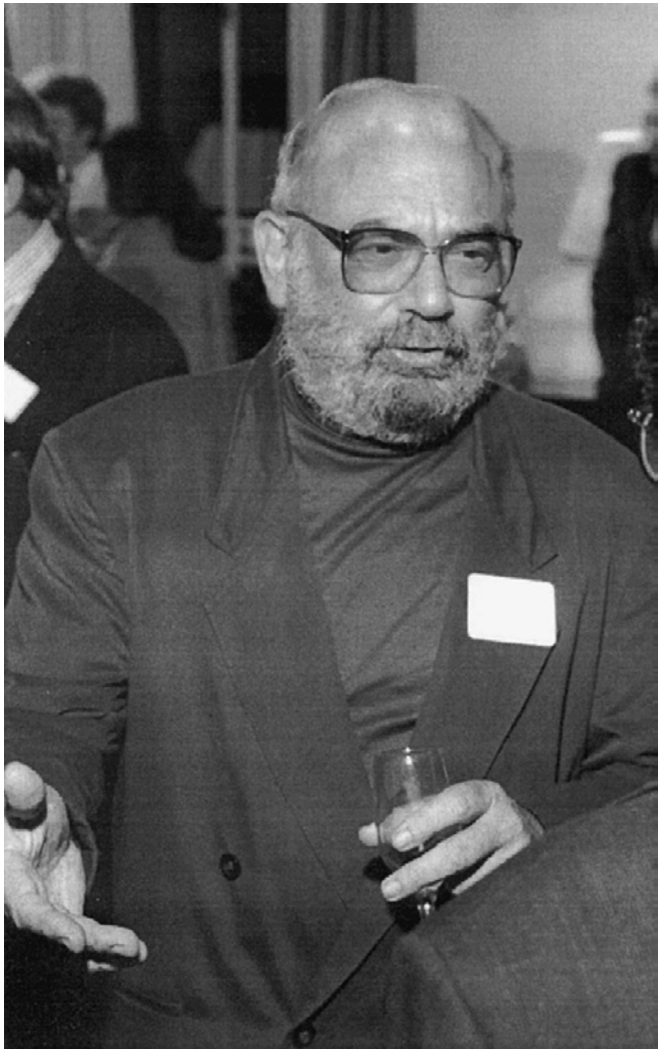
Random Forests for Regression and Classification

Adele Cutler

Utah State University

Leo Breiman, 1928 - 2005



1954: PhD Berkeley (mathematics)

1960 -1967: UCLA (mathematics)

1969 -1982: Consultant

1982 - 1993 Berkeley (statistics)

1984 “Classification & Regression Trees”
(with Friedman, Olshen, Stone)

1996 “Bagging”

2001 “Random Forests”

Random Forests for Regression and Classification

Outline

- Background.
- Trees.
- Bagging predictors.
- Random Forests algorithm.
- Variable importance.
- Proximity measures.
- Visualization.
- Partial plots and interpretation of effects.

What is Regression?

Given data on predictor variables (inputs, X) and a **continuous response variable** (output, Y) build a model for:

- Predicting the value of the response from the predictors.
- Understanding the relationship between the predictors and the response.

e.g. predict a person's **systolic blood pressure** based on their age, height, weight, etc.

Regression Examples

- Y: **income**

  X: age, education, sex, occupation, …
- Y: **crop yield**

  X: rainfall, temperature, humidity, …
- Y: **test scores**

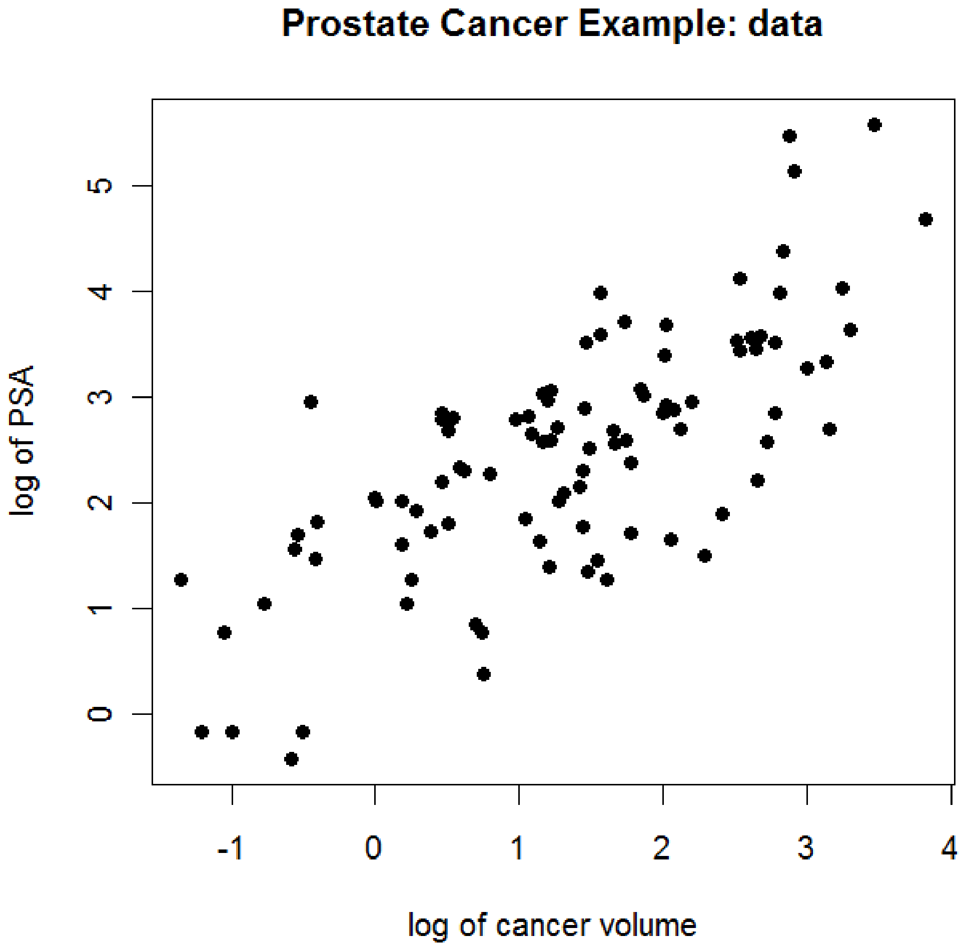
  X: teaching method, age, sex, ability, …
- Y: **selling price of homes**

  X: size, age, location, quality, …

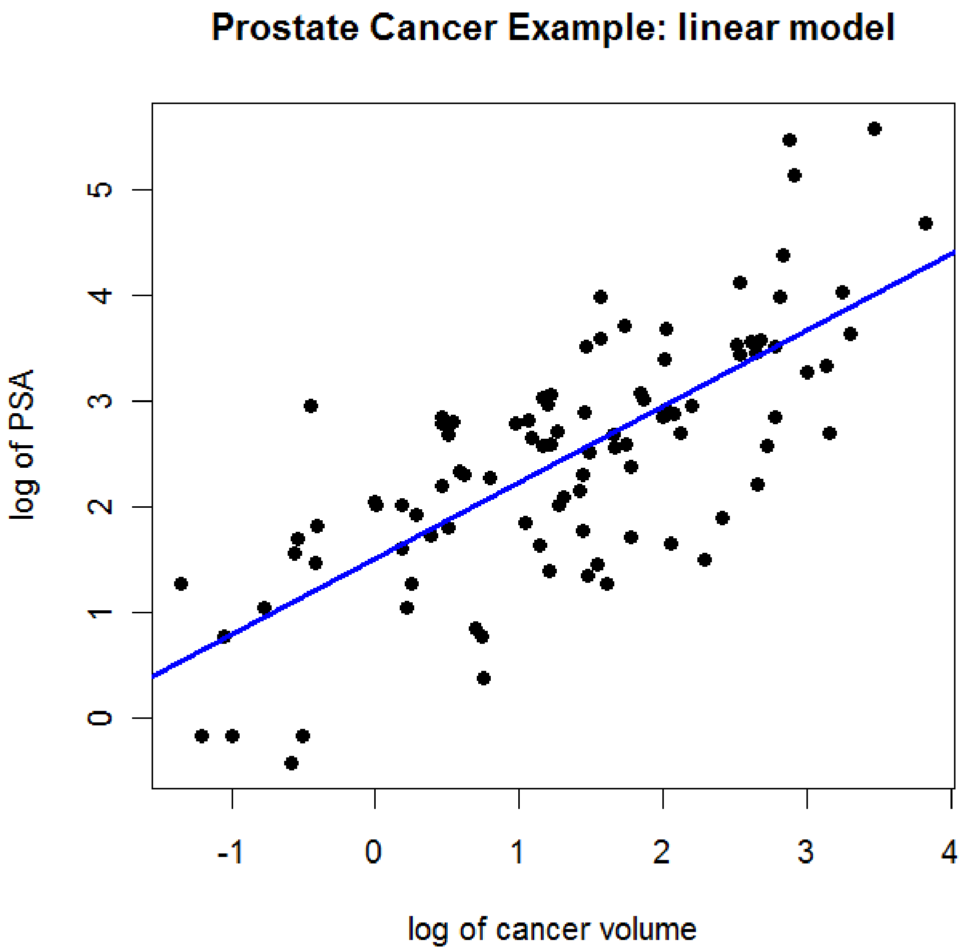
Regression Background

- Linear regression
- Multiple linear regression
- Nonlinear regression (parametric)
- Nonparametric regression (smoothing)
  - Kernel smoothing
  - B-splines
  - Smoothing splines
  - Wavelets

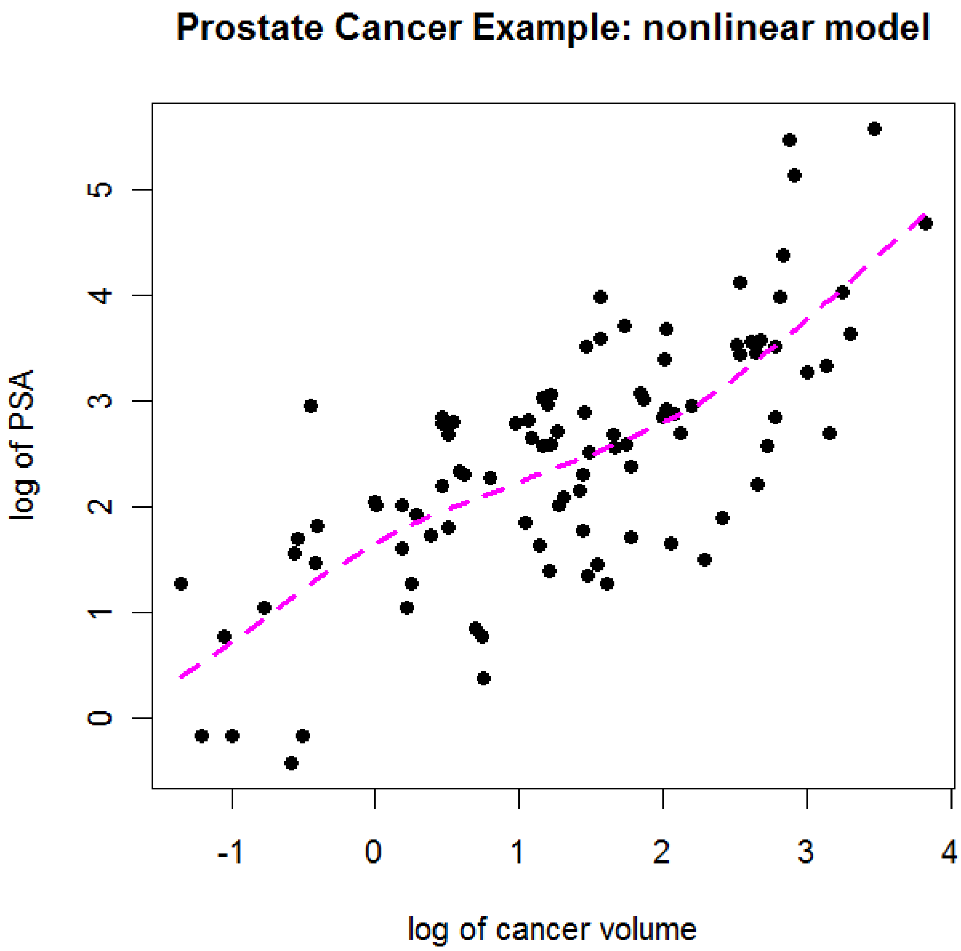
Regression Picture



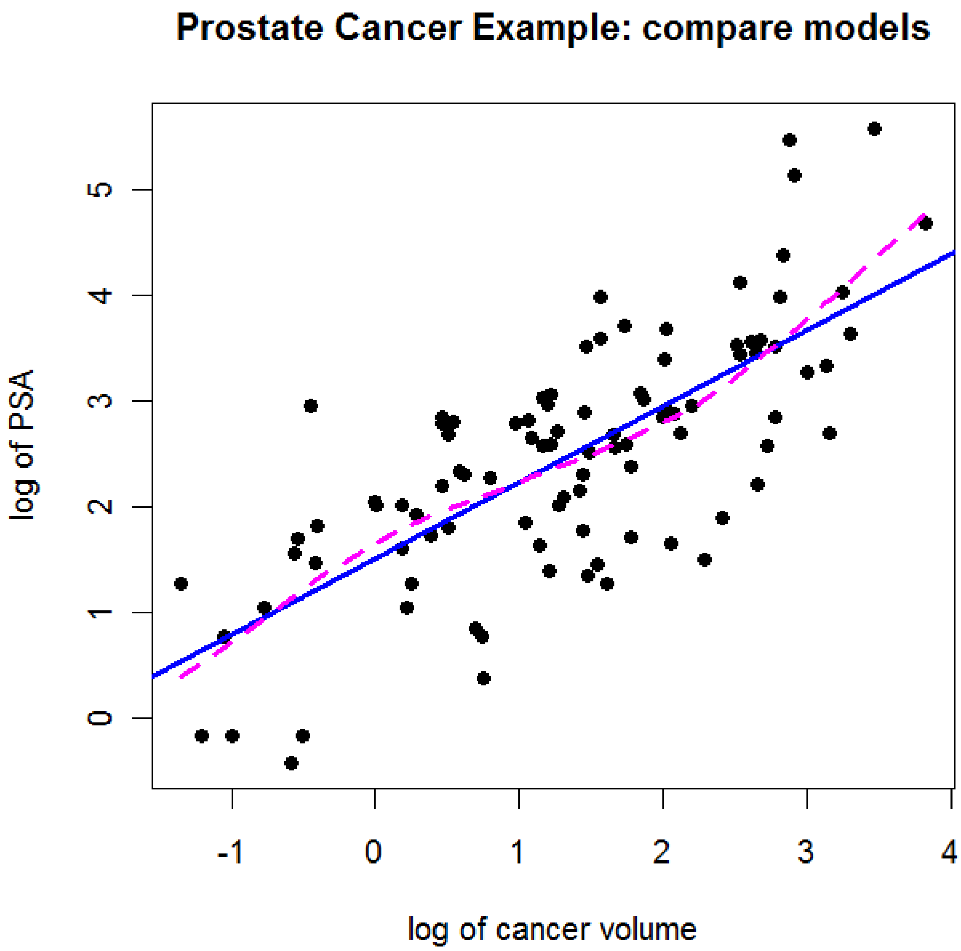
Regression Picture



Regression Picture



Regression Picture



What is Classification?

Given data on predictor variables (inputs, X) and a **categorical response variable** (output, Y) build a model for:

- Predicting the value of the response from the predictors.
- Understanding the relationship between the predictors and the response.

e.g. predict a person's **5-year-survival (yes/no)** based on their age, height, weight, etc.

Classification Examples

- Y: **presence/absence of disease**

  X: diagnostic measurements
- Y: **land cover (grass, trees, water, roads…)**

  X: satellite image data (frequency bands)
- Y: **loan defaults (yes/no)**

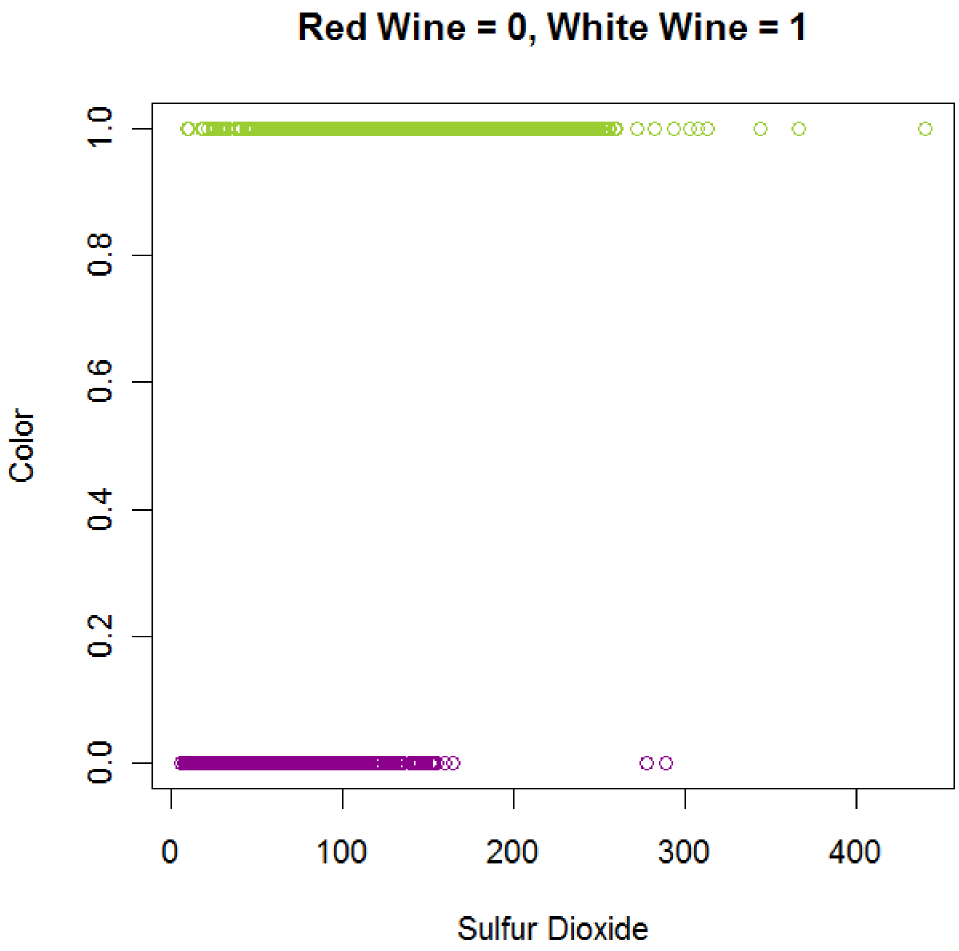
  X: credit score, own or rent, age, marital status, …
- Y: **dementia status**

  X: scores on a battery of psychological tests

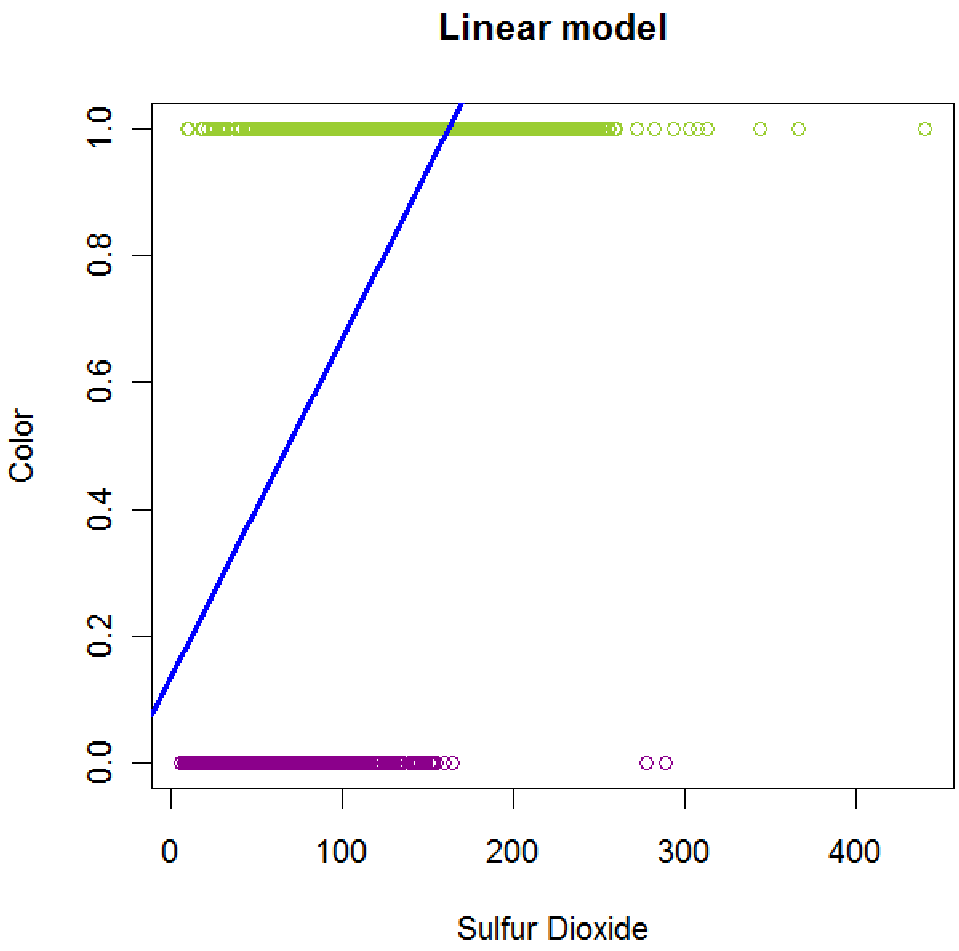
Classification Background

- Linear discriminant analysis (1930’s)
- Logistic regression (1944)
- Nearest neighbors classifiers (1951)

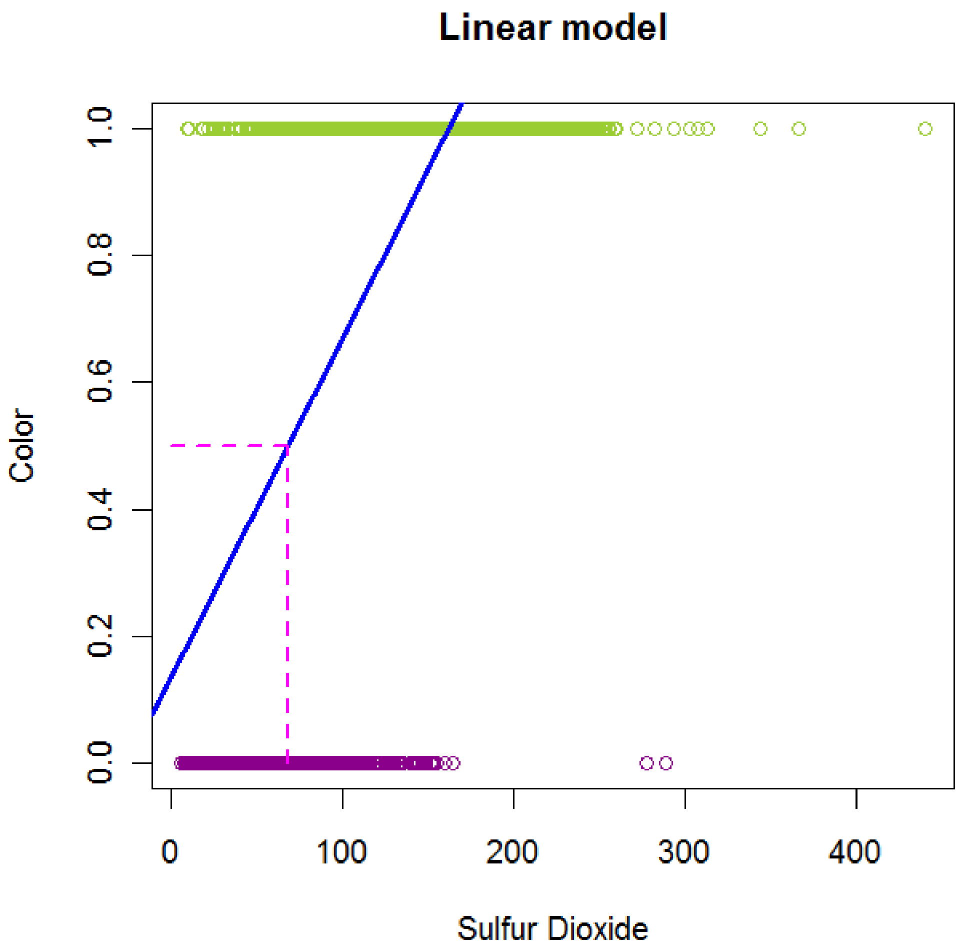
Classification Picture



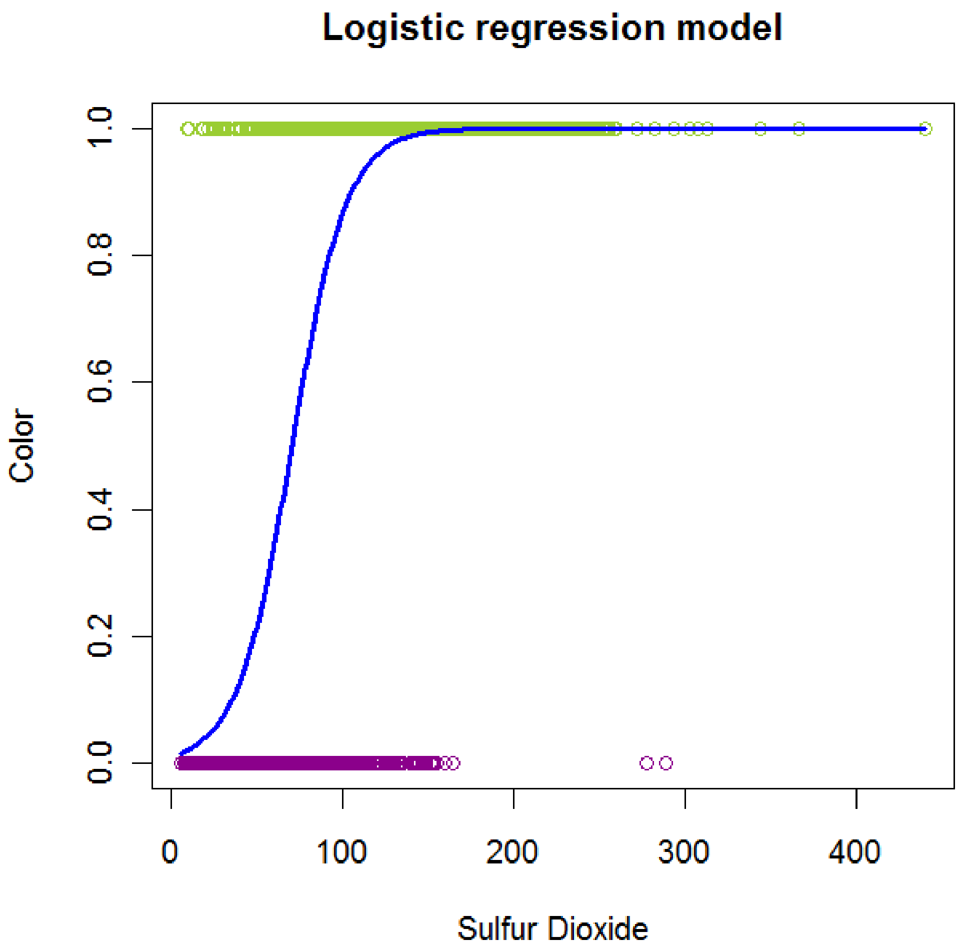
Classification Picture



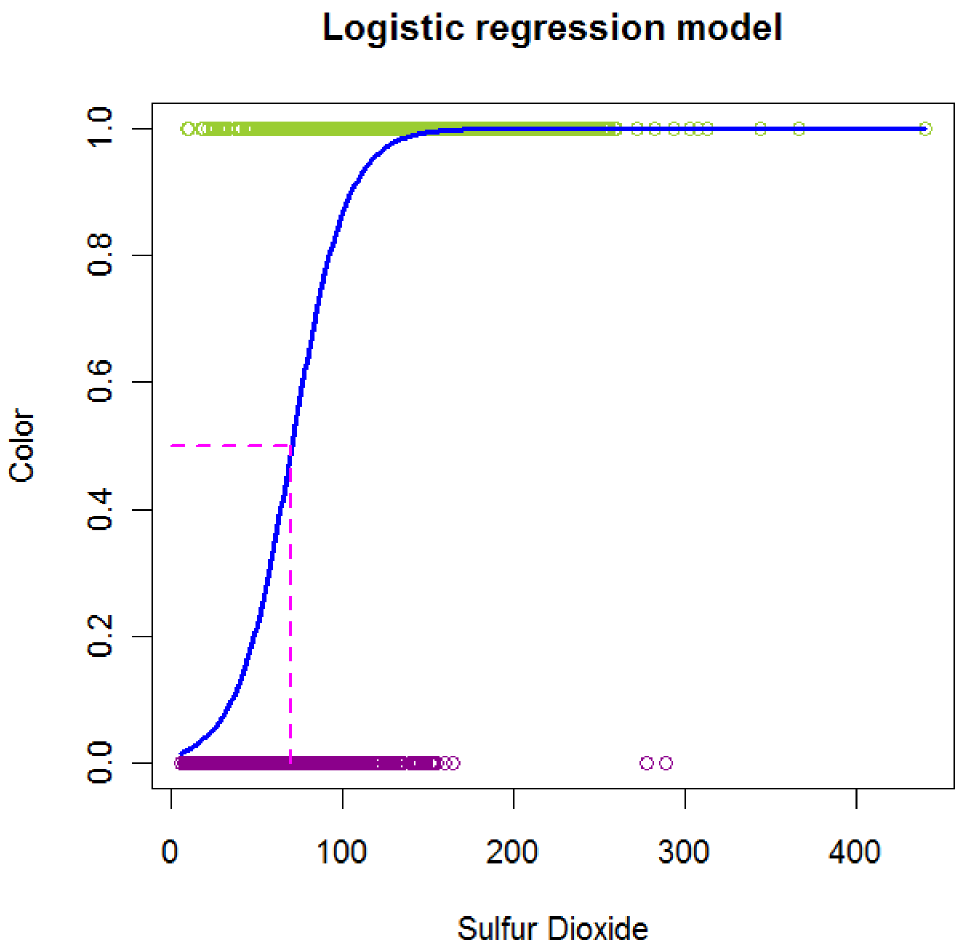
Classification Picture



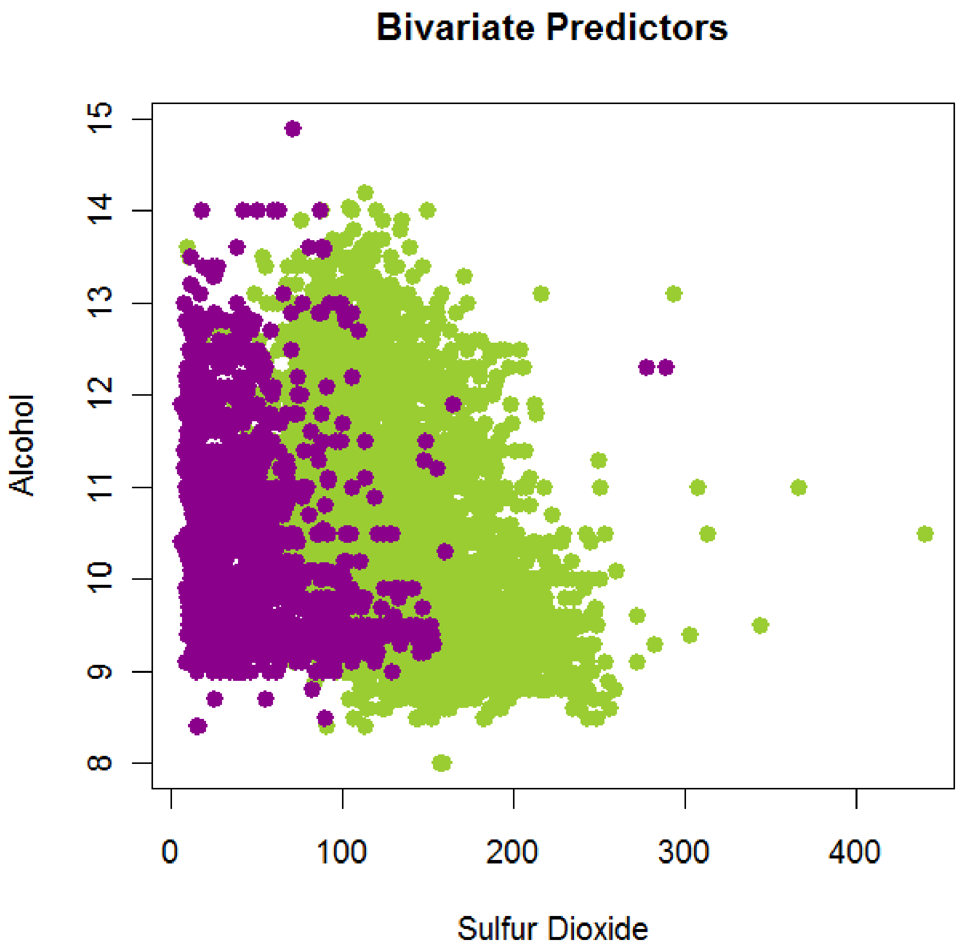
Classification Picture



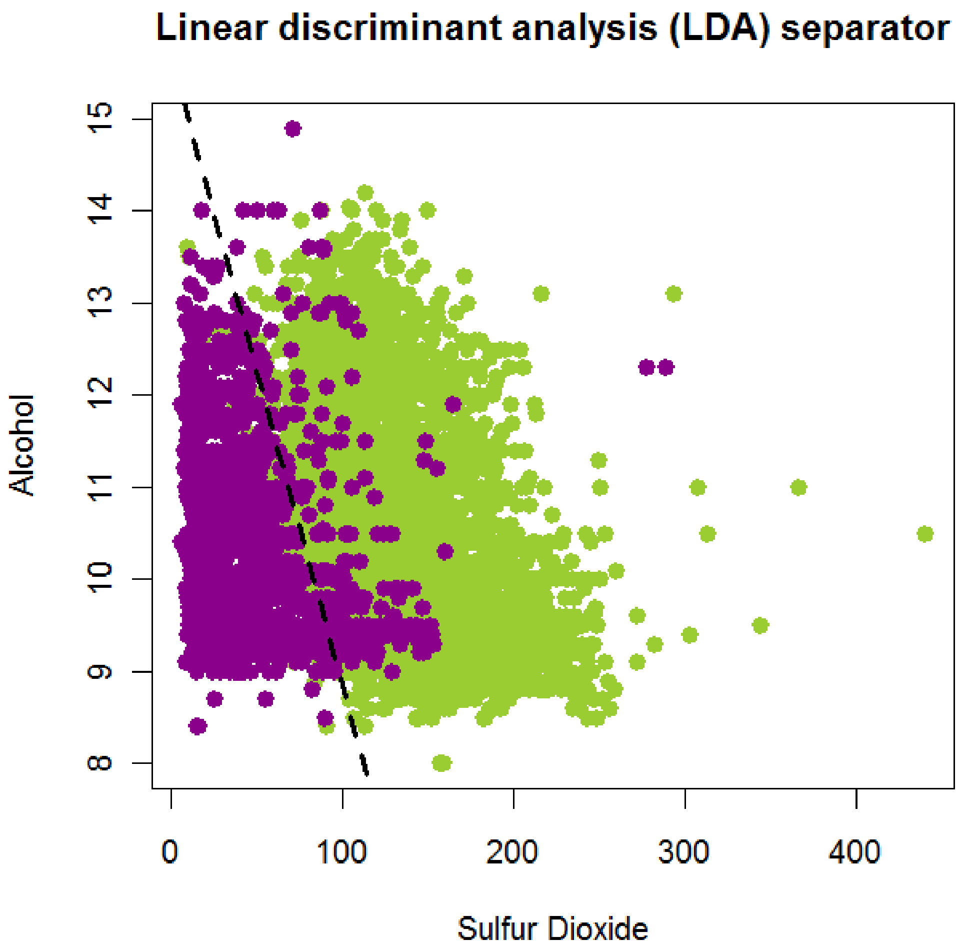
Classification Picture



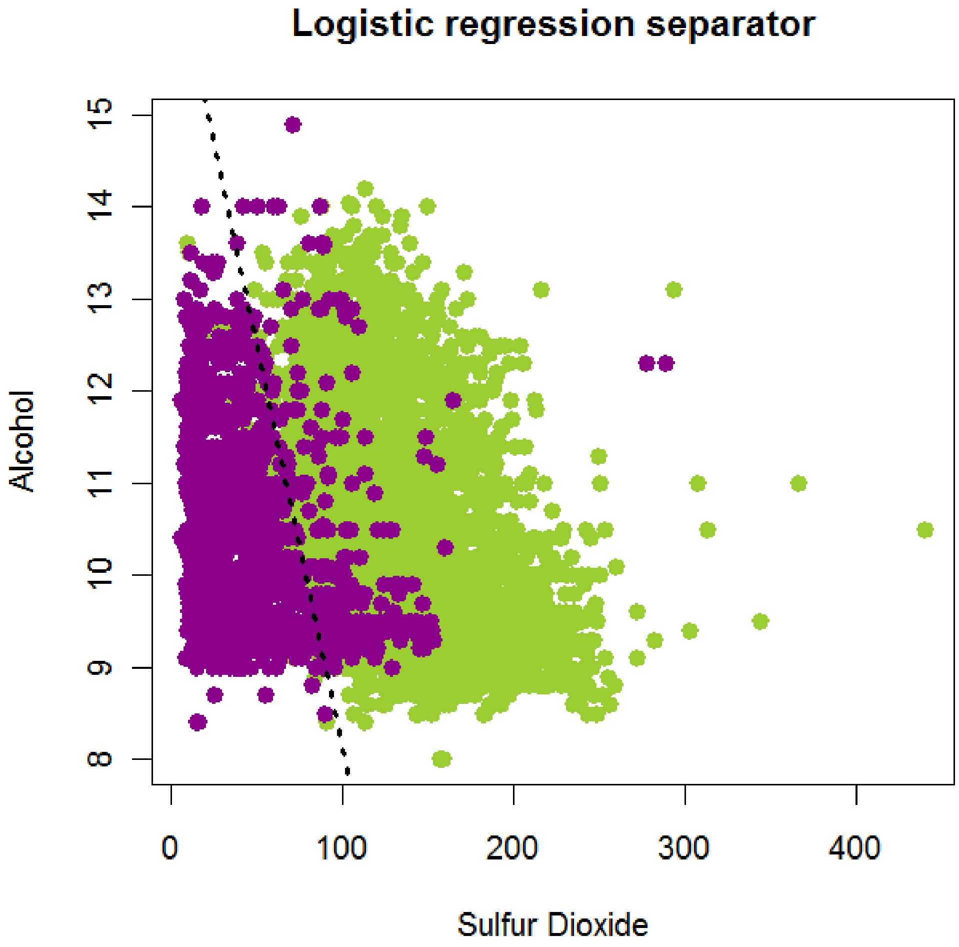
Classification Picture



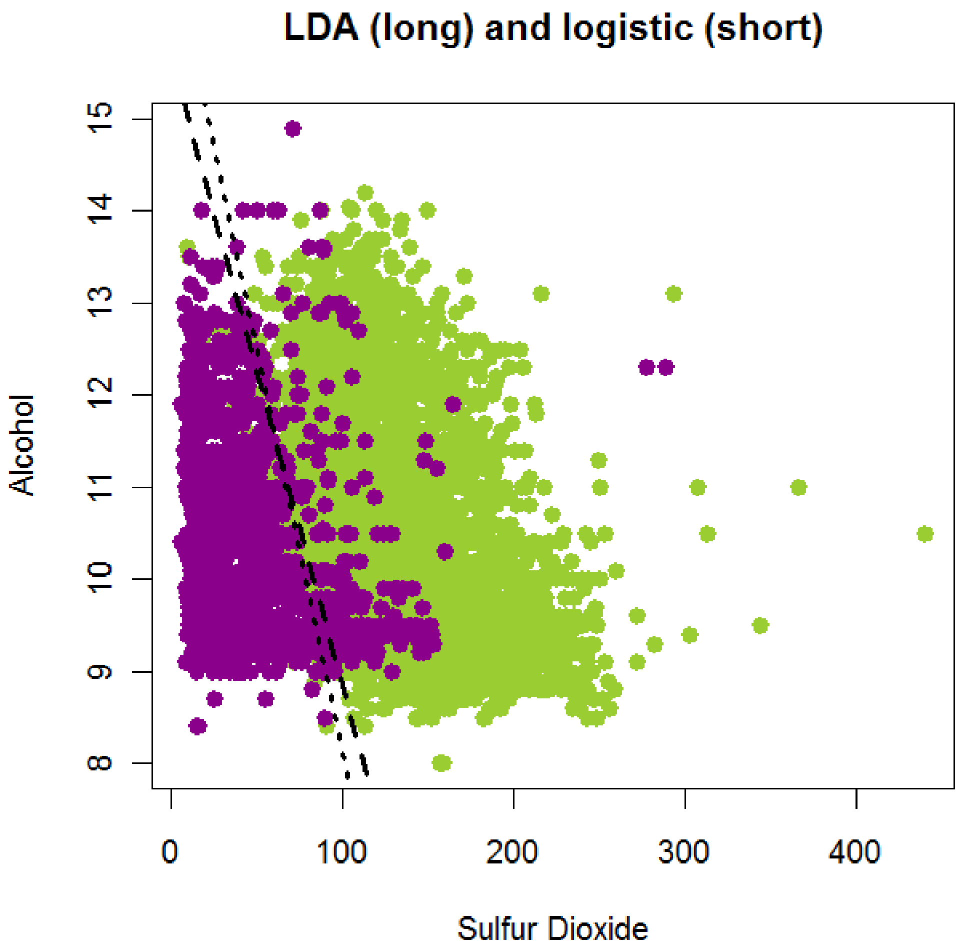
Classification Picture



Classification Picture



Classification Picture



Regression and Classification

Given data

$$\mathcal{D} = \{ (\mathbf{x}_i, y_i), i=1,\ldots,n\}$$

where $\mathbf{x}_i = (x_{i1},\ldots,x_{ip})$, build a model f-hat so that Y-hat = f-hat ($\mathbf{X}$) for random variables $\mathbf{X} = (X_1,\ldots,X_p)$ and Y.

Then f-hat will be used for:

- Predicting the value of the response from the predictors: y_0-hat = f-hat($\mathbf{x}_0$) where $\mathbf{x}_0 = (x_{o1},\ldots,x_{op})$.
- Understanding the relationship between the predictors and the response.

Assumptions

- Independent observations
  - Not autocorrelated over time or space
  - Not usually from a designed experiment
  - Not matched case-control
- Goal is prediction and (sometimes) understanding
  - Which predictors are useful? How? Where?
  - Is there "interesting" structure?

Predictive Accuracy

- Regression
  - Expected mean squared error
- Classification
  - Expected (classwise) error rate

Estimates of Predictive Accuracy

- Resubstitution
  - Use the accuracy on the training set as an estimate of generalization error.
- AIC etc
  - Use assumptions about model.
- Crossvalidation
  - Randomly select a training set, use the rest as the test set.
  - 10-fold crossvalidation.

10-Fold Crossvalidation

Divide the data at random into 10 pieces, D_1,…,D_{10}.

- Fit the predictor to D_2,…,D_{10}; predict D_1.
- Fit the predictor to D_1,D_3,…,D_{10}; predict D_2.
- Fit the predictor to D_1,D_2,D_4,…,D_{10}; predict D_3.
- …
- Fit the predictor to D_1,D_2,…,D_9; predict D_{10}.

Compute the estimate using the assembled predictions and their observed values.

Estimates of Predictive Accuracy

Typically, resubstitution estimates are optimistic compared to crossvalidation estimates.

Crossvalidation estimates tend to be pessimistic because they are based on smaller samples.

Random Forests has its own way of estimating predictive accuracy (“out-of-bag” estimates).

Case Study: Cavity Nesting birds in the Uintah Mountains, Utah

- Red-naped sapsucker (*Sphyrapicus nuchalis*) (*n* = 42 nest sites)

© Jack Murray/CLO

- Mountain chickadee
- (*Parus gambeli*) (*n* = 42 nest sites)

- Northern flicker (*Colaptes auratus*) (*n* = 23 nest sites)

- *n* = 106 non-nest sites

Case Study: Cavity Nesting birds in the Uintah Mountains, Utah

- Response variable is the presence (coded 1) or absence (coded 0) of a nest.
- Predictor variables (measured on 0.04 ha plots around the sites) are:
  - Numbers of trees in various size classes from less than 1 inch in diameter at breast height to greater than 15 inches in diameter.
  - Number of snags and number of downed snags.
  - Percent shrub cover.
  - Number of conifers.
  - Stand Type, coded as 0 for pure aspen and 1 for mixed aspen and conifer.

Assessing Accuracy in Classification

Actual Class	**Predicted Class**		**Total**
	Absence	Presence	
	0	1	
Absence, 0	a	b	a+b
Presence, 1	c	d	c+d
Total	a+c	b+d	n

$$Specificity = 100\% \times \frac{a}{a+b} \qquad Sensitivity = 100\% \times \frac{d}{c+d} \qquad PCC = 100\% \times \frac{a+d}{n}$$

$$\kappa = \frac{(Observed\ agreement) - (Chance\ agreement)}{1 - (Chance\ agreement)}$$

$$Chance\ agreement = \frac{a+b}{n} \times \frac{a+c}{n} + \frac{c+d}{n} \times \frac{b+d}{n} \qquad Observed\ agreement = \frac{a+d}{n}$$

Assessing Accuracy in Classification

Actual Class	**Predicted Class**		**Total**
	Absence	Presence	
	0	1	
Absence, 0	a	b	a+b
Presence, 1	c	d	c+d
Total	a+c	b+d	n

Error rate = (c + b) / n

Resubstitution Accuracy (fully grown tree)

Actual Class	Predicted Class		Total
	Absence	Presence	
	0	1	
Absence, 0	105	1	106
Presence, 1	0	107	107
Total	105	108	213

Error rate = (0 + 1)/213 = (approx) 0.005 or 0.5%

Crossvalidation Accuracy (fully grown tree)

Actual Class	**Predicted Class**		**Total**
	Absence	Presence	
	0	1	
Absence, 0	83	23	106
Presence, 1	22	85	107
Total	105	108	213

Error rate = (22 + 23)/213 = (approx) .21 or 21%

Outline

- Background.
- Trees.
- Bagging predictors.
- Random Forests algorithm.
- Variable importance.
- Proximity measures.
- Visualization.
- Partial plots and interpretation of effects.

Classification and Regression Trees

Pioneers:

- Morgan and Sonquist (1963).
- **Breiman, Friedman, Olshen, Stone (1984).** ***CART***
- Quinlan (1993). *C4.5*

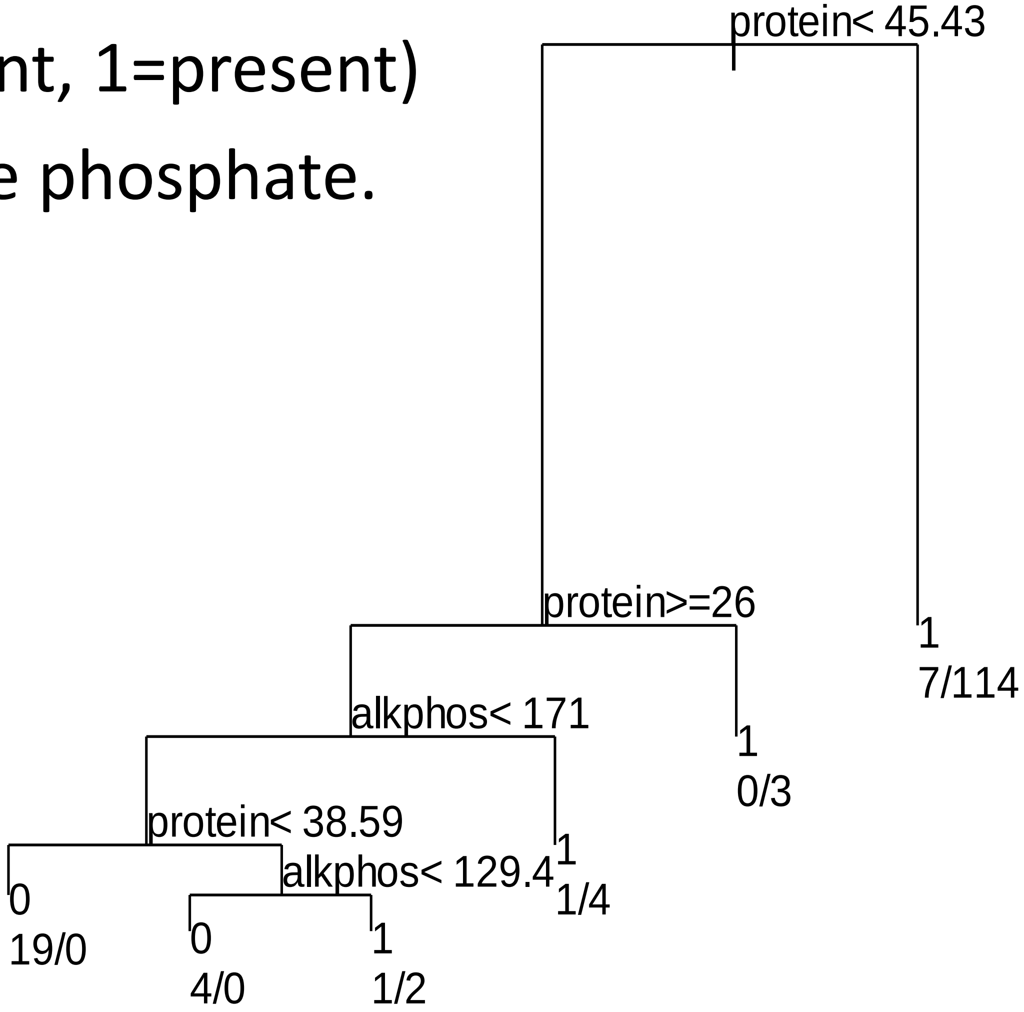
Classification and Regression Trees

- Grow a binary tree.
- At each node, “split” the data into two “daughter” nodes.
- Splits are chosen using a splitting criterion.
- Bottom nodes are “terminal” nodes.
- For regression the predicted value at a node is the *average* response variable for all observations in the node.
- For classification the predicted class is the *most common class* in the node (majority vote).
- For classification trees, can also get estimated probability of membership in each of the classes

A Classification Tree

Predict hepatitis (0=absent, 1=present) using protein and alkaline phosphate.

“Yes” goes left.



Splitting criteria

- **Regression**: residual sum of squares

  $$RSS = \sum_{left} (y_i - y_L{*})^2 + \sum_{right} (y_i - y_R{*})^2$$

  where $y_L{*}$ = mean y-value for left node

  $y_R{*}$ = mean y-value for right node

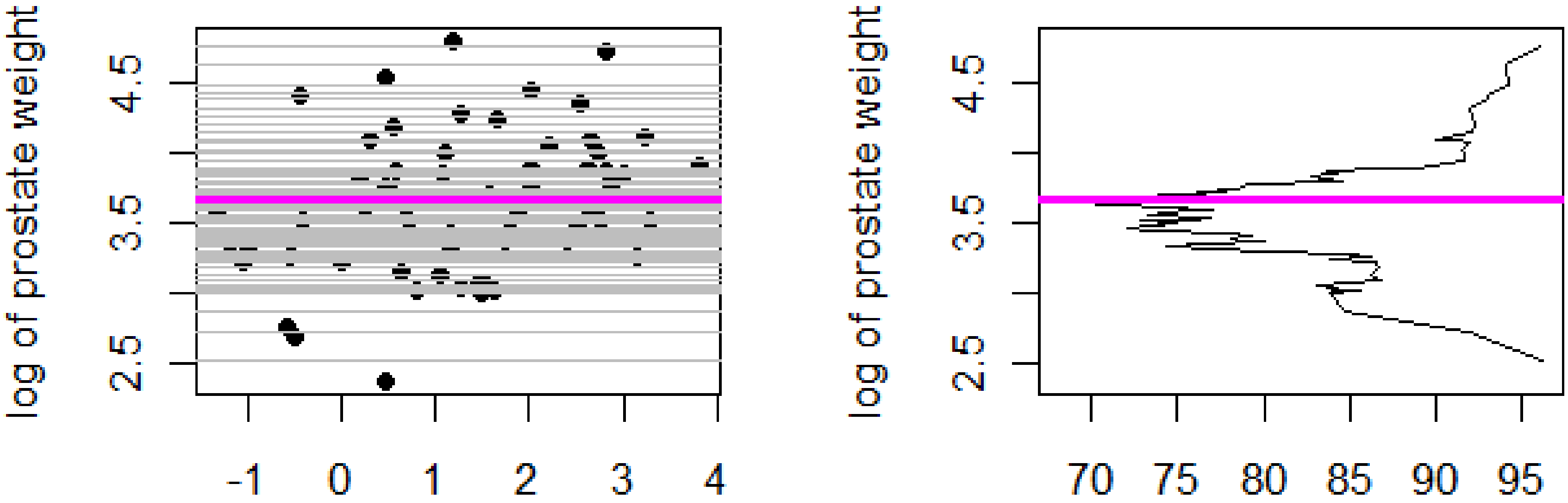
- **Classification**: Gini criterion

  $$Gini = N_L \sum_{k=1,\ldots,K} p_{kL} (1- p_{kL}) + N_R \sum_{k=1,\ldots,K} p_{kR} (1- p_{kR})$$

  where p_{kL} = proportion of class k in left node

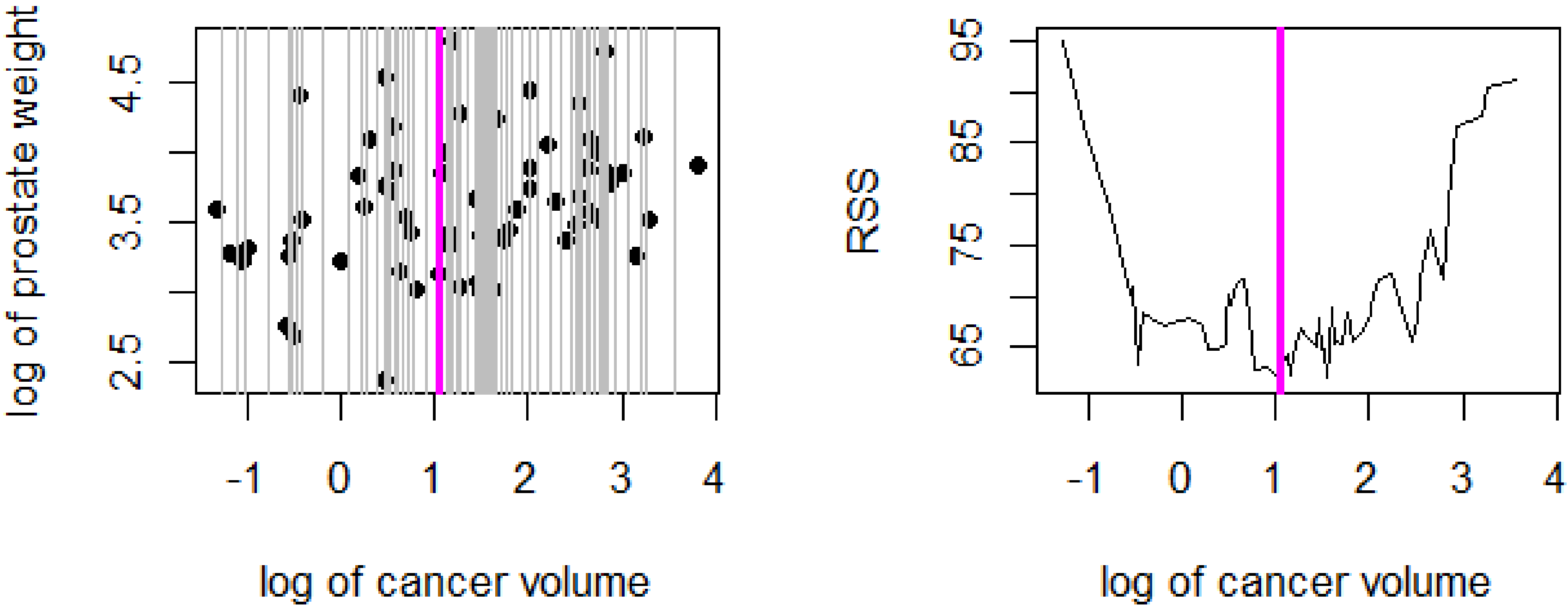
  p_{kR} = proportion of class k in right node

Choosing the best horizontal split



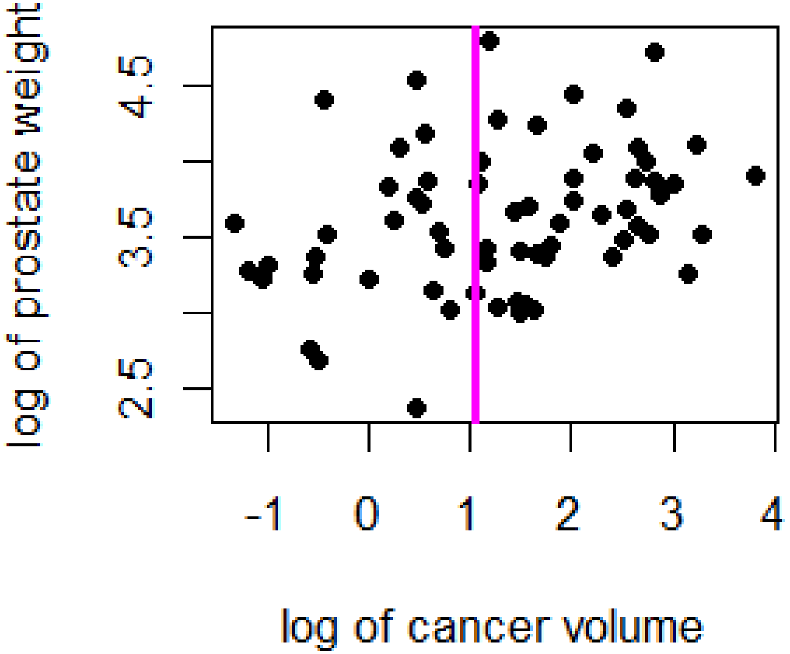
Best horizontal split is at 3.67 with RSS = 68.09.

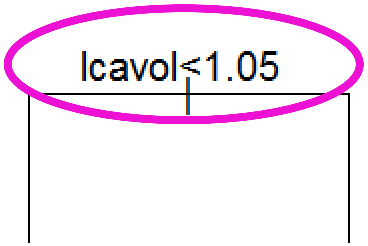
Choosing the best vertical split



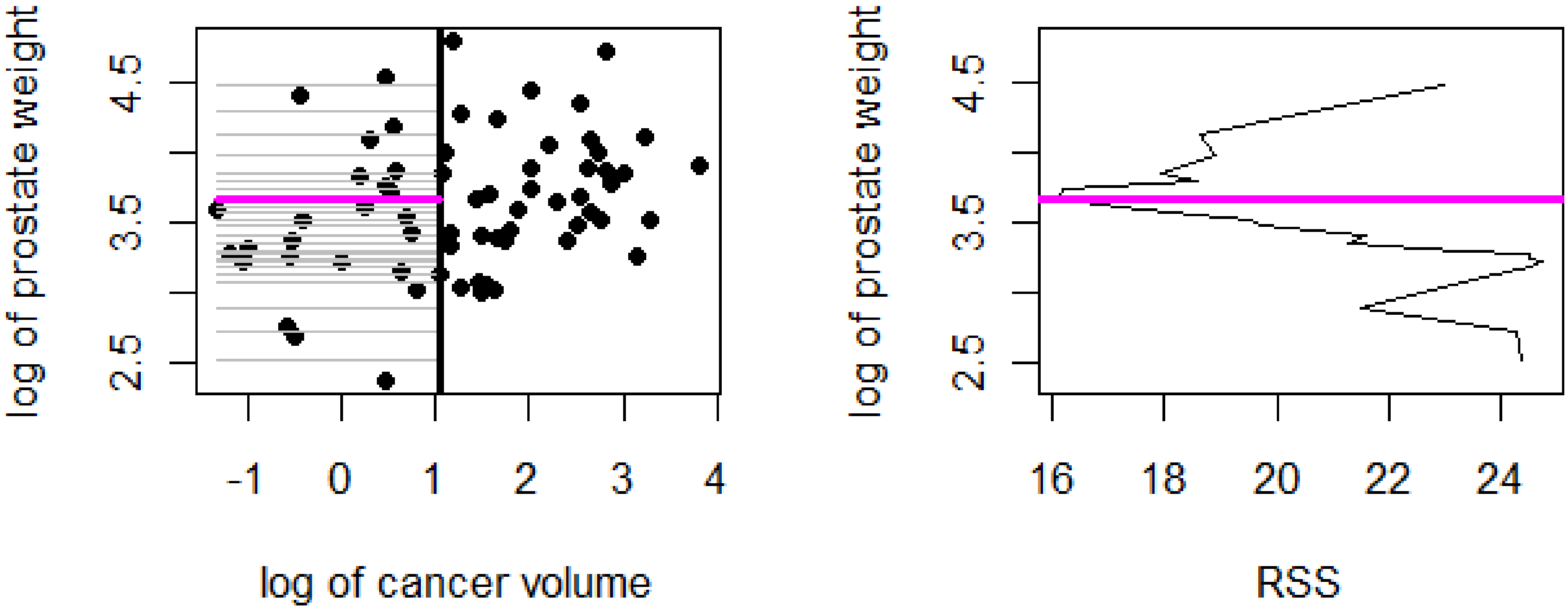
Best vertical split is at 1.05 with RSS = 61.76.

Regression tree (prostate cancer)



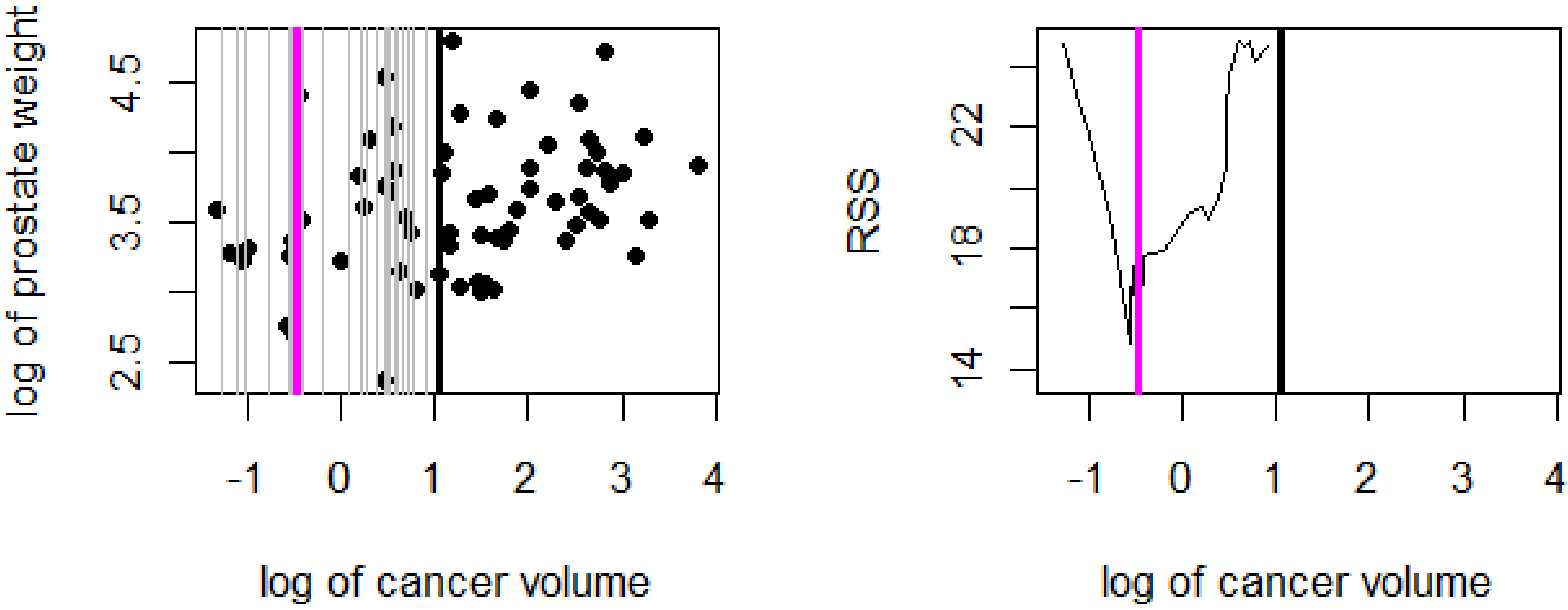


Choosing the best split in the left node



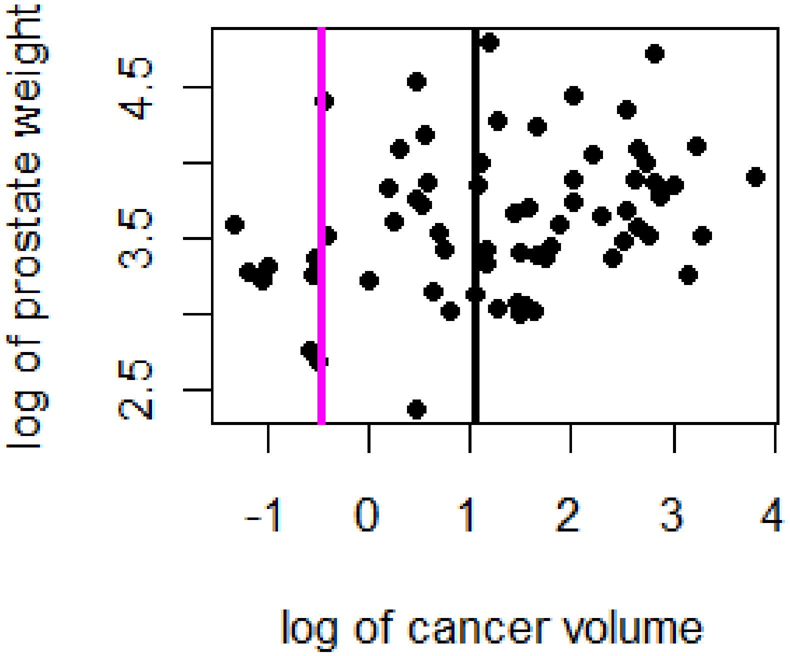
Best horizontal split is at 3.66 with RSS = 16.11.

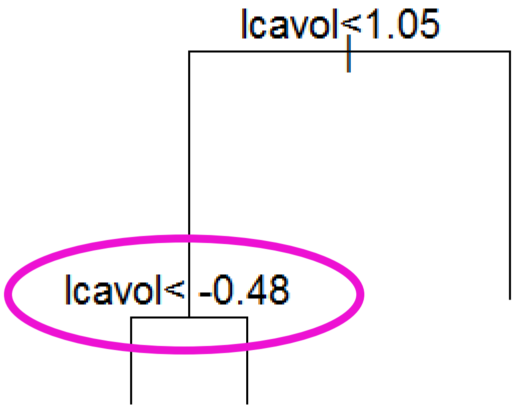
Choosing the best split in the left node



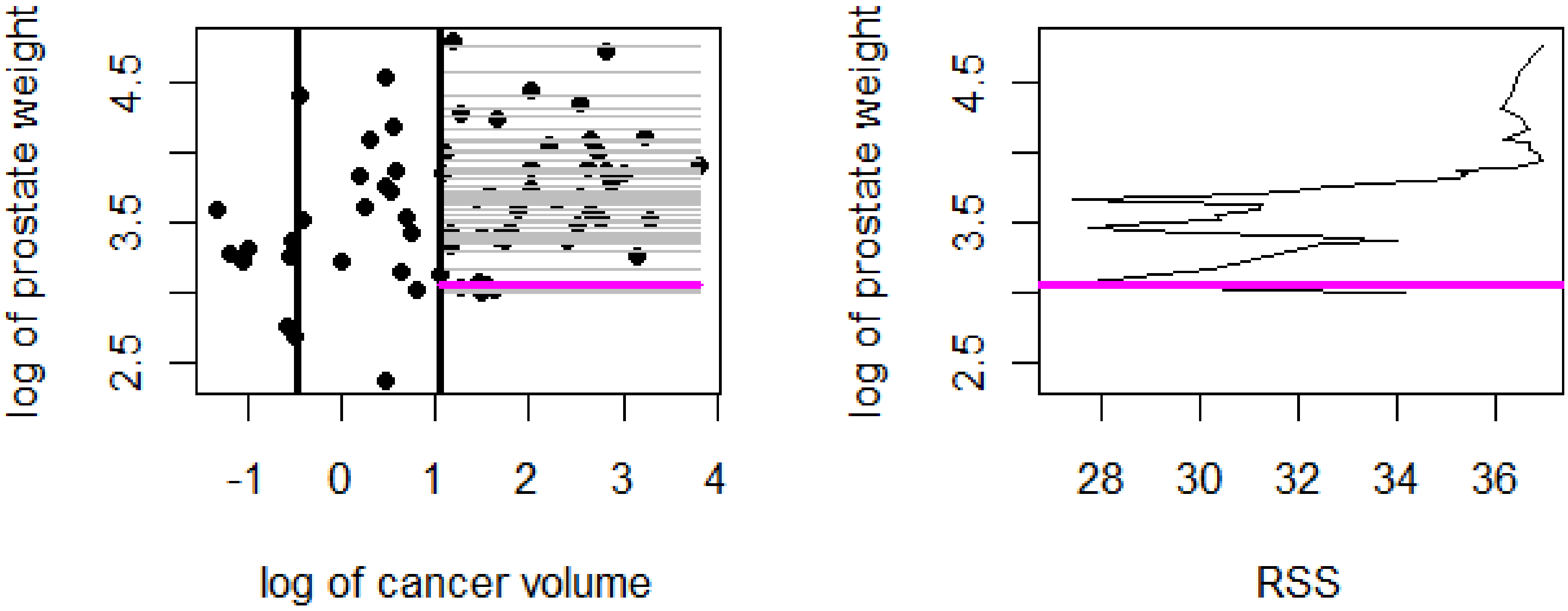
Best vertical split is at -.48 with RSS = 13.61.

Regression tree (prostate cancer)



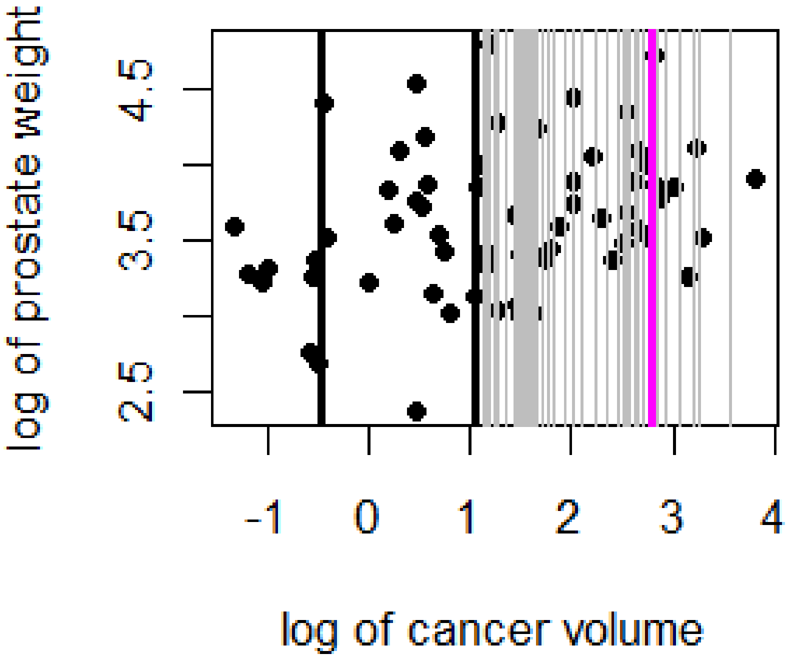


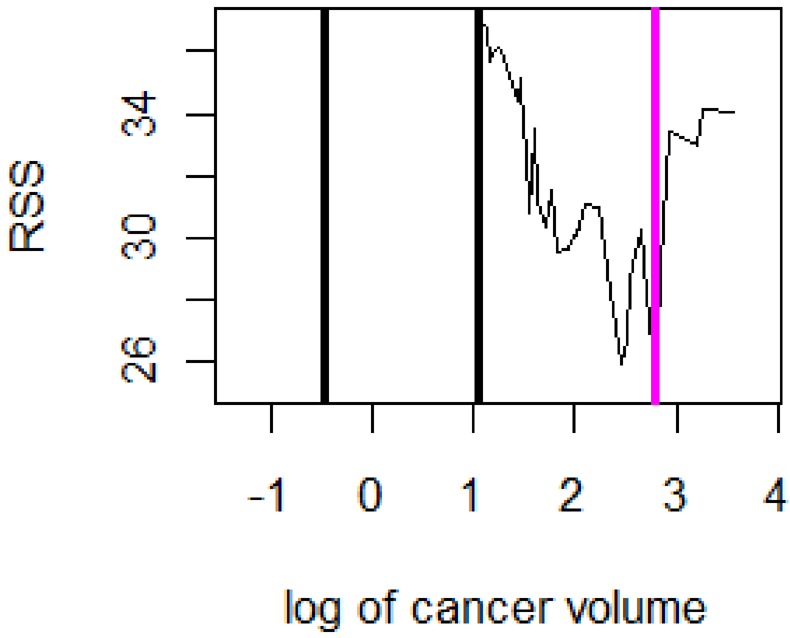
Choosing the best split in the right node



Best horizontal split is at 3.07 with RSS = 27.15.

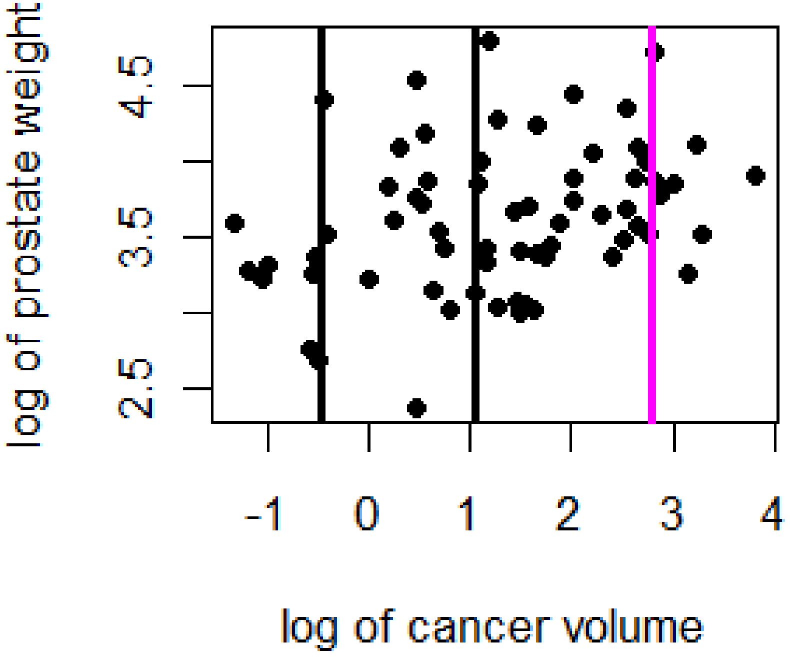
Choosing the best split in the right node

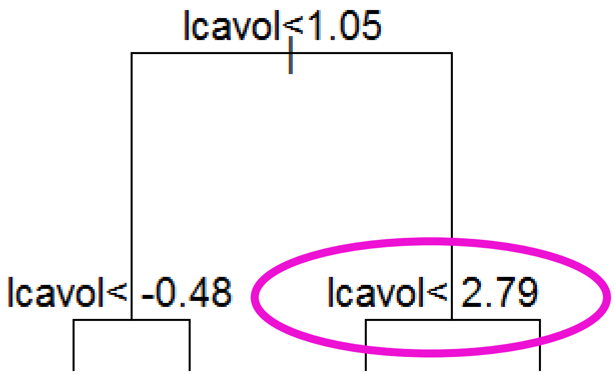




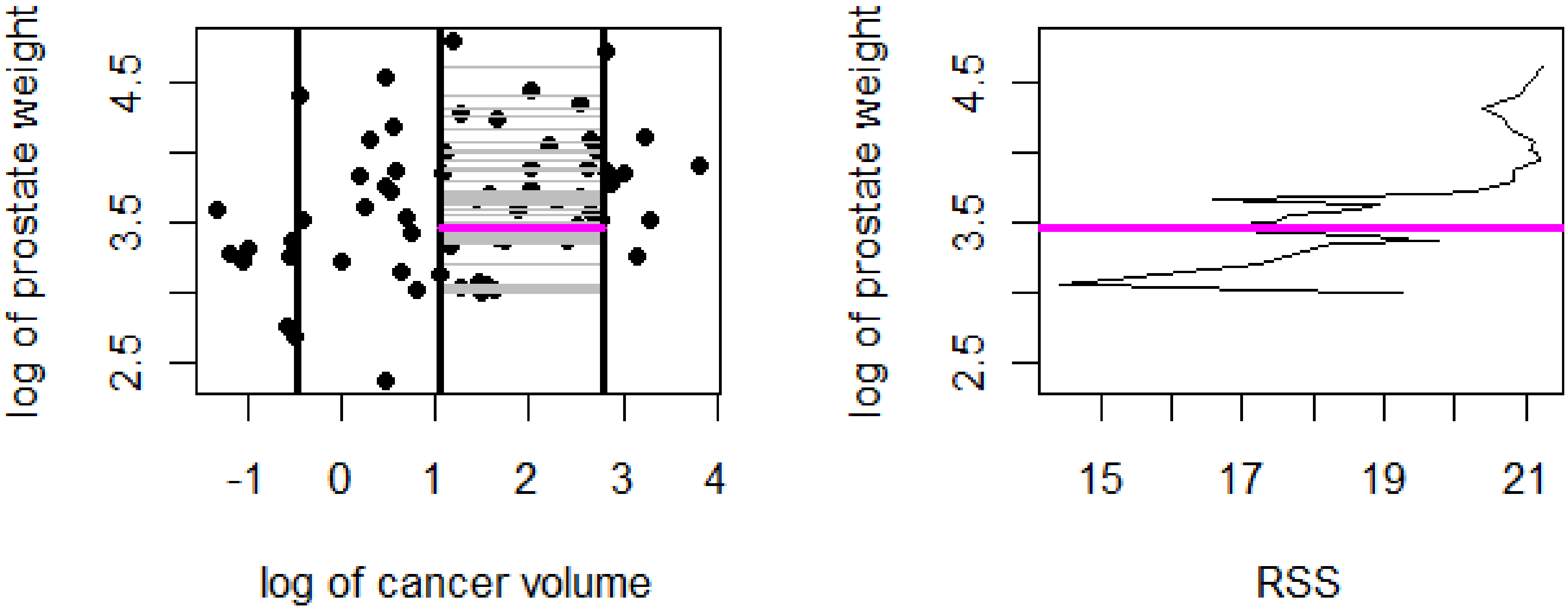
Best vertical split is at 2.79 with RSS = 25.11.

Regression tree (prostate cancer)



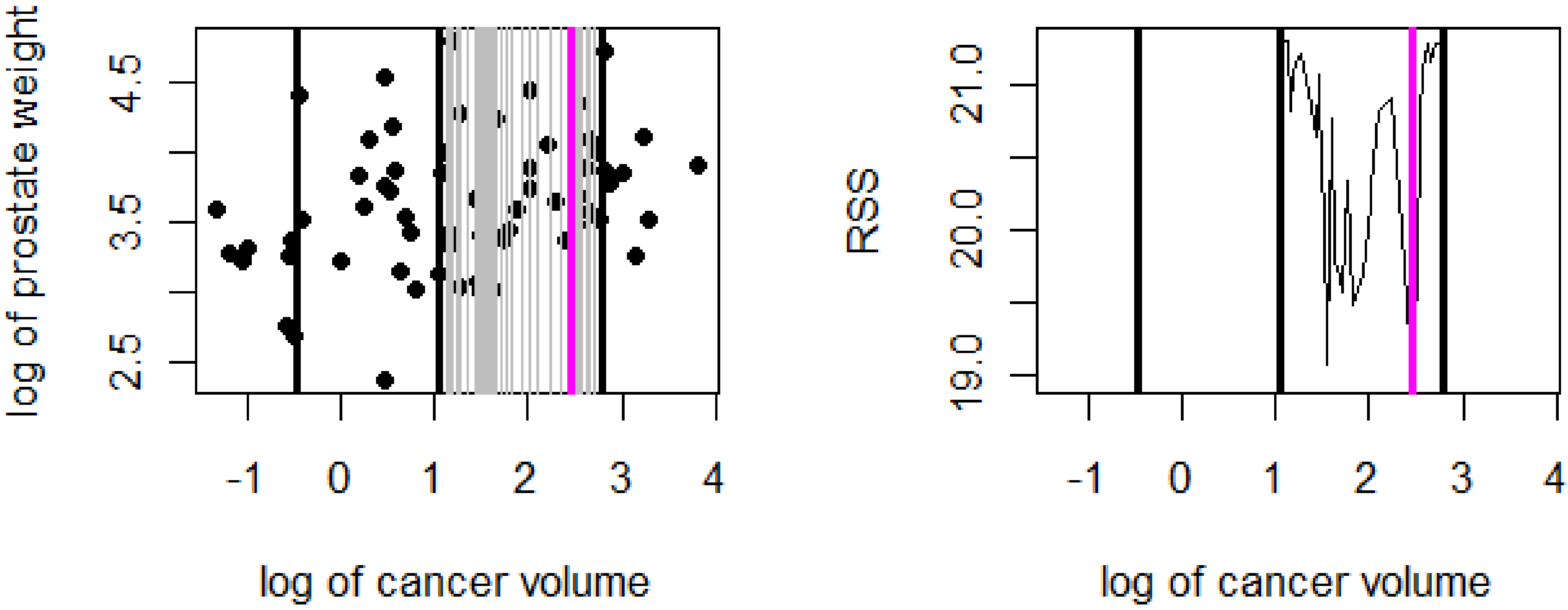


Choosing the best split in the third node



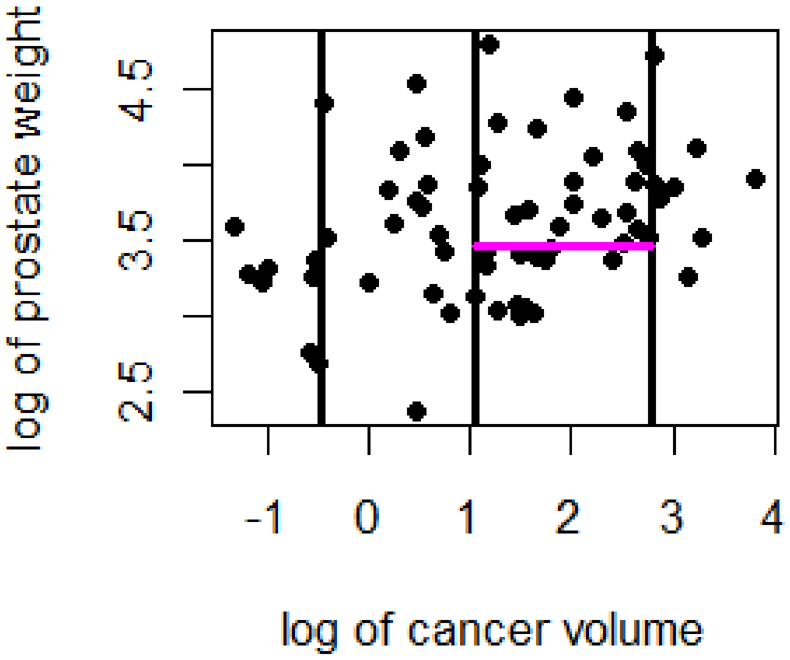
Best horizontal split is at 3.07 with RSS = 14.42, but this is too close to the edge. Use 3.46 with RSS = 16.14.

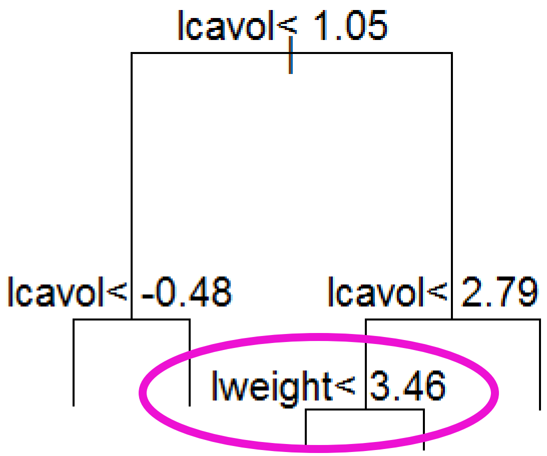
Choosing the best split in the third node



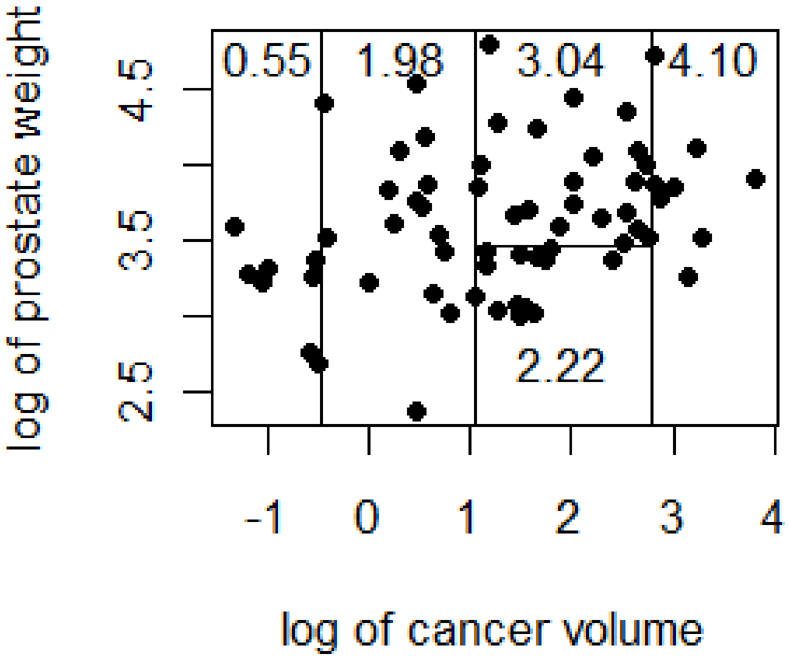
Best vertical split is at 2.46 with RSS = 18.97.

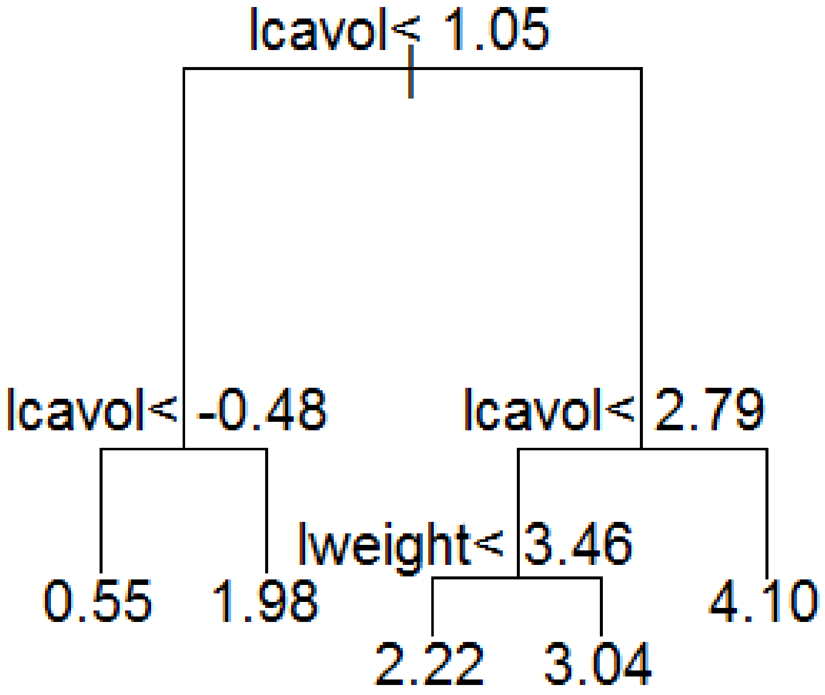
Regression tree (prostate cancer)



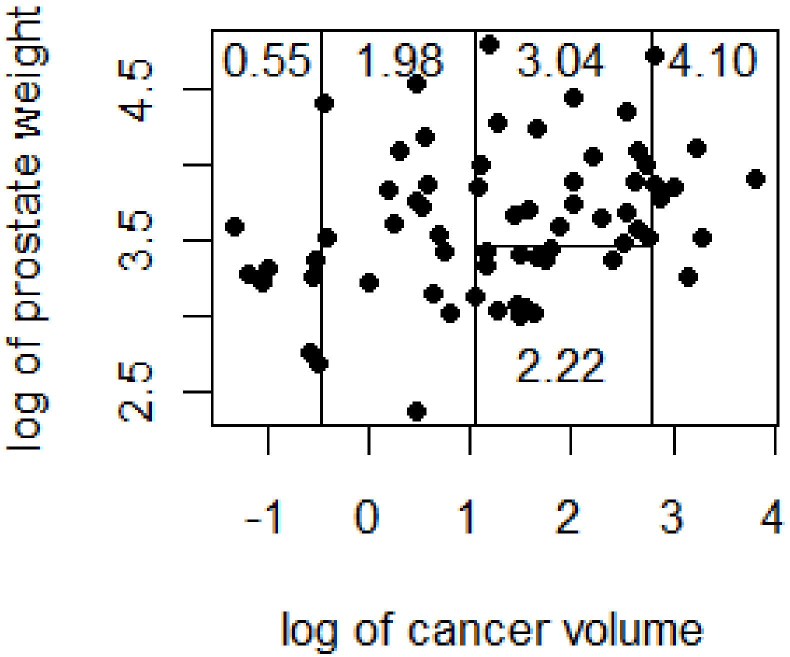


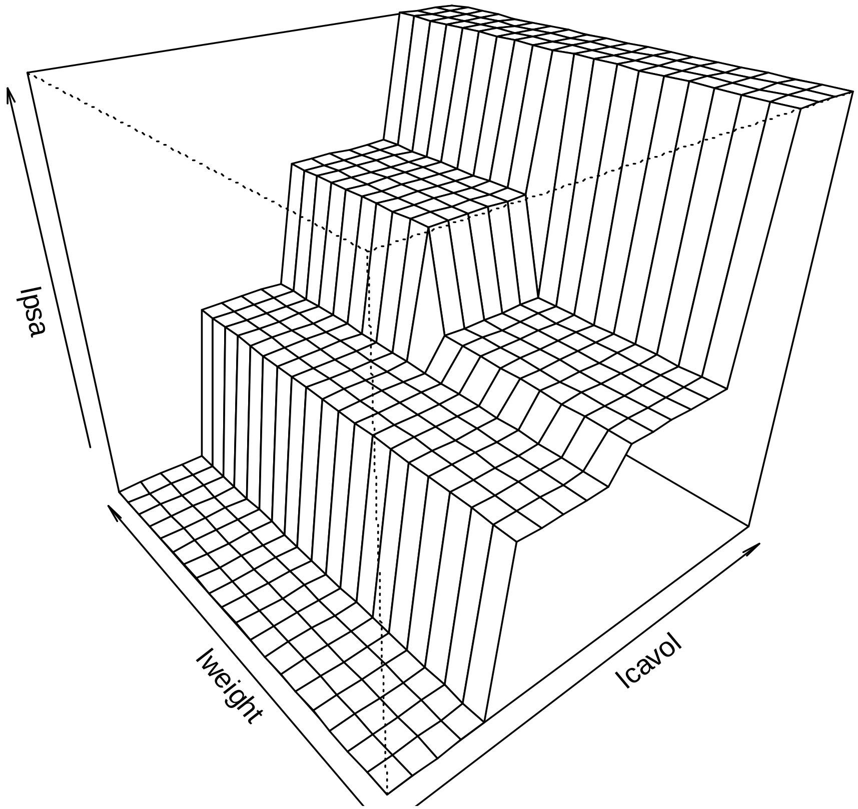
Regression tree (prostate cancer)



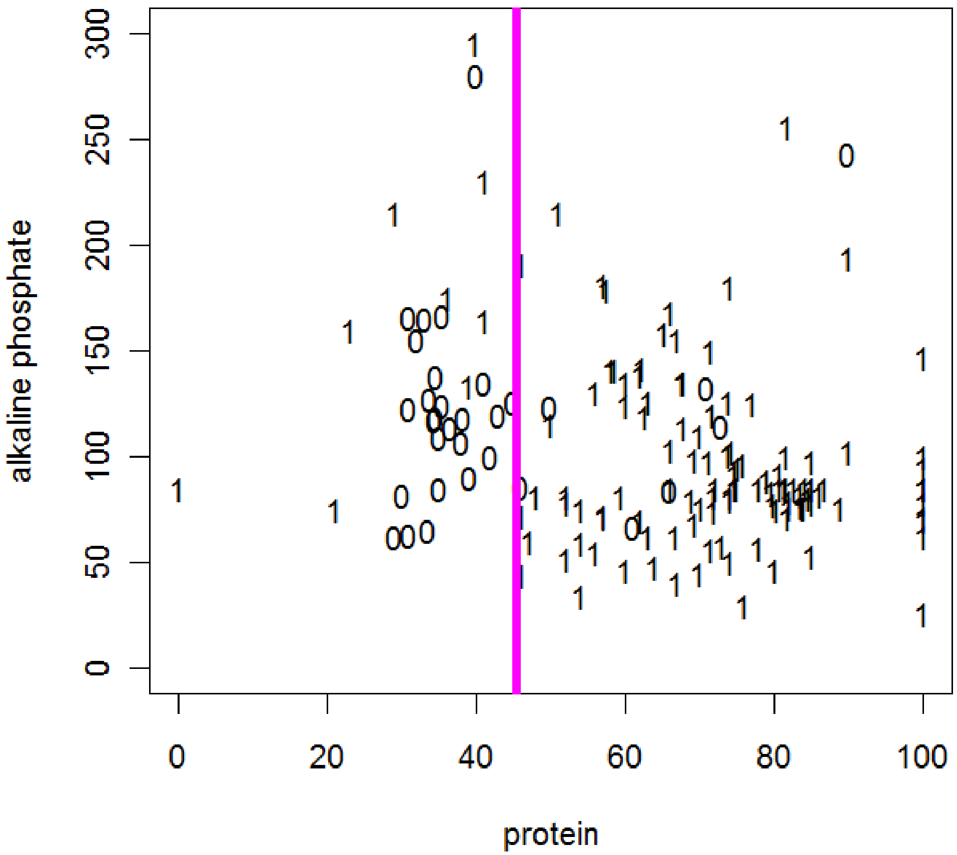


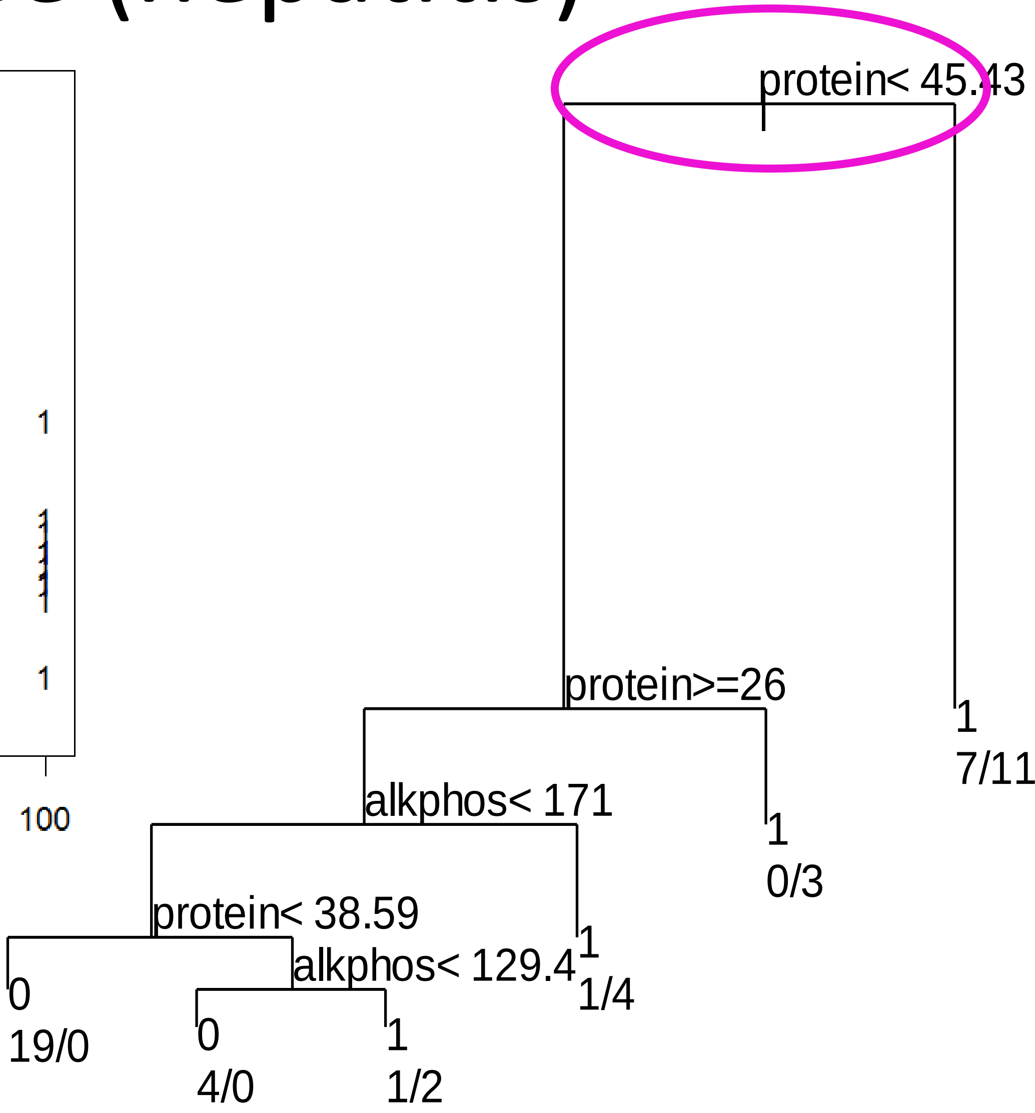
Regression tree (prostate cancer)





Classification tree (hepatitis)




protein< 45.43

protein>=26

1
7/11

alkphos< 171

1
0/3

protein< 38.59

1
1/4

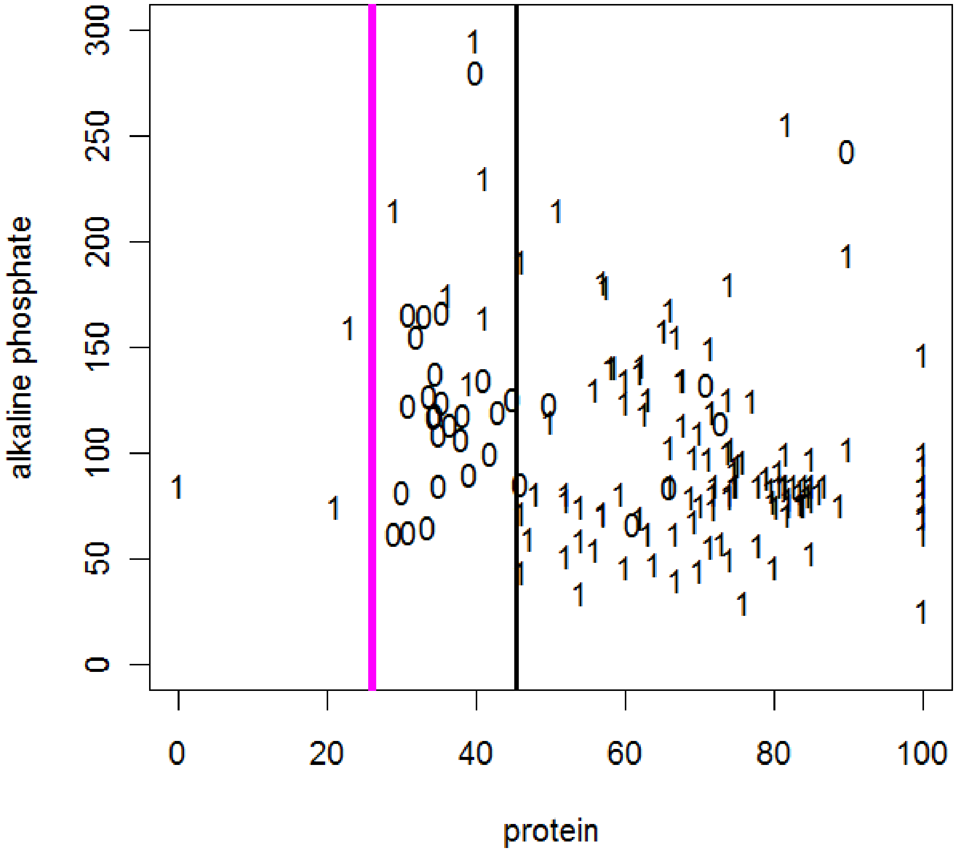
alkphos< 129.4

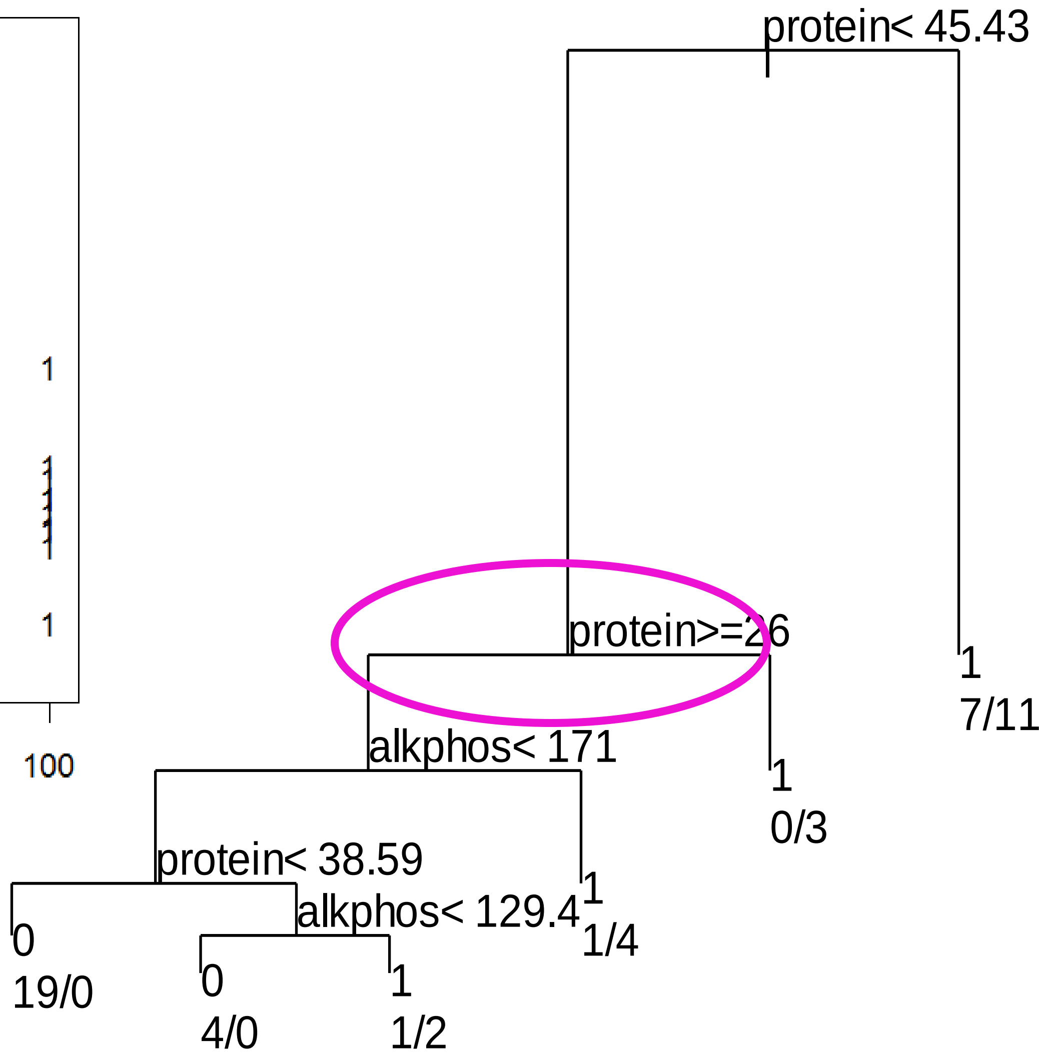
0
19/0

0
4/0

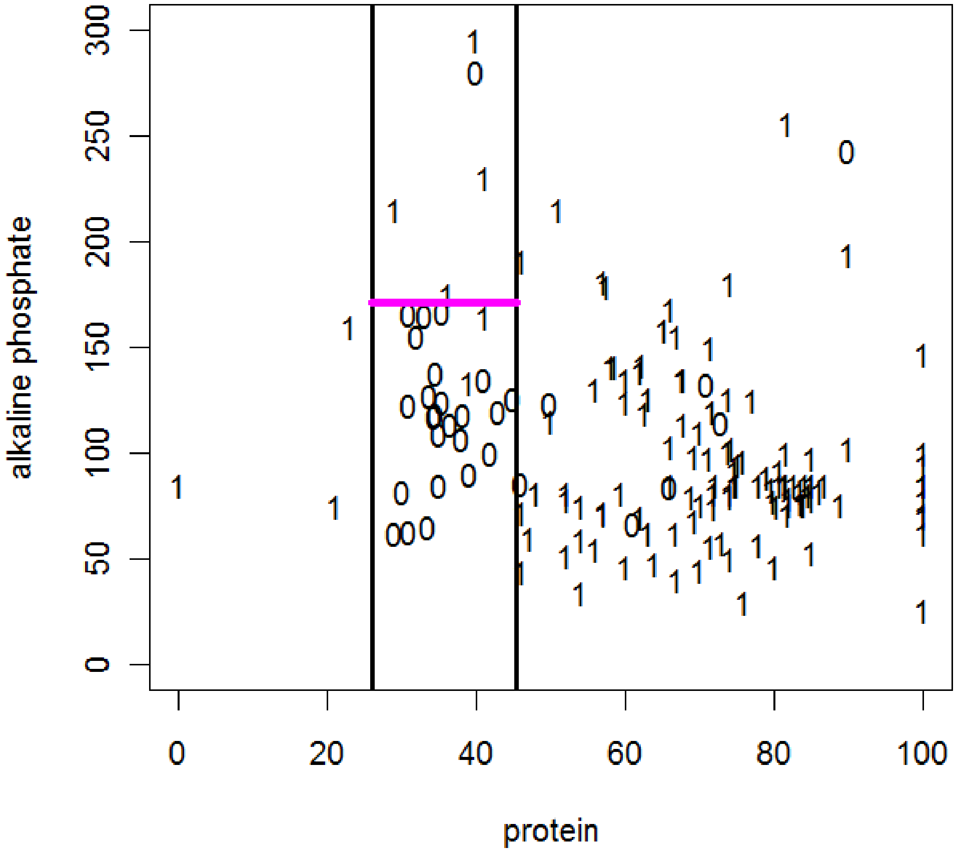
1
1/2

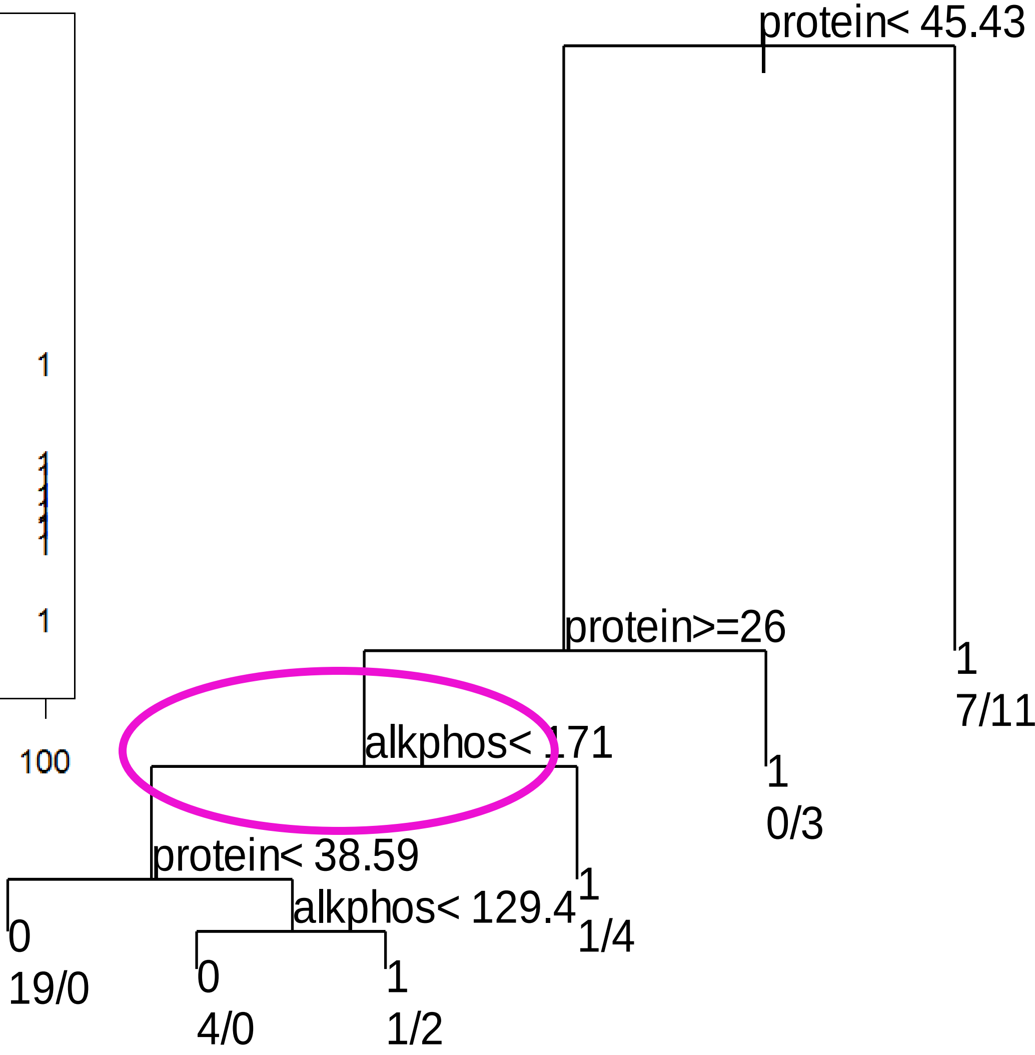
Classification tree (hepatitis)





Classification tree (hepatitis)





protein< 45.43

protein>=26

1
7/11

alkphos< 171

1
0/3

protein< 38.59

1
1/4

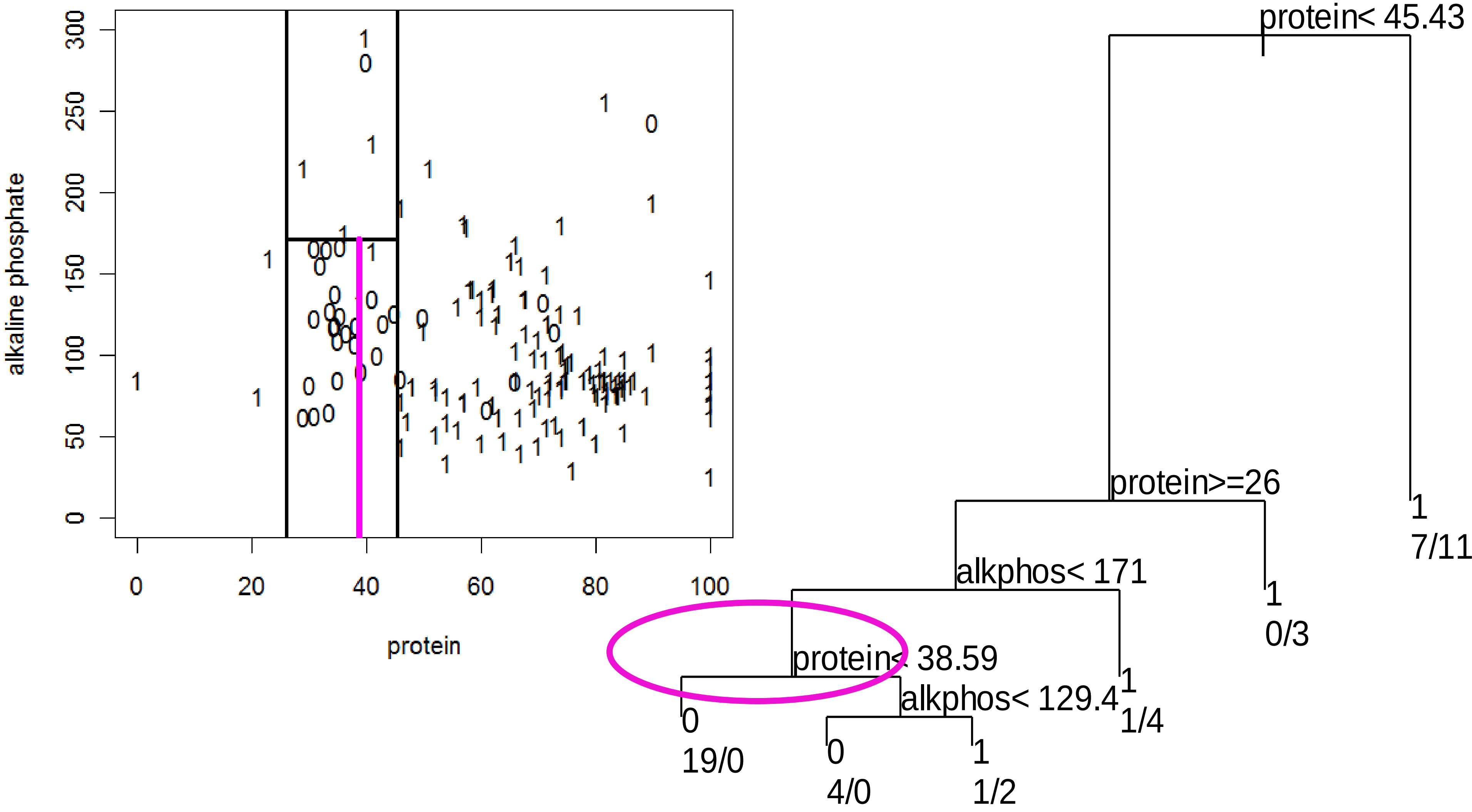
alkphos< 129.4

0
19/0

0
4/0

1
1/2

Classification tree (hepatitis)



protein< 45.43

protein>=26

1
7/11

alkphos< 171

1
0/3

protein< 38.59

1
1/4

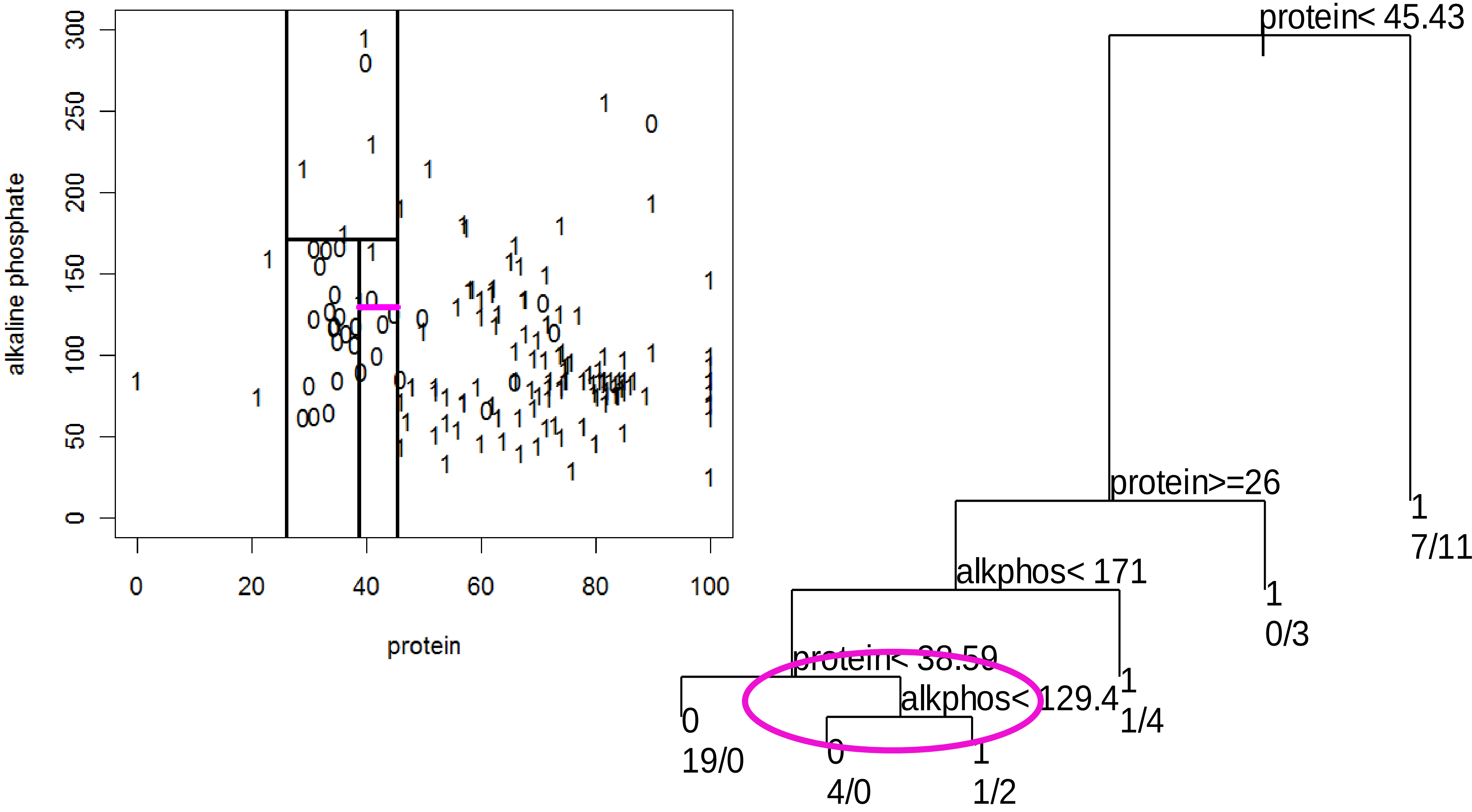
alkphos< 129.4

0
19/0

0
4/0

1
1/2

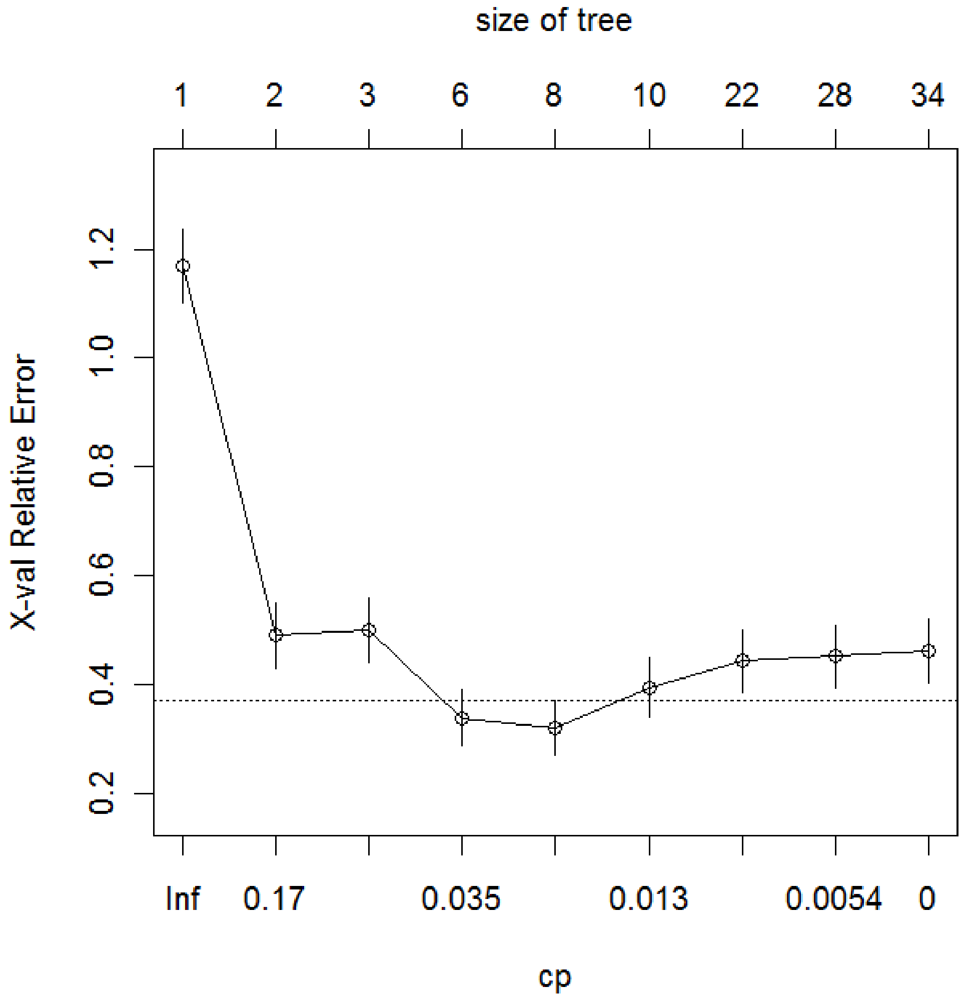
Classification tree (hepatitis)



Pruning

- If the tree is too big, the lower “branches” are modeling noise in the data (“overfitting”).
- The usual paradigm is to grow the trees large and “**prune**” back unnecessary splits.
- Methods for **pruning** trees have been developed. Most use some form of crossvalidation. Tuning may be necessary.

Case Study: Cavity Nesting birds in the Uintah Mountains, Utah



Choose cp = .035

Crossvalidation Accuracy (cp = .035)

Actual Class	Predicted Class		Total
	Absence	Presence	
	0	1	
Absence, 0	85	21	106
Presence, 1	19	88	107
Total	104	109	213

Error rate = (19 + 21)/213 = (approx) .19 or 19%

Classification and Regression Trees

Advantages

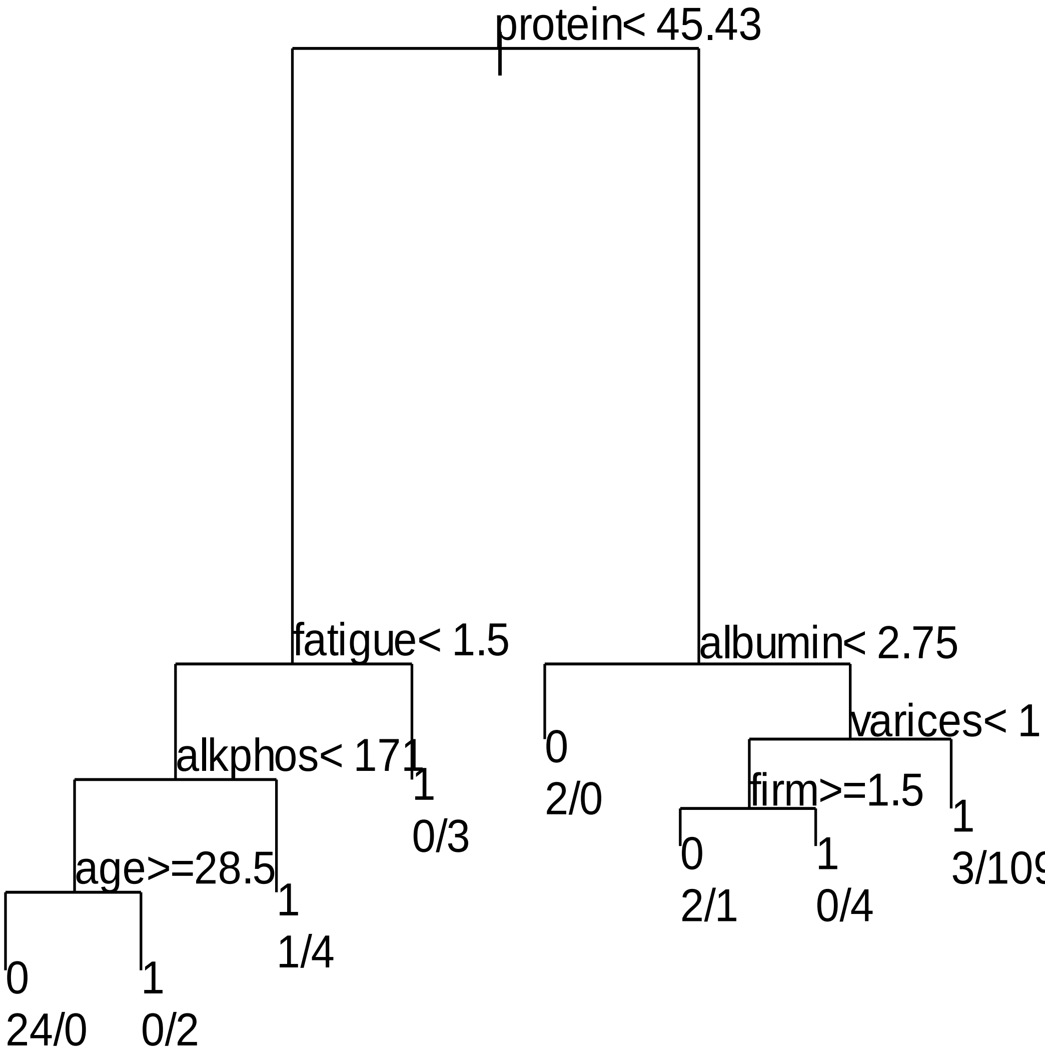
- Applicable to both regression and classification problems.
- Handle categorical predictors naturally.
- Computationally simple and quick to fit, even for large problems.
- No formal distributional assumptions (non-parametric).
- Can handle highly non-linear interactions and classification boundaries.
- Automatic variable selection.
- Handle missing values through surrogate variables.
- Very easy to interpret if the tree is small.

Classification and Regression Trees

Advantages (ctnd)

- The picture of the tree can give valuable insights into which variables are important and where.

- The terminal nodes suggest a natural clustering of data into homogeneous groups.



Classification and Regression Trees

Disadvantages

- *Accuracy* - current methods, such as support vector machines and ensemble classifiers often have 30% lower error rates than CART.
- *Instability* – if we change the data a little, the tree picture can change a lot. So the interpretation is not as straightforward as it appears.

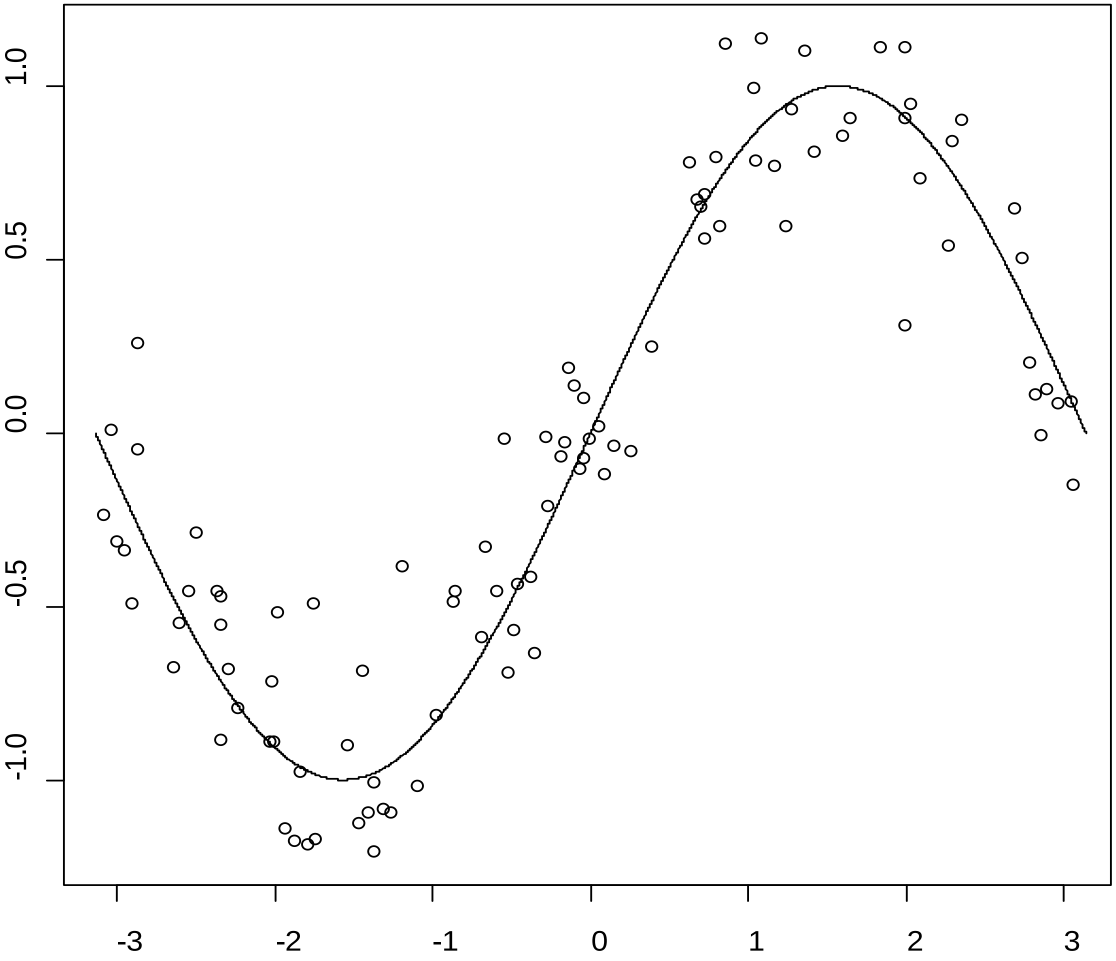
Today, we can do better!

Random Forests

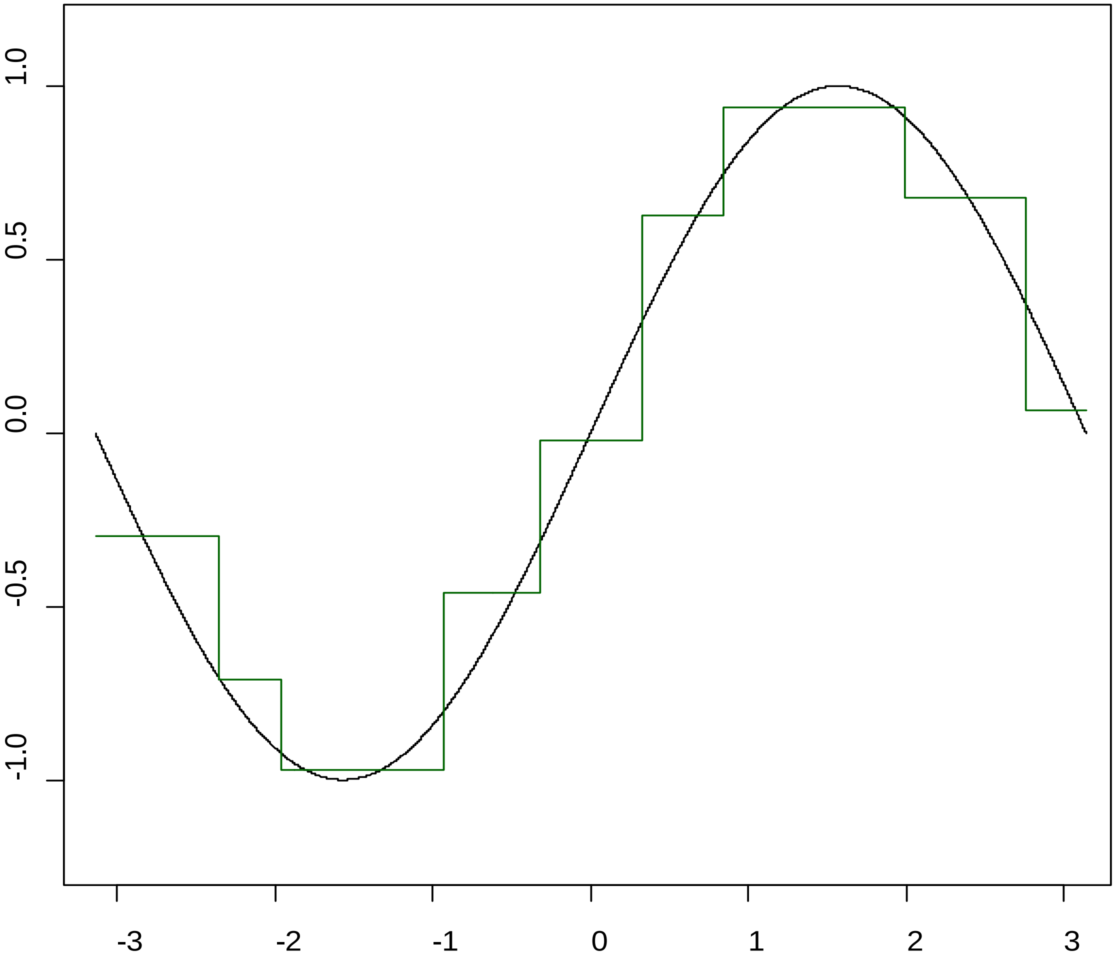
Outline

- Background.
- Trees.
- Bagging predictors.
- Random Forests algorithm.
- Variable importance.
- Proximity measures.
- Visualization.
- Partial plots and interpretation of effects.

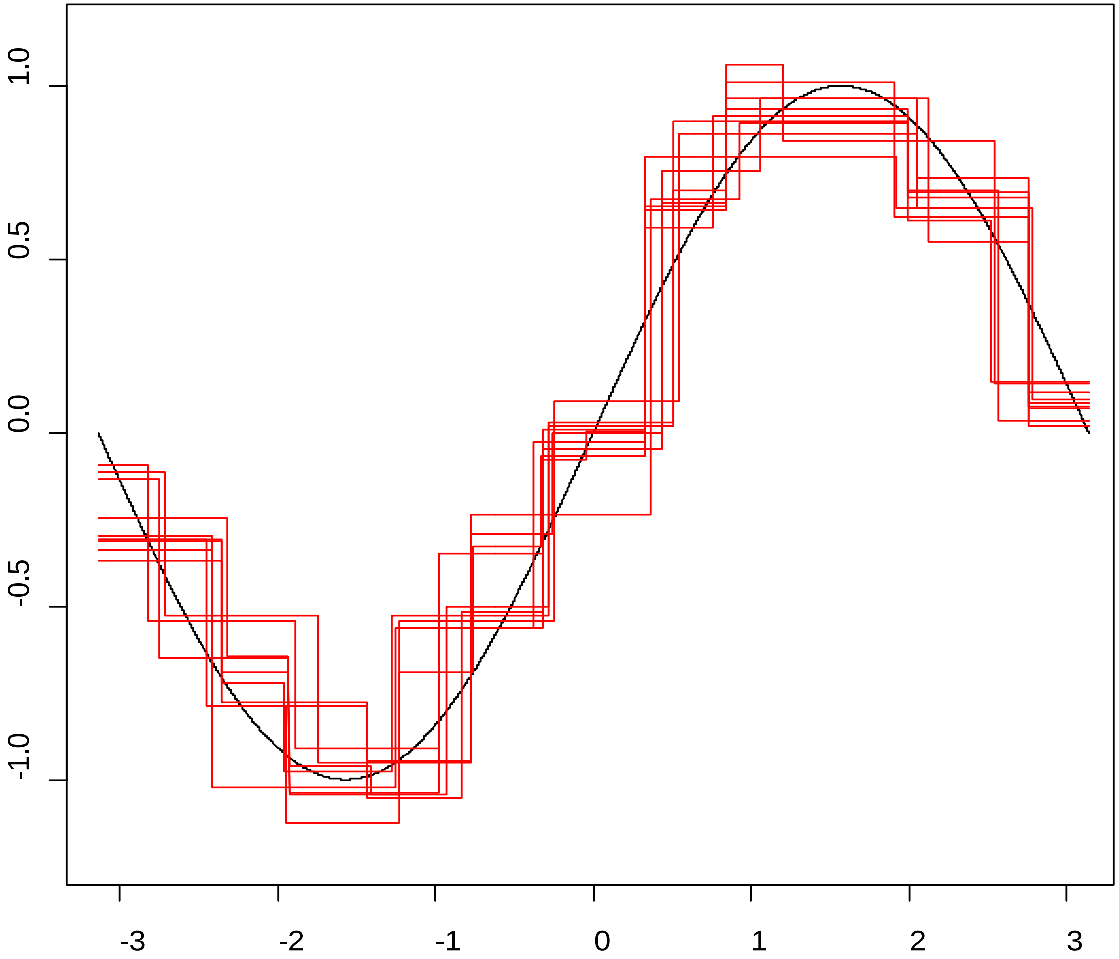
Data and Underlying Function



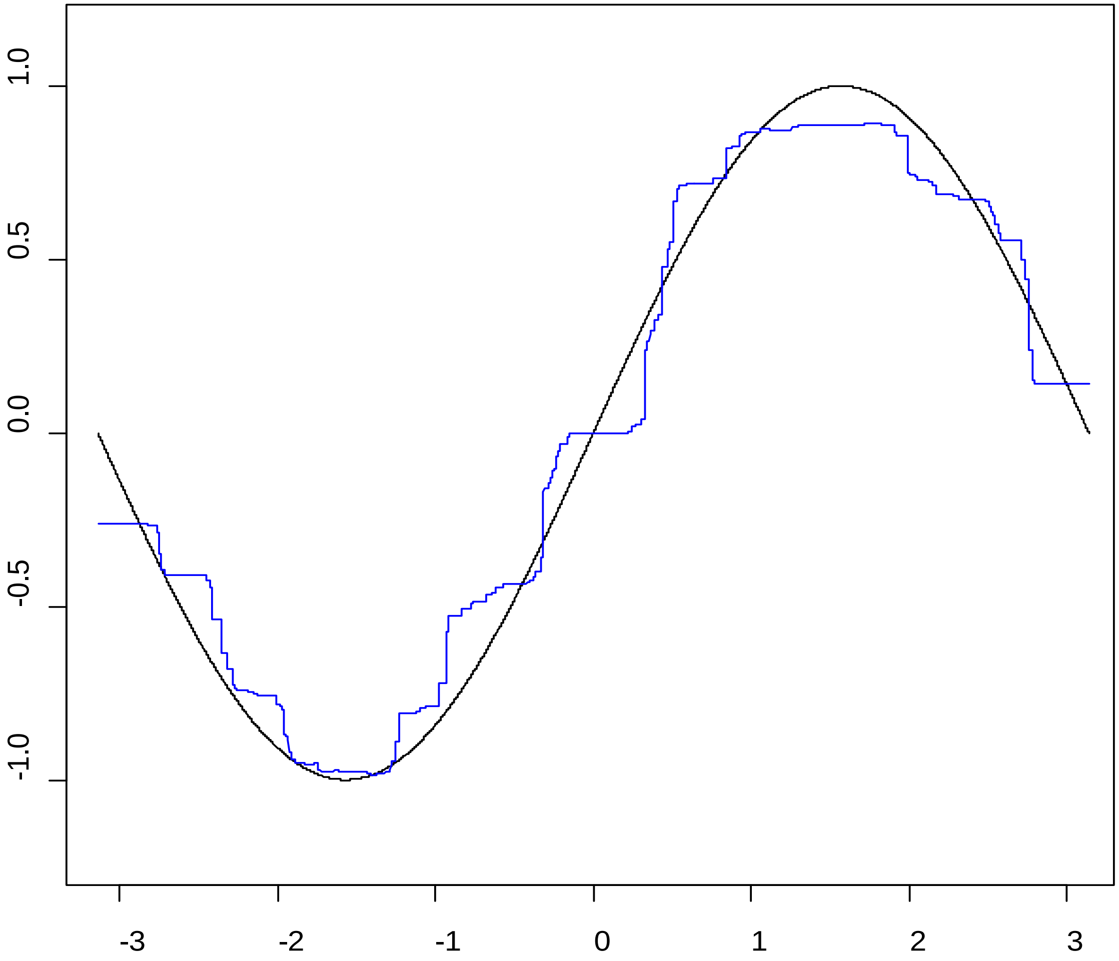
Single Regression Tree



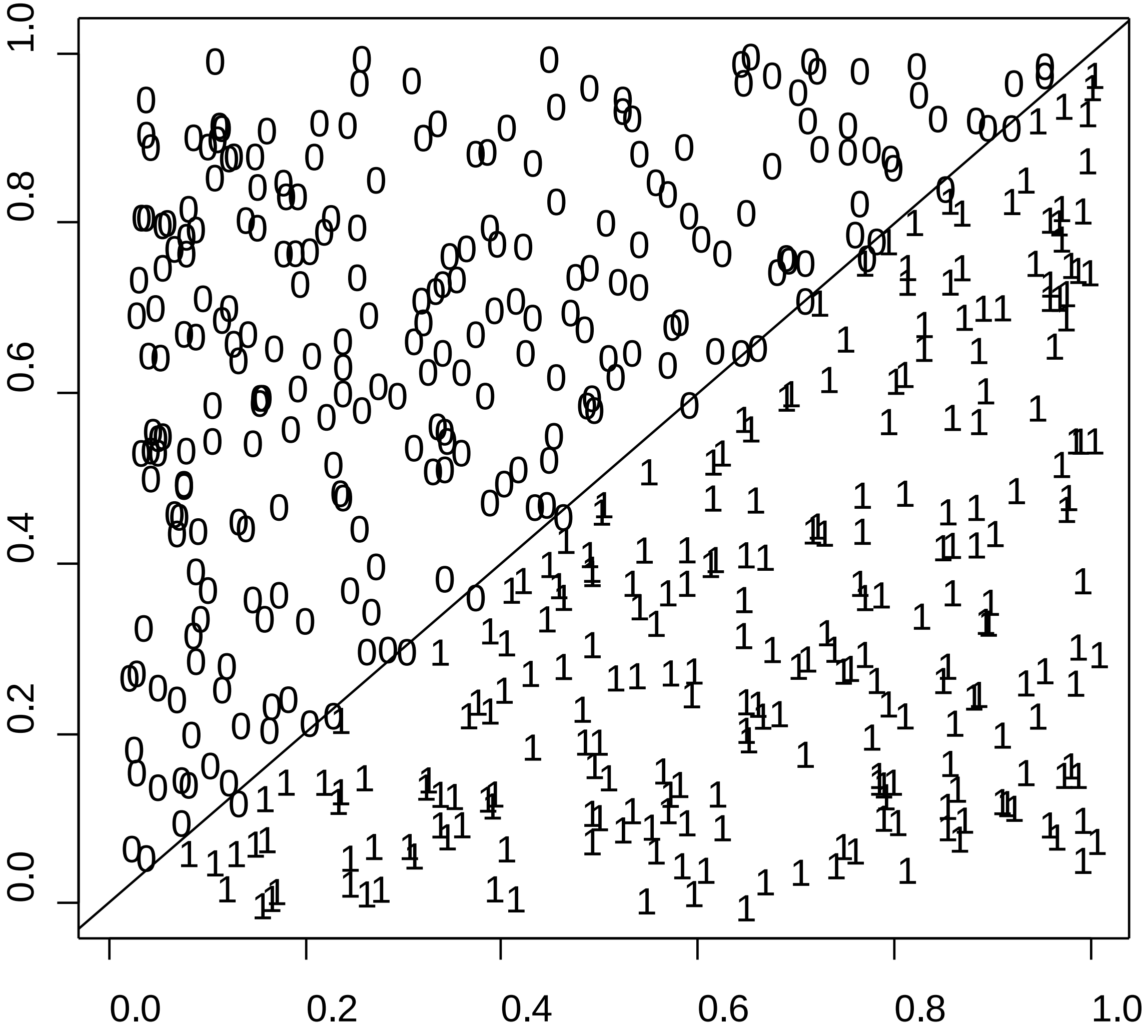
10 Regression Trees



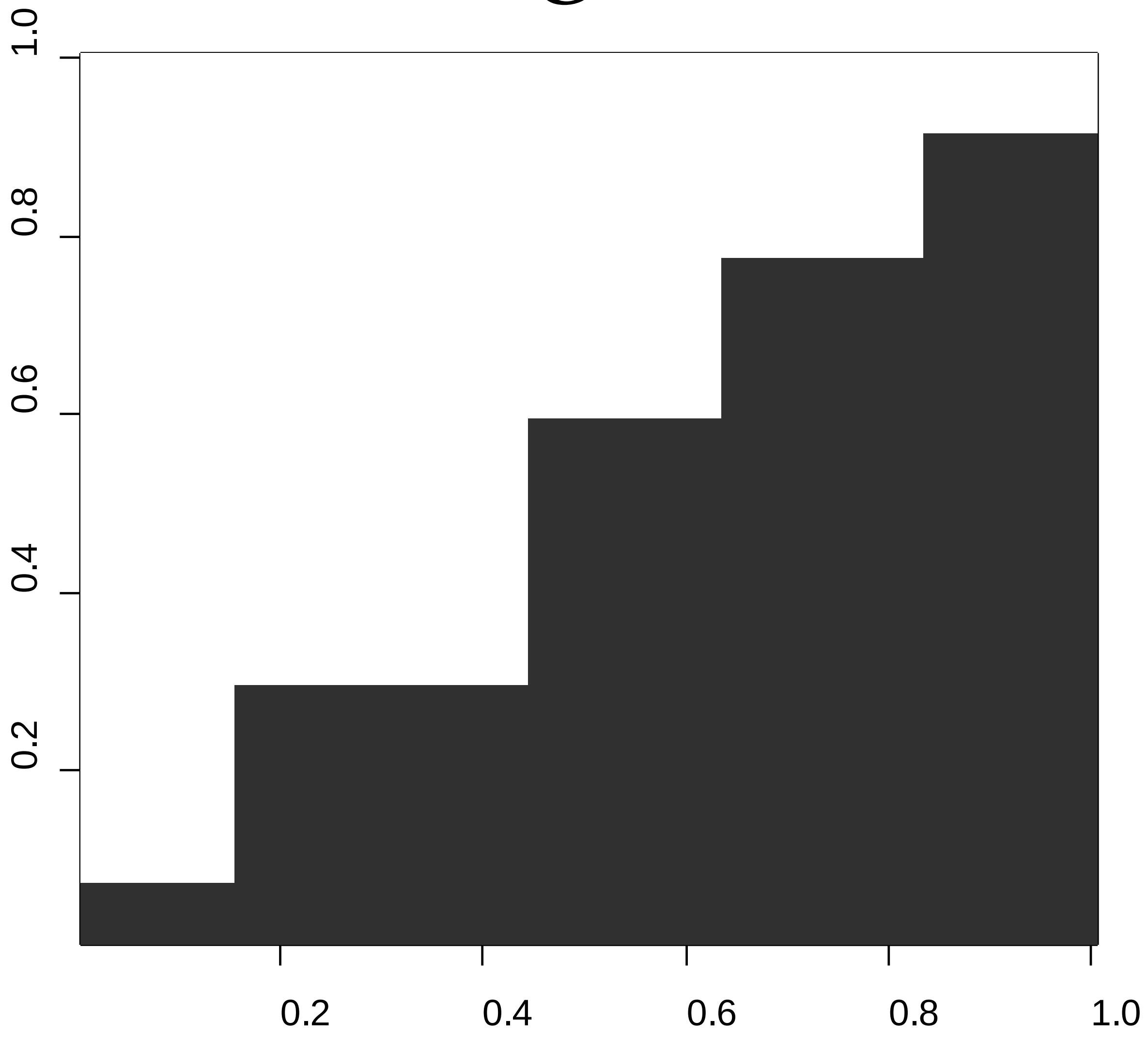
Average of 100 Regression Trees



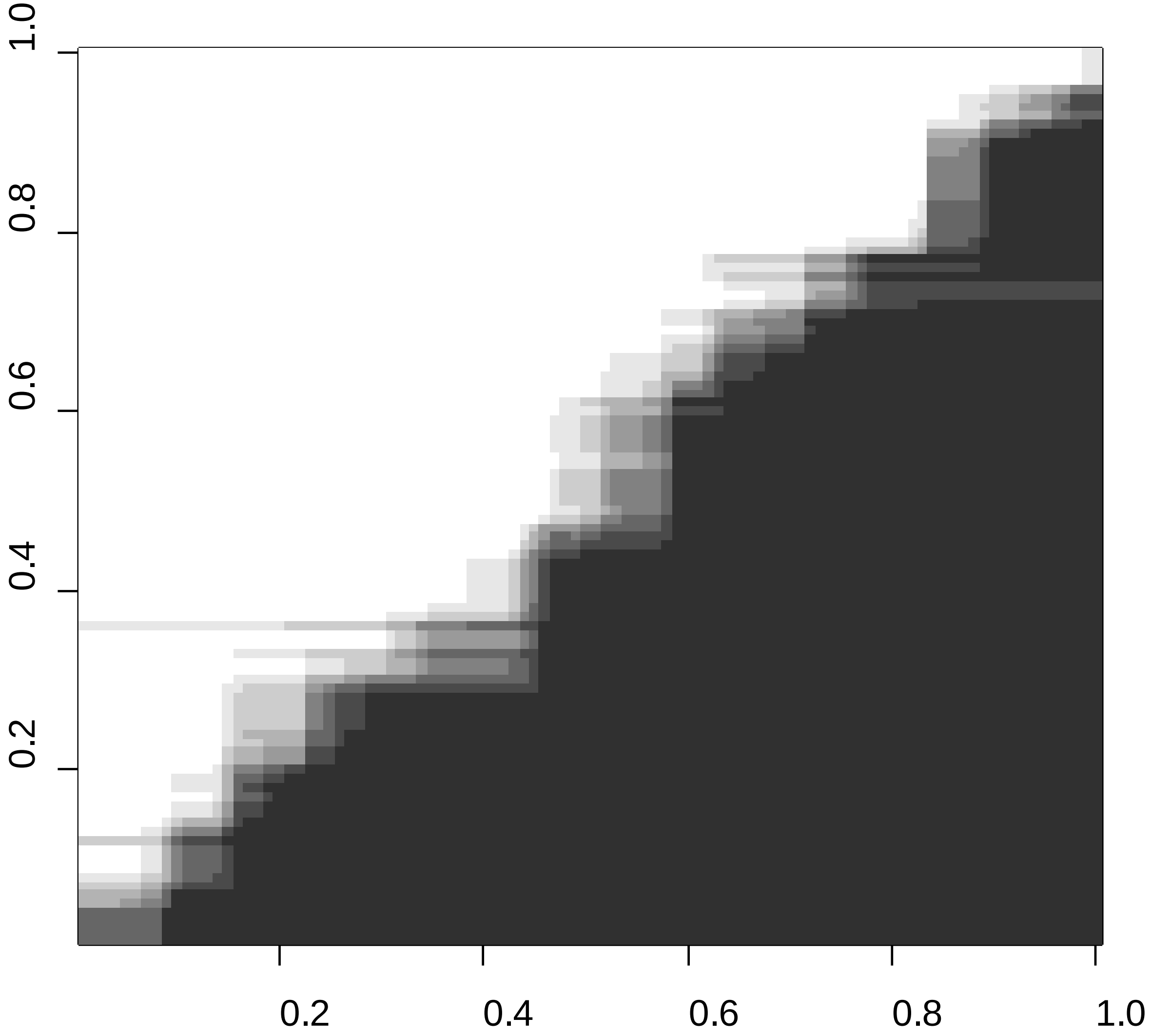
Hard problem for a single tree:



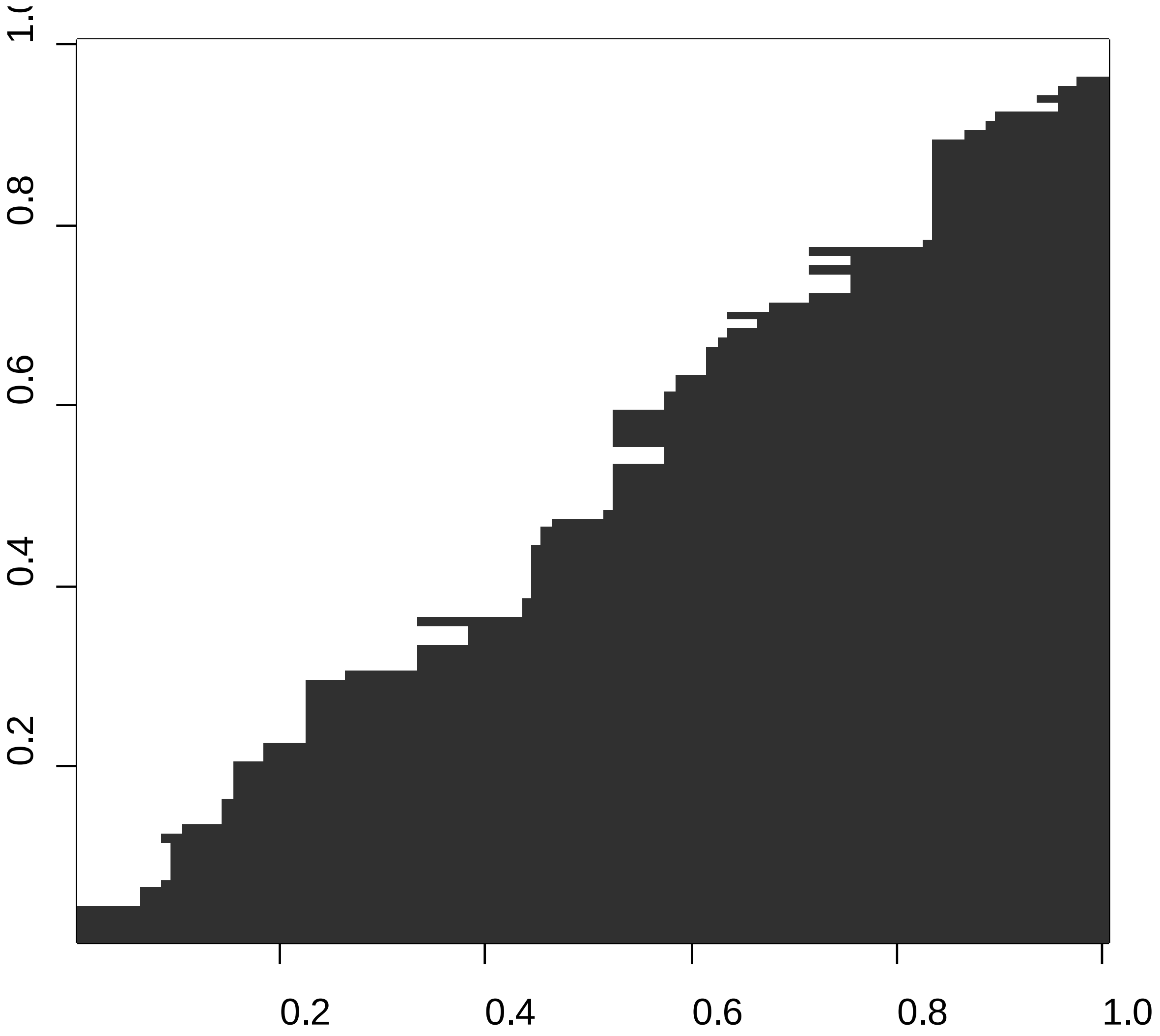
Single tree:



25 Averaged Trees:



25 Voted Trees:



Bagging (Bootstrap Aggregating)

Breiman, “Bagging Predictors”, *Machine Learning*, 1996.

Fit classification or regression models to bootstrap samples from the data and combine by voting (classification) or averaging (regression).

MODEL AVERAGING

Bootstrap sample ⇛ $f_1(x)$

Bootstrap sample ⇛ $f_2(x)$

Bootstrap sample ⇛ $f_3(x)$

…

Bootstrap sample ⇛ $f_M(x)$

Combine $f_1(x),\ldots, f_M(x)$ ⇛ $f(x)$

$f_i(x)$’s are “base learners”

Bagging (Bootstrap Aggregating)

- A bootstrap sample is chosen at random *with* replacement from the data. Some observations end up in the bootstrap sample more than once, while others are not included (“out of bag”).
- Bagging reduces the *variance* of the base learner but has limited effect on the *bias*.
- It’s most effective if we use *strong* base learners that have very little bias but high variance (unstable). E.g. trees.
- Both bagging and boosting are examples of “ensemble learners” that were popular in machine learning in the ‘90s.

Bagging CART

Dataset	# cases	# vars	# classes	CART	Bagged CART	Decrease %
Waveform	300	21	3	29.1	19.3	34
Heart	1395	16	2	4.9	2.8	43
Breast Cancer	699	9	2	5.9	3.7	37
Ionosphere	351	34	2	11.2	7.9	29
Diabetes	768	8	2	25.3	23.9	6
Glass	214	9	6	30.4	23.6	22
Soybean	683	35	19	8.6	6.8	21

Leo Breiman (1996) “Bagging Predictors”, Machine Learning, 24, 123-140.

Outline

- Background.
- Trees.
- Bagging predictors.
- Random Forests algorithm.
- Variable importance.
- Proximity measures.
- Visualization.
- Partial plots and interpretation of effects.

Random Forests

Dataset	# cases	# vars	# classes	CART	Bagged CART	Random Forests
Waveform	300	21	3	29.1	19.3	17.2
Breast Cancer	699	9	2	5.9	3.7	2.9
Ionosphere	351	34	2	11.2	7.9	7.1
Diabetes	768	8	2	25.3	23.9	24.2
Glass	214	9	6	30.4	23.6	20.6

Leo Breiman (2001) “Random Forests”, Machine Learning, 45, 5-32.

Random Forests

Grow a **forest** of many trees. (R default is 500)

Grow each tree on an independent **bootstrap sample*** from the training data.

At each node:

1. Select ***m*** variables **at random** out of all ***M*** possible variables (independently for each node).
2. Find the best split on the selected ***m*** variables.

Grow the trees to maximum depth (classification).

Vote/average the trees to get predictions for new data.

*Sample N cases at random with replacement.

Random Forests

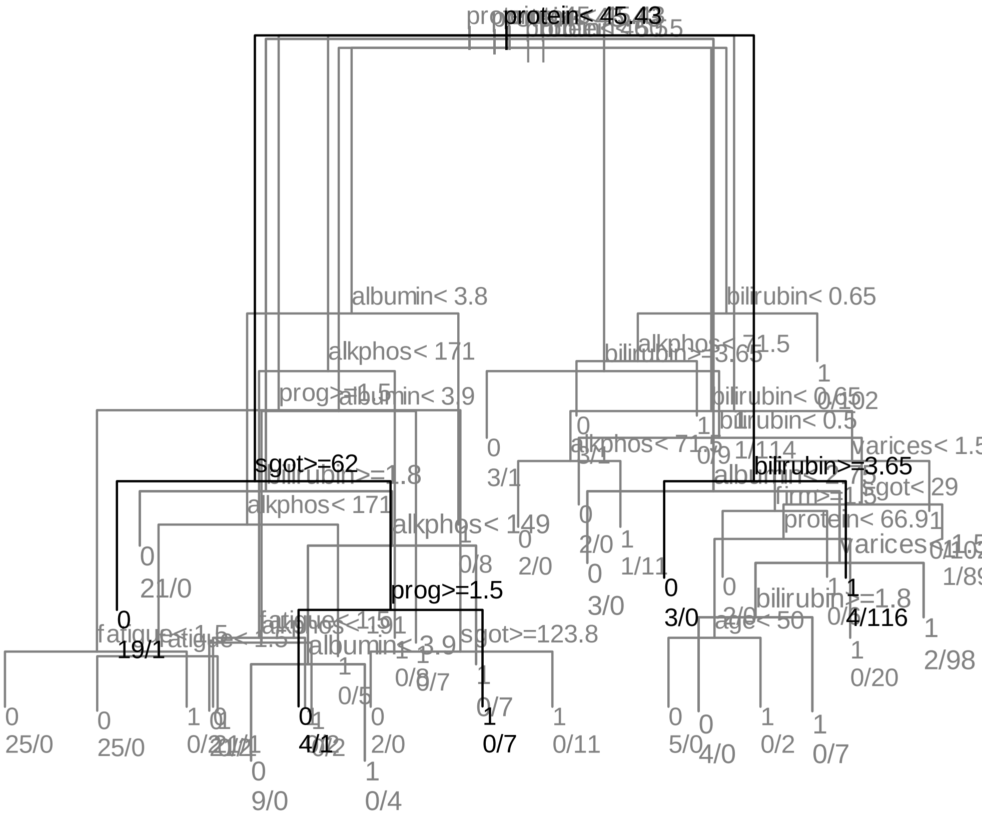
Inherit many of the advantages of CART:

- Applicable to both regression and classification problems. Yes.
- Handle categorical predictors naturally. Yes.
- Computationally simple and quick to fit, even for large problems. Yes.
- No formal distributional assumptions (non-parametric). Yes.
- Can handle highly non-linear interactions and classification boundaries. Yes.
- Automatic variable selection. Yes. But need variable importance too.
- Handles missing values ~~through surrogate variables.~~ Using proximities.
- ~~Very easy to interpret if the tree is small~~. NO!

Random Forests

But do not inherit:

- ~~The picture of the tree can give valuable insights into which variables are important and where.~~ NO!
- ~~The terminal nodes suggest a natural clustering of data into homogeneous groups.~~ NO!



Random Forests

Improve on CART with respect to:

- *Accuracy* – Random Forests is competitive with the best known machine learning methods (but note the “no free lunch” theorem).

- *Instability* – if we change the data a little, the individual trees may change but the forest is relatively stable because it is a combination of many trees.

Two Natural Questions

1. ***Why bootstrap? (Why subsample?)***

   Bootstrapping → out-of-bag data →

   – Estimated error rate and confusion matrix
   – Variable importance

2. ***Why trees?***

   Trees → proximities →

   – Missing value fill-in
   – Outlier detection
   – Illuminating pictures of the data (clusters, structure, outliers)

The RF Predictor

- A case in the training data is *not* in the bootstrap sample for about one third of the trees (we say the case is "out of bag" or "oob"). Vote (or average) the predictions of *these trees* to give ***the RF predictor.***

- The ***oob error rate*** is the error rate of the ***RF predictor.***

- The ***oob confusion matrix*** is obtained from the ***RF predictor.***

- For new cases, vote (or average) *all* the trees to get the ***RF predictor.***

The RF Predictor

For example, suppose we fit 1000 trees, and a case is out-of-bag in 339 of them, of which:

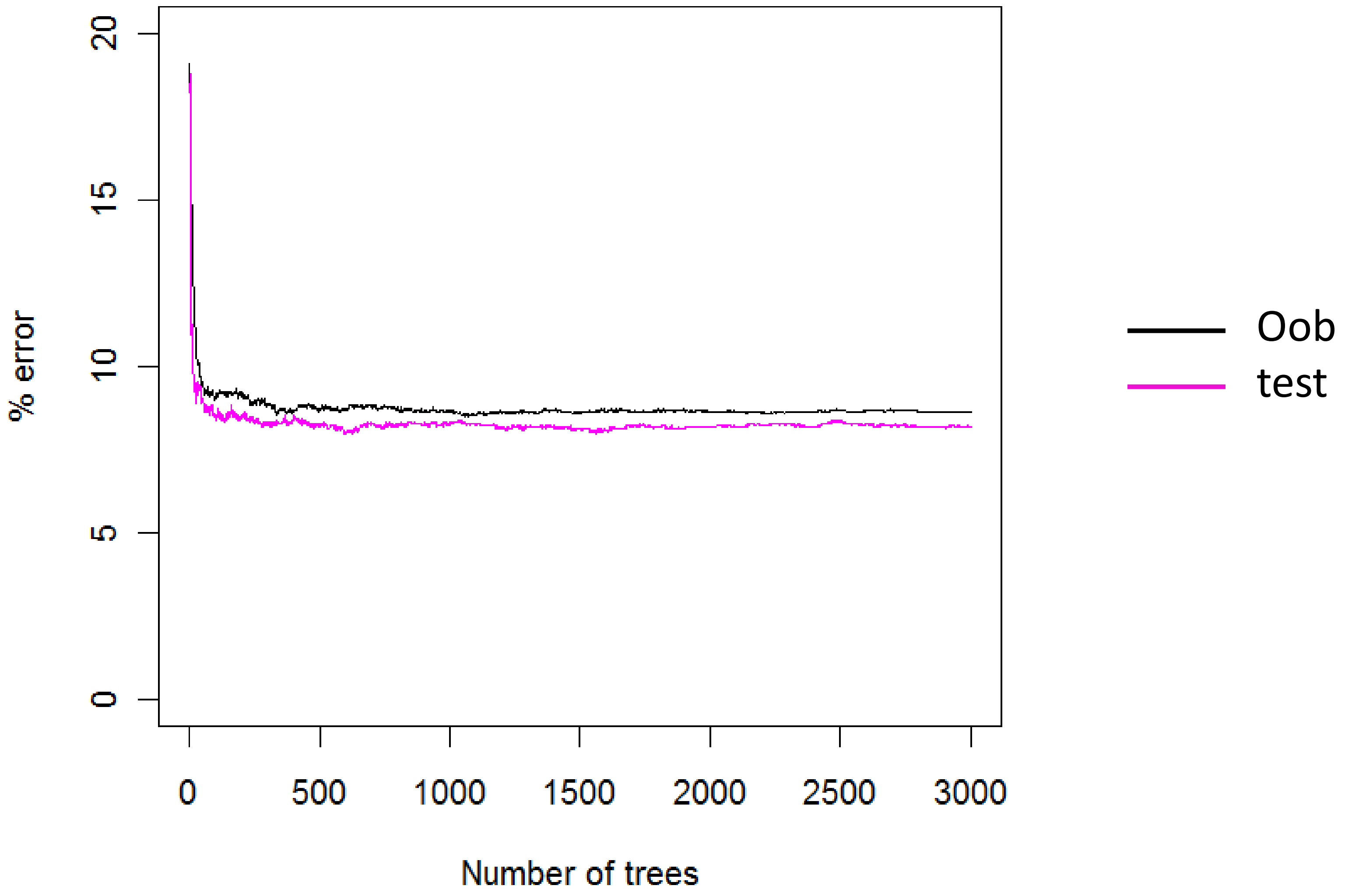
283 say “class 1”

56 say “class 2”

The RF predictor for this case is class 1.

The “oob” error gives an estimate of test set error (generalization error) as trees are added to the ensemble.

RFs do not overfit as we fit more trees



RF handles thousands of predictors

Ramón Díaz-Uriarte, Sara Alvarez de Andrés
Bioinformatics Unit, Spanish National Cancer Center
March, 2005 http://ligarto.org/rdiaz

Compared

- SVM, linear kernel
- KNN/crossvalidation (Dudoit et al. JASA 2002)
- DLDA
- Shrunken Centroids (Tibshirani et al. PNAS 2002)
- Random forests

"Given its performance, random forest and variable selection using random forest should probably become part of the standard tool-box of methods for the analysis of microarray data."

Microarray Datasets

Data	*P*	*N*	*# Classes*
Leukemia	3051	38	2
Breast 2	4869	78	2
Breast 3	4869	96	3
NCI60	5244	61	8
Adenocar	9868	76	2
Brain	5597	42	5
Colon	2000	62	2
Lymphoma	4026	62	3
Prostate	6033	102	2
Srbct	2308	63	4

Microarray Error Rates

Data	*SVM*	*KNN*	*DLDA*	*SC*	*RF*	rank
Leukemia	.014	.029	.020	.025	.051	**5**
Breast 2	.325	.337	.331	.324	.342	**5**
Breast 3	.380	.449	.370	.396	.351	**1**
NCI60	.256	.317	.286	.256	.252	**1**
Adenocar	.203	.174	.194	.177	.125	**1**
Brain	.138	.174	.183	.163	.154	**2**
Colon	.147	.152	.137	.123	.127	**2**
Lymphoma	.010	.008	.021	.028	.009	**2**
Prostate	.064	.100	.149	.088	.077	**2**
Srbct	.017	.023	.011	.012	.021	**4**
Mean	**.155**	**.176**	**.170**	**.159**	**.151**	

RF handles thousands of predictors

- Add noise to some standard datasets and see how well Random Forests:
  - predicts
  - detects the important variables

RF error rates (%)

	No noise added	***10 noise variables***		***100 noise variables***	
Dataset	***Error rate***	***Error rate***	***Ratio***	***Error rate***	***Ratio***
breast	3.1	2.9	**0.93**	2.8	**0.91**
diabetes	23.5	23.8	**1.01**	25.8	**1.10**
ecoli	11.8	13.5	**1.14**	21.2	**1.80**
german	23.5	25.3	**1.07**	28.8	**1.22**
glass	20.4	25.9	**1.27**	37.0	**1.81**
image	1.9	2.1	**1.14**	4.1	**2.22**
iono	6.6	6.5	**0.99**	7.1	**1.07**
liver	25.7	31.0	**1.21**	40.8	**1.59**
sonar	15.2	17.1	**1.12**	21.3	**1.40**
soy	5.3	5.5	**1.06**	7.0	**1.33**
vehicle	25.5	25.0	**0.98**	28.7	**1.12**
votes	4.1	4.6	**1.12**	5.4	**1.33**
vowel	2.6	4.2	**1.59**	17.9	**6.77**

RF error rates

Error rates (%)		*Number of noise variables*			
Dataset	*No noise added*	*10*	*100*	*1,000*	*10,000*
breast	3.1	2.9	2.8	3.6	8.9
glass	20.4	25.9	37.0	51.4	61.7
votes	4.1	4.6	5.4	7.8	17.7

Outline

- Background.
- Trees.
- Bagging predictors.
- Random Forests algorithm.
- Variable importance.
- Proximity measures.
- Visualization.
- Partial plots and interpretation of effects.

Variable Importance

RF computes two measures of variable importance, one based on a rough-and-ready measure (Gini for classification) and the other based on permutations.

To understand how permutation importance is computed, need to understand local variable importance. But first…

RF variable importance

		10 noise variables		***100 noise variables***	
Dataset	***m***	***Number in top m***	***Percent***	***Number in top m***	***Percent***
breast	9	9.0	**100.0**	9.0	**100.0**
diabetes	8	7.6	**95.0**	7.3	**91.2**
ecoli	7	6.0	**85.7**	6.0	**85.7**
german	24	20.0	**83.3**	10.1	**42.1**
glass	9	8.7	**96.7**	8.1	**90.0**
image	19	18.0	**94.7**	18.0	**94.7**
ionosphere	34	33.0	**97.1**	33.0	**97.1**
liver	6	5.6	**93.3**	3.1	**51.7**
sonar	60	57.5	**95.8**	48.0	**80.0**
soy	35	35.0	**100.0**	35.0	**100.0**
vehicle	18	18.0	**100.0**	18.0	**100.0**
votes	16	14.3	**89.4**	13.7	**85.6**
vowel	10	10.0	**100.0**	10.0	**100.0**

RF error rates

Number in top m		*Number of noise variables*			
Dataset	*m*	*10*	*100*	*1,000*	*10,000*
breast	9	9.0	9.0	9	9
glass	9	8.7	8.1	7	6
votes	16	14.3	13.7	13	13

Local Variable Importance

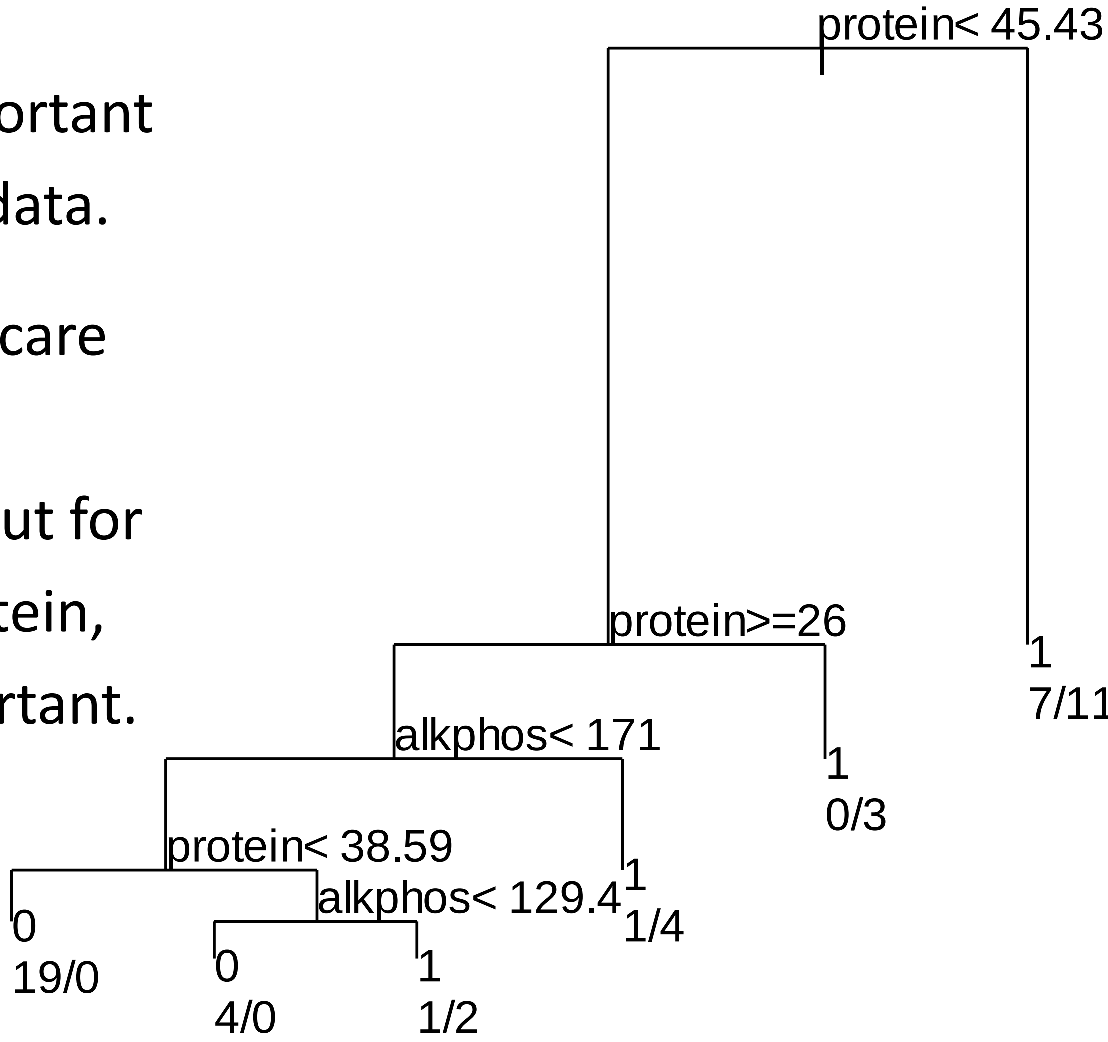
We usually think about variable importance as an overall measure. In part, this is probably because we fit models with global structure (linear regression, logistic regression).

In CART, variable importance is local.

Local Variable Importance

Different variables are important in different regions of the data.

If protein is high, we don’t care about alkaline phosphate. Similarly if protein is low. But for intermediate values of protein, alkaline phosphate is important.



Local Variable Importance

For each tree, look at the out-of-bag data:

- randomly permute the values of variable *j*
- pass these perturbed data down the tree, save the classes.

For case *i* and variable *j* find

$$\left\{\begin{array}{l}\text{error rate with}\\ \text{variable } j \text{ permuted}\end{array}\right\} - \left\{\begin{array}{l}\text{error rate with}\\ \text{no permutation}\end{array}\right\}$$

where the error rates are taken over all trees for which case *i* is out-of-bag.

Local importance for a class 2 case

TREE	No permutation	Permute variable 1	…	Permute variable m
1	2	2	…	1
3	2	2	…	2
4	1	1	…	1
9	2	2	…	1
…	…	…	…	…
992	2	2	…	2
% Error	10%	11%	…	35%

Outline

- Background.
- Trees.
- Bagging predictors.
- Random Forests algorithm.
- Variable importance.
- Proximity measures.
- Visualization.
- Partial plots and interpretation of effects.

Proximities

Proximity of two cases is the proportion of the time that they end up in the same node.

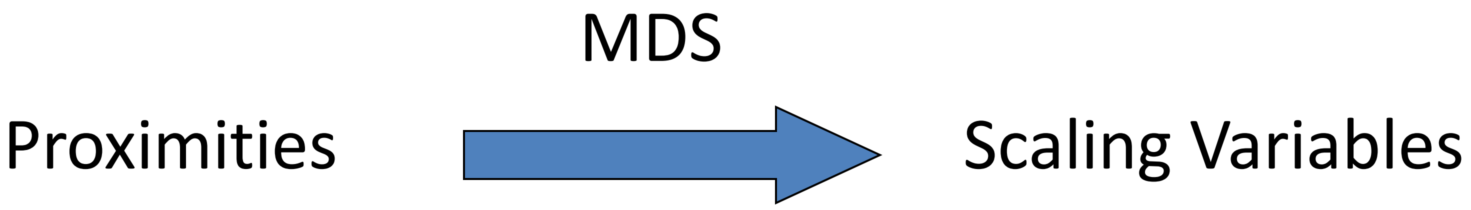
The proximities don’t just measure similarity of the variables - they also take into account the importance of the variables.

Two cases that have quite **different** predictor variables might have **large** proximity if they differ only on variables that are **not important**.

Two cases that have quite **similar** values of the predictor variables might have **small** proximity if they differ on inputs that are **important.**

Visualizing using Proximities

To “look” at the data we use classical multidimensional scaling (MDS) to get a picture in 2-D or 3-D:



Proximities $\xrightarrow{\text{MDS}}$ Scaling Variables

Might see clusters, outliers, unusual structure.

Can also use nonmetric MDS.

Visualizing using Proximities

- at-a-glance information about which classes are overlapping, which classes differ
- find clusters within classes
- find easy/hard/unusual cases

With a good tool we can also

- identify characteristics of unusual points
- see which variables are locally important
- see how clusters or unusual points differ

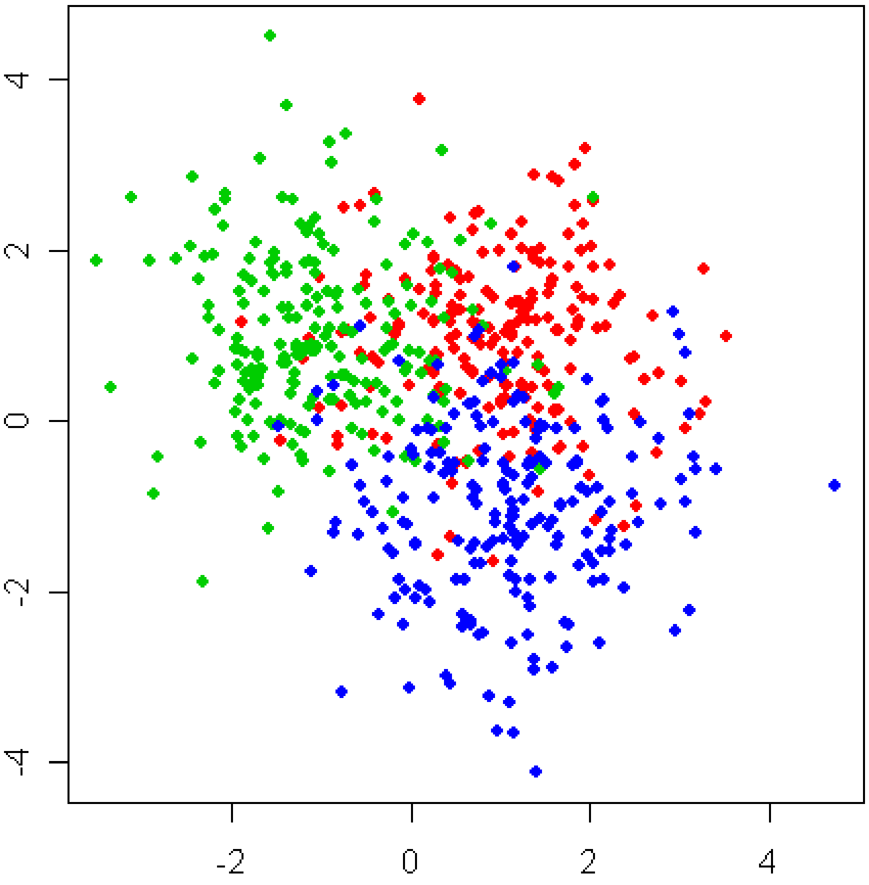
Visualizing using Proximities

Synthetic data, 600 cases

2 meaningful variables

48 “noise” variables

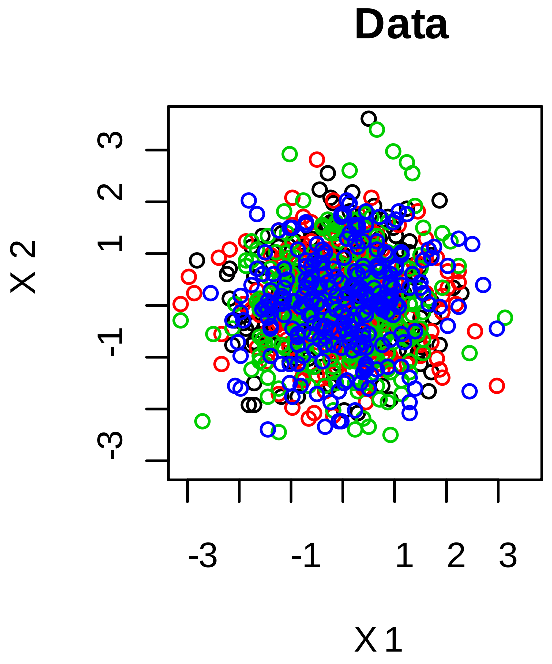
3 classes

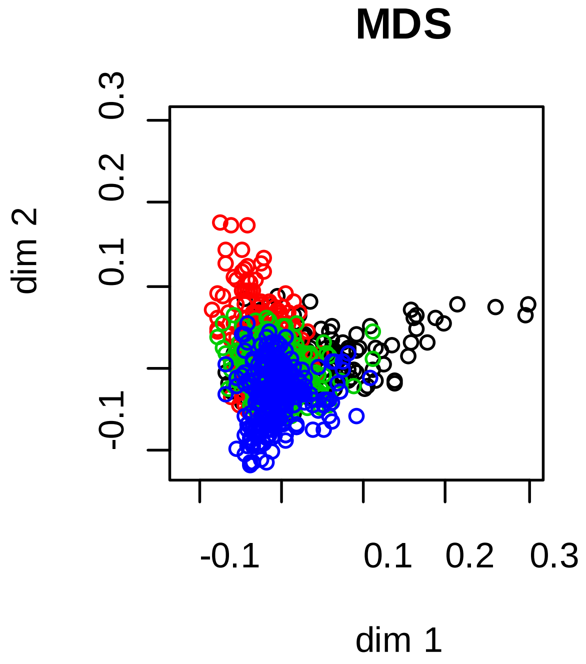


The Problem with Proximities

Proximities based on *all* the data overfit!

e.g. two cases from different classes must have proximity zero if trees are grown deep.





Proximity-weighted Nearest Neighbors

RF is like a nearest-neighbor classifier:

- Use the proximities as weights for nearest-neighbors.
- Classify the training data.
- Compute the error rate.

Want the error rate to be close to the RF oob error rate.

BAD NEWS! If we compute proximities from trees in which both cases are OOB, we don’t get good accuracy when we use the proximities for prediction!

Proximity-weighted Nearest Neighbors

Dataset	*RF*	*OOB*
breast	2.6	2.9
diabetes	24.2	23.7
ecoli	11.6	12.5
german	23.6	24.1
glass	20.6	**23.8**
image	1.9	2.1
iono	6.8	6.8
liver	26.4	26.7
sonar	13.9	**21.6**
soy	5.1	5.4
vehicle	24.8	**27.4**
votes	3.9	3.7
vowel	2.6	**4.5**

Proximity-weighted Nearest Neighbors

Dataset	*RF*	*OOB*
Waveform	15.5	16.1
Twonorm	3.7	4.6
Threenorm	14.5	15.7
Ringnorm	5.6	5.9

New Proximity Method

Start with P = I, the identity matrix.

For each observation *i*:

For each tree in which case *i* is oob:

- Pass case i down the tree and note which terminal node it falls into.
- Increase the proximity between observation *i* and the *k* in-bag cases that are in the same terminal node, by the amount *1/k*.

Can show that except for ties, this gives the same error rate as RF, when used as a proximity-weighted nn classifier.

New Method

Dataset	*RF*	*OOB*	*New*
breast	2.6	2.9	2.6
diabetes	24.2	23.7	24.4
ecoli	11.6	12.5	11.9
german	23.6	24.1	23.4
glass	20.6	**23.8**	20.6
image	1.9	2.1	1.9
iono	6.8	6.8	6.8
liver	26.4	26.7	26.4
sonar	13.9	**21.6**	13.9
soy	5.1	5.4	5.3
vehicle	24.8	**27.4**	24.8
votes	3.9	3.7	3.7
vowel	2.6	**4.5**	2.6

New Method

Dataset	*RF*	*OOB*	*New*
Waveform	15.5	16.1	15.5
Twonorm	3.7	4.6	3.7
Threenorm	14.5	15.7	14.5
Ringnorm	5.6	5.9	5.6

But…

The new “proximity” matrix is not symmetric!

→ Methods for doing multidimensional scaling on asymmetric matrices.

Other Uses for Random Forests

- Missing data imputation.
- Feature selection (before using a method that cannot handle high dimensionality).
- Unsupervised learning (cluster analysis).
- Survival analysis without making the proportional hazards assumption.

Missing Data Imputation

Fast way: replace missing values for a given variable using the median of the non-missing values (or the most frequent, if categorical)

Better way (using proximities):

1. Start with the fast way.
2. Get proximities.
3. Replace missing values in case **i** by a weighted average of non-missing values, with weights proportional to the proximity between case **i** and the cases with the non-missing values.

Repeat steps 2 and 3 a few times (5 or 6).

Feature Selection

- Ramón Díaz-Uriarte:

  varSelRF R package.

- In the NIPS competition 2003, several of the top entries used RF for feature selection.

Unsupervised Learning

Global histone modification patterns predict risk of prostate cancer recurrence

David B. Seligson, Steve Horvath, Tao Shi, Hong Yu, Sheila Tze, Michael Grunstein and Siavash K. Kurdistan (all at UCLA).

Used RF clustering of 183 tissue microarrays to find two disease subgroups with distinct risks of tumor recurrence.

http://www.nature.com/nature/journal/v435/n7046/full/nature03672.html

Survival Analysis

- Hemant Ishwaran and Udaya B. Kogalur: randomSurvivalForest R package.

Outline

- Background.
- Trees.
- Bagging predictors.
- Random Forests algorithm.
- Variable importance.
- Proximity measures.
- Visualization.

Case Study: Cavity Nesting birds in the Uintah Mountains, Utah

- Red-naped sapsucker (*Sphyrapicus nuchalis*) (*n* = 42 nest sites)

© Jack Murray/CLO

- Mountain chickadee (*Parus gambeli*) (*n* = 42 nest sites)

- Northern flicker (*Colaptes auratus*) (*n* = 23 nest sites)

- *n* = 106 non-nest sites

Case Study: Cavity Nesting birds in the Uintah Mountains, Utah

- Response variable is the presence (coded 1) or absence (coded 0) of a nest.
- Predictor variables (measured on 0.04 ha plots around the sites) are:
  - Numbers of trees in various size classes from less than 1 inch in diameter at breast height to greater than 15 inches in diameter.
  - Number of snags and number of downed snags.
  - Percent shrub cover.
  - Number of conifers.
  - Stand Type, coded as 0 for pure aspen and 1 for mixed aspen and conifer.

Autism

Data courtesy of J.D.Odell and R. Torres, USU

154 subjects (308 chromosomes)

7 variables, all categorical (up to 30 categories)

2 classes:

- **Normal, blue (69 subjects)**
- **Autistic, red (85 subjects)**

Brain Cancer Microarrays

Pomeroy et al. Nature, 2002.
Dettling and Bühlmann, Genome Biology, 2002.

42 cases, 5,597 genes, 5 tumor types:

- **10 medulloblastomas BLUE**
- **10 malignant gliomas PALE BLUE**
- **10 atypical teratoid/rhabdoid tumors (AT/RTs) GREEN**
- **4 human cerebella ORANGE**
- **8 PNETs RED**

Dementia

Data courtesy of J.T. Tschanz, USU

516 subjects

28 variables

2 classes:

- **no cognitive impairment, blue (372 people)**
- **Alzheimer’s, red (144 people)**

Metabolomics

(Lou Gehrig’s disease)
data courtesy of Metabolon (Chris Beecham)

63 subjects
317 variables
3 classes:

- blue (22 subjects) ALS (no meds)
- green (9 subjects) ALS (on meds)
- red (32 subjects) healthy

Random Forests Software

- Free, open-source code (FORTRAN, java) www.math.usu.edu/~adele/forests
- Commercial version (academic discounts) www.salford-systems.com
- R interface, independent development (Andy Liaw and Matthew Wiener)

Java Software

Raft uses VisAD
www.ssec.wisc.edu/~billh/visad.html

and ImageJ
http://rsb.info.nih.gov/ij/

These are both open-source projects with great mailing lists and helpful developers.

References

- Leo Breiman, Jerome Friedman, Richard Olshen, Charles Stone (1984) “Classification and Regression Trees” (Wadsworth).
- Leo Breiman (1996) “Bagging Predictors” Machine Learning, 24, 123-140.
- Leo Breiman (2001) “Random Forests” Machine Learning, 45, 5-32.
- Trevor Hastie, Rob Tibshirani, Jerome Friedman (2009) “Statistical Learning” (Springer).