LINEAR REGRESSION

Thomas Tiam-Lee, PhD

Norshuhani Zamin, PhD

De La Salle University Manila
College of Computer Studies

Linear Regression

- A **supervised learning algorithm**
  - Contains a target variable (label) that we want to predict
- A model designed for **regression**
  - The label is a **continuous numerical value**
- **Example:** *predict the price of the house given its lot area*

The Data

Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425
60	1657
62	1457
64	1504
66	1522
68	1594

The Data

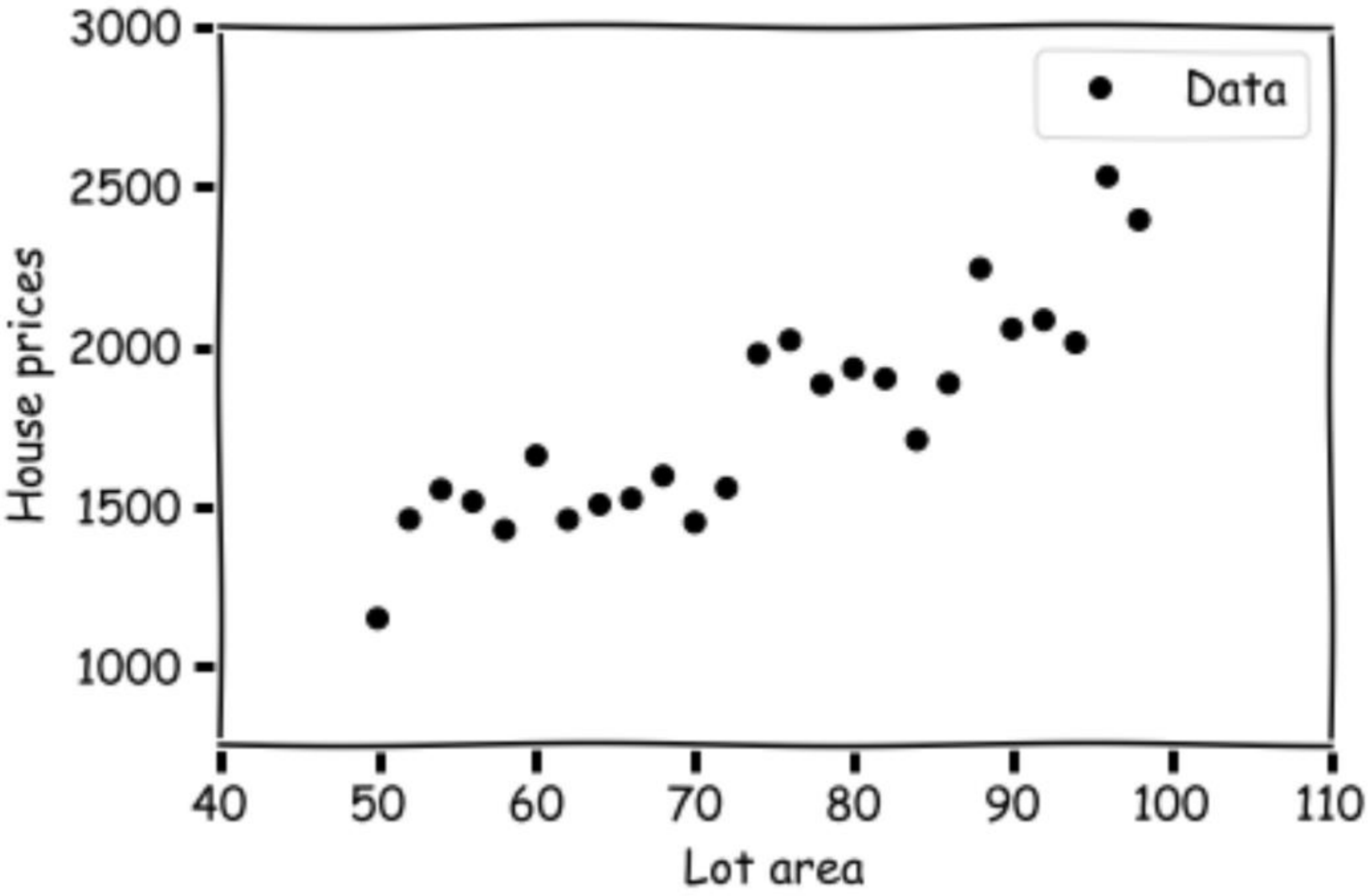
Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425
60	1657
62	1457
64	1504
66	1522
68	1594

Key idea: we can visualize **the relationship between the features** (lot area) **and the label** (house price) using a scatterplot.

Lot Area = Independent variable
House Price = Dependent variable (target variable)

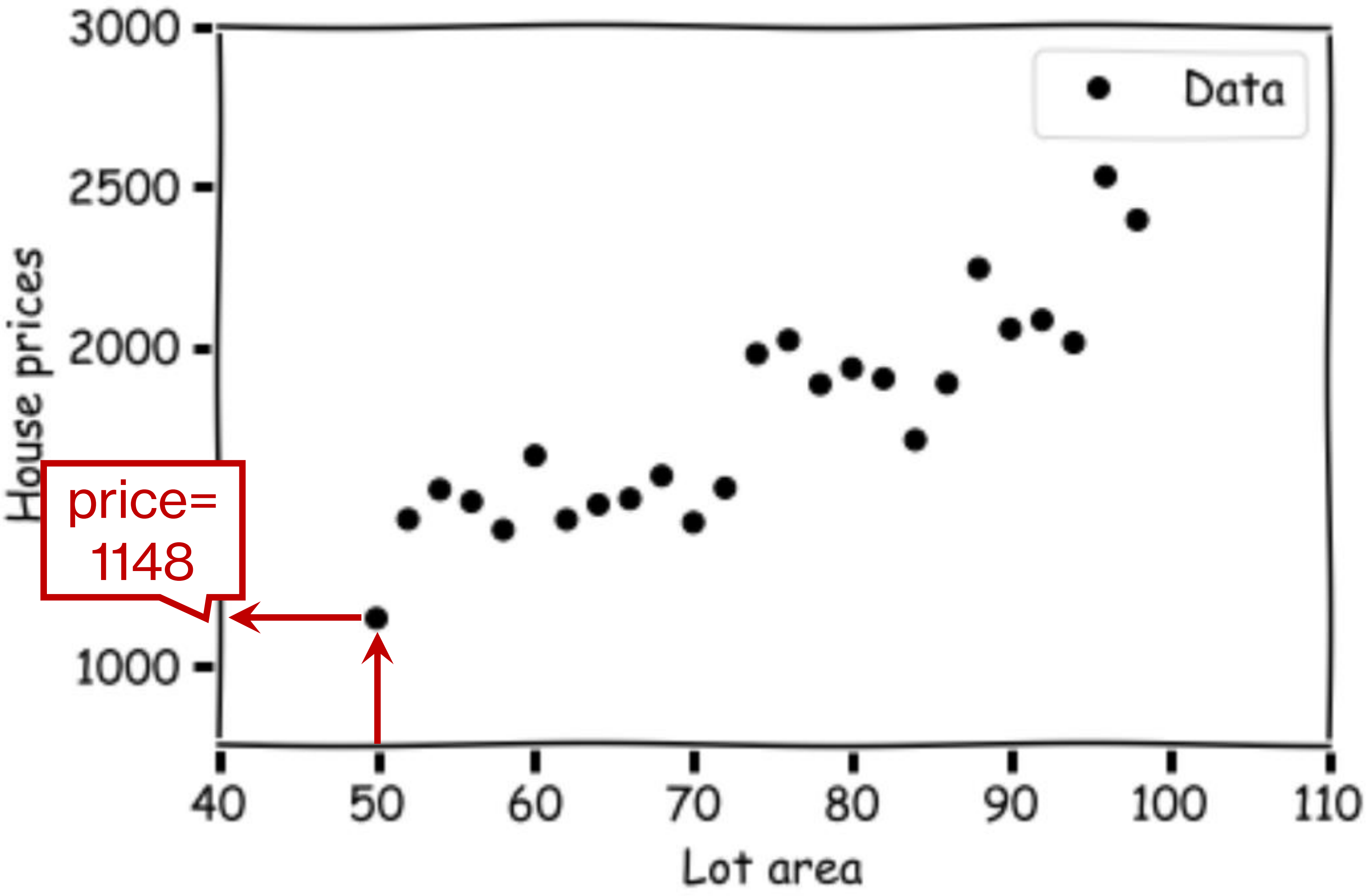
The Data

Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425
60	1657
62	1457
64	1504
66	1522
68	1594



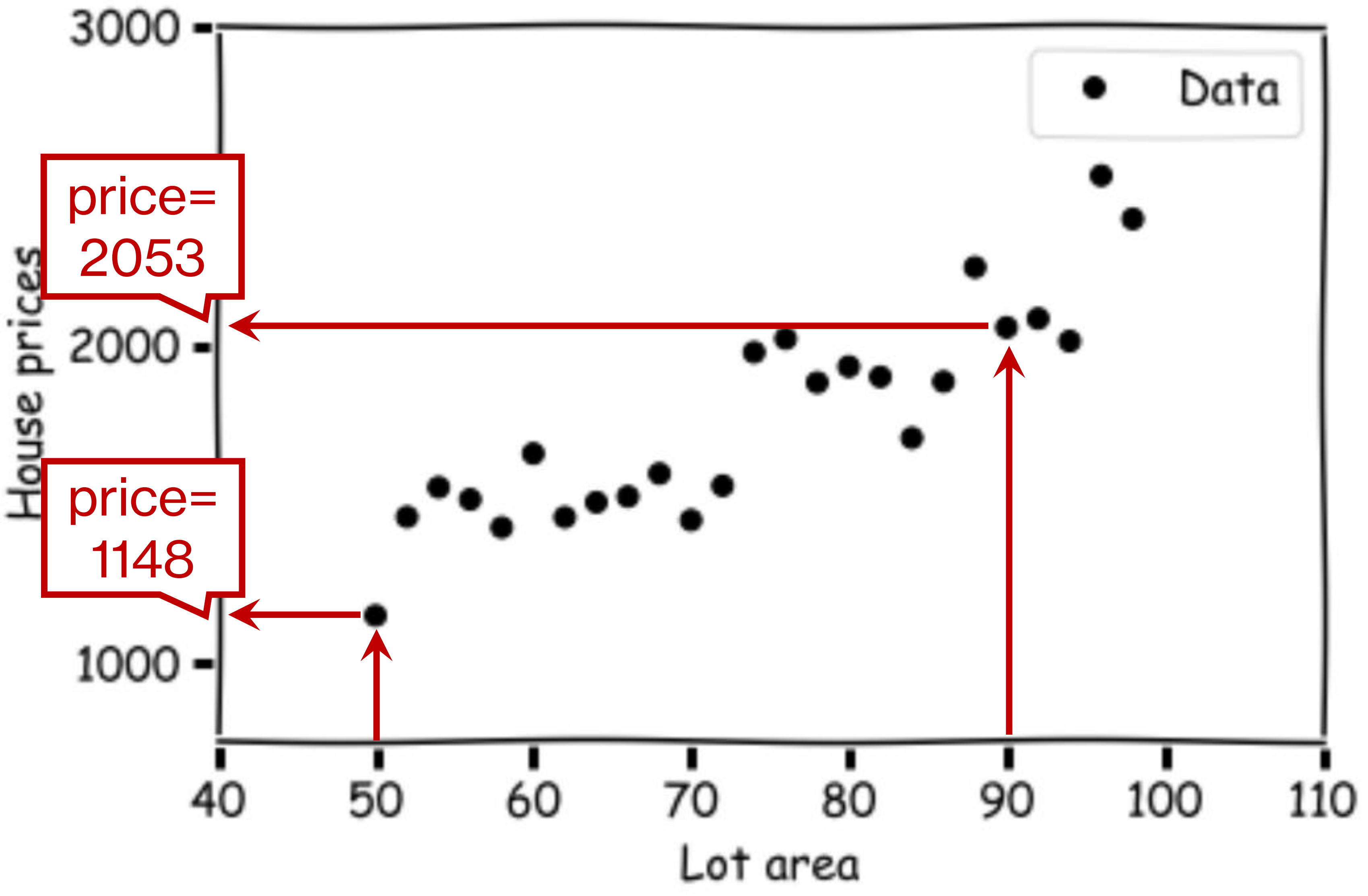
The Data

Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425
60	1657
62	1457
64	1504
66	1522
68	1594



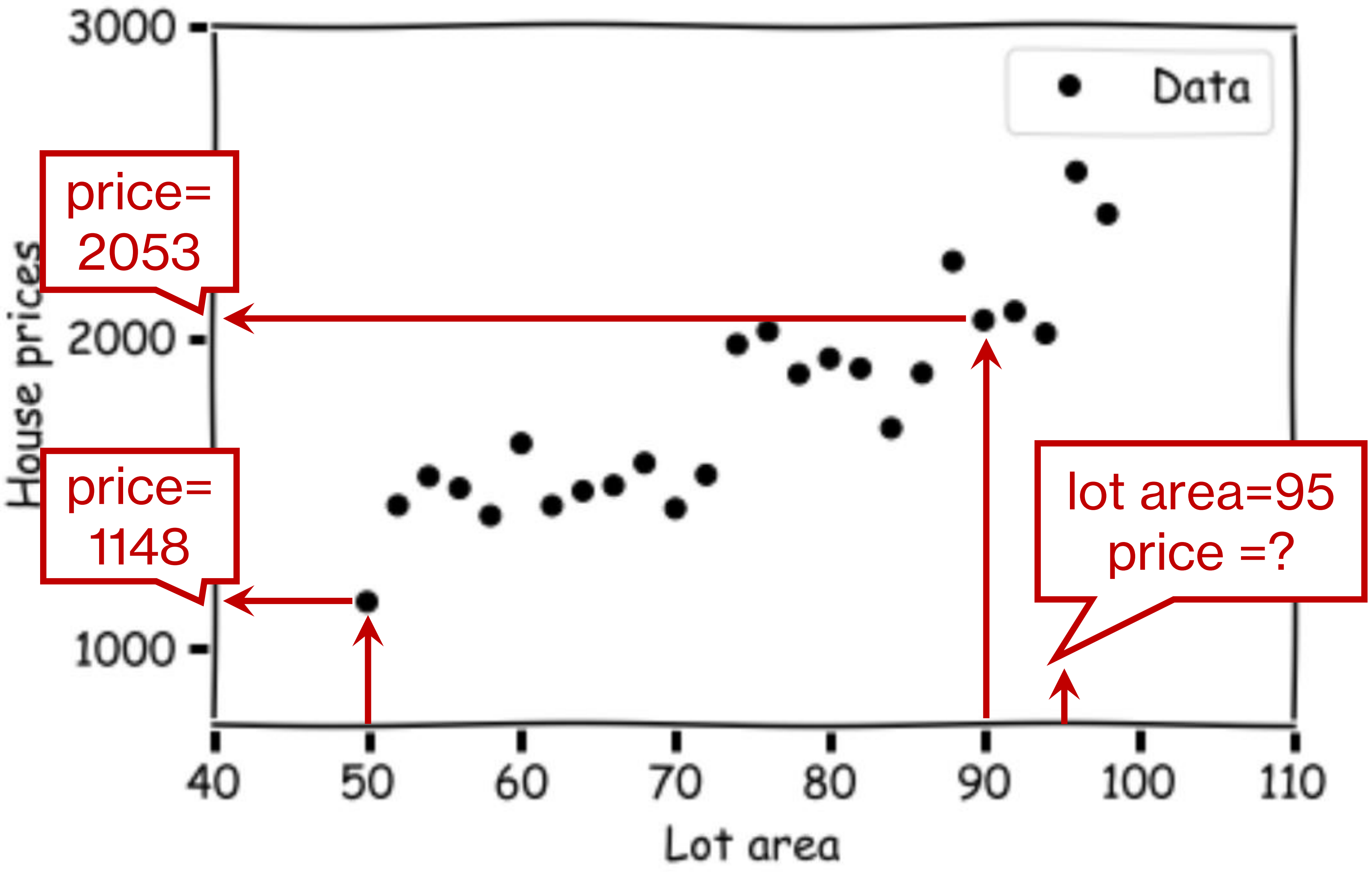
The Data

Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425
60	1657
62	1457
64	1504
66	1522
68	1594



The Data

Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425
60	1657
62	1457
64	1504
66	1522
68	1594



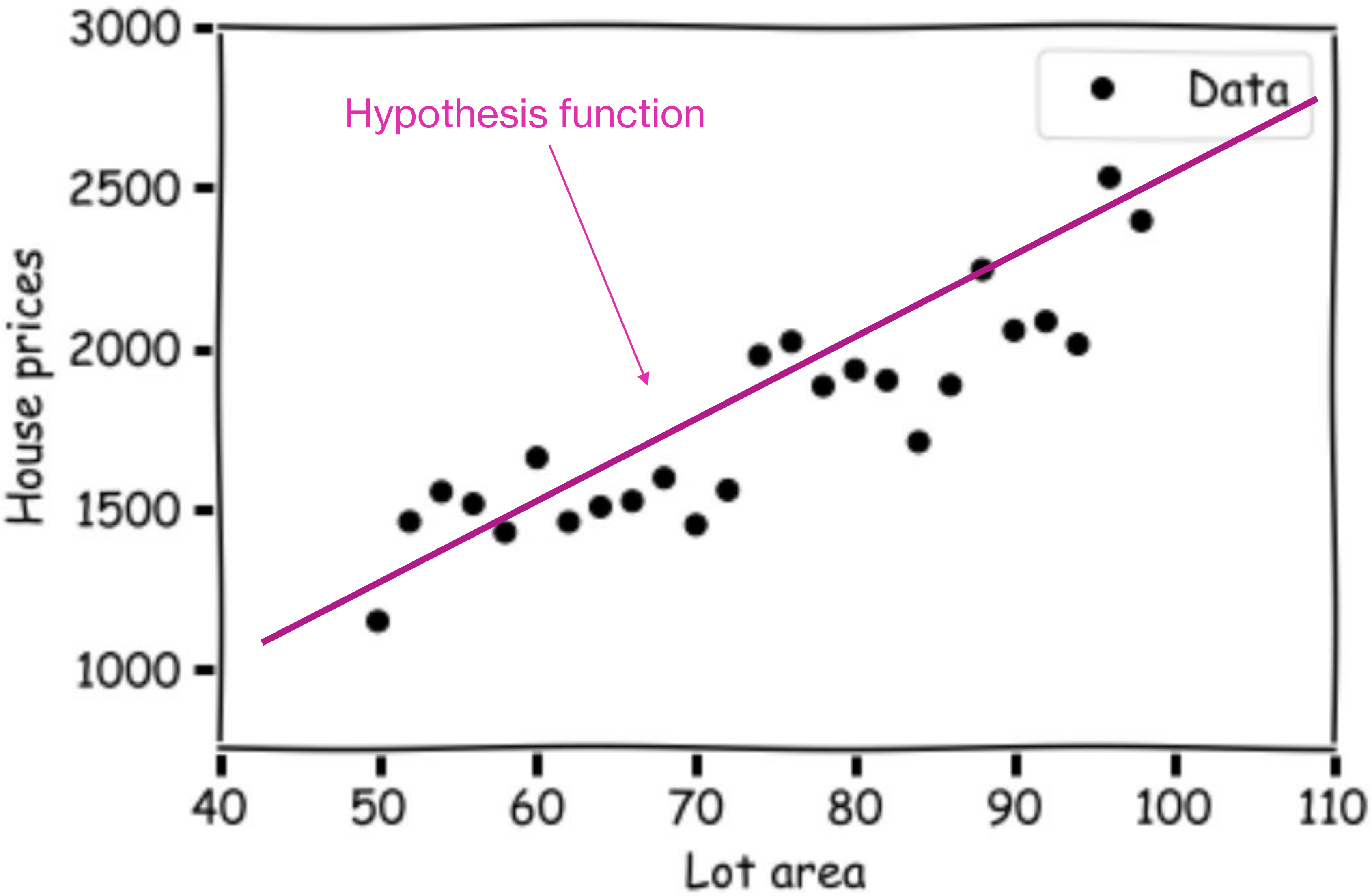
How can we predict the price of a house that we have not seen yet before?

Modeling: Linear Regression

- In linear regression, the hypothesis function (model) is a **linear equation**.
  - In its most basic form (1 feature), this can be visualized as a **line**.

The Data

Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425
60	1657
62	1457
64	1504
66	1522
68	1594



Modeling: Linear Regression

- **Equation of a line:**
  - $y = mx + b$
    - feature (lot area): x
    - label (price): y

- m and b are the **parameters** of the model
  - m is the **slope** of the line
  - b is the **y-intercept**

Modeling: Linear Regression

- **Rewritten in a different form:**
  - $y = w_1 x + w_0$
    - feature (lot area): x
    - label (price): y

- w_1 and w_0 are the **parameters** of the model
  - w_1 is the **slope** of the line
  - w_0 is the **y-intercept**

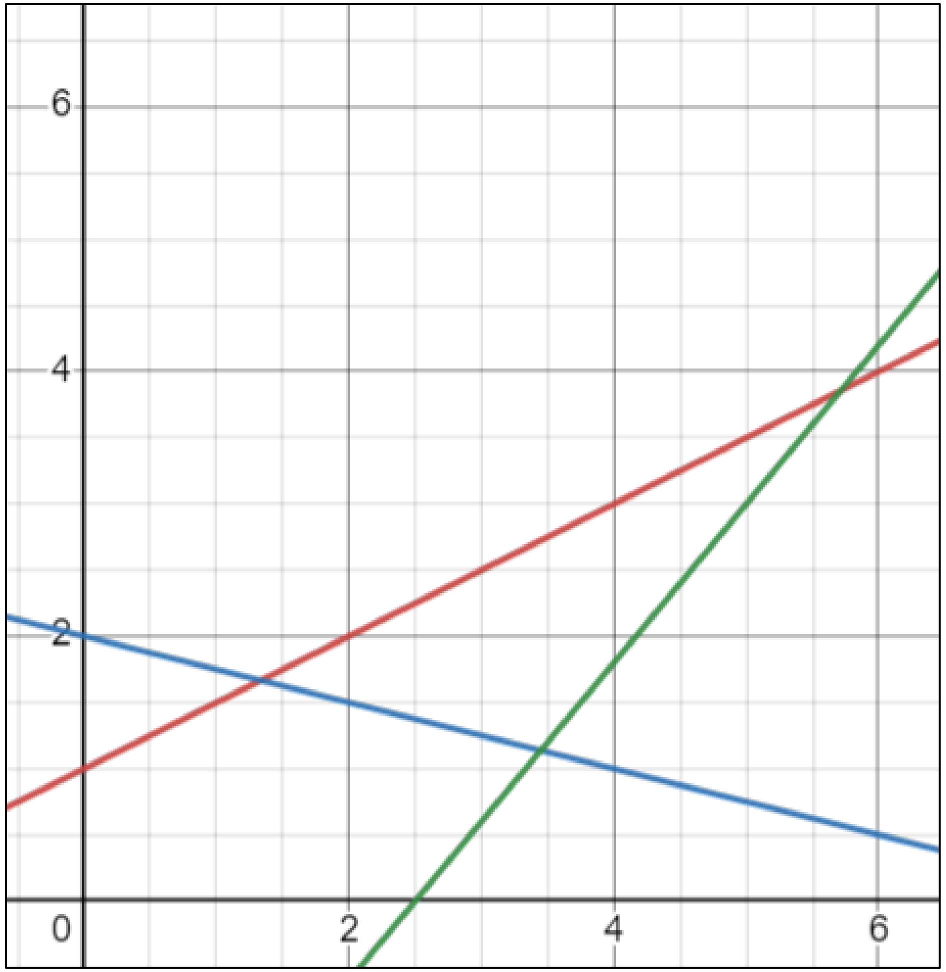
Modeling: Linear Regression

- **Key Idea:** By **changing the parameters** of the model, we can **change how the model behaves** (i.e., the orientation of the line)

$w_1 = 0.5$
$w_0 = 1$

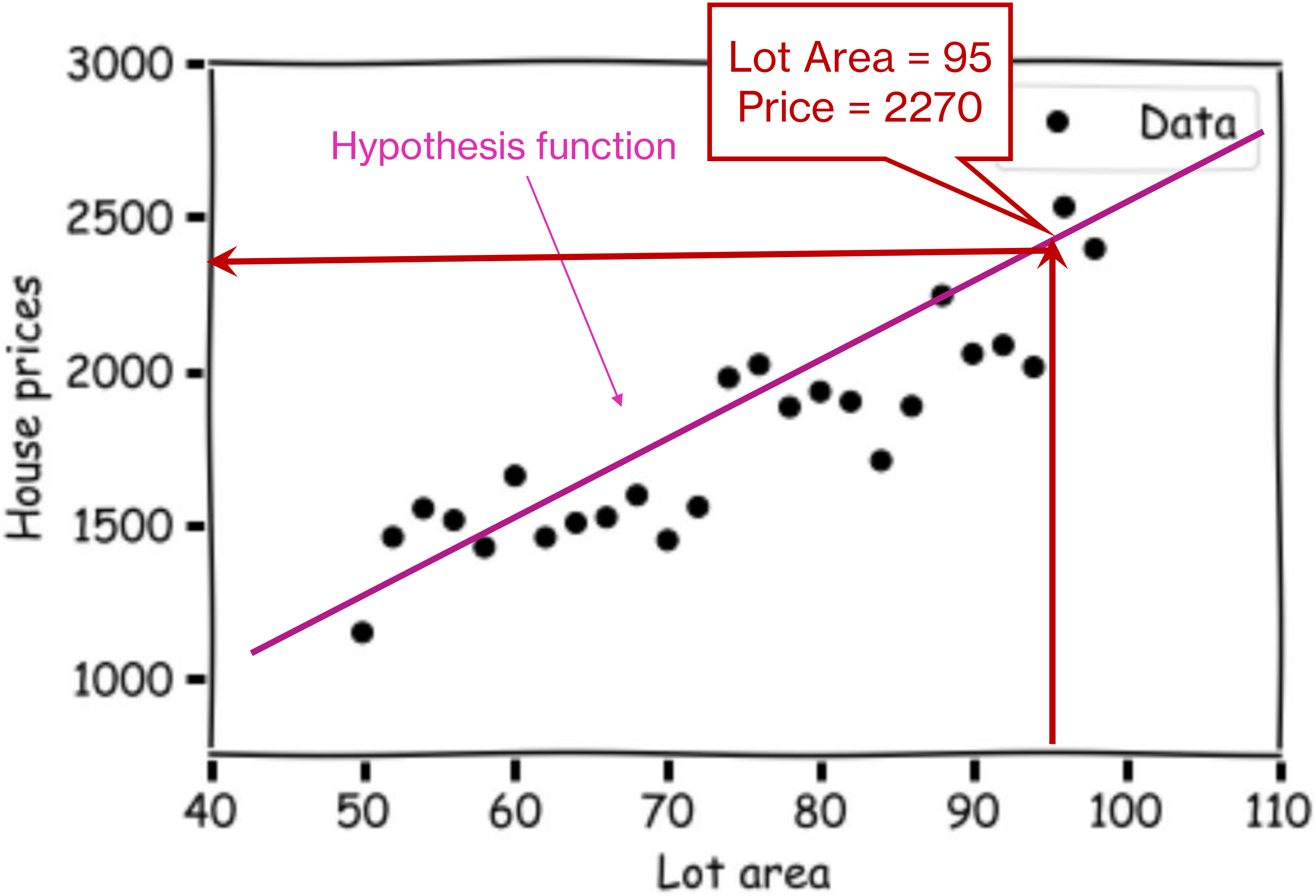
$w_1 = -0.25$
$w_0 = 2$

$w_1 = 1.2$
$w_0 = -3$



The Data

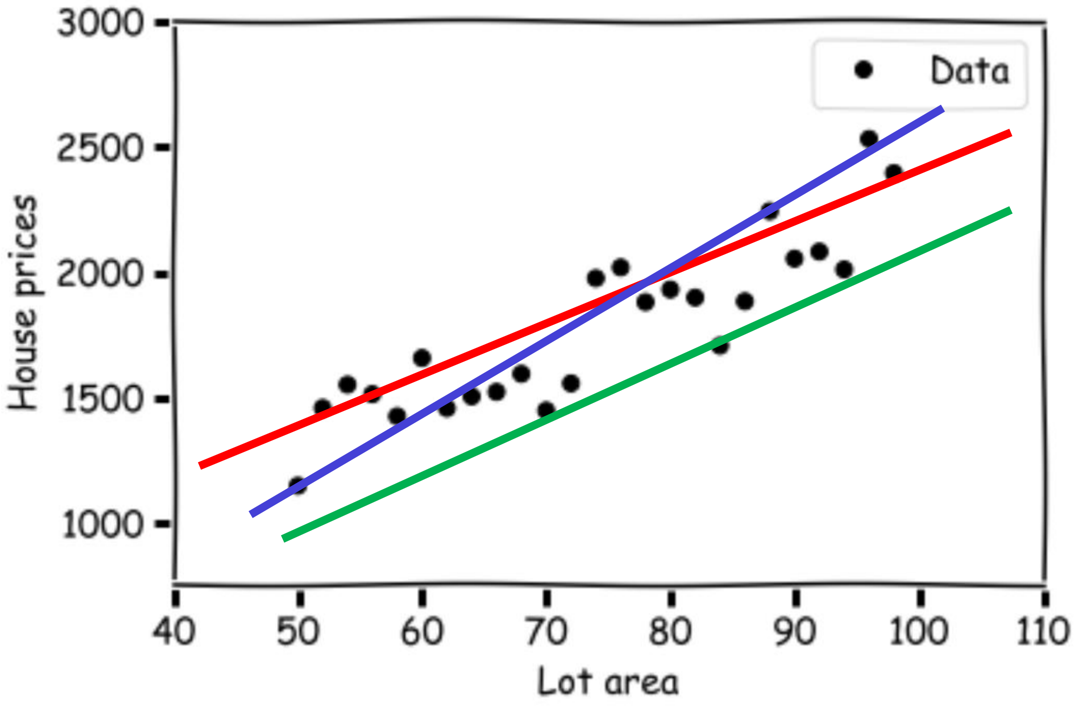
Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425
60	1657
62	1457
64	1504
66	1522
68	1594



Given the line, we can make a prediction by plugging in the lot area (x) and solving the equation!

The **Learning** Algorithm

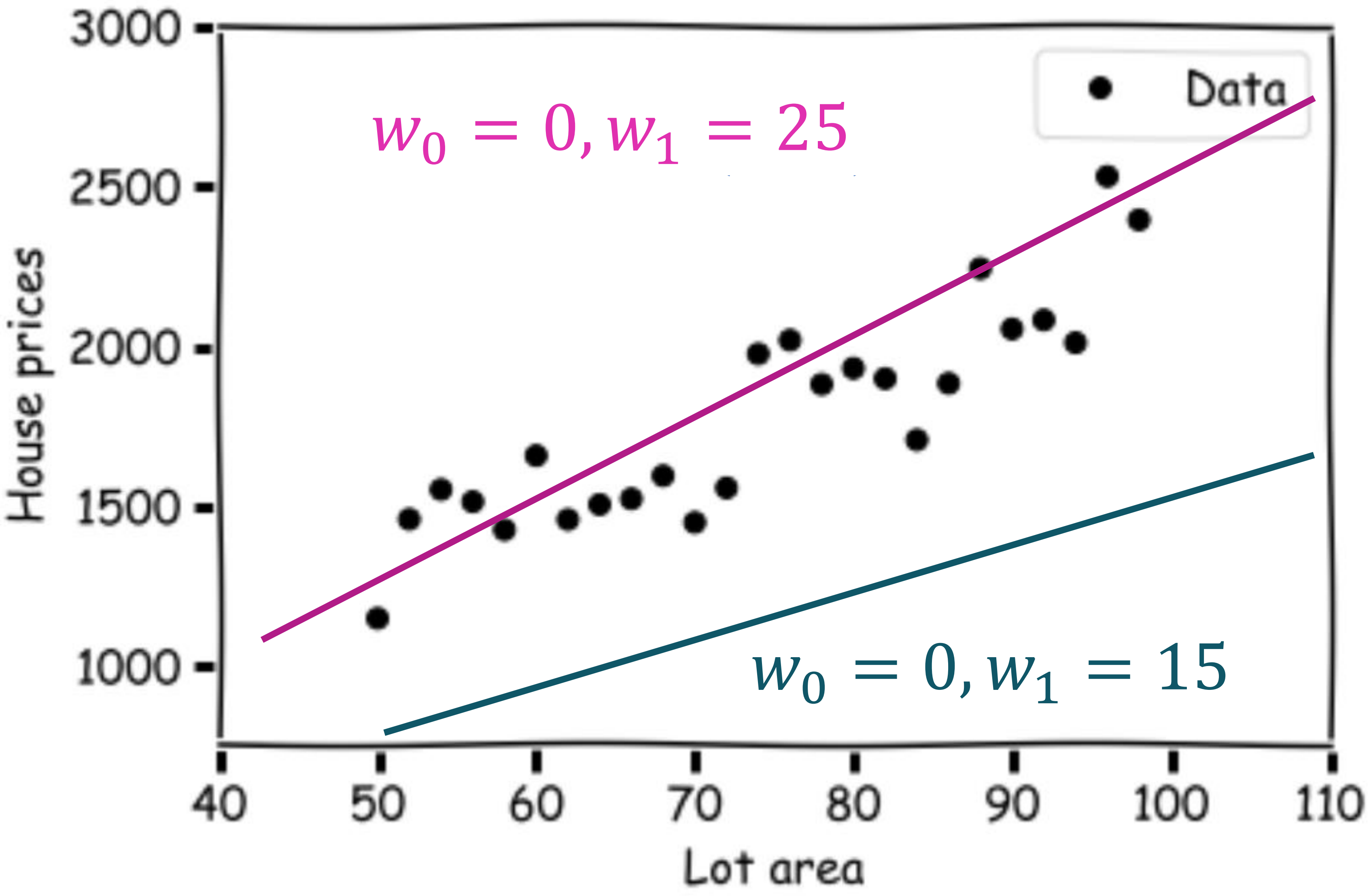
- The "magic" of machine learning lies within:
  - Given a set of training data (points), how can we **find the values of the parameters** to make a line that **best fits** the data?
- We're going to find the best line systematically using a **learning algorithm**.

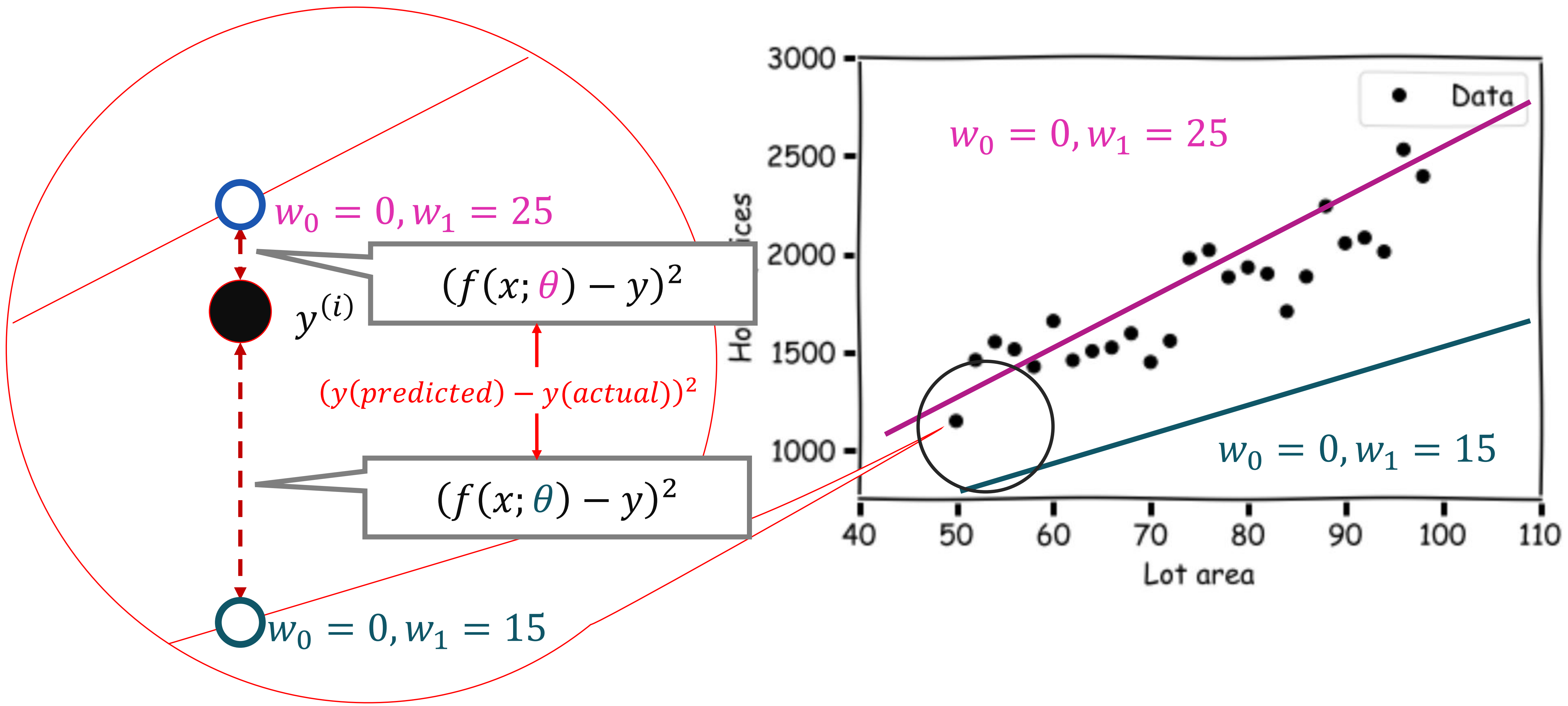


Which line is the best?

But First

- Need a **mathematical way** to measure “how good a line is”





$w_0 = 0, w_1 = 25$
$y^{(i)}$
$(f(x;\theta) - y)^2$
$(y(predicted) - y(actual))^2$
$(f(x;\theta) - y)^2$
$w_0 = 0, w_1 = 15$
3000
2500
2000
1500
1000
Data
$w_0 = 0, w_1 = 25$
$w_0 = 0, w_1 = 15$
40
50
60
70
80
90
100
110
Lot area

Loss Function

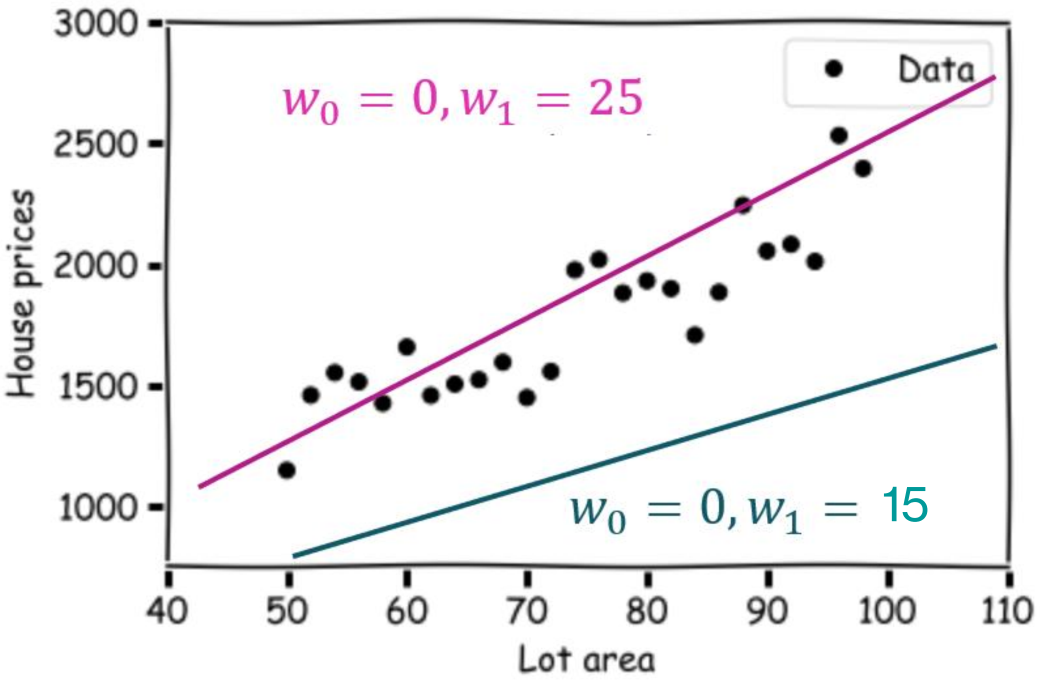
- (also known as **objective function**, **cost function**)
- A function that accepts a model and the training data and returns **a numerical measure of how well the model fits the data**.
- A loss of 0 is the best possible score (the model fits the data perfectly) otherwise keep the function minimum.

https://www.youtube.com/watch?v=erfeZg27B7A

A Linear Regression Loss Function

$$l(\theta) = \frac{1}{2n}\sum(f(x;\theta) - y)^2$$

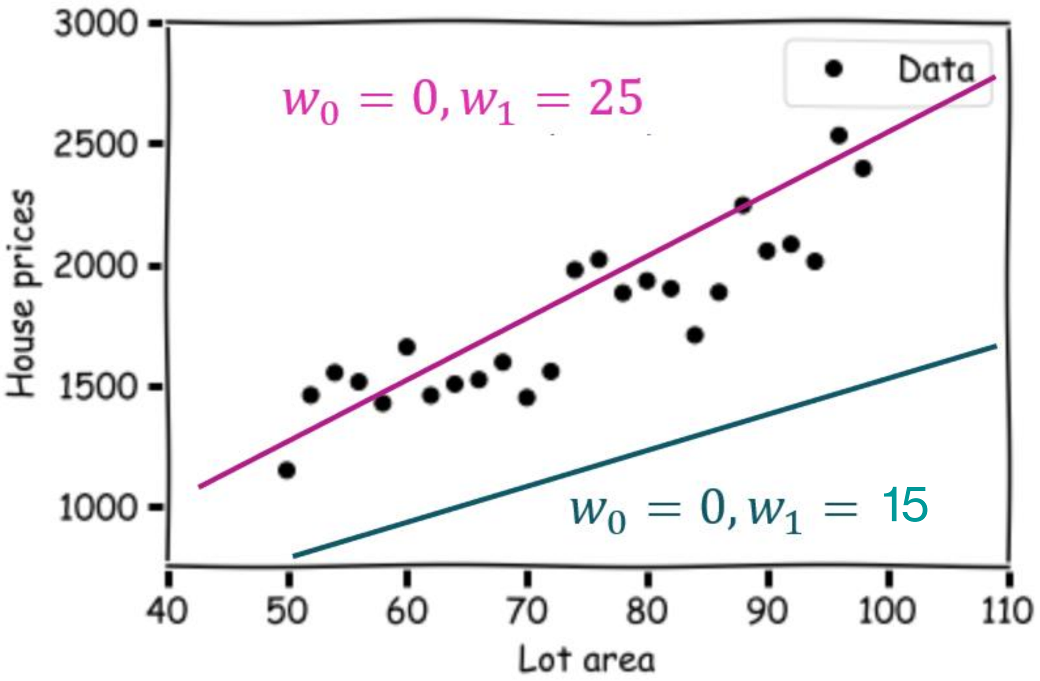
- Also known as the **Mean Squared Error**
- Measures the average "error" of each prediction of the model on the training data



- **Why squared?**
  - Remove negative values
  - Penalize larger errors more

Goal of Learning Algorithm

$$l(\theta) = \frac{1}{2n}\sum(f(x;\theta) - y)^2$$

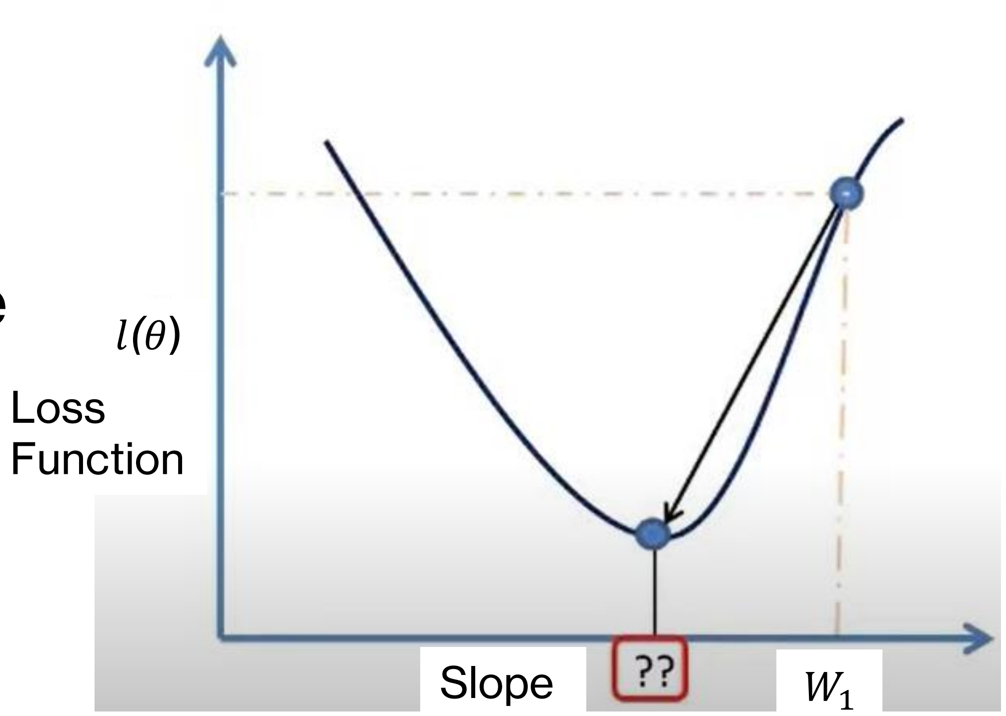


- How do we find the set of parameters θ (w_0 and w_1) that will **minimize the loss function**?
- Parameters are also known as coefficients.

Gradient Descent Learning Algorithm

- Gradient Descent is an optimization algorithm to find the minimum of a function.
- To make things simple, let us first assume that our w_0 (y-intercept) is fixed (always 0, no intersection). We can only change w_1 (slope).
- Goal:

To find the best slope that will make the line best fit the data.



Gradient Descent Learning Algorithm

procedure Gradient Descent(θ):

while not converged do:

$$\theta_i := \theta_{i-1} - \alpha \frac{\partial y}{\partial x}$$

return θ

θ is the slope

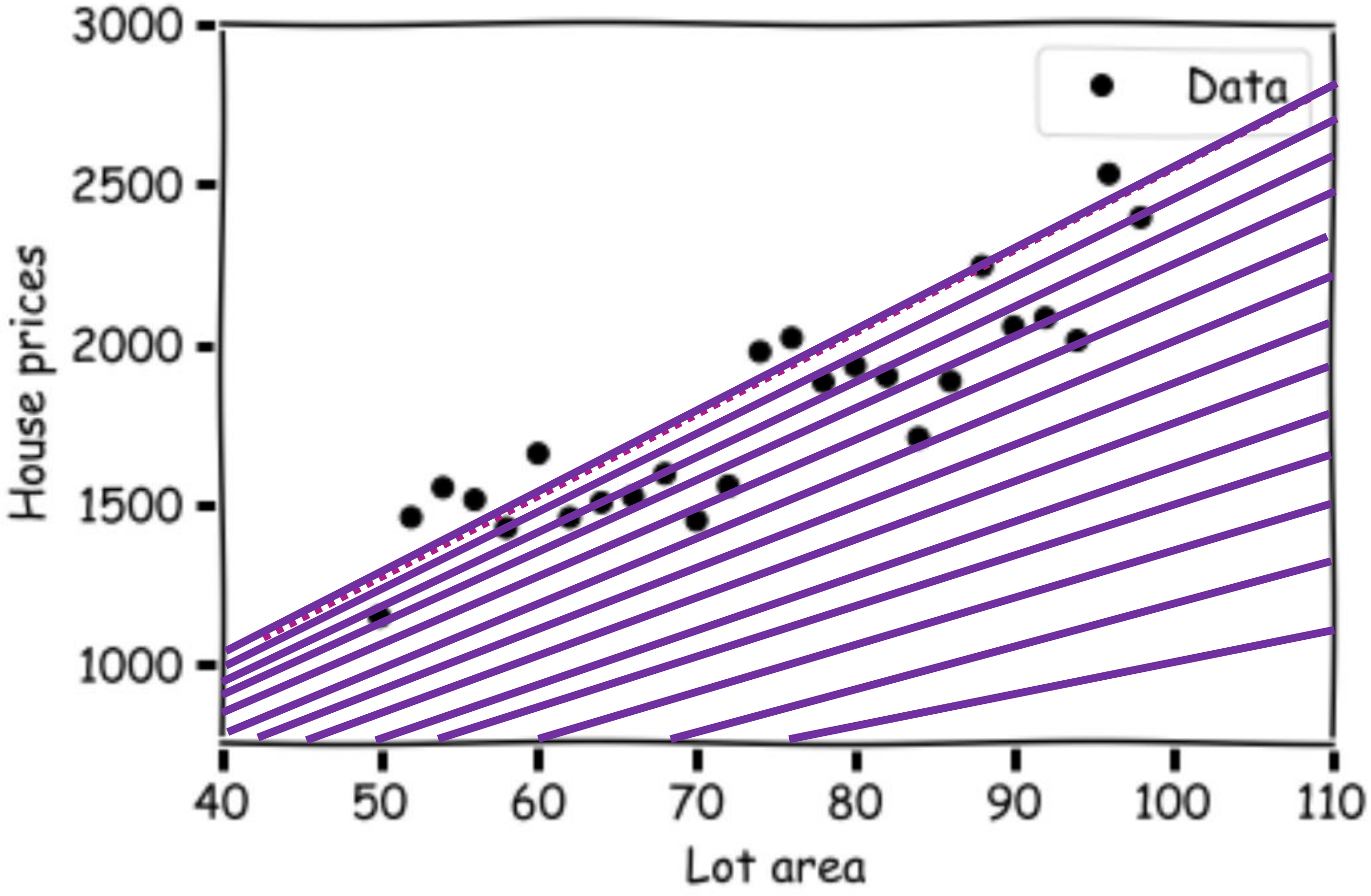
α is the learning rate, determines how large the update will be. α is usually kept at 0.01

$\frac{\partial y}{\partial x}$ is the gradient of the loss

STEP by STEP:
https://www.youtube.com/watch?v=Gbz8RljxIHo

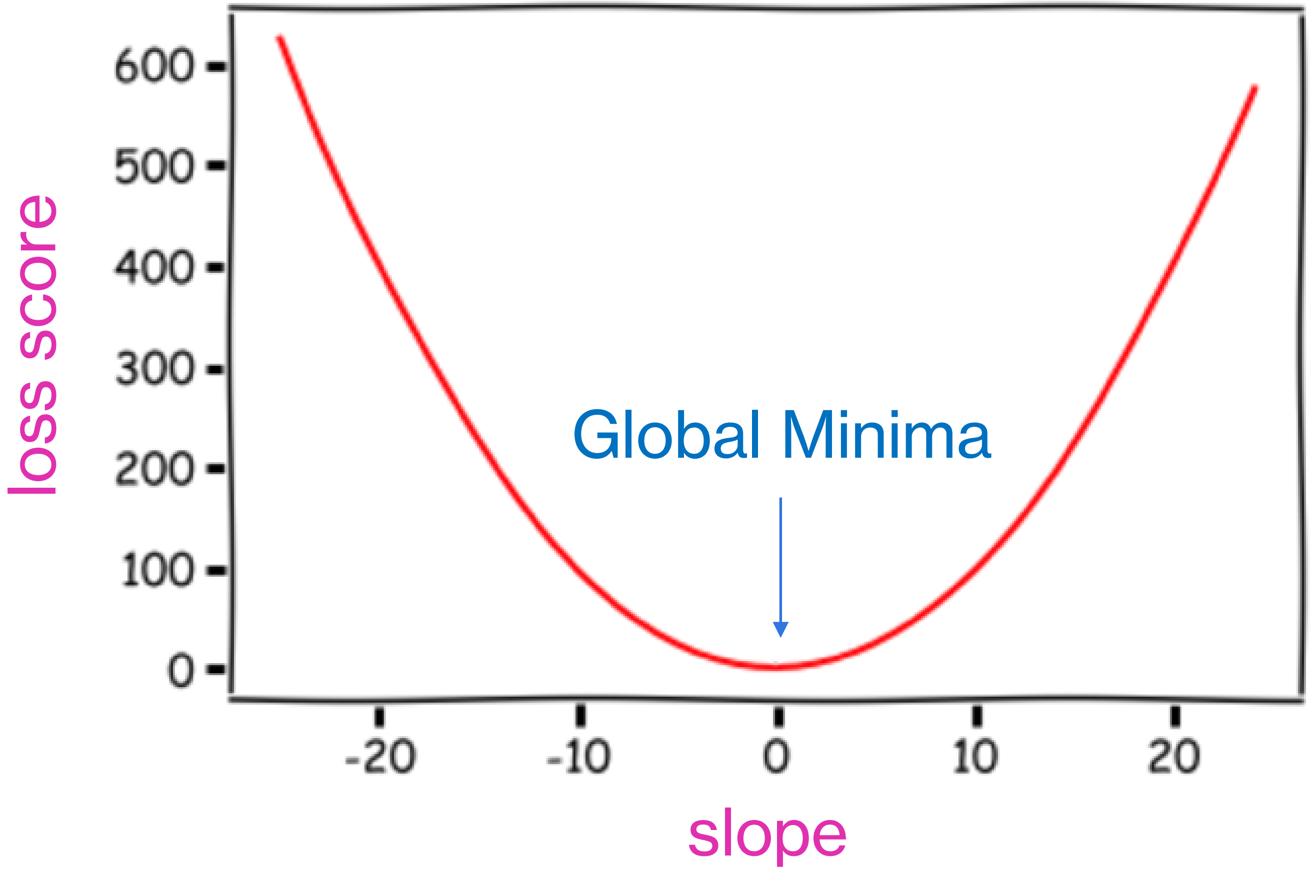
Gradient Descent Learning Algorithm

- **Key idea:** start with a random slope, then keep adjusting until the loss function score improves!



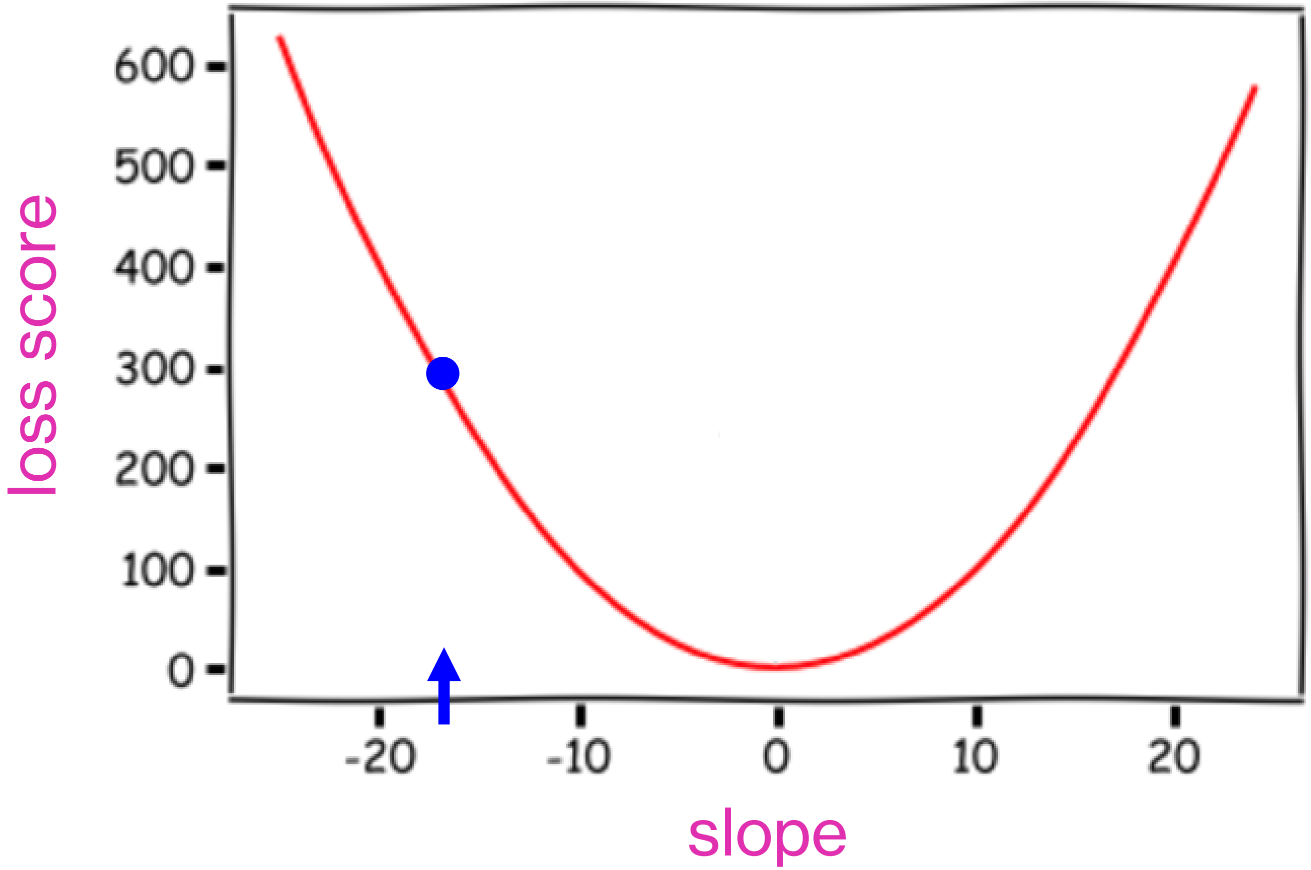
Gradient Descent Learning Algorithm

- **Key idea:** want to try out different slopes until you reach the lowest point!



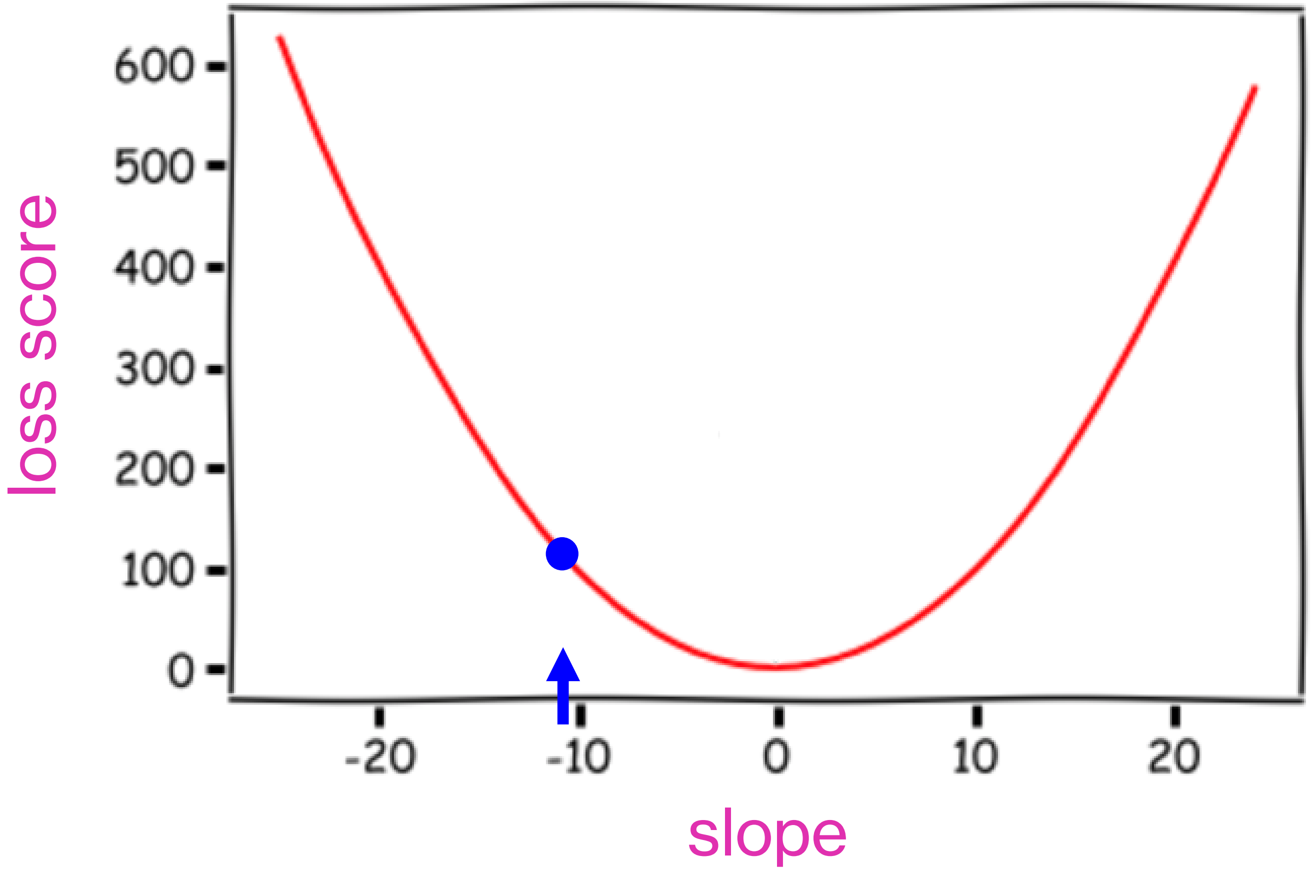
Gradient Descent Learning Algorithm

- Start at a random point. Compute the score.



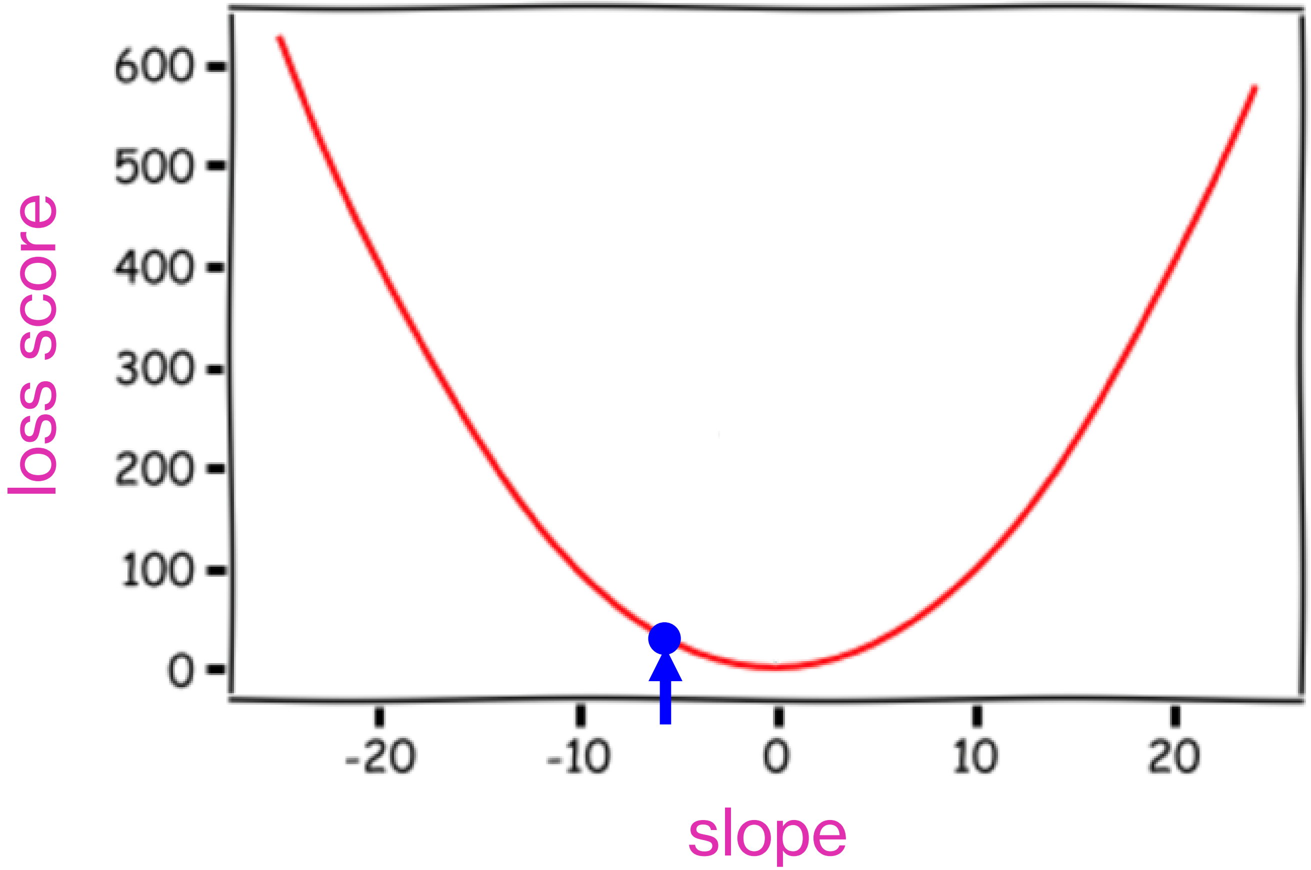
Gradient Descent Learning Algorithm

- Adjust the point. Compute the score.



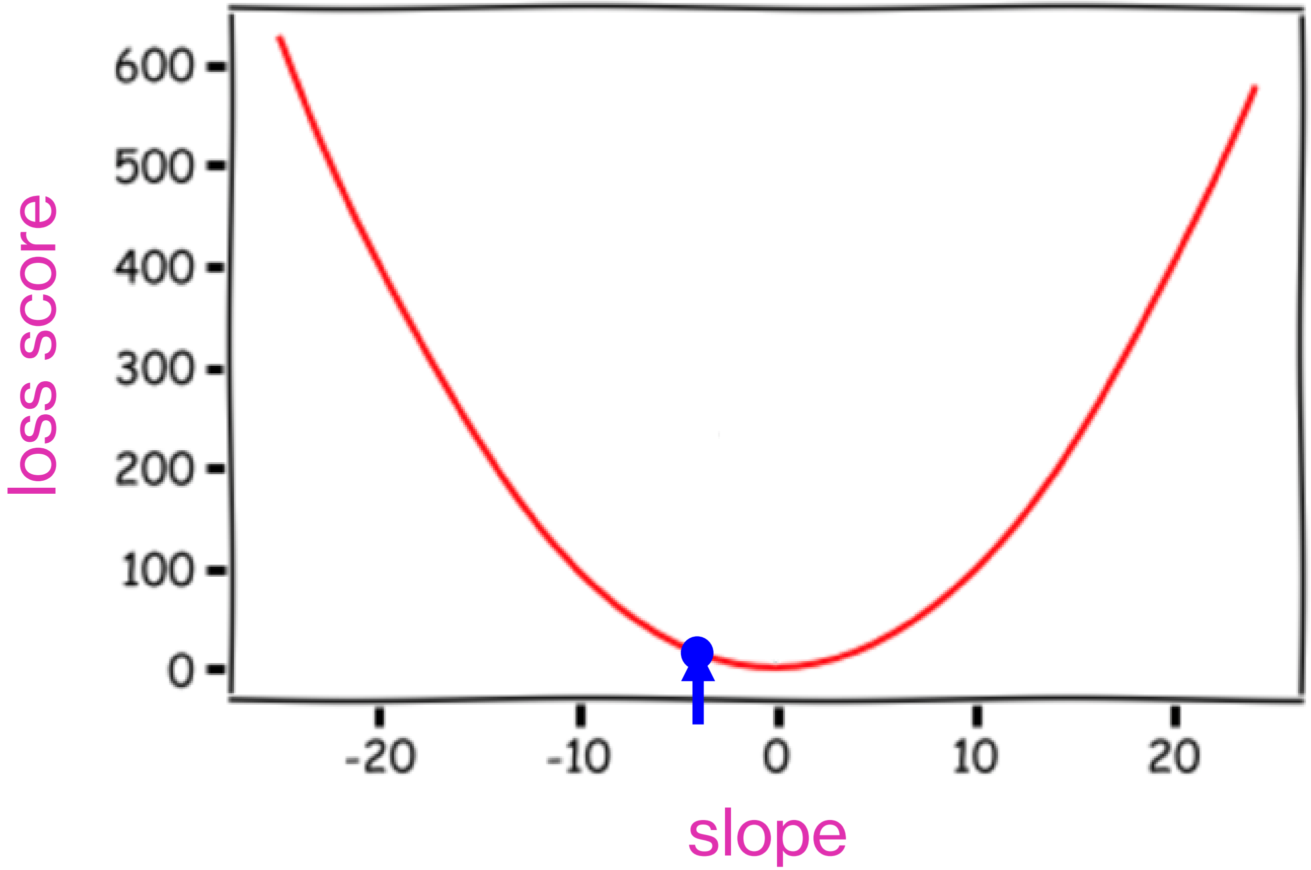
Gradient Descent Learning Algorithm

- Adjust the point. Compute the score.



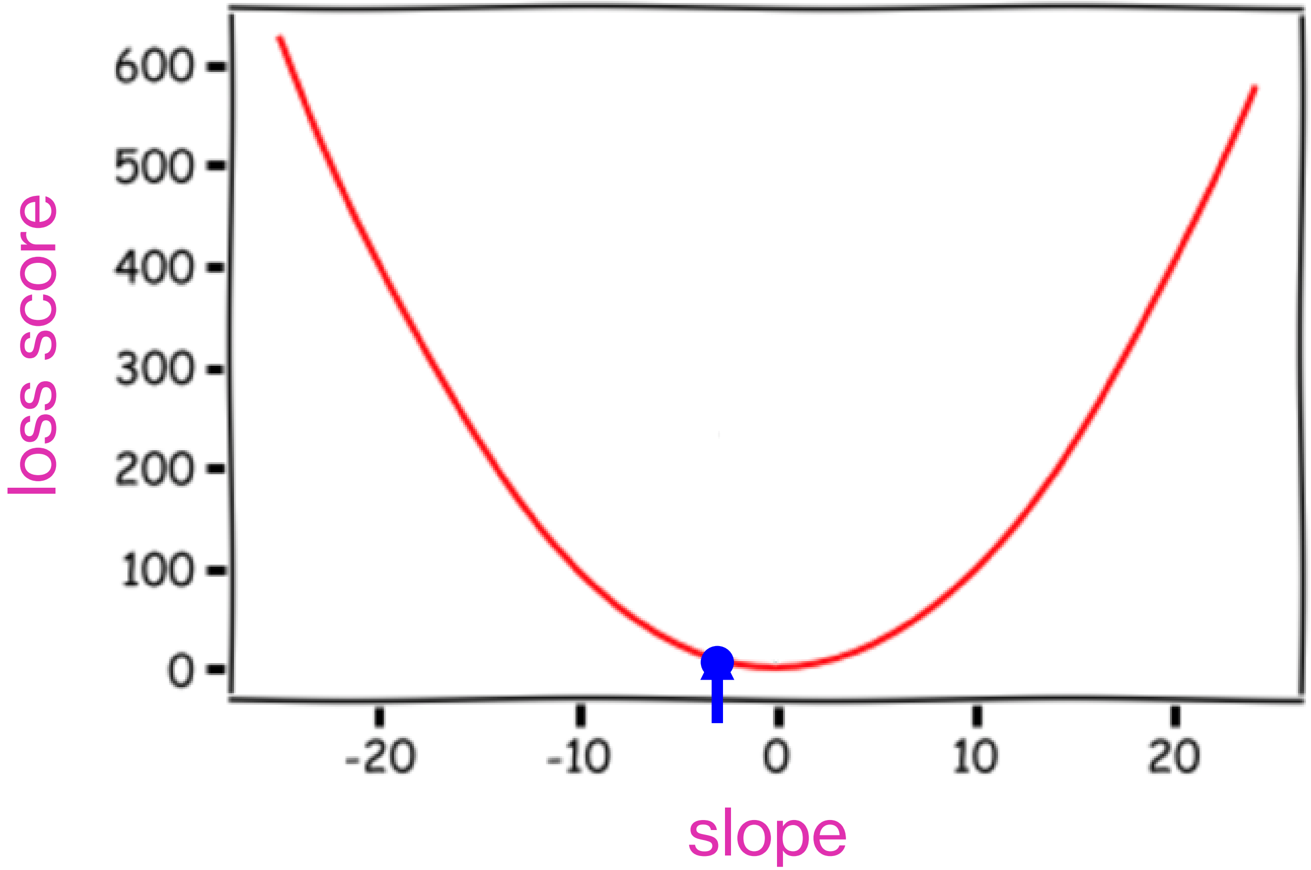
Gradient Descent Learning Algorithm

- Adjust the point. Compute the score.



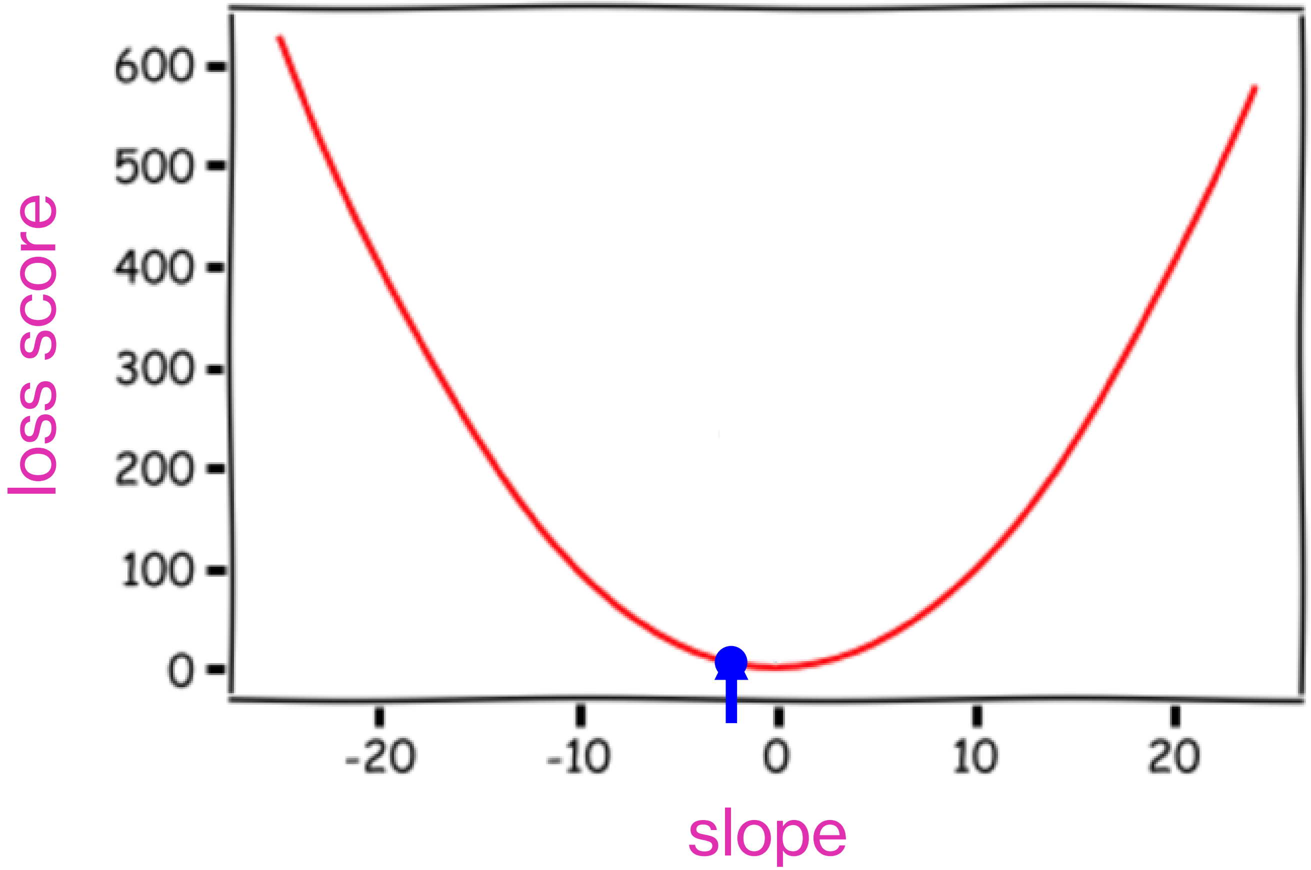
Gradient Descent Learning Algorithm

- Adjust the point. Compute the score.



Gradient Descent Learning Algorithm

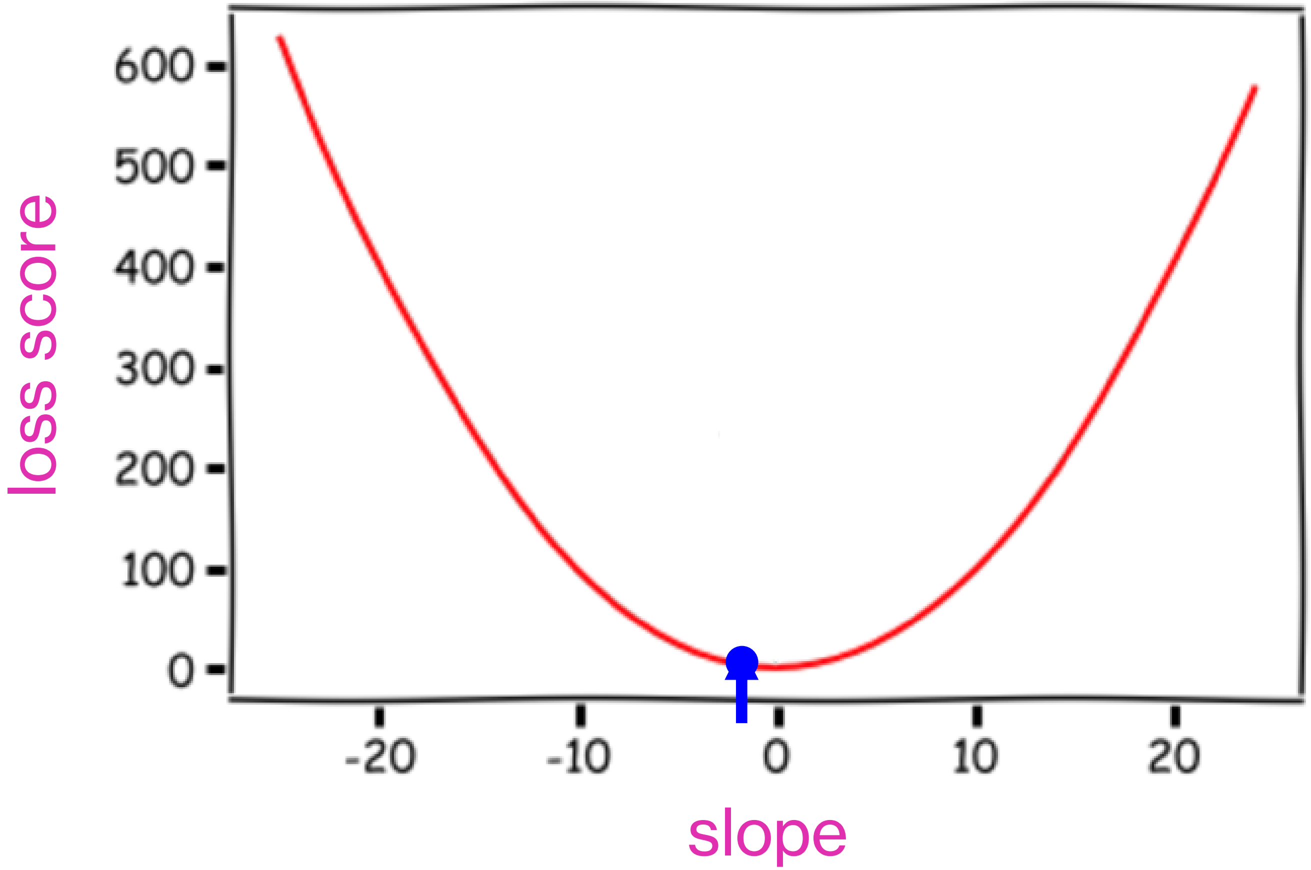
- Adjust the point. Compute the score.



Gradient Descent Learning Algorithm

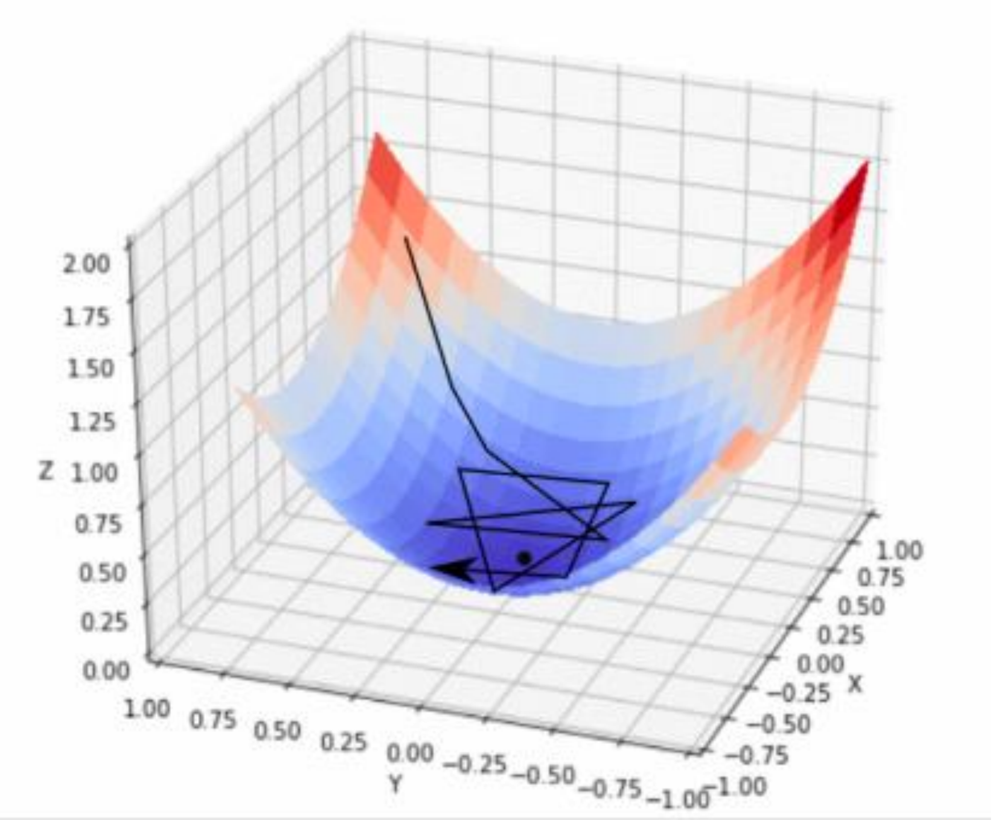
- Adjust the point. Compute the score.

- **Question:** how do we know which direction to move and how much?



Case of Multiple Parameters

- When we consider both w_1 and w_0, the graph of the loss function will look like this.
- **Gradient descent concept still applies**!



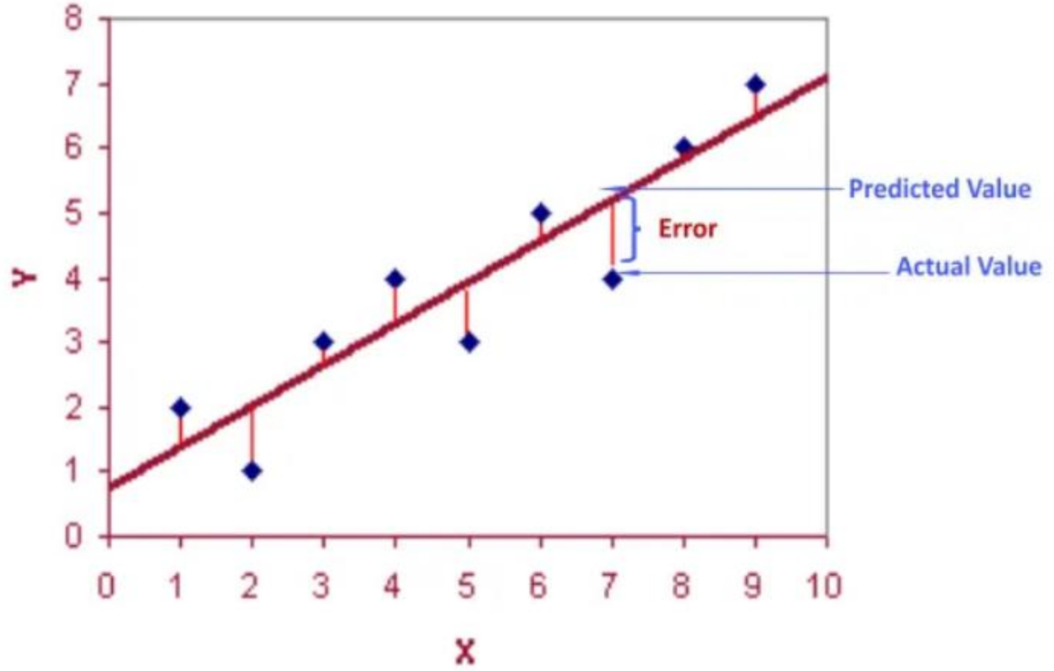
Extension to Multiple Features

- **Example:** We are considering not only the lot area, but also the floor space.
- $y = w_1 x_1 + w_2 x_2 + w_0$
  - features (lot area and floor space): x_2
  - label (price): y
- w_0, w_1, w_2 are the **parameters** of the model
- Analyzing the relationship between a single dependent variable against multiple independent variables.
- Can be extended to as many features as we want!
- The learning algorithm is called – Multiple Regression
- Same principles of linear regression apply

Evaluating Linear Regression Model

- We can use **Sum of Squared Error** to measure the performance of the model.
- SSE finds the difference between the actual and the predicted values.
- RMSE indicates average model prediction error
- The lower values indicate a better fit.
- It is measured in same units as the target variable.

$$SSE = \sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2$$



Acknowledgments

- Previous STINTSY slides by the following instructors:
  - Courtney Ngo
  - Arren Antioquia