RANDOM FOREST

Ignacio Rodriguez Pereira

DIAMONDS

A data frame with **53940 rows** and **10 variables**

- Price
- Carat
- Cut
- Color
- Clarity
- X
- Y
- Z
- Depth
- Table

DIAMONDS

- Price - price in US dollars (\$326--\$18,823)
- Carat - weight of the diamond (0.2--5.01)
- Cut - quality of the cut (Fair, Good, Very Good, Premium, Ideal)
- Color - diamond colour, from D (best) to J (worst)
- Clarity - a measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))
- X - length in mm (0--10.74)
- Y - width in mm (0--58.9)
- Z - depth in mm (0--31.8)
- Depth - total depth percentage = z / mean(x, y) = 2 * z / (x + y) (43--79)
- Table - width of top of diamond relative to widest point (43--95)

IMPORTANT PARAMETERS

- **Number of trees** : Number of trees used for building the model.
- **Number of independent variables** : Total of variables used for building the model.
- **Mtry** : Number of variables to possibly split at in each node.
- **Variable importance mode (Impurity)**: When a tree is built, the decision about which variable to split at each node uses a calculation of the Gini impurity.
- **Splitrule (Variance)** : Splitting rule.
- **OOB prediction error (mse)** : Out Of Bag prediction error
- **RMSE** : root mean square error is a measure of the differences between values predicted by a model and the values observed.

CODE

```
21 ########################################
22 set.seed(1010)
23 ################
24 percentage = 0.8
25 ################
26 train <- sample(nrow(data), percentage*nrow(data), replace = FALSE)
27 TrainSet <- data[train,]
28 ValidSet <- data[-train,]
29 #########
```

```
29 #########
30 numTree = 30
31 numVar = 4
32 maxDepth = 20
33 minNodeSize = 5
34 #########
35 model <- ranger(price~carat+cut+color+clarity+depth+table, data = TrainSet,
36                          num.trees = numTree,
37                          mtry = numVar,
38                          max.depth = maxDepth,
39                          min.node.size = minNodeSize
40                  importance = "impurity")
41 model
42 ########################################
```

CODE

```
42 ########################################
43 pred <- predict(model, validSet)$predictions
44 rmse(validSet$price, pred)
45 ########################################
```

```
45 ####################################
46 ggplot(  ) +
47   geom_jitter( aes(x = data$carat, y = data$price, color = "#b83b5e", alpha = 0.5)) +
48   geom_jitter( aes(x = validSet$carat, y = pred, color = "#f08a5d",  alpha = 0.5)) +
49   labs(x = "Carat", y = "Price", color = "", alpha = 'Transperency') +
50   scale_color_manual(labels = c("Real", "Predicted"), values = c("#b83b5e","#f08a5d"))
51 ####################################
52 imps <- data.frame(var = model$forest$independent.variable.names,
53                    imps = model$variable.importance/max(model$variable.importance))
54 imps %>%
55   ggplot(aes(imps, x = reorder(var, imps))) +
56   geom_point(size = 3, colour = "#b83b5e") +
57   coord_flip() +
58   labs(x = "Predictors", y = "Importance scores") +
59   theme_bw(18)
60 ########################################
```

CODE

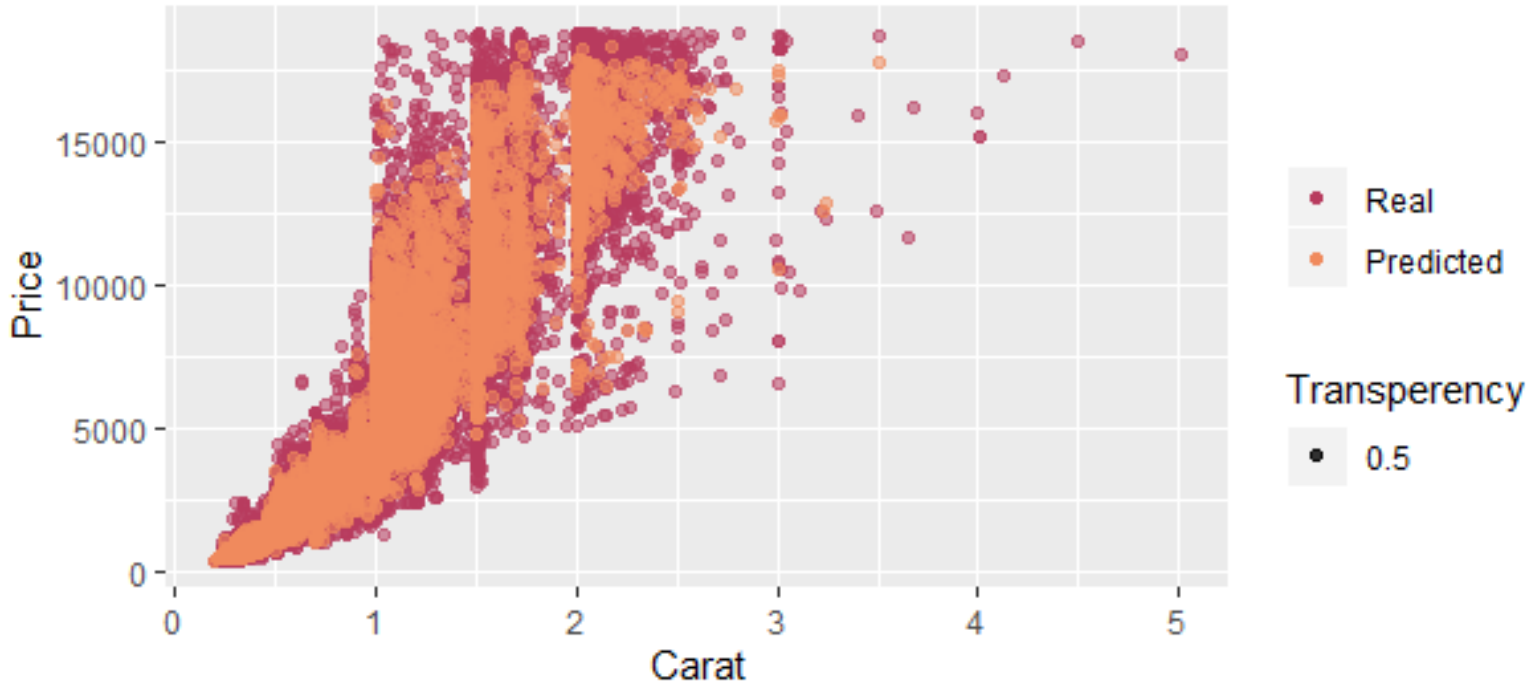
```r
########################################
col1 <- colorRampPalette(c("#f9ed69", "#f08a5d" , "#b83b5e", "#6a2c70"))
C <- cor(data)
corrplot(C, method = "circle", col = col1(100))
########################################
########################################
y=c()
i=1
for (i in 2:6) {
  model <- ranger(price ~ carat+cut+color+clarity+depth+table,
                  data = TrainSet,
                  num.trees = 30,
                  mtry = i,
                  importance = "impurity")
  print(i)
  pred <- predict(model, ValidSet)$predictions
  y[i-1] = rmse(ValidSet$price, pred)
}
x <- c(2,3,4,5,6)
ggplot() + geom_point(aes(x = x, y = y, colour = "#f08a5d", size = 4)) +
  labs(x = "Mtry", y = "RMSE", color = "", size = "") +
  theme(legend.position="none")
########################################################################
```

FIRST RANDOM FOREST MODEL

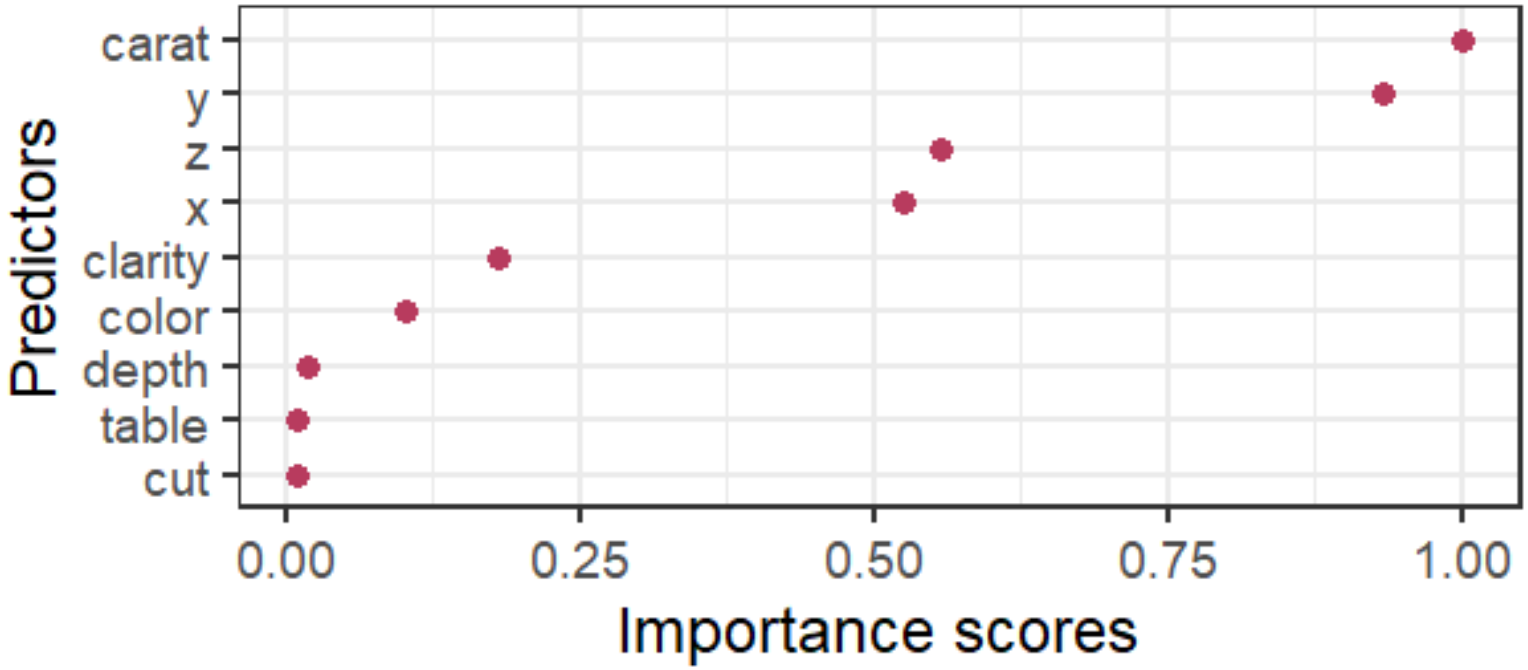
Type:	Regression
Number of trees:	10
Sample size:	37758
Number of independent variables:	9
Mtry:	3
Target node size:	5
Variable importance mode:	impurity
Splitrule:	variance
OOB prediction error (MSE):	448917.9
R squared (OOB):	0.9718941

RMSE : 573.6342

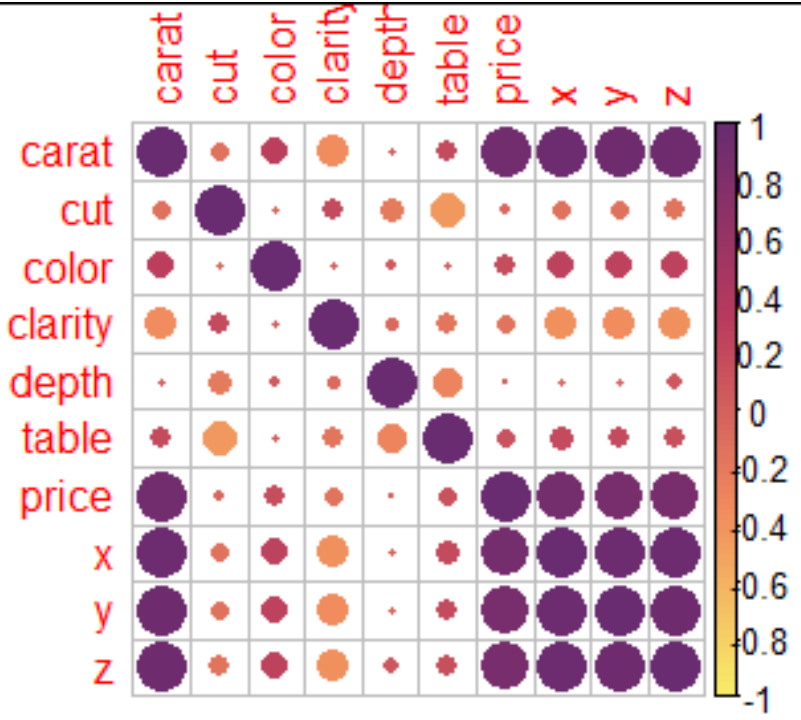
FIRST RANDOM FOREST MODEL



FIRST RANDOM FOREST MODEL



FIRST RANDOM FOREST MODEL



Correlation

The importance of others is significantly reduced since effectively the impurity they can remove is already removed by the first feature.

As a consequence, they will have a lower reported importance. This is not an issue when we want to use feature selection to reduce overfitting, since it makes sense to remove features that are mostly duplicated by other features, But *when interpreting the data*, it can lead to the incorrect conclusion that one of the variables is a strong predictor while the others in the same group are unimportant, while actually they are very close in terms of their relationship with the response variable.

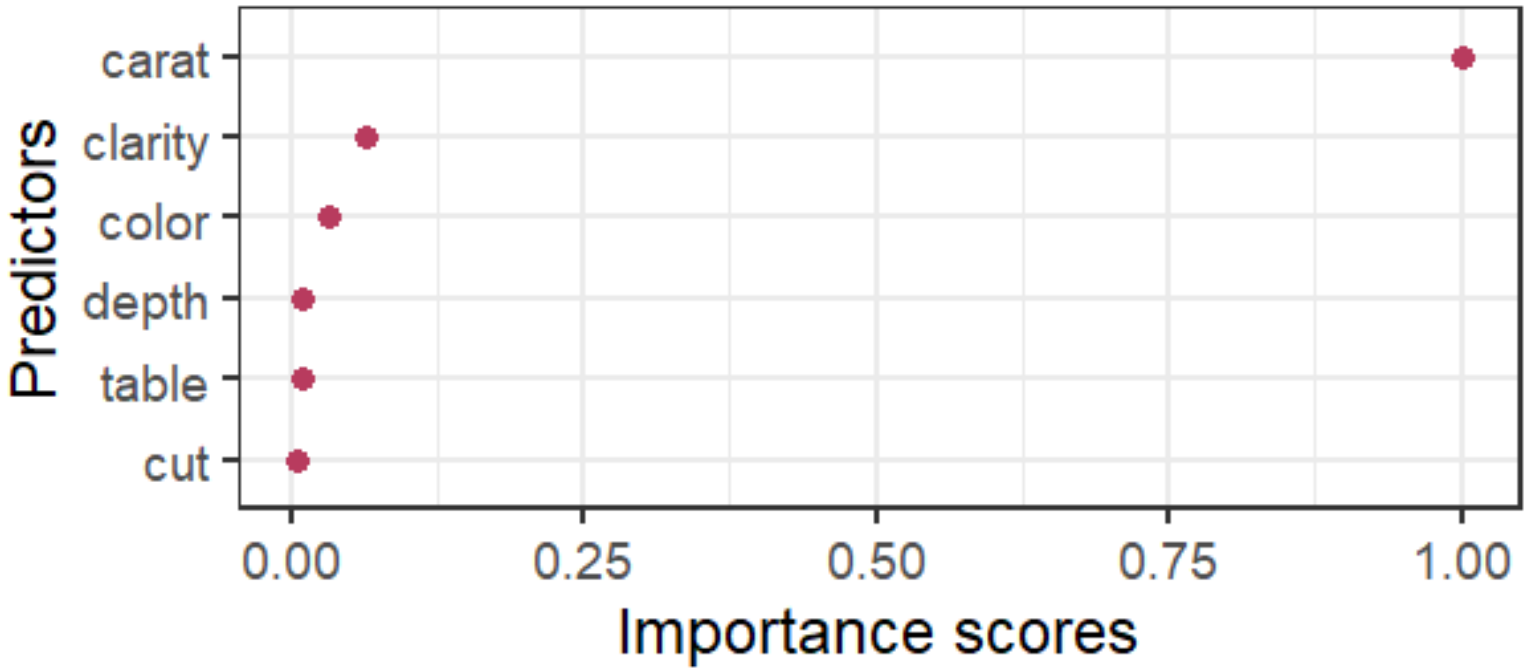
The effect of this phenomenon is reduced thanks to random selection of features at each node creation, but in general the effect is not removed completely

SECOND RANDOM FOREST MODEL

Type:	Regression
Number of trees:	10
Sample size:	37758
Number of independent variables:	6
Mtry:	3
Target node size:	5
Variable importance mode:	impurity
Splitrule:	variance
OOB prediction error (MSE):	402407.6
R squared (OOB):	0.974806

RMSE : 558.8155

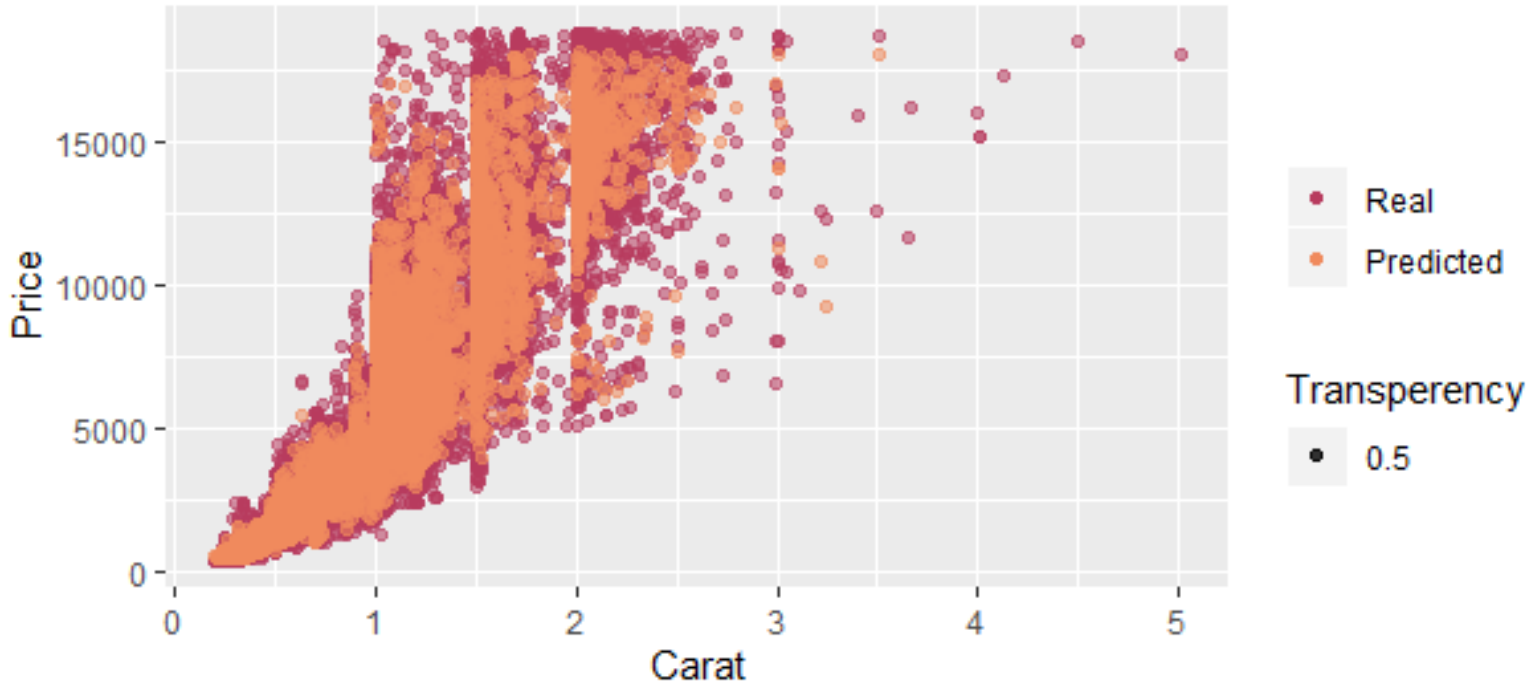
SECOND RANDOM FOREST MODEL



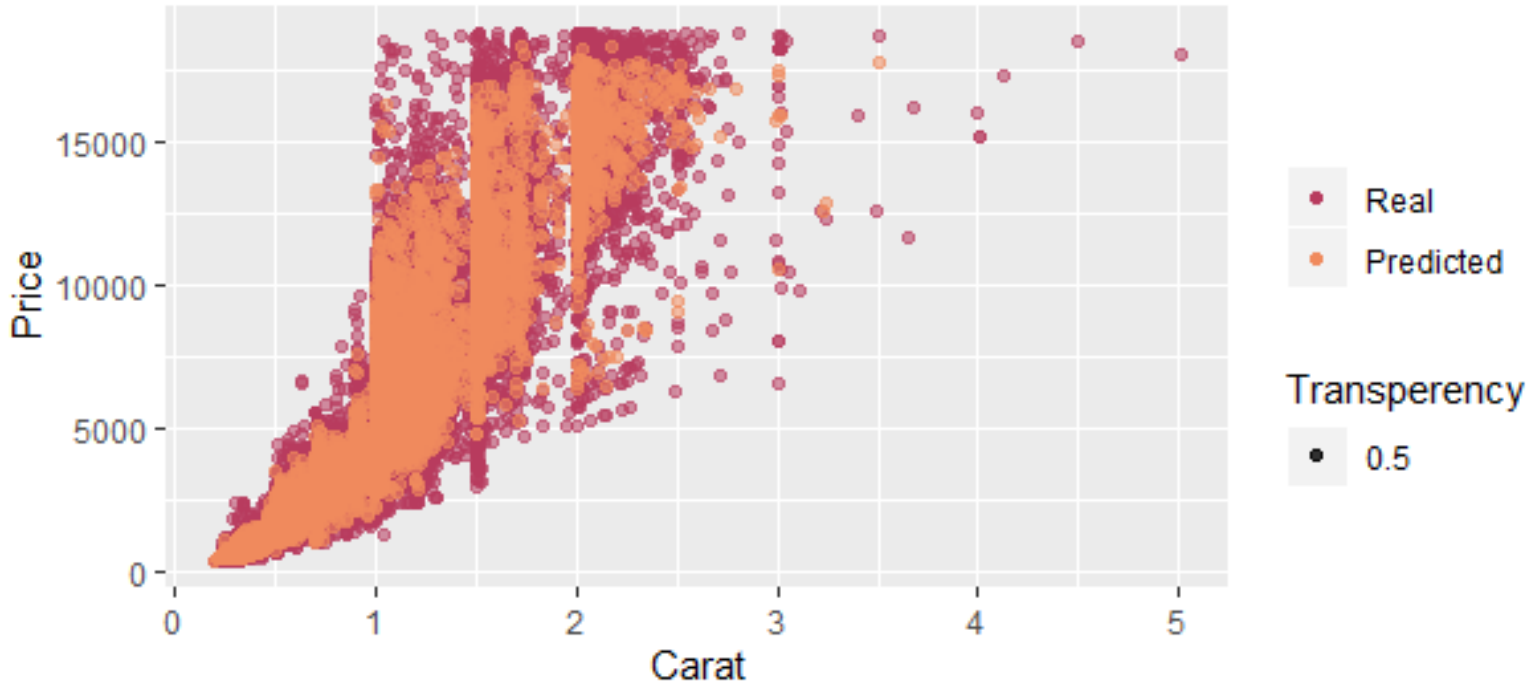
Importance

Random forest use at each node a random subset of the variables. The ones providing the split with best entropy (or other criteria) will be kept, while others will be discarded and possibly tested in a subsequent / different node. If a variable does not provide any information about the split will not be used in the final model. From the other point of view, a variable can become informative after a split on another variable, and, possibly, this variable will be useful at prediction time.

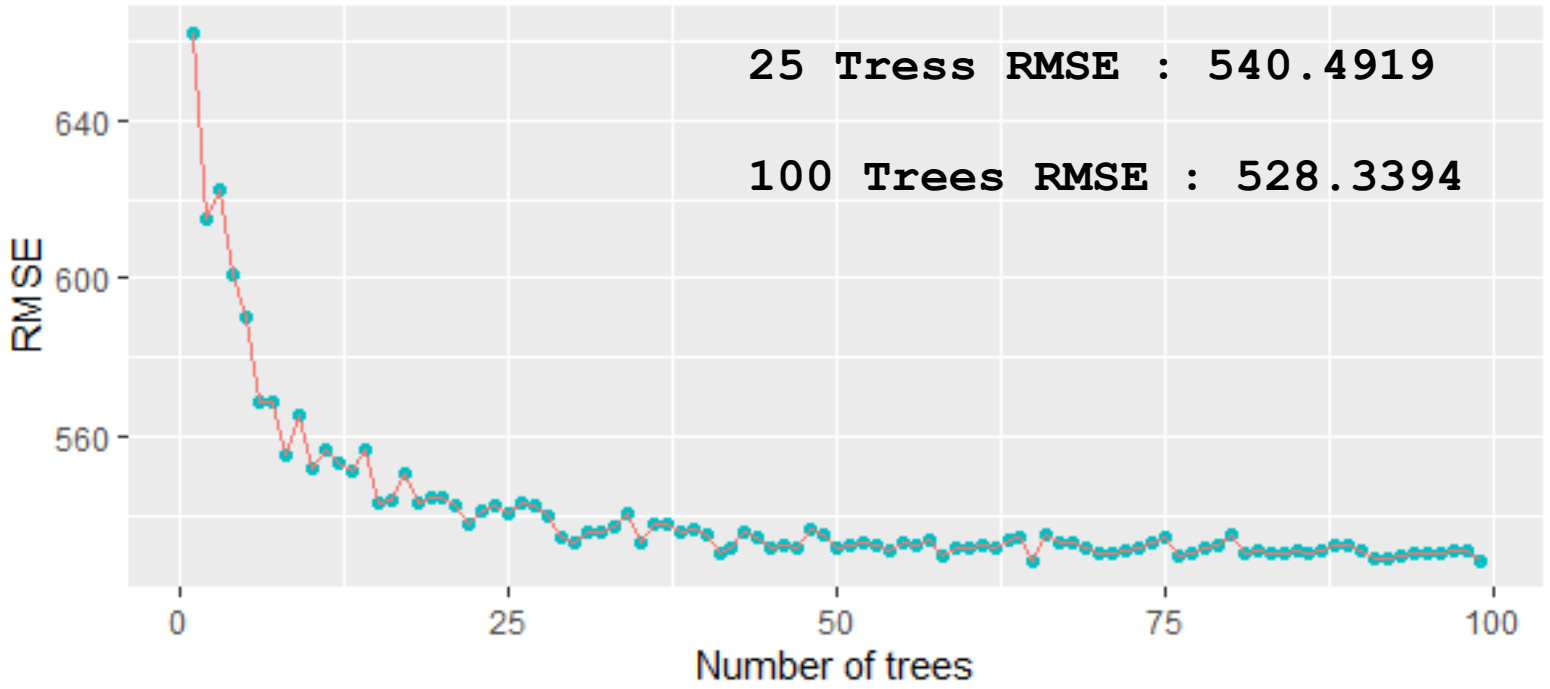
SECOND RANDOM FOREST MODEL



FIRST RANDOM FOREST MODEL



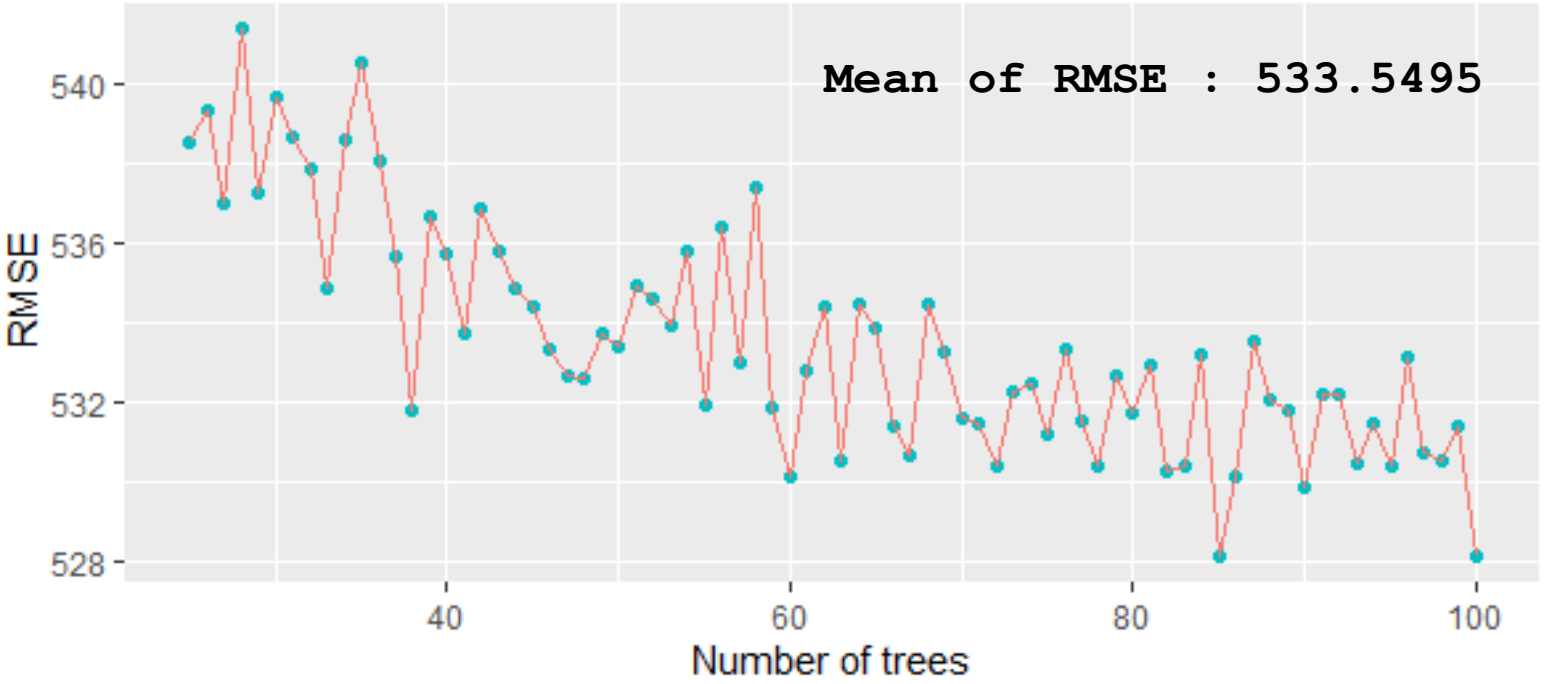
TESTING DIFFERENT CONFIGURATIONS



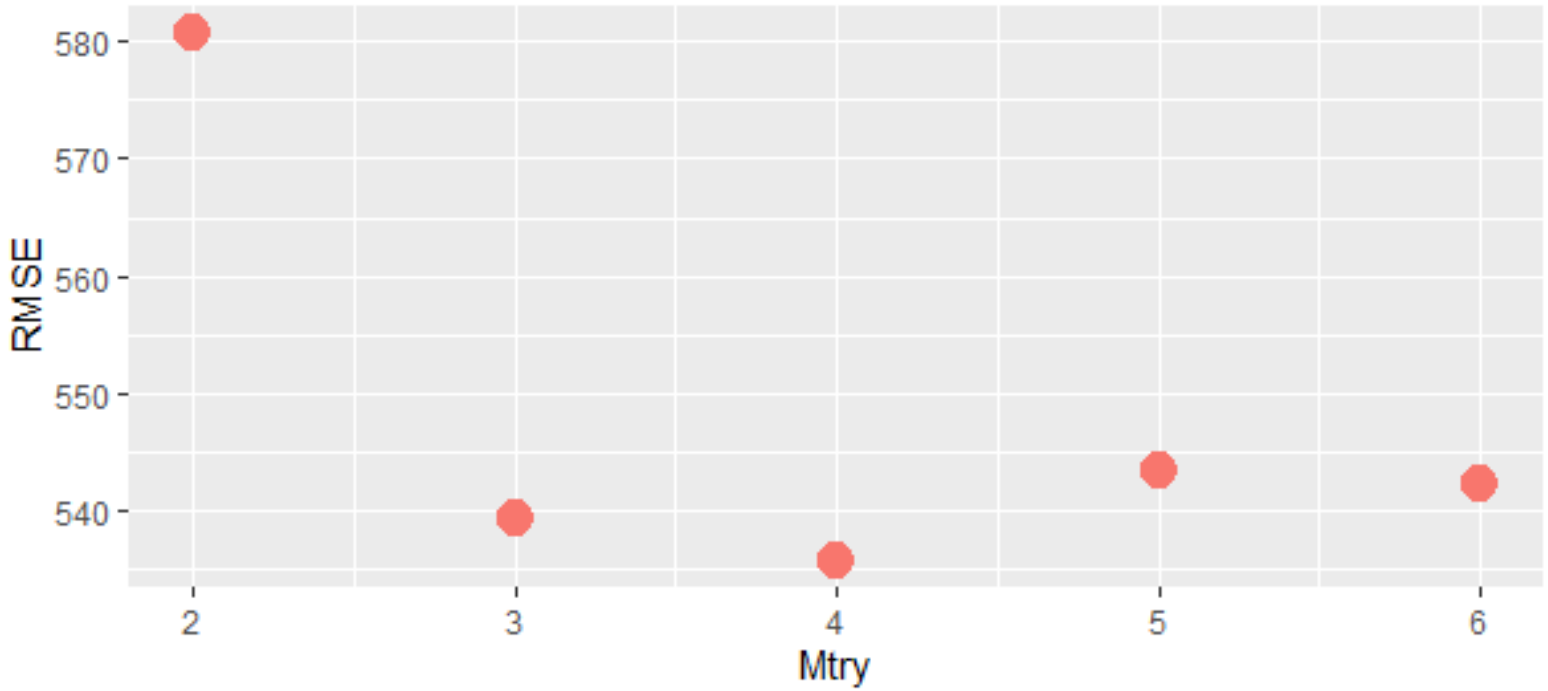
Number of trees

In general, the more trees you use the better the results. However, the improvement decreases as the number of trees increases. At a certain point the benefit in prediction performance from learning more trees will be lower than the cost in computation time for learning these additional trees.

TESTING DIFFERENT CONFIGURATIONS



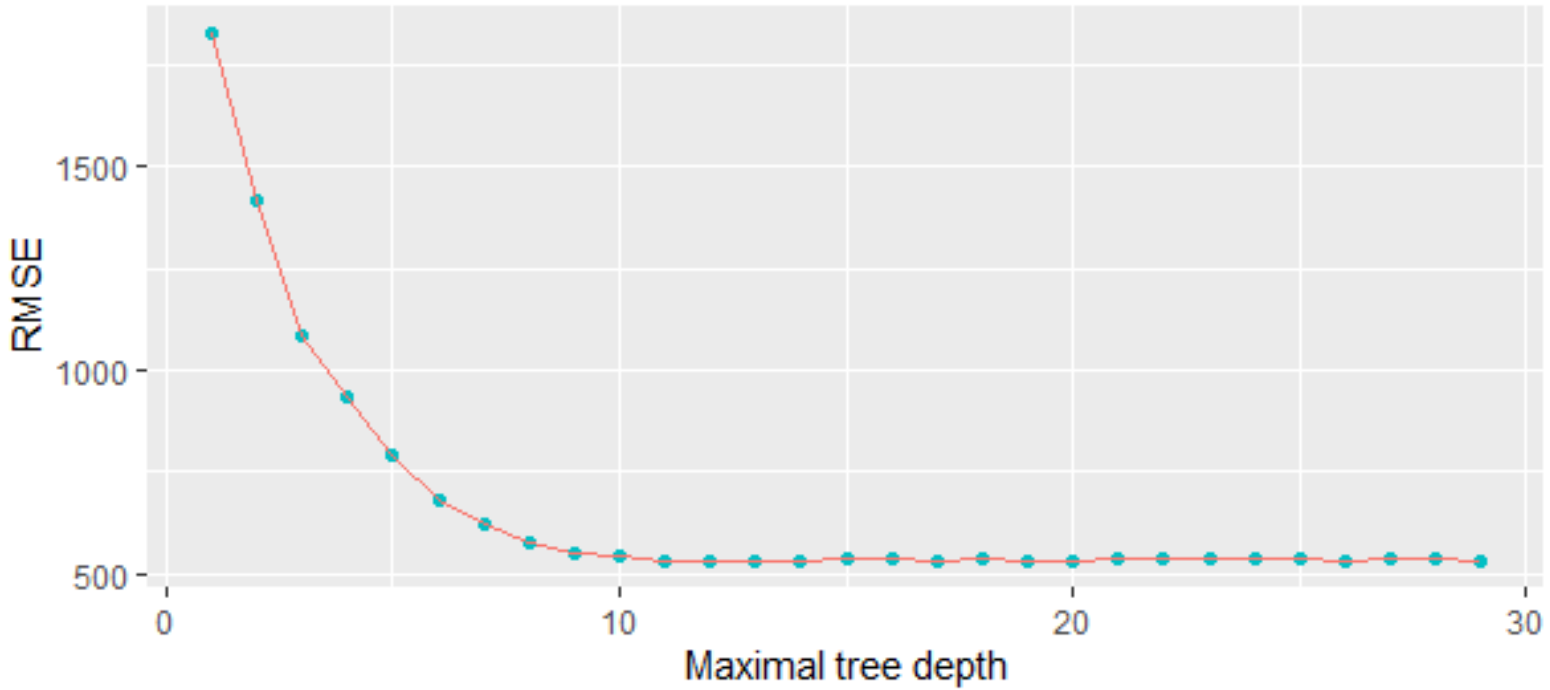
TESTING DIFFERENT CONFIGURATIONS



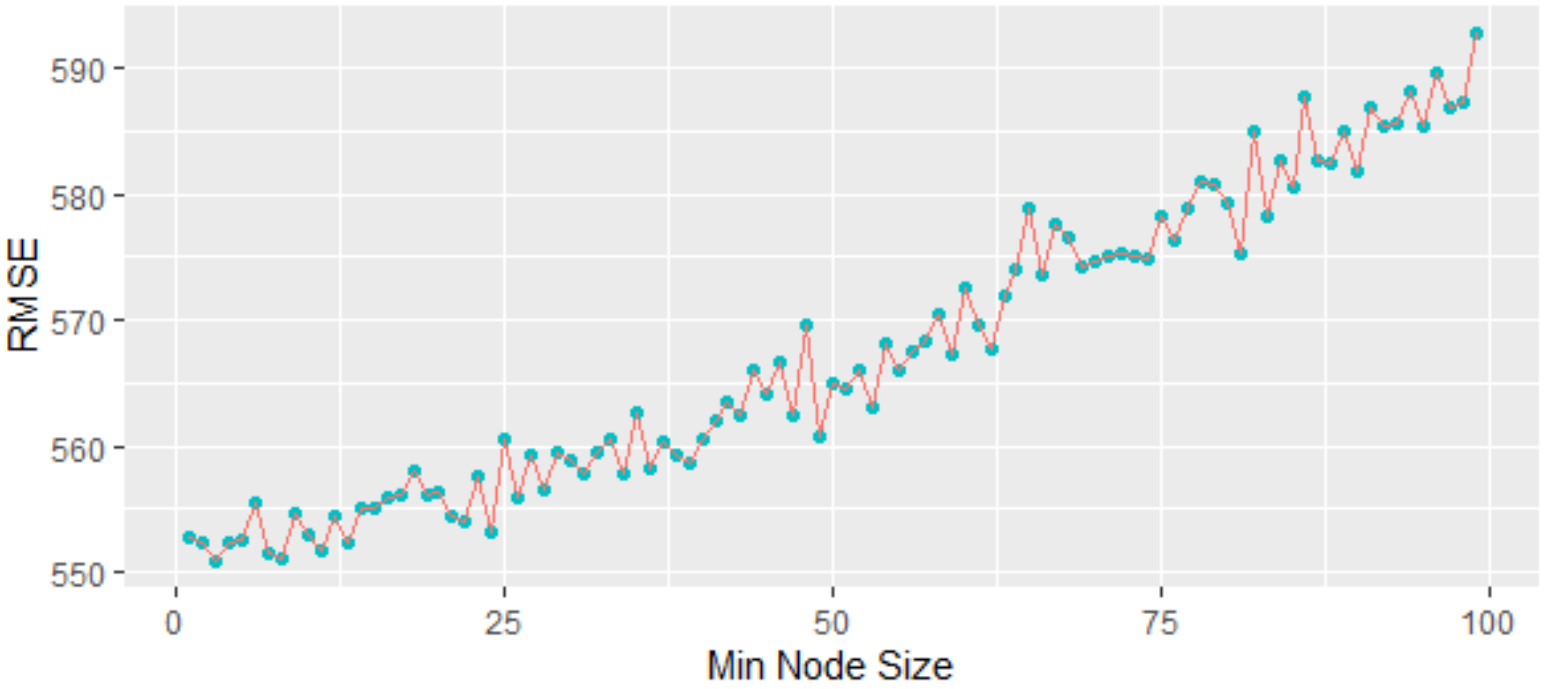
MTRY

Random forest grow in each split in a tree, the algorithm randomly selects mtry variables from the set of predictors available. Hence when forming each split *a different random set of variables* is selected within which the best split point is chosen.

TESTING DIFFERENT CONFIGURATIONS



TESTING DIFFERENT CONFIGURATIONS



Min node size

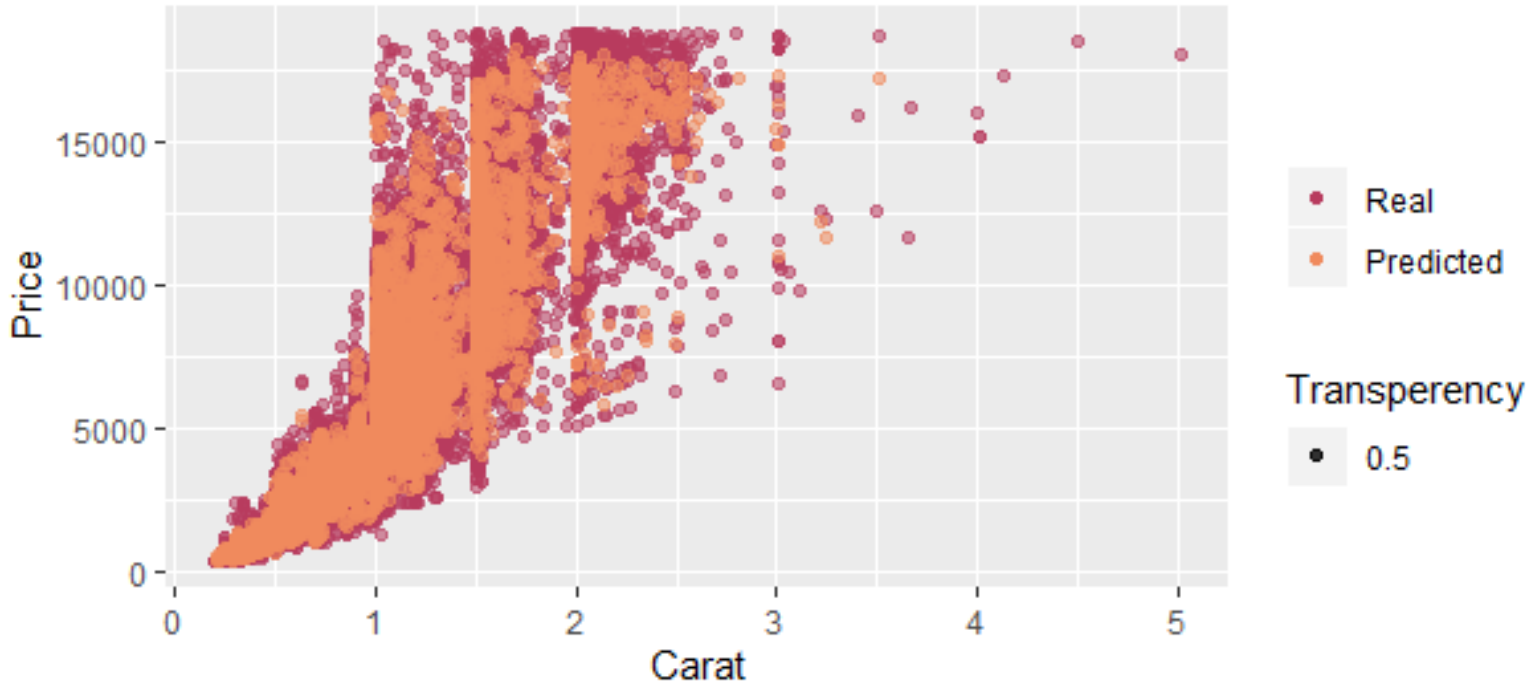
Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown.

FINAL RANDOM FOREST MODEL

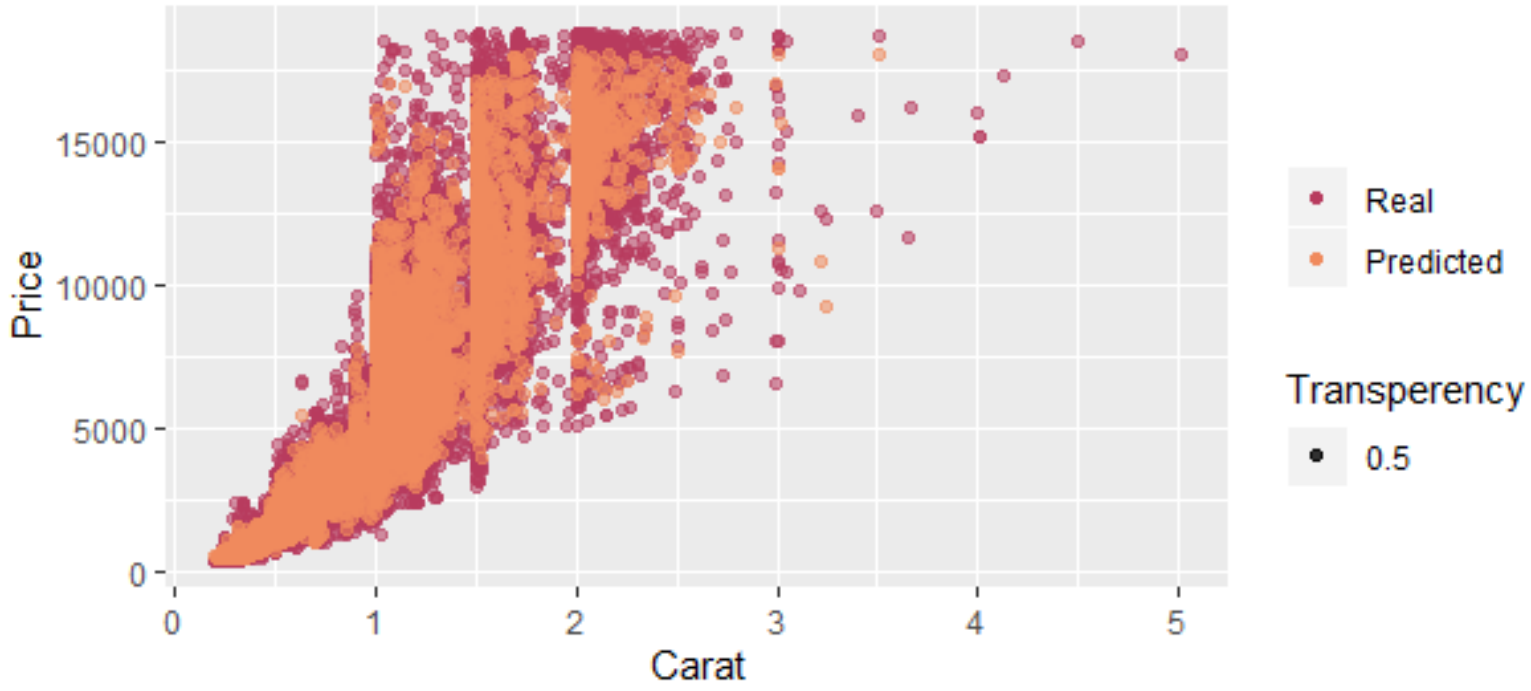
```
Type:                                  Regression    NumTrees = 30
Number of trees:                       30            Mtry  = 4
Sample size:                           37758         maxDepth = 20
Number of independent variables:       6             minNodeSize = 5
Mtry:                                  4
Target node size:                      5
Variable importance mode:              impurity
Splitrule:                             variance
OOB prediction error (MSE):            319452.7
R squared (OOB):                       0.9799997
```

RMSE : 535.9543

FINAL RANDOM FOREST MODEL



SECOND RANDOM FOREST MODEL



TESTING MORE CONFIGURATIONS

Type:	Regression
Number of trees:	30
Sample size:	37758
Number of independent variables:	6
Mtry:	4
Target node size:	5
Variable importance mode:	impurity
Splitrule:	extratrees
OOB prediction error (MSE):	323004.9
R squared (OOB):	0.9797773

RMSE : 530.122

Type:	Regression
Number of trees:	30
Sample size:	37758
Number of independent variables:	6
Mtry:	4
Target node size:	5
Variable importance mode:	impurity
Splitrule:	maxstat
OOB prediction error (MSE):	600085.2
R squared (OOB):	0.9624298

RMSE : 711.8259

TESTING MORE CONFIGURATIONS

Type:	Regression
Number of trees:	30
Sample size:	26970
Number of independent variables:	6
Mtry:	4
Target node size:	5
Variable importance mode:	impurity
Splitrule:	variance
OOB prediction error (MSE):	334335.8
R squared (OOB):	0.9787962

RMSE : 553.7359

Type:	Regression
Number of trees:	30
Sample size:	48546
Number of independent variables:	6
Mtry:	4
Target node size:	5
Variable importance mode:	impurity
Splitrule:	variance
OOB prediction error (MSE):	304904.8
R squared (OOB):	0.9807989

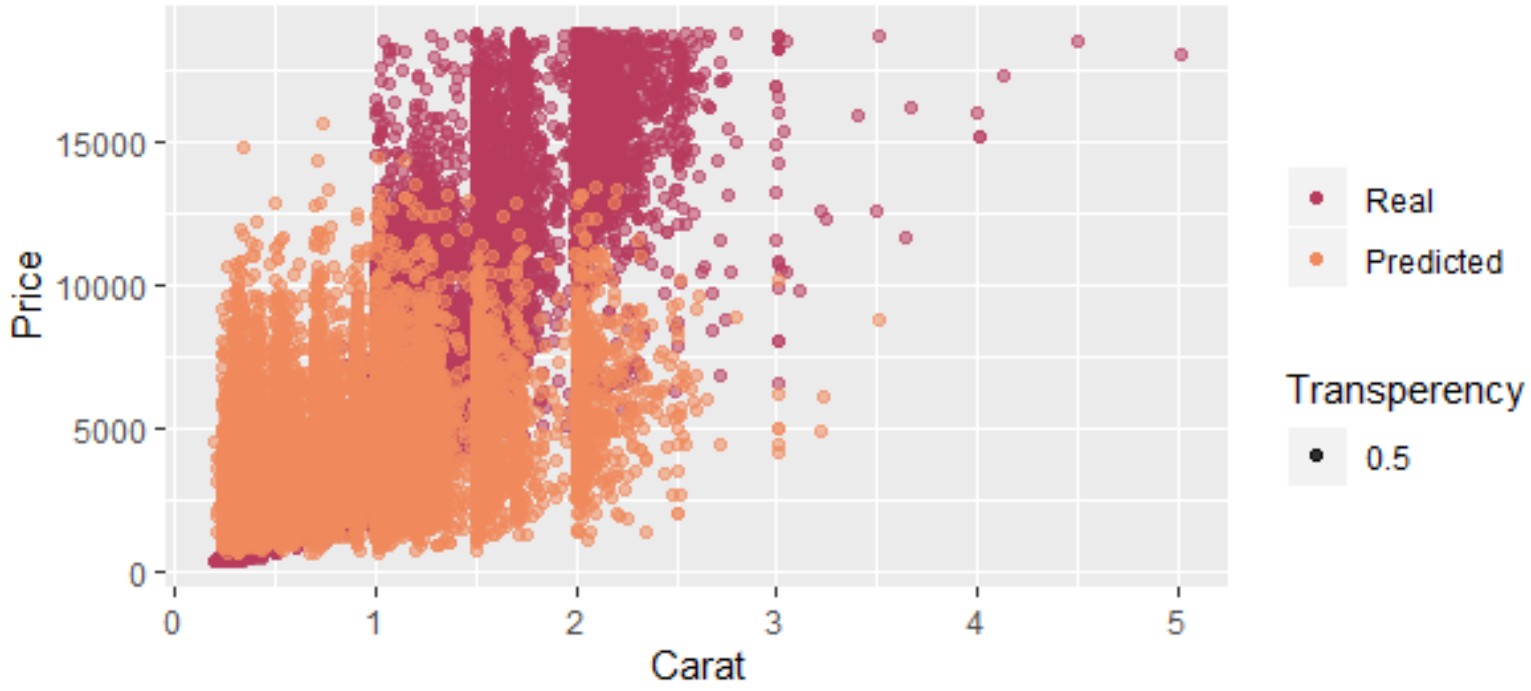
RMSE : 542.6433

TESTING MORE CONFIGURATIONS

```
Type:                             Regression
Number of trees:                  30
Sample size:                      37758
Number of independent variables:  5
Mtry:                             4
Target node size:                 5
Variable importance mode:         impurity
Splitrule:                        variance
OOB prediction error (MSE):       15751517
R squared (OOB):                  0.01382692
```

RMSE : 3931.741

TESTING MORE CONFIGURATIONS

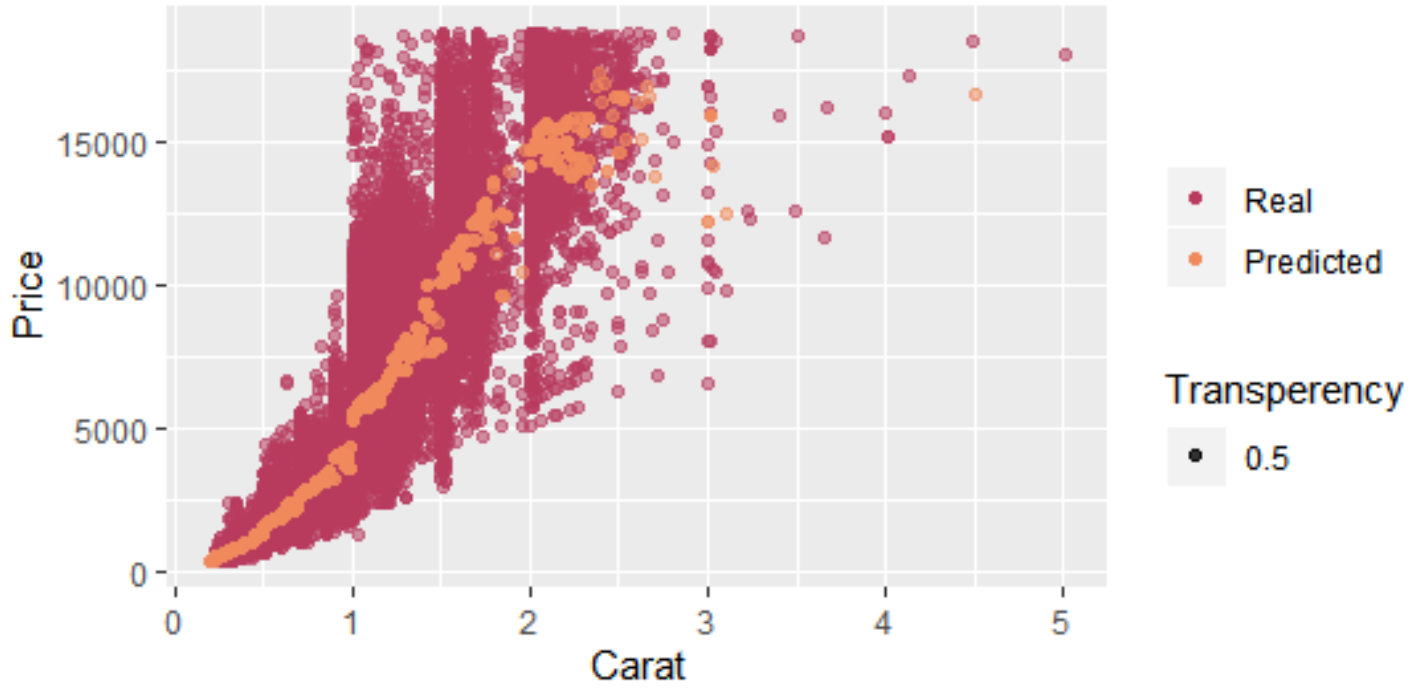


TESTING MORE CONFIGURATIONS

```
Type:                                Regression
Number of trees:                     30
Sample size:                         43152
Number of independent variables:     1
Mtry:                                1
Target node size:                    5
Variable importance mode:            impurity
Splitrule:                           variance
OOB prediction error (MSE):          2054518
R squared (OOB):                     0.8716175
```

RMSE : 1416.038

TESTING MORE CONFIGURATIONS



LIBRARIES

- `library(ggplot2)`
- `library(corrplot)`
- `library(Metrics)`
- `library(dplyr)`
- `library(ranger)`

CODE : https://paste.ofcode.org/35AZkSBWXecRwwSxwuCRhPN (Expires in one week)

GITHUB REPOSITORY : https://github.com/BestMilleLire/RandomForest