Toronto

Module2: EDA of Rain Forecasting in Australia

Dhairya Dave[a] (002110382)

Parth Savalia[a] (002982303)

Manan Soni[a] (002982645)

[a] *College of Professional Studies, Master of Professional Studies in Analytics.*

***Subject**: ALY6040*

Under the guidance of

Prof. Kamran Hootan

Introduction:

This data is from Kaggle with titles “Rain in Australia”. By gathering quantitative information about the atmosphere's current condition at a specific location and utilising meteorology to predict how the atmosphere will evolve, weather forecasts are created. To forecast whether or not it will rain tomorrow, use the Rain Dataset. About 10 years' worth of daily weather measurements from several Australia locales are included in the dataset. As of seeing this, this will be use for classification purposes which can be said as weather forecasting. Speaking of weather predictions, they might have an impact on daily activities as well as industries like the food sector, tourism, emergency healthcare, etc. There is a target variable that can be either "Yes" or "No”. Yes, if there was at least 1mm of rain that day. The variables in our dataset that are most likely to cause rain to fall are pressure, humidity, pressure, clouds, and sunlight. Finding a link between them that is supported by statistical evidence is therefore essential for predicting when it will rain.

Problem statement for this dataset are as follows:

a) Using classification algorithms like Random Forest, Gradient Boosting and XgBoost to create prediction models that predicts whether or not it will rain tomorrow with more accuracy.
b) To check if we can scale the model to forecast rain for a weekly basis or for the day after tomorrow.

Fig. 1 Information of Dataframe

```
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Date           145460 non-null  object
 1   Location       145460 non-null  object
 2   MinTemp        143975 non-null  float64
 3   MaxTemp        144199 non-null  float64
 4   Rainfall       142199 non-null  float64
 5   Evaporation    82670 non-null   float64
 6   Sunshine       75625 non-null   float64
 7   WindGustDir    135134 non-null  object
 8   WindGustSpeed  135197 non-null  float64
 9   WindDir9am     134894 non-null  object
 10  WindDir3pm     141232 non-null  object
 11  WindSpeed9am   143693 non-null  float64
 12  WindSpeed3pm   142398 non-null  float64
 13  Humidity9am    142806 non-null  float64
 14  Humidity3pm    140953 non-null  float64
 15  Pressure9am    130395 non-null  float64
 16  Pressure3pm    130432 non-null  float64
 17  Cloud9am       89572 non-null   float64
 18  Cloud3pm       86102 non-null   float64
 19  Temp9am        143693 non-null  float64
 20  Temp3pm        141851 non-null  float64
 21  RainToday      142199 non-null  object
 22  RainTomorrow   142193 non-null  object
dtypes: float64(16), object(7)
memory usage: 25.5+ MB
```

Fig.2 Description of Data

	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	\
count	145460.0	143975.0	144199.0	142199.0	82670.0	75625.0	
mean	24.0	12.0	23.0	2.0	5.0	8.0	
std	14.0	6.0	7.0	8.0	4.0	4.0	
min	0.0	-8.0	-5.0	0.0	0.0	0.0	
25%	11.0	8.0	18.0	0.0	3.0	5.0	
50%	24.0	12.0	23.0	0.0	5.0	8.0	
75%	36.0	17.0	28.0	1.0	7.0	11.0	
max	48.0	34.0	48.0	371.0	145.0	14.0	

	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	...	Humidity9am
count	145460.0	135197.0	145460.0	145460.0	...	142806.0
mean	8.0	40.0	8.0	8.0	...	69.0
std	5.0	14.0	5.0	5.0	...	19.0
min	0.0	6.0	0.0	0.0	...	0.0
25%	4.0	31.0	3.0	4.0	...	57.0
50%	9.0	39.0	8.0	8.0	...	70.0
75%	13.0	48.0	12.0	12.0	...	83.0
max	16.0	135.0	16.0	16.0	...	100.0

	Humidity3pm	Pressure9am	Pressure3pm	Cloud9am	Cloud3pm	Temp9am
count	140953.0	130395.0	130432.0	89572.0	86102.0	143693.0
mean	52.0	1018.0	1015.0	4.0	5.0	17.0
std	21.0	7.0	7.0	3.0	3.0	6.0
min	0.0	980.0	977.0	0.0	0.0	-7.0
25%	37.0	1013.0	1010.0	1.0	2.0	12.0
50%	52.0	1018.0	1015.0	5.0	5.0	17.0
75%	66.0	1022.0	1020.0	7.0	7.0	22.0
max	100.0	1041.0	1040.0	9.0	9.0	40.0

	Temp3pm	RainToday	RainTomorrow
count	141851.0	145460.0	145460.0
mean	22.0	0.0	0.0
std	7.0	0.0	0.0
min	-5.0	0.0	0.0
25%	17.0	0.0	0.0
50%	21.0	0.0	0.0
75%	26.0	0.0	0.0
max	47.0	2.0	2.0

Exploratory Data Analysis:

It is strongly advised to have a clean dataset in order to achieve better results, thus it is crucial to eliminate duplication, comprehend the dataset's structure, and assign the proper data type. For that same, we have appended the percentage of null values in the report.

```
Percentages of Null values in Features :
Sunshine         48.01
Evaporation      43.17
Cloud3pm         40.81
Cloud9am         38.42
Pressure9am      10.36
Pressure3pm      10.33
WindDir9am        7.26
WindGustDir       7.10
WindGustSpeed     7.06
Humidity3pm       3.10
WindDir3pm        2.91
Temp3pm           2.48
RainTomorrow      2.25
Rainfall          2.24
RainToday         2.24
WindSpeed3pm      2.11
Humidity9am       1.82
Temp9am           1.21
WindSpeed9am      1.21
MinTemp           1.02
MaxTemp           0.87                    ,
```

Fig 3 : Percentage of Null values in Features

Analyzing from above table we are focusing and aligning our resources to two target variables which are:

- RainTomorrow

Analyzing data of RainTomorrow we came up with amount of percentage of their values; which are:

No 75.839406
Yes 21.914616
NaN 2.245978

We need to remove null values from RainTomorrow as we need to predict and replacing with 0, mean or median can lead our model. It’s 2% so we can remove from our target variable from the d ataframe .

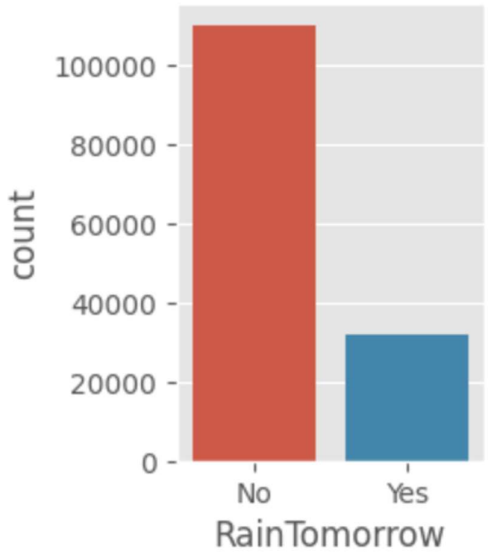


Fig. 4 Count of each value

This graph depicts the count of individual variable of the same attribute RainTomorrow. Number of count of NOs are relatively more than that of YES.

EDA for Categorical value:

Now, EDA is performed on categorical vales of the dataset. Out of 23 total attribute we have 7 categorical values. Below is the picture that represent the attribute in category.

```
Number of variables:  7
Variables : ['Date', 'Location', 'WindGustDir', 'WindDir9am', 'WindDir3pm', 'RainToday', 'RainTomorrow']
```

Below it displays the top 10 values of the categorical variables.

	Date	Location	WindGustDir	WindDir9am	WindDir3pm	RainToday	RainTomorrow
0	2008-12-01	Albury	W	W	WNW	No	No
1	2008-12-02	Albury	WNW	NNW	WSW	No	No
2	2008-12-03	Albury	WSW	W	WSW	No	No
3	2008-12-04	Albury	NE	SE	E	No	No
4	2008-12-05	Albury	W	ENE	NW	No	No
5	2008-12-06	Albury	WNW	W	W	No	No
6	2008-12-07	Albury	W	SW	W	No	No
7	2008-12-08	Albury	W	SSE	W	No	No
8	2008-12-09	Albury	NNW	SE	NW	No	Yes
9	2008-12-10	Albury	W	S	SSE	Yes	No

Table 1: Top 10 Value of Categorical variable

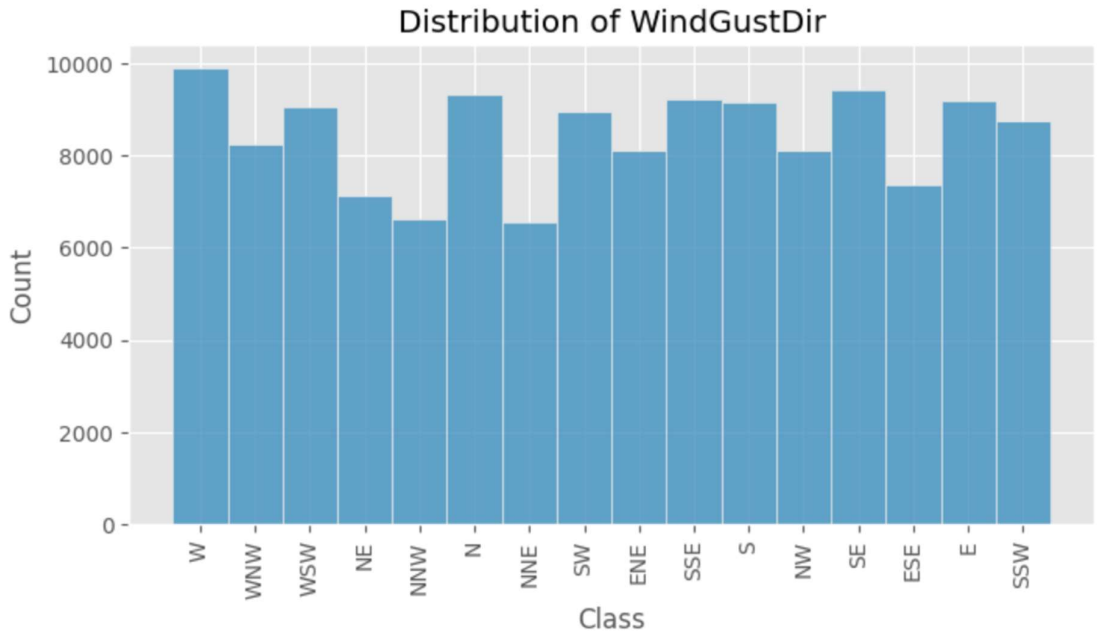


Fig 5: Distribution of WindGustdir

Here, this graph depicts the distribution of WindGustDir over the directions of wind gusting in that particular direction. Take aways from the following graph is that the graph distribution is more or less uniform all around.

Likewise, distributions of WindDir3pm and WindDir9am has shown same response as uniform distribution.

Cardinality

The number of distinct values allocated to a dimension is referred to as its cardinality. A certain number of distinct values are specified for some dimensions. Cardinality checking is important because it specifies the relation between other categorical variables also, if we discover high cardinality in a dataset then it will provide an extremely large matrix which in result makes building of model extremely difficult or it will cause under-fitting. Depicting from the graph we can see that location has the highest count of unique categories which we will use for advance classification.

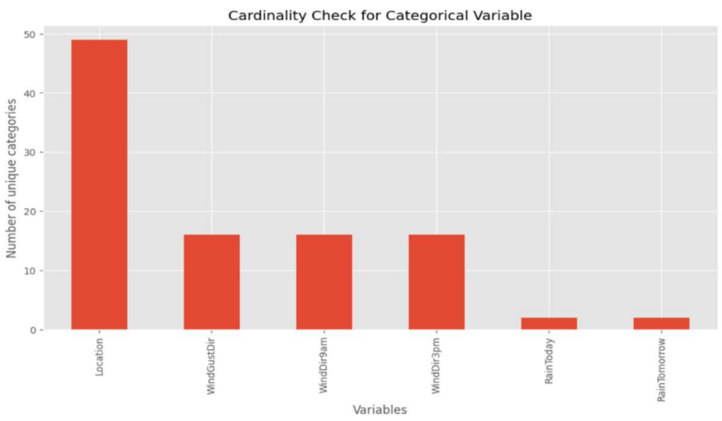


Fig 6: Percentage of null values of categorical values

```
Null % in categorical variables:
WindDir9am      7.264
WindGustDir     7.099
WindDir3pm      2.907
RainTomorrow    2.246
RainToday       2.242
Date            0.000
Location        0.000
```

Fig 5: Null percentage in categorical variable

For cleaning Rainfall and RainToday of the categorical variables, we have to be sure that as we cannot just simply impute the data with mode and median only as it depends on the rainfall and hence we have to check whether the Rainfall has happen or not, with respect to that only we have to clean the data for RainToday. Hence, we have imputed the values on that condition only.

	Rainfall	RainToday
0	0.6	Yes
1	0.0	No
2	0.0	No
3	0.0	No
4	1.0	Yes
...	...	...
145455	0.0	No
145456	0.0	No
145457	0.0	No
145458	0.0	No
145459	0.0	No

Table 2: Cleaning of rainfall

As per the procedure we corrected Null value to mean of Rainfall and at same place we are replacing Yes with NaN for RainToday. As there is 1 mm of rain we need Yes in RainToday.

EDA for Numerical Features:

Firstly, we display name of the numerical data and thereby along with that we also defined each integer to a data type equal to float64

```
['MinTemp',
 'MaxTemp',
 'Rainfall',
 'Evaporation',
 'Sunshine',
 'WindGustSpeed',
 'WindSpeed9am',
 'WindSpeed3pm',
 'Humidity9am',
 'Humidity3pm',
 'Pressure9am',
 'Pressure3pm',
 'Cloud9am',
 'Cloud3pm',
 'Temp9am',
 'Temp3pm']
```

Fig 7: Numerical value

Clean data is necessary which is input for our model. When it comes to imputation of missing values for numeric data, we need to check the distribution. In addition, two methods of mean and median, we have to decide whether mean will be more suitable or median for a particular numeric variable. This selection of method is determined on the basis of distribution of each variable. If the graph is normally distributed then we impute the value with respect to mean and when the distribution is right or left skewed, we opt for imputing the data with respect to median.

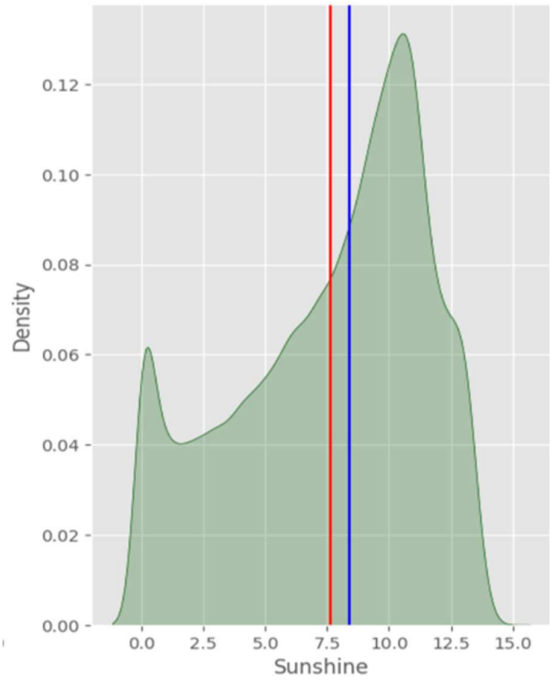
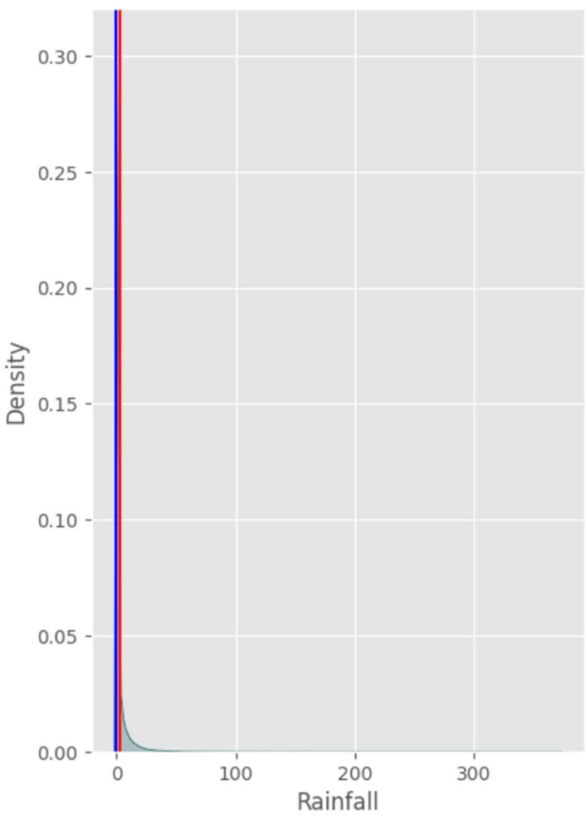


Fig 8: Distribution of Sunshine

In this distribution of sunshine, the graph is unevenly distributed and weighted right side. And we can see that red line which is mean of the sunshine is present at the left side and hence we opt for mean to impute data.



Here in this distribution of Rainfall we can see that data is having 0 for a lot of samples. Hence, for this mean and median both on the same line hence any of that is preferred to get imputed with.

Likewise, every distribution has its own mean and median and we have to select which suits better. For most of the data of the numeric value we used mean.

Fig 9: Distribution of Rainfall

Collinearity and Feature Engineering

Primarily, we need to factorise features into numeric. To add to that, we need to plot a heat map or correlation chart, to determine the collinearity between the feature.

As these features are realistic there are hardly strong correlation between the variable. So we need to calculate the variation inflation factor (VIF) to identity multicollinearity between each variable. There are categorical variable with more unique value or cardinality, it can be ambiguous to understand for collinearity.

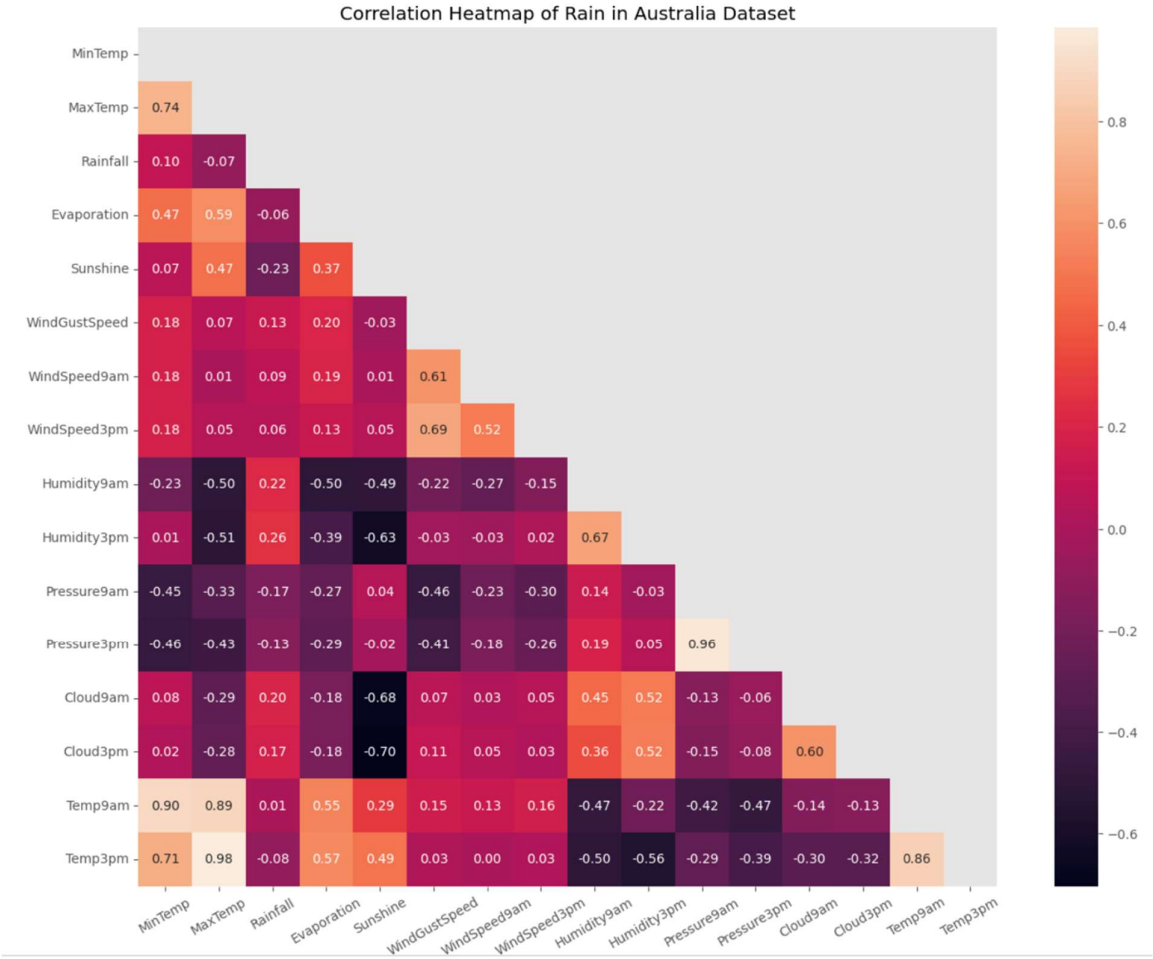


Fig. 10 Heatmap

	VIF	Features
0	32040.544068	Intercept
1	10.028495	MinTemp
2	43.550294	MaxTemp
3	1.159974	Rainfall
4	2.201594	Evaporation
5	3.243247	Sunshine
6	4.027628	Humidity9am
7	6.611909	Humidity3pm
8	19.692239	Pressure9am
9	19.812638	Pressure3pm
10	2.221865	Cloud9am
11	2.275500	Cloud3pm
12	52.071299	Temp3pm
13	22.818551	Temp9am

Fig. 11 VIF factor

If value of VIF greater than 5 then it indicates higher potential correlation between the variables. This is not reliable for categorical data if that has cardinality.

VIF is generally used for regression models but for learning collinearity we can use it.

Conclusion

In nutshell, after this EDA, we have enough evidence for feature engineering for our classification model. We cleaned a variable with mean, median and mode with use of distribution and logic. From the heatmap we weren't sure for which variable we should go for and there we can use correlation values as well as VIF to judge.

- We still haven't figure out that how wind can cause a rain with help of other, so we left it as another goal for future scope.
- Considering a milestone, we are more tends to develop a model for classification and then go for higher scale like with the use of Date and Location variable and also the use of wind variables.

Note: We are bound for 1400 words are less for explaining all of the stuff. Rest is in notebook file. Attaching **GitHub** repo as reference.

References

Z., & posts by Zach, V. A. (2020, July 20). *How to Calculate VIF in Python - Statology*. Statology. Retrieved October 9, 2022, from https://www.statology.org/how-to-calculate-vif-in-python/

M. (2022, October 9). *GitHub - Mxnxn/Rain_Forecasting_Australia*. GitHub. Retrieved October 9, 2022, from https://github.com/Mxnxn/Rain_Forecasting_Australia