Customer Analytics

Subset Selection, LASSO, Decision Trees & Random Forests

George Knox

Tilburg University

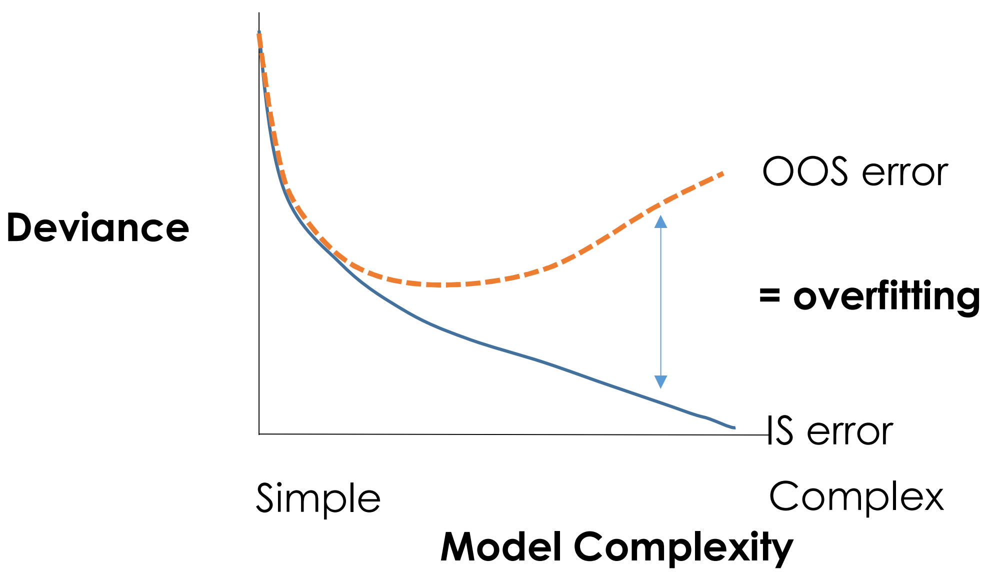
Subset selection

reading link:

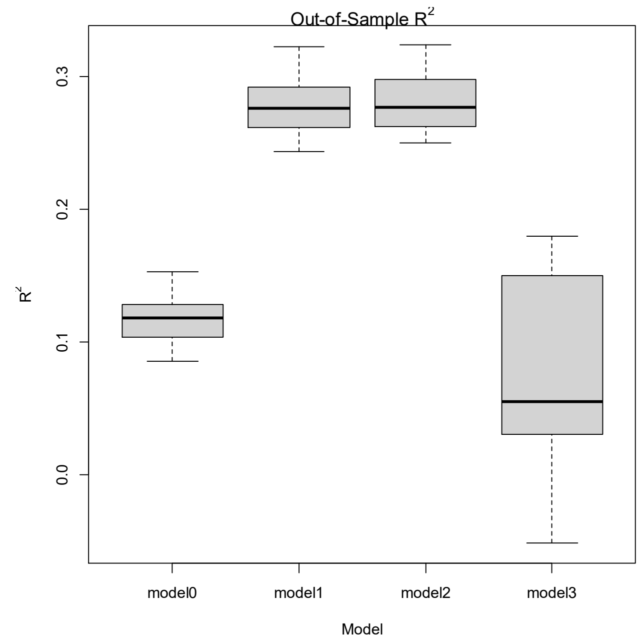
ISLR Ch. 6.1-2

From last lecture

What we want



What we got



How do we systematically search for the best model?

Best subset

There are p independent variables excluding the intercept. (23 covariates in model 1 of telco L3).

Loop over model size in steps. Start with a model that has only intercept $\mathcal{M}_0$.

1. Fit all p models that have 1 predictor, choose the one that has the best IS R^2. Call that model $\mathcal{M}_1$.

2. Fit all $\binom{p}{2} = \frac{p(p-1)}{2}$ models with 2 predictors, choose the one that has the IS R^2. Call that model $\mathcal{M}_2$.

p. Fit $\binom{p}{p} = 1$ model with p predictors, choose the one that has the best IS R^2. Call that model $\mathcal{M}_p$

Select best model from $\mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_2 \ldots \mathcal{M}_p$ using cross-validation (e.g., best OOS R^2). Run it on the full data set.

Forward stepwise regression

Loop over model size in steps. Start with a model that has only intercept $\mathcal{M}_0$.

0. Fit all $p - 0$ models that that augment the predictors in $\mathcal{M}_0$ with **one additional predictor**, choose the one that has the best IS R^2. Call that model $\mathcal{M}_1$.

1. Fit all $p - 1$ models that augment the predictors in $\mathcal{M}_1$ with **one additional predictor**, choose the one that has the best IS R^2. Call that model $\mathcal{M}_2$.

p-1. Fit $p - (p-1) = 1$ model that augments the predictors in $\mathcal{M}_{p-1}$ with **one additional predictor**, choose the one that has the best IS R^2. Call that model $\mathcal{M}_p$

- Select best model from $\mathcal{M}_1, \mathcal{M}_2 \ldots \mathcal{M}_p$ (e.g., best OOS R^2). Run it on the full data set.

Alternative to CV

- The R function we use to do forward stepwise regression, step, uses a penalized deviance to select models rather than cross-validation.
- We saw that IS deviance (and R^2) tend to overfit. The idea is to penalize the IS fit measures based on how many parameters they use.
- Akaike information criterion, is used by the step program.

$$\text{AIC} = 2p + \text{Dev}$$

Telco data set

Step		Df	Deviance	Resid. Df	Resid. Dev	AIC
1		NA	NA	7031	8143	8145
2	+ Contract	-2	1380.83	7029	6763	6769
3	+ InternetService	-2	413.97	7027	6349	6359
4	+ tenure	-1	284.48	7026	6064	6076
5	+ PaymentMethod	-3	53.93	7023	6010	6028
6	+ PaperlessBilling	-1	33.72	7022	5976	5996
7	+ OnlineSecurity	-1	27.40	7021	5949	5971
8	+ TotalCharges	-1	29.20	7020	5920	5944
9	+ PhoneService	-1	25.16	7019	5895	5921
10	+ TechSupport	-1	22.58	7018	5872	5900
11	+ MonthlyCharges	-1	11.31	7017	5861	5891
12	+ OnlineBackup	-1	11.41	7016	5849	5881
13	+ SeniorCitizen	-1	8.92	7015	5840	5874
14	+ MultipleLines	-1	4.62	7014	5836	5872
15	+ Dependents	-1	3.32	7013	5832	5870
16	+ DeviceProtection	-1	2.61	7012	5830	5870

Problems with forward selection

Time: Takes about 10 seconds for 7000 responses 20 covariates.

Unstable: small changes in the data lead to large differences in model selection

Alternative: estimate all coefficients but shrink the estimates towards zero.

Regularization: LASSO

Penalty term

$$\hat{\beta} = \arg\min_{\beta} \left\{ \mathrm{Dev}(\beta) + \lambda \sum_{j=1}^{p} \left|\beta_j\right| \right\}, \qquad \lambda \geq 0$$

The penalty weight λ shrinks the size of the β's

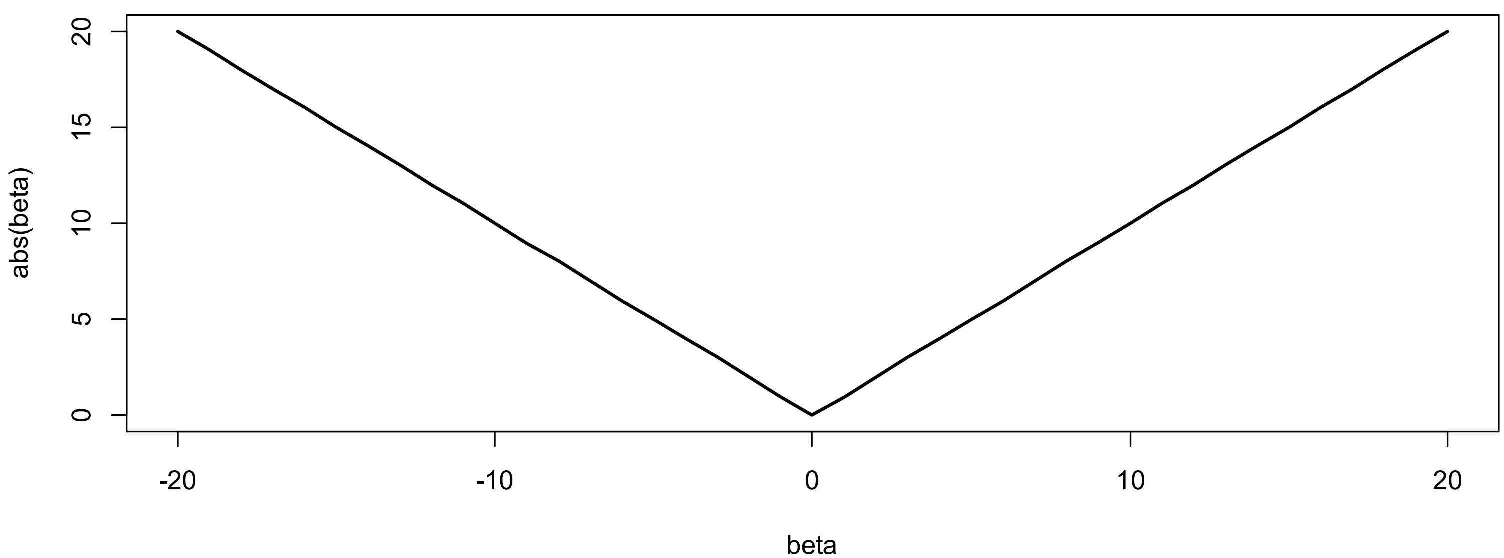
- The larger λ, the more $\hat{\beta}$'s are exactly zero. LASSO performs variable selection and yields "sparse" models.
- If λ = 0, we get logistic regression

Shrinking β's means that the predictions shrink to the mean

- Idea is the same from L2: when you don't know, shrink to the mean

Absolute value

$|\beta|$



The shape of the penalty function means that some coefficients will be exactly zero

Regularization path

- Start with a large penalty term λ_1 so that all coefficients are zero.

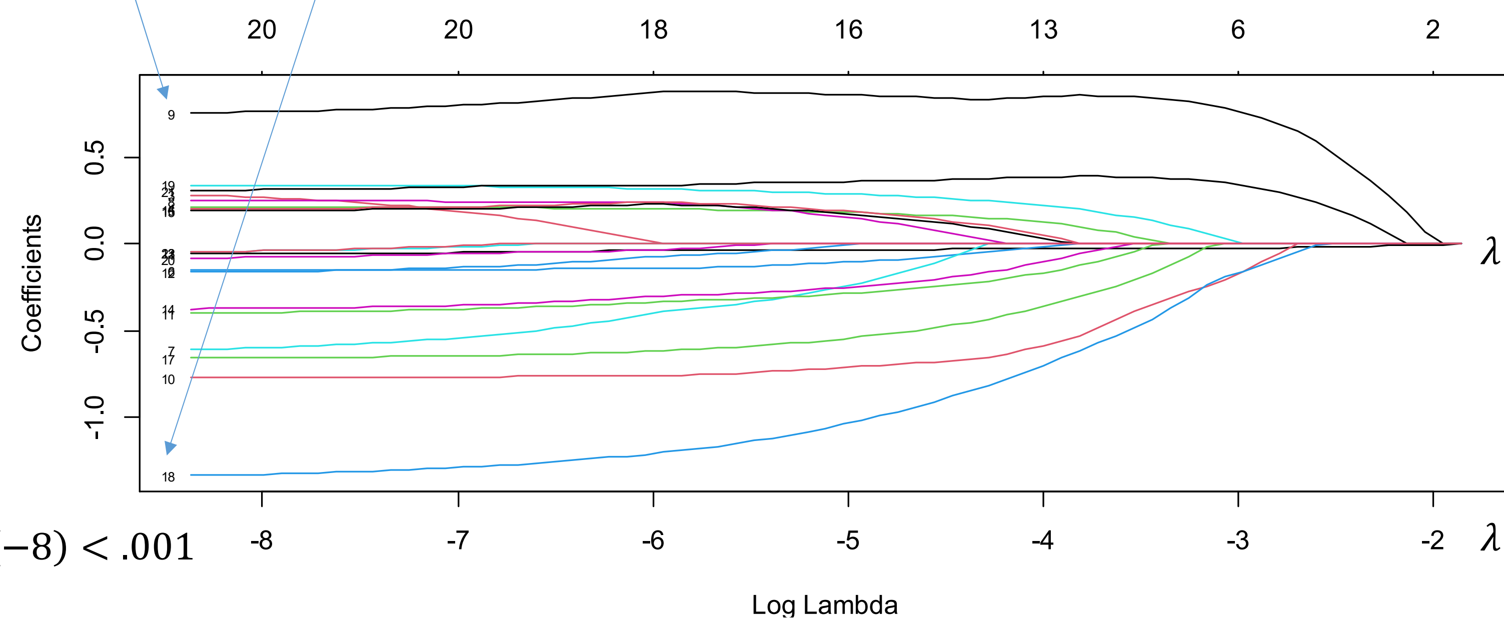
- There are a set of critical penalty weights values $\lambda_1 > \lambda_2 \ldots > \lambda_p$, where the active set of nonzero coefficients changes. They can be solved for analytically, speeding up computation.

- Between these critical values each coefficient increases or decreases linearly.

- A smart algorithm computes the entire regularization path for about the same computational cost as ordinary regression.

```
 [1] "telco.tenure"                       "telco.MonthlyCharges"
 [3] "telco.TotalCharges"                 "SeniorCitizen"
 [5] "PartnerYes"                         "DependentsYes"
 [7] "PhoneServiceYes"                    "MultipleLinesYes"
 [9] "InternetServiceFiber.optic"         "InternetServiceNo"
[11] "OnlineSecurityYes"                  "OnlineBackupYes"
[13] "DeviceProtectionYes"                "TechSupportYes"
[15] "StreamingTVYes"                     "StreamingMoviesYes"
[17] "ContractOne.year"                   "ContractTwo.year"
[19] "PaperlessBillingYes"                "PaymentMethodCredit.card..automatic."
[21] "PaymentMethodElectronic.check"      "PaymentMethodMailed.check"
```

Number of non-zero coefficients



λ for which all beta = 0

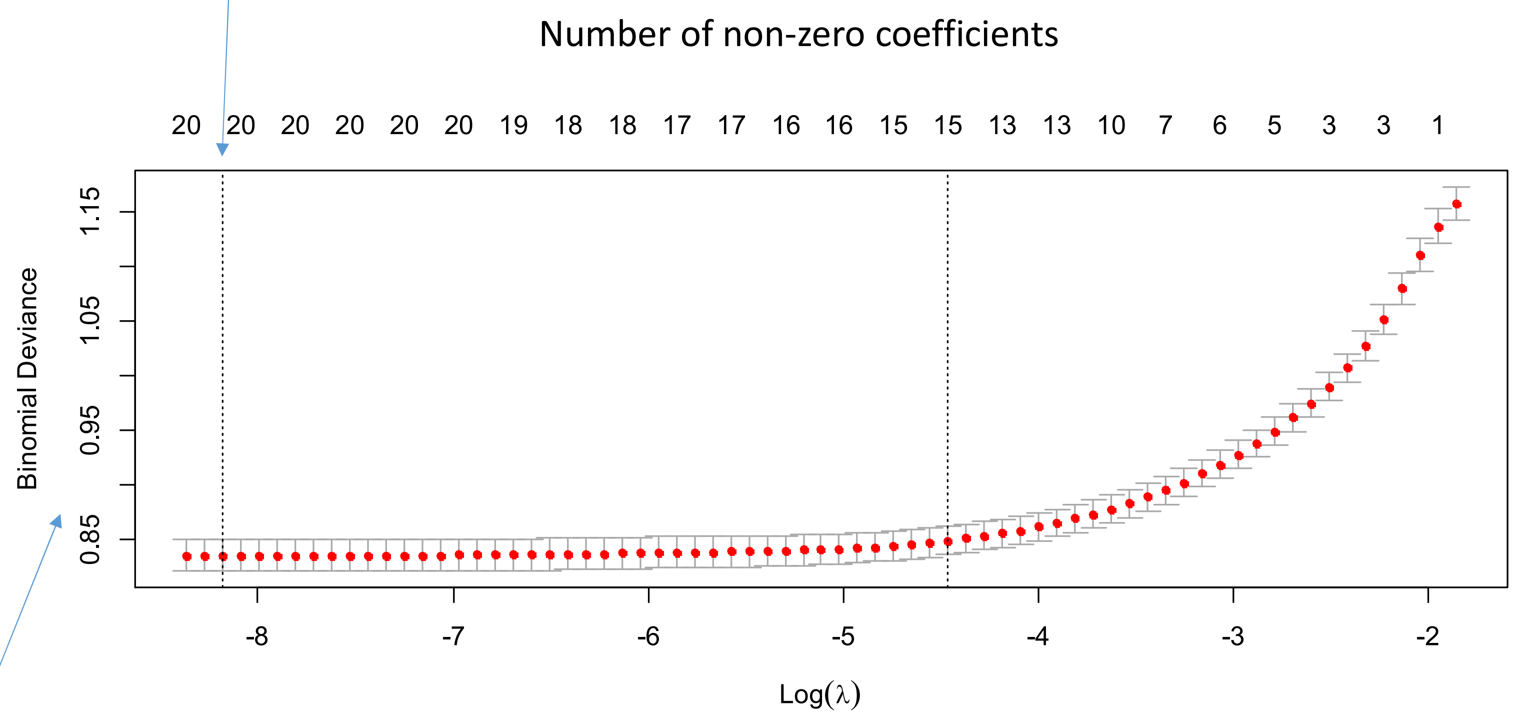
$\lambda = \exp(-8) < .001$

$\lambda = \exp(-2) = .14$

Small penalty cost $\log(\lambda)$ Large penalty cost

Stability means this scales to many big data app's

Choose λ such that K-fold CV deviance is minimized



Deviance divided by number of obs

That model is

Note no standard errors

2 variables didn't make the cut

```
(Intercept)                              0.0480
telco.tenure                            -0.0548
telco.MonthlyCharges                     .
telco.TotalCharges                       0.2604
SeniorCitizen                            0.2136
PartnerYes                               .
DependentsYes                           -0.1475
PhoneServiceYes                         -0.5920
MultipleLinesYes                         0.2475
InternetServiceFiber.optic               0.7701
InternetServiceNo                       -0.7711
OnlineSecurityYes                       -0.3918
OnlineBackupYes                         -0.1557
DeviceProtectionYes                     -0.0359
TechSupportYes                          -0.3688
StreamingTVYes                           0.1972
StreamingMoviesYes                       0.2067
ContractOne.year                        -0.6548
ContractTwo.year                        -1.3247
PaperlessBillingYes                      0.3402
PaymentMethodCredit.card..automatic.    -0.0735
PaymentMethodElectronic.check            0.3163
PaymentMethodMailed.check               -0.0352
```

Decision Trees

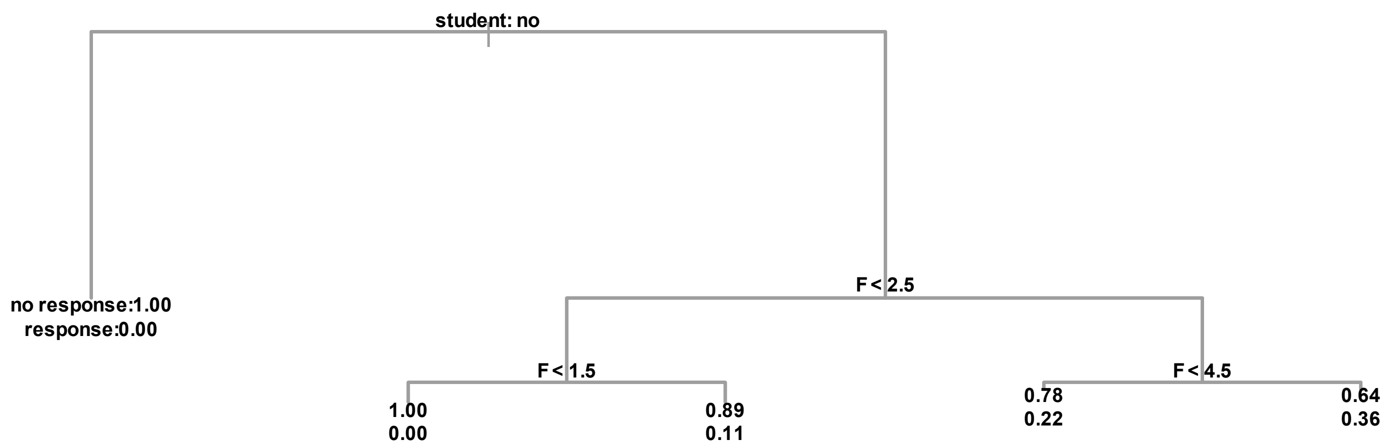
Reading:

BKN Ch. 17

ISLR Ch. 8.1

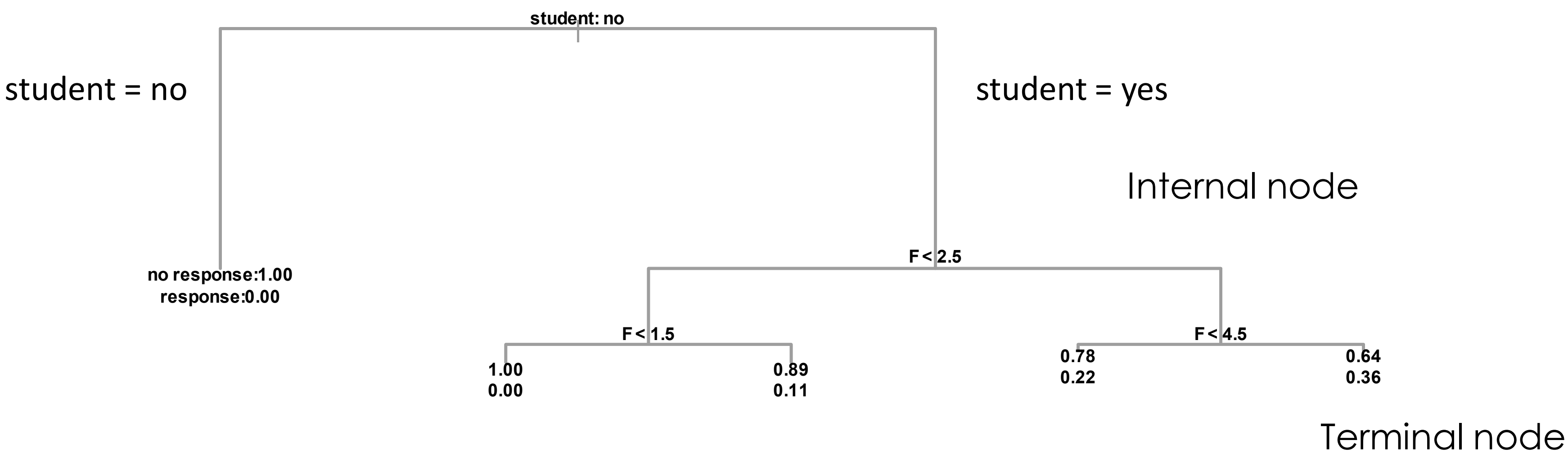
Motivation

We want a model that is simple to understand and communicate



Closer look

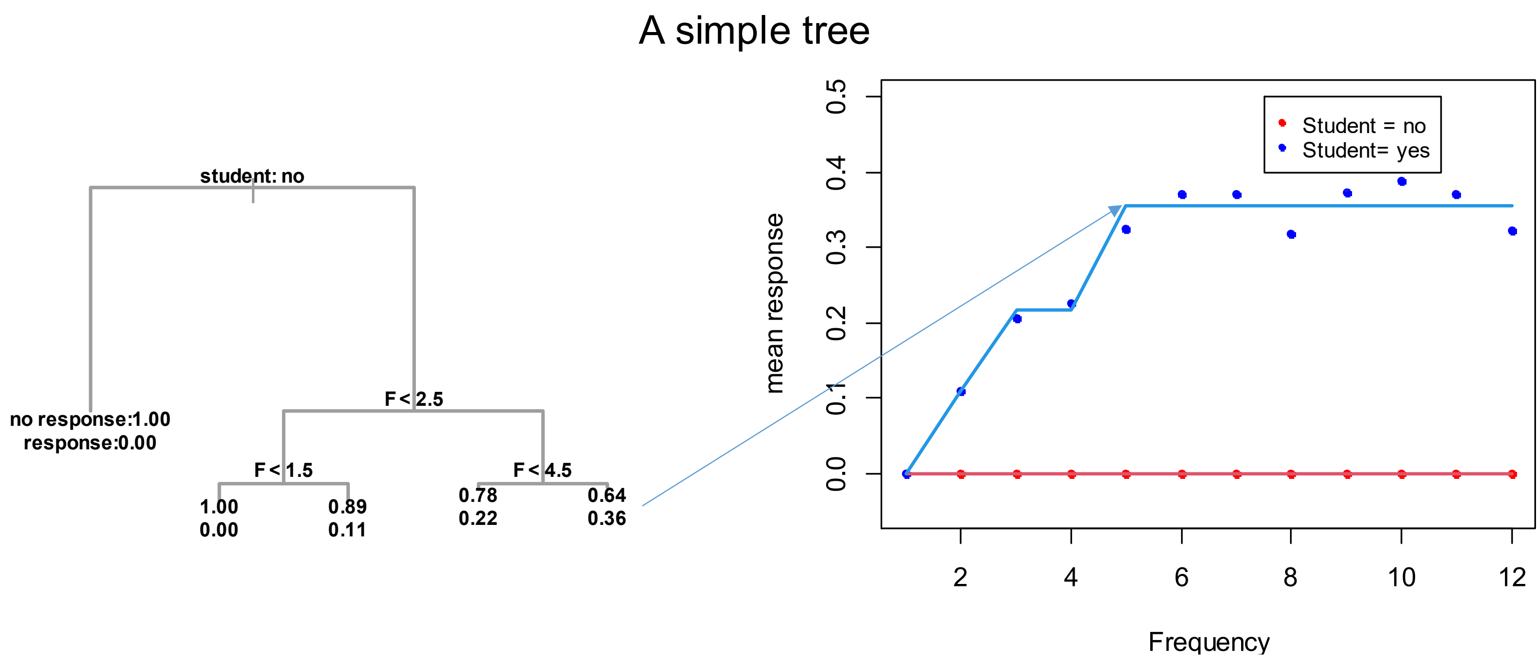
Root node



Every leaf (terminal node) has a prediction: the average response rate of that group

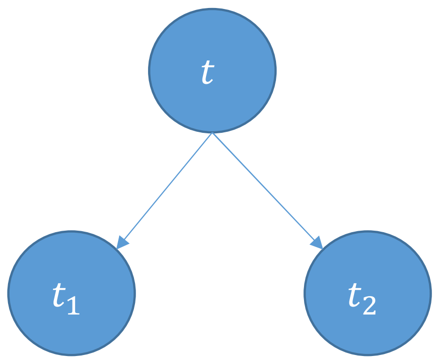
How it fits the data

```
Leaf (terminal) nodes## Classification tree:
## tree(formula = respmail ~ ., data = subset(ebeer, select = c(respmail,
##     F, student)), mindev = 0.005)
## Number of terminal nodes:  5
## Residual mean deviance:  0.505 = 2500 / 4950
## Misclassification error rate: 0.124 = 616 / 4952
```



How does it decide where to split?

CRT: Gini impurity



Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset.

Gini index of impurity

$$i(t) = 1 - \sum_j p(j|t)^2$$

Response	100	100
No response	0	100

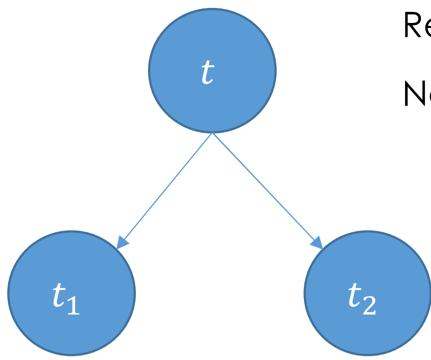
$i(t_1) = 1 - 1^2 - 0^2 = 0$

$i(t_2) = 1 - 0.5^2 - 0.5^2 = 0.5$

Minimal impurity

Maximal impurity

Splitting algorithm: CRT



Response	200
No response	100

$$i(t) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$$

Response	100	100
No response	0	100

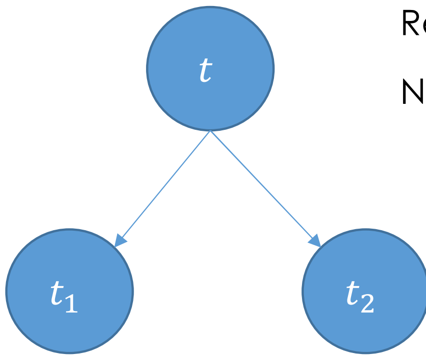
$$i(t) = 1 - \sum_j p(j|t)^2$$

$$i(t_1) = 1 - 1^2 - 0^2 = 0$$

$$i(t_2) = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

TILBURG UNIVERSITY

Splitting algorithm: CRT



Response	200
No response	100

$$i(t) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$$

	t_1	t_2
Response	100	100
No response	0	100

$$i(t) = 1 - \sum_j p(j|t)^2$$

$$i(t_1) = 1 - 1^2 - 0^2 = 0$$

$$i(t_2) = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

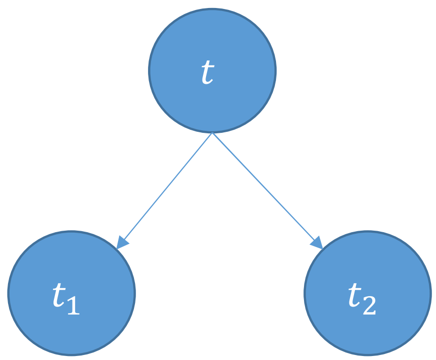
Decrease in impurity by split S

$$\Delta i(S,t) = i(t) - \left(\frac{n_1}{n}\right) i(t_1) - \left(\frac{n_2}{n}\right) i(t_2)$$

$$\Delta i(S,t) = \frac{4}{9} - \frac{1}{3} * 0 - \frac{2}{3} * \frac{1}{2} = \frac{1}{9}$$

Splitting algorithm: CRT

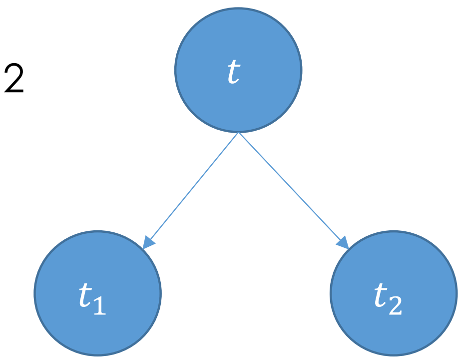
Potential split 1



	t_1	t_2
Response	100	100
No response	0	100

$$\Delta i(S,t) = \frac{1}{9}$$

Potential split 2



	t_1	t_2
Response	150	50
No response	0	100

$i(t_1) = 0 \qquad i(t_2) = \frac{4}{9}$

$$\Delta i(S,t) = \frac{4}{9} - \frac{1}{2} * 0 - \frac{1}{2} * \frac{4}{9} = \frac{2}{9}$$

Splitting algorithm: CRT

$$\Delta i(S,t) = i(t) - \left(\frac{n_1}{n}\right) i(t_1) - \left(\frac{n_2}{n}\right) i(t_2)$$

Decrease in impurity by split 1 at node t $\Delta i(x,t) = \frac{1}{9}$

Decrease in impurity by split 2 at node t $\Delta i(y,t) = \frac{2}{9}$

Decrease in impurity is larger when we split with Y than X, so choose Y split.

We stop when the decrease is smaller than some threshold, or when leaves are small (few observations)

Decision Tree vs. Logistic regression

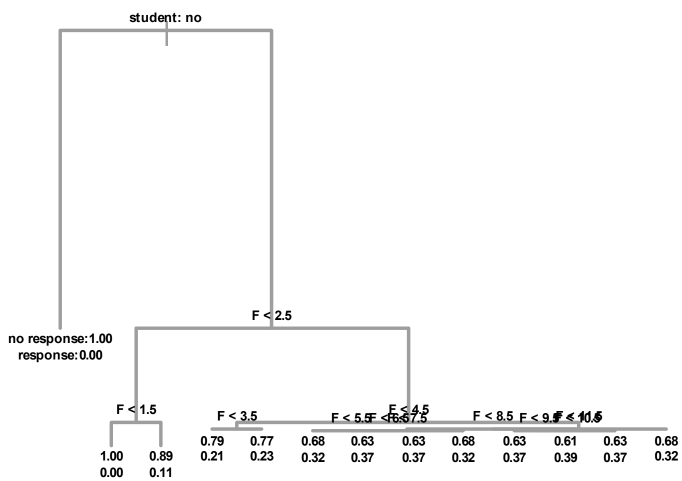
$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots \beta_p x_p$$

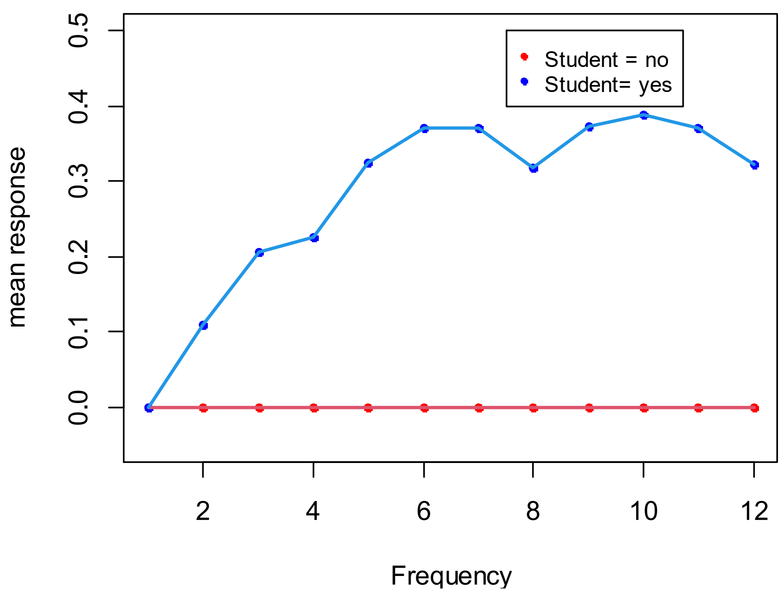
$$p = \beta_1 1\{X \in \text{ Leaf}_1\} + \beta_2 1\{X \in \text{ Leaf}_2\} + ..$$

Non-parametric: no assumption made about relationship between x and p.

We can fit the in-sample data arbitrarily well

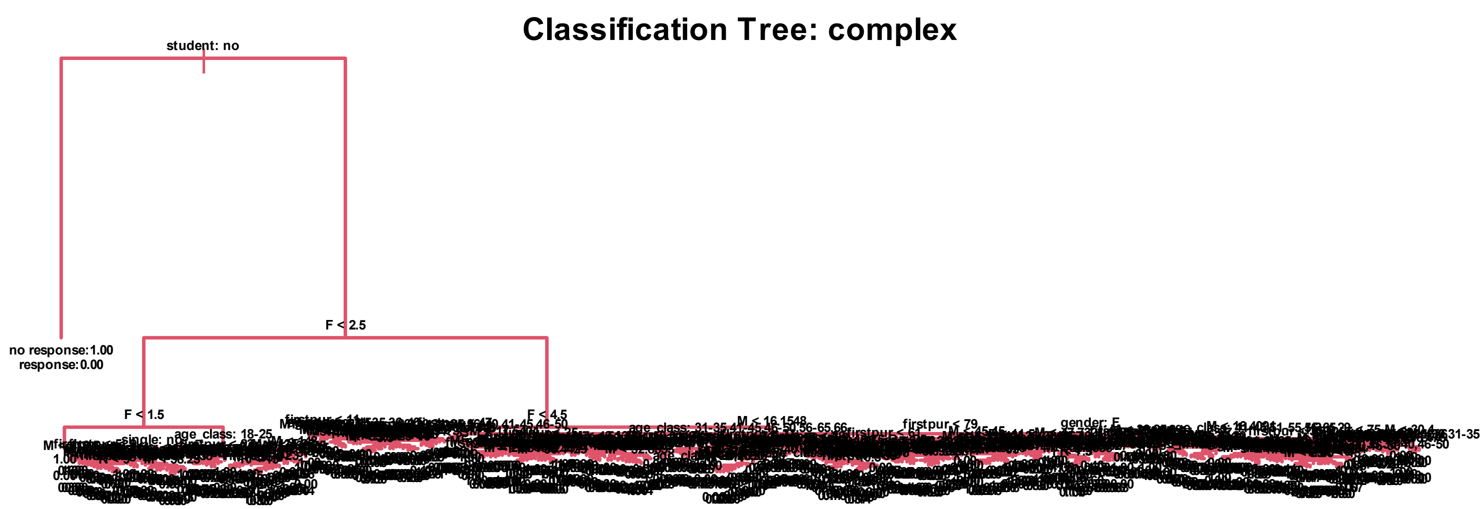
A complex tree





We lower the threshold for improvement to zero, the tree grows as complex as the data.

We can fit the in-sample data arbitrarily well



What problems do you foresee?

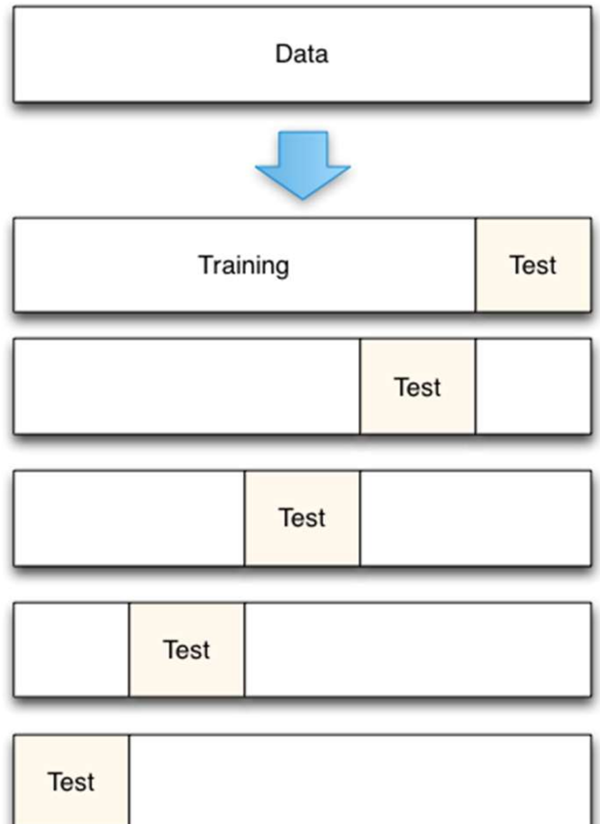
Decision trees

- Advantages:
  - Interpretability
  - Nonparametric: more flexible than logistic regression

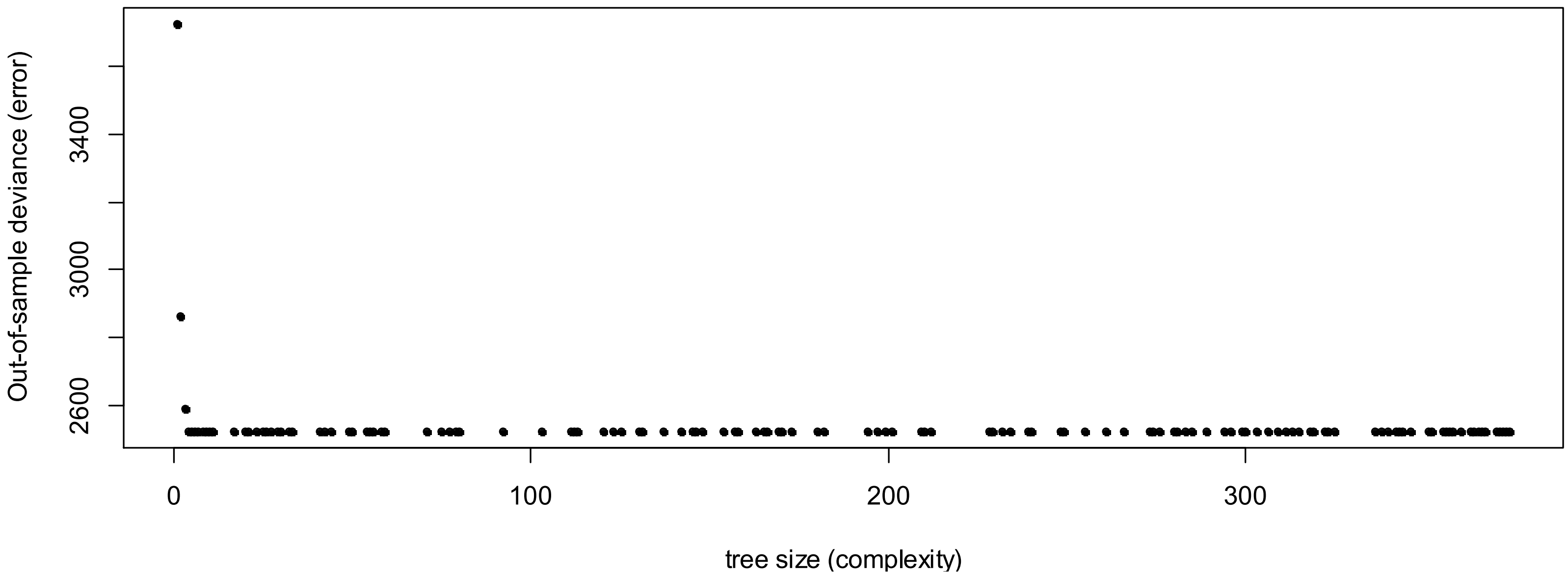
- Disadvantages:
  - Unstable -> irrelevant variables can change the model results
  - **Tendency to overfit the data**

K-fold cross validation

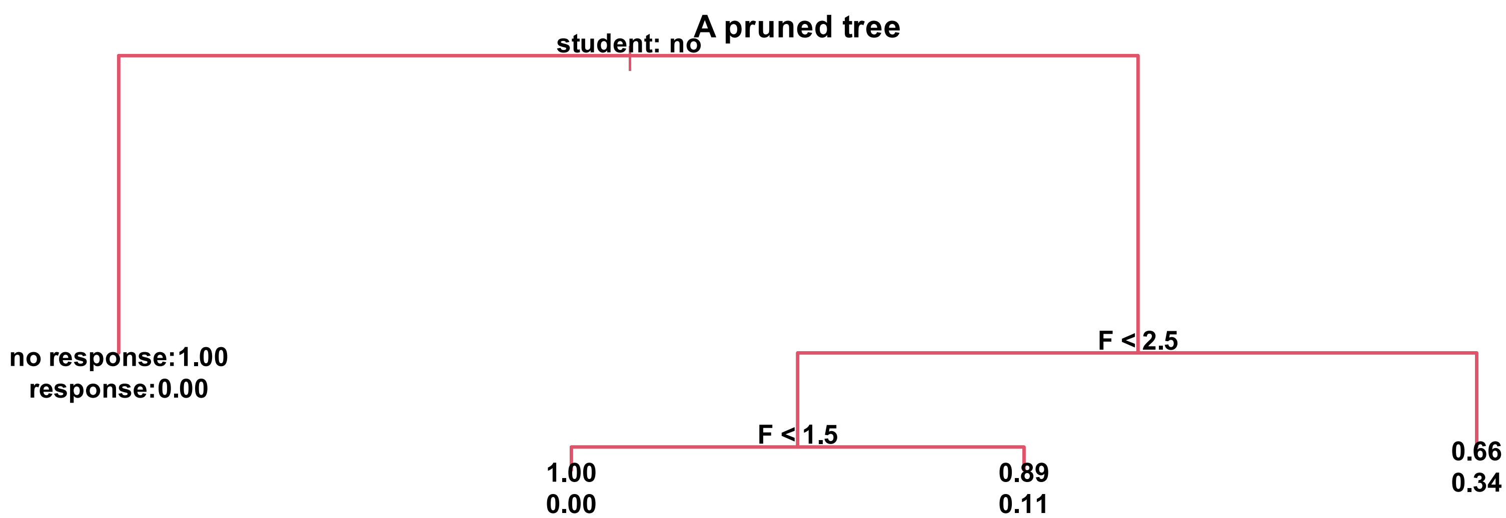
- Here K = 5.
- Data randomly split into 5 equally sized groups of 20% each.
- 4 groups used to fit, one group to validate.
- Repeat so that all data is used.



Comparing OOS error



No improvement over 4


A pruned tree
student: no
no response:1.00
response:0.00
F < 2.5
F < 1.5
1.00
0.00
0.89
0.11
0.66
0.34

Random Forests

Reading:

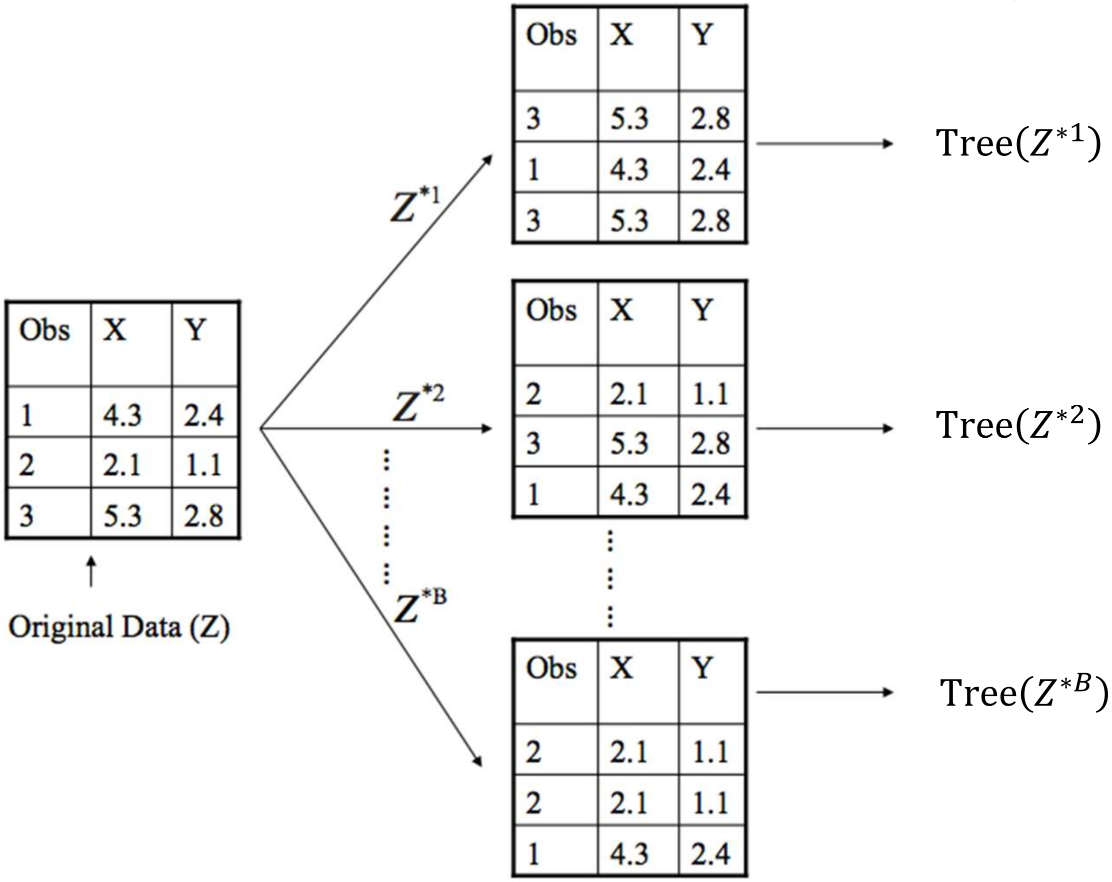
ISLR Ch. 8.2

Extension: bagging

- Idea: averaging a set of observations reduces variance
  - One tree has high variance, but an average of many trees will have low variance
- Bagging = bootstrap aggregation
- From L1: bootstrap sampling = take a sample of same size from the original dataset, but <u>with replacement</u>
  - Same observation can occur multiple times

Create B bootstrapped samples

For each b = 1, .. B samples, estimate a tree



Final model is an <u>average over all B trees.</u>

Extension: random forest

- Idea: averaging a set of uncorrelated observations reduces variance even further than correlated observations

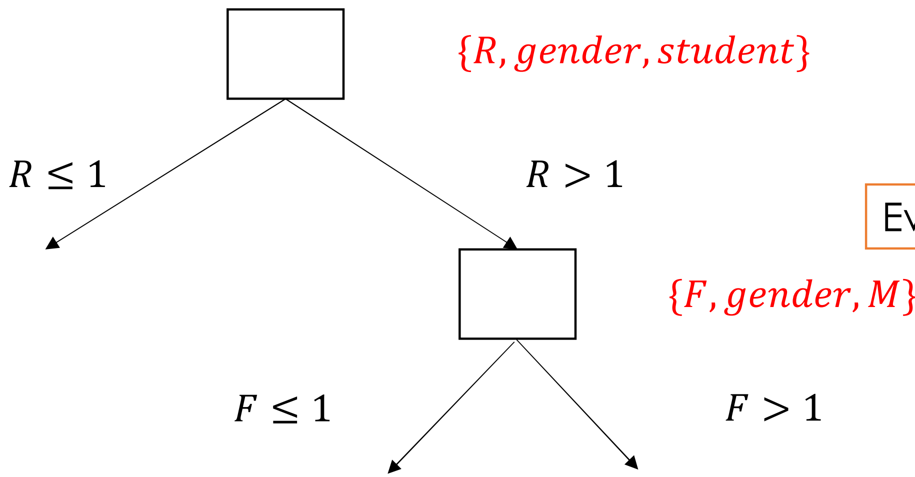
- Each time a split is considered, only a random sample of m predictors is chosen as split candidates from the full set of p predictors

$$m \approx \sqrt{p}$$

- Random sample of 3 out of 8 predictors at each split considered

example

Random sample of $m = 3$ predictors



Every split, randomly choose 3

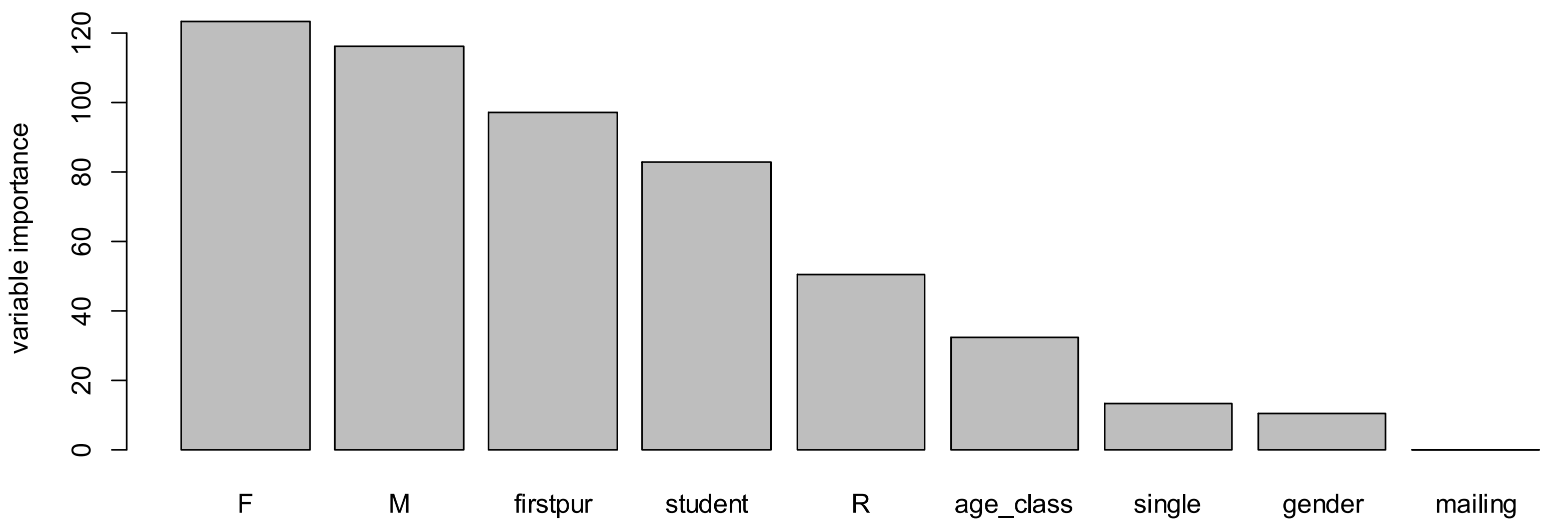
If we choose $m = p$, then random forest is the same as bagging

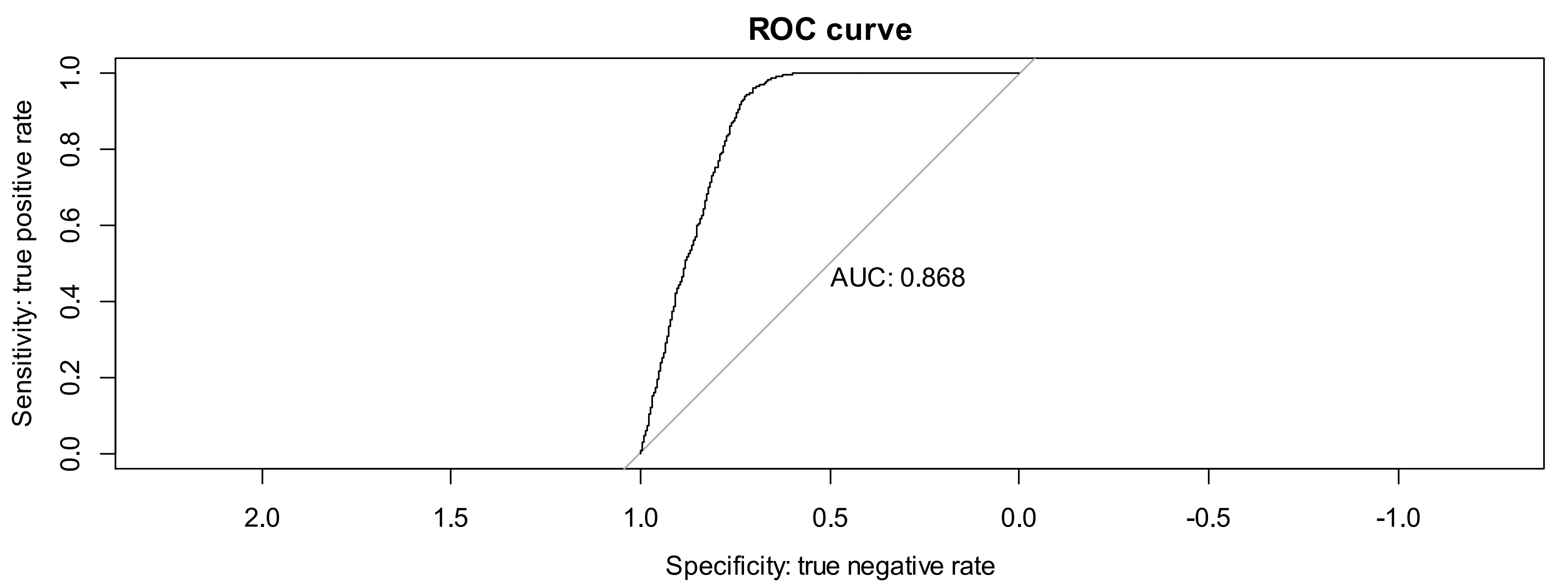
Why?

- Under bagging, models are highly correlated
  - A strong predictor will appear in all bagged trees, and predictions across bagged trees will be correlated
  - An average over many correlated models

- Random forests de-correlate models
  - Even a strong predictor will have a $\frac{p-m}{p}$ fraction of times not in the tree

Random forest variable importance





Tilburg University