Welcome

Back to the classroom

Agenda

- Introduction to AIML world
- Demystify Machine Learning
- Relate many popular problems
- Abstracting the problems
- Understand the common structure
- Three Simple Algorithms (that still work in real world)
- Discussions

Introduction

Where are we now?

AI and ML at Work

Autonomous Cars and Navigation

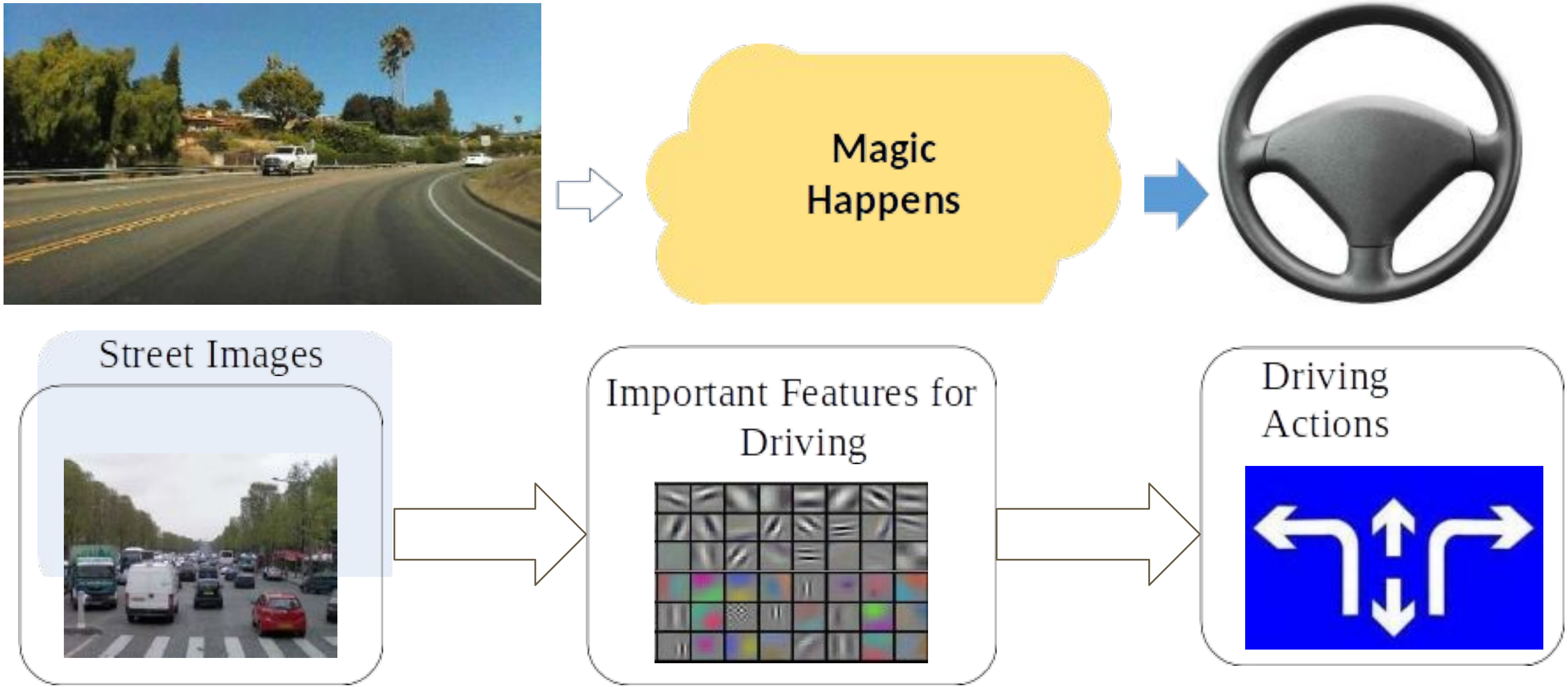
“Alexa”,“Siri”,“Cortana” etc.

AI and ML at Work

Creativity: Generated Images

Playing Games better than Human

Modern AI: End 2 End Driving



What is Modern AI and ML?

Modern "AI"

NLP

Computer Vision

Speech Processing

Robotics

A Simplified View

A simple question

- 1, 3, 5, 7, 9, ... What is the next number?

  Ans: 11 Odd numbers $2x + 1$

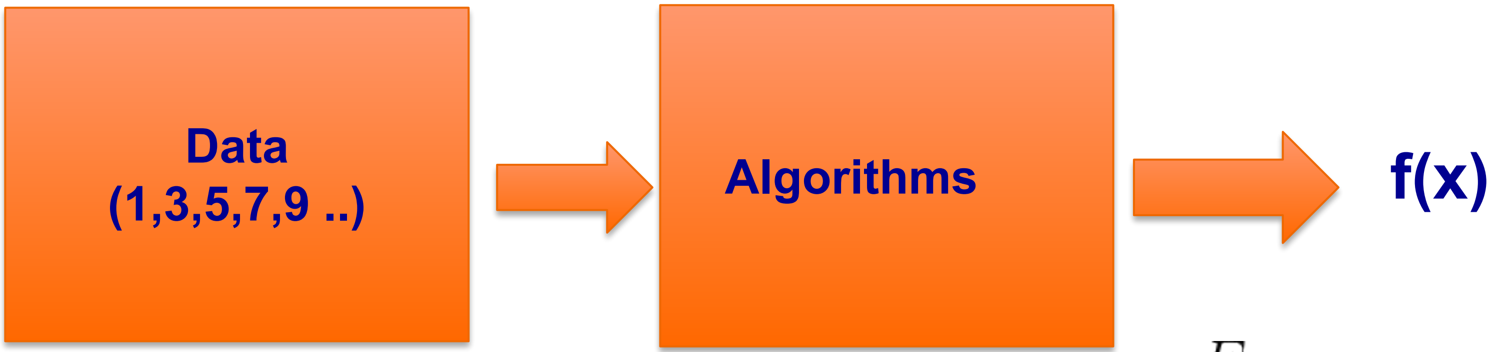
- 1, 3, 9, 19, 33, ... What is the next number?

  Ans: 51 $2x^2 + 1$

A simple question

- How do we solve such problems?
- Find a pattern from the examples
  - (function f(n) = $2x^2 + 1$ or model the data)
- Use it to predict the next number (or solve the problem)
- How do we design a computational procedure?

A Simplified View of ML



$$
\begin{array}{l}
Eg: \\
int\ \ function(int\ \ x[\,])\,\{ \\
\ldots \\
\}
\end{array}
$$

A simple question (cont.)

- We know: 1, 3, 9, 19, 33, ... What is the next number?

  Ans: 51 $2x^2 + 1$

- 0.99, 3.02, 9.00, 18.98, 33.01, ... What next?

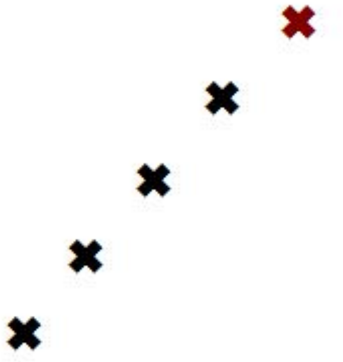
A simple question (cont.)

Consider a series of 2D points

- (1,3), (2,6), (3,9), (4,12), ...
- What is the next point?

- (x,3x) Or
  - Function:

    Y = f(X)=3.X



What makes it Difficult?

- **When numbers are “uncertain”**
  - Noise in measurements
  - Missing values
- **When numbers are not just “simple numbers”**
  - 2D points, 3D points
  - 100 Dimensional points

What makes it Difficult?

- **When the function is complex or function nature is unknown**
  - Simple linear functions are easy to guess.
    - Eg. $F(x) = w_1x + w_2$
  - Finding “best” parameters/coefficients can be hard.
    - What is the best “w” that suits the data?

More Examples

- Given a set of numbers {7,26,17,11,25,32,5,8,92}, partition into two sets: (Unsupervised Learning)

  - Odd (7,17,11,25,5) and Even (26,32,8,92)

  - Why this? Why not single and two digit?
  - Both mine and your solutions can be right?

More Examples

- Given a set of male people with and without anemia, their hemoglobin levels are: (Supervised Learning)
  - Positive cases: {8.5.9.2.7.4.7.8}
  - Negative cases: {15.0, 14.9, 14.2,13.8}
  - Does a patient with 7.7 have anemia?
  - Classification is simple: “anemia if $f(x) < 10$”
  - Why 10? Why not 12?
  - Multiple solutions. Both works well now. Future?

Closer Look..

- Who gives samples/examples?
  - The Data
  - Data + interpretations (***X***,Y)=(sample, label)
    - Interpretations are the “supervisory signals”
- Who gives functional form?
  - Most problems need complex functions

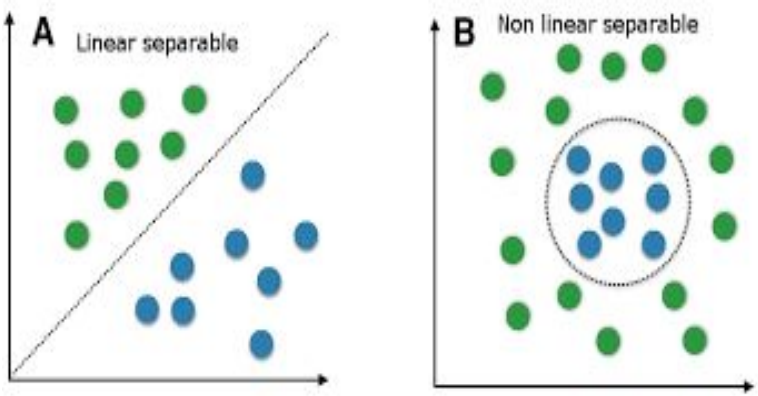
  (Note that simple “Linear” solutions are also good in many cases.)

Closer Look..

- How to find the “optimal” parameters?
  - Optimization problem
    - Find the best “w” (coefficients) for a given data/problem?
  - Training
  - Computing
- How do we expect that it will work well in the future?

“Classification”: A popular problem

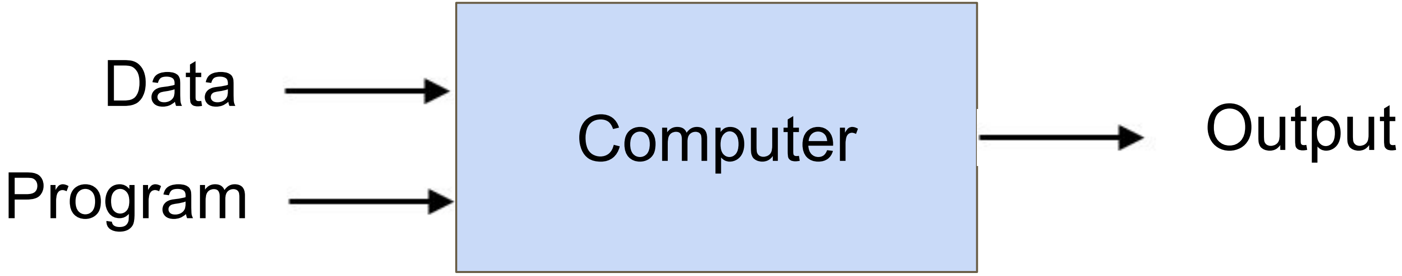
- Example:
  - Given medical records, predict presence of Malaria
- Data: A set of Samples { X } labeled by experts.
- Performance: Predict accurately on unseen data
- {0,1} classification
  - “Yes” or “No”
  - Yes if $f(x) > 0$
- Multiclass classification
- Many more variants



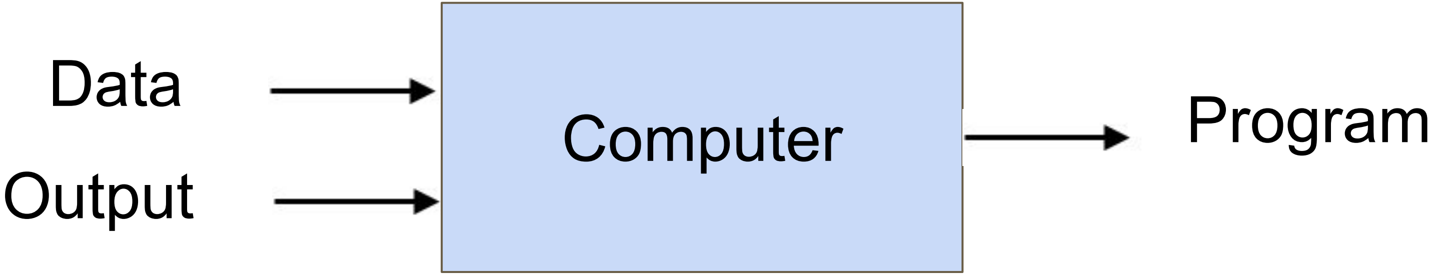
Problem Space

- **Feature Extraction**: Find *X* corresponding to an entity/item *I* (such as an image, web page, ECG etc.)
- **Classification:** Find a parameterized function $f_w(X)$ which can make the right predictions *Y*
- **End to End:** Can we learn *Y* directly from *I*

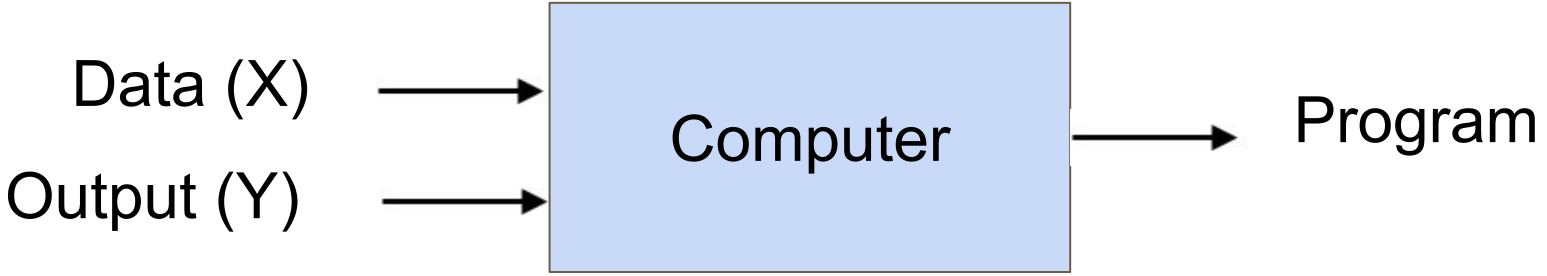
Traditional Programming



Machine Learning



Machine Learning



AI-ML Avatars

What is Machine Learning?

- A branch of Artificial Intelligence
  - The design and development of algorithms
  - computers to capture and model behaviors
  - based on empirical data
- Intelligence require Knowledge
  - It is necessary for the computers to acquire Knowledge
  - Learn from external world; “teachers” etc. and solve problems
  - Data provides knowledge in many cases

What is Machine Learning?

- A very popular area now
  - Lots of data
  - Many recent success stories

What is Machine Learning?

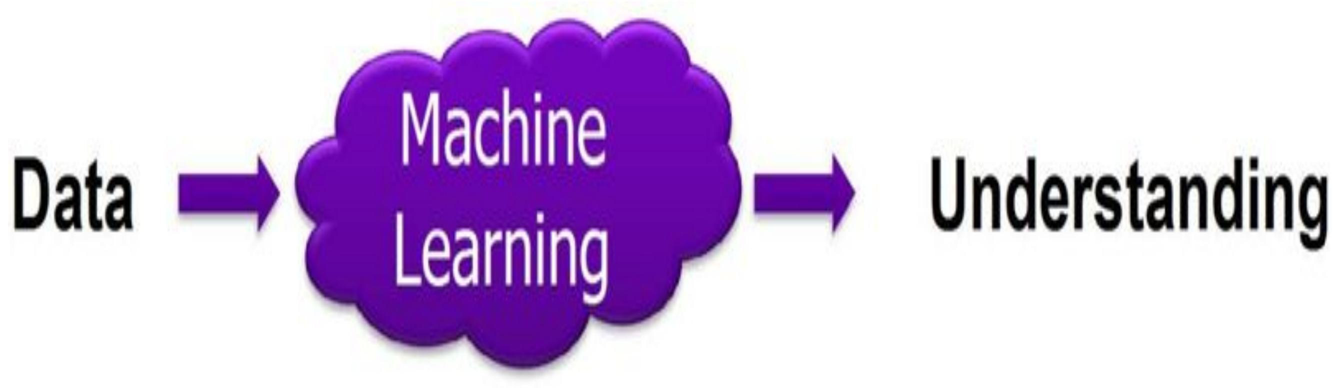
- [Arthur Samuel, 1959]
  - Study that gives computers the ability to learn without being explicitly programmed

- [Kevin Murphy] algorithms that
  - Automatically detect patterns in data
  - Use the uncovered patterns to predict future data or other outcomes of interest

What is Machine Learning?

- [Tom Mitchell] algorithms that
  - Improve their performance (P)
  - At some task (T)
  - With experience (E)

What is Machine Learning?



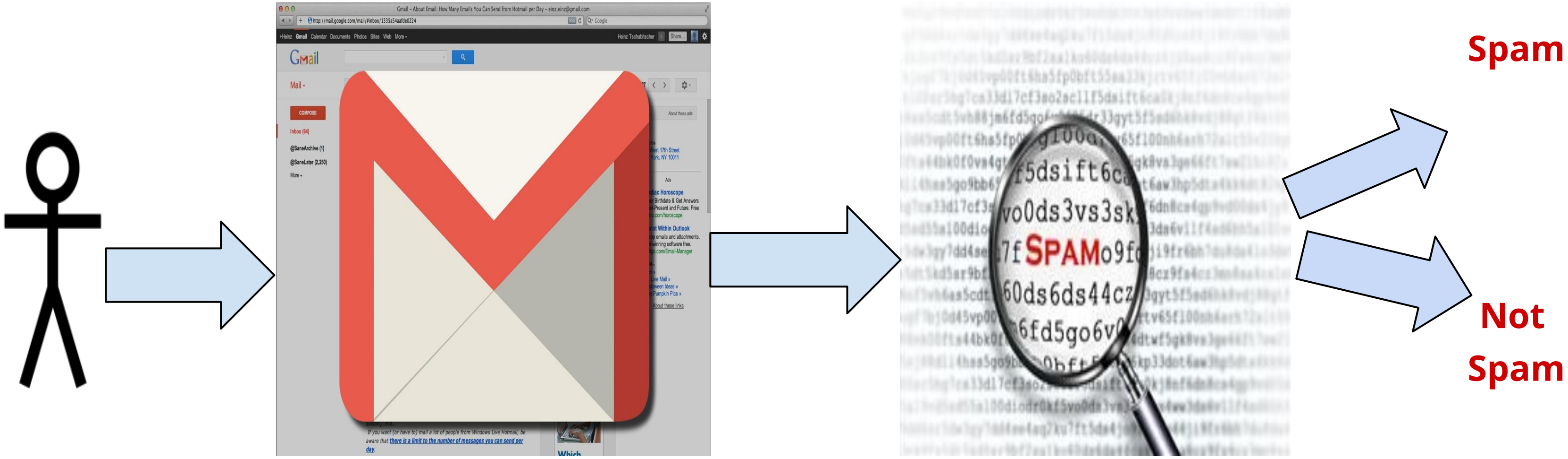
Problem Space

- **Feature Extraction**: Find *X* corresponding to an entity/item *I* (such as an image, web page, ECG etc.)

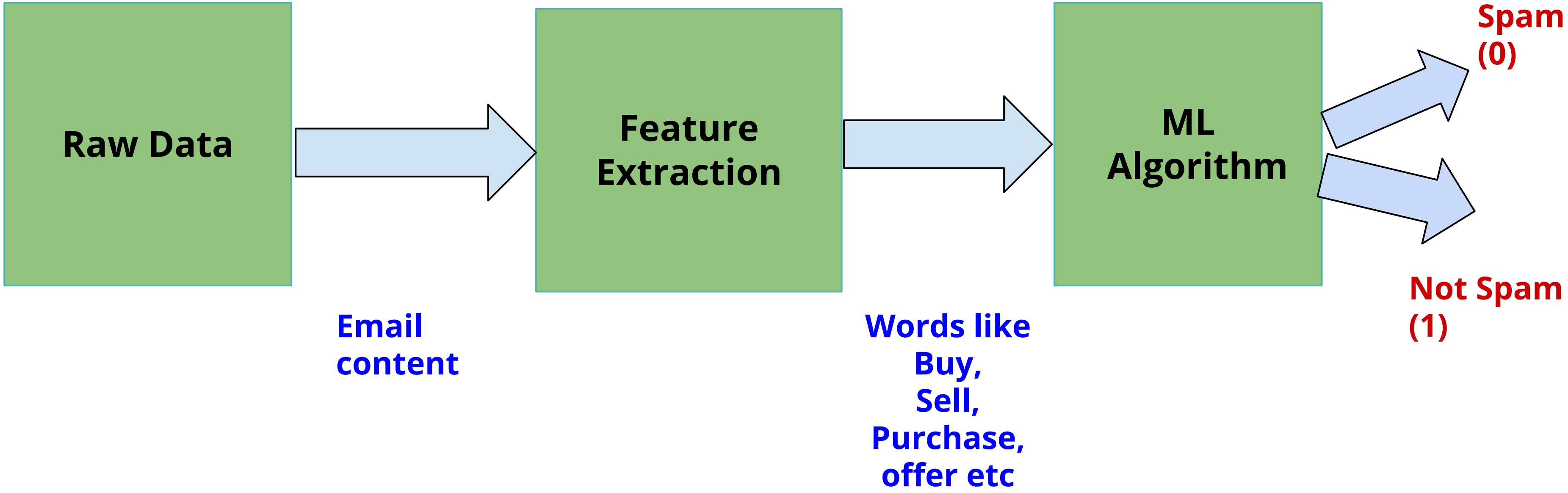
- **Classification:** Find a parameterized function $f_w(X)$ which can make the right predictions *Y*

- **End to End:** Can we learn *Y* directly from *I*

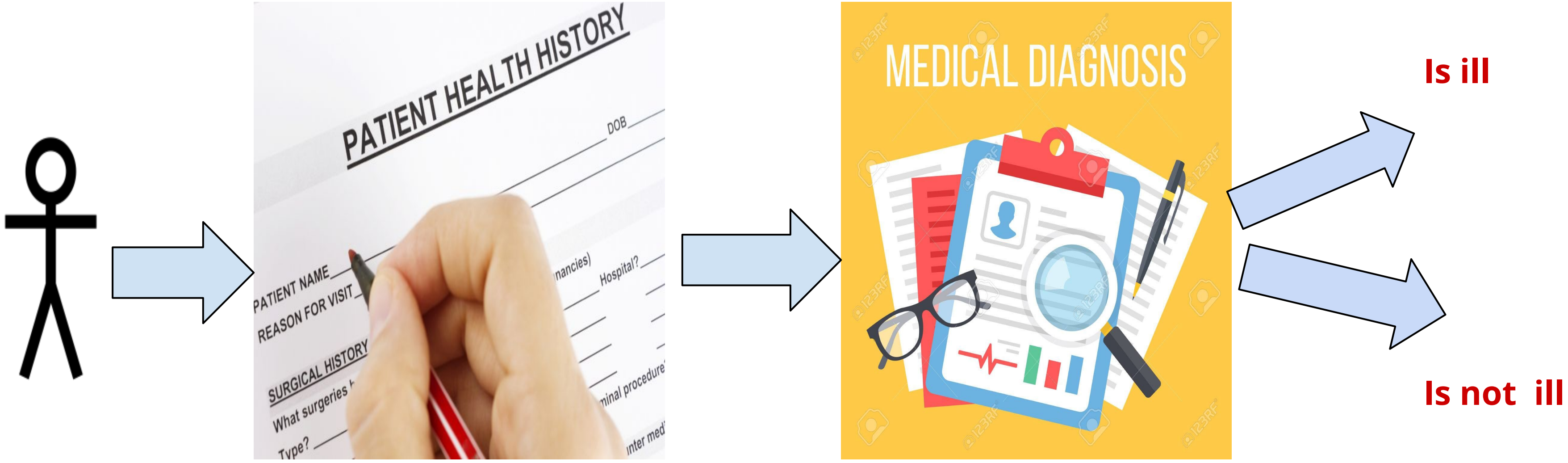
Spam Detection



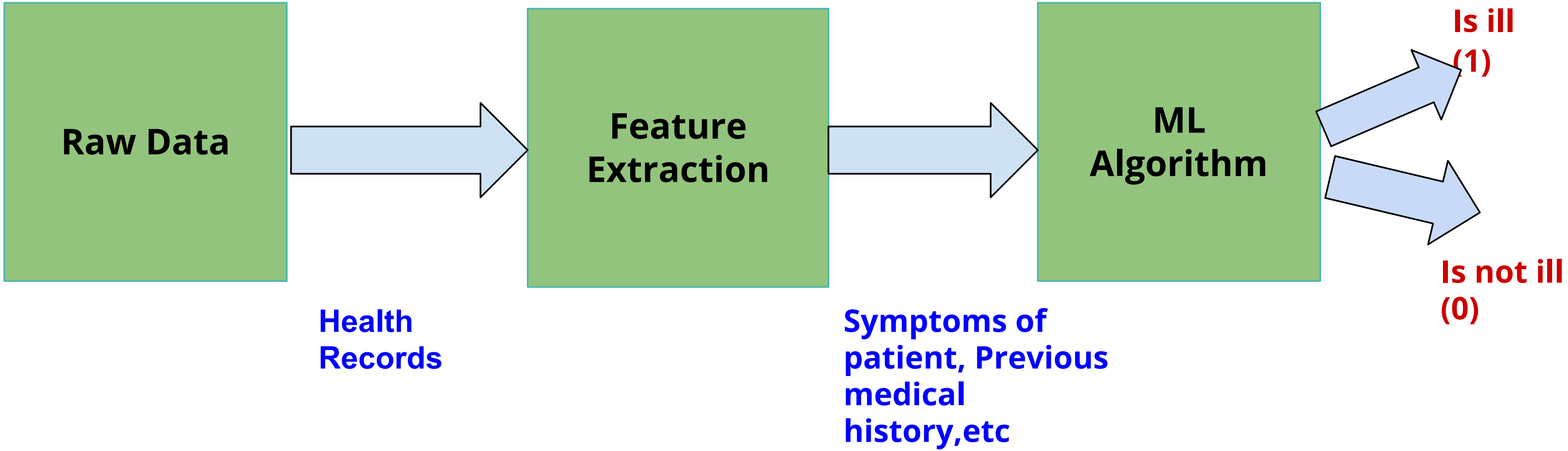
Spam Detection



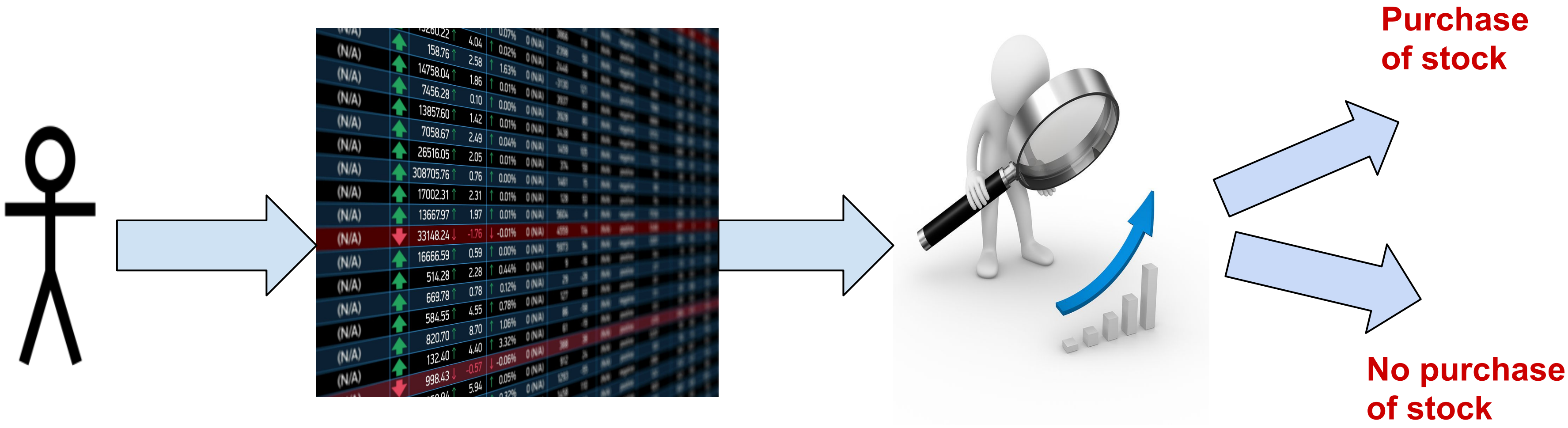
Medical Diagnosis



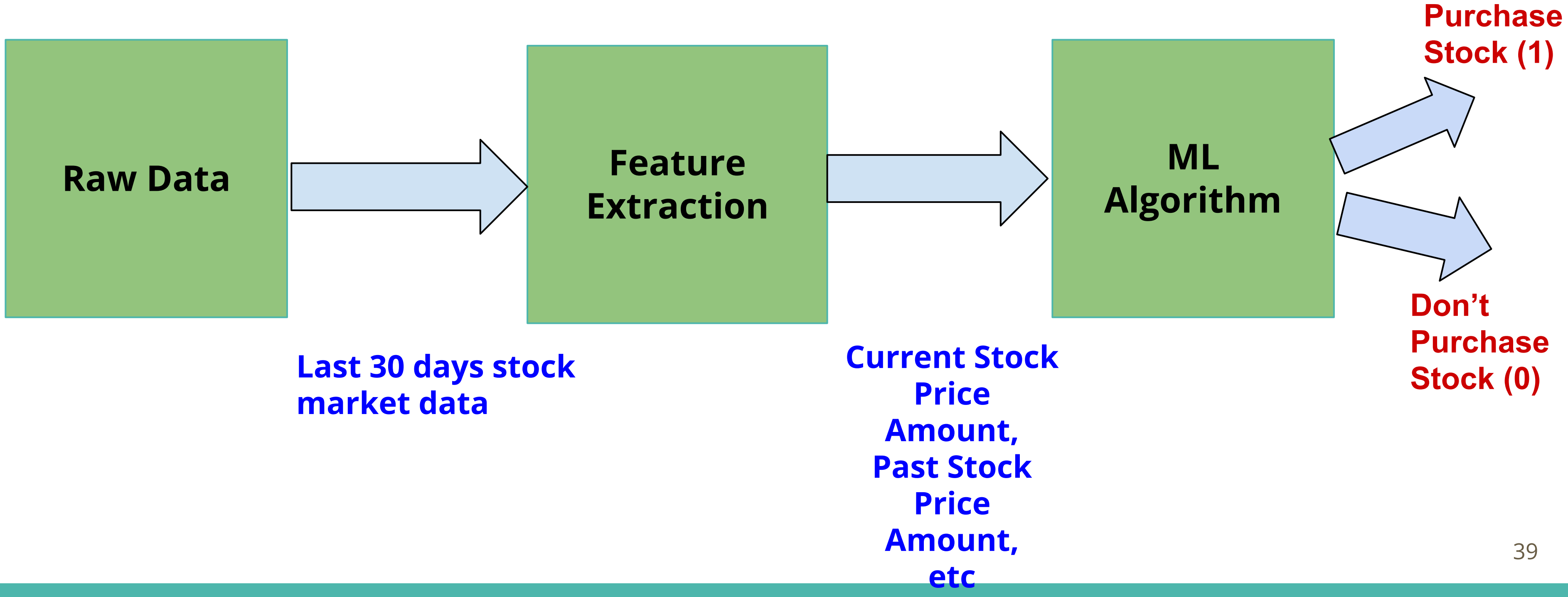
Medical Diagnosis



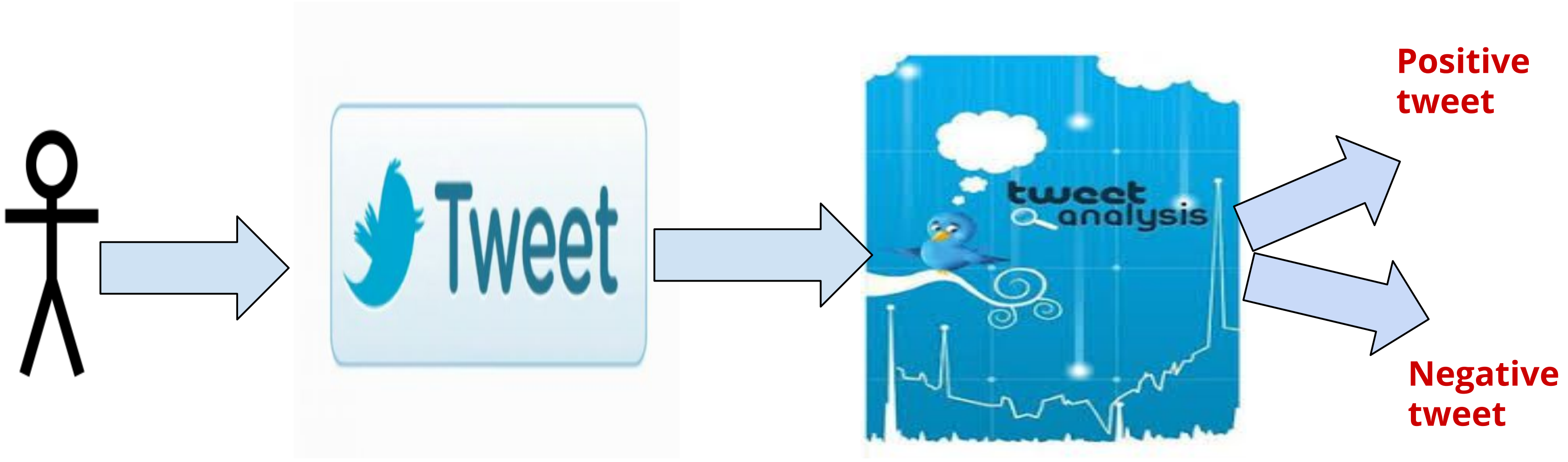
Stock Trading



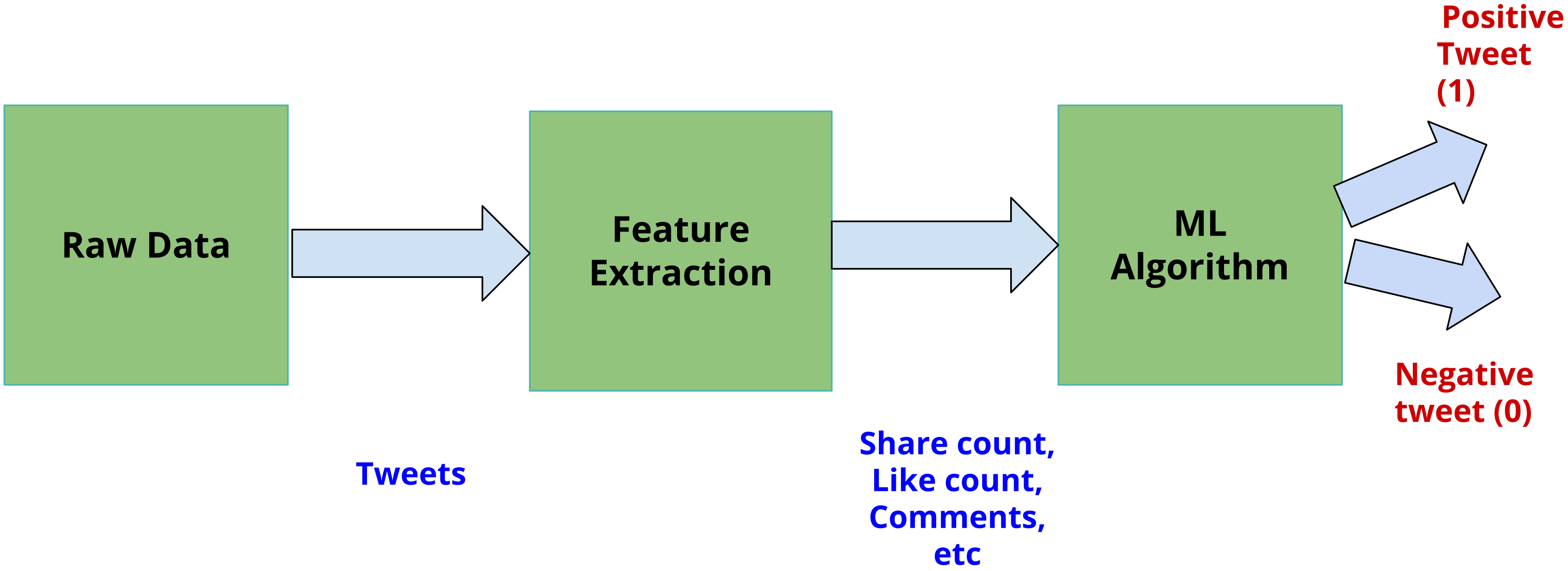
Stock Trading



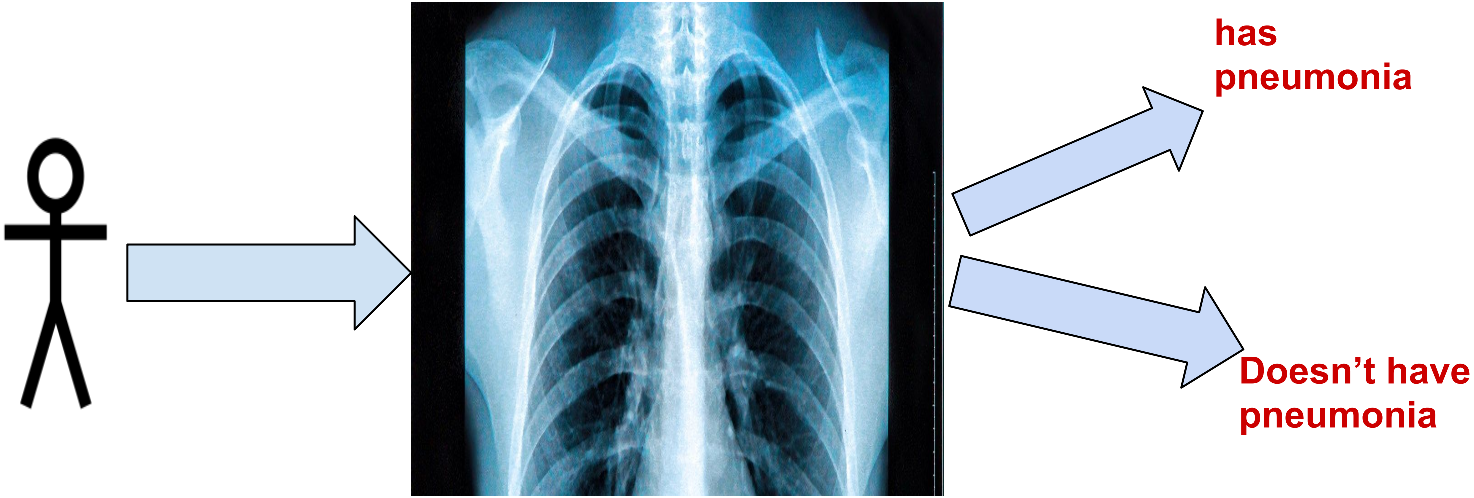
Sentiment Analysis



Sentiment Analysis

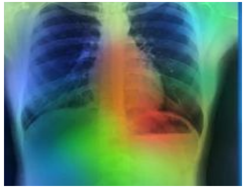


Disease Confirmation



Disease Confirmation

Raw Data → Feature Extraction → ML Algorithm



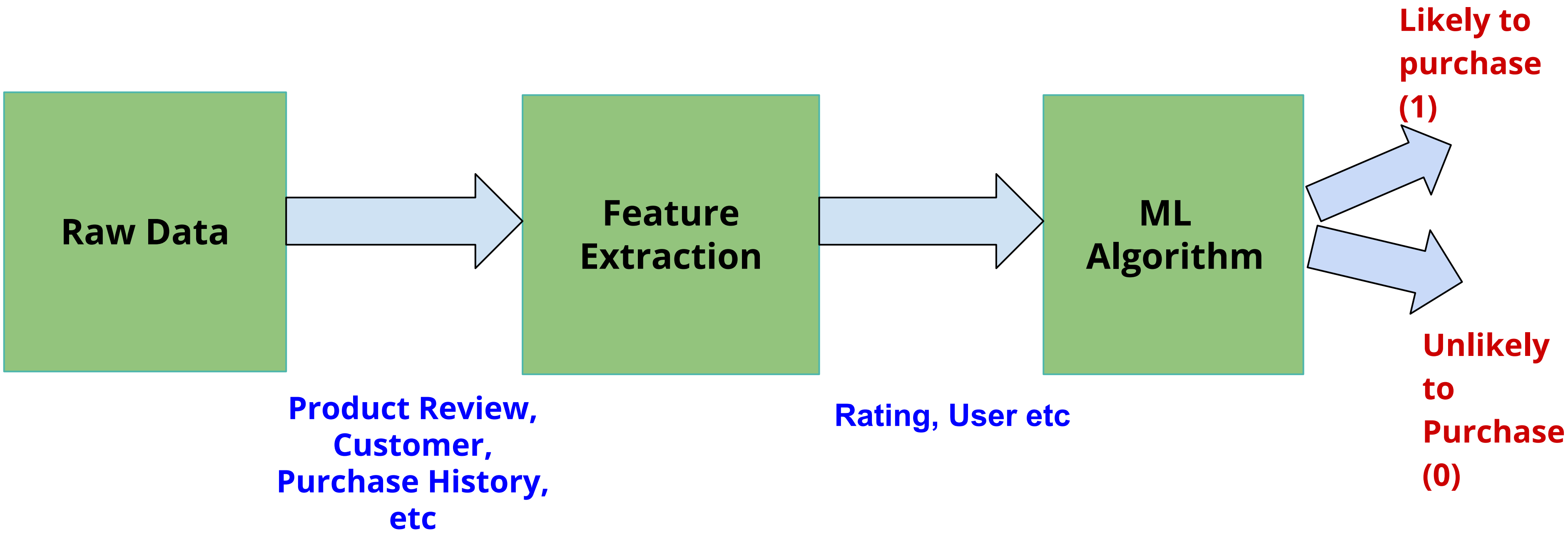
Has Pneumonia (1)

Doesn't have Pneumonia (0)

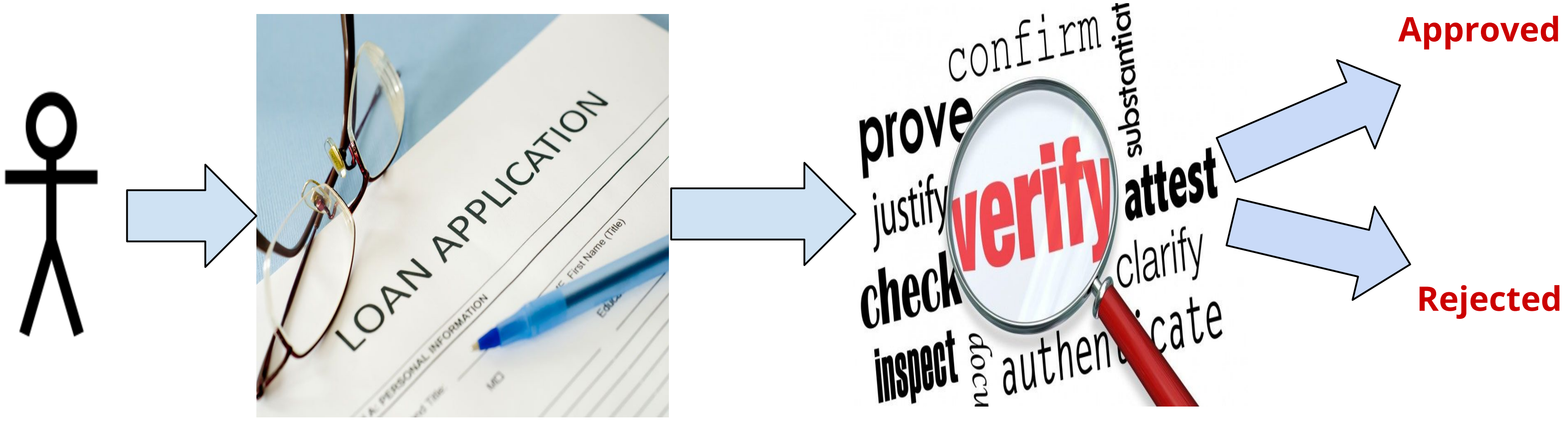
Roughness of pixels in image,
Unevenness, etc.

Product Recommendation

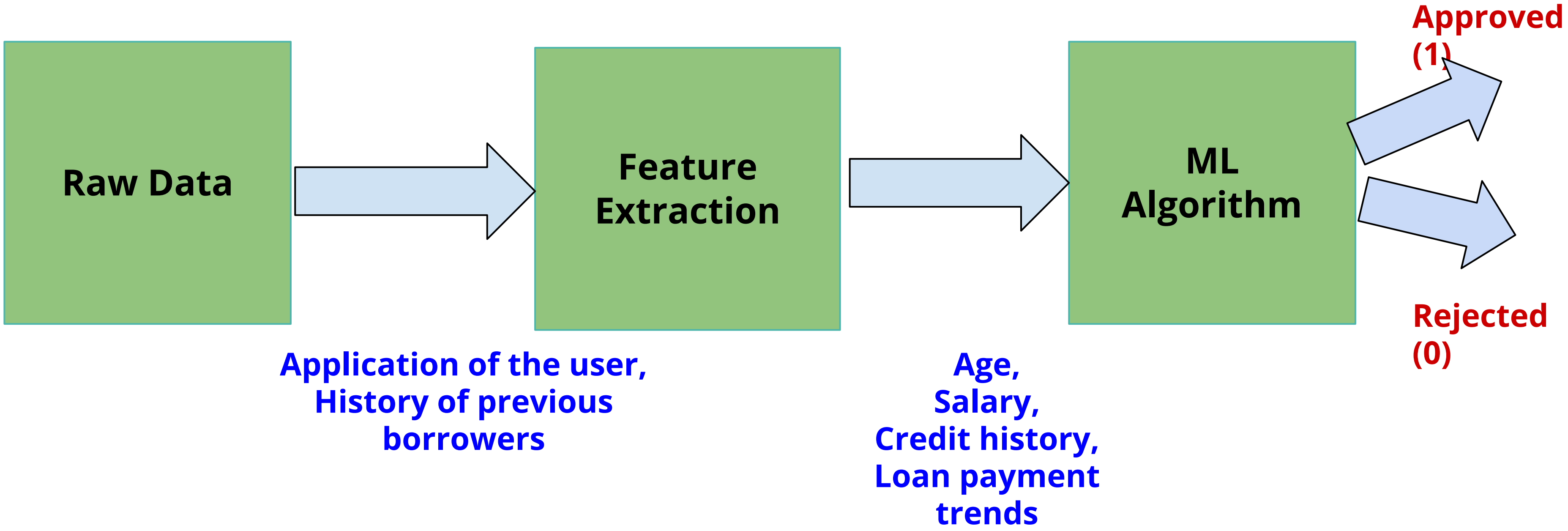
Product Recommendation



Loan Approval



Loan Approval



Face Recognition

Bahubali

Not Bahubali

Bahubali

Raw Data

Images

Feature Extraction

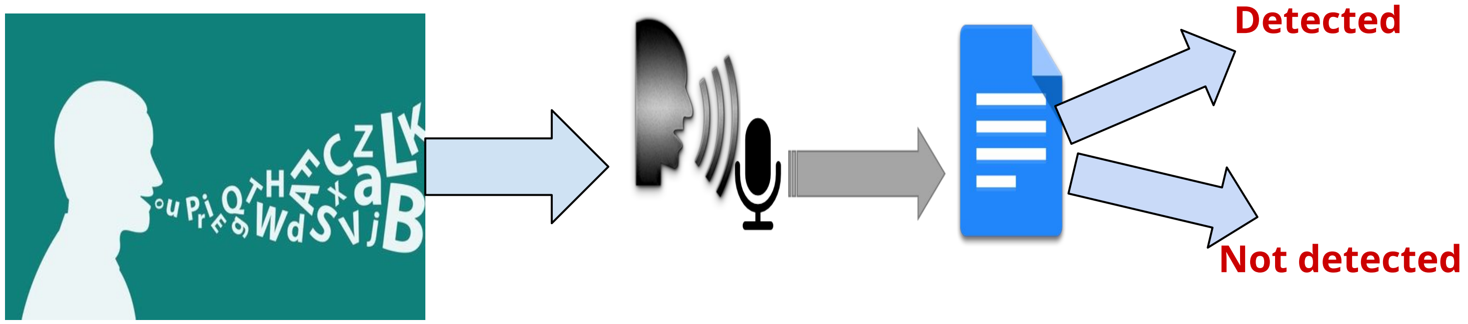
Pixels, etc

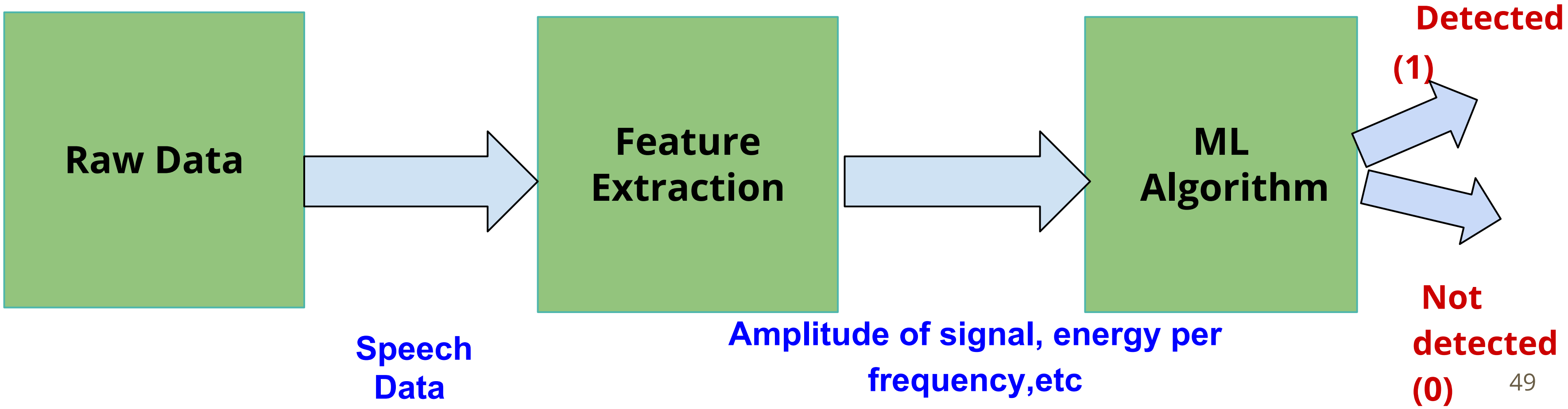
ML Algorithm

(1)

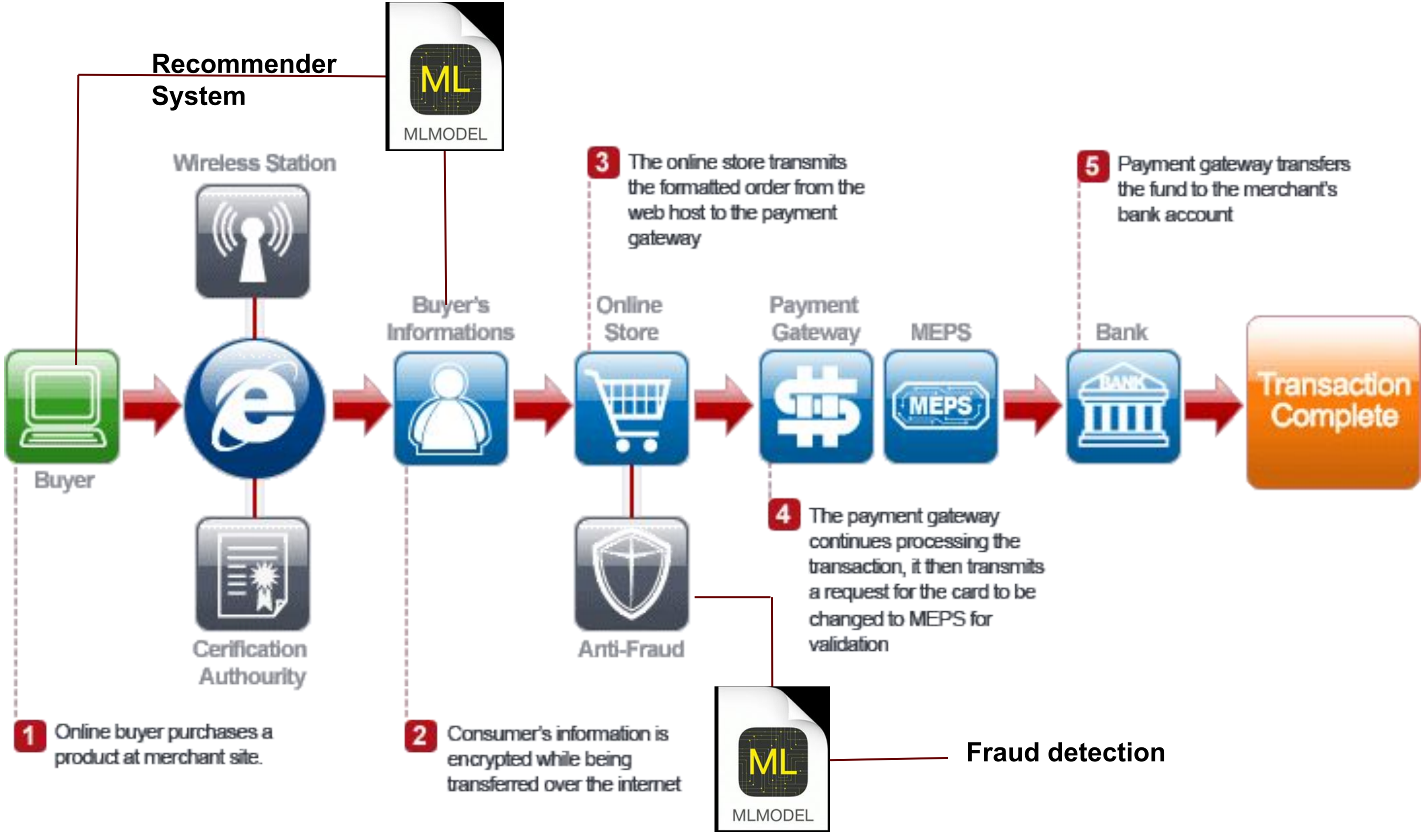
Not Bahubali (0)

Speech Recognition





ML is the "intelligent" block in a large software system



A bit more formal look

The Machine Learning Framework

- Apply a prediction function to a feature representation of the "sample" to get the desired output:

f() = "apple"

f() = "tomato"

f() = "cow"

The Machine Learning Framework

$$Y = f(x)$$

Output — **prediction function** — **feature representation**

Training: given a training set, estimate the prediction function **f** by minimizing the prediction error

Testing: apply **f** to never before seen test example x and output predicted value **y = f(x)**

The underlying abstraction

$$y = f(\mathbf{x})$$

- What are x, y for spam detection?
- What are x, y for image classification?
- What are x, y for sentiment analysis?
- What are x, y for <insert your problem here>?

The underlying abstraction

$$y = f(\mathbf{x})$$

- What are f, x, y for the classification tasks?
- X is often a vector (column matrix).
- Y is either 0 or 1 (binary) or {1,2,...,p} (multiclass)

The underlying abstraction

$$y = f(w,\mathbf{x})$$

- What is really f()? (*This is what we need to find.)*
- Who gives **w**? *(Data gives.)*
- Who gives the form of f ()? (eg. Quadratic, linear etc.?)

Regression and Time series

Regression

- Predicting a real number is regression

Time series Forecasting

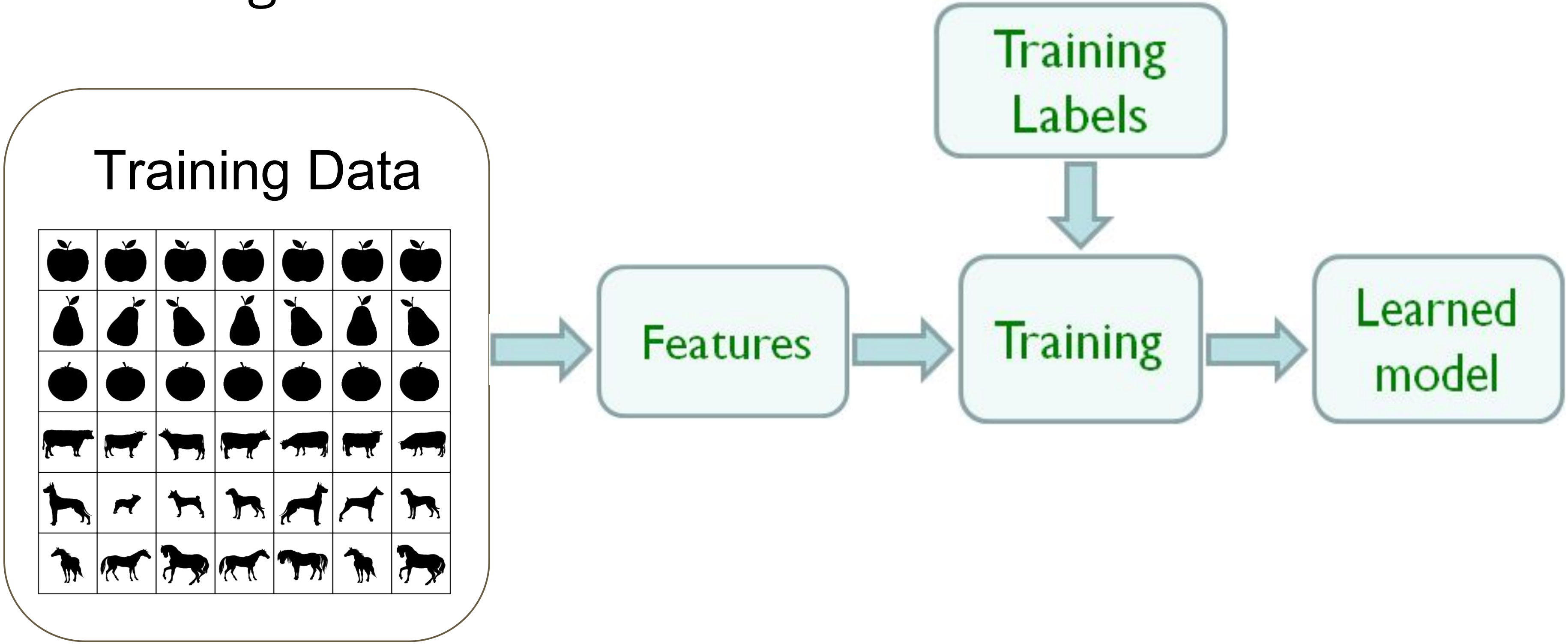
- Predicting based on prior time tagged data

Training and Testing

Creating and Evaluating Models

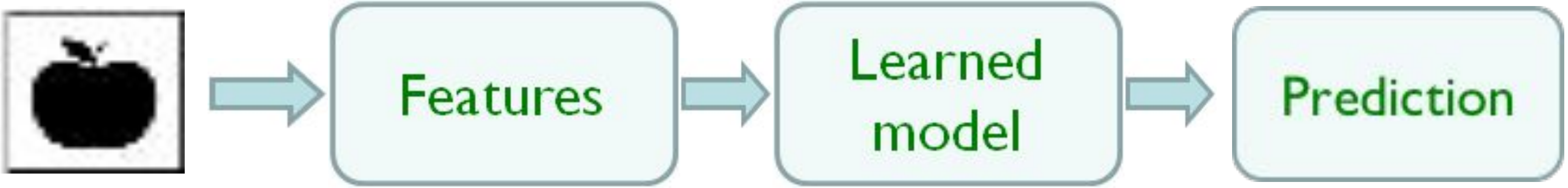
Steps

- Training

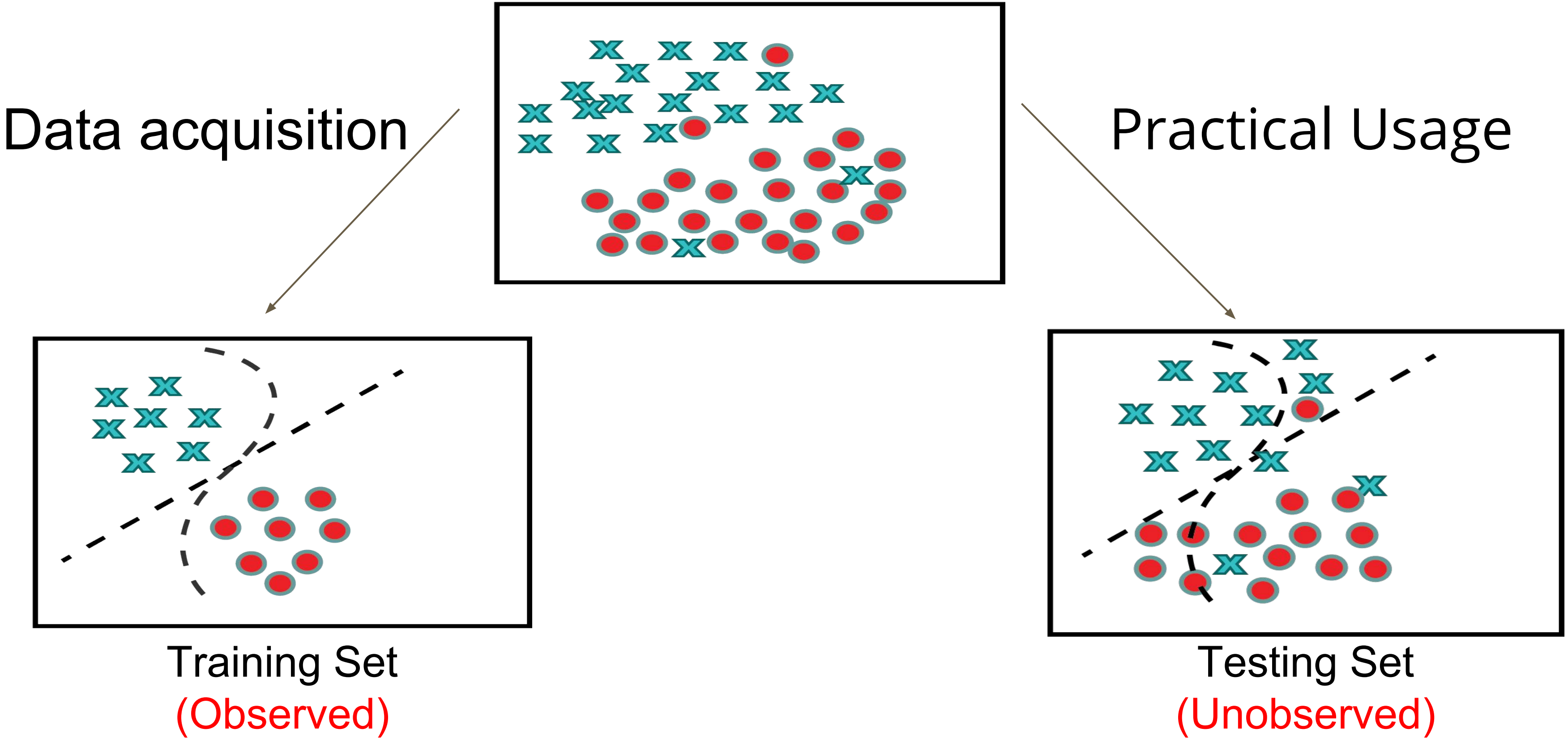


Steps(Cont..)

Testing

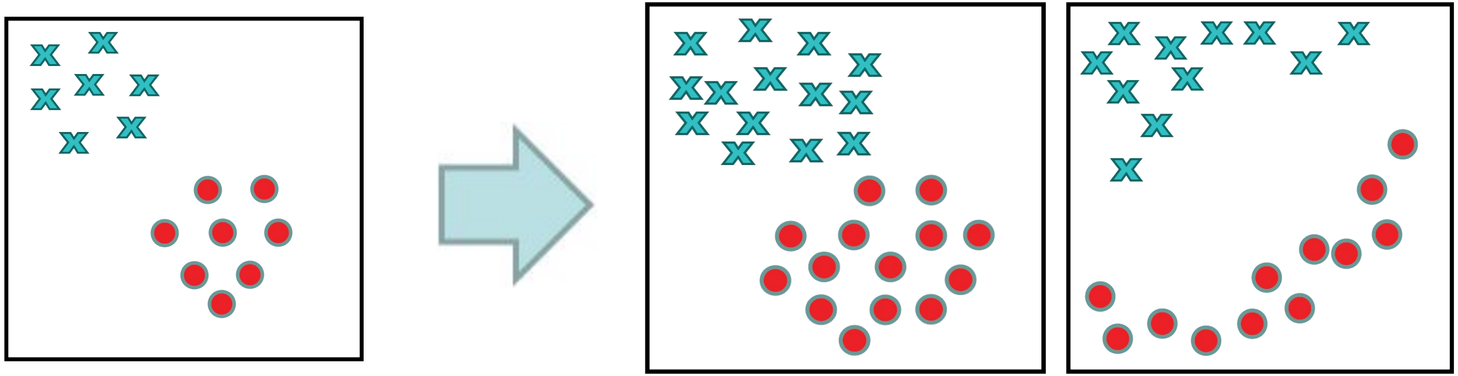


Training and testing



Training and testing

- Training is the process of making the system able to learn
- Assumptions:
  - Training set and testing set come from the same distribution
  - Need to make some assumptions or bias



Two Prominent Learning Paradigms

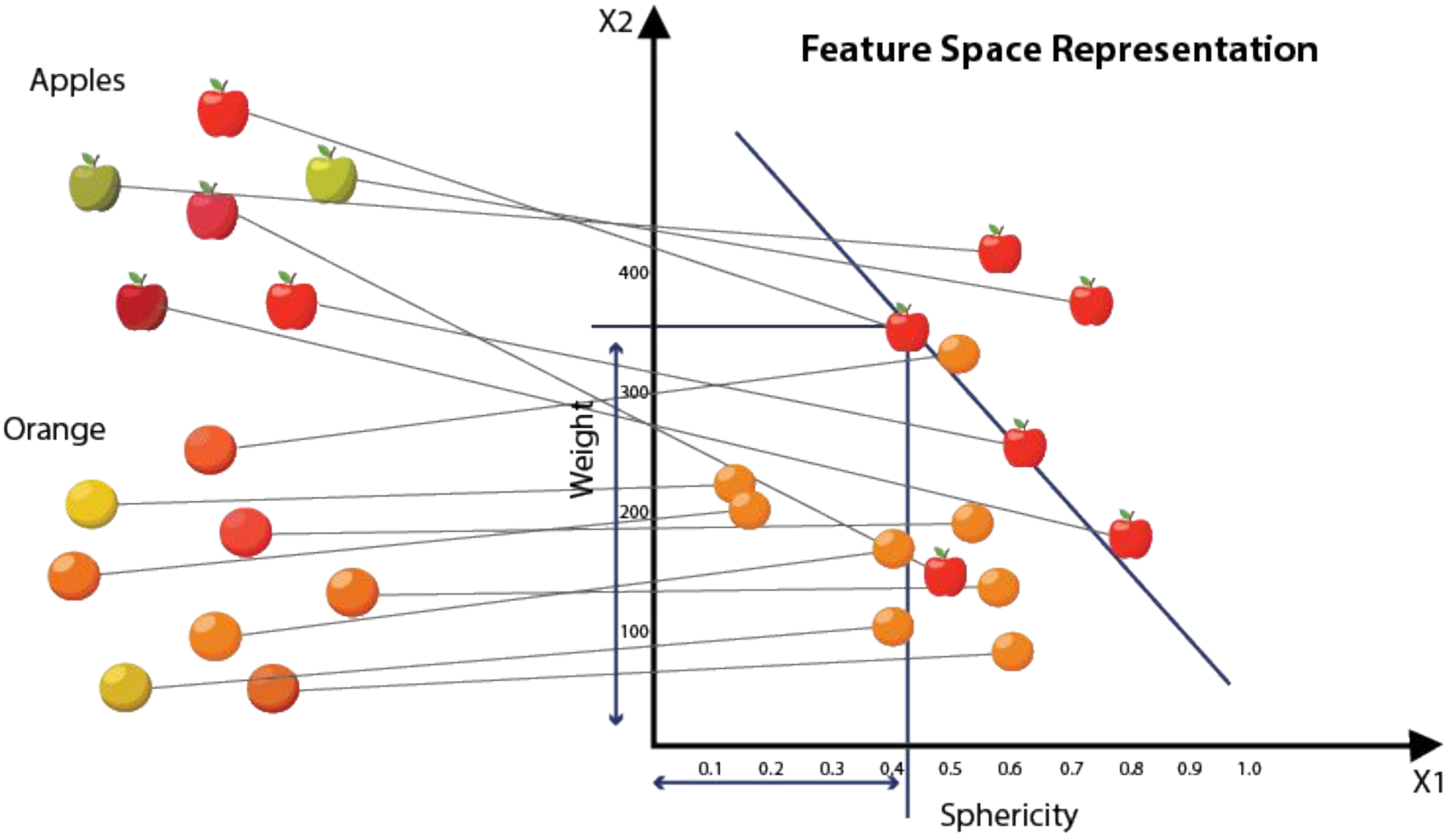
- **Supervised learning :** It is the machine learning task of inferring a function from labeled training data

- **Unsupervised Learning :** Learn patterns from unlabeled data. Often look for a structure

A 'toy' classification problem

- Apples vs Oranges
- We have measured colour, sphericity
- Some labeled data
- Given unlabeled data decide which fruit it is

Visualizing a Sample in 2D



Sample/Point and Representation

A sample is easy to visualize in 2D

$$x = (x, y) or (x_1, y_2) \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and sometime in 3D with some effort $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

And we often need much larger dimensionality in practice

$$x = \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_{100} \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_d \end{bmatrix}$$

Examples of Learned Function or Model

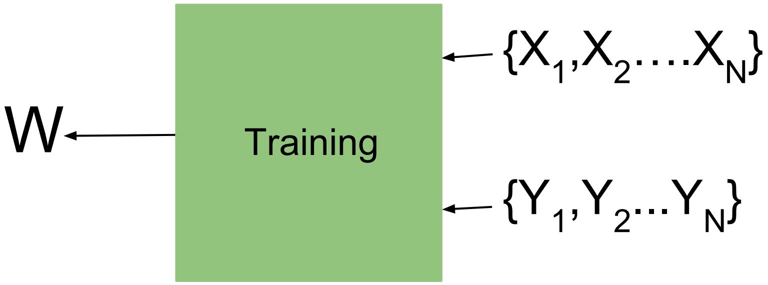
```
bool f(x){
   Z = 0
    for i in 1..d:
         Z = Z + w[i]*x[i]
    if(z>0)
         return "apple"//1
    else
         return "orange"//-1
    }
```

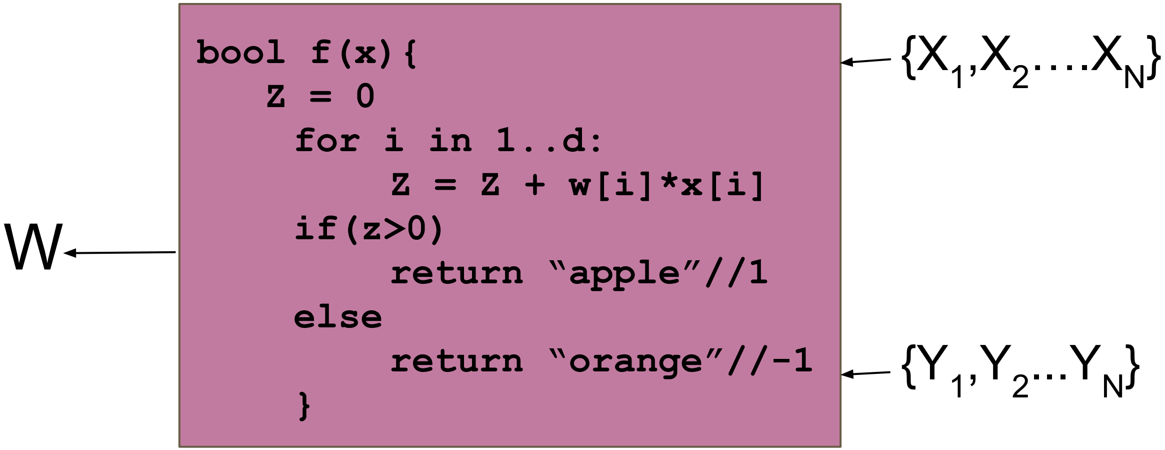
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

$$\mathbf{w} \cdot \mathbf{x} = x_1 \cdot w_1 + x_2 \cdot w_2 + \ldots x_d \cdot w_d$$

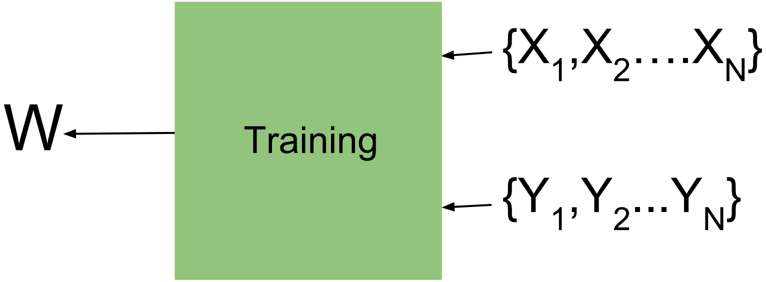
$$f(\mathbf{w}, \mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x})$$

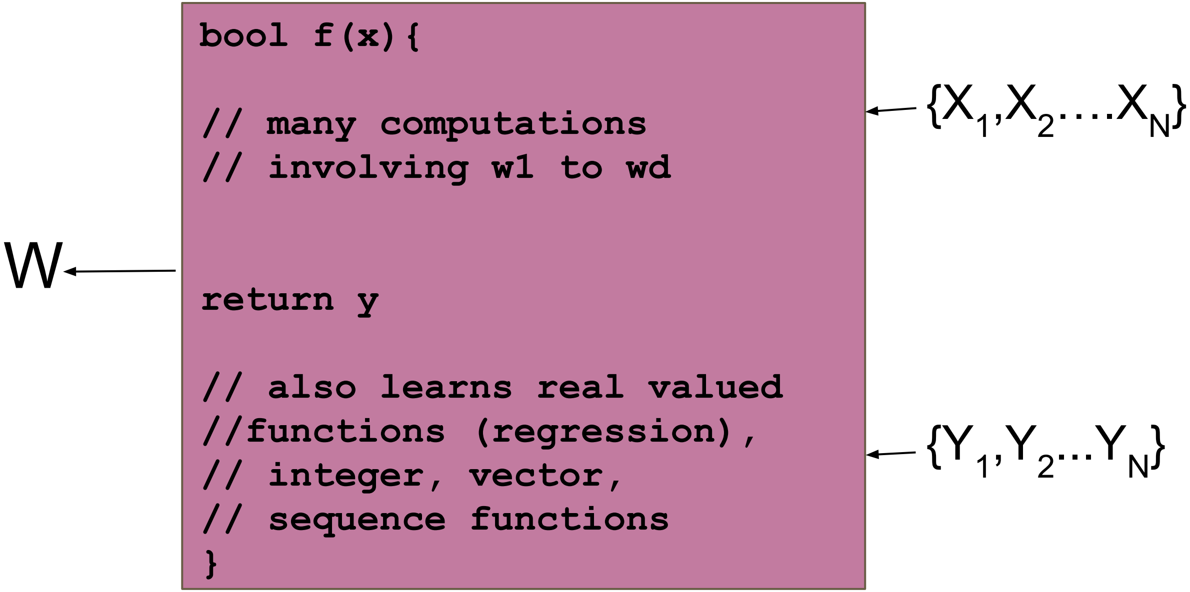
Examples of Learned Model





In general ...


W
Training
$\{X_1, X_2 \dots X_N\}$
$\{Y_1, Y_2 \dots Y_N\}$


bool f(x){
// many computations
// involving w1 to wd
return y
// also learns real valued
//functions (regression),
// integer, vector,
// sequence functions
}
W
$\{X_1, X_2 \dots X_N\}$
$\{Y_1, Y_2 \dots Y_N\}$

Classification Algorithms

KNN, Linear Classifier and Decision tree

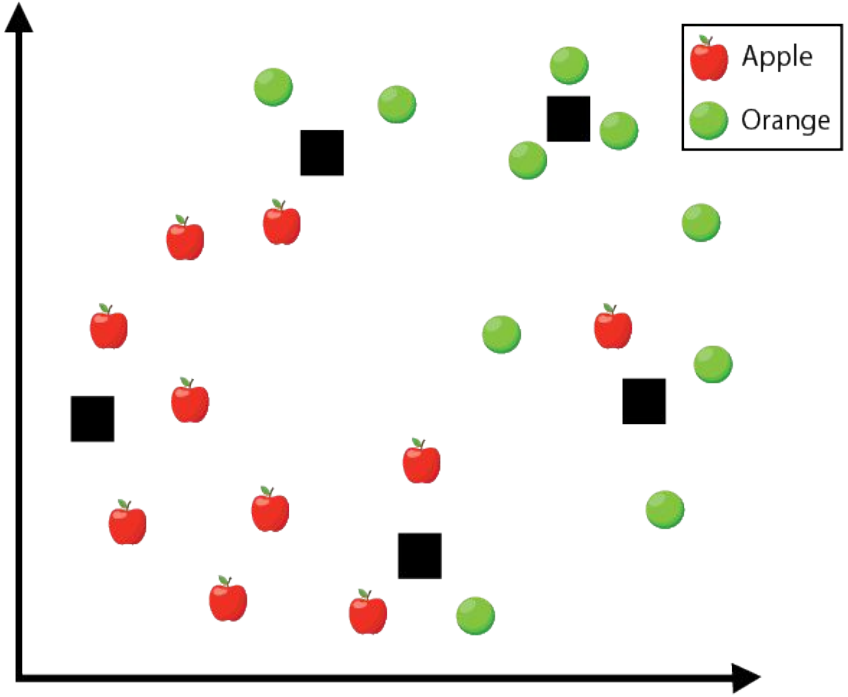
Goal: Assign label for unknown samples

Data:

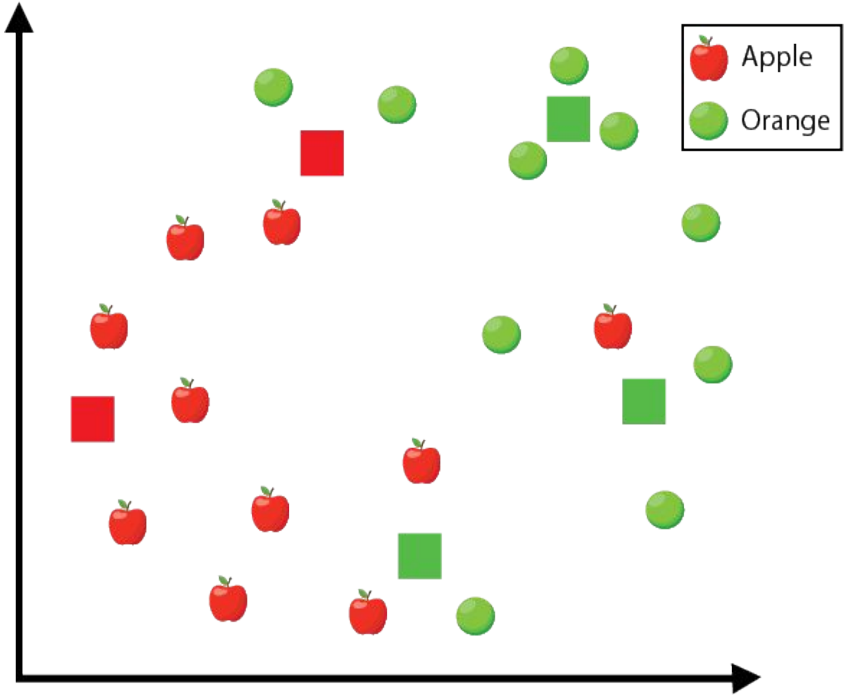
- Training data (10+10)
    - Apples (red) and Oranges (blue)
- Test (5): Unknown label. Black

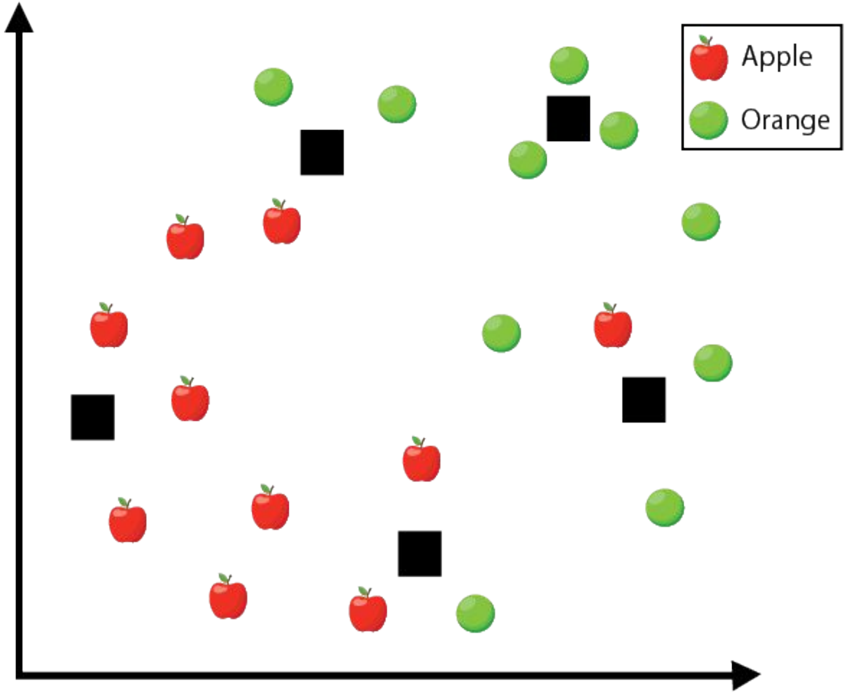
Attributes (assume):

- Sphericity and Color



(Tester, User) knows the truth for unknowns





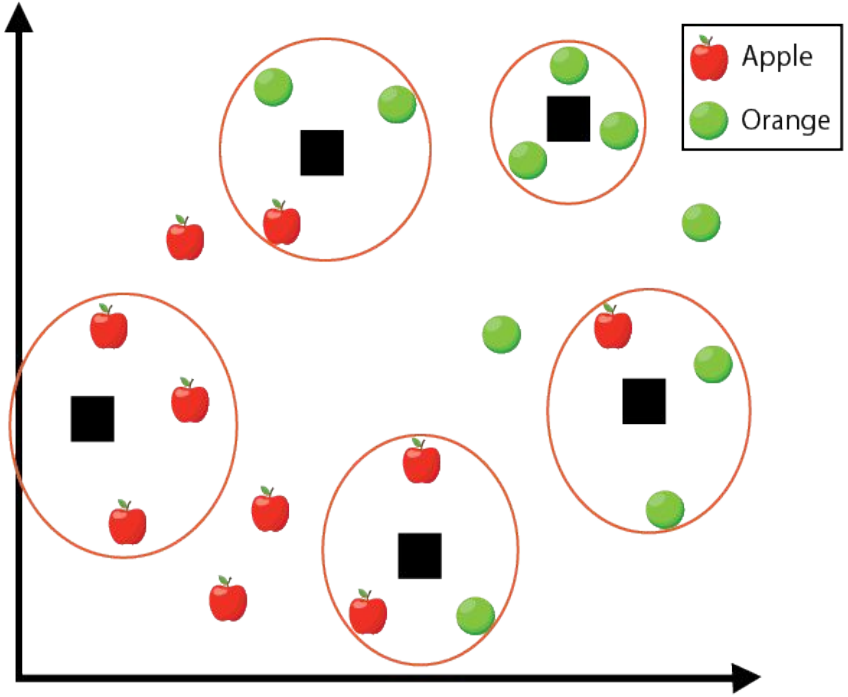
Goal: Assign label for unknown samples

Idea:

- Like people groups

Method:

- Look 3 Nearest Neighbours
- Assign majority label



What is the accuracy?

- 3 out of 5 got correct:
  - Accuracy = 60%
  - Error = 40%
- (A random guess could have given 50%!!)

Comments

- We “assumed” k =3. It can also be 5, 7 or any number (often odd. Why?)
- The data can have many more classes (fruits). Apples, Oranges and Mangoes
- Distances need not be Euclidean. Many other distance functions exists in the literature

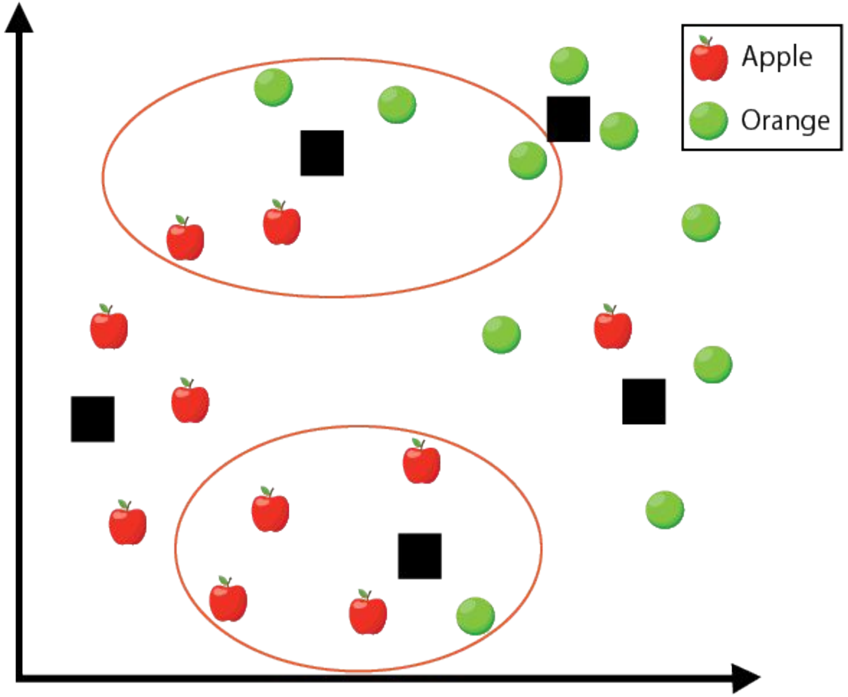
Will another K help? Some times.

Idea:

- Try different k and pick best

Method:

- Try k =5
- (only two shown)
- Accuracy = 80%

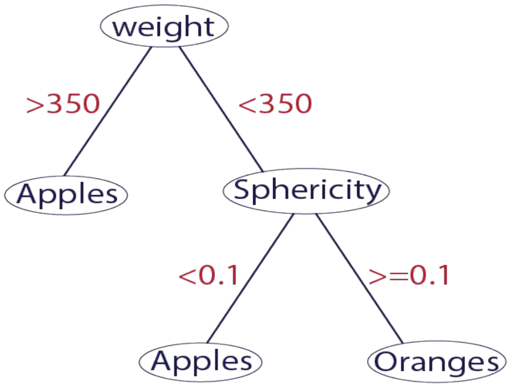


K Nearest Neighbours (KNN)

`Given K, Data and Distance Function`

- **Find the distance from z to all the samples in X**
- **Identify the K-nearest neighbours (smallest distances) and their class labels**
- **Classify z as the majority label from the K-nearest neighbours**

Decision Tree




```
if (weight is >350)
        it is an apple
else
        if (sphericity > 0.1)
               it is an orange
         else
               it is an apple
```

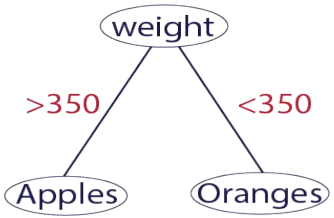
Decision Tree

- Splitting Criterion 1: Based on color?
- Splitting Criterion 2: Based on Sphericity?

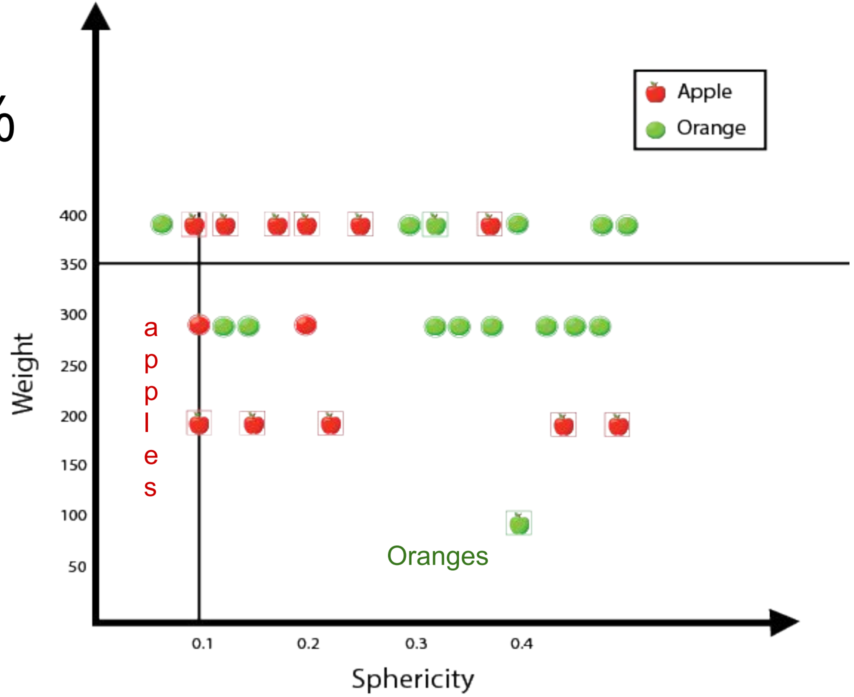
Pseudo code for Decision Tree

- Place the best attribute of the dataset at the root of the tree
- Split the training set into subsets
- Repeat step 1 and step 2 on each subset until you find leaf nodes in all the branches of the tree

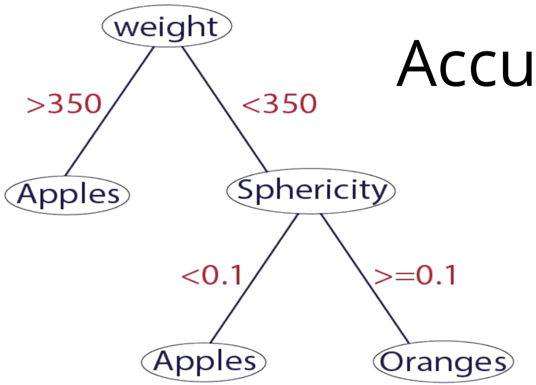
Decision tree at level-1



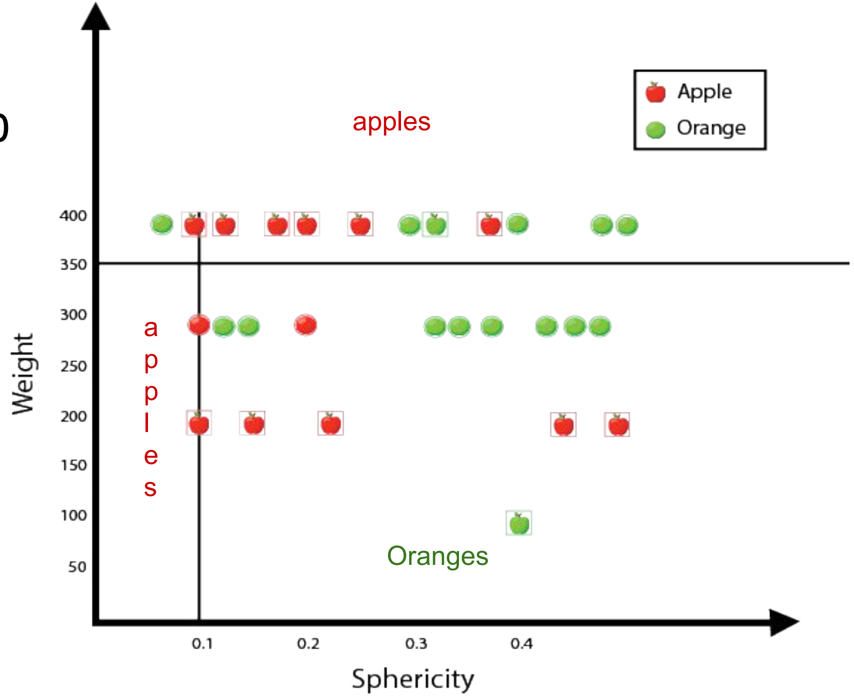
Accuracy = 29.6%



Decision tree at level-2



Accuracy = 59.6%



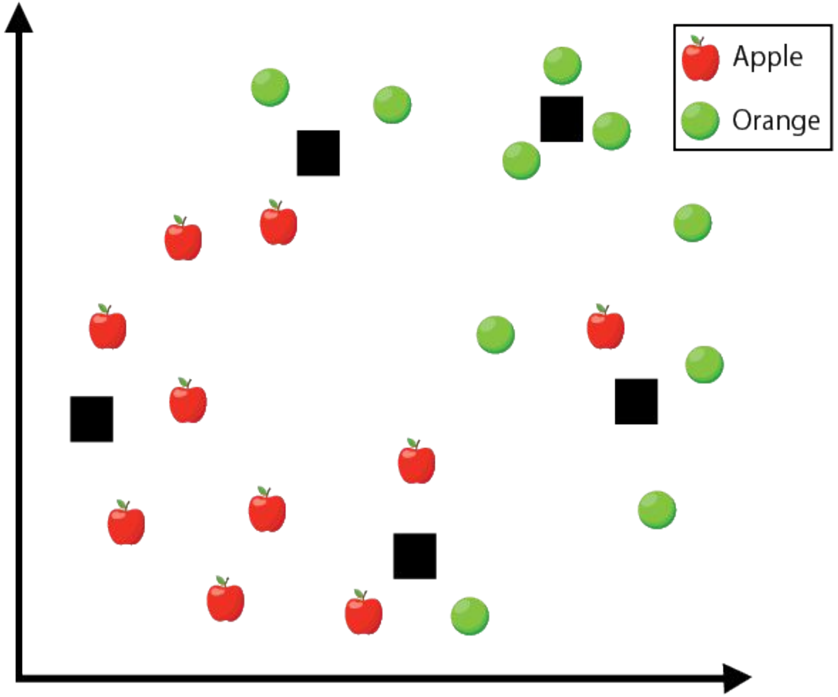
Simple Linear Classifier

Data:

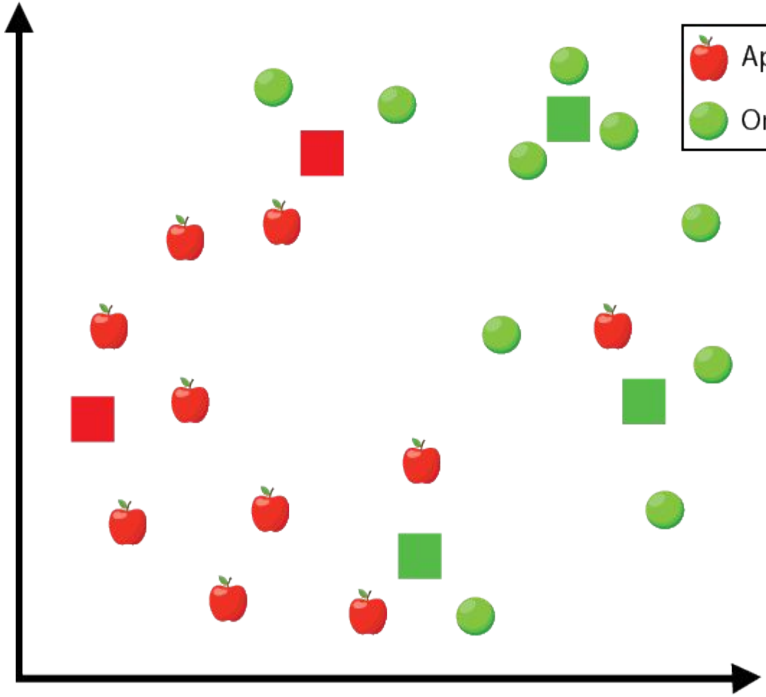
- Apples (red) and Oranges (blue)
- Test: Unknown label black

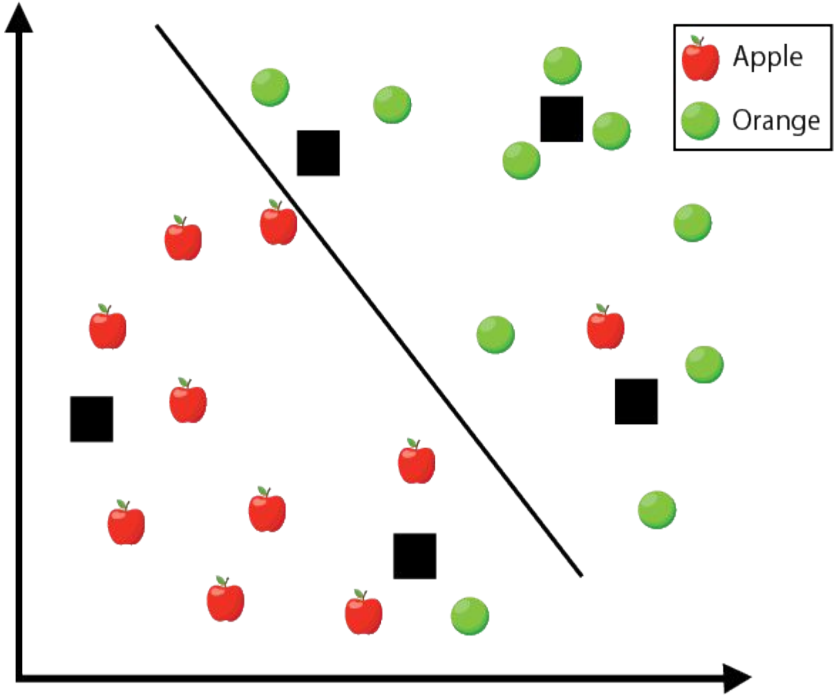
Goal:

- Find a line that can separate



Best Solution: $2x_1 - 4x_2 - 5 = 0$

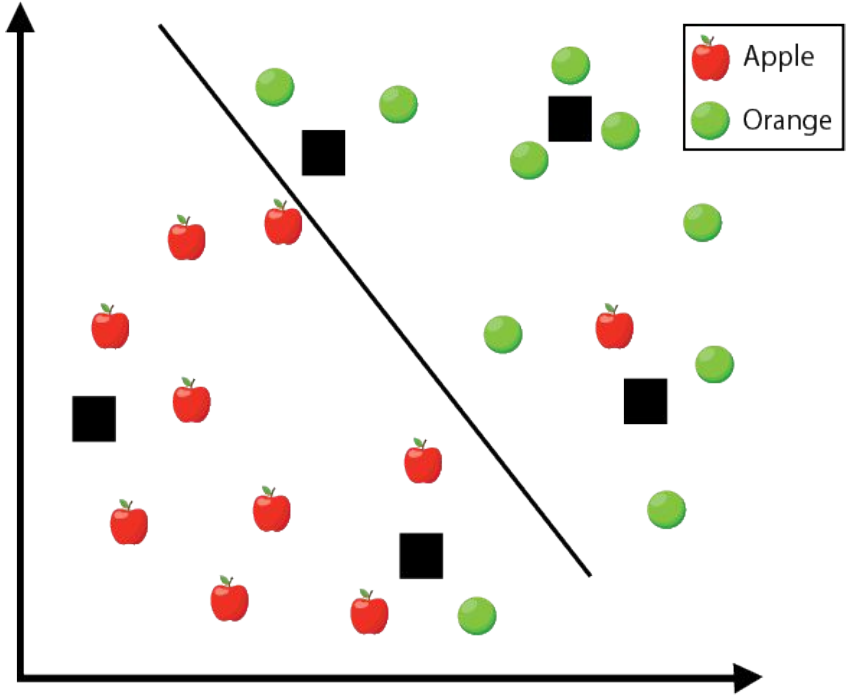




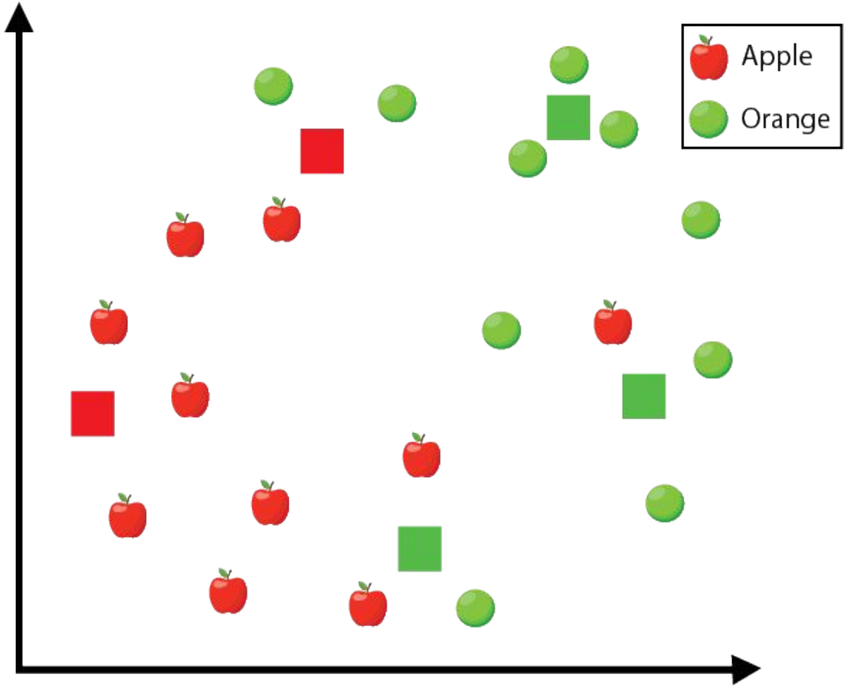
What is the error/accuracy?

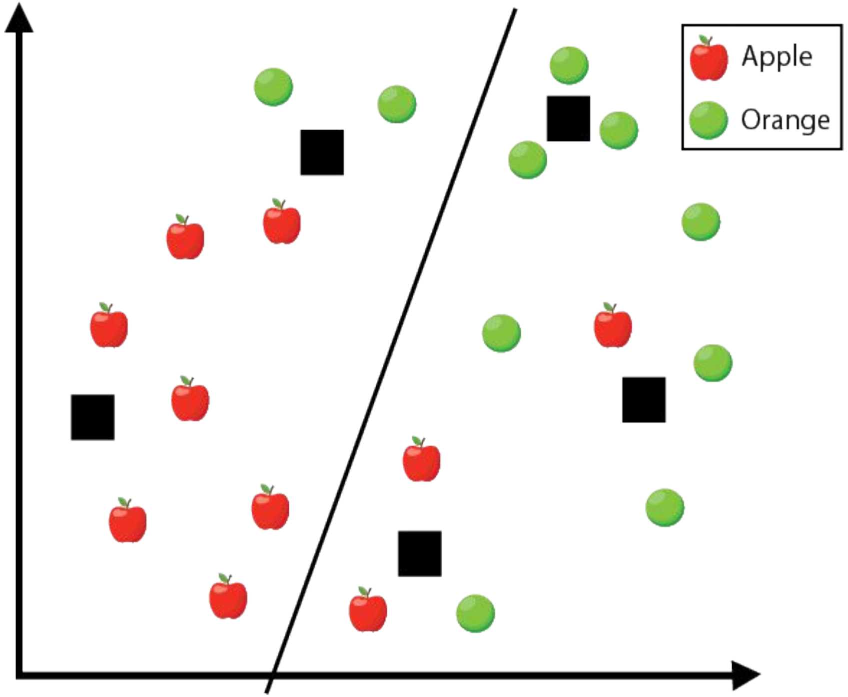
On the test data: 40% error

On the training data (resubmission error): 10% error



Best Solution: $x_1 + 2x_2 + 2 = 0$

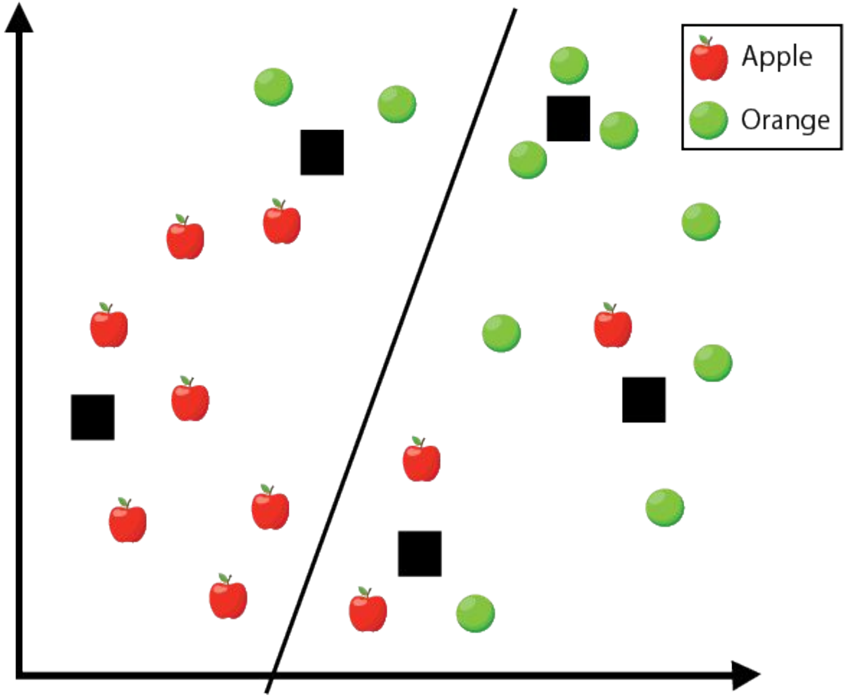




What is the error/accuracy?

On the test data: 0% error

On the training data (resubmission error): 25% error



Simple method to predict/test/evaluate

```
bool f(x){
    Z = 0
    for i in 1..d:
        Z = Z + w[i]*x[i]
    if(z>0)
        return "apple"//1
    else
        return "orange"//-1
    }
```

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

$$\mathbf{w} \cdot \mathbf{x} = x_1 \cdot w_1 + x_2 \cdot w_2 + \ldots x_d \cdot w_d$$

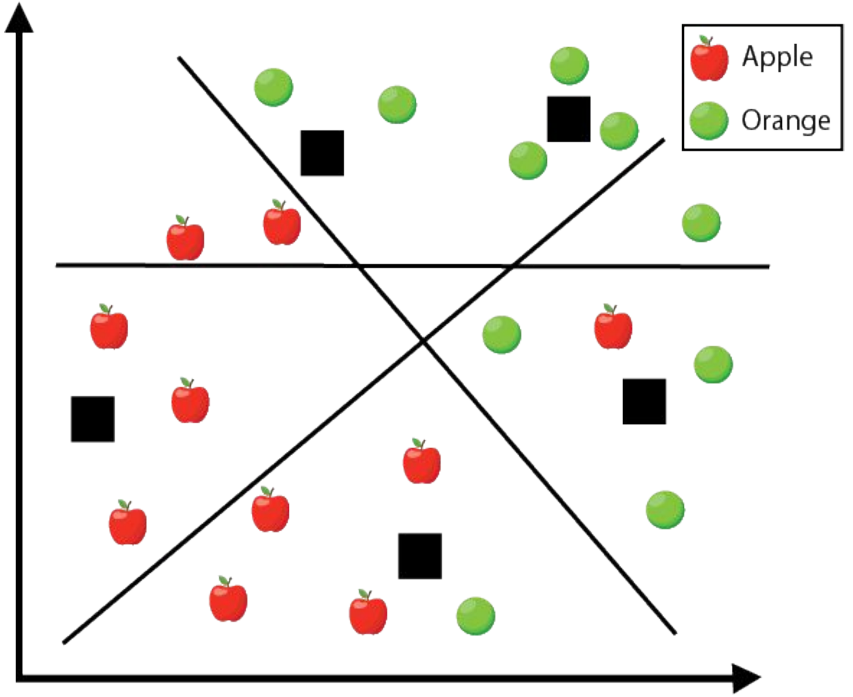
$$f(\mathbf{w}, \mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x})$$

What is the “learning” problem?

Find the best line given (only) the training data

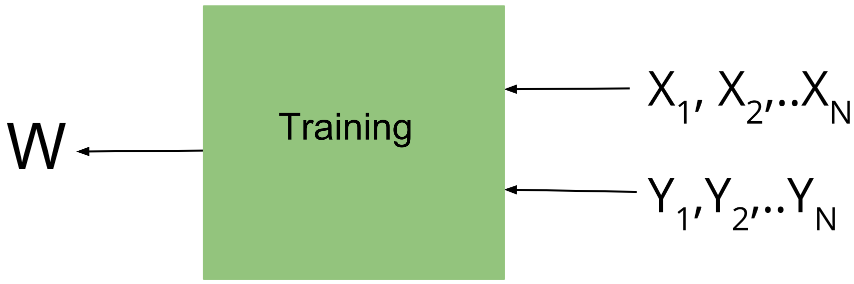
Find w_1, w_2, w_0 in

$w_1.x_1 + w_2.x_2 + w_0 = 0$



Data gives w through the "training" process

```
bool f(x){
    Z = 0
    for i in 1..d:
        Z = Z + w[i]*x[i]
    if(z>0)
        return "apple"//1
    else
        return "orange"//-1
    }
```



Comments

Comments - KNN

- KNN: n distance computations each of O(d)

    - **Too many operations**
    - **Needs all the samples at test time; Storage intensive**

n : Total number of samples
d : Total number of features or dimensionality

Comments - Linear Classifier

- Linear Classifier: $(d - 1)$ **additions;** d **multiplications**
- **Simple at test time.**
- **Needs an additional "offline" training to find w**

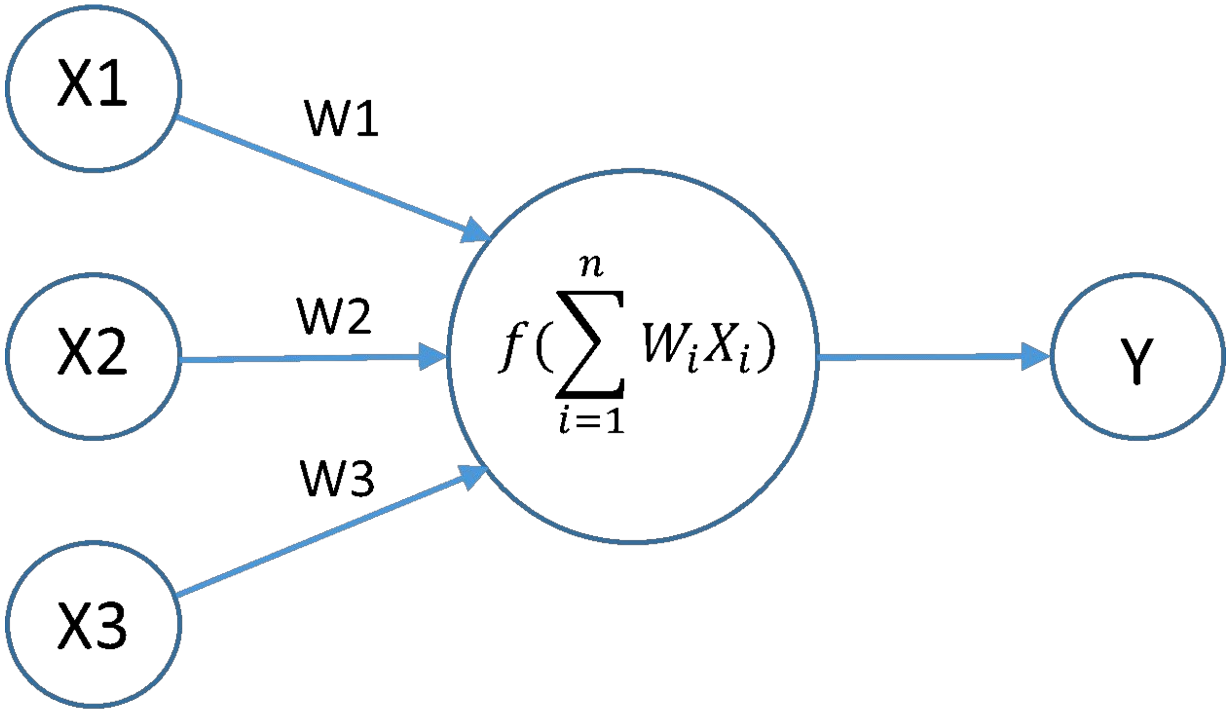
n : Total number of samples
d : Total number of features or dimensionality

Comments - Decision Tree (ID3 alg.)

- Decision Tree:
    - Simple Human understandable solution
- Training: Finding the best tree
- Testing: (loosely) d attributes provided tests to allow the instances to be differentiated into required bins

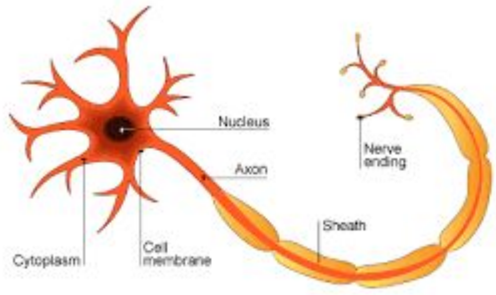
n : Total number of samples
d : Total number of features or dimensionality

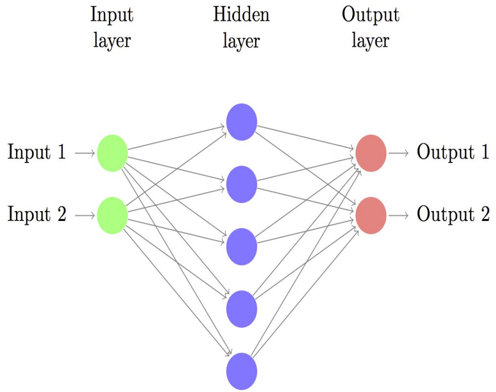
Understanding Linear Classifier as “Neuron”



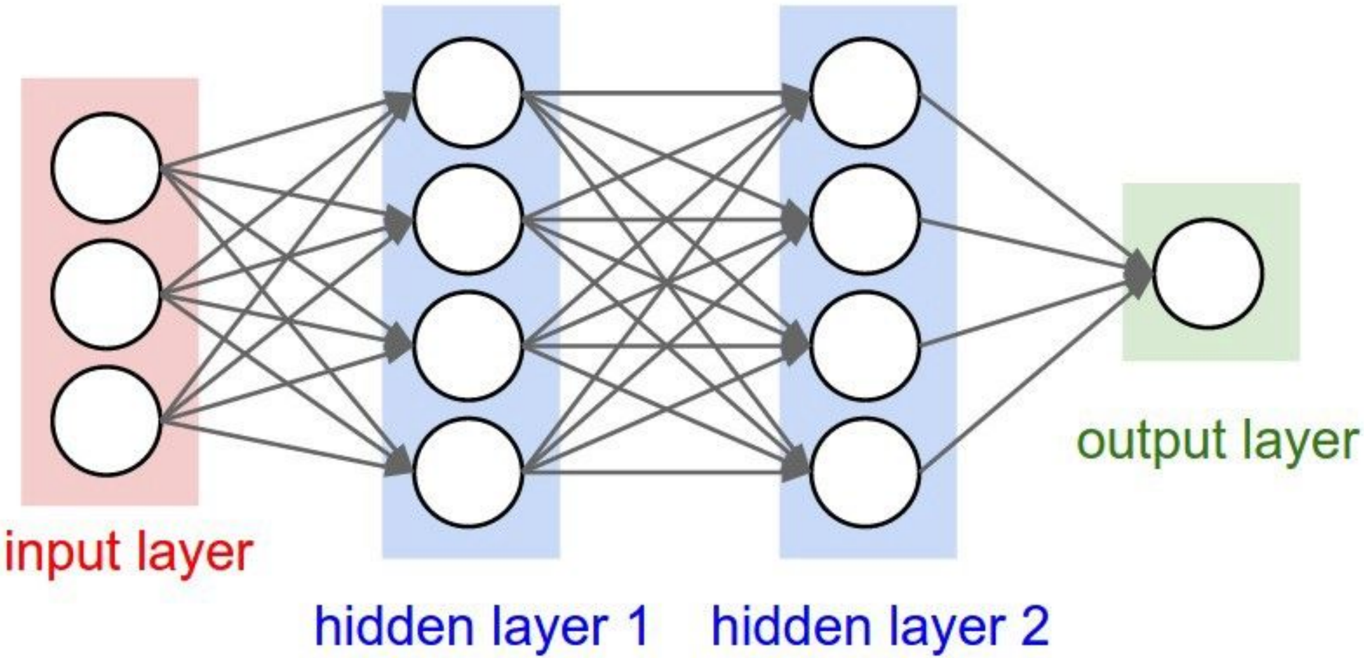
Neural Networks

- Biologically inspired networks.
- Complex function approximation through composition of functions.
- Can learn arbitrary Nonlinear decision boundary





A Peep into Deep Neural Networks



Thanks!

Questions?