Nearest neighbors: similarity/distance

Lecture 09

by Marina Barsky

K-NN classifier: lazy classifier

Training set

Late payments, L	Spending ratio, R	Bankruptcy
3	Very low	No
1	Very low	No
4	Low	No
2	Low	No
0	Normal	No
1	Medium	No
1	High	No
6	Very low	Yes
7	Very low	Yes
6	Low	Yes
3	Normal	Yes
2	Medium	Yes
4	High	Yes
2	High	Yes

New sample

L	R	B
2	Low	?

Classify

=

Model

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

K-NN classification algorithm

Input:

set ***T*** of *N* labeled records,
K,
instance ***A*** to classify

Classification:

for *i* **from** 1 **to** *N*
compute ***distance*** *d* ($\boldsymbol{A}$,$\boldsymbol{T}_i$)
sort ***T*** *asc* by *d* ($\boldsymbol{A}$,$\boldsymbol{T}_i$) into $\boldsymbol{T}_{sorted}$
from top ***K*** records in $\boldsymbol{T}_{sorted}$
extract class labels $\boldsymbol{L}_{1\ldots K}$

Output:

return ***combination*** ($\boldsymbol{L}_{1\ldots K}$)

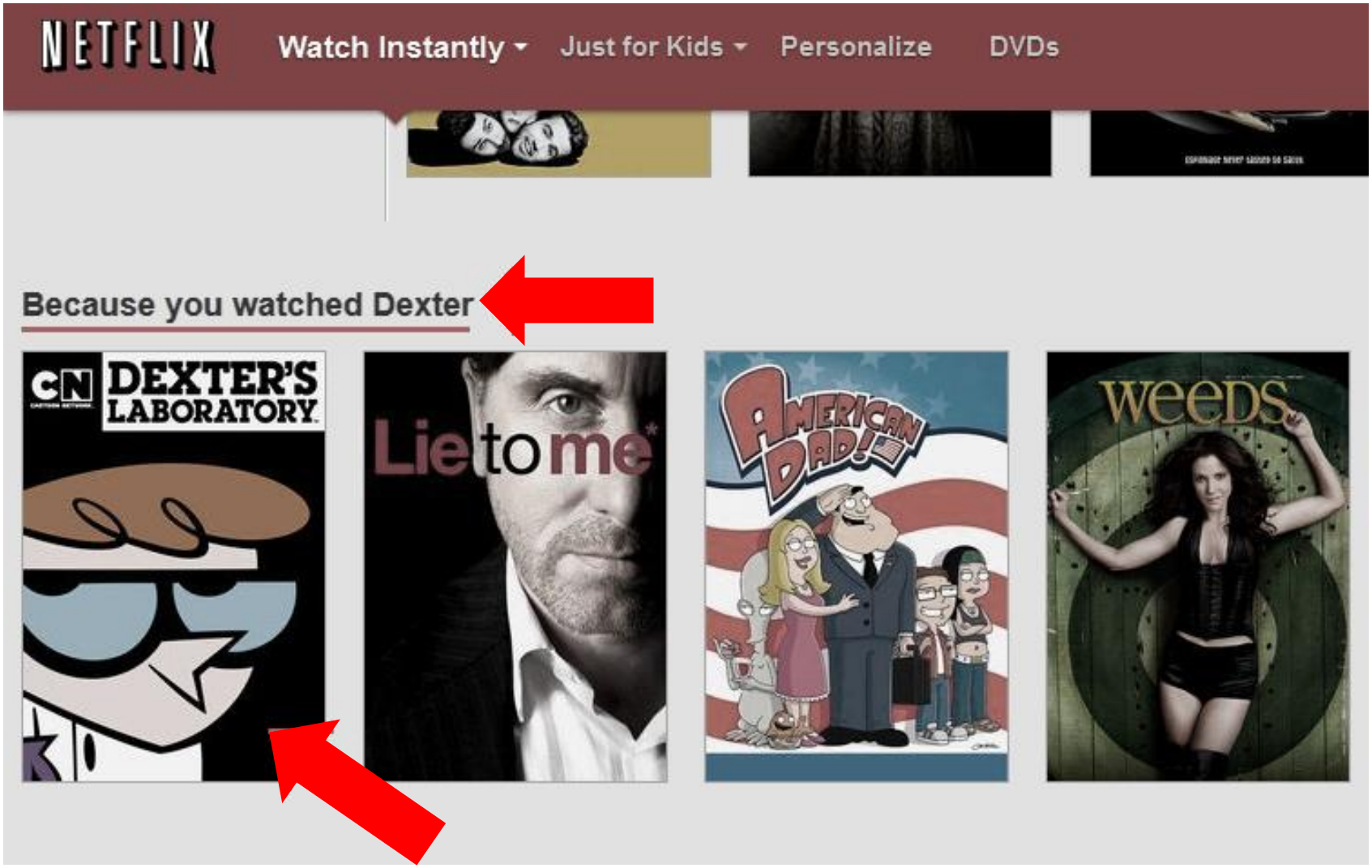
K-NN: round 2

I. Distance/similarity between data records
II. How many neighbors: choice of K
III. Combining neighbor votes
IV. How many features (dimensions)

K-NN: round 2

I. Distance/similarity between data records
II. How many neighbors: choice of K
III. Combining neighbor votes
IV. How many features (dimensions)

How do we define proximity?



Numeric *proximity* (similarity or distance) between data records

- Combination of proximity measures for each attribute
- Each attribute is considered a separate and independent (in this approach) dimension of the data

- First step: translate all fields into numeric variables, to be able to compute similarity (distance) across each dimension

Types of attributes

1. True measures (continuous)
2. Ranks (ordinal)
3. Categorical (nominal)

The distances are Increasingly harder to convert into a numeric scale

How do we define the proximity measure for a single attribute of each type?

1. True measures

- True measures measure the value from a meaningful "0" point. The ratio between values is meaningful, and the distance is just an absolute difference of values.

- Examples: age, weight, length

2. Ordinal (Ranks)

- These values have an order, but the distance between different ranks is not defined

2. Ordinal (Ranks)

Example 1:

quality attribute of a product : {poor, fair, OK, good, wonderful}

Order is important, but exact difference between values is undefined

Solution: map the values of the attribute to successive integers

{poor=0, fair=1, OK=2, good=3, wonderful=4}

Dissimilarity (distance)

$d(p,q) = |p - q| / (\text{max\_d} - \text{min\_d})$

e.g. d(wonderful, fair) = |4-1| / (4-0) = .75

Not always meaningful, but the best we can do

Similarity

$s(p,q) = 1 - d(p,q)$ **e.g.** s(wonderful, fair) = .25

2. Ordinal (Ranks)

Example 2:

Top 10 swimmers - 50m Fly				
1	KONOVALOV, Nikita	88	RUS	22.70
2	GOVOROV, Andriy	92	UKR	22.70
3	LEVEAUX, Amaury	85	FRA	22.74
4	CZERNIAK, Konrad	89	POL	22.77
5	KOROTYSHKIN, Evgeny	83	RUS	22.88
6	EIBLER, Steffen	87	GER	22.89
7	FESIKOV, Sergey	89	RUS	22.96
8	HEERSBRANDT, Francois	89	BEL	22.98
9	MUNOZ PEREZ, Rafael	88	ESP	23.07
10	JAMES, Antony	89	GBR	23.14

Distance between athlet 3 and 1 (0.04 sec) is not the same as distance between 10 and 8 (0.16). It is better to use the numeric attributes (actual time) which contributed to this ranking

3. Categorical (nominal) attributes

- Each value is one of a set of unordered categories. We can only tell that $X \neq Y$, but not how much X is greater than Y.

- Example: ice cream *pistachio* is not equal to *butter pecan*, but we cannot tell which one is greater and which one is closer to *black cherry* ice cream
- The general approach: if equal then similarity = 1, if not equal then similarity = 0

Summary on proximity measures for a single attribute

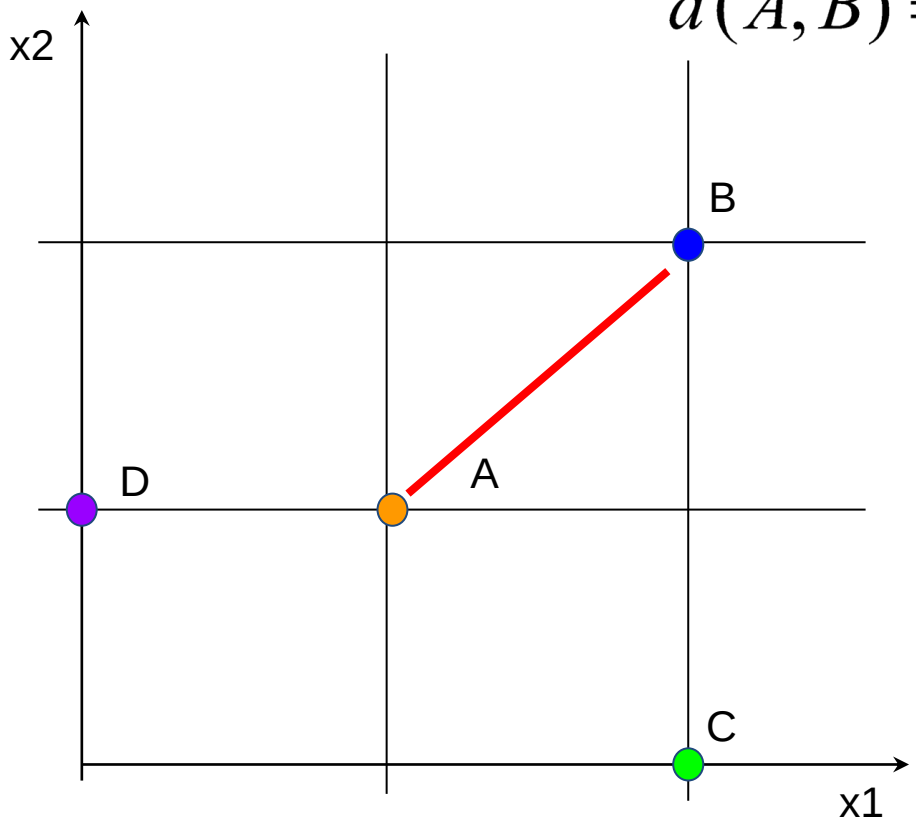
Attribute type	Distance (dissimilarity)	Similarity
True measures	d=|x-y|	s=-d, s=1/(1+d), s=1-(d-min_d)/(max_d-min_d)
Ordinal	d=|x-y|/(n-1) (values mapped to integers 0 to n-1 where n is the number of values)	s=1-d
Nominal (Categorical)	d= 0 if x=y d=1 if x≠y	s=1 if x=y s=0 if x≠y

Combining measures of separate attributes into a proximity measure between a pair of data records

- Hundreds of similarity measures were proposed
- We will look at:
  - Euclidean distance
  - Jaccard index
  - Tanimoto coefficient
  - Cosine similarity
  - Pearson similarity

Euclidean distance.
All attributes are numeric

$$d(A,B)=\sqrt{|A_X-B_X|^2+|A_Y-B_Y|^2}$$



For N dimensions:

$$d(A,B)=\sqrt{\sum_{i=1}^{N}|A_i-B_i|^2}$$

Similarity:

$$s(A,B)=1/(1+d(A,B))$$

It is hard to visualize points in more than 3 dimensions, but for computer it is not a problem

Matching coefficients. All attributes are binary

	Y	Y
X	M_{11}	M_{10}
X	M_{01}	M_{00}

M_{11}: number of attributes with value 1 in both X and Y
M_{10}: number of attributes with value 1 in X and 0 in Y
M_{01}: number of attributes with value 0 in X but 1 in Y
M_{00}: number of attributes with value 0 in both X and Y

Matching coefficients and Jaccard index

	Y	Y
X	M_{11}	M_{10}
X	M_{01}	M_{00}

Jaccard index is used for asymmetric binary attributes, where only value 1 is important

Simple Matching Coefficient

SMC = number of matches / number of all attributes (dimensions)

$= (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$

Jaccard Index

J = number of M_{11} matches / number of not-both-zero attributes values

$= (M_{11}) / (M_{01} + M_{10} + M_{11})$

SMC and Jaccard example

x=(	1	0	0	0	0	0	0	0	0	0	)
y=(	0	0	0	0	0	0	1	0	0	1	)

$SMC = (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00}) = (0+7)/10=0.7$

$J = M_{11} / (M_{01} + M_{10} + M_{11}) = (0)/3=0.0$

The choice is application-dependent

SMC and Jaccard example

x=(	1	0	0	0	0	0	0	0	0	0	)
y=(	0	0	0	0	0	0	1	0	0	1	)

$$SMC= (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00}) = (0+7)/10=0.7$$

$$J= M_{11} / (M_{01} + M_{10} + M_{11}) = (0)/3=0.0$$

The choice is application-dependent

Which measure to choose for:

- Comparing documents by common words?
- Comparing transactions by common items?
- Comparing students by knowledge of 10 topics?

Tanimoto similarity coefficient

- Jaccard index is defined as the number of attributes with value 1 in both records, divided by the total number of records for which there is at least one 1 value:

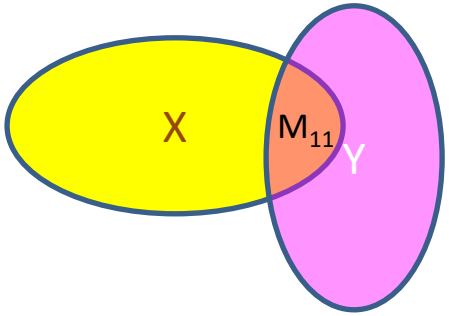
$J = M_{11} / (M_{01} + M_{10} + M_{11})$

- Tanimoto coefficient is similar but is defined in terms of set operations: it is an intersection over union of all attribute values without attributes for which both binary values are False(0):

$T = M_{11} / (M_{-1} + M_{1-} - M_{11})$

The formulas show that Jaccard and Tanimoto are **exactly the same**!

	Y	Y
X	M_{11}	M_{10}
X	M_{01}	M_{00}



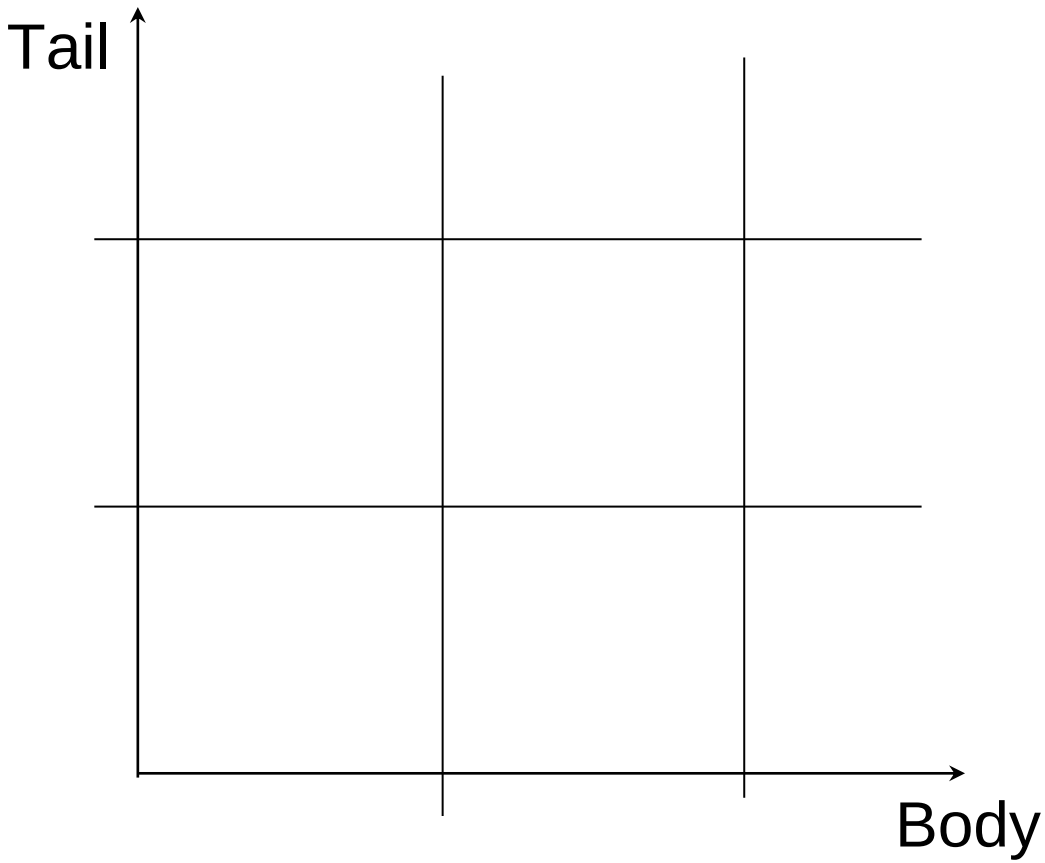
M_{11} ← intersection

$(M_{01} + M_{10} + M_{11}) = (M_{-1} + M_{1-} - M_{11})$ ← union

Cosine similarity

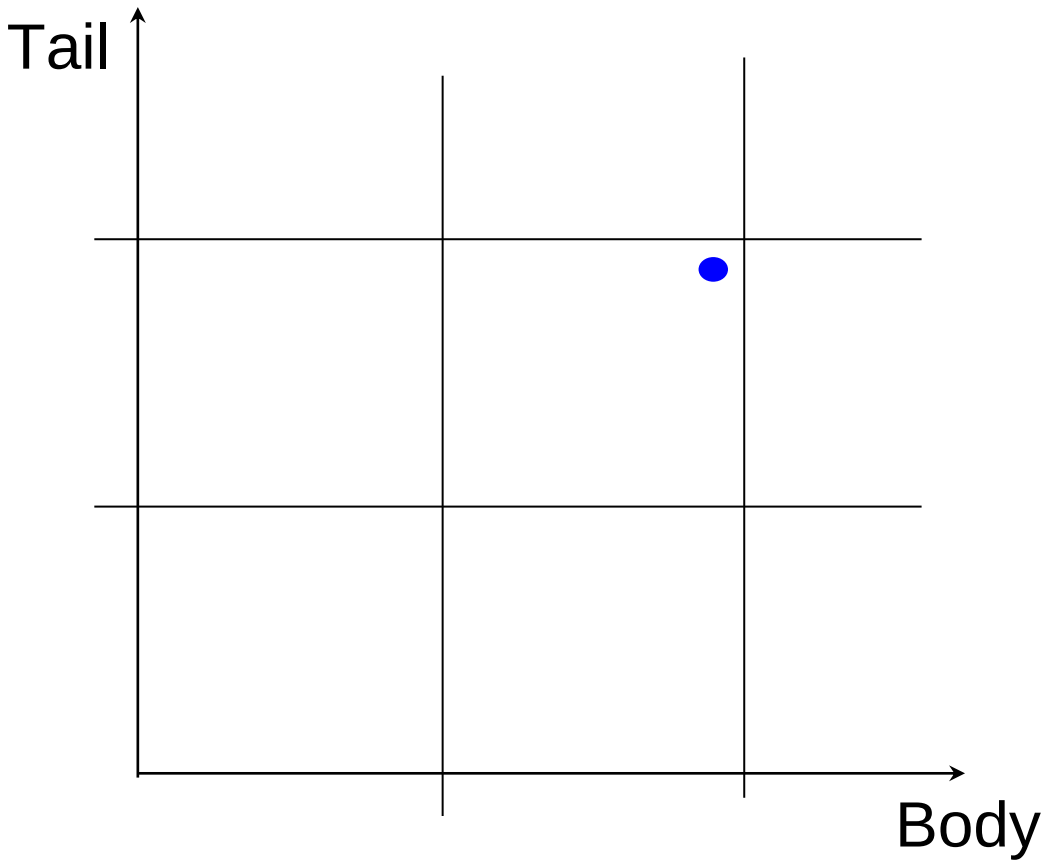
- Sometimes it makes more sense to consider two records closely associated because of similarities in the way the attributes *within each record are related*

Cat or bear classifier



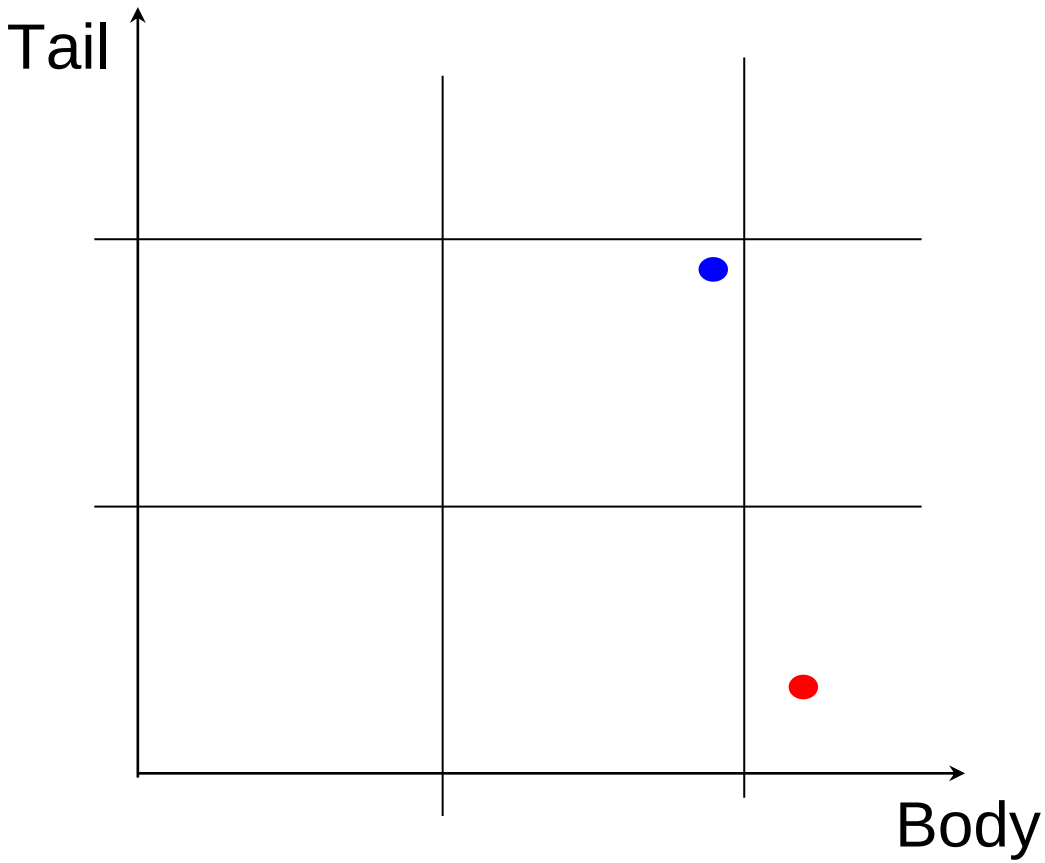
Cat or bear?

Cat or bear classifier

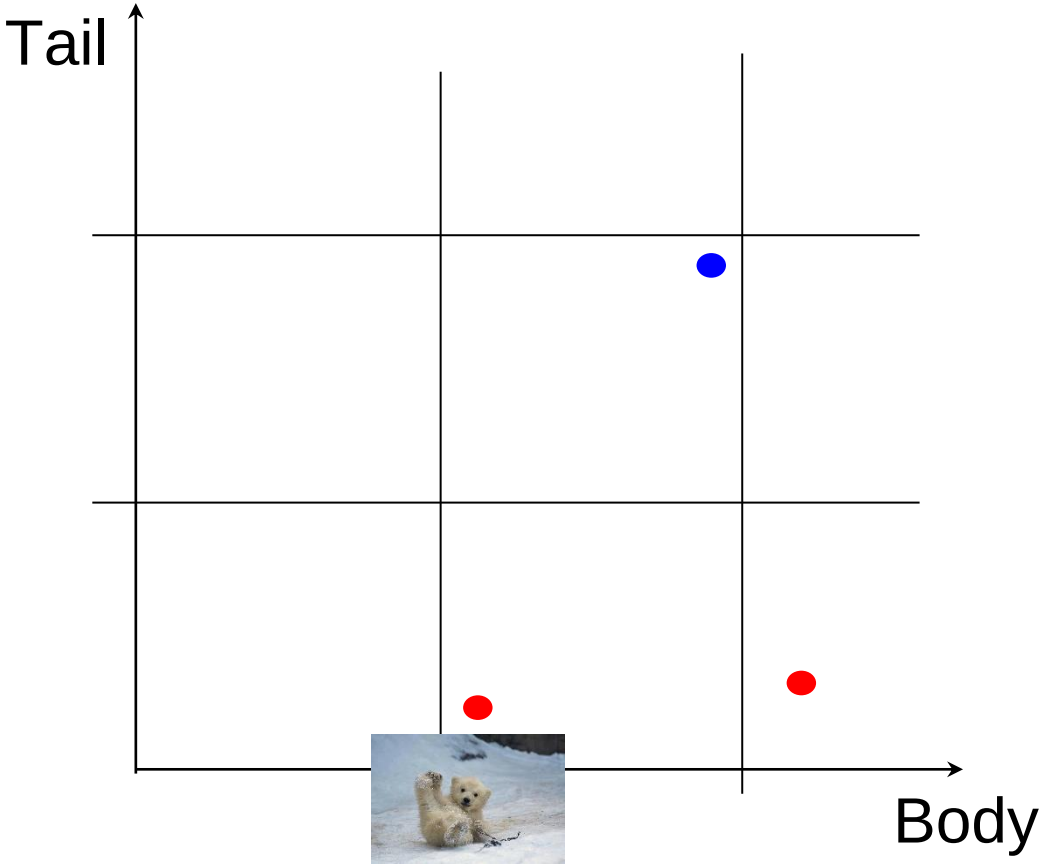


Cat or bear?

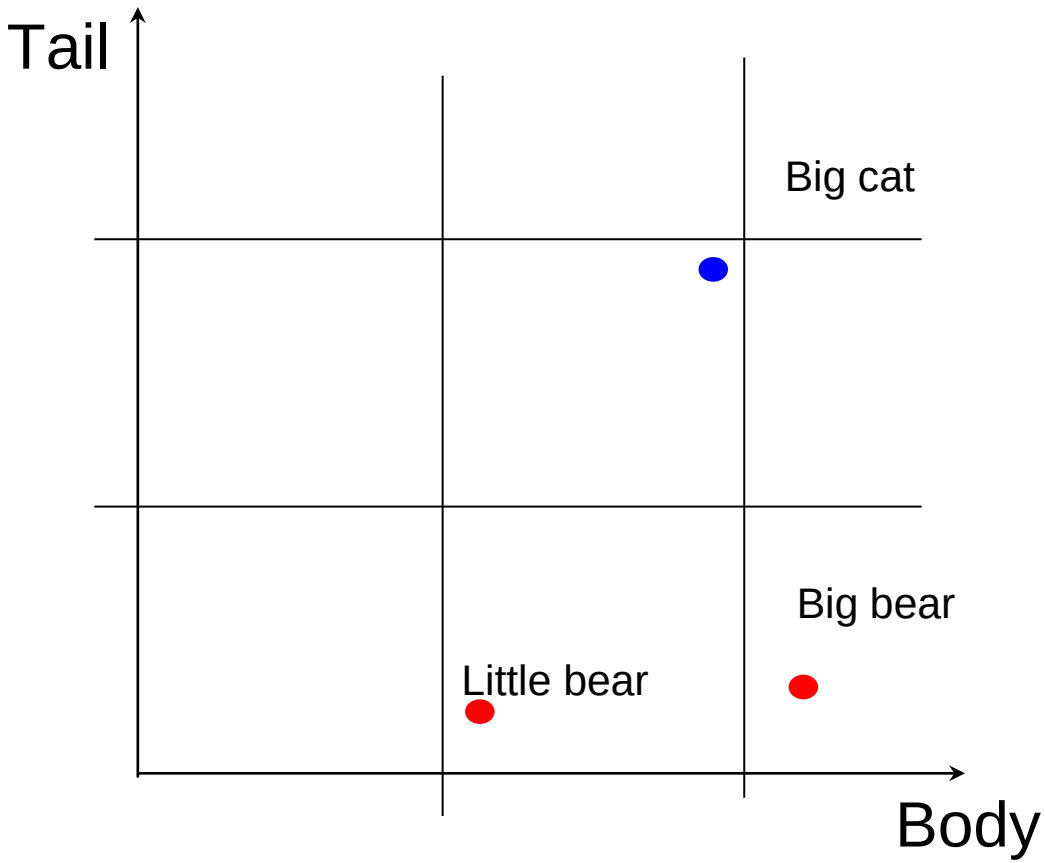
Cat or bear classifier



Cat or bear?

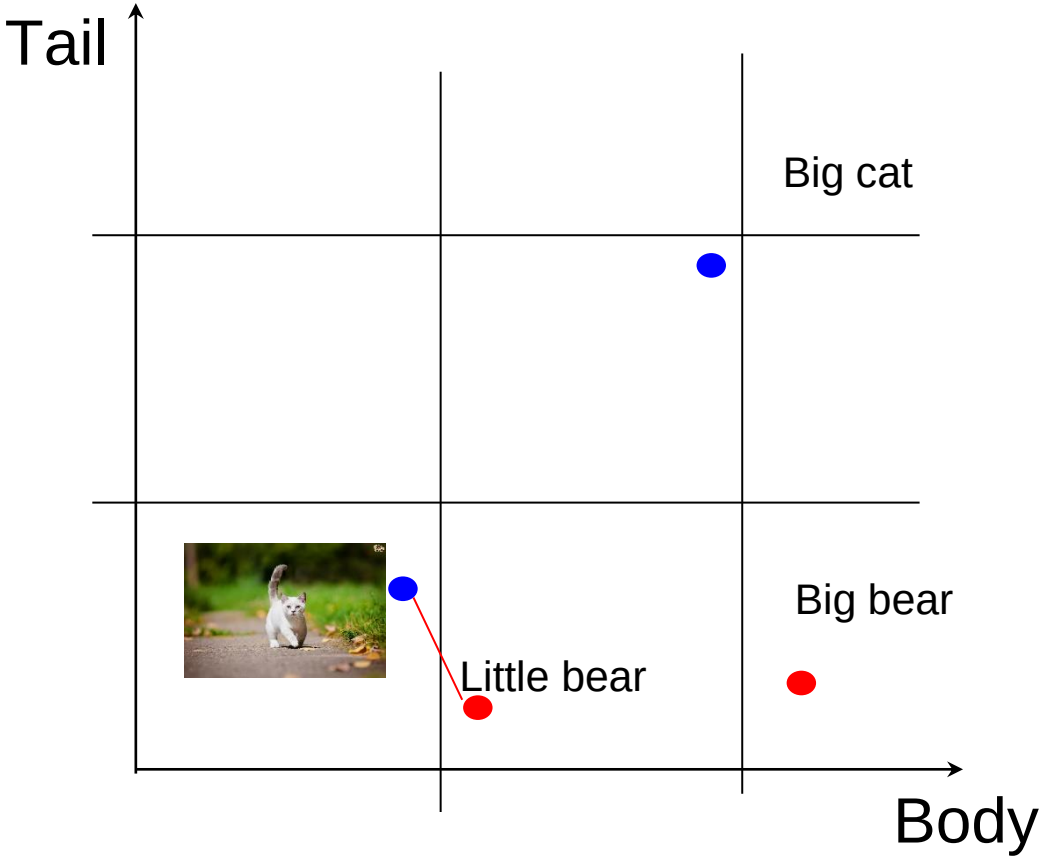


Cat or bear classifier

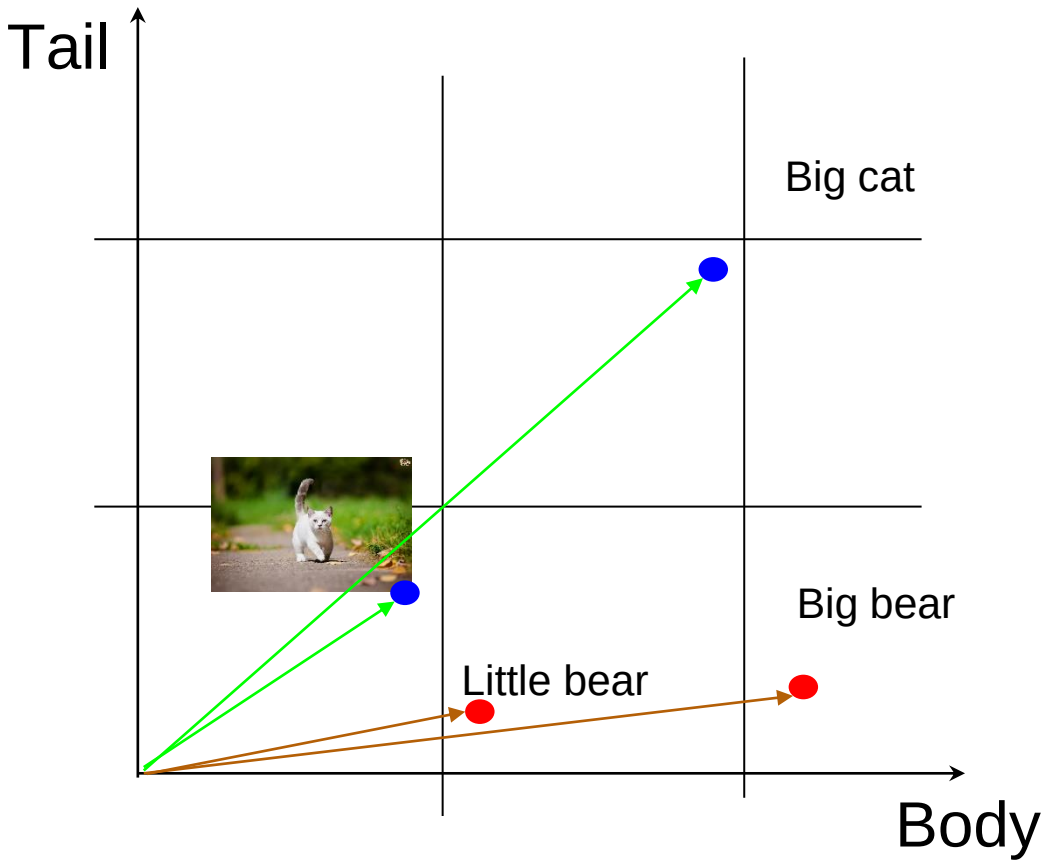


Cat or bear?

Cat or bear?



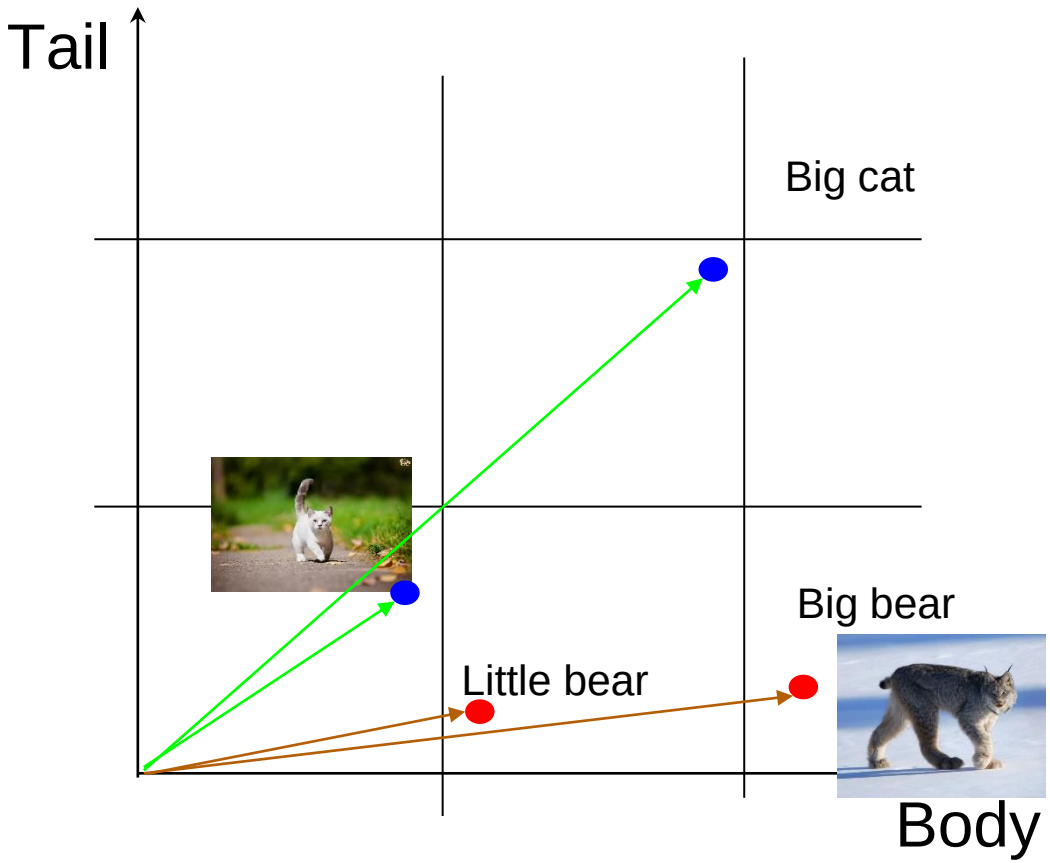
Cat or bear?
Consider angle between vectors



Cat or bear?

Canadian Lynx

Cat or bear?
Consider angle between vectors

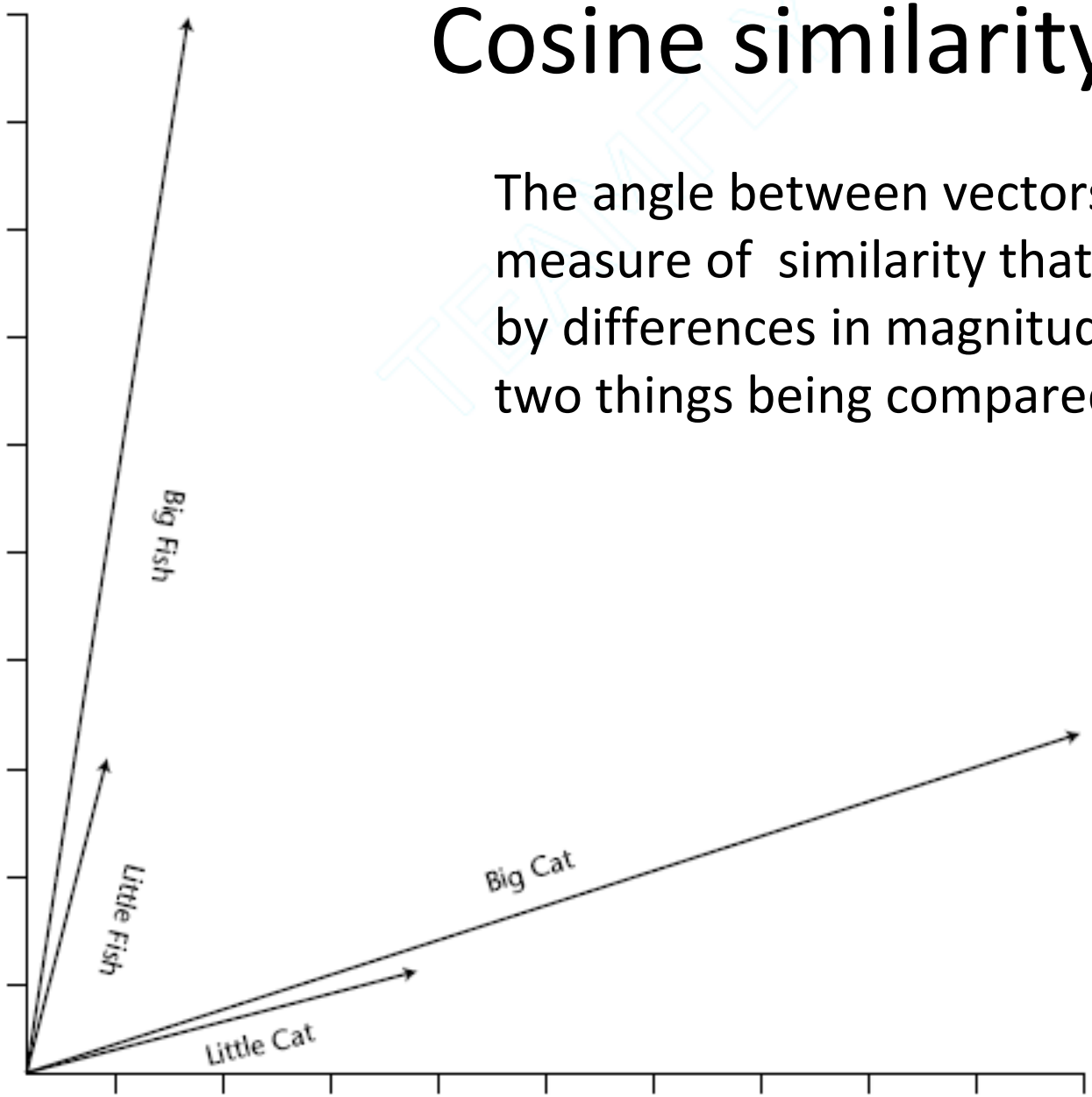


Cosine similarity

- Sometimes it makes more sense to compare records based on the way the fields ***within each record are related***
- Sardines should be closer to cod and tuna, while kittens closer to cougars and lions, but if we use the Euclidean distance of body-part lengths, the sardine is closer to a kitten than it is to a catfish
- Solution: use a different geometric interpretation. Instead of thinking of *X and Y as points in space*, think of them as *vectors and measure the angle between them*
- In this context, a vector is the line segment connecting the origin of a coordinate system to the point described by the vector values

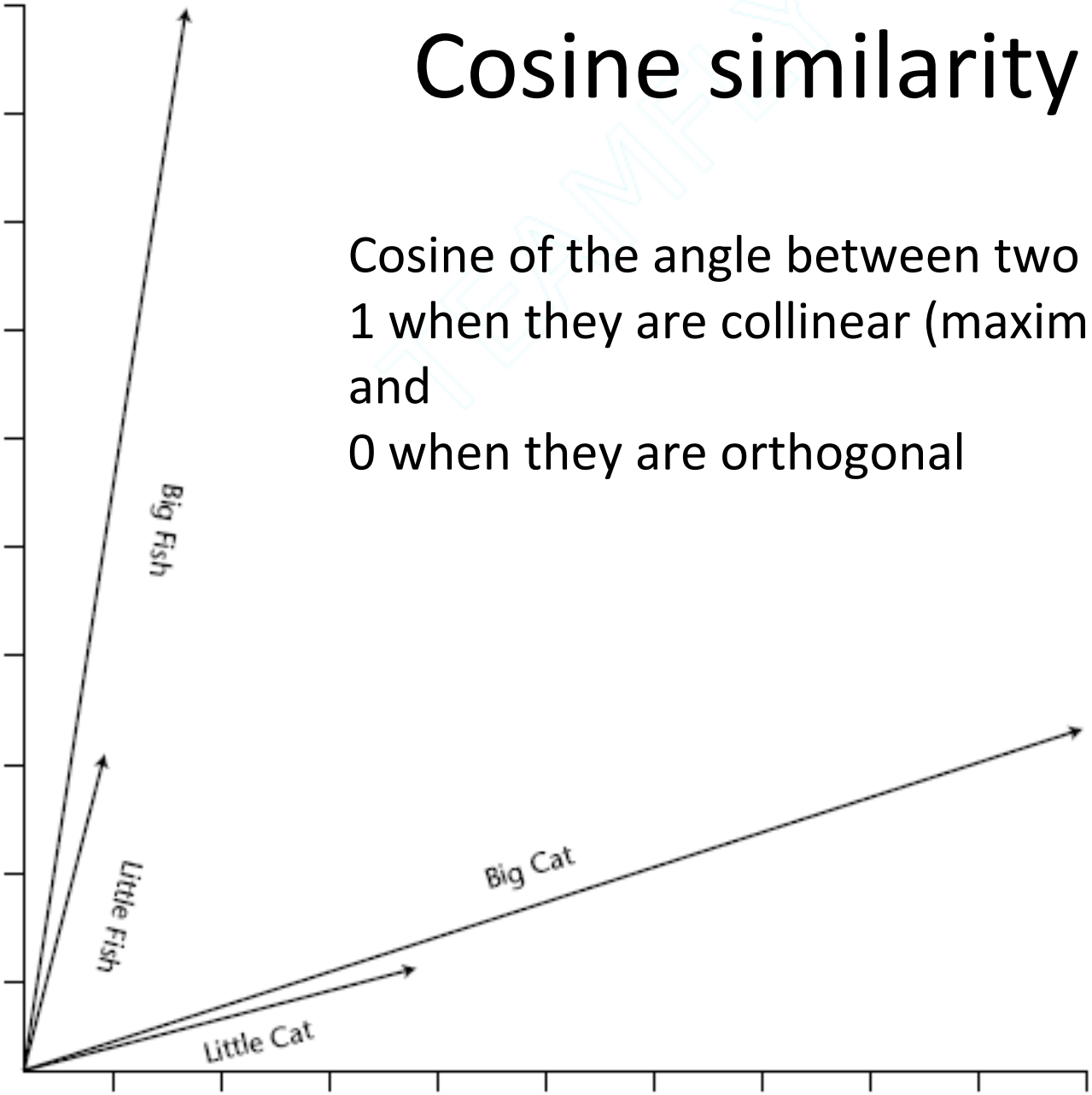
Cosine similarity

The angle between vectors provides a measure of similarity that is not influenced by differences in magnitude between the two things being compared



Cosine similarity

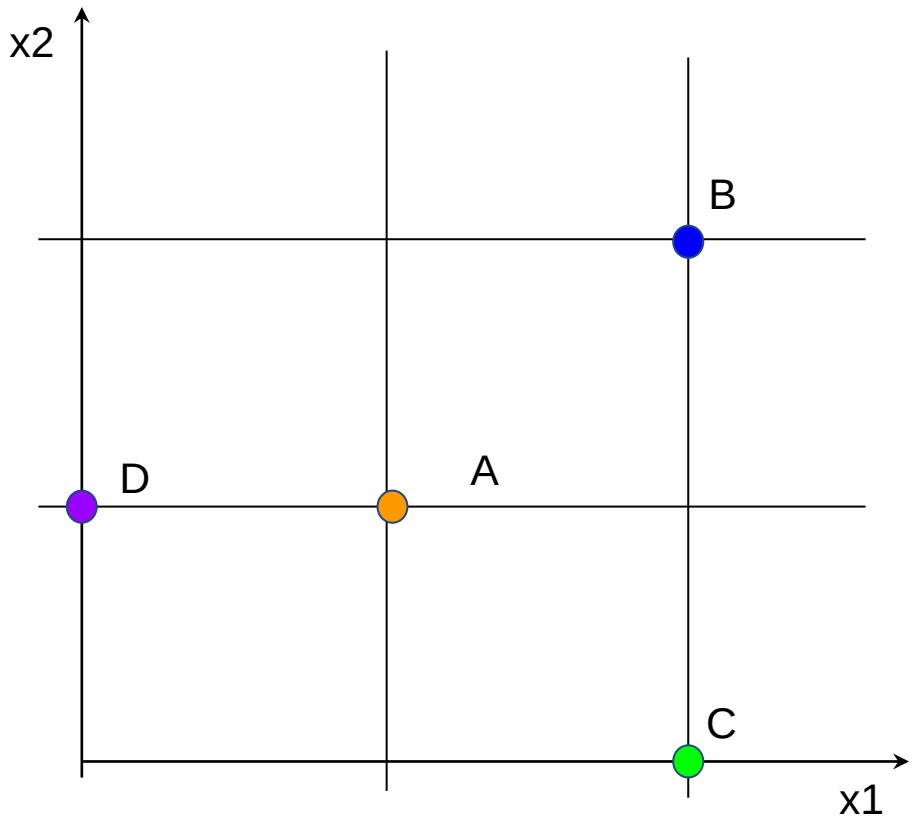
Cosine of the angle between two vectors is
1 when they are collinear (maximum similarity)
and
0 when they are orthogonal



Cosine similarity

$$s(\mathrm{A},\mathrm{B}) = cos(\mathbf{A}, \mathbf{B}) = (\mathbf{A} \cdot \mathbf{B}) / \|\mathbf{A}\| \cdot \|\mathbf{B}\|$$

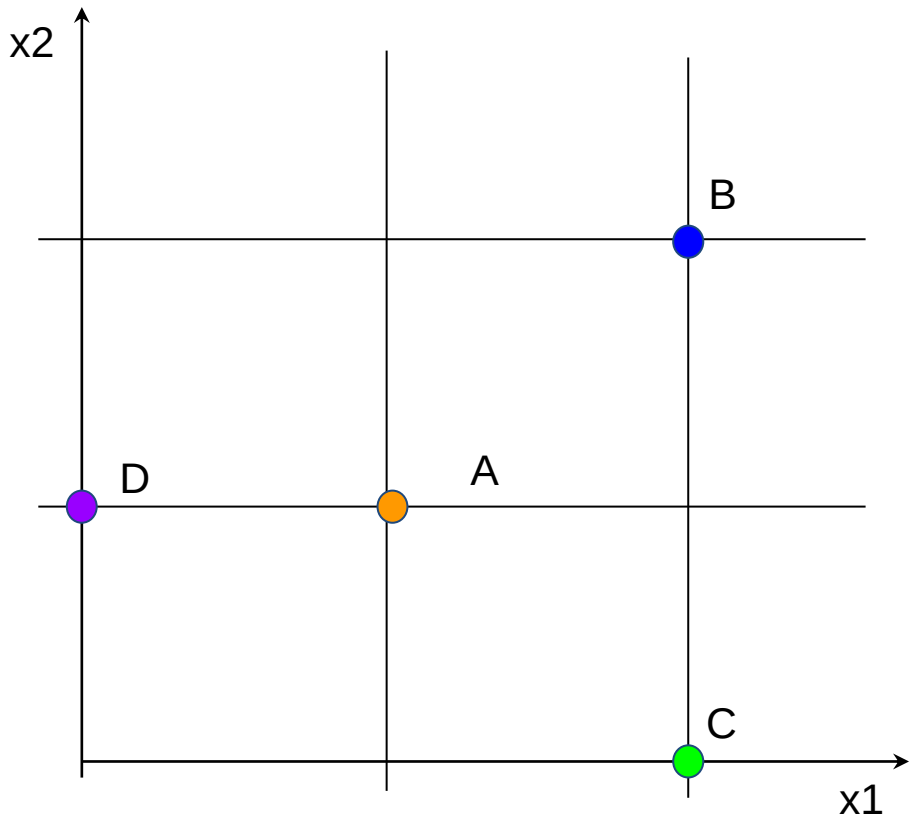
Dot-product of vectors



Cosine similarity

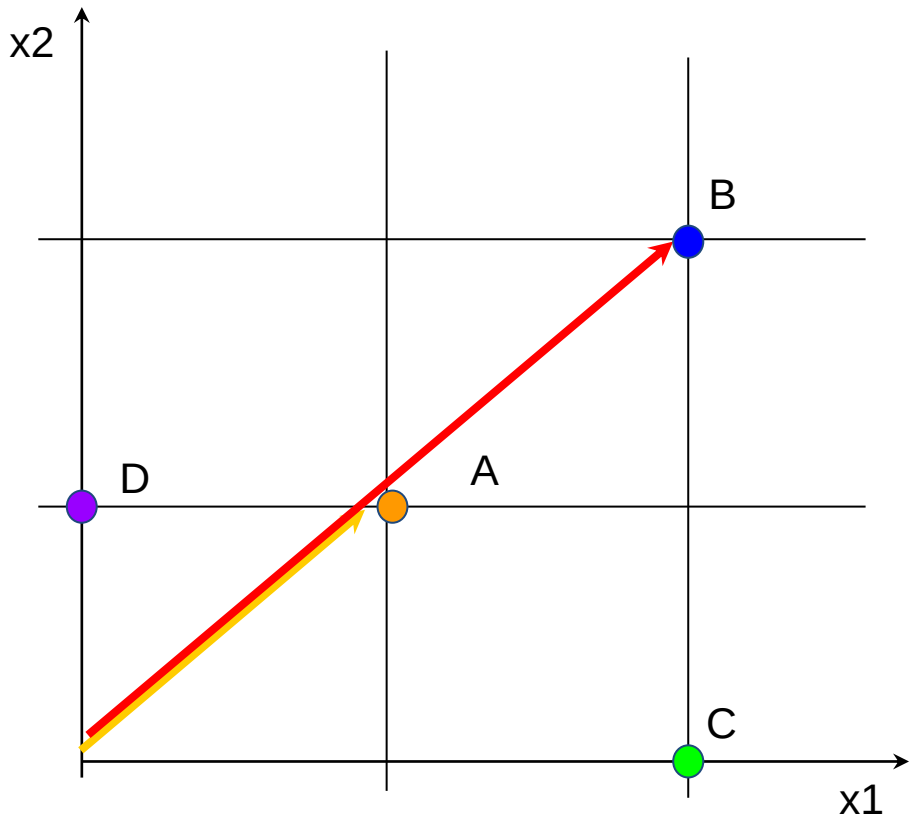
$$s(\mathrm{A},\mathrm{B}) = cos(\mathbf{A}, \mathbf{B}) = (\mathbf{A} \cdot \mathbf{B}) / \|\mathbf{A}\| \cdot \|\mathbf{B}\|$$

Absolute length of vectors **A** and **B**



Cosine similarity

$$s(A,B)=cos(\mathbf{A}, \mathbf{B}) = (\mathbf{A} \cdot \mathbf{B}) / \|\mathbf{A}\| \cdot \|\mathbf{B}\|$$



$\mathbf{A}=(1,1)$

$\mathbf{B}=(2,2)$

$\mathbf{A} \cdot \mathbf{B}=1*2+1*2=4$

$\|\mathbf{A}\|=\text{sqrt}(1+1)$

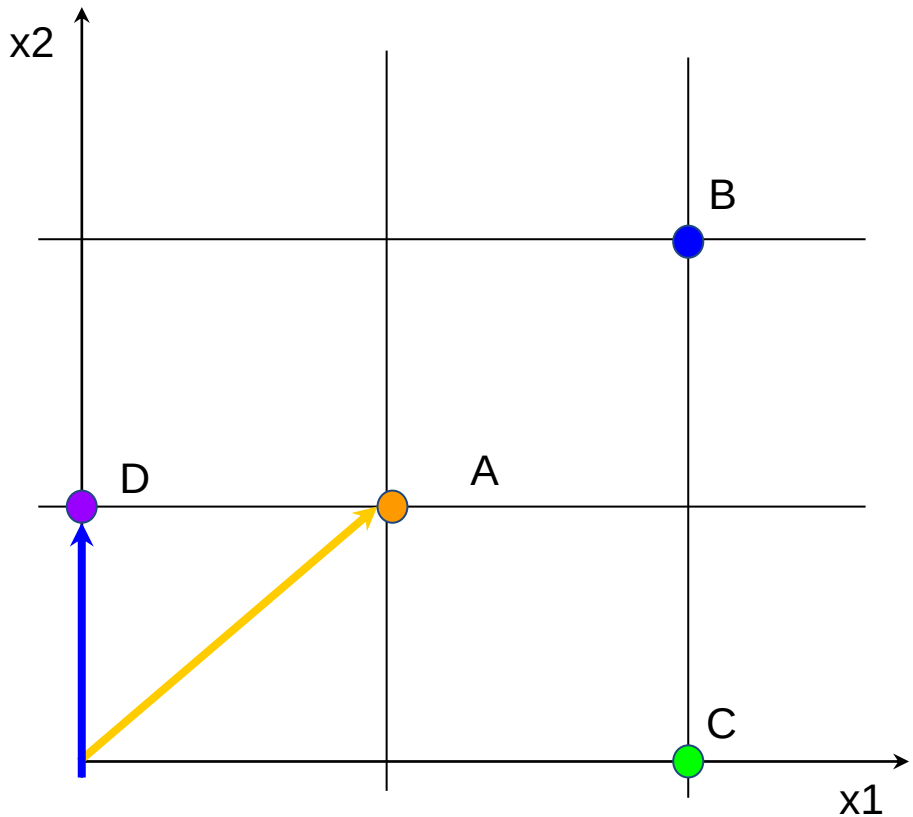
$\|\mathbf{B}\|=\text{sqrt}(4+4)$

$\|\mathbf{A}\|.\|\mathbf{B}\|=\text{sqrt}(16)$

$$s(A,B)=cos(\mathbf{A}, \mathbf{B})=1$$

Cosine similarity

$$s(\mathrm{A},\mathrm{D}) = cos(\mathbf{A}, \mathbf{D}) = (\mathbf{A} \cdot \mathbf{D}) / \|\mathbf{A}\| \cdot \|\mathbf{D}\|$$



$\mathbf{A}=(1,1)$

$\mathbf{D}=(0,1)$

$\mathbf{A} \cdot \mathbf{D}=0+1=1$

$\|\mathbf{A}\|=\mathrm{sqrt}(2)$

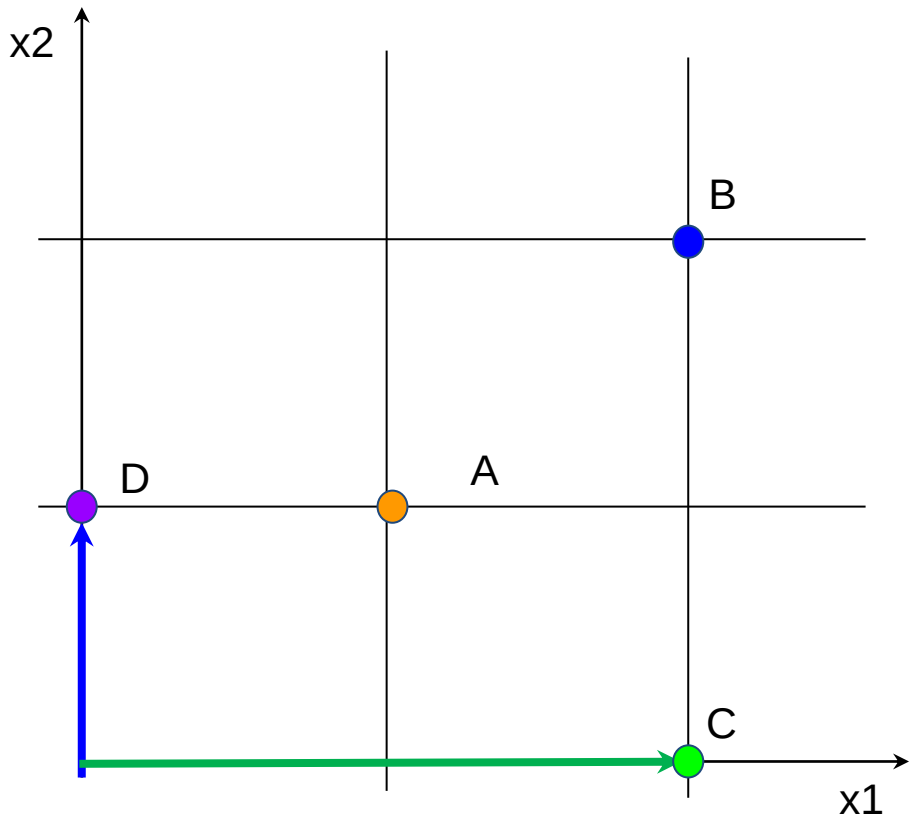
$\|\mathbf{D}\|=1$

$\|\mathbf{A}\|.\|\mathbf{D}\|=\mathrm{sqrt}(2)$

$$s(\mathrm{A},\mathrm{D}) = cos(\mathbf{A}, \mathbf{D}) = \mathrm{sqrt}(1/2) \approx 0.7$$

Cosine similarity

$$s(\mathrm{C},\mathrm{D}) = cos(\mathbf{C}, \mathbf{D}) = (\mathbf{C} \cdot \mathbf{D}) / \|\mathbf{C}\| \cdot \|\mathbf{D}\|$$



$\mathbf{C}=(2,0)$

$\mathbf{D}=(0,1)$

$\mathbf{C} \cdot \mathbf{D}=0$

$s(\mathrm{C},\mathrm{D}) = cos(\mathbf{C}, \mathbf{D}) = 0$

Cosine Similarity for document vectors

	w1	w2	w3	w4	w5	w6	
x=(	1	0	0	0	0	0	)
y=(	0	0	0	1	2	0	)
z=(	0	0	0	4	8	0	)

Cosine between **x** and **y** is 0 (dot-product is 0). These documents are not similar.

Cosine between **y** and **z** is 1: though the number of times each word occurs in y and z is different, these documents are about the same topic

Pearson correlation

- A ***correlation*** is a number between -1 and +1 that measures the degree of association between two variables (in our case – between 2 data objects for which we recorded *n* observations)
- A positive value for the correlation implies a positive association (the values across all observations vary in the same direction)
- A negative value for the correlation implies a negative or inverse association – which makes 2 data objects dissimilar
- A value close to 0 implies that there is no correlation between two data objects

Pearson correlation formula

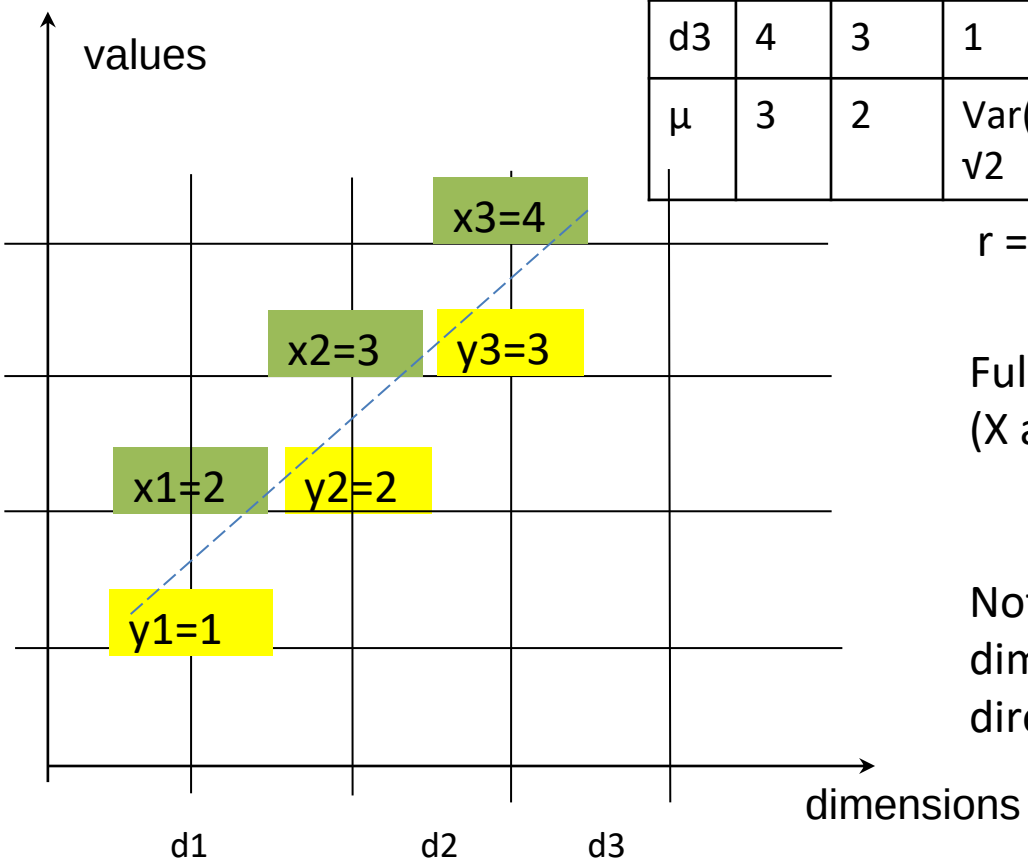
$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- In numerator we see a *covariance* - a measure of the joint variability of 2 data objects across *n* observations
- We normalize it by dividing by a *variance* inside each separate data object

Pearson correlation: example 1

Are all (3) observations for objects X and Y change in the same direction?

	X	Y	$x_i - \mu_x$	$y_i - \mu_y$	Cov(X,Y)
d1	2	1	-1	-1	1
d2	3	2	0	0	0
d3	4	3	1	1	1
μ	3	2	Var(x) = √2	Var(y) = √2	2



$r = 2/(\sqrt{2}*\sqrt{2}) = 1$

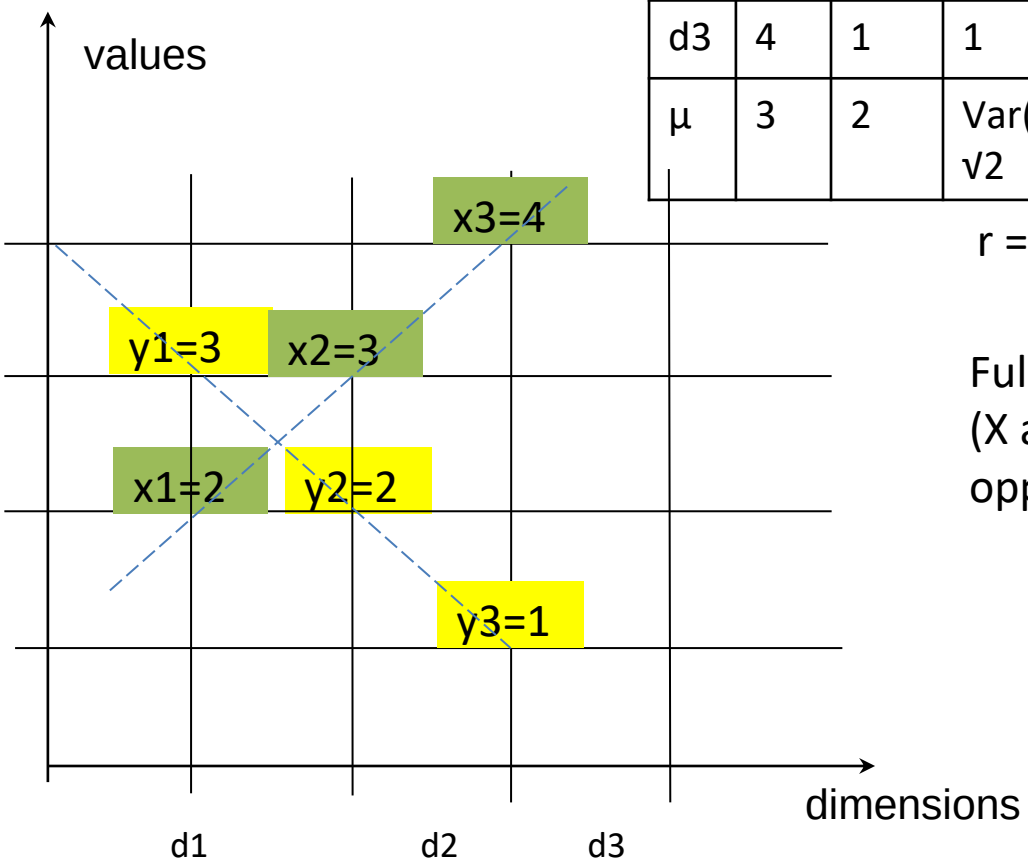
Full positive correlation
(X and Y are very similar)

Note that absolute values across each dimension do not play any role – only direction is important

Pearson correlation: example 2

Are all (3) observations for objects X and Y change in the same direction?

	X	Y	$x_i - \mu_x$	$y_i - \mu_y$	Cov(X,Y)
d1	2	3	-1	1	-1
d2	3	2	0	0	0
d3	4	1	1	-1	-1
μ	3	2	Var(x) = √2	Var(y) = √2	-2



$r = -2/(\sqrt{2}*\sqrt{2}) = -1$

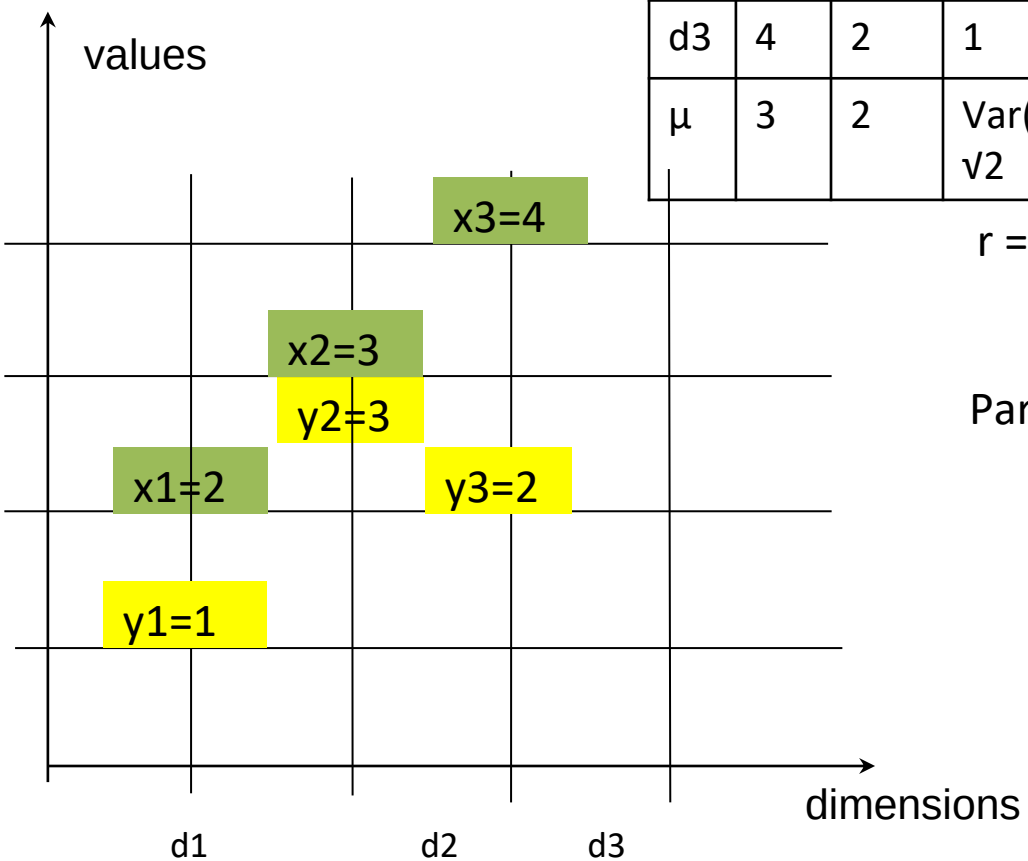
Full negative correlation
(X and Y are least similar – quite opposite)

Pearson correlation: example 3

Are all (3) observations for objects X and Y change in the same direction?

	X	Y	$x_i - \mu_x$	$y_i - \mu_y$	Cov(X,Y)
d1	2	1	-1	-1	1
d2	3	3	0	1	0
d3	4	2	1	0	0
μ	3	2	Var(x) = √2	Var(y) = √2	1

r = 1/(√2*√2) = 1/2



Partly correlated objects (less similar)

Relationship between Pearson correlation and cosine similarity

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

- By subtracting mean from each value, we effectively just transposing vectors to center them around mean
- Pearson correlation is nothing else but a cosine between two vectors after they are centered around the mean for each dimension
- Pearson correlation is *a cosine of centered vectors*

Combining Similarities of different types

- Sometimes attributes are of many different types, but an overall similarity/dissimilarity is needed.
- For each type of attributes *k*, compute a similarity s_k
- Then average,

$$\text{similarity}(p,q) = \frac{\sum_{k=1}^{n} s_k}{n}$$

- Similar formula for dissimilarity (distance)

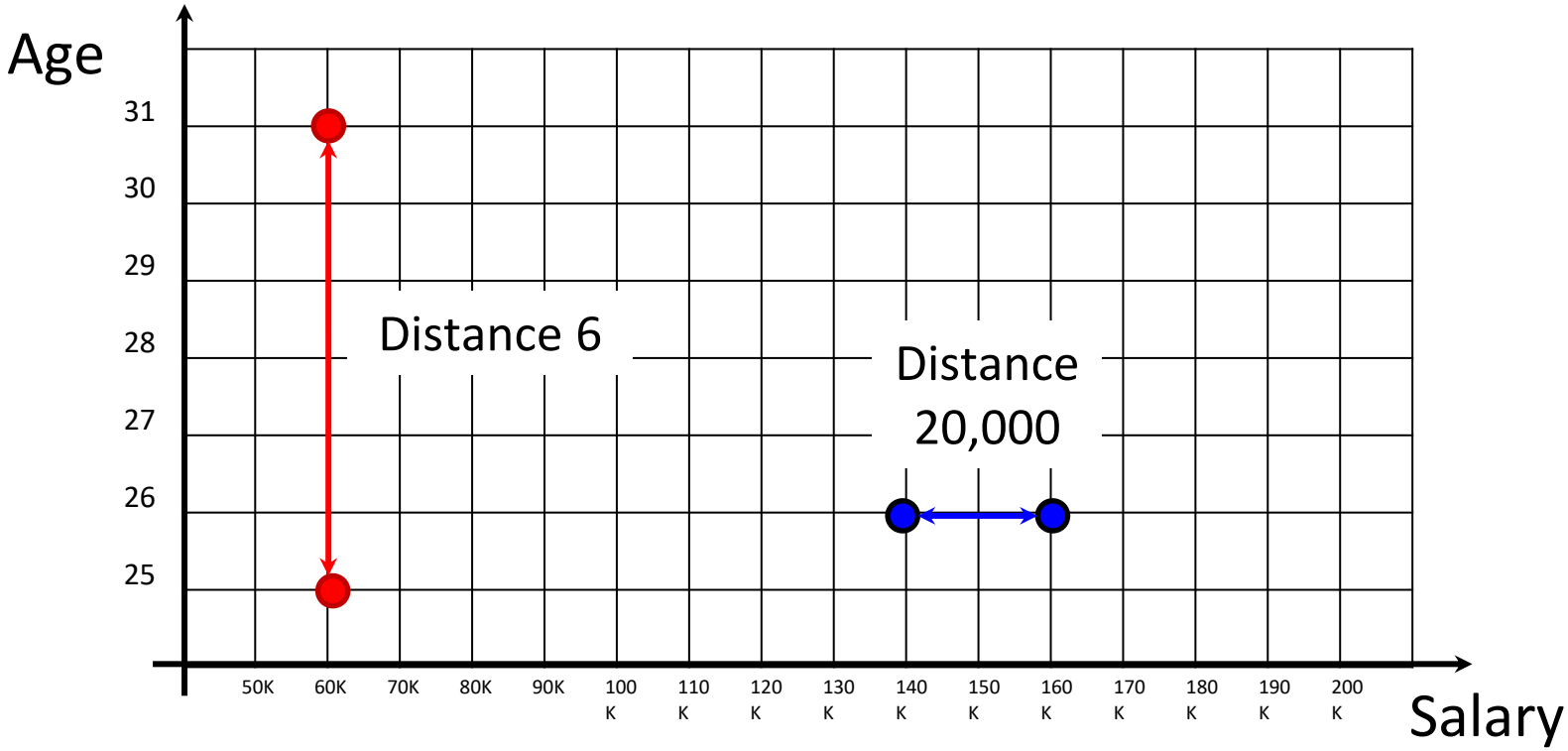
Scaling attributes for consistency

- X- in yards, Y in cm
- X- number of children, Y – income

Difference in 1 dollar = difference in 1 child?

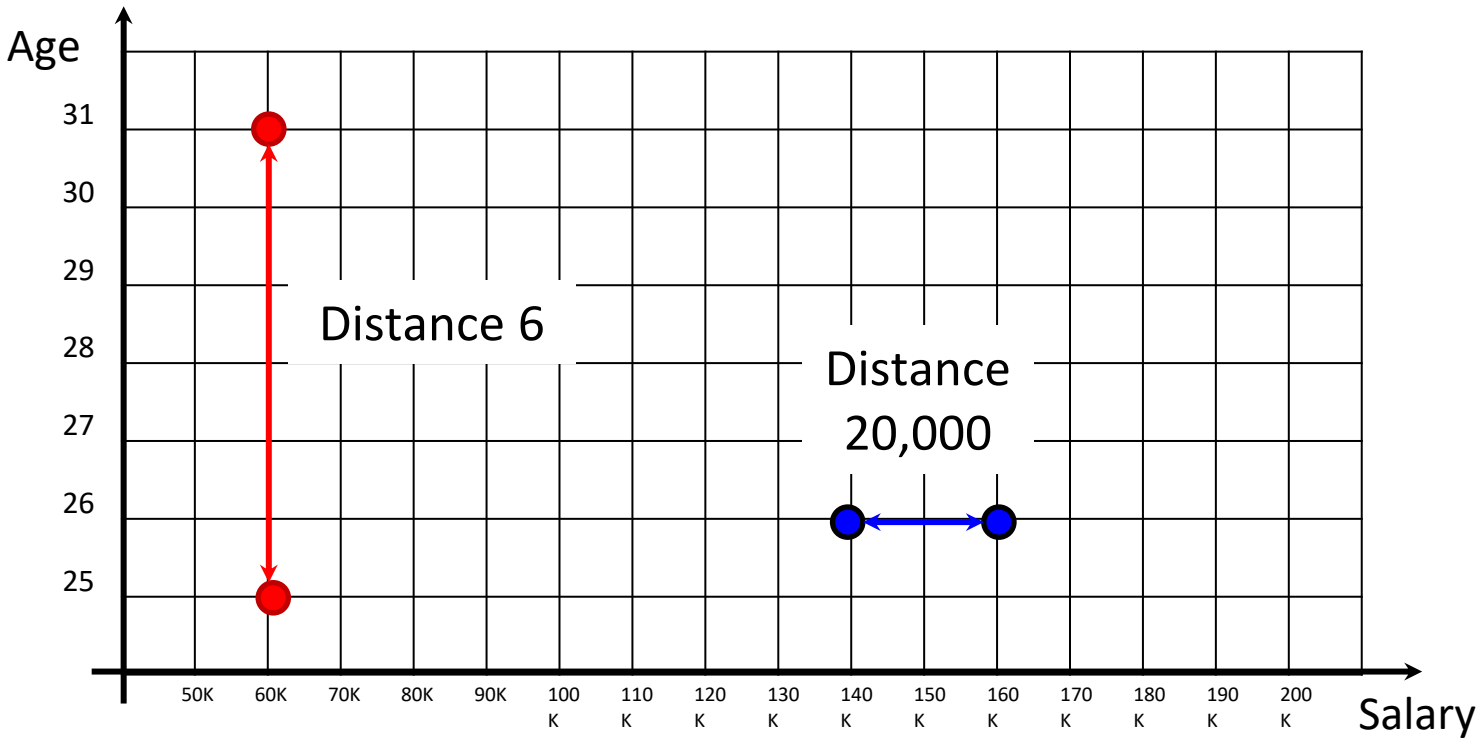
Scaling: map all variables to a common range 0-1

Example: need to scale



Example: scaling

$$a_i = \frac{v_i - \min(all\ v)}{\max(all\ v) - \min(all\ v)}$$

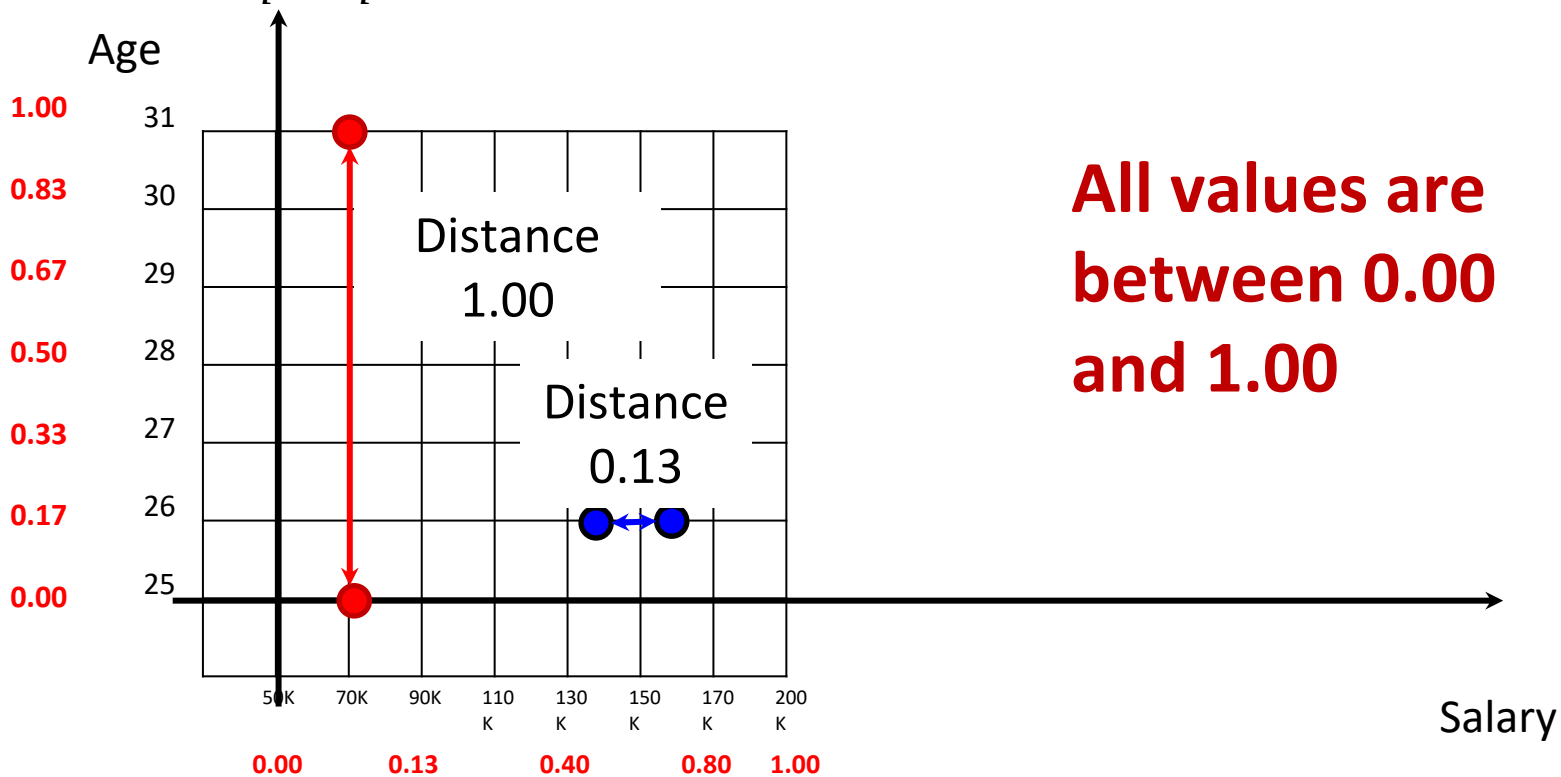


Example: result

$$a_i = \frac{v_i - \min(all\ v)}{\max(all\ v) - \min(all\ v)}$$

For Age: a_i=(v_i-25)/(31-25)

For Salary: a_i=(v_i-50,000)/(200,00-50,00)



All values are between 0.00 and 1.00

Scaling vectors

- Vector normalization – changes the vector values so that the length of the vector is 1, only the direction is compared
- X={Debt=200,000 equity=100,000}
- Y={Debt=2,000 equity=1,000}

Emphasizes internal relation between different attributes of each record

Encode expert knowledge with weights

- Changes in one variable should not be more significant only because of differences in magnitudes of values

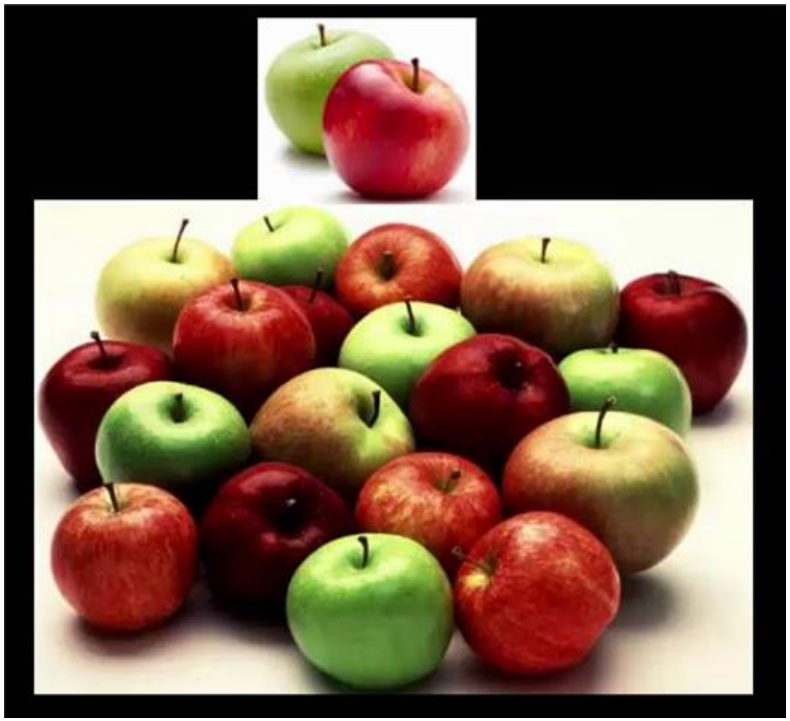
- After scaling to get rid of bias due to units, use weights to introduce bias based on expert knowledge of context:
  - 2 families with the same income and number of children are more similar than 2 families living in the same neighborhood
  - Number of children is more important than the number of credit cards

Data-dependent proximity

Two Apples among Apples

are less similar than

Two Apples among Pears



S. Aryal, K. M. Ting, G. Haffari and T. Washio, "Mp-Dissimilarity: A Data Dependent Dissimilarity Measure," *ICDM*, 2014

K-NN: round 2

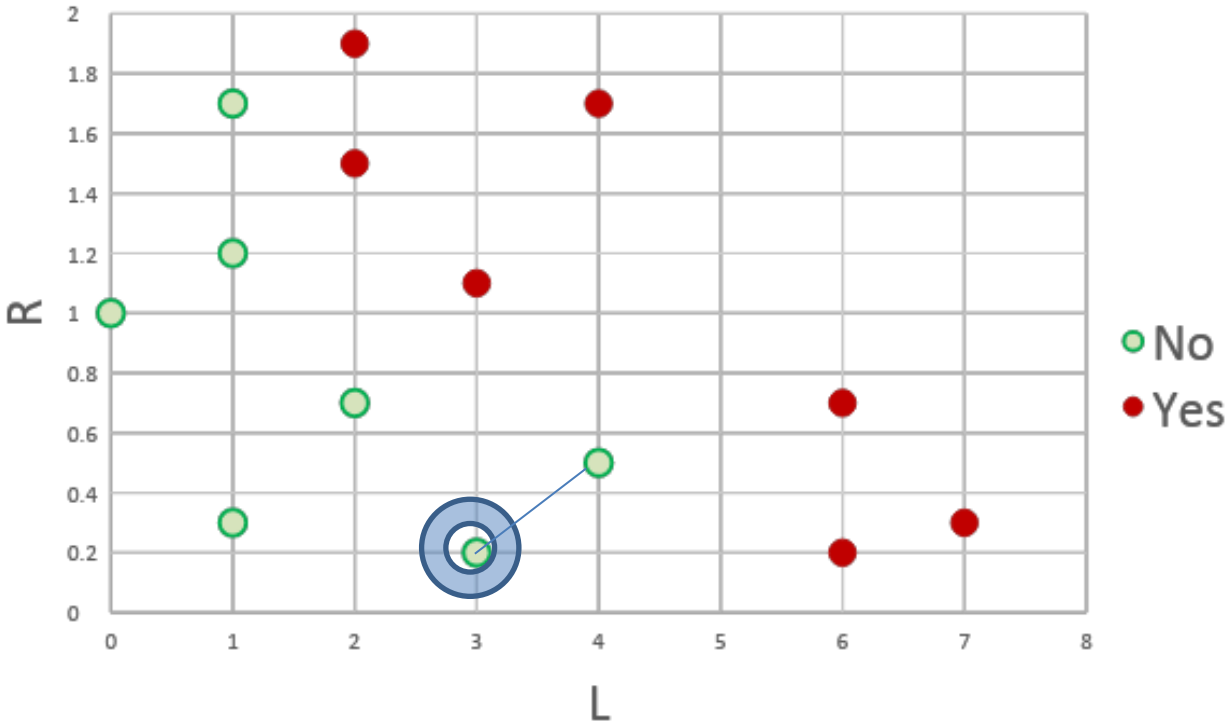
I. Distance/similarity between data records
II. How many neighbors: choice of K
III. Combining neighbor votes
IV. How many features (dimensions)

How many neighbors?
application-dependent

- Vary K from 1 to N
- Use cross-validation to find optimal value of K

Leave-one-out cross validation: *K*=1

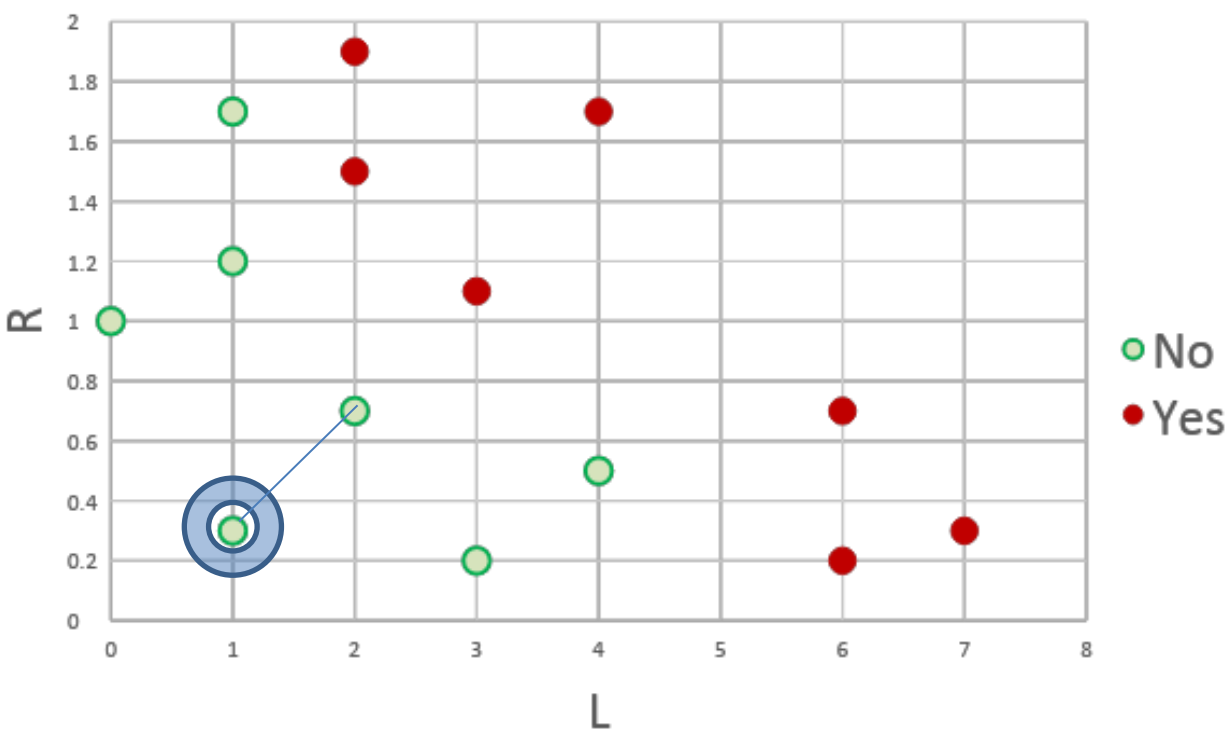
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

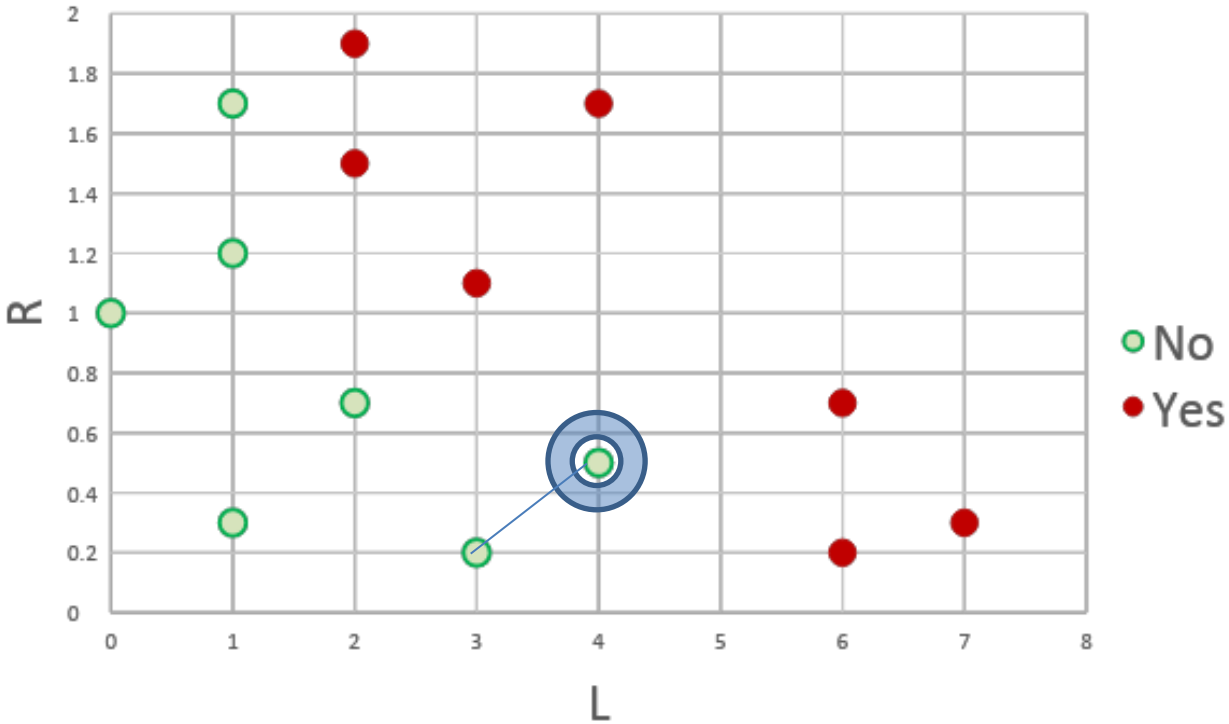
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

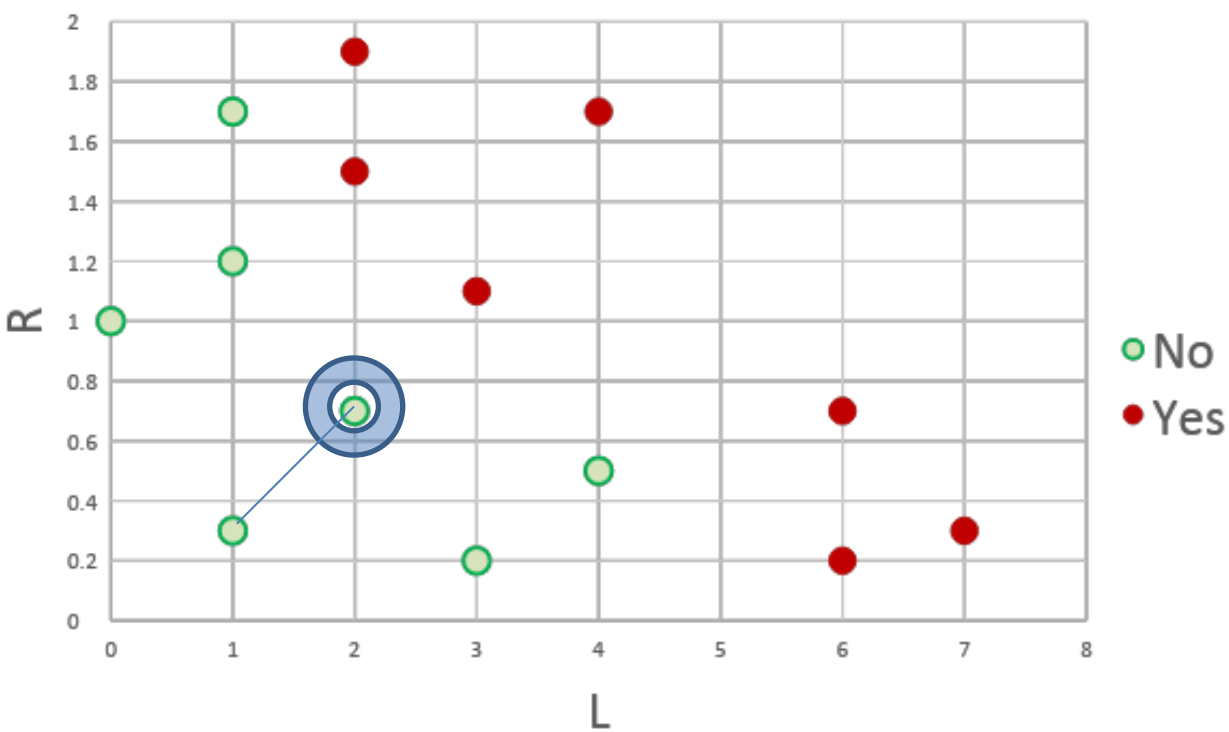
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

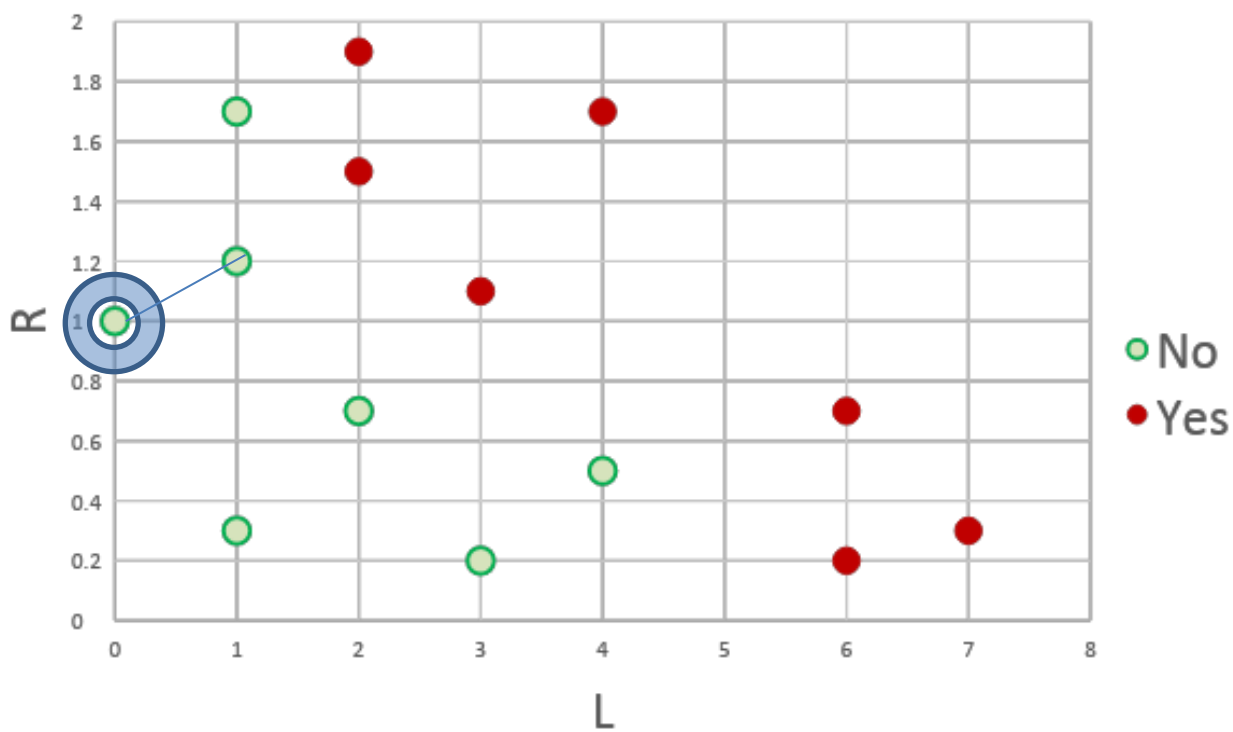
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

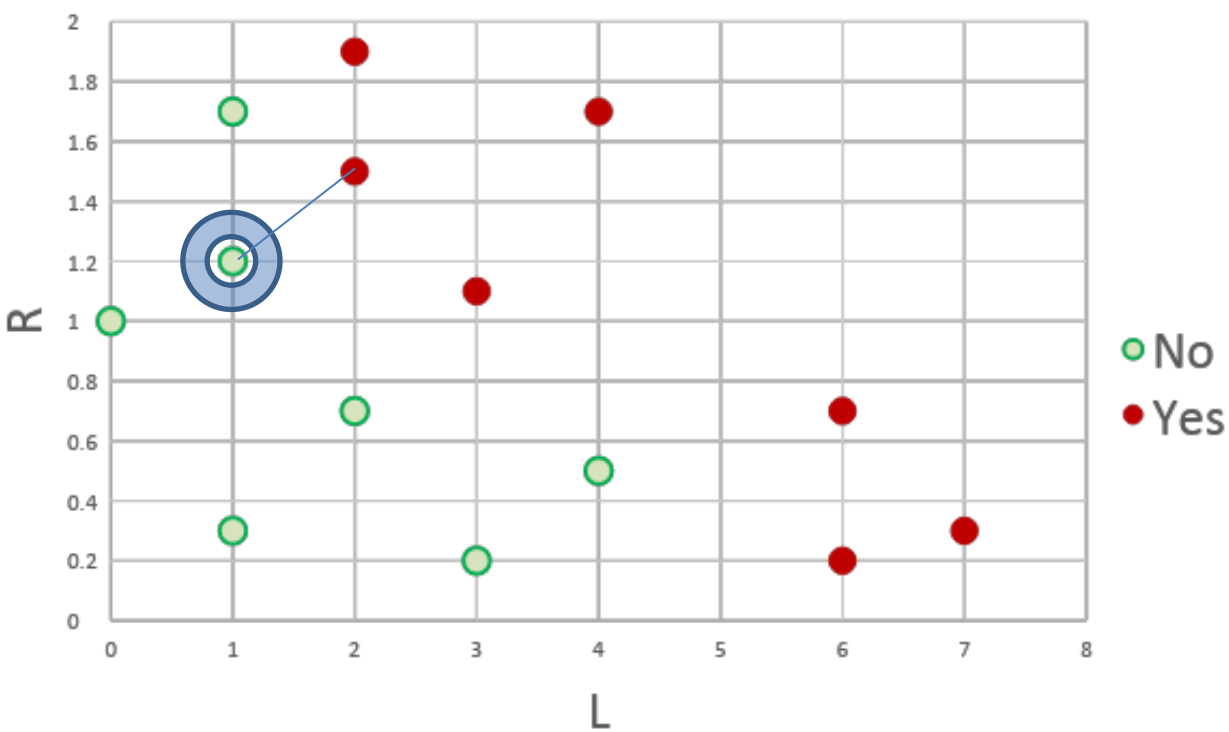
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

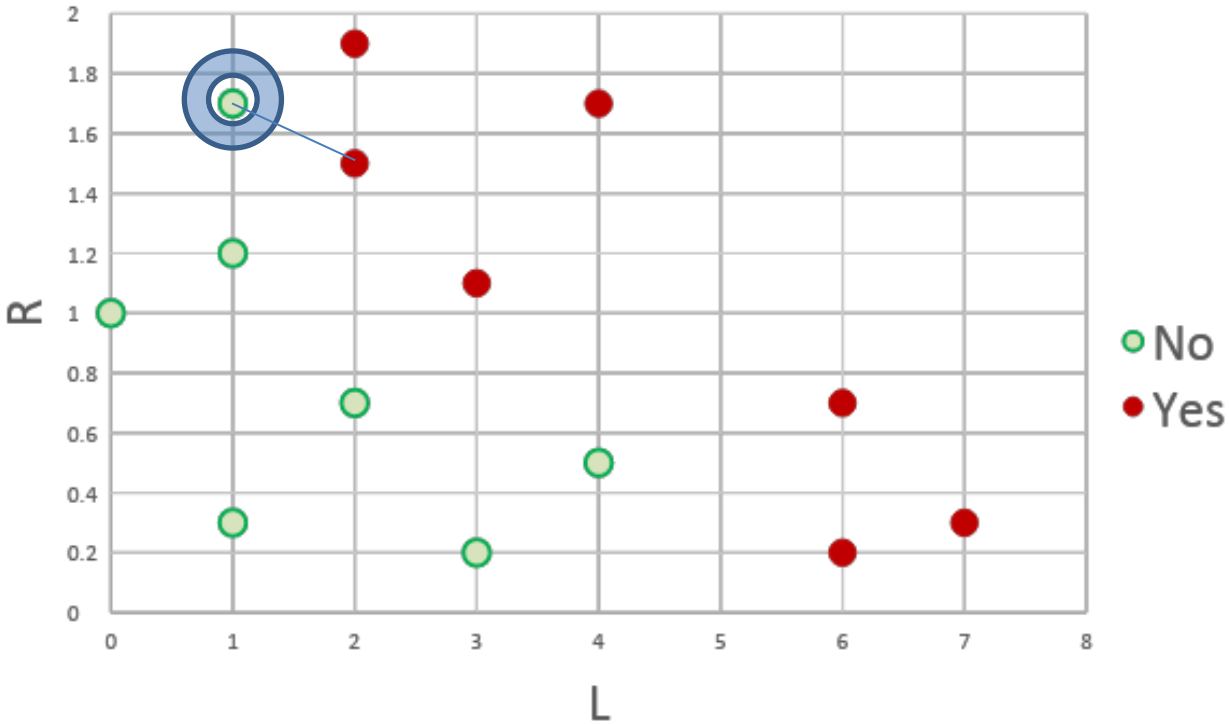
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

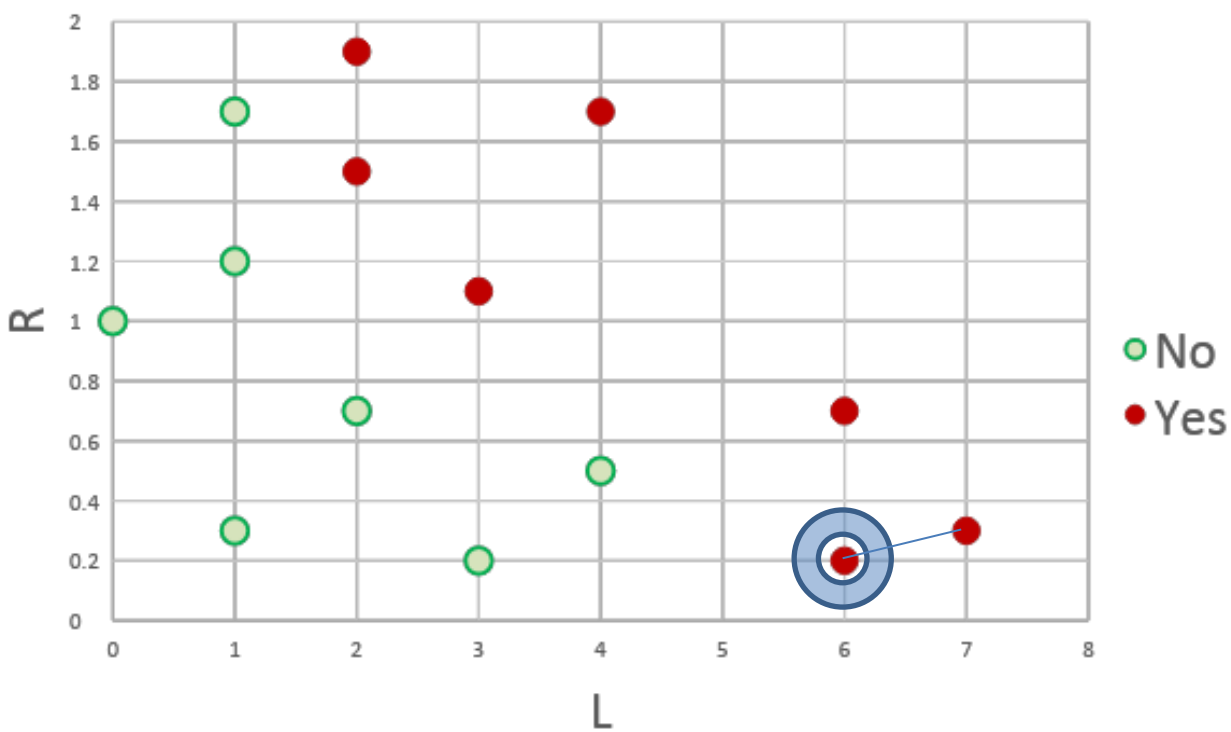
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

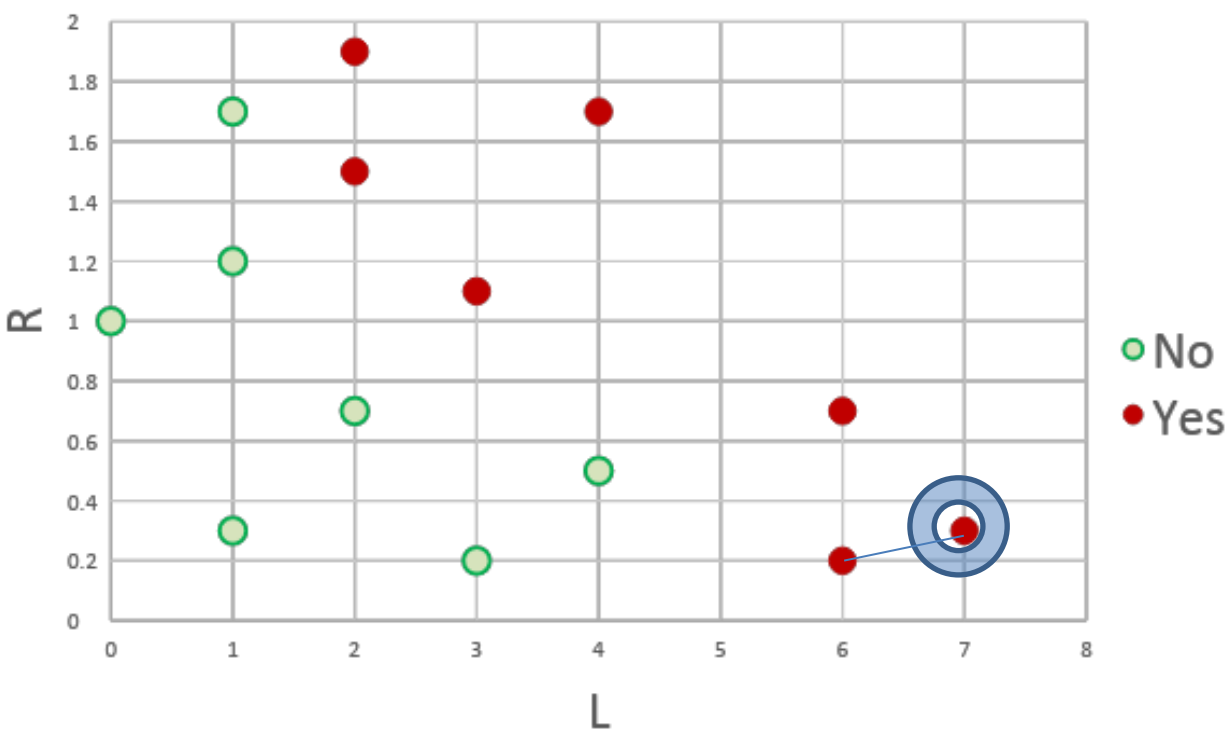
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

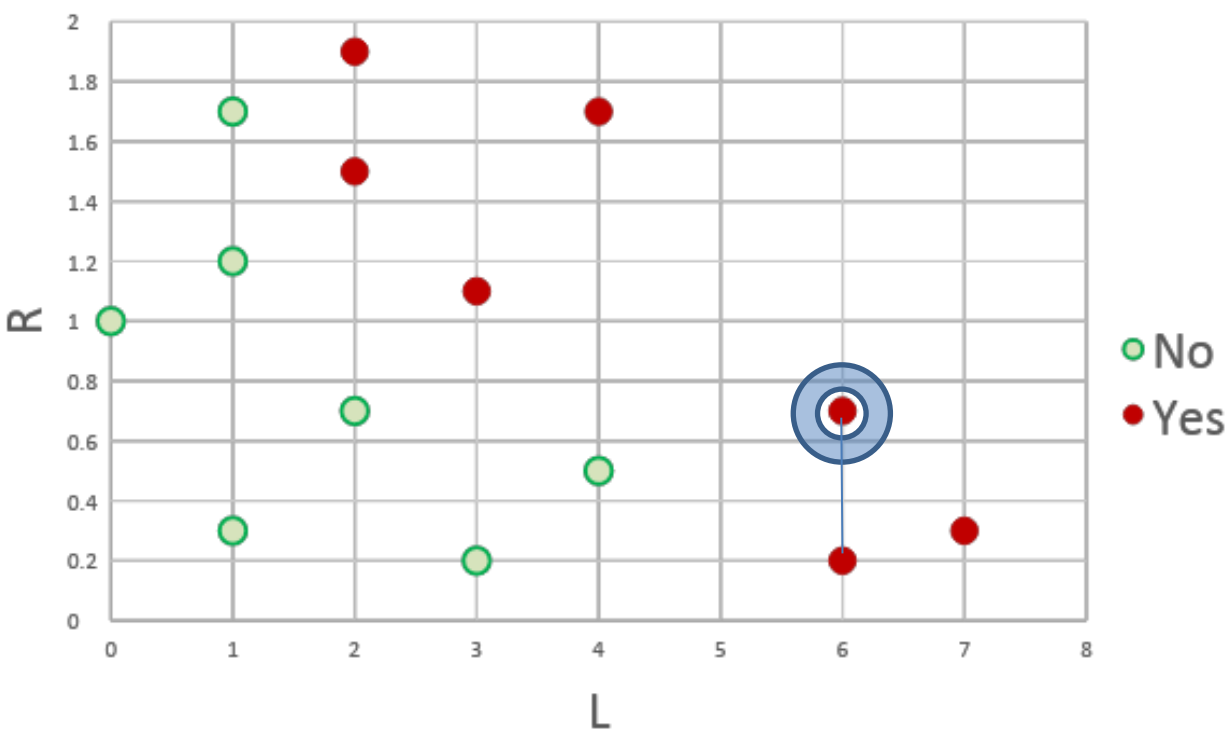
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

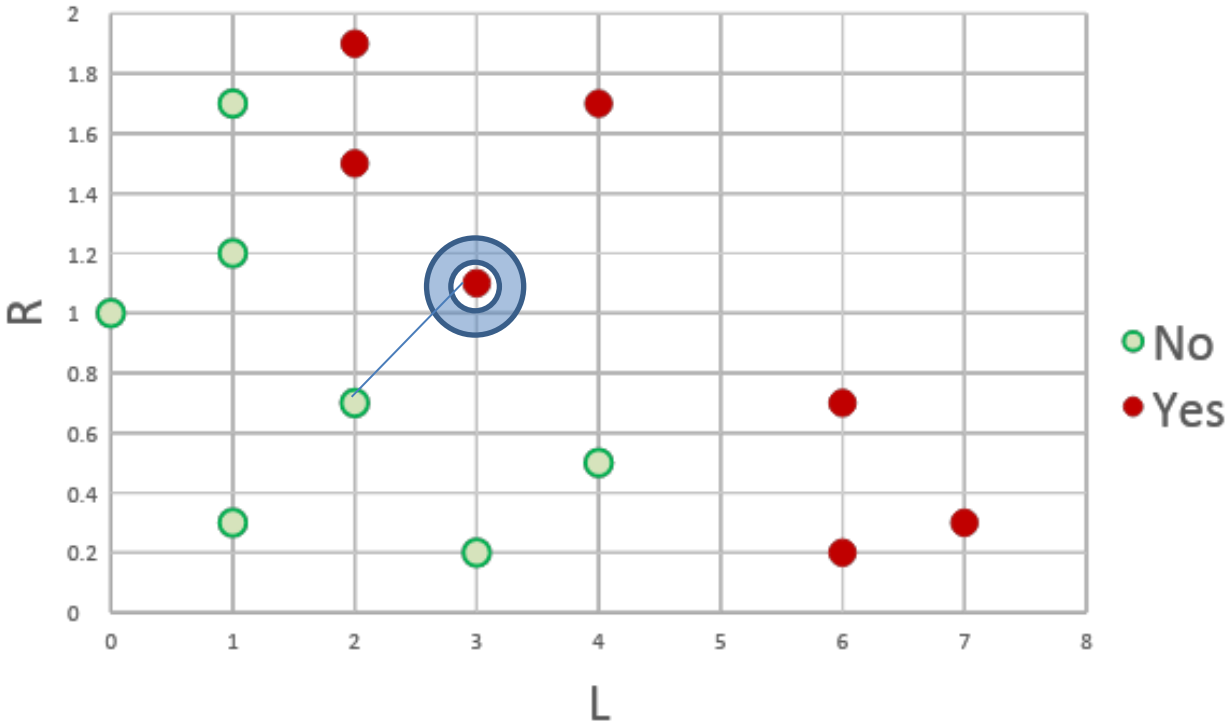
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

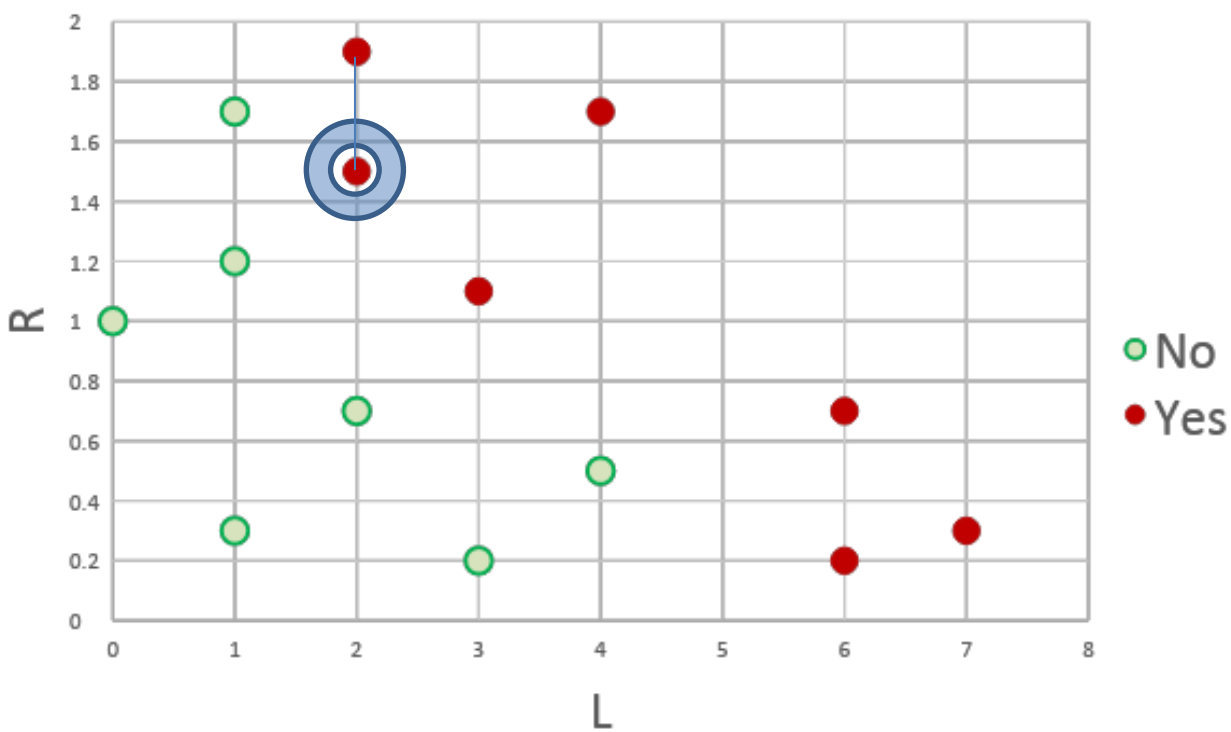
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

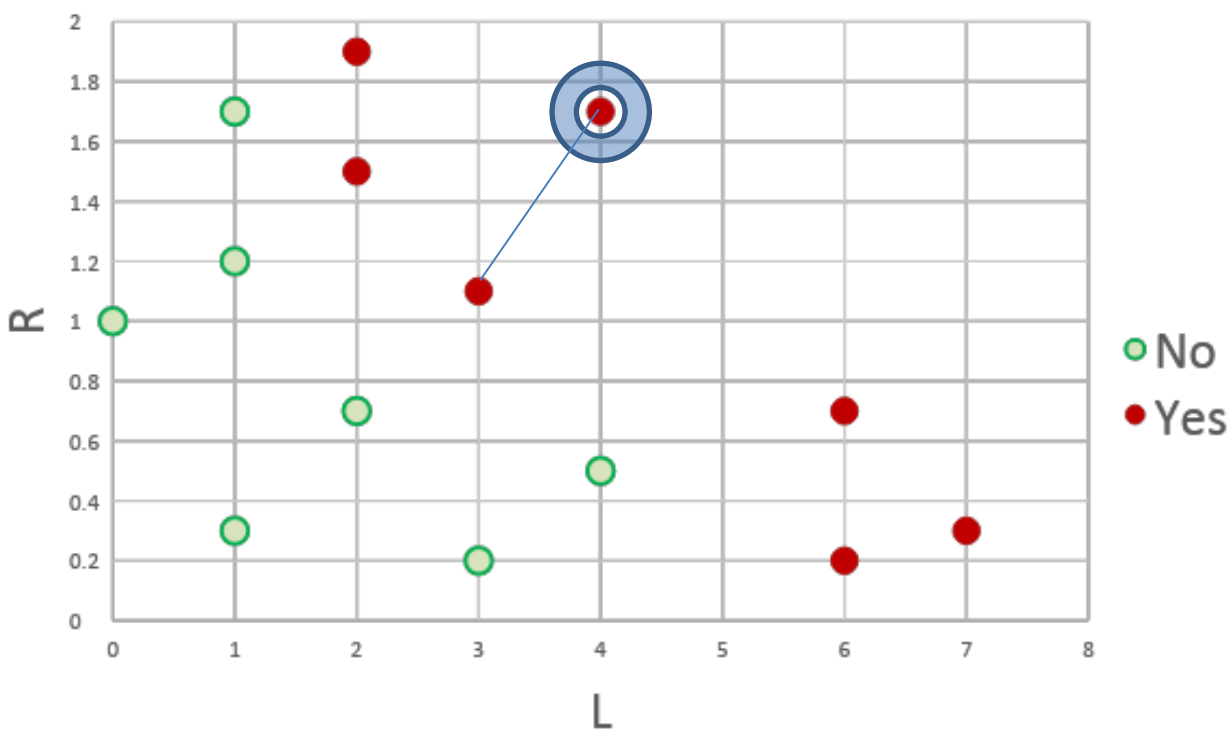
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

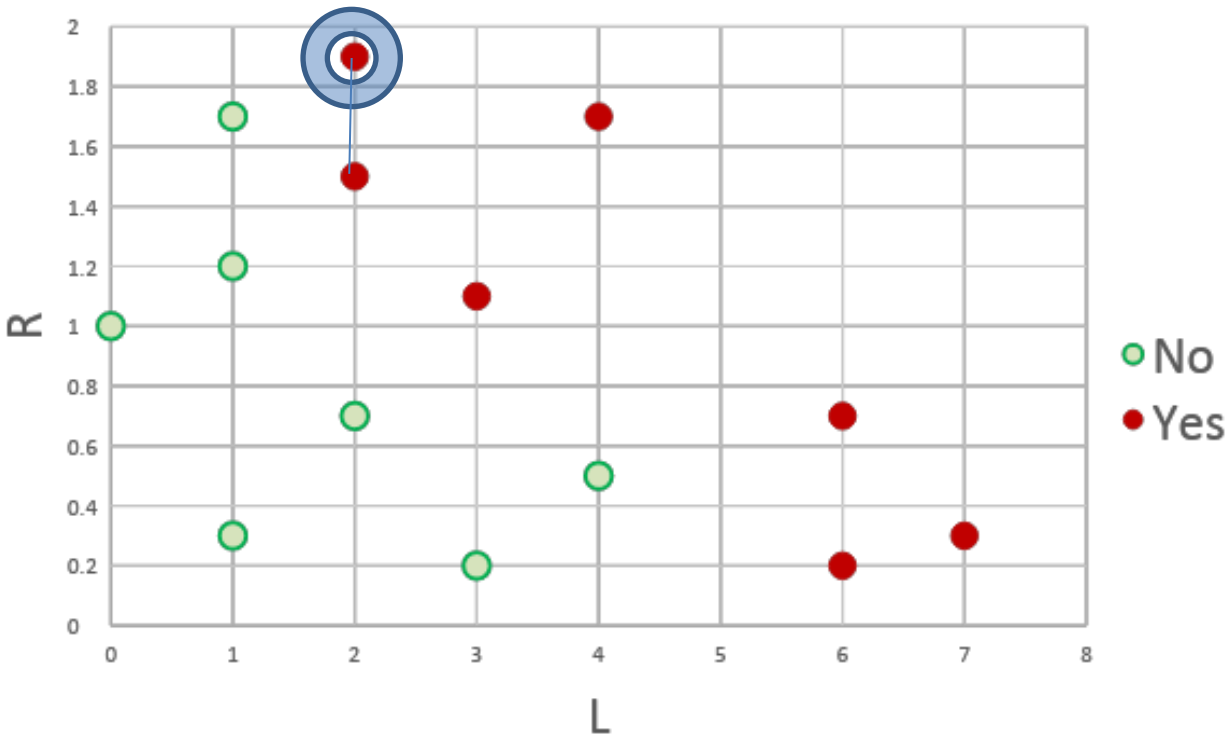
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

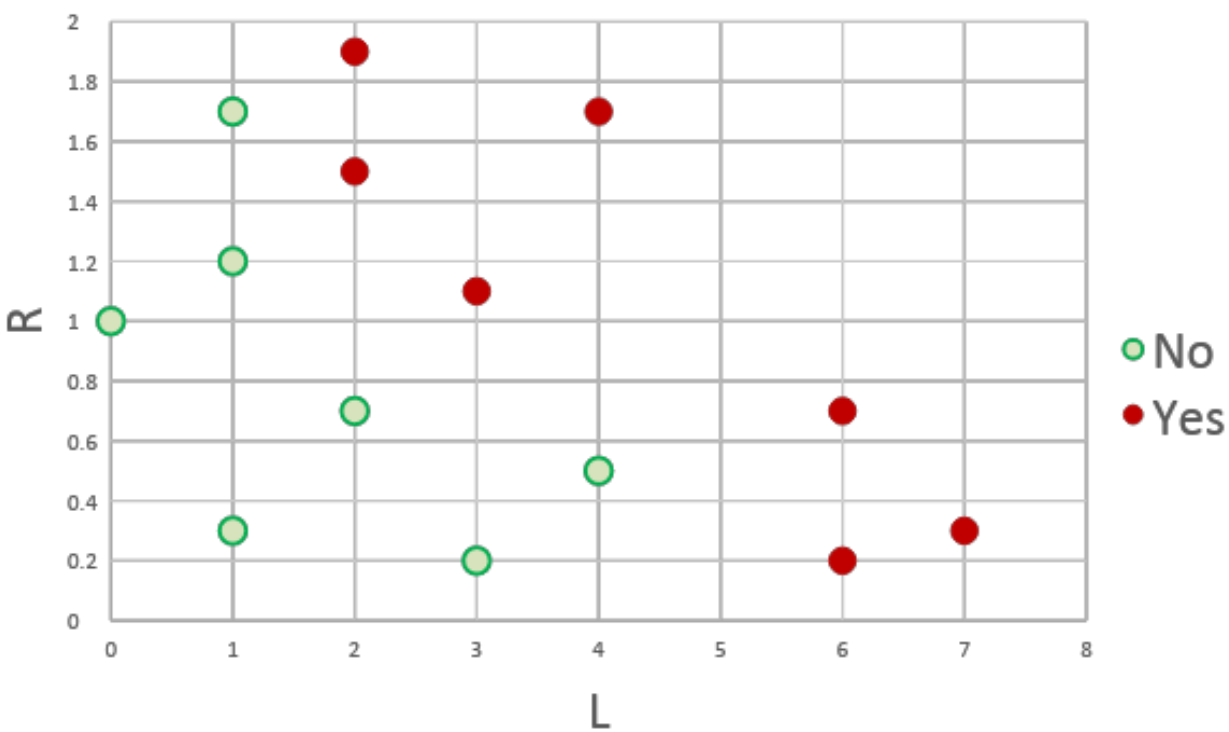
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=1

L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	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

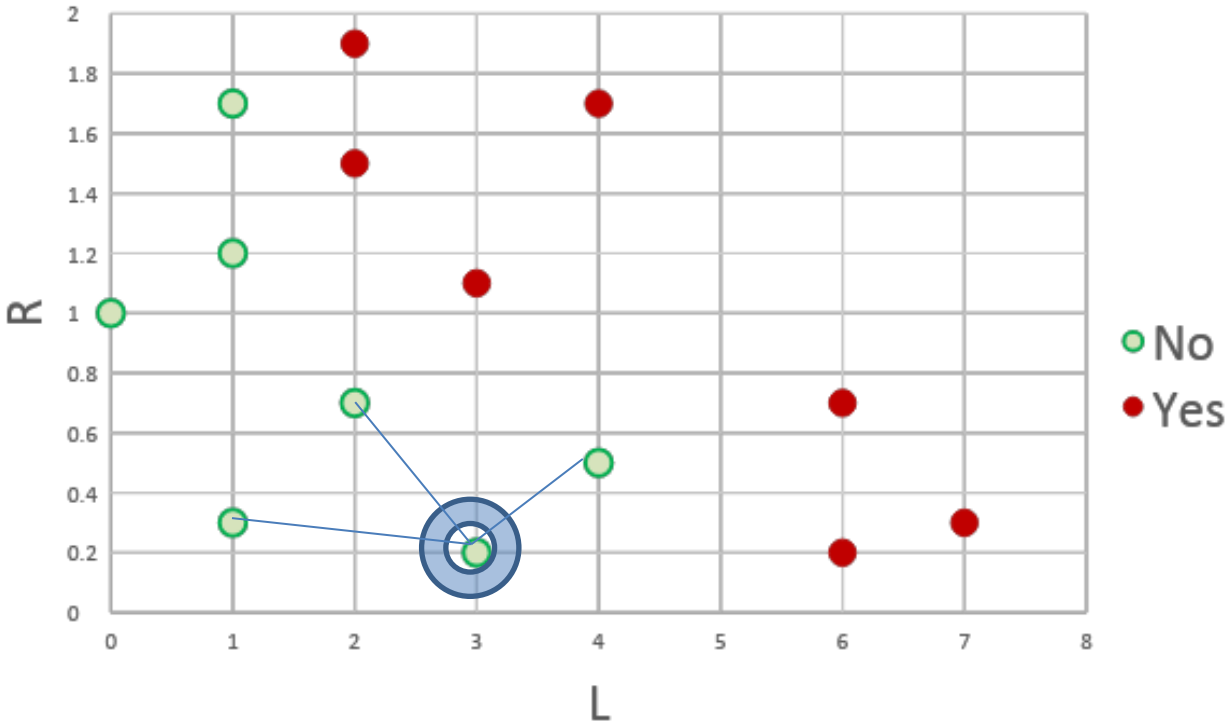


For *K*=1:

Error rate 3/14

Leave-one-out cross validation: *K*=3

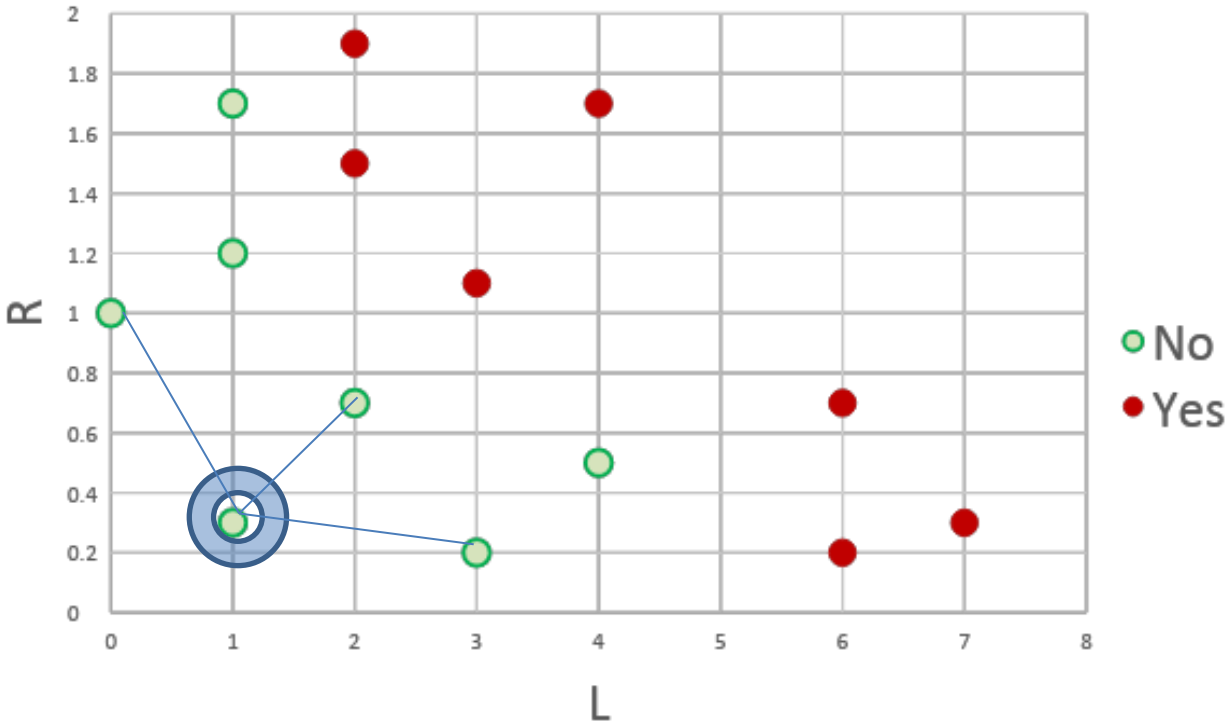
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

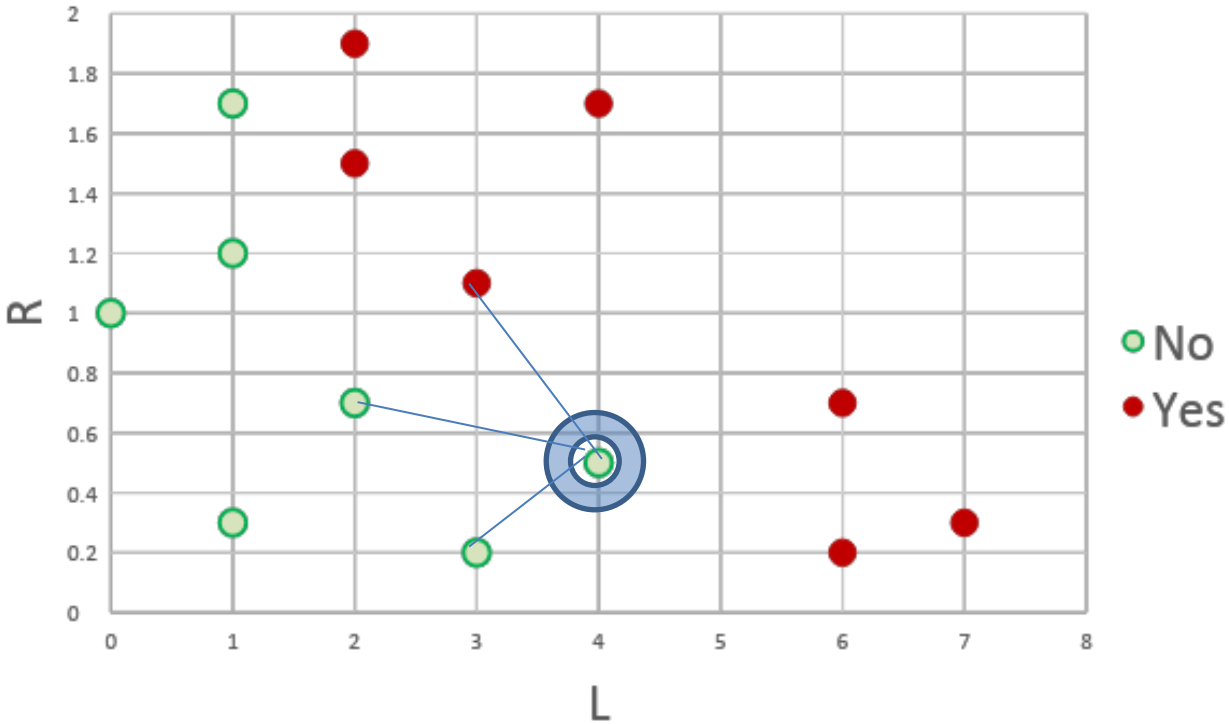
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

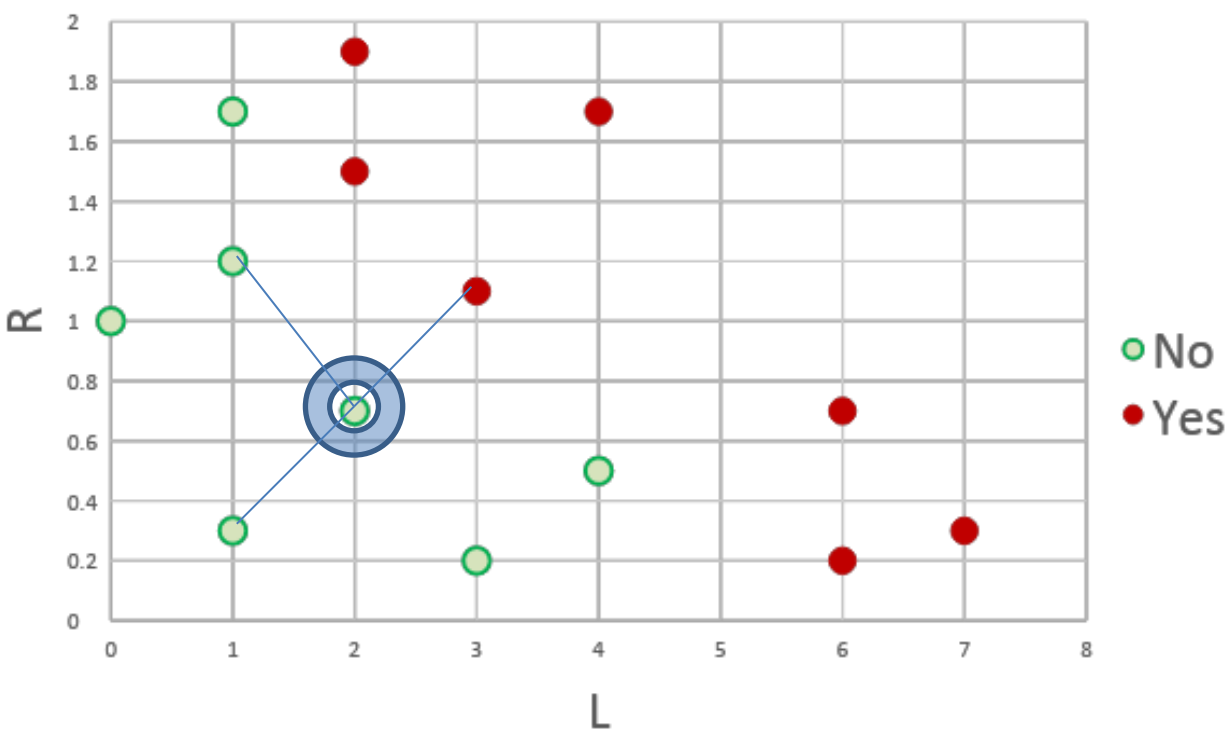
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

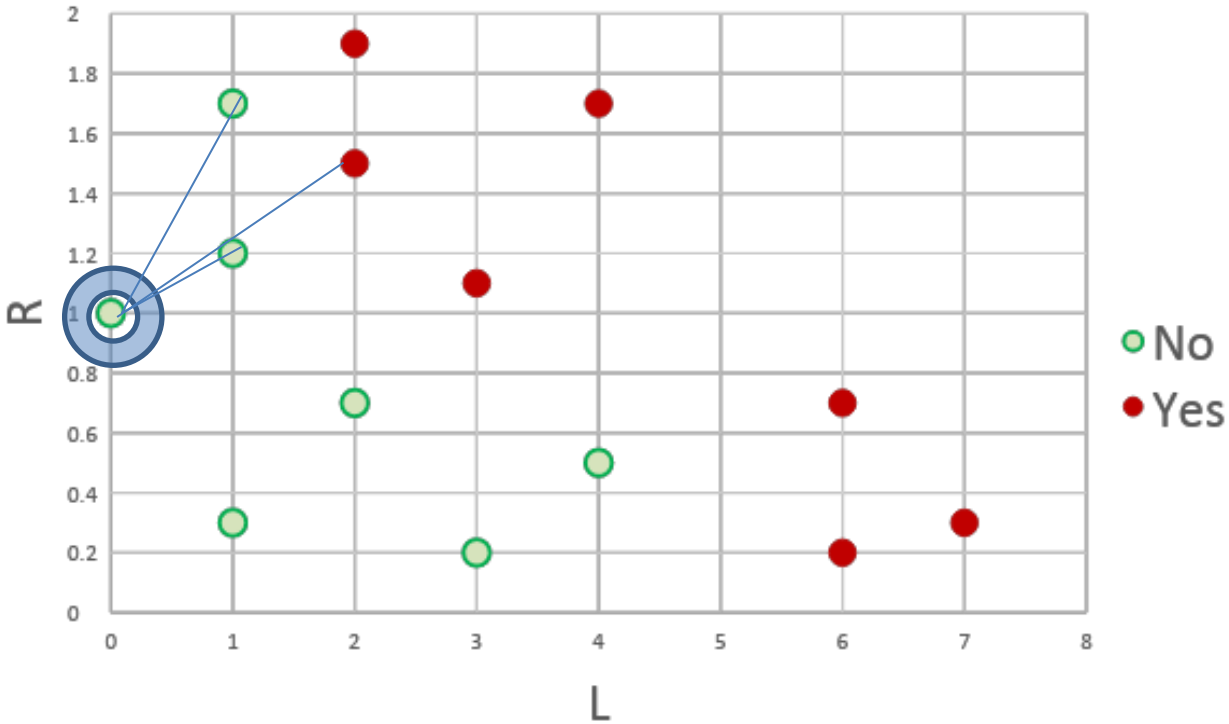
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

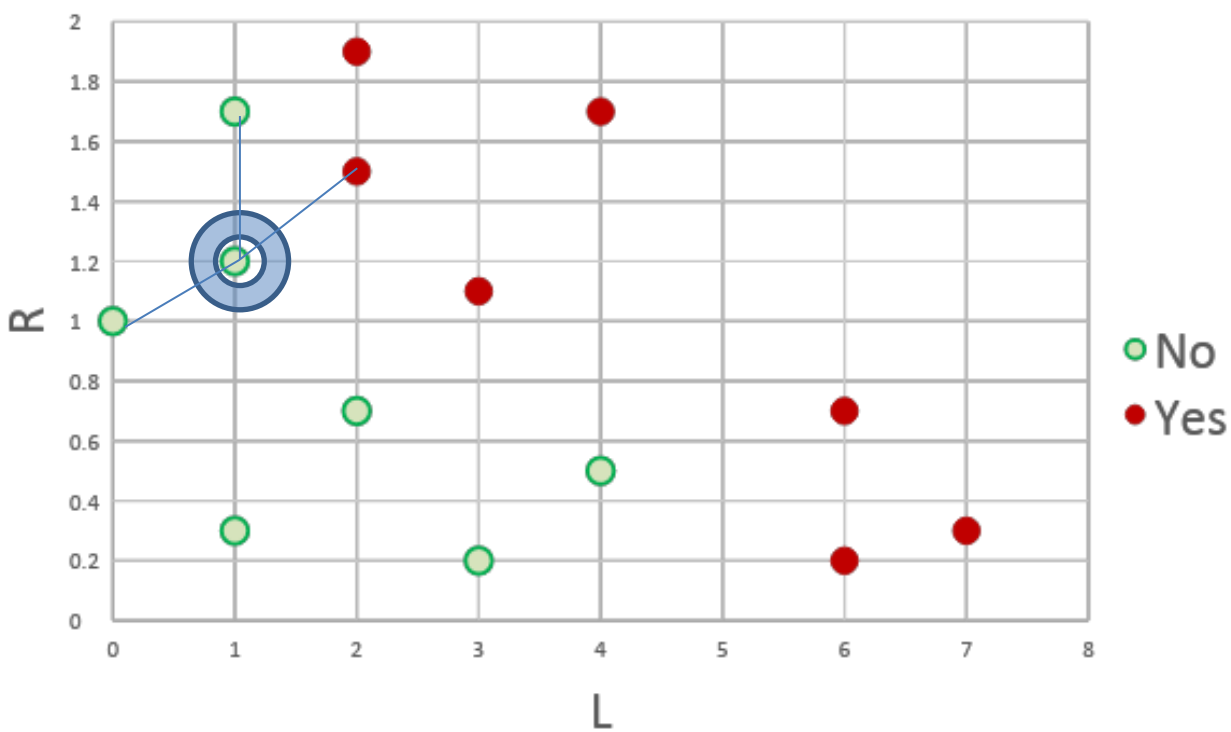
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

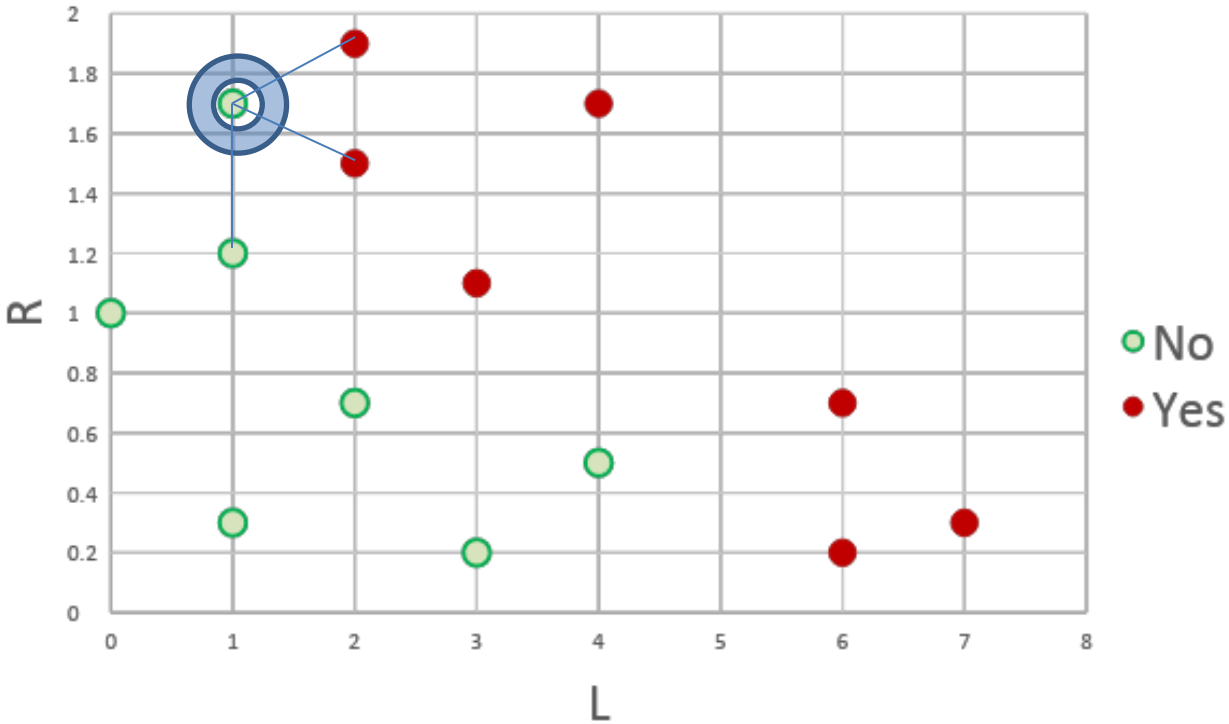
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

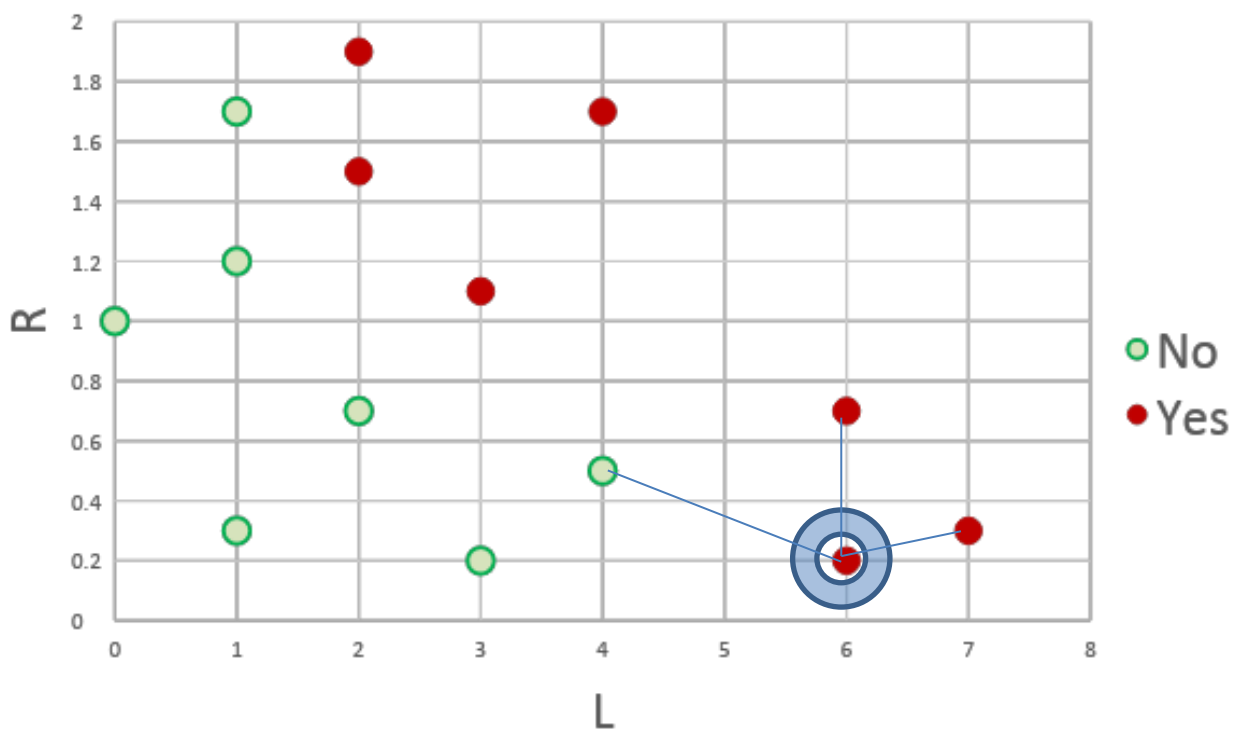
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

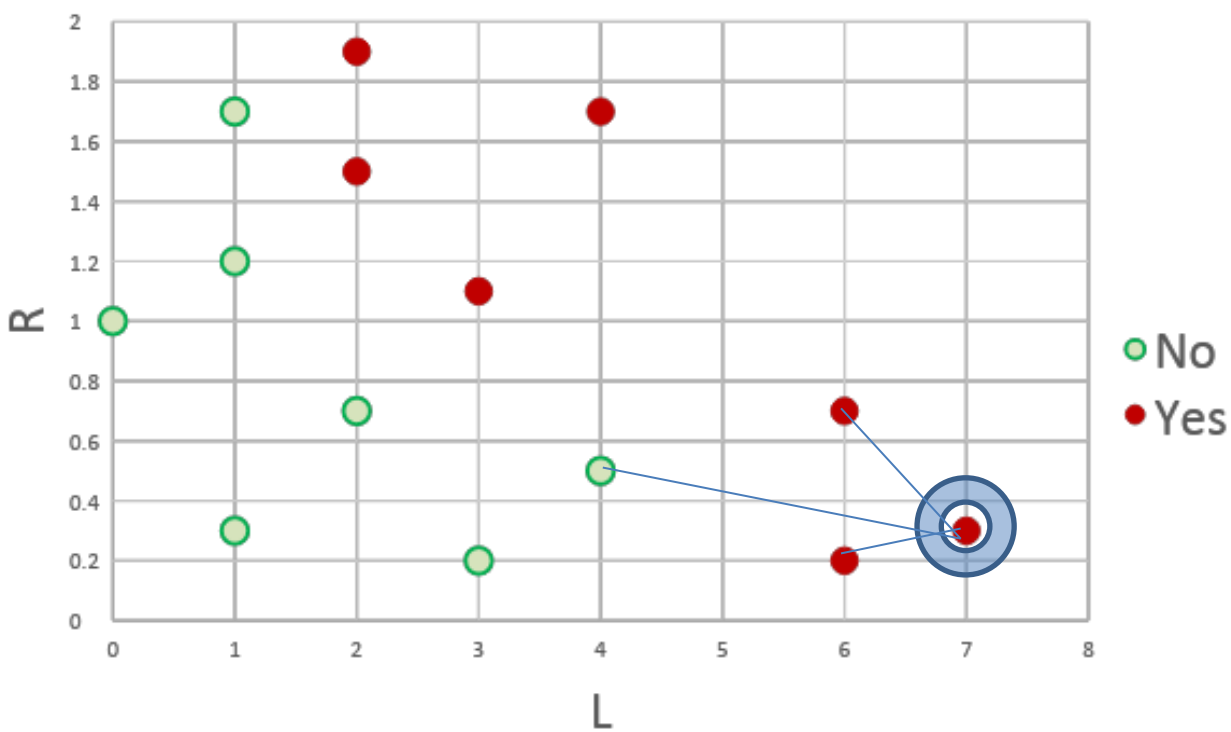
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

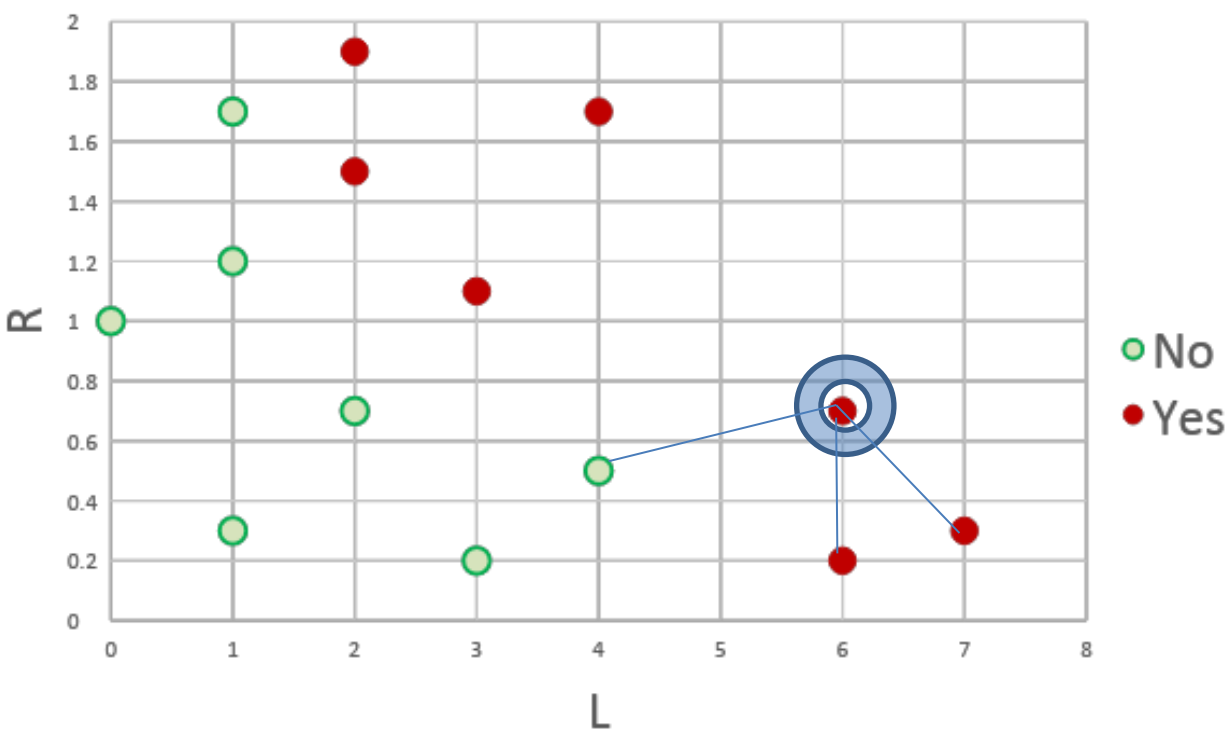
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

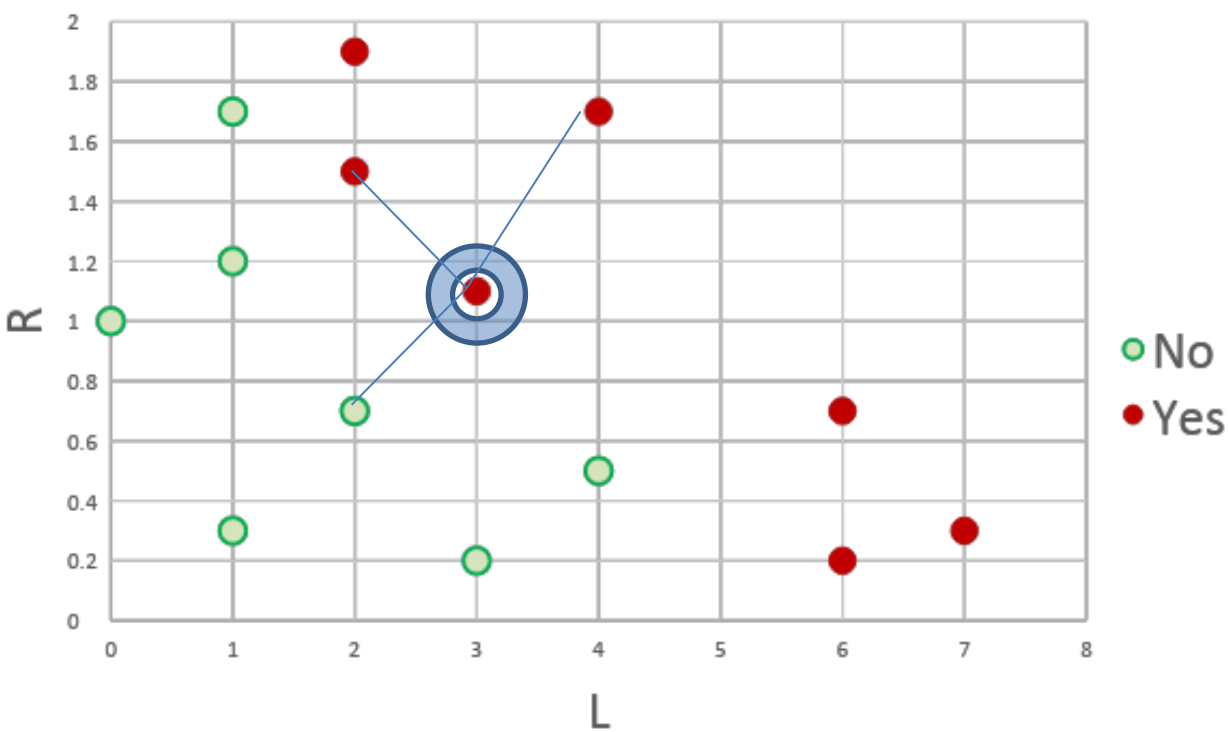
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

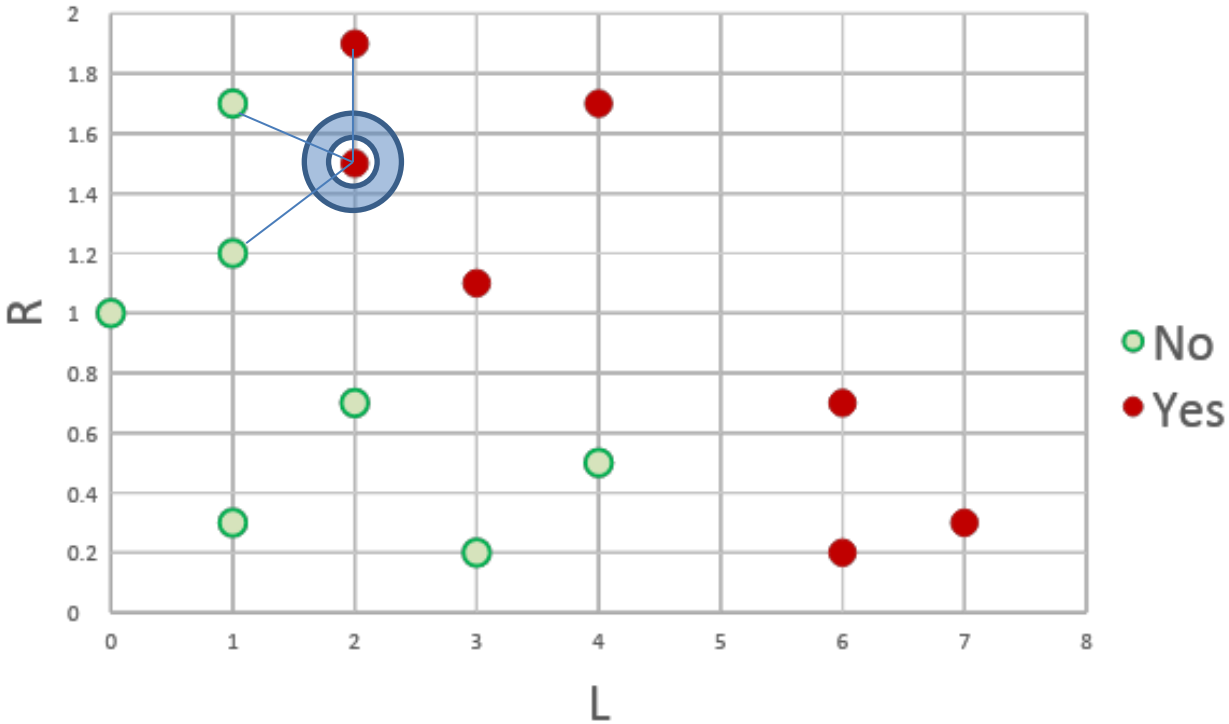
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

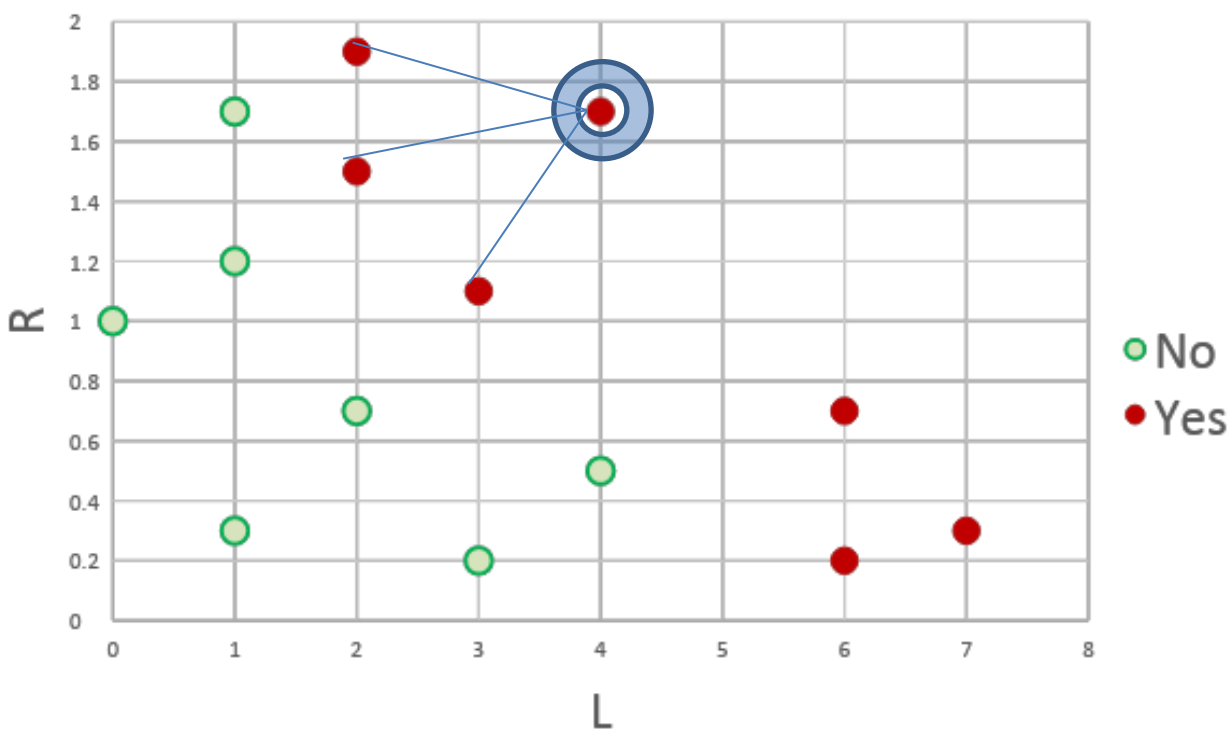
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

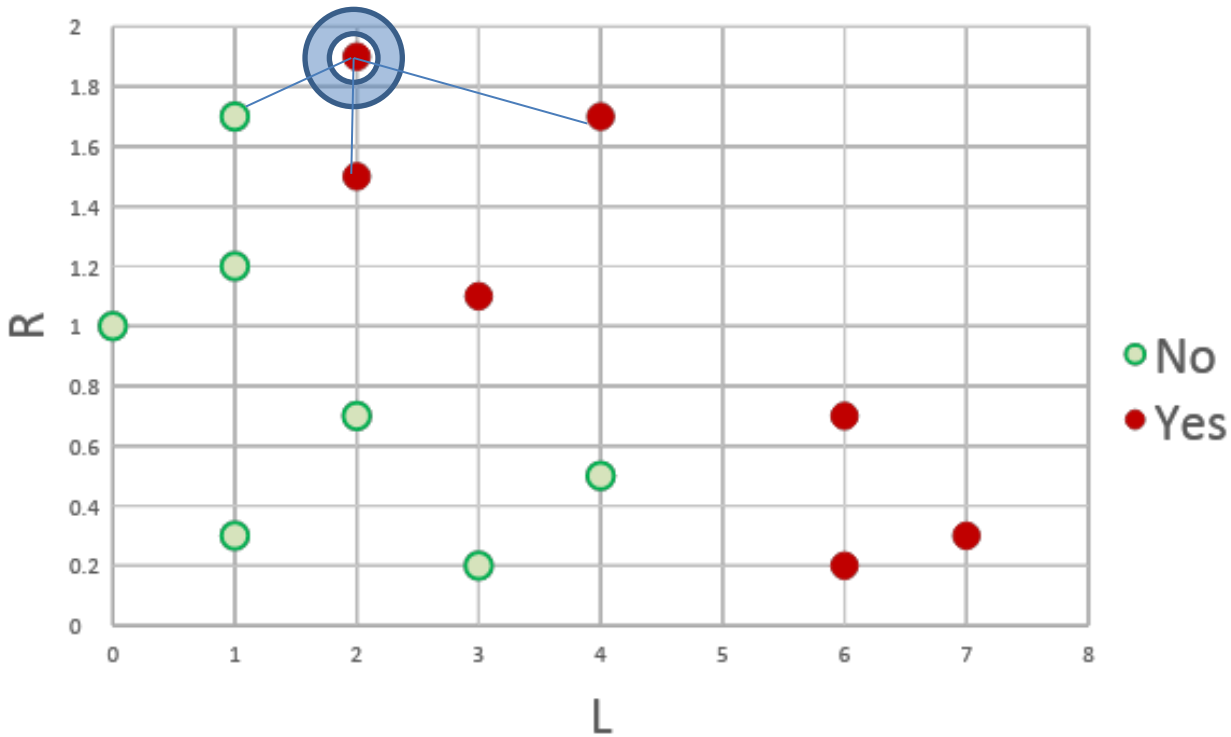
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

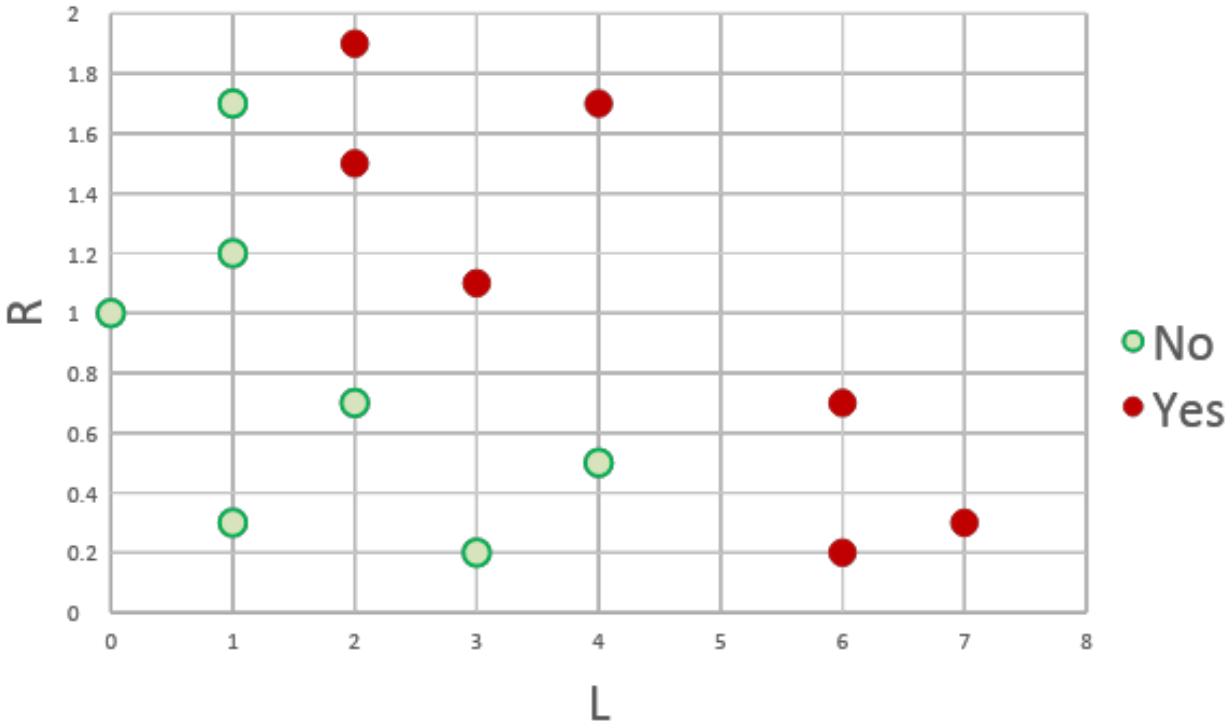
L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	No
1	1.2	No
1	1.7	No
6	0.2	Yes
7	0.3	Yes
6	0.7	Yes
3	1.1	Yes
2	1.5	Yes
4	1.7	Yes
2	1.9	Yes



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

Leave-one-out cross validation: *K*=3

L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	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

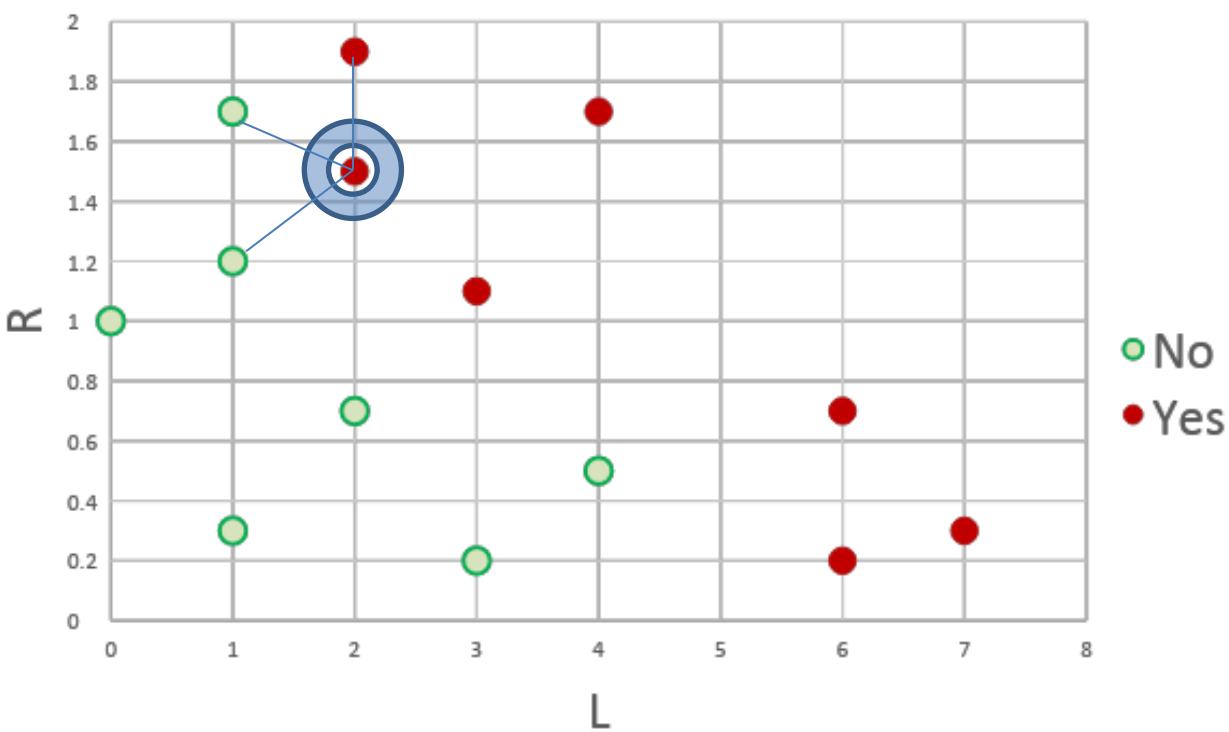


For ***K*=1**:
Error rate **3/14**

For ***K*=3**:
Error rate **2/14**

Leave-one-out cross validation: new error with *K*=3

L	R	B
3	0.2	No
1	0.3	No
4	0.5	No
2	0.7	No
0	1	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



For *K*=1:
Error rate 3/14

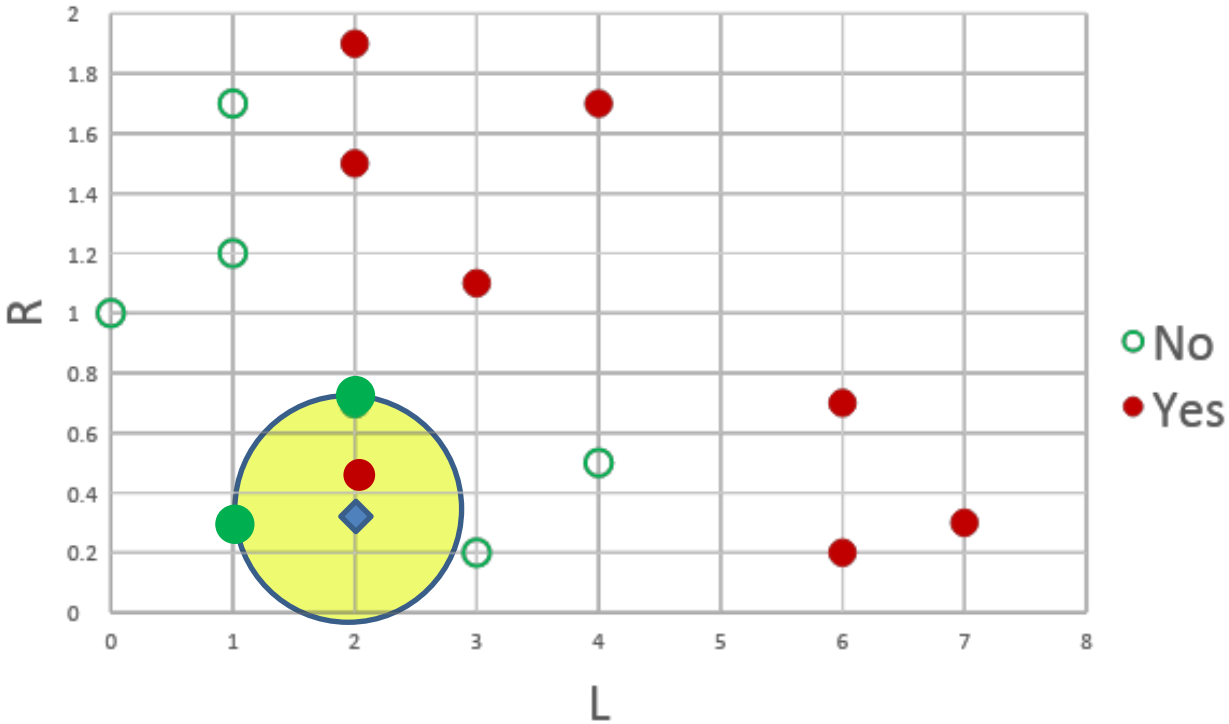
For *K*=3:
Error rate 2/14

K-NN: round 2

I. Distance/similarity between data records
II. How many neighbors: choice of K
III. Combining neighbor votes
IV. How many features (dimensions)

Majority voting (democracy)

L	R
2	0.3

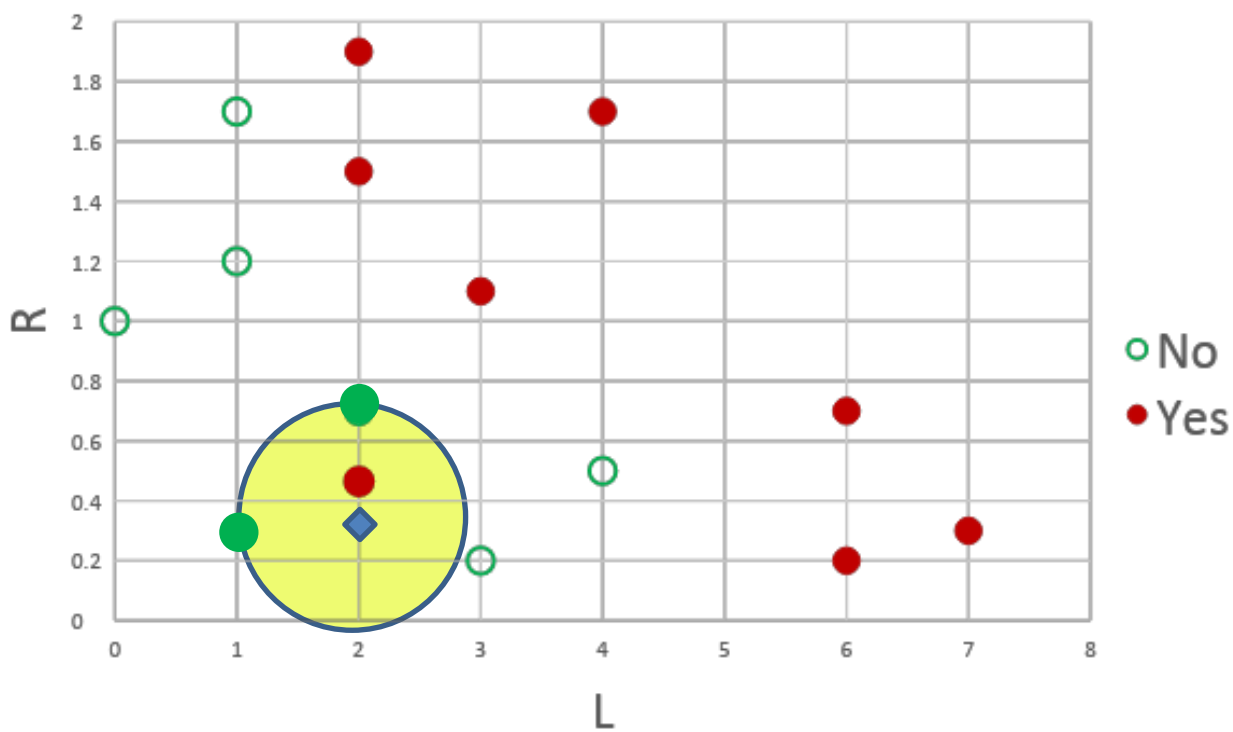


L: #late payments / year
R: expenses / income ratio

Blue diamond is classified as **No** (No bankrupt)

Weighted voting (shareholder democracy)

L	R
2	0.3



1/0.5 Yes+1/1.5 No + 1/1.5 No=2 Yes + 1.33 No = Yes!

The closest neighbor outweighs the majority class

K-NN: round 2

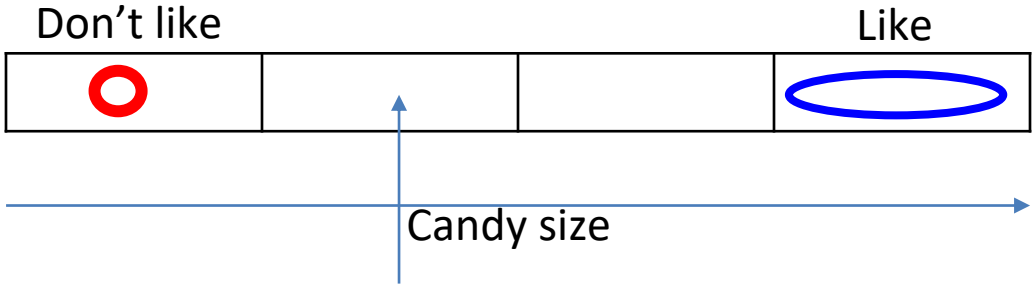
I. Distance/similarity between data records
II. How many neighbors: choice of K
III. Combining neighbor votes
IV. How many features (dimensions)

How many dimensions?

- Imagine you have one-dimensional data which can be a straight line (the line where the floor and the wall meets) and plot 100 data points
- Now let's make this a 2D - a wall. Plot the same 100 points.
- Moving on, let's imagine a 3D which can be the room that has the wall in it. Again plot the 100 points.
- The points become more sparse as we move from a line to a wall and to a room. In a high dimensional space the same number of points are now separated by an exponentially large distance.
- The prediction in sparse high-dimensional space will be less reliable: the distance between points increases exponentially thus making predictions on sparse data becomes next to impossible.

The curse of dimensionality: example

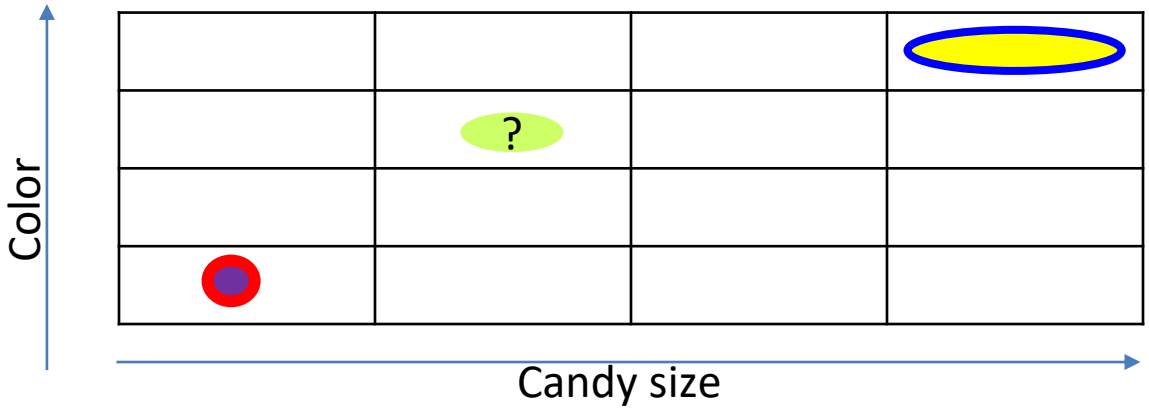
1 dimension



Predict: 

Prediction is always reliable

2 dimensions



Predict: 

Prediction is not reliable:
need more data points

K-NN algorithm. Summary

- The training set ***is the*** model
- Advantages:
  - Building a classifier: zero work
  - Updating the model with every new record: zero work
  - Interpretable: we can justify our classification
  - Good for predicting numeric values (Regressor)
- Disadvantages:
  - The query is computationally expensive!