A short-term precipitation forecast based on Australian meteorological data

Ye Su [1], Peiyao Wei [2],Haonan Chen [3],Nuozhou Wang [4],Kainan Liu [5]

[1] School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan, P.R.China; 2395295097@qq.com(Y.S.)
[2] School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan, P.R.China; 1292587164@qq.com(P.W.)
[3] School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan, P.R.China; 739465907@qq.com(H.C)
[4] School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan, P.R.China; 505318225@qq.com(N.W.)
[5] School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan, P.R.China; 563799345@qq.com(K.L.)

Abstract: With the accelerating pace of people's life, the impact of weather on human beings is also growing, so weather prediction has become a necessary and valuable work. In this paper, we use the daily weather data of 49 cities of Australia within ten years to carry out precipitation forecast experiments and adopt the SMOTE method to deal with the label imbalance. Finally, we select the most appropriate prediction model based on ensemble learning and build a web application for readers to learn more about our project visually. Our study can be used as a reference for future weather prediction, which will facilitate people's production and daily life in Australia.

Keywords: precipitation forecast; imbalance problem; ensemble learning

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *J. Mar. Sci. Eng.* **2022**, *10*, x. https://doi.org/10.3390/xxxxx

Academic Editor: Firstname Lastname

Received: date
Accepted: date
Published: date

1. Introduction

With the accelerating pace of people's life, the impact of weather on people's life is also growing, so weather prediction has become a necessary work. However, the diversity and complexity of climatic and meteorological conditions have brought great difficulties to accurate weather forecasting. Therefore, establishing an accurate short-term rainfall prediction model has always been a hot issue in meteorological science.

Our research is a short-term precipitation forecast based on Australian meteorological data. The task of this study is to predict whether there will be rainfall in the next day according to the temperature, sunshine, air pressure and other climatic factors of one day in a given Australian city, so as to forecast floods and other natural disasters, which can provide valuable information for Australian agricultural production. Besides, predicting the weather in daily life will also help us reasonably arrange travel. In view of the above-mentioned facts, we decide to select the most appropriate prediction model through experiments, build a web application for users to query the weather of a given Australian city in the past and learn more about the project visually.

At present, the main methods for rainfall prediction are physical mapping method, traditional statistical method, machine learning method and deep learning method. For the physical mapping method, the global navigation satellite system (GNSS) can effectively retrieve the perceptible water vapor (PWV), Dan.L et al. focused on the positive correlation between the time-varying characteristics of Atmospheric Perceptible Water and rainfall, and found that the PWV content increases sharply within a few hours of rainfall. Therefore, a short-term and imminent rainfall prediction method based on GNSS PWV is proposed and established[1].

Traditional statistic method can also perform well in rainfall prediction. Shan.Y used high-order Markov Chain in rainfall forecasting[2]. In that paper, the problem of multi period dependence of rainfall prediction is solved, which is more in line with objective facts. According to the characteristics of large amount of uncertainty in the precipitation process, the classification standard of precipitation sequence is established through cluster analysis. The standardized autocorrelation coefficients of each order are used as weights, and the moving average Markov chain model is used to predict the precipitation state in the future period through the state transition probability matrix, the prediction accuracy is high, which provides a possible way for rainfall prediction[3]. With the development of data science, machine learning methods are used in many aspect of life. Hu.Y and his workmates adopt improved Adaboost-C4.5 method in rainfall prediction and got a positive effect.[4]

Ensemble learning and deep learning usually perform well in prediction problems, Guan.P and his team work out the rainfall prediction problem with ensemble and deep learning, They compared and analyzed the prediction effects of ensembled models such as random forest, GBDT and XGBoost, and proposed an improved Adaboost regression model based on GBDT feature selection and a stacking model based on the combination of multiple strong repressors.[5] Zhao.Q, Liu.Y, Yao.Y Establishment of short-term and impending rainfall prediction model using least square support vector mechanism.[6]

Neural network has strong decision-making ability in dealing with nonlinear problems. Huang.J, Qin.L Study on correlation between temperature and rainfall using ALSTM model.[7] The algorithm uses the previous data in the weather time series to predict the rainfall and temperature in the next hour, so as to realize the multi value prediction of weather elements. Zhang.J et.al. established a multilayer BP neural network prediction model by analyzing the causes of precipitation and 18 meteorological factors such as ground pressure field, temperature field, humidity field and wind field as input factors of the model, and determined the final network structure with Tansig losing as transfer function and traindx as training function.[8] Liao.M proposed a spatiotemporal rainfall prediction model based on the limit gradient lifting decision tree (XGBoost), graph sampling aggregation model (GraphSAGE) and long short memory neural network (LSTM), which effectively improved the rainfall prediction effect.[9]

2. Materials and Methods

2.1 Data description

The data set is from **weather-dataset-rattle-package** on the Kaggle website and contains 145651 observations. The dataset records the weather changes in 49 cities of Australia from January 1, 2009 to June 25, 2017. There are 23 features totally in the data, including 6 category features and 17 numerical features.

Table 1 Features of the dataset

Features	Meaning	Features	Meaning
Date	Date of Observation	WindSpeed3pm	Wind speed (km/hr) averaged over 10 minutes prior to 3pm
Location	Location of the weather station	Humidity9am	Humidity (percent) at 9am
MinTemp	The minimum temperature	Humidity3pm	Humidity (percent) at 3pm
MaxTemp	The maximum temperature	Pressure9am	Atmospheric pressure (hpa) reduced to mean sea level at 9am
Rainfall	The amount of rainfall recorded for the day in mm	Pressure3pm	Atmospheric pressure (hpa) reduced to mean sea level at 3pm
Evaporation	The so-called Class A pan evaporation (mm) in the 24 hours to 9am	Cloud9am	Fraction of sky obscured by cloud at 9am.*
Sunshine	The number of hours of bright sunshine in the day.	Cloud3pm	Fraction of sky obscured by cloud at 3pm.*
WindGustDir	The direction of the strongest wind gust in the 24 hours to midnight	Temp9am	Temperature at 9am
WindGustSpeed	The speed (km/h) of the strongest wind gust in the 24 hours to midnight	Temp3pm	Temperature at 3pm
WindDir9am	Direction of the wind at 9am	RainToday	1 if precipitation (mm) in the 24 hours to 9am exceeds 1mm, otherwise 0
WindDir3pm	Direction of the wind at 3pm	RainTomorrow	The amount of next day rain in mm. Used as response variable RainTomorrow. A kind of measure of the "risk".
WindSpeed9am	Wind speed (km/h) averaged over 10 minutes prior to 9am		

*: This is measured in "oktas", which are a unit of eigths. It records how many eigths of the sky are obscured by cloud. A 0 measure indicates completely clear sky whilst an 8 indicates that it is completely overcast.

The distributions of numerical features are shown in **Figure 1**:

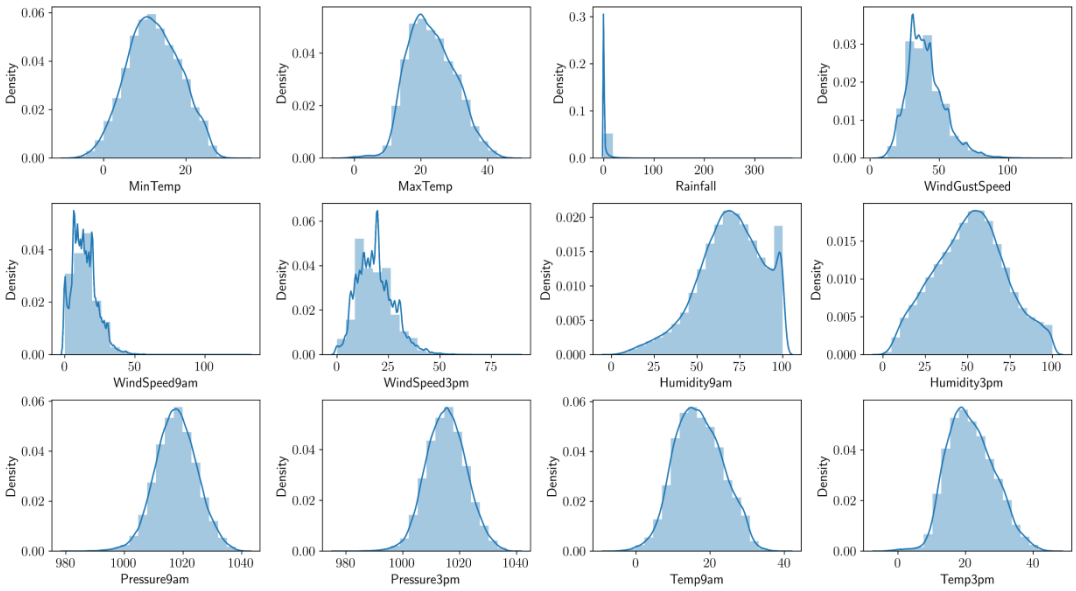

Figure 1 Distributions of numerical features

2.2 Methods

2.2.1 Logistic regression

Log probability regression is a method to solve binary classification tasks by using generalized linear regression, aiming at finding the parameter value β of the minimum loss function L.

So as to achieve the purpose of classifying dependent variables, the main idea of log probability regression is to find a suitable monotone differentiable function to link the real markers $y \in (0,1)$ of the classification task with the predicted values generated by the linear regression model$z = w^T x + b$. The most ideal connection function is the unit step function.

$$y = \begin{cases} 0, & z < 0 \\ 0.5, & z = 0 \\ 1, & z > 0 \end{cases}$$

When the predicted value is positive, it is judged as a positive example, and when the predicted value is negative, it is judged as a negative example. The limitation of unit step function lies in its discontinuity. Therefore, sigmoid function and cumulative density function of standard normal distribution are selected to replace unit step function to obtain logistic regression model.

$$Logistic\ Function: F(x,\beta) = \emptyset(x_i\beta) = \int_{-\infty}^{(x_i\beta)} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$$

Among that,

$$\begin{cases} P(y=1|x) = F(x,\beta) \\ P(y=0|x) = 1 - F(x,\beta) \end{cases}$$

2.2.2 ANN

An artificial neuron network (neural network) is a computational model that mimics the way nerve cells work in the human brain. It has three or more layers that are interconnected. The first layer consists of input neurons. Those neurons send data on to the deeper layers, which in turn will send the final output data to the last output layer.

All the inner layers are hidden and are formed by units which adaptively change the information received from layer to layer through a series of transformations. Each layer acts both as an input and output layer that allows the ANN to understand more complex objects. Collectively, these inner layers are called the neural layer.

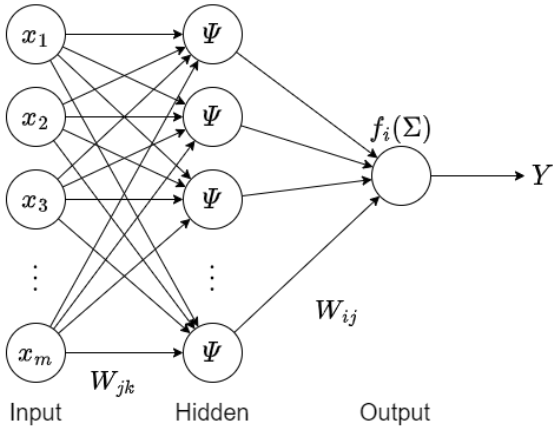

Figure 2 ANN structure

The units in the neural layer try to learn about the information gathered by weighing it according to the ANN's internal system. These guidelines allow units to generate a transformed result, which is then provided as an output to the next layer.

An additional set of learning rules makes use of backpropagation, a process through which the ANN can adjust its output results by taking errors into account. Through backpropagation, each time the output is labeled as an error during the supervised training phase, the information is sent backward. Each weight is updated proportionally to how much they were responsible for the error.

$$Loss\ Function: L = \frac{1}{2}\sum_{i=1}^{m_K}\left(Y_i^{(K)} - T_i\right)^2 = \frac{1}{2}\sum_{i=1}^{m_K}(\delta_i)^2$$

Where $\delta_i = Y_i^{(K)} - T_i$ represents the difference between the i element of the output vector and the i element of the label vector.

The purpose of Ann's network training is to obtain a set of network weights W to minimize the value of loss function (here W is the set of all weights and offsets). Therefore, the most traditional method is to let the derivative of loss function to W be 0, and

$$\frac{\partial L}{\partial W} = 0$$

For complex networks, it is difficult to obtain the solution or solution set of the above formula in an analytical way. Therefore, the numerical solution method favored by computers and the most important numerical solution method in the field of machine learning, stochastic gradient descent (SGD), are adopted. The expression is as follows:

$$W \leftarrow W - \eta\frac{\partial L}{\partial W}$$

Where η is the learning rate of the ANN and the step size for searching the weight value, $\frac{\partial L}{\partial W}$ is the gradient, "-" means that the weight is updated in the direction of decreasing the loss function. However, the above formula is only an abstract expression, and we cannot directly use it to iteratively solve the weights of each layer of ANN. Therefore, mathematicians skillfully developed the error back propagation method, and calculated the gradient layer by layer from the output layer and updated the weights layer by layer.

Therefore, our main goal is to calculate the gradient $\frac{\partial L}{\partial W^k}$ and $\frac{\partial L}{\partial b^k}$ of each layer.

An ANN has several advantages but one of the most recognized of these is the fact that it can actually learn from observing data sets. In this way, ANN is used as a random function approximation tool. These types of tools help estimate the most cost-effective and ideal methods for arriving at solutions while defining computing functions or distributions.

ANN takes data samples rather than entire data sets to arrive at solutions, which saves both time and money. ANNs are considered fairly simple mathematical models to enhance existing data analysis technologies.

2.2.3 Ensemble learning

Ensemble learning is not a single machine learning algorithm, but by building and combining multiple base learners to complete the learning task. Ensemble learning is often regarded as a meta algorithm.

Ensemble learning can be used for classification problem integration, regression problem integration, feature selection integration, outlier detection integration and so on.

For the training set data, we can train several individual weak learners, and through certain combination strategies, we can finally form a strong learner to achieve the purpose of absorbing the strengths of others.

Ensemble learning includes parallel method and sequential method. In the parallel method, multiple basic learners are constructed at the same time, and the independence of these basic learners is used to improve the performance of the final model. For example, bagging is one of the Ensemble learning of parallel methods; In the sequential method, multiple learners are constructed in sequence. When building the back learners, the back learners can avoid the errors of the front learners, so as to improve the performance after aggregation. For example, boosting is a kind of sequential method integration learning.

2.2.3.1 RF (Random Forest)

Random Forest algorithm is an extended variant of bagging algorithm. It's based on the bagging ensemble constructed by decision tree based learners, and the random attribute selection model is further introduced into the training process of decision tree.

Random forest is composed of multiple decision trees. Each decision tree is different. When building the decision tree, we randomly select some samples from the training data, and we will not use all the features of the data, but randomly select some features for training. Each tree uses different samples and features, and the training results are also different.

The principle of maximizing data set division is to make disordered data more orderly. The change of information before and after data set division is called information gain. The calculation formula is shown in equation 1. In this way, the information gain obtained by each feature after data set division can be calculated. The feature with the highest information gain is the best choice.

$$\begin{gathered}\text{Information entropy:}\quad Info(D) = -\sum_{i=1}^{m} p_i \, log_2(p_i) \\ \text{Information entropy after feature partition:}\quad Info_R(D) = \sum_{j=1}^{k} \frac{|D_j|}{|D|} \times Info(D_j) \\ \text{information gain:}\quad Gain(R) = Info(D) - Info_R(D)\end{gathered} \tag{1}$$

2.2.3.2 GBDT (Gradient Boosting Decision Tree)

The full name of GBDT is gradient boosting decision tree, which is a kind of boosting algorithm. It is an ensemble learning algorithm that the weak learner must be cart model and requires the sample loss predicted by the model to be as small as possible.

Boosting refers to adding multiple weak learners to produce a new strong learner. Generally speaking, it is equivalent to connecting multiple learners in series. If each weak learner uses$f_i(x)$, then the strong learner of Boosting can be expressed as shown in equation 2.

$$F(x) = \sum f_i(x) \tag{2}$$

CART classification tree is similar to the tree generation algorithm of ID3 and C4.5, except that Gini coefficient is used in feature selection, and the calculation formula of Gini index is shown in equation 3.

$$Gini(D) = 1 - \sum_{k=1}^{K} (\frac{|C_k|}{|D|})^2 \tag{3}$$

Similarly, information gain is also required, $Gini(D, A) - Gini(D)$.

2.2.3.3 XGBoost（eXtreme Gradient Boosting）

XGBoost is also an algorithm based on the boosting framework. In XGBoost, we take the difference of the second-order Taylor expansion of the loss function as the learning objective, which is equivalent to using Newton method for optimization to approximate the minimum value of the loss function, that is, to make the loss function be zero, The specific process is shown in equation 4.

$$F_n(x) = \sum_{i=1}^{n} f_i(x)$$

$$F_n(x) = F_{n-1}(x) - \frac{L'(F_{n-1}(x))}{L''(F_{n-1}(x))} \quad (4)$$

2.2.3.4 LightGBM（Light Gradient Boosting Machine）

LightGBM is a framework for implementing GBDT algorithm and supports efficient parallel training. The followings are its design philosophy : first, a single machine uses as much data as possible without sacrificing speed. Second, when multiple computers are parallel, the cost of communication is as low as possible, and linear acceleration can be achieved in computing.

In the XGBoost algorithm, the complexity of finding the optimal splitting point = the number of features × Number of split points × Therefore, if you want to make some optimization on XGBoost, you can start from the above three perspectives. The histogram algorithm in LightGBM is to reduce the number of split points, the unilateral gradient sampling algorithm in LightGBM is to reduce the number of samples, and the mutually exclusive feature binding algorithm in LightGBM is to reduce the number of features. And the latter two are the highlights of LightGBM.

The histogram algorithm discretizes the continuous floating-point features into k integers (that is, the idea of bucket bins), and performs gradient accumulation and number statistics according to the bin where the features are located. When traversing the data, the statistics are accumulated in the histogram according to the discretized value as an index. After traversing the data once, the histogram accumulates the required statistics, and then according to the discrete value of the histogram, Traverse to find the optimal segmentation point.

2.2.3.5 CatBoost

CatBoost, XGBoost and LightGBM are the three mainstream artifacts of GBDT, which are all improved implementations under the framework of GBDT algorithm. As the name says, CatBoost has made many improvements in the processing of category features.

The main innovations of CatBoost algorithm lie in the Ordered Target Statistics numerical coding method of category features, the feature combination method based on greedy strategy, the Ordered Boosting method to avoid prediction offset, and the use of symmetric binary tree as the base model, which has regular function and extremely fast prediction.

For category features, if the number of categories is small, onehot coding can be used, but if the number of categories is hundreds or thousands, using onehot coding will lead to an explosion in the number of features. CatBoost has designed a method based on the statistical value of prediction targets to convert category features into numerical features. The so-called Ordered Target Statistics coding method means that the samples are randomly scrambled first, and then each sample uses only the samples in front of it to calculate the numerical code of its category characteristics. This prevents the leakage of the label, and can reasonably evaluate the true effectiveness of this feature. However, this method will cause the numerical code estimation of the category characteristics of the samples in front of it to be inaccurate, In order to reduce this impact, CatBoost will design a random arrangement of multiple samples (4 by default), and select a random arrangement from them before each tree building. The above is the so-called Ordered Target Statistics coding method, which is also the most important innovation of CatBoost.

2.2.4 SMOTE

SMOTE (Synthetic Minority Over-Sampling Technique) is an oversampling method for unbalanced data sets. Based on the k-nearest neighbor technique, smote transforms unbalanced data sets into balanced data sets by synthesizing samples of the minority classes. Let the number of minority samples be N, and the kNN parameter be k. The specific synthesis method is as follows:

1. For a sample D of the minority class, find out its K neighbors of the same class through Euclidean distance, and randomly select one of its K neighbors as D', and let the coordinate vectors of D and D' be $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively. New sample point is generated by the following equation:

$$\boldsymbol{c} = \boldsymbol{a} + \text{rand}(0,1) \cdot |\boldsymbol{a} - \boldsymbol{b}|$$

Perform the above calculations for all the N samples of the minority classes.

2. According to the number of positive and negative samples, determine the sampling multiple T, and carry out step 1 for T times.

2.*3 Construction of our Machine Learning System*

In order to better present the details of our project, such as the data, research ideas, model construction, prediction results and other relevant information, we decide to develop an information system (IS) named Short-term Precipitation Forecast Based on Australian Meteorological Data. Besides presenting some basic information, it needs to quickly query the historical weather information of Australia according to the city's name and the date we give, and also vividly display the temperature, wind speed and humidity in that day. In addition, our system requires to be able to accept the new observation input by users, and predict the weather of the next day using the built-in previously-trained machine learning model.

2.3.1 Selection of developing framework and tools

Because the information scale in our project is not large, and there are not many interactive functions we required, we choose the lightweight web framework **Dash** to develop the information system. **Dash** is an efficient and concise Python framework, which is based on **Flask, Poltly.js** and **React.js**, and can help data analysts to quickly develop interactive data visualization web applications in pure Python.

In order to further reduce the amount of code and speed up the development, we almost universally adopt the **feffery-antd-components** (fac) library. It is redeveloped based on the famous **React UI** component library **ant design**, and introduces many practical components and features of **ant design** into Dash, which helps developers to quickly develop modern interactive web applications in a modular way with a small amount of pure Python code.

2.3.2 System design

Although the Dash framework does not strictly require us to use a specific system design pattern, considering the expansion and maintenance of the system in the future, we adopt the general MVC design pattern to realize the layered development of the IS. Simply speaking, the MVC pattern abstracts a web application into three levels:

Model: A model represents an object that accesses data. It can also have logic to update the controller when the data changes.

View: A view represents the visualization of the data contained in the model.

Controller: The controller makes effects on models and views. It controls the data flow to model objects and updates the view when the data changes. It puts the view out of the model.

Their relationship is shown in **Figure 3**.

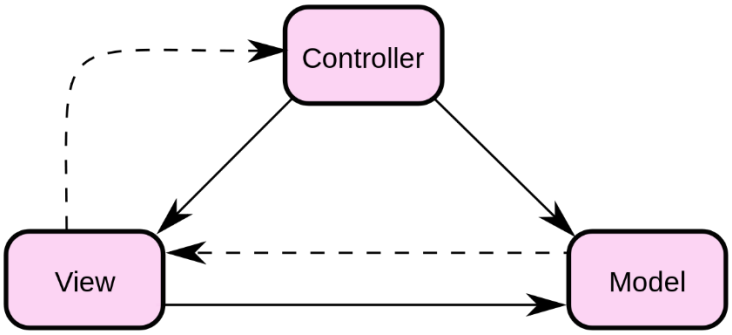

Figure 3 Relationships between Controller, View and Model

Specifically, in our development of the Dash application, we will encapsulate the operations on data into the model module; integrate interactive functions such as data query and model prediction into the controller (i.e. app.py); define the layout and content of the pages in the view module accordingly. These ideas will help us realize the decoupling development of the whole system, and facilitate the expansion and maintenance of the system in the future.

3. Results

3.1. Experiment Settings

3.1.1. Preprocessing Settings

The climate of Australia is mainly tropical desert climate and tropical savanna climate, with less rainfall. Therefore, in the collected data, rain and no rain account for 75.84% and 24.16% respectively, which is an unbalanced data set. At present, there are mainly two processing methods for unbalanced data sets: from algorithm level or data level. At the algorithm level, the commonly used method is to set different costs on misclassification. By giving different penalties for misclassification of two classess, the classifier cannnot simply predict all samples as the majority. At the data level, the distribution of positive and negative cases of the data set is adjusted by resampling the data set.

In this data set, positive samples account for 75.84%. If undersampling is used, many types of samples will be lost, resulting in serious information loss. If repeated sampling is used, the model will be over-fitted in the large number of repeated sampling data. Therefore, SMOTE method is adopted in this study, and some new samples are synthesized, which not only balances the data, but also reduces the risk of over-fitting to some extent.

In the experiment, we selected three different strategies for preprocessing: not using oversampling (control group), setting misclassification cost (experimental group 1) and using SMOTE oversampling (experimental group 2), and carried out experiments on the neural network model. The experimental results show that the F_1value is shown in **Figure 4**.

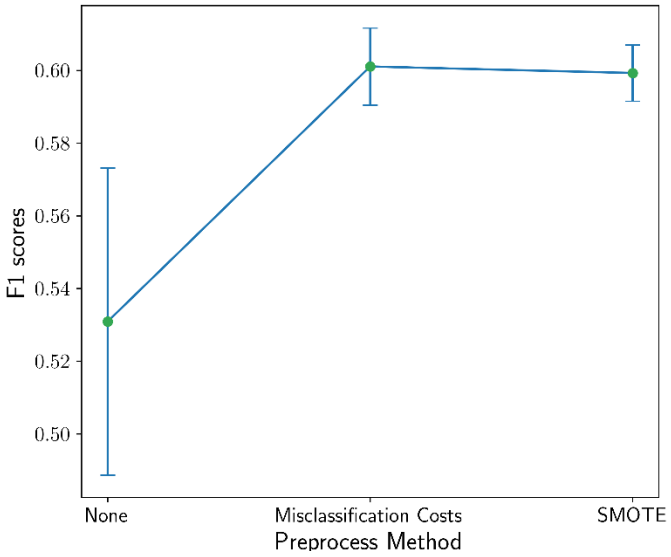

Figure 4 F_1scores on different preprocessing methods

It can be seen that the F_1 score of the model using preprocess methods has significantly improved. The variance of F_1 score of the model using SMOTE is small, and the mean value is almost the same as the model setting misclassification costs. Therefore, SMOTE method is selected as the preprocessing method in this study.

3.1.2 Criterion Settings

In this experiment, model accuracy and F_1 score on test set are used to evaluate the generalized ability of the model. In the unbalanced data set of this experiment, the lowest accuracy on the test set should be the accuracy of random guess, which is 77.48%. Only when the accuracy rate is higher than 77.48% can the model be regarded as effective. When all samples are predicted to be majority samples, the F_1 score of the model is 0. Therefore, the combination of accuracy rate and F_1 value can measure whether the model is better than random guess and whether all samples are simply predicted as the majority class.

3.2 Model Results

3.2.1 Results of Baseline

Because the logistic regression is easy to implement, the modeling idea is simple, and has good accuracy, in order to measure the effectiveness of the model used, the logistic regression is used as our baseline, and its predictive ability is as follows:

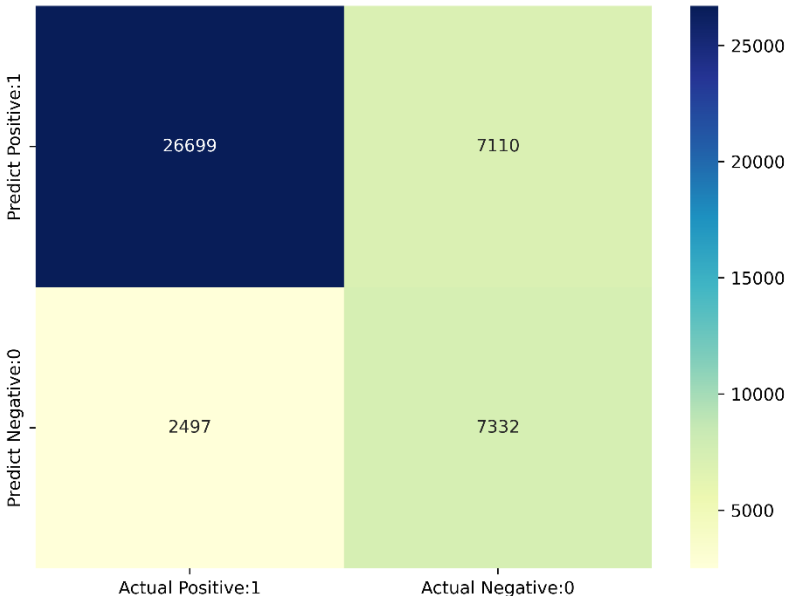

Figure 5 Confuse matrix of Logistics Regression

The Logistics model has an F_1 score of 0.60 and an accuracy rate of 77.98%.

3.2.2 Results of Models used

With Logistics regression as the baseline, repeat the experiment 50 times, and the accuracy of the model is shown in **Figure 6**.

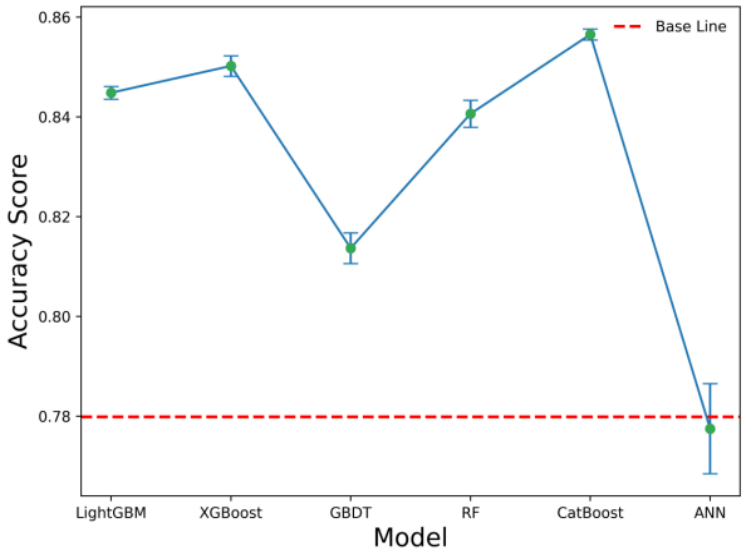

Figure 6 Accuracy scores of different models

It is found that the accuracy of ANN is not obviously improved compared with baseline, and the variance is large, so the stability of the model is poor. The accuracy of the other five ensemble models is significantly higher than that of baseline, and CatBoost is the best one, which has the highest accuracy, the smallest variance and higher model stability.

Similarly, with Logistic regression as the baseline, repeat the experiment 50 times, and the F_1 score of the models are in **Figure 7**.

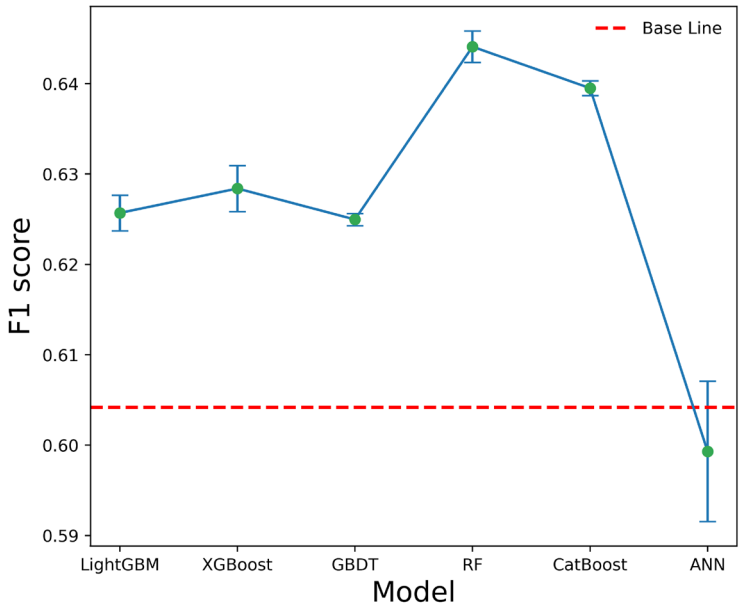

Figure 7 F_1 scores of different models

The results are almost consistent with the previous conclusions: the F_1 score of ANN is not significantly higher than that of baseline, and the variance is large, so the stability of the model is poor. The F_1 scores of the other five ensemble models are significantly higher than baseline, and random forest has the largest F_1 score, but the stability of the models is not so good as CatBoost whose F_1 score is second only to that of random forest.

Considering the accuracy and F_1 score, we think that CatBoost's model has the best fitting result, which takes into account the accuracy of prediction and the stability of the model.

According to the theory of CatBoost model, when dealing with category features, aiming at the problem of feature explosion caused by one-hot coding, the Ordered Target Statistics method is creatively used to transform category features, and a feature crossing algorithm based on greedy strategy is proposed, which will make CatBoost achieve better results on data sets with more category features than other ensemble models. Moreover, for the ensemble learning model, we can control the complexity of the model by adjusting the number of base classifiers and the depth of the tree, thus reducing the over-fitting degree of the model. Therefore, compared with the ANN model, ensemble learning has stronger generalization ability.

4. Discussion and Conclusion

The study uses the daily weather data of 49 cities of Australia within ten years to carry out experiments, adopts the SMOTE method to solve the problem of label imbalance, tries to use Logistic regression, ensemble learning and neural network to predict the next day's weather in a given Australian city, and lastly selects the best model for the construction of our web application according to the evaluation indexes. In addition, the information system we built can also provide weather forecast services well, and its details are shown in the Appendix below.

To conclude, SMOTE solves the imbalance problem quite well. It can not only balance the data set, but also, to some extent, reduce the over-fitting on samples of the minority class. Among various prediction algorithms, ensemble learning methods perform the best in terms of accuracy and stability. Compared with the ANN model, ensemble learning enjoys a stronger generalization ability, for people can control the complexity of the model by adjusting the number of base classifiers and the depth of the tree. Among all the ensemble models used, CatBoost has the best effect, for which the adoption of

"Ordered Target Statistics" and a feature crossing algorithm based on greedy strategy make this model better than other ensemble models on datasets with more categorial data. Our study can be used as a reference for future weather prediction, which will facilitate people's production and daily life.

Author Contributions: Conceptualization, Y.S. and P.W.; methodology, H.C. and P.W.; software/information system, Y.S.; formal analysis, Y.S. and H.C.; investigation, N.W. and K.L.; data resources, P.W.; writing—original draft preparation, N.W. and K.L.; writing—review and editing, Y.S. and H.C.; visualization, P.W. and H.C.; supervision, Y.S.; project administration, Y.S.; All authors have read and agreed to this version of the manuscript.

Appendix

The source code of our machine learning system has been hosted by GitHub, and the repository's name is **DashboardForRainInAustralia**. The website was designed in Chinese. Some of the pages are as follows. You can visit it by searching http://150.158.97.38:5000/ in the browser.

Figure 8 The homepage of our ML system

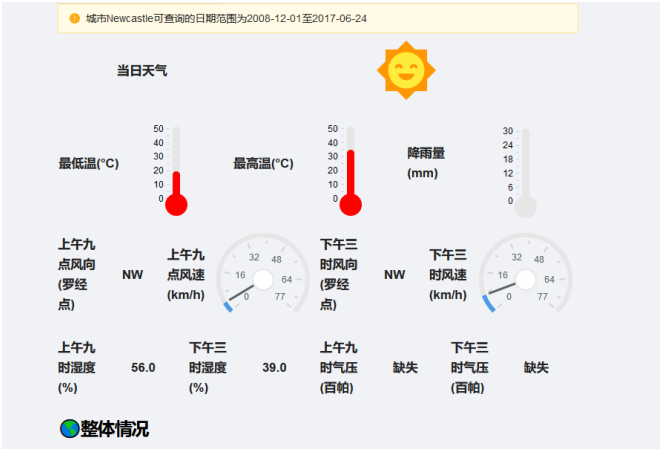

Figure 9 A screenshot of the website

References

1. Shan.L,Yao.Y, Zhao.Q, Liu.L Short term and impending rainfall prediction method based on GNSS PWV *Surveying and mapping geographic information*, 2019,44 (01): 22-26 DOI:10.14188/j.2095-6045.2017102.
2. Shan Y. Application of high order Markov chain in rainfall prediction, *Xiamen University*, 2019.
3. Li.J Zhang.W, Ma.Y Application of moving average Markov model in precipitation prediction, *Water and soil conservation research*, 2005 (06): 200-202+209.
4. Hu.Y, Du.J, Dong.Y, Teng.D Rainfall prediction based on improved adaboost-c4.5 algorithm, *Modern electronic technology*, 2021,44 (14): 6-10 DOI:10.16652/j.issn. 1004-373x. 2021.14.002.
5. Guan.P, Wang.M, Li.Q. Short term rainfall prediction model based on ensemble learning and deep learning, *[publisher unknown]*, 2017:2-23.
6. Zhao.Q, Liu.Y, Yao.Y Construction of short-term and imminent rainfall prediction model using least squares support vector machine, *Geodesy and geodynamics*, 2021,41 (02): 152-156 DOI:10.14075/j.jgg. 2021.02.008.
7. Huang.J, Qin.L Study on temperature and rainfall correlation prediction using alstm model, *Journal of Guangxi University (NATURAL SCIENCE EDITION)*, 2021,46 (04): 1024-1035 DOI:10.13624/j.cnki. issn. 1001-7445.2021.1024.
8. Zhang.J, Wang.P, Zhang.L, Wang.Y Application of artificial neural network in Short-term Precipitation Prediction, *Science and technology wind*, 2016 (17): 123-124 DOI:10.19392/j.cnki. 1671-7341.201617107.
9. Liao.M Spatiotemporal prediction of rainfall in semi-arid areas based on deep learning, *Lanzhou University*, 2021 DOI:10.27204/d.cnki. glzhu.2021.001501.