Introduction to Data Science 3

Overview

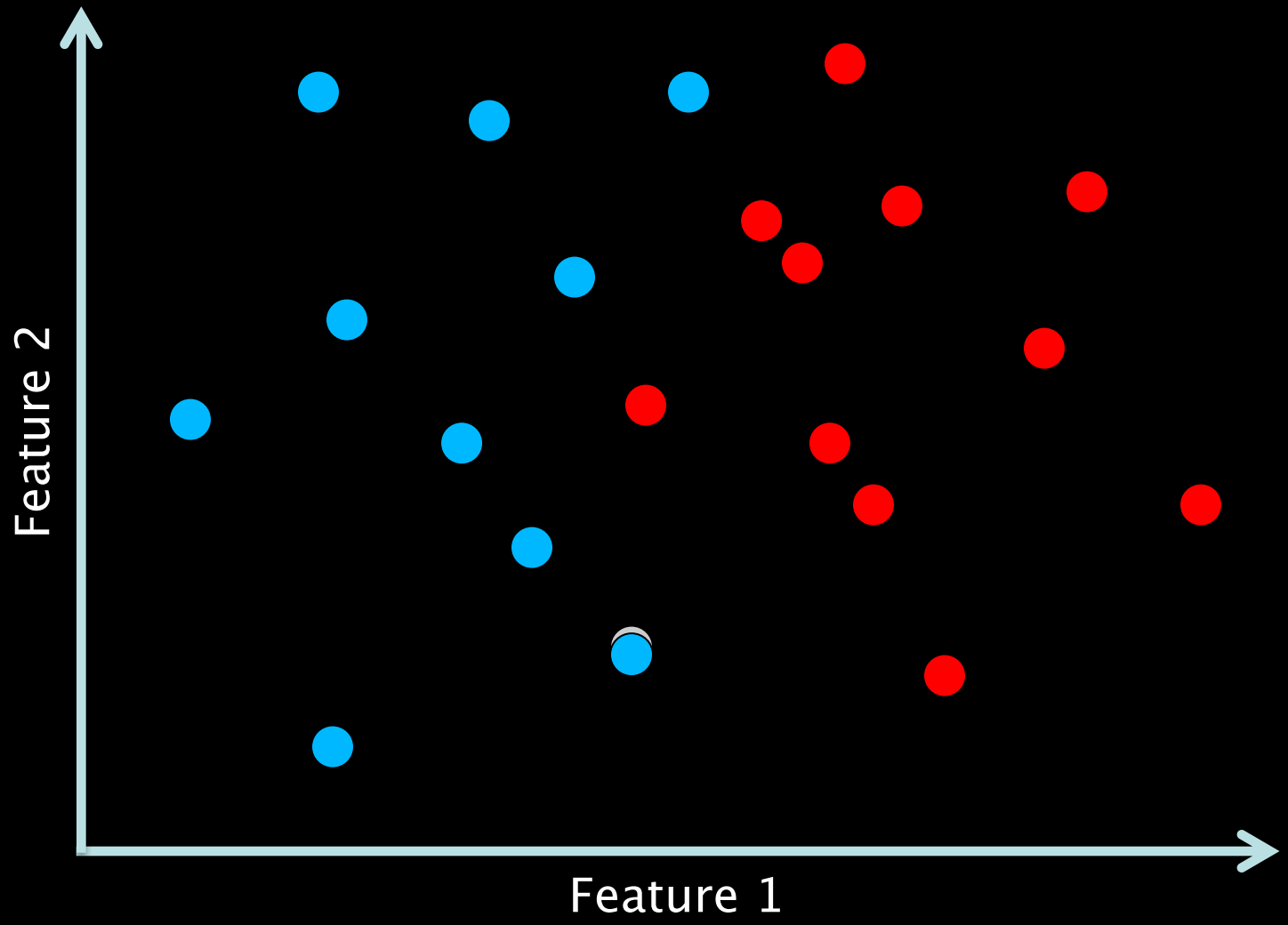
Cross-validation

From classification to regression (and back)

Model complexity

Underfitting and overfitting

Evaluation:
how well does it estimate the class labels of the dots?



Evaluation on Training Set

Determines how well a classifier can reproduce what it has seen before.

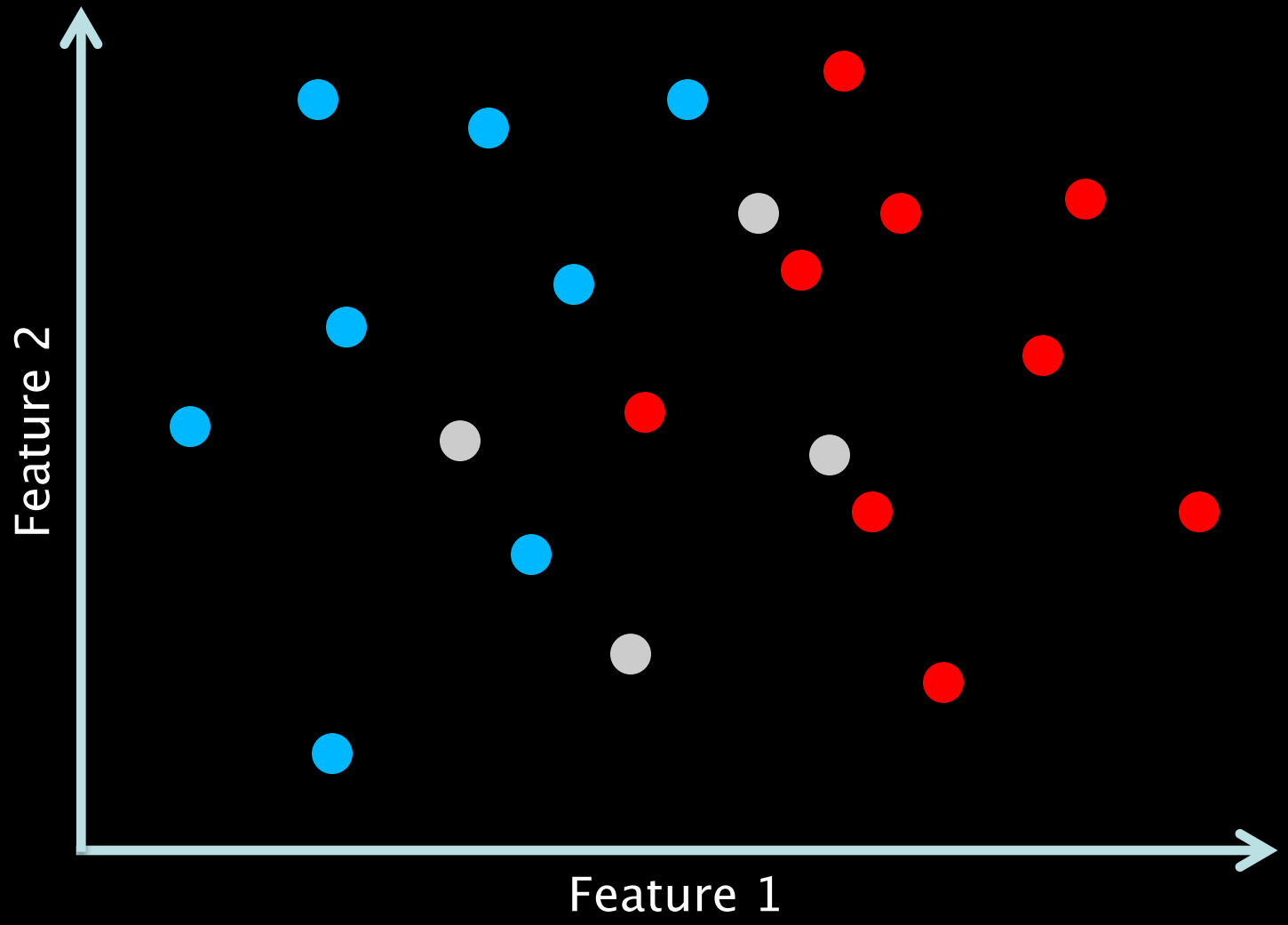
In case of the 1-NN classifier, reproduction is perfect (100%) if it has a feature that uniquely points to an instance and its label. (e.g., passenger-id in Titanic.arff)

We are NOT interested in reproduction!

Skills Class 2

Evaluation (“testing”) on test set

Evaluation:
how well does it estimate the class labels of the GRAY (unlabeled) dots?



Evaluation on Test Set

Determines how well a classifier can predict the labels of previously unseen instances on the basis of what it has seen before (in the training set).

We ARE interested in prediction!

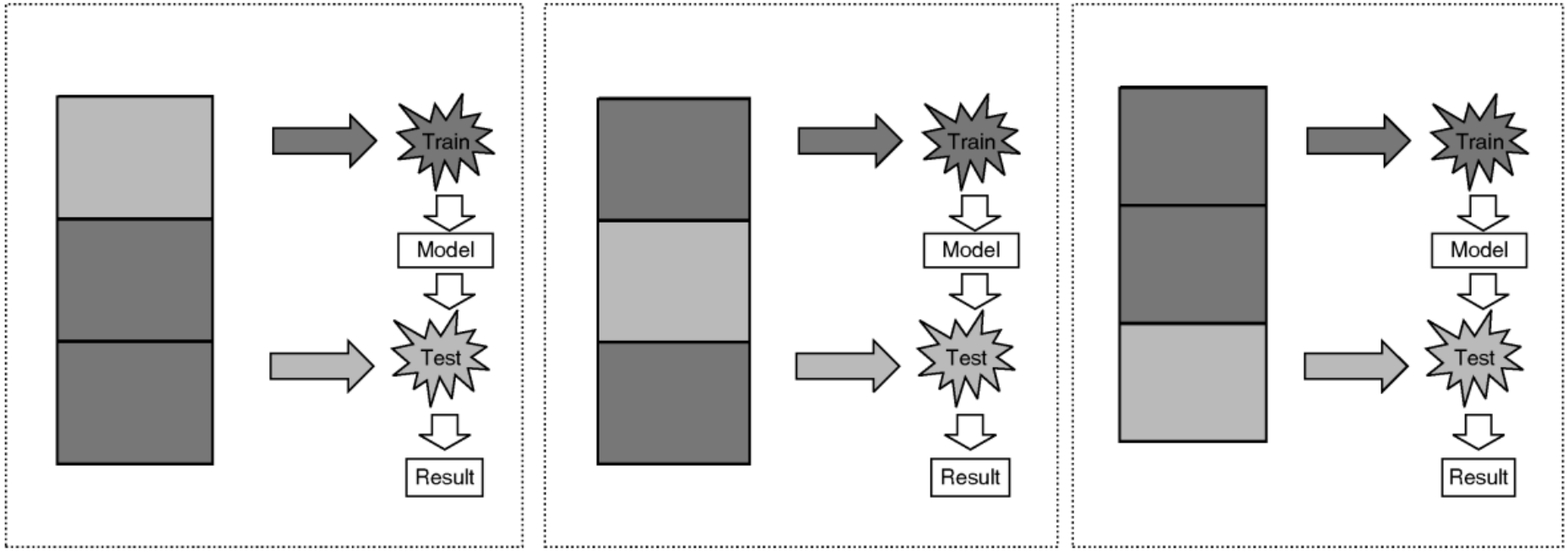
Selection bias

The selection of the test set (e.g., 10% of dataset) can affect the test results (positively or negatively) depending on the representativeness of the instances in the test set of the entire data set.

How to remove this selection bias?

Cross validation

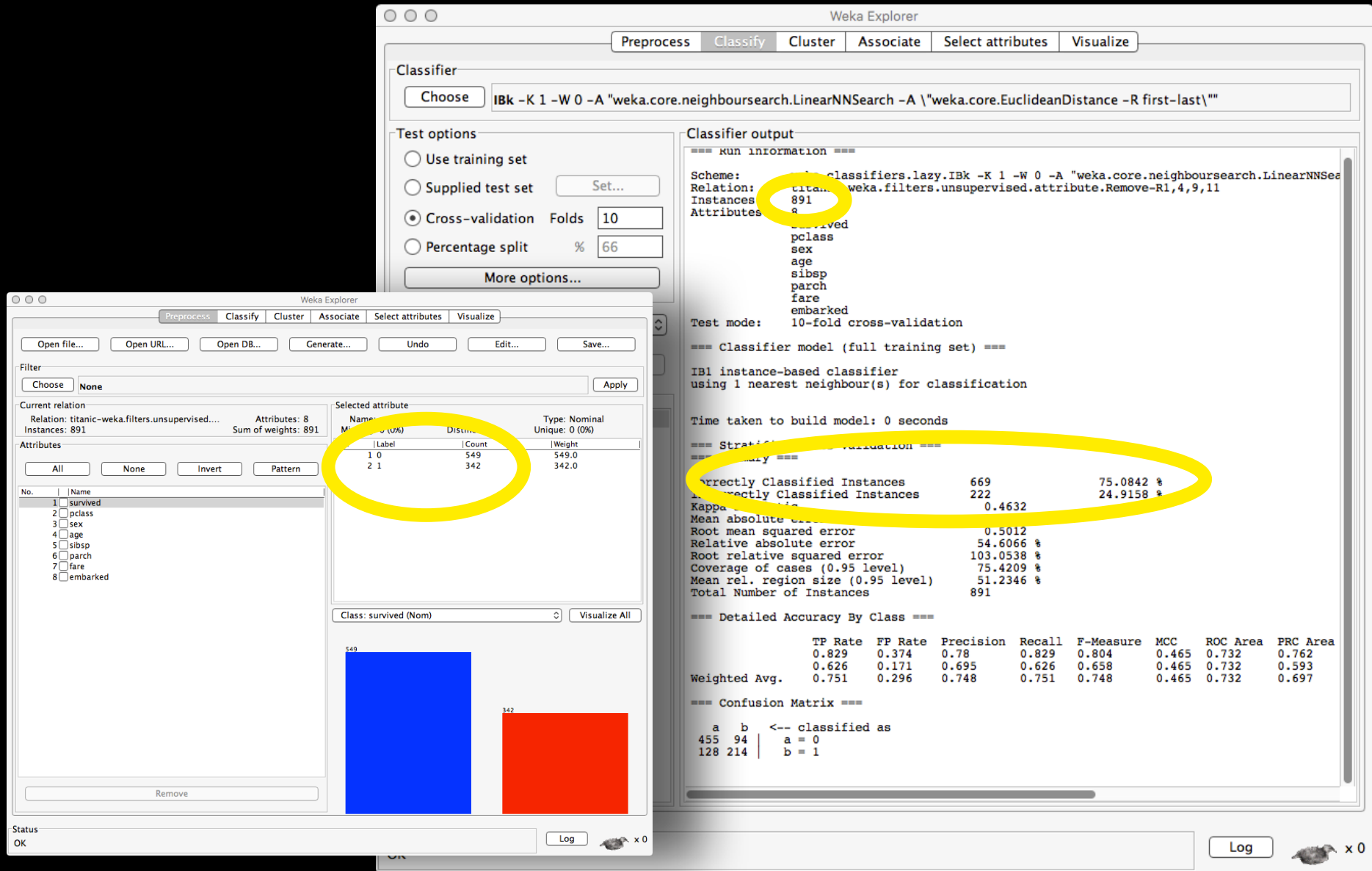
3-fold Cross Validation



Cross-Validation. Figure 1. Procedure of three-fold cross-validation.

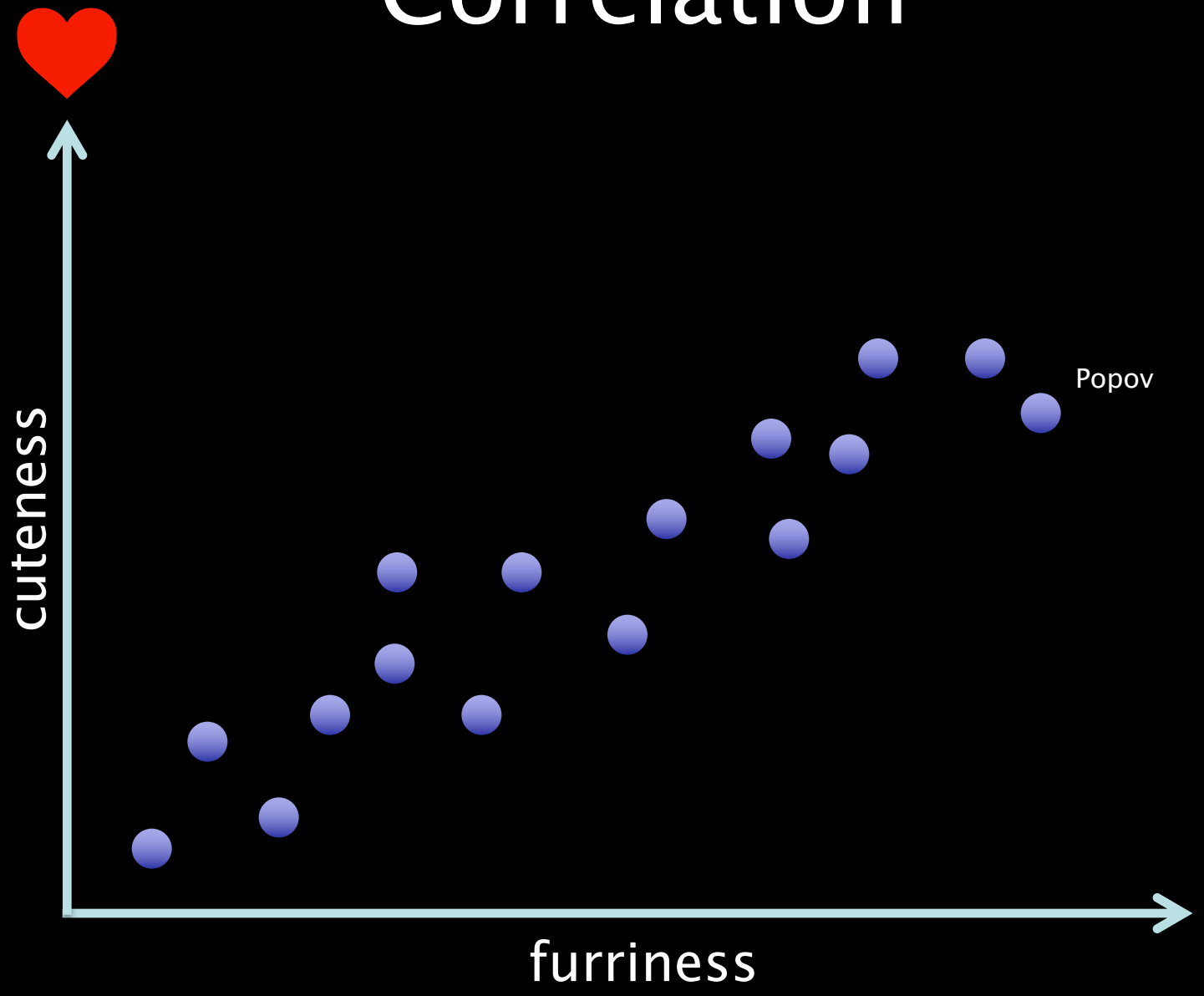
General case: k-fold Cross Validation.
When k=N: Leaving-One-Out Cross Validation.

WEKA


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
=== Run information ===
Scheme:
Relation:
Instances 891
Attributes 8
pclass
sex
age
sibsp
parch
fare
embarked
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
IB1 instance-based classifier
using 1 nearest neighbour(s) for classification
Time taken to build model: 0 seconds
Root mean squared error 0.5012
Relative absolute error 54.6066 %
Root relative squared error 103.0538 %
Coverage of cases (0.95 level) 75.4209 %
Mean rel. region size (0.95 level) 51.2346 %
Total Number of Instances 891
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area
0.829 0.374 0.78 0.829 0.804 0.465 0.732 0.762
0.626 0.171 0.695 0.626 0.658 0.465 0.732 0.593
Weighted Avg. 0.751 0.296 0.748 0.751 0.748 0.465 0.732 0.697
=== Confusion Matrix ===
a b <-- classified as
455 94 | a = 0
128 214 | b = 1
Log
x 0
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Current relation
Relation: titanic-weka.filters.unsupervised....
Instances: 891
Attributes: 8
Sum of weights: 891
Selected attribute
Type: Nominal
Unique: 0 (0%)
Label Count Weight
1 0 549 549.0
2 1 342 342.0
Attributes
All
None
Invert
Pattern
No. Name
1 survived
2 pclass
3 sex
4 age
5 sibsp
6 parch
7 fare
8 embarked
Remove
Class: survived (Nom)
Visualize All
549
342
Status
OK
Log
x 0

From classification to regression

Correlation



Description versus Prediction

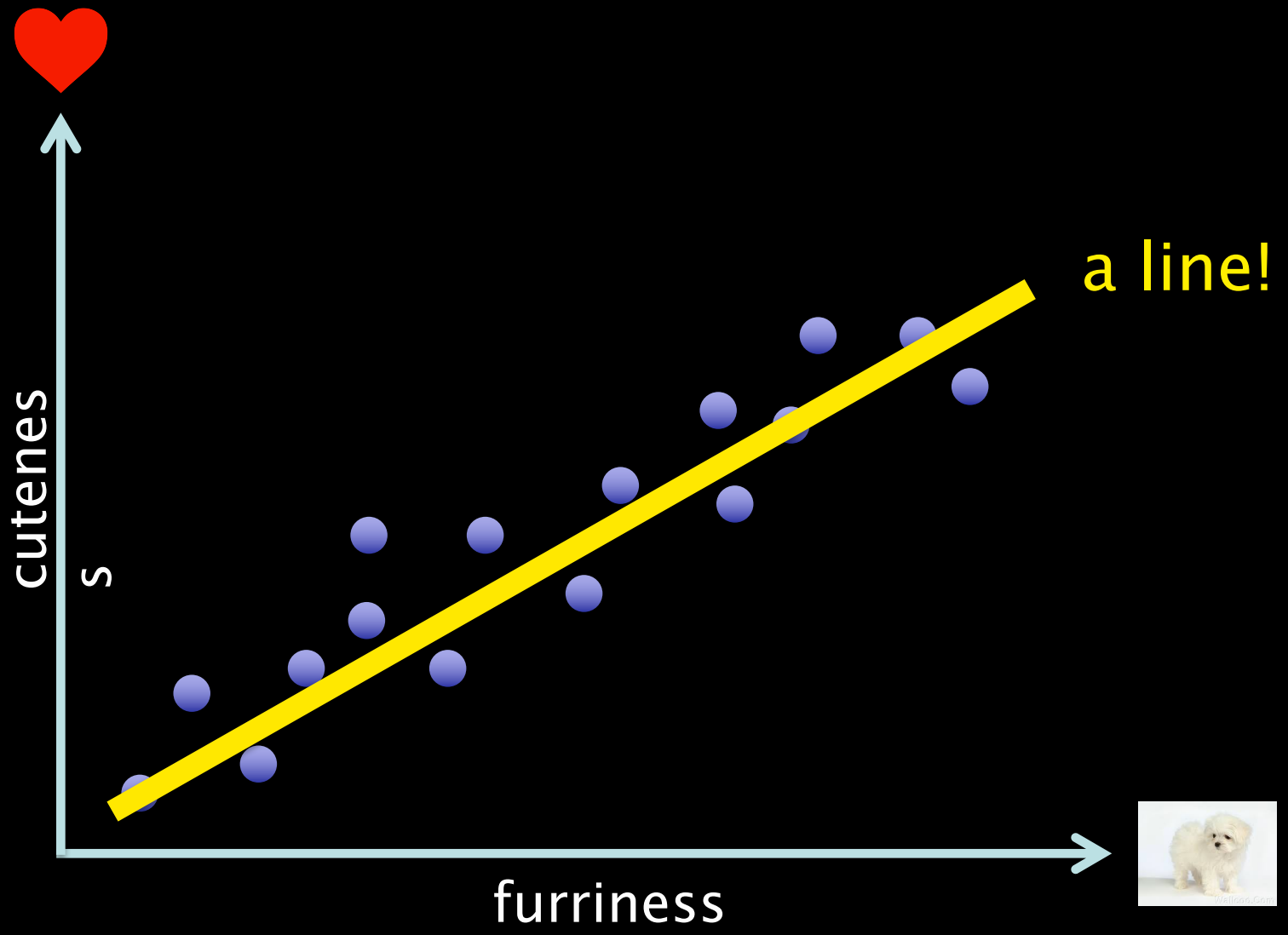
- Correlation is **describing** the data
- We want to **predict** data,
- therefore we **model** the data

What is a model?

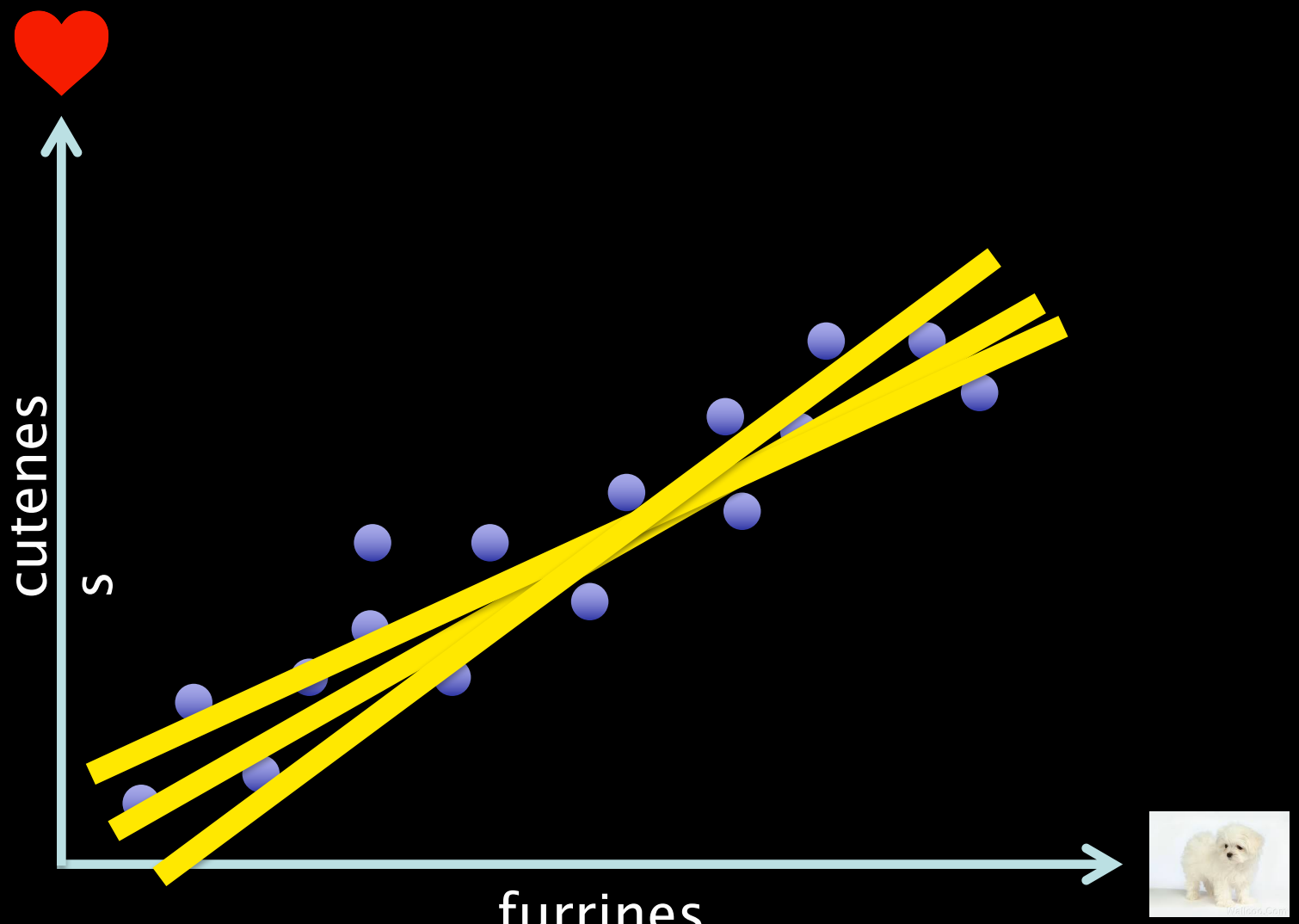
- In machine learning, a model takes the feature(s) as input and generates as output a label estimate

- Input: furriness
- Output: estimate of cuteness
- Evaluation of the model: difference between estimates and true cuteness values

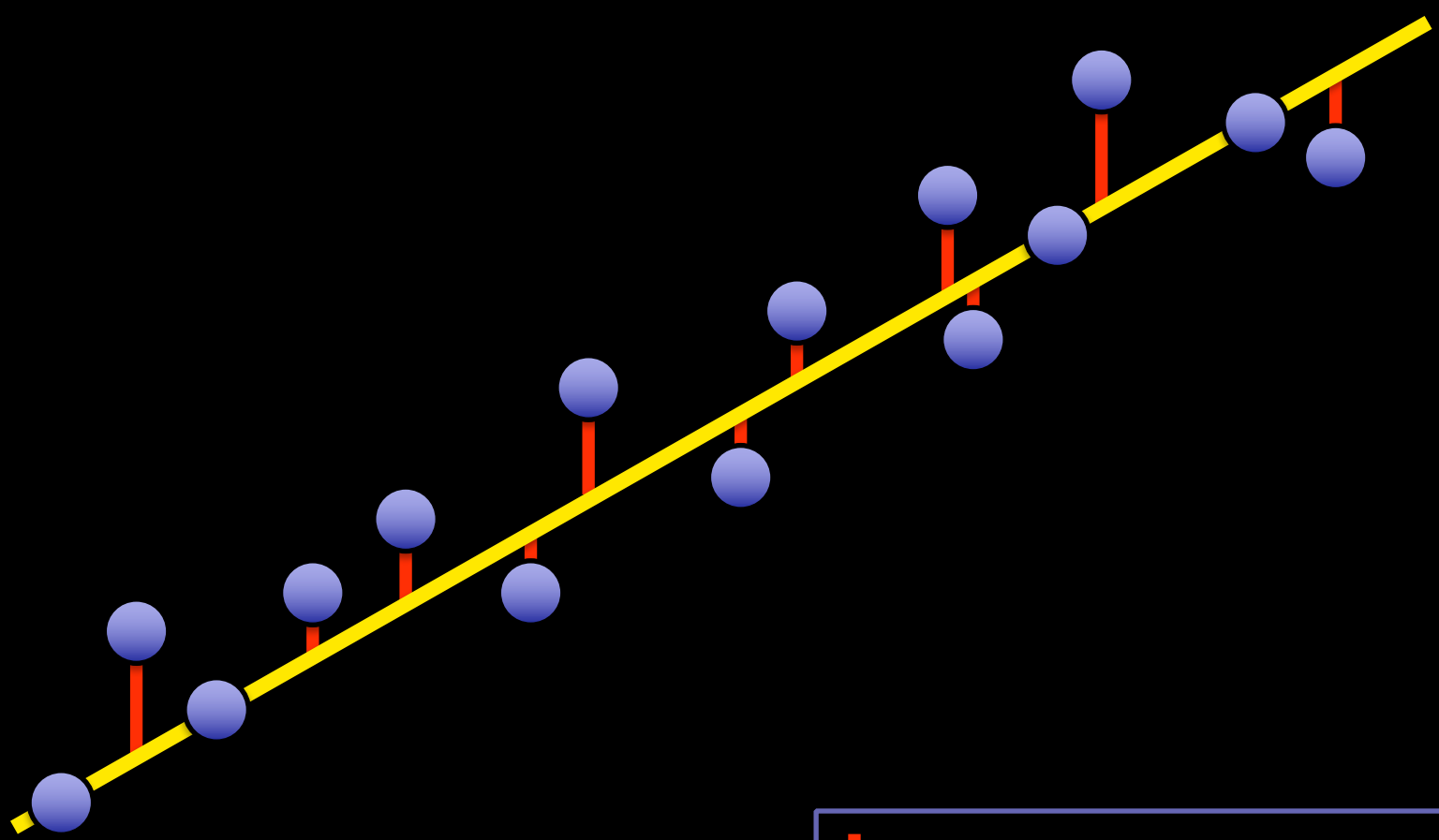
What is a suitable model?



What is the best line?

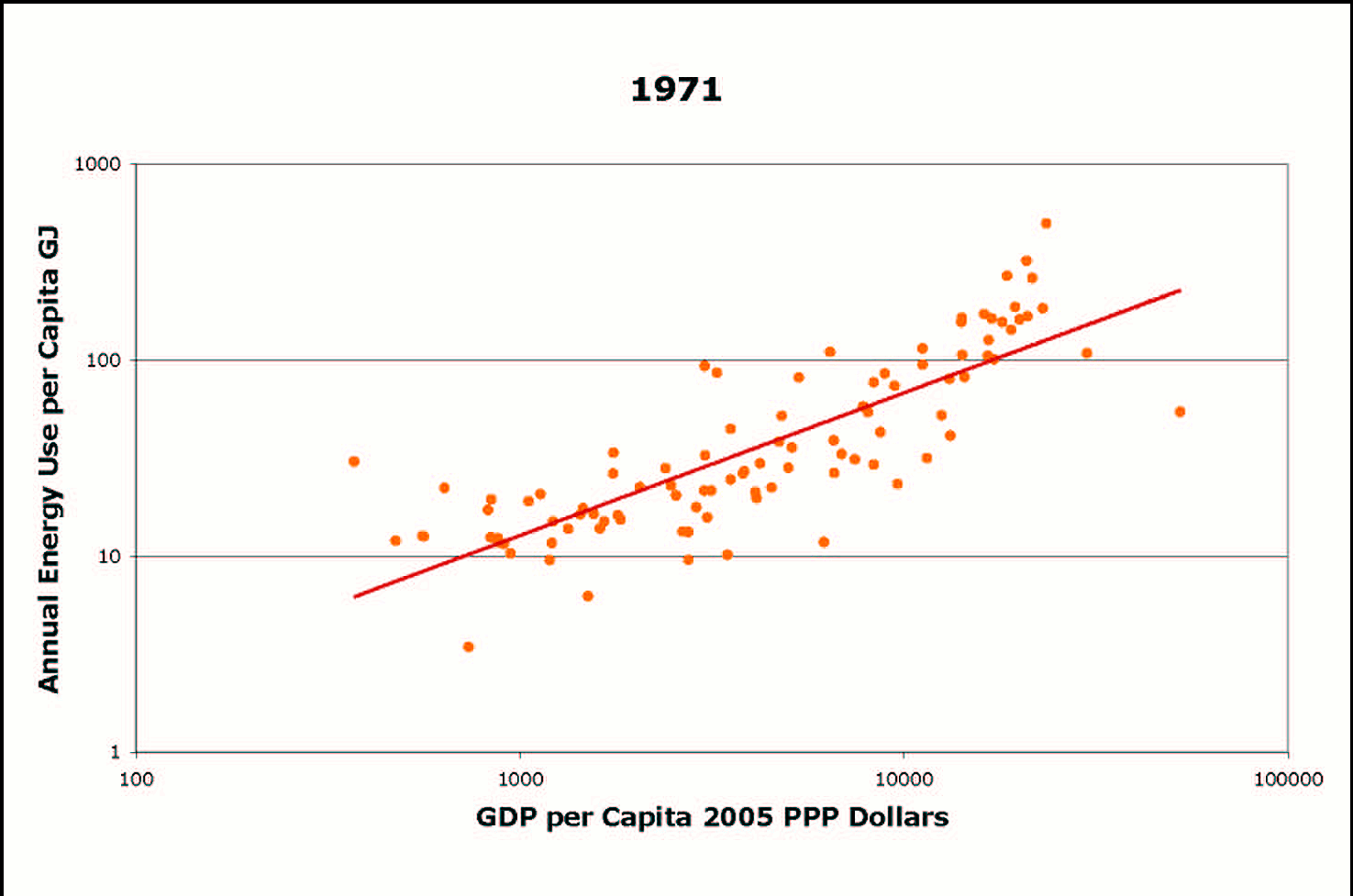


Answer: the best-fitting line



deviations from the model = error

The best-fitting line can be determined automatically



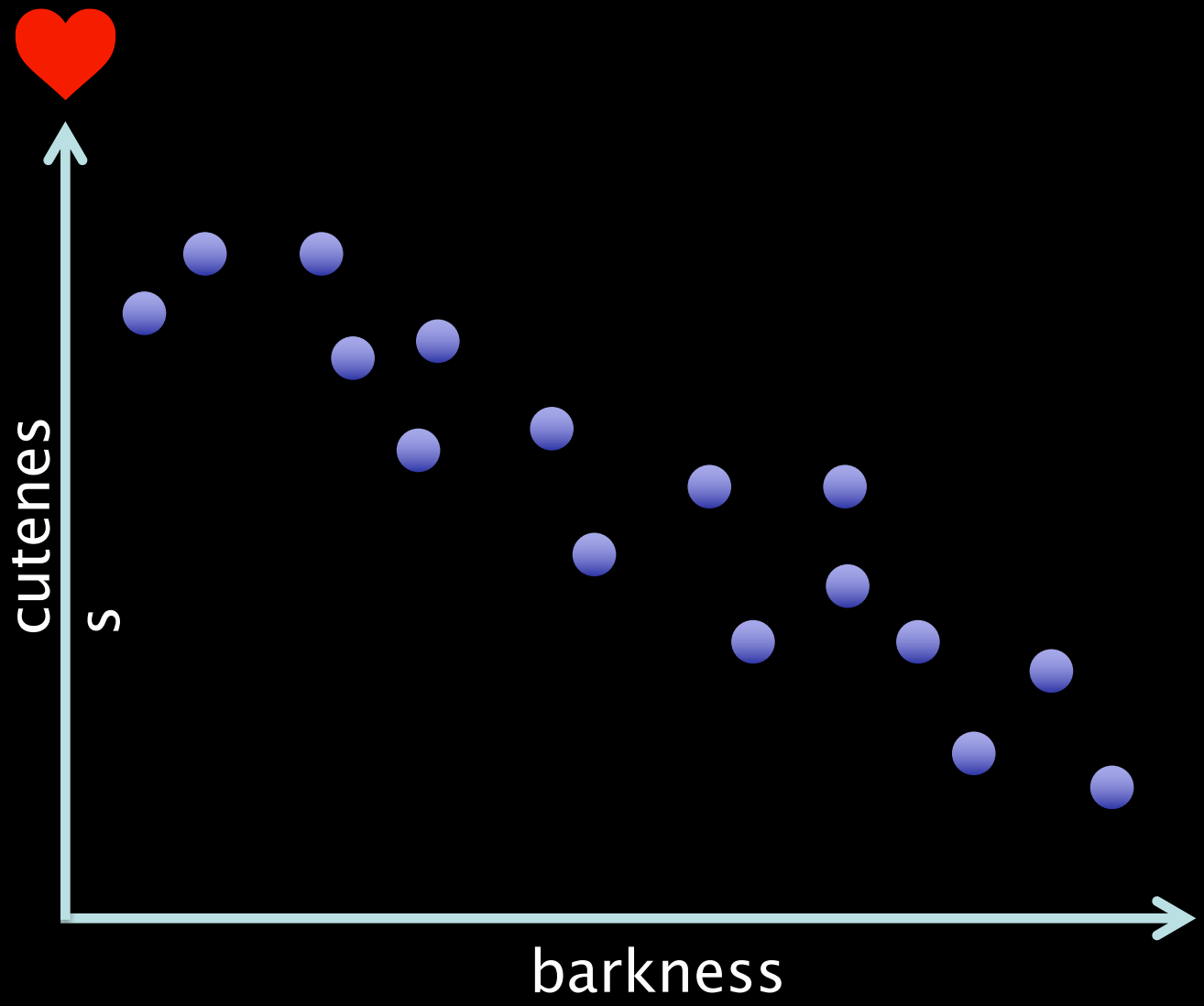
Regression Equation

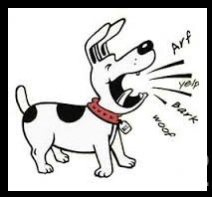
CUTENESS = **a** FURRINESS + **c**

a is the slope of the FURRINESS line —> positive value

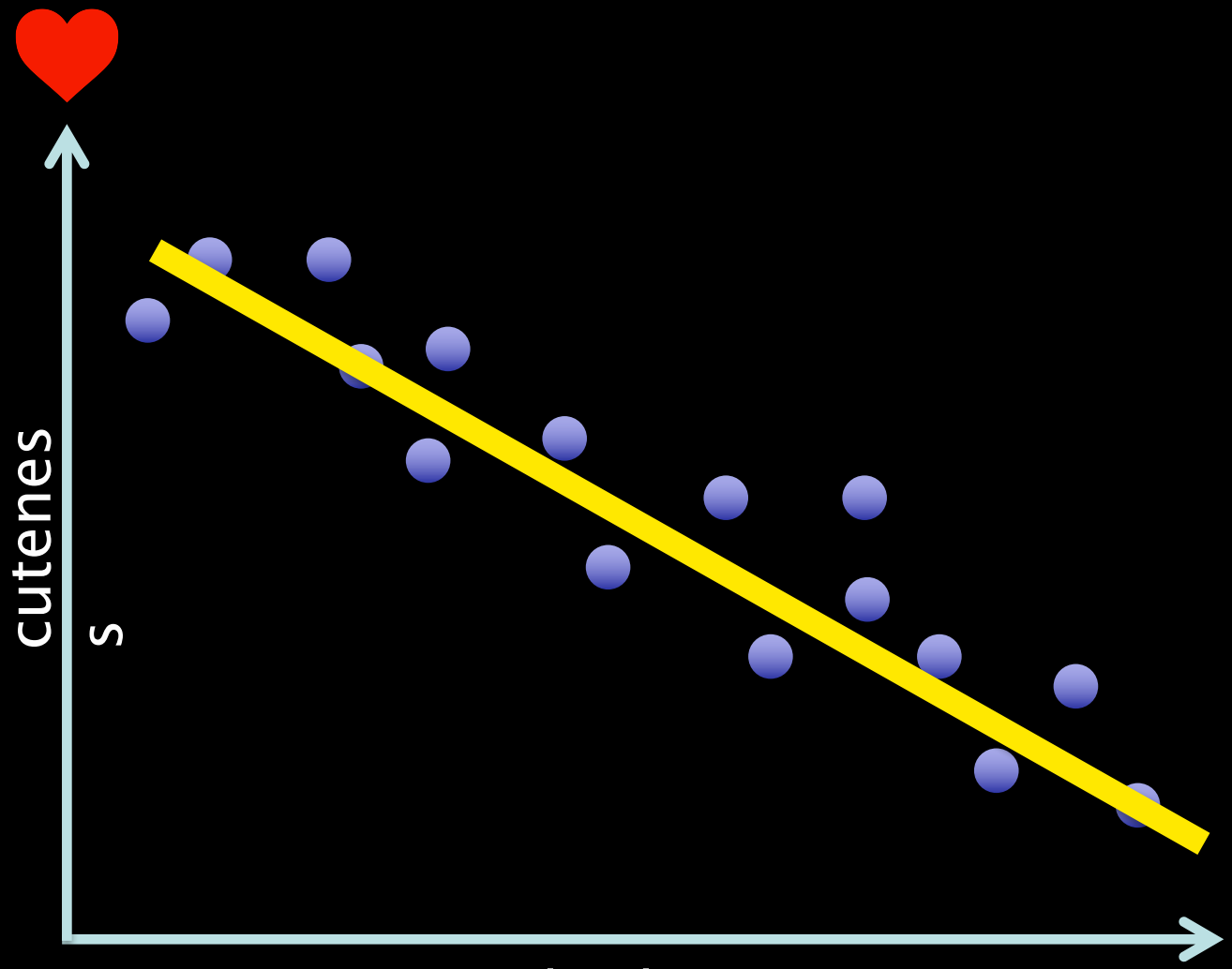
c is a constant

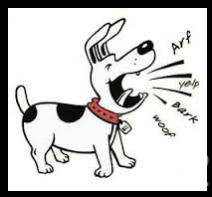
Adding a second feature..



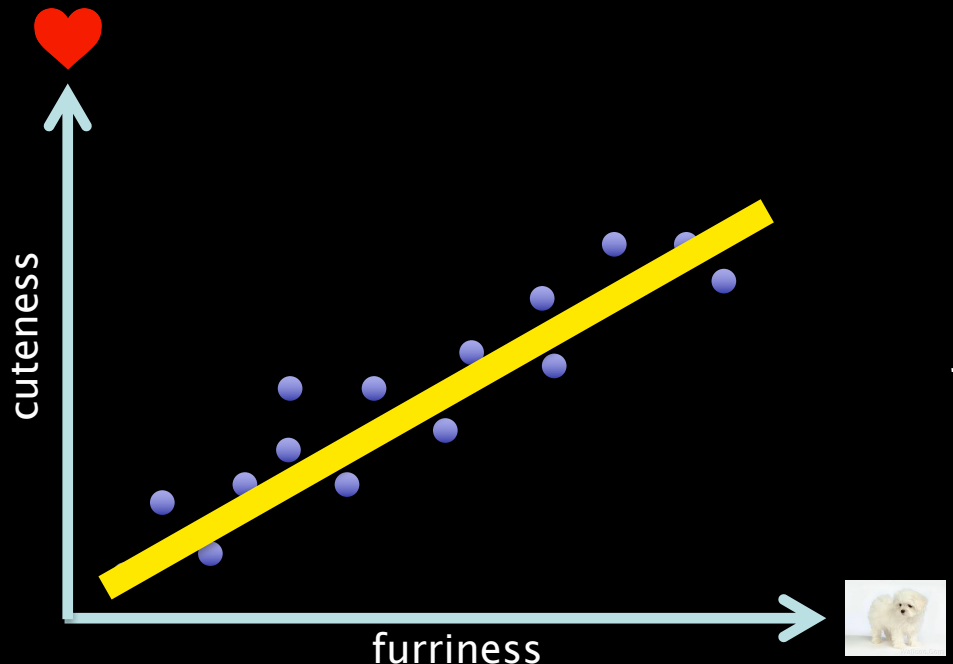


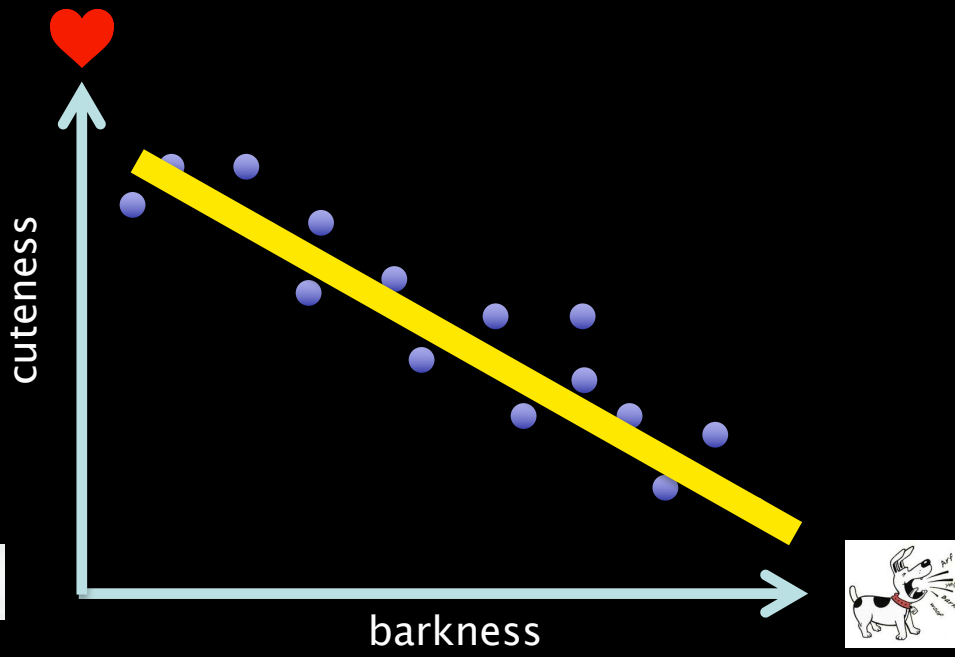
and a second model…





Multivariate regression



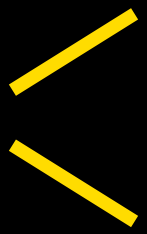


Regression Equation

CUTENESS = **a** FURRINESS + **b** BARKNESS + **c**

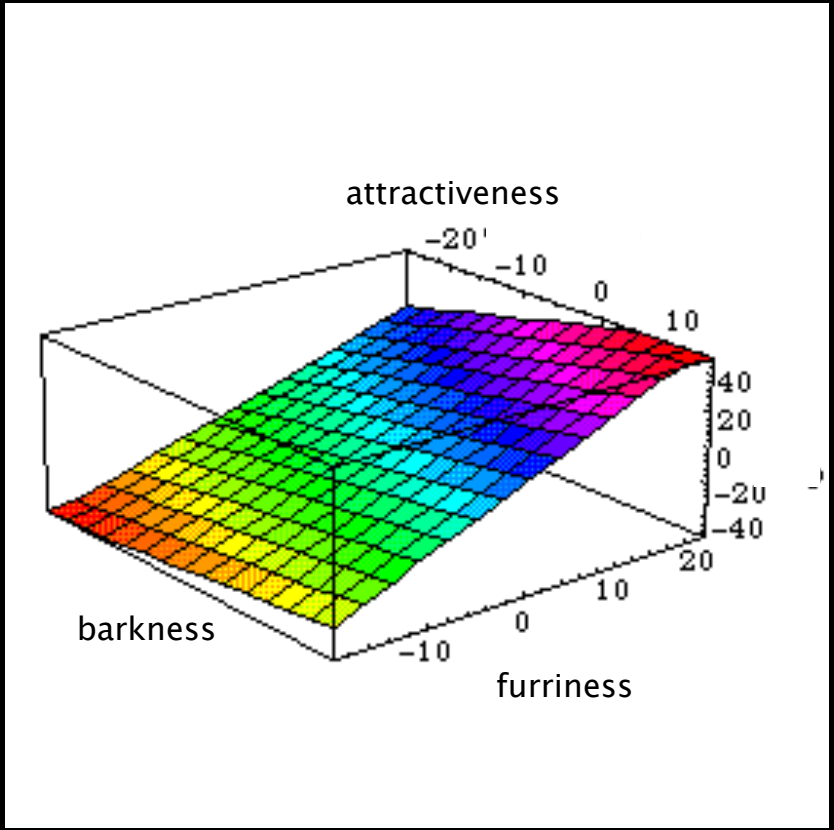
a is the slope of the FURRINESS line —> positive value

b is the slope of the BARKNESS line —> negative value



c is a constant

Multivariate regression surface can be automatically determined



In WEKA…

How to measure the prediction power?

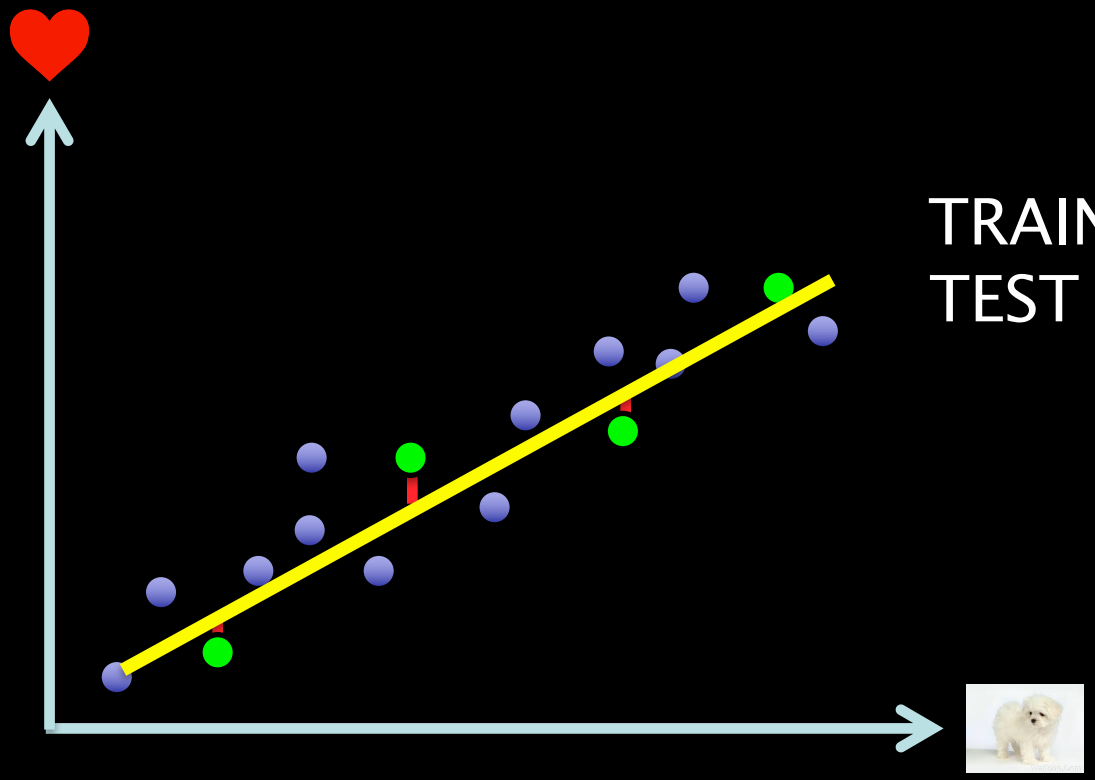
Linear regression is still model fitting, i.e., **description** rather than **prediction**

In other words: reproduction rather than prediction (and we are NOT interested in reproduction)

Prediction using Linear Regression

Fit the model to a subset of the data (training)

Test the predictions on the remaining data in a cross-validation procedure



WEKA

Linear Regression applied to Housing.arff…

www.cs.cmu.edu/afs/cs/academic/class/15381-s07/www/hw6/cal_housing.arff

```
=== Run information ===

Scheme:       weka.classifiers.functions.LinearRegression -S 0 -R 1.0E-8
Relation:     cal_housing
Instances:    20640
Attributes:   9
              longitude
              latitude
              housingMedianAge
              totalRooms
              totalBedrooms
              population
              households
              medianIncome
              medianHouseValue
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===


Linear Regression Model

medianHouseValue =

  -42823.7438 * longitude +
  -42576.7219 * latitude +
    1156.3039 * housingMedianAge +
      -8.1816 * totalRooms +
     113.4107 * totalBedrooms +
     -38.5351 * population +
      48.3083 * households +
   40248.5142 * medianIncome +
-3594022.942
```

label

weighted features

Note the very large coefficients!
What is the cause?

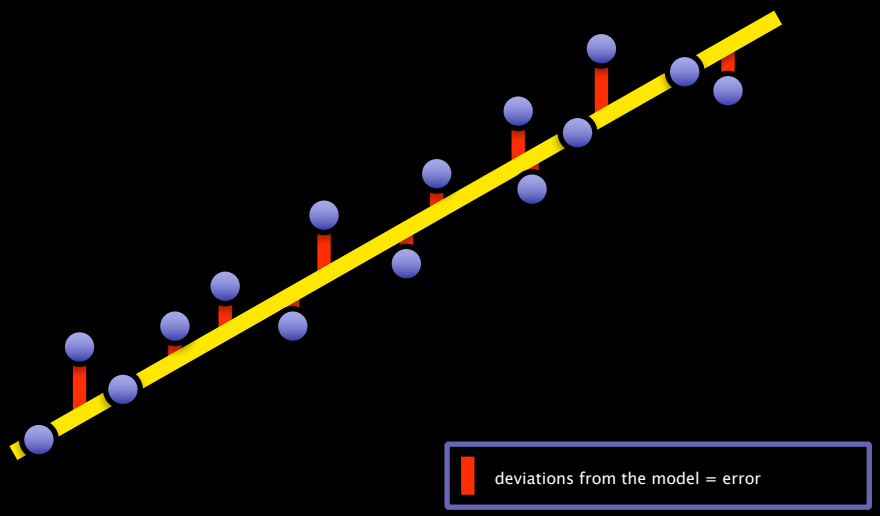
Results of WEKA's Linear Regression applied to Housing regression task:

```
=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.7974
Mean absolute error                  50806.1304
Root mean squared error              69637.9114
Relative absolute error                 55.7203 %
Root relative squared error             60.3415 %
Total Number of Instances            20640
```

MAE

MSE



MAE: Mean of the absolute deviations
MSE: Mean of the squared deviations

http://www.cs.cmu.edu/afs/cs/academic/class/15381-s07/www/hw6/cal_housing.arff

Model Complexity

in regression & classification

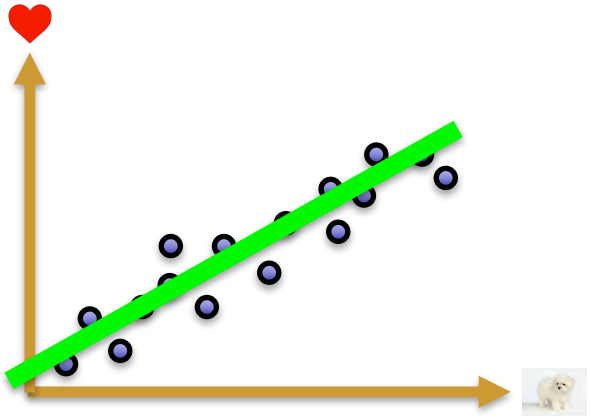
PLEASE NOTE!
Classification versus Regression (related to 2 types of scatterplots)

- In classification, the model induced from the data defines a decision boundary that **separates** the data described by 2 features into 2 classes (e.g., *cats* versus *dogs*) or more.

NOT NOW

separates the data

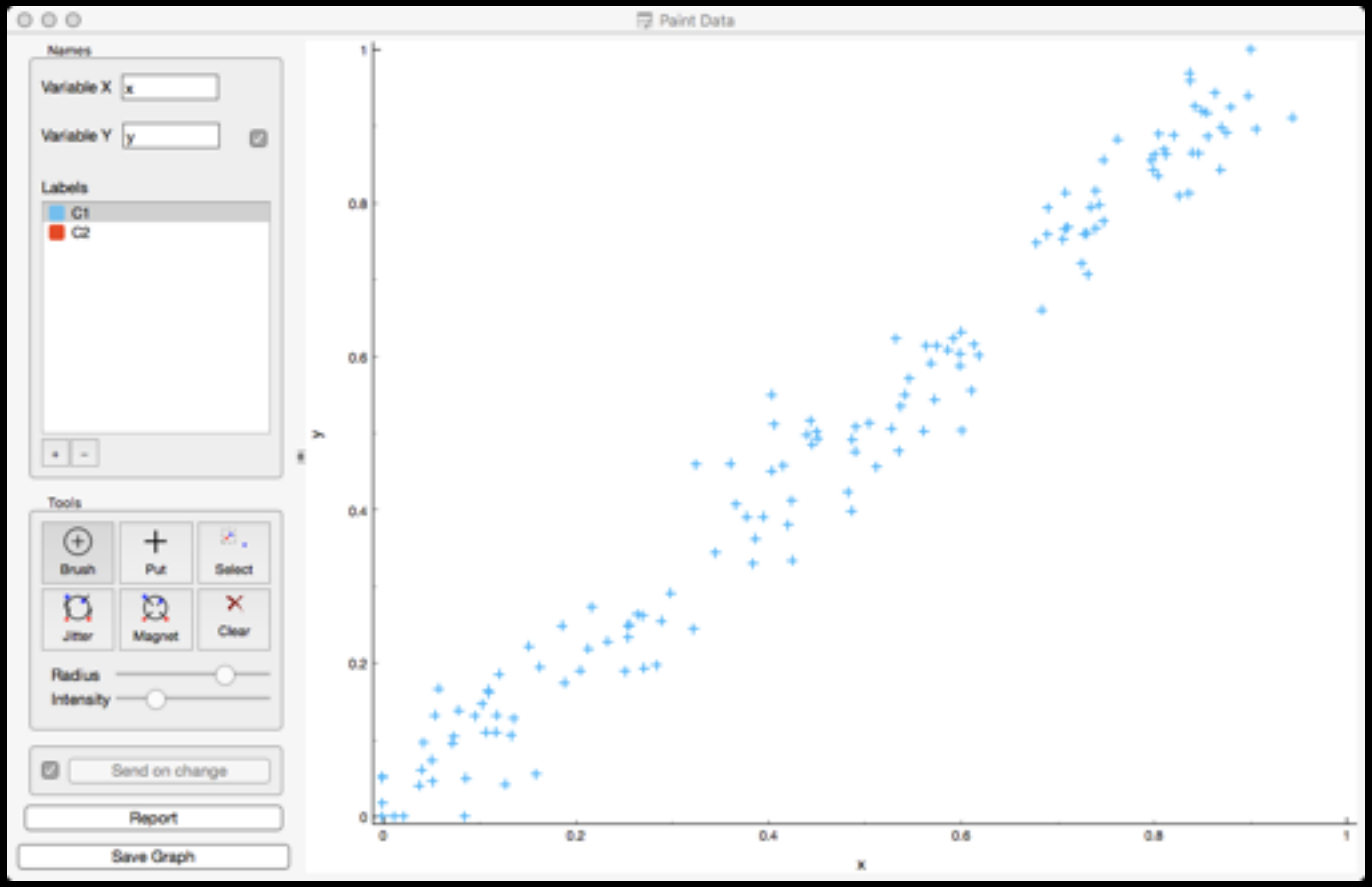
- In regression, the model induced from the data **fits** the data to describe the relation between 2 features or between a feature (e.g., *furriness*) and the label (e.g., *cuteness*)



fits the data

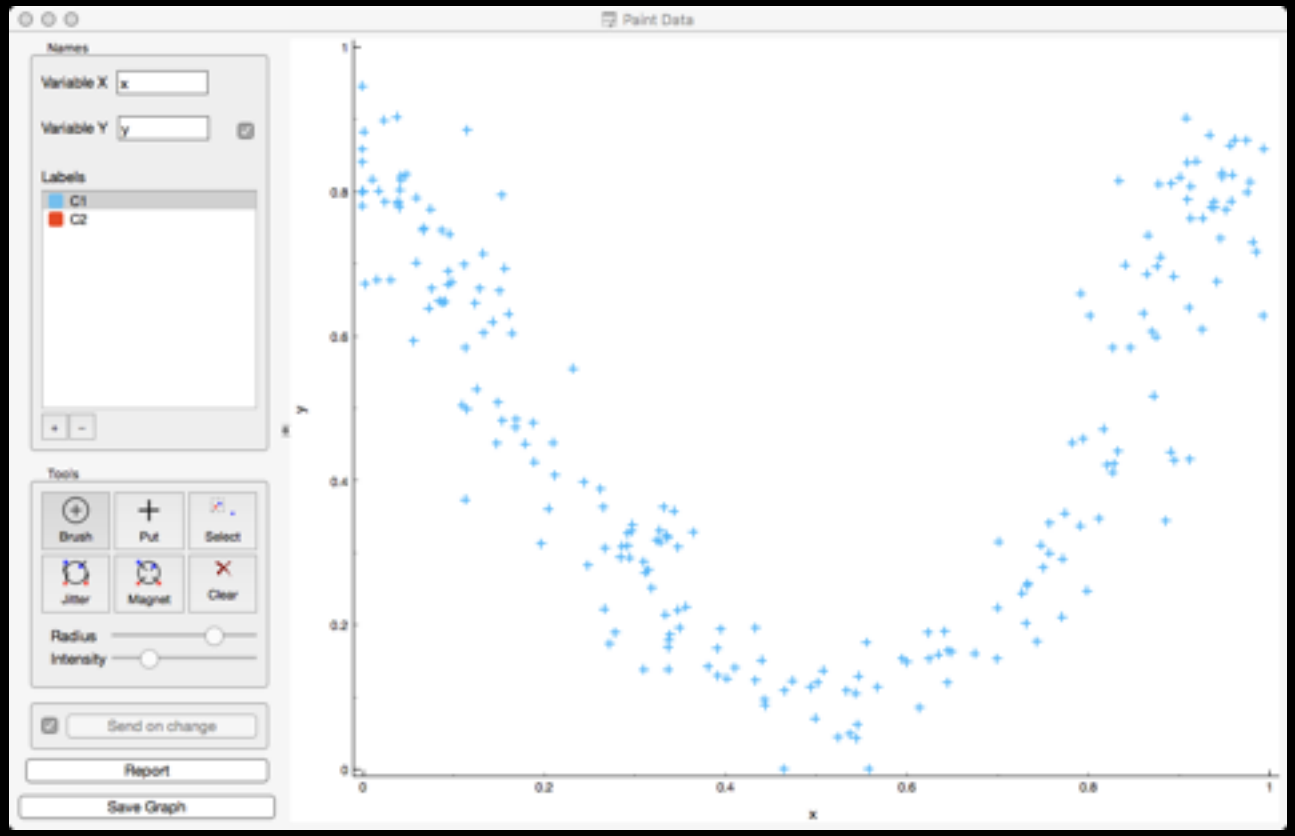
Regression

Case 1: Feature x, label y



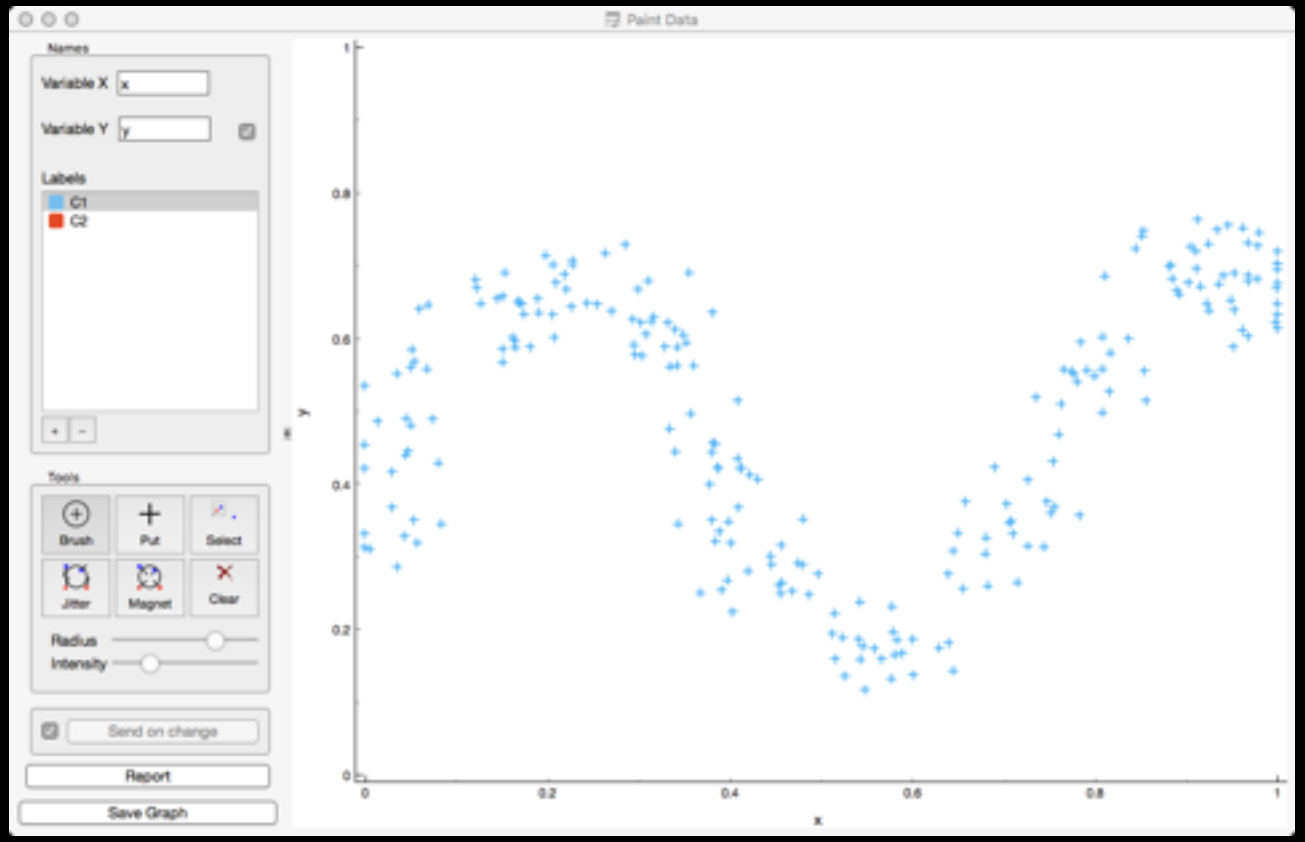
What is the “best” model?

Case 2: Feature x, label y



What is the “best” model?

Case 3: Feature x, label y



What is the “best” model?

Model fitting

A polynomial is a curve that has a parameter that varies its complexity from 1 (linear=straight line) to a large number (very wiggly line)

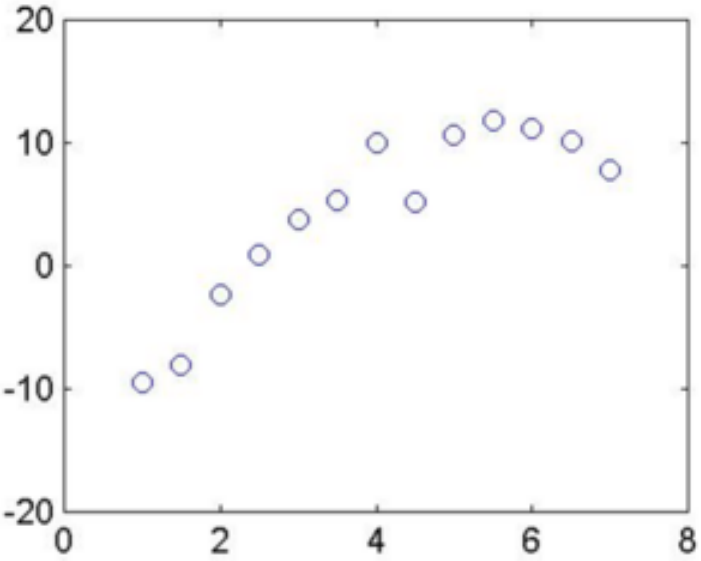
Fitting a model to data
3 situations

Underfitting:
model is too simple for the data

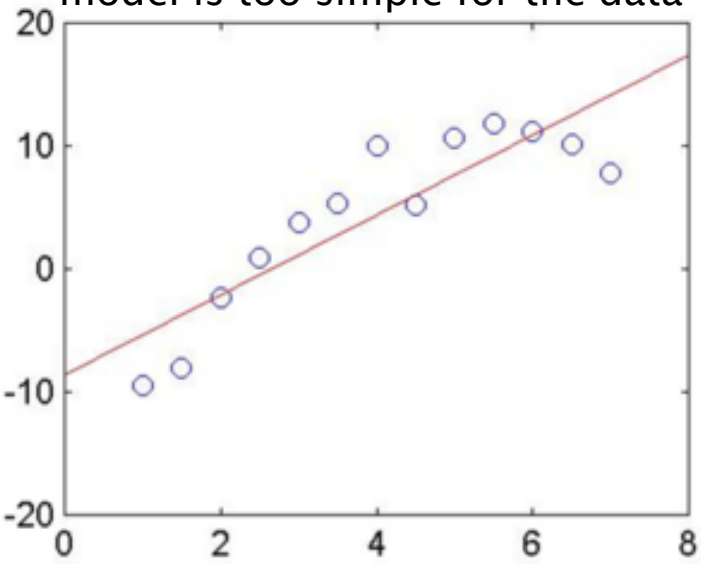
Overfitting:
model is too complex for the data

Best fit:
The simplest possible fitting model

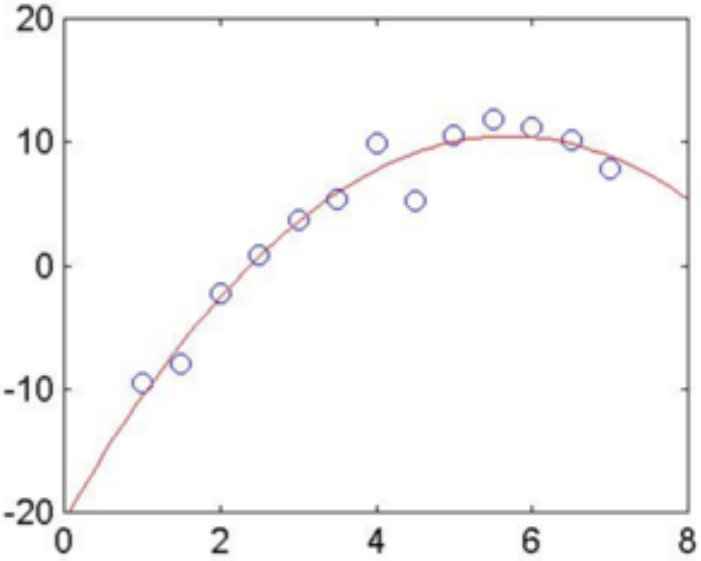
regression problem



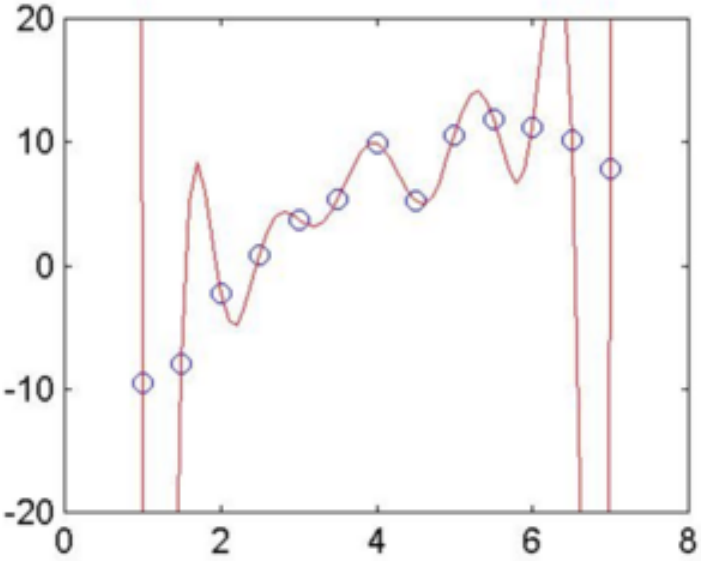
underfitting
model is too simple for the data



best fitting
model fits the data



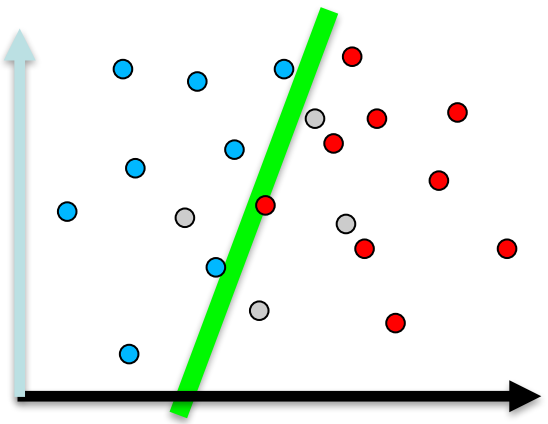
overfitting
model is too complex for the data



Classification

PLEASE NOTE!
Classification versus Regression (related to 2 types of scatterplots)

- In classification, the model induced from the data defines a decision boundary that **separates** the data described by 2 features into 2 classes (e.g., *cats* versus *dogs*) or more.



separates the data

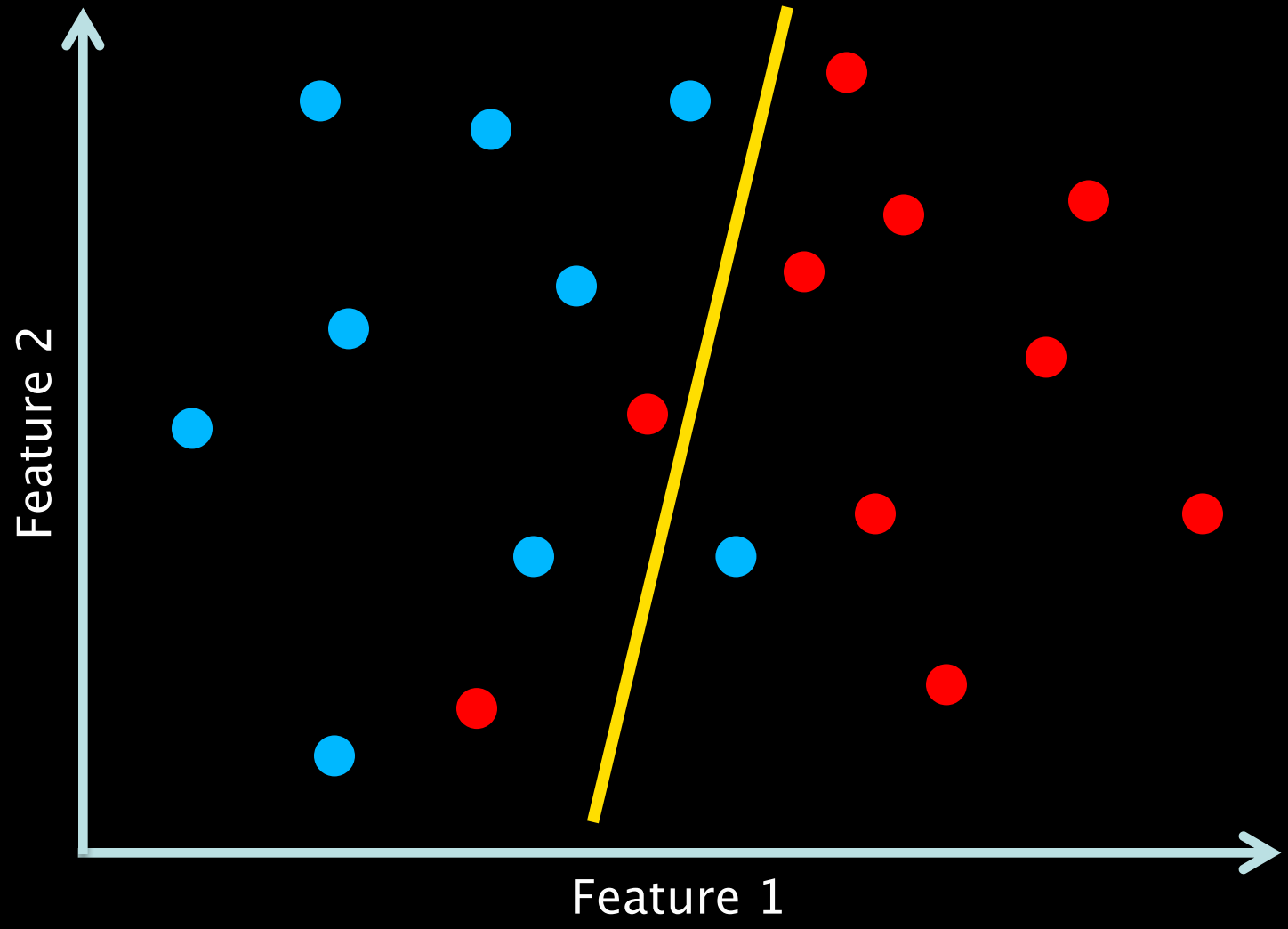
- In regression, the model induced from the data **fits** the data to describe the relation between 2 features or between a feature (e.g., *furriness*) and the label (e.g., *cuteness*)

NOT NOW

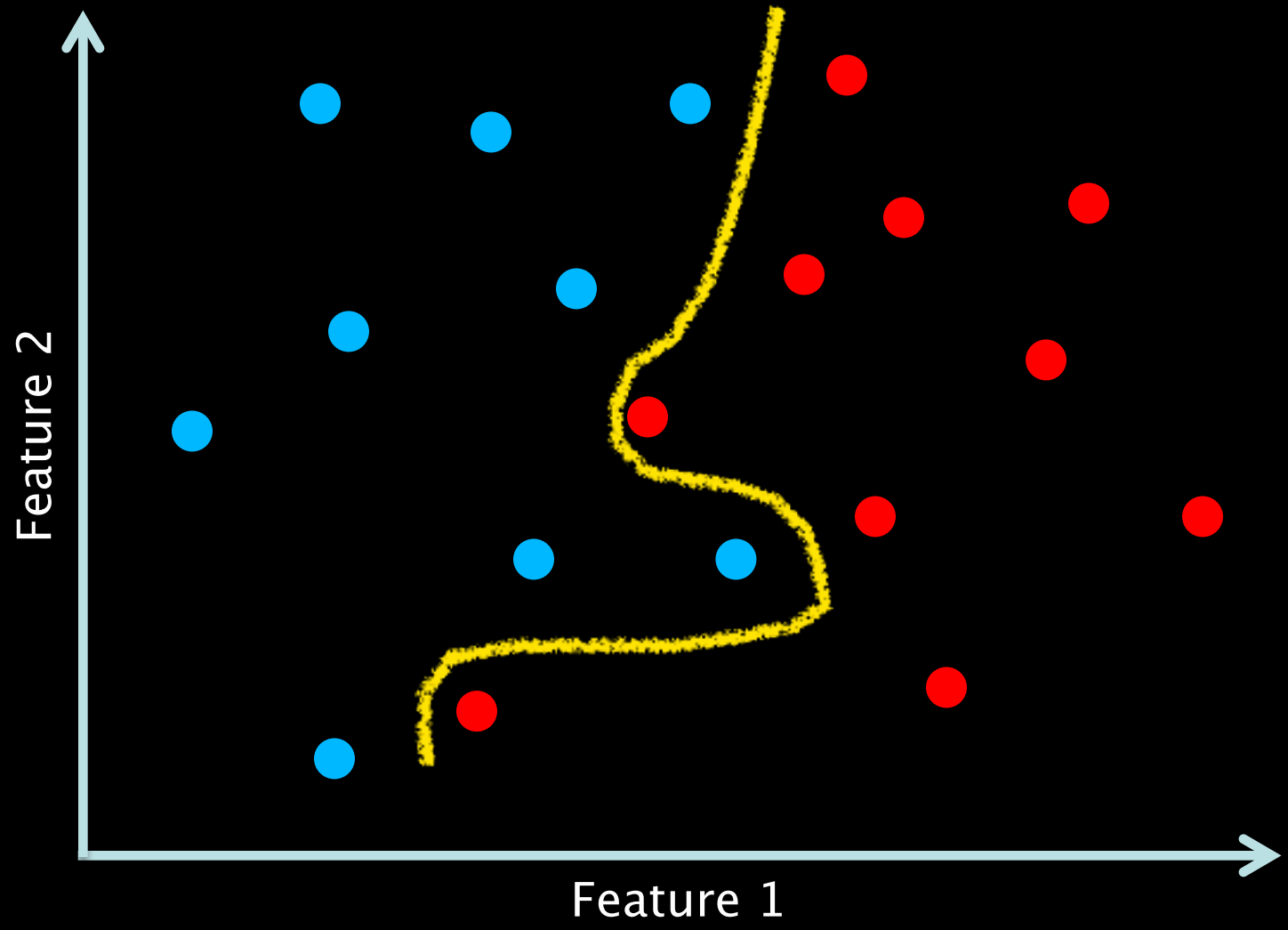
fits the data

TILBURG UNIVERSITY

Separation by a simple model (– =decision boundary)

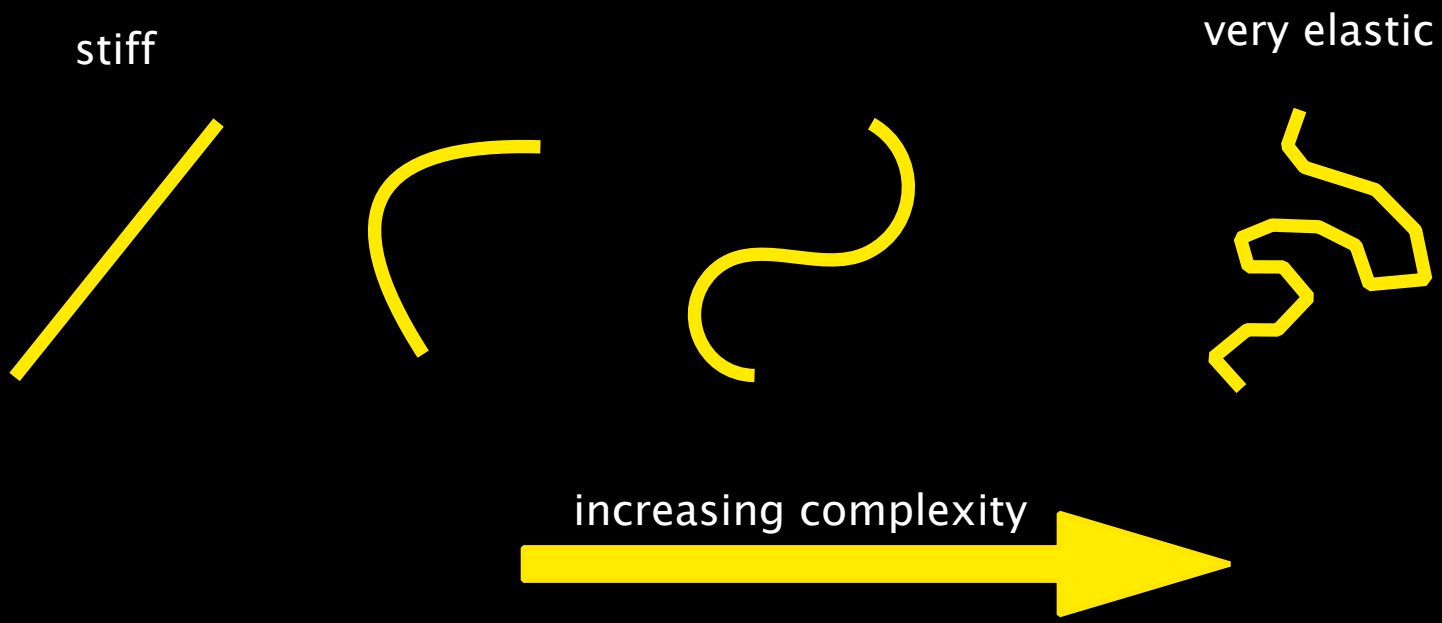


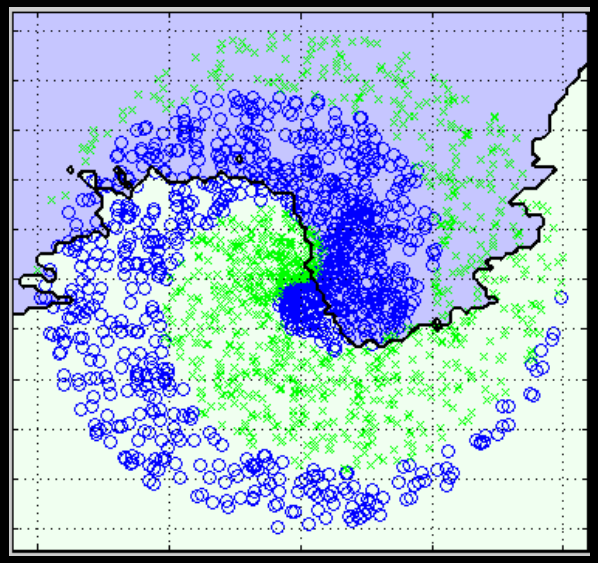
Separation by a complex model (– =decision boundary)



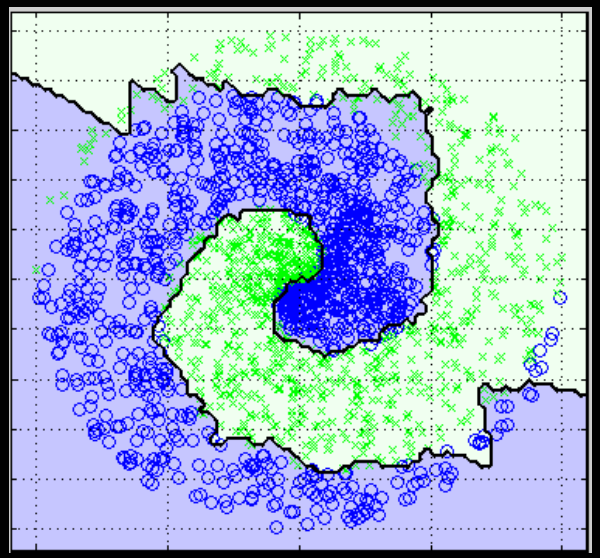
In classification…

- The complexity of a model is the flexibility of the decision boundary (could be a polynomial or something else)

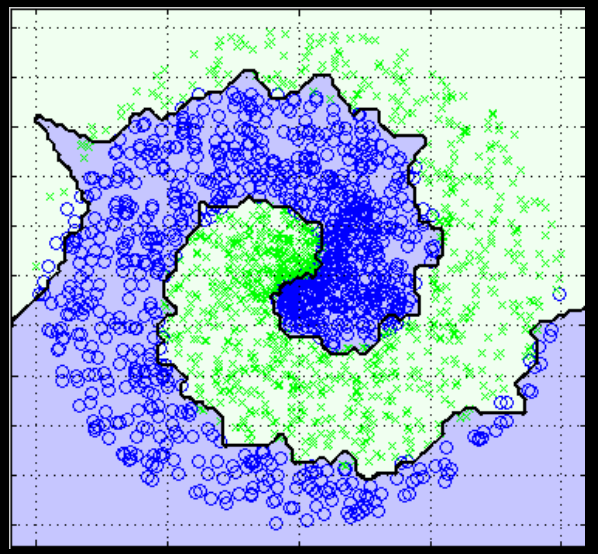




k = 100



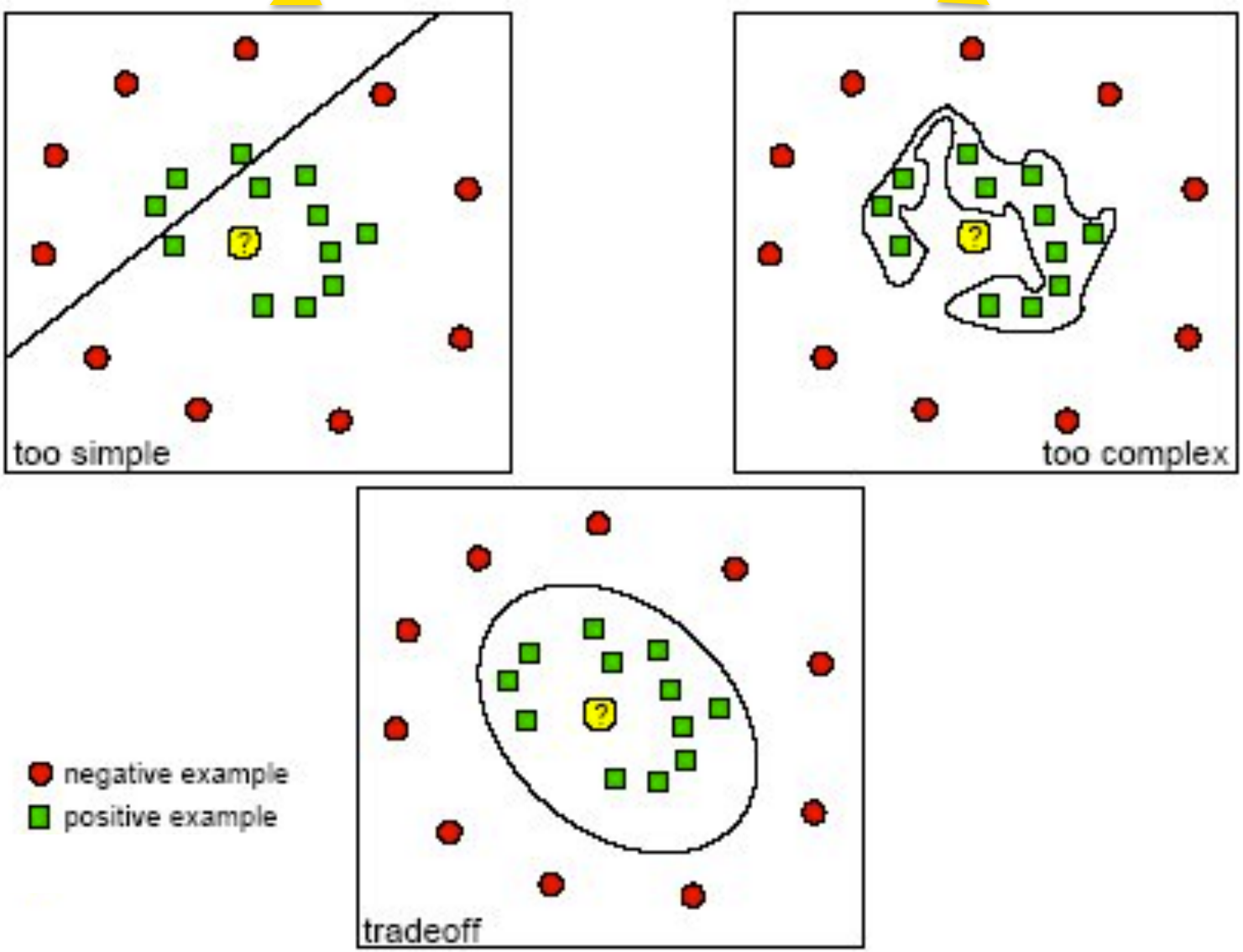
k = 10



k = 1

Increasing model complexity

Underfitting and Overfitting



How to determine model complexity?

- Depends on complexity of the separation between the classes
- Start with the simplest model (large k in kNN), and increase complexity (smaller k) or vice versa
- The simplest model is always preferred! Why?

Required Reading

WEKA book:

Chapter 4, Section 4.7 Instance-based learning

Chapter 5, Intro + Sections 5.1 – 5.4
(training & testing, cross validation)

Chapter 6, Intro + Sections 6.1 and 6.5

(decision trees and instance based learning)