INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 2 - V7I2-1543)
Available online at: https://www.ijariit.com

Machine Learning Algorithms to Predict Next Day Rain in Australia

P. Sai Dinesh Reddy
psai.dineshreddy2017@vitstudent.ac.in
Vellore Institute of Technology, Vellore, Tamil Nadu

ABSTRACT

This paper predicts whether it will rain next day or not in Australia. This paper compares 4 machine learning algorithms namely Random Forest Classifier, XGBoost, Light GBM and Logistic Regression models by training and testing it with the data set. The XGBoost Model performed the best when compared to Random Forest Classifier, Logistic Regression and LightGBM. XGBoost produced an accuracy of 94.03%. The models could have performed better if date was treated as a cyclic continuous feature because weather itself is cyclic which shows a similar trend during the same seasons.

Keywords — *Weather Prediction, Random Forest Classifier, Logistic Regression, Lightgbm and Machine Learning.*

1. INTRODUCTION

Usually, for prediction, the weather people often require domain expertise and they use numerical weather prediction. If the prediction of weather fails the amount of loss will be tremendous. If someone has the historical weather data of those particular places or regions then one can predicted the possible weather. To achieve this one has to train deep learning or machine learning models. These techniques will tremendously increase the accuracy of weather prediction. If one could predict rain well in advance, so farmers can harvest their crop at a safe time or one could plan their next day accordingly. There are immense advantages of predicting next day rain well in advance [1].

In this paper, the author used raspberry Pi to collect data from the real world but they just trained their data against random forest classifier and not compared their accuracy by training against other machine learning models [2]. This paper has compared 7 machine learning methods namely Genetic Programming, Support Vector Regression, Radial Basis Neural Networks, M5 Rules, M5 Model trees, and k-nearest neighbors for rainfall prediction in weather against 42 cities [3].

In this paper, used machine learning techniques which require far less computational power when compared to any deep learning techniques [4]. Performance of the model was increased by pre-processing the data before feeding the data to the models, which is essential to produce accurate results.

2. METHODOLOGY ADOPTED

This paper performed initial data exploratory analysis of the weather data set, therefore found noticeable insights. Understood that this is a binary classification problem. Handling the outliers and then treating the null values. Finally, the data was split into training and testing. After training all the models were tested against the testing data. The given diagram gives details of the methodology adopted.

Fig. 1: Flow chart of the methodology adopted for predicting the next day rain in Australia.

3. DATA SET DETAILS

The data set consists of weather details from the year 2009 to 2017. There are a total of 23 variables and 14,5459 instances. A lot of missing data was present irrespective of the column which was taken care of during the pre-process of the data set.

The columns were labelled as Date, Location, MinTemp, MaxTemp, Rainfall, Evaporation, Sunshine, Wind, GustDir, WindGustSpeed, WindDir9am, WindDir3pm, WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm, Pressure9am, Pressure3pm, Cloud9am, Cloud3pm, Temp9am, Temp3pm, RainToday and RainTomorrow. These information were found during the initial data exploration.

Humidity9	Humidity3	Pressure9	Pressure3	Cloud9am	Cloud3pm	Temp9am	Temp3pm	RainToday	RainTomorrow
71	22	1007.7	1007.1	8	NA	16.9	21.8	No	No
44	25	1010.6	1007.8	NA	NA	17.2	24.3	No	No
38	30	1007.6	1008.7	NA	2	21	23.2	No	No
45	16	1017.6	1012.8	NA	NA	18.1	26.5	No	No
82	33	1010.8	1006	7	8	17.8	29.7	No	No
55	23	1009.2	1005.4	NA	NA	20.6	28.9	No	No
49	19	1009.6	1008.2	1	NA	18.1	24.6	No	No
48	19	1013.4	1010.1	NA	NA	16.3	25.5	No	No
42	9	1008.9	1003.6	NA	NA	18.3	30.2	No	Yes
58	27	1007	1005.7	NA	NA	20.1	28.2	Yes	No
48	22	1011.8	1008.7	NA	NA	20.4	28.8	No	Yes
89	91	1010.5	1004.2	8	8	15.9	17	Yes	Yes

Fig. 2: The above picture is a sample of the data set.

4. PRE-PROCESSING AND MODEL IMPLEMENTATION DETAILS

Further exploration of the data revealed that there are a lot of outliers. The methodology adopted by this paper to detect outliers is if a value is greater than the 1.5 *IQR above the upper quartile (Q3) or below 1.5*IQR (Inter quartile range) the value will be considered as outlier. These outliers were dropped.

In this paper, the data set which one has to deal is imbalance as number of "No" cases were more when compare to "Yes". If the imbalance were not treated well in advance, the machine learning model would get biased to "No" because a greater number of cases were "No". To overcome this oversampling was performed on the data set. As most of the machine learning algorithm cannot deal with missing values, we have to handle the missing values As KNN imputer is prone to outliers and we have handled the outlier before feeding the missing values to the model. For handling the missing values, we used K-Nearest Neighbors imputer [5]. After a thorough analysis we used k value to be 5. Min max scaler was applied to the data set after this step.

Though there are a lot of columns, date and Sunshine was drop. Both the variables will not have any significant use in predicting the target value and a lot of data was missing in case of sunshine. Split the data into training and testing, 70 and 30 percent respectively. Target parameter was the next day's rain.

In order to predict the next day's rain, we compared in total 4 Machine learning models namely Random Forest Classifier, XGBoost, LightGBM and Logistic Regression [6] [7] [8]. Evaluated models were evaluated using suitable methods.

In order to evaluate these algorithms, we used precision which is given by the formula

$$Precision = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Positive\ (FP)}$$

Recall which is given by the formula

$$Recall = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Negative\ (FN)}$$

f1 is given by the formula

$$f1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

5. RESULT

As this is a Classification problem, we will use Precision, recall and accuracy to evaluate a model's performance. After validation, each model with the testing data set revealed that the logistic regression model achieved an accuracy of 77.64%, random forest classifier achieved an accuracy of 89%, light GBM model achieved an accuracy of 83% and XGBoost achieved an accuracy of 94.03%. All the model's accuracy was compared in the fig3.The following table gives the necessary information to evaluate the model. XGBoost performed the best when compared to other models. As f1 score is good for XGBoost it clearly shows that we have achieved a good precision and good recall.

Table-1: Matric for evaluating XGBoost

-	Precision	Recall	F1-score	Support
0	0.96603	0.91270	0.93861	32876
1	0.91727	0.96791	0.94191	32876
Accuracy	-	-	0.94031	65752
Macro avg	0.94165	0.94031	0.94026	65752
Weighted avg	0.94165	0.94031	0.94026	65752

Table-2: Matric for evaluating Light GBM

-	Precision	Recall	F1-score	Support
0	0.83645	0.82732	0.83186	32876
1	0.82919	0.83824	0.83369	32876
Accuracy	-	-	0.83278	65752
macro avg	0.83282	0.83278	0.83278	65752
weighted avg	0.83282	0.83278	0.83278	65752

Table-3: Matric for evaluating Random Forest Classifier

-	Precision	Recall	F1-score	Support
0	0.83645	0.82732	0.83186	32876
1	0.82919	0.83824	0.83369	32876
Accuracy	-	-	0.89641	65752
Macro avg	0.89723	0.89641	0.89636	65752
Weighted avg	0.89723	0.89641	0.89636	65752

Table-4: Matric for evaluating Logistic Regression

-	Precision	Recall	F1-score	Support
0	0.76972	0.78878	0.77914	32876
1	0.78342	0.76402	0.77360	32876
Accuracy	-	-	0.77640	65752
Macro avg	0.77657	0.77640	0.77637	65752
Weighted avg	0.77657	0.77640	0.77637	65752

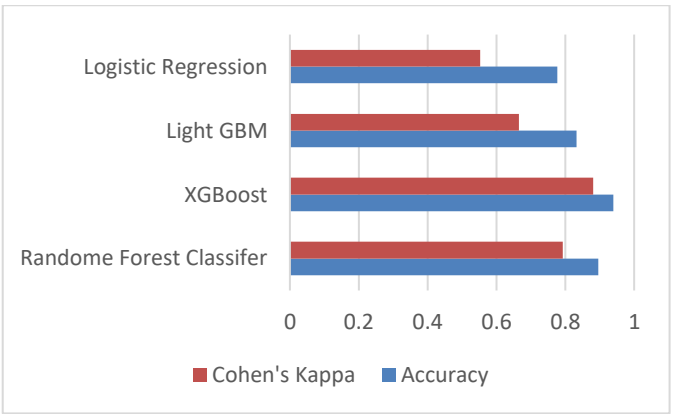

Fig. 3: following graph compare models by their accuracy and cohen's kappa

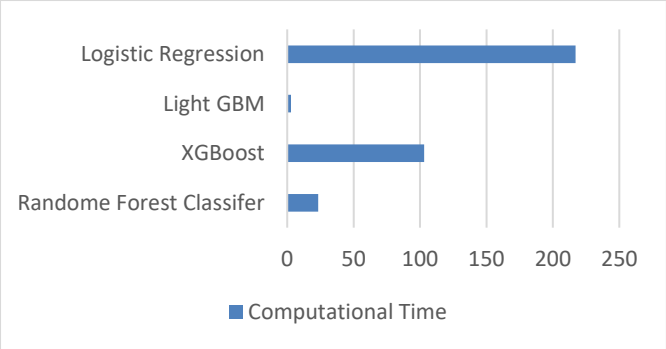

Fig. 4: following graph compares the model's Computational time

6. CONCLUSION AND SCOPE OF IMPROVEMENT

In conclusion, we can observe that XGBoost performed better compared to other models. As far as speed is concerned Random forest classifier out performed other models which we were comparing. Finally, we can conclude that if speed is concerned, we should use Random forest but in terms of accuracy XGBoost performed better for this data set.

Instead of predicting the possibility of rain irrespective of area if we create and segregate the data location wise. Using the segregated date to train and predict the accuracy of each model will increase irrespective of the model. Reason for this is each location will have different conditions so treating all the sublocation of Australia different will increase the accuracy. Another approach could have been treating the data as cyclic.

7. REFERENCES

[1] K. Prasad, "oneindia," 17 September 2018. [Online]. Available:https://www.oneindia.com/agriculture/advantage-of-weather-forecast-services-2777426.html.

[2] N. Singh, S. Chaturvedi and S. Akhter, "Weather Forecasting Using Machine Learning Algorithm," in International Conference on Signal Processing and Communication (ICSC), 2019.

[3] S. Cramer, M. Kampouridis, A. A. Freitas and A. K. Alexandridis, "An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives," Expert Systems with Applications, vol. 85, pp. 169-181, 2017.

[4] E. Hernandez, V. Sanchez-Anguix, V. Julian, J. Palanca and N. Duque, "Rainfall Prediction: A Deep Learning Approach," in International Conference on Hybrid Artificial Intelligence System, 2016.

[5] G. E. A. P. A. Bastista and M. C. Monard, "A Study of K-Nearest Neighbour as an Imputation Method," in HIS, 2002.

[6] Breiman and Leo, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 32-45, 2001.

[7] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference, 2016.

[8] Ke, G. a. Meng, Q. a. Finley, T. a. Wang, T. a. Chen, W. a. Ma, W. a. Ye, Q. a. Liu and Tie-Yan, "Lightgbm: A highly efficient gradient boosting decision tree," Advances in neural information processing systems, pp. 3146-3154, 2017.

