Haoyang Tan

CS 472

Professor: Thien Huu Nguyen

5/11/2023

Dear Professor Thien,

I am writing to propose my final project. This proposal outlines the methodology and rough process that I will use in my final project to make the project of predicting whether it will rain the next day in various regions of Australia more accurate.

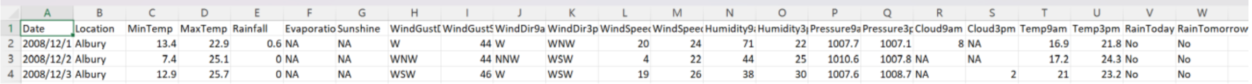
Executive Summary:

The context is to predict **next-day rain** by training classification models on the target variable **Rain Tomorrow**. I will use **KNN** and **logistic regression** to solve this problem.

Introduction:

I obtained this data by searching for classification tasks on the Kaggle ML website. This dataset contains about 10 years of daily weather observations from many locations across Australia. **Rain Tomorrow** is the target variable to predict. It means -- did it rain the next day, Yes or No? This column is Yes if the rain for that day was 1mm or more.

Data Process:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporatio	Sunshine	WindGustD	WindGustS	WindDir9a	WindDir3p	WindSpeed	WindSpeed	Humidity9a	Humidity3p	Pressure9a	Pressure3p	Cloud9am	Cloud3pm	Temp9am	Temp3pm	RainToday	RainTomorrow
2	2008/12/1	Albury	13.4	22.9	0.6	NA	NA	W	44	W	WNW	20	24	71	22	1007.7	1007.1	8	NA	16.9	21.8	No	No
3	2008/12/2	Albury	7.4	25.1	0	NA	NA	WNW	44	NNW	WSW	4	22	44	25	1010.6	1007.8	NA	NA	17.2	24.3	No	No
4	2008/12/3	Albury	12.9	25.7	0	NA	NA	WSW	46	W	WSW	19	26	38	30	1007.6	1008.7	NA	2	21	23.2	No	No

This is an overview image of my data and what you can see is that this dataset has a total of **22 feature columns** and one result column, as well as **145,461 sample rows**. Some of the features are represented by strings, which I need to convert to numbers. For example, the wind direction will be represented by three letters and I will convert it into 16 numbers from 1-16 as a way to allow the program to better identify the wind direction. For sample features that appear as NA, I will set them to 0 so that they do not affect other features during the calculation.

In order for the test set to be close to the model, I will separate the data set into a **70% training set** and a **30% test set**.

Machine Learning Methods:

I will use two ways to try to solve this problem. As this is a classification problem and the sample size for this project is relatively small, so it is not appropriate to use overly complex methods. So I have chosen KNN and logistic regression.

The KNN algorithm is relatively simple to implement and theoretically, its disadvantage is that it can take a lot of time to test. This is because the distance is calculated for each test against all the previous training samples. My solution is to use a small volume test set first, and then use a larger test set when the program is debugged until it is close to being problem-free.

In logistic regression, I will use the sigmoid function to convert the predicted values into probability values. Then a gradient descent method will be used to find better weighting parameters. After several iterations to achieve the classification effect. If the training speed is too slow, I will use mini batching to speed up the iterations.

After using both methods to achieve the classification task, I will evaluate my models in terms of training time, testing time, testing accuracy, degree of overfitting, etc., and compare the two models.

Sincerely,

Haoyang Tan