

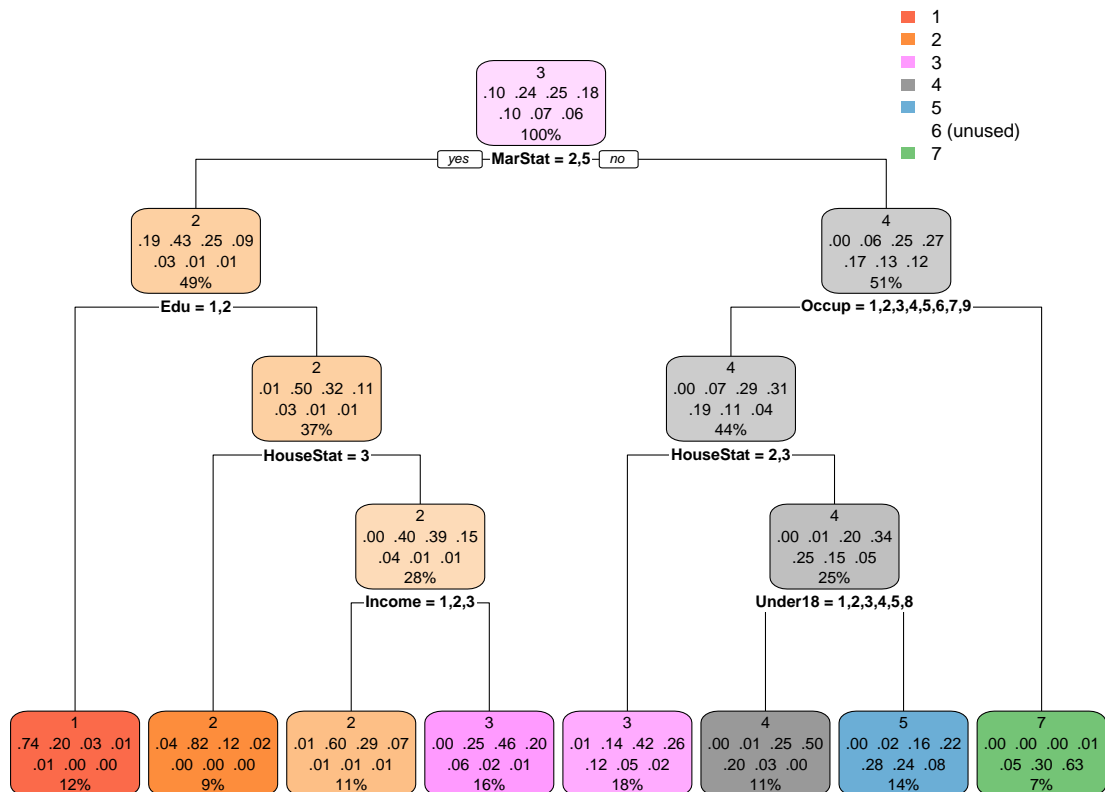
# STATS305B - HW1

Thibaud Bruyelle

5/1/2020

## Question 1

```
library(rpart)
library(rpart.plot)
# Question 1
age <- read.csv("handout/Data/age_stats315B.csv", header= T)
for (i in 1:ncol(age)){
  age[,i] <- as.factor(age[,i])
}
tree <- rpart(age ~ ., data = age, method = "class")
rpart.plot(tree)
```



```
summary(tree)
```

```
## Call:
## rpart(formula = age ~ ., data = age, method = "class")
##   n= 8710
##
##           CP nsplit rel error   xerror   xstd
## 1 0.12821297    0 1.0000000 1.0000000 0.006179661
## 2 0.08629131    1 0.8717870 0.8717870 0.006791529
## 3 0.06058752    2 0.7854957 0.7854957 0.007024341
```

```

## 4 0.03855569      3 0.7249082 0.7249082 0.007111824
## 5 0.02210832      4 0.6863525 0.6863525 0.007136257
## 6 0.01269890      6 0.6421359 0.6433599 0.007135251
## 7 0.01000000      7 0.6294370 0.6308140 0.007129350
##
## Variable importance
##      Edu      Occup      MarStat HouseStat      DualInc      Income      Under18      TypeHome
##      20      18      17      15      11      9      6      2
## Persons
##      2
##
## Node number 1: 8710 observations,      complexity param=0.128213
## predicted class=3 expected loss=0.7504018 P(node) =1
## class counts: 828 2083 2174 1556 870 636 563
## probabilities: 0.095 0.239 0.250 0.179 0.100 0.073 0.065
## left son=2 (4268 obs) right son=3 (4442 obs)
## Primary splits:
##      MarStat splits as RLRRL,      improve=555.6238, (0 missing)
##      HouseStat splits as RRL,      improve=511.5696, (180 missing)
##      Edu splits as LLRRRR,      improve=499.3220, (76 missing)
##      Occup splits as RLRRRLRL, improve=403.8474, (0 missing)
##      Income splits as LRRRRRRRR, improve=354.8621, (338 missing)
## Surrogate splits:
##      DualInc splits as LRR,      agree=0.832, adj=0.658, (0 split)
##      HouseStat splits as RLL,      agree=0.724, adj=0.437, (0 split)
##      Occup splits as RLLRRLRL, agree=0.699, adj=0.386, (0 split)
##      Income splits as LLLRRRRRR, agree=0.663, adj=0.312, (0 split)
##      Edu splits as LLRRRR,      agree=0.600, adj=0.184, (0 split)
##
## Node number 2: 4268 observations,      complexity param=0.08629131
## predicted class=2 expected loss=0.5707591 P(node) =0.4900115
## class counts: 816 1832 1063 376 111 44 26
## probabilities: 0.191 0.429 0.249 0.088 0.026 0.010 0.006
## left son=4 (1050 obs) right son=5 (3218 obs)
## Primary splits:
##      Edu splits as LLRRRR,      improve=562.3539, (38 missing)
##      HouseStat splits as RRL,      improve=310.8559, (92 missing)
##      Under18 splits as RLLLLLLLLL, improve=295.2636, (0 missing)
##      Income splits as LRRRRRRRR, improve=261.9943, (178 missing)
##      Occup splits as RLRRRLRL, improve=229.7293, (0 missing)
## Surrogate splits:
##      Under18 splits as RRLLLRLRL, agree=0.806, adj=0.210, (38 split)
##      Occup splits as RRRRLRLRL, agree=0.758, adj=0.016, (0 split)
##
## Node number 3: 4442 observations,      complexity param=0.06058752
## predicted class=4 expected loss=0.7343539 P(node) =0.5099885
## class counts: 12 251 1111 1180 759 592 537
## probabilities: 0.003 0.057 0.250 0.266 0.171 0.133 0.121
## left son=6 (3806 obs) right son=7 (636 obs)
## Primary splits:
##      Occup splits as LLLLLLRL, improve=322.67010, (0 missing)
##      Under18 splits as RLLLLLLLLL, improve=145.86890, (0 missing)
##      HouseStat splits as RLL,      improve=101.76260, (88 missing)
##      DualInc splits as RLR,      improve= 94.68224, (0 missing)

```

```

##      Persons splits as RLLLLLLL, improve= 93.83632, (122 missing)
##      Surrogate splits:
##      MarStat splits as L-LR-, agree=0.862, adj=0.033, (0 split)
##      Under18 splits as LLLLLLLL, agree=0.857, adj=0.002, (0 split)
##
## Node number 4: 1050 observations
## predicted class=1 expected loss=0.2590476 P(node) =0.1205511
## class counts: 778 214 33 11 8 5 1
## probabilities: 0.741 0.204 0.031 0.010 0.008 0.005 0.001
##
## Node number 5: 3218 observations, complexity param=0.02210832
## predicted class=2 expected loss=0.4972032 P(node) =0.3694604
## class counts: 38 1618 1030 365 103 39 25
## probabilities: 0.012 0.503 0.320 0.113 0.032 0.012 0.008
## left son=10 (817 obs) right son=11 (2401 obs)
## Primary splits:
## HouseStat splits as RRL, improve=174.1940, (83 missing)
## Edu splits as --LLRR, improve=165.2050, (27 missing)
## Occup splits as RLRRRLRL, improve=164.6654, (0 missing)
## Persons splits as RLLLLLLL, improve=159.3146, (153 missing)
## Income splits as LLRRRRRRR, improve= 97.9920, (114 missing)
## Surrogate splits:
## Under18 splits as RLLLLLLR-R, agree=0.769, adj=0.100, (83 split)
## Persons splits as RRRLRLRL, agree=0.746, adj=0.011, (0 split)
##
## Node number 6: 3806 observations, complexity param=0.03855569
## predicted class=4 expected loss=0.6918024 P(node) =0.436969
## class counts: 11 249 1111 1173 726 402 134
## probabilities: 0.003 0.065 0.292 0.308 0.191 0.106 0.035
## left son=12 (1596 obs) right son=13 (2210 obs)
## Primary splits:
## HouseStat splits as RLL, improve=90.42640, (70 missing)
## Under18 splits as RLLLLLLL-, improve=73.85239, (0 missing)
## TypeHome splits as RRLRL, improve=48.33088, (0 missing)
## LiveBA splits as LLLLR, improve=36.00277, (381 missing)
## Persons splits as RRLLLLLL, improve=35.43460, (87 missing)
## Surrogate splits:
## TypeHome splits as RRLRL, agree=0.801, adj=0.526, (70 split)
## Income splits as LLLLLRRRR, agree=0.691, adj=0.262, (0 split)
## MarStat splits as R-LR-, agree=0.645, adj=0.153, (0 split)
## DualInc splits as LRR, agree=0.641, adj=0.142, (0 split)
## Occup splits as RLLRL-L, agree=0.632, adj=0.121, (0 split)
##
## Node number 7: 636 observations
## predicted class=7 expected loss=0.3663522 P(node) =0.07301952
## class counts: 1 2 0 7 33 190 403
## probabilities: 0.002 0.003 0.000 0.011 0.052 0.299 0.634
##
## Node number 10: 817 observations
## predicted class=2 expected loss=0.1811506 P(node) =0.09380023
## class counts: 31 669 95 15 2 4 1
## probabilities: 0.038 0.819 0.116 0.018 0.002 0.005 0.001
##
## Node number 11: 2401 observations, complexity param=0.02210832

```

```

## predicted class=2 expected loss=0.604748 P(node) =0.2756602
## class counts:      7   949   935   350   101   35   24
## probabilities: 0.003 0.395 0.389 0.146 0.042 0.015 0.010
## left son=22 (976 obs) right son=23 (1425 obs)
## Primary splits:
##   Income splits as LLLLLLLLLL, improve=96.70470, (68 missing)
##   Occup splits as RLRRLLRLR, improve=85.93692, (0 missing)
##   Edu splits as --LLRR, improve=83.32219, (19 missing)
##   Persons splits as RRLLLLLL, improve=38.69356, (117 missing)
##   LiveBA splits as LLLRR, improve=19.48925, (221 missing)
## Surrogate splits:
##   Occup splits as RRRLLLLRL, agree=0.700, adj=0.263, (68 split)
##   Edu splits as --LRRR, agree=0.625, adj=0.077, (0 split)
##   TypeHome splits as RRRLL, agree=0.611, adj=0.043, (0 split)
##   Under18 splits as RRRRLRLR-R, agree=0.595, adj=0.004, (0 split)
##
## Node number 12: 1596 observations
## predicted class=3 expected loss=0.5814536 P(node) =0.1832377
## class counts:      9   216   668   416   184   73   30
## probabilities: 0.006 0.135 0.419 0.261 0.115 0.046 0.019
##
## Node number 13: 2210 observations, complexity param=0.0126989
## predicted class=4 expected loss=0.6574661 P(node) =0.2537313
## class counts:      2    33   443   757   542   329   104
## probabilities: 0.001 0.015 0.200 0.343 0.245 0.149 0.047
## left son=26 (975 obs) right son=27 (1235 obs)
## Primary splits:
##   Under18 splits as RLLLLLR-L-, improve=78.61524, (0 missing)
##   Persons splits as RRLLLRLR, improve=40.62282, (46 missing)
##   DualInc splits as RLR, improve=20.79117, (0 missing)
##   LiveBA splits as RLLLR, improve=15.60334, (219 missing)
##   MarStat splits as L-LR-, improve=12.78300, (0 missing)
## Surrogate splits:
##   Persons splits as RRLLLLLRL, agree=0.833, adj=0.622, (0 split)
##   Ethnic splits as RLLRLRRR, agree=0.573, adj=0.032, (0 split)
##   Occup splits as RRLRLR-R, agree=0.566, adj=0.015, (0 split)
##   TypeHome splits as RRLRR, agree=0.561, adj=0.004, (0 split)
##
## Node number 22: 976 observations
## predicted class=2 expected loss=0.397541 P(node) =0.1120551
## class counts:      5   588   285   65   12   9   12
## probabilities: 0.005 0.602 0.292 0.067 0.012 0.009 0.012
##
## Node number 23: 1425 observations
## predicted class=3 expected loss=0.5438596 P(node) =0.1636051
## class counts:      2   361   650   285   89   26   12
## probabilities: 0.001 0.253 0.456 0.200 0.062 0.018 0.008
##
## Node number 26: 975 observations
## predicted class=4 expected loss=0.4974359 P(node) =0.1119403
## class counts:      1    14   241   490   192   34   3
## probabilities: 0.001 0.014 0.247 0.503 0.197 0.035 0.003
##
## Node number 27: 1235 observations

```

```
## predicted class=5 expected loss=0.7165992 P(node) =0.141791
## class counts:      1      19      202      267      350      295      101
## probabilities: 0.001 0.015 0.164 0.216 0.283 0.239 0.082

# pruned <- prune(tree, cp = 0.012699)
# rpart.plot(pruned)
```

The plot of the tree shows 7 splits and 15 nodes (leaves included). We also notice that there is not prediction for people who are between 55 and 64 years old. It seems relevant that the marital status is a great split to classify since the population gets married or dies at approximately the same age. Then education and occupation are also good splits because they gather people with the same age, and reached almost everybody.

(a)

Yes, some surrogates variables were used during the construction. Let us give an explanation of the output from `summary(tree)` : for the root node that use the marital status to do the split, there were no missing values so there was no need to use any surrogate variables. Conversely, for node 2 (**Education**) there were 38 missing values and the surrogate variable **Under18** was used to handle these missing values.

(b)

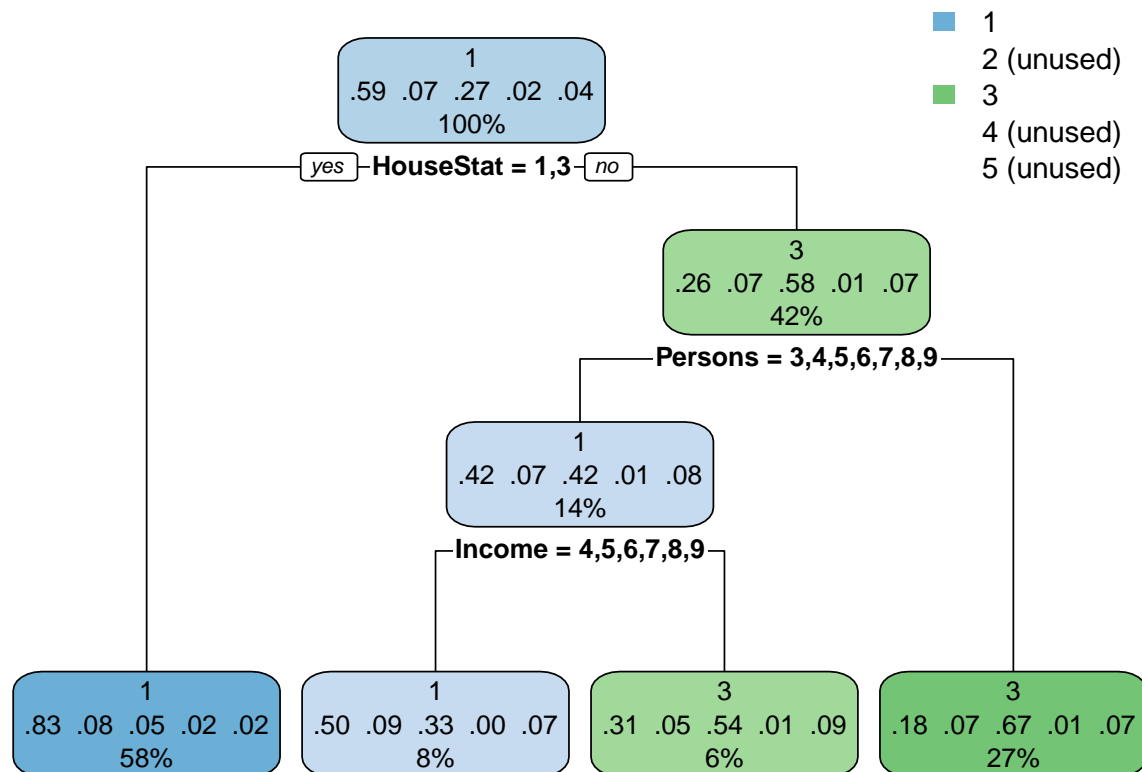
```
new = matrix(ncol = 13, nrow =1)
new = data.frame(new)
new[1,] = as.factor(c(6,3,1,5,6,1,1,1,4,0,2,7,3))
colnames(new) <- colnames(age)[-1]
print(predict(tree, new))
```

```
##           1           2           3           4           5           6           7
## 1 0.005122951 0.602459 0.2920082 0.06659836 0.01229508 0.009221311 0.01229508
```

Therefore my predicted age is 18 - 24 years olds which is great because I am 23.

## Question 2

```
housetype <- read.csv("handout/Data/housetype_stats315B.csv", header =T)
for (i in 1:ncol(housetype)){
  housetype[,i] <- as.factor(housetype[,i])
}
housetype_tree <- rpart(TypeHome ~. , data = housetype, method = "class")
rpart.plot(housetype_tree)
```



```
fit.val <- predict(housetype_tree, housetype[,-1], type = "class")
table(housetype$TypeHome, fit.val)
```

```
##      fit.val
##      1      2      3      4      5
## 1 4707      0    612      0      0
## 2  475      0    199      0      0
## 3  510      0   1944      0      0
## 4  132      0     30      0      0
## 5  181      0    223      0      0
```

```
1 - (sum(diag(table(housetype$TypeHome, fit.val))))/length(fit.val)
```

```
## [1] 0.2620659
```

```
# pruned <- prune(housetype_tree, cp = 0.1)
# rpart.plot(pruned)
# fit.val_pruned <- predict(pruned, housetype[,-1], type="class")
# c = table(actual = housetype$TypeHome, fitted = fit.val_pruned)
# 1 - (sum(diag(c)))/length(fit.val_pruned)
```

The model returns an optimal tree with 4 splits and 9 nodes. It is surprising to notice that only class 1 and 3 are predicted by the model. It is probably due to the fact that 86 % of the data represents these two classes but this is still a weakness of the model. We can also see that the prediction of class 1 is straightforward in most cases. Indeed, when people fall in the ‘Own’ category, they are directly predicted as “House”, which makes sense. The misclassification error with the optimal tree is : 0.2620659. As an alternative example, with a pruned tree that provides 2 splits, the misclassification error becomes 0.2763786.

### Question 3

- Reason 1 : If our model is not trained enough, we will underfit the data and consequently, when trying to do predictions on another set of data, we will get large errors. In other words, there is an important bias in our model.
- Reason 2 : Conversely, if our model is too much trained, it will overfit the data used to build it. Therefore, when testing it, we will also get large errors because our model will do predictions by only using the training dataset structure and relationships. This error is due to a high variance in our model that is responsible for high sensitivity to small fluctuations.

### Question 4

We cannot choose a prediction function among all possible functions for complexity purposes. We need to put constraints and restrictions when we search for the best predictor because otherwise it would be beyond our computational abilities.

### Question 5

The target function  $f^*$  can be defined as :  $\arg \min_f \mathbb{E}(l(f(X), Y))$  where  $l$  is the loss function. In other words, the target function is the function that, for a given measure of the loss, will minimize the error in our predictions. The accuracy of a target function depends on the constraints of the class of functions we are working on and also to the nature of the problem.

### Question 6

It cannot always be a good surrogate for prediction risk. Indeed, prediction error on the training data can be very low but if our model is overfitted, then the error on the actual population will be much higher. As an example, classification trees are prone to high variance so they easily overfit. If there is no overfitting and underfitting, it might be an option to use the empirical risk classification.

### Question 7

Let us assume that the misclassification loss  $l_{l,k} = l(c_l, c_k)$  is such that :  $\boxed{l_{l,k} = I(k \neq l)}$ . Then the misclassification risk when predicting  $c(\underline{X}) = c_k$  is given by  $r_k = \mathbb{E}_{Y, \underline{X}}(l(Y, c_k)) = \sum_{l=1}^K l_{l,k} \mathbb{P}(\{Y = c_l | \underline{X}\}) = 1 - \mathbb{P}(Y = c_k | \underline{X}) = \mathbb{P}(Y \neq c_k | \underline{X})$ . The latter is equal to the error rate. Then Bayes optimal prediction rule satisfies :  $\boxed{k^* = \arg \min_{1 \leq k \leq K} \mathbb{P}(Y \neq c_k | \underline{X})}$  with the optimal classifier  $c^*(\underline{X}) = k^*$ .

### Question 8

It is not always true because wrong estimates of  $(\mathbb{P}(Y \neq c_k | \underline{X}))_{1 \leq k \leq K}$  can lead to a low error rate (by choosing the wrong optimal rule and then computing the wrong error rate). In this case, we would be mistaken if we thought that our estimations of these probabilities are good.

### Question 9

We are always looking for models with small bias (when we do too restrictive assumptions) and small variance (high sensitivity to small fluctuations usually caused by overfitting). However models with small variance usually have a high bias and on the contrary, models with small bias have a high variance. As a consequence, there is tradeoff between these two effects that we want to minimize.

## Question 10

Surrogate variables are meant to mimic the split of a primary variable so it makes no sense to use them as primary split variables because the split is not computed with respect to the same criteria. A good surrogate variable may not behave as a good primary variable. Sometimes a variable can be both a primary split variable and a surrogate split variable. We will notice that the way this variable is splitted in both cases is different because it is not meant to have the same functionalities.

## Question 11

Let us define  $\alpha_N := \sum_{i=1}^N [y_i^2 - 2y_i \sum_{m=1}^M c_m I(\mathbf{x}_i \in \mathcal{R}_{\uparrow}) + \sum_{1 \leq l, m \leq M} c_m c_l I(\mathbf{x}_i \in \mathcal{R}_{\uparrow}) I(\mathbf{x}_i \in \mathcal{R}_{\downarrow})]$ . Then we have for  $m \in \{1, \dots, M\}$ :  $\frac{\partial \alpha_N}{\partial c_m} = 0 - 2 \sum_{i=1}^N [I(\mathbf{x}_i \in \mathcal{R}_{\uparrow}) + c_m I(\mathbf{x}_i \in \mathcal{R}_{\downarrow})]$ . Finally since  $\hat{c}_m$  satisfies  $\frac{\partial \alpha_N}{\partial c_m} = 0$ , we have the result.

## Question 12

After such a split,  $F(\mathbf{x})$  becomes  $G(\mathbf{x}) = \sum_{l=1}^{M+1} c_m I(\mathbf{x} \in \mathcal{R}_m)$ . The difference of estimated risk is :

$$\hat{r}_F - \hat{r}_G = \sum_i (y_i - \hat{F}(\mathbf{x}_i))^2 - (y_i - \hat{G}(\mathbf{x}_i))^2 \quad (1)$$

We can notice that :  $\hat{c}_{l,r} = \bar{y}_{l,r}$  and that  $\hat{c}_m = \frac{1}{n}(n_l \bar{y}_l + n_r \bar{y}_r)$ . So finally we can rewrite (1) as :

$$\sum_{i=1}^N [2y_i(c_l I(\mathbf{x}_i \in \mathcal{R}_l) + c_r I(\mathbf{x}_i \in \mathcal{R}_r) - c_m I(\mathbf{x}_i \in \mathcal{R}_m)) + c_m^2 I(\mathbf{x}_i \in \mathcal{R}_m) - c_l^2 I(\mathbf{x}_i \in \mathcal{R}_l) - c_r^2 I(\mathbf{x}_i \in \mathcal{R}_r)]$$

By replacing  $c$  by  $\hat{c}$ , we get :

$$2[n_l \bar{y}_l^2 + n_r \bar{y}_r^2 - \frac{1}{n}(n_l \bar{y}_l + n_r \bar{y}_r)^2] - n \times \frac{1}{n^2}(n_l \bar{y}_l + n_r \bar{y}_r)^2 - n_l \bar{y}_l^2 - n_r \bar{y}_r^2$$

It yields to :

$$-\frac{2}{n} n_l n_r \bar{y}_l \bar{y}_r + (n_l - \frac{n_l^2}{n}) \bar{y}_l^2 - (n_r - \frac{n_r^2}{n}) \bar{y}_r^2 = \frac{n_l n_r}{n} (\bar{y}_l - \bar{y}_r)^2$$

since  $n = n_l + n_r$ .

## Question 13

Let us assume that  $y_o$  changes from  $\mathcal{R}_l$  to  $\mathcal{R}_r$ . Then  $\bar{y}_{l,new} \leftarrow \frac{n_l}{n_l-1} \bar{y}_{l,old} - \frac{y_o}{n_l-1}$  and  $\bar{y}_{r,new} \leftarrow \frac{n_r}{n_r+1} \bar{y}_{r,old} + \frac{y_o}{n_r+1}$ . As a consequence, the new improvement can be written as :

$$\boxed{\frac{(n_l-1)(n_r+1)}{n} \left( \frac{n_l}{n_l-1} \bar{y}_{l,old} - \frac{y_o}{n_l-1} - \frac{n_r}{n_r+1} \bar{y}_{r,old} - \frac{y_o}{n_r+1} \right)^2}$$

## Question 14

Enlarging the class of functions to get a better MSE is good idea as long as it requires affordable computational cost. Usually it will reduce the MSE on future data but if the *true* function holds in a smaller class (*e.g.* linear function when we look for more complex polynomial functions), it will overfit the data and MSE will not be better on these data. Conversely, reducing the class of functions can be great for complexity purposes. Nonetheless it implies that our model will be more biased and probably the MSE will be high on future data.



### **Question 15**

One advantage would be the ability to predict more than two subgroups at each node of the tree. It will be a means to represent more complex patterns in the data. Nonetheless the splits at such a node could become meaningless and less effective. Knowing whether we should do such a split appears to be another issue.

### **Question 16**

With such relationships, the split could approximate linear patterns that exist within the training data which can be great in some cases. If such relationships between the inputs do not exist (because it might be simpler or even more complex, ie, quadratic), the model will be error-prone or with a really high variance.