Practical Application 2

Machine Learning Assignment

David Cabornero Pascual
david.cabornero@alumnos.upm.es

June 18, 2022

1 Introduction

In this assignment, a Dataset from the Australian Government Bureau of Meteorology will be used to predict the weather observed the next day of the observation. In fact, there are two classes to predict: whether it rains or not the following day.

The main objective of this assignment is to implement three classification algorithms (Logistic Regression, Bayesian Classifiers and Discriminant Analysis) and some Metaclassifiers and analyze them in four cases:

- With all variables.
- With an univariate filter feature subset selection.
- With a multivariate filter feature subset selection.
- With a multivariate wrapper feature subset selection.

After that, we will analyze results like accuracy, F-score or execution time and extract some conclusions about them. Additionally, we also have to infer the behavior of the algorithms when it's possible with the tools provided by *Weka*.

2 Problem Description

The Dataset [2] trained in this Assignment is from the Australian Government and provides weather information about each day from 01/11/2007 to 25/06/2017. With this variables, we have to be able to predict whether it rained the following day. The 22 features that form the dataset are:

- *Date*: the date of the observation. This variable needs to be removed or modified to be useful, since we can get advantage of the month or the season, but not of the date itself.
- *Location*: the place where the observation is carried out. It's necessary, since there may be drier zones.
- *Rainfall*: the amount of rainfall on the day expressed in cubic millimeters per square millimeters (broadly speaking, that's what we usually know as millimeters).
- *Evaporation*: measures the effective evaporation in millimeters with the Class A evaporation pan method (we look how many millimeters of the water in the pan have been evaporated during the last 24 hours).

- *Sunshine*: number of hours of sunshine.
- *MinTemp*, *MaxTemp*, *Temp9am*, *Temp3pm*: shows the temperatures detected in 4 cases: the minimum, maximum, at 9 A.M. and at 3 P.M. It's expressed in degrees Celsius.
- *WindGustDir*, *WindGustSpeed*, *WindDir9am*, *WindSpeed9am*, *WindDir3pm*, *WindSpeed3pm*: we have the direction (categorical variables expressed in cardinal directions) and the speed (numerical variables expressed in km/h) in 3 cases: the strongest wind gust, 9 A.M. and 3 P.M.
- *Humidity9am*,*Humidity3pm*: percentage of humidity at 9 A.M. and 3 P.M.
- *Pressure9am*, *Pressure3pm*: the atmospheric pressure measured in hectopascals (hpa) at 9 A.M. and 3 P.M.
- *Cloud9am*, *Cloud3pm*: fraction of sky covered by clouds measured in *oktas* at 9 A.M. and 3 P.M. This measure is based in how many eighths of the sky are hidden by clouds. Consequently, the range of values are *integers* from 0 *oktas* from 8 *oktas*, both included.
- *RainToday*: a categorical variable that explains if the rainfall had exceeded 1 mm.

The response variable we have to predict is *RainTomorrow*, which explains if the following day the rainfall had exceeded 1 mm. Some of these variables are redundant or have strong correlation among them, so we need to remove or modify some in the next section.

3 Methodology

Ordinary difference

- $d = 1$

$$y'_t = y_t - y_{t-1}$$

- $d = 2$

$$y''_t = y'_t - y'_{t-1} = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) = y_t - 2 \cdot y_{t-1} + y_{t-2}$$

Seasonal difference, $M = 12, 7, 4 \ldots$

- $D = 1$

$$y'_t = y_t - y_{t-M}$$

- $D = 2$

$$y''_t = y'_t - y'_{t-M} = (y_t - y_{t-M}) - (y_{t-M} - y_{t-2M}) = y_t - 2 \cdot y_{t-M} + y_{t-2M}$$

Mixed differences

- $d = 1, D = 1$

$$\begin{aligned} y'_t &= y_t - y_{t-1}, \\ y''_t &= y'_t - y'_{t-M} = (y_t - y_{t-1}) - (y_{t-M} - y_{t-M-1}) \end{aligned}$$

The Dataset is formed by 142.000 samples, but many have missing values. After analyzing that all variables are missing sometimes (i.e. we don't have any special variable which produces the miss), removing the samples with missing values was decided. There are better options (for instance, substitution by the mode or the mean), but it is the simplest one. Once done this, 56.000 samples remain.

In contrast with other kind of Machine Learning assignments, now we are interested in analyzing certain prediction models and how feature selection affects them. For this reason, an exhaustive Grid Search is not going to be carried out, we are only interested in doing *honest* models. Besides, we can only use the train set in the *Feature Filter* and the *Wrapper*. That is why the Dataset has been split into a permanent train set and a permanent test set that has been used for all the experiments. In fact, the test set is a 30% of the Dataset; in other words we have almost 17.000 samples to test.

Besides, there is a strong unbalance among classes: almost in the 80% of the days it does not rain. It can be a problem that can be solved with many techniques, such as *undersampling* (removing randomly samples of the majoritarian class), *oversampling* introducing repetitions of elements of the minoritarian class or creating new synthetic samples (for instance, with SMOTE). We are going to see with the variables of the univariate filter whether the classifiers and metaclassifiers improves their predictions by applying *undersampling* to the training set[1].

Three tables are going to be shown: the first one for the accuracy, the second one for the F-Score and the last one for the execution times (including train and test time). The first table is to measure the results obtained in general, the second one is to see whether the unbalance has affected the classifier and the execution times is to see the efficiency of the algorithms purposed.

3.1 Features discarded and modified

Furthermore, some features have been discarded of modified, even before the first classification. In first place, the **wind direction** of *WindGustDir*, *WindDir9am* and *WindDir3pm* are nominal attributes, but we would improve this if we could tell the algorithm that *N* and *NNE* are nearer than *N* and *S*. For this, the variable was transformed to radians, but with this new transformation 0 and 2π are the more distant values, but they are the same[2]. To avoid that, each variable has been divided into two new attributes: sine and cosine.

Secondly, the date has been transformed to a categorical variable which can be more useful: the season. Besides, the variable *RainToday* has been removed because of redundancy: the variable is completely explained with *Rainfall*.

Finally, some variables with strong linear dependency to the rest of the Dataset have been removed to avoid convergence problems in logistic regression. In particular, we are talking about *Temp9am*, *Temp3pm*, *TempMin*, *WindDir3pmSin*, *Pressure3pm* and *WindDir3pmSin*.

3.2 Classifiers, metaclassifiers and feature selection algorithms

In this assignment, three classifiers are going to be manipulated. Firstly, we have a **logistic regression** model, where the attributes are going to be standardized and it's going to iterate until reaching convergence. In second place, the bayesian classifier is a **TAN** model, and for this model we have to convert all the variables into discrete variables. In this case, the discretization has

[1]This is important, we only can modify the training set to try to change the behaviour of the predictors. The test set must reflect the reality.

[2]This fact opens a new branch in statistics: directional statistics. There are many distributions that fixes the problem, such as the *von Mises distribution*, but we are going to choose a more straightforward and *homemade* solution.

been made separating the values into 10 bins of the same length. Finally, a **linear discriminant analysis** has been carried out. In this case, we need to convert all variables to numeric, using the *Discretize* method provided by *Weka*.

Regarding filtering and wrappers, we have three cases (apart from the no-filtering case). For univariate filters, the **gain ratio** method has been used. The threshold has been set in 0.1, i.e., if an attribute has a ratio of less than 0.1 it will be discarded. For the multivariate filter, **CFS** algorithm has been used in this case. Lastly, the **Wrapper** method provided by *Weka* has been used. The metric provided to evaluate is accuracy.

Finally, three classifiers are going to be used. In this case, we are going to use only the univariate filter, to simplify the results and to focus on the interpretation. In first place, the metaclassifier **Bagging with Neural Networks** will be used. The results of one neural network (with one hidden layer with as neurons as the input layer) are going to be shown too in order to compare all the metrics. Since training the model takes a long time, two threads will be used for bagging. The second one is a **Random Forest** where each attribute importance is going to be shown. Each tree will remove two random attributes before classifying. Regard to the third one, a **Fusion Classifier** based on majority vote has been implemented. The classifiers used are kNN with 20 neighbors and euclidean distance, an SVM with a polynomial kernel of first degree, a RIPPER classifier, logistic regression and discriminant analysis. The individual results of classifiers are going to be shown to facilitate the interpretation of the improvement.

4 Results

For a start, we have to take into account three different feature selection types: when we have a raw dataset, a discretized one or the numeric version. Since the numeric one creates new attributes, this case is going to be analyzed in the Table 1. In contrast, the another two cases can be put on the Table 2. On the other hand, the information about accuracy, F-score and execution time has been put in Table 3, Table 4 and Table 5 respectively. Finally, the results obtained by the Metaclassifiers and some auxiliary Classifiers are in Table 6. Furthermore, to deal with the balance some results are shown after undersampling in Table 7 and a small comparison with the unbalance dataset is shown in Figure 1. In addition, some extra screenshots may be found in Appendix A.

```
    a     b   <-- classified as
12391   777 |     a = No
 1868  1890 |     b = Yes
```

(a) Unbalanced Logistic Regression.

```
    a     b   <-- classified as
12120  1048 |     a = No
 1658  2100 |     b = Yes
```

(b) Unbalanced TAN.

```
    a     b   <-- classified as
12512   656 |     a = No
 1859  1899 |     b = Yes
```

(c) Unbalanced Random Forest.

```
    a     b   <-- classified as
10479  2689 |     a = No
  847  2911 |     b = Yes
```

(d) Balanced Logistic Regression.

```
    a     b   <-- classified as
10366  2802 |     a = No
  863  2895 |     b = Yes
```

(e) Balanced TAN.

```
    a     b   <-- classified as
10408  2760 |     a = No
  792  2966 |     b = Yes
```

(f) Balanced Random Forest.

Figure 1: Confusion matrices in some classifiers before and after balancing the training set.

	Univariate	Multivariate	Logistic	Bayes
Season			x	
Location			x	x
Rainfall	x	x	x	
Sunshine	x	x	x	x
WindGustSpeed	x		x	x
WindSpeed9am			x	
Humidity9am	x		x	
Humidity3pm	x	x	x	x
Pressure9am	x	x	x	
Cloud3pm	x	x		
Cloud9am	x		x	
WindGustDirSin				x

Table 1: Attributes chosen when the classifier allows only categorical attributes or when it allows both.

	Univariate	Multivariate	Discriminant
Location=(4 Values)	x		
Location=(3 Values)			x
Rainfall	x	x	x
Sunshine	x	x	
WindGustSpeed	x		
Humidity9am	x		
Humidity3pm	x	x	
Pressure9am	x	x	
Cloud3pm	x	x	
Cloud9am	x		
WindGustDirSin			x

Table 2: Attributes chosen when the classifier allows only numerical attributes

	All	Univariate	Multivariate	Wrapper
Logistic	0,851	0,844	0,838	0,85
Bayesian	0,842	0,842	0,846	0,843
Discriminant	0,815	0,799	0,787	0,783

Table 3: Accuracies of classifiers.

	All	Univariate	Multivariate	Wrapper
Logistic	0,612	0,588	0,569	0,608
Bayesian	0,607	0,614	0,596	0,573
Discriminant	0,646	0,625	0,61	0,402

Table 4: F-scores of classifiers.

	All	Univariate	Multivariate	Wrapper
Logistic	2,25	0,41	0,26	1,36
Bayesian	0,67	0,09	0,12	0,09
Discriminant	0,78	0,1	0,18	0,05

Table 5: Execution times of classifiers (in seconds).

	Accuracy	F-Score	Time
NN	0,846	0,592	19,21
Bagging	0,848	0,572	61,5
RandomForest	0,855	0,612	17,3
kNN	0,845	0,56	40,5
SVM	0,844	0,574	9,76
RIPPER	0,845	0,594	18,08
Logistic	0,844	0,588	0,42
Discriminant	0,799	0,625	0,09
Vote	0,847	0,597	65,12

Table 6: Evaluation of Metaclassifiers and auxiliary Classifiers.

	Accuracy	F-Score	Time
Logistic	0,791	0,622	0,33
TAN	0,783	0,612	0,12
Discriminant	0,791	0,62	0,21
Bagging	0,777	0,619	41,99
RandomForest	0,79	0,625	3,26
Vote	0,792	0,623	25,53

Table 7: Results of Classifiers and Metaclassifiers after balancing.

5 Discussion

Firstly, a brief description of the results will be given. After that, each algorithm will be analyzed more deeply, observing how it works and its behaviour.

Regarding the univariate and multivariate filters, we can observe that in both cases (numeric and discrete) the multivariate filter is a subset of the univariate one, but this is a coincidence. We also can remark that the attributes *Rainfall*, *Sunshine*, *Humidity3pm* and *Pressure9am* and *Cloud3pm* has been selected in almost all the filters/wrappers. In contrast, the season, the wind parameters (except *WindGustSpeed*), *Evaporation* and *MaxTemp* has hardly ever been selected. With this information, we can figure out which are the relevance of some variables. Looking at the accuracy and F-Score we can observe that Discriminant Analysis worsen with filters or wrappers and TAN is slightly better with the univariate filter.

In terms of accuracy and F-Score, Logistic Regression with all variables reveals one of the best results, and Discriminant Analysis is the best classifier if we are looking for a good F-Score. However, Logistic Regression is the worst classifier regarding the time.

Now, let's see the behaviour that we can infer of each classifier.

5.1 Logistic Regression

Weka is predicting the class *No* as our positive class, so the β vector in this case indicates:

$$\mathbf{P}(C = \text{No}|\mathbf{x}) = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1+\ldots+\beta_n x_n)}},$$

being $\mathbf{x}$ the vector of values of our attributes and C the label of the class. The *odds* of $\mathbf{x}$ are defined as:

$$\textbf{Odds}(\mathbf{x}) = e^{\beta_0+\beta_1 x_1+\ldots+\beta_n x_n},$$

and the *odd ratio* of an attribute x_j that *Weka* provides us is the corresponding e_j^{β}.

Weka also provides us the *logit* form in the *coefficients* column shown in Figure 2. It returns the logarithm of the previous result, i.e. β_j. Being θ the threshold, when $\beta^T\mathbf{x}$ is bigger than a certain value it means that $\mathbf{P}(C = \text{No}|\mathbf{x}) > \theta$, so the prediction will be *No*. On the contrary, if $\mathbf{P}(C = \text{No}|\mathbf{x}) < \theta$ it means that the prediction will be *Yes*.

In other words, if $\beta_j >> 0$ it means that big values of x_j will contribute to make $\mathbf{P}(C = \text{No}|\mathbf{x})$ greater, so x_j is an indicator of a dry day. Consequently, if $\beta_j << 0$ it means that a big x_j is an indicator of a rainy day. It also applies to categorical values, since they are transformed to binary attributes.

So which are the attributes with a higher β_j? Regarding all the attributes the most important ones are Hobart, Townsville, Sale, MelbourneAirport and NorfolkIsland locations, the winter and *Sunshine*. On the contrary, the attributes with a lower β_j are CoffsHarbour, Williamtown, Brisbane, Cairns and Perth locations and *Cloud3pm*. While the first ones contribute to predict a dry day, the second ones contribute to predict a rainy day. A value too near to zero indicates that the attribute is irrelevant.

We can observe also that the coefficient of a certain attribute has a similar value with all variables, filters or wrappers (the only perceptible change is in the *intercept* coefficient, which is normal since we have a distinct number of variables and consequently we need different bias). This can give us an idea of the stability that this method provides, since we obtain similar coefficients when we are working with the same dataset.

5.2 Tree-Augmented Naive Bayes

In this predictor, a graph of dependencies without cycles is generated. As we can see in 4, all the attributes has only two arrows: one from the class and one from one attribute, except the root attribute, that only that only has the first one. The connections between two attributes are chosen to regard the mutual information conditioned to the class, i.e. the certainty that one variable gives about the another one once we know the class.

If we have n attributes, we have to choose $n-1$ edges for the graph. These edges are those that have more mutual information between the attributes without creating cycles. Once we have formed the undirected graph, one of the nodes are selected at random to be the root node.

What can we infer of this information? Mainly, the TAN model can help us to understand which variables are explained better among themselves. Nonetheless, we cannot say that two attributes have low mutual information only because the edge does not appear in the graph; it may be removed in order to avoid cycles.

Regarding only the attributes of the filters and the wrapper (that are supposed to be the most relevant), we can find some of the strongest relationships really useful: *Sunshine* is strongly related to the *Humidity*, *Clouds* and *Pressure*; the *Rainfall* is indeed related with *Humidity* and the *Location* has a strong link with *Humidity* and the *Wind*. These are just a few examples, and we can find much more information in the diagram with all the variables. For instance, it's curious how all *Wind* variables are connected to each other, existing one single link: the *Location*. On the other hand, there are some trivial relationships, such as the *Rainfall-Humidity* one or the *Clouds-Sunshine* one.

5.3 Linear Discriminant Analysis

In LDA, (also known as Fischer's Linear Discriminant), there are two main assumptions: $X|c_i \sim \mathcal{N}(\mu_i, \Sigma_i)$ (where $X|c$ is the distribution of samples of a certain class, μ_i is the mean vector and Σ_i the covariance matrix) and all covariance matrices are equal for all the classes.

In this case, we have two classes which are supposed to be split by a hyperplane defined as:

$$w^T(x - x_0) = 0,$$

where x is the variable (same dimension as a sample) and the another two vectors are determined by the samples. We can write it in another way to understand the boundary in terms of the values given by *Weka*:

$$0 = w^T(x - x_0) = w^T x - w^T x_0 = w^T x - \text{Threshold},$$

so w^T is the vector given by the *Weights*. In that case, we can predict that the class is *No* if and only if:

$$w^T x > \text{Threshold}.$$

In other case, the class predicted is *Yes*. With this, we think that we can obtain the same conclusions about which attributes contribute to predict *Yes* or *No* depending on the sign of the weight regarding Figure 3, but that is not true. We can observe that the weights of the variables changes a lot depending on the filter or wrapper we are using. That is because the defined hyperplane does not have to cut perpendicularly the density function, so there are infinite different planes we can choose to represent the same cut. Consequently, we cannot infer much information about this hyperplane easily.

5.4 Metaclassifiers

Starting with **Bagging** with Neural Networks, we can't appreciate a better performance in comparison with a single neural network. Why is happening this? Because Bagging is a method that uses several times the same classifier, bootstrapping all the samples[3]. We have thousands of samples, so our generated datasets are very similar and this method become worthless in our particular case.

Random Forest follows a similar principle to Bagging, but in this case it is usually applied to Classification Trees and two random variables will be removed for each individual classification. We can see that Random Forest shows the best performance of all the classifiers, exceeding the 85% of accuracy and the 0.6 of F-Score. We can also see in Figure 5 the attribute importance based on the average impurity decrease[4], and we can infer which are the most important attributes: the number of hours of sunlight, the amount of rainfall registered the previous day and the maximum wind speed. Since the metaclassifier is removing different attributes, it's not "fair" at all to compare this this case with the classic Classification Tree.

Finally, we have the **Majority Vote** classifier. That's probably the most transparent metaclassifier due to its simplicity. We can see that it improves the results of all the classifiers; it is obvious in terms of accuracy, and in terms of F-Score it can be overcome by Discriminant Analysis, but this classifier has the worst performance regarding accuracy. Since the same seed has been used in the five individual classifiers and in the metaclassifier, the results shown in Table 6 tells us exactly the behaviour of the Fusion Metaclassifier itself.

5.5 Unbalance problem

Finally, we can observe the problem of unbalance and try to solve it. As we can see in Table 7, the accuracy decreased significantly in all the classifiers and metaclassifiers, and it's something we can expect because we are training our model with a subset of the previous training set and the test set is unbalanced yet. A possibility could be that we had a problem of overfitting that could be solved removing samples, but we are not in this case. The most interesting result is the F-Score, that increases significantly (a 5% in some cases).

We can observe specifically what happened with Logistic Regression, TAN, Bagging and Random Forest (we have not gain much information about *Vote* and *Discriminant Analysis* classifiers). Regarding **Logistic Regression**, the coefficients remain more or less at the same values. If we look at the **TAN** classifier instead, we can see that the tree is exactly the same. Regard to the Random Forest metaclassifier, we can appreciate that the *Rainfall* parameter is less important, but anything else. Finally, the accuracy in the *Out-of-bag* information in **Bagging** has increased almost a 2%, and it makes sense since the test set is now significantly different from the test set (referring to the proportion of classes, indeed).

We can think that *undersampling* has not changed much more the results, but if we check the confusion matrices shown in Figure 1 we can see that now the F-Score is low mainly because of the misclassification of the class *No* (in contrast with the previous case, where the main misclassification proportion was in the class *Yes*). Consequently, in this model a much bigger proportion of samples has been classified as *Yes*, and it may be interesting depending on what we are searching.

[3] That is, we obtain another dataset of the same size picking samples of the original one with repetitions.

[4] Probably this *impurity* is based in the Gini Index.

6 Conclusion

In this assignment, a weather prediction database has been handled in order to predict whether it was going to rain the following day to the sample. Filters and wrappers have been used to infer which were the most important variables in the dataset (mainly *Rainfall*, *Sunshine*, *Humidity* and *Pressure*) and multiple classifiers and metaclassifiers have exhibited good predictions. One of the most relevant problems we have to take into account with this dataset is the unbalance, that has caused a certain bias on the predictions and reduced significantly the F-Score.

First of all, three probabilistic classifiers have been managed in this assignment: Logistic Regression, TAN and Discriminant Analysis. The most graphical model is obtained with TAN: we can see a tree with the most important dependencies between variables, which allows us to learn how the dataset behaves. *Weka* is relatively transparent with the coefficients obtained by the other two classifiers, but we cannot interpret the results as easy as before. Logistic Regression allows us to induce which variables contribute more to predict each class, but anything else. On the contrary, Discriminant Analysis is completely opaque regarding the interpretability: we cannot infer any easy information from the hyperplane obtained.

After that, various metaclassifiers have been trained with a filtered training set: Bagging with Neural Networks, Fusion based on Majority Vote and Random Forest. Generally, all the metaclassifiers have provided better results than the previous classifiers we have mentioned and than the ones that make up our metaclassifiers. Bagging and Vote are more opaque; we could observe the out-of-bag estimates or the metrics of the classifiers that build the metaclassifiers, but we cannot obtain much newer information. Random Forest tells us what are the relevance of each variable based on the average impurity decrease, so we can extract some new information about our model.

Finally, the unbalance problem has been partially solved by the *undersampling* technique, i.e. removing random samples of the majoritarian class to force a balance. When it has been applied, the minoritarian class (*Yes*) had more chances to be chosen and the F-Score has been slightly improved to the detriment of the accuracy.

References

[1] C. Bielza Lozoya and P. Larrañaga Mújica. *Data-driven computational neuroscience : machine learning and statistical models*. Cambridge University Press, Cambridge, 2020.

[2] A. G. B. of Meteorology. Rain in australia. 12 2017. Online, accessed 01-Nov-2021. Link: http://www.bom.gov.au/climate/data.

[3] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

A Explanatory images

This little appendix has been created to insert the images of the classifiers behaviour with a comfortable size, allowing the reader an easier understanding.

Variable	Class No
Season=winter	0.2402
Season=spring	-0.034
Season=summer	-0.1022
Season=autumn	-0.106
Location=Cobar	0.1398
Location=CoffsHarbour	-0.2822
Location=Moree	-0.0499
Location=NorfolkIsland	0.2413
Location=Sydney	-0.2095
Location=SydneyAirport	0.0756
Location=WaggaWagga	-0.1707
Location=Williamtown	-0.2863
Location=Canberra	0.0461
Location=Sale	0.5176
Location=MelbourneAirport	0.4932
Location=Melbourne	0.21
Location=Mildura	0.1327
Location=Portland	-0.091
Location=Watsonia	0.2163
Location=Brisbane	-0.6464
Location=Cairns	-0.2669
Location=Townsville	0.4969
Location=MountGambier	-0.0138
Location=Nuriootpa	-0.0691
Location=Woomera	0.0791
Location=PerthAirport	-0.354
Location=Perth	-0.5955
Location=Hobart	0.5496
Location=AliceSprings	0.0628
Location=Darwin	0.0654
MaxTemp	-0.0408
Rainfall	-0.0208
Evaporation	0.0216
Sunshine	0.1562
WindGustSpeed	-0.0591
WindSpeed9am	0.0128
WindSpeed3pm	0.0181
Humidity9am	-0.0067
Humidity3pm	-0.0516
Pressure9am	0.0509
Cloud9am	0.0202
Cloud3pm	-0.1238
WindGustDirCos	0.0608
WindGustDirSin	-0.0347
WindDir3pmCos	-0.0716
WindDir9amCos	-0.0366
WindDir9amSin	-0.2865
Intercept	-44.7772

(a) All variables.

Variable	Class No
Season=winter	0.1809
Season=spring	-0.0578
Season=summer	-0.0554
Season=autumn	-0.0688
Location=Cobar	0.0405
Location=CoffsHarbour	-0.3302
Location=Moree	-0.2196
Location=NorfolkIsland	0.2191
Location=Sydney	-0.2321
Location=SydneyAirport	0.0904
Location=WaggaWagga	-0.1631
Location=Williamtown	-0.3473
Location=Canberra	0.16
Location=Sale	0.6301
Location=MelbourneAirport	0.5509
Location=Melbourne	0.2529
Location=Mildura	0.1781
Location=Portland	0.0831
Location=Watsonia	0.2479
Location=Brisbane	-0.7017
Location=Cairns	-0.2825
Location=Townsville	0.4334
Location=MountGambier	0.0936
Location=Nuriootpa	0.0453
Location=Woomera	0.2019
Location=PerthAirport	-0.4016
Location=Perth	-0.6027
Location=Hobart	0.6368
Location=AliceSprings	0.0002
Location=Darwin	-0.1863
Rainfall	-0.0173
Sunshine	0.1439
WindGustSpeed	-0.0529
WindSpeed9am	0.0224
Humidity9am	-0.0031
Humidity3pm	-0.0482
Pressure9am	0.0603
Cloud3pm	-0.1306
Intercept	-55.4417

(b) Multivariate wrapper

Coefficients...

Variable	Class No
Rainfall	-0.0234
Sunshine	0.1462
WindGustSpeed	-0.0368
Humidity9am	-0.0027
Humidity3pm	-0.0477
Pressure9am	0.0613
Cloud9am	0.0371
Cloud3pm	-0.1241
Intercept	-57.0191

(c) Univariate Filter

Variable	Class No
Rainfall	-0.0264
Sunshine	0.1383
Humidity3pm	-0.0433
Pressure9am	0.088
Cloud3pm	-0.134
Intercept	-85.9105

(d) Multivariate Filter

Figure 2: Coefficients of the betas in Logistic Regression.

```
Threshold: 29.040382750390087

Weights:

Season=winter:   0.11016073944822871
Season=spring:   0.016579673966971818
Season=summer:  -0.051921587875299116
Season=autumn:  -0.07523638792915575
Location=Cobar:         -0.14315012582682735
Location=CoffsHarbour:  -0.07919187818315981
Location=Moree:         -0.12946437926981808
Location=NorfolkIsland:          0.3030650789225468
Location=Sydney:         0.0033124100439883195
Location=SydneyAirport:          0.09669748615992281
Location=WaggaWagga:    -0.19090706417899692
Location=Williamtown:   -0.08982826100881422
Location=Canberra:      -0.002366037988534884
Location=Sale:   0.2883260624990383
Location=MelbourneAirport:       0.23900779720991555
Location=Melbourne:      0.1418841427595052
Location=Mildura:       -0.16955781325132083
Location=Portland:      -0.11761396487099506
Location=Watsonia:       0.12239837916007569
Location=Brisbane:      -0.18080501482911943
Location=Cairns:         0.005826242479521354
Location=Townsville:     0.3793581833967101
Location=MountGambier:  -0.05671620313480038
Location=Nuriootpa:     -0.12953691643009618
Location=Woomera:       -0.19172789340041702
Location=PerthAirport:  -0.11083590415533048
Location=Perth:         -0.21181554194477228
Location=Hobart:         0.389045424529805
Location=AliceSprings:  -0.2673855892571009
Location=Darwin:         0.10193856219090049
MaxTemp:        -0.023180331265802774
Rainfall:       -0.019303657071070863
Evaporation:     0.008567124873815809
Sunshine:        0.12396584473011629
WindGustSpeed:  -0.03848525426164349
WindSpeed9am:    0.005645706206263158
WindSpeed3pm:    0.020894855939882172
Humidity9am:    -2.23548427905305E-5
Humidity3pm:    -0.03413173785113554
Pressure9am:     0.031267938008533024
Cloud9am:        0.03514800480270891
Cloud3pm:       -0.02129341674719234
WindGustDirCos:          0.007807910967558498
WindGustDirSin:          0.007107591917717302
WindDir3pmCos:  -0.008323168117921489
WindDir9amCos:  -0.058009295820236485
WindDir9amSin:  -0.15020908492861554
```

(a) All variables.

```
Threshold: 242.21391642236762

Weights:

Rainfall:       -0.1413806032893283
Sunshine:        0.8458748041511317
WindGustSpeed:  -0.14414220471250158
Humidity9am:     0.012134525506176474
Humidity3pm:    -0.17416193369967792
Pressure9am:     0.24776485615846097
Cloud9am:        0.3228171489590834
Cloud3pm:       -0.21829430608214365
```

(b) Univariate Filter

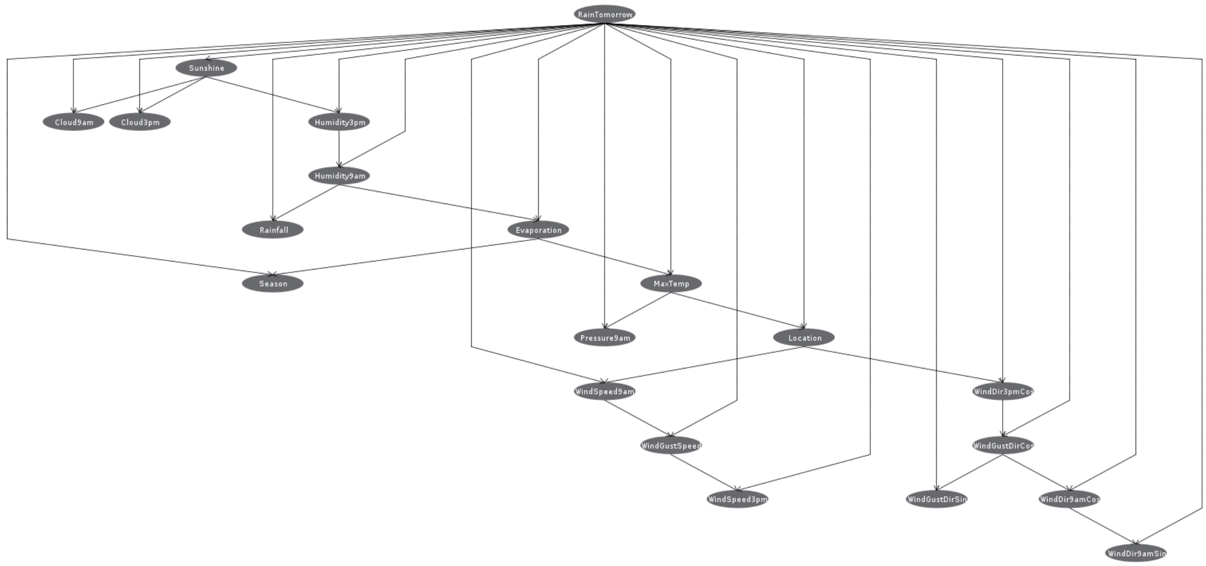
```
Threshold: 399.6631613849737

Weights:

Rainfall:       -0.16404765000849614
Sunshine:        0.8550001629278853
Humidity3pm:    -0.16421775144158787
Pressure9am:     0.3985613466468508
Cloud3pm:       -0.23715917226727168
```

(c) Multivariate filter

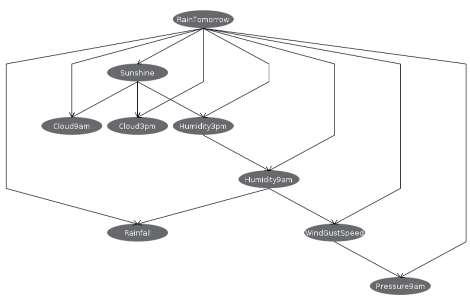
```
Threshold: -1.0321665197378225

Weights:

Location=Canberra:       0.5944296862868151
Location=Nuriootpa:      0.2398010572572602
Location=Darwin:        -0.12772394165273904
Rainfall:       -0.31310894464958333
WindGustDirSin:         -0.689056009703399
```

(d) Multivariate wrapper

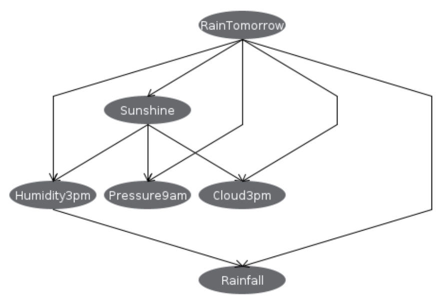
Figure 3: Coefficients of the hyperplane in Linear Discriminant Analysis.



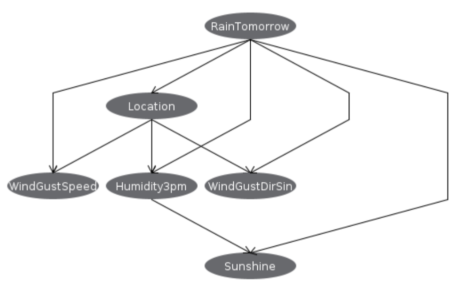
(a) All variables.



(b) Univariate Filter



(c) Multivariate filter



(d) Multivariate wrapper

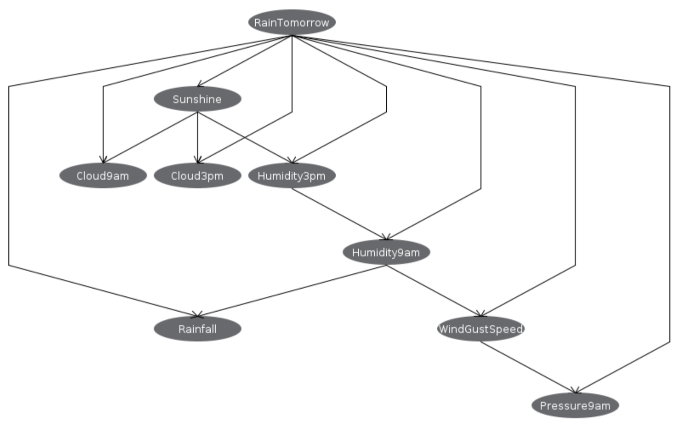
Figure 4: Graphs generated by TAN algorithm.

0.41	(45486)	Rainfall
0.39	(78217)	Sunshine
0.36	(56694)	WindGustSpeed
0.33	(64386)	Humidity9am
0.3	(51367)	Humidity3pm
0.27	(63411)	Pressure9am
0.26	(23474)	Cloud9am
0.25	(19058)	Cloud3pm

Figure 5: Attribute importance in Random Forest.

Variable	Class No
Rainfall	-0.0292
Sunshine	0.1563
WindGustSpeed	-0.0392
Humidity9am	-0.0036
Humidity3pm	-0.0445
Pressure9am	0.0686
Cloud9am	0.0435
Cloud3pm	-0.1195
Intercept	-65.8942

(a) Coefficients in Logistic Regression.



(b) Graph generated in TAN.

0.36	(47219)	Sunshine
0.35	(43356)	Humidity9am
0.34	(40480)	WindGustSpeed
0.33	(40435)	Humidity3pm
0.33	(26356)	Rainfall
0.33	(43263)	Pressure9am
0.26	(21284)	Cloud9am
0.25	(17563)	Cloud3pm

(c) Attribute importance in Random Forest.

```
*** Out-of-bag estimates ***

Correctly Classified Instances       13595               79.0959 %
Incorrectly Classified Instances      3593               20.9041 %
Kappa statistic                          0.5819
K&B Relative Info Score                 47.0415 %
K&B Information Score                 8085.4974 bits      0.4704 bits/instance
Class complexity | order 0           17188      bits      1      bits/instance
Class complexity | scheme            10904.6788 bits      0.6344 bits/instance
Complexity improvement     (Sf)       6283.3212 bits      0.3656 bits/instance
Mean absolute error                      0.2774
Root mean squared error                  0.3776
Relative absolute error                 55.488  %
Root relative squared error             75.5269 %
Total Number of Instances            17188
```

(d) Out-of-bag information in Bagging.

Figure 6: Results of some classifiers after balancing the classes.