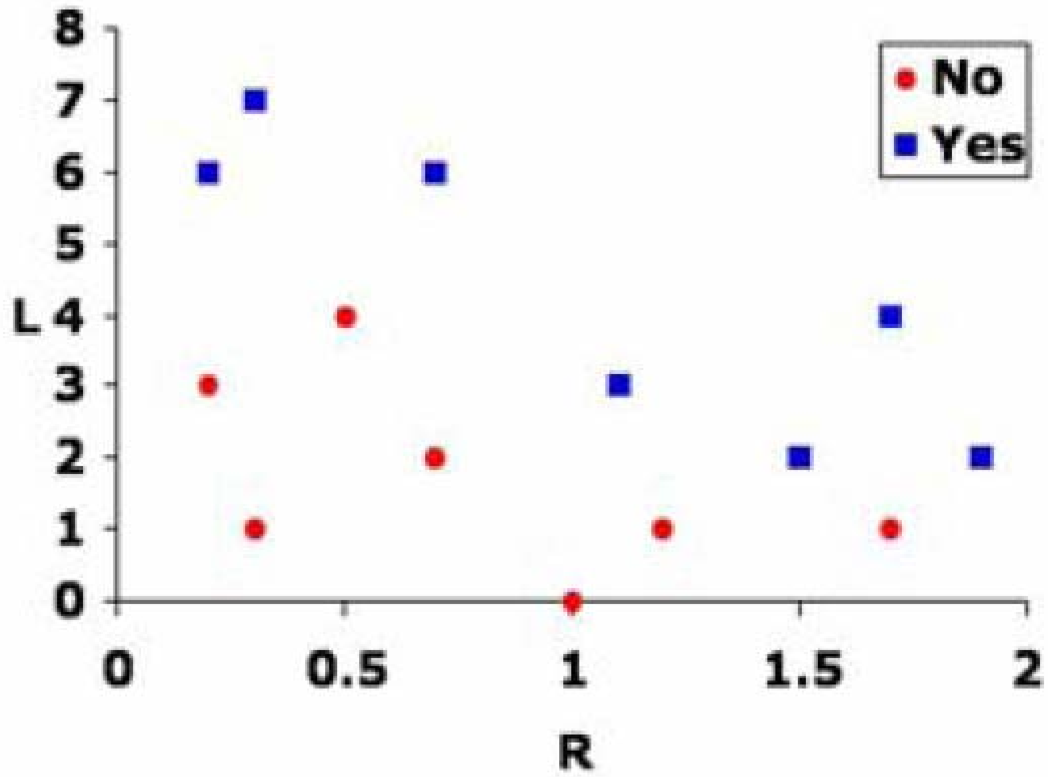
K-Nearest Neighbor

Predicting Bankruptcy

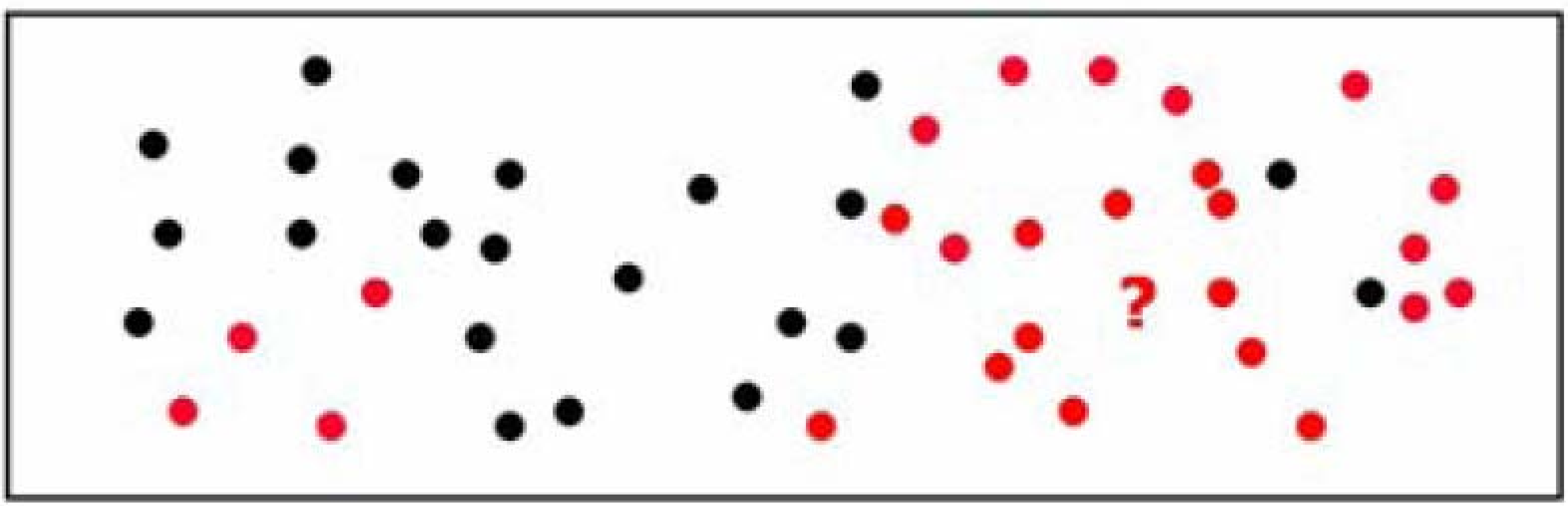
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1.0	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Love thy 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 to have mean 0 and variance 1:

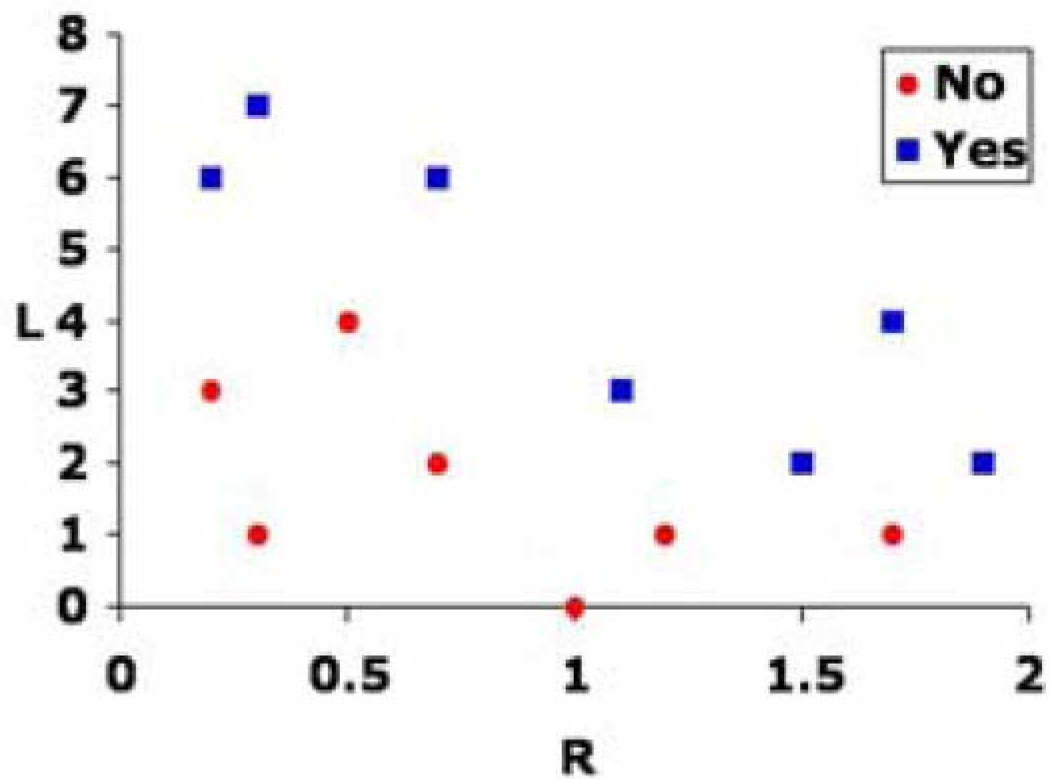
$$x' = \frac{x - \bar{x}}{\sigma_x}$$

average

standard deviation

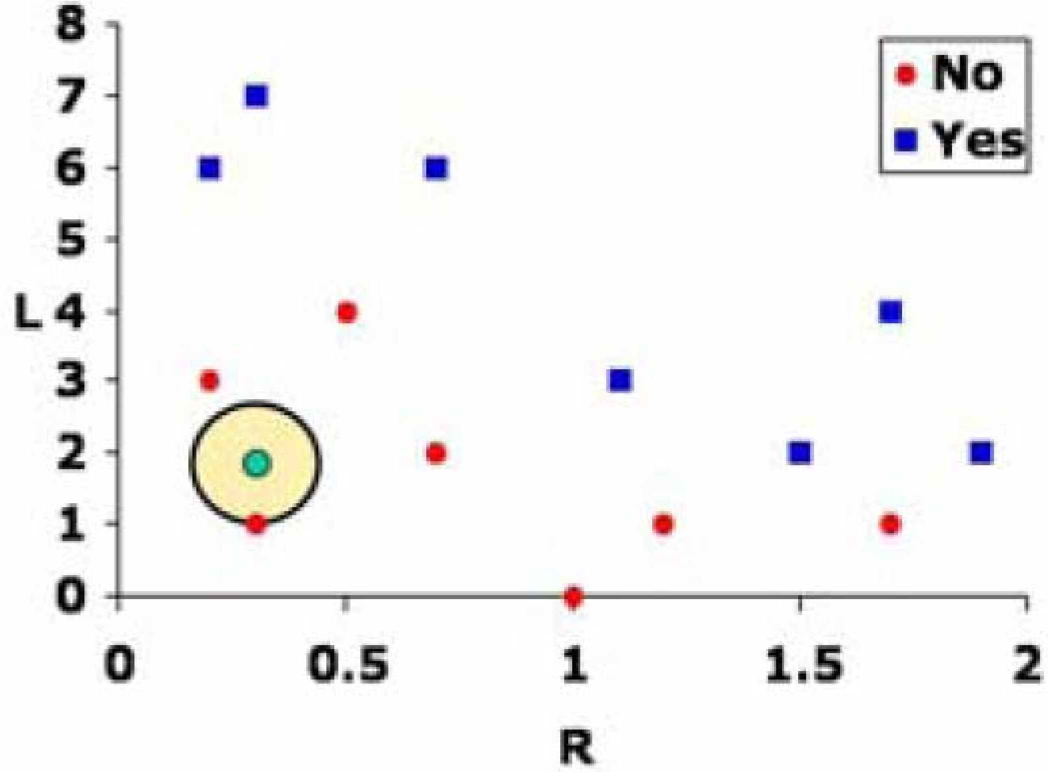
- Or, build knowledge in by scaling features differently

Predicting Bankruptcy



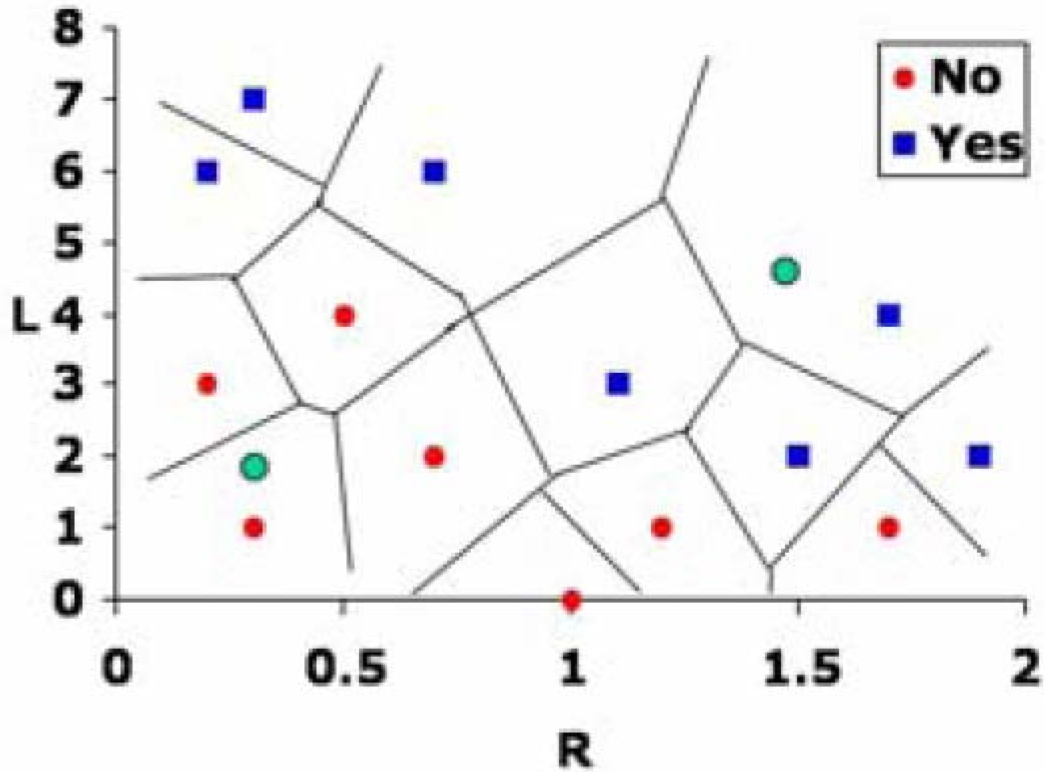
$$D(x^i, x^k) = \sqrt{\sum_j (L^i - L^k)^2 + (5R^i - 5R^k)^2}$$

Predicting Bankruptcy



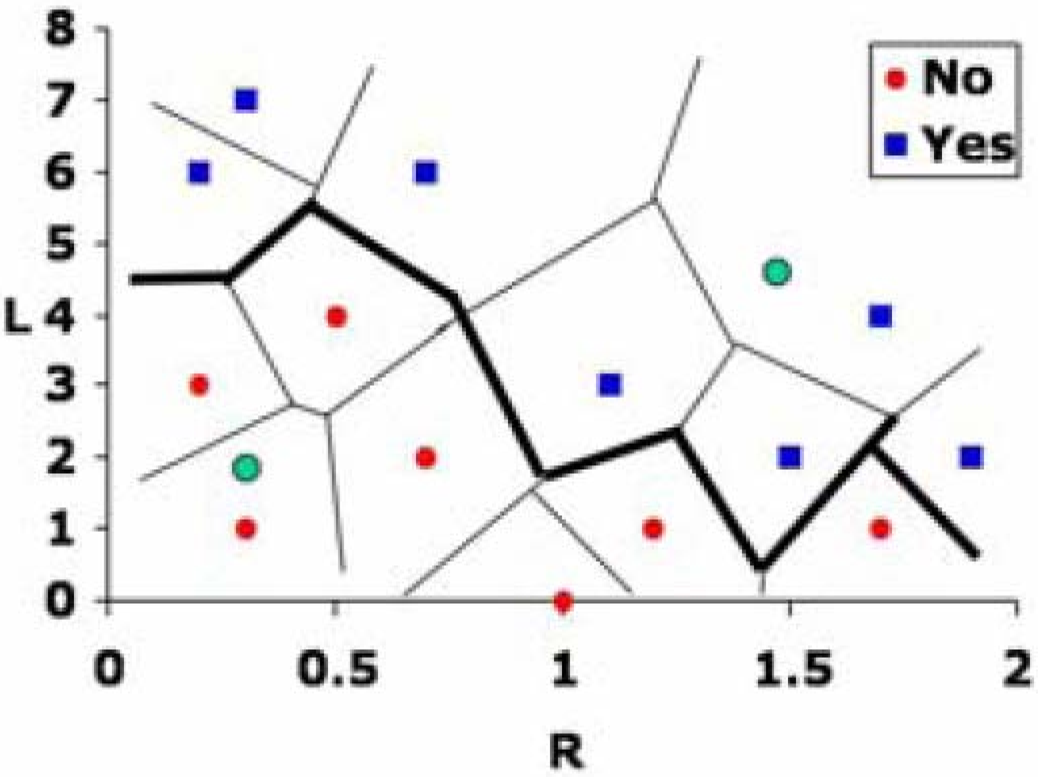
$$D(x^i, x^k) = \sqrt{\sum_j (L^i - L^k)^2 + (5R^i - 5R^k)^2}$$

Hypothesis



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

Hypothesis

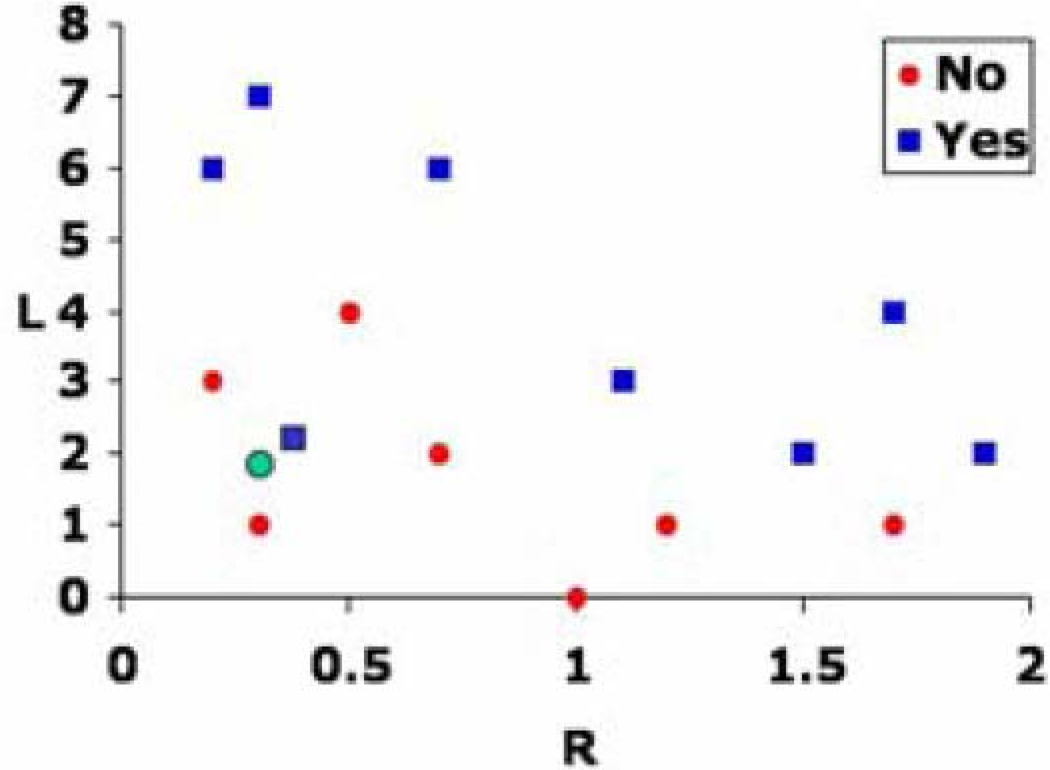


$$D(x^i, x^k) = \sqrt{\sum_j (L^i - L^k)^2 + (5R^i - 5R^k)^2}$$

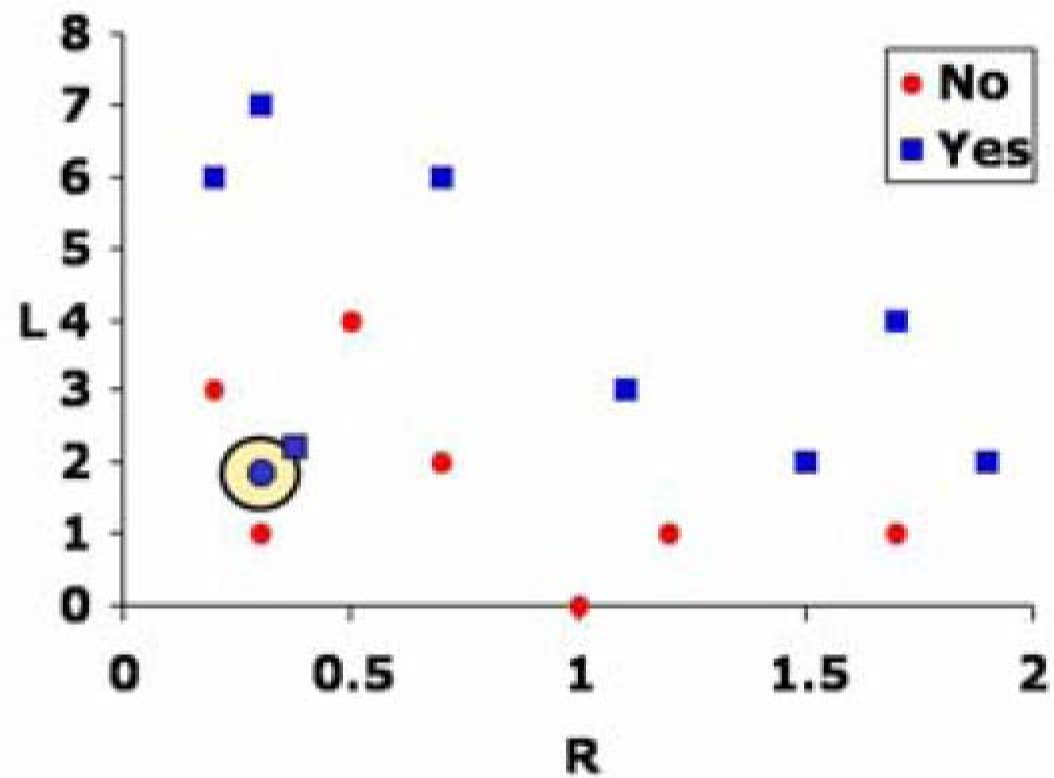
Time and Space

- Learning is fast
- Lookup takes about m*n computations
  - storing data in a clever data structure (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 boundary

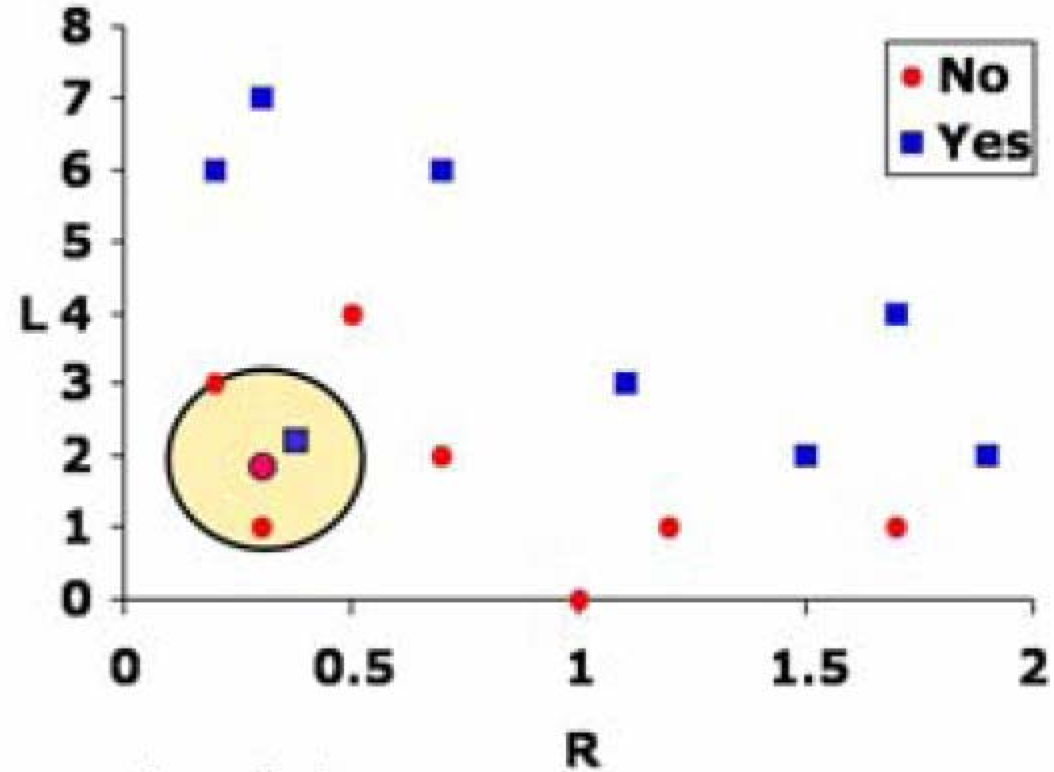
Noise



Noise



k-Nearest Neighbor



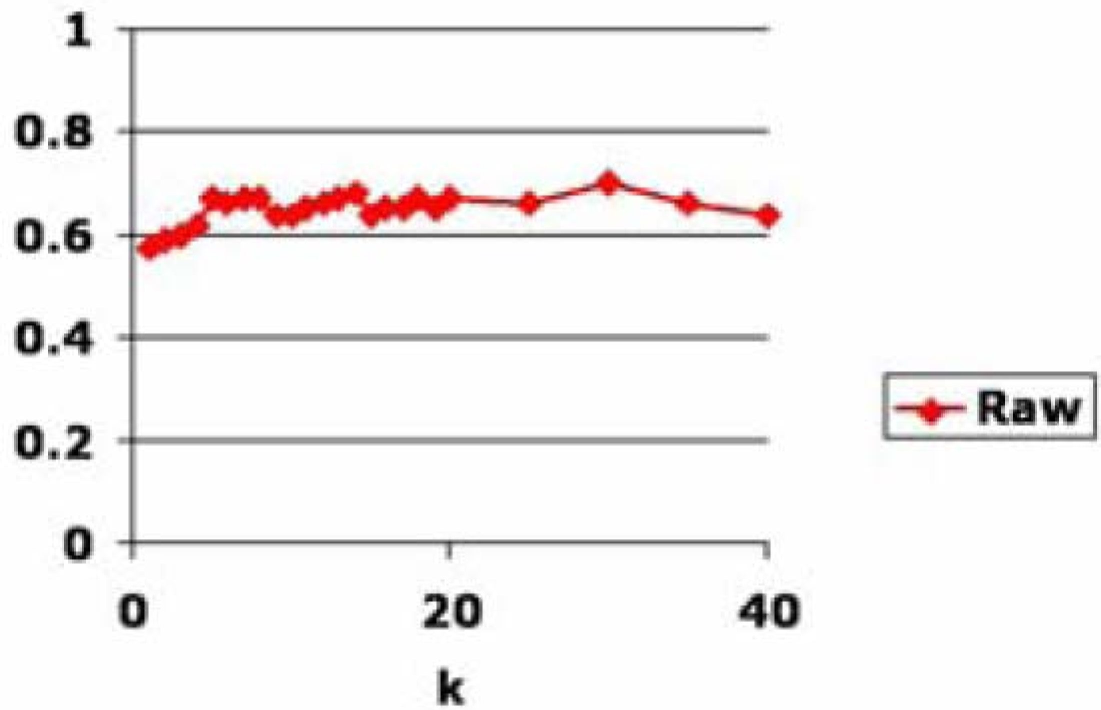
- Find the k nearest points
- Predict output according to the majority

Test Domains

- Heart Disease: predict whether a person has significant narrowing of the arteries, based on tests
  - 26 features
  - 297 data points
- Auto MPG: predict whether a car gets more than 22 miles per gallon, based on attributes of car
  - 12 features
  - 385 data points

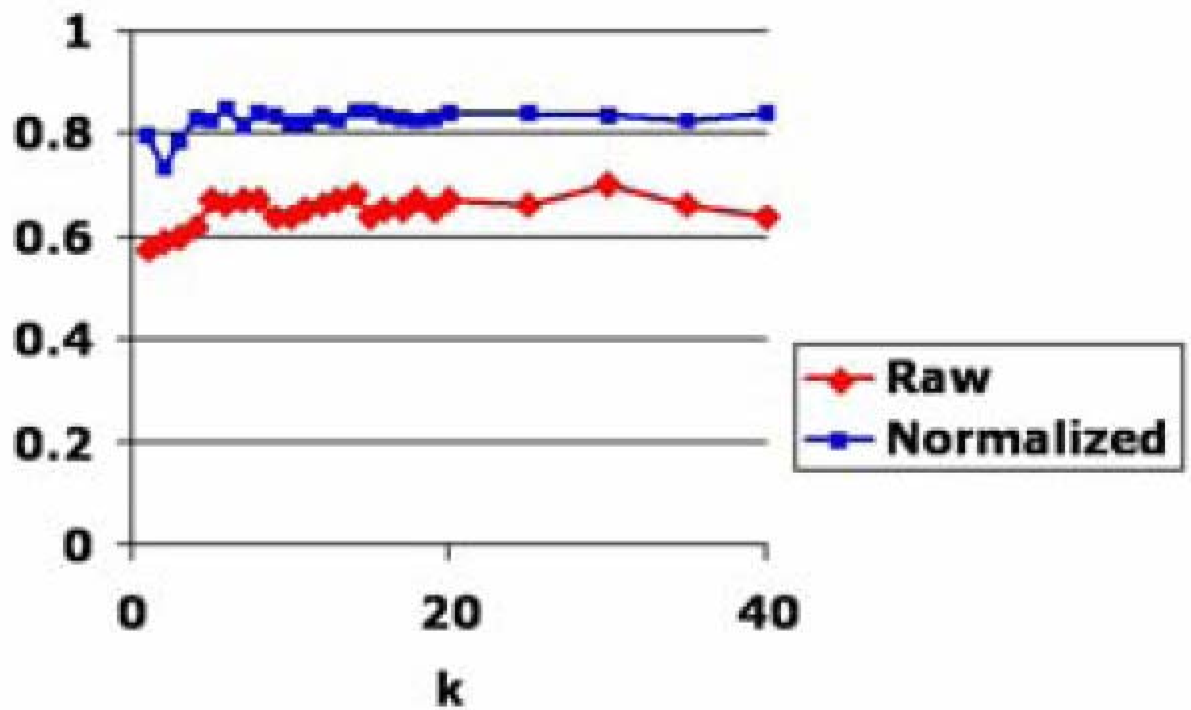
Heart Disease

- Relatively insensitive to k



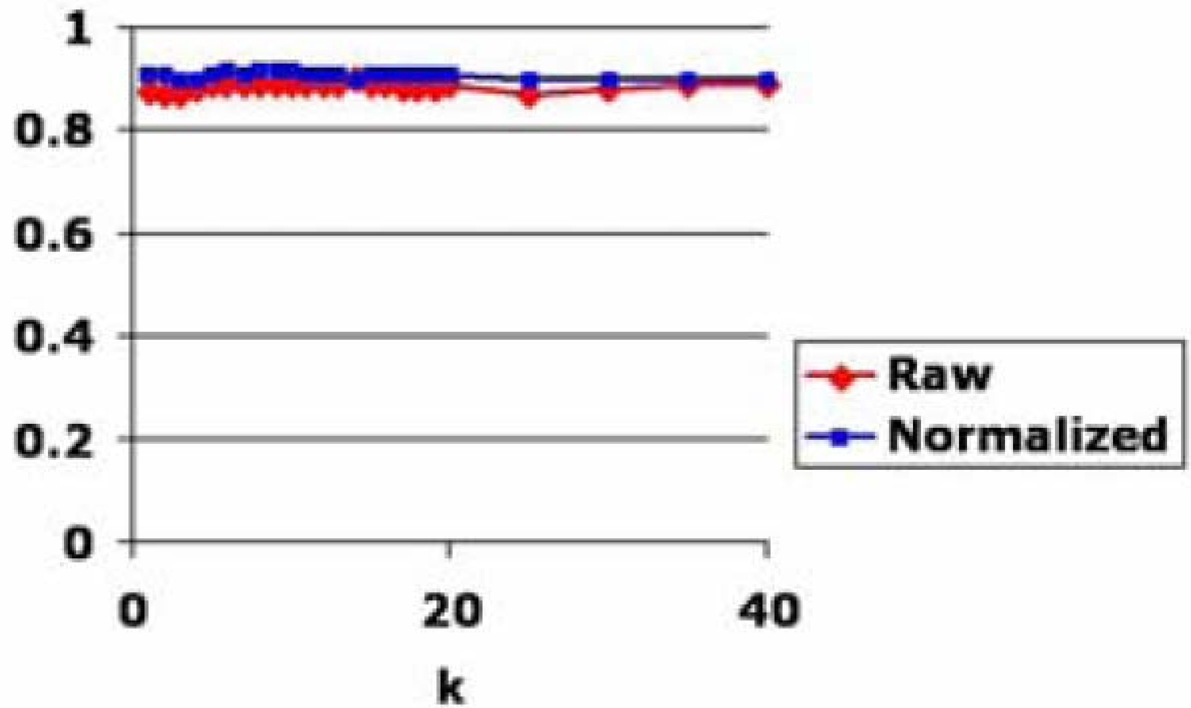
Heart Disease

- Relatively insensitive to k
- Normalization matters!



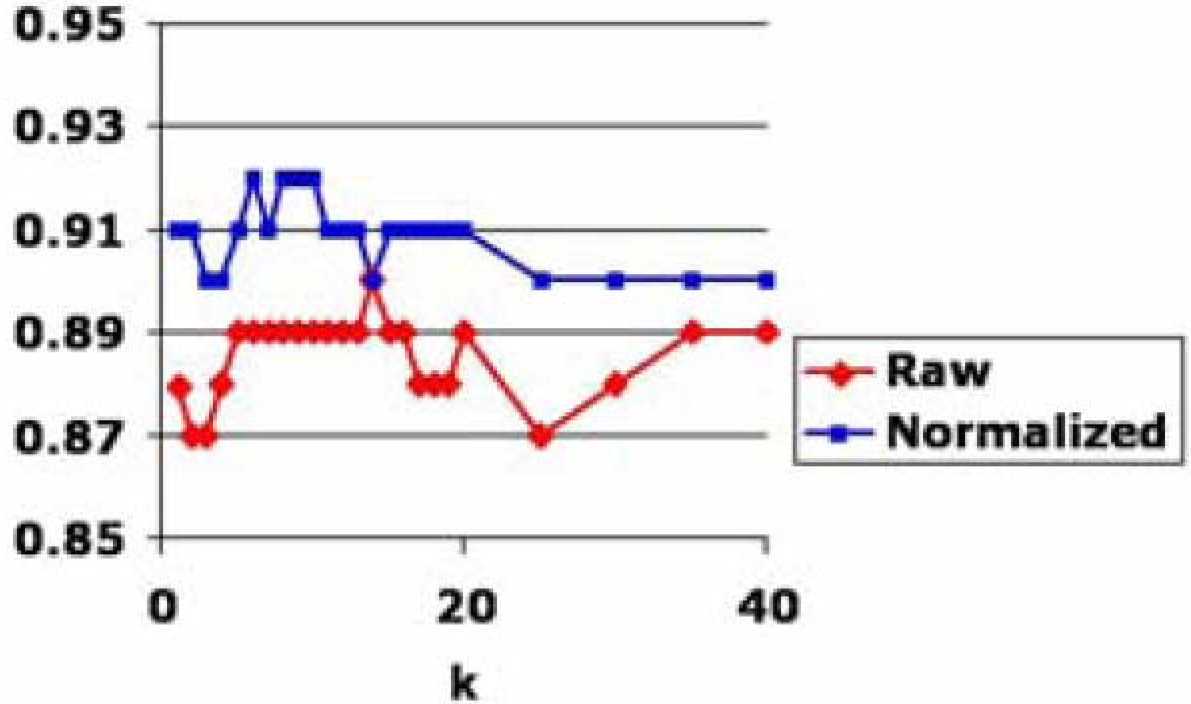
Auto MPG

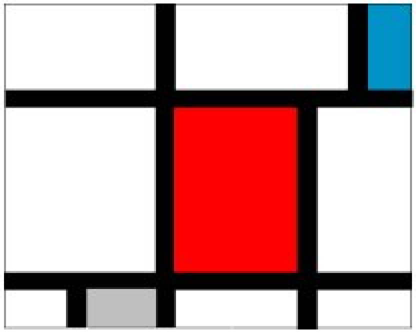
- Relatively insensitive to k
- Normalization doesn't matter much

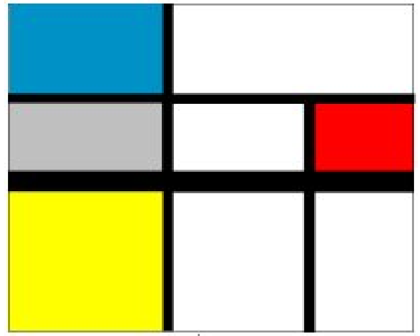


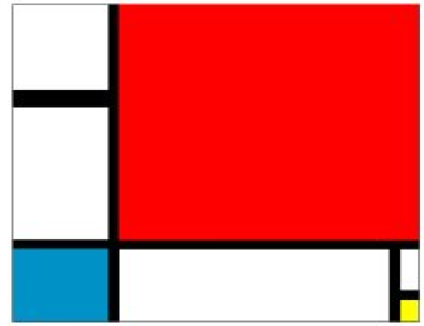
Auto MPG

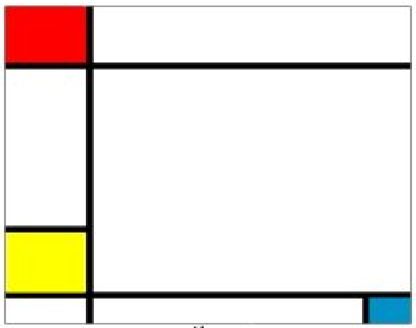
- Now normalization matters a lot!
- Watch the scales on your graphs

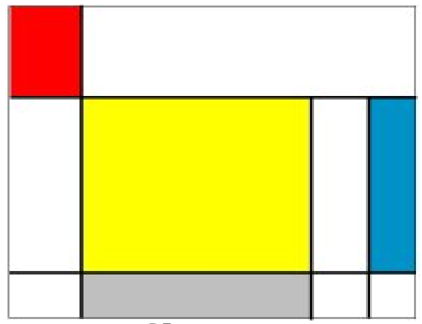


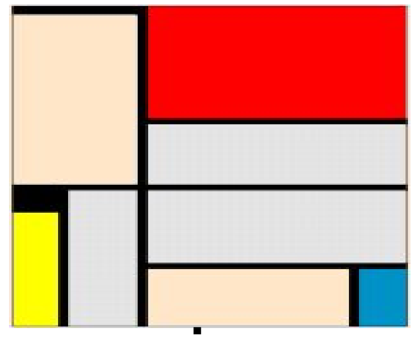

one

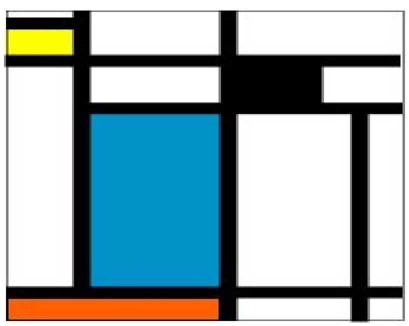

two

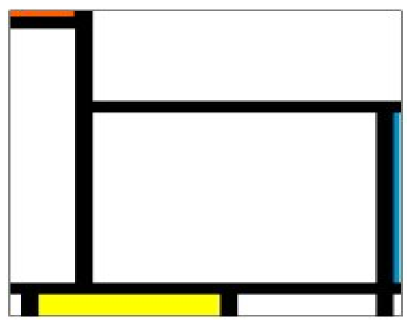

three


four


five


six


seven


Eight ?

Training data

Number	Lines	Line types	Rectangles	Colours	Mondrian?
1	6	1	10	4	*No*
2	4	2	8	5	*No*
3	5	2	7	4	Yes
4	5	1	8	4	Yes
5	5	1	10	5	*No*
6	6	1	8	6	Yes
7	7	1	14	5	*No*

Number	Lines	Line types	Rectangles	Colours	Mondrian?
8	7	2	9	4	

Normalised training data

Number	Lines	Line types	Rectangles	Colours	Mondrian?
1	0.632	-0.632	0.327	-1.021	*No*
2	-1.581	1.581	-0.588	0.408	*No*
3	-0.474	1.581	-1.046	-1.021	Yes
4	-0.474	-0.632	-0.588	-1.021	Yes
5	-0.474	-0.632	0.327	0.408	*No*
6	0.632	-0.632	-0.588	1.837	Yes
7	1.739	-0.632	2.157	0.408	*No*

Test instance

Number	Lines	Line types	Rectangles	Colours	Mondrian?
8	1.739	1.581	-0.131	-1.021	

Distances of test instance from training data

Example	Distance of test from example	Mondrian?
1	2.517	*No*
2	3.644	*No*
3	2.395	Yes
4	3.164	Yes
5	3.472	*No*
6	3.808	Yes
7	3.490	*No*

Classification

1-NN	Yes
3-NN	Yes
5-NN	*No*
7-NN	*No*