**Project Report – Rainfall Prediction in Australia**

Submitted by: Deepak Kumar

Date: 16th November 2022

*for*

Fulfilment of Requirement for ‘UCD Specialist Certificate in Data Analytics’

**GitHub URL**:

<https://github.com/deepakkumar20/UCDPA_DeepakKumar>

**Abstract**:

This project is done as a fulfilment of final requirement for completion of ‘UCD Specialist Certificate in Data Analytics’. This project work is using the dataset for Rain in Australia to predict a target variable ‘Rain Tomorrow’ which represents if it will rain the next day based on the historically available data. The data is gathered from multiple sources and collected in the Kaggle dataset. Before applying the machine learning techniques, data is first loaded, cleaned and analysed visually for any correlations between the input and target variable. Useful features are extracted and multiple classification models are applied and tested it to find the accuracy of our model.

**Introduction**:

Predicting weather is an important and general use cases across countries these days. Machine learning provides us a way to analyse the historical data and generate meaningful insights which can help in taking informed decisions. This use case is suitable for implementing classification methods (binary class classification) where the output can be either YES/NO or 0/1.

In this project scenario, the task is to predict if it will rain tomorrow which can have either a YES/NO value.

This dataset is chosen as it has a fairly large number of records which will help us to train and test our model better and also make our data split much better.

**Dataset**:

Dataset chosen for this project work is: ‘**Rain in Australia**’

This dataset consists of around 10 years of daily weather observations across a number of locations within Australia.

The reason this data is selected because keeping the target to perform binary classification (two class classification) where we should be able to predict the target variable. This dataset is suitable for performing binary classification as the target variable to be predicated is ‘RainTomorrow’ which can have 2 possible values i.e. YES or NO.

The value of target variable will be YES if there is possibility of Rain the next day. By rain we assume rain more than 1mm on that particular day.

Total number of Rows in Dataset = 142267

Total number of columns in Dataset = 23

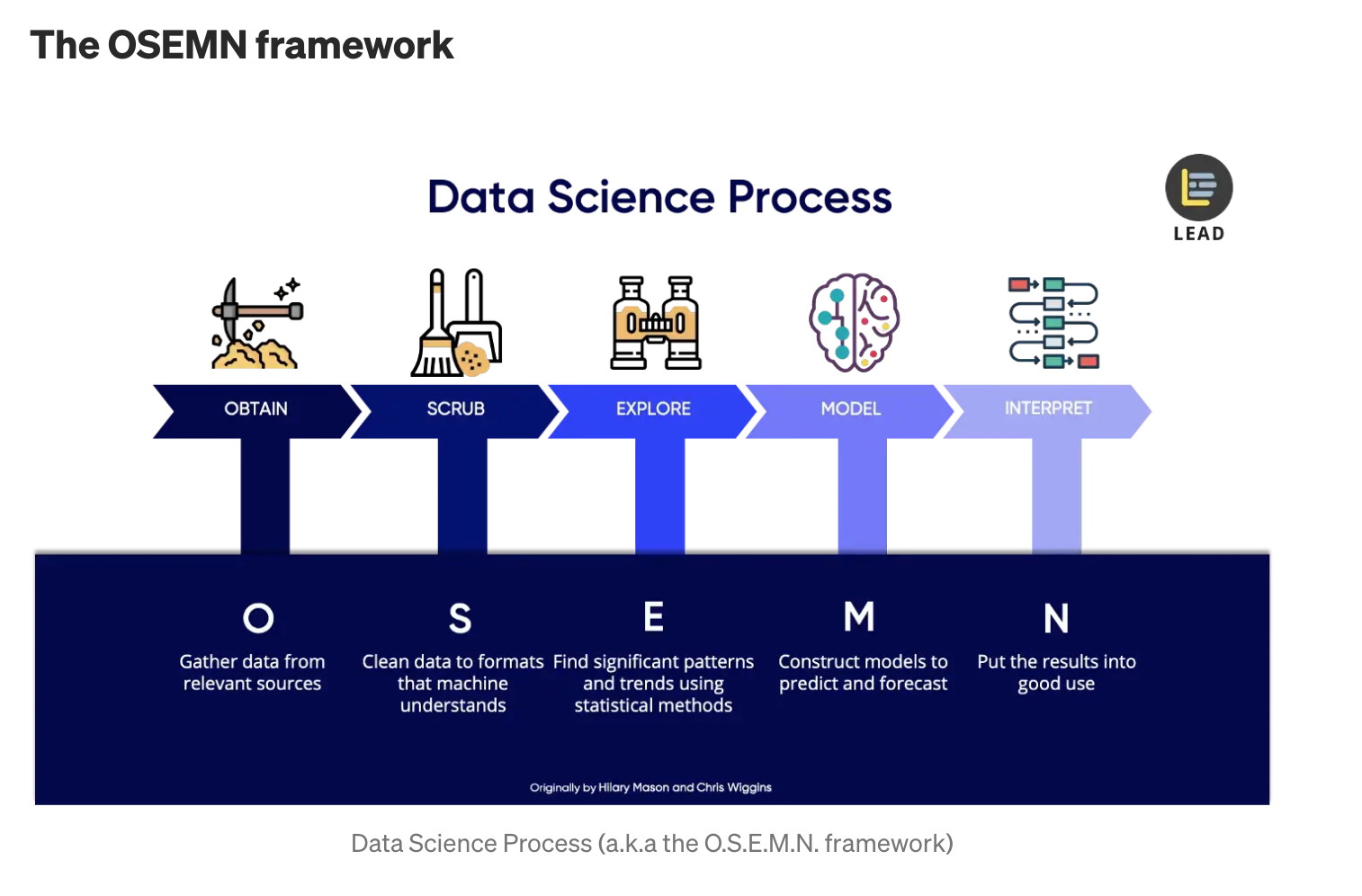
The link to the dataset is:

<https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package>

**Implementation Process:**

The implementation process follows a very famous approach for such projects knows as OSEMN framework (image below).

This framework defines 5 major steps in a data science process. The approach was initially created by ‘Hilary Mason and Chris Wiggins’.

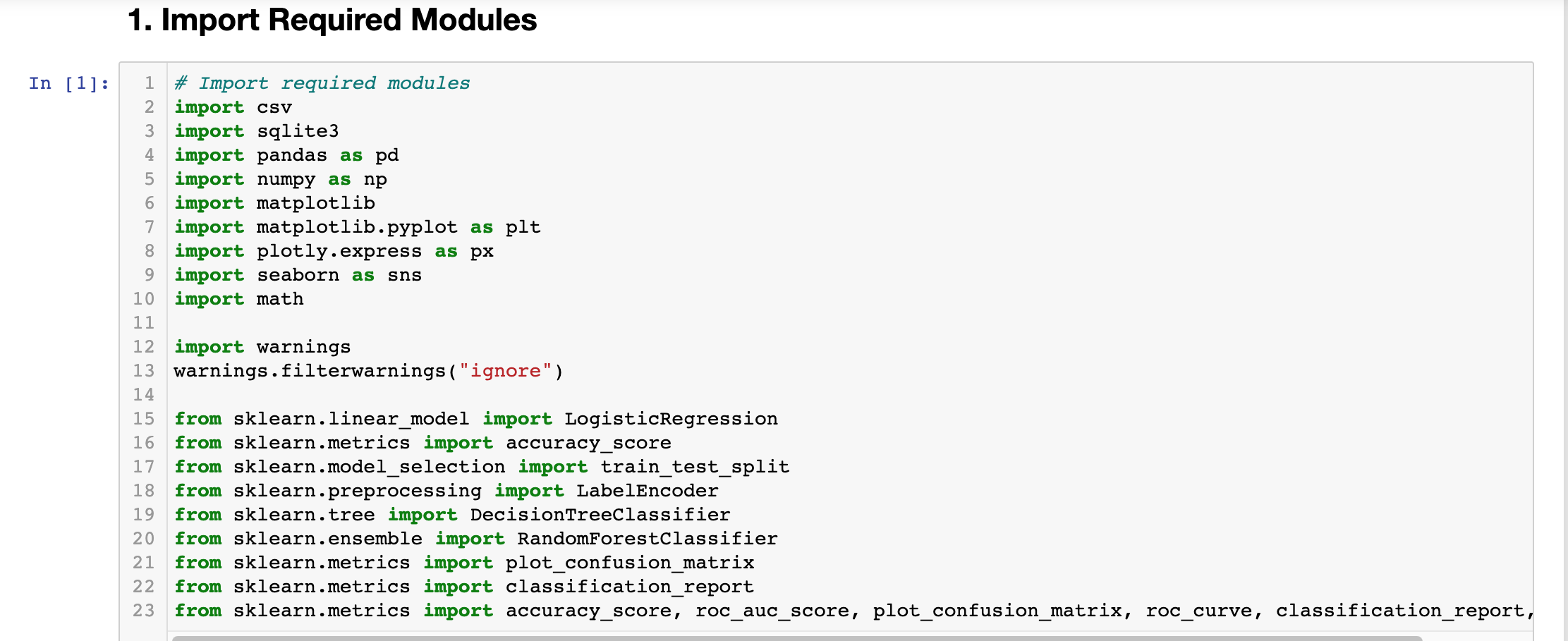


**Image Reference: [6]**

The steps are followed in the project in the same sequence. Below are low level details:

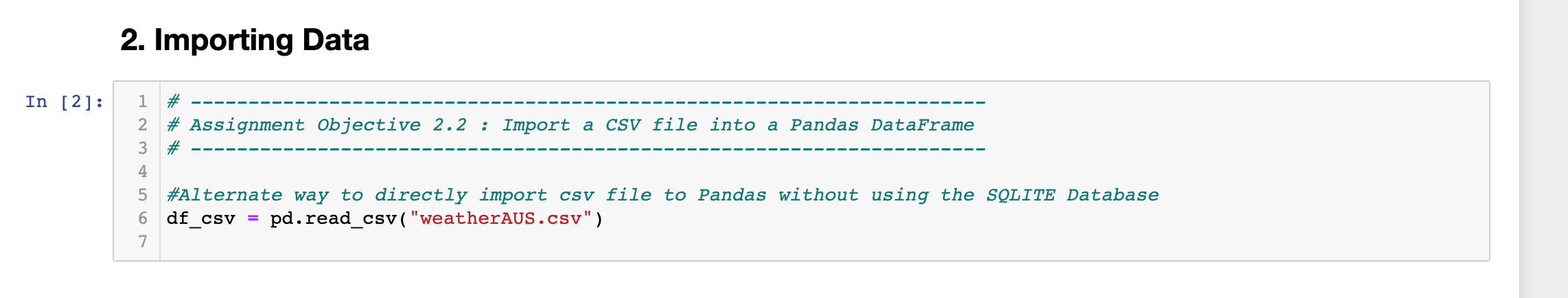
**Step 1: Obtain – Import the required data**

This step is broken down into 2 sub parts. First part includes importing the required libraries for performing our work and 2nd part involves importing the actual data.

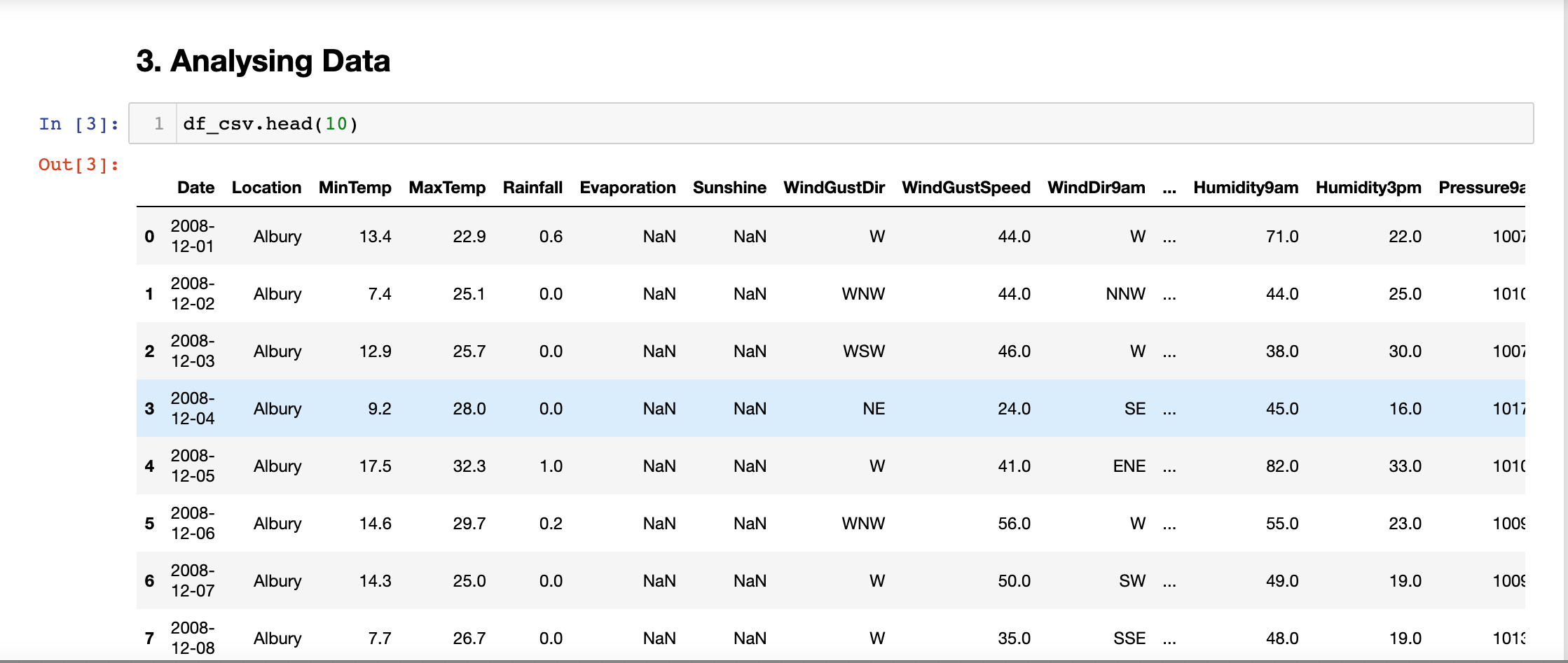


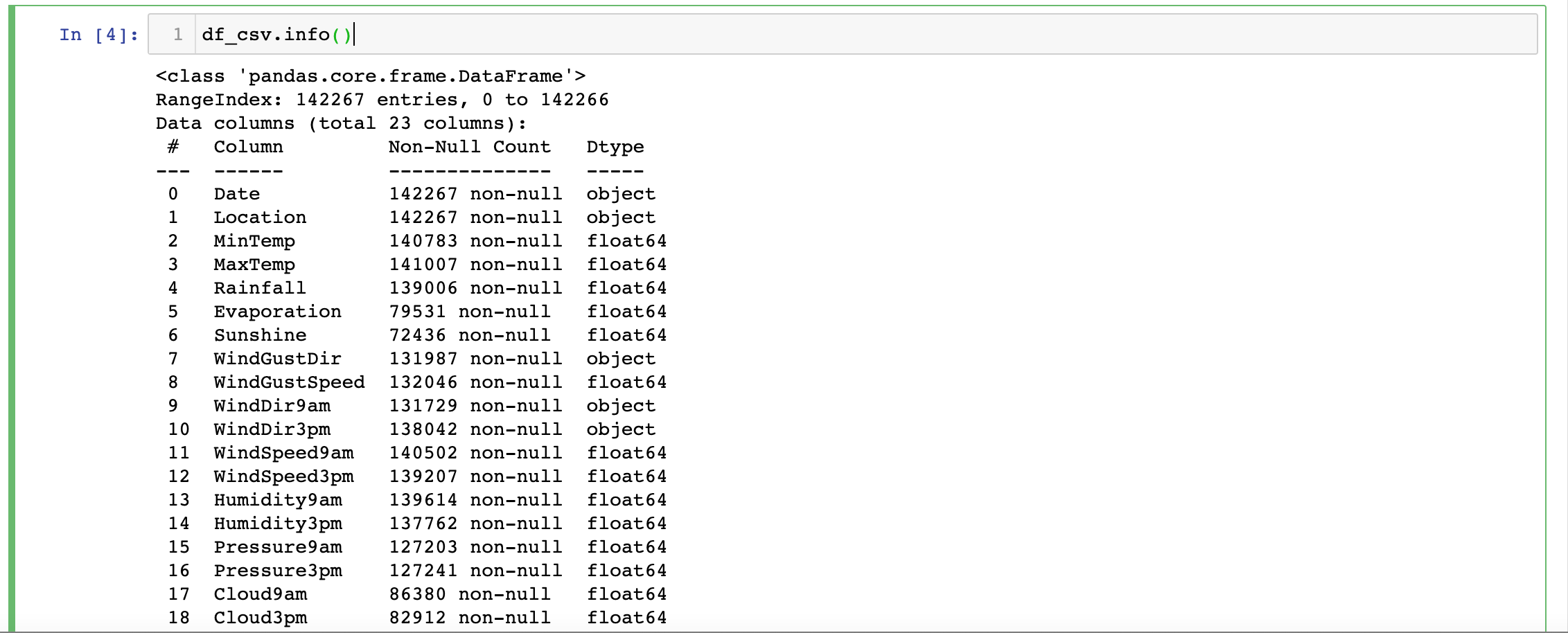
Second step involves importing the Dataset of rainfall.

For this step, Pandas library has been used to directly import the csv file into the Pandas data frame. Once we have the data frame, we can perform further data cleaning and analysis in an efficient manner.



Post obtaining the data, the data is checked if it has been loaded correctly, what are the different columns, what are the data types of different columns etc.



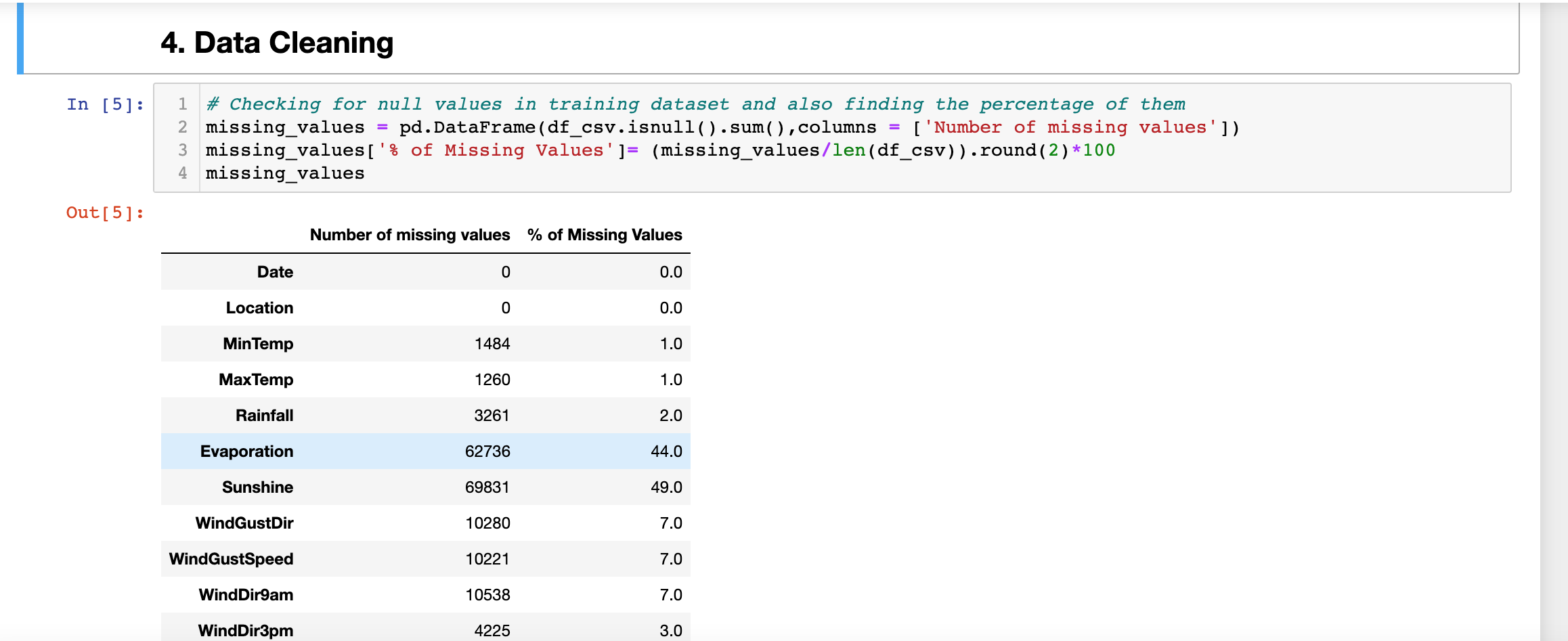


The result of info commands shows that there are a total of 142267 entries and 23 columns of object, float64 data types.

**Step 2: Scrub – Data Cleaning**

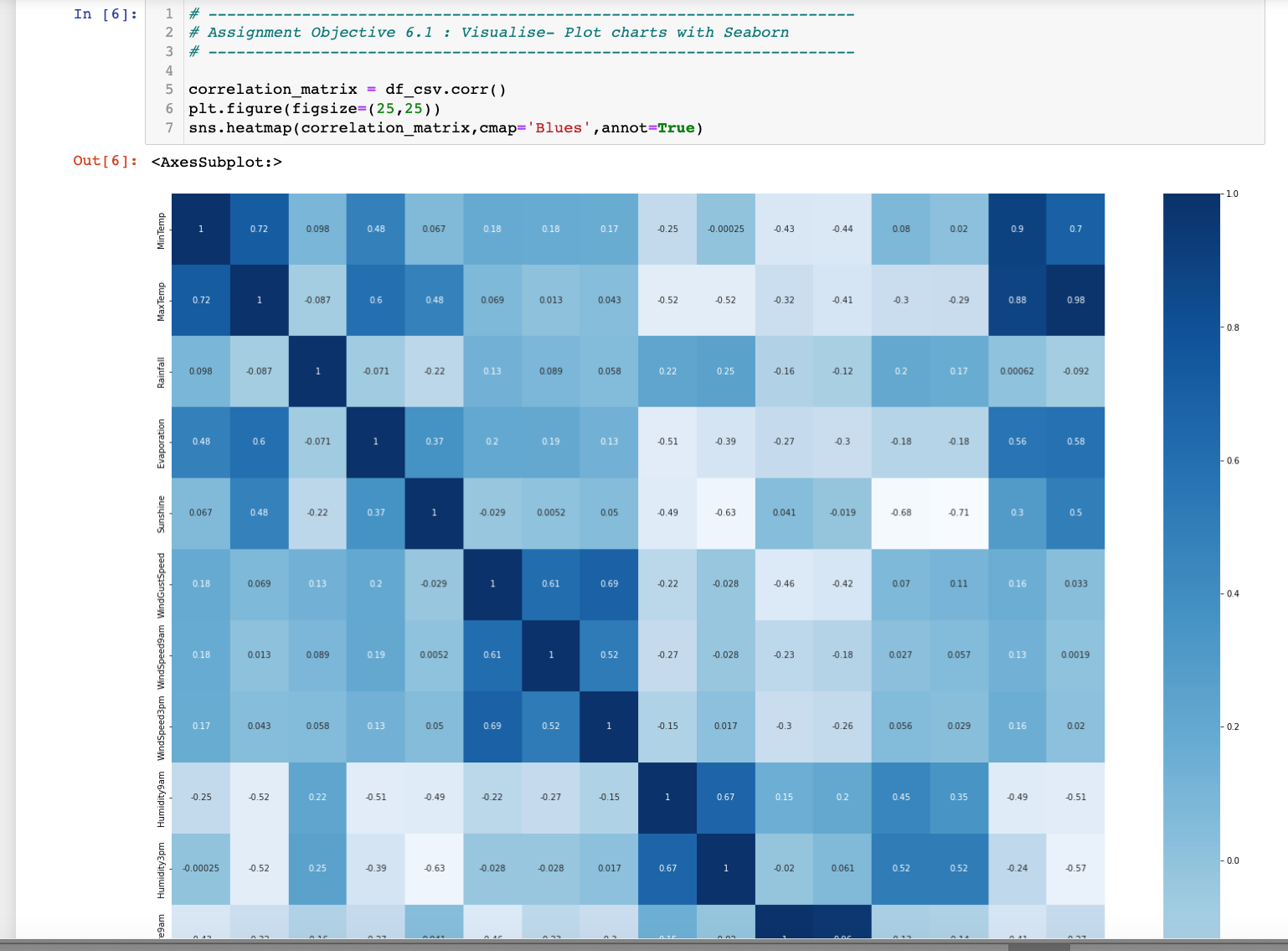
Once we have loaded the data, we start the data cleaning process.

First we check if there are any missing values in the data and if yes, what is the count and percentage of those missing values.



From above result, we can see that there are a number of missing values in most of the columns and some columns have a very high percentage of missing values like Evaporation, Sunshine etc.

Before dropping any column from the data frame, we can check the correlation matrix and find out the correlation values between these input variables and the target variable.



Max temperature is highly correlated to Temperature at 9am and Minimum temperature.

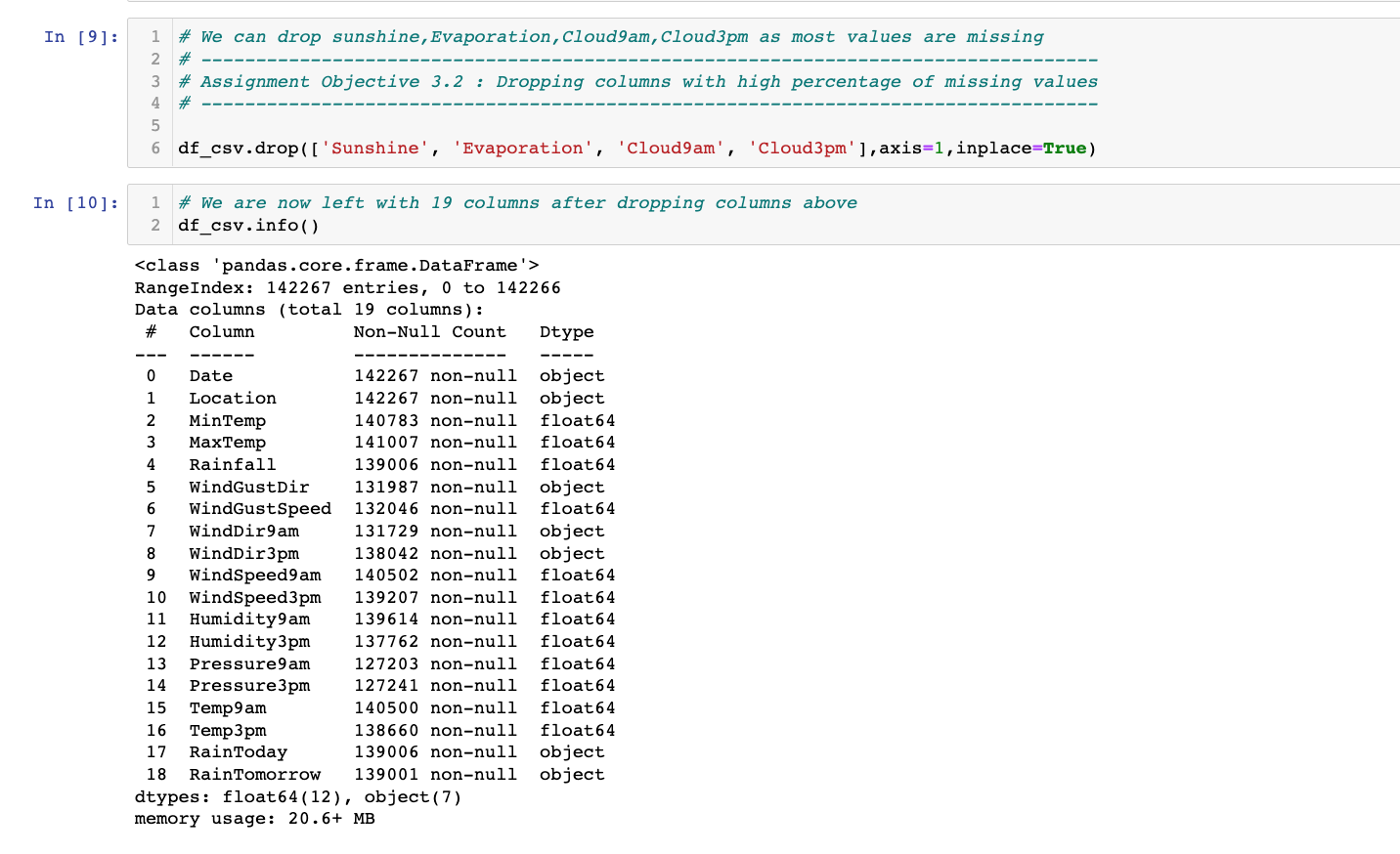
Pressure at 9am is highly correlated to Pressure at 3pm, it seems the pressures impact each other at different time periods during the day.

Cloud and Sunshine are negatively correlated, which also makes sense. Wind and Pressure are also negatively correlated, although the correlation is weak. Humidity and Cloud levels are positively correlated, with a weak correlation.

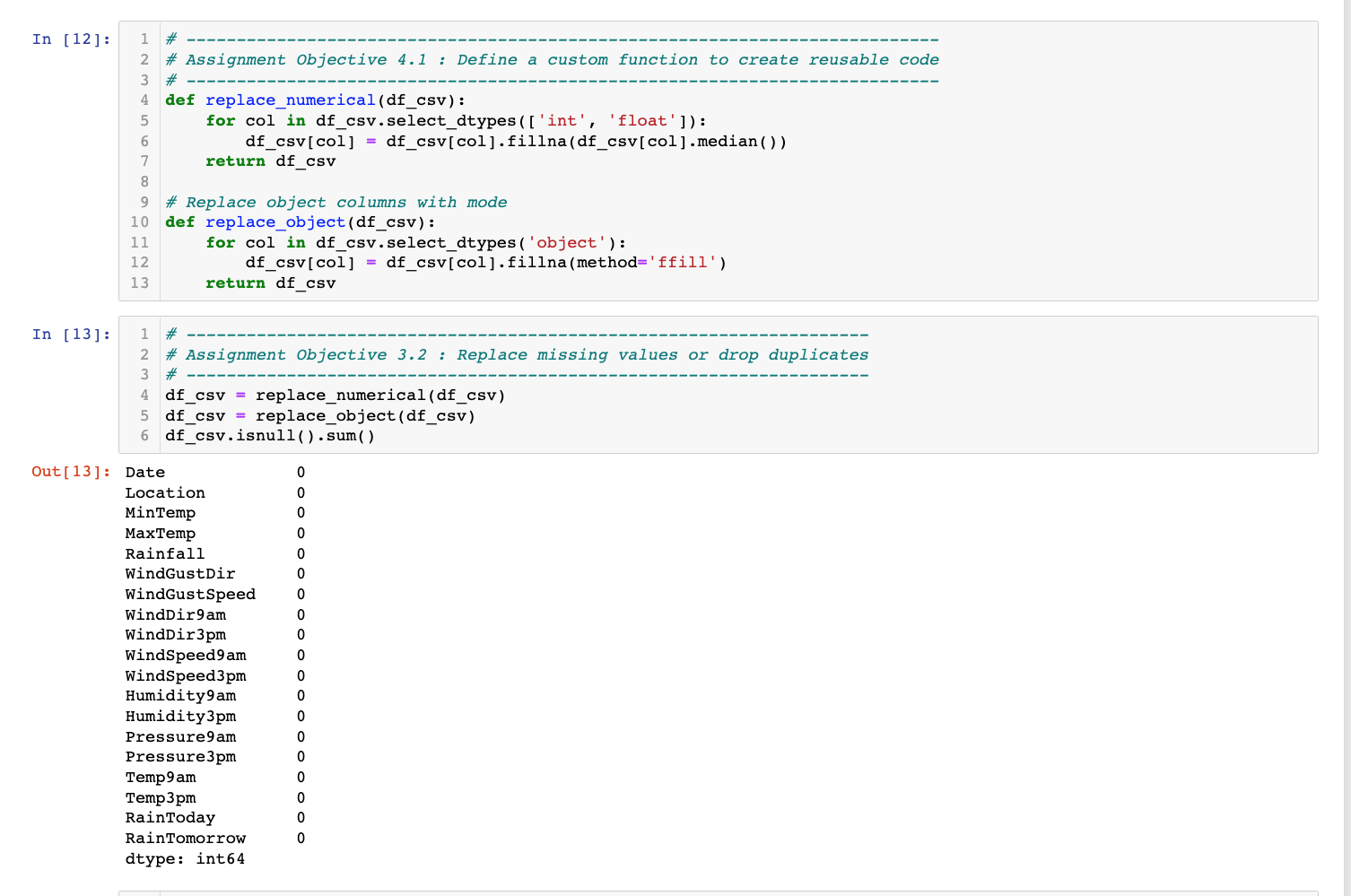
We can also check for the duplicate values. If there are any duplicates, we can delete those rows and they will not be of use to us and will pollute the data.



Based on the above 2 methods, we can see there are four columns which have a low correlation value, are having high number of missing values. So, we can drop these 4 columns from our dataset to only have the relevant columns for further analysis.



Now we need to handle the columns which have missing values. We can treat the object and float datatypes differently. To handle this for all the columns, helper methods are created to do this more cleanly.



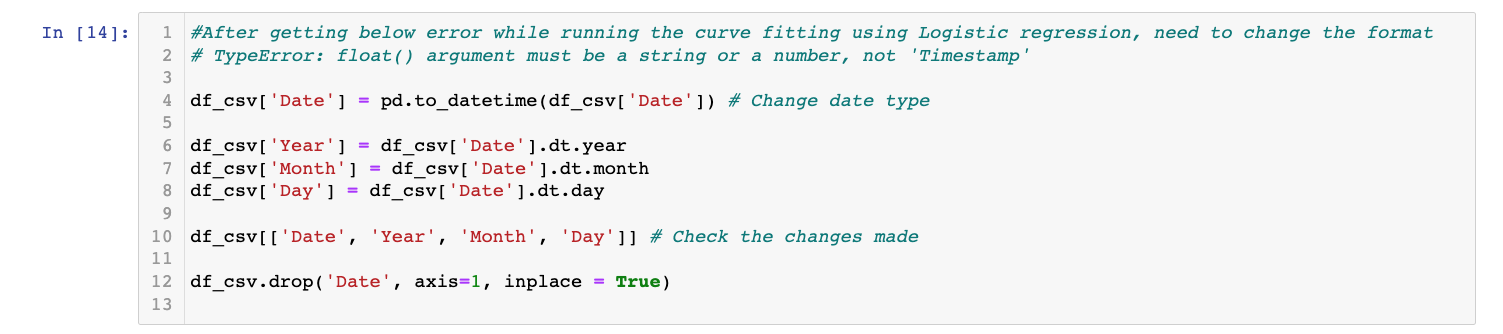
The helper methods are used to fill in the missing values in data frame.

For the int and float values, median values are used to fill missing columns.

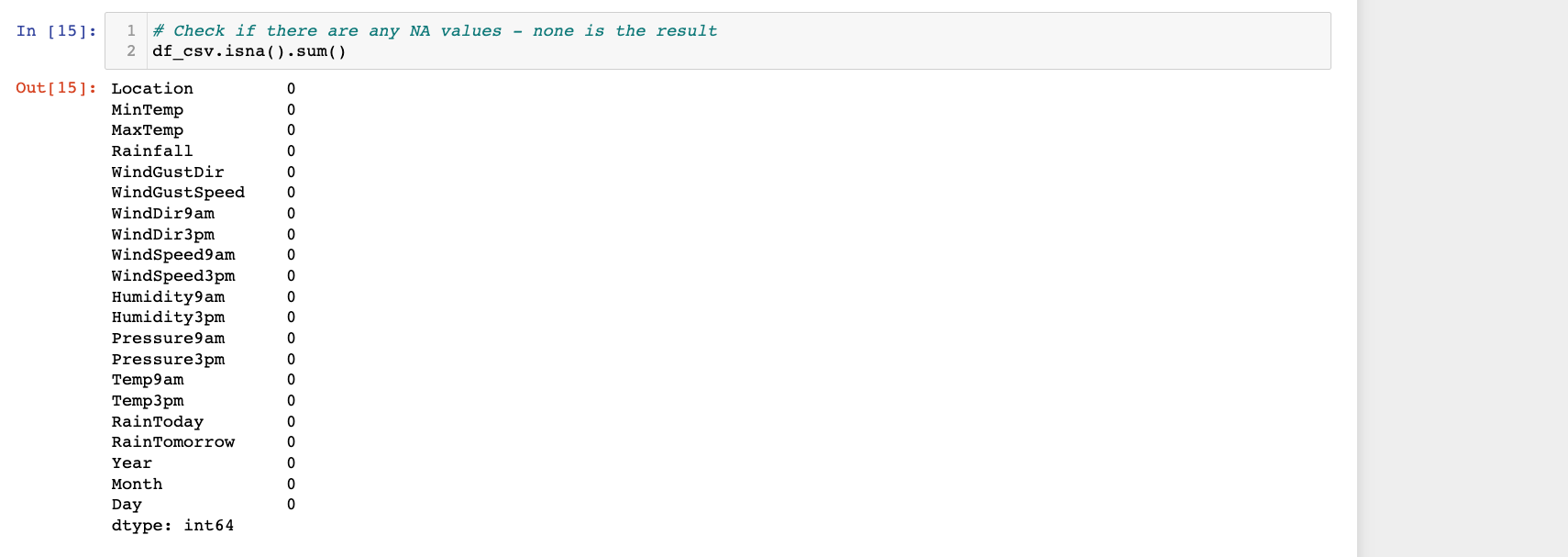
For the object datatype columns, the forward fill method ‘**ffill’** is used to complete the data frame.

As a result, we can see from above that the NULL values are now zero for all the columns.

The Date column in our data is Timestamp type. As we need to do the analysis based on year, month and day, it is better to drop this column and extract the year, month and date values from the timestamp. While running the Logistic regression later in this project, this error was encountered for Date datatype, hence I needed to backtrack and find the root cause.

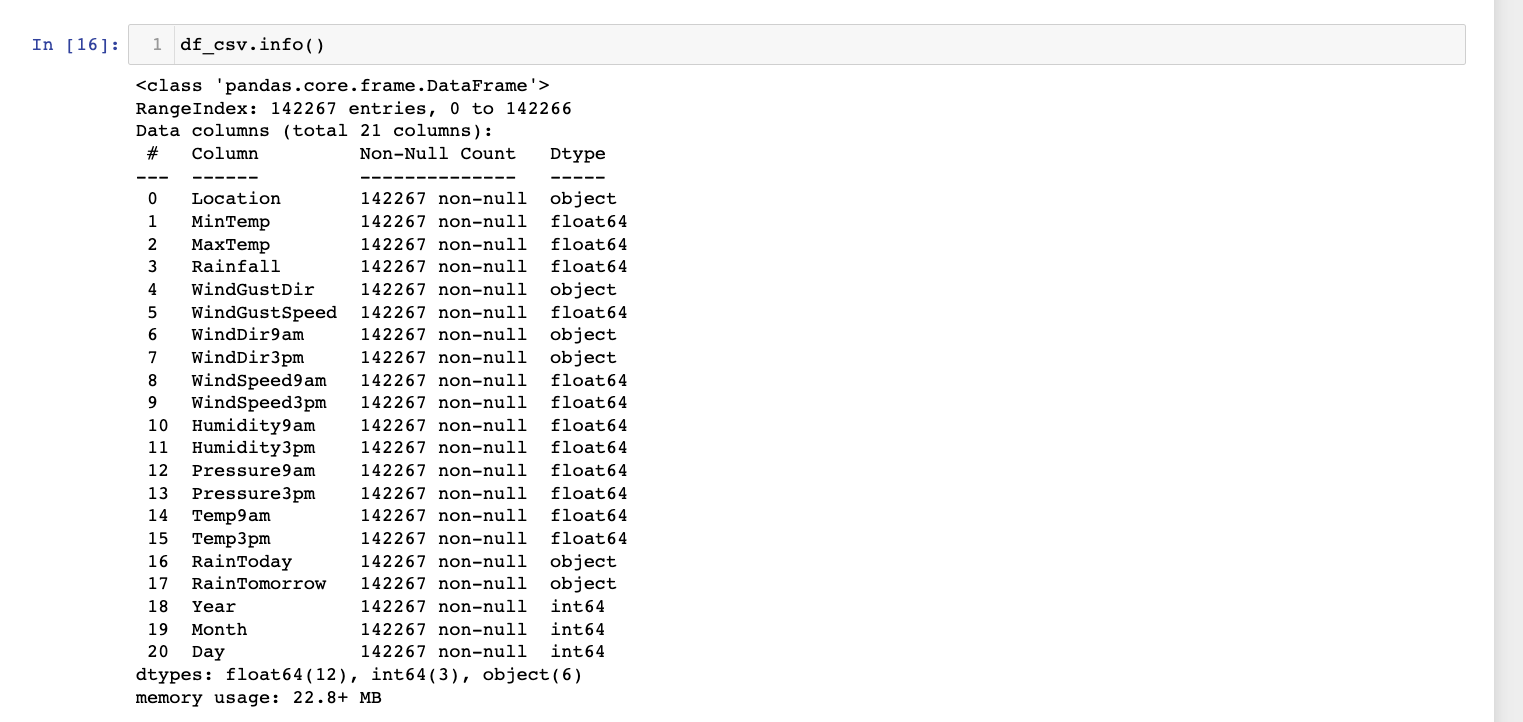


We also need to check if there are any NA(Not applicable) values in our dataset. We need to handle the NA values as well. To do this, we run the below command.



There are no NA values hence nothing is required.

Now our data is clean with the below final informative state.



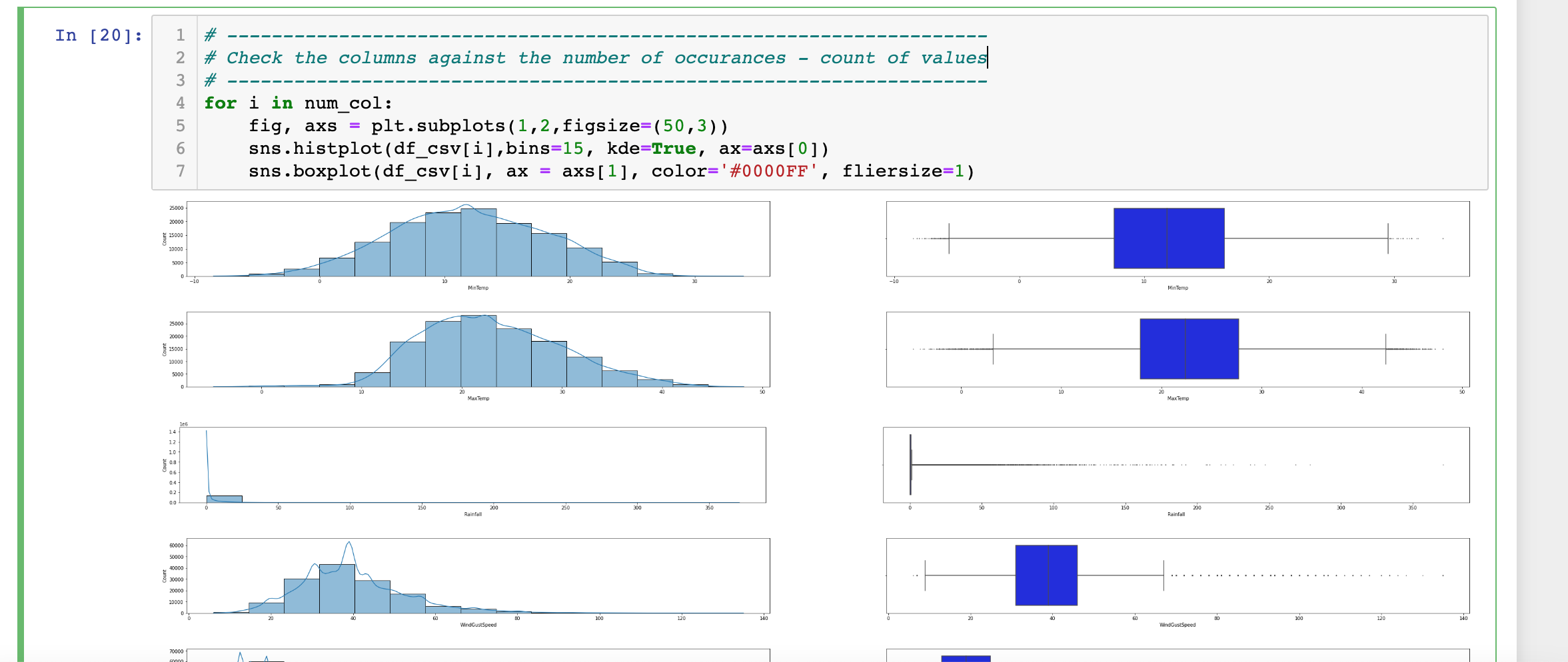
**Step 3: Explore – Analysis of data using Visualisations**

Once we have loaded and cleaned our data, we need to explore the data using visualisations. They help us in finding meaningful insights and patterns from the data so that we can make right decisions while modelling.

As a first step, few functions are defined to be used later on while analysis.



First, we can see the count using the histogram and boxplot and see how each columns values are dispersed. For some columns we have a range of values, whereas some others are having values concentrated in a particular value.

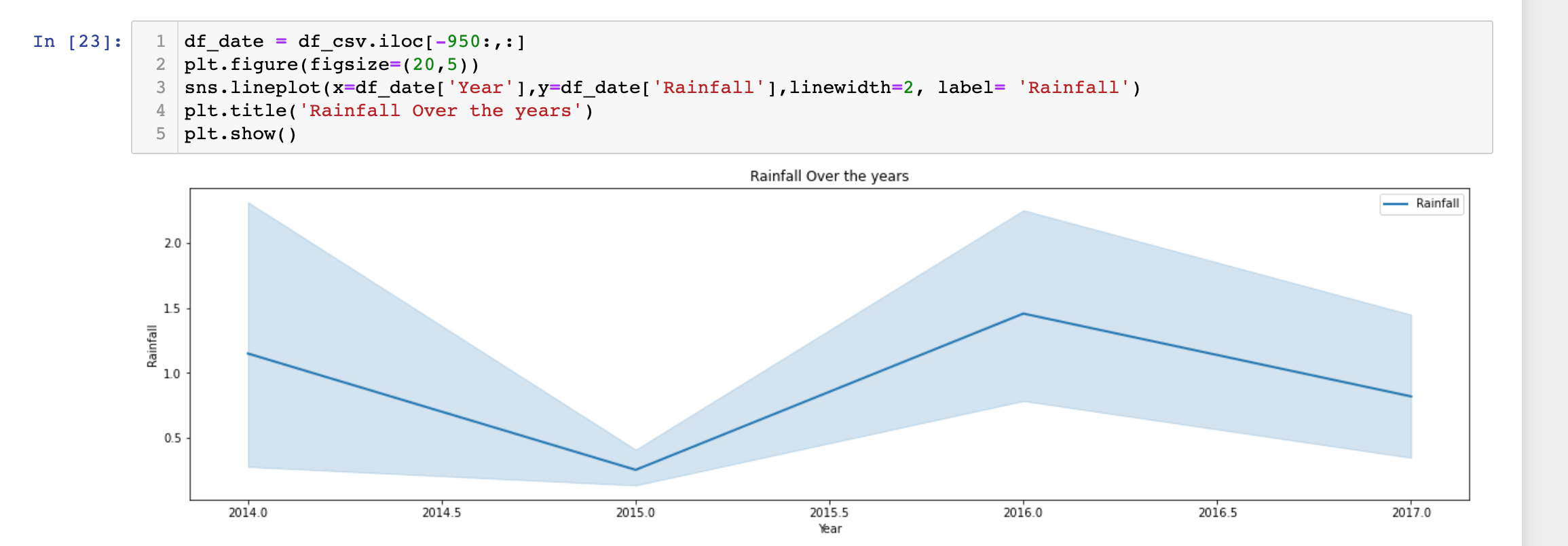


Next we can plot a count plot with Locations in Y axis and and count in X-Axis. This plot lets us see the insights into the data that which particular location is having the maximum and minimum number of values. If there is a huge difference, we can get biased results.



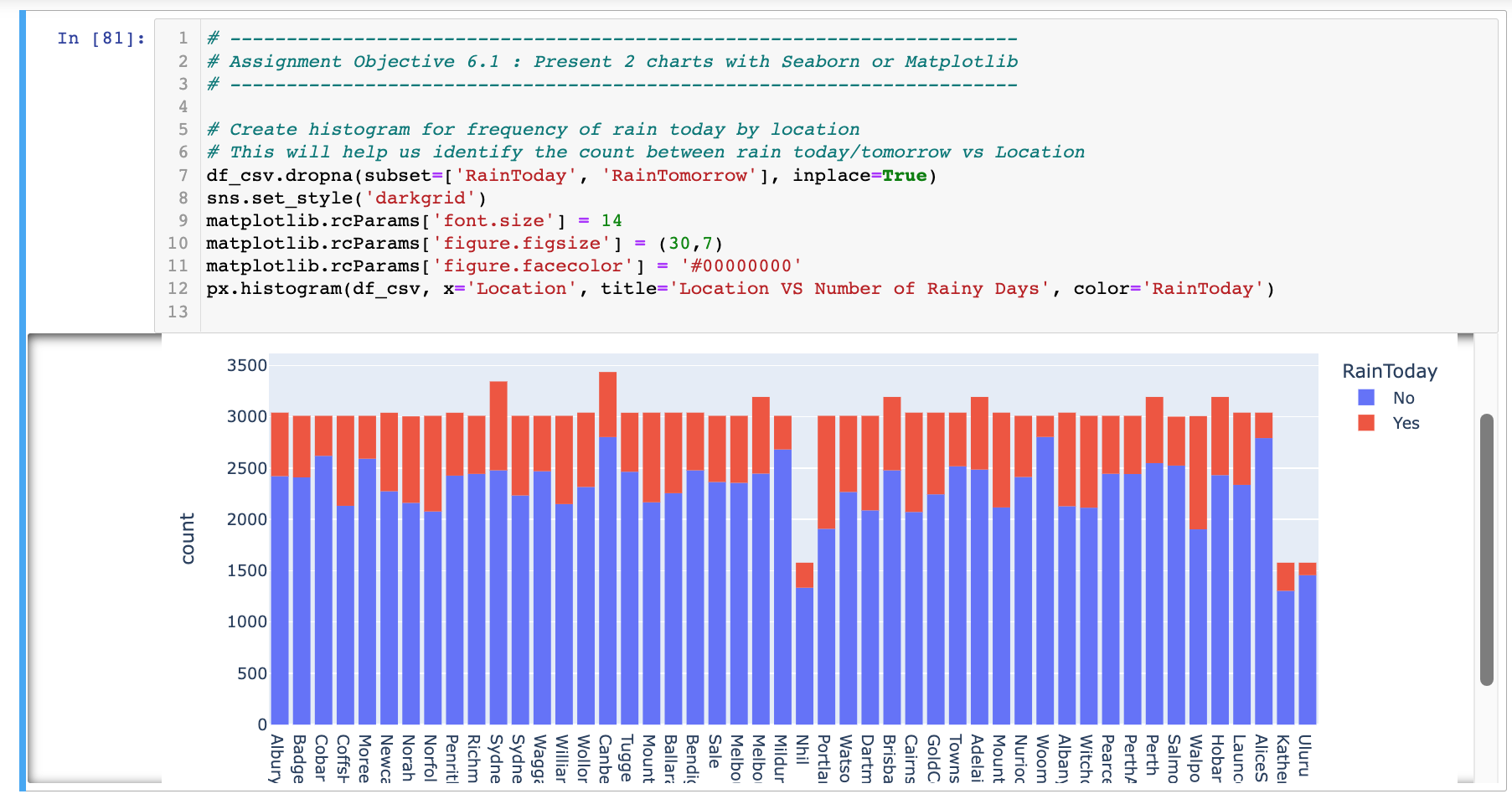
From the above chart, we can see Canberra, Sydney and Hobart have the maximum number of weather recordings. Other cities also have comparable numbers. Variation is not high.

We can also plot a chart of rainfall over years and see which months or year are having high rainfall as compared to others.



Here, we can see 2014,2016 starting of the years are having high rainfalls as compared to others. 2015 was one year when the rainfall was less in beginning but it later picked on.

We can plot a chart to see location vs number off rainy days. It will help us identify at which locations it rains more than others.





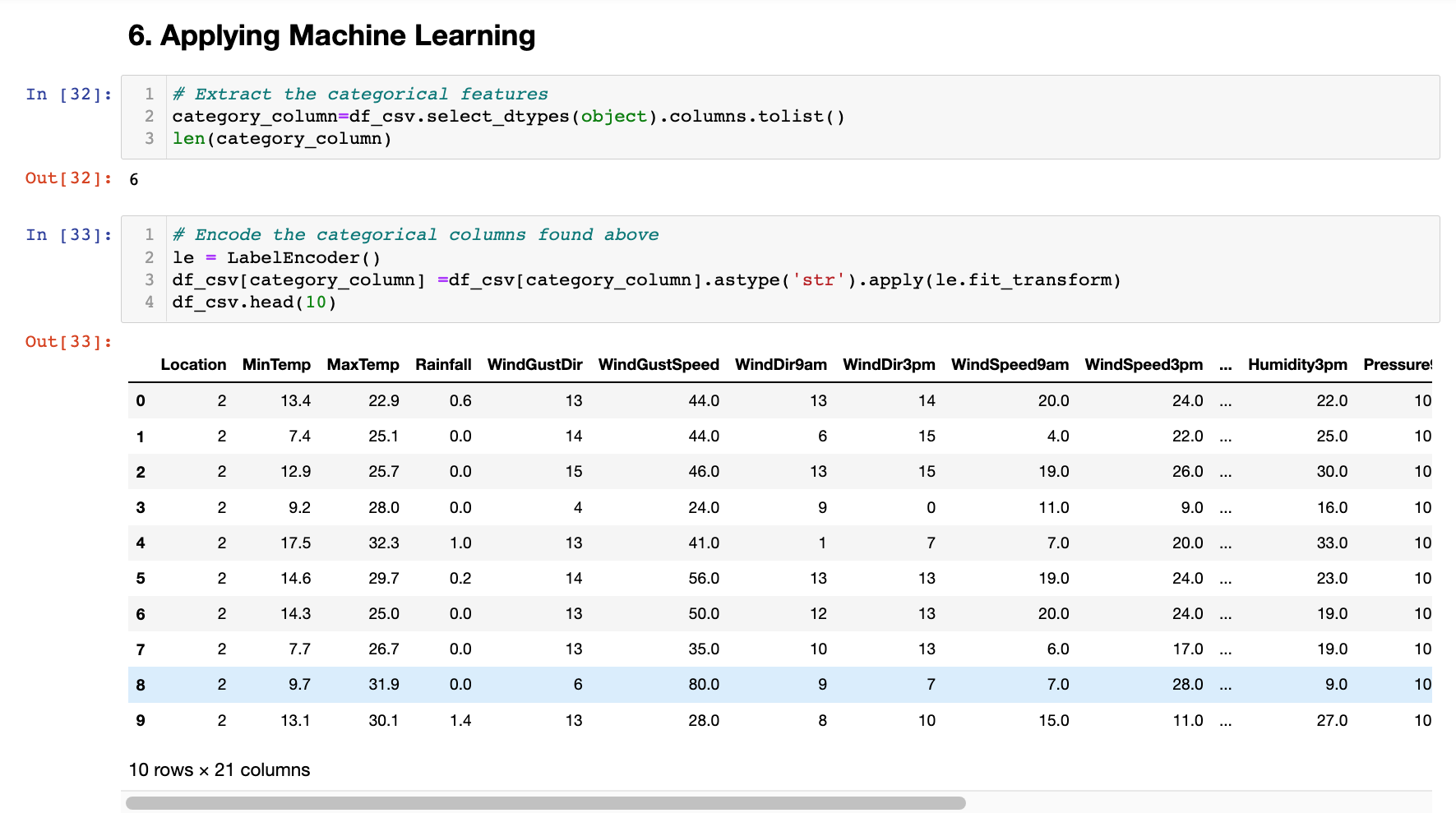


Step 4 and Step 5 of OSEMN are covered in Results and Insights section below respectively.

**Results:**

Before applying the machine learning methods, final step is done to encode the categorical columns to appropriate numerical values.

This is done by using the Label Encoder.



*Data is then Randomised into Train and Test datasets.*



3 different classification methods are used in the project namely:

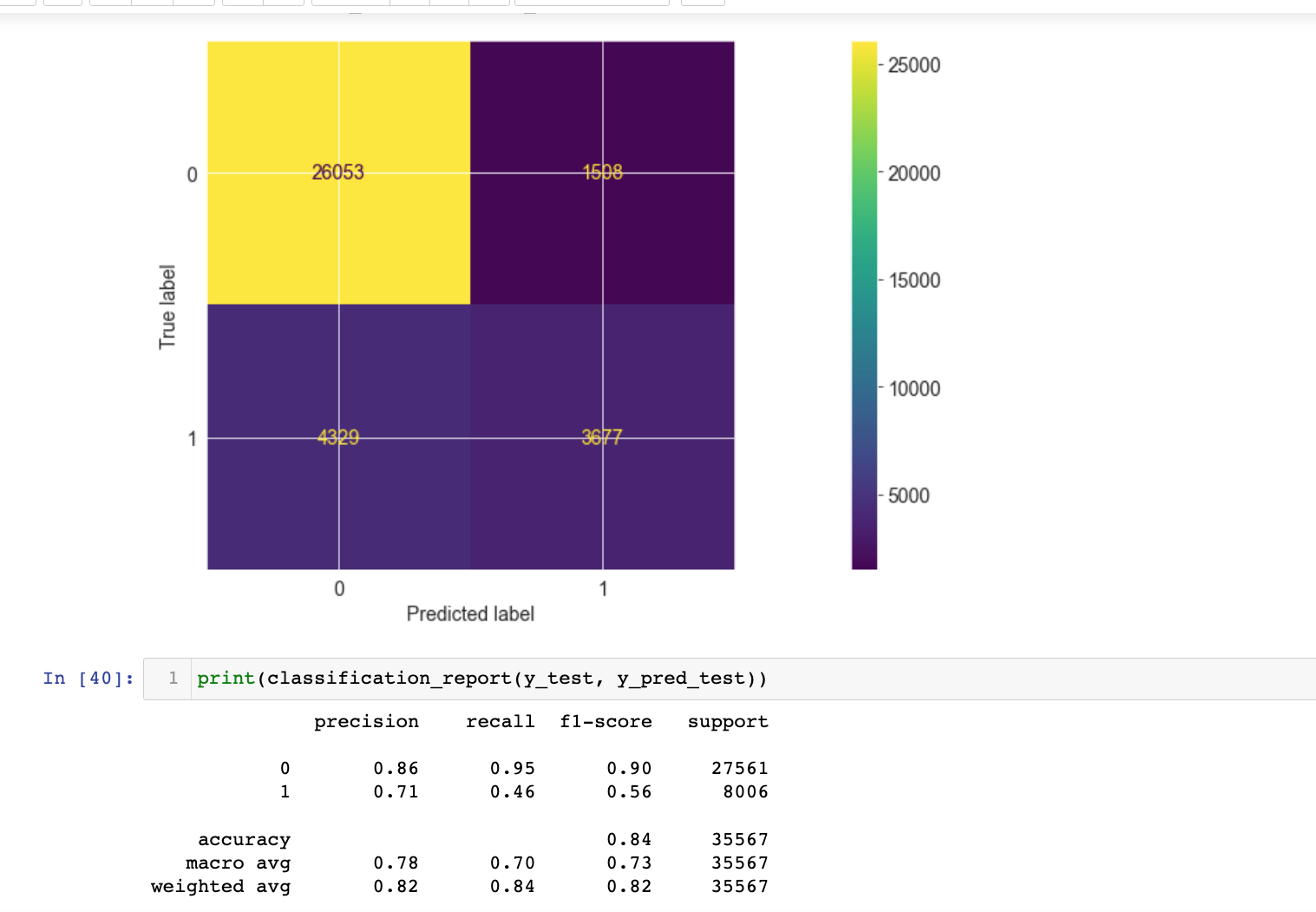
1. Logistic Regression
2. Decision Tree Classifier
3. Random Forest Classifier

**Logistic Regression**

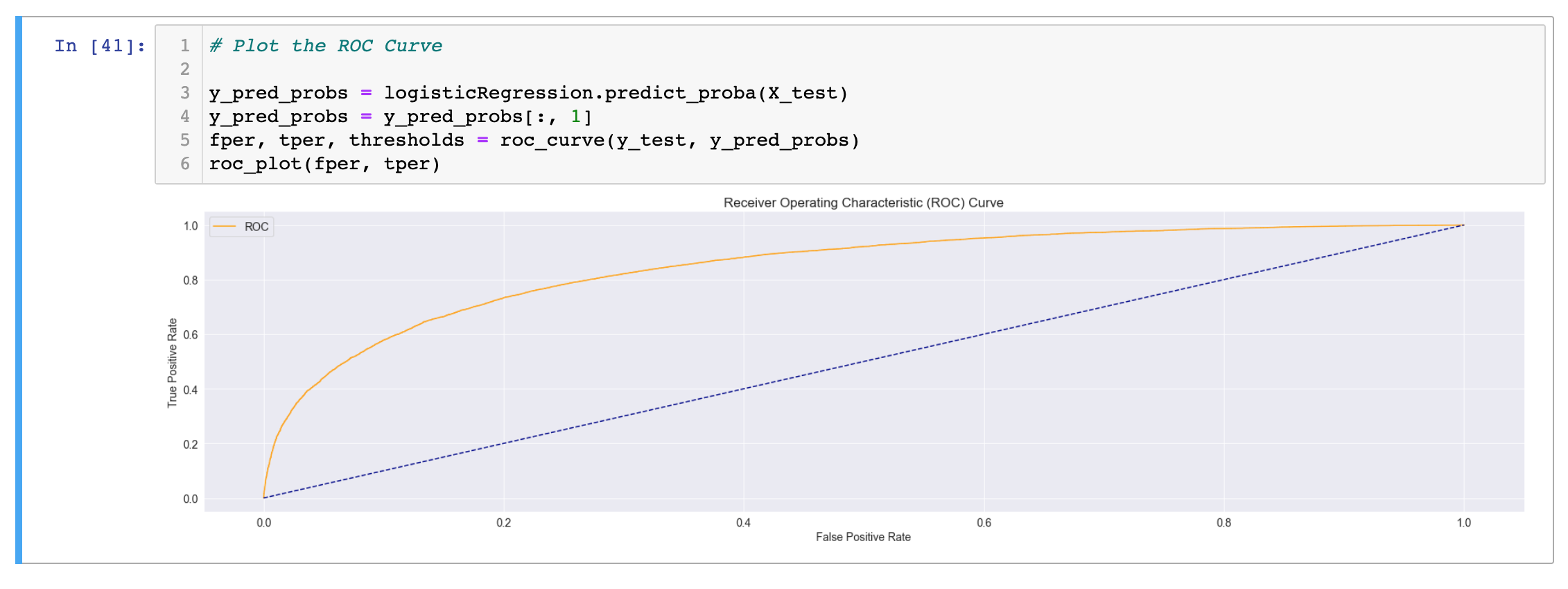
Liblinear solver is taken from the set of: ['liblinear', 'newton-cg', 'lbfgs', 'sag', 'saga'] without going into much mathematical details of these.

****

The confusion matrix and the classification report using the sklearn libaray are printed below.

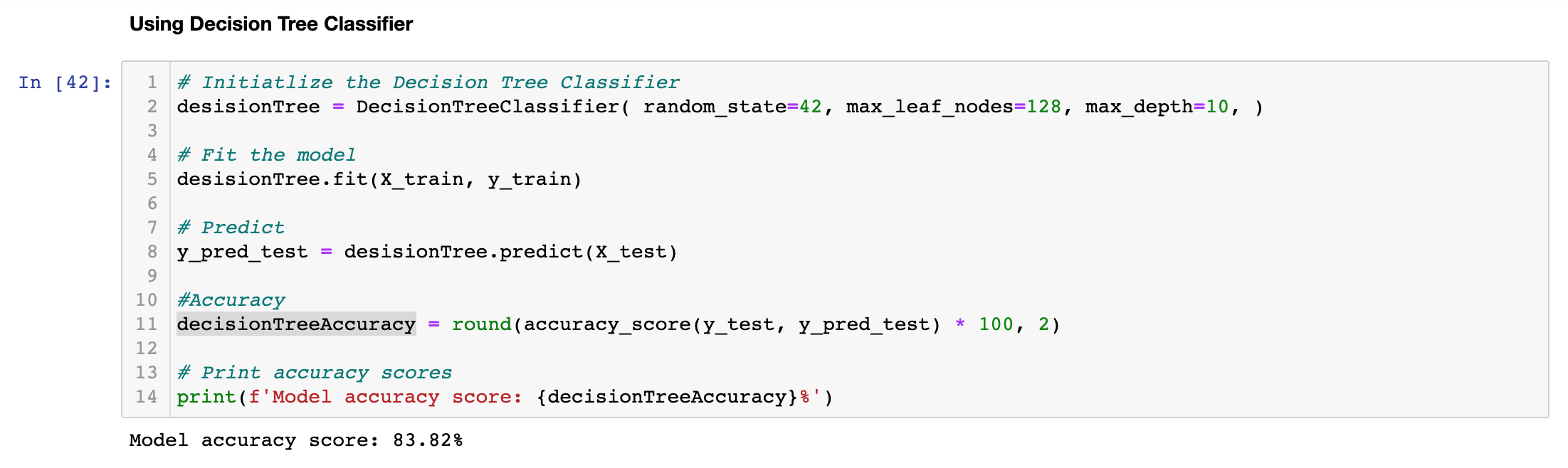


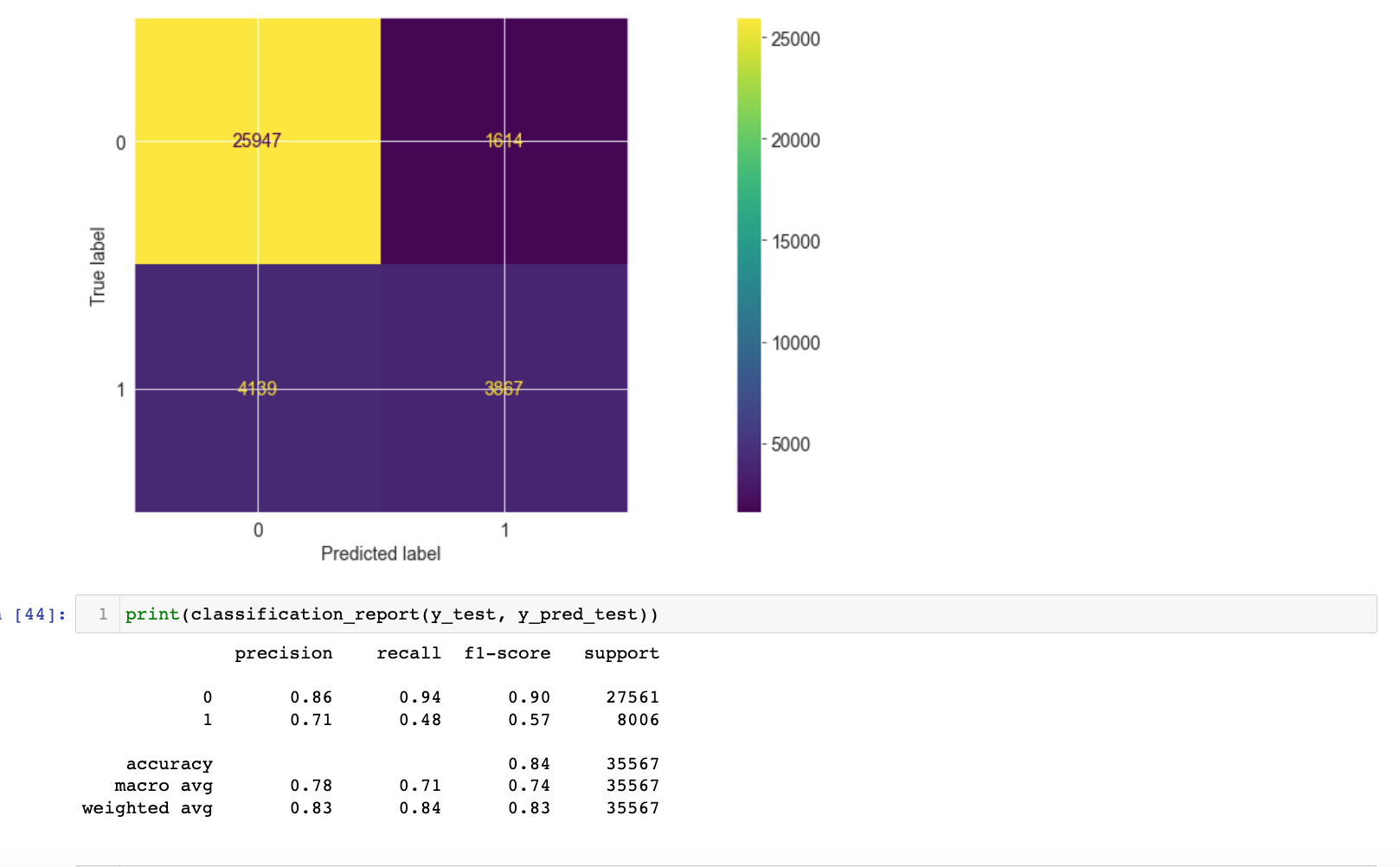
We can see an Accuracy score of **83.59%** is obtained by using the Logistic regression method.



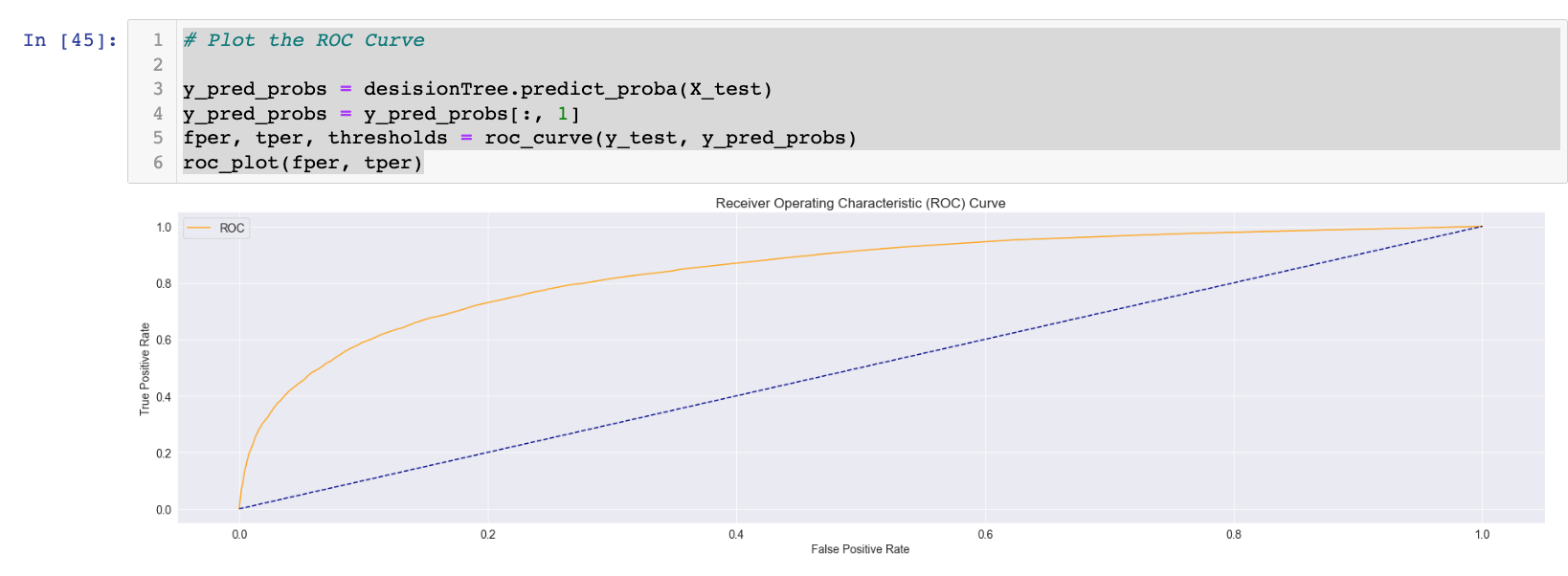
As part of more exploratory process, additional machine learning methods are also employed.

**Decision Tree Classifier**

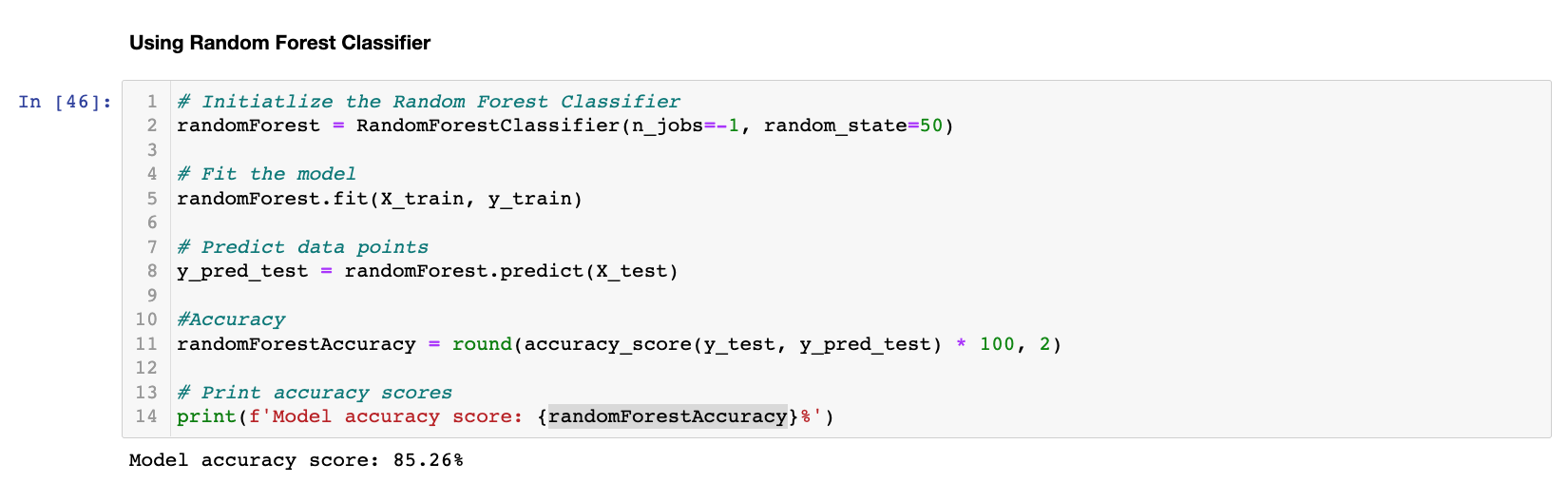


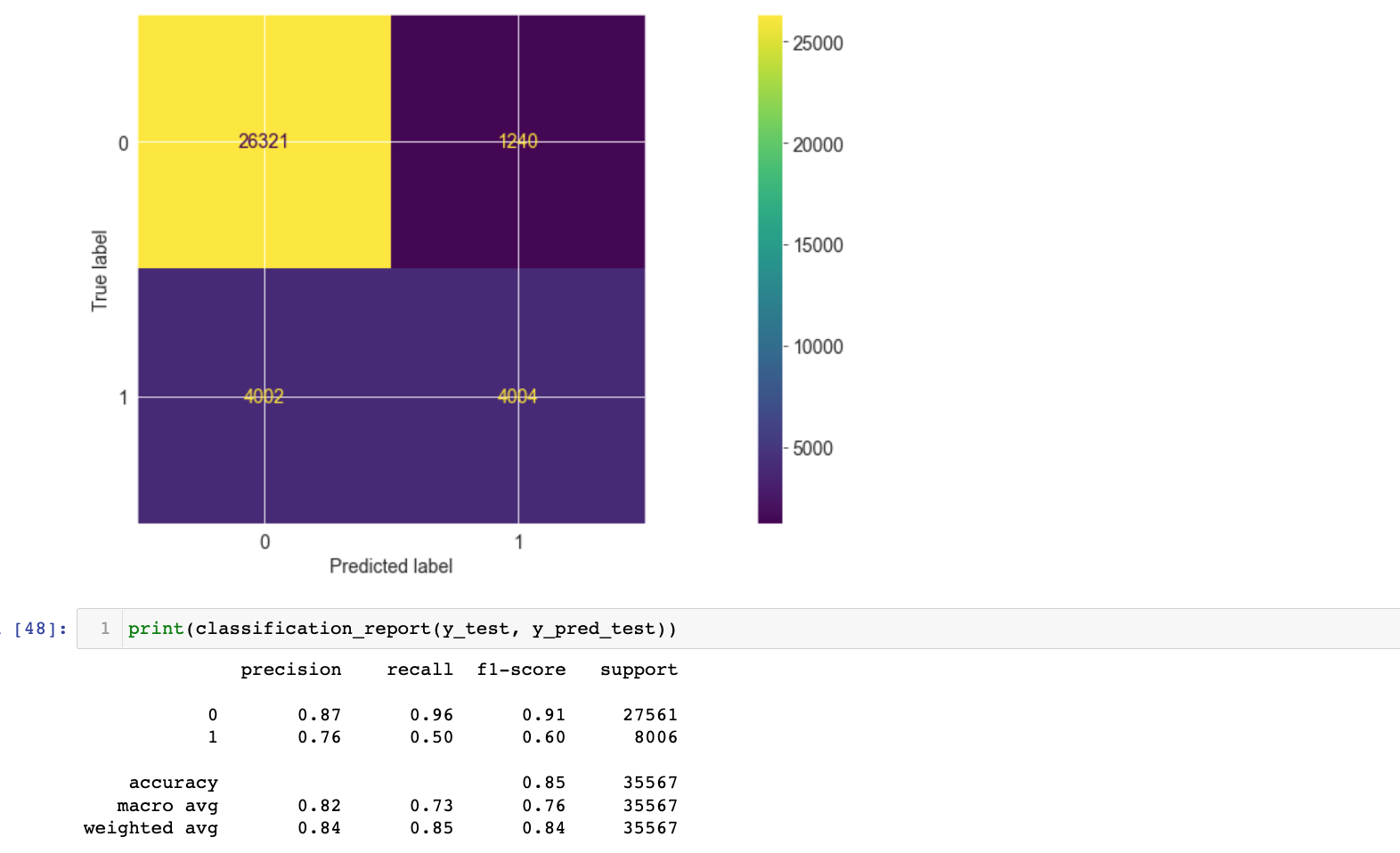


The accuracy obtained by using Decision tree is: **83.82%**

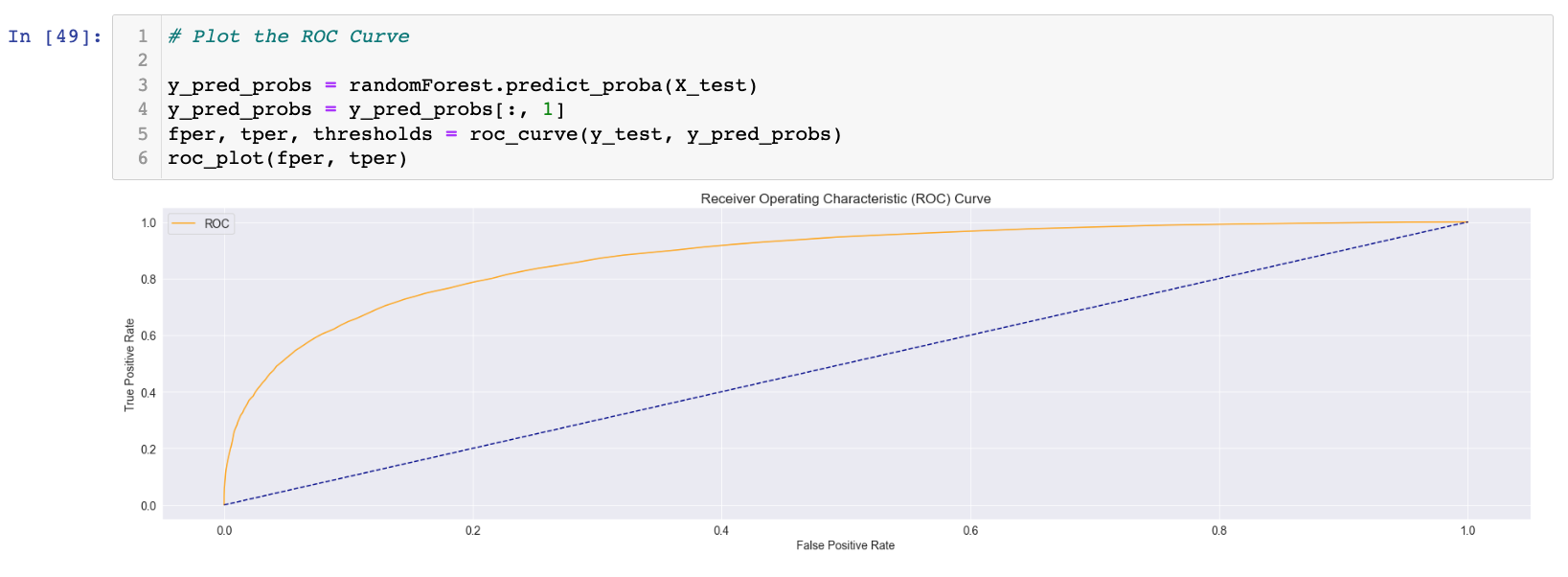
****

**Random Forest Classifier**

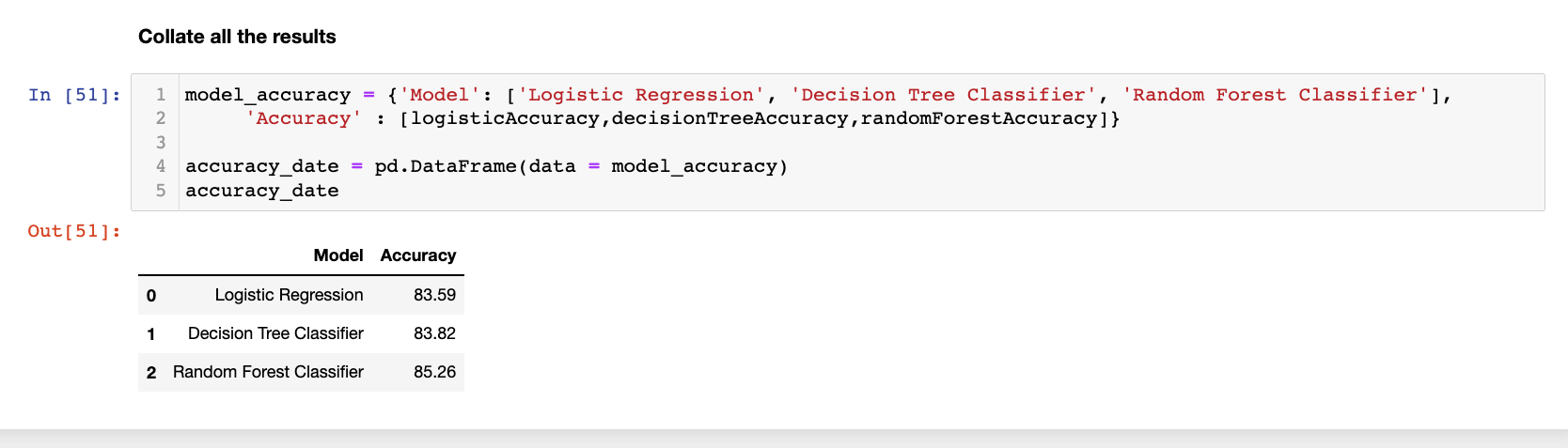




The accuracy obtained by using Random Forest Classifier is : 85.26%



The final modelling results are :



From the above picture, we can see that Random Forest classifier is giving the best accuracy results.

The Random forest based model has an accuracy of ~85. We can improve this by using hyper parameter tuning and feature engineering.

The precision and recall scores are good for days predicting no rainfall. The scores are very low for days predicting rainfall, this is due to an imbalanced dataset.

The F1-scores are good for days predicting no rainfall (0.91) while it is very low for days predicting rainfall (0.60).

Based on this, the model will predict most days as days with no rainfall.

Form the analysis of the ROC curve for Random Forest classifier, we can see that the occurrences will he high where the random forest classifier will be able to separate the positive and negative class values.

The reason for this is because the classifier has detected more number of TP(True positive values) and TN(True negative values) as compared to the FN(False negative) and FP(False positive values).

**Insights**:

* There were some NaN and NA values in the dataset which were cleaned post loading. The method of cleaning employed for Object datatypes and Float64 and Int data types was different. Also, 4 columns were removed from the dataset as they were having very high number of null values.
* There are 48 unique locations in the dataset, top 5 locations are Canberra, Sydney, Darwin, Perth and Brisbane.
* Portland, Caims and Walpole receives the highest rainfall on a frequent basis.
* Woomera, Canberra, Alice Springs receive the lowest rainfall on a frequent basis.
* From our Rainfall over the year’s chart, we can clearly see that Australia receives maximum rains during start of the year as compared to rest of the year. The pattern was bit different for year 2015 but it gradually improved.
* Max temperature is highly correlated to Temperature at 9am and Minimum temperature.
* Pressure at 9am is highly correlated to Pressure at 3pm.
* Cloud and Sunshine are negatively correlated, which also makes sense. Wind and Pressure are also negatively correlated, although the correlation is weak. Humidity and Cloud levels are positively correlated, with a weak correlation.

## References

1. <https://www.kaggle.com/jsphyg/weather-dataset-rattle-package>
2. [https://scikitlearn.org/stable/modules/generated/sklearn.metrics.classification\_report.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html)
3. <https://scikit-learn.org/stable/supervised_learning.html>
4. <https://seaborn.pydata.org/>
5. <https://pandas.pydata.org/>
6. <https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492>