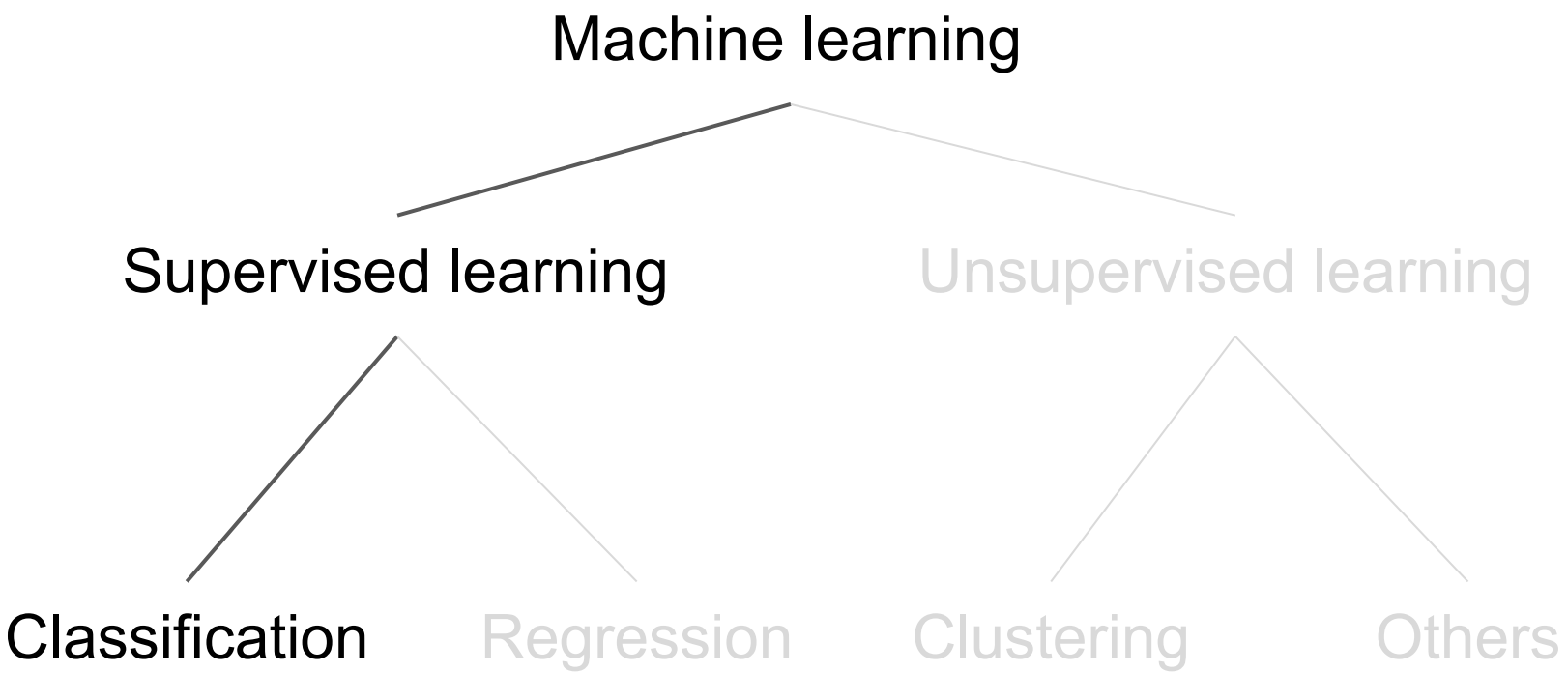
Introduction to Machine Learning
(Part 2)

Speaker: Johann Chu

CloudMile

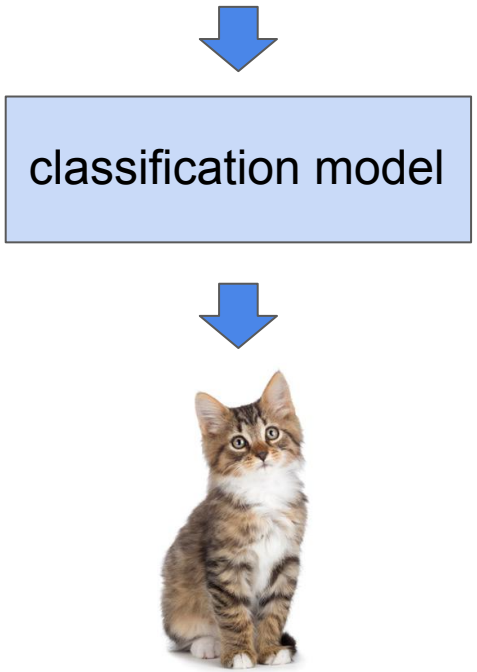
What we have learned



What classification can do

- A classification model enables us to classify data into different discrete categories, such as the animal clinic example.

Distance from eyes to nose	Age	Hoofed	Four-legged
2.2	12	False	True

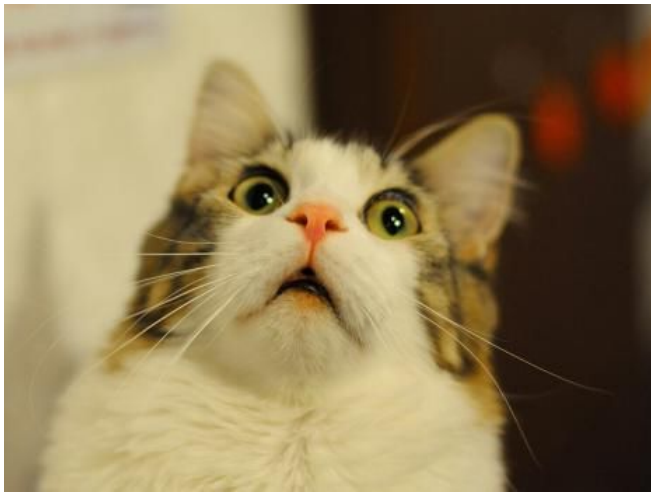


What if we want to predict something else?

Have owner	Age	Spayed / neutered
False	12	False

some magic model

Life expectancy: 13



Predicting a value instead of a category

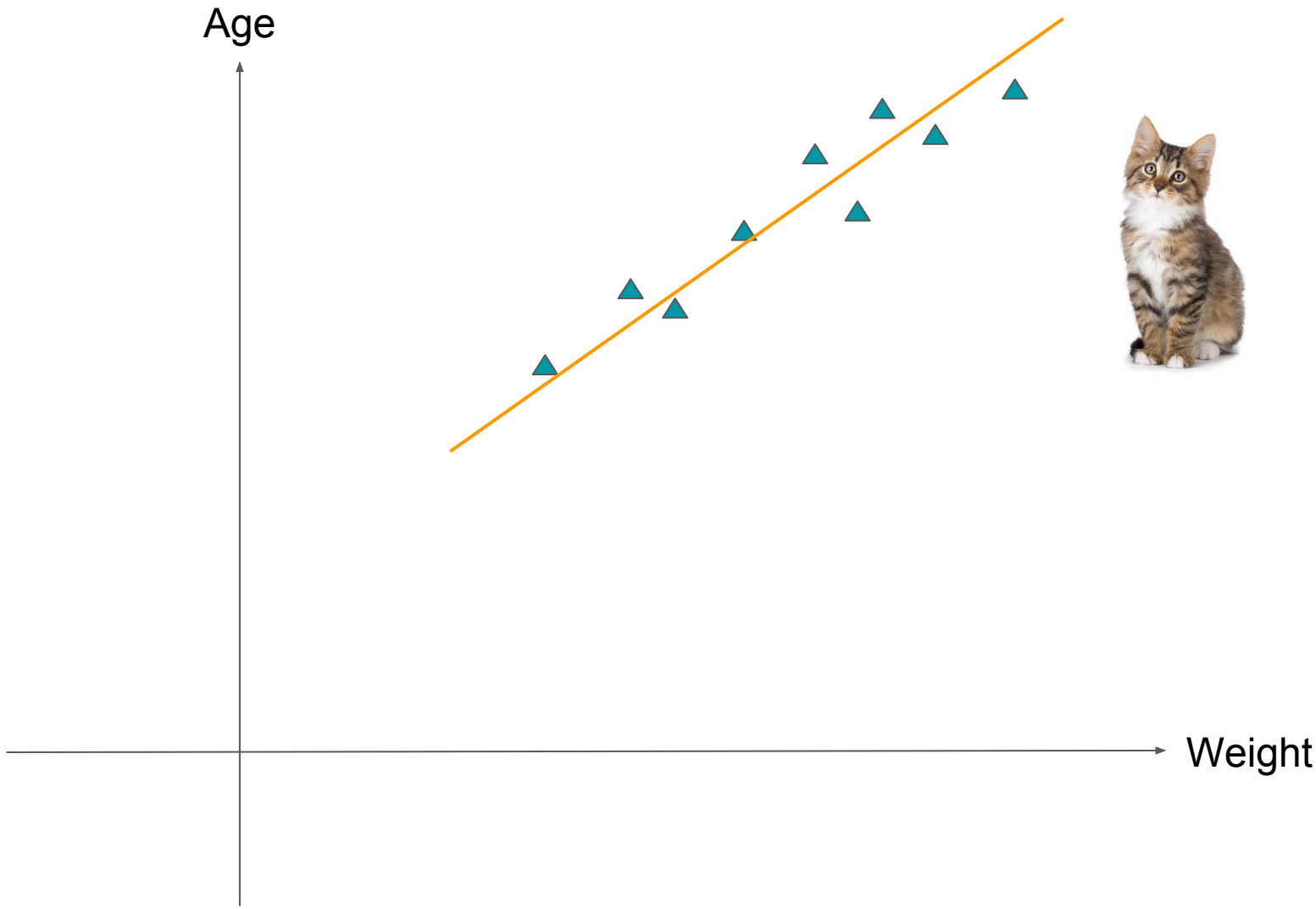
- In real world, we are very familiar with the results of value prediction, such as weather report, house pricing, your insurance quote, etc

How we can (try to) predict value

- When observing real world data, we (optimistically) assume that there should be some kind of "reasons" behind why some phenomenon happens.

- If we can find a way to figure out the reasons, we might be able to quantify the importance of each factor and predict the phenomenon.

- These "reasons" are mathematically modeled with attributes and functions.


Age
Weight

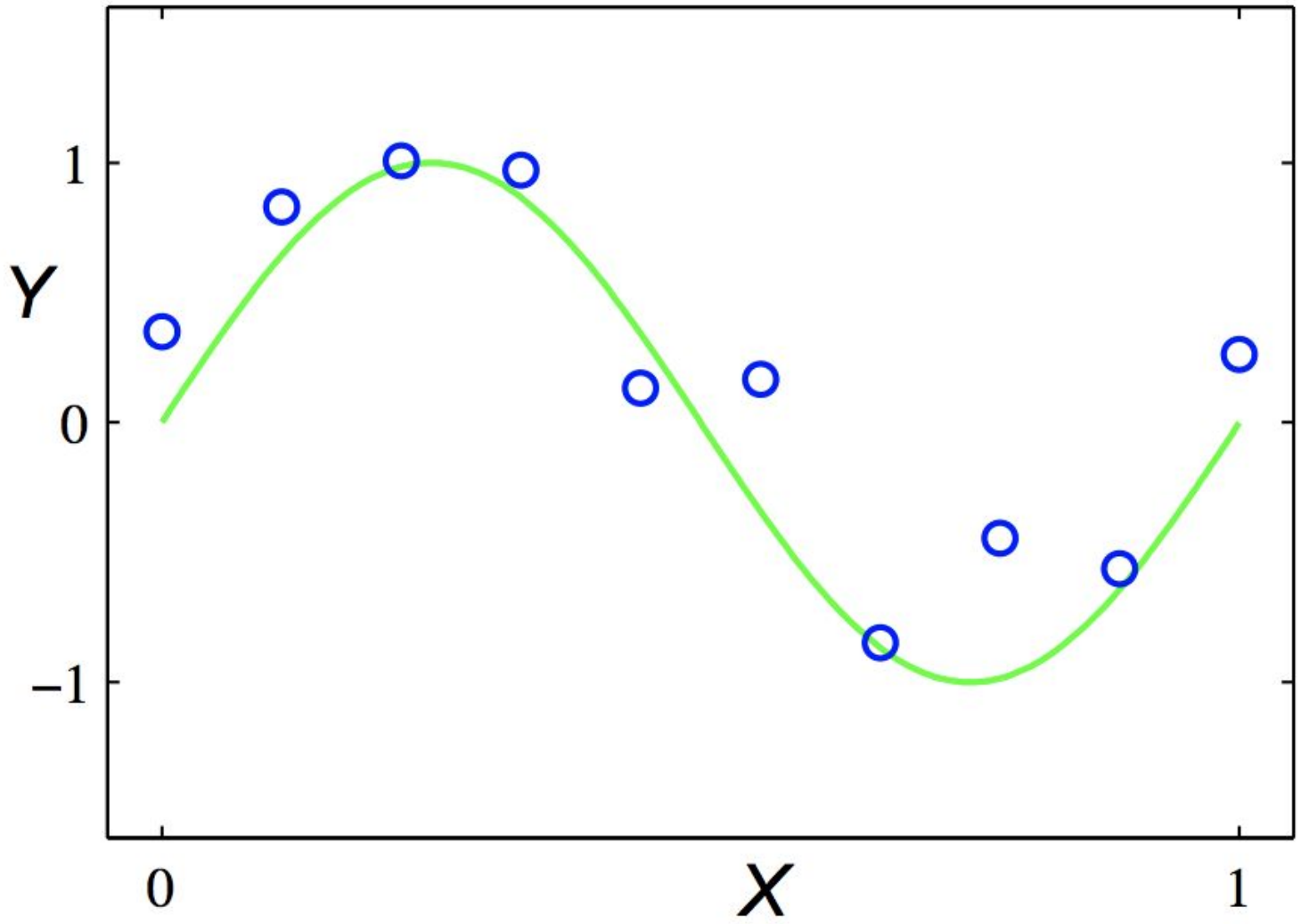
Another type of supervised learning: Regression

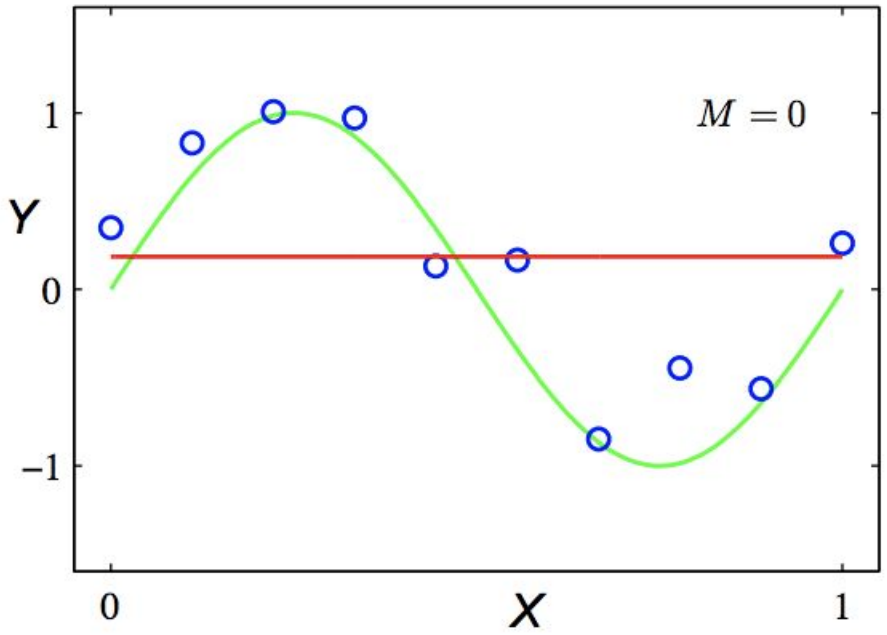
- What you see on previous slide is regression. In statistical modeling, regression is a process of finding the relationships among variables.

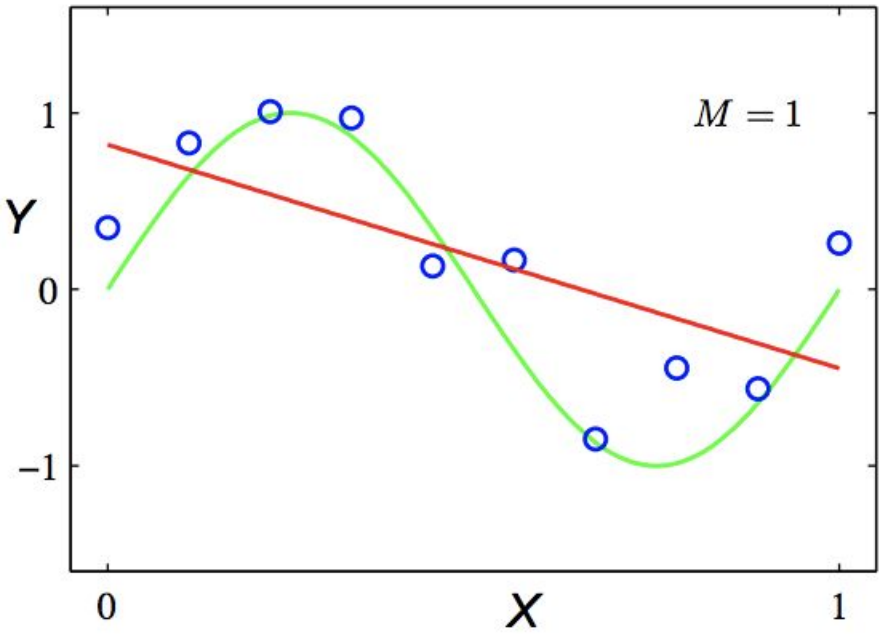
- To be more specific, we have demonstrated linear regression on last page when we assume that a *linear* relationship exists between the weight of a cat and its age.

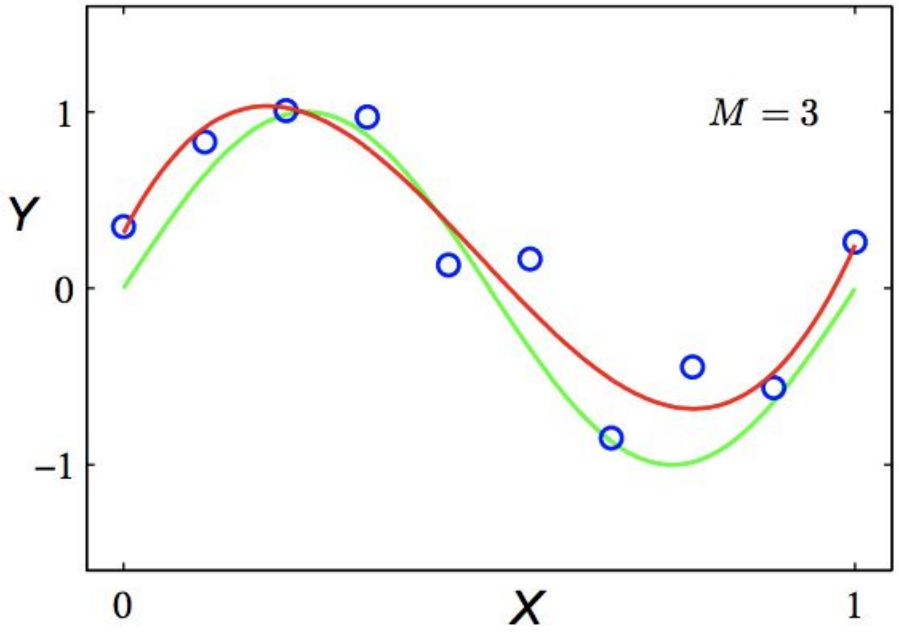
- In real world, though, the relationship is almost always more complex ;)

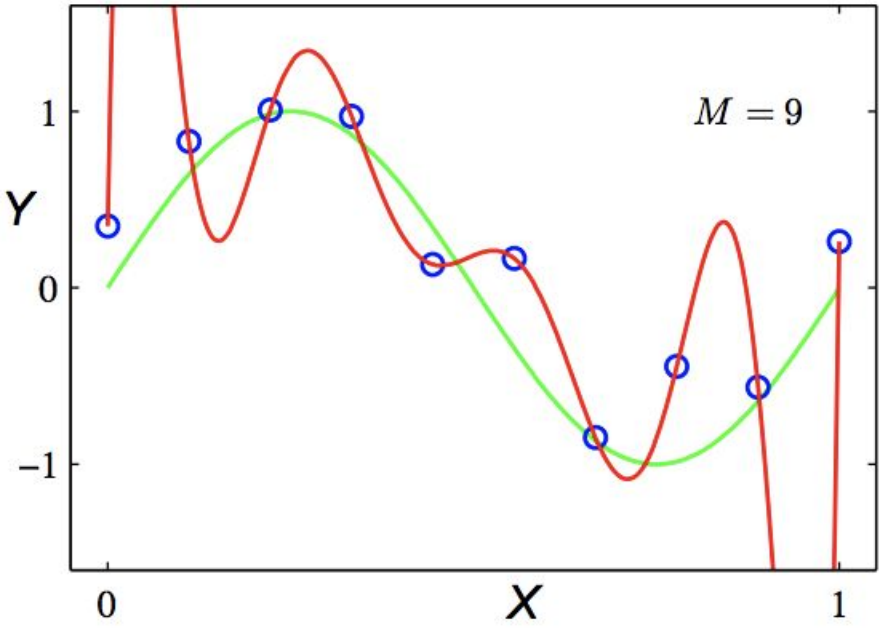
Dataset: 10 (X,Y) points generated from a sin function, with noise




M = 0
1
0
−1
Y
0
X
1

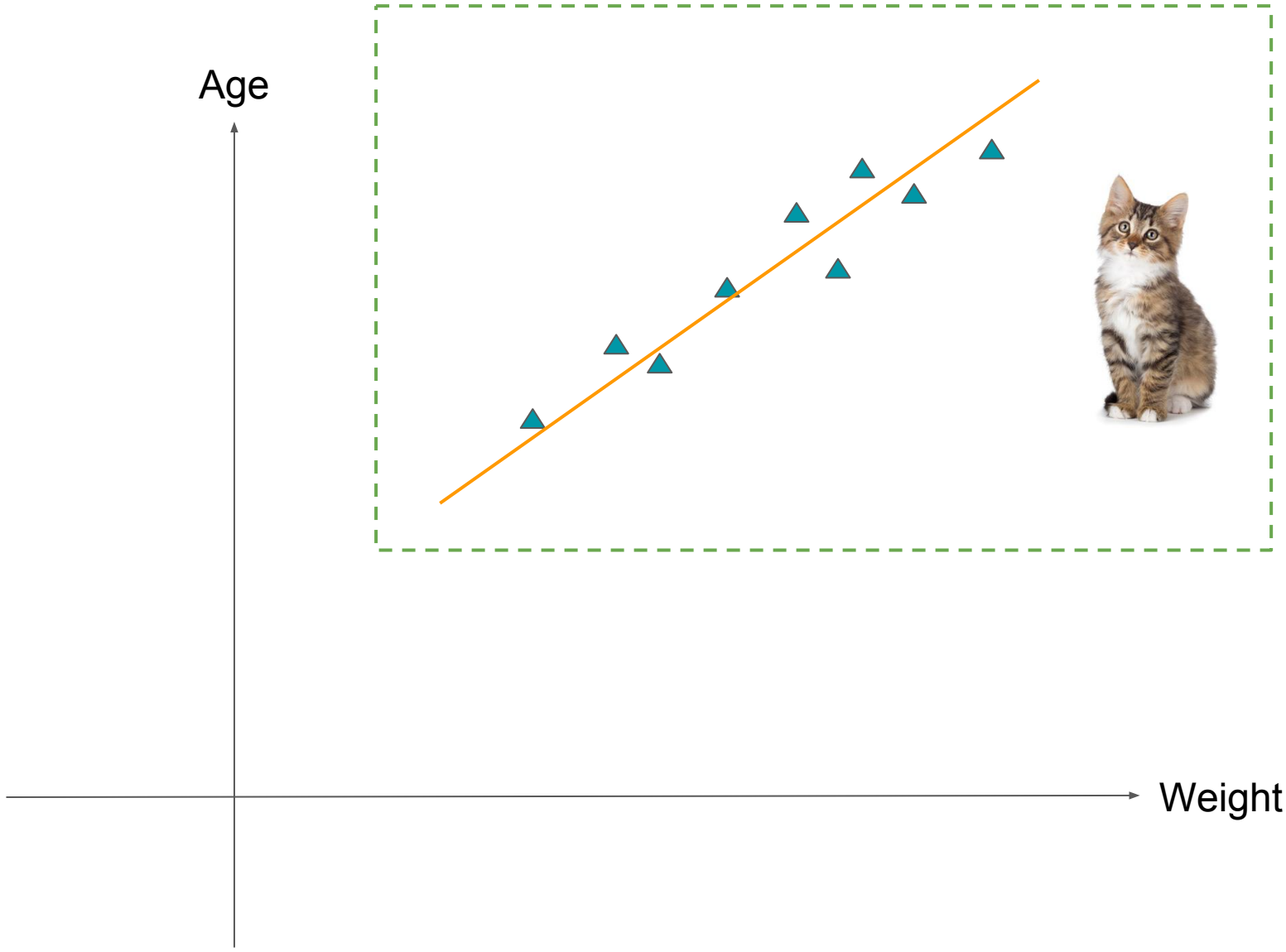

M = 1
1
0
−1
Y
0
X
1


M = 3
1
0
−1
Y
0
X
1


M = 9
1
0
−1
Y
0
X
1

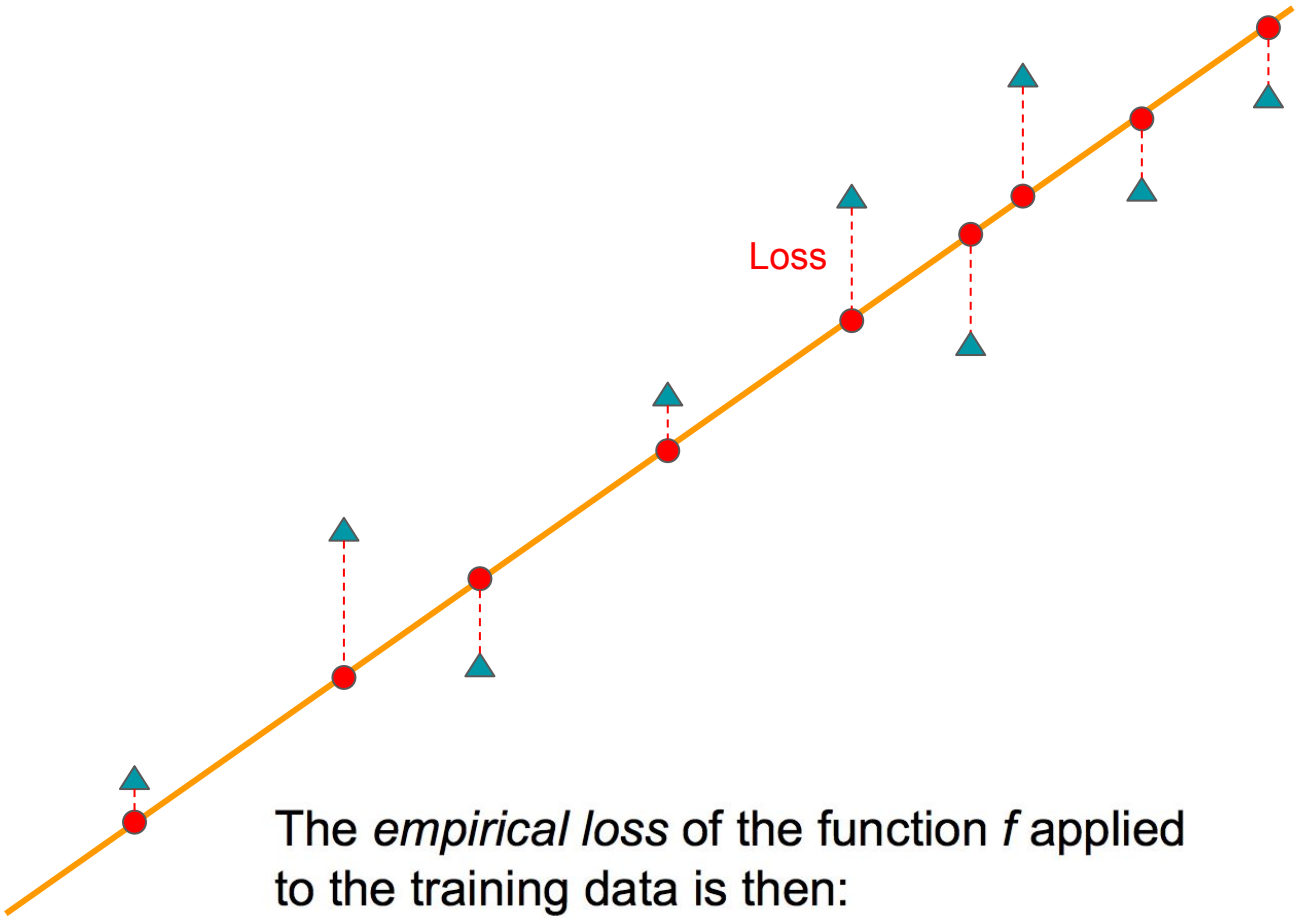
How to measure the error

- To measure how good (or bad) a function fits the data, we calculate the difference between the prediction (generated by the function) and the real data point. This difference is called "loss".


Age
Weight

For regression, a common choice is squared loss:

$$L(y_i, f(x_i)) = (y_i - f(x_i))^2$$



The *empirical loss* of the function *f* applied to the training data is then:

$$\frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i)) = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2$$

(Also called Mean Squared Error)

How to reduce the error (to find the best fit)?

- Mathematically, the best regression line is the one that minimizes the loss. In statistics, we employ the least square methodology by finding the derivatives with respect to each independent variable.

Assuming $y=\beta_0+\beta_1 x$

Find β_1

Find β_0

$$\frac{\partial \text{Loss}(\hat{\beta}_0,\hat{\beta}_1)}{\partial \beta_0}=\frac{\partial \sum_{i=1}^{n}(y_i-\hat{\beta}_0-\hat{\beta}_1 x_i)^2}{\partial \beta_0}=0$$

$$\Rightarrow -2\sum_{i=1}^{n}(y_i-\hat{\beta}_0-\hat{\beta}_1 x_i)=0$$

$$\Rightarrow \hat{\beta}_0=\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{\beta}_1 x_i\right)$$

$$\Rightarrow \hat{\beta}_0=\bar{y}-\hat{\beta}_1\bar{x}$$

$$\frac{\partial \text{Loss}(\hat{\beta}_0,\hat{\beta}_1)}{\partial \beta_1}=\frac{\partial \sum_{i=1}^{n}(y_i-\hat{\beta}_0-\hat{\beta}_1 x_i)^2}{\partial \beta_1}=0$$

$$\Rightarrow -2\sum_{i=1}^{n}(y_i-\hat{\beta}_0-\hat{\beta}_1 x_i)x_i=0$$

$$\Rightarrow \sum_{i=1}^{n}(y_i-\hat{\beta}_0-\hat{\beta}_1 x_i)x_i=0$$

$$\Rightarrow \sum_{i=1}^{n}y_i x_i-\sum_{i=1}^{n}\hat{\beta}_1 x_i^2-\sum_{i=1}^{n}\hat{\beta}_0 x_i=0$$

$$\Rightarrow \sum_{i=1}^{n}y_i x_i-\sum_{i=1}^{n}\left(\bar{y}-\hat{\beta}_1\bar{x}\right)x_i-\hat{\beta}_1\sum_{i=1}^{n}x_i^2=0$$

$$\Rightarrow \sum_{i=1}^{n}y_i x_i-\sum_{i=1}^{n}\bar{y}x_i+\sum_{i=1}^{n}\hat{\beta}_1\bar{x}x_i-\hat{\beta}_1\sum_{i=1}^{n}x_i^2=0$$

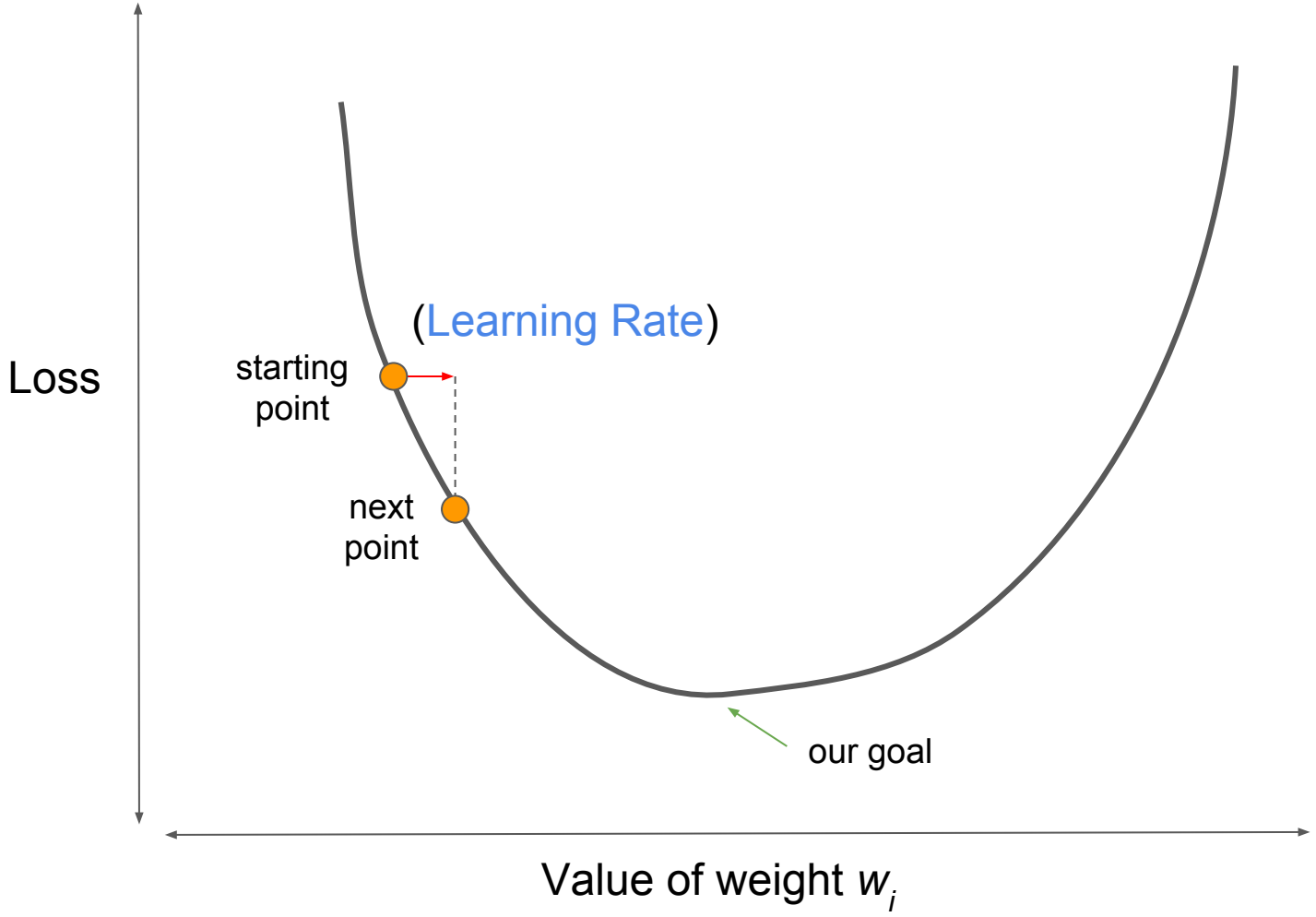
$$\Rightarrow \hat{\beta}_1\sum_{i=1}^{n}(x_i-\bar{x})x_i=\sum_{i=1}^{n}(y_i-\bar{y})x_i$$

$$\Rightarrow \hat{\beta}_1=\frac{\sum_{i=1}^{n}(y_i-\bar{y})(x_i-\bar{x})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

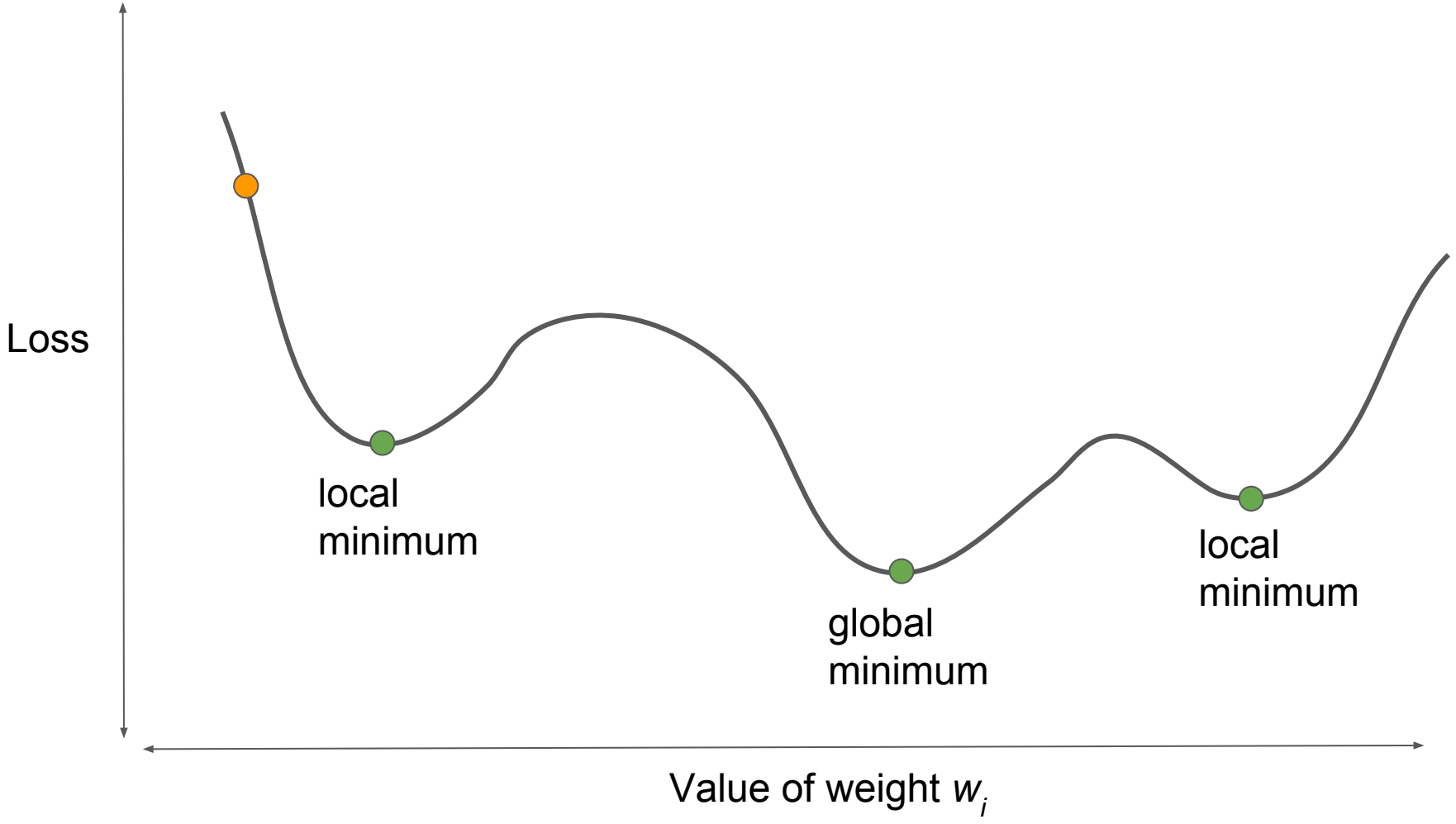
- However, in real-world cases, it is not always possible to find an analytical solution to the problem (either too complex or computationally expensive).

- Thus, instead of solving the problem directly, we can use certain approach to iteratively find the solution. One of such approach is gradient descent.

Gradient Descent



But life won't be always so kind...




早安～您好～
知足常樂
• Sometimes you need to be very lucky to get the optimal solution.
• It is *okay* to have sub-optimal solution.
• Be satisfied with what you have as long as you are improving.
認同請分享～

Don't forget about the overfitting problem

- To avoid overfitting multivariate function during regression (which could easily happen), we can do more than simply minimizing the loss. For example, we can try penalizing the complex models; such principle is called regularization.

- Two common ways to think of model complexity:
  - Model complexity as a function of the weights of all features in the model.
  - Model complexity as a function of the total number of features with nonzero weights.

This is called L_2 regularization!

L_2 Regularization

- The L_2 regularization defines the regularization term as the sum of the squares of all the feature weights:

$$L_2 \text{ regularization term} = ||\boldsymbol{w}||_2^2 = w_1^2 + w_2^2 + \ldots + w_n^2$$

For example, a linear model with the following weights:

$$\{w_1 = 0.2, w_2 = 0.5, w_3 = 5, w_4 = 1, w_5 = 0.25, w_6 = 0.75\}$$

Has an L_2 regularization term of 26.915:

$$w_1^2 + w_2^2 + \boldsymbol{w_3^2} + w_4^2 + w_5^2 + w_6^2$$

$$= 0.2^2 + 0.5^2 + \boldsymbol{5^2} + 1^2 + 0.25^2 + 0.75^2$$

$$= 0.04 + 0.25 + \boldsymbol{25} + 1 + 0.0625 + 0.5625$$

$$= 26.915$$

Kind of regression, kind of not: Logistic Regression

- When talked about regression, we would often encounter a technique called logistic regression. Although named regression, it is actually a (linear) *classification* scheme.

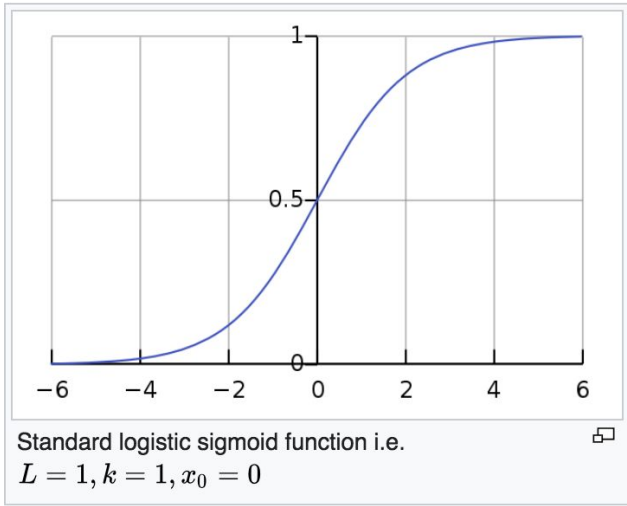
- Logistic regression is helpful when we want to estimate the probability of data regarding some binary dependent variable. (E.g. win/lost, pass/fail, etc)

- But before we talk about logistic regression, we should start by talking about logistic function, which is what the method is based on.

Logistic Function

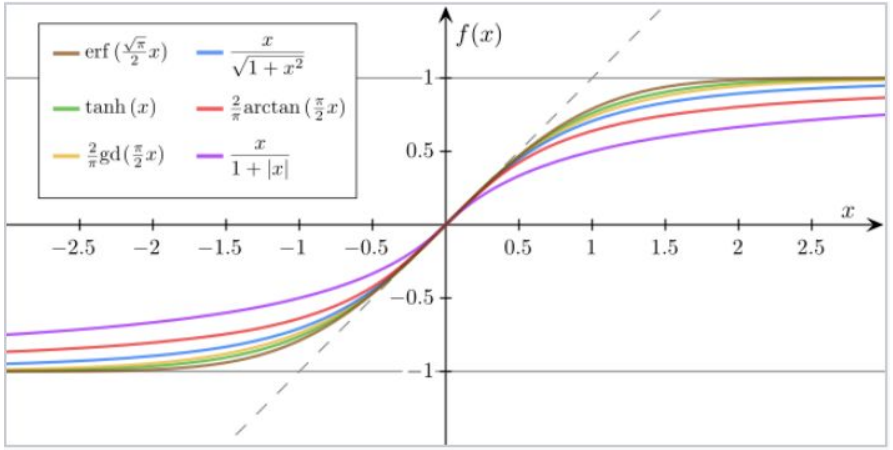
- A logistic function is a common "S" shape function with equation:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$



Standard logistic sigmoid function i.e.
$L = 1, k = 1, x_0 = 0$

- Logistic function is one type of sigmoid function. Other types include hyperbolic tangent, arctangent function, etc.



Wait, what does "logistic" mean?

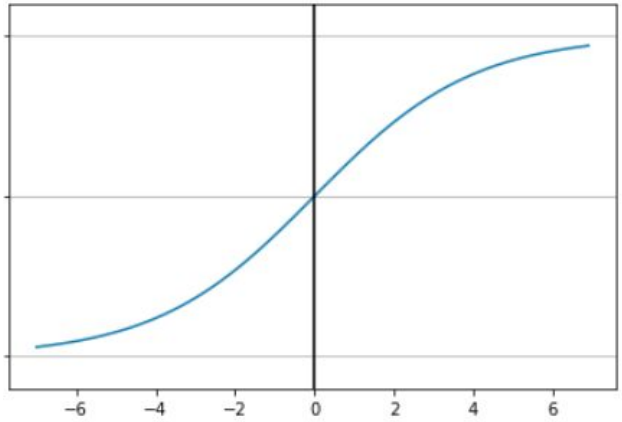
- The function was named in 1844 by Pierre François Verhulst. However, he never actually explained *why* it was named this way.

- People speculated the name might have a connection with the "logarithmic" basis of the function. Logarithm was coined by John Napier (1550-1617) from Greek logos (ratio, proportion, reckoning) and arithmos (number). Logistic comes from the Greek logistikos (computational). In the 1700's, logarithmic and logistic were synonymous.

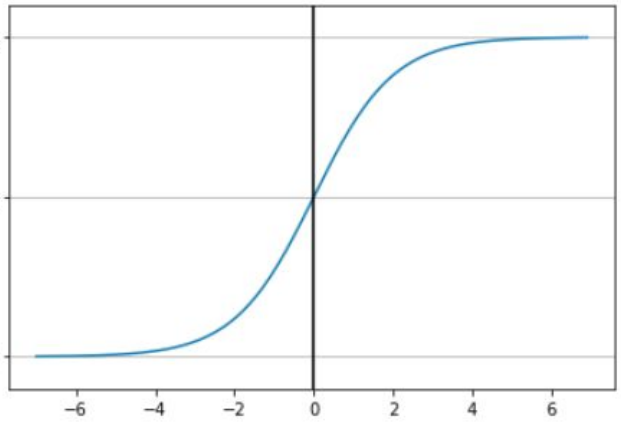
Some explanation on the numbers

- L decides the height (max value).
- k decides the steepness.
- x_0 decides the x-value of midpoint

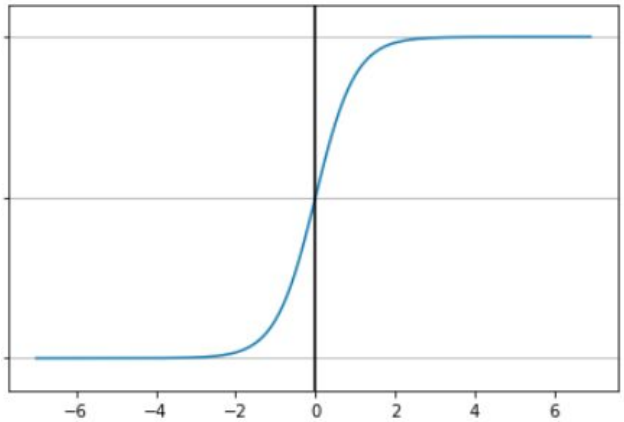
$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$



k = 0.5



k = 1



k = 2

So how does it work?

We want a linear classifier in the form: $z = b + w_1x_1 + w_2x_2 + \ldots w_Nx_N$

Ideally, the classifier should produce positive value if the data point belongs to situation 1, negative if 0

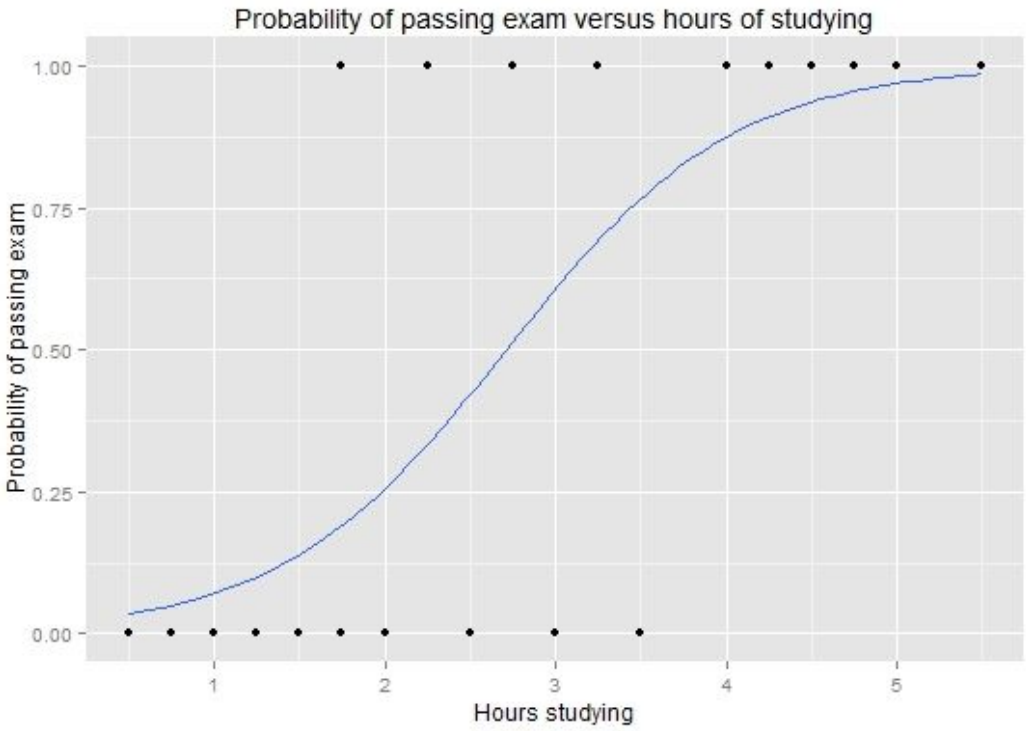
We also hope that the likelihood of a data point being correctly classified can be as high as possible (of course...)

As a result, we want a function that can map the output *z* to probability that not only gives you a higher output with a higher *z*, but also allows us to use maximum likelihood estimation to find the right coefficients

That is why we use logistic function!

Example: Study hours / Probability of passing exam

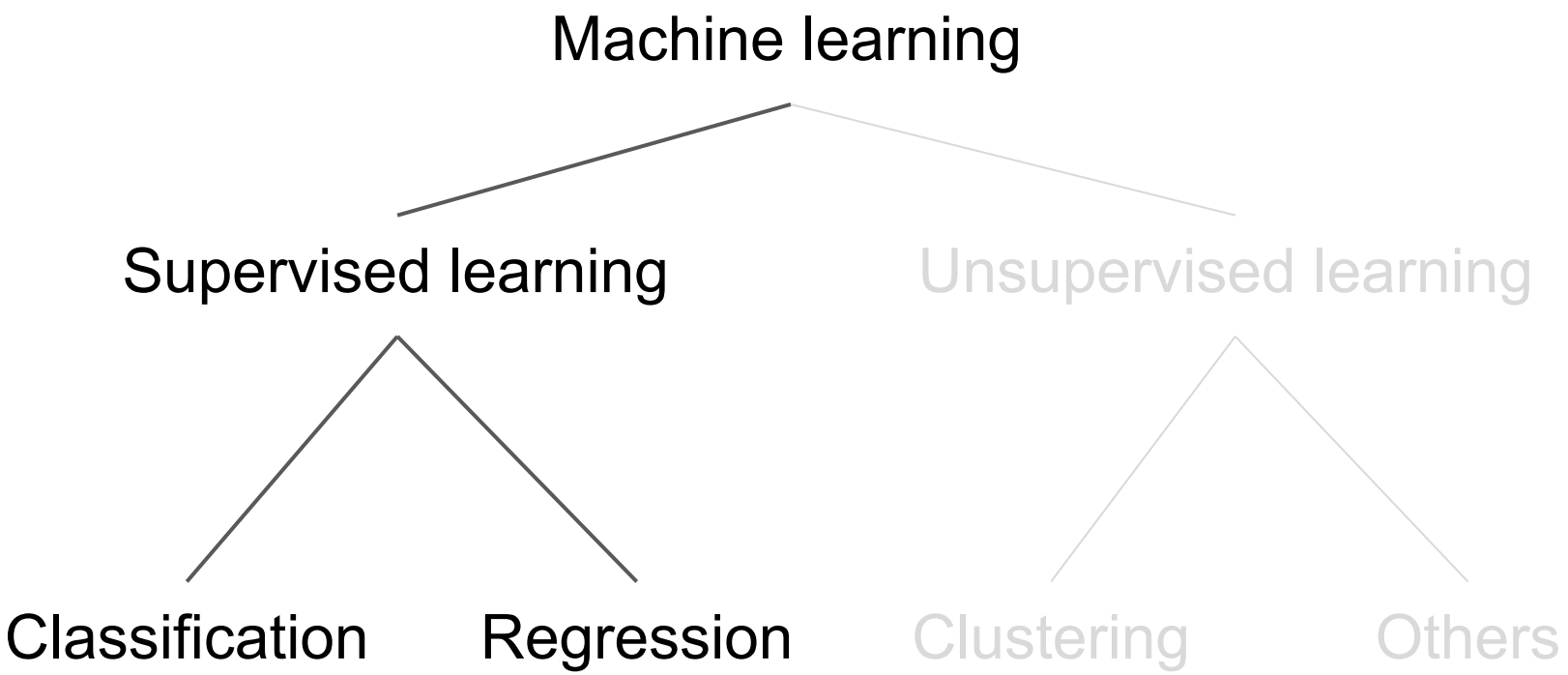
Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1



"Enough!!!"

--- Loki, *The Avengers*.

What we have learned



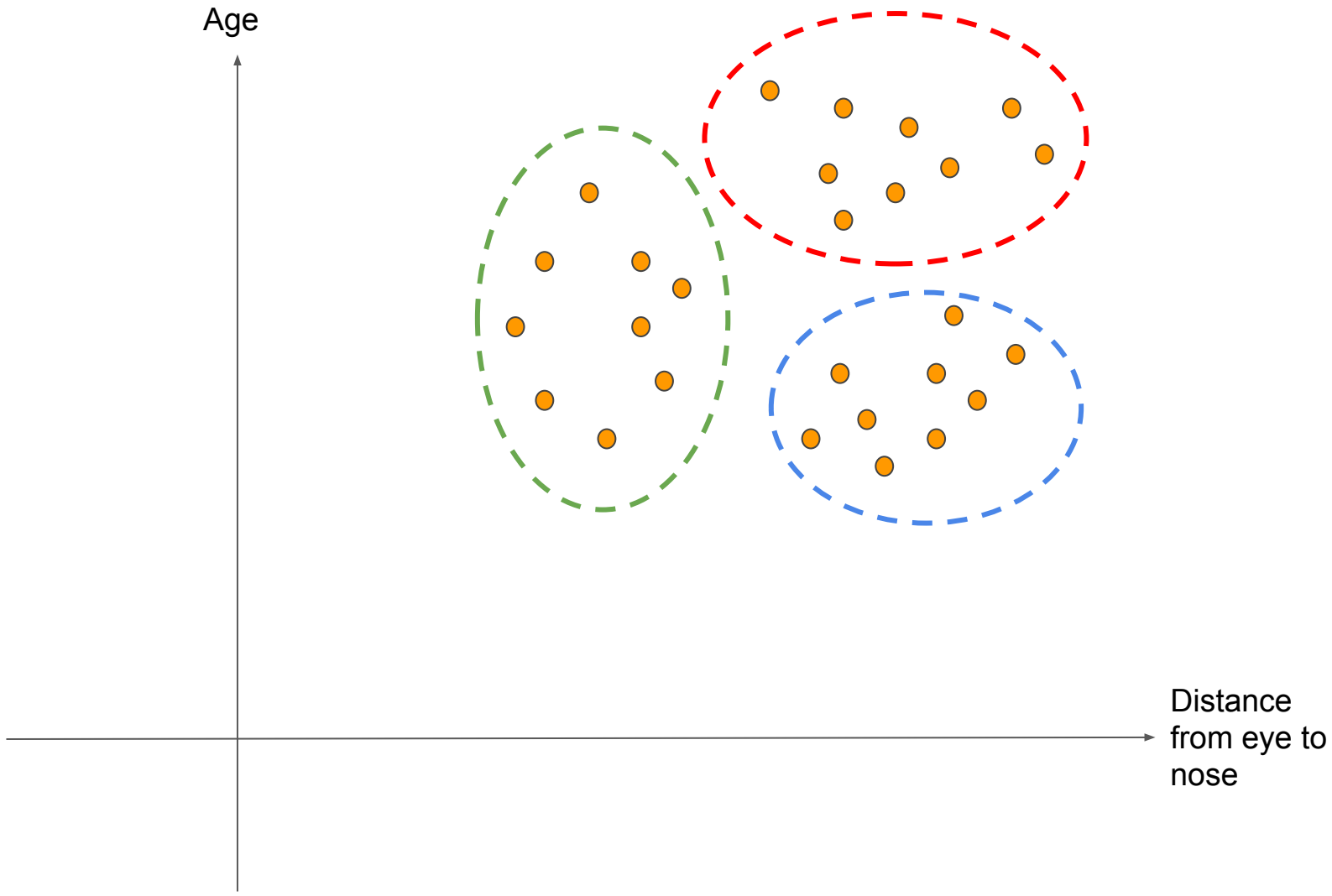
When all hope (data) seems to fade

- We have been talking about cases where our data has some kind of labels or values. What if the data we are dealing with has no answer?

- This type of task is called unsupervised learning; that is, the data would *not* have a ground truth to serve as the supervision to the training process.

- The most common method of unsupervised learning is clustering, in which we attempt to aggregate data into different groups.

Clustering



The most popular clustering method: K-means

- K-means clustering aims to partition data points into *k* clusters in which each point belongs to the cluster with the nearest mean.

K-means algorithm:

Random select *k* points as the seeds to represent *k* groups

assign each point to the nearest seed (group)

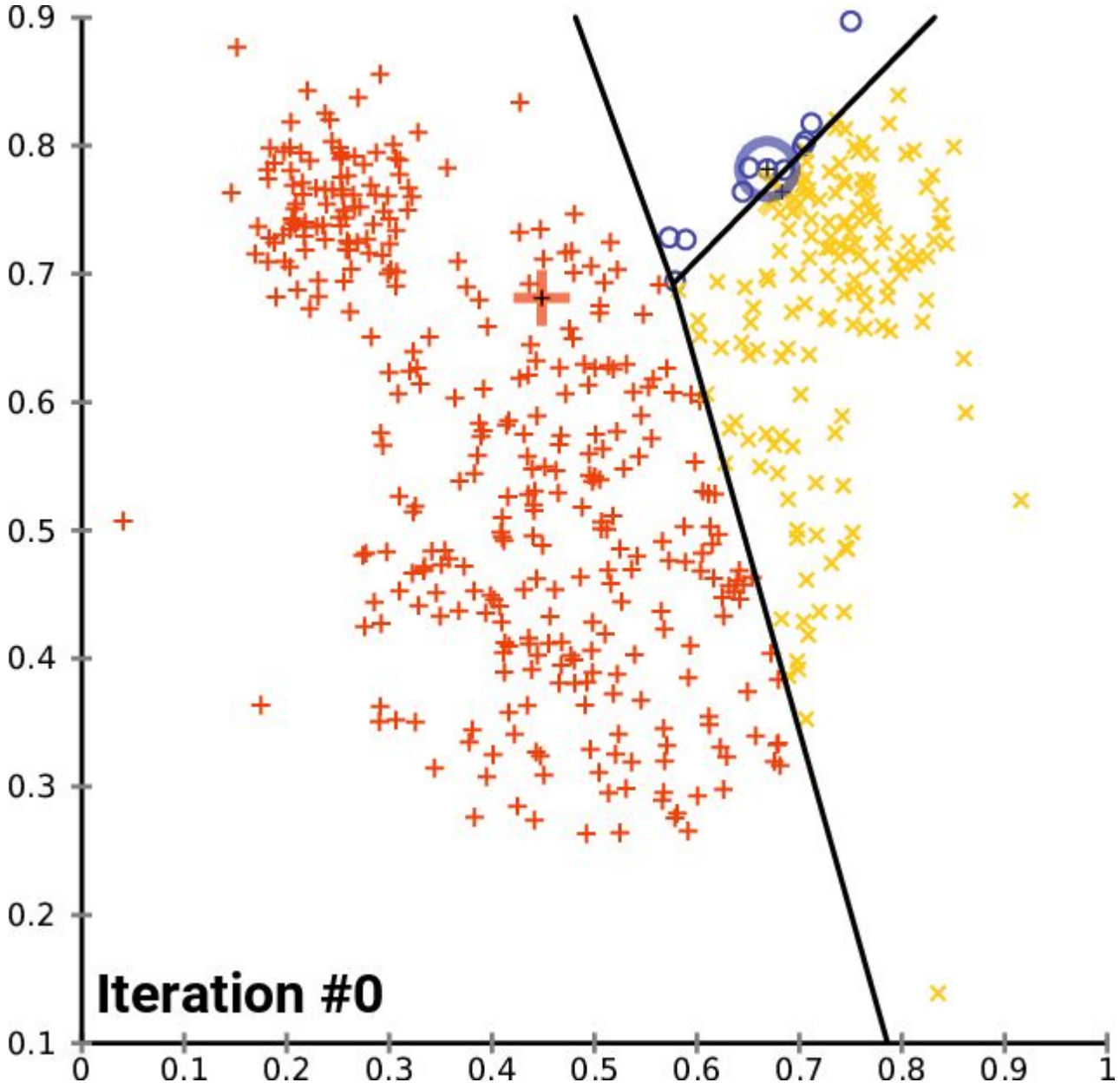
calculate new centroid for each group

repeat:

assign each point to the nearest centroid

calculate new centroid for each group

until membership stop changing


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Iteration #0

Some other things we can do with the features

- For structured data, it is not unusual to have more than a dozen features. However, it will be very complicated if we want to analyze the interactions between the features or leverage all of them for tasks.

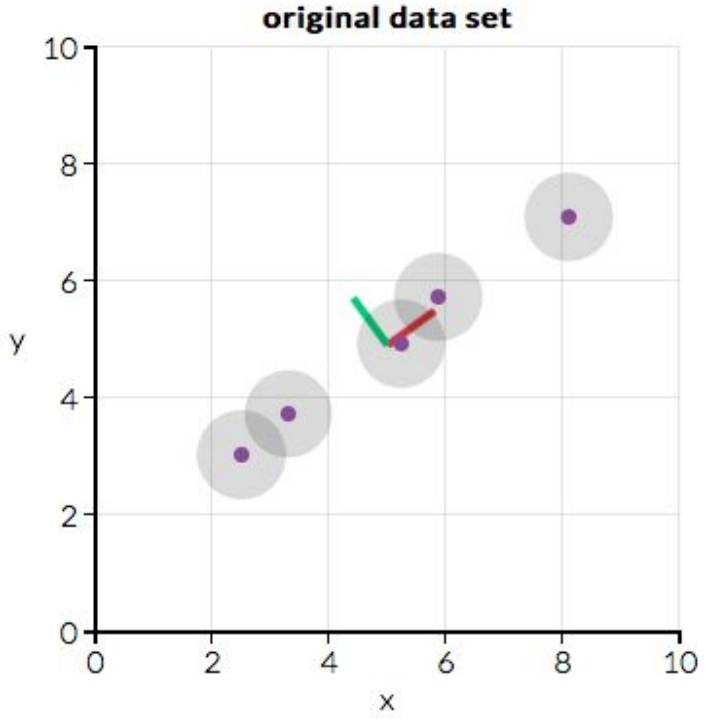
- *Feature extraction* refers to cherry-picking the essential features from the data; sometimes we also need to do dimensionality reduction to filter out the features that do not really play important roles.

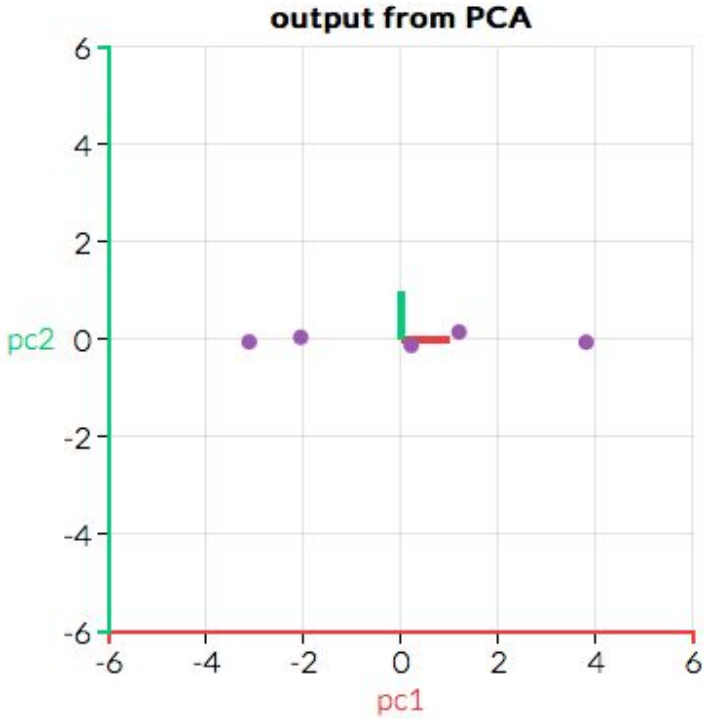
- Principal Component Analysis (PCA) can be used in such a manner.

Principal Component Analysis (PCA)

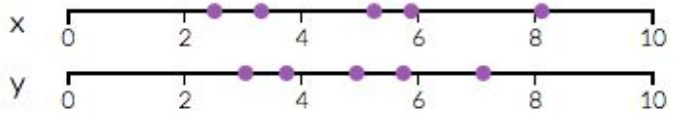
- PCA is used to decompose a multivariate dataset in a set of successive orthogonal components that explain a maximum amount of variance through the linear combination of attributes.

- To explain it in another way, PCA can be thought of as rotating the axis of the original dataset to point to the directions along which give us the most variation.

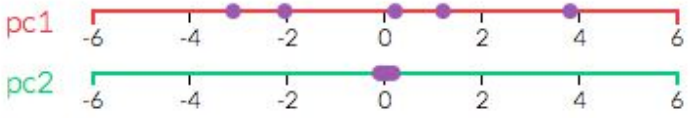




PCA is useful for eliminating dimensions. Below, we've plotted the data along a pair of lines: one composed of the x-values and another of the y-values.

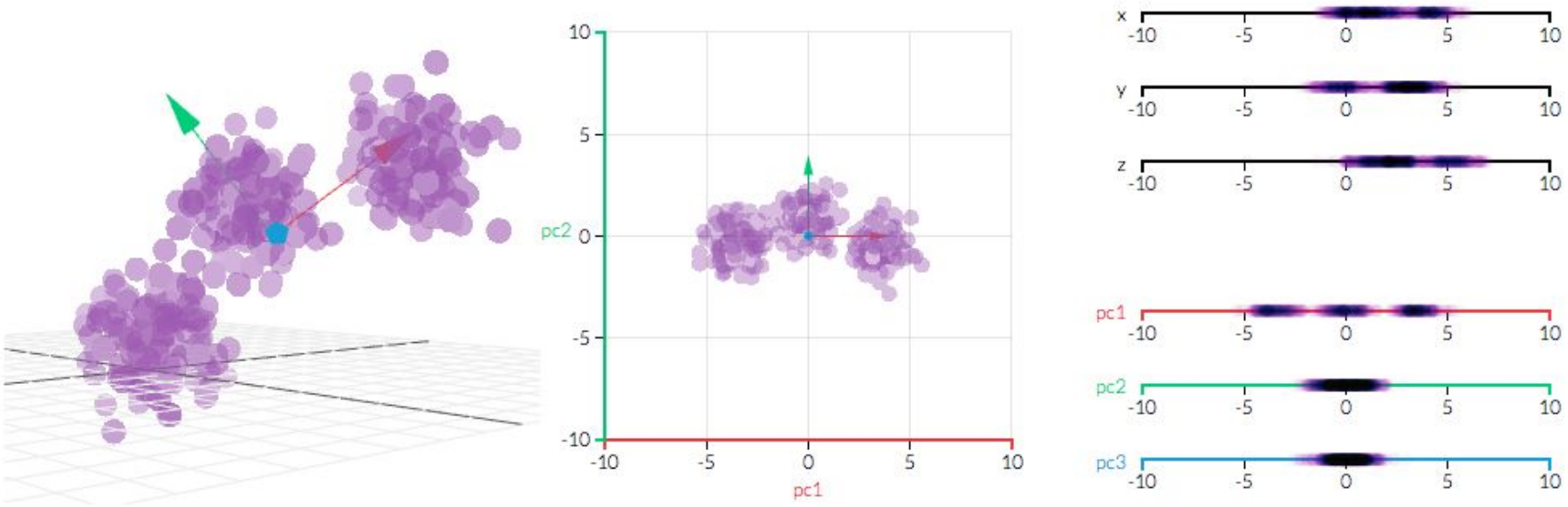


If we're going to only see the data along one dimension, though, it might be better to make that dimension the principal component with most variation. We don't lose much by dropping PC2 since it contributes the least to the variation in the data set.



3D example

With three dimensions, PCA is more useful, because it's hard to see through a cloud of data. In the example below, the original data are plotted in 3D, but you can project the data into 2D through a transformation no different than finding a camera angle: rotate the axes to find the best angle. To see the "official" PCA transformation, click the "Show PCA" button. The PCA transformation ensures that the horizontal axis PC1 has the most variation, the vertical axis PC2 the second-most, and a third axis PC3 the least. Obviously, PC3 is the one we drop.



How PCA works (mathematically)

- The actual process of running PCA can be listed as follows:
  - Calculate the covariance matrix X of data points.
  - Calculate eigen vectors and corresponding eigen values.
  - Sort the eigen vectors according to their eigen values in decreasing order.
  - Choose first *k* eigen vectors and that will be the new *k* dimensions.
  - Transform the original *n* dimensional data points into *k* dimensions.

Usage of PCA in other fields

- The concept of PCA can be applied in information retrieval.

- Assuming you have some documents, each containing a large number of vocabulary. We can apply PCA on the documents to find the most important dimensions (words) that can be used to describe these documents.

- We call this kind of method latent semantic analysis (LSA).

"That's all."

--- Miranda Priestly, *The Devil Wears Prada*.