Practical Application 2

Rafael Sojo García

rafael.sojo@alumnos.upm.es

November 2022

1. Introduction

Weather forecasting has always been an important problem for which there is not an easy task to find a solution, even nowadays with the best machine learning models. As it was proved in the first Practical Application [1] , it was hard to find real good results. This time, the same Dataset of Rain in Australia [2] will be used, but for probabilistic classifiers and metaclassifiers. The techniques used are:

Probabilistic
Linear Regression
Naïve Bayes
Tree Augmented Naïve-bayes (TAN)
Linear Discrminant Analysis (LDA)

Table 1. Probabilistic classifiers.

Metaclassifiers
Stacking
Bagging
AdaBoost
Random Forest
Logistic Model Trees (LMT)
Vote

Table2. Metaclassifiers.

As in the first assignment, these methods have been implemented in conjunction with four different sets of features. These are:

- All features
- Features selected with an univariate feature subset selection filter
- Features selected with a multivariate feature subset selection filter
- Features selected with a multivariate feature subset selection wrapper

2. Problem Description

Now, a brief recapitulation of what was this Dataset about. There are around 150.000 instances of weather measurements from Australian weather stations from the last 10 years, and total of 23 columns including the target variable, "*RainTomorrow"*. Here there is an explanation of each feature:

- *Date*: date where the observation was made.
- *Location*: location of the weather station.
- *MinTemp, MaxTemp*: minimum and maximum temperatures in degrees.
- *Rainfall*: amount of rainfall recorded in mm.
- *Evaporation*: Class A pan evaporation in mm.
- *Sunshine*: hours of bright sunshine in the day.
- *WinGustDir*: direction of the strongest wind in the 24 hours to midnight.
- *WindGustSpeed*: speed of the strongest wind in the 24 hours to midnight.
- *WindDir9pm, WinDir3pm*: direction of the wind at 9am and 3pm.
- *WindSpeed9am, WindSpeed3pm*: wind speed in km/h 10 min prior to 9am and 3pm.
- *Humidity9am, Humidity3pm*: humidity percent at 9am and 3pm.
- *Pressure9am, Pressure3pm*: atmospheric pressure at sea level at 9am and 3pm.
- *Cloud9am, Cloud3pm*: fraction of the sky obscured by cloud at 9am and 3pm.
- *Temp9am, Temp3pm*: temperature in degrees at 3pm.

- *RainToday*: yes if precipitation in the 24 hours to 9am exceeds 1mm, no otherwise.
- *RainTomorrow*: target variable.

3. Methodology

In the first Practical Application, it was explained that one of the biggest issues with this Dataset was the imbalance of its classes, since 85% of the days in *RainTomorrow* are **NO.** It was also showed how **SMOTE** [3] could be used to create artificial data to solve this problem, and in fact, achieving great improvements with respect the original weights. This time, the same technique is used, however, the models have been implemented using Weka [4]. Thus, due to the fact that one of the requirements is to use honest methods, all the models have been tested using the default **10 folds cross validation** configuration, for which we don't have the freedom of performing this process in a custom way, so that the validation splits could be tested with the original weights as it was done in the previous application. Nevertheless, this is not the goal of the project, therefore, no more comments will be added regarding this matter.

On the other hand, the same initial pre-processing steps have been done using Python. These were:

- Deletion of missing values from *RainTomorrow, RainToday* and *Shunshine.*
- Label encoding of the categorical variables based on their frequency with respect the class **YES**
- Creation of a new feature called *Months* to consider the relevance of the frequently rainy months from the feature *Date* and deletion of this last.
- Deletion of outliers
- Filling of the remaining *Nan* values with the mean of its belonging feature.

In addition, since the correlation between features could be harmful for our models, the correlation matrix has been studied to delete the features with the highest correlations, see figure 1. After that, these were the conclusions:

- *MaxTemp* and *Temp3pm* have a correlation of the **98%,** having *Temp3pm* the highest correlation with respect the class variable.
- *MinTemp* and *Temp9am* have a correlation of the **90%,** having *MinTemp* the highest correlation with respect the class variable.
- *Pressure9am* and *Pressure3pm* have a correlation of the **98%,** having *Pressure9am* the highest correlation with respect the class variable.
- *RainToday* and *Rainfall* have a correlation of the **85%,** having *Rainfall* the highest correlation with respect the class variable.

For these reasons, *MaxTemp, Temp9am, Pressure3pm* and *RainToday* were directly deleted. However, *Temp3pm* and *MinTemp* still having a **72%** of correlation, which could be caused by the fact that in the rainy days their difference is reduced. A good approach could be to create a new variable to substitute these two for their difference, nevertheless, this action introduces a new correlation with *Humidiy3pm*, as it is shown in figure 2. Then, the final decision was to also delete *MinTemp.*

Temp3pm	MaxTemp	0.983117
MaxTemp	Temp3pm	0.983117
Pressure9am	Pressure3pm	0.964199
Pressure3pm	Pressure9am	0.964199
Temp9am	MinTemp	0.907344
MinTemp	Temp9am	0.907344
MaxTemp	Temp9am	0.875485
Temp9am	MaxTemp	0.875485
Temp3pm	Temp9am	0.852038
Temp9am	Temp3pm	0.852038
RainToday	Rainfall	0.850849
Rainfall	RainToday	0.850849
MinTemp	MaxTemp	0.738233
MaxTemp	MinTemp	0.738233
Temp3pm	MinTemp	0.716241
MinTemp	Temp3pm	0.716241

Figure 1. Highest initial correlations.

TempDiff	Humidity3pm	0.748616
Humidity3pm	TempDiff	0.748616
Cloud3pm	Sunshine	0.634803
Sunshine	Cloud3pm	0.634803
WindSpeed3pm	WindGustSpeed	0.630115
WindGustSpeed	WindSpeed3pm	0.630115
Humidity9am	Humidity3pm	0.615846
Humidity3pm	Humidity9am	0.615846
Cloud9am	Sunshine	0.606183
Sunshine	Cloud9am	0.606183
WindDir3pm	WindGustDir	0.605218
WindGustDir	WindDir3pm	0.605218

dtype: float64

Figure 2. Highest correlations after initial deletions.

After these treatments, we end up with a Dataset of around 55.000 instances, from which a 15% subsample has been selected, i.e., **16.754 rows and 18 columns**.

3.1. Models

Now, the configuration of each model of Weka is explained.

- **Logistic Regression** → Default configuration of **Logistic.**
- **Naïve Bayes** → Default configuration of **NaiveBayes.**
- **Tree Augmented Naïve-Bayes** → TAN is set as the search algorithm in **BayesNet**
- **Linear Discriminant Analysis** → Default configuration of **LDA**
- **Stacking** → **REPTree** is used as the metaclassifier for 5 different classifiers: kNN, RIPPER, Logistic Regression, TAN and LDA.
- **Vote** → Average of Probabilities is set as the combination rule for the same 5 classifiers as Stacking: kNN, RIPPER, Logistic Regression, TAN and LDA**.**
- **Bagging** → **REPTree** is set as the classifier, i.e., default configuration.
- **AdaBoost** → **REPTree** is set as the classifier with a weight threshold of 50 in **AdaBoostM1.**
- **Random Forest** → Default configuration of **RandomForest.**
- **Logistic Model Trees** → Default configuration of **LMT.**

3.2. Feature Subset Selection

On the other hand, the selected feature subset selection methods have been the following.

- **Univariate Filter** → Default configuration of **CorrelationAttributeEval**, for which a threshold of 0.15 has been used.
- **Multivariate Filter** → Default configuration of **CfsSubsetEval**, using BestFirst as search method with the direction parameter set to backward.
- **Wrapper** → F-measure is used as the evaluation measure for a 5 fold cross validation, for which the feature selection criteria was to appear in at least 4/5 folds, i.e., at least 80% of the times. Then, the selected variables are tested in a 10 fold cross validation training as usual.

3.3. Metrics

All the models have been evaluated using the Precision, Recall, F-Measure and ROC Area metrics. Likewise, the confusion matrix of each of the models is given.

4. Results.

In table 3, the selected variables for the univariate and multivariate filters. Likewise in table 4, but for the wrapper.

Univariate	Multivariate
Humidity3pm	Humidity3pm
Sunshine	Sunshine
Cloud3pm	Cloud3pm
Cloud9am	Cloud9am
Rainfall	Rainfall
WindDir3pm	WindDir3pm
WindGustSpeed	WindGustSpeed
Pressure9am	Pressure9am
Location	Location
Humidity9am	Humidity9am
WindGustDir	WindGustDir
WindDir9am	WindDir9am
Temp3pm	Temp3pm
Month	Month
WindSpeed9am	WindSpeed9am
Evaporation	Evaporation
WindSpeed3pm	WindSpeed3pm

Table 3. Univariate and Multivariate Variables.

Wrapper Logistic Regression	Wrapper Naïve Bayes	Wrapper TAN	Wrapper LDA	Wrapper Stacking	Wrapper Bagging	Wrapper AdaBoost	Wrapper Random Forest	Wrapper LMT	Wrapper Vote
Humidity3pm	Humidity3pm	Humidity3pm	Humidity3pm	Humidity3pm	Humidity3pm	Humidity3pm	Humidity3pm	Humidity3pm	Humidity3pm
Sunshine	Sunshine	Sunshine	Sunshine	Sunshine	Sunshine	Sunshine	Sunshine	Sunshine	Sunshine
Cloud3pm	Cloud3pm	Cloud3pm	Cloud3pm	Cloud3pm	Cloud3pm	Cloud3pm	Cloud3pm	Cloud3pm	Cloud3pm
Cloud9am	Cloud9am	Cloud9am	Cloud9am	Cloud9am	Cloud9am	Cloud9am	Cloud9am	Cloud9am	Cloud9am
Rainfall	Rainfall	Rainfall	Rainfall	Rainfall	Rainfall	Rainfall	Rainfall	Rainfall	Rainfall
WindDir3pm	WindDir3pm	WindDir3pm	WindDir3pm	WindDir3pm	WindDir3pm	WindDir3pm	WindDir3pm	WindDir3pm	WindDir3pm
WindGustSpeed	WindGustSpeed	WindGustSpeed	WindGustSpeed	WindGustSpeed	WindGustSpeed	WindGustSpeed	WindGustSpeed	WindGustSpeed	WindGustSpeed
Pressure9am	Pressure9am	Pressure9am	Pressure9am	Pressure9am	Pressure9am	Pressure9am	Pressure9am	Pressure9am	Pressure9am
Location	Location	Location	Location	Location	Location	Location	Location	Location	Location
Humidity9am	Humidity9am	Humidity9am	Humidity9am	Humidity9am	Humidity9am	Humidity9am	Humidity9am	Humidity9am	Humidity9am
WindGustDir	WindGustDir	WindGustDir	WindGustDir	WindGustDir	WindGustDir	WindGustDir	WindGustDir	WindGustDir	WindGustDir
WindDir9am	WindDir9am	WindDir9am	WindDir9am	WindDir9am	WindDir9am	WindDir9am	WindDir9am	WindDir9am	WindDir9am
Temp3pm	Temp3pm	Temp3pm	Temp3pm	Temp3pm	Temp3pm	Temp3pm	Temp3pm	Temp3pm	Temp3pm
Month	Month	Month	Month	Month	Month	Month	Month	Month	Month
WindSpeed9am	WindSpeed9am	WindSpeed9am	WindSpeed9am	WindSpeed9am	WindSpeed9am	WindSpeed9am	WindSpeed9am	WindSpeed9am	WindSpeed9am
Evaporation	Evaporation	Evaporation	Evaporation	Evaporation	Evaporation	Evaporation	Evaporation	Evaporation	Evaporation
WindSpeed3pm	WindSpeed3pm	WindSpeed3pm	WindSpeed3pm	WindSpeed3pm	WindSpeed3pm	WindSpeed3pm	WindSpeed3pm	WindSpeed3pm	WindSpeed3pm

Table 4. Wrapper Variables.

Finally, the results for all models.

All	Precision	Recall	F-measure	ROC Area
Logistic Regression	0.775	0.767	0.771	0.854
Naïve Bayes	0.740	0.768	0.754	0.831
TAN	0.780	0.711	0.744	0.839
LDA	0.775	0.762	0.768	0.854
Stacking	0.866	0.953	0.907	0.943
Bagging	0.849	0.902	0.875	0.943
AdaBoost	0.833	0.875	0.854	0.912
Random Forest	0.894	0.956	0.924	0.980
LMT	0.822	0.898	0.859	0.885
Vote	0.834	0.899	0.865	0.936

(a) All.

Univariate Subset	Precision	Recall	F-measure	ROC Area
Logistic Regression	0.773	0.762	0.768	0.852
Naïve Bayes	0.741	0.775	0.757	0.832
TAN	0.754	0.737	0.745	0.834
LDA	0.775	0.754	0.764	0.851
Stacking	0.824	0.891	0.856	0.904
Bagging	0.815	0.869	0.841	0.915
AdaBoost	0.801	0.834	0.817	0.865
Random Forest	0.846	0.912	0.878	0.949
LMT	0.804	0.859	0.830	0.864
Vote	0.808	0.859	0.833	0.908

(b) Univariate Filter.

Wrapper Subset	Precision	Recall	F-measure	ROC Area
Logistic Regression	0.776	0.768	0.772	0.852
Naïve Bayes	0.777	0.767	0.771	0.853
TAN	0.775	0.769	0.772	0.852
LDA	0.774	0.771	0.773	0.852
Stacking	0.866	0.953	0.907	0.943
Bagging	0.847	0.906	0.876	0.944
AdaBoost	0.839	0.875	0.856	0.915
Random Forest	0.904	0.946	0.925	0.979
LMT	0.823	0.893	0.856	0.885
Vote	0.838	0.895	0.866	0.934

(c) Wrapper.

Multivariate Subset	Precision	Recall	F-measure	ROC Area
Logistic Regression	0.766	0.763	0.765	0.846
Naïve Bayes	0.755	0.769	0.762	0.84
TAN	0.752	0.760	0.756	0.841
LDA	0.767	0.757	0.762	0.842
Stacking	0,79	0,849	0.818	0.876
Bagging	0.798	0.841	0.819	0.895
AdaBoost	0.775	0.788	0.781	0.843
Random Forest	0.813	0.865	0.838	0.910
LMT	0.784	0.830	0.807	0.856
Vote	0.795	0.834	0.814	0.890

(d) Multivariate Filter.

Figure 3. Results.

5. Discussion

First, let's begin analyzing the most frequently selected variables, these are *Humidity3pm, Pressure9am, Sunshine*, *RainFall* and *WindGustSpeed*, appearing in all the different subsets for the case of *Humidiy3pm* and *Pressure9am,* and almost all except for the **Random Forest** wrapper subset for the case of *Sunshine*, the **Naïve Baye**s wrapper subset for the case of *Rainfall*, and the multivariate subset for the case of *WindGustSpeed.*

If we look to the final correlation matrix, *Humidity3pm* and *Pressure9am* have both the highest correlations with respect the class variable, which explains why they were selected so often in first place. After these two, we have *Cloud9am* and *Cloud3pm*, however, both are very correlated with *Shushine*, therefore, it could introduce noise in some of the classifiers. Note that it occurs the same with *Temp3pm*.

Now, if we consider these facts we can realize why *Pressure9am, RainFall* and *WindGustSpeed* are the following considering their capability for explaining the class variable.

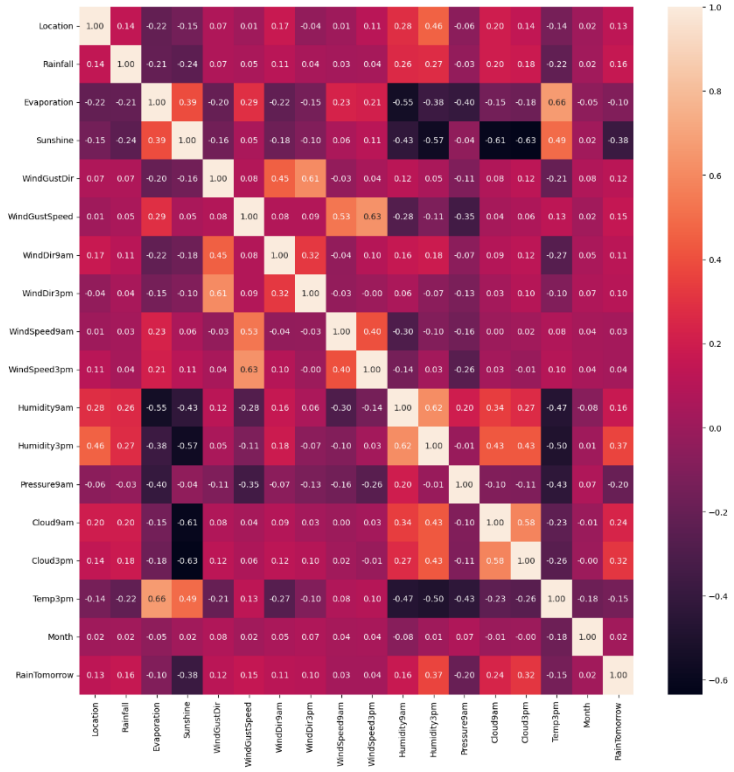


Figure 4. Final correlation matrix.

Let's continue analyzing the results for the **Logistic Regression**, in figure 5, the confusion matrixes.

```
    a       b      <-- classified as
6424.18 1841.82 |       a = 0
1926.59 6339.41 |       b = 1
```

(a) All.

```
    a       b      <-- classified as
6418.85 1847.15 |       a = 0
1965.19 6300.81 |       b = 1
```

(b) Univariate Filter.

```
    a       b      <-- classified as
6428.32 1837.68 |       a = 0
1913.72 6352.28 |       b = 1
```

(c) Wrapper.

```
    a       b      <-- classified as
6343.07 1922.93 |       a = 0
1955.54 6310.46 |       b = 1
```

(d) Multivariate Filter.

Figure 5. Logistic Regression confusion matrixes

Even though there is not much variation in the results for any of the different subsets, the models with less variables have a slightly worst performance, being the wrapper subset the best of the four. This could mean that most of the noise of the dataset was deleted during the pre-processing steps, losing part of the crucial information when trying to reduce more this list. Just a few variables were deleted for the wrapper subset, but as it was said previously, the is not much difference in the results.

```
Odds Ratios...
                        Class
Variable                   No
========================
Location               0.9852
Rainfall               0.9733
Sunshine               1.4061
WindGustDir              0.98
WindGustSpeed          0.6974
WindDir9am             1.0073
WindSpeed9am           1.1016
WindSpeed3pm           1.1049
Humidity3pm            0.6553
Pressure9am            1.2612
Cloud9am               1.1084
Cloud3pm               0.8703
Temp3pm                0.9215
```

```
Coefficients...
                        Class
Variable                   No
========================
Location              -0.0149
Rainfall              -0.0271
Sunshine               0.3408
WindGustDir           -0.0202
WindGustSpeed         -0.3604
WindDir9am             0.0073
WindSpeed9am           0.0968
WindSpeed3pm           0.0998
Humidity3pm           -0.4226
Pressure9am            0.2321
Cloud9am               0.1029
Cloud3pm              -0.1389
Temp3pm               -0.0817
Intercept              1.3314
```

Figure 6. Odds Ratio and Coefficients.

On the other hand, if we take a look to the Odds Ratios and Coefficients, all the variables are pretty close to each other, therefore all of them seems to be equally important, what supports the ideas of the previous paragraphs. In any case *Sunshine* and wind-related variables seem to have the strongest importance.

The fact that the best results were achieved with almost no deletion of variables is a constant except for the two **Naïve Bayes** models.

```
   a       b      <-- classified as
6032.25 2233.75 |       a = 0
1913.72 6352.28 |       b = 1
```

(a) All.

```
   a       b      <-- classified as
6028.11 2237.89 |       a = 0
1862.26 6403.74 |       b = 1
```

(b) Univariate Filter.

```
   a       b      <-- classified as
6442.53 1823.47 |       a = 0
1929.81 6336.19 |       b = 1
```

(c) Wrapper.

```
   a       b      <-- classified as
6201.57 2064.43 |       a = 0
1907.29 6358.71 |       b = 1
```

(d) Multivariate Filter.

Figure 7. Naïve Bayes confusion matrixes.

Looking at the results of figure 3 and 7, the best results were also achieved by the wrapper subset, which contains the smallest number of variables. This behavior has to do the implicit considerations that are made when using Naïve Bayes, like normality and independency. This independency allows us to use the following equality.

$$P(X_1 = x_1, ..., X_n = x_n | Y = y_j) \propto P(Y = y_j) \prod_{i=1}^{n} P(X_i = x_i | Y = y_j)$$

Figure 8. Naïve Bayes equality.

"Where $P(X_i = x_i | Y = y_j)$ is the probability of an instance of class y_j having the observed attribute value x_i" [5]. This basically means that we just need linear number of parameters, while for conditionally dependent models we tend to need an exponential number.

Continuing with bayes, **TAN** is also one of the models with less variables. For this model, we obtain the tree of figure 9.

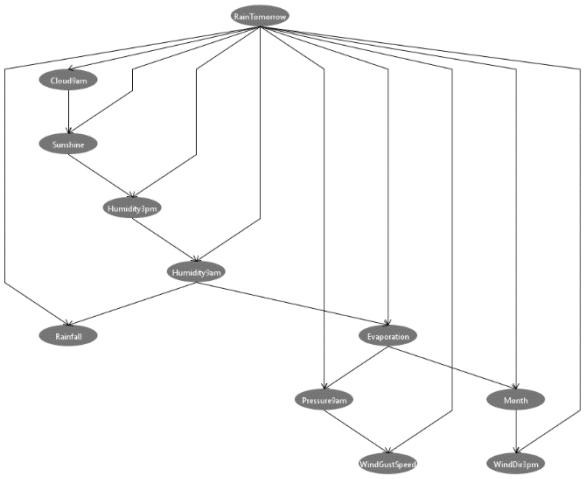


Figure 9. TAN Tree.

Here, the most important relationships between variables can be seen. For example, *Cloud9am* being the parent of *Sunshine* have sense if we look at figure 4, since from the selected variables it has one of highest correlations with the class, meanwhile sunshine, despite of having the highest correlation, has also a strong relationship with *Humidity3pm*. Likewise, *Humidity3pm* has less correlation with the class than *Sunshine*, but it has a strong correlation with *Humidity9am*, therefore, being its child.

From the last two models, we can obtain the same results as **Logistic Regression**, but with much smaller number of variables, what remarks the importance of making the right assumptions before initializing any model. Thus, reducing the noise.

Moving to **LDA**, it returned similar results to **Logistic Regression** in all the cases as it can be seen in figure 3. Thus, the best results are given for the wrapper subset. See the confusion matrixes at figure 10.

```
   a       b      <-- classified as
6433.06 1832.94 |       a = 0
1965.19 6300.81 |       b = 1
```

(a) All.

```
   a       b      <-- classified as
6451.41 1814.59 |       a = 0
2029.51 6236.49 |       b = 1
```

(b) Univariate Filter.

```
   a       b      <-- classified as
6408.78 1857.22 |       a = 0
1891.21 6374.79 |       b = 1
```

(c) Wrapper.

```
   a       b      <-- classified as
6369.71 1896.29 |       a = 0
2007    6259    |       b = 1
```

(d) Multivariate Filter.

Figure 10. LDA confusion matrixes.

It behaves similarly to PCA in that it attempts to account for class variability by reducing the data to a lower dimension.

Checking the means for each class we can have a better idea of the main differences. For example, *Sunshine* has a mean of 5.53 for the class 0 (i.e., NO), while 2.98 if 1, what fits with the idea that when rains there are less hours of sun, therefore, it is more likely to have rains the next day.

```
Sunshine: 5.53
```

(a)

```
Sunshine: 2.98
```

(b)

Figure 11. Means of *Sunshine*. NO | YES.

Having explained the probabilistic classifiers, it is time to discuss the results of the metaclassifiers, which have returned the best results. The first metaclassifier used was **Stacking**, for which 2 non-probabilistic and 3 probabilistic classifiers have been used. This model returned the second-best results with an F-measure of 0.907. Here, the confusion matrixes.

```
   a    b   <-- classified as
7137 1240 |    a = No
 396 7981 |    b = Yes
```

(a) All | Wrapper.

```
   a    b   <-- classified as
6786 1591 |    a = No
 912 7465 |    b = Yes
```

(b) Univariate.

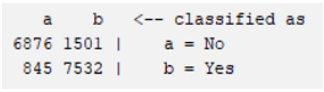
```
   a    b   <-- classified as
6484 1893 |    a = No
1264 7113 |    b = Yes
```

(c) Multivariate.

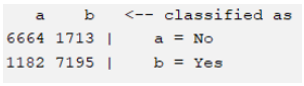
Figure 12. Confusion Matrixes of Stacking.

The first thing that we notice is that the wrapper has selected all the variables, and this is not surprising if we have in mind the fact that almost all the classifiers have selected almost all the variables. In fact, as it is shown in figure 12 the results are much worse when this number is reduced, what makes sense considering the exhaustive analysis that was made during the pre-preprocessing and the fact that different classifiers might find useful different combinations.

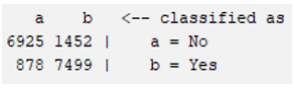
Alternatively, **Vote**, which also combines the outputs of the same different classifiers has returned a noticeably worst performance. See the confusion matrixes in figure 13.



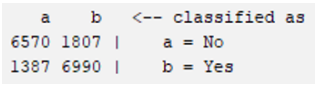
(a) All.



(b) Univariate Filter.



(c) Wrapper.

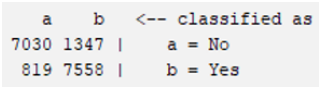


(d) Multivariate Filter.

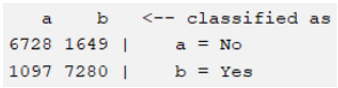
Figure 13. Confusion Matrixes of Vote.

The reason behind this, could be the fact that the results of each of the classifiers are combined just by averaging their probabilities, nothing else is considered as it does **Stacking.** Again, the best results were returned by the wrapper, which also contains almost all the variables except *Cloud9am.*

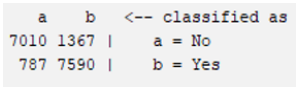
Bagging also performed well returning the third best results among all with REPTree, an unstable type of classifier. This type of techniques tends to perform better with bagging, since they usually take better advantage of the little changes in the inputs during the bootstrap. Here, the wrapper subset was also the best after deleting *WindDir3pm*, as it can be seen in figures 3 and 14.



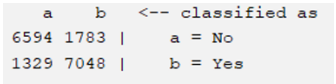
(a) All.



(b) Univariate Filter.



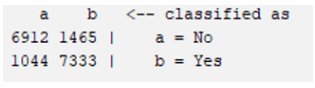
(c) Wrapper.



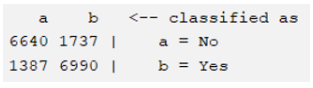
(d) Multivariate Filter.

Figure 14. Confusion Matrixes of Bagging.

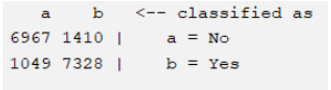
In the case of **AdaBoost**, it was the metaclassifier with the worst performance next to **LMT**, although both still giving better results than any of the tested classifiers. In both cases, the wrapper subsets were the best models, see figure 15 and 16. Here there is not much else to comment about.



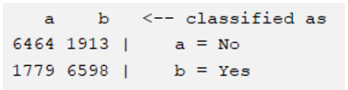
(a) All.



(b) Univariate Filter.

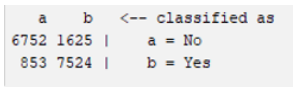


(c) Wrapper.

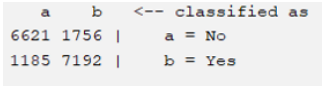


(d) Multivariate Filter.

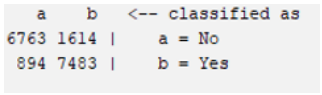
Figure 15. Confusion Matrixes of AdaBoost.



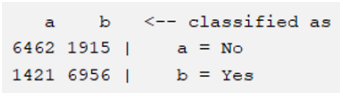
(a) All.



(b) Univariate Filter.



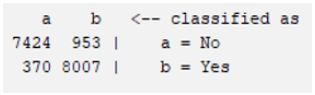
(c) Wrapper.



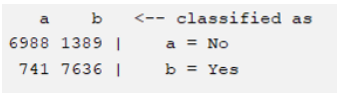
(d) Multivariate Filter.

Figure 16. Confusion Matrixes of LMT.

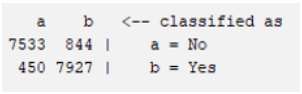
Finally, the best results by large were those achieved by Random Forest, more specifically for the wrapper subset, see figure 17. Here, the randomness of the vectors for the random trees had a clearly positive effect, returning an F-measure of 0.925. Unfortunately, this model does not return any visualizable tree to make a comment about.



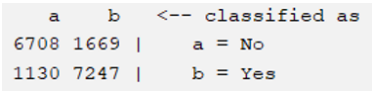
(a) All.



(b) Univariate Filter.



(c) Wrapper.



(d) Multivariate Filter.

Figure 17. Confusion Matrixes of Random Forest.

6. Conclusion

In conclusion, there are many things that we should consider before anything else. A good pre-processing as well as a good study of the correlation between variables is fundamental for the understanding of our data and how it could affect to the different models. However, it of course depends on which model do we pretend to use, take for instance the **Logistic Regression**, for which almost all the variables were equally important for predicting *RainTomorrow*, and any of the **Naïve Bayes** models, which achieved the same results with less than half of the variables.

On the other hand, some problems cannot be properly solved using just one model, and weather forecasting is one of them. The metaclassifiers demonstrated to be much better than the rest even in the worst cases, as it was the case for **AdaBoost** and **LMT**. Giving extraordinary results using Randomization in **Random Forest** or learning in which models to rely on when analyzed together in **Stacking.**

Finally, with respect to the different subsets, it is worth to mention that the **Wrapper** approach returned the best results for all the cases. However, it is a process that might not be the best for large Datasets as it can take to much time for selecting the right variables. In any case, it must be said that *Humidity3pm*, *Sunshine*, and wind-related variables had generally the most importance for the class variable.

References

[1] R. S. García, «Github,» [En línea]. Available: https://github.com/rafasj13/ML-Assignments/blob/master/ML-Assignment1/PracticalApplication1_RafaelSojo.pdf.

[2] «Kaggle,» [En línea]. Available: https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package.

[3] K. W. B. L. O. H. W. P. K. N. V. Chawla, «arxiv,» [En línea]. Available: https://arxiv.org/pdf/1106.1813.pdf.

[4] «waikato,» [En línea]. Available: https://www.cs.waikato.ac.nz/ml/weka/.

[5] J. M. Martinez-Otzeta, B. Sierra, E. Lazkano, M. Ardaiz y E. Jauregi Iztueta, «researchgate,» [En línea]. Available: https://www.researchgate.net/publication/220071669_Edited_Naive_Bayes.