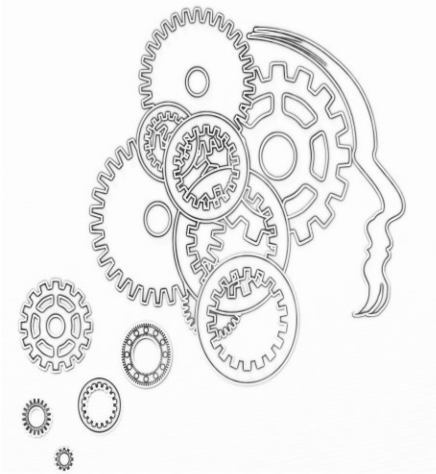
Artificial Intelligence

CSC-462

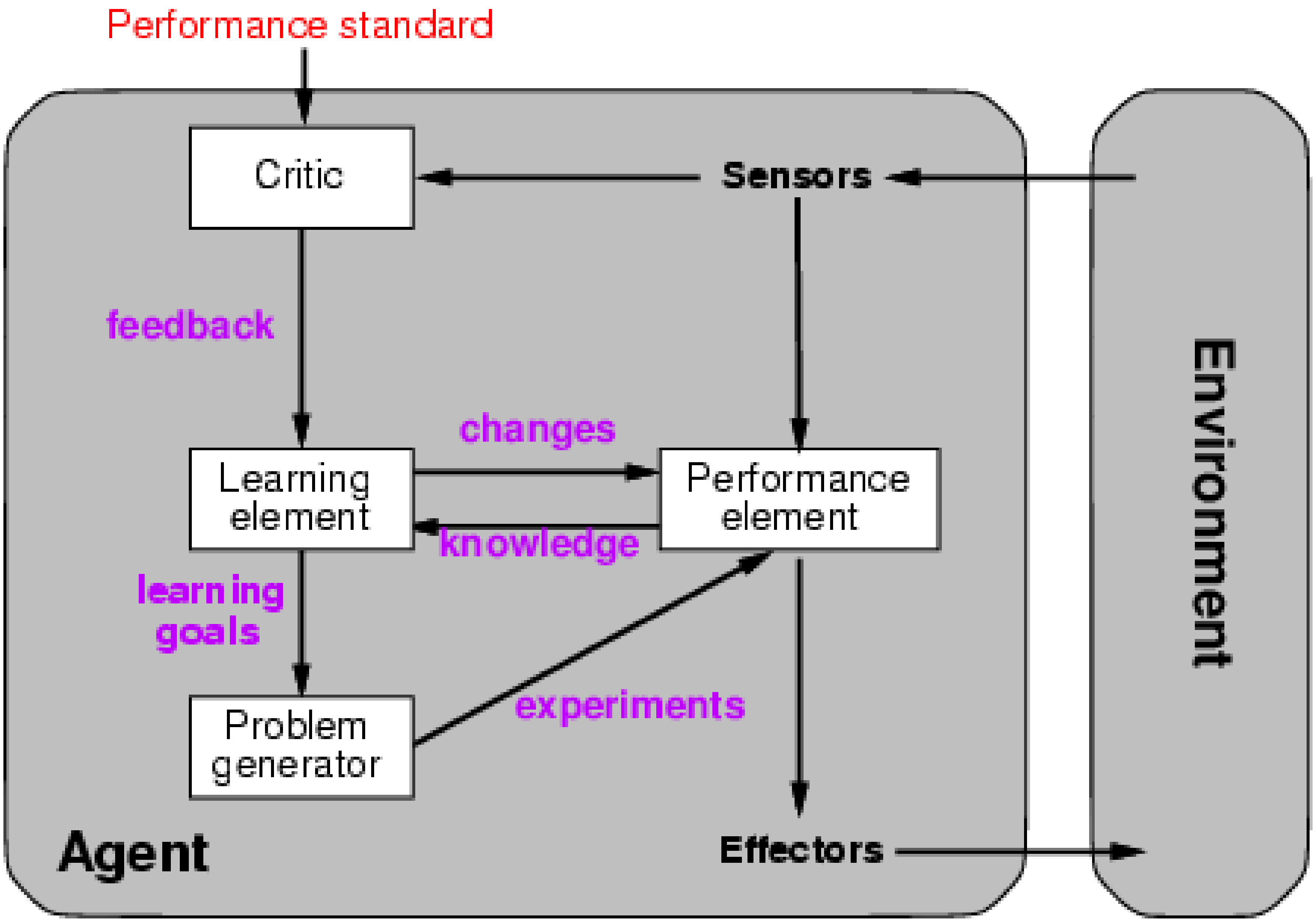
Muhammad Najam Dar

Today’s Learning



Introduction to Machine Learning
Machine Learning Basics

Learning Agents



Learning element

Learning modifies the agent's decision mechanisms to improve performance.

Type of feedback:

- Supervised learning: correct answers for each example
- Unsupervised learning: correct answers not given
- Reinforcement learning: occasional rewards

What is Machine Learning?

- Machine Learning
  - Study of algorithms that
  - improve their performance
  - at some task
  - with experience
- Optimize a performance criterion using example data or past experience.
- Role of Statistics: Inference from a sample

What is Machine Learning?

- "Any change in a System that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population."
- Adapt to / learn from data
  - To optimize a performance function

Can be used to:

- Extract knowledge from data
- Create software that improves over time

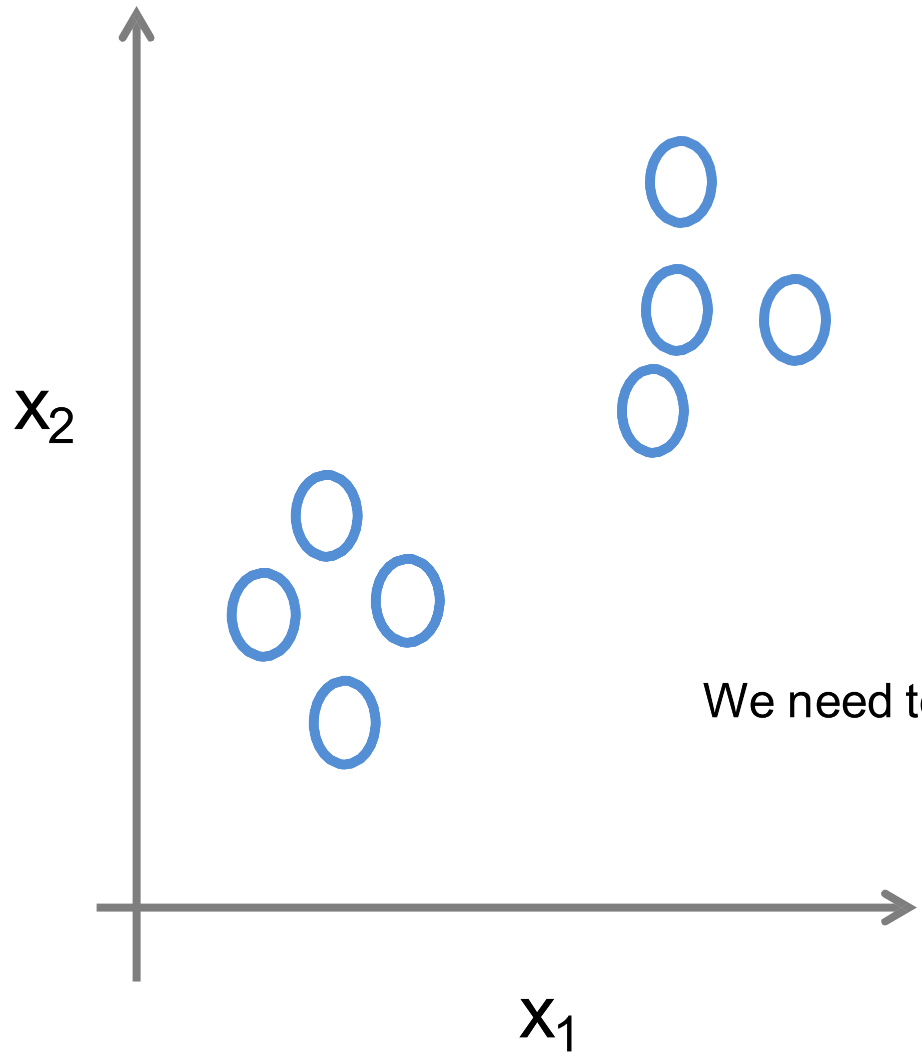
Applications of Machine Learning

- Machine learning is preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - Computational biology
- This trend is accelerating
  - Improved machine learning algorithms
  - Improved data capture, networking, faster computers
  - New sensors / IO devices
  - Demand for self-customization to user, environment

Categories

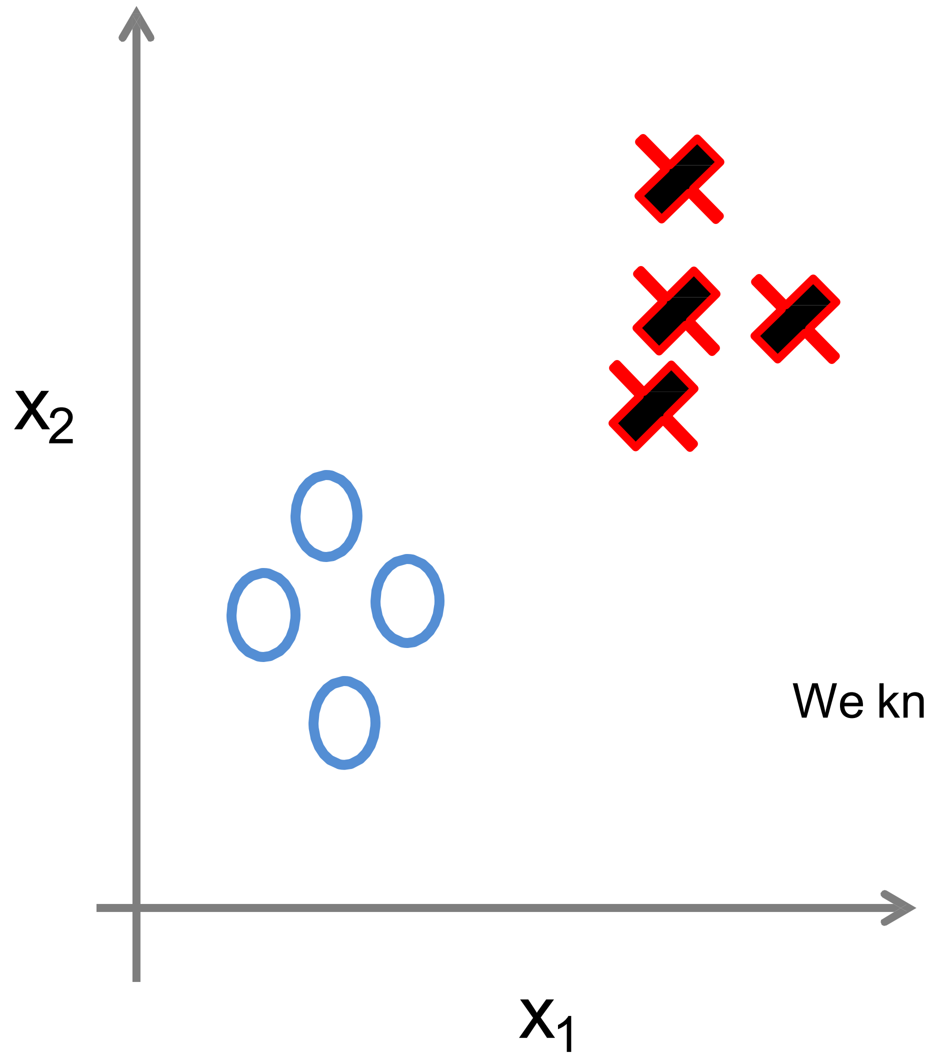
- Unsupervised Learning
- Supervised Learning
  - Classification
  - Regression/Prediction
- Reinforcement Learning

Unsupervised Learning



We need to figure out the patterns

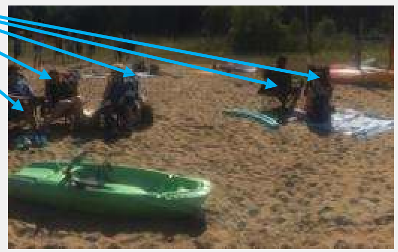
Supervised Learning



We knew the correct answers

Classification: Answers are Categories

Person



Model

Trained Model

Fit

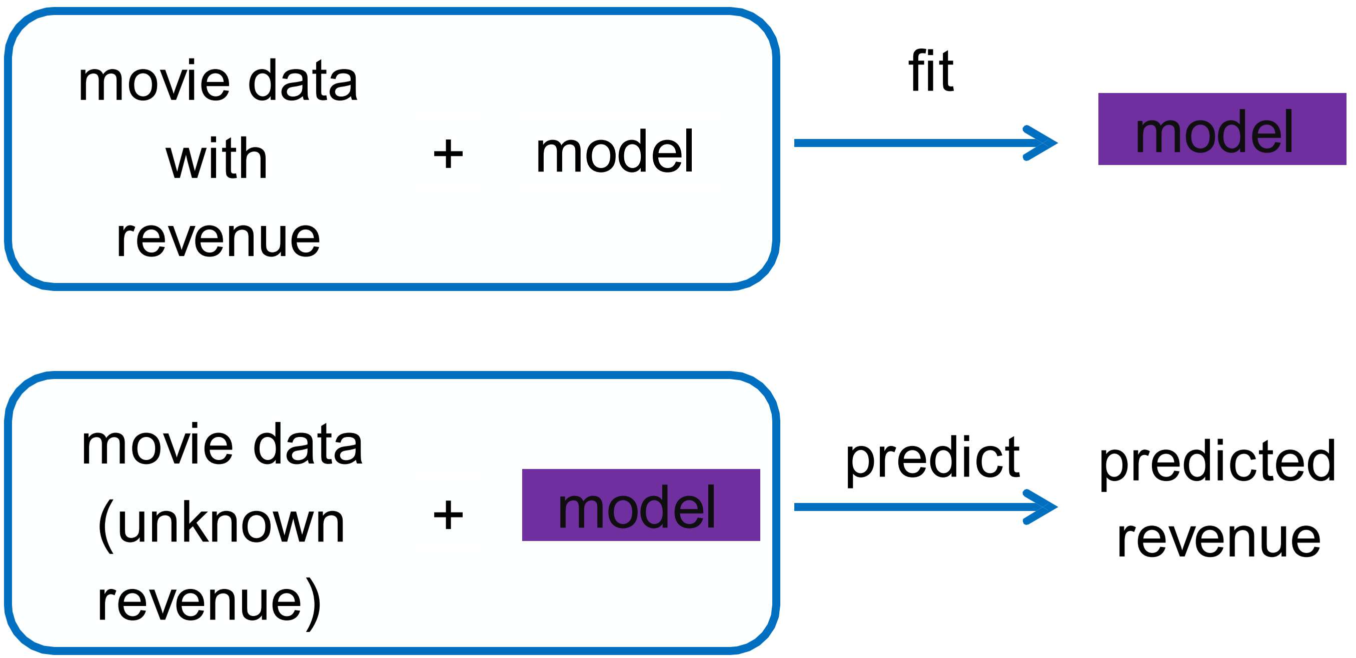
?

Trained Model

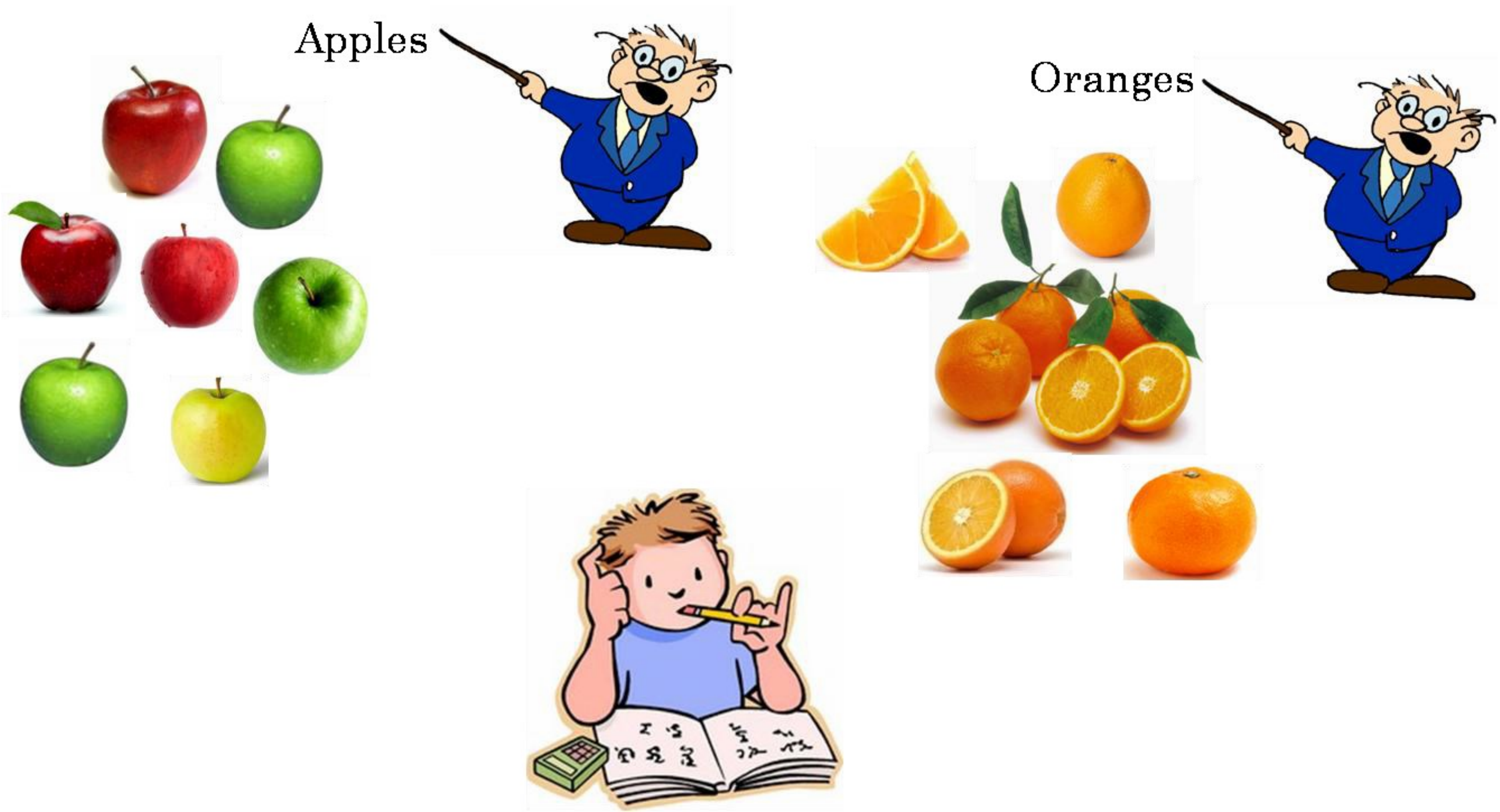
Predict

Predict if there is a person in this photo (yes | no)

Regression: Numeric Answers



Classification



Classification

Given: training images and their categories

What are the categories of these test images?

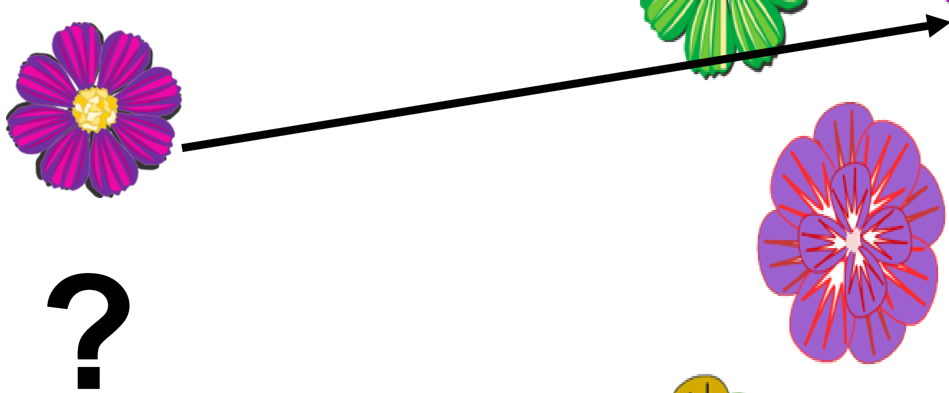
What is Classification?

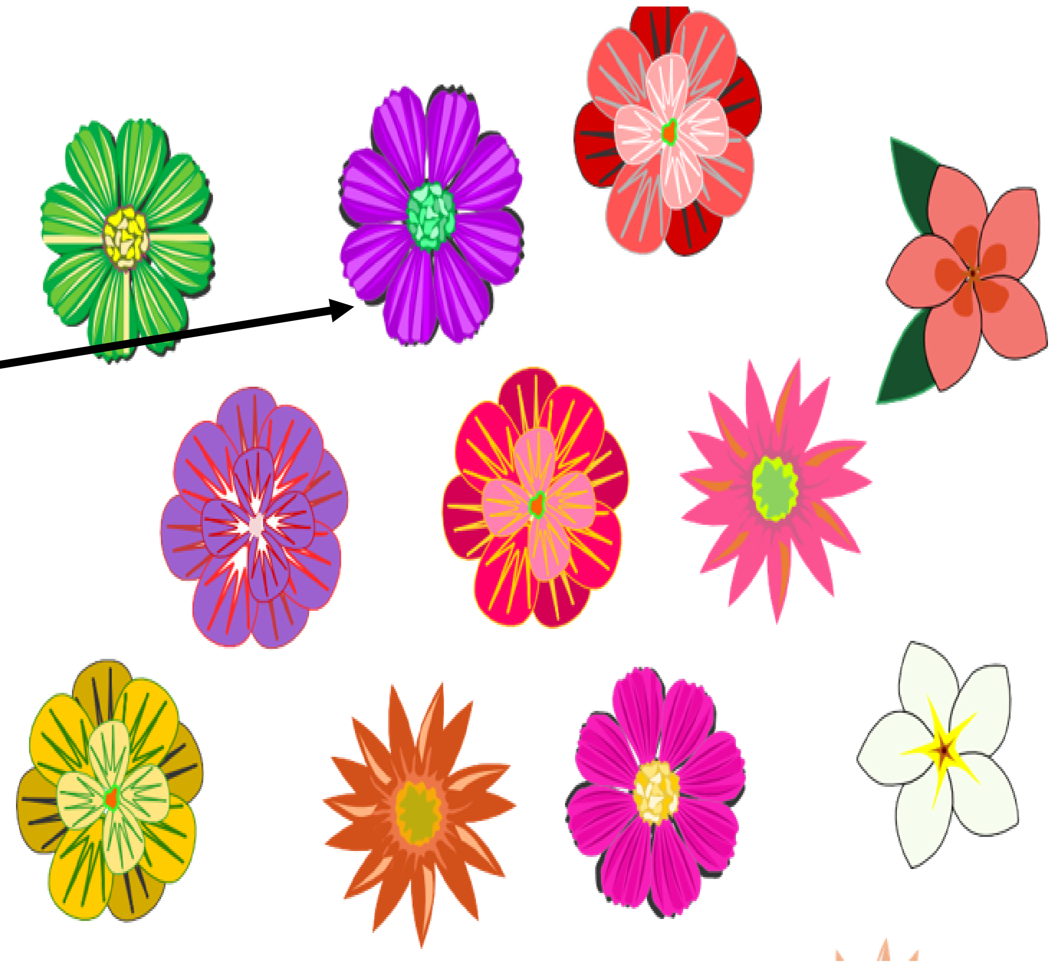
Which flower is a customer most likely to purchase based on similarity to previous purchase?

?

What is Classification?

Which flower is a customer most likely to purchase based on similarity to previous purchase?



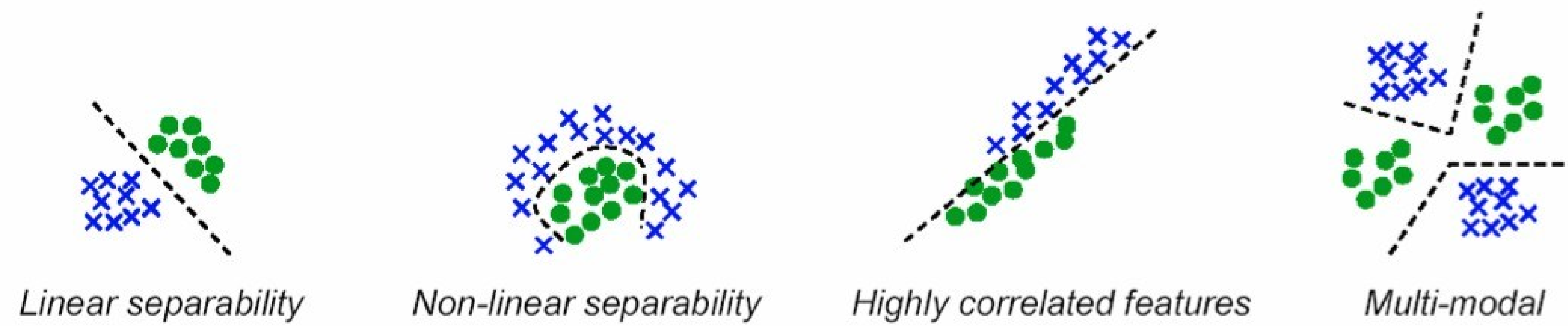


What is Needed for Classification?

- Model data with:
  - Features that can be quantitated
  - Labels that are known
- Method to measure similarity

Features

"Good" features
"Bad" features



Linear separability
Non-linear separability
Highly correlated features
Multi-modal

Machine Learning Vocabulary

- **Target:** predicted category or value of the data (column to predict)
- **Features**: properties of the data used for prediction (non-target columns)
- **Example:** a single data point within the data (one row)
- **Label:** the target value for a single data point

Machine Learning Vocabulary

sepal length	sepal width	petal length	petal width	species
6.7	3.0	5.2	2.3	virginica
6.4	2.8	5.6	2.1	virginica
4.6	3.4	1.4	0.3	setosa
6.9	3.1	4.9	1.5	versicolor
4.4	2.9	1.4	0.2	setosa
4.8	3.0	1.4	0.1	setosa
5.9	3.0	5.1	1.8	virginica
5.4	3.9	1.3	0.4	setosa
4.9	3.0	1.4	0.2	setosa
5.4	3.4	1.7	0.2	setosa

Target

Machine Learning Vocabulary

Features

sepal length	sepal width	petal length	petal width	species
6.7	3.0	5.2	2.3	virginica
6.4	2.8	5.6	2.1	virginica
4.6	3.4	1.4	0.3	setosa
6.9	3.1	4.9	1.5	versicolor
4.4	2.9	1.4	0.2	setosa
4.8	3.0	1.4	0.1	setosa
5.9	3.0	5.1	1.8	virginica
5.4	3.9	1.3	0.4	setosa
4.9	3.0	1.4	0.2	setosa
5.4	3.4	1.7	0.2	setosa

Machine Learning Vocabulary

Example

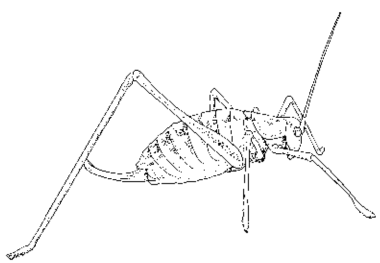
sepal length	sepal width	petal length	petal width	species
6.7	3.0	5.2	2.3	virginica
6.4	2.8	5.6	2.1	virginica
4.6	3.4	1.4	0.3	setosa
6.9	3.1	4.9	1.5	versicolor
4.4	2.9	1.4	0.2	setosa
4.8	3.0	1.4	0.1	setosa
5.9	3.0	5.1	1.8	virginica
5.4	3.9	1.3	0.4	setosa
4.9	3.0	1.4	0.2	setosa
5.4	3.4	1.7	0.2	setosa

Machine Learning Vocabulary

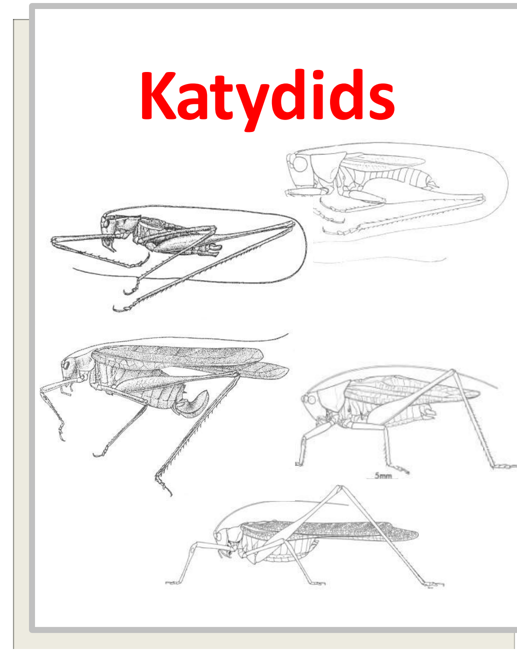
sepal length	sepal width	petal length	petal width	species
6.7	3.0	5.2	2.3	virginica
6.4	2.8	5.6	2.1	virginica
4.6	3.4	1.4	0.3	setosa
6.9	3.1	4.9	1.5	versicolor
4.4	2.9	1.4	0.2	setosa
4.8	3.0	1.4	0.1	setosa
5.9	3.0	5.1	1.8	virginica
5.4	3.9	1.3	0.4	setosa
4.9	3.0	1.4	0.2	setosa
5.4	3.4	1.7	0.2	setosa

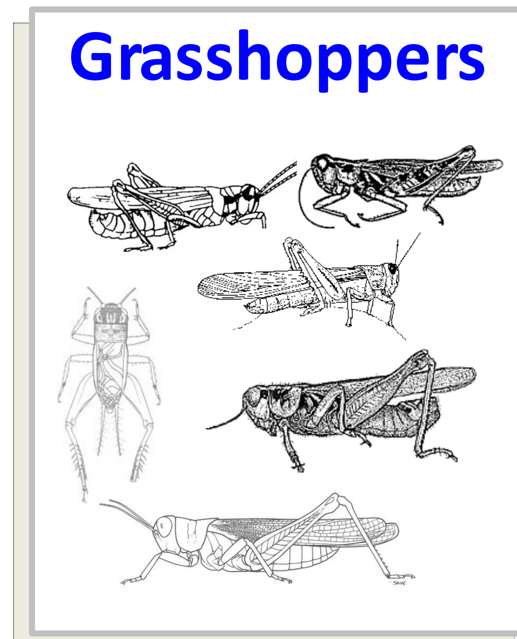
Label

Given a collection of annotated data. In this case 5 instances of **Katydids** and five of **Grasshoppers**, decide what type of insect the unlabeled example is.

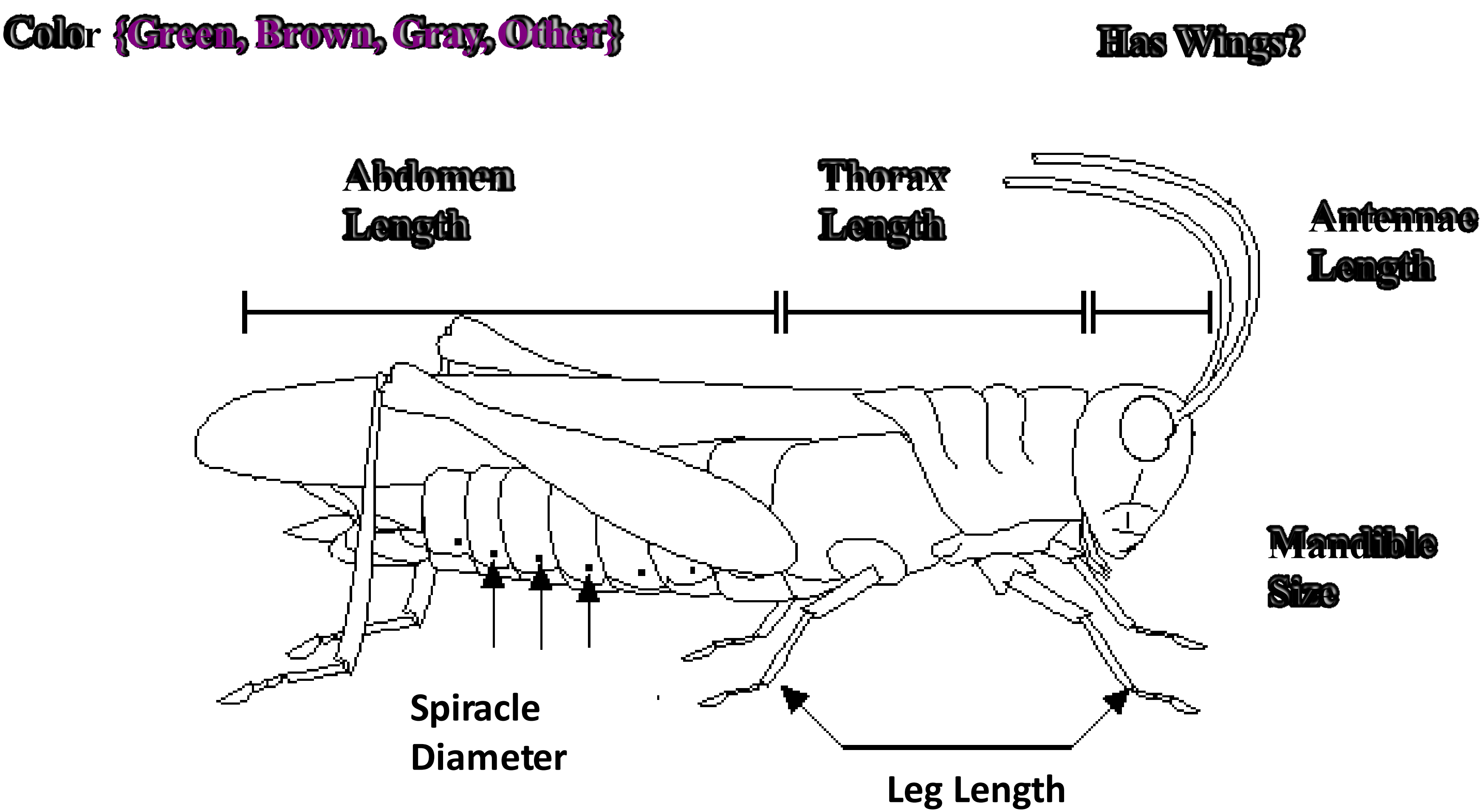


Katydid or **Grasshopper**?





For any domain of interest, we can measure *features*



We can store features in a database.

The classification problem can now be expressed as:

- Given a training database (My_Collection), predict the class label of a previously unseen instance

My_Collection

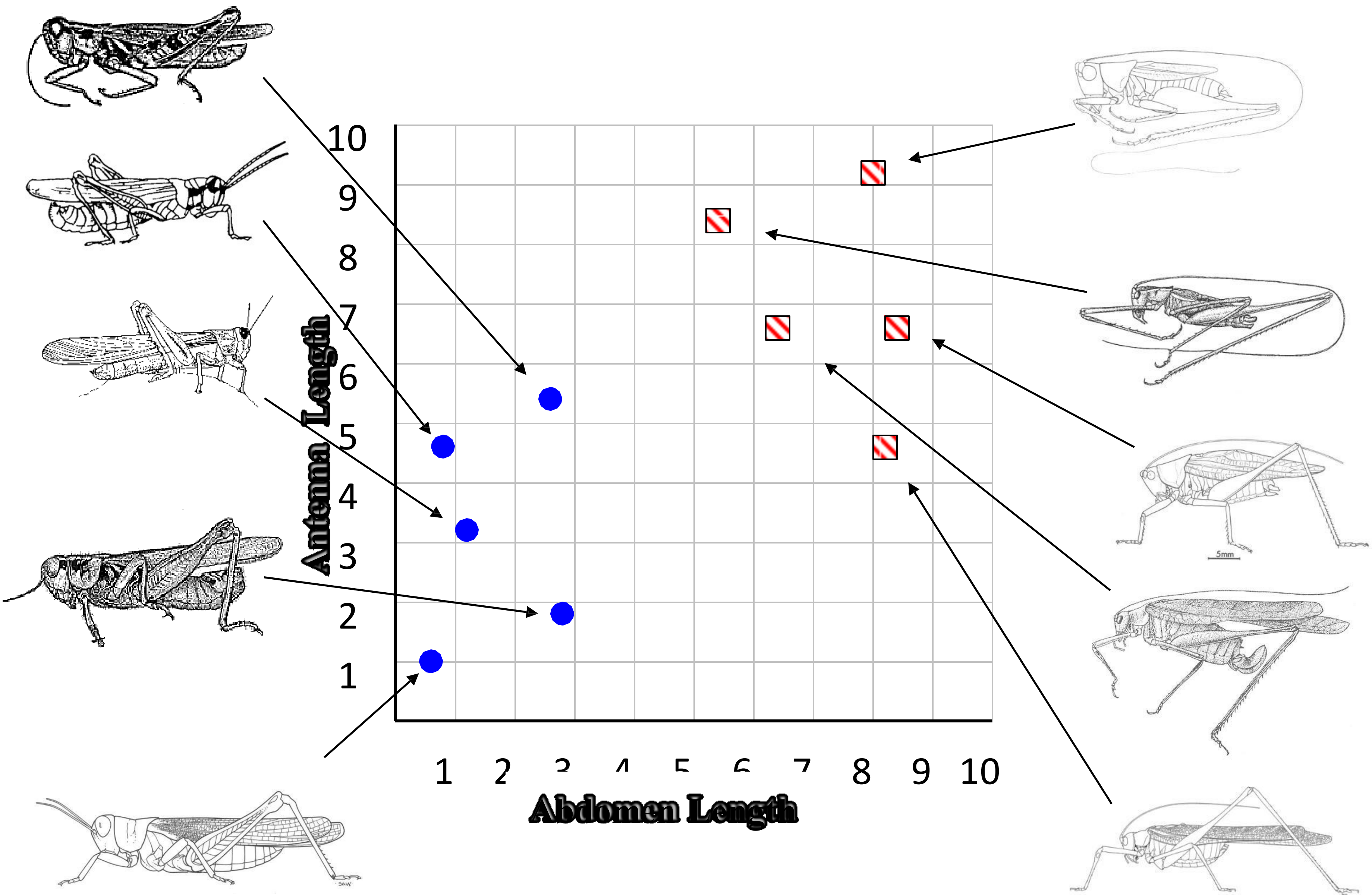
Insect ID	Abdomen Length	Antennae Length	Insect Class
1	2.7	5.5	Grasshopper
2	8.0	9.1	Katydid
3	0.9	4.7	Grasshopper
4	1.1	3.1	Grasshopper
5	5.4	8.5	Katydid
6	2.9	1.9	Grasshopper
7	6.1	6.6	Katydid
8	0.5	1.0	Grasshopper
9	8.3	6.6	Katydid
10	8.1	4.7	Katydids

previously unseen instance =

11	5.1	7.0	???????

Grasshoppers

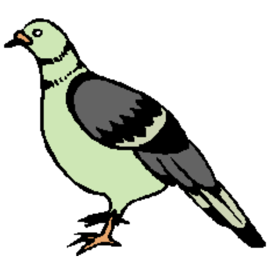
Katydids



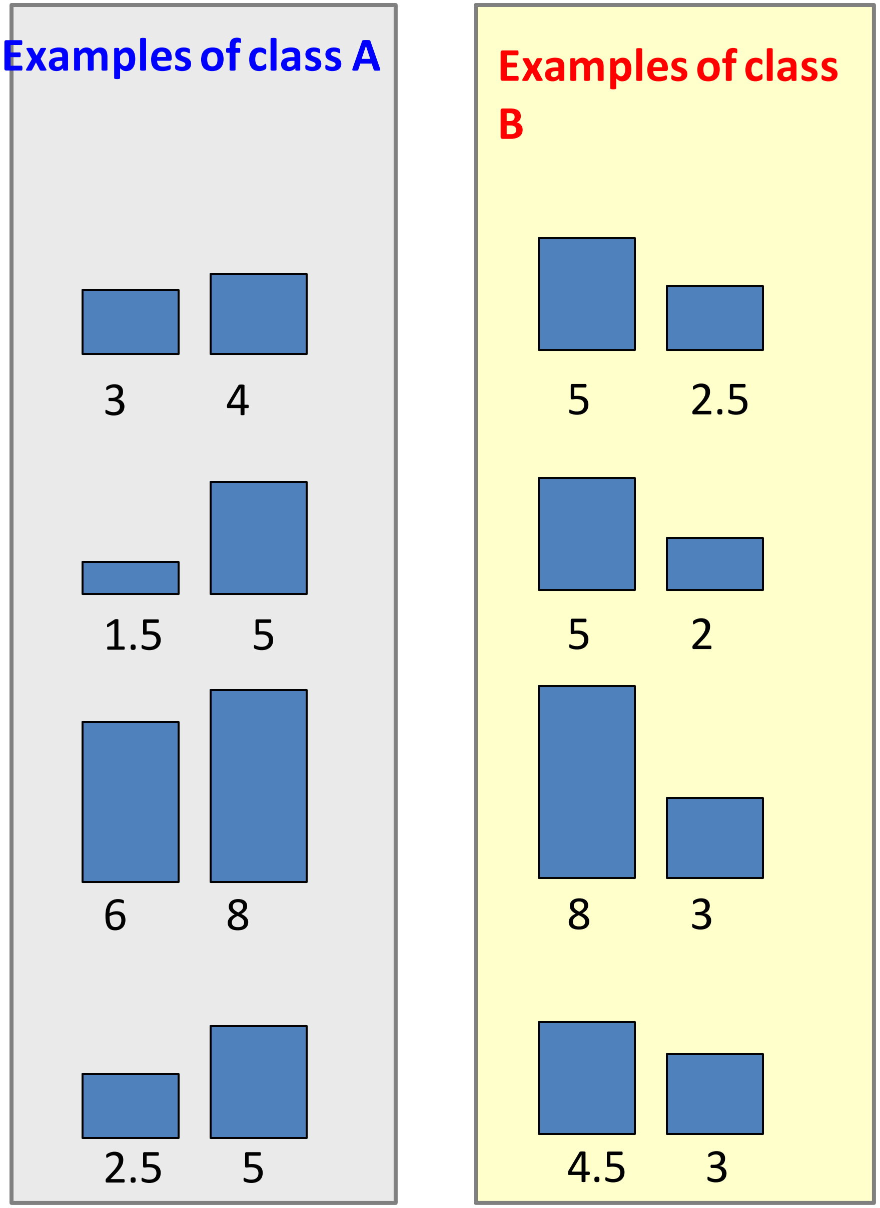
We will return to the previous slide in two minutes. In the meantime, we are going to play a quick game.

I am going to show you some classification problems which were shown to pigeons!

Let us see if you are as smart as a pigeon!



Pigeon Problem 1



Pigeon Problem 1

Examples of class A

3 4

1.5 5

6 8

2.5 5

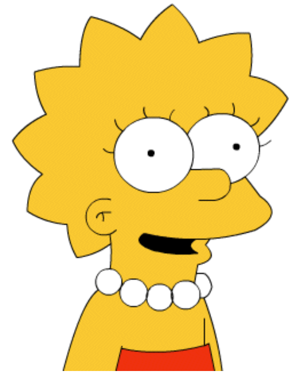
Examples of class B

5 2.5

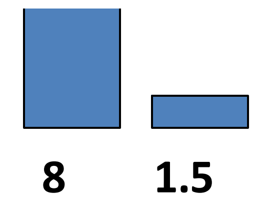
5 2

8 3

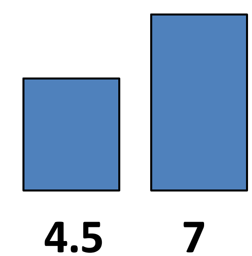
4.5 3



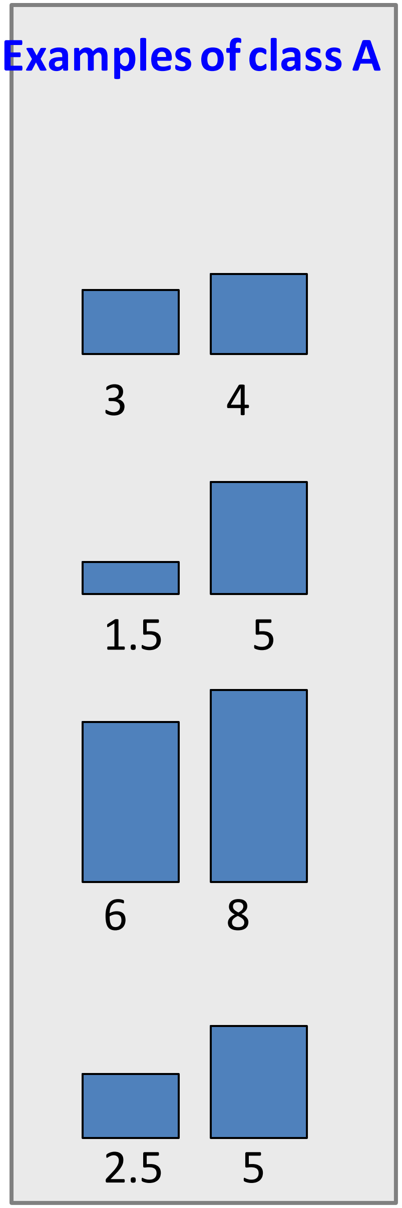
What class is this object?

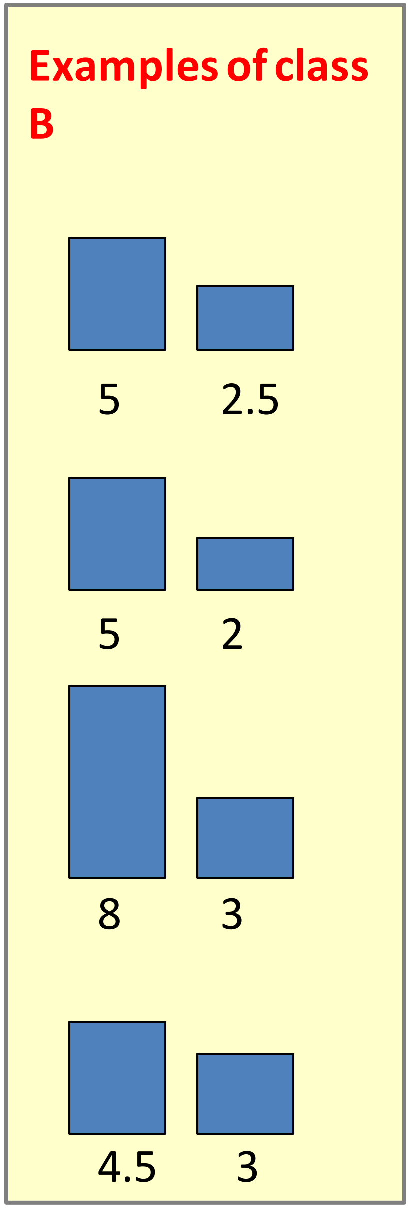


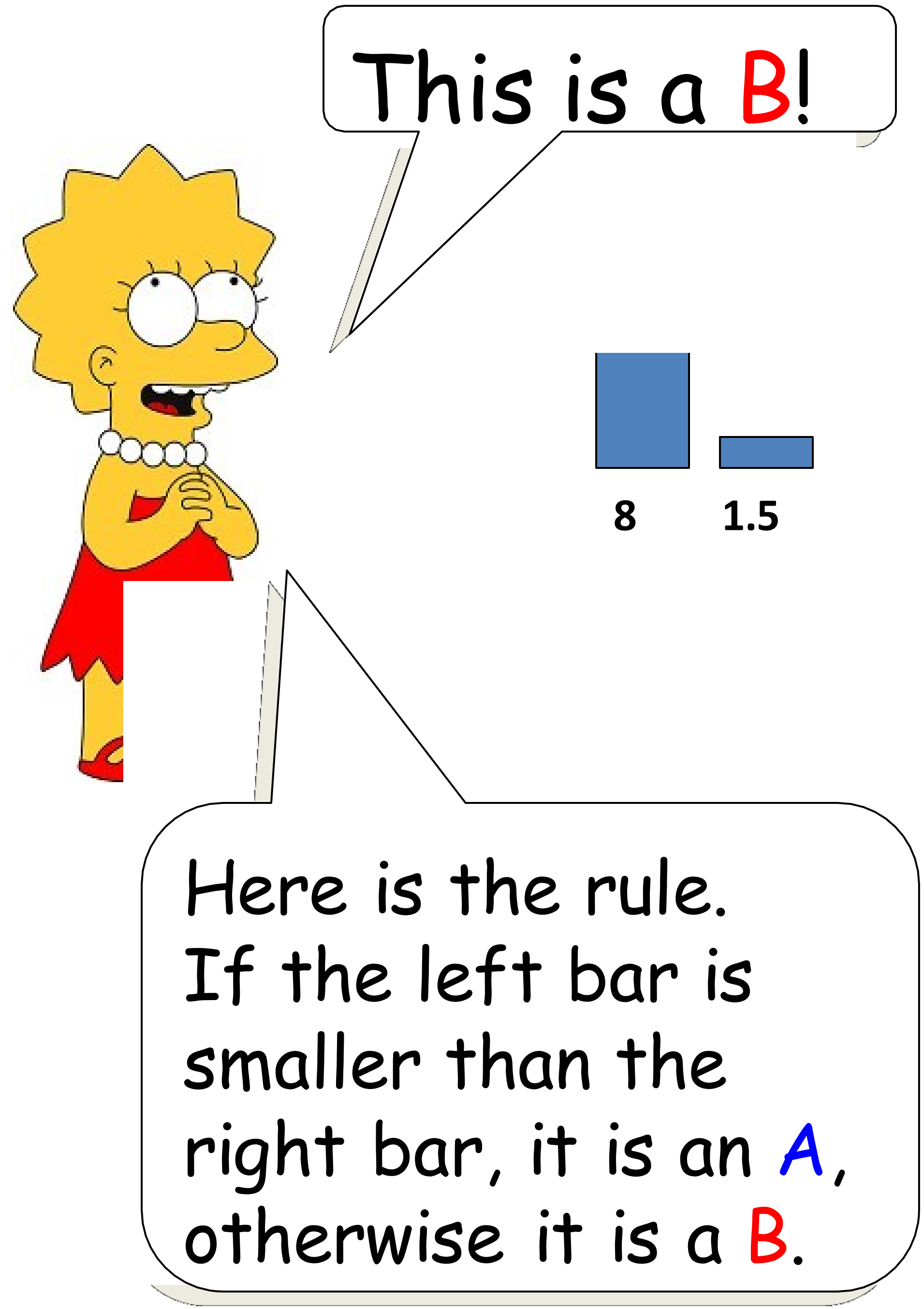
What about this one, A or B?



Pigeon Problem 1

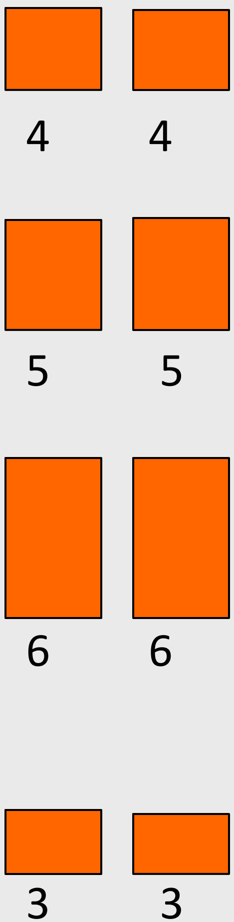




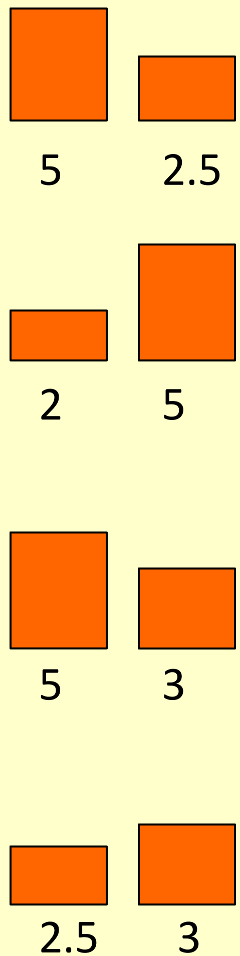


Pigeon Problem 2

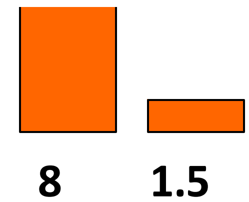
Examples of class A



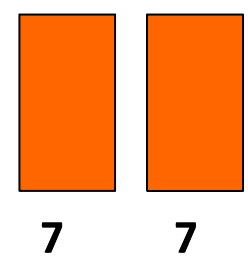
Examples of class B



Oh! This ones hard!

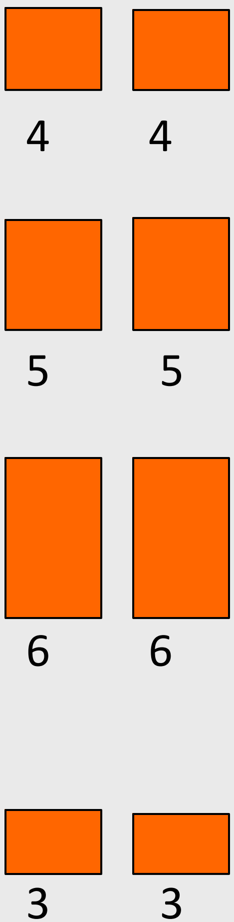


Even I know this one

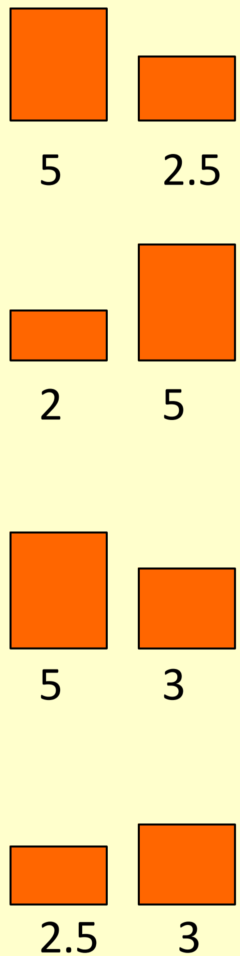


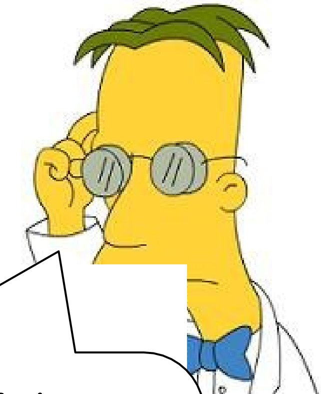
Pigeon Problem 2

Examples of class A



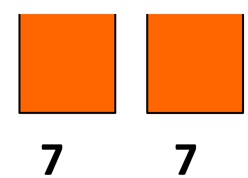
Examples of class B





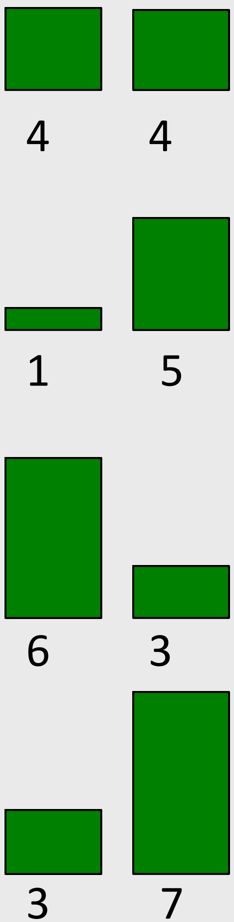
The rule is as follows, if the two bars are equal sizes, it is an A. Otherwise it is a B.

So this one is an A.

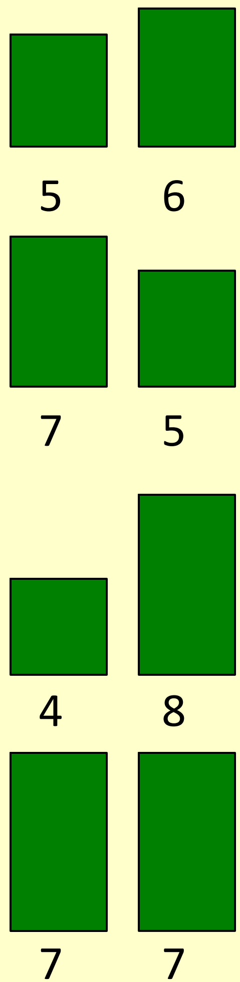


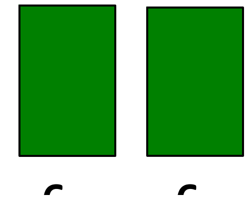
Pigeon Problem 3

Examples of class A



Examples of class B

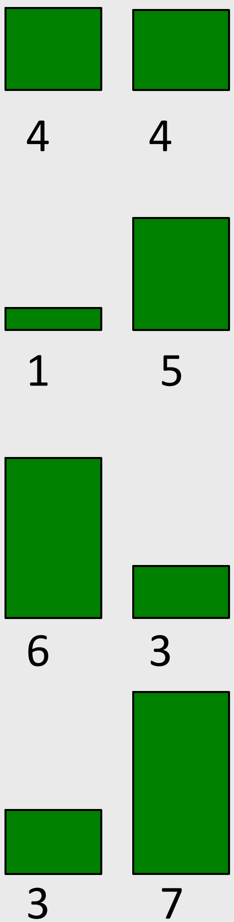




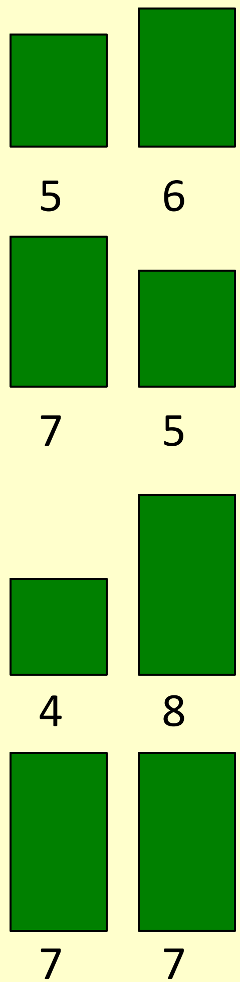
This one is really hard!
What is this, A or B?

Pigeon Problem 3

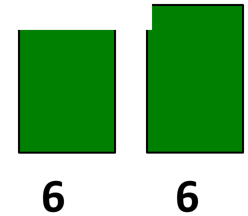
Examples of class A



Examples of class B



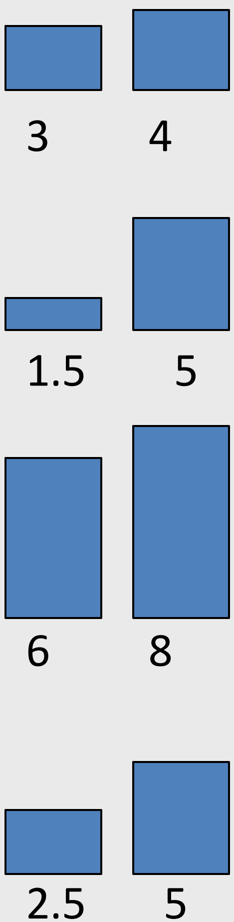
It is a B!



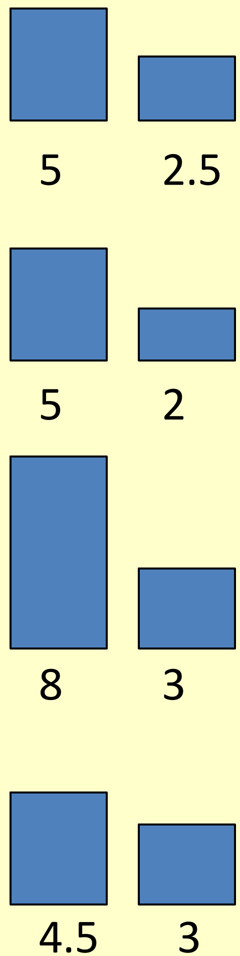
The rule is as follows, if the sum of the two bars is less than or equal to 10, it is an A. Otherwise it is a B.

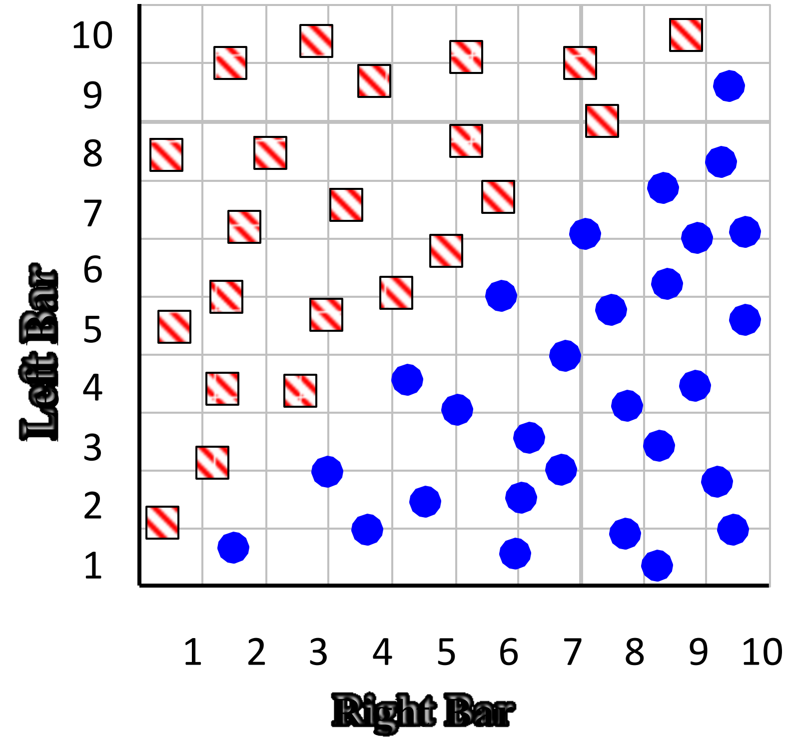
Pigeon Problem 1

Examples of class A



Examples of class B

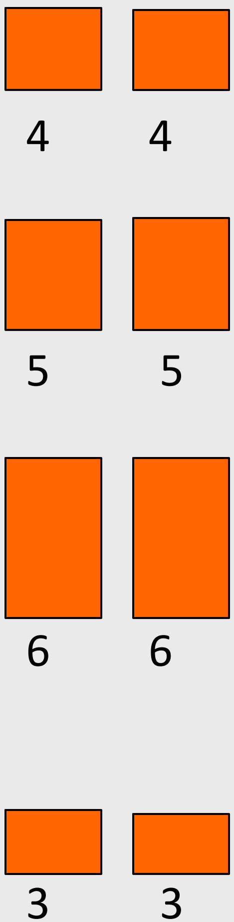




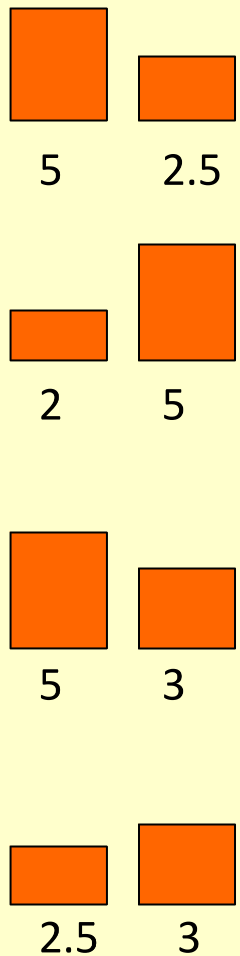
Here is the rule again. If the left bar is smaller than the right bar, it is an A, otherwise it is a B.

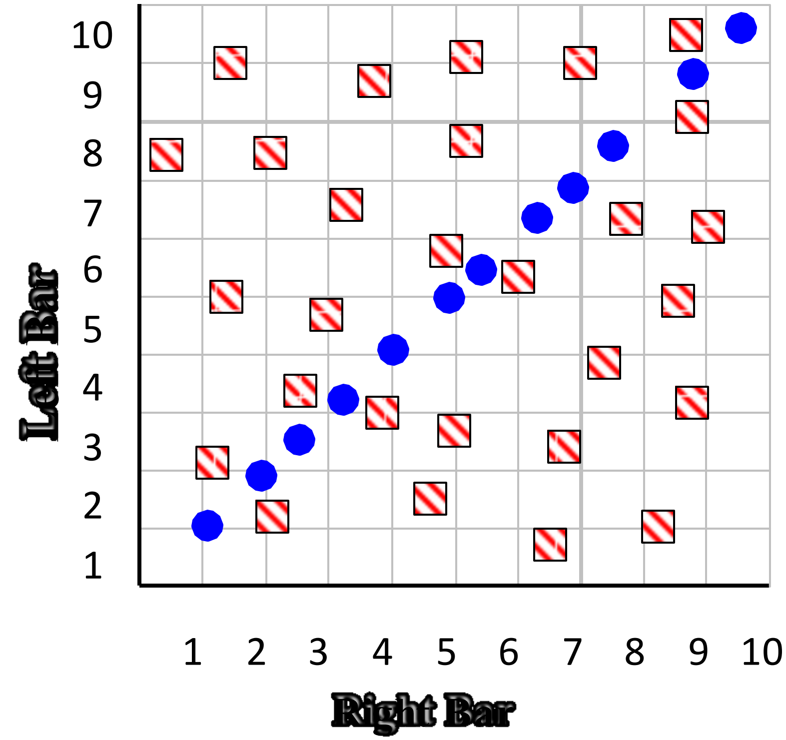
Pigeon Problem 2

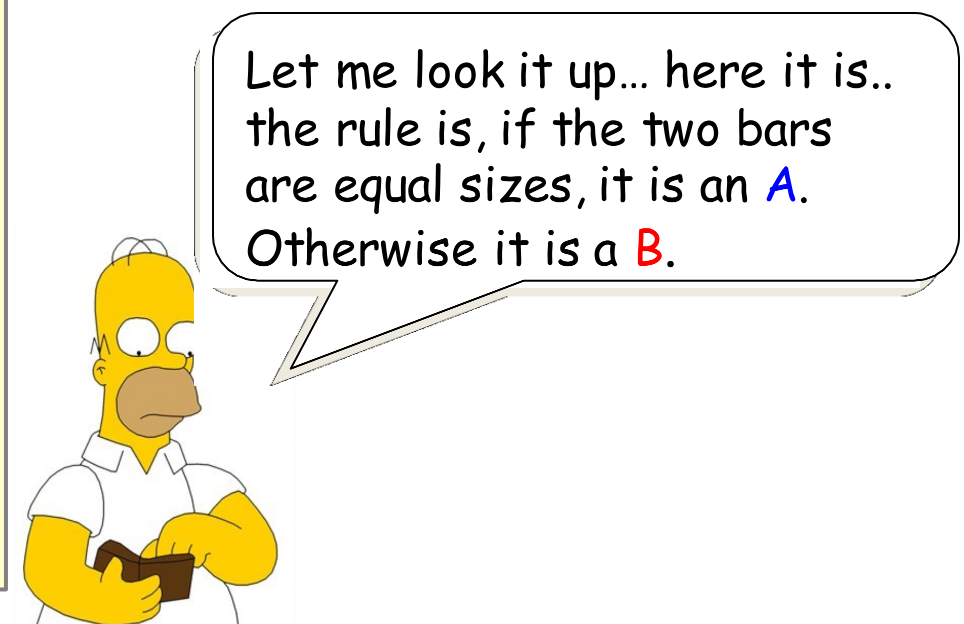
Examples of class A



Examples of class B

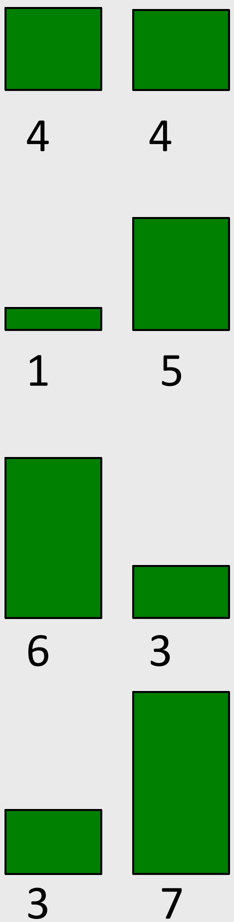




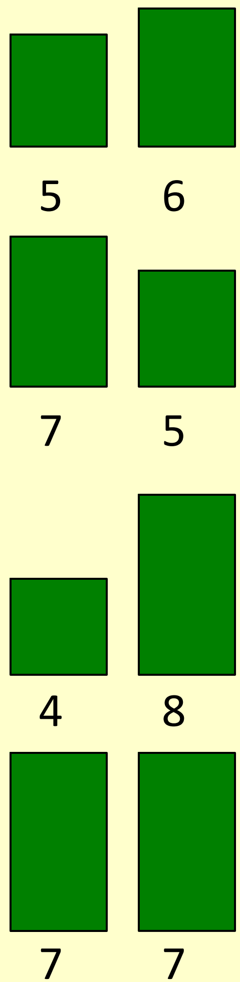


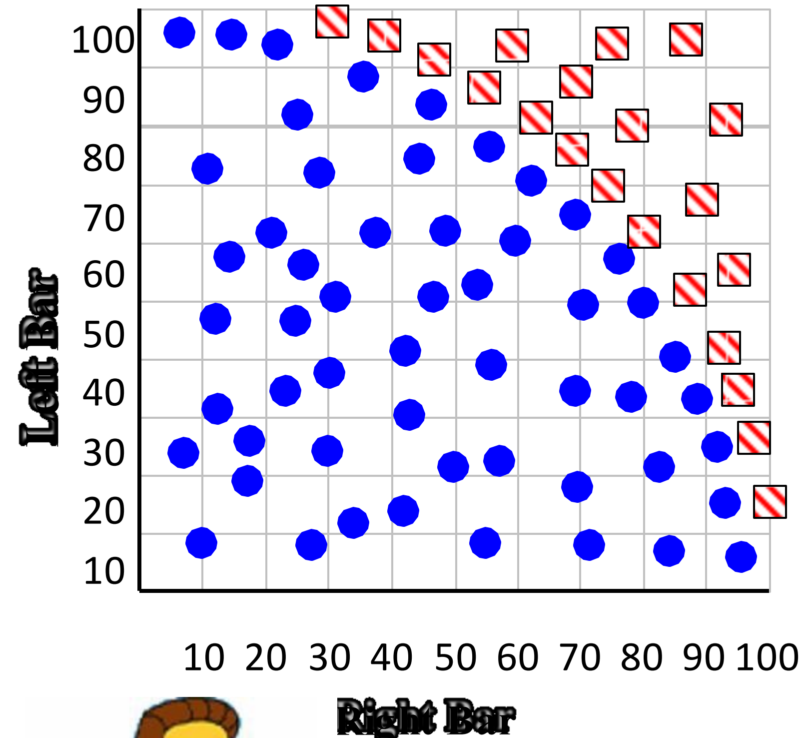
Pigeon Problem 3

Examples of class A



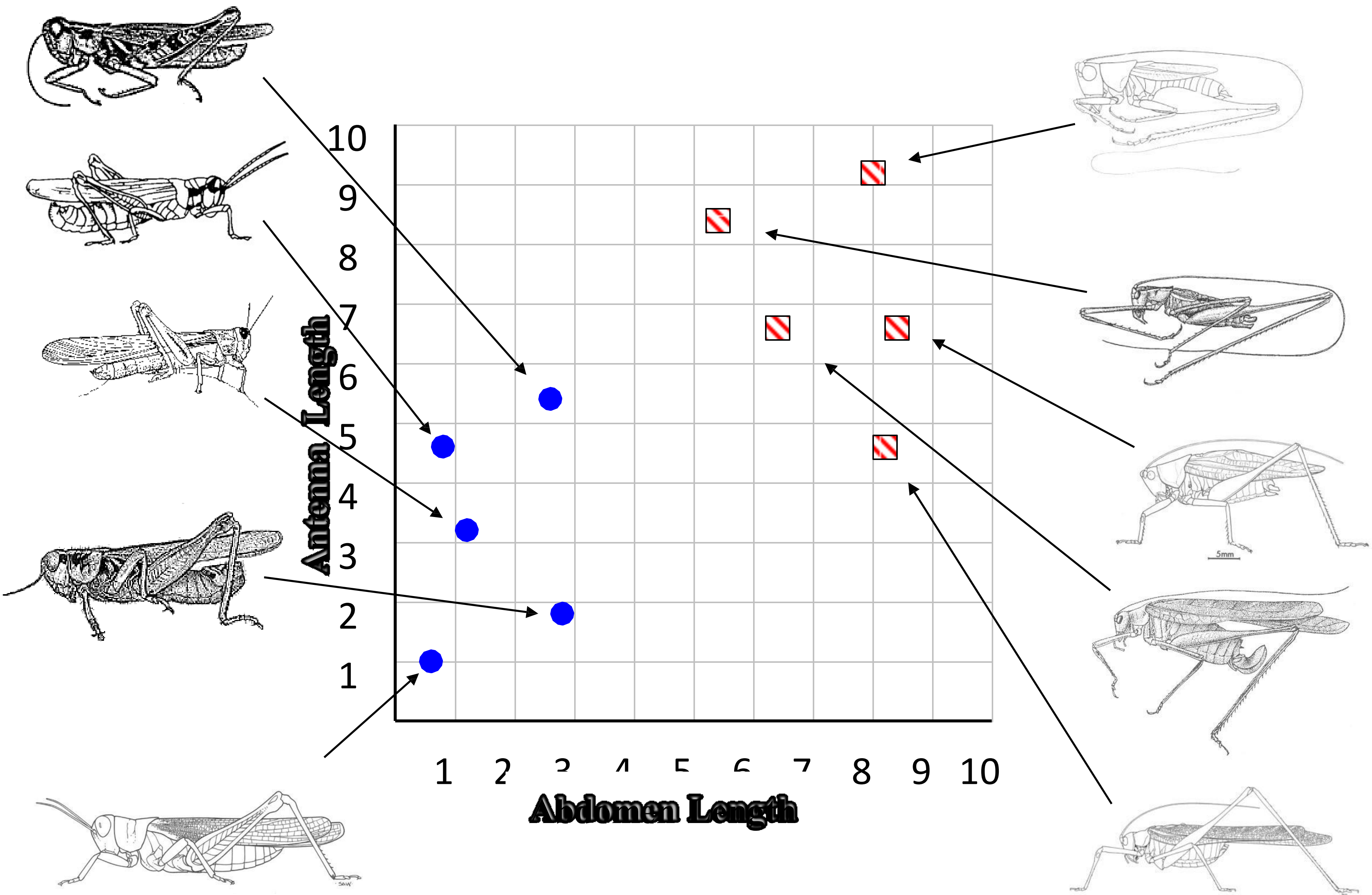
Examples of class B





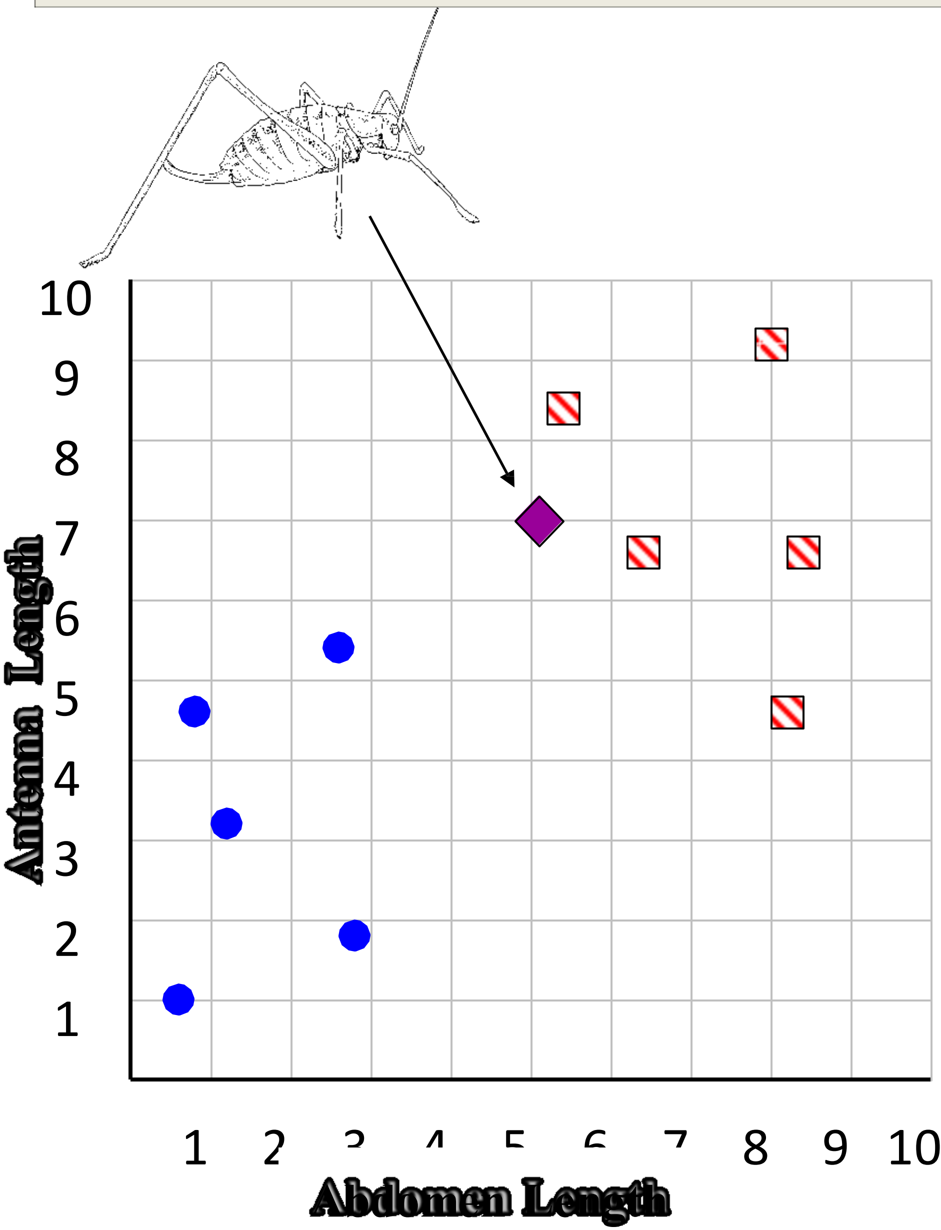
The rule again:
if the square of the sum of the two bars is less than or equal to 100, it is an A. Otherwise it is a B.

Grasshoppers **Katydids**



previously unseen instance =

11	**5.1**	**7.0**	**???????**



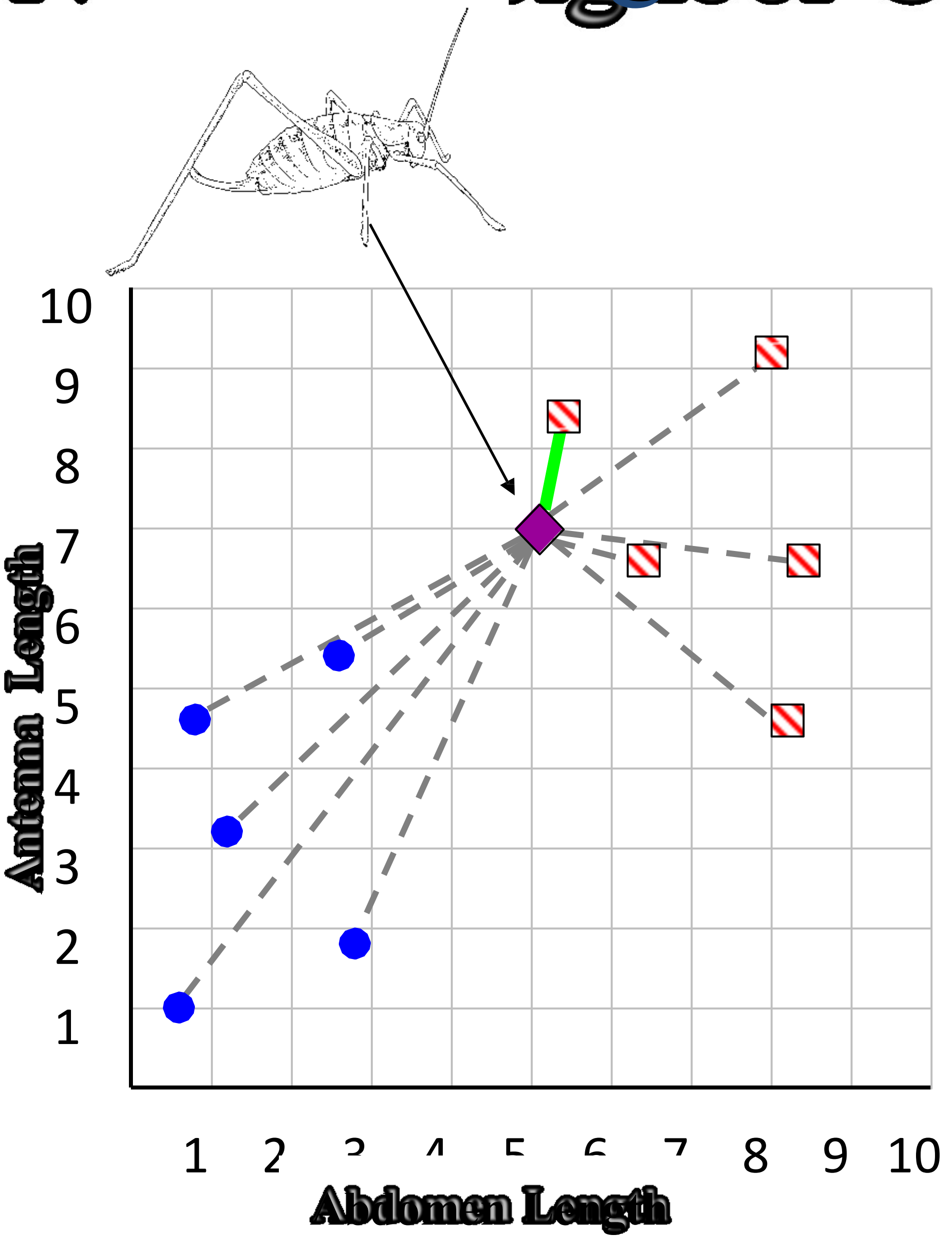
- We can "project" the previously unseen instance into the same space as the database.

- We have now abstracted away the details of our particular problem. It will be much easier to talk about points in space.

Katydids

Grasshoppers

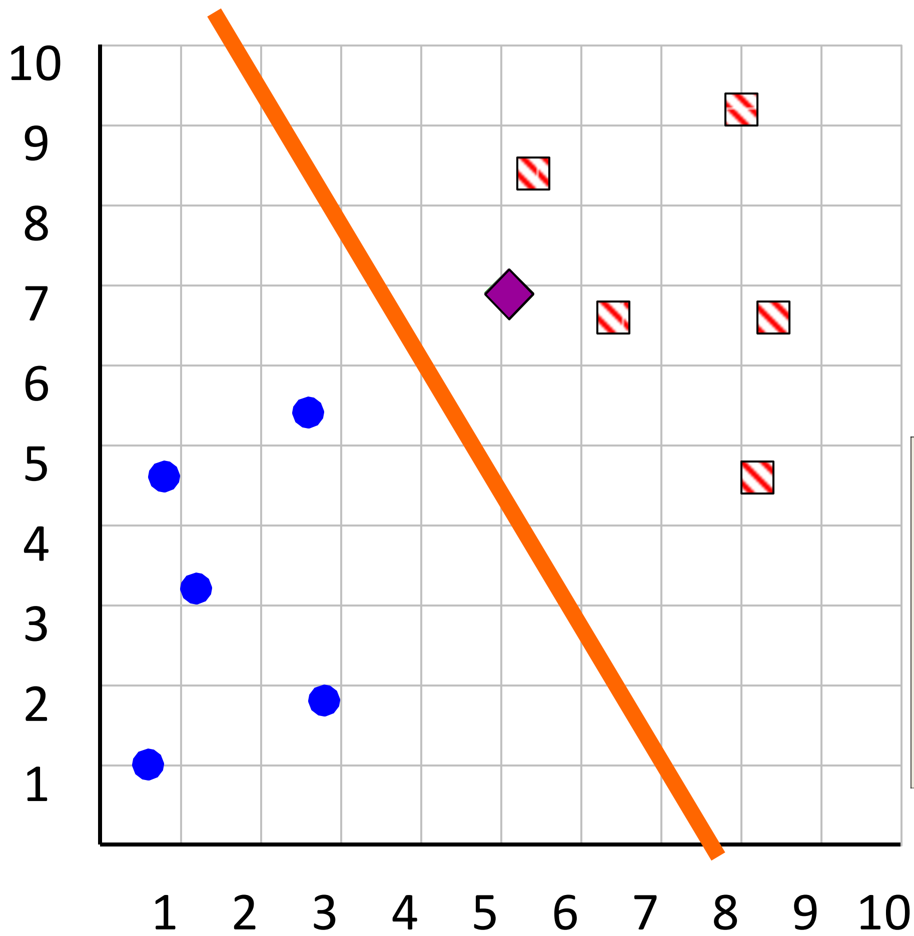
Nearest Neighbor Classifier



If the **nearest** instance to the **previously unseen instance** **is a** **Katydid**
  class is **Katydid**
else
  class is **Grasshopper**

Katydids
Grasshoppers

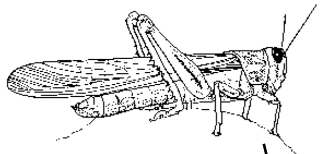
Simple Linear Classifier

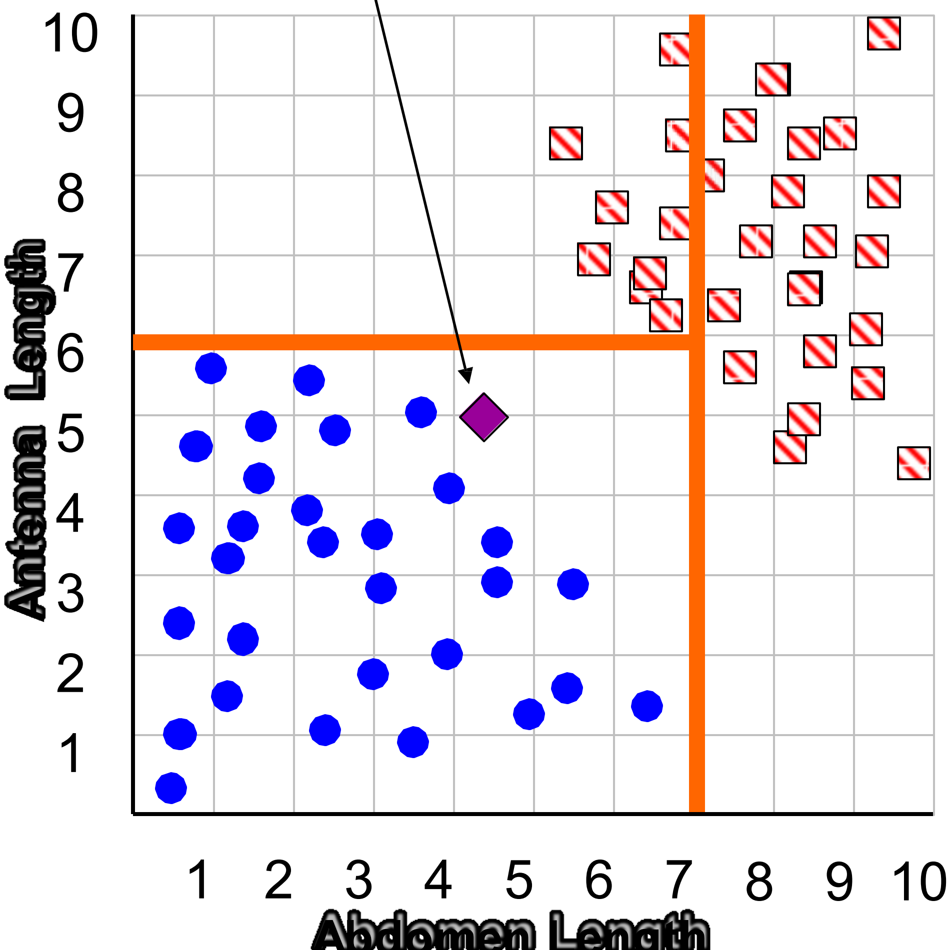


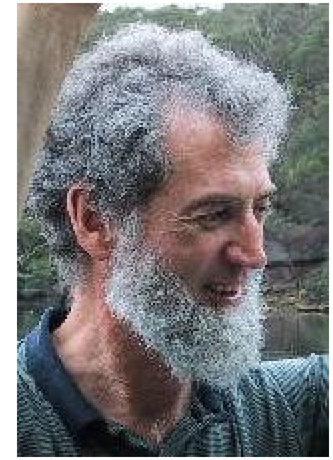
R.A. Fisher
1890-1962

If previously unseen instance above the line
then
    class is Katydid
else
    class is Grasshopper

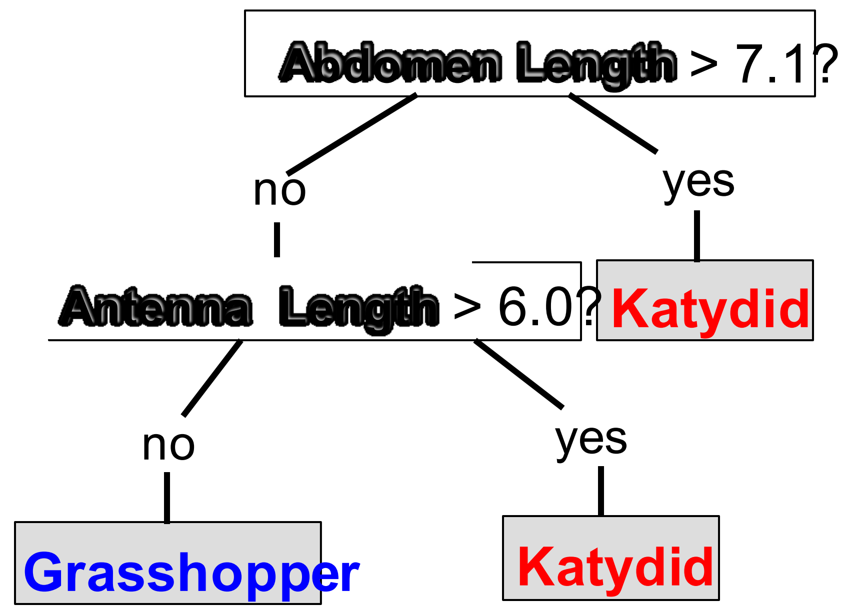
Decision Tree Classifier







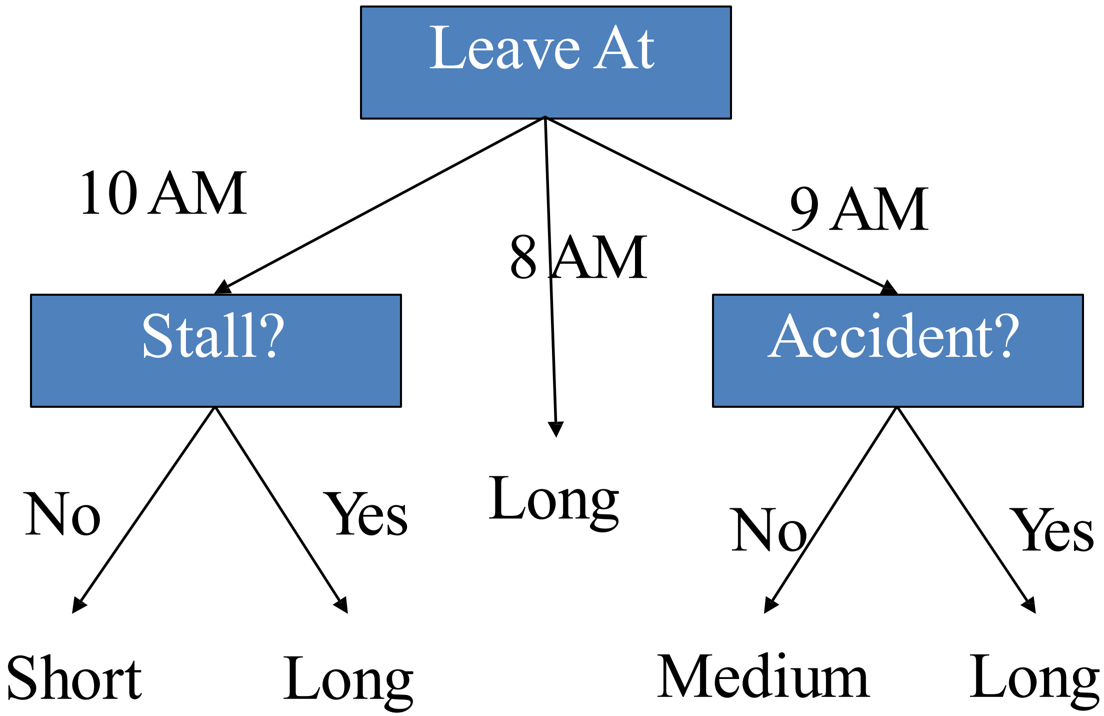
Ross Quinlan



What is a Decision Tree?

- An *inductive learning task*
  - Use particular facts to make more generalized conclusions
- A predictive model based on a branching series of Boolean tests
  - These smaller Boolean tests are less complex than a one-stage classifier
- Let's look at a sample decision tree…

Predicting Commute Time



If we leave at 10 AM and there are no cars stalled on the road, what will our commute time be?

Inductive Learning

- In this decision tree, we made a series of Boolean decisions and followed the corresponding branch
  - Did we leave at 10 AM?
  - Did a car stall on the road?
  - Is there an accident on the road?
- By answering each of these yes/no questions, we then came to a conclusion on how long our commute might take

Decision Tree as a Rule Set

```
if hour == 8am
    commute time = long
else if hour == 9am
    if accident == yes
        commute time = long
    else
        commute time = medium
else if hour == 10am
    if stall == yes
        commute time = long
    else
        commute time = short
```

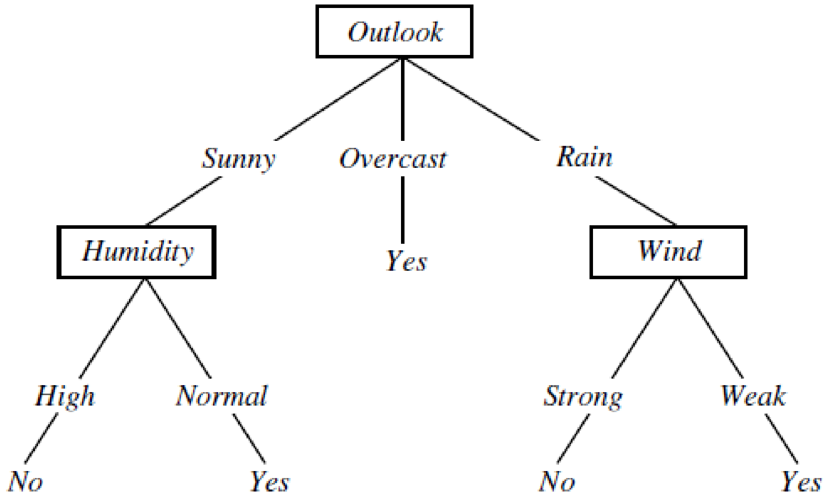
- We could have represented as a set of rules.
  However, this may be much harder to read...

Training Examples

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Building a Decision Tree

- An internal node is a test on an attribute
- A branch represents an outcome of the test, e.g. **outlook=sunny**
- A leaf node represents a class label
- At each node, one attribute is chosen to split training examples into distinct classes as much as possible
- A new case is classified by following a matching path to a leaf node



Decision Tree Algorithms

- The basic idea behind any decision tree algorithm is as follows:
  - Choose the *best* attribute(s) to split the remaining instances and make that attribute a decision node
  - Repeat this process recursively for each child
  - Stop when:
    - All the instances have the same target attribute value
    - There are no more attributes
    - There are no more instances

Entropy

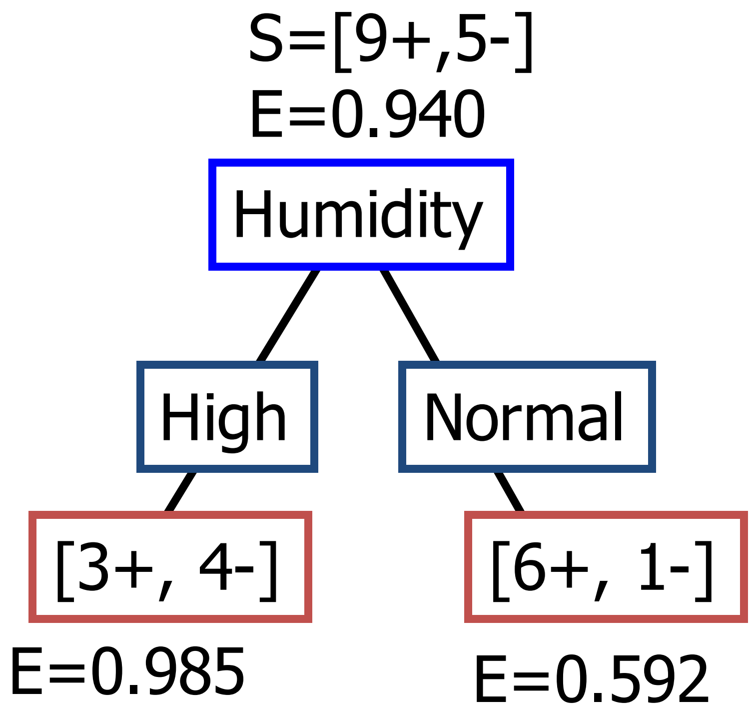
- Calculation of entropy
  - Entropy(S) = $\sum_{(i=1 \text{ to } l)} -|S_i|/|S| * \log_2(|S_i|/|S|)$
    - S = set of examples
    - S_i = subset of S with value v_i under the target attribute
    - l = size of the range of the target attribute

Entropy: Weather example

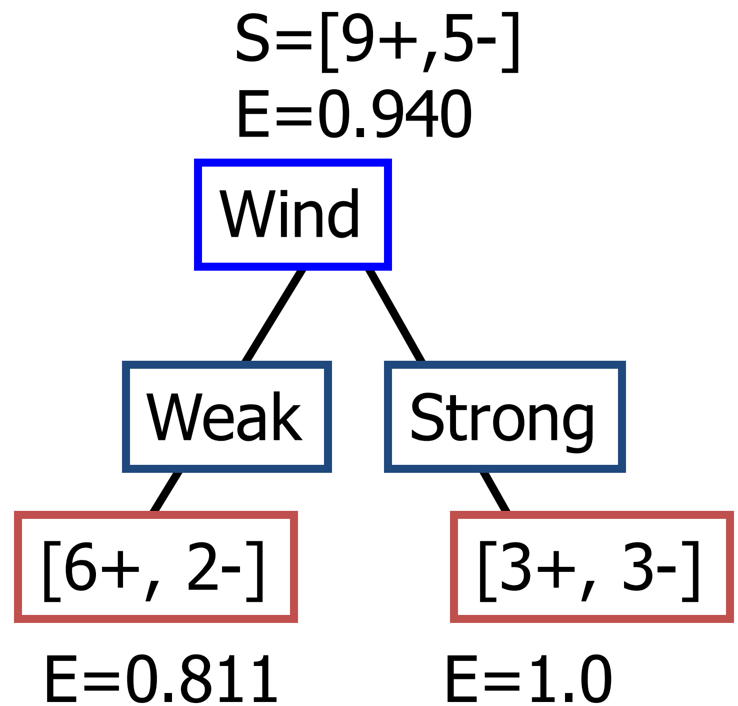
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	**No**
D2	Sunny	Hot	High	Strong	**No**
D3	Overcast	Hot	High	Weak	**Yes**
D4	Rain	Mild	High	Weak	**Yes**
D5	Rain	Cool	Normal	Weak	**Yes**
D6	Rain	Cool	Normal	Strong	**No**
D7	Overcast	Cool	Normal	Strong	**Yes**
D8	Sunny	Mild	High	Weak	**No**
D9	Sunny	Cool	Normal	Weak	**Yes**
D10	Rain	Mild	Normal	Weak	**Yes**
D11	Sunny	Mild	Normal	Strong	**Yes**
D12	Overcast	Mild	High	Strong	**Yes**
D13	Overcast	Hot	Normal	Weak	**Yes**
D14	Rain	Mild	High	Strong	**No**

$$\begin{aligned} Entropy([9+, 5-]) &= -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) \\ &= 0.940 \end{aligned}$$

Selecting the Next Attribute



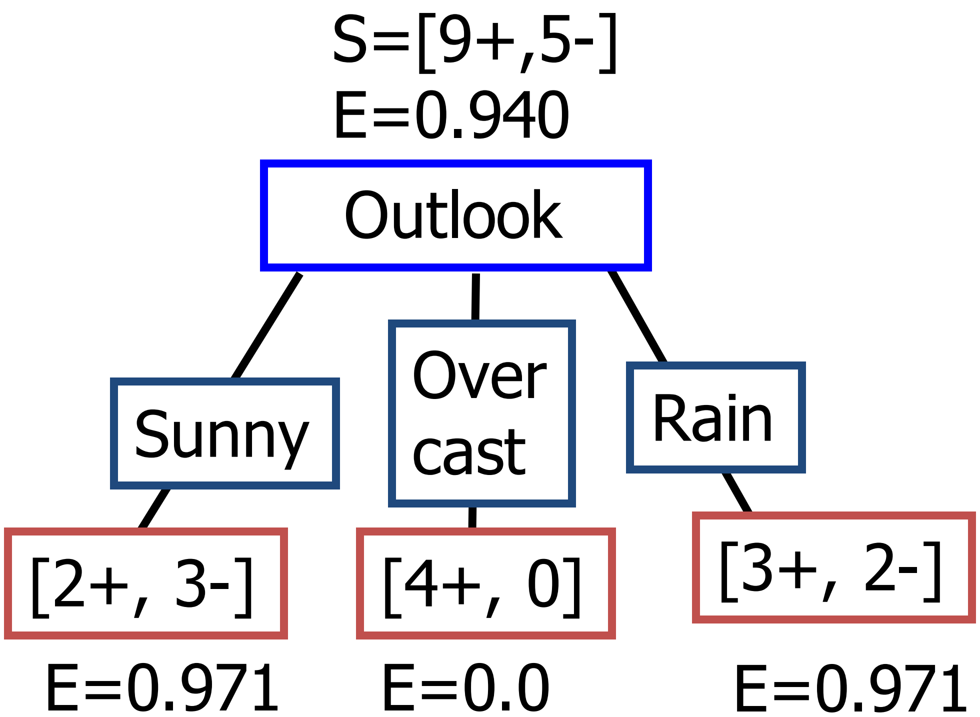
Gain(S,Humidity)
=0.940-(7/14)*0.985
– (7/14)*0.592
=0.151



Gain(S,Wind)
=0.940-(8/14)*0.811
– (6/14)*1.0
=0.048

Humidity provides greater info. gain than Wind, w.r.t target classification.

Selecting the Next Attribute



Gain(S,Outlook)
=0.940-(5/14)*0.971
-(4/14)*0.0 – (5/14)*0.0971
=0.247

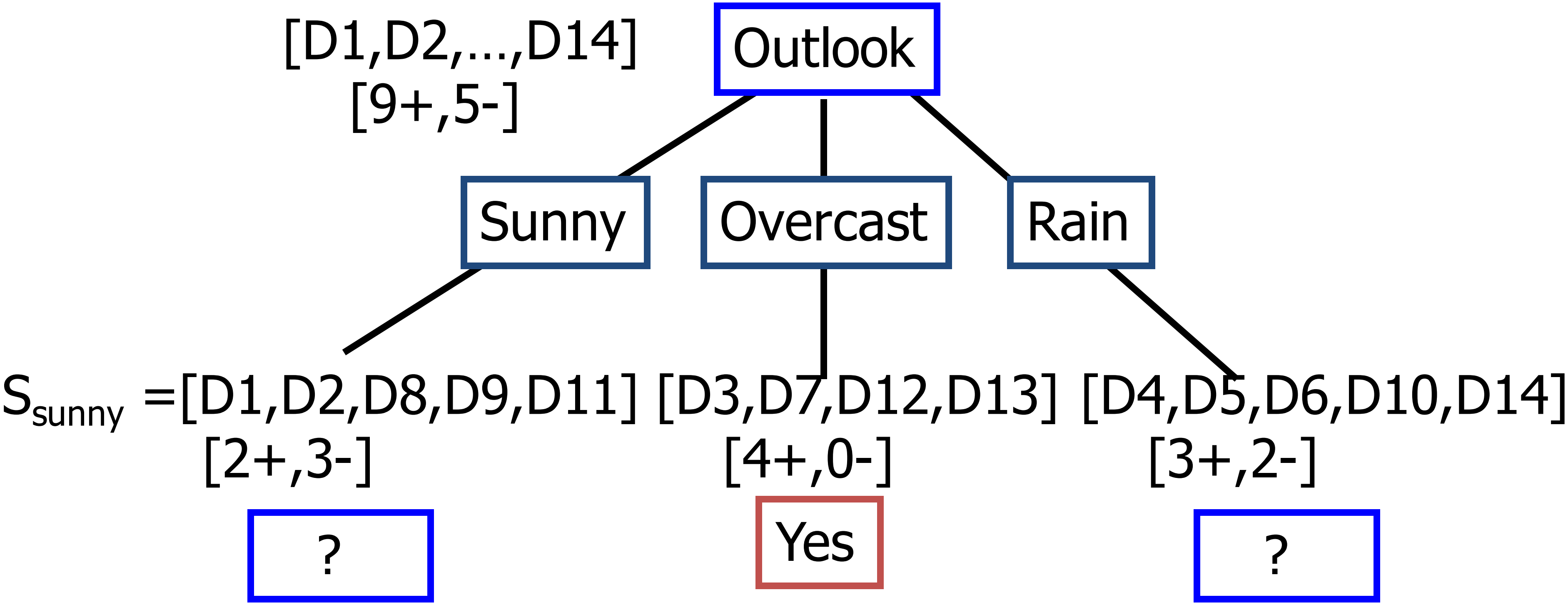
Selecting the Next Attribute

The information gain values for the 4 attributes are:

- Gain(S,Outlook) =0.247
- Gain(S,Humidity) =0.151
- Gain(S,Wind) =0.048
- Gain(S,Temperature) =0.029

where S denotes the collection of training examples

ID3 Algorithm

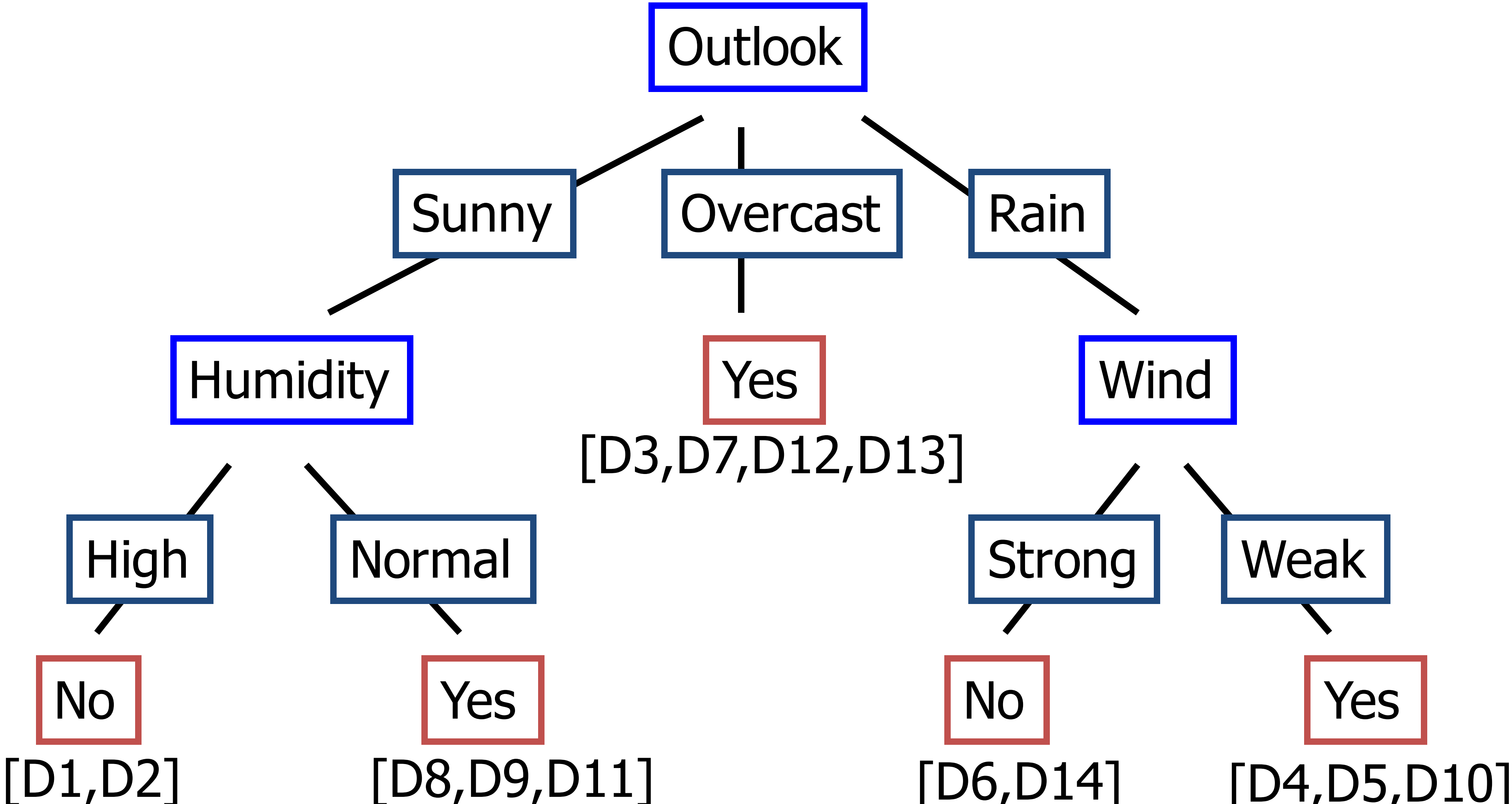


Gain(S_{sunny}, Humidity)=0.970-(3/5)0.0 – 2/5(0.0) = 0.970

Gain(S_{sunny}, Temp.)=0.970-(2/5)0.0 –2/5(1.0)-(1/5)0.0 = 0.570

Gain(S_{sunny}, Wind)=0.970= -(2/5)1.0 – 3/5(0.918) = 0.019

ID3 Algorithm



[mistake]

False positives – i.e. falsely predicting an event
False negatives – i.e. missing an incoming event

Similarly, we have “true positives” and “true negatives”

		Prediction	
		0	*1*
Truth	*0*	TN	**FP**
	1	**FN**	TP

Training and Test Sets

Training set (actual)

- Fit the model

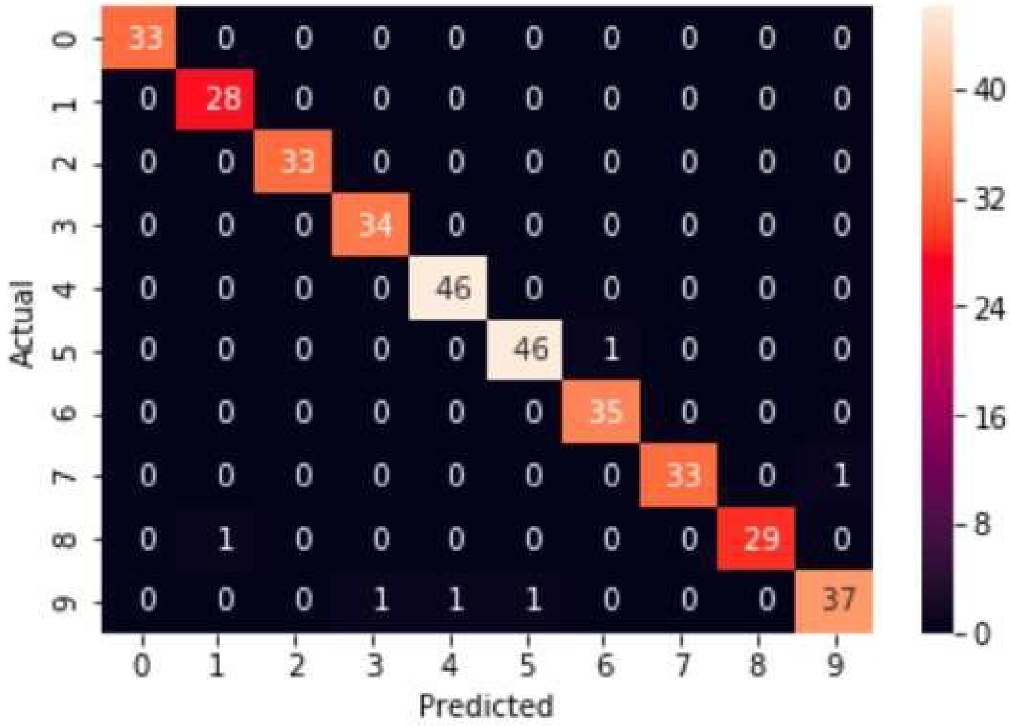
Test set (predicted)

- Measure performance

  Predict y with model

  Compare with actual y

  Measure error



Accuracy Measures

- Accuracy= (TP+TN)/(P+N)
- Sensitivity or true positive rate (TPR)= TP/(TP+FN) = TP/P
- Specificity or TNR= TN/(FP+TN) = TN/N
- Positive Predictive value (Precision) (PPV)= Tp/(Tp+Fp)
- Recall = Tp/(Tp+Fn)