CSCI 1470

Eric Ewing

Friday, 1/24/25

Deep Learning

Day 2: Machine Learning Fundamentals

What do you want to get out of the Class?

- Understanding Applications and Real-World Implementation (36.8% of responses)
- Theoretical Understanding of Deep Learning (why it works) (35.4% of responses)
- Career Development (a job) (28.5% of responses)
- Practical Programming Skills (22.9% of responses)
- Domain-Specific Applications (5.6% of responses)

What do you want to get out of the Class?

- Additional notable patterns:
  - Interest in understanding both the **theoretical and practical** aspects together
  - Understand **modern AI technologies** better, especially in light of recent developments
  - **Independent projects** and gaining the confidence to implement systems without supervision
  - Interest in **ethical considerations and societal impacts** of deep learning

What do you wish your professors knew about your experiences as a student?

1. Course Organization and Support (37.3% of responses)
   1. Access to resources/TAs, clear deadlines and expectations, etc.
2. Background and Experience Variation (23.0% of responses)
   2. For some people, this is their first course in Python
   3. For many students, this is their first course in AI
   4. Just because you've taken linear algebra doesn't mean you remember anything
3. Workload and Time Management (15.1% of responses)
   3. Students are stressed, especially around exam weeks
4. Learning Preferences and Styles (12.7% of responses)
   4. Many students prefer project-based learning
5. Career and Future Goals (6.3% of responses)
   5. Internship/job interviews can cause conflicts
   6. Goal of your education is to get a job after and want to work towards that goal
6. Accessibility Needs (4.0% of responses)

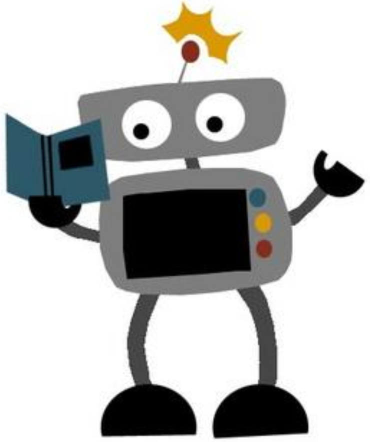
What do you want to get out of the Class?

- A seat…

Any pending override approvals not accepted by 5pm today will be revoked and new overrides will be given out.

Aiming for ~225 students

Recap: Machine Learning



Input: X

Output: Y

”Cooking?”

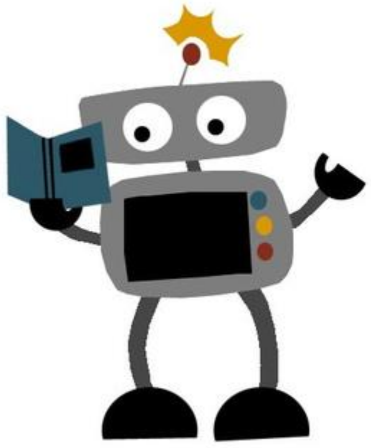
Function: f

$f(X) \longrightarrow Y$

Todays Goals

1) How do we represent Input/Output? What are X and y?
2) How can we learn a function f?
3) How do you know if a ML model is “Good”?

How do we represent Input/Output?



Input: X

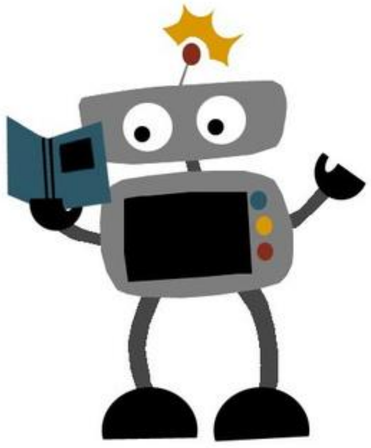
Output: Y

"Cooking?"

Function: f

$f(X) \longrightarrow Y$

How do we represent Input/Output?



Input: X

Output: Y

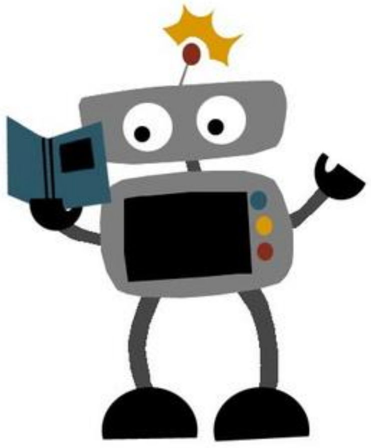
"Cooking?"

Computers work with numbers!

Function: f

$f(X) \longrightarrow Y$

How do we represent Input/Output?



Input: X

Computers work with numbers!

Function: f

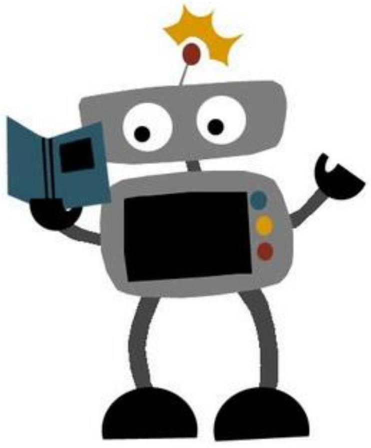
f(X) ⟶ Y

Output: Y

”Cooking?”

How can we represent output labels as numbers?

How do we represent Input/Output?



Input: X

Output: Y

"Cooking?"

Computers work with numbers!

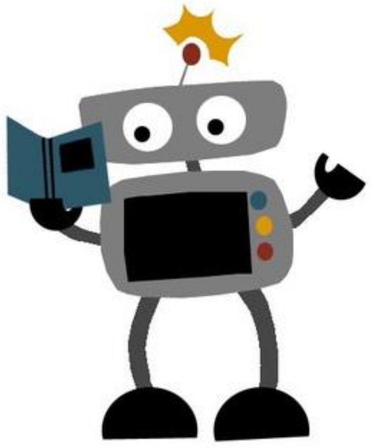
Function: f

How can we represent output labels as numbers?

How can we represent Input with numbers?

$f(X) \longrightarrow Y$

How do we represent Input/Output?



Input: X

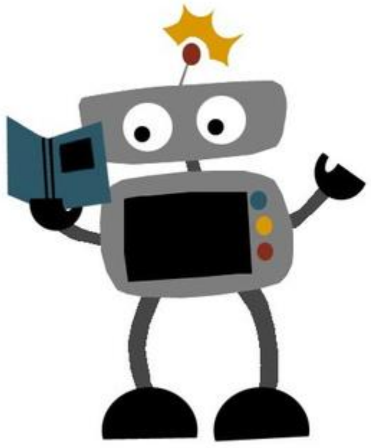
Function: f

$f(X) \longrightarrow Y$

Output: Y

”Cooking?”

How do we represent Input/Output?



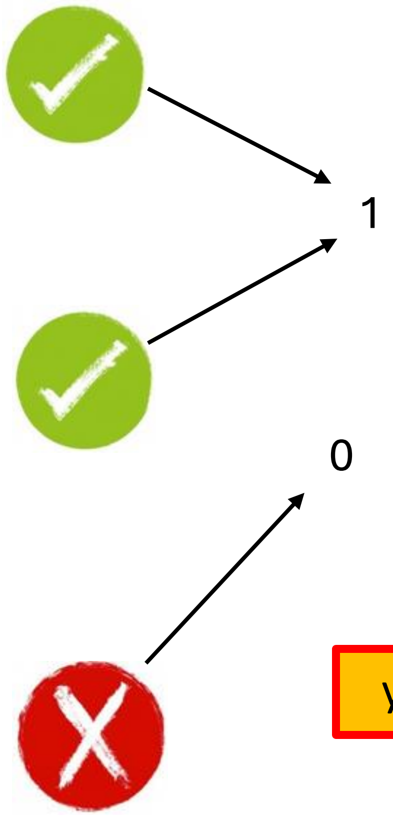
Input: X

Function: f

$f(X) \longrightarrow Y$

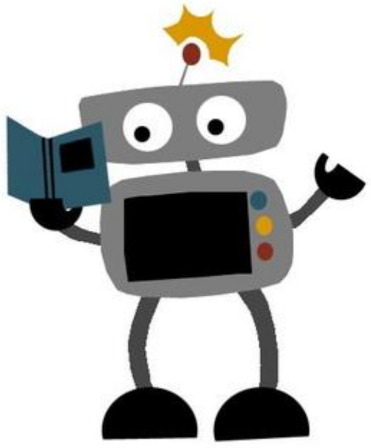
Output: Y

”Cooking?”

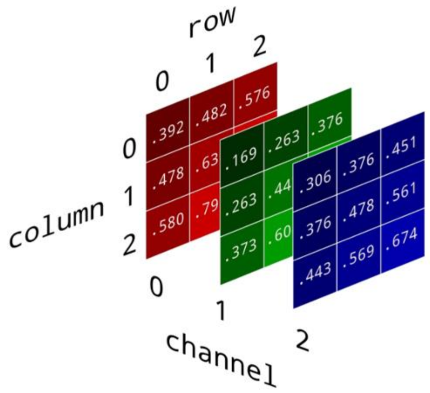


$y \in \{0,1\}$

How do we represent Input/Output?



Input: X



Function: f

Output: Y

”Cooking?”

1

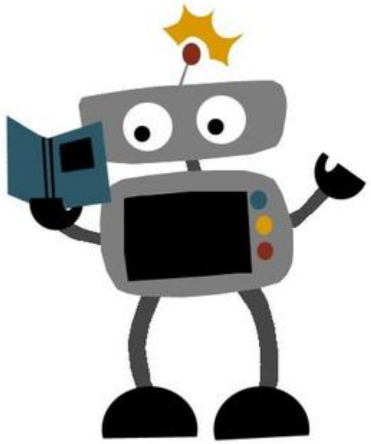
0

$X \in \mathbb{R}^{HxWx3}$

$f(X) \longrightarrow Y$

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

Classification



When y is discrete, the task is **classification**

Input: X

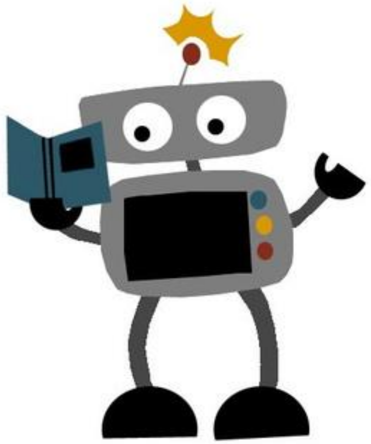
Function: f

$f(X) \longrightarrow Y$

Output: Y

”Cooking?”

Classification



When y is discrete, the task is **classification**

When $y \in \{0, 1\}$ the task is **Binary Classification**

Input: X

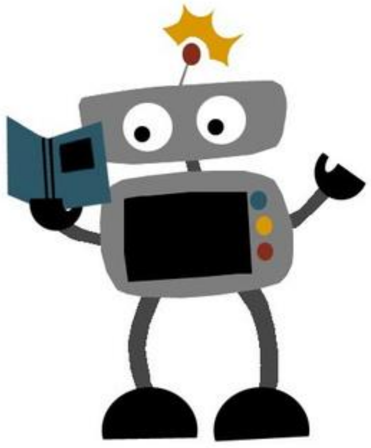
Function: f

f(X) ⟶ Y

Output: Y

”Cooking?”

Classification



When y is discrete, the task is **classification**

When $y \in \{0, 1\}$ the task is **Binary Classification**

What's an example of **multi-class Classification?**

Input: X

Function: f

$f(X) \longrightarrow Y$

Output: Y

"Cooking?"

Some Notation

Some Notation

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

Some Notation

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

$v \in \mathbb{R}^d$: A **vector** in dimension d

Some Notation

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

$v \in \mathbb{R}^d$: A **vector** in dimension d

$V \in \mathbb{R}^{H \times W}$: A **matrix** of dimensions $H \times W$

Some Notation

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

$v \in \mathbb{R}^d$: A **vector** in dimension d

$V \in \mathbb{R}^{H \times W}$: A **matrix** of dimensions $H \times W$

$V \in \mathbb{R}^{H \times W \times C}$: A **tensor** of dimensions $H \times W \times C$

Some Notation

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

$v \in \mathbb{R}^d$: A **vector** in dimension d

$V \in \mathbb{R}^{H \times W}$: A **matrix** of dimensions $H \times W$

$V \in \mathbb{R}^{H \times W \times C}$: A **tensor** of dimensions $H \times W \times C$

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

Some Notation

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

$v \in \mathbb{R}^d$: A **vector** in dimension d

$V \in \mathbb{R}^{H \times W}$: A **matrix** of dimensions $H \times W$

$V \in \mathbb{R}^{H \times W \times C}$: A **tensor** of dimensions $H \times W \times C$

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

$\mathbb{Y}$: A set of target variables (outputs/labels) for supervised learning

Some Notation

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

$v \in \mathbb{R}^d$: A **vector** in dimension d

$V \in \mathbb{R}^{H \times W}$: A **matrix** of dimensions $H \times W$

$V \in \mathbb{R}^{H \times W \times C}$: A **tensor** of dimensions $H \times W \times C$

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

$\mathbb{Y}$: A set of target variables (outputs/labels) for supervised learning

$x^{(k)}$: k'th example (input) from dataset

Some Notation

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

$v \in \mathbb{R}^d$: A **vector** in dimension d

$V \in \mathbb{R}^{H \times W}$: A **matrix** of dimensions $H \times W$

$V \in \mathbb{R}^{H \times W \times C}$: A **tensor** of dimensions $H \times W \times C$

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

$\mathbb{Y}$: A set of target variables (outputs/labels) for supervised learning

$x^{(k)}$: k'th example (input) from dataset

$y^{(k)}$: k'th example (output) associated with $x^{(k)}$

Simpler example: How do we represent input/output?

Input: $\mathbb{X}$

"Temperature"

$x^{(1)}$ 100.1 °F

$x^{(2)}$ 80.0 °F

$x^{(3)}$ 30.3 °F

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

Function: f

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)

Simpler example: How do we represent input/output?

Input: $\mathbb{X}$

"Temperature"

$x^{(1)}$ 100.1 °F

$x^{(2)}$ 80.0 °F

$x^{(3)}$ 30.3 °F

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

Function: f

Do you see a trend here?

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)

Simpler example: How do we represent input/output?

Input: $\mathbb{X}$

"Temperature"

$x^{(1)}$ 100.1 °F

$x^{(2)}$ 80.0 °F

$x^{(3)}$ 30.3 °F

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

→ Function: f →

Do you see a trend here?

What is different about the output here?

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)

Simpler example: How do we represent input/output?

Regression

Input: $\mathbb{X}$

"Temperature"

$x^{(1)}$ 100.1 °F

$x^{(2)}$ 80.0 °F

$x^{(3)}$ 30.3 °F

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

Function: f

f(X)→Y

Do you see a trend here?

What is different about the output here?

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)

Learning function f

Input: $\mathbb{X}$

"Temperature"

$x^{(1)} = 100.1$

$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

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

Function: f

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"

$x^{(1)} = 100.1$

$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

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

Function: f

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)

Step 1: Model Hypothesis: What function do we think best fits the data

(Image only for explaining concept, not drawn accurately)

Learning function f

Input: $\mathbb{X}$

"Temperature"

$x^{(1)} = 100.1$

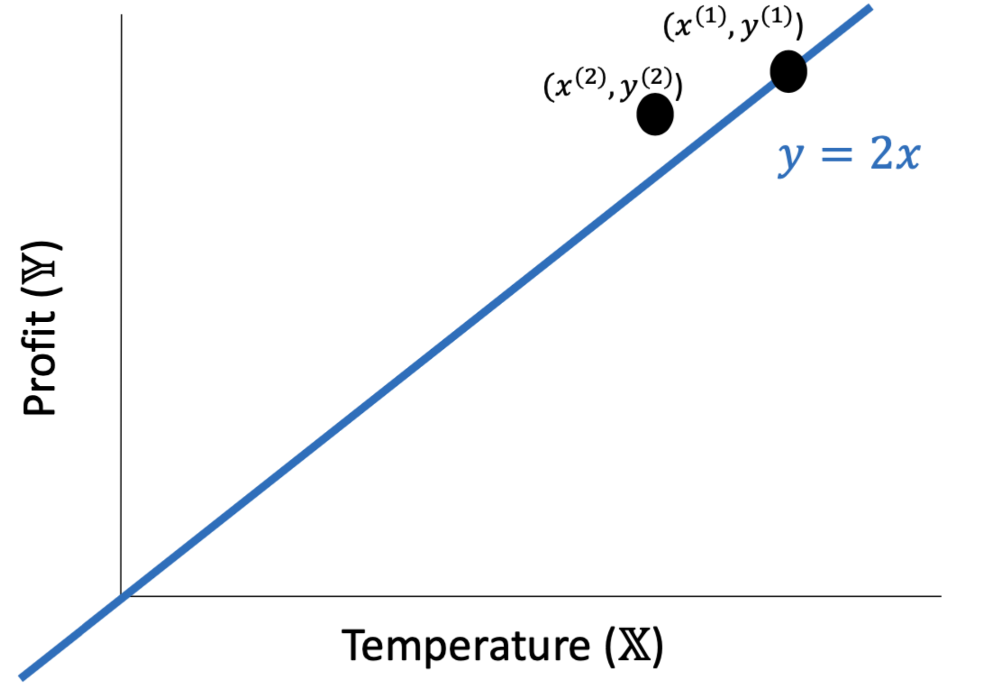
$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

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

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

Have you seen this equation before?

Input: $\mathbb{X}$

"Temperature"

$x^{(1)} = 100.1$

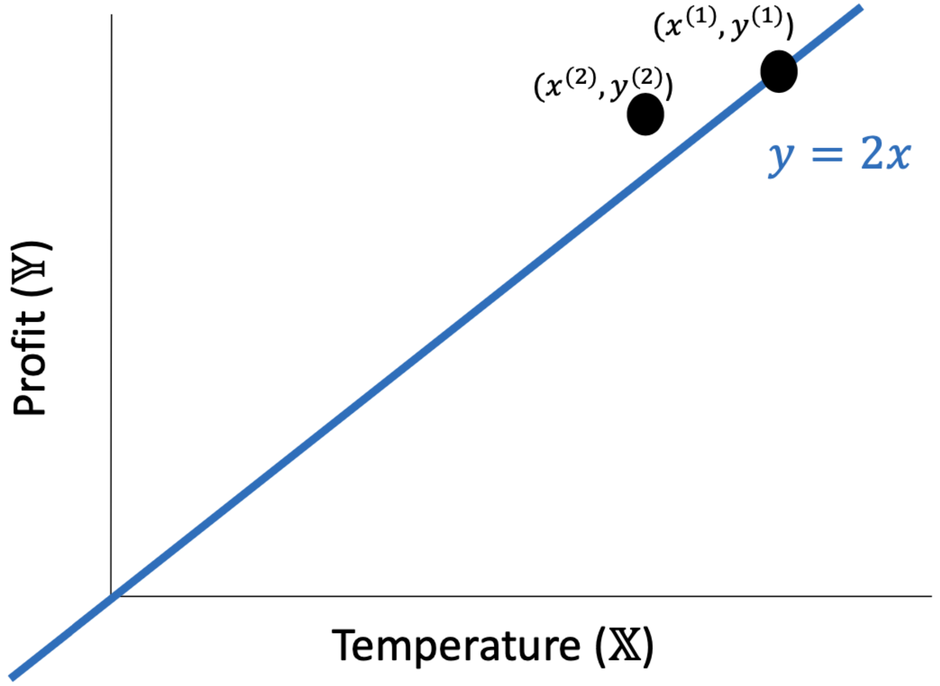
$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

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

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"

$x^{(1)} = 100.1$

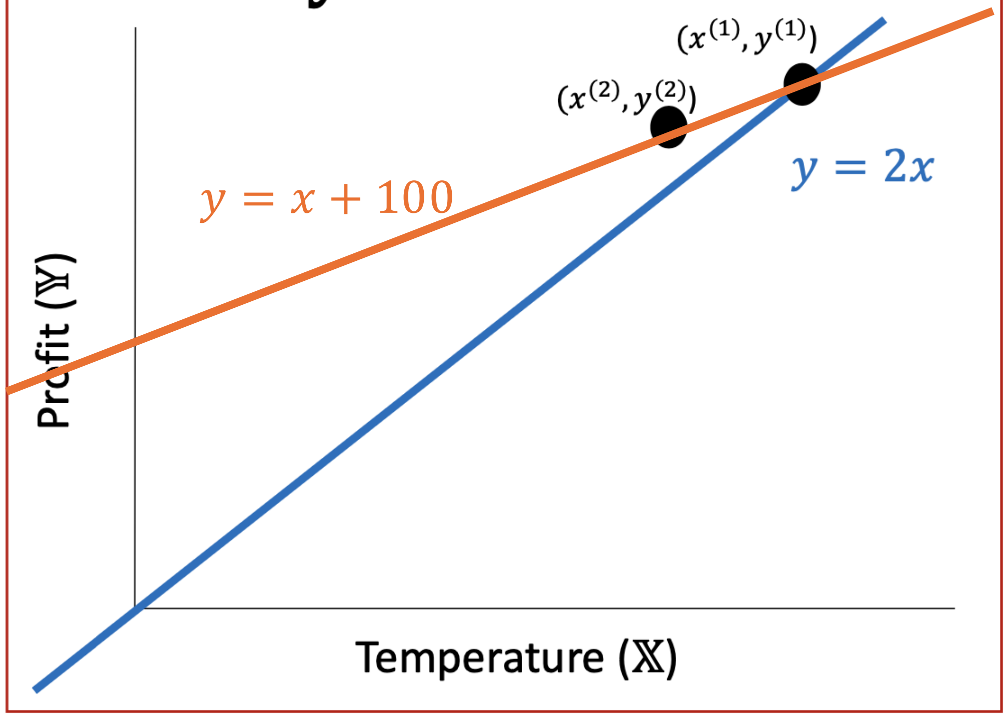
$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

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

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"

$x^{(1)} = 100.1$

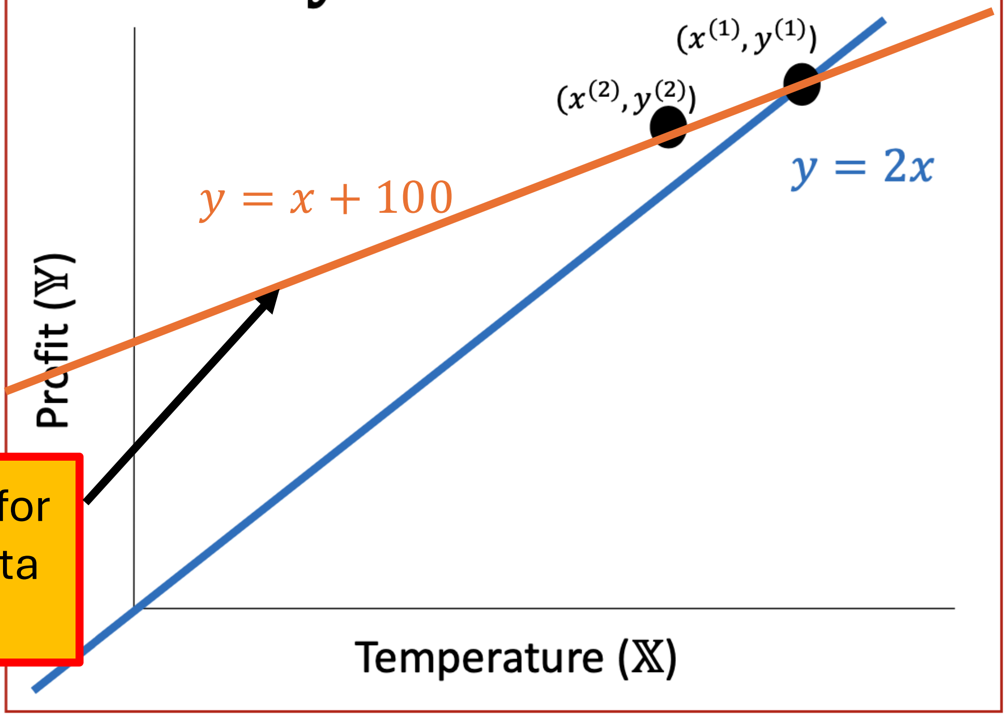
$x^{(2)} = 80.0$

$x^{(3)} = 20.2$

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

Linear function

$$y = wx + b$$



Bias term is necessary for best fit line to fit the data well

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"

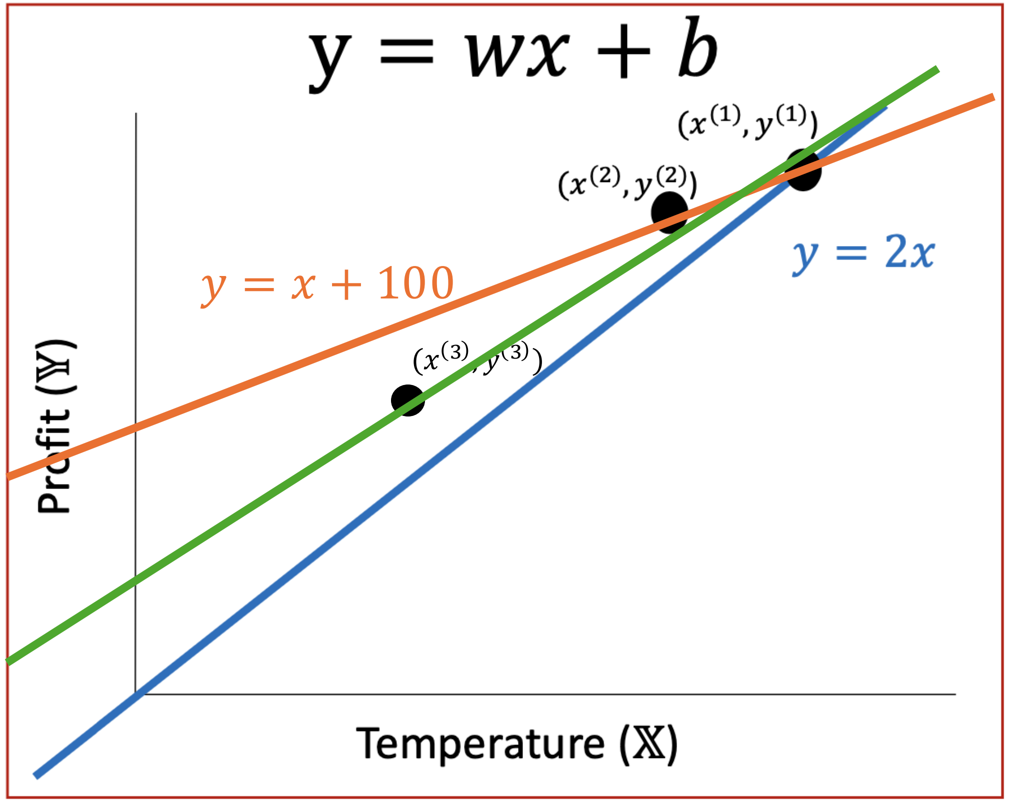
$x^{(1)} = 100.1$

$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

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

Linear function



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"

$x^{(1)} = 100.1$

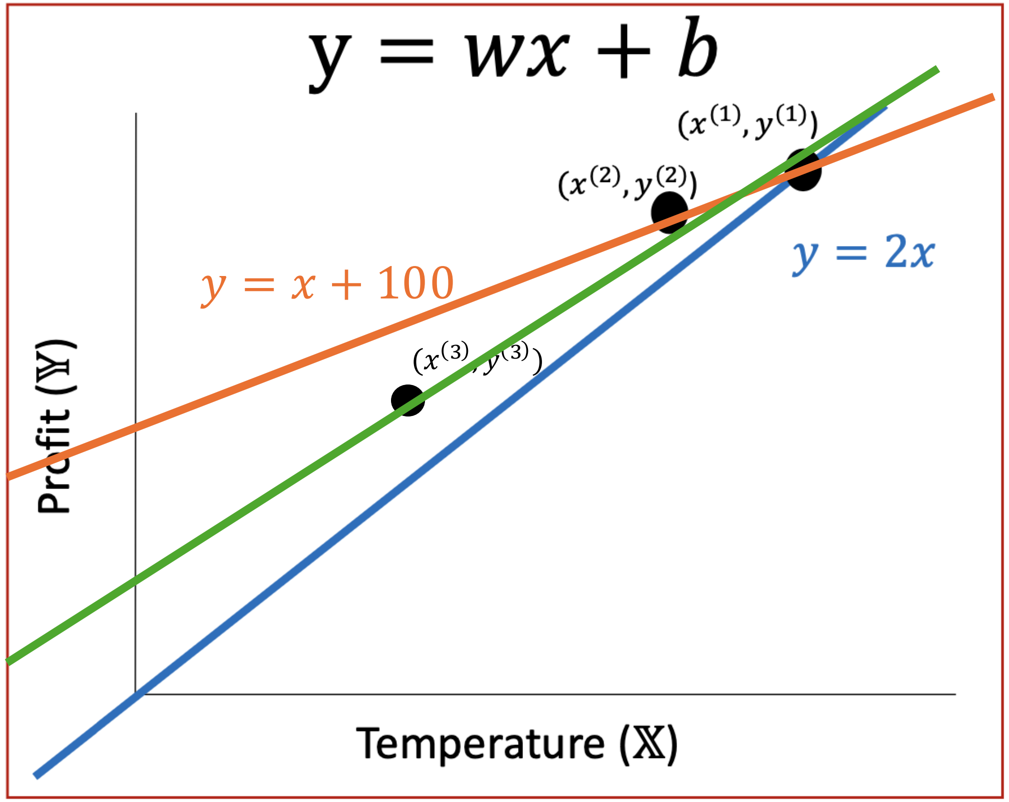
$\mathbb{X} \in \mathbb{R}$

$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

Hard to find these functions by hand...

Linear function



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"

$x^{(1)} = 100.1$

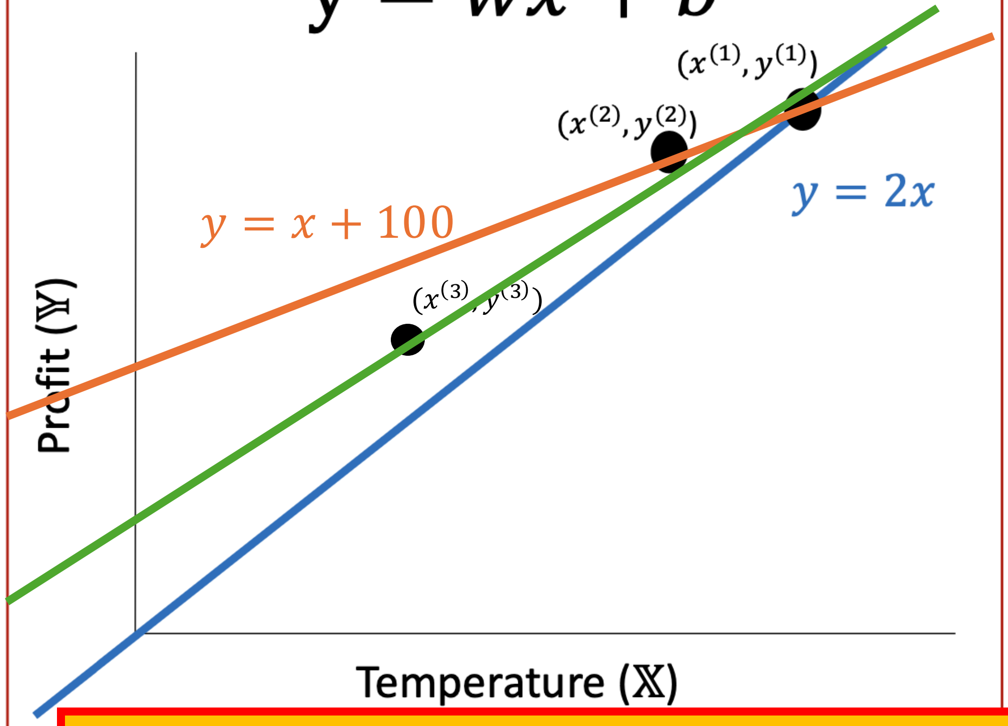
$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

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

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)

Hard to find these functions by hand...

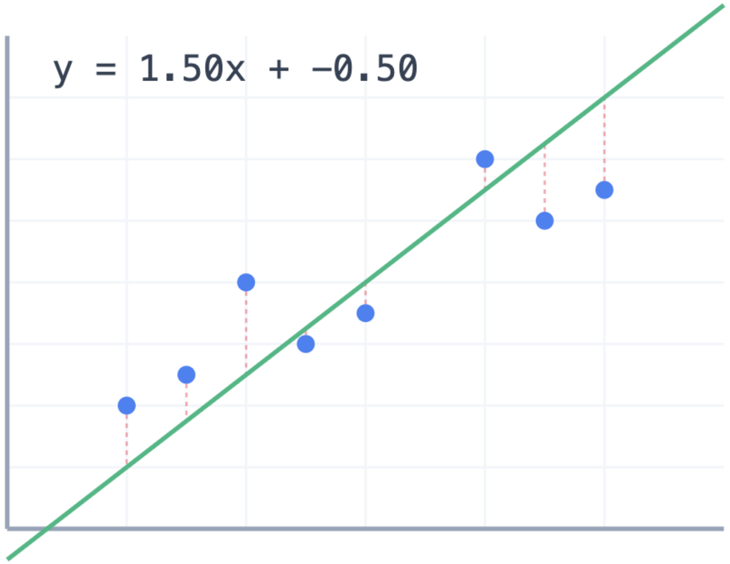
Machine Learning automatically learns good **approximations** of f from **data** (or at least tries to)

(Image only for explaining concept, not drawn accurately)

What makes a good approximation?

Loss Function: A function that describes how closely our approximation matches our data

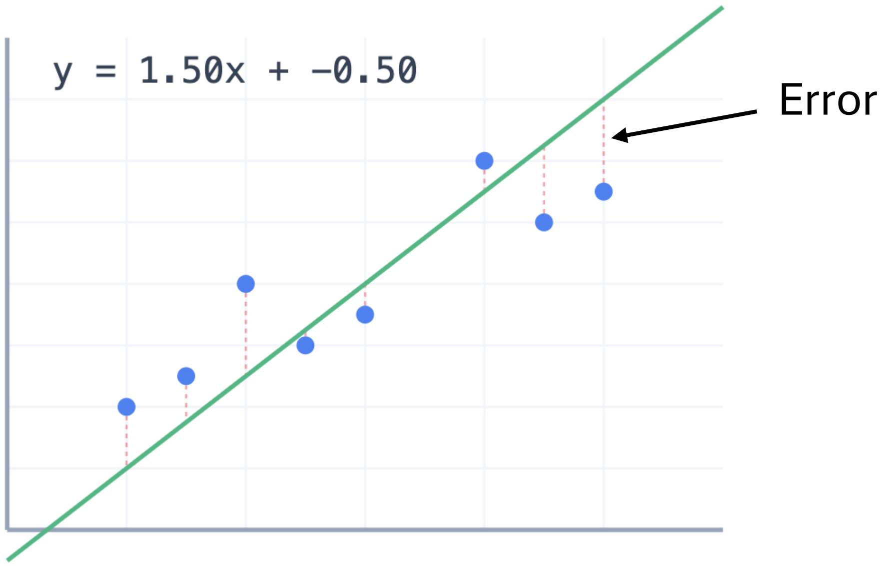
The standard loss function for Linear Regression is **Mean Squared Error (MSE)**



What makes a good approximation?

Loss Function: A function that describes how closely our approximation matches our data

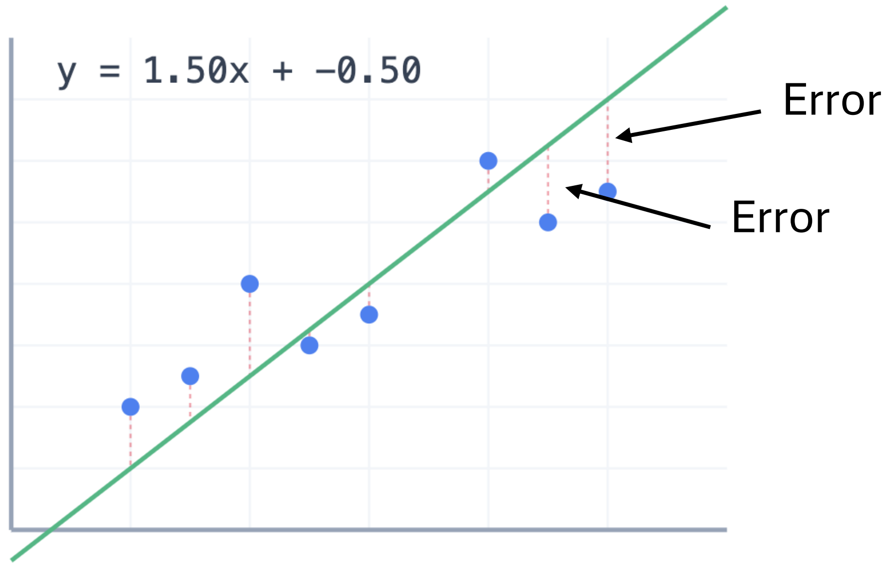
The standard loss function for Linear Regression is **Mean Squared Error (MSE)**



What makes a good approximation?

Loss Function: A function that describes how closely our approximation matches our data

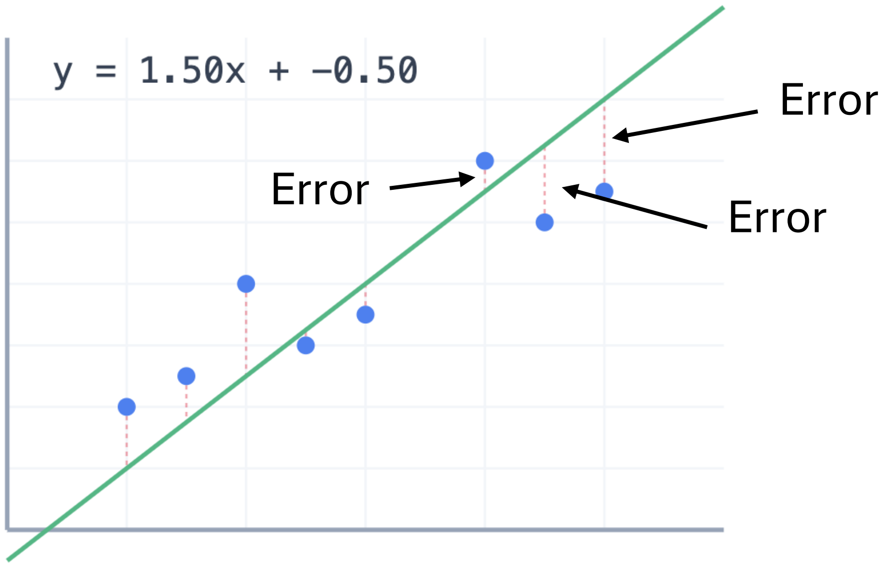
The standard loss function for Linear Regression is **Mean Squared Error (MSE)**



What makes a good approximation?

Loss Function: A function that describes how closely our approximation matches our data

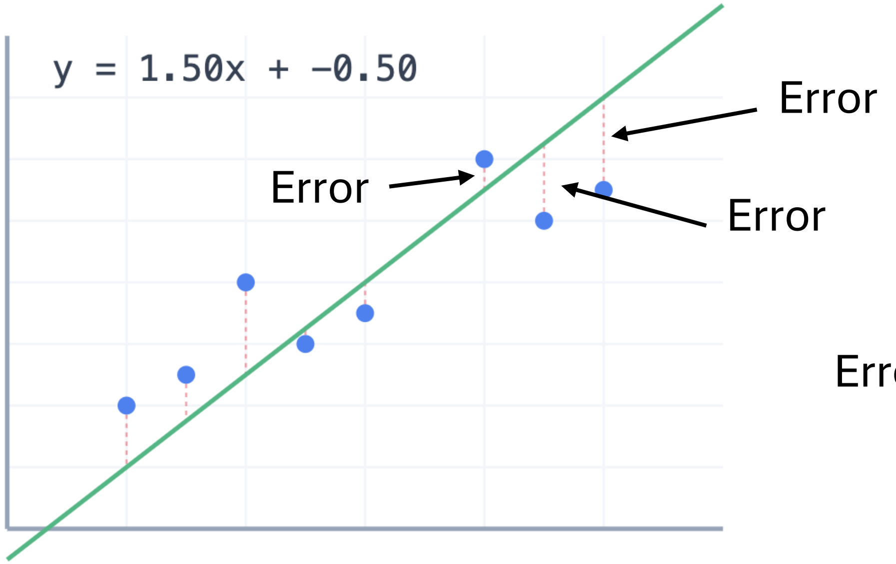
The standard loss function for Linear Regression is **Mean Squared Error (MSE)**



What makes a good approximation?

Loss Function: A function that describes how closely our approximation matches our data

The standard loss function for Linear Regression is **Mean Squared Error (MSE)**



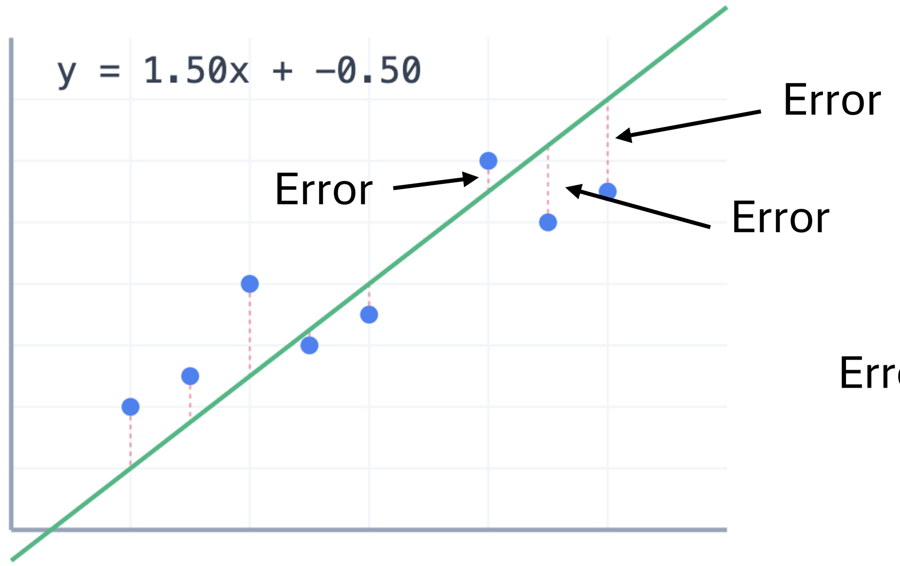
Error = $f\left(x^{(i)}\right) - y^{(i)}$

What makes a good approximation?

Loss Function: A function that describes how closely our approximation matches our data

The standard loss function for Linear Regression is **Mean Squared Error (MSE)**

$$MSE = \frac{\sum_i^n \left(f\left(x^{(i)}\right) - y^{(i)}\right)^2}{n}$$

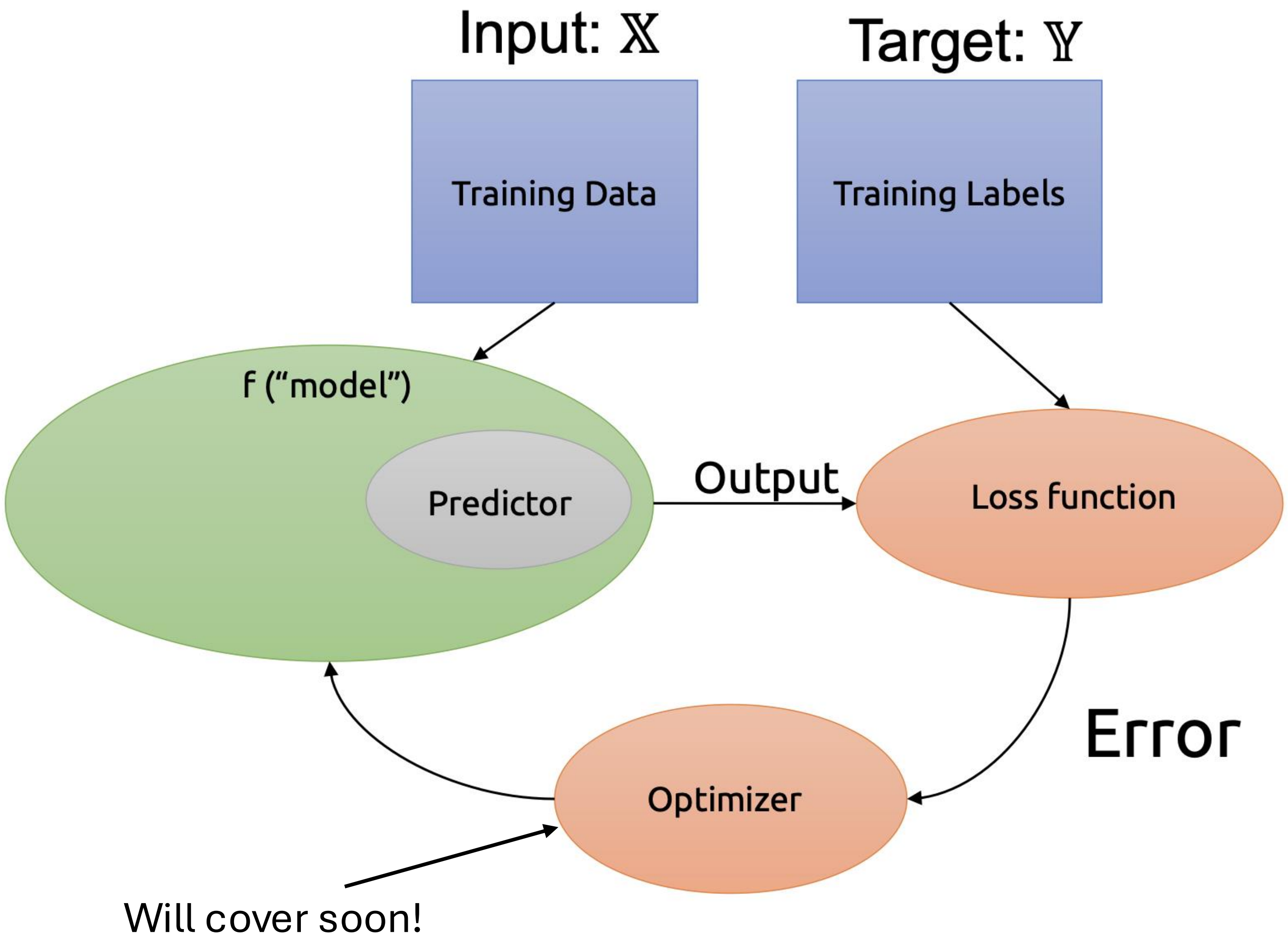


Error = $f\left(x^{(i)}\right) - y^{(i)}$

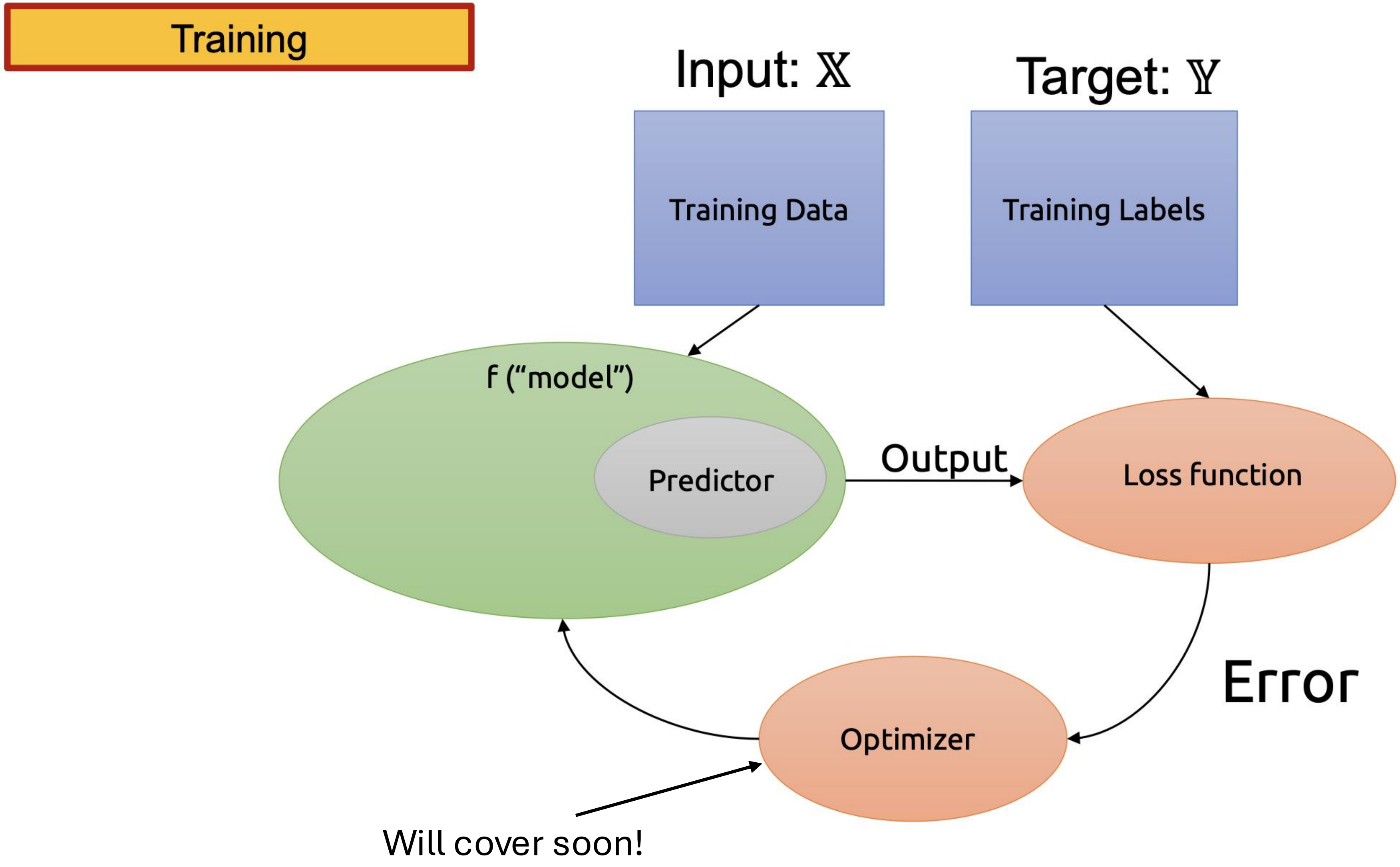
What is the best approximation?

- https://brown-deep-learning.github.io/dl-website-s25/visualizations/visualizations.html

“Classic” Supervised Learning in Machine Learning

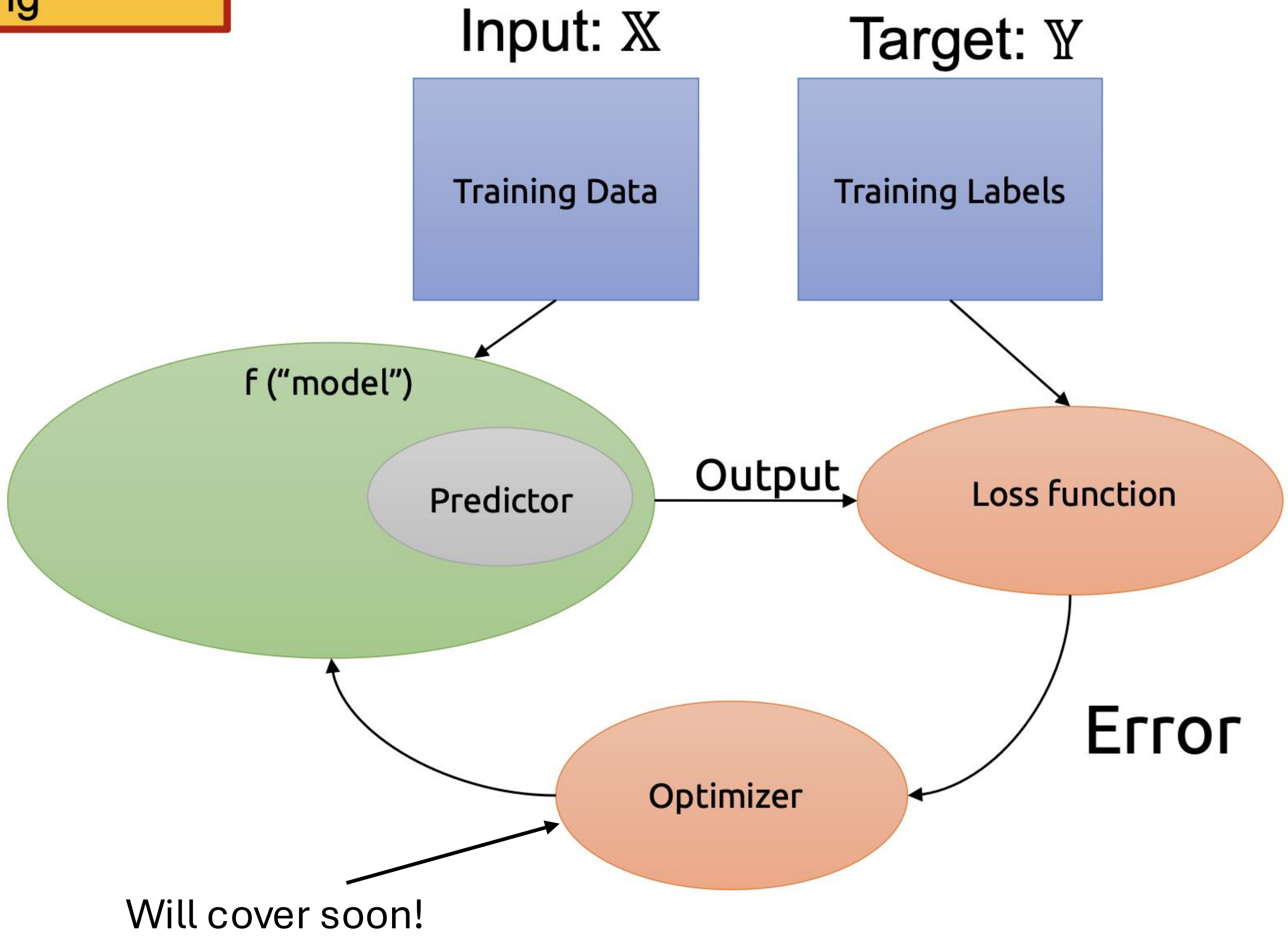


“Classic” Supervised Learning in Machine Learning



“Classic” Supervised Learning in Machine Learning

Training



Any questions?

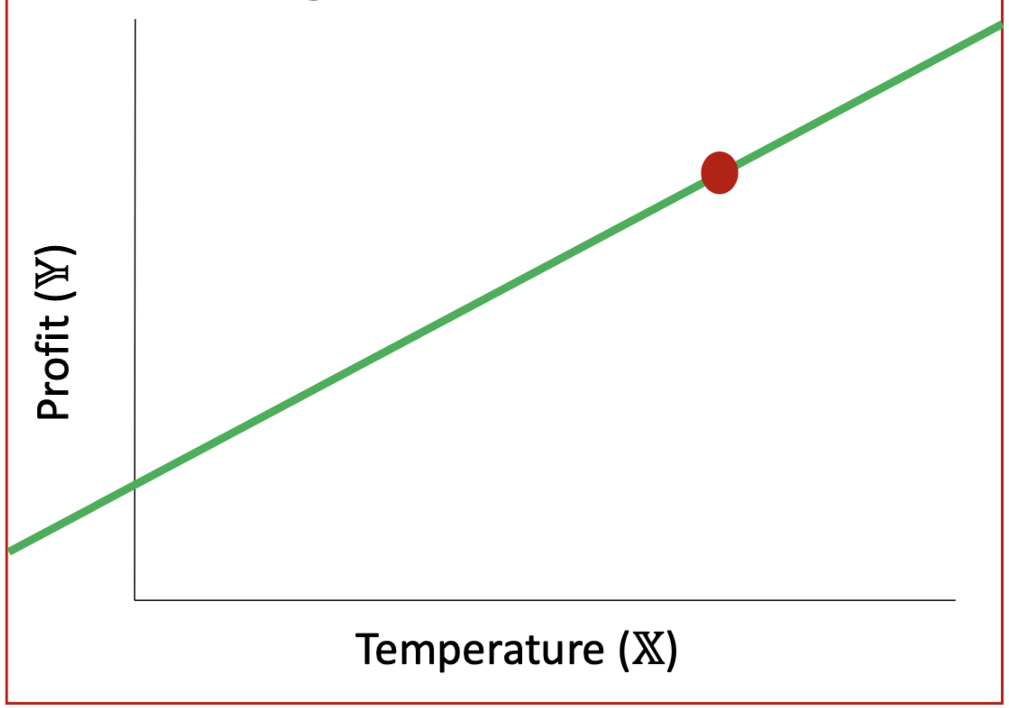
Testing our model

"Temperature"

$x' = 70$

Linear function

$$y = wx + b$$



"Profit made on selling lemonade"

Prediction

$y' = 175$

(Image only for explaining concept, not drawn accurately)

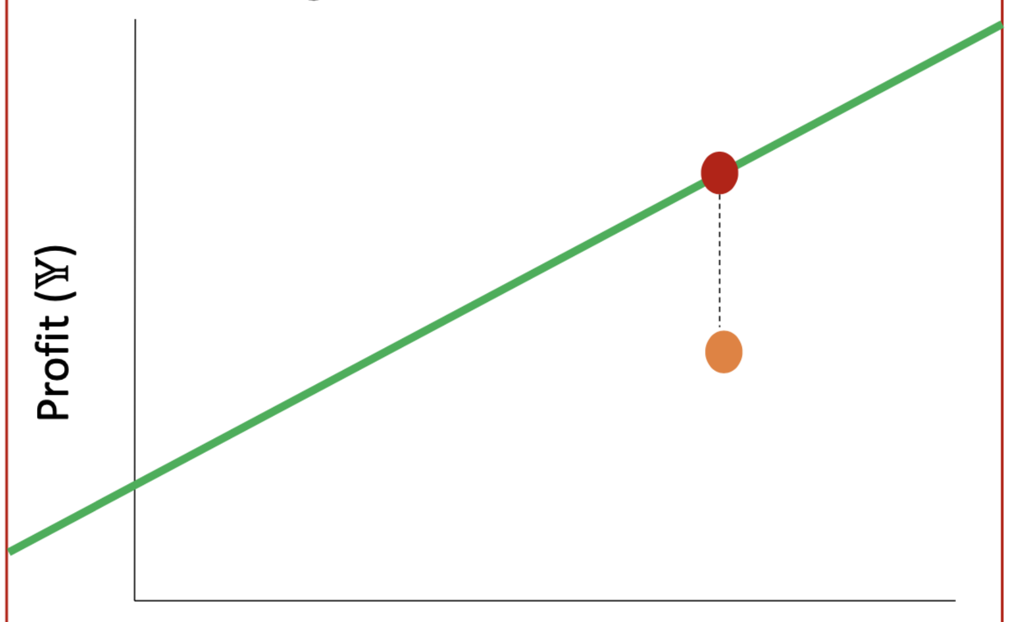
Testing our model

"Temperature"

"Profit made on selling lemonade"

Linear function

$$y = wx + b$$



$x' = 70$

$\hat{x} = 70$

Prediction

$y' = 175$

True observation

$\hat{y} = 140$

(Image only for explaining concept, not drawn accurately)

Learning better models – Collect more data

Input: $\mathbb{X}$

"Temperature"

$x^{(1)} = 100.1$

$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

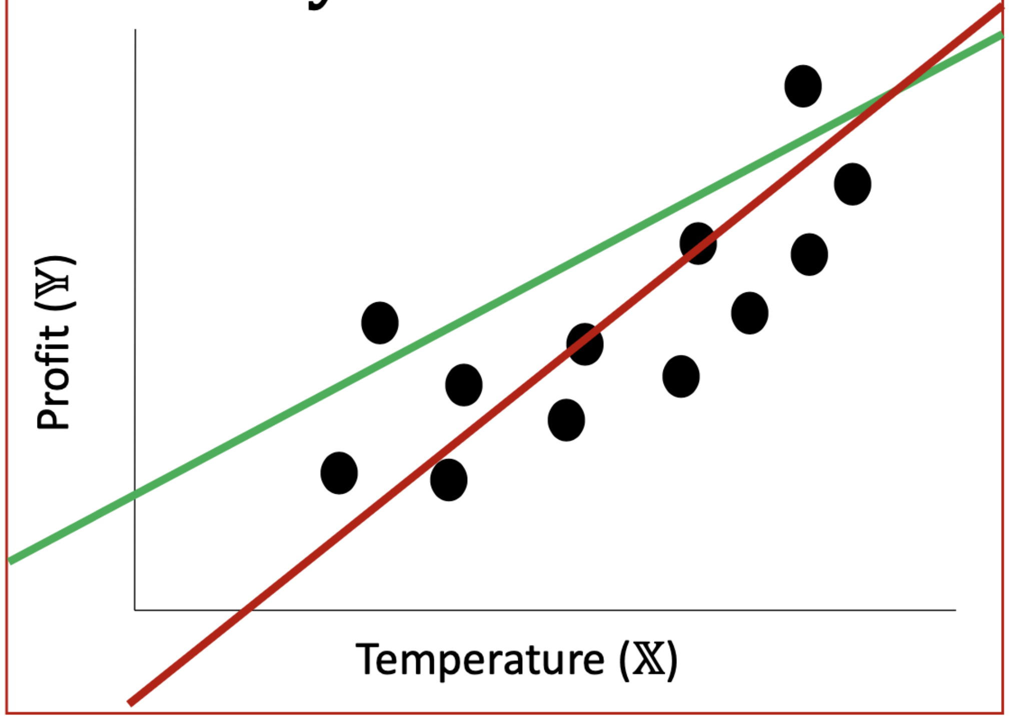
.
.
.
.

$x^{N} = \dots$

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

Linear function

$$y = wx + b$$



Target: $\mathbb{Y}$

"Profit made on selling lemonade"

$y^{(1)} = 200.0$

$y^{(2)} = 180.5$

$y^{(3)} = 115.1$

.
.
.
.

$y^{N} = \dots$

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

(Numerical output)

(Image only for explaining concept, not drawn accurately)

Learning better models – Try different functions

Input: $\mathbb{X}$

"Temperature"

$x^{(1)} = 100.1$

$x^{(2)} = 80.0$

$x^{(3)} = 30.3$

.
.
.
.

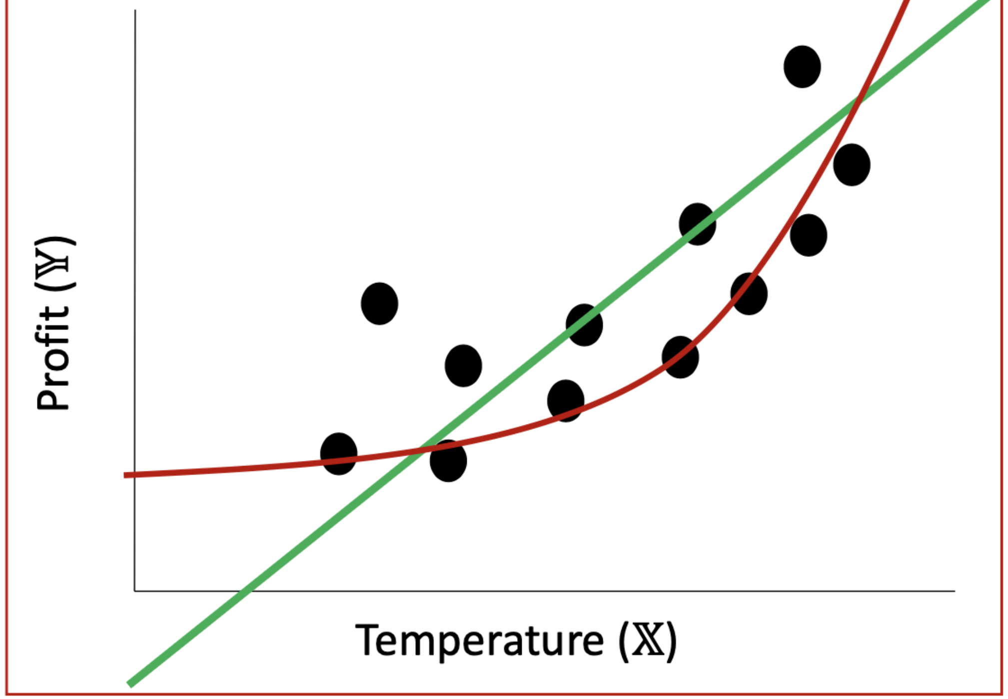
$x^{N} = \ \ldots$

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

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?

$$y = wx + b$$

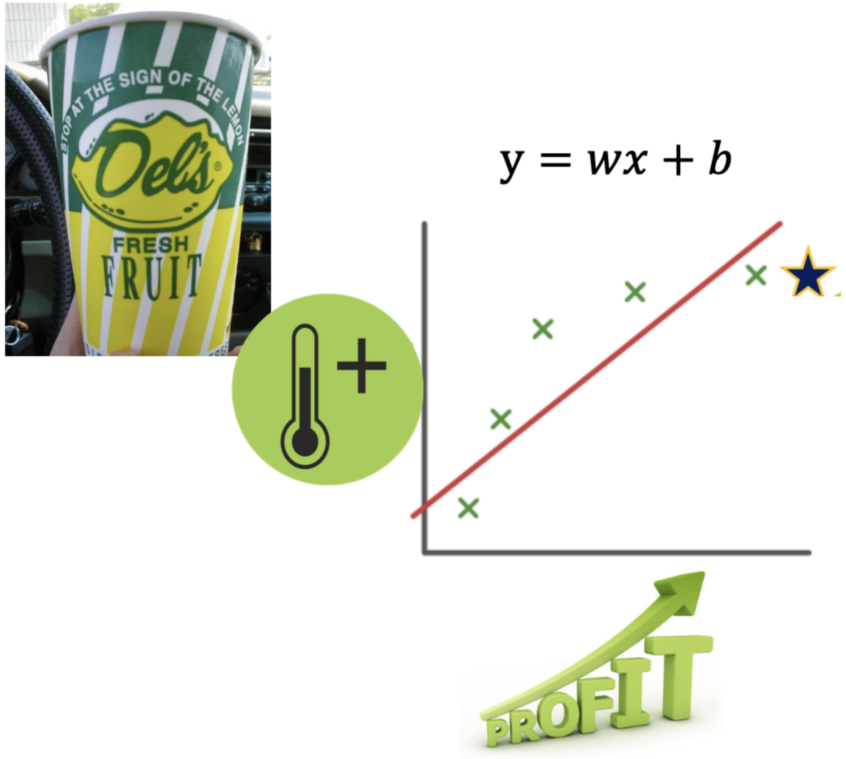
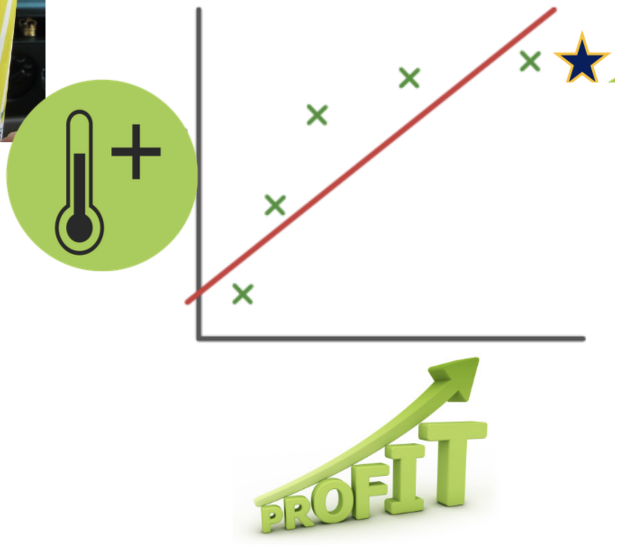


Image courtesy: https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

How to know which function is the best?

$$y = wx + b$$



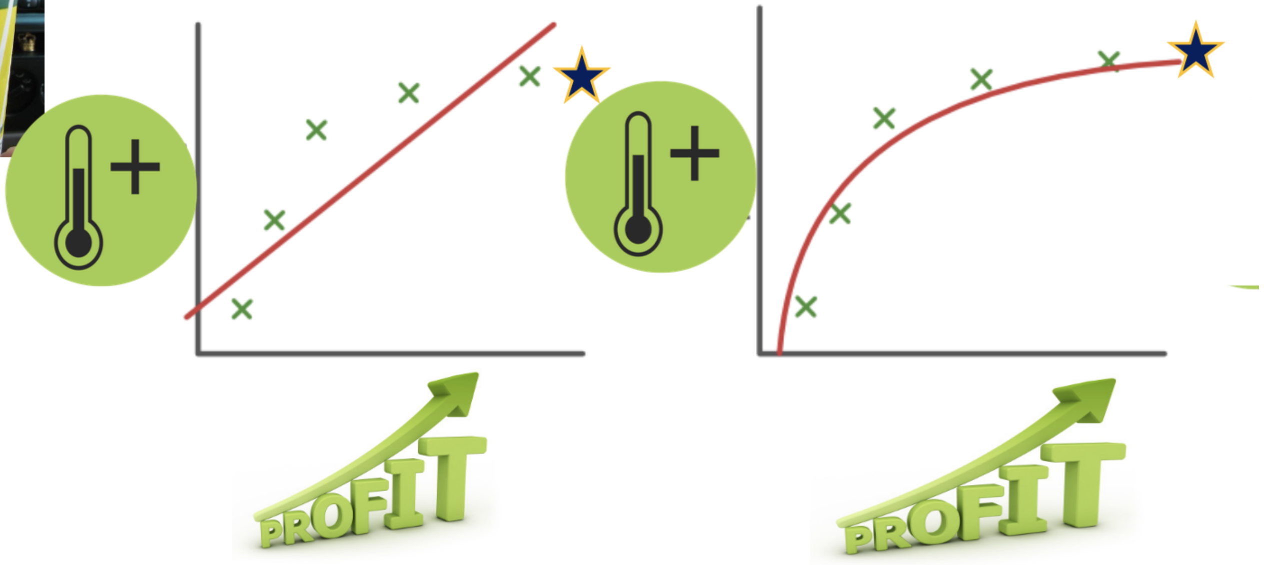
"My model is not doing that well on the given data and new data" ☹

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

How to know which function is the best?

$y = wx + b$

$y = w_1x^2 + w_2x + b$

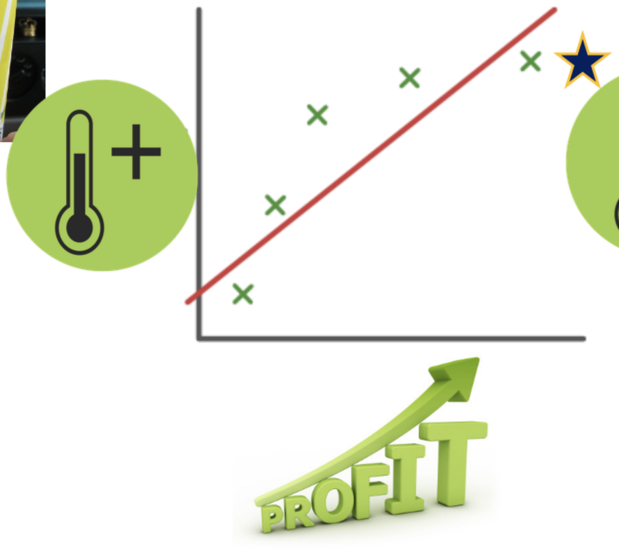


"My model is not doing that well on the given data and new data" ☹

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

How to know which function is the best?

$y = wx + b$



$y = w_1x^2 + w_2x + b$

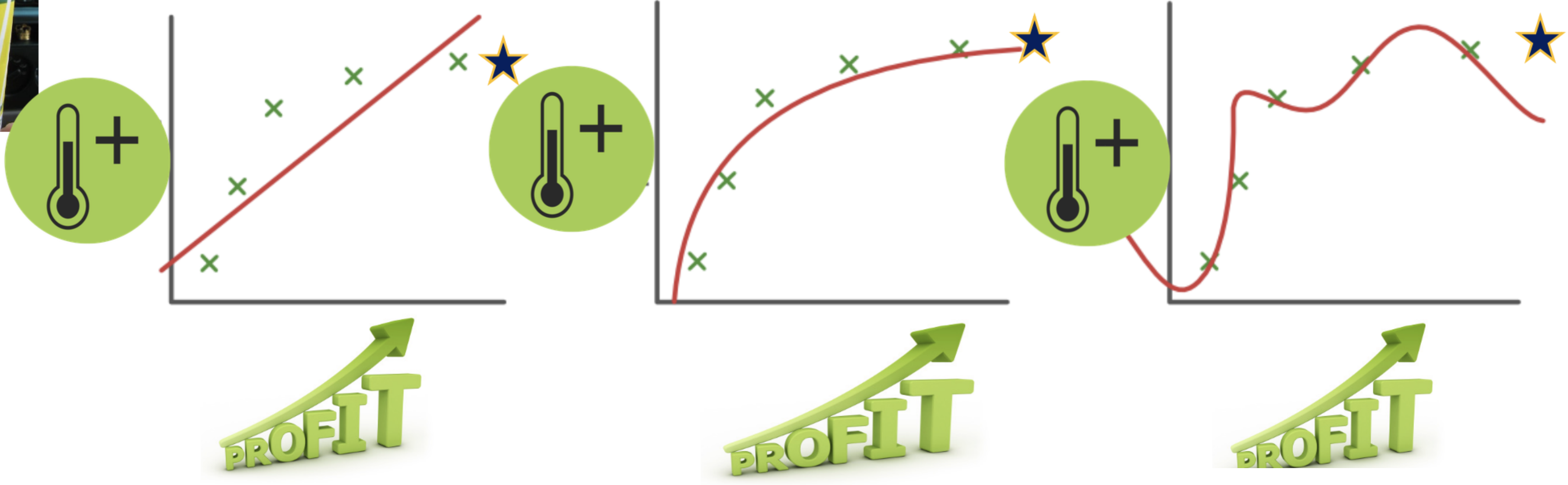
"My model is not doing that well on the given data and new data" ☹

"My model is doing well on the given data AND the new data point!! ☺

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

How to know which function is the best?

$y = wx + b$ $\quad$ $y = w_1x^2 + w_2x + b$ $\quad$ $y = w_1x^4 + w_2x^3 + w_3x^2 + w_4x + b$



"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!! ☺

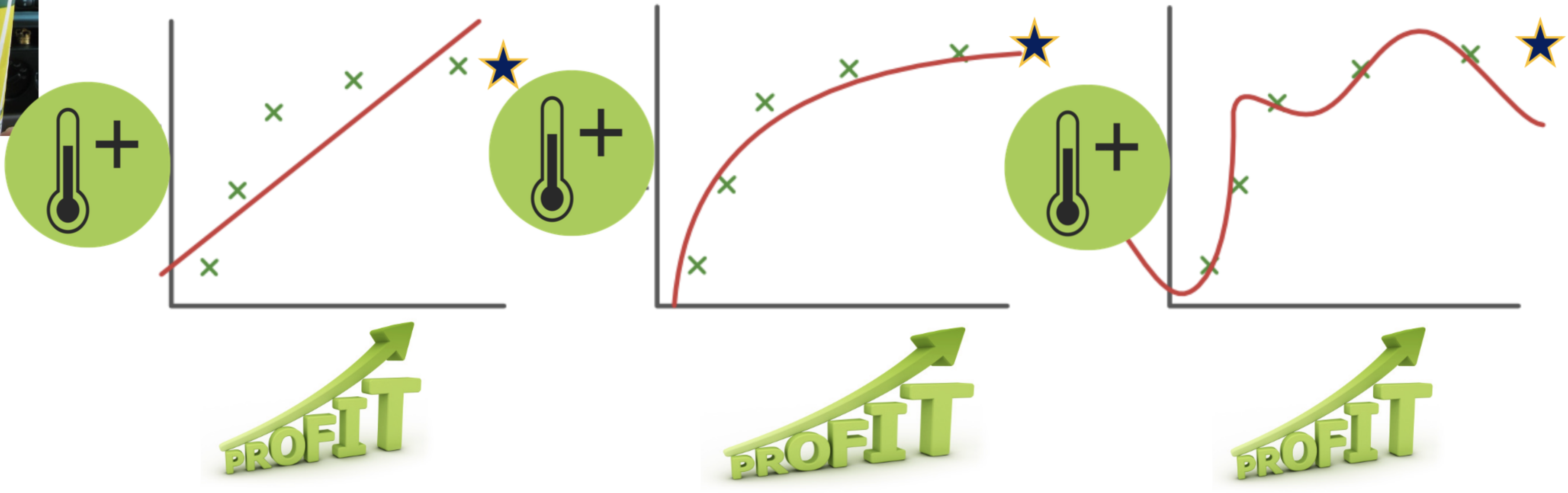
Image courtesy: https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

How to know which function is the best?

$y = wx + b$

$y = w_1x^2 + w_2x + b$

$y = w_1x^4 + w_2x^3 + w_3x^2 + w_4x + b$



"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!! ☺

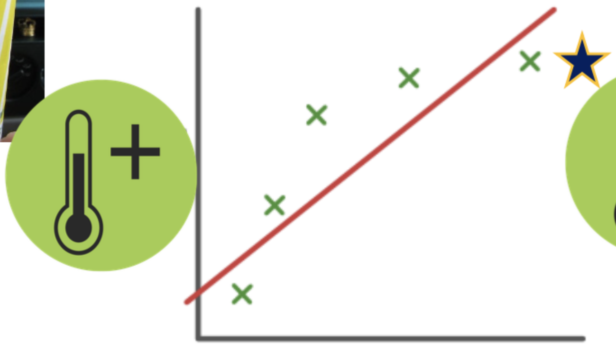
"The performance is bad on new data point" ☹

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

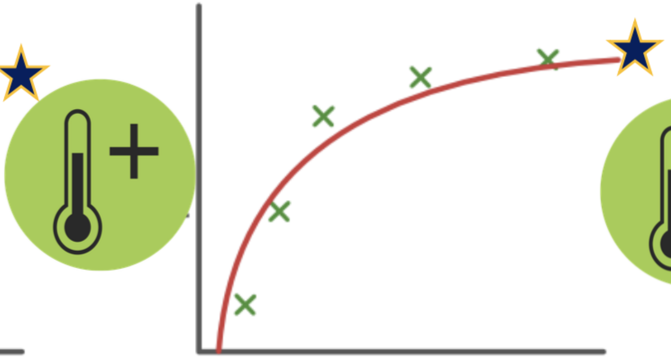
How to know which function is the best?

Underfit

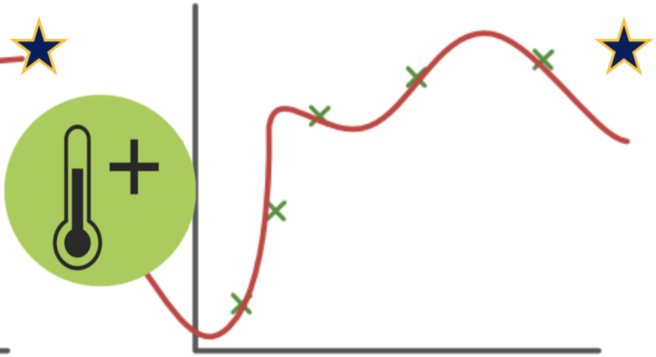
$y = wx + b$



$y = w_1x^2 + w_2x + b$



$y = w_1x^4 + w_2x^3 + w_3x^2 + w_4x + b$



"My model is doing really well on the given data!! ☺"

"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!! ☺

"The performance is bad on new data point" ☹

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

How to know which function is the best?

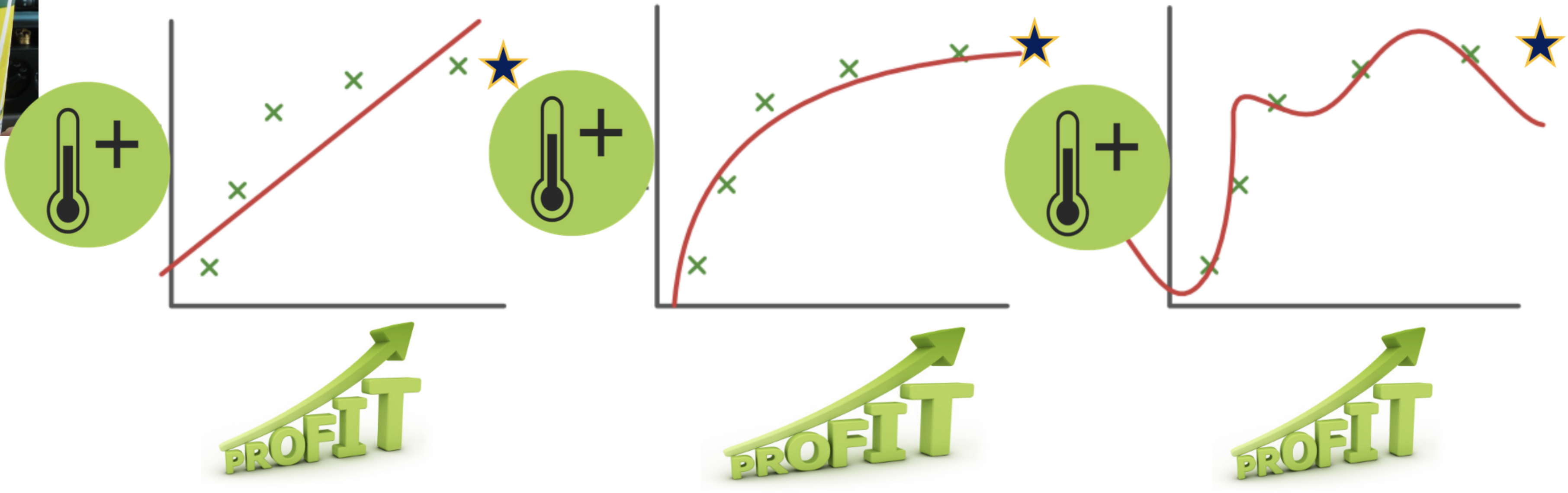
Underfit

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!! ☺

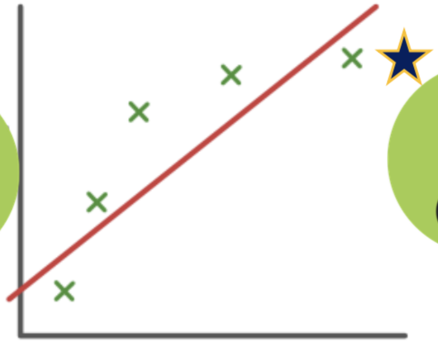
"The performance is bad on new data point" ☹

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

How to know which function is the best?

Underfit

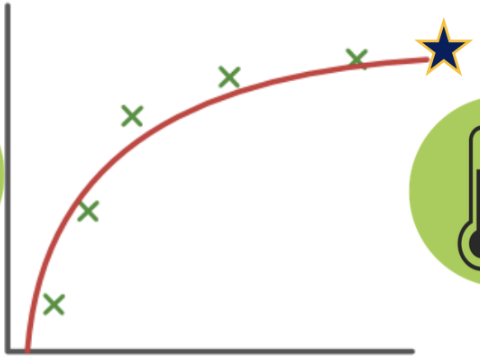
$y = wx + b$



"My model is not doing that well on the given data and new data" ☹

Good fit

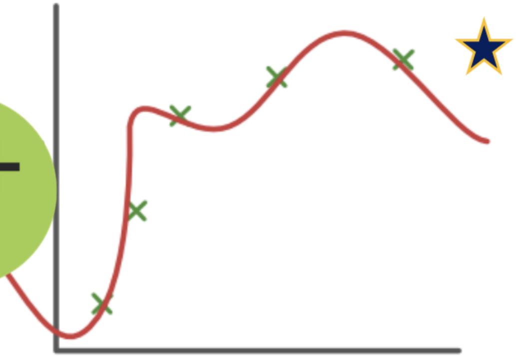
$y = w_1x^2 + w_2x + b$



"My model is doing well on the given data AND the new data point!! ☺

Overfit

$y = w_1x^4 + w_2x^3 + w_3x^2 + w_4x + b$



"My model is doing really well on the given data!! ☺

"The performance is bad on new data point" ☹

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

Model Complexity

- Model complexity refers to… the model's complexity
  - Polynomial regressions are more complex than linear regressions
- Models with higher complexity can approximate more function types well
- More complex functions also **tend** to overfit

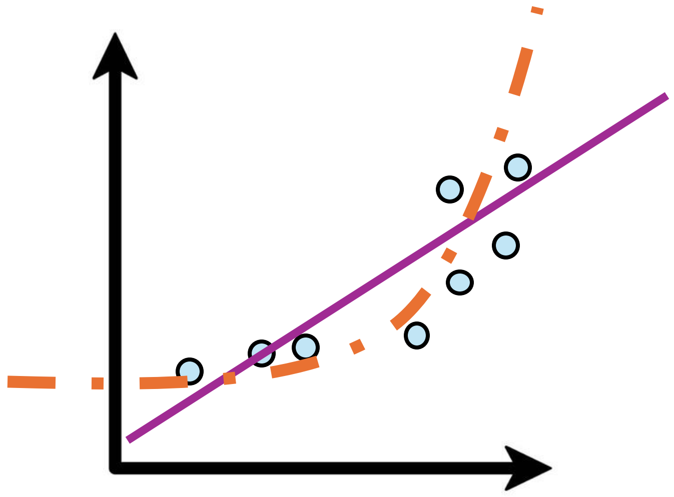
Open Question: A 100 degree polynomial tends to be way overfit. Neural Networks will be even more complex, why do neural networks not overfit?

How to know which function is the best?

$\mathbb{X}$

$x^{(1)}$
$x^{(2)}$
$x^{(3)}$
$x^{(4)}$
$x^{(5)}$
$x^{(6)}$
$x^{(7)}$
$x^{(8)}$

f_1 or f_2?

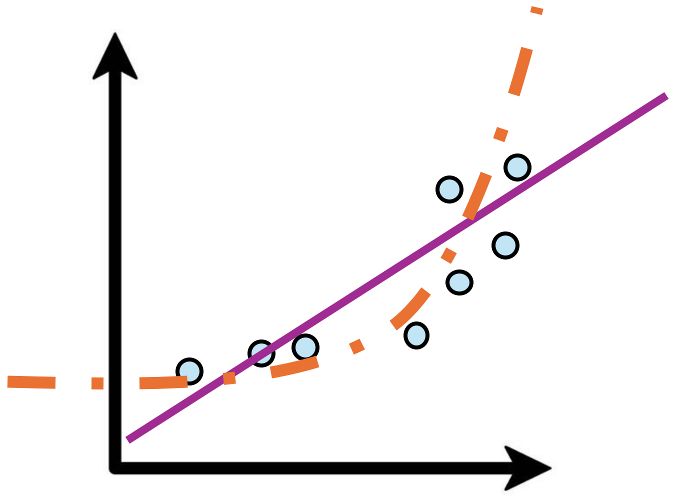


How to know which function is the best?

$\mathbb{X}$

$x^{(1)}$
$x^{(2)}$
$x^{(3)}$
$x^{(4)}$
$x^{(5)}$
$x^{(6)}$
$x^{(7)}$
$x^{(8)}$

f_1 or f_2?



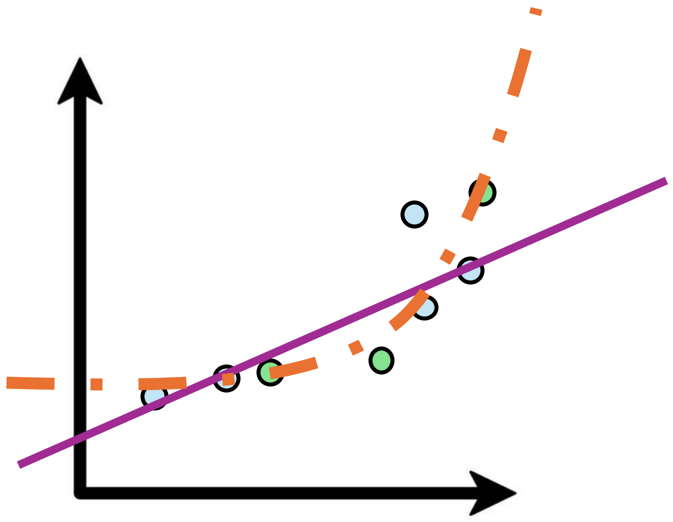
Compare MSE between them?

How to know which function is the best?

$\mathbb{X}$

$\mathbb{X}$	
$x^{(1)}$	Training Set
$x^{(2)}$	Training Set
$x^{(3)}$	Training Set
$x^{(4)}$	Training Set
$x^{(5)}$	Training Set
$x^{(6)}$	Test Set
$x^{(7)}$	Test Set
$x^{(8)}$	Test Set

f_1 or f_2?



How to know which function is the best?

$\mathbb{X}$

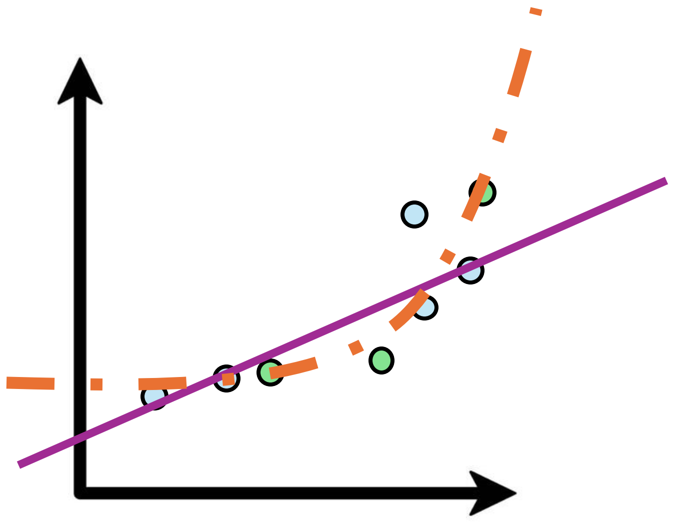
$x^{(1)}$
$x^{(2)}$
$x^{(3)}$
$x^{(4)}$
$x^{(5)}$

Training Set

$x^{(6)}$
$x^{(7)}$
$x^{(8)}$

Test Set

f_1 or f_2?



Compare MSE on what data?

How to know which function is the best?

$\mathbb{X}$

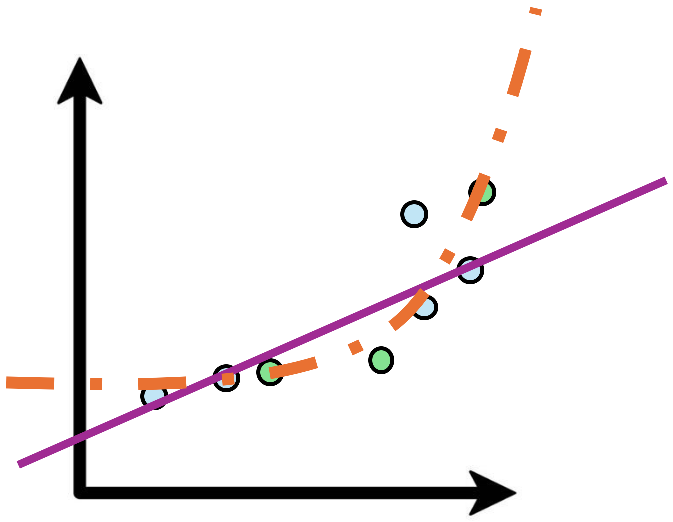
$x^{(1)}$
$x^{(2)}$
$x^{(3)}$
$x^{(4)}$
$x^{(5)}$

Training Set

$x^{(6)}$
$x^{(7)}$
$x^{(8)}$

Test Set

f_1 or f_2?



Compare MSE on what data?

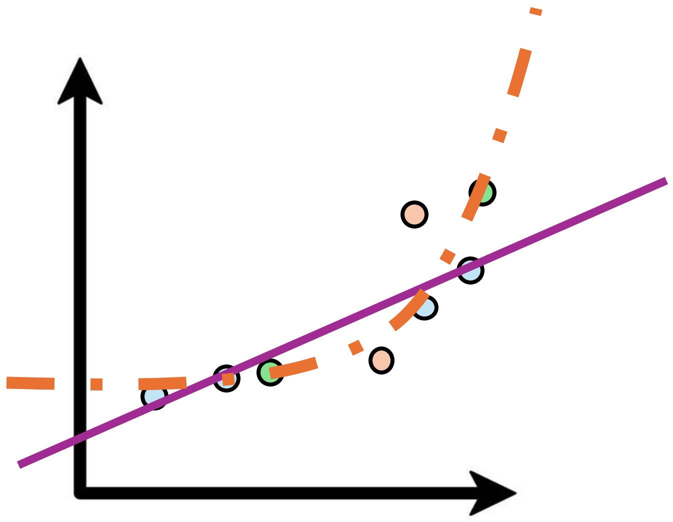
When might we want to overfit?

How to know which function is the best?

$\mathbb{X}$

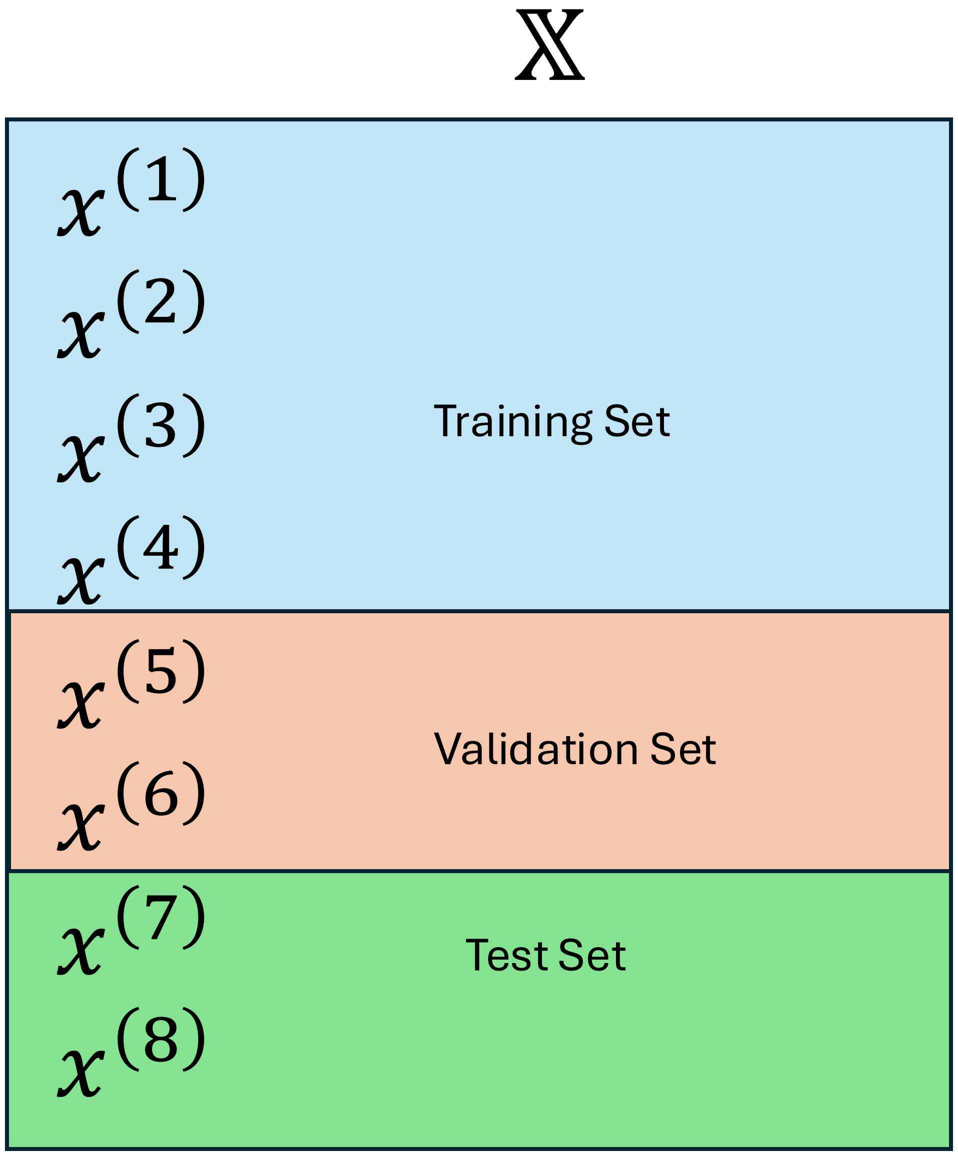
$x^{(1)}$ $x^{(2)}$ $x^{(3)}$ $x^{(4)}$	Training Set
$x^{(5)}$ $x^{(6)}$	Validation Set
$x^{(7)}$ $x^{(8)}$	Test Set

f_1 or f_2?



1. Train model on training set
2. Validate performance on validation set
3. Report results on test set

How to know which function is the best?



In this class

1. Train model on provided training data
2. Validate your model locally with validation set
3. Submit to Gradescope and we have a separate test set

In real world

1. Train model on provided training data
2. Validate your model locally with validation set
3. Deploy your model to real world and track performance

How to know which function is the best?

$\mathbb{X}$

$x^{(1)}$	Training Set
$x^{(2)}$	
$x^{(3)}$	
$x^{(4)}$	
$x^{(5)}$	Validation Set
$x^{(6)}$	
$x^{(7)}$	Test Set
$x^{(8)}$	

In this class

1. Train model on provided training data
2. Validate your model locally with validation set

it to Gradescope and we have a separate test set

Any questions?

In real world

1. Train model on provided training data
2. Validate your model locally with validation set
3. Deploy your model to real world and track performance

Other ways to improve performance

Collect additional information

	x_1 Temperature	x_2 Sunny?	x_3 Day of Week	y Profit
$x^{(1)}$	90	Yes	Sat	$200
$x^{(2)}$	80	No	Mon	$91
$x^{(3)}$	62	No	Wed	$54

Other ways to improve performance

Collect additional information

	x_1 Temperature	x_2 Sunny?	x_3 Day of Week	y Profit
$x^{(1)}$	90	Yes	Sat	$200
$x^{(2)}$	80	No	Mon	$91
$x^{(3)}$	62	No	Wed	$54

How can we represent binary variables?

Other ways to improve performance

Collect additional information

	x_1 Temperature	x_2 Sunny?	x_3 Day of Week	y Profit
$x^{(1)}$	90	Yes	Sat	$200
$x^{(2)}$	80	No	Mon	$91
$x^{(3)}$	62	No	Wed	$54

How can we represent binary variables?

$$x_2^{(k)} \in \{0, 1\}$$

Other ways to improve performance

Collect additional information

	x_1 Temperature	x_2 Sunny?	x_3 Day of Week	y Profit
$x^{(1)}$	90	Yes	Sat	$200
$x^{(2)}$	80	No	Mon	$91
$x^{(3)}$	62	No	Wed	$54

How can we represent binary variables?

How can we represent binary variables?

$$x_2^{(k)} \in \{0, 1\}$$

Other ways to improve performance

Collect additional information

	x_1 Temperature	x_2 Sunny?	x_3 Day of Week	y Profit
$x^{(1)}$	90	Yes	Sat	$200
$x^{(2)}$	80	No	Mon	$91
$x^{(3)}$	62	No	Wed	$54

How can we represent binary variables?

How can we represent binary variables?

Idea 1: Mon=0, Tue=1, Wed.=2

$$x_2^{(k)} \in \{0, 1\}$$

Other ways to improve performance

Collect additional information

	x_1 Temperature	x_2 Sunny?	x_3 Day of Week	y Profit
$x^{(1)}$	90	Yes	Sat	\$200
$x^{(2)}$	80	No	Mon	\$91
$x^{(3)}$	62	No	Wed	\$54

How can we represent binary variables?

How can we represent binary variables?

$$x_2^{(k)} \in \{0, 1\}$$

Idea 1: Mon=0, Tue=1, Wed.=2

The problem: Is Wednesday being 2x Tuesday meaningful?
Why use this ordering and not a random ordering?

Other ways to improve performance

Collect additional information

	x_1 Temperature	x_2 Sunny?	x_3 Day of Week	y Profit
$x^{(1)}$	90	Yes	Sat	\$200
$x^{(2)}$	80	No	Mon	\$91
$x^{(3)}$	62	No	Wed	\$54

How can we represent binary variables?

$x_2^{(k)} \in \{0, 1\}$

How can we represent binary variables?

Idea 1: Mon=0, Tue=1, Wed.=2

The problem: Is Wednesday being 2x Tuesday meaningful?
Why use this ordering and not a random ordering?

Idea 2: Use a series of binary variables
If day==Mon, x_4=1, else 0
If day==Tue, x_5=1, else 0
…

Other ways to improve performance

Collect additional information

	x_1 Temperature	x_2 Sunny?	x_3 Day of Week	y Profit
$x^{(1)}$	90	Yes	Sat	$200
$x^{(2)}$	80	No	Mon	$91
$x^{(3)}$	62	No	Wed	$54

How can we represent binary variables?

$x_2^{(k)} \in \{0, 1\}$

How can we represent binary variables?

Idea 1: Mon=0, Tue=1, Wed.=2

The problem: Is Wednesday being 2x Tuesday meaningful?
Why use this ordering and not a random ordering?

Idea 2: Use a series of binary variables
If day==Mon, x_4=1, else 0
If day==Tue, x_5=1, else 0
…

"One-Hot Vector": Turn categorical variables into a vector of binary variables

Key Ideas Review

How to represent Input/Output

- Input/Output need to be numbers
- Classification: Predicting categorical outputs (y is discrete)
- Regression: Predicting numerical outputs (y is continuous)

Supervised Learning

- Goal of supervised learning: Find good approximation of data
- Train on Training set
- Evaluate on validation Set
- But what we really care about is performance on test set

Important Questions

- Linear Regression seeks to find a best fit function by minimizing MSE.
  - How can we find the **best** possible linear regression?
  - Are there **more than one** best fit line?
  - **Why did we choose MSE**? Why not Mean Error? Are there other loss functions that make sense?
- You can see how we can convert images to numbers, since pixels are stored as (r, g, b) values. But what about **other input types** like natural language? The protein for protein fold prediction? A chess board?