Linear Regression

Original Slides by:
Courtney Anne Ngo
Daniel Stanley Tan, PhD
Arren Antioquia

Updated (AY 2023 – 2024 T3) by:
Thomas James Tiam-Lee, PhD

College of
Computer Studies

Linear Regression

- **Supervised**, **regression** learning algorithm

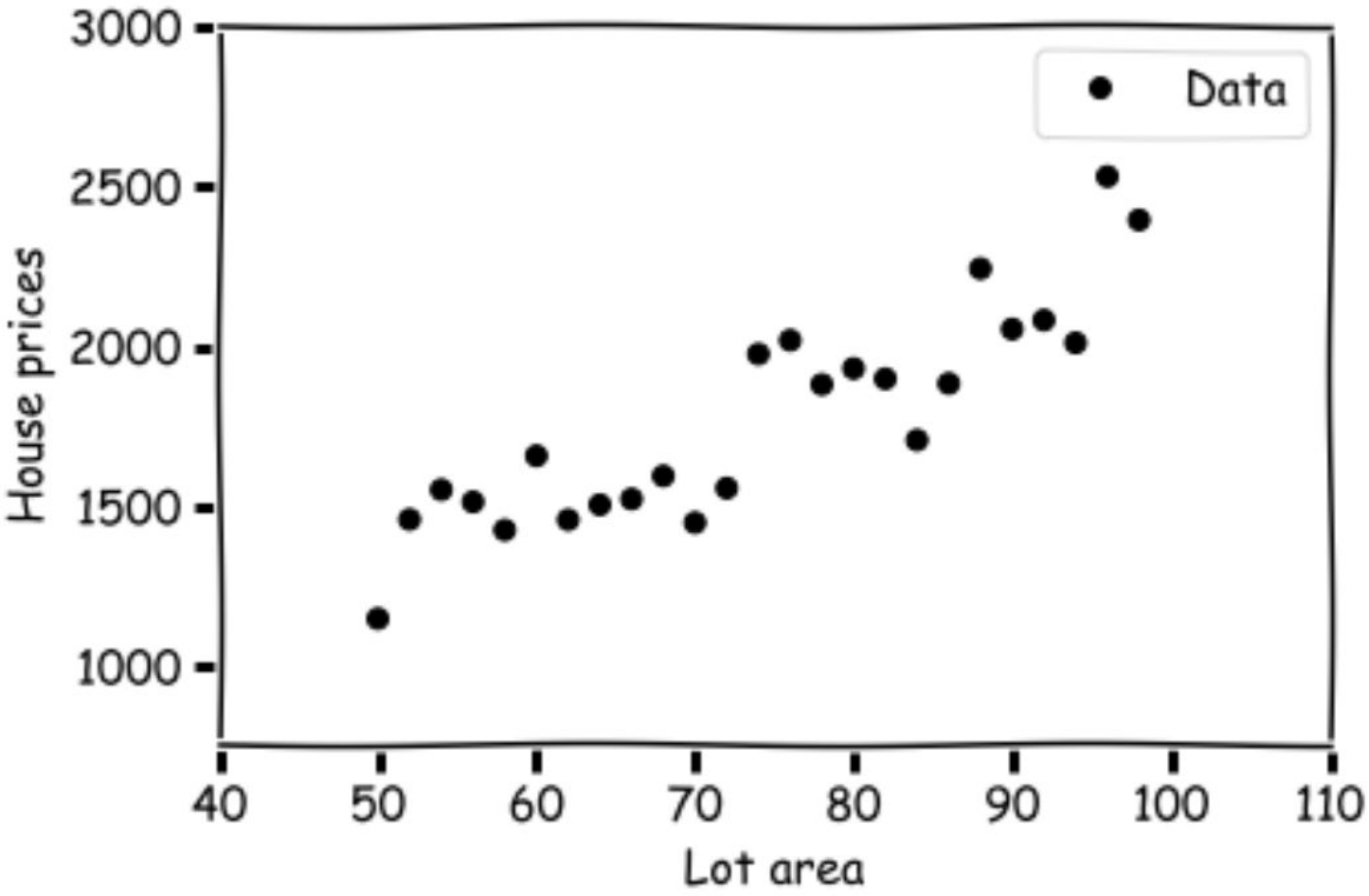
- Example:
  - predict the price of the house given its lot area

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
70	1448
72	1556

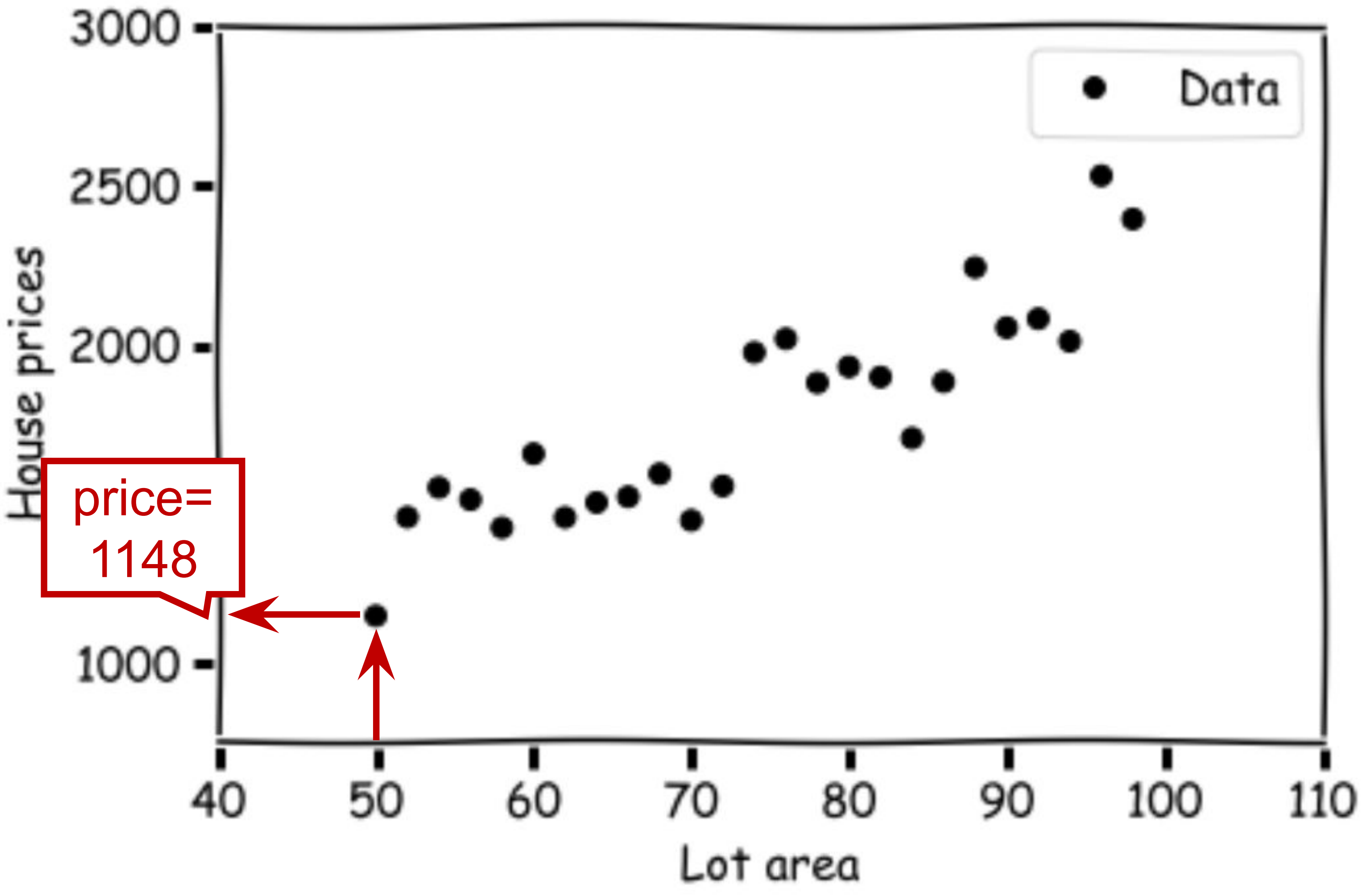
The Data

Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425
60	1657
62	1457
64	1504
66	1522
68	1594
70	1448
72	1556



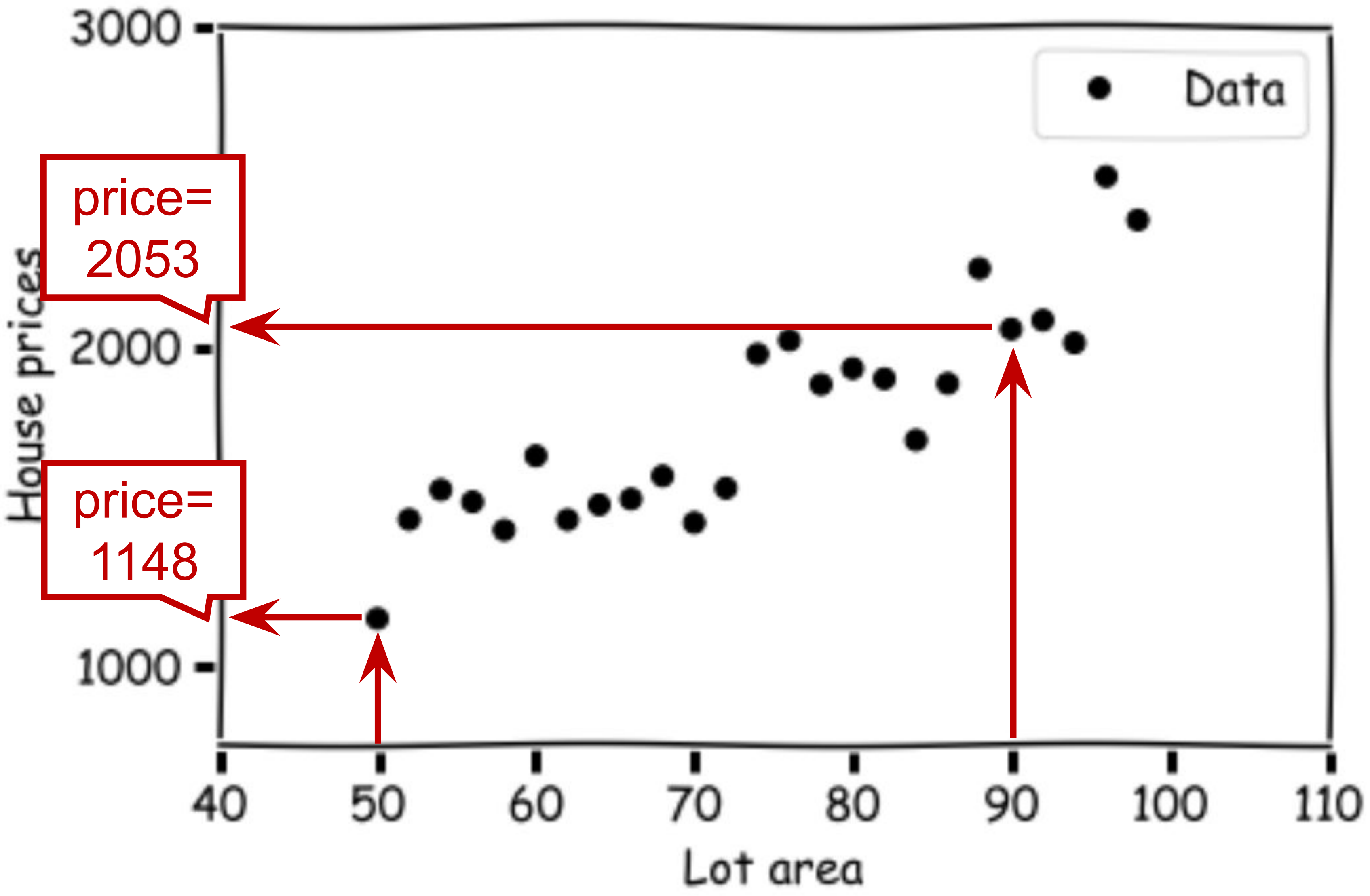
The Data

Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425
60	1657
62	1457
64	1504
66	1522
68	1594
70	1448
72	1556



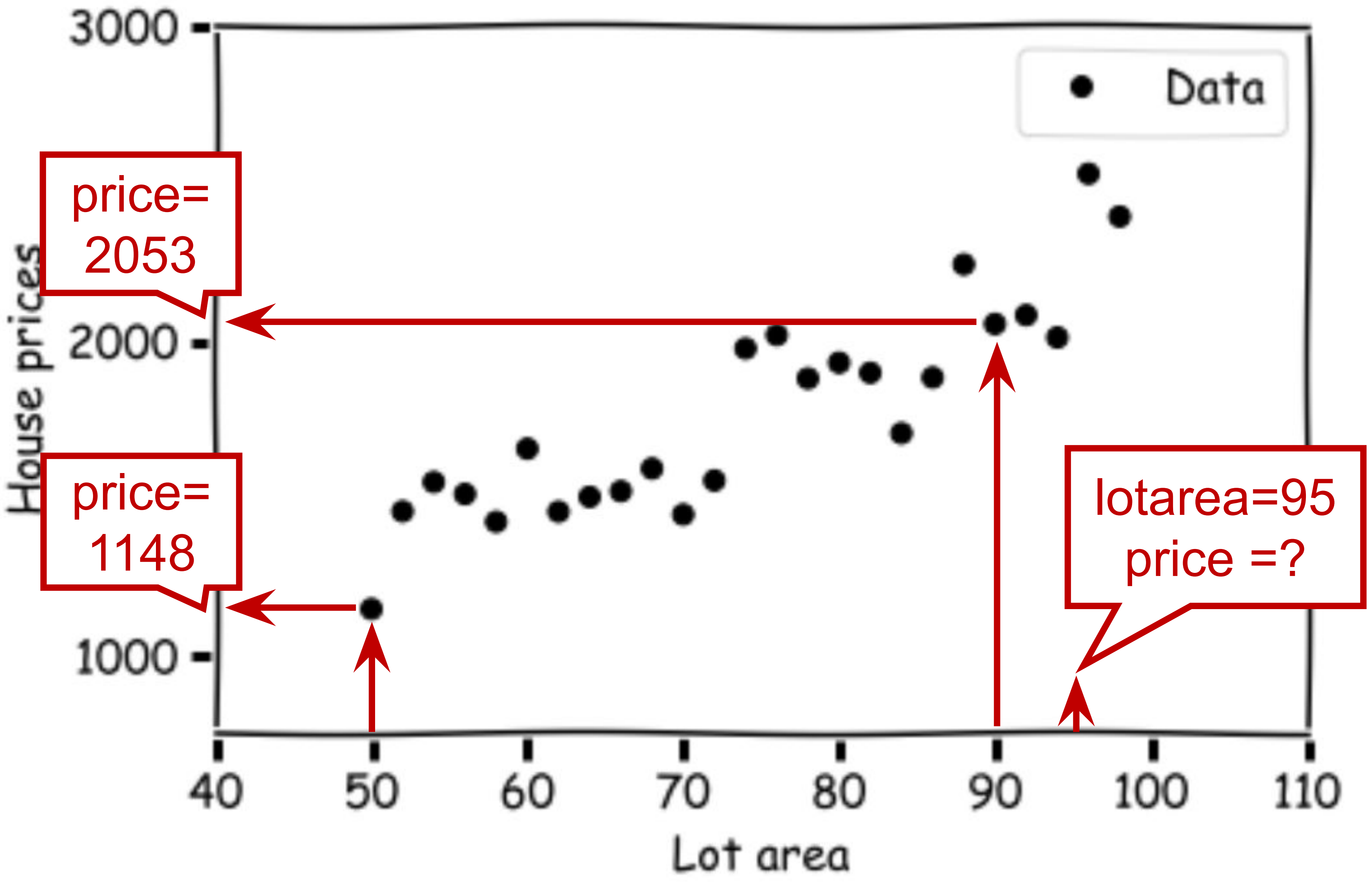
The Data

Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425
60	1657
62	1457
64	1504
66	1522
68	1594
70	1448
72	1556



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
70	1448
72	1556



Linear Regression Model

- Recall: in ML, the model is the **representation of the “patterns” that were learned** in the data.

- In linear regression with 1 feature, the model can be thought of as a **line**.

Linear Regression Model

- Equation of a line:

$$\hat{y} = \boldsymbol{mx} + \boldsymbol{b}$$

Linear Regression Model

- Equation of a line:

$$\hat{y} = \boldsymbol{\theta_1 x} + \boldsymbol{\theta_0}$$

Linear Regression Model

- Equation of a line:

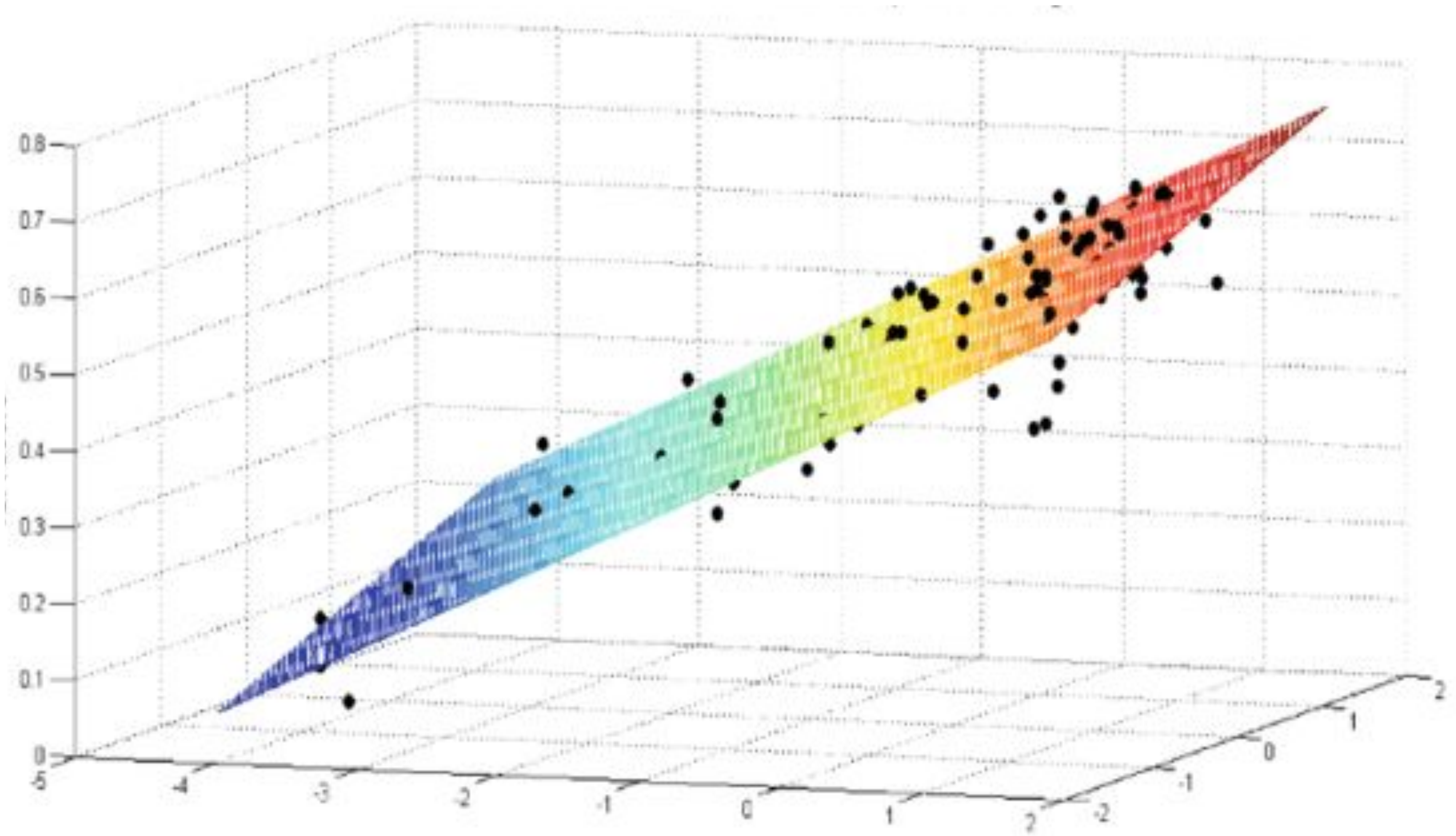
$$\hat{y} = \boldsymbol{\theta_1} x + \boldsymbol{\theta_0}$$

parameters of the model

θ_1: slope of the line
θ_0: intercept of the line

Linear Regression Model

- Extension to multiple features:



- With 2 features, the line becomes a plane!
- With more than 2 features, it becomes a hyperplane!

Linear Regression Model

- Extension to multiple features:

$$\hat{y} = \boldsymbol{\theta_1} \boldsymbol{x_1} + \boldsymbol{\theta_2} \boldsymbol{x_2} + \boldsymbol{\theta_0}$$

Linear Regression Model

- Extension to multiple features:

$$\hat{y} = \boldsymbol{\theta_1} x_1 + \boldsymbol{\theta_2} x_2 + \cdots + \boldsymbol{\theta_d} x_d + \boldsymbol{\theta_0}$$

Linear Regression Model

- For convenience, we add a feature that is always 1.

$$\hat{y} = \boldsymbol{\theta}_1 x_1 + \boldsymbol{\theta}_2 x_2 + \cdots + \boldsymbol{\theta}_d x_d + \boldsymbol{\theta}_0 x_0$$

where x_0 **is always 1**.

Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425

X0	Lot area	House Price
1	50	1148
1	52	1458
1	54	1551
1	56	1513
1	58	1425

Linear Regression Model

- For convenience, we add a feature that is always 1.

$$\hat{y} = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_d x_d$$

where x_0 **is always 1**.

Lot area	House Price
50	1148
52	1458
54	1551
56	1513
58	1425

X0	Lot area	House Price
1	50	1148
1	52	1458
1	54	1551
1	56	1513
1	58	1425

Linear Regression Model

- This allows us to vectorize the operation

$$\hat{y} = \boldsymbol{\theta}_0 \boldsymbol{x}_0 + \boldsymbol{\theta}_1 \boldsymbol{x}_1 + \boldsymbol{\theta}_2 \boldsymbol{x}_2 + \cdots + \boldsymbol{\theta}_d \boldsymbol{x}_d$$

$$\downarrow$$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x}$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \cdots \\ \theta_d \end{bmatrix} \quad x = \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_d \end{bmatrix}$$

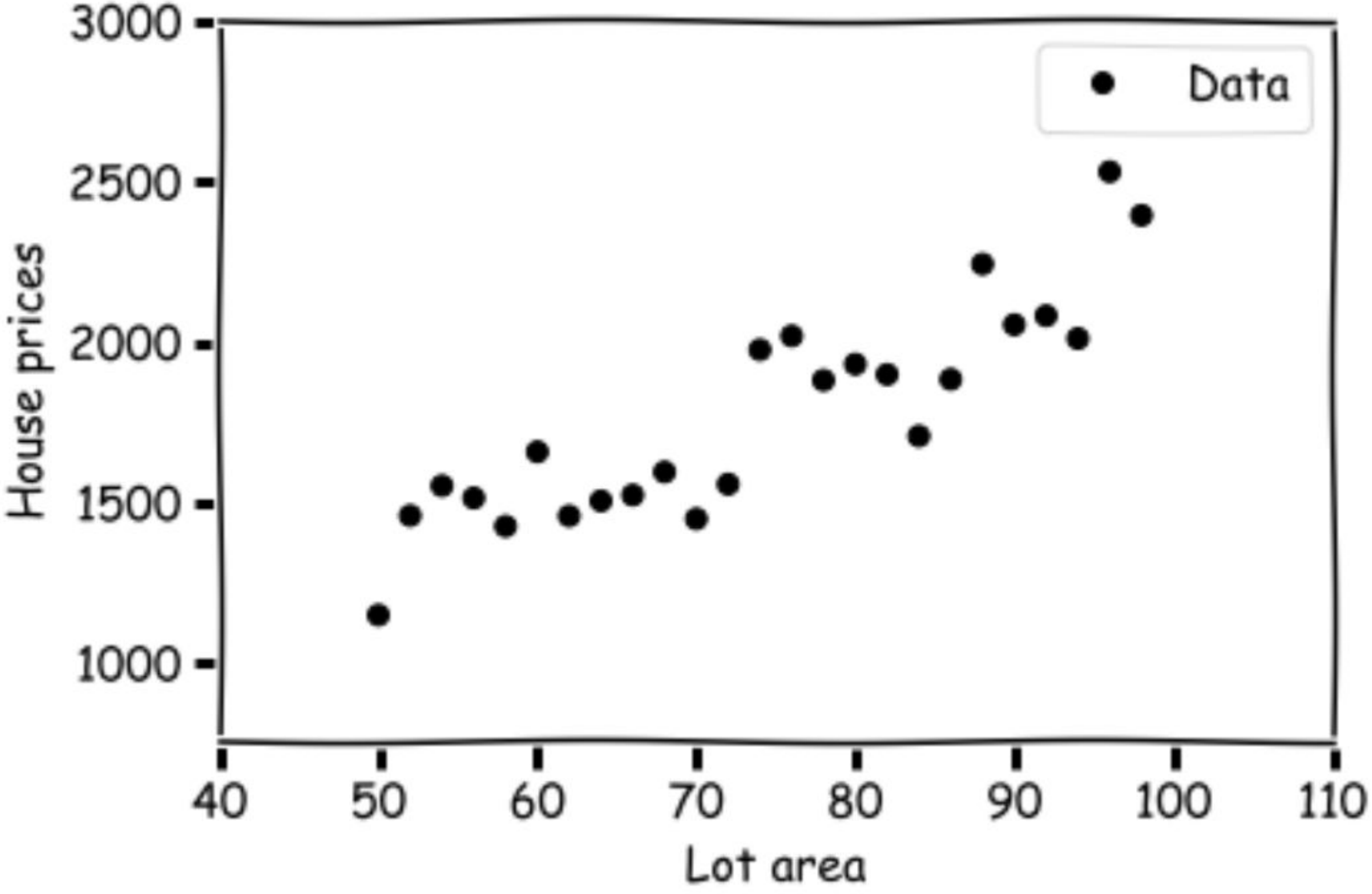
Linear Regression Model

- We assume the input and output are related with the following equation:

$$y = \theta^T x + \varepsilon$$

- Where $\varepsilon \sim N(0, \sigma^2)$ represents the measurement error or some other random noise

Linear Regression Model



How to Choose the Parameters?

- Our goal is the come up with a good set of parameters, such that our model “fits” the data.

- Given a set of parameters θ, how do we **objectively measure** how “good” the model is?

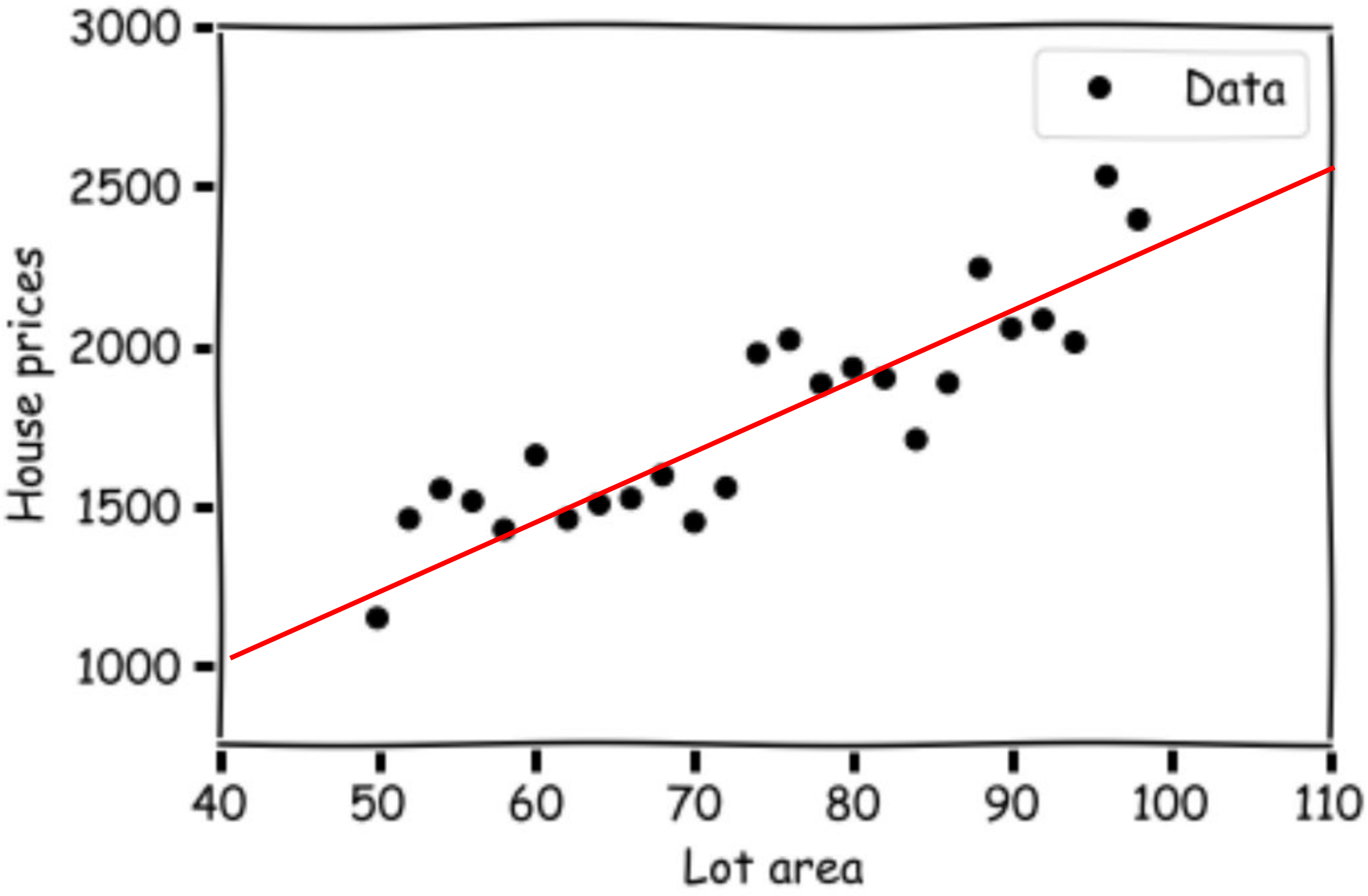
Loss Function

- Also known as **objective function** or **cost function**.

- **Input:** parameters of a model + a dataset
- **Returns:** a **numerical value** representing how well the model fits the dataset

  - $l(\theta, X, y) =$ loss (the lower value, the better)

Linear Regression Model



Linear Regression Loss Function

$$l(\theta, X, y) = \frac{1}{n}\sum(\hat{y} - y)^2$$

number of instances

prediction of model

correct answer

Linear Regression Loss Function

$$l(\theta, X, y) = \frac{1}{2n} \sum (\hat{y} - y)^2$$

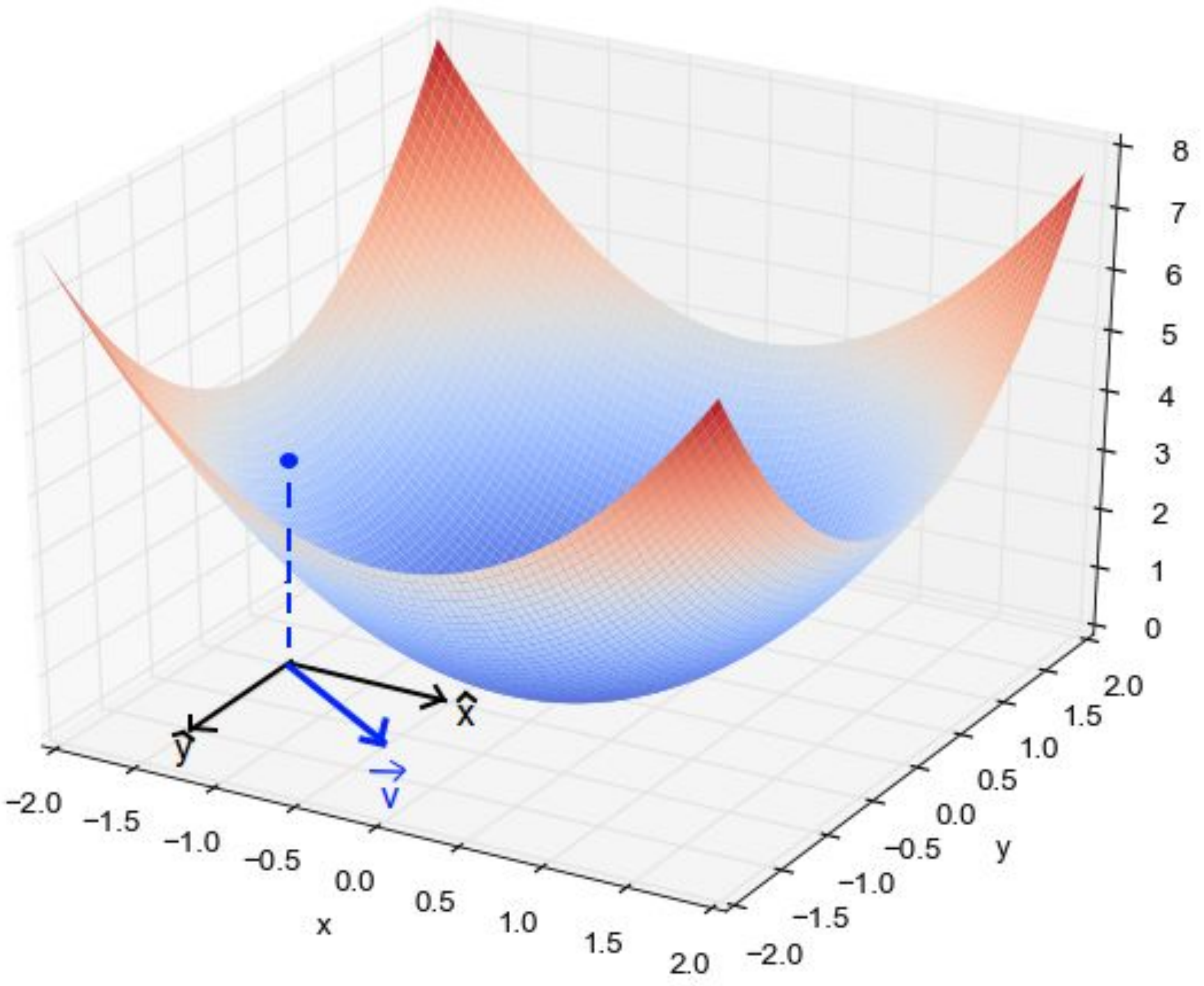
number of instances

prediction of model

correct answer

We add 2 here, for convenience later (this does not affect the intended purpose of the loss function)

Loss Landscape



Optimization Problem

$$\underset{\theta}{\operatorname{argmin}}\, l(\theta, X, y)$$

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{2n} \sum (\hat{y} - y)^2$$

Find the value of θ such that the loss function will return the smallest possible value.

Derivative of Loss Function

$$\frac{\partial}{\partial \theta}\frac{1}{2n}\sum(\hat{y}-y)^2$$

$$\theta = \begin{bmatrix}\theta_0\\ \theta_1\\ \theta_2\end{bmatrix}$$

How does θ_0 affect the loss?

$$\frac{\partial}{\partial \theta_0}\frac{1}{2n}\sum(\hat{y}-y)^2 = \frac{1}{2n}\frac{\partial}{\partial \theta_0}\sum(\hat{y}-y)^2 = \frac{1}{2n}\sum\frac{\partial}{\partial \theta_0}(\hat{y}-y)^2$$

$$= \frac{1}{2n}\sum 2(\hat{y}-y)\frac{\partial}{\partial \theta_0}(\hat{y}-\mathrm{y})$$

$$= \frac{1}{2n}\sum 2(\hat{y}-y)\frac{\partial}{\partial \theta_0}(\theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_d x_d - y)$$

$$= \frac{1}{2n}\sum 2(\hat{y}-y)(x_0) = \frac{1}{n}\sum(\hat{y}-y)(x_0)$$

Derivative of Loss Function

$$\frac{\partial}{\partial\theta}\frac{1}{2n}\sum(\hat{y}-y)^2$$

$$\theta = \begin{bmatrix}\theta_0\\ \theta_1\\ \theta_2\end{bmatrix}$$

How does θ_1 affect the loss?

$$\frac{\partial}{\partial\theta_1}\frac{1}{2n}\sum(\hat{y}-y)^2 = \frac{1}{2n}\frac{\partial}{\partial\theta_1}\sum(\hat{y}-y)^2 = \frac{1}{2n}\sum\frac{\partial}{\partial\theta_1}(\hat{y}-y)^2$$

$$= \frac{1}{2n}\sum 2(\hat{y}-y)\frac{\partial}{\partial\theta_1}(\hat{y}-\mathrm{y})$$

$$= \frac{1}{2n}\sum 2(\hat{y}-y)\frac{\partial}{\partial\theta_1}(\theta_0 x_0+\theta_1 x_1+\cdots+\theta_d x_d-y)$$

$$= \frac{1}{2n}\sum 2(\hat{y}-y)(x_1) = \frac{1}{n}\sum(\hat{y}-y)(x_1)$$

Derivative of Loss Function

$$\frac{\partial}{\partial \theta}\frac{1}{2n}\sum(\hat{y}-y)^2$$

$$\theta = \begin{bmatrix}\theta_0\\ \theta_1\\ \theta_2\end{bmatrix}$$

How does θ_i affect the loss?

$$\frac{\partial}{\partial \theta_i}\frac{1}{2n}\sum(\hat{y}-y)^2 = \frac{1}{2n}\frac{\partial}{\partial \theta_i}\sum(\hat{y}-y)^2 = \frac{1}{2n}\sum\frac{\partial}{\partial \theta_i}(\hat{y}-y)^2$$

$$= \frac{1}{2n}\sum 2(\hat{y}-y)\frac{\partial}{\partial \theta_i}(\hat{y}-\mathrm{y})$$

$$= \frac{1}{2n}\sum 2(\hat{y}-y)\frac{\partial}{\partial \theta_i}(\theta_0 x_0+\theta_1 x_1+\cdots+\theta_d x_d - y)$$

$$= \frac{1}{2n}\sum 2(\hat{y}-y)(x_i) = \frac{1}{n}\sum(\hat{y}-y)(x_i)$$

Derivative of Loss Function

$$\frac{\partial}{\partial\theta}\frac{1}{2n}\sum(\hat{y}-y)^2$$

$$\theta = \begin{bmatrix}\theta_0\\ \theta_1\\ \theta_2\end{bmatrix}$$

To summarize...

$$\frac{\partial}{\partial\theta} l(\theta, X, y) = \left[\frac{1}{n}\sum(\hat{y}-y)(x_0), \frac{1}{n}\sum(\hat{y}-y)(x_1), \ldots, \frac{1}{n}\sum(\hat{y}-y)(x_d)\right]$$

In vector form:

$$\frac{\partial}{\partial\theta} l(\theta, X, y) = \frac{1}{n}(\boldsymbol{X\theta}-\boldsymbol{y})^T\boldsymbol{X}$$

Two Solutions to Linear Regression

1. Analytical Solution
2. Gradient Descent

Analytical Solution

- Set the derivative to 0 and solve for θ

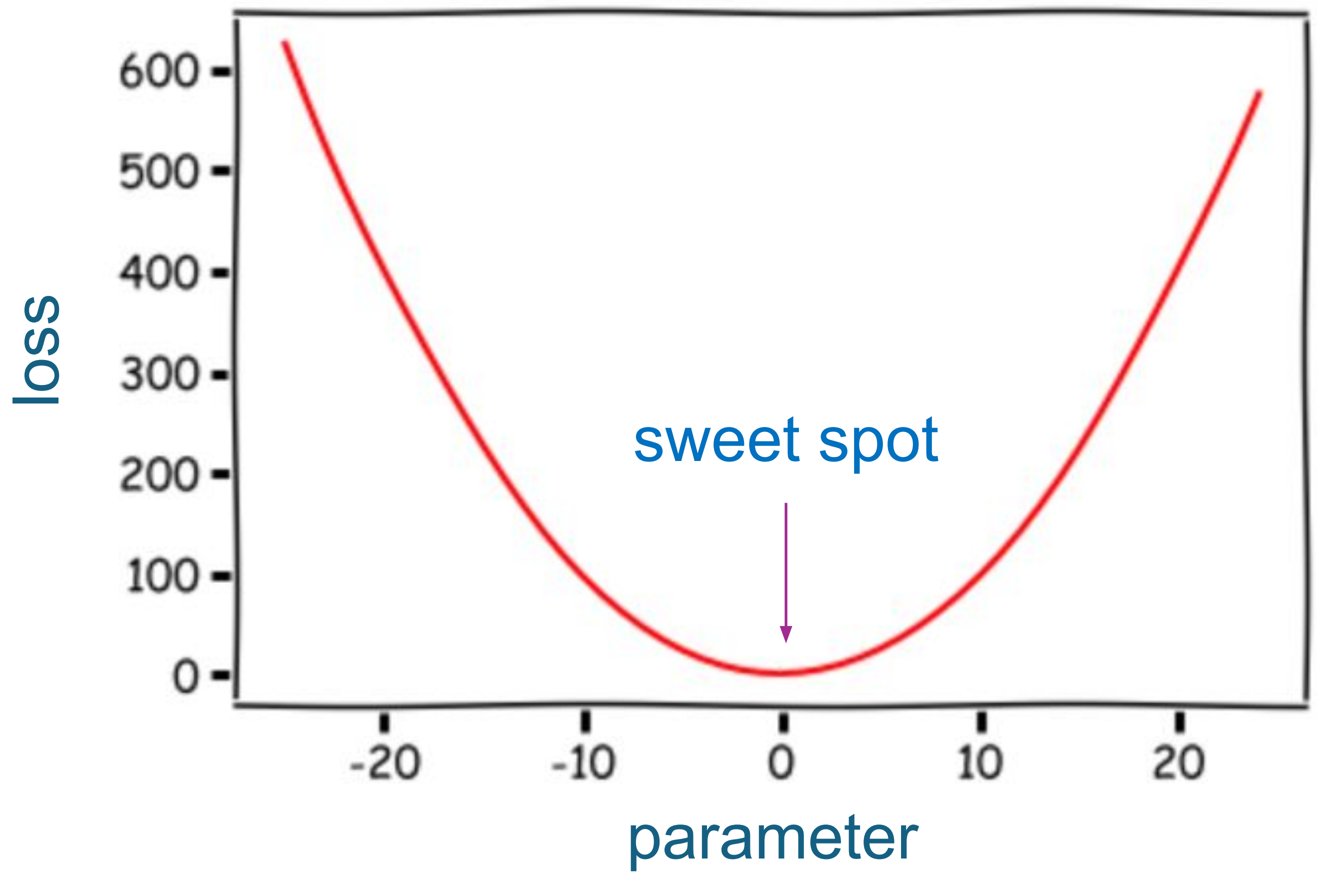
- $\frac{1}{n}(X\theta - y)^T X = 0$
- $(X\theta - y)^T X = 0$
- $X^T X\theta - X^T y = 0$
- $X^T X\theta = X^T y$
- $\theta = \left(X^T X\right)^{-1} X^T y$

Ignore $\frac{1}{n}$ for simplicity

A^{-1} means the matrix inverse of A

Gradient Descent

- Try out different parameters until you reach the lowest point.
- Use the gradient (derivative evaluated at the current point) to guide the exploration.



Gradient Descent

procedure gradientdescent(θ):
while not converged do:

$$\theta := \theta - \alpha \frac{\partial}{\partial \theta} l(\theta)$$

return θ

α is the
learning rate,
determines how large the
update will be

$\frac{\partial}{\partial \theta_i} l(\theta)$ is the
gradient of the loss

The Learning Rate (α) Hyperparameter

- Controls how “fast” the learning happens.
- Common values are 0.01, 0.001, 0.0001, and so on…

- **If α is too small…**
  - Convergence may take too long
- **If α is too large…**
  - Algorithm may overshoot the minimum

- Different values of α can be tried out as part of hyperparameter tuning.

Stochastic, Mini-Batch Gradient Descent

Gradient Descent	Stochastic GD	⭐ Mini-Batch GD
Updates θ based on gradient of the **whole dataset**	Updates θ based on gradient of **one data instance**	Updates θ based on gradient of a **subset of the whole dataset**
Runs slow	Runs fast, but jittery	Runs faster than GD, and path is not as jittery as stochastic GD
1 iter = N instances N = size of whole dataset	1 iter = 1 instance	1 iter = M instances, M = size of batch

Standardization of Features

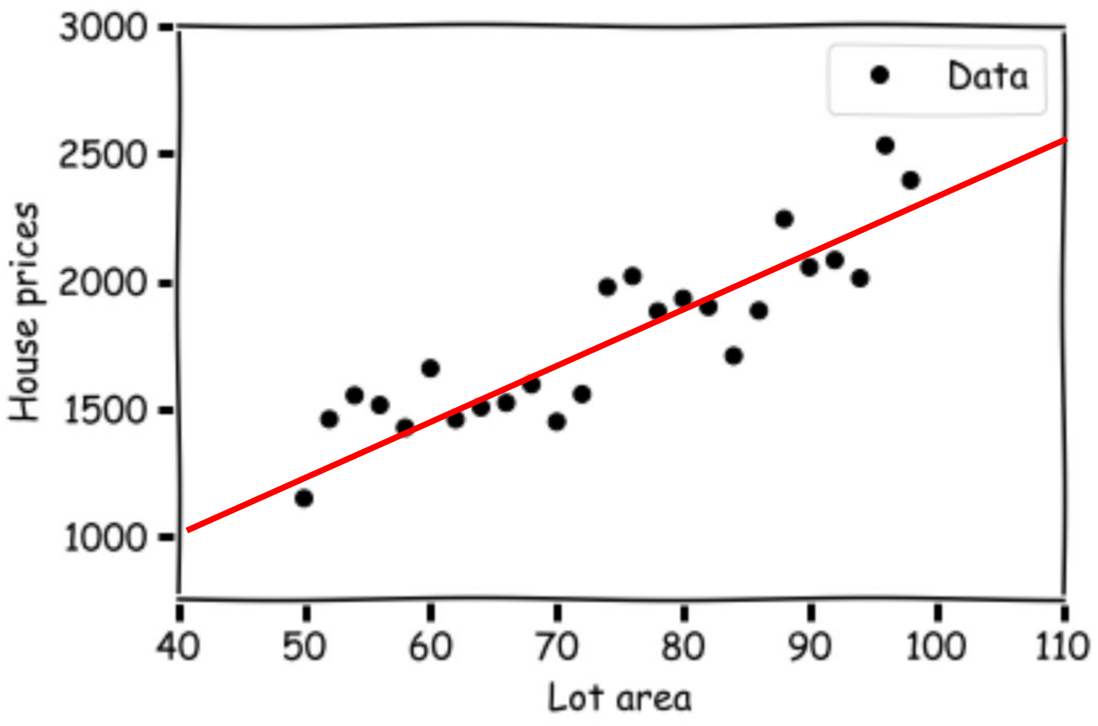
- Subtract each value by the mean of that feature, then divide it by the standard deviation of that feature.

$$x_{norm} = \frac{x - \mu}{\sigma}$$

- Sometimes necessary to prevent underflow or overflow

Evaluating Regression Tasks (test set)

- Mean sum of squared error (MSE)
  - $\frac{1}{n}\Sigma(\hat{y} - y)^2$
- Mean absolute error (MAE)
  - $\frac{1}{n}\Sigma|\hat{y} - y|$
- Root mean squared error (RMSE)
  - $\sqrt{\frac{1}{n}\Sigma(\hat{y} - y)^2}$
- Coefficient of Determination (R^2)
  - $\frac{\Sigma(\hat{y}-y)^2}{\Sigma(y-\bar{y})^2}$, where $\bar{y}$ is the mean of the labels



Linear Regression Advantages and Disadvantages

- **Advantages**
  - Relatively fast to train
  - Relatively fast to test
- **Disadvantages**
  - Can only be used for regression tasks
  - Features may sometimes need to be standardized
  - Features may sometimes need to be transformed