ECE ING4
MACHINE LEARNING

Jeremy Cohen

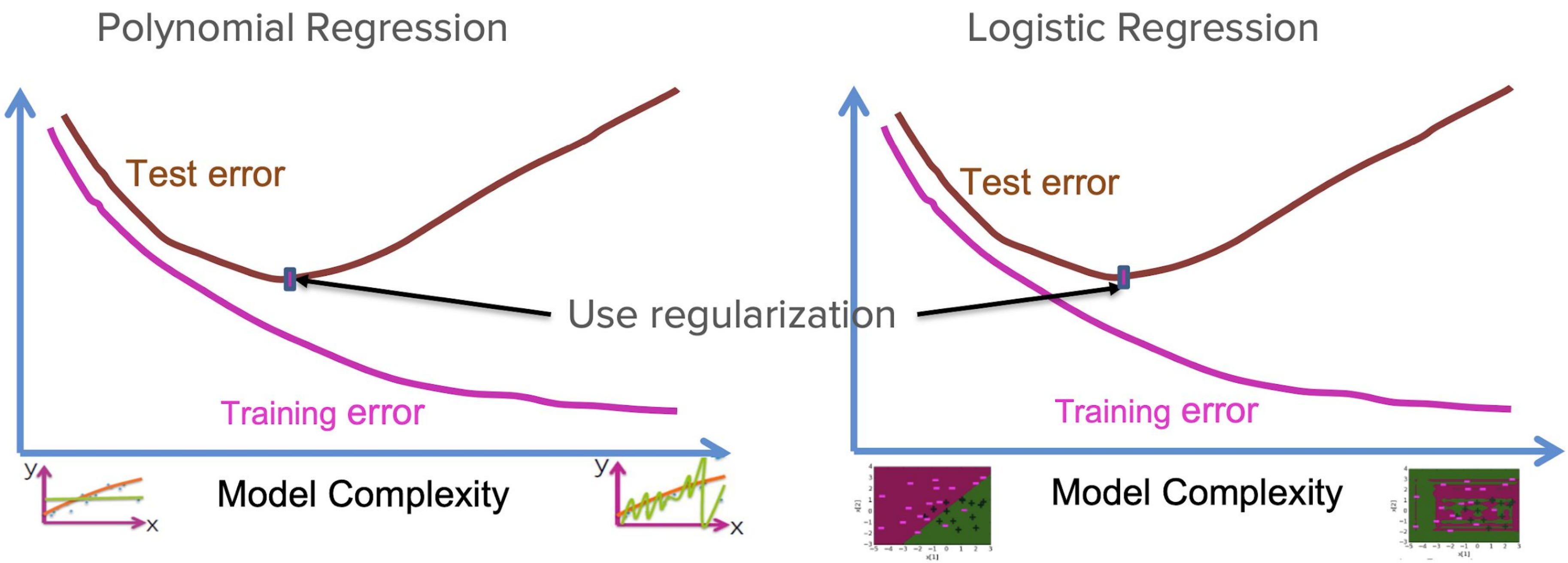
Final Week

- Full Review
- Regularization
- Mini Project

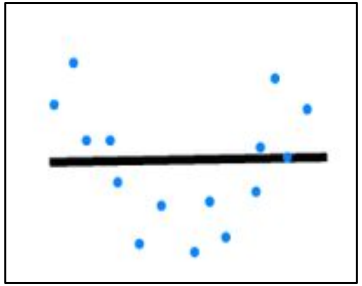
Full Review

Our Final Course: Regularization

The Problem

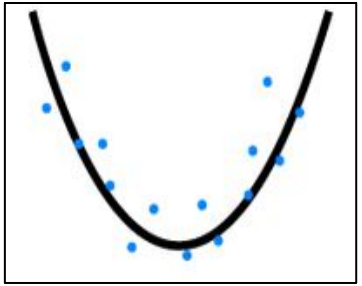

Polynomial Regression
Logistic Regression
Test error
Test error
Use regularization
Training error
Training error
Model Complexity
Model Complexity
y
x
y
x

The Problem



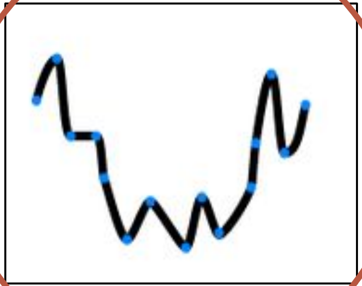
UNDERFITTING

$\theta_0 + \theta_1 x$



JUST RIGHT

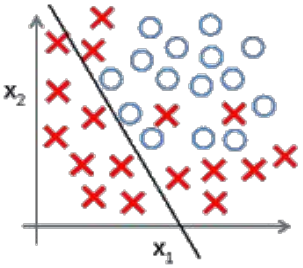
$\theta_0 + \theta_1 x + \theta_2 x^2$



OVERFITTING

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

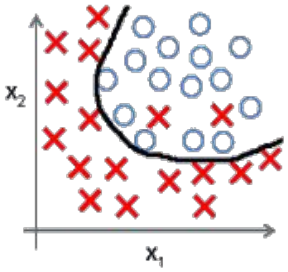
For Logistic Regression



UNDERFITTING

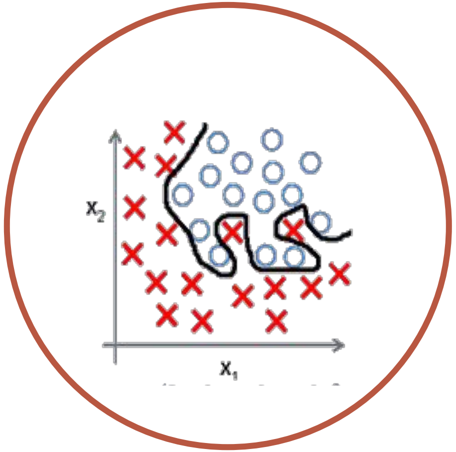
$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

(g = sigmoid function)



JUST RIGHT

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 \\ + \theta_3 x_1^2 + \theta_4 x_2^2 \\ + \theta_5 x_1 x_2)$$



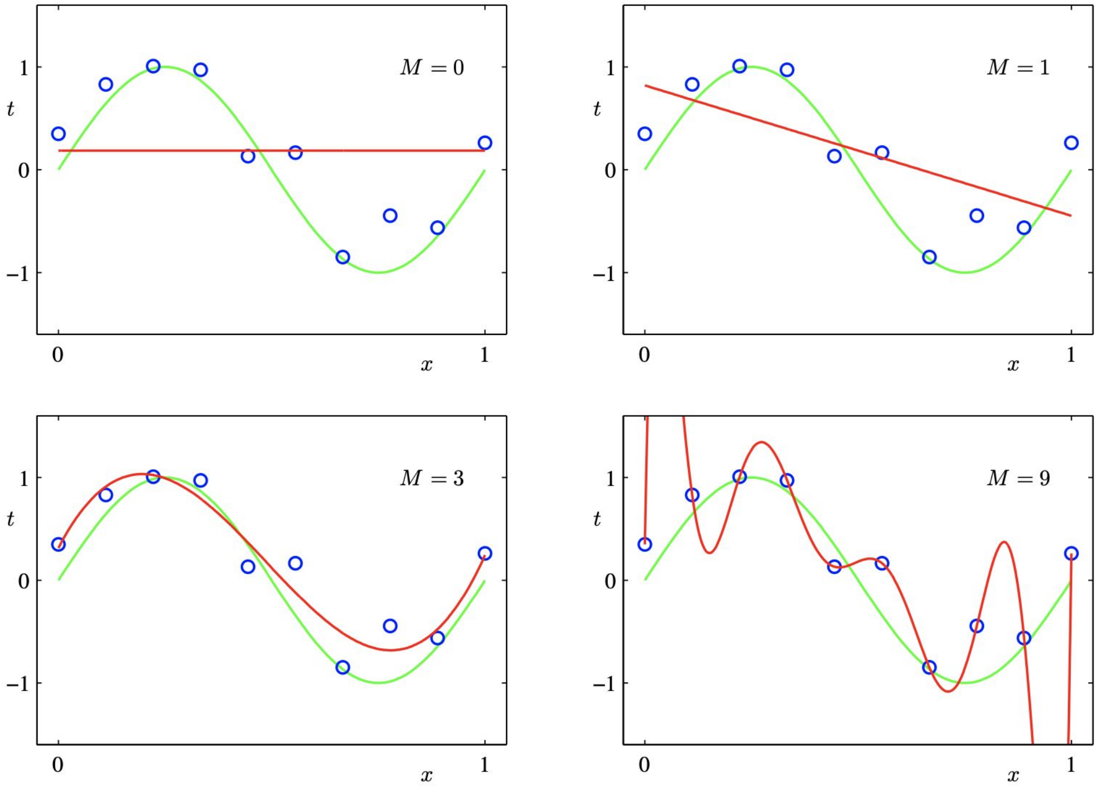
OVERFITTING

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 \\ + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 \\ + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

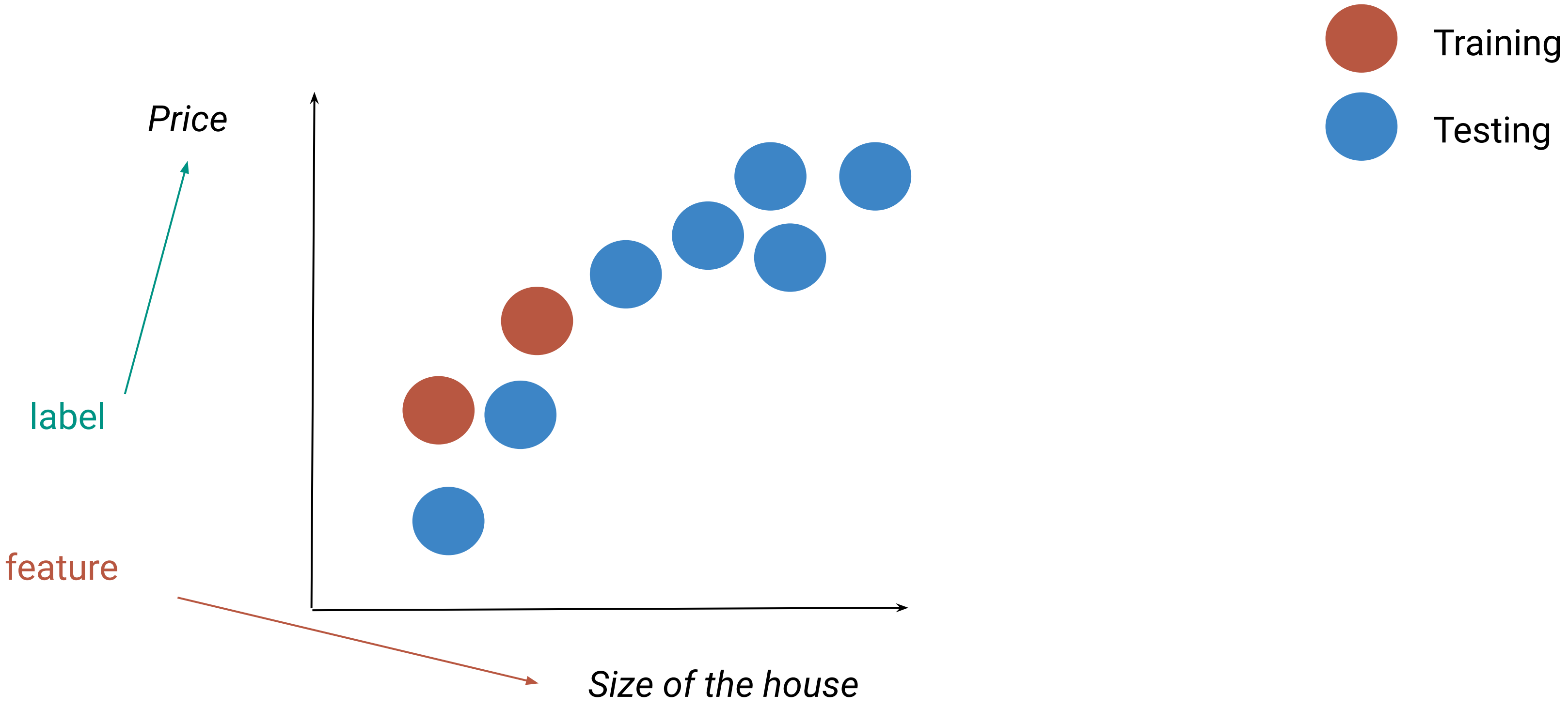
Overfitting

We fit the training data correctly but fail to generalize

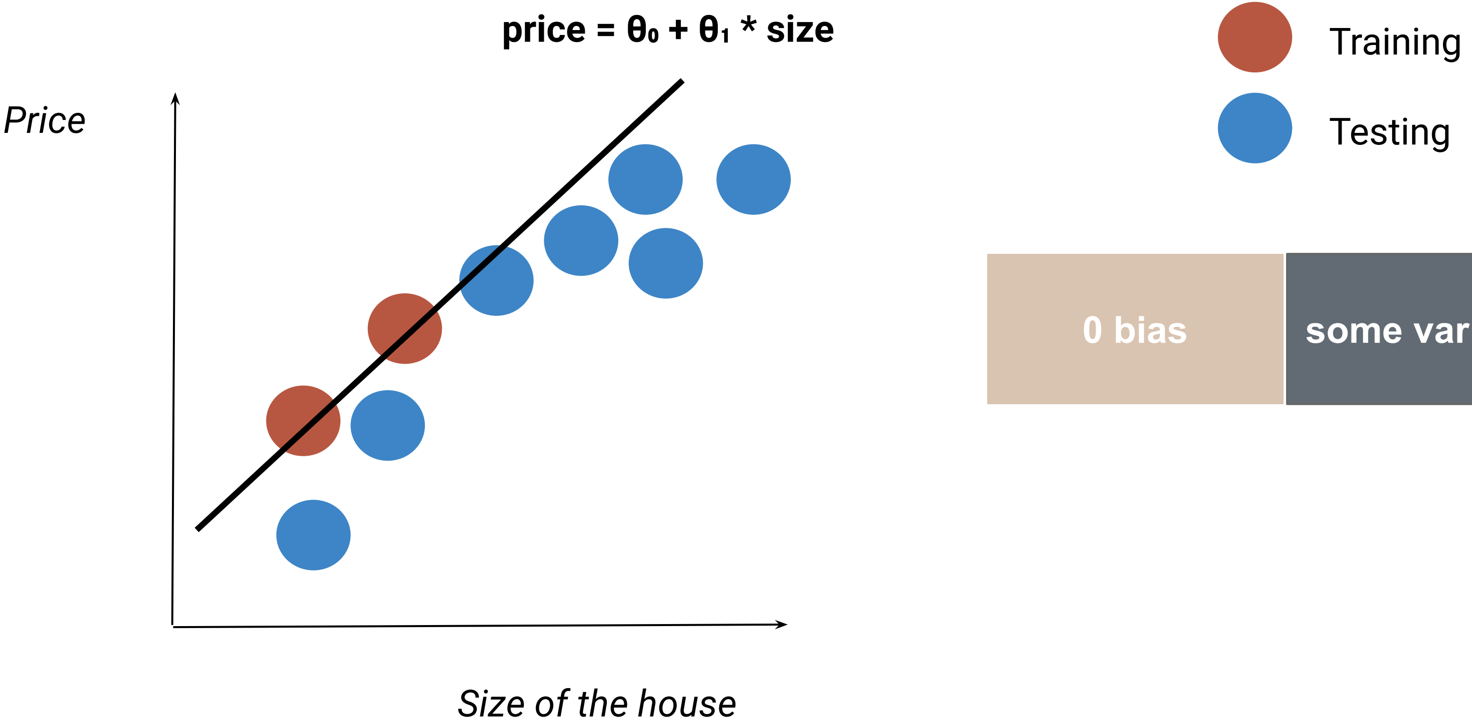
For Logistic Regression



Regression

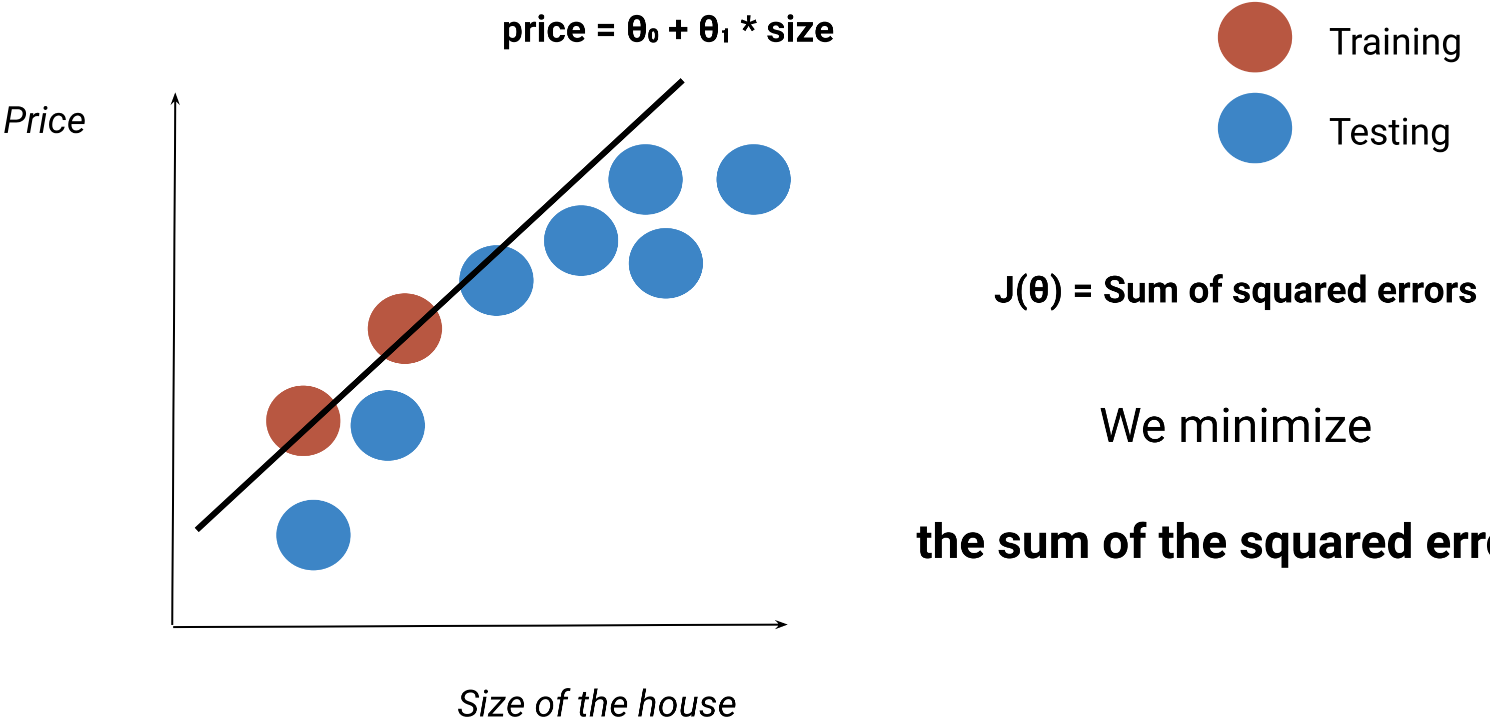


Regression



0 bias | some variance

Regression



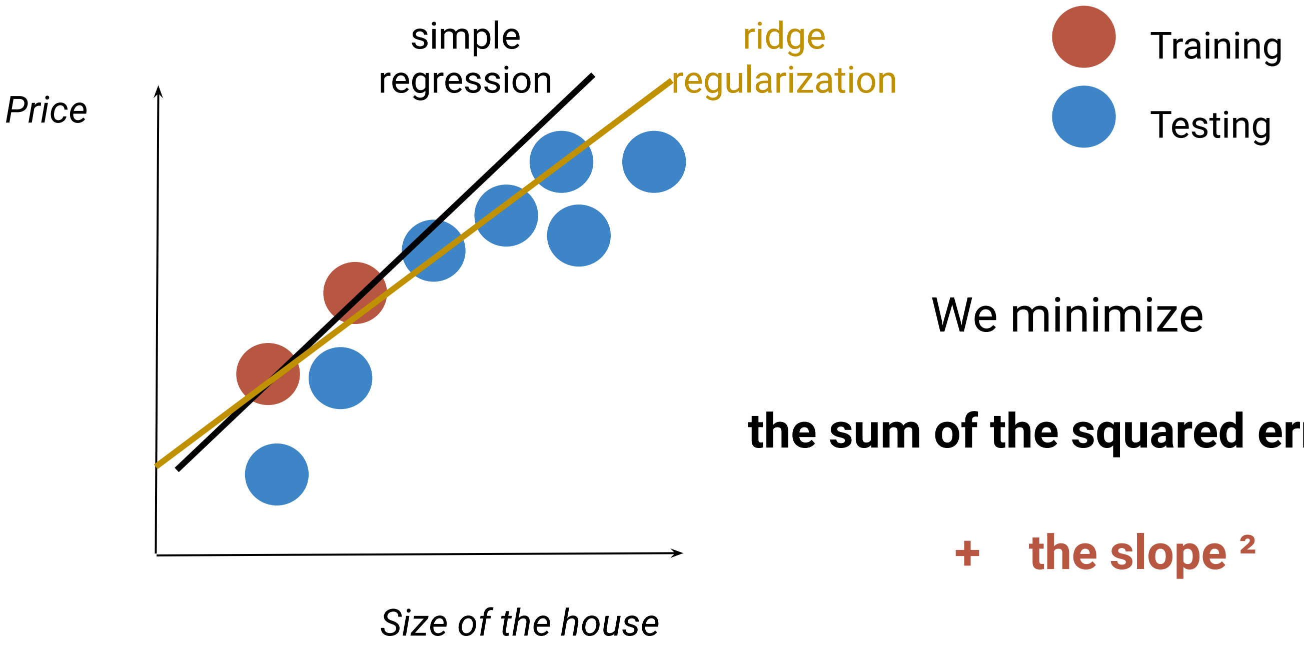
$J(\theta)$ = Sum of squared errors

We minimize

the sum of the squared errors

Regularization

J(θ) = Sum of squared errors + λ * slope²



We minimize

the sum of the squared errors

+ the slope ²

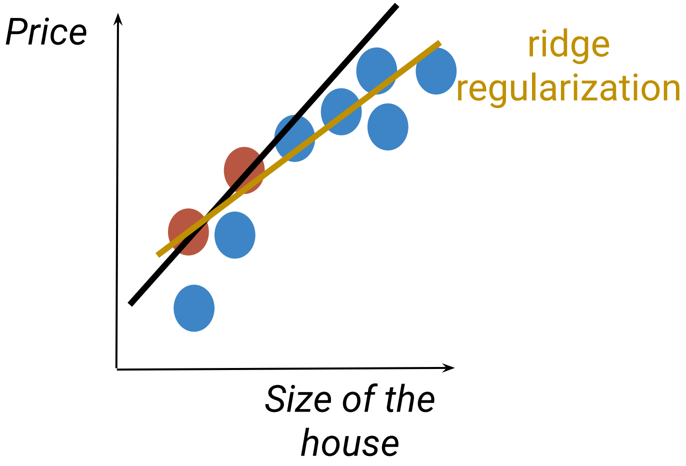
Regularization

$J(\theta)$ = **Sum of squared errors + λ * slope²**

adds a penalty

determine how strong the penalty is

price = $\theta_0 + \theta_1$ * size



We minimize

the sum of the squared errors

+ the slope ²

Ridge Regularization

$J(\theta)$ = Sum of squared errors + λ * slope^2

price = 0.4 + 1.3 * size

for now: $\lambda = 1$

$J(\theta) = 0 + 1 * 1.3^2$

$J(\theta) = 1.69$

Ridge Regularization

$J(\theta)$ = Sum of squared errors + λ * slope^2

price = 0.9 + 0.8 * size

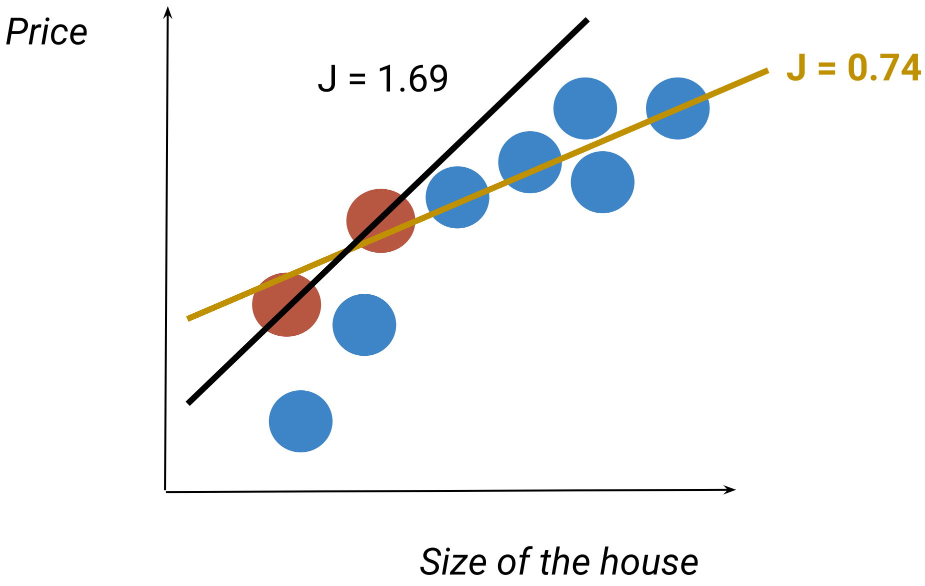
for now: $\lambda = 1$

$J(\theta) = 0.3^2+0.1^2 + 1 * 0.8^2$

$J(\theta) = 0.74$

Ridge Regularization

The price is less sensitive to the size of the house

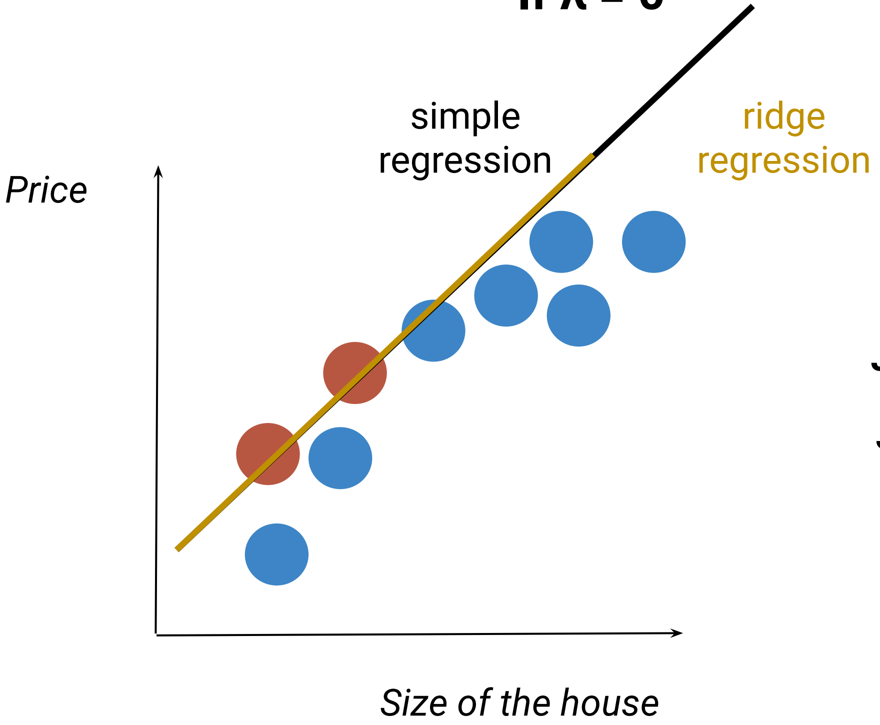


some bias

low variance

What about λ

If λ = 0

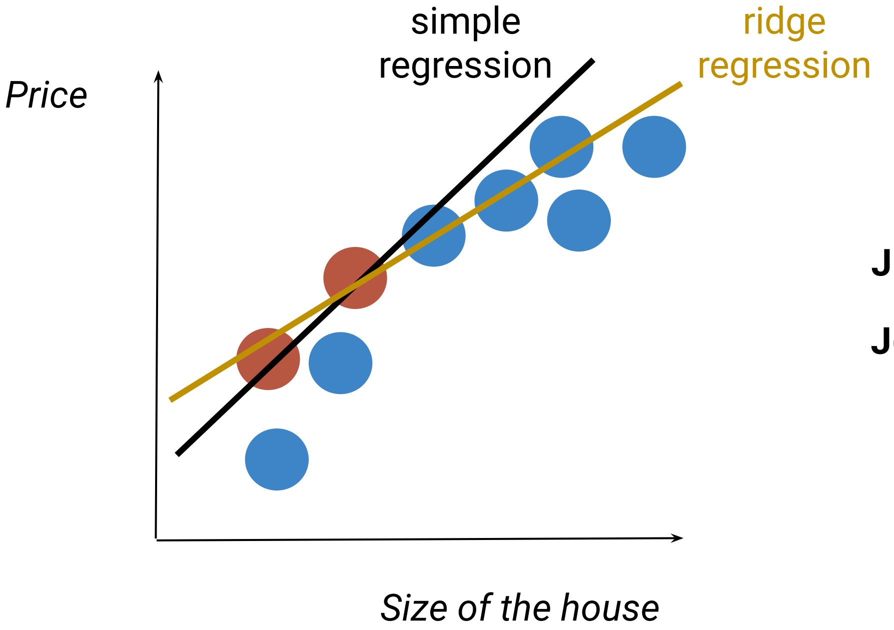


J(θ) = Sum of squared errors + λ*slope²

J(θ) = Sum of squared errors + 0

What about λ

If λ = 1

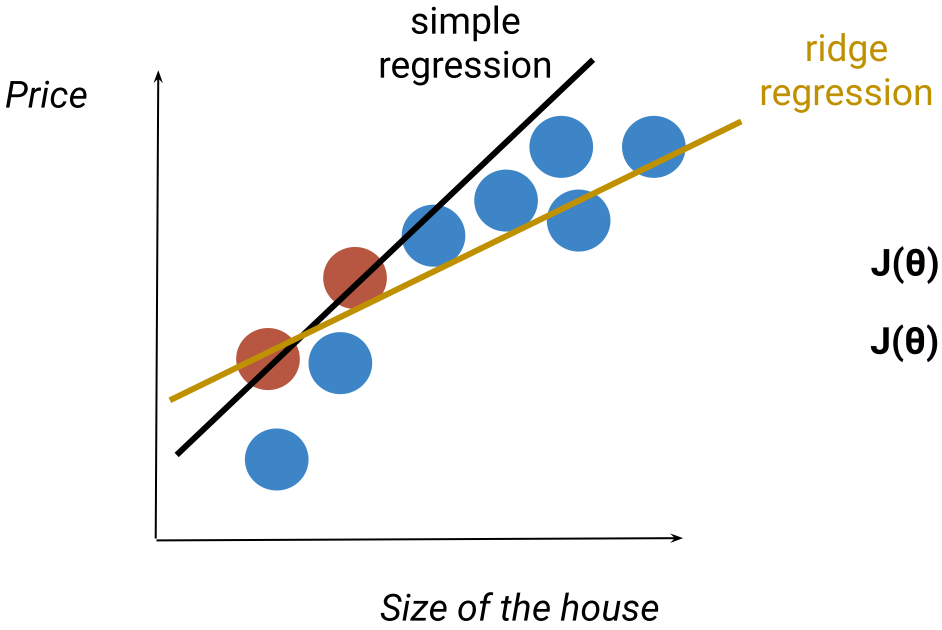


J(θ) = Sum of squared errors + λ*slope²

J(θ) = Sum of squared errors + 1*slope²

What about λ

If λ = 2

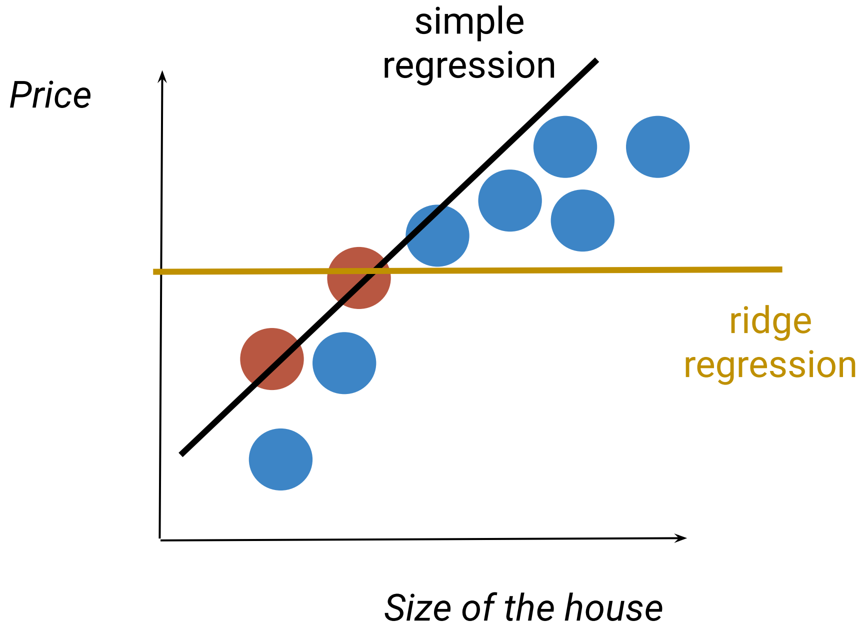


J(θ) = Sum of squared errors + λ*slope²

J(θ) = Sum of squared errors + 2*slope²

What about λ

If λ = 10000

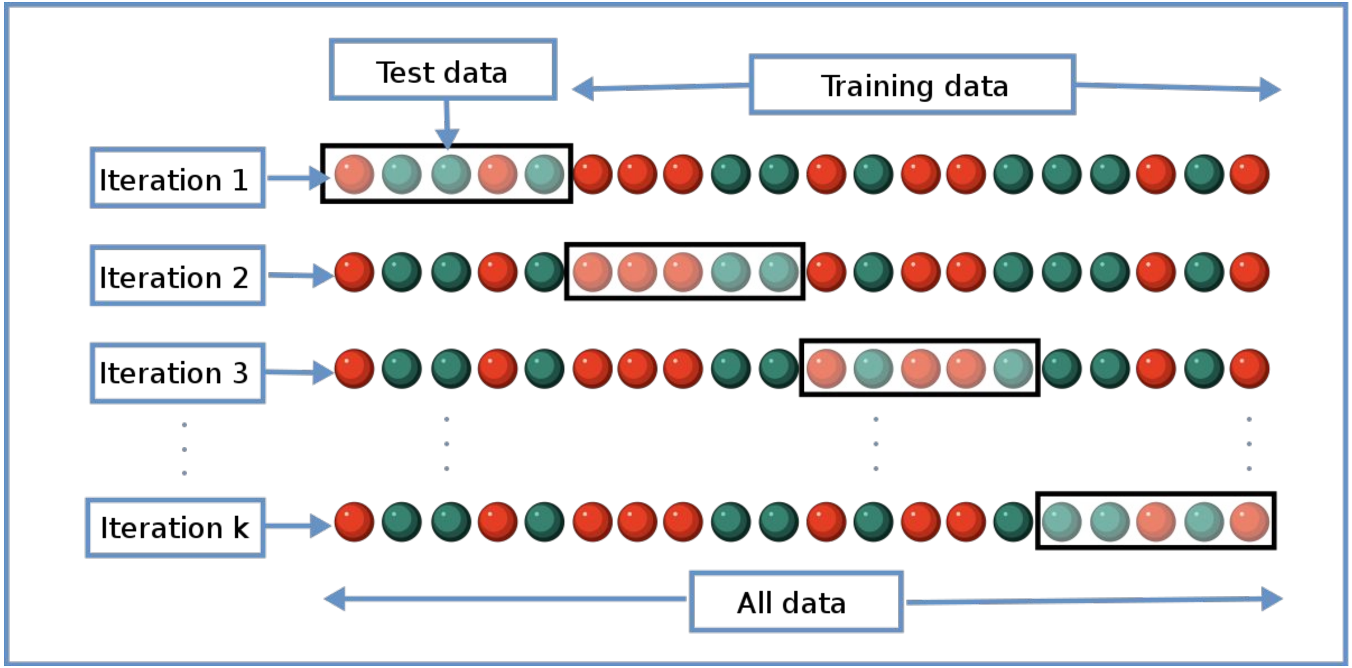


J(θ) = Sum of squared errors + λ*slope²

J(θ) = Sum of squared errors + 10000*slope²

What about λ

To estimate λ, we use cross-validation



Multiple Features

price = θ_0 + slope1 * size + slope2* rating + slope3 * number of bedrooms+...

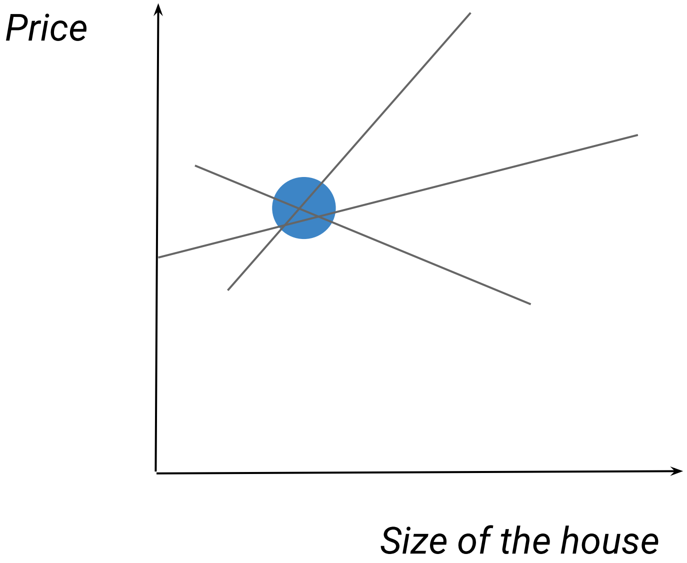
We minimize

the sum of the squared errors

+ λ (slope1^2 + slope2^2 + slope3^2)

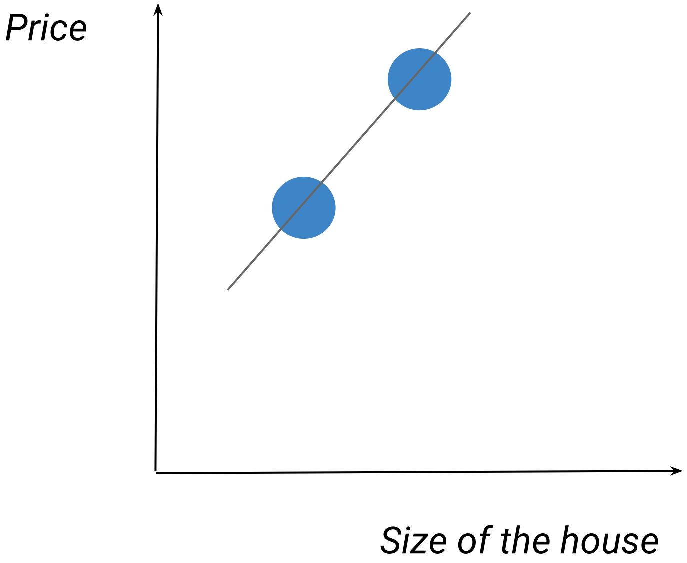
Multiple features

2 features - 1 data point



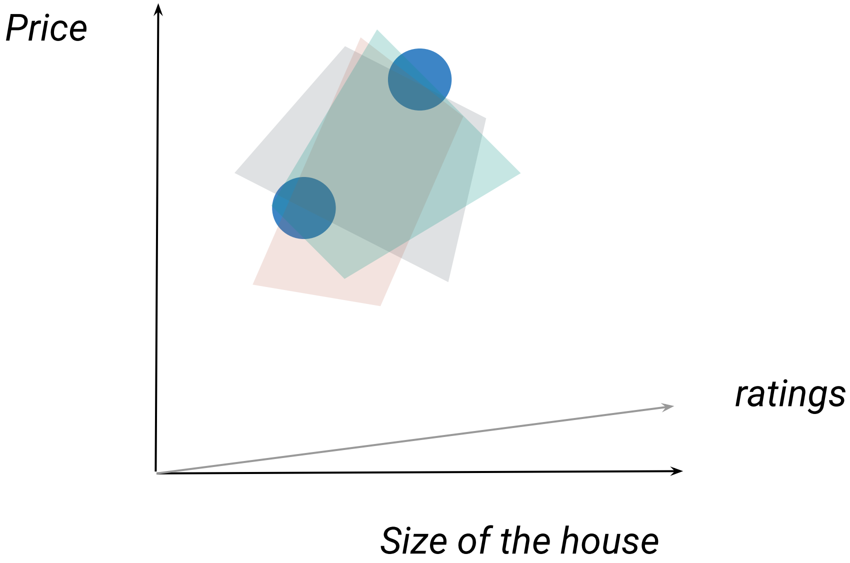
Multiple features

2 features - 2 data points



Thousands of features

3 features - 2 data points



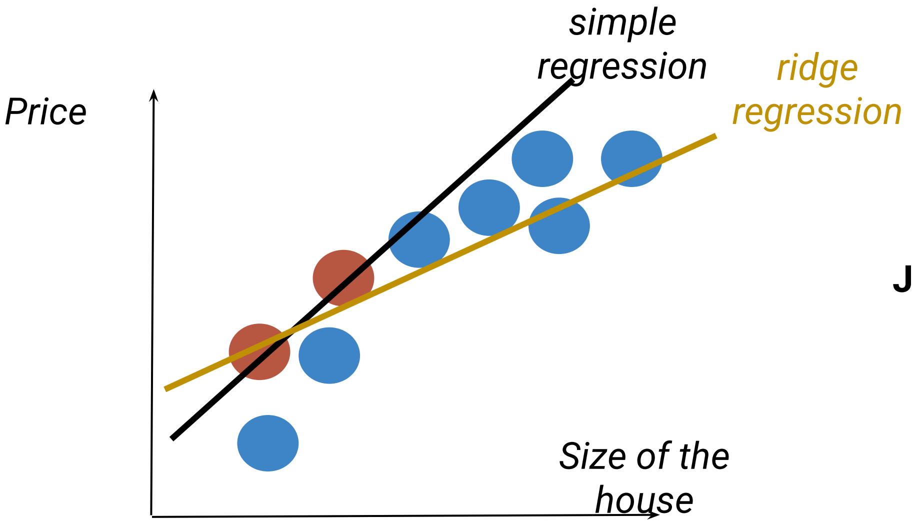
Thousands of features

If we have 4 parameters, we need 4 data points

In case of a dataset with not enough data points compared to the number of features, regularization can help setting some parameters to 0

Summary

Ridge Regularization makes the regression less sensitive to the training data (especially when in low number) and helps reduce overfitting by adding a penalty to the cost function.



J(θ) = Sum of squared errors + λ*slope²

Ridge Regularization

Ridge Regression (L_2 regularization)

$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} (\theta_j)^2 \right]$$

Demo

http://madrury.github.io/smoothers/

Lasso Regularization

Ridge Regression (L_2 regularization)

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n}(\theta_j)^2\right]$$

Lasso Regression (L_1 regularization)

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n}|\theta_j|\right]$$

λ is the regularization parameter:

- Ridge: Encourages small weights θ but not exactly 0
- Lasso: "Shrink" some weights θ exactly to 0

Gradient Descent

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \quad \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

With regularization: $$\theta_j := \theta_j \boxed{(1 - \alpha \frac{\lambda}{m})} - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

α, λ are learning parameters to choose manually

In practice: (1 – αλ/m) is between 0.99 and 0.95

Logistic Regression

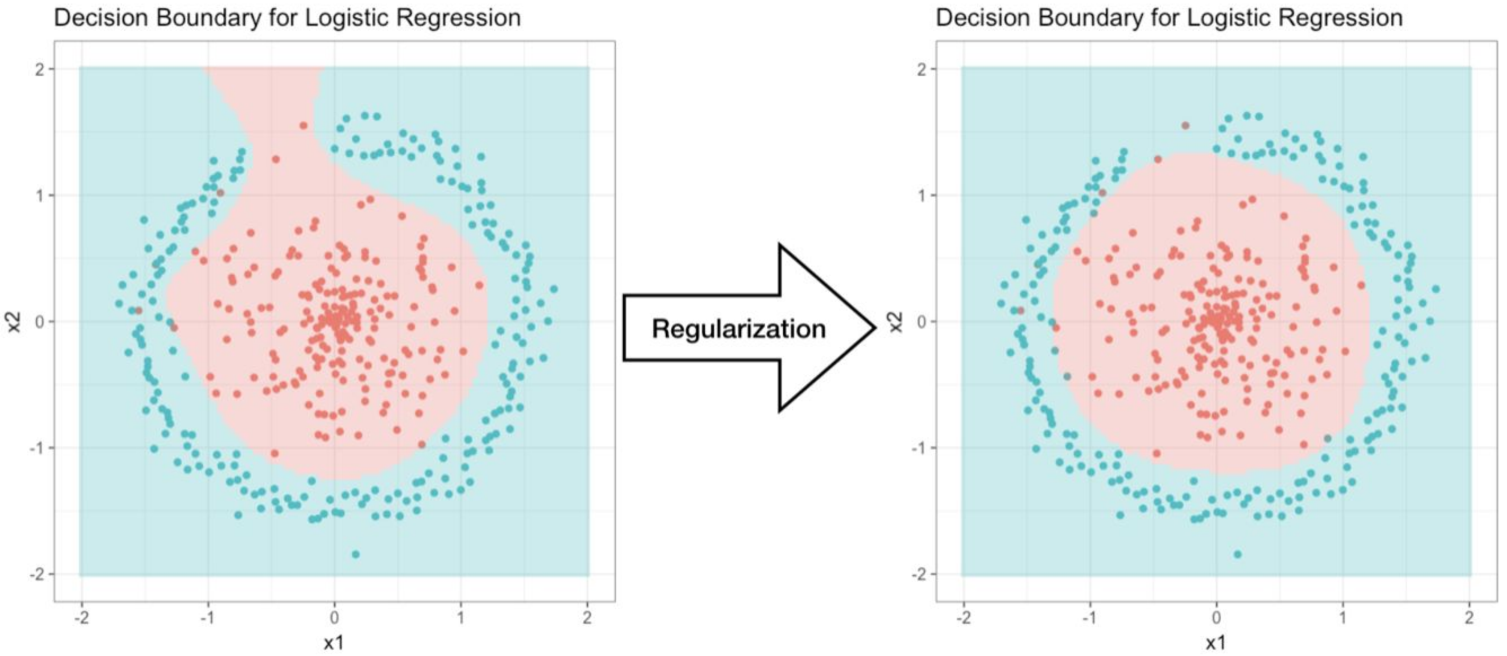
Original Formula

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\ \log(h_\theta(x^{(i)})) + (1-y^{(i)})\ \log(1-h_\theta(x^{(i)}))]$$

Updated Formula

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\ \log(h_\theta(x^{(i)})) + (1-y^{(i)})\ \log(1-h_\theta(x^{(i)}))] + \underbrace{\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2}_{\text{regularization term}}$$

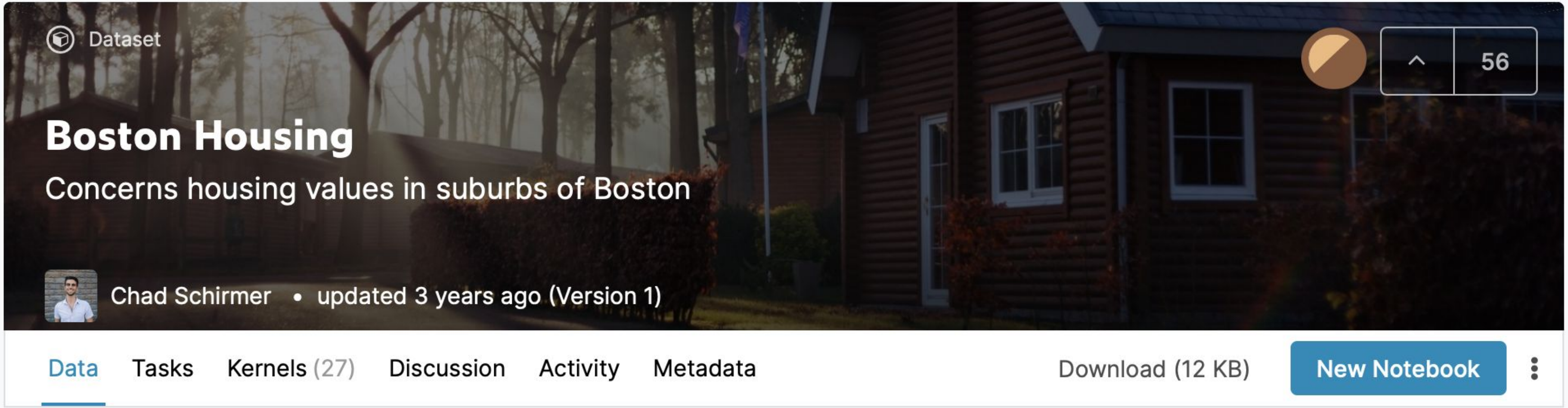
Logistic Regression



MINI PROJECT

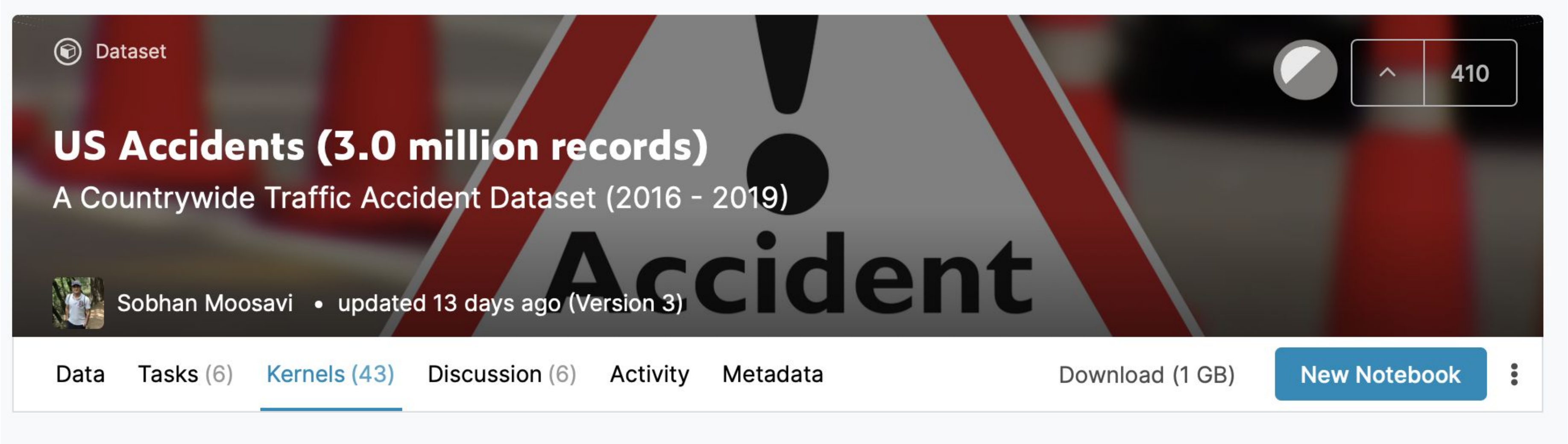
What now?

BOSTON HOUSING PRICE



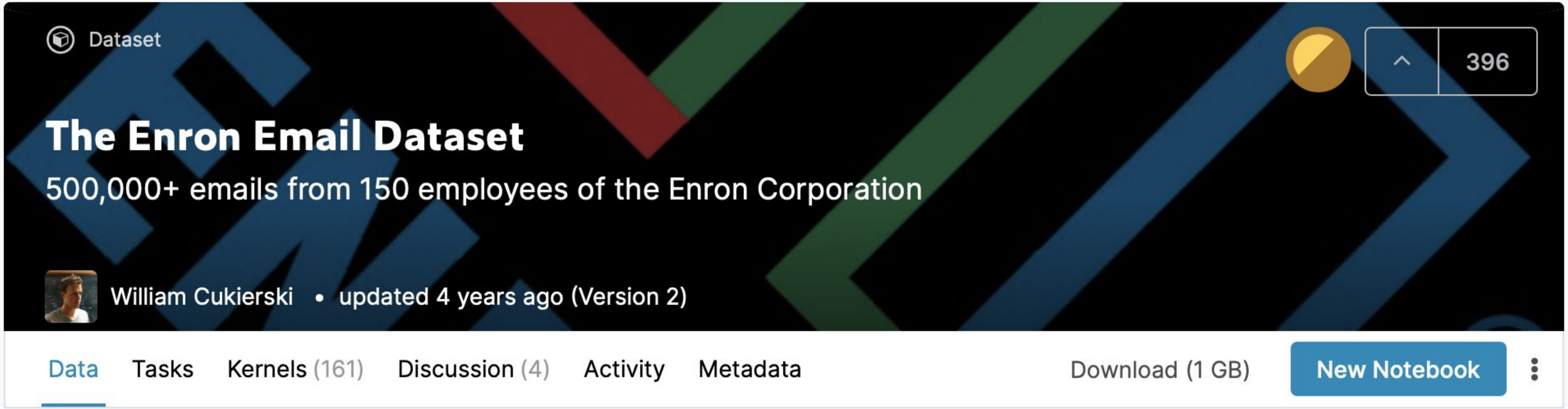
https://www.kaggle.com/schirmerchad/bostonhoustingmlnd

US ACCIDENTS



https://www.kaggle.com/sobhanmoosavi/us-accidents/kernels

ENRON



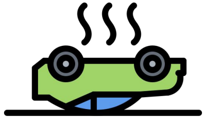
https://www.kaggle.com/wcukierski/enron-email-dataset

FREE CHOICE

Solve a real-world problem of your choice

Solve a problem using Machine Learning

BOSTON HOUSE PRICES



US ACCIDENTS

ENRON SCANDAL

FREE CHOICE

Teaming Up is allowed (up to 2)

FINAL THANK YOU

ECE STUDENTS GET 50% OFF ALL COURSES!

https://jeremycohen.podia.com

COUPON
ECE20

ECE PARIS
ÉCOLE D'INGÉNIEURS

One Step Ahead

See you soon!