

Introduction



Machine Learning

Introduction

Welcome

Examples of Machine Learning (ML):

- **DATABASE MINING** : large datasets available from growth of automation/web (e.g. web click data/click-stream data, medical records, computational biology, engineering)
- **APPLICATIONS THAT CANNOT BE PROGRAMMED BY HAND** : e.g. autonomous helicopters, handwriting recognition, natural language processing (NLP), computer vision.
- **SELF-CUSTOMISING PROGRAMS** : e.g. Amazon and Netflix recommendations.
- **UNDERSTANDING HUMAN LEARNING** : brain, real Artificial Intelligence (AI)

What is ML?

Even among ML practitioners, there isn't a well accepted definition of what is and what isn't ML. We will illustrate a couple of examples of the ways people have tried to define it.

- **Arthur Samuel (1959):**

'ML is the field of study that gives computers the ability to learn without being explicitly programmed'.

This is an older, informal definition.

Samuel (1901-1990) was an American pioneer in the field of computer gaming and AI. Samuel got famous at the end of the 1950's when he wrote a **checkers playing program**. The amazing thing of this checkers playing program was that Arthur Samuel himself wasn't a very good checkers player. What he did was programming maybe tens of thousands of games against himself, and by watching what sorts of board positions tended to lead to wins and what sort of board positions tended to lead to losses, the checkers playing program learned over time what are good board positions and what are bad board positions. And eventually learn to play checkers better than the Arthur Samuel himself was able to.



- **Tom Mitchell (1998)**

'A well posed learning problem is defined as follows: a computer program is said to learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .'

Tom Michael Mitchell is an American computer scientist and Professor at the Carnegie Mellon University.

For the checkers playing examples, the experience E would be the experience of having the program play tens of thousands of games itself. The task T would be the task of playing checkers, and the performance measure P will be the probability that wins the next game of checkers against some new opponent.



Question: Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam. What is the task T in this setting?

Classify emails as spam or not spam.

"A computer program is said to *learn* from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."

Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam. What is the task T in this setting?

- ☒ Classifying emails as spam or not spam. T
- ☐ Watching you label emails as spam or not spam. E
- ☐ The number (or fraction) of emails correctly classified as spam/not spam.
- ☐ None of the above—this is not a machine learning problem. P

There are many types of ML algorithms, the main types being:

- **SUPERVISED LEARNING:** we are going to teach the computer how to do something
- **UNSUPERVISED LEARNING:** we are going to let the computer learn it by itself

Others: **REINFORCEMENT LEARNING, RECOMMENDER SYSTEMS.**

Supervised Learning

In supervised learning, we are given a data set and **already know what our correct output should look like**, having the idea that there is a relationship between the input and the output.

Supervised learning problems are categorized into "**regression**" and "**classification**" problems.

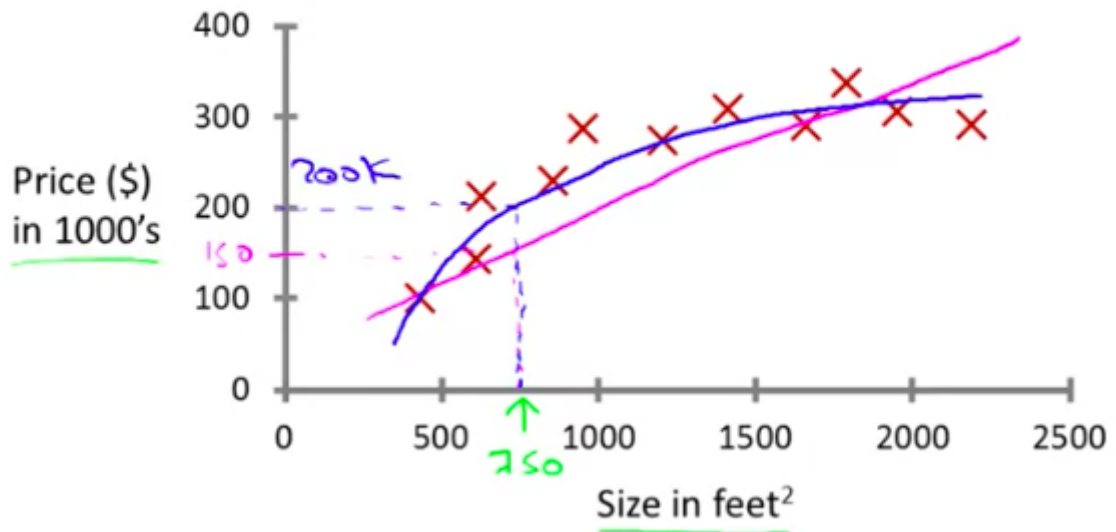
- In a **regression problem**, we are trying to predict results within a **continuous output**, meaning that we are trying to map input variables to some continuous function.
- In a **classification problem**, we are instead trying to predict results in a **discrete output**. In other words, we are trying to map input variables into discrete categories.

Example 1:

Given data about the size of houses on the real estate market, try to predict their price. Price as a function of size is a continuous output, so this is a regression problem.

For example, using a ML algorithm we might be able to fit a straight line into the data and predict a house's price based on its size in squared feet. However, you might use a different learning algorithm giving a better performance: for example one that fits a second order polynomial (i.e. quadratic) function to the data.

Housing price prediction.

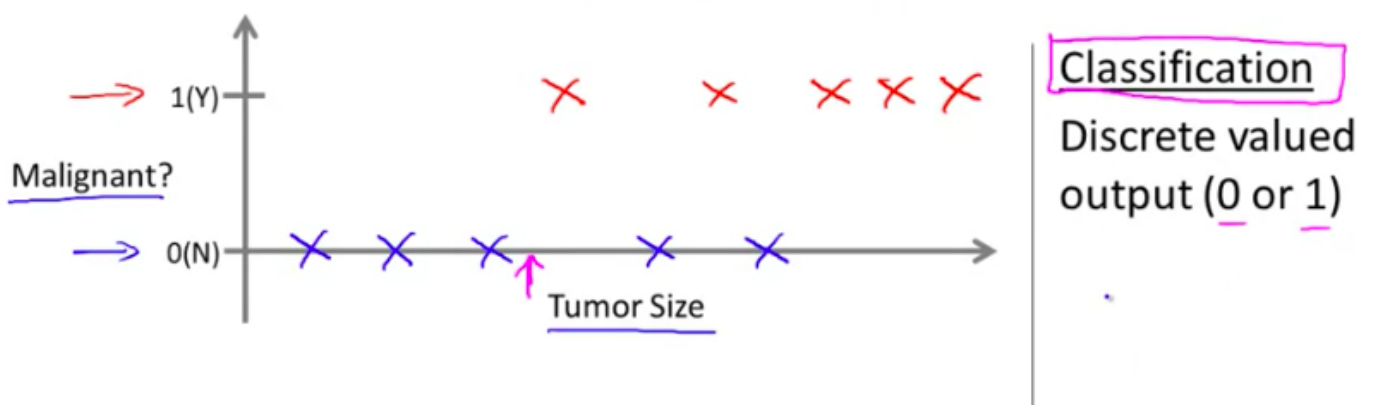


We could turn this example into a classification problem by instead making our output about whether the house "sells for more or less than the asking price." Here we are classifying the houses based on price into two discrete categories.

Example 2:

- (a) Regression - Given a picture of a person, we have to predict their age on the basis of the given picture
- (b) Classification - Given a patient with a tumor, we have to predict whether the tumor is malignant or benign.

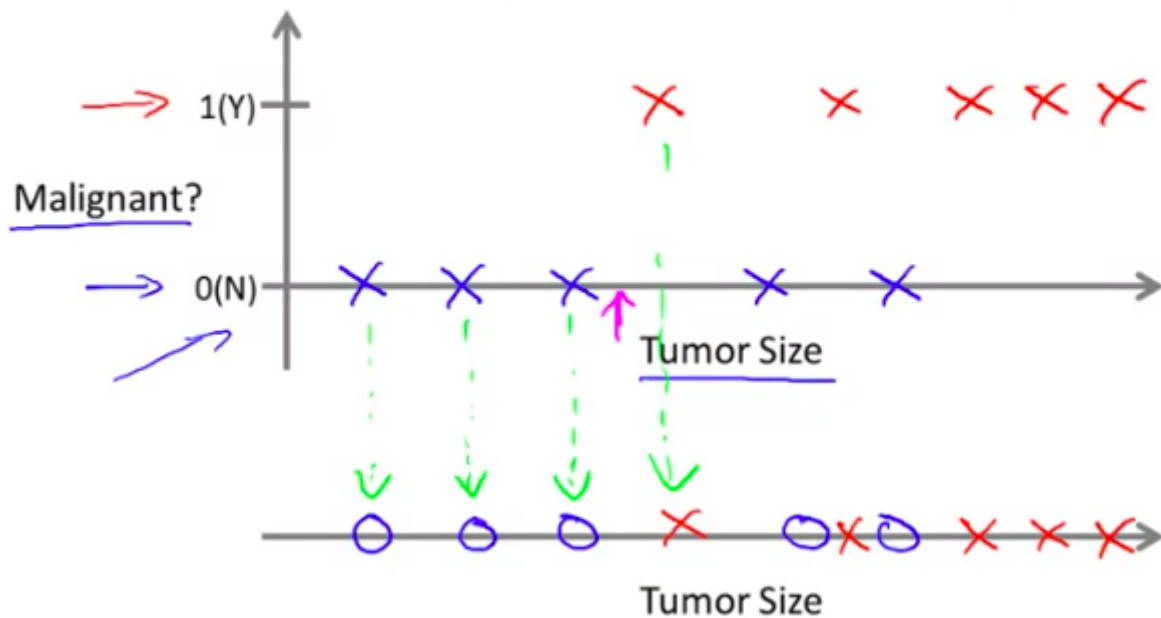
Breast cancer (malignant, benign)



It turns out that in classification problems you might have more than 2 possible outcomes (e.g. 3 types of breast cancer)

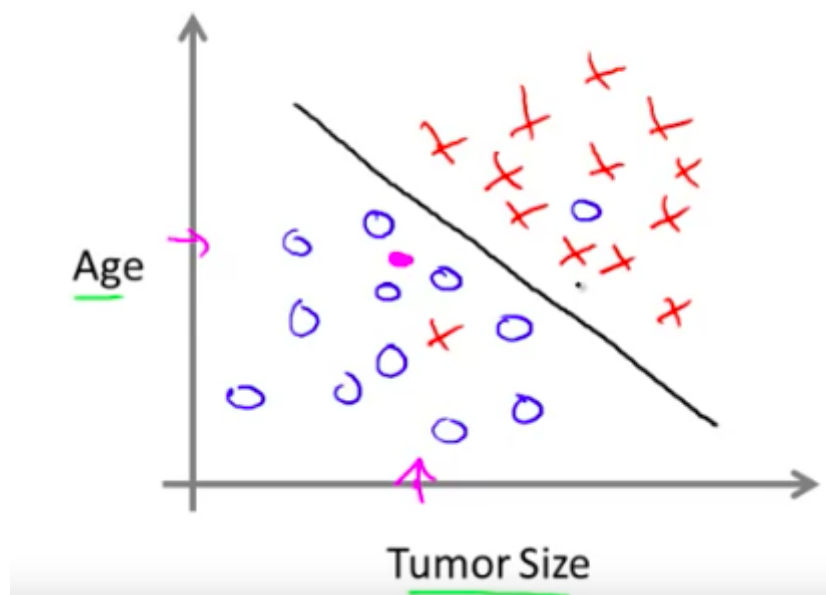
In a classification problem there is another way of plotting the data (circle = benign, cross = malignant)

Breast cancer (malignant, benign)



In this example, tumour size is the only attribute to predict malignancy or benignity of the mass.

However, in other classification problems we might have multiple attributes/features available (e.g. tumour size + patient age):



(circle = benign, cross = malignant)

Given such a dataset, what a learning algorithm might do is fitting a straight line to the data, to try to separate out the malignant tumours from the benign ones.

In other ML problems we have more often more than 2 features. One of the most interesting learning algorithms we will explore in this course is a learning algorithm that can deal with an **infinite number of features**, the **Support Vector Machine** algorithm (it uses a neat mathematical trick to allow a computer to deal with an infinite number of features)

***Question: You're running a company, and you want to develop learning algorithms to address each of two problems.

- Problem 1: You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.
 - Problem 2: You'd like software to examine individual customer accounts, and for each account decide if it has been hacked/compromised.
- Should you treat these as classification or as regression problems?***

Answer: Problem 1 -> regression, Problem 2 -> classification.

You're running a company, and you want to develop learning algorithms to address each of two problems.

1000's
↗ Problem 1: You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.

↗ Problem 2: You'd like software to examine individual customer accounts, and for each account decide if it has been hacked/compromised.

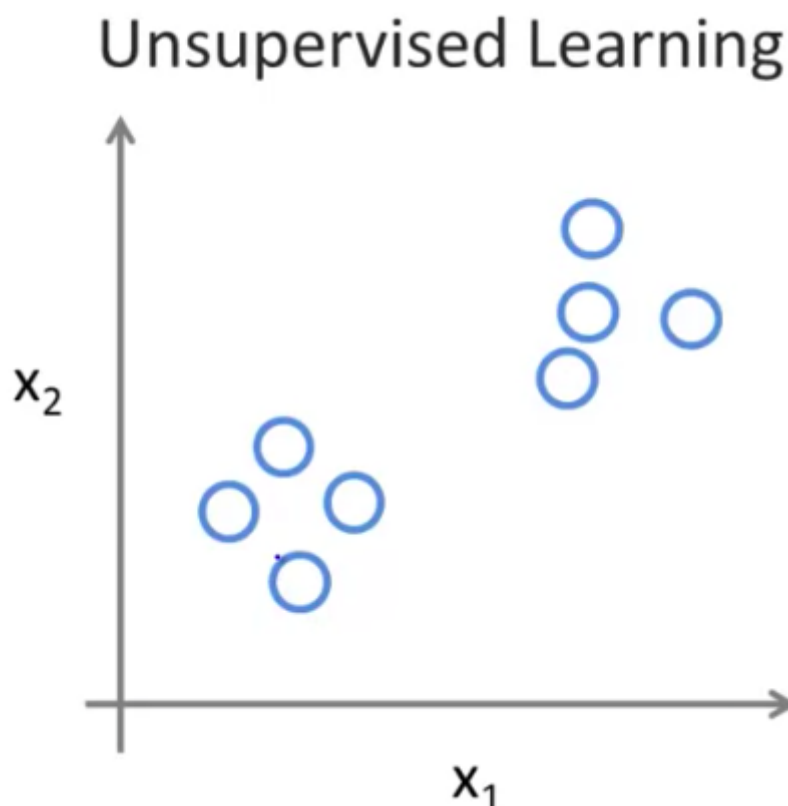
0 - not hacked
1 - hacked

Should you treat these as classification or as regression problems?

- ☐ Treat both as classification problems.
- ☐ Treat problem 1 as a classification problem, problem 2 as a regression problem.
- ↗ ☐ Treat problem 1 as a regression problem, problem 2 as a classification problem.
- ☐ Treat both as regression problems.

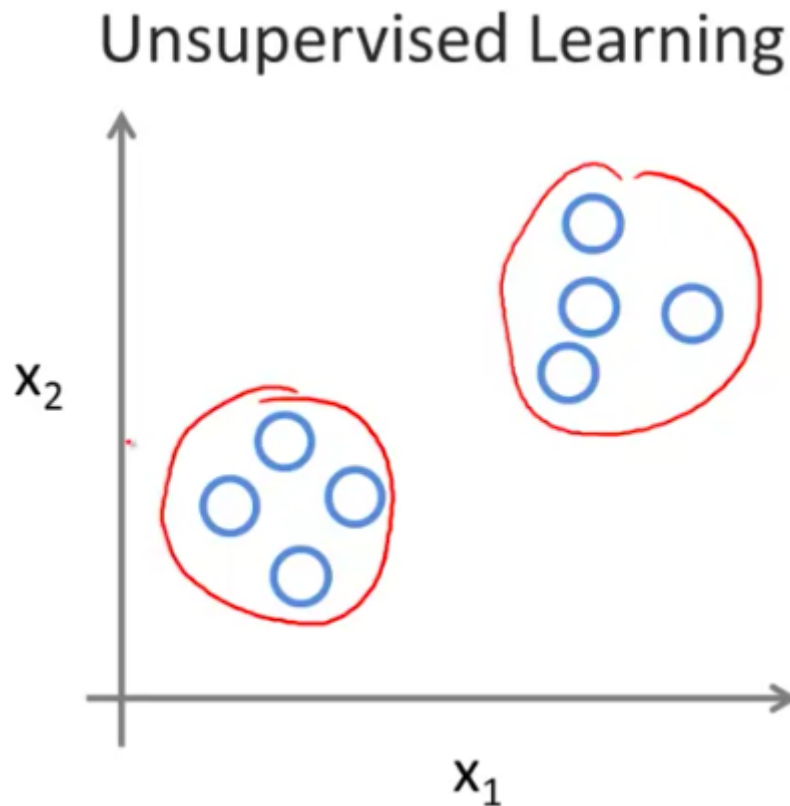
Unsupervised Learning

Unsupervised learning allows us to approach problems with **little or no idea what our results should look like**.



The data does not have any label attached. We can derive **structure** from data where we don't necessarily know the effect of the variables.

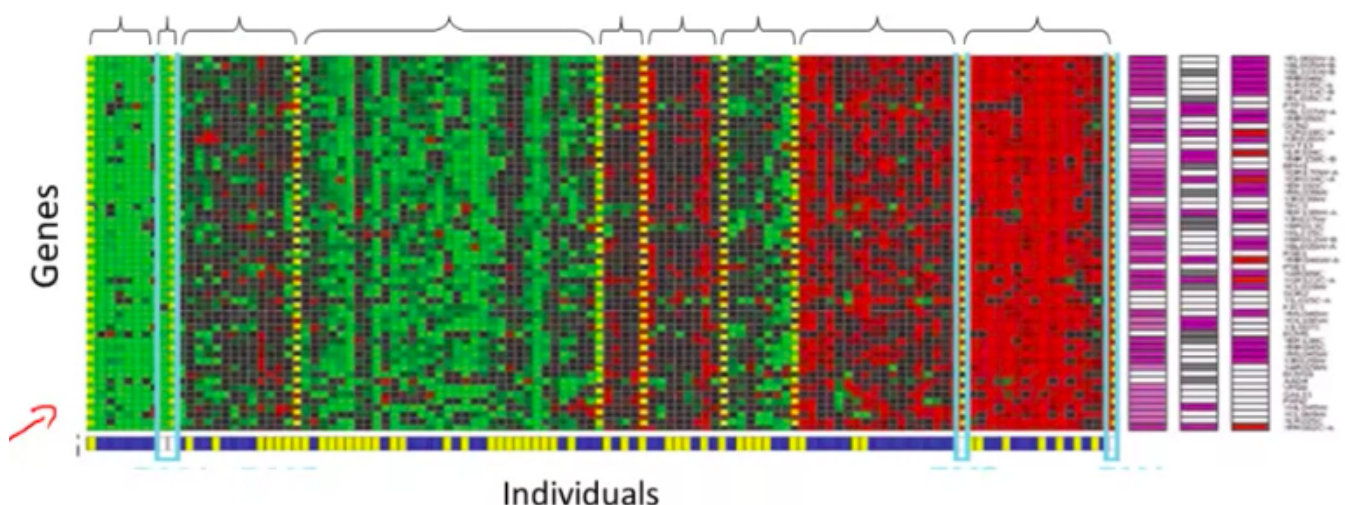
We can derive this structure by **clustering the data** based on relationships among the variables in the data. For example, an unsupervised ML algorithm might decide that the data lives in 2 different clusters:



With unsupervised learning there is no feedback based on the prediction results.

Examples:

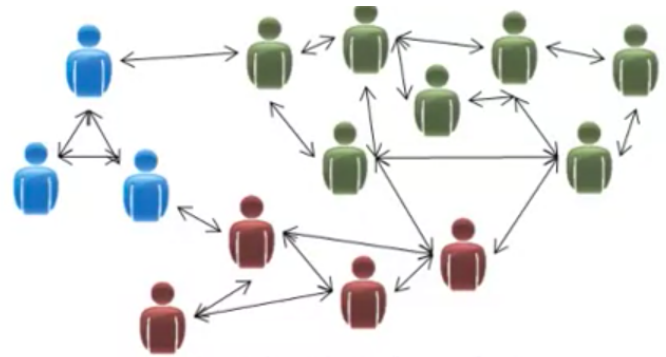
- **Clustering:** Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables, such as lifespan, location, roles, and so on. In the image below, people are grouped on the basis of patterns of expression of a list of genes:



Other examples for which non-supervised ML (clustering algorithm) is used:



Organize computing clusters



Social network analysis



Market segmentation



Astronomical data analysis

- **Non-clustering:** The "Cocktail Party Algorithm", allows you to find structure in a chaotic environment. (i.e. identifying individual voices and music from a mesh of sounds at a cocktail party). The algorithm consists in just one line of code:

```
[W,s,v] = svd(( repmat(sum(x.*x,1),size(x,1),1). *x) * x');
```

SVD function = Singular Value Decomposition (it is a linear algebra routine). The code is implemented in Octave language.

Question: *Of the following examples, which would you address using an unsupervised learning algorithm?*

- Given a set of news articles found on the web, group them into sets of articles about the same stories.
- Given a database of customer data, automatically discover market segments and group customers into different market segments.