Introduction to Data Science

Assignment 2: EDA, parallel programming and cross validation with Titanic Data

Date: 20/11/2020

Name: Alejo González García

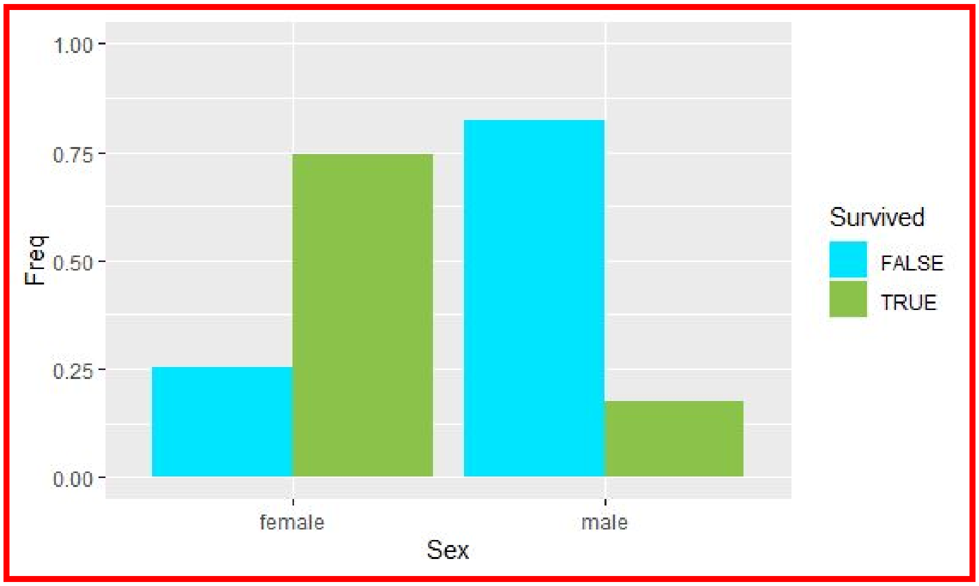
Name: Andrés Navarro Pedregal

Introduction

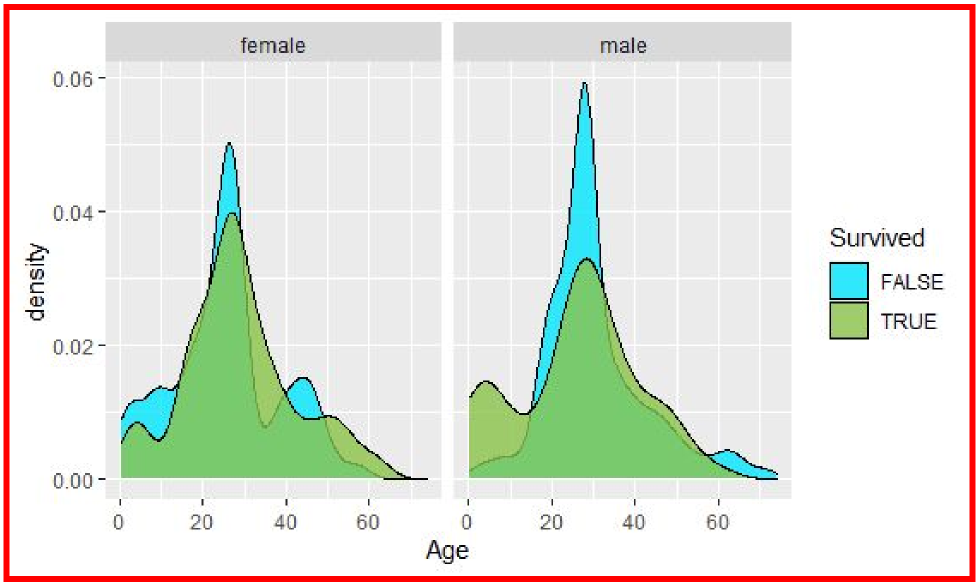
In this project, we are going to use algorithms such as **KNN**, **SVM**, **DT** or **RF** (explained below) in order to predict if a passenger who was travelling on the Titanic survived or not. To do so, we are getting the maximum accuracies for each of the methods implemented and we are keeping the best ones.
It is important to mention that in order to be capable of using some of these algorithms we had to delete the columns of **Ticket** and **Cabin** because it did not apport information and caused the programs to crash while executing Random Forest and some others. The reason is that in the case of Ticket, there are many different types and cannot be used to predict, and in the case of the Cabin variable, most of the passengers did not have data and caused troubles.

Exploratory Data Analysis

After analyzing the data, we saw that the variable Sex and Age are the most important ones as there are a clear relationship with the Survived variable as shown in the following graph.



Moreover, we could analyze that the most frequent age group that did not survive was the adult. However men the ones that did not have a lot of chances to survive, young male children were the ones that survived the most in proportion to the ones that died for the same age and sex group.



Finally, the last part in our EDA was to calculate the probability for each group according to their Age, Sex, SibSp, and Parch and see which groups were the ones with highest probabilty. After computing the probabilities we got the following table.

```
      AgeInterval    Sex      SibSpInterval    ParchInterval    total   totalSurvived   probabilitySurvived
## 5      60 to 75 Female        0 to 2           0 to 2          4           4                1.000
## 13     30 to 45 Female        2 to 4           0 to 2          2           2                1.000
## 14     45 to 60 Female        2 to 4           0 to 2          1           1                1.000
## 19     45 to 60   Male        2 to 4           0 to 2          1           1                1.000
## 46      0 to 15   Male        0 to 2           2 to 4          5           5                1.000
## 6       0 to 15   Male        0 to 2           0 to 2         12          11                0.917
## 3      30 to 45 Female        0 to 2           0 to 2         46          42                0.913
## 1       0 to 15 Female        0 to 2           0 to 2         12          10                0.833
## 52     15 to 30 Female        2 to 4           2 to 4          6           5                0.833
```

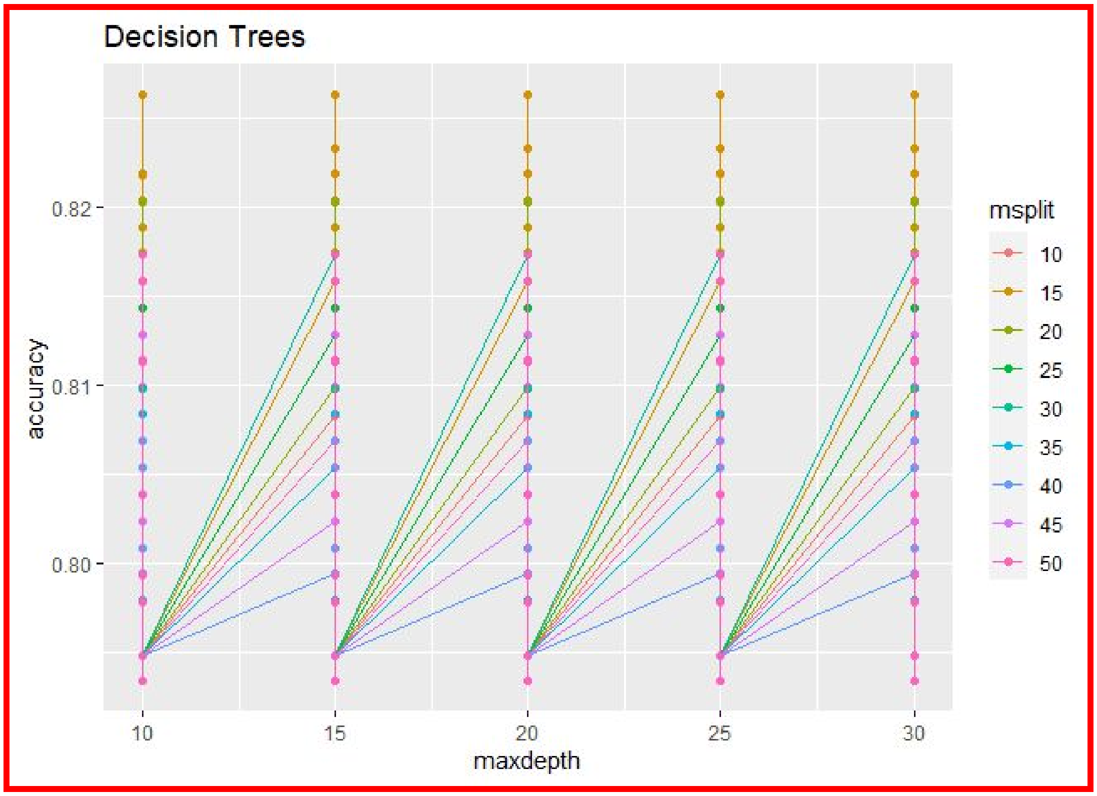
In this table we can observe that females between the ages of 30 to 75 and male children had the highest probability to survive. Therefore we can make a risky assumption that in case of an accident, usually the ones that rescue crews save first are women and children.

In conclusion, after the EDA we must focus our attention to the age and sex group as they are the most important variables of our datasheet. Moreover, we decided to delete the variables ticket and cabin as they did not show any relation with the Survived variable and they did not work with the models.

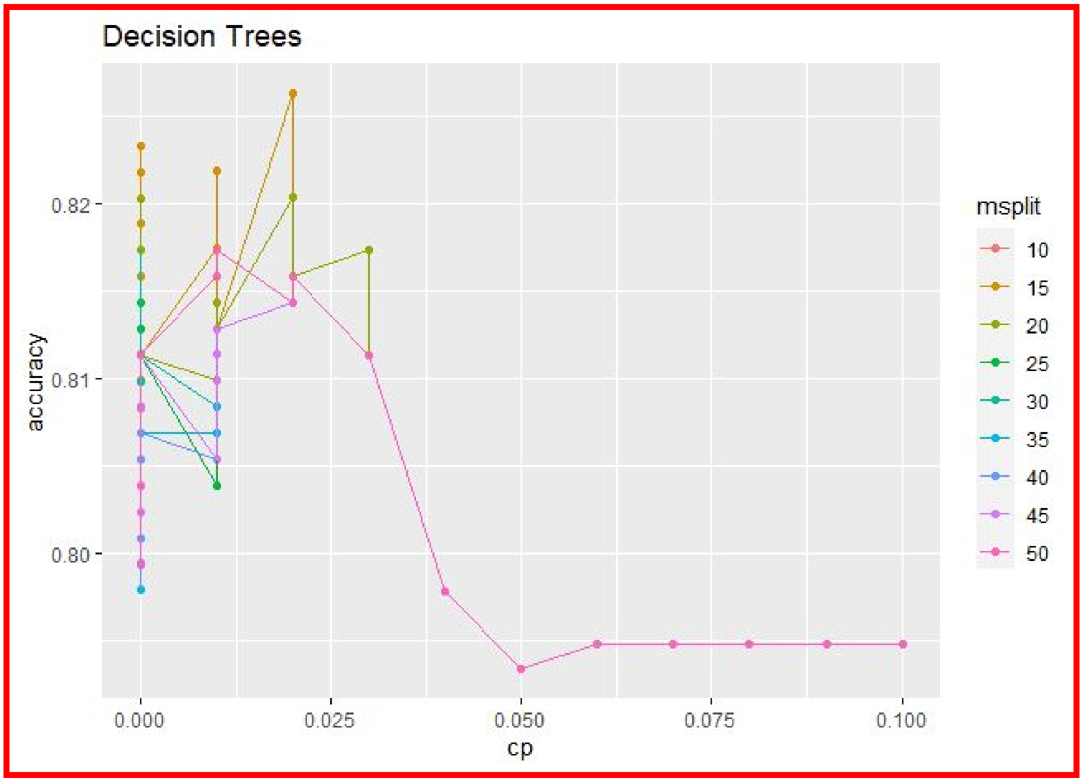
Decision Trees

This algorithm consists on splitting the data into different leaves and takes as input values such *minsplit* (minimum number of observations needed in order to split the set), ***maxdepth*** (the maximum length while splitting), ***complexity parameter*** (is the complexity of each tree) and ***minbucket*** (minimum observations that each tree needs to have).

We have chosen different values for each of these parameters and thanks to do cross validation, we are capable of knowing the best prediction tree. The obtained accuracy is around **0.82**, a quite good result, which means that you can predict if a passenger survived successfully in **82%** of the cases. Is so important to take into account that the accuracy cannot be close to 100% because there will be overfitting (when new data is added it does not fit well in those trees).



In this plot we can see how the accuracy does not change while maxdepth variates.



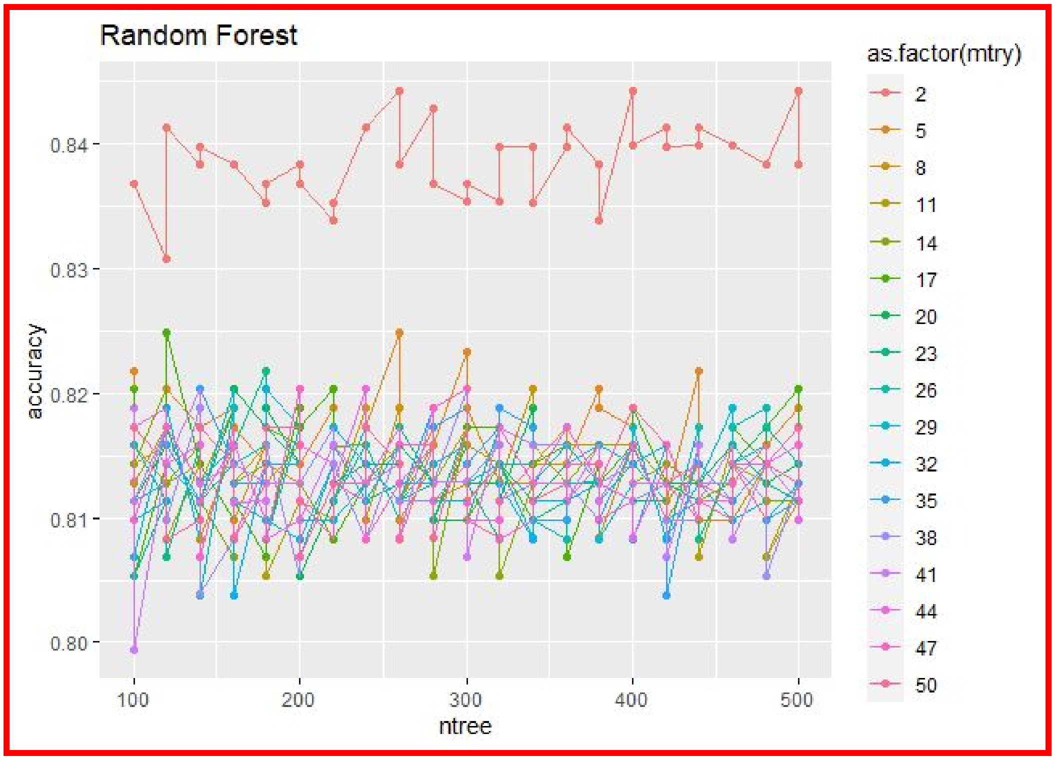
Depending on the complexity parameter, the accuracy variates, cp is only useful in the range 0 to 0.03

Random Forest

In this algorithm, we are using many trees explained above. A random forest is a huge amount of trees that working in order are capable of predicting a variable and are pretty good at avoiding overfitting.

The parameters we have been working on are ***mtry*** (number of variables randomly sampled as candidates at each split), ***ntree*** (the number of decision trees used) and ***replace*** (meaning if the samples should be done with or without replacement).

This model seems to be the most accurate one with **0.848** as maximum, also is pretty good avoiding the mentioned overfitting.



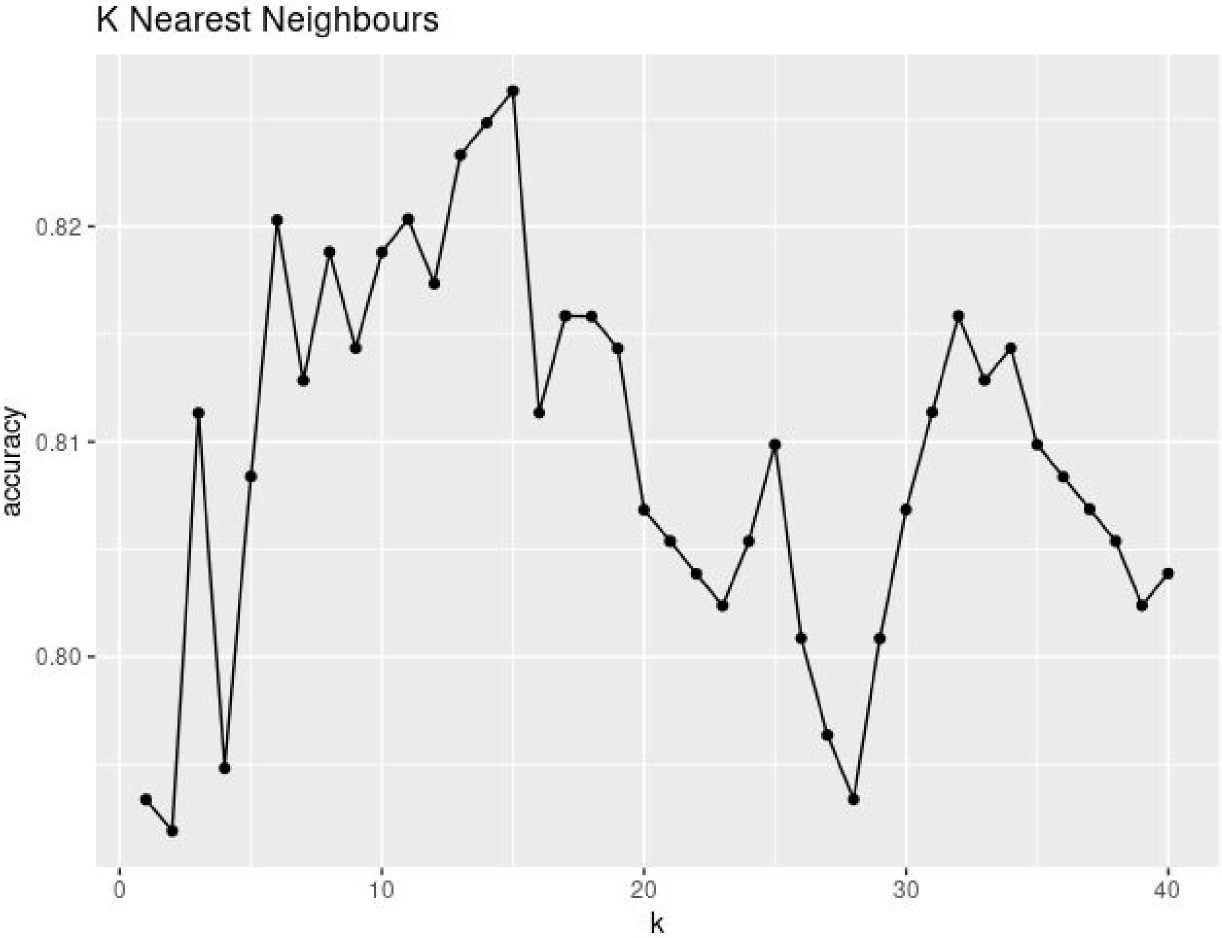
This plot allows us to see how the best accuracy is obtained with 2 as mtry value, also that accuracy does not variate with the different values of mtry, being the accuracy in an interval of 0.8 to 0.825.

K-Nearest Neighbours (KNN)

KNN is a supervised machine learning algorithm that relies on training data and this must be numeric. It works by getting the value of the **k-nearest neighbours** for that value, and then repeating this process again and again to get an accurate prediction.

In our model, we used a minimum of **1** neighbour and a maximum of **40** neighbours and we tested each model to get the best one and also we standardized all the variables to get the maxim accuracy.

After a number of attempts, we got a maximum accuracy of **83.8%** which was good. In our opinion, it was because as the data frame consists of a lot of mixed up information where you can find a real division between the characteristics of the people that survived and the people that died. Therefore it is not the best model to use with this data set.

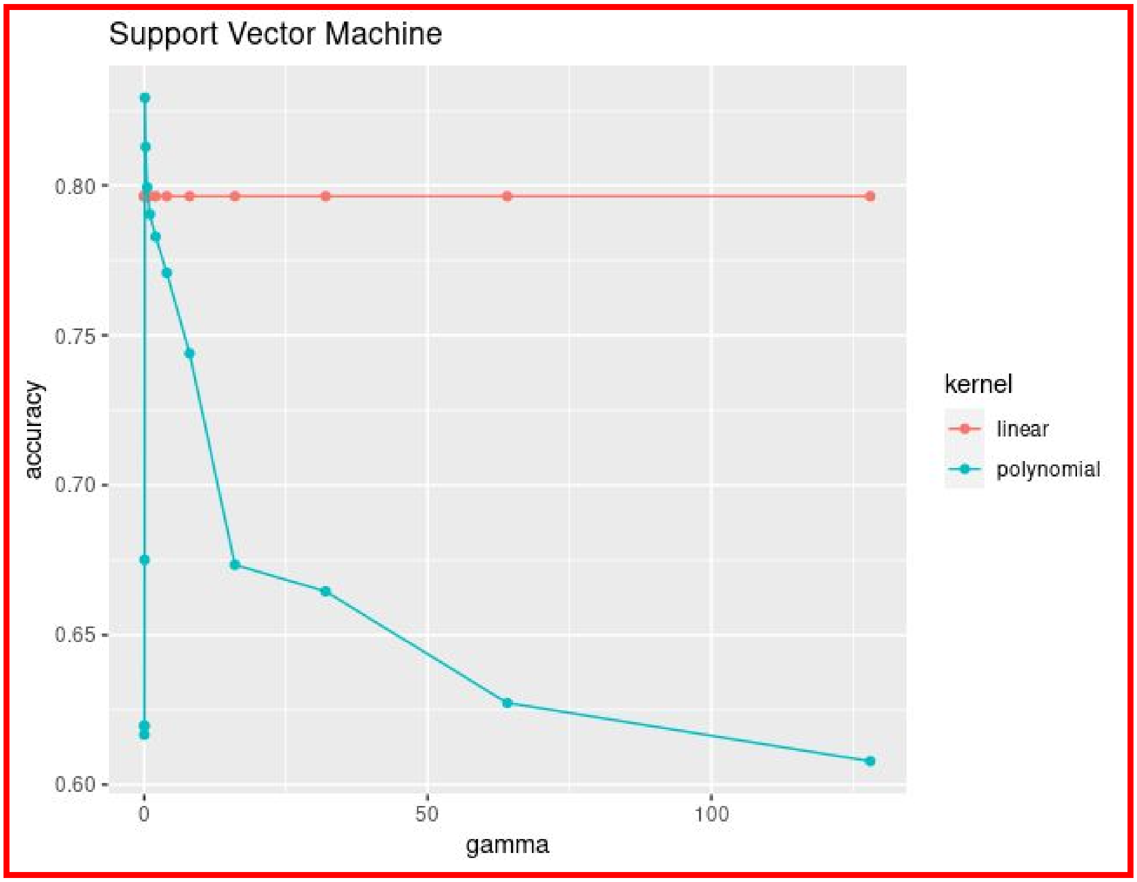


Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm which classifies the data separating it with line, parabola, plane, etc. This algorithm tries to get a separation where most of the data is divided. This is achieved by using kernels, which are a specific decision boundary such as linear, polinomial, radial, or sigmoid.

In our model, we use C-Classifications, values of gamma between **0.007** and **130** and polynomial and linear kernels.

After a number of attempts, we got a maximum accuracy of **82,9%** for ***gamma* = 0.125** and a polynomial kernel , which hardly ever is the highest accuracy of all the models. Therefore, SVM is a good model to predict the data.



Optimal Model

As we have been seeing, the best model in order to predict in this dataset is the **Random Forest**.
The accuracy rounds **0.849** for the values of ***mtry = 2*** and ***ntree = 440***.

It is so important to remark that we have been setting a seed (it can be seen in the RFile) to always make the same sample and get the same results, but as soon as you remove it, you can check that the results keep being almost identical.

Mention the use of **Parallel Programming** (Use of all the processor cores at the same time) to compute the results about 5-8 times faster depending on the processor cores. And also, **Cross Validation** to get a random sample to train the data and make the model the most accurate for any given data.

Conclusion

We can end up saying that we have achieved our target. We are capable of predicting the variable Survived using the mentioned algorithms and the most accurate one is the Random Forest, so with an 84% of accuracy we can determine if any passenger of a greater dataset survived or not.