

THE NUTS AND BOLTS  
OF  
MACHINE LEARNING

Nuts and Bolts ①

## A Definition

A computer program is said to learn from **experience  $E$**  with respect to some class of **tasks  $T$**  and **performance measure  $P$** , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ .

Tom Mitchell, *Machine Learning* (1997)

Experience / Task / Performance

What is experience?

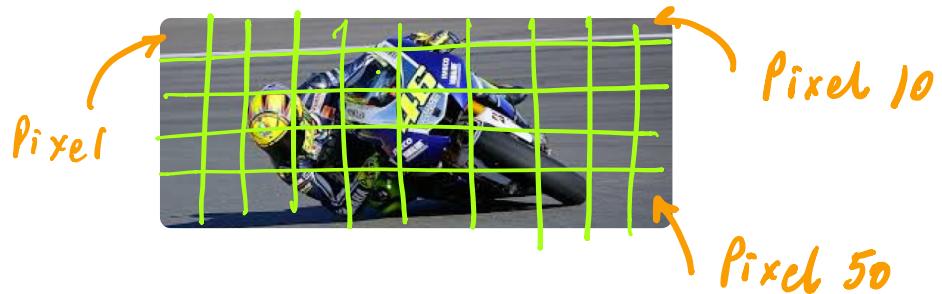
Experience  $\neq$  Rules

Experience = Data

Data = Table of Numbers



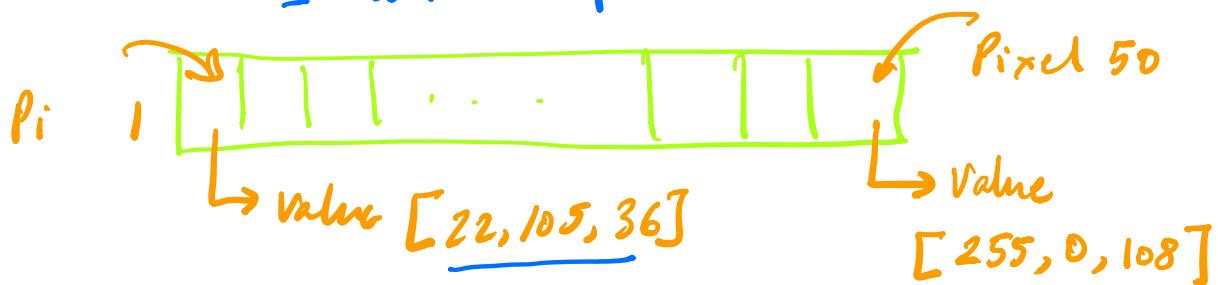
Nuts and Bolts ②  
An image is  
a grid of pixels



Each pixel is represented  
by 3 numbers  
each between 0 and 255

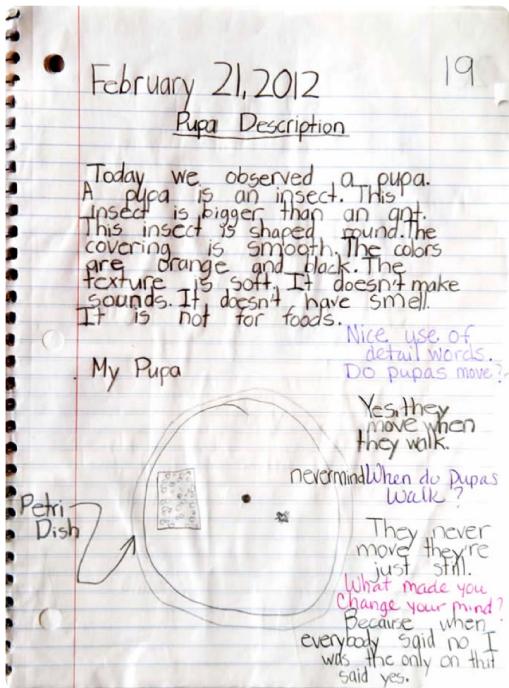
red value  
green value  
blue value

An image is a row of pixel values  
= a row of numbers



Actually: a row of lists of numbers

## Nuts and Bolts (3)

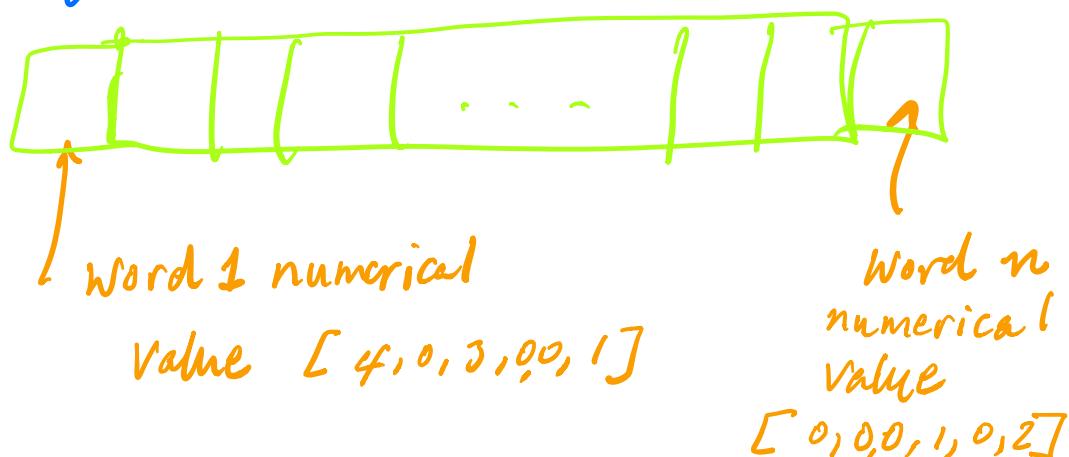


A document is a row of numbers.

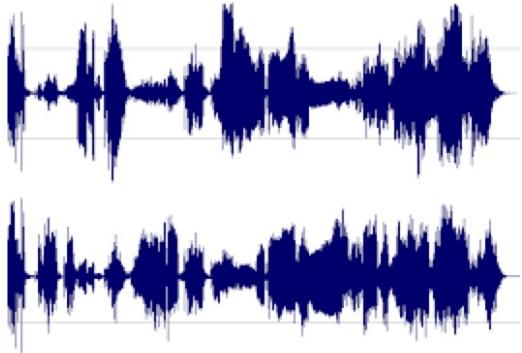
Each word in a document is represented by a list of n

We'll see later what these numbers are.)

A document is a row of numbers that represent the words occurring in the document.

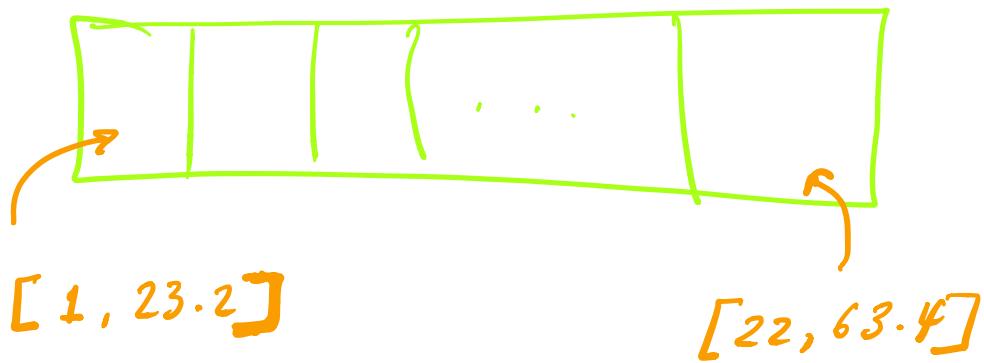


Nuts and Bolts (4)



An audio Stream  
is a row of  
numbers.

For example, these numbers can  
be [time, amplitude] pairs.



Nuts and Bolts ⑤

Column n

1 LastName	2 Sex	3 Age	4 Weight	5 Smoker	6 BloodPressure	7 Trials
'SMITH'	Male	38	176	1	124	93 18
'JOHNSON'	Male	43	163	0	109	77 [11,13,22]
'WILLIAMS'	Female	38	131	0	125	83 []
'JONES'	Female	40	133	0	117	75 [6,12]
'BROWN'	Female	49	119	0	122	80 [14,23]
'DAVIS'	Female	46	142	0	121	70 19
'MILLER'	Female	33	142	1	130	88 13
'WILSON'	Male	40	180	0	115	82 []
'MOORE'	Male	28	183	0	115	78 2
'TAYLOR'	Female	31	132	0	118	86 11

$m \times n$  table  
 ↗      ↖  
 rows      columns

A spreadsheet of data is a bunch of rows and columns of numbers.

Words like "Male" and "Female"  
 are encoded as numbers.

Similarly, Smoker = 1  
 NonSmoker = 0

## Inputs / Features

Nuts and Bolts (6)

f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>5</sub>	f <sub>6</sub>	Output

m  
rows

A dataset  
is a spreadsheet  
of  $m \times n$   
numbers

n inputs per row (here  $n=6$ )

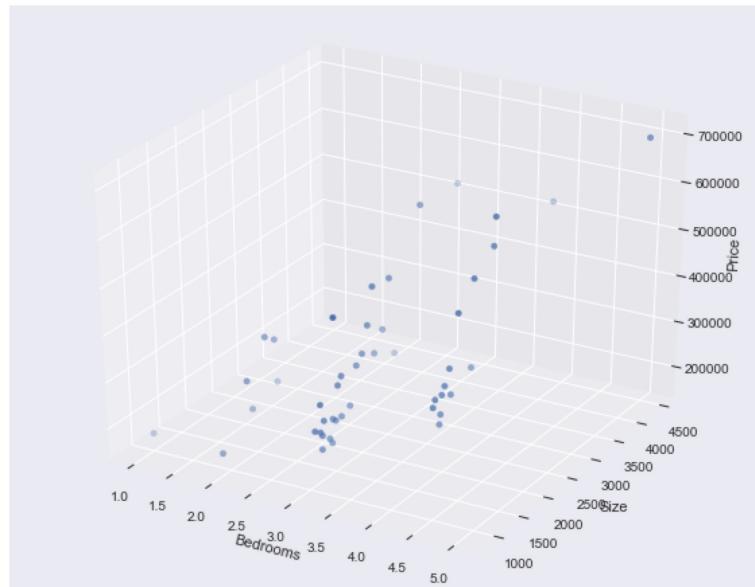
- Inputs are called "features"
- Inputs & outputs are always numbers.
  - They can be integers - 0, 1, 2, etc.
  - They can be reals - 0.5, 26.4, etc.
  - They can be positive or negative

Nuts and Bolts (7)

## Tasks

---

- Predict a number
- Predict a Category / Class



House Prices in Portland

TASK: Predict the price of a  
house not in this dataset.

Nuts and Bolts ⑧

How do we do this task?

Let's start with the data.

Feature 1	Feature 2	Output/target
Num of Br rooms	Size in sq ft.	Price
4	2500	\$350,000
5	3400	\$380,000

Task = Predict output based on features.

## Nuts and Bolts ⑨

Bedrooms	Sq.ft.	Price
$x_1^{(1)}$	$x_2^{(1)}$	$y^{(1)}$
$x_1^{(2)}$	$x_2^{(2)}$	$y^{(2)}$
$x_1^{(3)}$	$x_2^{(3)}$	$y^{(3)}$
$x_1^{(4)}$	$x_2^{(4)}$	$y^{(4)}$
.	.	.

(1)      ← row #

Notation

$x_1 \leftarrow$  feature #

This is the value of  
the first feature on  
the first row of the  
dataset.

What's the notation for the second feature  
in the sixteenth row of the dataset?

## Nuts and Bolts (10)

To predict the price, we're going to pretend that the price can be constructed by adding and multiplying together some numbers.

$$(w_1 \times \# \text{ bedrooms}) + (w_2 \times \underline{\text{sqr. ft.}}) = \text{price}$$

Parameters                                      Output

↓  
features

*Nuts and Bolts* (11)

Actually, for mathematical reasons,  
it's like this:

$$w_0 x_0 + w_1 x_1 + w_2 x_2 = y$$

Where  $w_0$  is called the "intercept" value. [We don't have to worry about it.]

$x_0$  is always = 1

Let's look at this from  
the standpoint of the  
dataset table.

Nuts and Bolts (12)

$w_0$	$x_0$	$w_1$	$x_1$	$w_2$	$x_2$	$y$

we've expanded the table by adding columns!

Our dataset now looks like this

row 1	$w_0$	$x_0^{(1)}$	$w_1$	$x_1^{(1)}$	$w_2$	$x_2^{(1)}$	$y^{(1)}$
row 2	$w_0$	$x_0^{(2)}$	$w_1$	$x_1^{(2)}$	$w_2$	$x_2^{(2)}$	$y^{(2)}$
row 3	$w_0$	$x_0^{(3)}$	$w_1$	$x_1^{(3)}$	$w_2$	$x_2^{(3)}$	$y^{(3)}$
row 4	$w_0$	$x_0^{(4)}$	$w_1$	$x_1^{(4)}$	$w_2$	$x_2^{(4)}$	$y^{(4)}$
.	.	.	.	.	.	.	.
row $m$	$w_0$	$x_0^{(m)}$	$w_1$	$x_1^{(m)}$	$w_2$	$x_2^{(m)}$	$y^{(m)}$

## Nuts and Bolts (13)

For each row, we're going to say:

$$\begin{aligned}
 w_0 x_0^{(1)} + w_1 x_1^{(1)} + w_2 x_2^{(1)} &= \hat{y}^{(1)} \\
 w_0 x_0^{(2)} + w_1 x_1^{(2)} + w_2 x_2^{(2)} &= \hat{y}^{(2)} \\
 w_0 x_0^{(3)} + w_1 x_1^{(3)} + w_2 x_2^{(3)} &= \hat{y}^{(3)} \\
 w_0 x_0^{(4)} + w_1 x_1^{(4)} + w_2 x_2^{(4)} &= \hat{y}^{(4)}
 \end{aligned}$$

Notice: While  
 the  $x_1$ s and  
 $x_2$ s and  
 $\hat{y}$ s are  
 different in  
 each row,  
 $w_0$ ,  $w_1$ , and  
 $w_2$  are

$w_0$ ,  $w_1$ , and  $w_2$  do not  
 change from row to row.

the SAME  
 in every row.

OK, let's get back to our data table and look at the first row.

## Nuts and Bolts (14)

	$w_0$	$x_0^{(1)}$	$w_1$	$x^{(1)}$	$w_2$	$x_2^{(1)}$	Size ↓	Price ↓	Predicted Price ↓
?	1	?	?	4	?	?	2500	350000	?

always = 1

(A) we'd like to calculate  $\hat{y}^{(1)}$  - the Predicted price by calculating:

$$\hat{y}^{(1)} = (w_0 \times x_0^{(1)}) + (w_1 \times x_1^{(1)}) + (w_2 \times x_2^{(1)})$$

(B) we'd like  $\hat{y}^{(1)}$  to be as close to  $y^{(1)}$  as possible

Given (A) and (B) what should the values of  $w_0, w_1$ , and  $w_2$  be?

First Try

$$(w_0 \times 1) + (w_1 \times 4) + (w_2 \times 2500) = 350000$$

In other words: Make the prediction exactly equal to the actual price of the house.

Great! We're focused on minimizing the difference between the predicted and the actual price.

But this still leaves us with

too many options for the values of  $w_0$ ,  $w_1$ , and  $w_2$

## Solution / Approach

Just guess the values  
of  $w_0, w_1, w_2$ .

How about

$$w_0 = -10$$

$$w_1 = 500$$

$$w_2 = 120.5$$

Nuts and Bolts (17)

This gives us a predicted price

$$\hat{y}^{(1)} = 318,440$$

Compared to  $\hat{y}^{(1)}$  - actual

price: 350,000

Not bad. A little lower

than the actual price.

How good is this prediction?

Nuts and Bolts (18)

The Thinking How much should we be penalized for this incorrect prediction?

Start With

What is the "cost" of being wrong?

$$\text{Cost} = \frac{\text{Predicted Value}}{\text{Actual Value}}$$

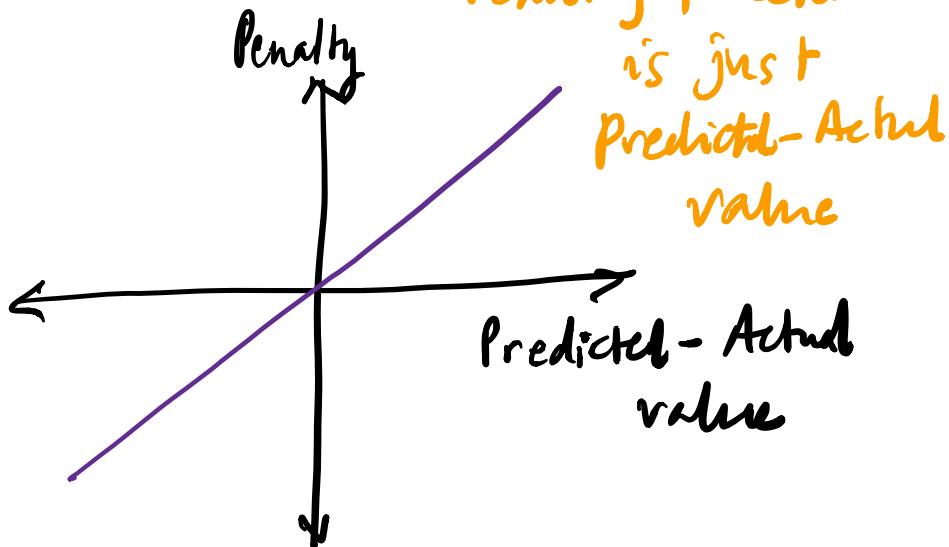
## Penalty Functions

How should we construct the penalty? This is a choice we can make / should make based on our knowledge of what's at stake when we make predictions.

## A SELECTION OF PENALTY FUNCTIONS

---

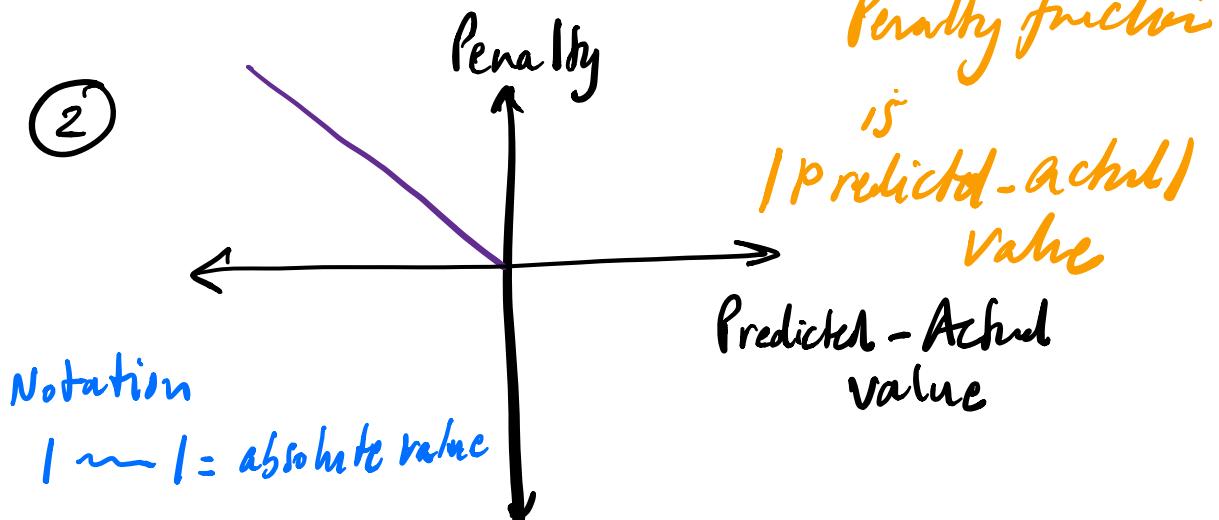
①

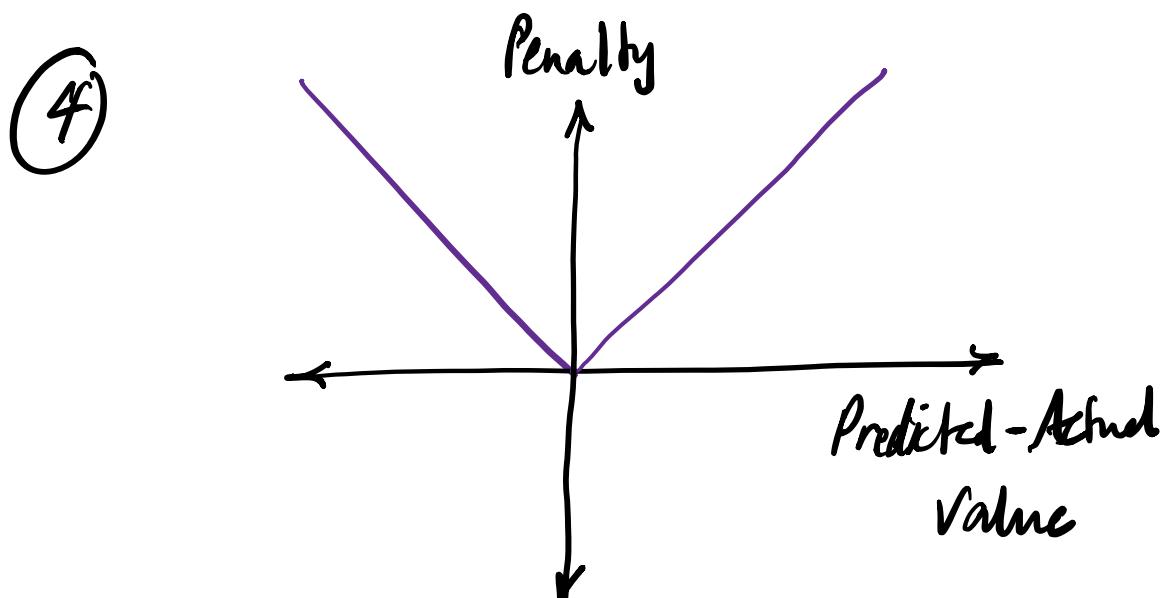
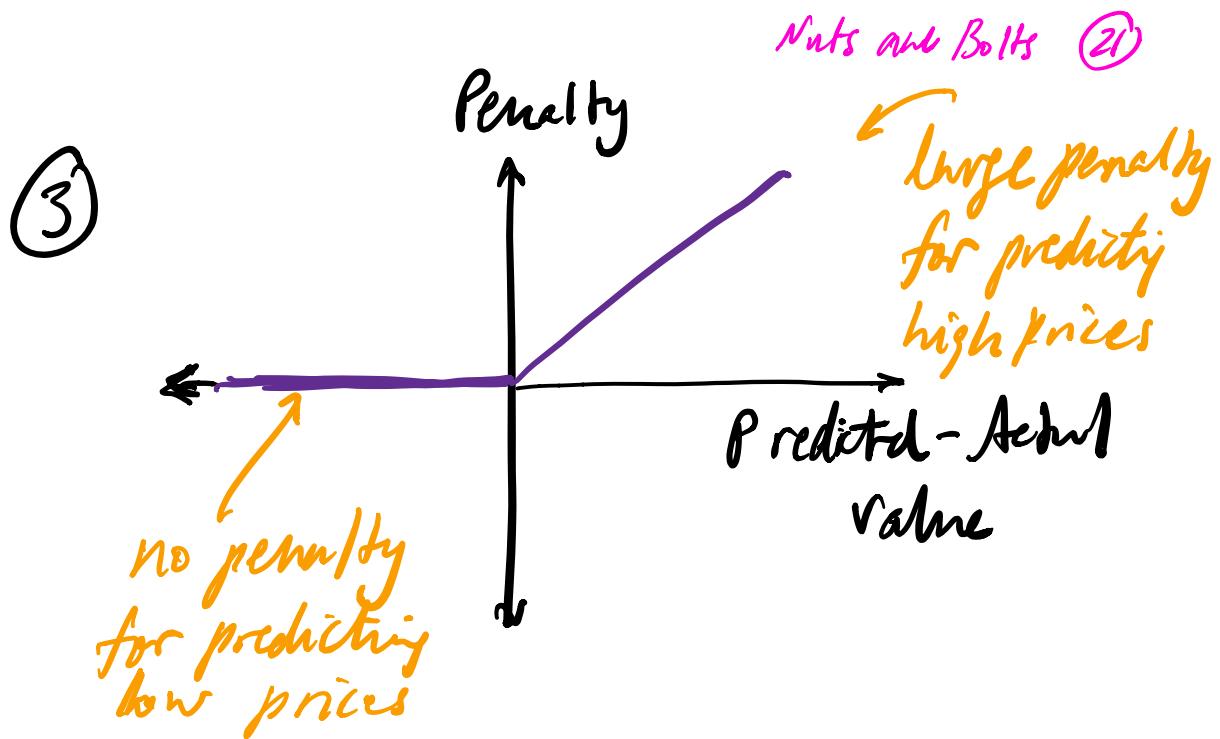


Is this a good penalty function?

Why or why not?

②

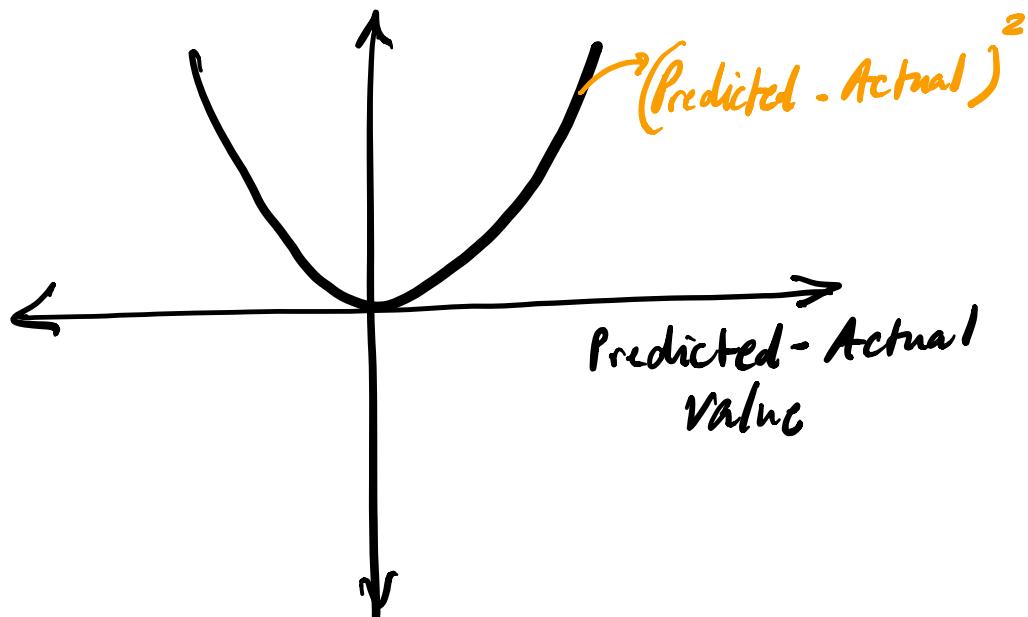




Describe this penalty function.  
Is it reasonable?

Nuts and Bolts (22)

Experience has taught us that some penalty functions work better than others. In particular, one that we'll see a lot of is this one:



$$\text{Penalty} = (\text{Predicted} - \text{Actual})^2$$

[Exercise]

Create 3 penalty functions we have not yet seen. For each one, describe it in a sentence or two.

## Nuts and Bolts (24)

Let's go back to our data table and expand it to include one more

column.

	Parameters			features		Actual	Predicted	Penalty	
	$w_0$	$x_0$	$w_1$	$x_1$	$w_2$	$x_2$	$y$	$\hat{y}$	
row1	$w_0$	1	$w_1$				$y$	$\hat{y}$	
row2	$w_0$	1	$w_1$				$y$	$\hat{y}$	
:	$w_0$	1	$w_1$				$y$	$\hat{y}$	
:	$w_0$	1	$w_1$				$y$	$\hat{y}$	
Row $m$	$w_0$	1	$w_1$				$y$	$\hat{y}$	

$x_0$  is always 1

□ The better the prediction the lower

the penalty.

□ The best prediction has the lowest / minimum penalty.

What does it mean to minimize the penalty?

Nuts and Bolts (25)

We have the values of  $w_0$ ,  $w_1$ , and  $w_2$  to set - they take on the same values for each row.

Let's say we pick the same values as before:  $w_0 = -10$   
 $w_1 = 500$   
 $w_2 = 120.5$

For these values, let's calculate the penalty for every row of the data table.

## Nuts and Bolts (26)

$w_0$	$x_0$	$w_1$	$x_1$	$w_2$	$x_2$	$y$	$\hat{y}$	Penalty
-10	$x_0^{(1)}$	500	$x_1^{(1)}$	120.5	$x_2^{(1)}$	$y^{(1)}$	$\hat{y}^{(1)}$	$p^{(1)}$
-10	$x_0^{(2)}$	500	$x_1^{(2)}$	120.5	$x_2^{(2)}$	$y^{(2)}$	$\hat{y}^{(2)}$	$p^{(2)}$
-10	$x_0^{(3)}$	500	$x_1^{(3)}$	120.5	$x_2^{(3)}$	$y^{(3)}$	$\hat{y}^{(3)}$	$p^{(3)}$
-10	$x_0^{(4)}$	500	$x_1^{(4)}$	120.5	$x_2^{(4)}$	$y^{(4)}$	$\hat{y}^{(4)}$	$p^{(4)}$
-10	$x_0^{(5)}$	500	$x_1^{(5)}$	120.5	$x_2^{(5)}$	$y^{(5)}$	$\hat{y}^{(5)}$	$p^{(5)}$
-10	$x_0^{(6)}$	500	$x_1^{(6)}$	120.5	$x_2^{(6)}$	$y^{(6)}$	$\hat{y}^{(6)}$	$p^{(6)}$

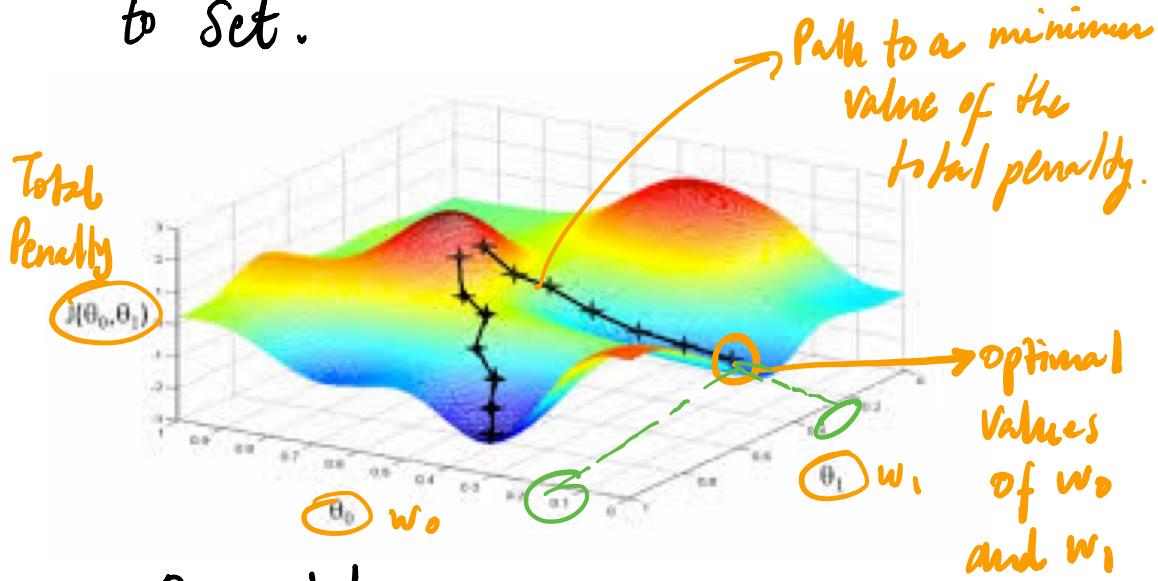
⋮

Sum of  
 $p^{(1)}, p^{(2)}, p^{(3)} \dots$

What We Want The values of

$w_0$ ,  $w_1$ , and  $w_2$  that minimizes  
 the sum of penalties.

Imagine we only had  $w_0$  and  $w_1$  to set. Nuts and Bolts (27)



$$\theta_0 = w_0$$

$$\theta_1 = w_1$$

NOTE The shape of this surface depends on the penalty function we choose.

For each set of  $w_0$  and  $w_1$  values, we plot the penalty (sum of all the penalties in the data set).

This gives us a surface that typically looks like hilly terrain.

Nuts and Bolts (28)

A machine learning algorithm is a recipe for finding the path that leads to the minimum value of the total penalty.

The values of  $w_0$  and  $w_1$  at the minimum value of the total penalty are the optimal parameter values.

The computer program has learned these parameter values from experience.

## Recap

- 1) Experience = Data (the complete dataset)
- 2) Task = Prediction
- 3) Performance = Penalty

best performance = minimize  
the sum of  
the penalties  
across the  
entire  
dataset.

- 4) The way to get to the best performance values for  $w_0$ ,  $w_1$ , and  $w_2$  is to use a machine learning algorithm.

The machine learning algorithm that does this job is simple to write down.

- 1) Pick any set of initial values for  $w_0$ ,  $w_1$ , and  $w_2$ .
- 2) Find the total penalty of the data set.
- 3) Look all around  $w_0$ ,  $w_1$ , and  $w_2$ . Find close values that result in a lower total penalty. If none of the close values result in a lower penalty, STOP.
- 4) Repeat 3.

## Recap (Continued)

Nuts and Bolts (31)

- 5) Machine learning is learning from experience and feedback without having an explicit set of rules.
  - hitting a forehand
  - finding a face in a picture
  - riding a bicycle
- 6) You don't need big data to do machine learning. (But it can help for some problems.)
- 7) Machine learning = giant numerical optimization problem.