Weather prediction in Australia

Jarod FOUCAUD

20 janvier 2022

Table des matières

Introduction

The dataset used in this project is the *Rain in Australia* dataset on **Kaggle**, you can find it here. This dataset contains about 10 years of daily weather observations from many locations across Australia. The aim is to predict next-day rain by training classification models on the target variable `RainTomorrow`. It means : did it rain the next day, `Yes` or `No` ? The value in this column is `Yes` if there was at least 1 mm of rain for that day. The rest of the dataset is a mixture of categorical and numerical values. A detailed explanation of the rest of the variables will be provided in 1.

The objective is to compare a few machine learning methods. Please note that this project was only created to help me try and improve my data science skills.

Source & Acknowledgements

— Observations were drawn from numerous weather stations. The daily observations are available from `http://www.bom.gov.au/climate/data`.
— An example of latest weather observations in Canberra : `http://www.bom.gov.au/climate/dwo/IDCJDW2801.latest.shtml`.
— Definitions adapted from `http://www.bom.gov.au/climate/dwo/IDCJDW0000.shtml`.
— Data source : `http://www.bom.gov.au/climate/dwo/` and `http://www.bom.gov.au/climate/data`,

1 Data

The dataset contains 145,460 rows and 23 columns :
— `Date`, the date of the observation,
— `Location`, the common name of the location of the weather station,
— `MinTemp`, the minimum temperature in degrees Celsius,
— `MaxTemp`, the maximum temperature in degrees Celsius,
— `Rainfall`, the amount of rainfall recorded for the day in mm,
— `Evaporation`, the so-called Class A pan evaporation (mm) in the 24 hours to 9am,
— `Sunshine`, the number of hours of bright sunshine in the day,
— `WindGustDir`, the direction of the strongest wind gust in the 24 hours to midnight (e.g. SE),
— `WindGustSpeed`, the speed (km/h) of the strongest wind gust in the 24 hours to midnight,
— `WindDir9am`, the direction of the wind at 9am (e.g. WNW),
— `WindDir3pm`, the direction of the wind at 3pm,
— `WindSpeed9am`, the wind speed (km/h) averaged over 10 minutes prior to 9am,
— `WindSpeed3pm`, the wind speed (km/h) averaged over 10 minutes prior to 3pm,
— `Humidity9am`, the humidity (%) at 9am,
— `Humidity3pm`, the humidity (%) at 3pm,
— `Pressure9am`, the atmospheric pressure (hPa) reduced to mean sea level at 9am,
— `Pressure3pm`, the atmospheric pressure (hPa) reduced to mean sea level at 3pm,
— `Cloud9am`, the fraction of sky obscured by clouds at 9am ; this is measured in oktas, which are a unit of eighths : it records how many eighths of the sky are obscured by clouds. A 0 measure indicates completely clear sky, whilst an 8 indicates that it is completely overcast,
— `Cloud3pm`, the fraction of sky obscured by clouds at 3pm (in oktas),
— `Temp9am`, the temperature (degrees Celsius) at 9am,
— `Temp3pm`, the temperature (degrees Celsius) at 3pm,
— `RainToday`, a boolean : `Yes` if precipitation (mm) in the 24 hours to 9am exceeds 1mm, otherwise `No`,
— `RainTomorrow`, a boolean : `Yes` if precipitation (mm) in the next-day 24 hours to 9am exceeds 1mm, otherwise `No`.

2 Data analysis

2.1 Cardinality of categorical features

First and foremost, we need to deal with categorical features. A basic idea to turn the numerical is to use either dummy variables, or one-hot encoding. Nevertheless, if we face a categorical feature with high cardinality, dummy variables or one-hot encoding will not be worth considering. Hence the need to check the cardinality of categorical features.

It turns out that the feature `Date` has 3,436 unique values. Indeed, this is not surprising at all since the dataset is supposed to give a report on 10 days of weather station data. As the column is encoded as an `object` type, we have to parse dates before extracting three new features : `Year`, `Month` and `Day`. The `Date` feature is no longer needed.

2.2 Missing values

As a picture is worth a thousand words, we choose to plot a heatmap on which a missing piece of data appears as a white line, while a present one is a blue line.

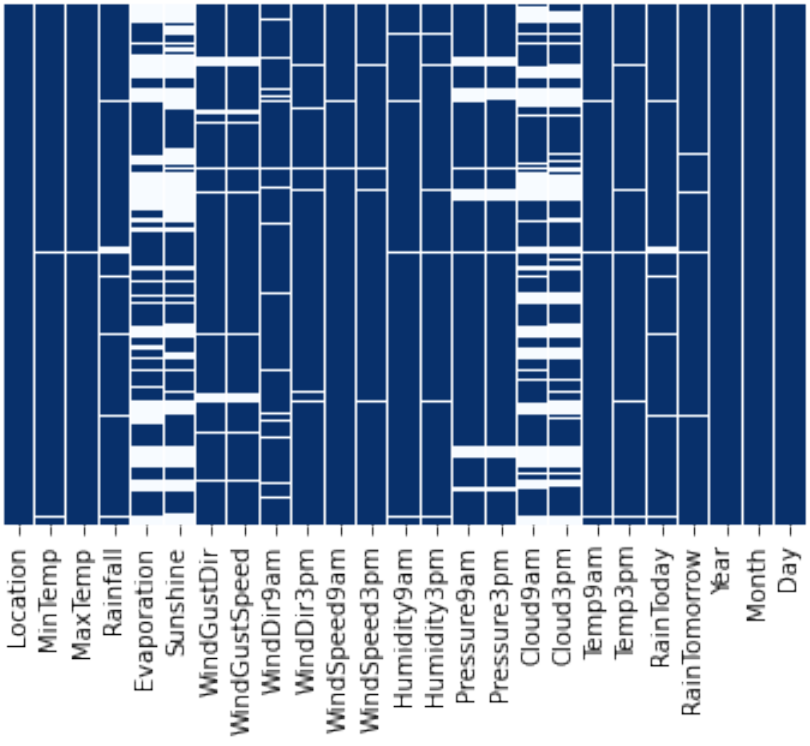

FIGURE 1 – Heatmap of missing data

2.2.1 Categorical features

The following table shows the number of null values for each categorical feature.

`Location`	0
`WindGustDir`	10,326
`WindDir9am`	10,566
`WindDir3pm`	4,228
`RainToday`	3,261
`RainTomorrow`	3,267

As we can see, all the locations are provided. Therefore we can immediately move on to the three features about wind : `WindGustDir`, `WindDir9am` and `WindDir3pm`. We may imagine that when there is a null value, it means that there was no wind during the day, hence the need to create a new category for this, which we call `no`.

Then, for the features `RainToday` and `RainTomorrow`, it would appear strange to fill missing values with any method before training on these guessed value. It is probably better to drop the rows with null values.

2.2.2 Numerical features

Before handling null values for numerical variables, we need to check for outliers. This is an important step because outliers can skew statistical measures on our dataset. As a result, the latter is not representative of the considered population any longer.

However, there are also drawbacks to removing outliers. Indeed, extreme or unusual meteorological events could have appeared during the period considered in our dataset, and our analysis of outliers will remove these values. Thus we could loose valuable information that might be helpful in order to predict `RainTomorrow` for those specific time periods.

As this dataset is quite small, it would be considered a bad idea to just remove rows with missing values. Therefore the benefit of removing outliers should outweigh the cost in that case. However, a more sophisticated analysis of Australian weather, with more available data, would stop here and handle the missing values using a different methodology. Most likely, this choice would lead to a better accuracy score.

The boxplot below illustrates the existence of outliers ; we handle them using the IQR (interquartile range) method.

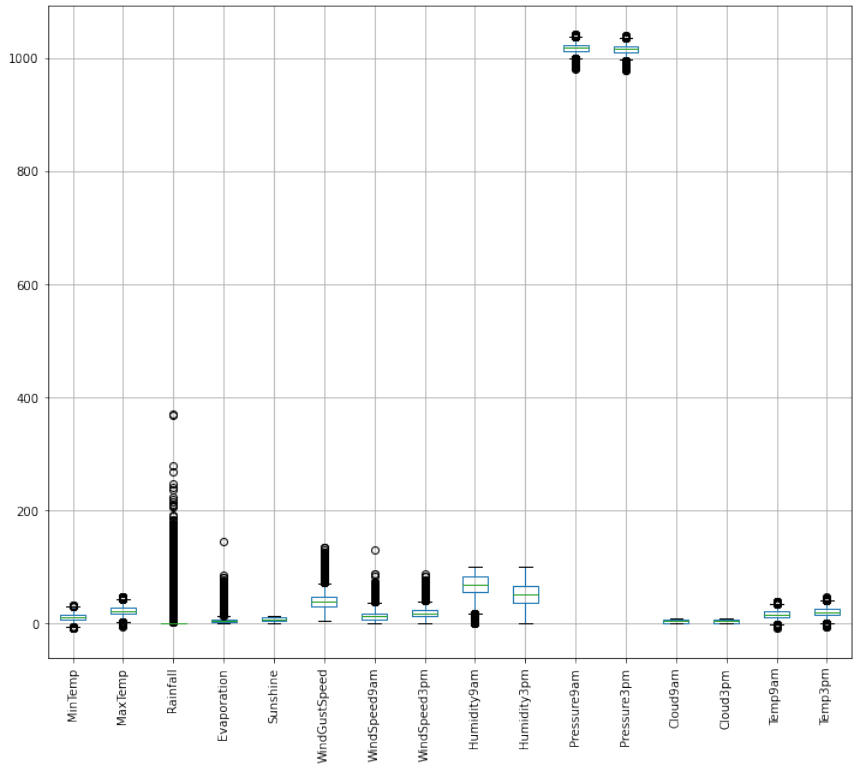

FIGURE 2 – Boxplot of numerical features

Now that we have removed outliers, we can handle missing values. We choose to fill them with the mean, since the fact that these values are missing probably does not have a proper significance : it is just a lack in the dataset.

2.3 Exploration of the dataset

As figure 3 shows, there are way more dry days than rainy ones — actually, it only rains about 22 % of the time in our data.

Then, the distribution of each numerical variable is displayed in figure 4. Their correlation matrix is provided in figure 5.

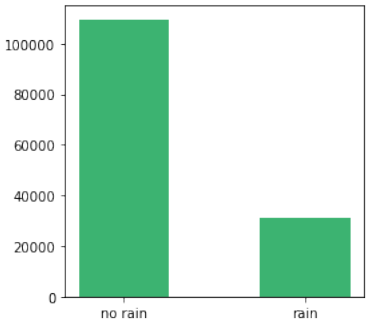

FIGURE 3 – Histogram counting dry and rainy days

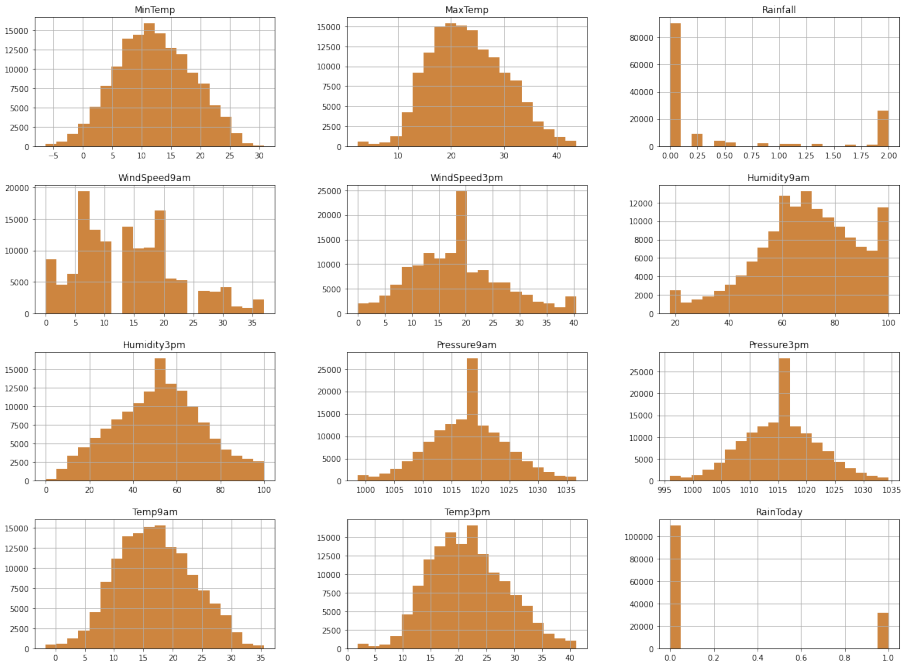

FIGURE 4 – Histograms of numerical features

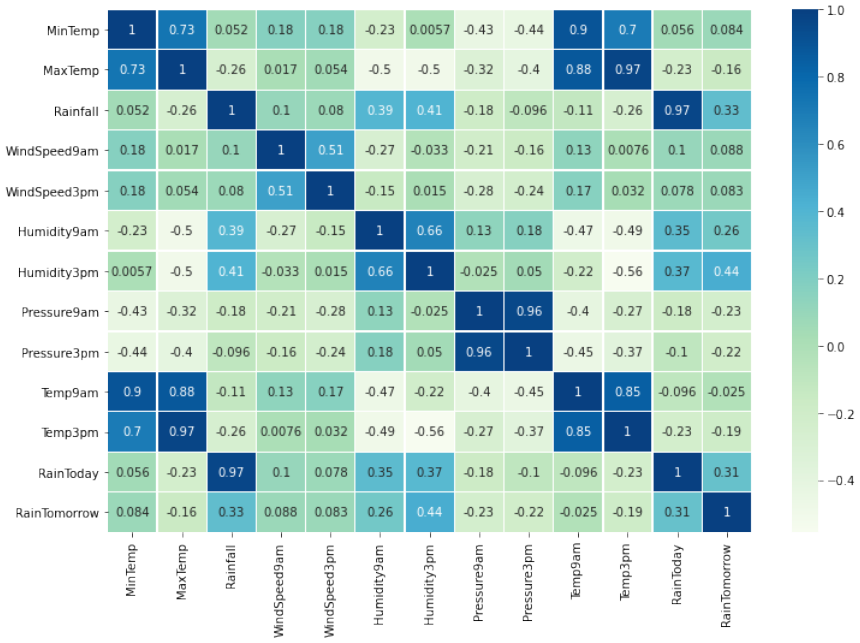

FIGURE 5 – Correlation matrix for numerical features

2.4 End of data preparation

A few steps remain before we can apply machine learning methods. First, we remove the `Location` feature since it will be hard for our algorithm to exploit. Then, we turn the columns `RainToday` and `RainTomorrow` to numerical by replacing `No` by `0` and `Yes` by `1`. After that, we need to create dummy variables for `WindGustDir`, `WindDir9am` and `WindDir3pm`. Finally, we split data into train and test dataset, with a ratio of 80 % for training data and 20 % to test.

3 Methods

3.1 Random forest

3.1.1 Null accuracy and first results

The very first idea is to build a naive model that predicts `No` for `RainTomorrow` every time. When this is done, we get a so-called *null accuracy* (or *baseline accuracy*) of 77.47 %. This value will be helpful to realize how accurate the random forest performed.

Then, we fit and predict with the random forest classifier. We arbitrarily choose to take 100 decision trees. The accuracy rate we have is 84.35 %, however the baseline accuracy is 77.47 %. The confusion matrix reveals that there are more false negatives than true negatives — i.e. a lot of type II errors. In the following subsection, we will try to make our model simpler through *feature selection*, as there are 60 features. Then we will try and improve the true negative rate.

3.1.2 Feature selection

In this subsection, we try to determine which features add value to the prediction. Feature selection (i.e. reducing the number of variables) will make our model simpler and reduce the chances of *overfitting*. We don't want to use variables that don't have much value when we train our model to predict out-of-sample data. The histogram below shows the importance of all the features for the random forest we used.

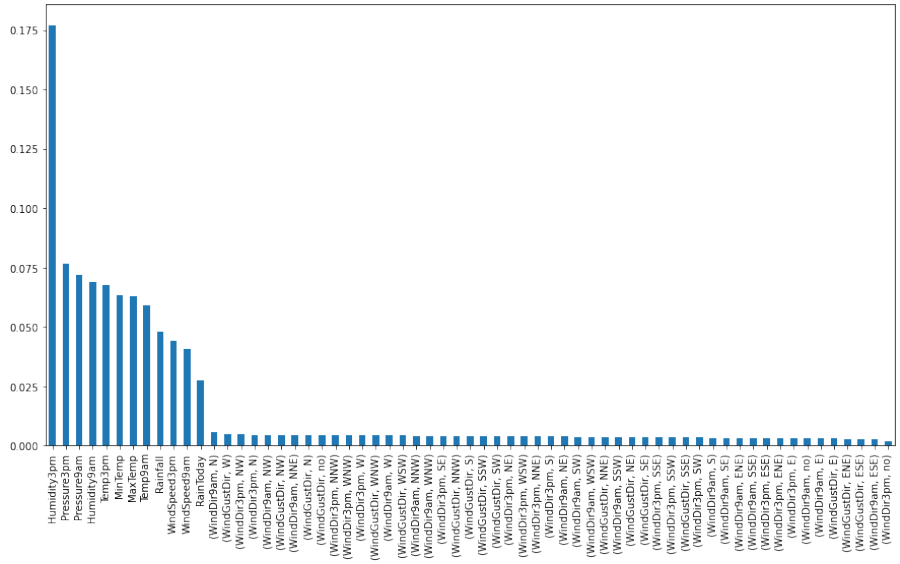

FIGURE 6 – Histogram of the feature importance

Remember our categorical variables about wind, `WindGustDir`, `WindDir9am` and `WindDir3pm`, were transformed into *dummy variables* to make the random forest algorithm perform better. Because they were transformed into *dummy variables*, then they would appear to rank low under the important features, as plotted above.

We may wonder whether the categorical (dummy) variables add much value to our model. So as to answer this question, we compute two accuracy rates :

— first, for the random forest using only the top 5 features (provided by figure 6) : we get 83.66 %,

— second, for the random forest using the top 5 features as well as the categorical variables : we get 83.97 %.

We can conclude that the categorical features do not add a significant value to the accuracy rate. Therefore we can remove the categorical (dummy) features.

3.1.3 How many features to keep for feature selection ?

Generally speaking, there is no right answer to this question. To help us choose a number of features to use, we visualise the relationship between the number of features used, and the accuracy rate, cf. figure 7.

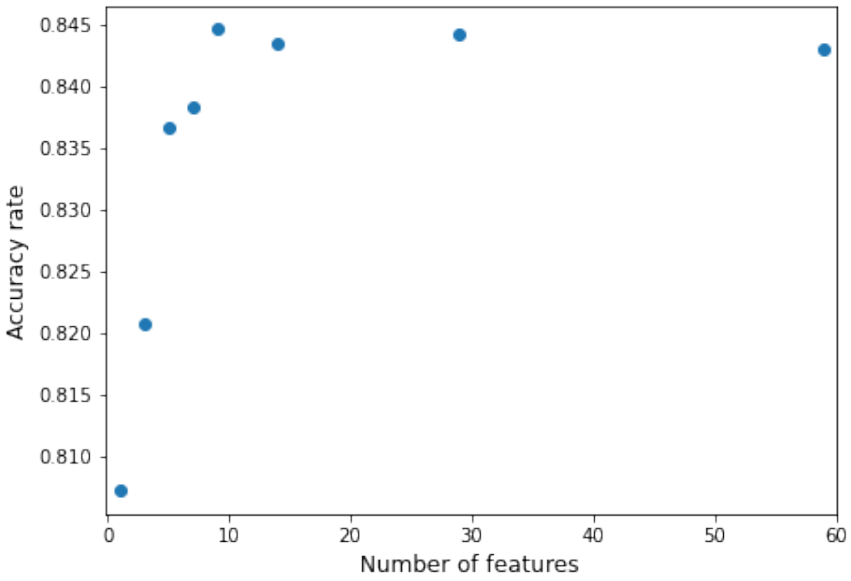

FIGURE 7 – Scatterplot of accuracy score in function of the number of features used

We can see that there is not much improvement to the accuracy rate when the number of features is 10 or more. To keep our model simple, we will use 10 features instead of 60. As we are about to see, the accuracy rate from 10 to 60 features differs by only 0.12 %.

When we use the top 10 features, we have an accuracy of 84.47 %. We would like to shed light on *sensitivity* (i.e. when it rains, how often our predictions are correct) and *specificity* (i.e. when it does not rain, how often our predictions are correct). Of course, as it does not rain about 78 % of the time, we expect the specificity rate to be higher. This is actually what we get :

Sensitivity	47.08 %
Specifity	95.34 %

To sum up : when it rained, we correctly predicted `Yes` 47.08 % of the time, whereas when it did not rain, we correctly predicted `No` 95.34 % of the time.

3.1.4 Threshold rate for the random forest

The random forest algorithm produces a probability score (the proportion of votes of the trees in the ensemble) in order to make classification. The following histogram shows probability scores for `Yes` and `No` (for `RainTomorrow`).

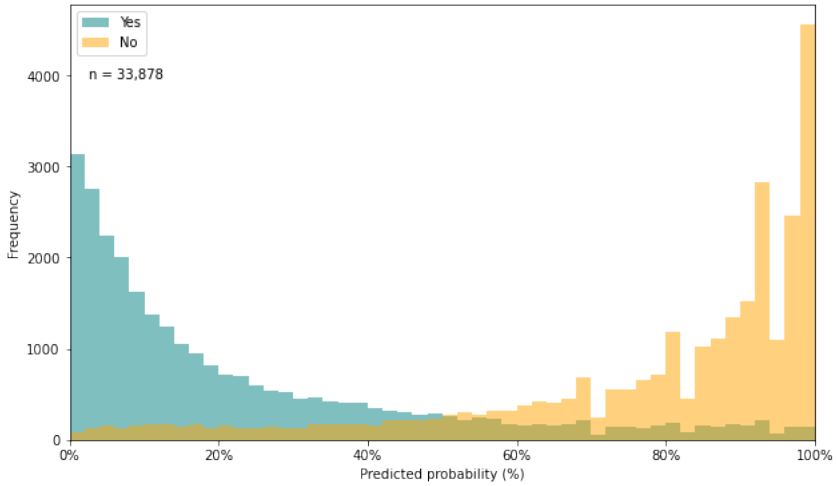

FIGURE 8 – Histogram of predicted probabilities

The random forest classication model uses 50 % as the threshold rate for classification. It means that if the prediction probability of `Yes` (for `RainTomorrow`) is higher than 50 %, it will predict `Yes` ; else, if the probability of `Yes` is lower than 50 %, then it will predict `No`.

Because of the binary relationship between `Yes` and `No`, the histogram has the highest frequency at 0 % for `Yes` and at 100 % for `No`.

We may wonder whether the cost of a false negative is greater than a false positive. In other words, we may think it is more prejudicial to predict `No` when it rained, than to predict `Yes` when it did not rain.

The 50 % threshold can be reduced in order to increase the sensitivity rate. However this will reduce the specificity rate because there is an inverse relationship between sensitivity and specificity.

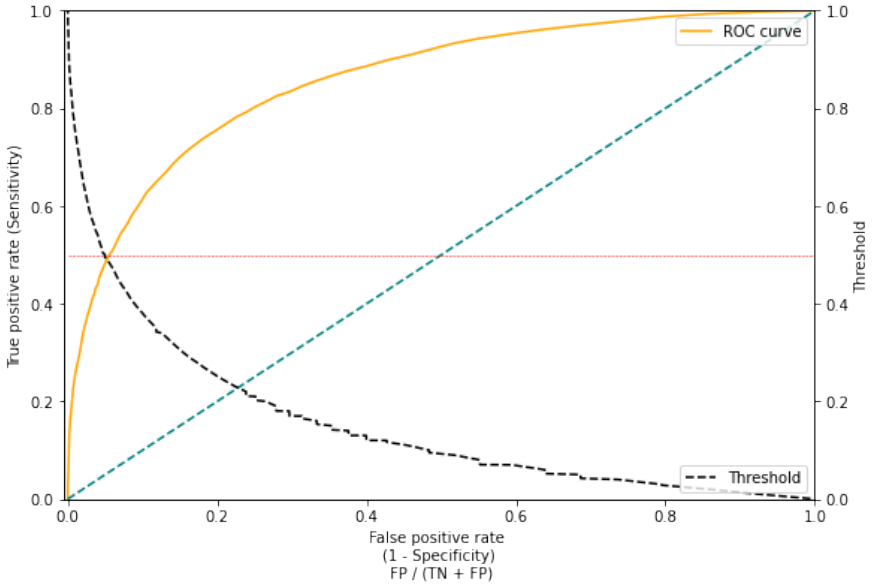

FIGURE 9 – ROC and threshold curves for the random forest classifier

The default threshold is 50 %, which had resulted in a low sensitivity rate of 47.08 % and a specificity rate of 95.34 %. Now we would like to change the threshold so that it provides a higher sensitivity rate at the cost of a lower specificity rate.

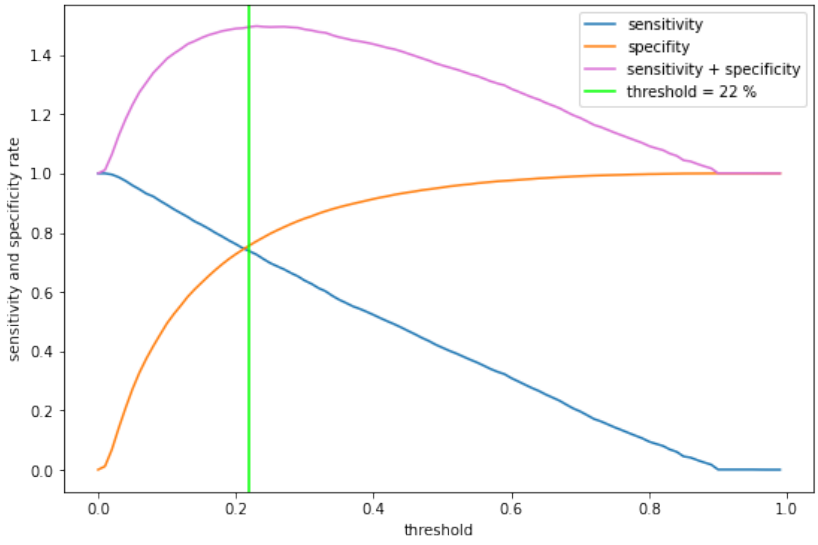

FIGURE 10 – Sensitivity and specificity rates in function of the threshold

As we can see on figure 10, we choose to reduce the threshold from 50 % to 22 %, in order to maximize the sum of sensitivity and specificity. Thus if, in our random forest, the probability of `Yes` is higher than 22 %, the model will predict `Yes` (for `RainTomorrow`). Conversely, if the probability of `Yes` is lower than 22 %, the model will predict `No`.

As a result, when we compute our selected-features and selected-threshold rate random forest model, we get the following scores :

Accuracy	77.67 %
Sensitivity	78.14 %
Specificity	77.53 %

As we can see, this makes our accuracy decrease, thus we can conclude our assumption stating that the cost of a false negative is greater than the one of a false positive is false.

However, we witness the phenomenon we expected : the sensitivity rate increases while the specificity rate dwindles : when it rained, we correctly predicted `Yes` 78.14 % of the time, while when it did not rain, we correctly predicted `No` 77.53 % of the time.

If the model produces a high sensitivity and specificity rate (which is what we would want to achieve), then the ROC curve will be stretched towards the top left hand corner of figure 9.

Next, we determine that the best threshold, i.e. the threshold which gives the highest accuracy score is about 49 %, therefore we should not change threshold since 50 % is a very good choice. This is the choice we will keep from now on.

The AUC (area under the ROC curve) provides an indication on how well the model performed in comparison to another model : we get 85.90 %.

3.2 Logistic regression

3.2.1 Using all the features

First, we train and predict with a logistic regression model using all the features of the dataset. The 10-fold cross validation method is used to calculate the accuracy score of the logistic regression model.

`Accuracy`	83.01 %

3.2.2 Feature selection, using recursive feature elimination (RFE)

As our final random forest model used 10 features, we will allow the same number of features for the logistic regression model. The recursive feature elimination gives the following selected features (by decreasing importance) :

1	`MinTemp`
2	`MaxTemp`
3	`Rainfall`
4	`WindSpeed9am`
5	`Humidity3pm`
6	`Pressure9am`
7	`Pressure3pm`
8	`Temp9am`
9	`Temp3pm`
10	`RainToday`

3.2.3 After feature selection

After that, we train and predict with our selected features-logistic regression model. The 10-fold cross validation method gives the following accuracy score. We can also get the sensitivity and specificity.

Accuracy	83.00 %
Sensitivity	41.20 %
Specificity	95.25 %

When it rained, we correctly predicted `Yes` (for `RainTomorrow`) 41.20 % of the time, whereas when it did not rain, we correctly predicted `No` 95.25 % of the time.

Although we noted it was not true, if we assume that the cost of a false positive is greater than the cost of a false negative, we may adjust the threshold rate to 22 % (cf. 3.1.4). We expect the same phenomenon as with the random forest, and the following scores corroborate this guess.

Accuracy	75.24 %
Sensitivity	73.71 %
Specificity	75.68 %

4 Results

The area under the ROC curve (AUC) is an indicator of which model has the strongest performance. As a reminder, the null accuracy score is 77.47 %.

	Random forest	Logistic regression
AUC score	85.90 %	82.28 %
Sensitivity (with 50 % threshold)	47.08 %	41.20 %
Specificity (with 50 % threshold)	95.34 %	95.25 %

The random forest model is the better performer as the AUC is 85.90 % vs. 82.28 % for the logistic regression model.

From comparing accuracy rates, the categorical features : `WindGustDir`, `WindDir9am` and `WindDir3pm`, offered little value. We saw the increase in the accuracy rates from having 1 feature to 63 and chose 10 features as that was the approximate point at which the increase in accuracy rate was very small. Having less features would simplify the model, reduce chances of overfitting, and provide better interpretability.

Both the random forest and the logistic regression had a low sensitivity rate which often incorrectly predicted that it won't rain the next day when it actually rained.

The ROC and threshold curves demonstrate the relationship between sensitivity and specificity at each threshold. Thinking that the cost of a false positive was greater than a false negative, we chose to reduce the threshold from 50 % to 22 %. This resulted in increasing the sensitivity rate for the random forest model, at the trade-off of decreasing the specificity rate. But as the accuracy rate also decreases, we determine that 50 % is a better threshold, and we choose to keep it.

The features used in the random forest and logistic regression differ. Features used in the final model were :

	Random forest	Logistic regression
1	`Humidity3pm`	`MinTemp`
2	`Pressure3pm`	`MaxTemp`
3	`Pressure9am`	`Rainfall`
4	`Humidity9am`	`WindSpeed9am`
5	`Temp3pm`	`Humidity3pm`
6	`MinTemp`	`Pressure9am`
7	`MaxTemp`	`Pressure3pm`
8	`Temp9am`	`Temp9am`
9	`Rainfall`	`Temp3pm`
10	`WindSpeed3pm`	`RainToday`