# Rain in Australia Report

By Rolando Santos

## Introduction

This report was built to observe several case studies utilizing weather data. The ultimate goal of this project was to look at predictive modeling and data analytics on weather data. Understanding weather patterns can benefit the day to day plans of the average person as well as logistics for many companies that depend on it. This report is based on weather data in Australia using real world data. In this report the following case studies were observed: Wind gust speed over a period of time (one year), average temperatures in Australian cities, ratio of rainy days on a month/year basis, the relationship between humidity and temperature, and utilizing machine learning for weather forecasting (whether or not it will rain tomorrow).

## Dataset

The dataset was taken from kaggle using this link. The dataset contains about 10 years of daily weather observations from numerous weather stations in Australia. The data was provided by the Australian Bureau of Meteorology. The dataset has a total of 145460 rows and 23 columns, and features a plethora of missing values. Some of the variables used frequently in our case studies are:

| **Variable** | **Type** | **Description** |
|---|---|---|
| Date | Time-Series | Date of observation (Y-m-d) |
| Location | Categorical | Common name of the location of the weather station |
| MinTemp | Numeric | The minimum temperature in degrees Celsius |
| MaxTemp | Numeric | The maximum temperature in degrees Celsius |
| WindGustSpeed | Numeric | The speed (km/h) of the strongest wind gust in the 24-hour period |
| RainToday | Binary | 1 if precipitation (mm) in the last 24 hours to 9am exceed 9mm, else 0 |

## Methods

### *Wind Gust Speed Over Time*

WindGustSpeed is described as the speed (in km/h) of the strongest wind gust (sudden burst of wind speed and force) in a 24 hour period. We want to visualize this variable and see if there is a relationship between this variable and the time of the year. Four locations in Australia as the different lines in our line plot (Canberra, Sydney, Darwin and Melbourne). The year 2015 was chosen as the time period.

The first step was to filter the date column to be greater than the end of 2014 (2014-12-21) and less than the year 2016 (2016-01-01). We then filtered by location (Canberra, Sydney, Darwin and Melbourne). To confirm there was a row for each day of the year for these four locations, I divided the number of rows in the sub-dataset by 4 and got a total of 365, confirming that each location had an entry per day. I then checked to see if there were any NaN entries and found that there were 30 missing. Since this was a relatively small amount of missing rows there was no need to impute or remove these rows. I then created a plotly line plot with our x-axis being the day of the year, y-axis being the wind gust speed, and a color was assigned to each location line.

### *Average Temperature in Cities*

This dataset features two variables for temperature, MinTemp and MaxTemp. MinTemp is the minimum temperature of that day (row) in degrees Celsius, whereas MaxTemp is the maximum temperature of that day (row) in degrees Celsius. We want to identify the spread of temperature across different locations in Australia during the middle of the year. Note that during this time it is considered winter. We want to see the highest average temperature (in Fahrenheit) of the ten locations with the most entries in the entire dataset with the time period being July 2016.

First, to find the ten locations with the most entries, we group our dataset by location and aggregate by the number of entries associated with each location. This sub-dataset is then sorted by the number of entries in descending order (highest to lowest). Another copy of the original dataset is created, in this copy we modify the date variable to be a pandas datetime object. A new column is added called AvgTemp which is the mean of MaxTemp and MinTemp. A lambda function calculating Fahrenheit is mapped to this variable. Finally, the data is filtered by the locations found earlier and by month and year. This sub-dataset is then grouped by location, missing rows are removed here as well. An aggregation finding the max of AvgTemp is performed and we sort by MaxAvgTemp to show on our ten locations.

### *Ratio of Rainy Days*

In this question we wanted to see the ratio of rainy days for a given month after and including the year 2010. In the dataset there is a binary variable RainToday that denotes whether or not it rained on that day. We want to find the top 10 ratio of “Yes” responses for each month/year combination. The location used for this question is the capital of Australia, Canberra.

A new copy of the original dataset is created, in this copy "Date" is modified to be a pandas datetime object. Two new variables are added, Month and Year. The sub-dataset is filtered by year to show only the years 2010 and above. The sub-dataset is filtered by location as well to show only entries in Canberra. The data is grouped by Month and Year. A lambda function calculating the mean of "Yes" responses in RainToday is mapped to each grouped combination. We then sort by mean and apply a limit to view the top 10 results.

### ***Humidity and Temperature Relationship***

Here we want to visualize the relationship between average humidity and average temperature in our dataset using a scatter plot. To find average humidity, we take the mean of Humidity9am (the percent humidity at 9am), and Humidity3pm (the percent humidity at 3pm). These percentages are multiplied by a 100 in the dataset, instead of 0.99 the value is 99 as an example. A new variable dew point, used to describe the level of moisture in the air, will be used as the color of the scatter plot. This plot will show results for the year 2016 in Canberra, Melbourne and Sydney.

A new variable AvgHumidity is created from taking the mean of Humidity9am and Humidity3pm. The dataset is filtered to show all results in 2016 for Sydney, Melbourne and Canberra. A function calculate_dew_point is created to bin AvgHumidity into three categories: dry for humidities 55% or lower, sticky for humidities between 55% and 65%, and moist for humidities greater than 65%. This function is mapped to a new variable DewPoint. AvgTemp is calculated in a similar manner to the second question and converted to Fahrenheit. A scatter plot is then created with average temperature being the x-axis, average humidity being the y-axis, and color being represented by dew point.

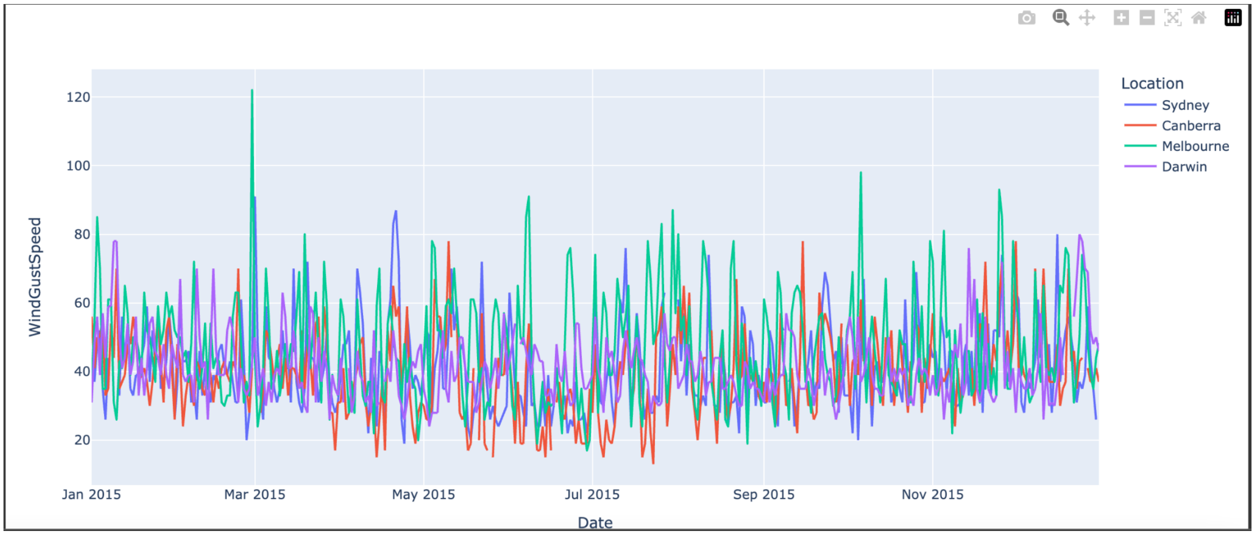
### ***RainTomorrow Prediction***

RainTomorrow is a binary categorical variable indicating whether or not it did rain the following day of that entry. A neural network was modeled using this dataset to predict this variable.

The first step was to clean the dataset. Three new variables were created to represent the date, Day, Month and Year. The date column was then removed as it was redundant. If RainToday and RainTomorrow had any missing rows, those rows would be removed for simplicity. The number of missing values for each other column of the dataset was calculated. If a column had more than 30% of its data missing we would remove that column from our dataset. All other columns with missing values below the 30% threshold would be imputed instead. Mean imputation was performed on numeric variables and mode imputation on categorical variables. Finally, we factorize our remaining categorical variables. The dataset was then split into three sets, training, testing and validation. The values in the X_train, X_validate and X_test were all normalized to have numeric values between 0 and 1. A keras model was created with an initial input dimension of 20 (representing the remaining number of variables post cleaning), with an output dimension of 16. A second layer with an output dimension of 32 was then built onto the model. Both these layers used relu as the activation function. A final layer with output dimension 2 representing our binary response was added with a sigmoid activation function. The model was then configured using binary_crossentropy as the loss function, adam as the optimizer and accuracy as the metric we wanted to measure. The model was fitted with 15 epochs with a batch size of 64 used against the validation set. The score of our model was then evaluated.

## Results

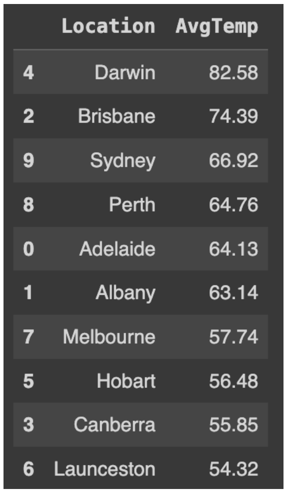
### ***Wind Gust Speed Over Time***



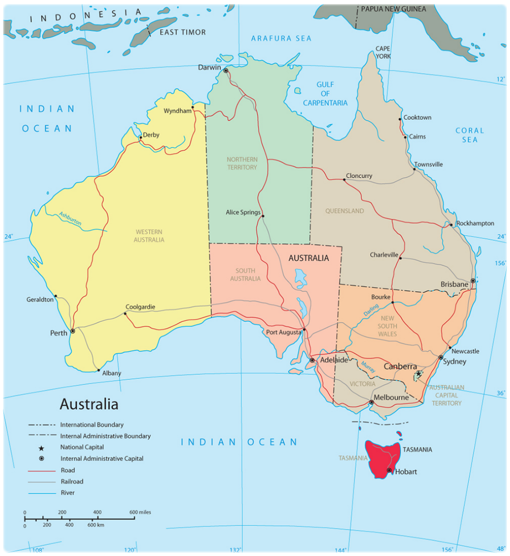
In our visualization we can see that there is no consistent relationship between gust wind speed throughout the year, as there are random spikes of data at different times of the year. We can see however, that Melbourne appears to have higher wind gust speed peaks on average compared to the other locations. Vice versa, we see that Canberra has lower troughs on average.

### *Average Temperature in Cities*

The locations that had the highest number of entries in the dataset were Canberra, Sydney, Adelaide, Melbourne, Perth, Hobart, Brisbane, Darwin, Albany and Launceston in that order, with Canberra having 3436 total entries and Launceston having 3040.

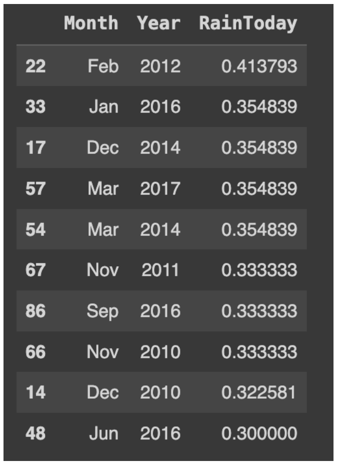


| | Location | AvgTemp |
|---|---|---|
| 4 | Darwin | 82.58 |
| 2 | Brisbane | 74.39 |
| 9 | Sydney | 66.92 |
| 8 | Perth | 64.76 |
| 0 | Adelaide | 64.13 |
| 1 | Albany | 63.14 |
| 7 | Melbourne | 57.74 |
| 5 | Hobart | 56.48 |
| 3 | Canberra | 55.85 |
| 6 | Launceston | 54.32 |



In the table shown, Darwin appears to have the highest average temperature during the time period of July 2016, whereas Launeston has the lowest. If we look at the map of Australia, we can see that Darwin is the northernmost city of Australia. Launceston is located in Tasmania, the island colored red south of Australia, and along with Hobart are the southernmost cities in Australia. Melbourne and Canberra are also located at the southern edge of the Australian landmass as well. We can conclude that during this time period, cities south of Darwin all experienced lower temperatures.

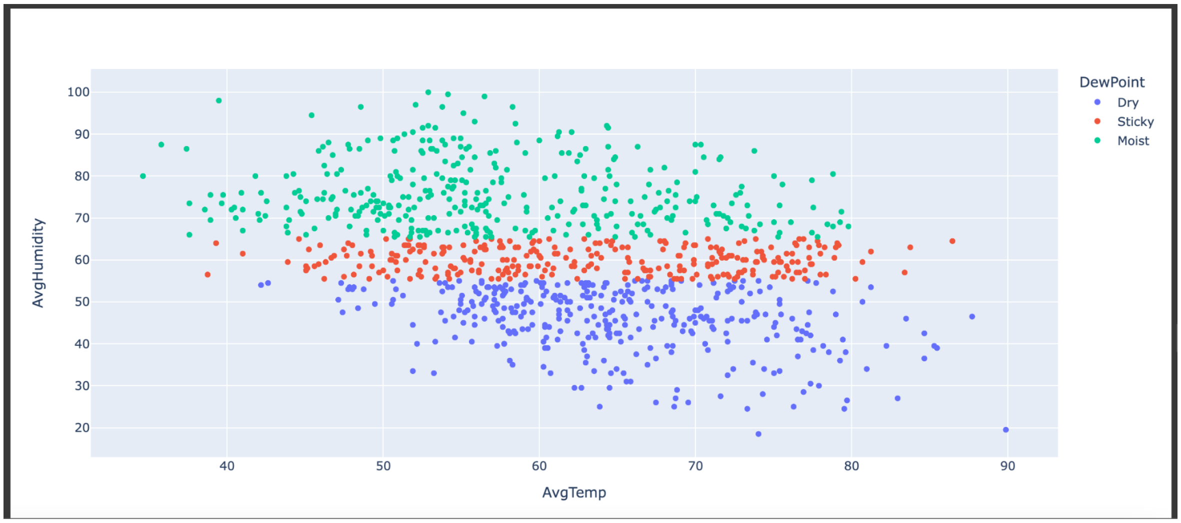
### *Ratio of Rainy Days*



| | Month | Year | RainToday |
|---|---|---|---|
| 22 | Feb | 2012 | 0.413793 |
| 33 | Jan | 2016 | 0.354839 |
| 17 | Dec | 2014 | 0.354839 |
| 57 | Mar | 2017 | 0.354839 |
| 54 | Mar | 2014 | 0.354839 |
| 67 | Nov | 2011 | 0.333333 |
| 86 | Sep | 2016 | 0.333333 |
| 66 | Nov | 2010 | 0.333333 |
| 14 | Dec | 2010 | 0.322581 |
| 48 | Jun | 2016 | 0.300000 |

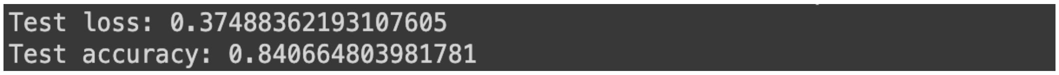
Our results show that in February 2012, rain was experienced 41.4% of the days in that period. On average however, most months experienced  a 33.3% ratio of rainy days. Something to note is that most of the months listed in the top 10 here are not during Australian winter (June, July and August).

***Humidity and Temperature Relationship***



When viewing the relationship between average humidity and average temperature, we can see a slight inverse linear relationship between the two variables. There appears to be a higher concentration of low temperatures at high humidity and a higher concentration of high temperatures at low humidity. However, it appears that most of the data is concentrated in the middle, so although we can make out a relationship between the two, it does not appear to be a strong one. Since dew point is a function of humidity we can see the correlation between the two in our data.

***RainTomorrow Prediction***



```
Test loss: 0.37488362193107605
Test accuracy: 0.840664803981781
```

The above was the results we ended up with after scoring our model against the testing dataset. An issue that was noticeable from the start was the number of missing values in the dataset. After some evaluation and research it was thought best to remove some variables that could potentially cause issues (thus the 30% missing value threshold was added). The following variables were removed as a result: Evaporation, Sunshine, Cloud3pm and Cloud9am. Sunshine had the highest amount of missing values with a total of 69835 out of 145460, close to half of all entries in the dataset. Mean and modal imputation were the simplest ways of fixing missing data, however if there were fewer variables that had missing values, a regression based imputation would have been considered. Logistic Regression, Decision Trees or SVMs are other machine learning algorithms that were considered for this type of binary classification, however neural networks seemed more appropriate since it was in the scope of this class.

## Appendix

All code used in this report can be found in the google colab linked below:
https://colab.research.google.com/drive/17q9N7Lzc941HC7ZiG5Thzw3GX4V57cNf?usp=sharing