Feature Engineering and Supervised Learning

Responsible:
Dr. Tegawendé F. BISSYANDE
tegawende.bissyande@uni.lu

Course Author:
Christopher Henard

Teacher Assistant:
Médéric Hurier
mederic.hurier@uni.lu

Course Features

- Sep. 20th - **Introduction to Big Data**

Part 1. Databases and Query Models for Big Data

- Sep. 27th - **Relational Databases: Reminders**
- Oct. 4th - **Relational Databases: Internals**
- Oct. 11th - **NoSQL & NewSQL Databases**
- Oct. 18th - **MapReduce Model**
- Oct. 25th - **Hadoop and Spark**
- Nov. 8th - **Datalog Model**

Part 2. Data Analysis and Machine Learning

- Nov. 15th - **Statistics and Data Analysis**
- Nov. 22th - **Communication and Visualization**
- Nov. 29th - **Feature Engineering and Supervised Learning**
- Dec. 5st - **Feature Preprocessing and Unsupervised Learning**
- Dec. 12th - **Homework Time**

Section Features

- Introduction
- Feature Engineering
- Supervised Learning
- Example: k-Nearest Neighbors
- Example: Decision Tree
- Bias, Variance, Measures

Introduction

The data driven approach

- How can we extract knowledge from data to help humans take decisions?
- How can we automate decisions from data?
- How can we adapt systems dynamically to enable better user experiences?

Write code explicitly to do the above tasks

Write code to make the computer learn how to do the tasks

Why is Machine Learning important?

- Humans are unable to explain their expertise
  - e.g. speech recognition, vision, language
- Solution changes in time (routing on a computer network)
- We cannot write the program ourselves
- Solution needs to be adapted to particular cases (user biometrics)
- The problem size is to vast for our limited reasoning capabilities
  - calculating web page ranks

Why is Machine Learning important?

- Some tasks cannot be defined well, except by examples
  - e.g. recognizing people
- Relationships and correlations can be hidden within large amounts of data. Machine Learning/Data Mining may be able to find these relationships.
- Human designers often produce machines that do not work as well in the environments in which they are used.
- Human expertise is absent (e.g. navigating on Mars)

Definition: Machine Learning

A branch of artificial intelligence, concerns with the construction and study of systems that
can learn from data.

Definition: Machine Learning

- Can be used to predict outcome in new situation

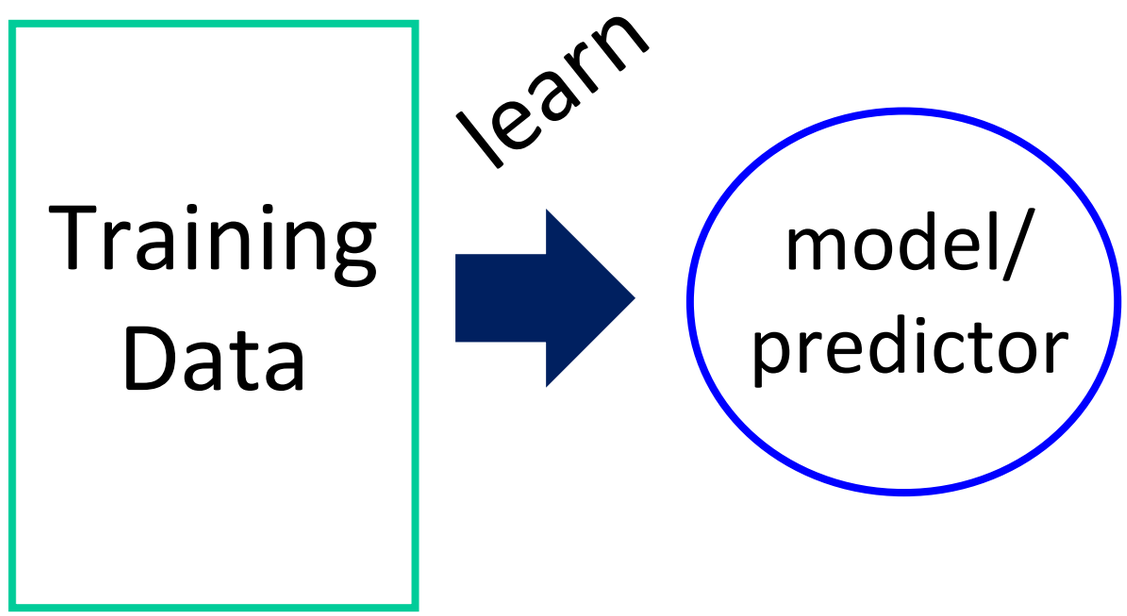
- Can be used to understand and explain how prediction is derived

- Methods originate from artificial intelligence, statistics, and research on databases

"Making computers behave like they do in the movies"

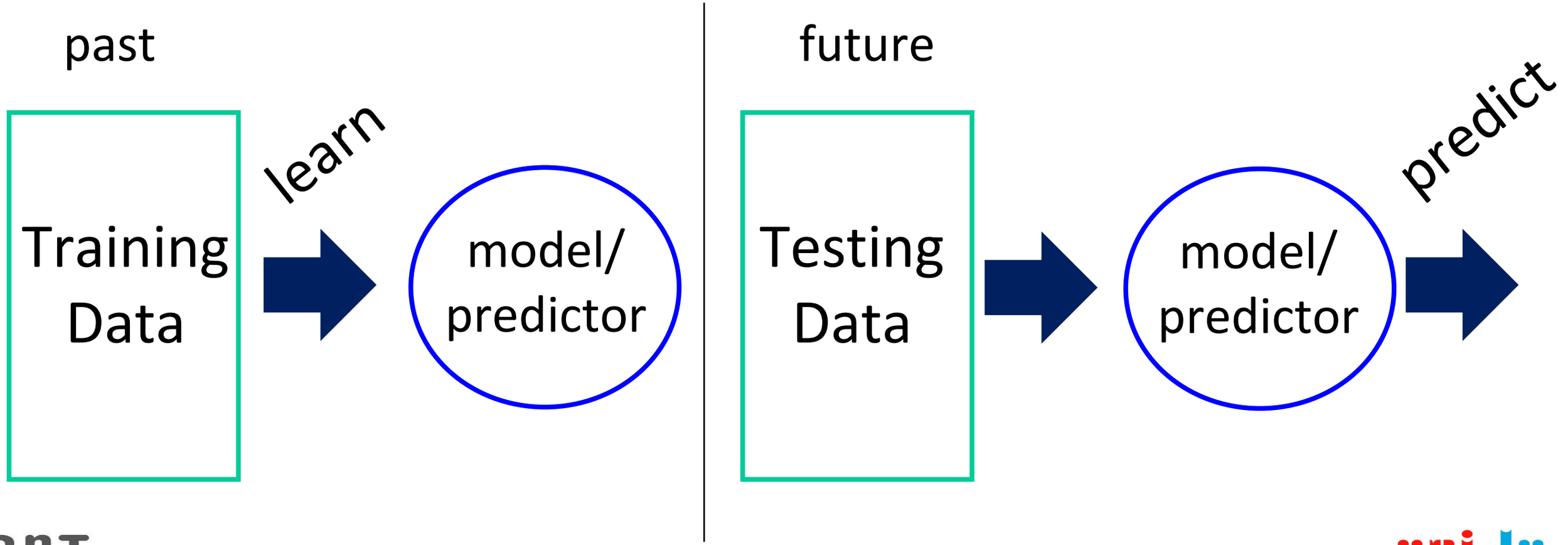
Machine Learning is…

… about predicting the future based on the past.

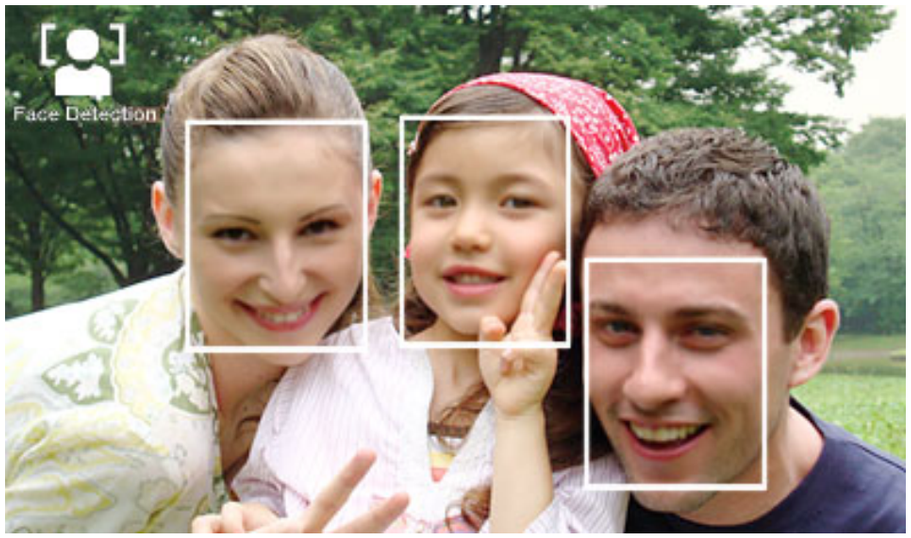


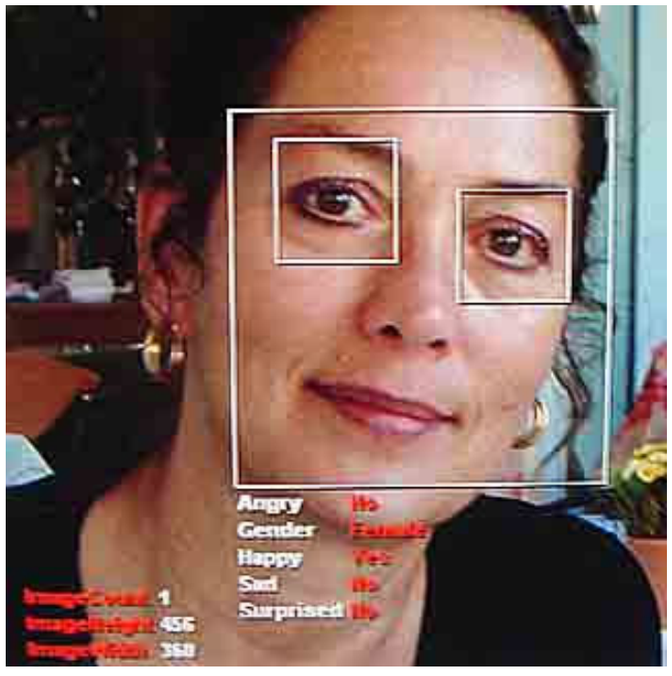
Machine Learning is…

…about predicting the future based on the past.

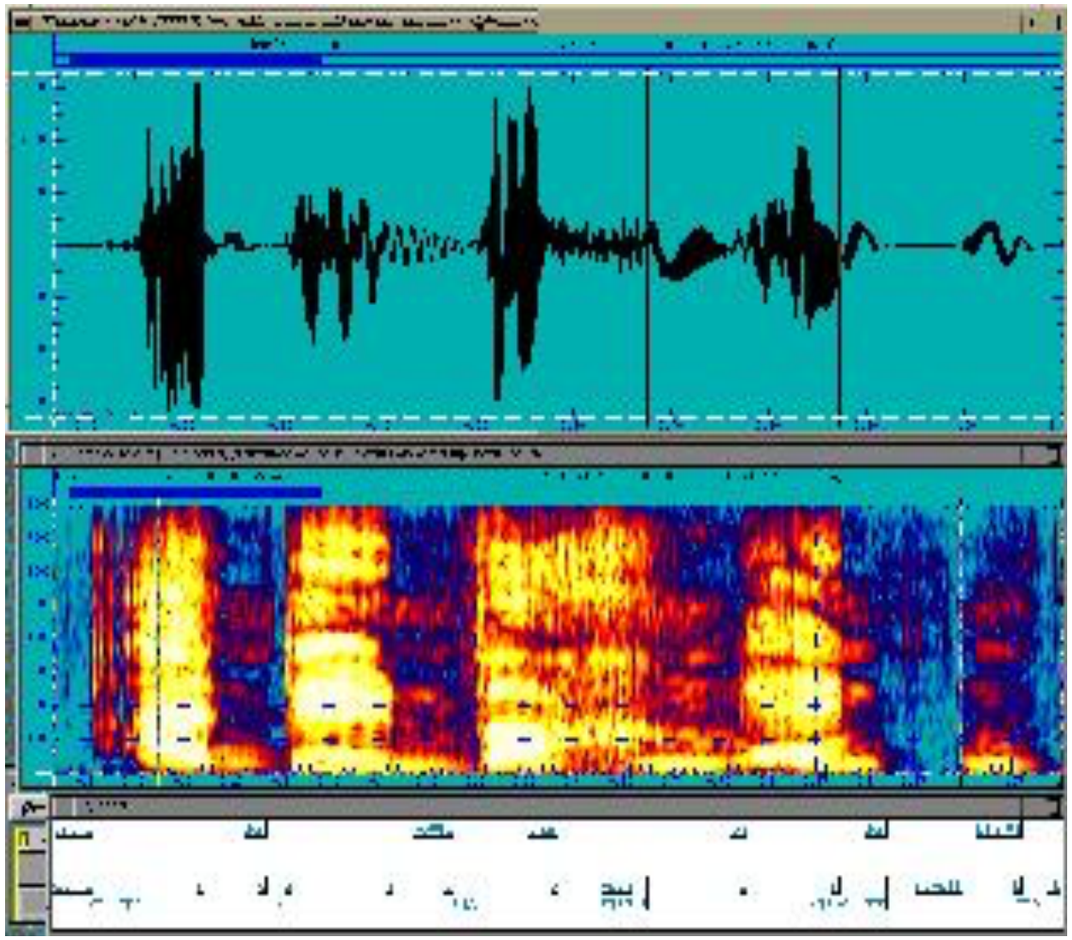


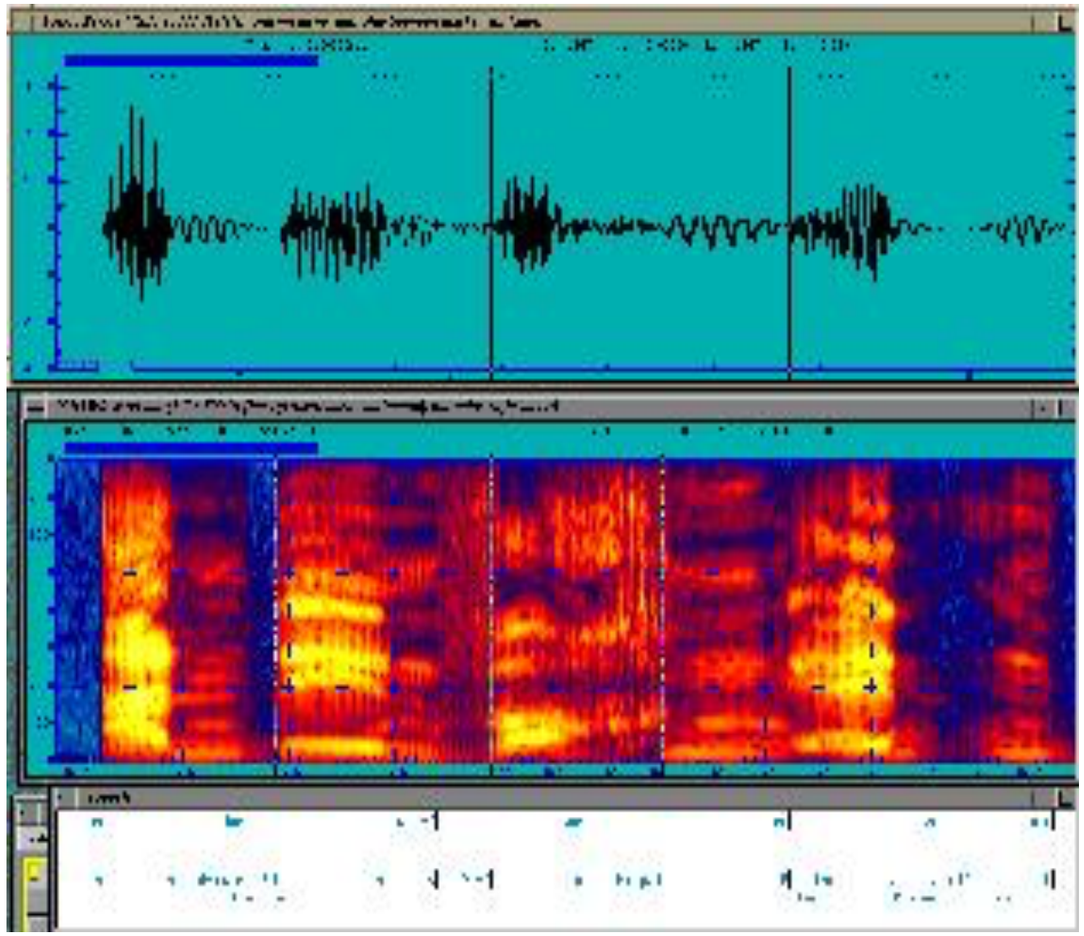
Identify faces and expressions





Identify vocal patterns





Detecting frauds, targeted advertising, etc.

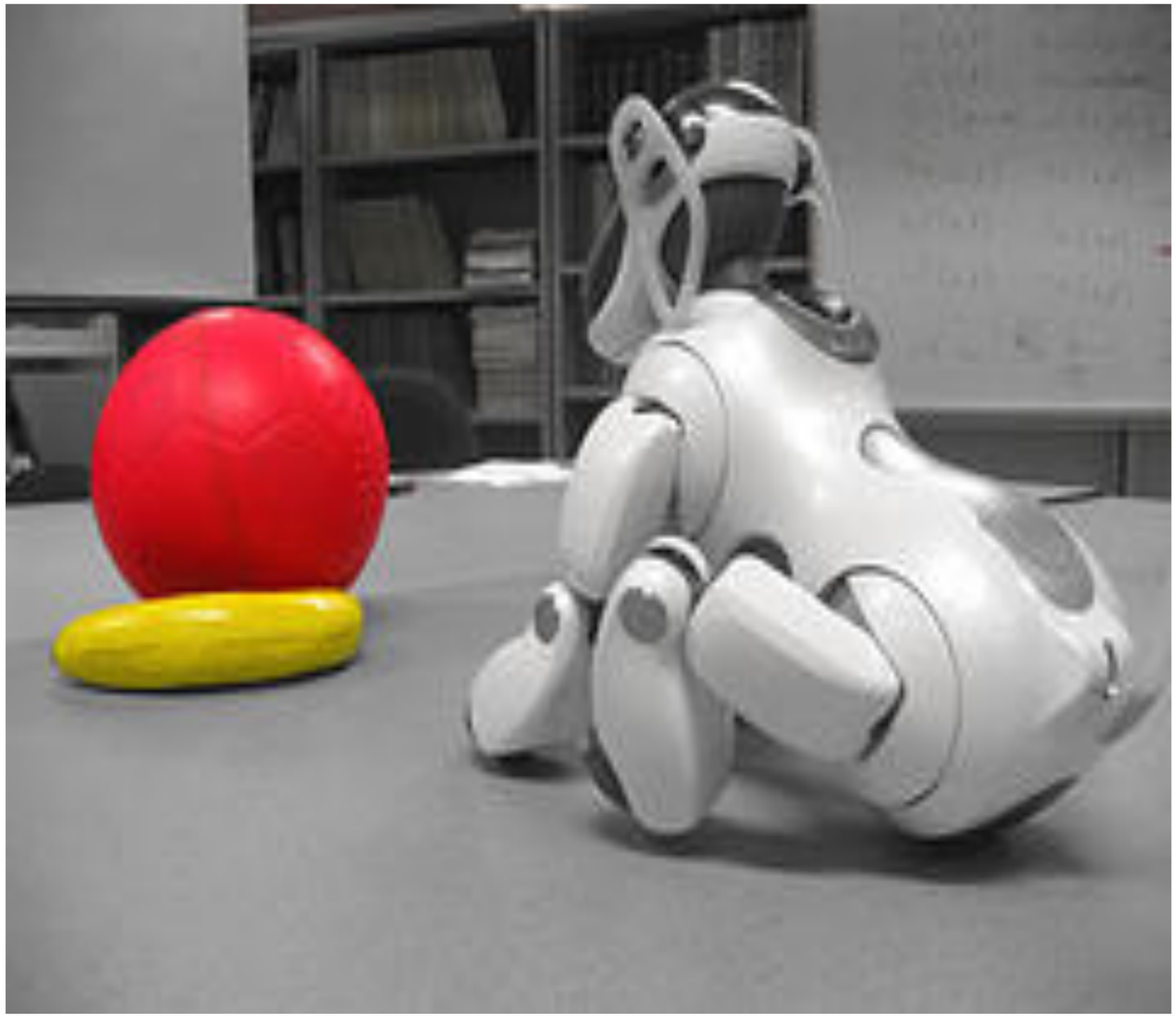
facebook

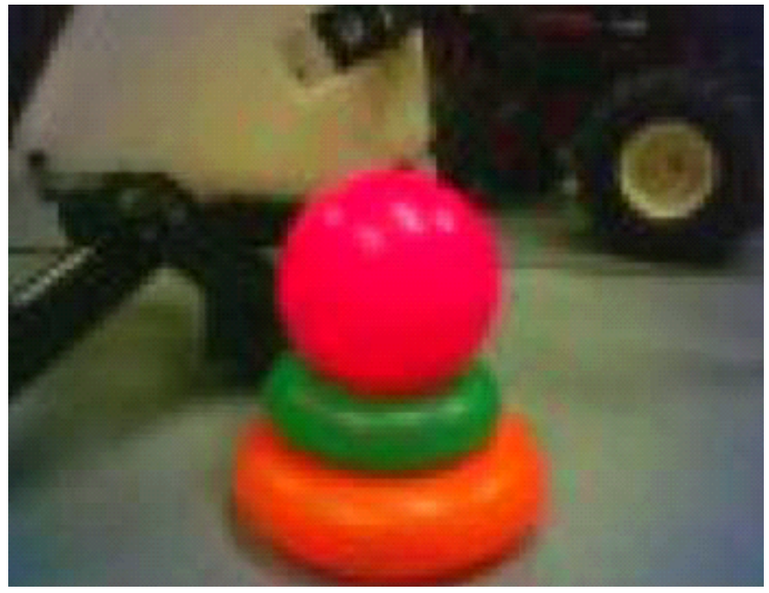
amazon.com

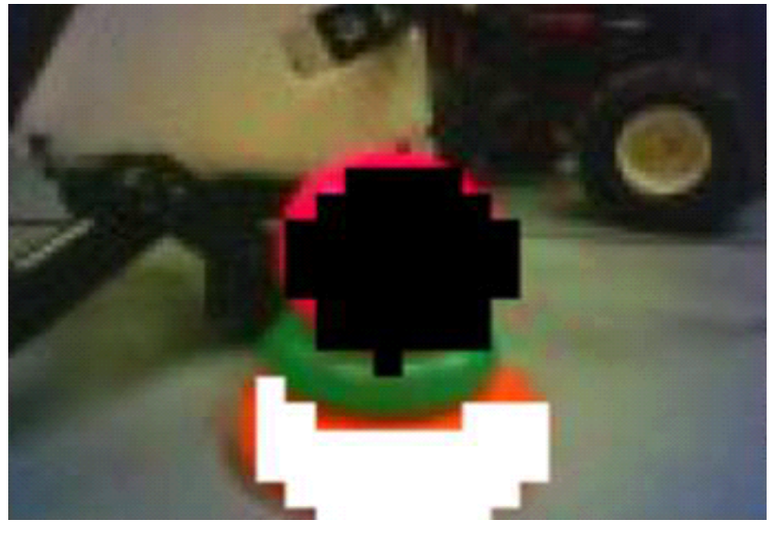
ebay

Lloyds TSB

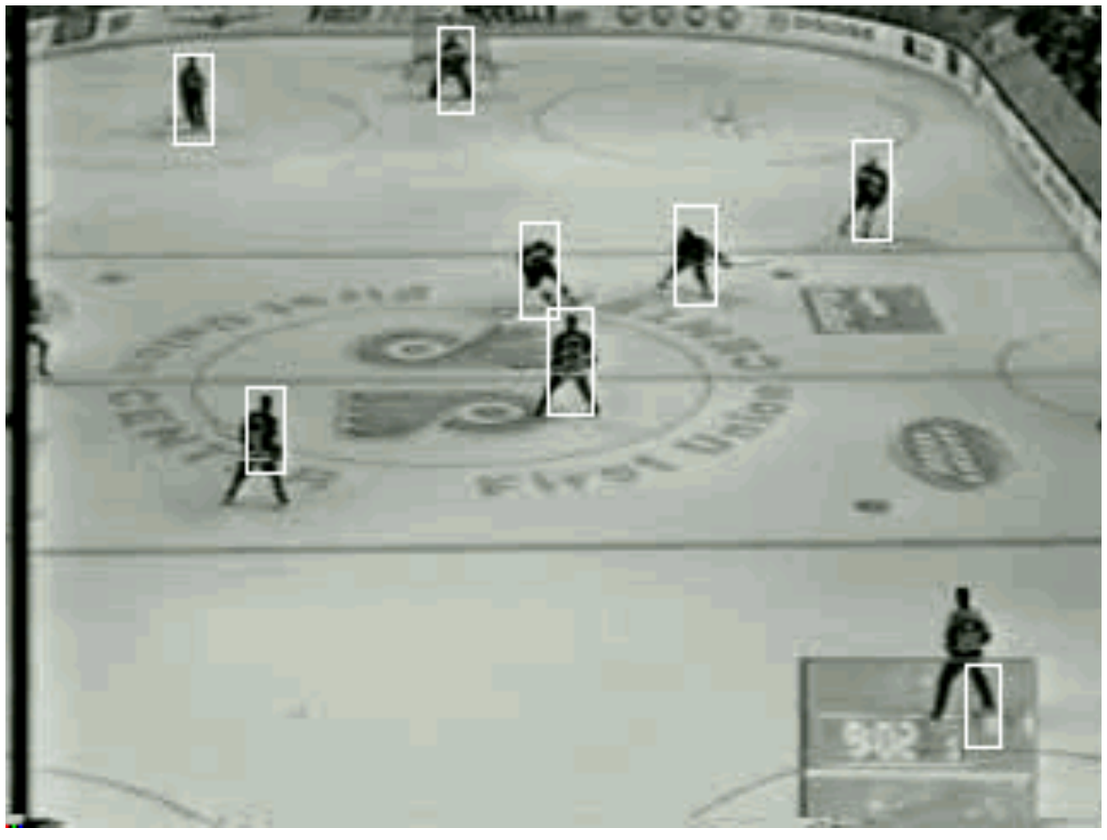
Robotics

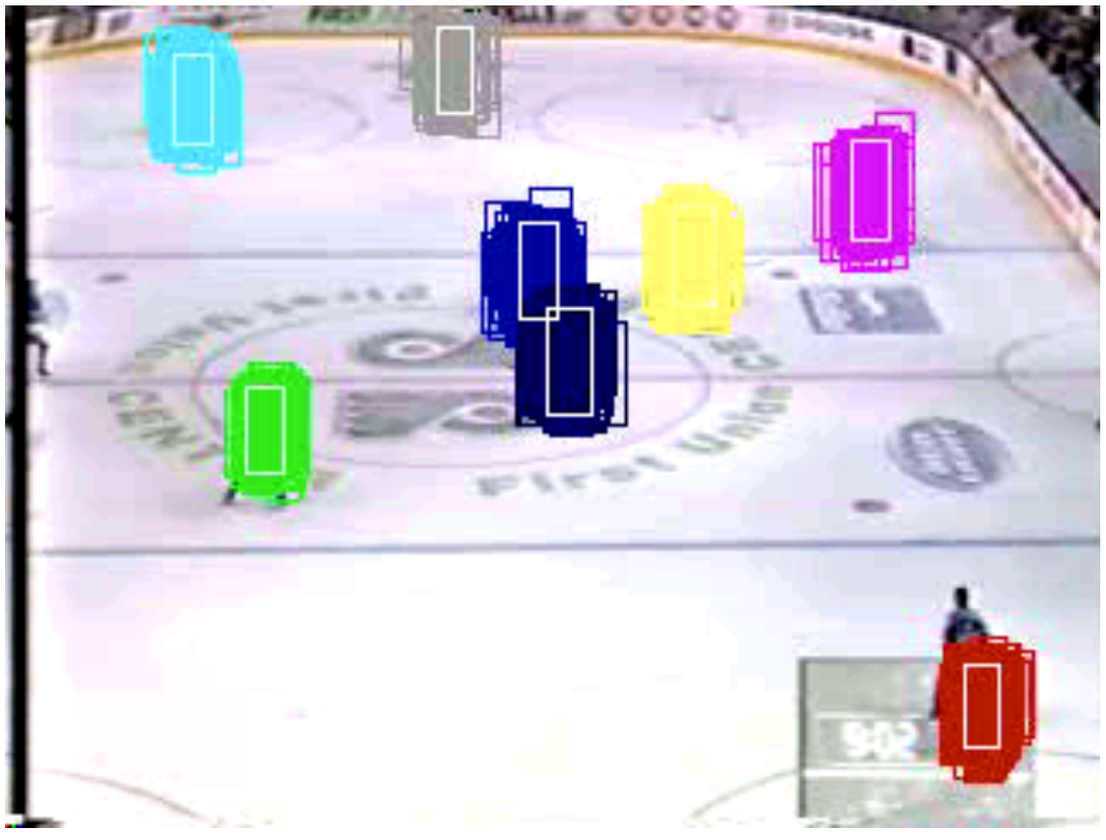






Tracking and recognition





And many other tasks

- Recognizing spam emails
- Recommending books
- Reading handwriting
- Recognizing speech, faces, handwriting, etc.
- …

Subdomains of Machine Learning

Supervised Learning	Unsupervised Learning	Reinforcement Learning
Classification	Clustering	Decision Process
Regression	Segmentation	Reward System
Ranking	Dimension Reduction	Recommendation Systems
	Association Mining	

Feature Engineering

Data

Data

Features

Examples

features

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

How our algorithms actually "view" the data

Features are the questions we can ask about the examples

Features

Examples

features

red, round, leaf, 3oz, …

green, round, no leaf, 4oz, …

yellow, curved, no leaf, 4oz, …

green, curved, no leaf, 5oz, …

How our algorithms actually “view” the data

Features are the questions we can ask about the examples

Features

Examples

features

red, round, leaf, 3oz, …

Where do they come from?

yellow, curved, no leaf, 4oz, …

green, curved, no leaf, 5oz, …

How our algorithms actually “view” the data

Features are the questions we can ask about the examples

UCI Machine Learning Repository

http://archive.ics.uci.edu/ml/datasets.html

Work with provided features

In many physical domains
(e.g. biology, medicine, chemistry, engineering, etc.)

- the data has been collected and the *relevant* features identified
- we cannot collect more features from the examples (at least "core" features)

We can often just use the provided features

Example of provided features

Predicting breast cancer recurrence

1. Class: no-recurrence-events, recurrence-events
2. age: 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99.
3. menopause: lt40, ge40, premeno.
4. tumor-size: 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54
5. inv-nodes: 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35
6. node-caps: yes, no.
7. deg-malig: 1, 2, 3.
8. breast: left, right.
9. breast-quad: left-up, left-low, right-up, right-low, central.
10. irradiated: yes, no.

What to do without some provided features ?

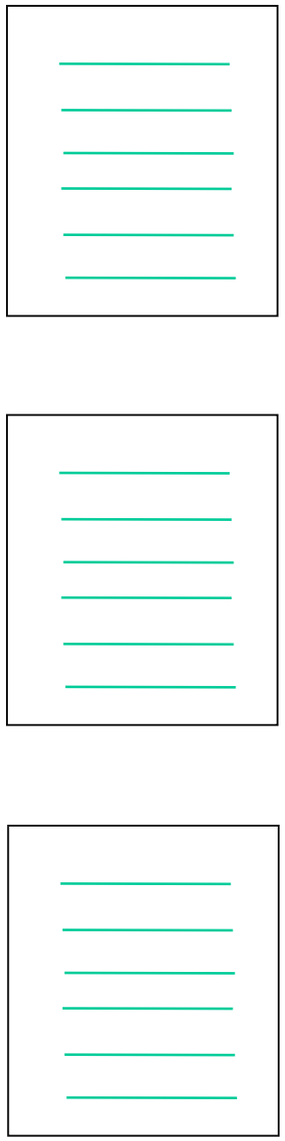
In many other domains, we are provided with the raw data, but must extract/identify features

For example

- image data
- text data
- audio data
- log data
- …

Text features: bag of words

Raw data



Features

Clinton said banana repeatedly last week on tv, “banana, banana, banana”

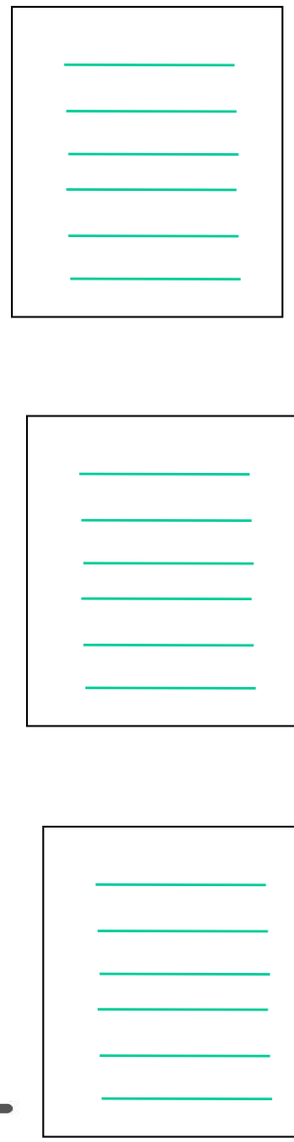
(1, 1, 1, 0, 0, 1, 0, 0, …)

banana, clinton, said, california, across, tv, wrong, capital

Occurrence of words

Text features: bag of words

Raw data



Features

Clinton said banana repeatedly last week on tv, “banana, banana, banana”

(4, 1, 1, 0, 0, 1, 0, 0, …)

banana, clinton, said, california, across, tv, wrong, capital

Frequency of word occurrence

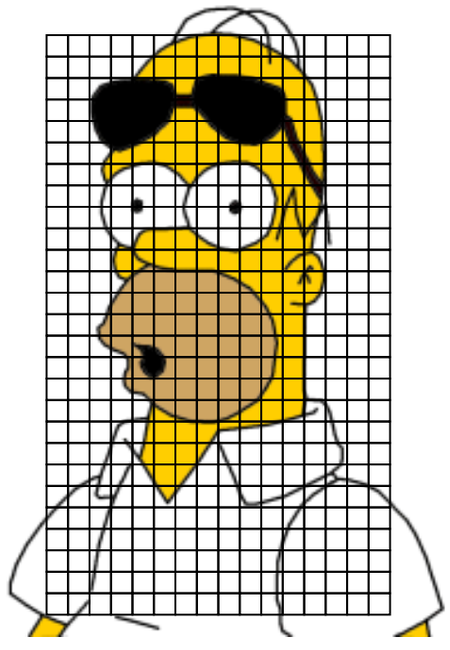
Do we retain all the information in the original document?

Text features: others

- Occurrence, counts, sequence
- Whether ‘banana’ occurred more times than ‘apple’
- If the document has a number in it
- …

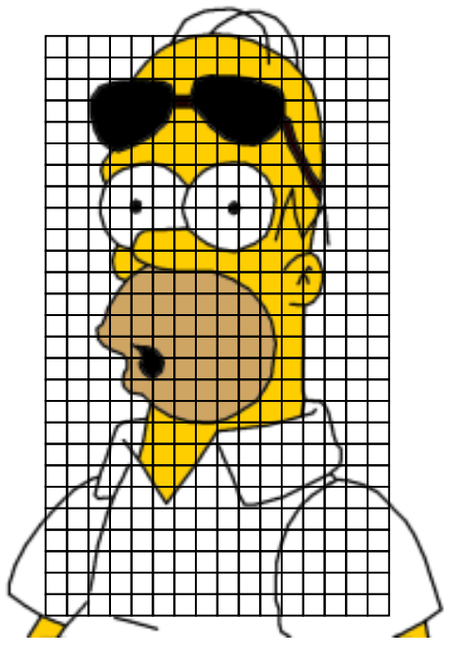
Features are very important !

Image features: pixels



- images are made up of pixels
- for a color image, each pixel corresponds to an RGB value
  i.e. three numbers

Image features: pixels

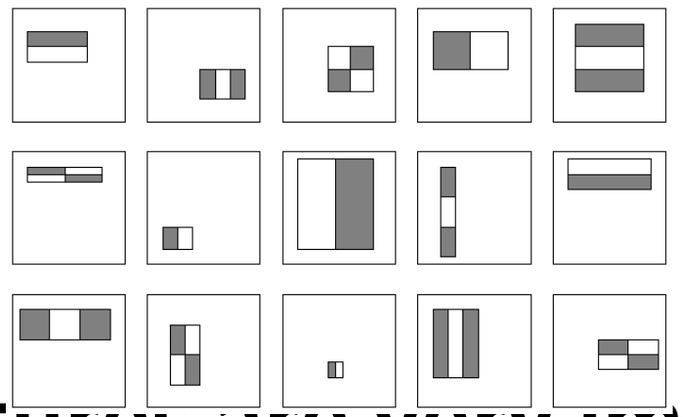


for each pixel:
R[0-255], G[0-255], B[0-255]

Do we retain all the information in the original document?

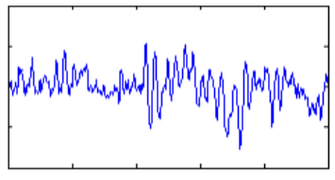
Image features: others

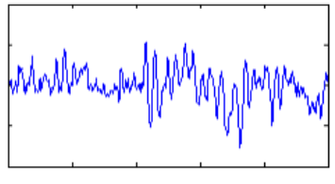
- Use “patches” rather than pixels
    - sort of like “bigrams” for text
- Different color representations (i.e. L*A*B*)
- Texture features, i.e. responses to filters

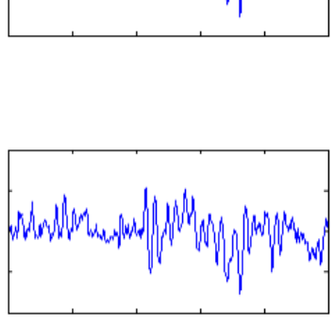


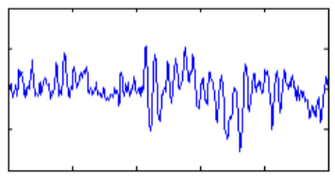
Features are very important !

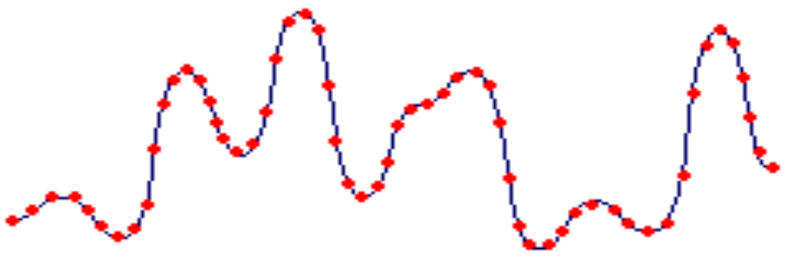
Audi features: time series









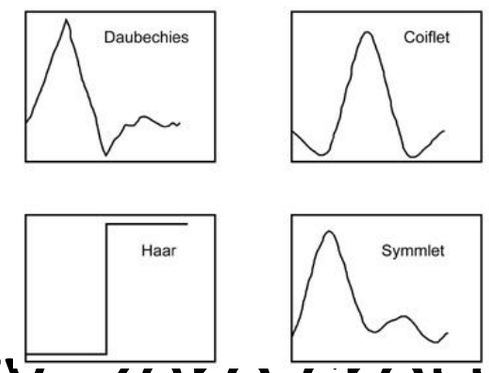


Many different file formats, but some notion of the frequency over time

Audio features?

Audio features: others

- frequencies represented in the data (FFT)
- frequencies over time (STFT)
- responses to wave patterns (wavelets)



- beat, timber, energy, zero crossings

Features are very important !

Summary: Obtaining features

Very often requires some domain knowledge

As ML algorithm developers, we often have to trust the "experts" to identify and extract reasonable features

That said, it can be helpful to understand where the features are coming from

Features are very important !

Supervised Learning

Machine Learning

Supervised Learning	Unsupervised Learning	Reinforcement Learning
Classification	Clustering	Decision Process
Regression	Segmentation	Reward System
Ranking	Dimension Reduction	Recommendation Systems
	Association Mining	

Supervised learning

apple

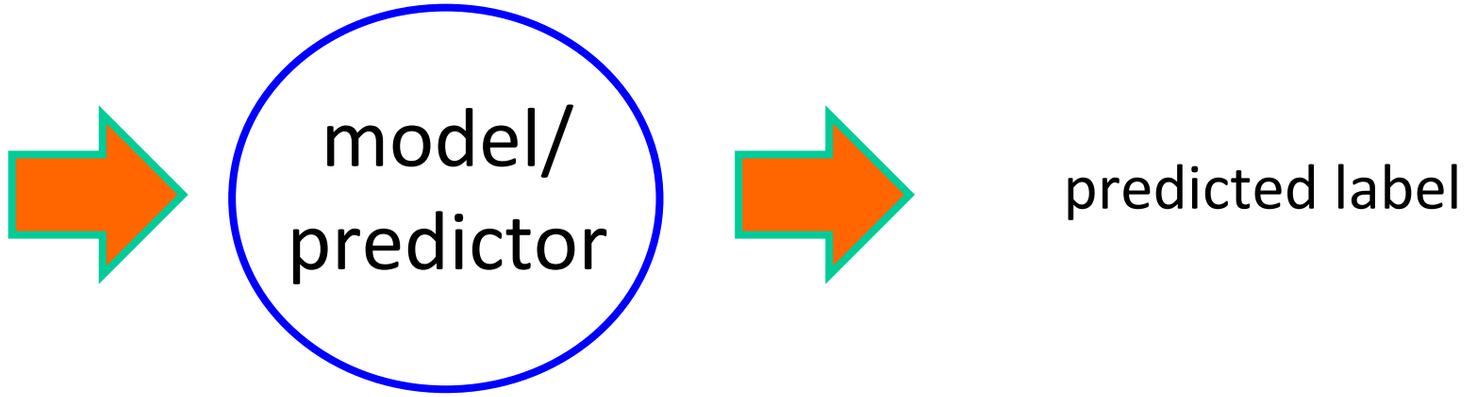
apple

banana

banana

labeled examples

Supervised learning



Supervised learning: learn to predict new example

Supervised learning: classification

label

apple

apple

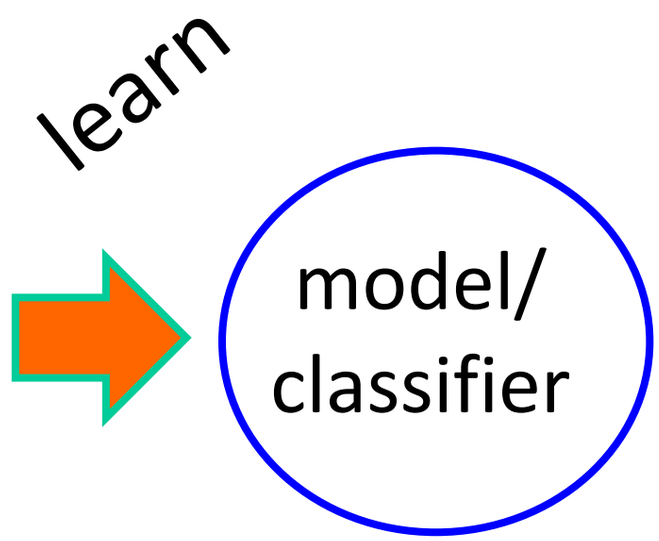
banana

banana

Classification: a finite set of labels

Classification

examples	label
red, round, leaf, 3oz, …	apple
green, round, no leaf, 4oz, …	apple
yellow, curved, no leaf, 4oz, …	banana
green, curved, no leaf, 5oz, …	banana



During learning/training/induction, learn a model of what distinguishes apples and bananas *based on the features*

Classification

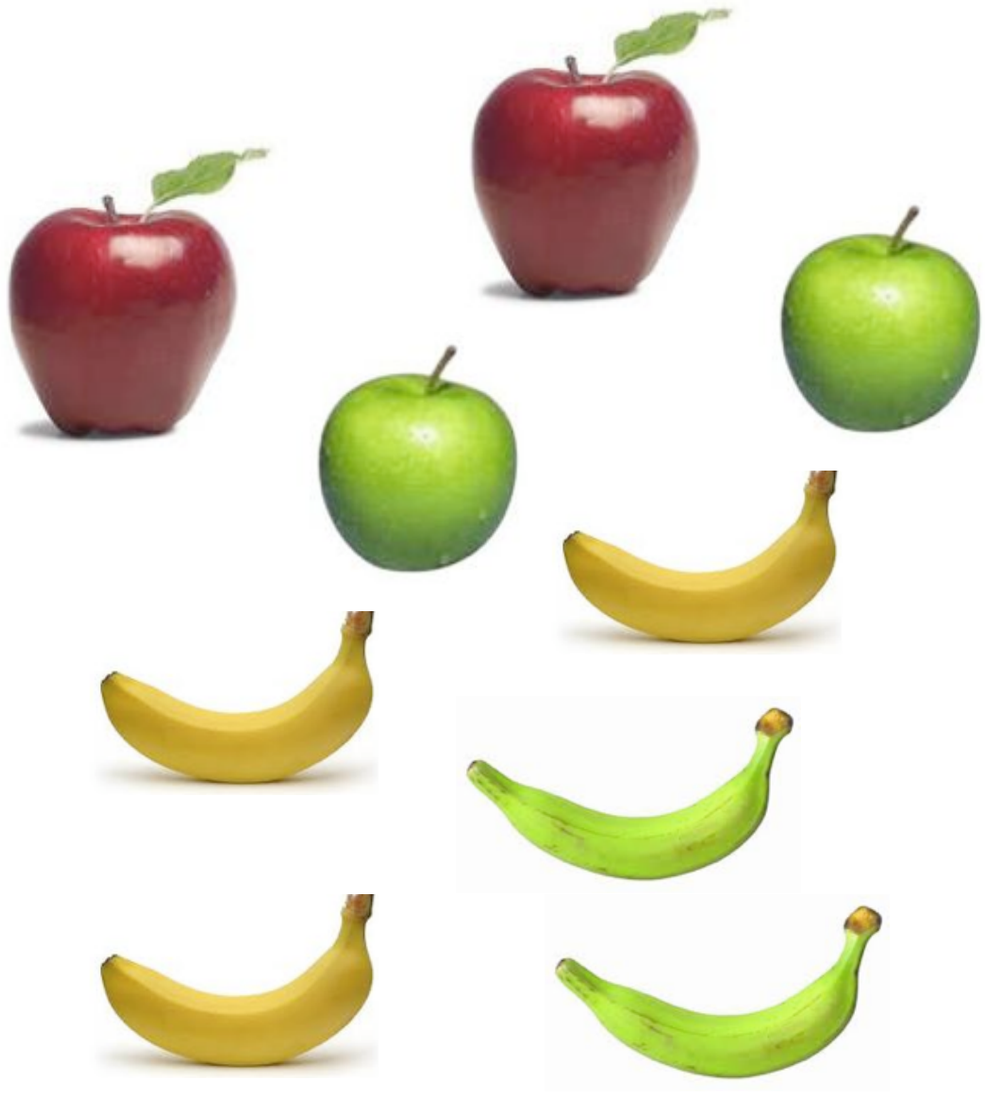
Training data		Test set
examples	label	
red, round, leaf, 3oz, …	apple	
green, round, no leaf, 4oz, …	apple	red, round, no leaf, 4oz, … ?
yellow, curved, no leaf, 4oz, …	banana	
green, curved, no leaf, 5oz, …	banana	

Learning is about ***generalizing*** from the training data

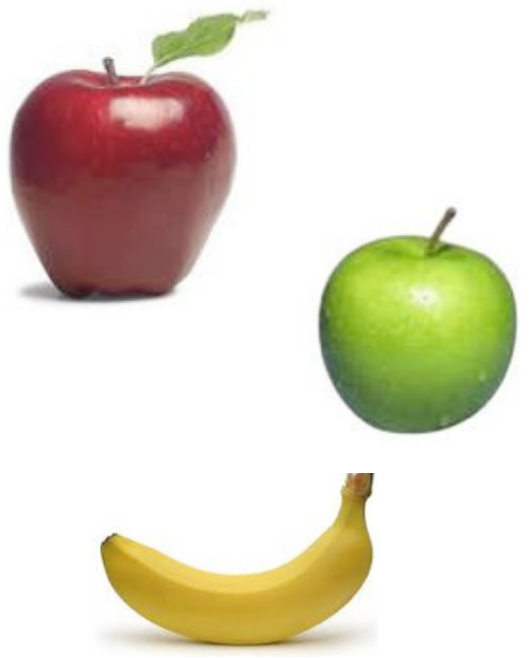
What does this assume about the training and test set?

Past predicts future

Training data

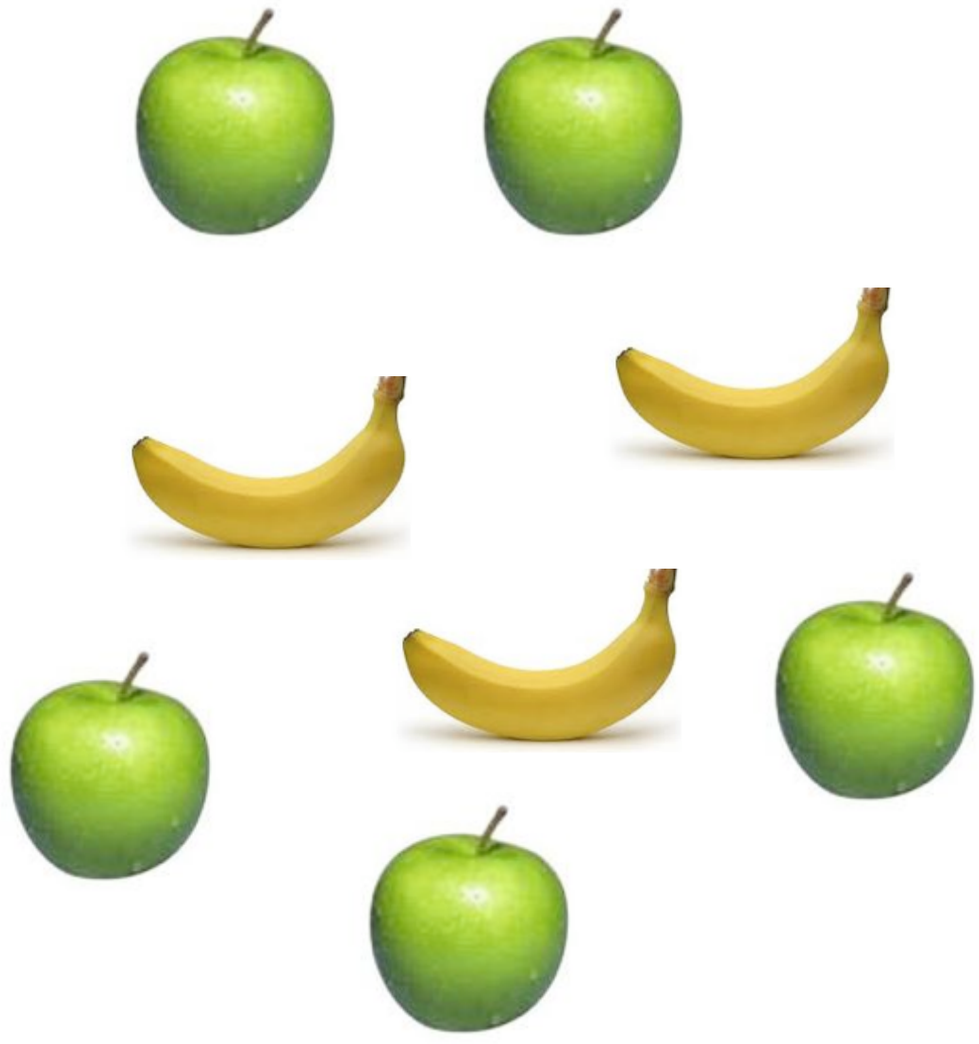


Test set

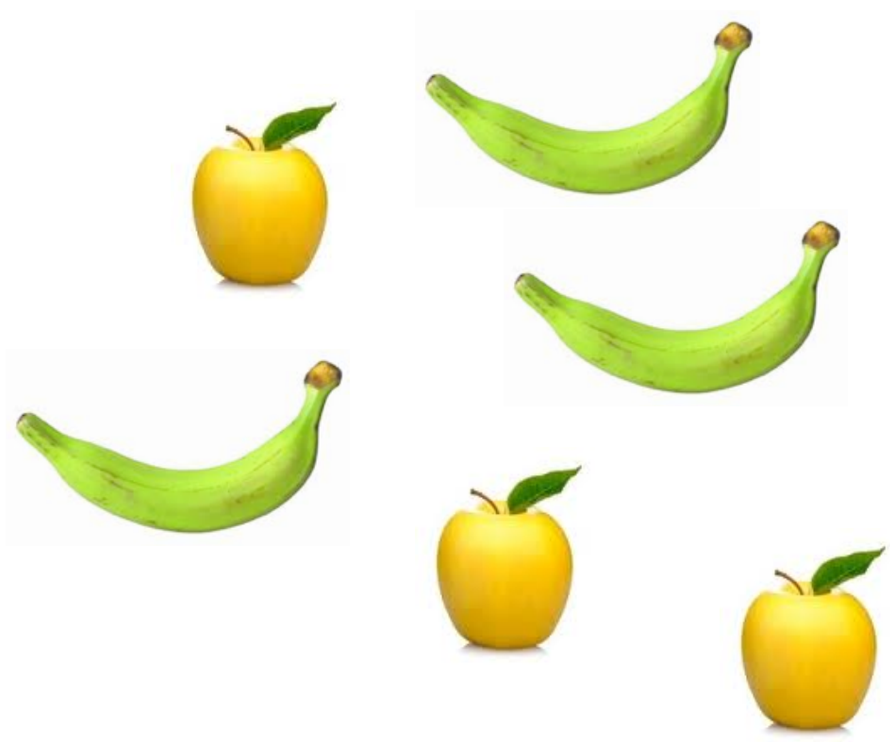


Past predicts future

Training data



Test set



Not always the case, but we'll often assume it is!

Classification Applications

- Face recognition
- Character recognition
- Spam detection
- Medical diagnosis
- Biometrics

Supervised learning: regression

label

-4.5

10.1

3.2

4.3

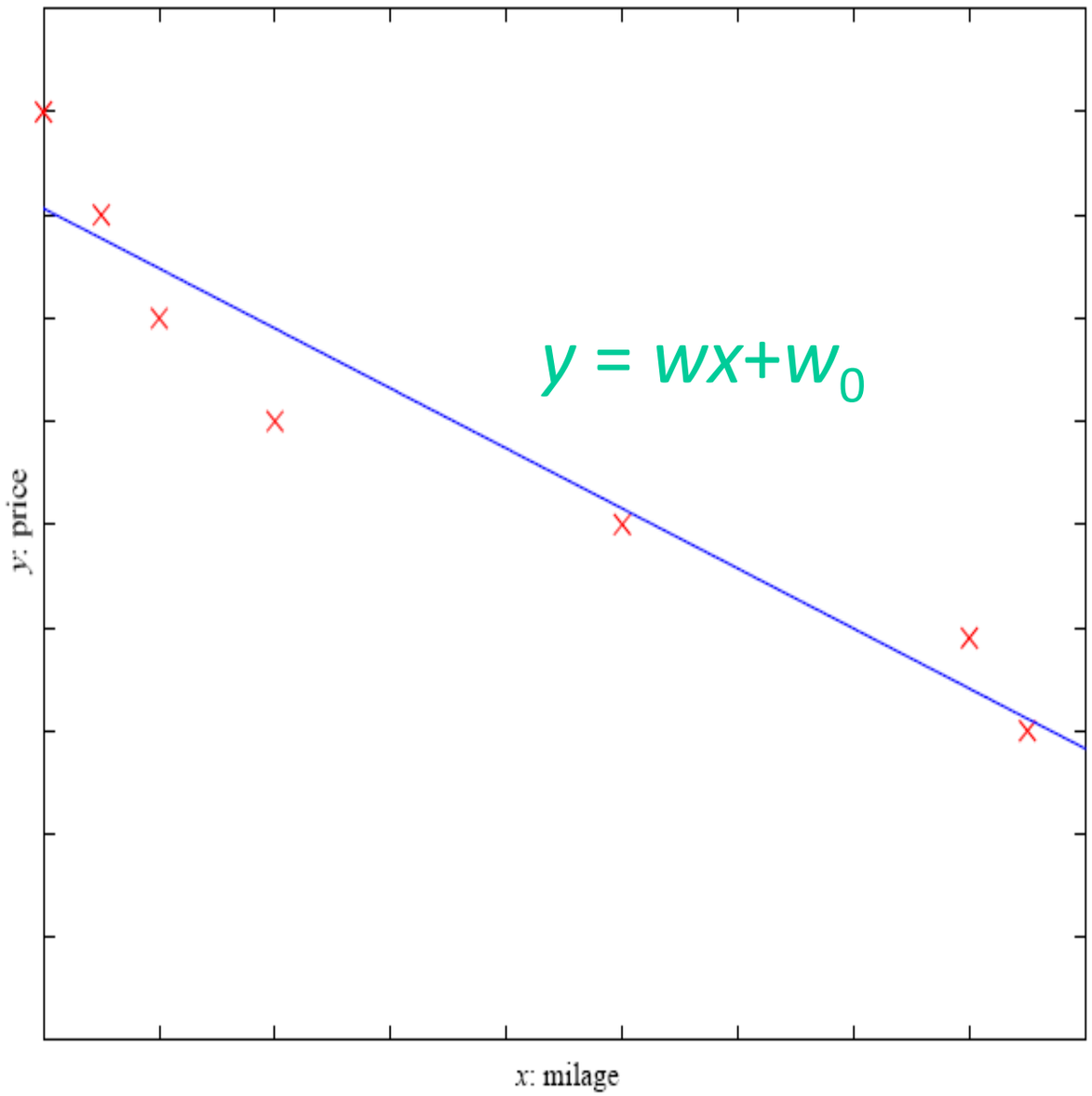
Regression: label is real-valued

Regression example

Price of a used car

x : car attributes
(e.g. mileage)

y : price



Regression Applications

- Economics/Finance: predict the value of a stock
- Epidemiology
- Car/plane navigation: angle of the steering wheel ...
- Temporal trends: weather over time

...

Supervised learning: ranking

label

1

4

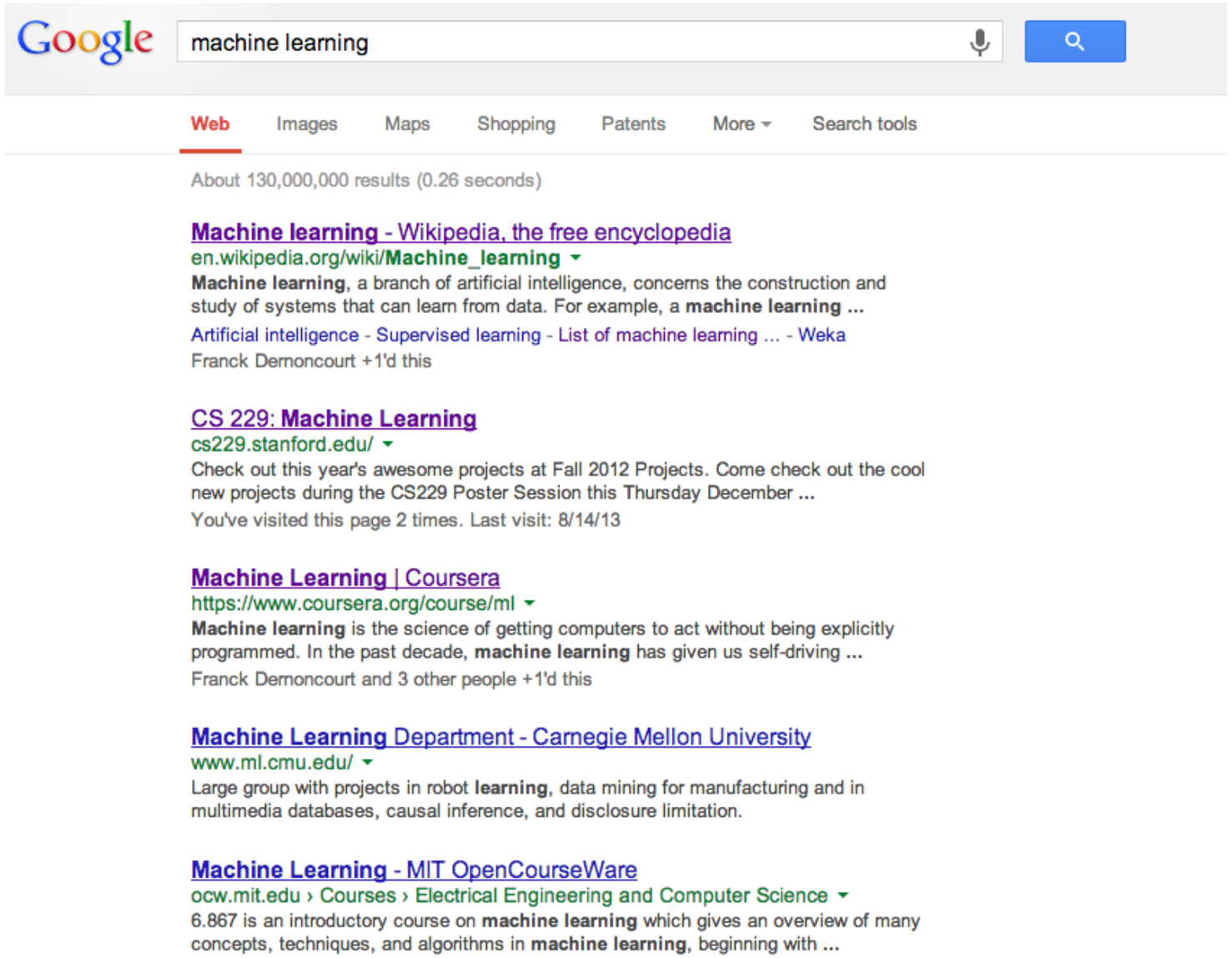
2

3

Ranking: label is a ranking

Ranking example

Given a query and a set of web pages, rank them according to relevance



Ranking Applications

- User preference, e.g. Netflix “My List”
- flight search (search in general)
- reranking N-best output lists
- …

Example: k-Nearest Neighbors

Apples vs. Bananas

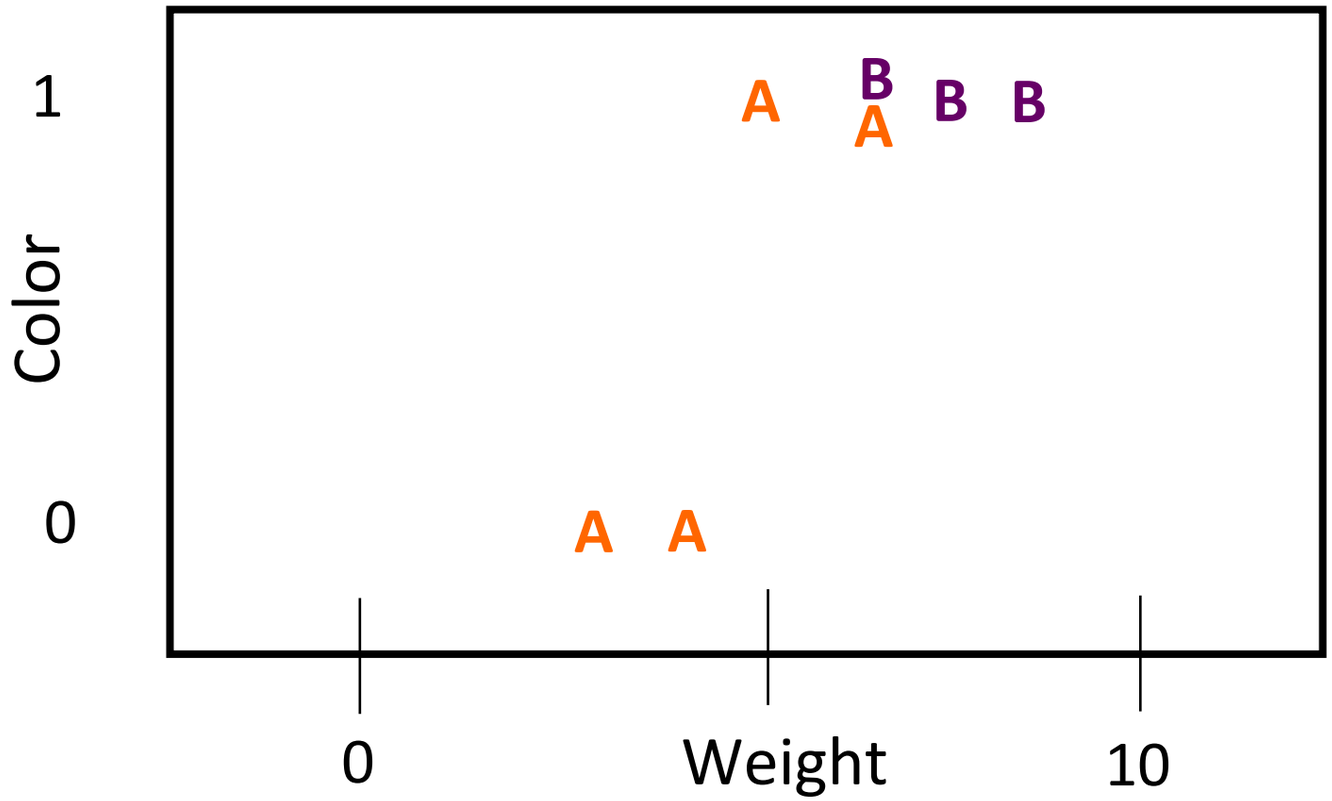
Weight	Color	Label
4	Red	Apple
5	Yellow	Apple
6	Yellow	Banana
3	Red	Apple
7	Yellow	Banana
8	Yellow	Banana
6	Yellow	Apple

Can we visualize this data?

Apples vs. Bananas

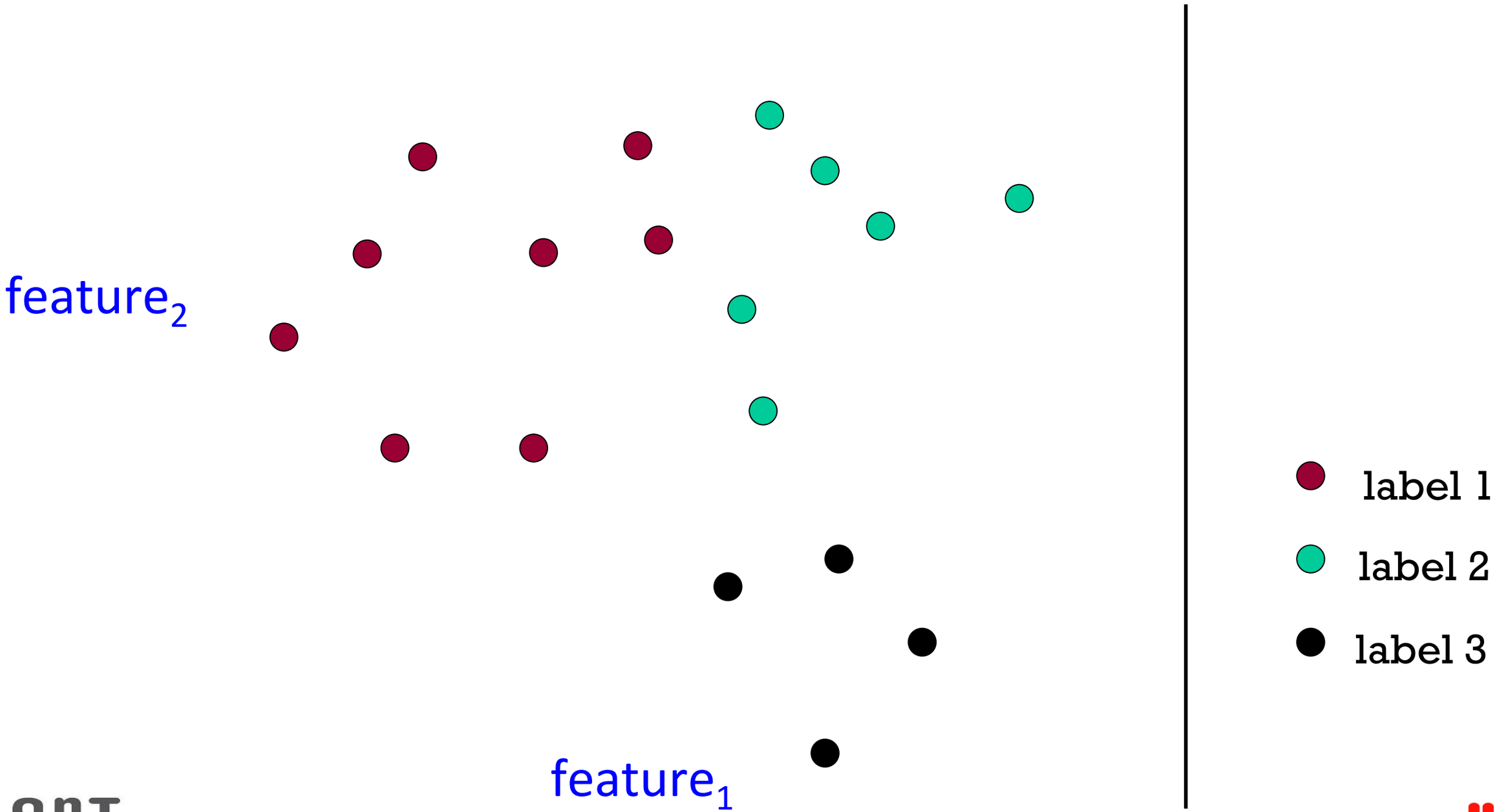
Turn features into numerical values

Weight	Color	Label
4	0	Apple
5	1	Apple
6	1	Banana
3	0	Apple
7	1	Banana
8	1	Banana
6	1	Apple

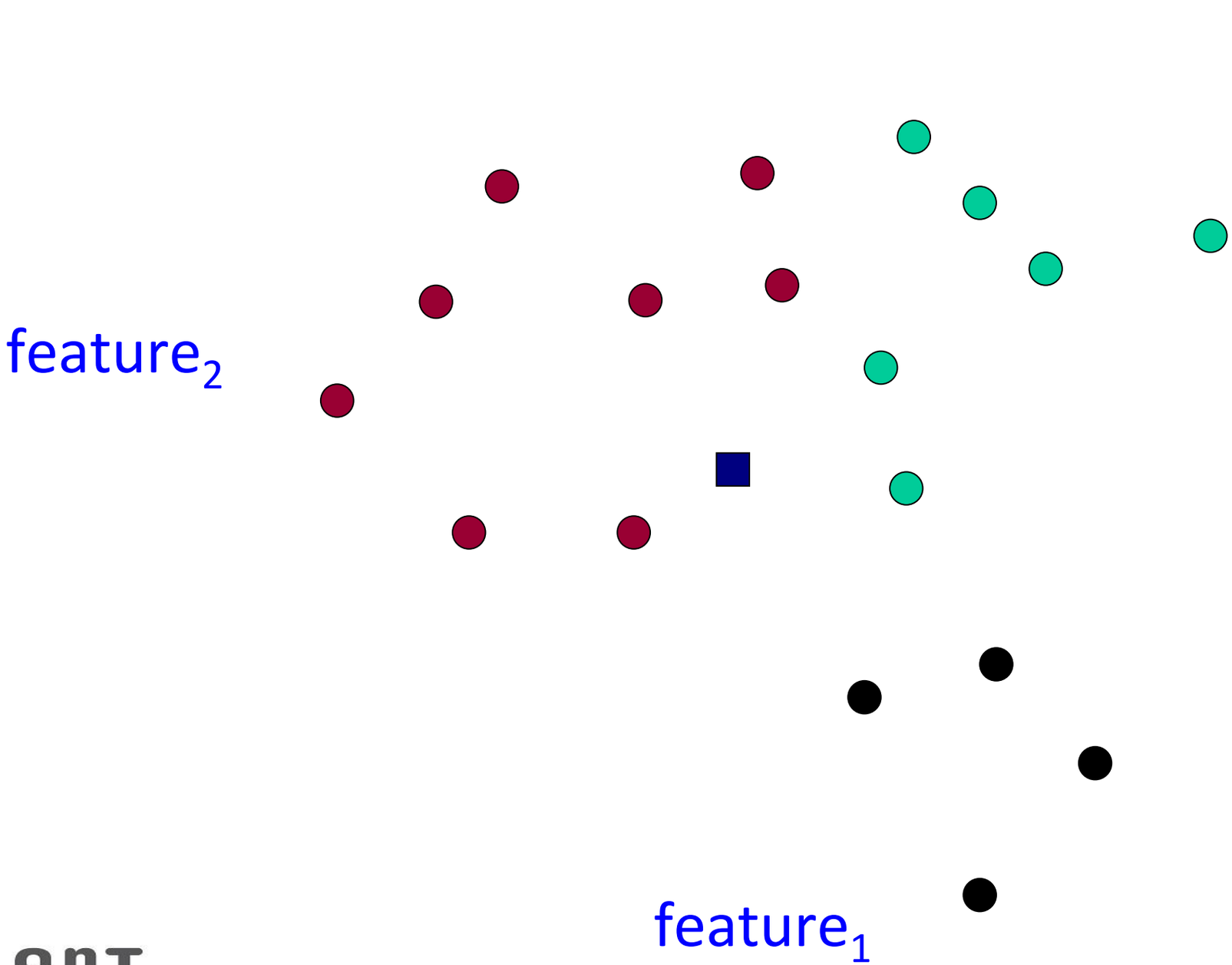


We can view examples as points in an *n*-dimensional space where *n* is the number of features

Examples in a feature space



Test data: what class?

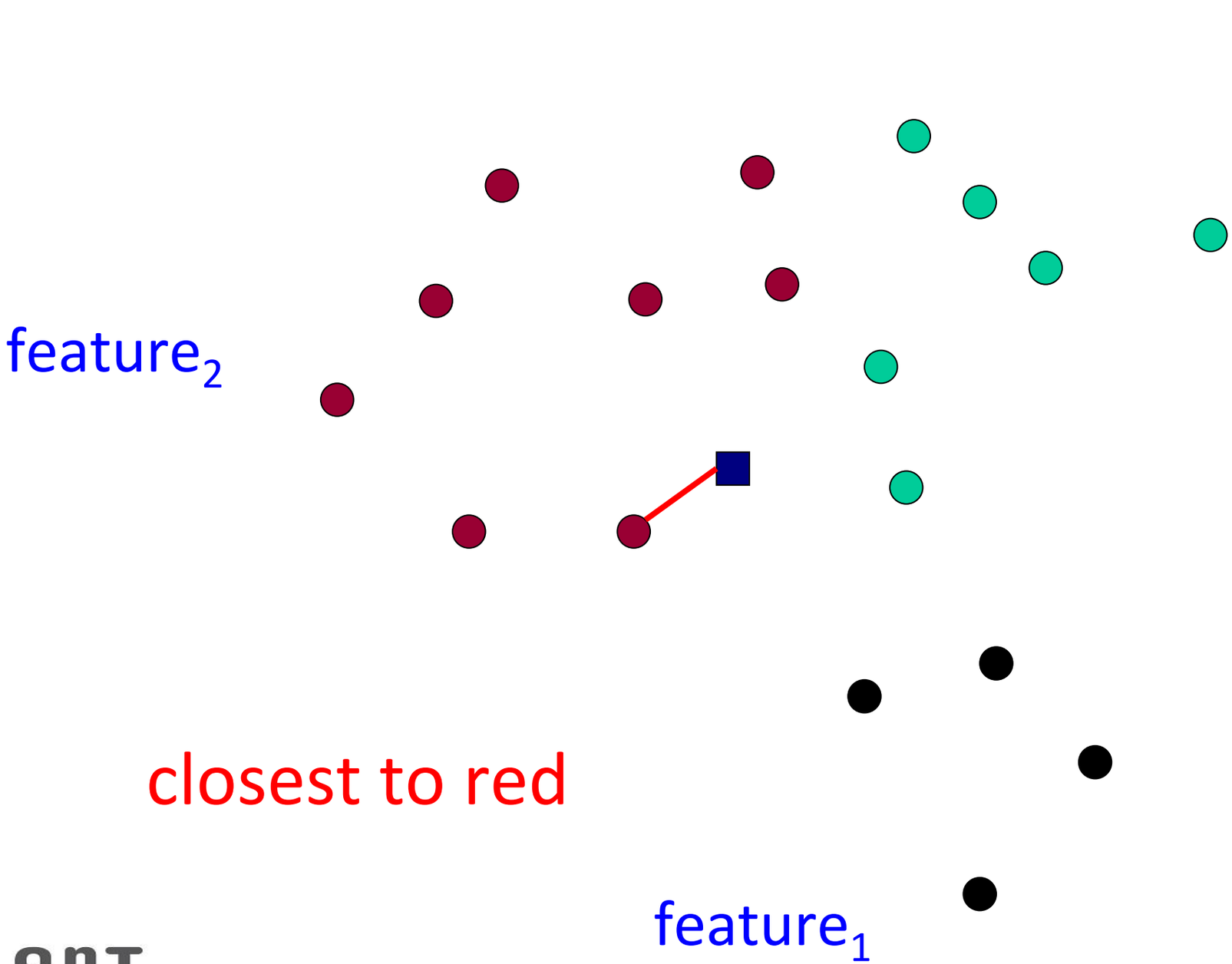


label 1

label 2

label 3

Test data: what class?

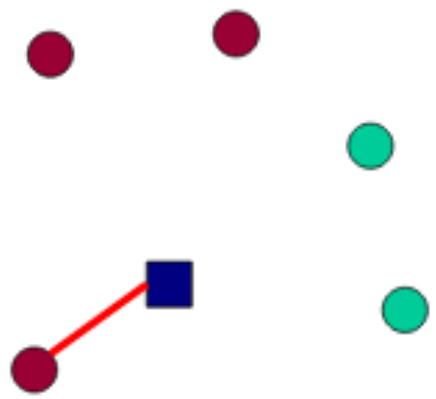


label 1
label 2
label 3

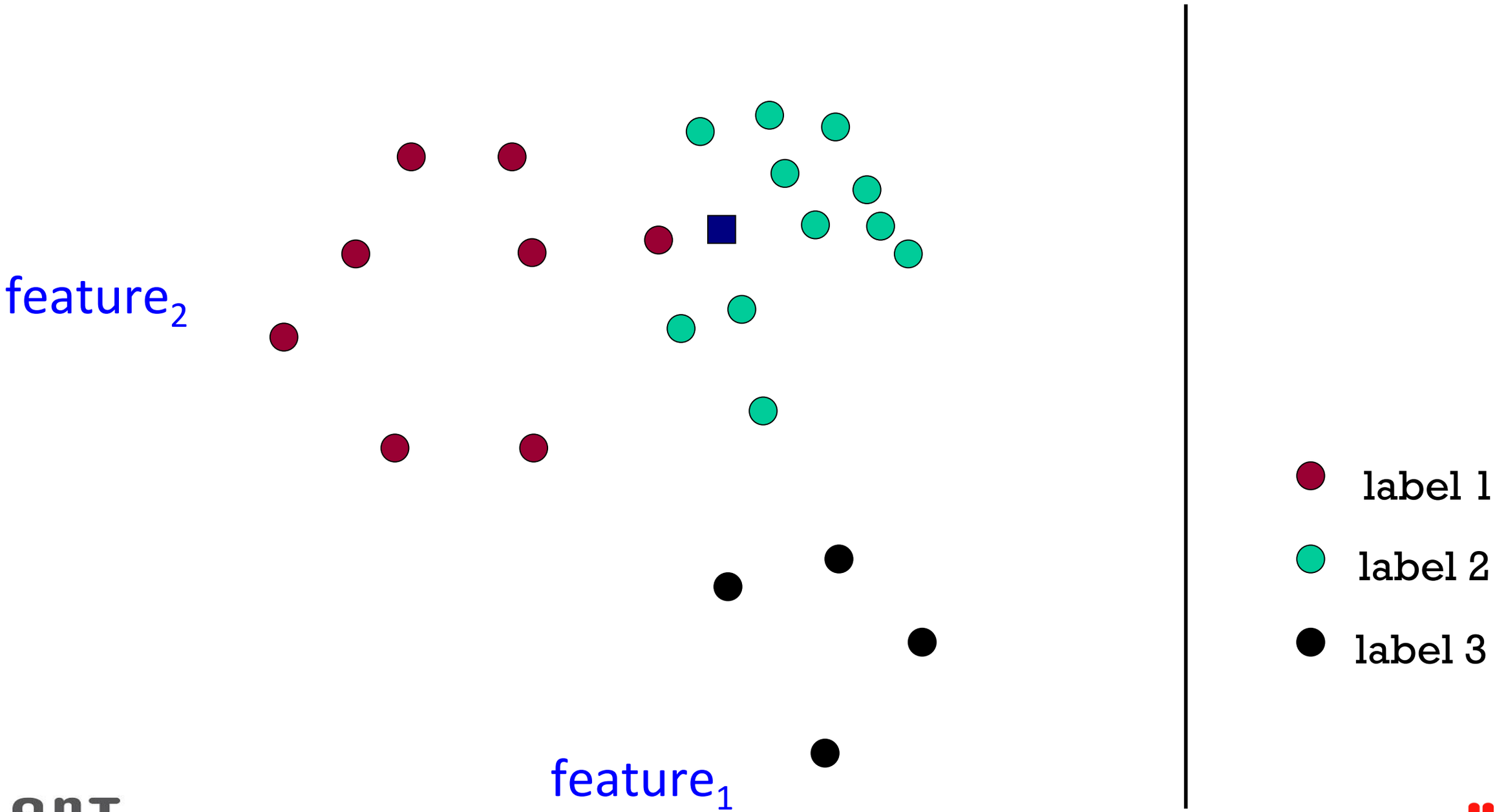
Nearest Neighbor

To classify an example ***d***:

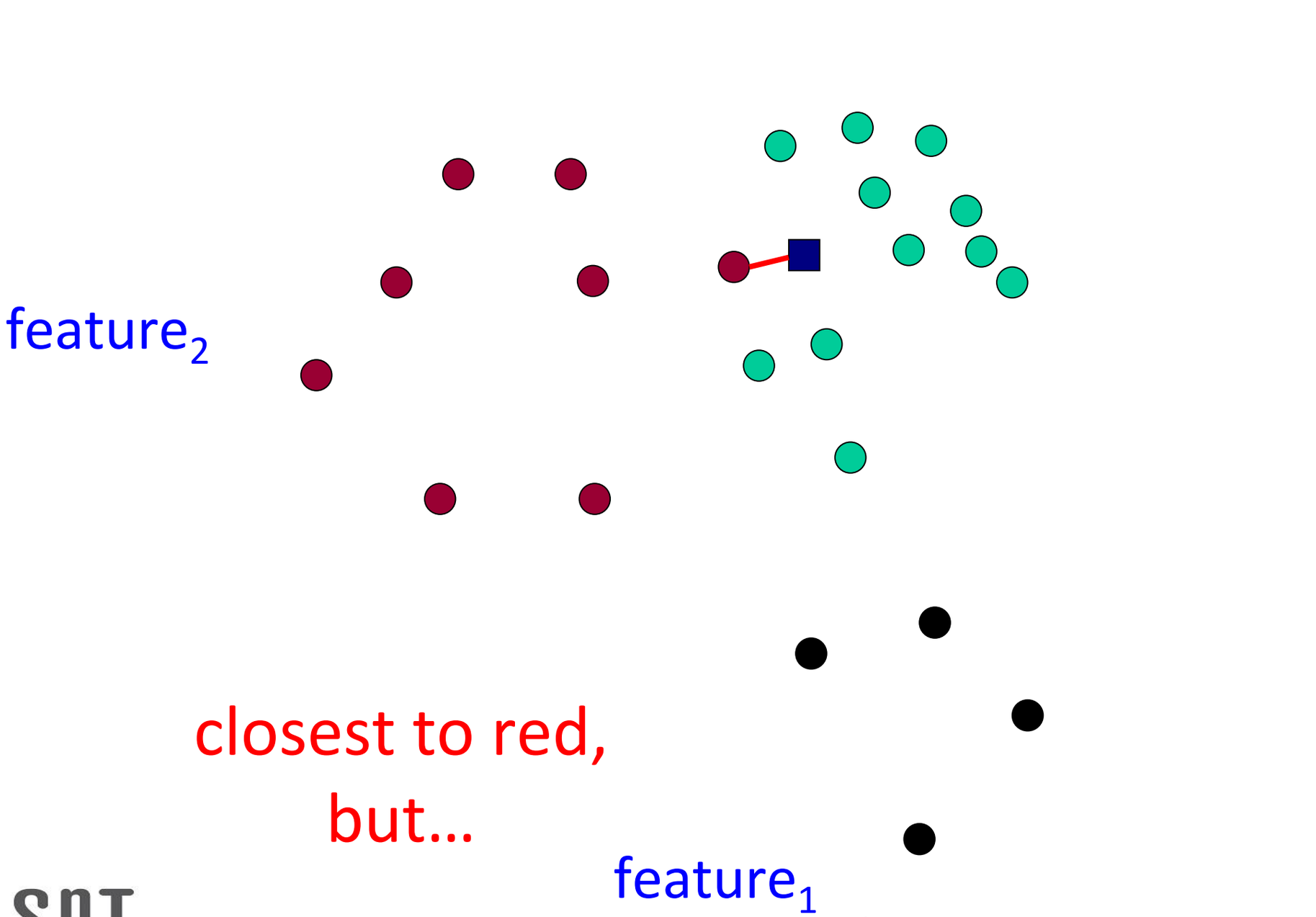
- Label ***d*** with the label of the closest example to ***d***



What about this example?

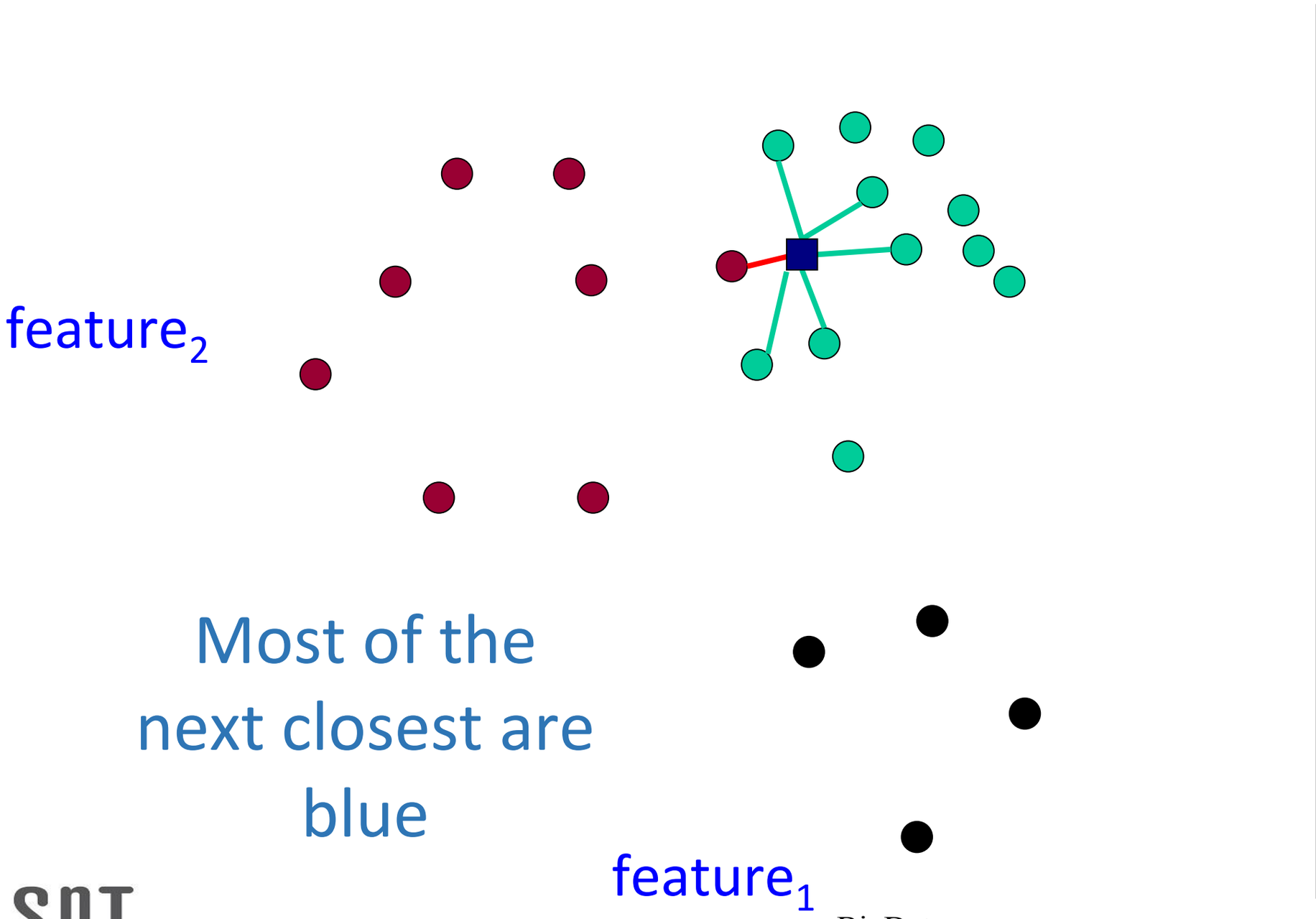


What about this example?



label 1
label 2
label 3

What about this example?

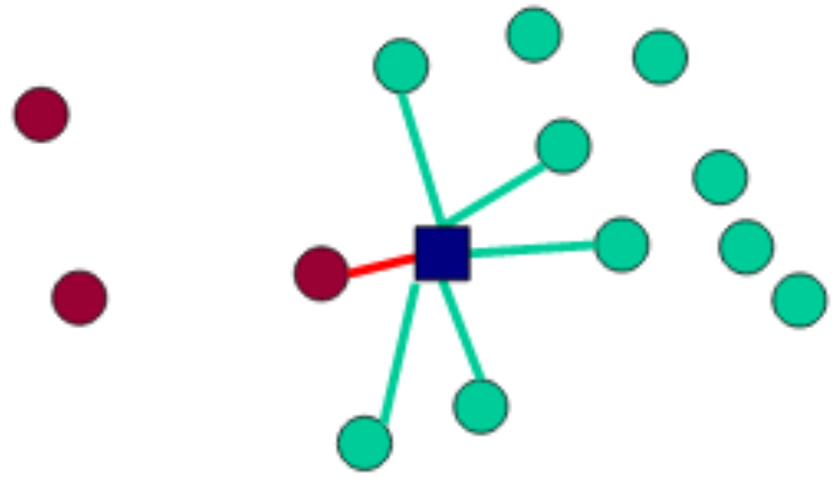


label 1
label 2
label 3

k-Nearest Neighbor (k-NN)

To classify an example ***d***:

- Find ***k*** nearest neighbors of ***d***
- Choose as the label the majority label within the ***k*** nearest neighbors



k-Nearest Neighbor (k-NN)

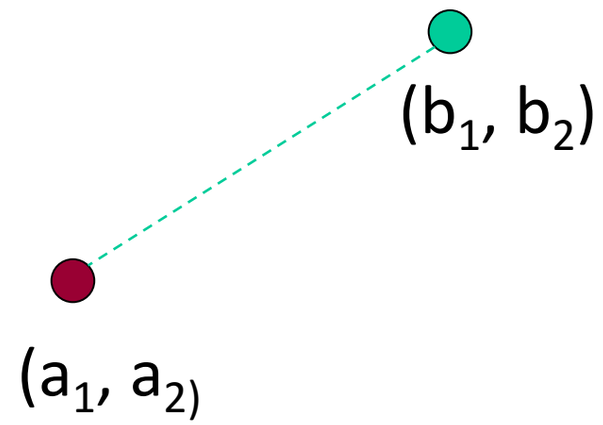
To classify an example ***d***:

- Find ***k*** *nearest* neighbors of ***d***
- Choose as the label the majority label within the ***k*** nearest neighbors

How do we measure “nearest”?

Euclidean distance

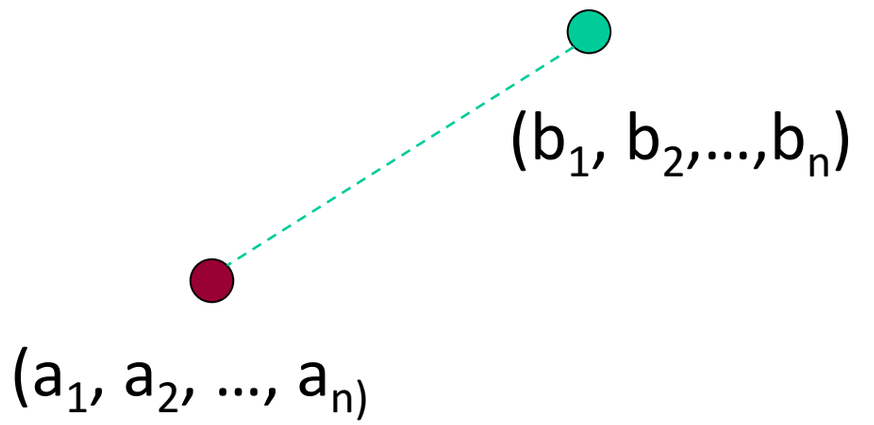
In two dimensions, how do we compute the distance?



$$D(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

Euclidean distance

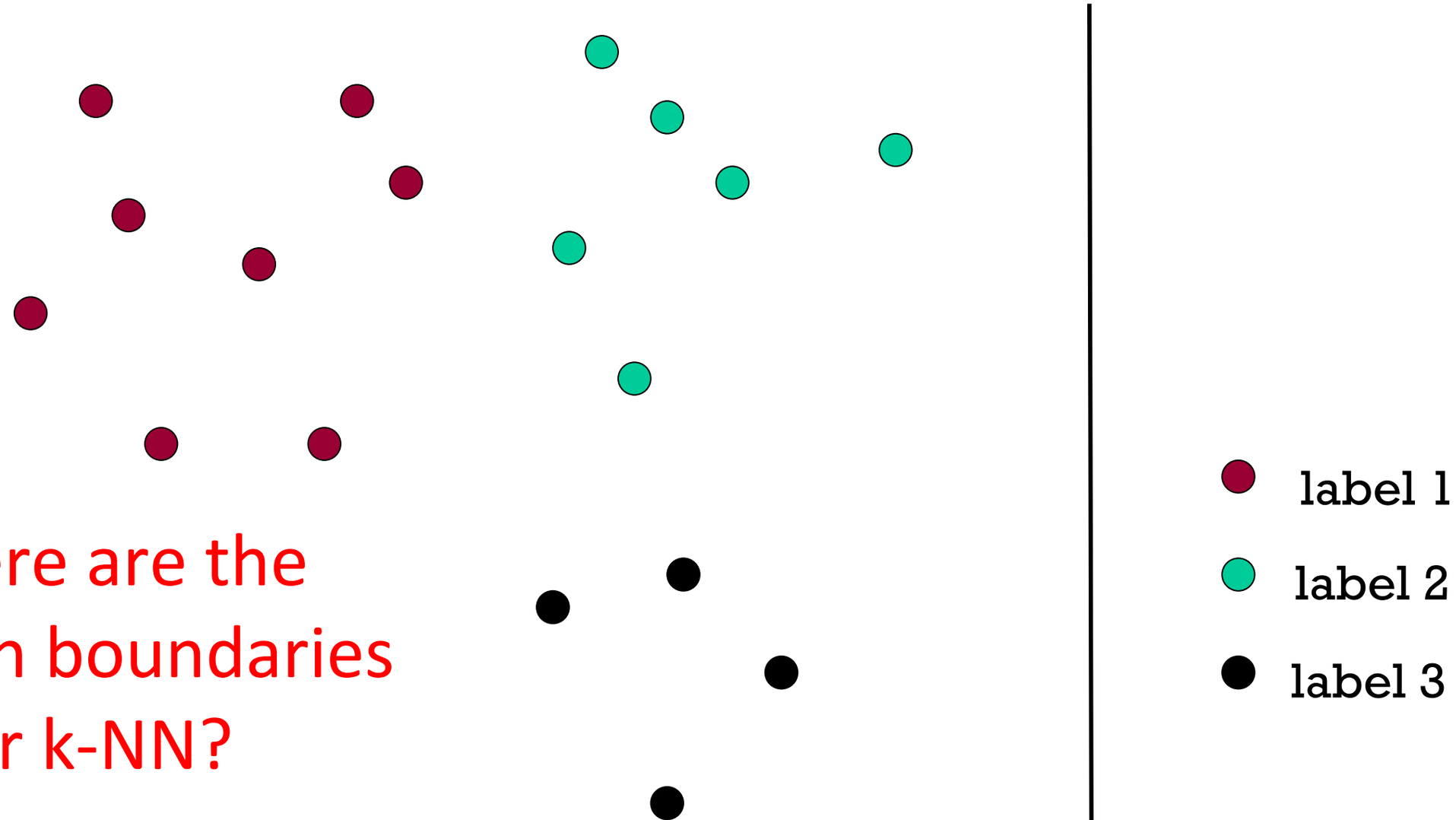
In n-dimensions, how do we compute the distance?



$$D(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_n - b_n)^2}$$

Decision boundaries

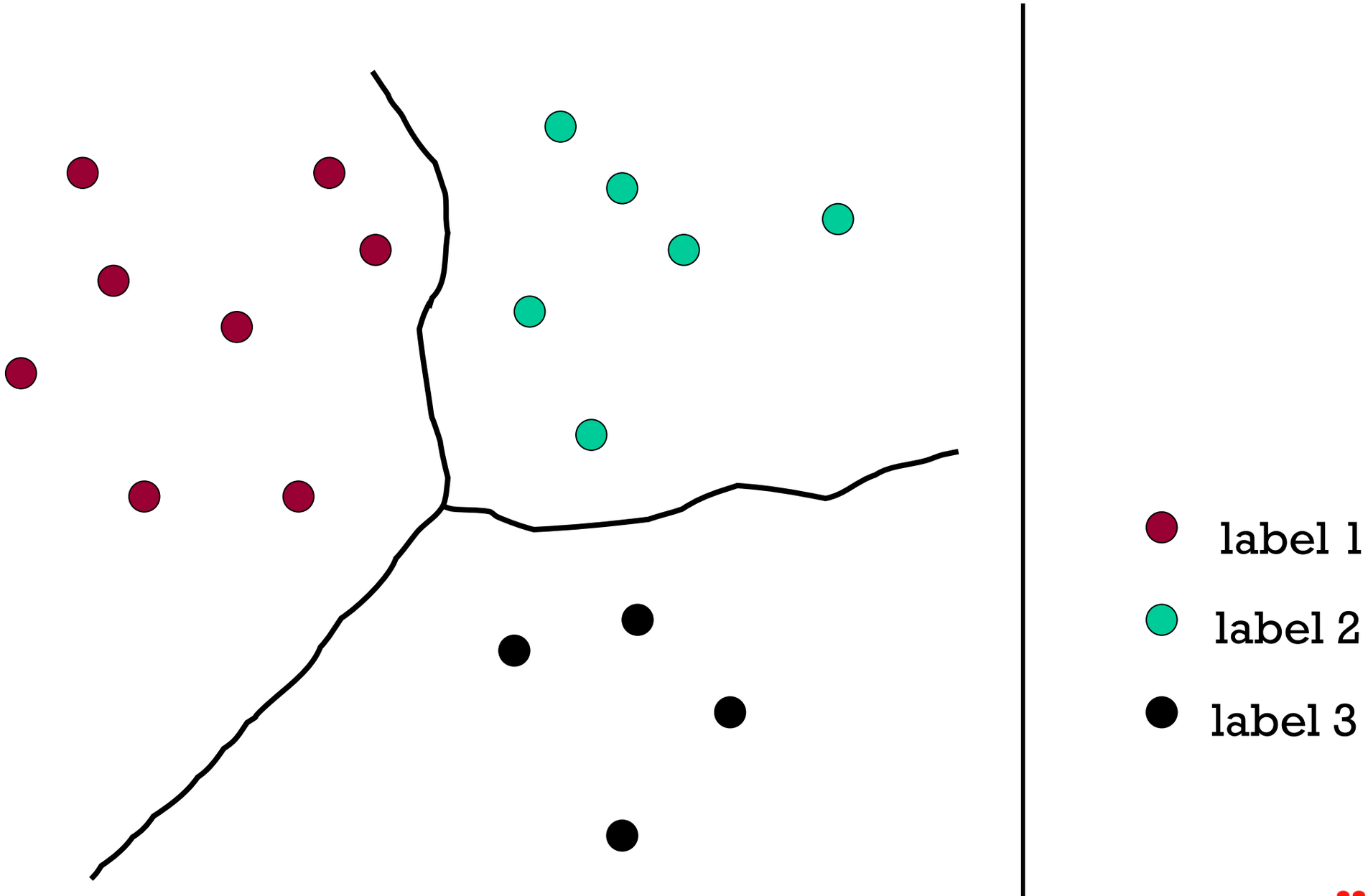
The **decision boundaries** are places in the features space where the classification of a point/example changes



Where are the decision boundaries for k-NN?

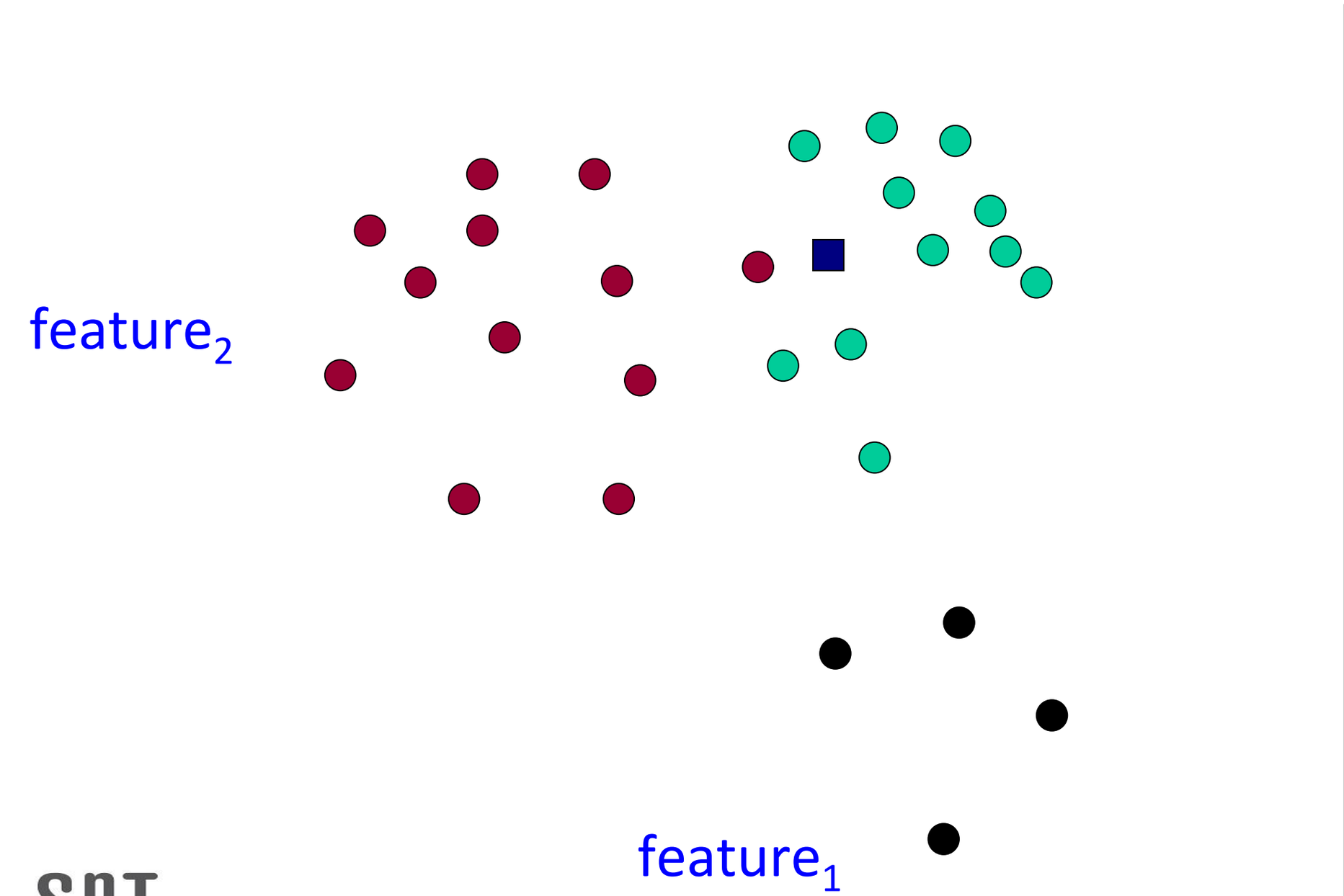
k-NN decision boundaries

k-NN gives locally defined decision boundaries between classes



Choosing k

What is the label with k = 1?



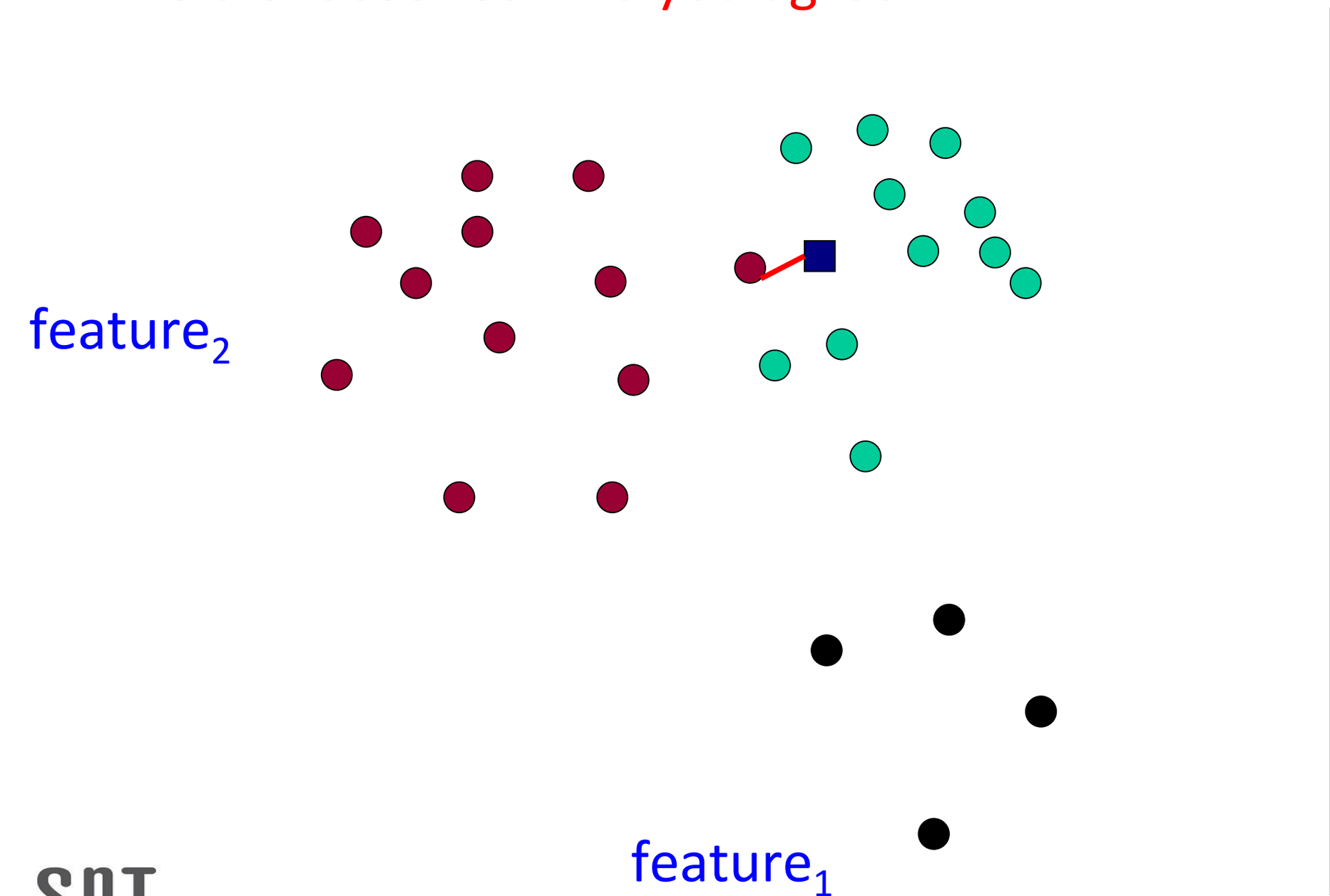
label 1

label 2

label 3

Choosing k

We'd choose red. Do you agree?



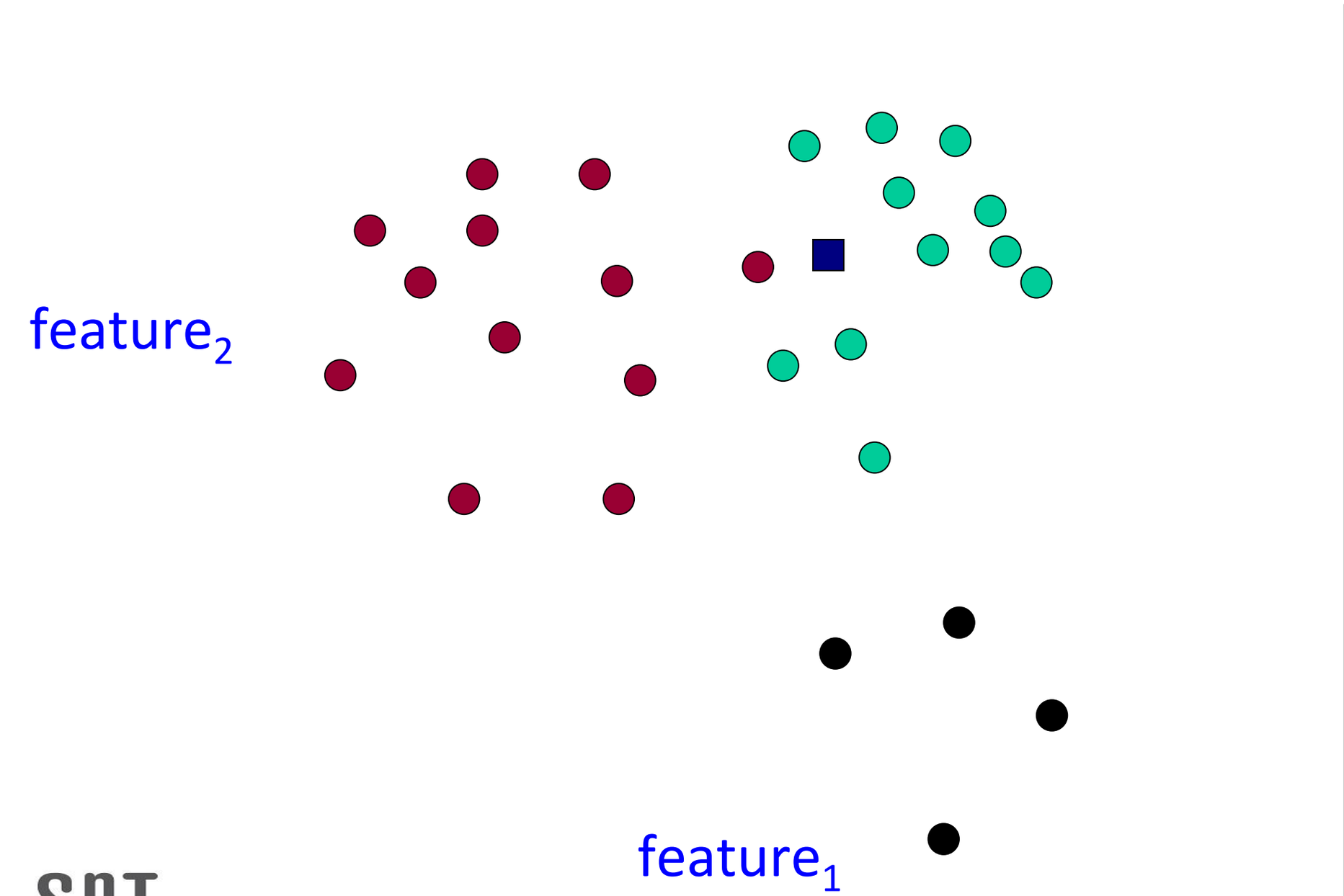
label 1

label 2

label 3

Choosing k

What is the label with k = 3?



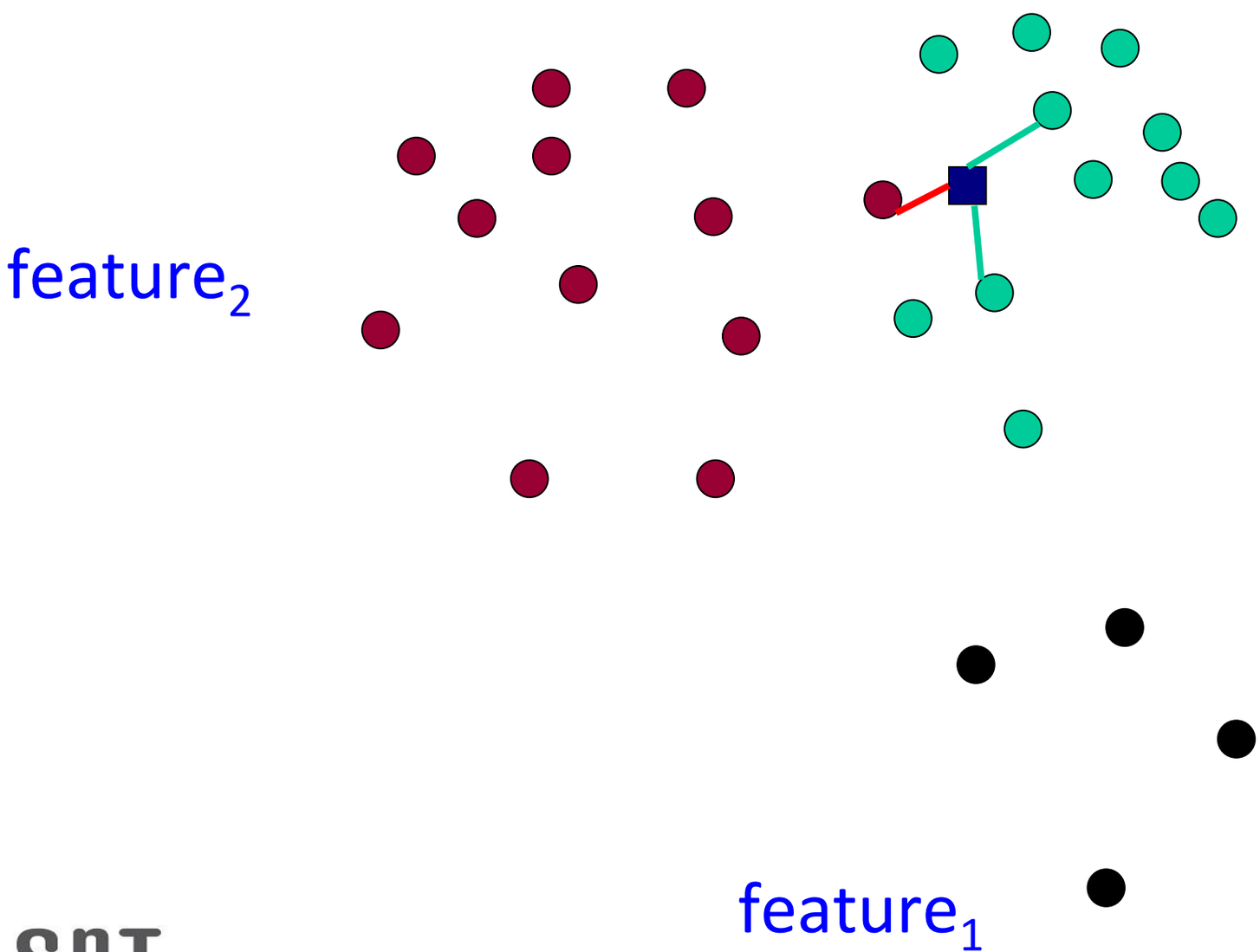
label 1

label 2

label 3

Choosing k

We'd choose blue. Do you agree?



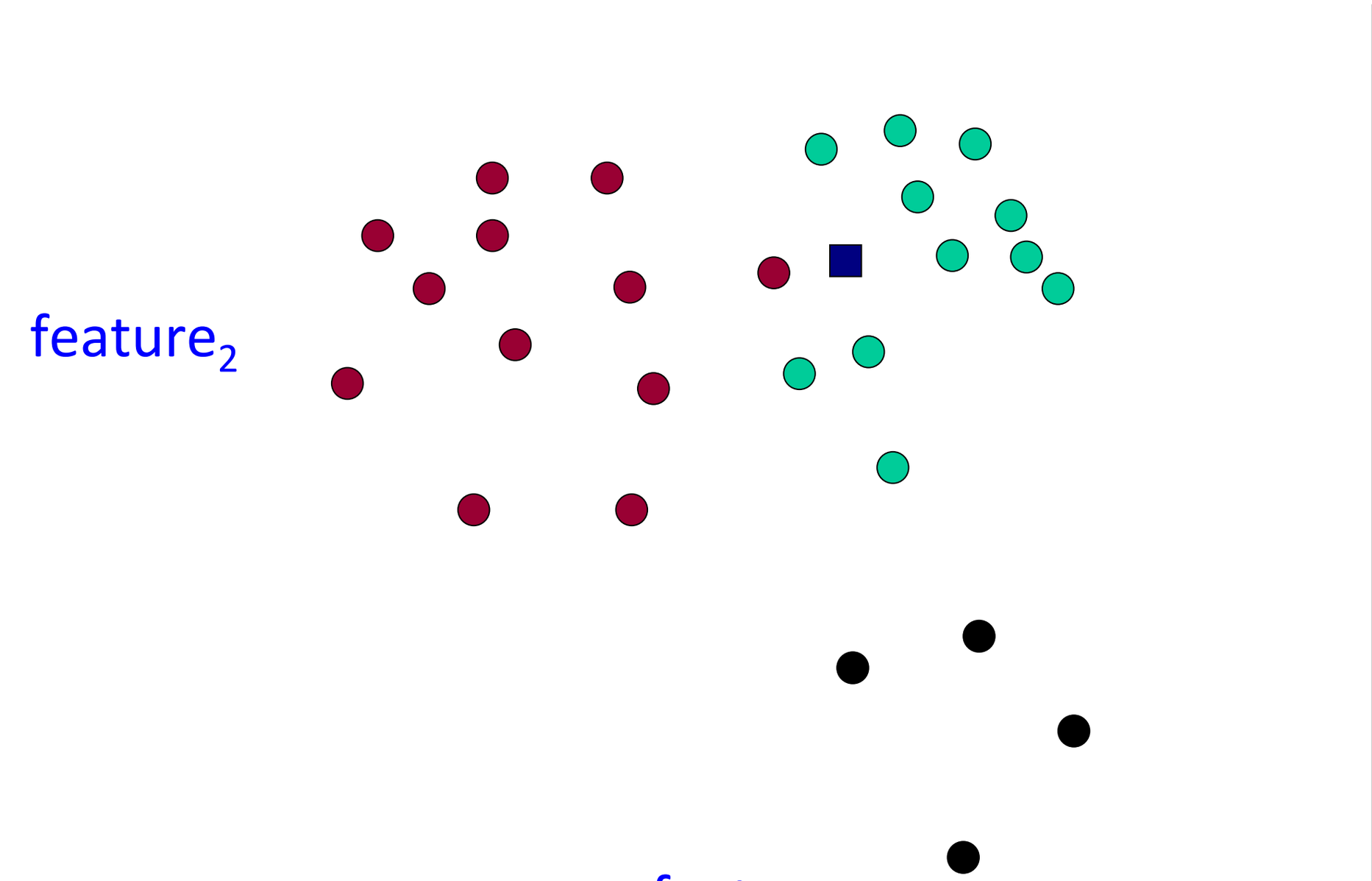
label 1

label 2

label 3

Choosing k

What is the label with k = 100?



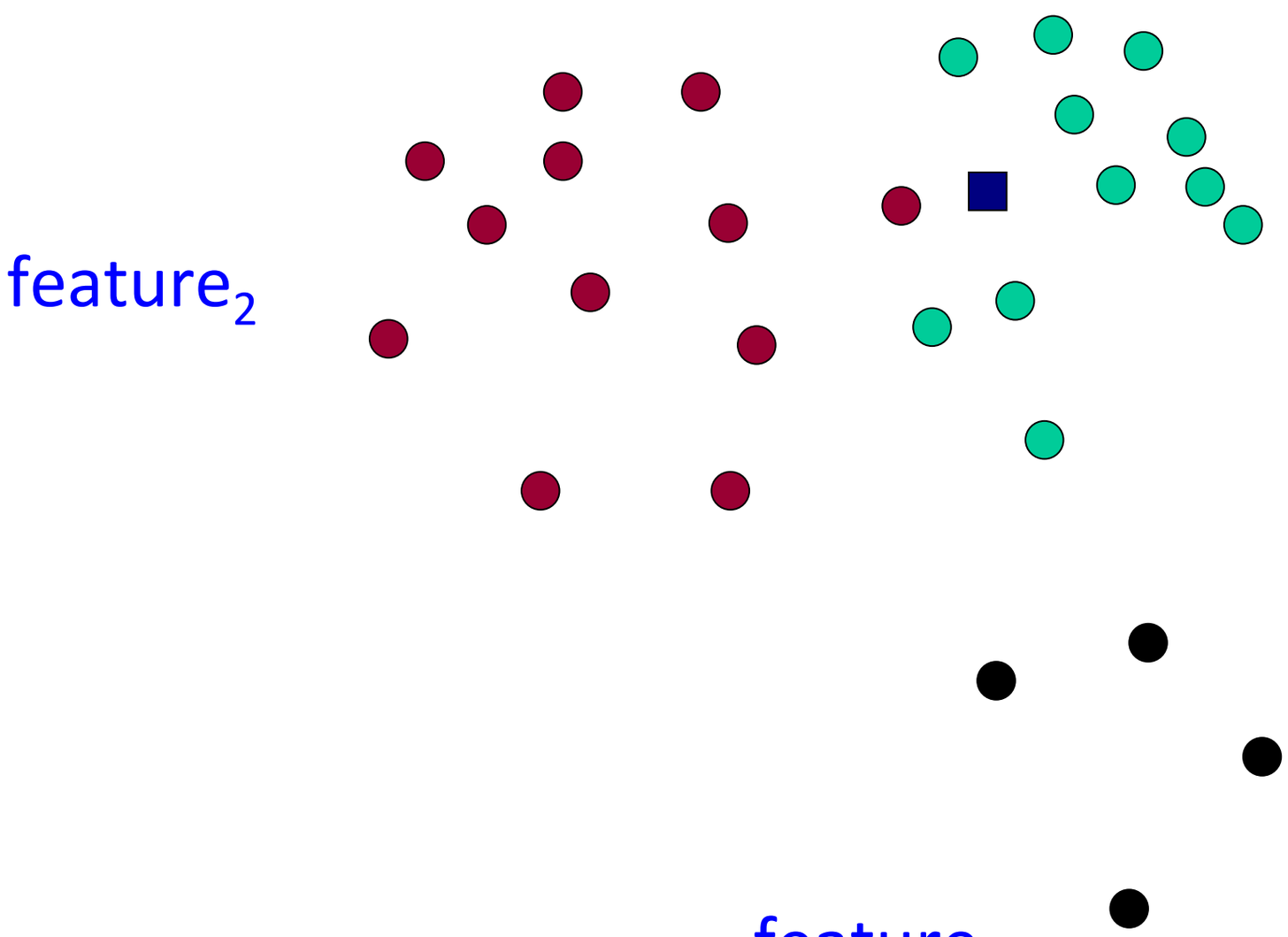
label 1

label 2

label 3

Choosing k

We'd choose red. Do you agree?

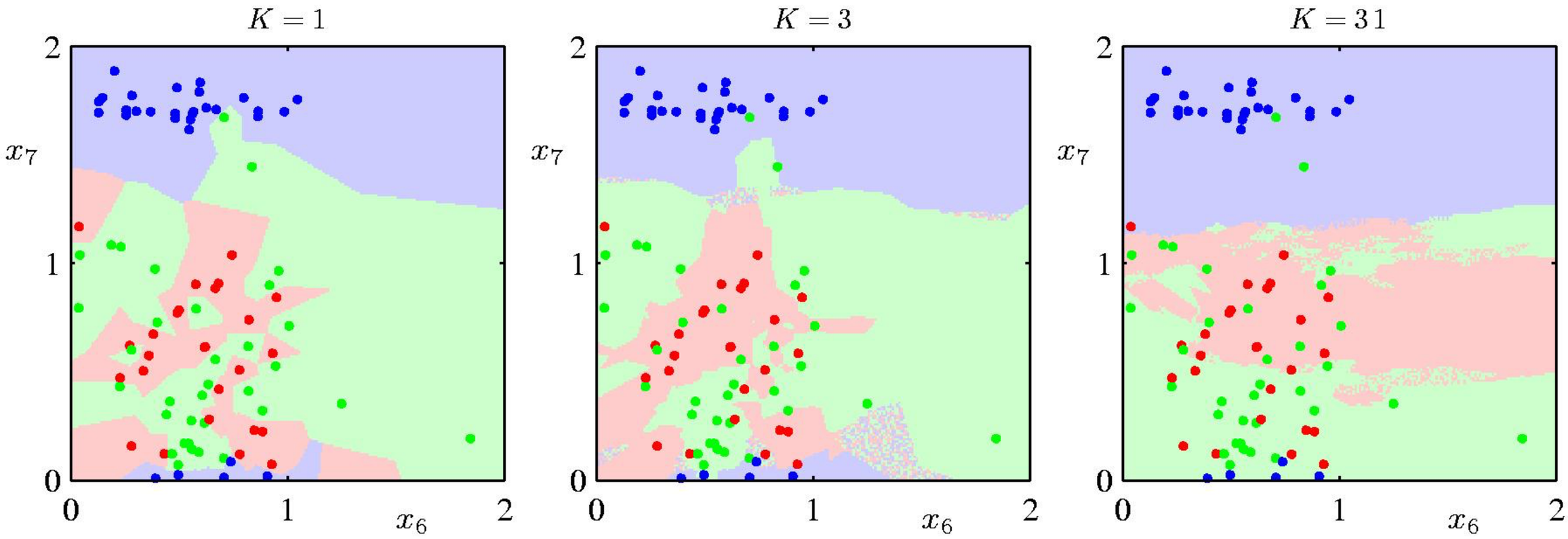


label 1

label 2

label 3

The impact of k



What is the role of k?

How to pick k

Common heuristics:

- often 3, 5, 7
- choose an odd number to avoid ties

Use development data

Summary K-Nearest Neighbor

- **Pros**
  - k-NN is simple! (to understand, implement)
  - Often used as a baseline for other algorithms
  - “Training” is fast: just add new item to database
- **Cons**
  - Most work done at query time: may be expensive
  - Must store O(n) data for later queries
  - Performance is sensitive to choice of distance metric
  - And normalization of feature values… (see next chapter)

Example: Decision Tree

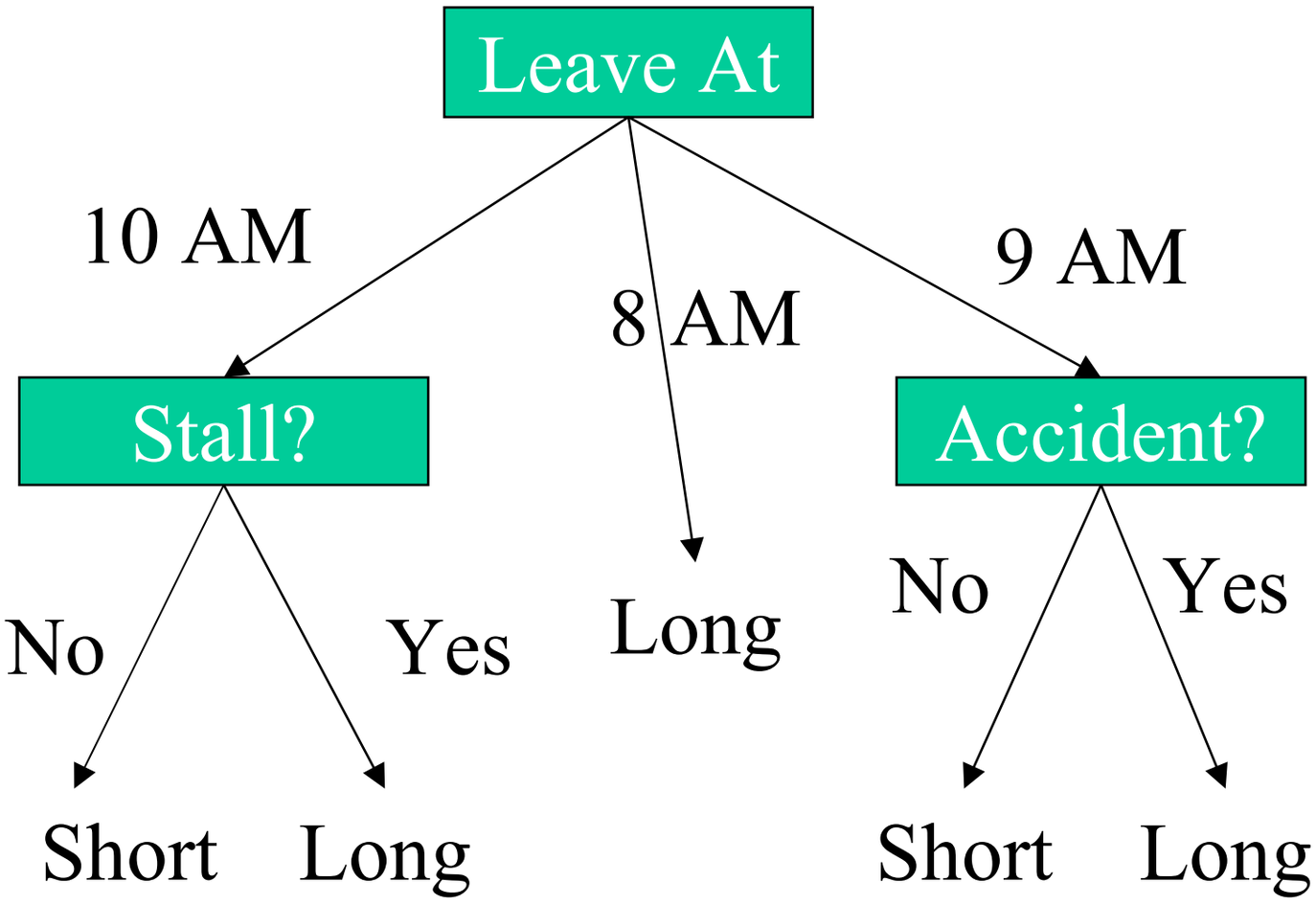
A sample data set

Features				**Label**
Hour	Weather	Accident	Stall	Commute
8 AM	Sunny	No	No	Long
8 AM	Cloudy	No	Yes	Long
10 AM	Sunny	No	No	Short
9 AM	Rainy	Yes	No	Long
9 AM	Sunny	Yes	Yes	Long
10 AM	Sunny	No	No	Short
10 AM	Cloudy	No	No	Short
9 AM	Sunny	Yes	No	Long
10 AM	Cloudy	Yes	Yes	Long
10 AM	Rainy	No	No	Short
8 AM	Cloudy	Yes	No	Long
9 AM	Rainy	No	No	Short

8 AM, Rainy, Yes, No?
10 AM, Rainy, No, No?

Can you describe a “model” that could be used to make decisions in general?

Decision trees

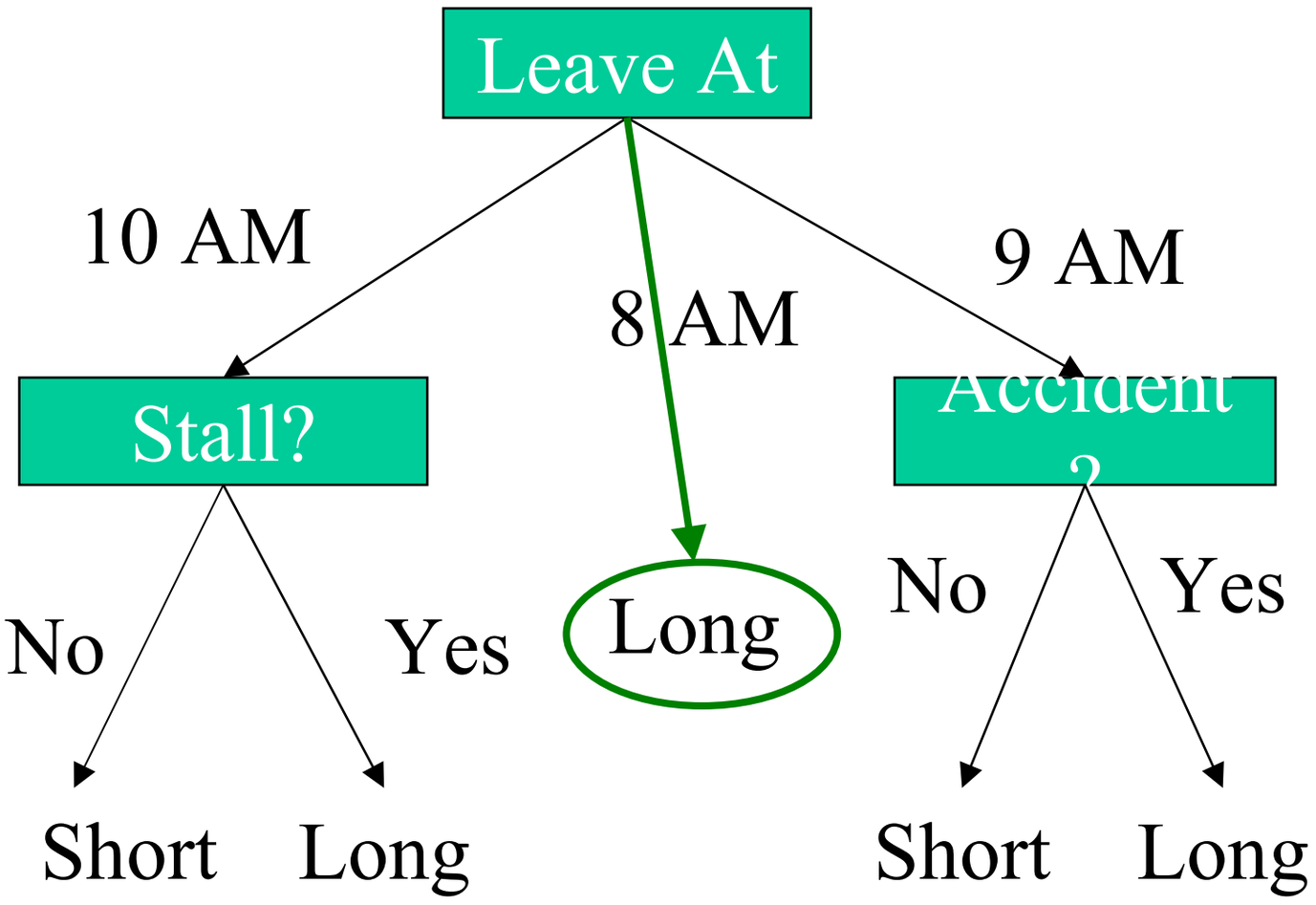


Tree with internal nodes labeled by features

Branches are labeled by tests on that feature

Leaves labeled with classes

Decision trees



Leave = 8 AM
Weather = Rainy

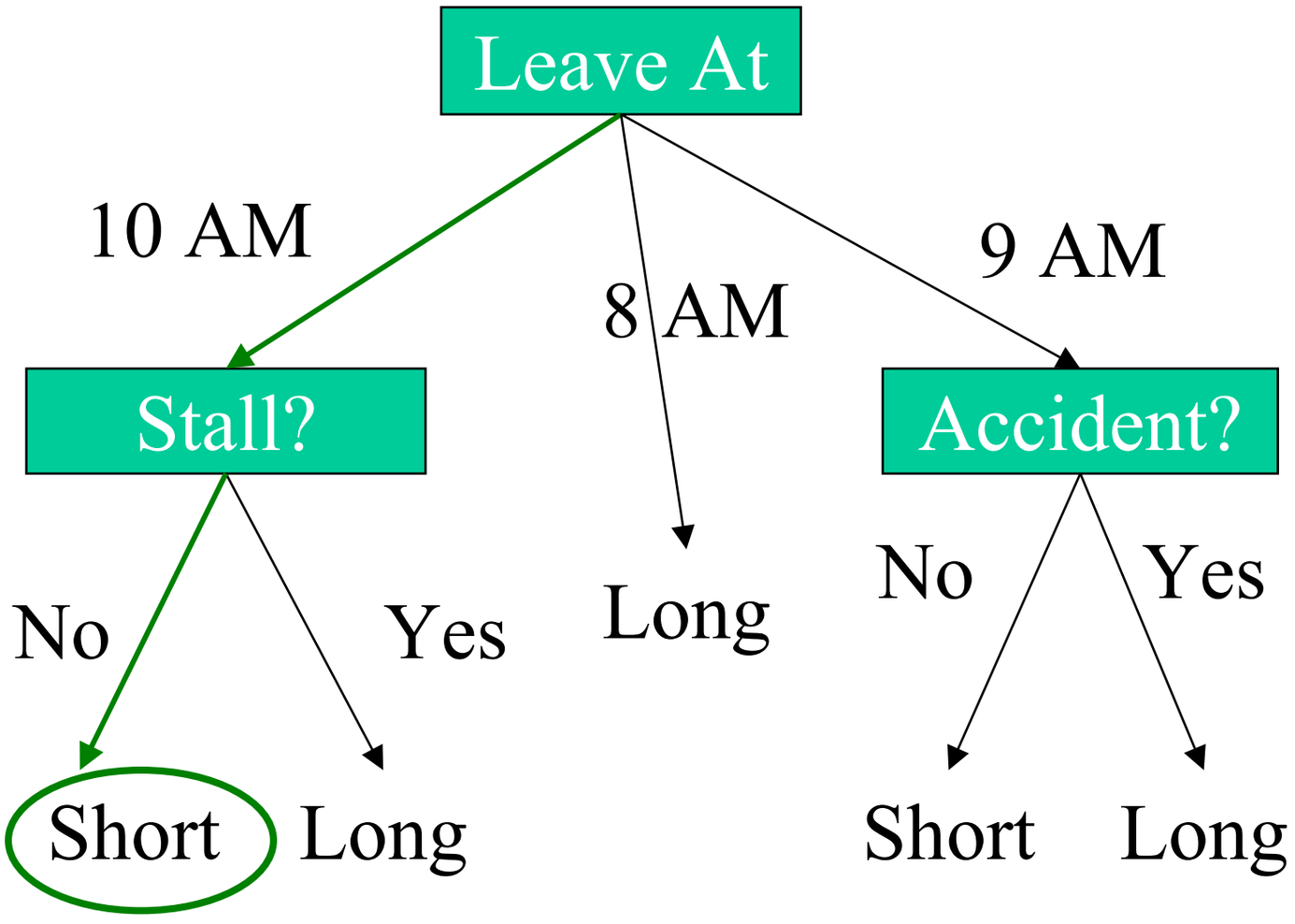
Accident = Yes
Stall = No

Tree with internal nodes labeled by features

Branches are labeled by tests on that feature

Leaves labeled with classes

Decision trees



Tree with internal nodes labeled by features

Branches are labeled by tests on that feature

Leaves labeled with classes

Leave = 10 AM
Weather = Rainy

Accident = No
Stall = No

Recursive approach

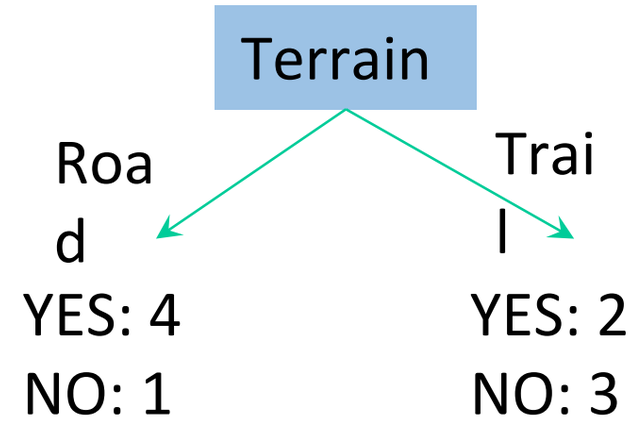
Base case: If all data belong to the same class, create a leaf node with that label

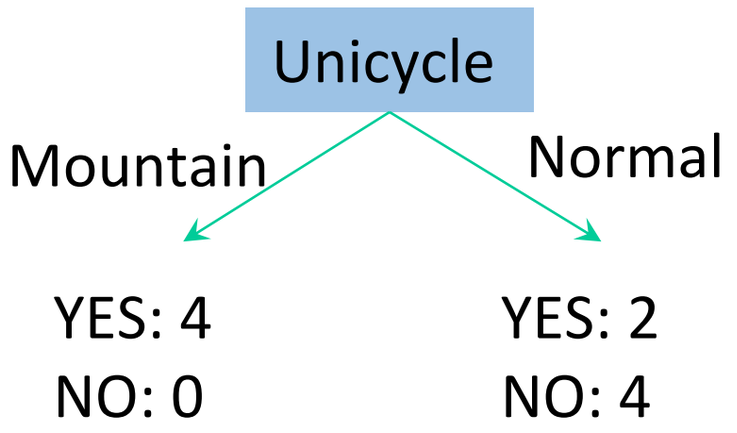
Otherwise:

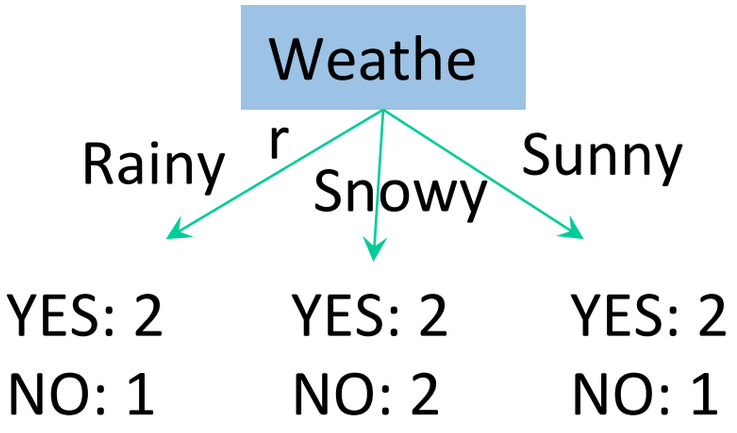
- calculate the "score" for each feature if we used it to split the data
- pick the feature with the highest score, partition the data based on that data value and call recursively

Partitioning the data

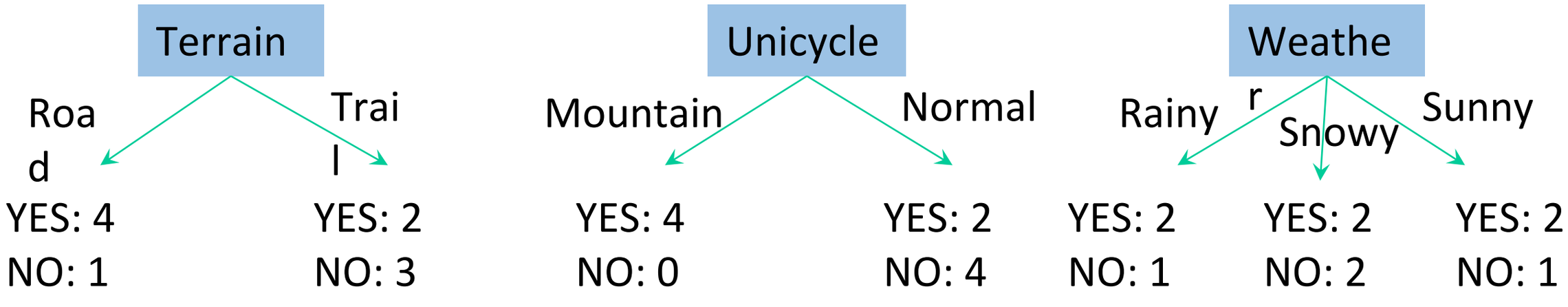
Terrain	Unicycle-type	Weather	Go-For-Ride?
Trail	Normal	Rainy	NO
Road	Normal	Sunny	YES
Trail	Mountain	Sunny	YES
Road	Mountain	Rainy	YES
Trail	Normal	Snowy	NO
Road	Normal	Rainy	YES
Road	Mountain	Snowy	YES
Trail	Normal	Sunny	NO
Road	Normal	Snowy	NO
Trail	Mountain	Snowy	YES







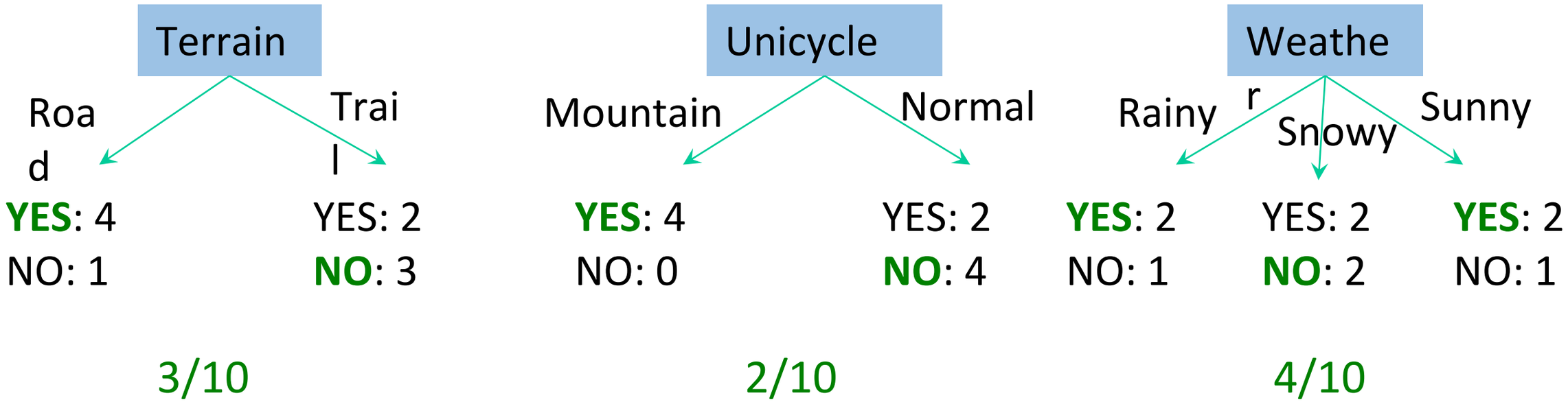
Partitioning the data



calculate the “score” for each feature
if we used it to split the data

What score should we use?
If we just stopped here, which tree would be best?

Decision trees

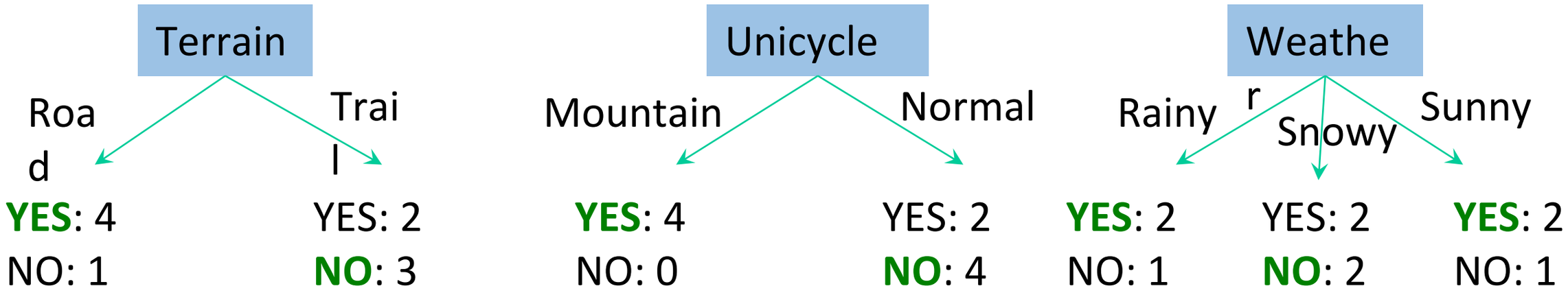


3/10 2/10 4/10

Training error: the average error over the training set

For classification, the most common "error" is the number of mistakes

Training error vs. accuracy



Training error:	3/10	2/10	4/10
Training accuracy:	7/10	8/10	6/10

training error = 1 - accuracy (and vice versa)

Training error: the average error over the training set

Training accuracy: the average percent correct over the training set

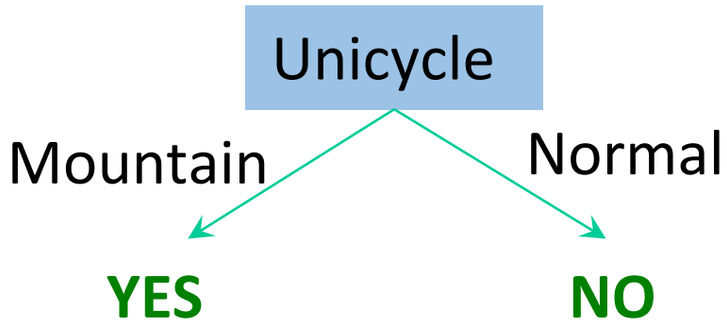
Problematic data

Terrain	Unicycle-type	Weather	Go-For-Ride?
Trail	Normal	Rainy	NO
Road	Normal	Sunny	YES
Trail	Mountain	Sunny	YES
Road	Mountain	Snowy	NO
Trail	Normal	Snowy	NO
Road	Normal	Rainy	YES
Road	Mountain	Snowy	YES
Trail	Normal	Sunny	NO
Road	Normal	Snowy	NO
Trail	Mountain	Snowy	YES

When can this happen?

Overfitting

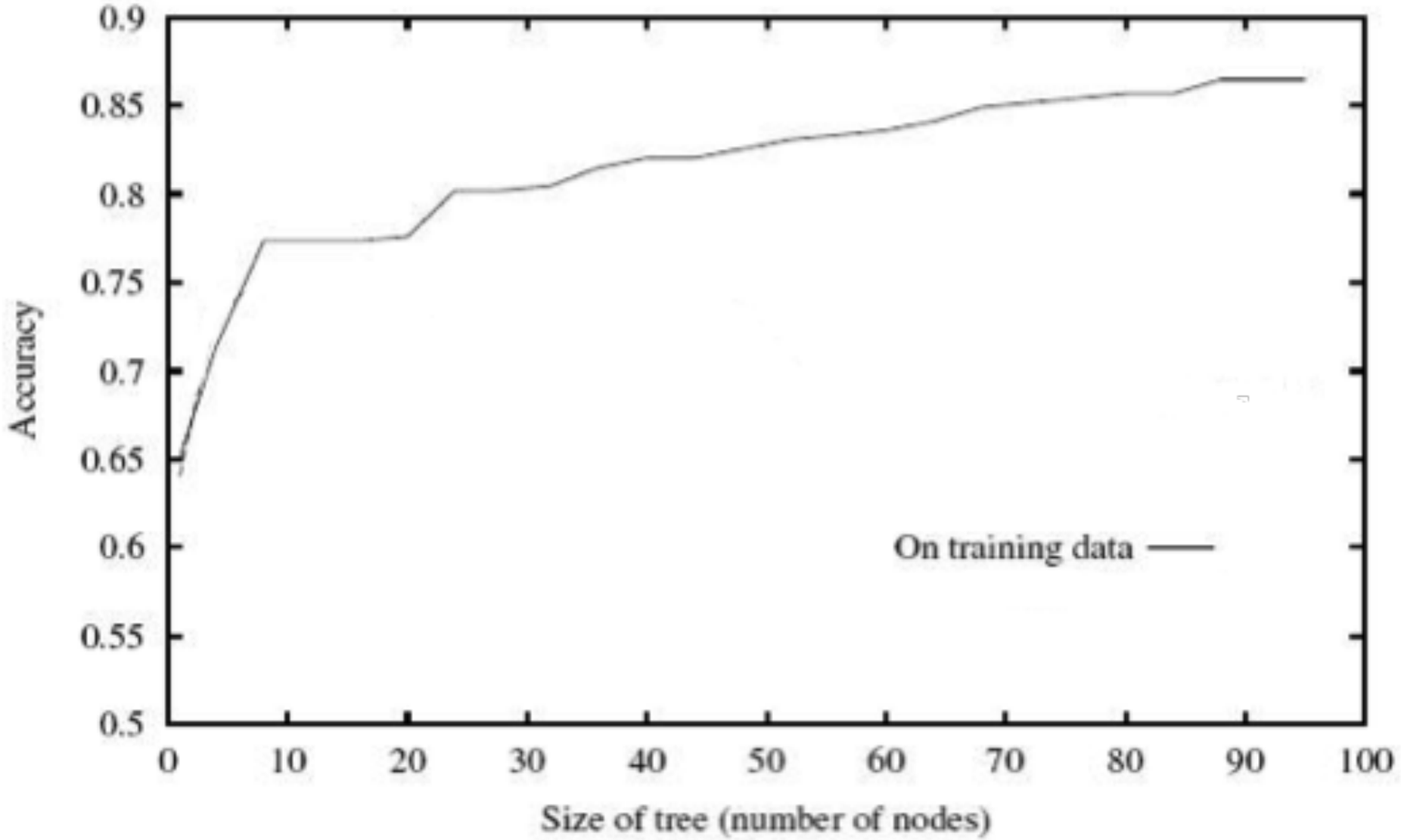
Terrain	Unicycle-type	Weather	Go-For-Ride?
Trail	Mountain	Rainy	YES
Trail	Mountain	Sunny	YES
Road	Mountain	Snowy	YES
Road	Mountain	Sunny	YES
Trail	Normal	Snowy	NO
Trail	Normal	Rainy	NO
Road	Normal	Snowy	YES
Road	Normal	Sunny	NO
Trail	Normal	Sunny	NO



Overfitting occurs when we bias our model too much towards the training data

Our goal is to learn a **general** model that will work on the training data as well as test data

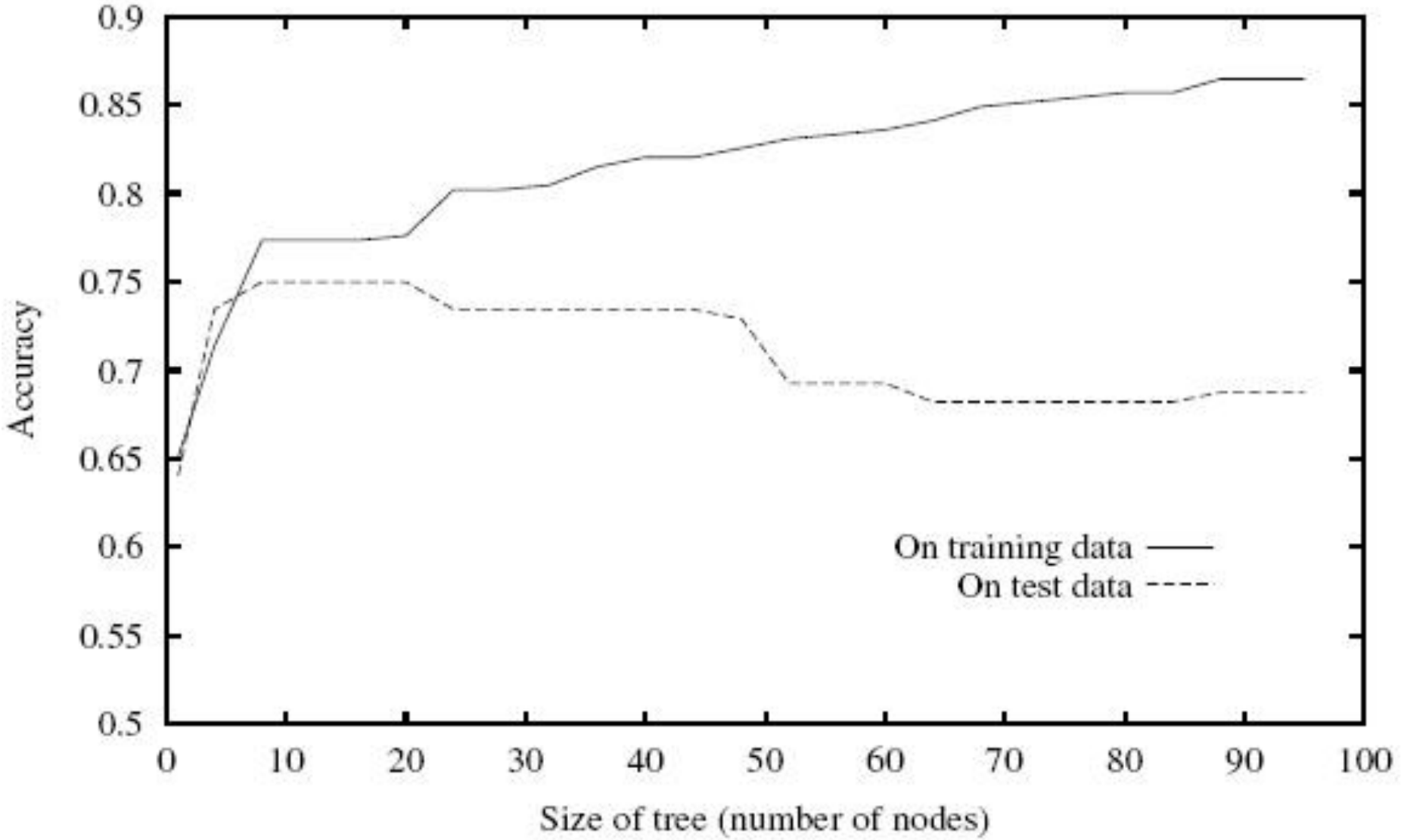
Overfitting



Our decision tree learning procedure always decreases training error

Is that what we want?

Overfitting



Even though the training error is decreasing,

the testing error can go up !

Preventing overfitting

- We've reached a particular depth in the tree
- We only have a certain number/fraction of examples remaining
- We've reached a particular training error
- Use development data

Decision trees: the good

- Very intuitive and easy to interpret
- Fast to run and fairly easy to implement
- Historically, perform fairly well
- No prior assumptions about the data

Decision trees: the bad

- Can be slow with large numbers of features
- Can’t learn some very simple data sets
    - e.g. some types of linearly separable data
- Pruning/tuning can be tricky to get right
- Very deep trees = overfiting to the training set

Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance

Bias, Variance, Measures

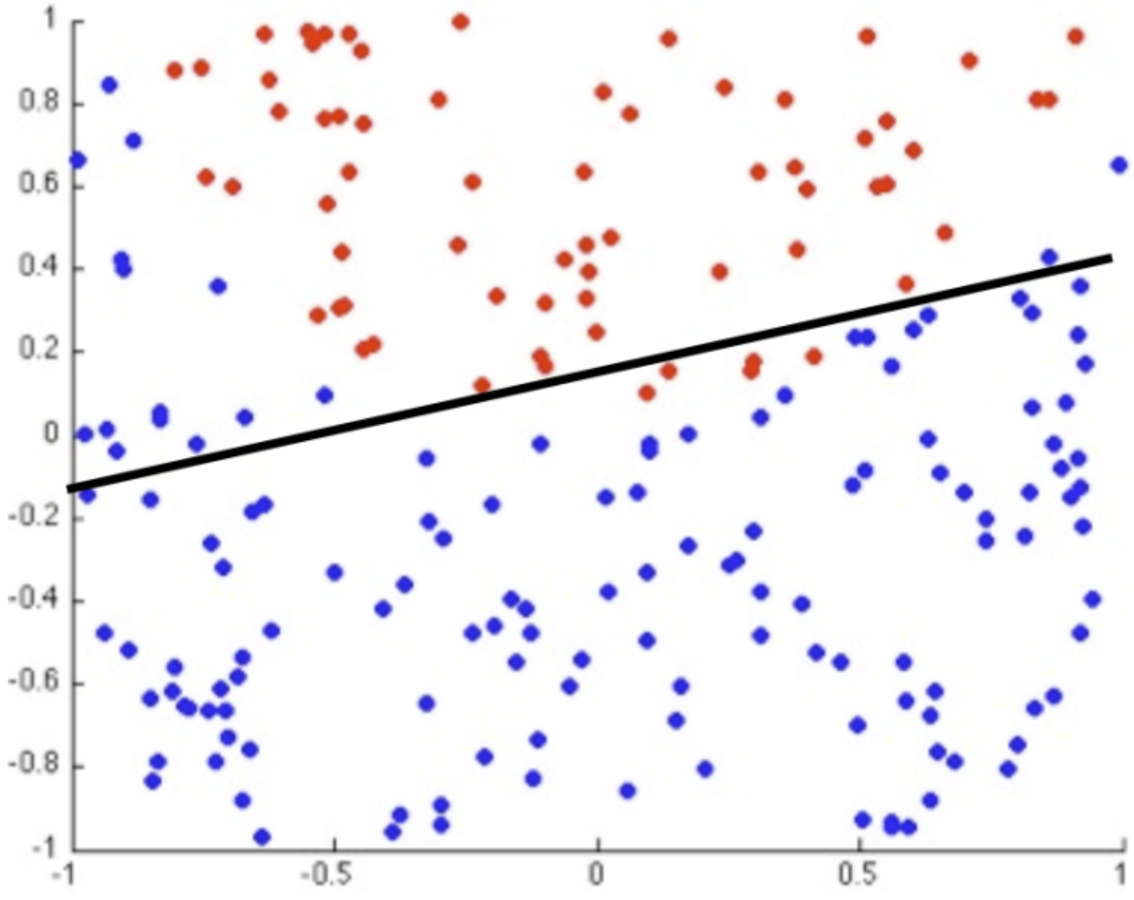
Bias and Variance

Classification error = Bias + Variance

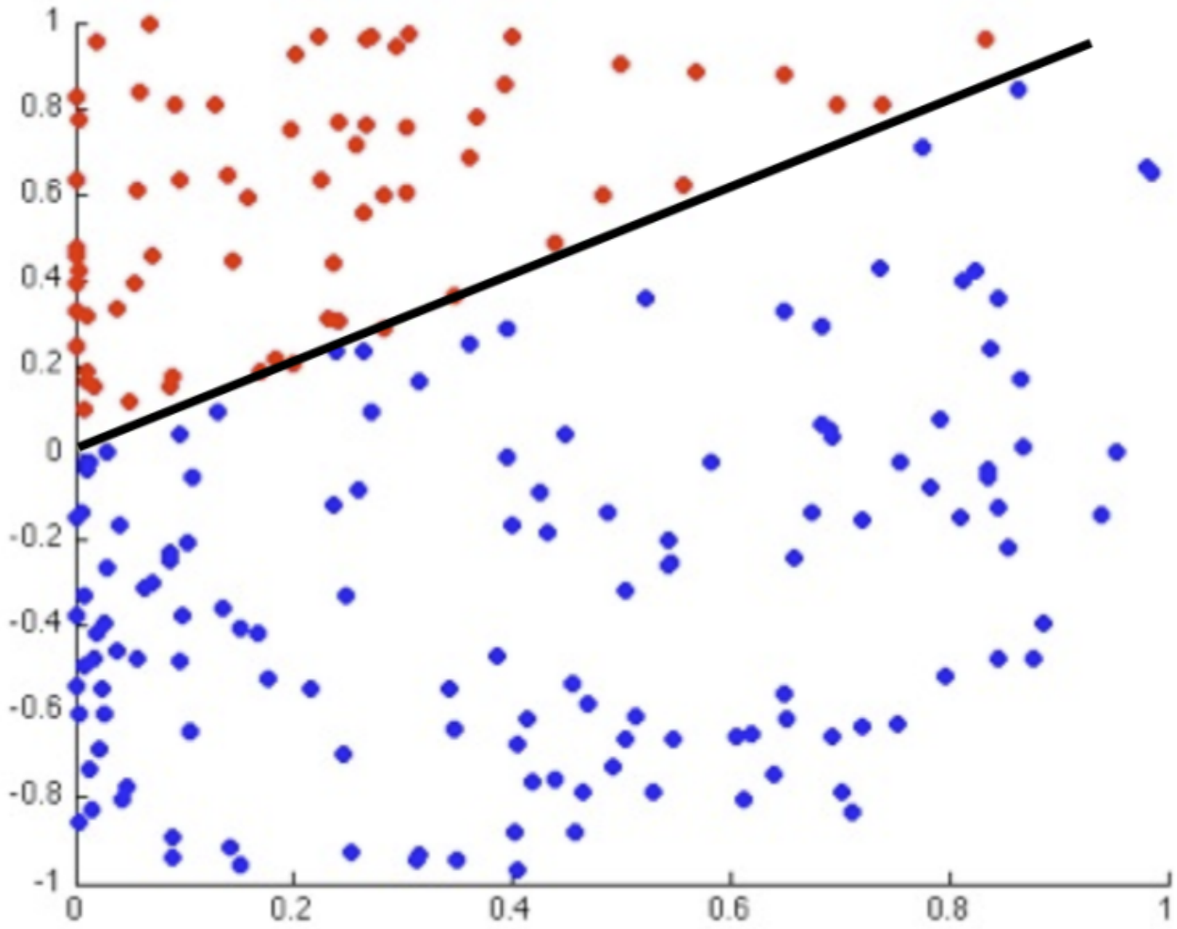
Bias is the true error of the best classifier in the concept class (e.g best linear separator)

Bias is high if the concept class cannot model the true data distribution well, and does not depend on training set size.

Bias and Variance



High Bias



Low Bias

Underfitting = high bias

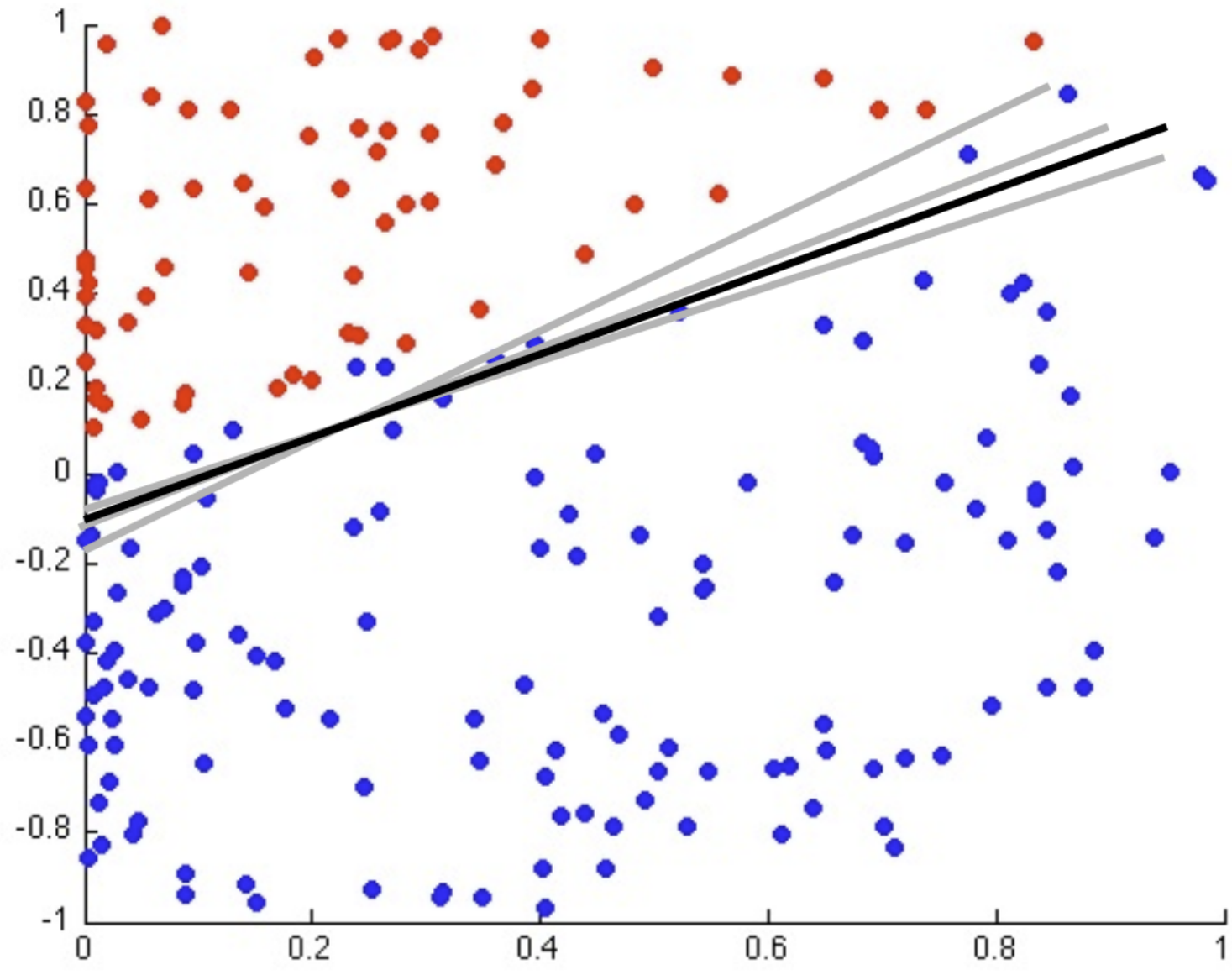
Bias and Variance

Classification error = Bias + Variance

Variance is the error of the trained classifier with respect to the best classifier in the concept class.

Variance depends on the training set size. It decreases with more training data, and increases with more complicated classifiers.

Bias and Variance



Overfitting: when you have extra variance

Bias and Variance

Classification error = Bias + Variance

If you have high bias

both training and test error will be high

If you have high variance,

training error will be low, and test error will be high

Performance Measures

Different outcomes for classifications

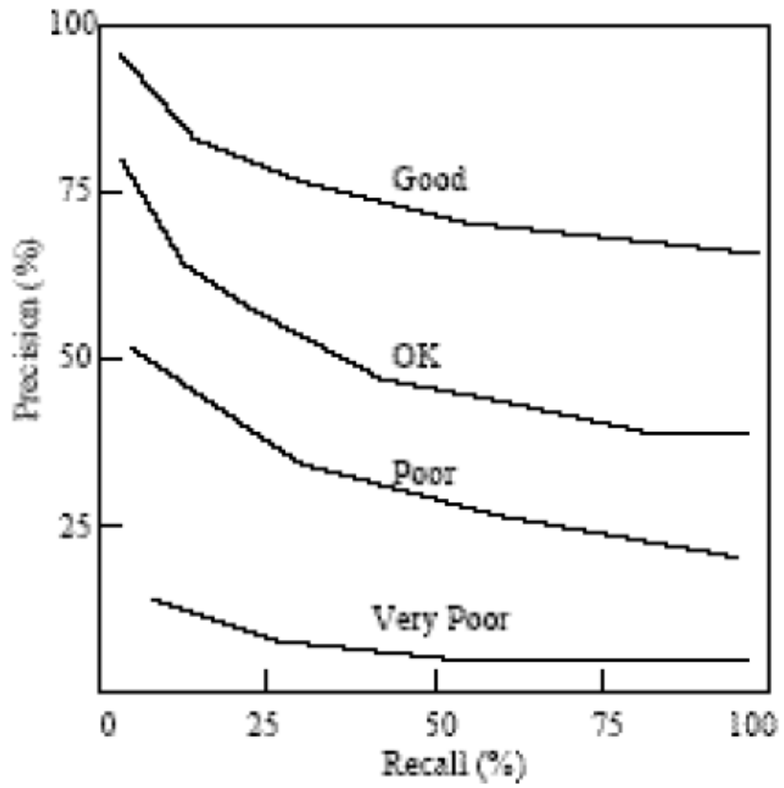
	Classified Positive	Classified Negative
Positive Examples	True Positive (TP)	False Negative (FN)
Negative Examples	False Positive (FP)	True Negative (TN)

Performance Measures

- **Accuracy**
  - Proportion of classifications that were correct
  - (TP + TN) / # of Training Examples
- **Recall**
  - Proportion of positive examples that were correctly classified
  - TP / (TP + FN)
- **Precision**
  - Proportion of correct positive classifications over all positive classifications
  - TP/(TP + FP)

Precision vs. Recall

- There is a trade-off between precision and recall
  - Precision: How often does system correctly classify spam emails?
  - Recall: How often are emails classified as spam actually spam?
- Precision/ Recall Curves



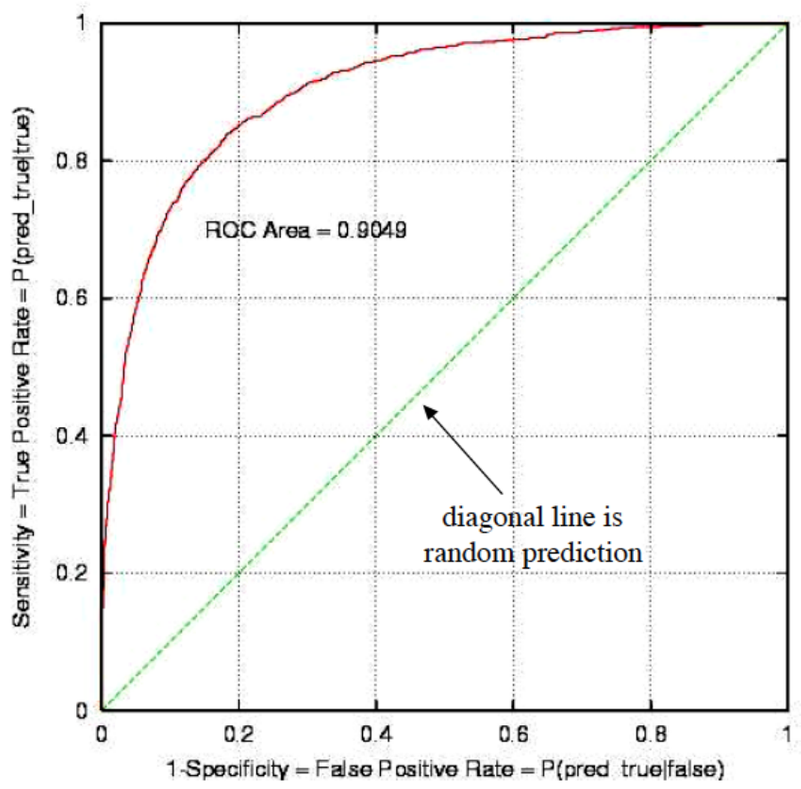
F1-Score

- “Harmonic average of precision and recall”
- F1 = 2 * (Precision * Recall) / (Precision + Recall)

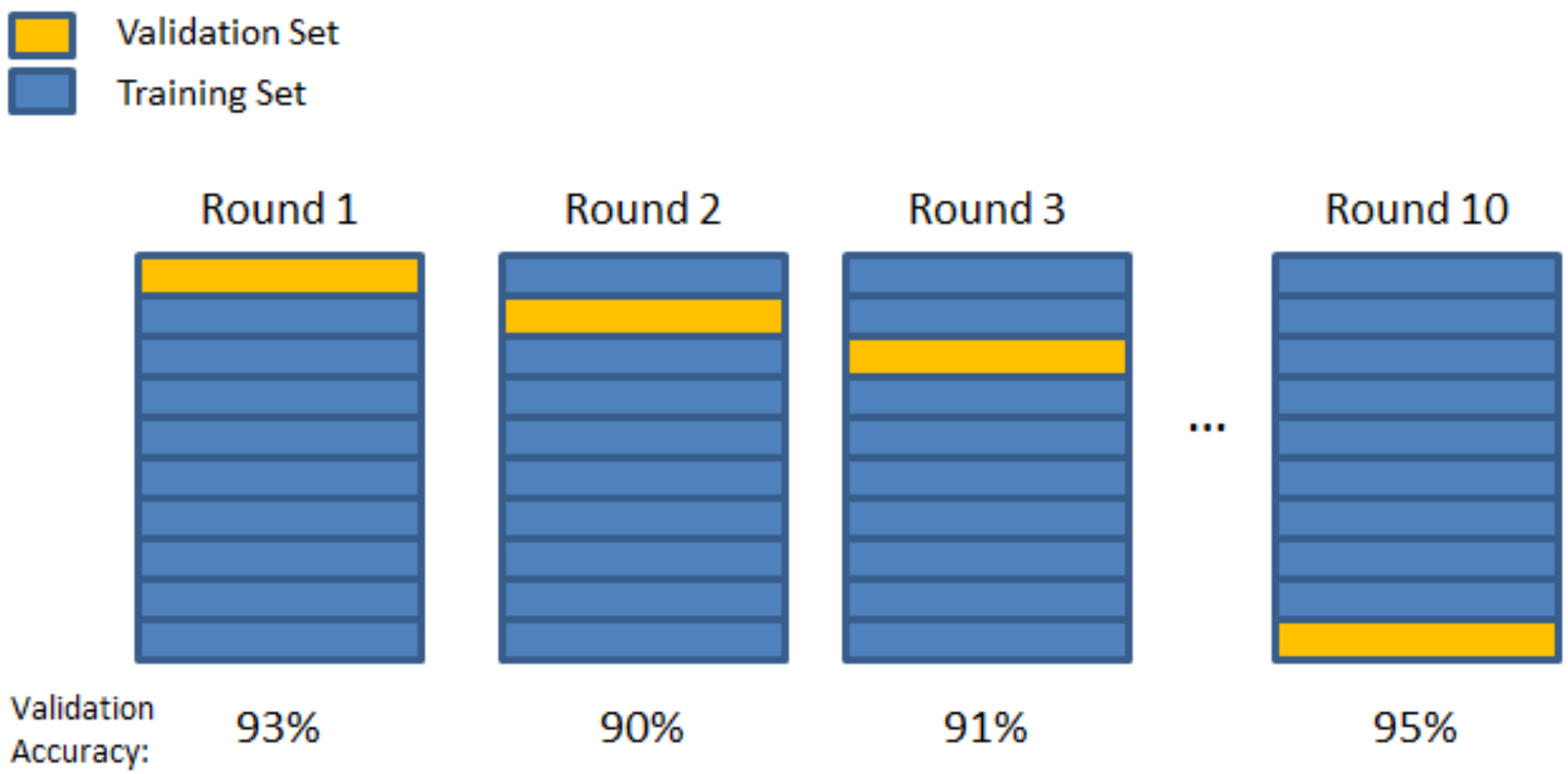
Break-even point is the point where precision = recall

ROC Plots

- Receiver Operator Characteristic
  - Sweep your threshold
  - Plot true positive rate vs. false positive rate
    - True Positive Rate = Recall = TP / (TP + FN)
    - False Positive Rate = FP / (FP + TN)



Cross Validation



Final Accuracy = Average(Round 1, Round 2, ...)

Thank you for your attention !

Questions ?