SECOND ASSIGNMENT
MACHINE LEARNING MODELS

Alejandro L. García Navarro
Olga Bonachera del Pozo
Universidad Carlos III de Madrid – Degree in Data Science and Engineering
Introduction to Data Science

INDEX

INTRODUCTION

On the one hand, in the very first assignment we saw that there is not a determined way of performing a good exploratory data analysis. What is more, we were taught some points to follow that could lead to a good analysis. Some of these points were generating questions about our data, and searching for answers by visualising, transforming, and modelling our data. Before starting, we had to know the type of each variable mainly for knowing their behaviour and the way we could relate one with the other.

On the other hand, for this second assignment, we also had to put into practice what was learned during the classes, which was machine learning methods and, to be more precise, supervised machine learning methods. Among these, we can find some like neural networks, naïve bayes, logistic regression, random forests...

Likewise, vital goals were set:

1. Using a supervised machine learning technique from those reviewed during the course, try to further analyze additional relationships among variables, which might influence survival (survived). This analysis complements the first assignment on exploratory data analysis, and it might help to support some of your previous conclusions, and also provide additional information which was difficult to extract manually.
2. You have to make a model for predicting the variable "survived" as a function of the rest of variables included in the data set, or a subset.

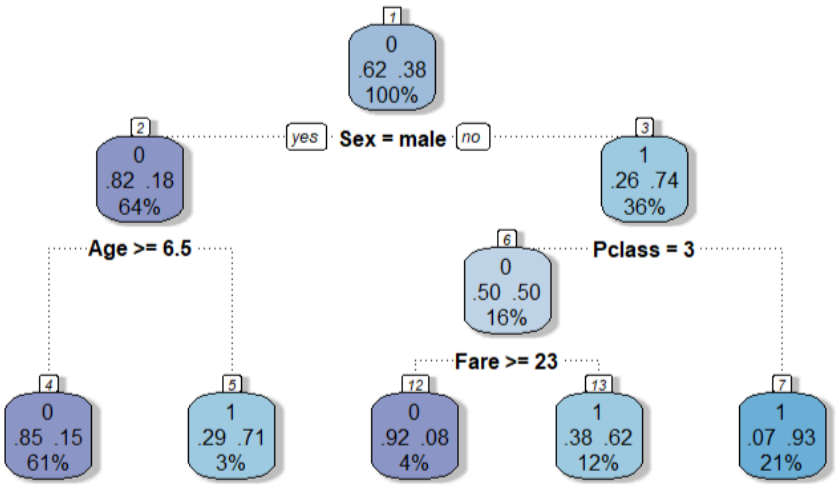
Before applying any machine learning algorithm, we have to make several tasks related to data, which could influence the over-fitting, under-fitting and validation process. Moreover, as the only machine learning techniques that will be used along the whole project are decision trees and random forests, these pre-processing steps are not necessary to be performed.

To begin with, for the first goal of the project, we used classification trees. Among its functionalities, the ability of predicting a result based on data and the simplistic results stands out. They are used when the dataset needs to be split into classes that belong to the response variable, which tend to be a solid yes or no.

In the second place, for the final part of the assignment, the creation of a model is required to be done. It is vital to take into account that in the search for the best model, the variable 'Survived' is going to be the pillar of this assignment. In order to find a useful prototype, several machine learning techniques must be performed so that we get several conclusions. From these conclusions we will select the model that has the highest accuracy, precision and specificity, as the higher the values, the better the performance of the model. In our case, we used both decision trees and random forests. Decision trees, as said before, allow for the rapid classification of new observations and can often result in a simple model which explains why the observations are either classified or predicted in a certain way. Likewise, random forests combine the output of multiple decision trees to reach a single result. To say it in simple words: random forests build multiple decision trees and merge them together to get a more accurate and stable prediction. For applying these techniques, several steps must be performed.

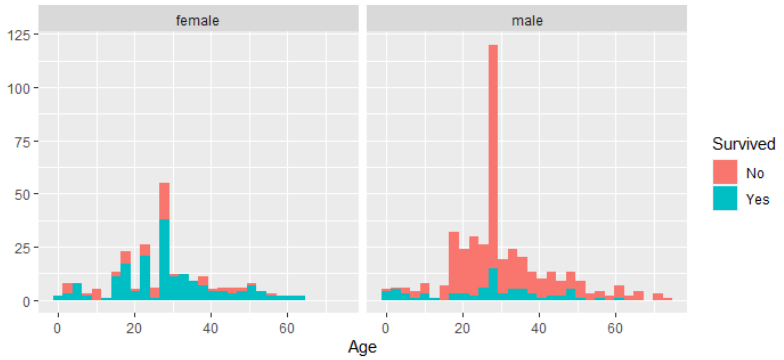
ITEM 1

In this part, the conclusions that were taken in the first assignment must be either confirmed or not. In the first place, by taking a look into our data set, we can assume that the variables Cabin and Ticket are not relevant for our purpose of predicting the chances of survival, so we will not take them into account and we could even delete them from our dataset, as we will expose in the Item 2. We are using a decision tree to further analyze the relevance of each variable with the survival one.

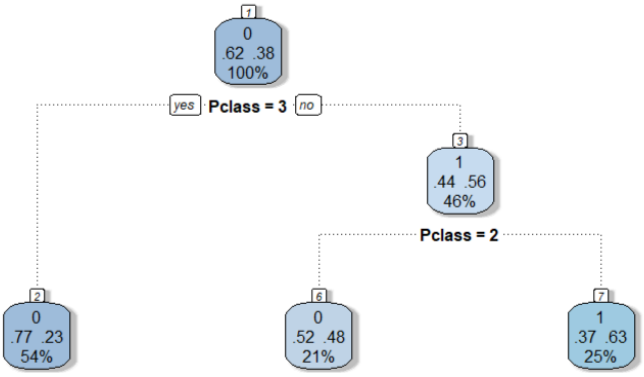


(Figure 1)

In the first node we get that 100% of our data reveals that the percentage of people not surviving is the highest (62%). Given that the first nodes into which the tree is divided are the most important ones, we know that the sex plays an important role in the chances of survival; we could even confirm that the variable sex is the most vital one, contrasting what we said in the first assignment. In fact we can see that there were a bigger number of men aboard the Titanic, but the percentage of them surviving is a lot lower (18%) than the women (74%). When the passenger is a man, the age is quite relevant given that being older than 6 years means that only a 15% survived. On the contrary, a very low number of men younger than 6 years did not survive as estimated in our first assignment. In addition, we get that there is the same percentage of women of third class surviving and not, while from the other classes, the 93% percentage of them survived. If any doubt arises, we might confirm it by looking at the graph from below.

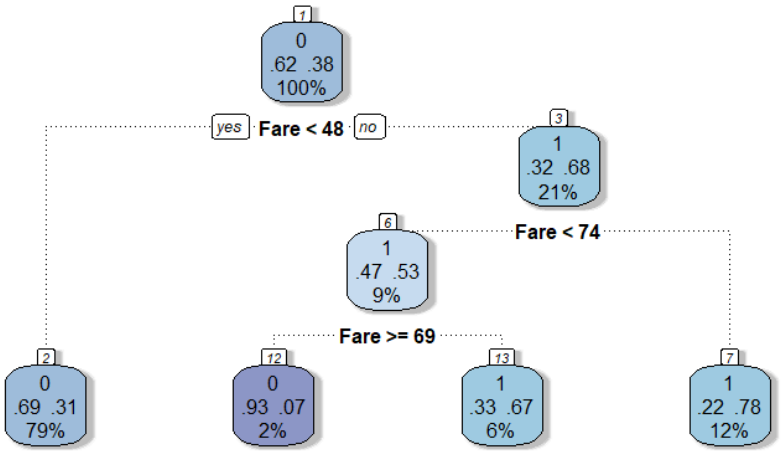


(Figure 2)



(Figure 3)

Even more, *Figure 3* confirms that third class has a lower percentage of survivors even though more than half of the passengers belonged to third class. This is useful for checking our previous assumptions from the first assignment, in which we confirmed the dependence between the chances of surviving and the class to which the passenger belongs.
Returning to *Figure 1,* it is notable that only a 4% of women travelling in third class paid a fare higher than 23 and 92% of them died. Our estimations from the first assignment are more or less wrong, as we were assuming that the higher the price, the higher the percentage of survivors.
As it is noticed, the variable SibSp, Parch and Embarked do not appear in the tree from *Figure 1*. This tree includes all the important variables for an optimal extraction of information that could lead to concise conclusiones. Hence, this means that those variables are not as relevant as we first may have thought.



(Figure 4)

Finally, we use the decision tree from *Figure 4* for the last conclusion of our first project in which we assumed that people with a higher fare were more likely to survive. As we can see in this tree, if the fare was lower than 48, which corresponds to the 79% of the passengers, the percentage of them surviving is quite low in comparison to the fare higher than 48 in which 68% of them survived. So our estimations were good.

ITEM 2

In this second item we are asked to build a model which predicts the variable “Survived” based on the other variables. Once the dataset has been graphically analyzed and the conclusions of the first assignment have been confirmed or not, the next step is to use a machine learning algorithm to create a model capable of representing the patterns in the training data and generalizing them to new observations.

Finding the best model is not easy, there are many algorithms, each with its own characteristics and with different parameters that must be adjusted. In our case, we will use two known techniques, decision trees and random forests. Decision trees are a decision support tool that uses a tree-like model of decisions and their possible consequences; likewise, random forests are a learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time.

We start by choosing an appropriate sampling method that, in this project, will be the k-fold cross-validation. This way of sampling is one of the most popular techniques to assess the accuracy of the model. Data is split into k equally sized subsets, also known as folds. This kind of validation is mainly used in applied machine learning to estimate the skill of a machine learning model on unseen data. For this specific sampling method, several steps must be performed.

1. Run the classification algorithm.
2. Use the function predict to apply the classification algorithm to the current data set.
3. Compute the confusion matrix. A confusion matrix in R is a table that will categorize the predictions against the actual values. It includes two dimensions, and among them one will indicate the predicted values and the other will represent the actual values.
4. Choose a performance measure for comparing models, which will compute error estimates. In our case, we choose three: accuracy (used to determine which model is best at identifying relationships and patterns between variables in a dataset based on the input), precision (indicator of a machine learning model's performance. It is the quality of a positive prediction made by the model. It also refers to the number of true positives divided by the total number of positive predictions), and specificity (the proportion of actual negatives, which got predicted as negative (or true negative). This implies that there will be another proportion of actual negative, which got predicted as positive and could be named as false positives. This proportion can also be called a false positive rate).
5. Choose how many folds we would like to perform. We will be splitting the data into 10 folds of data randomly selected, that will be used to estimate the error of our model.
6. Create a lapply function. This is applied for operations on list/objects and returns a list/object of the same length of the original set. This function is performed in base of the folds we created and a function. Inside this, we need to select the training and test set according to the current split and steps 1, 2, 3 and 4 from above.

Moreover, we need to do these steps twice, one for decision trees and another one for random forests. The next step is to select the hyperparameters that optimize the performance measure. Note that many models contain parameters that cannot be learned from the training data and therefore must be set by us. These are known as hyperparameters. The results of a model can depend to a great extent on the value that its hyperparameters take, however, it is not possible to know in advance which is the appropriate one. The most common way to find optimal values is by trying different possibilities. Hence, we select the best combination of hyperparameters by trying several combinations and comparing the accuracy, precision and specificity. We should make the average in precision, accuracy and specificity and select the best model.

It is important that, for this part, we delete the variables Ticket and Cabin. If we take a minute to observe them, we get to see that they are alphanumeric variables, which will make it arduous and impossible to perform the project if these are present. Even more, that is not the only reason why we delete them; we do this because they are not vital for analyzing the variable Survived. If we were to do so for another variable, maybe they could contribute something. Moreover, this is not the case. A more valid reasoning may be that these variables are composed of many values, very different from each other, impossible to classify.

1. DECISION TREES

1. K-FOLD CROSS-VALIDATION

For the case of decision trees, k-fold cross-validation requires the steps that were stated two pages before. At a very first instance, we get the accuracies from the table attached below.

Accuracy	Precision	Specificity
0.8130072	0.8127329	0.8294619

As observed in the results, if we only did the sampling method, we would get a pretty inaccurate model. Even though it has a rough 80% of accuracy and could be even considered a good model, we can get better results by looking for hyperparameters that allow us to have a better outcome.

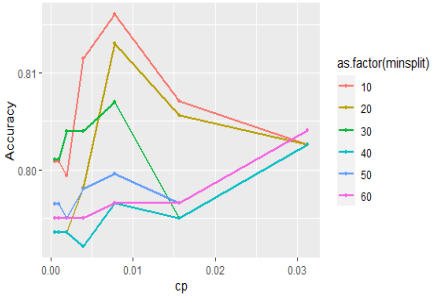
2. HYPERPARAMETERS

For getting better results, we start our search for hyperparameters. Some hyperparameters of decision trees are:

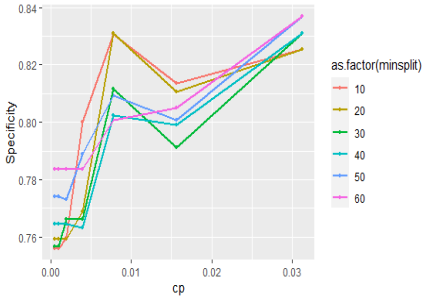
1. minsplit: it is the minimum number of observations that must exist in a node in order for a split to be attempted.
2. minbucket: the minimum number of observations that are allowed in a terminal node.
3. cp: complexity parameter. It is the minimum improvement in the model needed at each node.
4. maxdepth: set the maximum depth of any node of the final tree, with the root node counted as depth 0.

In our case, we set minsplit = seq(10, 60, 10) and 'cp' with very small values for having a better result (2^(-5:-11)). Now the same steps as we did for k-fold cross-validation must be performed. Contrary to those steps, now we need to add a new step. May we use the function 'apply 'over the 'lapply' function. This takes the data frame as an input and gives output in vector, list or array. It is used mainly to avoid explicit uses of loop construction. Once these steps are performed, we are able to get the best model with the following results:

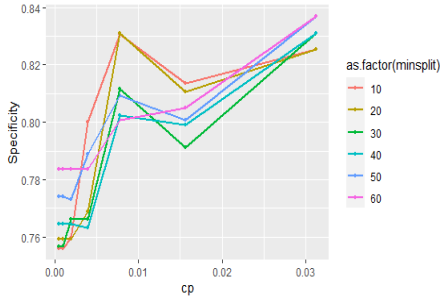
Accuracy	Precision	Specificity
0.8160222	0.8148509	0.8307395



(Figure 5)



(Figure 6)



(Figure 7)

May we observe that the results get a better model than the previous one. Moreover, we do not get the expected improvement, so we check the results of our random forest. The main reason why we are all the time looking forward to using random forests is because they provide the best model from 'x' number of decision trees.

2. RANDOM FORESTS

The random forest algorithm is an extension of the bagging method that creates a combination of decision trees (independent one from the other, randomly selected), and each of them is composed of a data sample drawn from the training set. Each individual tree in the random forest gives a classification and the class with the most votes becomes our model's prediction.

1. K-FOLD CROSS-VALIDATION

Another validation is required for assessing the effectiveness of our model. Moreover, in this case, we need to run the classification algorithm implemented in the package 'randomForest' with default hyperparameter values. This classification algorithm controls hyperparameters associated with decision trees. Then, we decide the number of folds(10), divide the dataset into training and test sets and, most importantly, choose the number of trees. Eventually, by computing a 10-folds cross validation, we get the next accuracies:

Accuracy	Precision	Specificity
0.8400172	0.8456387	0.8369757

Seeing the precious values we got, we can compare them with the ones obtained in the cross validation from the decision trees, and these last ones make the model more accurate. As we obtained best values computing the hyperparameters in the decision tree, now we will use hyperparameters in order to optimize the model and see which classification model is the best.

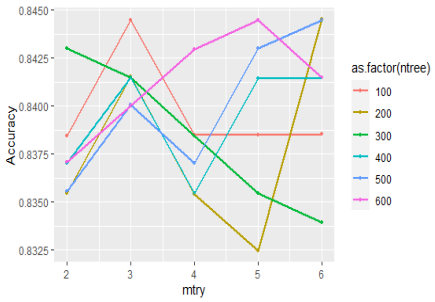
2. HYPERPARAMETERS

An outstanding functionality of the randomForest package is that we can find hyperparameters along the lines of:

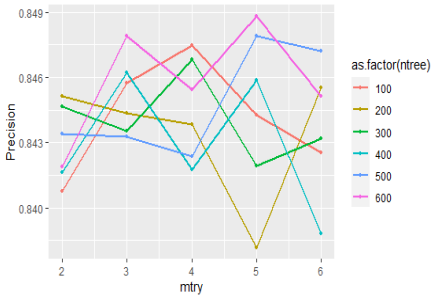
1. ntree (number of trees to grow, by default 500).
2. mtry (number of variables randomly sampled as candidates at each split, by default is the square root of the number of variables).

We have an additional argument 'classifier' which controls hyperparameters associated with decision trees. We also use the lapply function and we receive the following values:

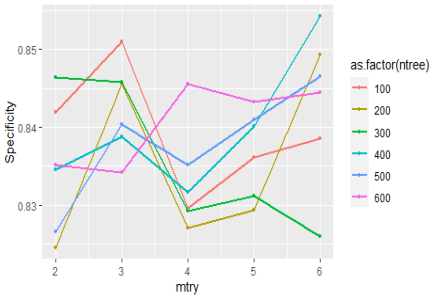
Accuracy	Precision	Specificity
0.8445181	0.845533	0.8493273



(Figure 8)



(Figure 9)



(Figure 10)

As seen, we have optimized our model and, indeed, we get a higher accuracy, precision and specificity than the ones obtained from the decision tree. The reason is that a random forest is composed of a certain number of decision trees and it compares the information that each tree gives to obtain the most "voted" ones and therefore the most reliable.

3. CHOOSING THE BEST MODEL

Once we have already pursued a further study of our dataset, it is important to choose which is the best. As observed in the results that have been stated above, we get to see that the model predicted using random forests was the one that got the higher performance. This actively demonstrates that, as supposed in page 9, we are keen on using random forests for the exact purpose that they provide the best model from 'x' number of decision trees.

Hence, it is time to fit the best model by creating a function that checks the performance of the model like the one from below (*Figure 11*). We could even evaluate the model with the given dataset. We get as an output the following results (*Figure 12*):

```
my_model = function(test_set){
  test_set$Ticket = NULL
  test_set$Cabin = NULL
  pred = predict(bestclassifier, test_set, type = "class")
  conf_matrix = table(test_set$Survived, pred)
  accuracy = sum(diag(conf_matrix))/sum(conf_matrix)
  precision = conf_matrix[1,1]/sum(conf_matrix[,1])
  specificity = conf_matrix[2,2]/sum(conf_matrix[,2])
  return(list(prediction = pred,
              conf_matrix = conf_matrix,
              accuracy = accuracy,
              precision =precision,
              specificity = specificity))
}

save(bestclassifier, my_model,file='BestModel.RData')
```

(Figure 11)

```
$conf_matrix
   pred
      0   1
  0 404   8
  1  44 212

$accuracy
[1] 0.9221557

$precision
[1] 0.9017857

$specificity
[1] 0.9636364
```

(Figure 12)

4. CONCLUSION

Both decision trees and random forests involve several advantages and disadvantages as we can observe in these tables.

DECISION TREES	
ADVANTAGES	DISADVANTAGES
1. Require less effort for data preparation during pre-processing. 2. Very intuitive and easy to explain.	1. A small change in the data can cause a large change in the structure. 2. Decision trees often involve higher time to train the model.

RANDOM FORESTS	
ADVANTAGES	DISADVANTAGES
1. Reduces overfitting in decision trees and helps improve the accuracy. 2. Works well with categorical and continuous values.	1. Requires much computational power. 2. Requires much time for training as it combines a lot of decision trees to determine the class.

As of right now, the main reason why we choose the random forest as our final model is due to the fact that a higher accuracy can be obtained than working with decision trees. Plus, seeing the disadvantages of random forests, it is worthier to take the risks that this type of machine learning algorithm may have given, as well as the fact that it provides better performance by compensating for the errors in the predictions of the different decision trees. In the decision tree model, this does not happen and therefore, as we proved, the error in terms of the accuracy is higher.
Random forest, in conclusion, is the best algorithm for producing our predictive model based on the survival of the passengers of the Titanic by the great results it gives to our model after the k-folds cross validation and computing hyperparameters.