ML4Bio
Lecture #1: Introduction

February 24th, 2016

Quaid Morris

Course goals

- Practical introduction to ML
  - Having a basic grounding in the terminology and important concepts in ML; to permit self-study,
  - Being able to select the right method for your problems,
  - Being able to use the multitude of ML tools and methods in R,
  - Being able to troubleshoot problems with tools,
  - Having a foundation to learn other tools: Python's scikit-learn, TensorFlow/Torch/Theano

How this course works

- Course website: Google "ML4BIO Alan Moses"
- Course email: (but email me if you have questions)
- Four problem sets, 25% of your grade
- Programming in R (other languages possible*, but unsupported)
- Two tutorials: linear algebra review (March 1st), intro to R(March 8th), details on website.

Outline

- Overview of ML
- Overfitting
- Cross-validation
- Measuring success

Some slides adapted from:

Probabilistic Modelling and
Bayesian Inference

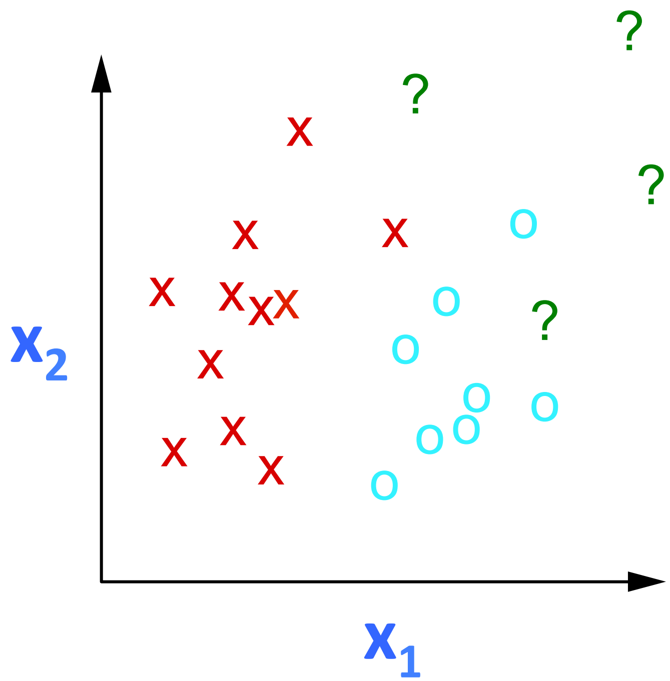
Zoubin Ghahramani

Department of Engineering
University of Cambridge, UK

zoubin@eng.cam.ac.uk
http://learning.eng.cam.ac.uk/zoubin/

MLSS Tübingen Lectures
2013

Classification example



e.g. is this cell a T-cell?

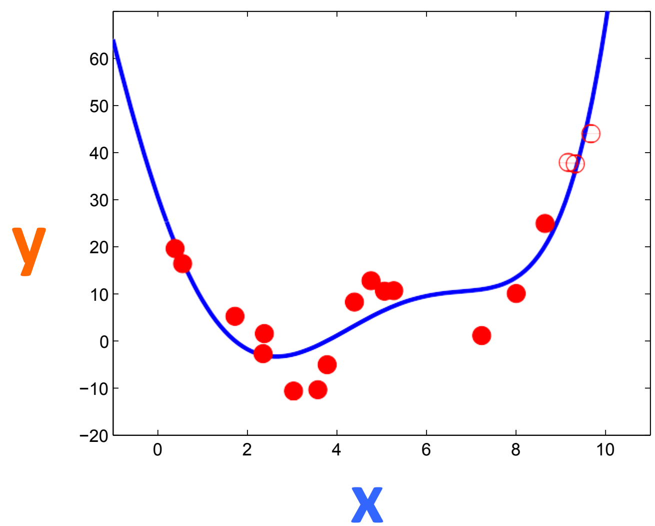
What is the correct label for the ?'s?

How certain am I?

How does the label depend on x?

Image from Z. Ghahramani

Regression example

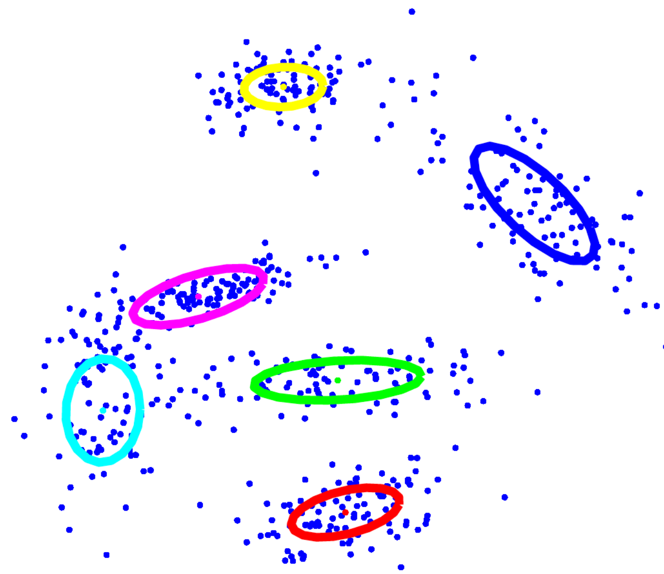


e.g. what is the temperature at this time of the year?

What is the relationship between x and y?
Given a new value of x, what's my best guess at y?
What is the range of variability in y for a given x?

Image from Z. Ghahramani

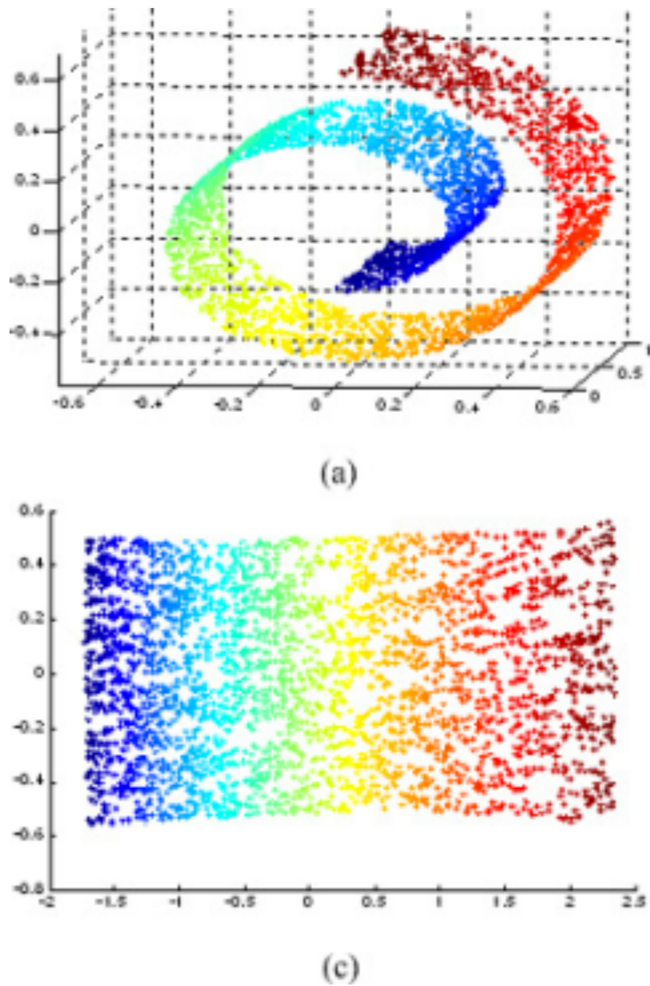
Clustering example



Are a given pair of datapoints from the same cluster?
How many clusters are there?
What are the characteristics of individual clusters?
Are there outliers?
How certain am I of the answers to the above questions?

Image from Z. Ghahramani

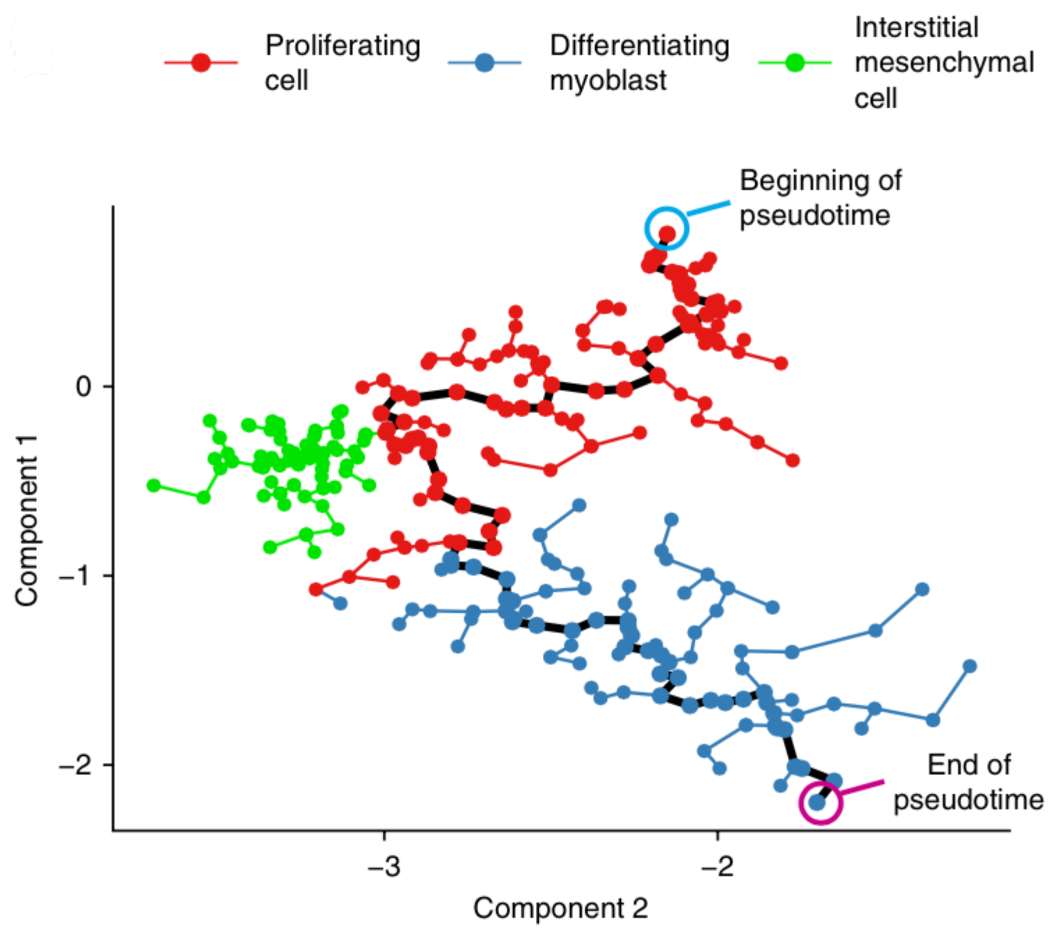
Dimensionality reduction example



Do the datapoints lie on a lower dimensional "*manifold*"?
If so, what is the dimensionality?
How far apart are two datapoints, if you can only travel on the manifold?

Image from PMID: 22595000

Dimensionality reduction example II



From “monocle”: Trapnell et al, NatBio 2014

How do to ML

- Four parts:
  1. Data D – describes the machine learning problem
  2. Model – defines the parameters Θ and describes how the data D depends on them
  3. Objective function E(Θ, D) – scores Θ for a given dataset D
  4. Optimization method – finds high scoring values of Θ

The Data

Supervised learning:

e.g. deep learning, random forests, SVMs

$$\mathbf{D} = \{(\mathbf{x}^{(1)},\mathbf{y}^{(1)}), (\mathbf{x}^{(2)},\mathbf{y}^{(2)}), \ldots, (\mathbf{x}^{(N)},\mathbf{y}^{(N)})\}$$

Inputs (aka features)

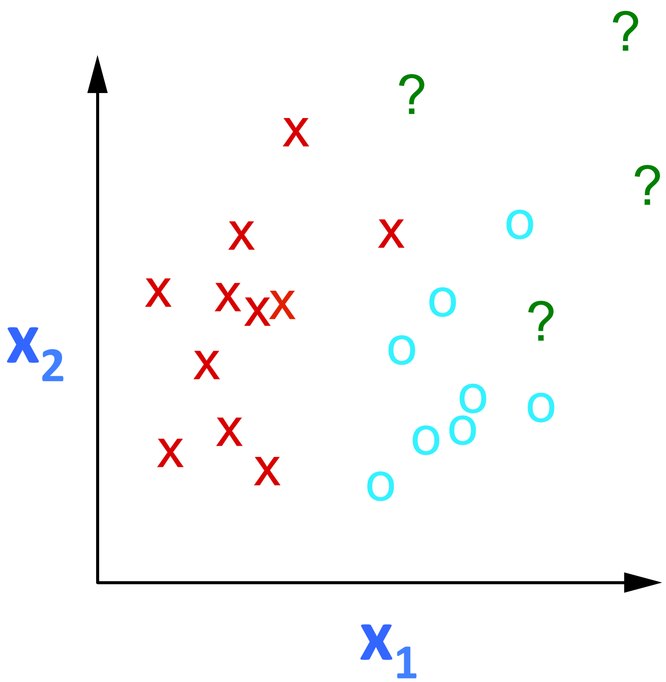
Targets (regression) or labels (classification)

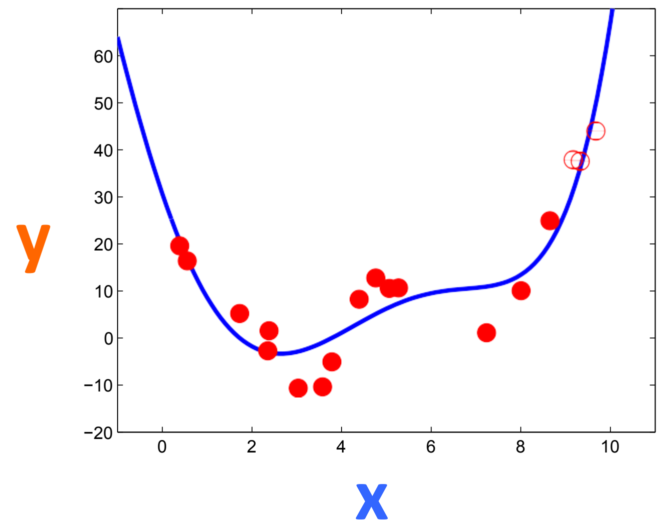
Unsupervised learning:

e.g. clustering, PCA, dimensionality reduction

$$\mathbf{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$$

Data: supervised learning





$y^{(n)}$ is a continuous value, aka a real number

- $y^{(n)}$ is a categorical value, sometimes called "discrete"

- If $y^{(n)}$ is either X or O: binary classification
- If $y^{(n)}$ is, e.g., either X, O, or +: multiclass classification
- If $y^{(n)}$ is, e.g., either (X, X), (O, X), (X, O) or (O,O): multilabel classification

Data: unsupervised learning

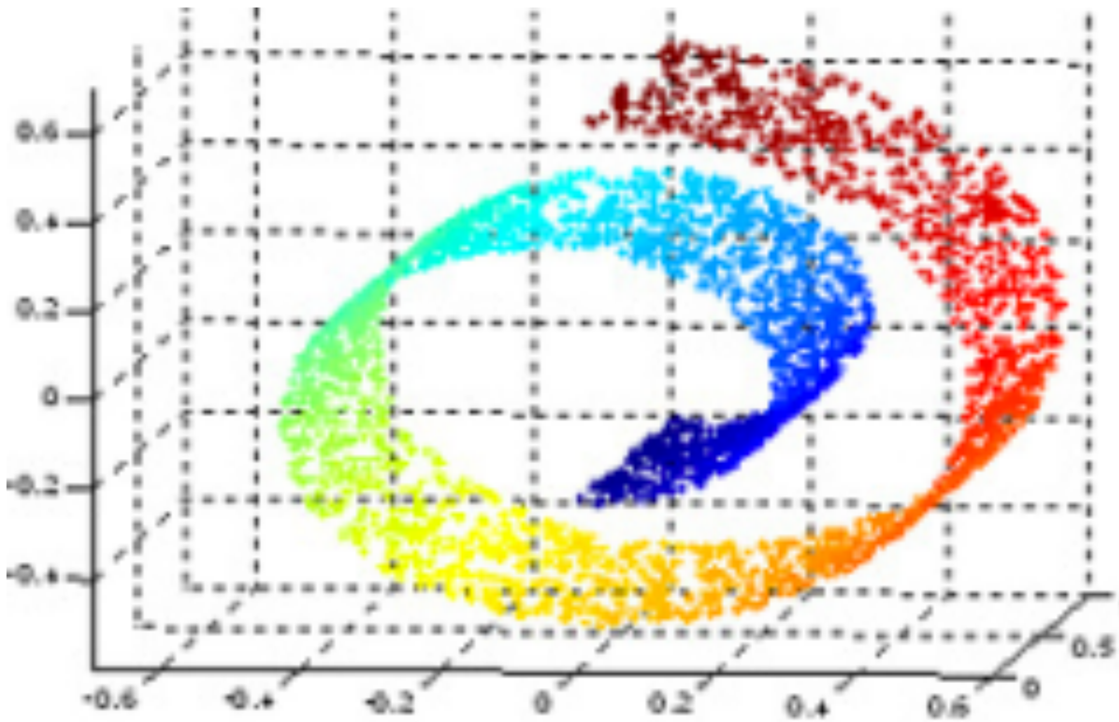


Image from PMID: 22595000

The Model

- A formal description of how the data depends on the parameter.
- E.g. linear regression. Data: inputs x, and target values y

Model's prediction, aka output: "y hat", this is compared to *target value "y"*

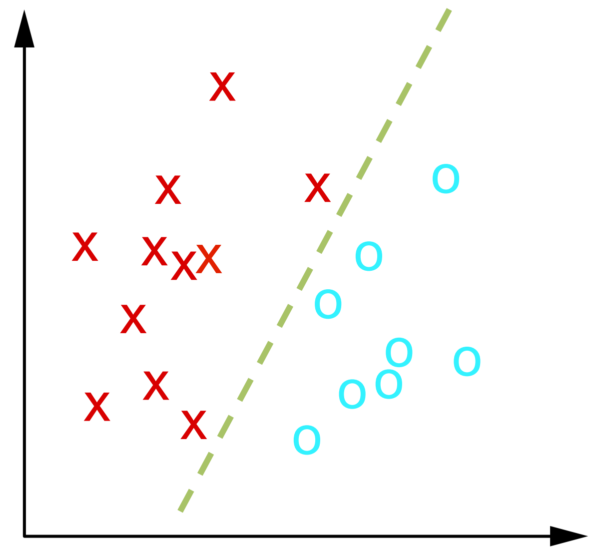
$$\hat{y} = \alpha x + \beta$$

model parameters

Inputs aka *features*

Goal: set α and β so that $\hat{y}^{(n)}$ is as close as possible to $y^{(n)}$ for a given $x^{(n)}$ for all *i*

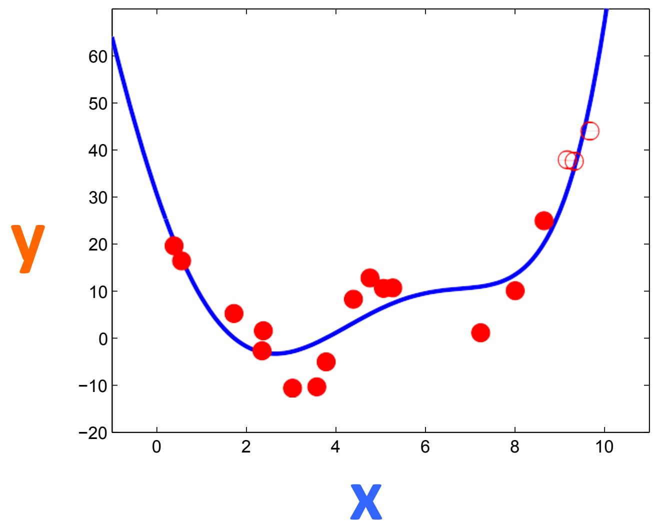
e.g. Classification



Model (logistic regression):

1. the "direction" of the classification boundary, given by a vector $\boldsymbol{\omega}$
2. A scalar value, γ, indicating how quickly confidence changes as we move away from the boundary

e.g. Polynomial regression



Model (e.g. fifth order polynomial):

$$\hat{y} = \theta_5 x^5 + \theta_4 x^4 + \theta_3 x^3 + \theta_2 x^2 + \theta_1 x^1 + \theta_0$$

The Objective function

$E(\Theta, D)$ or just $E(\Theta)$

- Depends on both the data D and the parameters Θ,
- Measures fit between the model's predictions and the data,
- Often contains a term to penalize "complex models", sometimes known as regularization
- D is fixed, goal is to optimize function with respect to Θ,
- Also known as: cost function, error function
- Examples: sum of squared errors (SSE):

$$E(\alpha, \beta) = \Sigma_n (y^{(n)} - \hat{y}^{(n)})^2$$

The Optimization Method

$$E(\alpha, \beta) = \Sigma_n \left(y^{(n)} - \hat{y}^{(n)}\right)^2$$

- E.g. try all values of α, β on a grid until, choose the best ones. (brute force)
- Take the partial derivatives of E with respect to α, β and find the critical points where they are zero, determine which are minima. (analytical)
- Start at a random point, follow the gradient of the function to a (local) minimum (gradient descent)

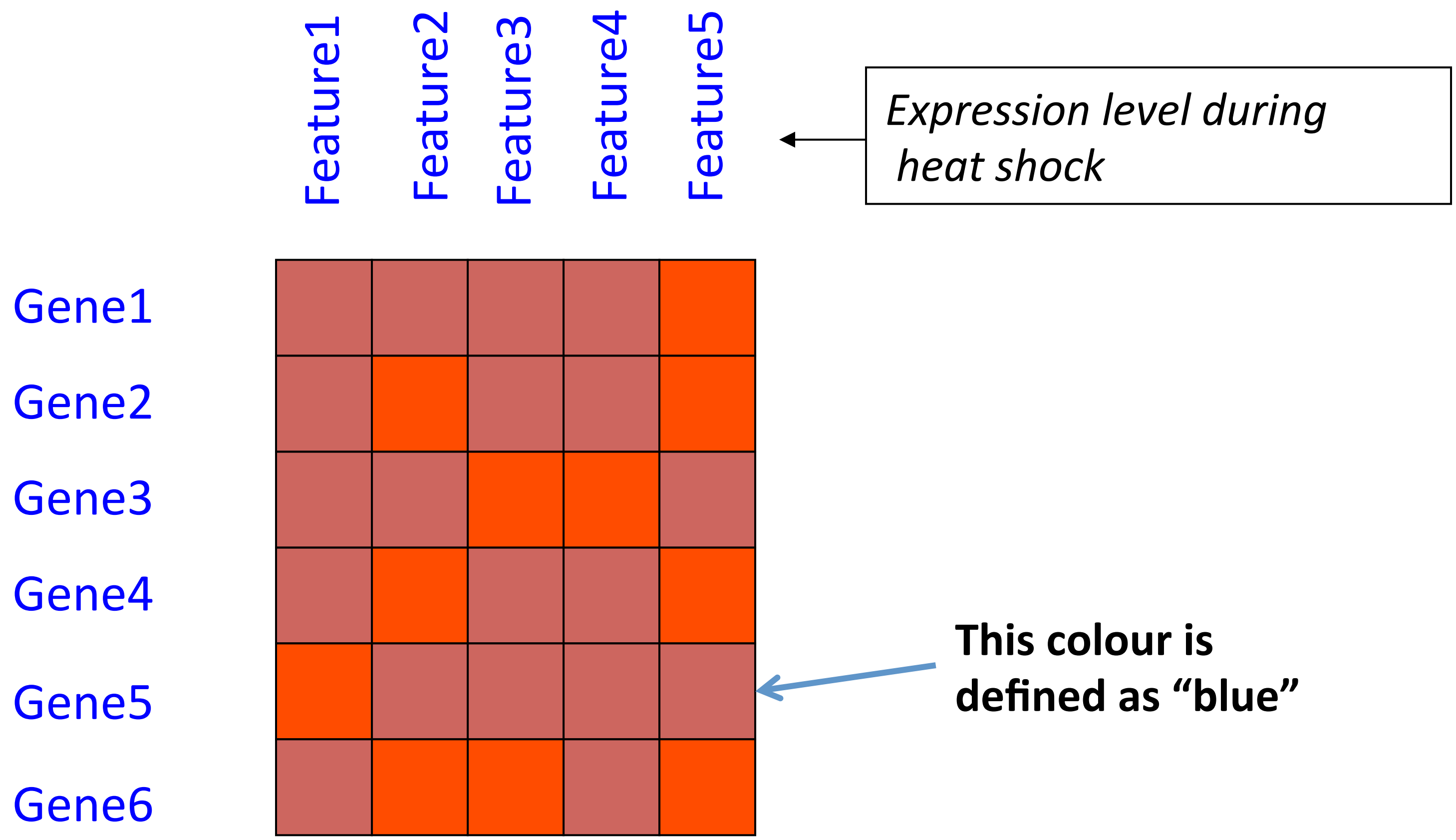
Important questions

1. Data – Is the data appropriate for my learning task? Do I have enough data? Are my training data representative? Is there a selection bias in my data?
2. Model – Is my model sufficiently complex to learn the task? Is overfitting a concern? Can I interpret the parameters or the model?
3. Objective function – Does my objective function score errors appropriately? Is it too sensitive to outliers? Is it properly regularized?
4. Optimization method – Will my optimization method find good solutions? Does it get stuck in suboptimal solution (aka local minima)? Am I using a method matched to the objective function? Is it fast enough?

Important concepts

- Training and test sets
- Uncertainty about classification
- Overfitting
- Cross-validation (leave-one-out)

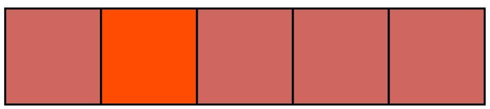
Put yourself in the machine’s shoes



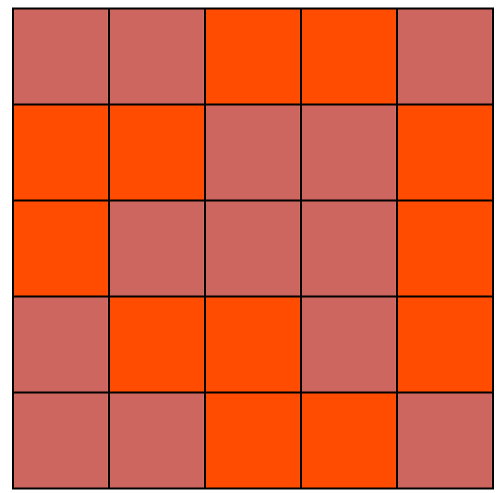
Which uncharacterized genes are involved in tRNA processing?

Training

Positives



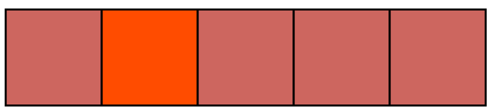
Negatives



What pattern distinguishes the positives and the negatives?

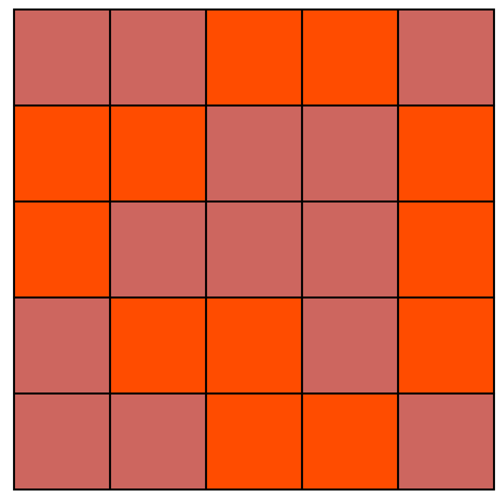
Training

Positives



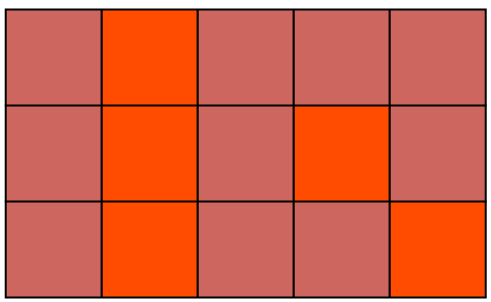
- 4 blue features
- features 1,3, and 5 are blue
- features 1 and 3 are blue and feature 2 is red
- features 1 and 3 are blue

Negatives



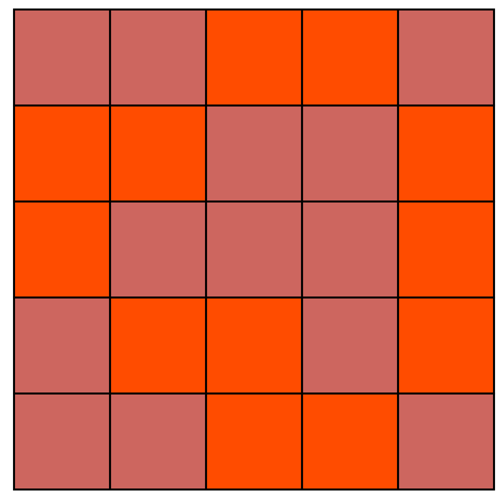
Training

Positives



- features 1 and 3 are blue and
  feature 2 is red
- features 1 and 3 are blue

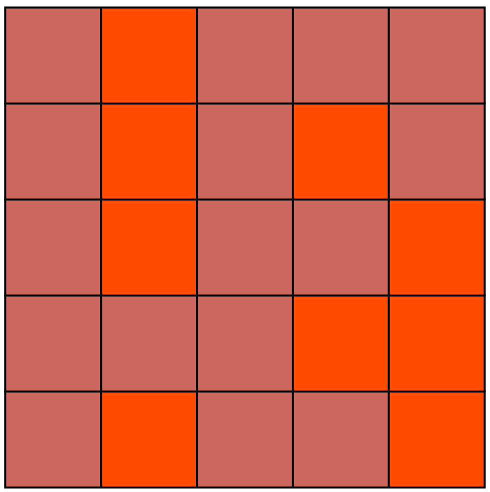
Negatives



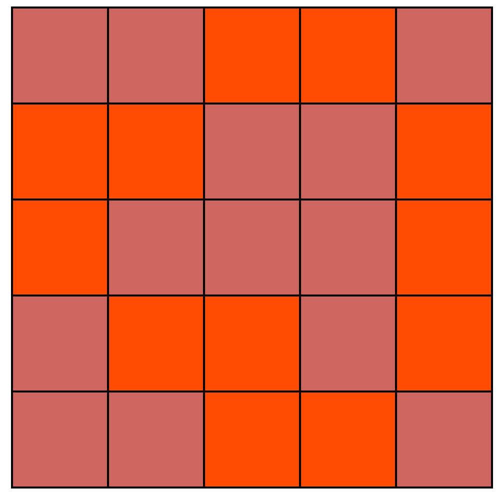
Known genes

Training

Positives



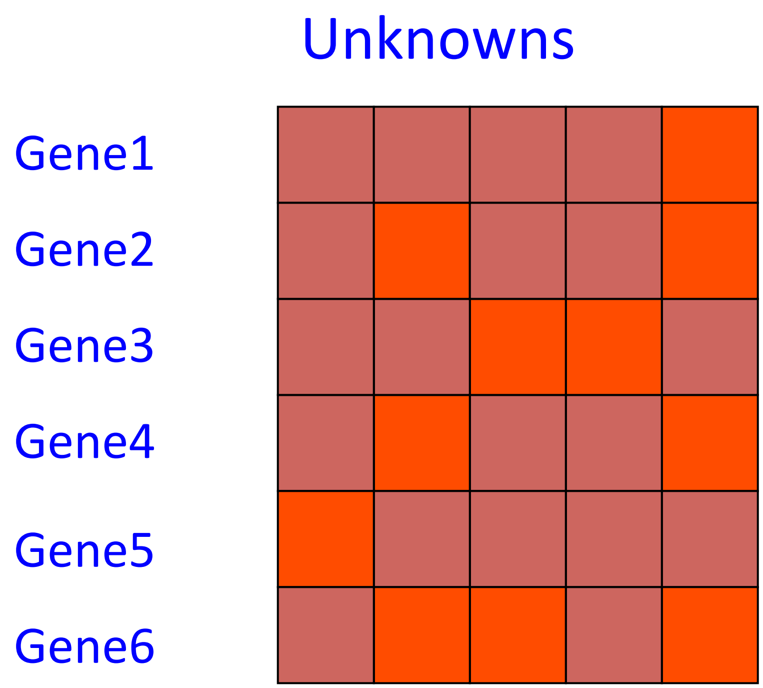
Negatives



- features 1 and 3 are blue

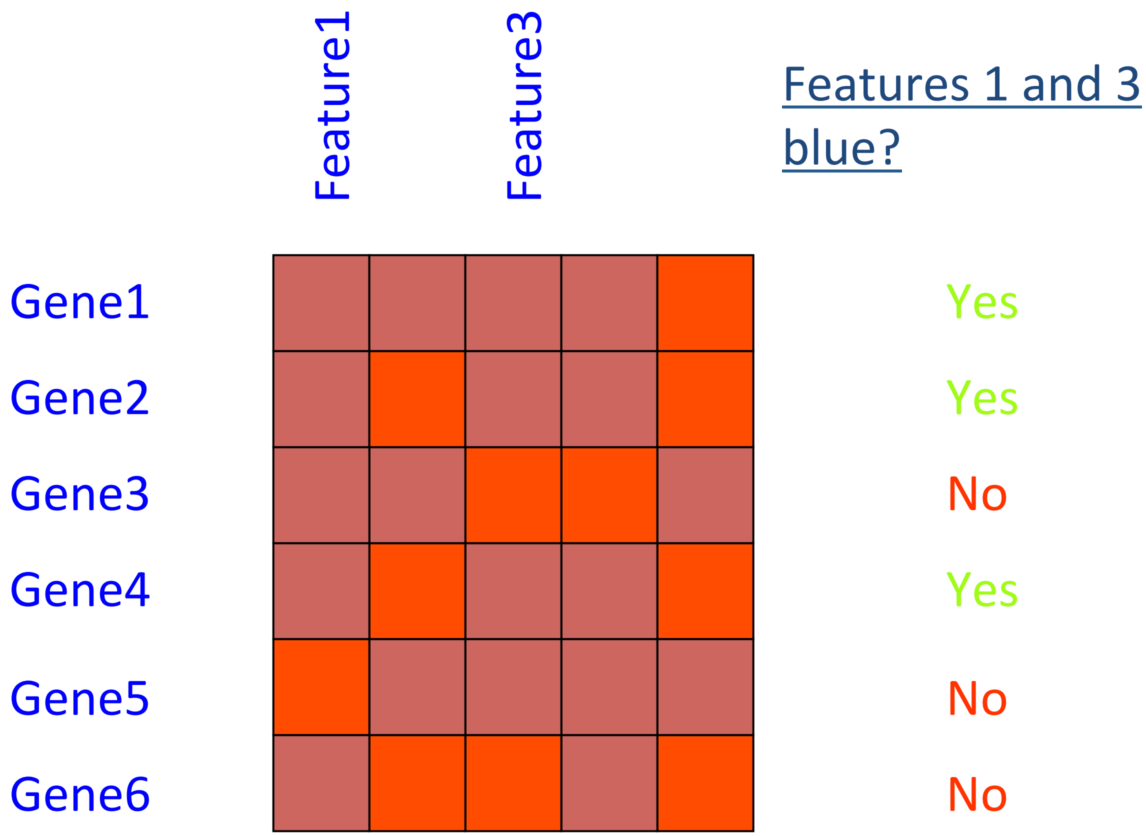
Known genes

Prediction



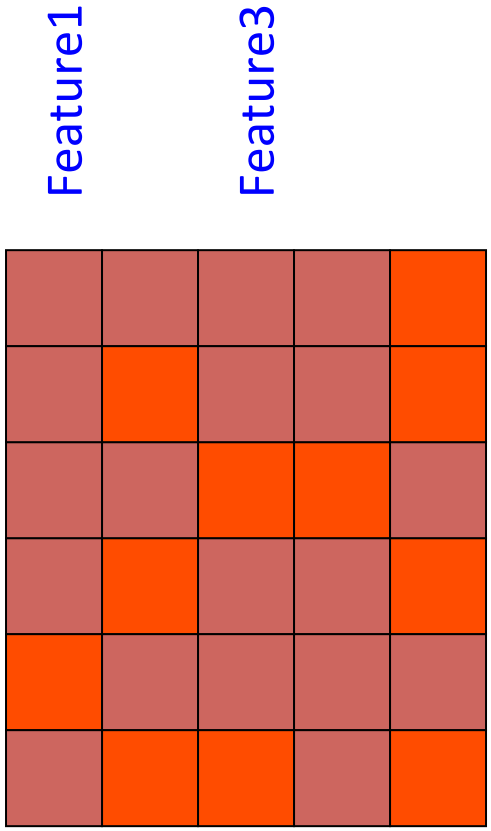
Which genes are involved in tRNA processing?

Prediction



Which genes are involved in tRNA processing?

Prediction

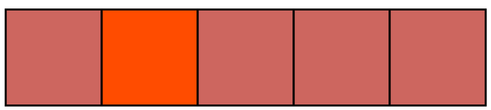


	Features 1 and 3 blue?	Prediction:	Assay:
Gene1	Yes	Involved	+
Gene2	Yes	Involved	+
Gene3	No	Not Involved	-
Gene4	Yes	Involved	+
Gene5	No	Not Involved	-
Gene6	No	Not Involved	-

Which genes are involved in tRNA processing?

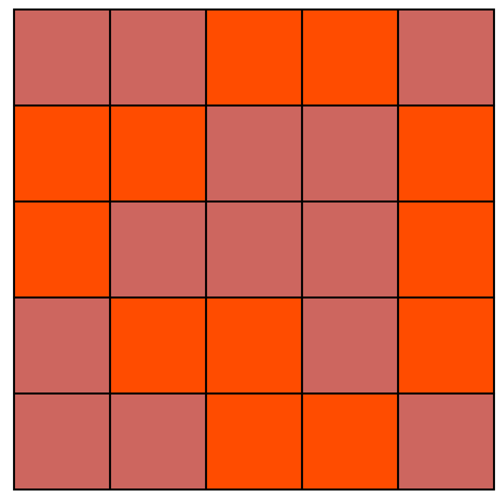
Training under sparse annotation

Positives



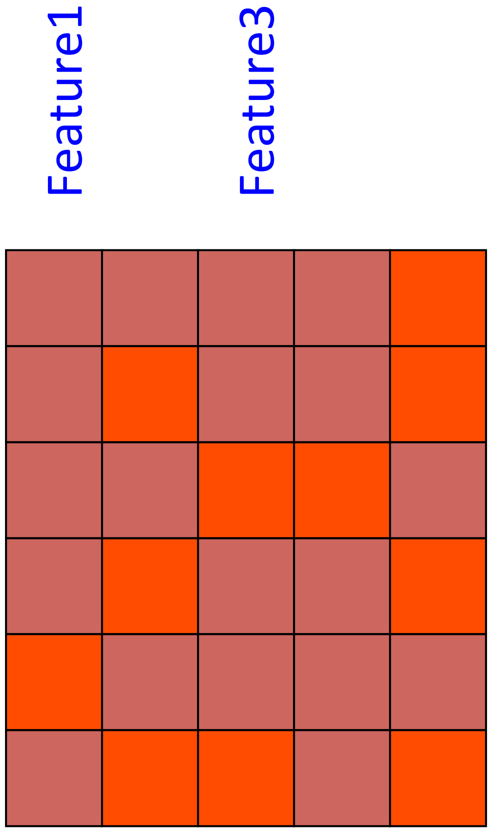
- 4 blue features
- features 1 and 3 are blue

Negatives



What pattern distinguishes the positives and the negatives?

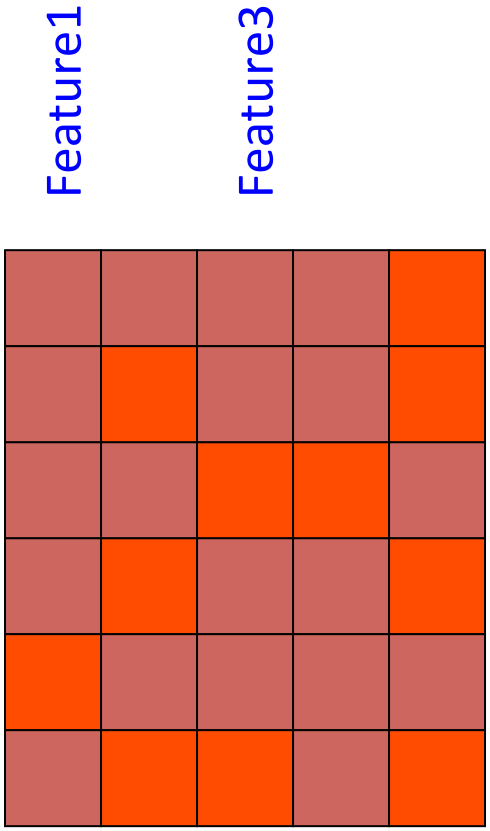
Prediction under sparse annotation



	Four blue features?	Features 1 and 3 blue?
Gene1	Yes	Yes
Gene2	No	Yes
Gene3	No	No
Gene4	No	Yes
Gene5	Yes	No
Gene6	No	No

Which genes are involved in tRNA processing?

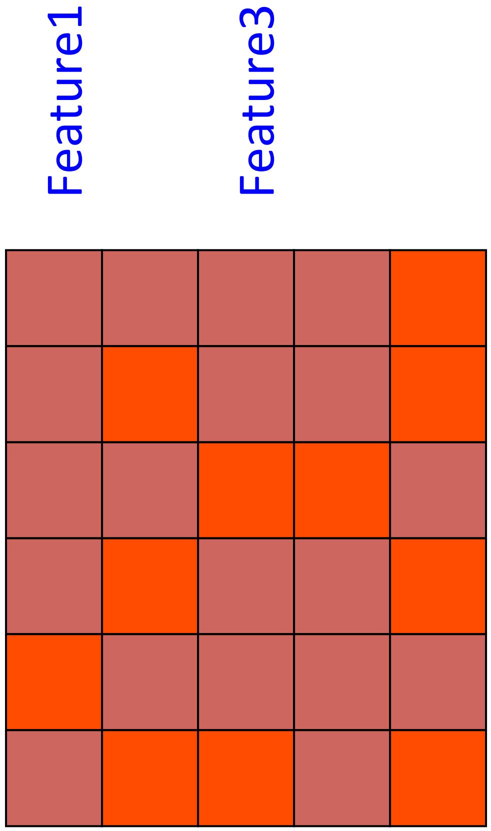
Prediction under sparse annotation



	Four blue features?	Features 1 and 3 blue?	Confidence
Gene1	Yes	Yes	1.0
Gene2	No	Yes	0.5
Gene3	No	No	0
Gene4	No	Yes	0.5
Gene5	Yes	No	0.5
Gene6	No	No	0

Legend		
	1.0	Definitely involved
	0.5	May be involved
	0	Definitely not involved

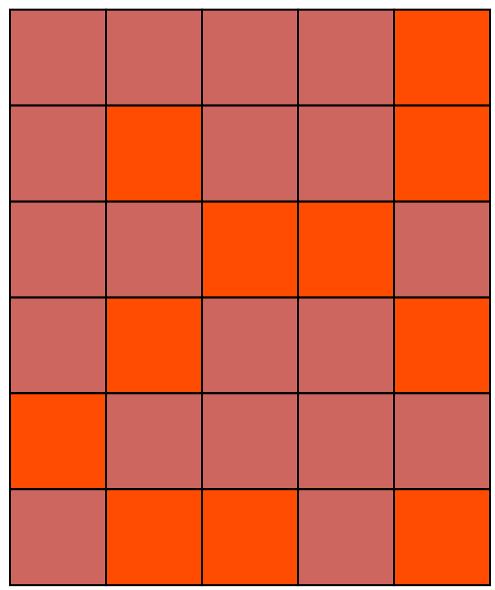
Prediction under sparse annotation



	Four blue features?	Features 1 and 3 blue?	Confidence
Gene1	Yes	Yes	1.0
Gene2	No	Yes	0.5
Gene3	No	No	0
Gene4	No	Yes	0.5
Gene5	Yes	No	0.5
Gene6	No	No	0

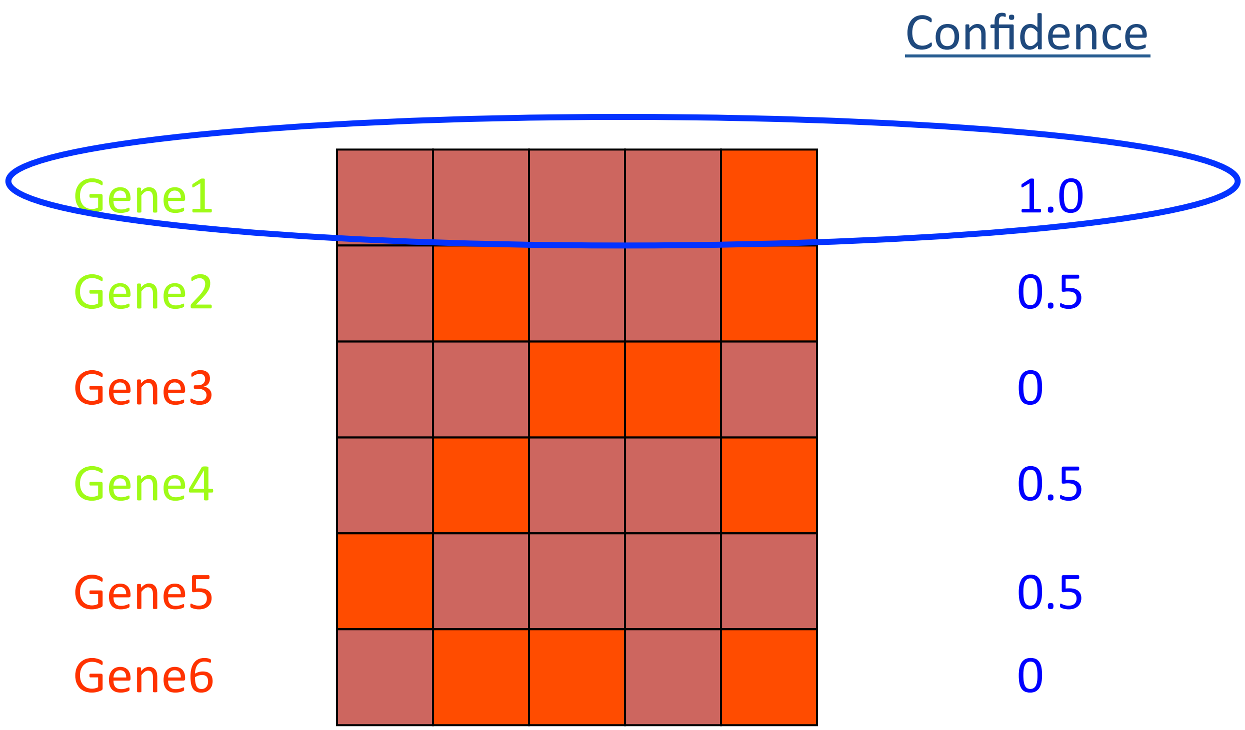
Prediction: Gene1, and probably Genes 2, 4, and 5 are involved in tRNA processing.

Experimental validation



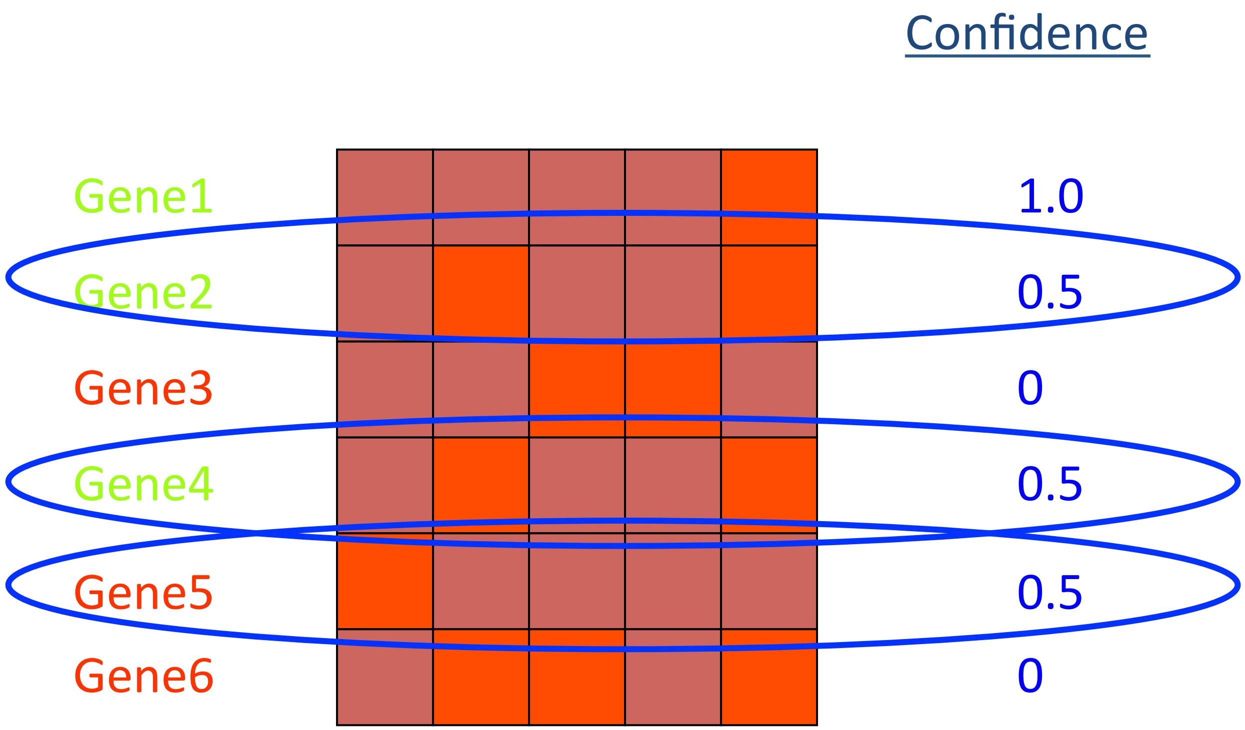
Label		Confidence
+	Gene1	1.0
+	Gene2	0.5
-	Gene3	0
+	Gene4	0.5
-	Gene5	0.5
-	Gene6	0

Experimental validation



One correct "confidence 1" prediction

Experimental validation



Two out of three "confidence 0.5" predictions correct.

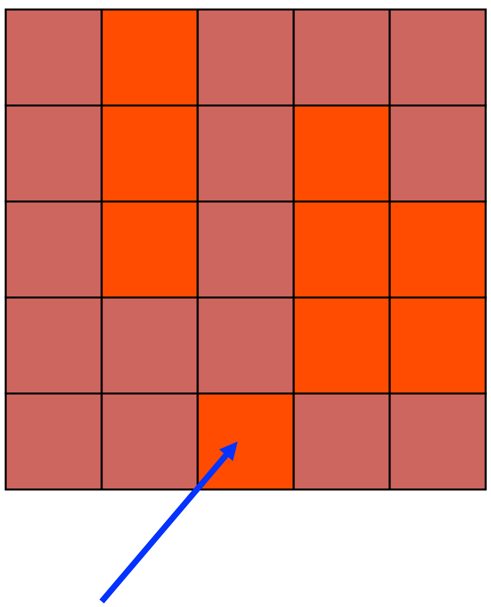
Validation results

Confidence Cutoff	# True Positives	# False Positives
1	1	0
0.5	3	1
0	3	3

	Confidence
Gene1	1.0
Gene2	0.5
Gene3	0
Gene4	0.5
Gene5	0.5
Gene6	0

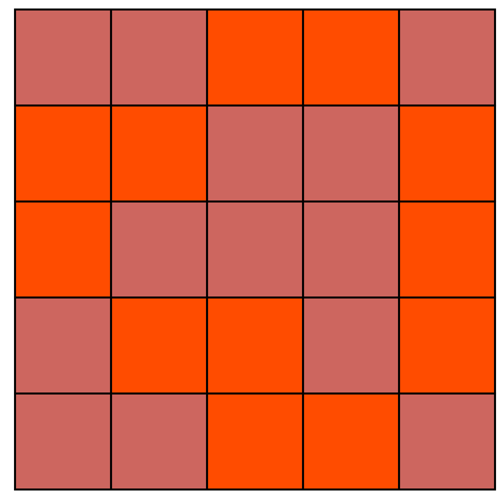
Noisy features

Positives



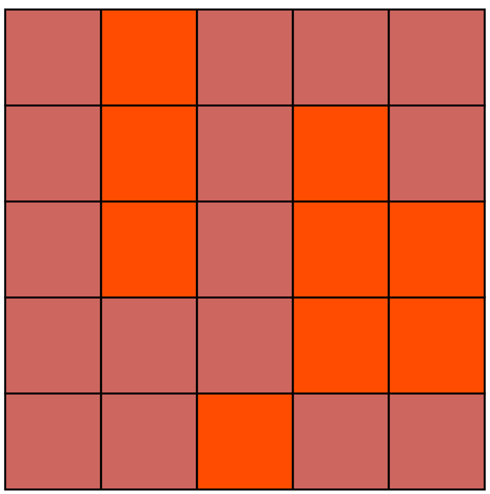
Incorrect measurement,
should be blue.

Negatives

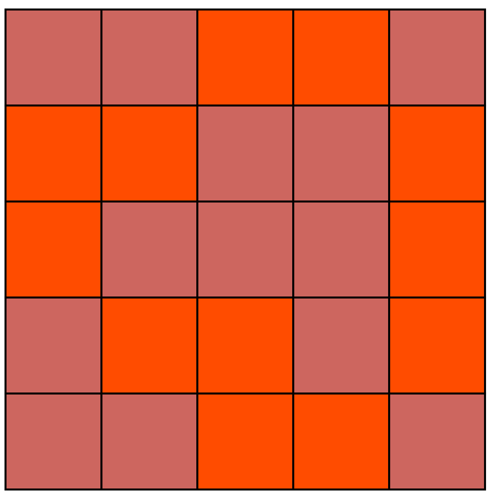


Noisy features

Positives



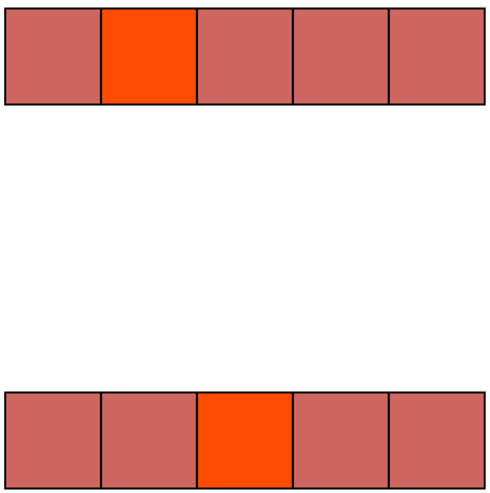
Negatives



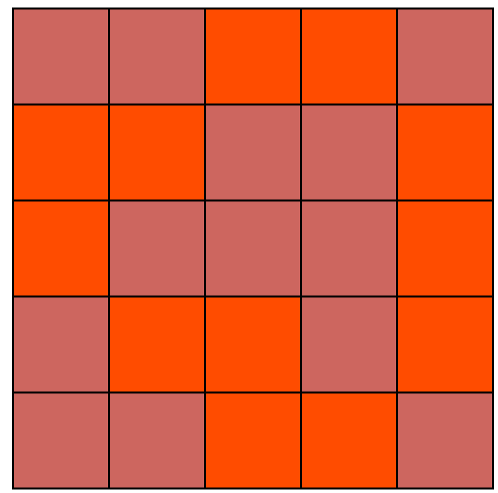
What distinguishes the
positives and the negatives?

Noisy features + sparse data = overfitting

Positives



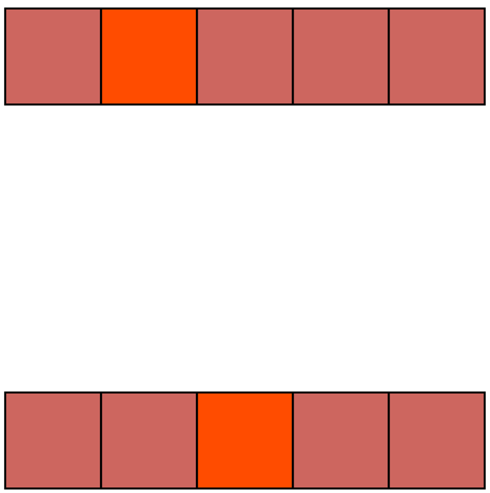
Negatives



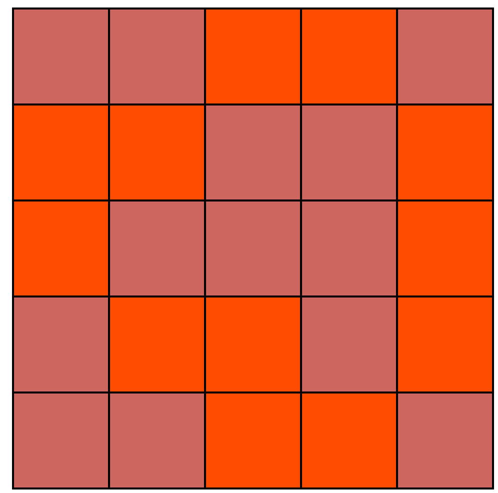
What distinguishes the positives and the negatives?

Training

Positives

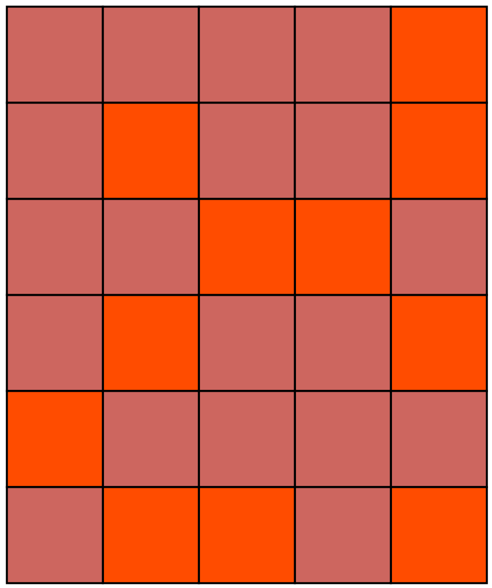


Negatives



- 4 blue features

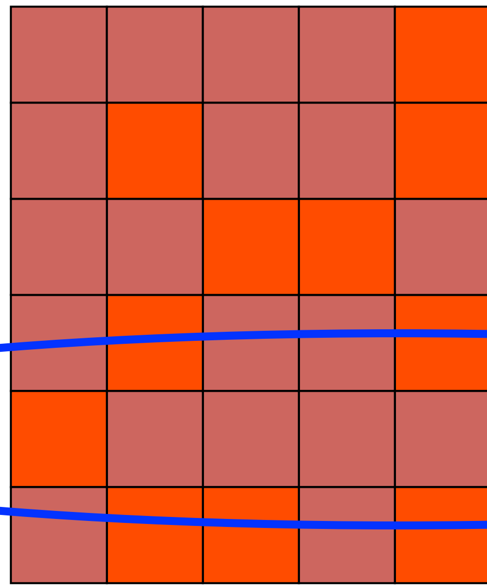
Prediction



	Four blue features?	Confidence
Gene1	Yes	1.0
Gene2	No	0
Gene3	No	0
Gene4	No	0
Gene5	Yes	1.0
Gene6	No	0

Prediction: Gene1 and 5 are involved in tRNA processing.

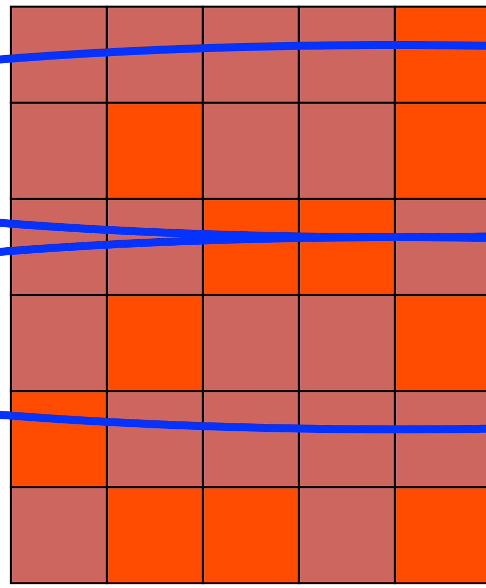
Experimental validation



	Four blue features?	Confidence
Gene1	Yes	1.0
Gene2	No	0
Gene3	No	0
Gene4	No	0
Gene5	Yes	1.0
Gene6	No	0

One incorrect high confidence prediction, i.e., one false positive

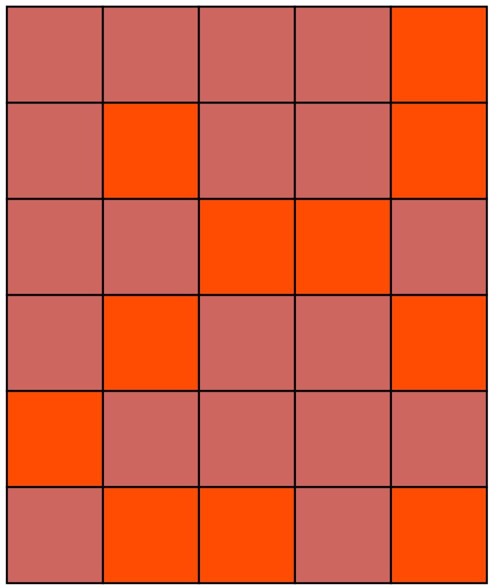
Experimental validation



	Four blue features?	Confidence
Gene1	Yes	1.0
Gene2	No	0
Gene3	No	0
Gene4	No	0
Gene5	Yes	1.0
Gene6	No	0

Two genes missed completely, i.e., two false negatives

Experimental validation



	Four blue features?	Confidence
Gene1	Yes	1.0
Gene2	No	0
Gene3	No	0
Gene4	No	0
Gene5	Yes	1.0
Gene6	No	0

One incorrect high confidence prediction, two genes missed completely

Validation results

Confidence Cutoff	# True Positives	# False Positives
1	1	1
0	3	3

	Confidence
Gene1	1.0
Gene2	0
Gene3	0
Gene4	0
Gene5	1.0
Gene6	0

What have we learned?

- Sparse data: many different patterns distinguish positives and negatives.
- Noisy features: Actual distinguishing pattern may not be observable
- Sparse data + noisy features: may detect, and be highly confident in, spurious, incorrect patterns.

Overfitting

Validation

- Different algorithms assign confidence to their predictions differently
- Need to
  1. Determine meaning of each algorithm's confidence score.
  2. Determine what level of confidence is warranted by the data

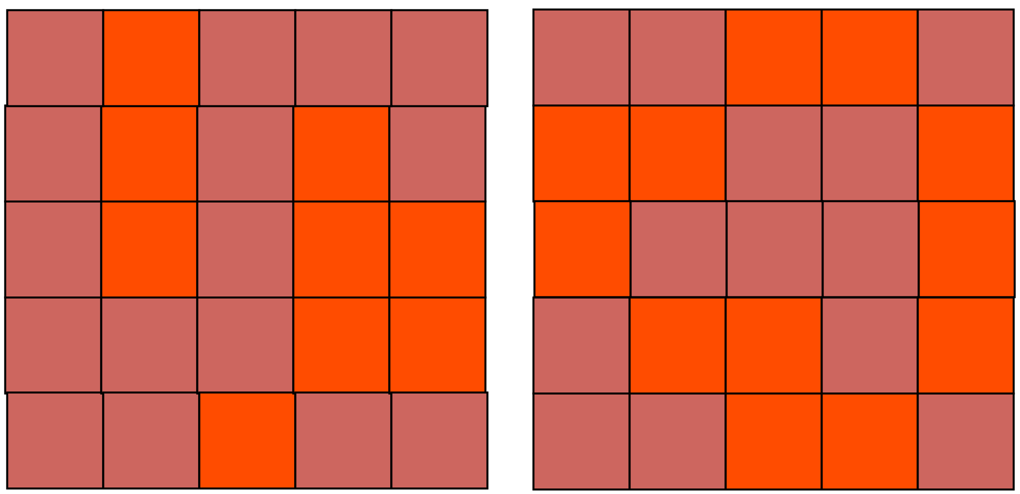
Cross-validation

Basic idea:

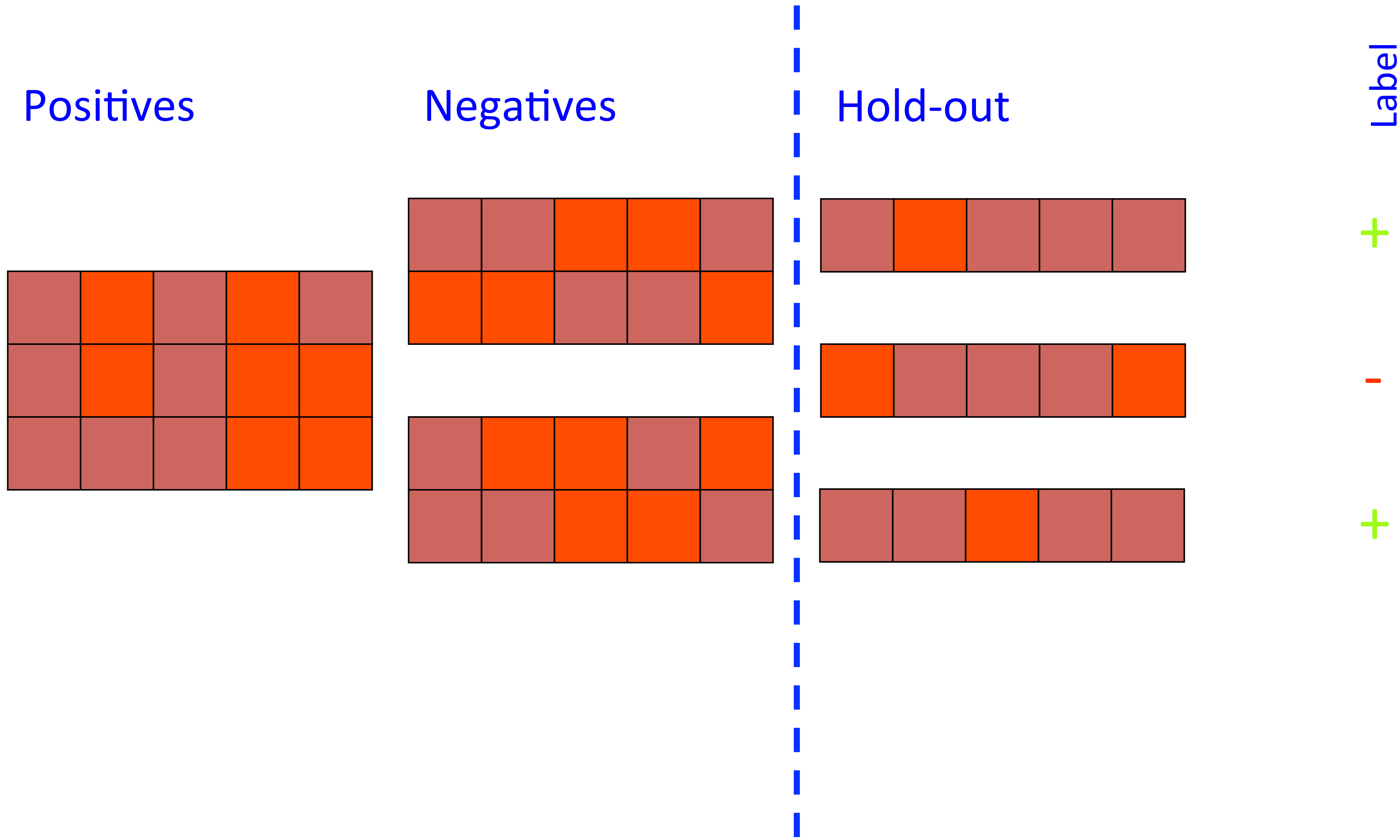
Hold out part of the data and use it to validate confidence levels

Cross-validation

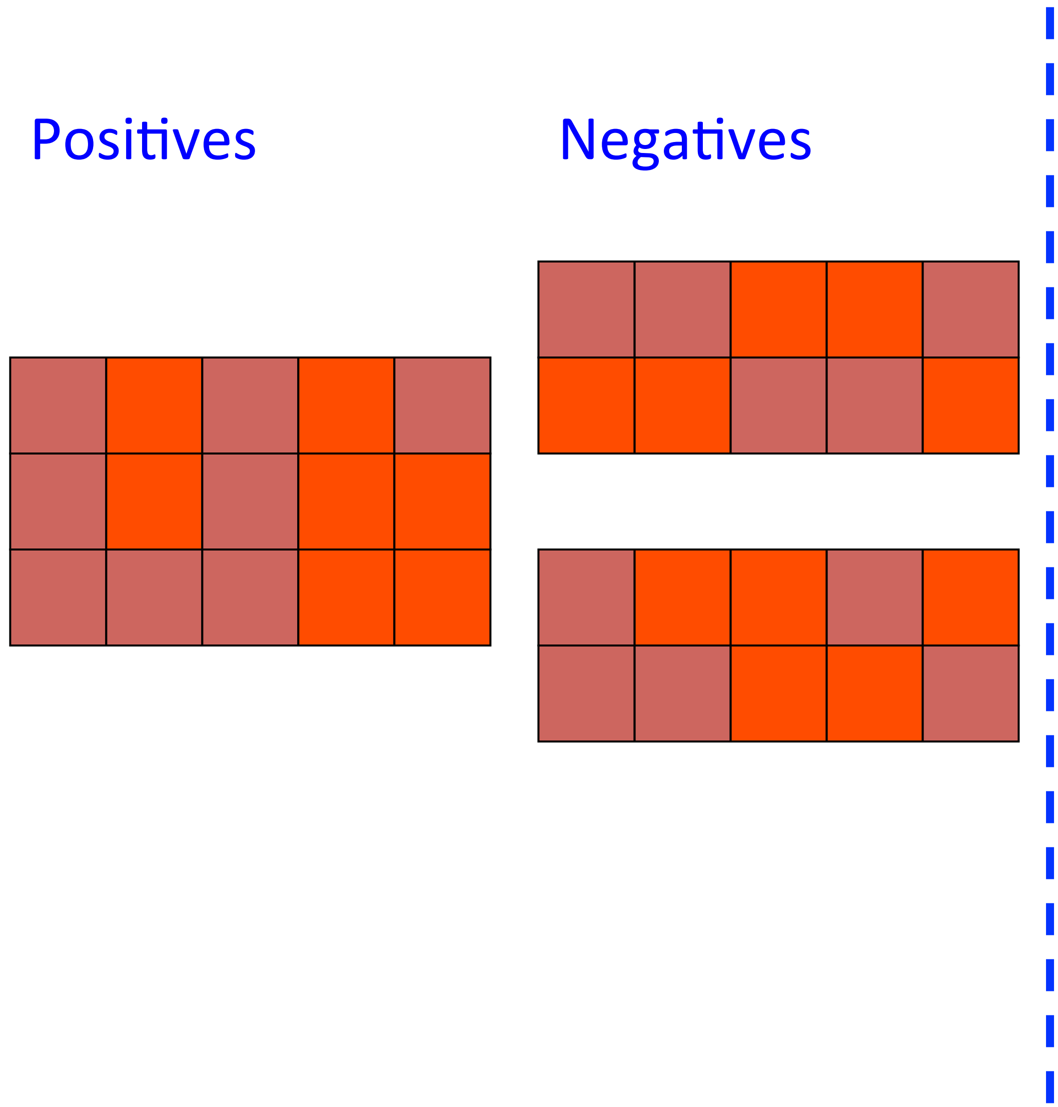
Positives Negatives



Cross-validation

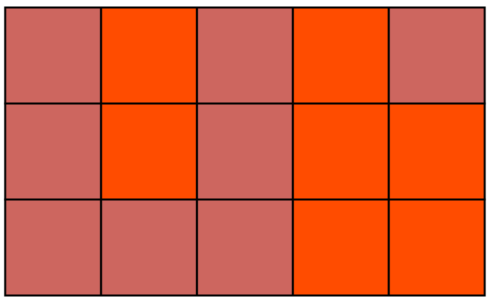


Cross-validation: training

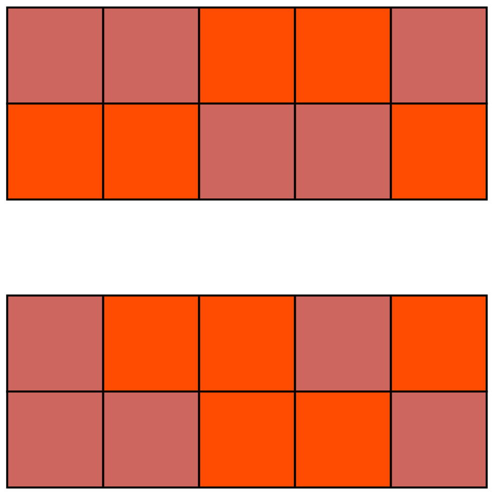


Cross-validation: training

Positives

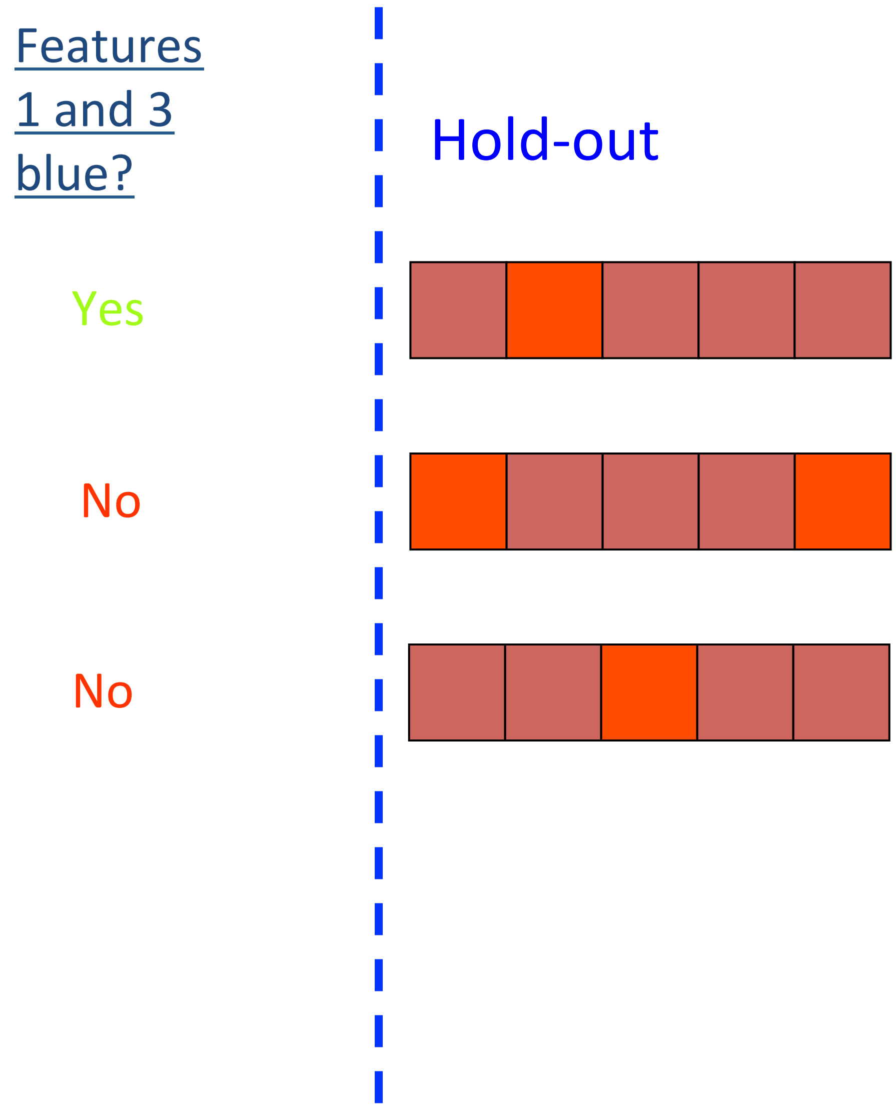


Negatives



- Features 1 and 3 are blue

Cross-validation: testing

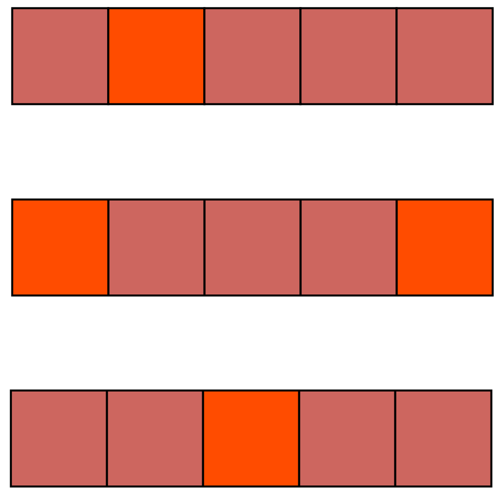


Cross-validation: testing

Features 1 and 3 blue?

Hold-out

Confidence



Yes — 1.0

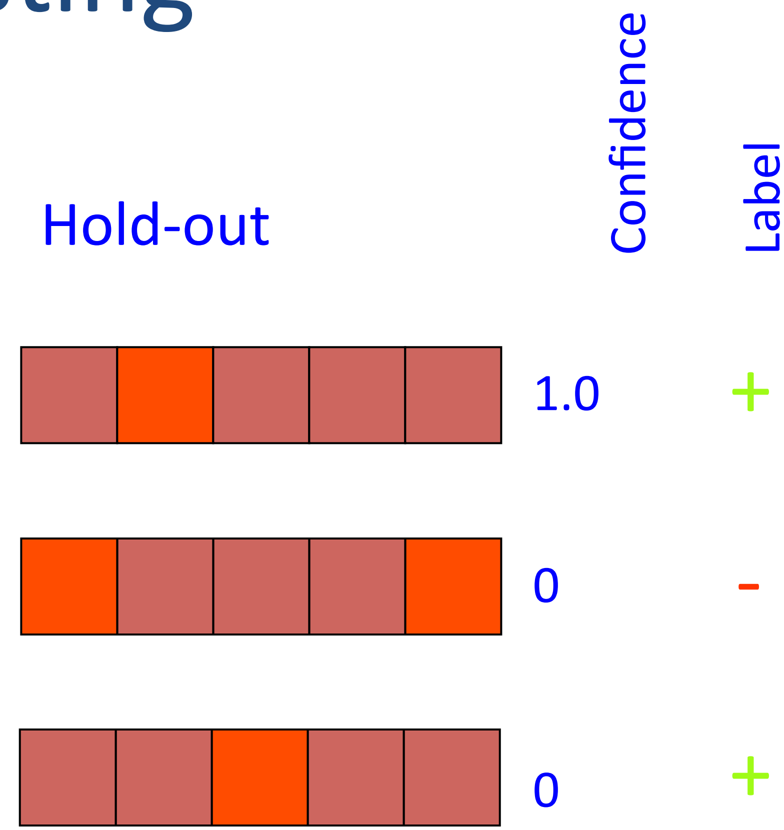
No — 0

No — 0

Cross-validation: testing

Cross-validation: testing

Confidence cutoff	# True Positives	# False Positives
1	1	0
0	2	1

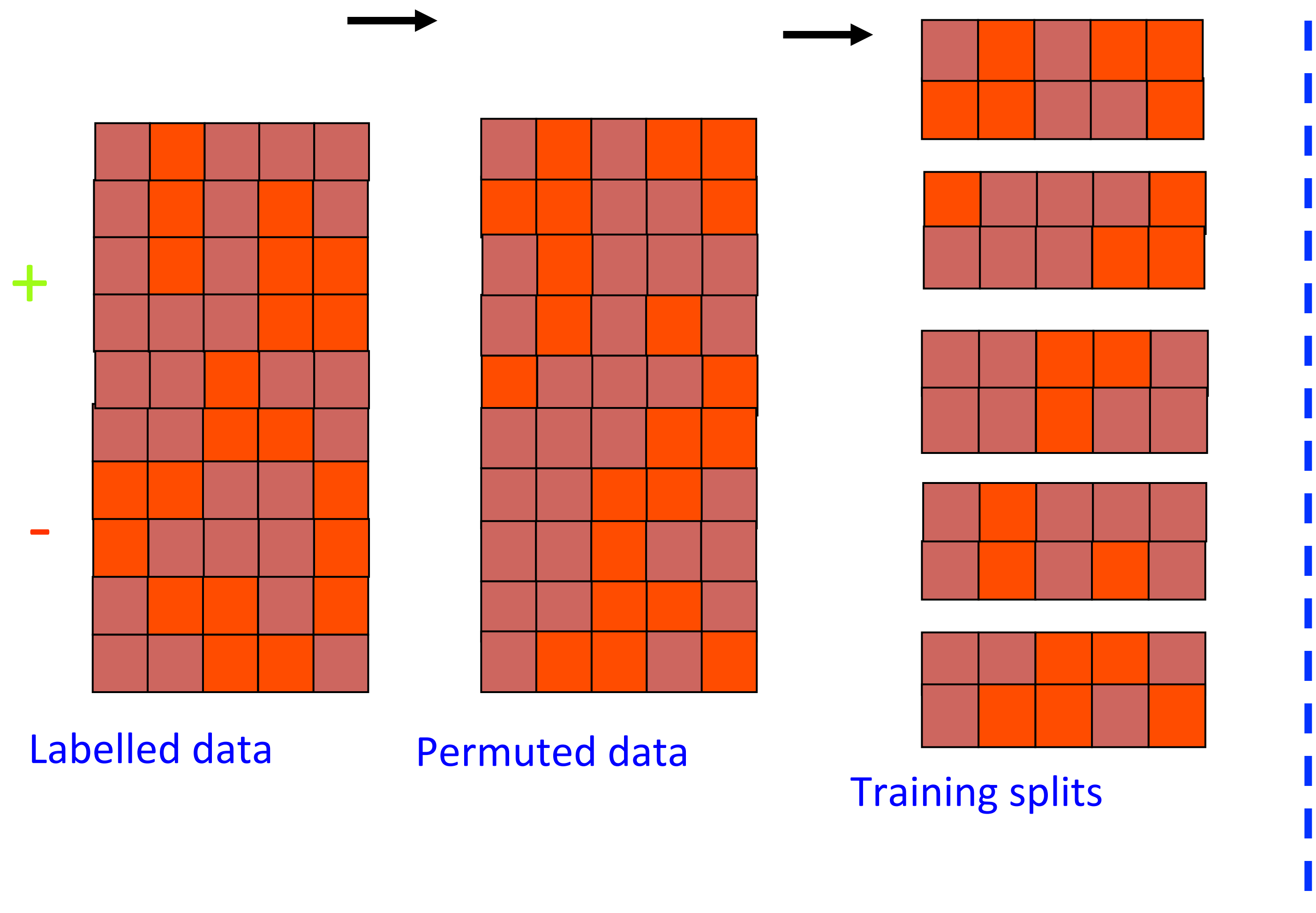


N-fold cross validation

Step 1: Randomly reorder rows

Step 2: Split into N sets (e.g. N = 5)

Step 3: Train N times, using each split, in turn, as the hold-out set

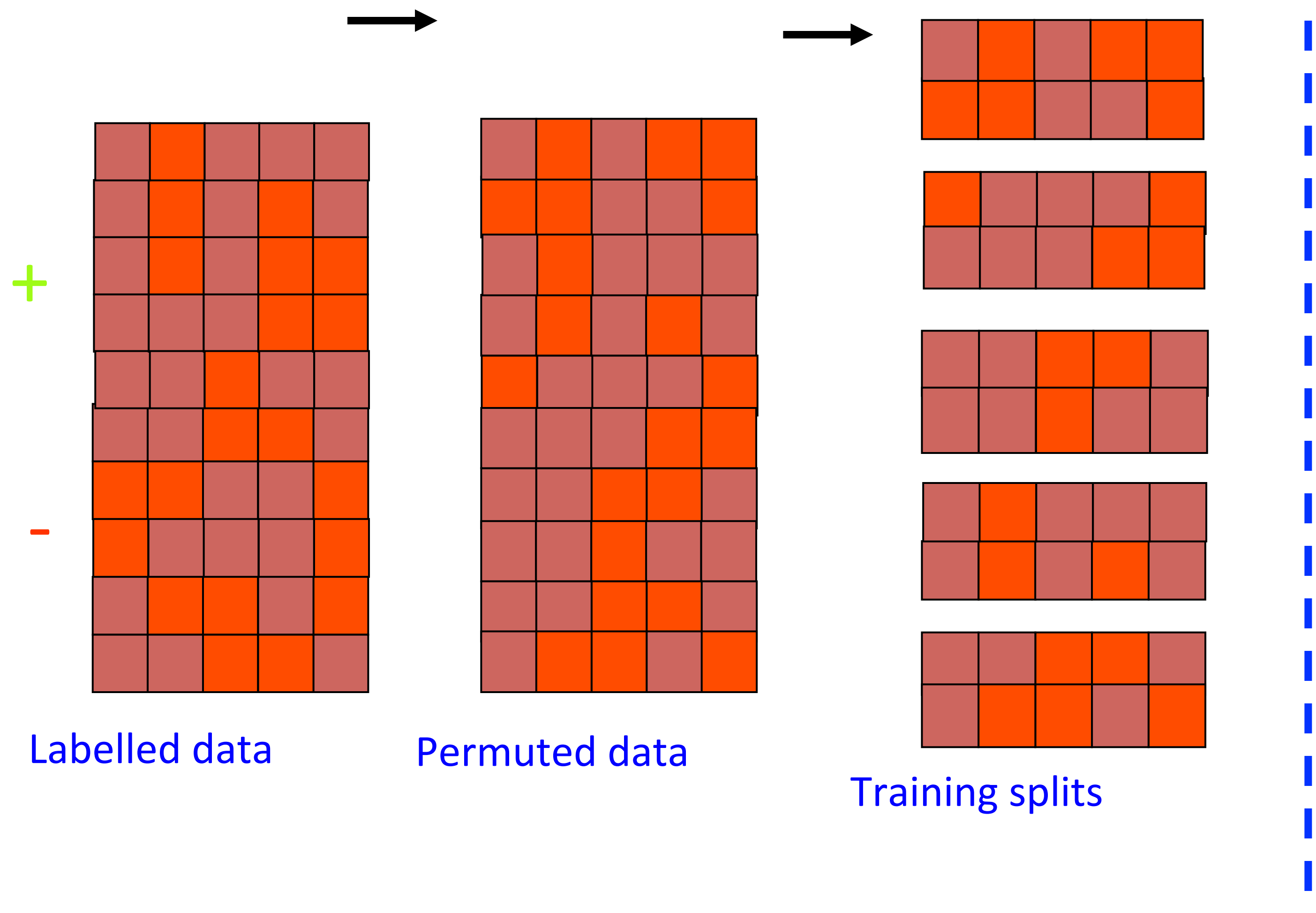


N-fold cross validation

Step 1: Randomly reorder rows

Step 2: Split into N sets
(e.g. N = 5)

Step 3: Train N times, using each split, in turn, as the hold-out set

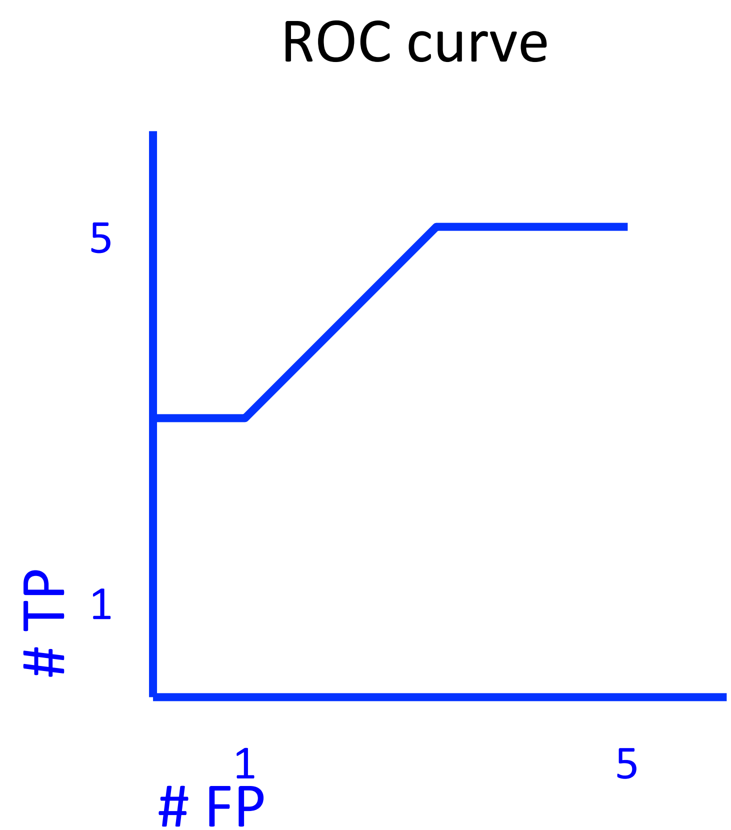


Cross-validation results

Confidence cutoff	# True Positives	# False Positives
1	3	0
0.75	3	1
0.5	4	2
0.25	5	3
0	5	5

Displaying results: ROC curves

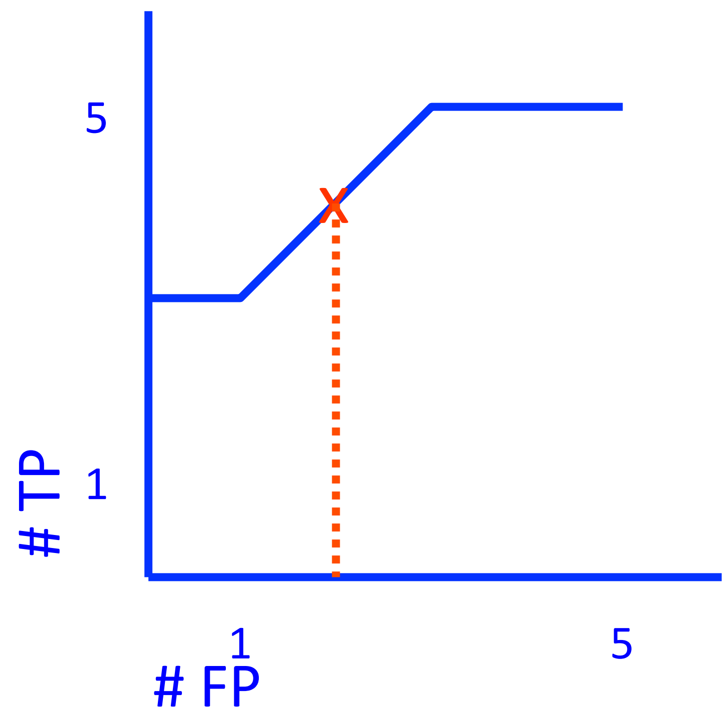
Confidence cutoff	# True Positives	# False Positives
1	3	0
0.75	3	1
0.5	4	2
0.25	5	3
0	5	5



Making new predictions

Confidence cutoff	# True Positives	# False Positives
1	3	0
0.75	3	1
0.5	4	2
0.25	5	3
0	5	5

ROC curve



Figures of merit

Precision: #TP / (#TP + #FP)
(also known as **positive predictive value**)

Recall: #TP / (#TP + #FN)
(also known as **sensitivity**)

Specificity: #TN / (#FP + #TN)

Negative predictive value: #TN / (#FN + #TN)

Accuracy: (#TP + #TN) / (#TP + #FP + #TN + #FN)

Confusion matrix

		Predicted	
		T	F
Actual	T	TP	FN
	F	FP	TN

Area under the ROC curve

Area Under the ROC Curve (AUC) =
Average proportion of negatives with confidence levels less than a random positive

Quick facts:

- 0 < AUC < 1
- AUC of random classifier = 0.5