Gradient Descent

Linear Regression with Gradient Descent

Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function.

Linear Regression with Gradient Descent

Gradient descent is a first-order **iterative** optimization algorithm for finding the minimum of a function.

Linear Regression with Gradient Descent

Gradient descent is a first-order **iterative optimization** algorithm for finding the minimum of a function.

Linear Regression with Gradient Descent

Gradient descent is a first-order **iterative optimization** algorithm for finding the **minimum** of a **function**.

Linear Regression with Gradient Descent

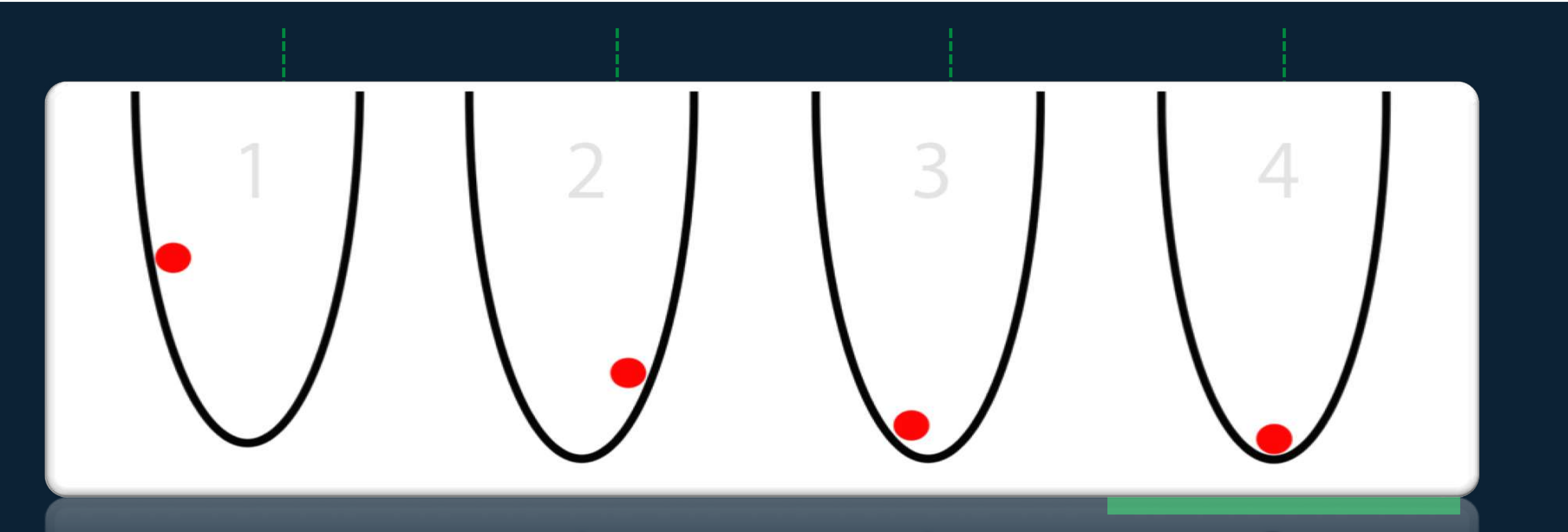
By tweaking **m and b,** we can create a line that will best describe the relationship. How do we know we're close? By using a thing called a cost function. It literally tells us the cost.

A high cost value means it's expensive — our approximation is far from describing the real relationship. On the other hand, a low cost value means it's cheap — our approximation is close to describing the relationship.

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

Linear Regression with Gradient Descent

- **Brute force isn't helpful**. A more efficient way is gradient descent. Imagine trying to find the lowest point blindfolded as can be seen below. What you would do is to check left and right and then feel which one brings you to a lower point. You do this every step of the way until checking left and right both brings you to a higher point.



Linear Regression with Gradient Descent

What we're doing here is applying partial derivatives with respect to both m and b to the cost function to point us to the lowest point.

derivative of zero means you are at either a local minima or maxima

Which means that the closer we get to zero, the better.

$$\frac{\partial}{\partial m} = \frac{2}{N}\sum_{i=1}^{N} -x_i(y_i - (mx_i + b))$$

$$\frac{\partial}{\partial b} = \frac{2}{N}\sum_{i=1}^{N} -(y_i - (mx_i + b))$$

What is partial derivative?

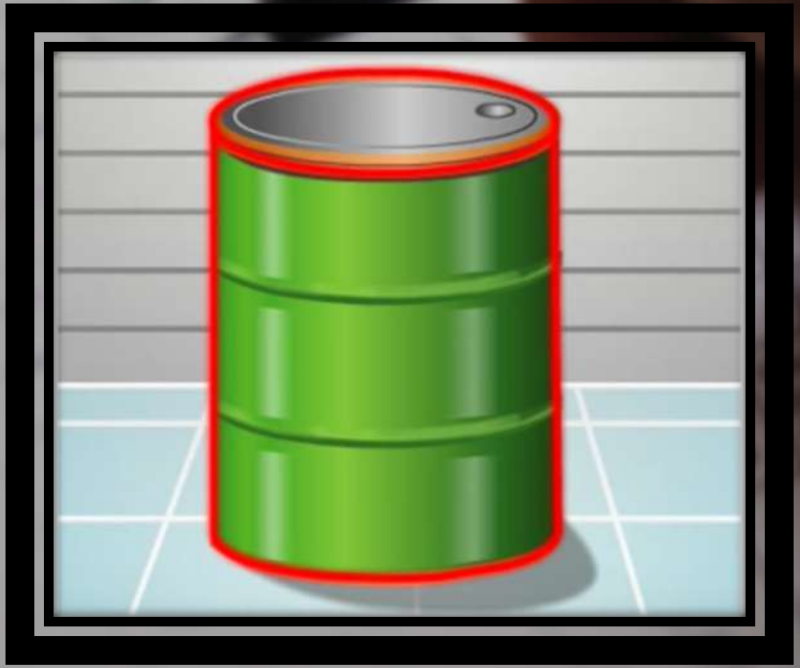
- Let us understand the function of several variables
- Volume of cylinder

$$V = \pi r^2 h$$

Where,

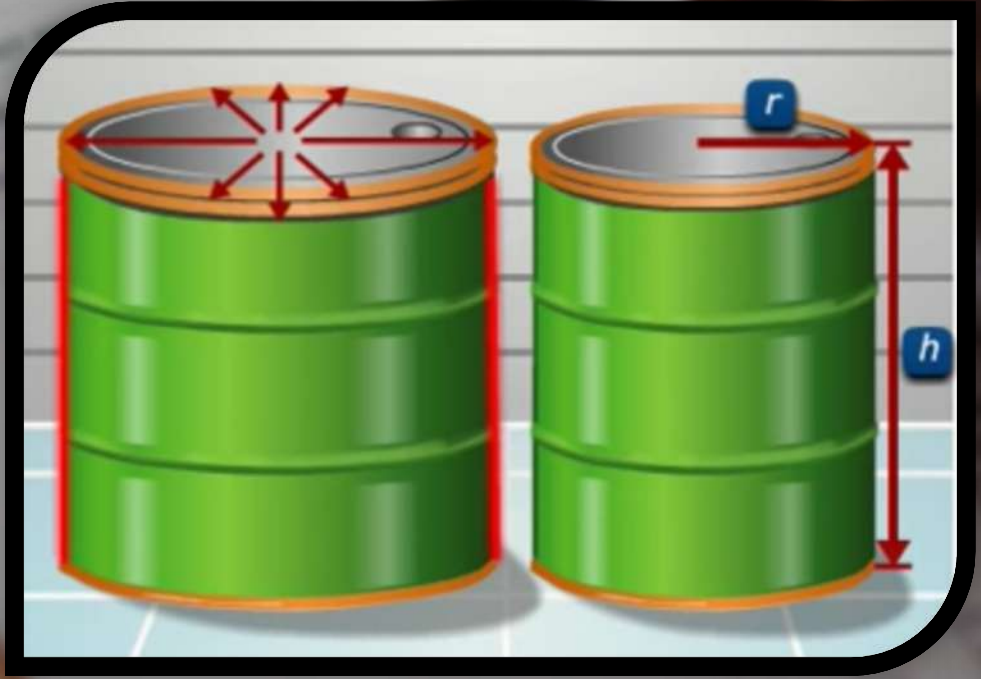
r is the radius of the cylinder

h is the height of the cylinder



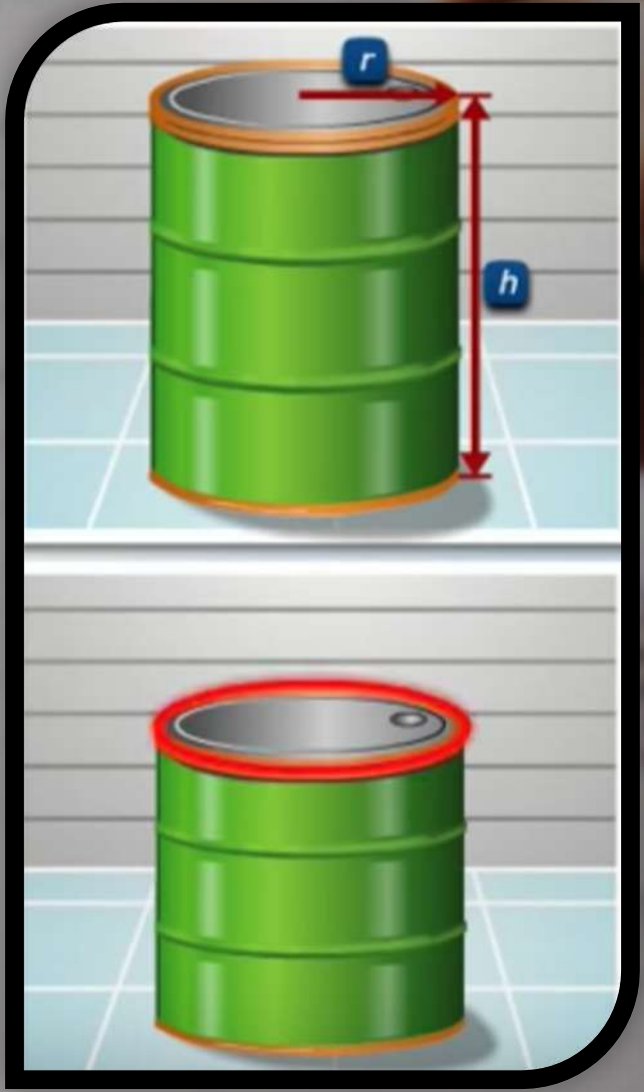
What is partial derivative?

- When we change value of r no Change in h



What is partial derivative?

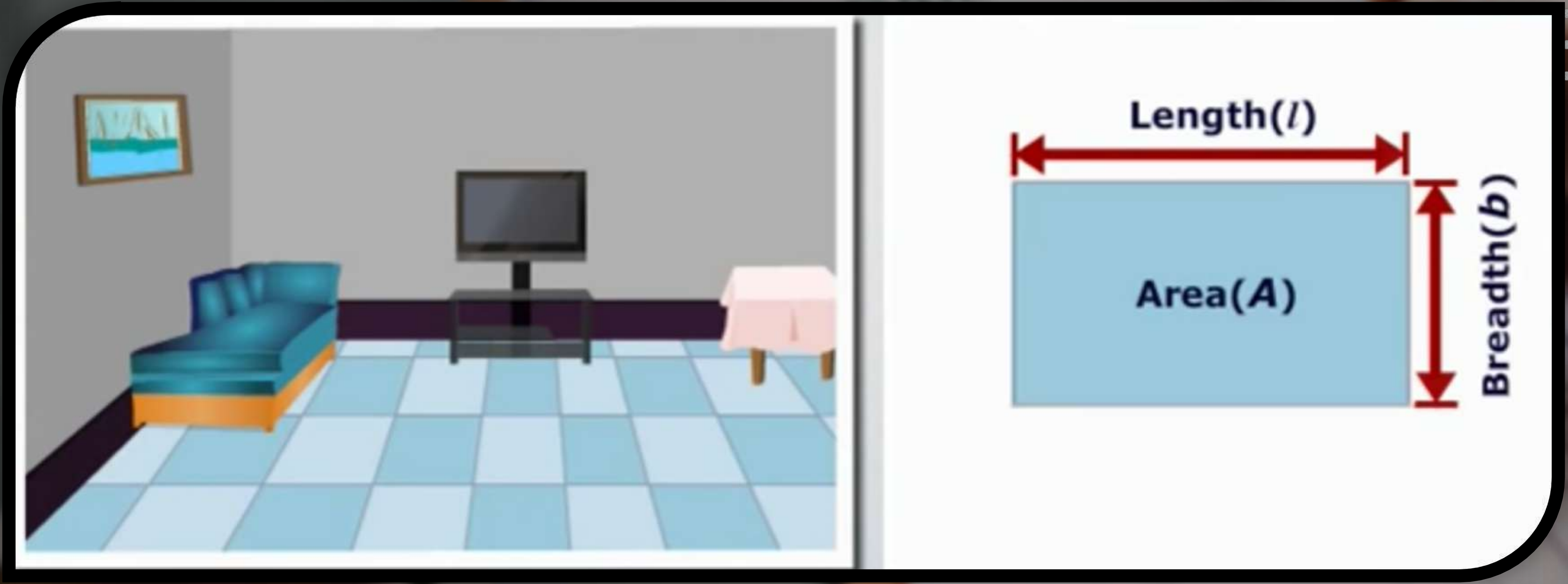
- When we change value of h no Change in r



What is partial derivative?

- Therefore we see that
- If r changes, is no change in h and vice versa.
- Therefore r and h are independent variables
- Therefore, V is an example of **Function of several variables**

What is partial derivative?



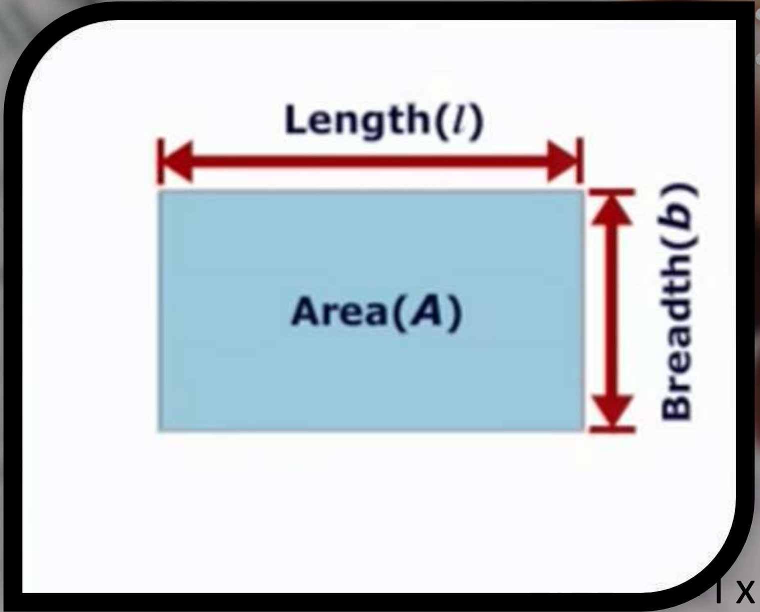
What is partial derivative?

Area, A = l x b

Rate of change of Area wrt Length

Area, A = l x b (we will make b as constant)

$$\text{Partial Derivative} = \frac{\partial A}{\partial l}$$



l x b

PARTIAL DIFFERENTIATION is a method to *Differentiate a Function* with respect to *One Independent Variable* while Treating the *Other Variables as Constant*. It is represented as

$$\text{Partial Derivative} = \frac{\partial(\text{function})}{\partial(\text{Independant Variable})}$$

Linear Regression with Gradient Descent

Step 1

Substitute m and b to zero

Step 2

Find the prediction and cost function. ie. summation(y-(mx+b))^2

Step 3

Find b_gradient and m_gradient by substituting in partial derivative function

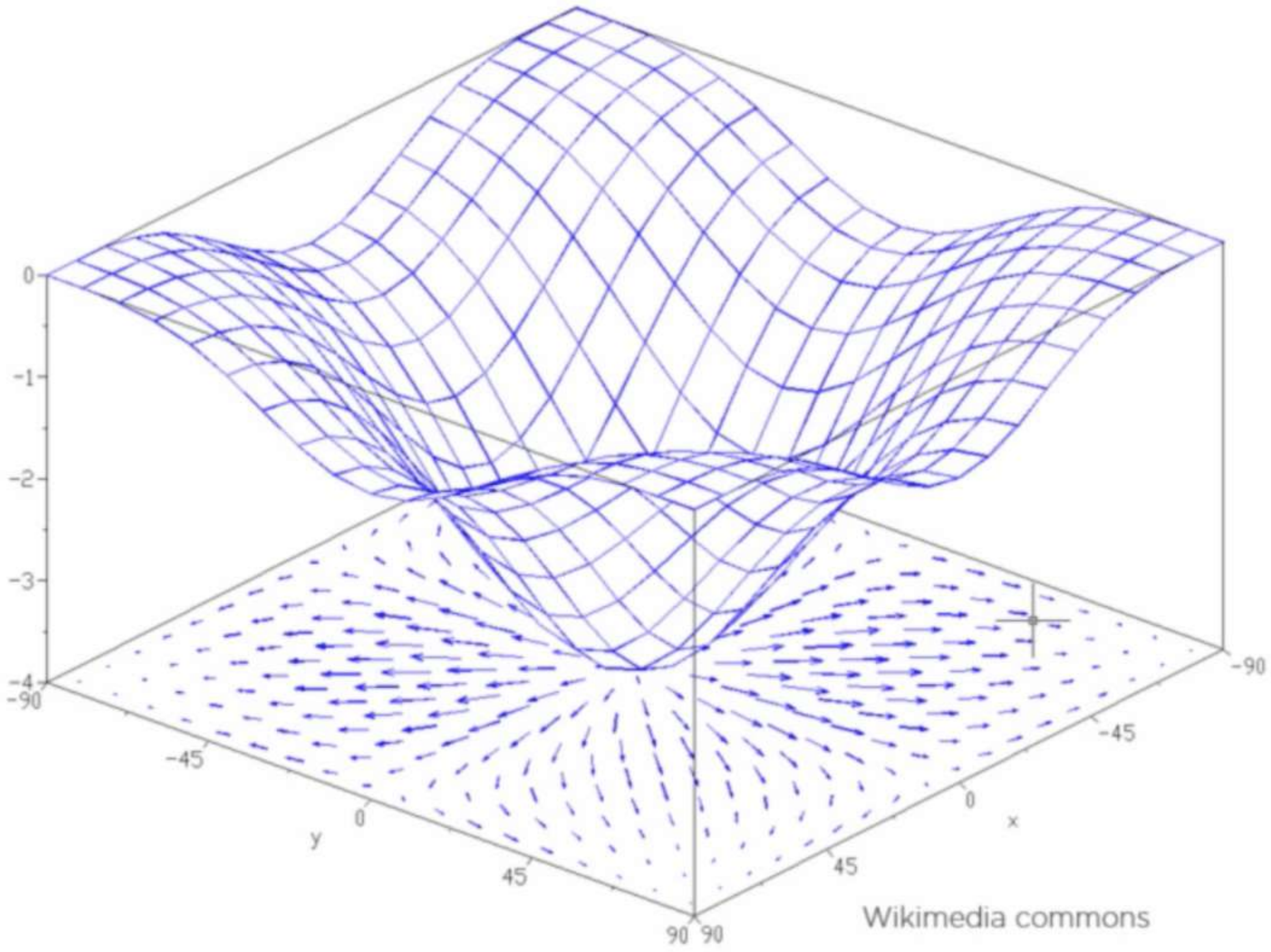
Step 4

Update b and m by with respective to gradient value found in step 3 with learning rate

Step 5

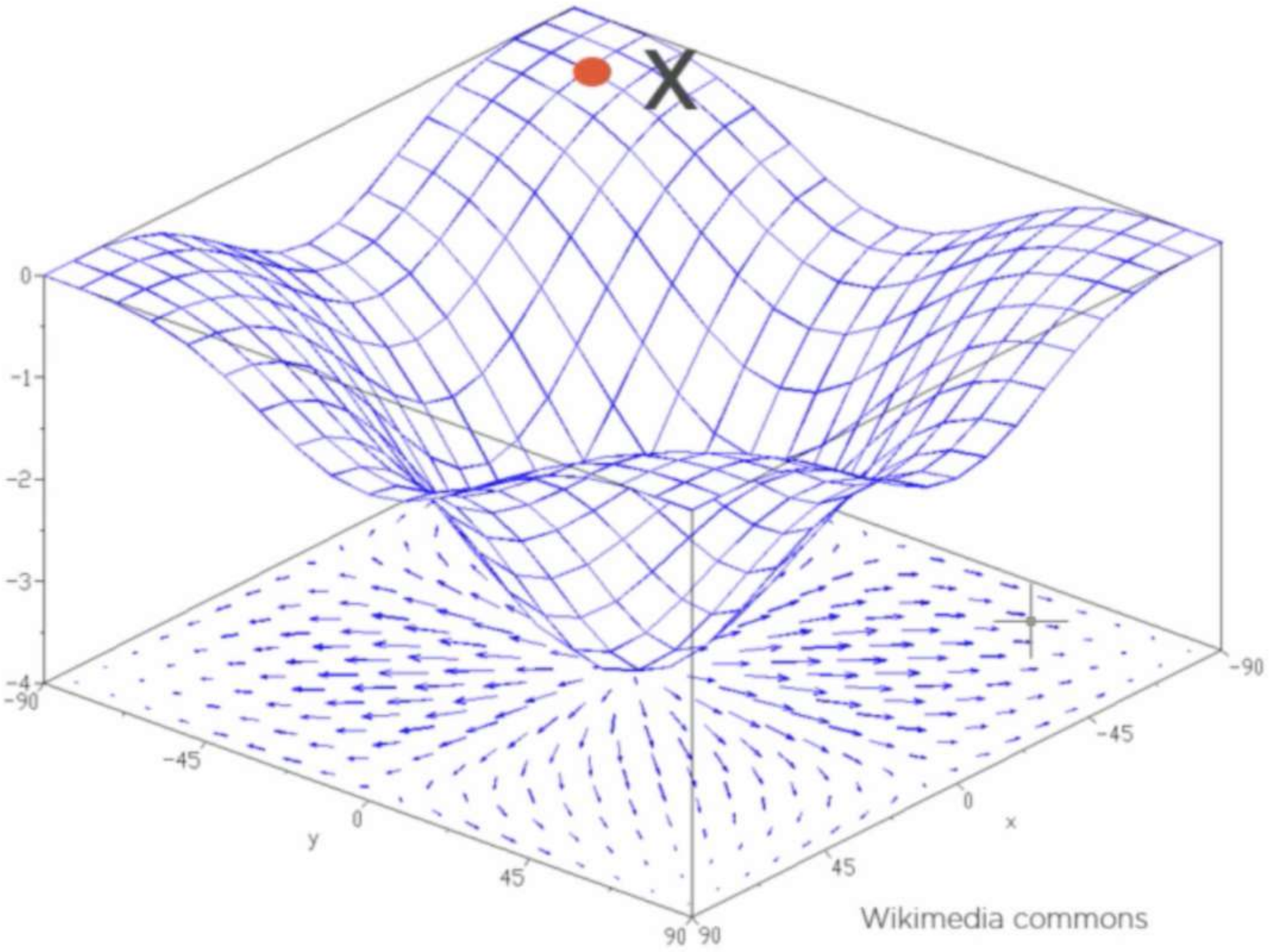
Find the cost function and repeat step 2 to 4

Gradient Descent



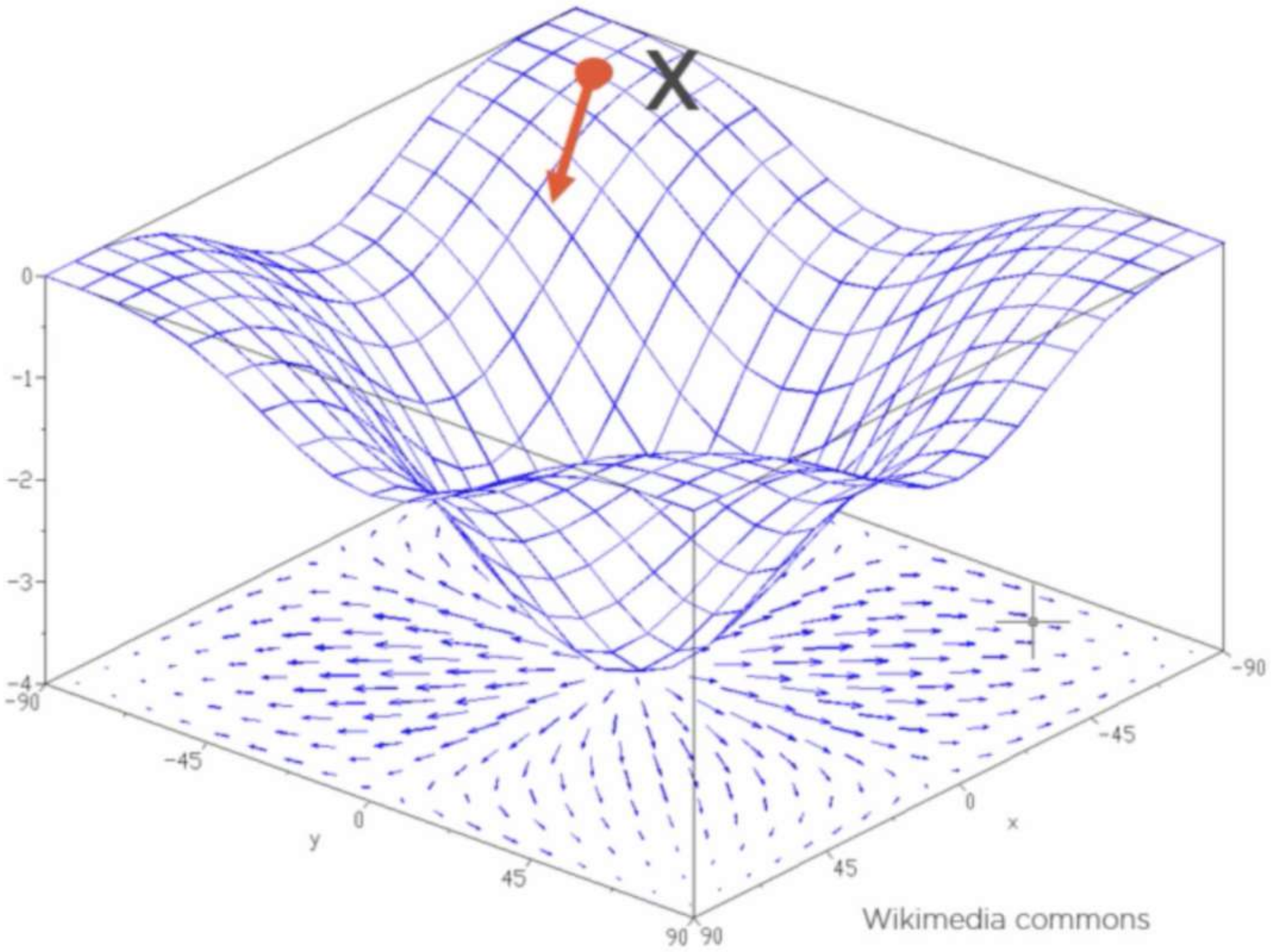
Wikimedia commons

Gradient Descent



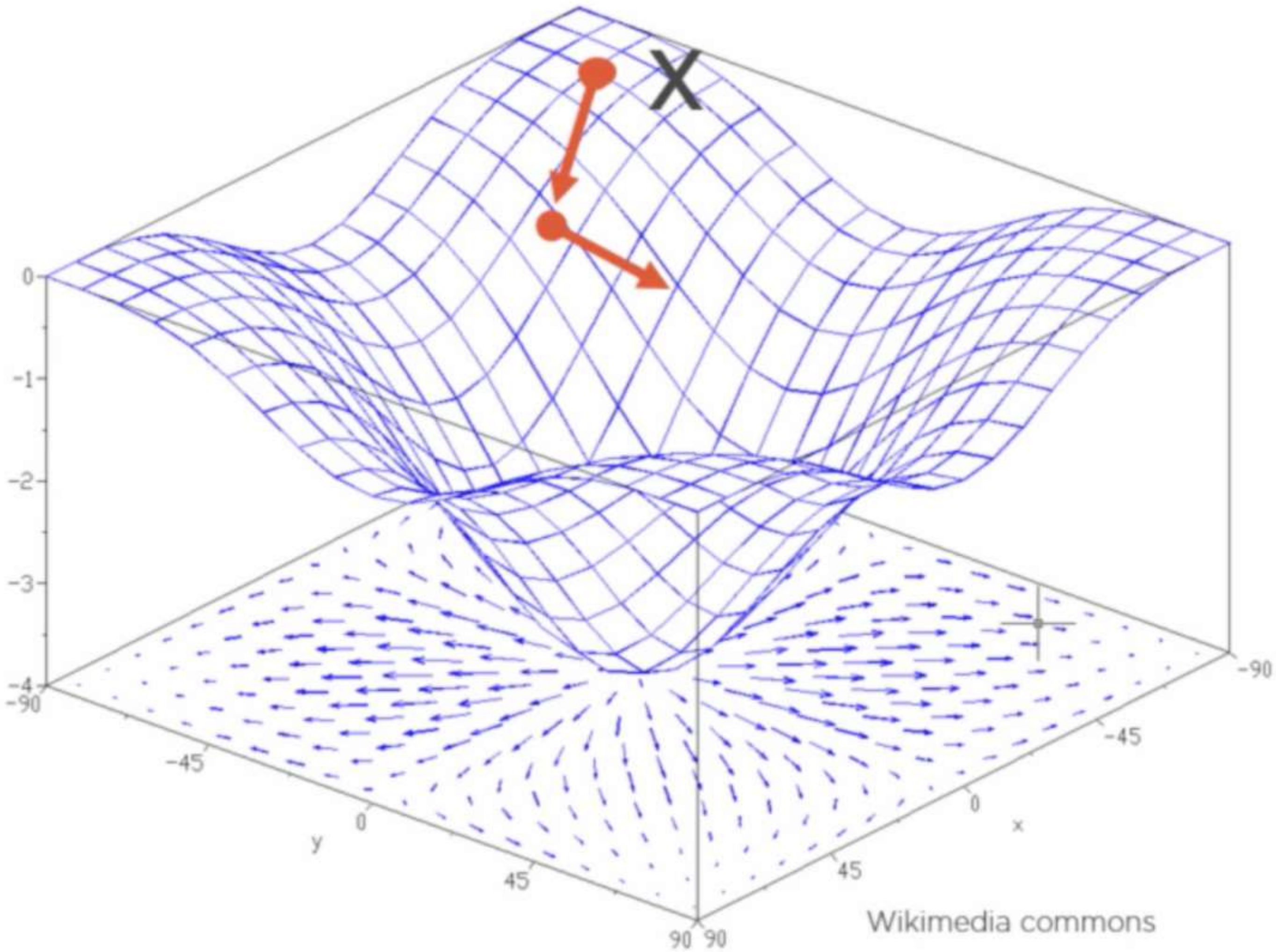
Wikimedia commons

Gradient Descent



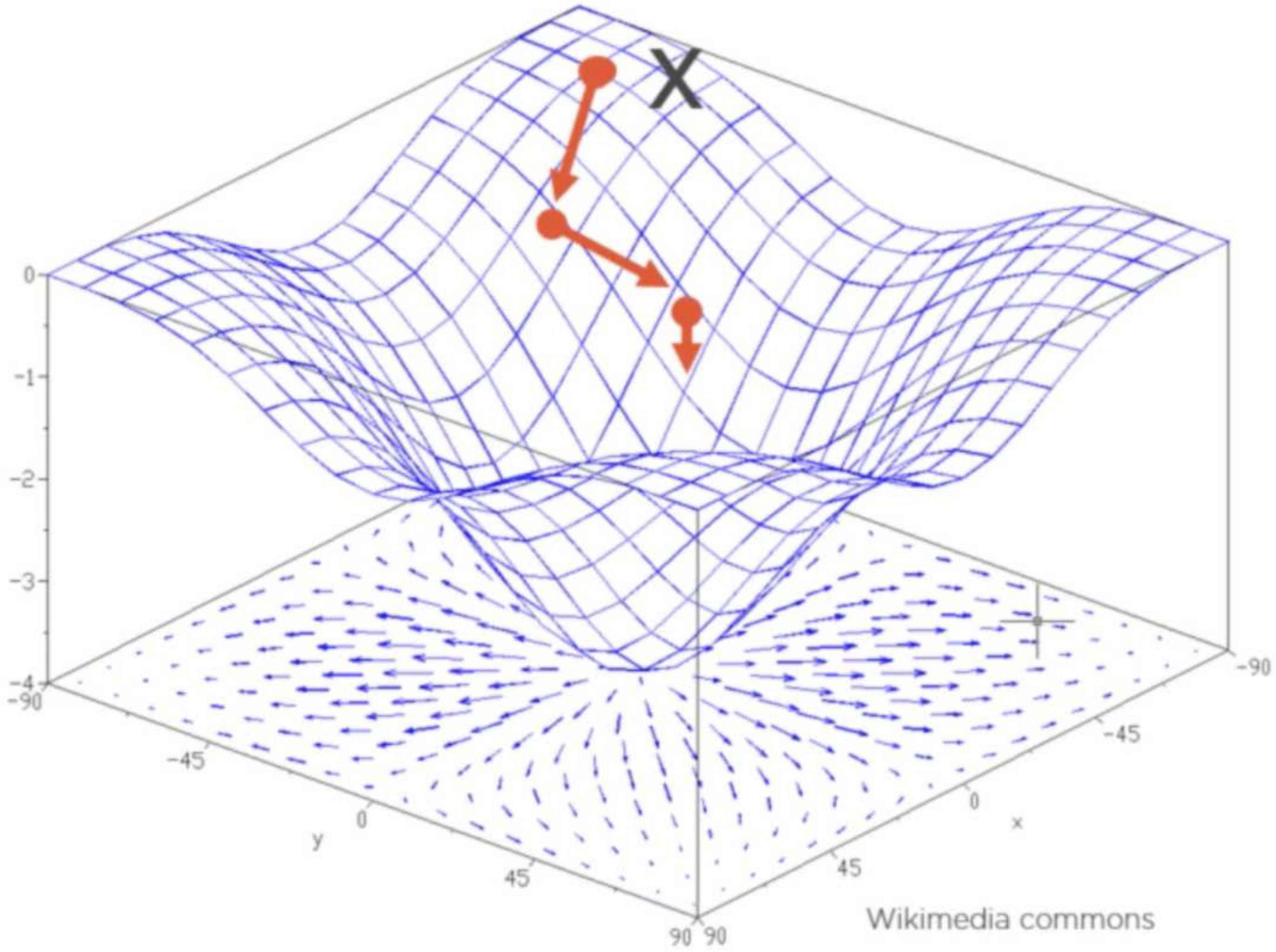
Wikimedia commons

Gradient Descent



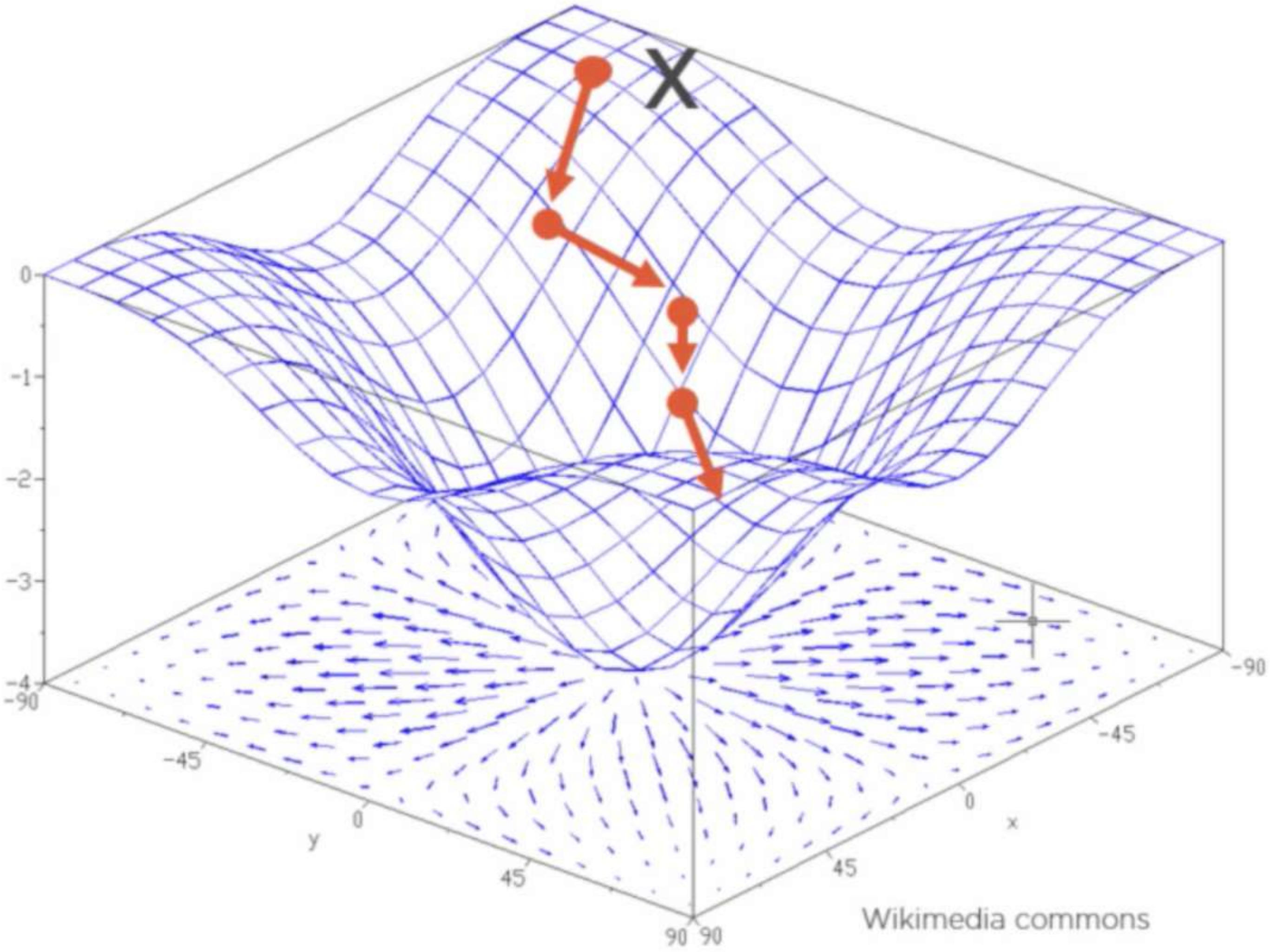
Wikimedia commons

Gradient Descent



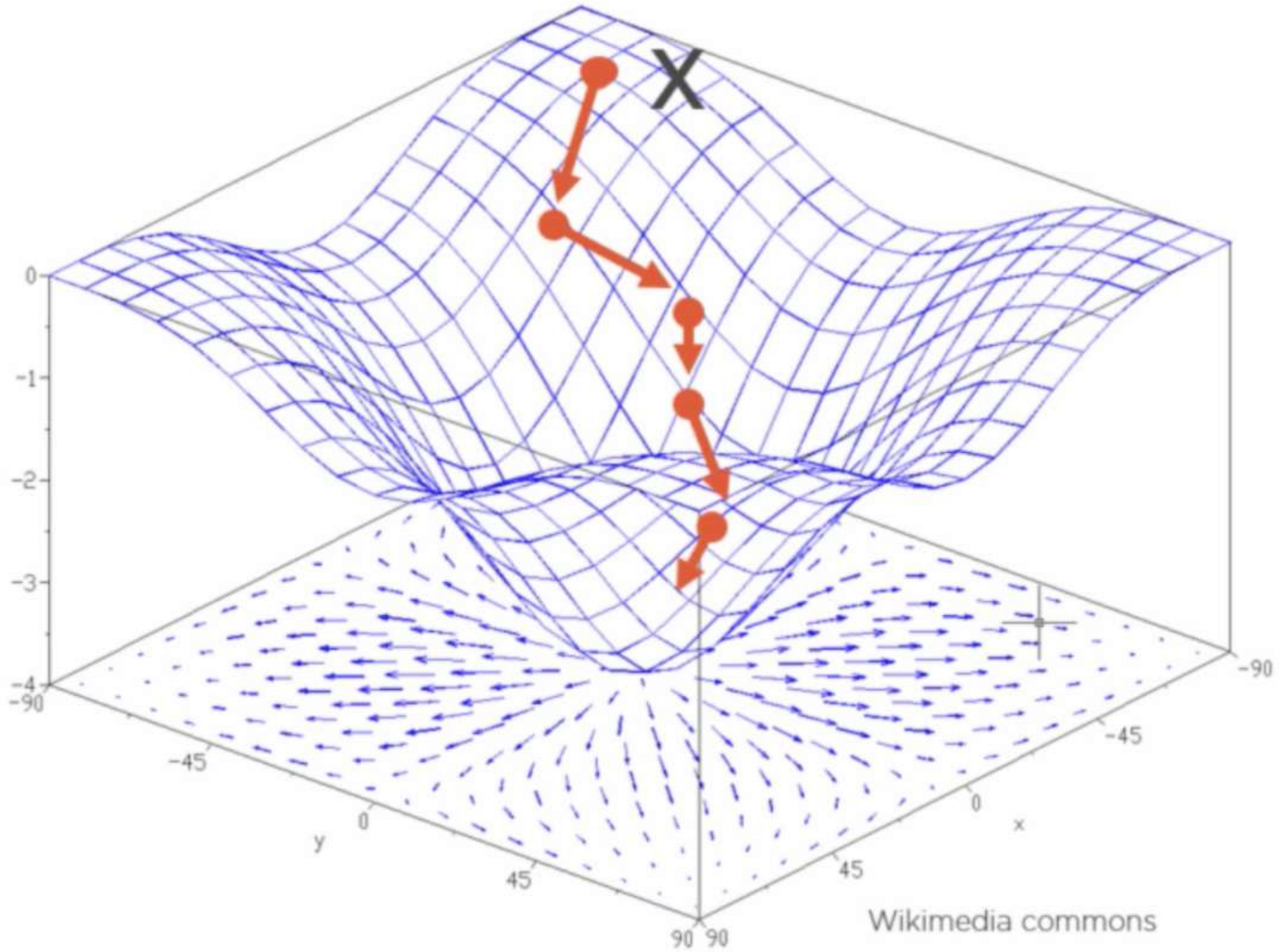
Wikimedia commons

Gradient Descent



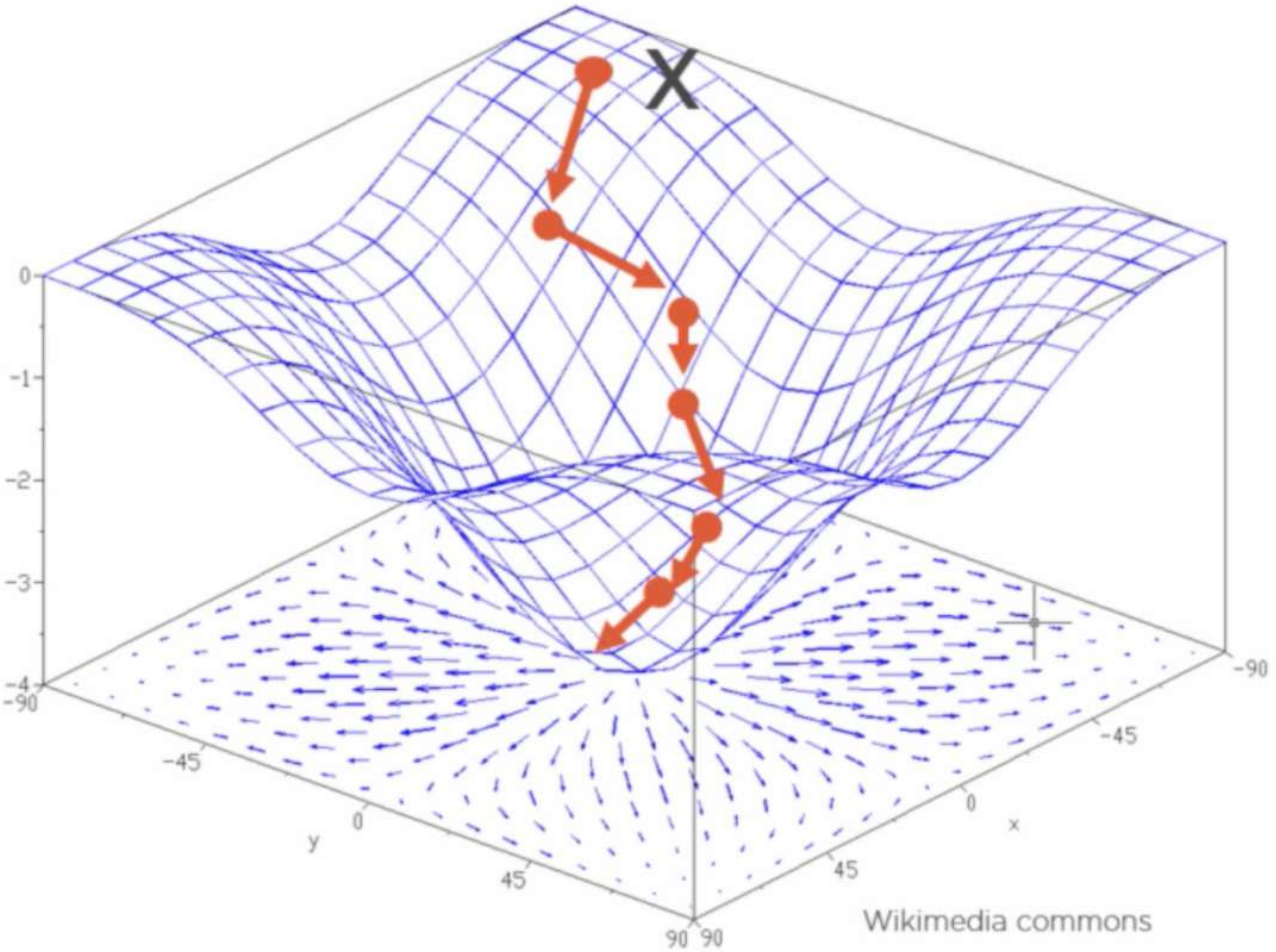
Wikimedia commons

Gradient Descent

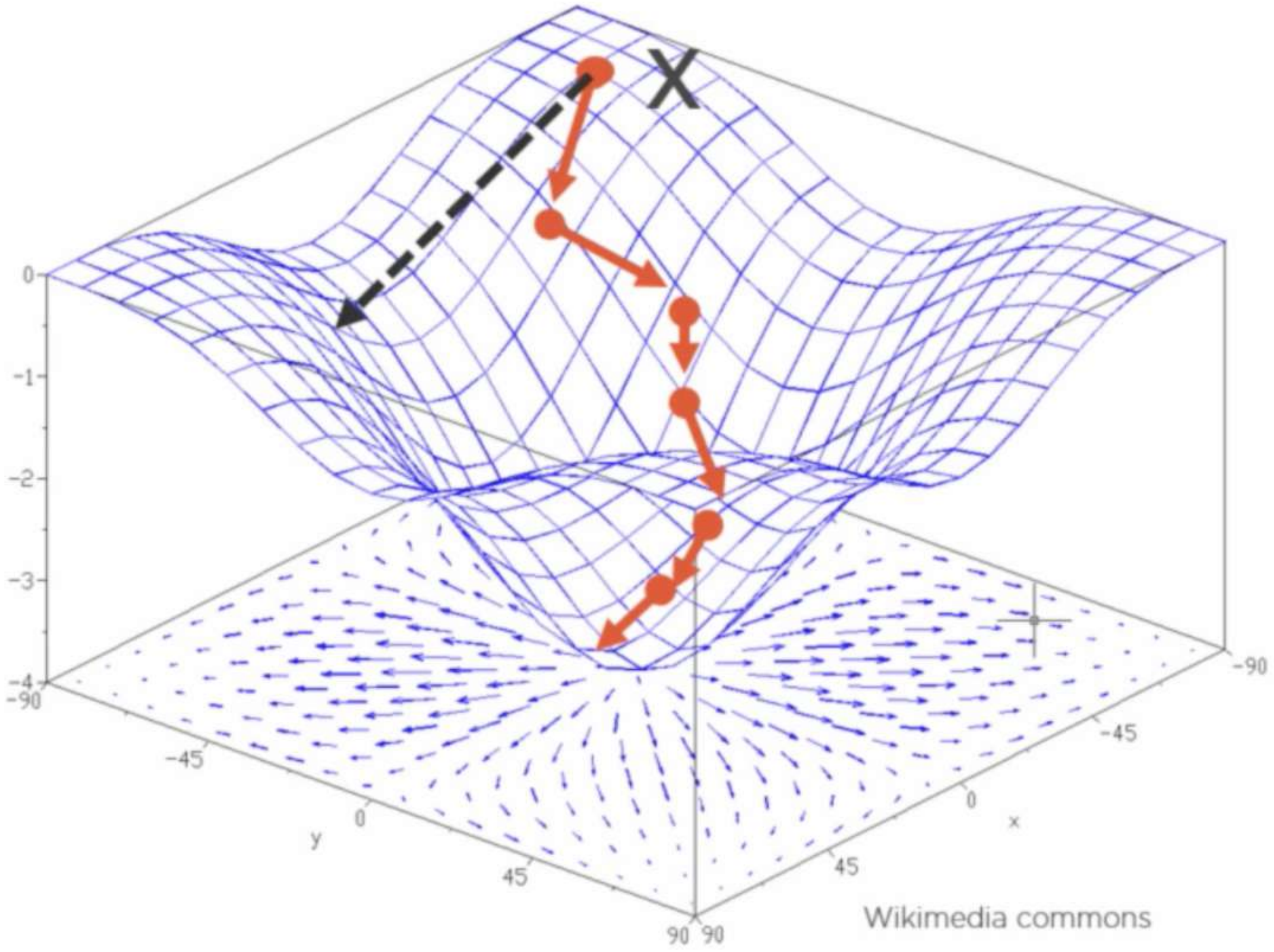


Wikimedia commons

Gradient Descent

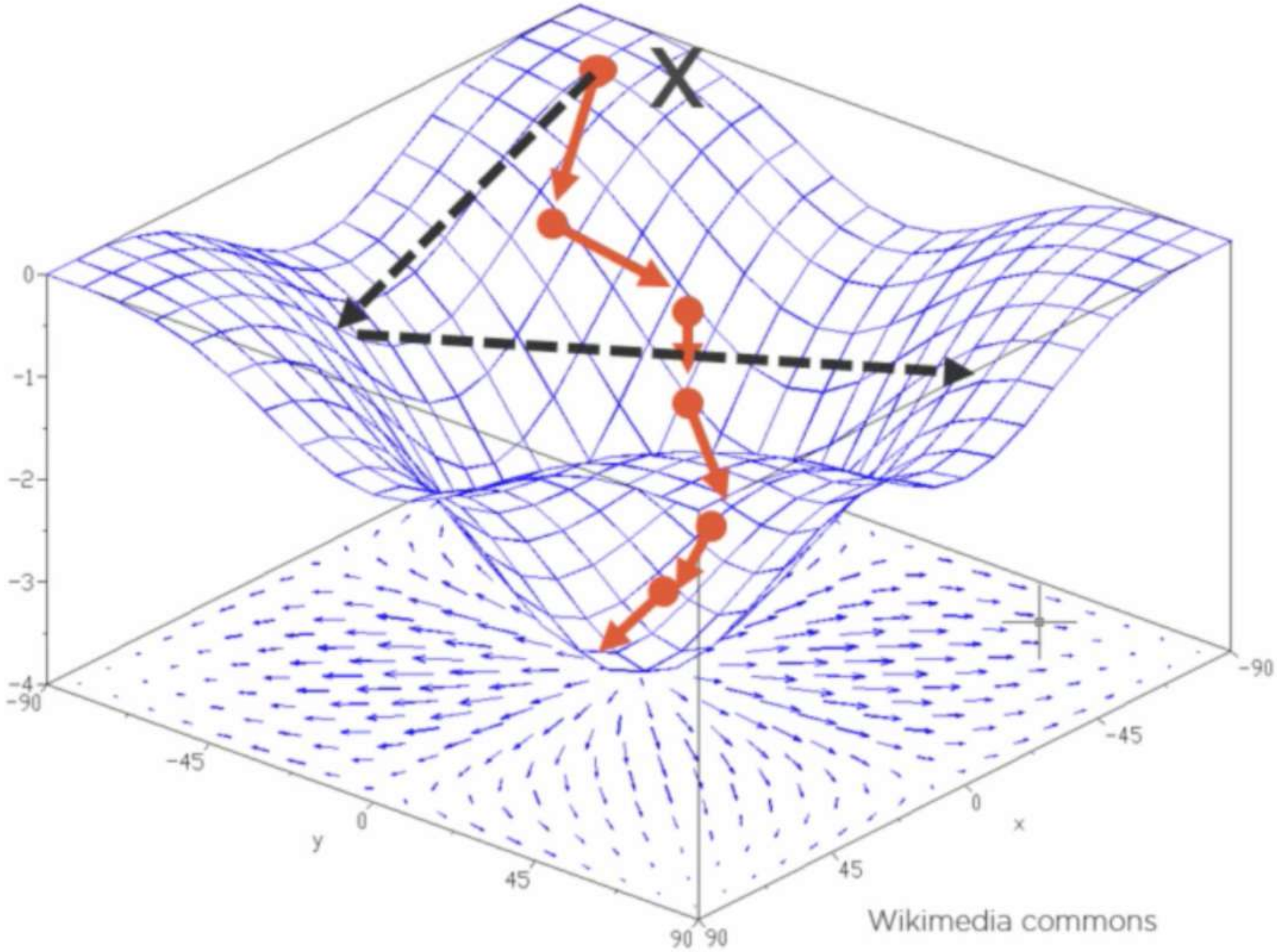


Gradient Descent



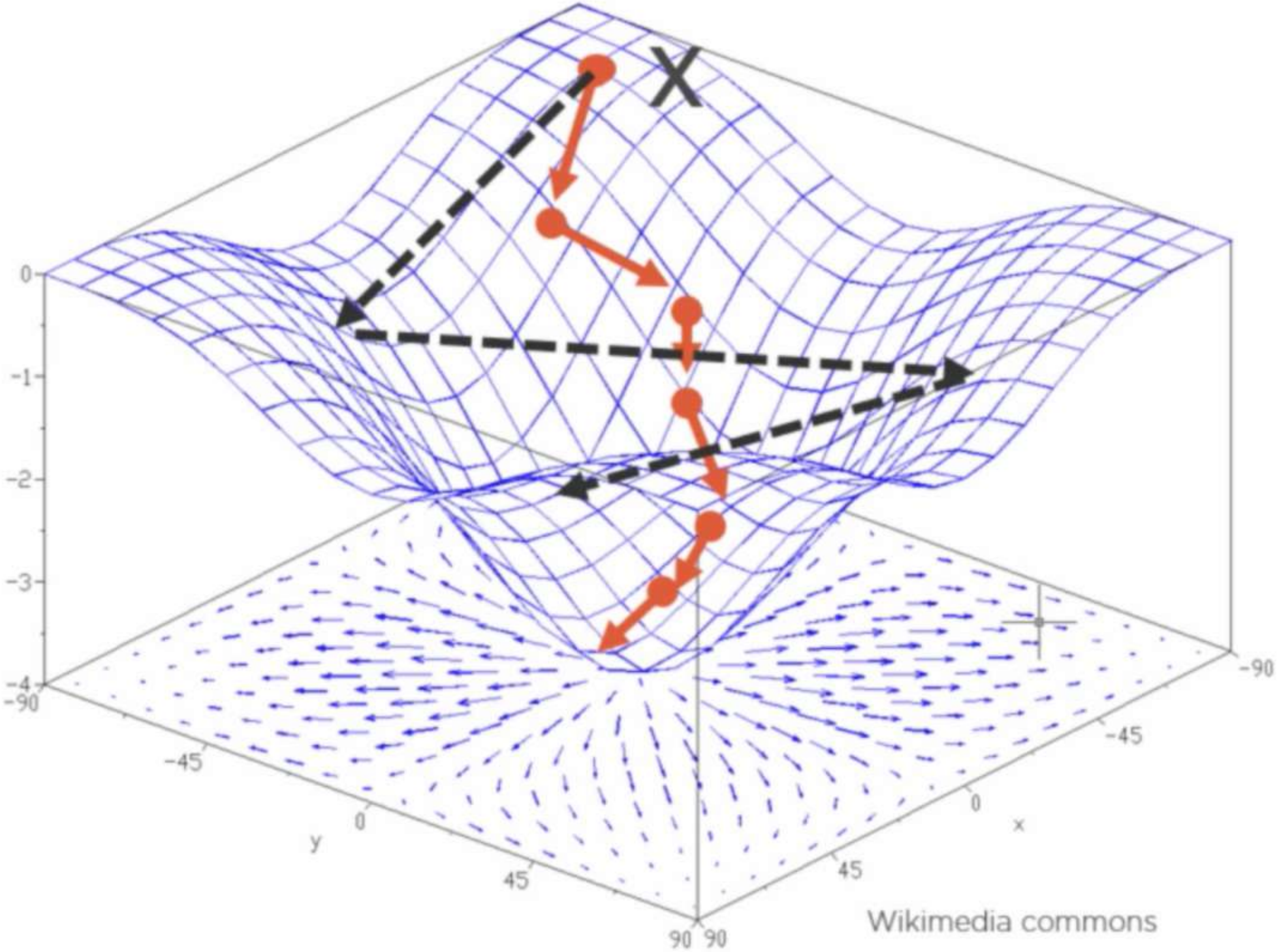
Wikimedia commons

Gradient Descent



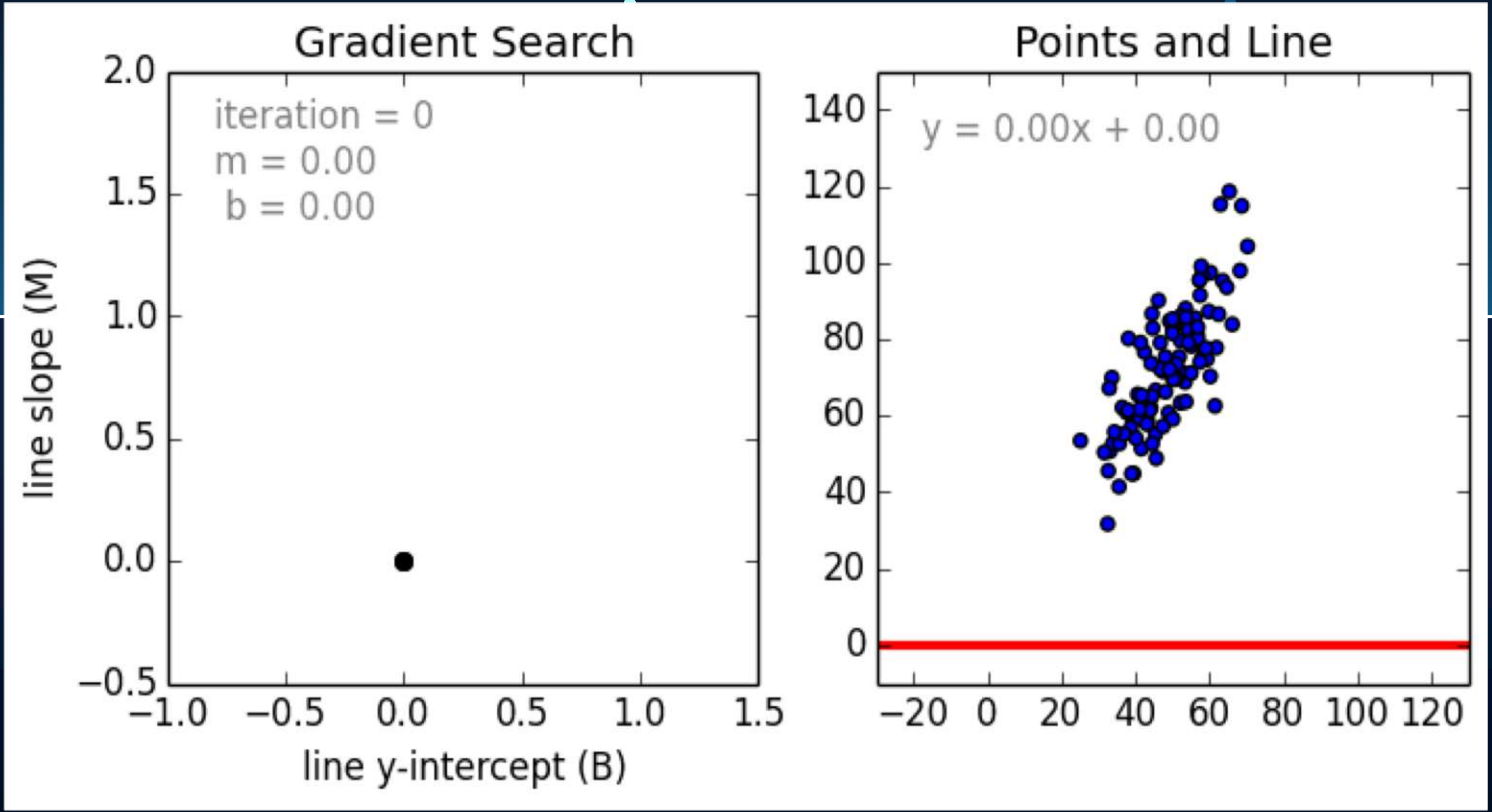
Wikimedia commons

Gradient Descent



Wikimedia commons

Gradient Descent



Coefficient of Determination, R^2

The coefficient of determination is the portion of the total variation in the dependent variable that is explained by variation in the independent variable

The coefficient of determination is also called R-squared and is denoted as R2

$$\boxed{R^2 = \frac{SSR}{SST}}$$

where

$$\boxed{0 \leq R^2 \leq 1}$$

Coefficient of Determination, R^2

- **Coefficient of determination**

$$R^2 = \frac{SSR}{SST} = \frac{\text{sum of squares explained by regression}}{\text{total sum of squares}}$$

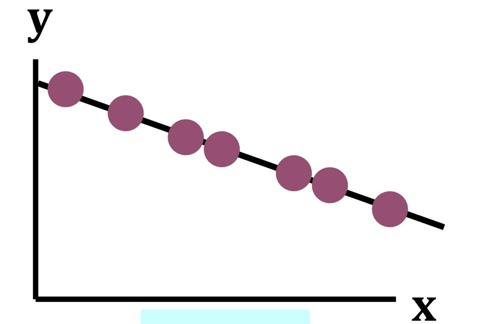
- **Note: In the single independent variable case, the coefficient of determination is**

$$R^2 = r^2$$

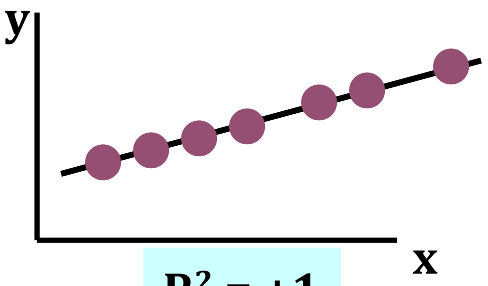
Where

R^2 = Coefficient of determination
r = Simple correlation coefficient

Examples of Approximate R^2 Values



$R^2 = 1$



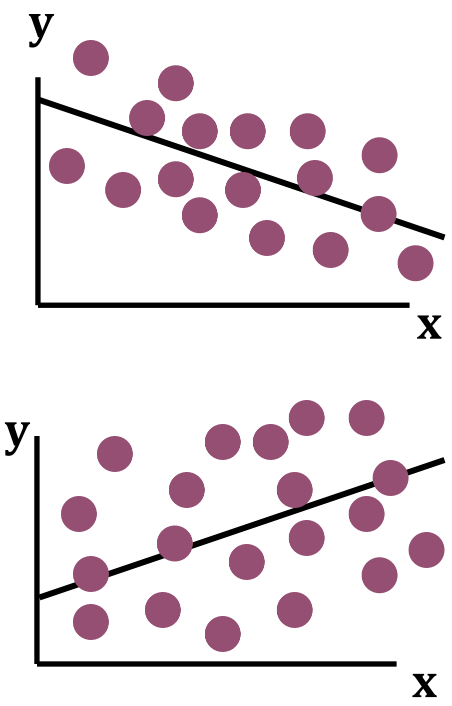
$R^2 = +1$

$R^2 = 1$

Perfect linear relationship between x and y:

100% of the variation in y is explained by variation in x

Examples of Approximate R^2 Values

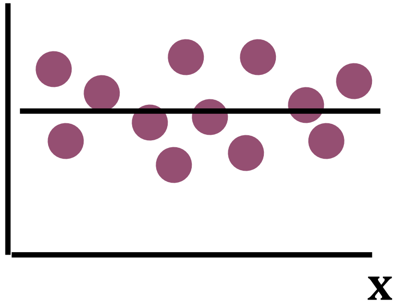


$0 < R^2 < 1$

Weaker linear relationship between x and y:

Some but not all of the variation in y is explained by variation in x

Examples of Approximate R^2 Values



$R^2 = 0$

No linear relationship between x and y:

The value of Y does not depend on x. (None of the variation in y is explained by variation in x)

Linear Regression with Gradient Descent

- **Congratulations!** That’s the first step in your machine learning and artificial intelligence journey.
- Get an intuitive feel for how gradient descent works because this is actually used in more advanced models also.

How hard it is to code?

Step 1

`import` statement:

```
from sklearn import linear_model
```

Step 2

I have the height and weight data of some people. Let's use this data to do linear regression and try to predict the weight of other people.

```
height=[[4.0],[4.5],[5.0],[5.2],[5.4],[5.8],[6.1],[6.2],[6.4],[6.8]]
weight=[  42 ,  44 , 49, 55  , 53  , 58   , 60  , 64  ,  66 ,  69]

print("height weight")
for row in zip(height, weight):
    print(row[0][0],"->",row[1])
```

Step 3

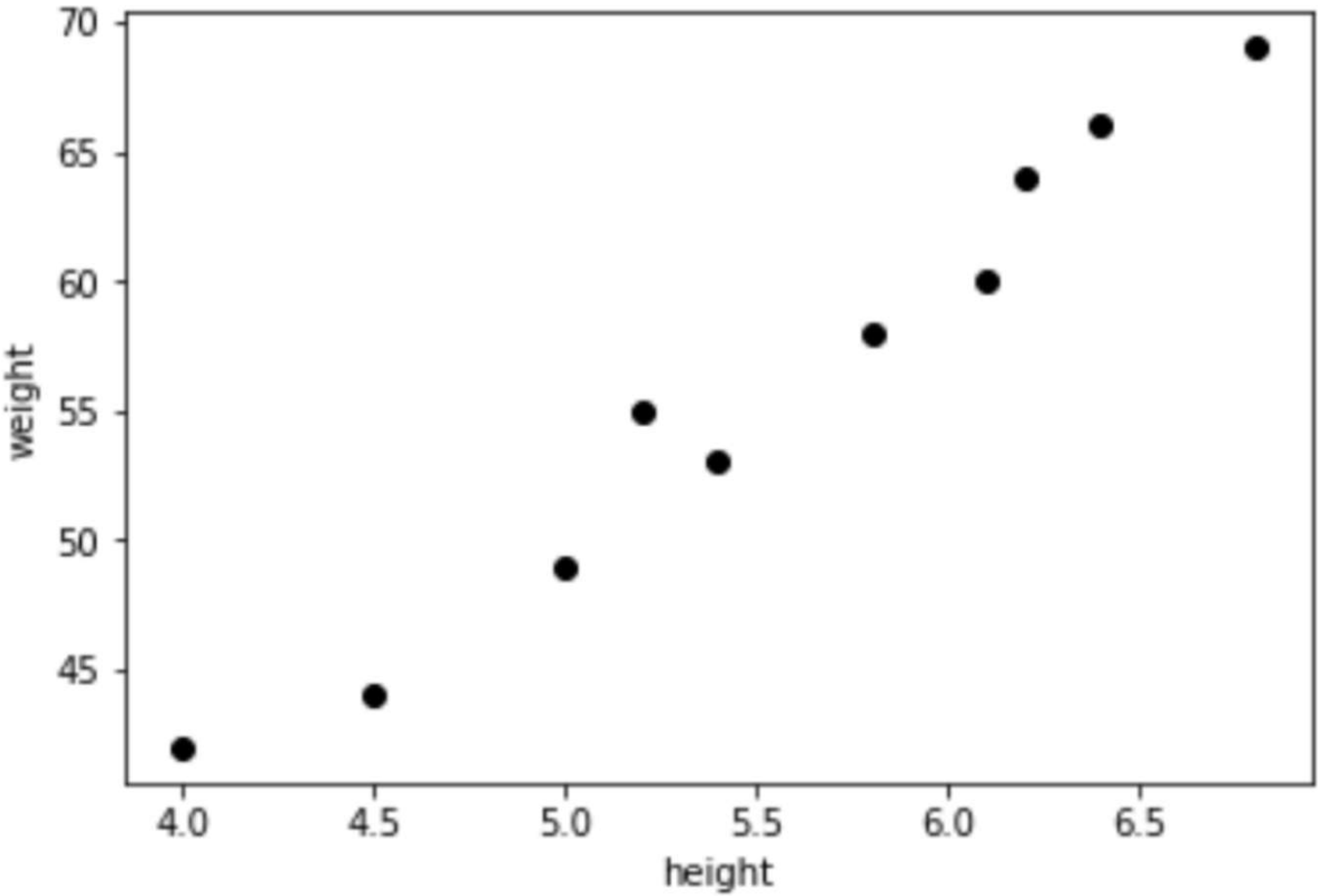
`import` statement to plot graph using `matplotlib`:

```
import matplotlib.pyplot as plt
```

Plotting the height and weight data:

```
plt.scatter(height,weight,color='black')
plt.xlabel("height")
plt.ylabel("weight")
```

Output:



Step 4

Declaring the linear regression function and call the `fit` method to learn from data:

```
reg=linear_model.LinearRegression()
reg.fit(height,weight)
```

Slope and intercept:

```
m=reg.coef_[0]
b=reg.intercept_
print("slope=",m, "intercept=",b)
```

Output:

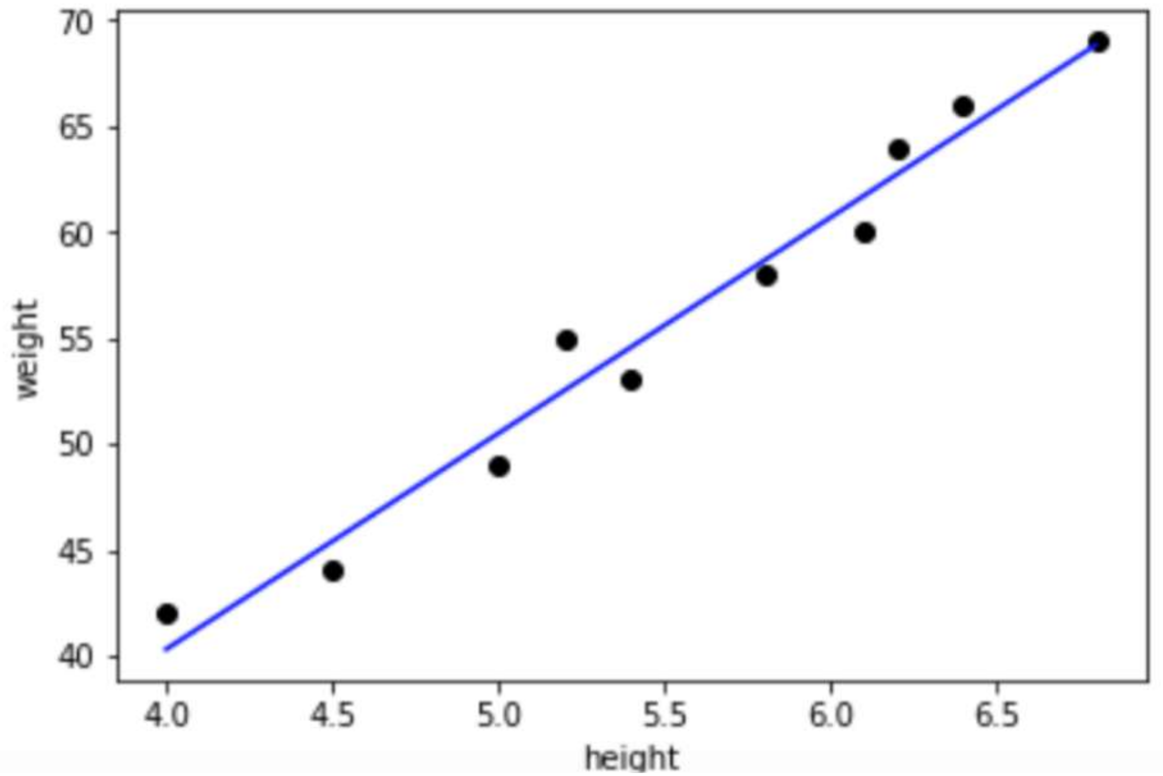
```
slope= 10.1936218679 intercept= -0.4726651480
```

Step 5

Using the values of slope and intercept to construct the line to fit our data points:

```
plt.scatter(height,weight,color='black')
predicted_values = [reg.coef_ * i + reg.intercept_ for i in height]
plt.plot(height, predicted_values, 'b')
plt.xlabel("height")
plt.ylabel("weight")
```

Output:



Error Metrics

Mean squared error	$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2$
Root mean squared error	$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$
Mean absolute error	$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} \lvert e_t \rvert$
Mean absolute percentage error	$\text{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n} \left\lvert \frac{e_t}{y_t} \right\rvert$

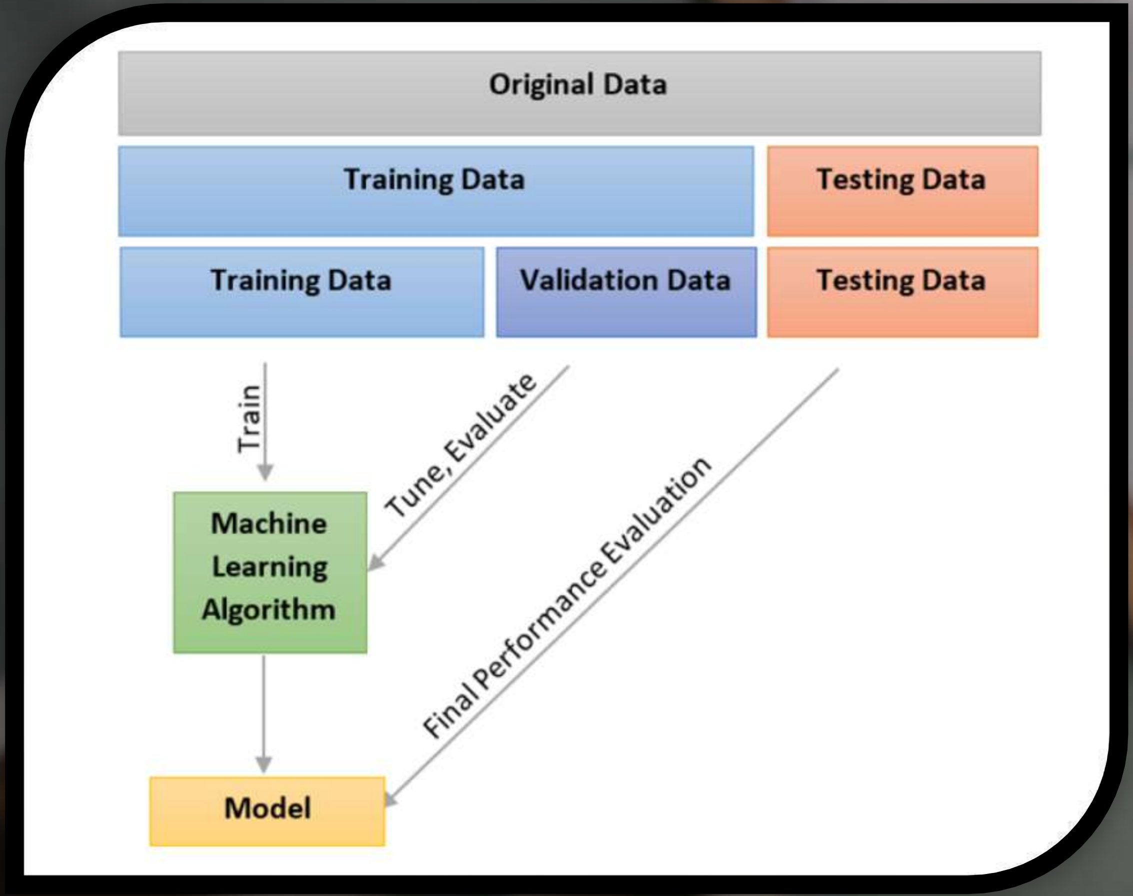
Advantage of Linear Regression

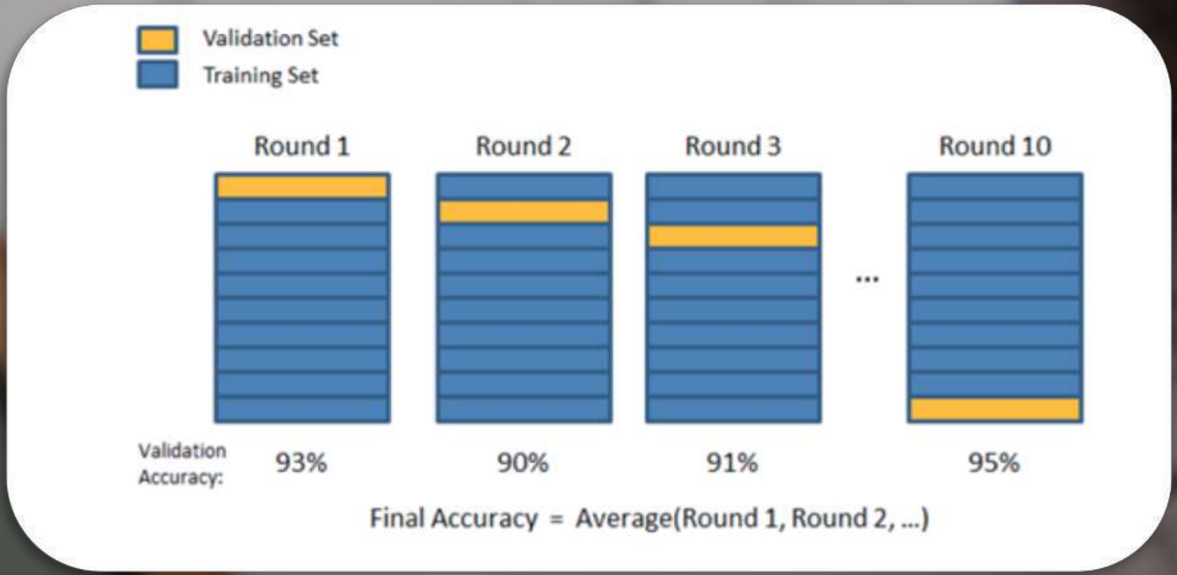
- Linear regression implements a statistical model that, when relationships between the independent variables and the dependent variable are almost linear, shows optimal results.
- Best place to understand the data analysis
- Easily Explicable

Disadvantages

- Linear regression is often inappropriately used to non-linear relationships.
- Linear regression is limited to predicting numeric output.
- A lack of explanation about what has been learned can be a problem.
- Prone to bias variance problem

How to evaluate our model?





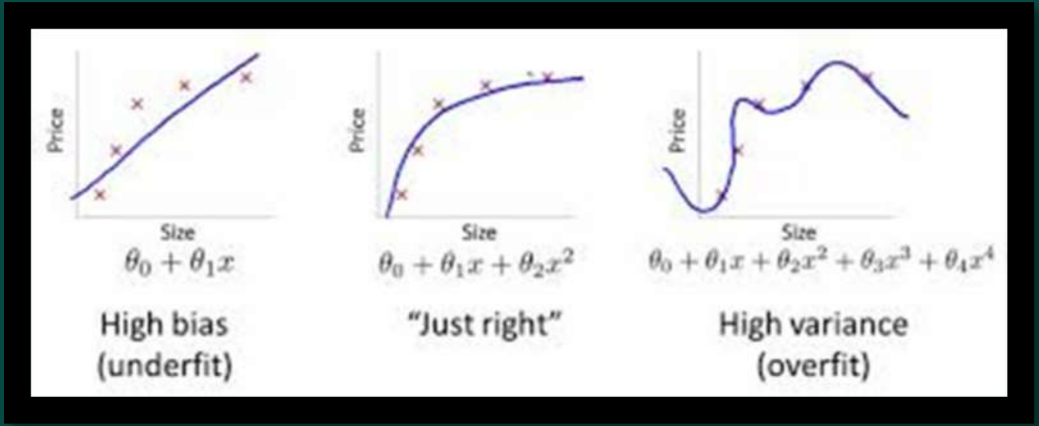
Training Data(Less Error)

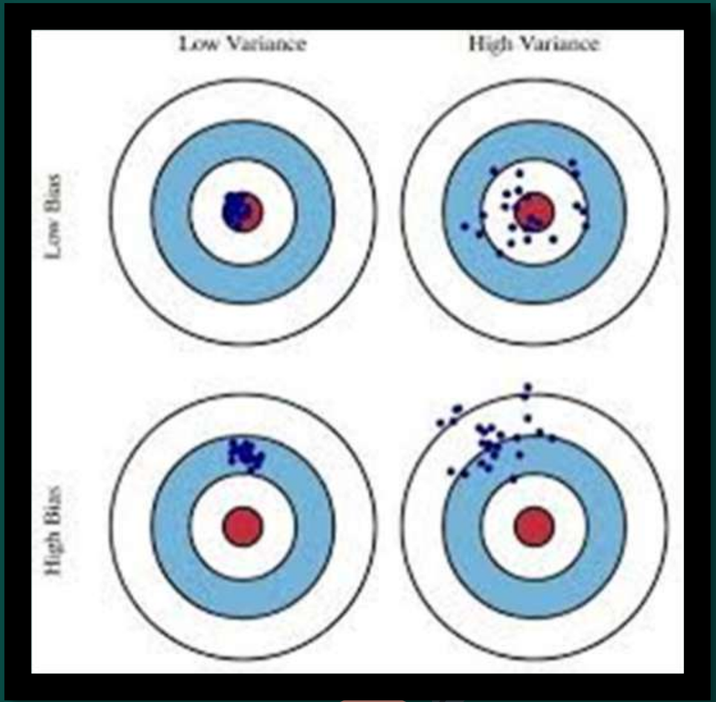
Testing Data (More Error)

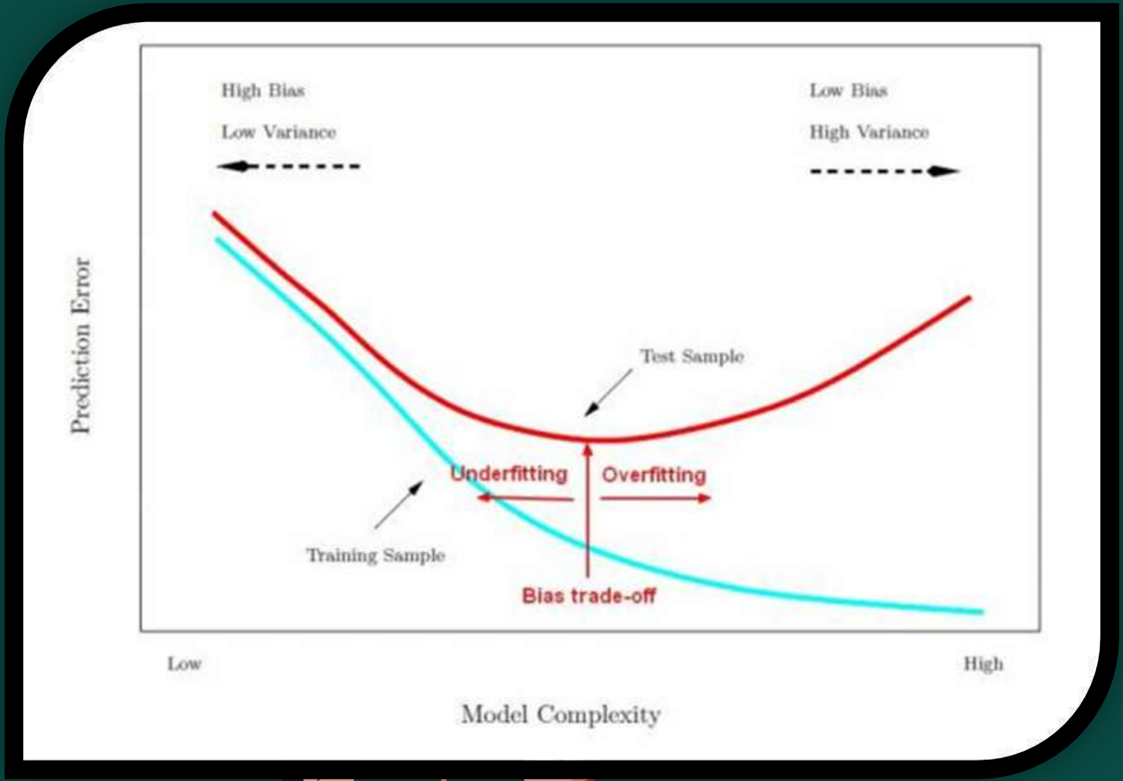
Training Data(Less Error)

Testing Data (More Error)

Variance and Bias Trade off






High Bias
Low Variance
Low Bias
High Variance
Prediction Error
Test Sample
Underfitting
Overfitting
Training Sample
Bias trade-off
Low
High
Model Complexity


THANK YOU