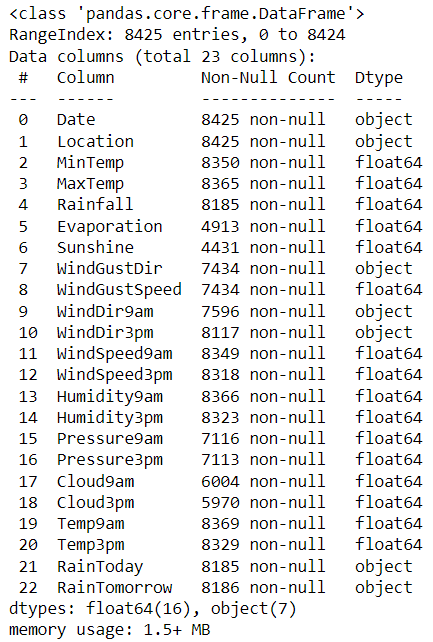
# First 5 rows of the dataset

` data.head()



# Data info

A brief overview of the dataset is given by data.info(), which also includes information about any missing values and the data types of each column.



# Shape of the dataset

(8425, 23)

1. **EDA**

The process of examining and analyzing record sets in order to recognize patterns, find outliers, and determine the relationships between variables is known as exploratory data analysis, or EDA. EDA is typically done as a first step before more formal statistical studies or modeling are done.

In data science, exploratory data analysis, or EDA, is a stage in the analytical process that use a number of methods to visualize, examine, and identify patterns in the data. The EDA method's creator, John Turkey, compared it to detective work since it requires thorough investigation before drawing any conclusions about the course of events.   
  
It is possible to find problems in your data, such as incorrect or missing values, typos, and anomalies (outliers), by conducting a thorough and thorough exploratory data analysis. Furthermore, you will get knowledge about the data distribution, the correlation between factors, and identify variables that might not have an impact on the intended result.

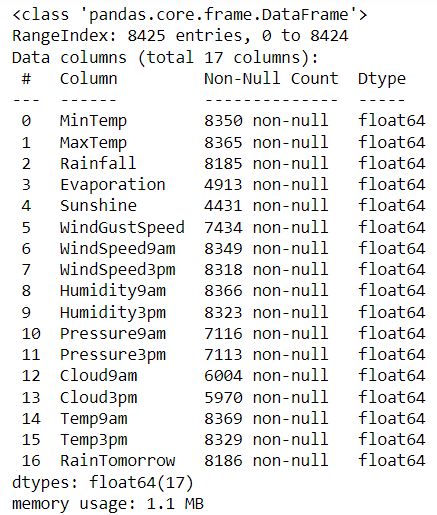
# Our target column is 'RainTomorrow'

data['RainTomorrow'] = data['RainTomorrow'].replace({'No': 0, 'Yes': 1})

# Extracting the numerical dataset

numerical\_data = data.select\_dtypes(include=['float64', 'int64'])

numerical\_data.info()



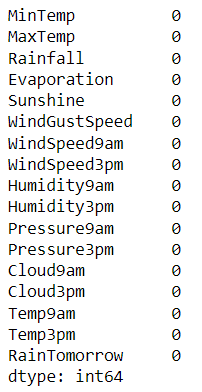
# Drop the null values from the numerical dataset

df = numerical\_data

df.dropna(inplace=True)

# Checking the null values

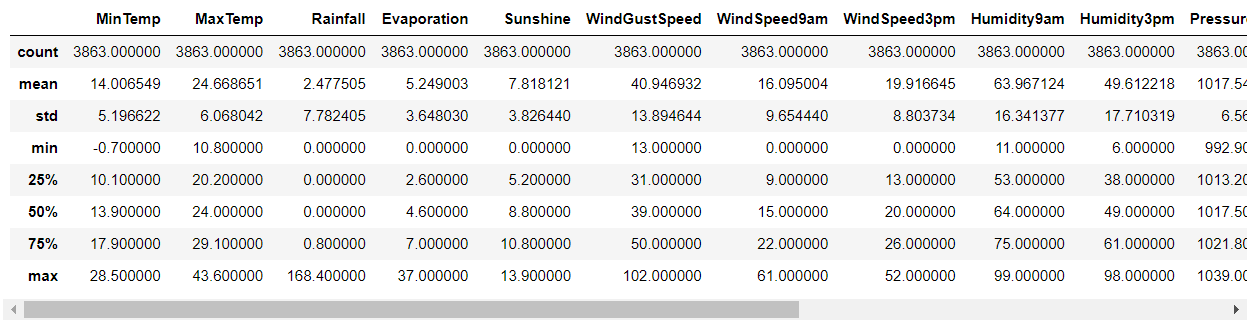
df.isnull().sum()



# Description of the dataset

The function data.describe() produces descriptive statistics, including count, mean, standard deviation, minimum, maximum, and quartile values, for the numerical columns in the dataset.

df.describe()



1. **Pre-processing Pipeline**

An essential phase in the data mining process is data preparation. It describes the steps taken to prepare data for analysis, such as cleaning, converting, and integrating it. Enhancing the quality of the data and tailoring it to the particular data mining task are the objectives of data preprocessing.

We have cleaned the dataset and prepared the numerical dataset to proceed further. Now we will check the multicollinearity and if found we will remove those columns.

Now let’s move forward with the code to check multicollinearity. To start with draw a heatmap to check correlation.

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# # Calculate the correlation matrix

correlation\_matrix = df.drop('RainTomorrow', axis=1).corr()

# # Create a mask for the upper triangle

mask = np.triu(np.ones\_like(correlation\_matrix, dtype=bool))

# Plot the heatmap

plt.figure(figsize=(20, 10))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', mask=mask, fmt=".2f")

plt.title('Correlation Heatmap')

plt.show()

# Find features with high correlation

high\_corr\_features = set()

for i in range(len(correlation\_matrix.columns)):

for j in range(i):

if abs(correlation\_matrix.iloc[i, j]) > 0.7:

colname = correlation\_matrix.columns[i]

high\_corr\_features.add(colname)

print("Features to remove: ",high\_corr\_features)

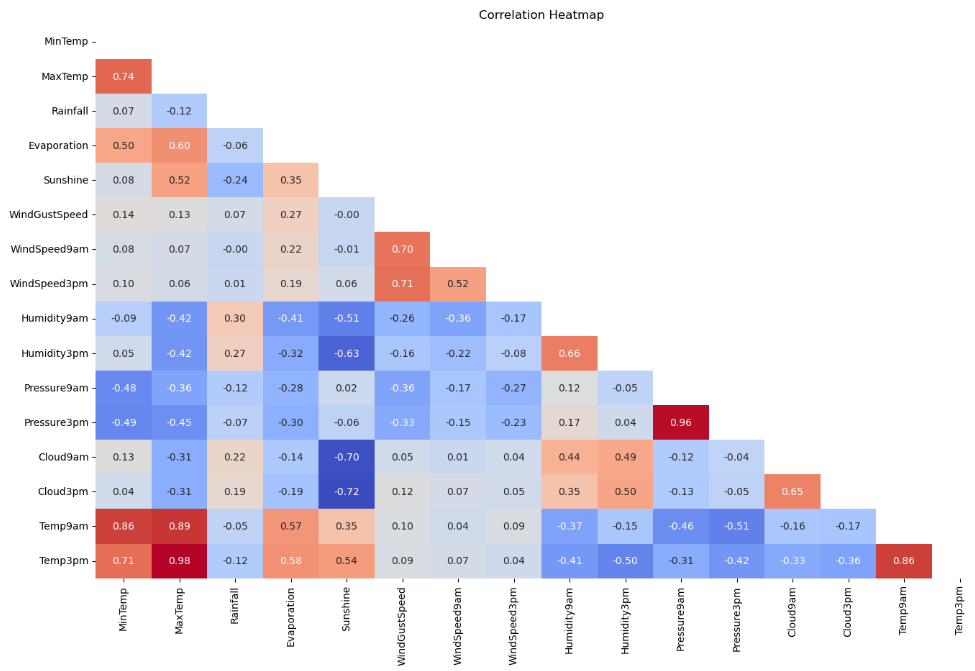
# Remove highly correlated features

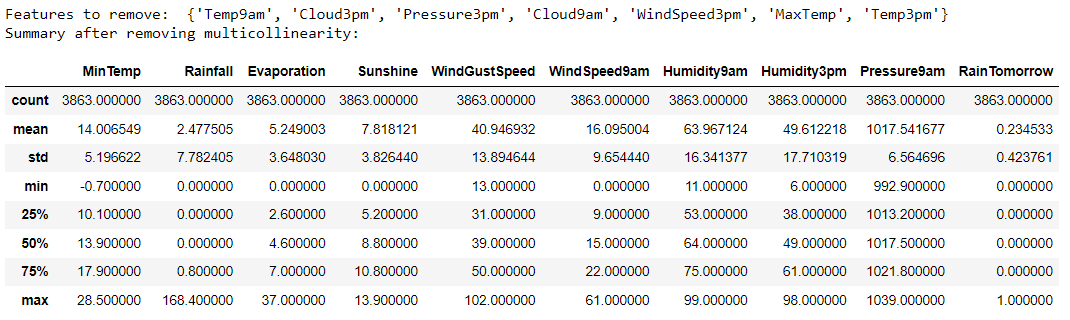
df = df.drop(high\_corr\_features, axis=1)

# Print summary after removing multicollinearity

print("Summary after removing multicollinearity:")

df.describe()





After applying the multicollinearity check at 0.7, we have found that there are some columns showing multicollinearity which can be neglected. These features include

{‘Temp9am’, ‘Cloud3pm’, ‘Pressure3pm’, ‘Cloud9am’, ‘WindSpeed3pm’, ‘MaxTemp’, ‘Temp3pm}

After removing these features we are left with the following features:

{‘MinTemp’, ‘Sunshine’, ‘WindGustSpeed’, ‘WindSpeed9am’, ‘Humidity9am’, ‘Humidity3pm’, ‘Pressure9am’}

Now these remaining features would be used for training purpose which we will see in the next step while build machine learning model.

1. **Building Machine Learning Models**

**Prepare Data:**

The target variable (y) and feature variables (X) make up the two halves of the dataset.   
All columns are included in the feature variables (X), with the exception of the final column, which is usually the column that represents the target variable.   
The final column of the dataset, which represents the variable we wish to predict or categorize, is contained in the target variable (y).

# Split the data into feature variables (X) and target variable (y)

X = df.iloc[:, :-1] # Features are all columns except the last one

y = df.iloc[:, -1] # Target variable is the last column

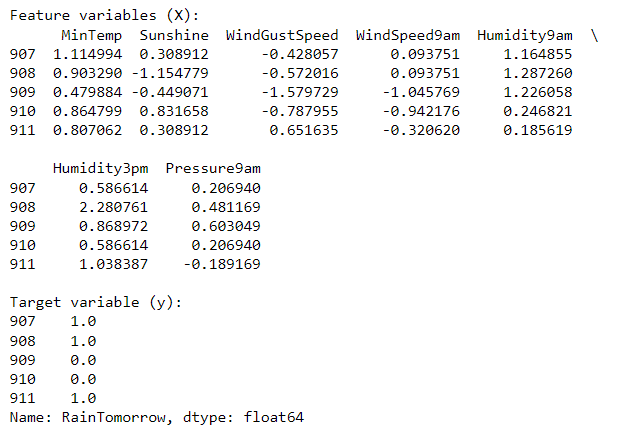
# Display the first few rows of X and y

print("Feature variables (X):")

print(X.head())

print("\nTarget variable (y):")

print(y.head())



**Data Division:**  
The train test split function from sklearn.model selection splits the dataset into training and testing sets.   
While random\_state fixes the random seed to ensure repeatability, test\_size determines how much of the dataset to include in the testing set.

from sklearn.model\_selection import train\_test\_split

# Divide the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

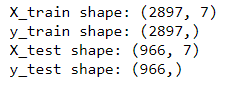
# Display the shapes of the training and testing sets

print("X\_train shape:", X\_train.shape)

print("y\_train shape:", y\_train.shape)

print("X\_test shape:", X\_test.shape)

print("y\_test shape:", y\_test.shape)



**Model Construction:**   
In this instance, the classification model is constructed using a decision tree classifier model. From sklearn.tree, the DecisionTreeClassifier is imported.

Depending on a number of meteorological factors, the DecisionTreeClassifier model is used in rainy weather forecasting to determine if it will rain tomorrow. The model finds patterns and links between features by evaluating past data and building decision trees. This helps with weather forecasting and planning by enabling precise forecasts of when rainfall will occur.

# Building model

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

**Training Models:**   
The DecisionTreeClassifier's fit method is used to train the model using the training set (X\_train, y\_train).

model = DecisionTreeClassifier()

model.fit(X\_train, y\_train)

**Assessment of the Model:**   
Following model training, the trained model's predict technique is used to make predictions on the testing data.   
To assess the performance of the model, the predicted values (y\_pred) are compared with the actual values (y\_test).   
In this instance, the sklearn.metrics accuracy\_score function is used to determine the model's accuracy.

# Make predictions on the testing data

y\_pred = model.predict(X\_test)

# Calculate the accuracy of the model

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)

**Results:**   
  
The model's accuracy on the testing data is printed out to assess the model's performance in accurately classifying situations.

Accuracy: 0.8664596273291926

1. **Concluding Remarks**

Data Integrity: The dataset utilized for this study includes a wide range of meteorological parameters that were collected from different parts of Australia during a ten-year period. The dataset needed preprocessing measures to address missing values and multicollinearity, despite its richness. To create predictive models that are trustworthy and accurate, data quality must be guaranteed.  
  
Feature Selection: Using correlation and exploratory data analysis, it was possible to identify which characteristics showed multicollinearity. These features were then eliminated from the dataset. This procedure assisted in reducing the feature set to those that were most important for rainfall prediction. Enhancing the interpretability and performance of a model requires careful feature selection.   
  
Training and Evaluation of the Model: To predict whether or not it will rain tomorrow, a decision tree classifier was trained using the chosen features. On the testing data, the model had a decent accuracy score of about 86.65%. While accuracy is a popular statistic for assessing classification models, to obtain a more complete picture of the model's performance, it's crucial to take into account additional performance metrics including precision, recall, and F1-score.

Interpretability: By visualizing decision trees, decision tree classifiers shed light on the decision-making process and give interpretability. This interpretability is helpful in figuring out how various meteorological factors affect rainfall forecast. The selection of a model should be contingent upon the particular demands of the forecasting activity, as more intricate models may compromise interpretability in the name of enhanced performance.   
Weather forecasting is a dynamic discipline that calls for constant progress and flexibility in response to shifting circumstances. To maintain accuracy and applicability, model performance should be routinely reviewed and updated with fresh information. Furthermore, feedback systems can be put in place to take user input into account and gradually enhance the forecasting system.

To sum up, the Rainfall meteorological Forecasting project showcases the utilization of machine learning methods to forecast rainfall by utilizing past meteorological information. Even though the decision tree classifier produced encouraging results, it still has to be improved and refined. To increase our comprehension of weather patterns and the precision of weather forecasting models, more study and development in this field are needed.