**Chart

Description automatically generatedExploratory of variables**

**Based on the statistical summaA picture containing table

Description automatically generatedry of the dataset ( figure!! ), there are serveral noteworthy insights explained below:**

* **The proportion of the target variable is 30.65% ( Not cloudy = 613, Cloudy = 1387 ).**
* **The mean of Pressure9am and Pressure3pm are not** **ably high.**
* **Most of the variables are skewed. For example, Rainfall and Evaporation are extremely right-skewed. That shows its median is larger than its mean. Contrariwise, Humidity9am and Sunshine are left-skewed.­**
* **A majority of variables contain NA values. Evaporation and Sunshine have 43% and 39% missing values.**
* **Since this analysis is predicting whether the following day will be cloudy, some variables are not essential to be predictors, namely Day, Month, and Year; hence they will be omitted.**

**On top of the insights, based on ( figure!! ), Rainfall is the most extreme variable as it has the highest range and lots of outliers. Few variables like Temp3pm and MaxTemp are of similar distribution.**

**2. Data Cleaning and Pre-processing ( half page )**

**Before modelling, the data pre-processing should be conducted which is gleaned from the part 1 summary.**

**2.1. Dealing with Categorical values**

**The data type of multiple variables is “Char” which is specified to be categorical. I used as.factor ( ) to categorise the variables of all wind direction, RainToday, location, and CloudTomorrow.**

**2.2. Handling missing values and outliers**

**All rows with NA values are removed as a compromise between complexity for modelling and information loss. Generally, it is assumed that these data points are missing at random (MAR), meaning that the tendency for a data point to be missing is not related to the missing data. The cause perhaps was human errors during data collection, or some points of sensors functioned improperly on those days that weather cannot be detected. Moreover, the analysis involves the comparison between models. Some of the classifiers like Boosting and ANN cannot handle missing values while others do. This context must be ensured that the dataset needs to be consistent when comparing models, therefore removing missing values is necessary.**

**On top of that, since all models are robust to outliers, they are not removed.**

**2.3. Data Splicing**

**At this stage, the dataset is cleaned and ready to model. The dataset is split into 70% for training and 30% for testing, with a random seed set as 31084222.**

**3. The comparison of the original models**

**The following is the code for five classifiers, all of which are at their default settings.**

#Decision Tree

tree.train = tree( CloudTomorrow ~ ., data=WAUS.train )

#Naïve Bayes

NB.train =naiveBayes( CloudTomorrow~.,data=WAUS.train )

#Bagging

bag.train= bagging( CloudTomorrow ~. ,data = WAUS.train, mfinal = **6** )

#Boosting

boosting.train= boosting( CloudTomorrow ~. ,data = WAUS.train, mfinal = **7** )

#Random Forest

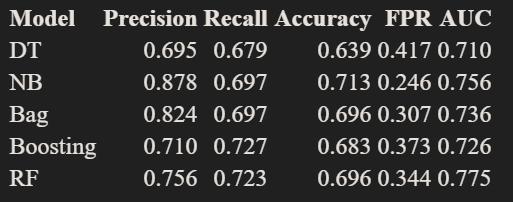
rf.train= randomForest( CloudTomorrow ~. ,data = WAUS.train )

**Assumption for the Naïve Bayes model: Naïve means that all variables are independent; however, since in real life it can hardly be achieved. Therefore, it is assumed that all variables are independent.**

**After testing each model, Random Forrest (RF) is the best model. The reasons are explained below:**

**Firstly, although the accuracy is lower than Naïve Bayes model, the difference is minor, and the value for RF is sufficiently high ( Figure !! ). That means there is 70% for RF to classify if tomorrow is cloudy correctly. The values also overall are higher than other models. The importance of each variable by the classifiers**

Figure 1 The summary of the performance for each model



Chart

Description automatically generated with medium confidence

**Secondly, AUC is where the best model was differentiated. Even though the accuracy of Bagging and RF is the same, the AUCs are different. That is, the dataset is biased towards “Cloudy” or “Not Cloudy”. Bagging might discriminate well on either side but RF tends to perform well overall.** AUC is the proof of this case since it is an evaluation of the classifiers as threshold varies over all possible values. It rates comparatively higher than others. The reason might be that b**y** partitioning the sample and randomly choosing the variables to model subtrees, the samples are more enriched, thus avoiding biases.

­

**3.8 The variable importance**

**Only Humidity3pm, MinTemp, Sunshine,** **WinDir3pm, WinDir9am and WinGustDir could be selected from the data because they are of higher effect on performance. The way the features are selected is based on the Gini index, which is a measure of variance. The higher the variance signifies the more misclassification exists, thus driving the values of Gini Index higher.**

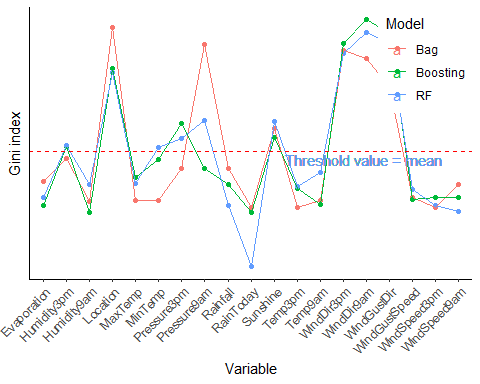


Figure 2 The importance of each variable by the classifiers

**In this scenario, ensemble methods like Bagging, Boosting and Random Forrest are used to determine the variable importance. The reason I do not use Naïve bayse and other classifiers is that their variables are of equal weighting. Whereas methods like Boosting are suitable as their task is to reduce the variance error; they analyse different subsets of data and generate a collective output at the end. Since Boosting weighs its variables from the bags, it is particularly useful. Therefore, variable importance is based on ensemble methods.**

**Three models follow similar patterns on weighing its variables. Hence I set the threshold value to the mean ( figure!! ). Only features higher than the mean, could they be selected, which yields better classification.**

**4.1. Evaluation of the improved model based on simplicity**

A decision tree is a simple classifier for a person to classify if it will be cloudy or not tomorrow.

Feature selection:

The conditions of being windy, cold, sunny, and high pressure are the four factors affecting the decision of the class, all of which are based on the mentioned important variables. For the sake of simplicity in modelling, all predictor variables are categorical. Therefore, threshold values are set to determine if the level of those conditions, which are listed below:

*1. Pressure is high if Pressure3pm and Pressure9am> 1013. (Standard average pressure)*

*2. It is cold when “MinTemp” < 10.*

*3. It is sunny if the hours of “sunshine” > 12.*

*4. It is windy if the “WindSpeed3pm” and “WindSpeed9am” > 15*

The process of modelling:





1. 15 rows were sampled from the WAUS.Train for training the model.
2. 5 rows were sampled from the WAUS.Test for test the model
3. Calculate the entropy Initial State.



3.1 calculate the Information Gain for each predictor variables.



Table 1.1. Information gain: Cold



Table 1.2. Information gain: Pressure



Table 1.3. Information gain: Sunshine



Table 1.4 Information gain: Windy

From the tables above, Sunshine has the greatest information gain, thus becoming the root node.

* 1. Attribute to split on next

Gain( Not Sunny, Windy )=0.2212

Gain( Not Sunny, Cold )=0.0968

Gain( Not Sunny, Pressure )=0.0642

The above are the values of information gain for each variable given that the day is not sunny. “Windy” has the highest value, therefore it is the node when it is not sunny.

Gain ( Sunny, Windy )=0.9710

Gain ( Sunny, Cold )=0.9710

Gain ( Sunny, Pressure )=0.9710

Likewise, the above is the values of information gain given that the day is sunny. Since one can observe that all the values are the same, therefore, the algorithm will select the feature randomly, in which “Pressure” is selected as a node when it is sunny.

* 1. Attribute to split on next

Gain ( Not Windy, Cold ) = 0.4553

Gain ( Windy, Cold ) = 0.4200

Diagram

Description automatically generatedSince the value for “Cold” given that it is not windy is larger, the node of cold will be selected when it is not sunny and not windy.

The accuracy of the model = ( 1+2 ) / ( 5 ) = 0.6

The recall of the model = ( 1 ) / ( 2+1 ) = 0.3

Although the accuracy of this model is merely different than other classifiers, that does not necessarily indicate that this model is good. Given the limited number of the training dataset, this model can be highly biased. Perhaps it is the randomness of the testing set that drives this accuracy that this model has 60 % in classify the label. However, the label of the training set is mostly “Not Cloudy” and “Not Sunny”. That is, it might be good to classify “Not Cloudy”, but the Recall rate and AUC could be very low. Hence, this model cannot be better than other classifiers.

**4.2. Evaluation of the improved model based on performance**

As Random Forrest is the best model, it is used to optimise so as to get the best result. The accuracy went up from .69 to .72 whereas AUC remained the same.

It is improved by adjusting the optimal parameters using trial and error, with the code as shown below.

rf.train= randomForest(CloudTomorrow ~. ,data = df.improved.train,

ntree = 50000, importance= T, mtry = 2, nodesize=42)

Importance = T **was adjusted so that the importance of predictors to be assessed. By removing the low-important features could decrease the time complexity of the algorithm. Also, it can increase the accuracy by reducing the variance.**

mtry = 2 **was to tune the number of predictors sampled for splitting the trees from 4 to 2. Since RF’s terminal nodes of each tree could be very small.**

ntree = 50000 was to increase the number of trees

nodesize=42 was to tune the m**inimum size of terminal nodes**

Note : The reason why Cross-validation was not used was that it utilised too many resources to increase .1 in accuracy with lower AUC, which is not worthy.

This model is found to be the best in terms of time complexity and performance. By setting up more trees means higher accuracy at the expense of slower learning. When realising this trade-off, some parameters above are to balance the runtime so that the time complexity would not beyond the one at default setting. For example, setting the node size **larger makes trees smaller to be grown, thus taking less time. And waiving the low-important variables can also achieve this purpose. Thus, tunning the parameters allows the model to achieve higher accuracy meanwhile running at the similar efficiency to the RF at default setting.**

**5. Artificial Neural Network classifier**

The features are selected based on the one done in Question 8. Through trial and error for modelling ANN, it is concluded that the variables of the best model are all categorical, all of which are converted to binary columns. This means that scaling is not needed. As mentioned above, all the rows with missing values are removed for model comparison, therefore, this step can be skipped.

nn <- neuralnet(CloudTomorrow0 + CloudTomorrow1 ~ . ,data = nn.train[,],

hidden=3, linear.output = F, threshold=0.01)

A picture containing chart

Description automatically generated**This comparison between ANN and other classifiers**

The performance of ANN does not result as great as others. Its accuracy is only .513 and AUC is .572. As shown in the ROC curve (figure!!), with most of the points lying towards the centre, ANN (Pink) does not perform as well as other classifiers, which indicates that lots of cloudy classes are labelled at random.

Generally, there are two reasons, in the aspects of dataset and fundamental.

*The model is overfitting and has high variance:* 2000 samples are not sufficient for a complex model like ANN to fit. ANN is considered a data-hungry model which needs millions of samples to form a big network in which its performance is strongly depended on the level of noise and richness of the dataset. However, if the samples were increased, then the trade-off is that it is incomparable to other classifiers with different size of datasets to draw an arbitrary conclusion.

*ANN model does not suit for solving this problem*: Evidently, no matter how much learning rate and threshold the ANN was changed in the parameter, the performance remains the same. That is, the model does not classify well if tomorrow is cloudy or not. The nature of problem defines what models to use. There is no the “perfect” classifier for solving all problems. As shown in the ROC curve, the only way is to implement every possible classifier that could work on the dataset, and then decide which model perform best for the problem. ANN is known to solve complex problems like image recognition but not the other ones. So, it is rational to infer that the model might not suit well for this problem.