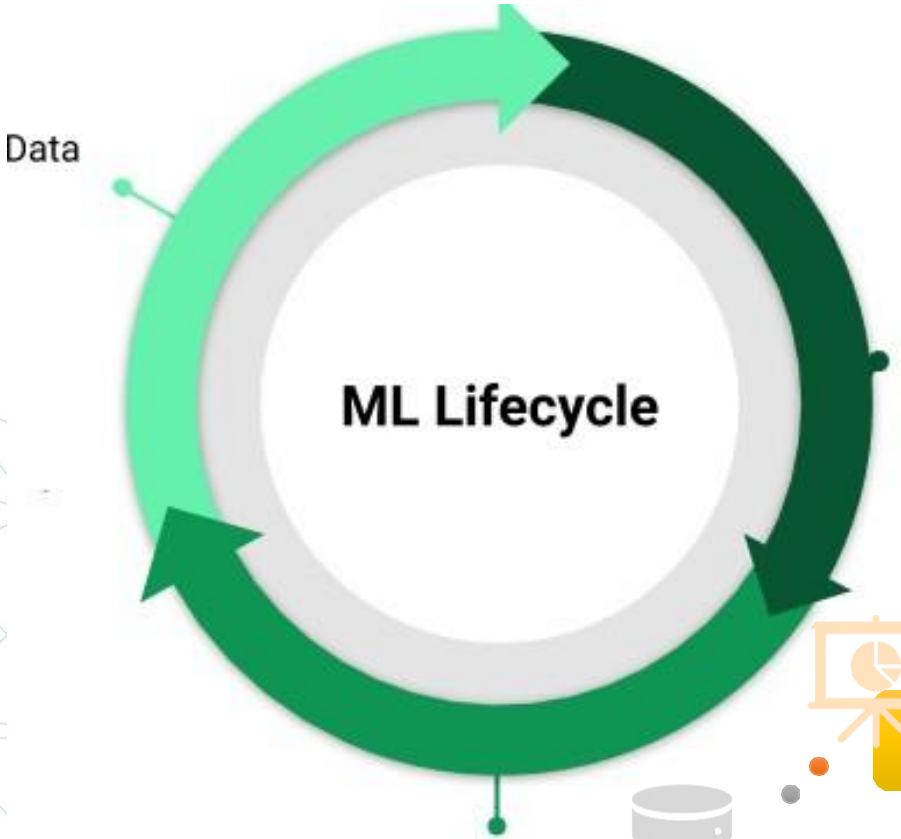
Machine Learning

DR. Eman Ruslan CLS



Evaluate

Train

Machine learning Model

Data Preparation

Data Exploration

Data Acquisition

Define Project Objectives

Supervised Learning

Supervised Learning

enjoy algorithms

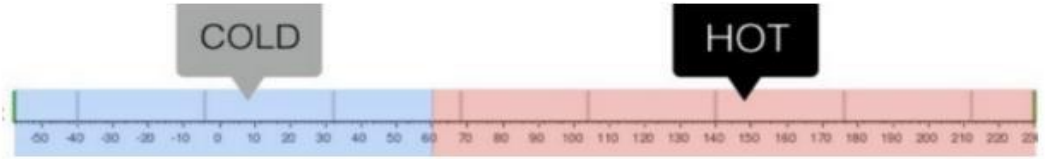
Regression

What will be the temperature tomorrow?

Fahrenheit

Classification

Will it be hot or cold tomorrow?



Fahrenheit

Model

A **Mathematical representation** of a **real world process** in the form of **input-output** relationship.

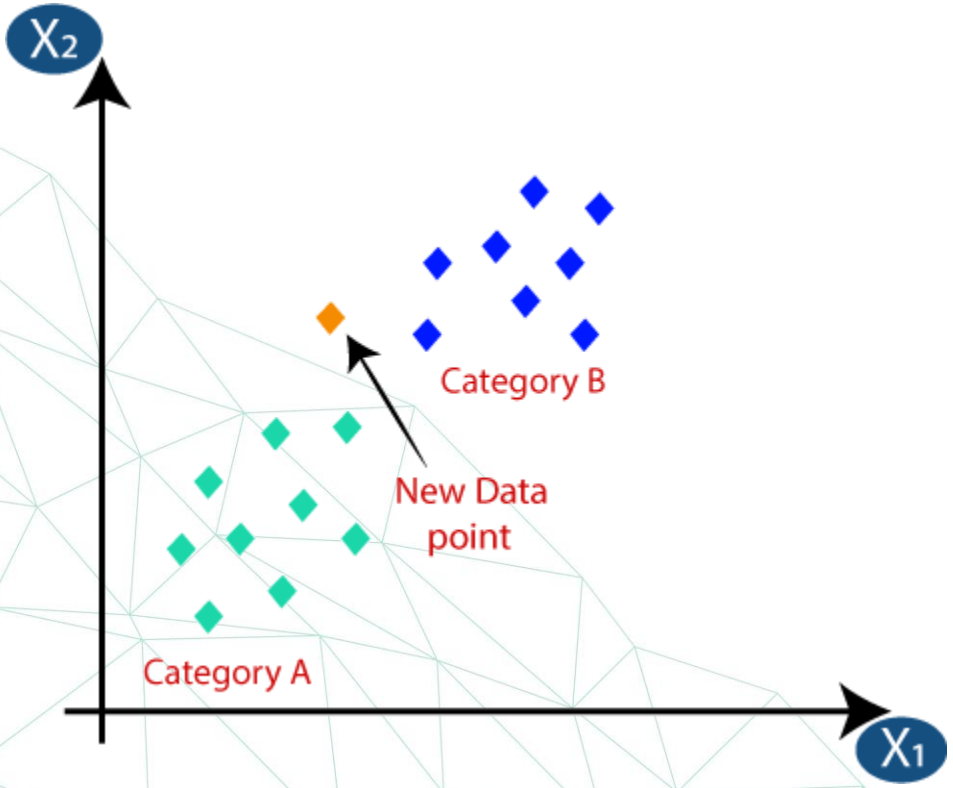
$$slices\ of\ pizza\ i'll\ eat = 2 * (hours\ since\ last\ meal) + 1$$

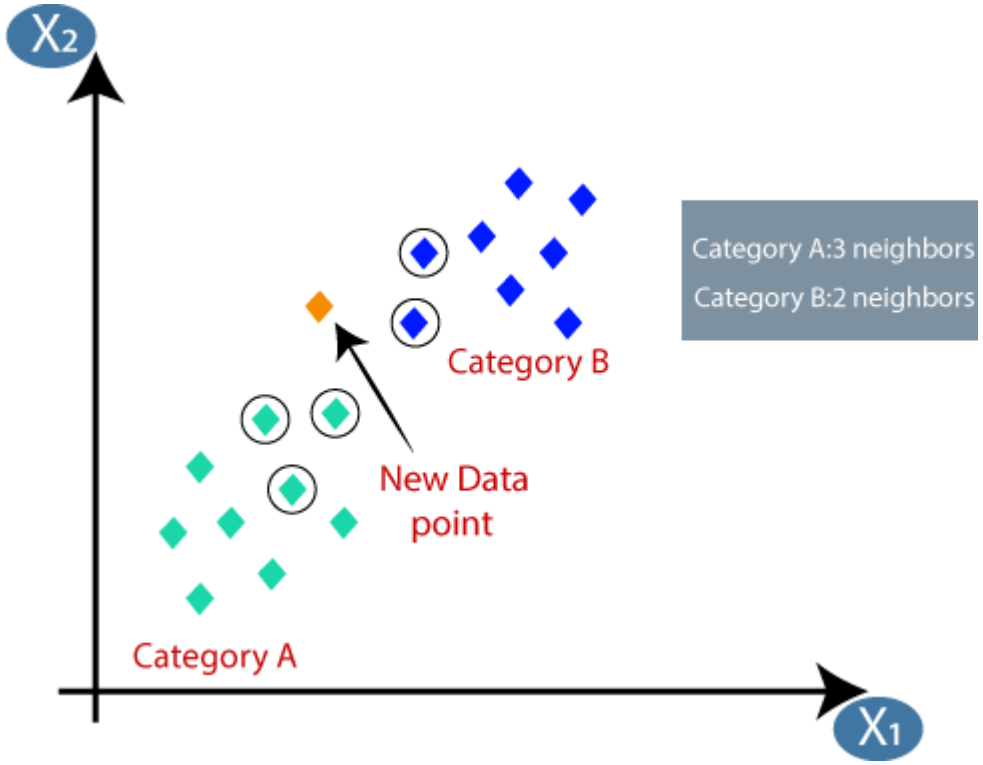
KNN

How does K-NN work?

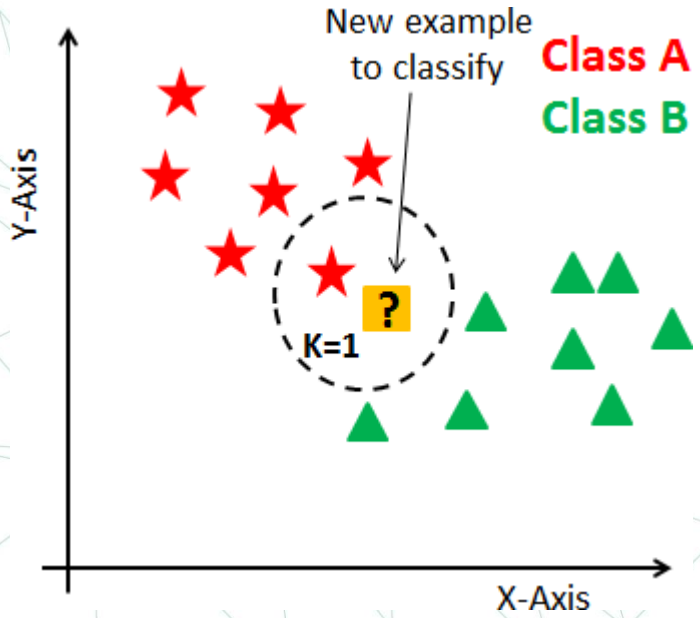
- The K-NN working can be explained on the basis of the below algorithm:
  - **Step-1:** Select the number K of the neighbors
  - **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
  - **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
  - **Step-4:** Among these k neighbors, count the number of the data points in each category.
  - **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.

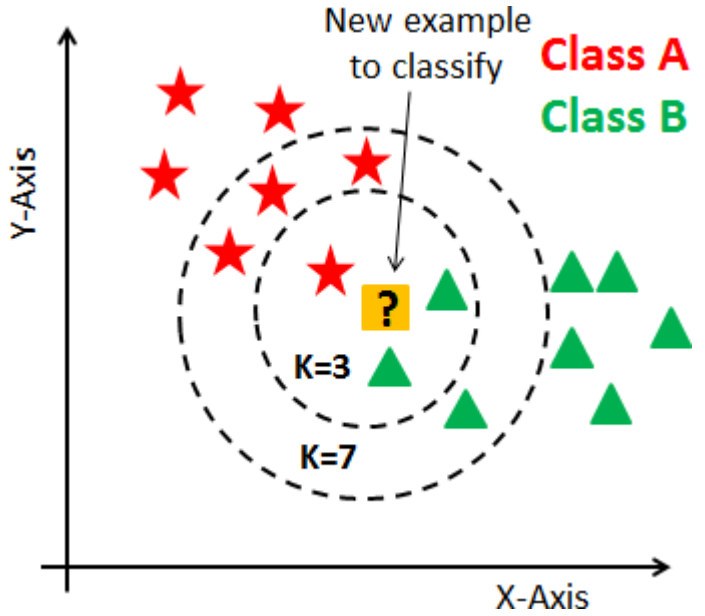
KNN - lazy learner algorithm





How do you Decide the Number of Neighbors in KNN?

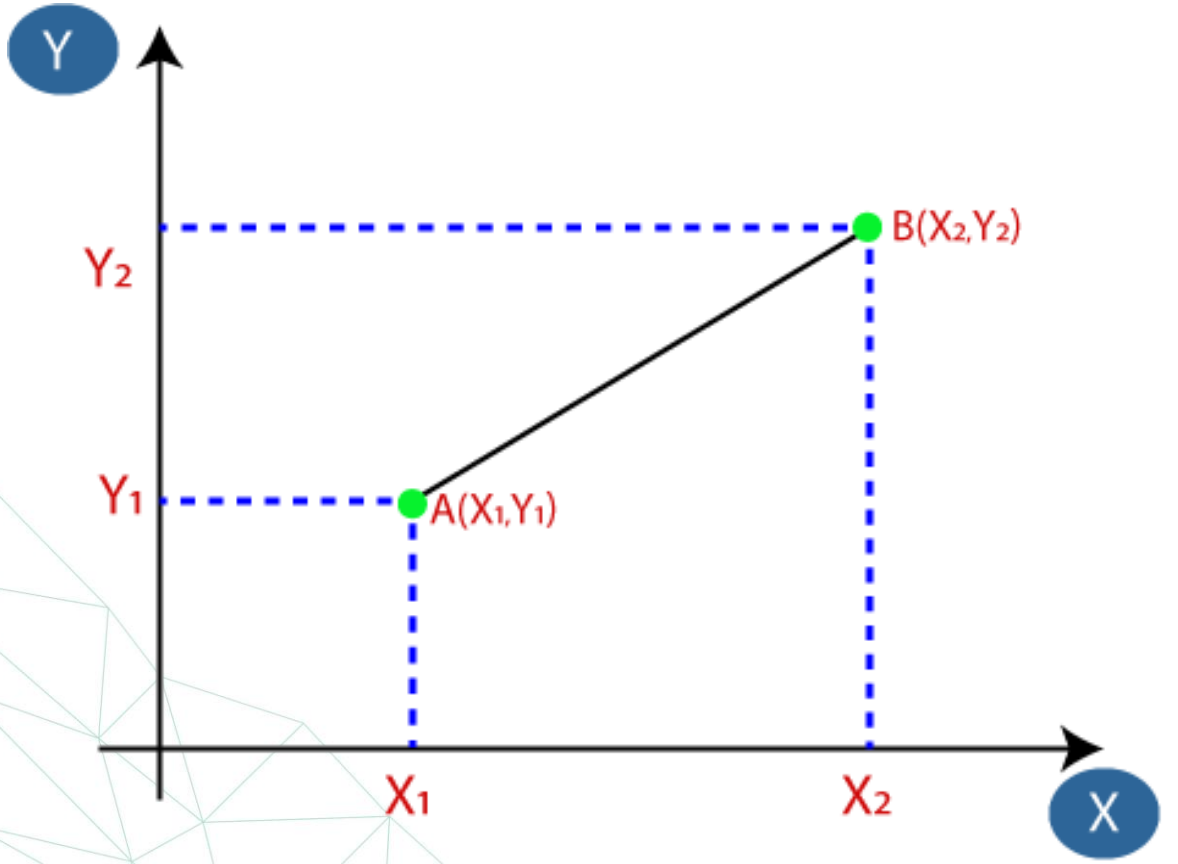




How do you Decide the Number of Neighbors in KNN?

- Defining k can be a balancing act as different values can lead to overfitting or underfitting.
- Lower values of k can have high variance, but low bias[overfitting].
- larger values of k may lead to high bias and lower variance[underfitting].
- The choice of k will largely depend on the input data as data with more outliers or noise will likely perform better with higher values of k.
- Overall, it is recommended to have an odd number for k to avoid ties in classification, and cross-validation tactics can help you choose the optimal k for your dataset.

Euclidean Distance



Euclidean Distance between A_1 and $B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

Advantages

- **Easy to implement**
- **Adapts easily**: As new training samples are added, the algorithm adjusts to account for any new data since all training data is stored into memory.
- **Few hyperparameters**: KNN only requires a k value and a distance metric.

Disadvantages

- **Does not scale well**: Since KNN is a lazy algorithm, it takes up more memory and data storage compared to other classifiers. This can be costly from both a time and money perspective.
- **Curse of dimensionality**: The KNN algorithm tends to fall victim to the curse of dimensionality, which means that it doesn't perform well with high-dimensional data inputs.
- **Prone to overfitting**: Due to the "curse of dimensionality", KNN is also more prone to overfitting. While feature selection and dimensionality reduction techniques are leveraged to prevent this from occurring, the value of k can also impact the model's behavior.

Preparing Data For KNN

- Rescale Data: KNN performs much better if all of the data has the same scale. Normalizing your data to the range between 0 and 1 is a good idea. It may also be a good idea to standardize your data if it has a Gaussian distribution.
- Address Missing Data: Missing data will mean that the distance between samples cannot be calculated. These samples could be excluded or the missing values could be imputed.
- Lower Dimensionality: KNN is suited for lower dimensional data. You can try it on high dimensional data (hundreds or thousands of input variables) but be aware that it may not perform as well as other techniques. KNN can benefit from feature selection that reduces the dimensionality of the input feature space.

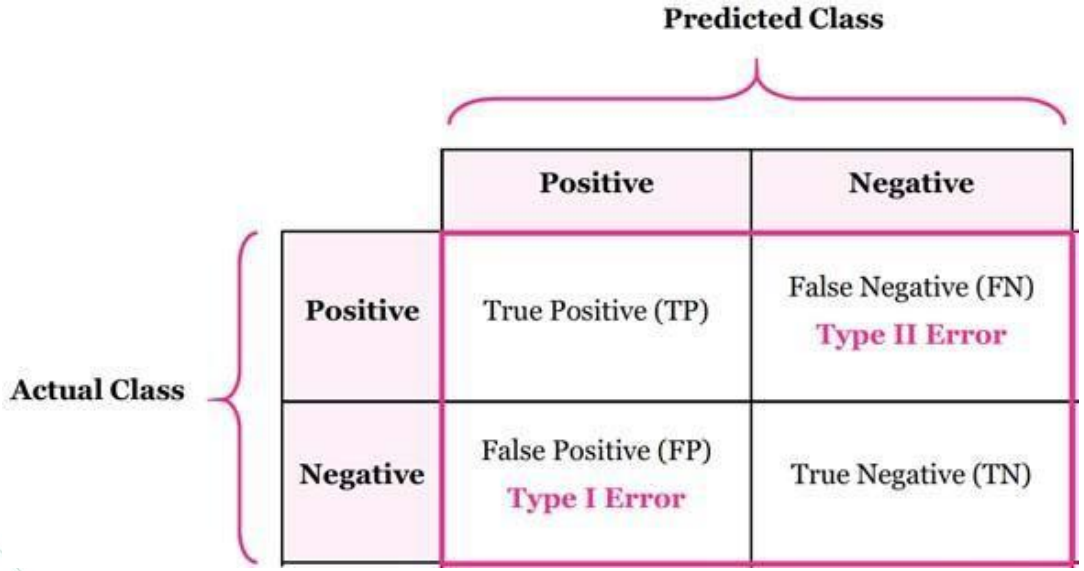
KNN Lab

Classification Metrics

What is Confusion Matrix?

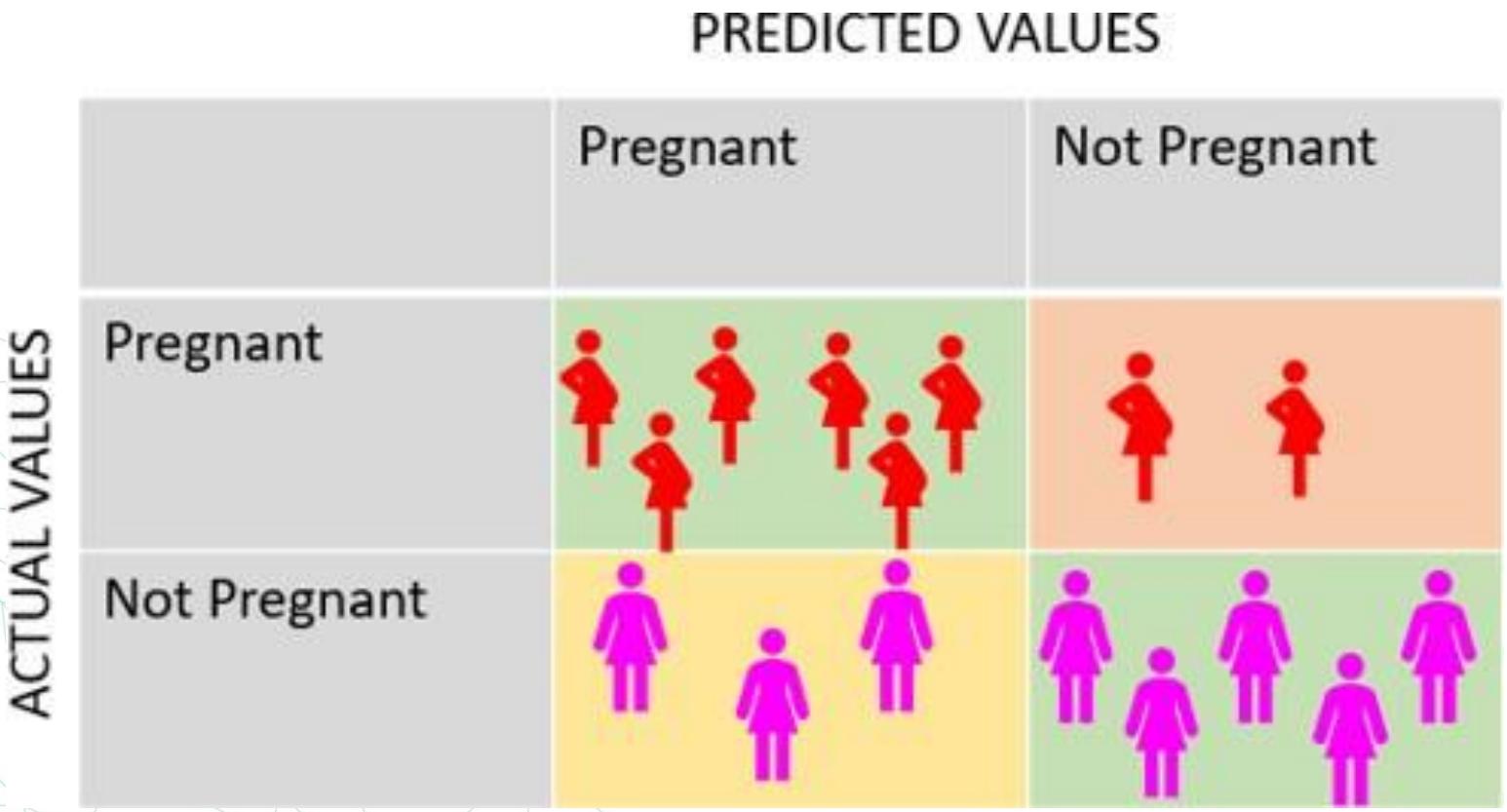
		Predicted Class: Positive	Predicted Class: Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN) **Type II Error**
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)

What is Confusion Matrix?

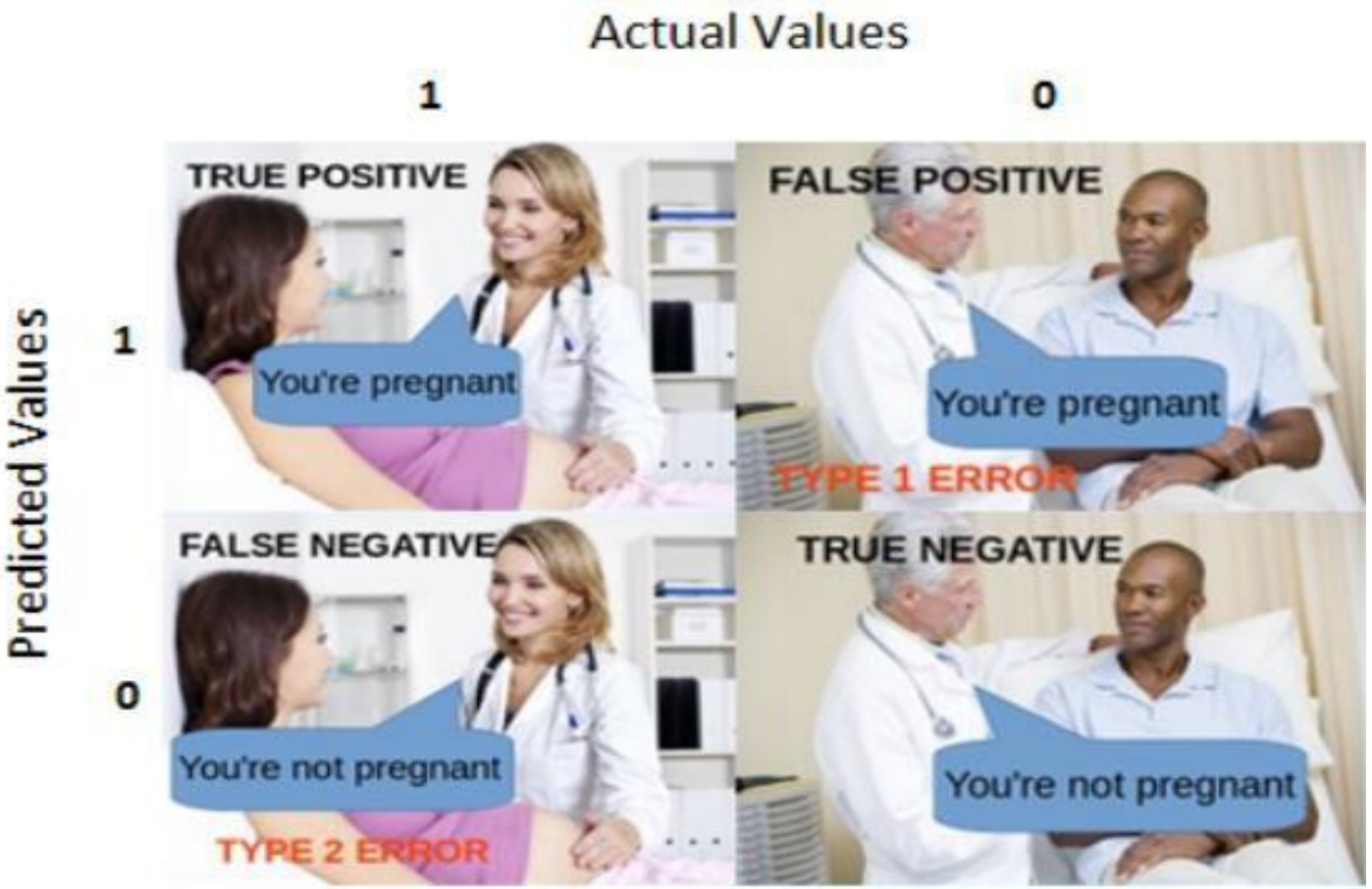


- **True Positive (TP) —** When the model says that the patient has cancer and the patient actually has it
- **False Positive (FP) —** When the model says that the patient has cancer but the patient doesn't have it
- **True Negative (TN)** — When the model says that the patient does not have cancer and the patient actually doesn't have it
- **False Negative (FN)** — When the model says that the patient doesn't have cancer but the patient actually has it. *We don't want this, do we?*

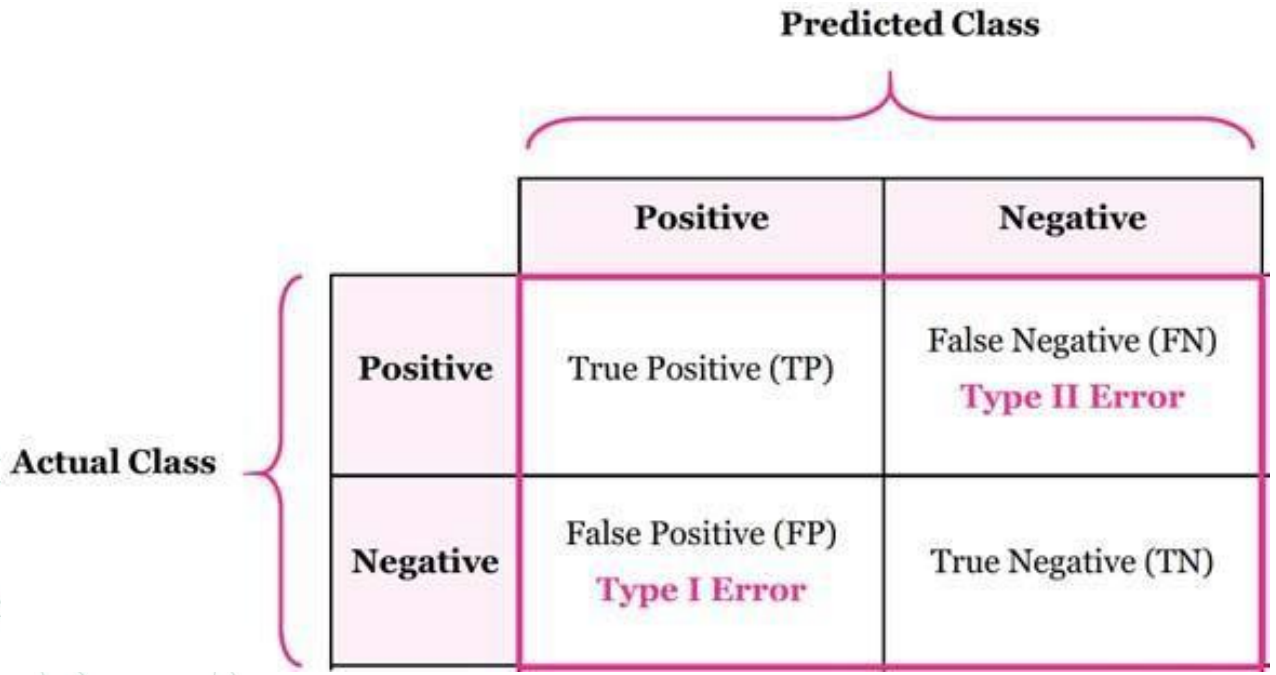
What is Confusion Matrix?



What is Confusion Matrix?



Performance evaluation Measures



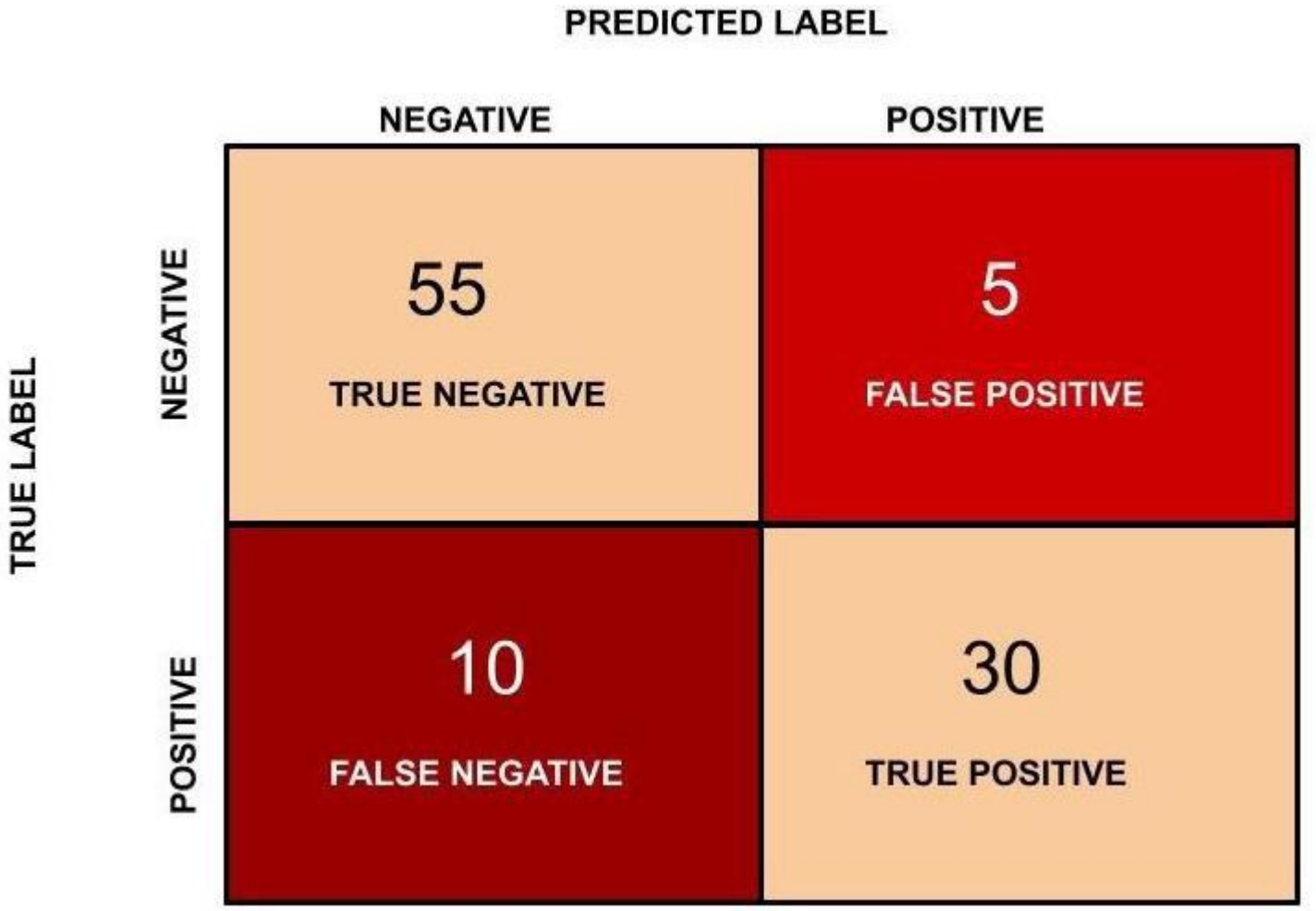
Accuracy = (TP+TN) / (TP+FP+FN+TN)

Exercise

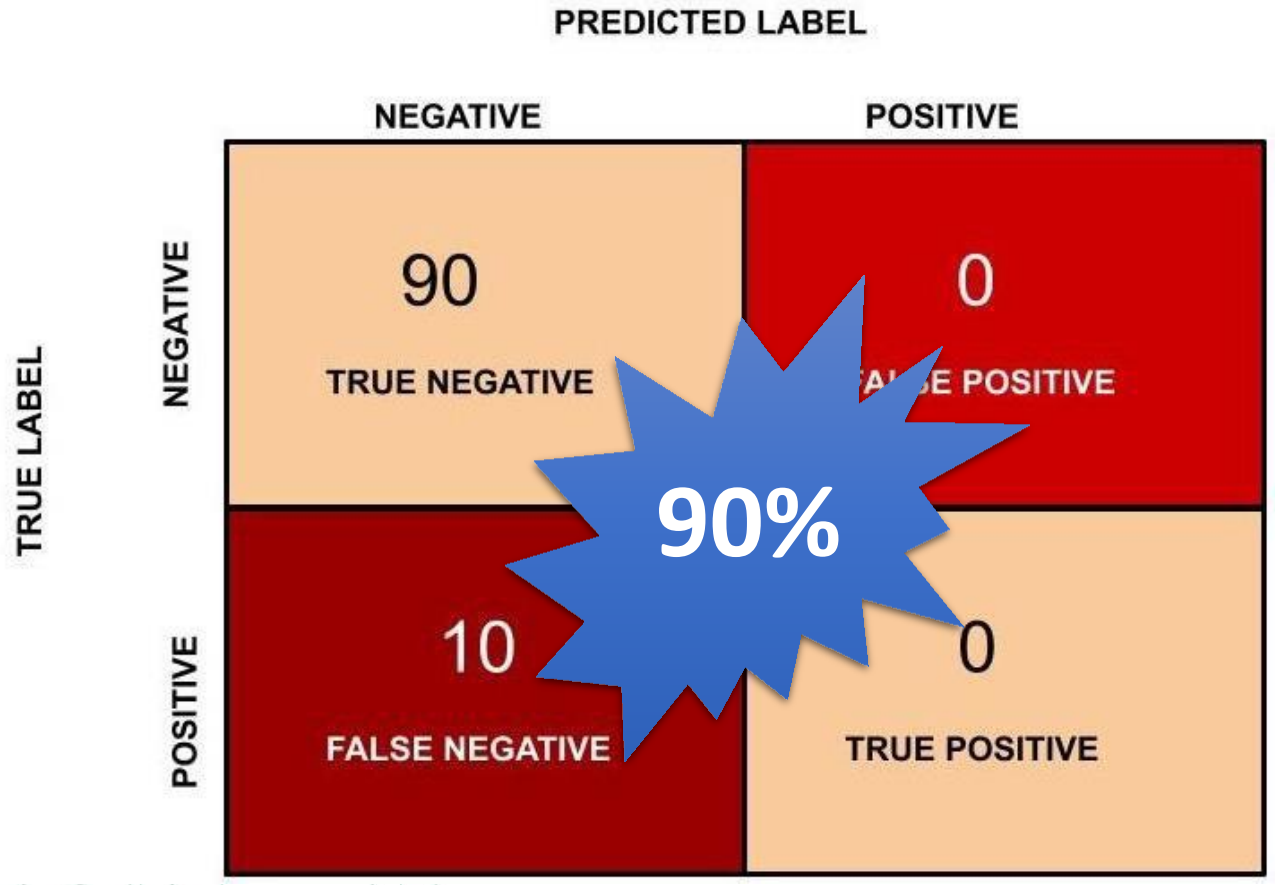
- We select 100 people which includes pregnant women, not pregnant women and men with fat belly. Let us assume out of this 100 people 40 are pregnant and the remaining 60 people include not pregnant women and men with fat belly. We now use a machine learning algorithm to predict the outcome.

- Out of 40 pregnant women 30 pregnant women are classified correctly and the remaining 10 pregnant women are classified as not pregnant by the machine learning algorithm.
- On the other hand, out of 60 people in the not pregnant category, 55 are classified as not pregnant and the remaining 5 are classified as pregnant.

Example

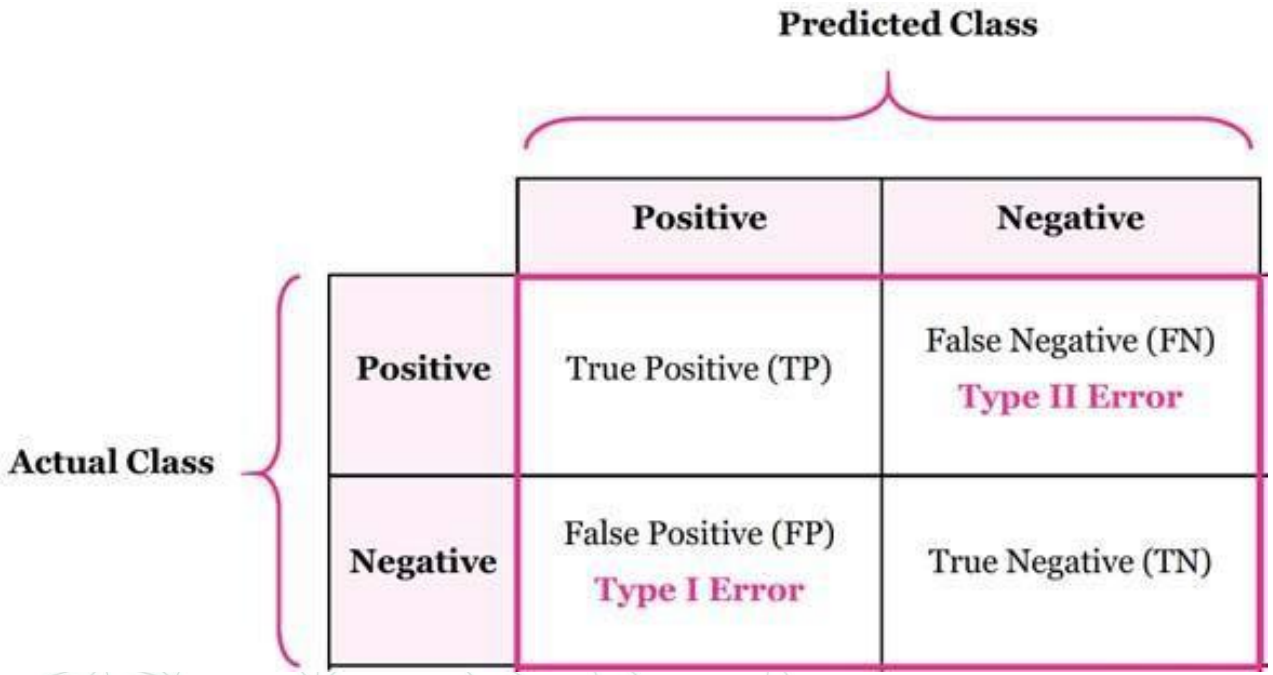

PREDICTED LABEL
NEGATIVE
POSITIVE
TRUE LABEL
NEGATIVE
POSITIVE
55
TRUE NEGATIVE
5
FALSE POSITIVE
10
FALSE NEGATIVE
30
TRUE POSITIVE

Is accuracy the best measure?



Accuracy = (TP+TN) / (TP+FP+FN+TN)

Performance evaluation Measures



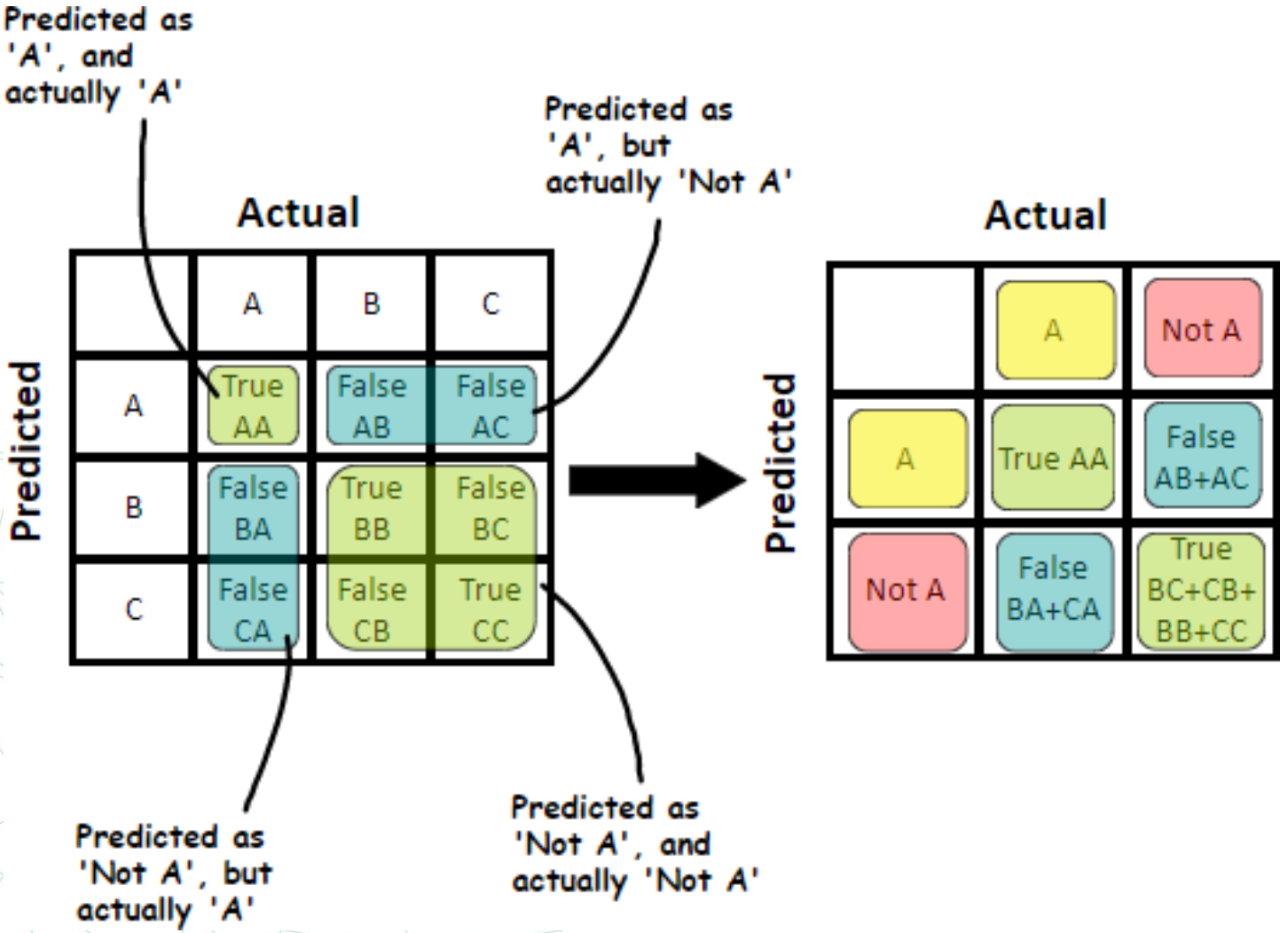
Accuracy = (TP+TN) / (TP+FP+FN+TN)

Precision = TP / (TP+FP)

Recall = TP / TP+FN

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

MultiClass Classification

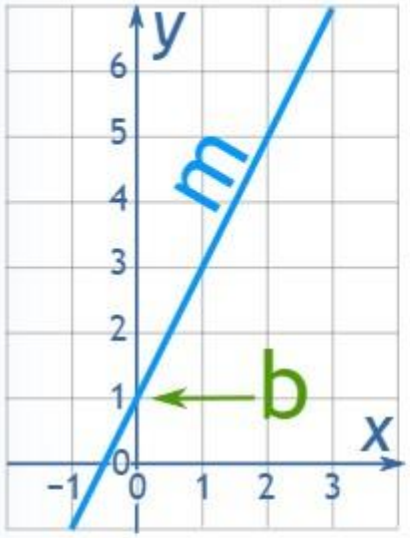


- In case of stock market crash or not prediction, false negative will be more important as if stock is predicted as not crash so you didn't withdraw money but actual it is crashed as result you lost money so here you are in loss(false negative),
- Second case :- false positive where you predict stock market crash so you withdraw money but actual it is not crash so result you were not able to earn profit becoz you withdraw money
- . Conclusion:- false negative is more important in case of stock market crash prediction.
- False negative is riskier;because if the person has infected with disease but predicted as negative it will be dangerous because it is contagious.
- As classifier assumes it as Spam and moved to trash, while it is not spam, then the user may misses the email and may causes a lot of trouble. While if it is spam but classifier failed to detect, then user will see the spam then it is not a big deal, use

- （1）In case of stock market crash or not prediction, false negative will be more important as if stock is predicted as not crash so you didn't withdraw money but actual it is crashed as result you lost money so here you are in loss(false negative), Second case :- false positive where you predict stock market crash so you withdraw money but actual it is not crash so result you were not able to earn profit becoz you withdraw money . Conclusion:- false negative is more important in case of stock market crash prediction.
- （2）False negative is riskier;because if the person has infected with disease but predicted as negative it will be dangerous because it is contagious.
- （3）Credit Card Fraud
- (4)As classifier assumes it as Spam and moved to trash, while it is not spam, then the user may misses the email and may causes a lot of trouble. While if it is spam but classifier failed to detect, then user will see the spam then it is not a big deal, user will delete it by one click.

Lines and Gradient

Equation of a Straight Line



$$y = mx + b$$

Slope or Gradient

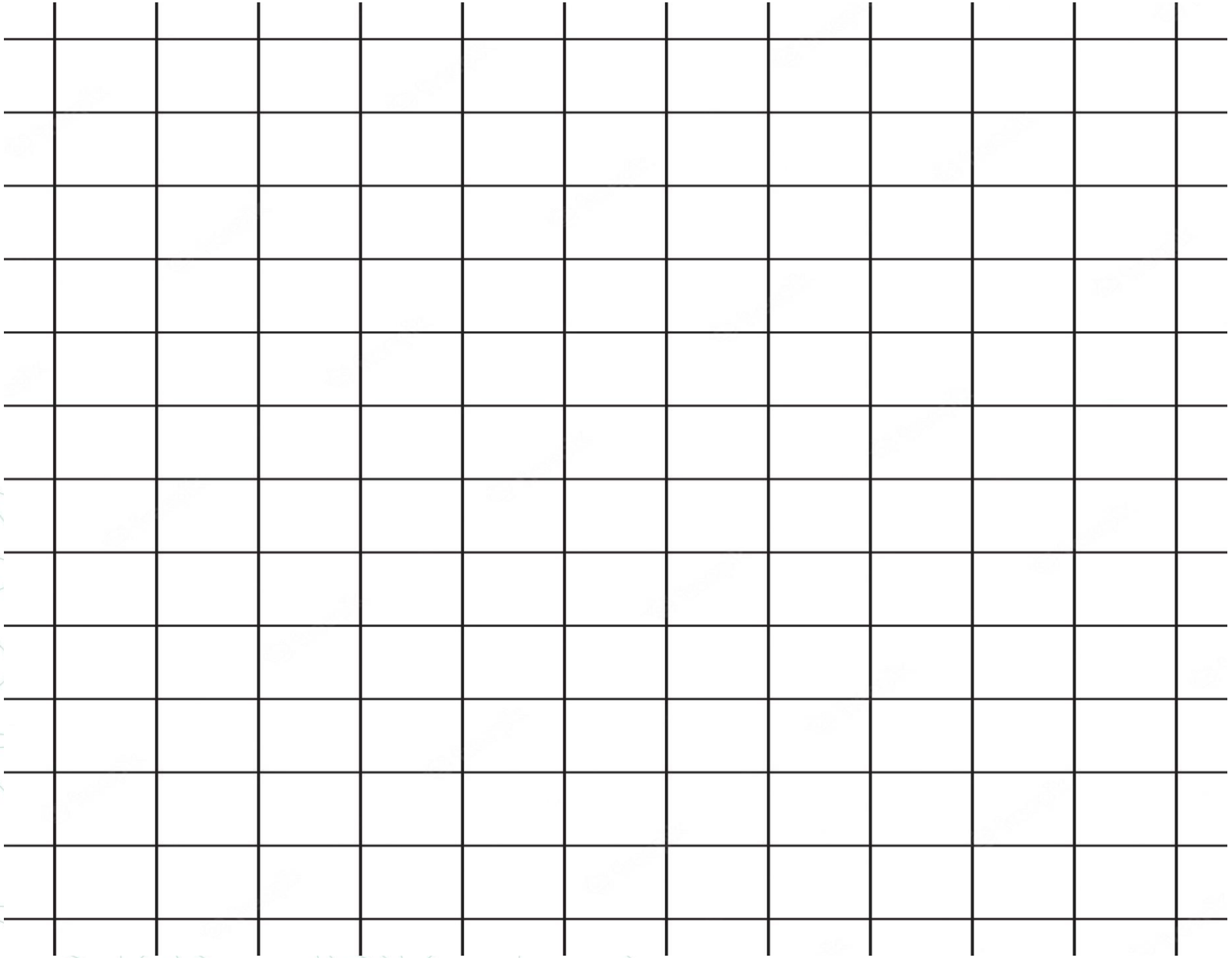
y value when **x=0** *(see Y Intercept)*

y = how far up

x = how far along

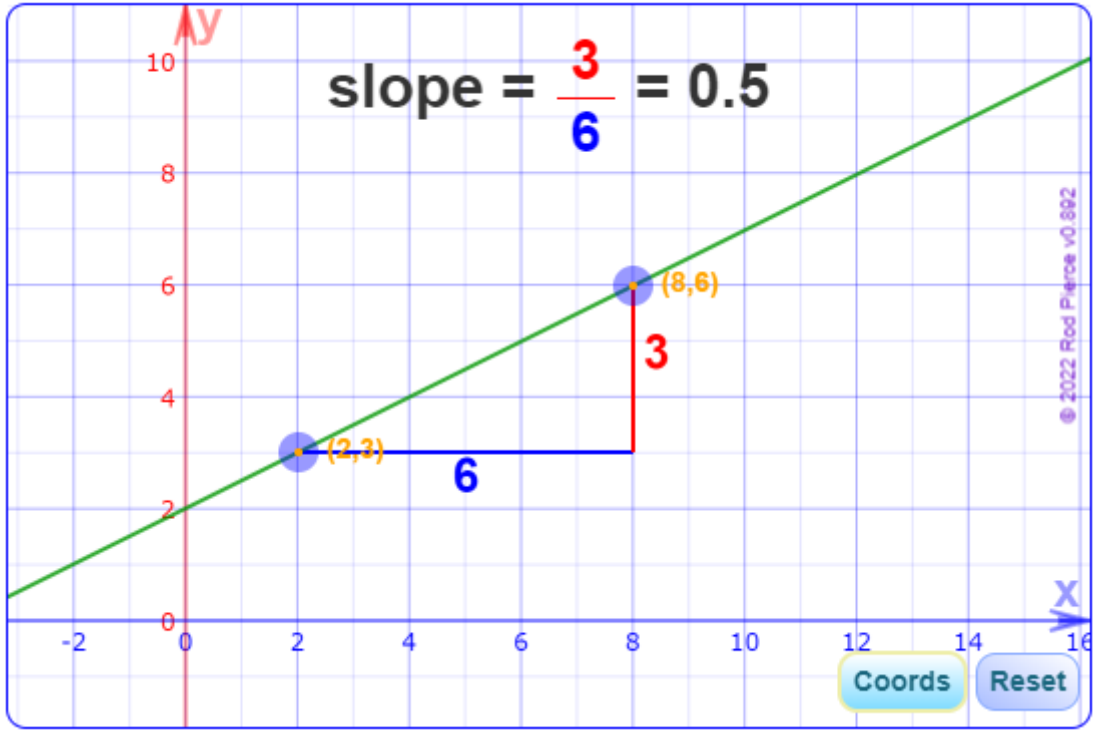
m = Slope or Gradient (how steep the line is)

b = value of **y** when **x=0**

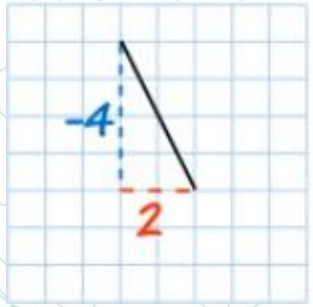


Gradient (Slope) of a Straight Line

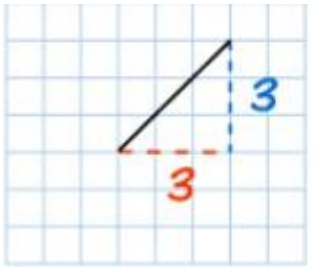
Divide the **change in height** by the **change in horizontal distance**



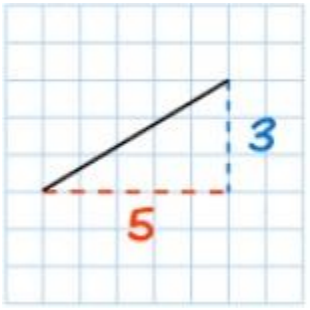
Gradient (Slope) of a Straight Line



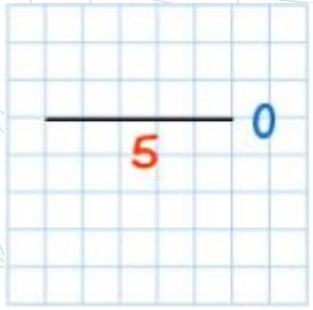
Gradient = $\frac{-4}{2} = -2$



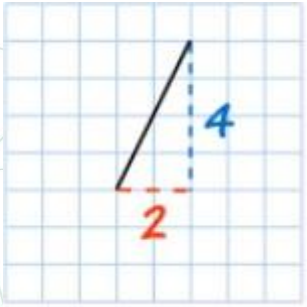
The Gradient = $\frac{3}{3} = 1$



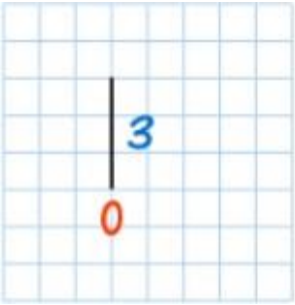
The Gradient = $\frac{3}{5} = 0.6$



Gradient = $\frac{0}{5} = 0$



The Gradient = $\frac{4}{2} = 2$

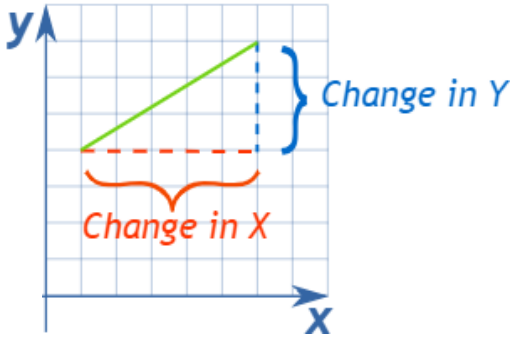


Gradient = $\frac{3}{0}$ = undefined

Derivatives

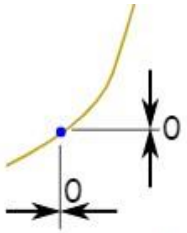
It is all about slope!

$$\text{Slope} = \frac{\text{Change in Y}}{\text{Change in X}}$$



But how do we find the slope **at a point**?

There is nothing to measure!

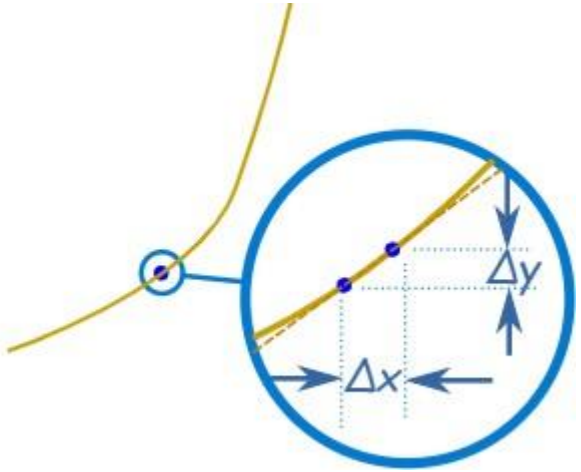


$$\text{slope} = \frac{0}{0} = ???$$

Sometimes we can't work something out directly ... but we **can** see what it should be as we get closer and closer!

But with derivatives we use a small difference ...

... then have it **shrink towards zero**.



Derivatives

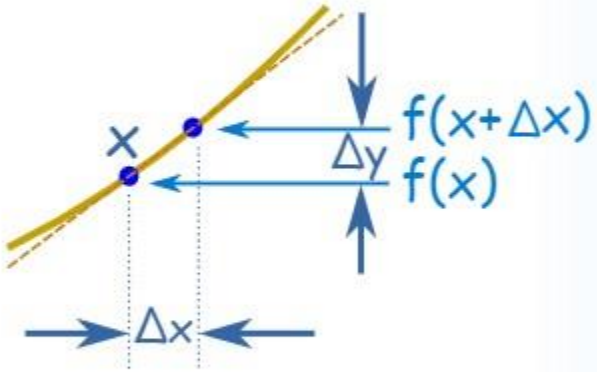
To find the derivative of a function $y = f(x)$ we use the slope formula:

$$\text{Slope} = \frac{\text{Change in Y}}{\text{Change in X}} = \frac{\Delta y}{\Delta x}$$

And (from the diagram) we see that:

x changes from x to $x+\Delta x$

y changes from $f(x)$ to $f(x+\Delta x)$



Now follow these steps:

- Fill in this slope formula: $\frac{\Delta y}{\Delta x} = \frac{f(x+\Delta x) - f(x)}{\Delta x}$
- Simplify it as best we can
- Then make **Δx** shrink towards zero.

Derivatives

We know $\mathbf{f(x) = x^2}$, and we can calculate $\mathbf{f(x+\Delta x)}$:

Start with: $\mathbf{f(x+\Delta x) = (x+\Delta x)^2}$

Expand $(x + \Delta x)^2$: $\mathbf{f(x+\Delta x) = x^2 + 2x\,\Delta x + (\Delta x)^2}$

The slope formula is: $\dfrac{f(x+\Delta x) - f(x)}{\Delta x}$

Put in $\mathbf{f(x+\Delta x)}$ and $\mathbf{f(x)}$: $\dfrac{x^2 + 2x\,\Delta x + (\Delta x)^2 - x^2}{\Delta x}$

Simplify (x^2 and $-x^2$ cancel): $\dfrac{2x\,\Delta x + (\Delta x)^2}{\Delta x}$

Simplify more (divide through by Δx): $= 2x + \Delta x$

Then, **as Δx heads towards 0** we get: $= 2x$

Derivatives

We write **dx** instead of **"Δx heads towards 0"**.

And "the derivative of" is commonly written $\frac{d}{dx}$ like this:

$$\frac{d}{dx}x^2 = 2x$$

"The derivative of x^2 *equals* $2x$*"*

or simply *"d dx of* x^2 *equals* $2x$*"*

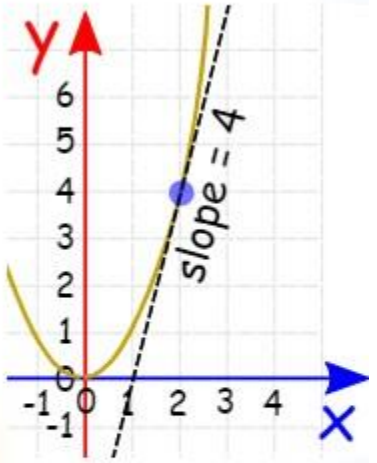
Derivatives

So what does $\frac{d}{dx}x^2 = 2x$ mean?

It means that, for the function x^2, the slope or "rate of change" at any point is **2x**.

So when **x=2** the slope is **2x = 4**, as shown here:

Or when **x=5** the slope is **2x = 10**, and so on.



Note: f'(x) can also be used for "the derivative of":

$$f'(x) = 2x$$

"The derivative of f(x) equals 2x"

or simply *"f-dash of x equals 2x"*

Derivatives

Example: What is $\frac{d}{dx}x^3$?

We know $\mathbf{f(x) = x^3}$, and can calculate $\mathbf{f(x+\Delta x)}$:

Start with: $\mathbf{f(x+\Delta x) = (x+\Delta x)^3}$

Expand $(x + \Delta x)^3$: $\mathbf{f(x+\Delta x) = x^3 + 3x^2\,\Delta x + 3x\,(\Delta x)^2 + (\Delta x)^3}$

The slope formula: $\frac{f(x+\Delta x) - f(x)}{\Delta x}$

Put in $\mathbf{f(x+\Delta x)}$ and $\mathbf{f(x)}$: $\frac{x^3 + 3x^2\,\Delta x + 3x\,(\Delta x)^2 + (\Delta x)^3 - x^3}{\Delta x}$

Simplify (x^3 and $-x^3$ cancel): $\frac{3x^2\,\Delta x + 3x\,(\Delta x)^2 + (\Delta x)^3}{\Delta x}$

Simplify more (divide through by Δx): $3x^2 + 3x\,\Delta x + (\Delta x)^2$

Then, **as Δx heads towards 0** we get: $3x^2$

Result: the derivative of $\mathbf{x^3}$ is $\mathbf{3x^2}$

Derivatives

"Shrink towards zero" is actually written as a limit like this:

$$f'(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

"The derivative of **f** equals
the limit as Δx goes to zero of f(x+Δx) - f(x) over Δx"

Or sometimes the derivative is written like this (explained on Derivatives as dy/dx):

$$\frac{dy}{dx} = \frac{f(x+dx) - f(x)}{dx}$$

The process of finding a derivative is called "differentiation".

You **do** differentiation ... to **get** a derivative.

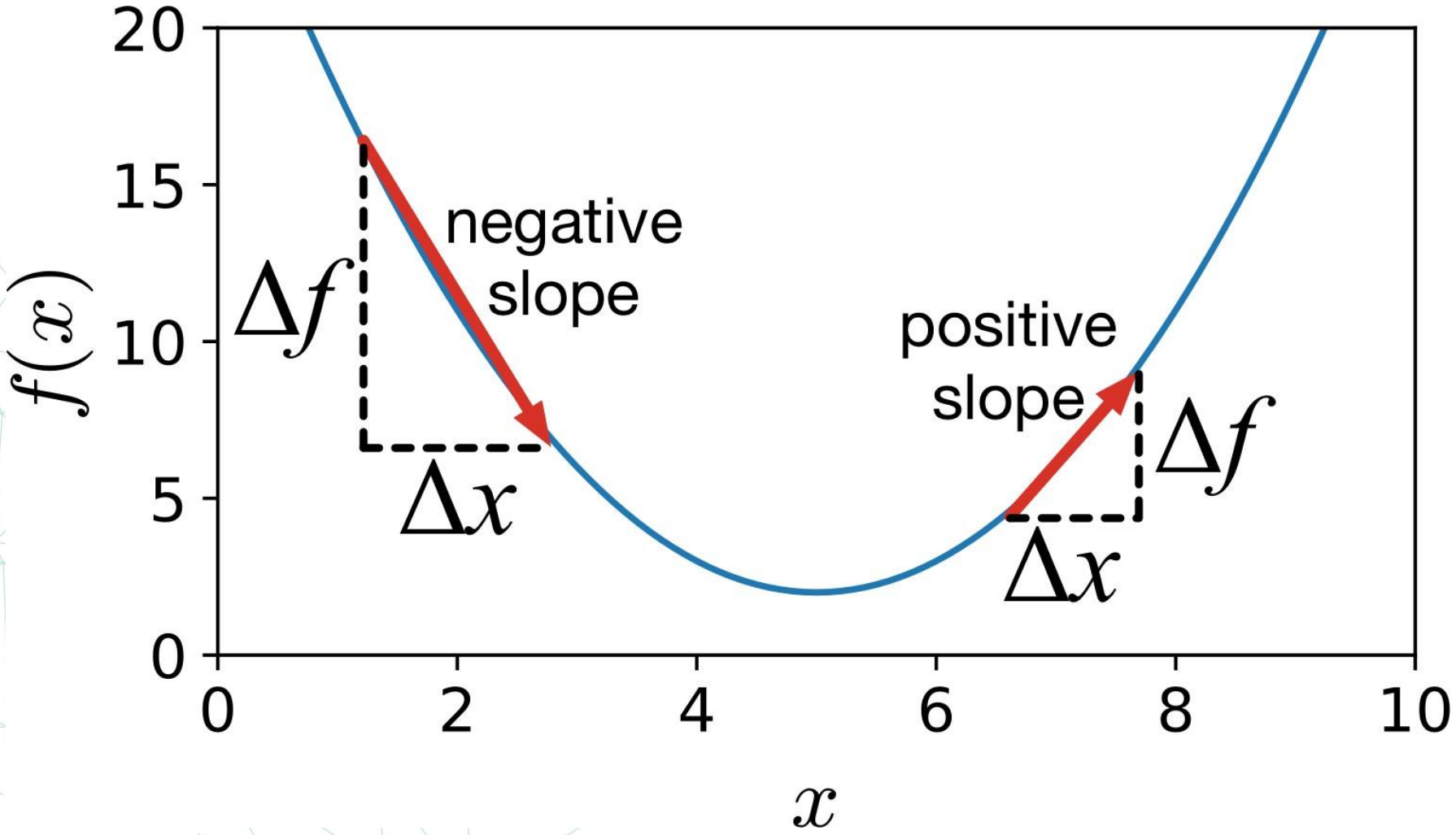
Derivatives

- https://www.mathsisfun.com/calculus/derivative-plotter.html
- https://www.mathsisfun.com/calculus/slope-function-point.html
- https://www.mathsisfun.com/calculus/derivatives-rules.html

Derivative Rules

Common Functions	Function	Derivative
Constant	c	0
Line	x	1
	ax	a
Square	x^2	2x
Square Root	√x	$(½)x^{-½}$
Exponential	e^x	e^x
	a^x	$\ln(a)\, a^x$
Logarithms	ln(x)	1/x

Derivative

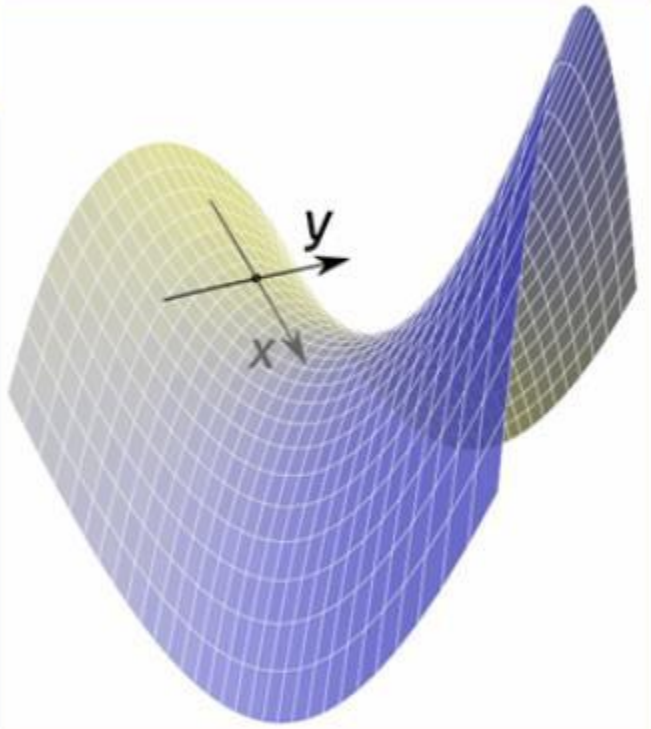

20
15
10
5
0
0
2
4
6
8
10
$f(x)$
x
negative slope
positive slope
Δf
Δx
Δf
Δx

Partial Derivatives

Example: a function for a surface that depends on two variables **x** and **y**

When we find the slope in the **x** direction (while keeping **y** fixed) we have found a partial derivative.

Or we can find the slope in the **y** direction (while keeping **x** fixed).



Partial Derivatives

But what about a function of **two variables** (x and y):

$$f(x, y) = x^2 + y^3$$

We can find its **partial** derivative **with respect to x** when we treat **y as a constant** (imagine y is a number like 7 or something):

$$f'_x = 2x + 0 = 2x$$

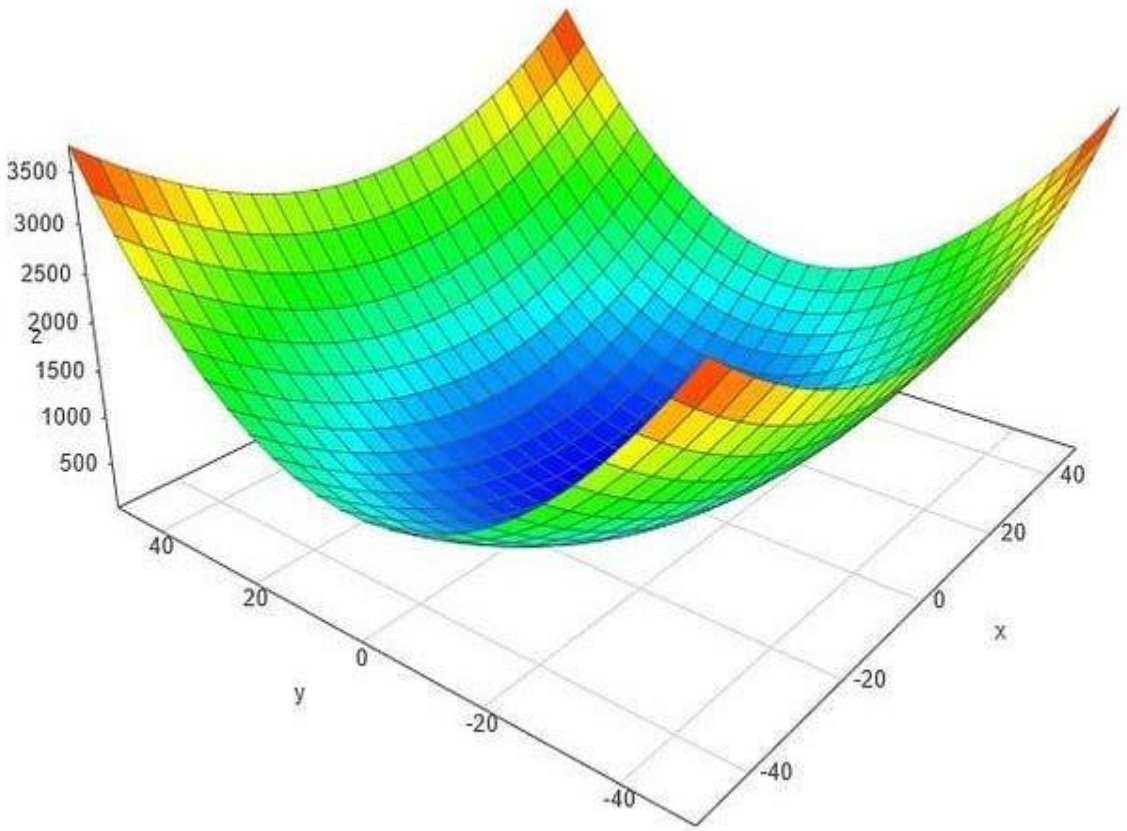
Explanation:

- *the derivative of x^2 (with respect to x) is 2x*
- *we **treat y as a constant**, so y^3 is also a constant (imagine y=7, then 7^3=343 is also a constant), and the derivative of a constant is 0*

To find the partial derivative **with respect to y**, we treat **x as a constant**:

$$f'_y = 0 + 3y^2 = 3y^2$$

$$f(x) = 0.5x^2 + y^2$$



Let's assume we are interested in a gradient at point p(10,10):

$$\frac{\partial f(x,y)}{\partial x} = x, \quad \frac{\partial f(x,y)}{\partial y} = 2y$$

so consequently:

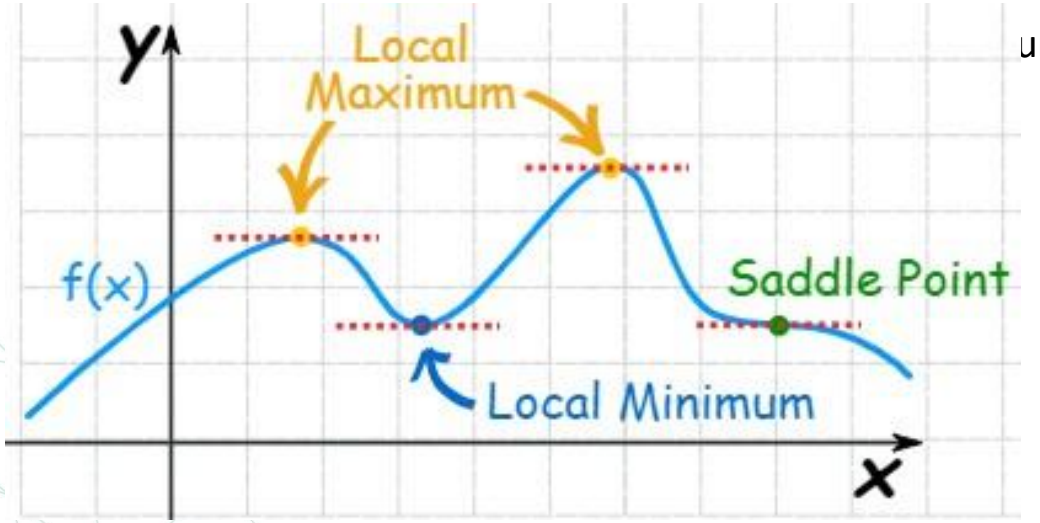
$$\nabla f(x,y) = \begin{bmatrix} x \\ 2y \end{bmatrix}$$

$$\nabla f(10,10) = \begin{bmatrix} 10 \\ 20 \end{bmatrix}$$

By looking at these values we conclude that the slope is twice steeper along the y axis.

Finding Maxima and Minima using Derivatives

- Where is a function at a high or low point?
- Calculus can help!

- 

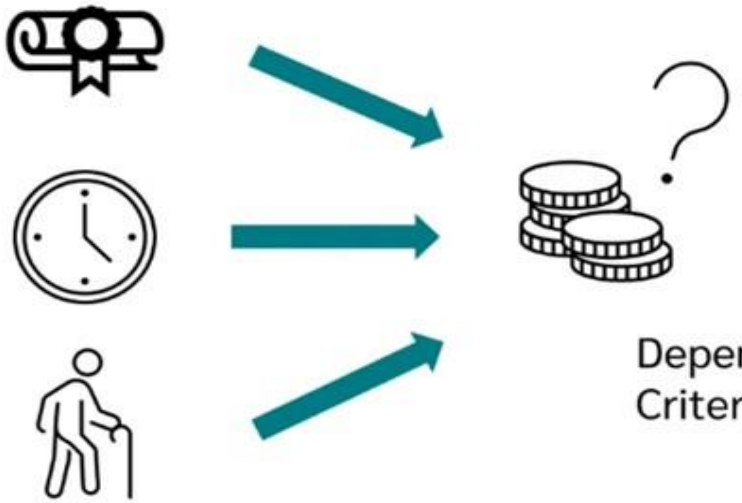
um is always where the function

Where does it flatten out? Where the **slope is zero**.
Where is the slope zero? The **Derivative** tells us!

Linear Regression

Regression Analysis

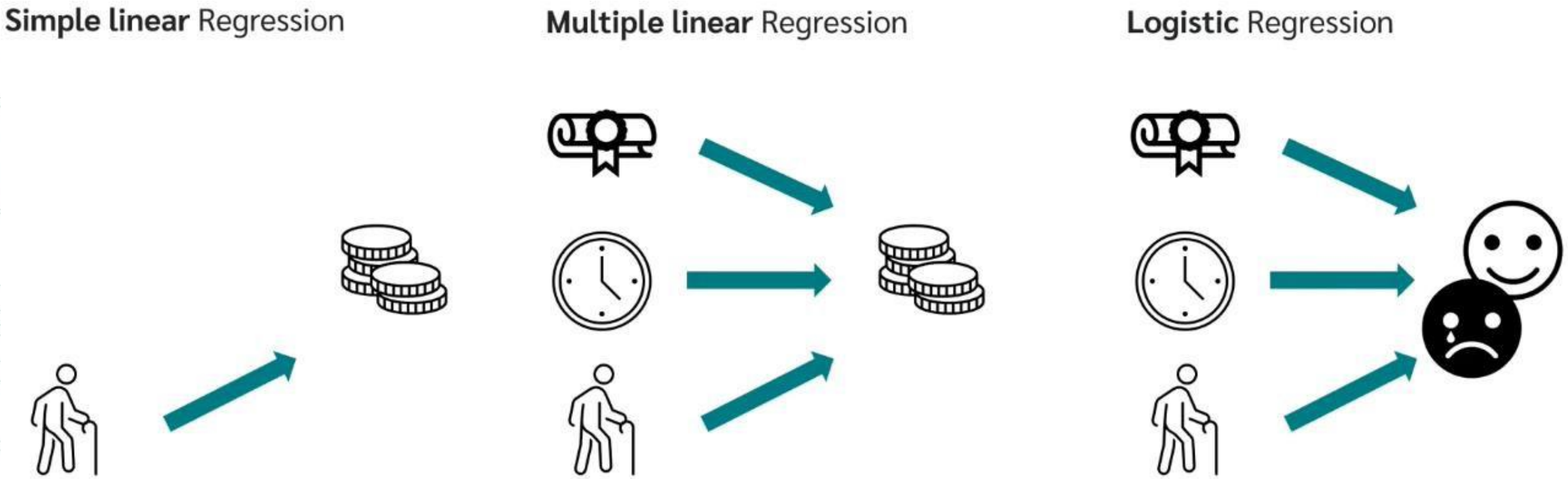
A **regression analysis** makes it possible to infer or predict another variable on the basis of one or more variables.



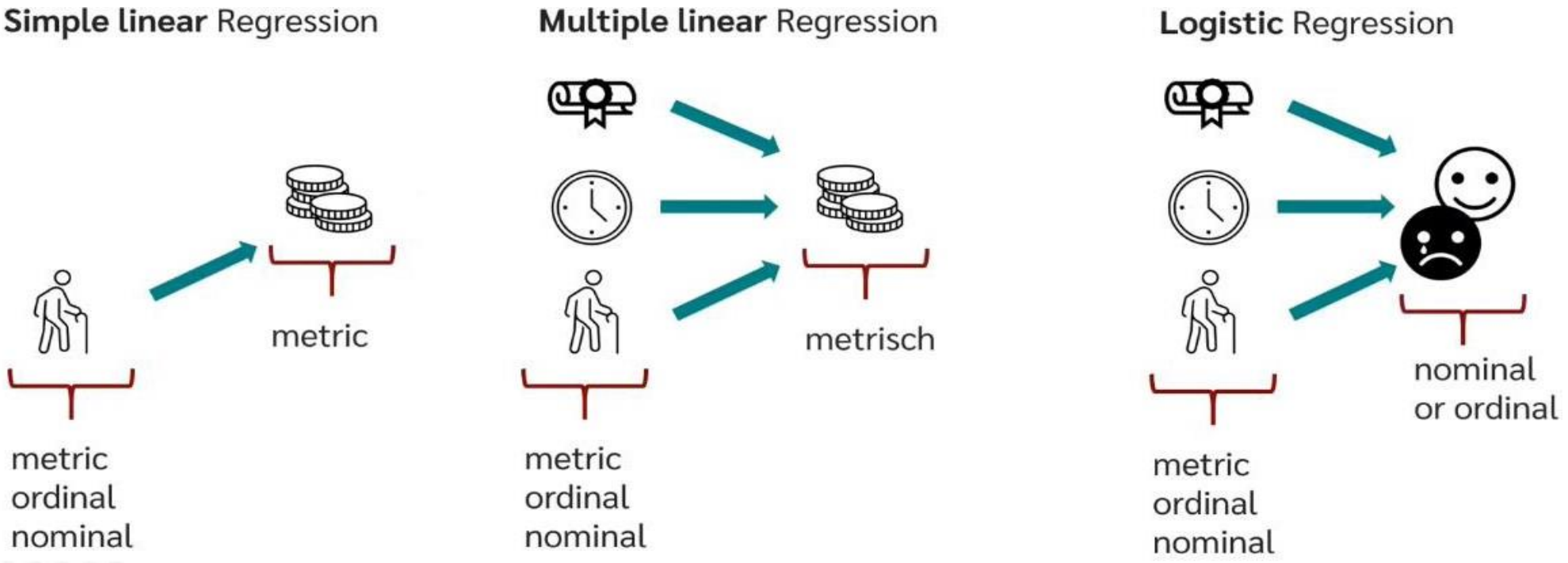
Dependent variables
Criterion

Independent variables
Predictors

Forms of Regression Analysis



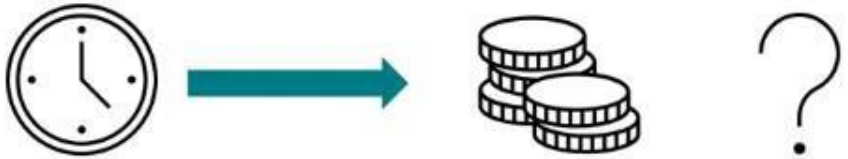
Forms of Regression Analysis



Forms of Regression Analysis

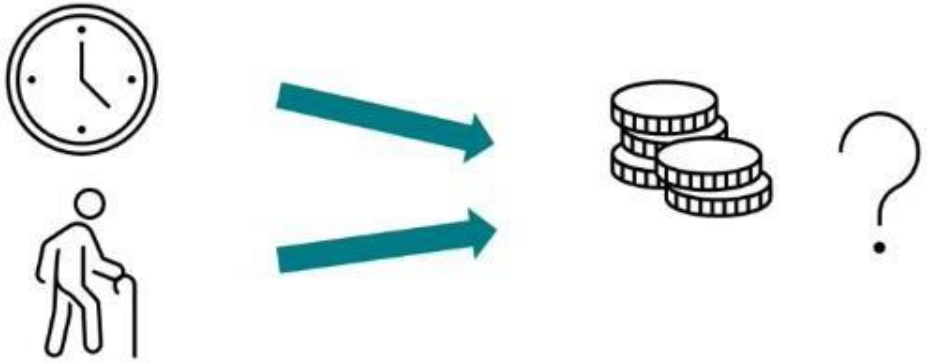
Simple linear regression

Does **the weekly working time** have an influence on the **hourly salary** of employees?

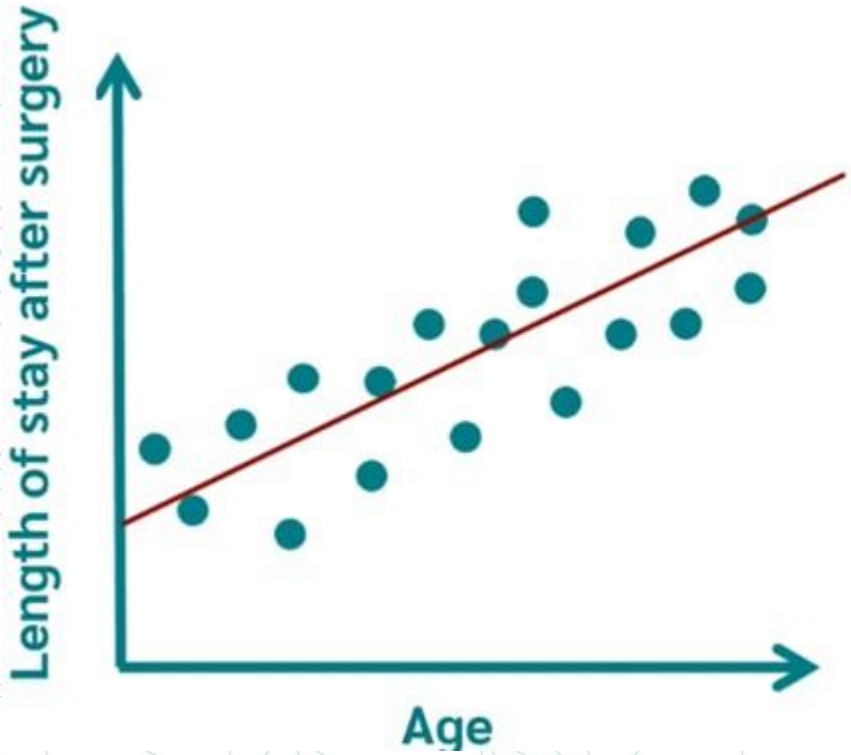


Multiple linear regression

Do the **weekly working hours** and the **age** of employees have an influence on their **hourly salary**?



Simple Linear Regression



Estimated length of stay — Age

$$\hat{y} = b \cdot x + a$$

$$\hat{y} = 0.14 \cdot x + 1.2$$

$$5.82 = 0.14 \cdot 33 + 1.2$$

The model

A simple linear regression model predicts the output as a linear function of the input feature x:

$$f(x; w_0, w_1) = w_0 + w_1 x$$

We refer to w_0 and w_1 as the *parameters* of the model.

To choose w_0 and w_1, we are given a data set of previous input-output measurements:

$$\left\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots (x^{(N)}, y^{(N)})\right\}$$

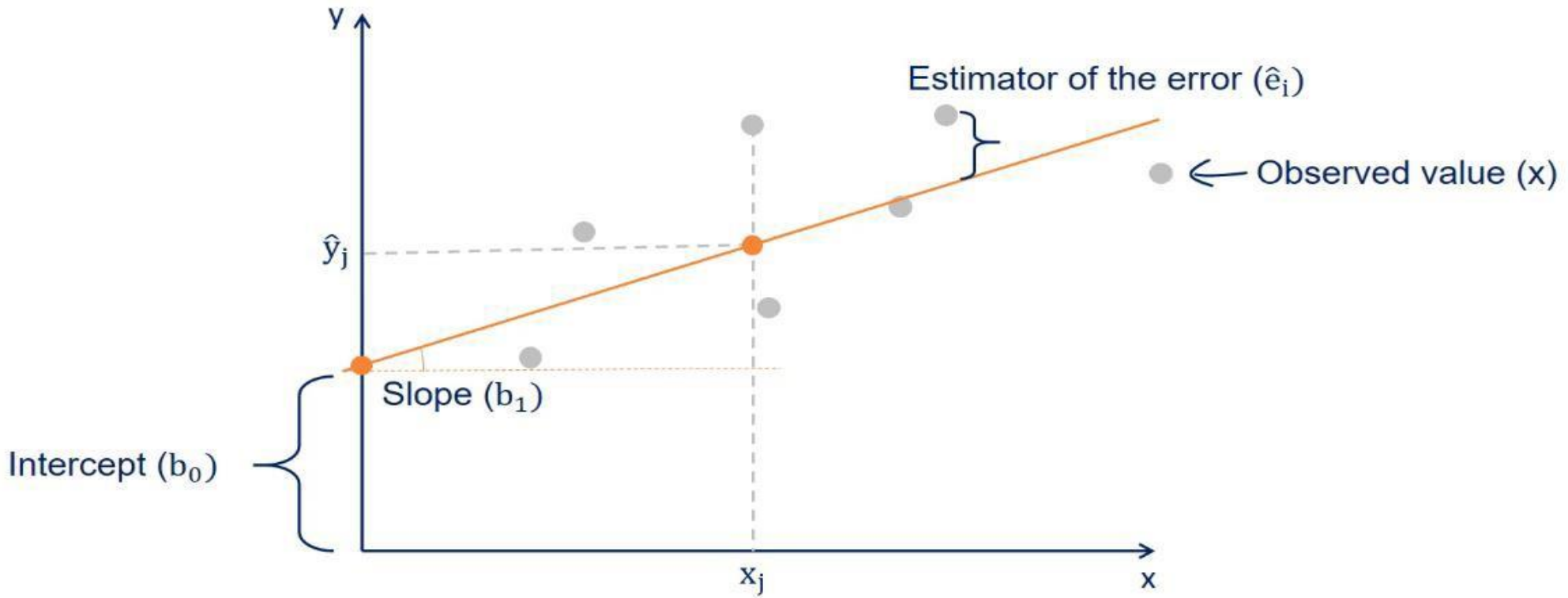
I will sometimes just write this as:

$$\left\{(x^{(n)}, y^{(n)})\right\}_{n=1}^{N}$$

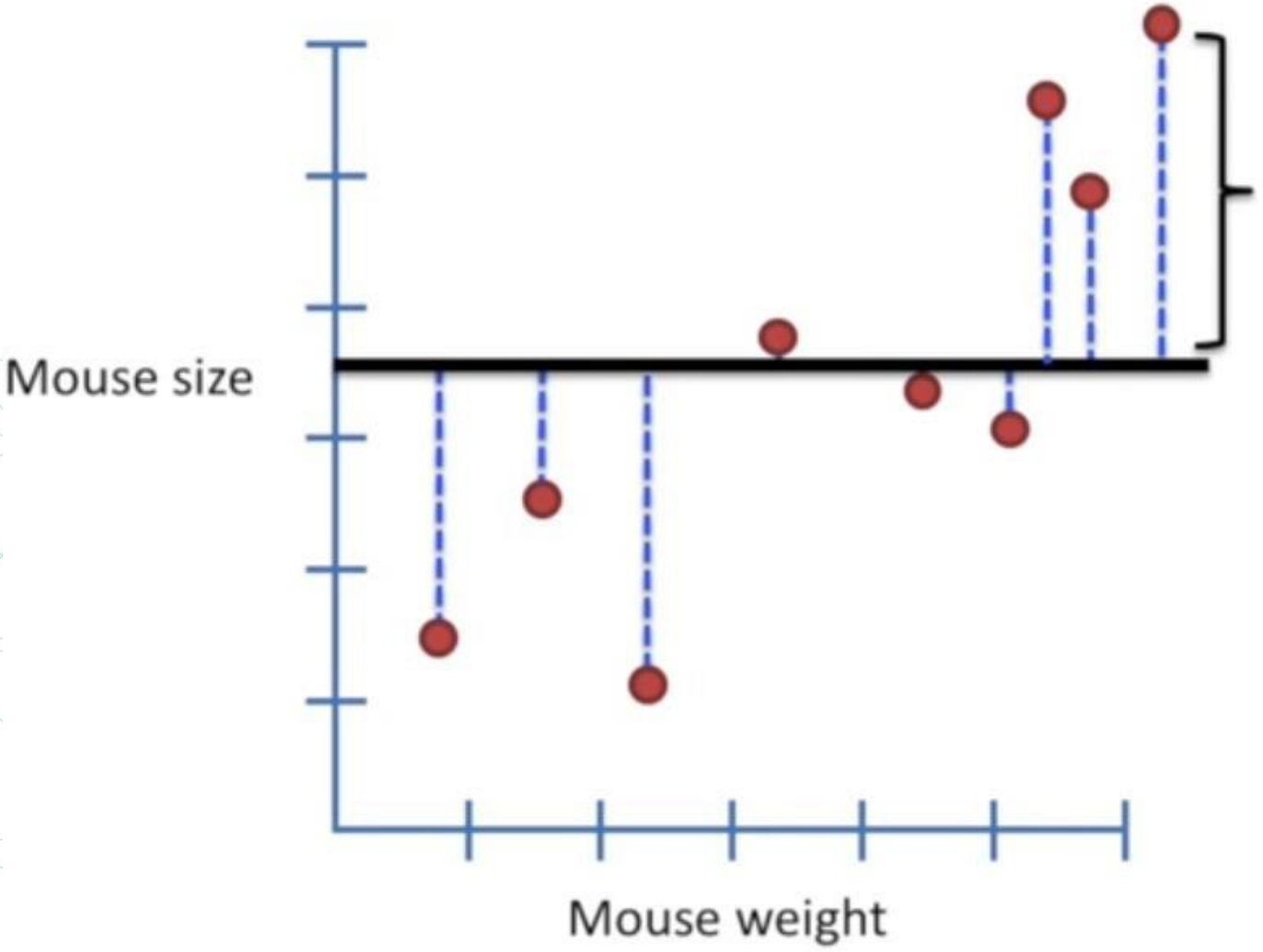
How do we choose w_0 and w_1 based on the data? We need some way to measure the "goodness" or "badness" of the parameters, given the data.

Simple Linear Regression

$$\hat{y}_i = b_0 + b_1 x_i$$



Simple Linear Regression

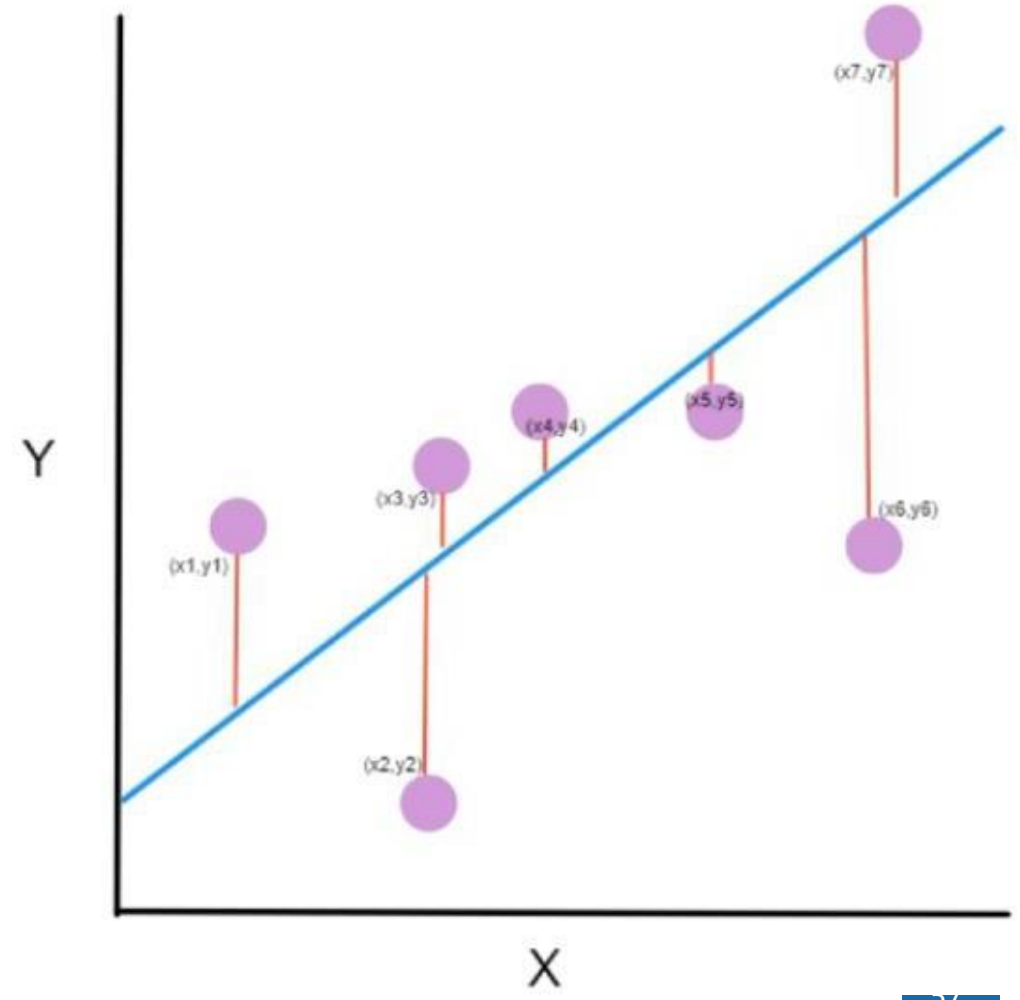


Second, measure the distance from the line to the data, square each distance, and then add them up.

Terminology alert!
The distance from a line to a data point is called a "**residual**".

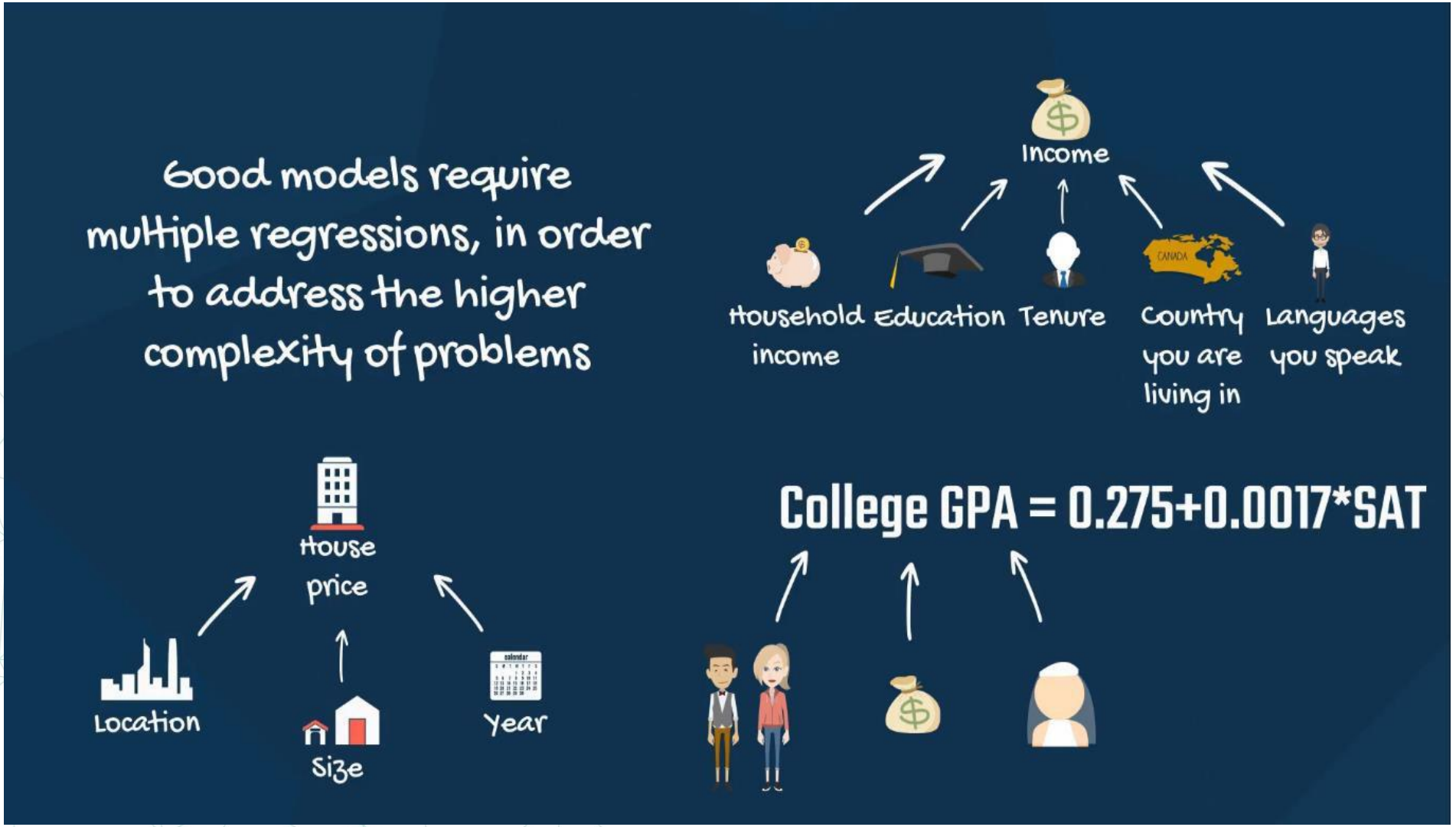
Error/Loss/Cost Function

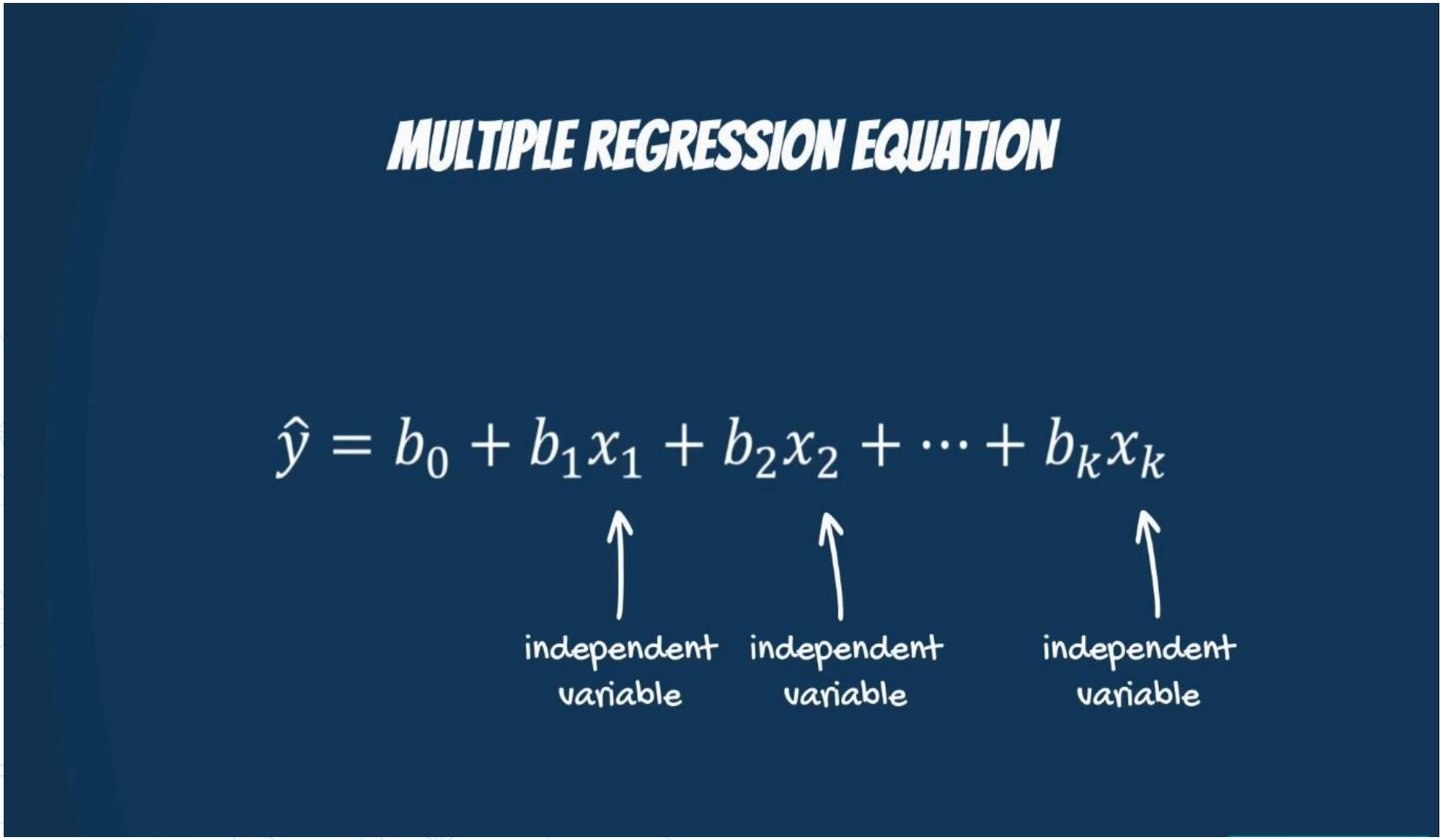
$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$



Multiple Linear Regression

- Unlike simple linear regression, multiple linear regression can include two or more independent variables.
- The goal is to estimate one variable based on several other variables.
- The variable to be estimated is called the **dependent variable (criterion).** The variables that are used for prediction are called **independent variables (predictors)**.
- Multiple linear regression is often used in empirical social research as well as in market research. In both areas it is of interest to find out what influence different factors have on a variable.


Good models require multiple regressions, in order to address the higher complexity of problems
Income
Household income
Education
Tenure
Country you are living in
Languages you speak
House price
Location
Size
Year
College GPA = 0.275+0.0017*SAT


MULTIPLE REGRESSION EQUATION
$\hat{y} = b_0 + b_1x_1 + b_2x_2 + \cdots + b_kx_k$
independent variable
independent variable
independent variable

Multiple Linear Regression

Simple linear Regression

$$\hat{y} = b \cdot x + a$$

Multiple linear Regression

$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$

- The coefficients are interpreted similarly to the linear regression equation.
- If all independent variables are 0, the value a is obtained.
- If an independent variable changes by one unit, the associated coefficient b indicates by how much the dependent variable changes.


PRICE
SIZE
LOCATION


? PRICE
SIZE
LOCATION

Metrics

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

less robust to outliers

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

less robust to outliers

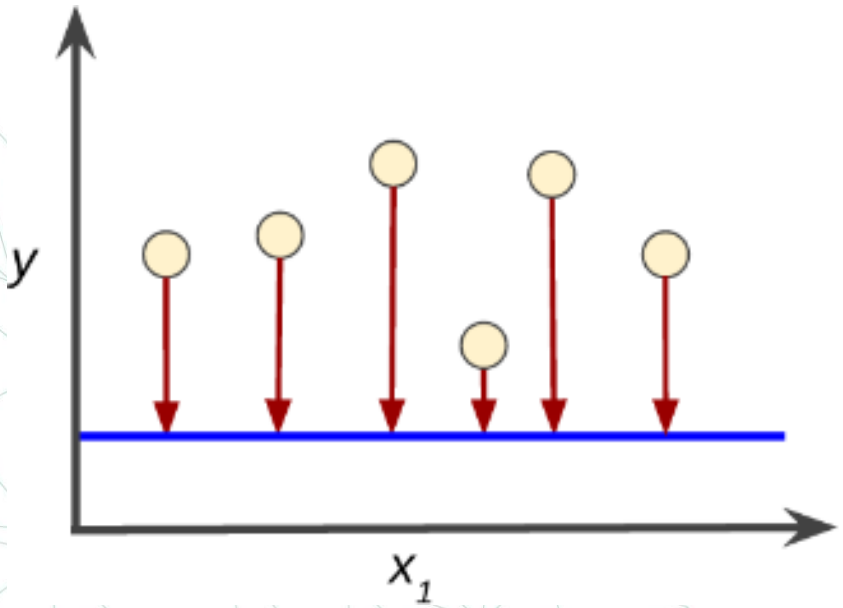
If you want to train a model which focuses on reducing large outlier errors then MSE is the better choice, whereas if this isn't important and you would prefer greater interpretability then MAE would be better.

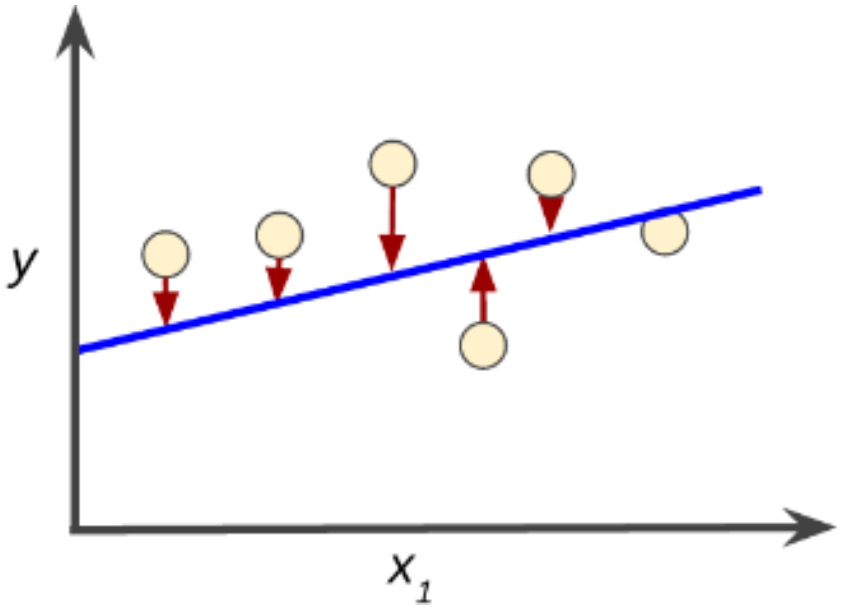
So a robust system or metric must be less affected by outliers. In this scenario it is easy to conclude that MSE may be less robust than MAE, since the squaring of the errors will enforce a higher importance on outliers.

Q&A

Questions and answers

Numerical Optimization


y
x_1


y
x_1

We have established the fact that all the straight lines are just different combination of model parameters a_0 and a_1

$$J(a) = \frac{1}{2m} \sum_{i=1}^{m} (a_0 + a_1 x_1^{(i)} - y_{i\,(act)})^2$$

and **Cost function (J)** is a function of parameter space $a = (a_0, a_1)$.

Therefore, by changing a_0 and a_1 we can change the **cost function.**

We will keep changing a_0 and a_1 till we find a combination where cost function is **minimized**

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$

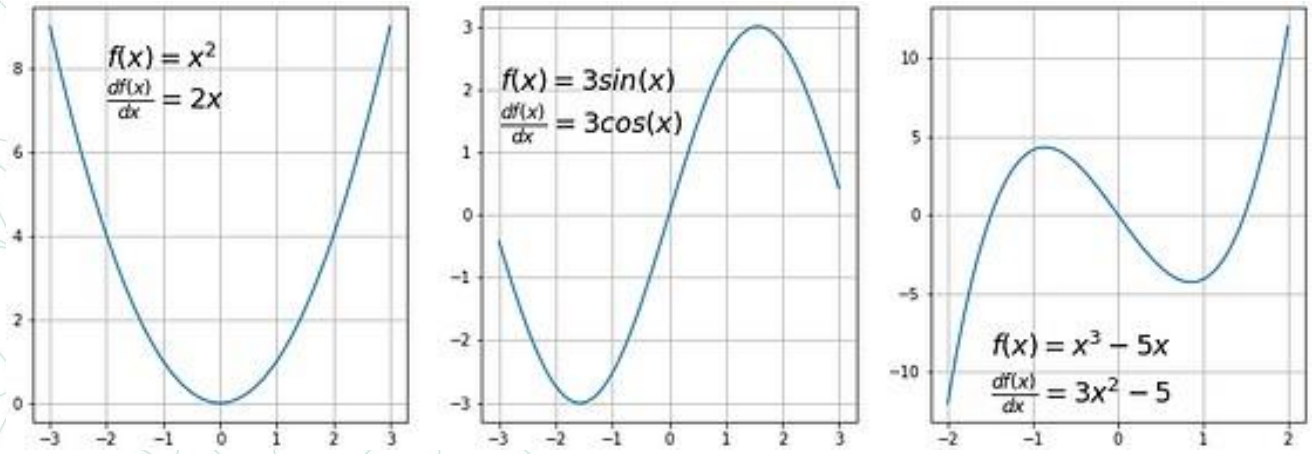
Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

The Gradient Descent Algorithm

- It's a powerful tool used to **optimize the parameters** of a model and minimize its error or cost function.
- In simple terms, gradient descent is a process of finding the minimum point of a function by following the steepest descent direction.
- It's used in a wide range of **machine learning algorithms**, Such as
  - **Linear regression,**
  - **Logistic regression,**
  - **Neural networks.**

Gradient

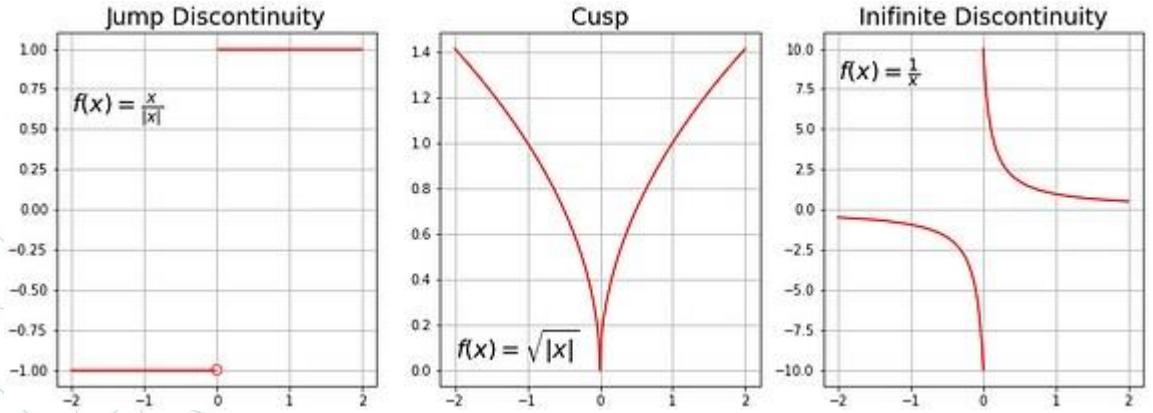
- Gradient descent algorithm does not work for all functions. There are two specific requirements. A function has to be:
  - **Differentiable (**has a derivative for each point in its domain**)**
  - **convex**



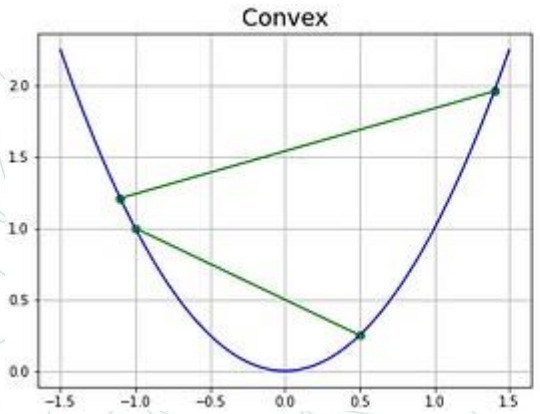
differentiable functions

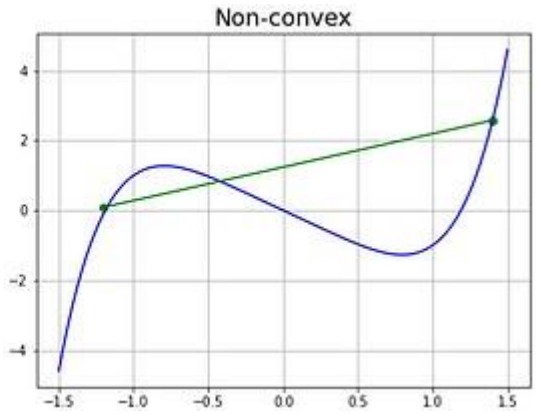
Gradient

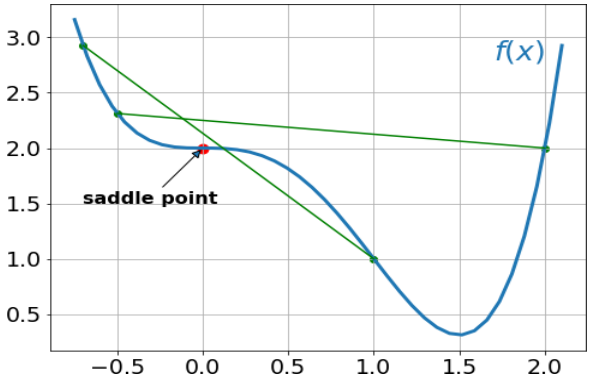
- Typical non-differentiable functions have a step a cusp or a discontinuity:



Gradient



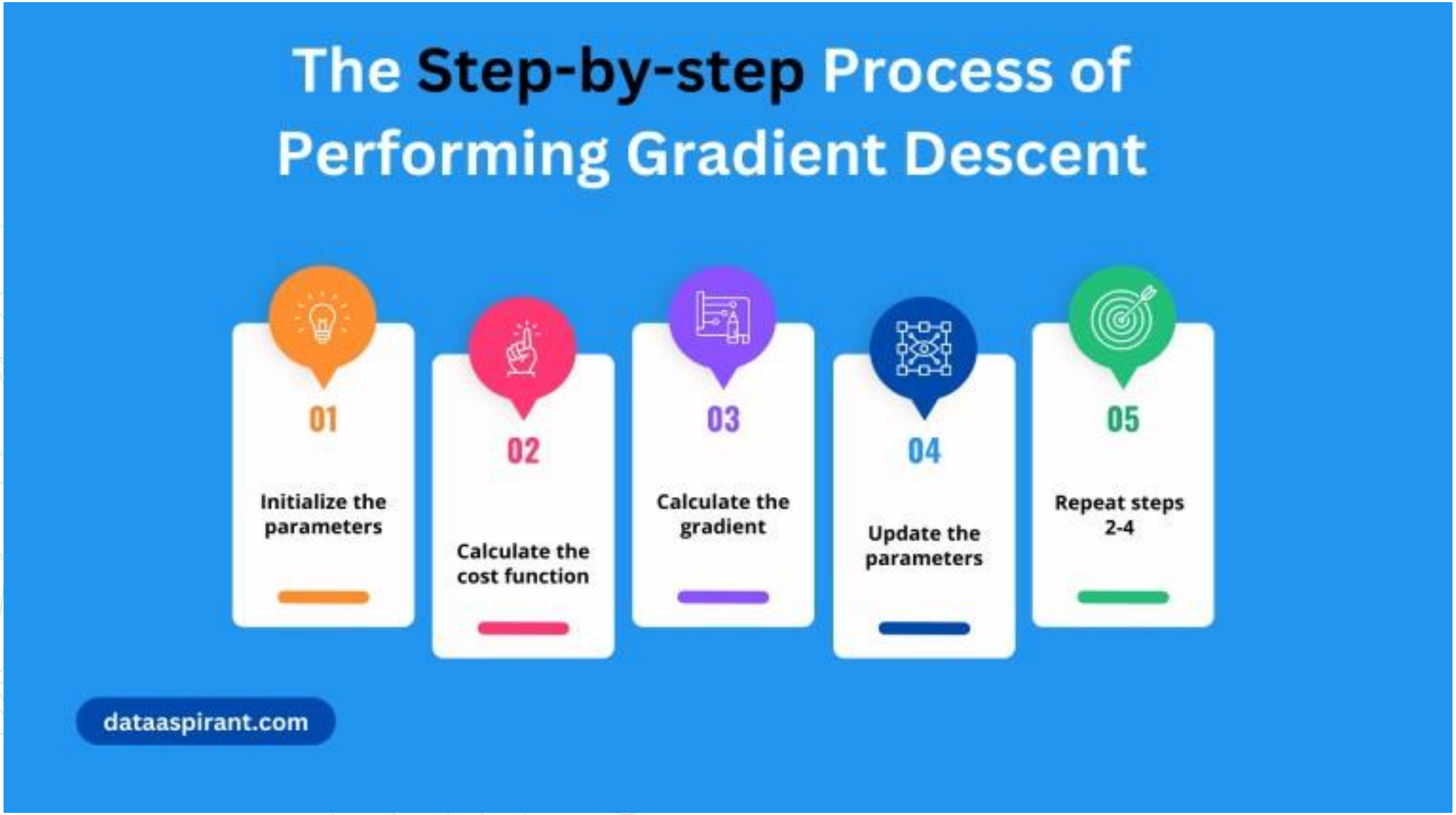




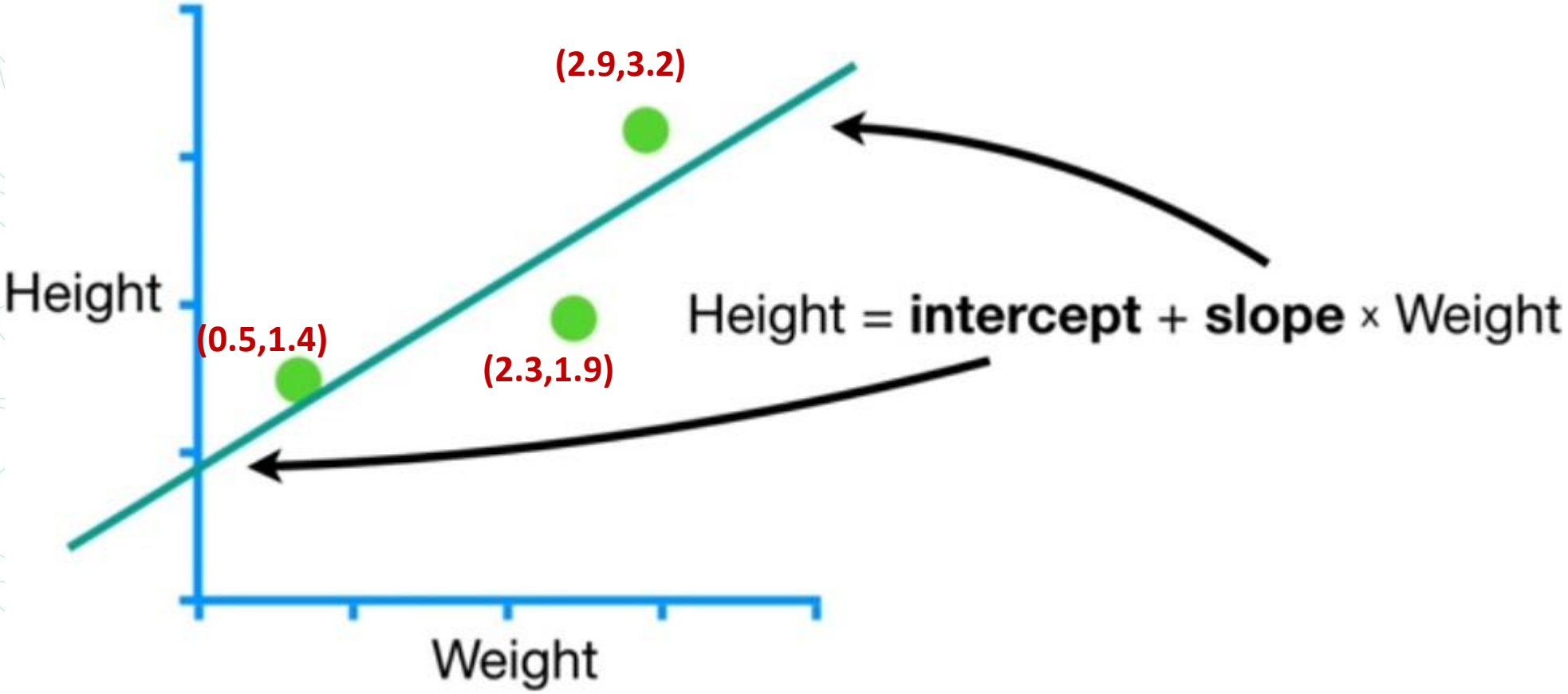
Gradient

- Intuitively it is a slope of a curve at a given point in a specified direction.
- In the case of **a univariate function**, it is simply the **first derivative at a selected point**.
- In the case of **a multivariate function**, it is a **vector of derivatives** in each main direction (along variable axes). Because we are interested only in a slope along one axis and we don't care about others these derivatives are called **partial derivatives**.
- A gradient for an n-dimensional function f(x) at a given point p is defined as follows:

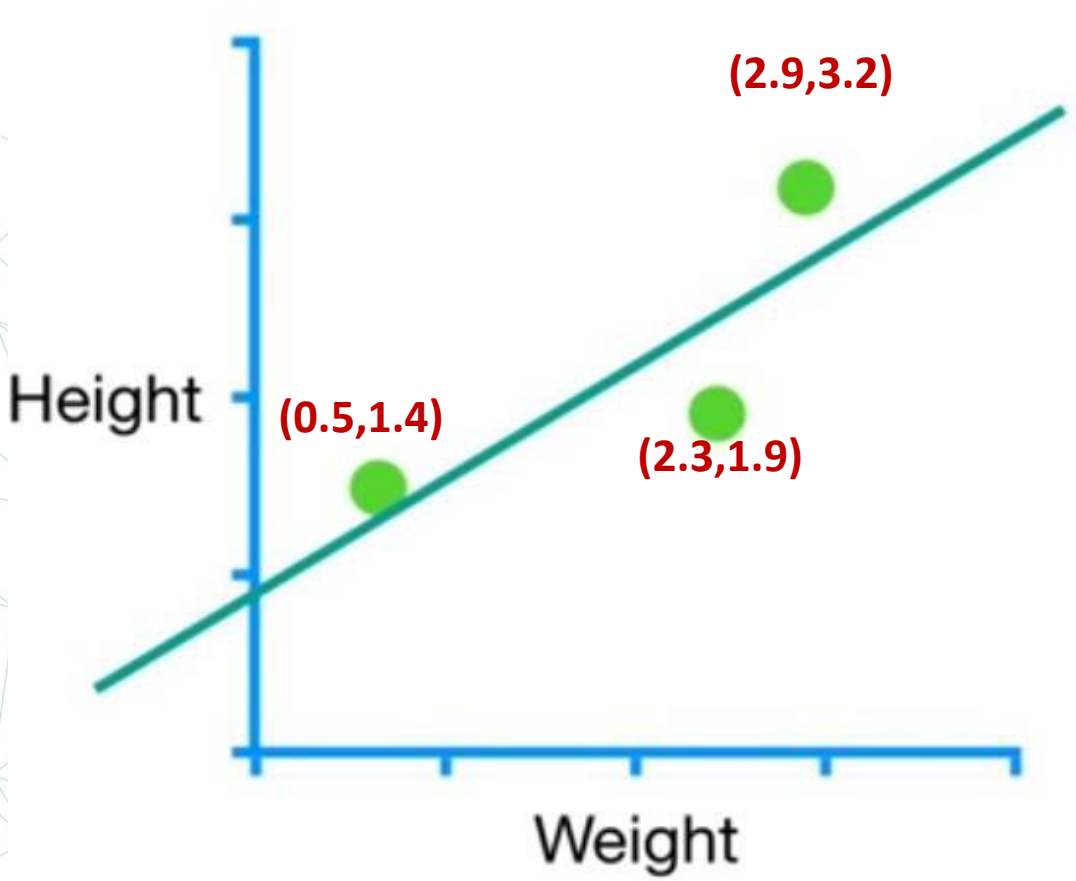
$$\nabla f(p) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(p) \\ \vdots \\ \frac{\partial f}{\partial x_n}(p) \end{bmatrix}$$


The Step-by-step Process of Performing Gradient Descent
01
Initialize the parameters
02
Calculate the cost function
03
Calculate the gradient
04
Update the parameters
05
Repeat steps 2-4
dataaspirant.com

When we fit a line with **Linear Regression**, we optimize the **Intercept** and **Slope**.

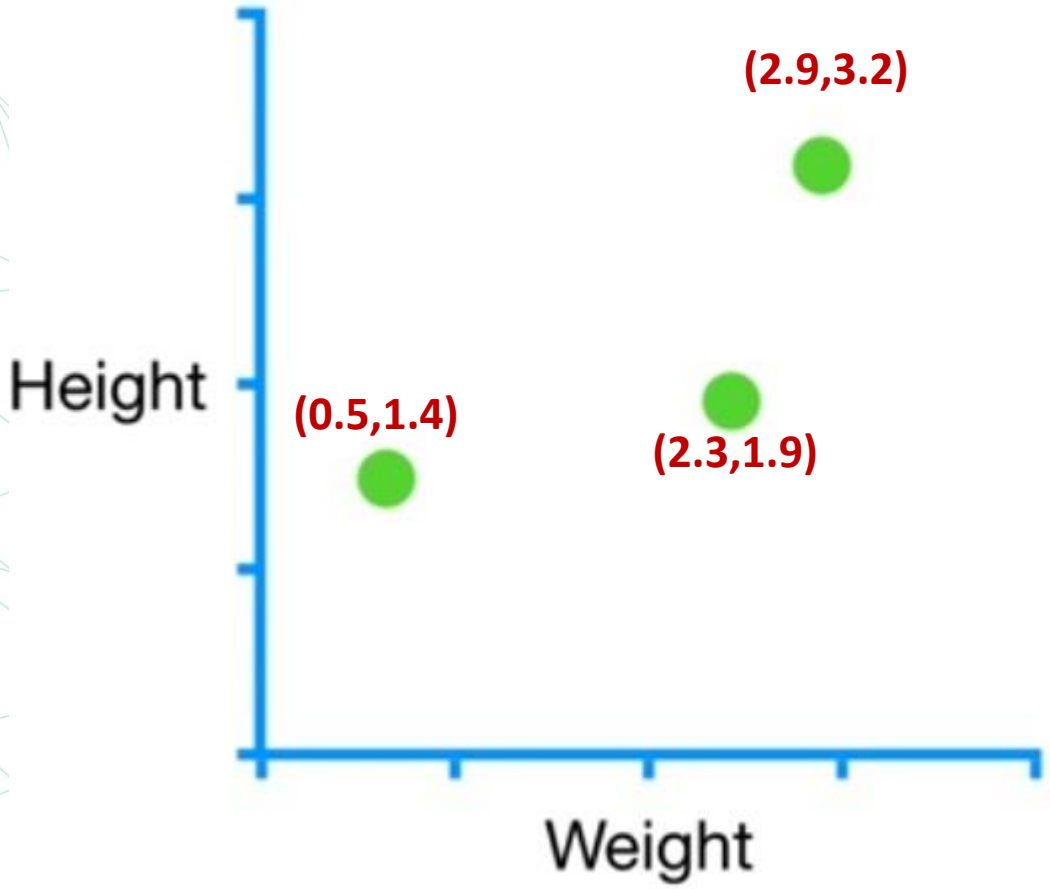


Predicted Height = intercept + slope × **Weight**



So for now, let's just plug in the **Least Squares** estimate for the **Slope**, **0.64**.

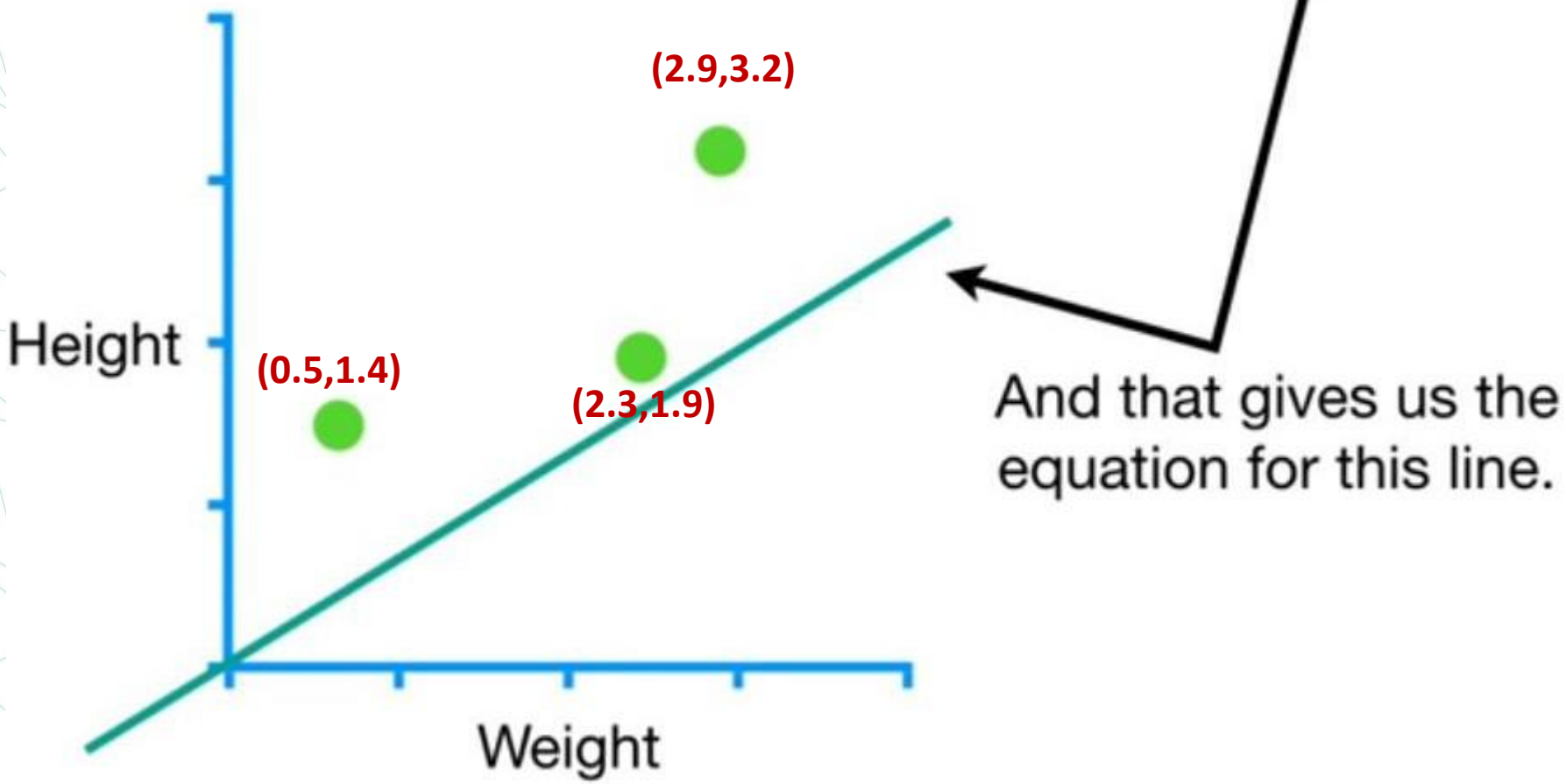
Predicted Height = intercept + 0.64 × **Weight**



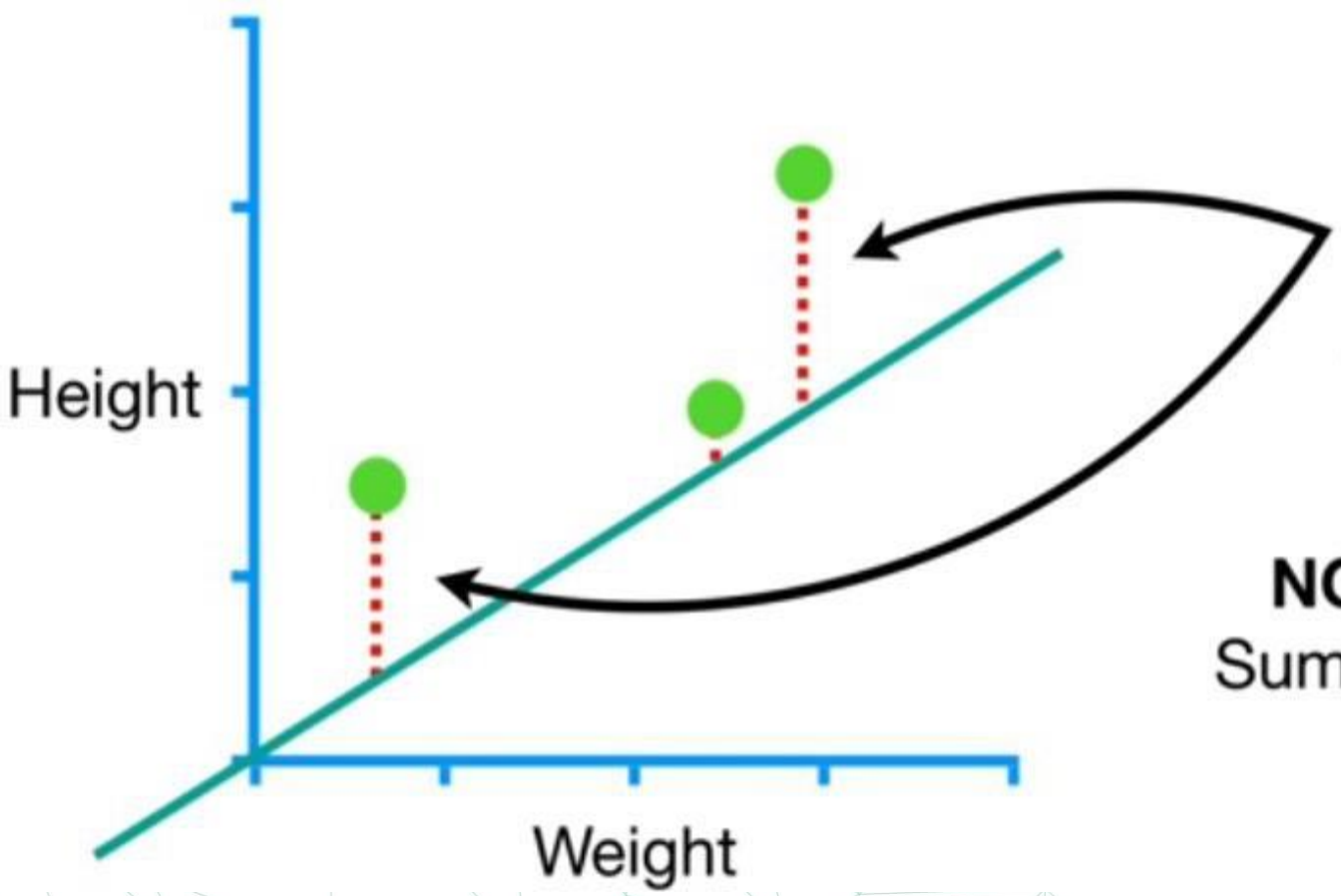
The first thing we do is pick a random value for the **Intercept**.

This is just an initial guess that gives **Gradient Descent** something to improve upon.

Predicted Height = 0 + 0.64 × **Weight**



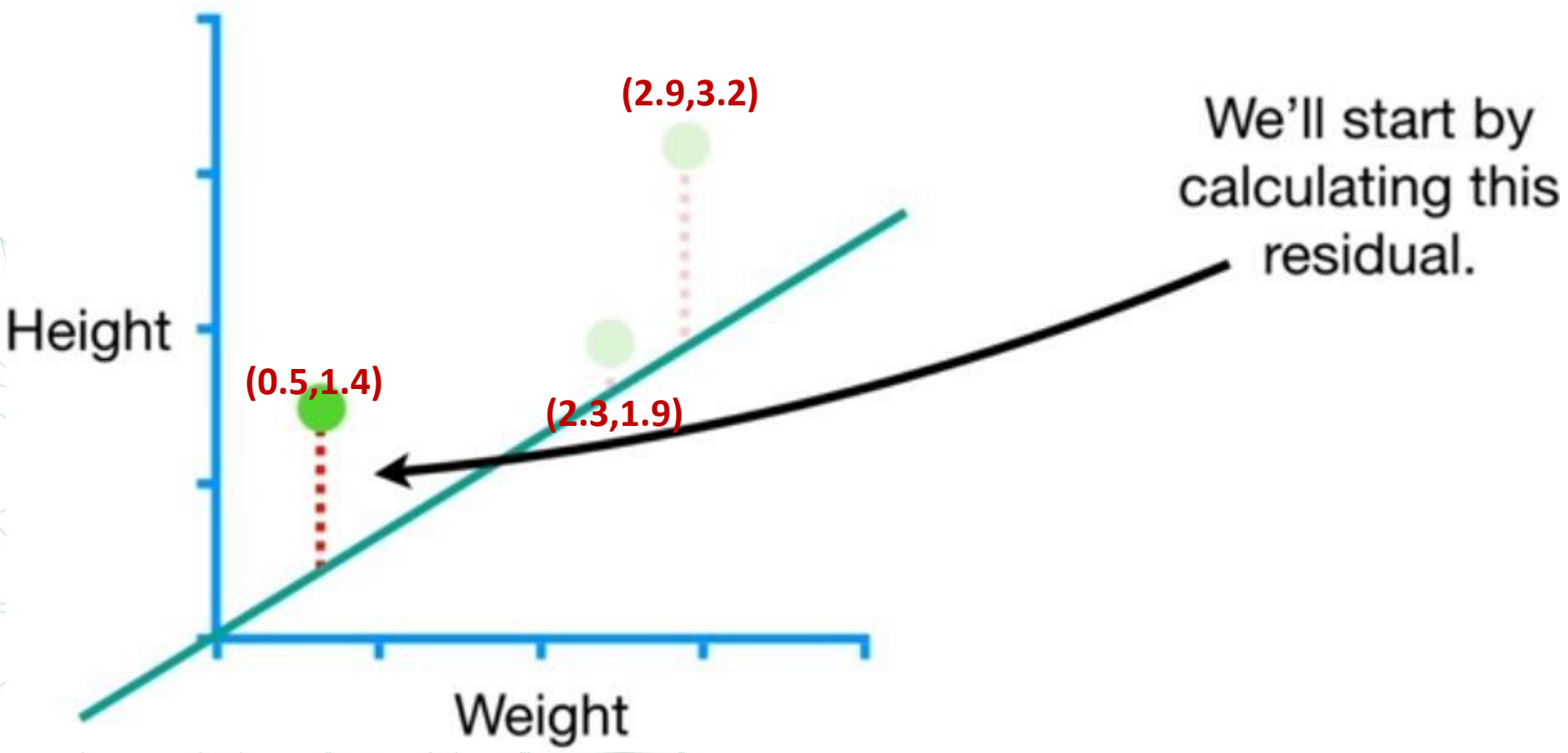
And that gives us the equation for this line.

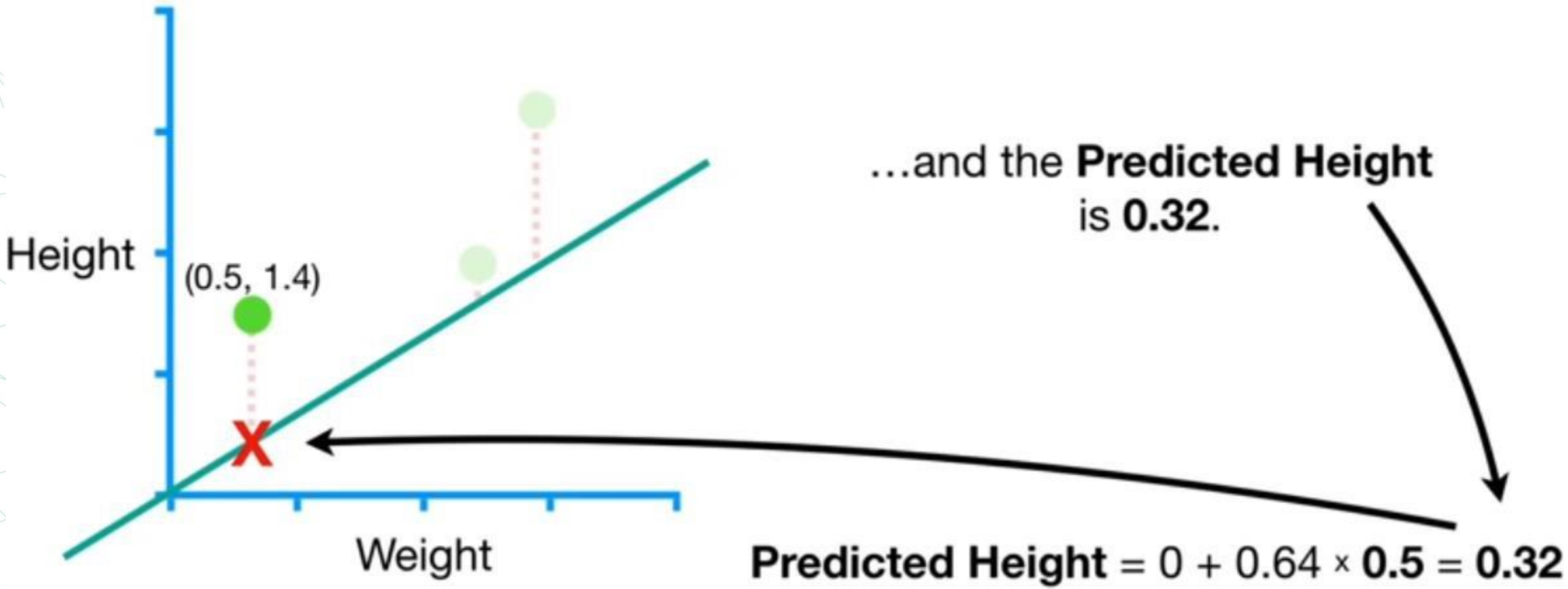


In this example, we will evaluate how well this line fits the data with the **Sum of the Squared Residuals**.

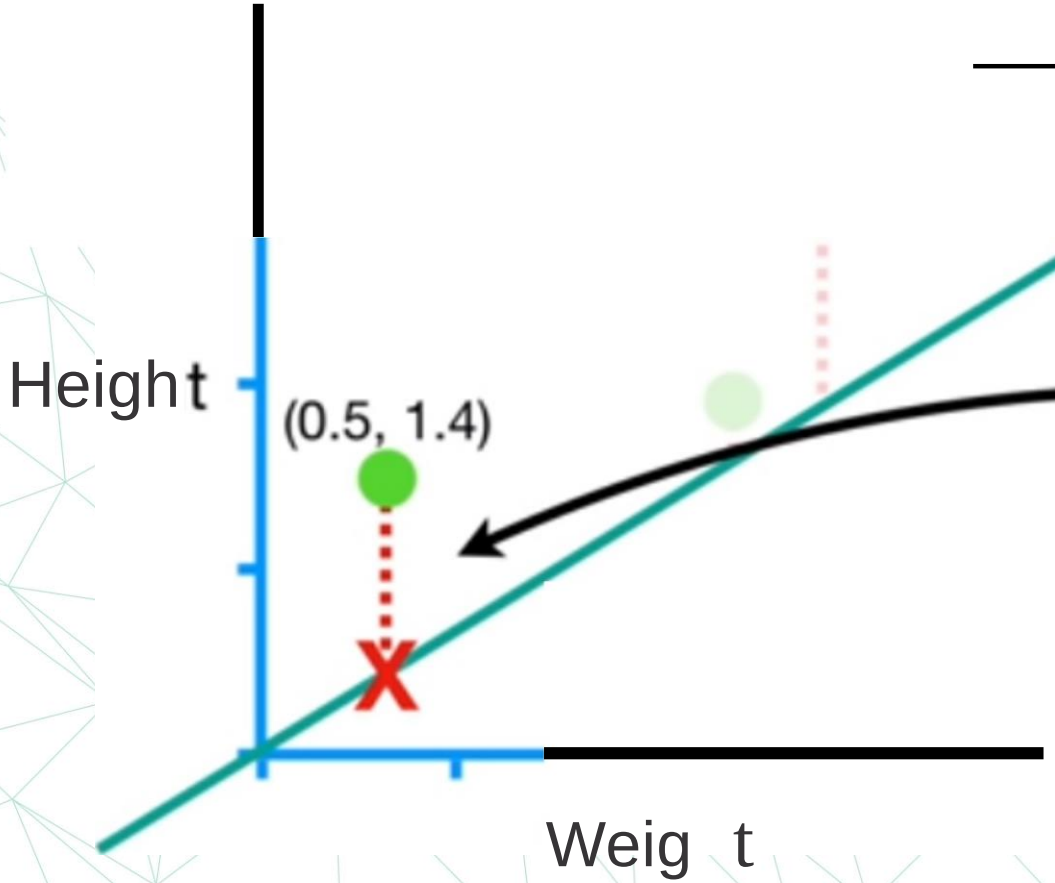
NOTE: In Machine Learning lingo, The Sum of the Squared Residuals is a type of **Loss Function**.

$$residual\ sum\ of\ squares = \sum_{i=1}^{n} (y_i - \hat{y_i})^2$$


(2.9,3.2)
We'll start by calculating this residual.
Height
(0.5,1.4)
(2.3,1.9)
Weight


Height
(0.5, 1.4)
X
Weight
...and the **Predicted Height** is **0.32**.
Predicted Height = 0 + 0.64 × **0.5** = **0.32**

$$residual\ sum\ of\ squares = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

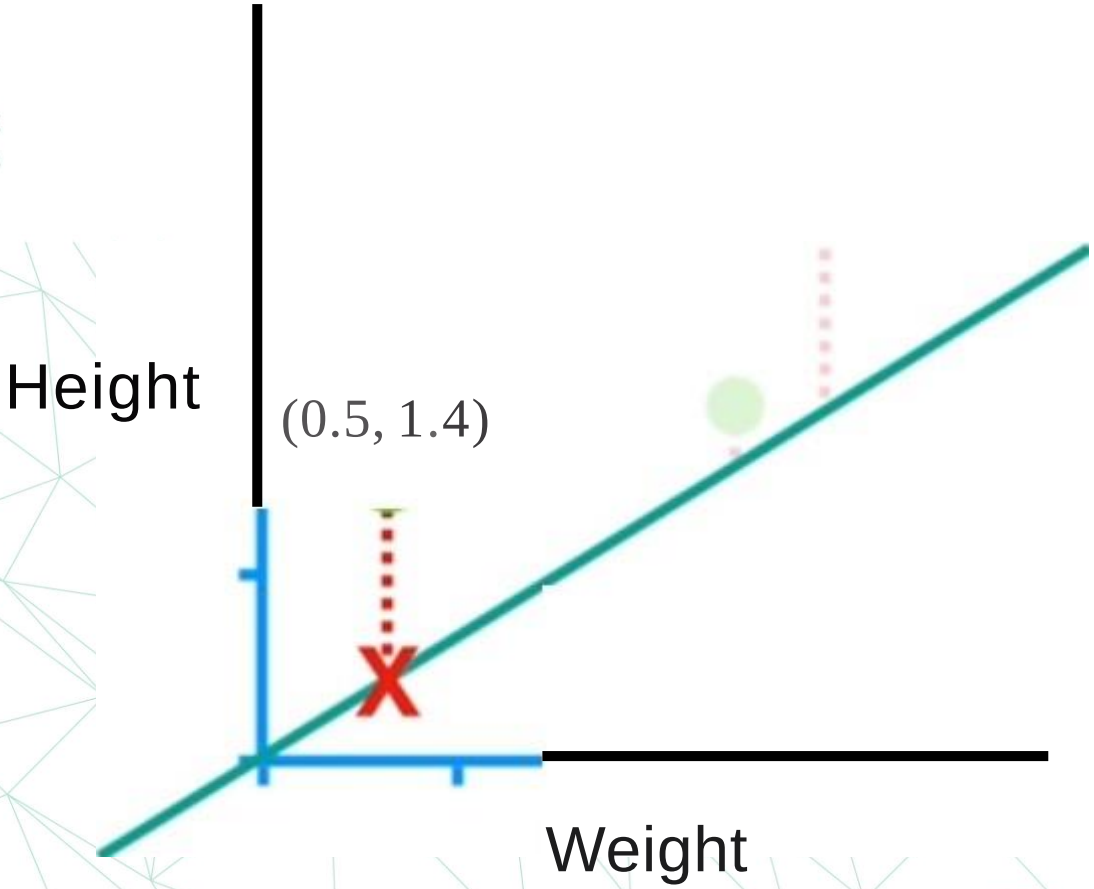


.. a d ha gives s 1.1
for h 1d al

es ual = 1 - 0.32 = **1.1**

Predicted Height 0.64 0.5 0.32

Sum of squared residuals = 1.1^2



Here's the square of the first residual...

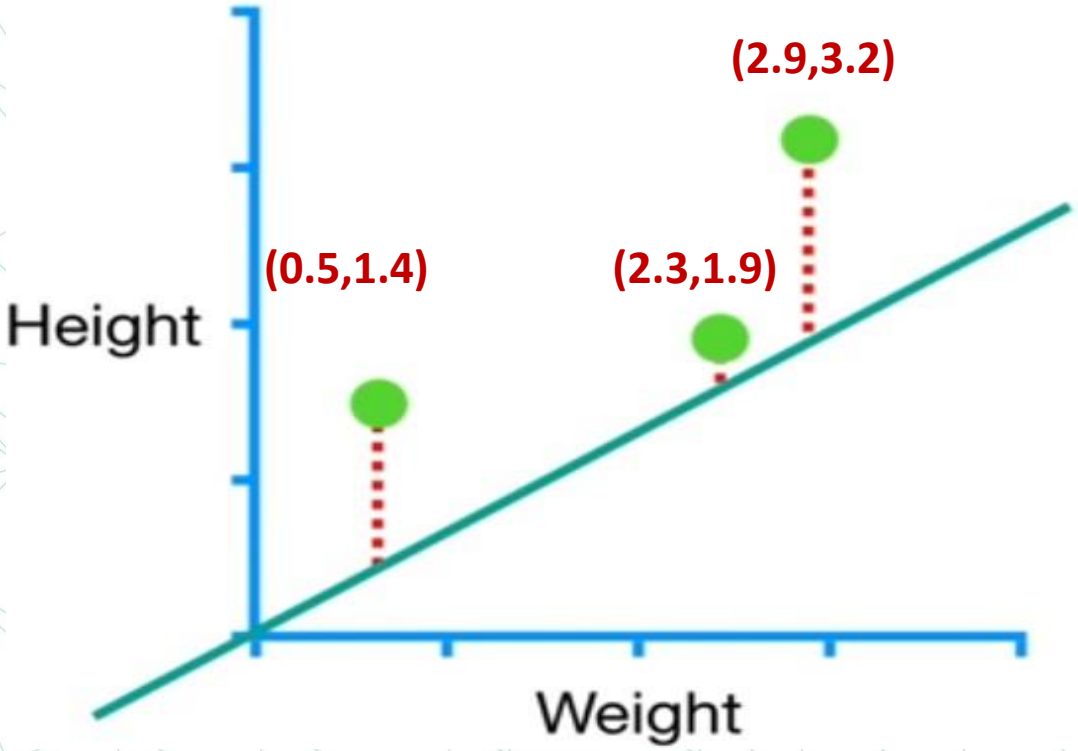
Residual = 1.4 - 0.32 = **1.1**

Predicted Height O O6 0.5 0.32

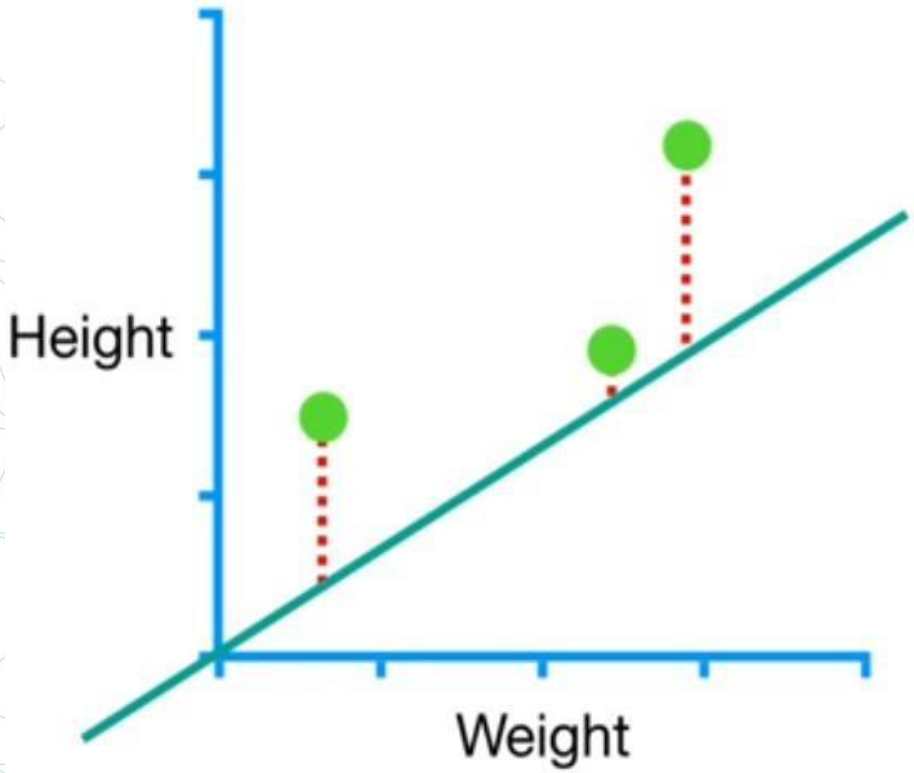
$$\sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

Sum of squared residuals = $1.1^2 + 0.4^2 + 1.3^2 =$ 3.1

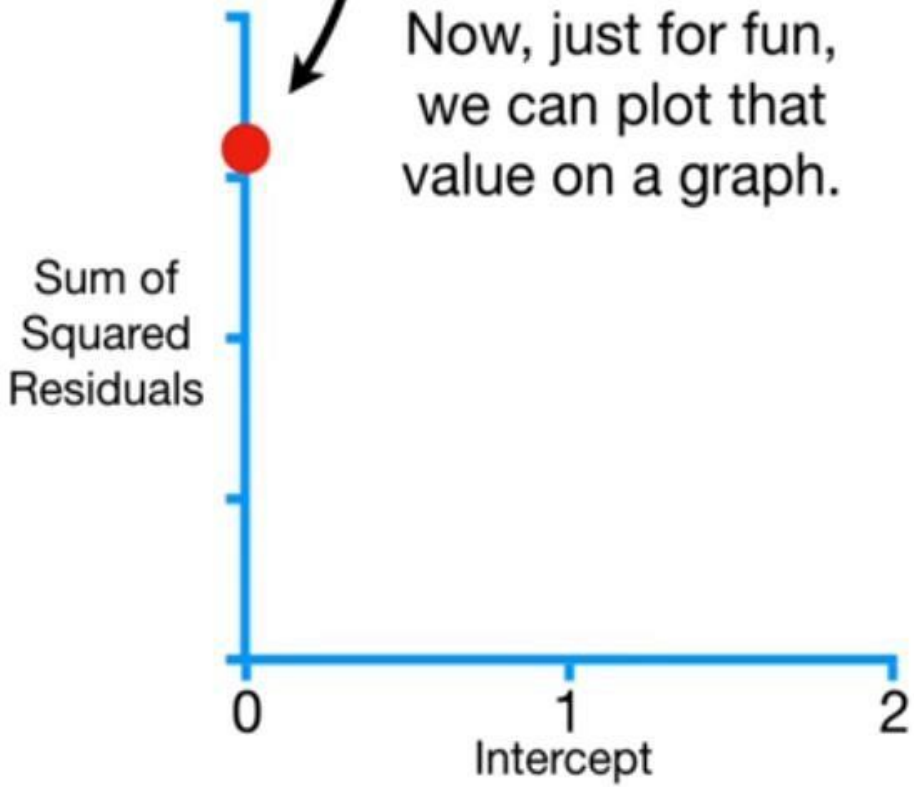
In the end, **3.1** is the Sum of the Squared Residuals.



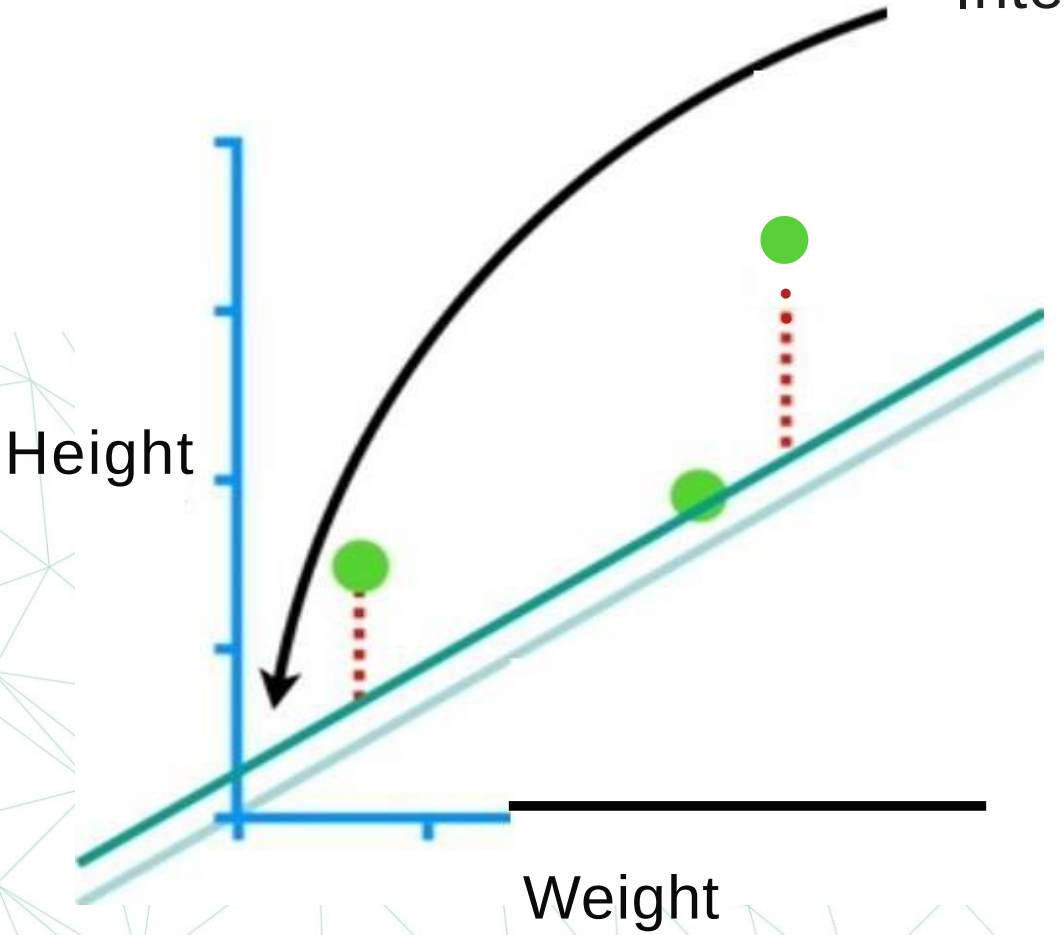
Sum of squared residuals = $1.1^2 + 0.4^2 + 1.3^2 = 3.1$



Now, just for fun, we can plot that value on a graph.

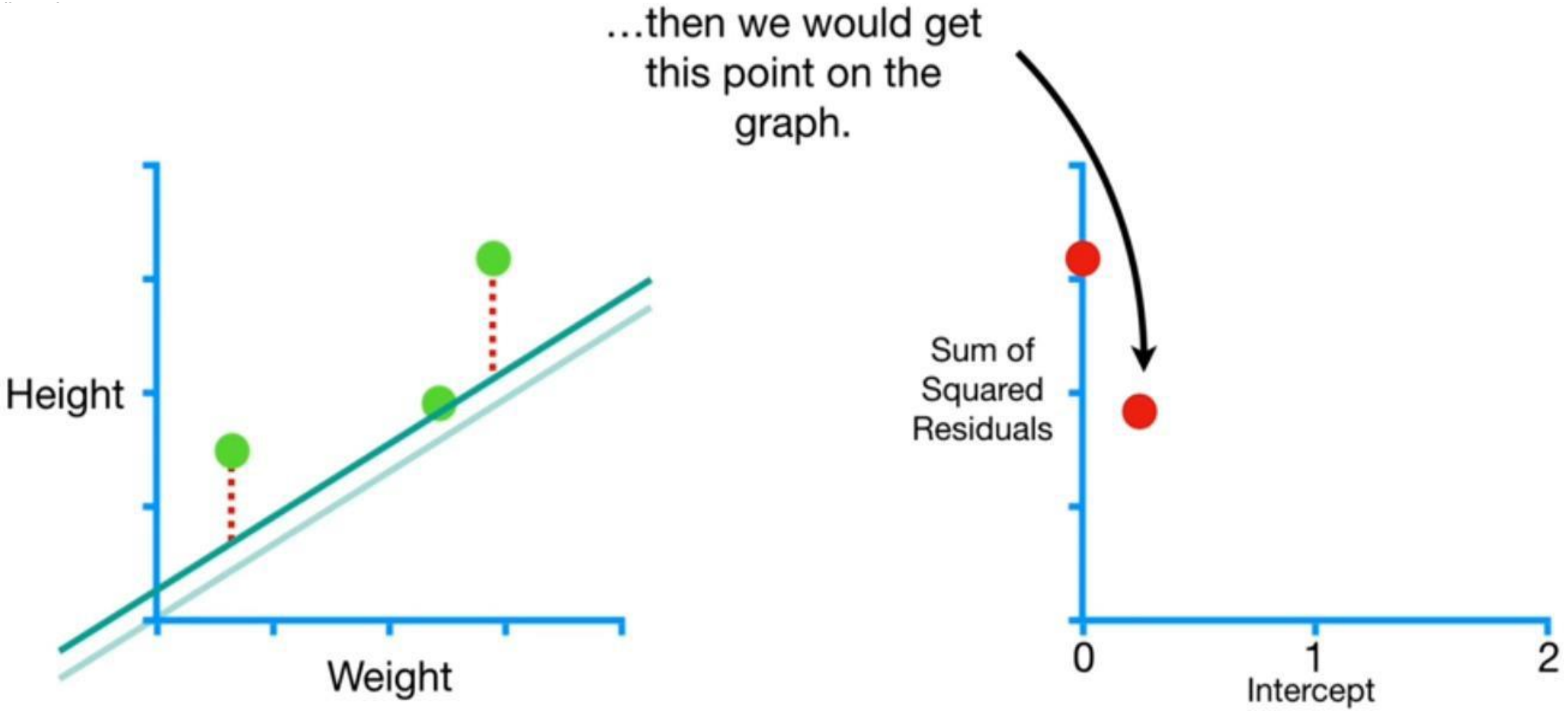


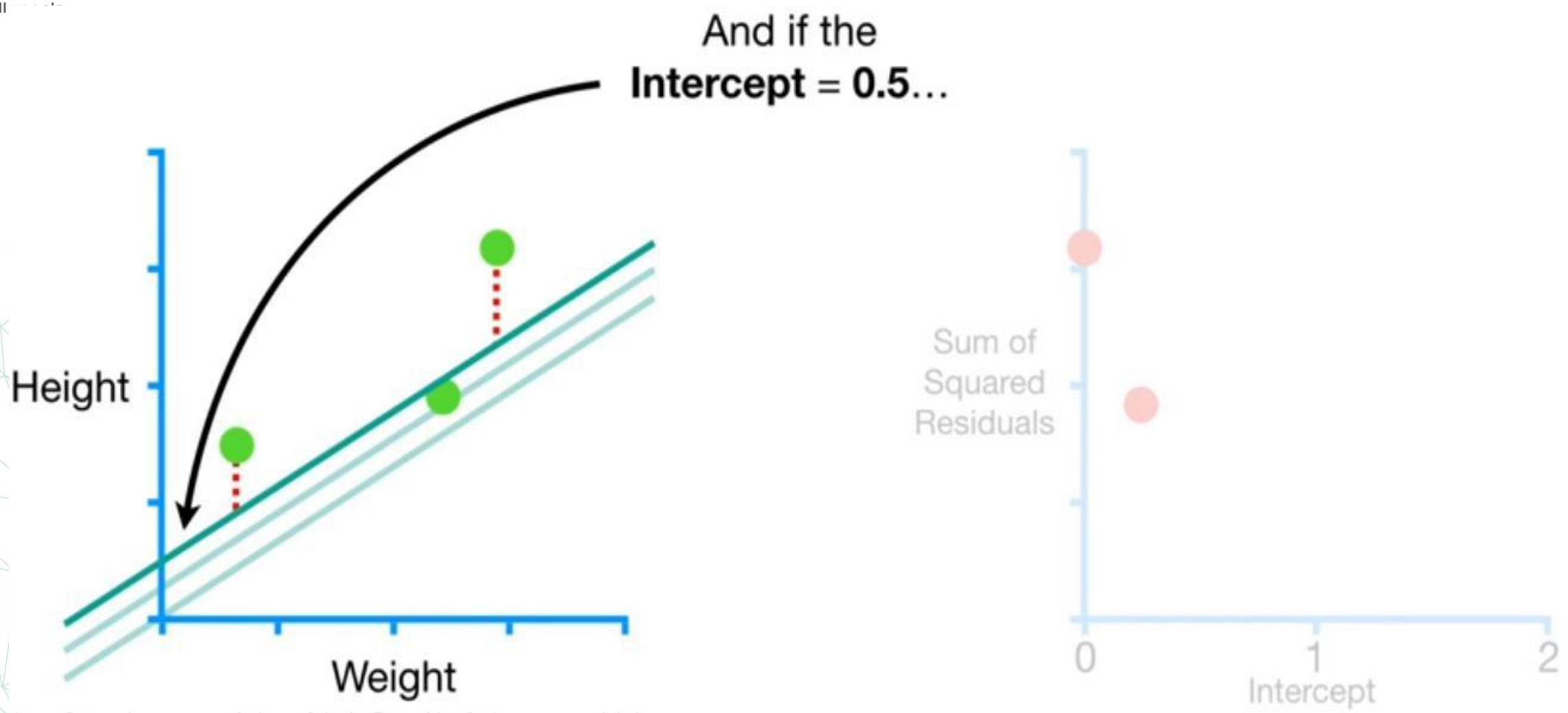
However, if the Intercept = 0.25…

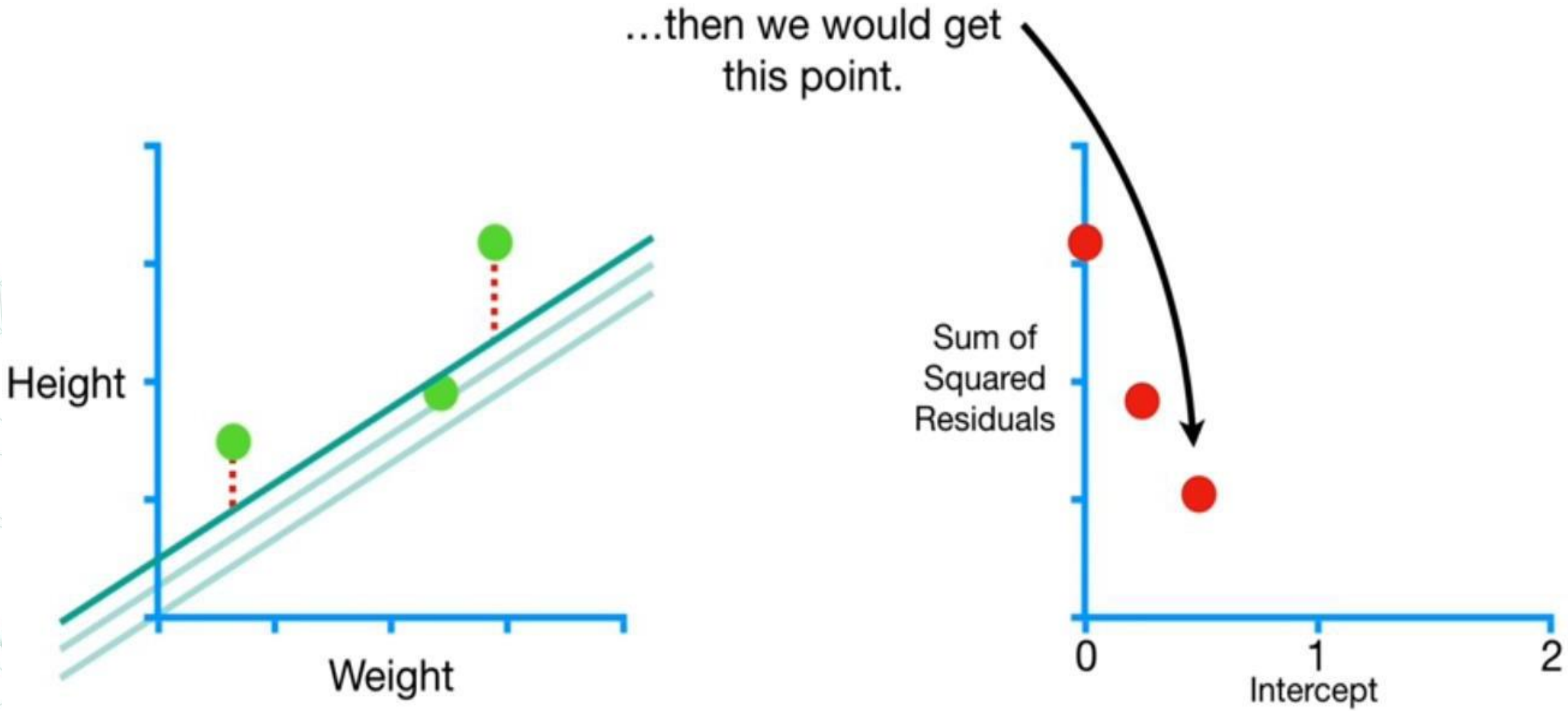


Sum of Squared Residuals

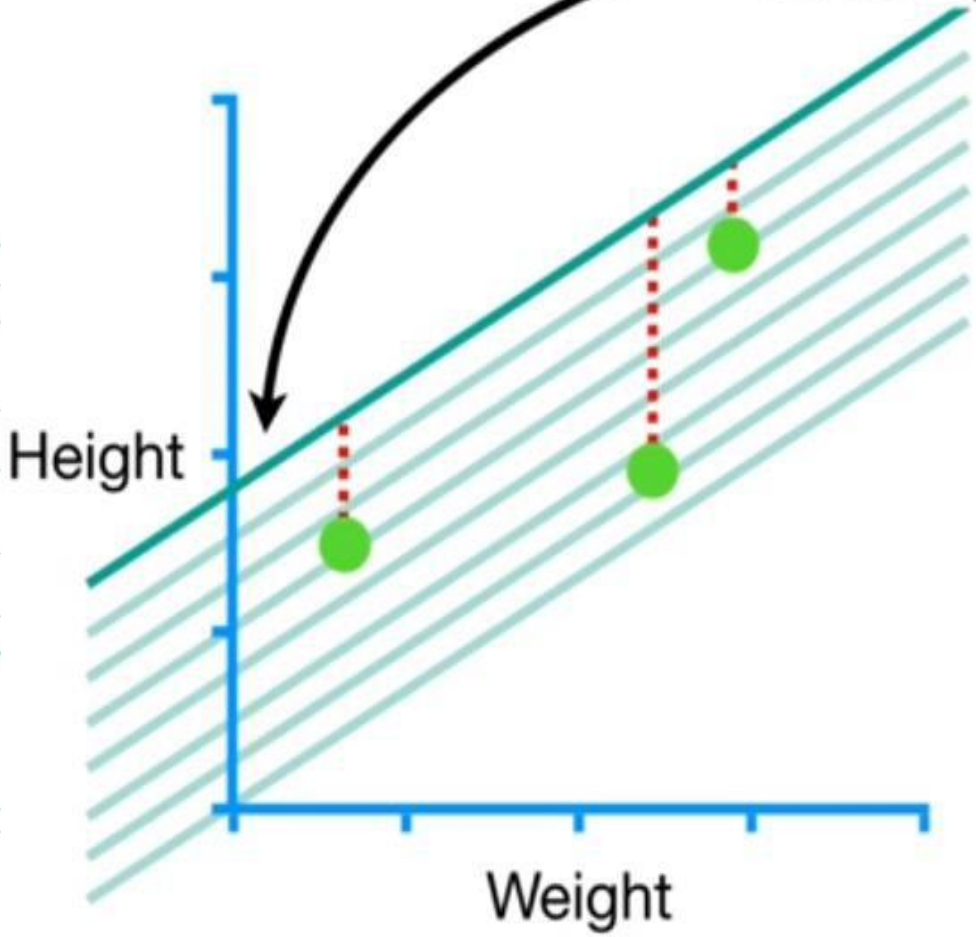
0 1 2

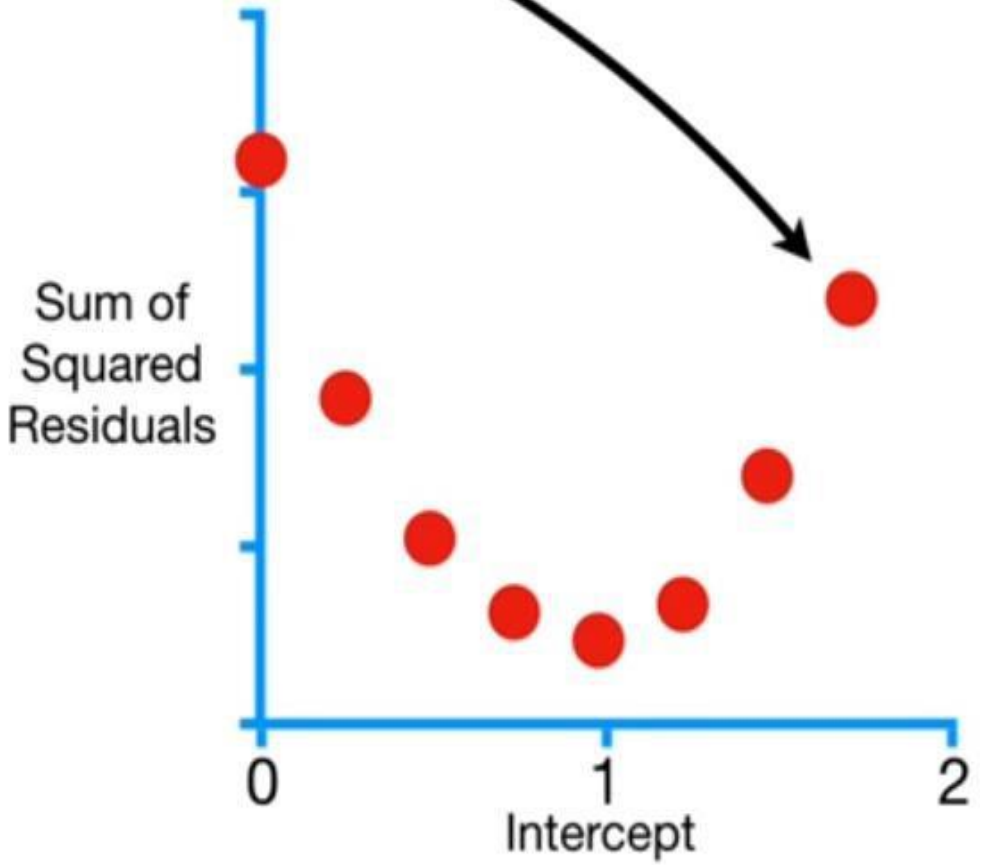

...then we would get this point on the graph.
Height
Weight
Sum of Squared Residuals
0
1
2
Intercept

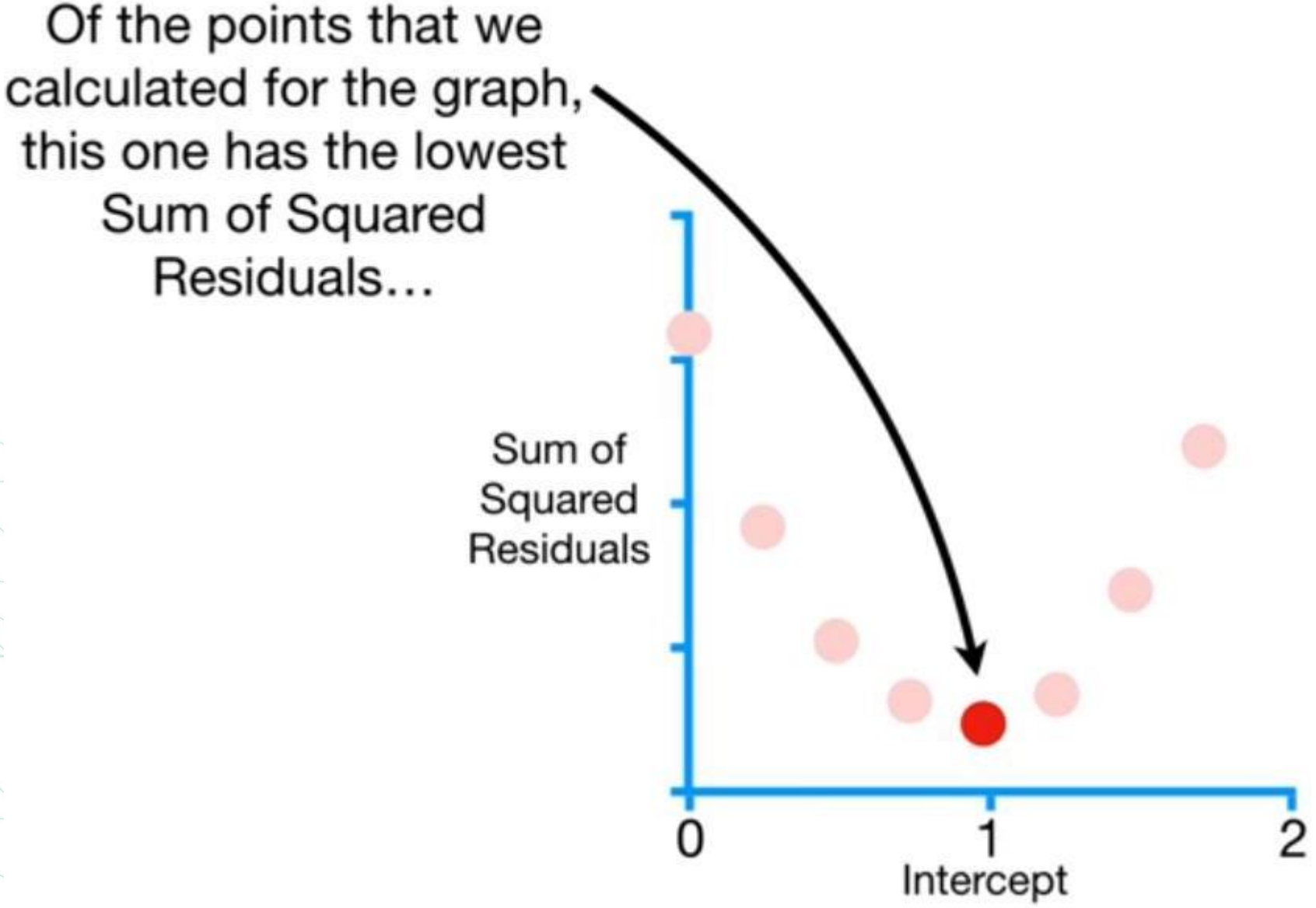

And if the
Intercept = 0.5...
Height
Weight
Sum of
Squared
Residuals
0
1
2
Intercept


...then we would get
this point.
Height
Weight
Sum of
Squared
Residuals
0
1
2
Intercept

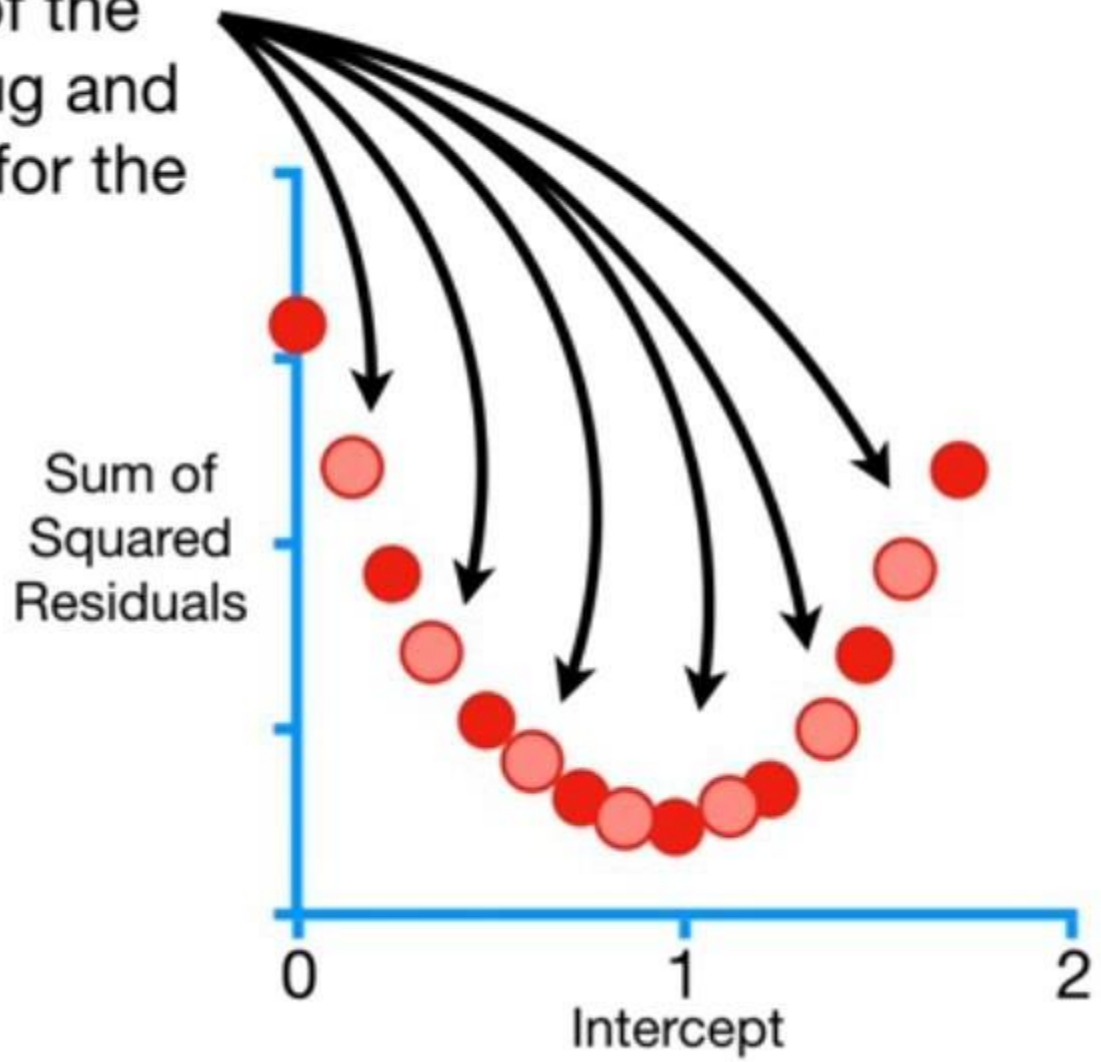
And for increasing values for the **Intercept**, we get these points.



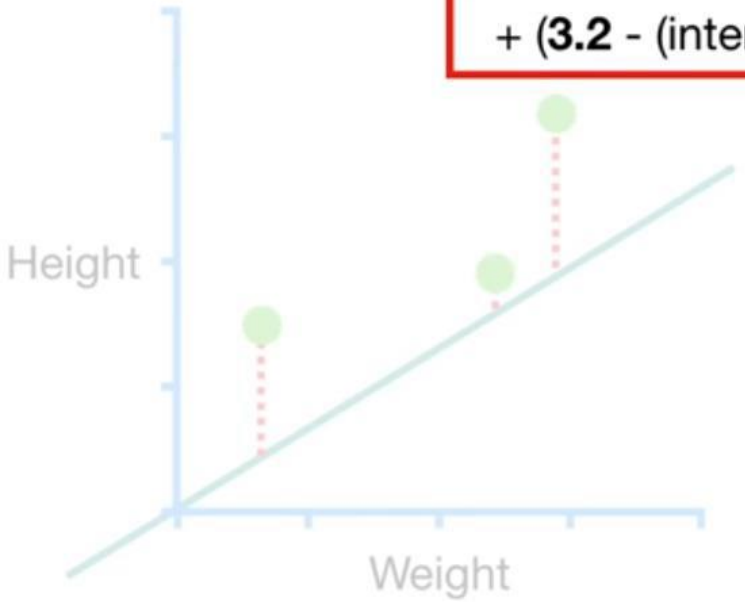



Of the points that we calculated for the graph, this one has the lowest Sum of Squared Residuals...
Sum of Squared Residuals
0
1
2
Intercept

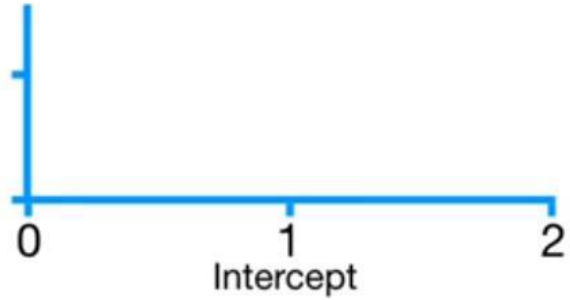
A slow and painful method for finding the minimal Sum of the Squared Residuals is to plug and chug a bunch more values for the **Intercept**.



Sum of squared residuals = (**1.4** - (intercept + 0.64 × **0.5**))^2
+ (**1.9** - (intercept + 0.64 × **2.3**))^2
+ (**3.2** - (intercept + 0.64 × **2.9**))^2

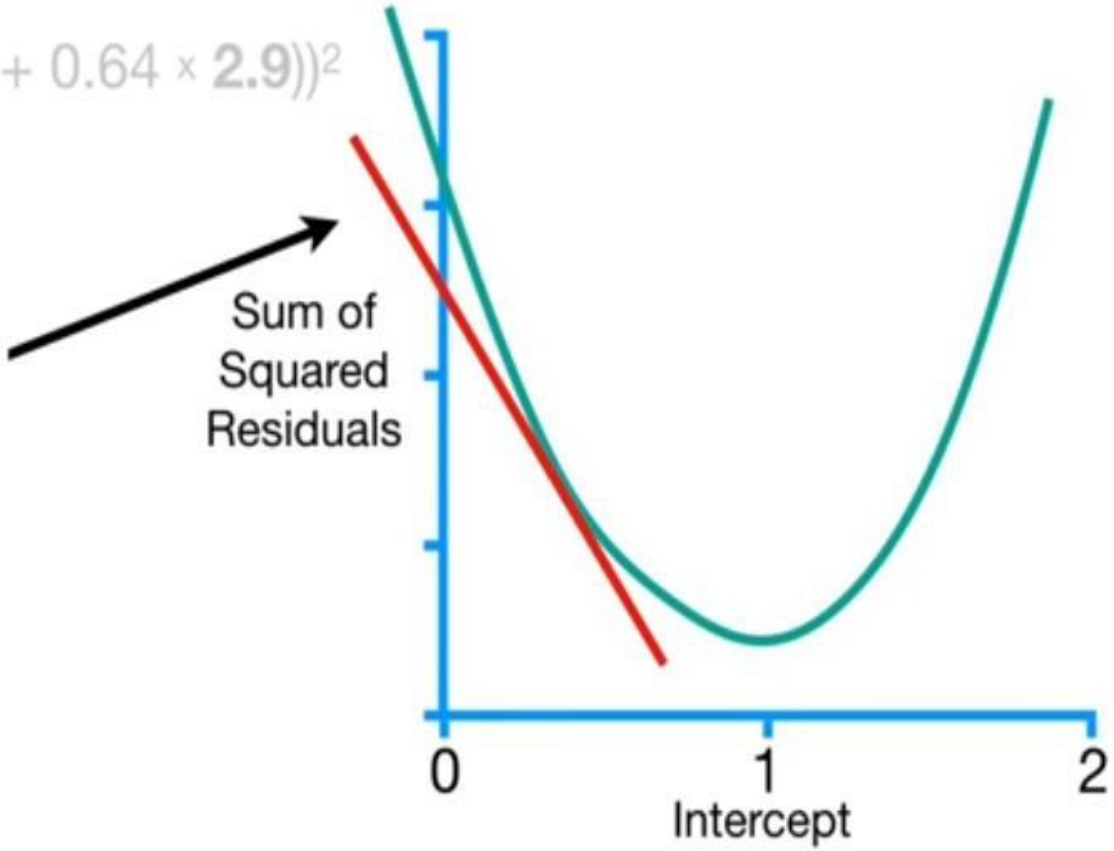


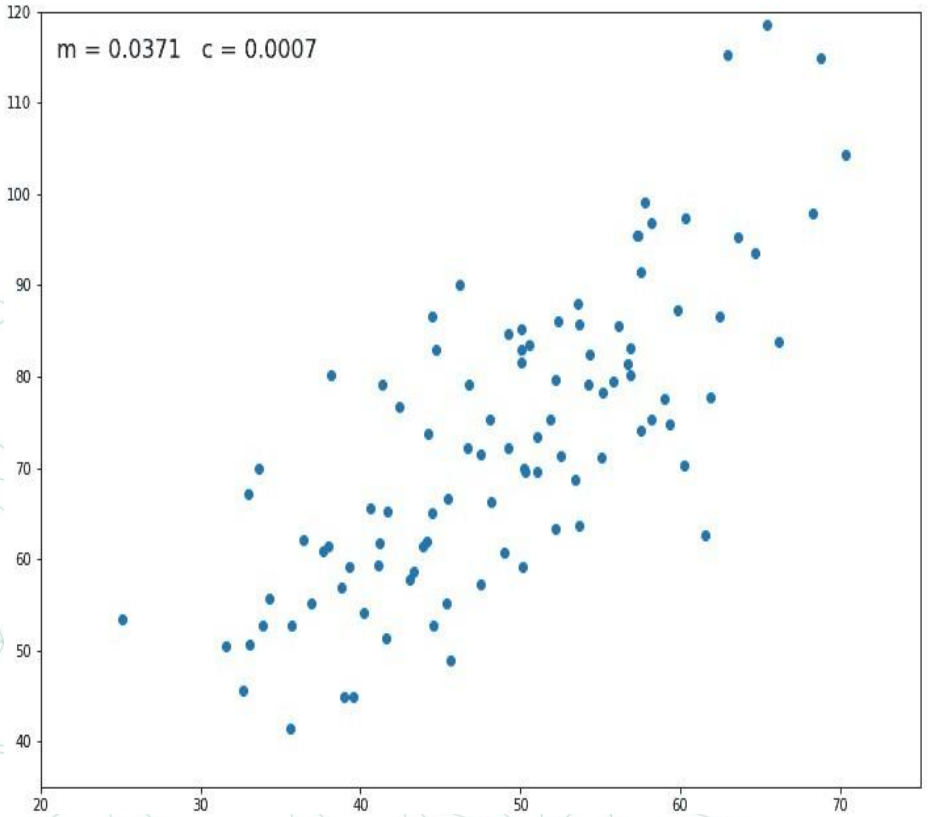
Now we can easily plug in any value for the **intercept**…

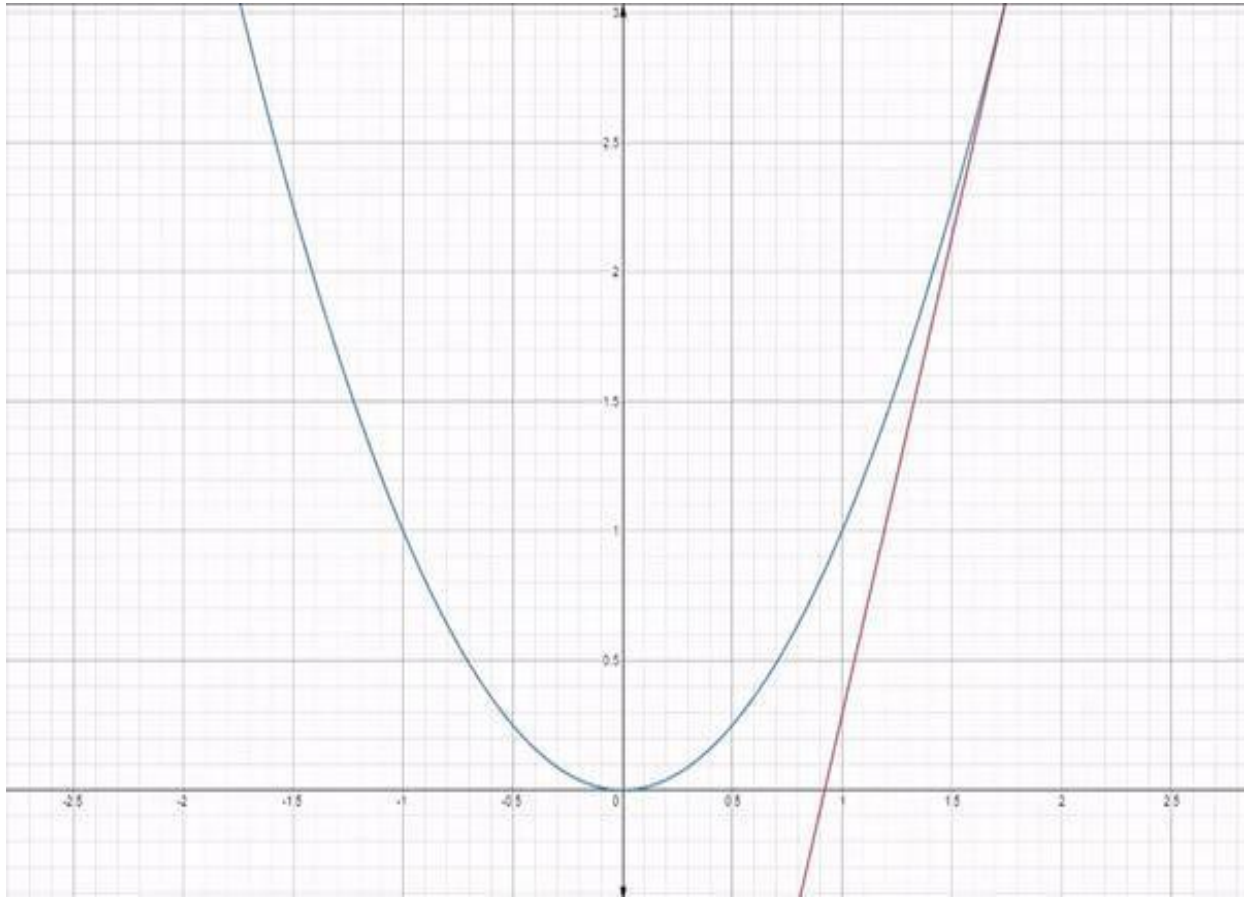


$$\text{Sum of squared residuals} = (\mathbf{1.4} - (\text{intercept} + 0.64 \times \mathbf{0.5}))^2 + (\mathbf{1.9} - (\text{intercept} + 0.64 \times \mathbf{2.3}))^2 + (\mathbf{3.2} - (\text{intercept} + 0.64 \times \mathbf{2.9}))^2$$

…and we can take the derivative of this function and determine the slope at any value for the **Intercept**.




m = 0.0371 c = 0.0007
120
110
100
90
80
70
60
50
40
20
30
40
50
60
70



$$\text{Sum of squared residuals} = (\mathbf{1.4} - (\text{intercept} + 0.64 \times \mathbf{0.5}))^2 + (\mathbf{1.9} - (\text{intercept} + 0.64 \times \mathbf{2.3}))^2 + (\mathbf{3.2} - (\text{intercept} + 0.64 \times \mathbf{2.9}))^2$$

Let's start by taking the derivative of the first part.

$$\frac{d}{d\,\text{intercept}} \text{Sum of squared residuals} = \frac{d}{d\,\text{intercept}} (\mathbf{1.4} - (\text{intercept} + 0.64 \times \mathbf{0.5}))^2 + \frac{d}{d\,\text{intercept}} (\mathbf{1.9} - (\text{intercept} + 0.64 \times \mathbf{2.3}))^2 + \frac{d}{d\,\text{intercept}} (\mathbf{3.2} - (\text{intercept} + 0.64 \times \mathbf{2.9}))^2$$

$$\frac{d}{d\,intercept}\left(\mathbf{1.4} - (intercept + 0.64 \times \mathbf{0.5})\right)^2 = 2(\mathbf{1.4} - (intercept + 0.64 \times \mathbf{0.5})) \times -1$$

$$\frac{d}{d\,intercept}\ \mathbf{1.4} - (intercept + 0.64 \times \mathbf{0.5})$$

$$\frac{d}{d\,intercept}\ \cancel{\mathbf{1.4}} + (-1)intercept - \cancel{0.64 \times \mathbf{0.5}} = -1$$

These parts don't contain a term for the **Intercept**, so they go away.

$$\frac{d}{d\,intercept}(\mathbf{1.4} - (intercept + 0.64 \times \mathbf{0.5}))^2 = 2(\mathbf{1.4} - (intercept + 0.64 \times \mathbf{0.5})) \times -1$$

$$= -2(\mathbf{1.4} - (intercept + 0.64 \times \mathbf{0.5}))$$

…and this…

…is the derivative of the first part…

$$\frac{d}{d\,intercept}\text{Sum of squared residuals} = \frac{d}{d\,intercept}(\mathbf{1.4} - (intercept + 0.64 \times \mathbf{0.5}))^2$$

$$+ \frac{d}{d\,intercept}(\mathbf{1.9} - (intercept + 0.64 \times \mathbf{2.3}))^2$$

$$+ \frac{d}{d\,intercept}(\mathbf{3.2} - (intercept + 0.64 \times \mathbf{2.9}))^2$$

I'll leave that as an exercise for the viewer.

$$\frac{d}{d\,intercept}\text{Sum of squared residuals} = -2(\mathbf{1.4} - (intercept + 0.64 \times \mathbf{0.5})$$

$$+ -2(\mathbf{1.9} - (intercept + 0.64 \times \mathbf{2.3}))$$

$$+ -2(\mathbf{3.2} - (intercept + 0.64 \times \mathbf{2.9}))$$

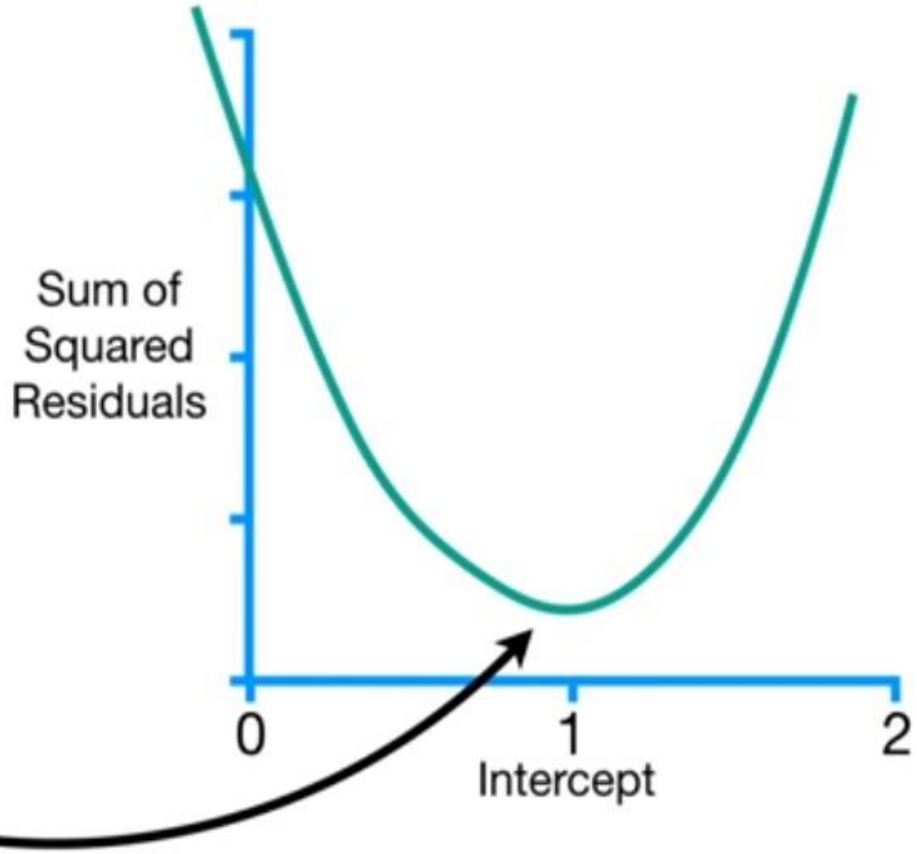
$$\frac{d}{d\ intercept}\ \text{Sum of squared residuals} =$$

$$-2(\mathbf{1.4} - (\text{intercept} + 0.64 \times \mathbf{0.5}))$$

$$+ -2(\mathbf{1.9} - (\text{intercept} + 0.64 \times \mathbf{2.3}))$$

$$+ -2(\mathbf{3.2} - (\text{intercept} + 0.64 \times \mathbf{2.9}))$$

Now that we have the derivative, **Gradient Descent** will use it to find where the Sum of Squared Residuals is lowest.



$$\frac{d}{d\ intercept}\ \text{Sum of squared residuals} =$$

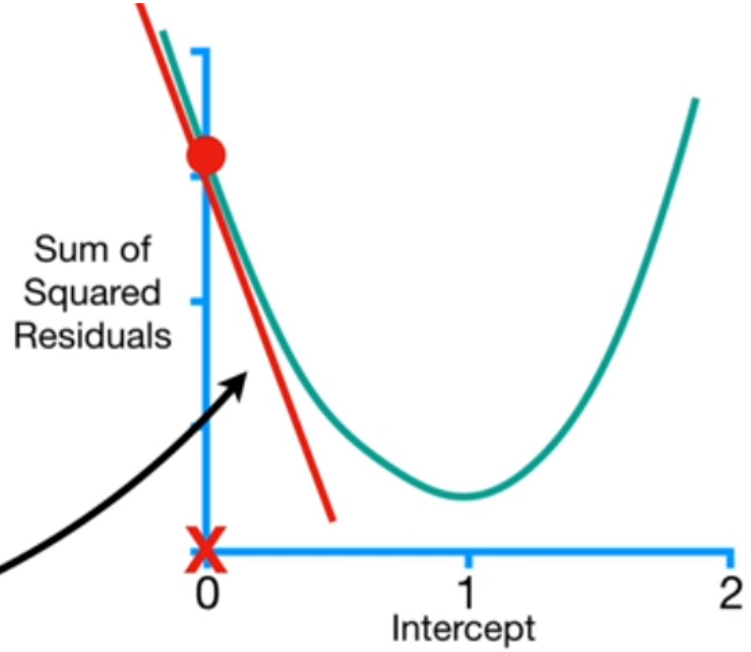
$$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5})$$

$$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$$

$$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$$

$$= -5.7$$

So when the **Intercept** = **0**, the slope of the curve = **-5.7**.



$$\frac{d}{d\,intercept}\text{Sum of squared residuals} =$$

$$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5})$$

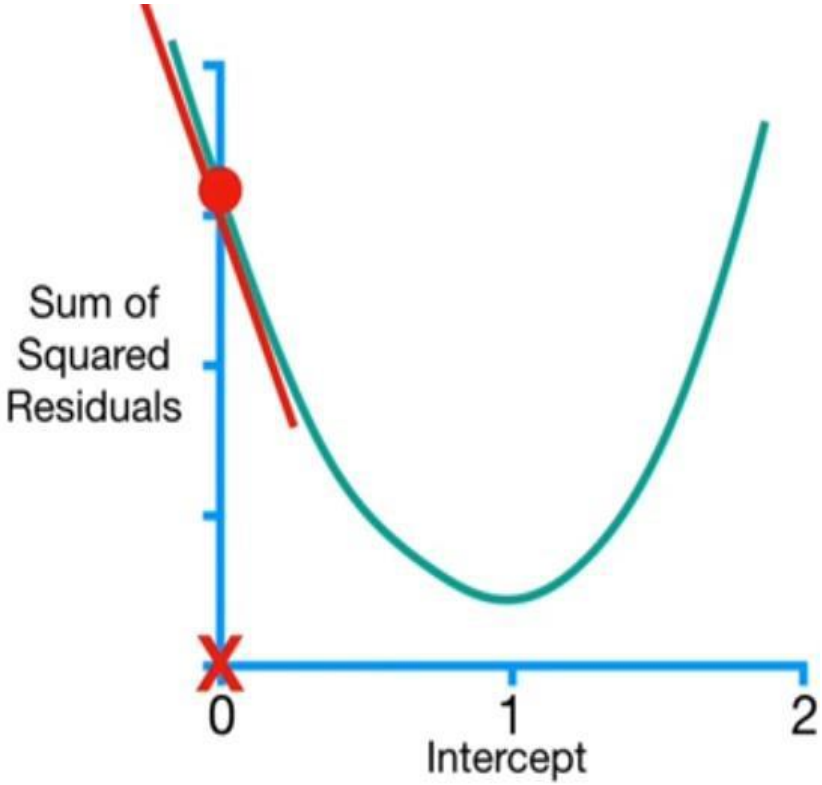
$$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$$

$$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$$

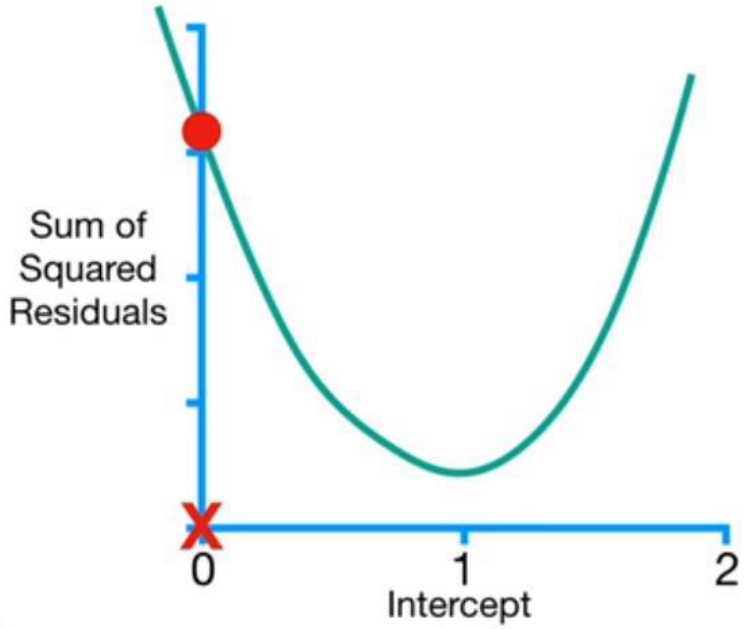
$$= -5.7$$

Step Size = -5.7 x 0.1

…by a small number called
The Learning Rate.



$$\frac{d}{d\,intercept}\text{Sum of squared residuals} =$$
$$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5}))$$
$$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$$
$$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$$
$$= -5.7$$



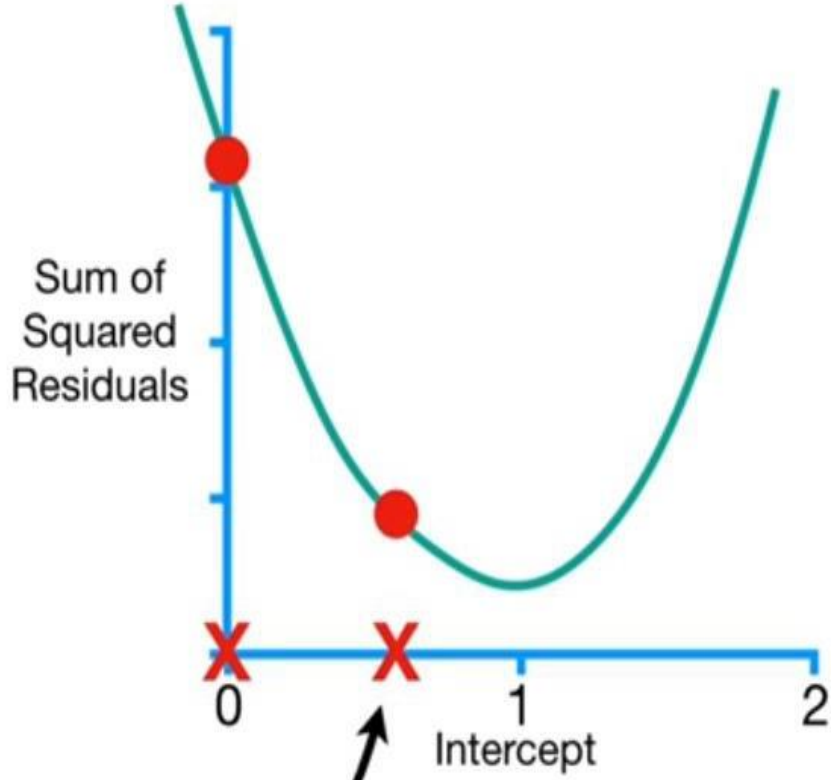
Step Size = -5.7 × 0.1 = -0.57

New Intercept = **Old Intercept** - **Step Size**

…minus the **Step Size**.

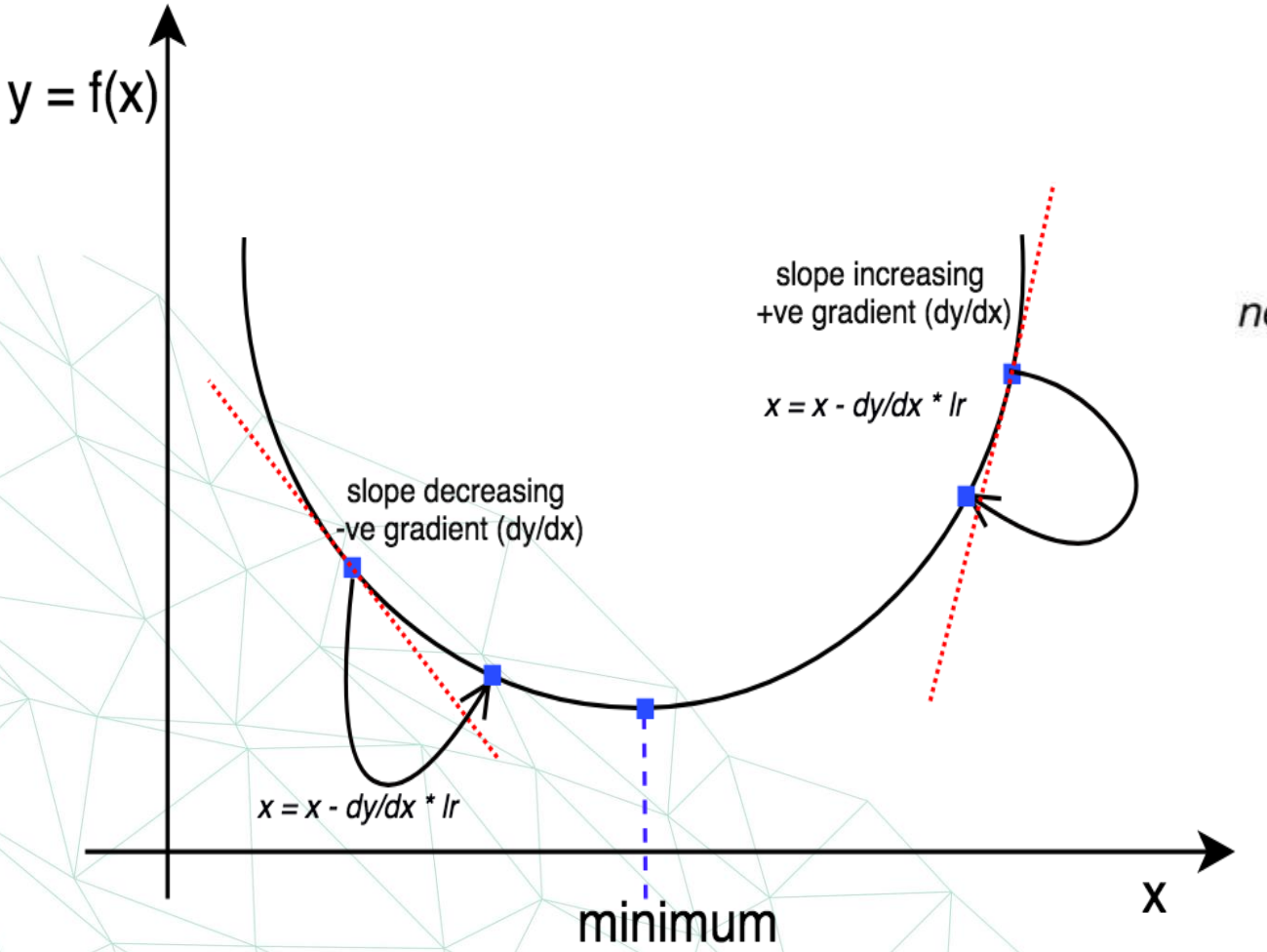
$$\frac{d}{d\,intercept}\text{Sum of squared residuals} = -2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5}) + -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3})) + -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9})) = -5.7$$

Step Size = -5.7 x 0.1 = -0.57



New Intercept = 0 - (-0.57) = **0.57**

…and the the **New Intercept** = **0.57**.



Gradient descent is an iterative optimization algorithm for finding the **minimum** of a function.

4 ●
next position

opposite direction

gradient at current position
-10

$$x_{i+1} = x_i - \alpha \nabla f(x_i)$$

current position
-5 ●

step size (learning rate)
set to 0.9 here, usually much smaller

We will use the Mean Squared Error function to calculate the loss. There are three steps in this function:

- Find the difference between the actual y and predicted y value(y = mx + c), for a given x.
- Square this difference.
- Find the mean of the squares for every value in X.

$$E = \frac{1}{n}\sum_{i=0}^{n}(y_i - \bar{y}_i)^2$$

Here y_i is the actual value and $\bar{y}_i$ is the predicted value. Lets substitute the value of $\bar{y}_i$:

$$E = \frac{1}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))^2$$

Calculating the partial derivative of the cost function with respect to slope (m), let the partial derivative of the cost function with respect to m be D_m

Let's try applying gradient descent to m and c and approach it step by step:

- Initially let m = 0 and c = 0. Let L be our learning rate. This controls how much the value of m changes with each step. L could be a small value like 0.0001 for good accuracy.
- Calculate the partial derivative of the loss function with respect to m, and plug in the current values of x, y, m and c in it to obtain the derivative value D.

$$D_m = \frac{1}{n}\sum_{i=0}^{n} 2(y_i - (mx_i + c))(-x_i)$$

$$D_m = \frac{-2}{n}\sum_{i=0}^{n} x_i(y_i - \bar{y}_i)$$

D_m is the value of the partial derivative with respect to m. Similarly lets find the partial derivative with respect to c, Dc :

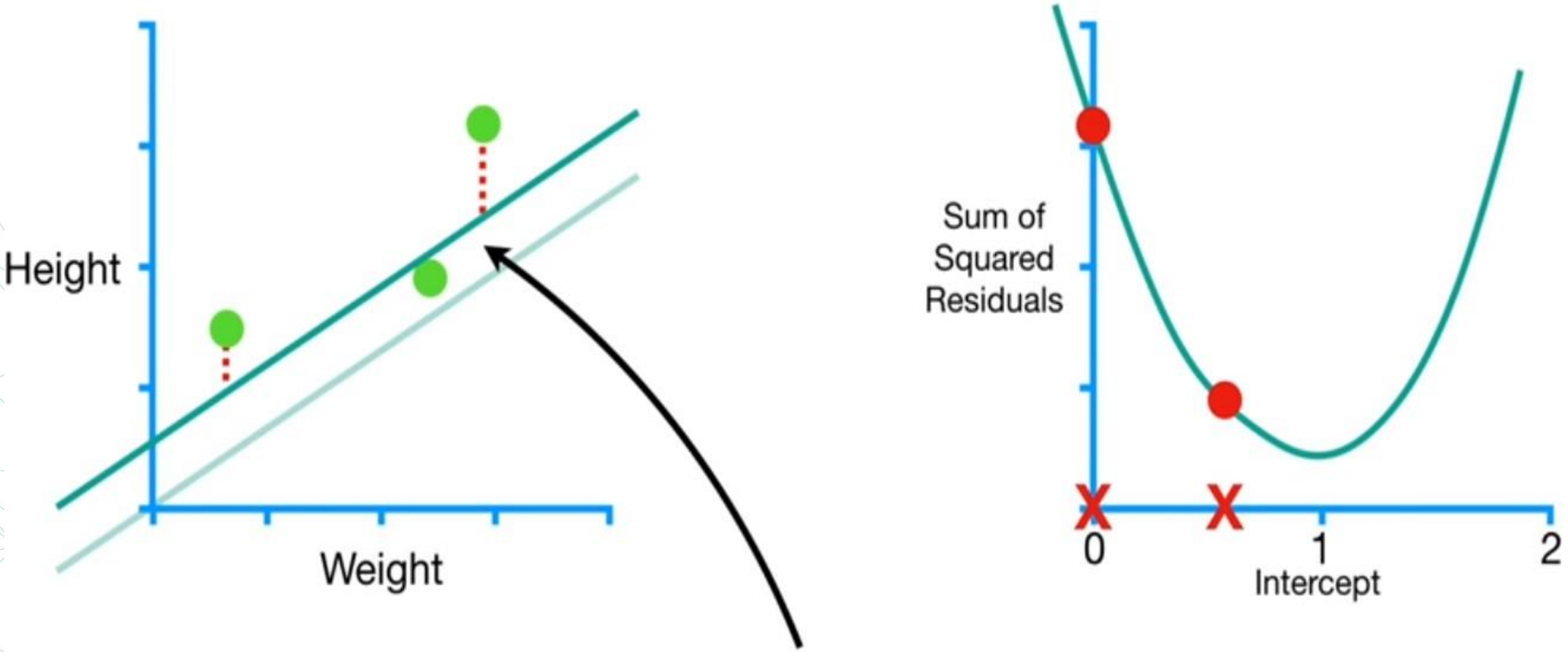
$$D_c = \frac{-2}{n}\sum_{i=0}^{n} (y_i - \bar{y}_i)$$

- Now we update the current value of m and c using the following equation:

$$m = m - L \times D_m$$

$$c = c - L \times D_c$$

- We repeat this process until our loss function is a very small value or ideally 0 (which means 0 error or 100% accuracy). The value of m and c that we are left with now will be the optimum values.



…we can see how much the residuals shrink when the **Intercept = 0.57**.

$$\frac{d}{d\,intercept}\text{Sum of squared residuals} =$$

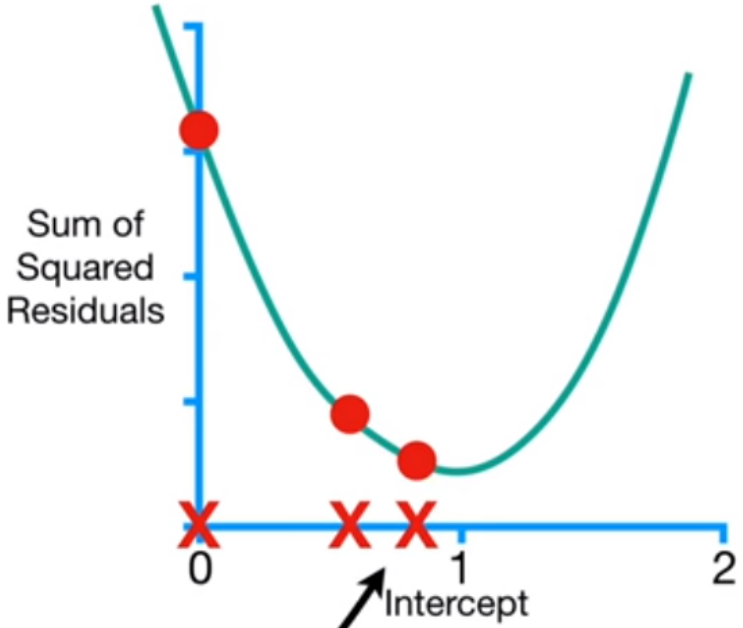
$$-2(\mathbf{1.4} - (0.57 + 0.64 \times \mathbf{0.5}))$$

$$+ -2(\mathbf{1.9} - (0.57 + 0.64 \times \mathbf{2.3}))$$

$$+ -2(\mathbf{3.2} - (0.57 + 0.64 \times \mathbf{2.9}))$$

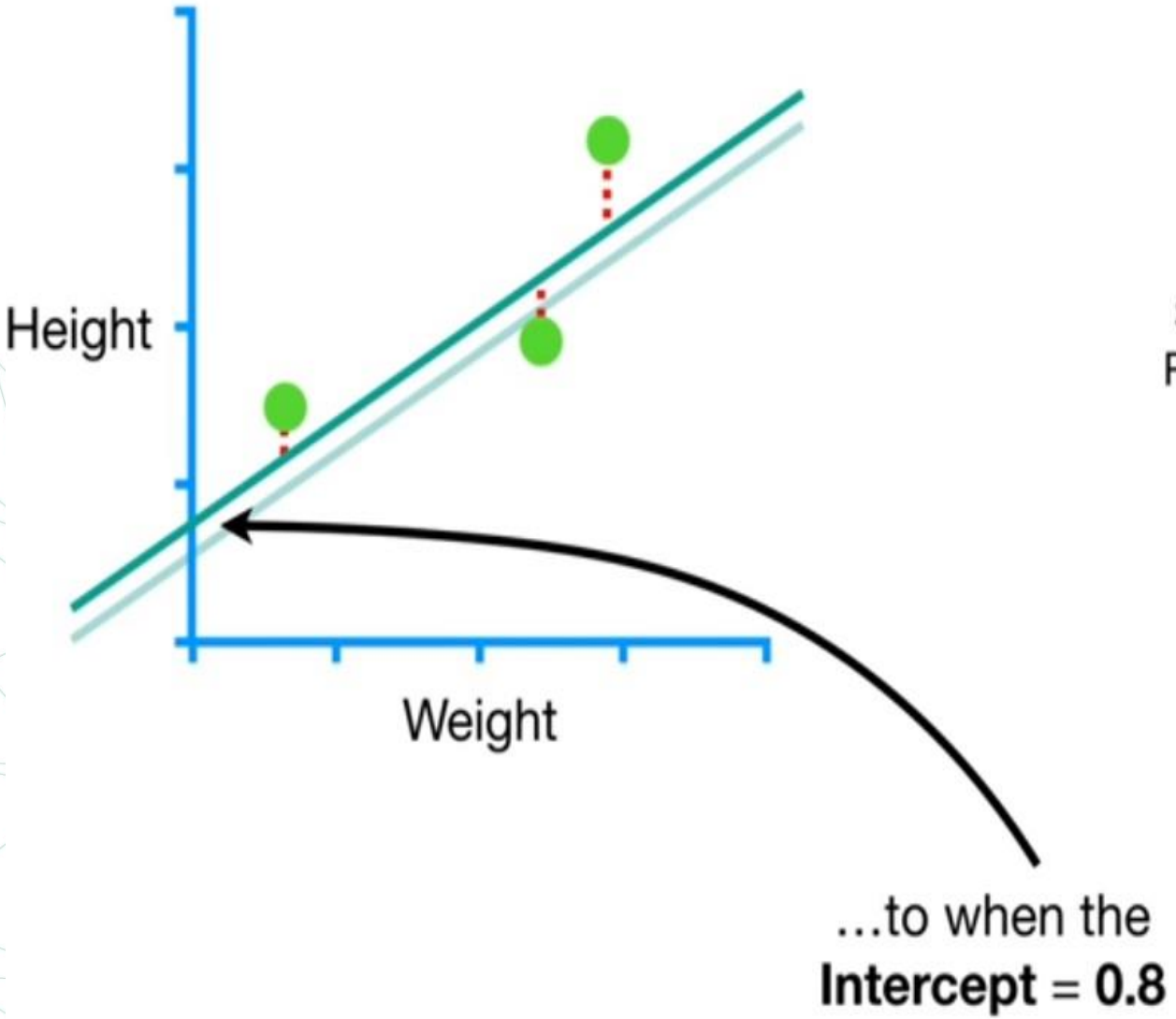
$$= -2.3$$

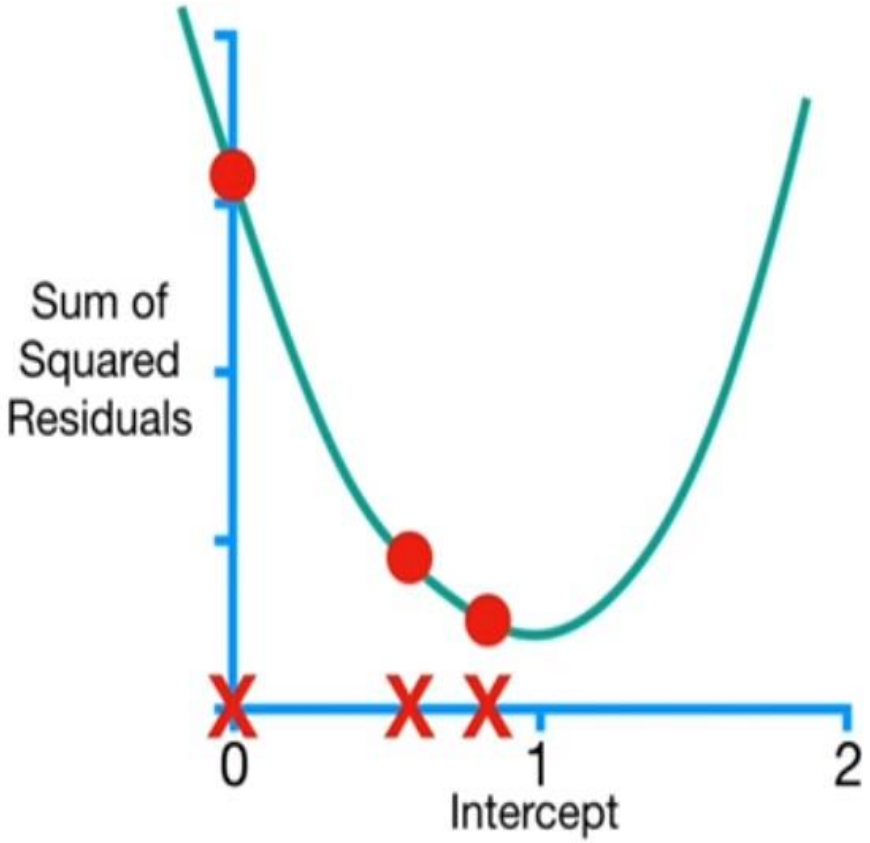
$$\text{Step Size} = -2.3 \times 0.1 = \mathbf{-0.23}$$



New Intercept = 0.57 - (-0.23) = **0.8**

…and the **New Intercept** = **0.8**





...to when the
Intercept = 0.8

$$\frac{d}{d\,intercept}\text{Sum of squared residuals} =$$

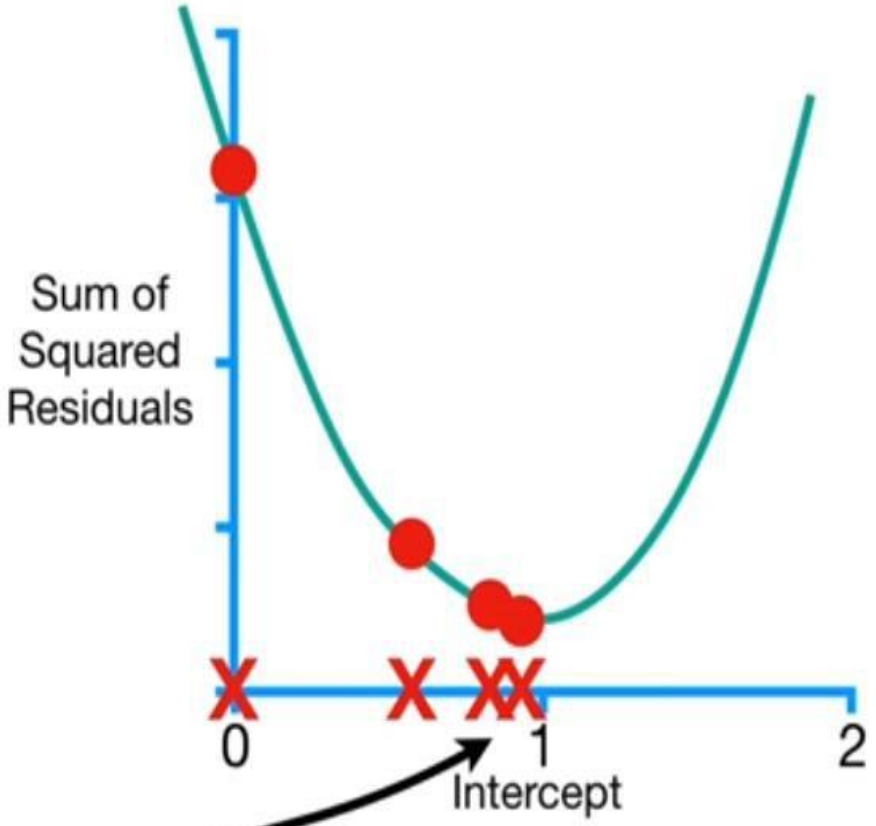
$$-2(\mathbf{1.4} - (0.8 + 0.64 \times \mathbf{0.5}))$$

$$+ -2(\mathbf{1.9} - (0.8 + 0.64 \times \mathbf{2.3}))$$

$$+ -2(\mathbf{3.2} - (0.8 + 0.64 \times \mathbf{2.9}))$$

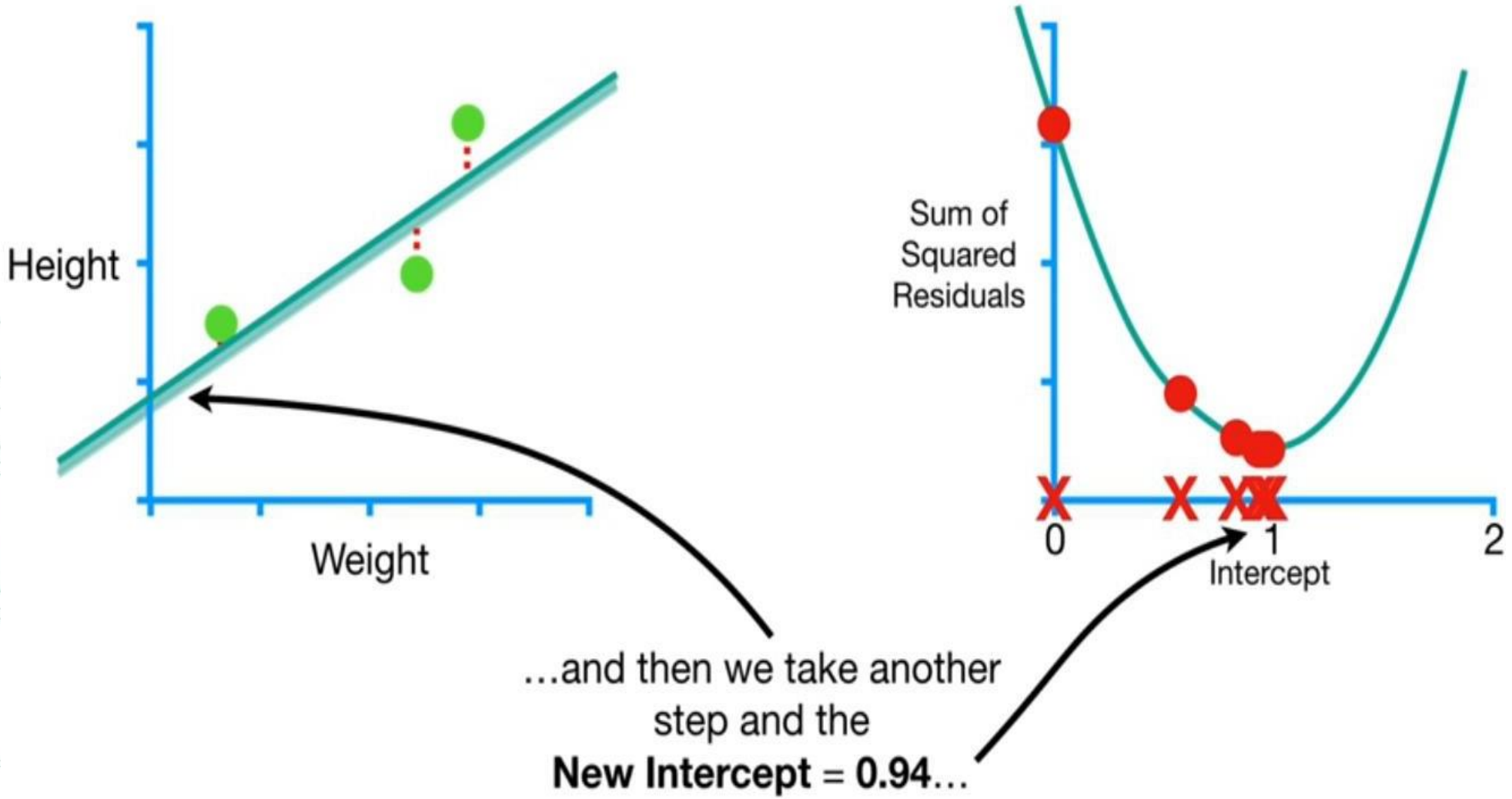
$$= \mathbf{-0.9}$$

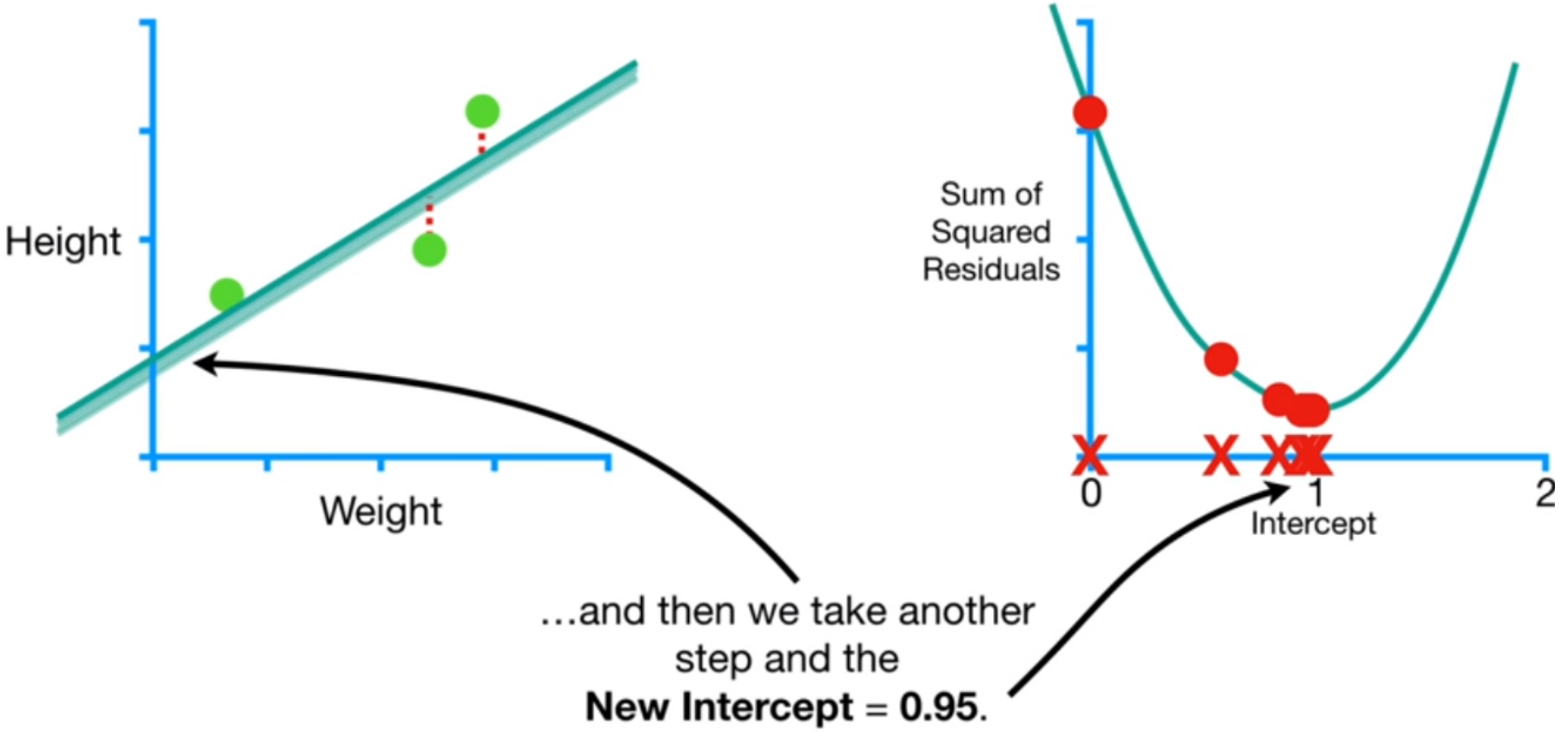
$$\text{Step Size} = -0.9 \times 0.1 = \mathbf{-0.09}$$

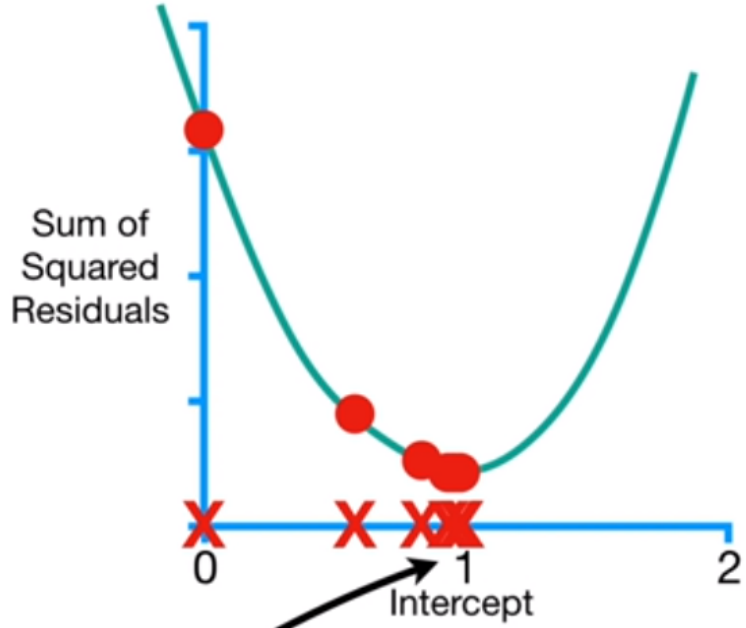


New Intercept = 0.8 - (-0.09) = **0.89**

…and the **New Intercept** = **0.89**


Height
Weight
Sum of
Squared
Residuals
0
1
2
Intercept
...and then we take another
step and the
New Intercept = 0.94...

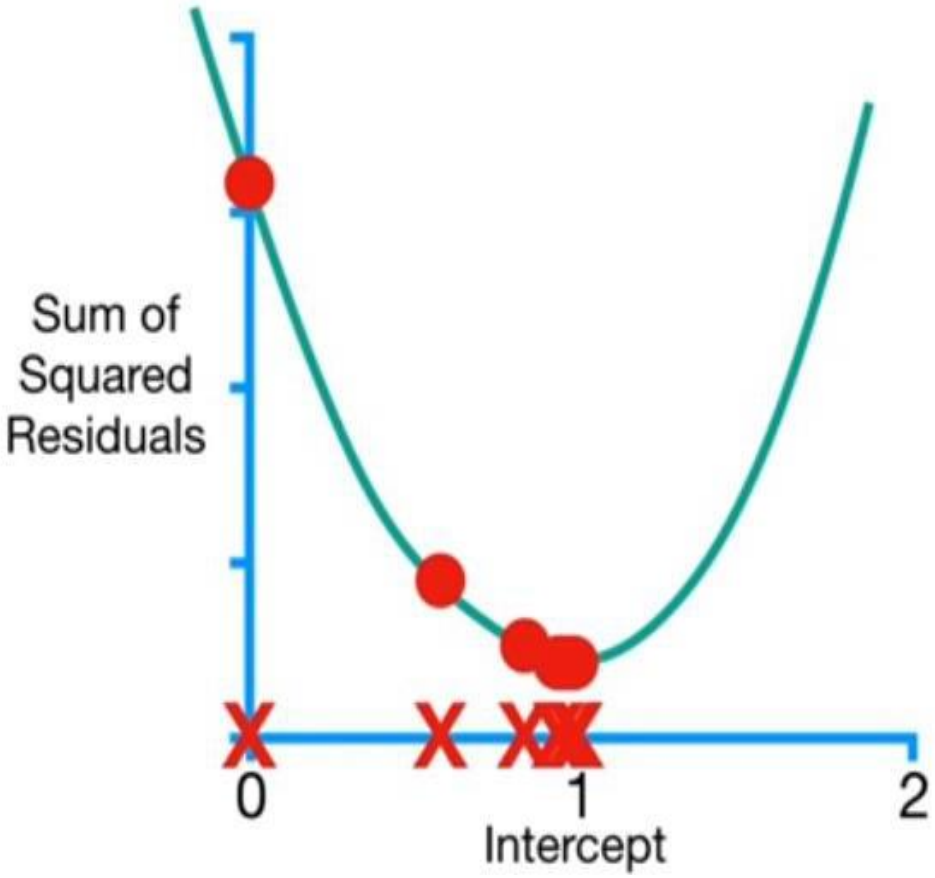

Height
Weight
Sum of
Squared
Residuals
0
1
2
Intercept
...and then we take another
step and the
New Intercept = 0.95.



Notice how each step gets smaller and smaller the closer we get to the bottom of the curve.

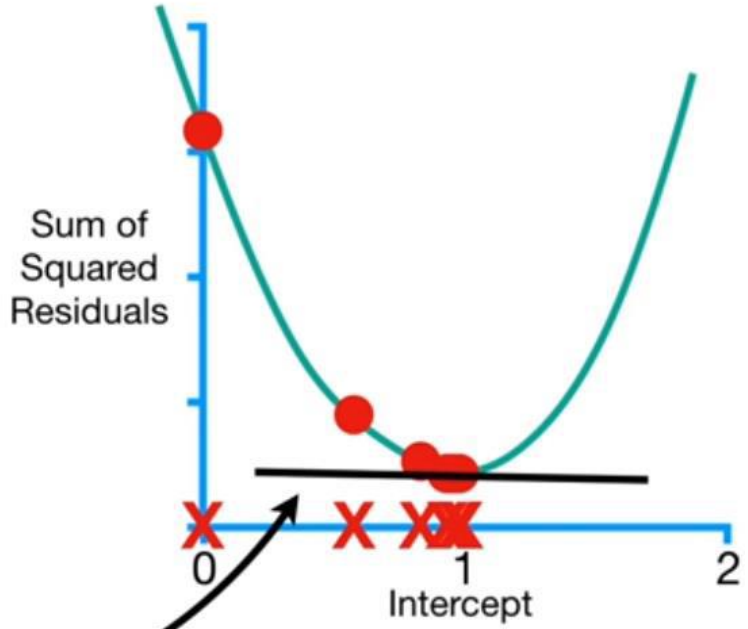
Gradient Descent stops when the **Step Size** is **Very Close To 0.**

Step Size = Slope x Learning Rate



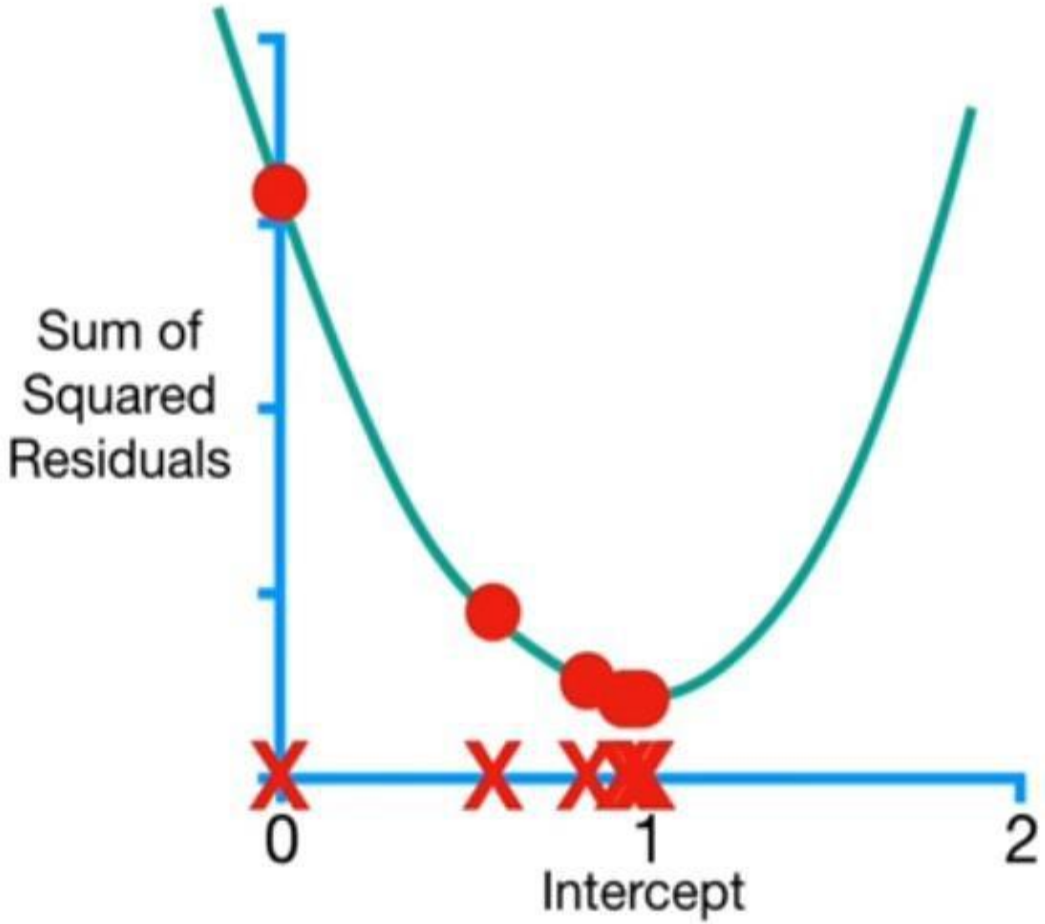
The **Step Size** will be **Very Close to 0** when the **Slope** is very close to **0**.

Step Size = **Slope** × **Learning Rate**

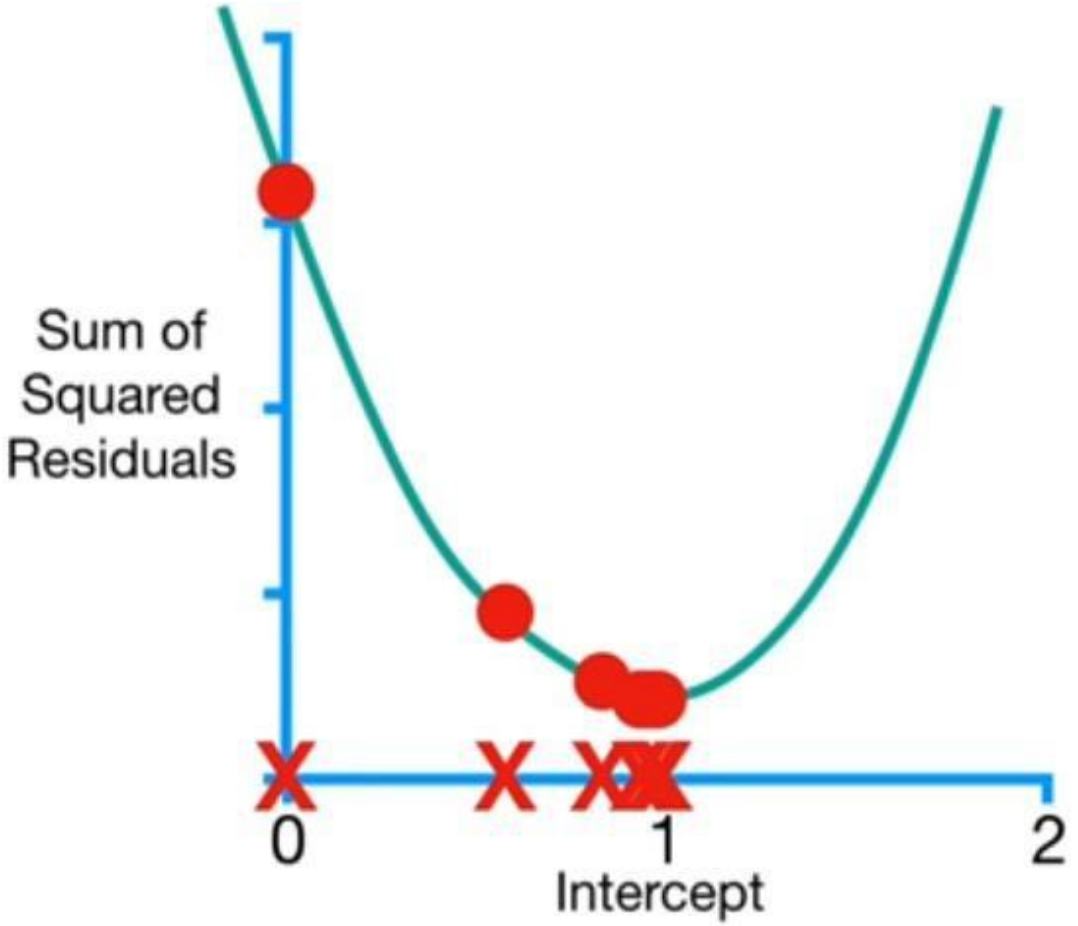


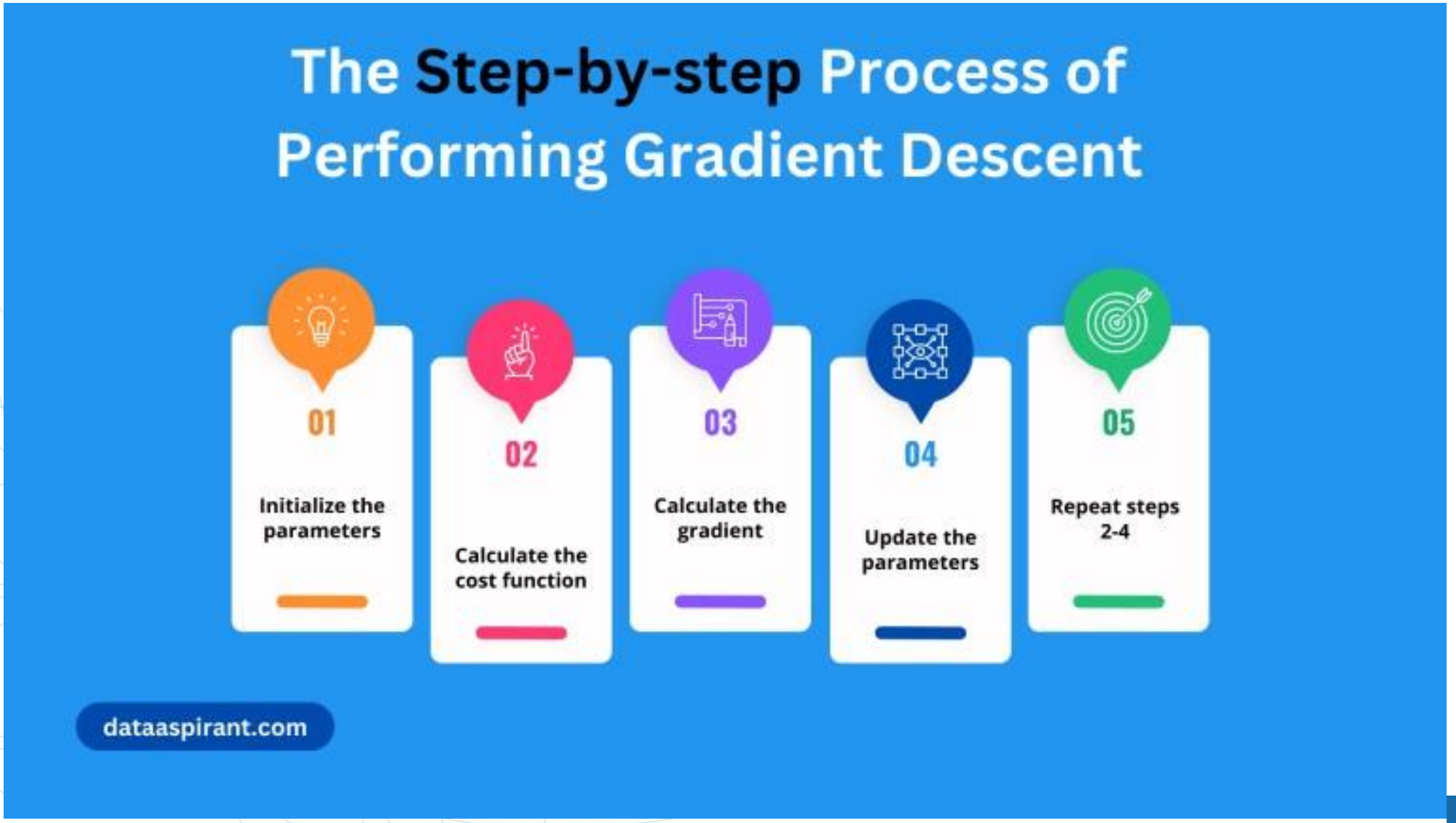
That said, **Gradient Descent** also includes a limit on the number of steps it will take before giving up.

In practice, the **Maximum Number of Steps** = **1,000** or greater.



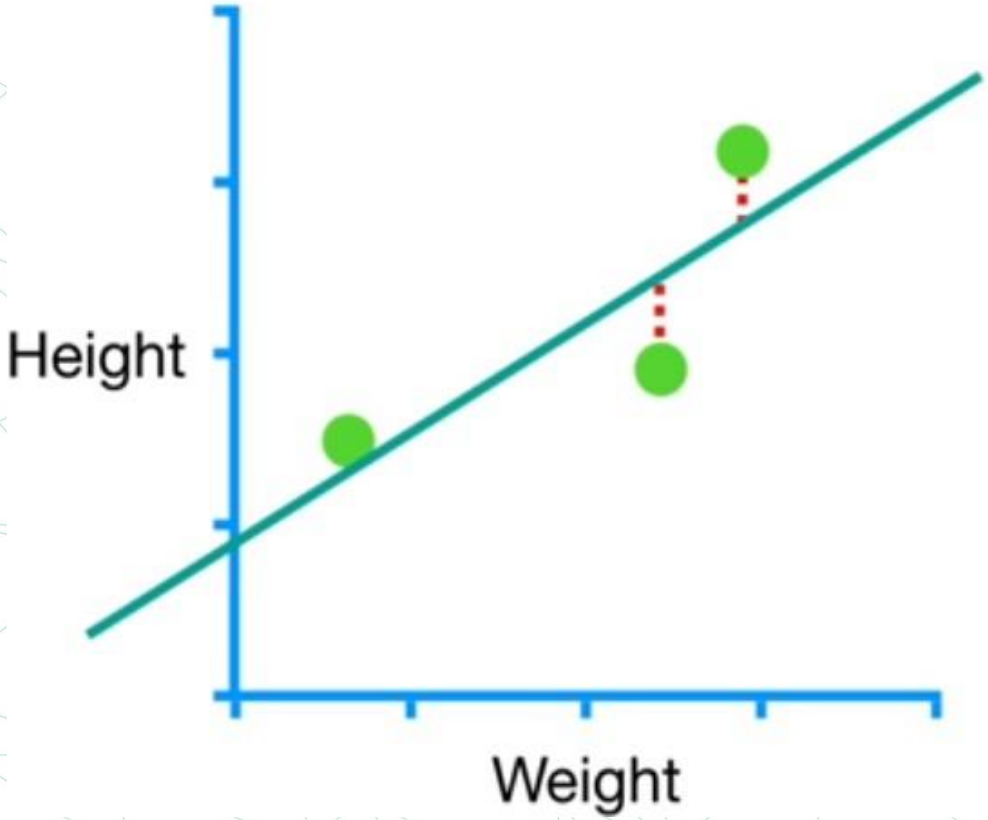
So, even if the **Step Size** is large, if there have been more than the **Maximum Number of Steps**, **Gradient Descent** will stop.



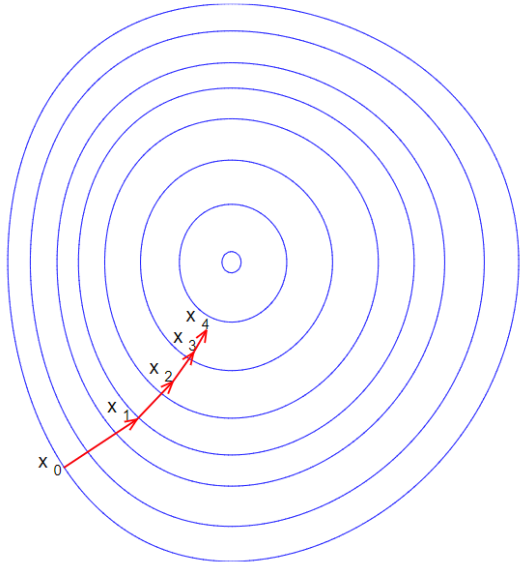

The Step-by-step Process of Performing Gradient Descent
01
Initialize the parameters
02
Calculate the cost function
03
Calculate the gradient
04
Update the parameters
05
Repeat steps 2-4
dataaspirant.com

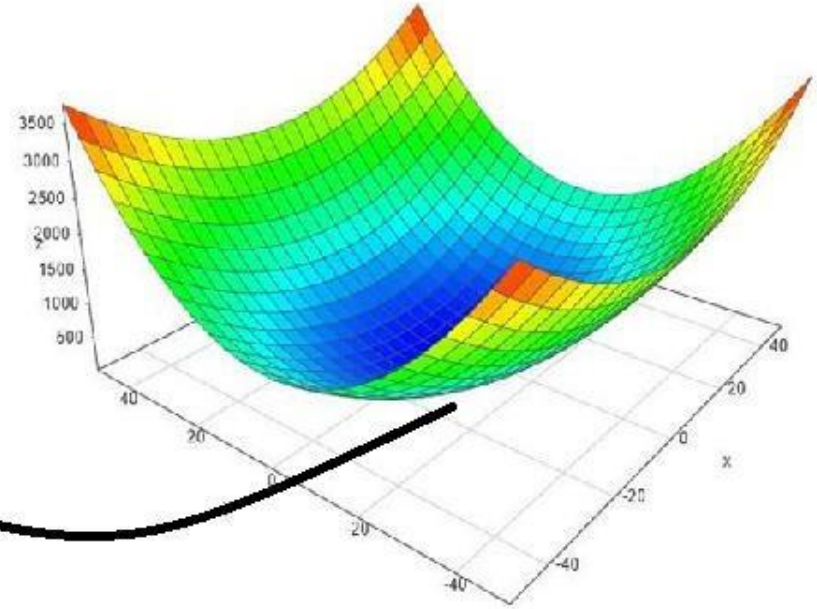
Predicted Height = intercept + slope × **Weight**

…let's talk abut how to estimate the **Intercept** *and* the **Slope**.



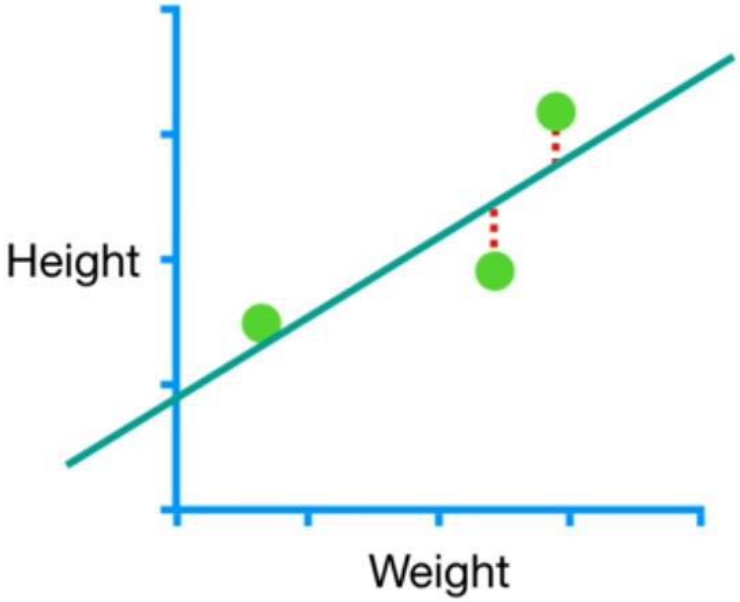
$$\text{Sum of squared residuals} = (\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))^2 + (\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))^2 + (\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))^2$$





We want to find the values for the **Intercept** and **Slope** that give us the minimum Sum of the Squared Residuals.

$$\text{Sum of squared residuals} = (\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))^2 + (\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))^2 + (\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))^2$$



So, just like before, we need to take the derivative of this function…

$$\text{Sum of squared residuals} = (\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))^2 + (\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))^2 + (\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))^2$$

We'll start by taking the derivative with respect to the intercept.

$$\frac{d}{d\,\text{intercept}} \text{Sum of squared residuals} =$$

$$\frac{d}{d\,intercept}\left(\mathbf{1.4} - (intercept + slope \times \mathbf{0.5})\right)^2 = 2(\mathbf{1.4} - (intercept + slope \times \mathbf{0.5})) \times -1$$

$$\frac{d}{d\,intercept}\ \mathbf{1.4} - (intercept + slope \times \mathbf{0.5})$$

$$\frac{d}{d\,intercept}\ \mathbf{1.4} + (-1)intercept - slope \times \mathbf{0.5}$$

Since we are taking the derivative with respect to the **Intercept**, we treat the **Slope** like a constant, and the derivative of a constant is **0**.

...and this whole thing is the derivative of the Sum of the Squared Residuals with respect to the **Intercept**.

$$\frac{d}{d\ intercept}\text{Sum of squared residuals} = -2(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))$$

$$+ -2(\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))$$

$$+ -2(\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))$$

$$\text{Sum of squared residuals} = (\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))^2$$
$$+ (\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))^2$$
$$+ (\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))^2$$

Now let's take the derivative of the Sum of the Squared Residuals with respect to the **Slope**.

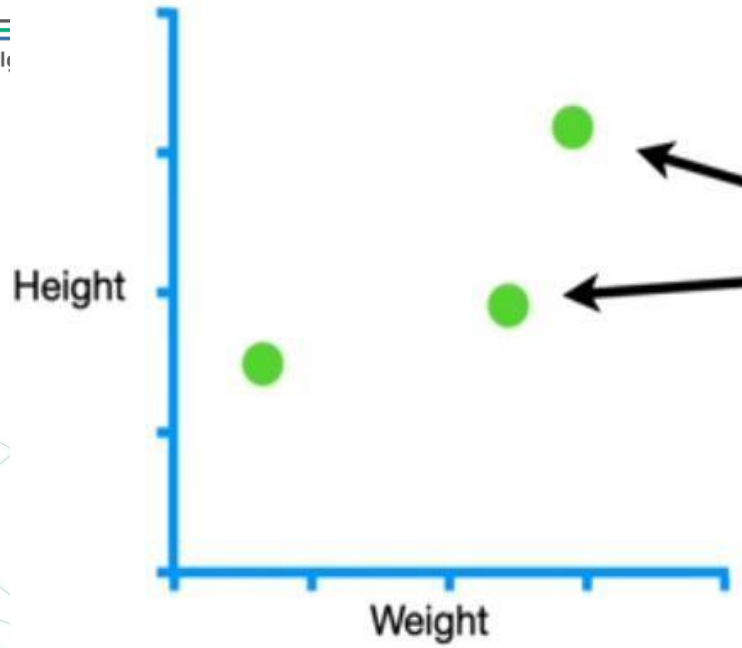
$$\frac{d}{d\,slope}\text{Sum of squared residuals} =$$

$$\frac{d}{d\,slope}\left(\mathbf{1.4} - (intercept + slope \times \mathbf{0.5})\right)^2 = 2(\mathbf{1.4} - (intercept + slope \times \mathbf{0.5})) \times \mathbf{-0.5}$$

$$\frac{d}{d\,slope}\ \mathbf{1.4} - (intercept + slope \times \mathbf{0.5})$$

$$\frac{d}{d\,slope}\ \mathbf{1.4} + (-1)intercept - slope \times \mathbf{0.5}$$

Since we are taking the derivative with respect to the **Slope**, we treat the **Intercept** like a constant, and the derivative of a constant is **0**.



Again, **2.3** and **2.9** are in bold to remind us that they are the weights of the second and third samples.

$$\frac{d}{d\,slope}\text{Sum of squared residuals} = \text{-2} \times \mathbf{0.5}(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))$$

$$+ \text{-2} \times \mathbf{2.3}(\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))$$

$$+ \text{-2} \times \mathbf{2.9}(\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))$$

$$\frac{d}{d\ intercept}\ \text{Sum of squared residuals} =$$

$$-2(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5})$$

$$+ -2(\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))$$

$$+ -2(\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))$$

NOTE: When you have two or more derivatives of the same function, they are called a **Gradient**.

$$\frac{d}{d\ slope}\ \text{Sum of squared residuals} =$$

$$-2 \times \mathbf{0.5}(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))$$

$$+ -2 \times \mathbf{2.9}(\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))$$

$$+ -2 \times \mathbf{2.3}(\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))$$

$$\frac{d}{d\,intercept}\text{Sum of squared residuals} = -2(1.4 - (\text{intercept} + \text{slope} \times 0.5) + -2(1.9 - (\text{intercept} + \text{slope} \times 2.3)) + -2(3.2 - (\text{intercept} + \text{slope} \times 2.9))$$

Just like before, we will start by picking a random number for the **Intercept**. In this case we'll set the **Intercept** = **0**...

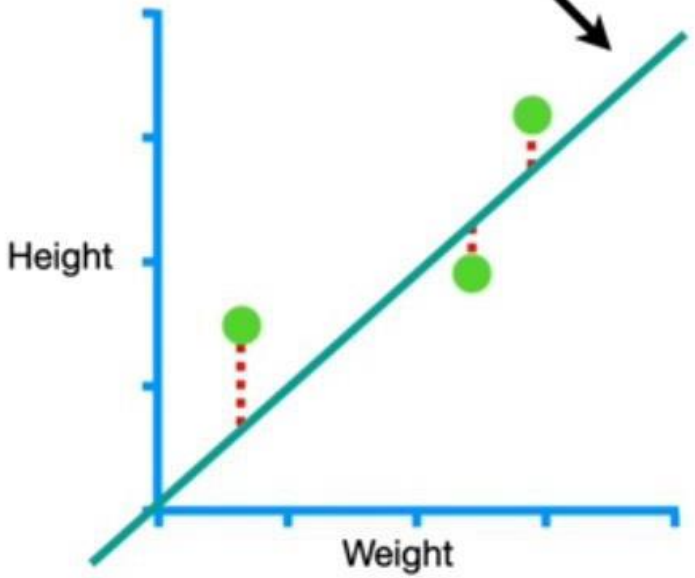
...and we'll pick a random number for the **Slope**. In this case we'll set the **Slope** = **1**.

$$\frac{d}{d\,slope}\text{Sum of squared residuals} = -2 \times 0.5(1.4 - (\text{intercept} + \text{slope} \times 0.5)) + -2 \times 2.9(3.2 - (\text{intercept} + \text{slope} \times 2.9)) + -2 \times 2.3(1.9 - (\text{intercept} + \text{slope} \times 2.3))$$

$$\frac{d}{d\,intercept}\text{Sum of squared residuals} = -2(1.4 - (\text{intercept} + \text{slope} \times 0.5)) + -2(1.9 - (\text{intercept} + \text{slope} \times 2.3)) + -2(3.2 - (\text{intercept} + \text{slope} \times 2.9))$$

$$\frac{d}{d\,slope}\text{Sum of squared residuals} = -2 \times 0.5(1.4 - (\text{intercept} + \text{slope} \times 0.5)) + -2 \times 2.9(3.2 - (\text{intercept} + \text{slope} \times 2.9)) + -2 \times 2.3(1.9 - (\text{intercept} + \text{slope} \times 2.3))$$

Thus, this line, with **Intercept = 0** and **Slope = 1**, is where we will start.



$$\frac{d}{d\,intercept}\text{ Sum of squared residuals} =$$

$$-2(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5})$$

$$+ -2(\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))$$

$$+ -2(\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))$$

Now let's plug in **0** for the **Intercept** and **1** for the **Slope**...

$$\frac{d}{d\,slope}\text{ Sum of squared residuals} =$$

$$-2 \times \mathbf{0.5}(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))$$

$$+ -2 \times \mathbf{2.9}(\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))$$

$$+ -2 \times \mathbf{2.3}(\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))$$

$$\frac{d}{d\,intercept}\text{ Sum of squared residuals} =$$

$$-2(\mathbf{1.4} - (0 + 1 \times \mathbf{0.5})$$

$$+ -2(\mathbf{1.9} - (0 + 1 \times \mathbf{2.3}))$$

$$+ -2(\mathbf{3.2} - (0 + 1 \times \mathbf{2.9})) = \mathbf{-1.6}$$

Step Size$_{\text{Intercept}}$ = -1.6 × **Learning Rate**

…now we plug the **Slopes** into the **Step Size** formulas…

$$\frac{d}{d\,slope}\text{ Sum of squared residuals} =$$

$$-2 \times \mathbf{0.5}(\mathbf{1.4} - (0 + 1 \times \mathbf{0.5}))$$

$$+ -2 \times \mathbf{2.9}(\mathbf{3.2} - (0 + 1 \times \mathbf{2.9}))$$

$$+ -2 \times \mathbf{2.3}(\mathbf{1.9} - (0 + 1 \times \mathbf{2.3})) = \mathbf{-0.8}$$

Step Size$_{\text{Slope}}$ = -0.8 × **Learning Rate**

$$\frac{d}{d\,intercept}\text{Sum of squared residuals} = -2(1.4 - (0 + 1 \times 0.5)) + -2(1.9 - (0 + 1 \times 2.3)) + -2(3.2 - (0 + 1 \times 2.9)) = -1.6$$

$$\text{Step Size}_{\text{Intercept}} = -1.6 \times 0.01$$

NOTE: The larger **Learning Rate** that we used in the first example doesn't work this time. Even after a bunch of steps, **Gradient Descent** doesn't arrive at the correct answer.

$$\frac{d}{d\,slope}\text{Sum of squared residuals} = -2 \times 0.5(1.4 - (0 + 1 \times 0.5)) + -2 \times 2.9(3.2 - (0 + 1 \times 2.9)) + -2 \times 2.3(1.9 - (0 + 1 \times 2.3)) = -0.8$$

$$\text{Step Size}_{\text{Slope}} = -0.8 \times 0.01$$

$$\frac{d}{d\,intercept}\text{Sum of squared residuals} = -2(1.4 - (0 + 1 \times 0.5) + -2(1.9 - (0 + 1 \times 2.3)) + -2(3.2 - (0 + 1 \times 2.9)) = -1.6$$

Step Size$_{Intercept}$ = -1.6 x 0.01 = **-0.016**

Anyway, we do the math and get two **Step Sizes.**

$$\frac{d}{d\,slope}\text{Sum of squared residuals} = -2 \times 0.5(1.4 - (0 + 1 \times 0.5)) + -2 \times 2.9(3.2 - (0 + 1 \times 2.9)) + -2 \times 2.3(1.9 - (0 + 1 \times 2.3)) = -0.8$$

Step Size$_{Slope}$ = -0.8 x 0.01 = **-0.008**

$$\frac{d}{d\,intercept}\text{ Sum of squared residuals} =$$

$$-2(\mathbf{1.4} - (0 + 1 \times \mathbf{0.5})$$

$$+ -2(\mathbf{1.9} - (0 + 1 \times \mathbf{2.3}))$$

$$+ -2(\mathbf{3.2} - (0 + 1 \times \mathbf{2.9})) = \mathbf{-1.6}$$

$$\text{Step Size}_{\text{Intercept}} = -1.6 \times 0.01 = \mathbf{-0.016}$$

$$\mathbf{New\ Intercept = Old\ Intercept - Step\ Size}$$

Now we calculate the **New Intercept** and **New Slope** by plugging in the **Old Intercept** and the **Old Slope**...

$$\frac{d}{d\,slope}\text{ Sum of squared residuals} =$$

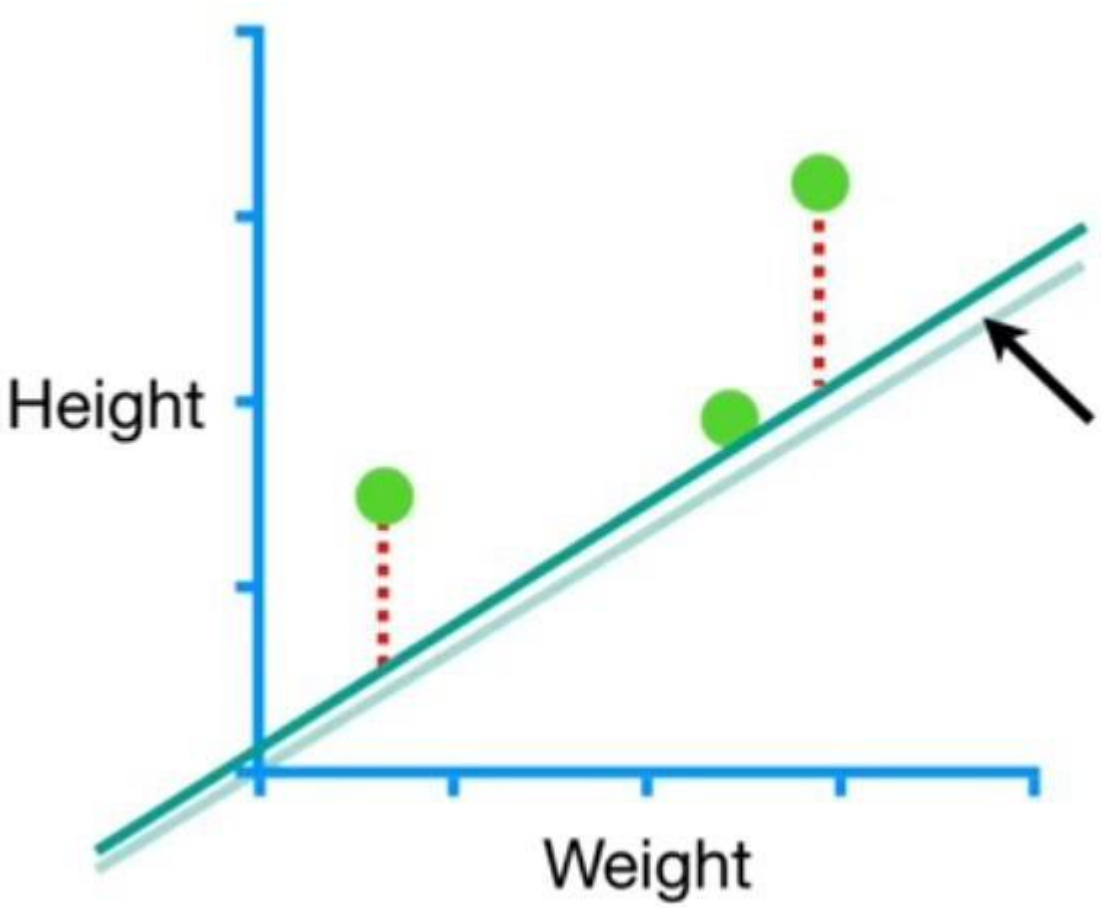
$$-2 \times \mathbf{0.5}(\mathbf{1.4} - (0 + 1 \times \mathbf{0.5}))$$

$$+ -2 \times \mathbf{2.9}(\mathbf{3.2} - (0 + 1 \times \mathbf{2.9}))$$

$$+ -2 \times \mathbf{2.3}(\mathbf{1.9} - (0 + 1 \times \mathbf{2.3})) = \mathbf{-0.8}$$

$$\text{Step Size}_{\text{Slope}} = -0.8 \times 0.01 = \mathbf{-0.008}$$

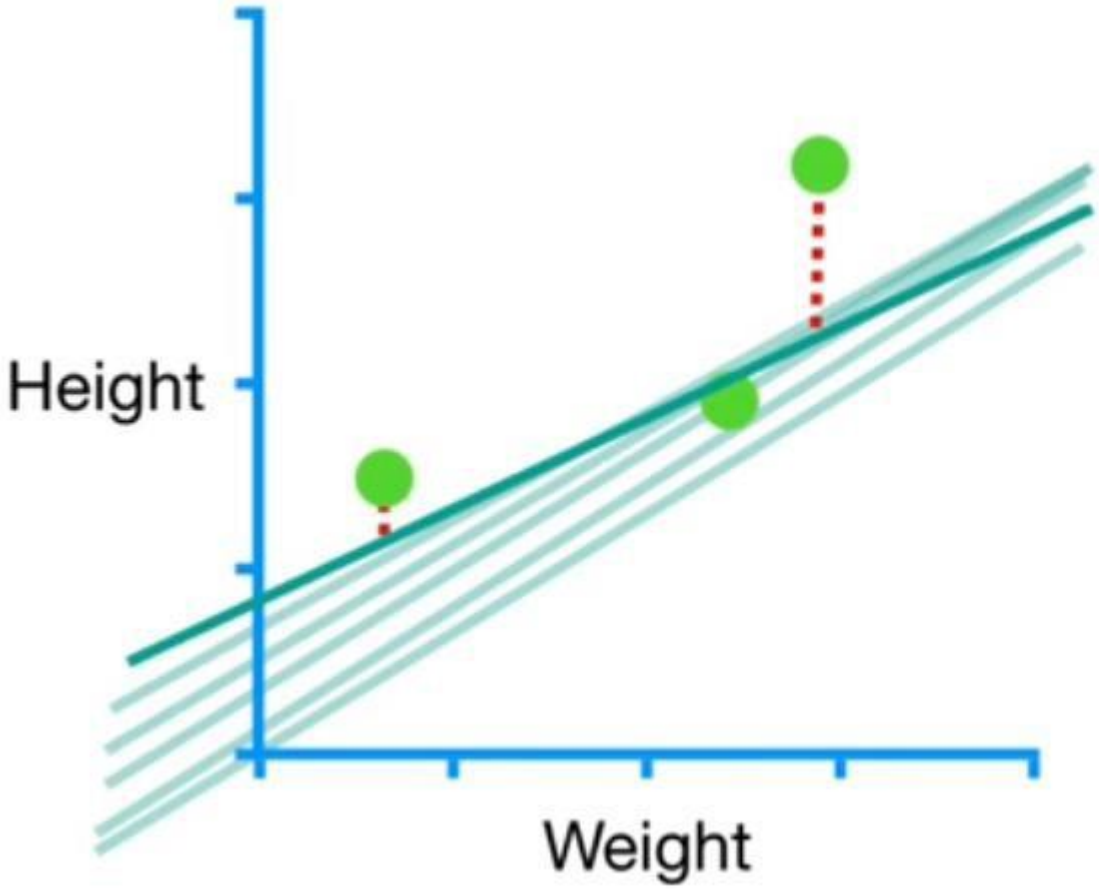
$$\mathbf{New\ Slope = Old\ Slope - Step\ Size}$$



New Intercept = 0 - (-0.016) = 0.016

...and this is the new line (with **Slope = 1.008** and **Intercept = 0.016**) after the first step.

New Slope = 1 - (-0.008) = 1.008



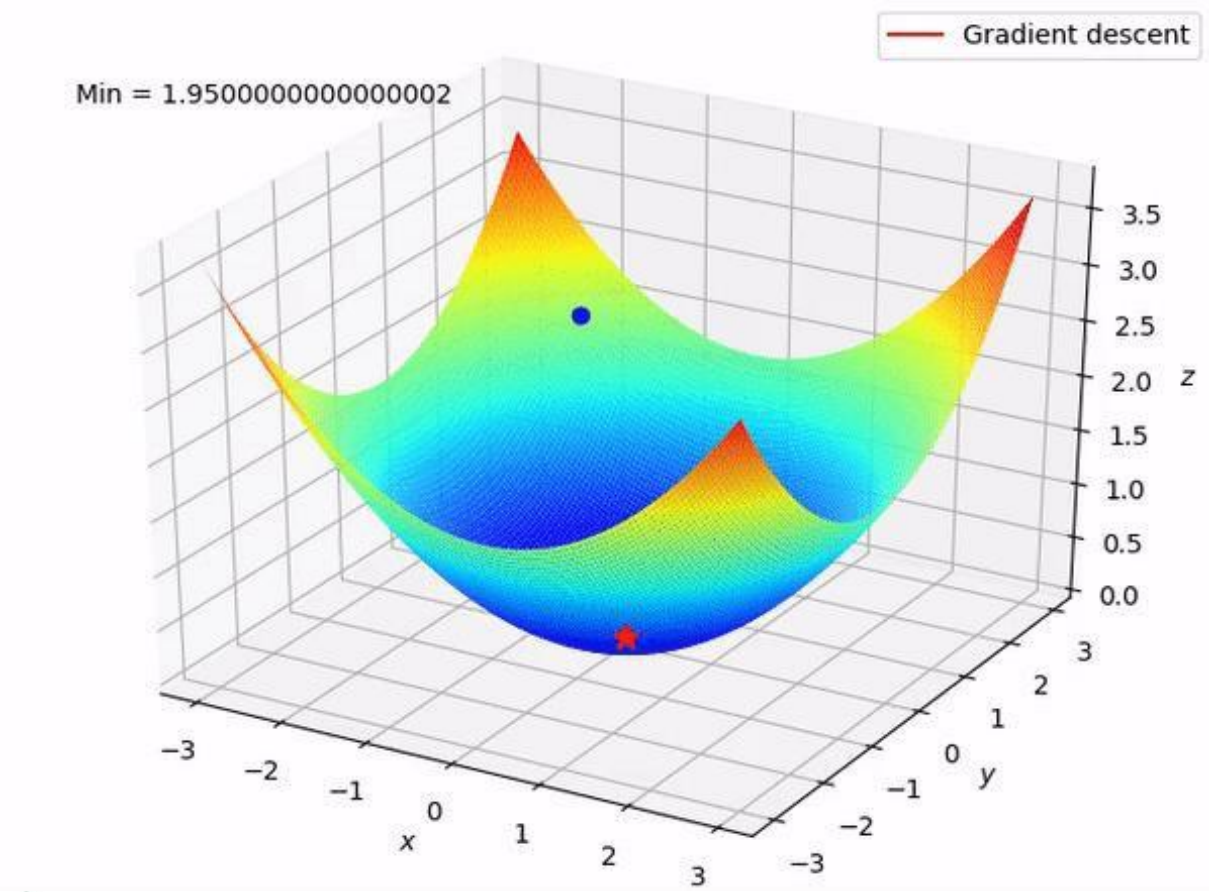
Now we just repeat what we did until all of the **Steps Sizes** are very small or we reach the **Maximum Number of Steps**.

```python
import numpy as np

def gradient_descent(start, gradient, learn_rate, max_iter, tol=0.01):
  steps = [start] # history tracking
  x = start

  for _ in range(max_iter):
    diff = learn_rate*gradient(x)
    if np.abs(diff)<tol:
      break
    x = x - diff
    steps.append(x) # history tracing

  return steps, x
```



Cost Function

$$J(\Theta_0, \Theta_1) = \frac{1}{2m}\sum_{i=1}^{m}[h_\Theta(x_i) - y_i]^2$$

$h_\Theta(x_i)$: Predicted Value; y_i: True Value

Gradient Descent

$$\Theta_j = \Theta_j - \alpha\frac{\partial}{\partial\Theta_j}J(\Theta_0, \Theta_1)$$

α: Learning Rate

Now,

$$\frac{\partial}{\partial\Theta}J_\Theta = \frac{\partial}{\partial\Theta}\frac{1}{2m}\sum_{i=1}^{m}[h_\Theta(x_i) - y]^2$$

$$= \frac{1}{m}\sum_{i=1}^{m}(h_\Theta(x_i) - y)\frac{\partial}{\partial\Theta_j}(\Theta x_i - y)$$

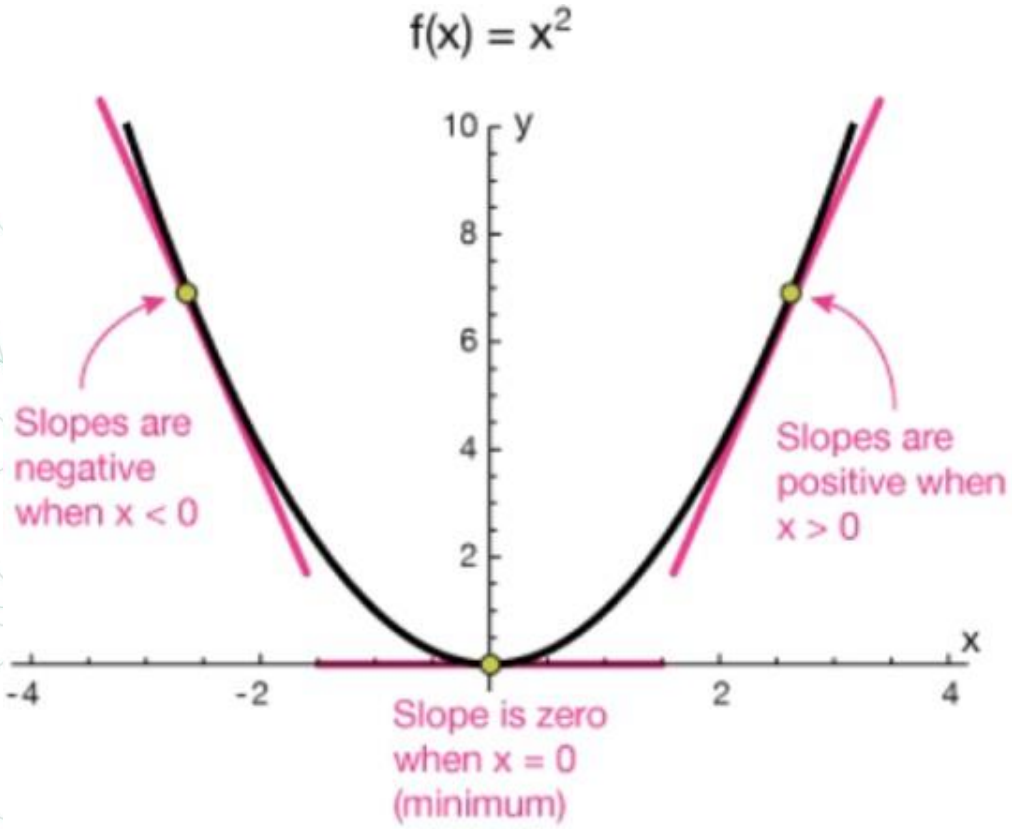
$$= \frac{1}{m}(h_\Theta(x_i) - y)x_i$$

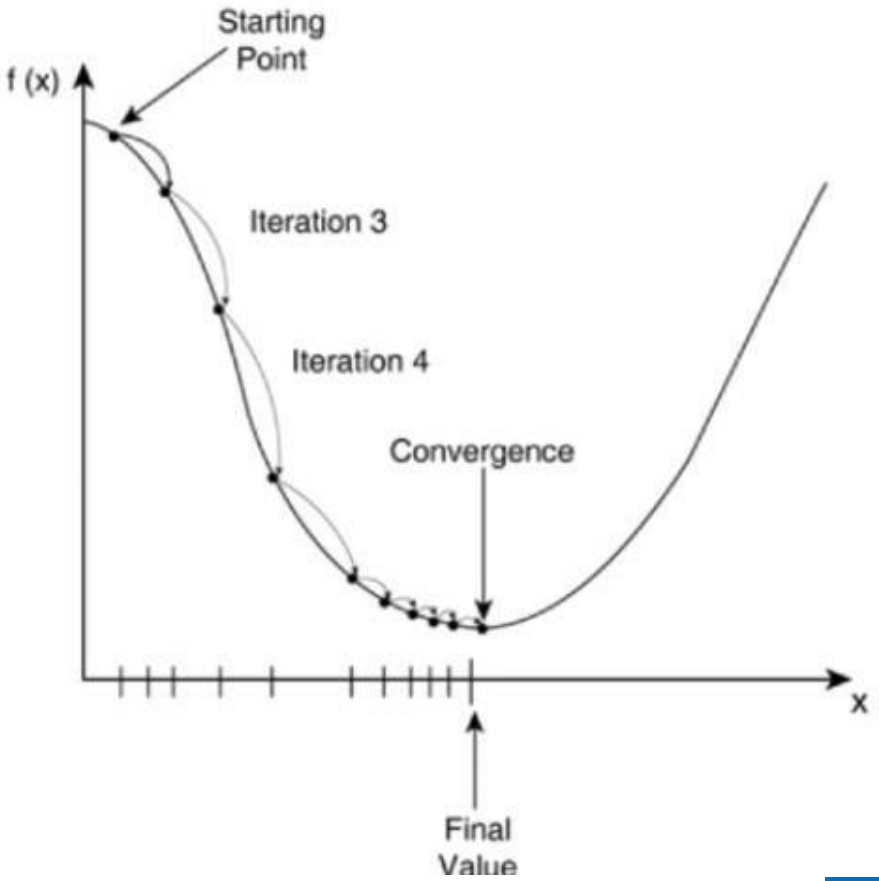
Therefore,

$$\Theta_j := \Theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}[(h_\Theta(x_i) - y)x_i]$$

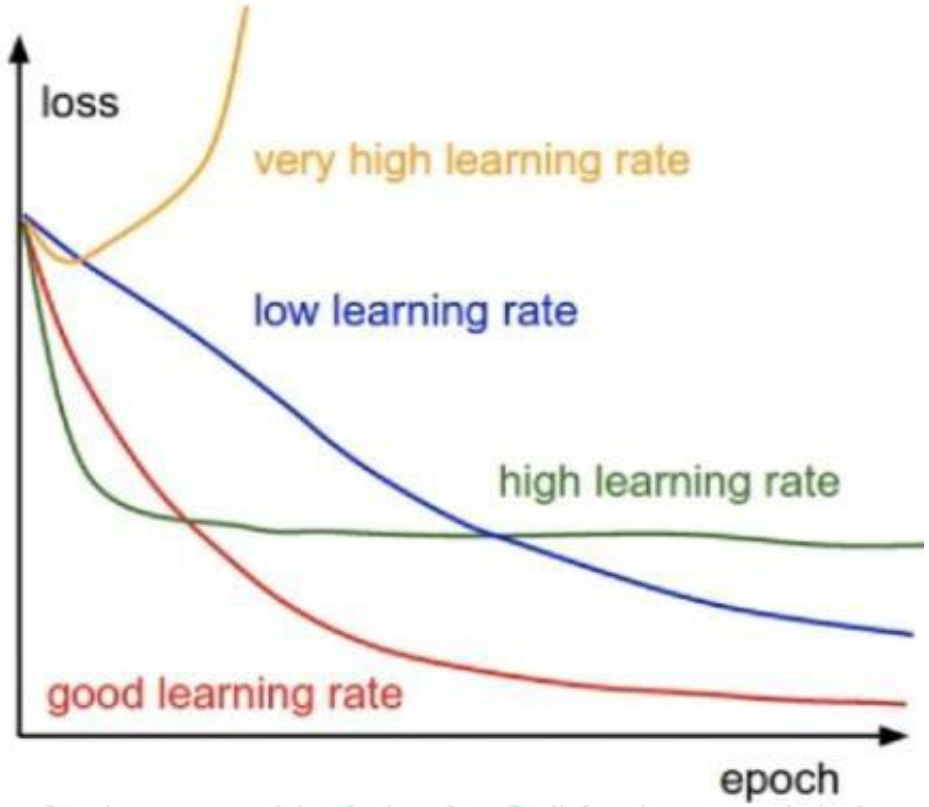
$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

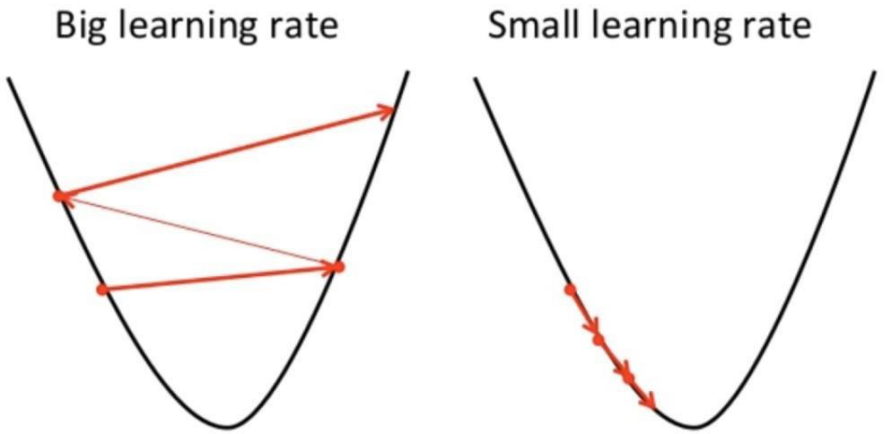
$$\theta_1 := \theta_1 - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)\cdot x^{(i)}$$


f(x) = x²
y
10
8
6
4
2
x
-4
-2
2
4
Slopes are negative when x < 0
Slopes are positive when x > 0
Slope is zero when x = 0 (minimum)


f (x)
Starting Point
Iteration 3
Iteration 4
Convergence
x
Final Value

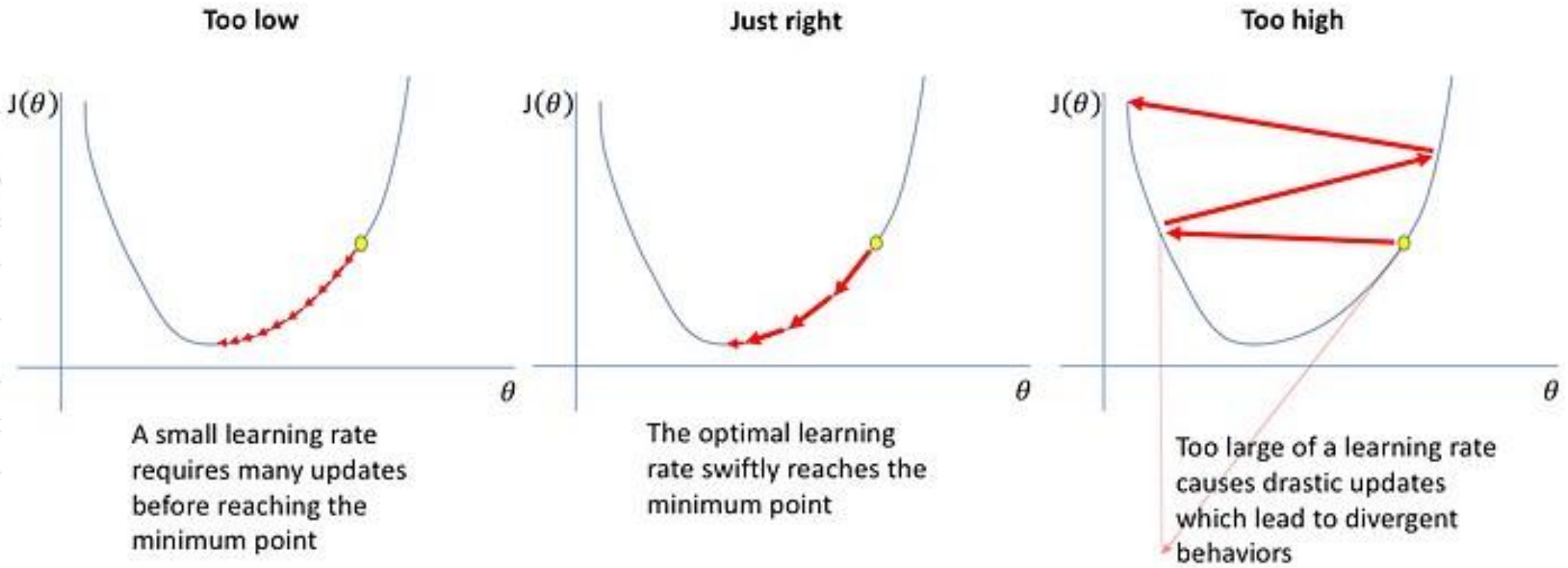
How to choose learning rate?



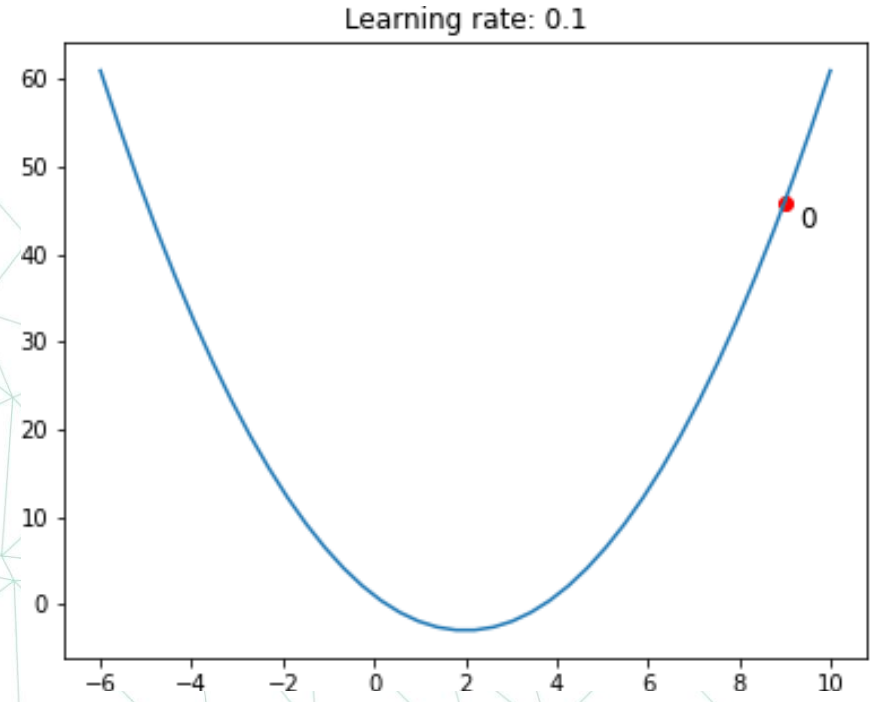


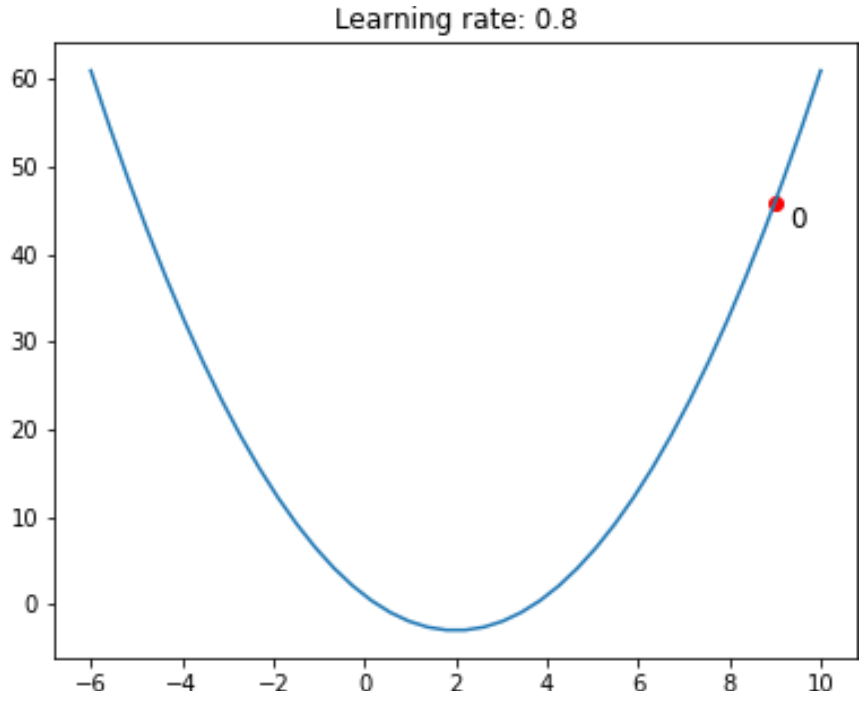
https://uclaacm.github.io/gradient-descent-visualiser/#playground

How to choose learning rate?



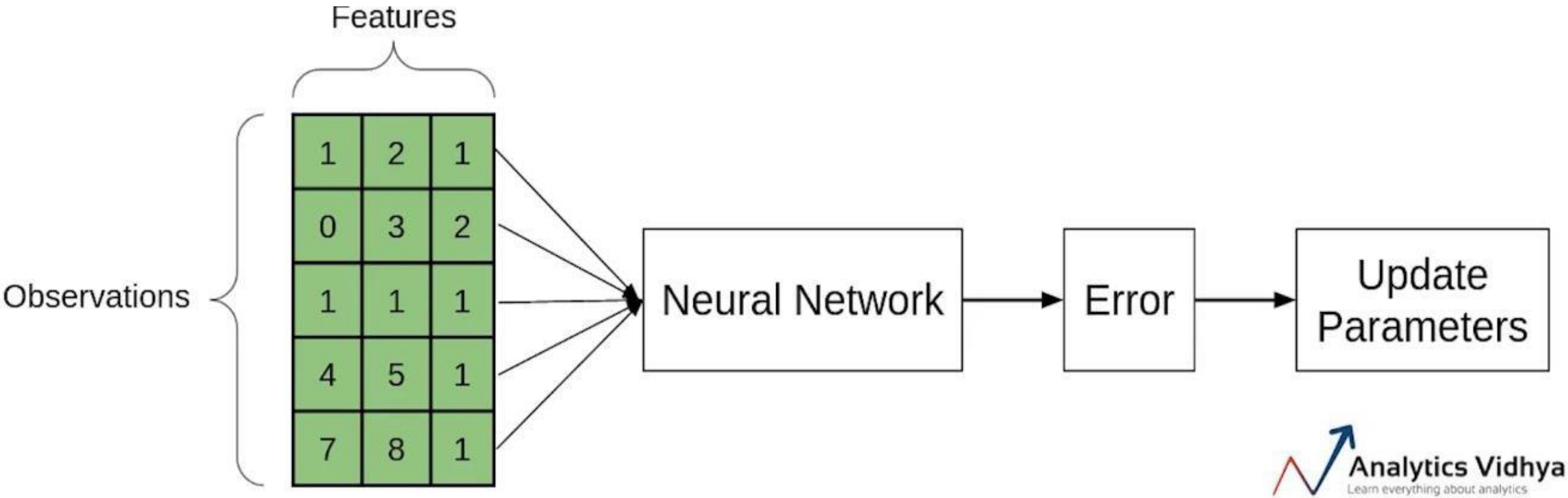
How to choose learning rate?





Variants of Gradient Descent Algorithm

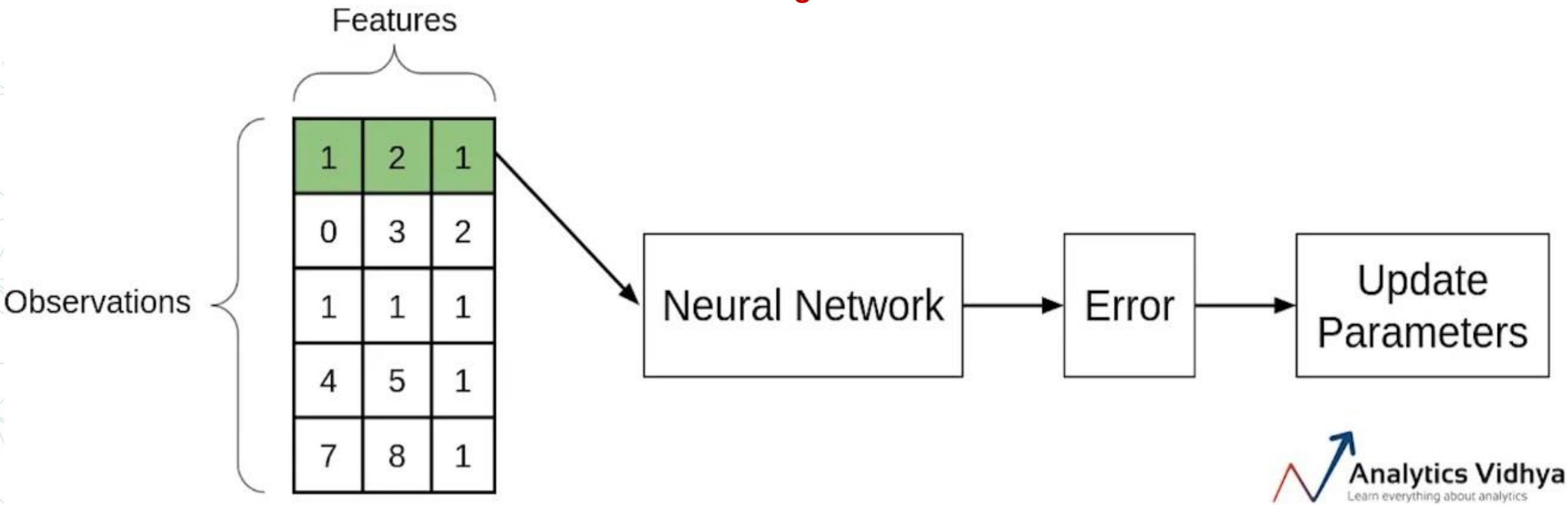
Batch gradient descent



Analytics Vidhya
Learn everything about analytics

Variants of Gradient Descent Algorithm

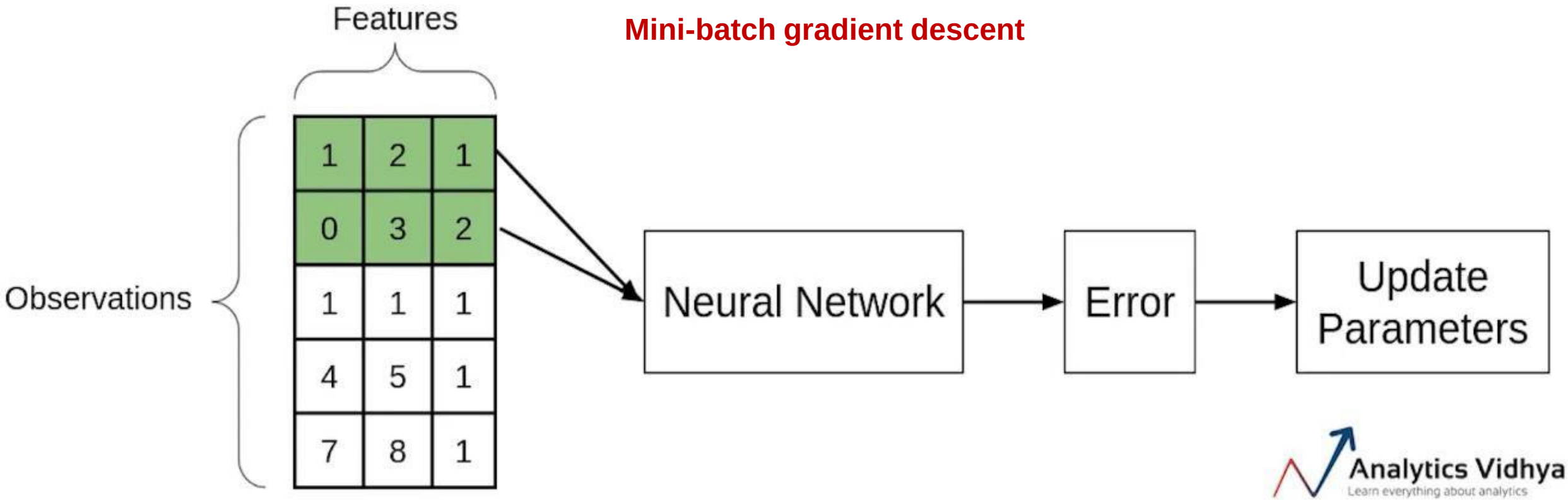
Stochastic gradient descent



Analytics Vidhya
Learn everything about analytics

Variants of Gradient Descent Algorithm

Mini-batch gradient descent

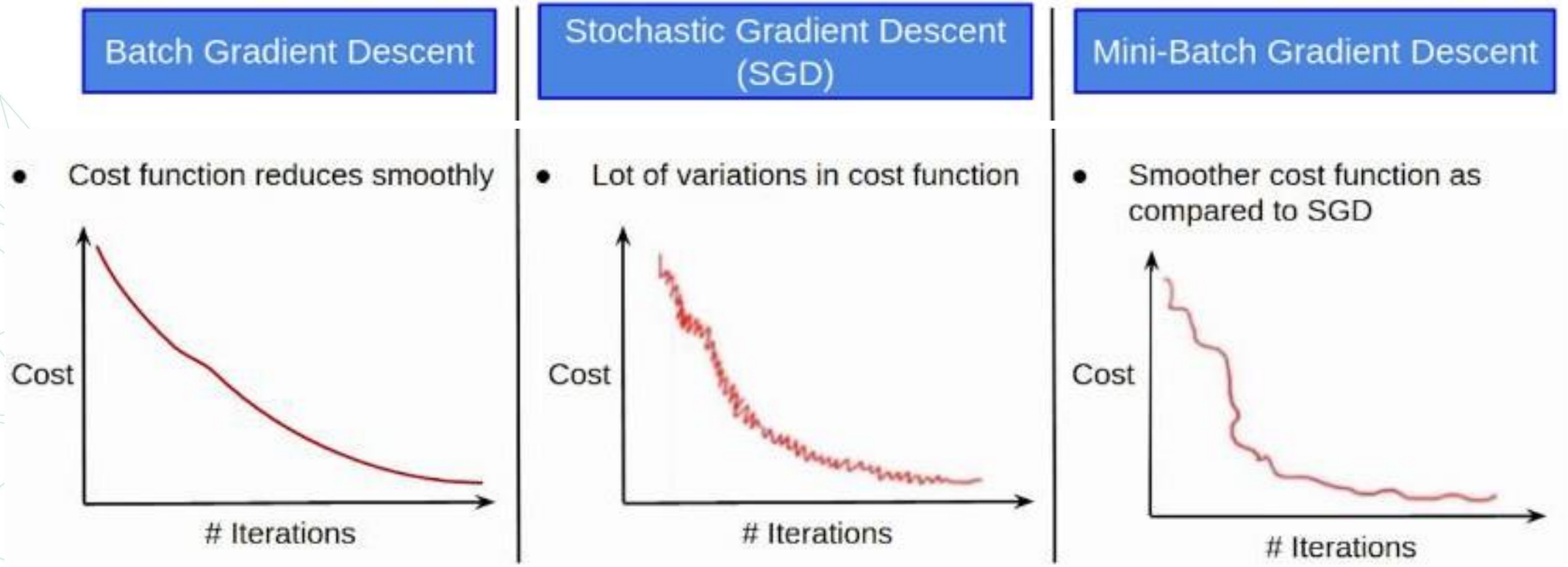


Analytics Vidhya
Learn everything about analytics

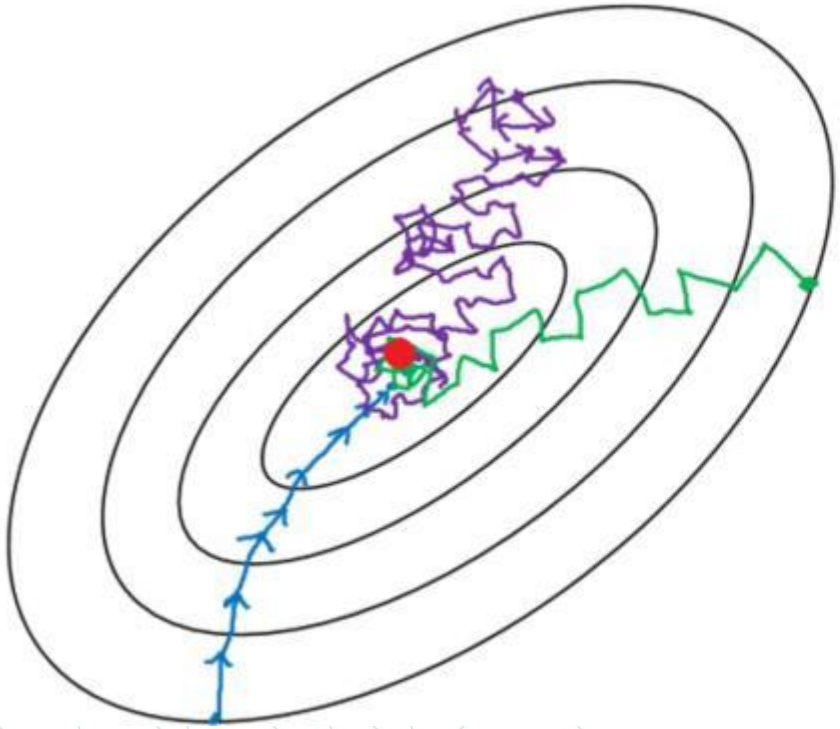
Variants of Gradient Descent Algorithm

Batch Gradient Descent	Stochastic Gradient Descent (SGD)	Mini-Batch Gradient Descent
• Entire dataset for updation	• Single observation for updation	• Subset of data for updation
• Cost function reduces smoothly	• Lot of variations in cost function	• Smoother cost function as compared to SGD
• Computation cost is very high	• Computation time is more	• Computation time is lesser than SGD
		• Computation cost is lesser than Batch Gradient Descent

Variants of Gradient Descent Algorithm

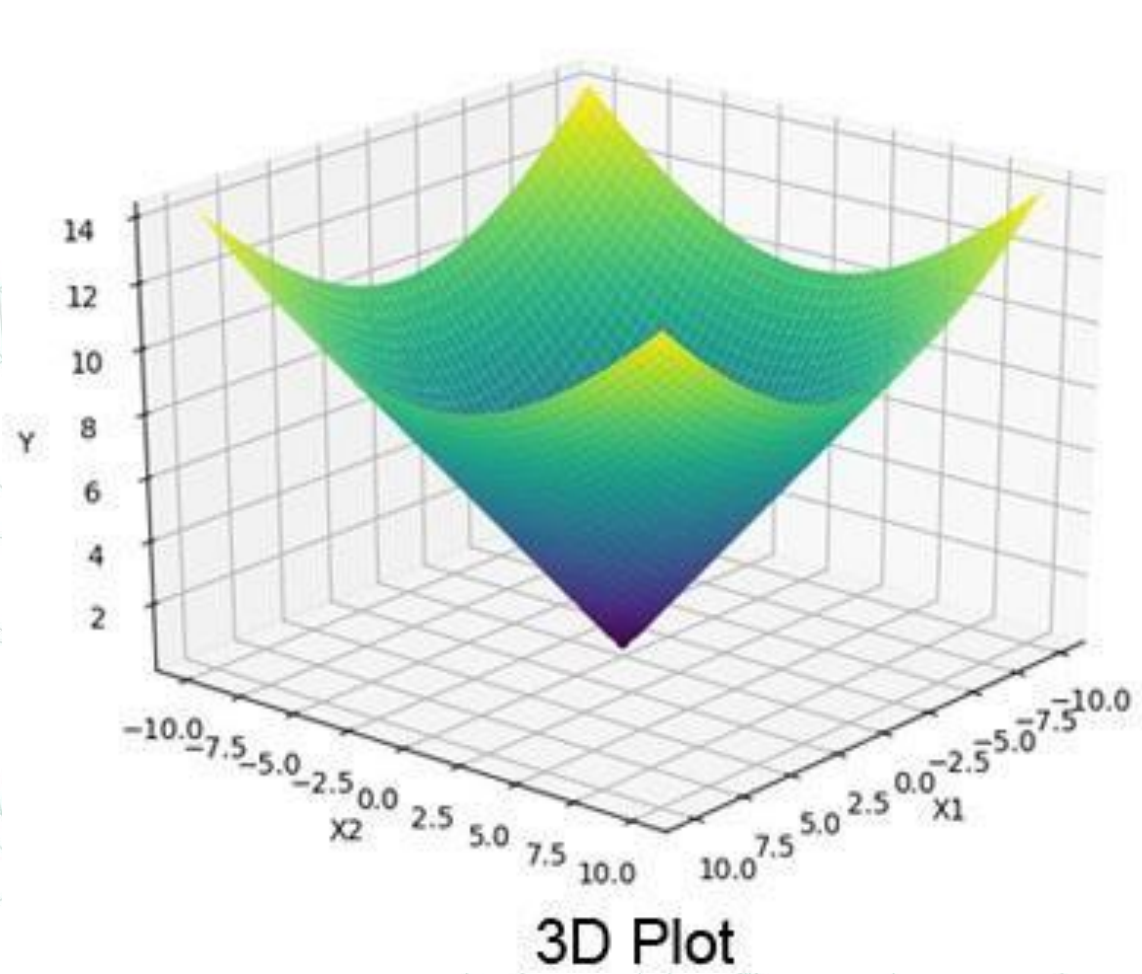


Variants of Gradient Descent Algorithm

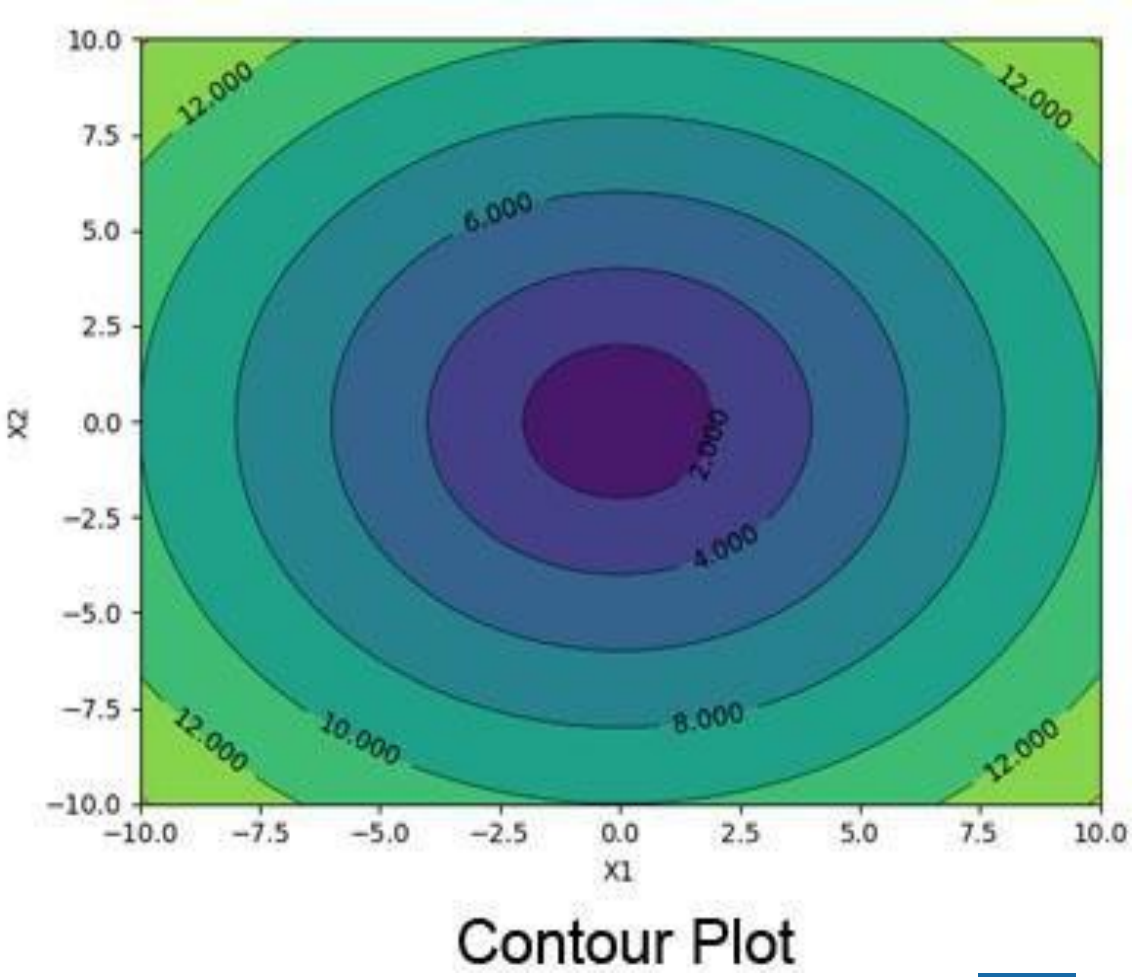


Batch gradient descent
Mini-batch gradient Descent
Stochastic gradient descent

Contour Plot

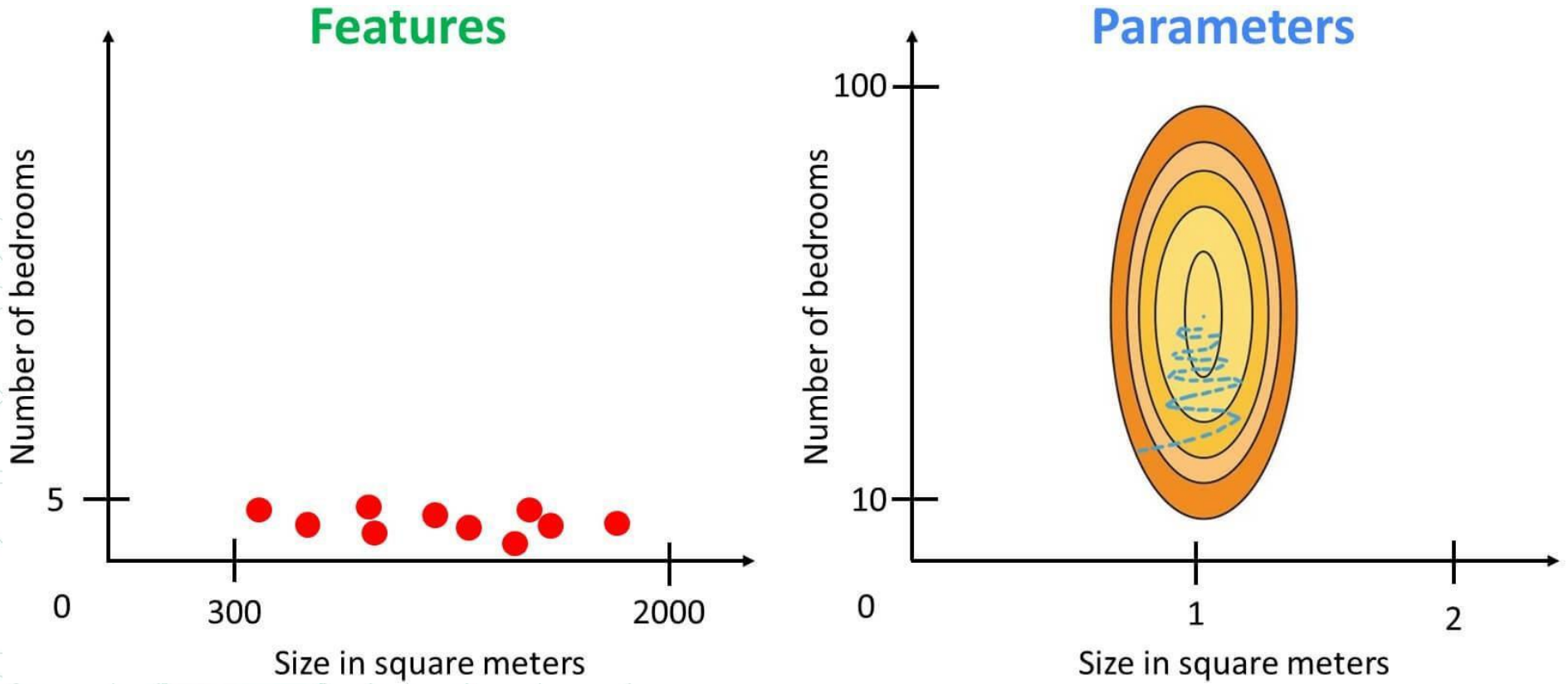


3D Plot

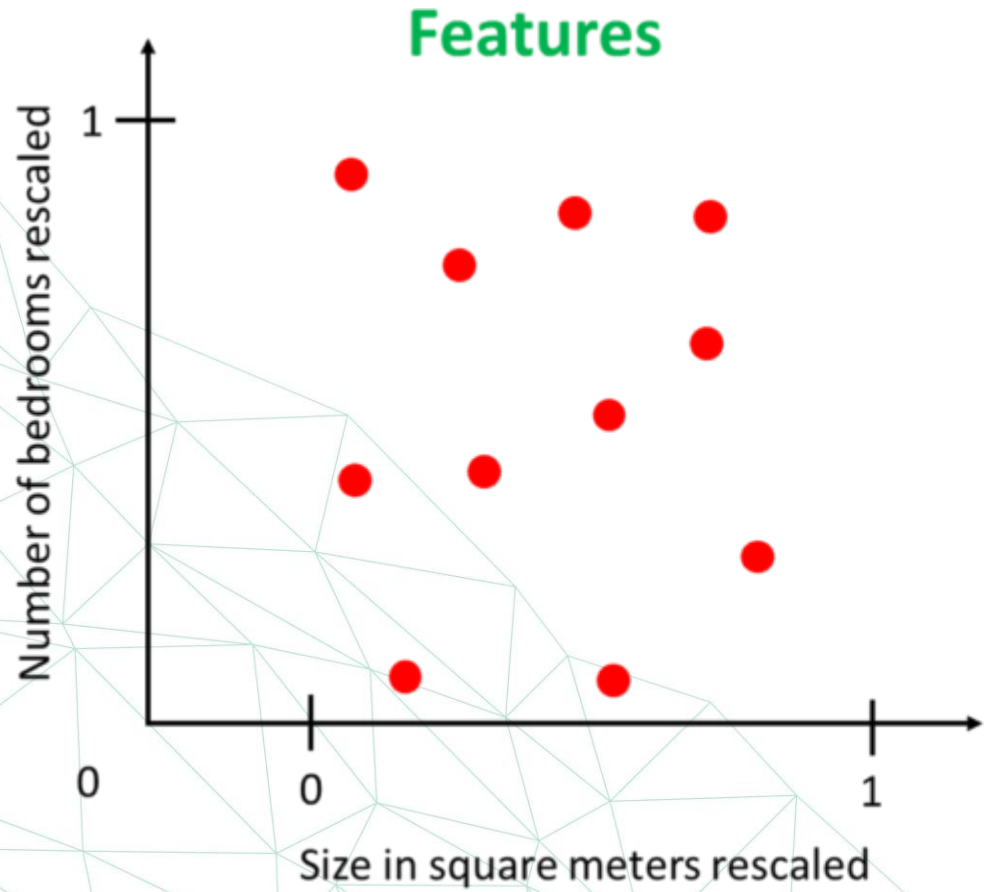


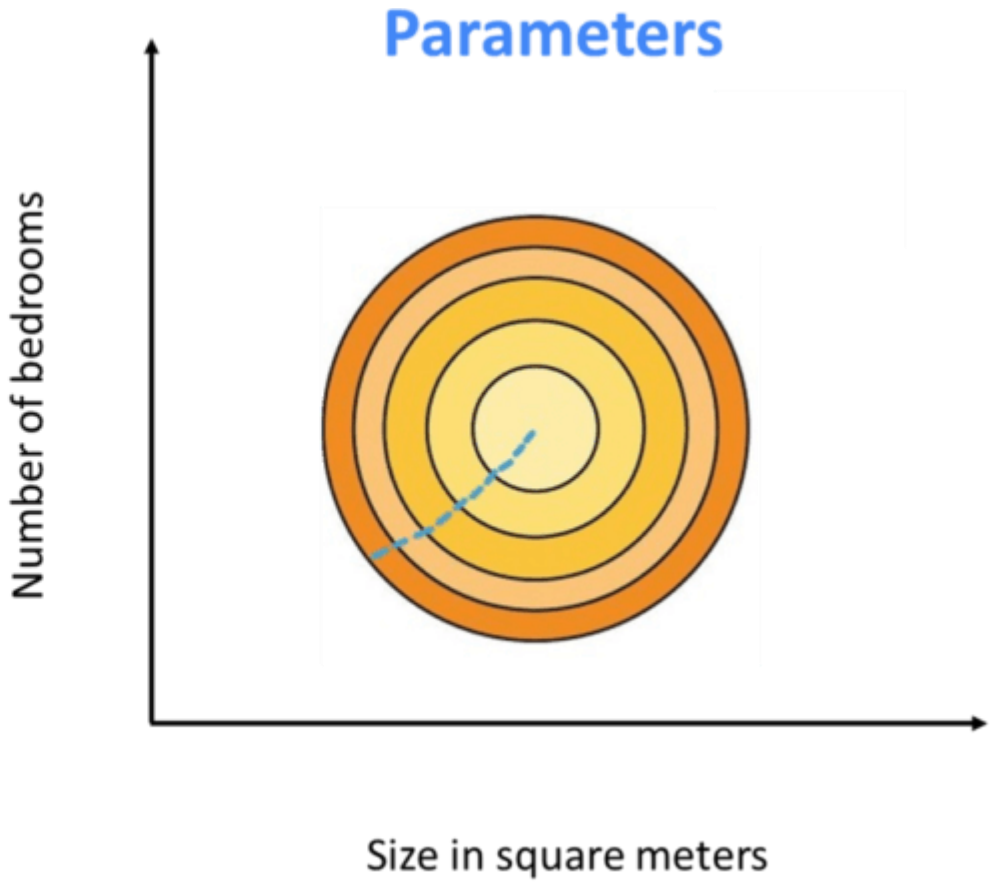
Contour Plot

The Role of Feature Scaling in Gradient Descent

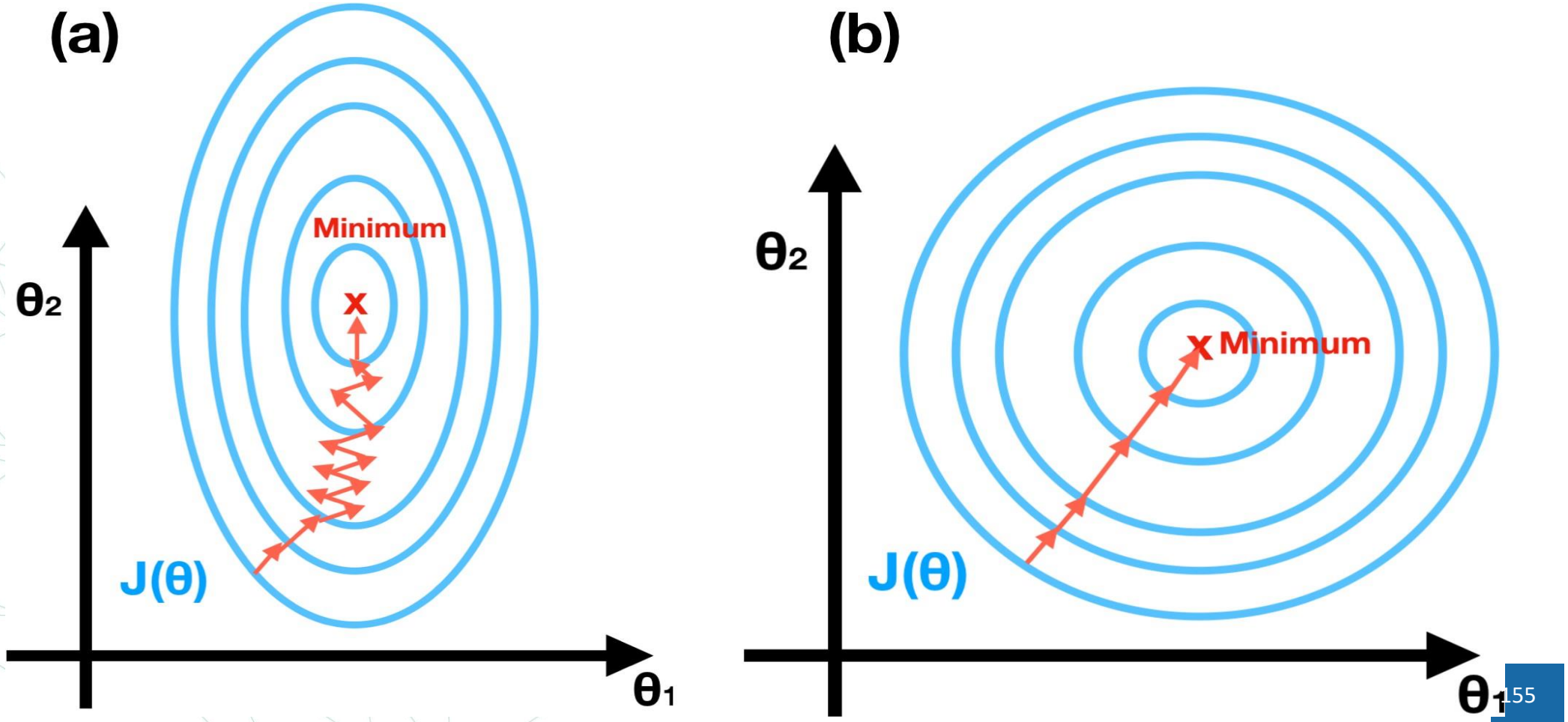


The Role of Feature Scaling in Gradient Descent





The Role of Feature Scaling in Gradient Descent



Simple Linear Regression

$$y = b_0 + b_1 x_1$$

Multiple Linear Regression

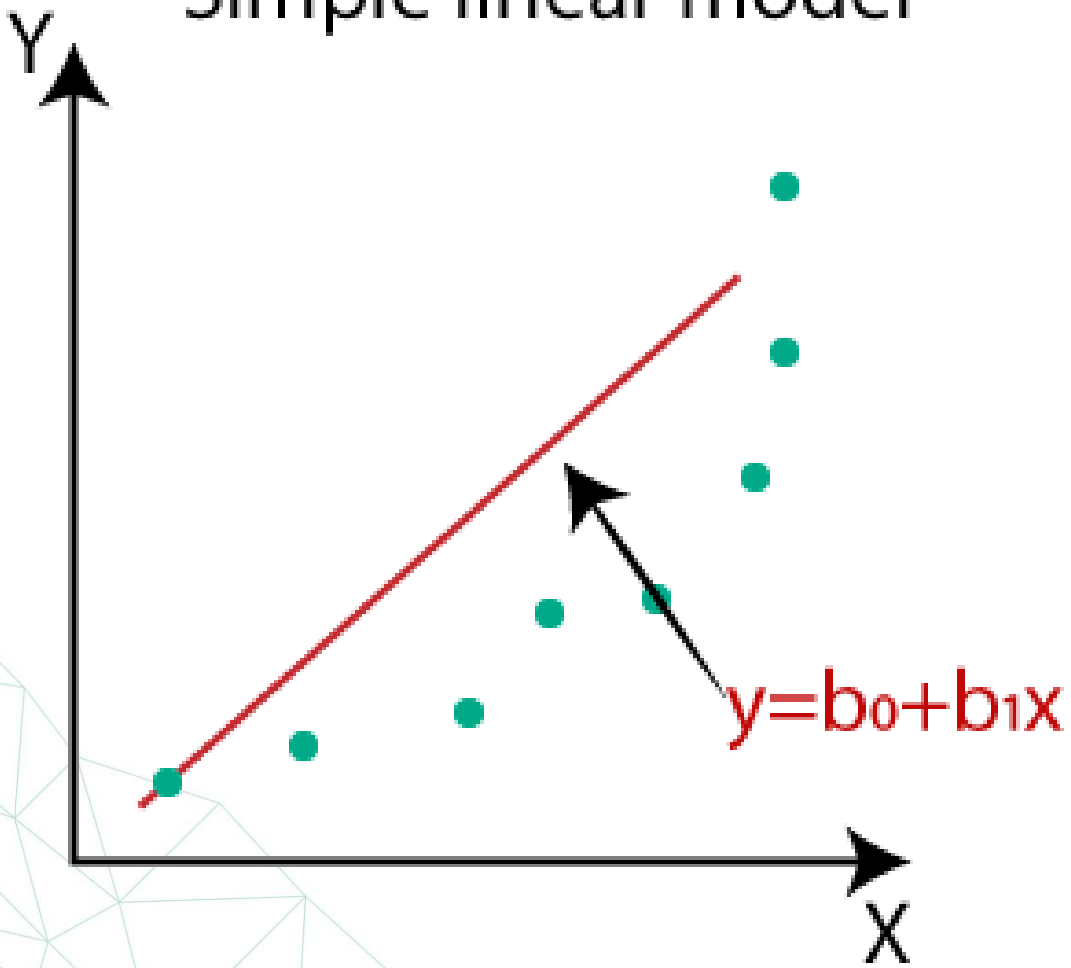
$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$$

Polynomial Linear Regression

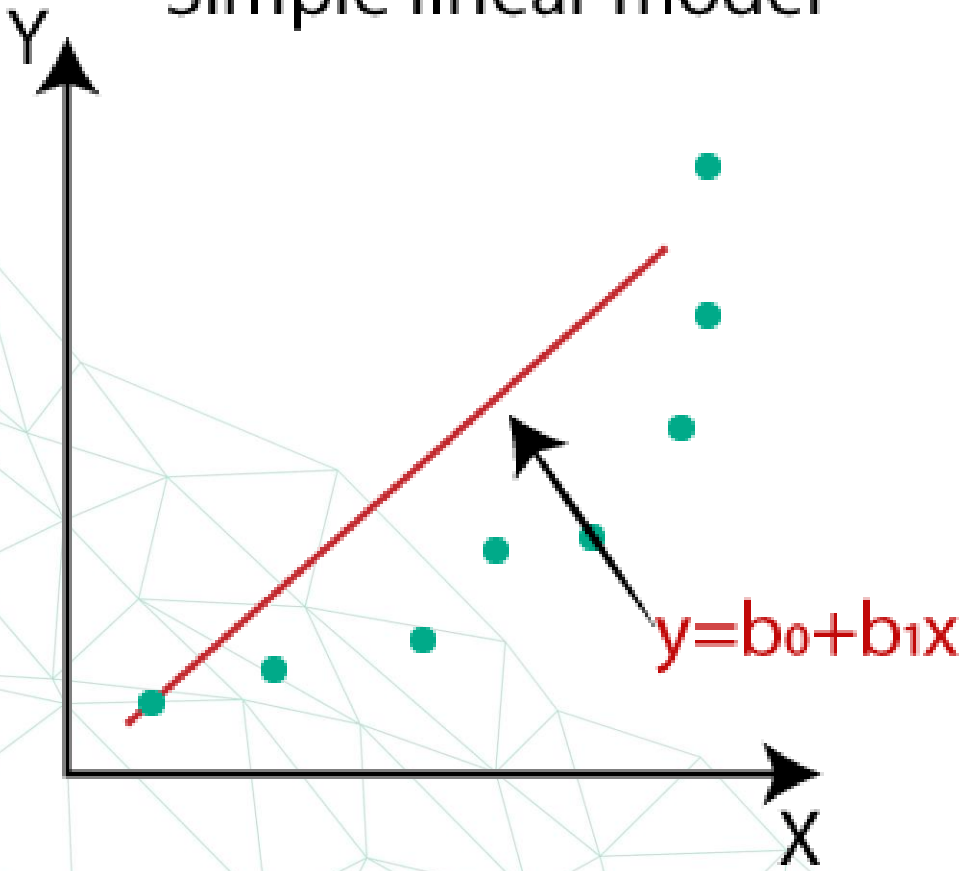
$$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$$

https://data36.com/polynomial-regression-python-scikit-learn/

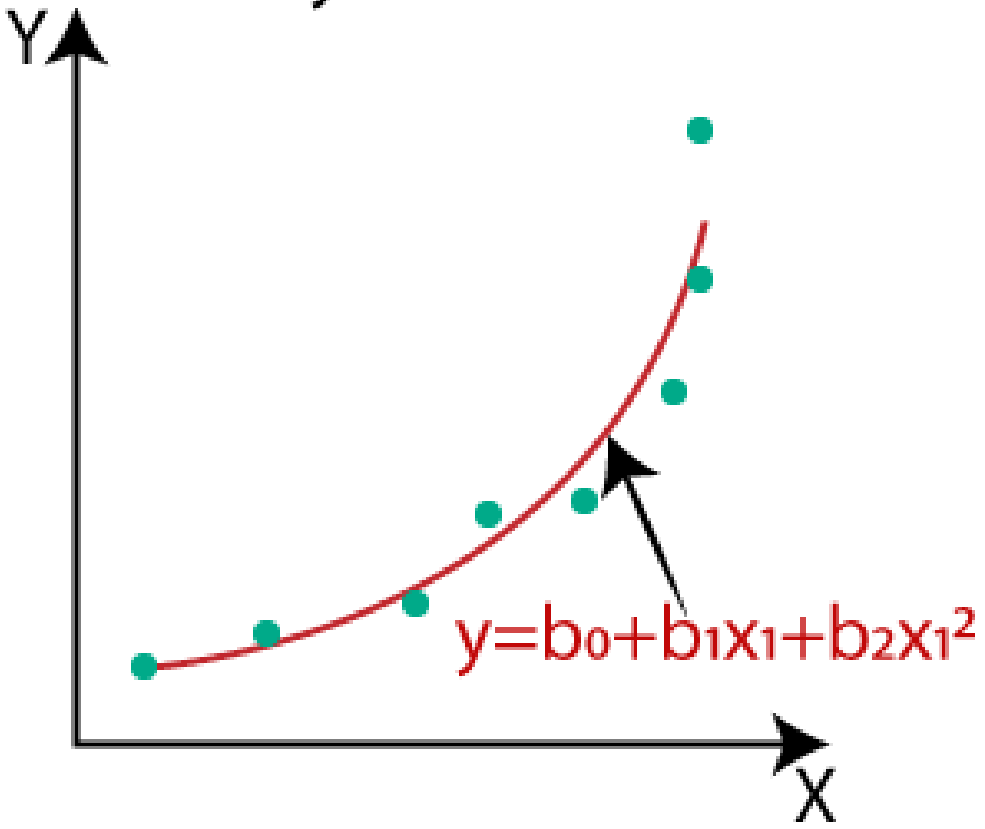
Simple linear model

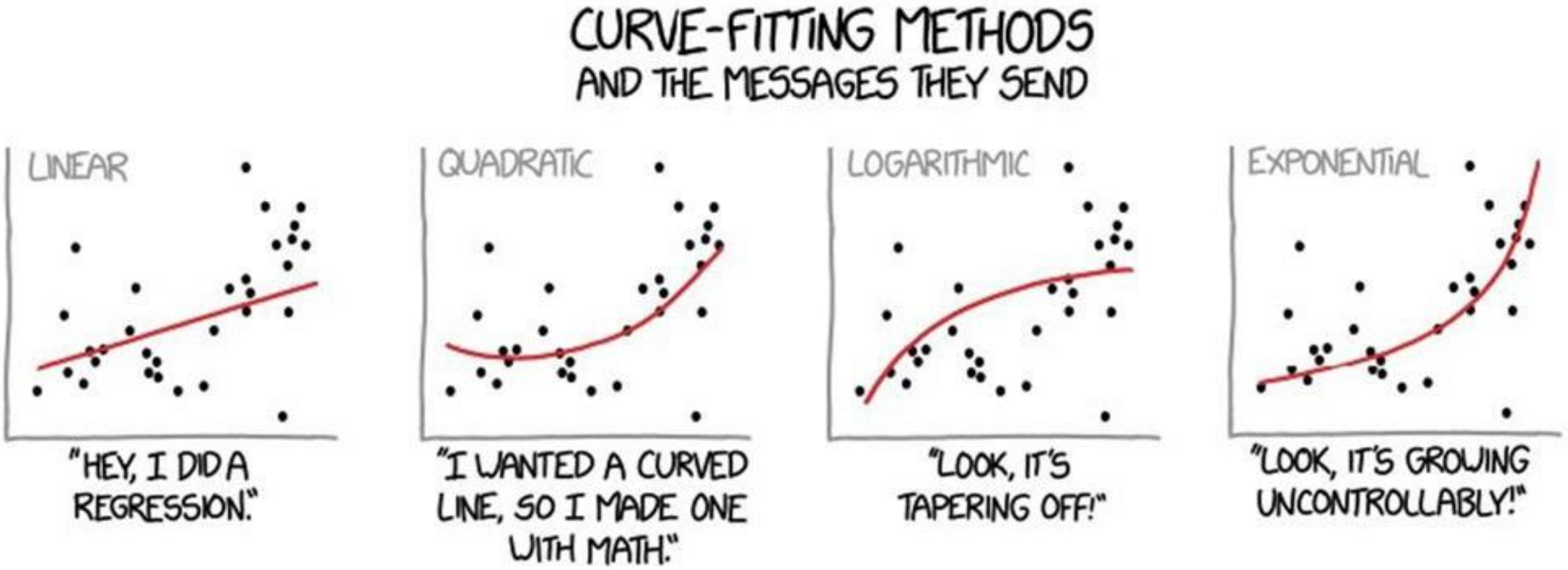


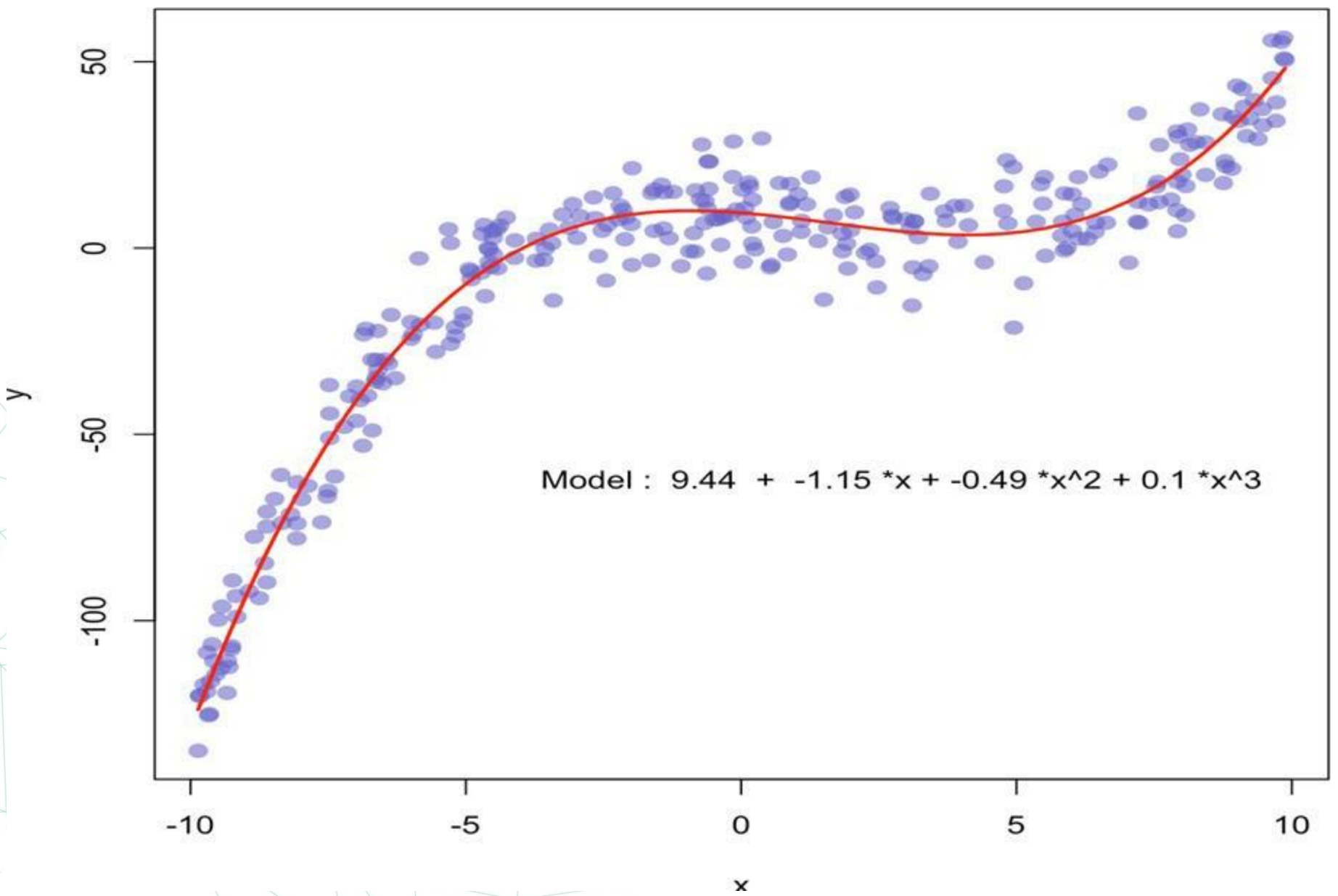
Simple linear model



Polynomial model




CURVE-FITTING METHODS
AND THE MESSAGES THEY SEND
LINEAR
"HEY, I DID A REGRESSION."
QUADRATIC
"I WANTED A CURVED LINE, SO I MADE ONE WITH MATH."
LOGARITHMIC
"LOOK, IT'S TAPERING OFF!"
EXPONENTIAL
"LOOK, IT'S GROWING UNCONTROLLABLY!"


Model : 9.44 + -1.15 *x + -0.49 *x^2 + 0.1 *x^3
y
x
50
0
-50
-100
-10
-5
0
5
10

1. What Are the Basic Assumption?

There are assumptions associated with a linear regression model:

1.Linearity: The relationship between X and the mean of Y is linear.
2.Homoscedasticity: The variance of residual is the same for any value of X.
3.Independence: Observations are independent of each other.
Normality: For any fixed value of X, Y is normally distributed.
log normal transformation, Box-Cox transformation, etc

2. Advantages

1.Linear regression performs exceptionally well for linearly separable data
2.Easy to implement and train the model
3.It can handle overfitting using dimensionlity reduction techniques and cross validation and regularization

3. Disadvantages

1.Sometimes Lot of Feature Engineering Is required
2.If the independent features are correlated it may affect performance
3.It is often quite prone to noise and overfitting

4. Whether Feature Scaling is required?

Yes

5. Impact of Missing Values?

It is sensitive to missing values

6. Impact of outliers?

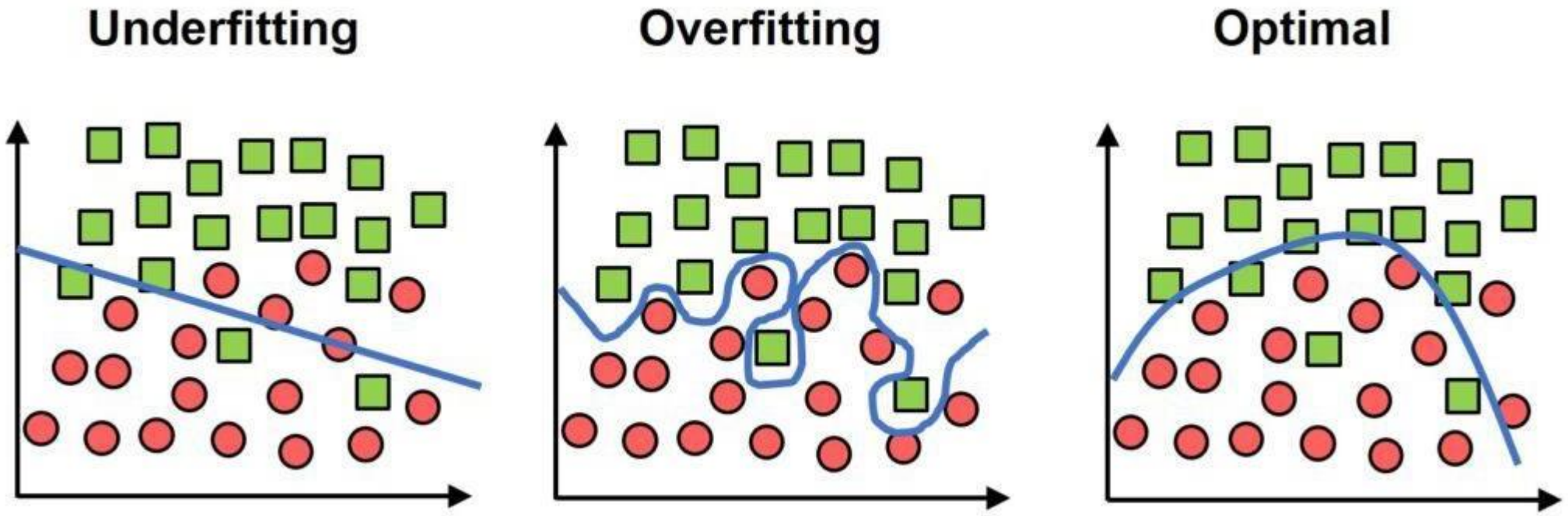
linear regression needs the relationship between the independent and dependent variables to be linear. It is also important to check for outliers since linear regression is sensitive to outlier effects.

Practical Lab

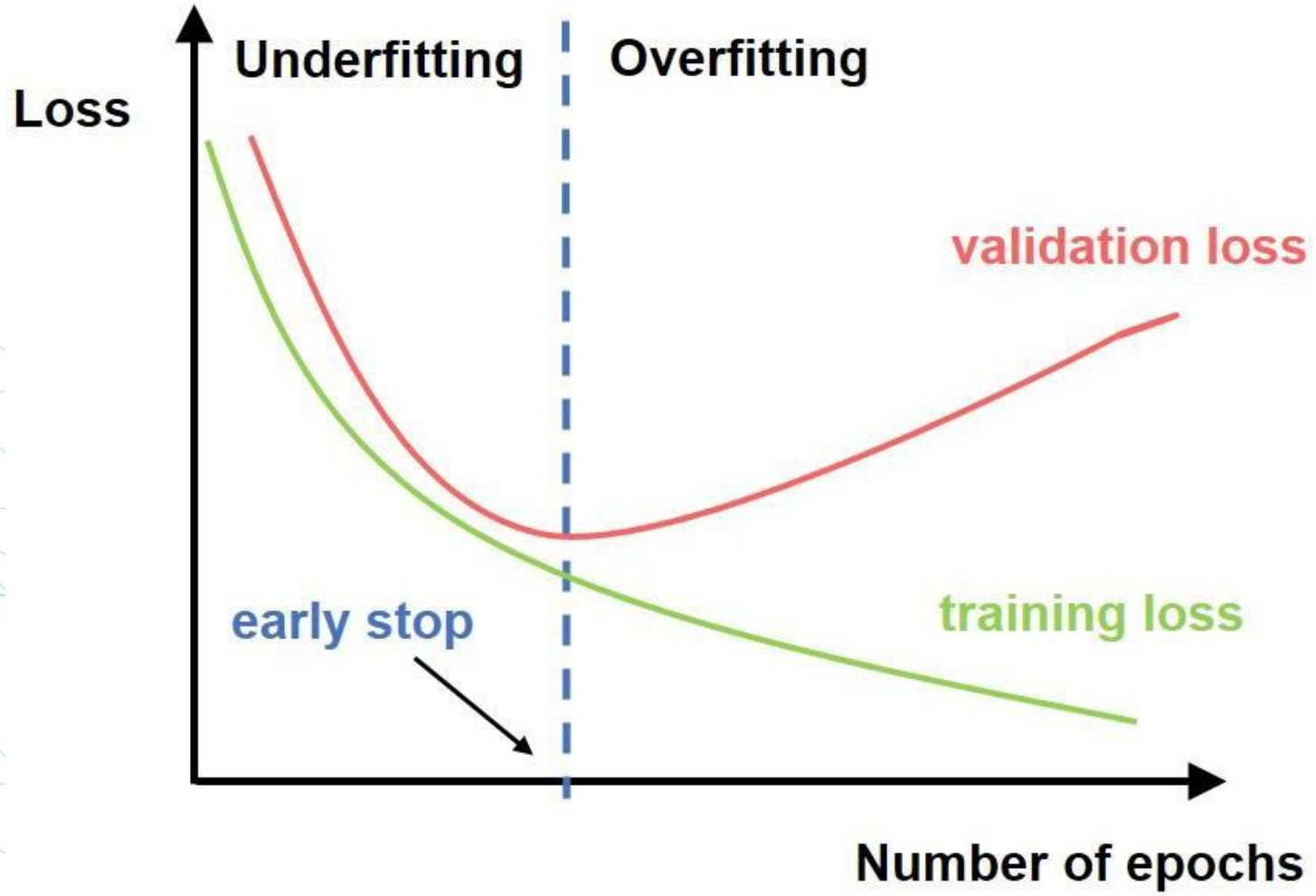
House Price Prediction Dataset

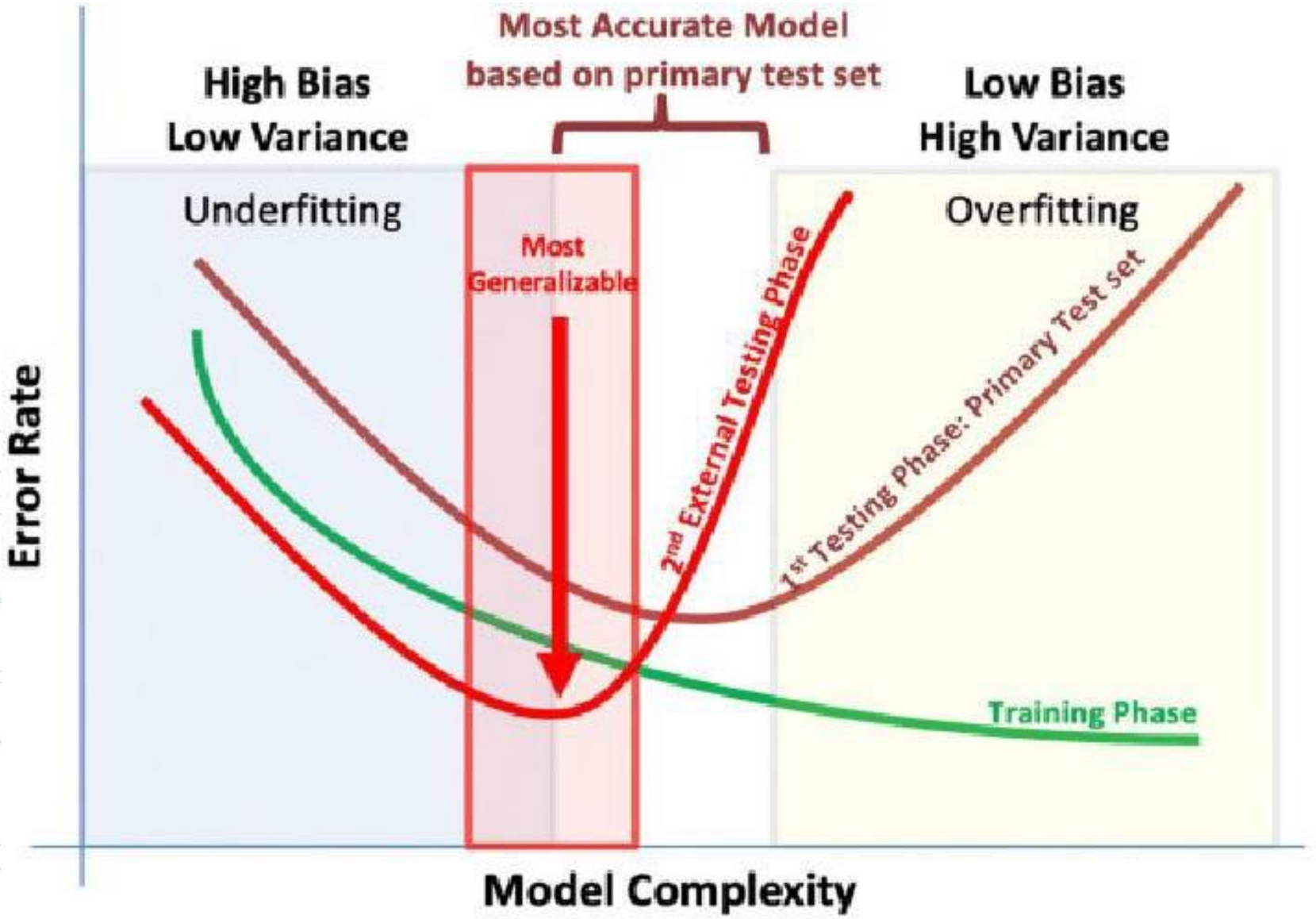
ML Concepts

Generalization, Overfitting, and Underfitting



Generalization, Overfitting, and Underfitting




Most Accurate Model
based on primary test set
High Bias
Low Variance
Low Bias
High Variance
Underfitting
Most
Generalizable
Overfitting
2nd External Testing Phase
1st Testing Phase: Primary Test set
Training Phase
Error Rate
Model Complexity

How to avoid Overfitting

- **Simplifying The Model**
- **Early Stopping**
- **Adding More Data To The Training Set**
- **Cross-validation**
- **Remove unnecessary features**
- **Regularization**

Regularisation Techniques

L1 Regularization

- This adds a penalty equal to the **L1 norm** of the weights vector(sum of the absolute value of the coefficients). It will shrink some parameters to **zero**.
- Hence some variables will not play any role in the model.
- L1 regression can be seen as a way to select features in a model.

$$\text{L1:} \quad R(\theta)=\|\theta\|_1=\sum_{i=1}^{n}|\theta_i|$$

$$LossFunction = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}-Y)^2+\lambda\sum_{i=1}^{N}|\theta_i|$$

L2 Regularization

- This adds a penalty equal to the L2 norm of the weights vector(sum of the squared values of the coefficients).
- It will force the parameters to be relatively small.

$$\text{L1:} \quad R(\theta)=\|\theta\|_1=\sum_{i=1}^{n}|\theta_i|$$

$$\text{L2:} \quad R(\theta)=\|\theta\|_2^2=\sum_{i=1}^{n}\theta_i^2$$

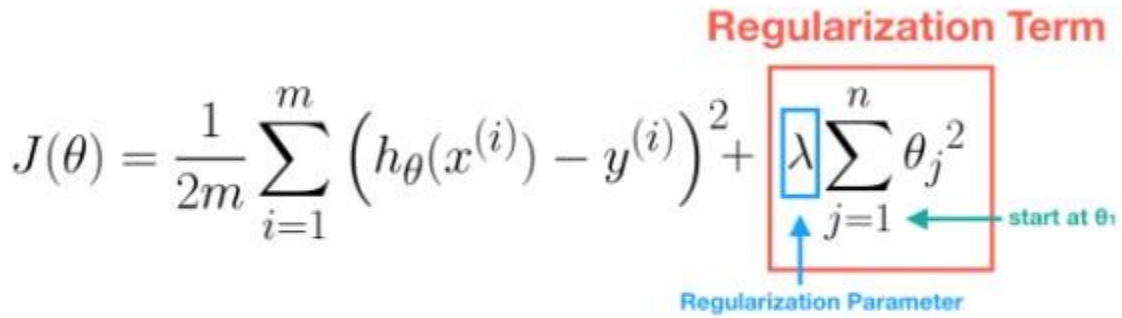
$$LossFunction = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}-Y)^2+\lambda\sum_{i=1}^{N}\theta_i^2$$

$$J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} (\hat{y}_i - y_i)^2$$

where

- **hθ(x(i))** is the predicted value of some datapoint x(i)
- **y(i)** is original

The penalized cost function looks like this

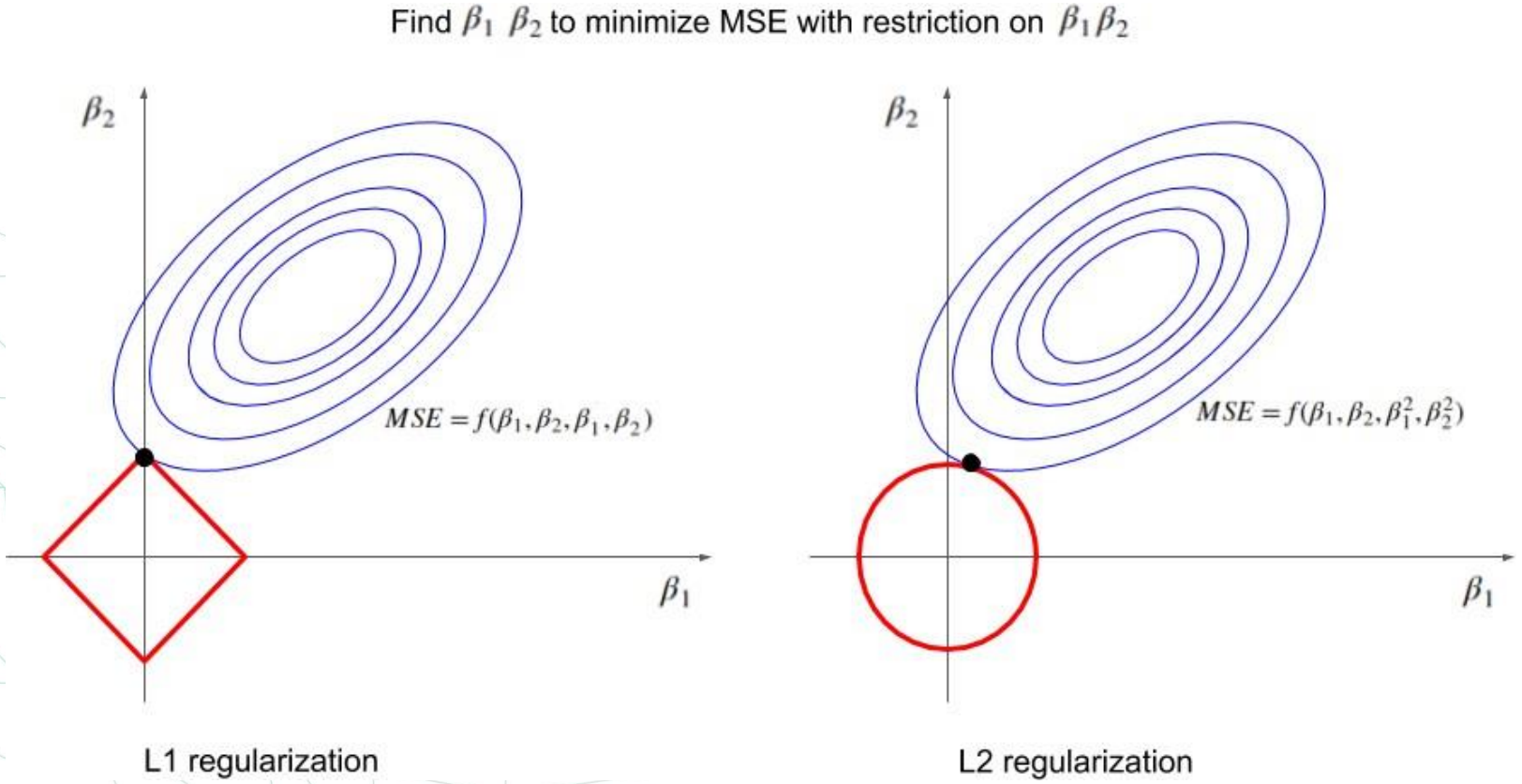


$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} {\theta_j}^2$$

where-

- **λ** is the tuning parameter that decides how much we want to penalize the flexibility of our model. It can be tuned using cross-validation.

***The difference is that**: while shrinking the quota, **L1 tends to cut off some factors by turning their coefficients to zero**, while L2 tend **s to shrinking these coefficients to a tiny number (none zero)**, keep some of their influence on Y.*



L1 regularization

L2 regularization

Comparison of L1 and L2 regularization	
L1 regularization	**L2 regularization**
Sum of absolute value of weights	Sum of square of weights
Sparse solution	Non-sparse solution
Multiple solutions	One solution
Built-in feature selection	No feature selection
Robust to outliers	Not robust to outliers (due to the square term)

By this I mean the number of solutions to arrive at one point. L1 regularization uses Manhattan distances to arrive at a single point, so there are many routes that can be taken to arrive at a point. L2 regularization uses Euclidean distances, which will tell you the fastest way to get to a point. This means the L2 norm only has 1 possible solution.

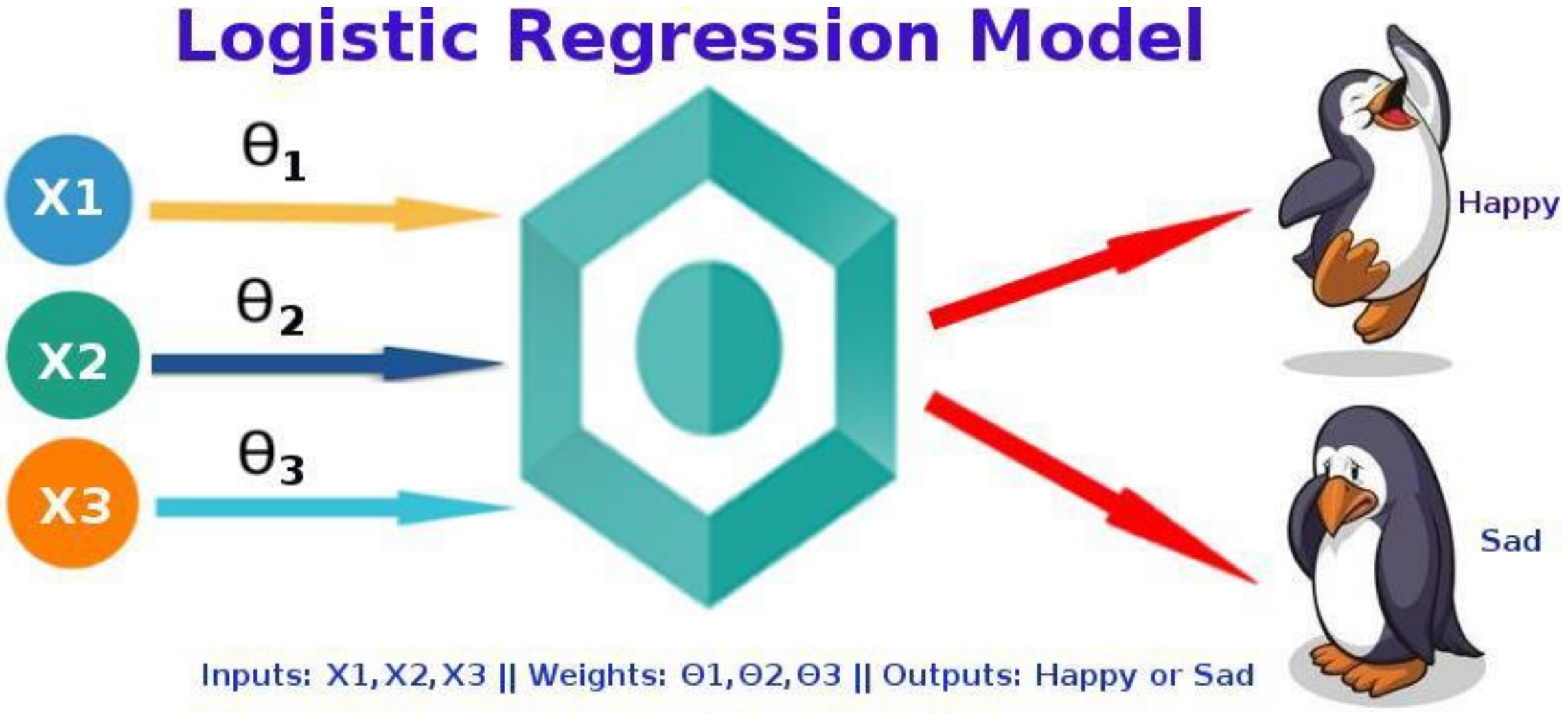
Which solution is less Computationally expensive? L2

- Regularization Lab

Q&A

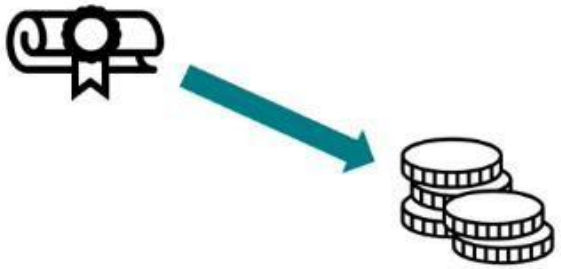
Questions and answers

Logistic Regression

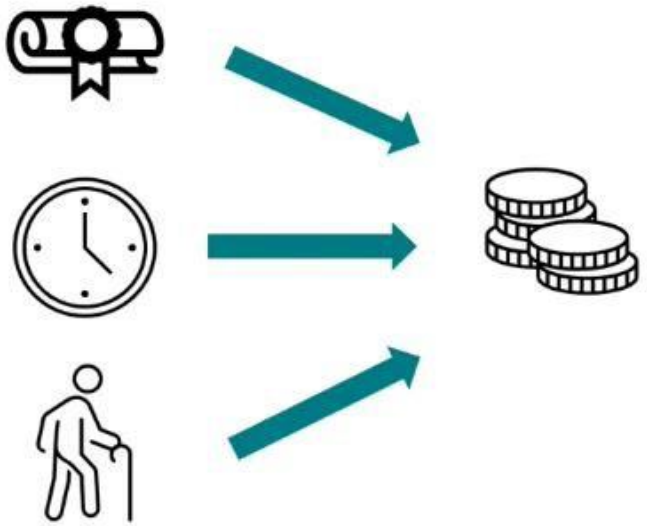

Logistic Regression Model
X1
X2
X3
θ1
θ2
θ3
Happy
Sad
Inputs: X1, X2, X3 || Weights: Θ1, Θ2, Θ3 || Outputs: Happy or Sad
@dataaspirant.com

What is Logistic Regression?

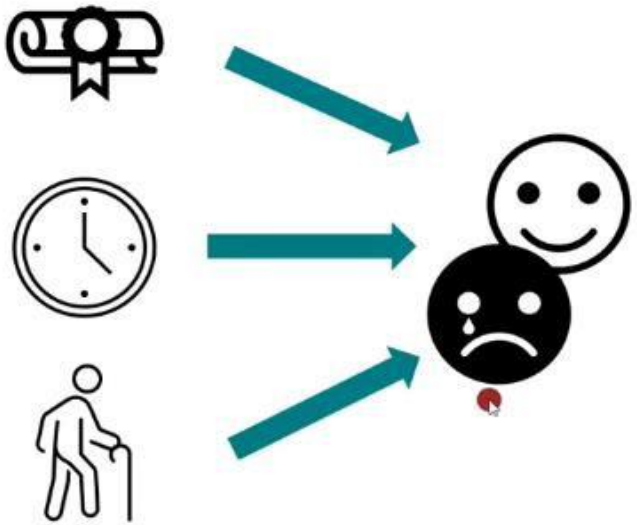
Simple linear regression



Multiple linear regression

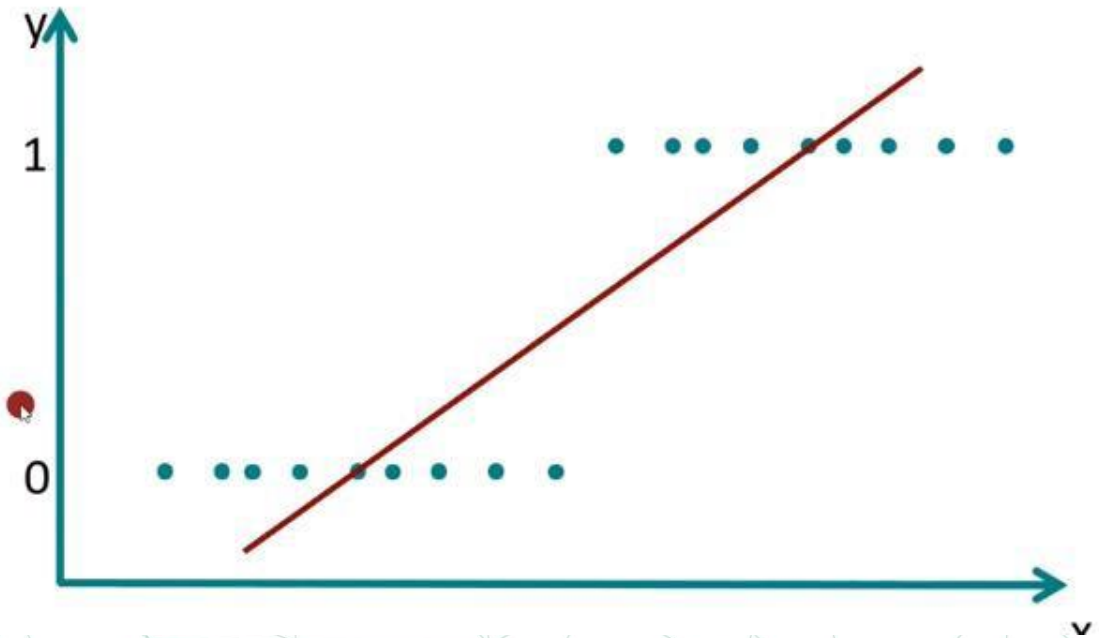


Logistic regression



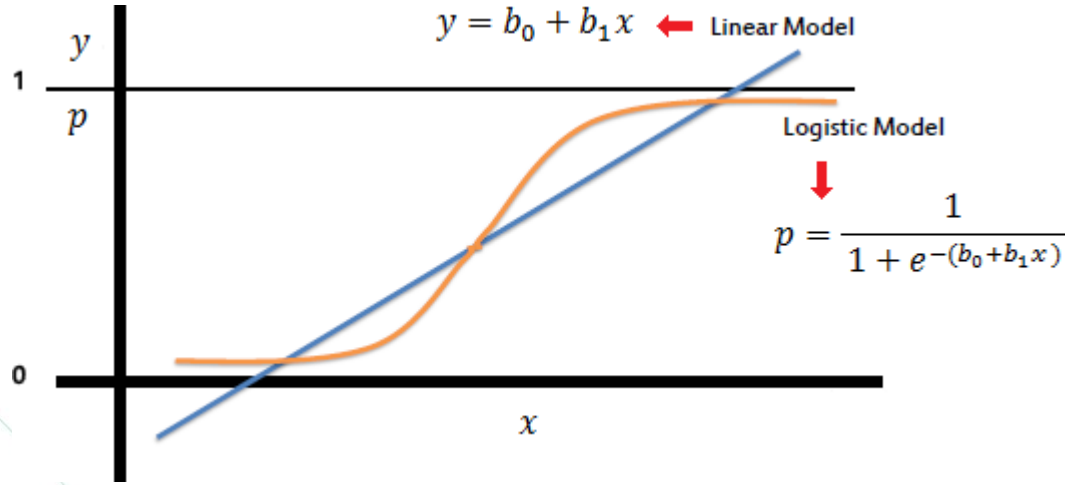
Why not just linear regression?

$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$



- The graph shows that values between **plus and minus infinity** can now occur.
- The goal of logistic regression is to estimate the **probability of occurrence**, not the value of the variable itself.
- The range of values for the prediction is restricted to the range between 0 and 1.
- To ensure that only values between 0 and 1 are possible, the logistic function f is used.

Logistic Regression



A linear regression will predict values outside the acceptable range (e.g. predicting probabilities outside the range 0 to 1)

Logistic Regression

In the logistic regression the constant (b_0) moves the curve left and right and the slope (b_1) defines the steepness of the curve. By simple transformation, the logistic regression equation can be written in terms of an odds ratio.

$$\frac{p}{1-p} = \exp(b_0 + b_1 x)$$

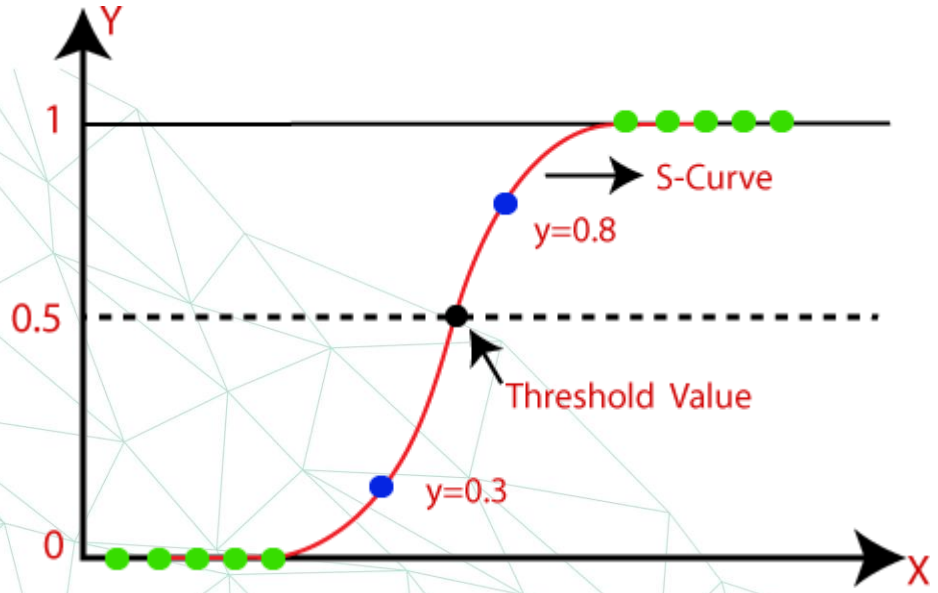
Finally, taking the natural log of both sides, we can write the equation in terms of log-odds (logit) which is a linear function of the predictors. The coefficient (b_1) is the amount the logit (log-odds) changes with a one unit change in x.

$$ln\left(\frac{p}{1-p}\right) = b_0 + b_1 x$$

As mentioned before, logistic regression can handle any number of numerical and/or categorical variables.

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p)}}$$

Logistic Regression



$$y' = b_0 + b_1x_1 + b_2x_2$$

$$if\ p < 0.5\ then\ \text{N}\ else\ \text{Y}$$

BUSAGE	DAYSDELQ	DEFAULT	y'	p	Prediction
87	2	N	-4.811	0.008	N
89	2	N	-4.795	0.008	N
100	26	Y	-2.261	0.094	N
275	54	Y	1.983	0.879	Y
42	57	Y	0.437	0.608	Y
88	53	N	0.395	0.597	Y

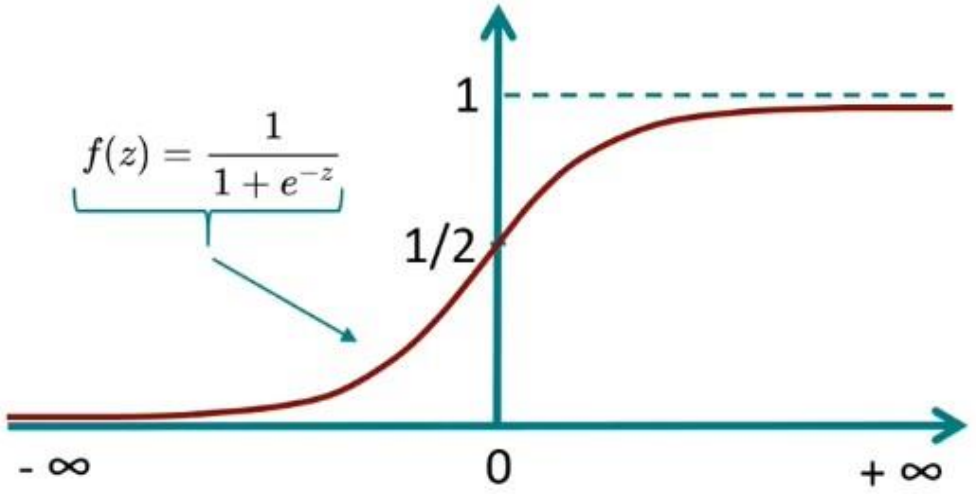
$$y' = -5.706 + 0.008 \times 87 + 0.102 \times 2 = -4.811$$

$$p = \frac{1}{1 + e^{-(b_0 + b_1x_1 + b_2x_2)}} = \frac{1}{1 + e^{-y'}}$$

$$p = \frac{1}{1 + e^{4.811}} = 0.008 \xrightarrow{<0.5} \text{N}$$

What is Logistic Regression?

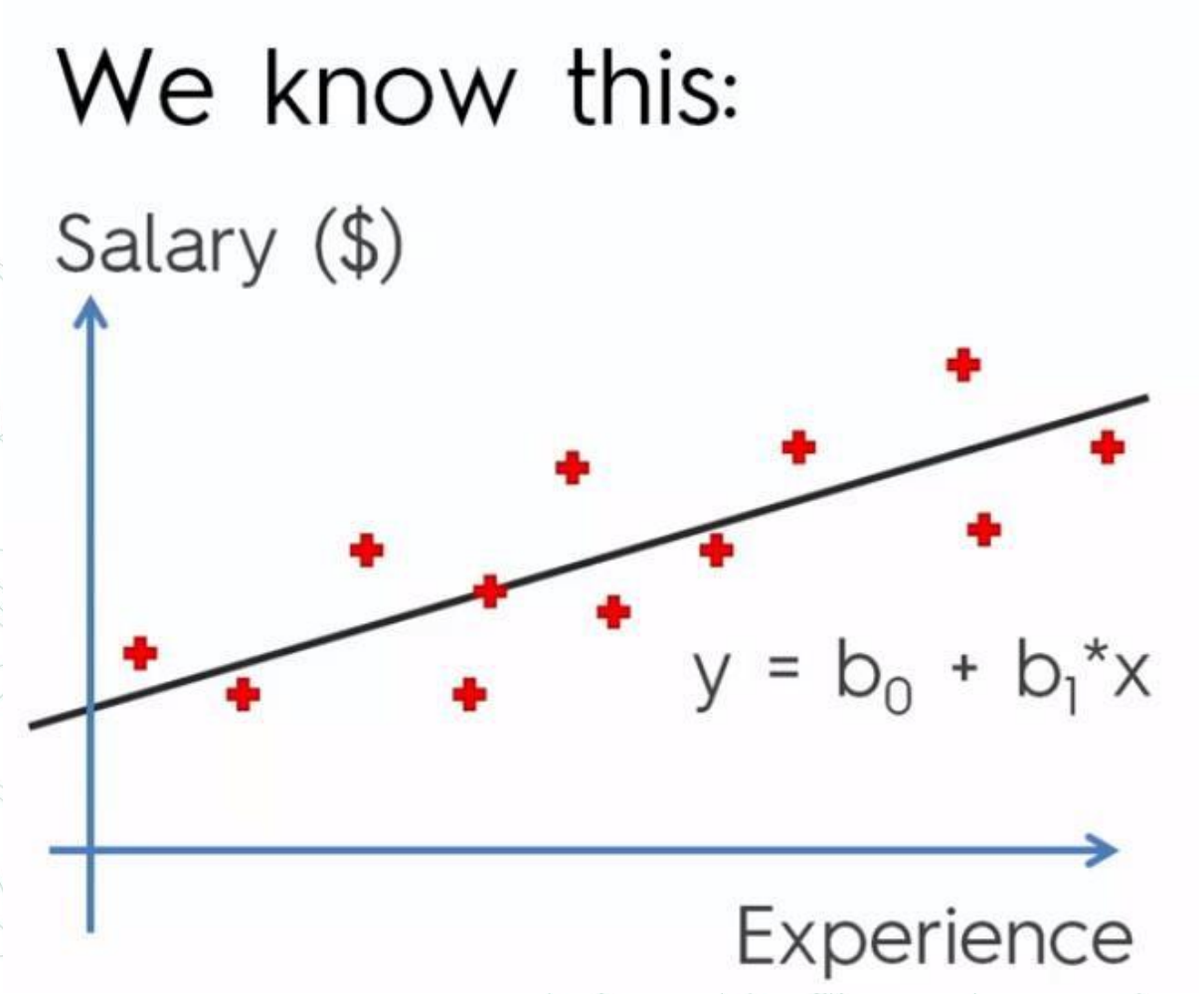
The logistic model is based on the logistic function.
The important thing about the logistic function is, that only values **between 0 and 1** are possible.

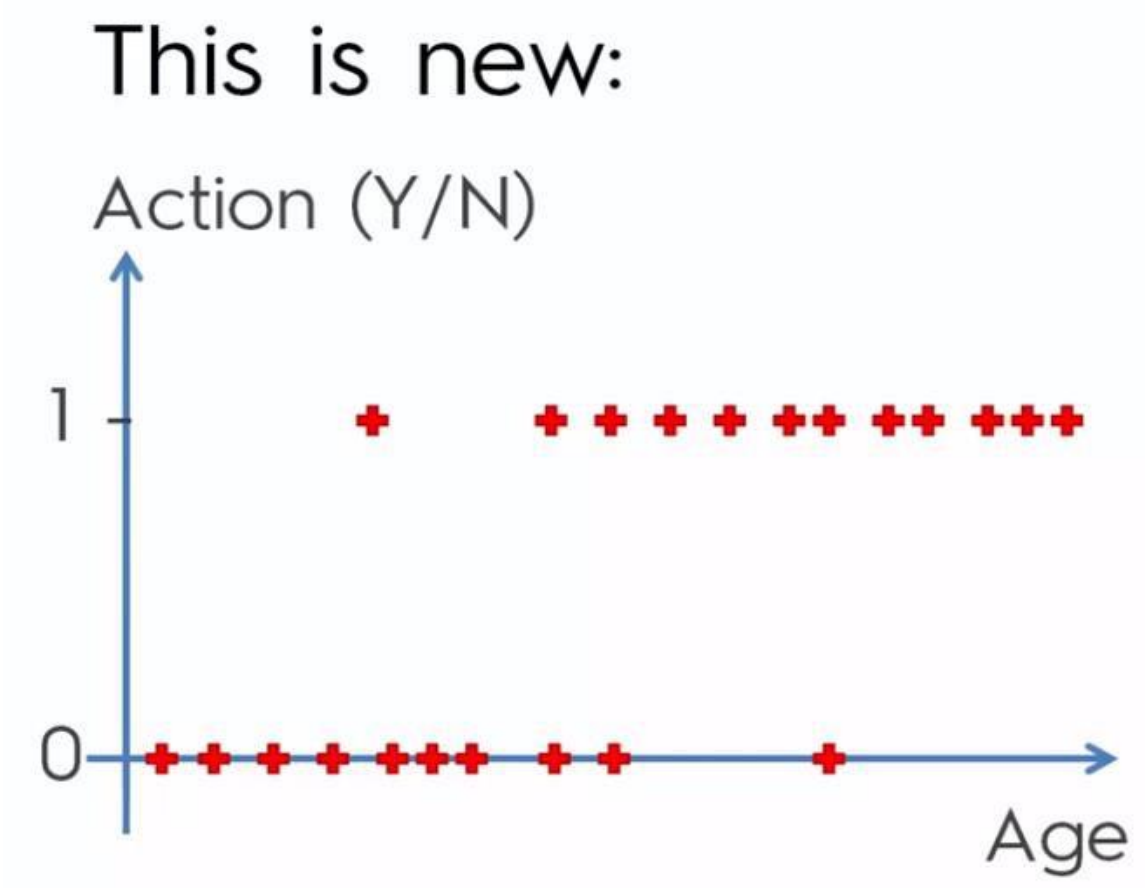


$$f(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(b_1 \cdot x_1 + \ldots + b_k \cdot x_k + a)}}$$

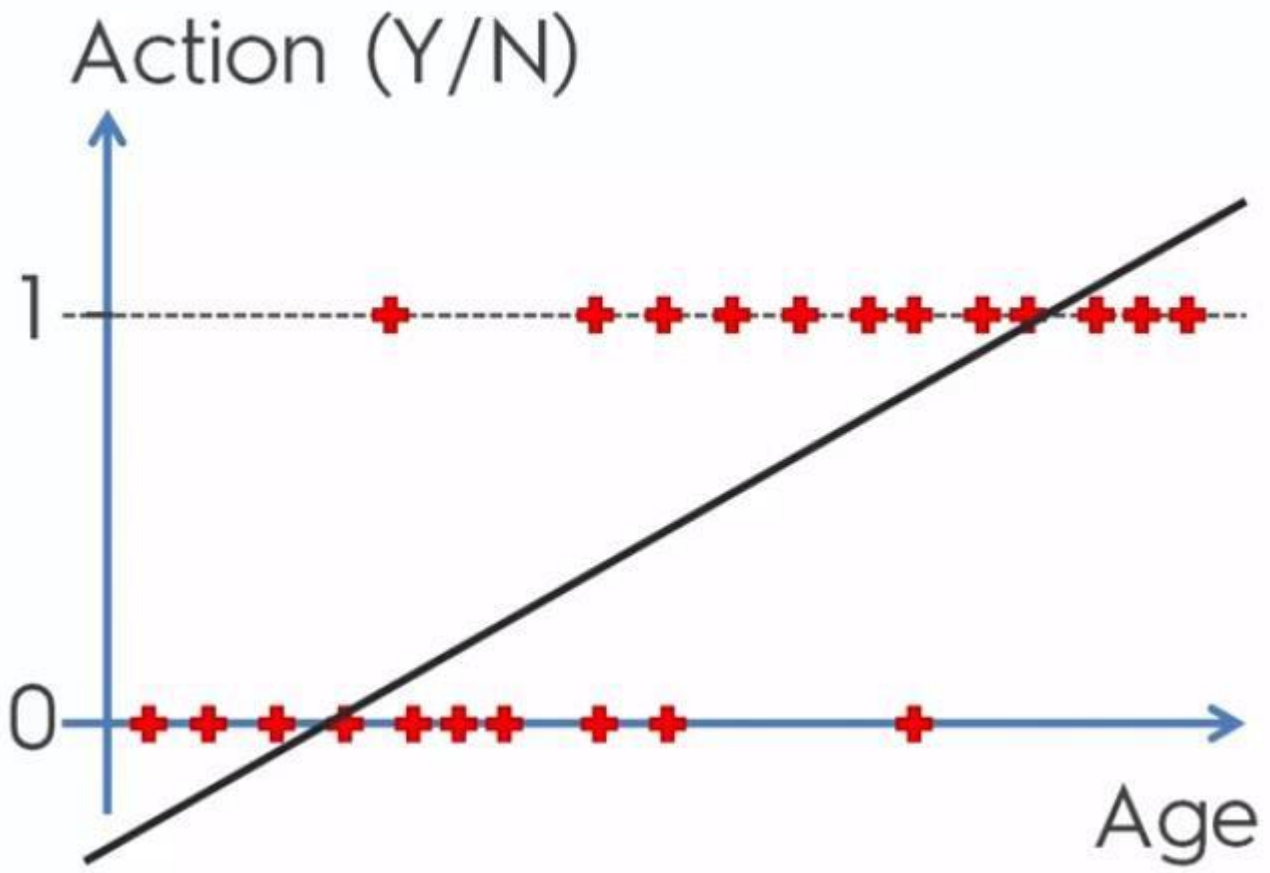
$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$

Logistic Regression

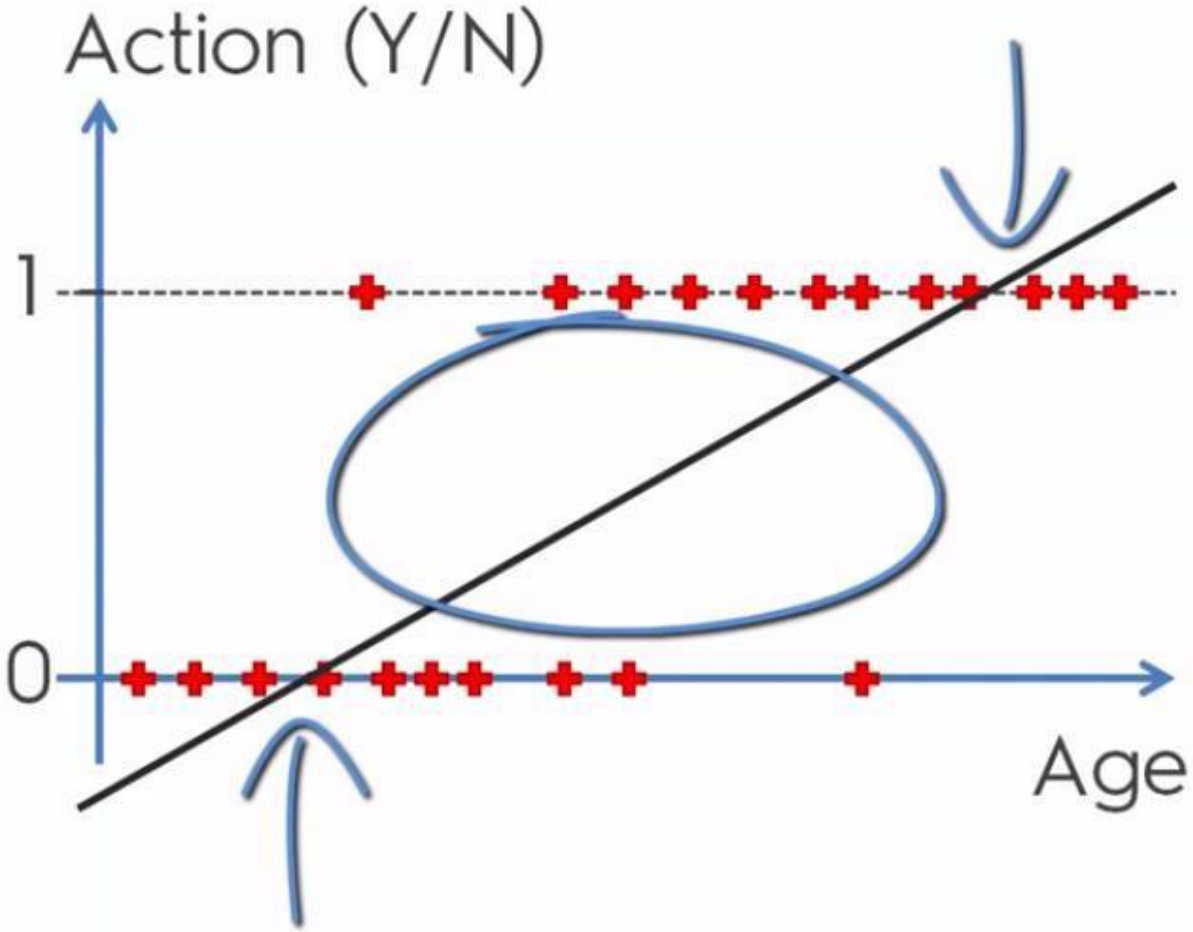




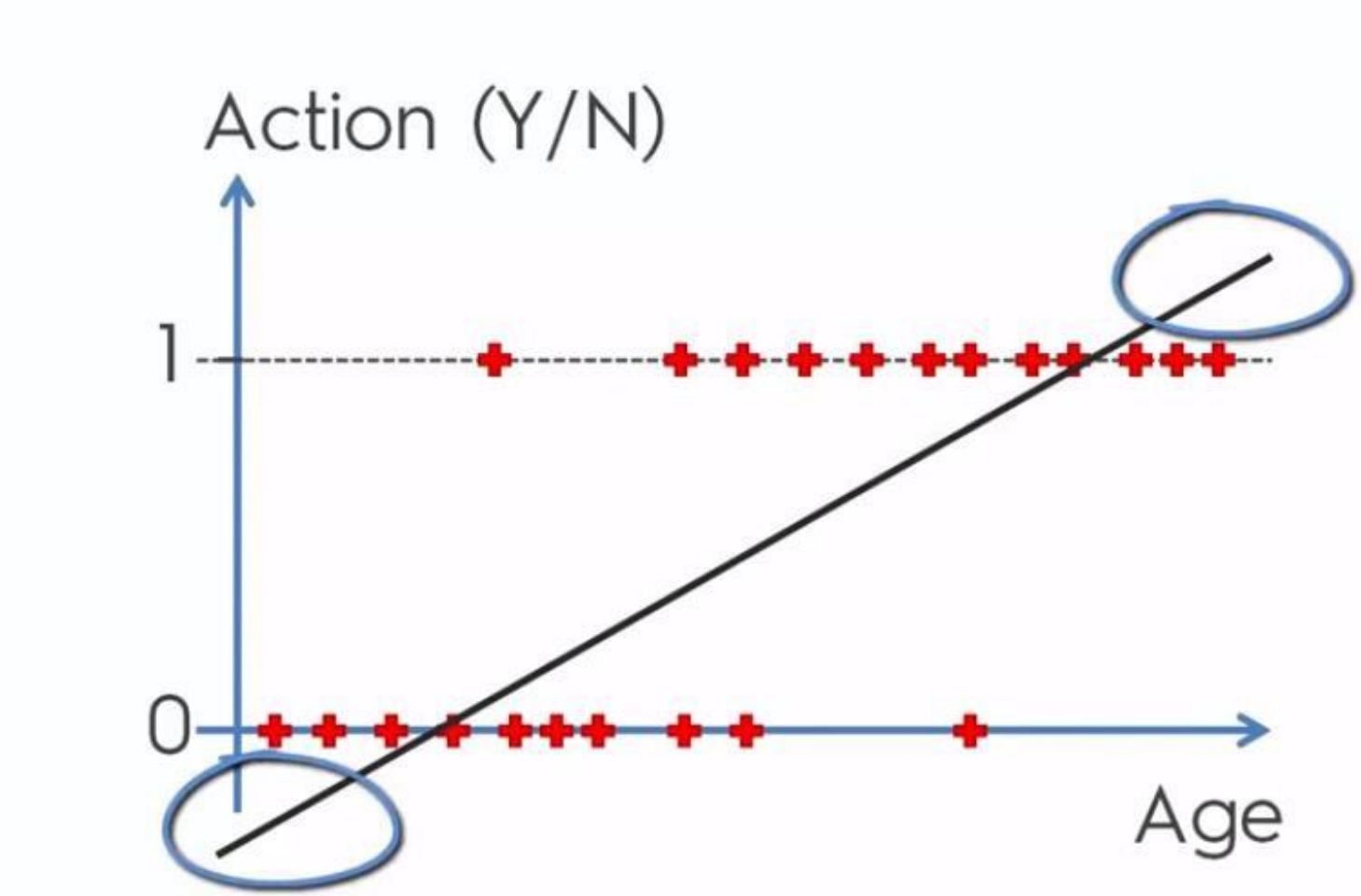
Logistic Regression



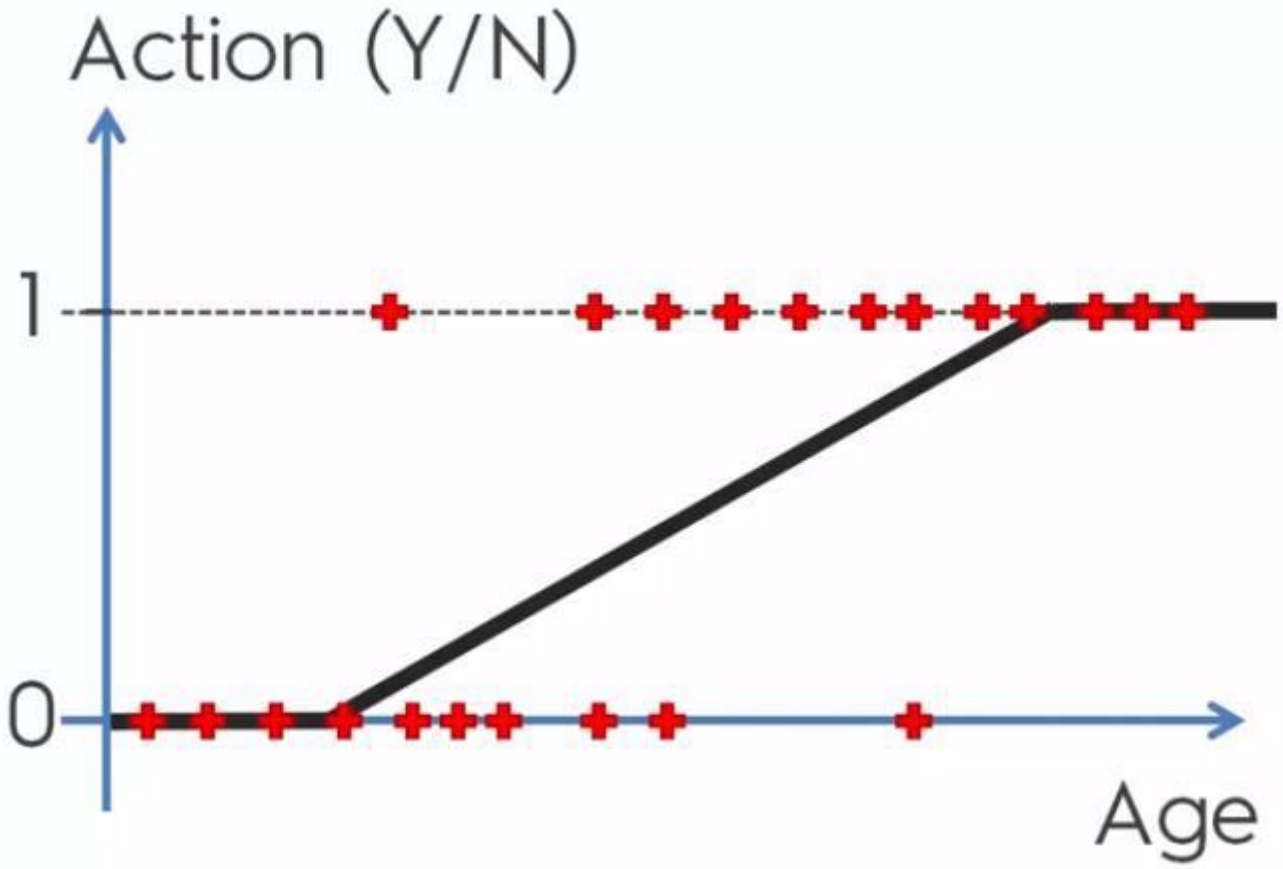
Logistic Regression



Logistic Regression



Logistic Regression

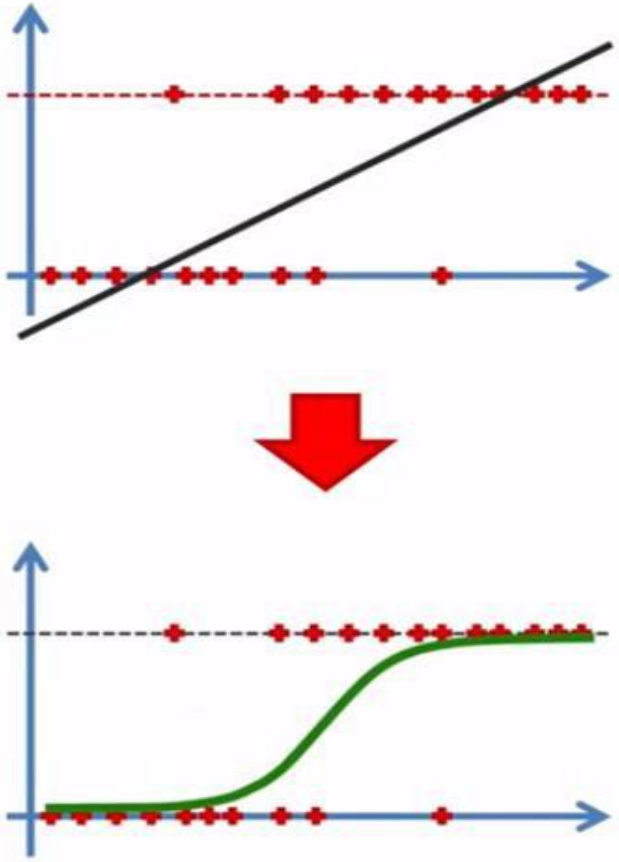


Logistic Regression

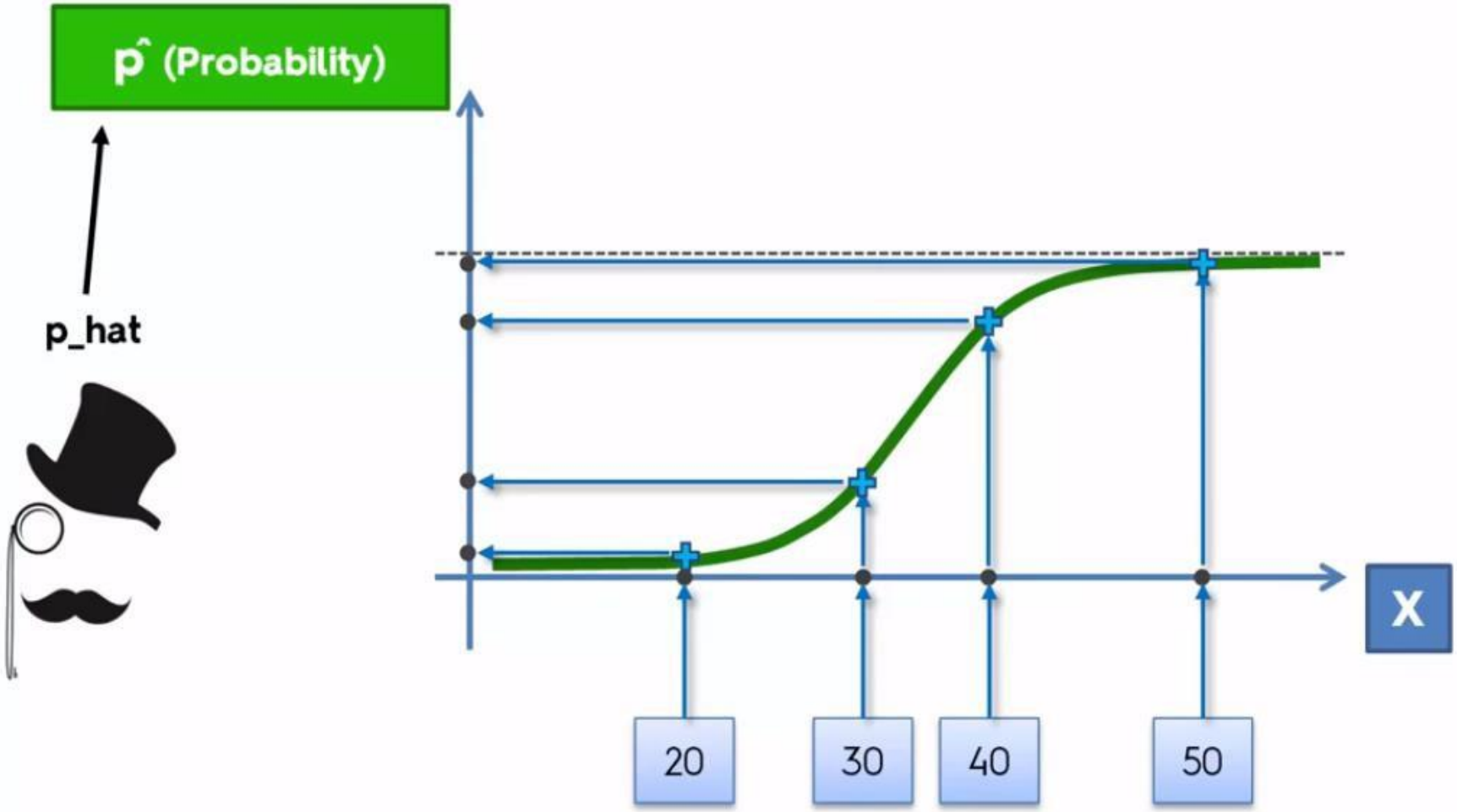
$$y = b_0 + b_1 * x$$

Sigmoid Function

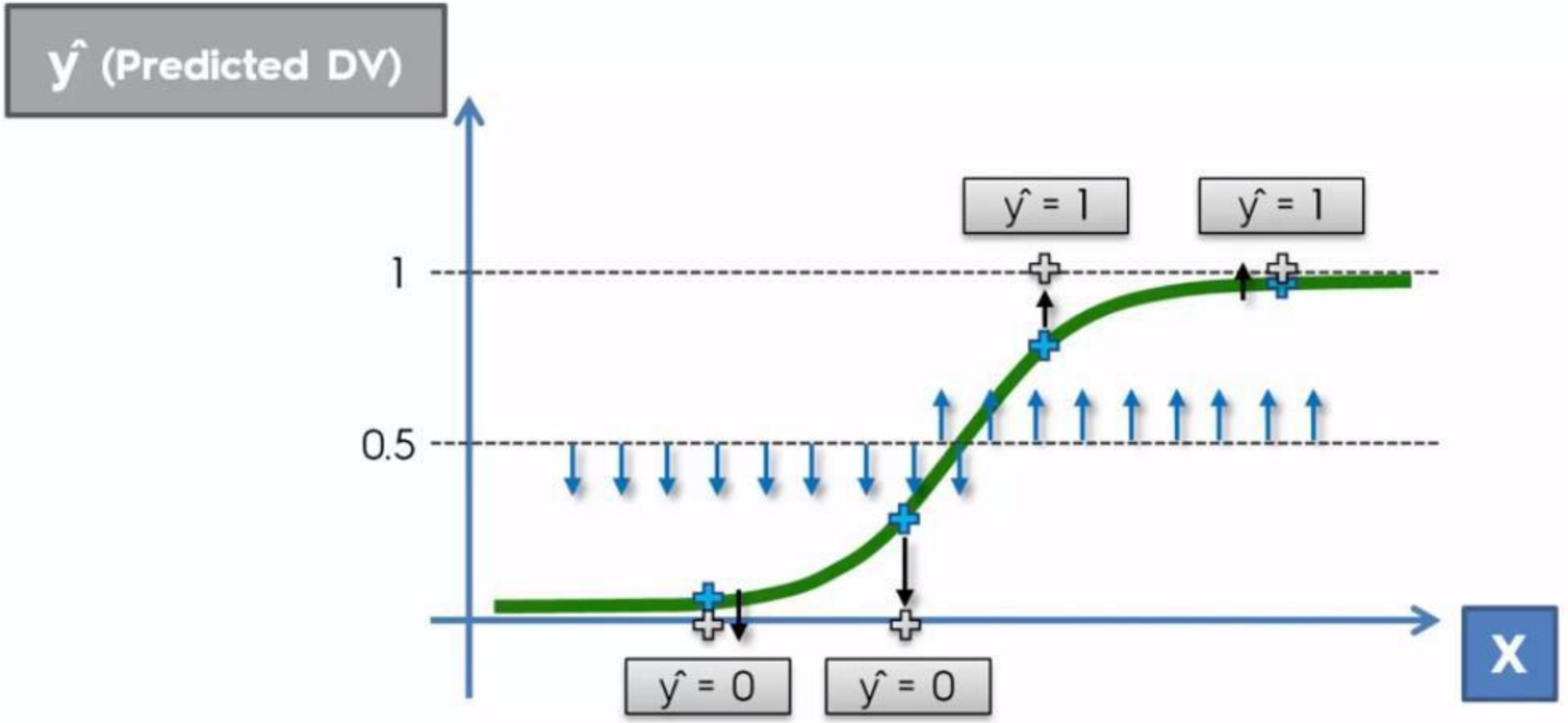
$$p = \frac{1}{1 + e^{-y}}$$



Logistic Regression

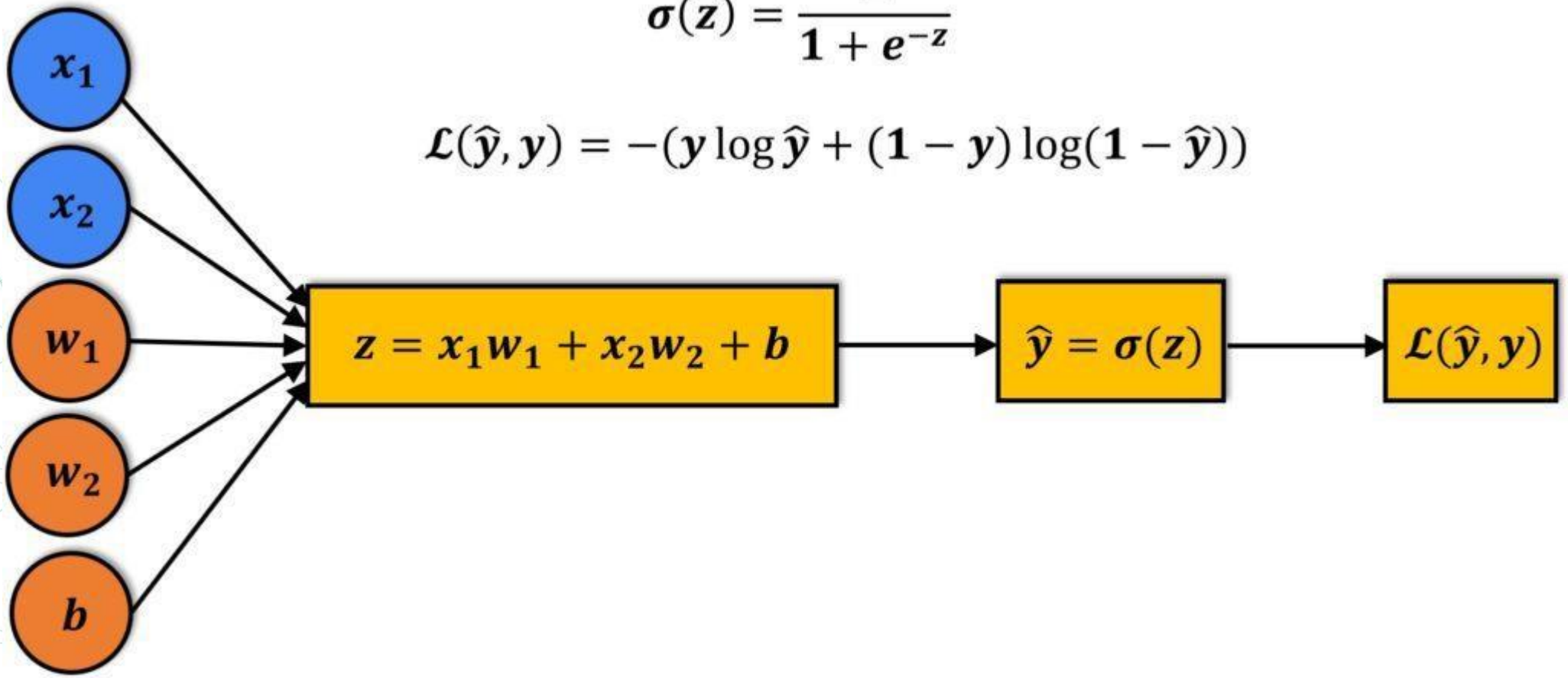


Logistic Regression

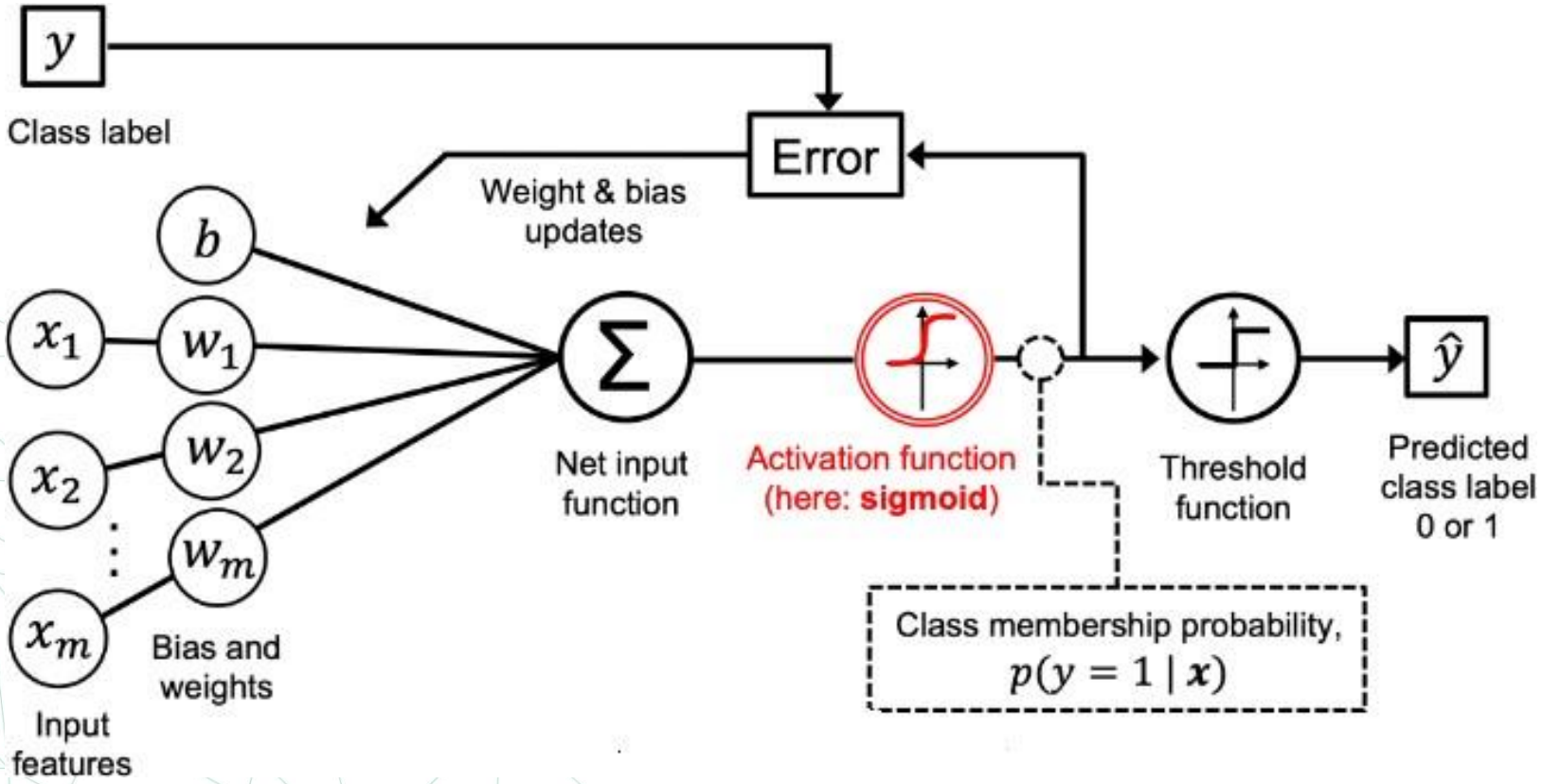


$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\mathcal{L}(\hat{y}, y) = -(y \log \hat{y} + (1 - y) \log(1 - \hat{y}))$$

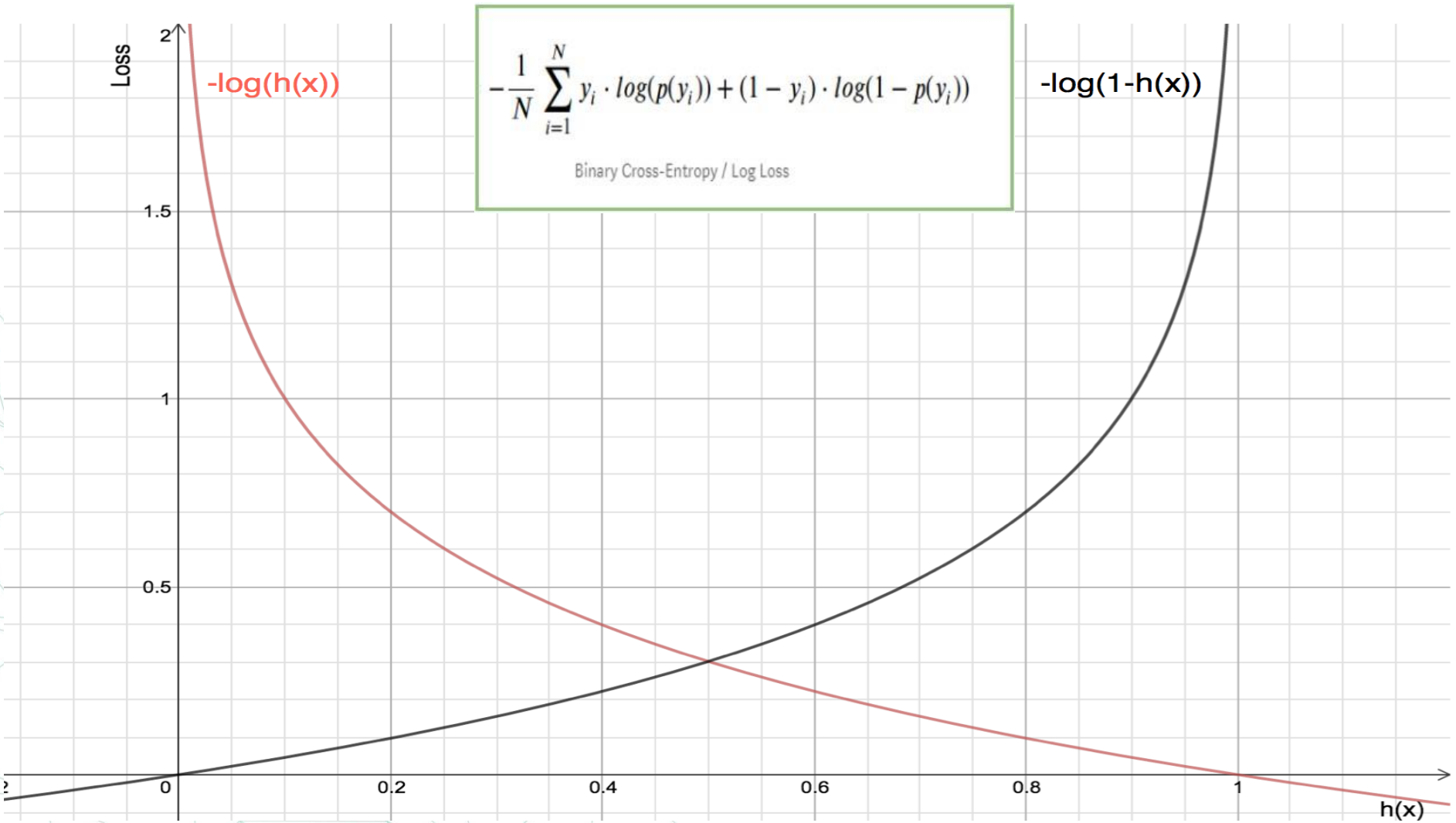


Logistic Regression

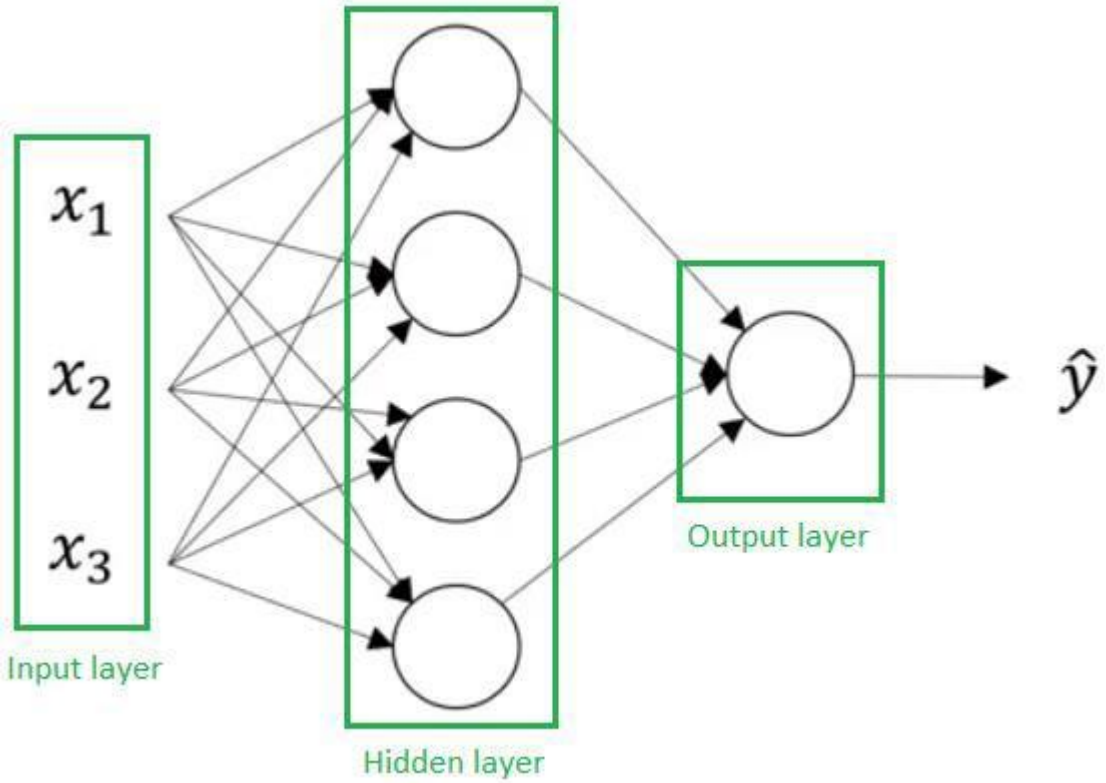


Cross Entropy Loss Function

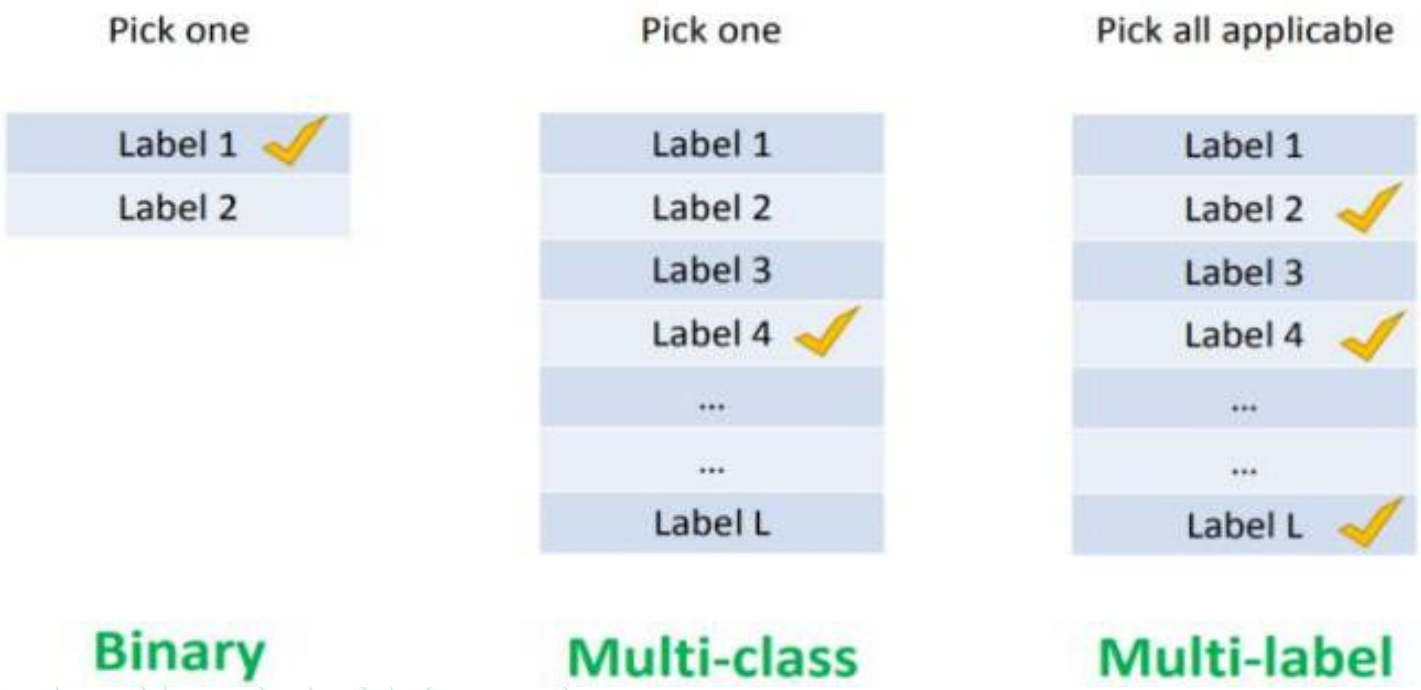
$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$


Loss
2
1.5
1
0.5
0
0.2
0.4
0.6
0.8
1
h(x)
-log(h(x))
-log(1-h(x))
$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1-y_i) \cdot log(1-p(y_i))$
Binary Cross-Entropy / Log Loss

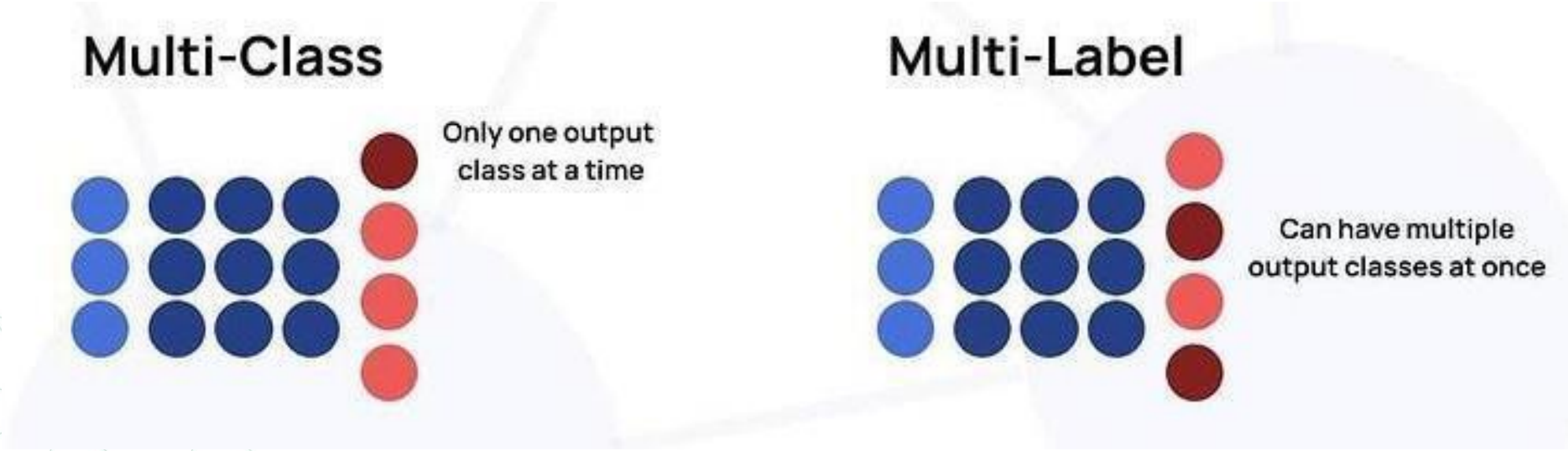
Neural Network



Classification



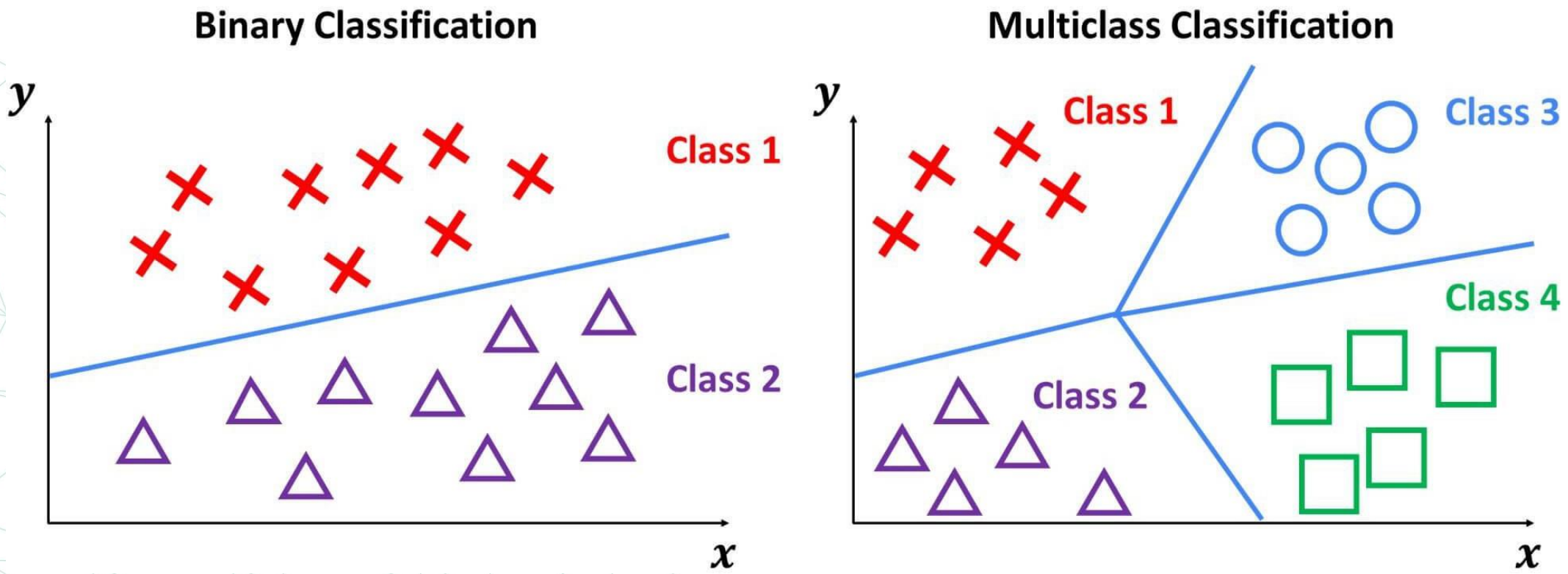
Multilabel classification



Review	sentiment
Very good quality though	Positive
The design is very odd	Negative
I advise EVERYONE DO NOT BE FOOLED!	Negative
So Far So Good!	Positive
Works great!	positive

movie	Adventure	Comedy	Fantasy	Crime
Jumanji (1995)	1	0	1	0
Puccini for Beginners (2006)	0	1	0	0
How the Grinch Stole Christmas! (1966)	0	1	1	0

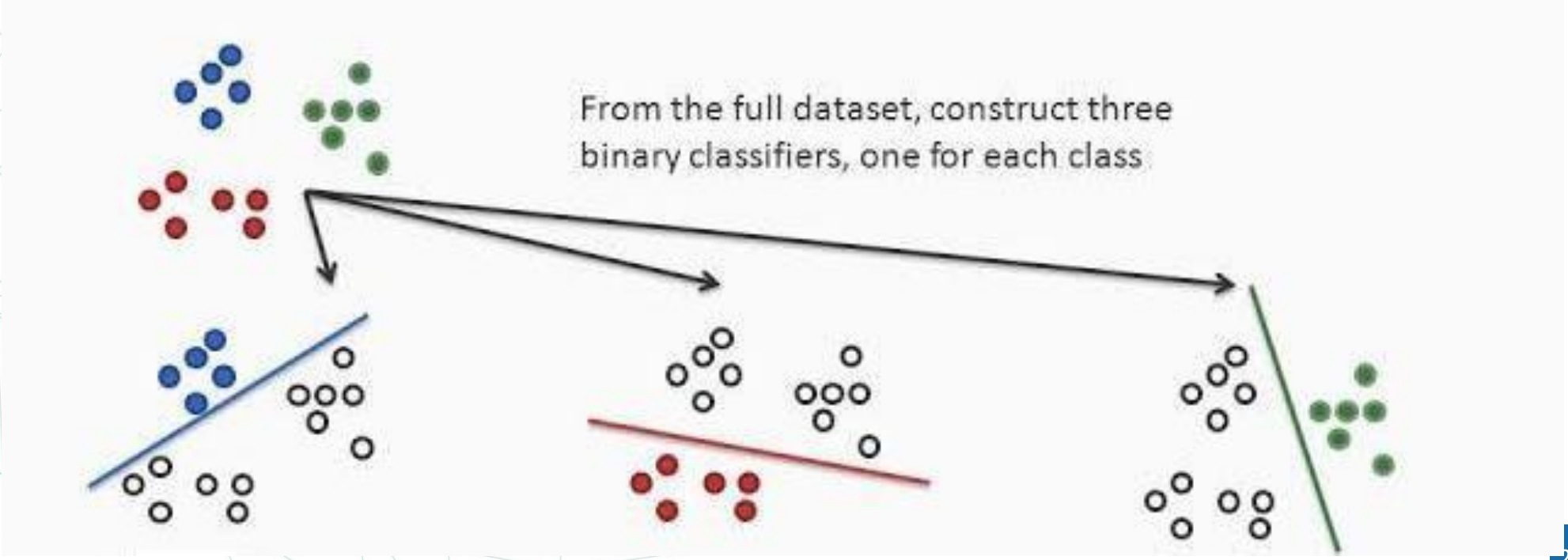
Multi-Class Classification



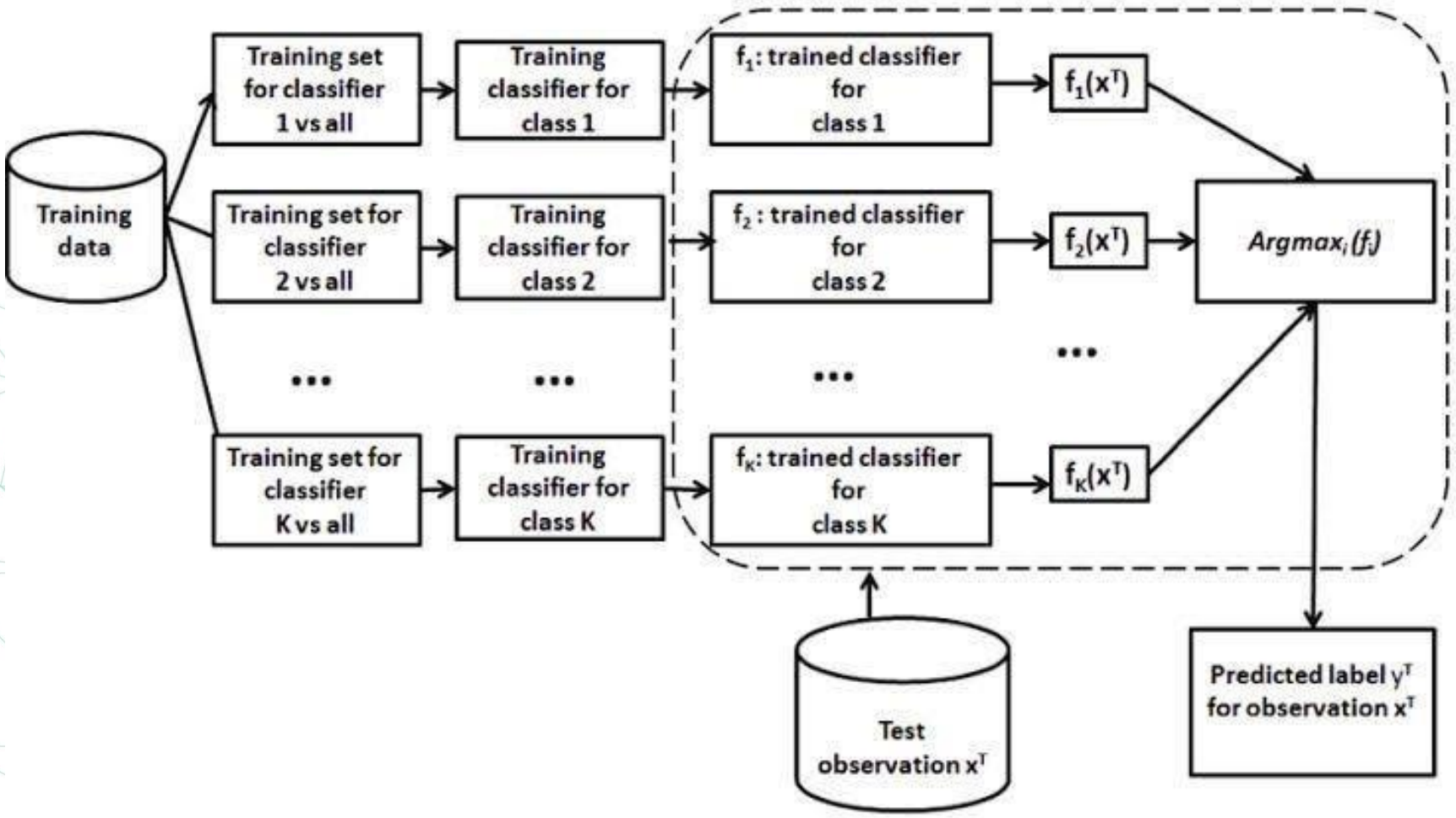
Multiclass Classification

- One-vs-All (One-vs-Rest):

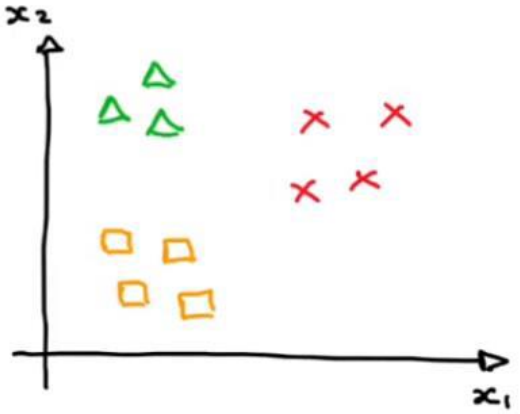
  converts a multiclass problem into a series of binary tasks for each class in the target

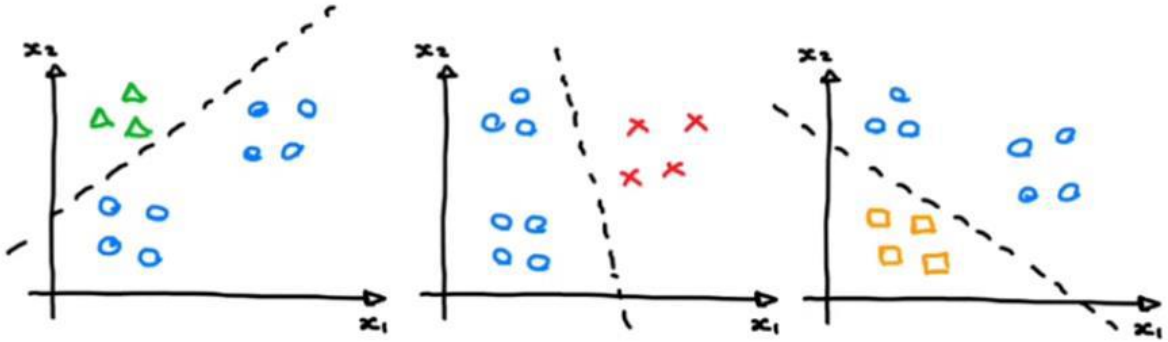


One-vs-All


Training data
Training set for classifier 1 vs all
Training classifier for class 1
f1: trained classifier for class 1
f1(xT)
Training set for classifier 2 vs all
Training classifier for class 2
f2 : trained classifier for class 2
f2(xT)
Argmaxi (fi)
Training set for classifier K vs all
Training classifier for class K
fK: trained classifier for class K
fK(xT)
Test observation xT
Predicted label yT for observation xT

One-vs-rest classification





Strategy: Train three classifiers with $y \in \{0, 1\}$, where each classifier considers another class as the positive class.

We then get three classification models:

$f_1(\underline{x}; \underline{w}_1)$ 1: △

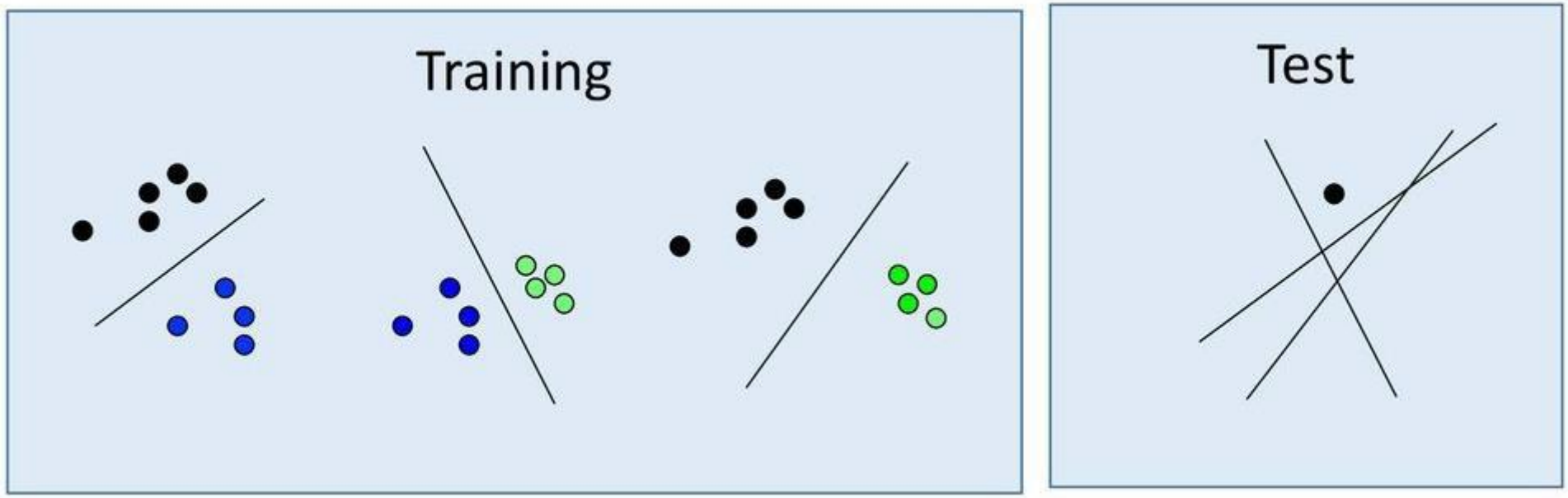
$f_2(\underline{x}; \underline{w}_2)$ 2: ×

$f_3(\underline{x}; \underline{w}_3)$ 3: □

Final predictions: $\arg\max_k f_k(\underline{x}; \underline{w}_k)$

Multiclass Classification

- All-vs-All (One-vs-One):
  - splits a multi-class problem into a single binary classification task for each pair of classes.



Practical Lab

Iris Dataset

Naive Bays

Naive Bayes

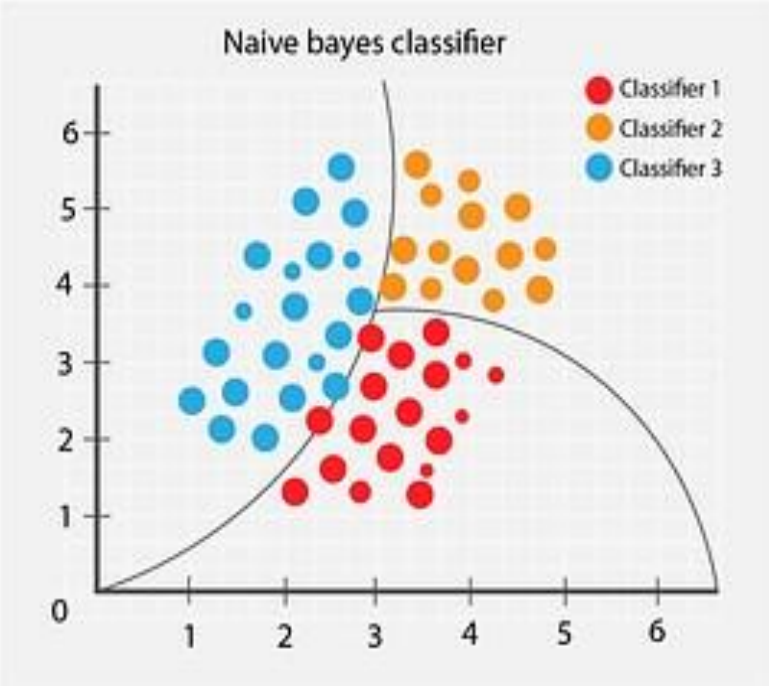
thatware.co

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$



Naïve Bayes

$$P(Y|X) = \frac{P(X|Y).P(Y)}{P(X)} \qquad (1)$$

$$P(X) = P(X_1, X_2, \dots\dots, X_n) = P(X_1).P(X_2) \dots\dots P(X_n)$$

$$because\ for\ independent\ events\ A\ and\ B, P(AB) = P(A).P(B)$$

Similarly because of conditional independence,

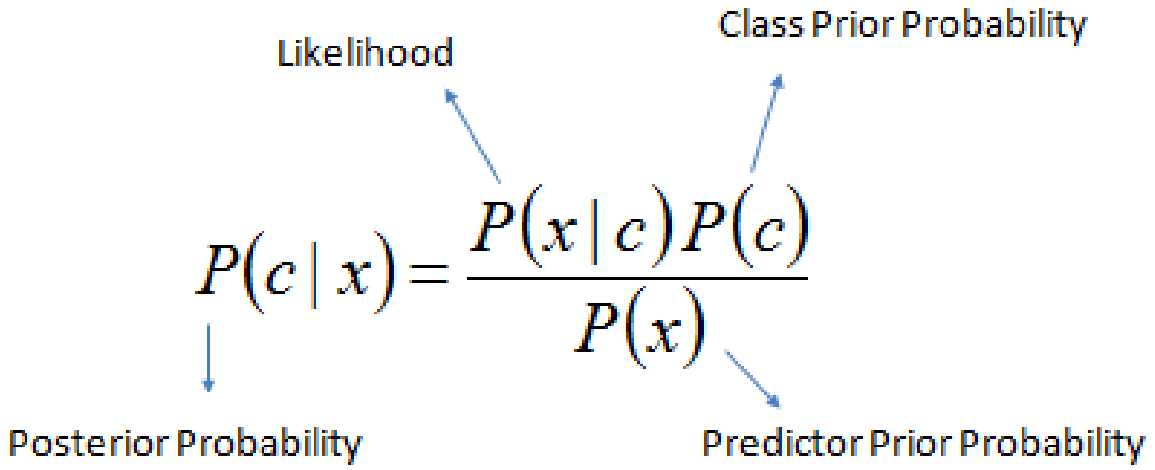
$$P(X|Y) = P(X_1, X_2, \dots\dots, X_n|Y) = P(X_1|Y).P(X_2|Y) \dots\dots P(X_n|Y)$$

Substituting these in (1), we get

$$P(Y|X) = \frac{P(X_1|Y).P(X_2|Y) \dots\dots P(X_n|Y).P(Y)}{P(X_1).P(X_2) \dots\dots P(X_n)}$$

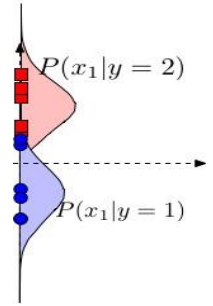
$$P(Y|X) \propto P(X_1|Y).P(X_2|Y) \dots\dots P(X_n|Y).P(Y)$$

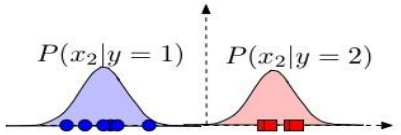
Naïve Bayes

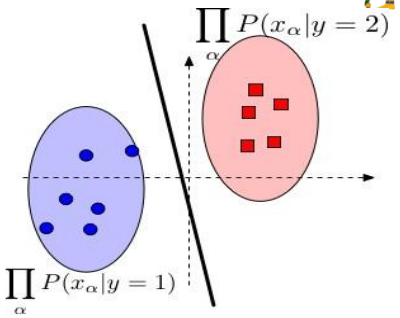


$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Naïve Bayes Classifier







Obs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
X1	A	A	A	A	A	B	B	B	B	B	C	C	C	C	C
X2	S	M	M	S	S	S	M	M	L	L	L	M	M	L	L
Y	0	0	1	1	0	0	0	1	1	1	1	1	1	1	0

Prior Probability

P(Y=1) = 9/15 P(Y=0)=6/15

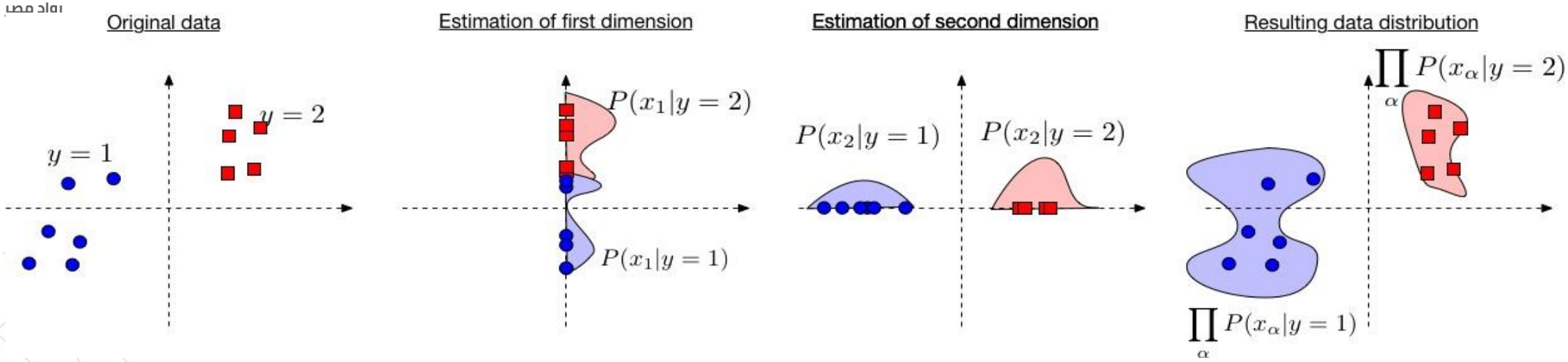
Conditional Probability

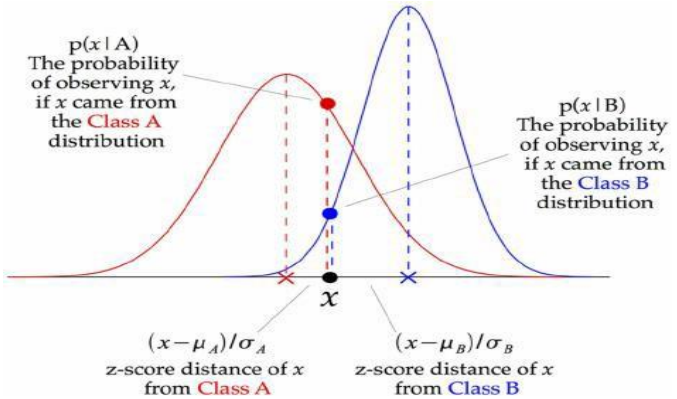
P(X1=A|Y=1)=2/9 P(X1=B|Y=1)=3/9 P(X1=C|Y=1)=4/9
P(X2=S|Y=1)=1/9 P(X2=M|Y=1)=4/9 P(X2=L|Y=1)=4/9
P(X1=A|Y=0)=3/6 P(X1=B|Y=0)=2/6 P(X1=C|Y=0)=1/6
P(X2=S|Y=0)=3/6 P(X2=M|Y=0)=2/6 P(X2=L|Y=0)=1/6

$P(Y=1)P(X1=B|Y=1)P(X2=S|Y=1)=1/45$

$P(Y=0)P(X1=B|Y=0)P(X2=S|Y=0)=1/15$

$P(Y=0)P(X1=B|Y=0)P(X2=S|Y=0) > P(Y=1)P(X1=B|Y=1)P(X2=S|Y=1)$, so $y=0$.





$$h(\vec{x}) = \operatorname*{argmax}_y P(y|\vec{x})$$

$$= \operatorname*{argmax}_y \frac{P(\vec{x}|y)P(y)}{P(\vec{x})}$$

$$= \operatorname*{argmax}_y P(\vec{x}|y)P(y) \qquad (P(\vec{x}) \text{ does not depend on } y)$$

$$= \operatorname*{argmax}_y \prod_{\alpha=1}^{d} P(x_\alpha|y)P(y) \qquad (\text{by the naive assumption})$$

Naïve Bayes Classifier

- Naïve Bayes Classifier can be trained easily and fast and can be used as benchmark model.
- When variable selection is carried out properly, Naïve Bayes can perform as well as or even better than other statistical models such as logistic regression.
- Naive Bayes requires a strong assumption of independent predictors, so when the model has a bad performance, the reason leading to that may be the dependence between predictors.
- Doesn’t require feature scaling

Naïve Bayes Example

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

$$P(x \mid c) = P(Sunny \mid Yes) = 3/9 = 0.33$$

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	3/9	2/5	5/14
	Overcast	4/9	0/5	4/14
	Rainy	2/9	3/5	5/14
		9/14	5/14	

$$P(x) = P(Sunny) = 5/14 = 0.36$$

$$P(c) = P(Yes) = 9/14 = 0.64$$

Posterior Probability: $P(c \mid x) = P(Yes \mid Sunny) = 0.33 \times 0.64 \div 0.36 = 0.60$

$$P(x \mid c) = P(Sunny \mid No) = 2/5 = 0.4$$

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
		9	5	14

$$P(x) = P(Sunny) = 5/14 = 0.36$$

$$P(c) = P(No) = 5/14 = 0.36$$

Posterior Probability: $P(c \mid x) = P(No \mid Sunny) = 0.40 \times 0.36 \div 0.36 = 0.40$

Frequency Table

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Likelihood Table

		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

Outlook	Temp	Humidity	Windy	Play
Rainy	Cool	High	True	?

$$P(Yes \mid X) = P(Rainy \mid Yes) \times P(Cool \mid Yes) \times P(High \mid Yes) \times P(True \mid Yes) \times P(Yes)$$

$$P(Yes \mid X) = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.00529 \rightarrow 0.2 = \frac{0.00529}{0.02057 + 0.00529}$$

$$P(No \mid X) = P(Rainy \mid No) \times P(Cool \mid No) \times P(High \mid No) \times P(True \mid No) \times P(No)$$

$$P(No \mid X) = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.02057 \rightarrow 0.8 = \frac{0.02057}{0.02057 + 0.00529}$$

Q&A

Questions and answers

Thanks