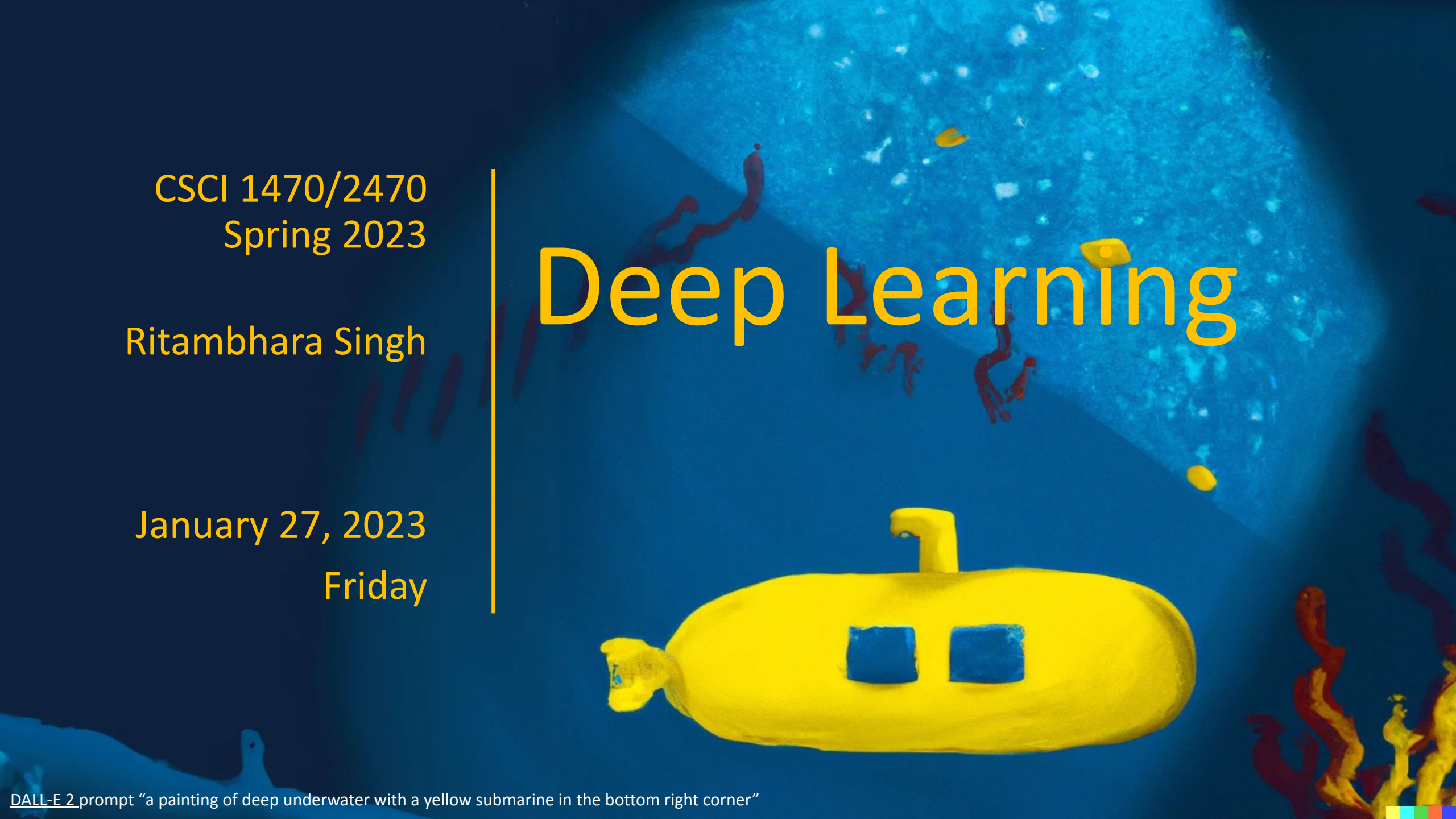
CSCI 1470/2470
Spring 2023

Ritambhara Singh

January 27, 2023

Friday

Deep Learning



DALL-E 2 prompt “a painting of deep underwater with a yellow submarine in the bottom right corner”

What do you hope to learn/be able to do by the end of this course?

~140 Responses! ☺

I hope to strengthen my abilities in computer science and apply deep learning in my future computation biology career. I want to learn about the difference between machine learning and deep learning as well.

I want to finish some useful projects during this course and hope to gain a general understanding of deep

Get a fundamental knowledge of major topics and emerging methods/tools in Deep Learning

Apply deep learning to the research I'm currently doing

Use deep learning for research in computer graphics and natural language processing.

make something cool :)

What is deep learning? How is it related to machine learning? How is it applied to perform tasks like classifying images or translating languages?

Be able to make a project from scratch using dl

What do you hope to learn/be able to do by the end of this course?

employment :)

understand chatGPT

APPLY DEEP LEARNING TO PROBLEMS I ENCOUNTER IN THE WILD

deep learning

I want to be able to code God.

Expedite the AI robot takeover.

What do you hope to learn/be able to do by the end of this course?

Major Themes

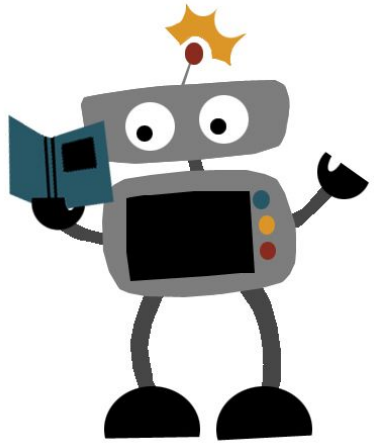
(1) Apply deep learning to real-world problems in various types of domains (NLP, Image, Biology) and/or current research

(2) Enhance understanding of concepts and mathematical background

(3) Develop (or improve) implementation skills

(4) Think critically about applying deep learning models and ethical considerations

Recap: What is Machine Learning?



Input: X

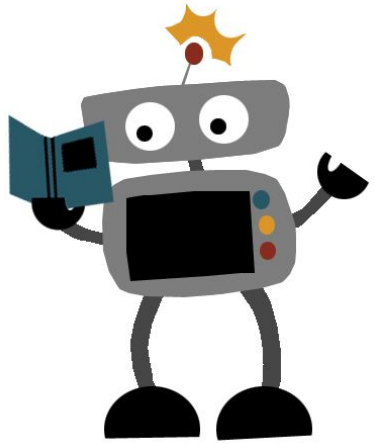
Output: Y

”Cooking?”

Function: f

f(X)□Y

Recap: What is Machine Learning?



Supervised Learning

Input: X

Learned function: f

Output: Y

”Cooking?”

$f(X) \rightarrow Y$

Today’s goal - Learn about some basic concepts of machine learning

(1) How do we represent input/output?

(2) Learning the function f

(3) Training a machine learning model

(4) Learning good models

How do we represent input/output?

Machines work with numbers!

How can we represent input image as numbers?

Input: X

"Model"

Function: f

f(X)□Y

Output: Y

"Cooking?"

How can we represent output labels as numbers?

How do we represent input/output?

Classification

Binary classification

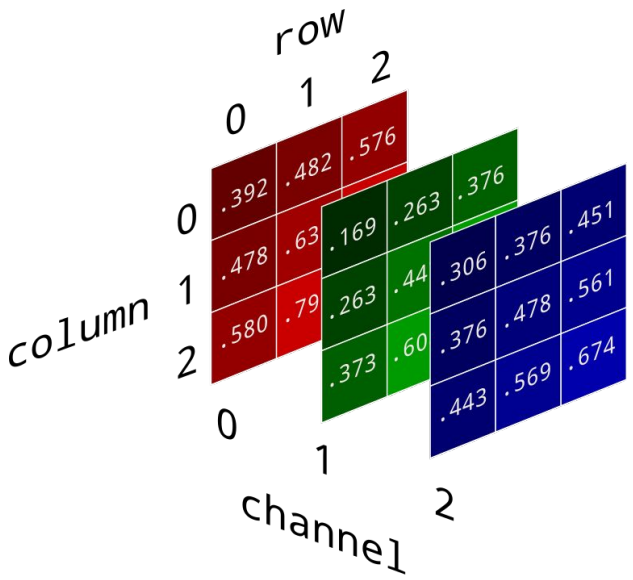
Input: X

Output: Y

"Cooking?"

Complicated input!

Let's use a simpler example!



$X \in \mathbb{R}^{H \times W \times 3}$

$Y \in \{0,1\}$

(Categorical output)

1 or 0

(Real number: a value of a continuous quantity)

Image courtesy: https://www.kdnuggets.com/2019/12/convert-rgb-image-grayscale.html

But first some notations…

$\mathbb{X}$: *A set of input data*

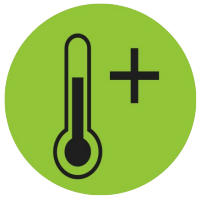
$\mathbb{Y}$: *Associated set of target values (outputs) for supervised learning*

$x^{(k)}$: k$^{\text{th}}$ example (input) from a dataset

$y^{(k)}$: *Target (output) associated with* $x^{(k)}$ *for supervised learning*

$\mathbb{R}$: A set of real numbers

Simpler example: How do we represent input/output?



Input: $\mathbb{X}$

"Temperature"

Regression

Target: $\mathbb{Y}$

"Profit made on selling lemonade"

$x^{(1)}$ 100.1 ℉

$x^{(2)}$ 80.0 ℉

$x^{(3)}$ 30.3 ℉

$\mathbb{X} \in \mathbb{R}$

Function: f

f(X)□Y

$y^{(1)}$ $200.0

$y^{(2)}$ $180.5

$y^{(3)}$ $115.1

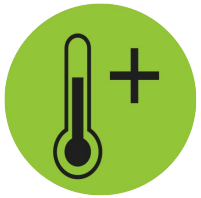
$\mathbb{Y} \in \mathbb{R}$

(Numerical output)

Do you see a trend here?

What is different about the output here?

Learning function f



Input: $\mathbb{X}$

"Temperature"

$x^{(1)} = 100.1$

$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

$\mathbb{X} \in \mathbb{R}$

Regression

Function: f

f(X)□Y

Target: $\mathbb{Y}$

"Profit made on selling lemonade"

$y^{(1)} = 200.0$

$y^{(2)} = 180.5$

$y^{(3)} = 115.1$

$\mathbb{Y} \in \mathbb{R}$

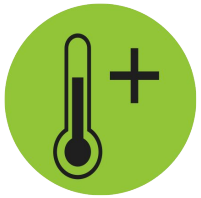
(Numerical output)

(Image only for explaining concept, not drawn accurately)

Learning function f

Have you seen this equation before?

Regression



Input: $\mathbb{X}$

"Temperature"

$x^{(1)} = 100.1$

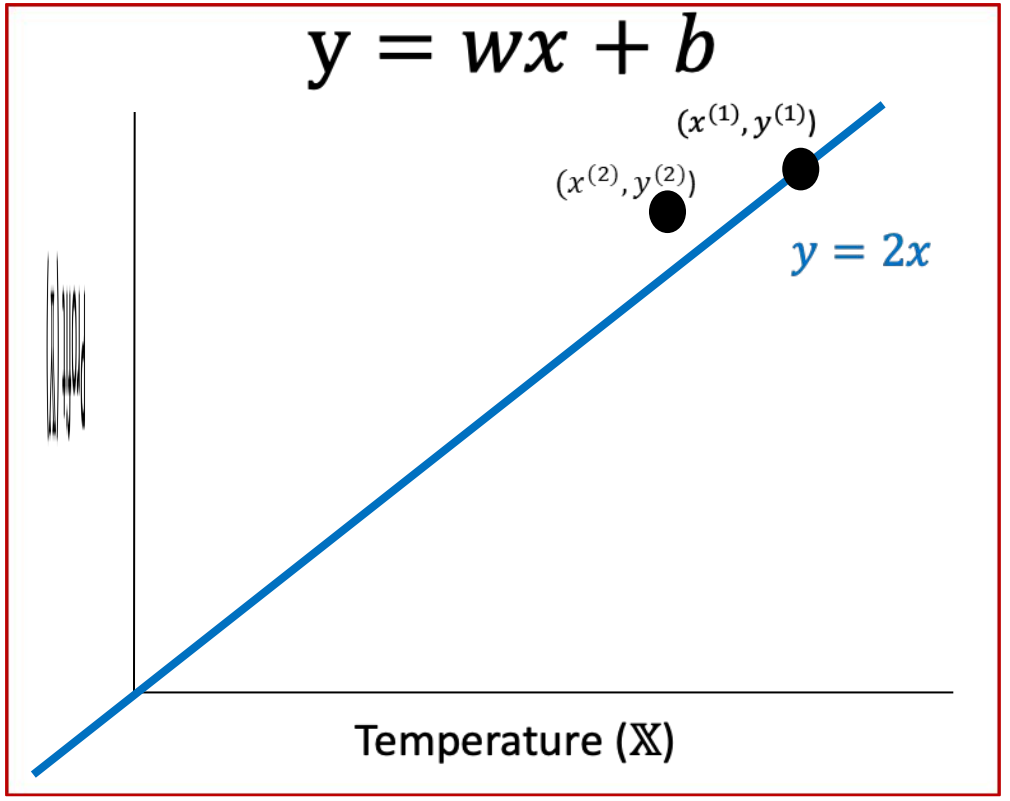
$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

$\mathbb{X} \in \mathbb{R}$

Linear function

$$y = wx + b$$



Target: $\mathbb{Y}$

"Profit made on selling lemonade"

$y^{(1)} = 200.0$

$y^{(2)} = 180.5$

$y^{(3)} = 115.1$

$\mathbb{Y} \in \mathbb{R}$

(Numerical output)

(Image only for explaining concept, not drawn accurately)

Learning function f

Input: $\mathbb{X}$

"Temperature"

Regression

Target: $\mathbb{Y}$

"Profit made on selling lemonade"

$x^{(1)} = 100.1$

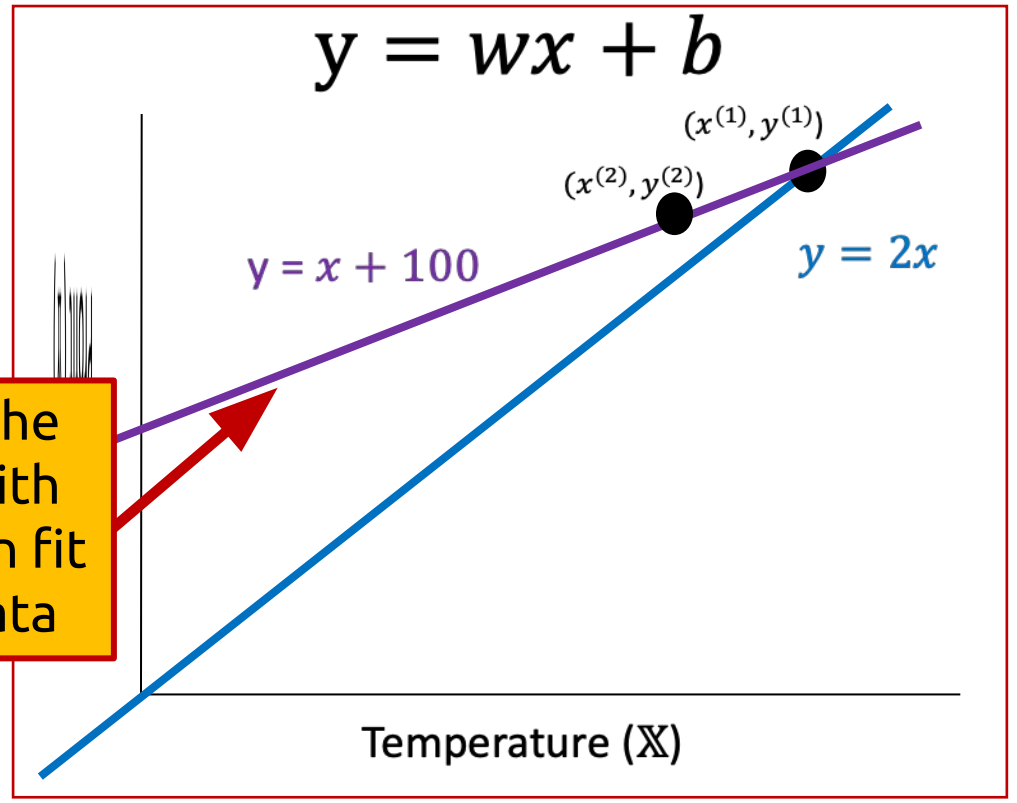
$x^{(2)} = 80.0$

$x^{(3)} =$

$\mathbb{X} \in \mathbb{R}$

Linear function

$$y = wx + b$$



Only the line with bias can fit the data

$y^{(1)} = 200.0$

$y^{(2)} = 180.5$

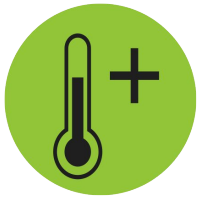
$y^{(3)} = 115.1$

$\mathbb{Y} \in \mathbb{R}$

(Numerical output)

(Image only for explaining concept, not drawn accurately)

Learning function f



Input: $\mathbb{X}$

"Temperature"

Regression

Target: $\mathbb{Y}$

"Profit made on selling lemonade"

$x^{(1)} = 100.1$

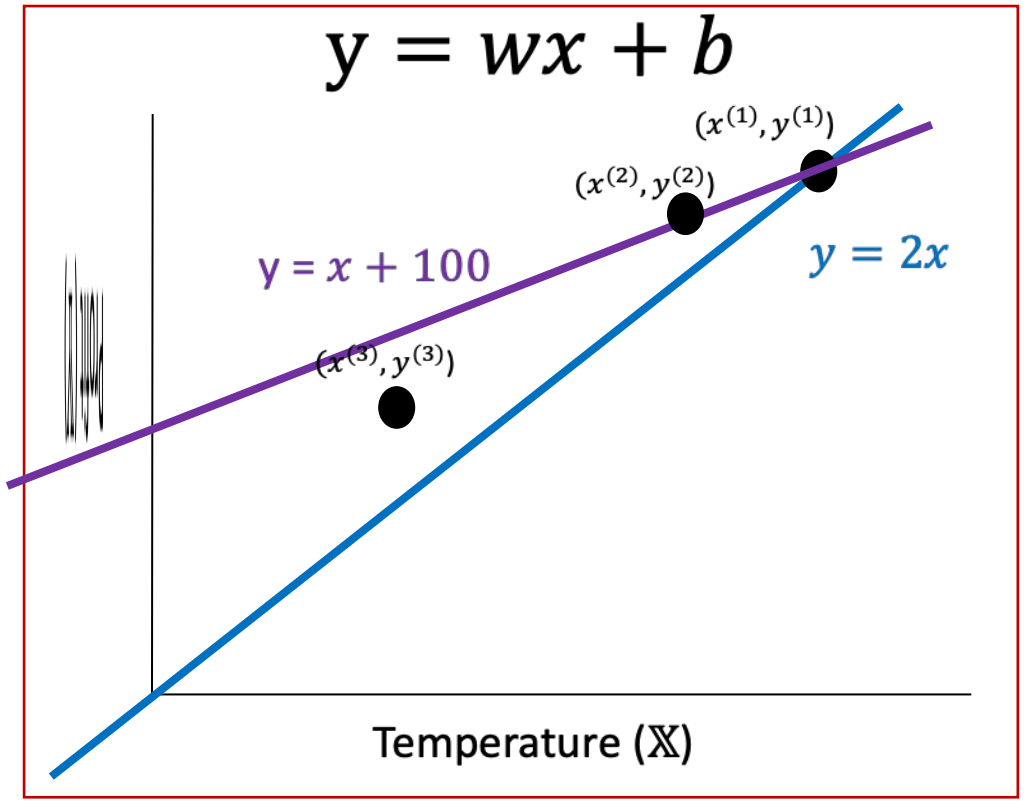
$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

$\mathbb{X} \in \mathbb{R}$

Linear function

$$y = wx + b$$



$y^{(1)} = 200.0$

$y^{(2)} = 180.5$

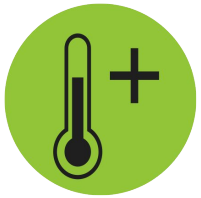
$y^{(3)} = 115.1$

$\mathbb{Y} \in \mathbb{R}$

(Numerical output)

(Image only for explaining concept, not drawn accurately)

Learning function f



Input: $\mathbb{X}$

"Temperature"

Regression

Target: $\mathbb{Y}$

"Profit made on selling lemonade"

Linear function

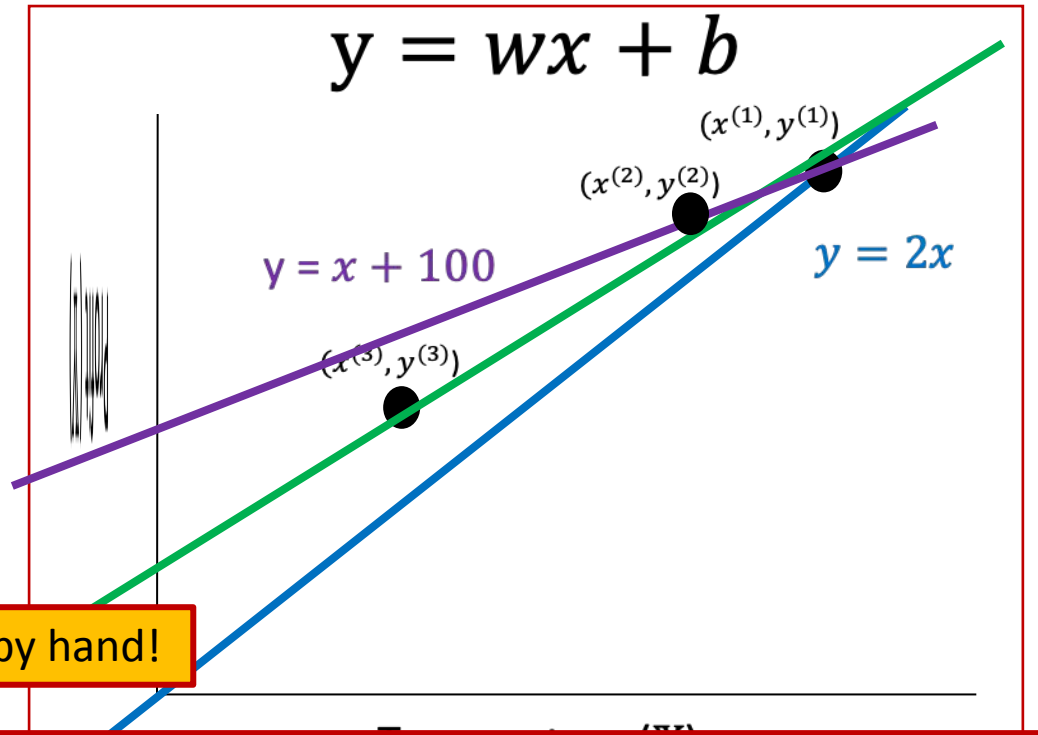
$$y = wx + b$$

$x^{(1)} = 100.1$

$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

$\mathbb{X} \in \mathbb{R}$



$y^{(1)} = 200.0$

$y^{(2)} = 180.5$

$y^{(3)} = 115.1$

$\mathbb{Y} \in \mathbb{R}$

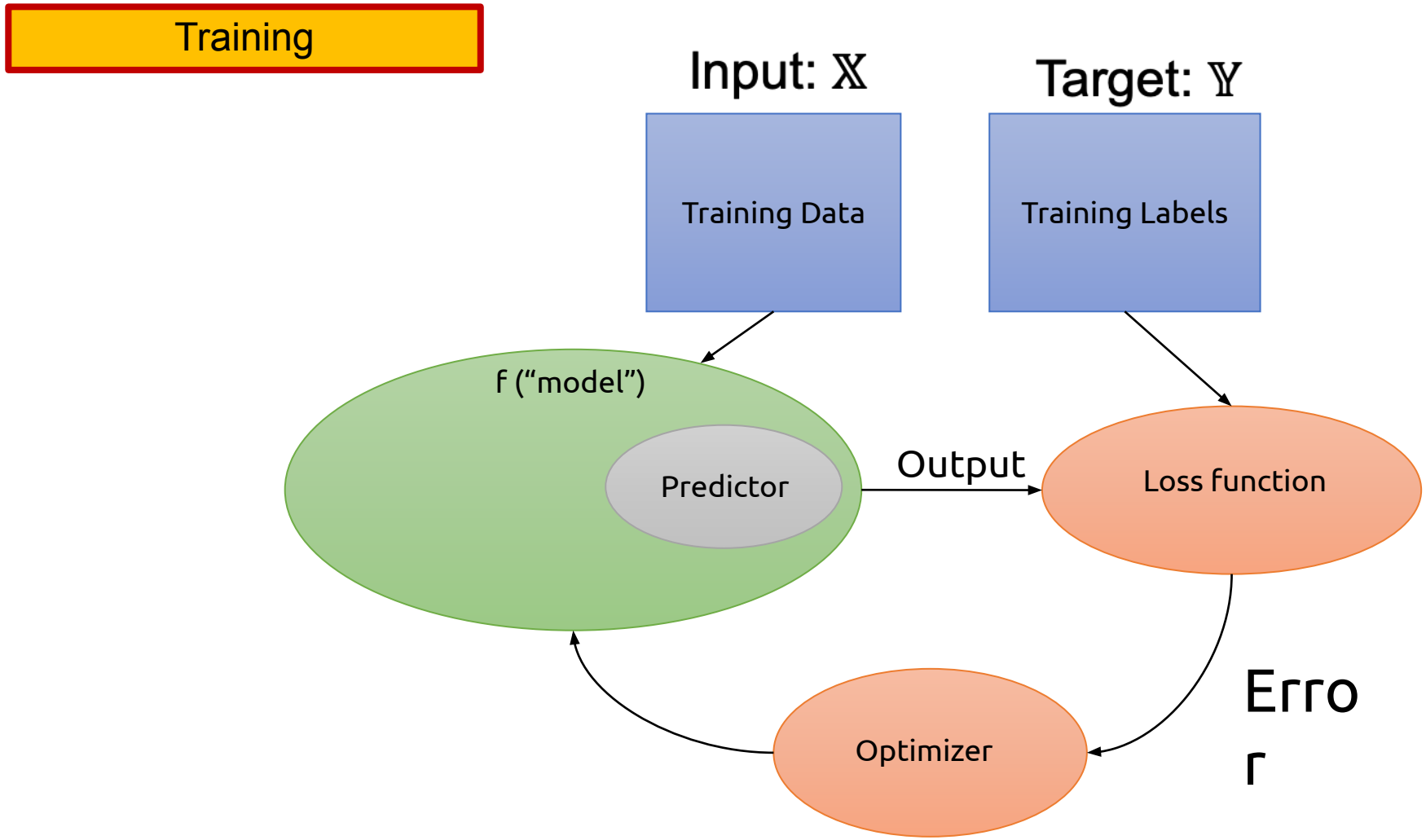
(Numerical output)

Very hard to learn these functions by hand!

Use machine learning to learn a good approximation of the function ***from data***

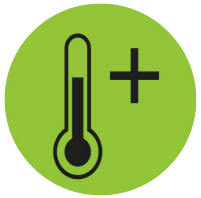
(Image only for explaining concept, not drawn accurately)

“Classic” Supervised Learning in Machine Learning



Any questions?

Testing our model

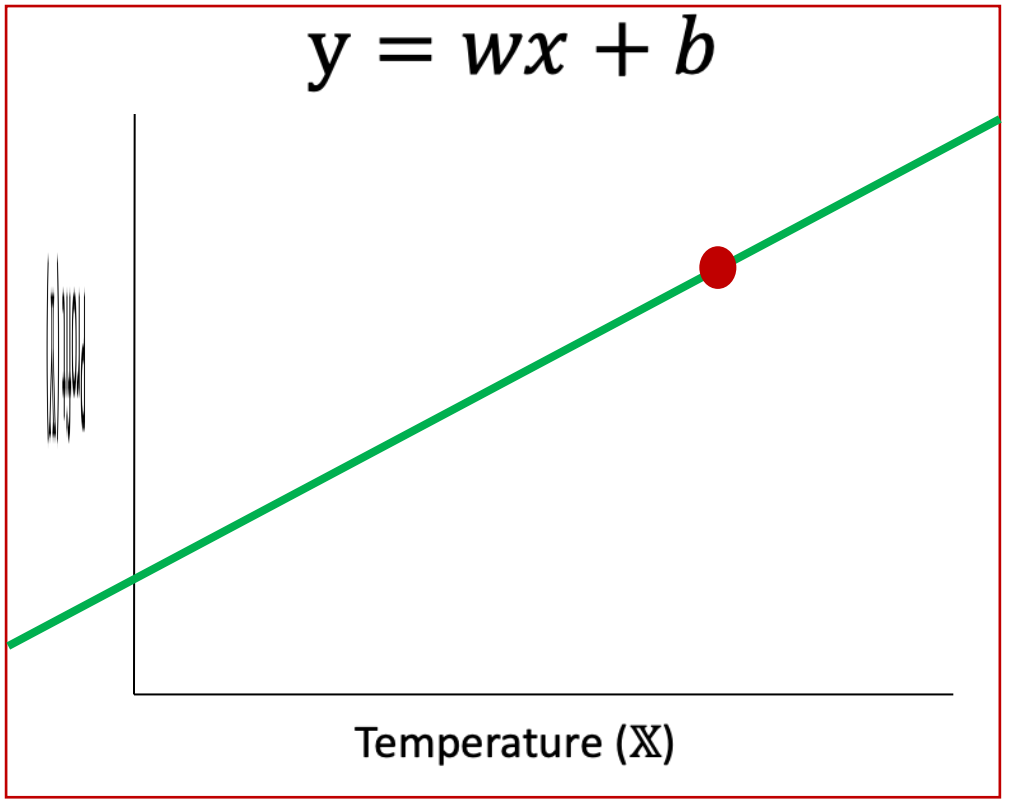


"Temperature"

"Profit made on selling lemonade"

Linear function

$$y = wx + b$$



$x' = 70$

Prediction

$y' = 175$

(Image only for explaining concept, not drawn accurately)

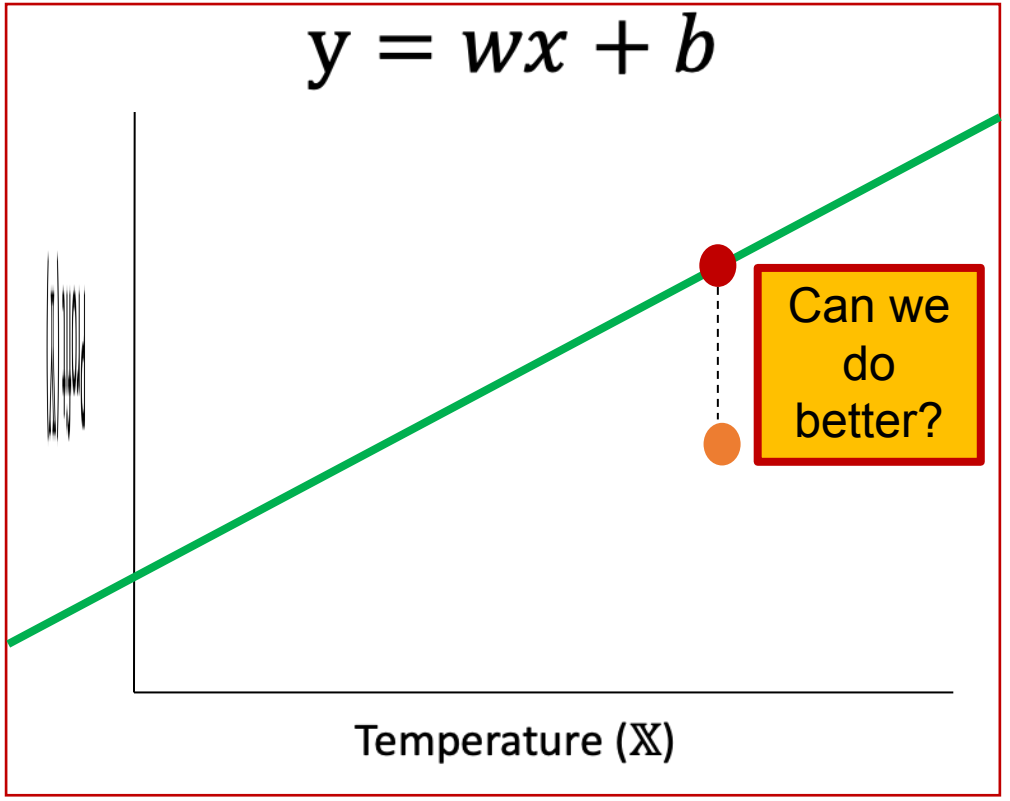
Testing our model

"Temperature"

"Profit made on selling lemonade"

Linear function

$$y = wx + b$$



$x' = 70$

$\hat{x} = 70$

Real-world deployment

Prediction

$y' = 175$

True observation

$\hat{y} = 140$

(Image only for explaining concept, not drawn accurately)

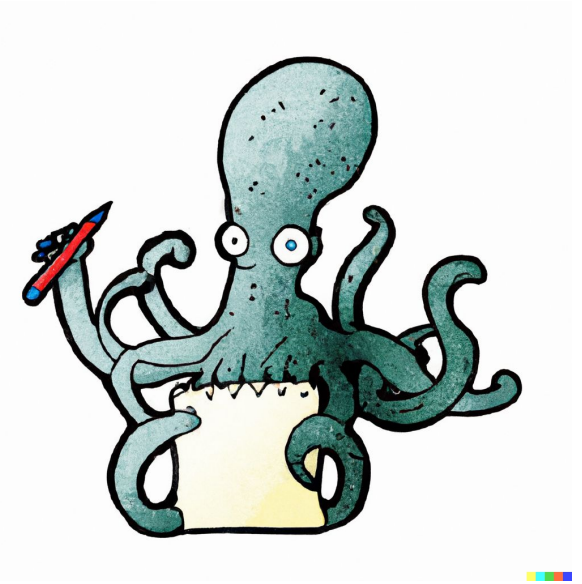
(An outlier is a data point that differs significantly from other observations)

Can we do better? – May be

How?

Go to **www.menti.com** and use the code **1587 1135**

Option 1: Collect more data and retrain
Option 2: Try a different function
Option 3: Do both 1 and 2



DALL-E 2 prompt “a cartoon octopus writing notes”

Learning better models – Collect more data

Input: $\mathbb{X}$

"Temperature"

$x^{(1)} = 100.1$

$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

.

.

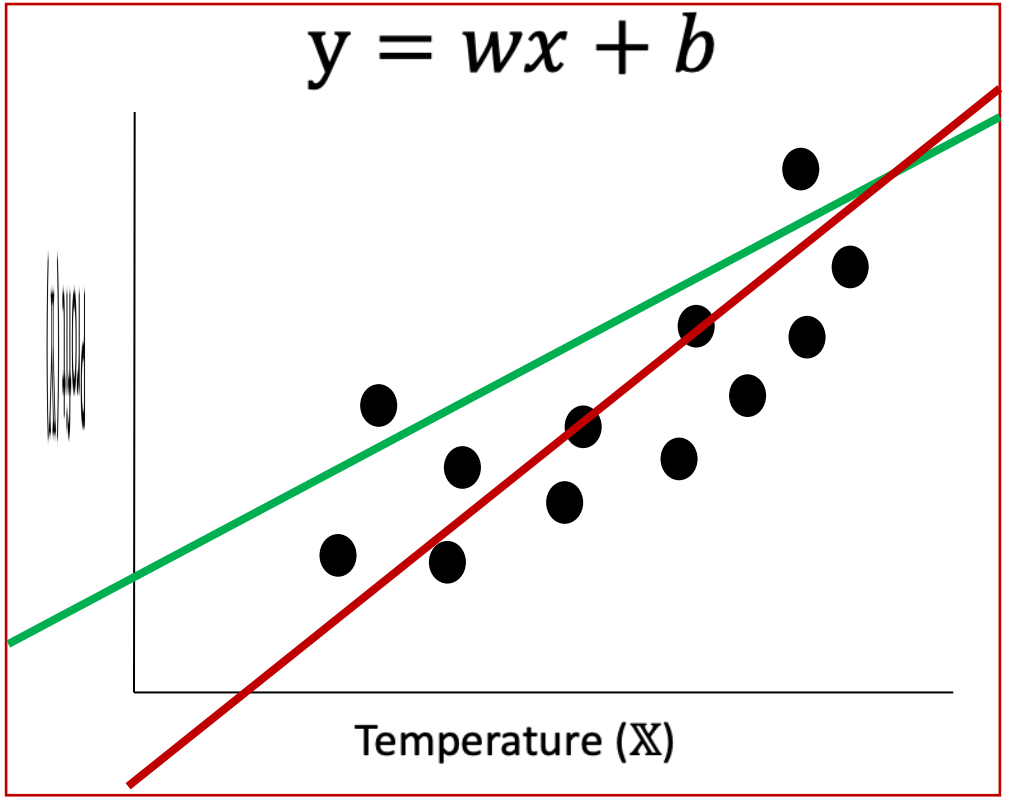
.

$x^{N} = \cdots$

$\mathbb{X} \in \mathbb{R}$

Linear function

$y = wx + b$



Target: $\mathbb{Y}$

"Profit made on selling lemonade"

$y^{(1)} = 200.0$

$y^{(2)} = 180.5$

$y^{(3)} = 115.1$

.

.

.

$y^{N} = \cdots$

$\mathbb{Y} \in \mathbb{R}$

(Numerical output)

(Image only for explaining concept, not drawn accurately)

Learning better models – Try different functions

Input: $\mathbb{X}$

"Temperature"

$\mathbb{X} \in \mathbb{R}$

$x^{(1)} = 100.1$

$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

.

.

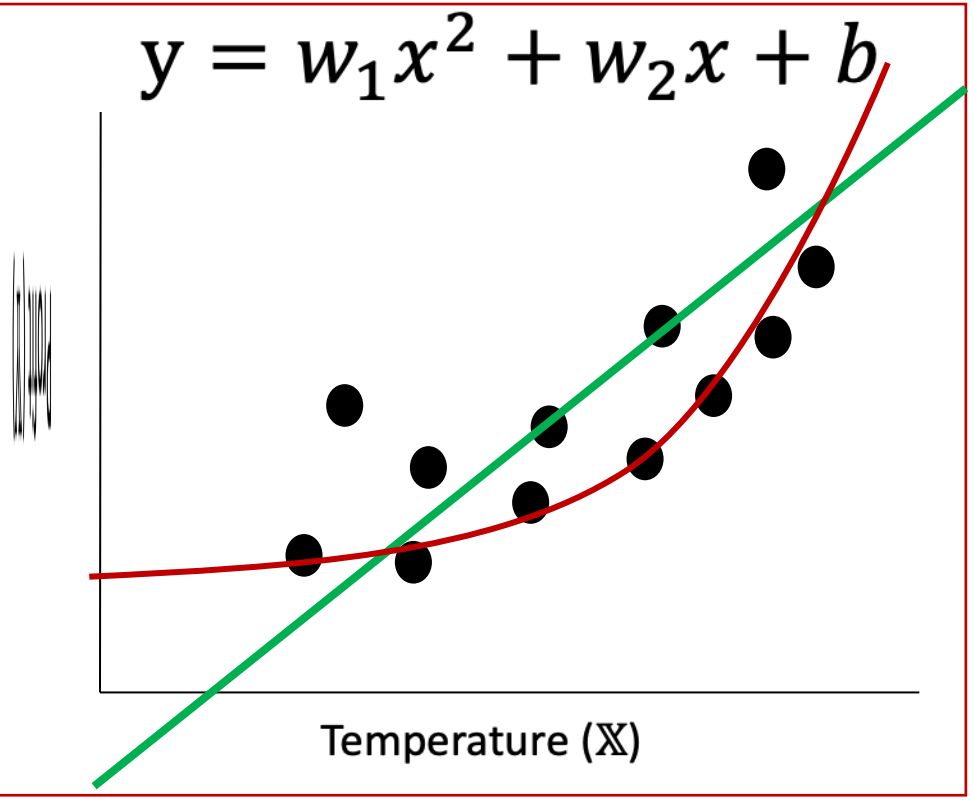
.

$x^N = \ldots$

Non-linear function

Polynomial function

$y = w_1 x^2 + w_2 x + b$



Target: $\mathbb{Y}$

"Profit made on selling lemonade"

$y^{(1)} = 200.0$

$y^{(2)} = 180.5$

$y^{(3)} = 115.1$

.

.

.

$y^N = \ldots$

$\mathbb{Y} \in \mathbb{R}$

(Numerical output)

(Image only for explaining concept, not drawn accurately)

How to know which function is the best?

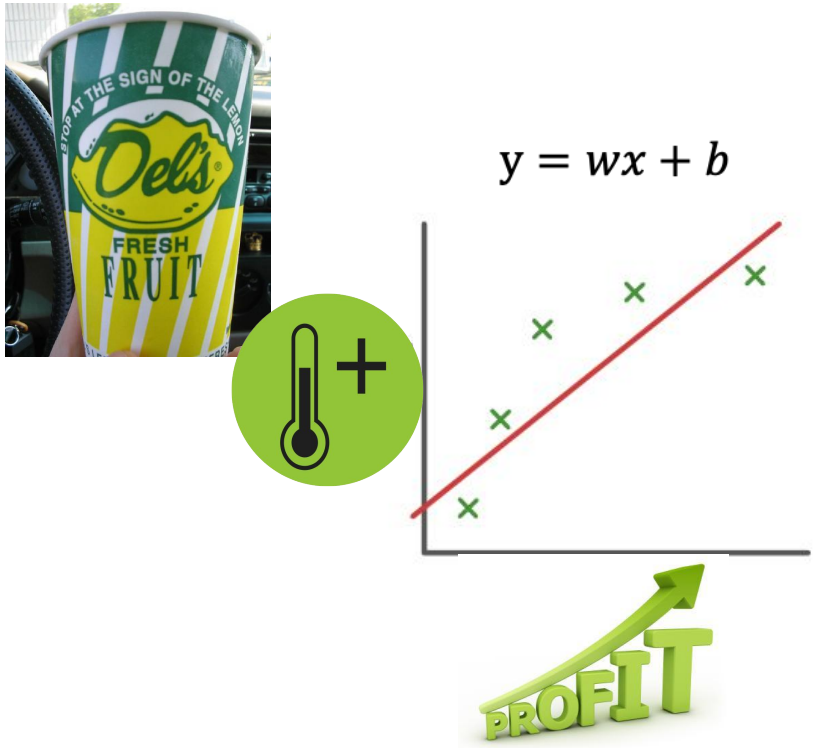


Image courtesy: https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

How to know which function is the best?

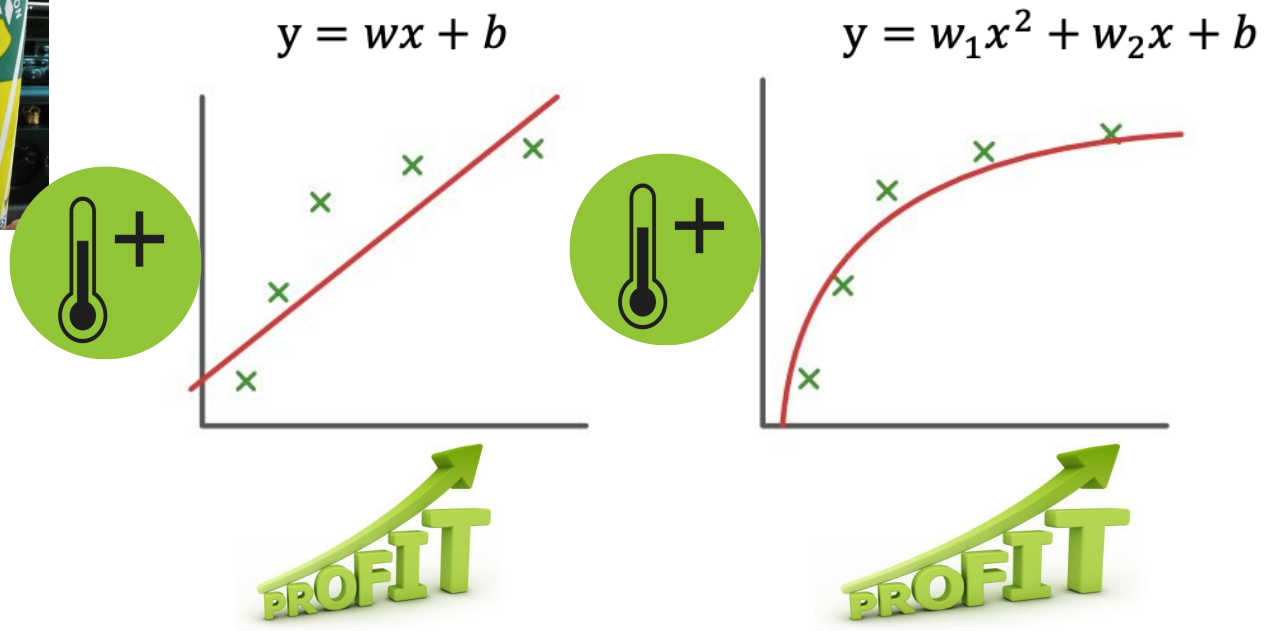
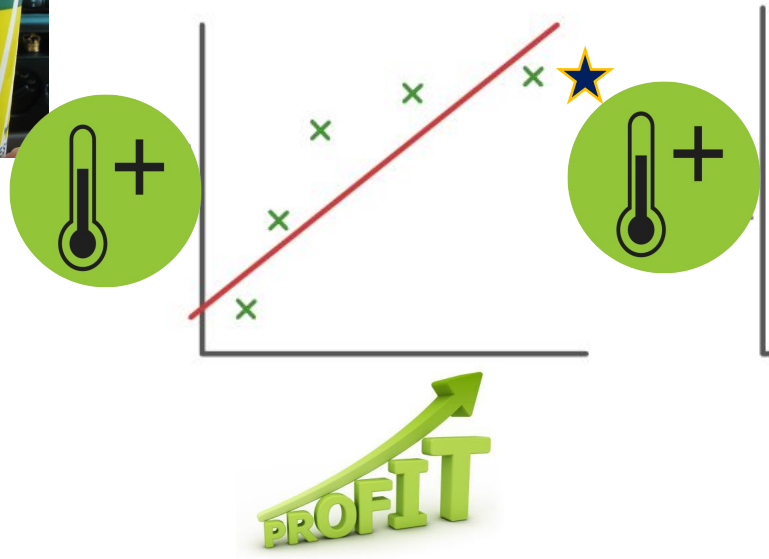
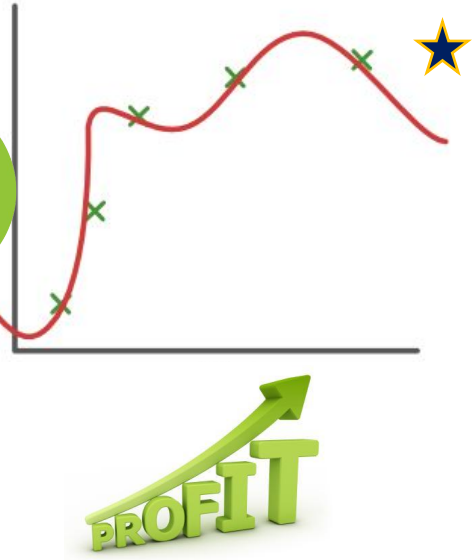


Image courtesy: https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

How to know which function is the best?

Underfit	Good fit	Overfit
$y = wx + b$	$y = w_1x^2 + w_2x + b$	$y = w_1x^4 + w_2x^3 + w_3x^2 + w_4x + b$
		
"My model is not doing that well on the given data and new data" ☹	"My model is doing well on the given data AND the new data point!! ☺	"My model is doing really well on the given data!! ☺ "The performance is bad on new data point" ☹

Any questions?

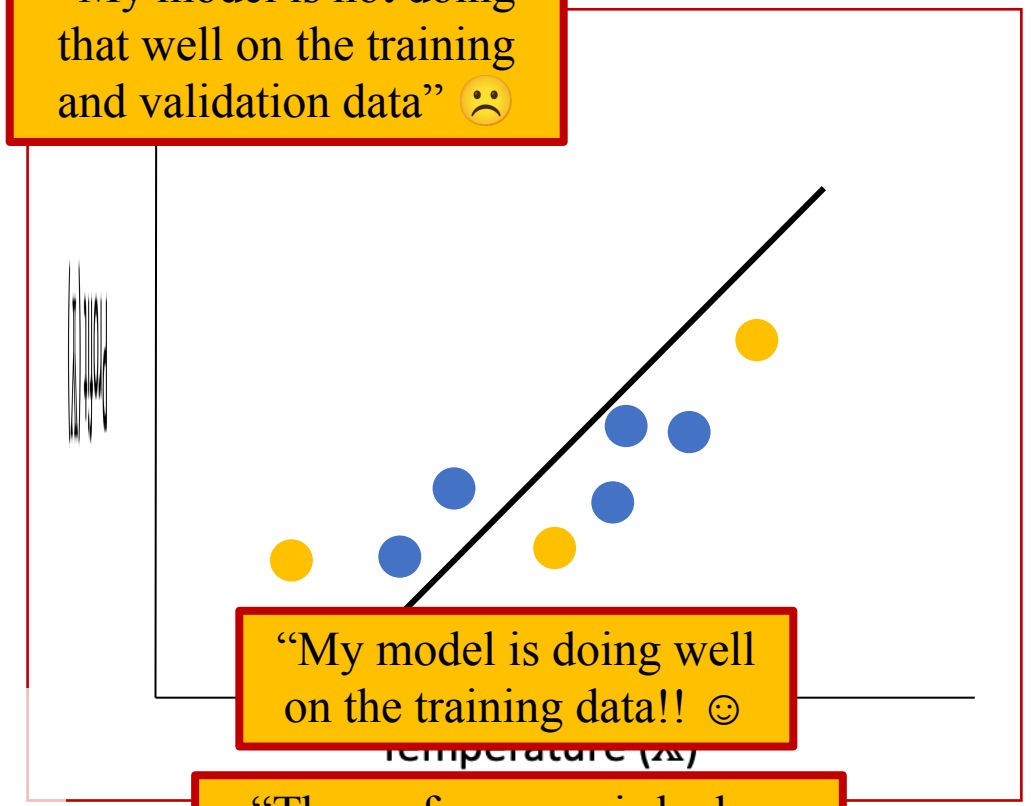
Image courtesy: https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

How to train your ~~dragon~~ model

	Input: $\mathbb{X}$ "Temperature"	Target: $\mathbb{Y}$ "Profit made on selling lemonade"
Training set	$x^{(1)} = 100.1$	$y^{(1)} = 200.0$
	$x^{(2)} = 80.0$	$y^{(2)} = 180.5$
	$x^{(3)} = 30.3$	$y^{(3)} = 115.1$
	.	.
Validation set	.	.
	.	.
	.	.
	.	.
Test set	.	.
	.	.
	$x^N = \dots$	$y^N = \dots$

Training phase

"My model is not doing that well on the training and validation data" ☹



"My model is doing well on the training data!! ☺

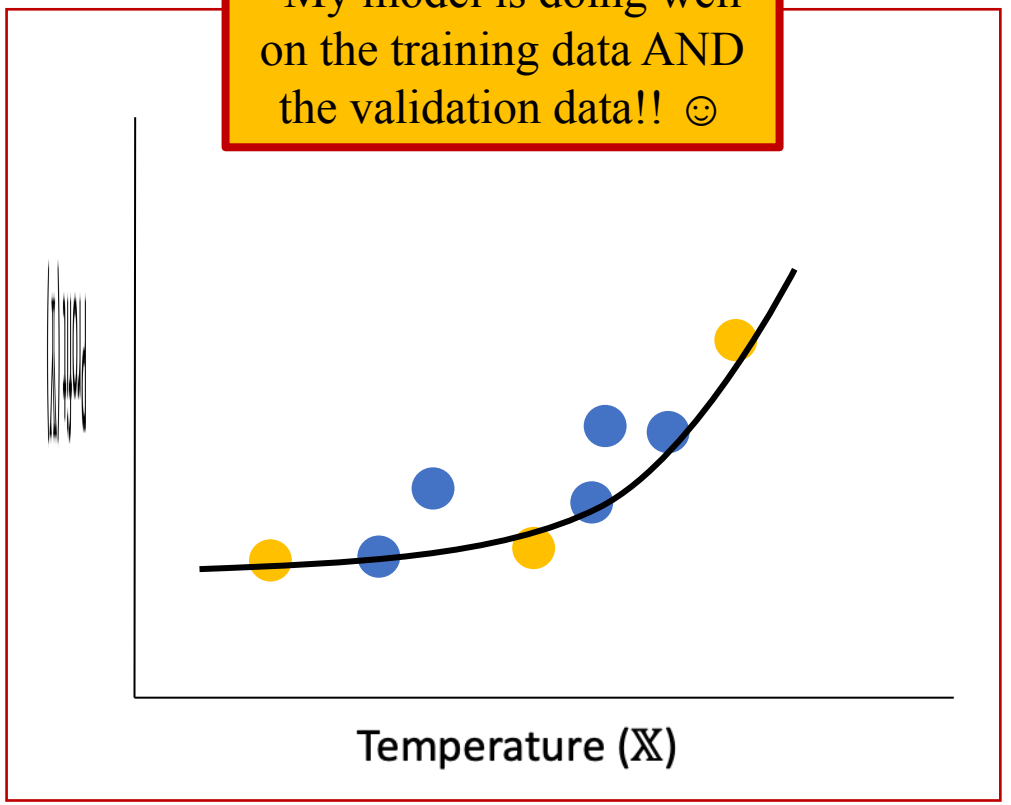
"The performance is bad on validation data" ☹

How to train your ~~dragon~~ model

	Input: $\mathbb{X}$	Target: $\mathbb{Y}$
	"Temperature"	"Profit made on selling lemonade"
Training set	$x^{(1)} = 100.1$	$y^{(1)} = 200.0$
	$x^{(2)} = 60.0$	$y^{(2)} = 160.5$
	$x^{(3)} = 30.3$	$y^{(3)} = 115.1$
	.	.
Validation set	.	.
	.	.
	.	.
	.	.
Test set	.	.
	.	.
	$x^N = \ldots$	$y^N = \ldots$

Training phase

"My model is doing well on the training data AND the validation data!! ☺

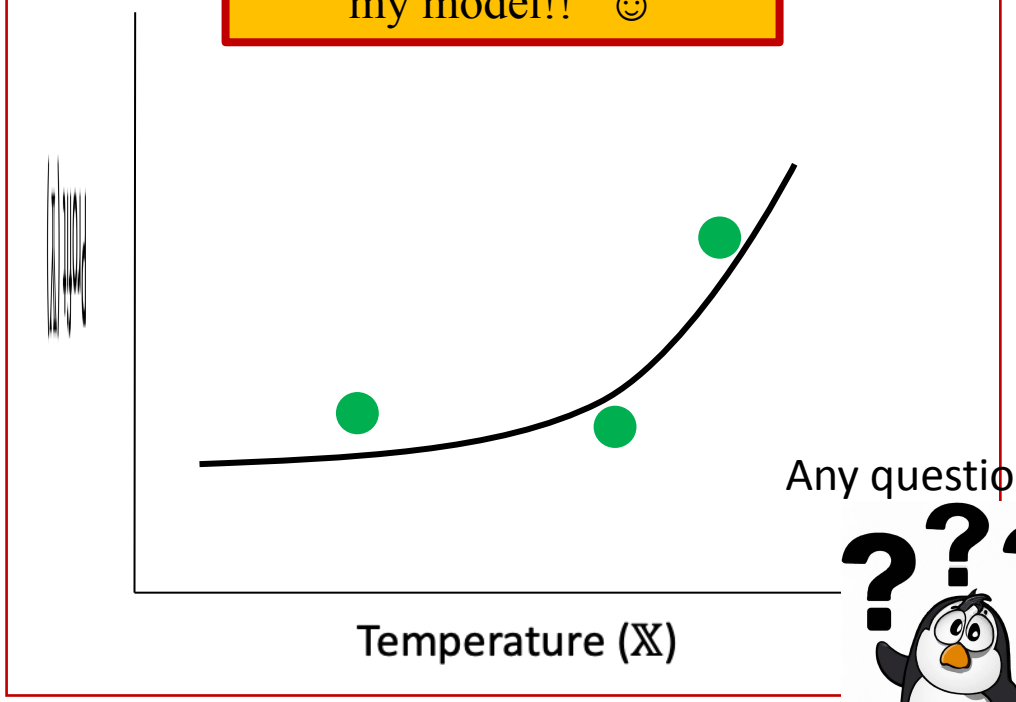


How to train your ~~dragon~~ model

	Input: $\mathbb{X}$	Target: $\mathbb{Y}$
	"Temperature"	"Profit made on selling lemonade"
Training set	$x^{(1)} = 100.1$	$y^{(1)} = 200.0$
	$x^{(2)} = 80.0$	$y^{(2)} = 180.5$
	$x^{(3)} = 30.3$	$y^{(3)} = 115.1$
	.	.
Validation set	.	.
	.	.
	.	.
Test set	.	.
	.	.
	$x^N = \ldots$	$y^N = \ldots$

Testing phase

"I have more confidence in the generalizability of my model!!" ☺



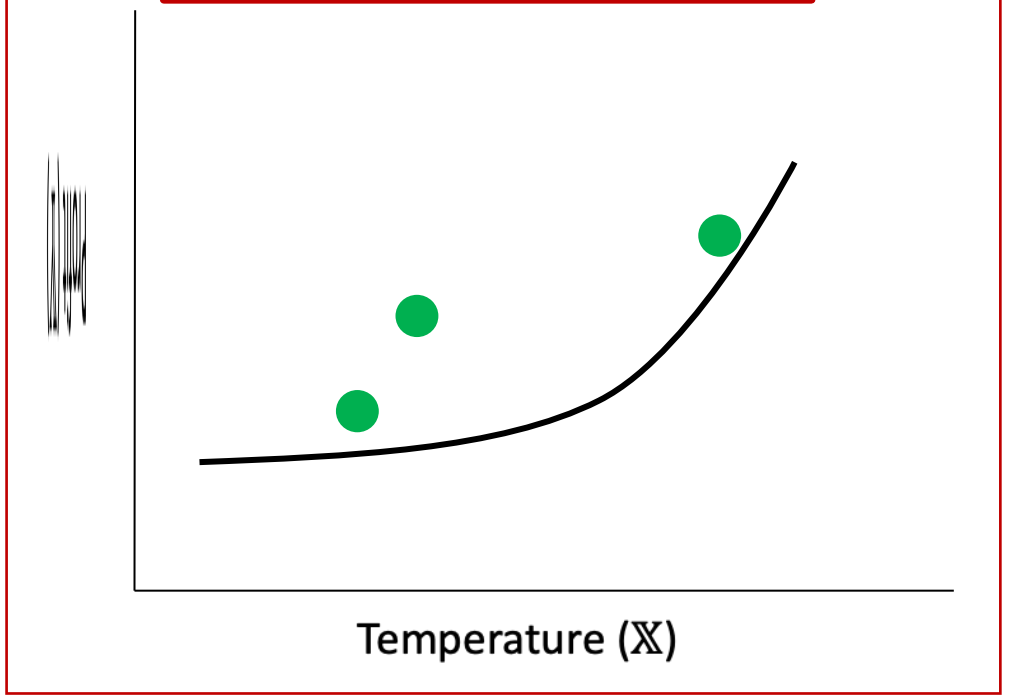
Any questions?

How to train your ~~dragon~~ model

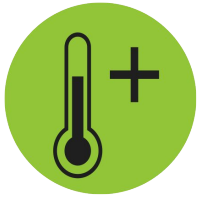
Input: $\mathbb{X}$	Target: $\mathbb{Y}$
"Temperature"	"Profit made on selling lemonade"
Training set	
$x^{(1)} = 100.1$	$y^{(1)} = 200.0$
$x^{(2)} = 80.0$	$y^{(2)} = 180.5$
$x^{(3)} = 30.3$	$y^{(3)} = 115.1$
.	.
Validation set	
.	.
Test set	
.	.
$x^N = \ldots$	$y^N = \ldots$

Testing phase

"Somethings off in my training strategy" ☹



Real world data tends to be complicated!



Input: $\mathbb{X}$

"Temperature" "Stand Hours" "Sunny?"

$$x_1^{(1)} = 100.1 \quad x_2^{(1)} = 8 \quad x_3^{(1)} = 1$$
$$x_1^{(2)} = 80.0 \quad x_2^{(2)} = 4 \quad x_3^{(2)} = 1$$
$$x_1^{(3)} = 30.3 \quad x_2^{(3)} = 8 \quad x_3^{(3)} = 0$$
$$\vdots \qquad \vdots \qquad \vdots$$
$$x_i^{(k)} = \cdots$$

$$\mathbb{X} \in \mathbb{R}^3$$

Now our function needs to capture the relationships of the combined feature space of the input and the output!

Target: $\mathbb{Y}$

"Profit made on selling lemonade"

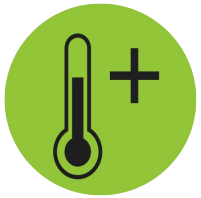
$$y^{(1)} = 200.0$$
$$y^{(2)} = 180.5$$
$$y^{(3)} = 115.1$$
$$\vdots$$
$$y^{(k)} = \ldots$$

$$\mathbb{Y} \in \mathbb{R}$$

(Numerical output)

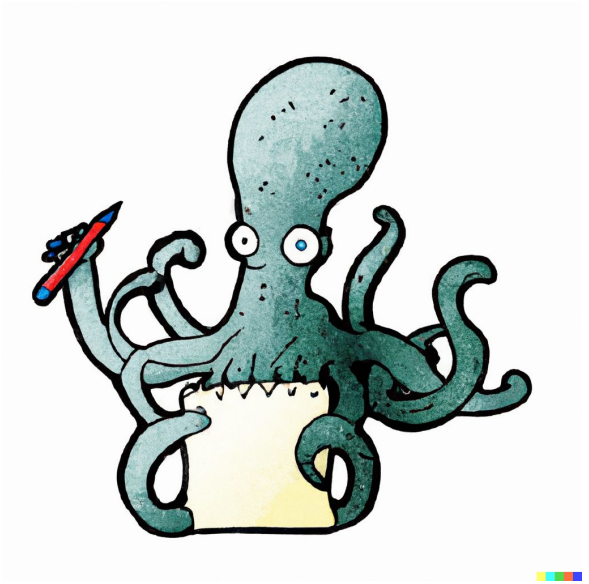
(Image only for explaining concept, not drawn accurately)

Recap



How to represent inputs and outputs

- Represent input and output as numbers
- Classification – predicting categorical outputs
- Regression – predicting numerical outputs



Supervised Learning

- Learn a function that approximates the data well
- Get more data!
- Try different models
- Pick a good model