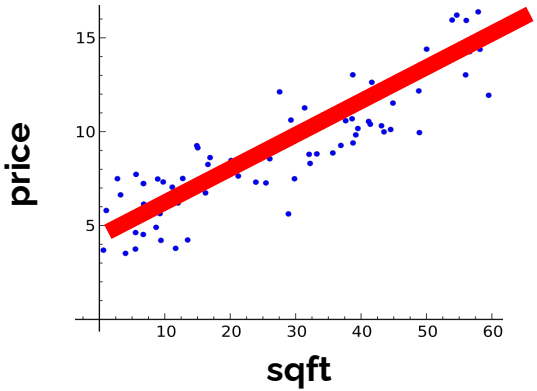
University of Exeter

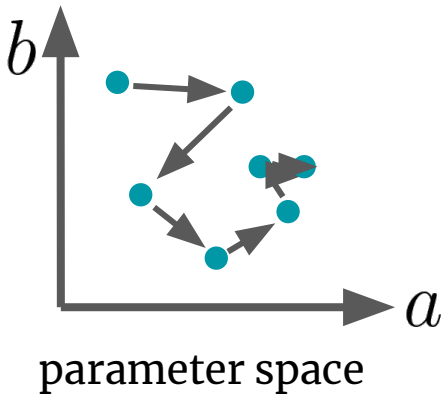
COM1011

Fundamentals of Machine Learning

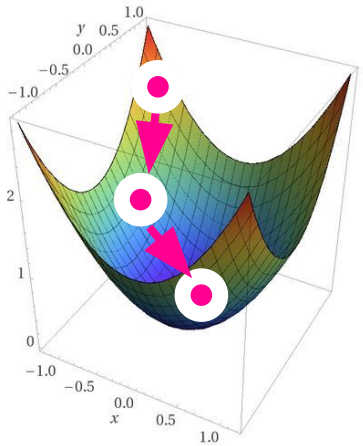
Previously on COM1011:

- Linear regression and how to fit parameters
- The core ideas behind gradient descent



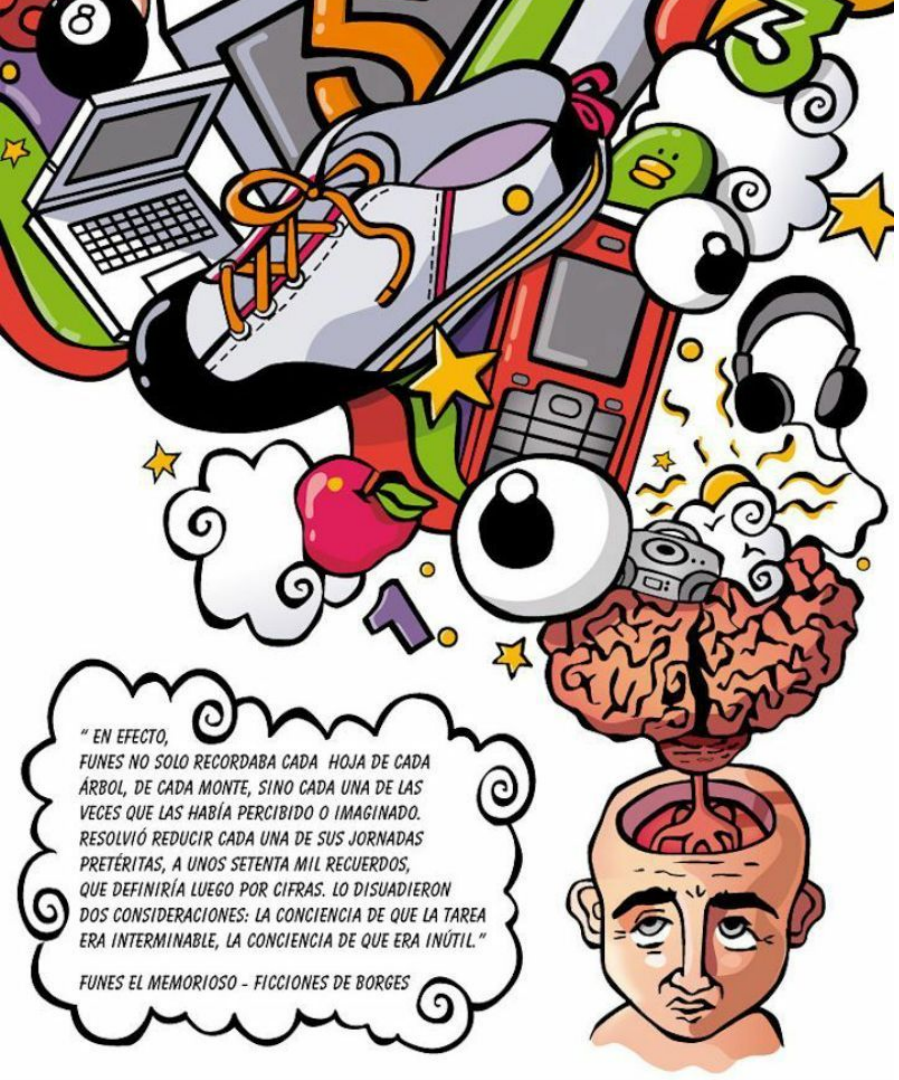


parameter space



Today:

- Overfitting
- Training data, testing data
- R^2
- Linear regression on more than one variable



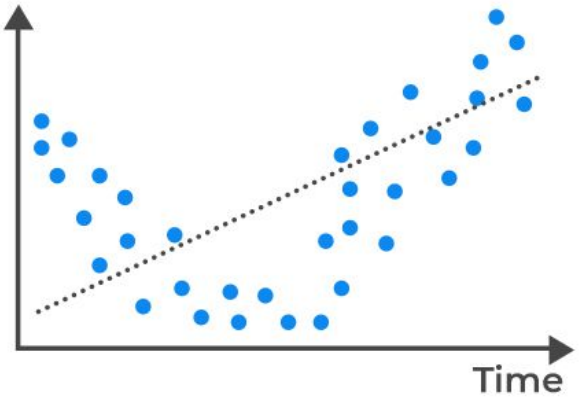
Funes, the memorious

A fantasy short story by
Jorge Luis Borges

- Is perfect memory equal to superintelligence?
- What did Funes lack?

Overfitting

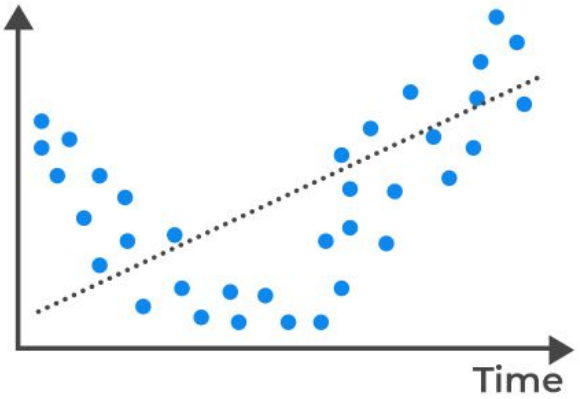
$$y = ax + b$$



Underfitted

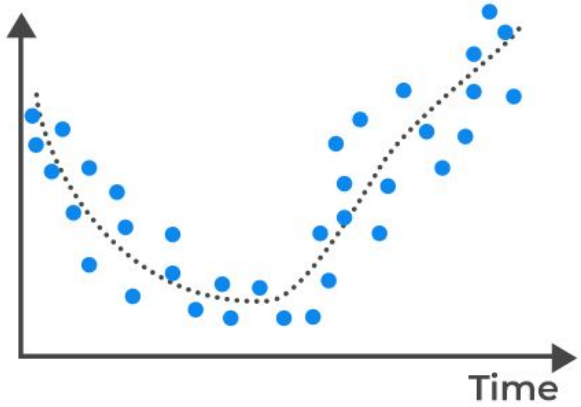
Overfitting

$$y = ax + b$$



Underfitted

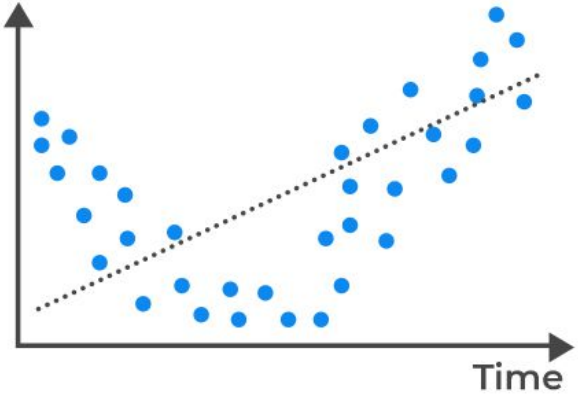
$$y = ax^2 + bx + c$$



Good Fit/Robust

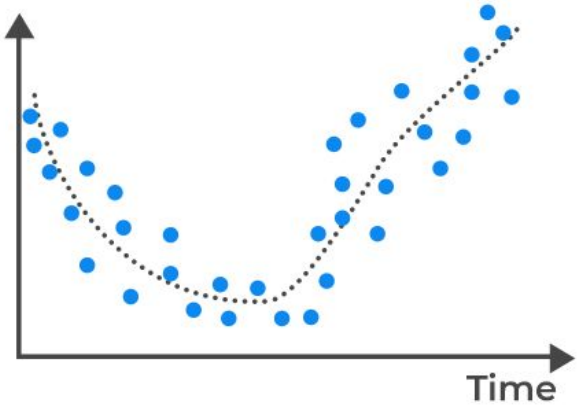
Overfitting

$y = ax + b$



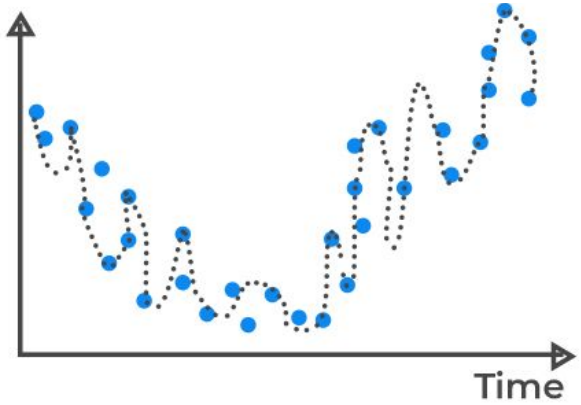
Underfitted

$y = ax^2 + bx + c$



Good Fit/Robust

Funes, the memorious

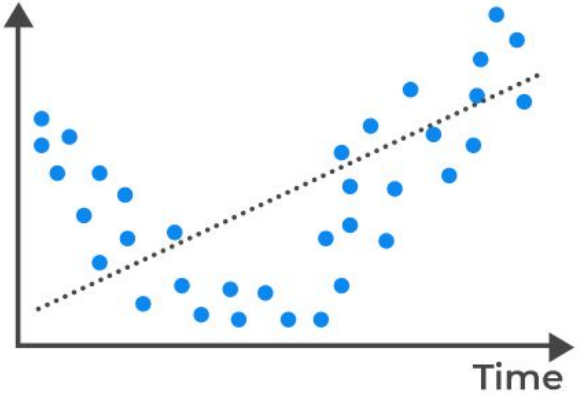


Overfitted

Overfitting

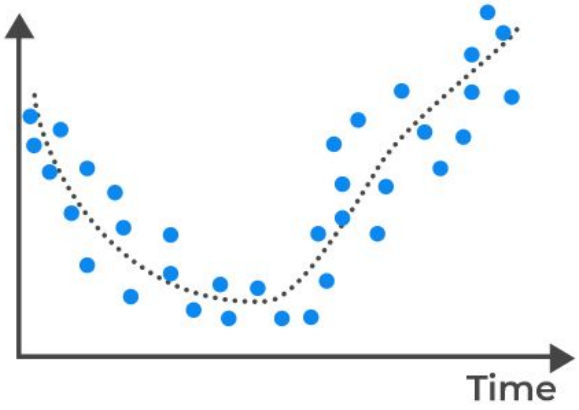
Funes lacked the ability to generalise.

$$y = ax + b$$

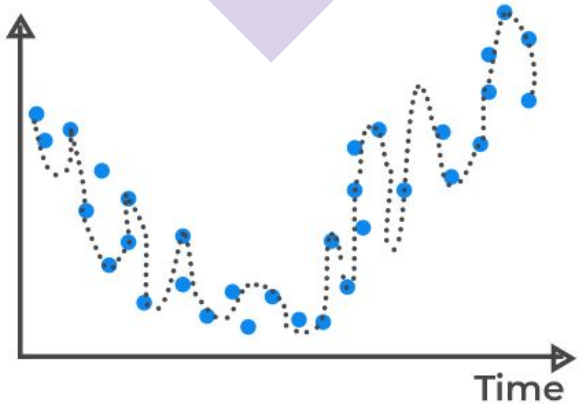


Underfitted

$$y = ax^2 + bx + c$$



Good Fit/Robust

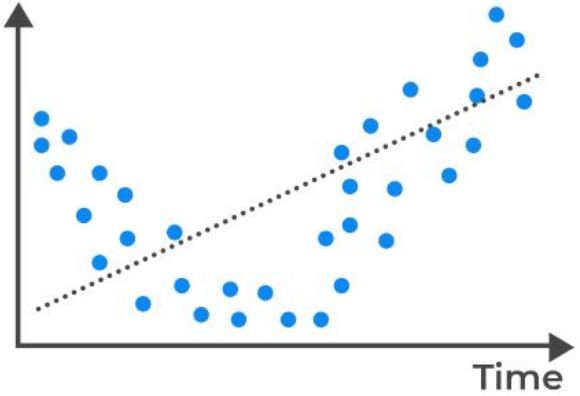


Overfitted

Overfitting

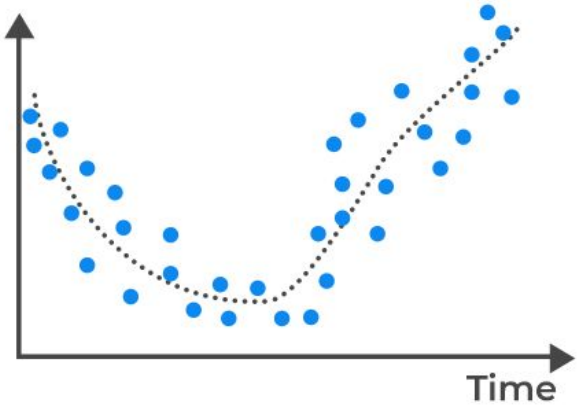
Funes lacked the ability to generalise.

$y = ax + b$

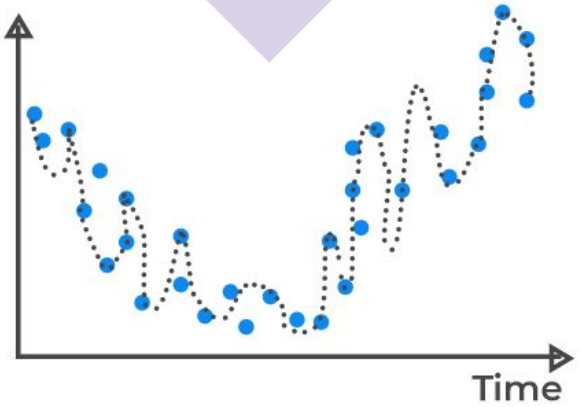


Underfitted

$y = ax^2 + bx + c$



Good Fit/Robust

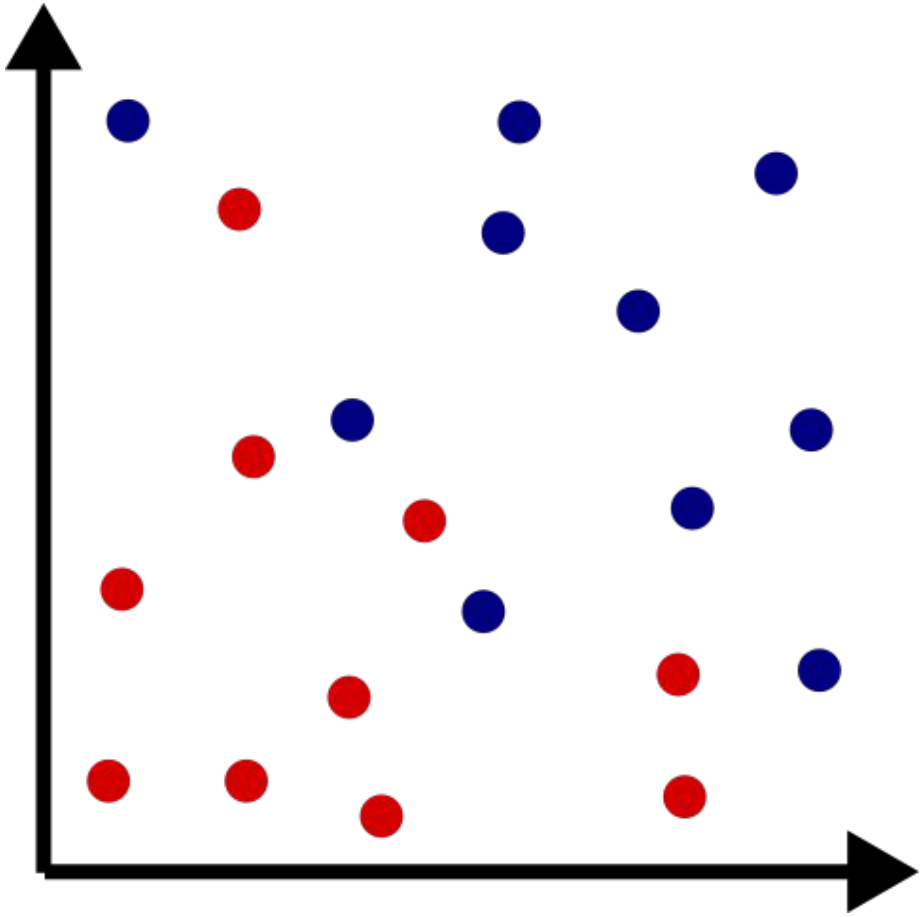


Overfitted

How can we avoid overfitting?

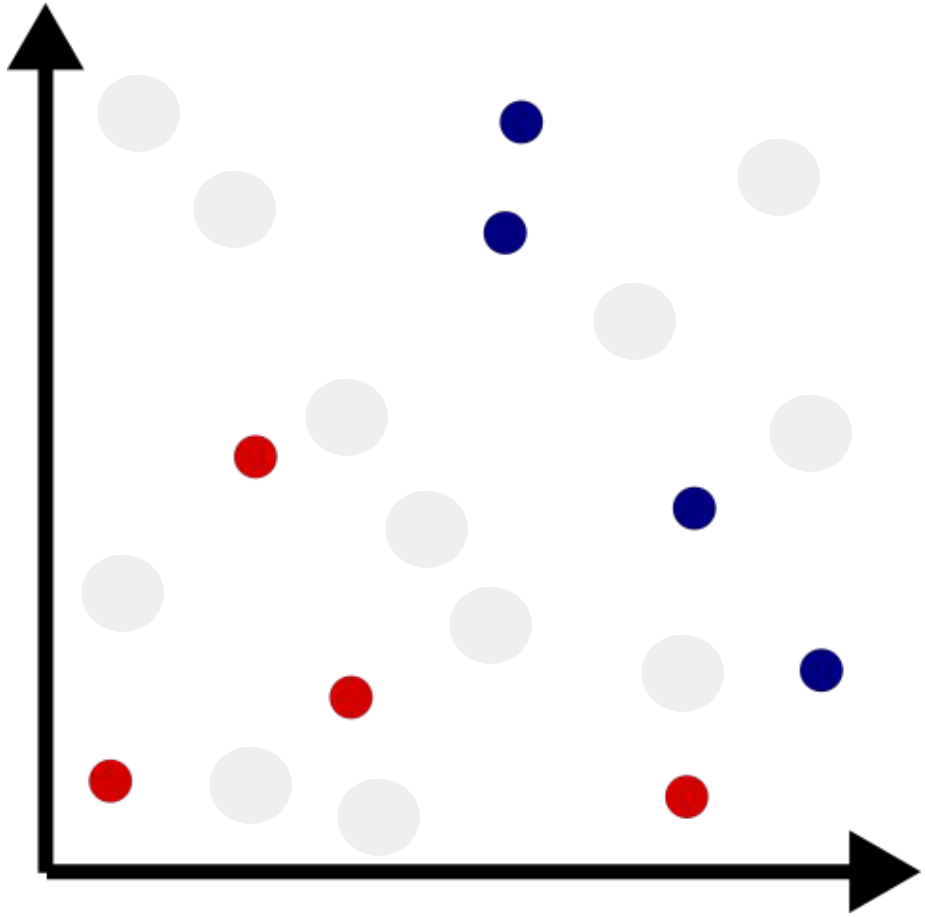
Train-test split

- Train your algorithm on part of the data



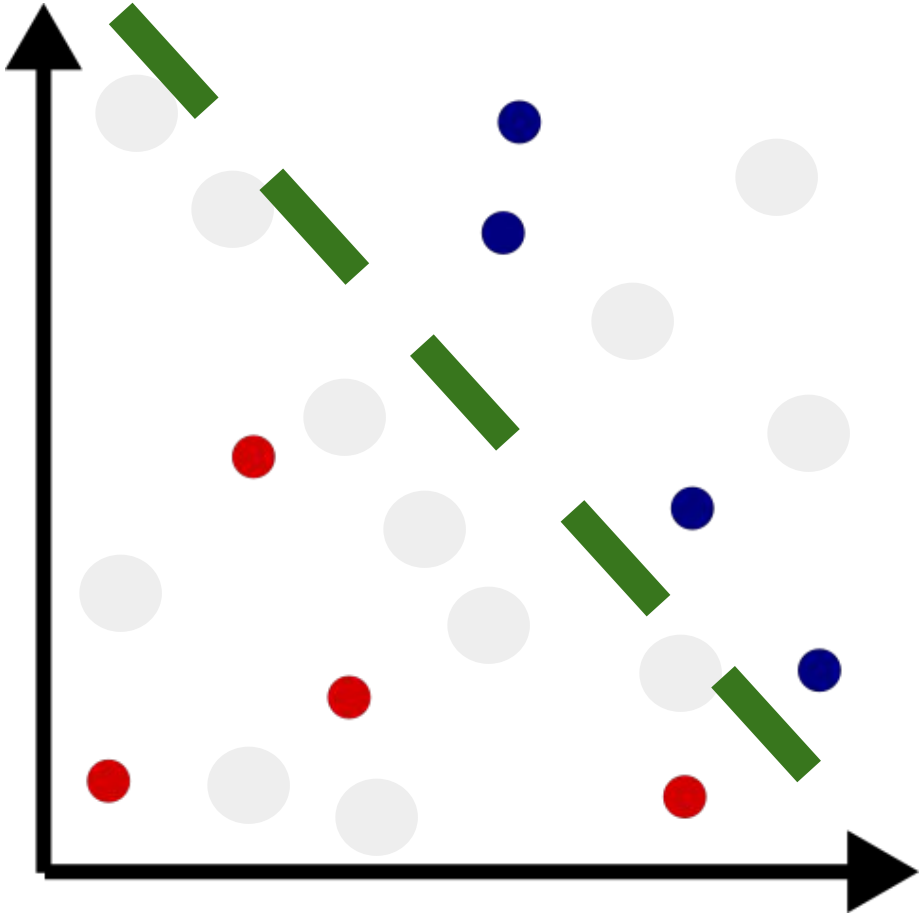
Train-test split

- Train your algorithm on part of the data



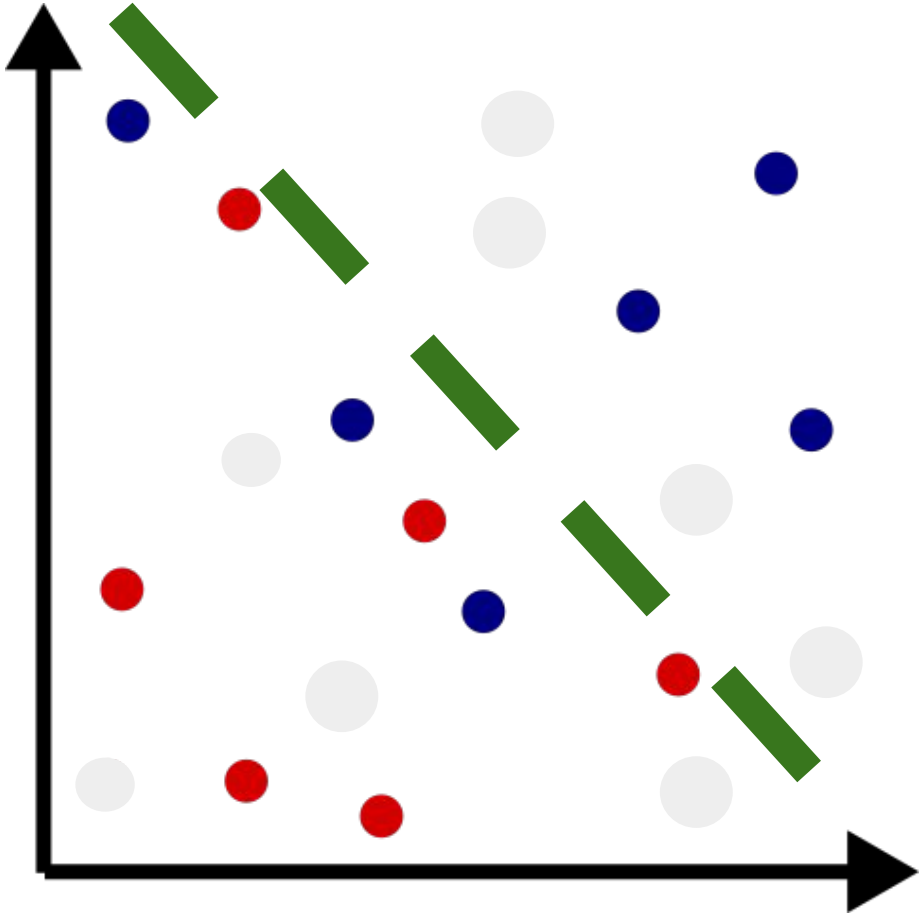
Train-test split

- Train your algorithm on part of the data



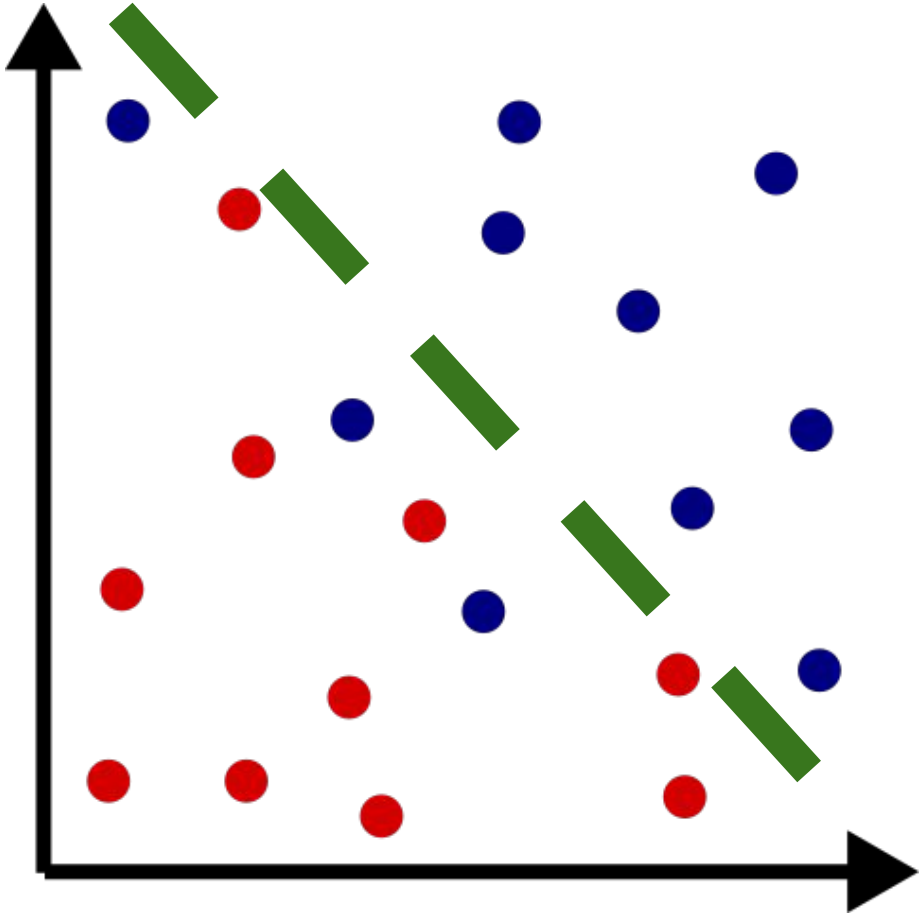
Train-test split

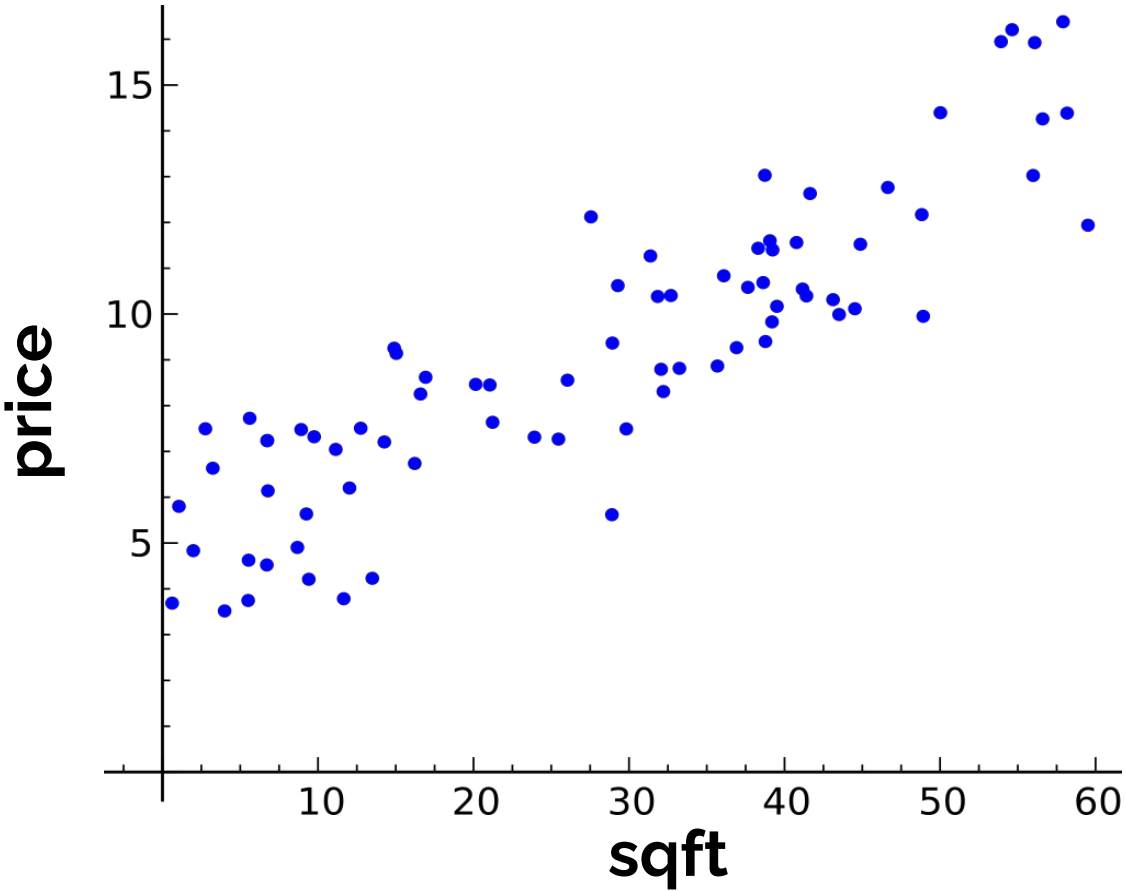
- Train your algorithm on part of the data
- Test it on another part



Train-test split

- Train your algorithm on part of the data
- Test it on another part
- Splits are usually around 80% train, 20% test
- This can be used with all sorts of ML algorithms

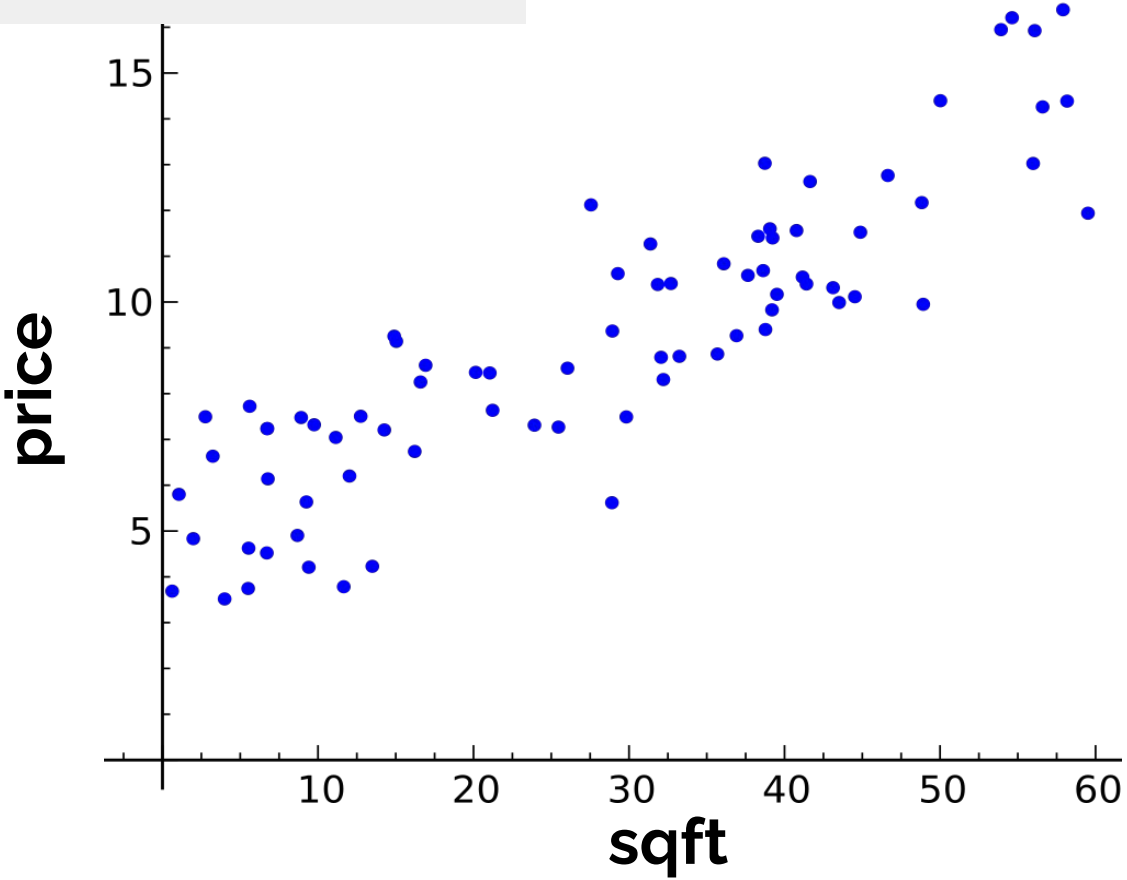




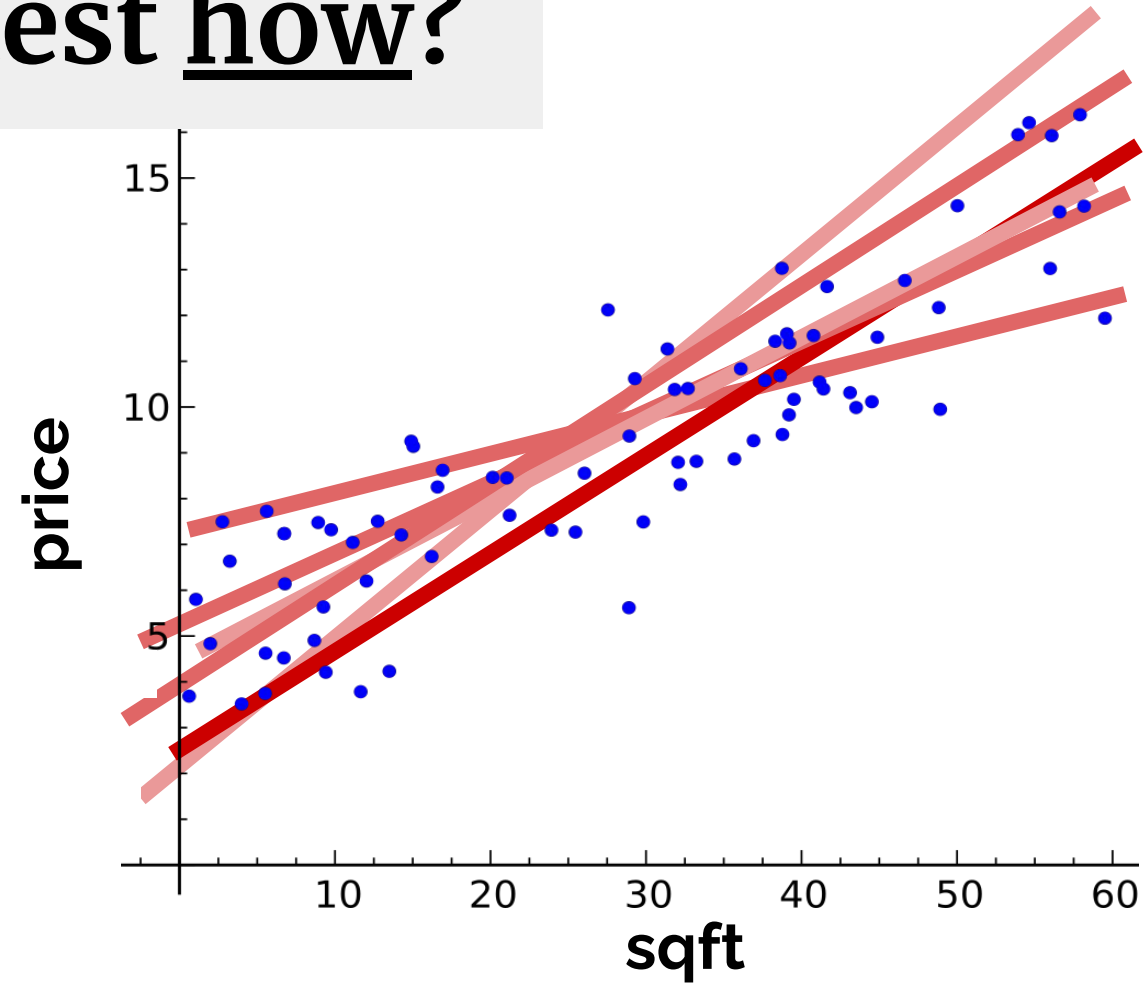
80% of the datapoints to train the linear regression

20% to test if it worked

But test how?

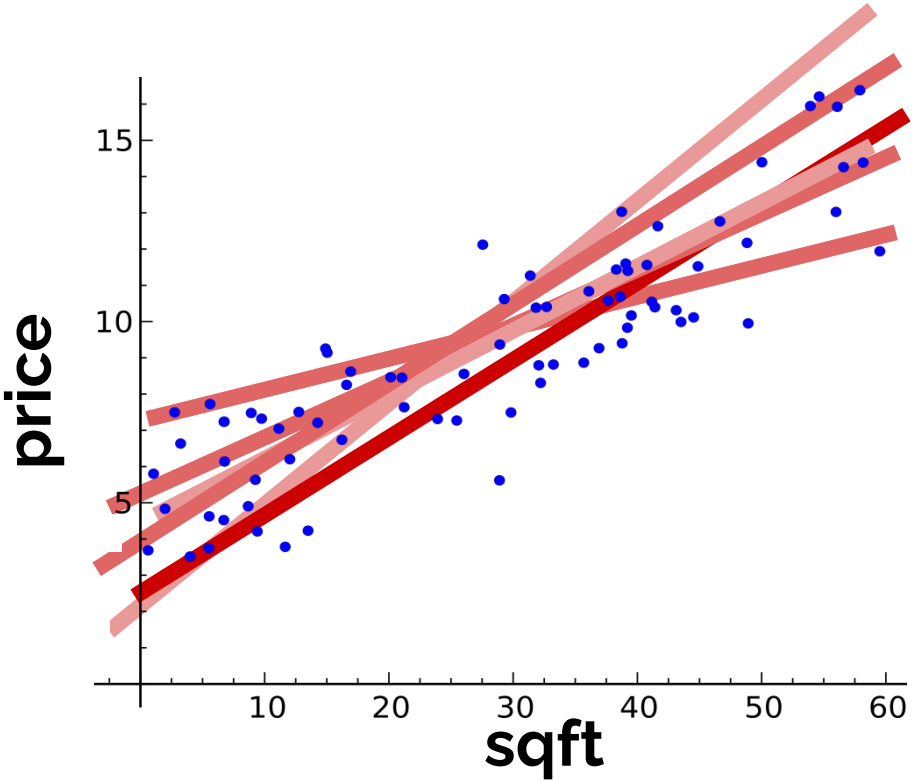


But test how?



But test how?

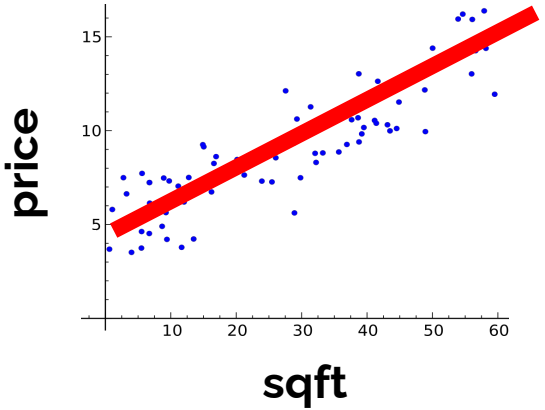
If you're simply comparing linear regressions on the same variable, whichever line minimises the loss function (such as the L2 norm) is the best fit.

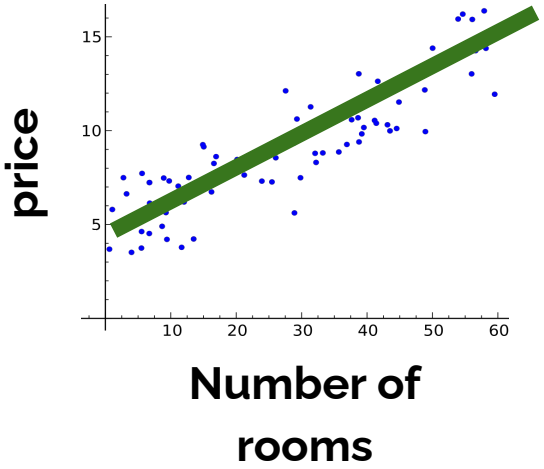


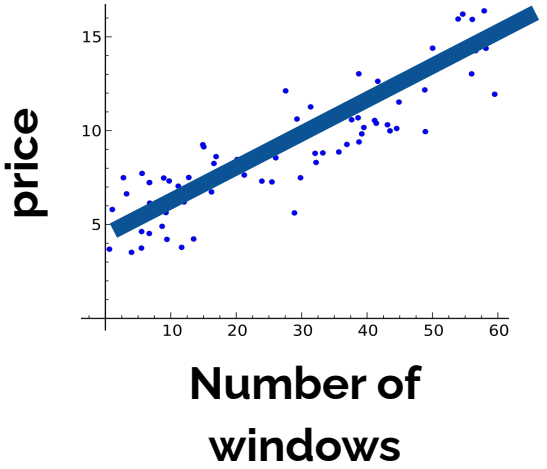
But test how?

What if they are on different variables?

Does it still work to compare loss functions?







But test how?

What if they are on different variables?

Does it still work to compare loss functions?

L1 norm:

$$\left\| y^{\text{pred}} - y^{\text{data}} \right\|_1 := \sum_{i=1}^{m} \left| y_i^{\text{pred}} - y_i^{\text{data}} \right|$$

L2 norm:

$$\left\| y^{\text{pred}} - y^{\text{data}} \right\|_2 := \left(\sum_{i=1}^{m} \left| y_i^{\text{pred}} - y_i^{\text{data}} \right|^2 \right)^{1/2}$$

But test how?

What if they are on different variables?

Does it still work to compare loss functions?

Answer:

NO.

L1 norm:

$$\left\| y^{\text{pred}} - y^{\text{data}} \right\|_1 := \sum_{i=1}^{m} \left| y_i^{\text{pred}} - y_i^{\text{data}} \right|$$

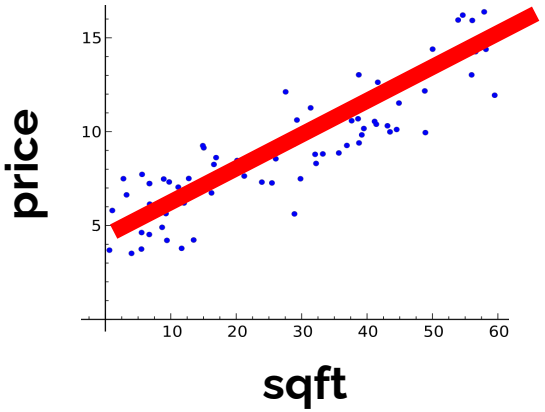
L2 norm:

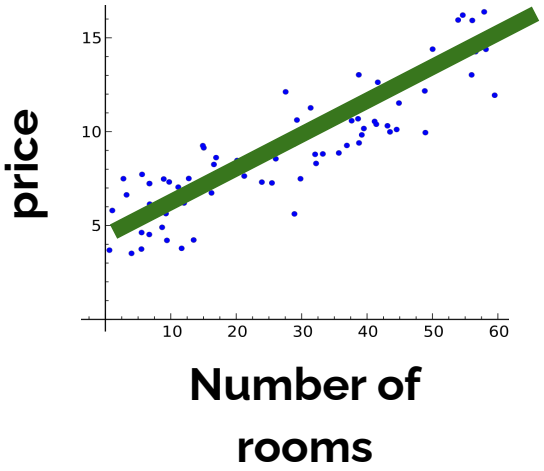
$$\left\| y^{\text{pred}} - y^{\text{data}} \right\|_2 := \left(\sum_{i=1}^{m} \left| y_i^{\text{pred}} - y_i^{\text{data}} \right|^2 \right)^{1/2}$$

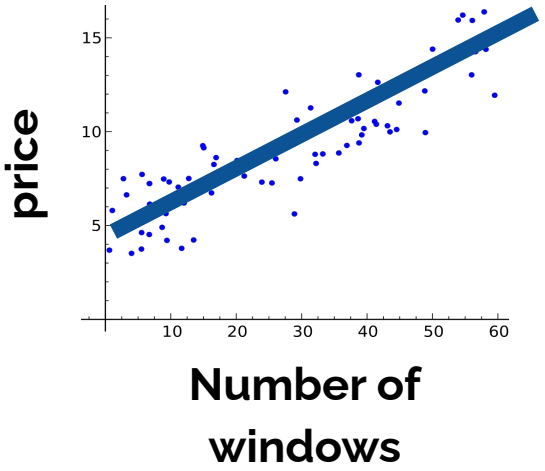
But test how?

To compare linear regressions, we need the same metric for all:

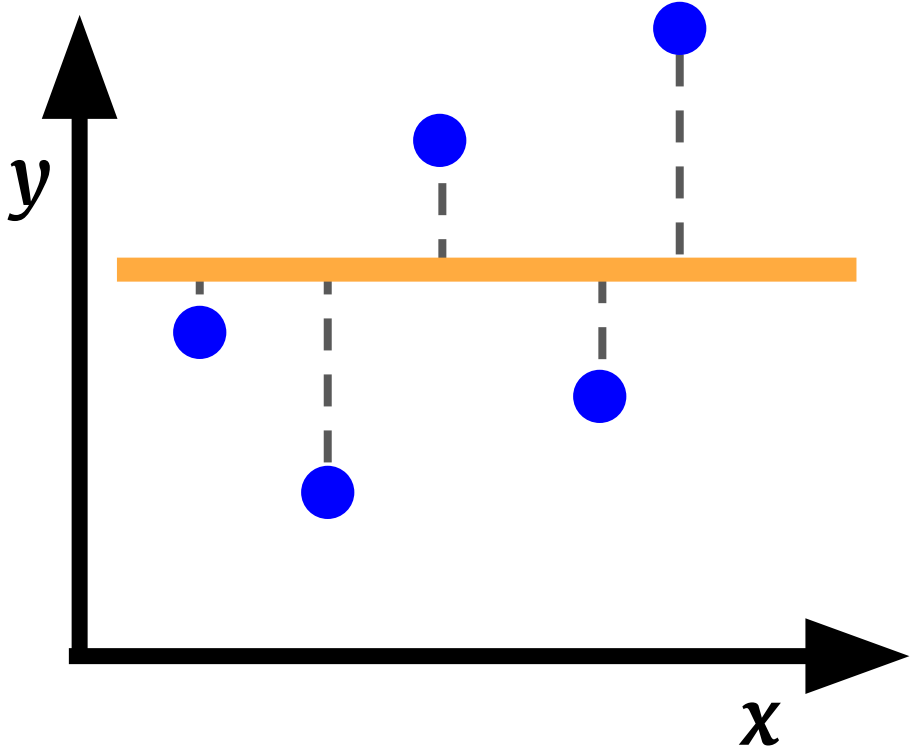
how good they are at predicting the *y* variable.







The R^2 metric

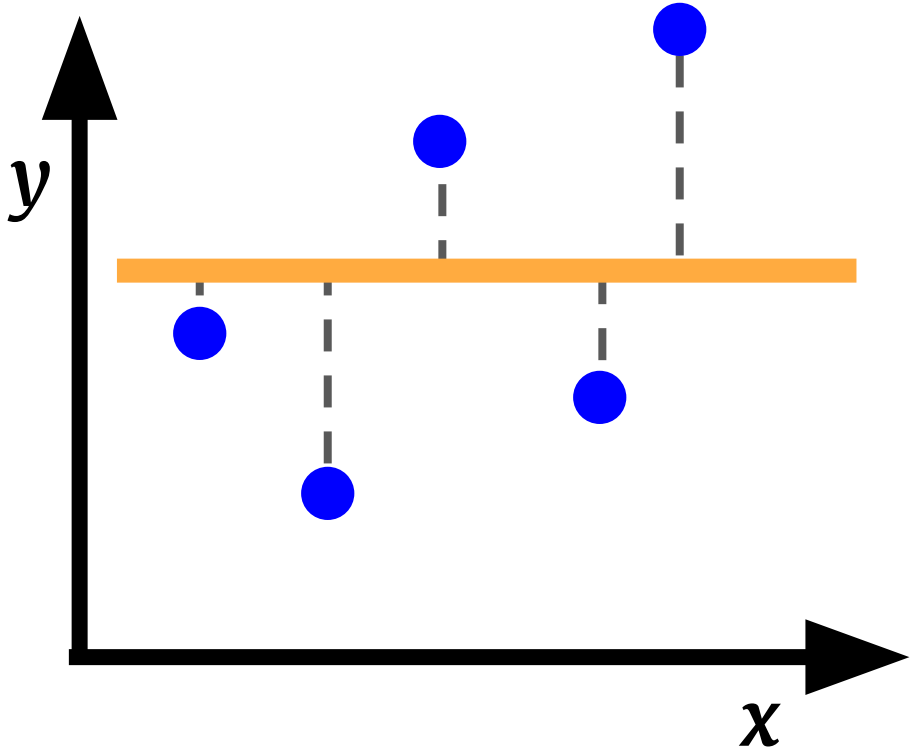


R^2 measures how good the prediction is, when compared to a model with **zero slope** (so... a constant.)

First we calculate the **Sum of Squared Estimates** for this simple model:

$$SSE^{\text{zero}} = \sum_{i=1}^{m} \left(y_i^{\text{zero}} - y_i^{\text{data}} \right)^2$$

The R^2 metric

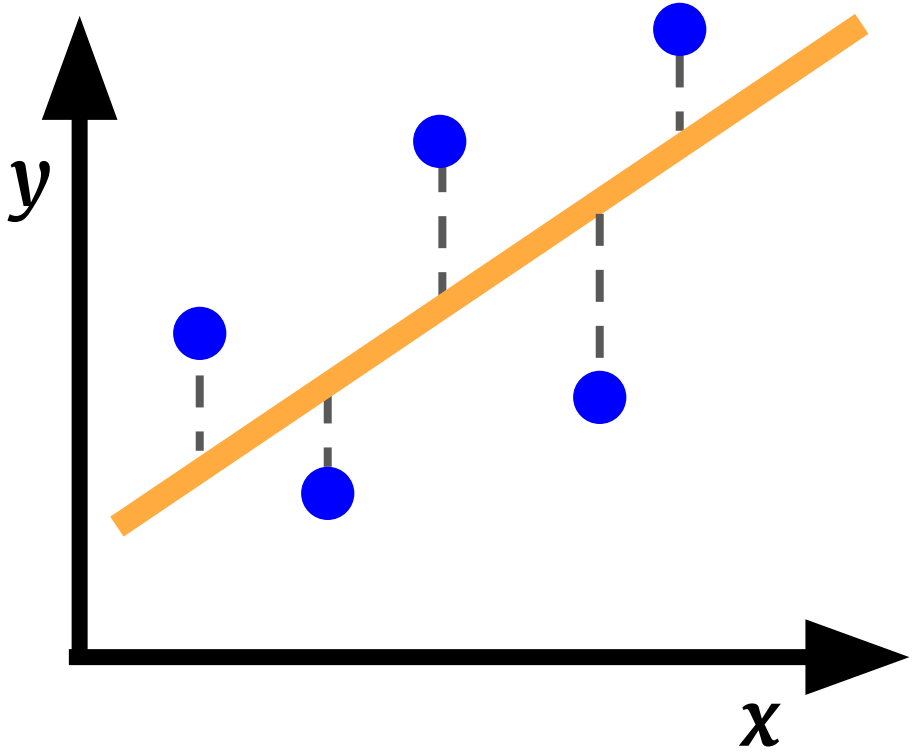


R^2 measures how good the prediction is, when compared to a model with **zero slope** (so... a constant.)

First we calculate the **Sum of Squared Estimates** for this simple model:

$$SSE^{\text{zero}} = \sum_{i=1}^{m} \left(\bar{y} - y_i^{\text{data}} \right)^2$$

The R^2 metric



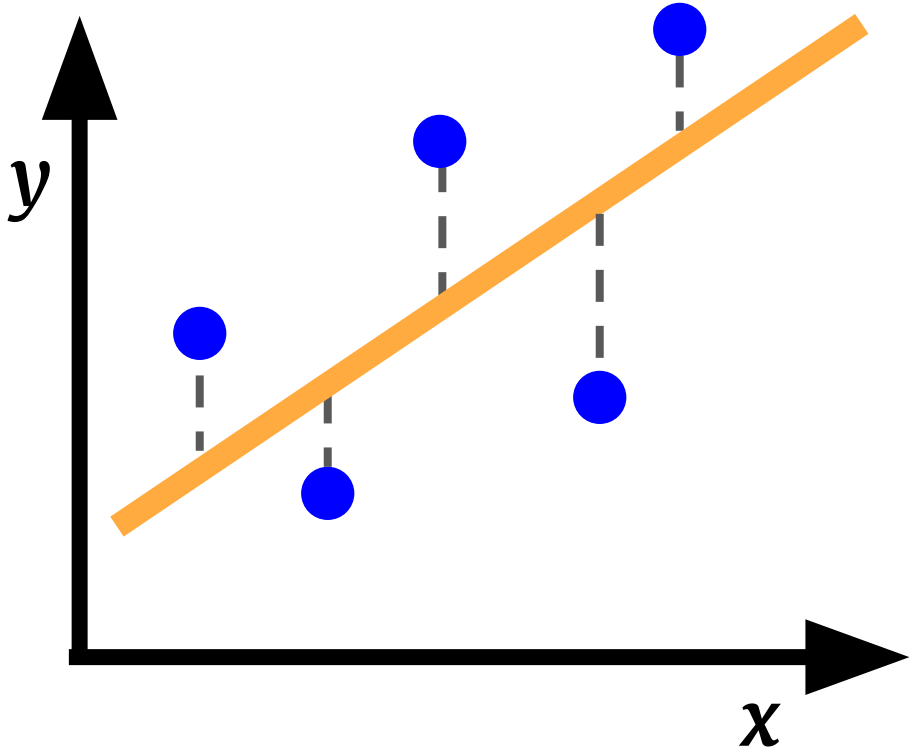
Then we look at every data point, and compare its predicted y value (y^{pred}) with its actual y value (y^{data}).

We then calculate the **Sum of Squared Estimates** for our linear model:

$$SSE = \sum_{i=1}^{m} \left(y_i^{\text{pred}} - y_i^{\text{data}} \right)^2$$

The R^2 metric

$$SSE^{\text{zero}} = \sum_{i=1}^{m} \left(\bar{y} - y_i^{\text{data}} \right)^2$$

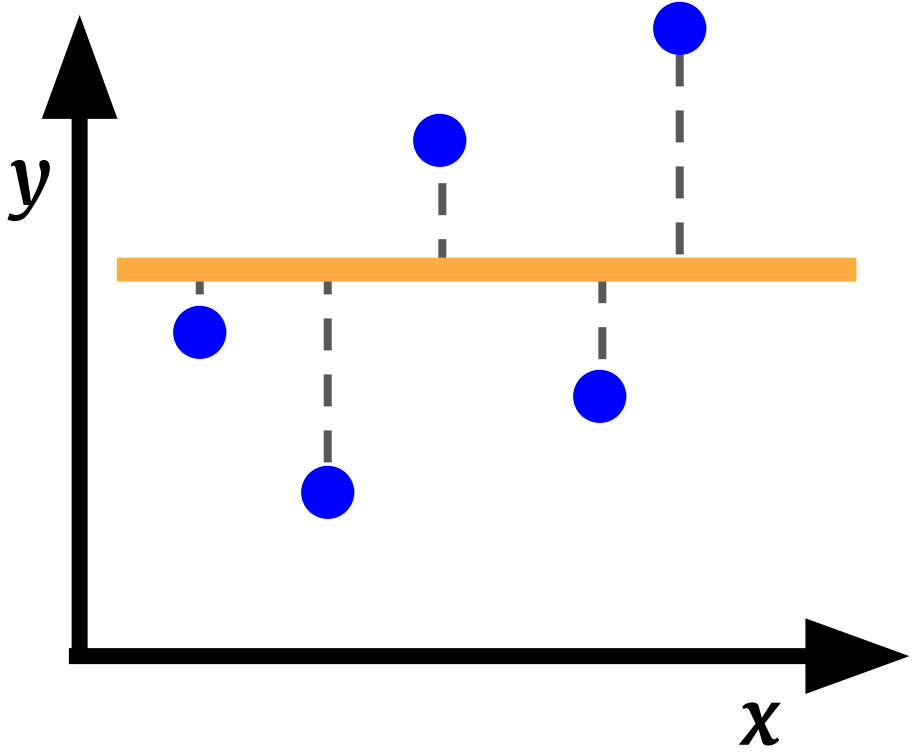


$$SSE = \sum_{i=1}^{m} \left(y_i^{\text{pred}} - y_i^{\text{data}} \right)^2$$

Combining both:

$$R^2 = 1 - \frac{SSE}{SSE^{\text{zero}}}$$

The R^2 metric



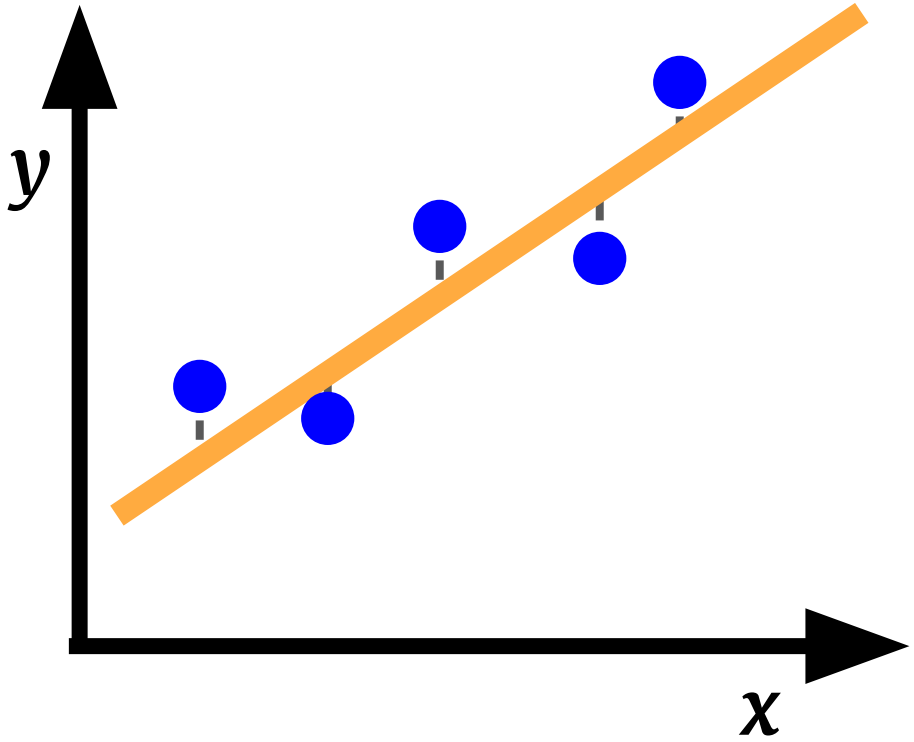
The linear model is just as good as the constant model

$\Longleftrightarrow$

$SSE \approx SSE^{zero}$ and **$R^2 \approx 0$**

$$R^2 = 1 - \frac{SSE}{SSE^{\text{zero}}}$$

The R^2 metric



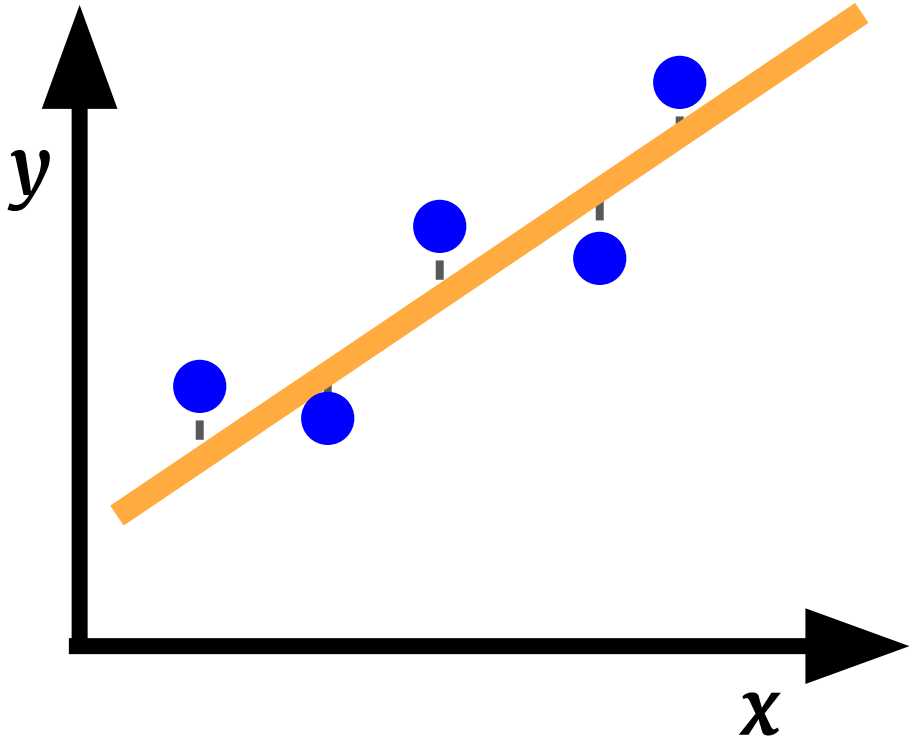
The linear model is almost as good as the original data

$\Longleftrightarrow$

SSE ≈ 0 and **R^2 ≈ 1**

$$R^2 = 1 - \frac{SSE}{SSE^{\text{zero}}}$$

The R^2 metric



Interpretation:

R^2 says how much of the variation in the y variable is explained by the linear regression.

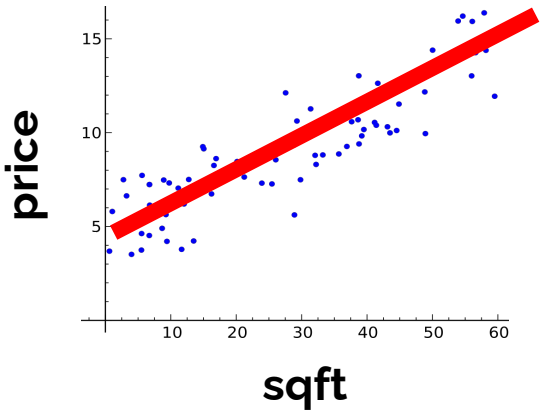
$R^2 \approx 0.65$

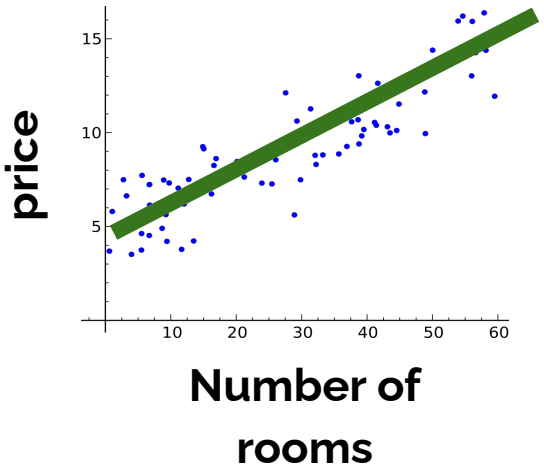
= 65% of the variation.

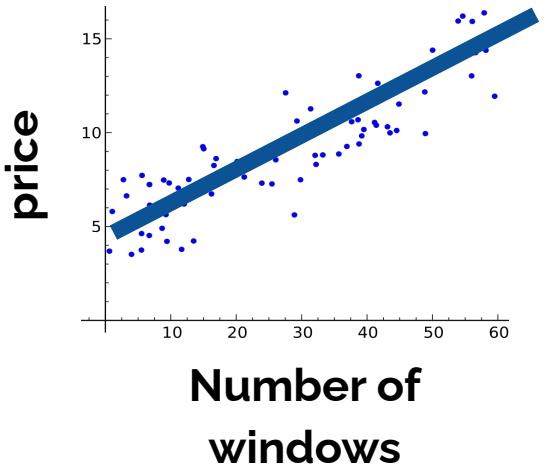
The R^2 metric

Another name for R^2:
coefficient of determination

The linear regression making the best prediction is the one with the highest R^2.



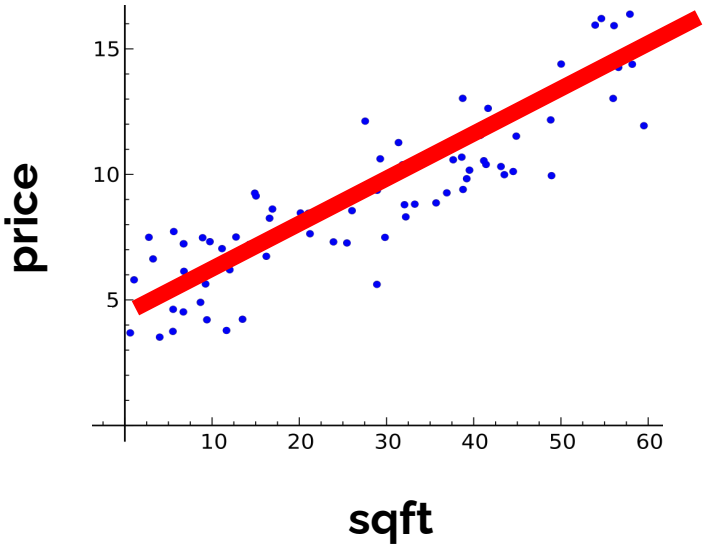




Linear regressions with N variables

y = price
x = sqft

$$y = ax + b$$



Linear regressions with *N* variables

y = price
x = sqft
z = number of rooms
w = number of windows

y = f (x, z, w)

$$y = ax + bz + cw + k$$

Linear regressions with *N* variables

y = price
x_1 = sqft
x_2 = number of rooms
x_3 = number of windows

$y = f(x_1, x_2, x_3)$ $y = a_1x_1 + a_2x_2 + a_3x_3 + b$

Linear regressions with *N* variables

y = price
x_1 = sqft
x_2 = number of rooms
x_3 = number of windows

$y = f(x_1, x_2, x_3)$

$$y = \sum_{i=1}^{3} a_i x_i + b$$

Different models = different assumptions

x = sqft
y = price

Option 1: linear model $y = ax + b$

Option 2: quadratic model $y = ax^2 + bx + c$

Option 3: sigmoid model $y = \frac{a}{1 + e^{(b-x)}}$

Different models = different assumptions

$$y = a_1 x_1 + b$$

$$y = a_2 x_2 + b$$

$$y = a_3 x_3 + b$$

Works the same way.

1. Define the model

Different models = different assumptions

$$y = a_1 x_1 + b$$

$$y = a_2 x_2 + b$$

$$y = a_3 x_3 + b$$

$$y = a_1 x_1 + a_2 x_2 + b$$

$$y = a_1 x_1 + a_3 x_3 + b$$

Works the same way.

1. Define the model

Different models = different assumptions

$$y = a_1 x_1 + b$$

$$y = a_2 x_2 + b$$

$$y = a_3 x_3 + b$$

$$y = a_1 x_1 + a_2 x_2 + b$$

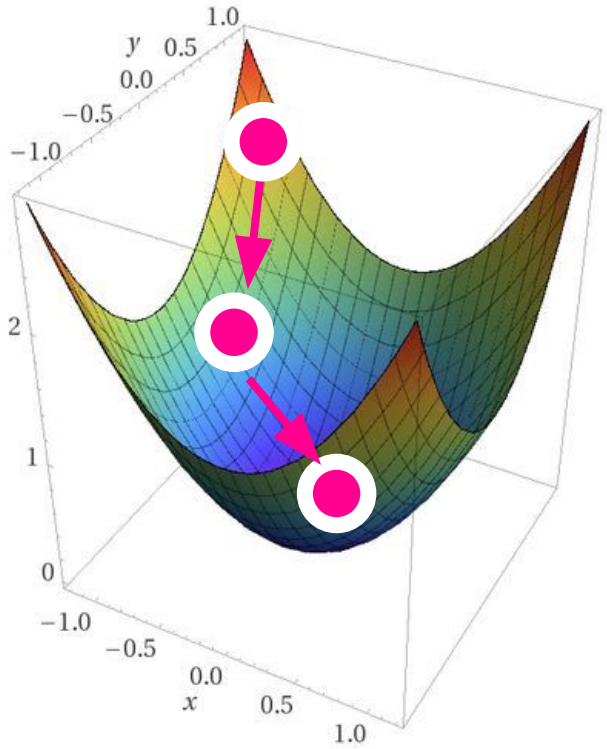
$$y = a_1 x_1 + a_3 x_3 + b$$

$$y = a_1 x_1 + a_2 x_2 + a_3 x_3 + b$$

Works the same way.

1. Define the model

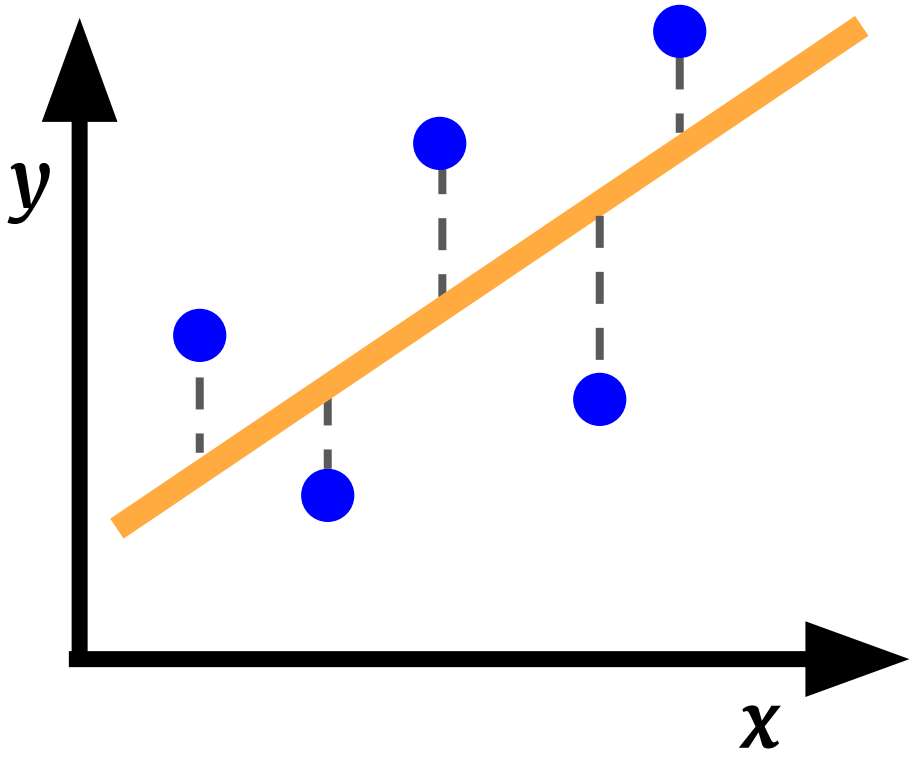
Different models = different assumptions



Works the same way.

1. Define the model
2. Fit the parameters of the model using the training dataset

Different models = different assumptions



Works the same way.

1. Define the model
2. Fit the parameters of the model using the training dataset
3. Calculate R^2 using the testing dataset

Example of result:

$$y = a_1x_1 + a_2x_2 + a_3x_3 + b$$

y = price
x_1 = sqft
x_2 = number of rooms
x_3 = number of windows

Example of result:

y = price
x_1 = sqft
x_2 = number of rooms
x_3 = number of windows

$$y = a_1x_1 + a_2x_2 + a_3x_3 + b$$

$$y = 0.65\,x_1 + 0.12\,x_2 + 0.01\,x_3 + 11.3$$

Example of result:

y = price
x_1 = sqft
x_2 = number of rooms
x_3 = number of windows

$$y = a_1x_1 + a_2x_2 + a_3x_3 + b$$

$$y = 0.65\,x_1 + 0.12\,x_2 + 0.01\,x_3 + 11.3$$