* **Introduction:**

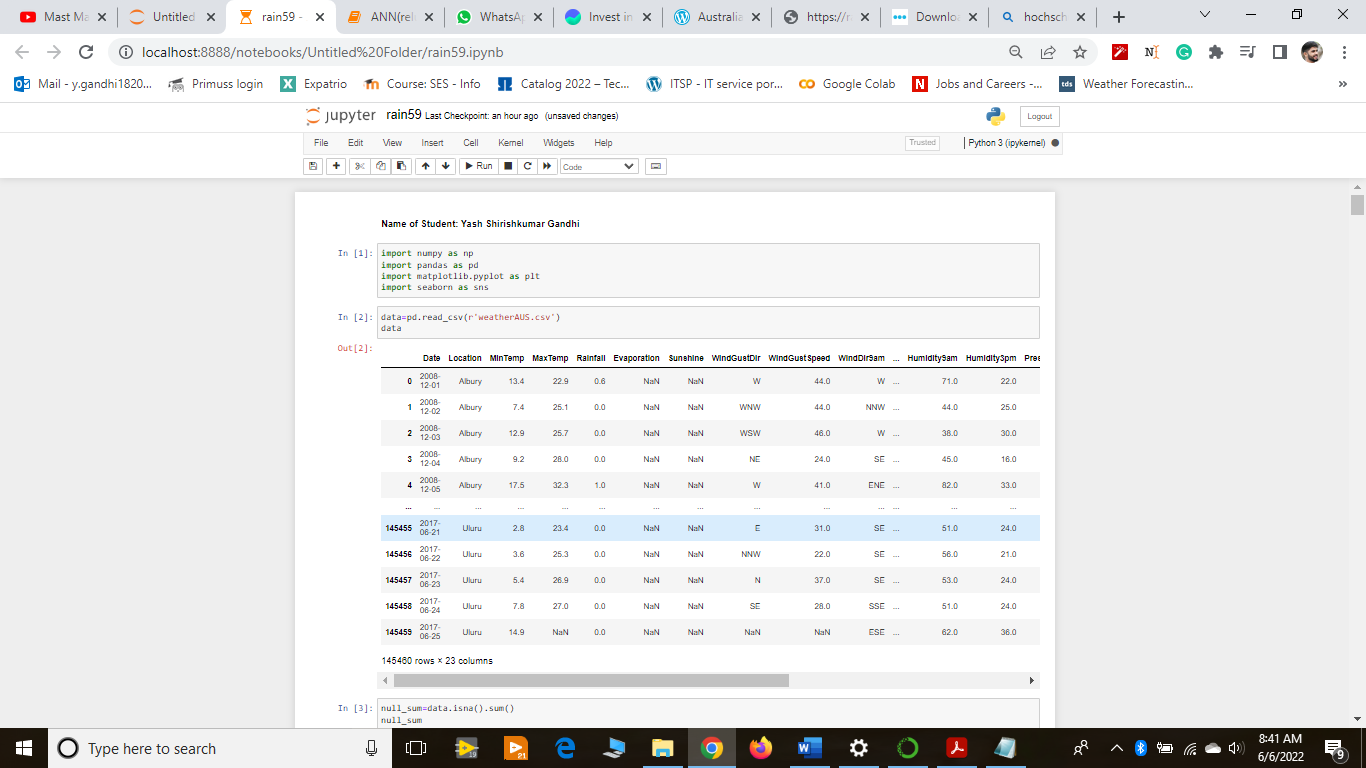
The data for the project has been provided by lecturer. This project covers the next-day rain prediction in Australia for the target variable Rain Tomorrow. The dataset contains about 10 years of daily weather observations from many locations across Australia. Rain Tomorrow is the target variable to predict i.e., will it rain next day- holding the values Yes or No. The column is yes if the rain for that day was 1mm or more. Therefore, this is a binary classification problem where we have only two outcomes either Yes or No (1 or 0). The data is in CSV format.

* **About the dataset:**

The data contains 145640 rows and 23 columns.

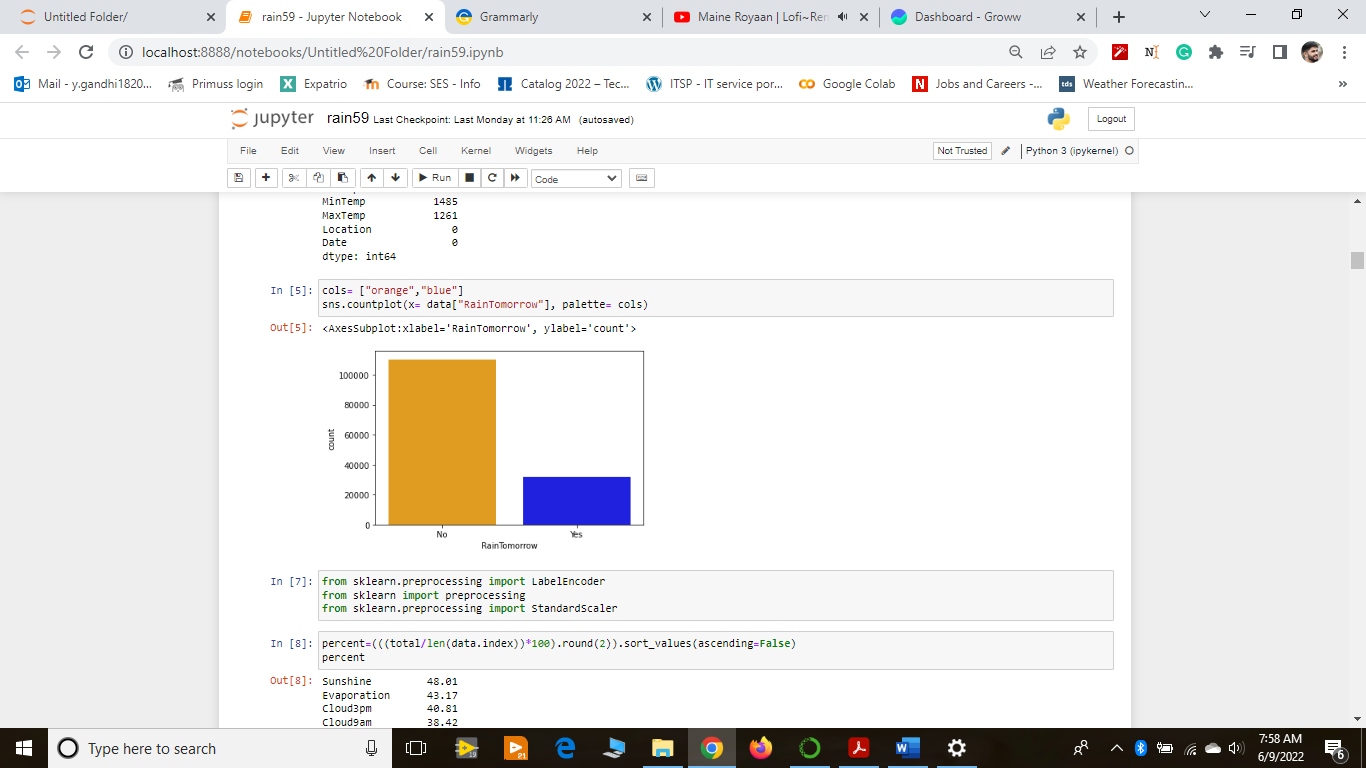
data = pd.read\_csv(r'weatherAUS.csv')

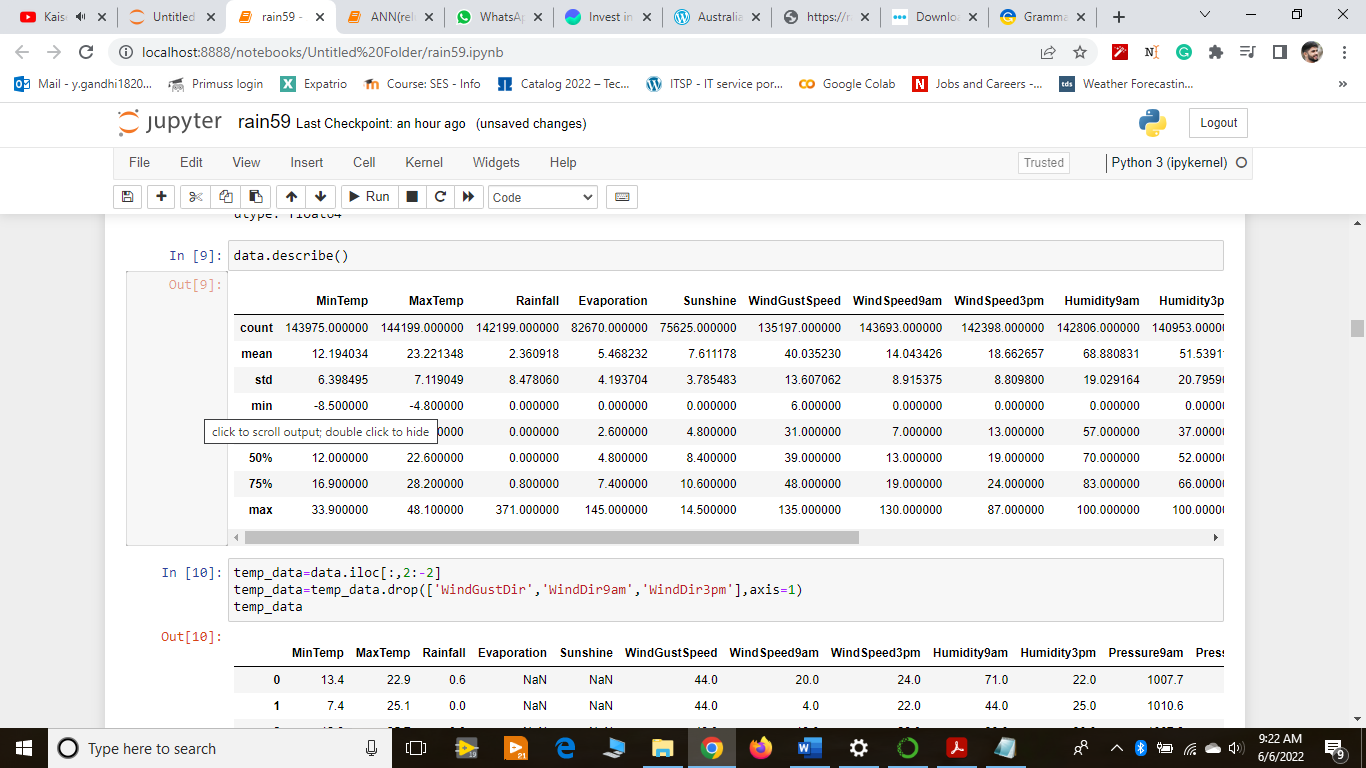
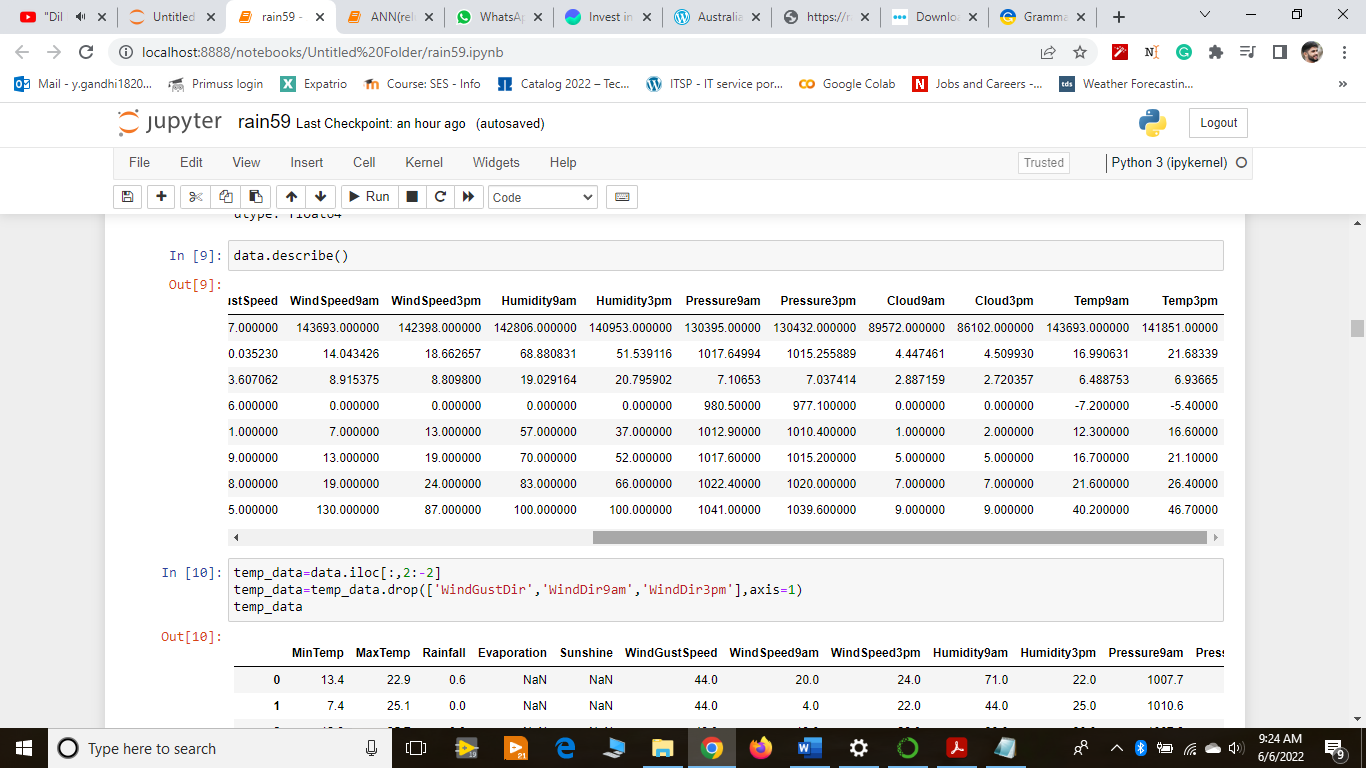
data

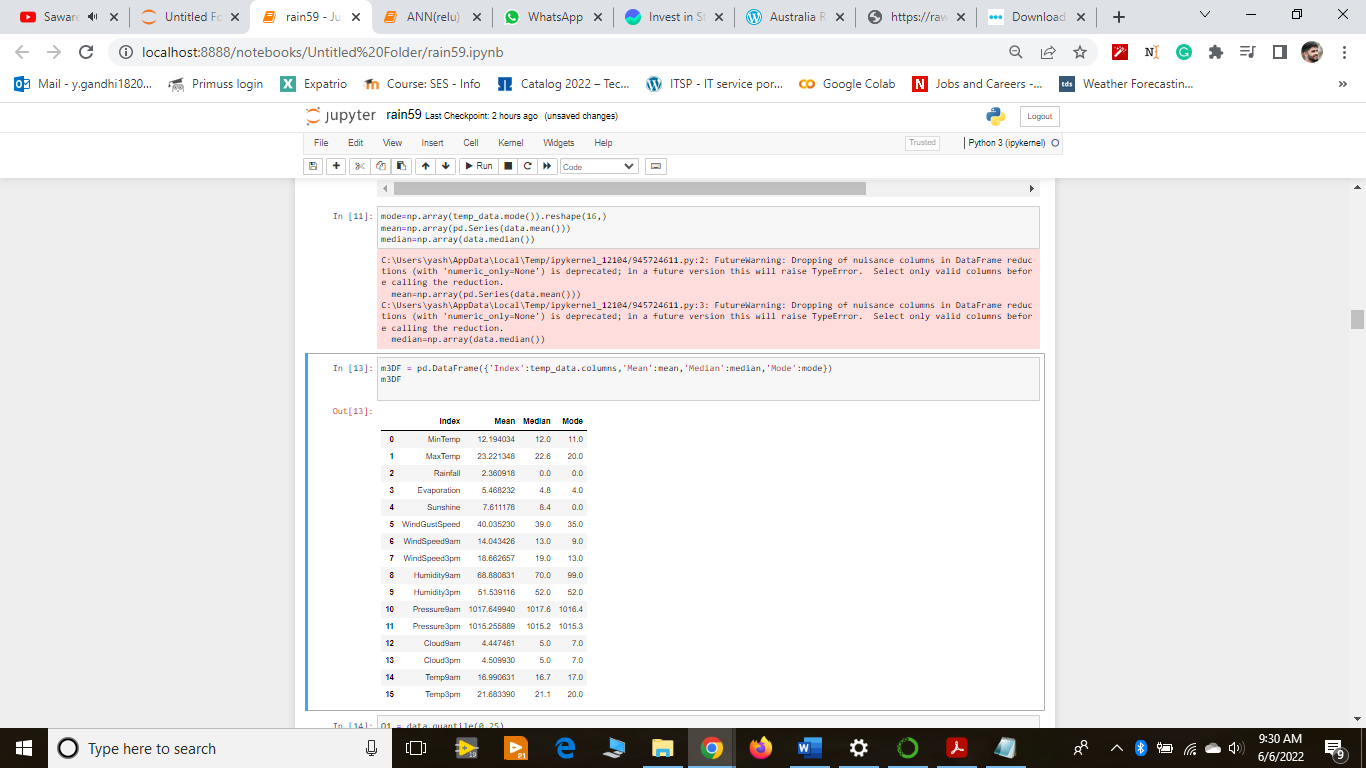


* **The data has the following columns:**

|  |  |
| --- | --- |
| **Column-Name** | **Description** |
| MinTemp | This is the minimum temperature recorded on that day by the weather station. |
| MaxTemp | This the maximum temperature recorded on that day by the weather station. |
| Date | This is the date for which the readings have been taken. |
| Location | This is the location in Australia for which the weather observations were recorded. |
| Rainfall | The rainfall recorded in that location on that date (All readings in millimeters) |
| Evaporation | This is the evaporation rate in the area for that date. |
| Sunshine | The measured intensity of sunshine by the sensors. |
| Industry | This is the sudden, brief increase in the speed of the wind followed by a lull. This column records the direction of the gust. |
| WindGustSpeed | The speed of the wind gust. |
| WindDir9am | This field records the direction of the wind at 9am. The directions are: E, W, N, S, means the wind flows from East, West, North and South direction respectively. NW,SW SE, NE respectively mean that wind flows from North West, South West, South East and North East directions respectively. ENE means that the wind is coming from halfway between East and North East(i.e. an angle of 56.25 to 78.75 degrees).ESE means angle halfway between East and Southeast directions (i.e. between 101.25and 123.75 degrees). Similarly, the rest of the values that were recorded (NNE, NNW,SSW, SSE, WNW, WSW) can be comprehended. |
| WindDir3pm | The readings for the direction of wind at 3pm. |
| Humidty9am | Percentage humidity recorded at 9am. |
| Humidity3pm | Percentage humidity recorded at 3pm. |
| Pressure9amandPressure3pm | These are the readings of atmospheric pressure at 9am and 3pm. |
| Cloud9amandCloud3pm | The clouds recorded by weather station at 9am and 3pm respectively. |
| Temp9am andTemp3pm | The readings of temperature on Celsius scale at 9am and 3pm respectively. |
| RainToday | This is the field which tells whether it rained today or not. So, it holds binary values, encoded as string i.e. Yes or No. |
| RainTomorrow | The field which tells whether it would rain tomorrow or not. So, this fi eld also records only two values Yes or No. |

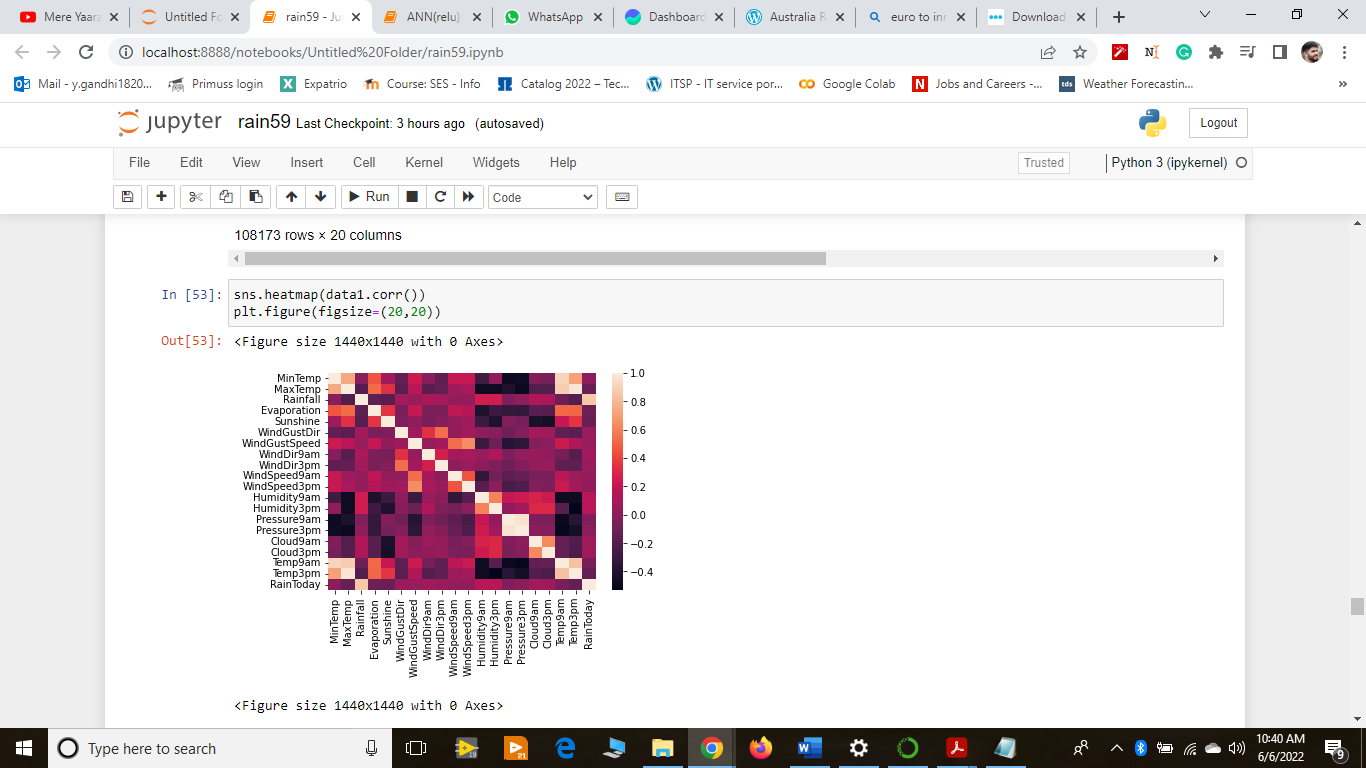
Afterthat I added chart in which we can see the count of Raintomorrow. 

* A glance at the data, that describes the count of valid values, mean, standard deviation, mini-mum value, 25th percentile, 50th percentile,75th percentile, and maximum value. 
* Measures for central tendency: the mean, median, and mode-Mean is described as the average value of the data i.e., we add up all the values in the data and divide the sum by the count of those values. Median is the center value (middle value) in data when it is arranged in ascending order. Mode is the most common number in the set.



* The Problems in the Data:

When I used the correlation heatmap of the Seaborn library provided by Python, the resultant heatmap was quite complex where many of the fields were highly correlated with each other. Hence, a clear pattern of a direct relationship could not be seen in the data.



* Solution for Complex Correlation Problem:

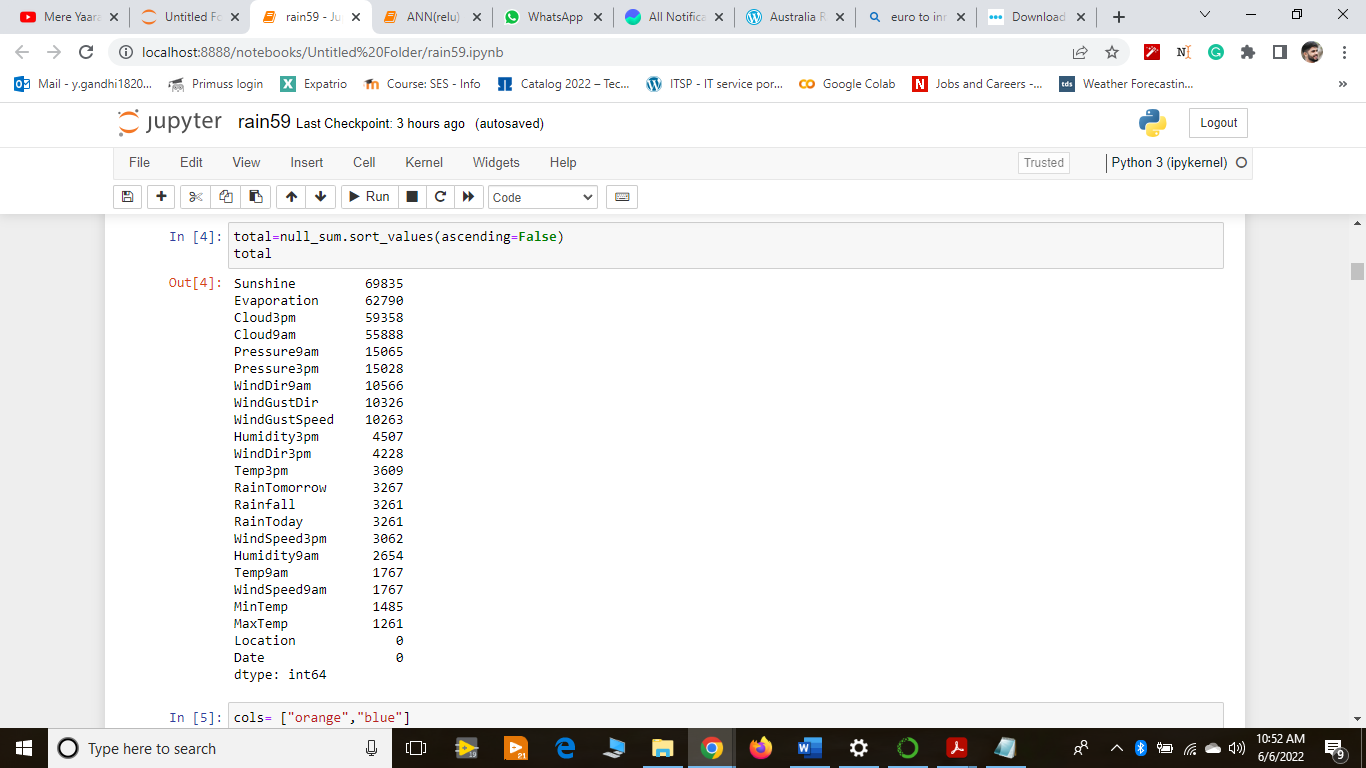
Therefore, I chose to apply deep learning to this dataset to understand the hidden patterns in the data and achieve a better score. I used an Artificial Neural Network(ANN) to solve this binary classifier cation problem.

1. The Missing Value Problem:

The data has a lot of missing values in most of the columns. The data might not have been recorded at the weather station for various reasons.

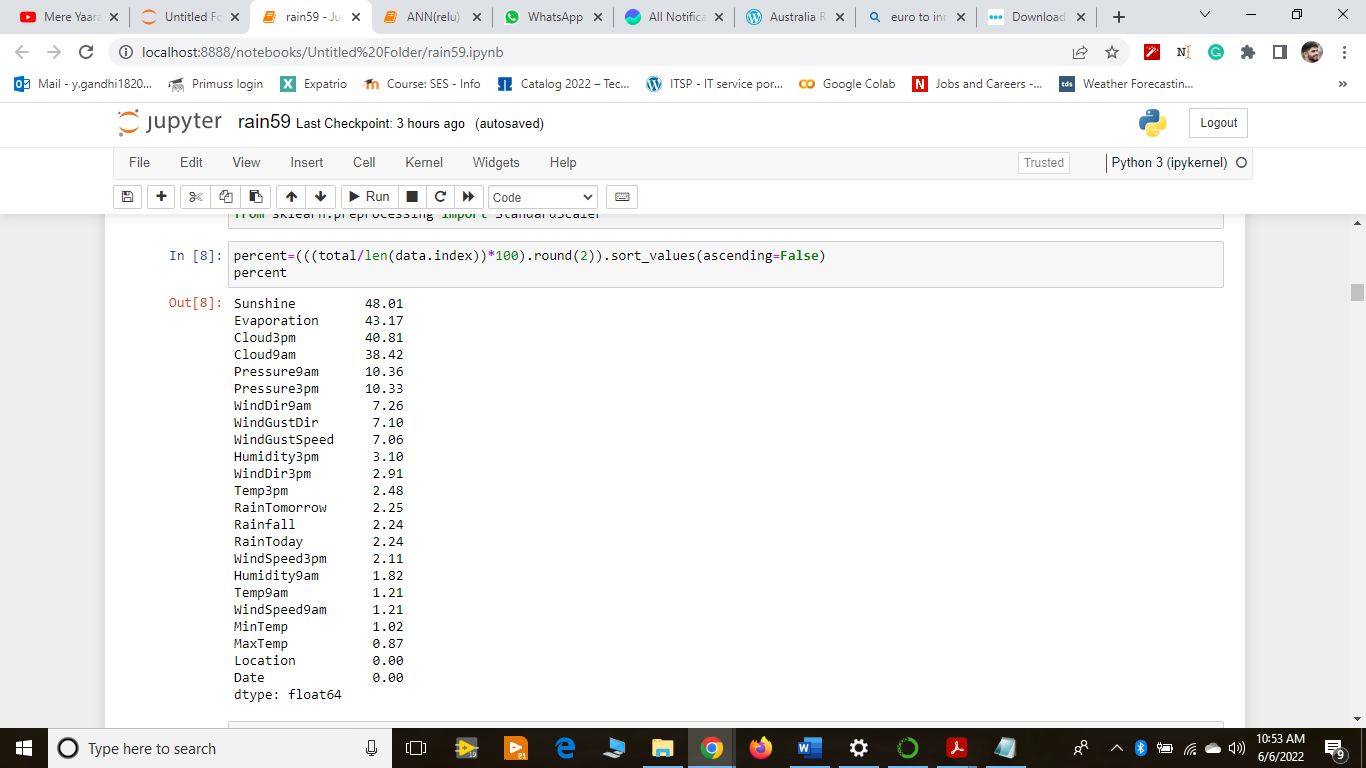
This leads to data misrepresentation as any analysis performed on the data would lead to underrepresentation of the data. If none of the values were missing, the most basic statistical analysis should look entirely different. The further predictions that are to be made will also be hence affected by the bias that the missing values may cause.

When I analyzed the data for missing values, the fields contained these many missing values:



Date and location had no missing values while minimum temperature and maximum temperature had1485 and 1261 missing values respectively. Similarly, we can interpret how many missing values we have for each column.

Since the count of missing values doesn’t make it any easier to comprehend what they convey for each column, I calculated the percentage of missing values in every column to get a clear picture of what share of values are missing for each column and hence how severely or moderately they affect the data for further analysis.



Solution for missing values:

**1.Removing the rows that contain missing values:**

One of the easiest solutions to this problem is that I entirely remove the missing values from the dataset.But this creates another problem. If I remove the rows that have missing values for Sunshine fi eld, then I would be removing 48.01% percent of the data in its entirety. What this means is that if the data contains1,50,000 rows rounded, then I would be removing approximately 50% of the data resulting in approx.75000 rows to process. But when I do so, I am compromising on most of the important data. Let’s say, for example, wind direction at a particular time is an important deciding factor in whether it will rain or not, and when I remove 50% of rows to eliminate missing values of Sunshine, there is a fair chance that I might eliminate most of the wind directions that result in rain. And removing them would mean that the model that I would train on this data would only know when it doesn’t rain but wouldn’t know when it would rain (basis wind direction).

Another approach to removing missing values could be considering fields that contain a low percentage of missing values. When I say low, I mean the fields that account for less than 5% of missing values in the data. This way we would not be eliminating a large share of data. In this dataset, Humidity3pm, WindDir3pm, Temp3pm, RainTomorrow, Rainfall, RainToday, WindSpeed3pm, Humidity9am,Temp9am, WindSpeed9am, MinTemp, and MaxTemp fi fields have missing values less than 5% in decreasing order.When I tried removing all the rows for all the columns that accounted for less than 5% of the data, I ended up with around 80000 rows in the dataset, which means that I still eliminated around 50% of the data. This happened because the rows that contained missing values for these columns didn’t coincide, which resulted in an elimination of quite many rows. This again would impact the final ANN model I would train in a not-so-good way.

**A final word on what fields should be removed from Australia Rain Prediction Dataset:**

The above two scenarios communicate that removing data for missing values, in this case, is not an option. But there are two fields for which we can remove missing values, namelyRainTodayandRainTomorrow. They account for 2.24 and 2.25 percent of missing values respectively. And eliminating them from the dataset seemed a good choice to me because firstly, they had a low ratio of missing values, and secondly, if for given information if it is not clear whether it would rain or not, the associated information holds no relevance. In simple words, these were removed because the outcome is unknown. If we don’t know whether a certain amount of humidity or temperature or the wind direction had resulted in rain or not, that piece of information is irrelevant to the model and in general. So, removing them is a very safe choice.

data.dropna(subset=['RainToday','RainTomorrow'],inplace=True)

**2.Replacing the missing value with mean, median, or mode:**

* There are a few points worth noting in this section before we proceed further in this section:
* Median can be calculated only for numerical variables or for sequenced categorical variables (not present in this case)
* Mean can be calculated only for numerical variables
* The mode can be calculated for both numerical and categorical variables alike.
* The skewness of the data is a measure of how much the data of a random variable deviates from its normal distribution (towards left or right).
* When mean and median are approximately equal, it is safe to assume that the skewness of the data is 0 or approaches 0.
* The data is said to be left-skewed when the distribution of the data seems to be stretched out to the left. In terms of the measures of central tendency, we can say that the data is said to be skewed to the left when the mean is less than the median and they are both less than the mode.

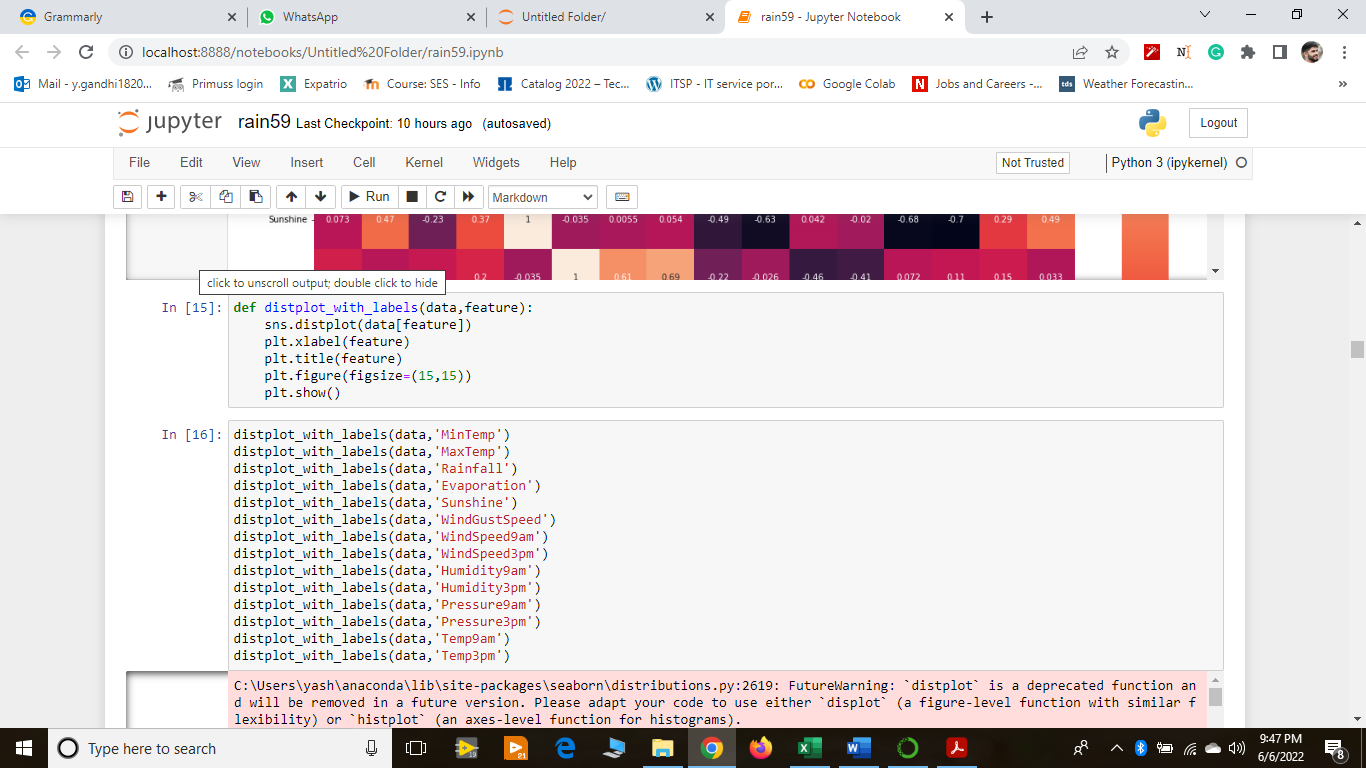
Mean<Median<Mode

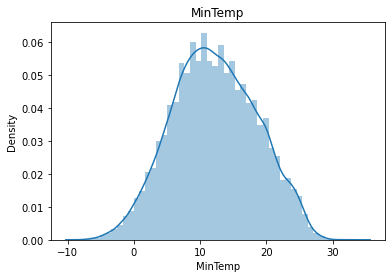
* The data is said to be right-skewed when the distribution of the data seems to be stretched out to the right. In terms of the measures of central tendency, we can say that the data is said to be skewed to the right when the mean is greater than the median and they both are greater than the mode.

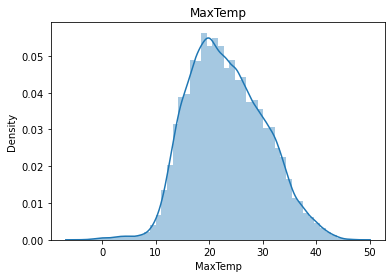
Mean>Median>Mode

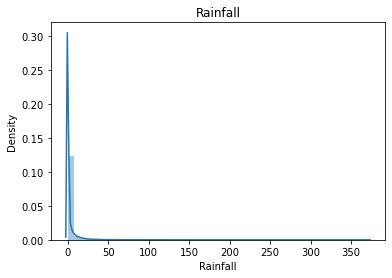
* When the data is symmetric i.e. skewness is 0, then we can replace the missing values with the mean. This is called mean imputation.
* When the data is skewed, we can replace the data with the median. This is called median imputation.
* In the case of categorical data, the concept of mean and median doesn’t mean anything. So, we must replace the missing values with the mode of the variable. This is called mode imputation.
* The choice between median and mode imputation for skewed data is based on the instinct of the analyst.

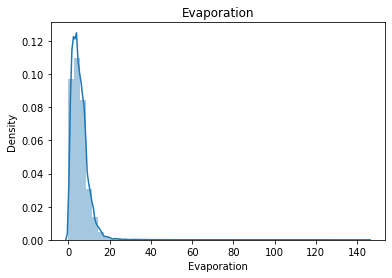
For skewness of the model, I plotted ed distribution plots from a seaborn library of Python which looks like this:

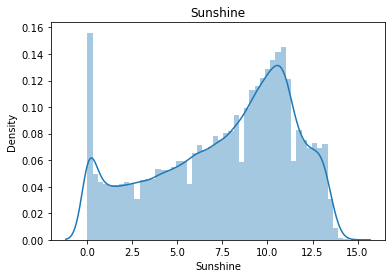


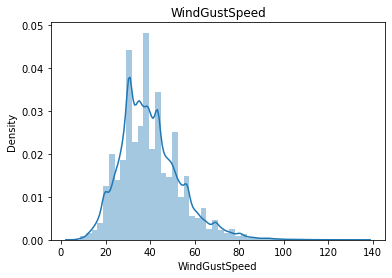


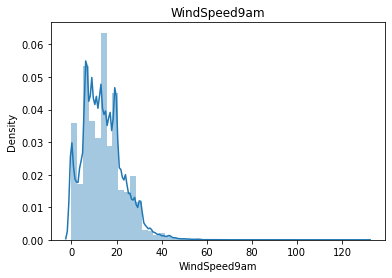


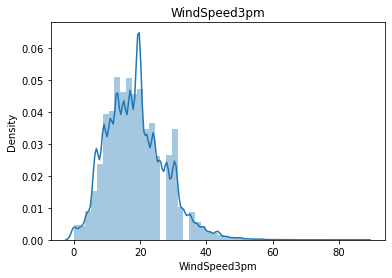


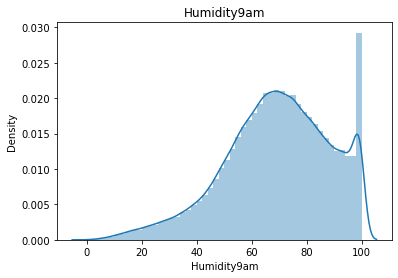


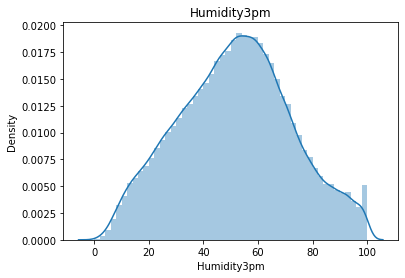


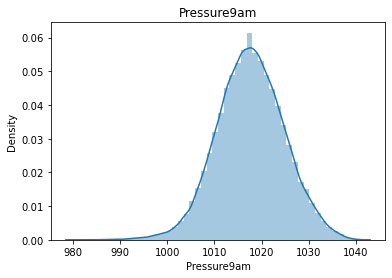


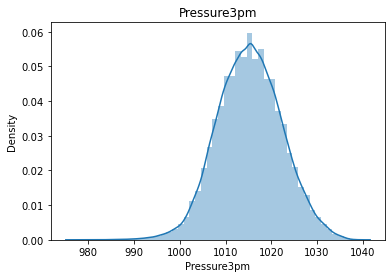


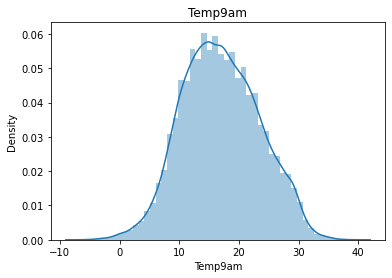


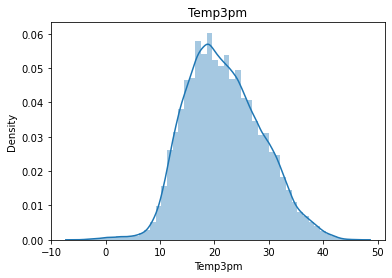












The Outlier Problem:

In very simple terms, outliers are those data points which are very far from other data points.

In statistical terms, outlier are the data points which fall below or above the upper or lower limit. The upper and lower limits are given by:

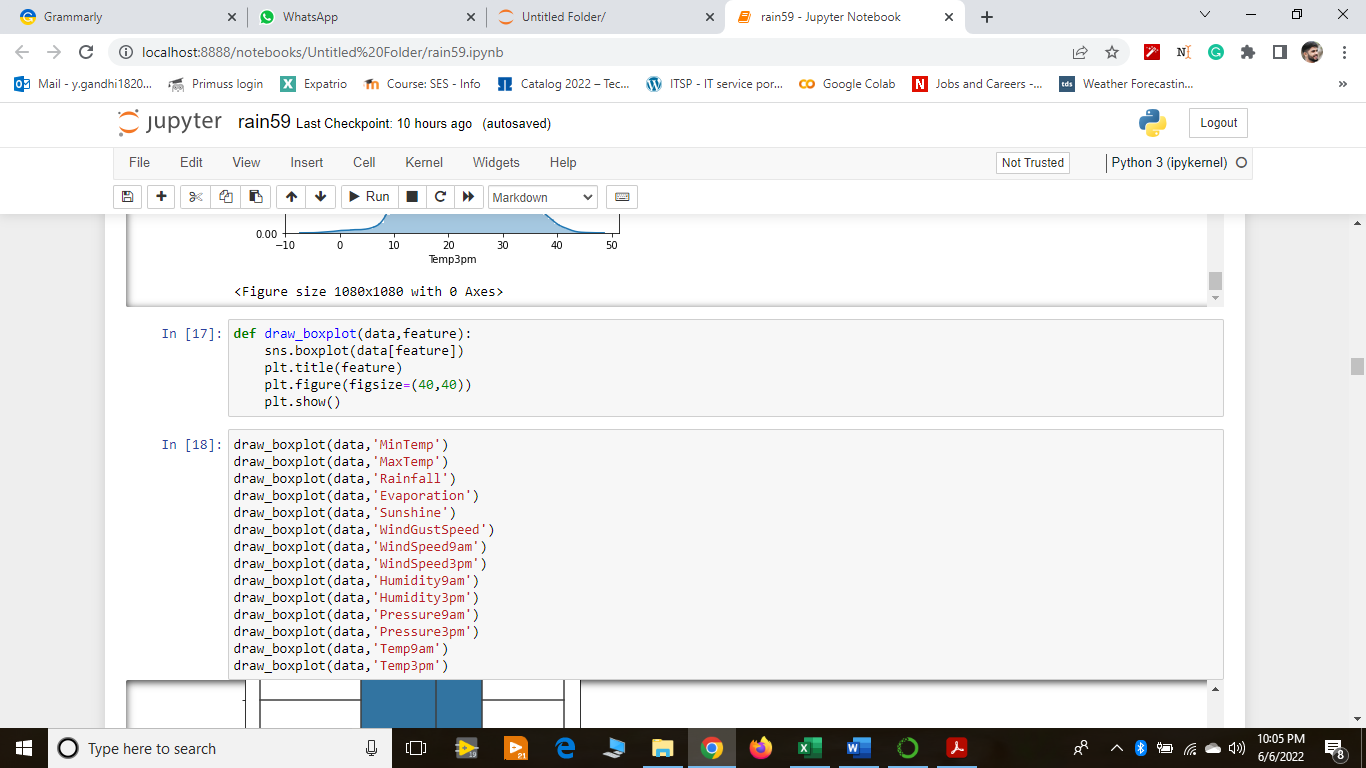
Upper Limit= Q3+1.5\*IQR

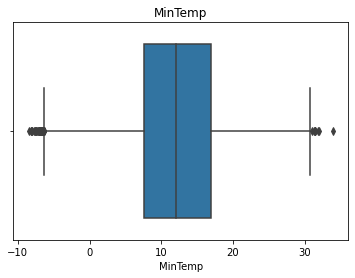
Lower limit=Q1-1.5\*IQR

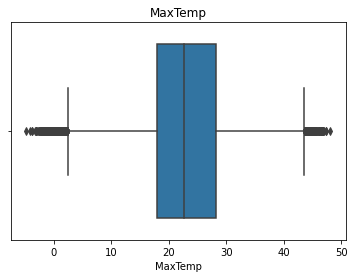
where IQR=Q3-Q1,

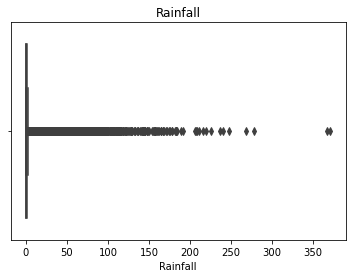
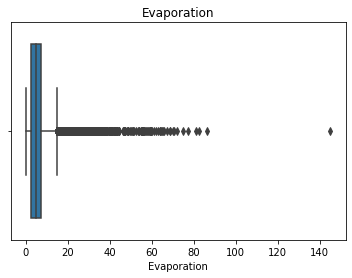
Q3 is the 75 percentile and Q1 is the 25 percentile.

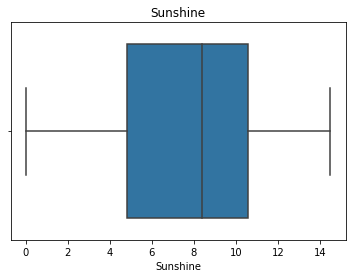
In this problem, I used the boxplot method to visualize the outliers, and the IQR method to remove the outliers.

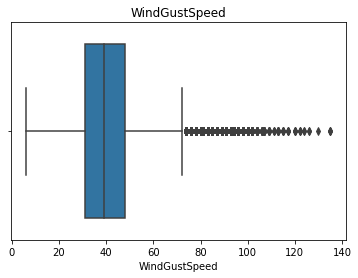
The boxplot for various fields is as given which tells there are so many values that are outliers.

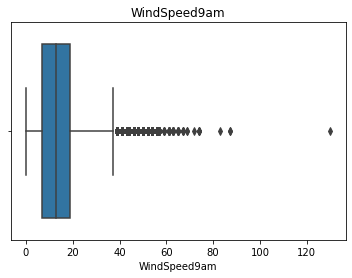


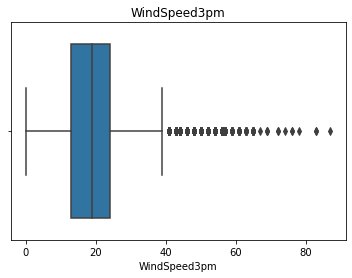


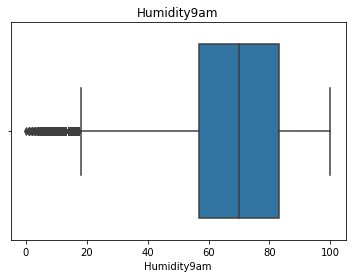


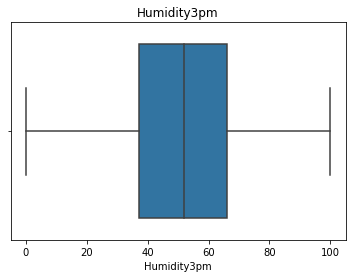


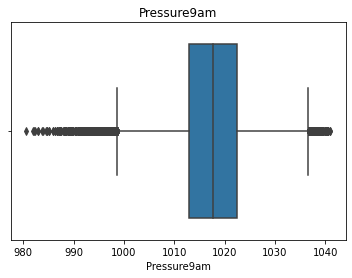


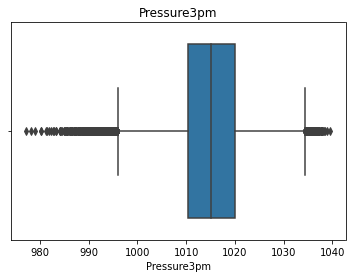


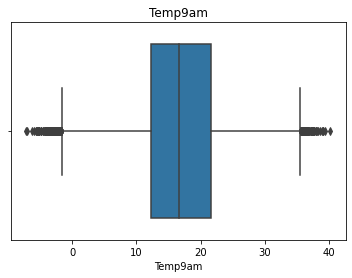


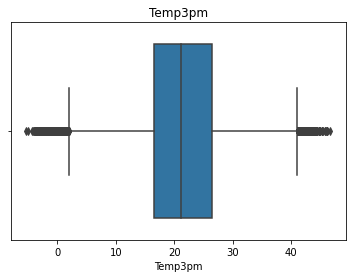








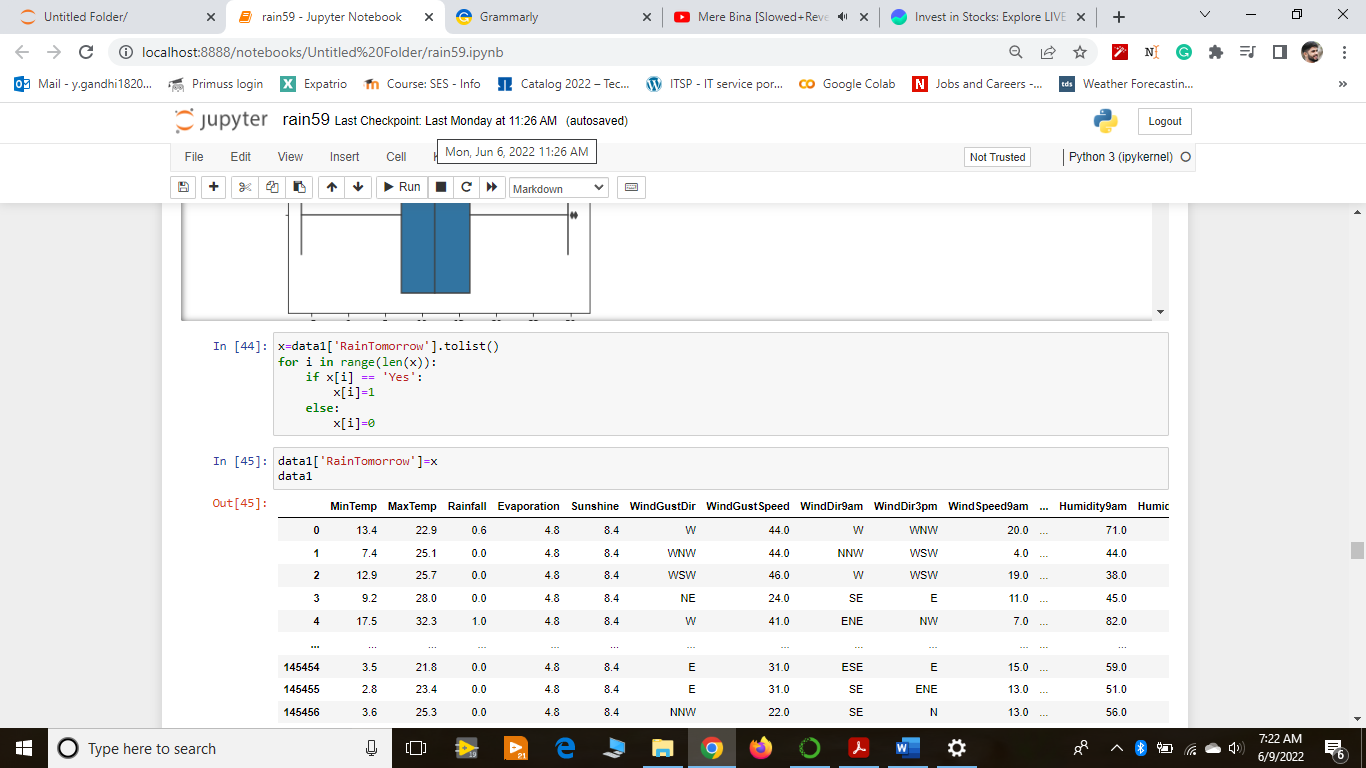


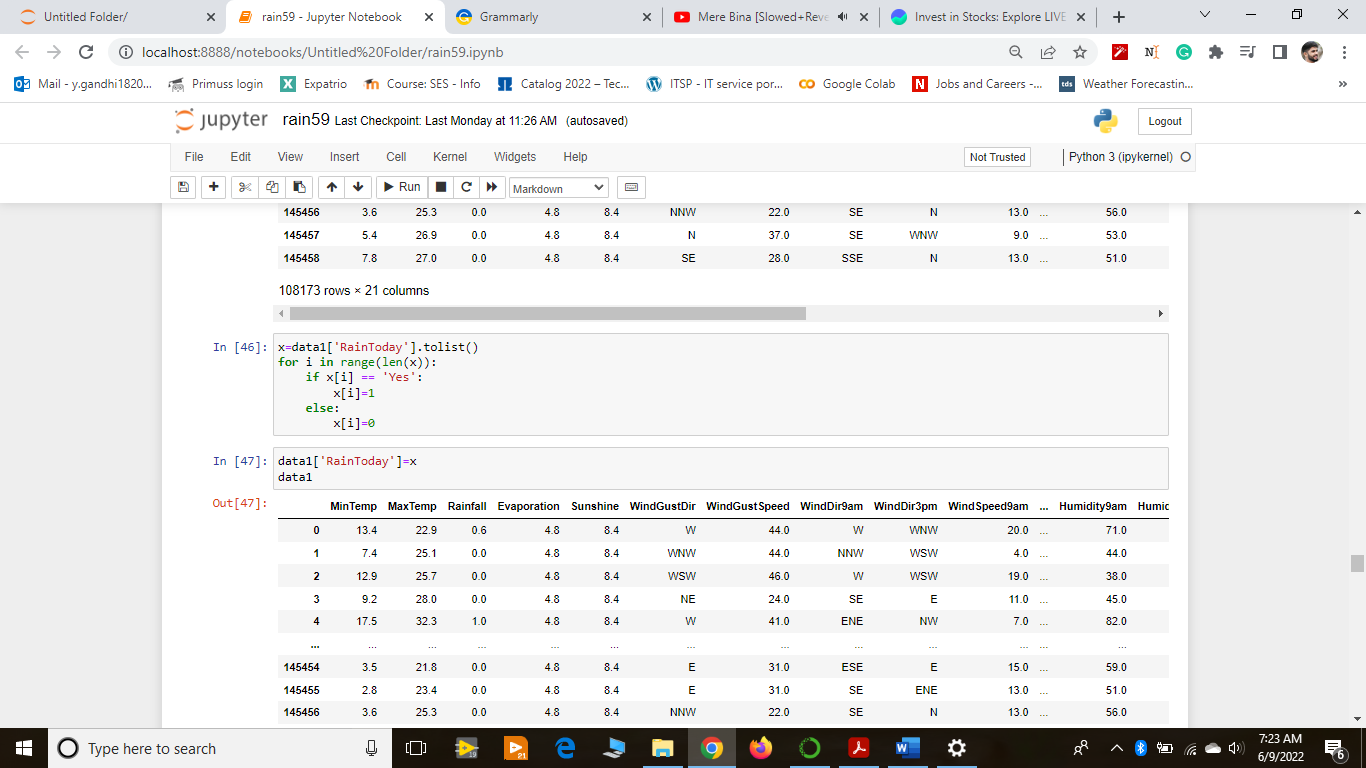


**Encoding categorical variables as numbers:**

We now have a few categorical variables which are in string format like direction and RainToday and RainTomorrow. For the system to process these variables, we need to encode them in numerical format.

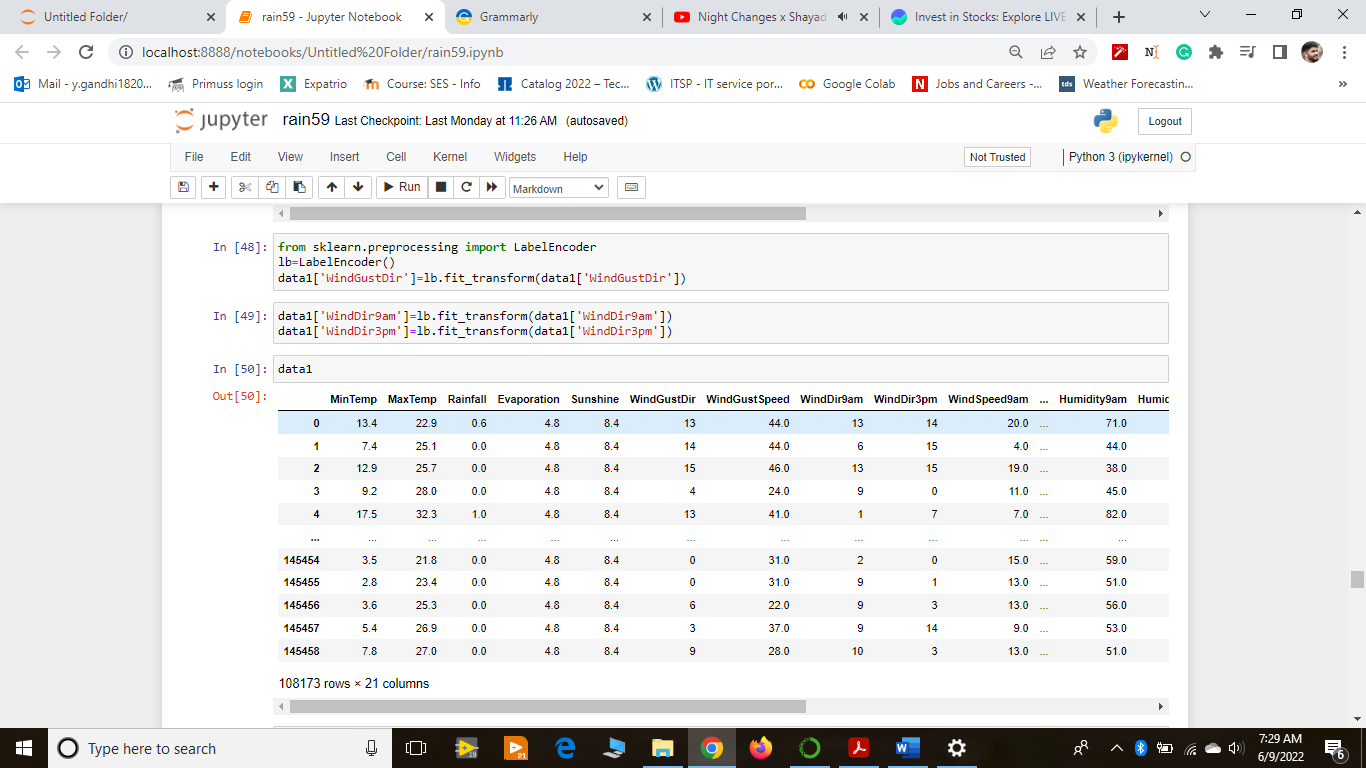
For binary variables RainToday and RainTomorrow, I have encoded **Yes**with value 1 and **No**with value 0.





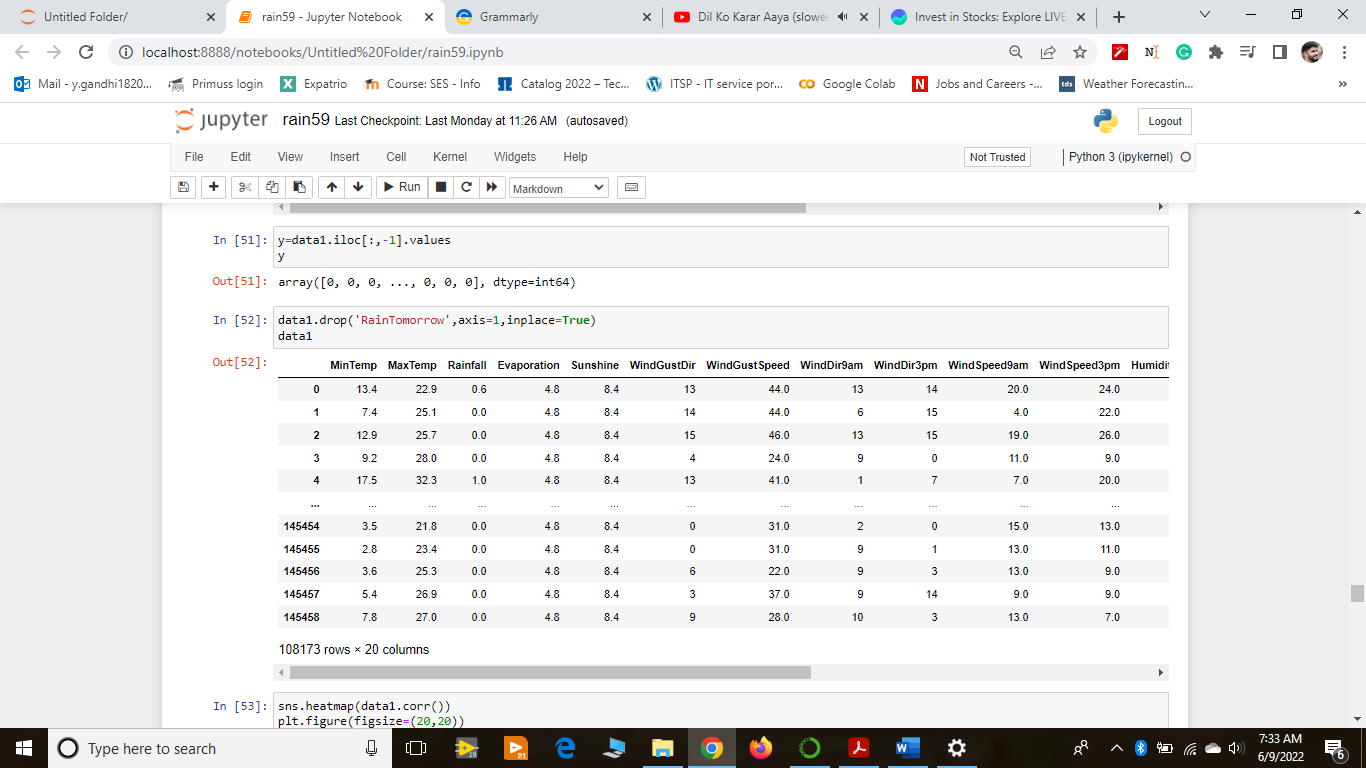
For variables that indicate wind direction, I have used LabelEncoder from scikit-learn library of python.I have used a LabelEncoder on WindGustDir, WindDir9am and WindDir3pm columns.

LabelEncoder assigns a numeric value to each category in the field so that the model can process the data. Each category has a unique numeric value.

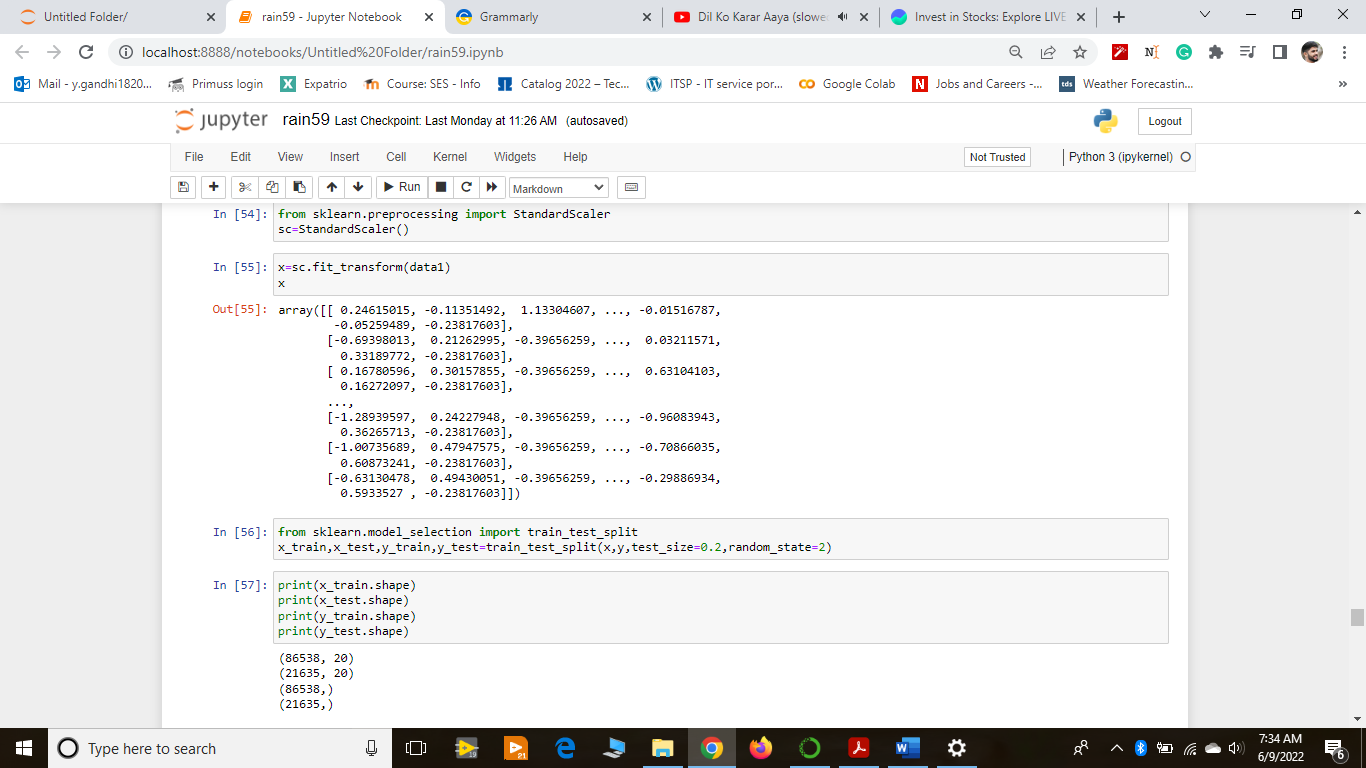


**ANN model for Australia Rain Prediction Dataset:**

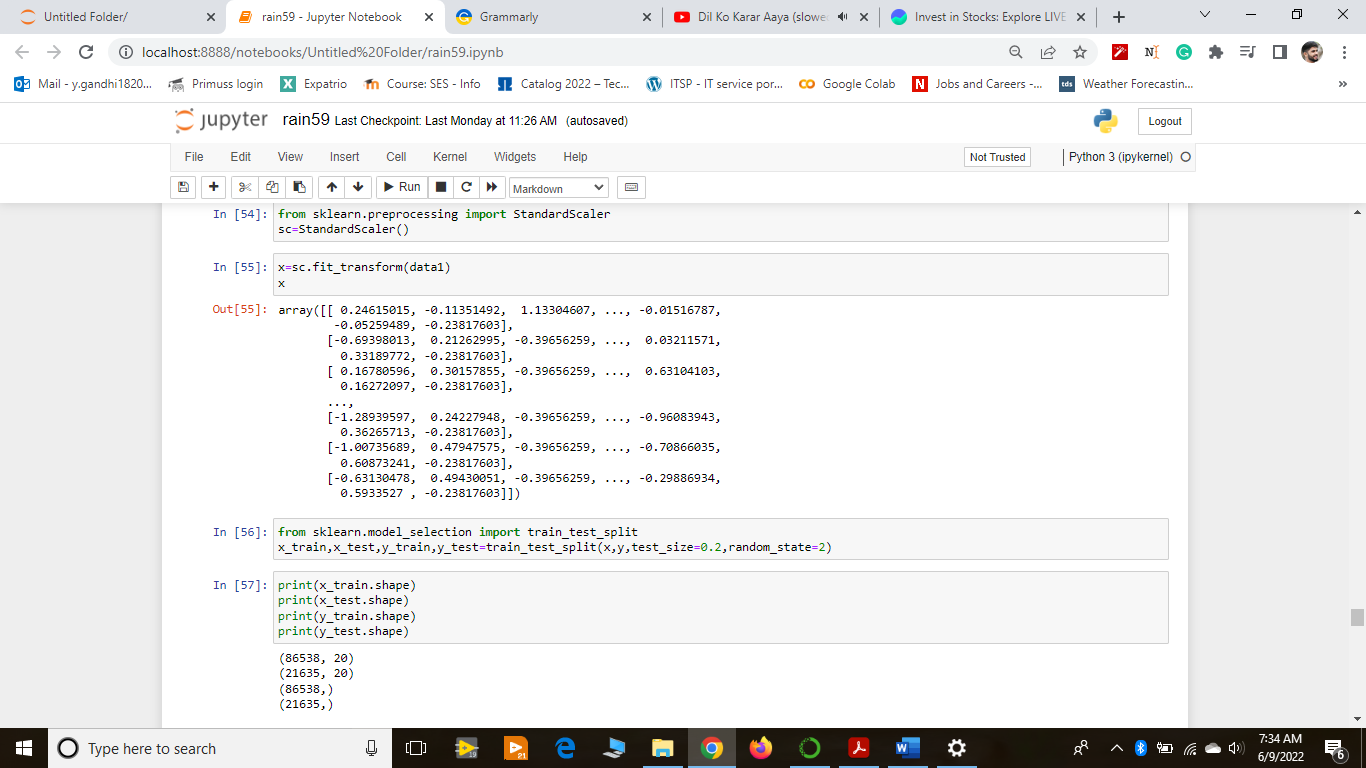
This problem is a binary classification problem, where we are supposed to predict whether it will rain or not.So, RainTomorrow column is the output variable.Also, I am not considering location and date variables for my model. We can categorize dates into seasons and use the seasons as a deciding parameter for whether it will rain or not. But for the sake of simplicity, I decided to skip these variables.This means my dependent variable is RainTomorrow and independent variables are all the variables except for RainTomorrow, Date, and Location.



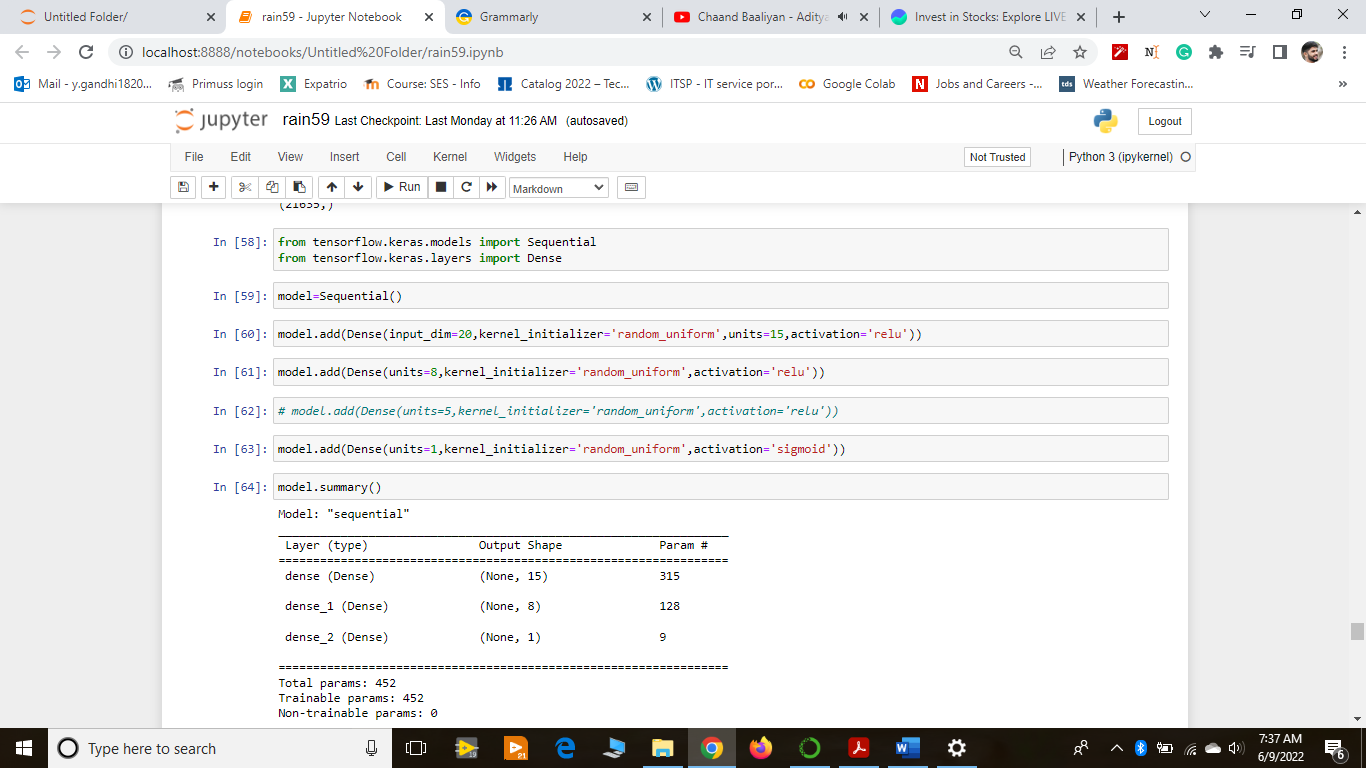
After this, I will fit my data into StandardScaler from sci-kit-learn and split my data for train and test. I used the 80:20 rule for splitting the data into train and test which means 80% of my data is used for training purposes and 20% of data is used for testing purposes.



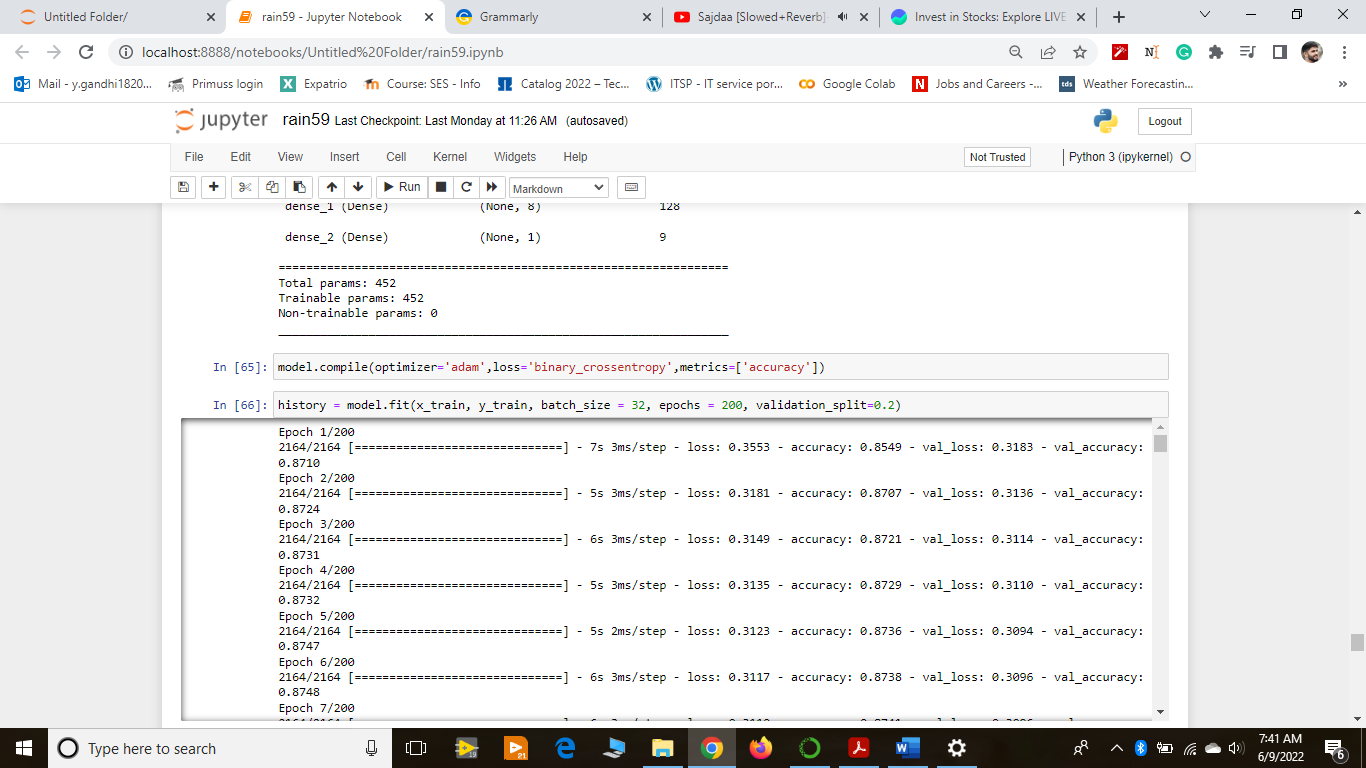
Hence, I get this shape for my train and test data:



I have used Sequential Model for this purpose. Then I added a dense layer to the model with 15 units and a relu activation function. The second dense layer has 8 units i.e. the output shape is 8. The third dense layer is to flatten the output. This is the final layer of the model which gives the output. Since the output is in the form of either 1 or 0, that is why I have used a sigmoid activation function for the 3 layer.

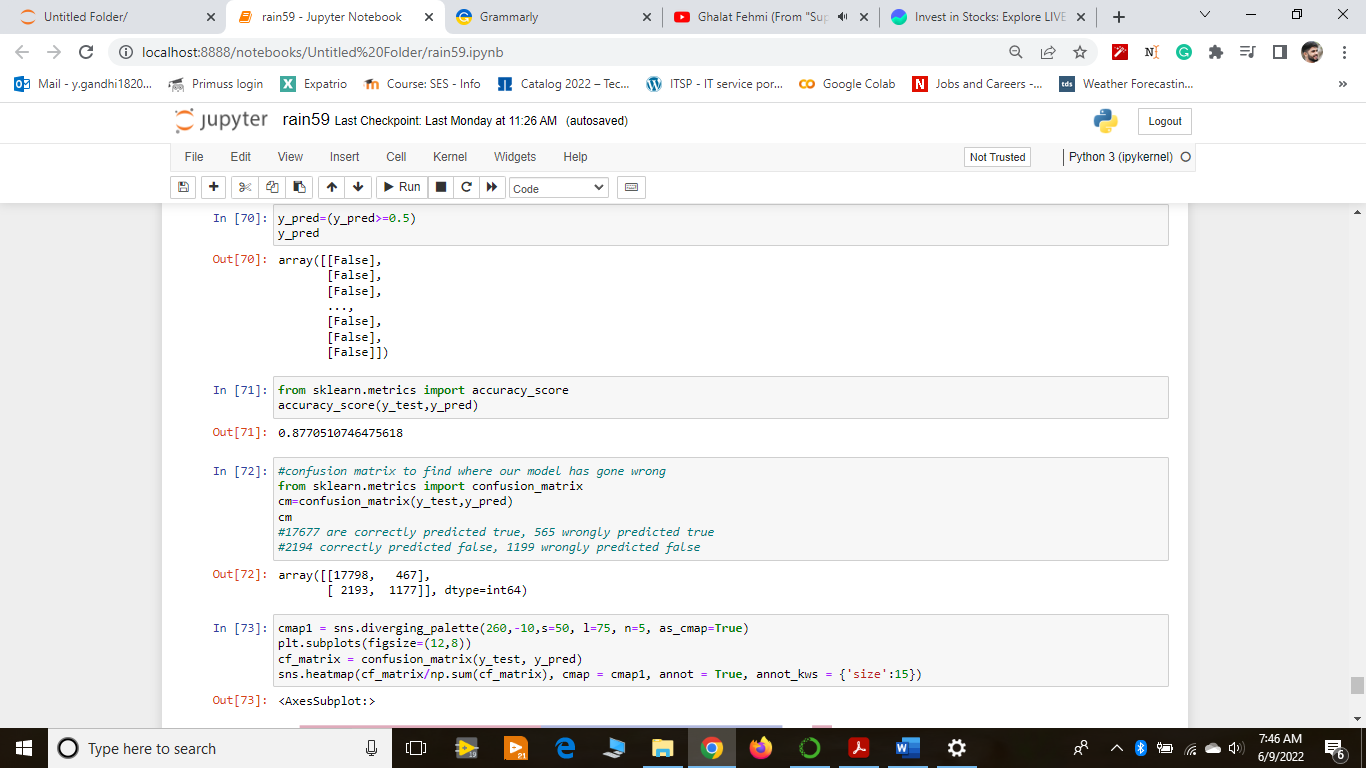


Here you can find the summary of my sequential model I used.

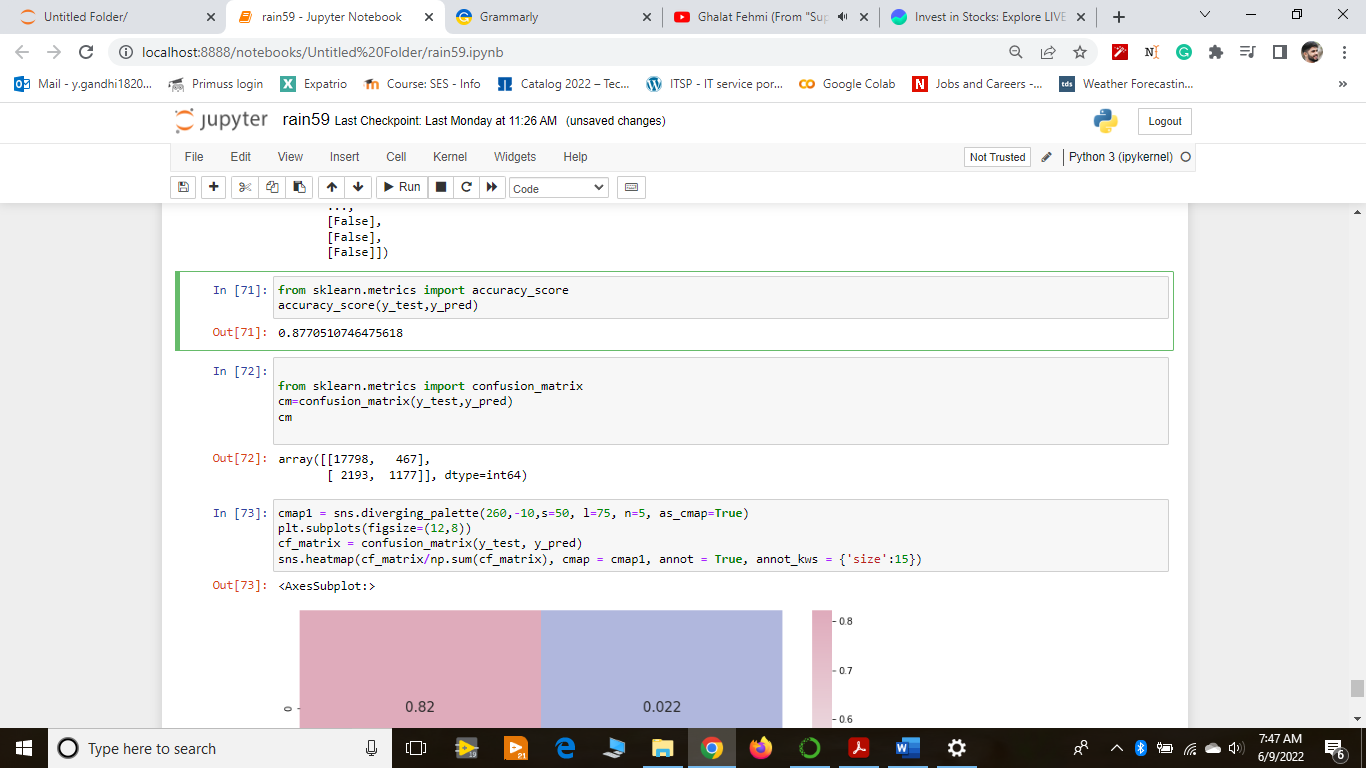
The model is compiled using **‘adam’** optimizer and loss function used is **‘binary\_crossentropy’.** 

I have used 20 epochs for training purpose and a batch size of 32.

After the model is successfully trained, I used predict() method on my model to predict the value of RainTomorrow variable. If the predicted value is greater than equal to 0.5, then it means that the model has predicted that it will rain tomorrow, else it will not rain.



The accuracy score of my ANN model is around 87.58% which is a good score.



The confusion matrix of the trained model looks like this.

