Weather Forecasting in Australia

KHOO CHUN WEN

Contents

Problem Definition

For this assignment, we have been tasked with gaining familiarity with building various classification models using R. Specifically, we would use these models to predict whether it would rain tomorrow in Australia based on ten separate locations within the country.

Data Preparation and Data Cleaning

In the first stage of every data analysis problem, we first explore the data, in order to gain initial insight, and to discover data quality issues, if any. The initial dataset, WAUS2019.csv, has 100,000 rows. From this, I have sampled 2,000 rows of 10 random locations without replacement, using my student ID as the random seed.

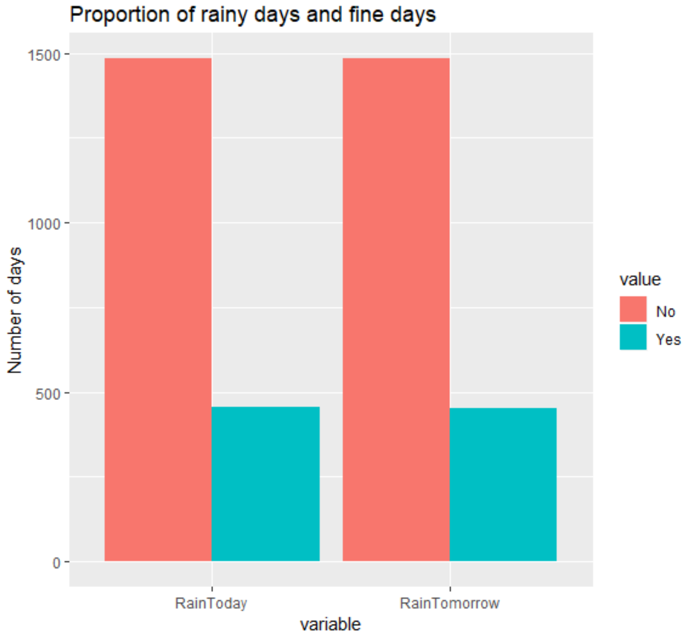

Figure 1 - Bar graph of proportion of rainy days and fine days for both actual and predicted outcomes

For starters, from the reduced dataset I will be using for the rest of this task, we can see that the proportion of actual rainy days to actual fine days is roughly 1:3.238. Thus, for every rainy day observed, we see approximately four fine days. Similarly, we can also observe that the proportion of predicted rainy days to predicted fine days is 1:3.252. This means that, for each rainy day observed, we expect there to be three, maybe four, fine days. This ties in with our earlier observation of the proportion of actual rainy days to actual fine days. It is also clear that there is an obvious class imbalance issue. However, this may just be due to the sampled locations having less rainy days.

There are two different types of data in this dataset, namely, categorical attributes, as well as numerical attributes. For the real-valued attributes, we can obtain descriptive statistics of these predictors, as seen in Figure 5. Interestingly, there are several variables with low variances, such as Evaporation and Sunshine. Thus, this may be an indication that these variables would not be good predictors. According to Alexander (2012), variables of high variance shows that there is a wide range of possibilities, both good and bad, that have a greater probability of occurring. Therefore, the converse that variables of low variance shows that there is a narrower range of possibilities, and would not be optimal for prediction, is true.

From this dataset, there are various independent variables that I believe would not be necessary to predict the weather. As an example, the variables Day, Month, and Year would not significantly affect the classification. This is because the weather does not depend on the specific date. Thus, these variables are removed from the dataset before proceeding to the next step. Furthermore, the variable Location is also removed, due to the sole reason that we are only looking to predict if it will rain tomorrow. Hence, where it rains is irrelevant in this context. Finally, I also removed the variables Evaporation, Sunshine, Cloud9am, and Cloud3pm, since these predictors contain too many null values. As seen in Figure 4, the variables stated contain approximately 40-46% NAs, thus would not be influential in the classification. Finally, I split it into two separate sets using my student ID number as the random seed, of which 70% would be used for training the models, and the remainder would be used to evaluate the models.

Building the Models

Using the modified dataset, I then built five different classifiers, utilizing the decision tree, the Naïve Bayes method, bagging, boosting, and random forests. All classifiers were generated using the default settings from each of their respective packages, apart from the bagging and boosting models, as these were generated using only 10 trees. Using the models built, we can then evaluate each of these classifiers based on their accuracy, confusion matrices, and the area under the receiver operating characteristic curves.

Model Evaluation

Using the test set with each classifier, each of the test cases are classified as ‘will rain tomorrow’ or ‘will not rain tomorrow’, and these can be summarized in a confusion matrix. From their respective confusion matrices, we can then calculate the accuracy of each model, which is done by summing up the true positives and the true negatives, divided by the total number of observations. Additionally, we can also calculate the confidence of predicting ‘will rain tomorrow’ for each case and construct the ROC curve of each classifier. A ROC curve visualizes the ‘goodness’ of a classifier through graphing varying confidence threshold values of a single classifier, whereas a confusion matrix only provides comparison at a specified confidence threshold.

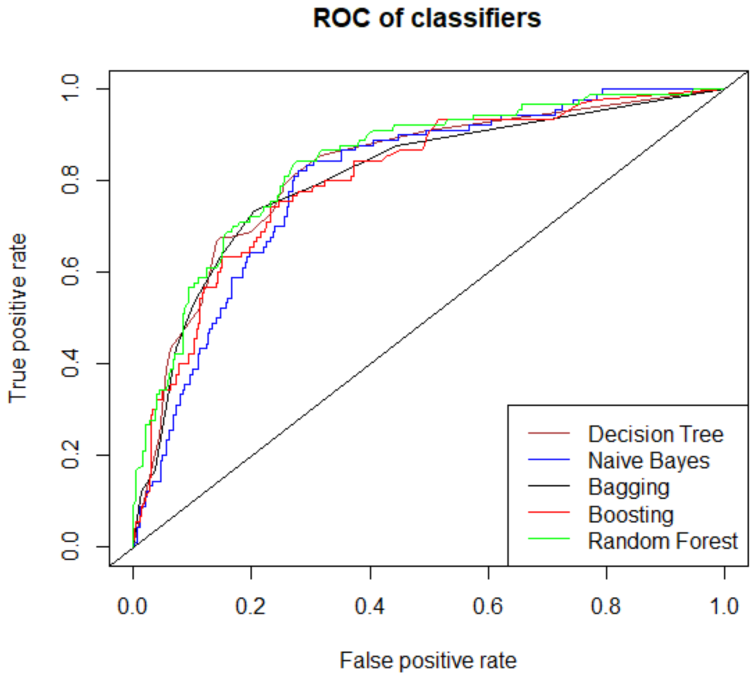

Figure 2 - Receiver Operating Characteristic curve of each classifier

From Figure 6, observe that the decision tree has the second-best accuracy, but the lowest AUC. This can be attributed to the fact that decision trees have high variance, hence one cannot be confident with the classifications of a decision tree. It can also be said that the Naïve Bayes model performs the worst, with the lowest accuracy and a relatively low AUC. This is because the Naïve Bayes method assumes predictor independence and may lose some of the underlying context. Conversely, we see that the random forest produced the highest accuracy with the highest AUC. Thus, we can conclude that the random forest best classifies this dataset.

Important Predictors

By examining the output of each model apart from the Naïve Bayes method, we can determine the important predictors used to classify each test case. The common factors from the decision tree, bagging, boosting, and random forest methods are WindGustDir, WindDir9am, WindDir3pm, and Humidity3pm. Therefore, I believe that these variables are more important in predicting whether or not it will rain tomorrow, more so than others. Additionally, we can also remove several predictors, such as MaxTemp, RainToday, Temp9am, WindSpeed9am, and WindSpeed3pm, as these variables have been deemed unimportant by the four classifiers mentioned.

Experimentation: Parameters, Pruning, and Cross-Validation

For starters, I performed cross validation and pruning on the initial decision tree. Based on the output, I decided to use the tree with 5 terminal nodes, as it has the lowest standard deviation. As seen in the table in Figure 7, pruning the decision tree has resulted in a higher accuracy, but smaller AUC, as compared to the original decision tree.

Additionally, I also tried increasing the final number of trees in the bagging method. By doing so, the overall accuracy of the model has increased to 83.38%, with an area under the ROC curve of 0.8365. Thus, we can see that increasing the final number of trees has slightly improved our accuracy from 81.33%, as well as the AUC by approximately 0.03.

Furthermore, the bagging and boosting methods can be improved upon by applying cross-validation. In this case, I have applied 10-fold cross validation, along with the bagging method, 100 final trees, and a complexity parameter of 0.01. This has resulted in a slight increase in accuracy, from 83.91% to 86.45%. I have also implemented 10-fold cross validation in the boosting methods, along with a complexity parameter of 0.01 and a Breiman learning coefficient. We see an increase in accuracy, with it jumping from 83.05% to 83.37%

Thus, by experimenting with different parameters, I was able to generate a better classifier from the initial ones built earlier, using cross-validation with the bagging method, producing an accuracy of 86.45%.

References

Alexander, J. (2012). Look Out for High Variance. Retrieved from https://codermetrics.org/2012/04/11/high-variance-lookout/