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?

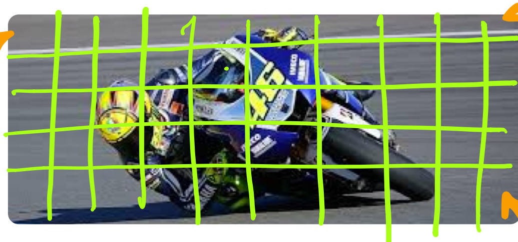
Experience ≠ Rules

Experience = Data

$\boxed{\text{Data} = \text{Table of Numbers}}$

An image is a grid of pixels

P el 1



Pixel 10

Pixel 50

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

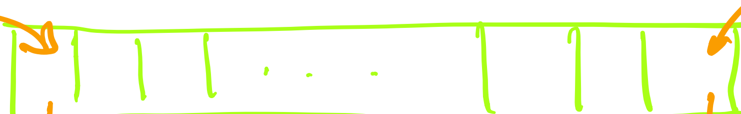
red value

green Value

blue value

An image is a row of Pixel Values = a row of numbers

Pi 1

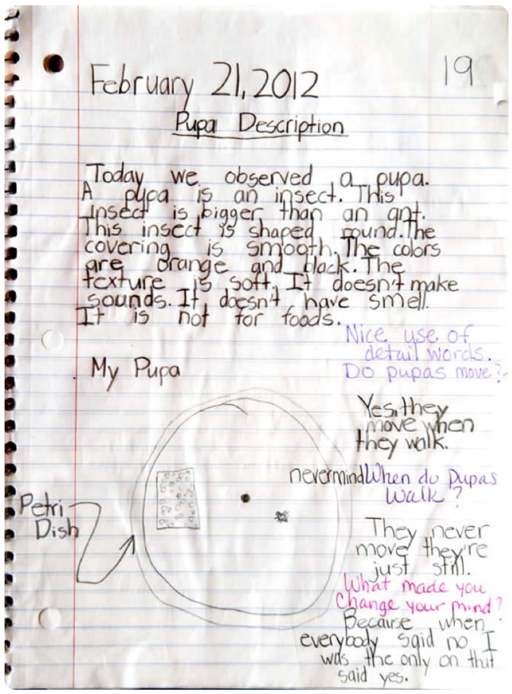


Pixel 50

Value [22, 105, 36]

Value [255, 0, 108]

chally : a row of lists of numbers


February 21, 2012 19

Pupa Description

Today we observed a pupa. A pupa is an insect. This insect is bigger than an ant. This insect is shaped round. The covering is smooth. The colors are orange and black. The texture is soft. It doesn't make sounds. It doesn't have smell. It is not for foods.

Nice use of detail words. Do pupas move?

My Pupa

Yes, they move when they walk.

nevermind When do Pupas walk?

They never move they're just still.

What made you change your mind?

Because when everybody said no I was the only on that said yes.

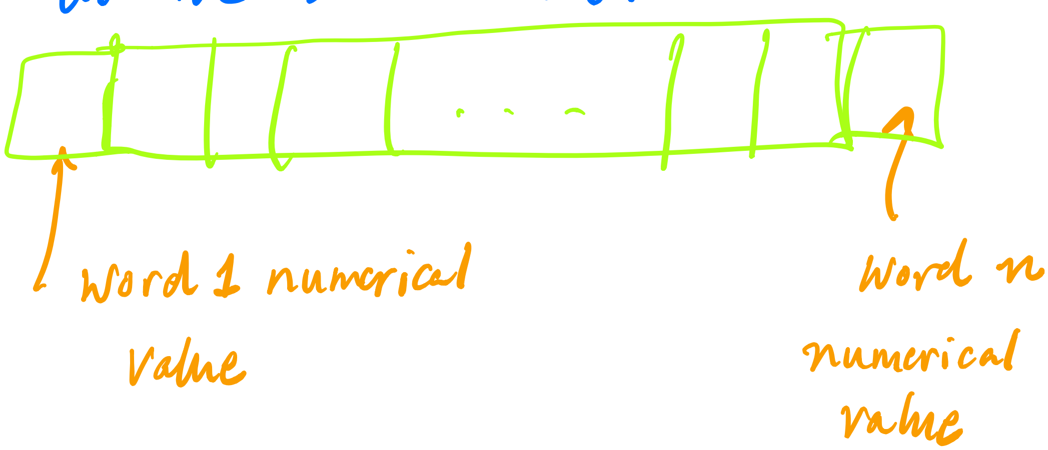
Petri Dish

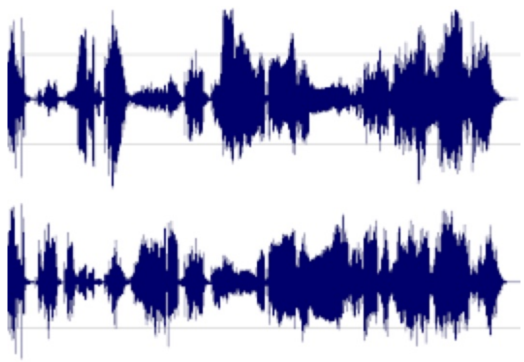
A document is a row of numbers.

Each word in a document is represented by a number.

We'll see later what these numbers are :)

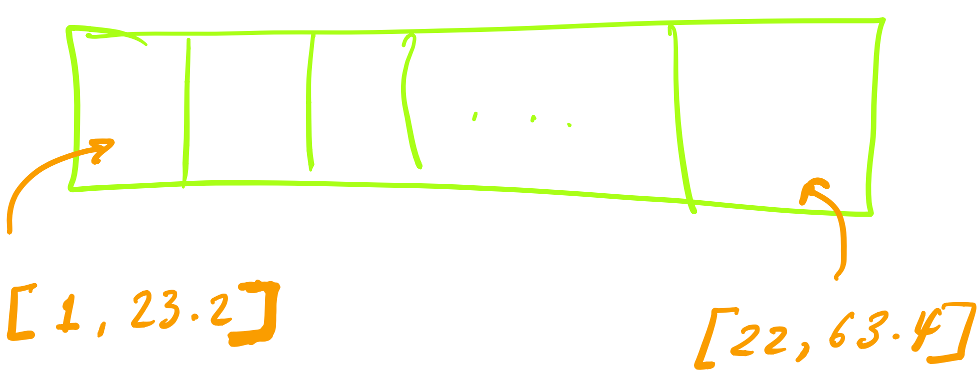
A document is a row of numbers that represent the words occurring in the document.





An audio stream is a row of numbers.

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



Column 1 Column n

row 1

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

row m

m x n table

rows columns

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

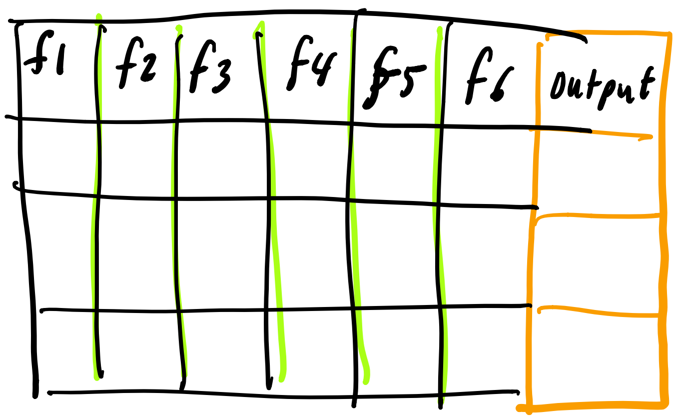
Words like "Male" and "Female" are encoded as numbers.

Similarly, Smoker = 1

NonSmoker = 0

Inputs / Features

m rows



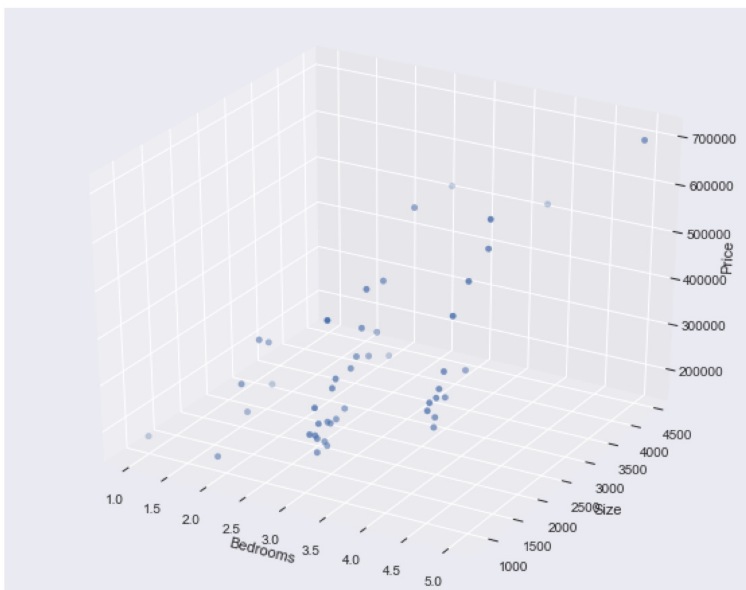
A dataset is a spreadsheet of $m \times n$ numbers

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

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

Tasks

- ☐ Predict a number
- ☐ Predict a Category / Class



House Prices in Portland

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

How do we do this task?

Let's start with the data.

Feature 1   Feature 2   Output/target

Num of Be rooms	Size in sq ft.	Price
4	2500	$350,000
5	3400	$380,000

Task = Predict output based on features.

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

Notation

row #

$$\underbrace{x_1^{(1)}}_{}$$

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 data set?

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

Like this

Parameters

Output

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

features

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.

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	$x_2^{(4)}$	$y^{(4)}$
$\vdots$		$\cdot$		$\vdots$	$\vdots$	$\vdots$	$\vdots$
row m	w_0	$x_0^{(m)}$	w_1	$x_1^{(m)}$	w_2	$x_2^{(m)}$	$y^{(m)}$

For each row, we're going to say:

$$w_0 x_0^{(1)} + w_1 x_1^{(1)} + w_2 x_2^{(1)} = \hat{y}^{(1)}$$
$$w_0 x_0^{(2)} + w_1 x_1^{(2)} + w_2 x_2^{(2)} = \hat{y}^{(2)}$$
$$w_0 x_0^{(3)} + w_1 x_1^{(3)} \quad w_2 x_2^{(3)} = \hat{y}^{(3)}$$
$$w_0 x_0^{(4)} + w_1 x_1^{(4)} \quad w_2 x_2^{(4)} = \hat{y}^{(4)}$$
$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

Notice: while the x_1s and x_2s and $\hat{y}$s are different in each row, w_0, w_1, and w_2 are the SAME in every row.

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

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

			# bedrooms		Size	Price	Predicted Price
w_0	$x_0^{(1)}$	w_1	$x_1^{(1)}$	w_2	$x_2^{(1)}$	$y^{(1)}$	$\hat{y}^{(1)}$
?	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

$$W_0 = -10$$

$$W_1 = 500$$

$$W_2 = 120.5$$

This gives us a predicted price

$\hat{y}^{(1)} = 318,40$

Compared to $y^{(1)}$ – actual price: 350,000

Not bad. A little lower than the actual price.

How <u>good</u> is this prediction?

<u>The Thinking</u> How much should we be penalized for this incorrect prediction?

<u>Start with</u>

What is the "cost" of being wrong?

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

Penalty Functions

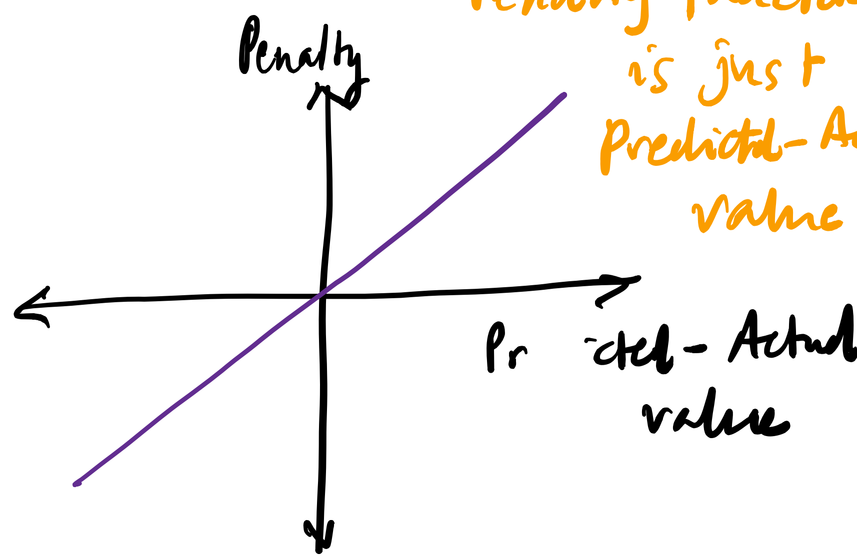
See p. 88 of Provost & Fawcett on the choice of penalty/objective functions.

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.

PENALTY FUNCTIONS A SELECTION

①

Penalty function is just Predicted - Actual value

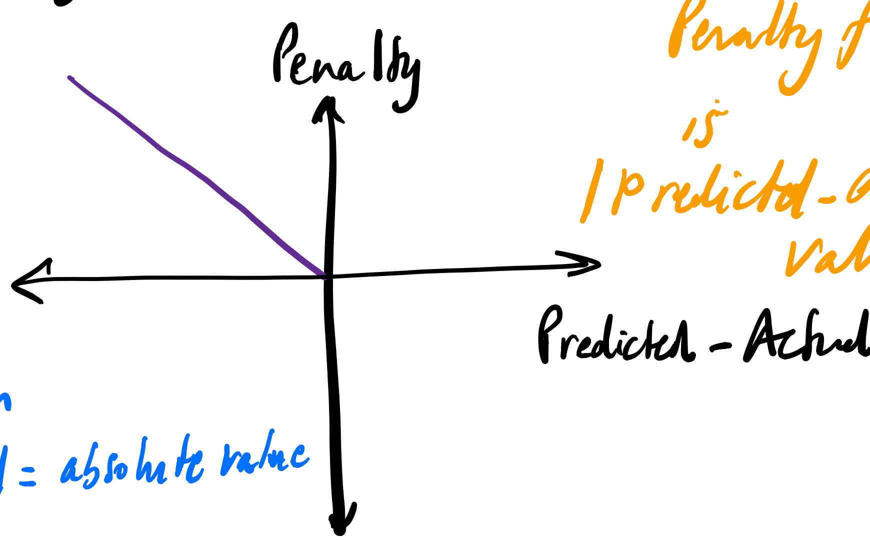


Is this a good penalty function?

Why or why not?

②

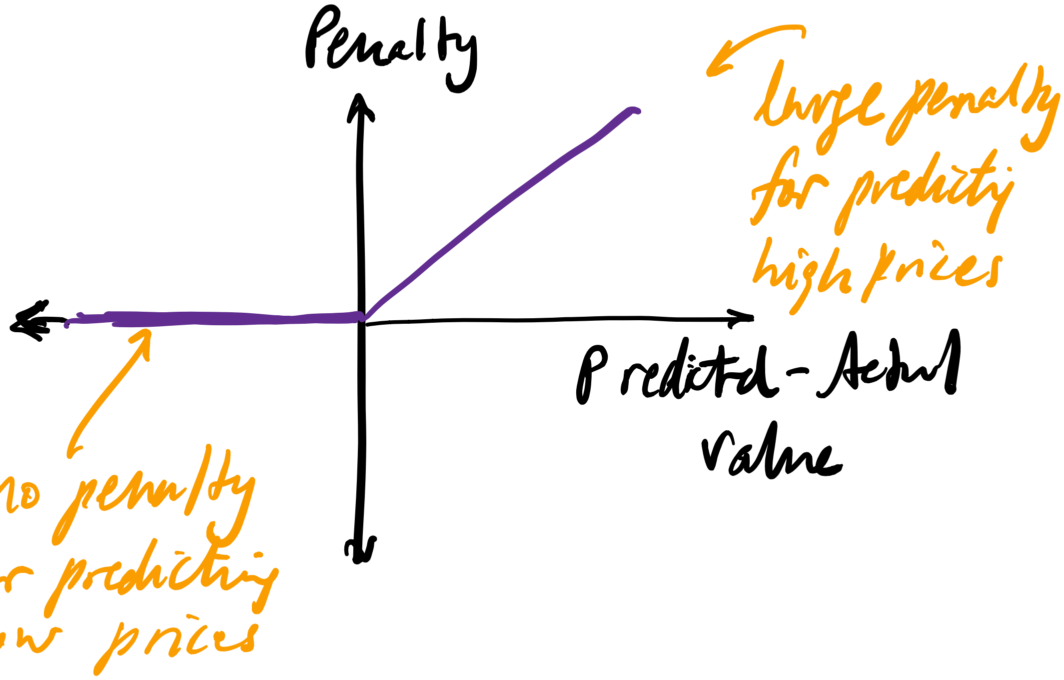
Penalty function is |Predicted - actual| value



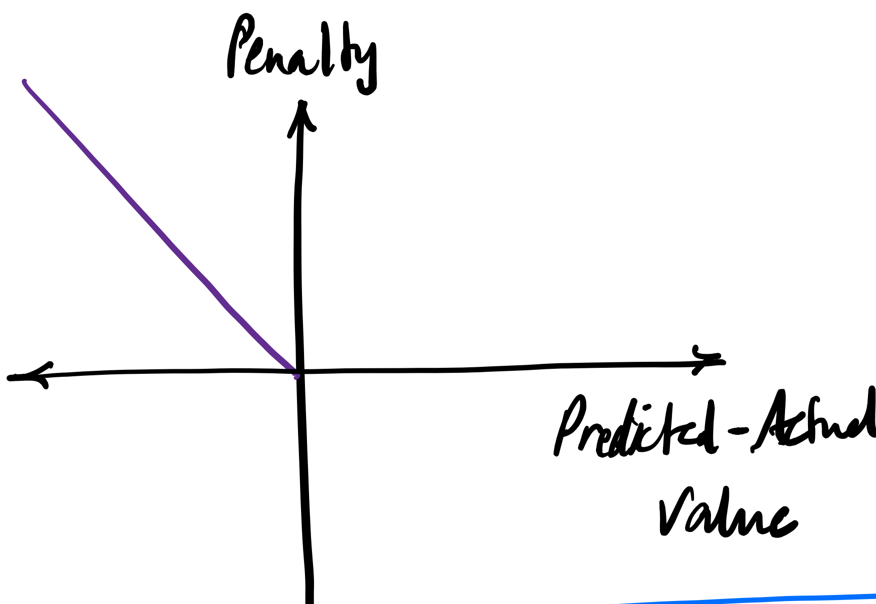
Notation

$| \sim |$ = absolute value

③

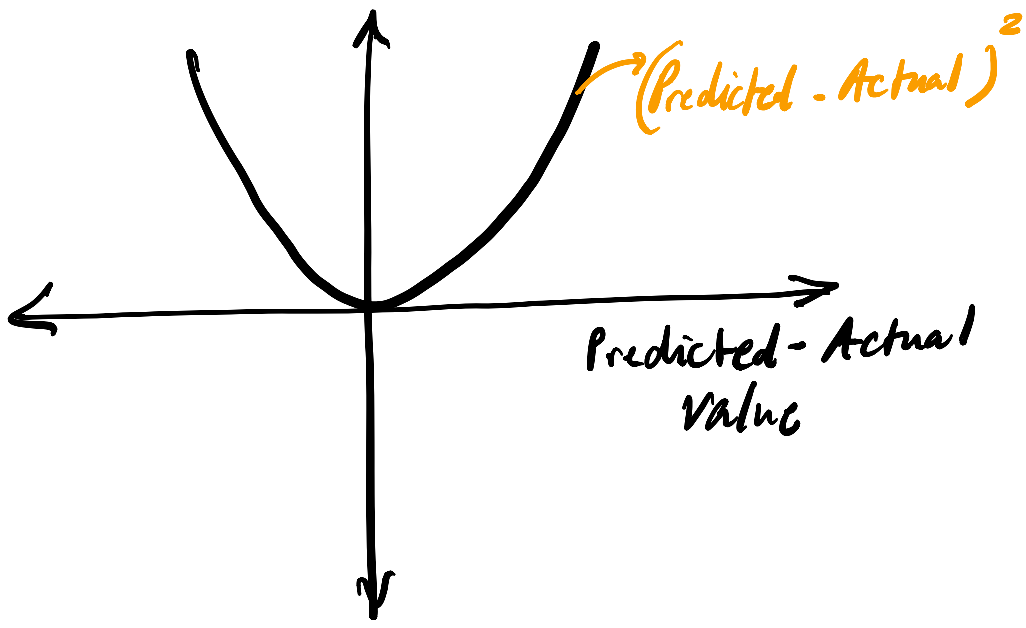


④



Describe this penalty function.
Is it reasonable?

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



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

$\boxed{\text{Exercise}}$ Create 3 penalty functions we have not yet seen. For each one, describe it in a sentence or two.

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

Parameters   features   Actual   Predicted

	w_0	x_0	w_1	x_1	w_2	x_2	y	$\hat{y}$	Penalty
row1		1							
row2		1							
⋮		1							
		1							
row m		1							

x_0 is always 1

- ☐ The better the prediction the lower the penalty.
- ☐ The best prediction has the <u>lowest</u> / <u>minimum</u> penalty.

What does it mean to minimize the penalty?

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 <u>every</u> row of the data table.

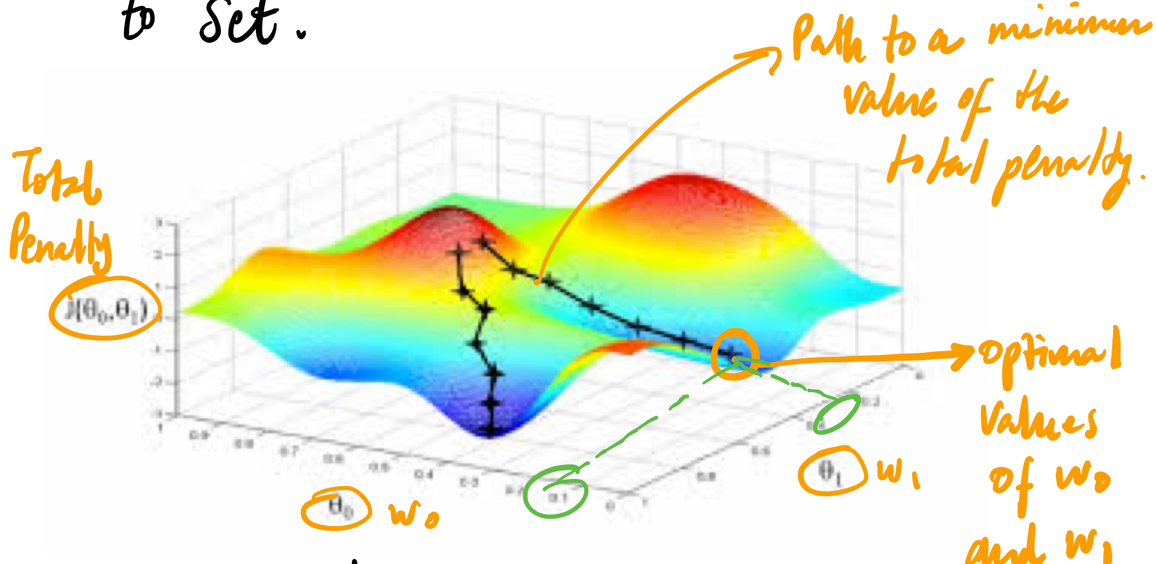
w_0	x_0	w_1	x_1	w_2	x_2	y	$\hat{y}$	Penalty
-10	$x_0^{(1)}$	500	$x_1^{(1)}$	120.5	$x_2^{(1)}$	$y^{(1)}$	$\hat{y}^{(1)}$	$p^{(1)}$
-10	$x_0^{(2)}$	500	$x_1^{(2)}$	120.5	$x_2^{(2)}$	$y^{(2)}$	$\hat{y}^{(2)}$	$p^{(2)}$
-10	$x_0^{(3)}$	500	$x_1^{(3)}$	120.5	$x_2^{(3)}$	$y^{(3)}$	$\hat{y}^{(3)}$	$p^{(3)}$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

$\vdots$

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

<u>What We Want</u> 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.



$\theta_0 = w_0$

$\theta_1 = w_1$

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.

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 <u>optimal</u> parameter values.

The computer program has <u>learned</u> these parameter values from experience.

Recap

1) Experience = Data (the complete data set)

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)

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.