Aprendizado de Máquina e Reconhecimento de Padrões 2021.2

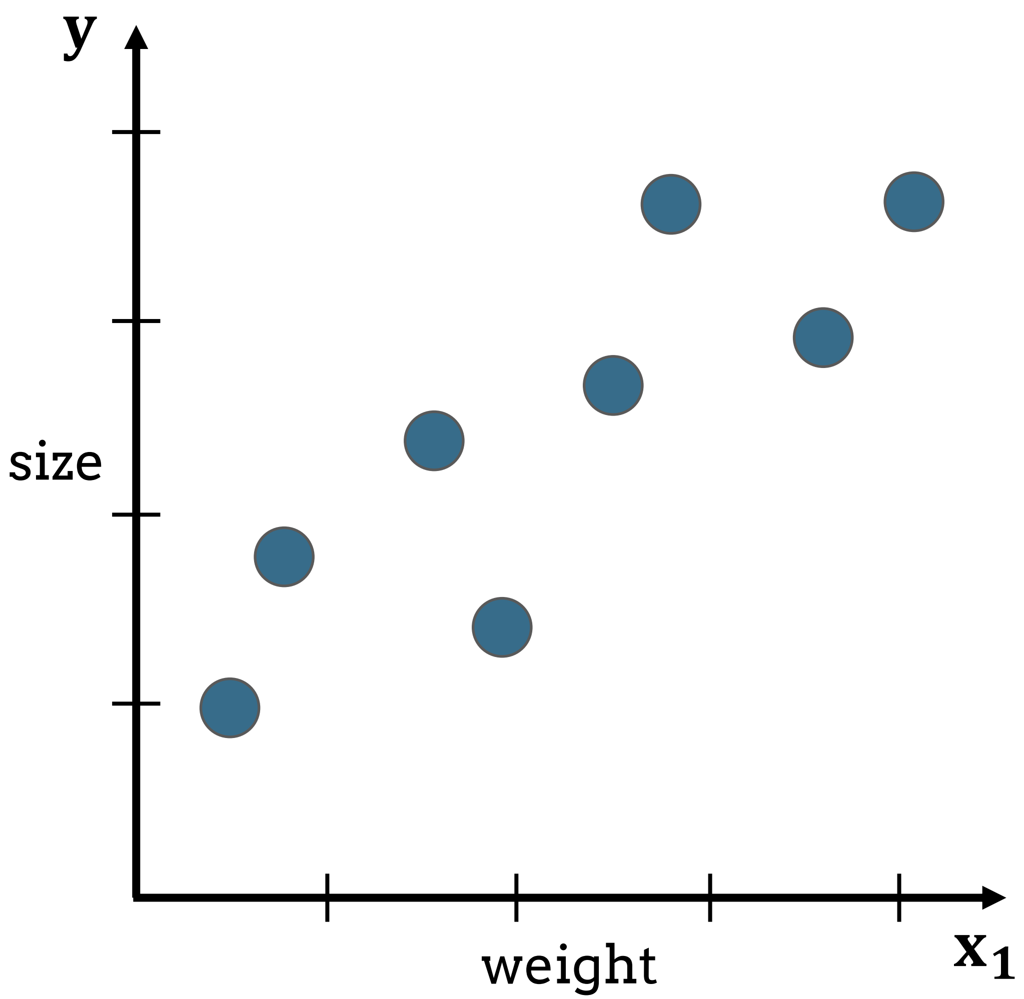
Regularization

Strongly based on videos from StatsQuest

Prof. Dr. Samuel Martins (Samuka)
samuel.martins@ifsp.edu.br

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
São Paulo
Campus Campinas

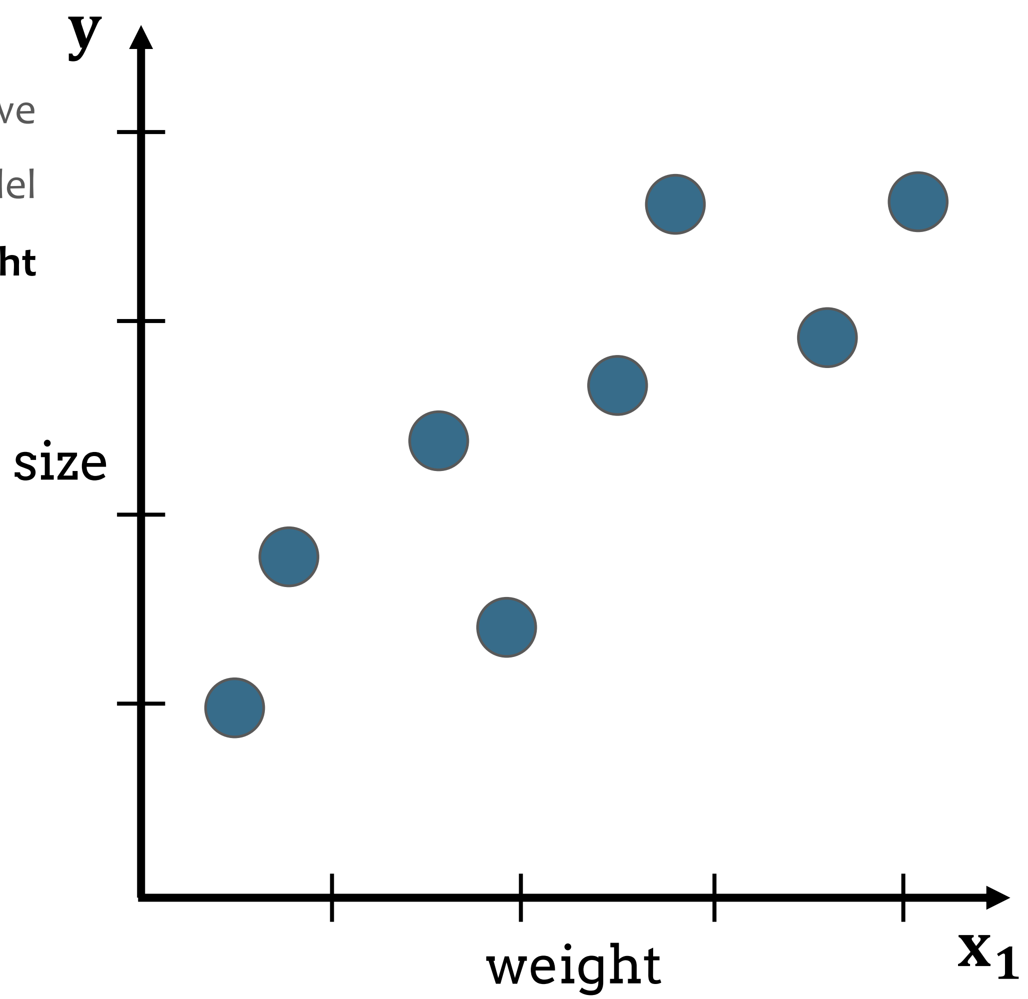
Suppose we have collected data from mice.



Suppose we have collected data from mice.

Since it looks **relatively linear**, we will use **Linear Regression** to model the relationship between **weight** ($\mathbf{x_1}$) and **size** ($\mathbf{y}$).

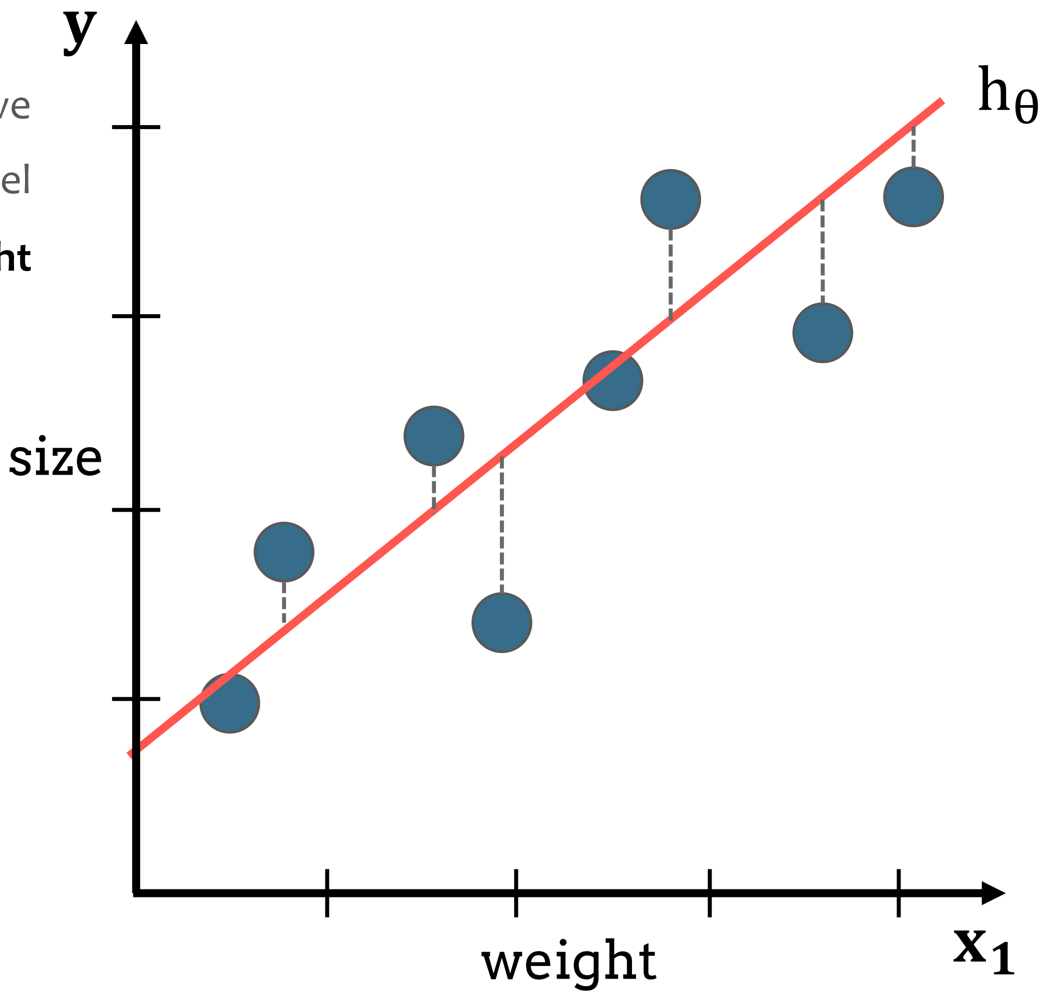
$$\hat{y} = h_\theta(\mathbf{x}) = \theta_0 + \theta_1 x_1$$



Suppose we have collected data from mice.

Since it looks **relatively linear**, we will use **Linear Regression** to model the relationship between **weight** ($\mathbf{x_1}$) and **size** ($\mathbf{y}$).

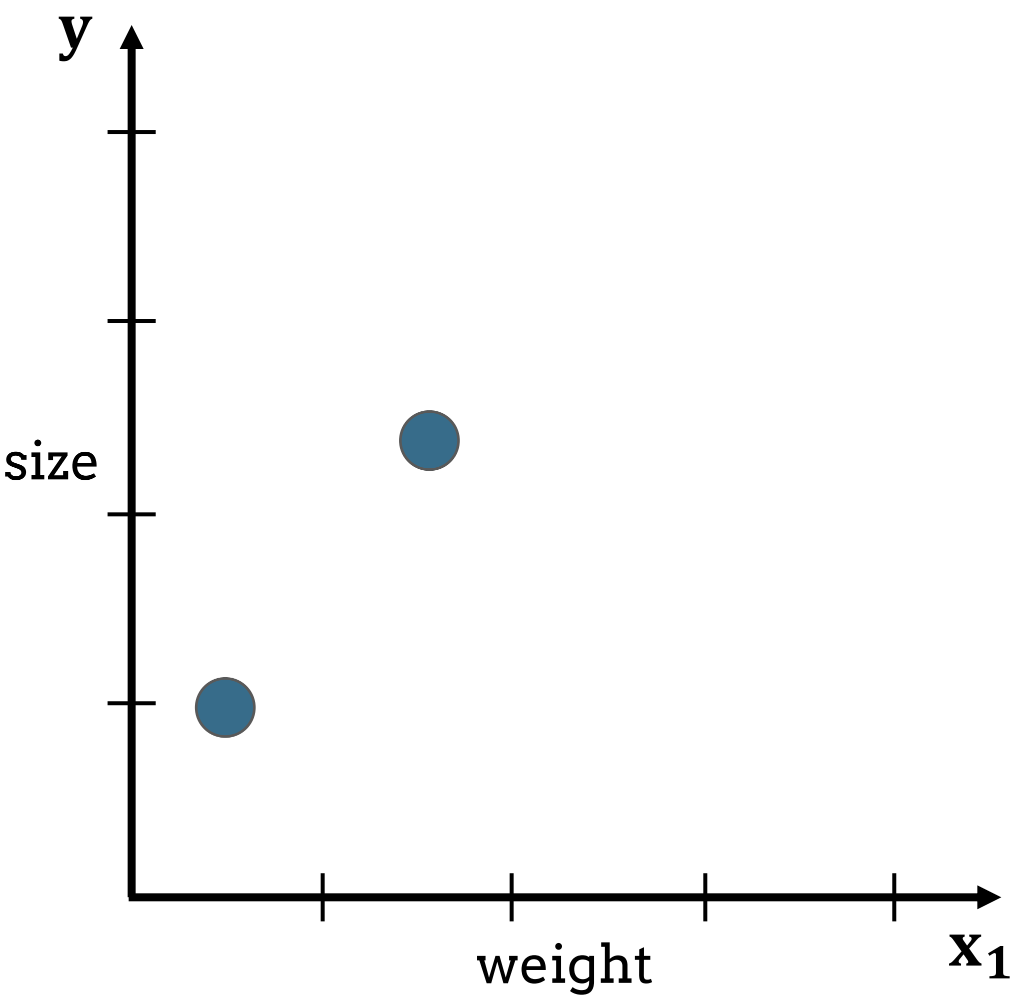
$$\hat{y} = h_\theta(\mathbf{x}) = \theta_0 + \theta_1 x_1$$



$$h_\theta(\mathbf{x}) = 0.9 + 0.75 x_1$$

Now, consider we have just a few training instances

(*e.g.*, two training instances).

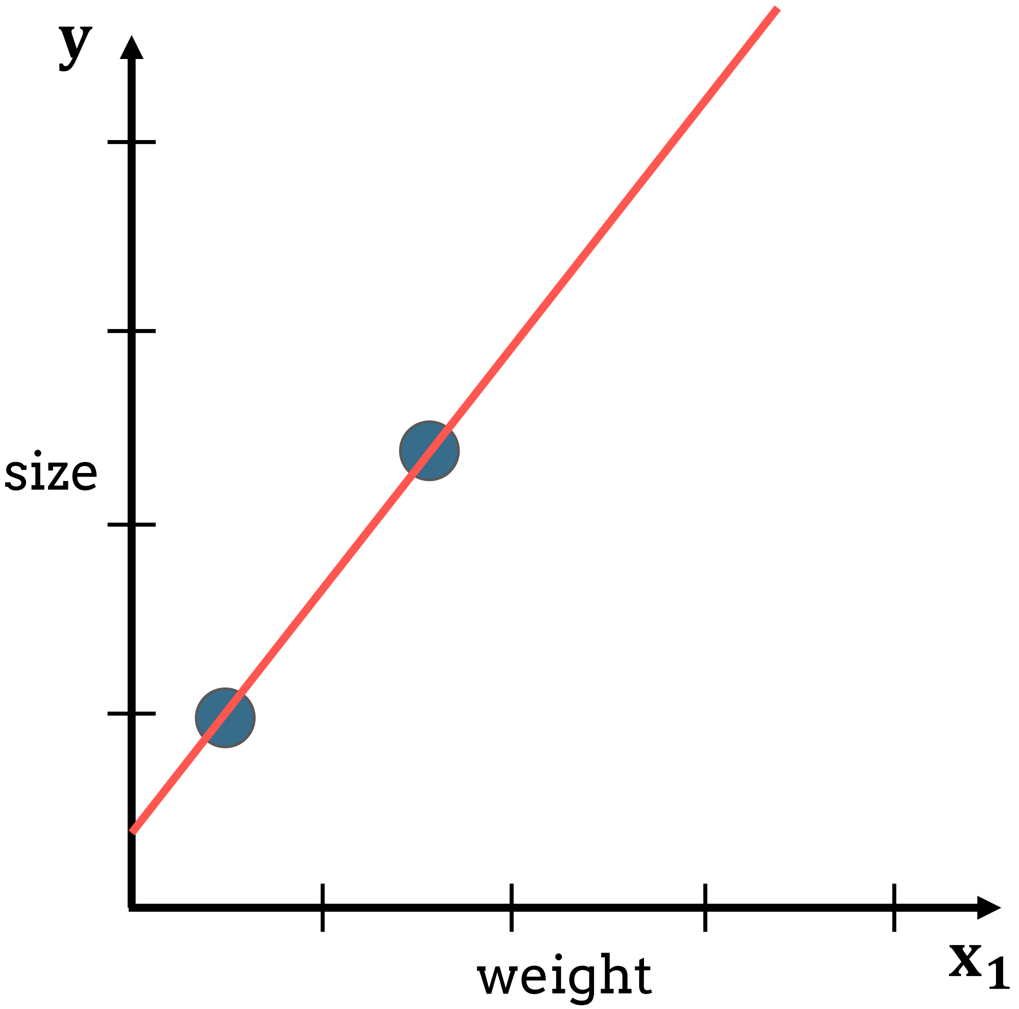


Now, consider we have just a few training instances

(*e.g.*, two training instances).

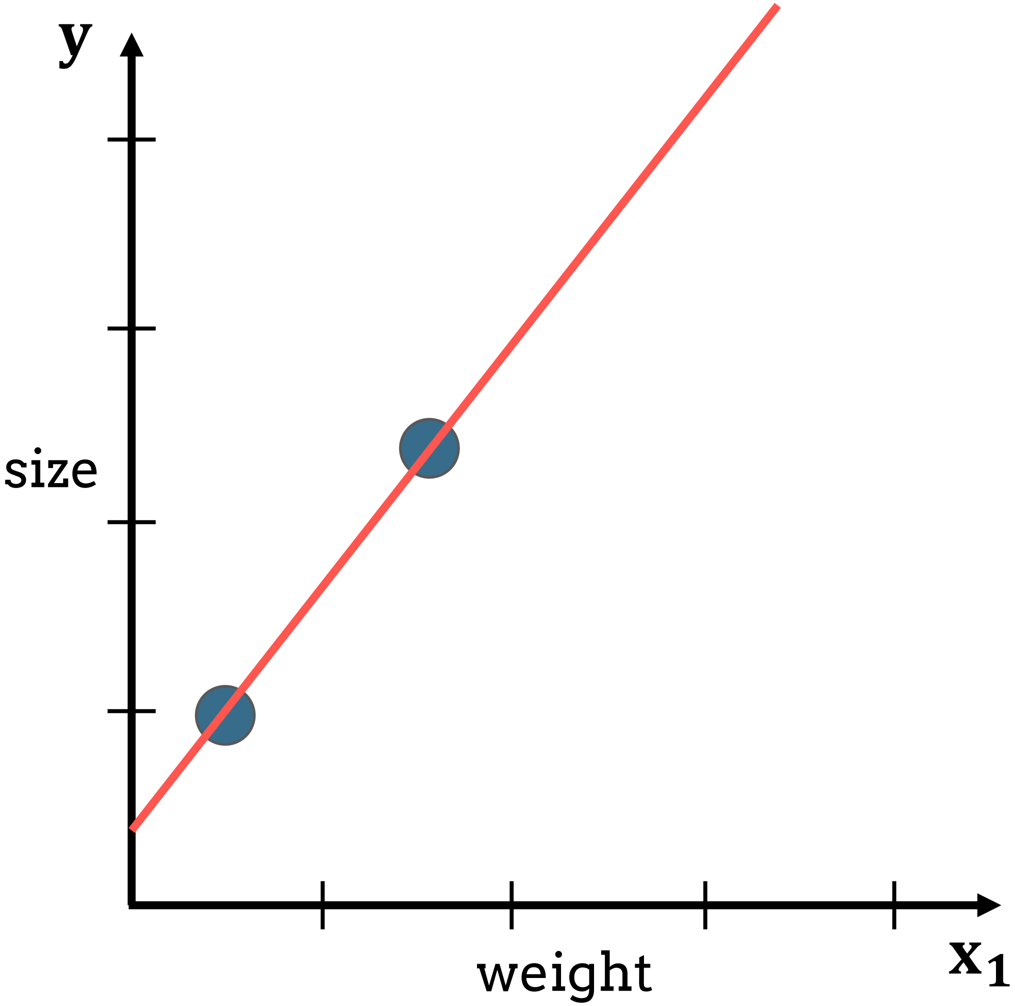
Our **Linear Regression** model exactly

overlaps the two training instances.



Now, consider we have just a few training instances

(*e.g.*, two training instances).



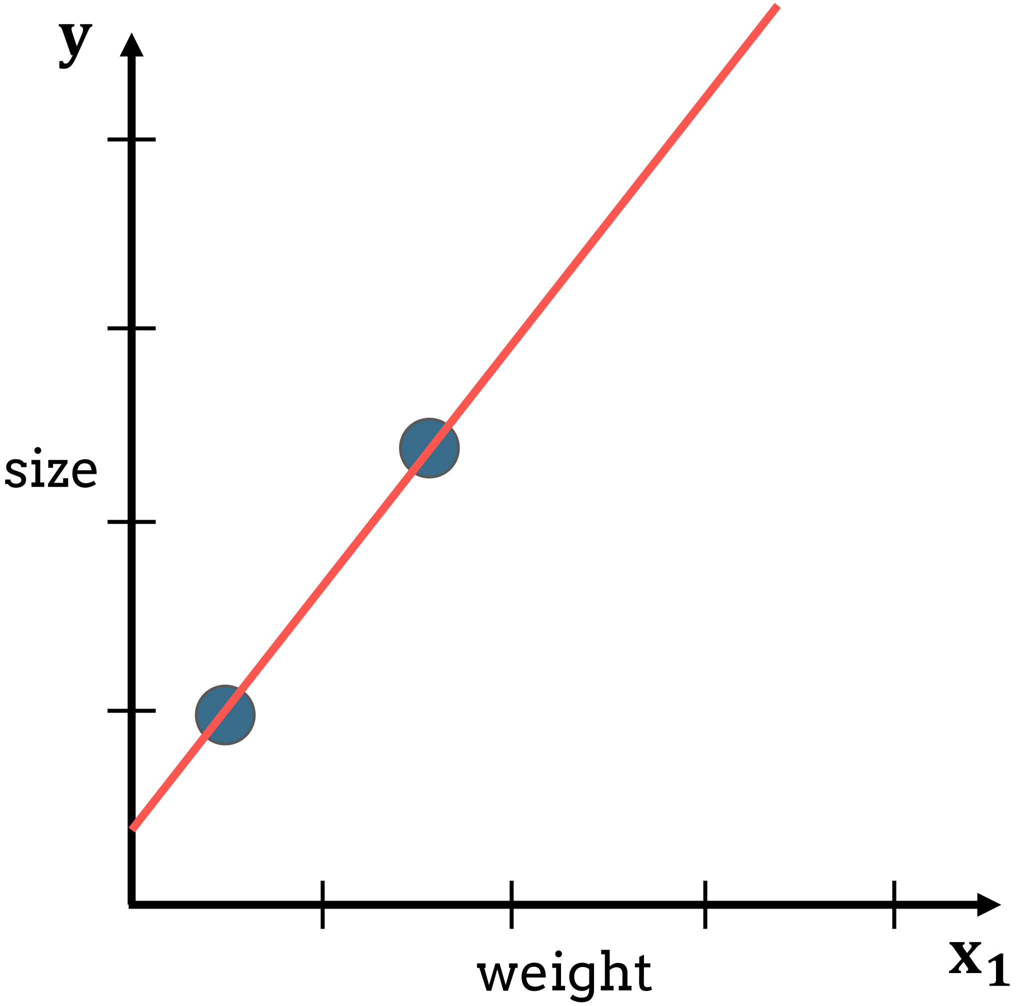
Our **Linear Regression** model exactly *overlaps* the two training instances.

↓

Sum of training errors/residuals (e.g., MSE) = **0**

Now, consider we have just a few training instances

(*e.g.*, two training instances).



Our **Linear Regression** model exactly

overlaps the two training instances.

↓

Sum of training errors/residuals

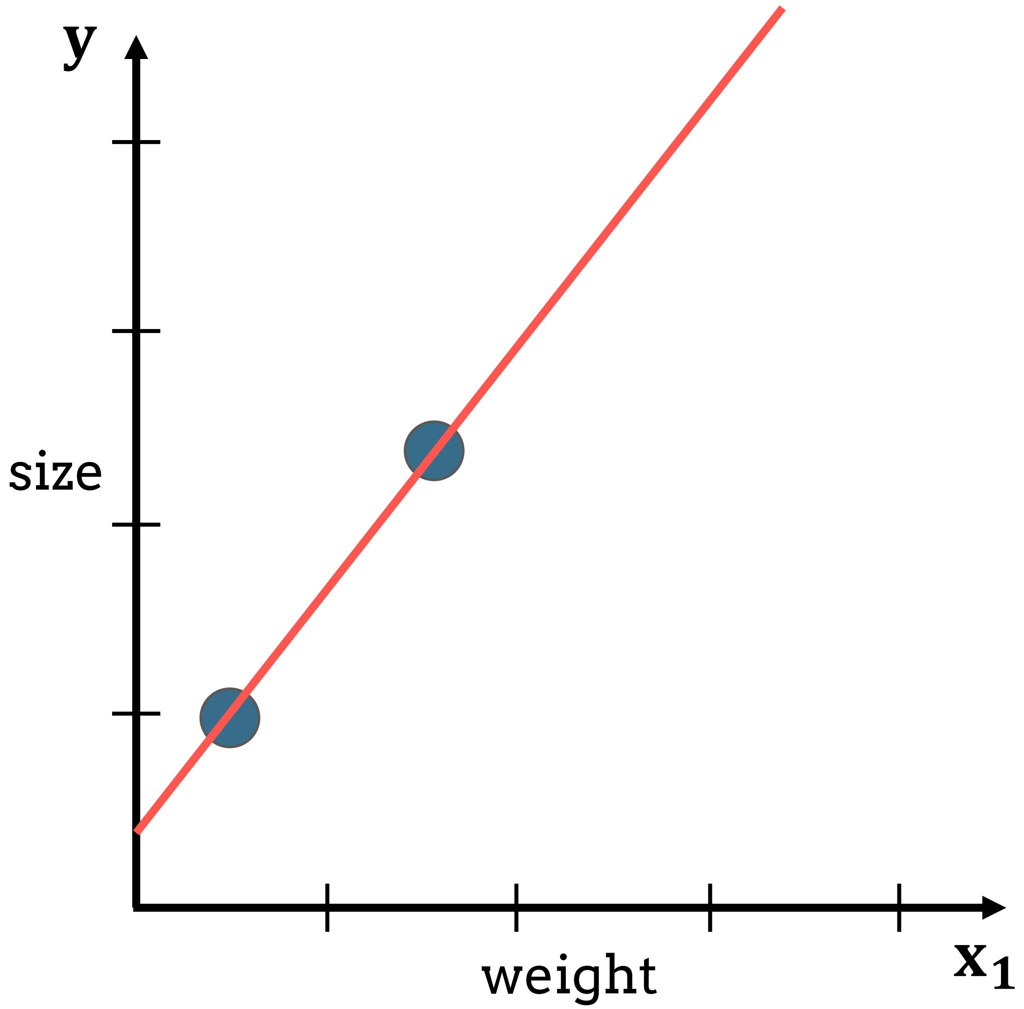
(e.g., MSE) = **0**

↓

(very) low bias

Now, consider we have just a few training instances

(*e.g.*, two training instances).



Our **Linear Regression** model exactly

overlaps the two training instances.

↓

Sum of training errors/residuals

(e.g., MSE) = **0**

↓

(very) low bias

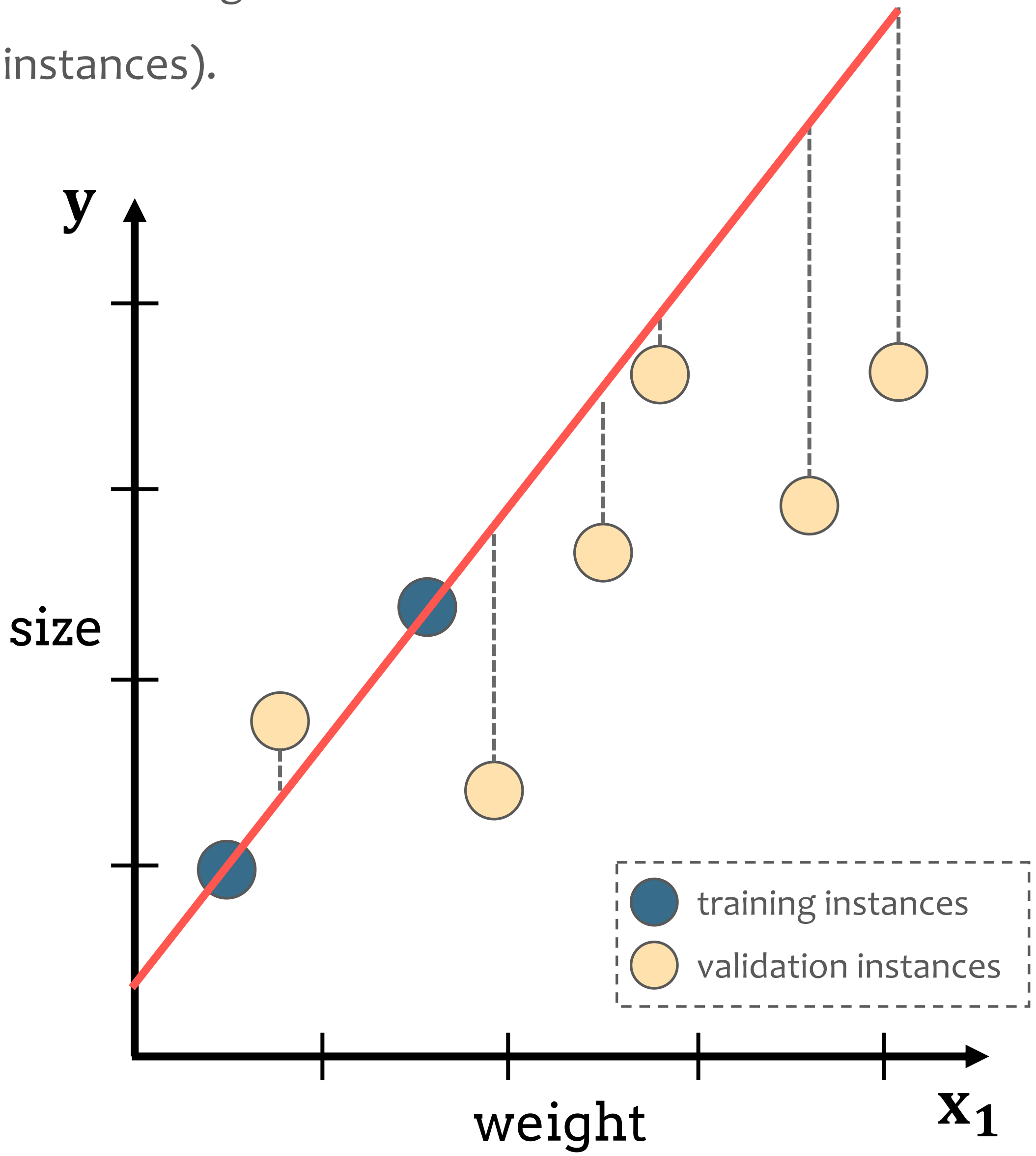
↓

possible indicate of

overfitting

Now, consider we have just a few training instances

(*e.g.*, two training instances).

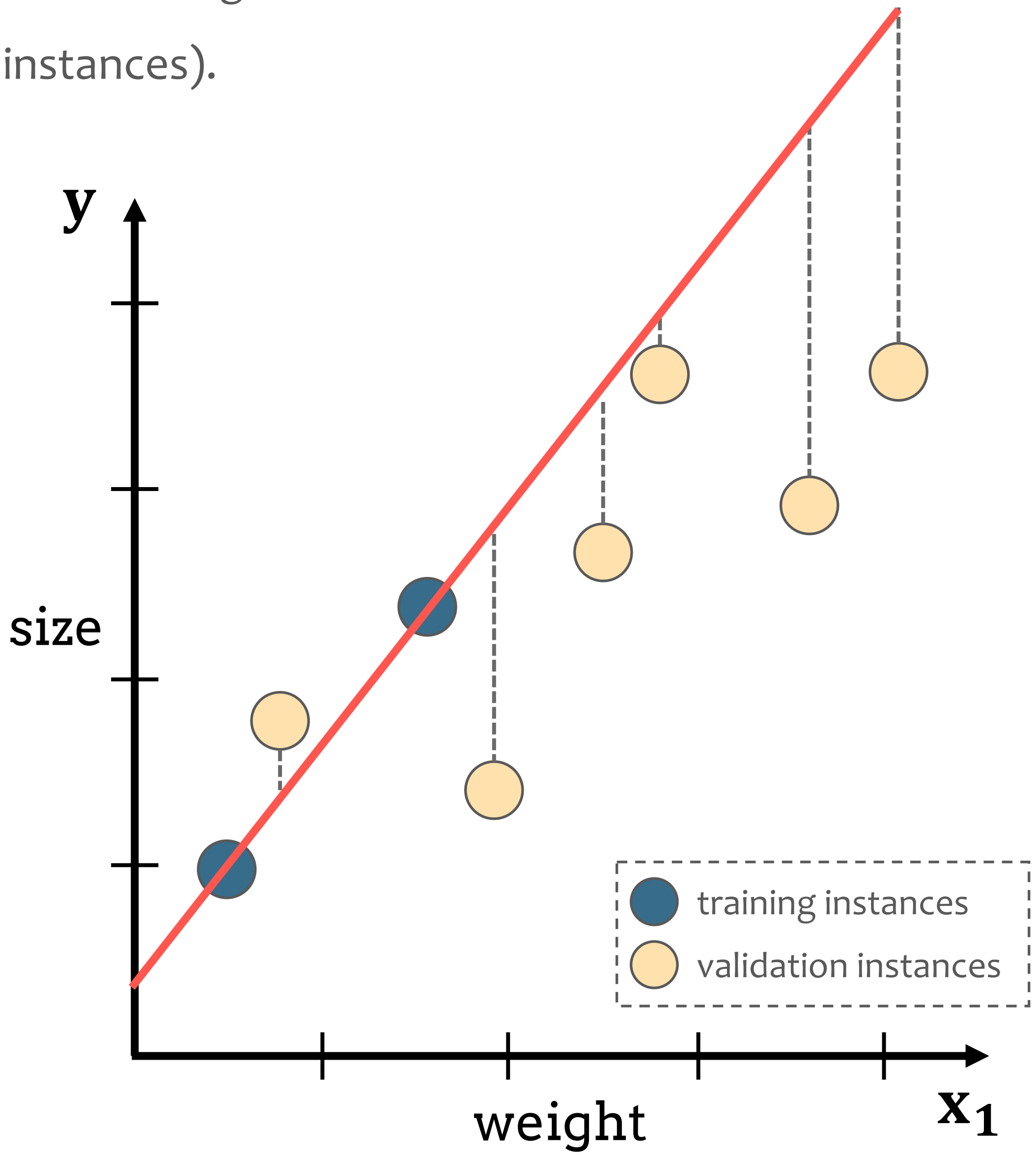


Now, consider we have just a few training instances

(*e.g.*, two training instances).

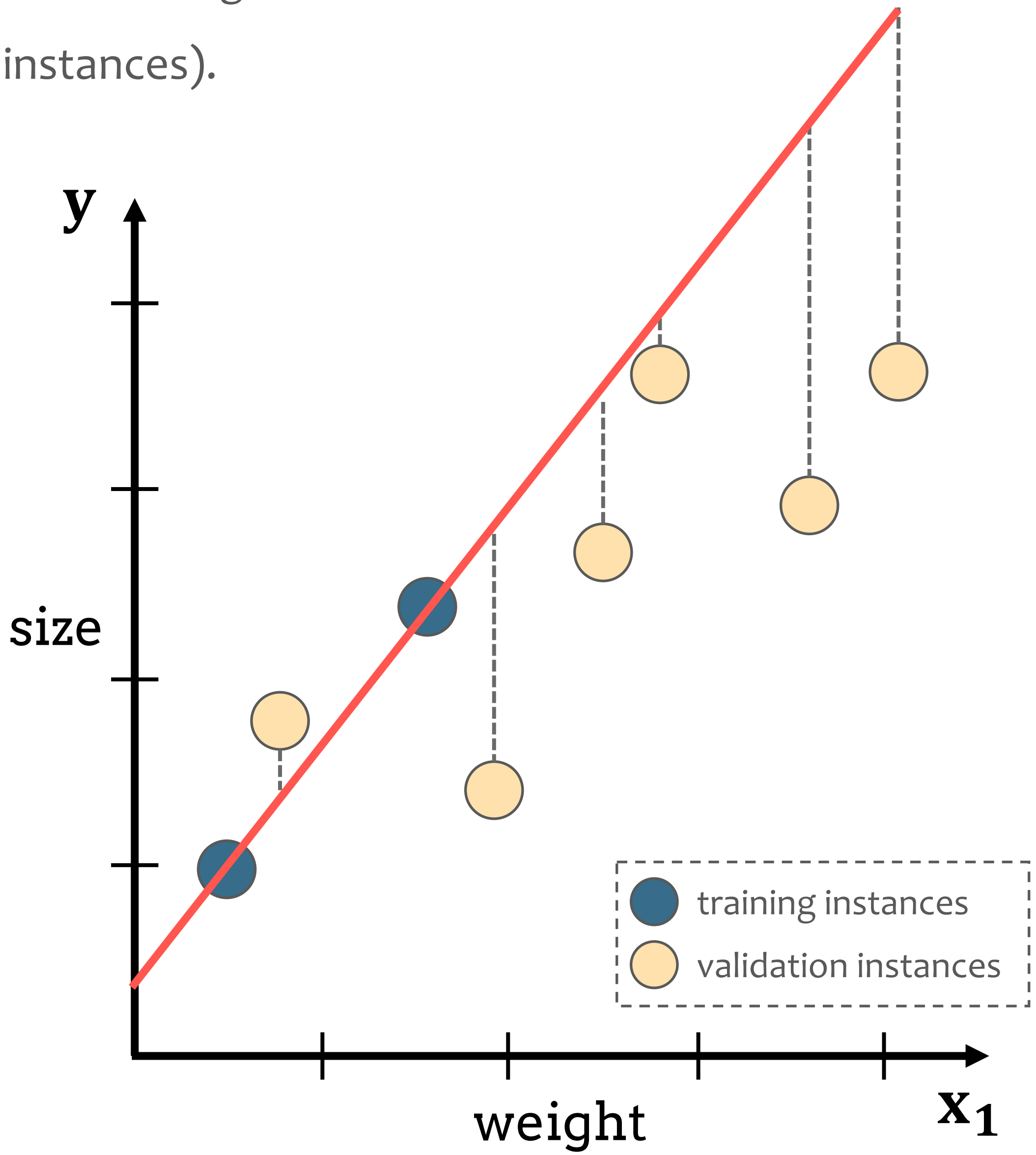
Sum of validation errors/residuals

(e.g., MSE) is **high**



Now, consider we have just a few training instances

(*e.g.*, two training instances).



Sum of validation errors/residuals

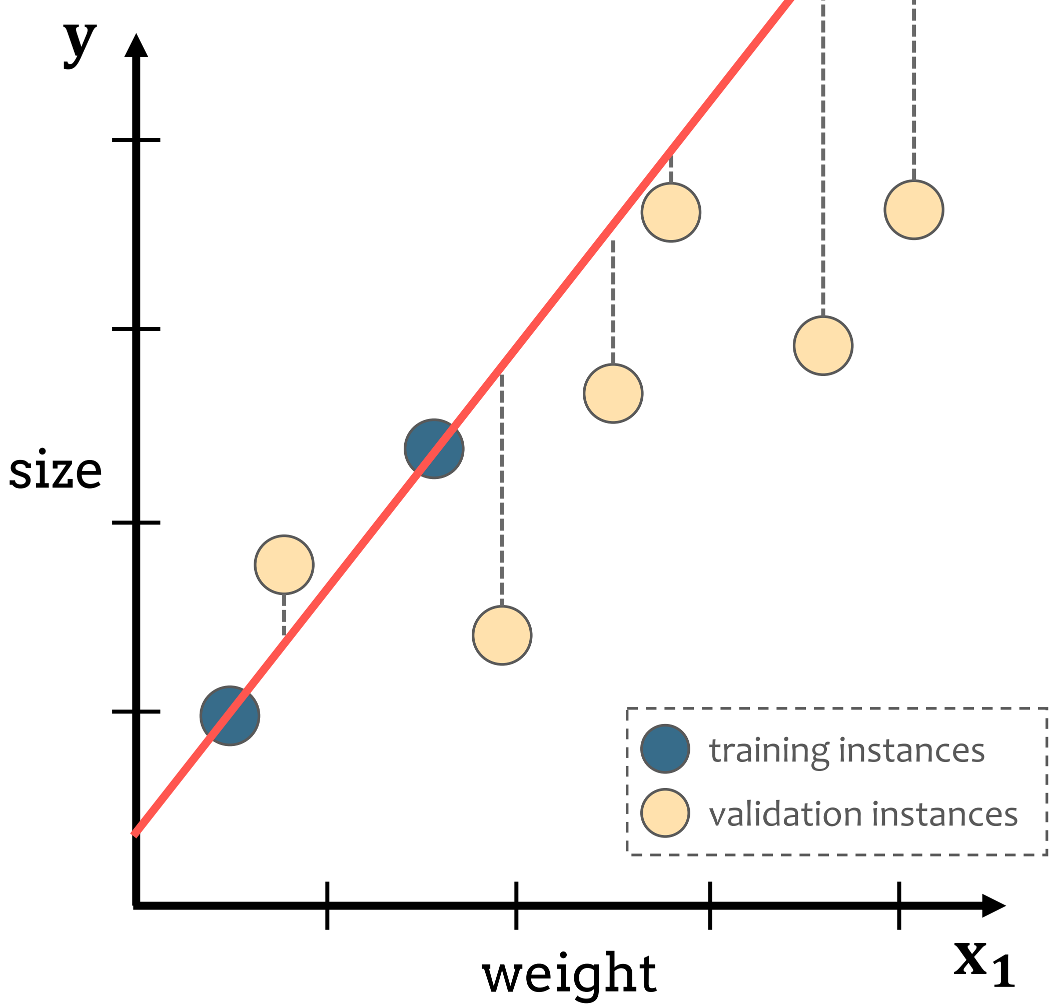
(e.g., MSE) is **high**

↓

high variance

Now, consider we have just a few training instances

(*e.g.*, two training instances).



Sum of validation errors/residuals

(e.g., MSE) is **high**

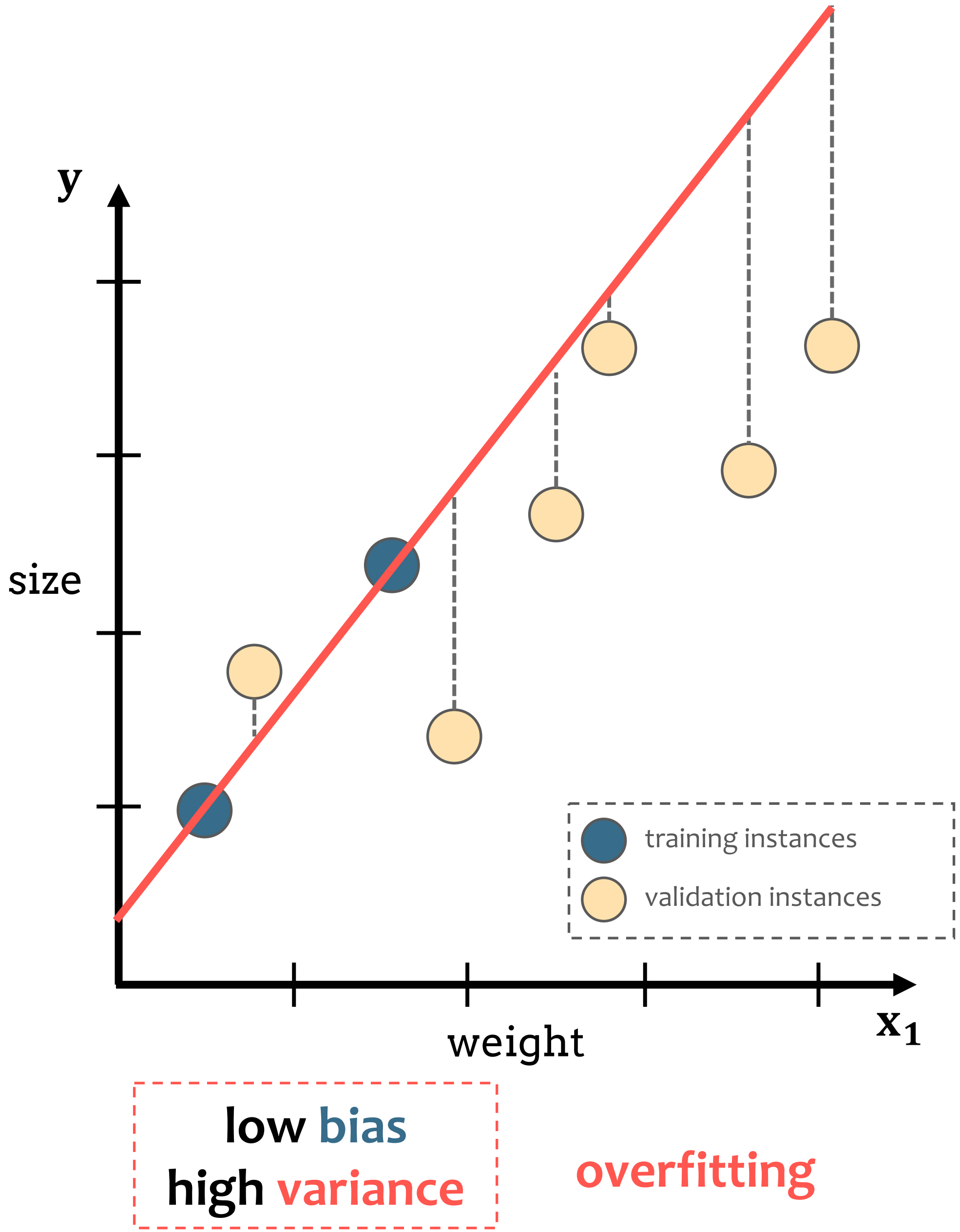
↓

high variance

low bias
high variance

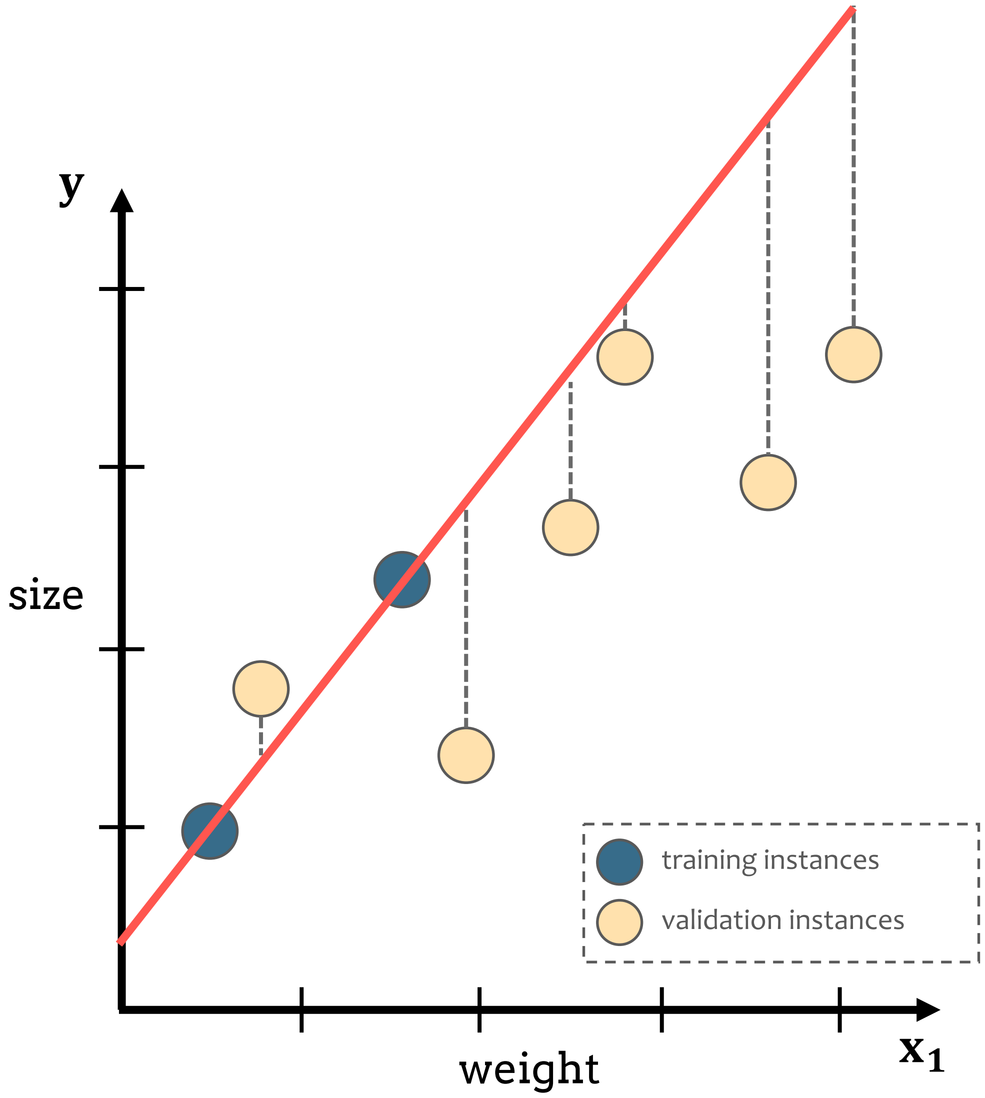
overfitting

We could find a way **to prevent** that **<u>does not fit</u>** the **training data** too much.



We could find a way **to prevent** that **<u>does not fit</u>** the **training data** too much.

In other words, we could introduce **a small amount of** bias into how the model is fit to the **training data**.



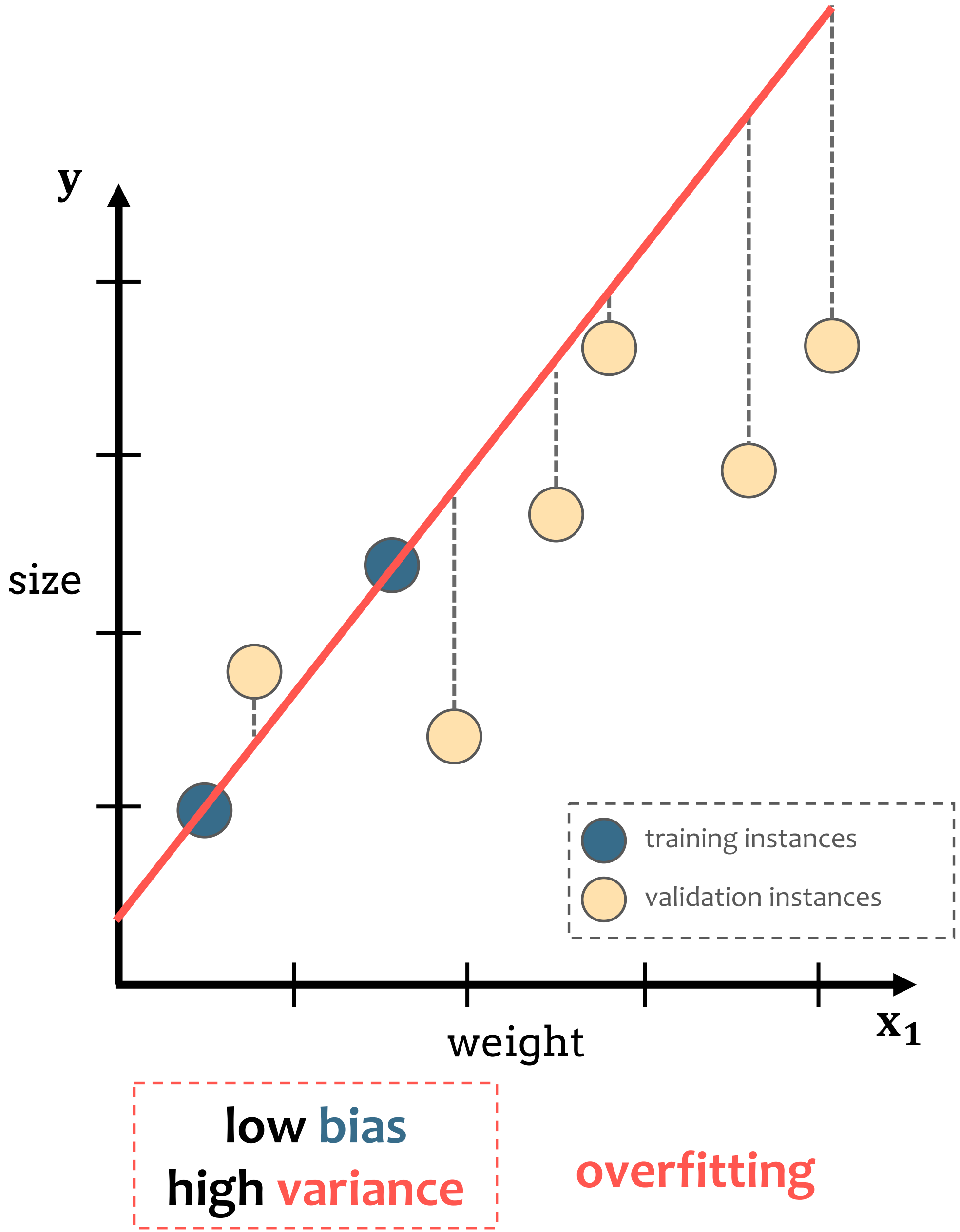
low bias
high variance

overfitting

We could find a way **to prevent** that **does not fit** the **training data** too much.

In other words, we could introduce **a small amount of** bias into how the model is fit to the **training data**.

Consequently, we would **decrease** the variance thus **avoiding** overfitting.



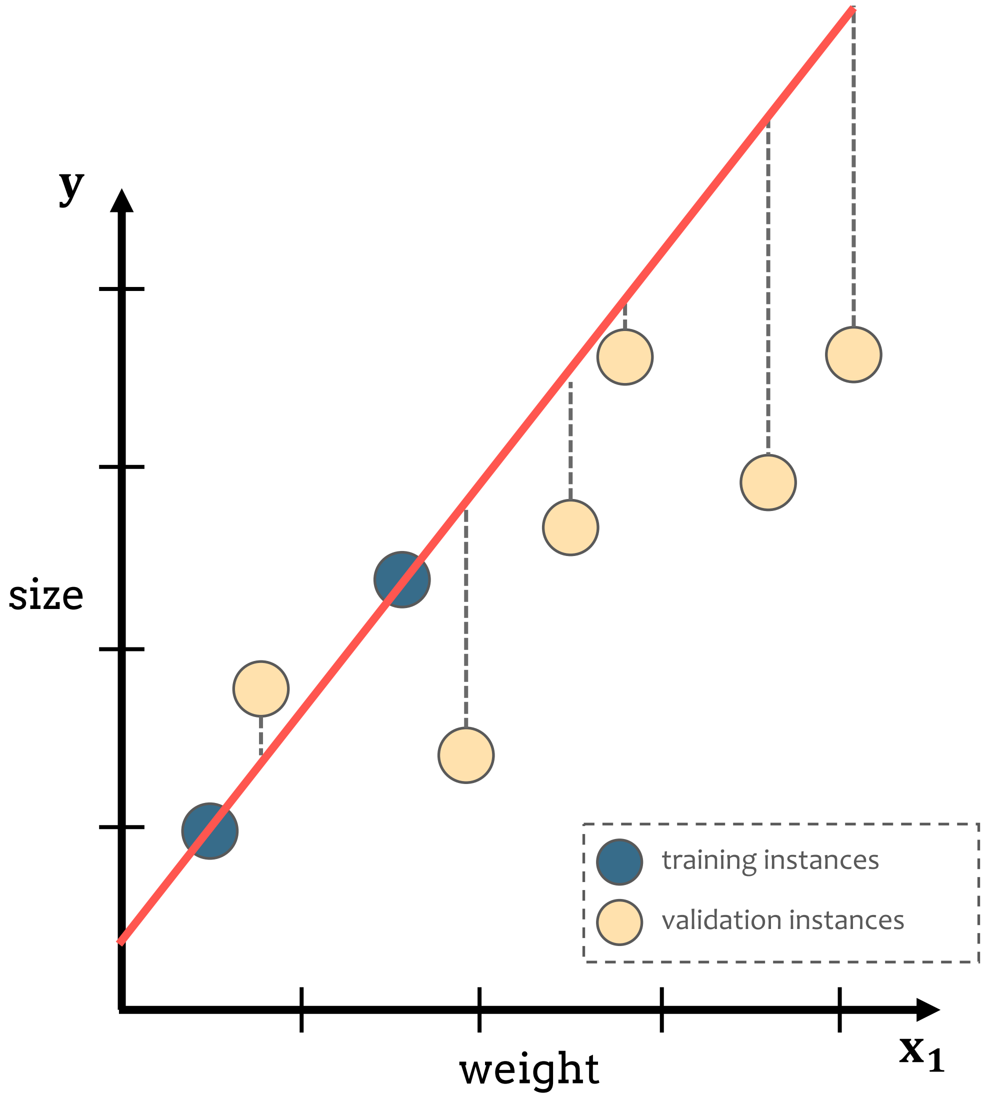
We could find a way **to prevent** that **does not fit** the **training data** too much.

In other words, we could introduce **a small amount of** bias into how the model is fit to the **training data**.

Consequently, we would **decrease** the variance thus **avoiding** overfitting.

Bias-Variance trade-off

- Increasing ↑ variance reduce ↓ bias, and vice versa.
- Reducing ↓ variance increase ↑ bias, and vice versa.



low bias
high variance

overfitting

We could find a way **to prevent** that **<u>does not fit</u>** the **training data** too much.

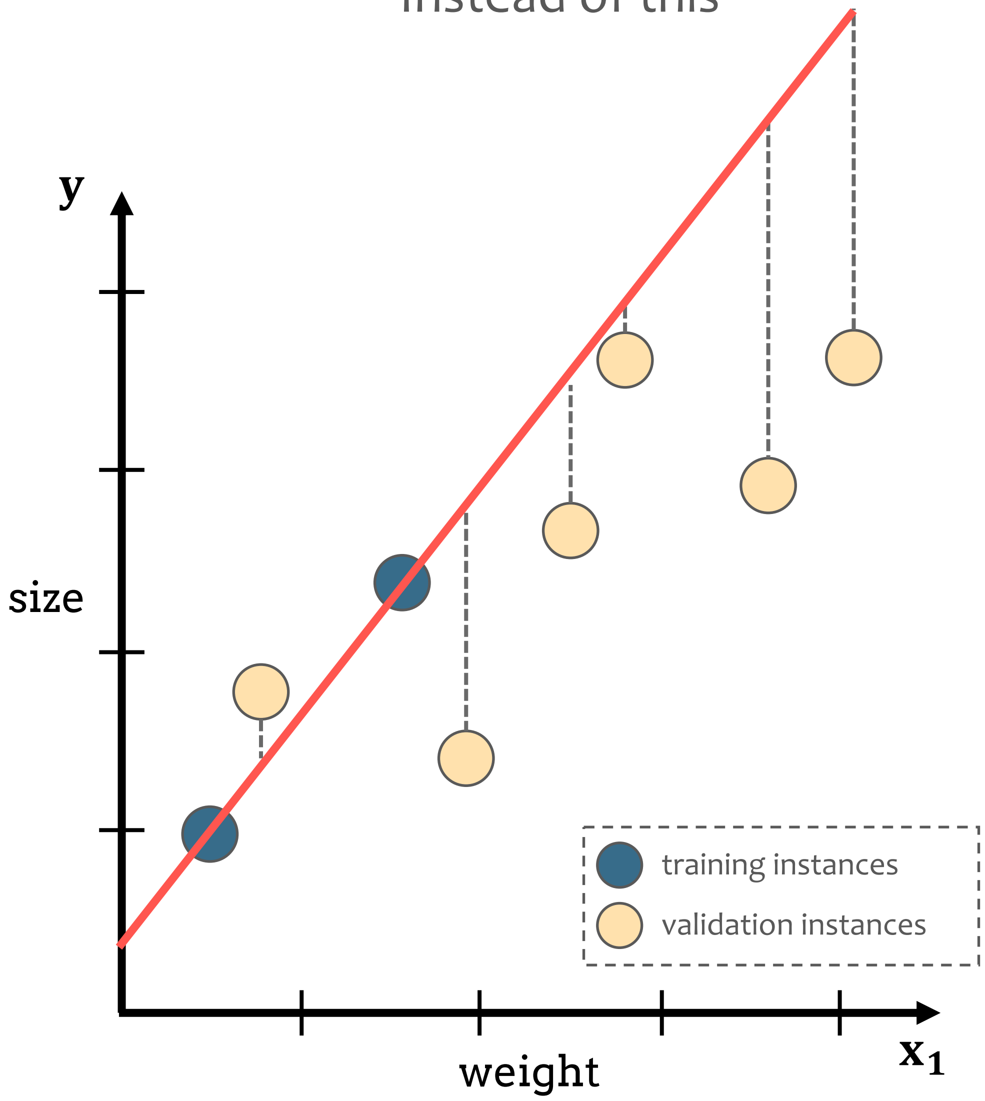
In other words, we could introduce **a small amount of bias** into how the model is fit to the **training data**.

Consequently, we would **decrease** the variance thus **avoiding** overfitting.

Bias-Variance trade-off

- Increasing ↑ variance reduce ↓ **bias**, and vice versa.
- Reducing ↓ variance increase ↑ **bias**, and vice versa.

Instead of this



low bias
high variance

overfitting

We could find a way **to prevent** that **<u>does not fit</u>** the **training data** too much.

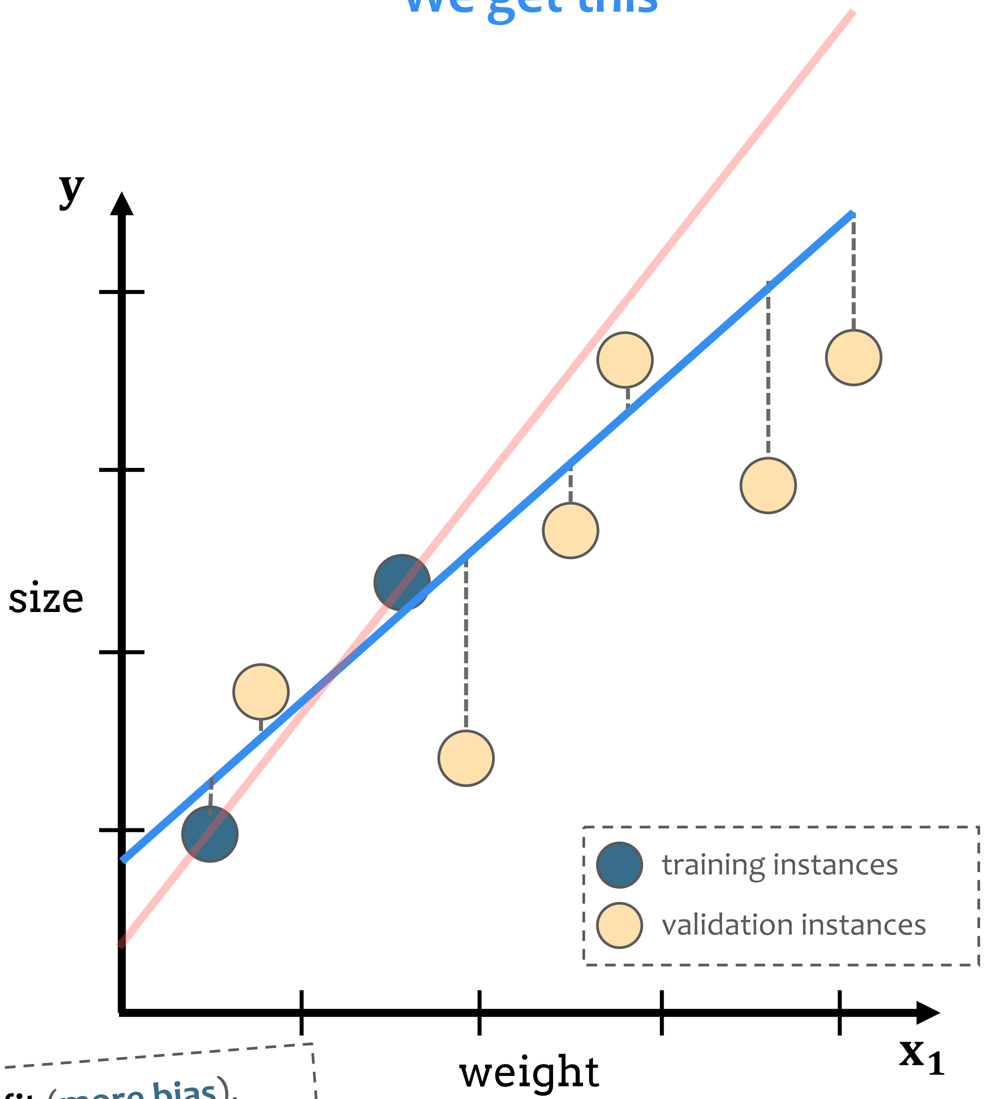
In other words, we could introduce **a small amount of** bias into how the model is fit to the **training data**.

Consequently, we would **decrease** the variance thus **avoiding** overfitting.

Bias-Variance trade-off

- Increasing ↑ variance reduce ↓ **bias**, and vice versa.
- Reducing ↓ variance increase ↑ **bias**, and vice versa.

We get this



From a slightly **worse fit** (more **bias**), we got a **significant drop** ↓ in variance.

We could find a way **to prevent** that **does not fit** the **training data** too much.

In other words, we could introduce **a small amount of bias** into how the model is fit to the **training data**.

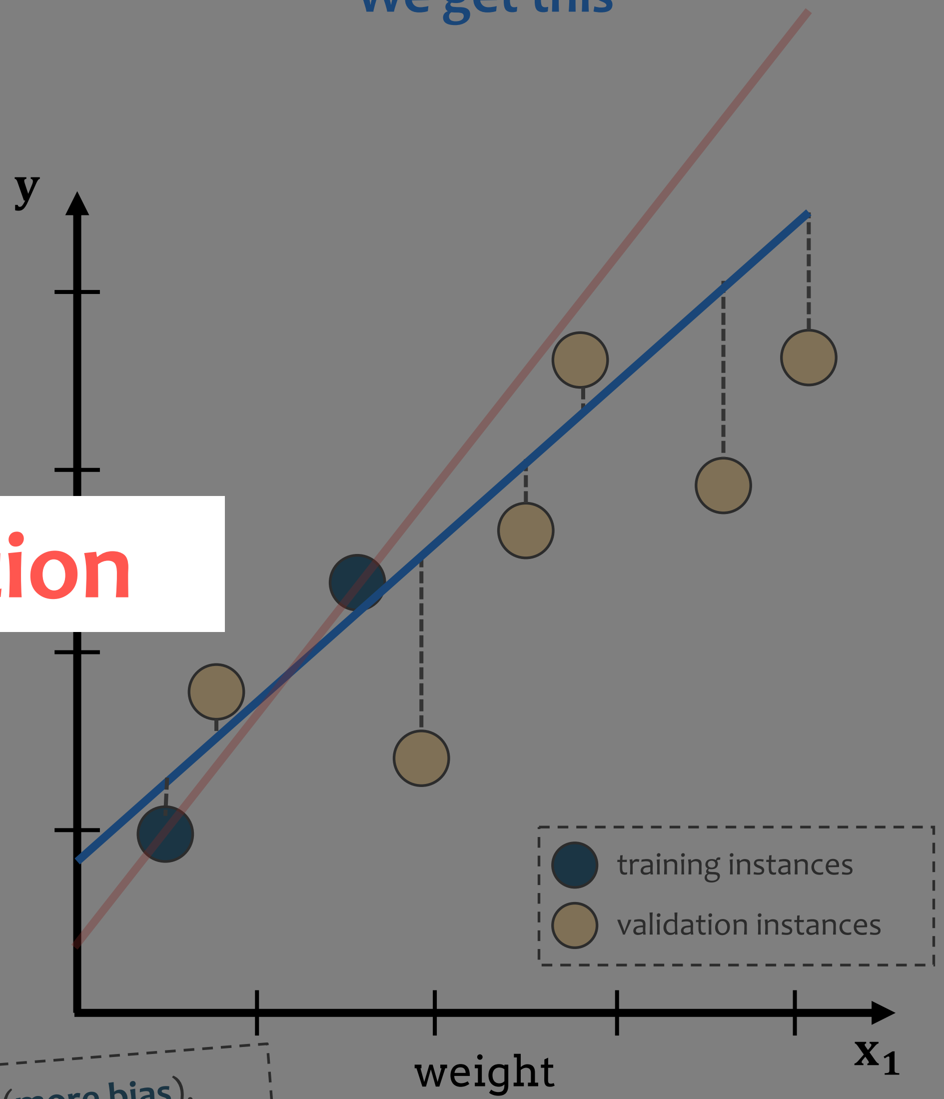
Consequently, we would **decrease** the **variance** thus **avoiding overfitting.**

Bias-Variance trade-off

- Increasing ↑ **variance** reduce ↓ **bias**, and vice versa.
- Reducing ↓ **variance** increase ↑ **bias**, and vice versa.

Regularization

We get this



From a slightly **worse fit** (more bias), we got a **significant drop** ↓ in **variance**.

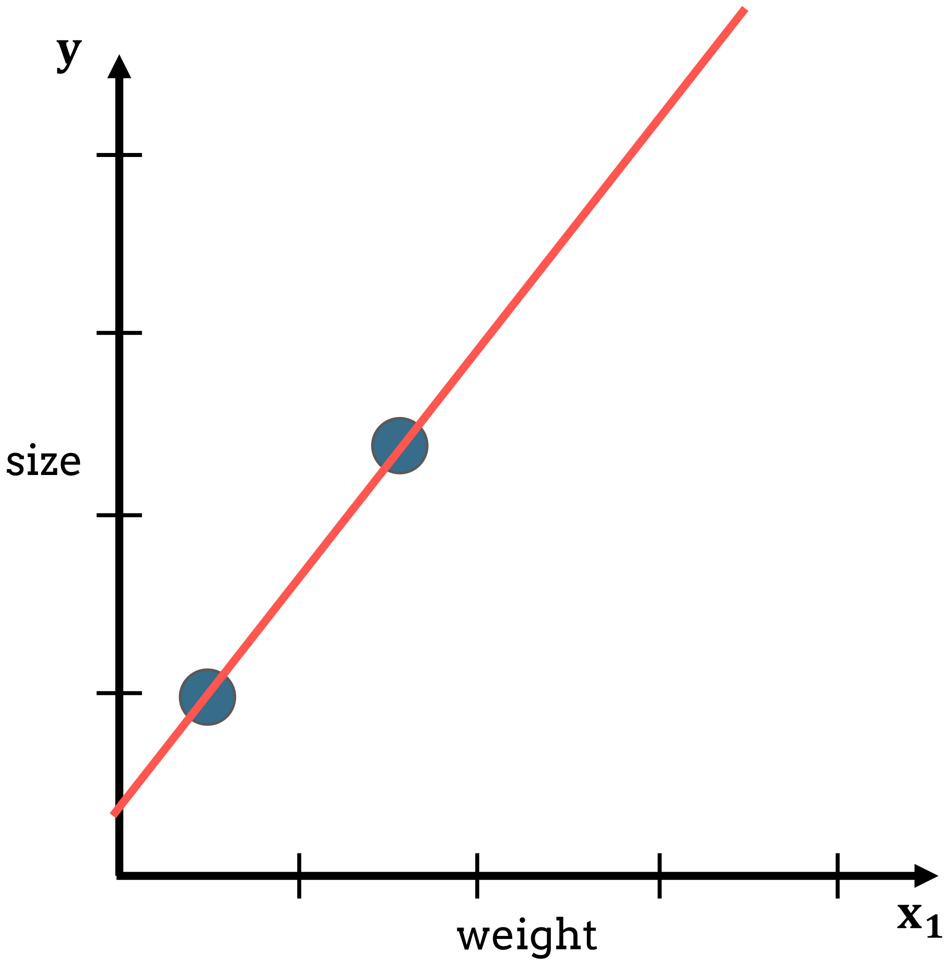
Regularization in ML

- Technique that **prevents** the model from **overfitting** by **adding extra information** to it.
- In Regression, it is a form of regression that shrinks the coefficient estimates towards zero.

Common regularization for Regression:

- Ridge regression
- Lasso Regression
- Elastic Net

Ridge Regression



Linear Regression

$$\hat{y} = h_\theta(\mathbf{x}) = \underbrace{\theta^{\boldsymbol{T}} \cdot \mathbf{x}}$$

n features

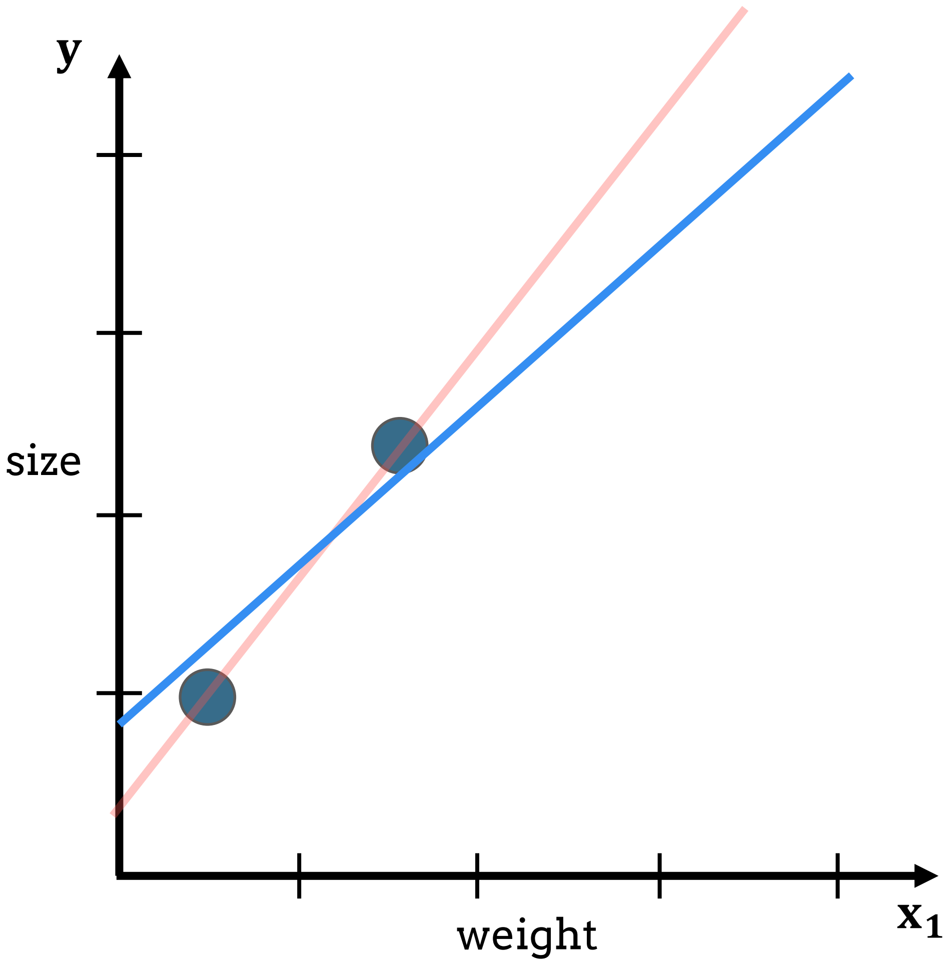
$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

for the case

$\theta_0 + \theta_1 x_1$

Goal: Find θ that **minimizes** the cost function $J(\theta)$

$$J(\theta) = \mathrm{MSE}(\mathbf{X}, h_\theta) = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$



Ridge Regression (L2 Regularization)

$$\hat{y} = h_\theta(\mathbf{x}) = \underbrace{\theta^{\boldsymbol{T}} \cdot \mathbf{x}}$$

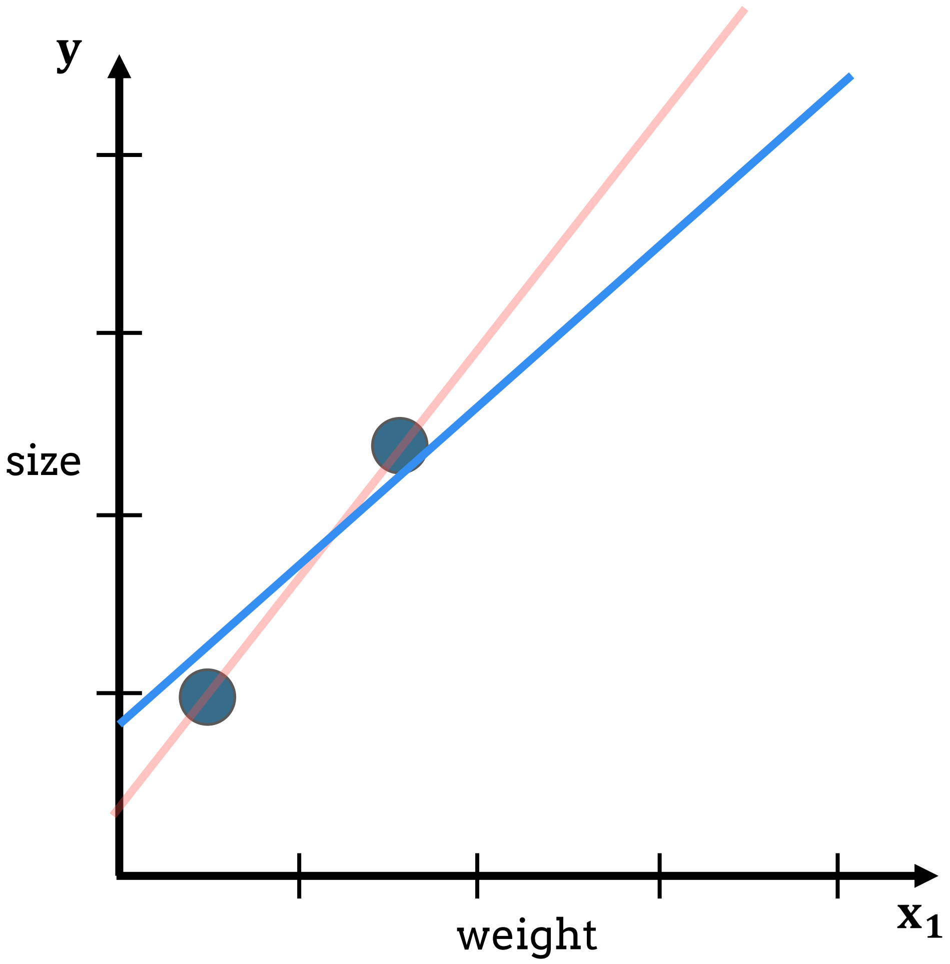
n features

$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

for the case

$\theta_0 + \theta_1 x_1$

Goal: Find θ that **minimizes** the cost function $J(\theta)$



Ridge Regression (L2 Regularization)

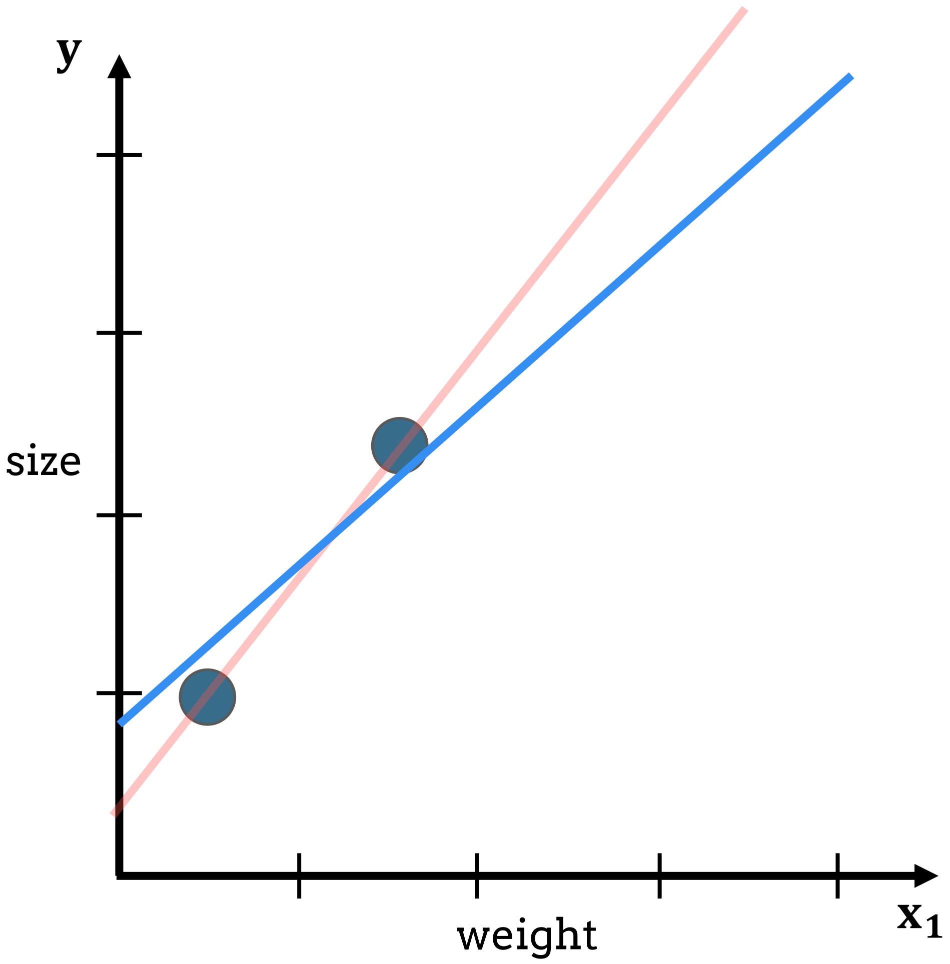
$$\hat{y} = h_\theta(\mathbf{x}) = \underbrace{\theta^T \cdot \mathbf{x}}$$

n features: $\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

for the case: $\theta_0 + \theta_1 x_1$

Goal: Find θ that **minimizes** the cost function $J(\theta)$

$$J(\theta) = \mathrm{MSE}(\mathbf{X}, h_\theta) + \underbrace{\alpha \sum_{i=1}^{n} \theta^2}_{\text{regularization term penalty}}$$



Ridge Regression (L2 Regularization)

n features

$$\hat{y} = h_\theta(\mathbf{x}) = \theta^{T} \cdot \mathbf{x}$$

$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

for the case

$\theta_0 + \theta_1 x_1$

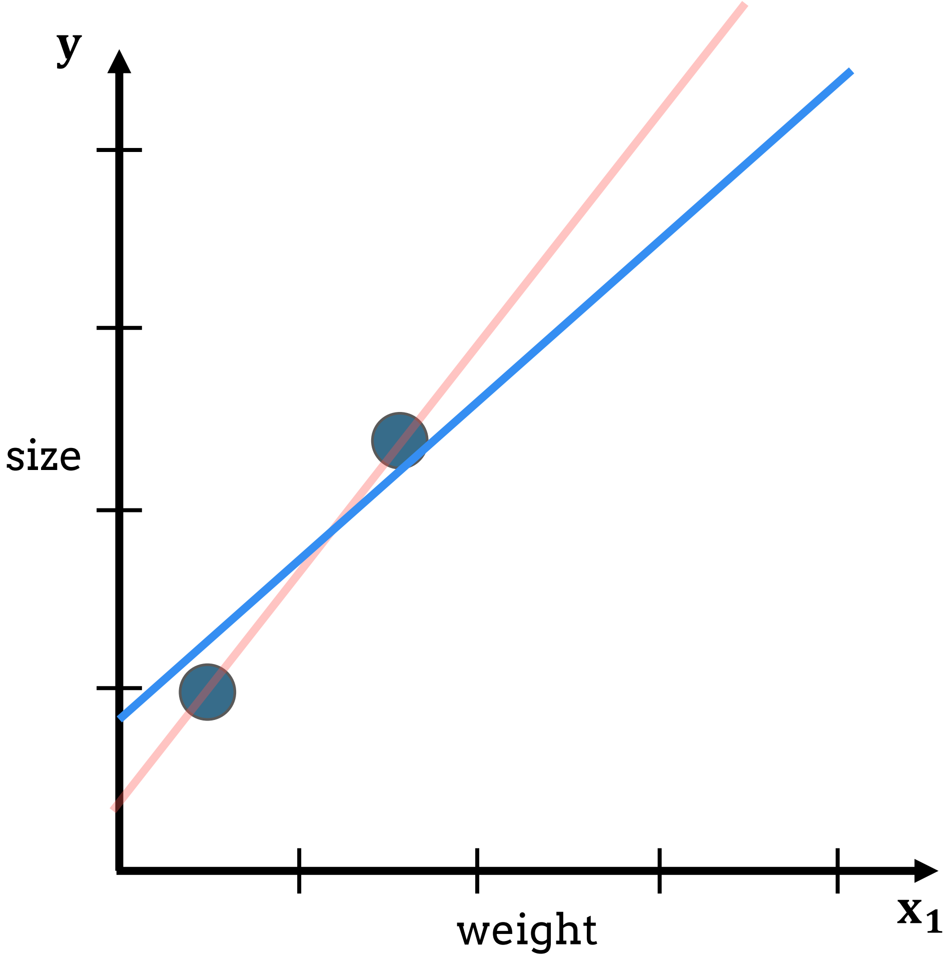
Goal: Find θ that **minimizes** the cost function $J(\theta)$

The intercept θ_0 is **not regularized**.

$$J(\theta) = \text{MSE}(\mathbf{X}, h_\theta) + \underbrace{\alpha \sum_{i=1}^{n} \theta^2}_{\text{regularization term penalty}}$$

for the case

$\alpha(\theta_1^2)$



Ridge Regression (L2 Regularization)

n features

$$\hat{y} = h_\theta(\mathbf{x}) = \underbrace{\theta^{T} \cdot \mathbf{x}}$$

$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

for the case

$\theta_0 + \theta_1 x_1$

Goal: Find θ that **minimizes** the cost function $J(\theta)$

The intercept θ_0 **is not regularized.**

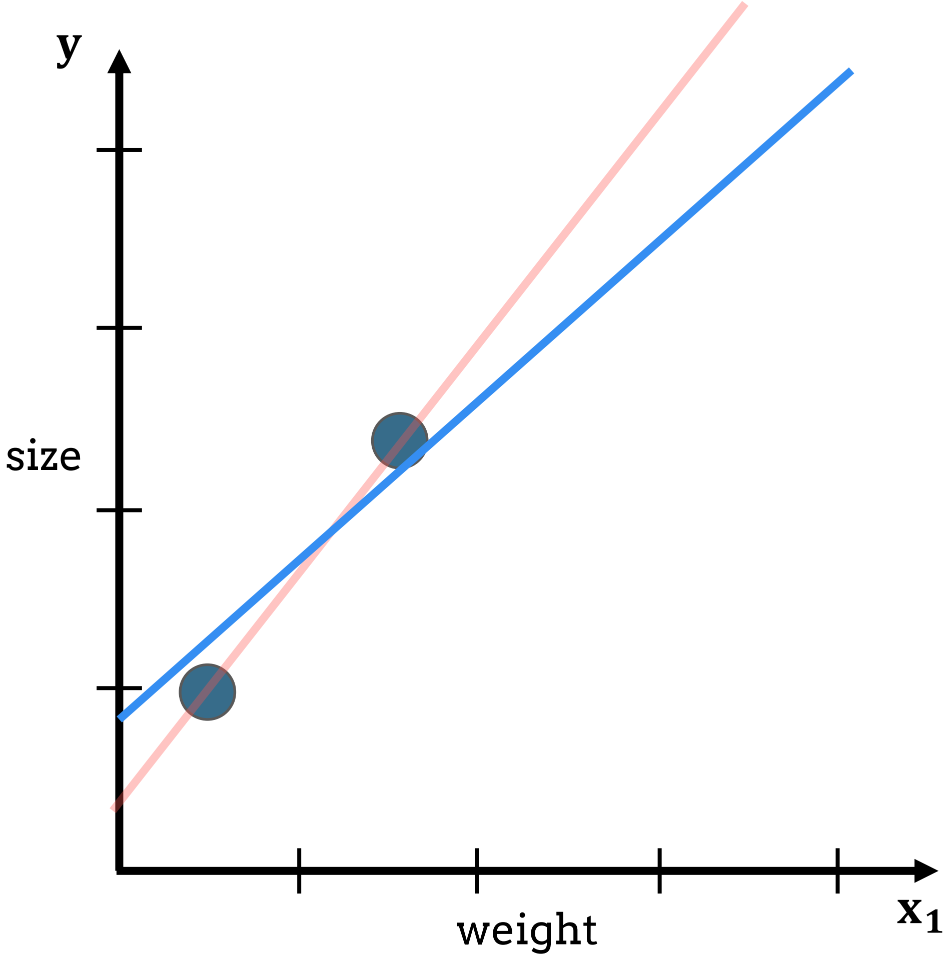
$$J(\theta) = \text{MSE}(\mathbf{X}, h_\theta) + \underbrace{\alpha \sum_{i=1}^{n} \theta^2}$$

for the case

$\alpha(\theta_1^2)$

It determines **how severe that penalty** is.

regularization term penalty



Ridge Regression (L2 Regularization)

$$\hat{y} = h_\theta(\mathbf{x}) = \underbrace{\theta^{T} \cdot \mathbf{x}}$$

n features

$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

for the case

$\theta_0 + \theta_1 x_1$

⚠ The **intercept** θ_0 **is not regularized**.

Goal: Find θ that **minimizes** the cost function $J(\theta)$

$$J(\theta) = \mathrm{MSE}(\mathbf{X}, h_\theta) + \underbrace{\alpha \sum_{i=1}^{n} \theta^2}$$

for the case

$\alpha(\theta_1^2)$

It determines **how severe that penalty** is.

regularization term penalty

This forces the learning algorithm to ***not only* fit the data** but also **keep the model weights** as small as possible.

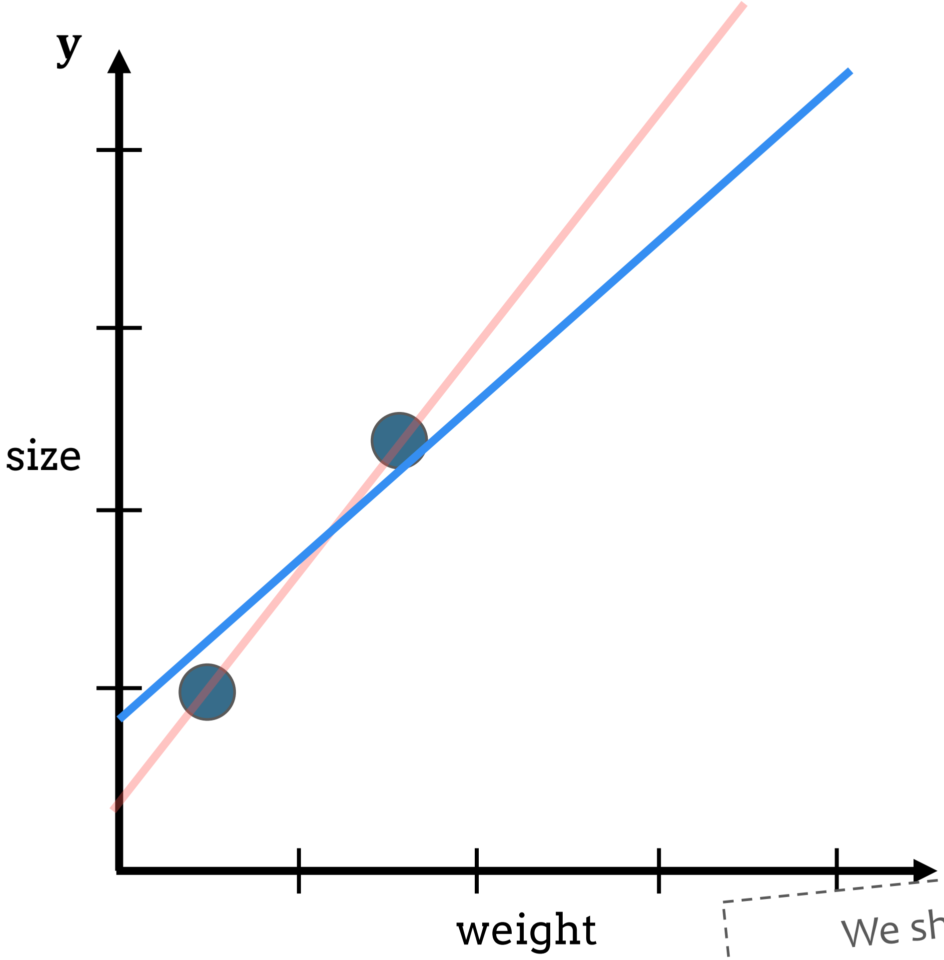
Ridge Regression (L2 Regularization)

n features

$$\hat{y} = h_\theta(\mathbf{x}) = \underbrace{\theta^T \cdot \mathbf{x}}$$

$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

for the case $\theta_0 + \theta_1 x_1$



Goal: Find θ that **minimizes** the cost function $J(\theta)$

The **intercept** θ_0 **is not regularized**.

$$J(\theta) = \mathrm{MSE}(\mathbf{X}, h_\theta) + \alpha \underbrace{\sum_{i=1}^{n} \theta^2}_{\text{regularization term penalty}}$$

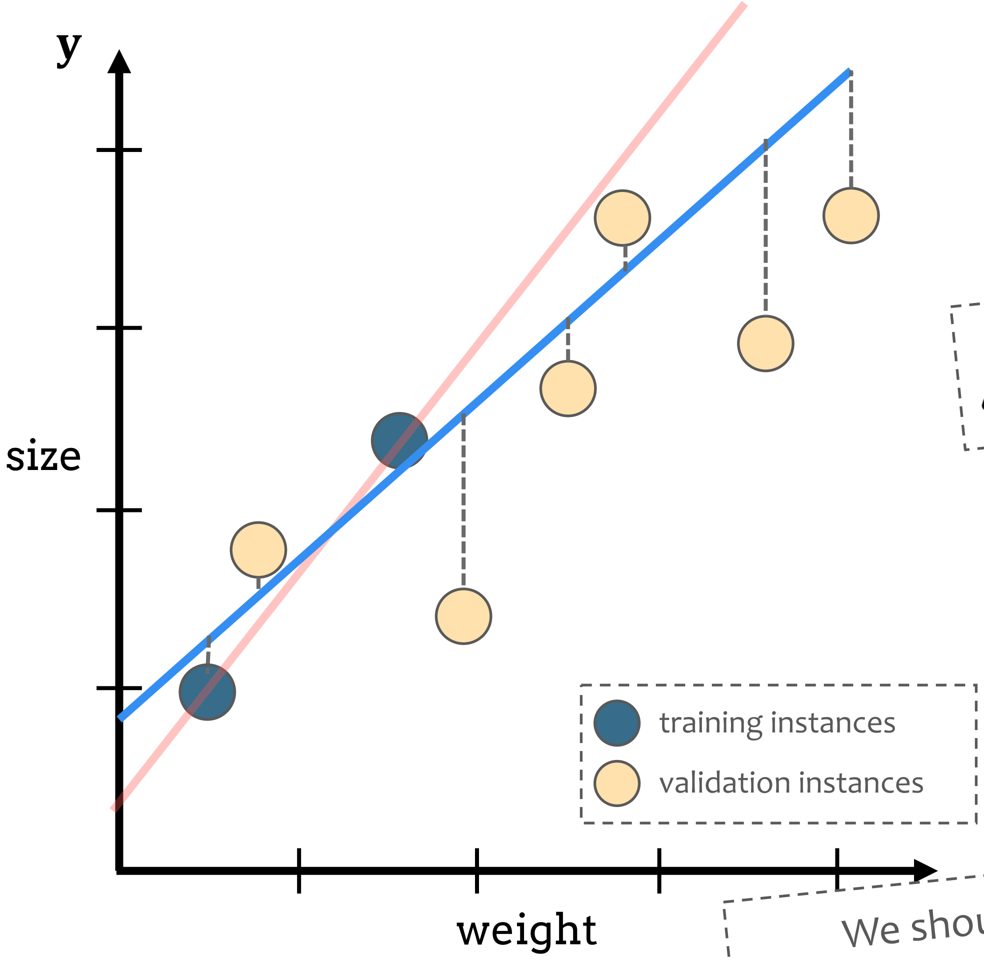
for the case $\alpha(\theta_1^2)$

It determines **how** severe that **penalty** is.

This forces the learning algorithm to ***not only* fit the data** but also **keep the model weights** as small as possible.

We should scale the data **before performing Regularization**, as it is sensitive to the feature scale.

By adding **a small amount of bias** during **model training**, we get **less variance**.



Ridge Regression (L2 Regularization)

$$\hat{y} = h_\theta(\mathbf{x}) = \underbrace{\theta^T \cdot \mathbf{x}}$$

n features: $\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

for the case: $\theta_0 + \theta_1 x_1$

Goal: Find θ that **minimizes** the cost function $J(\theta)$

The intercept θ_0 is not regularized.

$$J(\theta) = \mathrm{MSE}(\mathbf{X}, h_\theta) + \underbrace{\alpha \sum_{i=1}^{n} \theta^2}_{\text{regularization term penalty}}$$

for the case: $\alpha(\theta_1^2)$

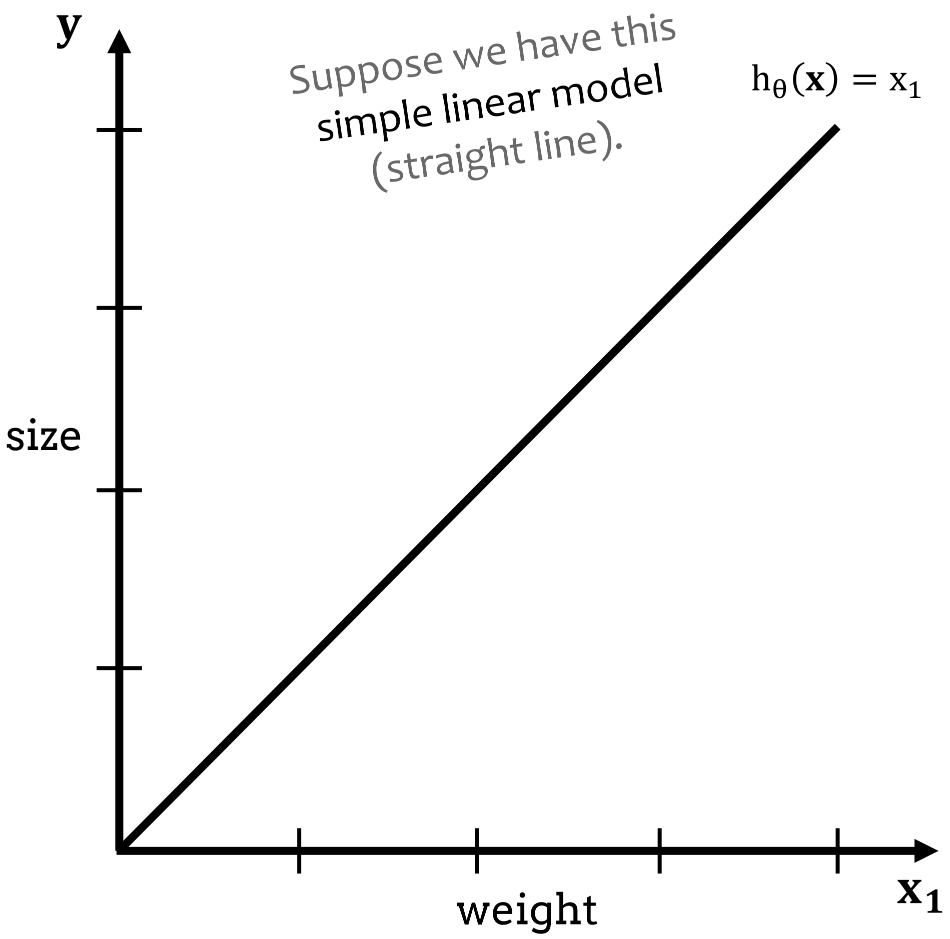
It determines **how severe that penalty** is.

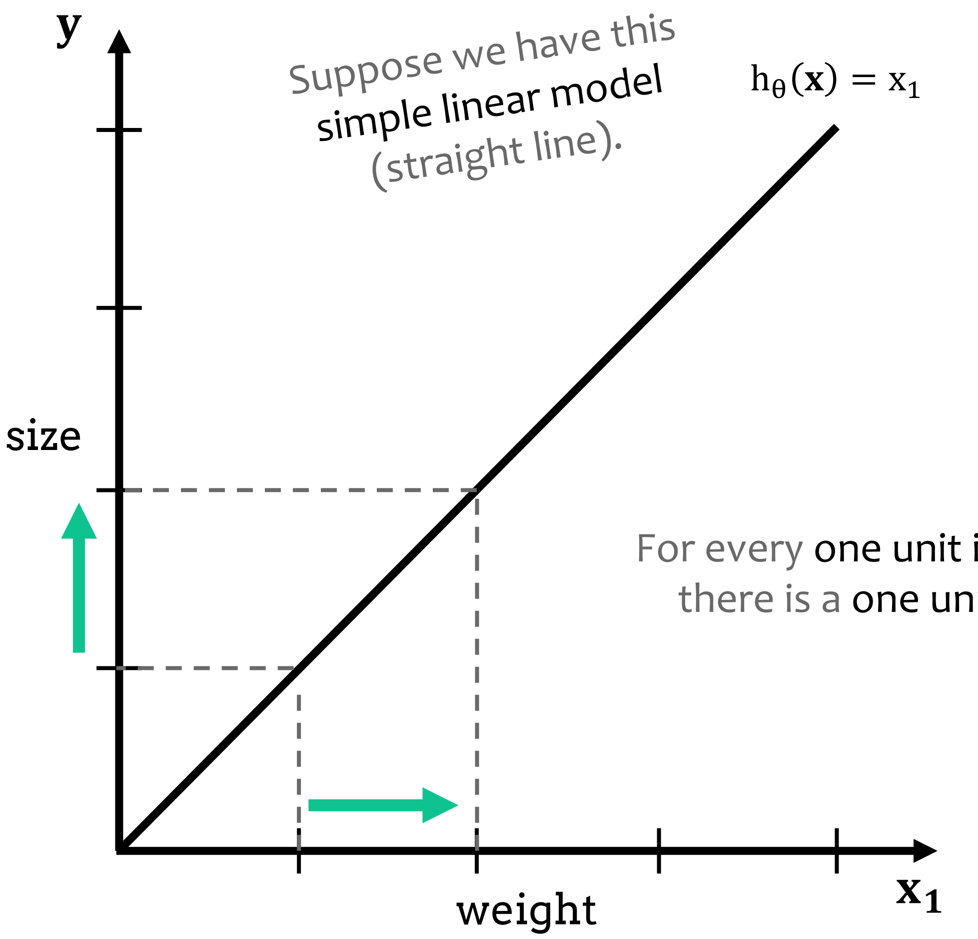
regularization term penalty

This forces the learning algorithm to ***not only* fit the data** but also **keep the model weights as small as possible.**

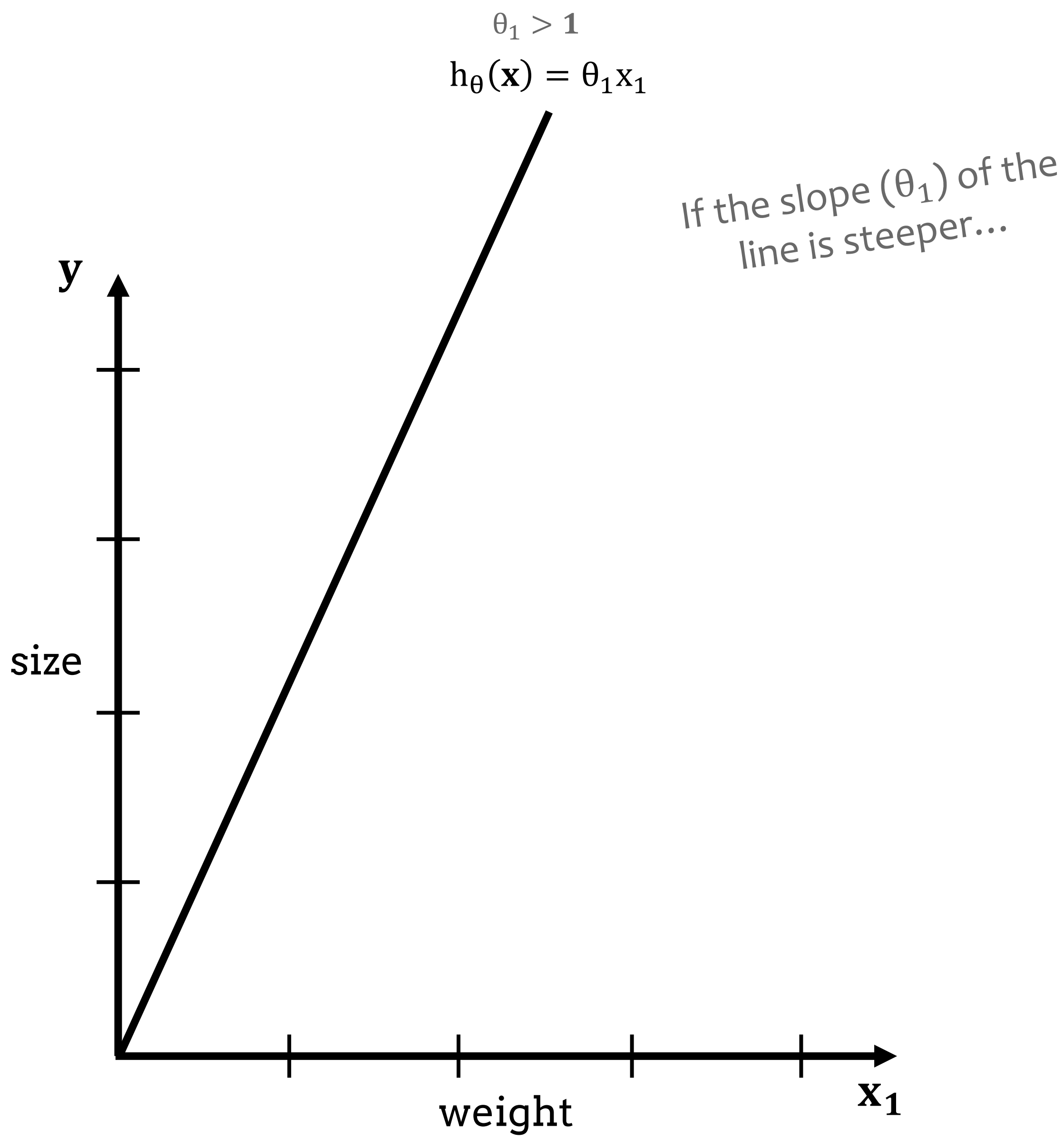
We should **scale the data before performing Regularization**, as it is sensitive to the feature scale.

Impact of the α factor

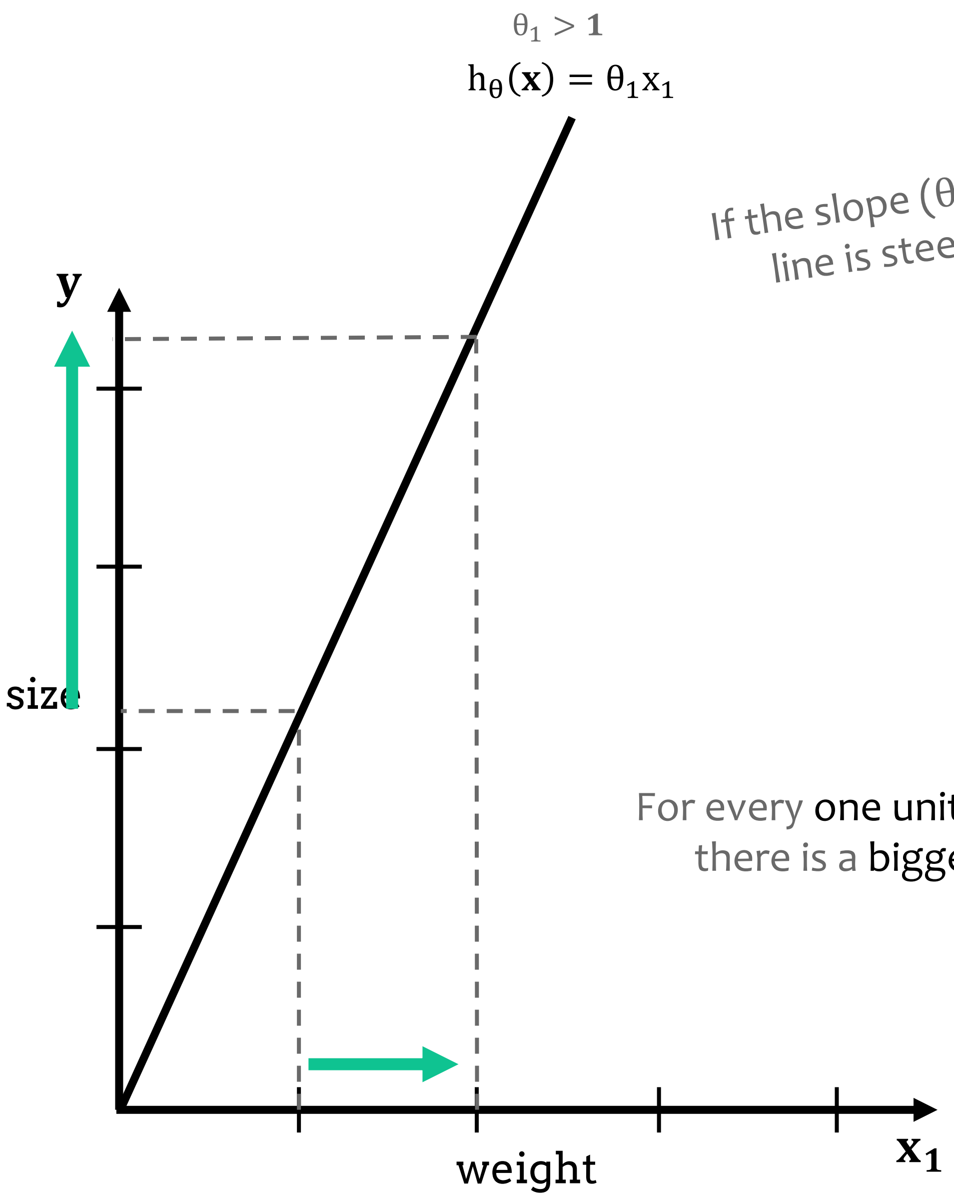

y
Suppose we have this
simple linear model
(straight line).
$h_\theta(\mathbf{x}) = x_1$
size
weight
$\mathbf{x_1}$



For every one unit increase in **weight** ($\mathbf{x_1}$),
there is a one unit increase in **size** ($\mathbf{y}$).

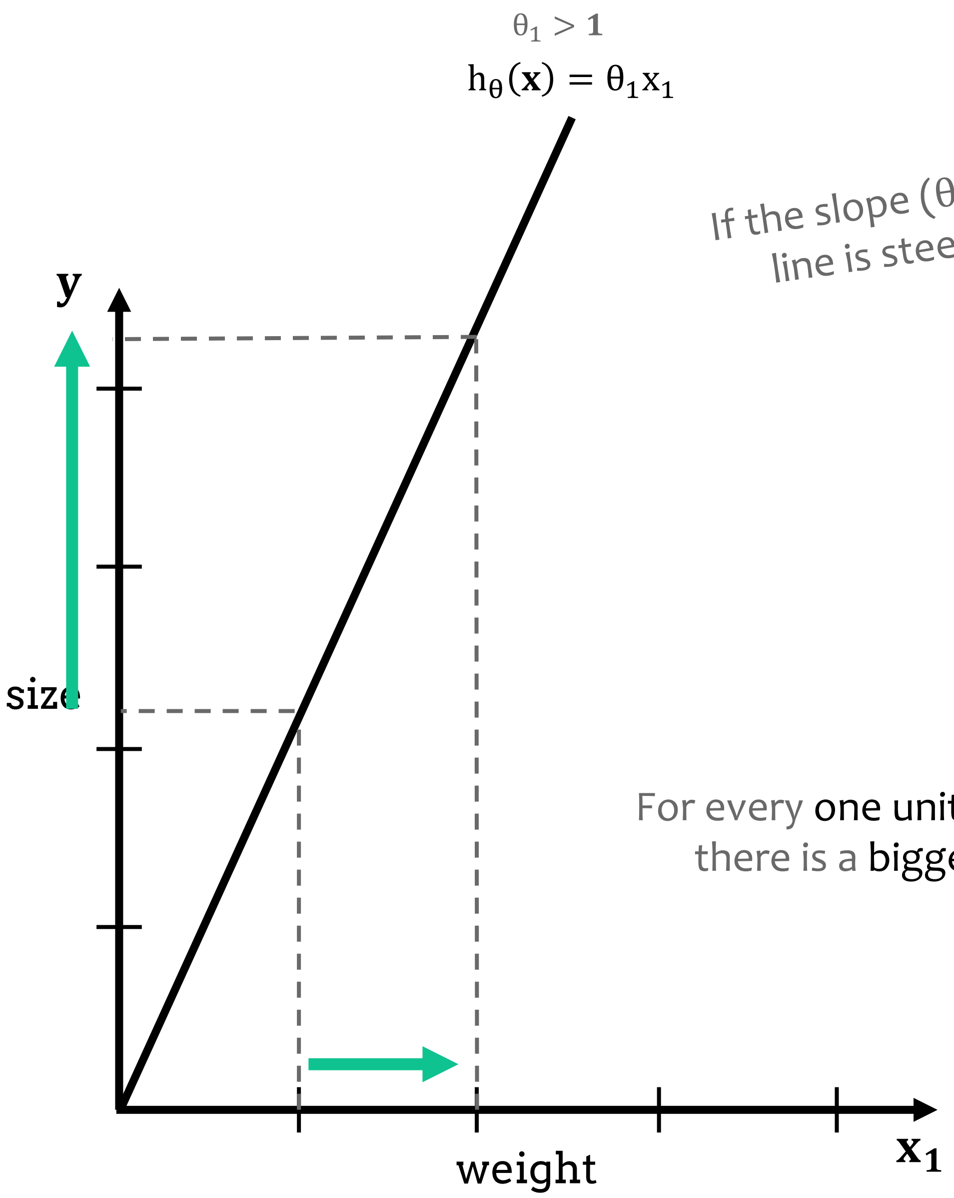


$\theta_1 > \mathbf{1}$
$h_\theta(\mathbf{x}) = \theta_1 x_1$
If the slope (θ_1) of the line is steeper…
y
size
weight
$\mathbf{x_1}$



If the slope (θ_1) of the line is steeper…

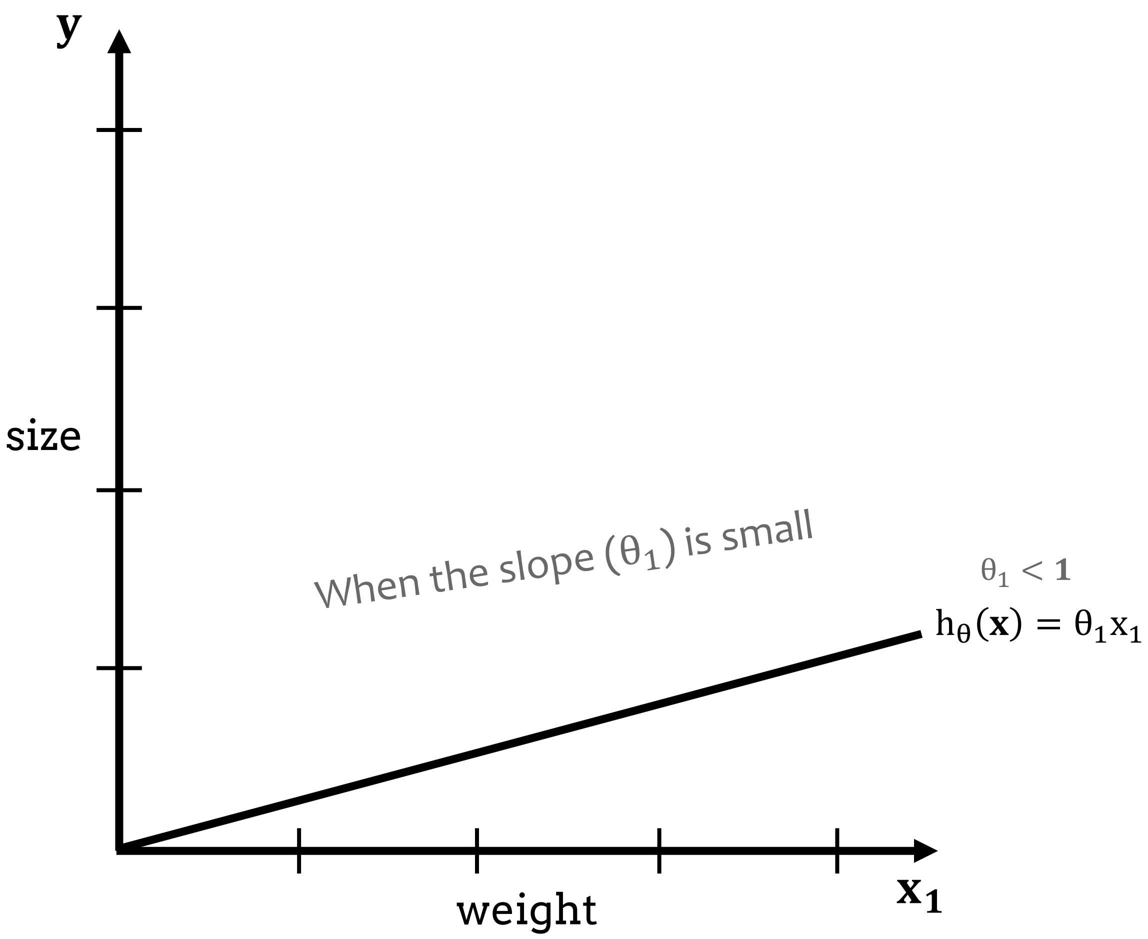
For every one unit increase in **weight** ($\mathbf{x_1}$), there is a bigger increase in **size** ($\mathbf{y}$).



If the slope (θ_1) of the line is steeper…

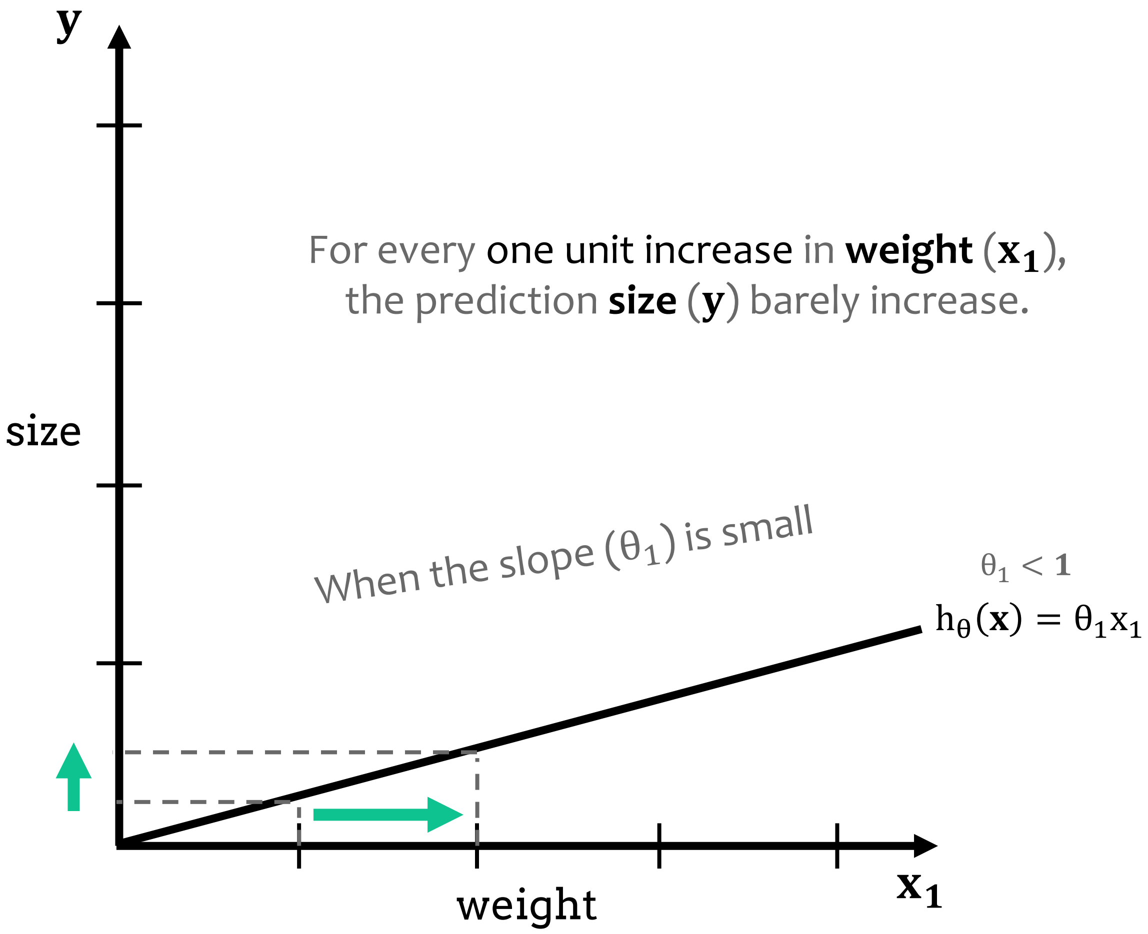
For every one unit increase in **weight** ($\mathbf{x_1}$), there is a bigger increase in **size** ($\mathbf{y}$).

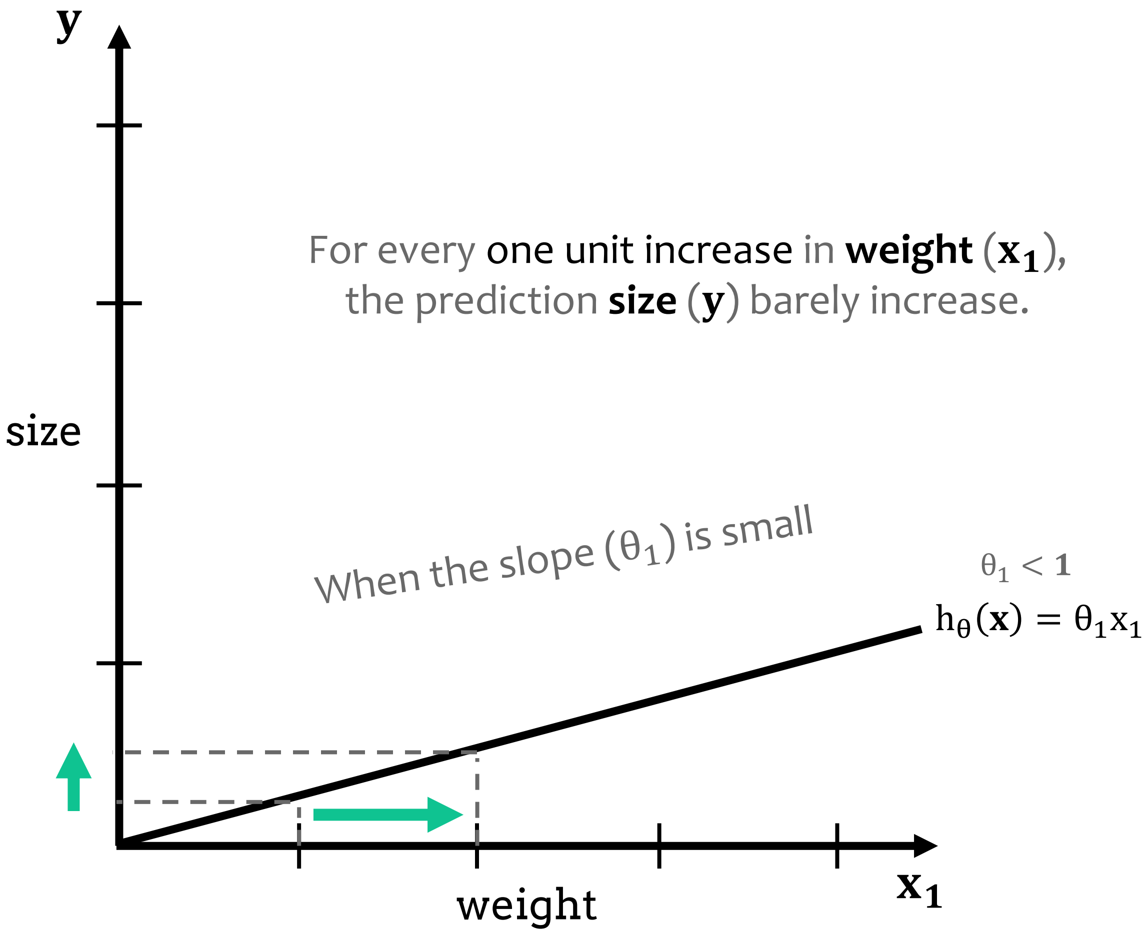
When the **slope of the line** (parameter θ_1) **is steep (high)**, then the prediction for Size **(dependent variable y)** is **very sensitive** to **relatively small changes** in Weight **(independent variable $\mathbf{x_1}$)** .



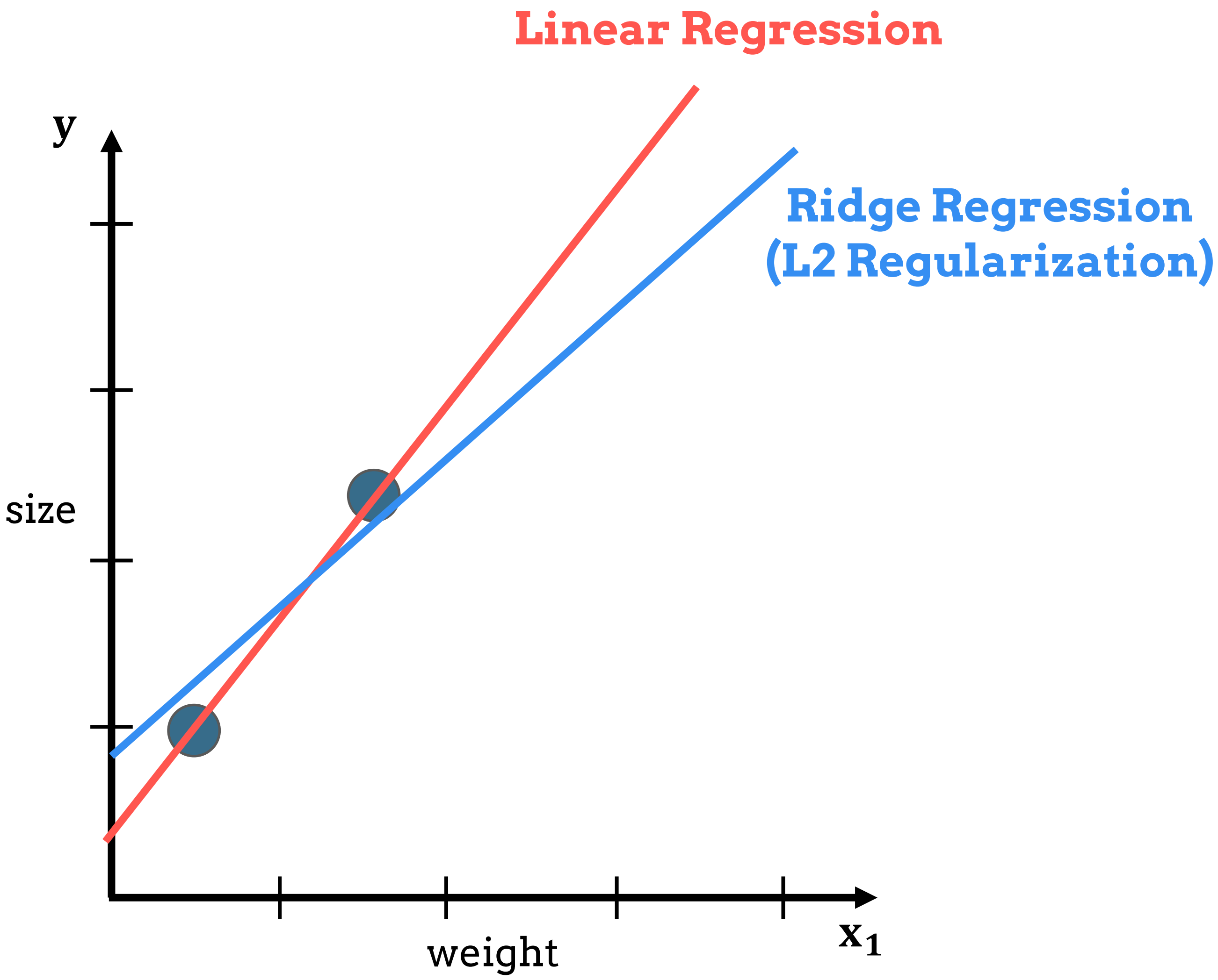
y
size
When the slope (θ1) is small
θ1 < 1
hθ(x) = θ1x1
weight
x1

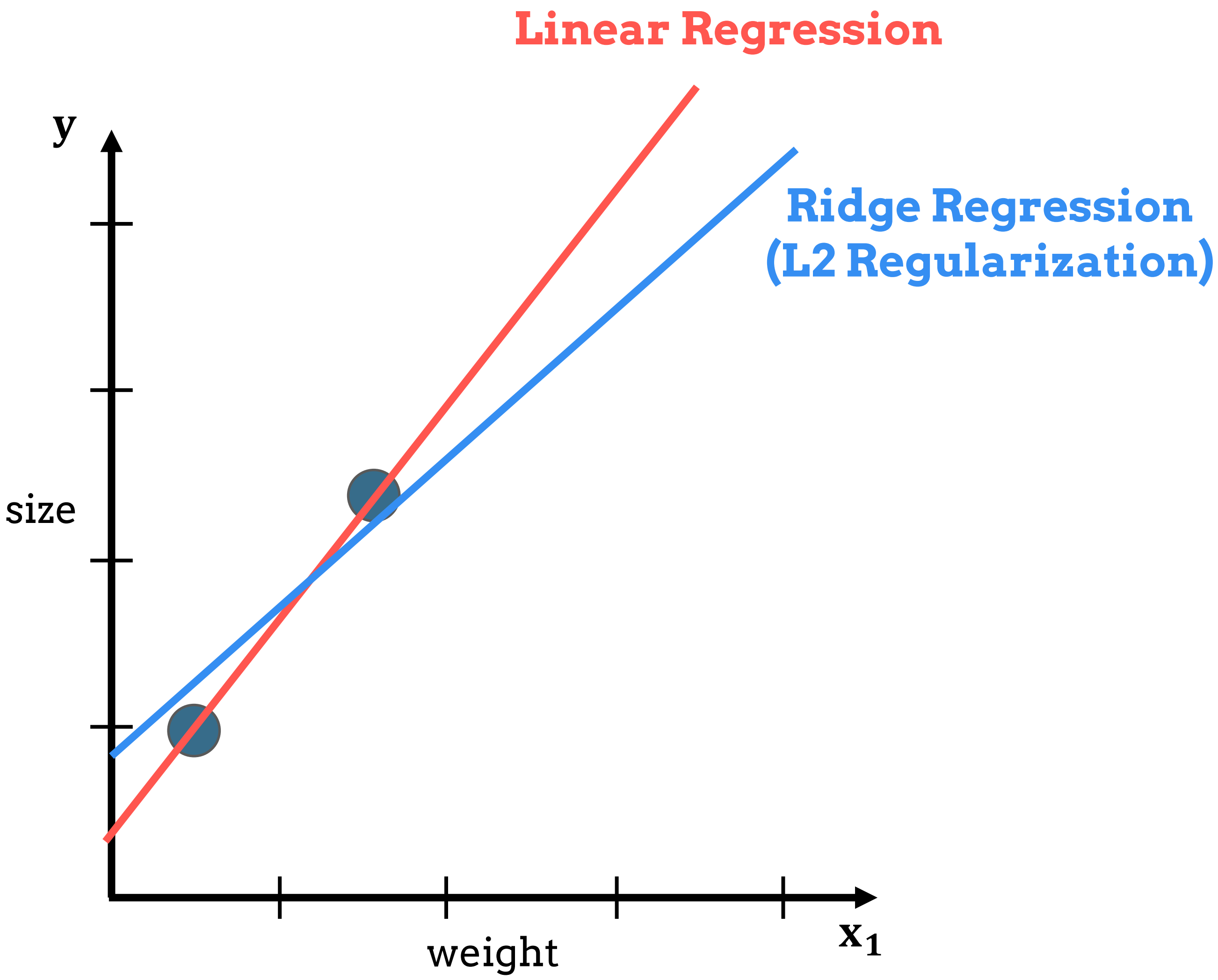
For every one unit increase in **weight** ($\mathbf{x_1}$),
the prediction **size** ($\mathbf{y}$) barely increase.



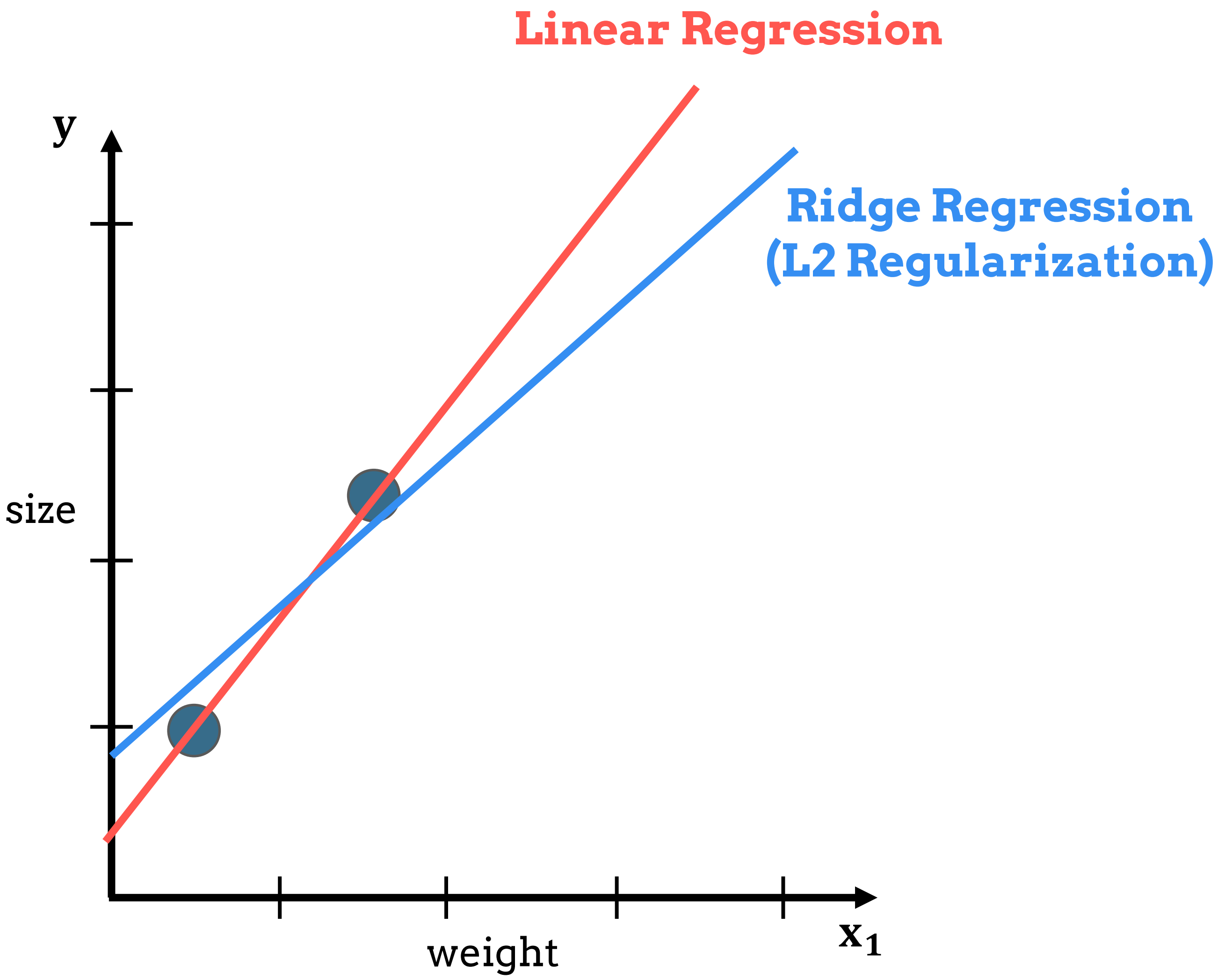


When the **slope of the line** (parameter θ_1) **is small**, then the prediction for Size **(dependent variable $\mathbf{y}$)** is **much less sensitive** to **changes** in Weight **(independent variable $\mathbf{x_1}$)**.


Linear Regression
Ridge Regression
(L2 Regularization)
y
size
weight
x_1



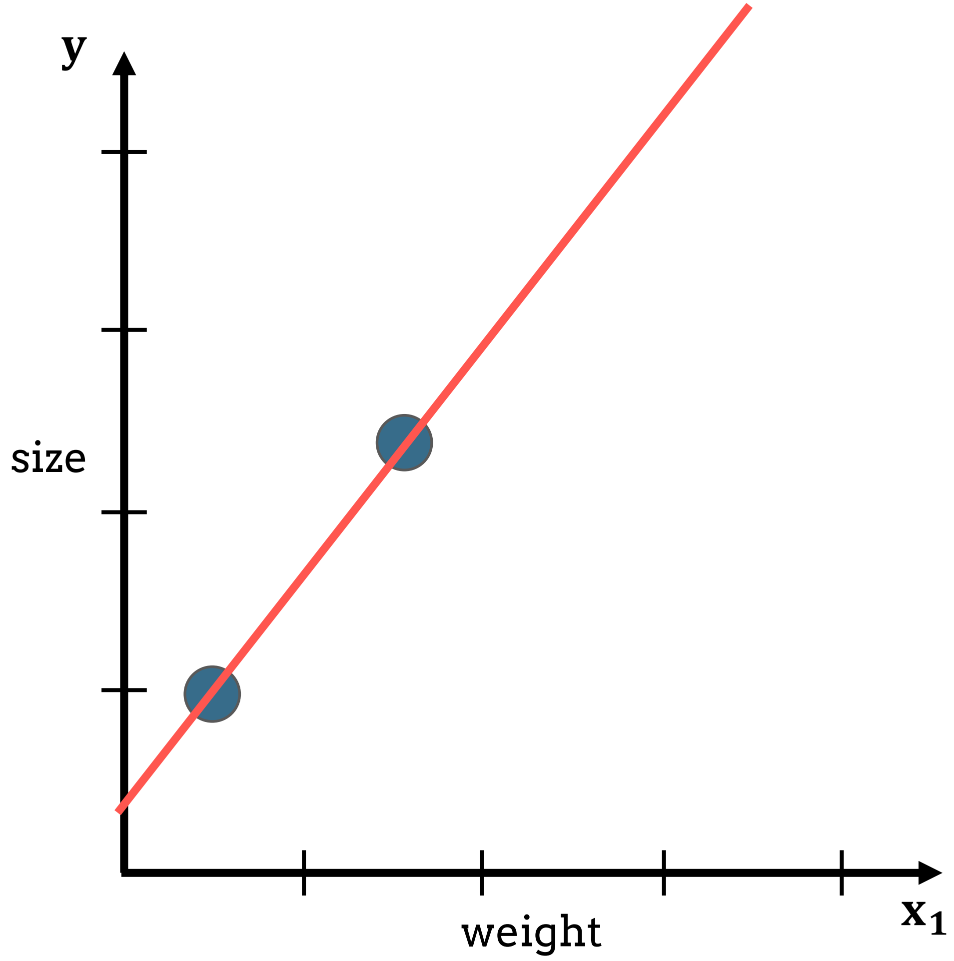
The **Ridge Regression Penalty** resulted in a line with **a smaller slope**.



The **Ridge Regression Penalty** resulted in a line with **a smaller slope**.

The predictions made with the **Ridge Regression model** are **less sensitive** to **changes** in Weight **(independent variable x_1)** are than the **Linear Regression model**.

$$J(\theta) = \text{MSE}(\mathbf{X}, h_\theta) + \alpha \sum_{i=1}^{n} \theta^2$$



TO BE CONTINUED....

Aprendizado de Máquina e Reconhecimento de Padrões 2021.2

Regularization

Prof. Dr. Samuel Martins (Samuka)
samuel.martins@ifsp.edu.br

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
São Paulo
Campus Campinas