WILL IT RAIN TOMORROW?

C7084

Github Link: https://github.com/BUCKERS99/C7084-Big-data--Will-it-rain-tomorrow-

APRIL 20, 2022

17239400
Dr Ed Harris

Contents

Background

Australia has a landmass of over 7.7 million square kilometres and contains all ecosystem biomes except arctic tundra; the most abundant Koppen-Geiger classification of land seen in Australia is Humid subtropical climate (Cfa) (climate-data, undated) , more information can be seen in Figure 1. According to Crosbie *et al*., (2012) it is predicted that the future climate of Australia will change with arid climate increasing from 76.5% to 81.7% along with temperate climates decreasing over 5%. This makes relying on rainfall a limiting factor for commercial agriculture within the country.

Table 1 - Koppen-Geiger Classification of Australia

Classification	Count	Köppen-Geiger	Examples
Humid subtropical climate	974	Cfa	Sydney, Brisbane, Newcastle, Wollongong, Ipswich
Oceanic climate	855	Cfb	Melbourne, Canberra, Hobart, Geelong, Launceston
Cold semi-arid climates	261	BSk	Mildura, Kerang, Hay, Kimba, Whyalla
Warm-summer Mediterranean climate	257	Csb	Albany, Warrnambool, Busselton, Victor Harbor, Port Fairy
Hot semi-arid climates	180	BSh	Alice Springs, Mount Isa, Broome, Charters Towers, Carnarvon
Hot-summer Mediterranean climate	153	Csa	Perth, Adelaide, Mandurah, Bunbury, Geraldton
Hot desert climates	117	BWh	Port Hedland, Roxby Downs, Exmouth, Port Augusta, Coober Pedy
Tropical savanna climate	75	Aw	Townsville, Darwin, Jabiru, Karumba, Nhulunbuy
Tropical monsoon climate	22	Am	Cairns, Ingham, Lucinda, Cardwell, Gordonvale
Tropical rainforest climate	16	Af	Babinda, South Mission Beach, Wongaling Beach, West Island, Mission Beach

Warm humid continental climate	4	Dfb	Mt Buller Village, Hotham Heights, Dinner Plain, Falls Creek
Cold desert climates	3	BWk	Rawlinna, Forrest, Cook
Subarctic climate	1	Dfc	Perisher Valley
Tundra climate	1	ET	ANARE Station, Macquarie Island
Subpolar oceanic climate	1	Cfc	Miena

Average rainfall has been seen to fluctuate for the country, but within certain remote and rural areas it is far below average. The rainfall within Australia relies heavily on El Nino-Southern Oscillation (ENSO), which can be in one of three phases: Neutral, El Nino, and La Nina. These phases are defined by the temperature of the sea surface in the central and eastern areas of the Pacific Ocean (Climate, 2014). With the fluctuations and unpredictability of weather events during these phases the prediction of rain within Australia is a complex with long range weather forecasting often being incorrect.

For agriculture in Australia, it is crucial that rain predictions should be accurate for a variety of reasons that here, in the UK, we take for granted. Rain can cut off communities from the most basic of human needs such as food and supply links; for animals such as cattle it can cause them to become trapped with no access to higher ground. It was therefore decided that the question Will it Rain Tomorrow? Will be answered using data found on Kaggle(2021) collected over the past 10 years by the Australian Bureau of Meteorology (2010).

The following objectives were set:

1. Achieve an accuracy of 70% or greater in predicting if it will rain the day after the data was recorded.
2. Out of the models tested: determine which type of model returns the highest accuracy with the lowest loss and time to completion
3. Display the results on an interactive map of Australia.

Methods

The data set was found on Kaggle; variables and description can be found below in Table 1.

Table 2 - Table of variables

Variable	Class	Definition
Date	Date	Date of the observations
Location	Factor	Location of the weather station
MinTemp	Double	Minimum temperature that day (°C)
MaxTemp	Double	Maximum temperature that day (°C)
Rainfall	Double	Rainfall that day in (mm)
Evaporation	Double	Evaporation in 24hrs (mm)

Sunshine	Double	Hours of sunshine
WindGustDir	Factor	Direction of wind gusts
WindGustSpeed	Double	Wind gust speed (Km/h)
WindDir9am	Factor	Direction of wind at 0900hrs
WindDir3pm	Factor	Direction of wind at 1500hrs
WindSpeed9am	Double	Wind speed at 0900hrs (Km/h)
WindSpeed3pm	Double	Wind speed at 1500hrs (Km/h)
Humidity9am	Double	Humiditiy at 0900hrs (%)
Humidity3pm	Double	Humidity at 1500hrs (%)
Pressure9am	Double	Pressure at 0900hrs (Hpa)
Pressure3pm	Double	Pressure at 1500hrs (Hpa)
Cloud9am	Double	Fraction of sky obscured at 0900hrs (Oktas)
Cloud3pm	Double	Fraction of sky obscured at 1500hrs (Oktas)
Temp9am	Double	Temperature at 0900hrs (°C)
Temp3pm	Double	Temperature at 1500hrs (°C)
RainToday	Factor	Factor of yes and no for if it rained that day
RainTomorrow	Factor	Factor of yes and no for if it rained the next day

The decision was taken to use python for the data analysis of this assignment, primarily the Amazon sage maker studio lab platform (Amazon, 2022). This was not possible however due to platform not accepting log in details and being unable to reset the credentials. Workarounds were used to complete the data analysis in Google Collaboratory and Jupiter lab through the Saturn cloud server.

Two models were completed using the Saturn cloud environment where the Jupiter server is connected to a tesla T4 GPU (Graphics Processing Unit). The T4 GPU contains 320 Turing tensor cores which accelerate the time taken to preform machine and deep learning training and modelling (NVIDIA, 2019). In a technical report conducted by Jia *et al*. (2019), it was found to outperform its predecessors in speed of inference. To compare the time taken for modelling the CPU option was used within google Collaboratory and, due to Jupiter labs not supporting the use of anaconda packaging, the GPU function for the use of NVIDIA's CUDA (Compute Unified Device Architecture) platform.

Experimental data analysis

The data frame was loaded into the environment using Pandas directly from Kaggle and then a series of validation checks were made to determine if the data frame created had been successfully read in. The data frame was then checked to see the number of null values (NA). It was found that there was a number of missing values in the dependant variable of Rain Tomorrow. In order for a classification model to be created these values had to be removed. Variables were then split into model features and the model target. Numerical and categorical variables were also split for exploratory graphs to be created.

Figure 1 shows that the data set used is unbalanced which had to be taken into account when running certain models. Plots were created for the rest of the variables individually to visually assess normality. The categorical variables were not checked for normality as it is known that it will not have a normal distribution due to there not being a fixed or known "score", or any order in the categories (Research Gate, 2021). The numerical variables were checked for normality using the Shapiro-Wilk test (Shapiro and Wilk, 1965) after the graphical inspection.

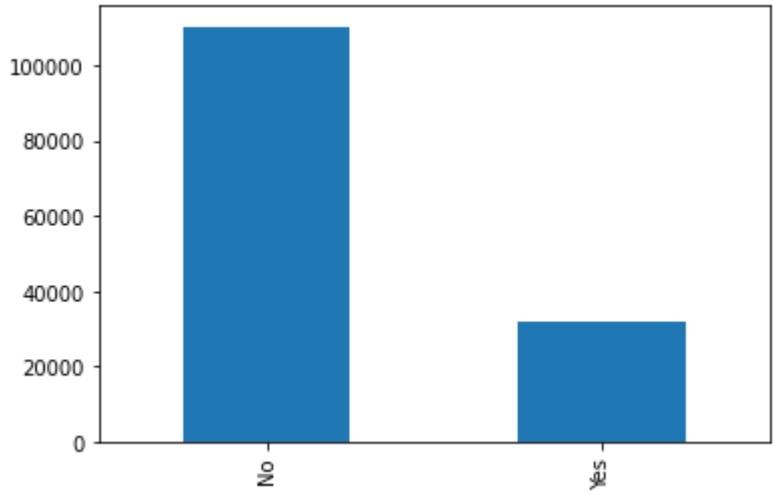

Figure 1 - Unbalanced data set

A correlation heat map was used to check for collinearity and variables were removed from modelling if they had a defined value of more than 0.7, this was defined using the 2012 study conducted by Dormann *et al*.

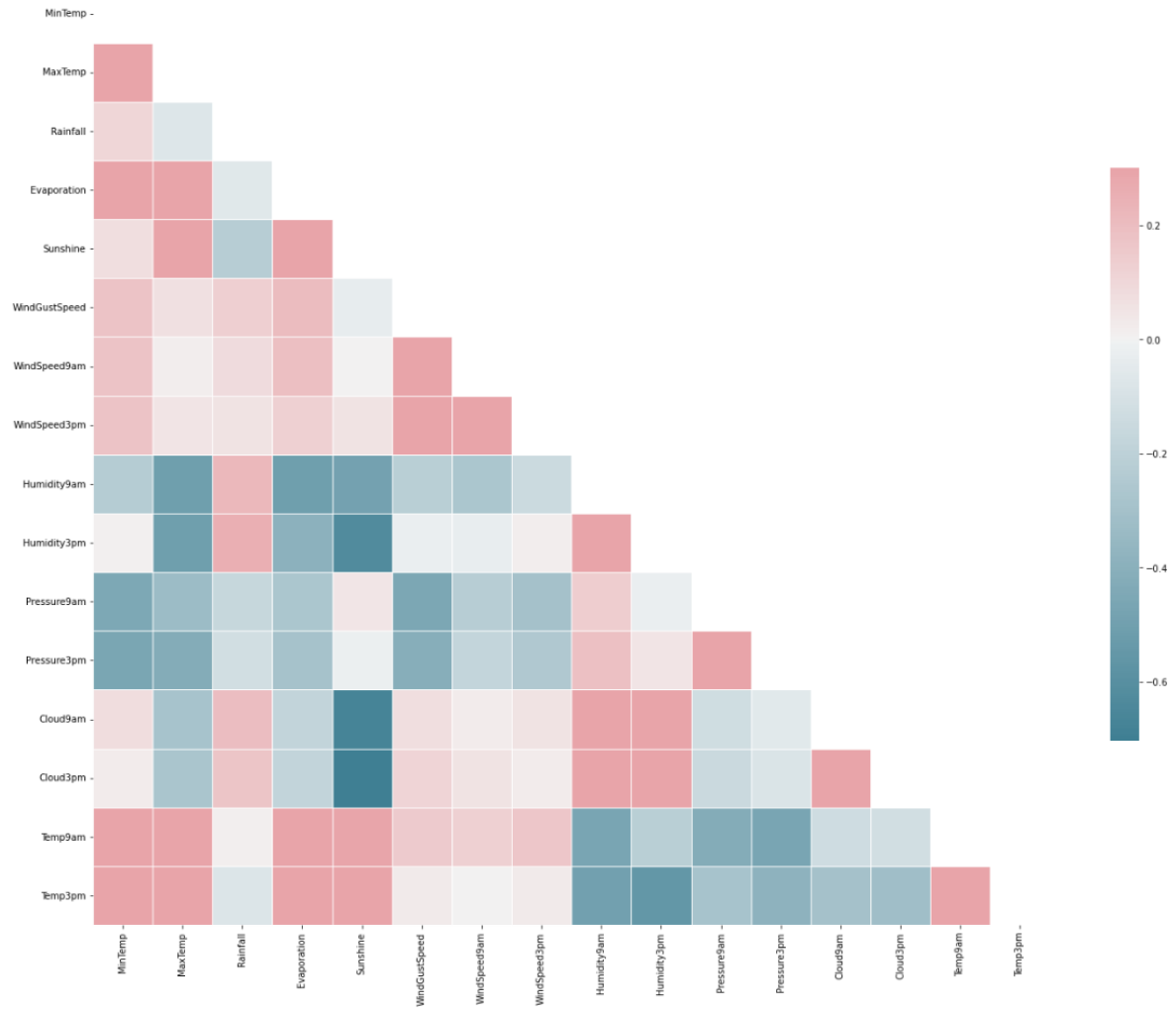

Figure 2 - Correlation plot

For all the models run the data was subset into three data frames following an 80:10:10 split: train (80), test (10) and validation (10).

k-Nearest-Neighbour (kNN)

For the first model the use of a the kNN algorithm was used as a benchmark alongside the CPU runtime in google Collaboratory. A pipeline was used in order to apply data transformations. This creates values for the NA values found within the data set and also scales the numerical values into similar orders of magnitude. This is done to automate the workflow for the machine learning algorithm in one step instead of having to code different steps multiple times for variables (Valizadeh *et al*., 2009). Using a kNN algorithm for classification problems can be viewed as a lazy learning method (Guo, *et al.,* 2003), however it is effective for predicting a binomial outcome. There is scope to tune the k parameter for the algorithm using the validation data set created if the accuracy is lower than the objective set at 70%.

The code chunk to record the time taken was also included to get a measurement of the processing speed.

CUDA framework

Logistic Regression

To run the NVIDIA package it was necessary for a specific GPU runtime to be selected in google Collaboratory. Due to Colaboratory automatically assigning runtimes to sessions it proved difficult to ensure that the required Tesla T4 GPU was selected. This required the runtime to be restarted until this was assigned.

Once assigned the NVIDIA rapids for Collaboratory needed to be installed into the session. This along with conda for colab was imported and installed as this is necessary for compatibility reasons. CUDA, along with its dependencies were then installed.

Similar steps were used to import the data from the github link instead using the cudf package to create the data frame. The data frame was then checked and categorical data was assigned to the correct data type. For this model all NA values were removed, this was due to previously running the model which included the NA values and it causing errors.

All categorical variables needed encoding due to the machine learning algorithm requiring all variables to be numerical. OneHot encoding was chosen for this manipulation as it was compatible with the NVIDIA framework, it was also noted in a 2018 study (Seger) that OneHot encoding produced the best performance compared to other methods.

A logistic regression from the Compute Unified Machine Learning (CUML) package was used in its basic form to complete this task to assess accuracy and time to completion.

KNN

This model was run using the CUDA framework in order to compare time to completion and accuracy against the standard run time model.

The same steps were performed as with the logistic regression model but using a different algorithm.

Results

Table 3 - Table of results

Model	Time to completion (seconds)	Accuracy (%)
CUML Logistic Regression Test	5	78
CUML-k Nearest Neighbor Train	0.2	86
CUML-k Nearest Neighbor Test	0.1	84
CUML-k Nearest Neighbor k-score	6.1	85
k Nearest Neighbor Train	196	90
k Nearest Neighbor Test	148	82
k Nearest Neighbor k-score	6143	85

KNN

The runtime kNN model completed in a time of 195.8 seconds producing an accuracy of 90% on the training data set. On the test data set it was seen to produce an accuracy of 83% and a time to completion of 23.6 seconds. All accuracies and runtimes have been documented in Table 3 and were correct when running them on the authors computer.

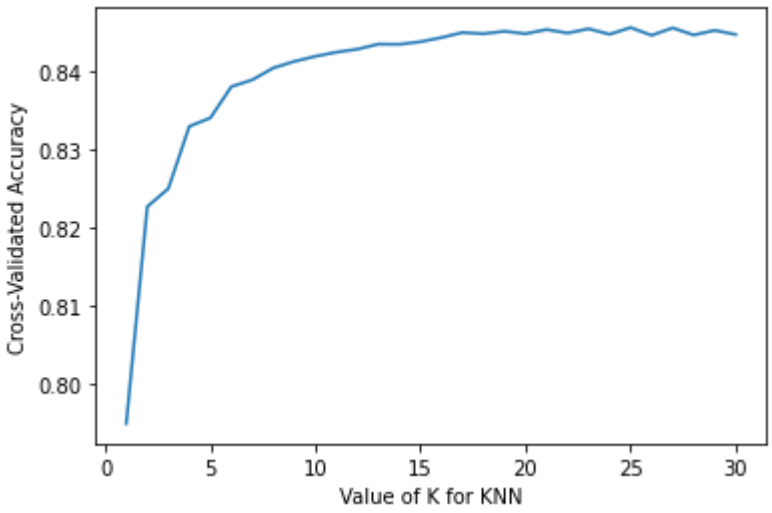

Figure 3 - Best value for k

Further analysis into the best value for k was completed even though the objective of 70% accuracy was reached. The best value for k was seen to be 25 as shown in Figure 4 this was completed in a time of 6143 seconds. This was seen as a good value to benchmark against when running the same process through the CUDA framework.

CUDA logistic regression

Figure 5 shows that using the GPU accelerated session recorded a time to completion of 1.8 seconds with a test accuracy of 78%. While this was a lower accuracy than seen with the kNN model it was deemed unnecessary to further tune the model to produce a higher accuracy; this allowed for the focussing on a direct comparison with a GPU accelerated kNN model.

CUDA KNN

When running the NVIDIA CUDA kNN model it produced an accuracy of 84% accuracy, correctly predicting 9507/11284 outcomes. The more important finding when running this test was the time to completion. It was seen that the model predicted with 86% accuracy in 0.28 seconds, while the time taken to complete the best value for k validation took 6.1 seconds to complete. Figure 6 below shows the outcome for the best value for k; it was found that 30 was the best value for k.

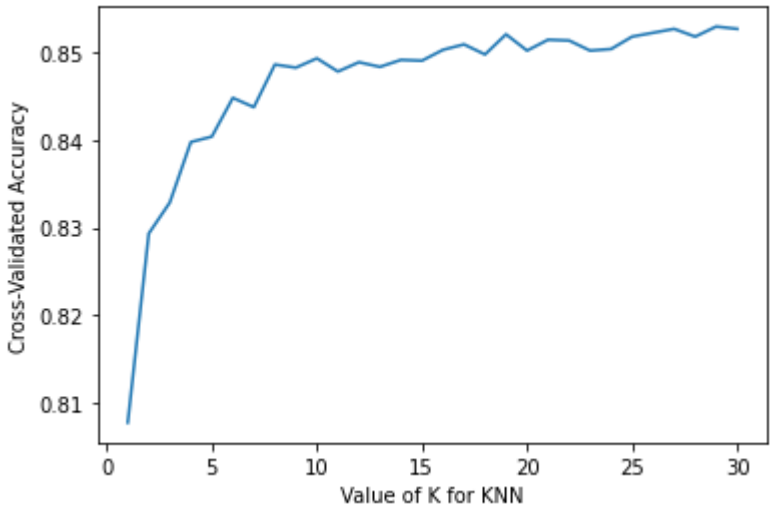

Figure 4 -Best value for k (CUDA)

Discussion

It has been shown in a number of studies that the use of a GPU will decrease the time taken to train models (Jhurani and Mullowney, 2015; Nishino *et al*., 2017). When completing a direct comparison, the time to completion for the validation steps of find the best value for k was seen to benefit most from the use of a GPU with the difference in time taken being 102 minutes. A good illustration to the industries view of using GPU's for classification and big data problems is shown in Figure 7.

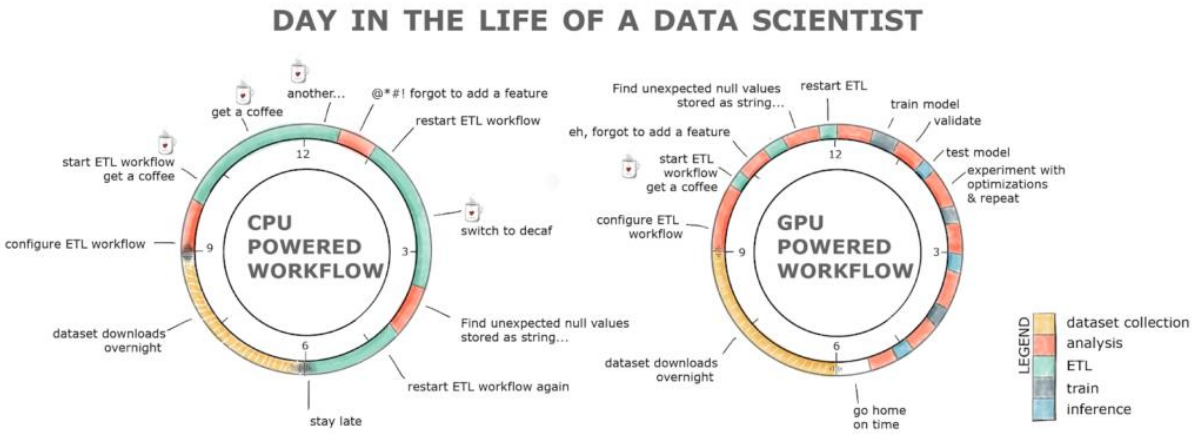

Figure 5 - Day in the life of a data scientist

While this figure is a light-hearted view, the increase in speed has serious consequences in the world of analysis and creating meaning from large data sets.

The need for encoding categoric variables is imperative to the successful running of machine learning models. This is because statistical understanding inherently uses mathematics, which needs numbers to work. Encoding can be as simple as making a "Yes" response into a 1 and a "No" into a 2 or dates into a string of numbers. There are many types of encoding such as target encoding, which converts a categorical value into the mean of the target variable, however care must be taken to choose the correct encoding for the model, as it can cause overfitting. For the machine learning problems encountered throughout this study the OneHot encoding method was used. OneHot encoding is used for nominal data, where the

data does not include any numerical values, and creates binary features for each categoric variable.

With the analysis reaching an accuracy of over 80% it shows that there is scope to implement a method of early warning to people in regional and rural areas that could be more accurate than a standard weather prediction. This could prevent isolation events that has large detrimental affect on human populations and, for farmers, create the ability to move stock before a large rain event. This could reduce the number of stranded cattle and deaths attributed to severe rain and flooding.

The creation of the animated map to document if it will rain tomorrow was difficult to produce. It was decided that it should be made in R Studio as the author had more confidence in completing the task. Most map making packages in R do not have a default outline for Australia, this problem was overcome using the ozmaps package (mdsumner, 2021) which supplied the correct geometry data needed to create an outline and state boundaries. The date function had to be correctly used and defined to recognise the correct date. If this was changed without defining the place holders it caused errors. The choice to use a slower transition speed throughout the animation was in order to allow the viewer time to view the output. A faster animation time moved through the frames too quickly and no meaning was able to be gathered from the plot. If this was to be created again a shiny app (Shiny, 2022) would allow the user to select dates of interest and possible compare data from other dates.

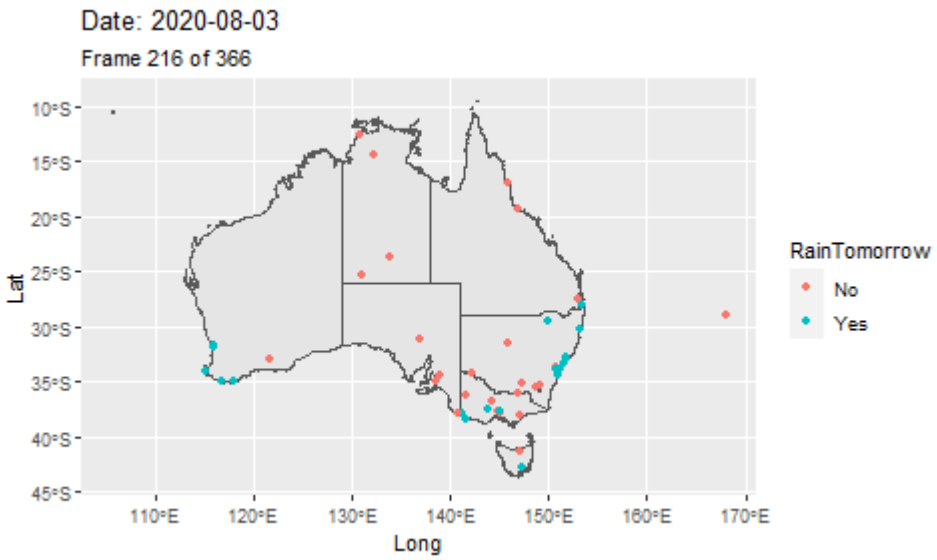

Figure 6 - Animated map of results

Figure 6 shows a still image of the map. The animated map can be viewed in the authors github, or by following this link. The full report and all files associated are also available through the link ok the title page.

References

Amazon. 2022. *Learn and experiment with machine learning.* [Online]. Available from: https://studiolab.sagemaker.aws/ [Accessed on 15/04/2022].

Climate. 2014. *What is the ElNino-Southern Oscillation (ENSO) in a nutshell?* [Online]. Available from: https://www.climate.gov/news-features/blogs/enso/what-el-ni%C3%B1o%E2%80%93southern-oscillation-enso-nutshell [Accessed on 15/04/2022].

Climate-data. Undated. *Climate-Australia*. [Online]. Available from: https://en.climate-data.org/oceania/australia-140/ [Accessed on 15/04/2022].

Crosbie, R.S., Pollock, D.W., Mpelasoka, F.S., Barron, O.V., Charles, S.P. and Donn, M.J., 2012. Changes in Köppen-Geiger climate types under a future climate for Australia: hydrological implications. *Hydrology and Earth System Sciences*, *16*(9), pp.3341-3349.

Dormann, C.F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., Marquéz, J.R.G., Gruber, B., Lafourcade, B., Leitão, P.J. and Münkemüller, T., 2013. Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, *36*(1), pp.27-46.

Jhurani, C. and Mullowney, P., 2015. A GEMM interface and implementation on NVIDIA GPUs for multiple small matrices. *Journal of Parallel and Distributed Computing*, *75*, pp.133-140.

Jia, Z., Maggioni, M., Smith, J. and Scarpazza, D.P., 2019. Dissecting the NVidia Turing T4 GPU via microbenchmarking. *arXiv preprint arXiv:1903.07486*.

Mdsumner. 2021. *Ozmaps.* [Online]. Available from: https://github.com/mdsumner/ozmaps [Accessed on 20/04/2022].

Nishino, R.O.Y.U.D. and Loomis, S.H.C., 2017. Cupy: A numpy-compatible library for nvidia gpu calculations. *31st confernce on neural information processing systems*, *151*.

NVIDIA. 2019 *NVIDIA T4 Tensor Core GPU.* [Online]. Available from: *https*://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-t4/t4-tensor-core-datasheet-951643.pdf [Accessed on 15/04/2022].

Research Gate. 2021. *Normality test for categoric variables.* [Online]. Available from: https://www.researchgate.net/post/Normality_test_for_categorical_variables [Accessed on 15/04/2022].

Seger, C., 2018. An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing.

Shapiro, S.S. and Wilk, M.B., 1965. An analysis of variance test for normality (complete samples). *Biometrika*, *52*(3/4), pp.591-611.

Shiny. 2022. *Shiny from R Studio.* [Online]. Available from: https://shiny.rstudio.com/ [Accessed 20/04/2022].

Valizadeh, S., Moshiri, B. and Salahshoor, K., 2009. Leak detection in transportation pipelines using feature extraction and KNN classification. In *Pipelines 2009: Infrastructure's Hidden Assets* (pp. 580-589).