BIOMISA

BIOmetrics, Medical Image and Signal Analysis Research Group

Machine Learning

EMERGING Technologies Lab

NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY PAKISTAN

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
- Role of Computer science: Efficient algorithms to
  - Solve the optimization problem
  - Representing and evaluating the model for inference

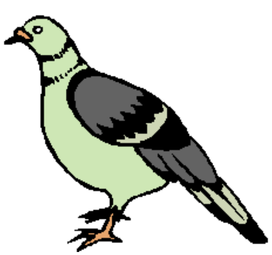
An Introduction to Machine Learning

- According to Herbert Simon, learning is, “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.” [G. F. Luger and W. A. Stubblefield, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, The Benjamin/Cummings Publishing Company, Inc. 1989.]

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

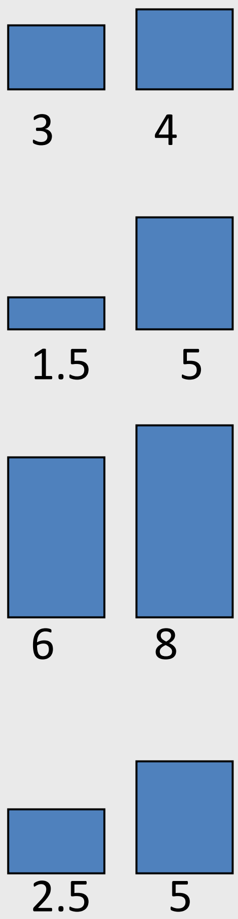
I am going to show you some classification problems which were shown to pigeons!

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

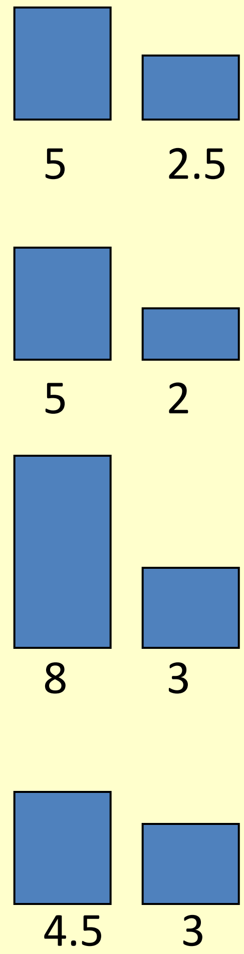


Pigeon Problem 1

Examples of class A

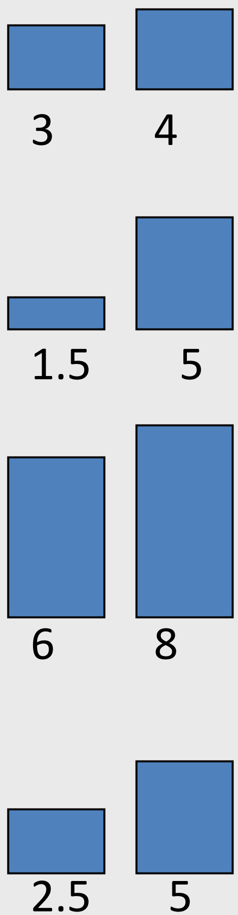


Examples of class B

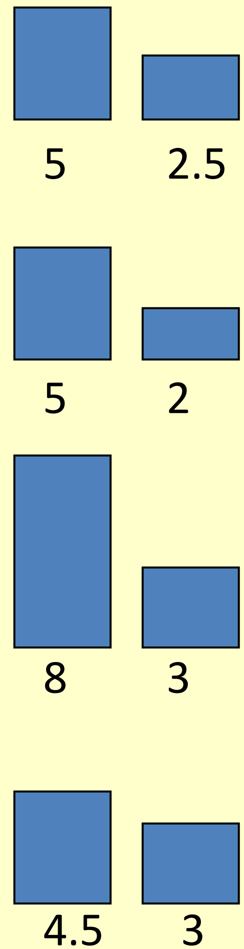


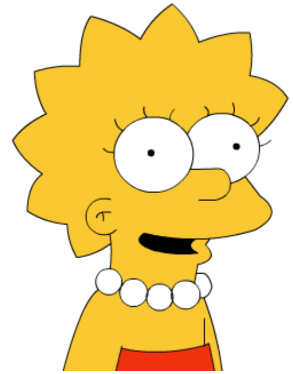
Pigeon Problem 1

Examples of class A

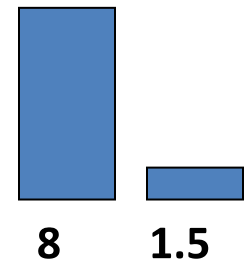


Examples of class B

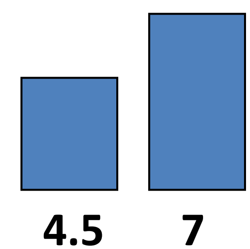




What class is this object?

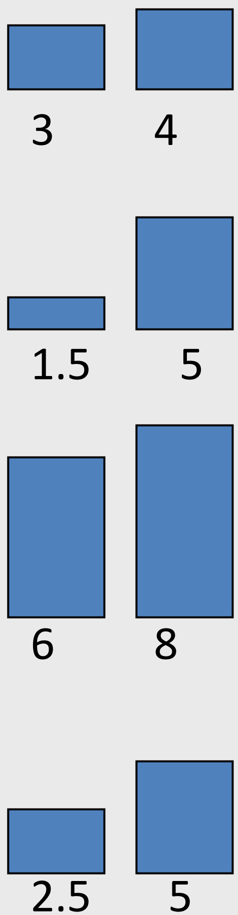


What about this one, A or B?

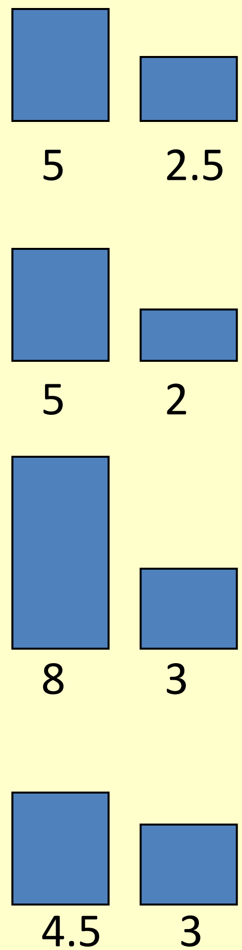


Pigeon Problem 1

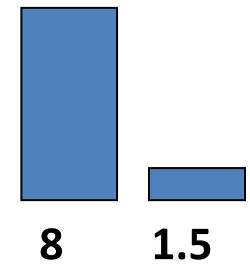
Examples of class A



Examples of class B



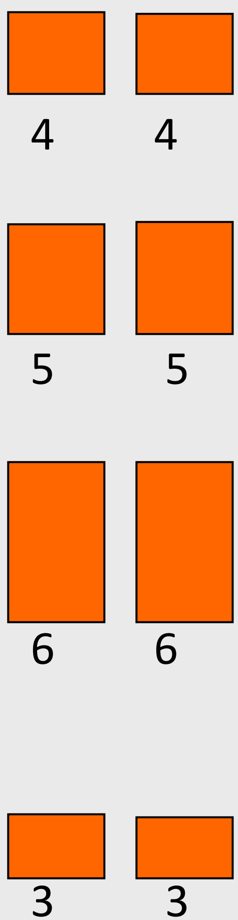
This is a B!



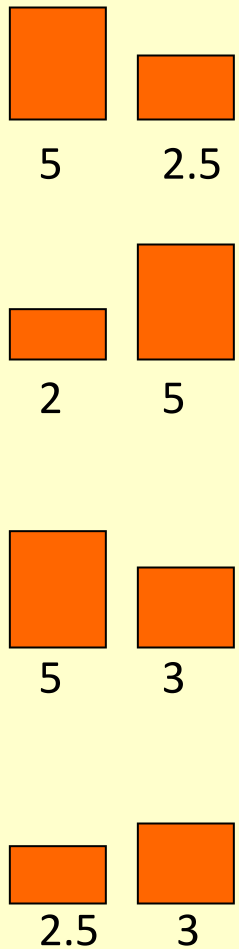
Here is the rule.
If the left bar is smaller than the right bar, it is an A, otherwise it is a B.

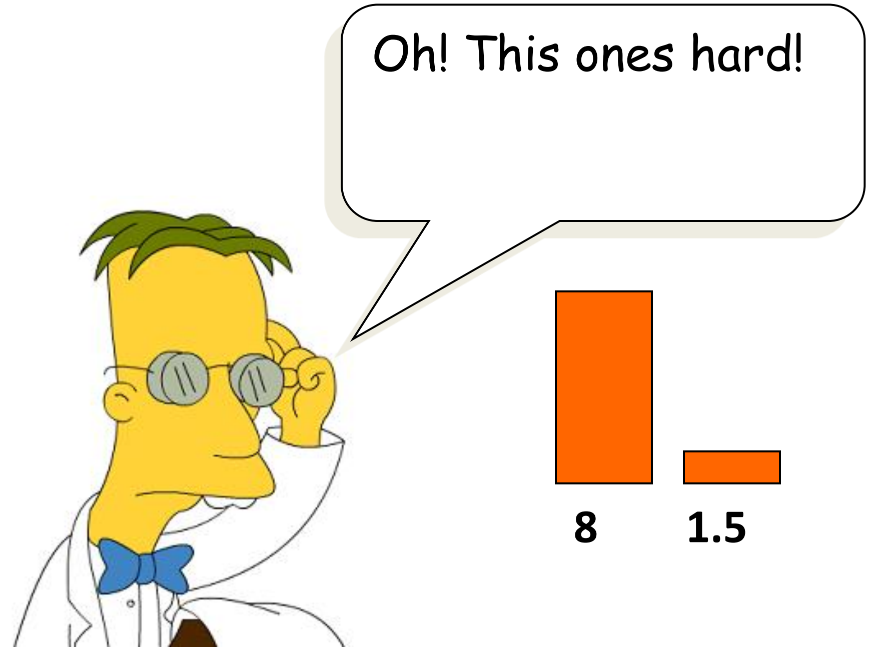
Pigeon Problem 2

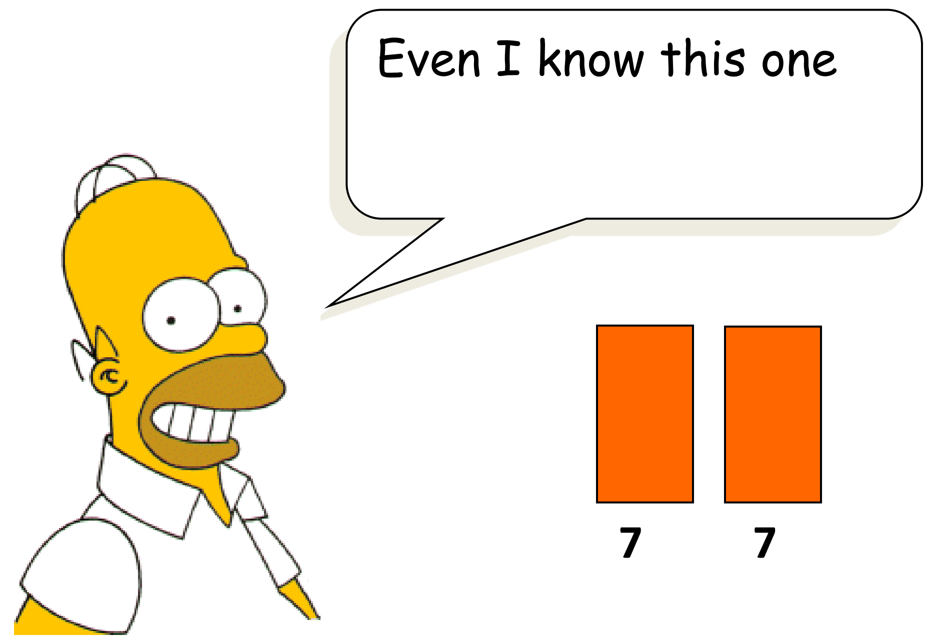
Examples of class A



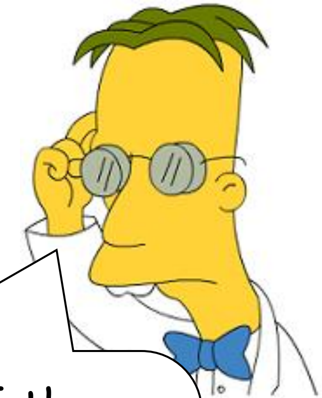
Examples of class B



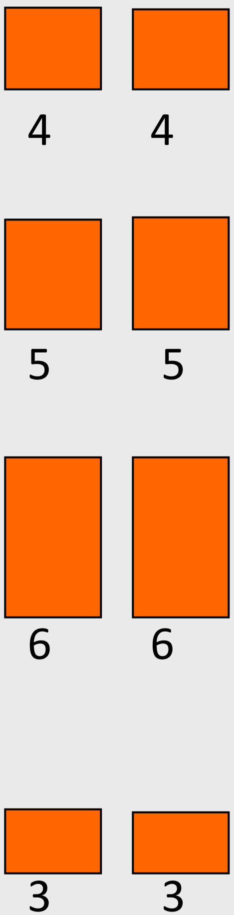




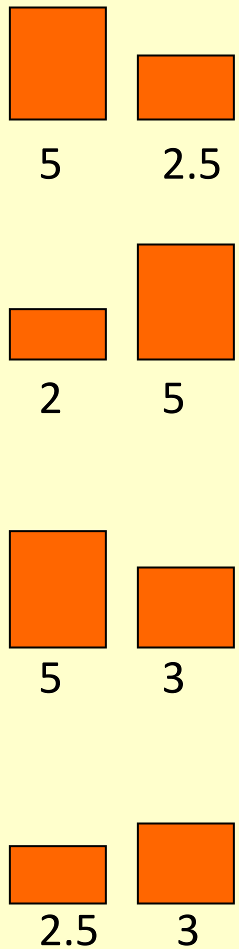
Pigeon Problem 2



Examples of class A

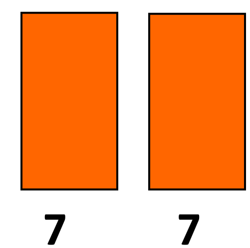


Examples of class B

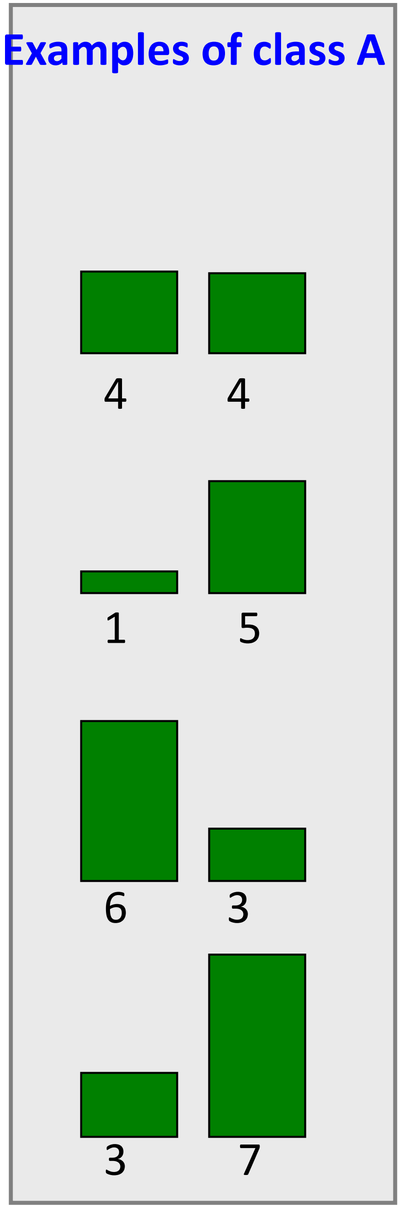


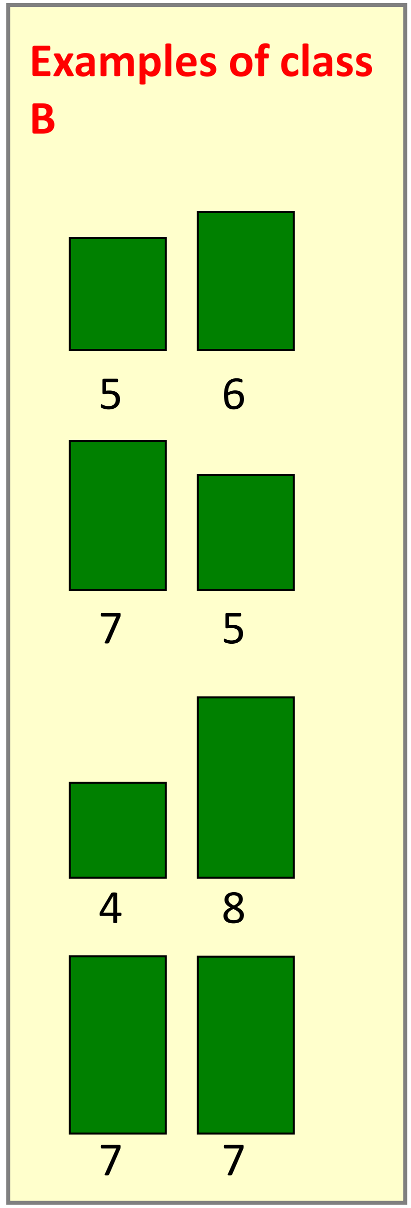
The rule is as follows, if the two bars are equal sizes, it is an A. Otherwise it is a B.

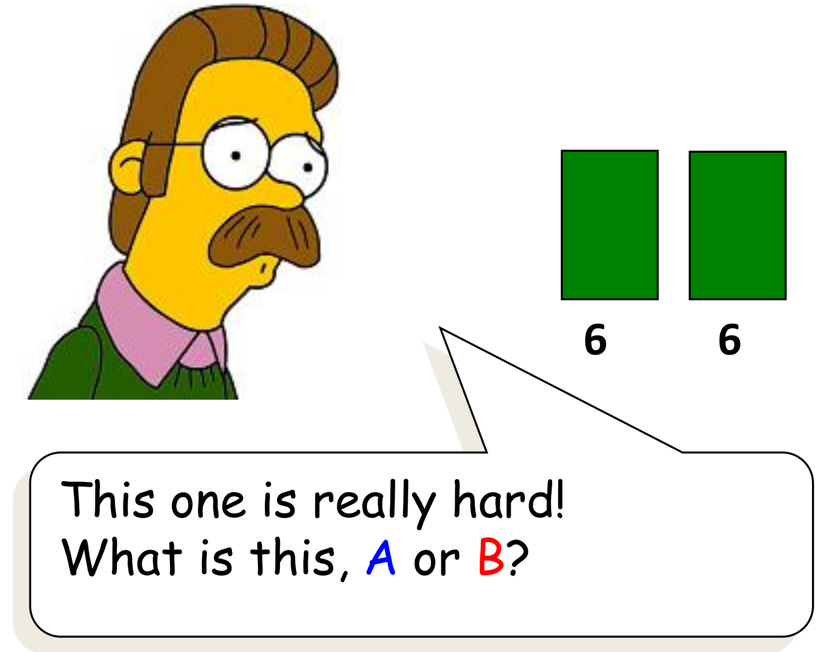
So this one is an A.



Pigeon Problem 3

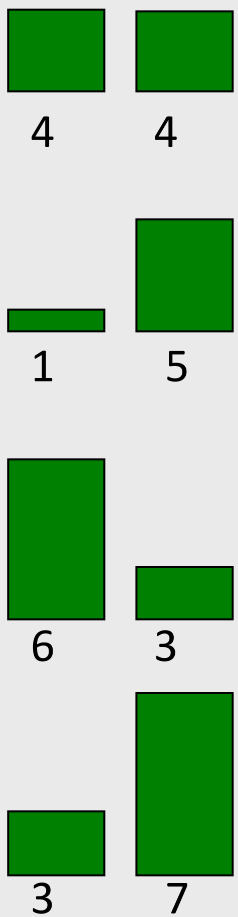




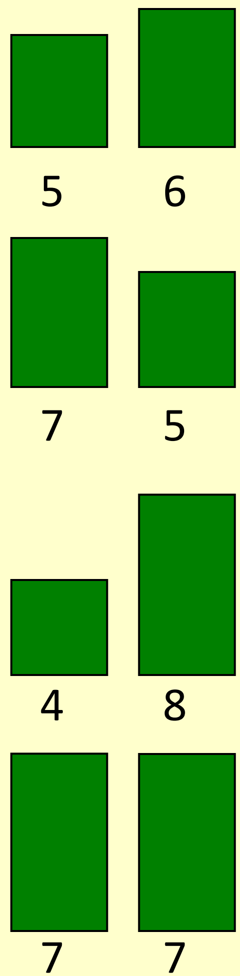


Pigeon Problem 3

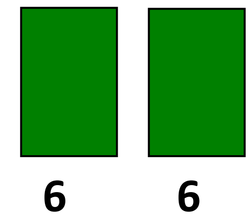
Examples of class A



Examples of class B

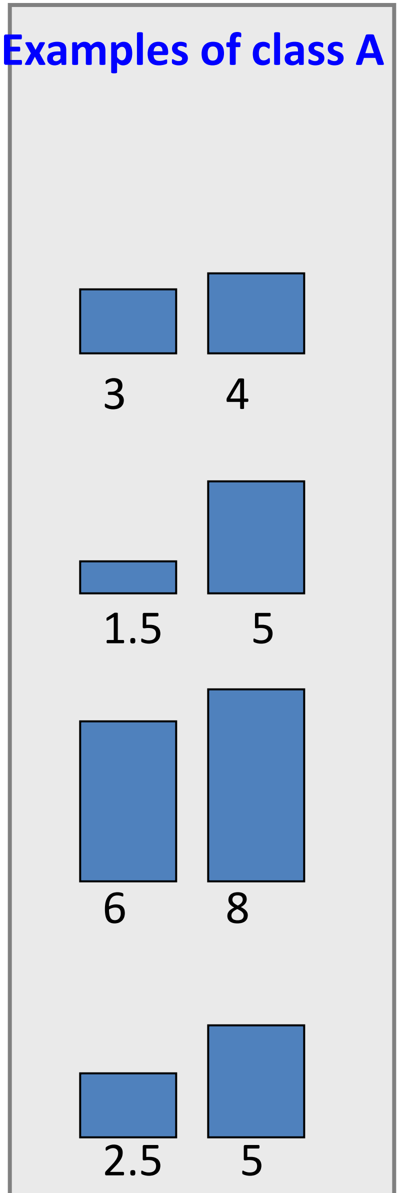


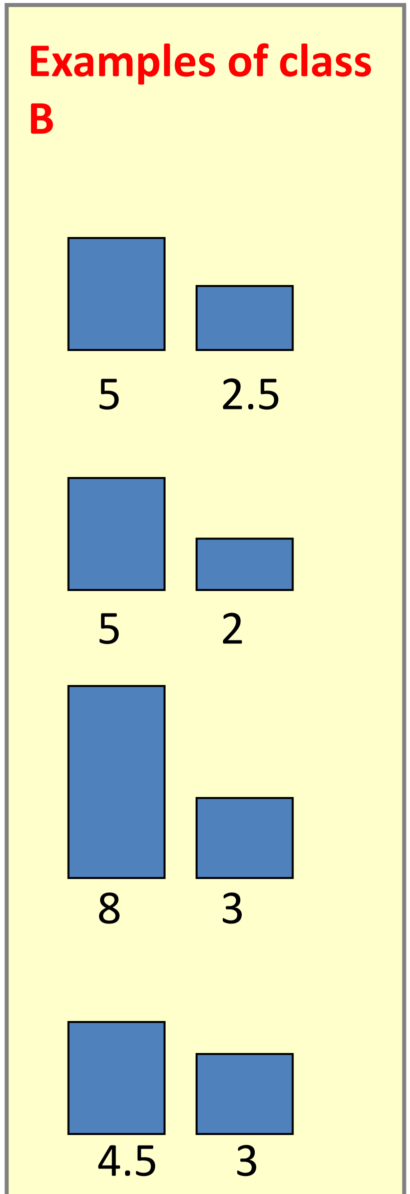
It is a B!

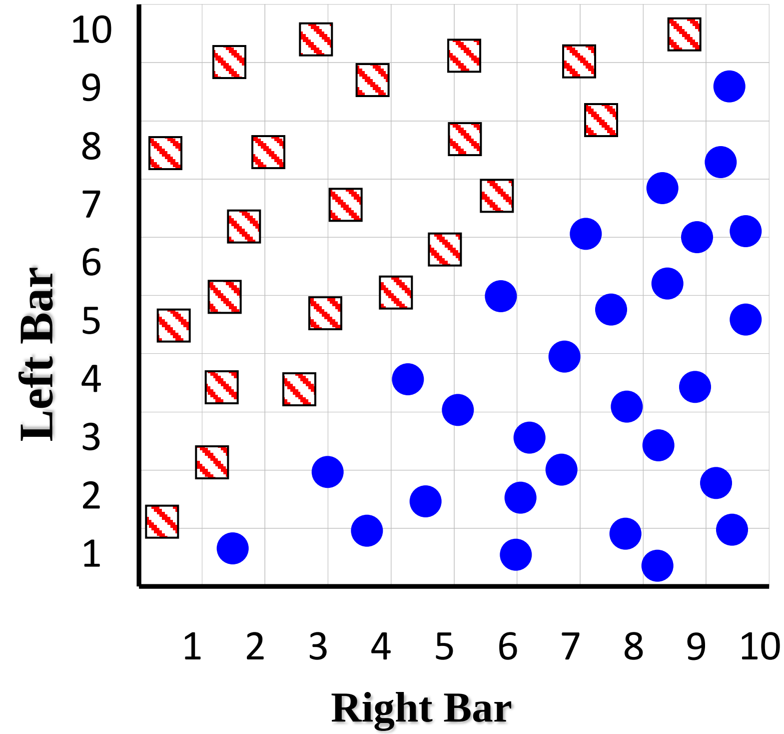


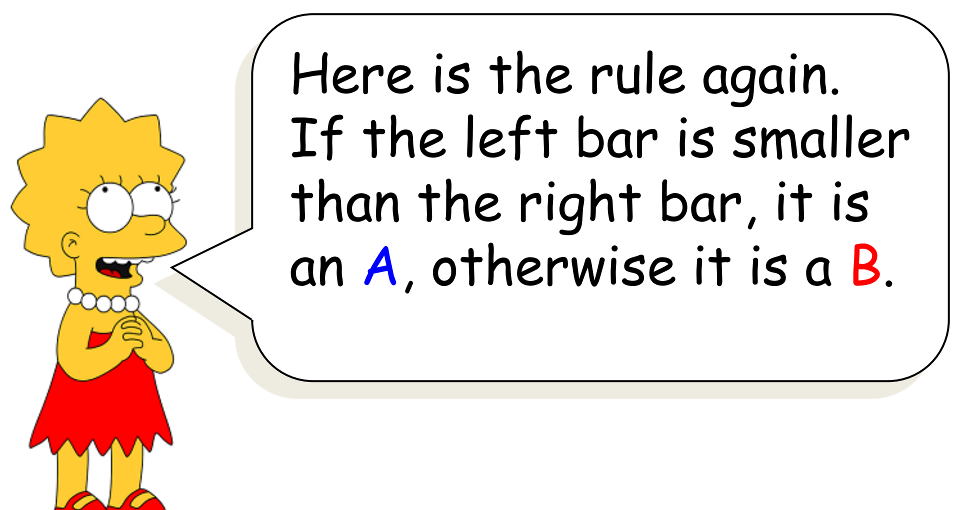
The rule is as follows, if the sum of the two bars is less than or equal to 10, it is an *A*. Otherwise it is a B.

Pigeon Problem 1

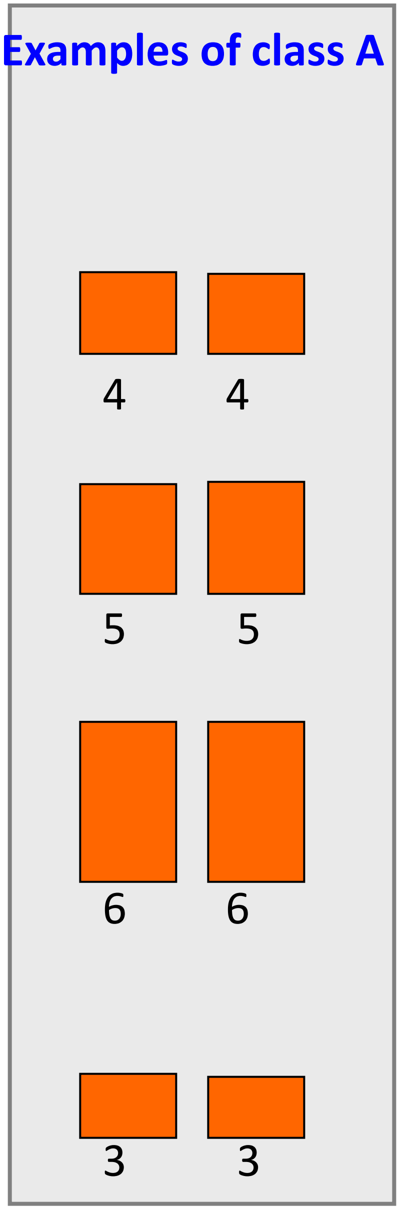


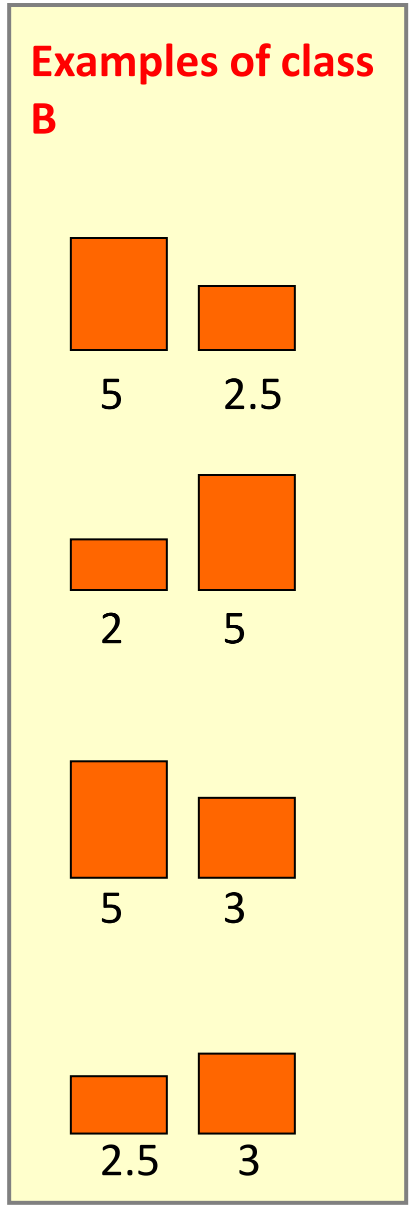


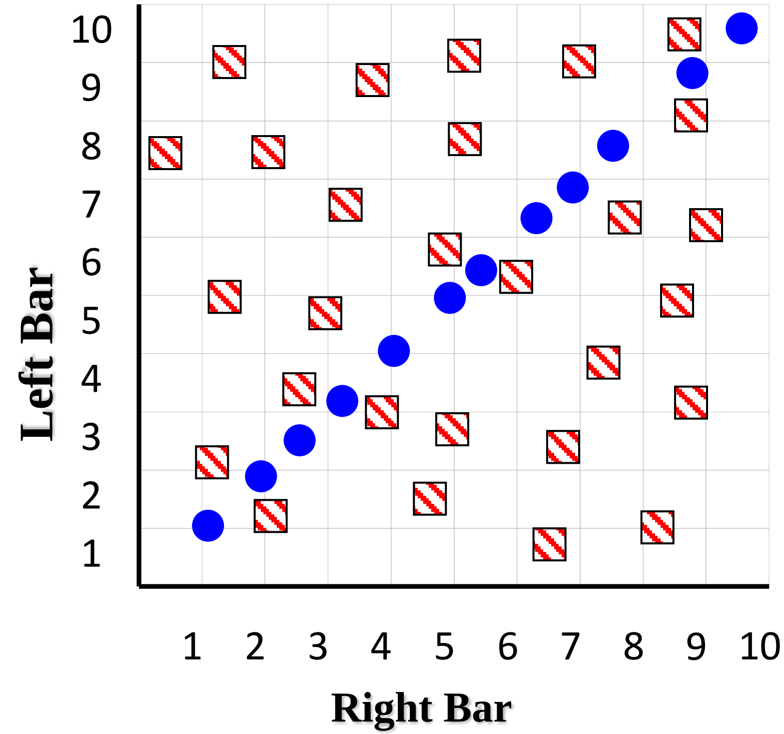


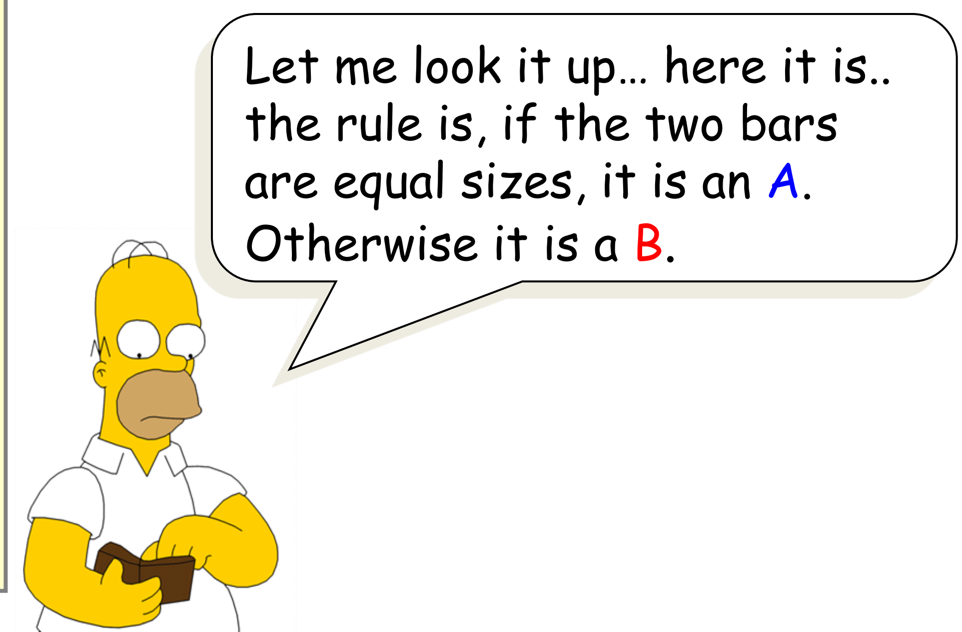


Pigeon Problem 2



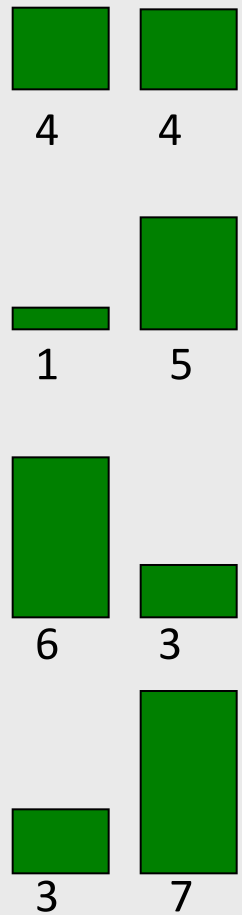




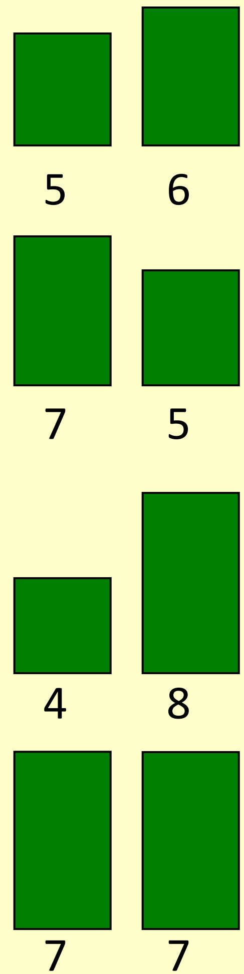


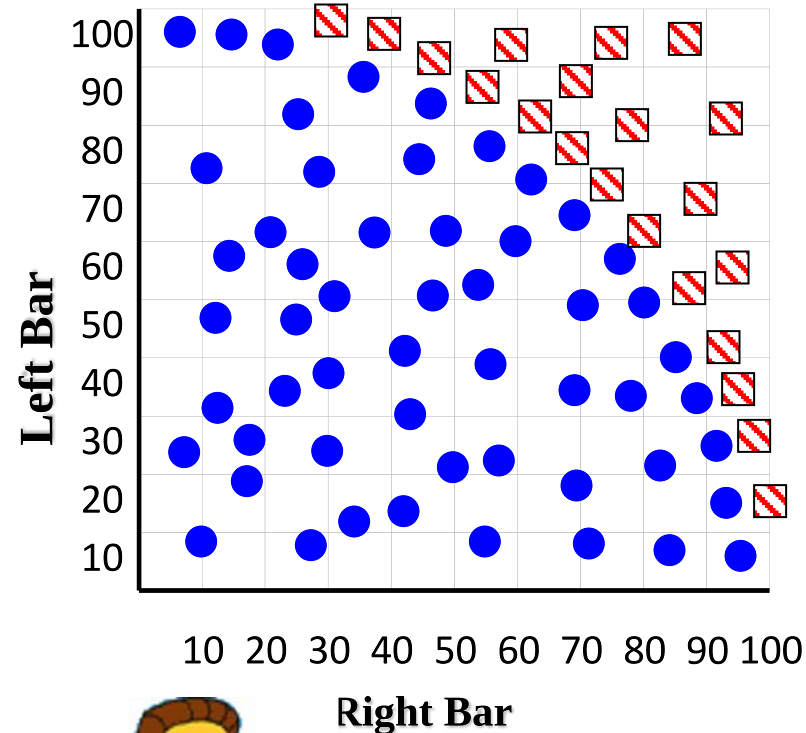
Pigeon Problem 3

Examples of class A



Examples of class B



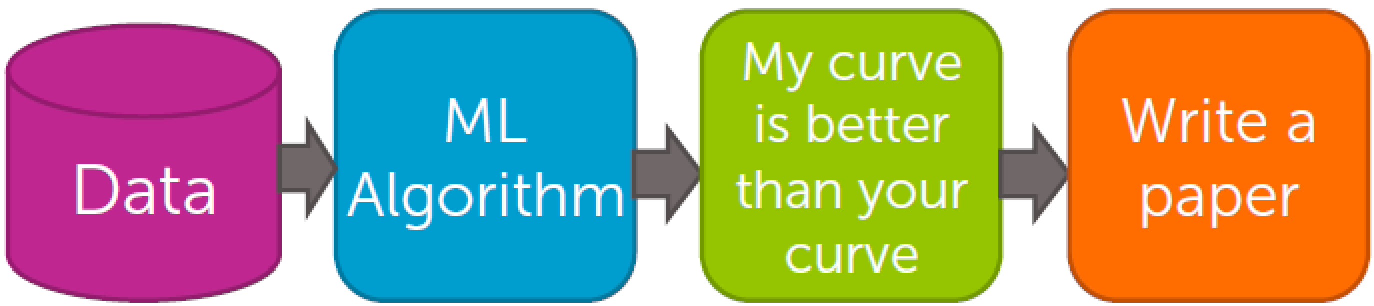


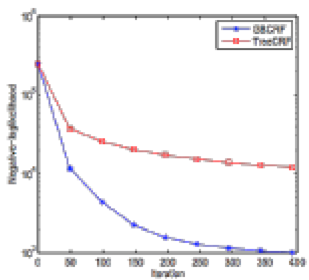
The rule again:
if the square of the sum of the two bars is less than or equal to 100, it is an A. Otherwise it is a B.

Why “Learn”?

- Machine learning is programming computers to optimize a performance criterion using example data or past experience.
- Learning is used when:
  - Human expertise does not exist (navigating on Mars),
  - Humans are unable to explain their expertise (speech recognition)
  - Solution changes in time (routing on a computer network)
  - Solution needs to be adapted to particular cases (user biometrics)

OLD VIEW OF ML:





Neural Information Processing Systems Foundation

ICML

Disruptive companies differentiated by

INTELLIGENT APPLICATIONS

using

Machine Learning

amazon
Retail

Google PageRank
Search

livingsocial
Coupons

NETFLIX
Movie Distribution

LinkedIn
Networking

PANDORA
Music

Obama'08
Campaigning

Google Adsense
Advertising

Zillow
Real Estate

glassdoor
Human Resources

Avvo
Legal Advice

eHarmony
Dating

fitbit
Wearables

UBER
Taxis

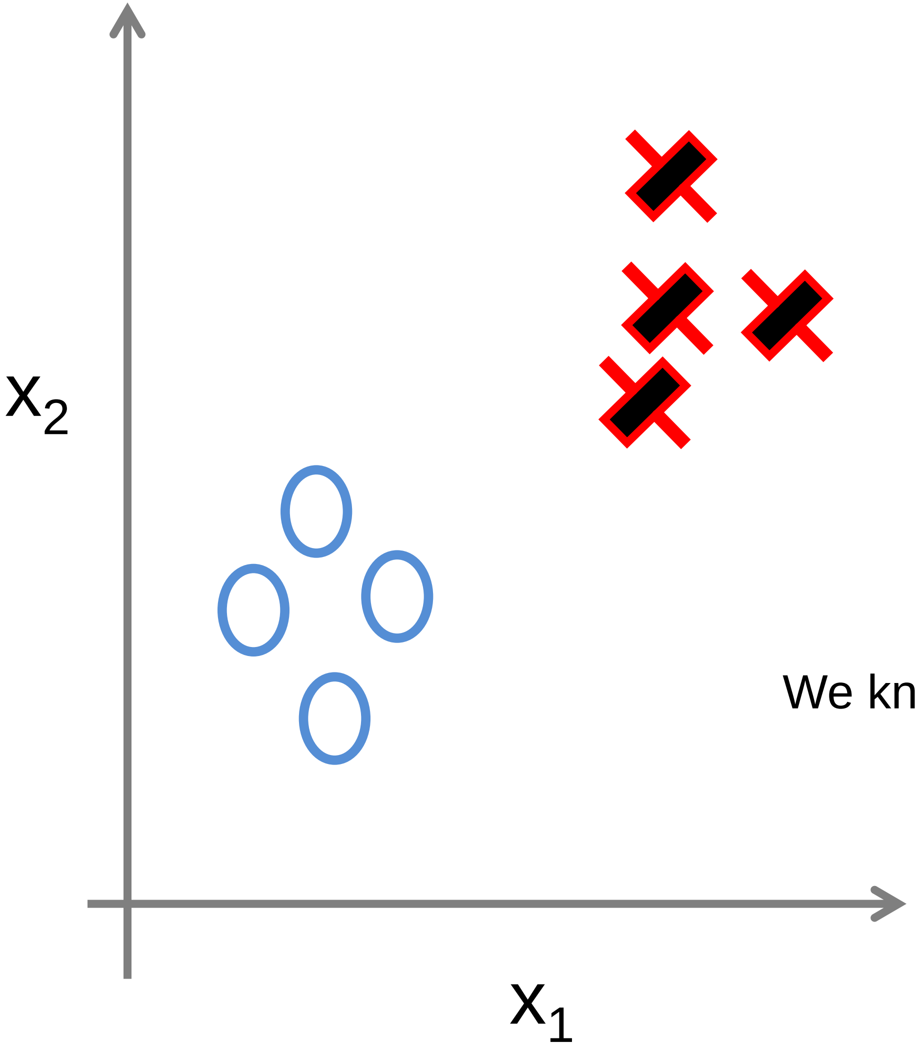
RelateIQ
CRM

The machine learning pipeline

ML Methods

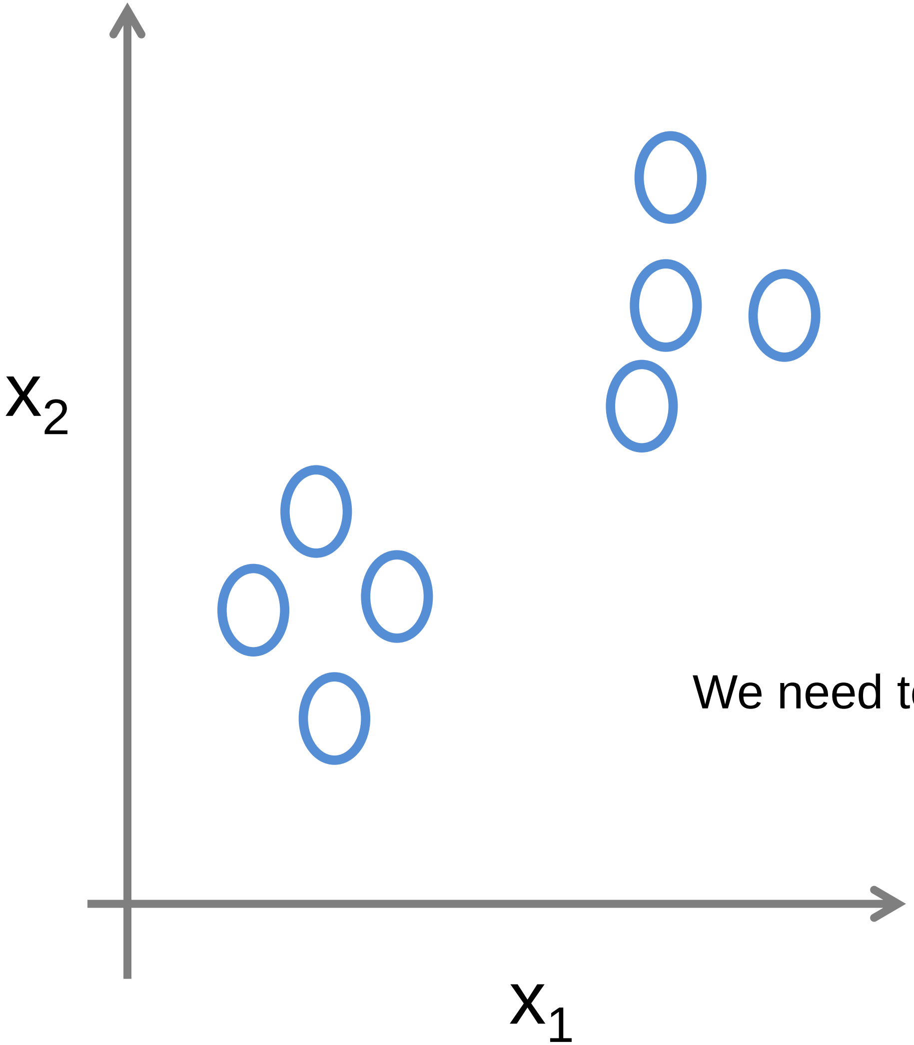
- Supervised Learning
  - Classification
  - Regression/Prediction
- Unsupervised Learning
- Association Analysis

Supervised Learning



We knew the correct answers

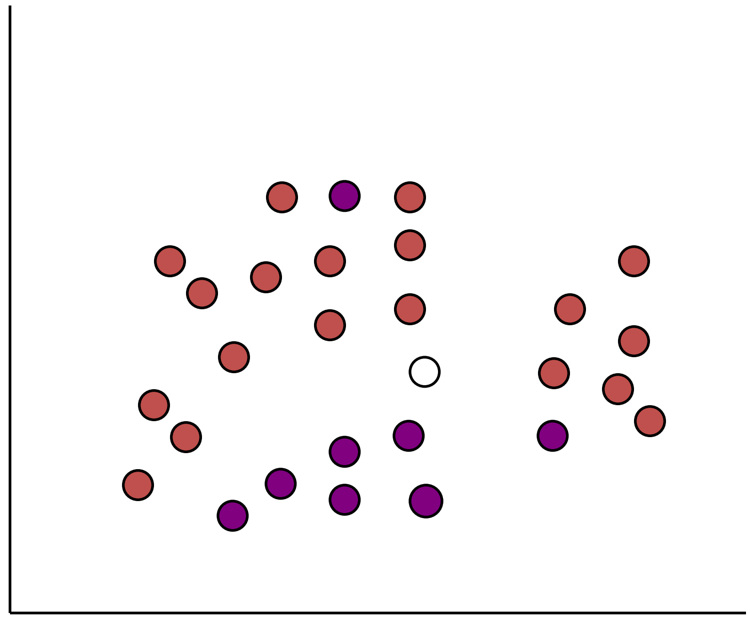
Unsupervised Learning



We need to figure out the patterns

Classification

Learn a method for predicting the instance class from pre-labeled (classified) instances

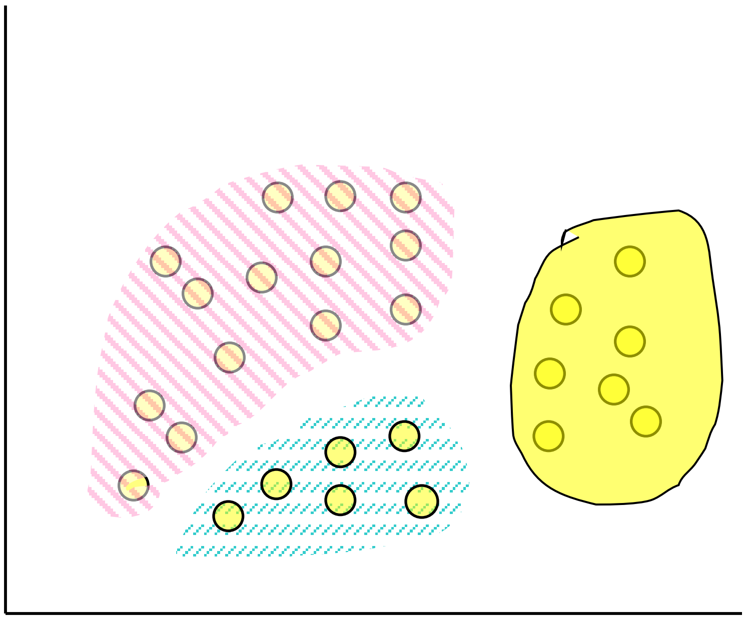


Many approaches: Statistics, Decision Trees, Neural Networks,

...

Clustering

Find "natural" grouping of instances given un-labeled data



Learning Associations

- Basket analysis:

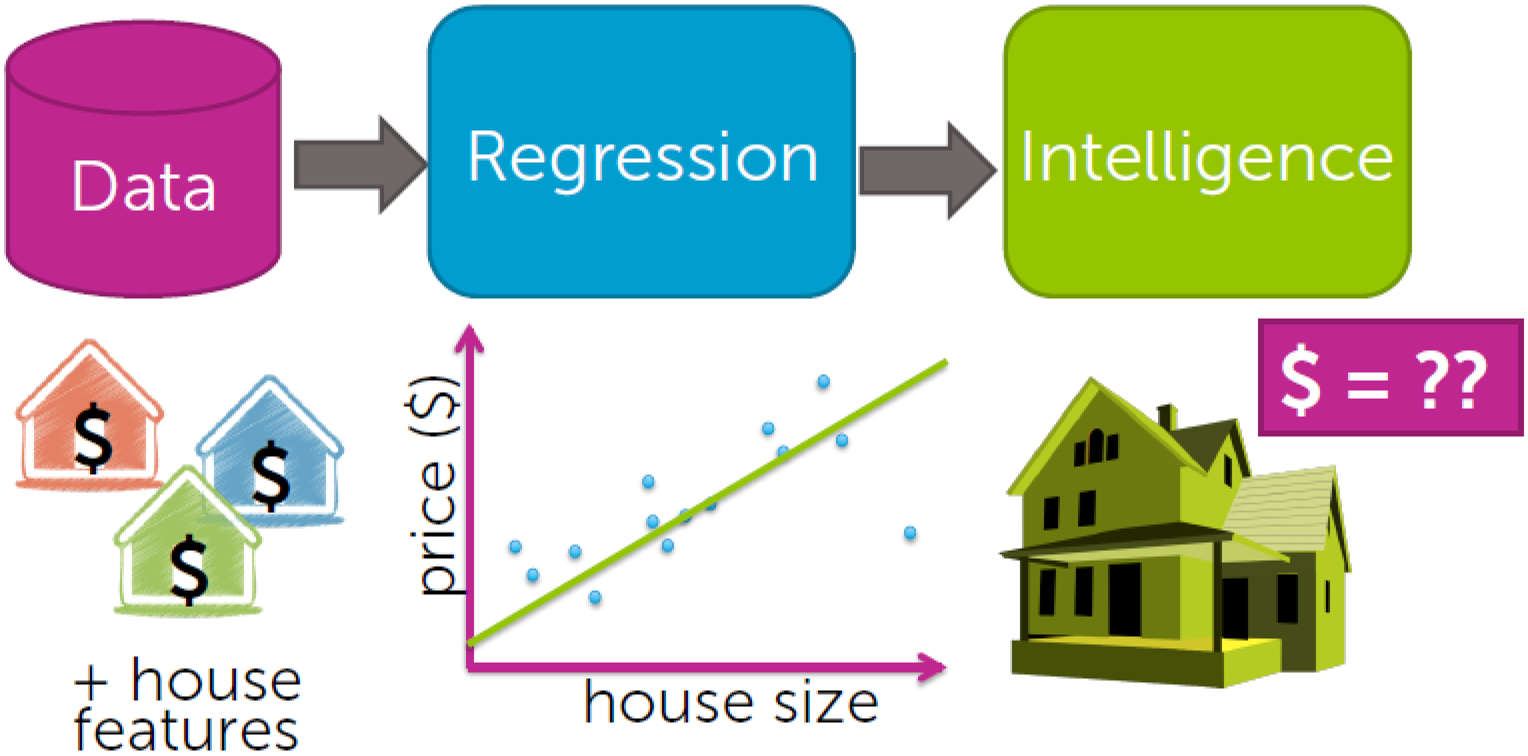
  $P(Y \mid X)$ probability that somebody who buys X also buys Y where X and Y are products/services.

  Example: P (bread | cold drink) = 0.7

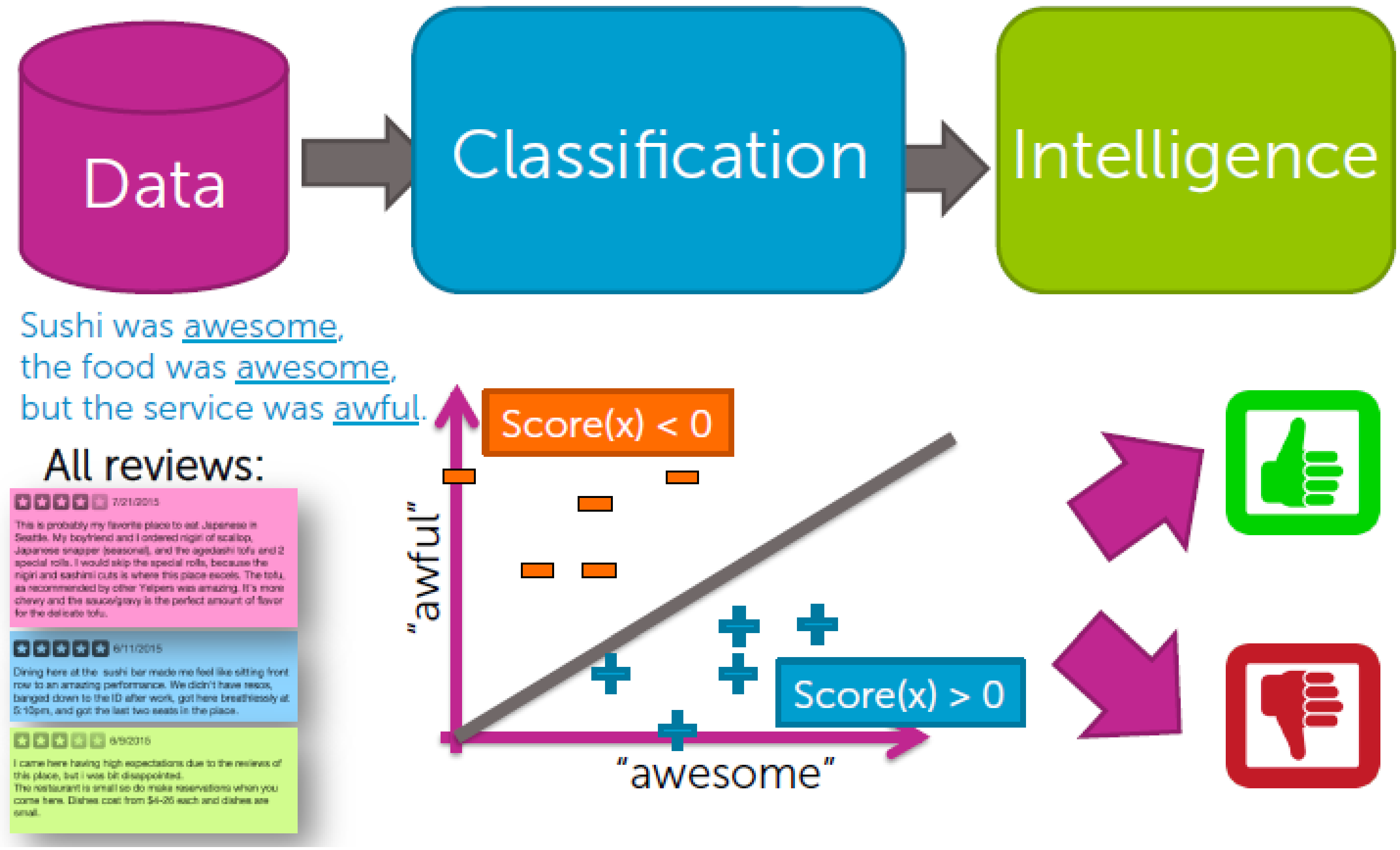
Market-Basket transactions

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Cold Drink, Eggs**
3	**Milk, Diaper, Cold Drink**
4	**Bread, Milk, Diaper, Cold Drink**
5	**Bread, Milk, Diaper, Water**

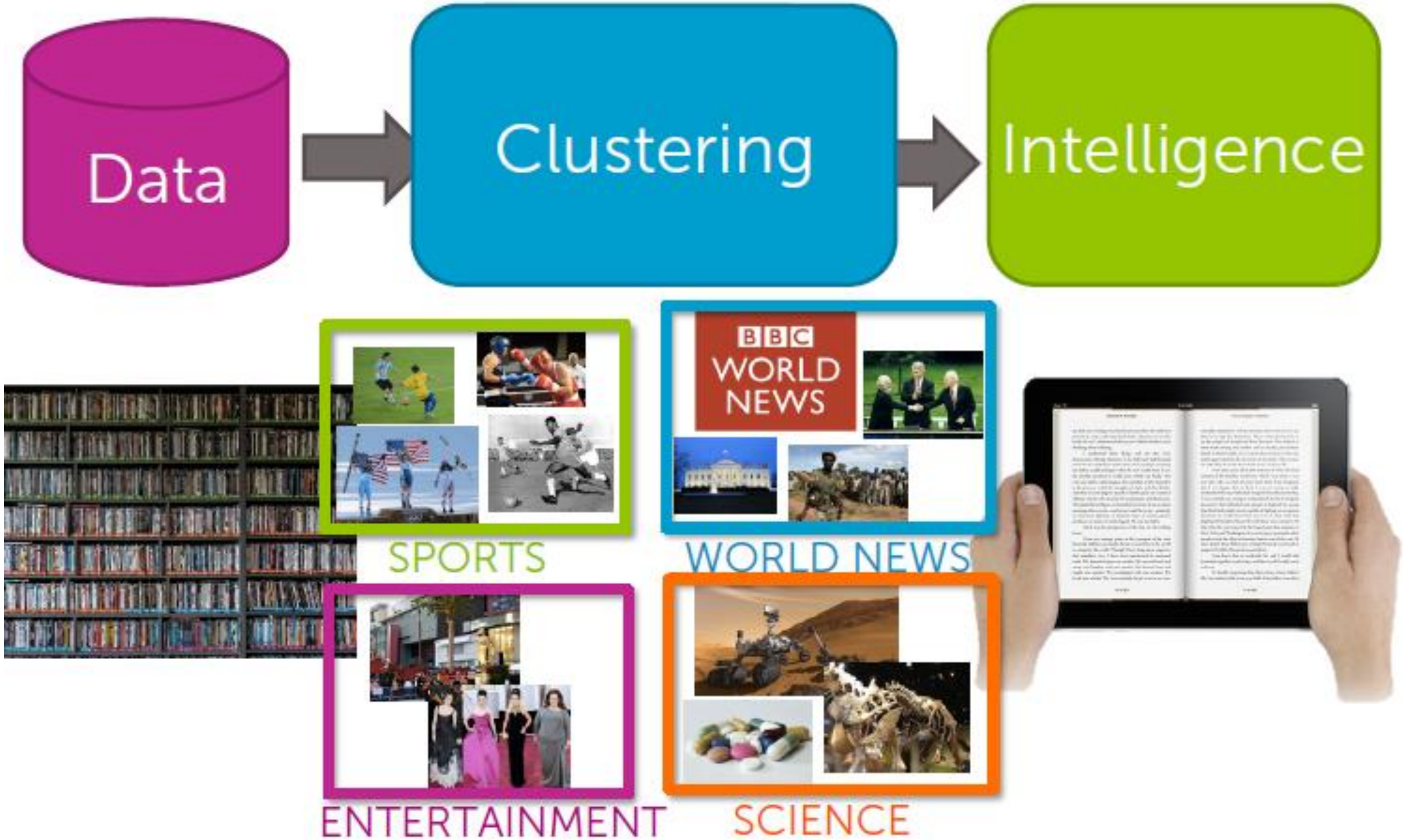
Predicting house prices



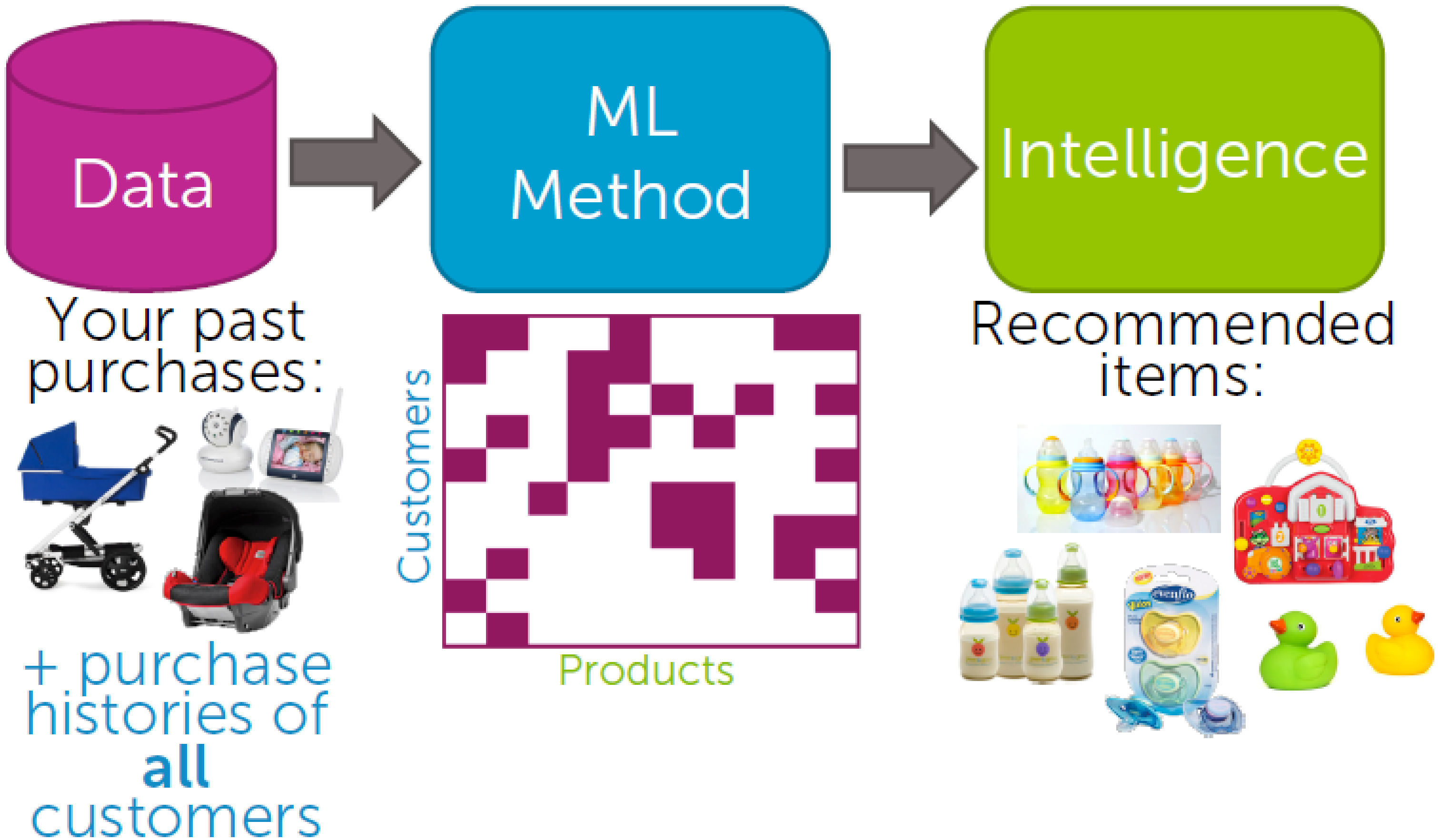
Sentiment analysis



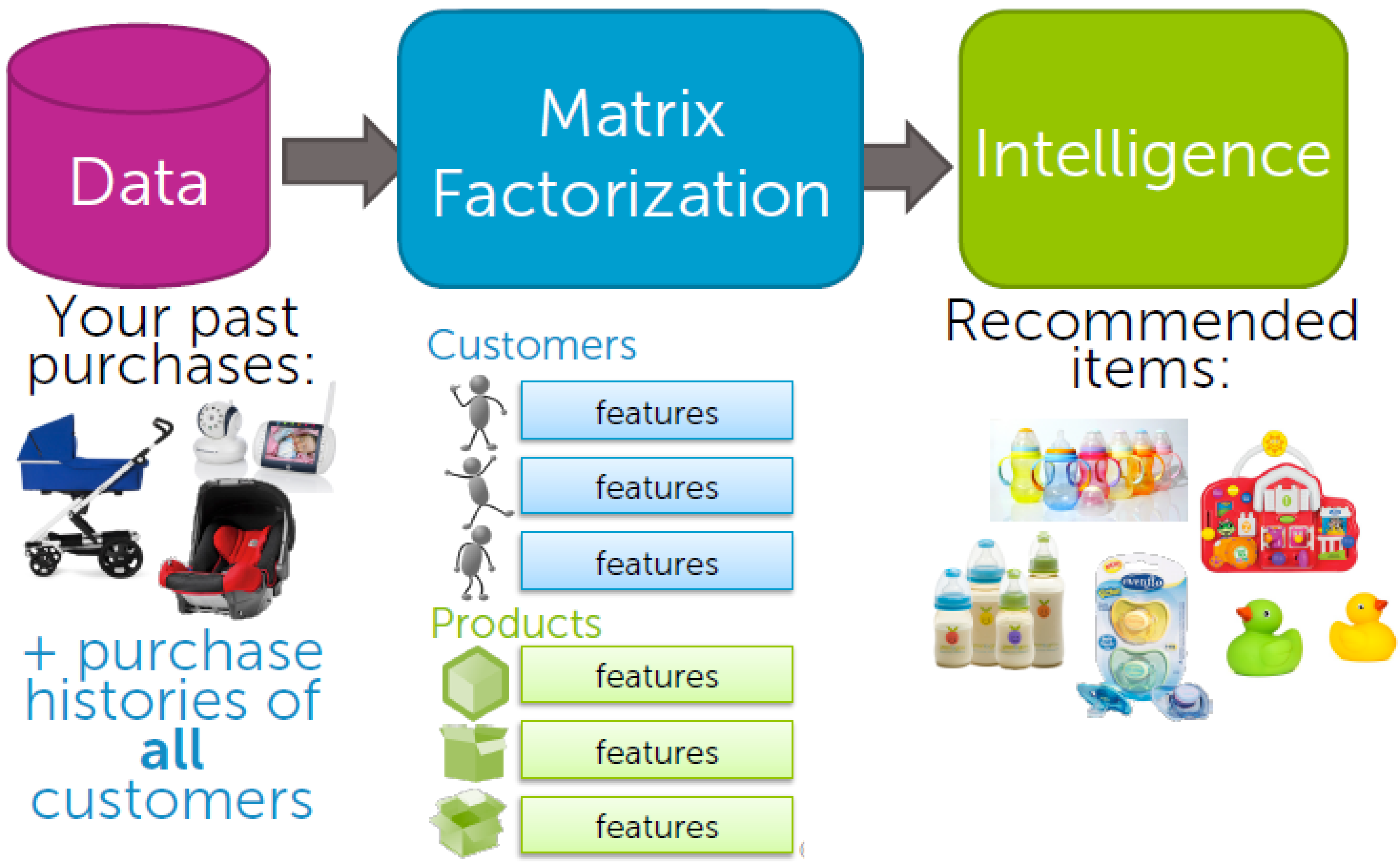
Document retrieval



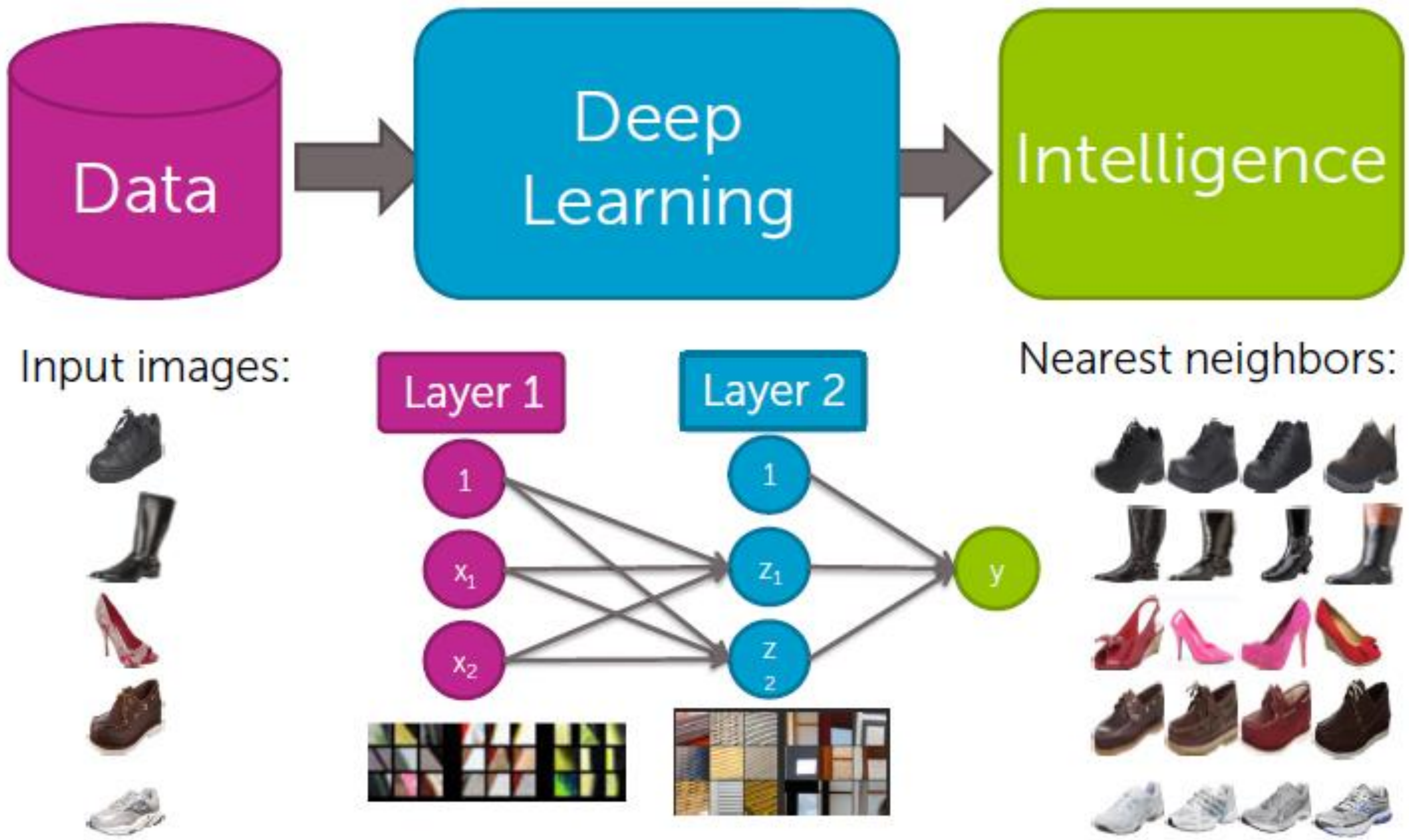
Product recommendation



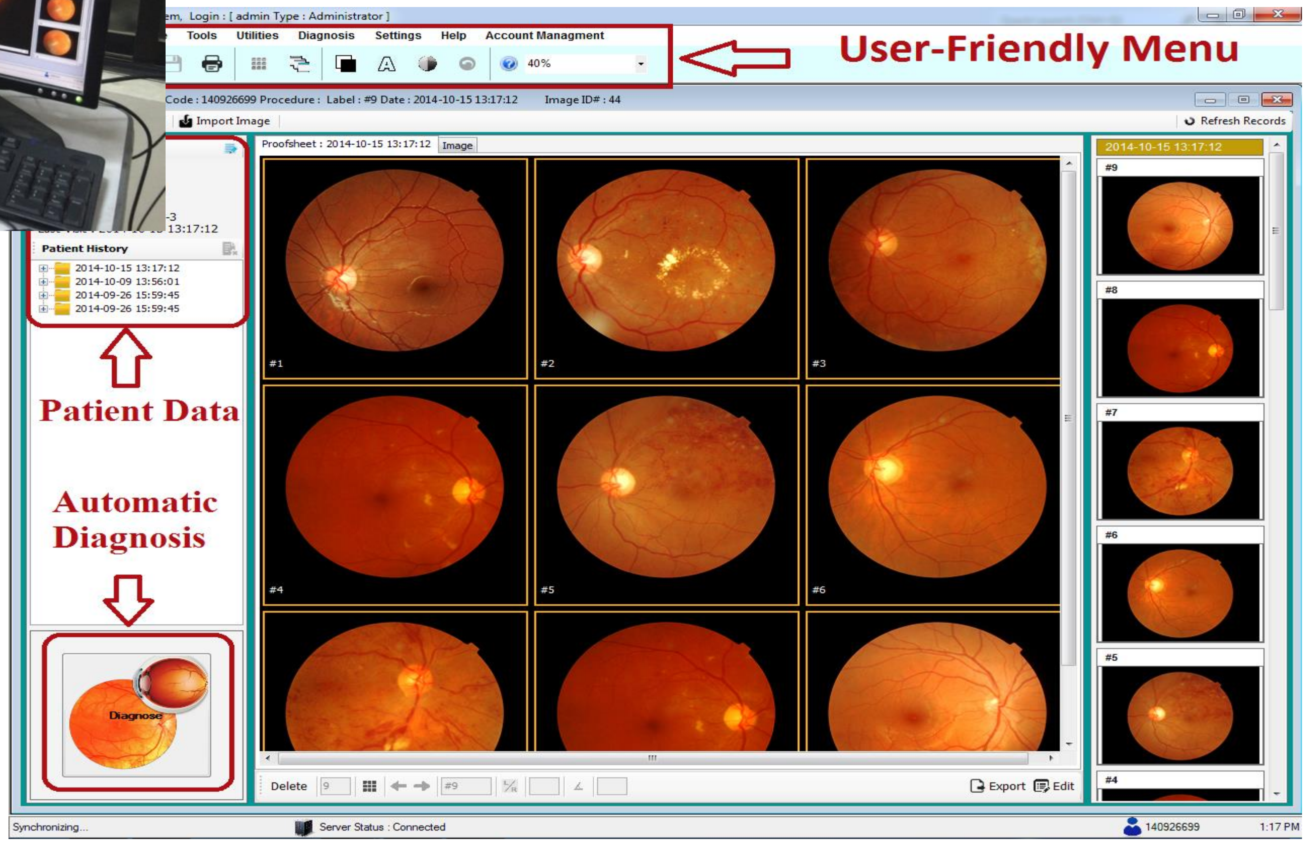
Product recommendation



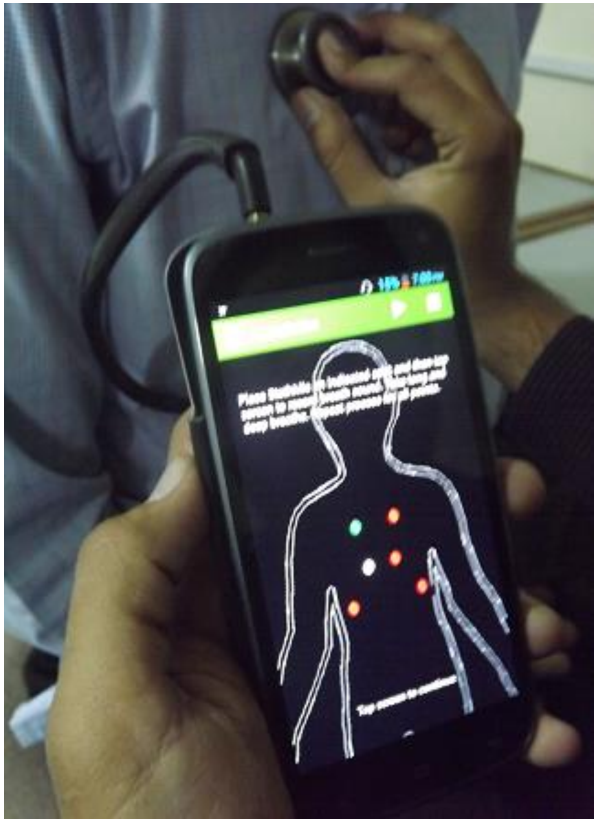
Visual Product recommender

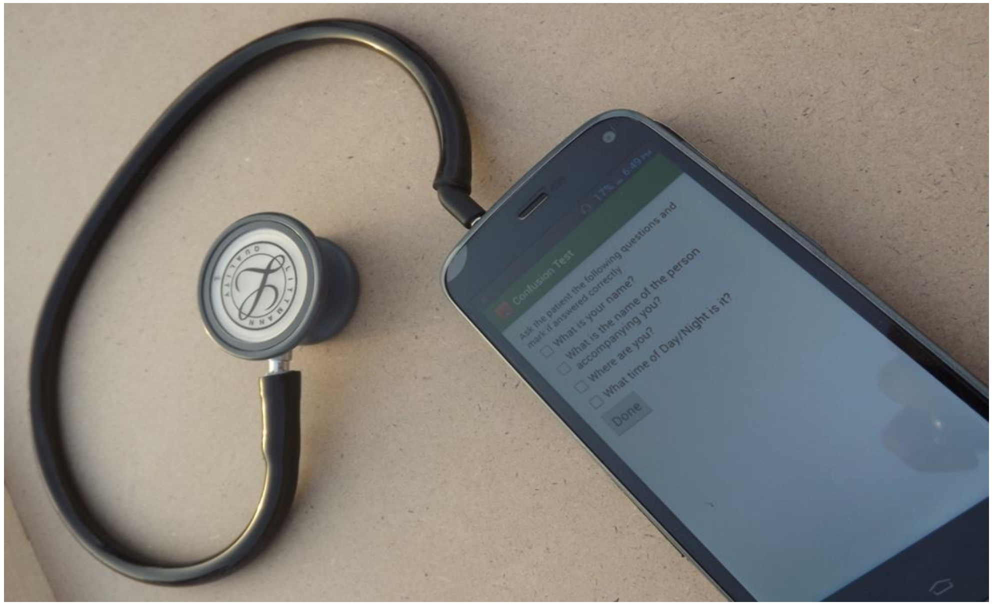


AL-BASR

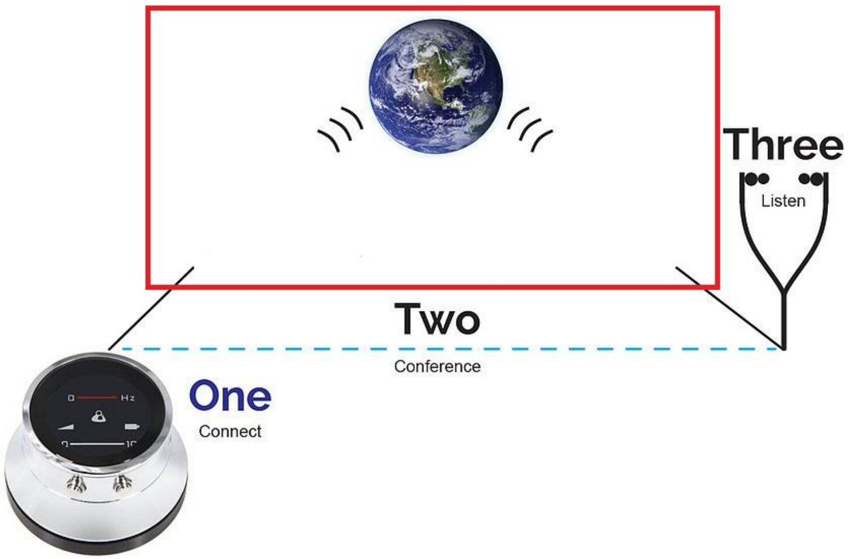

User-Friendly Menu
Patient Data
Automatic Diagnosis
Tools Utilities Diagnosis Settings Help Account Managment
Patient History
Diagnose
Server Status : Connected

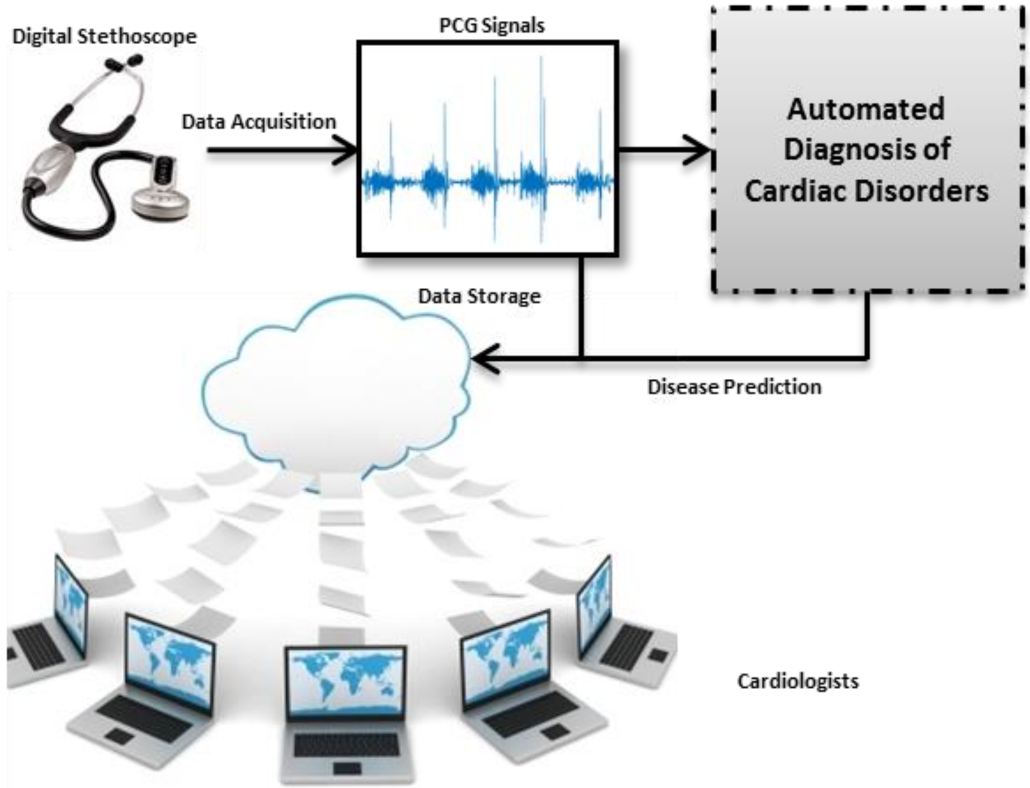
SmartSteth



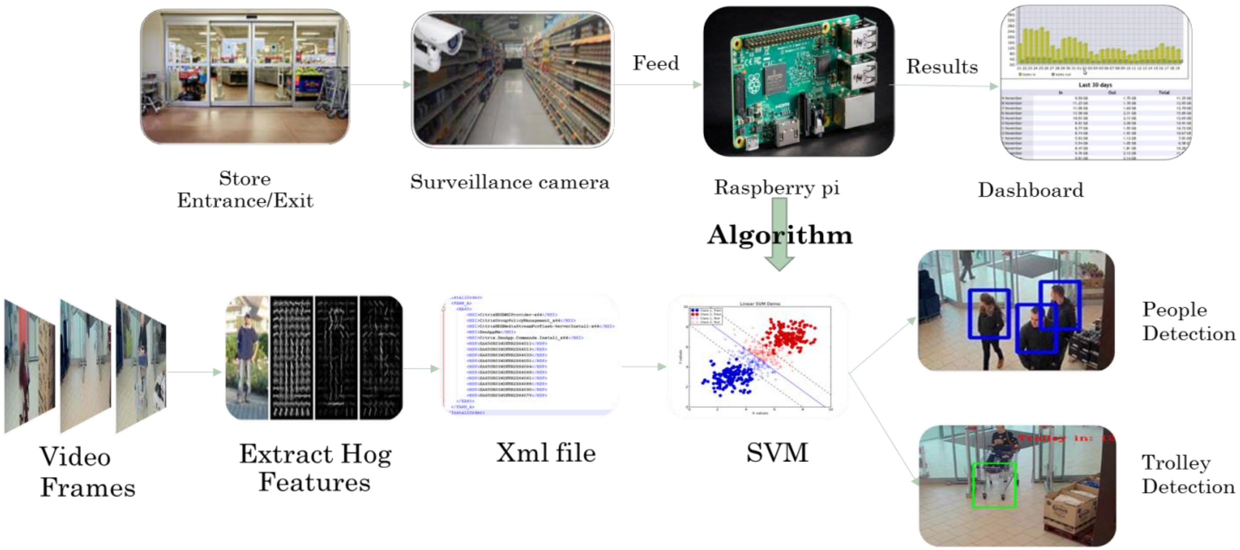

Confusion Test
Ask the patient the following questions and mark if answered correctly
What is your name?
What is the name of the person accompanying you?
Where are you?
What time of Day/Night is it?
Done

SmartSteth: Analysis of PCG Signals for Detection of Cardiac Disorders



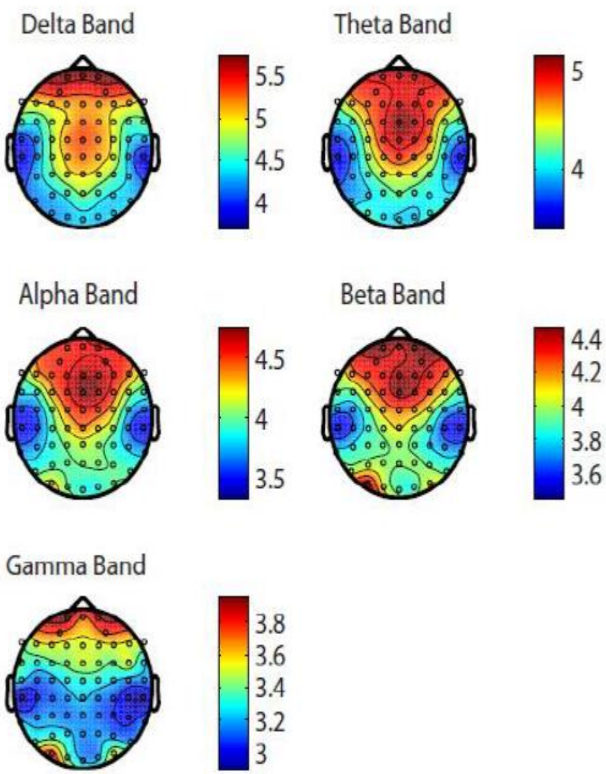


IVBS (Intelligent Video Based Solution for Retailers)

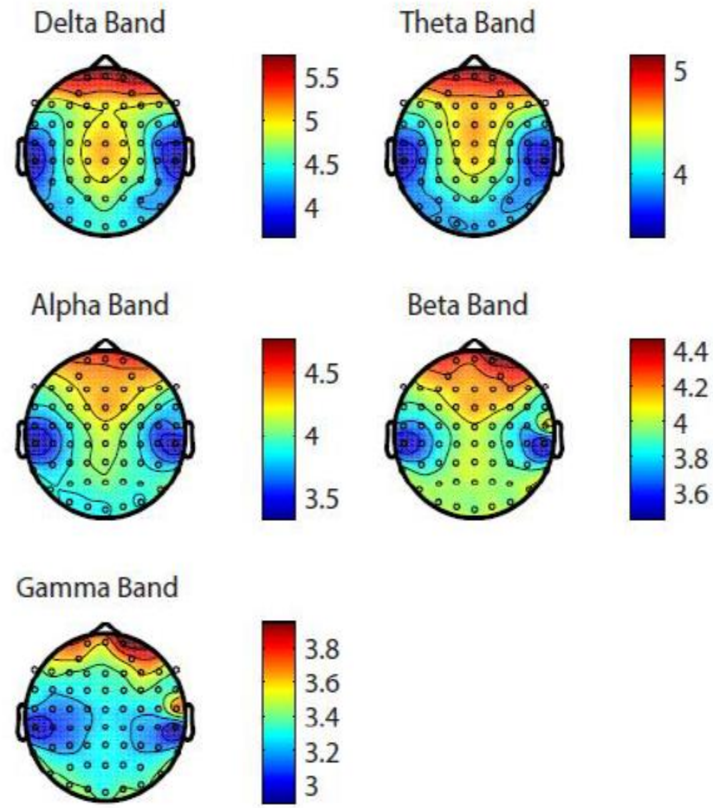


Analysis of EEG Signals for Diabetes Patients

Energy distribution for the healthy volunteers group (6 mmol/L), in 5 frequency bands



Energy distribution for the diabetic patients group (6 mmol/L), in 5 frequency bands



*"the effect of hyperglycemia measured by EEG in diabetic patients" by Giulia Berto, Rasmus Wilberg, Jakob Polciwartek

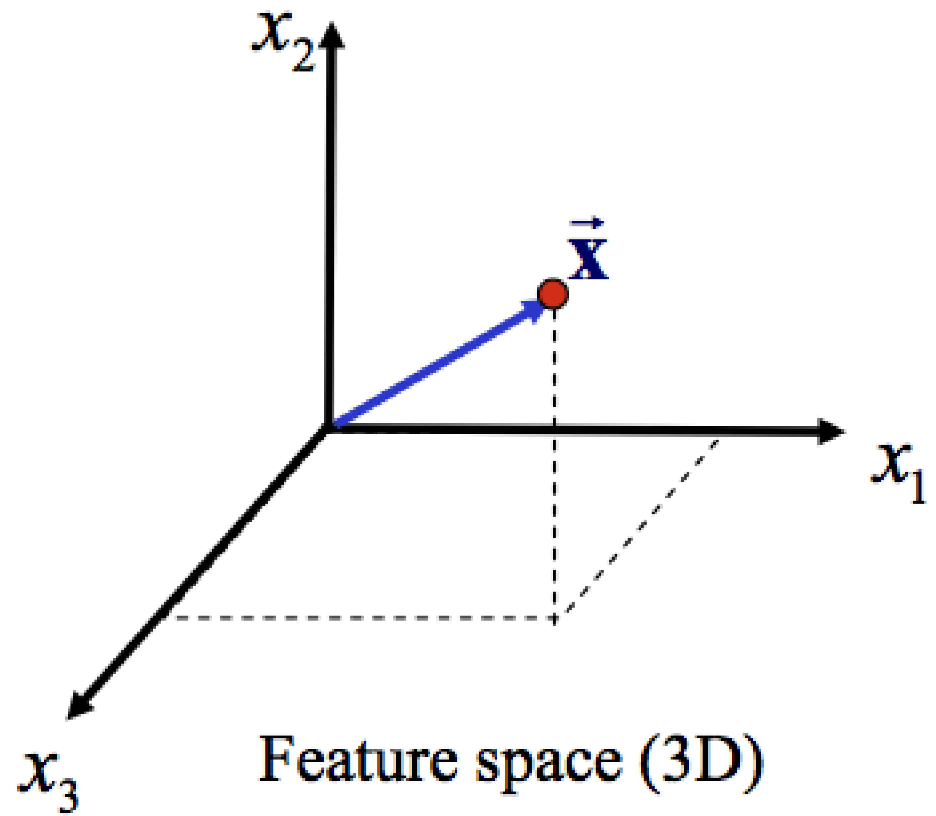
Model Choice

– What type of *classifier* shall we use? How shall we select its parameters? Is there best classifier...?
– How do we train...? How do we adjust the parameters of the model (*classifier*) we picked so that the model fits the data?

Features

- **Features**: a set of variables believed to carry discriminating and characterizing information about the objects under consideration
- **Feature vector**: A collection of d features, ordered in some meaningful way into a d- dimensional column vector, that represents the signature of the object to be identified.
- **Feature space**: The d-dimensional space in which the feature vectors lie. A d-dimensional vector in a d-dimensional space constitutes a point in that space.

Features



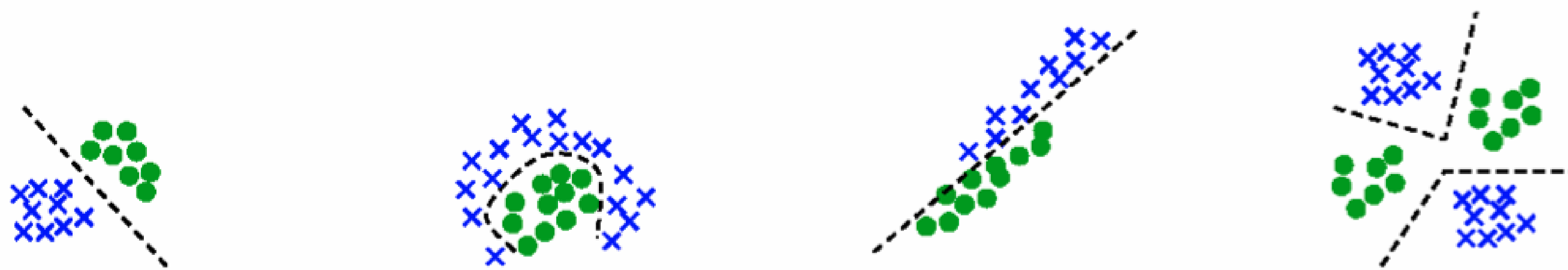
Feature space (3D)

Features

- Feature Choice
  - Good Features
    - Ideally, for a given group of patterns coming from the same class, feature values should all be similar
    - For patterns coming from different classes, the feature values should be different.
  - Bad Features
    - irrelevant, noisy, outlier?

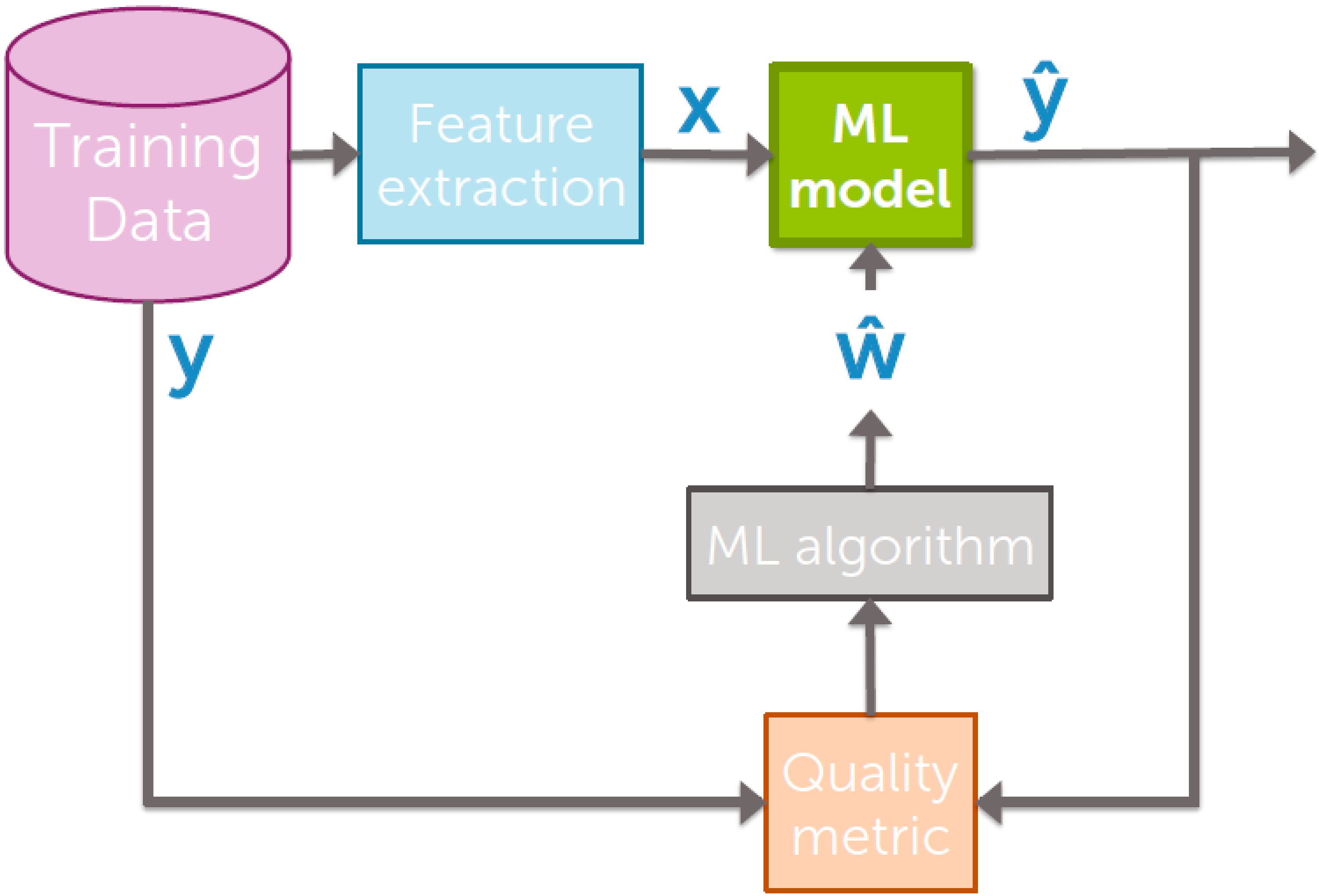
Features

“Good” features “Bad” features

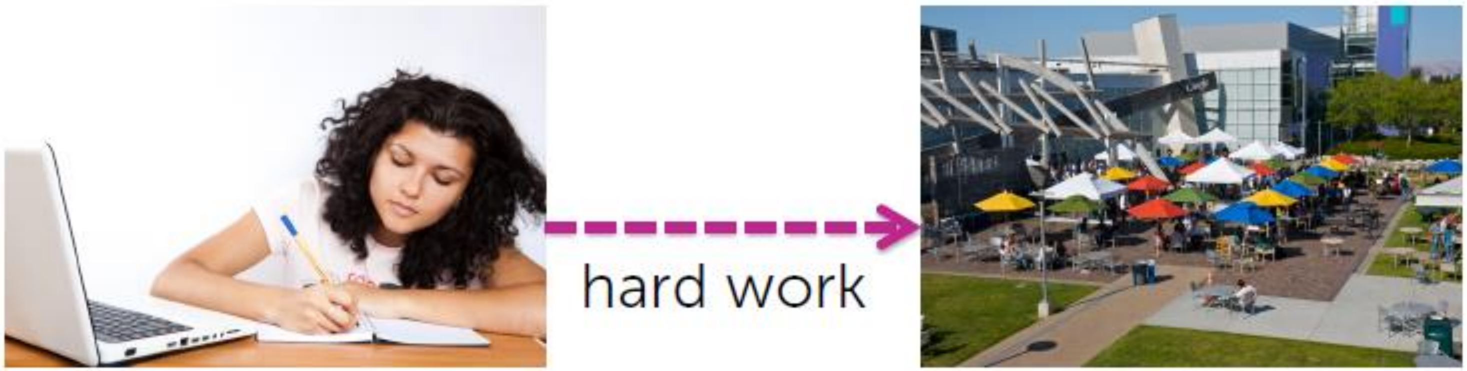


Linear separability Non-linear separability Highly correlated features Multi-modal

ML Overview


Training Data
Feature extraction
x
ML model
ŷ
y
ŵ
ML algorithm
Quality metric

Salary after ML specialization



- How much will your salary be? (y=$$)
- Depend on x=performance in courses, quality of capstone project, # of forum responses,….

Salary after ML specialization

hard work

$$\hat{y} = \hat{w}_0 + \hat{w}_1 \, \text{performance} + \hat{w}_2 \, \text{capstone} + \hat{w}_3 \, \text{forum}$$

Informed by other students who completed specialization

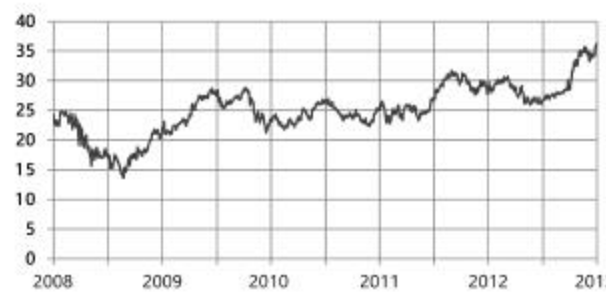
Stock Prediction

- Predict the price of a stock
- Depends on

-Recent history of stock price

-News events

-Related commodities



Tweet popularity

How many people will retweet your tweet?

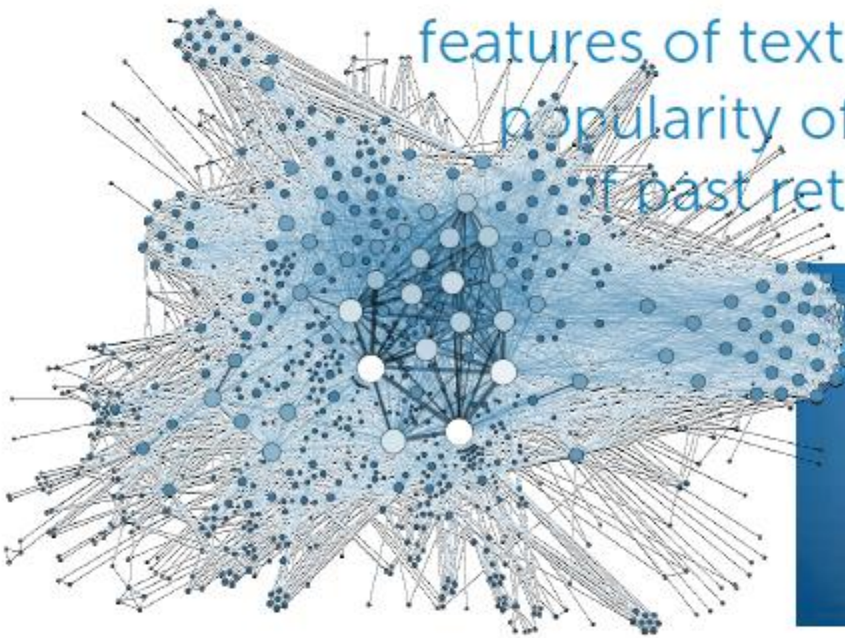
Depends on #followers,

#of followers of followers,

features of text tweeted,

popularity of hashtag,

f past retweets,...



Smart houses

- Smart houses have many disturbed sensors
- What's the temperature at your desk? (no sensor)
- -Learn spatial function to predict temp
- Also depends on

-Thermostat setting

-Blinds open/closed or

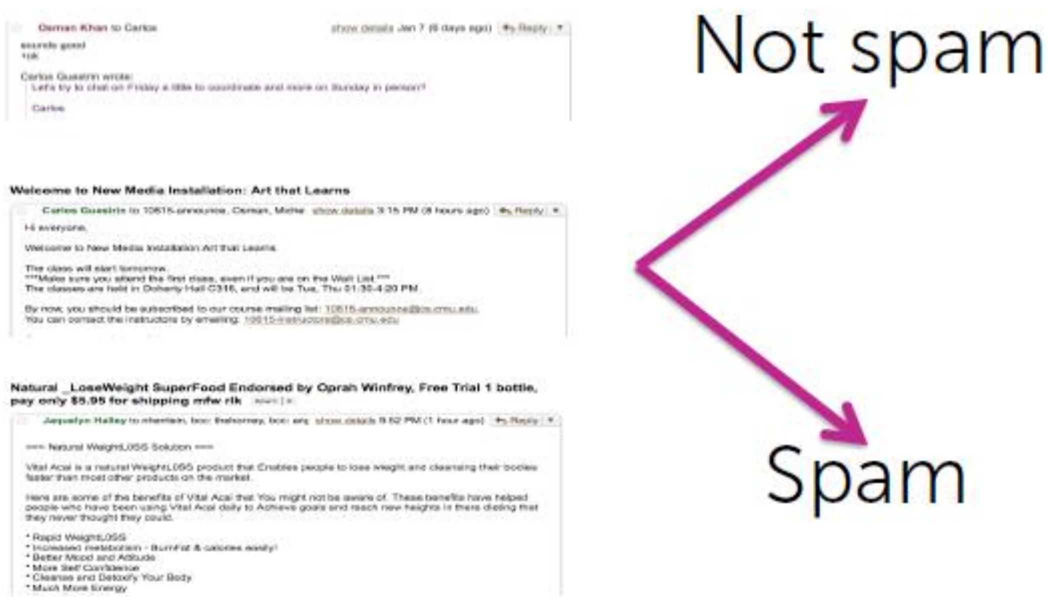
window tint

-Vents

-Temperature outside

-Time of day

Spam filtering

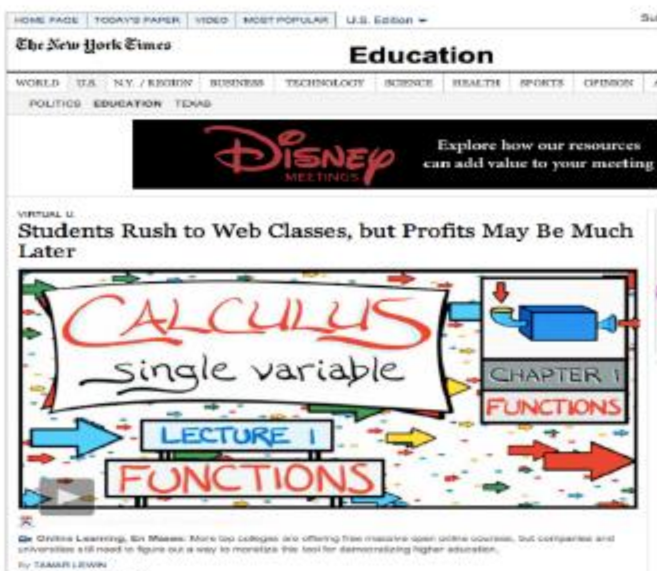


Input: x

Output: y

Example multiclass classifier

Output y has more than 2 categories



Education

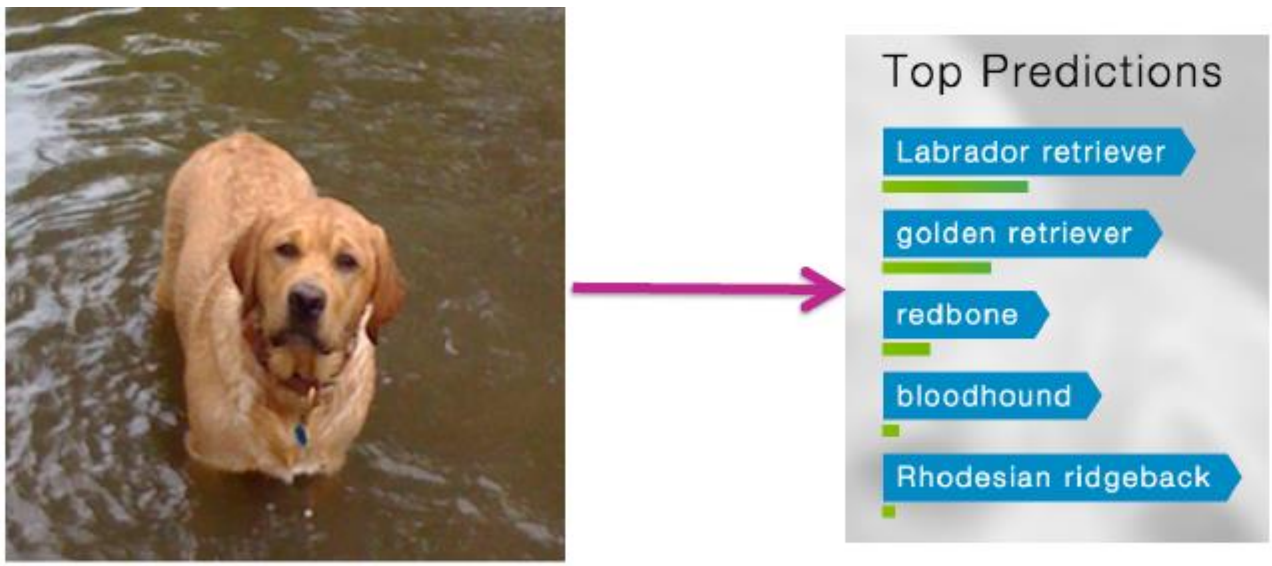
Finance

Technology

Input: x
webpage

Output: y

Image classification



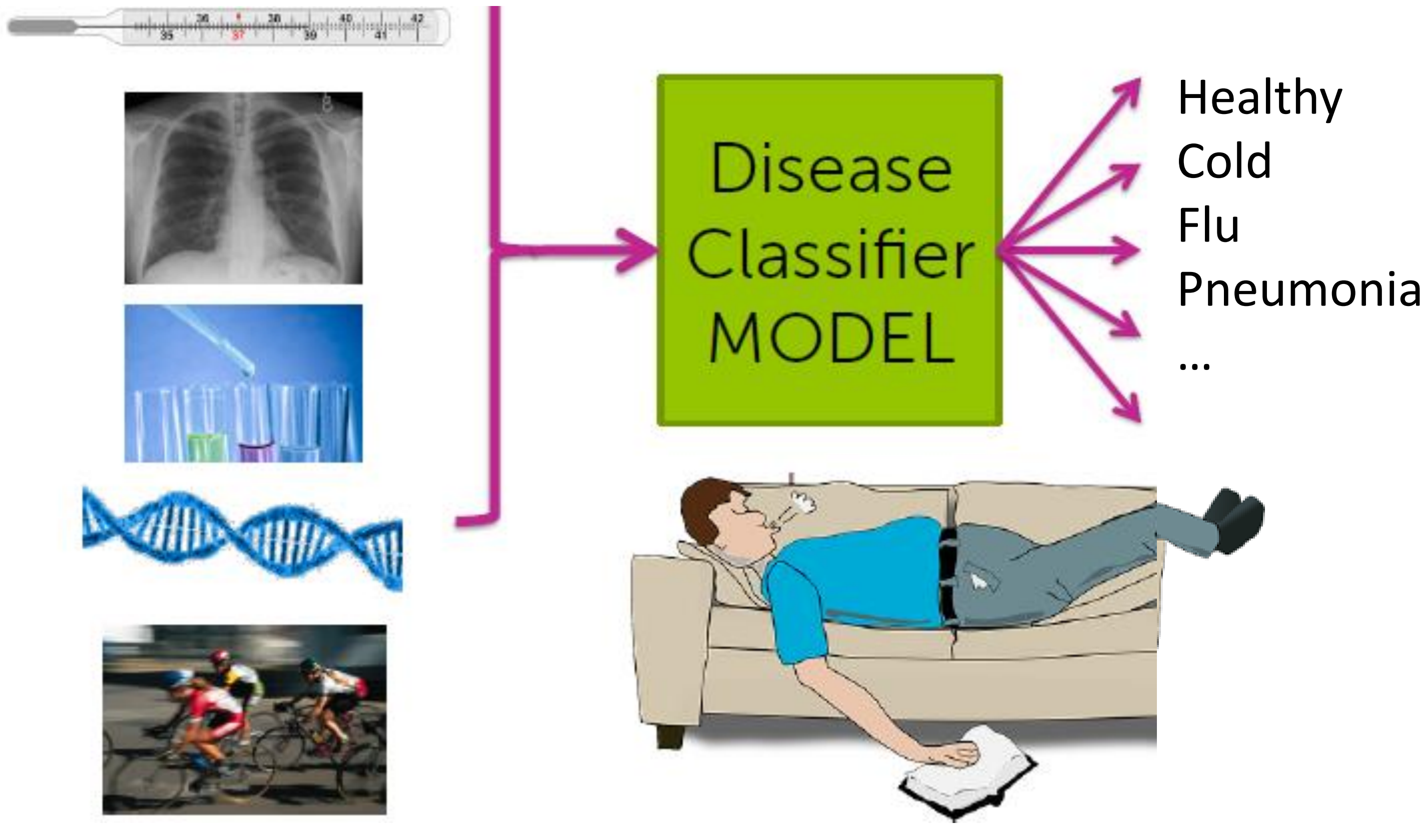
Input: x
Image pixels

Output: y
Predicted object

Personalized medical diagnosis

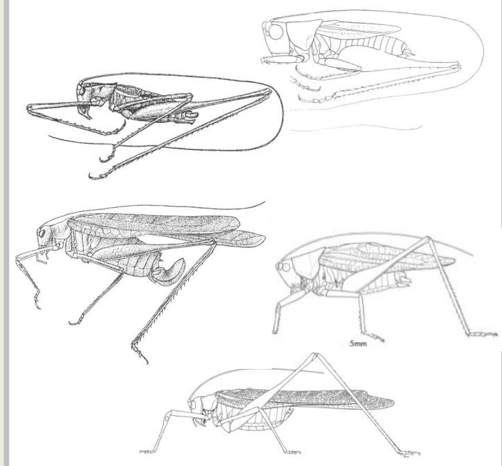
Input: x

Output: y

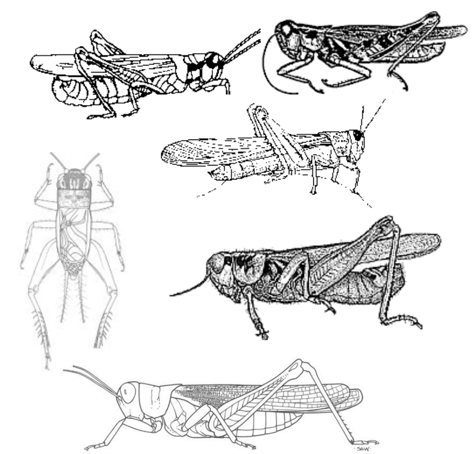


A Classification Problem Example

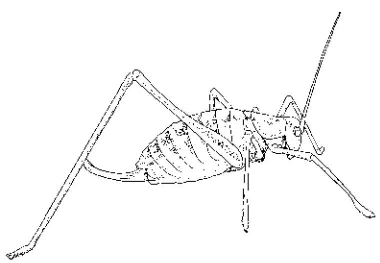
Katydids



Grasshoppers

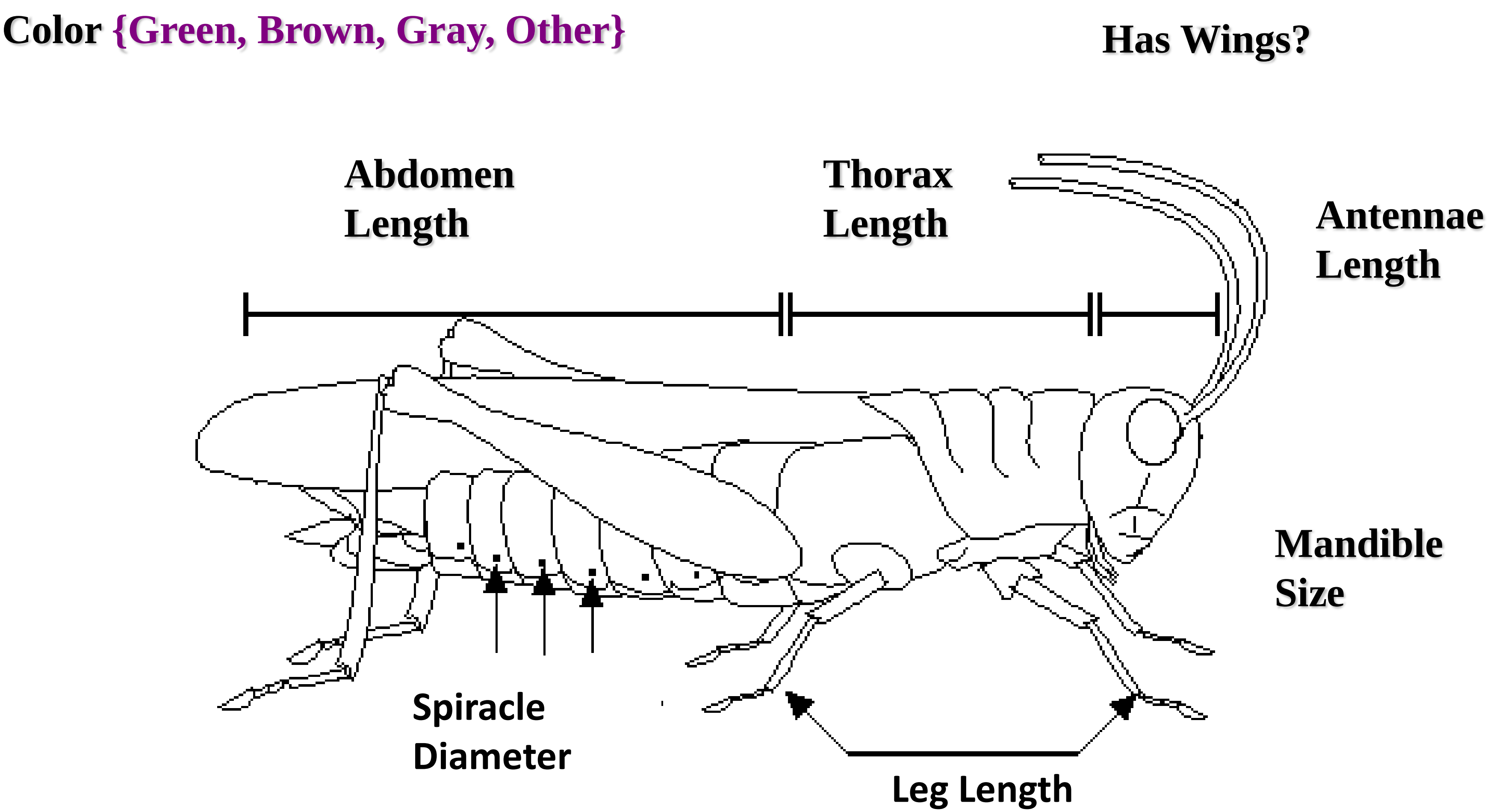


Given a collection of annotated data. In this case 5 instances of **Katydids** and five of **Grasshoppers**, decide what type of insect the unlabeled example is.



Katydid or **Grasshopper?**

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



We can store features in a database.

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**

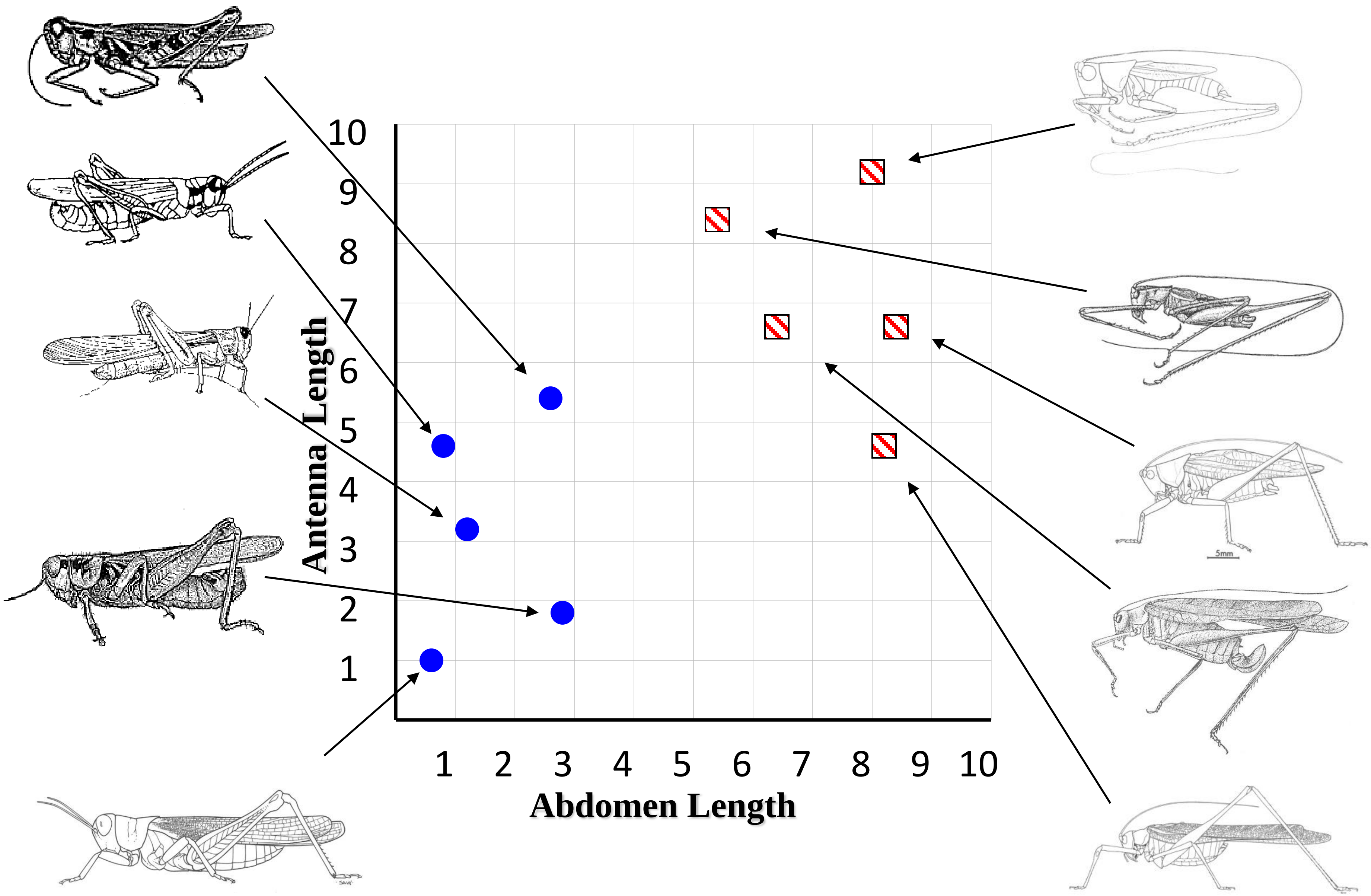
The classification problem can now be expressed as:

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

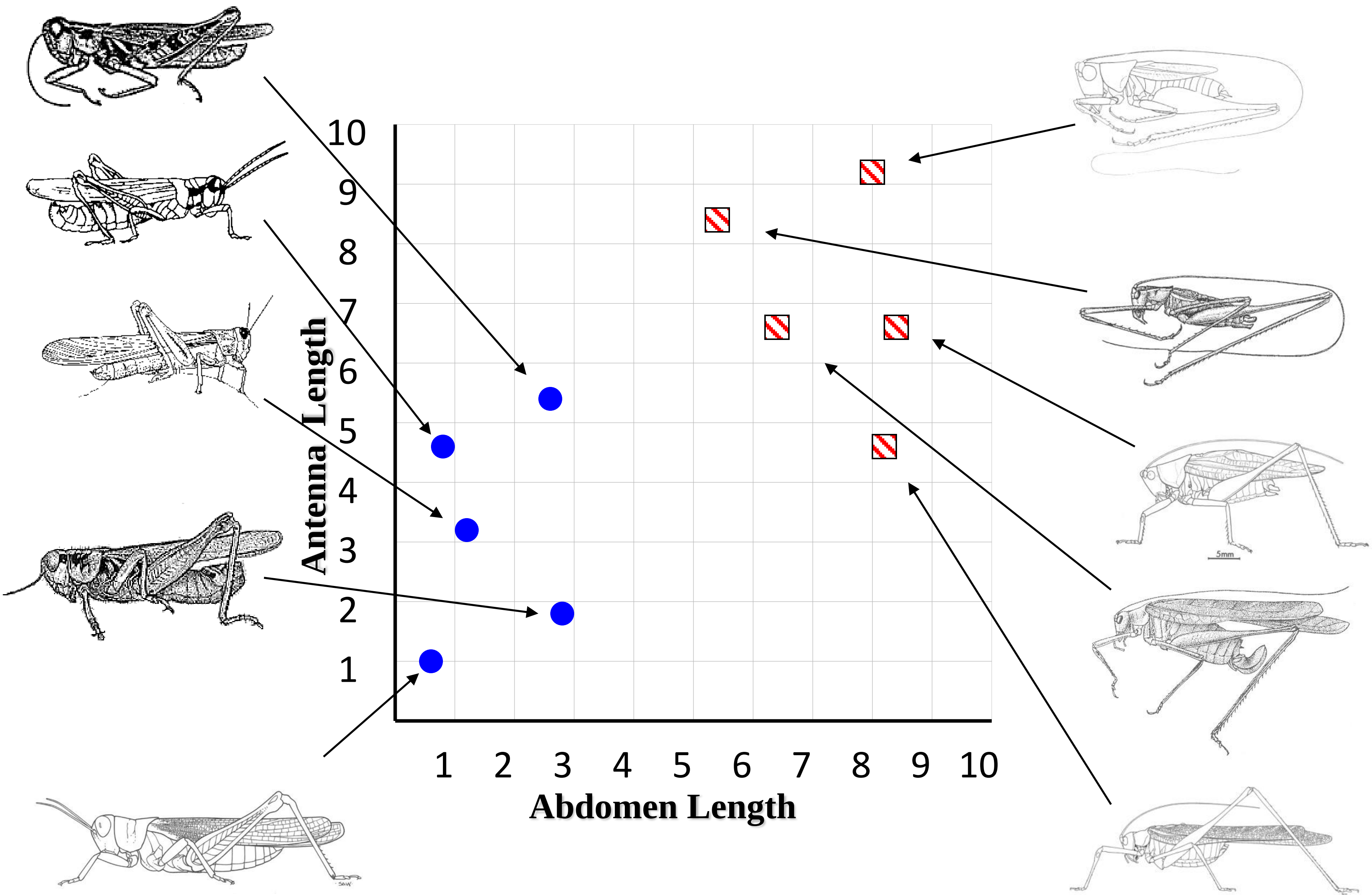
previously unseen instance =

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

Grasshoppers **Katydids**

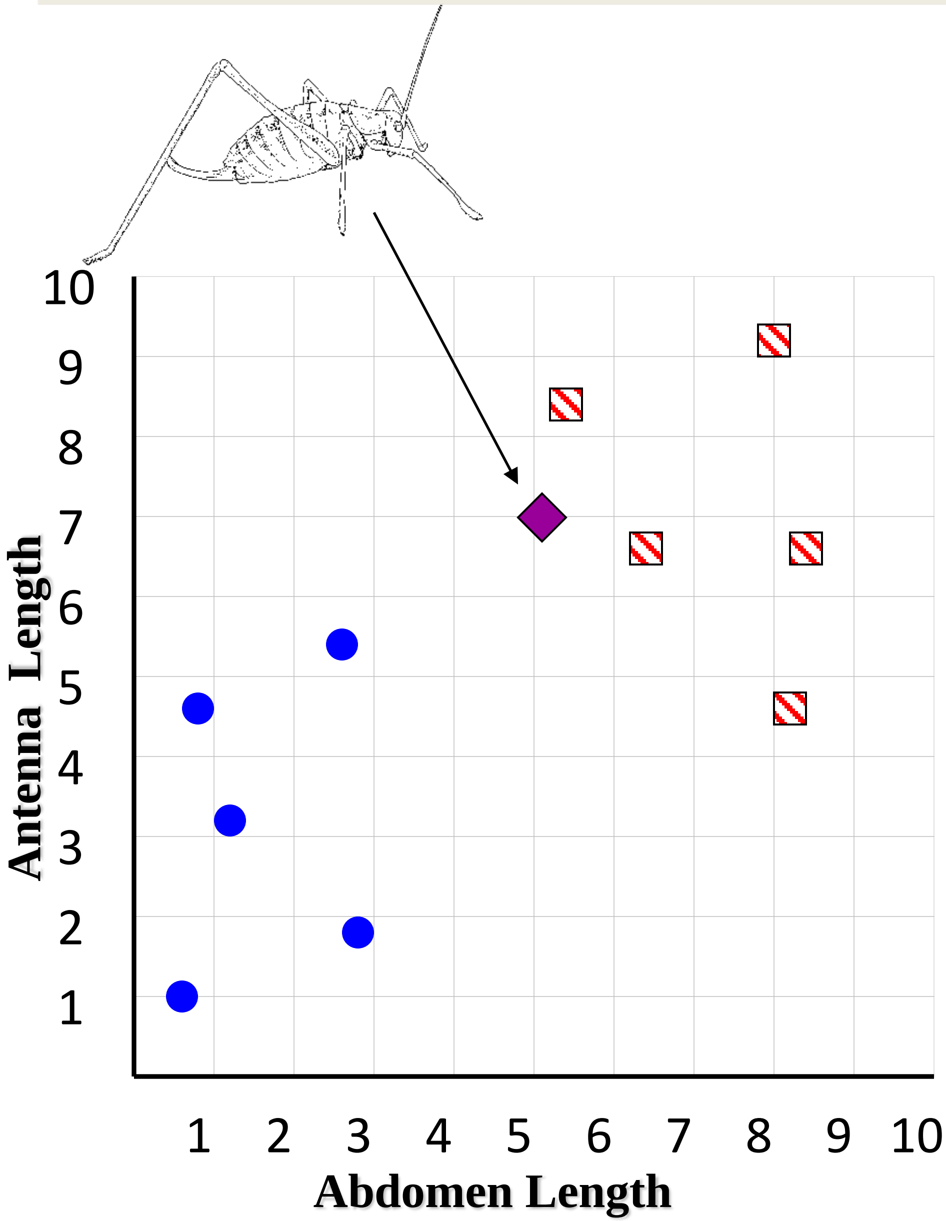


Grasshoppers
Katydids


Antenna Length
10
9
8
7
6
5
4
3
2
1
1 2 3 4 5 6 7 8 9 10
Abdomen Length
5mm

previously unseen instance =

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

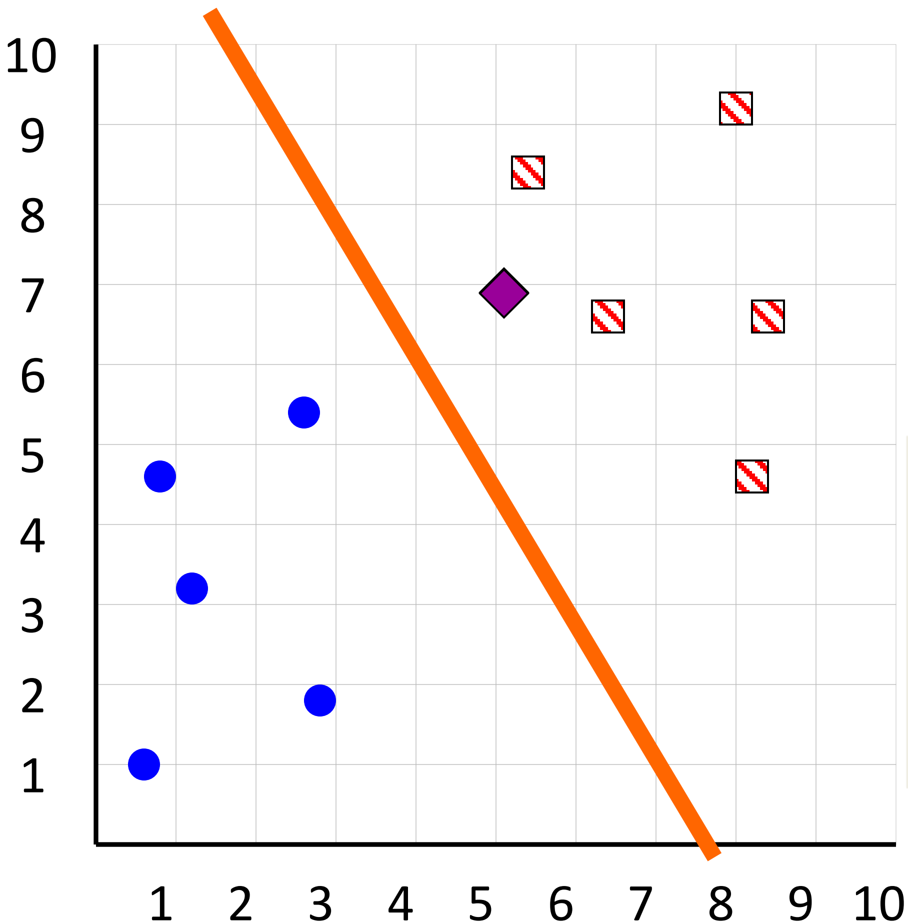


- We can "project" the previously unseen instance into the same space as the database.

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

Katydids
Grasshoppers

Simple Linear Classifier



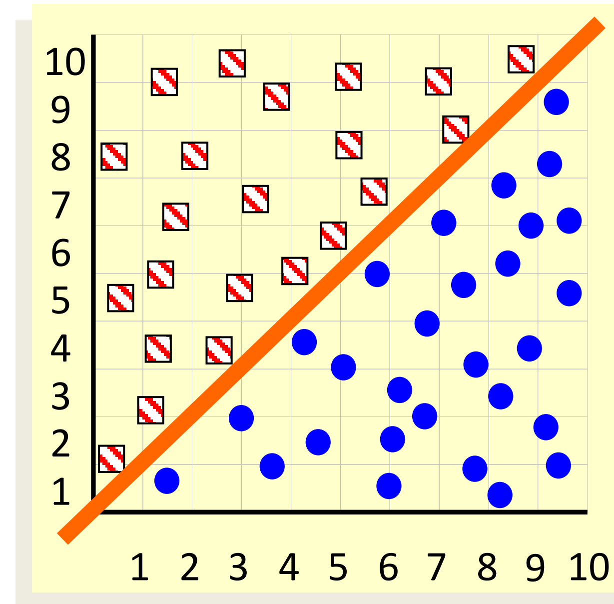
R.A. Fisher
1890-1962

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

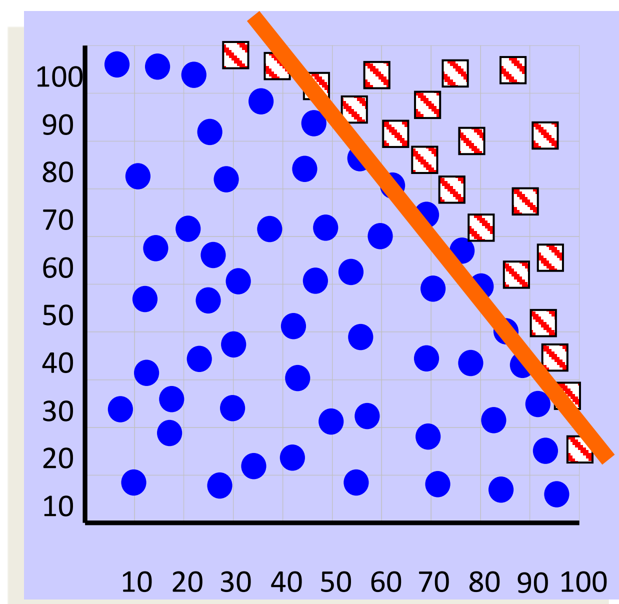
Katydids
Grasshoppers

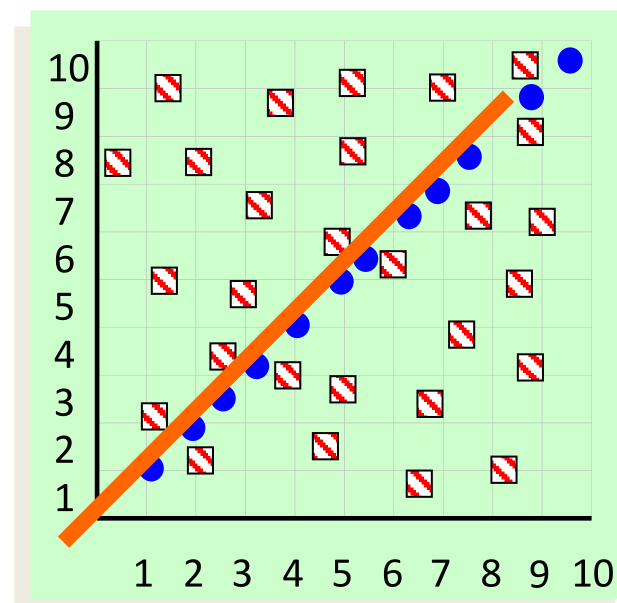
Which of the “Pigeon Problems” can be solved by the Simple Linear Classifier?

1) Perfect
2) Useless
3) Pretty Good

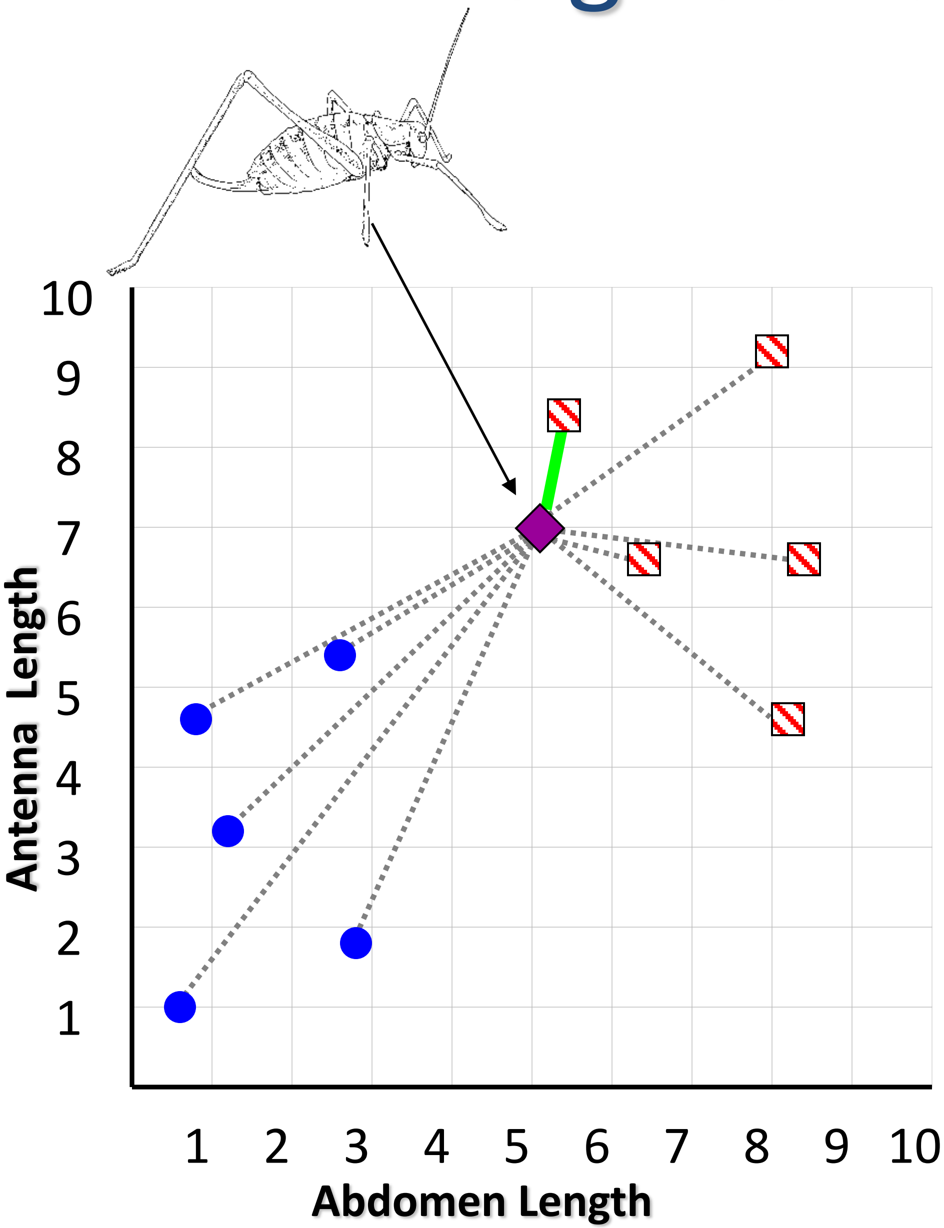


Problems that can be solved by a linear classifier are call **linearly separable**.





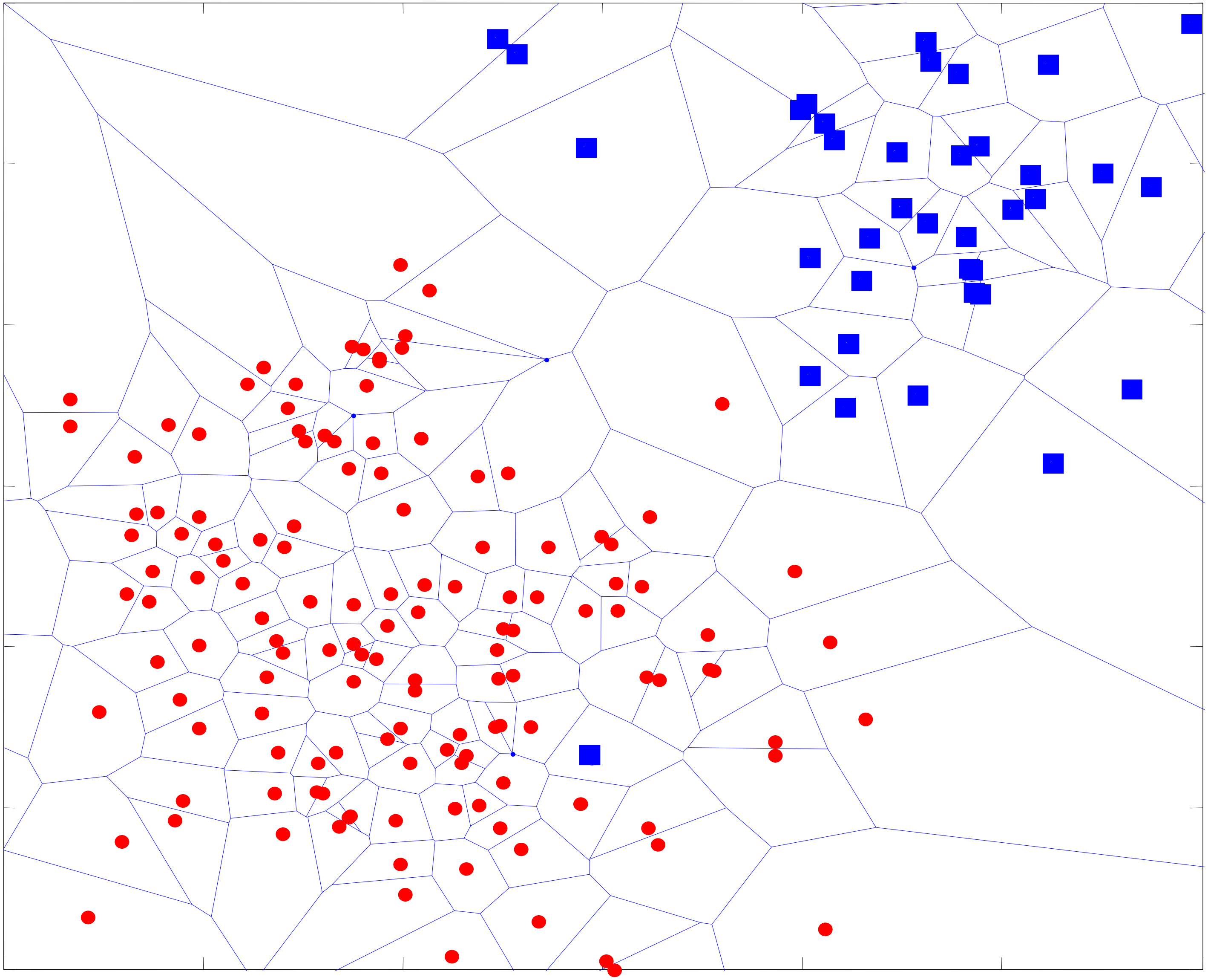
Nearest Neighbor Classifier



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

Katydids
Grasshoppers

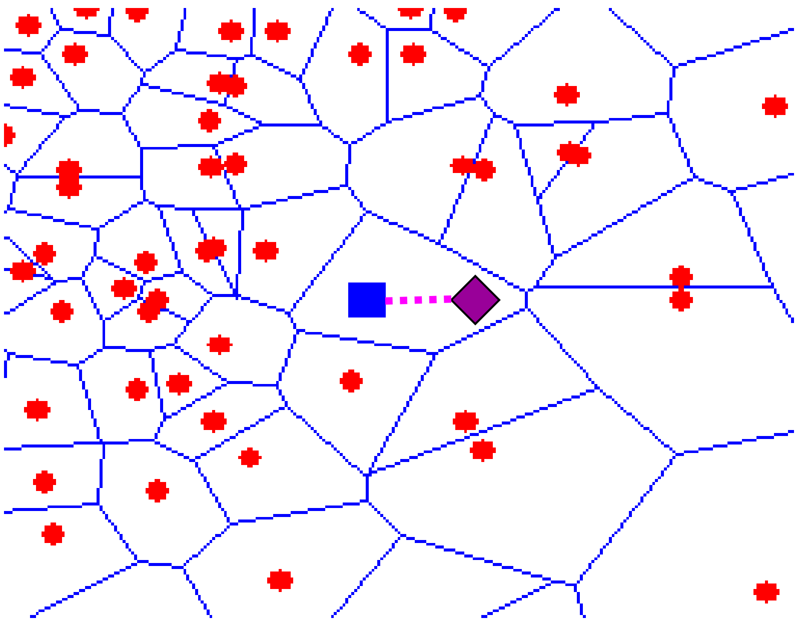
The nearest neighbor algorithm is sensitive to outliers…



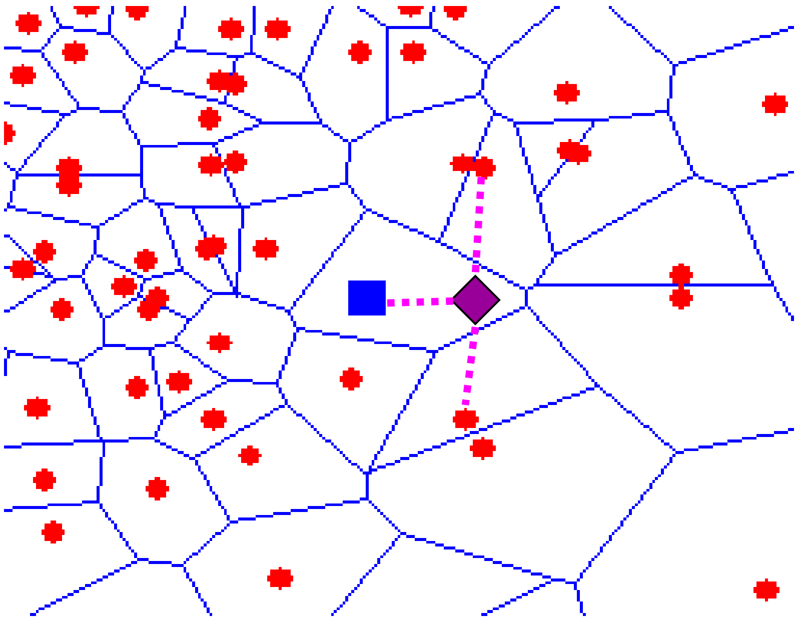
The solution is to…

We can generalize the nearest neighbor algorithm to the K- nearest neighbor (KNN) algorithm.

We measure the distance to the nearest K instances, and let them vote. K is typically chosen to be an odd number.



K = 1



K = 3

The K-Nearest Neighbour Algorithm

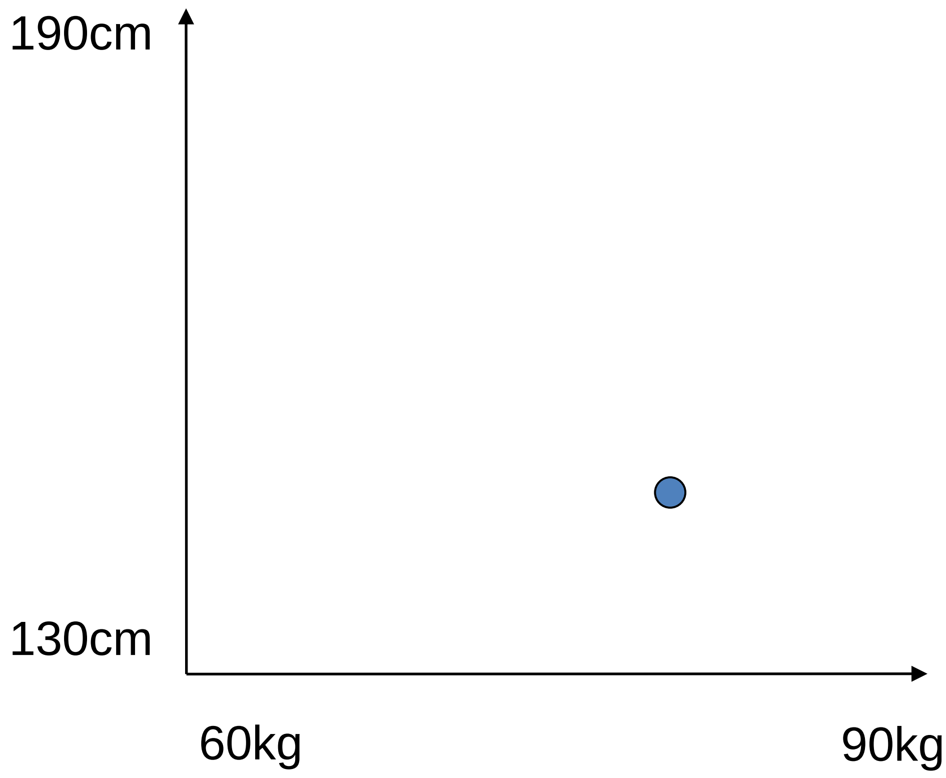
```
for each testing point
   measure distance to every training point
   find the k closest points
   identify the most common class among those k
   predict that class
end
```

- Advantage: Surprisingly good classifier!
- Disadvantage: Have to store the entire training set in memory

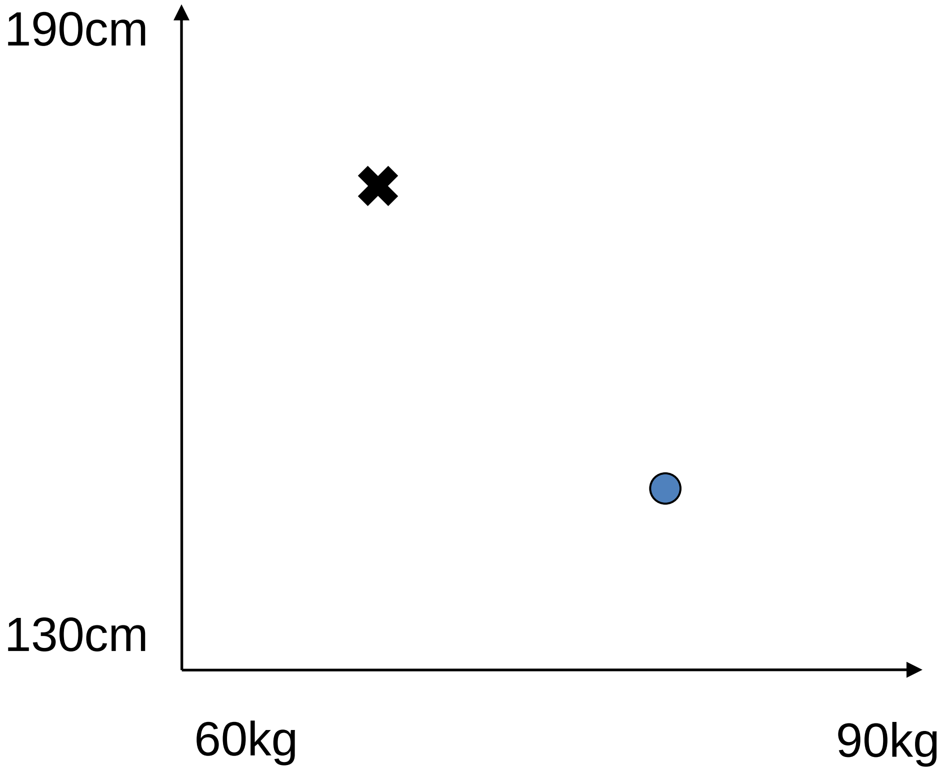
Can we LEARN to recognise a rugby player?

What are the “features” of a rugby player?

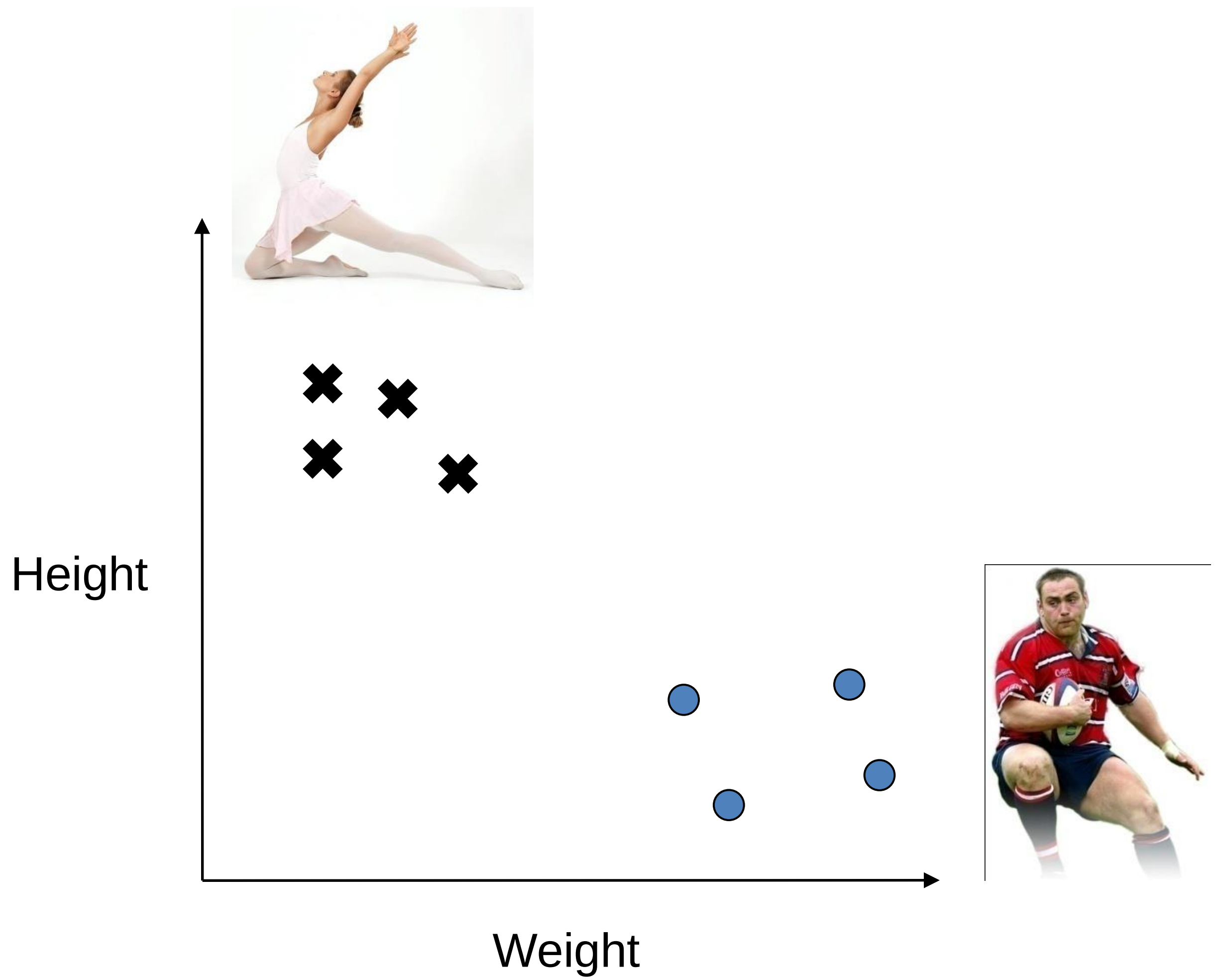
Rugby players = short + heavy?



Ballet dancers = tall + skinny?

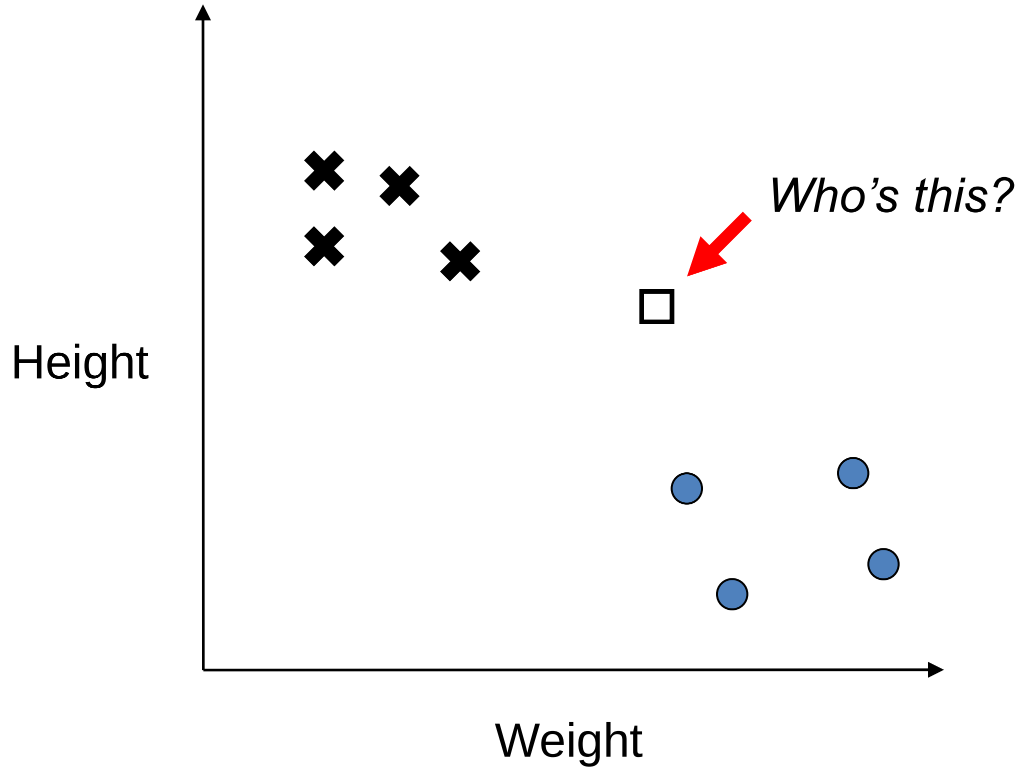


Rugby players "cluster" separately in the space.



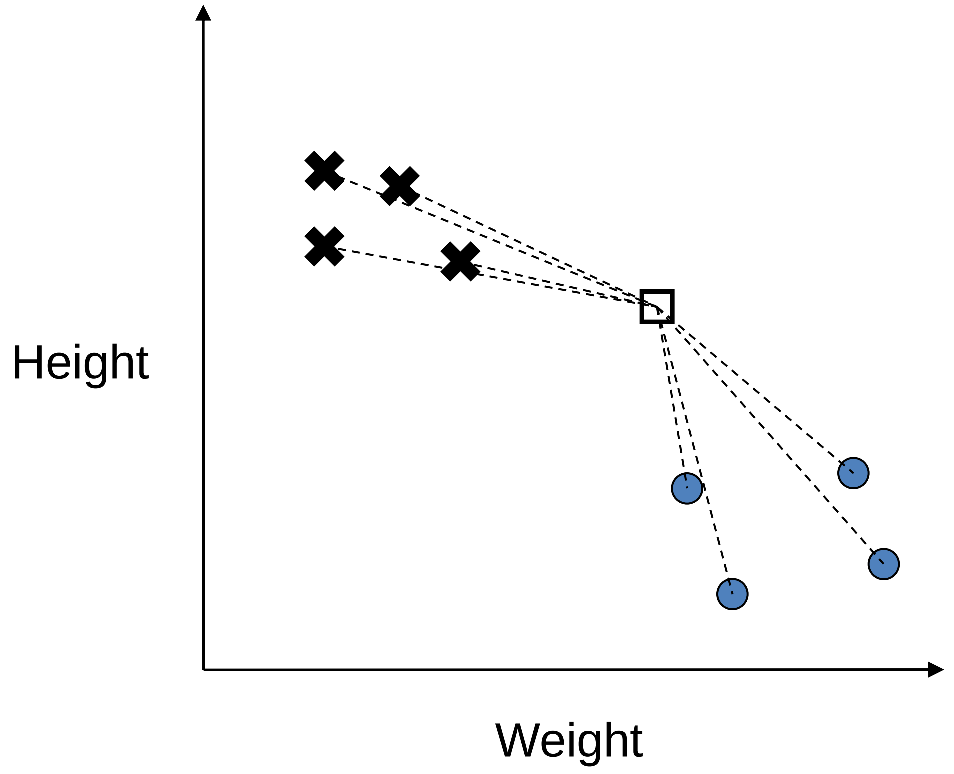
K Nearest Neighbors

The K-Nearest Neighbour Algorithm



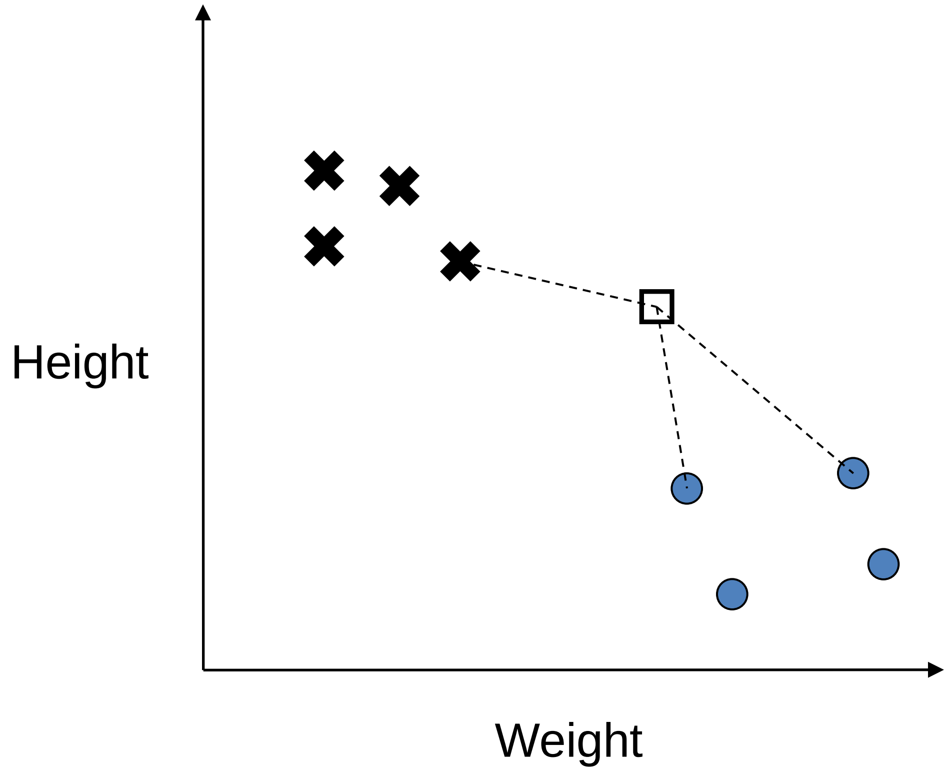
The K-Nearest Neighbour Algorithm

1. *Measure distance to all points*



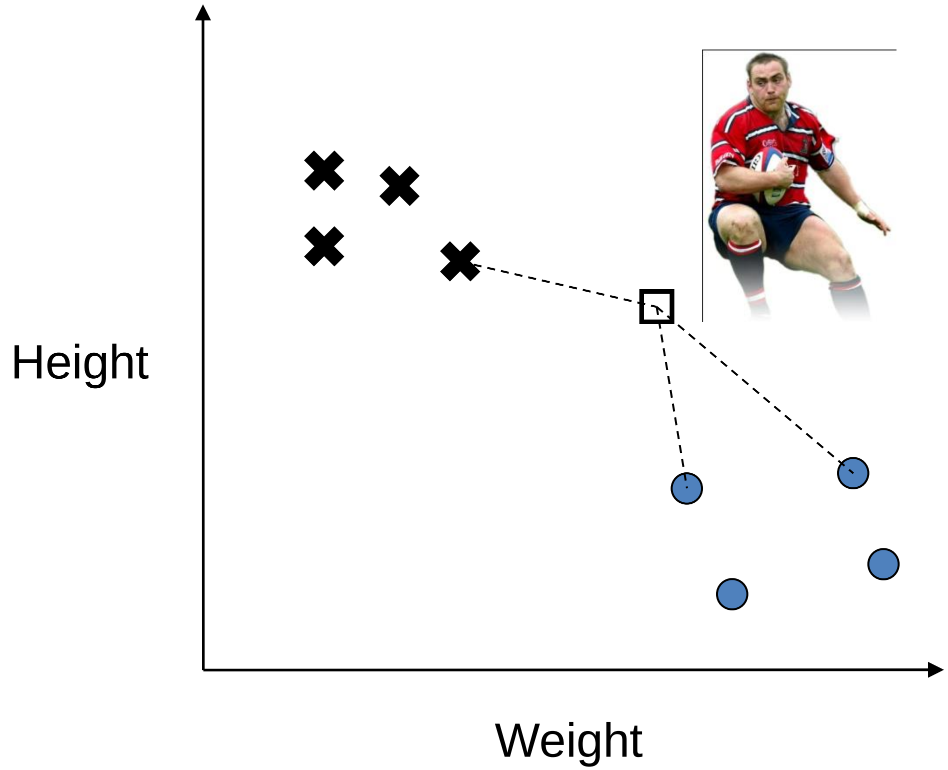
The K-Nearest Neighbour Algorithm

1. *Measure distance to all points*
2. *Find closest "k" points* **← (here k=3, but it could be more)**



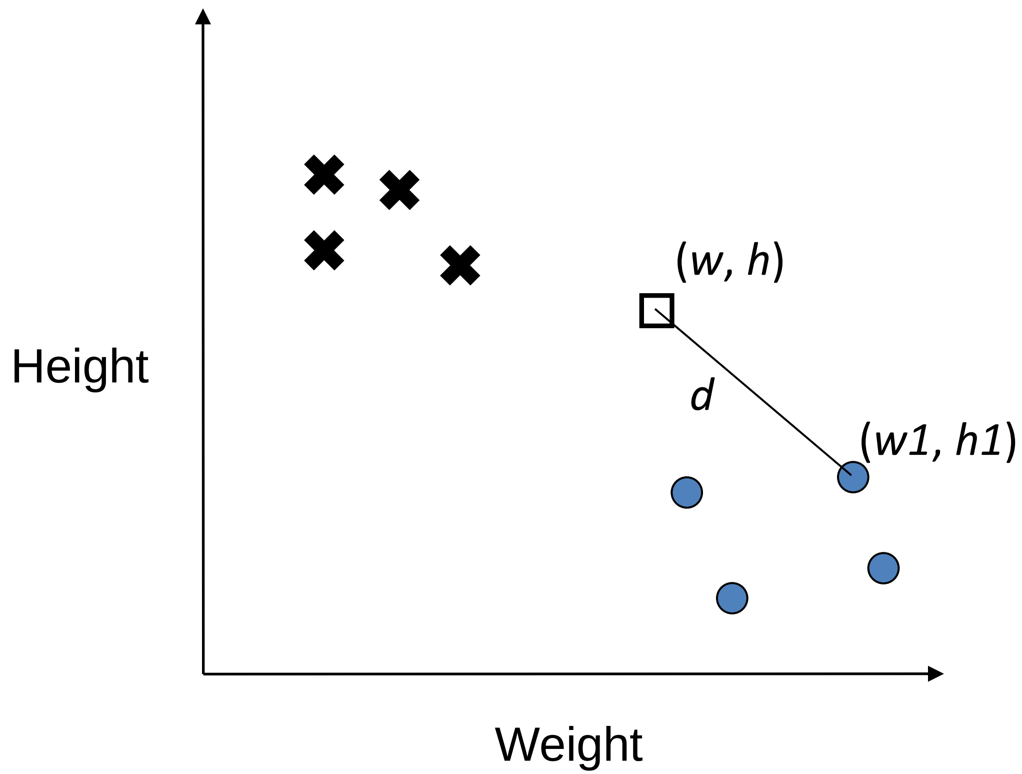
The K-Nearest Neighbour Algorithm

1. *Measure distance to all points*
2. *Find closest "k" points* ← **(here k=3, but it could be more)**
3. *Assign majority class*



“Euclidean distance”

$$d = \sqrt{(w - w_1)^2 + (h - h_1)^2}$$



The K-Nearest Neighbour Algorithm

```
for each testing point
    measure distance to every training point
    find the k closest points
    identify the most common class among those k
    predict that class
end
```

- Advantage: Surprisingly good classifier!
- Disadvantage: Have to store the entire training set in memory

Euclidean distance still works in 3-d, 4-d, 5-d, etc….

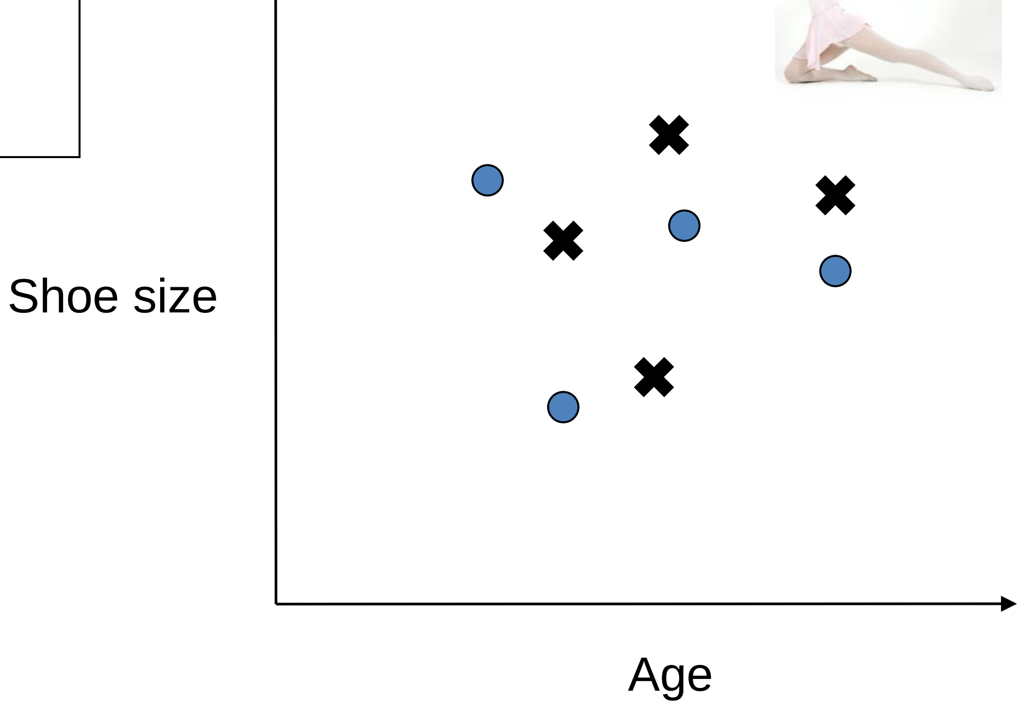
$$d = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2}$$

x = *Height*
y = *Weight*
z = *Shoe size*

Choosing the wrong features makes it difficult, too many and it's computationally intensive.

Possible features:
- Shoe size ✓
- Height
- Age ✓
- Weight

?



K-Nearest Neighbour Model

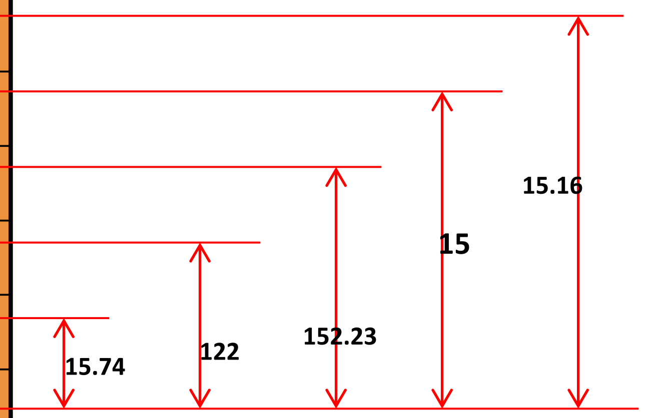
- Example : Classify whether a customer will respond to a survey question using a 3-Nearest Neighbor classifier

Customer	Age	Income	No. credit cards	Response
John	35	35K	3	No
Rachel	22	50K	2	Yes
Hannah	63	200K	1	No
Tom	59	170K	1	No
Nellie	25	40K	4	Yes
David	37	50K	2	?

K-Nearest Neighbour Model

- Example : 3-Nearest Neighbors

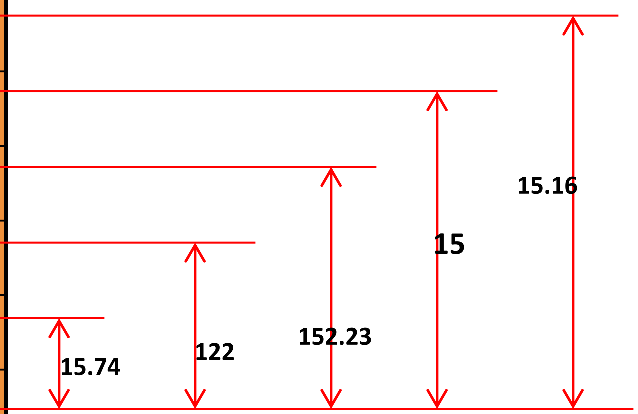
Customer	Age	Income	No. credit cards	Response
John	35	35K	3	No
Rachel	22	50K	2	Yes
Hannah	63	200K	1	No
Tom	59	170K	1	No
Nellie	25	40K	4	Yes
David	37	50K	2	?



K-Nearest Neighbour Model

- Example : 3-Nearest Neighbors

Customer	Age	Income	No. credit cards	Response
John	[illegible]	[illegible]	[illegible]	No
Rachel	[illegible]	[illegible]	[illegible]	Yes
Hannah	63	200K	1	No
Tom	59	170K	1	No
Nellie	[illegible]	[illegible]	[illegible]	Yes
David	37	50K	2	?

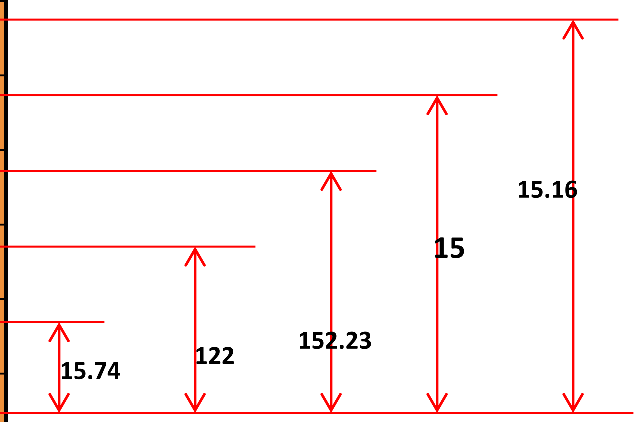


Three nearest ones to David are: No, Yes, Yes

K-Nearest Neighbour Model

- Example : 3-Nearest Neighbors

Customer	Age	Income	No. credit cards	Response
John	[illegible]	[illegible]	[illegible]	No
Rachel	[illegible]	[illegible]	[illegible]	Yes
Hannah	63	200K	1	No
Tom	59	170K	1	No
Nellie	[illegible]	[illegible]	[illegible]	Yes
David	37	50K	2	Yes



Three nearest ones to David are: No, Yes, Yes

K-Nearest Neighbour Model

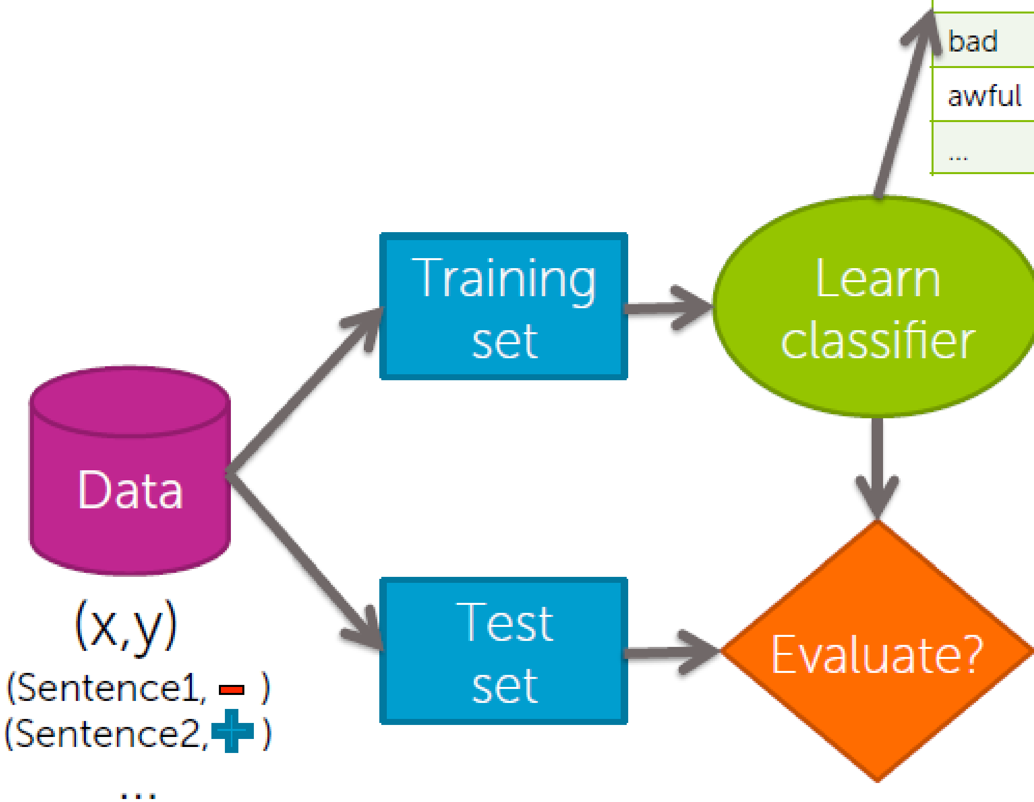
- Example: For the example we saw earlier, pick the best K from the set {1, 2, 3} to build a K-NN classifier

Customer	Age	Income	No. credit cards	Response
John	35	35K	3	No
Rachel	22	50K	2	Yes
Hannah	63	200K	1	No
Tom	59	170K	1	No
Nellie	25	40K	4	Yes
David	37	50K	2	?

Training and evaluating a classifier

Training a classifier= Learning the weights

Word	Weight
good	1.0
awesome	1.7
bad	-1.0
awful	-3.3
...	...



Classification error & accuracy

- Error measures fraction of mistakes

$$\text{error} = \frac{\text{\# of mistakes}}{\text{Total \# of Sentences}}$$

-Best possible value is 0.0

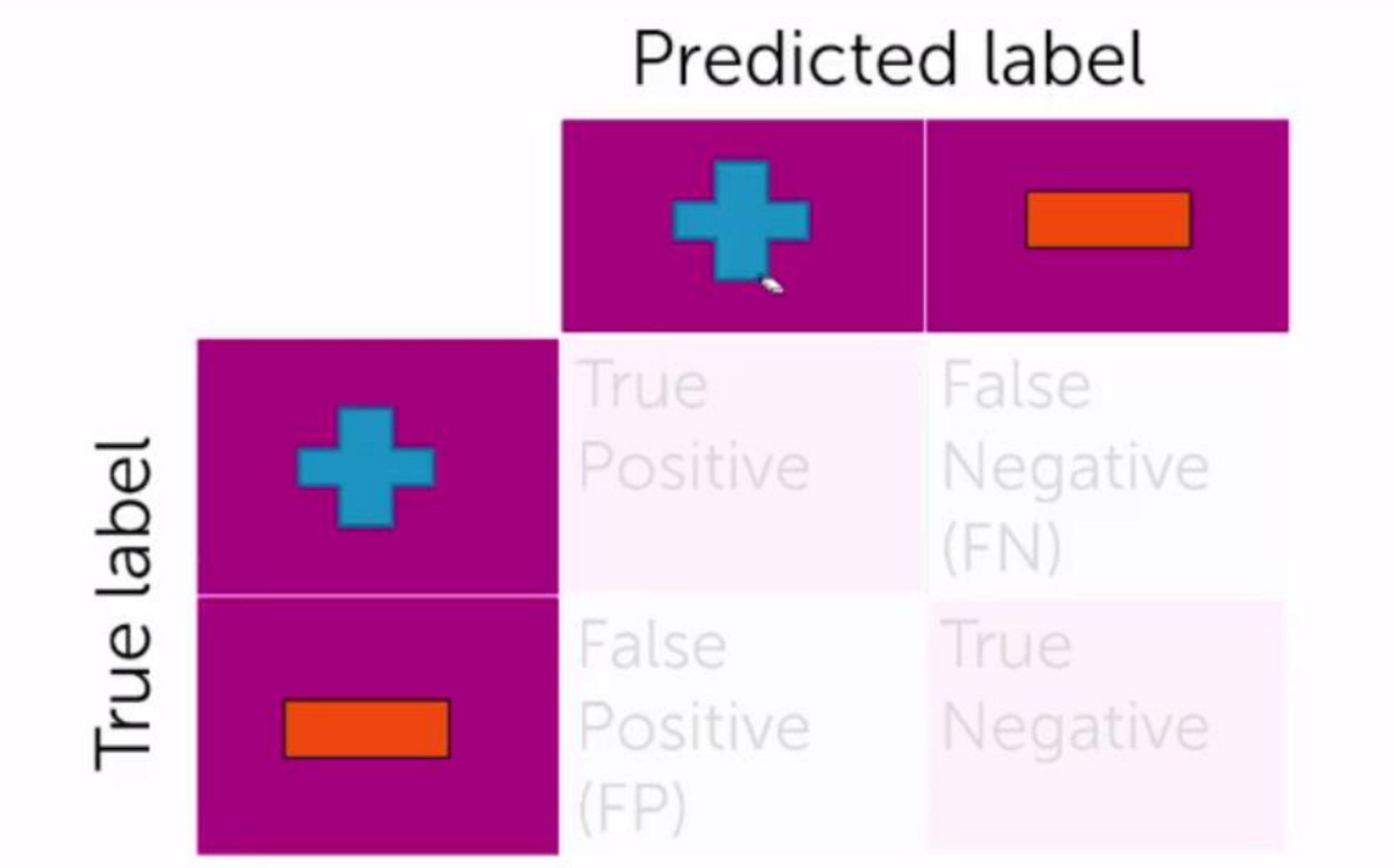
- Often, measure accuracy

-Fraction of correct predictions

$$\text{accuracy} = \frac{\text{\# of Correct}}{\text{Total \# of Sentences}}$$

- -Best possible value is 1.0

Type of mistakes



SPLITTING OF TRAINING AND TEST DATA

Dividing Up Data

- We need independent data sets to train, set parameters, and test performance
- Thus we will often divide a data set into three
  - Training set
  - Parameter selection set
  - Test set
- These **must** be independent
- Data set 2 is not always necessary

Dataset

Inputs		Labels
15	95	1
33	90	1
78	70	0
70	45	0
80	18	0
35	65	1
45	70	1
31	61	1
50	63	1
98	80	0
73	81	0
50	18	0

Inputs		Labels
15	95	1
33	90	1
78	70	0
70	45	0
80	18	0
35	65	1
45	70	1
31	61	1
50	63	1
98	80	0
73	81	0
50	18	0

50:50
split

15	95	1
33	90	1
78	70	0
70	45	0
80	18	0
35	65	1

45	70	1
31	61	1
50	63	1
98	80	0
73	81	0
50	18	0

- Can be 70:30 or any other

15	95		1
33	90		1
78	70		0
70	45		0
80	18		0
35	65		1

Training set

Train a K-NN on this…

45	70		1
31	61		1
50	63		1
98	80		0
73	81		0
50	18		0

Testing set

… then, test it on this!

"simulates" what it might be like to see new data in the future

Cross Validation

- We can solve our dilemma by repeating the training many times on different partitioning
- This is known as K-fold cross validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

$$D = \{D_i\}_{i=1}^{P} \quad D_i = (x_i, y_i)$$

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Test Set Training Set

5-fold cross-validation

$E_g =$ 5.1

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set Test Set Training Set

5-fold cross-validation

$$E_9 = \quad 3.7$$

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set Test Set Training Set

5-fold cross-validation

$E_g =$ 4.6

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set   Test Set   Training Set

5-fold cross-validation

$E_g =$ 4.6

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set Test Set

5-fold cross-validation

$E_g =$ 3.3

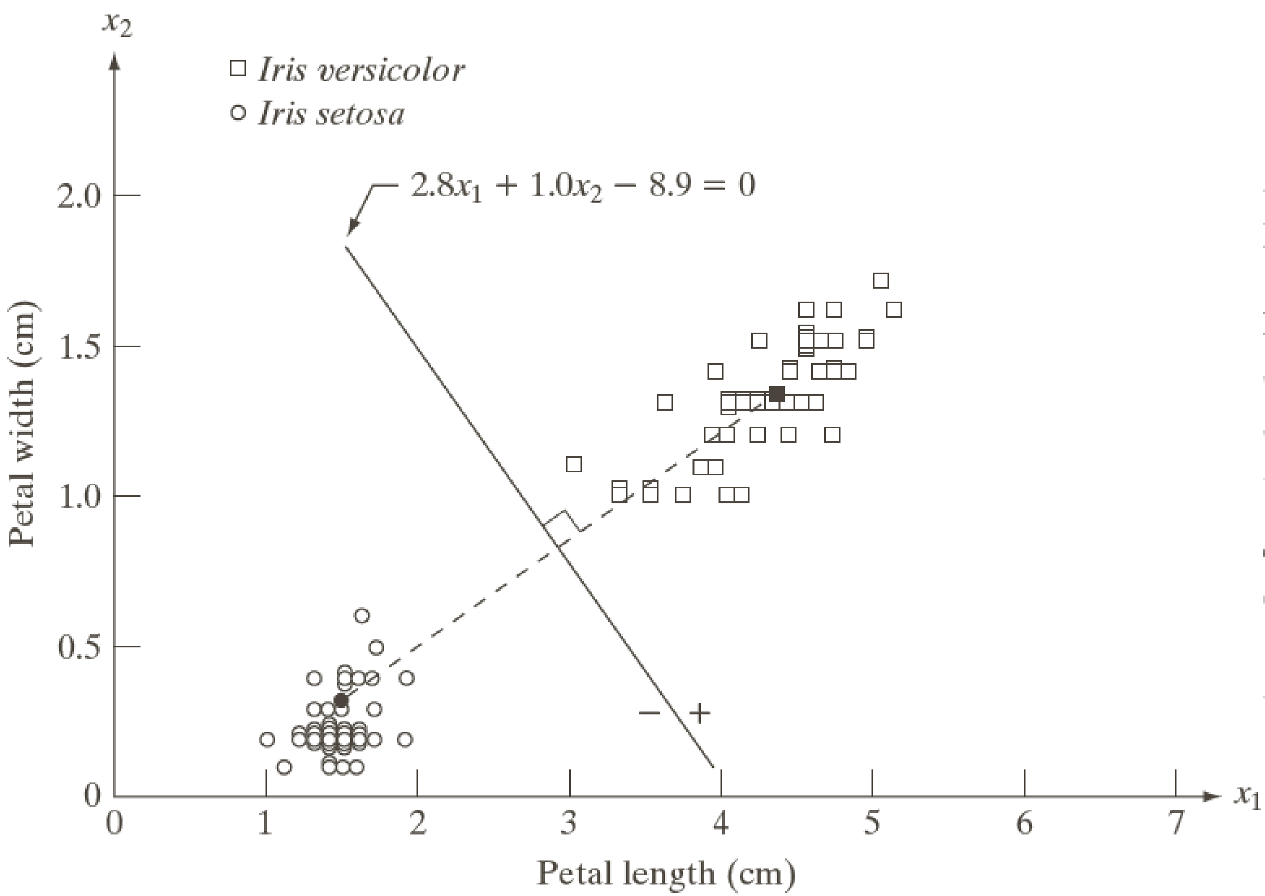
Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

$$\langle E_g \rangle = \frac{5.1 + 3.7 + 4.6 + 4.6 + 3.3}{5} = 4.3$$

Minimum Distance

Minimum Distance Classifier



Decision boundary of minimum distance classifier for the classes of *Iris versicolor* and *Iris setosa*. The dark dot and square are the means.

Minimum Distance Classifier

- For a test sample X, compute $D_j(X)$ for each class j
- Assign class with minimum D(x) value

$$D_j(\mathbf{x}) = \left\| \mathbf{x} - \mathbf{m}_j \right\|$$

- Here m_j $= \left[(\mathbf{x} - \mathbf{m}_j)^T (\mathbf{x} - \mathbf{m}_j) \right]^{1/2}$ n j^{th} class

Minimum Distance Classifier

Manipulating $D_j(X)$

$$\begin{aligned} D_j^2(\mathbf{x}) &= \left\|\mathbf{x}-\mathbf{m}_j\right\|^2 = (\mathbf{x}-\mathbf{m}_j)^T(\mathbf{x}-\mathbf{m}_j) \\ &= \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\mathbf{m}_j + \mathbf{m}_j^T\mathbf{m}_j \\ &= \mathbf{x}^T\mathbf{x} - \mathbf{2}\left(\mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j\right). \end{aligned}$$

Minimum Distance Classifier

- Now instead of $D_j(X)$, we compute discriminent function $d_j(X)$ for each class
- Assign class with maximum $d_j(X)$ value

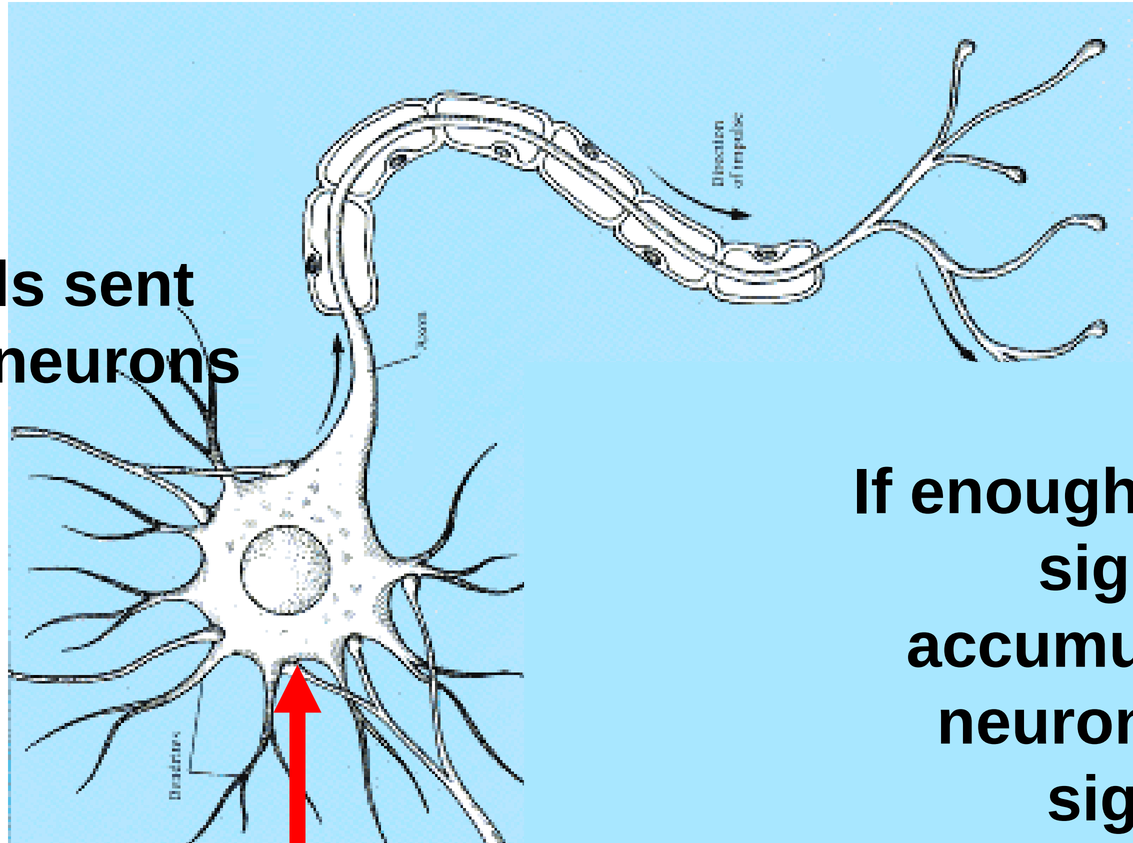
$$d_j(\mathbf{x}) = \mathbf{x}^T \mathbf{m}_j - \frac{1}{2} \mathbf{m}_j^T \mathbf{m}_j \qquad j = 1, 2, \ldots, W$$

- Equation for decision boundary between two classes *i* and *j*

$$d_{ij}(\mathbf{x}) = \mathbf{x}^T (\mathbf{m}_i - \mathbf{m}_j) - \frac{1}{2} (\mathbf{m}_i^T \mathbf{m}_i - \mathbf{m}_j^T \mathbf{m}_j)$$

Artificial Neural Network - Perceptron

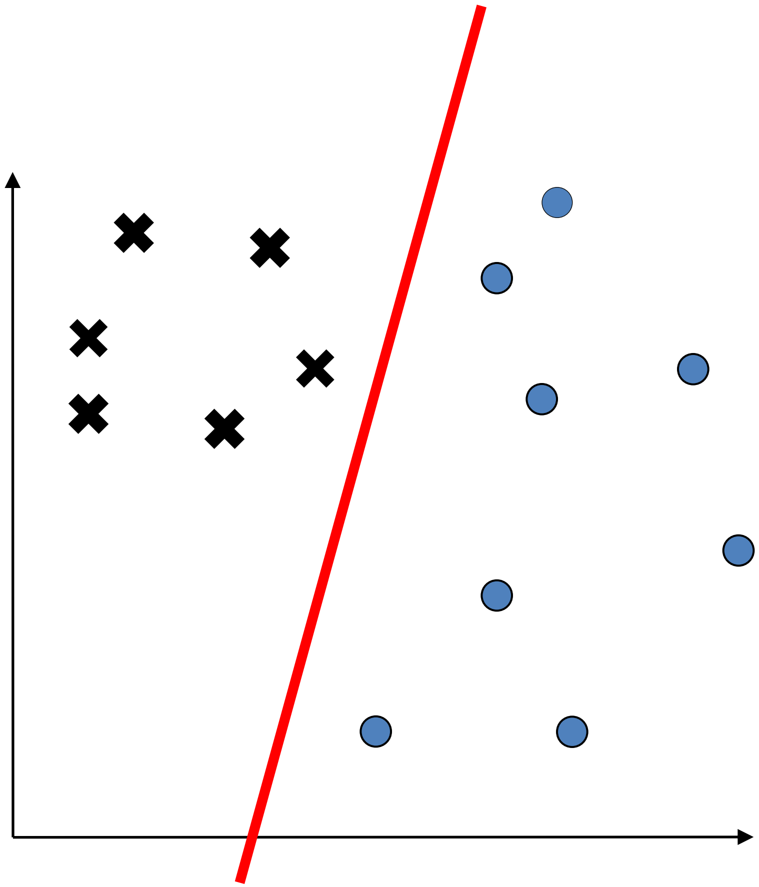
Input signals sent from other neurons



If enough sufficient signals accumulate, the neuron fires a signal.

Connection strengths determine how the signals are accumulated

A (Linear) Decision Boundary



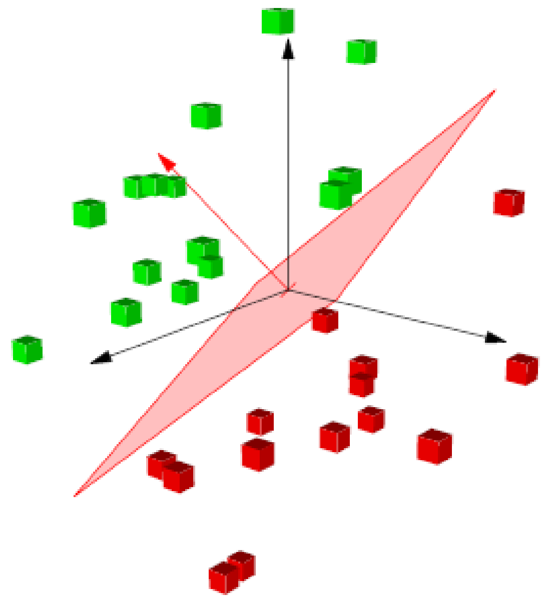
Represented by:
One artificial neuron called a "Perceptron"

-

Low space complexity

Perceptron

- The perceptron with a step function performs **classification**
- The perceptron can be 'visualised' as a decision boundary in input space



- The perceptron can only separate linear-separable inputs

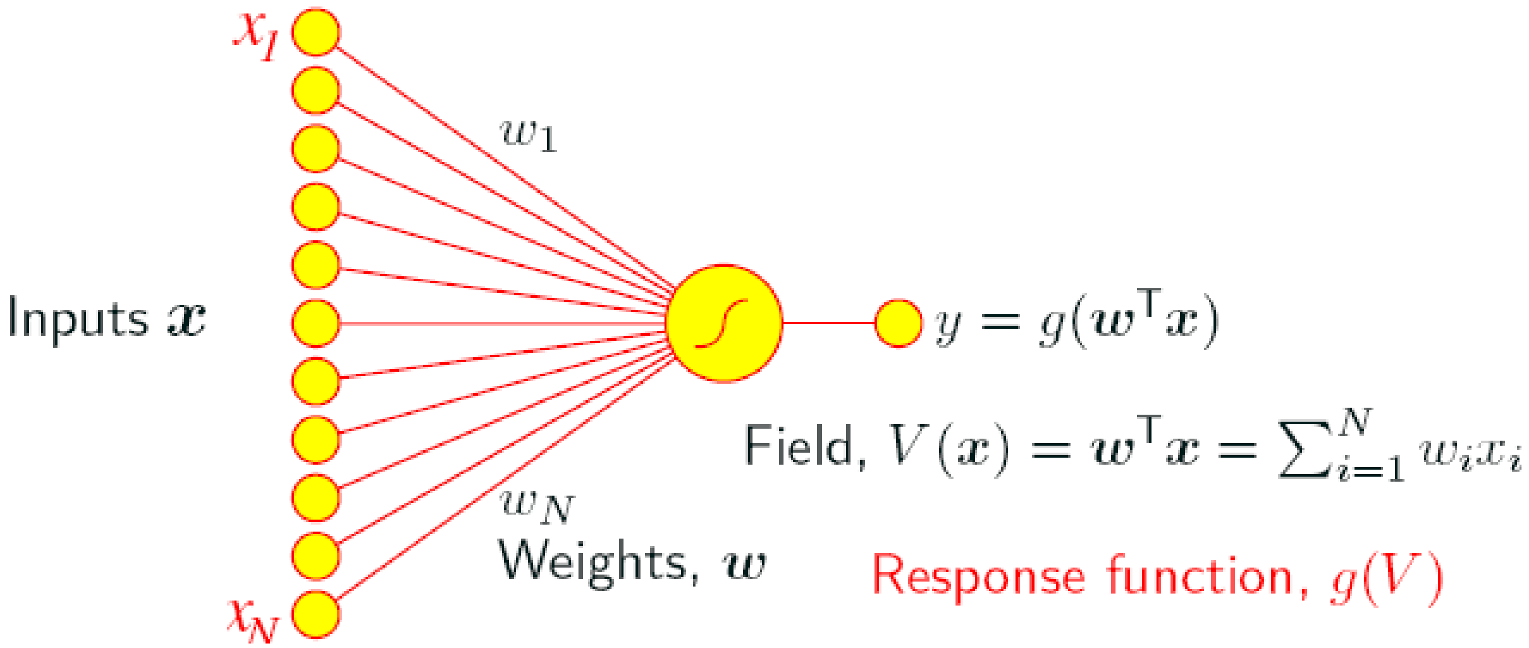
Perceptron

- Given (numeric) input features $x = (x_1, x_2, \ldots, x_n)$
- Prediction given by $f(x; w)$
- w are parameters or "weights" that we train
- The **perceptron** provides the classic example of a **parametric** learning algorithm

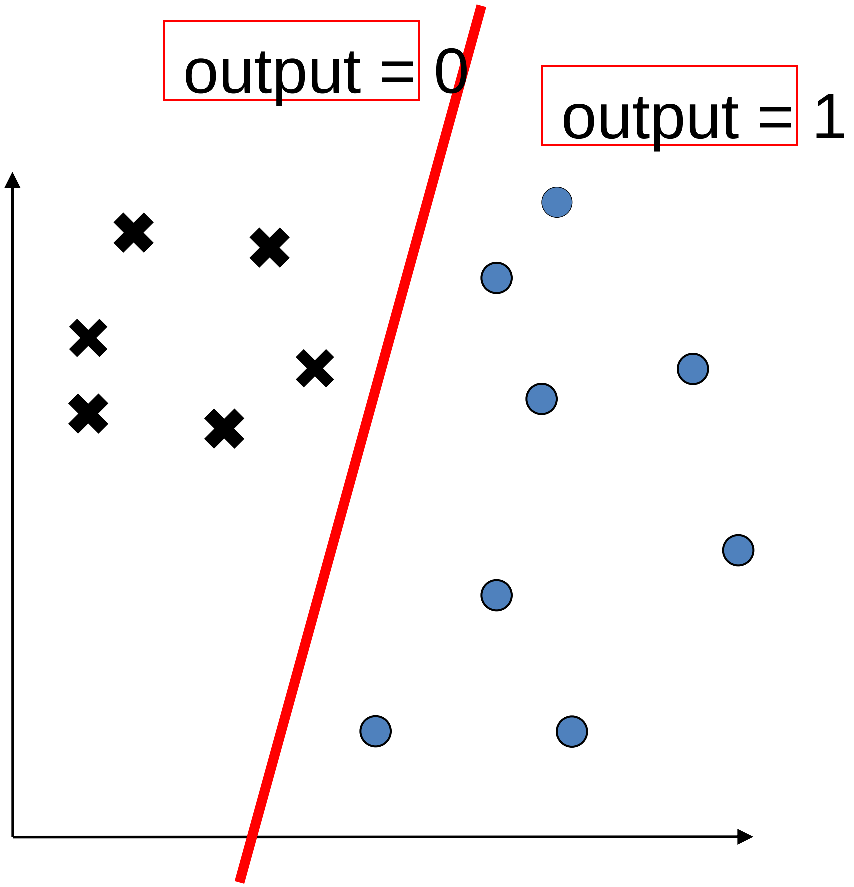
Perceptron

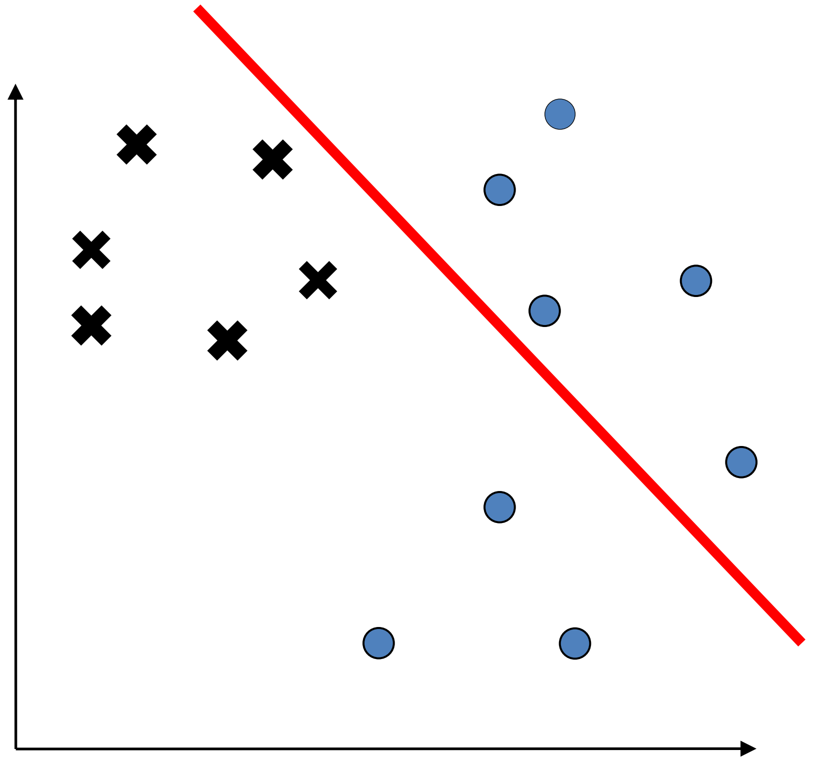
- Proposed by Frank Rosenblatt (1958) (Widrow and Hoff proposed **adaline** at same time)

- Schematic representation



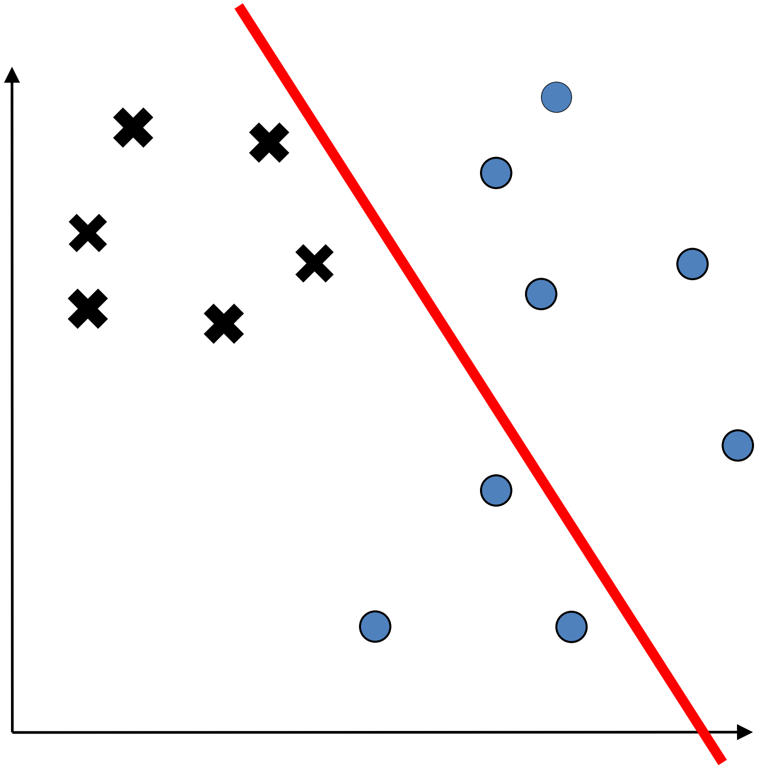
$$\textit{if} \quad \left(\sum_{i=1}^{M} x_i w_i \right) \quad > \quad t \qquad \text{then } output = 1, \text{ else } output = 0$$





Is this a good decision boundary?

$$\textit{if} \quad \left(\sum_{i=1}^{M} x_i w_i \right) \quad > \quad t \qquad \text{then } output = 1, \ \text{else } output = 0$$

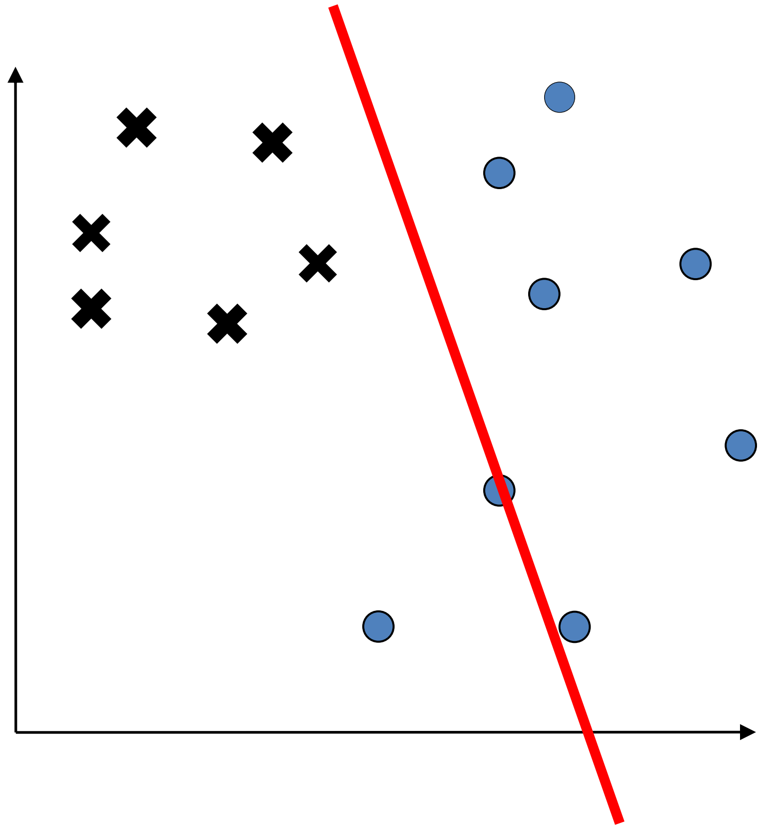


$w_1 = 1.0$

$w_2 = 0.2$

$t = 0.05$

$$if \quad \left(\sum_{i=1}^{M} x_i w_i \right) \quad > \quad t \qquad \text{then } output = 1, \ \text{else } output = 0$$

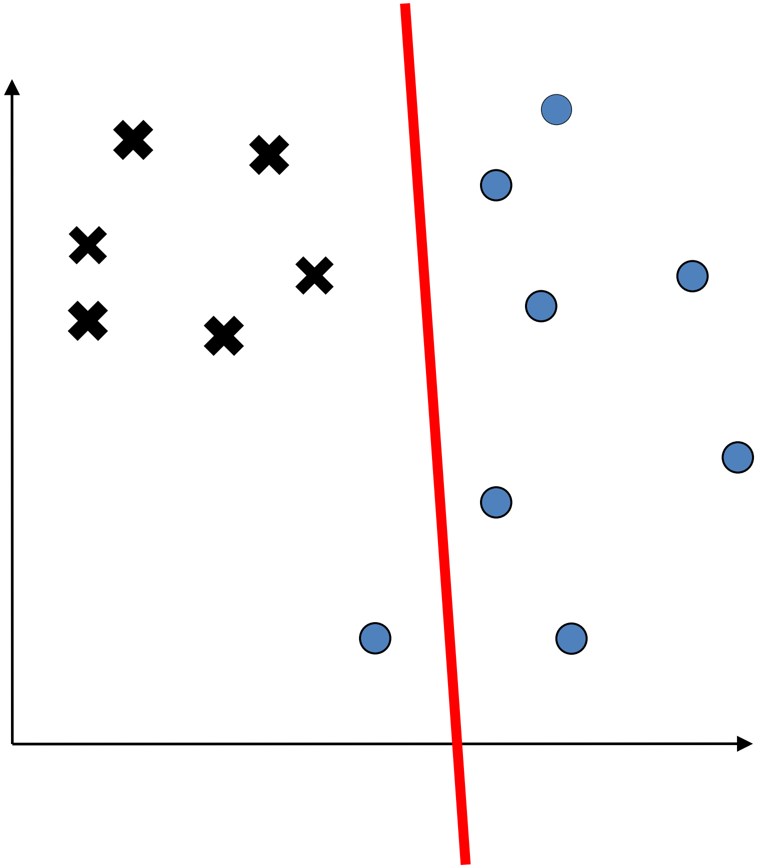


$w_1 = 2.1$

$w_2 = 0.2$

$t = 0.05$

$$if \left(\sum_{i=1}^{M} x_i w_i \right) > t \quad \text{then}\ output = 1,\ \text{else}\ output = 0$$

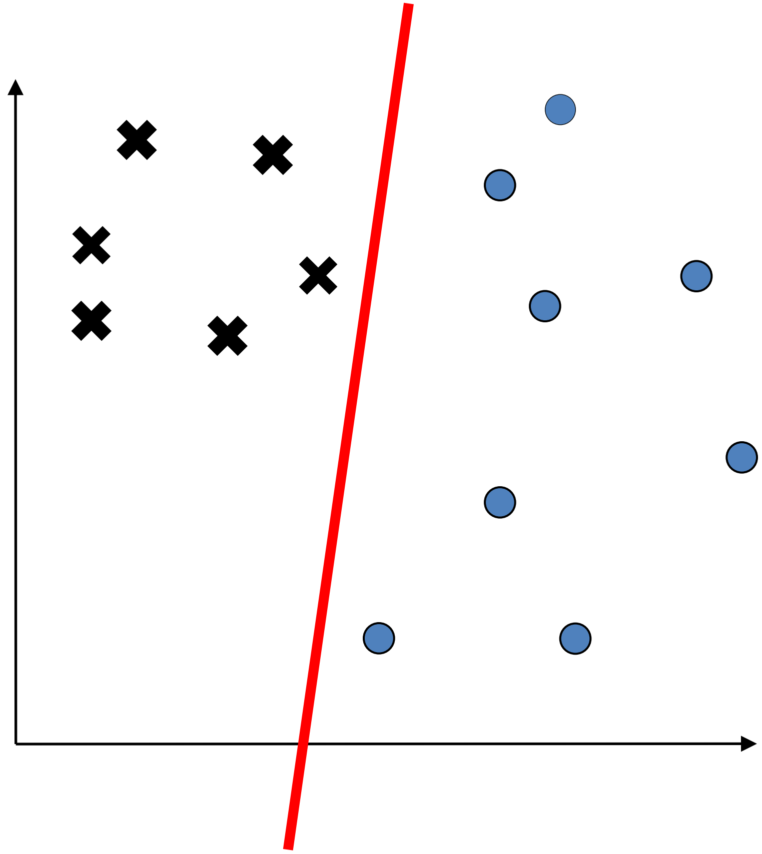


$w_1 = 1.9$

$w_2 = 0.02$

$t = 0.05$

$$\text{if} \quad \left(\sum_{i=1}^{M} x_i w_i \right) \quad > \quad t \qquad \text{then } output = 1, \ \text{else } output = 0$$



$w_1 = -0.8$

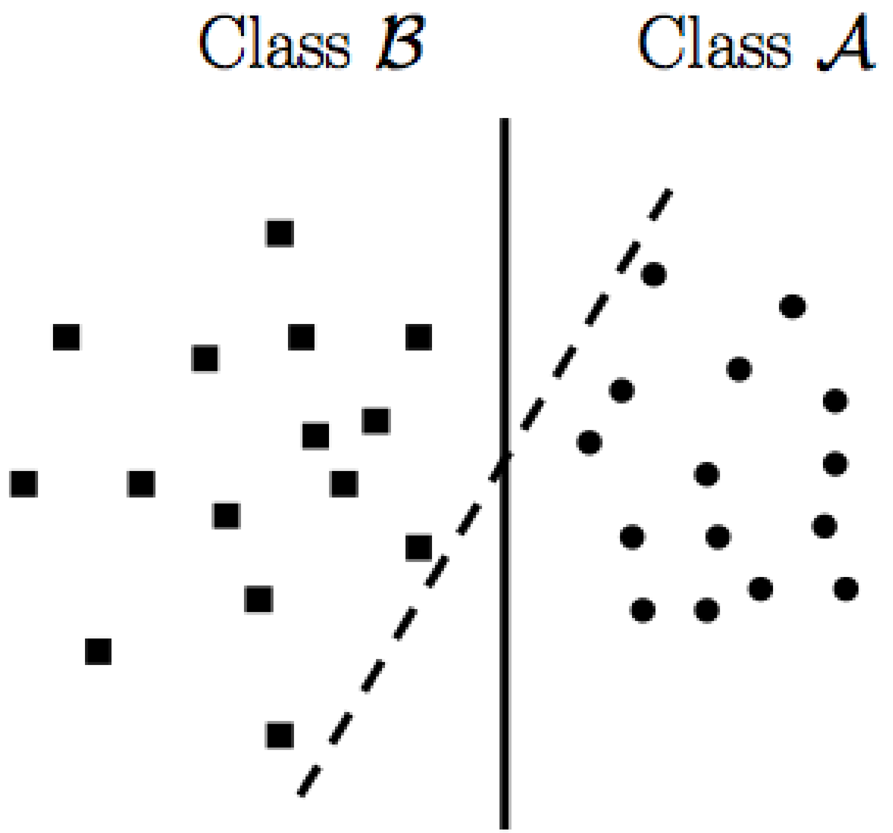
$w_2 = 0.03$

$t = 0.05$

Changing the weights/threshold makes the decision boundary move.

Overview of SVM w.r.t. Perceptron

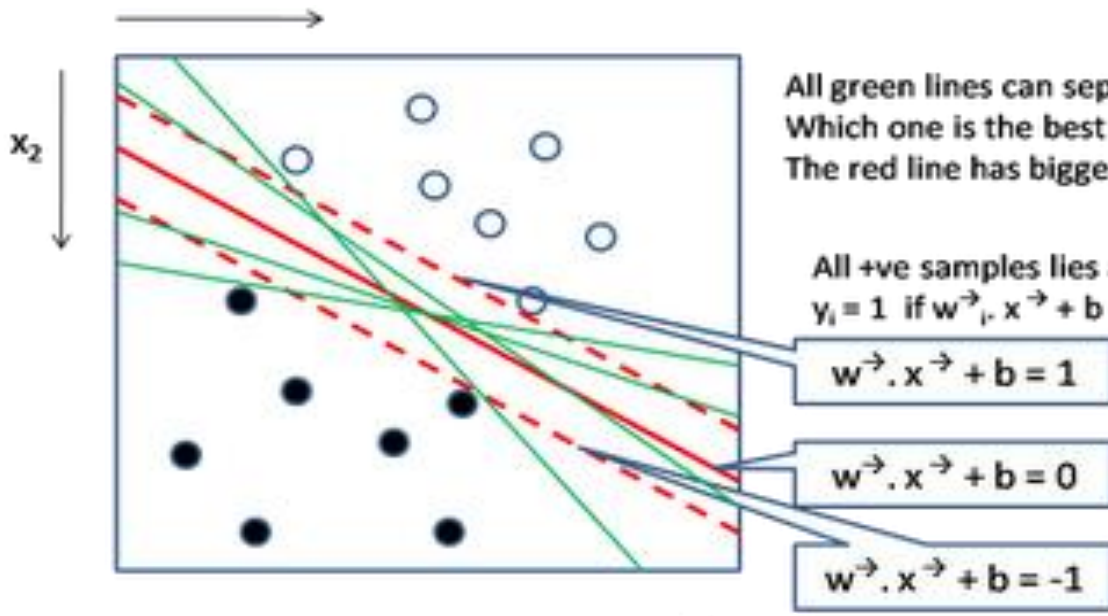
Perceptron



Which plane is best?

Perceptron VS SVM

All green lines can separate the points
Which one is the best ?
The red line has biggest margin



All +ve samples lies above this line
$y_i = 1$ if $w^{\rightarrow}_i . x^{\rightarrow} + b > 1$

All -ve samples lies below this line
$y_i = -1$ if $w^{\rightarrow}_i . x^{\rightarrow} + b < -1$

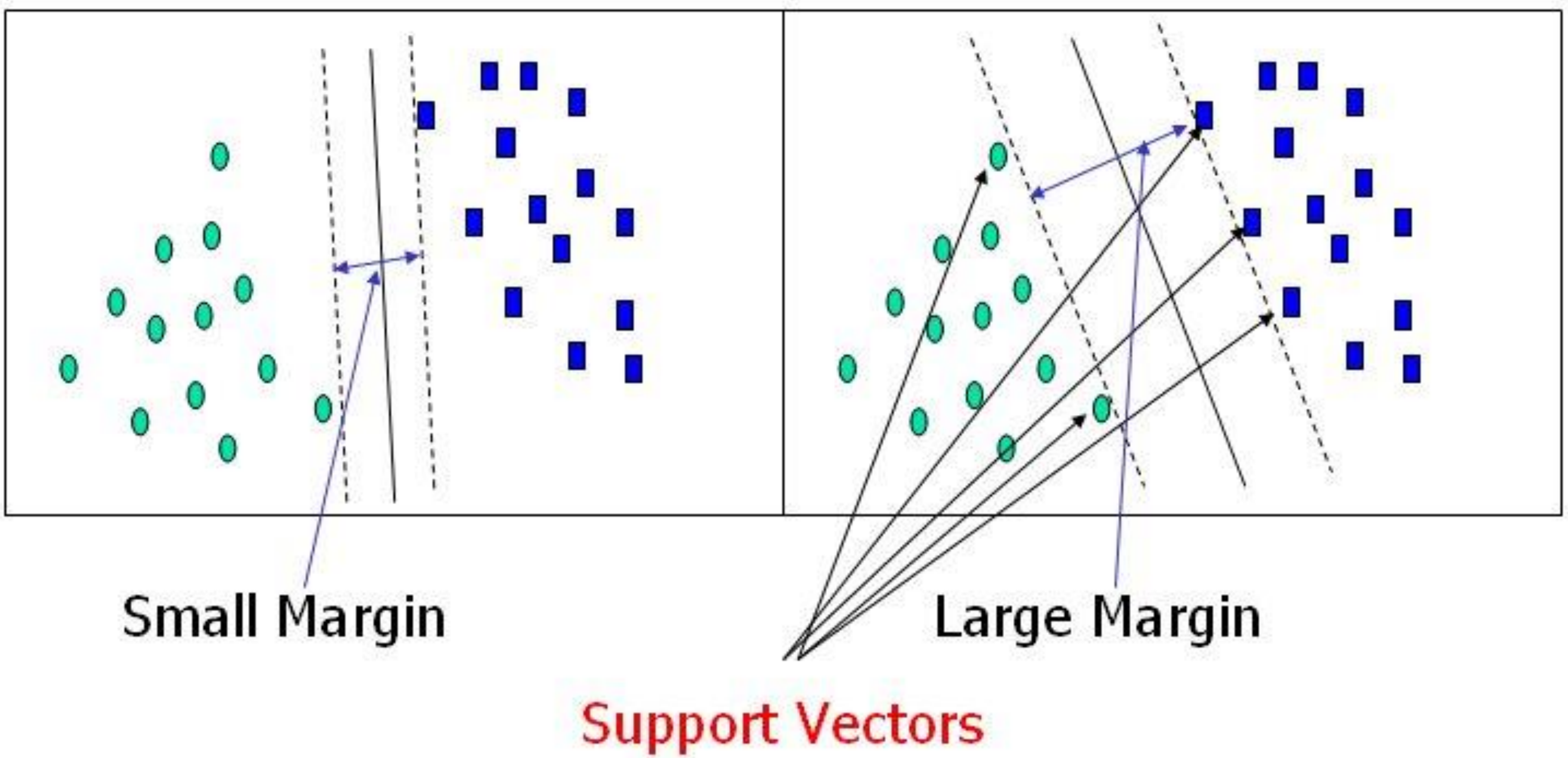
Margin = $(\text{point}_{\text{upperline}} - \text{point}_{\text{lowerline}}) . w^{\rightarrow} / |w|$
since $(\text{point}_{\text{upperline}}). w^{\rightarrow} + b = 1$
And $(\text{point}_{\text{lowerline}}). w^{\rightarrow} + b = -1$
So margin = $(1 - b + 1 + b) / |w| = 2 / |w|$
Maximize margin is max $2 / |w|^2$ is min $|w|^2 / 2$

<u>Problem:</u>
Minimize $(½)\, w^{\rightarrow} . w^{\rightarrow}$
With constraint: $y_i(w^{\rightarrow}_i . x^{\rightarrow} + b) > 1$

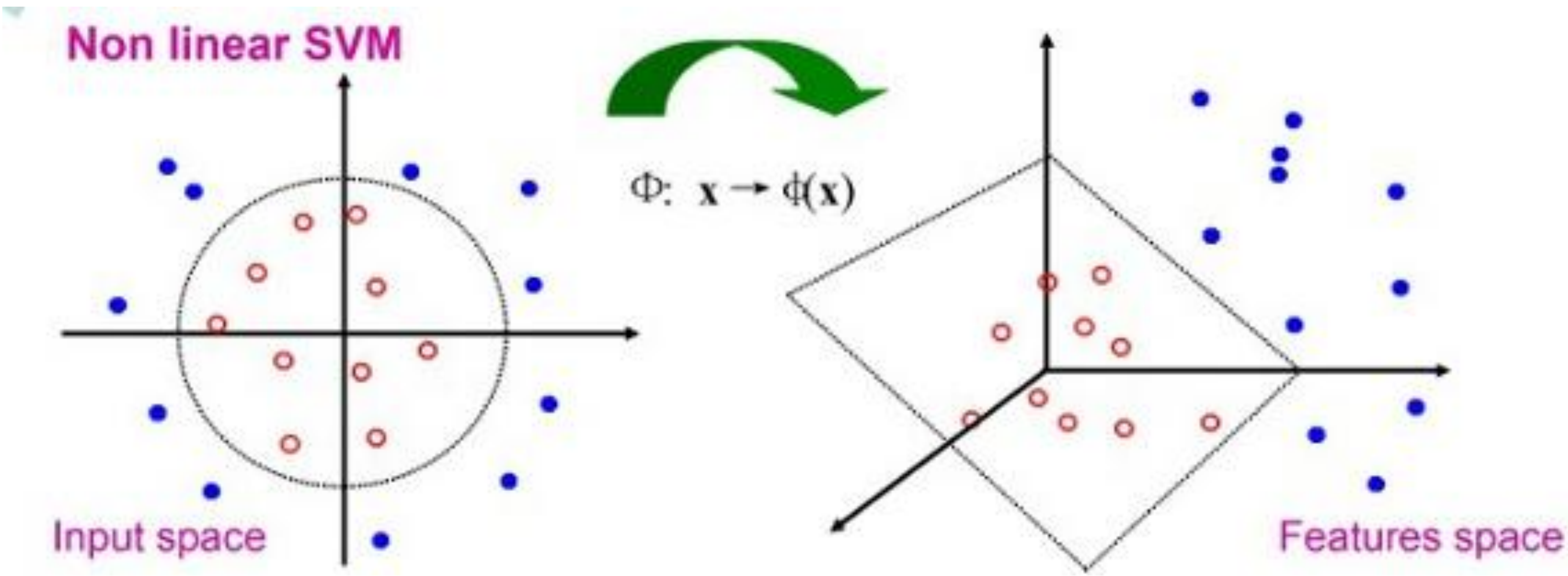
Perceptron VS SVM

- The Perceptron does not try to optimize the separation "distance". As long as it finds a hyperplane that separates the two sets, it is good. SVM on the other hand tries to maximize the "support vector", i.e., the distance between two closest opposite sample points.
- The SVM typically tries to use a "kernel function" to project the sample points to high dimension space to make them linearly separable, while the perceptron assumes the sample points are linearly separable.
- SVM Requires more parameters as compared to
  - choice of kernel
  - selection of kernel parameters
  - selection of the value of the margin parameter

SVM and Margins



SVM for Nonlinear Data



Resources: Datasets

- UCI Repository: http://www.ics.uci.edu/~mlearn/MLRepository.html
- UCI KDD Archive: http://kdd.ics.uci.edu/summary.data.application.html
- Statlib: http://lib.stat.cmu.edu/
- Delve: http://www.cs.utoronto.ca/~delve/

Resources: Journals

- Journal of Machine Learning Research
  www.jmlr.org
- Machine Learning
- IEEE Transactions on Neural Networks
- IEEE Transactions on Pattern Analysis and Machine Intelligence
- Annals of Statistics
- Journal of the American Statistical Association
- ...

Resources: Conferences

- International Conference on Machine Learning (ICML)
- European Conference on Machine Learning (ECML)
- Neural Information Processing Systems (NIPS)
- Computational Learning
- International Joint Conference on Artificial Intelligence (IJCAI)
- ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)
- IEEE Int. Conf. on Data Mining (ICDM)

Acknowledgements

Material in these slides has been taken from, the following resources

- Emily Fox & Carlos Guestrin, Machine Learning Courses, University of Washington, Coursera
- Introduction to Machine Learning, Alphaydin
- Statistical Pattern Recognition: A Review – A.K Jain et al., PAMI (22) 2000
- Pattern Recognition and Analysis Course – A.K. Jain, MSU
- *Pattern Classification"* by Duda et al., John Wiley & Sons.

THANK YOU