



Artificial Intelligence and Machine Learning

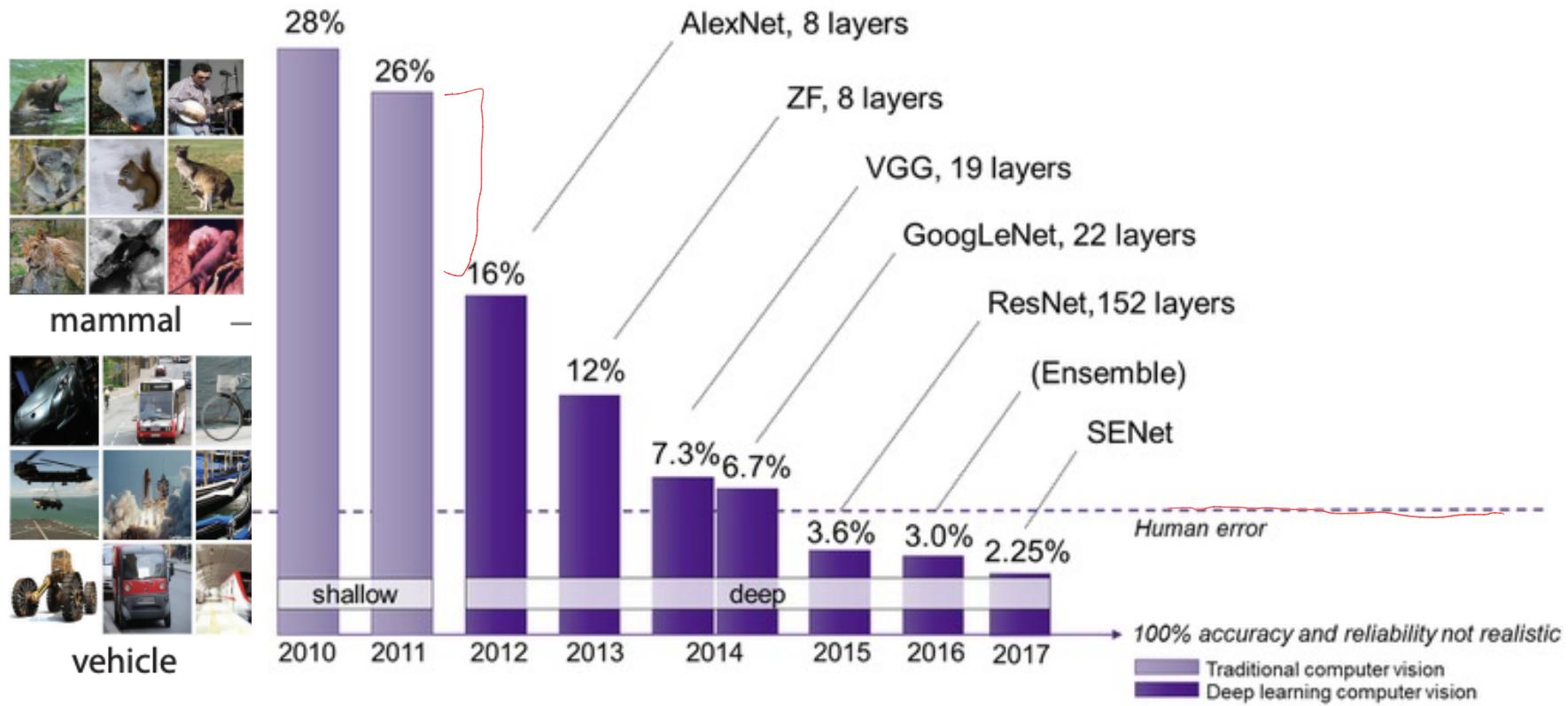
Linear Regression

Lecture 1: Outline

- Introduction to ML
- Linear Regression
- Optimization
- Linear Regression: Probabilistic Interpretation
- Bias-Variance Tradeoff
- Regularization

Introduction to ML

- Motivation



Visual recognition (Error on Y-



Introduction to ML

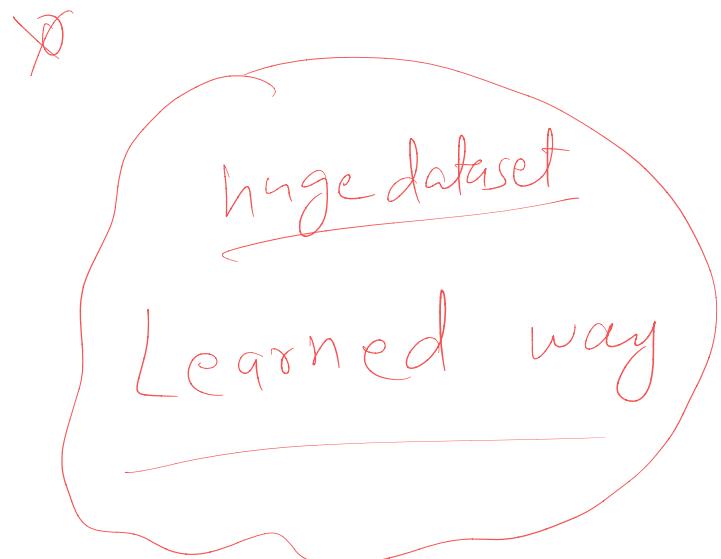
- Motivation

Model Based Approach

- Humans observe physical processes and produce mathematical models.

Introduction to ML

- Motivation

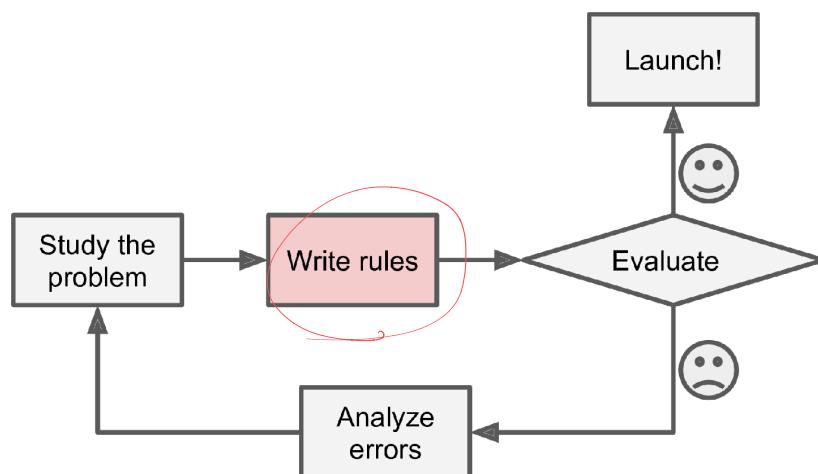


Model Based Approach

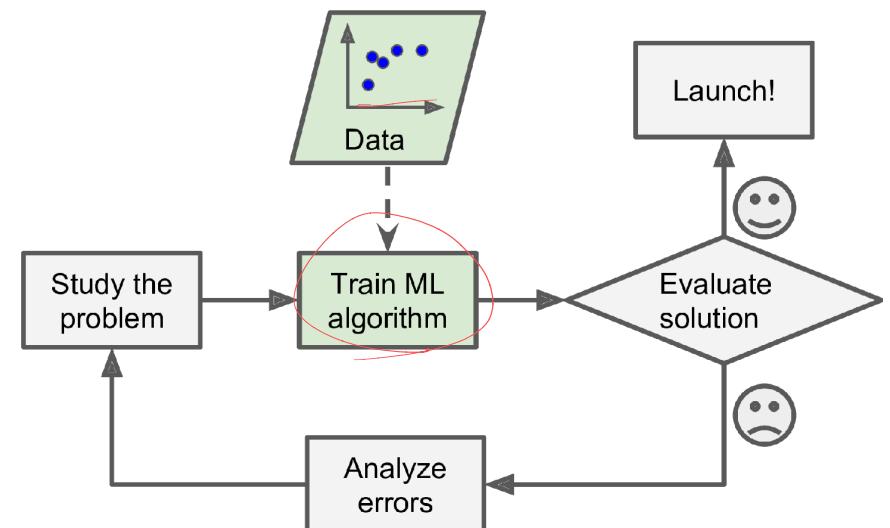
- Humans observe physical processes and produce mathematical models.
- **Limited by human capacity to process data**

Introduction to ML

- **Machine Learning** is the science (and art) of programming computers so they can learn from data.

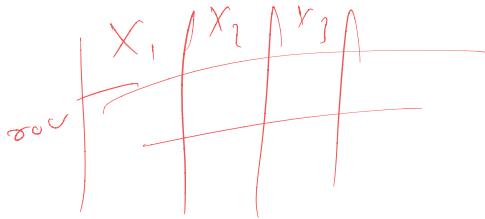


The traditional approach



The Machine Learning approach

Data Types

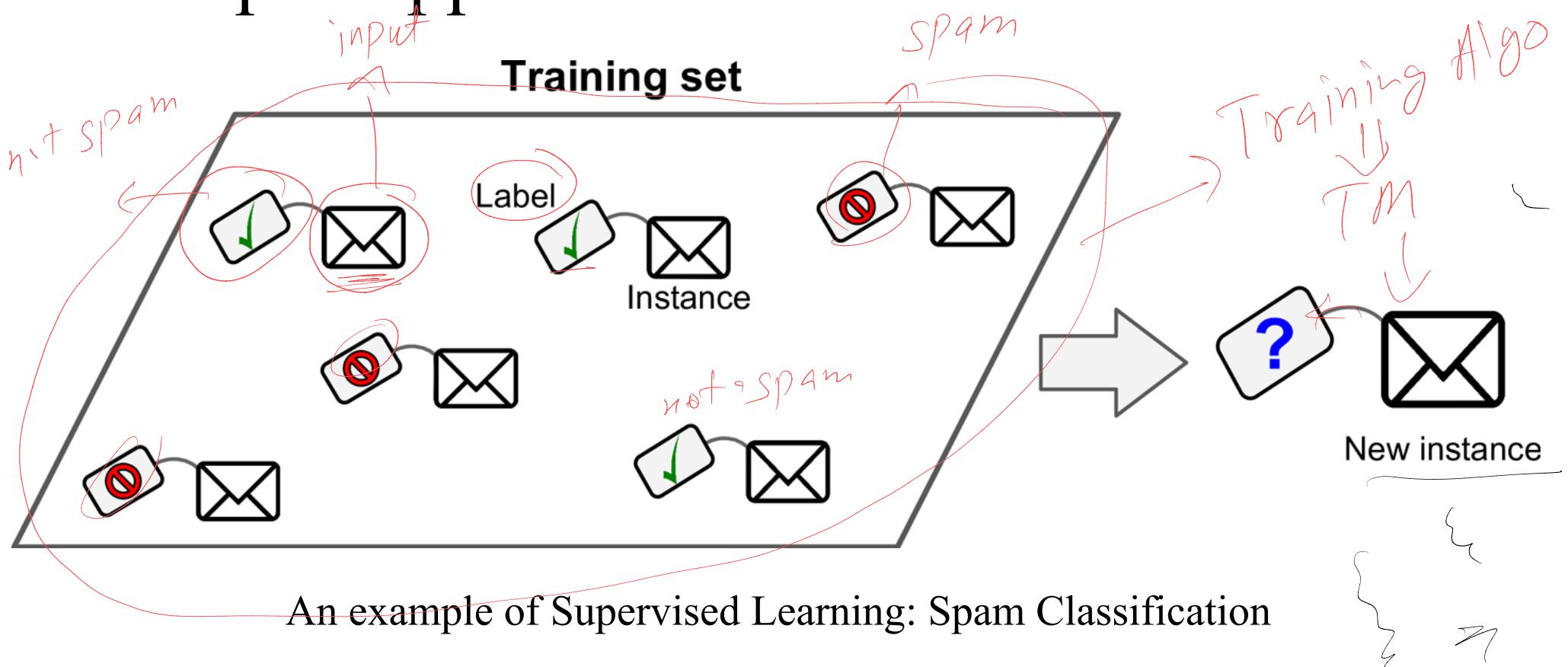


- **Tabular Data** (e.g., spreadsheets, databases)
 - Note: Columns are called **Features**. Rows are called **Samples**.
- **Time-Series Data** (e.g., stock prices, weather forecasts, IoT sensor data)
- **Text Data** (Natural Language Processing, e.g., emails, social media posts, documents)
- **Images and Videos** (Computer Vision, e.g., medical imaging, surveillance, facial recognition)
- **Audio Data** (Speech Recognition, Music Processing, e.g., voice commands, podcasts, sound classification)

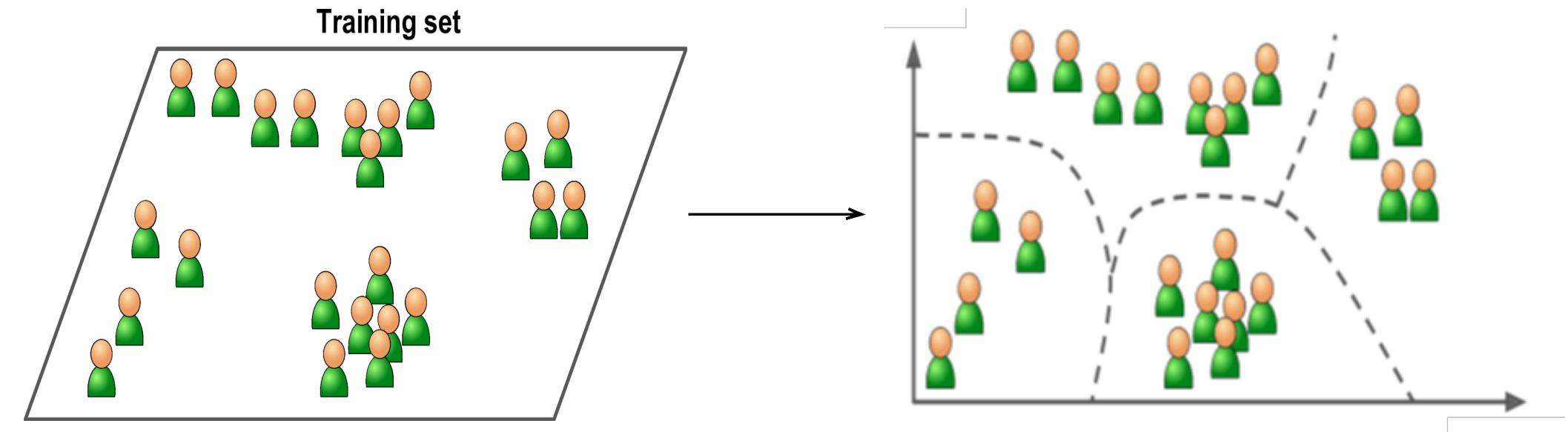
ML Algorithms Types

- **Supervised:** There is a target we want to predict.
 - **Regression:** Predict continuous value (e.g. house prices).
 - **Classification:** Predict discrete value (e.g. spam/not-spam).
- **Unsupervised:** There is no target. We are interested in things like:
 - **Clustering:** Grouping
 - **Dimensionality Reduction:** Reducing the Dimensions
 - **Anomaly Detection:** Detecting outliers

Example Application



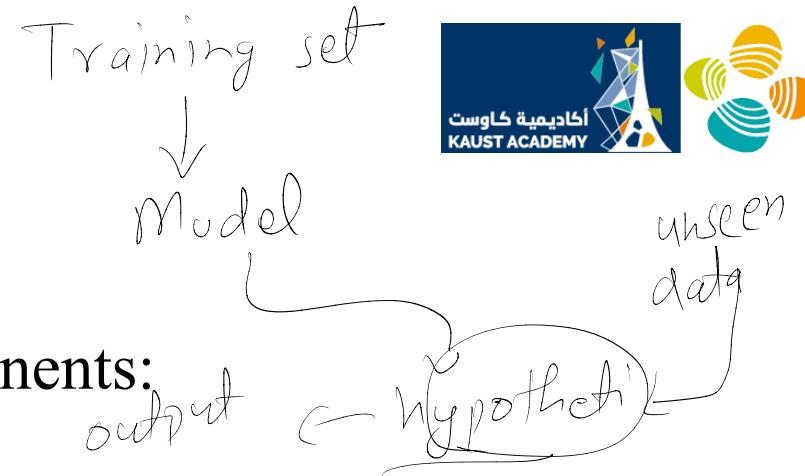
Example Application



An example of Unsupervised Learning: Clustering

How Does ML Work?

- Any ML system consists of three main components:



Hypothesis (Model): The function that approximates the target.

- E.g. Linear Regression, Logistic Regression, SVM, Decision Trees, NN,...

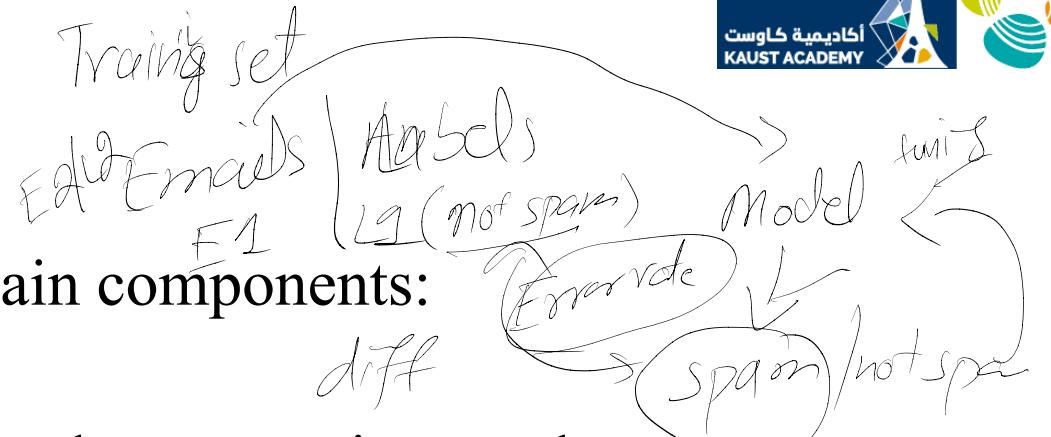
Optimizer: The mechanism for improving predictions of our model.

Loss Function: The measure of how wrong the predictions are.

Objective
function

How Does ML Work?

- Any ML system consists of three main components:



Hypothesis (Model): The function that approximates the target.

- E.g. Linear Regression, Logistic Regression, SVM, Decision Trees, NN,...

Optimizer: The mechanism for improving predictions of our model.

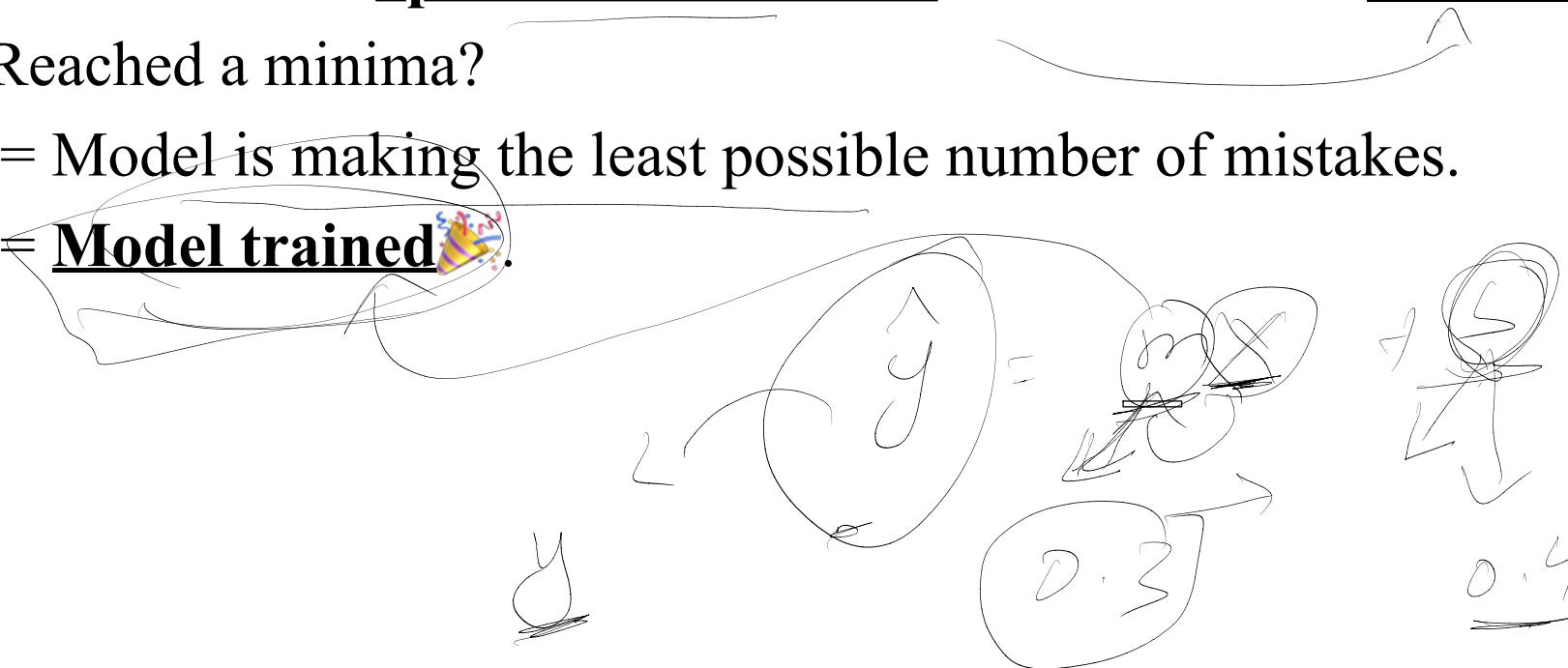
Loss Function: The measure of how wrong the predictions are.

How Does ML Work?

- How are they related to each other?🤔

How Does ML Work?

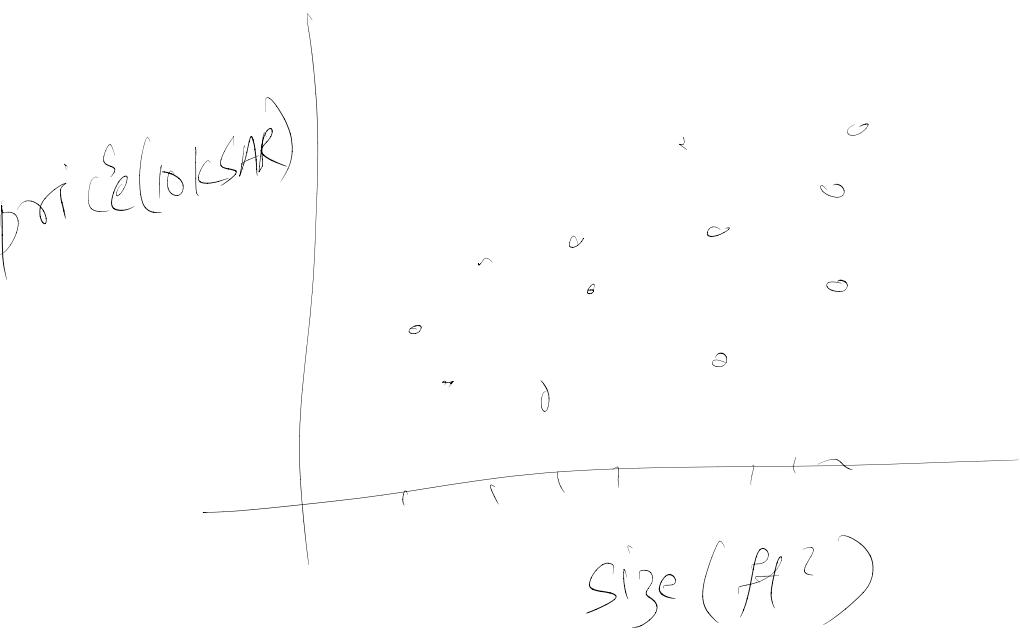
- We firstly define our task (classification/regression) then choose an appropriate model.
- We will use an optimization method to minimize the loss function.
- Reached a minima?
 - = Model is making the least possible number of mistakes.
 - = Model trained.



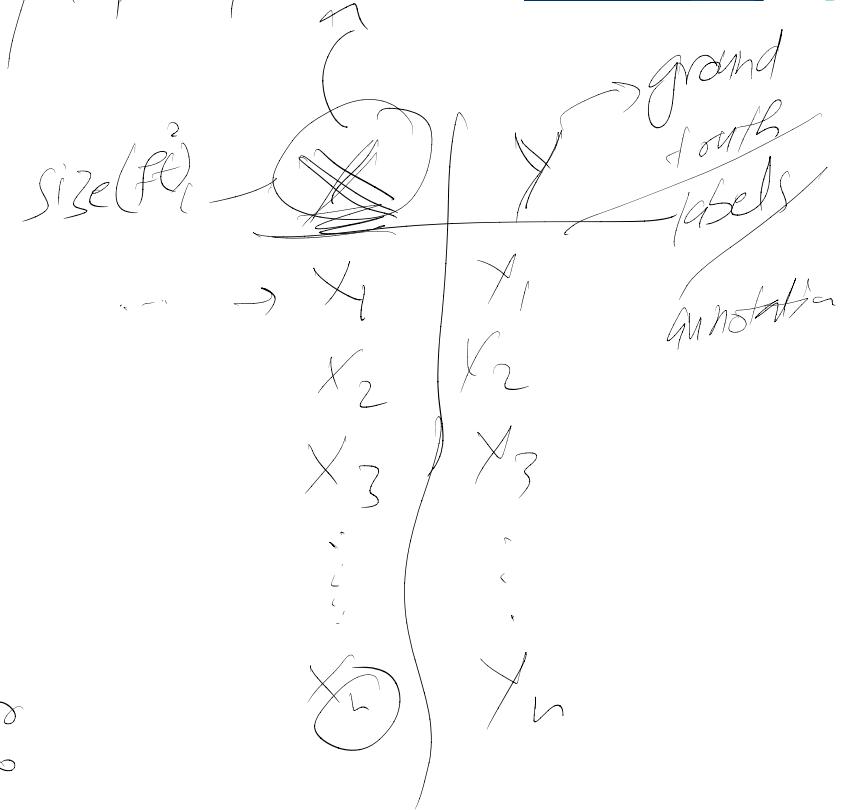
Linear Regression: Motivation

- Linear Regression is “still” one of the more widely used ML/DL Algorithms
- Easy to understand and implement
- Efficient to Solve
- We will use Linear Regression to Understand the concepts of:
 - Data
 - Models
 - Loss
 - Optimization

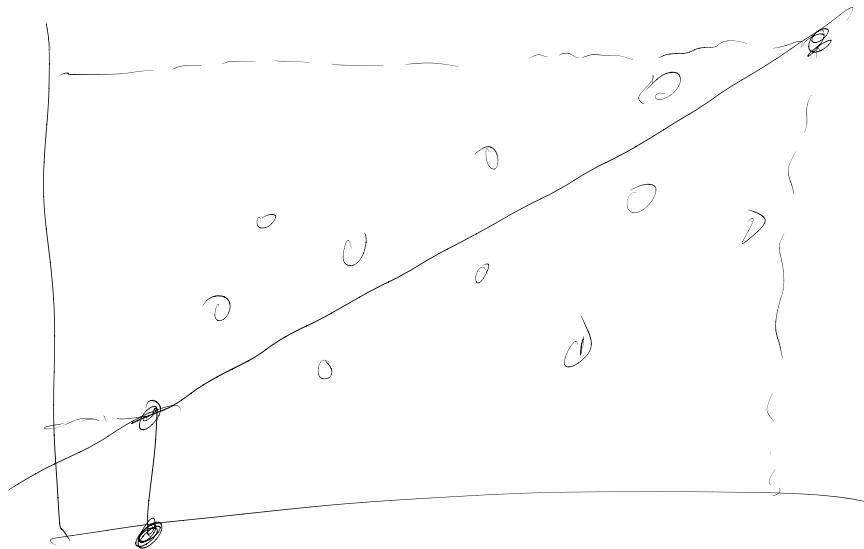
Linear Regression: Data



attribute / features / input / data



Linear Regression: Model



average / - - -

$y = mx + b$

$f(m)$

model

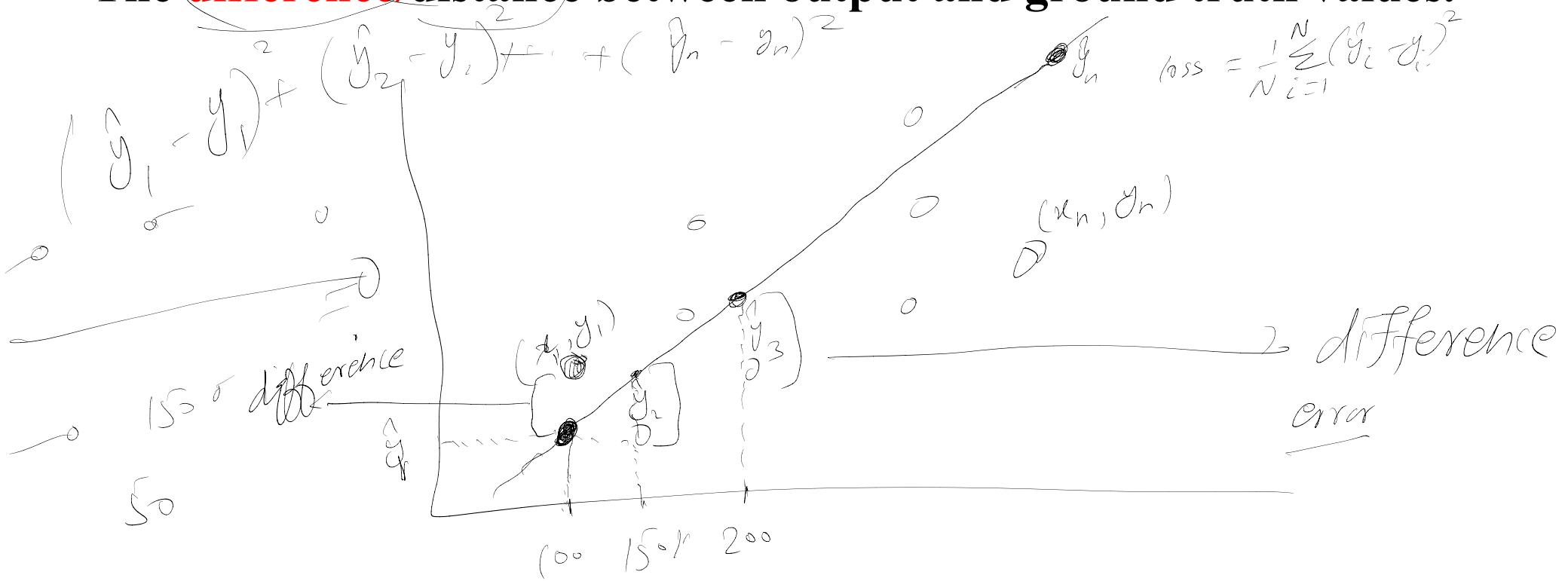
A hand-drawn diagram of the linear regression equation $y = mx + b$. The letter y is written above the equation. The term mx is enclosed in a circle, with the label $f(m)$ written below it. The term b is also enclosed in a circle. The word "model" is written below the equation. Above the equation, the words "average / - - -" are written, with a dashed line extending from the text towards the y -intercept of the line in the scatter plot.

Linear Regression: Model

- The linear regression model is function of ... ? $\rightarrow m \in \mathbb{R}$
- What are constant & variables

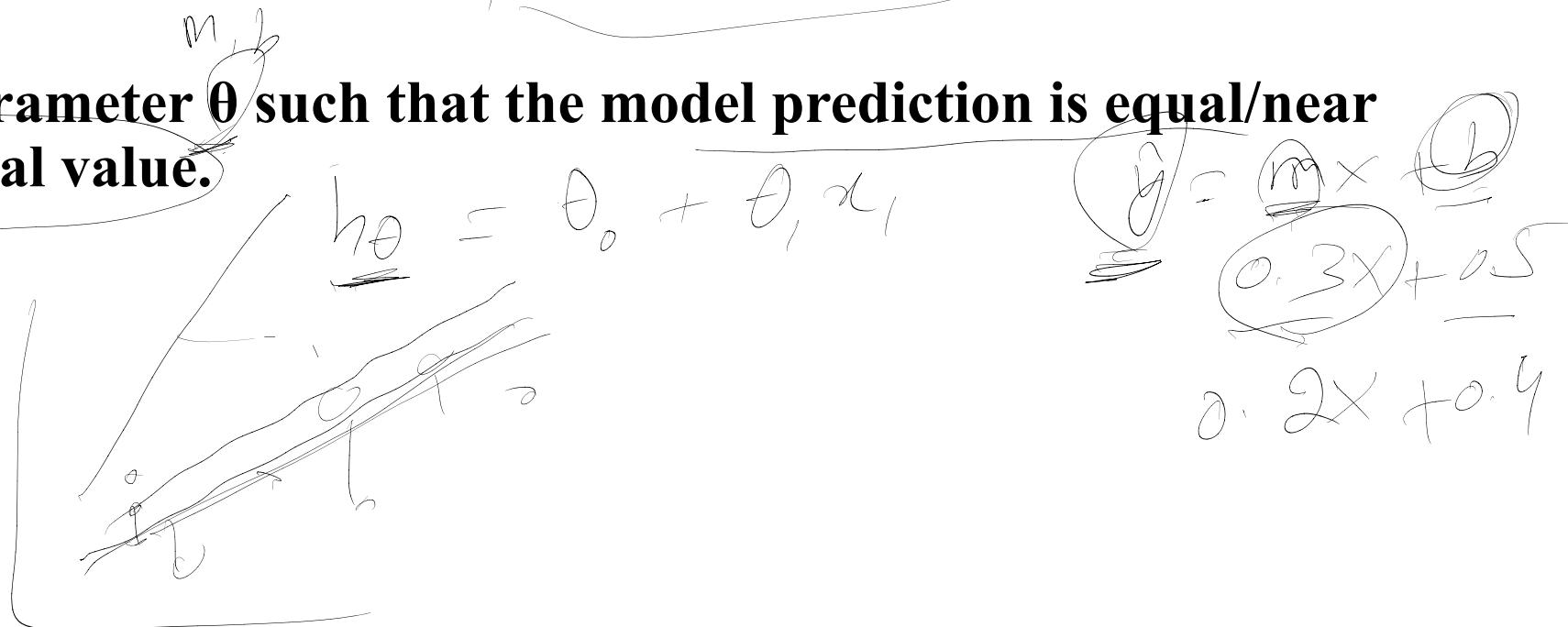
Linear Regression: Loss / Objective

- The difference/distance between output and ground truth values.



Linear Regression: Optimization

- Choose parameter θ such that the model prediction is equal/near to the actual value.



Simple Linear Regression

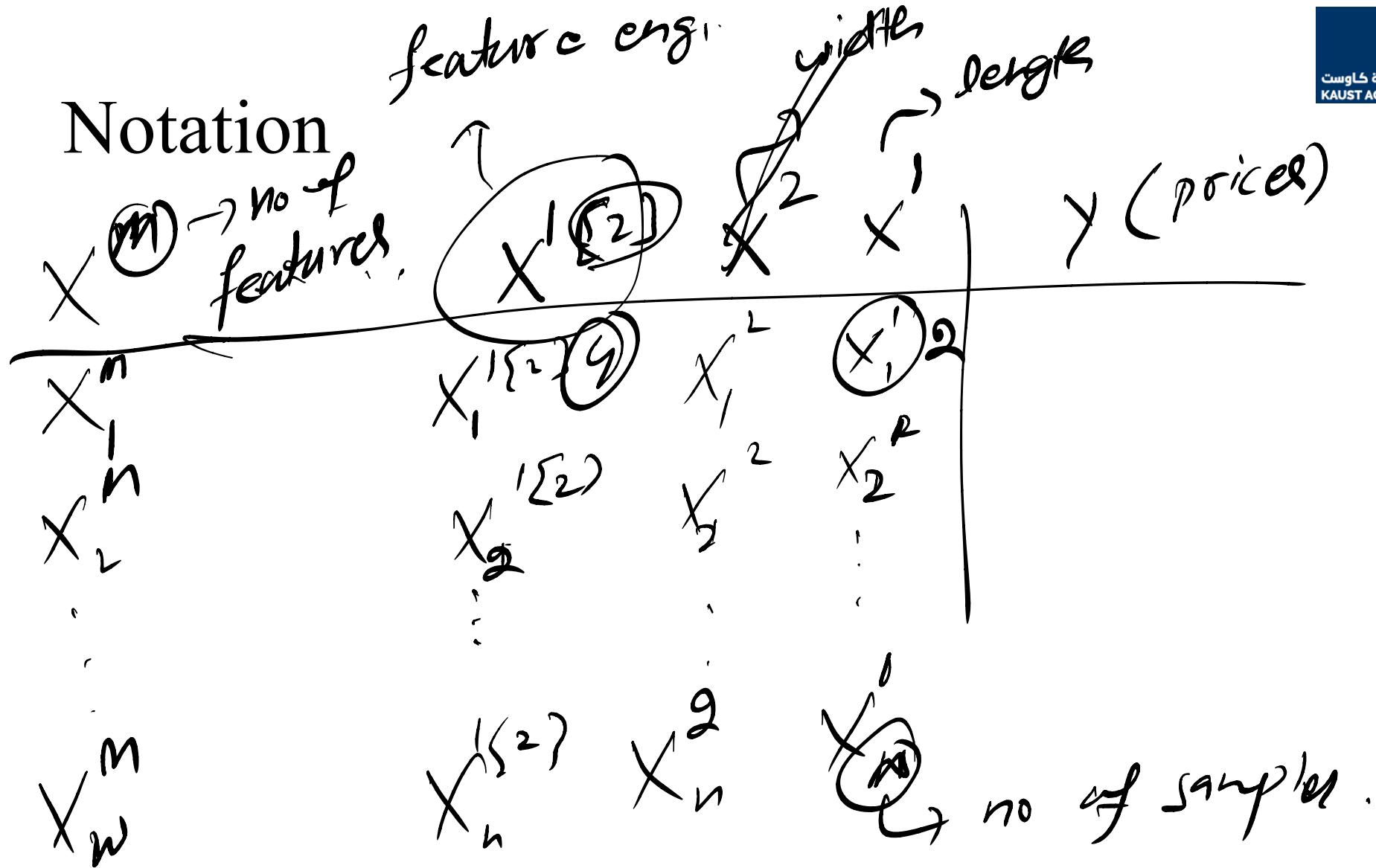
- **Hypothesis:** $\hat{y}_i = mx_i + b$
- **Input:** data (x_i, y_i) , $i \in \{1, 2, \dots, l\}$
 - (e.g., house size x and price y)

Notation

i → no of samples
 j → no of feature

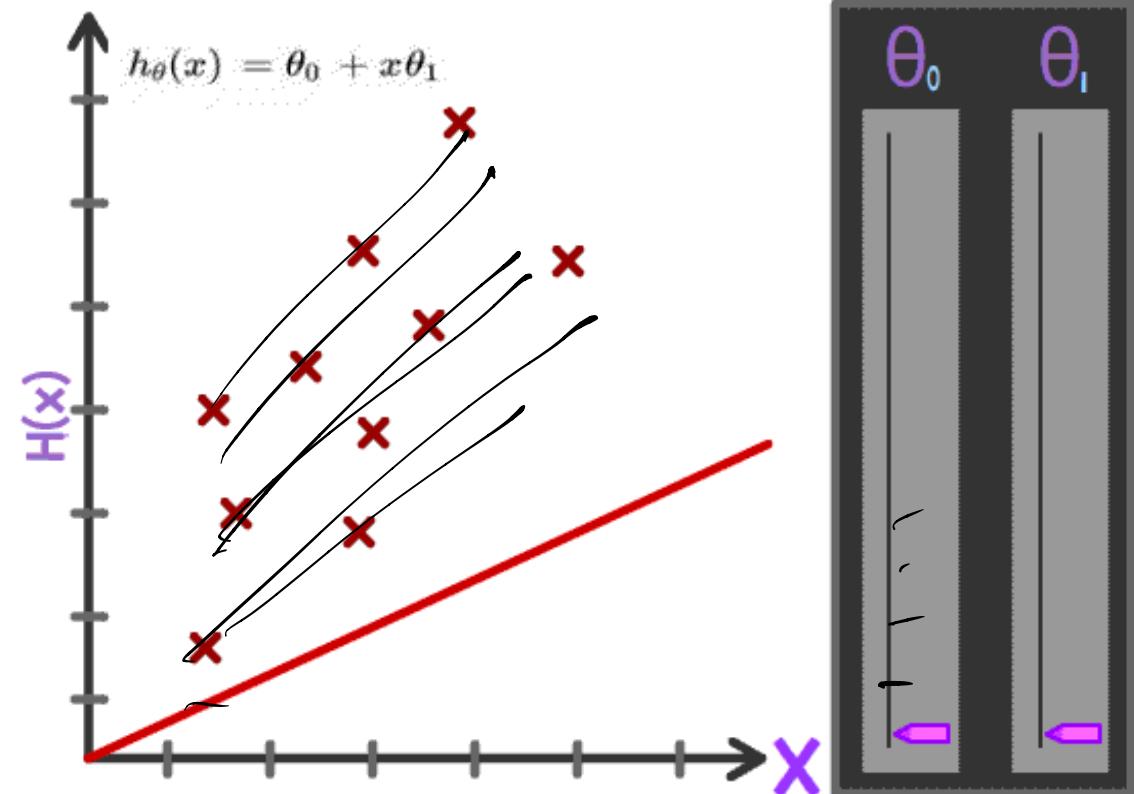


Notation



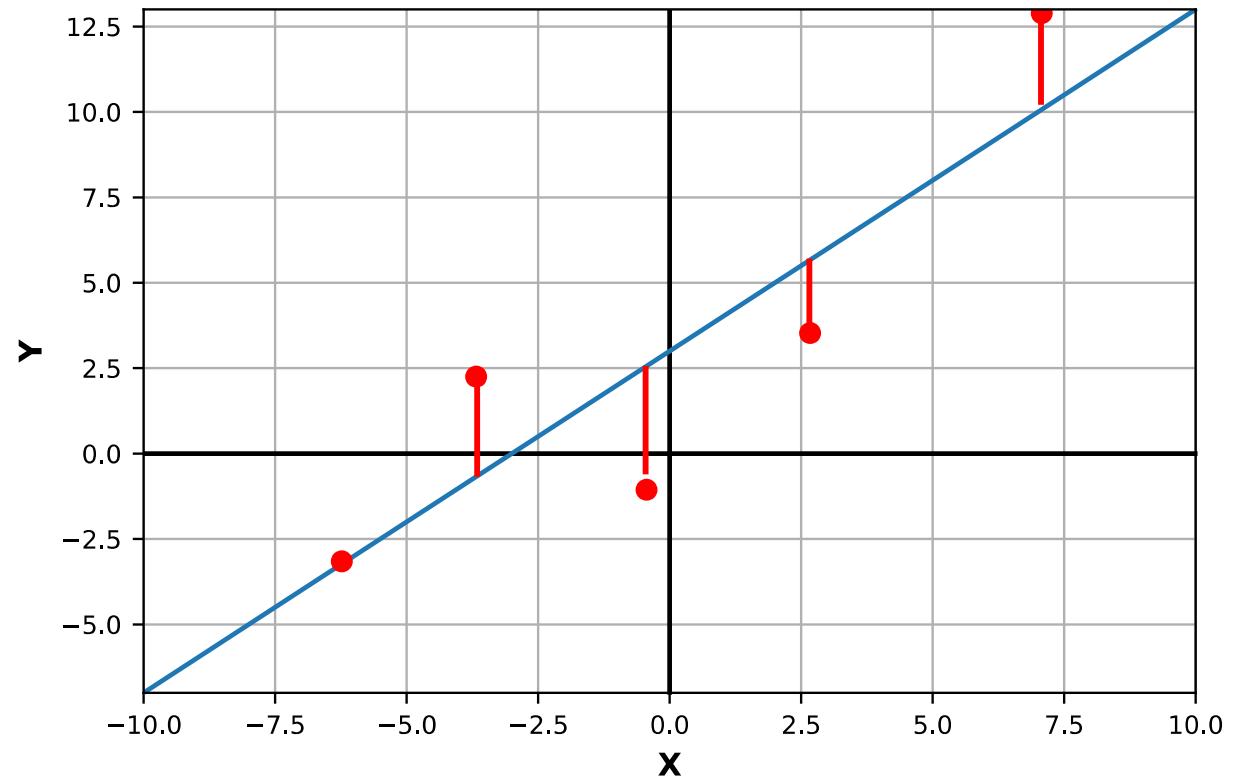
Solution Strategy for Solving the Problem

- There are countless possible lines.
- We want a line which is in some sense the “average line” that represents the data.
- Any ideas as to how we can do it?



Optimization

- To find the "best line," we should minimize the **distances** between our line's predictions and all the data points.
- How to define that mathematically?

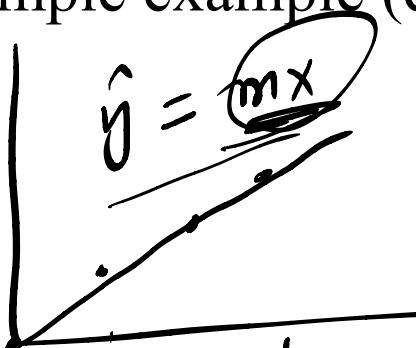


Loss Function

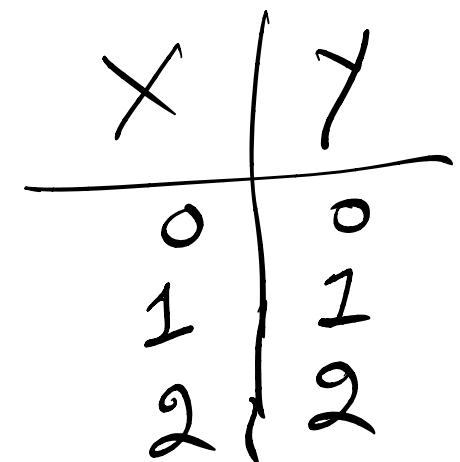
- For one sample, this can be represented as:
$$y - \hat{y}$$
 (Error)
- But this could result in negative values. To remove the negative sign:
$$(y - \hat{y})^2$$
 (Squared Error)
- Mean Squared Error:
$$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$
 (Mean Squared Error)

Example Data and its Solution

- Let's take a simple example (concrete data) and solve it for m .



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



Example Data and its Solution

- Let's take a simple example (concrete data) and solve it for m .

$$\begin{aligned}
 J(m) &= \frac{1}{N} \sum_{i=1}^N (m \cancel{x_i} - y)^2 \\
 &= \frac{1}{3} \left[(m \cdot 0 - 0)^2 + (m \cdot 1 - 1)^2 + (m \cdot 2 - 2)^2 \right] \\
 &= \frac{1}{3} \left[(m - 1)^2 + (2m - 2)^2 \right].
 \end{aligned}$$

Example Data and its Solution

- Let's take a simple example (concrete data) and solve it for m .

$$\mathcal{J}(0) = \frac{1}{3} \left[(\underline{x} - \bar{x})^2 + (\underline{y} - \bar{y})^2 + (\underline{z} - \bar{z})^2 \right]$$

$$\mathcal{J}(0) = \frac{1}{3} \left[1 + 4 \right] = \frac{5}{3}$$

Example Data and its Solution

- Let's take a simple example (concrete data) and solve it for m .

$$m_2 = \frac{1}{3} [0 + 0]$$

$$\boxed{J(1) = 0}$$

$$m_j \stackrel{i=1}{=} m_j - \frac{\partial^2}{\partial m^2}$$

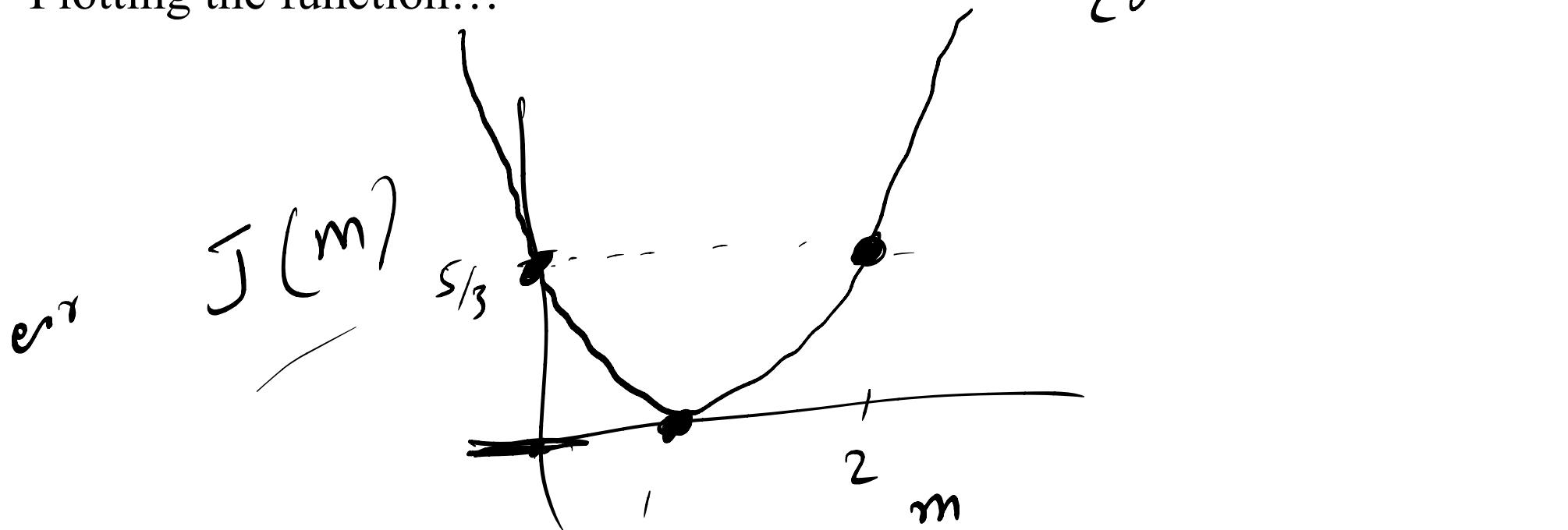
$$m \alpha - j$$

$$m = \frac{2}{3} J(2) = \frac{1}{3} [0 + 4]$$

$$\boxed{J(2) = \frac{4}{3}}$$

Example Data and its Solution

- Plotting the function...



Example Data and its Solution

- Plotting the function...

$$J(m) = \frac{dJ}{dm} = 0 \quad \frac{dJ}{dm} = \frac{d}{dm} (\text{loss fn})$$

- Convex function i.e., it has only one critical point, which is the global minimal.
- Set derivative equal to zero and the critical point acquired is the minimal

5, 12

Example Data and its Solution

- Now solve it for the optimal value.

$$\begin{aligned}
 \frac{dJ}{dm} &= \frac{d}{dm} \left(\frac{1}{3} ((m-1)^2 + (2m-2)^2) \right) \\
 &= \frac{1}{3} \left[\frac{d}{dm} (m-1)^2 + \frac{d}{dm} (2m-2)^2 \right]. \\
 &= \frac{1}{3} \left[2(m-1) + 2(2m-2) \cdot 2 \right]. \\
 &= \frac{1}{3} [\underline{2m-2} + \underline{\underline{8m-8}}]
 \end{aligned}$$

Example Data and its Solution

- Now solve it for the optimal value.

$$\frac{dJ}{dm} = \frac{1}{3} \left\{ (2m - 10) \right\}$$

$$\frac{dJ}{dm} = 0$$

$$\cancel{\frac{1}{3}} m - \cancel{\frac{10}{3}} = 0$$

m = 1

Example Data and its Solution

- Now solve it for the optimal value.

① Select model based on data

- ↳ find the loss \Rightarrow suitable me
- ↳ for the optimization

② derivative = $\partial \rightarrow$ clos

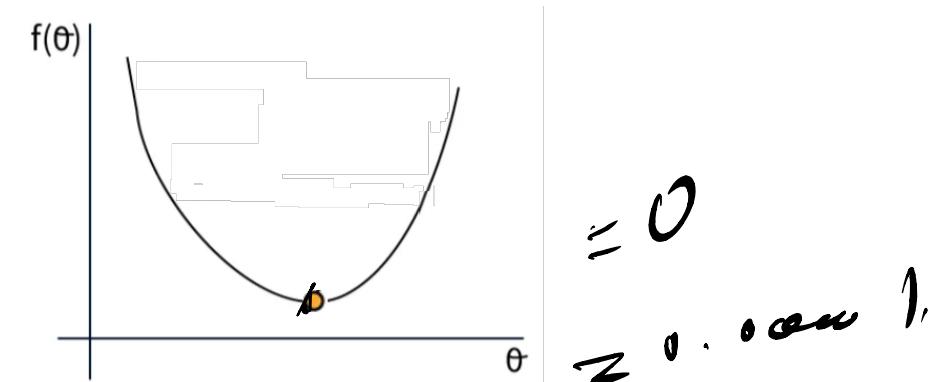


How to find minima of a function (Review):

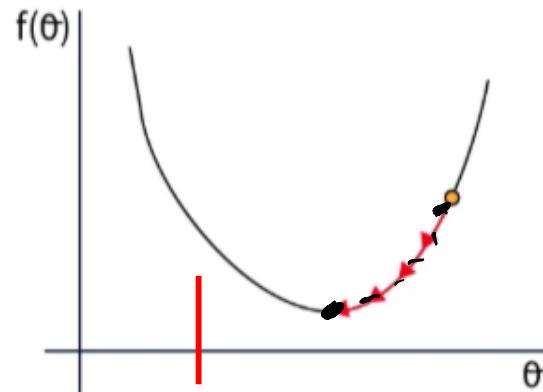
- There are two approaches to fin
- **Exact (Closed-form)**: Directly calculate solving for $f'(x) = 0$.
Important Note: can be used only

How to find minima of a function (Review):

Closed-form:



Iterative:



- Example: $y = x^2$ (Solution: $x = 0$)
 - Closed-form Final Result: $x = 0$

Hypothesis Function with 2 Variables

- Let's setup regression for linear fur
- The hypothesis function is:

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

$$\hat{y}_i = mx_i + b$$

Gradient of the loss function

- We get the following expressions for the gradient of the cost function

$$\frac{\partial J}{\partial m} = \frac{1}{N} \sum_{i=1}^N -2(y_i - \hat{y}_i)x_i$$

$\frac{\partial}{\partial} (y_i - \hat{y}_i)^2$

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

$\frac{\partial}{\partial} (mx_i + b)$

Gradient of the loss function

- Simplifying the above expressions, we get:

$$\frac{\partial J}{\partial m} = \cancel{\frac{2}{N}} \sum_{i=1}^N y_i x_i + \cancel{\frac{2m}{N}} \sum_{i=1}^N x_i^2 + \cancel{\frac{2b}{N}} \sum_{i=1}^N x_i$$

$$\frac{\partial J}{\partial b} = \cancel{-\frac{2}{N}} \sum_{i=1}^N y_i + \cancel{\frac{2m}{N}} \sum_{i=1}^N x_i + \cancel{\frac{2b}{N}} \sum_{i=1}^N 1$$

Gradient of the loss function

- Setting the Gradient equal to 0, and solving for m and b, we get

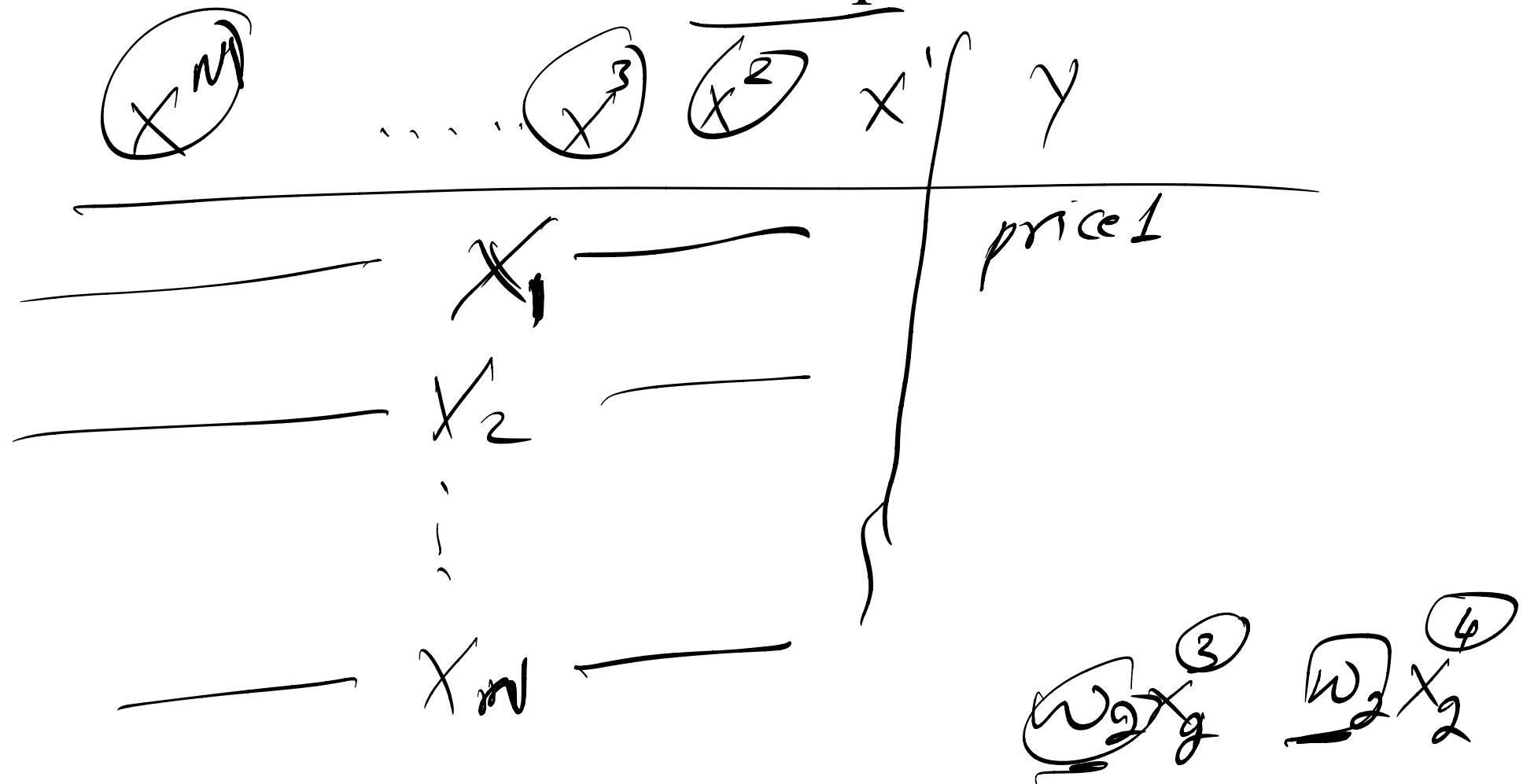
$$\left[\begin{matrix} \frac{\sum_i x_i^2}{N} & \frac{\sum_i x_i}{N} \\ 1 & 1 \end{matrix} \right] \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \frac{\sum_i x_i y_i}{N} \\ \frac{\sum_i y_i}{N} \end{bmatrix}$$

~~A~~ . ~~B~~

$\theta = A^{-1} B$

$\theta = A^{-1} B$

Solution for Data with Multiple Features



Solution for Data with Multiple Features

$$\hat{y}_1 = \underline{1} \cdot \underline{w_0} + \underline{w_1} \underline{x_1^1} + \dots + \underline{w_m} \underline{x_1^m}$$

$$\hat{y}_2 = \underline{1} \cdot \underline{w_0} + \underline{w_1} \underline{x_2^1} + \dots + \underline{w_m} \underline{x_2^m}$$

$$\hat{y}_3 =$$

$$\vdots$$

$$\hat{y}_n = \underline{1} \cdot \underline{w_0} + \underline{w_1} \underline{x_n^1} + \underline{w_2} \underline{x_n^2} + \underline{w_3} \underline{x_n^3} + \dots + \underline{w_m} \underline{x_n^m}$$

Solution for Data with Multiple Features



Solution for Data with Multiple Features

$$\begin{aligned}
 \bar{Y} &= X \\
 \text{Loss} &= \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\
 &= \sum_{i=1}^n (\epsilon_i)^2
 \end{aligned}$$

Solution for Data with Multiple Features

$$\begin{aligned}
 \text{Loss} &= \frac{1}{n} \sum_{i=1}^n \epsilon_i^2 \\
 &= \frac{1}{n} \underbrace{\epsilon \cdot \epsilon^T}_{\epsilon^T \cdot \epsilon} \\
 \text{Loss} &= \underbrace{\epsilon^T \cdot \epsilon}_{\sum (\epsilon_1^2 + \epsilon_2^2 + \dots + \epsilon_n^2)} \\
 &\quad \left[\epsilon_1^T \cdot \epsilon_1, \epsilon_2^T \cdot \epsilon_2, \dots, \epsilon_n^T \cdot \epsilon_n \right]
 \end{aligned}$$

Solution for Data with Multiple Features

$$\begin{aligned}
 \text{loss} &= \underline{\epsilon}^T \cdot \underline{\epsilon} \\
 &= (\hat{y} - y)^T \cdot (\hat{y} - y) \\
 \text{loss} &= (\underbrace{m_x - y}_m)^T (\underbrace{m_x - y}_m) = 0 \\
 \frac{\partial}{\partial m} (\underbrace{m_x - y}_m)^T (\underbrace{m_x - y}_m) &= 0
 \end{aligned}$$

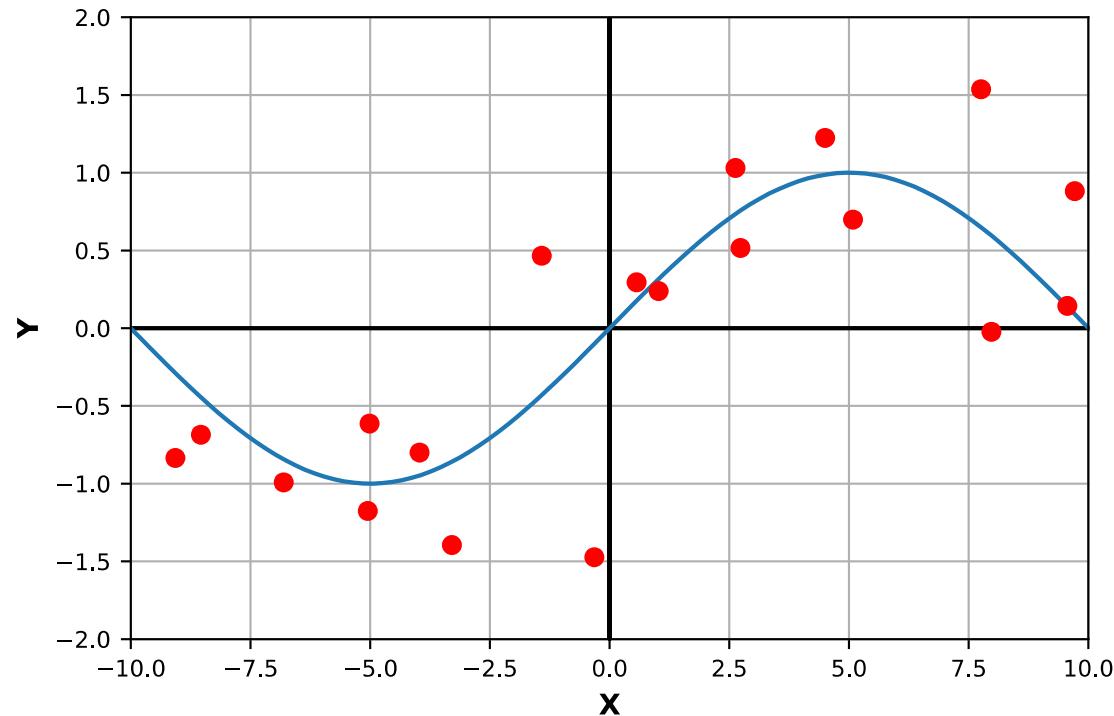
Solution for Data with Multiple Features

$$\theta = \frac{(X^T X)^{-1} (X^T y)}{A^{-1} B}$$

Normal Equation

Fitting Non-linear Data

- What if y is a non-linear function of x , will this approach still work?



Transforming the Feature Space (Feature Engineering)

- We can transform features x_i

$$x_i = (x_i^1, x_i^2, x_i^3, \dots, x_i^m)$$

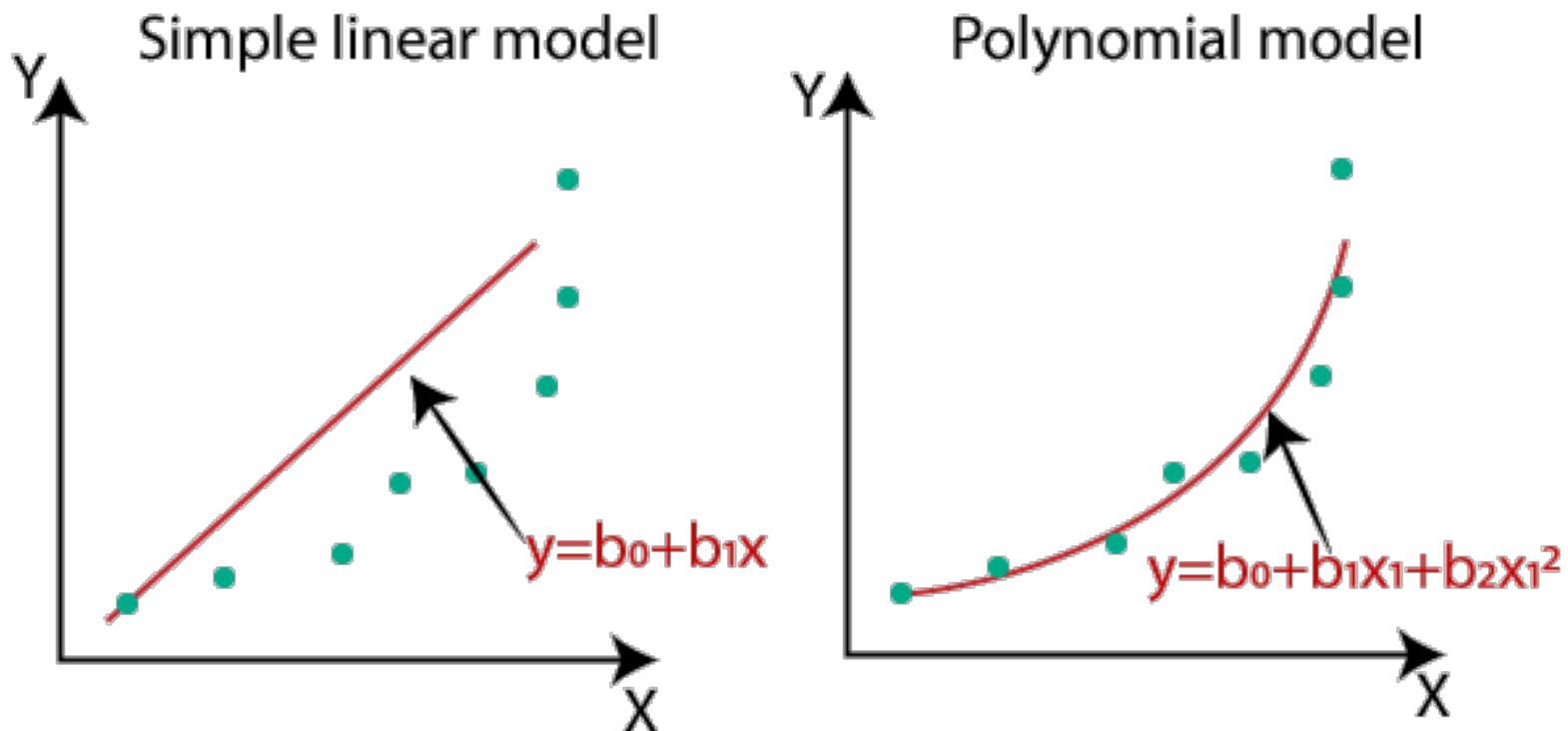
- We will apply some non-linear transformation ϕ :

$$\phi : \mathbb{R}^m \rightarrow \mathbb{R}^M$$

- For example, Polynomial transformation:

$$\phi(x_i) = \{1, x_i^1, x_i^{1,[2]}, \dots, x_i^{1,[k]}, x_i^2, x_i^{2,[2]}, \dots, x_i^{2,[k]}, \dots, x_i^m, x_i^{m,[2]}, \dots, x_i^{m,[k]}\}$$

Transforming the Feature Space (Feature Engineering)



Transforming the Feature Space (Feature Engineering)

- Example: assume you have:

x_1 :

x_2 :

You can add x_3 : $Area = x_1 * x_2$ to

Gradient of the loss function

- Let's get back to the gradients...

Issues with the Approach

- Assume we have 100 variables instead of 2.
- Calculating gradients like this can quickly become tedious
- **Notice:** Each term on either side of the expression can be written a dot product of two vectors (maybe we can calculate it more efficiently)?
- Let's explore if we can do something better through **vectorization** (Writing equations as matrices).

Vectorization

- To truly appreciate the power of problem a little more complex. 1

$$\hat{y}_i = w_0 + w_1 x_i^1 +$$

Vectorization

- Now let's collect the above equations

$$y_1 = \hat{y}_1 + \epsilon_1$$

$$y_2 = \hat{y}_2 + \epsilon_2$$

.

.

.

$$y_N = \hat{y}_N + \epsilon_N$$

Vectorization

- Replacing the values of \hat{y} , we get

$$y_1 = w_0 + w_1 x_1^1 + w_2 x_1^2 + \dots + w_M x_1^M + \epsilon_1$$
$$y_2 = w_0 + w_1 x_2^1 + w_2 x_2^2 + \dots + w_M x_2^M + \epsilon_2$$

.

.

.

$$y_N = w_0 + w_1 x_N^1 + w_2 x_N^2 + \dots + w_M x_N^M + \epsilon_N$$

Vectorization

- Collecting the equations in matrix form:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & x_1^1 & x_1^2 & \dots & x_1^M \\ 1 & x_2^1 & x_2^2 & \dots & x_2^M \\ 1 & x_3^1 & x_3^2 & \dots & x_3^M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N^1 & x_N^2 & \dots & x_N^M \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ \vdots \\ w_M \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \vdots \\ \vdots \\ \epsilon_N \end{bmatrix}$$

Vectroization

- Notice the rows of the matrix on the right are data samples:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} \dots & \mathbf{x}_1 & \dots \\ \dots & \mathbf{x}_2 & \dots \\ \dots & \mathbf{x}_3 & \dots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \dots & \mathbf{x}_N & \dots \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ \vdots \\ w_M \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \vdots \\ \epsilon_N \end{bmatrix}$$

Vectorization

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$$

- Let's formalize some notaitons:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_N \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} \dots & \mathbf{x}_1 & \dots \\ \dots & \mathbf{x}_2 & \dots \\ \dots & \mathbf{x}_3 & \dots \\ & \ddots & \\ & \ddots & \\ \dots & \mathbf{x}_N & \dots \end{bmatrix} \quad \boldsymbol{\theta} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ \vdots \\ w_M \end{bmatrix} \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \vdots \\ \epsilon_N \end{bmatrix}$$

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\epsilon}$$

Cost function for the Vectorized form

- Notice that we are using the MSE cost function:
- Using the definition of epsilon we can write the above as:

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

- Using the definition of dot product the above can be written as:

$$J = \frac{1}{N} \sum_{i=1}^N (\epsilon_i)^2 = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon}$$

Optimization

- The optimization problem is now:

$$\min_{\theta} \epsilon^T \epsilon$$

$$\min_{\theta} \epsilon^T \epsilon = \min_{\theta} (\mathbf{y} - (\mathbf{X}\theta))^T (\mathbf{y} - (\mathbf{X}\theta))$$

- We will use chain rule to calculate the gradient of the cost function:

$$\frac{\partial}{\partial \theta} J = \frac{dJ}{d\epsilon} \nabla_{\theta} \epsilon$$

Linear Least Squares

- We get:

$$\frac{\partial}{\partial \theta} J = \mathbf{X}^T 2(\mathbf{y} - \mathbf{X}\theta)$$

- Setting it equal to zero we can solve for θ :

$$\theta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Closed-form solution for Linear Regression

$$x^2 + y^2 = r$$

$$2x^2 + 2y^2 = 2r$$

Regularization

$$\omega = \begin{bmatrix} 1 \\ 5 \\ 0 \\ -1 \end{bmatrix}$$

- Non-linear function with multiple independent variables.
- What do we do for single variable?

Figure showing a 3D plot of a non-linear function \hat{y} versus x^1 , x^2 , and x^3 . The axes are labeled x^1 , x^2 , x^3 and y .

$\hat{y} = w_0 + w_1 x^1 + w_2 x^2 + w_3 x^3$

$\hat{y} = 1 + 5x^1 + 0 + x^3$

one solution



$$nor = \sqrt{w_1^2 + w_2^2 + w_3^2}$$

Regularization

- Non-linear function with **multiple independent variables**.
- What do we do for single variable?

$$\hat{y} = 1 + 4x^1 + 0 + (x^3) + x^1 + x^3$$

$$\hat{y} = 1 + 4x^1 + 2x^3$$

$$\frac{\partial J}{\partial \theta} = \begin{bmatrix} 9 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 4 \\ 0 \\ 2 \end{bmatrix}$$

$$nor = \sqrt{16 + 0} = 4$$

Regularization

- Non-linear function with **multiple independent variables**.
- What do we do for single variable?

$$\theta = (X^T X + \lambda I) (X^T y)$$

$(\bar{X} - y)^T (\bar{X} - y) + \lambda \theta^T \theta$

Regularization

- Non-linear function with **multiple independent variables**.
- What do we do for single variable?

$$\text{Let } (x^\theta - j)^\top (x^\top \theta \cdot y) + \sum_{i=1}^m |\theta_i|$$