Linear & Logistic Regression

FTI UII
13 Januari 2020

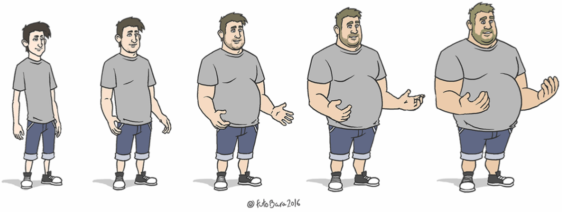
Dr. Ing. Ridho Rahmadi, S.Kom., M.Sc.

UNIVERSITAS
ISLAM
INDONESIA

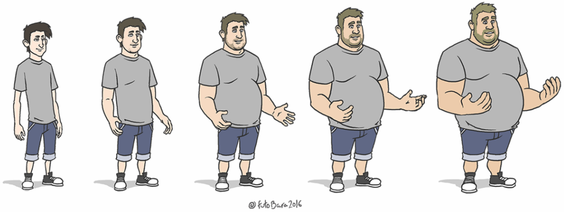
- Head of Center of Data Science, UII
- Head of Research Laboratory, Informatics, UII
- Education
  - Universitas Islam Indonesia (S1)
  - Czech Technical University in Prague (S2)
  - Johannes Kepler University, Austria (S2)
  - Radboud University Nijmegen, the Netherlands (S3)
  - Carnegie Mellon University, USA (Visiting scholar)
- Research interest
  - Machine learning, deep learning, causal modeling, stability selection, multi-objective evolutionary algorithms

- ridho.rahmadi@uii.ac.id
- 081129513045

How do we *see* the world?



How do we *see* the world?



The more cookies I eat, the more weight I gain

How do we *see* the world?

Mood

How do we *see* the world?

Mood

I am not necessarily always happier as time passes

How do we *see* the world?

- If I eat 1 cookie per day, I gain 2 Kg weight
- If I eat 2 cookies per day, I gain 4 Kg weight

How do we *see* the world?

- If I eat 1 cookie per day, I gain 2 Kg weight
- If I eat 2 cookies per day, I gain 4 Kg weight

The more cookies I eat, the more weight I gain.

How do we *see* the world?

- If I eat 1 cookie per day, I gain 2 Kg weight
- If I eat 2 cookies per day, I gain 4 Kg weight

The more cookies I eat, the more weight I gain.

We call such an example as a **linear model**.

How do we *see* the world?

- If I eat 1 cookie per day, I gain 2 Kg weight
- If I eat 2 cookies per day, I gain 4 Kg weight

The more cookies I eat, the more weight I gain.

We call such an example as a **linear model**. For example,

- If A goes up, so does B, OR
- If A goes up, B goes down

How do we *see* the world?

- If I eat 1 cookie per day, I gain 2 Kg weight
- If I eat 2 cookies per day, I gain 4 Kg weight

The more cookies I eat, the more weight I gain.

We call such an example as a **linear model**. For example,

- If A goes up, so does B, OR
- If A goes up, B goes down

Given two variables, a linear relationship among them indicates consistent directions of changes.

How do we *see* the world?

I just woke up and

- I am happy (minute 1)
- I am angry as no one WA me (minute 10)
- I am happy as 1 WA comes (minute 11)
- I am nervous as I'll have an exam (minute 30)

How do we *see* the world?

I just woke up and

- I am happy (minute 1)
- I am angry as no one WA me (minute 10)
- I am happy as 1 WA comes (minute 11)
- I am nervous as I'll have an exam (minute 30)

I am not necessarily always happier as time passes.

How do we *see* the world?

I just woke up and

- I am happy (minute 1)
- I am angry as no one WA me (minute 10)
- I am happy as 1 WA comes (minute 11)
- I am nervous as I'll have an exam (minute 30)

I am not necessarily always happier as time passes.

We call such an example as **nonlinear model**.

How do we *see* the world?

I just woke up and

- I am happy (minute 1)
- I am angry as no one WA me (minute 10)
- I am happy as 1 WA comes (minute 11)
- I am nervous as I'll have an exam (minute 30)

I am not necessarily always happier as time passes.

We call such an example as **nonlinear model**.

For example, if A goes up, B alternates up and down.

How do we *see* the world?

- If I eat 1 cookie per day, I gain 2 Kg of weight
- If I eat 2 cookies per day, I gain 4 Kg of weight

What if I eat 3 cookies? How many Kg of weight will I gain?

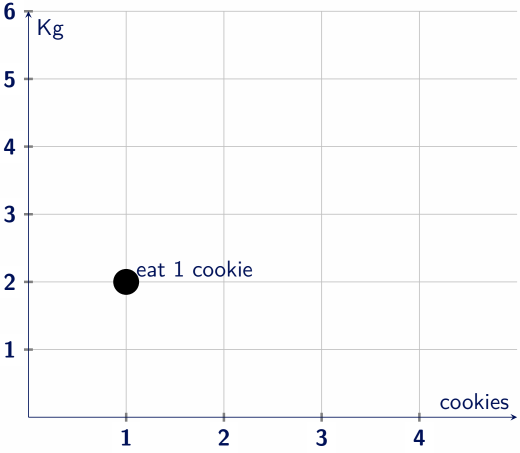
How do we *see* the world?

I just woke up and

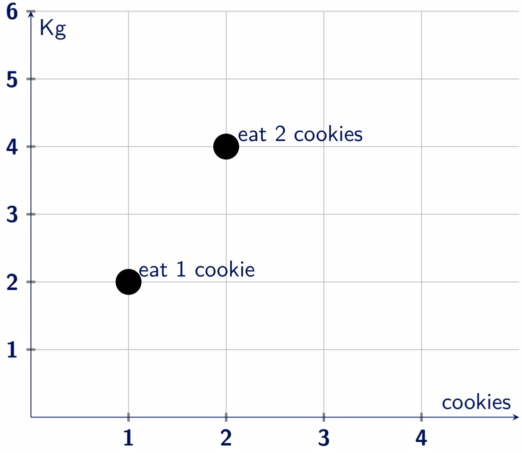
- I am happy (minute 1)
- I am angry as no one WA me (minute 10)
- I am happy as 1 WA comes (minute 11)
- I am nervous as I'll have an exam (minute 30)

Will I be happy in minute 110? Or nervous?

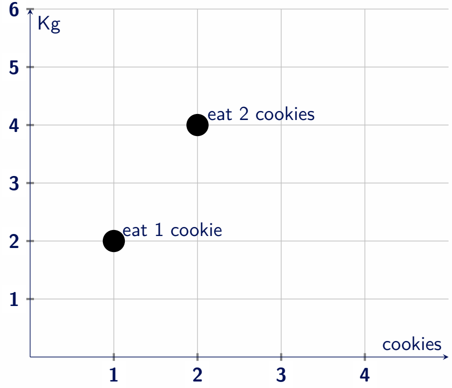
Say it in Mathematics



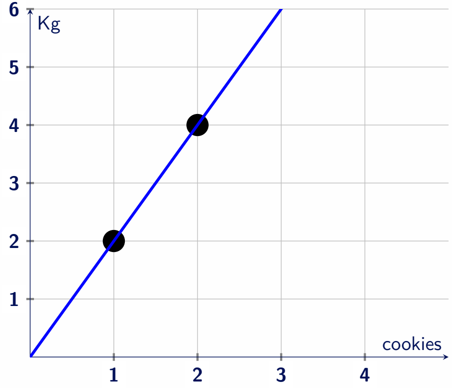
Say it in Mathematics



Say it in Mathematics

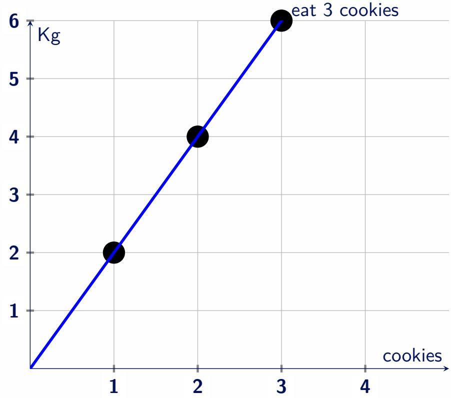


What if I eat 3 cookies?



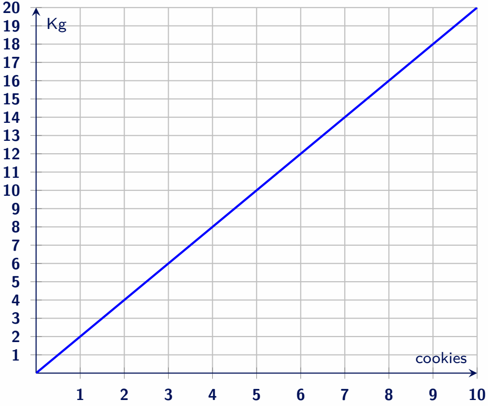
We can draw a **line** that passes through the two points.

Say it in Mathematics



Using the line, we can predict "what if I eat 3 cookies?"

Say it in Mathematics



With the line, you can predict the weight gain for any (positive) number of cookies eaten.

Say it in Mathematics

In terms of mathematical function, a line can be represented by

$$y = \theta_0 + \theta_1 x,$$

where θ_0 is the **intercept** and θ_1 is the **slope**.

Say it in Mathematics

In terms of mathematical function, a line can be represented by

$$y = \theta_0 + \theta_1 x,$$

where θ_0 is the **intercept** and θ_1 is the **slope**.

The intercept indicates the y-coordinate through which the line passes. It is the value of y when $x = 0$.

Say it in Mathematics

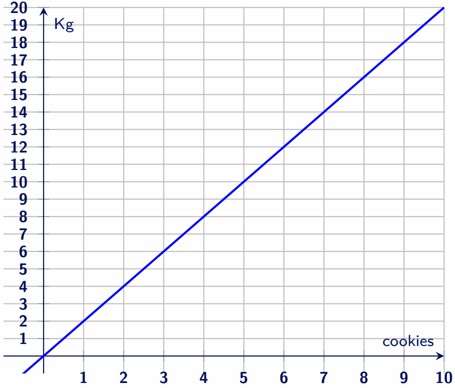
In terms of mathematical function, a line can be represented by

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

where $\boldsymbol{\theta_0}$ is the **intercept** and $\boldsymbol{\theta_1}$ is the **slope**.

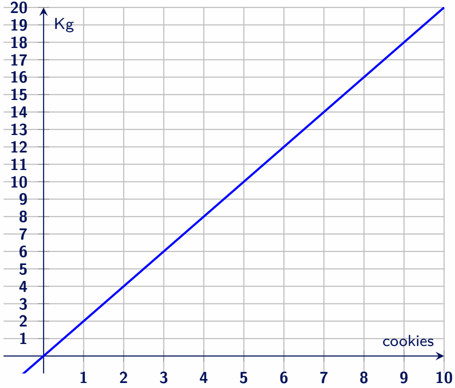
The intercept indicates the $\boldsymbol{y}$-coordinate through which the line passes. It is the value of $\boldsymbol{y}$ when $\boldsymbol{x = 0}$.

The slope represents how steep the line is.



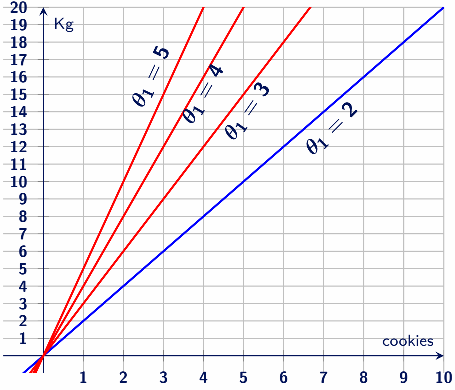
From the previous example, we have $\boldsymbol{y = 0 + 2x}$.

Say it in Mathematics

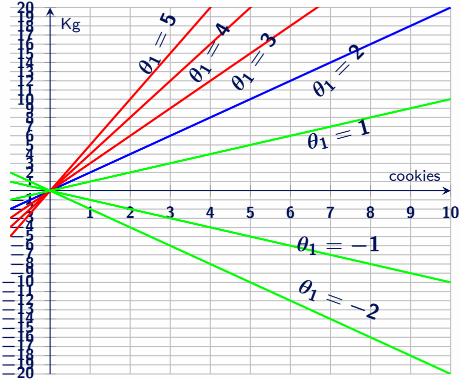


From the previous example, we have $\boldsymbol{y} = \mathbf{0} + \mathbf{2}\boldsymbol{x}$. Now let's play a bit with the slope and intercept.

Say it in Mathematics

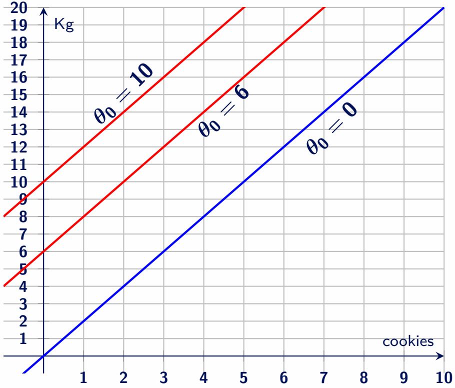


From $\boldsymbol{y = 0 + 2x}$, we have $\boldsymbol{\theta_1 = 2}$. See what happens if we increase the slope $\boldsymbol{\theta_1}$.



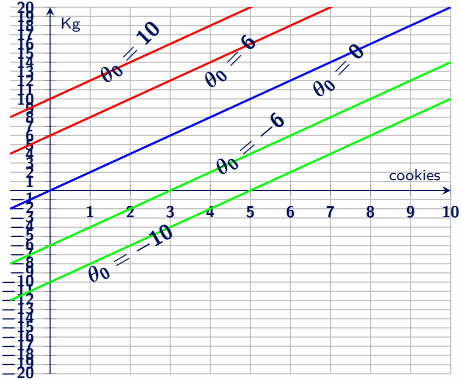
From $y = 0 + 2x$, we have $\theta_1 = 2$. See what happens if we decrease the slope θ_1.

Say it in Mathematics



From $\boldsymbol{y = 0 + 2x}$, we have $\boldsymbol{\theta_0 = 0}$. See what happens if we increase the intercept $\boldsymbol{\theta_0}$.

Say it in Mathematics



From $\boldsymbol{y} = \mathbf{0} + \mathbf{2}\boldsymbol{x}$, we have $\boldsymbol{\theta_0} = \mathbf{0}$. See what happens if we decrease the intercept $\boldsymbol{\theta_0}$.

Linear regression

The previous example illustratively describes the idea behind the **linear regression**.

That is, based on the data we have, we want to predict the weight gained, given the number of cookies we eat, by fitting a line.

Next, we will describe the linear regression in more detail.

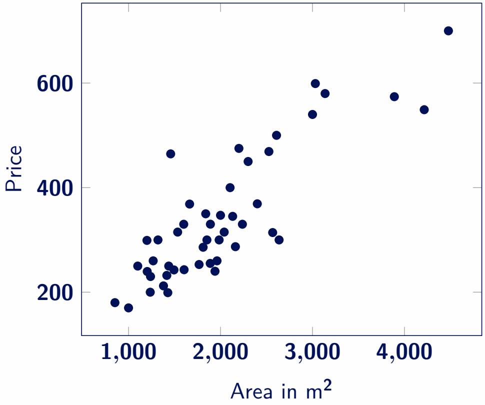
Real-world cases

In real-world cases, the data sets are often of the form $(\boldsymbol{x}^{(\boldsymbol{i})}, \boldsymbol{y}^{(\boldsymbol{i})})$; $\boldsymbol{i} = \mathbf{1}, \ldots, \boldsymbol{m}$. For example,

House area ($\boldsymbol{x}$)	Price ($\boldsymbol{y}$)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

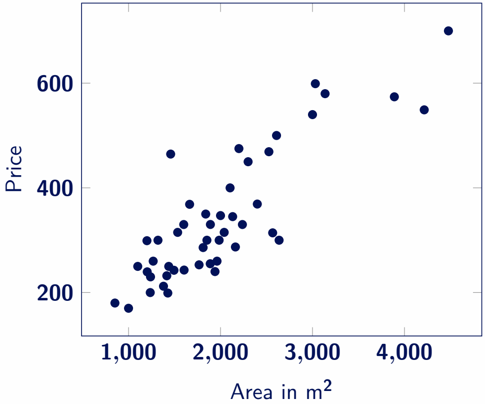
Based on the data above, a typical question is, e.g., what is the price of a house if the area is 558 M^2?

Plot the data



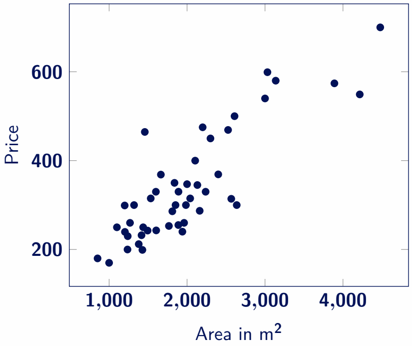
Does the data distribution form a linear fashion?

Plot the data



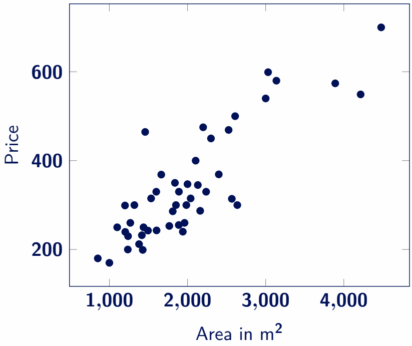
A typical question is, e.g., what is the price of a house if the area is 558 M^2?

Plot the data



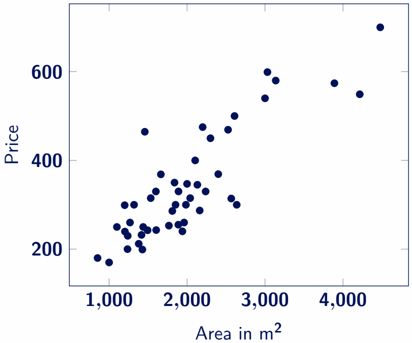
- Note that we are interested in to predict *unseen* $\hat{\boldsymbol{y}}$ based on $\hat{\boldsymbol{x}}$ from other population

Plot the data



- Note that we are interested in to predict *unseen* $\hat{\boldsymbol{y}}$ based on $\hat{\boldsymbol{x}}$ from other population
- Solution: we can find a line (model) that constitutes the data we have, and use the line to predict $\hat{\boldsymbol{y}}$ based on $\hat{\boldsymbol{x}}$

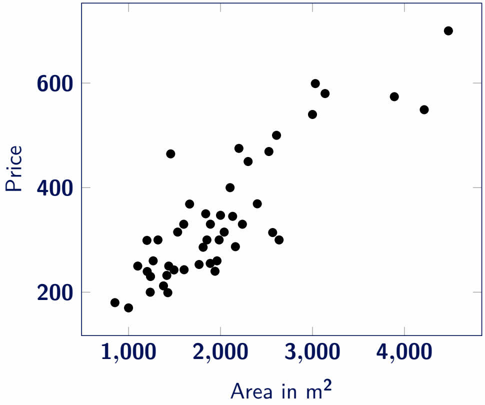
Plot the data



- Note that we are interested in to predict *unseen* $\hat{\boldsymbol{y}}$ based on $\hat{\boldsymbol{x}}$ from other population
- Solution: we can find a line (model) that constitutes the data we have, and use the line to predict $\hat{\boldsymbol{y}}$ based on $\hat{\boldsymbol{x}}$

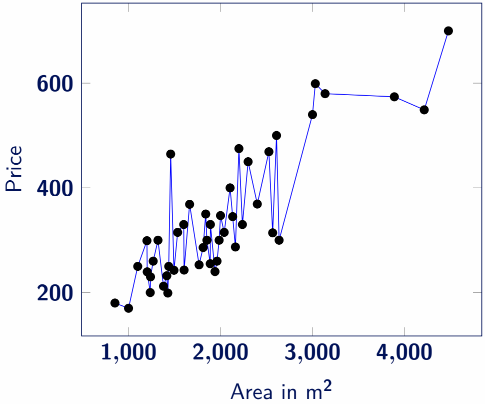
This is called a model **generalization**, i.e., you obtain a model from a data set and apply it to another data set(s). This is a fundamental concept in Machine & Deep Learning.

Linear regression



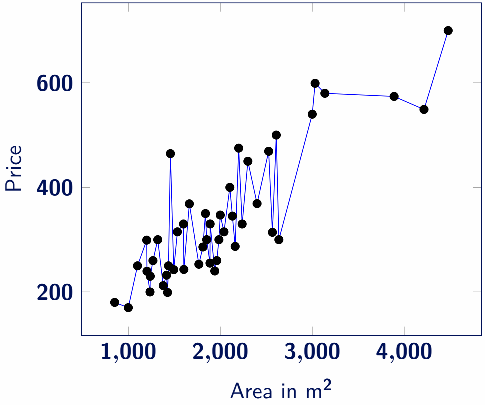
The data do form a linear fashion, but how to find a good line/model?

Linear regression



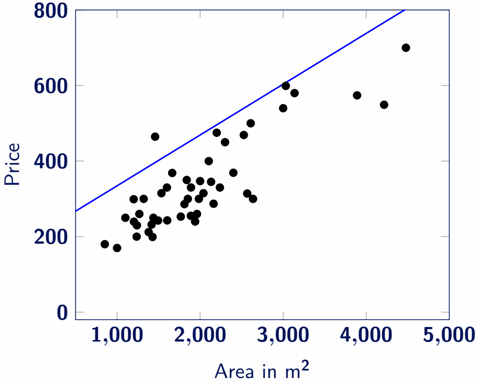
Draw a line by connecting all the points like this?

Linear regression



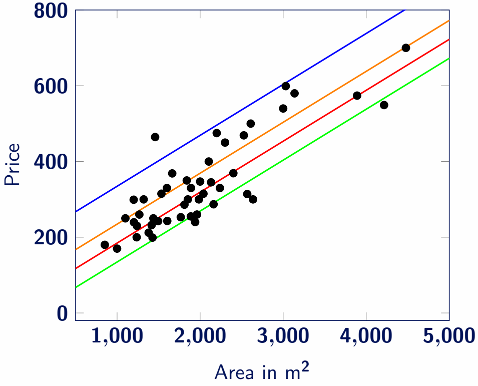
Draw a line by connecting all the points like this? Recall that our objective is a model generalization; the model above will not fit well other data.

Linear regression



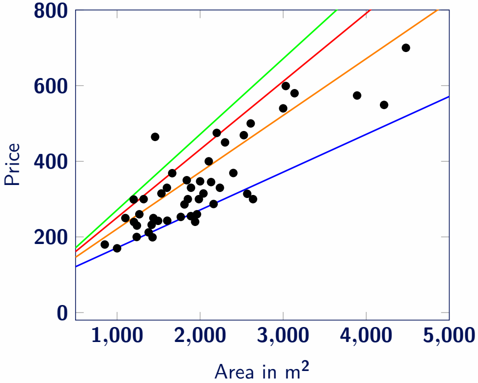
Draw a line like this?

Linear regression



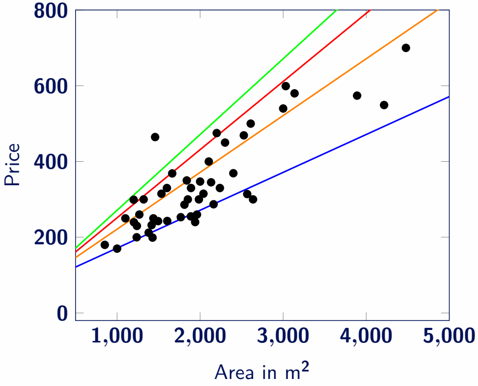
But which line?

Linear regression

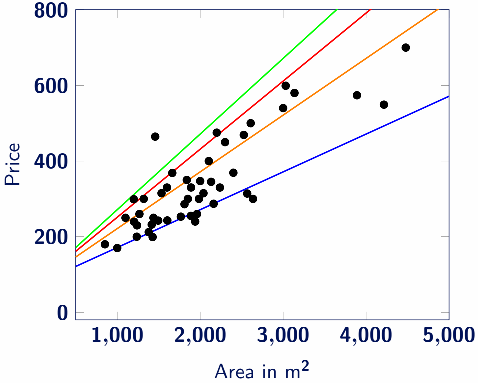


But which line?

Linear regression

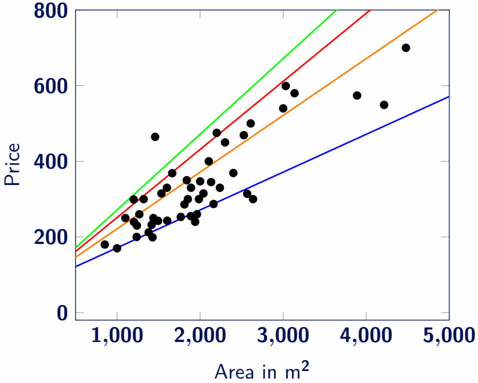


But which line? What is the criteria of a *good* line/model? Define ones.



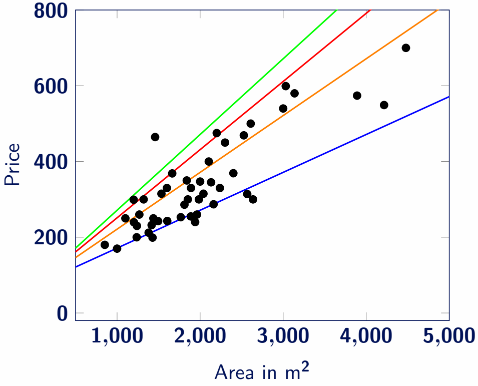
Informally, we can think of a *good* line/model is the one that is generally close to all data points.

Linear regression



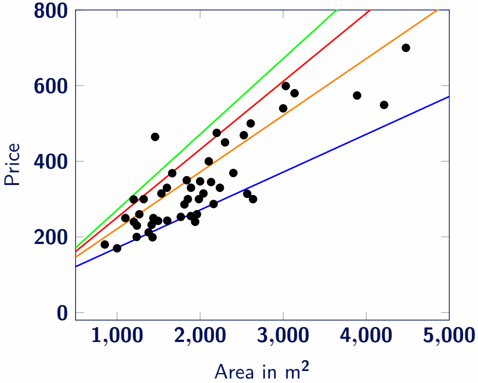
To indicate how close a line, we can measure distances between data points to the line. The total distance is often called **error**; the lower it is, the better.

Linear regression



Linear regression is about to find the *best* line by selecting the one with the minimum error. How?

Linear regression



Recall that we can search lines by changing the values of intercept θ_0 and slope θ_1 in $y = \theta_0 + \theta_1 x$. But of course, we do not want to search **randomly**.

Linear Regression

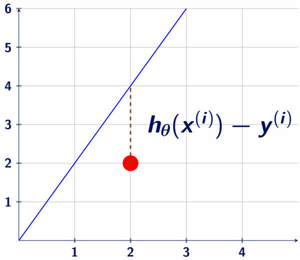
More formally, a straight line can be represented by,

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x,$$

More formally, a straight line can be represented by,

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x,$$

- The error term indicates the distance between a point $y^{(i)}$ to the line $h_\theta(x^{(i)})$, i.e., $h_\theta(x^{(i)}) - y^{(i)}$.



The total error, or often called **cost function** in ML/DL, thus can be written

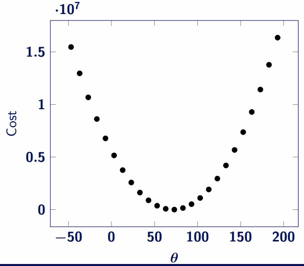
$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2.$$

Linear Regression

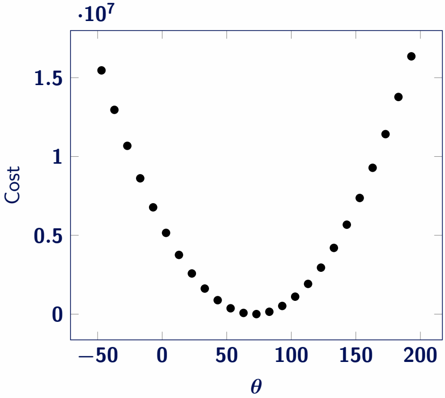
The total error, or often called **cost function** in ML/DL, thus can be written

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2.$$

The function above is called the **ordinary least square**. Here is the plot.

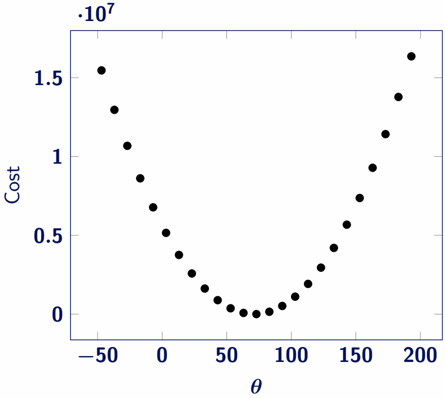


Ordinary least square



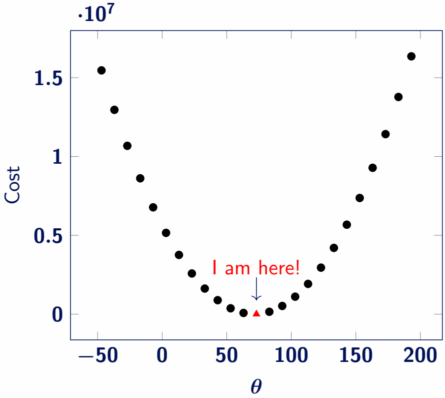
- Hey, we get some clue here!

Ordinary least square



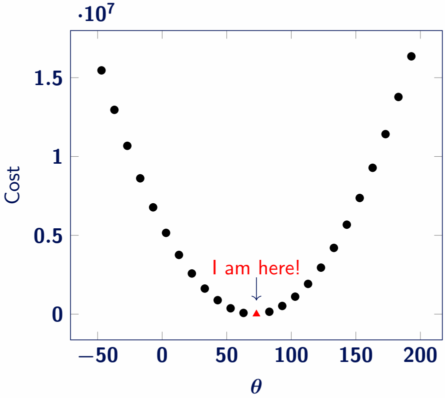
- Hey, we get some clue here!
- Obviously we know which one the minimum cost is, don't we?

Ordinary least square



- Hey, we get some clue here!
- Obviously we know which one the minimum cost is, don't we?
- That is, look at "I am here"

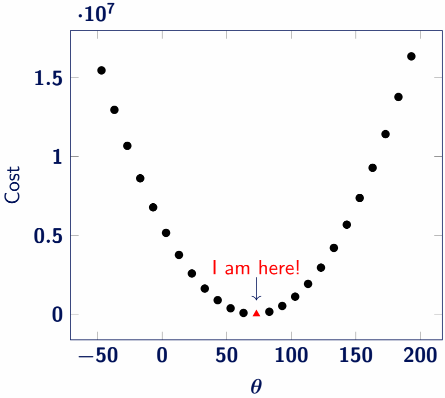
Ordinary least square



- Hey, we get some clue here!
- Obviously we know which one the minimum cost is, don't we?
- That is, look at "I am here"

Each cost (data point) above represents an error term of a line/model. The question: How to find the minimum error?

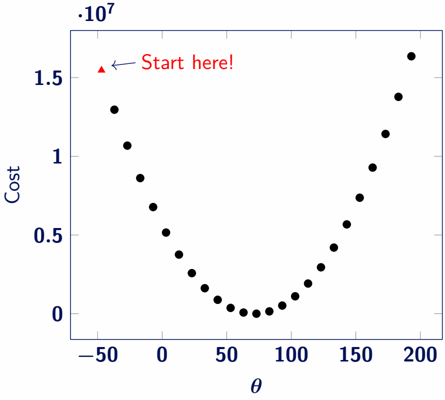
Ordinary least square



- Hey, we get some clue here!
- Obviously we know which one the minimum cost is, don't we?
- That is, look at "I am here"

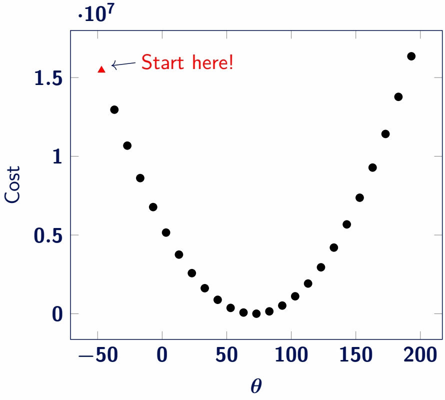
How to find the minimum error? Answer: Simply going down to the *valley bottom*.

Ordinary least square



- Typically your randomly initialized line returns the "start here" cost. How can we go down?

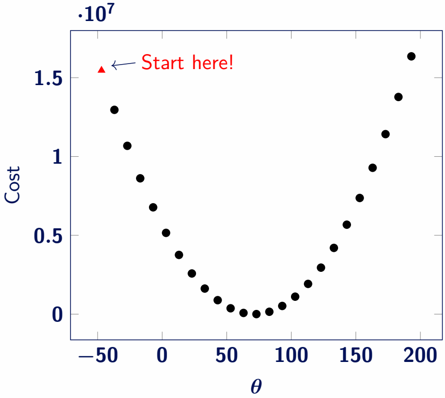
Ordinary least square



- Typically your randomly initialized line returns the "start here" cost. How can we go down?
- We can use the **gradient descent** to update **parameter** $\boldsymbol{\theta}$, so as to get the minimum cost

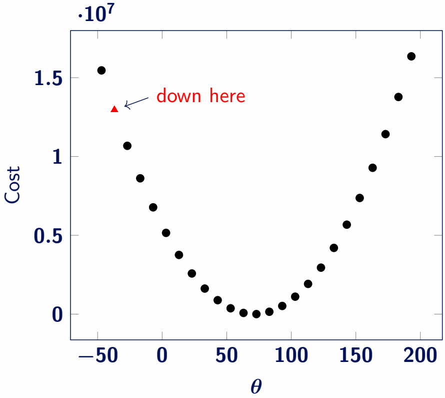
We now use the term **model parameter** to represent slope and intercept.

Gradient descent



- Initially *guest* $\boldsymbol{\theta}$, compute the cost (see Start here)

Gradient descent



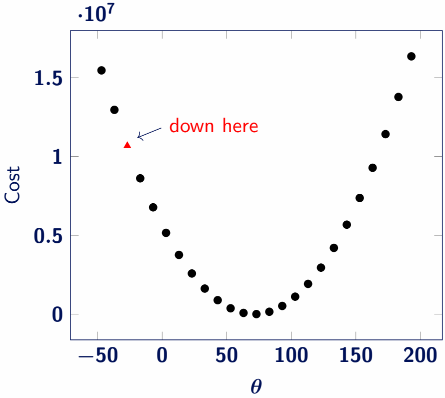
- Initially *guest* $\boldsymbol{\theta}$, compute the cost (see Start here)
- Repeatedly, update the parameter (see "down here") via

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

simultaneously for all $j = 0, \ldots, n$.

down again...

Gradient descent



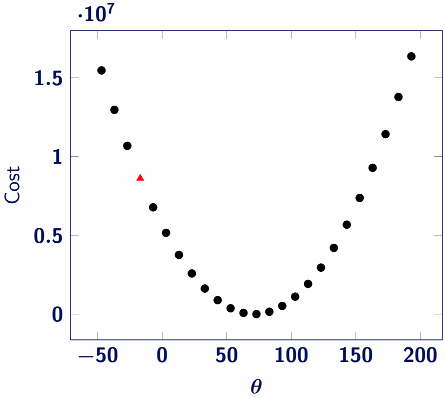
- Initially *guest* $\boldsymbol{\theta}$, compute the cost (see Start here)
- Repeatedly, update the parameter (see "down here") via

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

simultaneously for all $j = 0, \ldots, n$.

down again...

Gradient descent



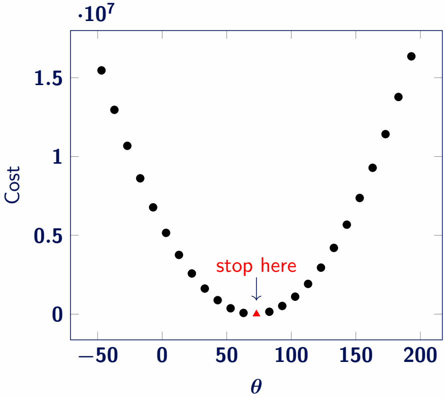
- Initially *guest* $\boldsymbol{\theta}$, compute the cost (see Start here)
- Repeatedly, update the parameter (see "down here") via

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

simultaneously for all $j = 0, \ldots, n$.

so on...

Gradient descent



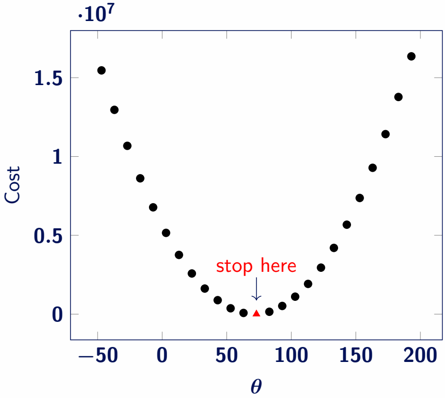
- Initially *guest* θ, compute the cost (see Start here)
- Repeatedly, update the parameter (see "down here") via

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

simultaneously for all $j = 0, \ldots, n$.

until reaching the minimum error.

Gradient descent



- Once you reach the minimum error, the corresponding parameter $\boldsymbol{\theta}$ should give you the *fittest* line/model

Linear Regression: the procedure

1. Pick an initial line/model $h(\theta)$ by randomly choosing parameter θ
2. Compute the corresponding cost function, e.g.,

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2,$$

3. Update the line/model $h(\theta)$ by updating θ that makes $J(\theta)$ smaller, using, e.g., the gradient descent that reads

$$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta),$$

   where α is the learning rate.
4. Repeat steps 2 and 3 until converges

Linear regression: a summary

In more detail

The learning procedure above is a typical approach in mostly (parametric) models of machine and deep learning.

Note that we have been so far assuming the $\boldsymbol{y}$ in our data set $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}); \boldsymbol{i} = \mathbf{1}, \ldots, \boldsymbol{m}$ is continuous, e.g., house price, height, temperature, etc.

Linear regression: a summary

In more detail
The learning procedure above is a typical approach in mostly (parametric) models of machine and deep learning.

Note that we have been so far assuming the $\boldsymbol{y}$ in our data set $(\boldsymbol{x}^{(\boldsymbol{i})}, \boldsymbol{y}^{(\boldsymbol{i})})$; $\boldsymbol{i} = \mathbf{1}, \ldots, \boldsymbol{m}$ is continuous, e.g., house price, height, temperature, etc.

What if $\boldsymbol{y}$ is discrete? E.g., pass/fail, yes/true, 1/0, etc, that leads to nonlinear fashion and a classification task.

Linear regression: a summary

In more detail

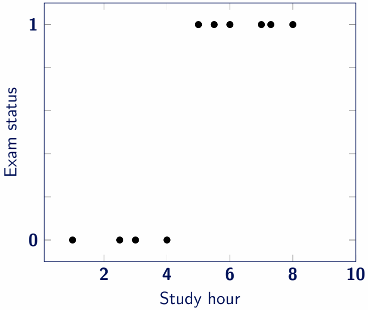
The learning procedure above is a typical approach in mostly (parametric) models of machine and deep learning.

Note that we have been so far assuming the $\boldsymbol{y}$ in our data set $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$; $\boldsymbol{i} = \mathbf{1}, \ldots, \boldsymbol{m}$ is continuous, e.g., house price, height, temperature, etc.

What if $\boldsymbol{y}$ is discrete? E.g., pass/fail, yes/true, 1/0, etc, that leads to nonlinear fashion and a classification task.

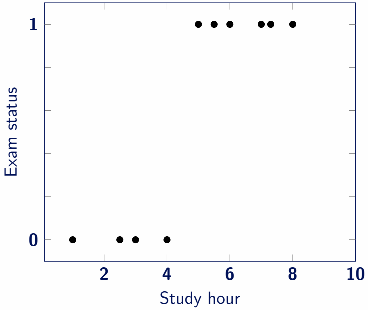
Let's see the plot example.

If y is a discrete variable



Exam status: 0 (fail), 1 (pass)

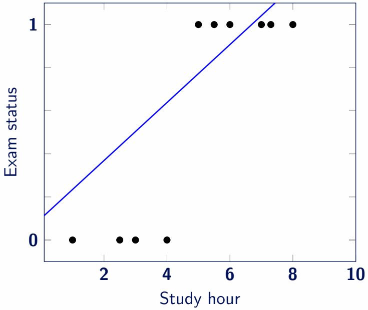
If y is a discrete variable



Exam status: 0 (fail), 1 (pass)

- Do you think the linear model $y = \theta_0 + \theta_1 * x$ will fit well?

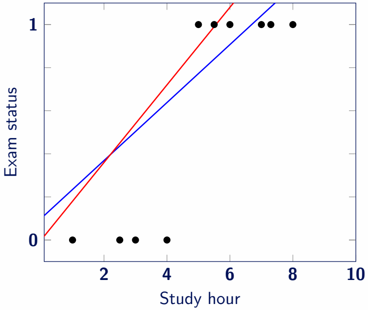
If y is a discrete variable



Exam status: 0 (fail), 1 (pass)

- Do you think the linear model $y = \theta_0 + \theta_1 * x$ will fit well?
- Let's try one.

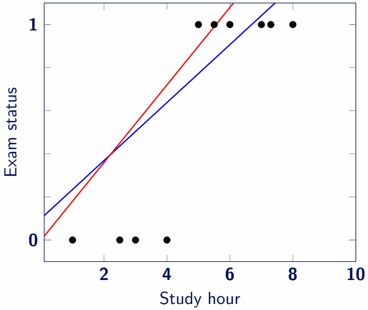
If y is a discrete variable



Exam status: 0 (fail), 1 (pass)

- Do you think the linear model $y = \theta_0 + \theta_1 * x$ will fit well?
- Let's try one
- Looks bad, try another one.

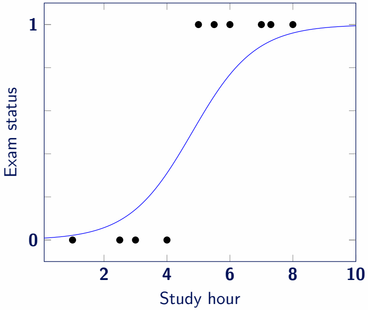
If y is a discrete variable



Exam status: 0 (fail), 1 (pass)

- Do you think the linear model $y = \theta_0 + \theta_1 * x$ will fit well?
- Let's try one
- Looks bad, try another one.
- We see that such a linear model won't fit well, because of the nonlinear fashion. We need another model!

If y is a discrete variable



Exam status: 0 (fail), 1 (pass)

- We can use the logistic (sigmoid) function that reads

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

- We typically predict "1" if $h_\theta(x) \geq 0.5$

Logistic Regression: the procedure

1. Pick an initial line/model $\boldsymbol{h}(\boldsymbol{\theta})$ by randomly choosing parameter $\boldsymbol{\theta}$
2. Compute the corresponding cost function, e.g.,

$$\boldsymbol{J}(\boldsymbol{\theta}) = \sum_{i=1}^{m} -y \log h_\theta(x^{(i)}) - (1 - y)\log(1 - h_\theta(x^{(i)}))$$

3. Update the line/model $\boldsymbol{h}(\boldsymbol{\theta})$ by updating $\boldsymbol{\theta}$ that makes $\boldsymbol{J}(\boldsymbol{\theta})$ smaller, using, e.g., the gradient descent that reads

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta),$$

   where α is the learning rate.
4. Repeat steps 2 and 3 until converges