NO MORE LOW COST

Statistical Learning Assignment

Pignatelli Nicolò

1.ABSTRACT

In this report I present my statistical learning assignment. I use R as the programming language. I base my analysis on a data set about air flights in India. After some data preparation and explorative data analysis, in the supervised learning part I try to build the best model in order to explain and interpret the data, while in the unsupervised part I try to discover interesting insights about some features of the data set as I have never seen it before. It is important to notice that both parts are implemented on the same data. In this report I try to explain as clearly as possible the most important findings about the analysis and provide a step-by-step commentary. In the end I try to draw some conclusions.

2.GOAL OF THE ANALYSIS AND DESCRIPTION OF THE DATA SET

In the August of 2022, Michael O'Leary, the CEO of "Ryanair", claimed that the era of the low-cost flights was over (https://www.ilpost.it/2022/08/11/ryanair-biglietti-costi/, from the Italian journal "il Post"). Given the reliable source, it is time to take the necessary countermeasures in order to go on holiday without spending all the money just for the flight. A good thing to do is to understand which features are the most impactful on the ticket price. So, the goal of the analysis is to build the best model in order to explain and interpret the data, in particular which features are the most important ones given the dependent variable, the price of the ticket of the air flight, and with which statistical technique I could predict the price in the most precise way.

I downloaded the data set from https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction. The original source is "EasemyTrip" website. The data set is composed of 300153 observations and 12 features. The features are:

1) X: the id number that identifies a ticket.
2) airline: the name of the airline company. It is a categorical variable. There are 6 companies in the data set (Air_India, AirAsia, GO_FIRST, Indigo, SpiceJet, Vistara).
3) flight: a code identifying each flight.
4) source_city: the city from which the flight takes off. It is a categorical variable. There are 6 Indian cities in the data set (Bangalore, Chennai, Delhi, Hyderabad, Kolkata, Mumbai).
5) departure_time: the time of the day during which the flight takes off. It is a categorical variable. There are 6 times of the day (Afternoon, Early_Morning, Evening, Late_Night, Morning, Night).
6) stops: how many stops occurs during the flight. It is a categorical variable. There are 3 possible values (zero, one, two_or_more)
7) arrival_time: the same as departure_time, but it is the time of the day during which the flight lands.
8) destination_city: the same as source_city, but it is the city where the flight lands.

9) class: the seat class, a dummy with “Business” and “Economy” as values.
10) duration: how many hours the flight lasts.
11) days_left: how many days in advance the ticket was bought.
12) price: the price of the ticket. The dependent variable of the analysis.

3.KEYPOINTS

1. The analysis is composed of both regression and classification task.

2. Random Forest for both tasks is the model with the highest predictive power, but for regression the tree predictor is worthy of note as it is a very good model for prediction and it is easy to interpret.

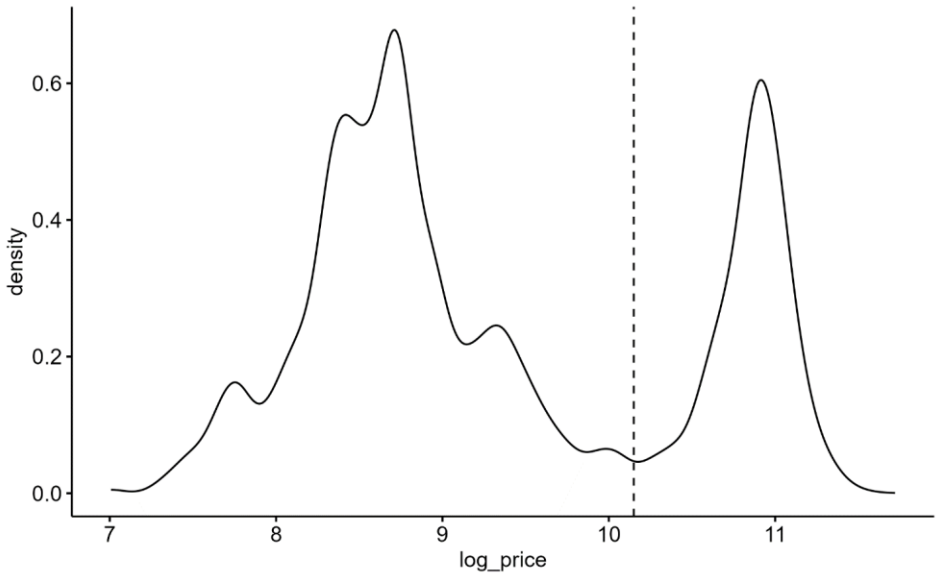
3. The most important variable is the same for the two tasks, but in general the importance of the other regressors and their ranking in the importance plot may vary, even greatly, so predicting the price as a continuous variable seems to be very different from predicting if the price is low or high.

4. The relationship between the price of the ticket and the duration of the flight is ambiguous and very difficult to tackle with unsupervised techniques.

4.COMMENTARY

4.1 DATA PREPARATION AND DESCRIPTIVE ANALYSIS

I import the dataset as “raw_db” and immediately check how many unique values each feature contains. I decide to drop 2 variables: “X” and “flight”. I drop the first one as it is completely useless for the analysis as I want to carry it out; the second one because I do not see how I can interpret it. After this, I inspect the number of NAs present in the data set, but all cells are filled with values. I create a new data set, “complete_db”, and convert all the categorical variables (airline, source_city, departure_time, arrival_time, destination_city, class and stops) into factors. Then I plot all the regressors in order to understand their distribution and verify if any typo or outlier is present. Overall, I do not find anything remarkable. The only important thing to signal is that I change the variables related to the cities and the times with “from_” before the values of the departing ones and “to_” before the others, in order to avoid confusion (e.g.: Mumbai becomes “from_Mumbai” in “source_city” and “to_Mumbai” in “destination_city”, likewise “Night” becomes “from_Night” and “to_Night” in, respectively, departure and arrival time features). After the independent variables, I study the price, the dependent one, but again I do not find any typo or outlier. I plot the distribution and observe a very particular shape. So, I decide to take the logarithm of the price to reduce the skewness towards large values. I do not even try to test for normality, but, as I see the distribution, I think that it could be appropriate to divide the feature into 2 groups: low prices and high prices.



I decide to create a new data set, "class_db", where the dependent variable is a dummy variable where "1" means high price (greater or equal than the threshold, logarithm of the price equal to 10.15) and "0" low price. The new variable has a prevalence of low prices, but, given the high total number of observations, there are still a good amount of high prices. So, I decide to try to extend my goal to both regression, where I will use the logarithm of the actual price, and classification task. I plot the correlation matrix for both data sets, but there are not very high correlation coefficients between the regressors, so I move on to the supervised part.

4.2 SUPERVISED LEARNING

As the first thing, I split both data sets into training and test set. I use the same indices to identify the two sets for both data sets, so all the models are tested on the same observations (they still have different dependent variables obviously). I do this in order to carry out a homogenous analysis, even if the comparison between the two tasks will be only partial.

4.2.1 REGRESSIONS

I start building the easiest models, the regressions, for both tasks. I do this because they are very straightforward and also because, if I have to decide in the end which model I would say that does the job in the best way, I may argue that I could sacrifice a bit of precision for more interpretability, that is usually provided by these models, even if in this case the high number of regressors may lead to choose other more interpretable ones.

The first model for the regression task is **linear regression**. The outcome is very good: all the coefficients are significant and the R squared of the model with the training data is 0.9154. From now on, in order to be able to compare the precision of the regression models, I compute the Root Mean Squared Error (RMSE). I predict the values of the training set with my model and I obtain a RMSE of 0.3237. I do this because now I have a benchmark. Then I test my model on the actual test set and the RMSE is **0.3238**, so I can be satisfied as the test error is very low and similar to the training error. Now I do some diagnostics in order to understand how much reliable these errors are. I verify if the model suffers from multicollinearity, but no variable has a square root of the vif greater than 2. Another possible issue is if the residuals are not normally distributed. I plot them and test the normality hypothesis. It is rejected, so we cannot say that our linear regression model is valid.

In order to overcome this violation of an assumption of the ordinary least squares, I run the **robust regression model**. I test the model and the RMSE is very similar to the previous model**: 0.3244**.

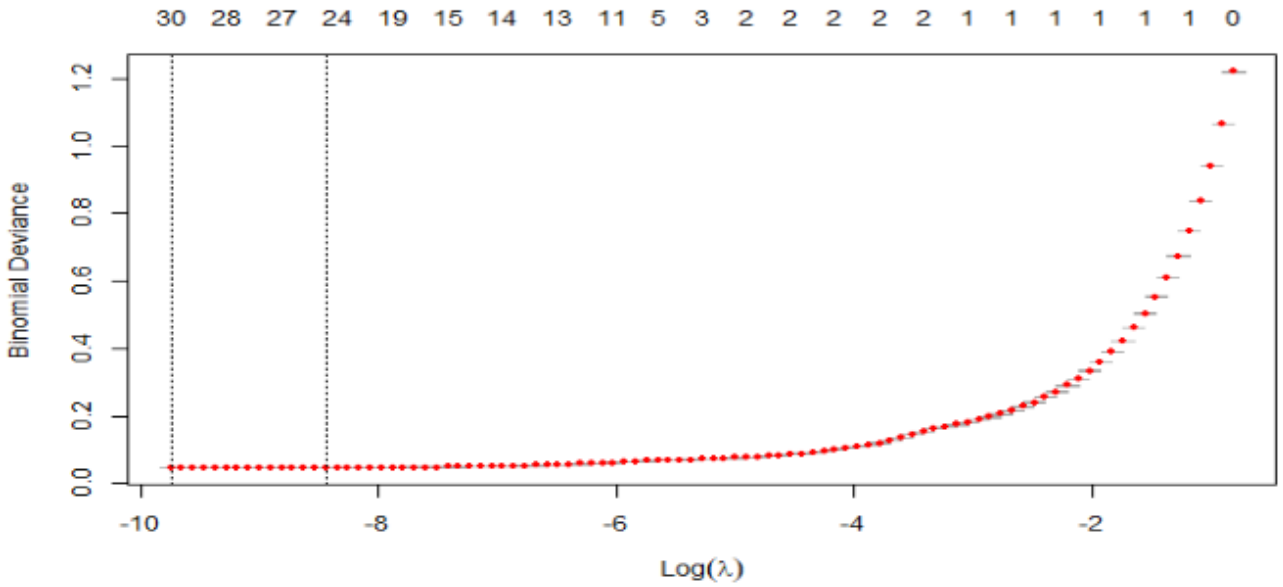
```
Coefficients:
                                 Value      Std. Error t value
(Intercept)                    11.1029      0.0047  2379.8910
airlineAirAsia                 -0.5642      0.0035  -159.7243
airlineGO_FIRST                -0.1130      0.0031   -36.7731
airlineIndigo                  -0.1943      0.0027   -73.1940
airlineSpiceJet                -0.0566      0.0043   -13.0623
airlineVistara                  0.1292      0.0017    73.8435
source_cityfrom_Chennai        -0.0491      0.0026   -18.8632
source_cityfrom_Delhi          -0.0331      0.0024   -14.0435
source_cityfrom_Hyderabad      -0.0904      0.0026   -35.0075
source_cityfrom_Kolkata         0.1362      0.0025    54.4583
source_cityfrom_Mumbai         -0.0444      0.0024   -18.8938
departure_timefrom_Early_Morning  0.0178    0.0023     7.6601
departure_timefrom_Evening     -0.0064      0.0024    -2.6988
departure_timefrom_Late_Night   0.0375      0.0109     3.4359
departure_timefrom_Morning      0.0313      0.0023    13.7439
departure_timefrom_Night       -0.0066      0.0026    -2.5794
stopstwo_or_more                0.2414      0.0035    69.3808
stopszero                      -0.4660      0.0026  -180.3028
arrival_timeto_Early_Morning   -0.0643      0.0037   -17.2819
arrival_timeto_Evening          0.0292      0.0024    12.1061
arrival_timeto_Late_Night       0.0125      0.0039     3.1836
arrival_timeto_Morning         -0.0113      0.0025    -4.4832
arrival_timeto_Night            0.0313      0.0024    13.2603
destination_cityto_Chennai     -0.0465      0.0026   -18.0495
destination_cityto_Delhi       -0.0300      0.0024   -12.4041
destination_cityto_Hyderabad   -0.0982      0.0026   -38.4613
destination_cityto_Kolkata      0.1038      0.0025    42.1039
destination_cityto_Mumbai      -0.0259      0.0024   -10.8837
classEconomy                   -2.0367      0.0017 -1202.8504
duration                        0.0030      0.0001    22.5110
days_left                      -0.0134      0.0001  -261.7206
```

Now that I have a robust regression, I may try to capture some information about the importance of the regressors. Considering only the categorical variables, it is easy to see that “class” seems the most important ones, as it has the highest value of the coefficient in absolute value. All the other ones have a way lower coefficient. It is difficult to interpret the coefficients of the numerical ones, as they are not scaled. In general, I can say that, as expected, “duration” has a positive correlation with the price (the longer the flight, the higher the price), whereas “days_left” has a negative one (more in advance I book, the lower the price).

The first model for the classification task is **logistic regression.** I obviously use the other data set, where my dependent variable has only two possible values, “0” and “1”. Unlike linear regression, this model has two coefficients with a p-value higher than 0.1. Anyway, I test the model and the confusion matrix shows that the accuracy is very high: **99.54%**.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 62789   196
         1   216 26843

               Accuracy : 0.9954
                 95% CI : (0.995, 0.9959)
    No Information Rate : 0.6997
    P-Value [Acc > NIR] : <2e-16
```

In order to try to improve the logistic, I decide to run the lasso. I choose this model because it may select only a subset of variables and drop others, so I may obtain a dimensionality reduction where only the important variables are kept. First of all, I tune the lambda parameter with cross-validation.



Once I have the optimal lambda, I run the model on the training set and then test it. I print the confusion matrix in order to evaluate the outcome.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 62785   191
         1   220 26848

               Accuracy : 0.9954
                 95% CI : (0.995, 0.9959)
    No Information Rate : 0.6997
    P-Value [Acc > NIR] : <2e-16
```

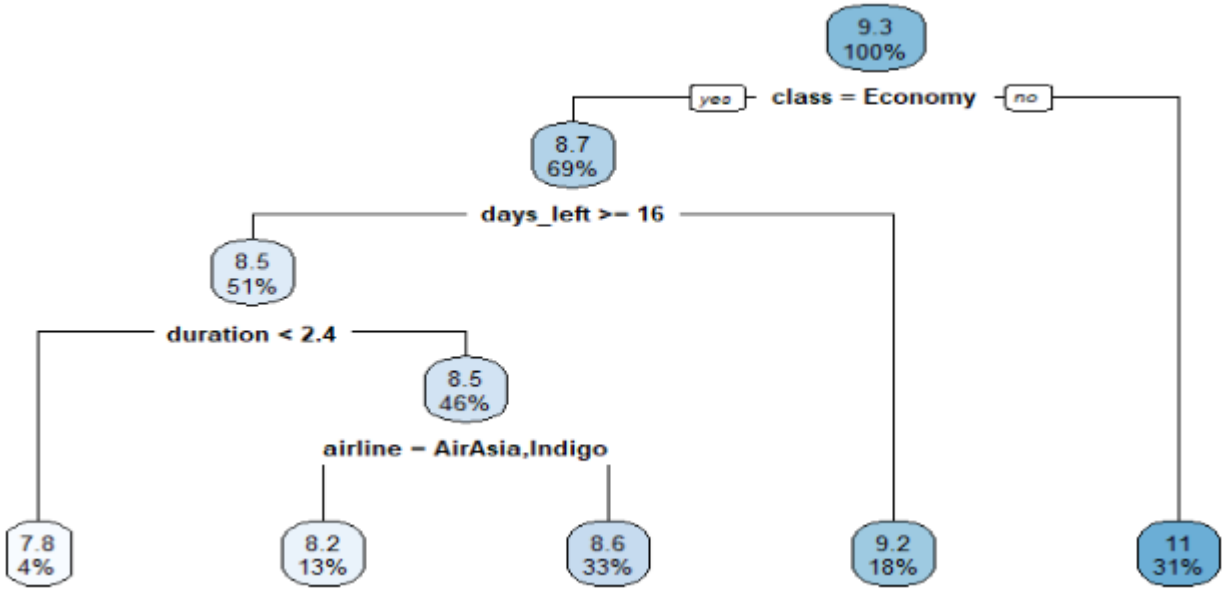
The accuracy is the same as the logistic regression, **99.54%**, but, overall, the model classifies correctly a unit more out of 90044. Not really the improvement desired but anyway, with model I built until this point, I can say that an advantage of building the lasso is dropping a regressor without reducing my accuracy, even if I lose some interpretability with respect to simple logistic regression.

	s0
(Intercept)	7.045311e+00
airlineAir_India	-1.111447e+00
airlineAirAsia	.
airlineGO_FIRST	-4.564419e-04
airlineIndigo	1.867966e-01
airlineSpiceJet	-1.479455e+00
airlineVistara	3.422517e-01
source_cityfrom_Chennai	2.569449e-01
source_cityfrom_Delhi	1.378669e+00
source_cityfrom_Hyderabad	-2.225419e+00
source_cityfrom_Kolkata	1.104687e+00
source_cityfrom_Mumbai	-1.418131e+00
departure_timefrom_Early_Morning	4.998942e-01
departure_timefrom_Evening	2.020031e-01
departure_timefrom_Late_Night	-2.015827e+00
departure_timefrom_Morning	4.941743e-01
departure_timefrom_Night	1.714762e-01
stopstwo_or_more	2.447290e-01
stopszero	-6.045531e+00
arrival_timeto_Early_Morning	-4.860940e-01
arrival_timeto_Evening	-3.030406e-01
arrival_timeto_Late_Night	-3.667720e-01
arrival_timeto_Morning	-6.505186e-01
arrival_timeto_Night	1.179569e-01
destination_cityto_Chennai	-1.623441e-01
destination_cityto_Delhi	1.242520e+00
destination_cityto_Hyderabad	-2.655348e+00
destination_cityto_Kolkata	1.043066e+00
destination_cityto_Mumbai	-1.579184e+00
classEconomy	-1.577413e+01
duration	1.544041e-01
days_left	-3.139102e-02

Taking into account the lasso coefficients, it is interesting to notice that the coefficient shrunk to 0 is one of the most important categorical variables in the robust regression in order to predict the logarithm of the price. The variable “class” is still the most important between the categorical. The continuous variables have the correlation coefficients with the same signs as in the robust linear regression. As I said before, the only comparison that can be carried on across the tasks is the one between the most important variables for both of them.

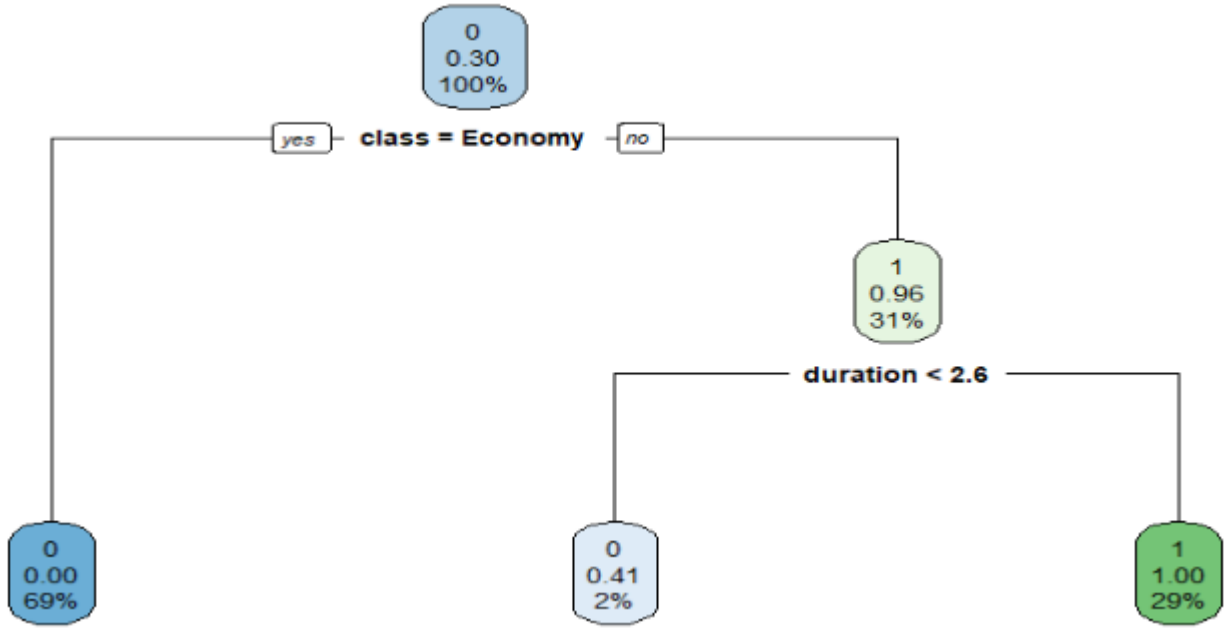
4.2.2 TREES

A model that can be used for both tasks is the **tree** predictor. It is also simple and very interpretable. The procedure to build the model is the same for both tasks. I start by building a very large tree. Then I prune until a certain threshold of CP (“cost of pruning”) is reached. In this way I obtain a tree that does not overfit and on which I can control the error. I start again with regression.



As expected, the tree predictor is very easy to interpret. Again, "class" seems the most important feature, as, if the passenger travels in Business class, the model immediately predicts the logarithm of price without even considering the other features. Surprisingly, if the "class" is Economy, the model predicts a higher price if "days_left" is high. Then, as expected, with lower duration the price is low. In the end, if the passenger travels with two airlines, the price is lower, compared with the other ones. One of them, "AirAsia", is the one important for the robust regression and shrunk to zero by the lasso. It seems like that this level of the factor "airline" is important to predict the value of the price, but not to classify if a price is low or high, conditioned to the class in this case. The test error of the model is low: the RMSE is **0.3301**. The tree seems to manage very well the trade-off between prediction accuracy and interpretability.

I go on with the tree for classification.



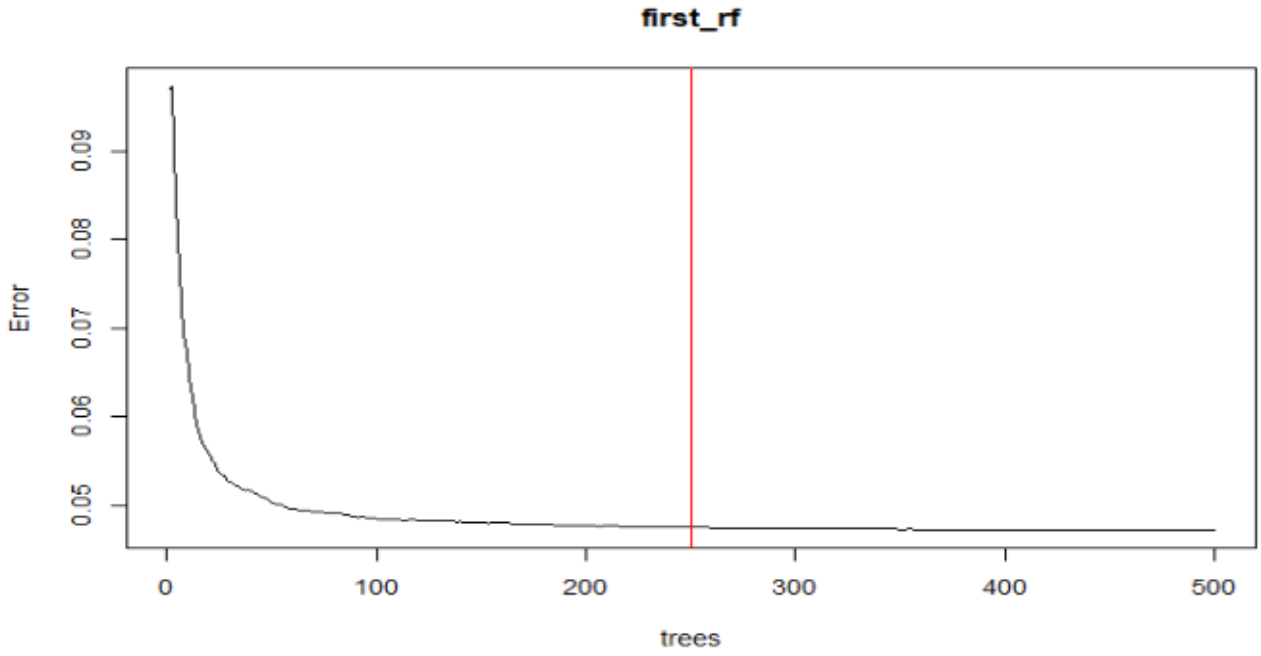
The result seems too much simple to be a robust model. The features that it takes into consideration are only “class” and “duration”. The behaviour is very similar to the regression tree: first thing to consider is the class, and, if it is Economy, it predicts low price. Then the only other feature is “duration” and, as in regression, if it is lower than a certain threshold, similar to the tree of the other task, it predicts low price, otherwise high price. In order to understand if the model manages the trade-off in a good way as the other tree, I print the confusion matrix.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 62945   812
         1    60 26227

               Accuracy : 0.9903
                 95% CI : (0.9897, 0.9909)
    No Information Rate : 0.6997
    P-Value [Acc > NIR] : < 2.2e-16
```

The accuracy is lower, **99.03%**, as expected, but not that much. In absolute terms, I cannot say that this model classifies poorly. The point is that it classifies very well the low prices but has a high error with high prices, more than 3%, whereas lasso has less than 1%. I am not completely satisfied with it.

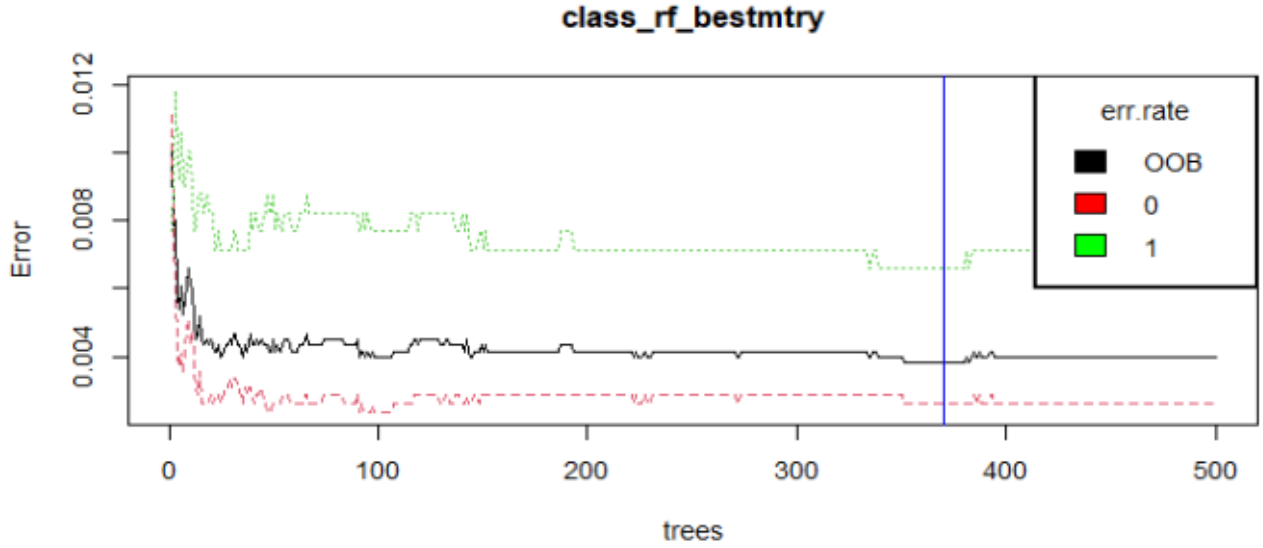
4.2.3 RANDOM FORESTS

Another model that can be used both for regression and classification is **random forest**. It is a powerful tool and I also expect a better performance than the trees one, given that the random forest combines more trees together to improve the precision of the model. For both tasks I am forced to split again the data sets into training and test set. I choose a smaller number of observations as my training data because the performance of my laptop is not amusing. This means also that the test set is bigger. With respect to the previous models, I train random forest and test it on different sets, but this should not be a problem until I choose randomly the sets. Furthermore, in both cases I have a high number of observations to test on. Again, the procedure for building the model is the same for both tasks. Firstly, I tune the “mtry” hyperparameter, i.e. the number of variables randomly chosen every time to split a node. I expect a number around 3, as my regressors are 9, but the best mtry is 9 in both regression and classification task. After this, I tune the other hyperparameter of the model, the number of trees. To avoid overfitting in regression, I plot the error as a function of the number of trees and I choose the lowest value for which the error may not be minimum but is very low and a higher value would not benefit so much the test error. The image below shows the choice. The value, 250 trees, may not be the one with the lowest value, but the error seems to be more or less stable from that point on. For classification task I choose the number of trees that minimizes the OOB error.



I start again with regression. After tuning the mtry and the number of trees, I test the final model and obtain a RMSE equal to **0.2158**, so the random forest is the regression model with the best predictive power.

For classification, as I said, I tune the mtry hyperparameter and then choose the number of trees that minimizes the OOB error. I am able to plot again this quantity as a function of the number of trees and even to identify with different colours the errors for value “0”, “1” and the general one.

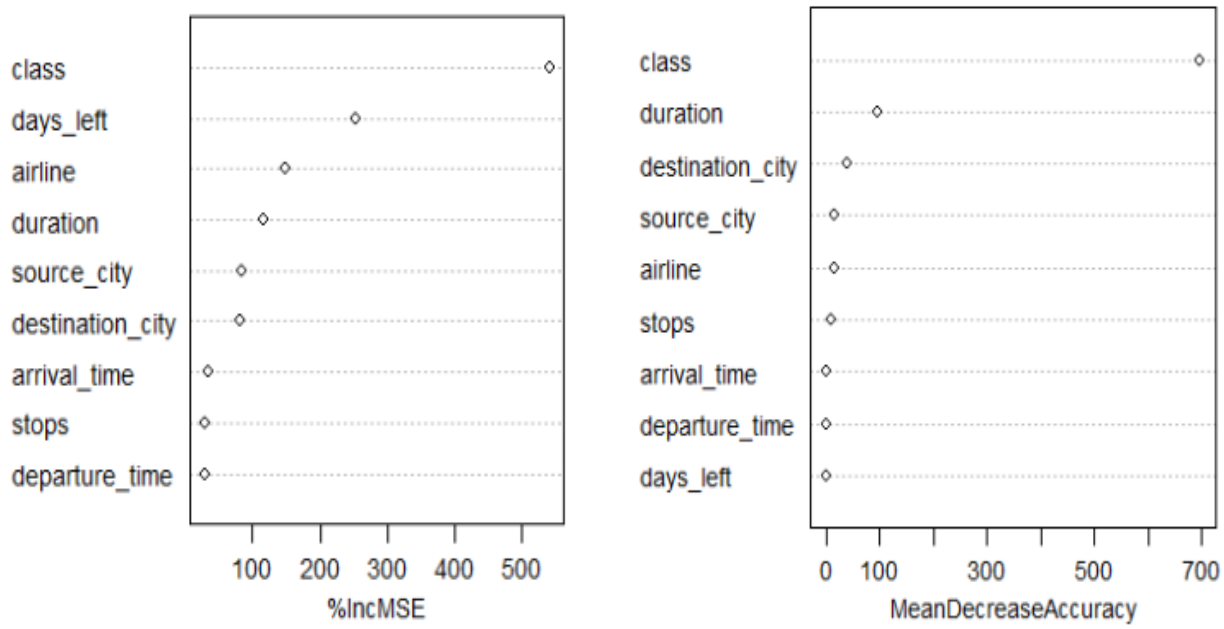


I build the model with the tuned hyperparameters and I test it. I print the confusion matrix in order to evaluate the result.

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 205502    791
         1    408  87447

               Accuracy : 0.9959
                 95% CI : (0.9957, 0.9962)
    No Information Rate : 0.7
    P-Value [Acc > NIR] : < 2.2e-16
```

As for the regression task, the random forest is the best model, as it has an accuracy of **99.59%** (remember that test set is bigger, so the number of errors is higher in absolute terms).

A good advantage of building a random forest model is that it is possible to measure the importance of every variable for the prediction as how much the error would grow without that regressor. Given that the random forest is the best model for both the tasks, it seems appropriate to plot these measurements in order to see which are for each task the most important variables and to compare if there is any difference across the two.

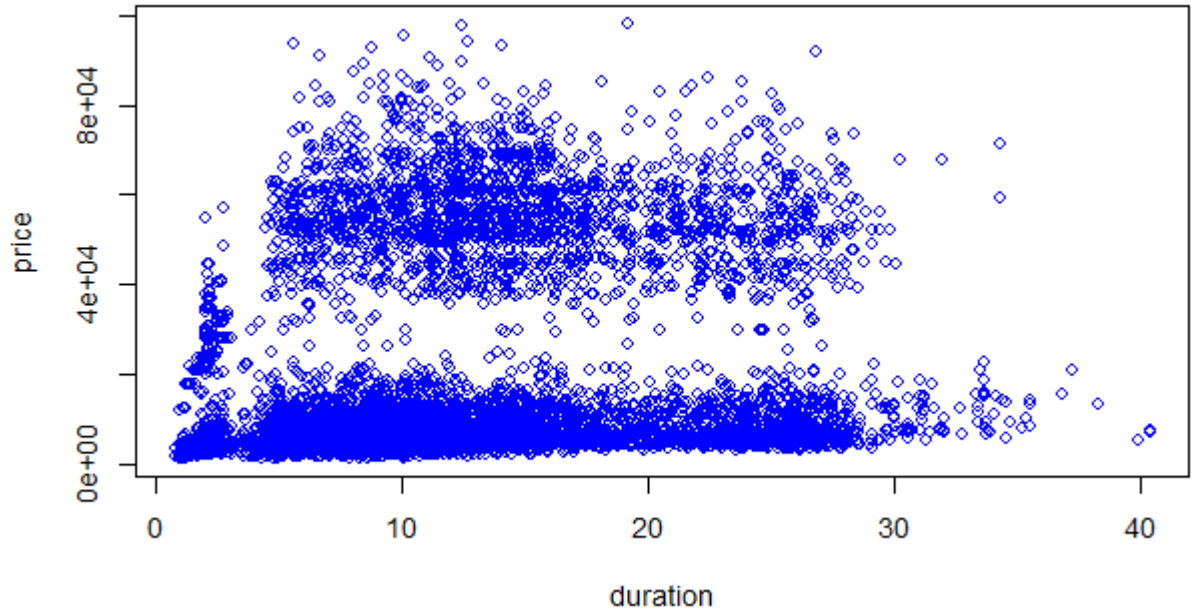


As expected, “class” is for both the most important variable, but in the classification task the growth in the error without this variable seems to be much bigger than without any other variable, whereas in regression the situation is more balanced. So, the importance of the first variable is higher in classification than in regression. This probably explains the difference between the trees. The most surprising difference is “days_left”: for regression (left panel) is the second most important variable, for classification is the least important one. All the other variables rank more or less in the same position, but still there are differences (e.g.: “airline”).

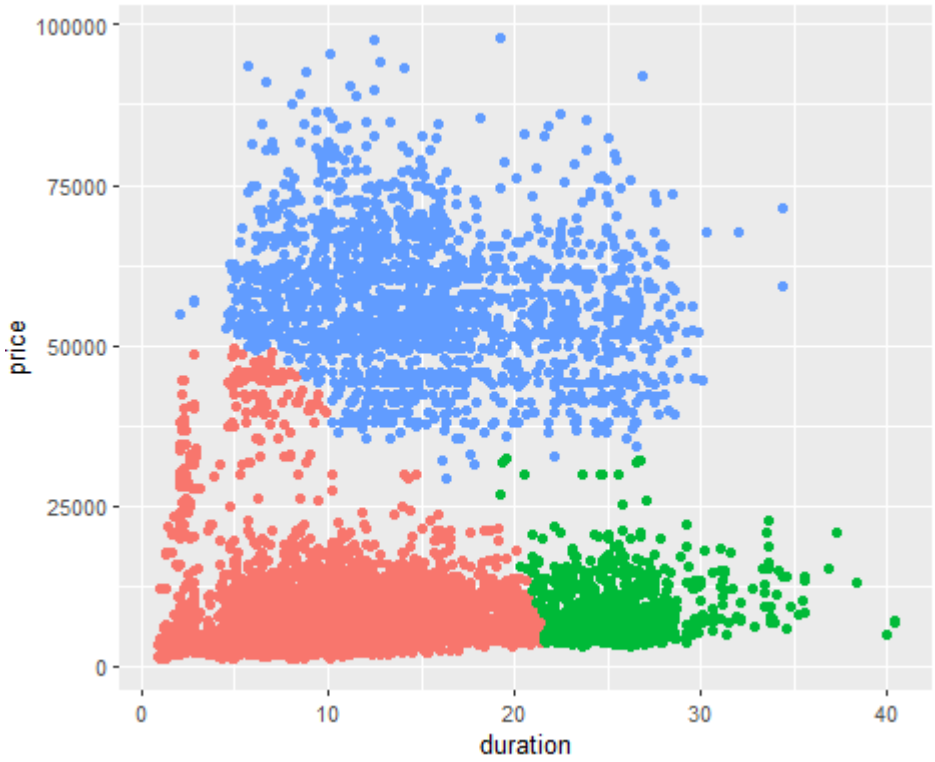
4.3 UNSUPERVISED LEARNING

In the unsupervised learning part I try to discover interesting insights about the data as I have never seen them before. I decide to perform the clustering technique as the PCA is usually done with a high number of variables, which I do not think it is my case. I prepare the data set and again I have to choose only a subset of data because otherwise I could not compute the clusters. I decide to plot

“price” (the original one, without any transformation) with the “duration”. I expect a positive correlation: longer the flight, higher the price. It does not seem this the case, or at least it is not that clear, so I decide to perform the hierarchical clustering.

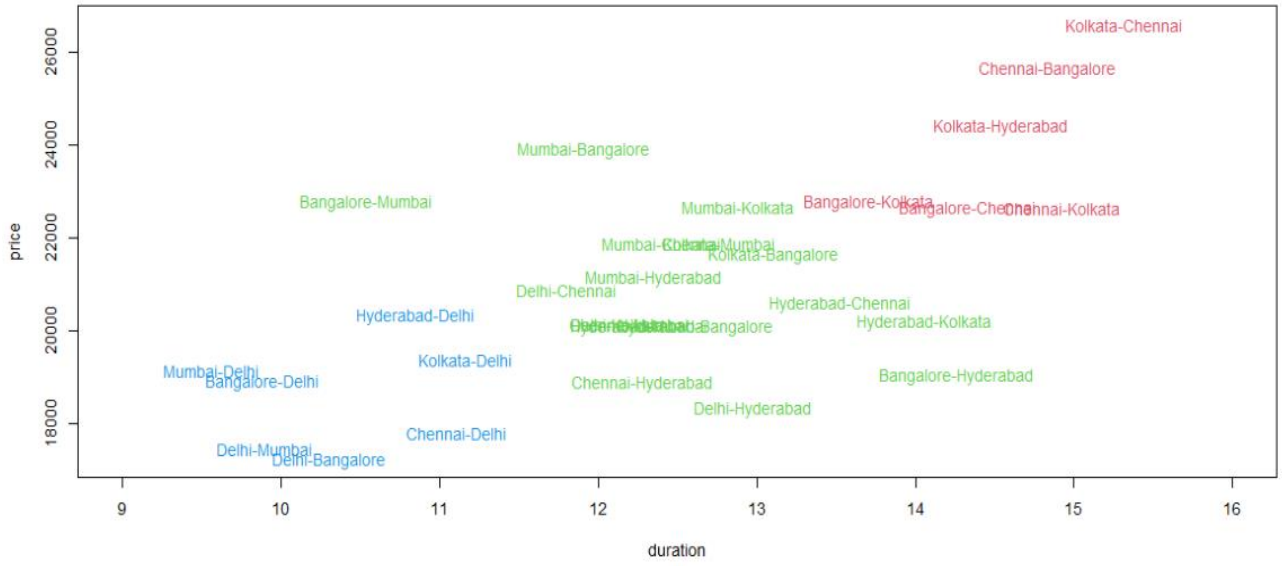


I scale the 2 features, compute the distance matrix and plot the dendrogram. I decide to cut high as I do not want to have many clusters upon which I cannot make easily any interpretation. I obtain 3 clusters and I plot again “price” and “duration” identifying every point with the related cluster.

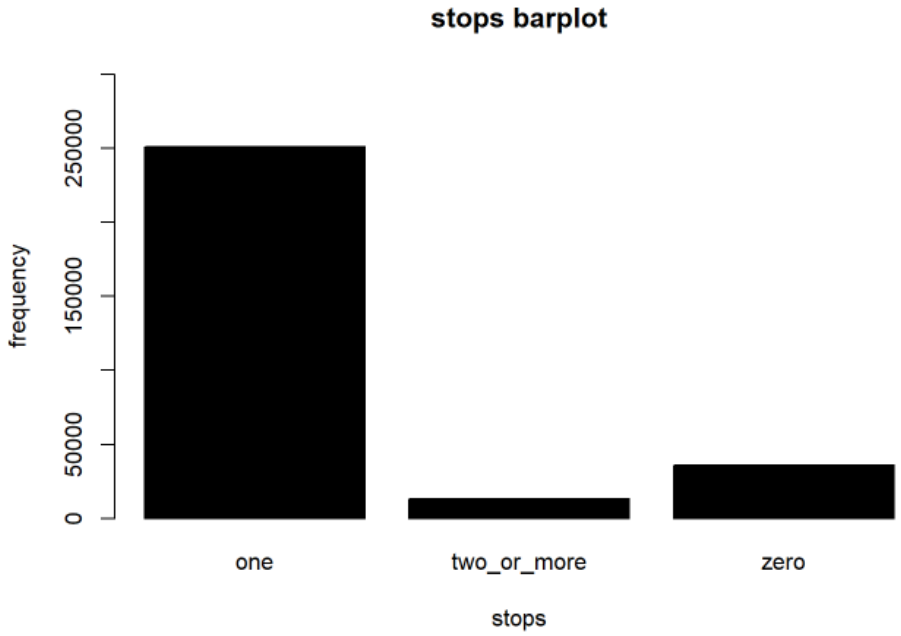


With this plot it is possible to say that there is a homogenous group of very long flights with prices as low as very short ones, so there is not a clear positive correlation between the two variables. Maybe this happens because a very long flight may have some stops and so this lowers the price. Anyway, I

think that it may be more interesting to see if it is possible to add to this analysis the different routes that are present in the dataset. My idea is to divide the routes in 3 clusters and plot them according to their average price and duration. In this way I may be able to extend the above conclusions to the routes. I create a new data set with 30 rows, where each row is an air route (in the data set there are 6 cities and every observation has a different "source_city" and "destination_city" value, in fact we have 30 rows), with the average duration and the average price. I scale again the two continuous feature and apply the K-means clustering with 3 clusters. I eventually plot the result.



Interestingly, if the route is taken into consideration, it is evident the positive correlation that I supposed before. The air routes are divided into the 3 clusters that reflects that, if the flight is longer, the price is higher. It is not really straightforward to understand why this happens. Perhaps the routes in red have just a small number of observations that have a low price, but on average they are costly. After all it seems to me that it makes more sense that for the same route the price varies more than the duration, even if for the latter the number of stops may have an impact. Maybe usually the number of stops is zero or one (this is consistent with the bar plot of this variable), but, when it increases, the price and the duration both go up. Further analysis may be needed.



5.CONCLUSIONS

I tried to understand which features are the most important to the determine the price of an air flight ticket, as from now on the prices should raise, and which model should be used to predict it in the most precise way. As expected, the random forest model is the best one for the latter purpose, but for predicting the price as a continuous variable also the tree predictor does a good job and has the advantage of being very easy to interpret. I decided to try to predict also if a price is low or high (a data-driven idea of low and high prices) and I saw that the two tasks share the same most important variable, the class, but also differs from the point of view of the other variables, especially how many days in advance the ticket is bought. I tried also to apply some unsupervised learning techniques to tackle the ambiguous relationship between the price and the duration of the flight, but the result is unclear.

I think that I found some interesting points about the price of an air flight. Obviously, a similar data set with Italian or European data would have been better, but anyway in the contemporary world it is easy to see common patterns across very different areas, considering also how much it is interconnected.