

Machine Learning

Gentle Introduction

Ali Abdelaal, Machine learning research engineer @ Mawdoo3.com

Machine learning introduction

topics to be discussed

- AI Applications
- AI, machine learning and deep learning
- How machine learning works
 - Gradient descent
- Supervised and unsupervised learning
- Basic tutorial on applying machine learning
- recommended resources

A close-up photograph of a white humanoid robot's hand holding a black smartphone. The robot's fingers are wrapped around the device, which is displaying a dark screen. The background is a plain, light-colored wall.

AI is everywhere !



كم يبلغ طول نهر النيل

الأخبار الكل

الإعدادات المزيد فيديو صور خرائط

حوالي 1,090,000 نتيجة (عدد العواني: 0.55)

نهر النيل / الطول

6,650 كم

يبحث الأشخاص أيضاً عن

نهر مسيسيبي 3.778 ألف كم

يانغتسي 6.3 ألف كم

المازون 6.84 ألف كم

People You May Know

See all friend recommendations

Creeper 1 mutual friend

Creeper 1 mutual friend

Creeper 2 mutual friends

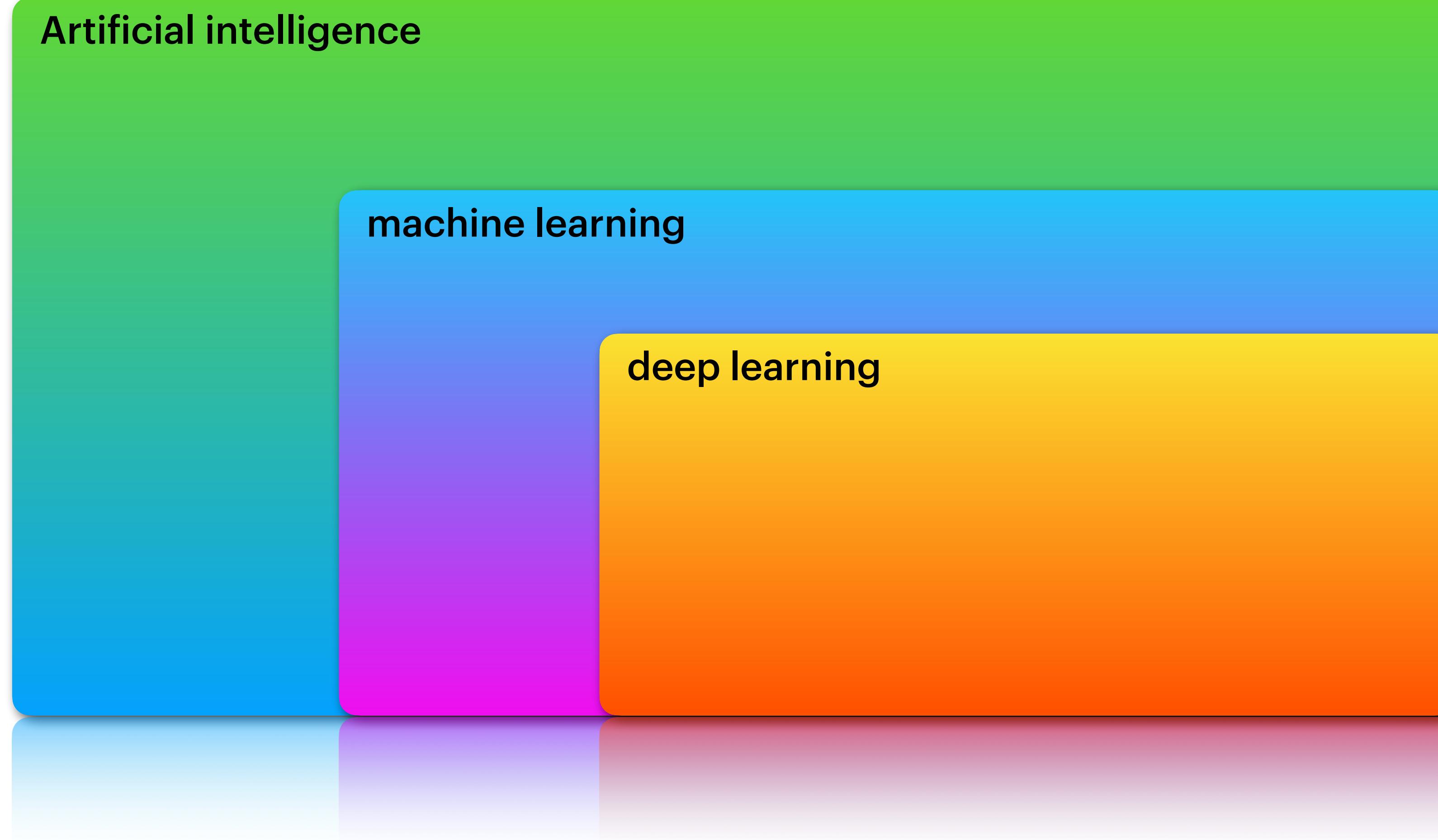
Creeper 1 mutual friend

AI Applications

they are everywhere !

- Siri, google assistant
- Snapchat, instagram filters
- self driving cars
- automatic replies in gmail
- Facebook product suggestions
- youtube videos recommender

AI vs Machine Learning vs Deep Learning



Natural language processing and computer vision

- We use machine learning and deep learning techniques and algorithms to achieve better results in NLP and CV
- The two fields were heavily dependent on hand crafted features
- The advances in machine learning helped the two fields advance drastically

What is machine learning ?

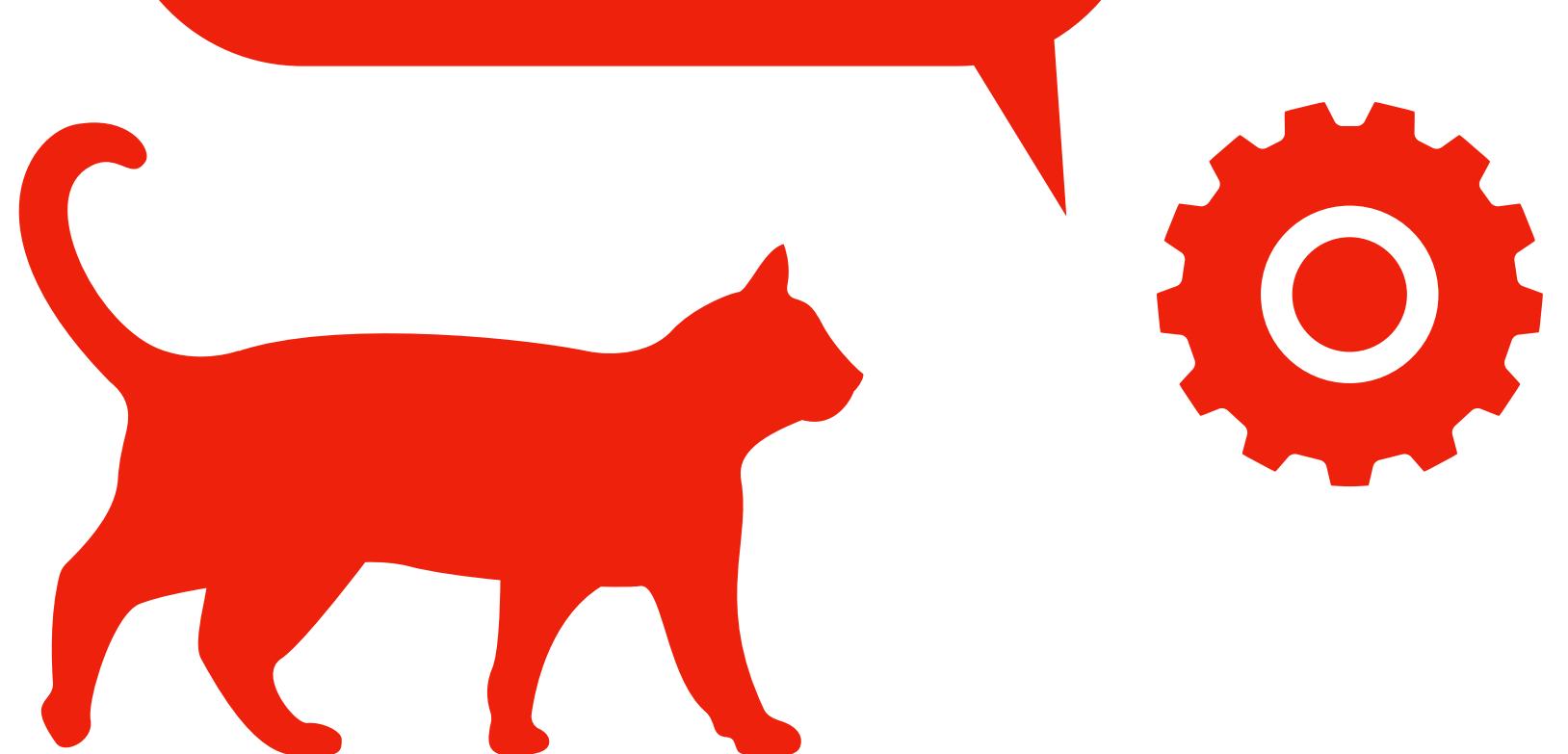
“Machine learning gives computers the ability to learn without being explicitly programmed”

Arthur Samuel, 1959

Traditional programming

```
if ear is triangle:  
    if face is circular:  
        if object has furr:  
            if image is cute:  
                "it's a cat!"
```

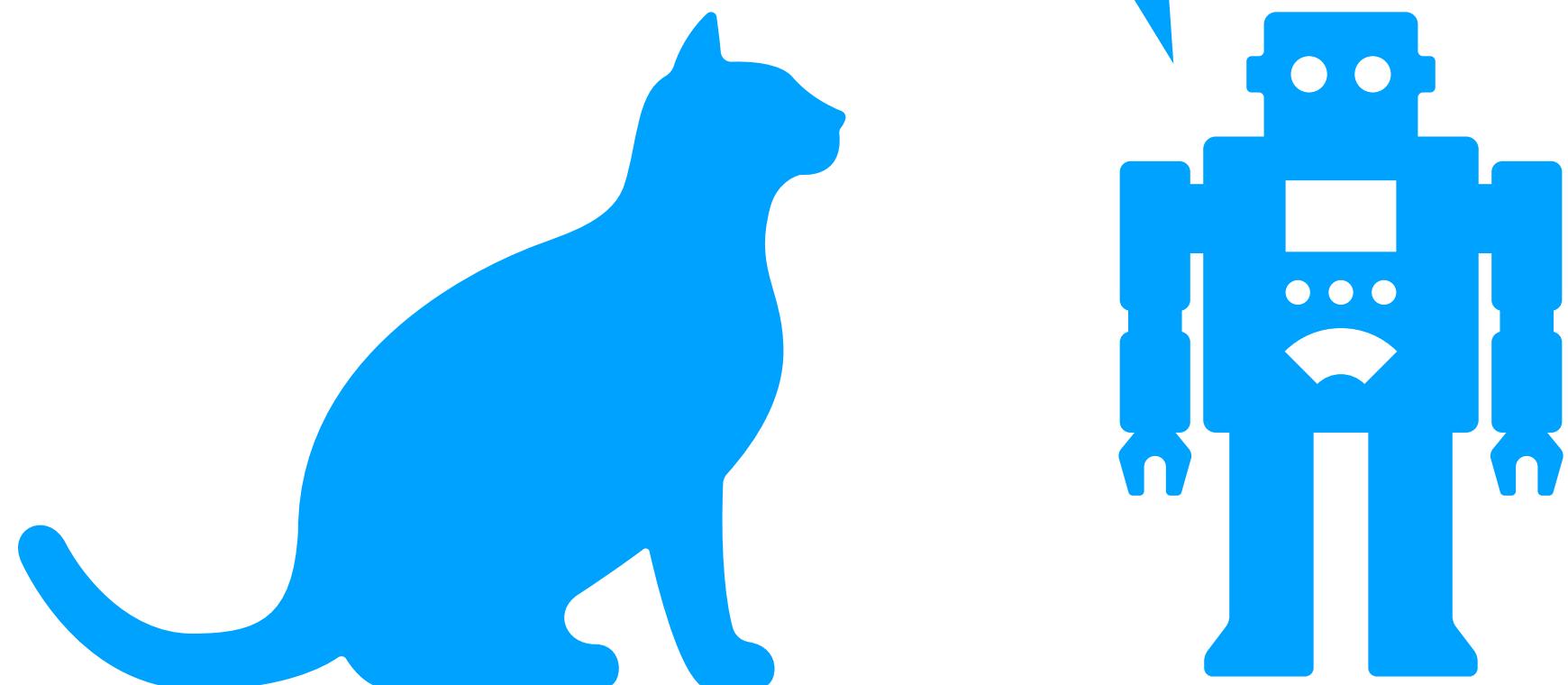
This one doesn't follow
the rules :/



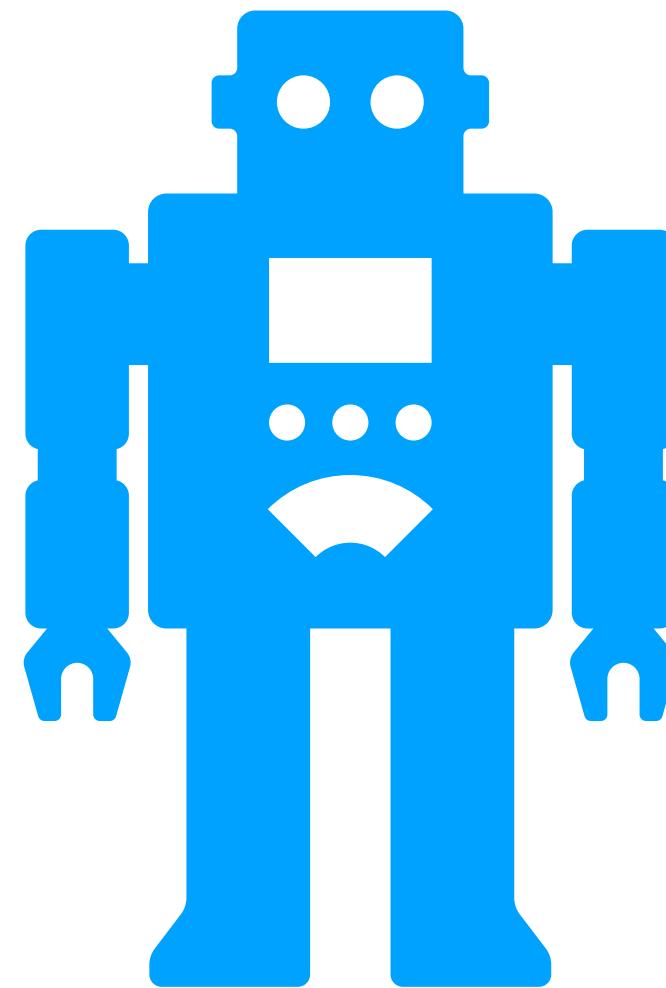
Machine learning

```
cats = [cat1, cat2, cat3, ...]  
machine_learning_model.learn_from(cats)  
machine_learning_model.predict(new_image)
```

looks like what i have
seen before !

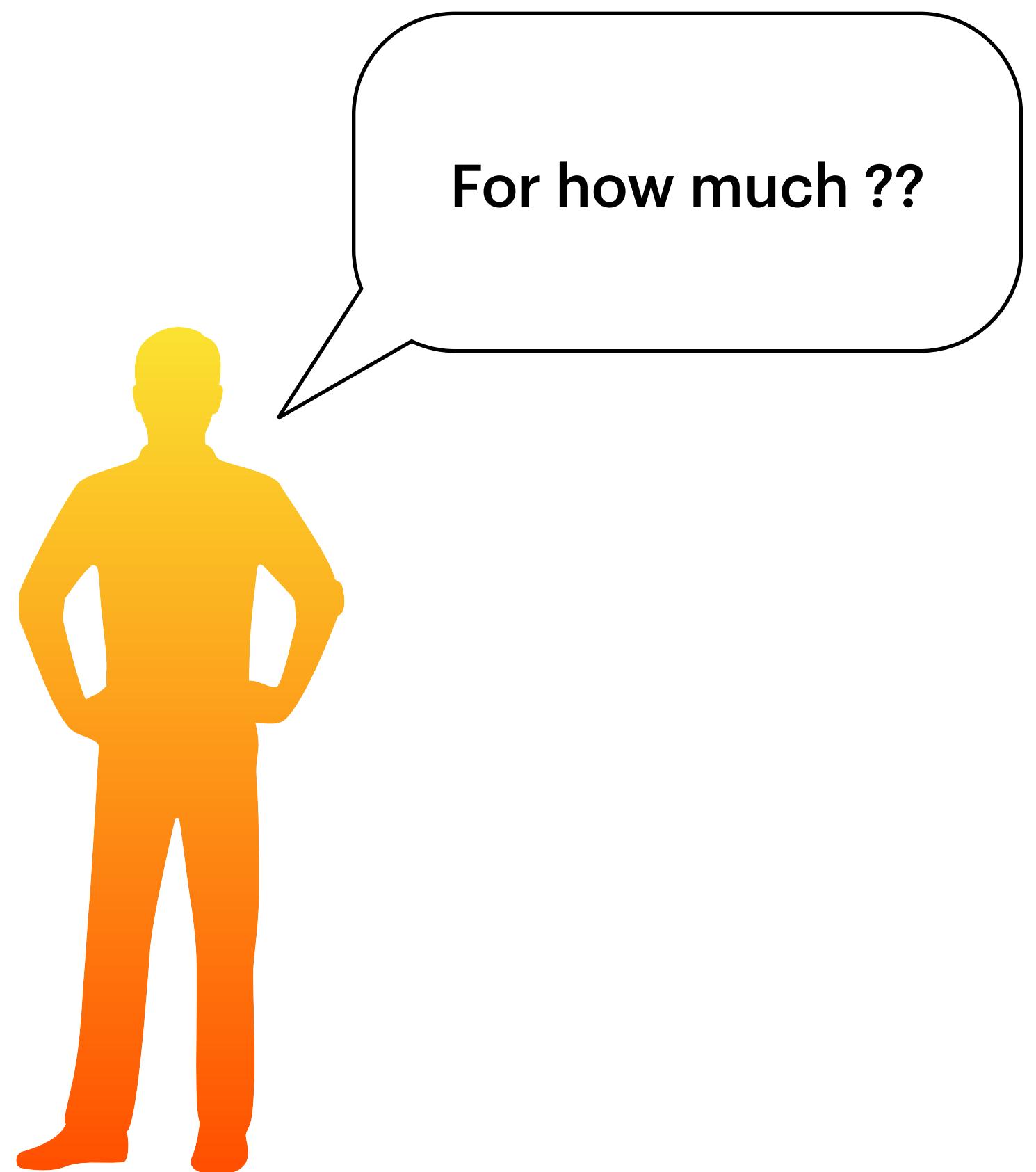


**So machine learning is all about finding
the pattern in your data,
but how ?**



Try it yourself

House floor	House size	price
3	300	\$9300.00
1	250	\$7600.00
5	100	\$800.00
2	220	???



Let's call floor **X**, the size **Y**

$$3 \text{ } \mathbf{X} + 300 \text{ } \mathbf{Y} = 9300$$

$$1 \text{ } \mathbf{X} + 250 \text{ } \mathbf{Y} = 7600$$

$$\text{So } \mathbf{X} = 7600 - 250 \text{ } \mathbf{Y}$$

$$\text{So } 3(7600 - 250 \text{ } \mathbf{Y}) + 300 \text{ } \mathbf{Y} = 9300$$

$$\text{So } \mathbf{Y} = 30$$

$$\text{So } \mathbf{X} = (7600 - 250 * \mathbf{30}) = 100$$

Then the new house price would be

$$\mathbf{100} * 2 + \mathbf{30} * 220 = 6800\text{\$}$$

Assign weights to features.

Sensor reading
to monitor your health and
predict if a problem is about
to happen.

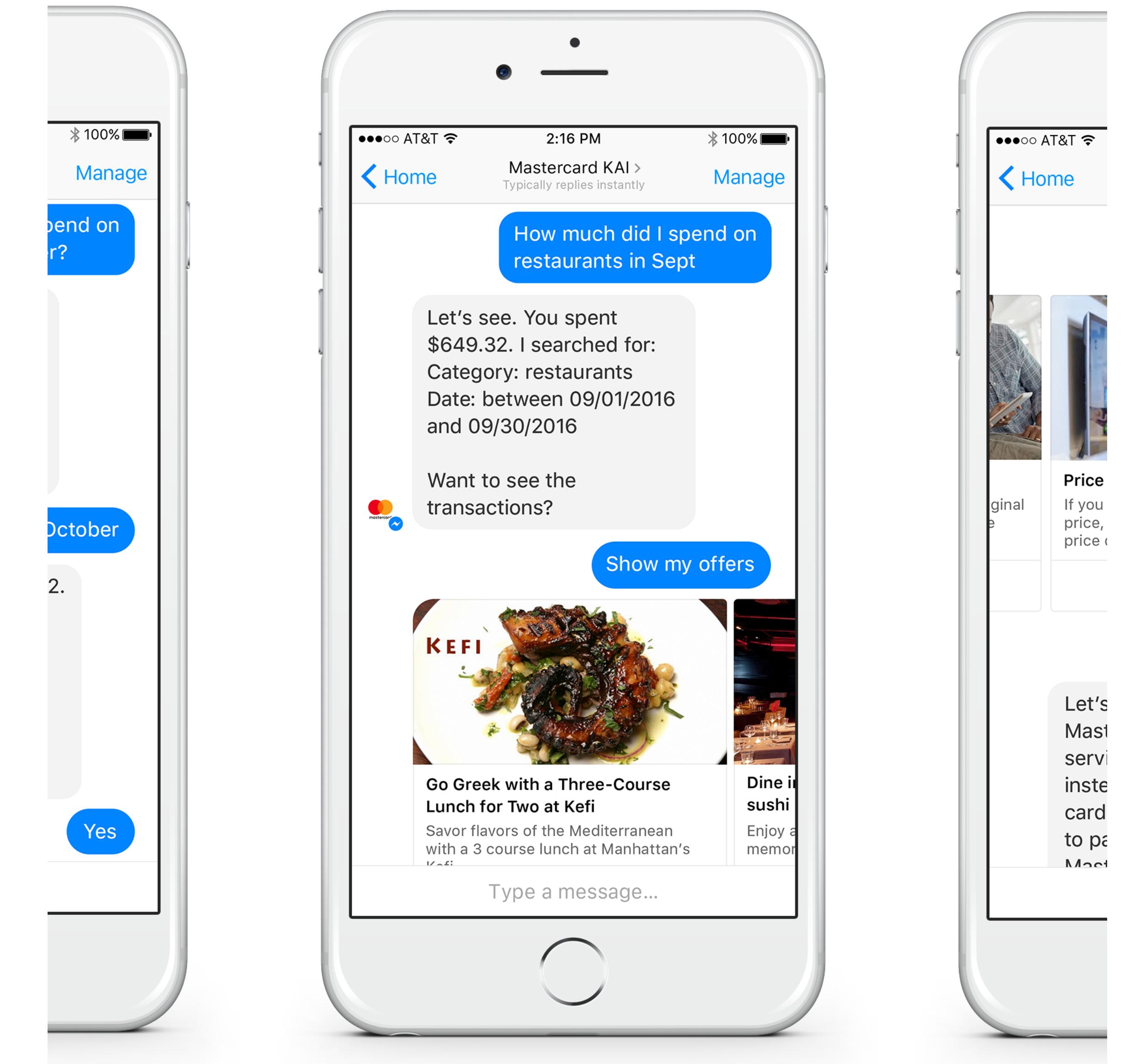


Image pixels

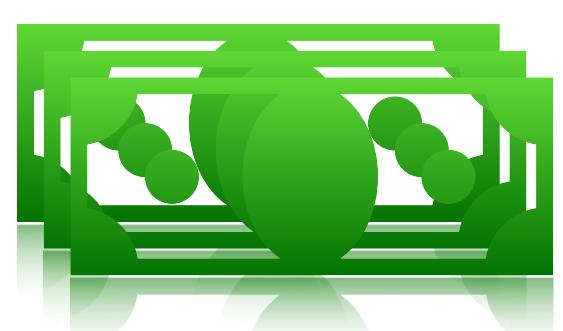
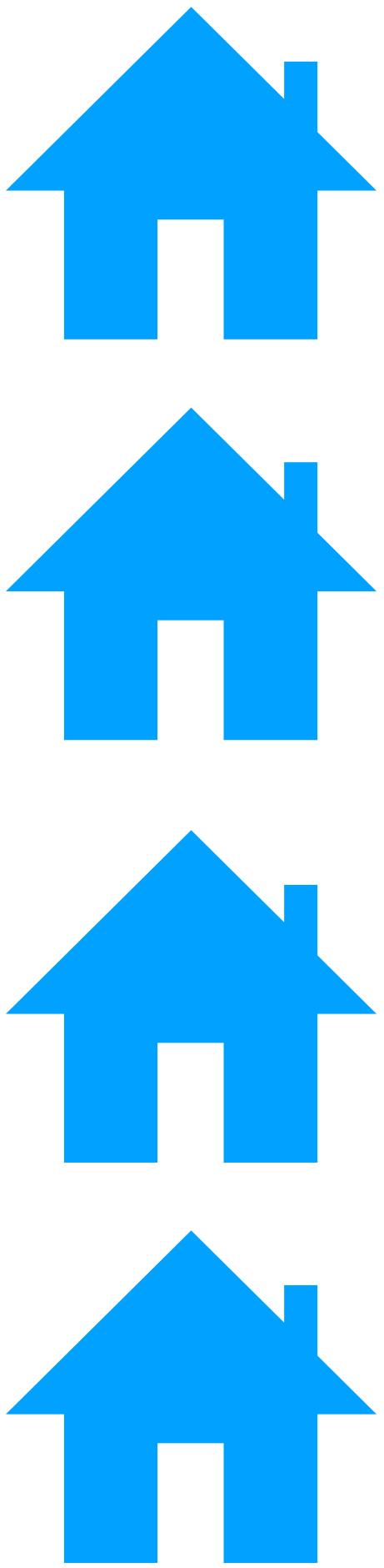
To check if a particular object is in the image or not, or to tell if this is certain user or not.



Your text/voice For a chatbot to know what to do or the best answer for your question.



Lots of variables to solve for

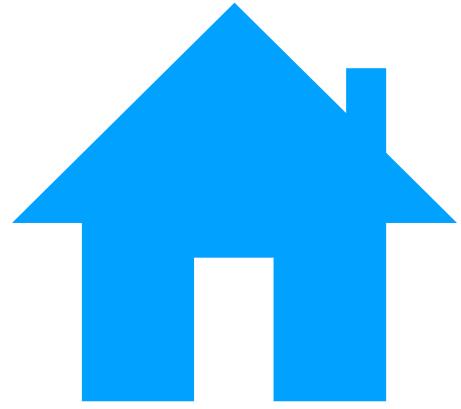


For how much ?

Lots of variables to solve for



feature_1 * W_1 + feature_2 * W_2 + ... + feature_n * W_n =



feature_1 * W_1 + feature_2 * W_2 + ... + feature_n * W_n =

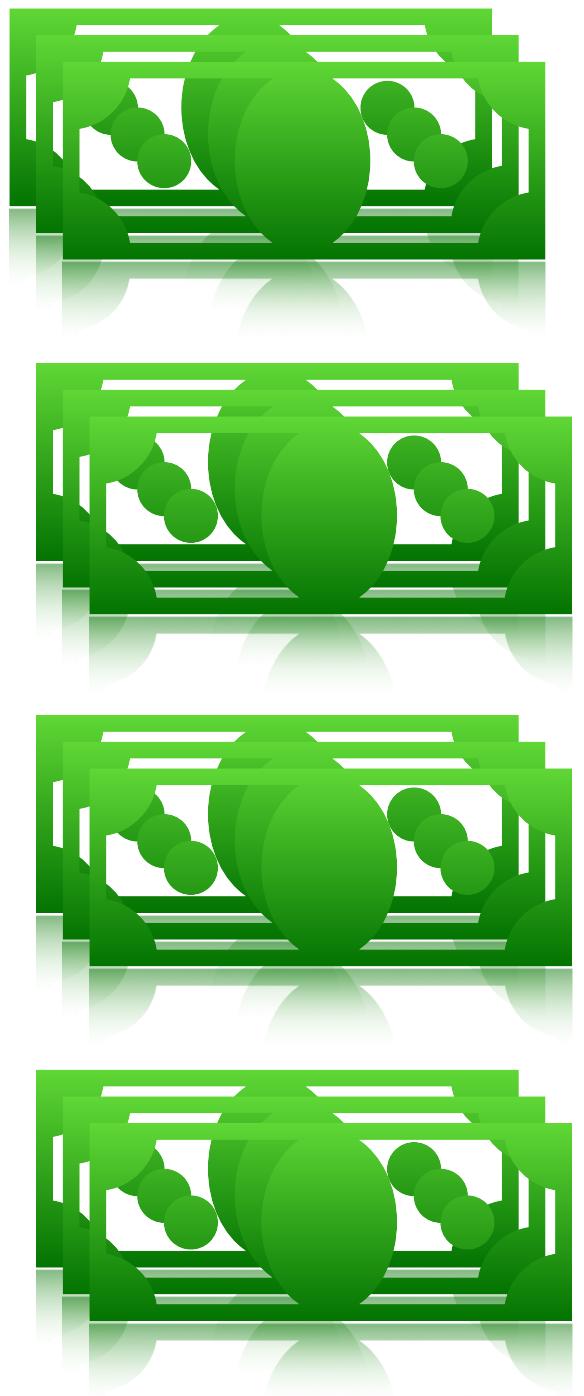


feature_1 * W_1 + feature_2 * W_2 + ... + feature_n * W_n =



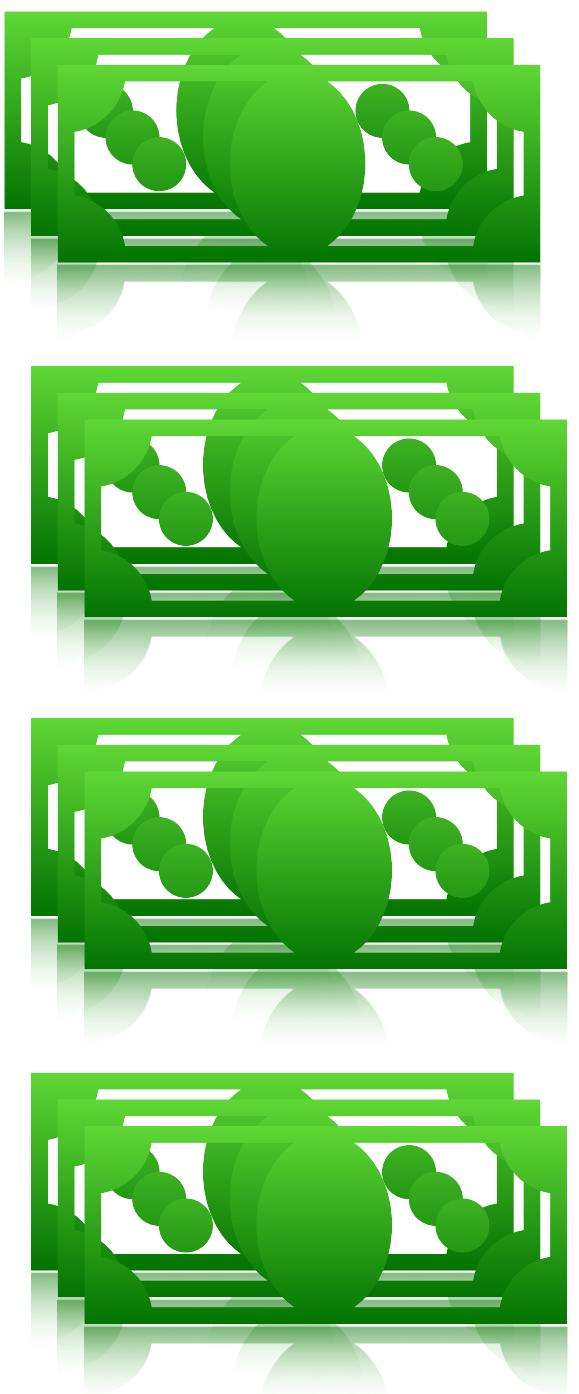
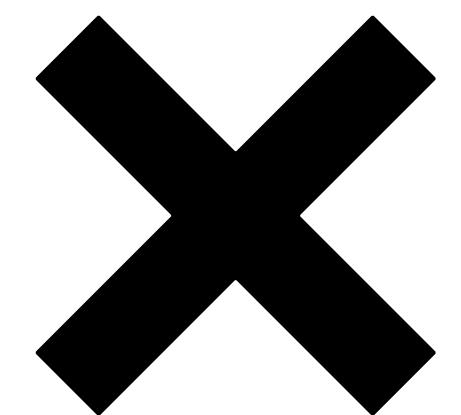
How to automate it

feature_1	feature_2	...	feature_n
253	3	...	11
314	2	...	7
190	3	...	3
204	9	...	6



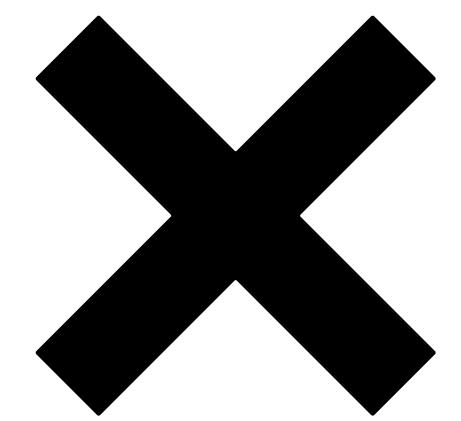
How to automate it

feature_1	feature_2	...	feature_n
253	3	...	11
314	2	...	7
190	3	...	3
204	9	...	6

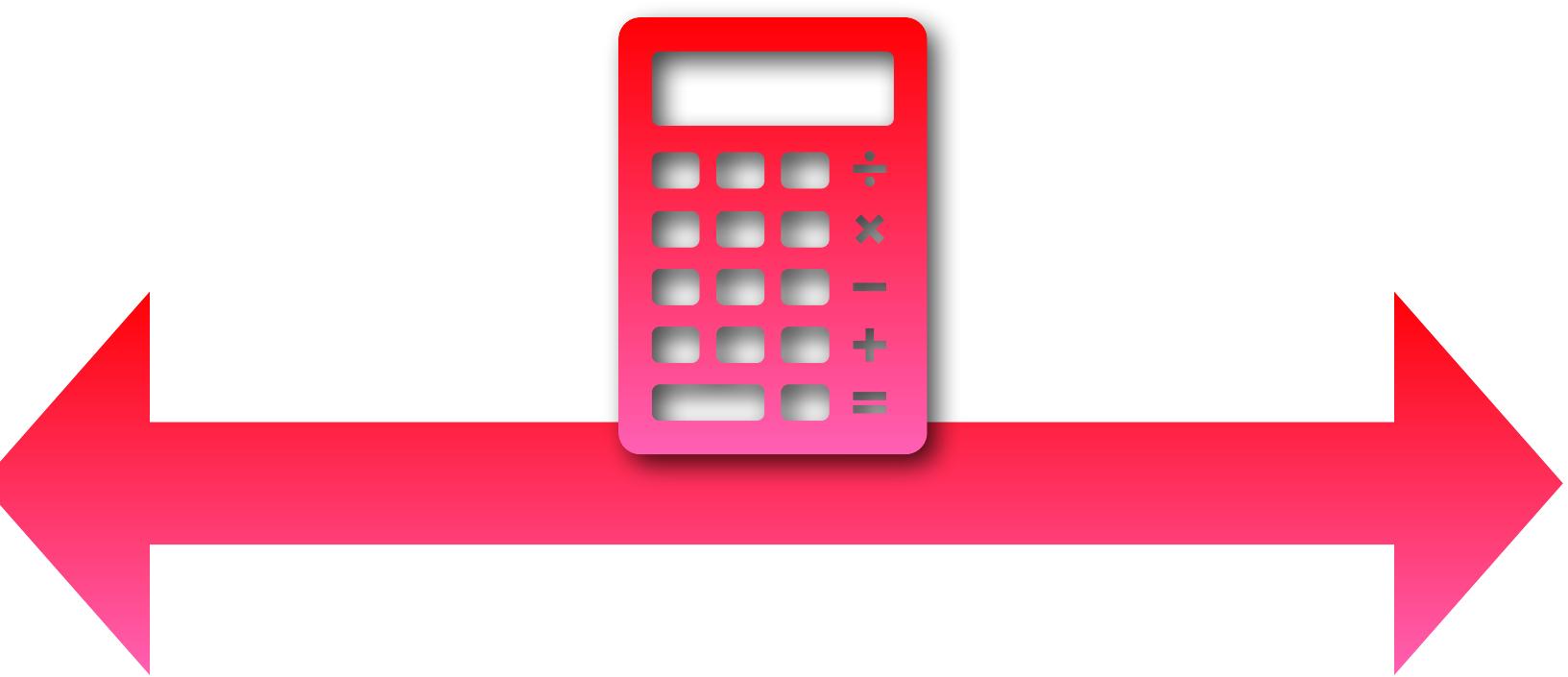
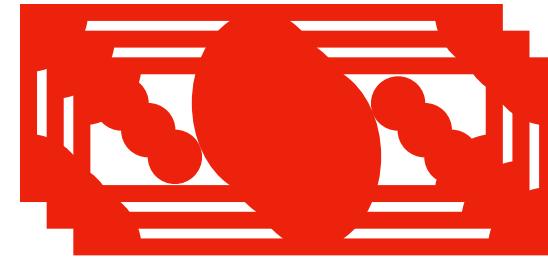
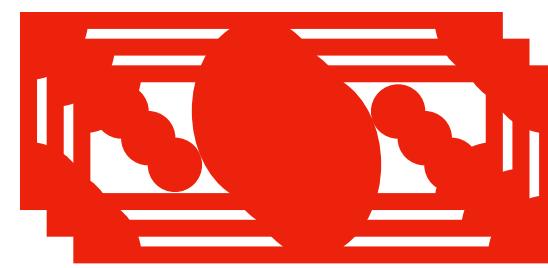
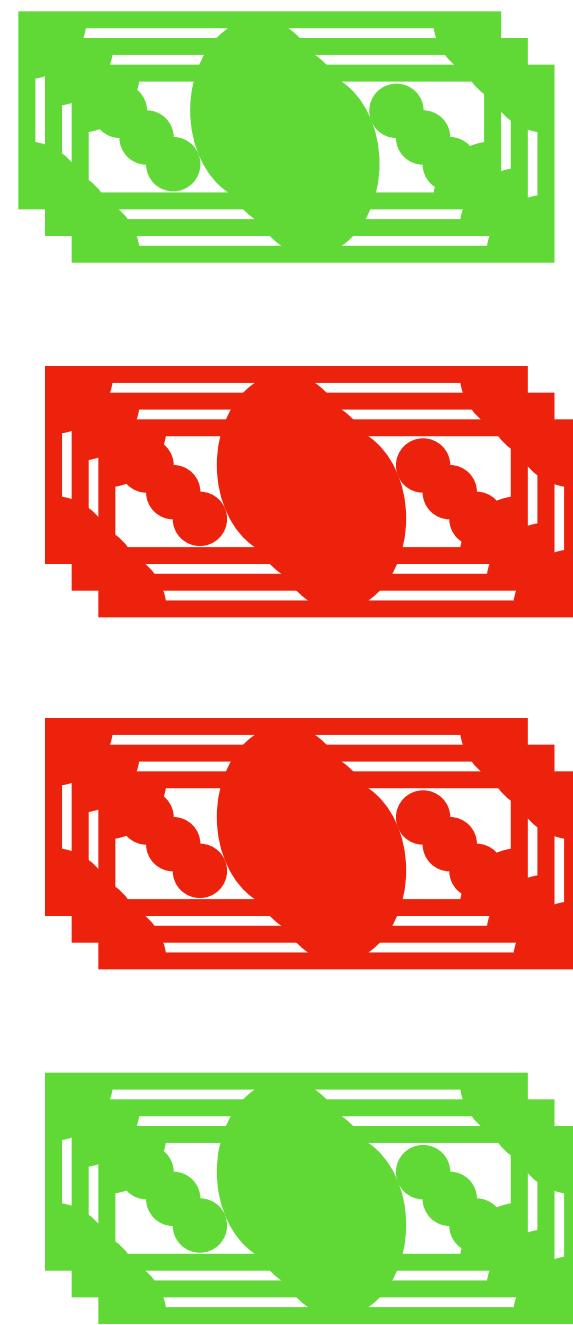


How to automate it

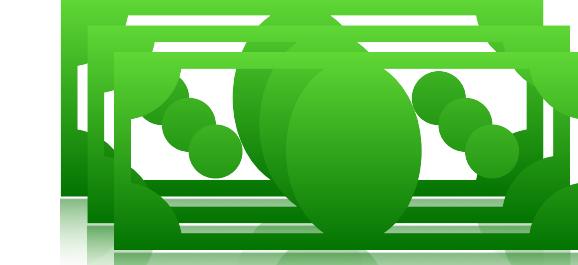
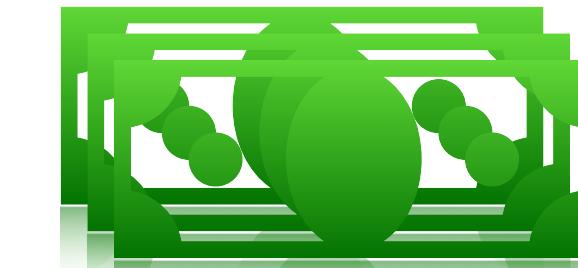
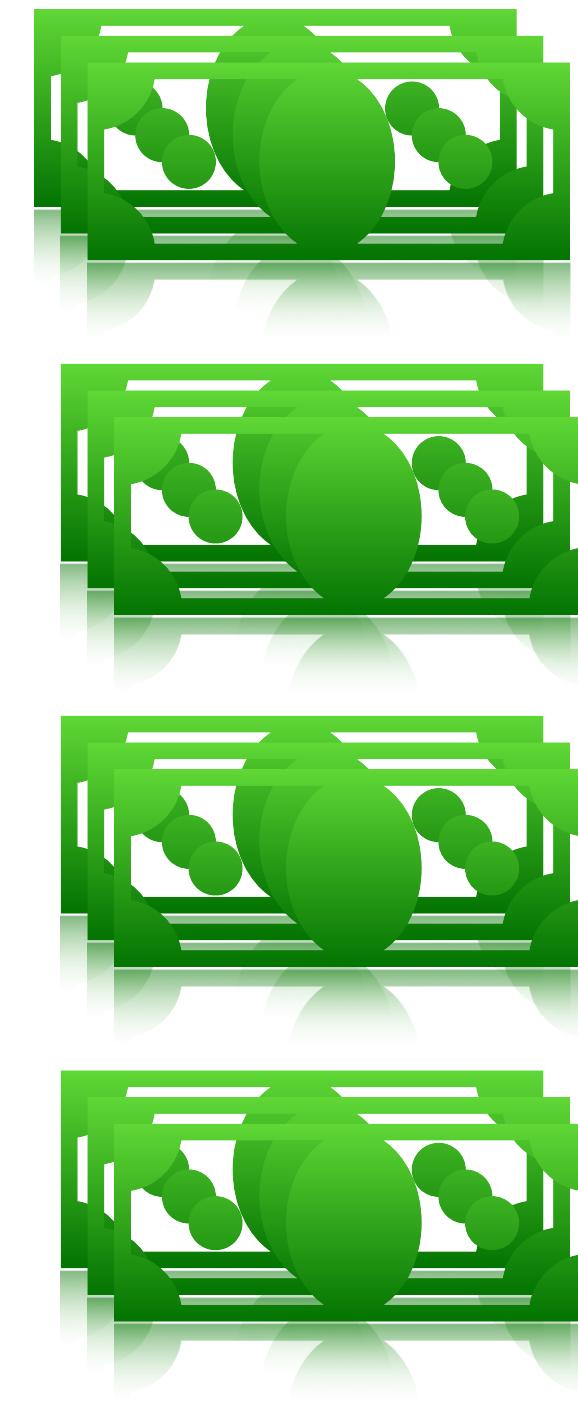
feature_1	feature_2	...	feature_n
253	3	...	11
314	2	...	7
190	3	...	3
204	9	...	6



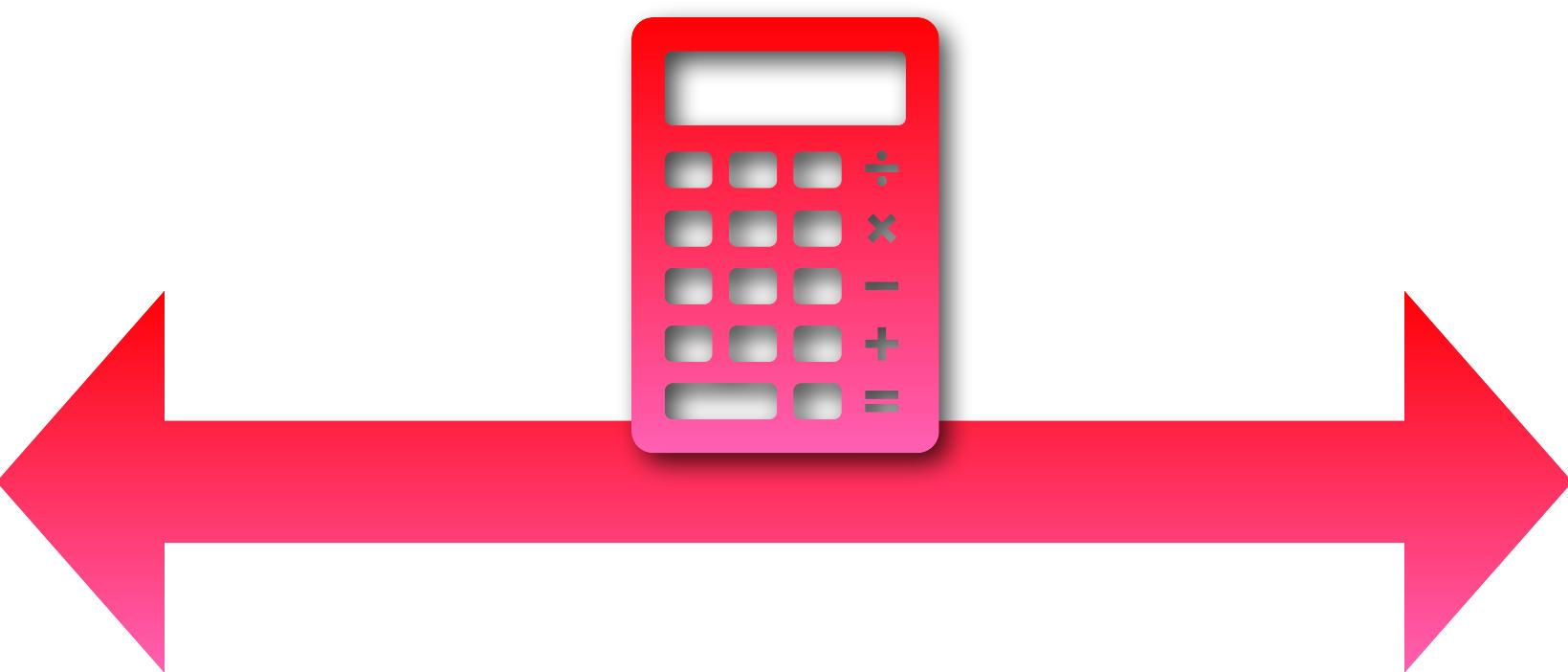
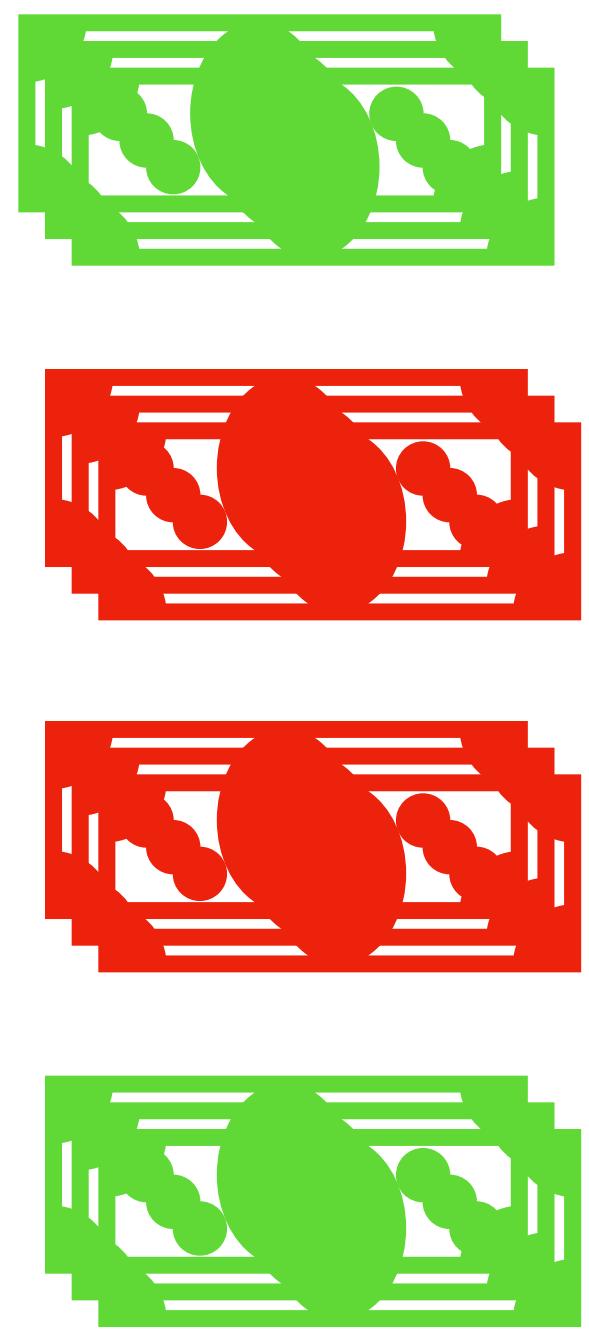
How to automate it



Calculate the error



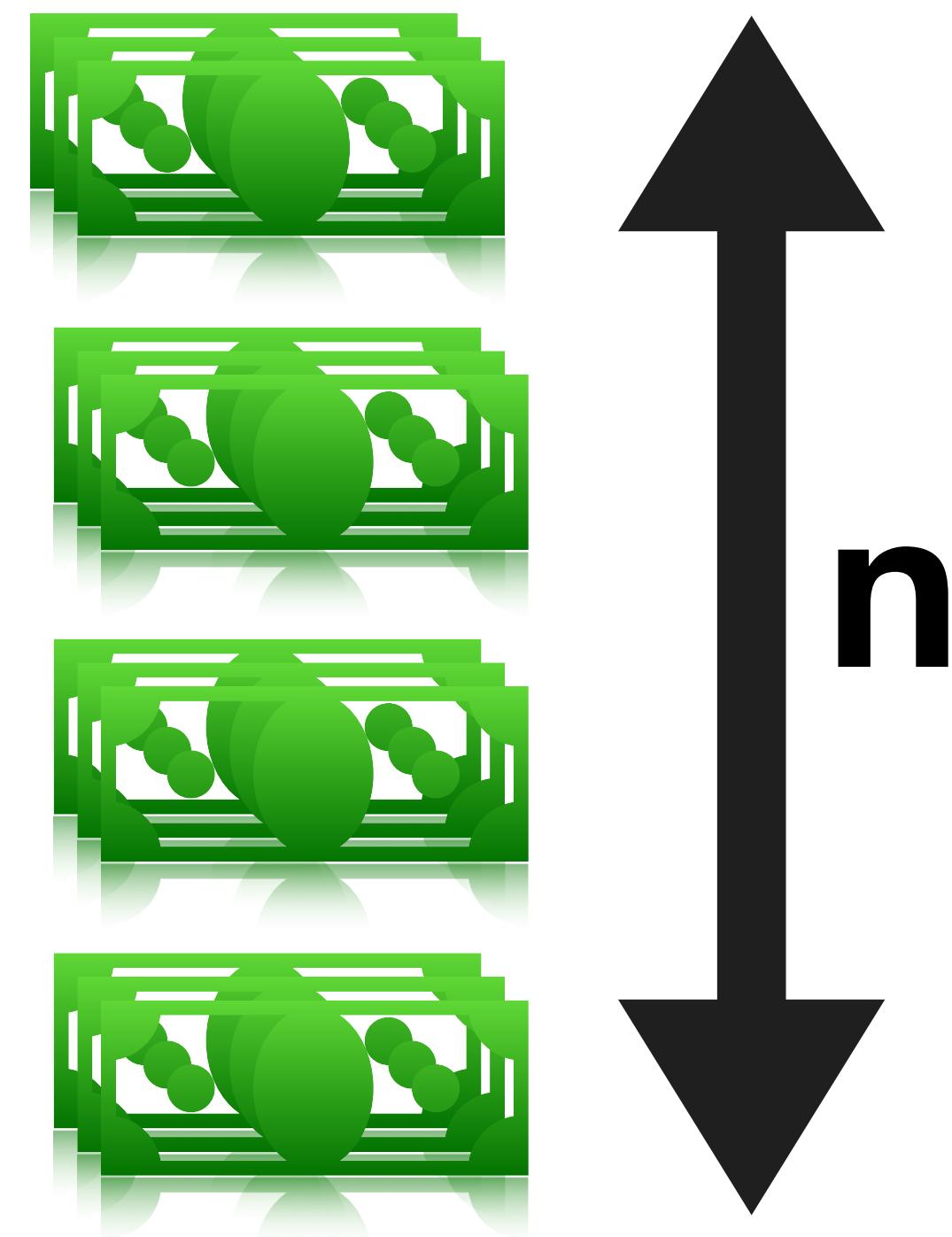
How to automate it



Calculate the error (MSE)

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

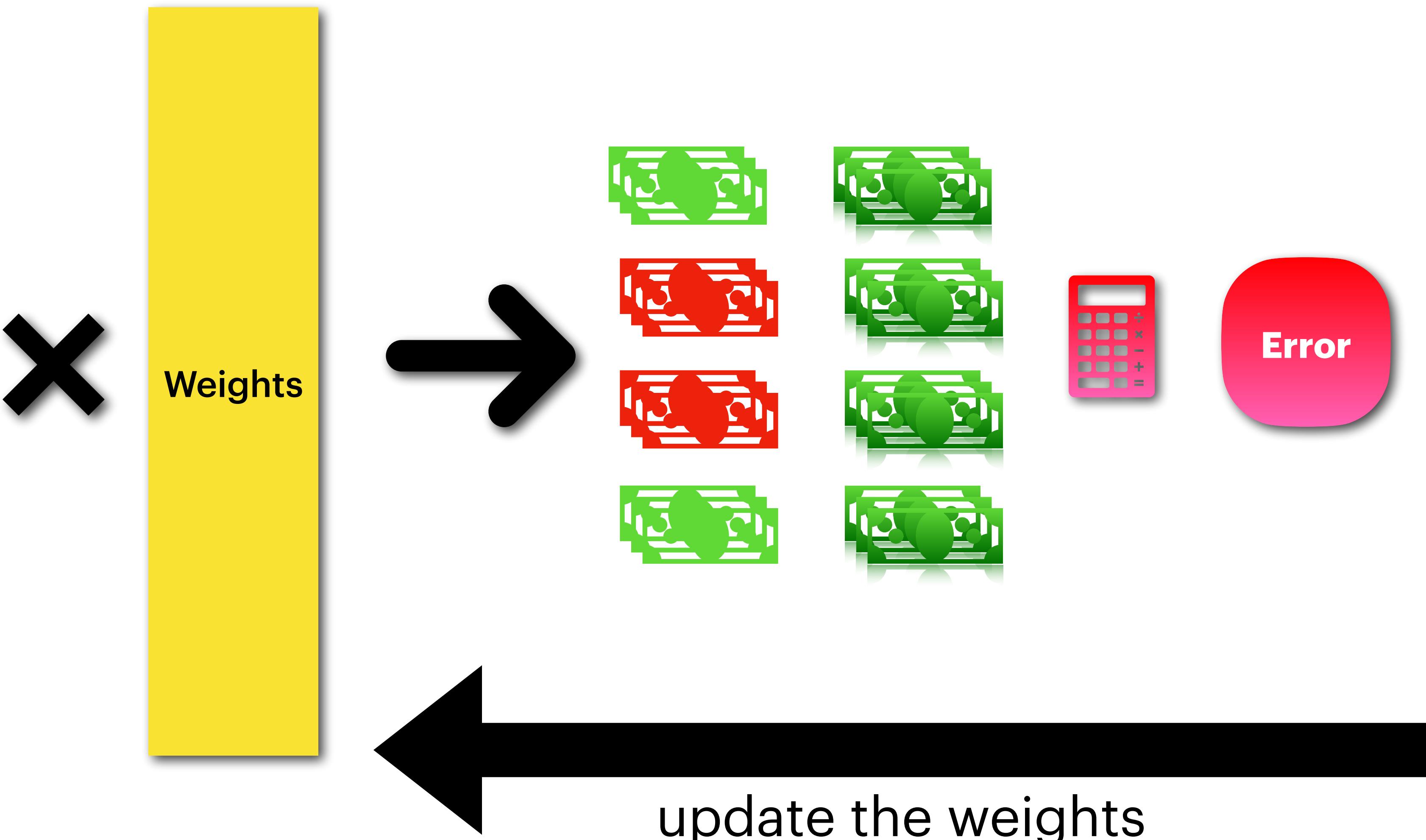
\tilde{y}



y

Update your weights

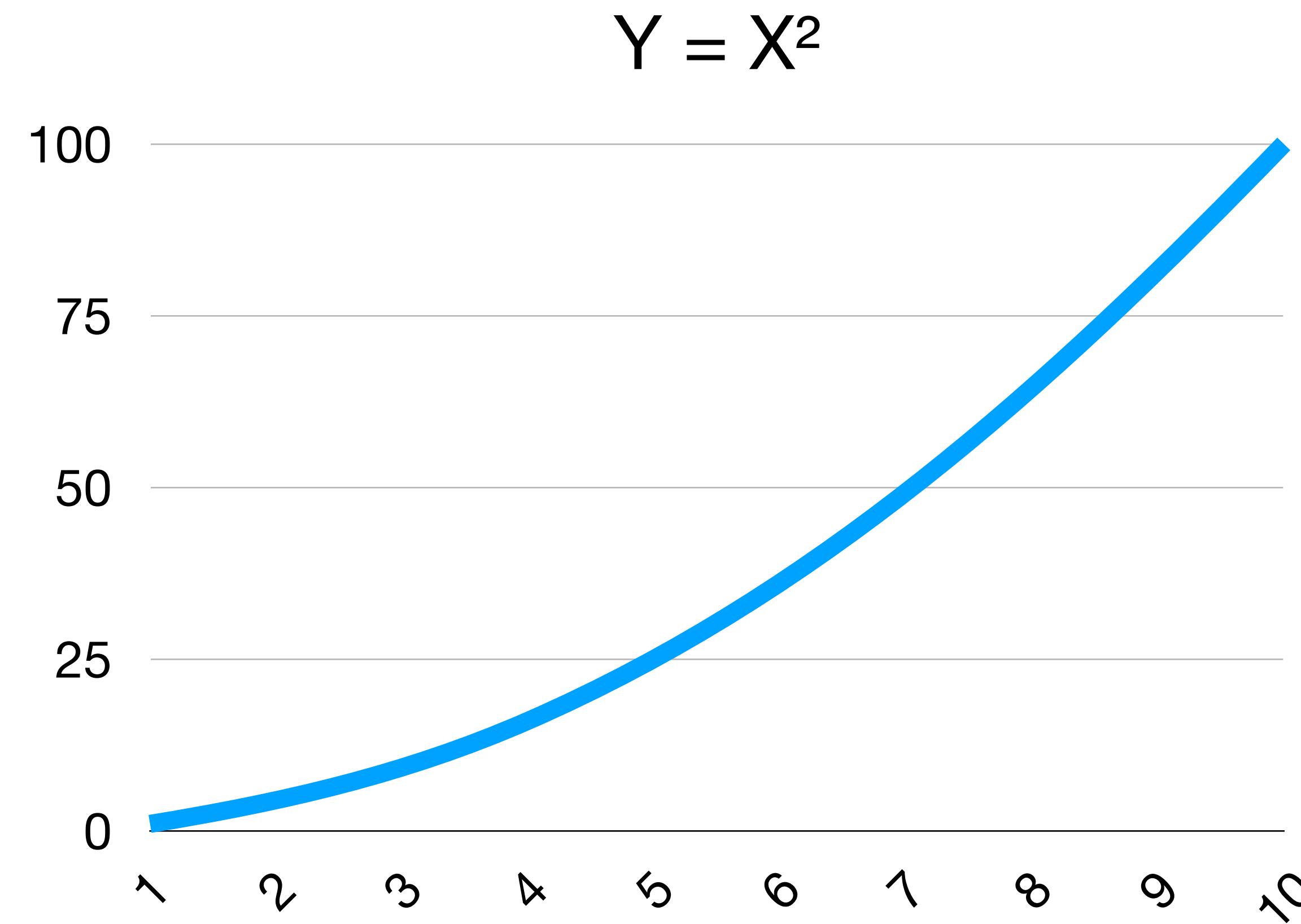
feature_1	feature_2	...	feature_n
253	3	...	11
314	2	...	7
190	3	...	3
204	9	...	6



How to update the weights ?

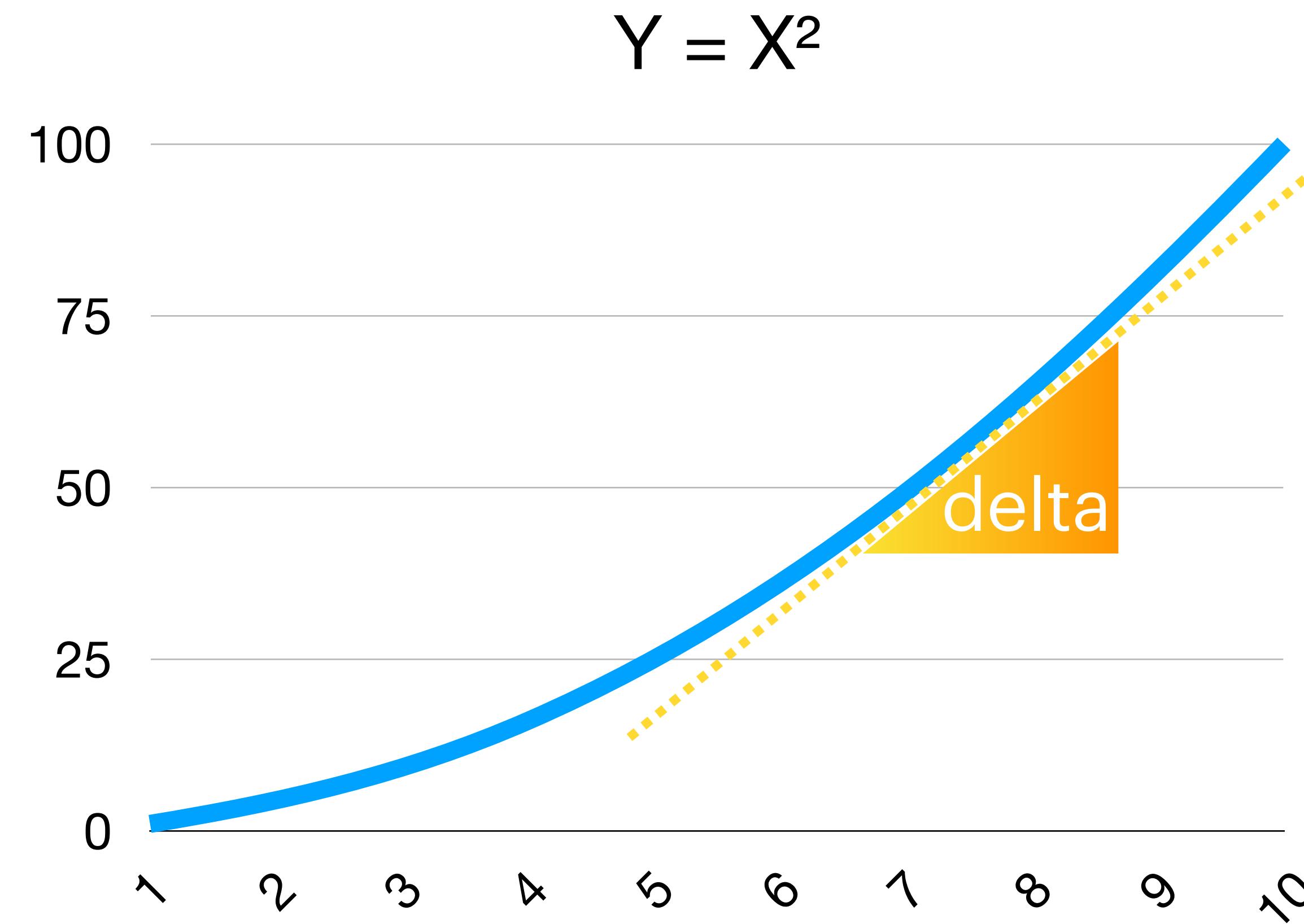
Recall differentiation

How much Y would change if we changed X



Recall differentiation

How much Y would change if we changed X



Recall differentiation

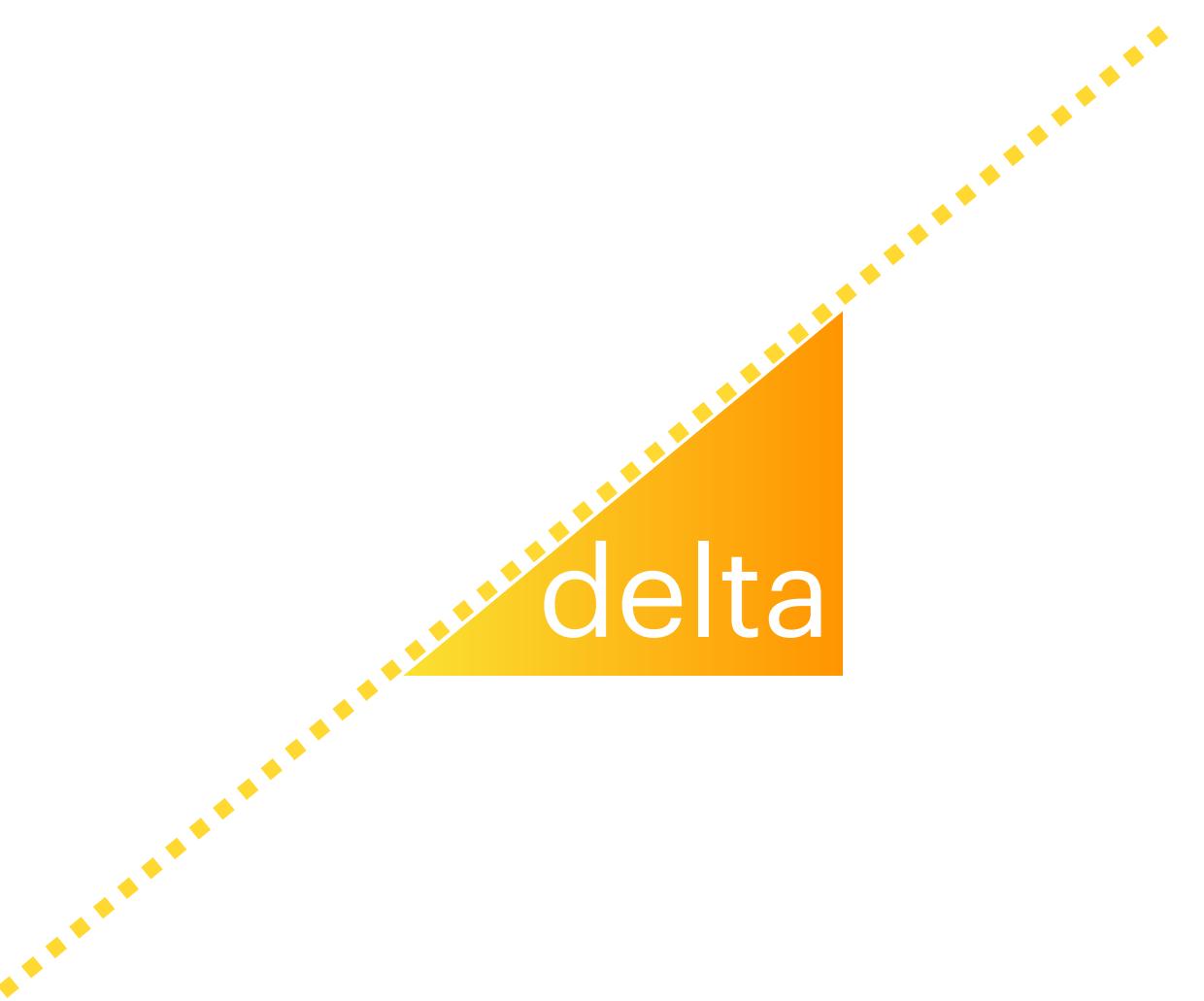
How much Y would change if we changed X

The slope at $(7, 49)$, $(8, 64)$ will be

$$(y_1 - y_2) / (x_1 - x_2) = (49 - 64) / (7 - 8) = 15$$

The slope at $(1, 1)$, $(2, 4)$ will be

$$(y_1 - y_2) / (x_1 - x_2) = (1 - 4) / (1 - 2) = 3$$

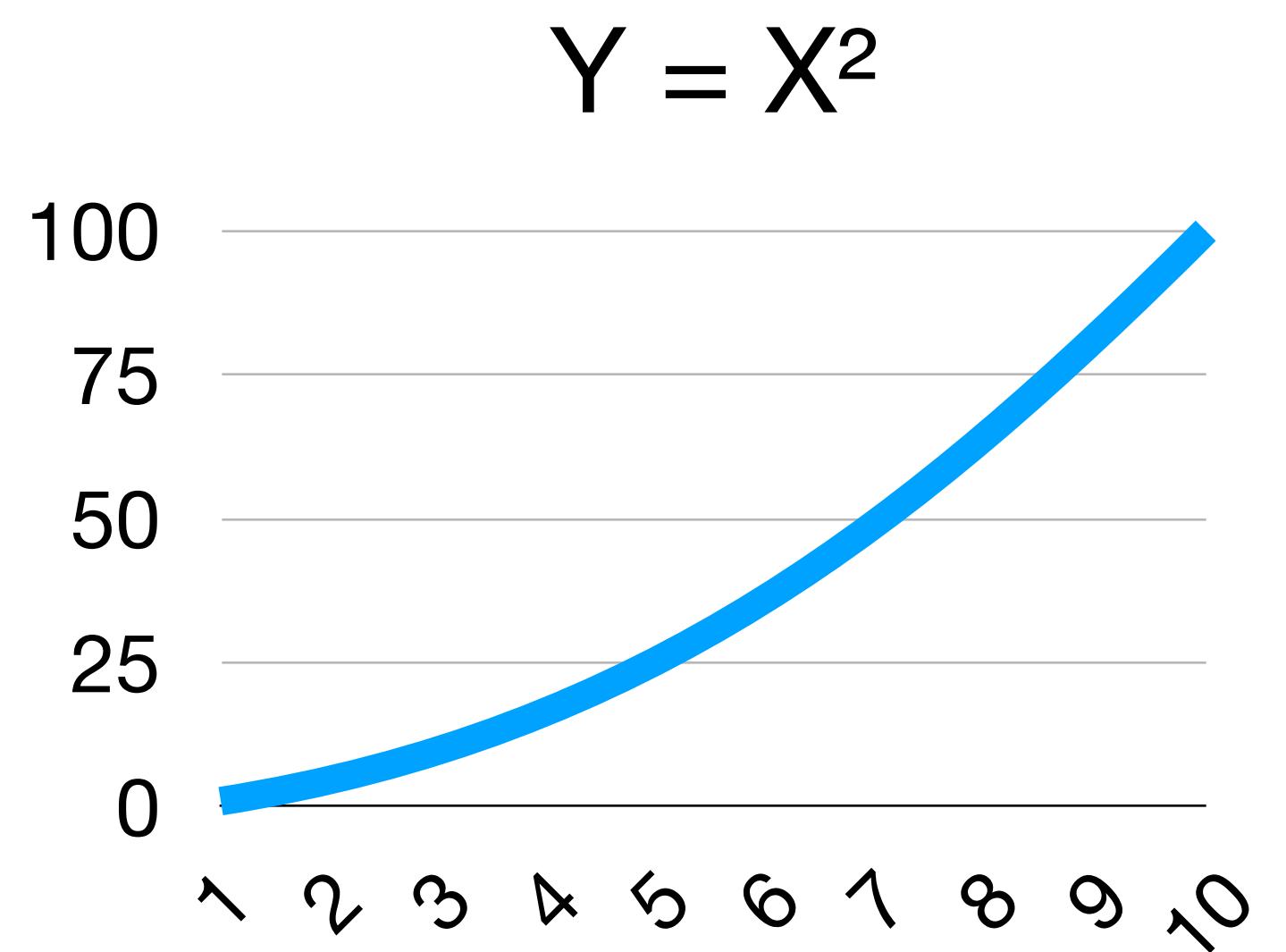


Recall differentiation

How much Y would change if we changed X

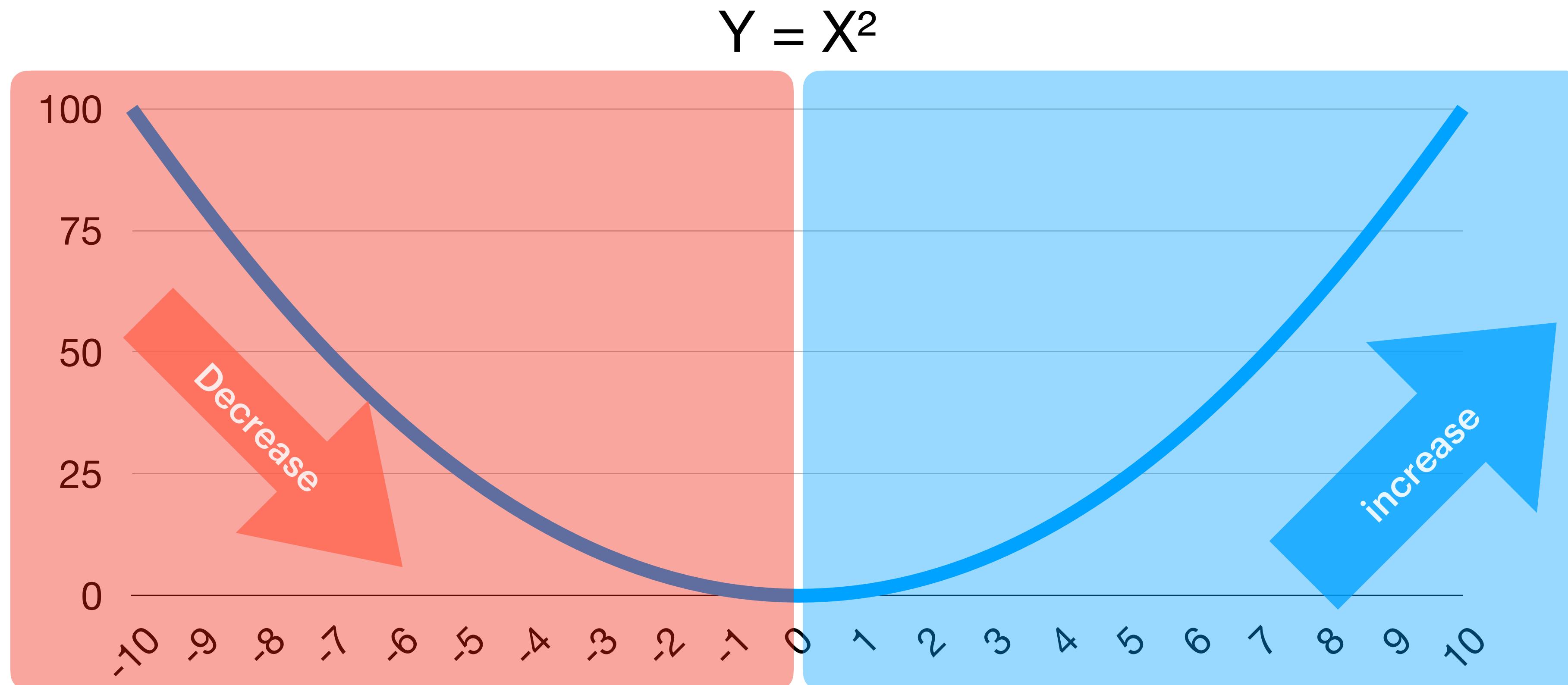
At any given point in the graph, what is
the slope ?

That is the direction where the function
increases



Recall differentiation

How much Y would change if we changed X



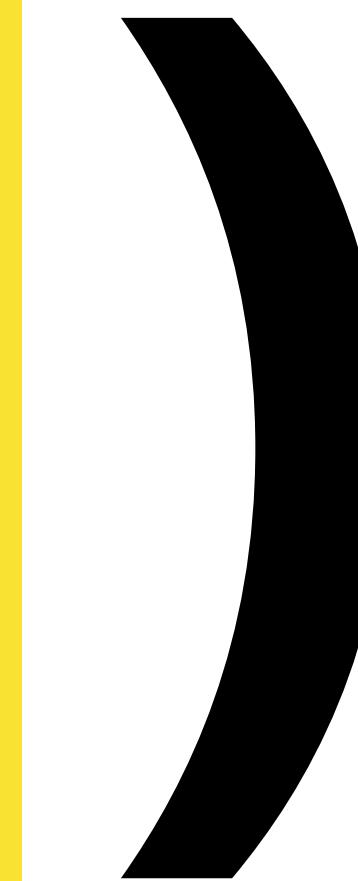
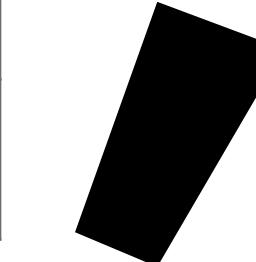
Back to the Error Function

Formulate the model

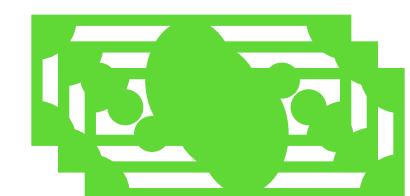
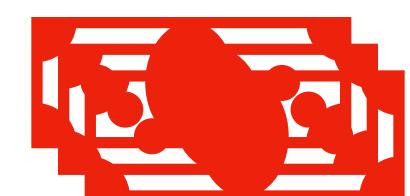
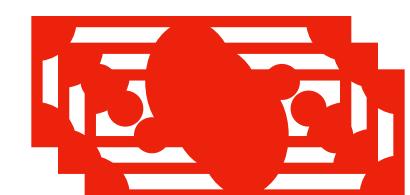
F (

feature_1	feature_2	...	feature_n
253	3	...	11
314	2	...	7
190	3	...	3
204	9	...	6

Weights



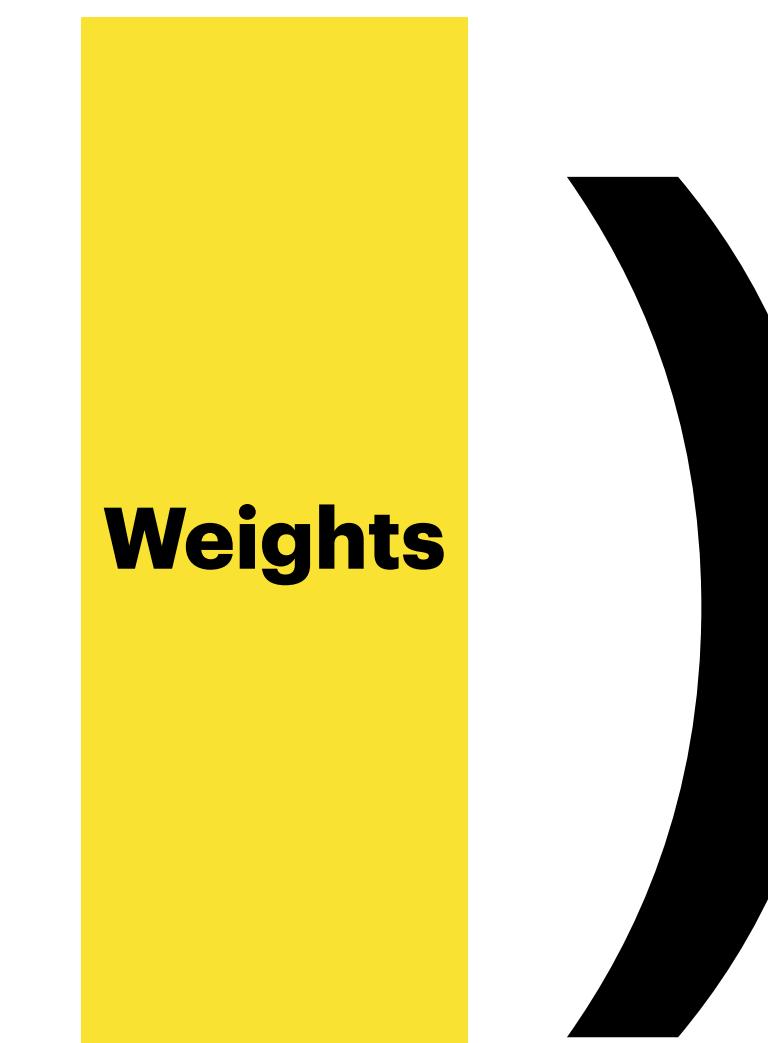
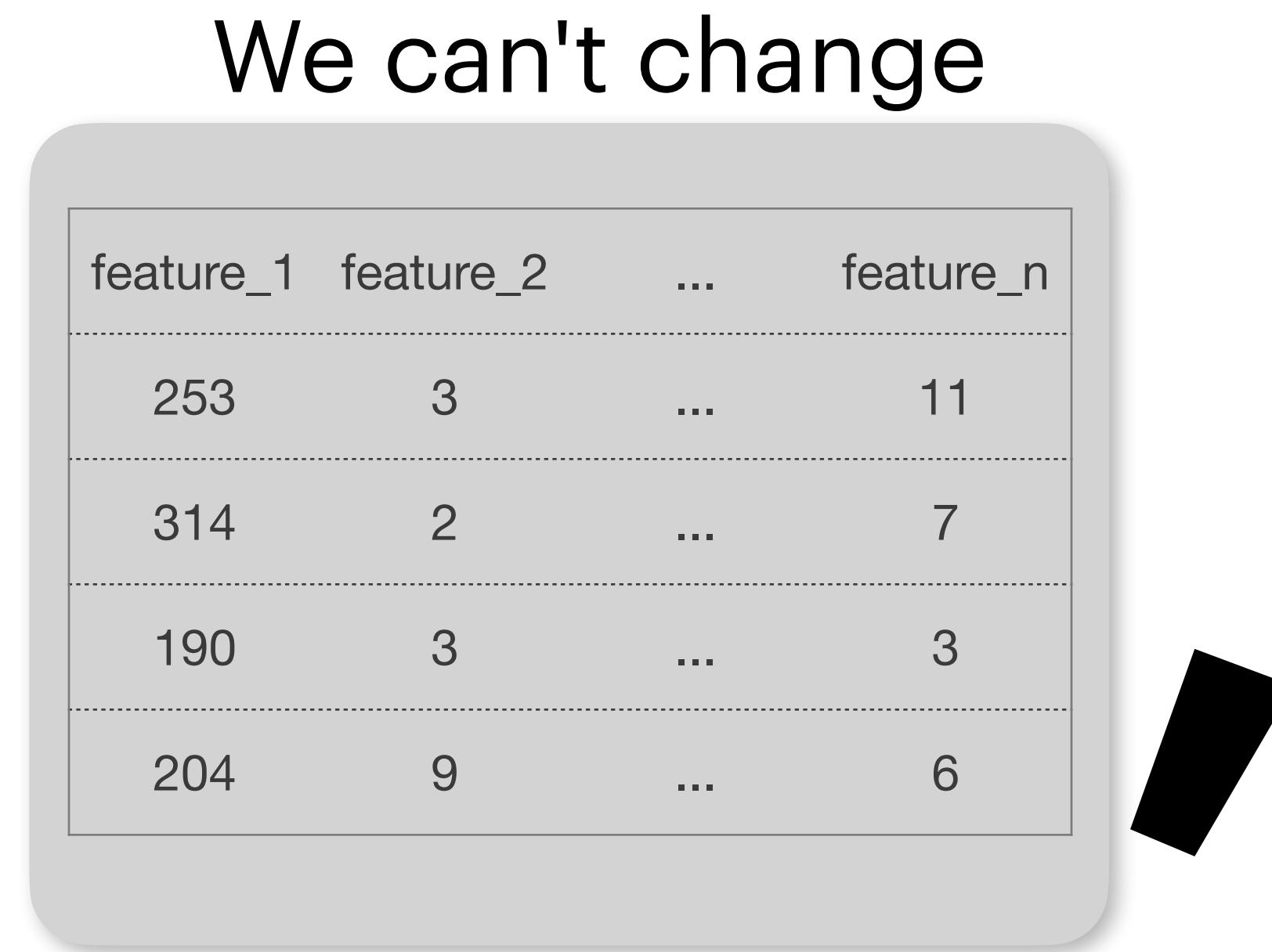
=



Our model takes weights and features and produce predictions

Formulate the model

F (



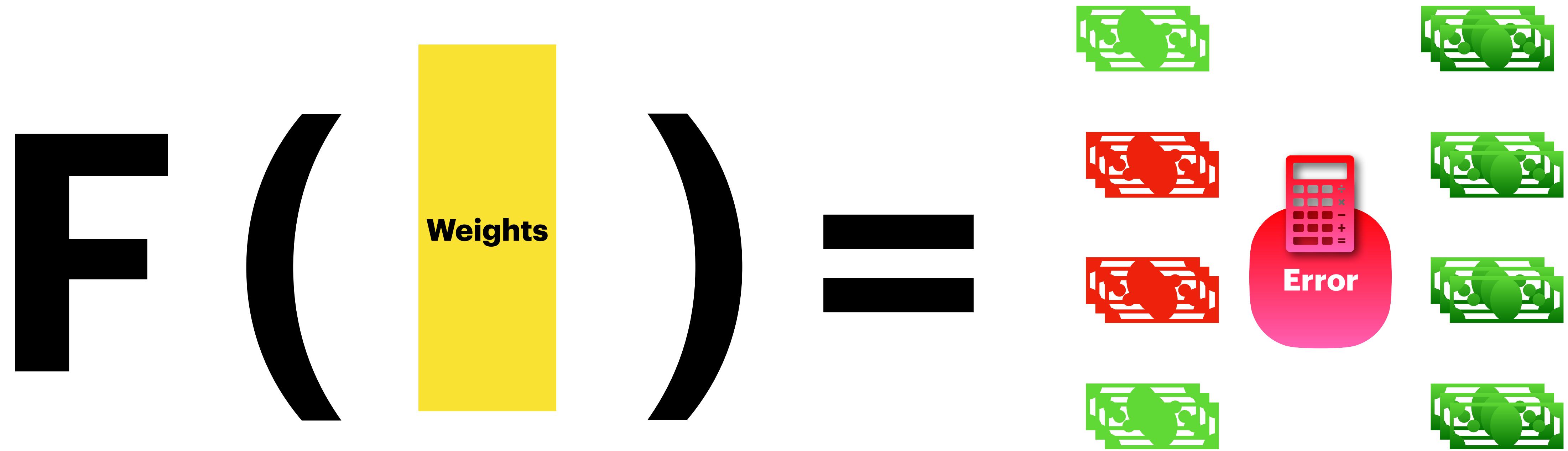
The feature can't be changed, we just apply the weights to it

Formulate the model

$$f(\text{Weights}) = \begin{matrix} \text{Green Car} \\ \text{Red Car} \\ \text{Red Car} \\ \text{Green Car} \end{matrix}$$

We can neglect feature values as they don't affect the error

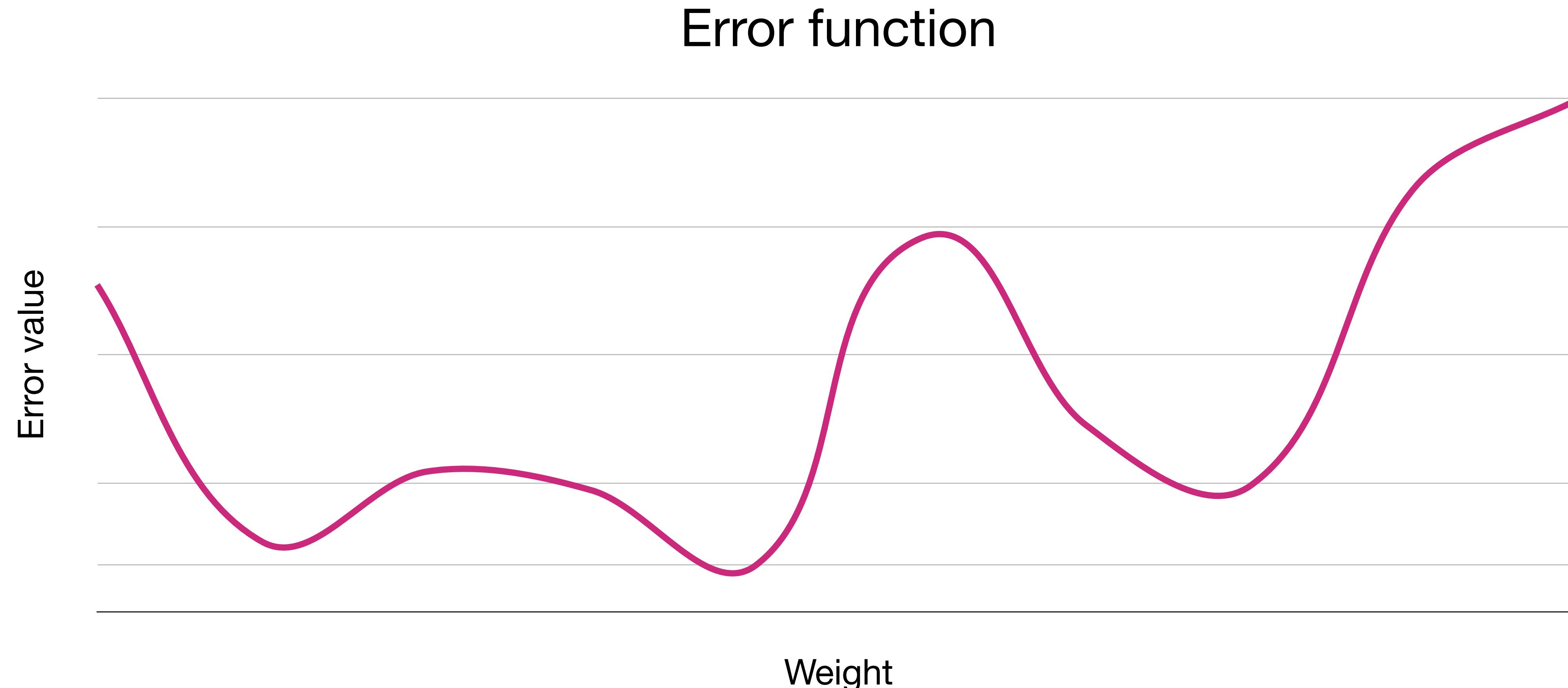
Formulate the model



We can say that the error is a function of the weights

Error for different weights

remember we want the error to be as little as possible



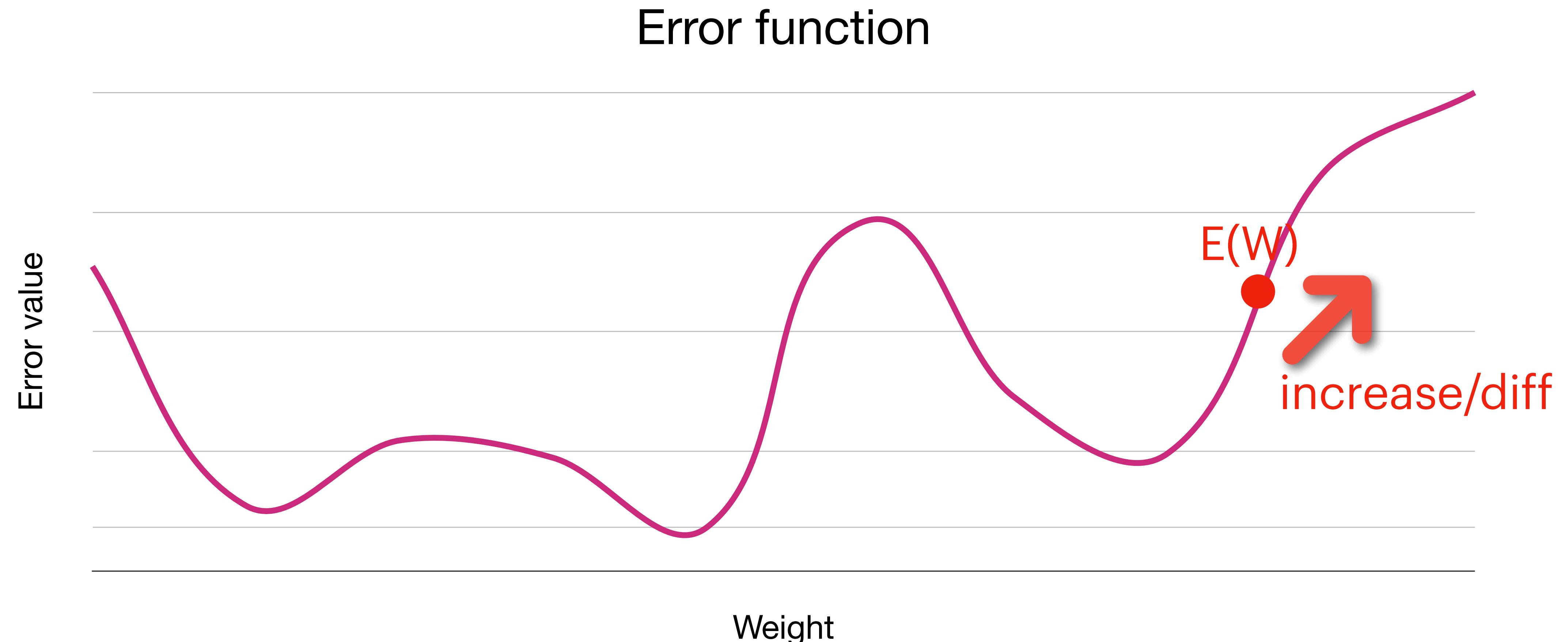
Error for different weights

remember we want the error to be as little as possible



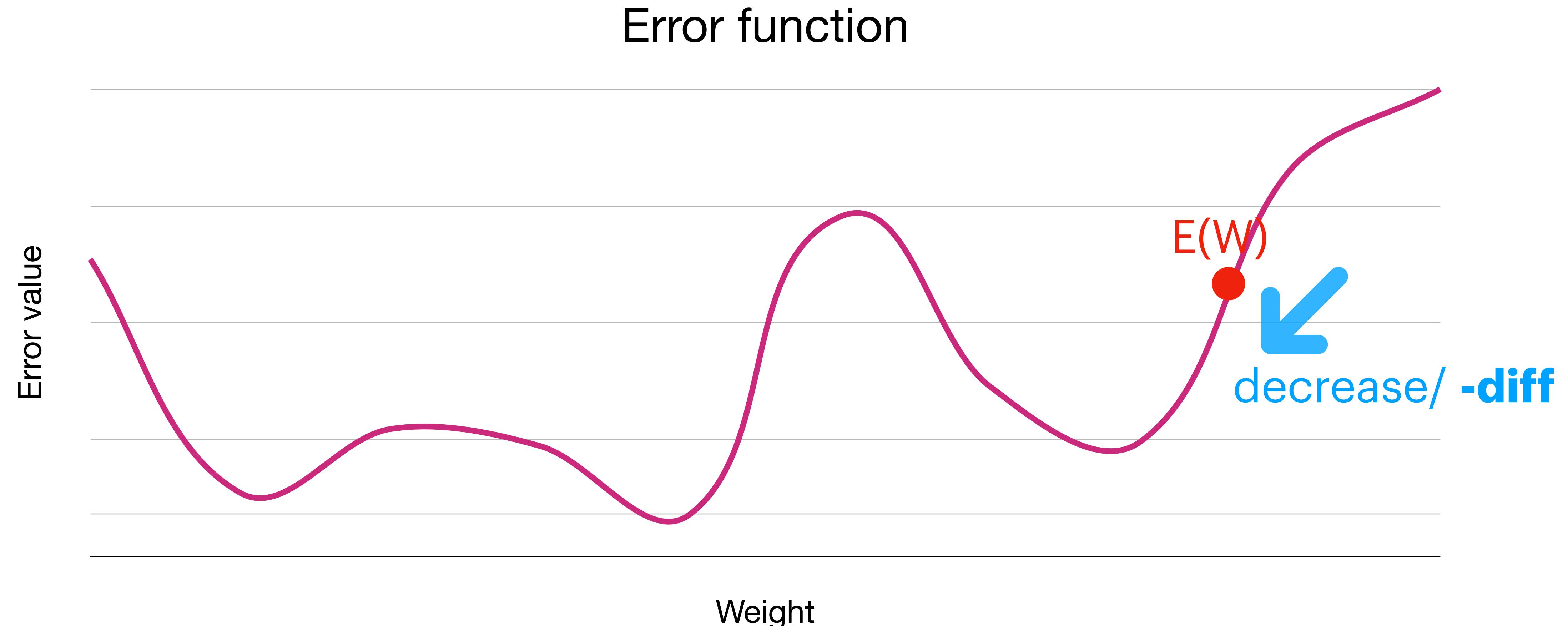
Error for different weights

remember we want the error to be as little as possible

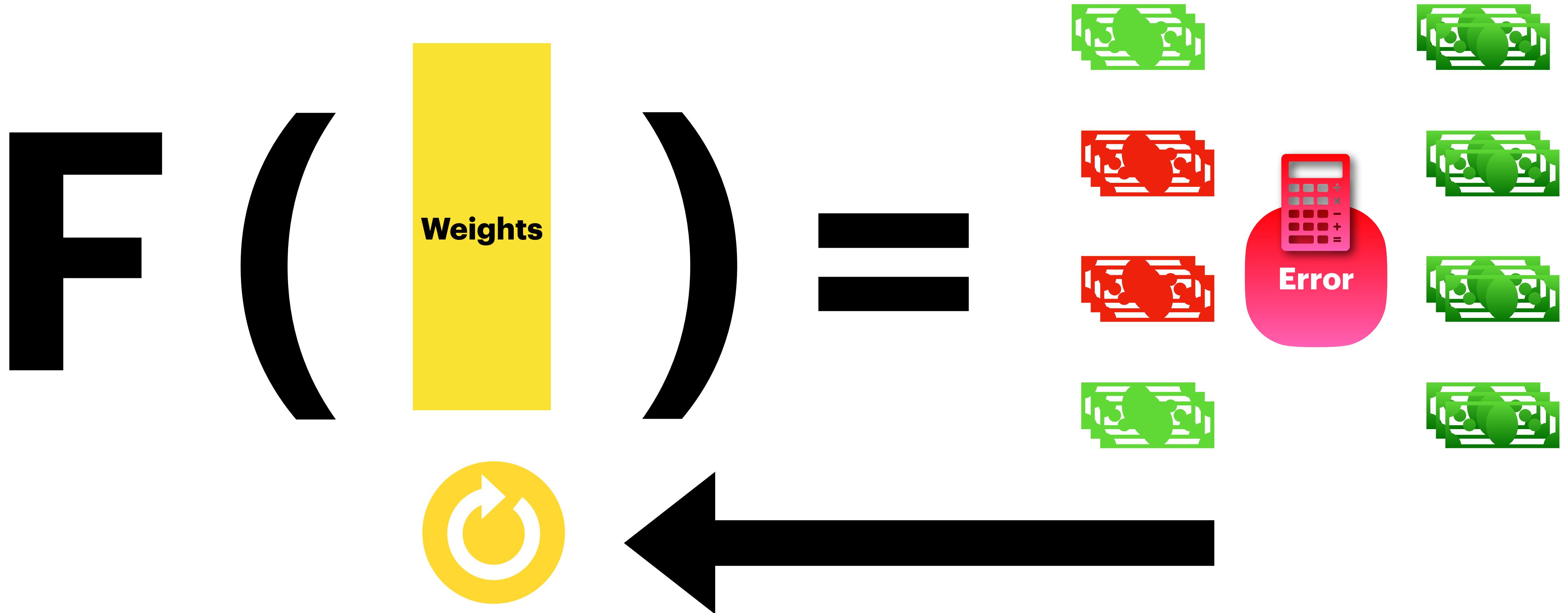


Error for different weights

remember we want the error to be as little as possible

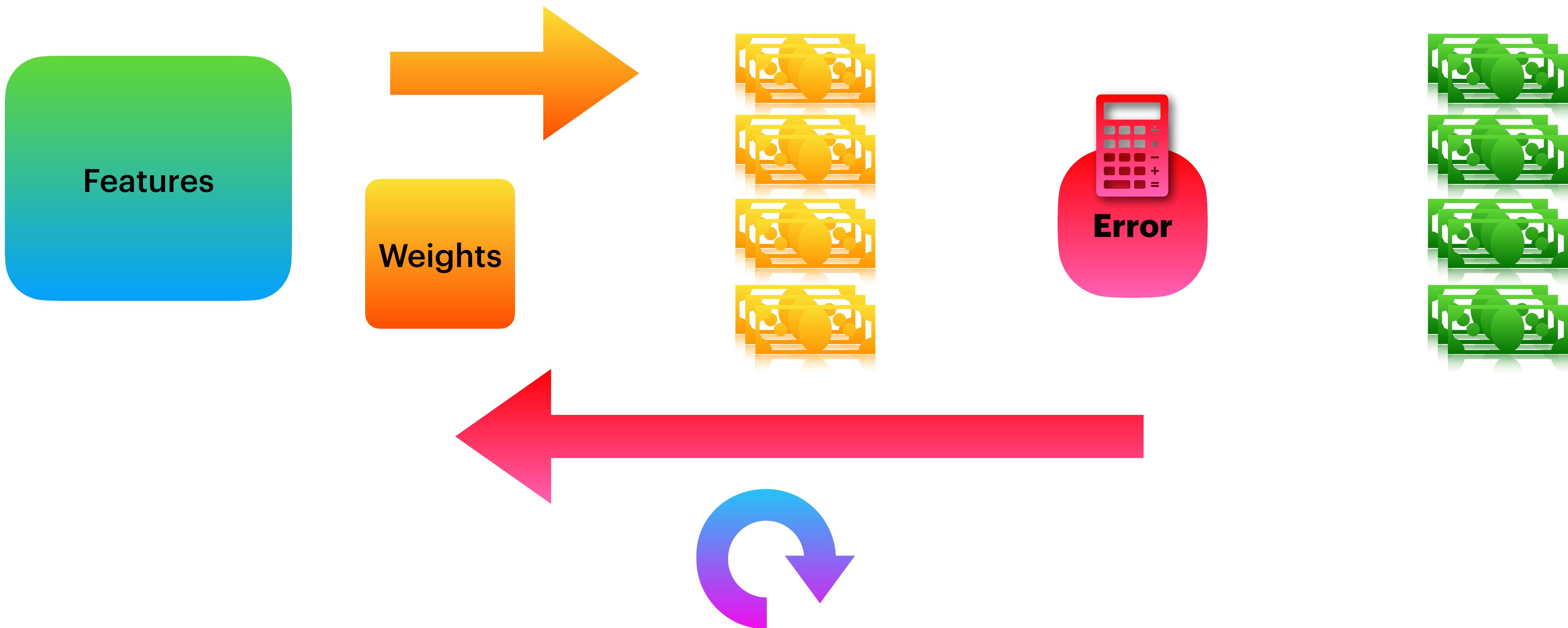


Gradient Descent



new weights = old weights - differentiation (negative slope)

Formulate the model



start with random weights, calculate the error, and update these weights

Machine learning model recap

- Choose the best features for your data
- Pick a suitable loss function
- apply gradient descent to minimize the loss function
- Stop when when your model is good enough

Line equation example

Line equation example

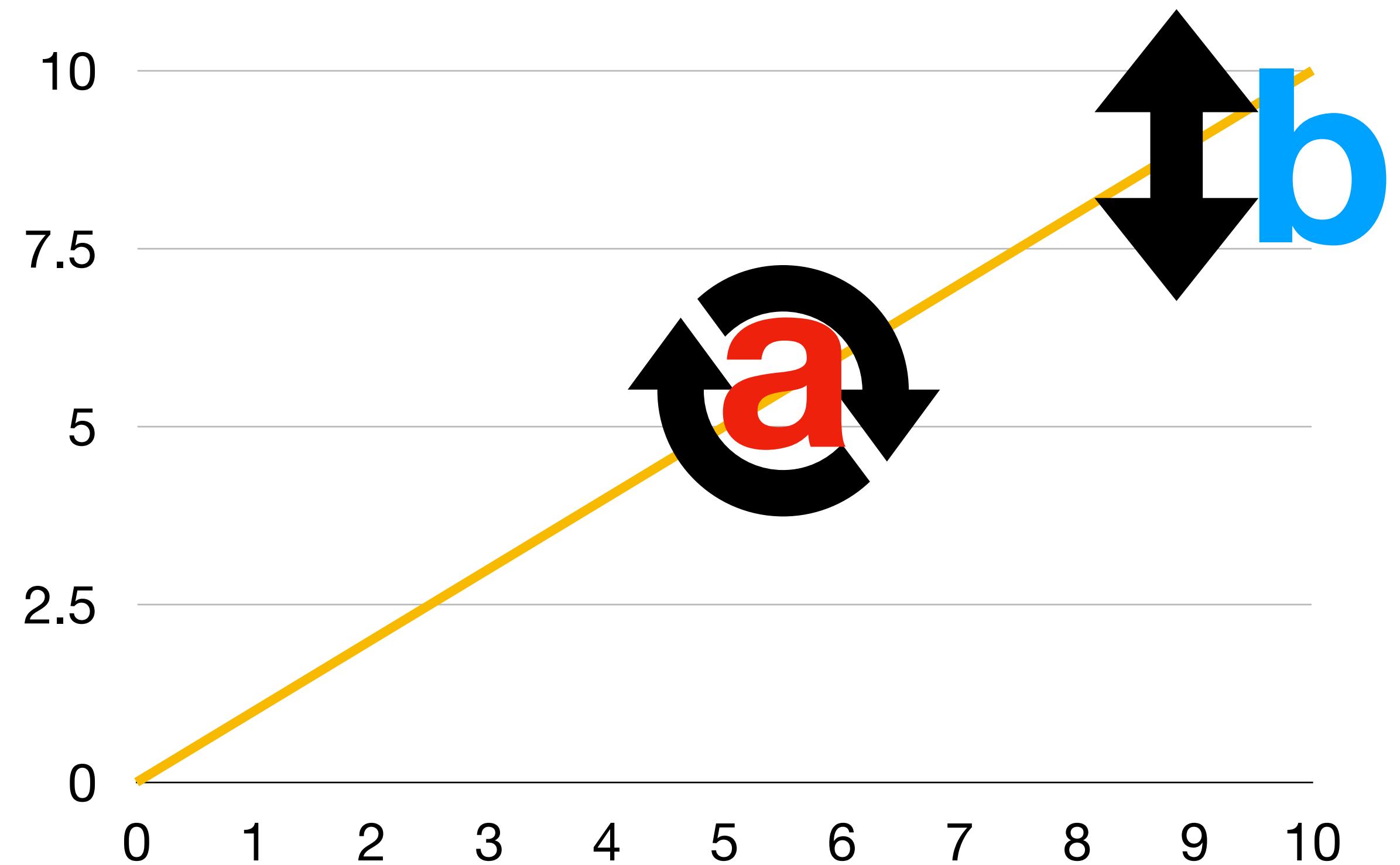
Think of weights as angels of freedom

$$Y = aX + b$$

Line equation example

Think of weights as angels of freedom

$$Y = aX + b$$



Time to code

Let's build a simple app

What's next ?

how should you start ?

- Refresh your mathematical background [here](#) or [here](#) or [here](#)
- Take Andrew Ng course on machine learning [here](#)
- Read the machine learning hands on book [here](#)
- Take Deeplearning.ai specialization on coursera [here](#)
- Practice and stay hungry for more!

A wide-angle photograph of a mountain range during a sunset or sunrise. The sky is filled with warm, orange, and yellow hues, transitioning into cooler blues at the top. Silhouettes of mountain peaks are visible against the bright sky. In the foreground, the dark silhouette of a person holding a walking stick is standing on a rocky outcrop. The overall atmosphere is peaceful and inspiring.

Stay safe.

Thanks !
feel free to contact me [@email](#) or [linkedIn](#) !