Home exercise 2

By Javier Rodriguez

Table of contents

1. Supervised
   1. Introduction
   2. Methods and Results
   3. Discussion
   4. Conclusion
2. Unsupervised
   1. Introduction
   2. Methods and Results
   3. Discussion
   4. Conclusion
3. Supervised
   1. Introduction

Before predicting or performing some other type of analysis, as we always do, I cleaned the data. I separated the date variable into individual variables (year, month, day and hour), I changed some data types, removed variable “X” and “datetime” since we split. For the train dataset we had to remove also “casual” and “registered” variables. After checking the correlation between the numeric variables, as it can be seen in Figure 1, I removed the variable “atemp” because it is almost a duplicate of the variable “temp” and it would mean multicollinearity to the model we ran.



Figure 1. Correlation heatmap

I then looked for outliers and removed those values that fell outside the boxplot min and max.

* 1. Methods and Results

Having cleaned the data I proceeded with finding the best possible prediction algorithms. The first algorithm is linear regression, but since some variables were not statistically significant and the result of the predictions were not great as it was seen in the bonus point predictions where this exact same model was used (147 RMSE, being RMSE explained in Figure 2).

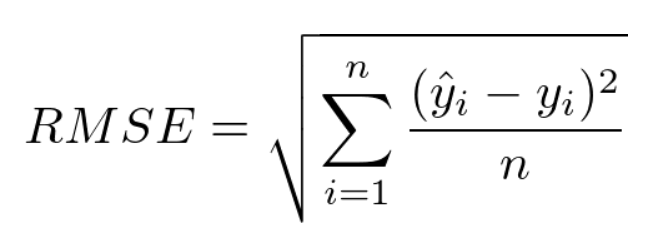


Figure 2. Root Mean Squared Error Formula

Y hat being sub I the prediction and y sub I the actual value of the target value and n the sample size.

With the goal of having a lower error, I performed Lasso regression and three variables were penalized to 0, which are: “holiday”, “day” and “windspeed”. I then removed these three variables in a new data frame and did a linear model again, but this time without including these three variables. The error dropped to 120 RMSE. The predictions plotted against the actual values can be seen in Figure 3.

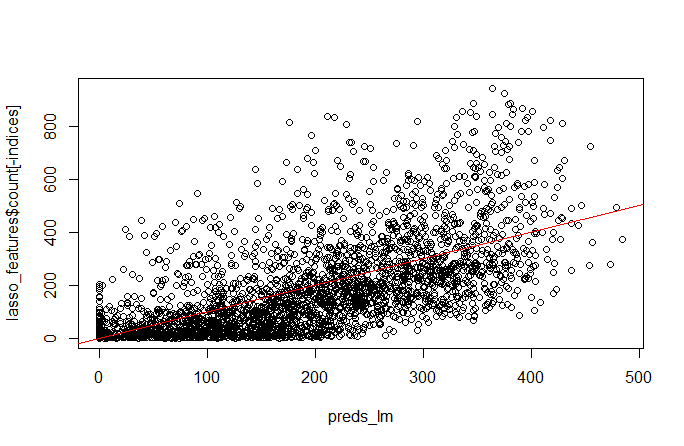


Figure 3. Predictions result for linear regression.

For all predictions I converted negative “count” predictions as 0. Since “count” is out target variable. And cannot have a negative value.

I also would like to mention that in some cases I did not do k-fold cross-validation with the goal of getting a better sense of the predictions because the for loops took a really long time, if it was for me I would have done it every time.

The second used algorithm is Random forest. My goal was to find the best parameters of number of trees and variables used to build those trees so the error was the lowest possible. For that, I created my own tune function with a for loop that performed cross-validation with many different parameters and save the errors to later check which was performing best. I used all variables, also including the ones that Lasso removed before since in this case more bad learners can lead to a better model. In the end, the best tune parameter was 10,000 trees and 11 variables to build each tree. The error was only 41 RMSE (for the bonus points it was 119). And how the plot looks like can be seen in Figure 4. I also checked which variables were the most important to predict our target variable which are “hour” with a big difference and then “temp”. all the rest were close to each other; these results can be found in Figure 5 and they are evaluated by R checking the purity of the variables in each tree.

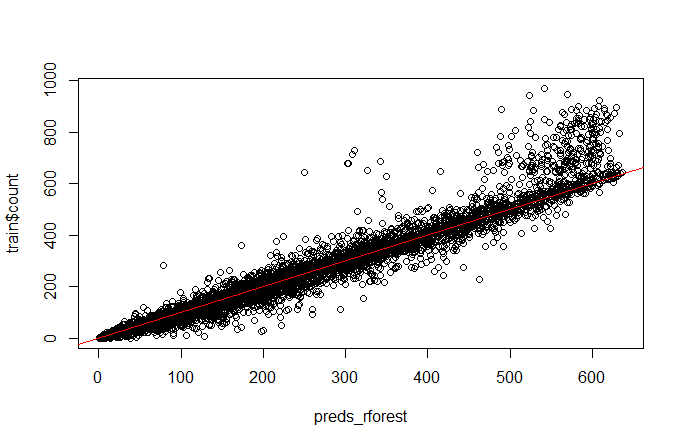


Figure 4. Prediction result for Random forest

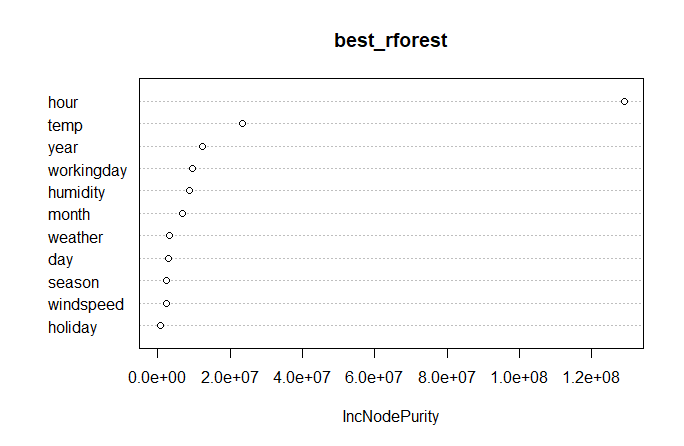


Figure 5. Most important variables to predict “count” by Random forest

Lastly, I tried some Support Vector algorithms and tuned them as they performed best. The best one was SVM with radial kernel having as values for cost 1 and 0.01 as gamma. The error was 120 RMSE and the plot can be seen in Figure 6. I used the lasso variables in this case since the model performed better this way.

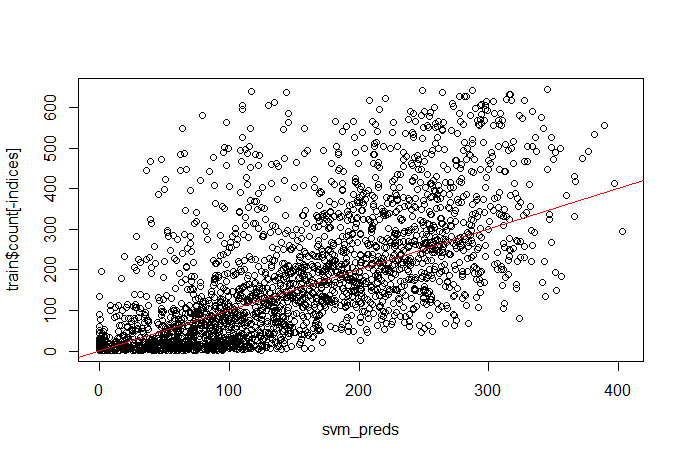


Figure 6. Prediction result for Support Vector Machine

* 1. Discussion

The best model, with a difference, is the random forest, it has the lowest error and by looking at the plot we can see how well it fits the data, it has problems with the extremely high count values; especially above 500.

With linear regression it is more spread and the main issue is that it predicts higher values than the actual ones for some reason. We can see how dense the points are below the trend line where they should be on the line.

In the case of SVM, I do not understand why but in the cross-validation I achieved 77 RMSE but when I perform the model with those parameters alone I get a higher error and the issue is the same as with linear, it predicts higher values than the real ones.

* 1. Conclusion

Random forest is the way to go for this dataset if our goal is prediction. It is the one with the lowest RMSE and if the value of our target variable is below 500 it will predict very closely. The other two algorithms predict above the real values.

1. Unsupervised
   1. Introduction

The goal was now to cluster the data to find groups of similar data

* 1. Methods and Results

I performed hierarchical clustering with complete linkage and k-means clustering.

The result for the first one, having calculated the Euclidean distance on the numerical data ("temp", "humidity", "windspeed") after scaling and then checking the clusters assigned in the original data.

cluster temp humidity windspeed

1 17.15771 69.19050 7.697839

2 15.72739 51.57004 20.935496

3 28.97906 59.50927 13.649494

We see one cluster with the highest temperature, humidity levels in the middle and windspeed also in the middle; this cluster could be named “hot summer ride”. We also have the lowest temperature one with the lowest humidity and the highest windspeed; we could name that one “freezing tornado ride” and lastly we have “sweaty but nice ride” since it is the one with the lowest wind, intermediate temperature and highest humidity.

For k-means clustering the results are very similar as it can be seen in the resulting clusterings:

cluster temp humidity windspeed

1 17.17899 69.31513 7.67794

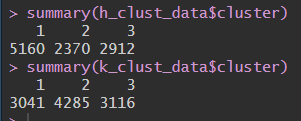
2 15.77712 51.83162 20.77148

3 28.96165 59.43480 13.59555

* 1. Discussion

The values of the clusters are very similar and we would not change the name or almost anything depending on the method we use for clustering, this way we can say the patters are in the data and not generated by the algorithm.

Even though we can see differences between both methods on how many values are in each cluster, we know the number of the cluster is irrelevant, but the amounts are not even close.



If we create a new variable for “count” so it is easier to visualize, having “low” if the value is below 40 rides (1st quartile), “medium” if it is between 40 and 271 (3rd quartile) and above that “high” we see these results linked t the clusters:

high low medium

1 693 1357 2069

2 427 390 1030

3 949 305 1062

They are mixed together, but for cluster 1 which is “sweaty but nice ride” the number of bikes rented is medium, low and high; for the second cluster which is “freezing tornado ride” is medium, high and low and the cluster 3 which is “hot summer ride” we have medium, high and low.

* 1. Conclusion

Having found the three clusters we could target in different ways each one of them with marketing strategies for instance. The two methods of clustering give very similar results, so the patterns are on the data and not created by us.

We can see the worst days are the high-humidity ones and the coldest and hottest are close to each other, but hotter days are somewhat better.