Principles of Artificial Intelligence(305450)

Lecture 11:

Machine Learning IV

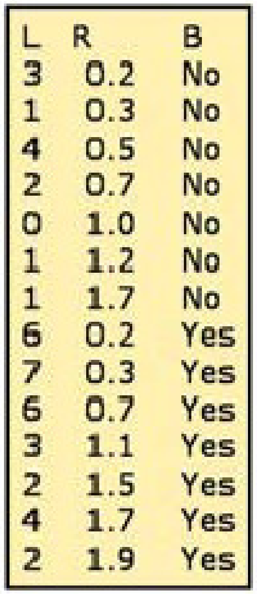
Feature Spaces

- Features can be much more complex
- Drawn from bigger discrete set (e.g., make of a car, age of a person)
  - If set is unordered (e.g., 4 different makes of cars), may use binary attributes to encode the values (00, 10, 10, 11) but preferable unary attributes (1000, 0100, 0010, 0001)
  - If set is ordered (e.g., person’s age), treated as real-valued

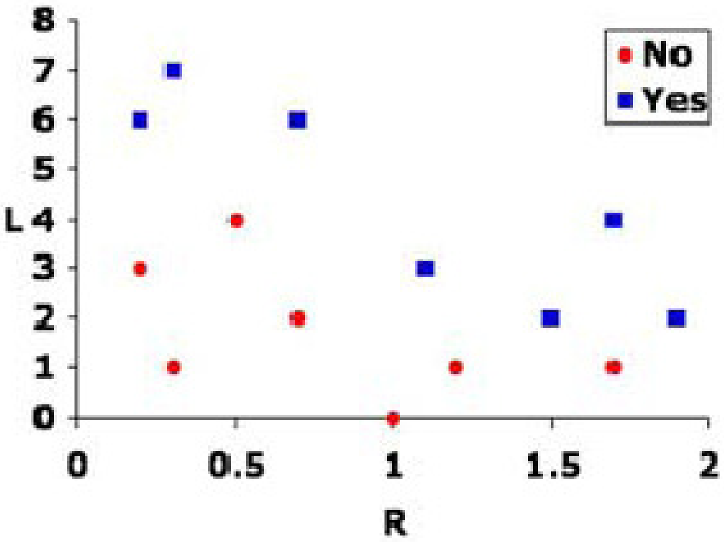
Feature Spaces

- Features can be much more complex
- Drawn from bigger discrete set (e.g., make of a car, age of a person)
  - If set is unordered (e.g., 4 different makes of cars), may use binary attributes to encode the values (00, 10, 10, 11) but preferable unary attributes (1000, 0100, 0010, 0001)
  - If set is ordered (e.g., person's age), treated as real-valued
- Real-valued: bias that inputs whose features have "nearby" values ought to have "nearby" outputs

Predicting Bankruptcy



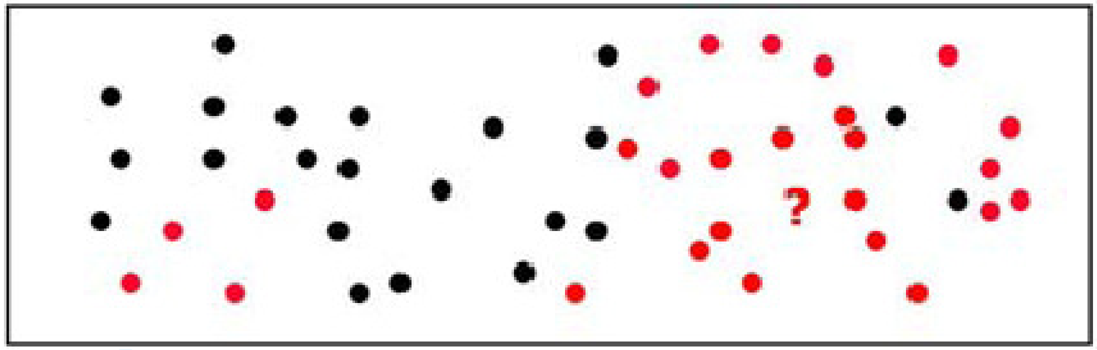
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1.0	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



L: # late payments / year
R: expenses / income

Nearest Neighbor

- Remember all your data
- When someone asks a question,
  - Find the nearest old data point
  - Return the answer associated with it



What do we mean by “nearest”?

- Need a distance function on inputs
- Typically use Euclidean distance (length of a straight line between the points)

$$D(x^i, x^k) = \sqrt{\sum_j (x_j^i - x_j^k)^2}$$

- Distance between character strings might be number of edits required to turn one into the other

Scaling

- What if we’re trying to predict a car’s gas mileage?
  - f_1 = weight in pounds
  - f_2 =number of cylinders
- Any effect of f_2 will be completely lost because of the relative scales
- So re-scale the inputs

Scaling

- What if we’re trying to predict a car’s gas mileage?
  - f_1 = weight in pounds
  - f_2 =number of cylinders
- Any effect of f_2 will be completely lost because of the relative scales
- So re-scale the inputs to have mean 0 and variance 1:

$$x' = \frac{x - \underbrace{\bar{x}}_{\boxed{\text{average}}}}{\underbrace{\sigma_x}_{\boxed{\text{standard deviation}}}}$$

Scaling

- What if we're trying to predict a car's gas mileage?
  - f_1 = weight in pounds
  - f_2 =number of cylinders
- Any effect of f_2 will be completely lost because of the relative scales
- So re-scale the inputs to have mean 0 and variance 1:

$$x' = \frac{x - \bar{x}}{\sigma_x}$$

average

standard deviation

- Or, build knowledge in by scaling features differently

Scaling

- What if we're trying to predict a car's gas mileage?
  - f_1 = weight in pounds
  - f_2 =number of cylinders
- Any effect of f_2 will be completely lost because of the relative scales
- So re-scale the inputs to have mean 0 and variance 1:

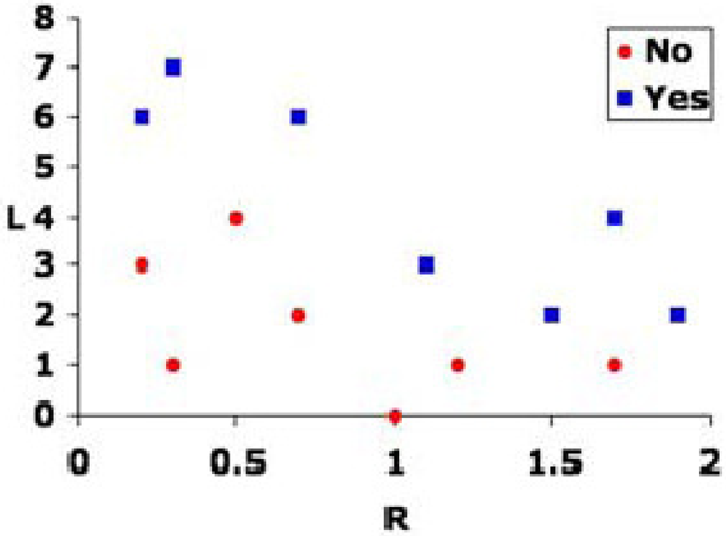
$$x' = \frac{x - \bar{x}}{\sigma_x}$$

average

standard deviation

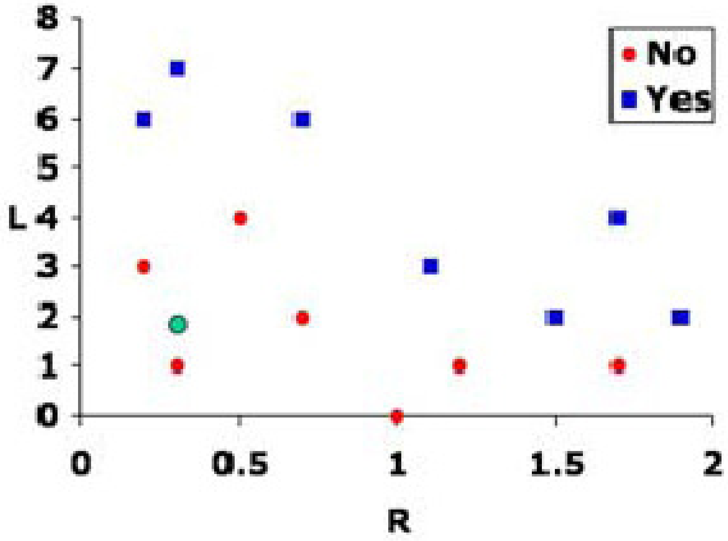
- Or, build knowledge in by scaling features differently
- Or, use cross-validation to choose scales

Predicting Bankruptcy



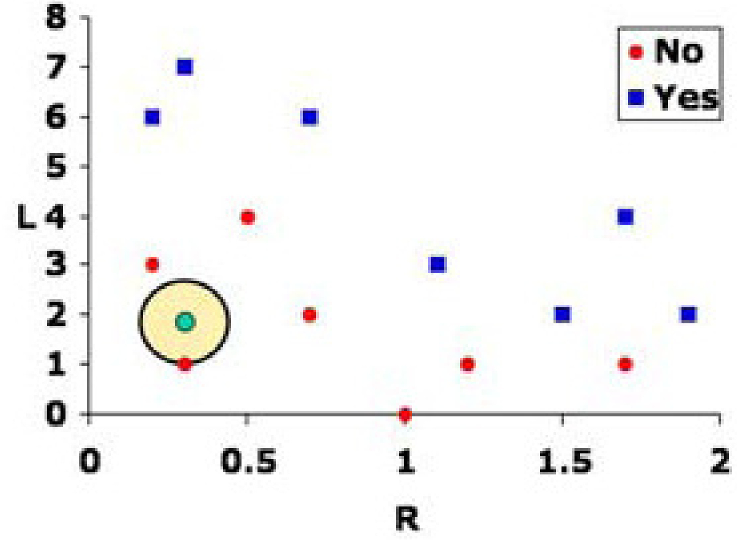
$$D(x^i, x^k) = \sqrt{\sum_j (L^i - L^k)^2 + (5R^i - 5R^k)^2}$$

Predicting Bankruptcy



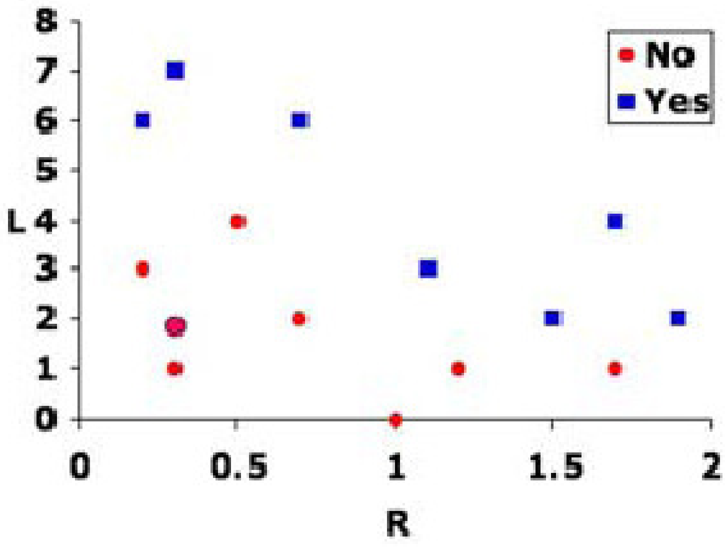
$$D(x^i, x^k) = \sqrt{\sum_j (L^i - L^k)^2 + (5R^i - 5R^k)^2}$$

Predicting Bankruptcy



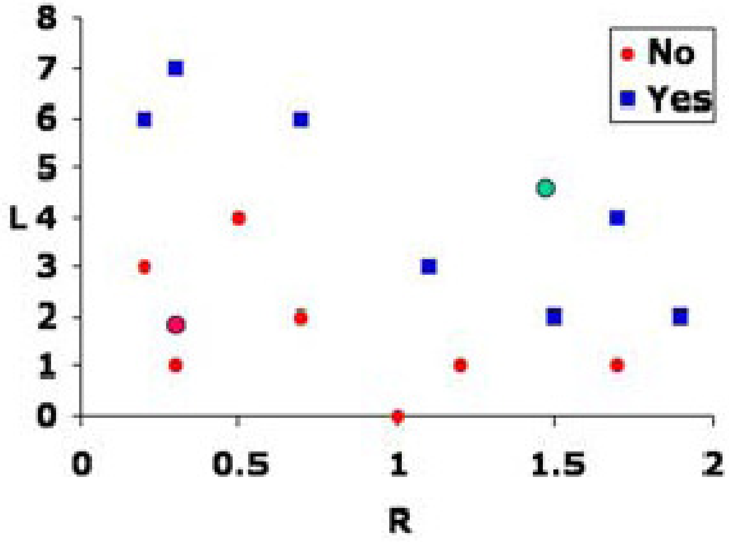
$$D(x^i, x^k) = \sqrt{\sum_j (L^i - L^k)^2 + (5R^i - 5R^k)^2}$$

Predicting Bankruptcy



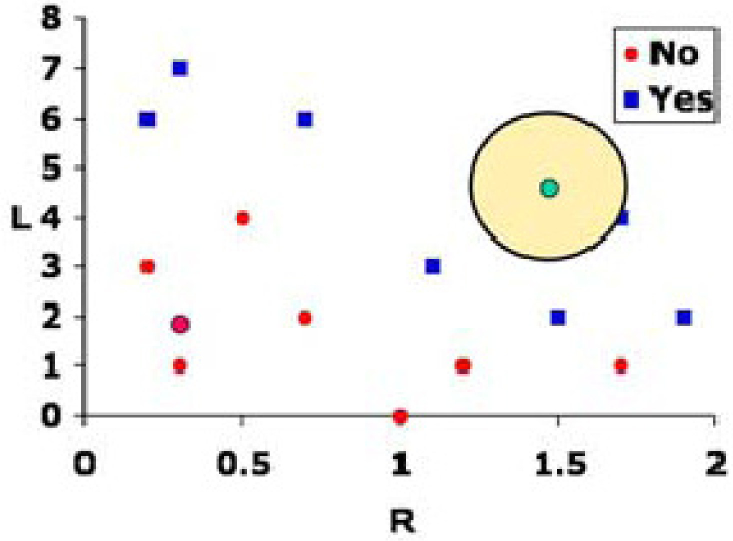
$$D(x^i, x^k) = \sqrt{\sum_j (L^i - L^k)^2 + (5R^i - 5R^k)^2}$$

Predicting Bankruptcy



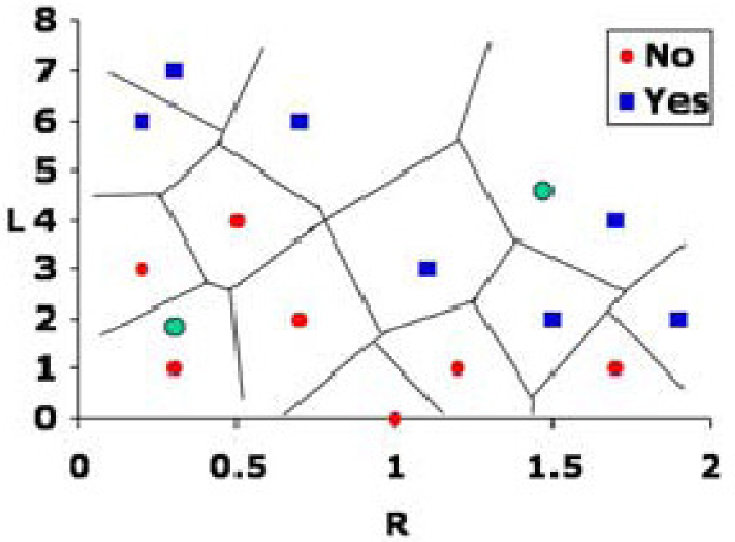
$$D(x^i, x^k) = \sqrt{\sum_j (L^i - L^k)^2 + (5R^i - 5R^k)^2}$$

Predicting Bankruptcy



$$D(x^i, x^k) = \sqrt{\sum_j (L^i - L^k)^2 + (5R^i - 5R^k)^2}$$

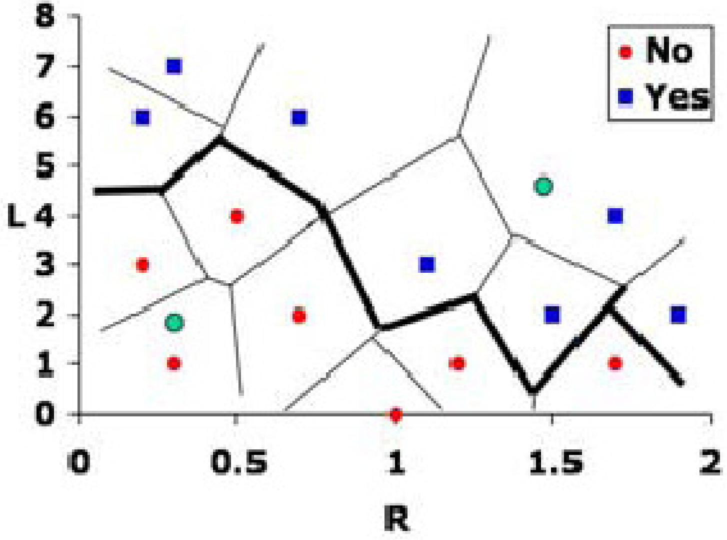
Hypothesis



$$D(x^i, x^k) = \sqrt{\sum_j (L^i - L^k)^2 + (5R^i - 5R^k)^2}$$

- Hypothesis in this algorithm is different from others, it isn’t explicitly Constructing a description of a hypothesis based on the data it sees.
- Given a set of points and a distance metric, you can divide the space up into regions, one for each point, which represent the set of points in space that are nearer to this designated point than to any of the others.
- This figure shows (somewhat inaccurate) picture of the decomposition of the space into such regions. It's called a "Voronoi partition" of the space.

Hypothesis



$$D(x^i, x^k) = \sqrt{\sum_j (L^i - L^k)^2 + (5R^i - 5R^k)^2}$$

- Now, we can think of our hypothesis as being represented by the edges in the Voronoi partition that separate a region associated with a positive point from a region associated with a negative one.
- It's important to note that we never explicitly compute this boundary; it just arises out of the "nearest neighbor" query process.

Time and Space

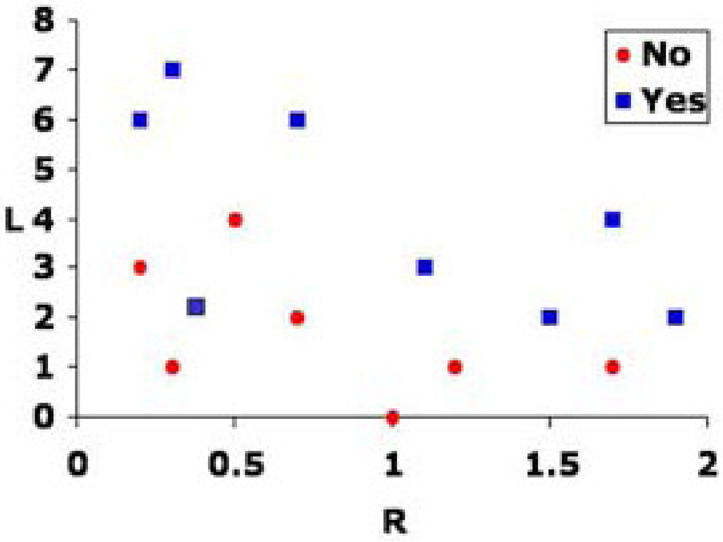
- Learning is fast
- Lookup takes about m*n computations

  where m: number of points in the training set

  n: number of features for each point

  - Storing data in a clever data structure (called KD-tree) reduces this, on average, to log(m)*n
- Memory can fill up with all that data
  - delete points that are far away from the boudary

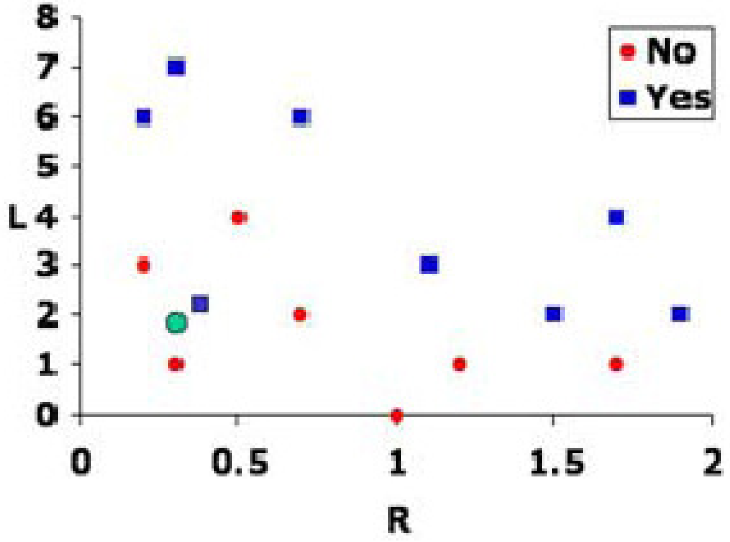
Noise



- In our example so far, there has not been much (apparent) noise; the boundary between positives and negatives is clean and simple.
- Let's consider the case where there's a blue point down among the reds. Someone with an healthy financial record goes bankrupt.
- There are two ways to deal with this data point.

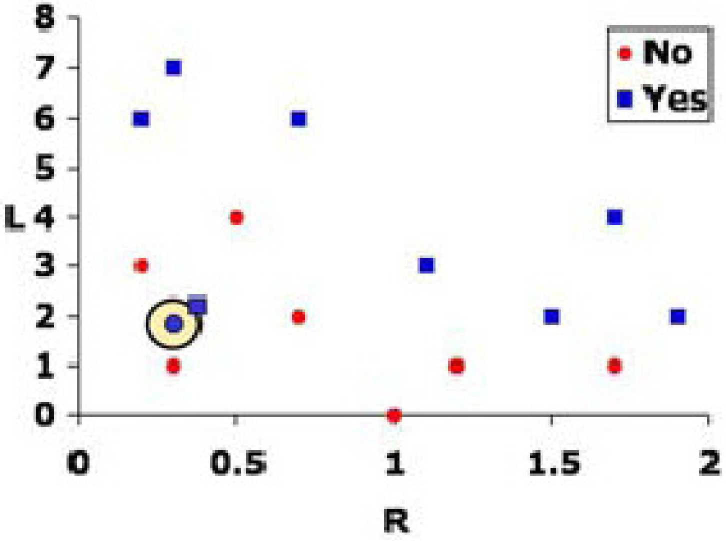
- One is to assume that it is not noise; i.e., there is some regularity that makes people like this one go bankrupt in general.
- The other is to say that this example is an "outlier". It represents an unusual case that we would prefer largely to ignore, and not to incorporate it into our hypothesis.

Noise



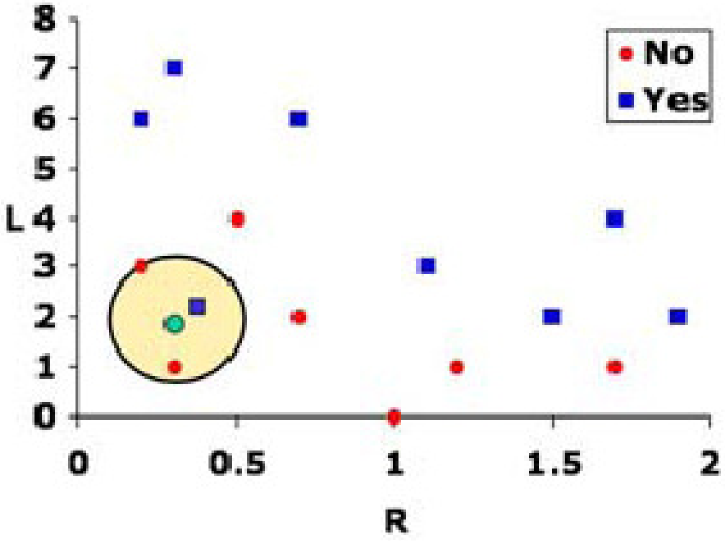
So, what happens in nearest neighbor if we get a query point next to this point?

Noise



- We find the nearest neighbor, which is a "yes" point, and predict the answer "yes".
- This outcome is consistent with the first view; that is, that this blue point represents some important property of the problem.

K-Nearest Neighbor



- Find the k nearest point
- Predict output according to the majority
- Choose k using cross-validation

Curse of Dimensionality

- Nearest neighbor is great in low dimensions (up to about 6)
- As n increases, things get weird:
  - In high dimensions, almost all points are far away from one another
  - They're almost all near the boundaries

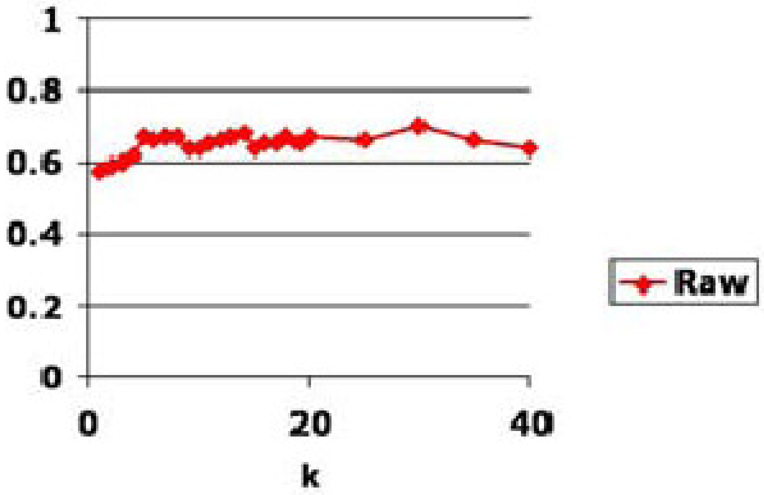
- All this means that the notions of nearness providing a good generalization principle, which are very effective in low-dimensional spaces, become fairly ineffective in high-dimensional spaces.
- There are two ways to handle this problem. One is to do "feature selection", and try to reduce the problem back down to a lower-dimensional one. The other is to fit hypotheses from a much smaller hypothesis class, such as linear separators, which we will see in the next Lecture.

Test Domains

- Heart Disease: predict whether a person has significant narrowing of the arteries, based on tests
  - 26 features (lots are booleans)
  - 297 points
- Auto MPG: predict whether a car gets more than 22 miles per gallon, based on attributes of car
  - 12 features (binary)
  - 285 points

Heart Disease

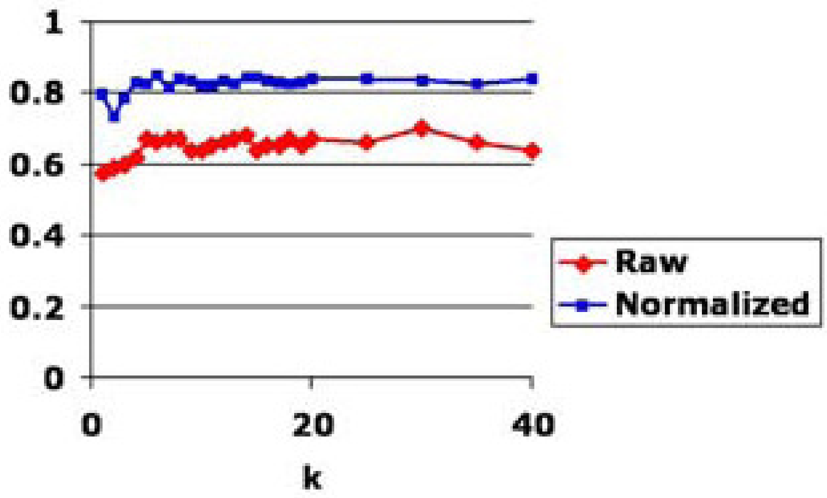
- Relatively insensitive to k



Here's a graph of the cross-validation accuracy of nearest neighbor on the heart disease data, shown as a function of k. Looking at the data, we can see that the performance is relatively insensitive to the choice of k, though it seems like maybe it's useful to have k be greater than about 5

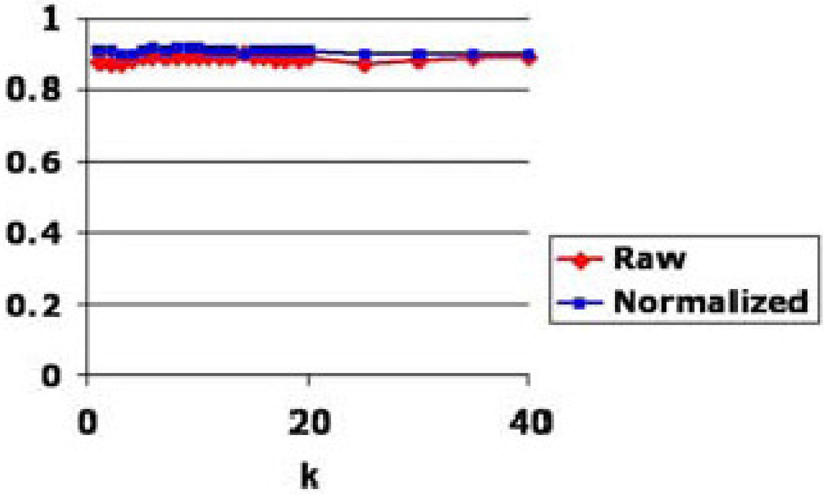
Heart Disease

- Relatively insensitive to k
- Normalization matters!



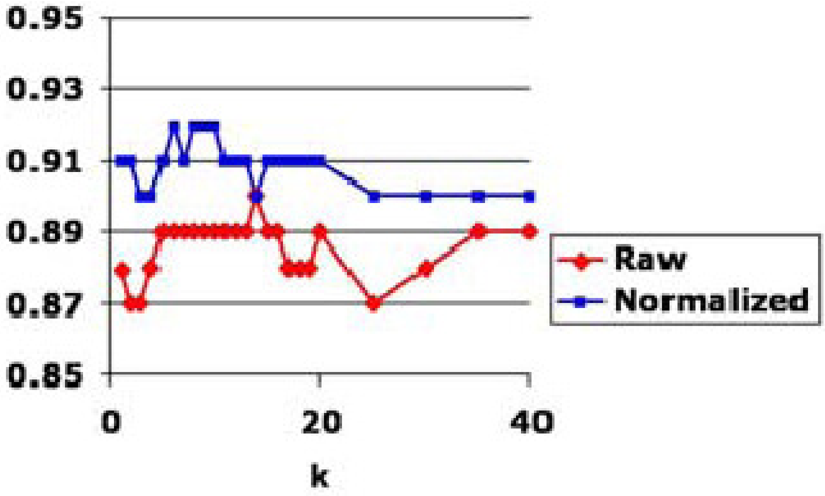
Auto MPG

- Relatively insensitive to k
- Normalization doesn’t matter much



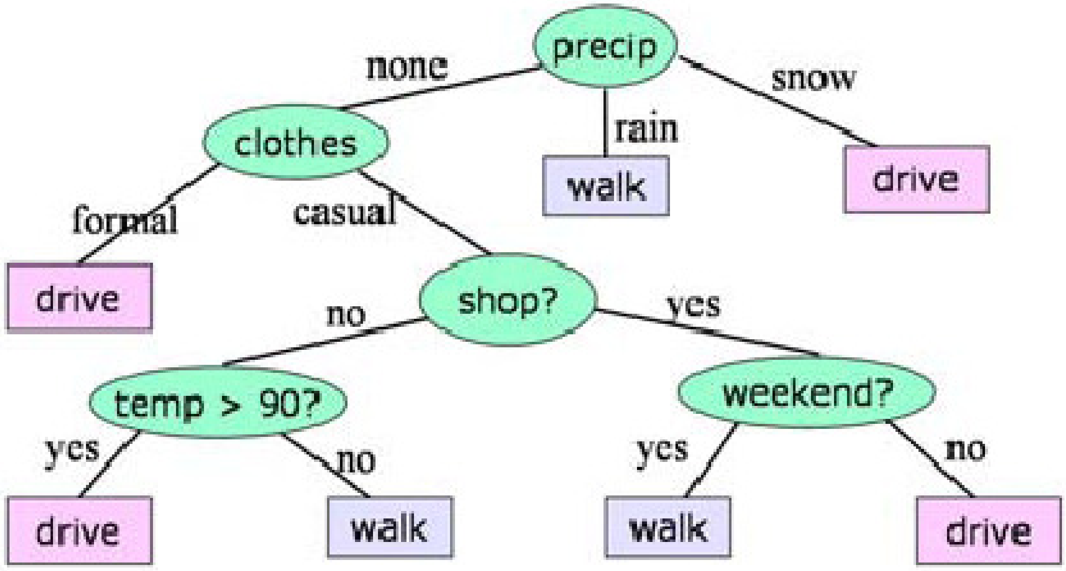
Auto MPG

- Now normalization matters a lot!
- Watch the scales on the graphs



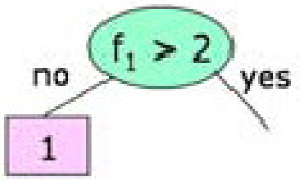
Remember Decision Trees

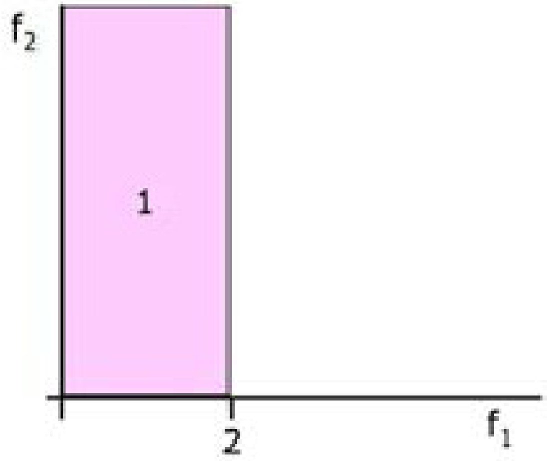
- Use all the data to build a tree of questions with answers at the leaves



Numerical Attributes

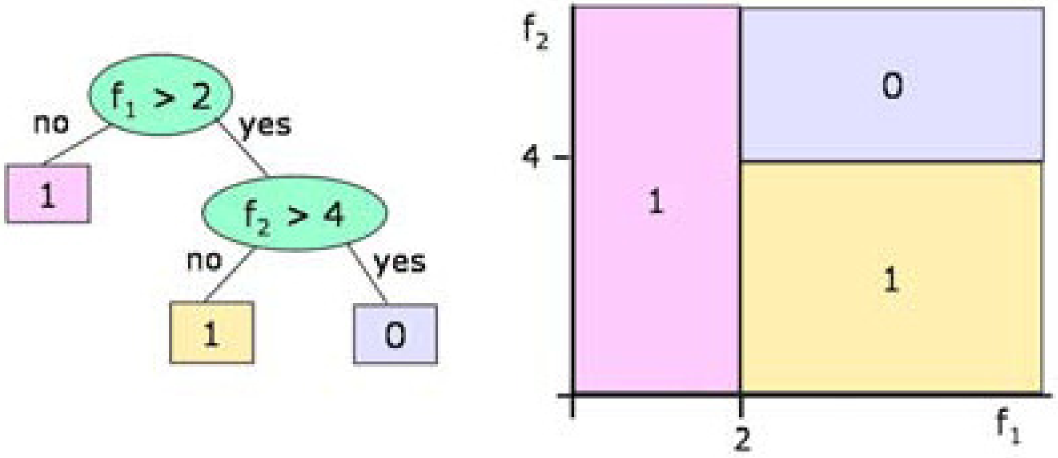
- Tests in nodes can be of the form x_j > constant
- Divides the space into axis-aligned rectangles





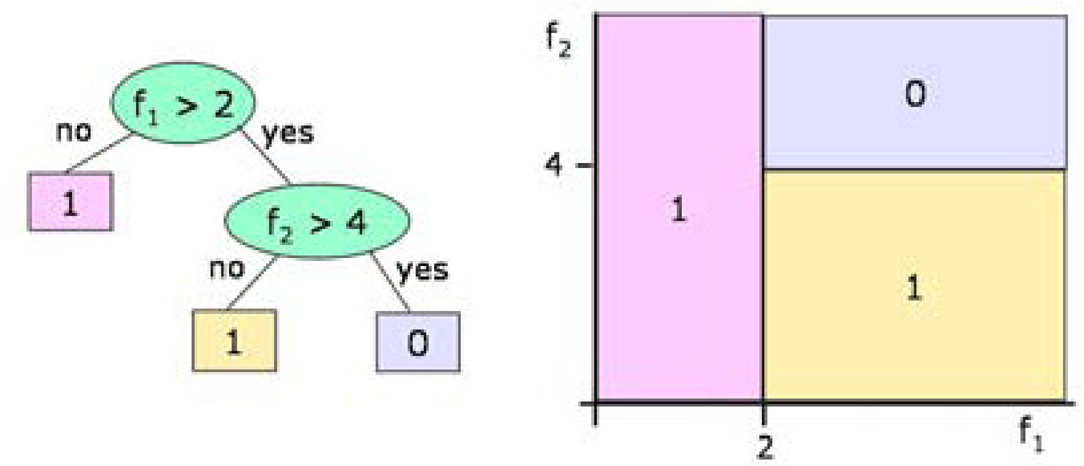
Numerical Attributes

- Tests in nodes can be of the form x_j > constant
- Divides the space into axis-aligned rectangles



Numerical Attributes

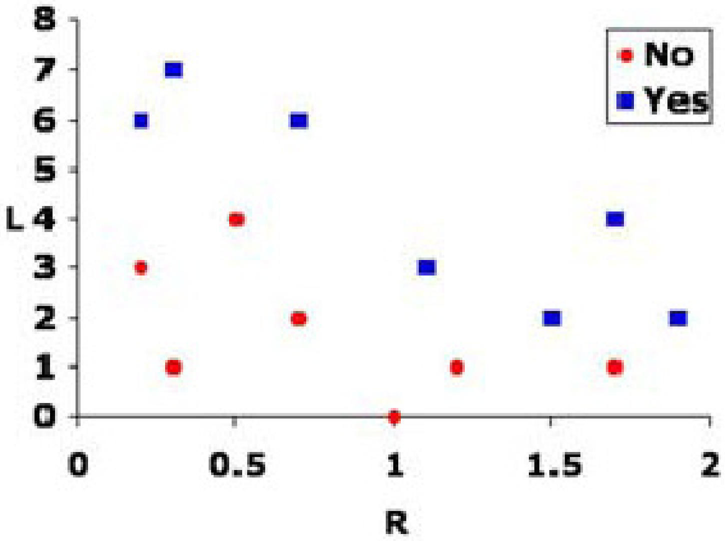
- Tests in nodes can be of the form x_j > constant
- Divides the space into axis-aligned rectangles



- Non-axis aligned hypothesis can be smaller but hard to find

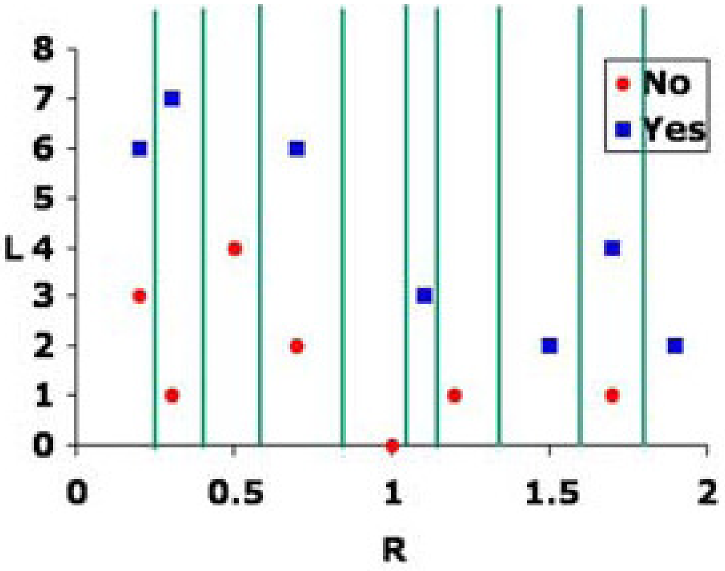
Considering Splits

- Considering a split between each point in each dimension



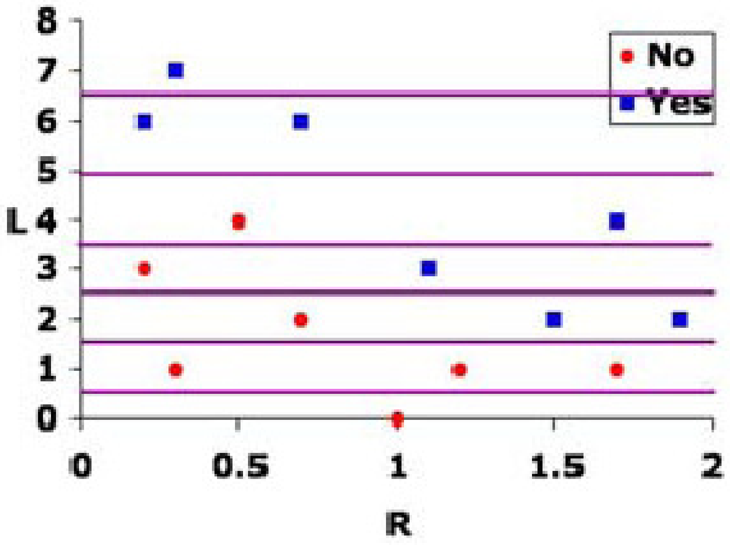
Considering Splits

- Considering a split between each point in each dimension



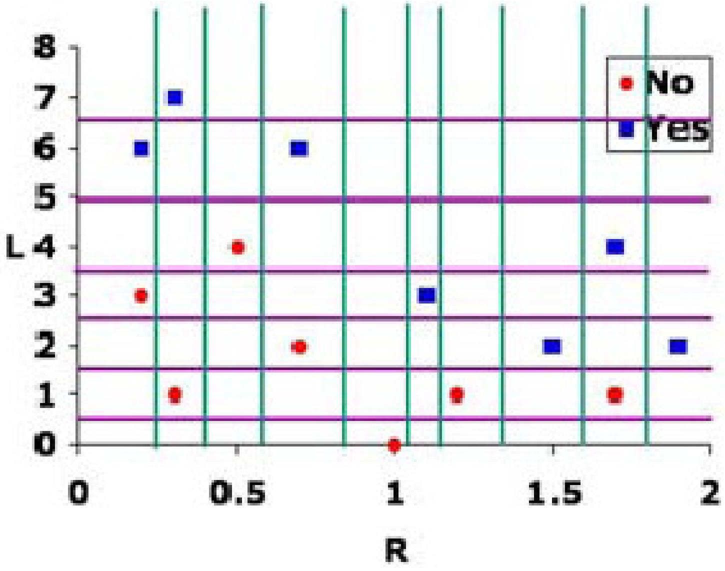
Considering Splits

- Considering a split between each point in each dimension



Considering Splits

- Considering a split between each point in each dimension



Bankruptcy Example

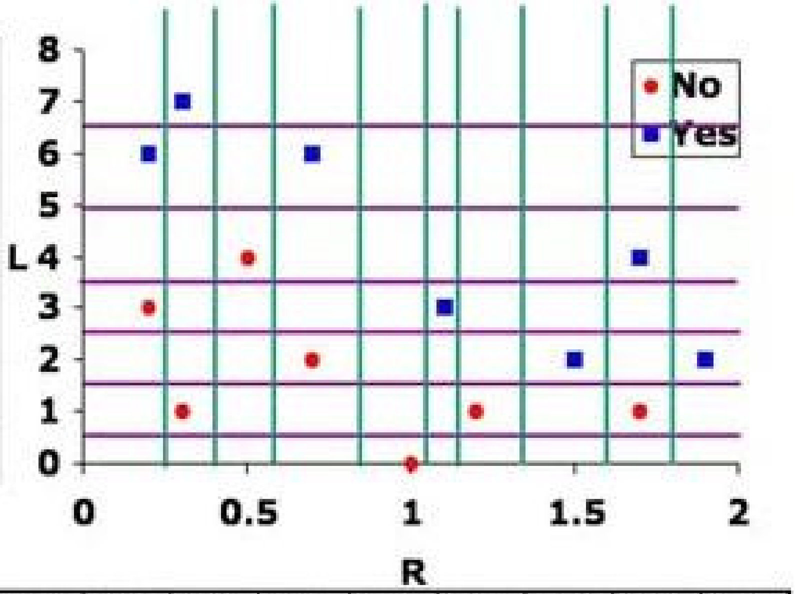
L<y	NL	PL	NR	PR	AE
6.5	7	6	0	1	0.93
5.0	7	4	0	3	0.74
3.5	6	3	1	4	0.85
2.5	5	2	2	5	0.86
1.5	4	0	3	7	0.63
0.5	1	0	6	7	0.93

# neg to left

# pos to left

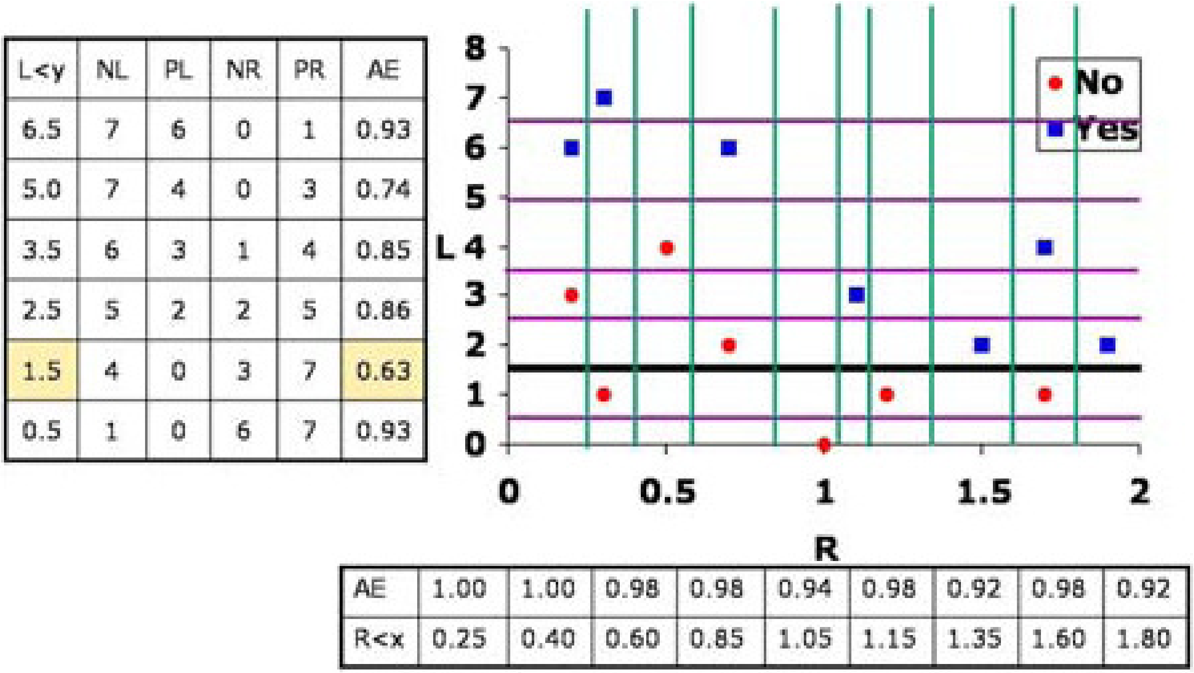
# neg to right

# pos to right



AE	1.00	1.00	0.98	0.98	0.94	0.98	0.92	0.98	0.92
R<x	0.25	0.40	0.60	0.85	1.05	1.15	1.35	1.60	1.80

Bankruptcy Example

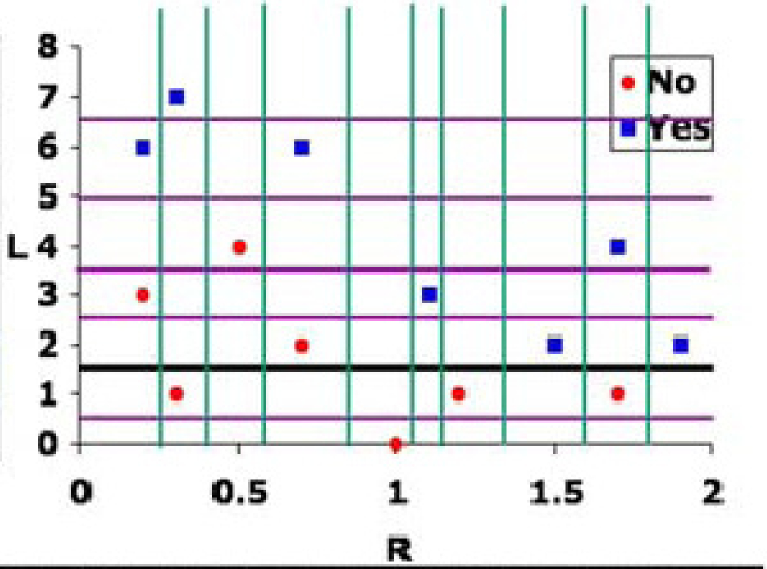


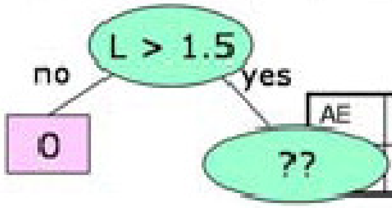
L<y	NL	PL	NR	PR	AE
6.5	7	6	0	1	0.93
5.0	7	4	0	3	0.74
3.5	6	3	1	4	0.85
2.5	5	2	2	5	0.86
1.5	4	0	3	7	0.63
0.5	1	0	6	7	0.93

AE	1.00	1.00	0.98	0.98	0.94	0.98	0.92	0.98	0.92
R<x	0.25	0.40	0.60	0.85	1.05	1.15	1.35	1.60	1.80

Bankruptcy Example

L<y	NL	PL	NR	PR	AE
6.5	7	6	0	1	0.93
5.0	7	4	0	3	0.74
3.5	6	3	1	4	0.85
2.5	5	2	2	5	0.86
1.5	4	0	3	7	0.63
0.5	1	0	6	7	0.93

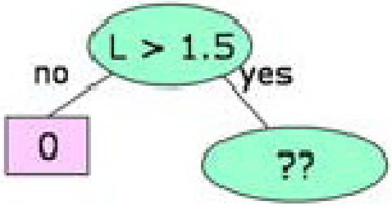


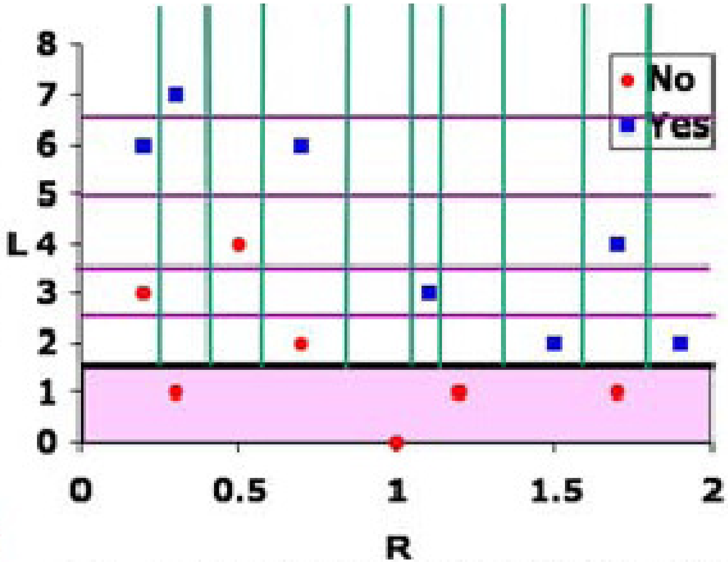


AE	1.00	1.00	0.98	0.98	0.94	0.98	0.92	0.98	0.92
	0.25	0.40	0.60	0.85	1.05	1.15	1.35	1.60	1.80

Bankruptcy Example

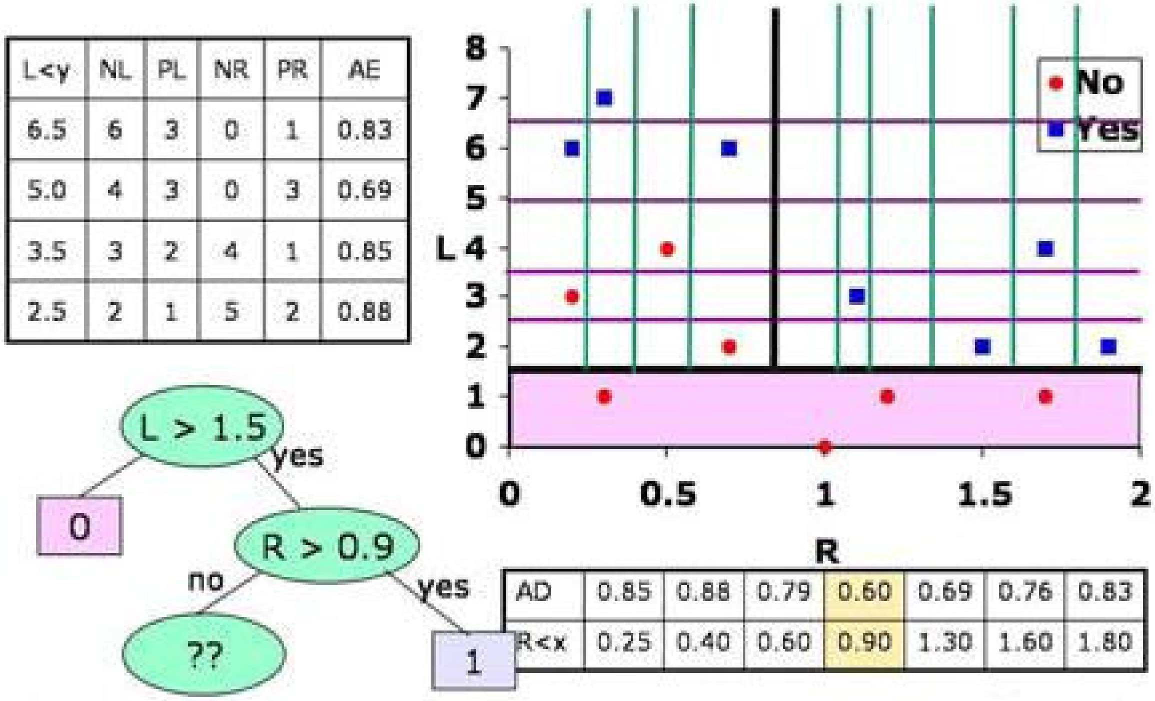
L<y	NL	PL	NR	PR	AE
6.5	6	3	0	1	0.83
5.0	4	3	0	3	0.69
3.5	3	2	4	1	0.85
2.5	2	1	5	2	0.88





AE	0.85	0.88	0.79	0.60	0.69	0.76	0.83
R<x	0.25	0.40	0.60	0.90	1.30	1.60	1.80

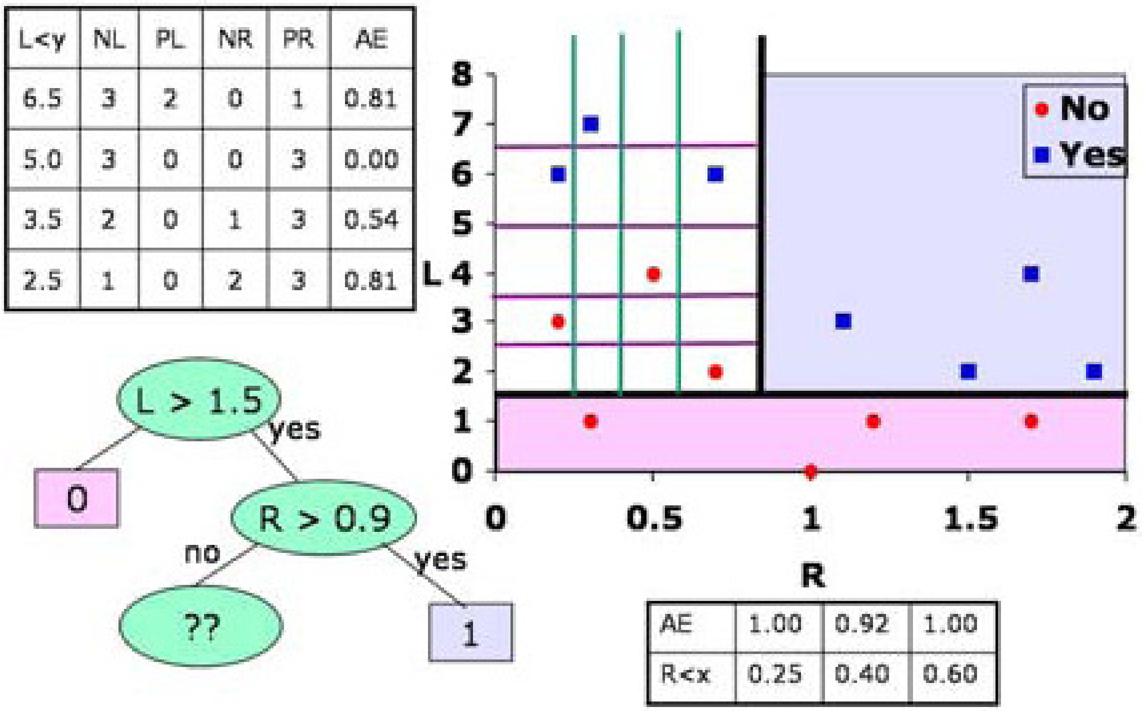
Bankruptcy Example



L<y	NL	PL	NR	PR	AE
6.5	6	3	0	1	0.83
5.0	4	3	0	3	0.69
3.5	3	2	4	1	0.85
2.5	2	1	5	2	0.88

AD	0.85	0.88	0.79	0.60	0.69	0.76	0.83
R<x	0.25	0.40	0.60	0.90	1.30	1.60	1.80

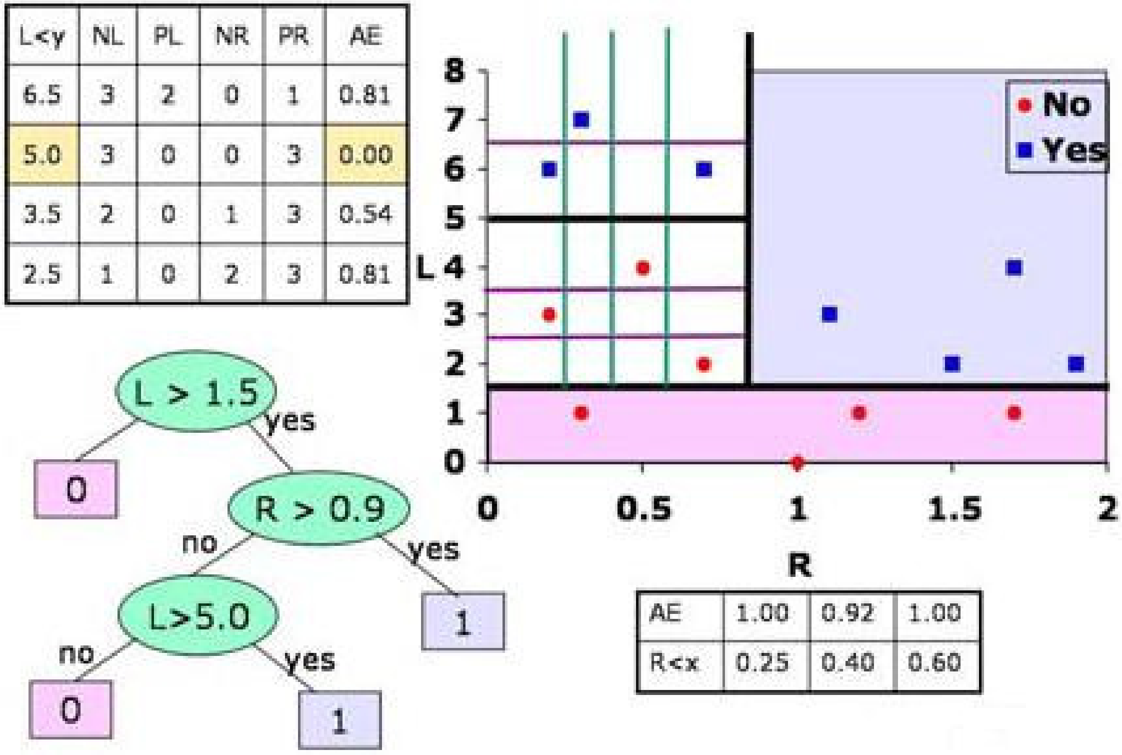
Bankruptcy Example



L<y	NL	PL	NR	PR	AE
6.5	3	2	0	1	0.81
5.0	3	0	0	3	0.00
3.5	2	0	1	3	0.54
2.5	1	0	2	3	0.81

AE	1.00	0.92	1.00
R<x	0.25	0.40	0.60

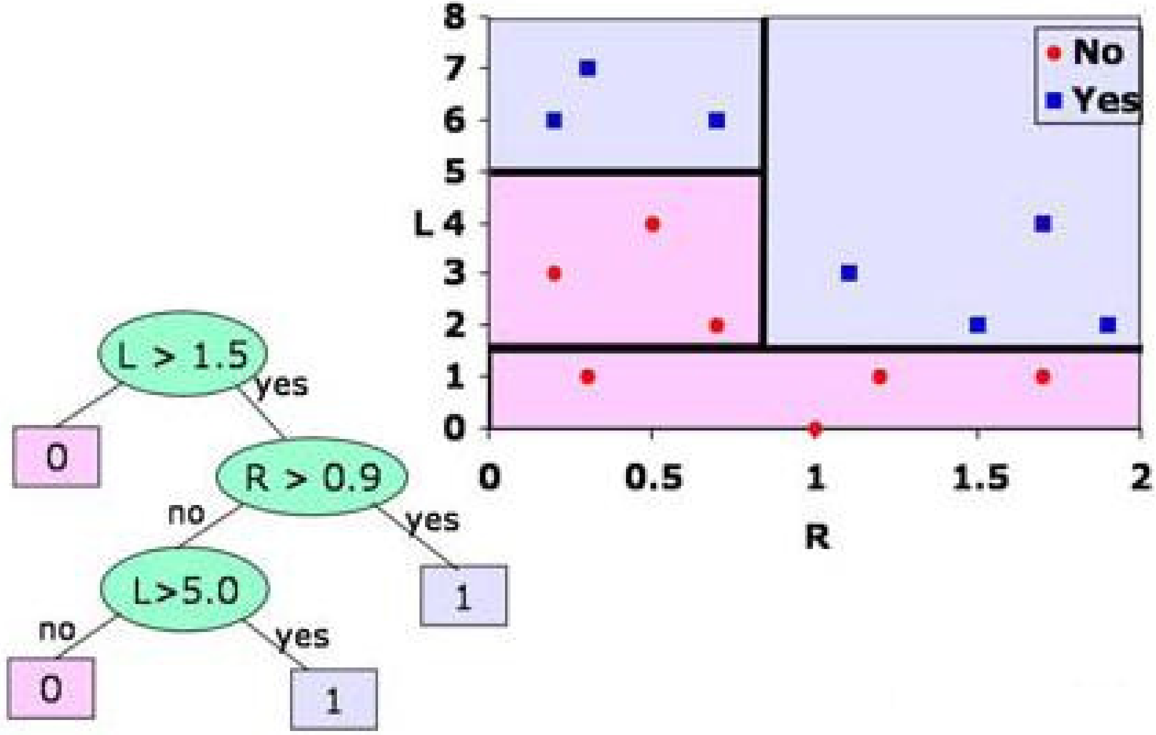
Bankruptcy Example



L<y	NL	PL	NR	PR	AE
6.5	3	2	0	1	0.81
5.0	3	0	0	3	0.00
3.5	2	0	1	3	0.54
2.5	1	0	2	3	0.81

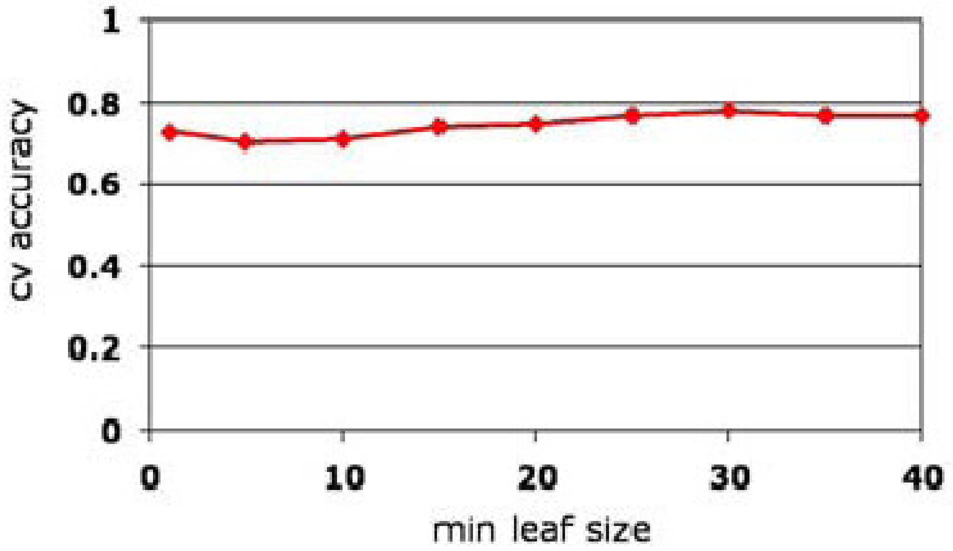
AE	1.00	0.92	1.00
R<x	0.25	0.40	0.60

Bankruptcy Example

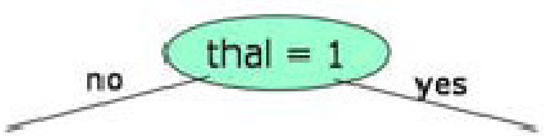


Heart Disease

- Best performance (.77) slightly worse than nearest neighbor (.81)

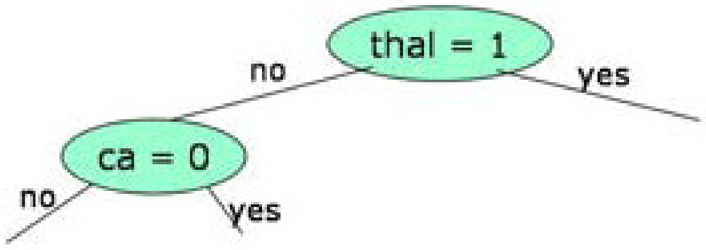


Heart Disease



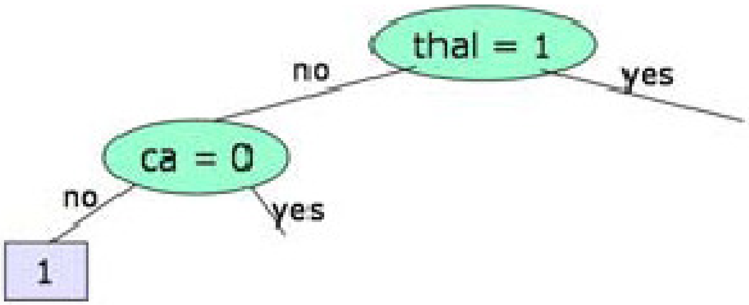
thal = 1: normal exercise thallium scintigraphy test

Heart Disease



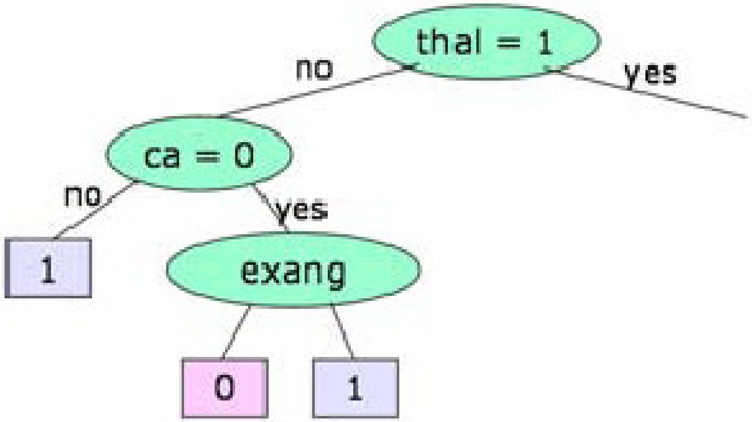
thal = 1: normal exercise thallium scintigraphy test
ca = 0: no vessels colored by fluoroscopy

Heart Disease



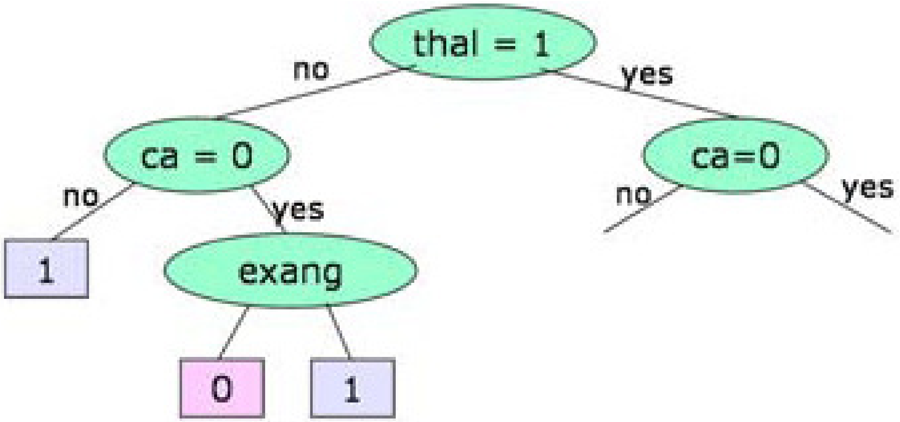
thal = 1: normal exercise thallium scintigraphy test
ca = 0: no vessels colored by fluoroscopy

Heart Disease



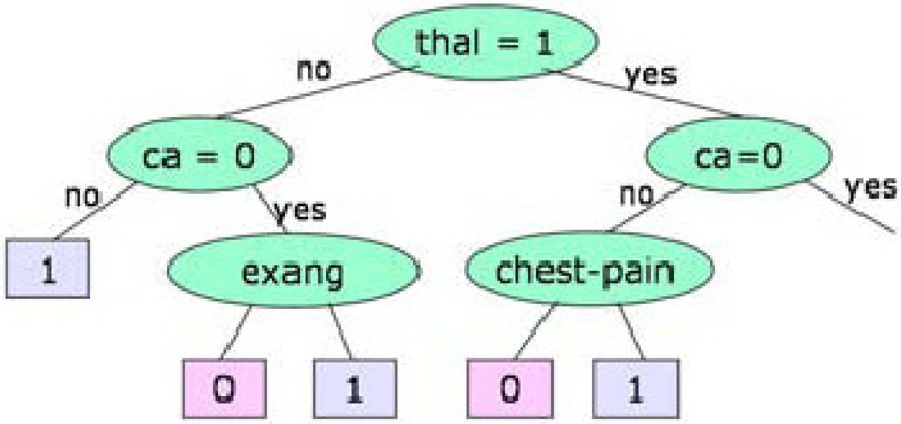
thal = 1: normal exercise thallium scintigraphy test
ca = 0: no vessels colored by fluoroscopy
exang: exercise induced angina

Heart Disease



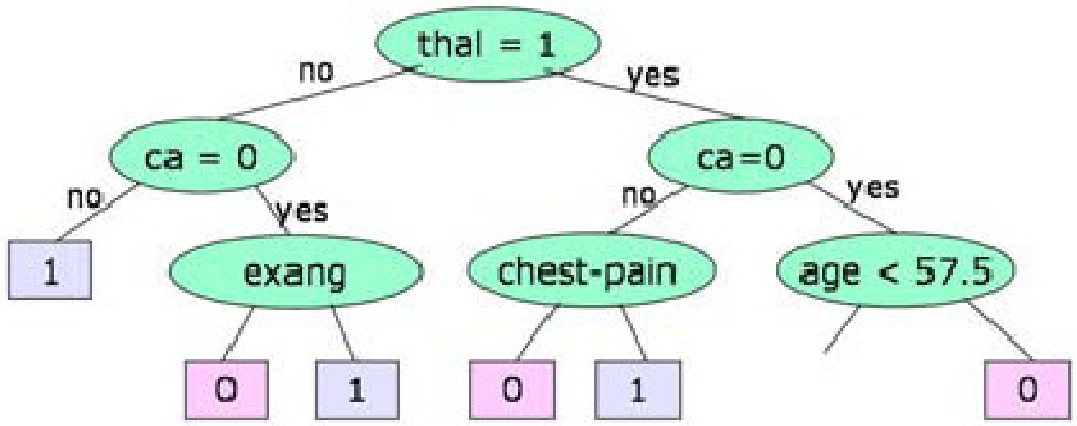
thal = 1: normal exercise thallium scintigraphy test
ca = 0: no vessels colored by fluoroscopy
exang: exercise induced angina

Heart Disease



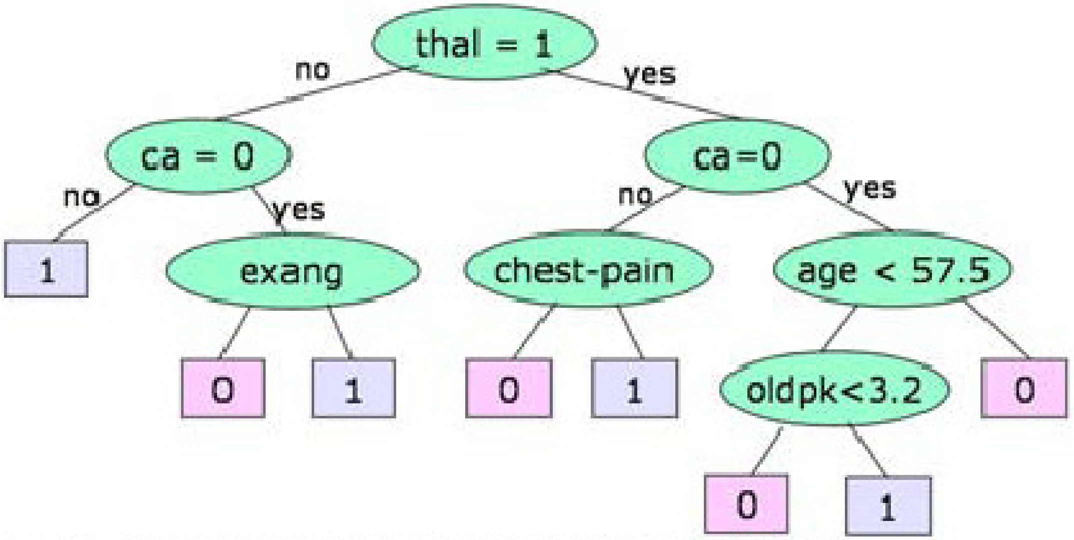
thal = 1: normal exercise thallium scintigraphy test
ca = 0: no vessels colored by fluoroscopy
exang: exercise induced angina

Heart Disease



thal = 1: normal exercise thallium scintigraphy test
ca = 0: no vessels colored by fluoroscopy
exang: exercise induced angina

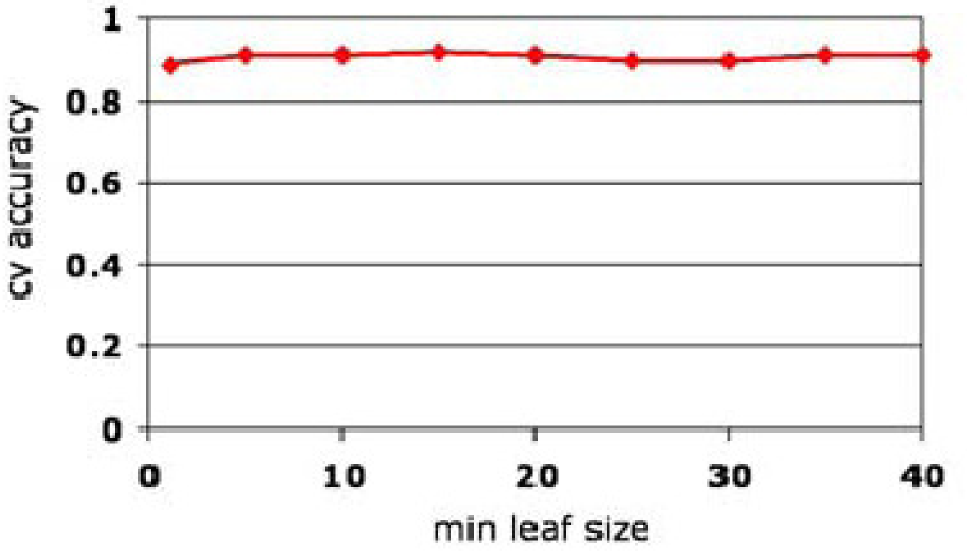
Heart Disease



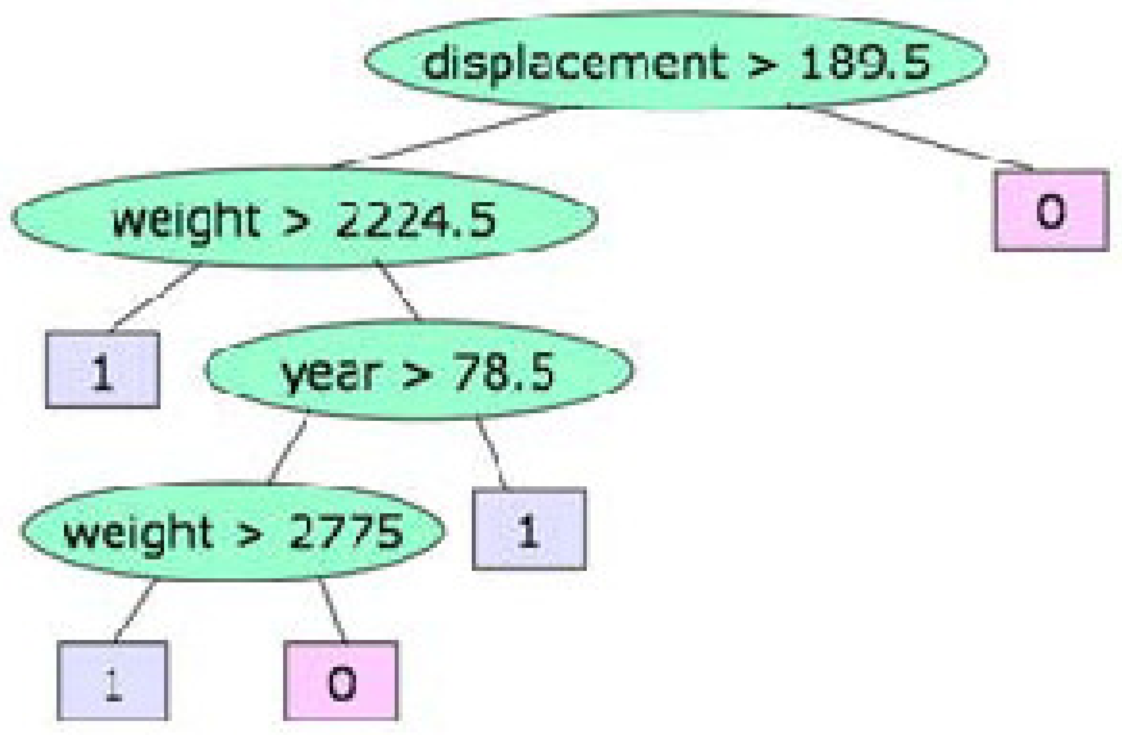
thal = 1: normal exercise thallium scintigraphy test
ca = 0: no vessels colored by fluoroscopy
exang: exercise induced angina
oldpk: feature of cardiogram

Auto MPG

- Performance (.91) essentially the same as nearest neighbor



More than 22 MPG?


displacement > 189.5
weight > 2224.5
0
1
year > 78.5
weight > 2775
1
1
0