# Machine Learning

## Week 1 - Introduction

**Dr. Temitayo Olugbade**

UNIVERSITY OF SUSSEX

# Content today: Introduction

- ❑ **Module information**
- ❑ Machine learning in our world
- ❑ Regression basics
- ❑ Classification basics

# Learning goals

- To gain comprehensive understanding of key aspects of machine learning and standard methods
- To become aware of relevant issues and current challenges in machine learning
- To develop skills for systematically and creatively building and evaluating machine learning models
- To practice appropriate data preparation to address a given problem & selection of the most suitable techniques to address the problem

# Canvas

## Homepage

### Key information and resources

- Module content
- Module information
- Reading list
- Module contacts
- Recordings
- Assessments and Feedback

**Quick links:** Study Timetable | Assessment Deadlines | Progress & Feedback | Results

**Support:** ITS Service Desk | Library | Student Hub | Student Centre | Disability Support

https://canvas.sussex.ac.uk/courses/31315/wiki

# Teaching & Support

- Teaching
  - Lecture
  - Lab (Lab notebooks & Ungraded quizzes)
- Support
  - Student/office hours for meeting with me
  - Teaching assistants (TAs) available every lab session
  - Feedback for some ungraded assignments
  - Suggested readings
  - Maths and stats refresher resources
  - Peer Assisted Learning
- For details, see https://canvas.sussex.ac.uk/courses/31315/pages/module-information

# Syllabus

| | |
|---|---|
| ❑ Introduction | **Week 1** |
| ❑ Supervised learning I & II & III | **Weeks 2-4** |
| ❑ Model validation I & II | **Weeks 5-6** |
| ❑ AI ethics (& Coursework release) | **Week 7** |
| ❑ Advanced neural networks | **Week 8** |
| ❑ Attention | **Week 9** |
| ❑ Beyond supervised learning | **Week 10** |
| ❑ Introduction to reinforcement learning | **Week 11** |

# Skills needed

- ❑ Critical thinking & reflection
(see https://www.sussex.ac.uk/skills-hub/critical-thinking#main)

- ❑ Curiosity
(see https://www.linkedin.com/learning/using-questions-to-foster-critical-thinking-and-curiosity/benefits-of-being-curious?resume=false&u=83331314)

- ❑ Programming
(see Autumn modules - Programming through Python; Data Science Research Methods)

- ❑ Maths & Statistics
(see Autumn modules - Mathematics & Computational Methods for Complex Systems; Data Science Research Methods)

# Assessment

- ❑ 100% coursework
- ❑ You will be:
  - given a dataset & a machine learning problem.
  - expected to use your knowledge and practice from the lectures and labs to solve the given tasks.
  - required to present your solution in form of:
    - a report – format will be provided in Week 7;
    - code; and
    - machine learning output.
- ❑ Details will be published in Week 7

# Academic integrity

- You must NEVER pass off any part of someone else's (or AI generated) work as yours.
- For more details, see https://canvas.sussex.ac.uk/courses/31315/pages/module-information

# A caution about data

- ❑ **Copyright laws** - You must NEVER use any data without clear permission, e.g. CC BY, for your use purpose. Availability is NOT necessarily license to use! You could be in breach of copyright otherwise.

- ❑ **Data protection law** - UK has regulations (GDPR) that must be followed for use of personal data, i.e. data about an identified living person.

- ❑ **Ethics** - Widely accepted ethical values include strict rules about the use of data about/from humans in research (including any university work).

# Content today: Introduction

- ❑ Module information
- ❑ **Machine learning in our world**
- ❑ Regression basics
- ❑ Classification basics

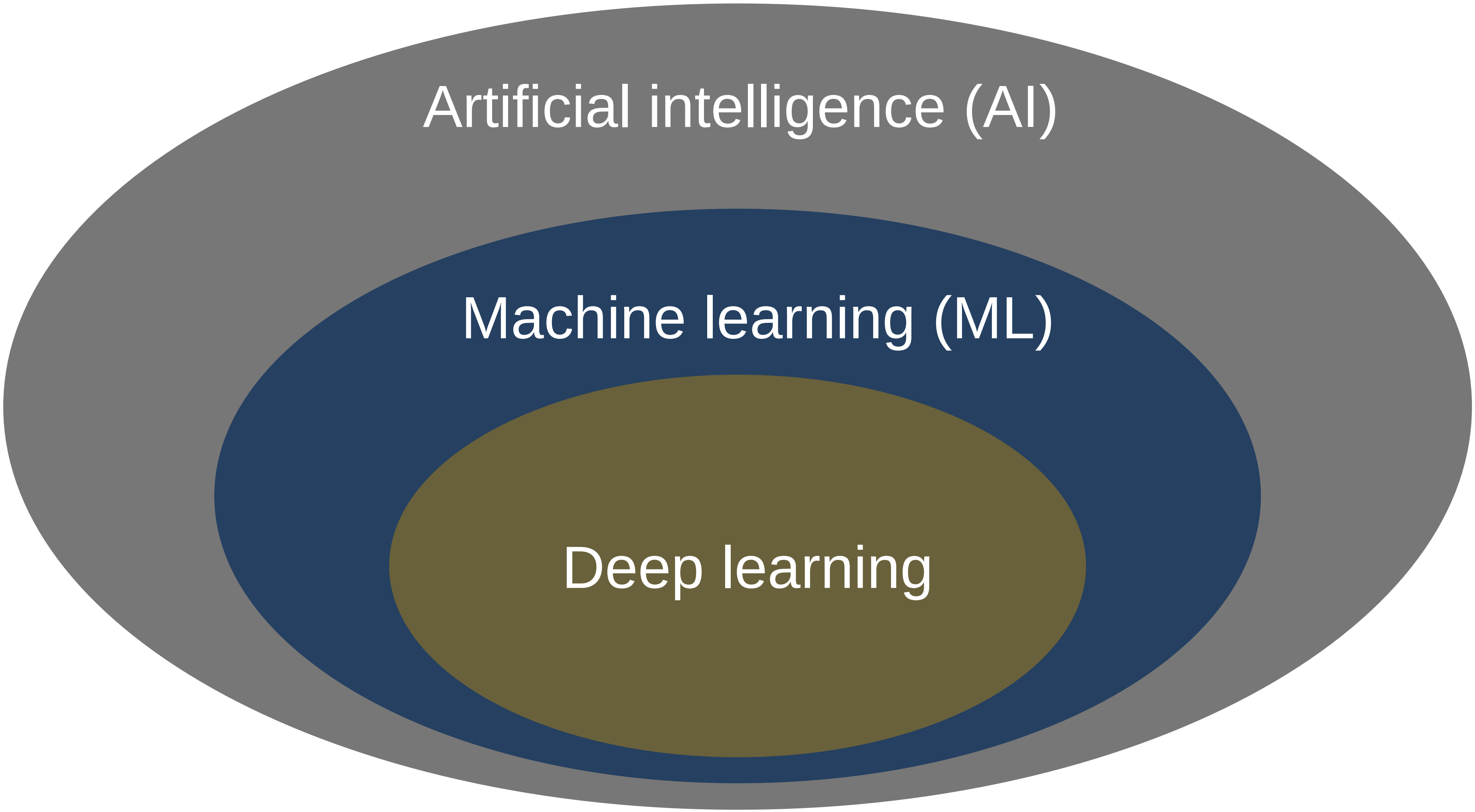
# Machine learning (1)

development of software that performs some task (or sets of tasks) based on its own learnt experience

# Machine learning (2)

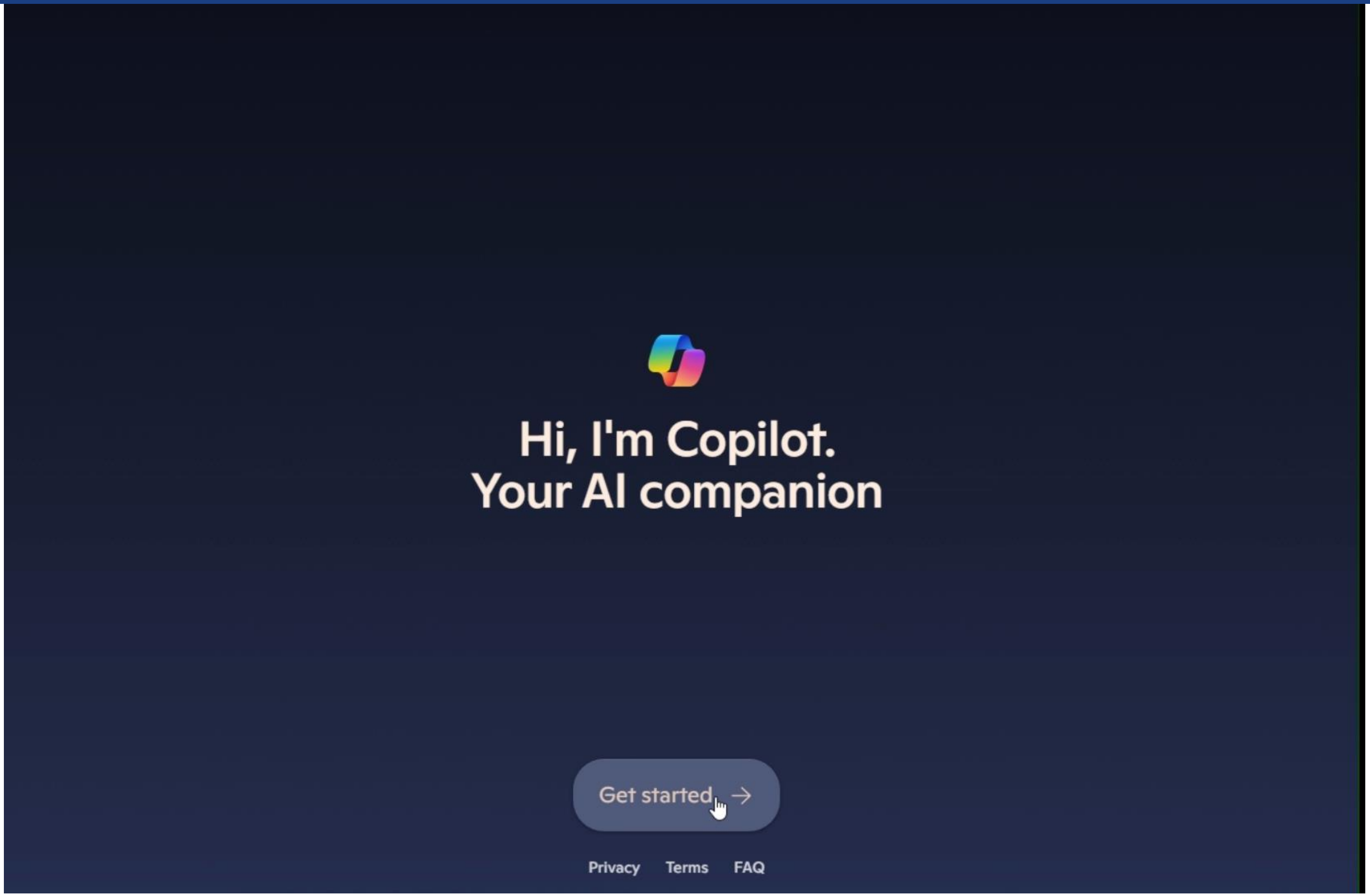
creation of a mathematical model that can deduce appropriate response to new stimuli from its previous experience

# ML vs AI


Artificial intelligence (AI)
Machine learning (ML)
Deep learning

# Example products & services that use ML

# Microsoft Copilot




Natural language processing

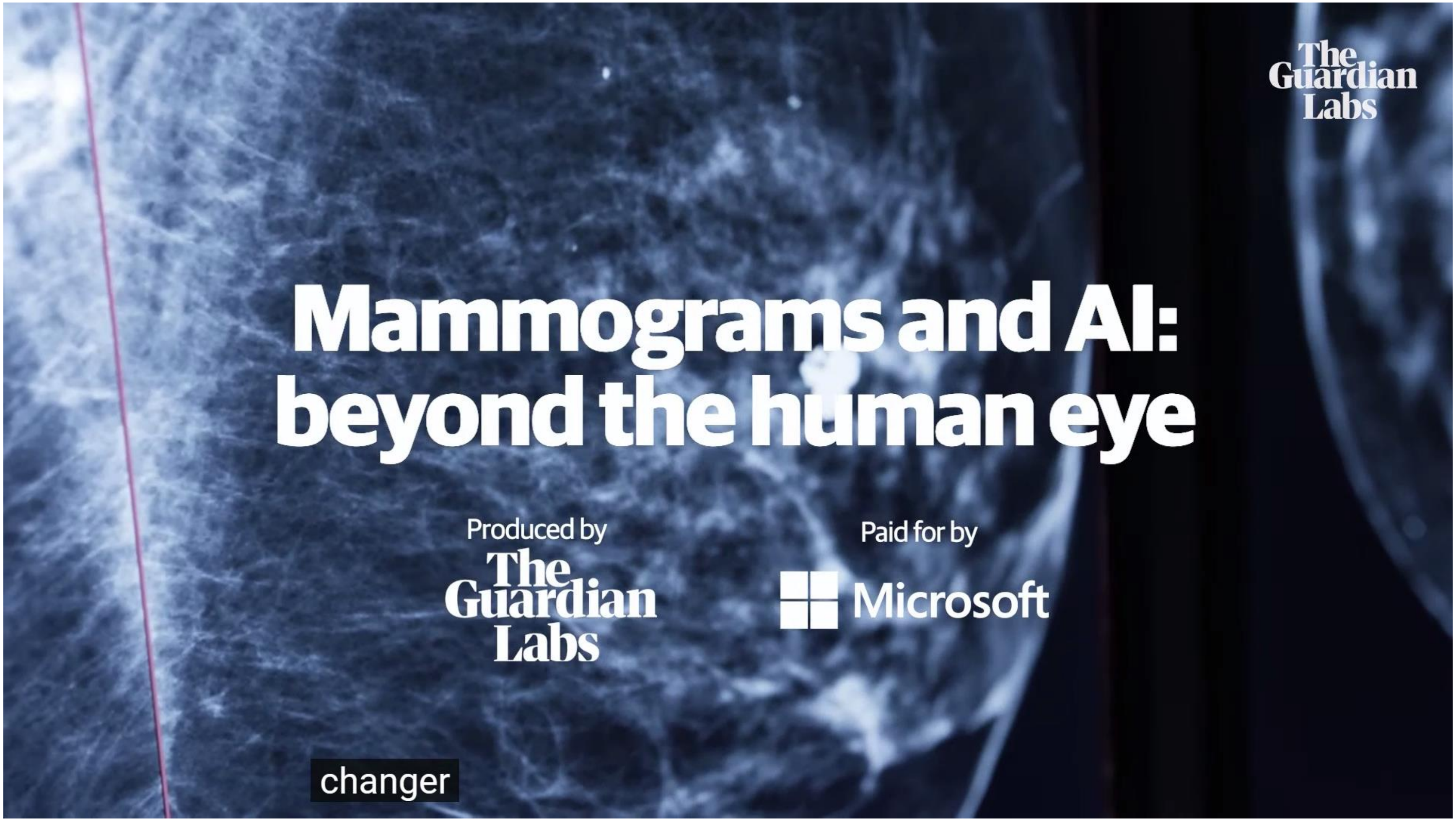
(https://copilot.microsoft.com/)

# Bosch Dryad


## Sensor data analysis

(https://www.youtube.com/watch?v=A4DK8jQnHbQ&t=1s)

# Kheiron Mia




## Computer vision

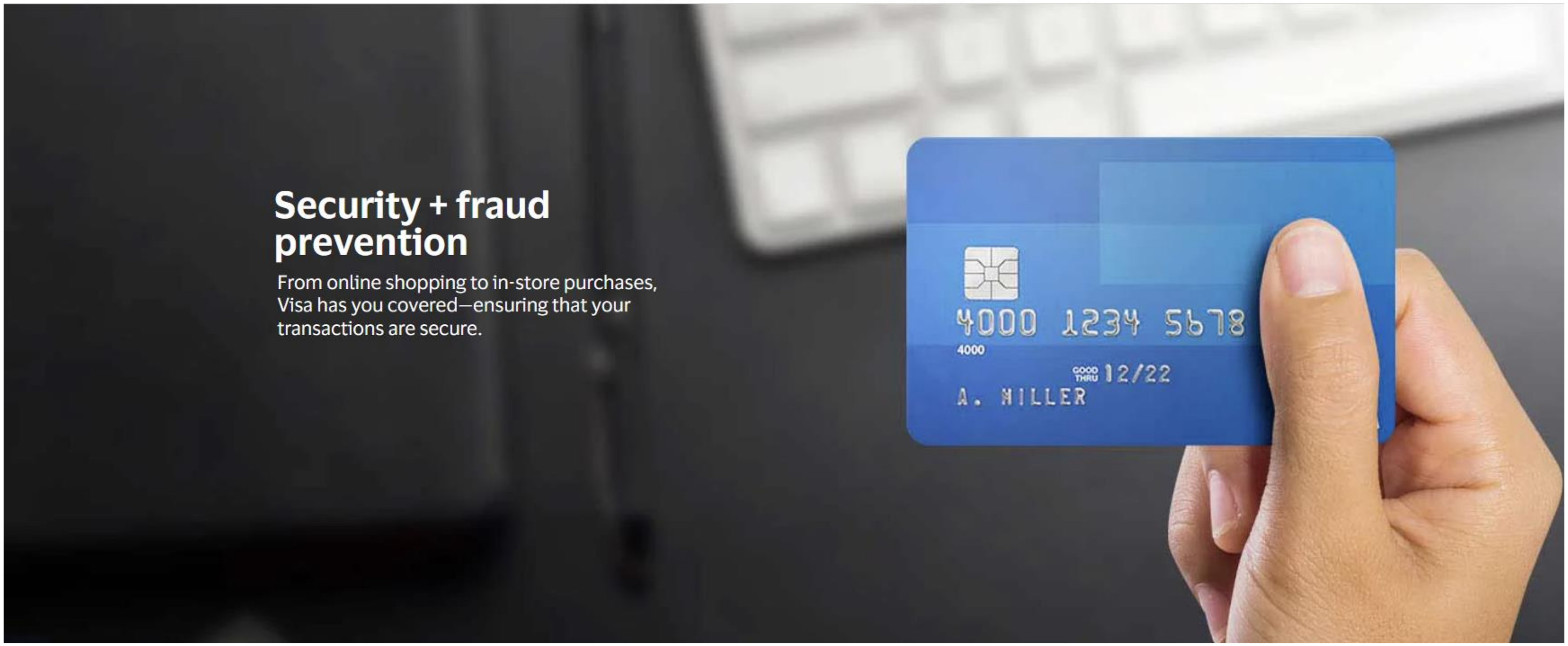
(https://www.youtube.com/watch?v=jUNo27MAfZM&t=1s)

# ML for surveillance and security


Veesion

(https://veesion.io/)

# ML for banking and finance




Visa

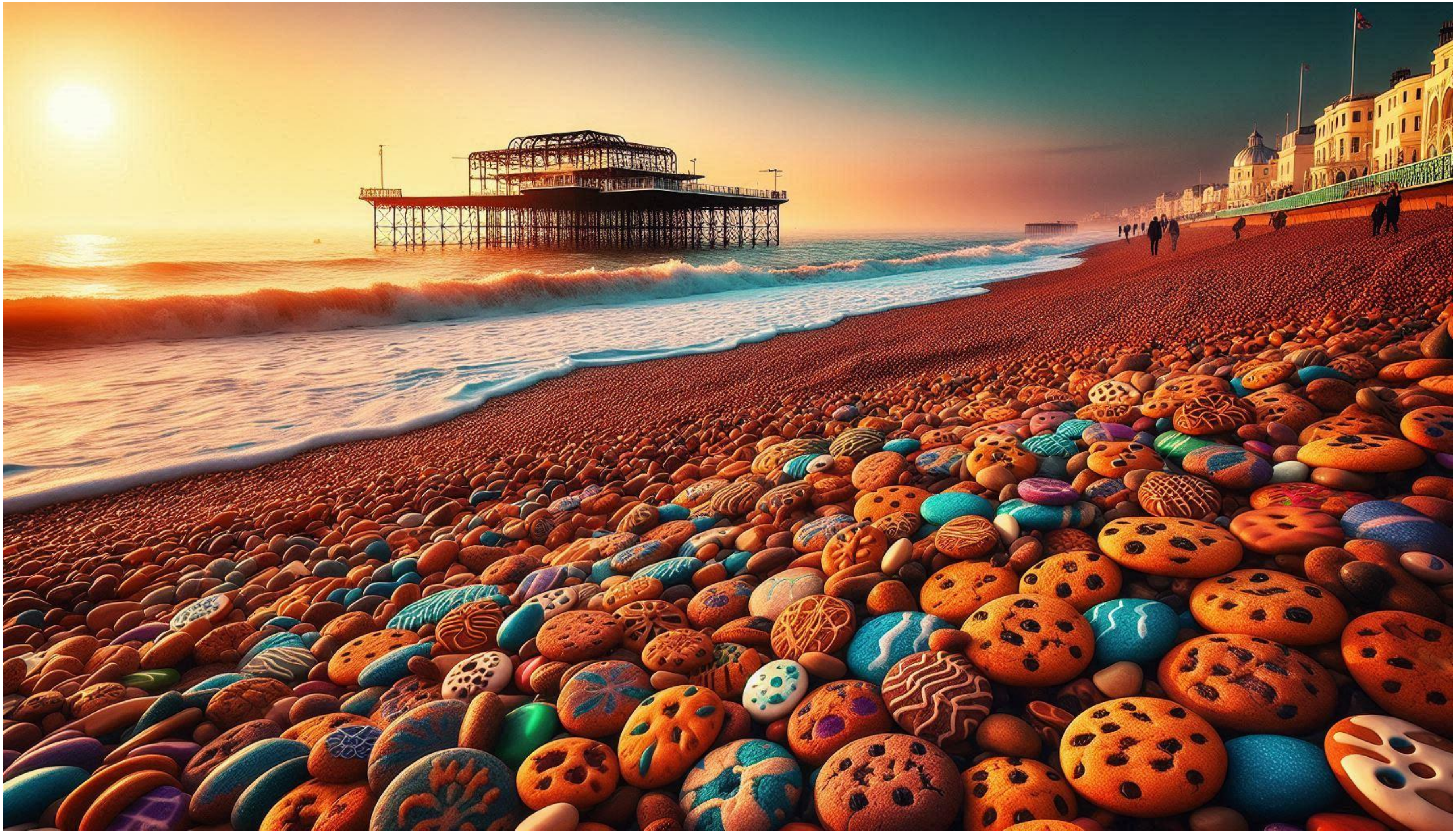
(https://usa.visa.com/run-your-business/visa-security/risk-solutions/authorization-optimization.html)

# ML for transport


Tesla

(https://www.tesla.com/en_GB/autopilot/)

# ML in arts and design



Generated by AI – *Microsoft Designer*

(**prompt** – "Brighton beach showing the West Pier with the pebbles looking like cookies on cold January morning with the sun setting on the sea and in warm colors in a hyper surreal style.")

# Reflect: AI ethics

**In what ways could AI threaten fairness and safety in the society?**

Contribute your thoughts to the discussion on Canvas:

Topic: Ethical AI - Fairness (sussex.ac.uk)

Topic: Ethical AI - Safety (sussex.ac.uk)

# Content today: Introduction

- Module information
- Machine learning in our world
- **Regression basics**
- Classification basics

# ML & Data

- Data is central to machine learning
  - it is the source of experience and ‘learning’ in ML
  - it is typically of two parts:
    - label(s)/output - $\boldsymbol{y}$
    - features/input - $\boldsymbol{x}$

- Recall – ML = creation of a mathematical model that can deduce appropriate response (**label(s)**) to new stimuli (**features**) from its previous experience

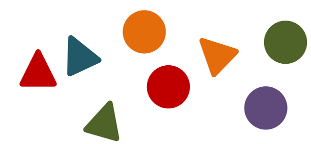
# Some ML concepts to start with

- **ML model** – software or mathematical model that:
  - takes in some input (features);
  - gives some output (label(s)); and
  - has capacity to learn from experience (data).
- **Training** – the 'learning' process when the ML model gains 'experience'
- **Inference** – giving a (trained) ML model some input and prompting it to give appropriate output

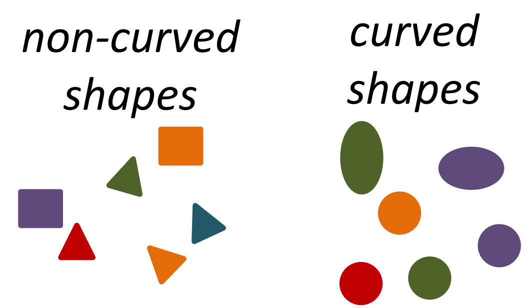
# Types of learning

- Supervised learning
  - Training data includes labels

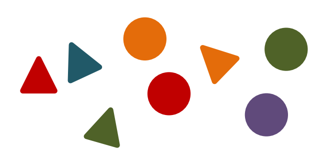

- Unsupervised learning
  - Training data does not include labels



- Semi-supervised learning
  - Training data includes labels but not those needed at inference time




- Self-supervised learning
  - The ‘labels’ are the features themselves, or some trivial derivative of the features



# Some maths notations to start with

- $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N}$

A set of $N$ elements, and the $n$th element in the set is a pair of tensors $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$

A tensor is a $d$-dimensional element.
$d$ could be any positive integer, and the tensor would usually be made up of real or integer values.
A scalar has $d = 0$; a vector has $d = 1$; a matrix has $d = 2$.

- $\boldsymbol{x}_n \in \mathbb{R}^{D_x}$

Denotes that $\boldsymbol{x}_n$ is made up of $D_x$ real values

- <u>Recall</u> – $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ would usually denote features/input and labels/output respectively

# Supervised learning: Regression

- Consider that there exists **data** instances $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N}$, $\boldsymbol{x}_n \in \mathbb{R}^{D_x}$, $\boldsymbol{y}_n \in \mathbb{R}^{D_y}$

- The goal is to find a **model**/function $f(.)$ that takes in as input $\boldsymbol{x}_n$ and outputs $\hat{\boldsymbol{y}}_n$ such that:

  - $f(\boldsymbol{x}_n) = \hat{\boldsymbol{y}}_n \approx \boldsymbol{y}_n$; and

  - $f(.)$ is **generalizable** beyond $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N}$

    i.e. $f(\boldsymbol{x}_m) = \hat{\boldsymbol{y}}_m \approx \boldsymbol{y}_m$ where $\boldsymbol{x}_m \notin \{\boldsymbol{x}_n\}_{n=1}^{N}$

# Toy data for illustration

- $\{\boldsymbol{x}_n\}_{n=1}^{3}, D_x = 4$

i.e. 3 data instances, each with 4 features

| n | Temperature (°C) | Relative humidity (%) | Wind speed (km/h) | Rain (mm) |
|---|---|---|---|---|
| 1 | 23 | 21 | 10 | 0 |
| 2 | 40 | 89 | 6 | 1 |
| 3 | 35 | 60 | 23 | 15 |

- $\{y_n\}_{n=1}^{3}, D_y = 1$

i.e. 1 label for each of 3 data instances

| n | Fire weather index |
|---|---|
| 1 | 0 |
| 2 | 30 |
| 3 | 15 |

# Generalizability

| | Temperature (°C) | Relative humidity (%) | Wind speed (km/h) | Rain (mm) | Fire weather index |
|---|---|---|---|---|---|
| training data | 23 | 21 | 10 | 0 | 0 |
| | 40 | 89 | 6 | 1 | 30 |
| | 35 | 60 | 23 | 15 | 15 |
| *unseen* data | 25 | 52 | 17 | 13 | ? |
| | 41 | 48 | 9 | 8 | ? |

**Training data** – Data used to train a model
**Unseen data** – Data not 'seen' by the model during training

# Re: Supervised learning (regression)

- Consider that there exists **training data** instances $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N}$, $\boldsymbol{x}_n \in \mathbb{R}^{D_x}$, $\boldsymbol{y}_n \in \mathbb{R}^{D_y}$
- The goal is to find a **model**/function $f(.)$ that takes in as input $\boldsymbol{x}_n$ and outputs $\hat{\boldsymbol{y}}_n$ such that:
  - $f(\boldsymbol{x}_n) = \hat{\boldsymbol{y}}_n \approx \boldsymbol{y}_n$; and
  - $f(.)$ is **generalizable** to **unseen data instances**

    i.e. $f(\boldsymbol{x}_m) = \hat{\boldsymbol{y}}_m \approx \boldsymbol{y}_m$ where $\boldsymbol{x}_m \notin \{\boldsymbol{x}_n\}_{n=1}^{N}$

# Basic linear model

Given $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N}$ , $\boldsymbol{x}_n \in \mathbb{R}^{D_x}$, $\boldsymbol{y}_n \in \mathbb{R}^{D_y}$

$$f(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{w} + b = \hat{\boldsymbol{y}}$$

<u>Notations</u>

- $f(\cdot)$ – basic linear model
- $\boldsymbol{x}$ – features (or model input)
- $\hat{\boldsymbol{y}}$ – predicted labels/targets (or model output)
- $\boldsymbol{w}$, $b$ – weights, bias (or model parameters)

# Linear regression

Given $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N}$, $\boldsymbol{x}_n \in \mathbb{R}^{D_x}$, $\boldsymbol{y}_n \in \mathbb{R}^{D_y}$

$$f(\boldsymbol{x}) = \hat{\boldsymbol{y}} = \boldsymbol{x}\boldsymbol{w} + b$$

- $\boldsymbol{y}_n \in \mathbb{R}^{D_y}$ (i.e. real-valued labels) implies that the supervised learning is a **regression** task
- $f(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{w} + b$ (i.e. basic linear model) with $\boldsymbol{y}_n \in \mathbb{R}^{D_y}$ implies a **linear regression** model

# Basic linear model (reframed)

$$f(\boldsymbol{x}) = \hat{\boldsymbol{y}} = \boldsymbol{x}\boldsymbol{w} + b$$

- In matrix form

$$\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_N \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{bmatrix} \times \begin{bmatrix} w_1 \\ \vdots \\ w_D \end{bmatrix} + \begin{bmatrix} b \\ \vdots \\ b \end{bmatrix}$$

- Rewriting to absorb $b$ in $\boldsymbol{w}$

$$f(\boldsymbol{x}) = \hat{\boldsymbol{y}} = \boldsymbol{x}\boldsymbol{w}$$

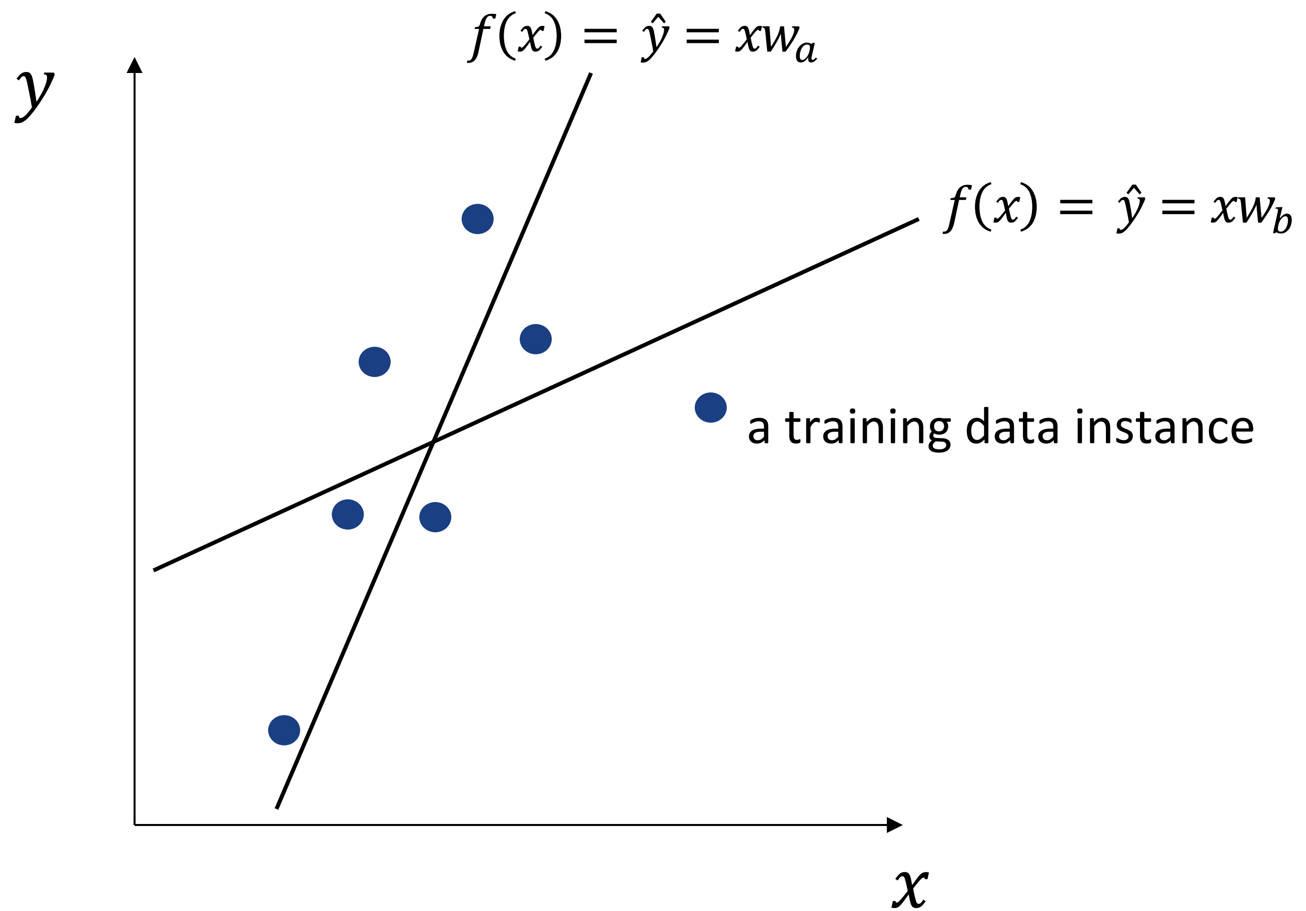
$$\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_N \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1D} & 1 \\ \vdots & \ddots & \vdots & \\ x_{N1} & \cdots & x_{ND} & 1 \end{bmatrix} \times \begin{bmatrix} w_1 \\ \vdots \\ w_D \\ b \end{bmatrix}$$

# Using the toy data for illustration

| Temperature (°C) | Relative humidity (%) | Wind speed (km/h) | Rain (mm) | Fire weather index |
|---|---|---|---|---|
| 23 | 21 | 10 | 0 | 0 |
| 40 | 89 | 6 | 1 | 30 |
| 35 | 60 | 23 | 15 | 15 |

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_{N=3} \end{bmatrix} = \begin{bmatrix} 23 & 21 & 10 & 0 & 1 \\ 40 & 89 & 6 & 1 & 1 \\ 35 & 60 & 23 & 15 & 1 \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_{D=4} \\ b \end{bmatrix}$$

# A linear model visualization: other toy data




$N = 7,\ D_x = 1,\ D_y = 1$

# Finding optimal model parameters

- Measure the model error (referred to as '**loss**')

  i.e. how far $\boldsymbol{y}$ ('true' label) is from $\widehat{\boldsymbol{y}}$ (predicted output)

$$L_2(\boldsymbol{w}) = \frac{1}{N}\sum_{n=1}^{N}(\widehat{\boldsymbol{y}} - \boldsymbol{y}_n)^2 = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n\boldsymbol{w} + b - \boldsymbol{y}_n)^2$$

$$L_2(\boldsymbol{w}) = \frac{1}{N}\|\boldsymbol{x}\boldsymbol{w} - \boldsymbol{y}\|^2$$

$L_2(\cdot)$ denotes L2 loss function (aka mean-squared error)

- Optimal model parameters $(\boldsymbol{w}, b)$ minimize the loss

- Training is the process of optimizing $\boldsymbol{w}$ and $b$

i.e. training is the process of minimizing the model loss

# Minimizing the loss

- The minimum of a function is when its gradient (derivative) is zero, i.e.

$$0 = \frac{d\mathrm{L}_2(\mathbf{w})}{d\boldsymbol{w}}$$

*expanding substituting* $\mathrm{L}_2(\mathbf{w})$ *with its value*

$$0 = \frac{1}{N} \times \frac{d(\|\boldsymbol{xw} - \boldsymbol{y}\|^2)}{d\boldsymbol{w}}$$

*expanding the numerator of the right hand side*

$$0 = \frac{d\left((\boldsymbol{xw} - \boldsymbol{y})^T(\boldsymbol{xw} - \boldsymbol{y})\right)}{d\boldsymbol{w}}$$

# Minimizing the loss (2)

$$0 = \frac{d\left((\boldsymbol{x}\boldsymbol{w} - \boldsymbol{y})^T(\boldsymbol{x}\boldsymbol{w} - \boldsymbol{y})\right)}{d\boldsymbol{w}}$$

*further expanding and collecting like terms*

$$0 = \frac{d\left(\boldsymbol{w}^T\boldsymbol{x}^T\boldsymbol{x}\boldsymbol{w} - 2\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y}\right)}{d\boldsymbol{w}}$$

*applying the derivative with respect to **w** to the right hand side*

$$0 = 2\boldsymbol{x}^T\boldsymbol{x}\boldsymbol{w} - 2\boldsymbol{x}^T\boldsymbol{y}$$

*making **w** the subject of the formula*

$$\boldsymbol{w} = (\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}^T\boldsymbol{y}$$

# Basic regression loss functions

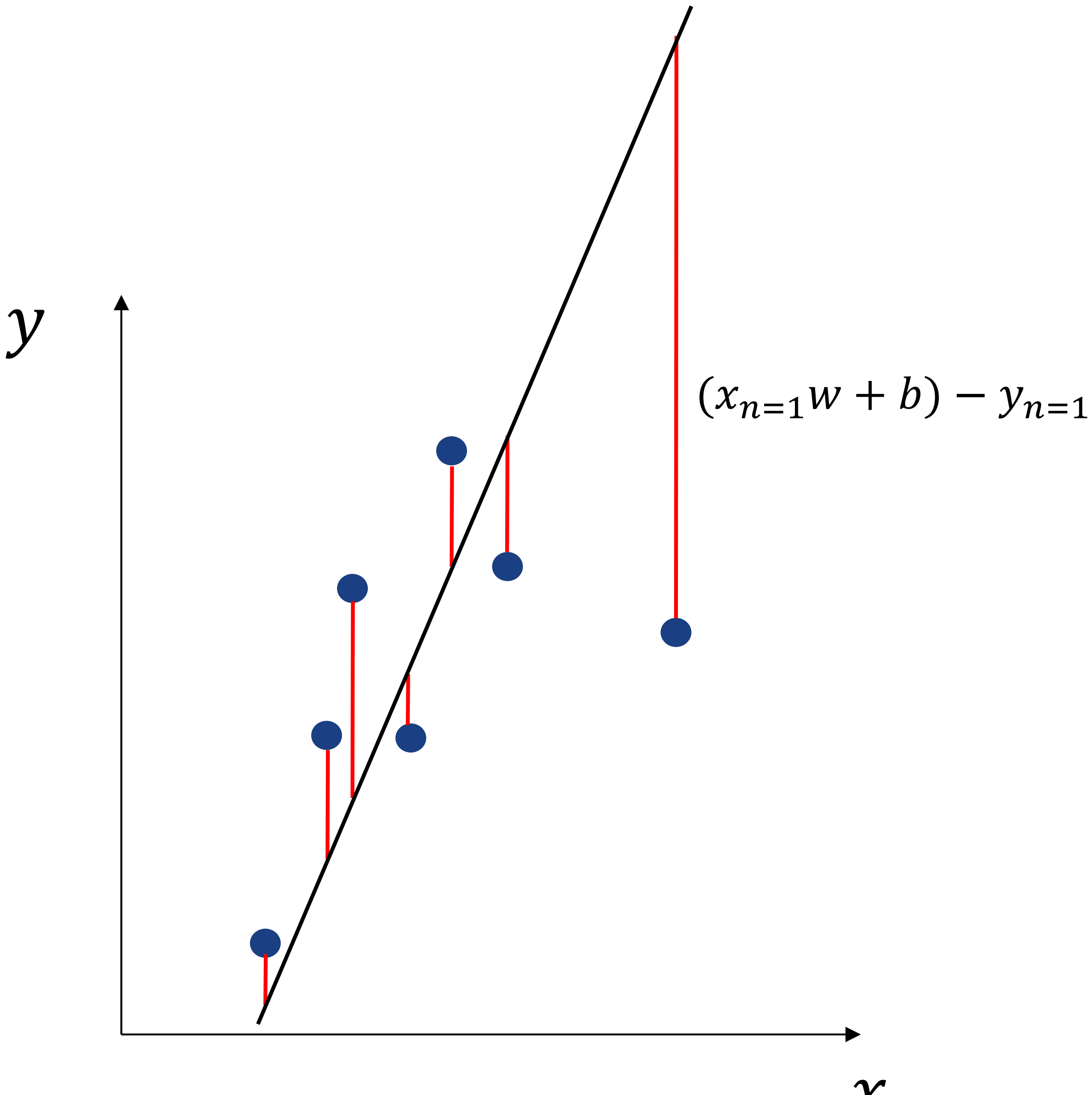
- L2 loss  (mean squared error)

$$L_2(\boldsymbol{w}) = \ \|\boldsymbol{x}\boldsymbol{w} - \boldsymbol{y}\|^2$$

- L1 loss (mean absolute error)

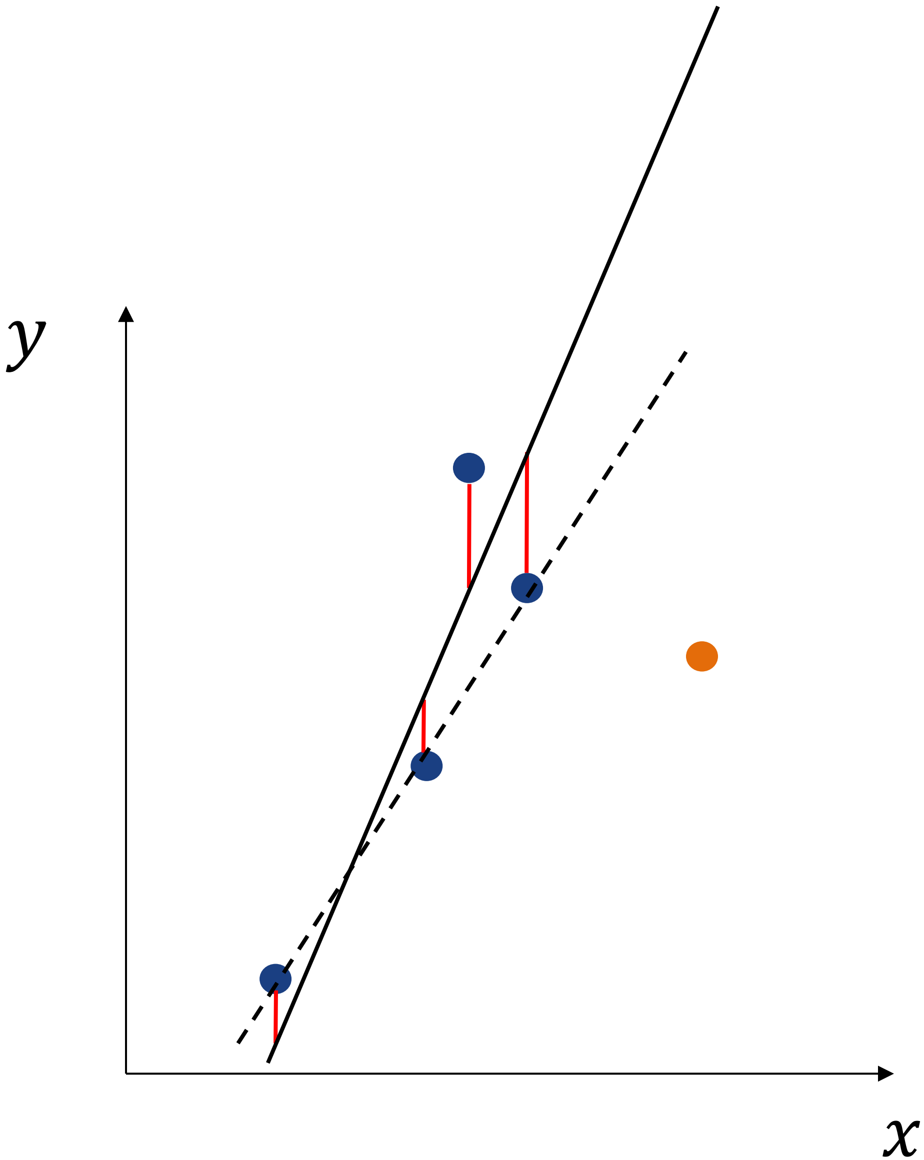
$$L_1(\boldsymbol{w}) = \ |\boldsymbol{x}\boldsymbol{w} - \boldsymbol{y}|$$

# Error visualization



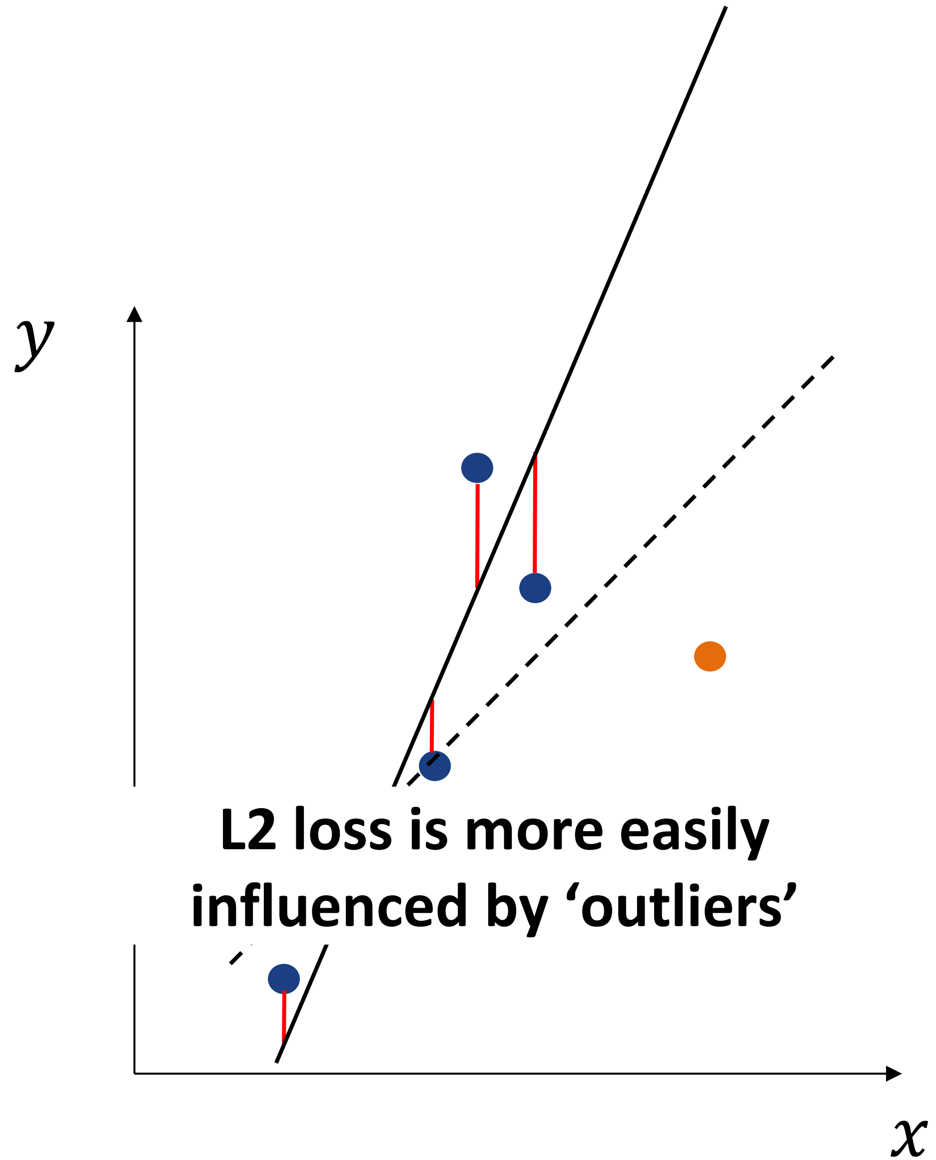
# Effect of an outlier for L1 loss

NB: The optimized models (regression lines) here are only illustration of expected characteristics but not plot based on experiment.



# Effect of an outlier for L2 loss

NB: The optimized models (regression lines) here are only illustration of expected characteristics but not plot based on experiment.




**L2 loss is more easily influenced by 'outliers'**

# Differentiability of the L2 loss

$$\frac{d\mathrm{L}_2(\mathbf{w})}{d\boldsymbol{w}} = \frac{1}{N} \times \frac{d\left(\|\boldsymbol{x}\boldsymbol{w} - \boldsymbol{y}\|^2\right)}{d\boldsymbol{w}}$$

$$\frac{d\mathrm{L}_2(\mathbf{w})}{d\boldsymbol{w}} = \frac{d\left((\boldsymbol{x}\boldsymbol{w} - \boldsymbol{y})^T(\boldsymbol{x}\boldsymbol{w} - \boldsymbol{y})\right)}{d\boldsymbol{w}}$$

$$\frac{d\mathrm{L}_2(\mathbf{w})}{d\boldsymbol{w}} = \frac{d\left(\boldsymbol{w}^T\boldsymbol{x}^T\boldsymbol{x}\boldsymbol{w} - 2\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y}\right)}{d\boldsymbol{w}}$$

$$\frac{d\mathrm{L}_2(\mathbf{w})}{d\boldsymbol{w}} = 2\boldsymbol{x}^T\boldsymbol{x}\boldsymbol{w} - 2\boldsymbol{x}^T\boldsymbol{y}$$

# Differentiability of the L1 loss

$$\frac{d\mathrm{L}_1(\mathbf{w})}{d\boldsymbol{w}} = \frac{1}{N} \times \frac{d(|\boldsymbol{x}\boldsymbol{w} - \boldsymbol{y}|)}{d\boldsymbol{w}}$$

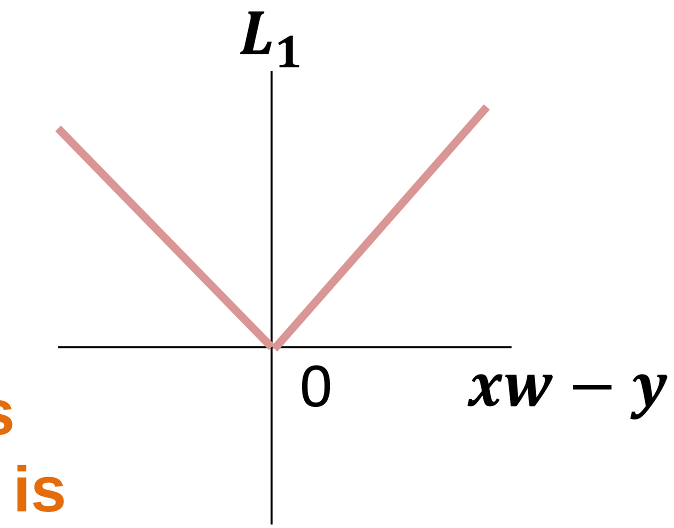
$$\frac{d\mathrm{L}_1(\mathbf{w})}{d\boldsymbol{w}} = \frac{d(\boldsymbol{x}\boldsymbol{w} - \boldsymbol{y})}{d\boldsymbol{w}}$$

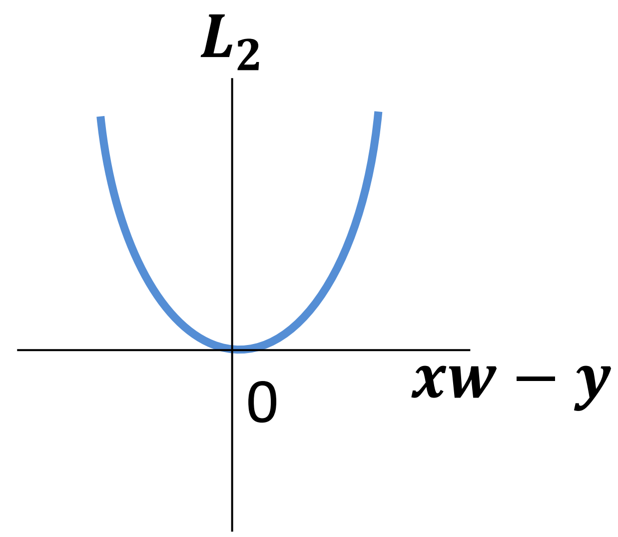
$$\frac{d\mathrm{L}_1(\mathbf{w})}{d\boldsymbol{w}} = \boldsymbol{x}$$

**L1 loss gradient is a constant!**

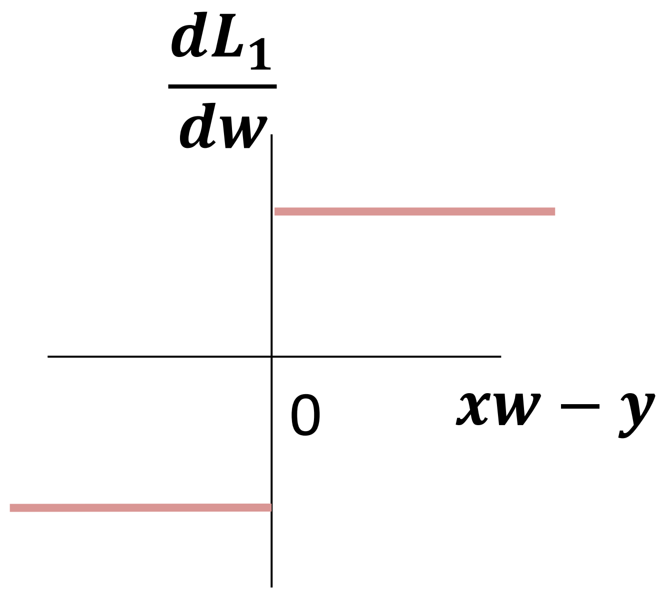
→ optimal $\boldsymbol{w}$ cannot be obtained analytically

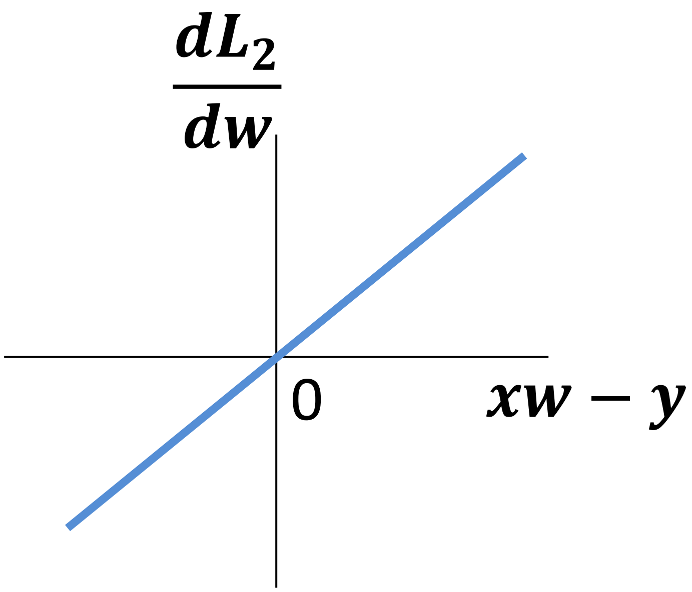
# L1 vs L2 loss and gradient




**L1 loss gradient is discontinuous at 0!**

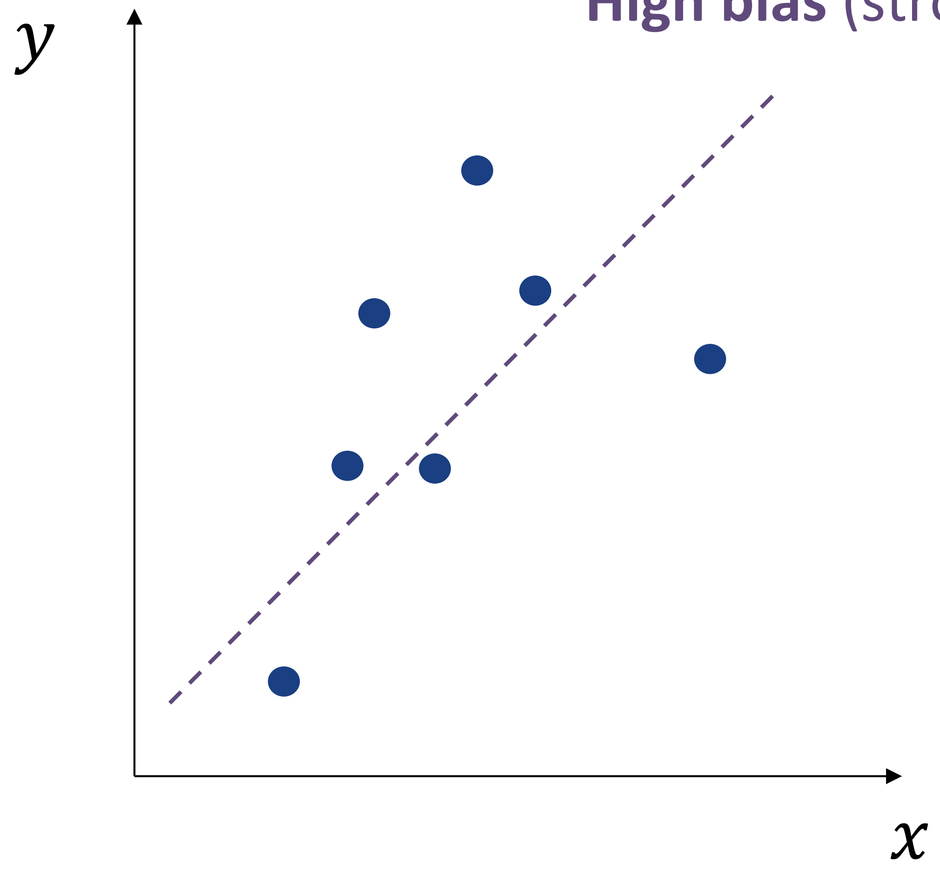


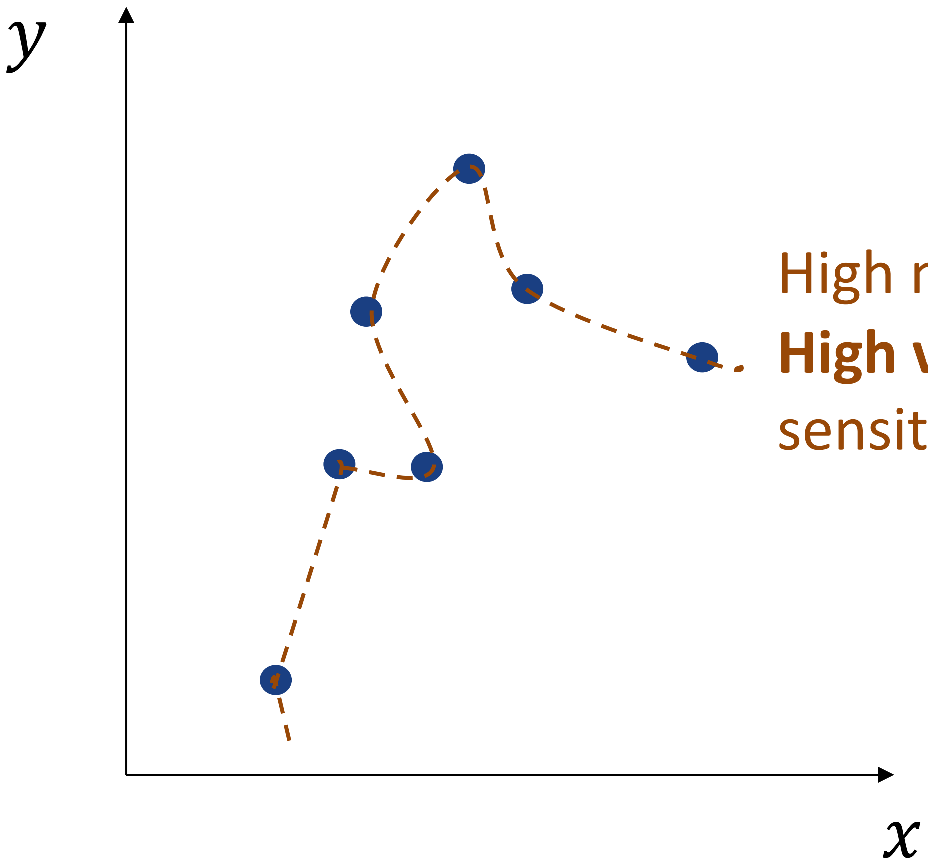
# Model generalizability errors: Bias

Low model complexity

**High bias** (strong assumption)



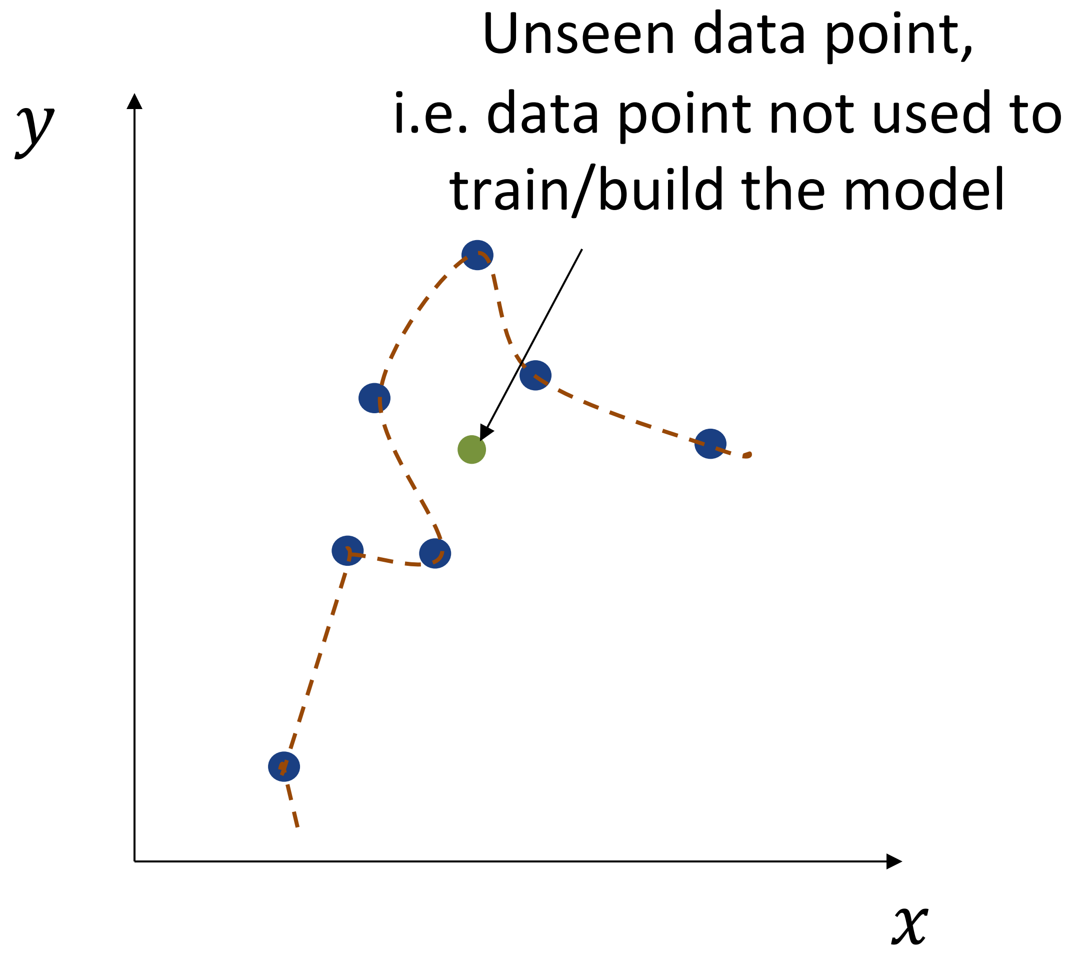
# Model generalizability errors: Variance




High model complexity

**High variance** (strong sensitivity to seen data)

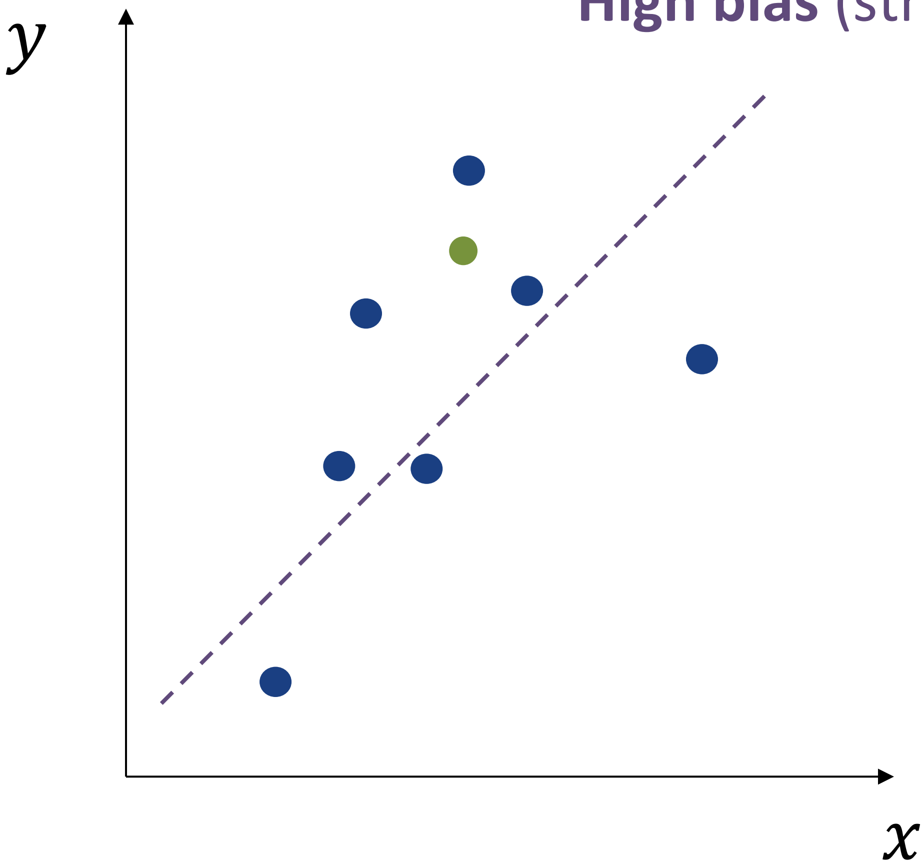
# Model generalizability errors: Variance (2)



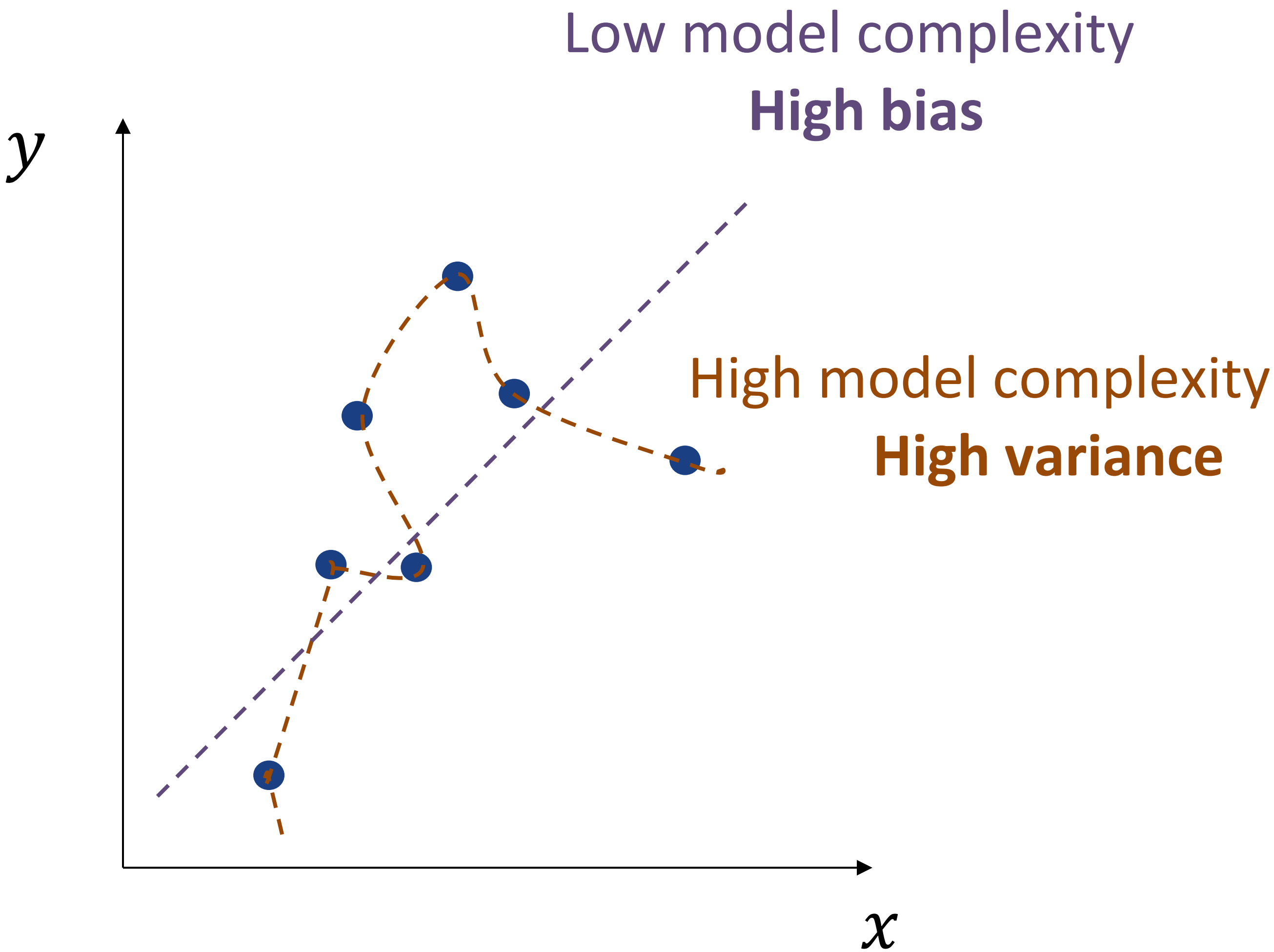
# Model generalizability errors: Bias (2)

Low model complexity

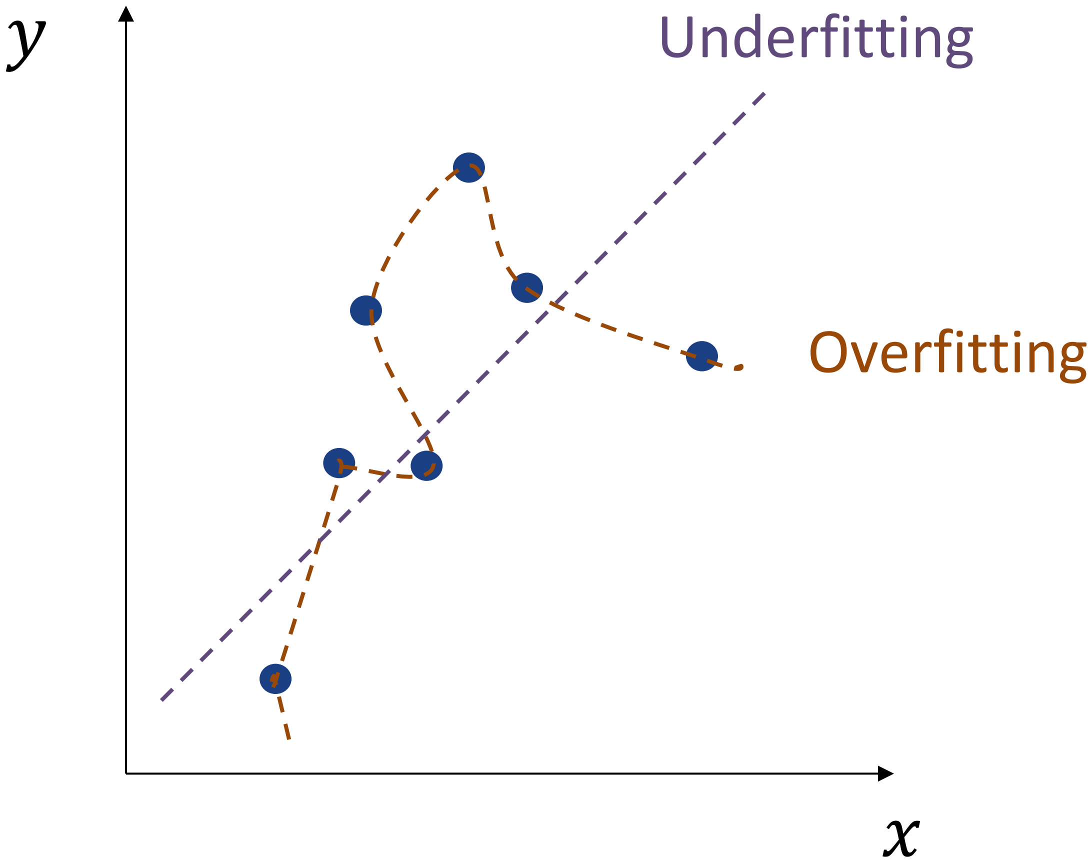
**High bias** (strong assumption)



# Generalizability: Bias-Variance Trade-off



# Reframing as 'Underfitting vs Overfitting'



# Regularization

- Overfitting (at the extreme) = memorization of the training data – the model cannot generalize to unseen data

- **Key ML challenge** – How to get a model to learn (i.e. not underfit) without just memorizing (overfitting)?

- A common strategy to address this is **regularization** i.e. adding an additional penalty term to the loss function

# L1 regularization

- Add a L1 penalty to the (L2) loss function, i.e.

$$L_{lasso}(\boldsymbol{w}) = \frac{1}{N}\|\boldsymbol{xw} - \boldsymbol{y}\|^2 + \alpha|w|$$

<u>Notations</u>

  - $\alpha|w|$ – regularization term
  - $w$ – weights (model parameter)
  - $\alpha$ – regularization strength (model <u>hyper</u>parameter)

- L1 regularization penalizes non-zero weights
  - this encourages zero weights
  - zero weights imply reduced model complexity

# Weights & Model complexity

- A linear model

$$f(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{w} + b = b + w_1 x_1 + w_2 x_2 + \cdots w_D x_D$$

- What happens when some weights are zero (while minimizing the prediction error), e.g.

$$f(\boldsymbol{x}) = b + 0x_1 + w_2 x_2 + 0x_3$$

  - uninformative features are ignored
  - the model is forced to find which features are informative and weigh them in accordingly

# L2 regularization

- Add a L2 penalty to the loss function, i.e.

$$L_{ridge}(\boldsymbol{w}) = \frac{1}{N}||\boldsymbol{xw} - \boldsymbol{y}||^2 + \alpha||\boldsymbol{w}||^2$$

<u>Notations</u>

  - $\alpha||\boldsymbol{w}||^2$ – regularization term
  - $w$ – weights (model parameter)
  - $\alpha$ – regularization strength (model <u>hyper</u>parameter)

- L2 regularization penalizes large weights
  - L2 encourages <u>very small</u> weights
  - much smaller weights reduce model complexity

# Summary: Regression basics

1. The most basic elements of machine learning are **data** (**features** & **labels**), **model** (defined by **weights**).
2. The most basic ML algorithm is **linear regression**, a **linear model** that learns real valued labels.
3. Training a ML model involves optimizing the model weights based on a **loss function**.
4. Achieving the goal of generalizability to unseen data is trading off between **bias** (**underfitting**) & **variance** (**overfitting**).
5. Overfitting can be addressed with **regularization**.

# A glossary to help

https://developers.google.com/machine-learning/glossary

## L

...

### $L_1$ loss

A **loss function** that calculates the absolute value of the difference between actual **label** values and the values that a **model** predicts. For example, here's the calculation of $L_1$ loss for a **batch** of five **examples**:

### $L_1$ regularization

A type of **regularization** that penalizes **weights** in proportion to the sum of the absolute value of the weights. $L_1$ regularization helps drive the weights of irrelevant or barely relevant features to *exactly 0*. A **feature** with a weight of 0 is effectively removed from the model.

...

### label

In **supervised machine learning**, the "answer" or "result" portion of an **example**.

Each **labeled example** consists of one or more **features** and a label. For example, in a spam detection dataset, the label would probably be either "spam" or "not spam." In a rainfall dataset, the label might be the amount of rain that fell during a certain period.

# Content today: Introduction

- ❑ Module information
- ❑ Machine learning in our world
- ❑ Regression basics
- ❑ **Classification basics**

# Supervised learning: Classification

- Consider that there exists training data instances $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N}$, $\boldsymbol{x}_n \in \mathbb{R}^{D_x}$, $\boldsymbol{y}_n \in \mathbb{Z}^{D_y}$

- The goal is to find a model $f(.)$ that takes in as input $\boldsymbol{x}_n$ and outputs $\hat{\boldsymbol{y}}_n$ such that:

  - $f(\boldsymbol{x}_n) = \hat{\boldsymbol{y}}_n = \boldsymbol{y}_n$; and

  - $f(.)$ is generalizable to unseen data instances

    i.e. $f(\boldsymbol{x}_m) = \hat{\boldsymbol{y}}_m = \boldsymbol{y}_m$ where $\boldsymbol{x}_m \notin \{\boldsymbol{x}_n\}_{n=1}^{N}$

# Basic linear model: Classification

Given $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N}$, $\boldsymbol{x}_n \in \mathbb{R}^{D_x}$, $\boldsymbol{y}_n \in \mathbb{Z}^{D_y}$

$$f(\boldsymbol{x}) = \sigma(\boldsymbol{x}\boldsymbol{w} + b) = \hat{\boldsymbol{y}}$$

Notations

- $f(\cdot)$ – basic linear model
- $\boldsymbol{x}$ – features (or model input)
- $\hat{\boldsymbol{y}}$ – predicted labels/targets (or model output)
- $\boldsymbol{w}$, $b$ – weights, bias (or model parameters)
- $\sigma(\cdot)$ – activation function (for discretizing real values)

# Toy data with categorical labels

$\{\boldsymbol{x}_n\}_{n=1}^{6}, D_x = (h, w, c), c = 3$ for R,G,B

$\{y_n\}_{n=1}^{6}, D_y = 1$

cat

Source: Muhammad Mahdi Karim
https://commons.wikimedia.org/wiki/File:Domestic_cat_felis_catus.jpg

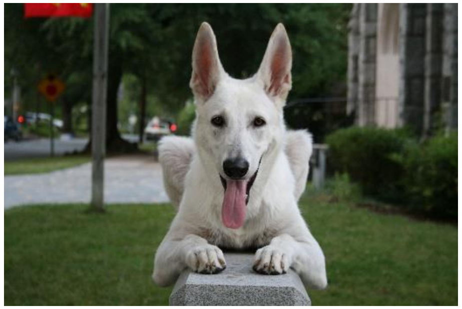
Source: Dimitri Torterat
https://commons.wikimedia.org/wiki/File:Domestic_shorthaired_cat_face.jpg

Source: Peter Forster
https://commons.wikimedia.org/wiki/File:Cat_Briciola_with_pretty_and_different_colour_of_eyes.jpg

dog

Source: Eugene0126jp
https://en.wikipedia.org/wiki/File:Dog.in.sleep.jpg

Source: Jina Lee
https://commons.wikimedia.org/wiki/File:Pug_dog_nose_face_detail.JPG



Source: IldarSagdejev
https://en.wikipedia.org/wiki/File:2008-06-26_White_German_Shepherd_Dog_Posing_3.jpg

# Categorical labels as numerical

$$\{\boldsymbol{x}_n\}_{n=1}^{6}, D_x = (h, w, c), c = 3 \text{ for R,G,B}$$

$$\{y_n\}_{n=1}^{6}, D_y = 1$$

+1

Source: Muhammad Mahdi Karim
https://commons.wikimedia.org/wiki/File:Domestic_cat_felis_catus.jpg

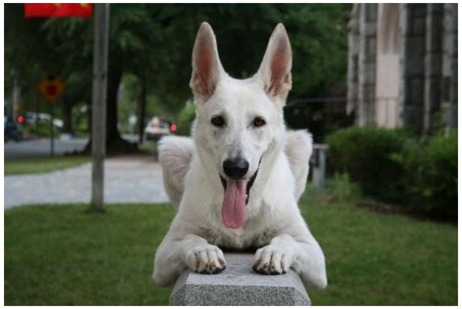
Source: Dimitri Torterat
https://commons.wikimedia.org/wiki/File:Domestic_shorthaired_cat_face.jpg

Source: Peter Forster
https://commons.wikimedia.org/wiki/File:Cat_Briciola_with_pretty_and_different_colour_of_eyes.jpg

-1

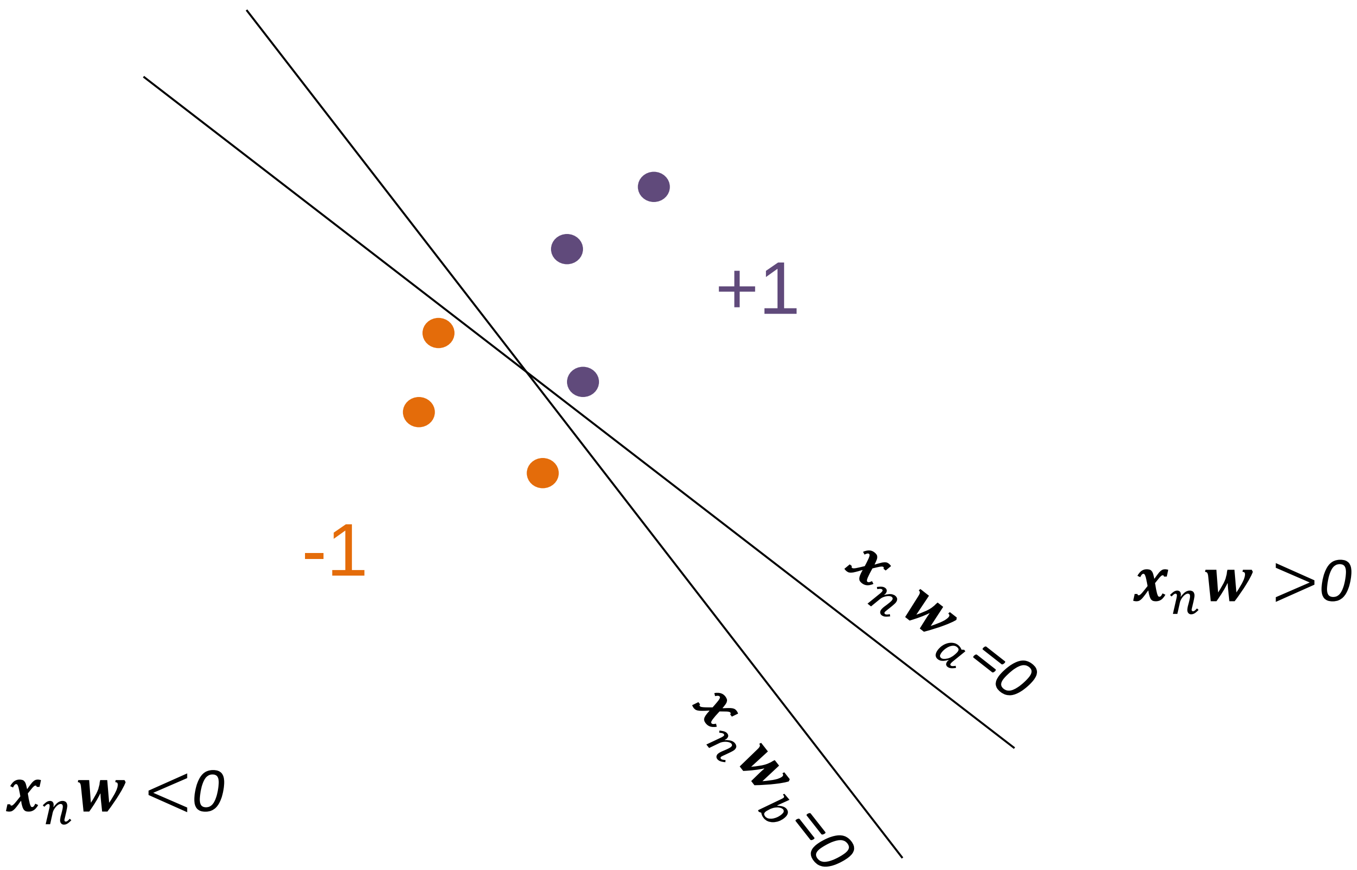
Source: Eugene0126jp
https://en.wikipedia.org/wiki/File:Dog.in.sleep.jpg

Source: Jina Lee
https://commons.wikimedia.org/wiki/File:Pug_dog_nose_face_detail.JPG

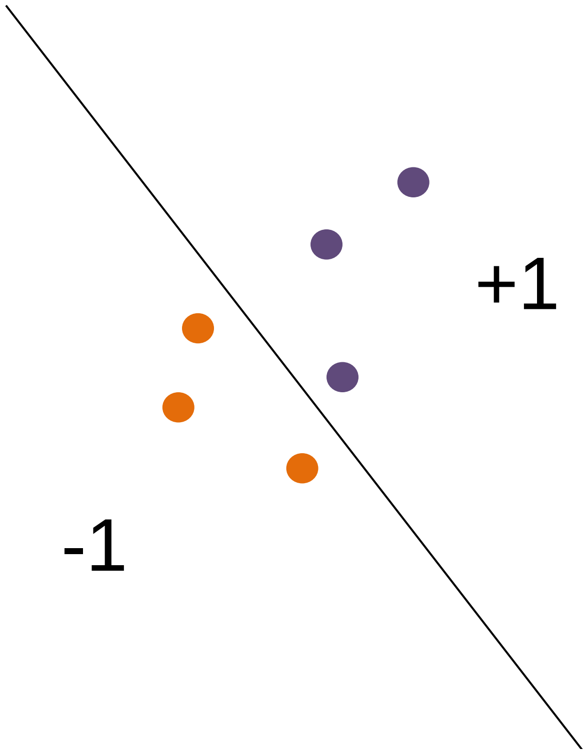


Source: IldarSagdejev
https://en.wikipedia.org/wiki/File:2008-06-26_White_German_Shepherd_Dog_Posing_3.jpg

# Linear model visualization: toy example



# Classification




$\boldsymbol{y}_n \in \mathbb{Z}^{D_y}$ (i.e. categorical labels) implies that the supervised learning is a **classification** task

# Finding optimal weights

- A simple classification loss function

$$L_0(\boldsymbol{w}) = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{I}.\, sign(\boldsymbol{x}_n\boldsymbol{w} + b) \neq y_n$$

$L_0(\cdot)$ is 0-1 loss function (aka sign loss)

- Hinge loss

$$L_{hinge}(\boldsymbol{w}) = \frac{1}{N}\sum_{n=1}^{N} max\left(0, \quad -\boldsymbol{y}_n(\boldsymbol{x}_n\boldsymbol{w} + b)\right)$$

# 0-1 & Hinge loss functions

**0-1 loss**

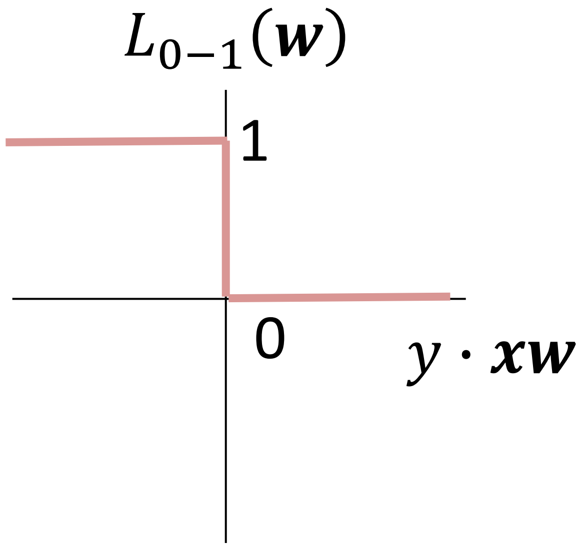
$\hat{y}$

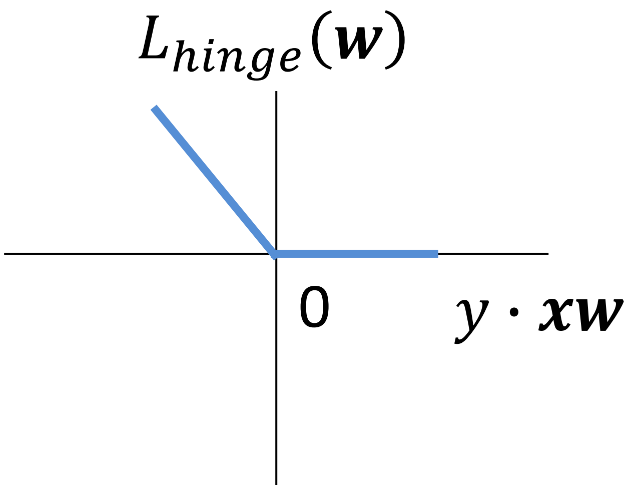
| | | -1 | +1 |
|---|---|---|---|
| $y$ | -1 | 0 | 1 |
| | +1 | 1 | 0 |

**Hinge loss**

$\hat{y}$

| | | -1 | +1 |
|---|---|---|---|
| $y$ | -1 | 0 | $y \cdot \boldsymbol{x}\boldsymbol{w}$ |
| | +1 | $y \cdot \boldsymbol{x}\boldsymbol{w}$ | 0 |

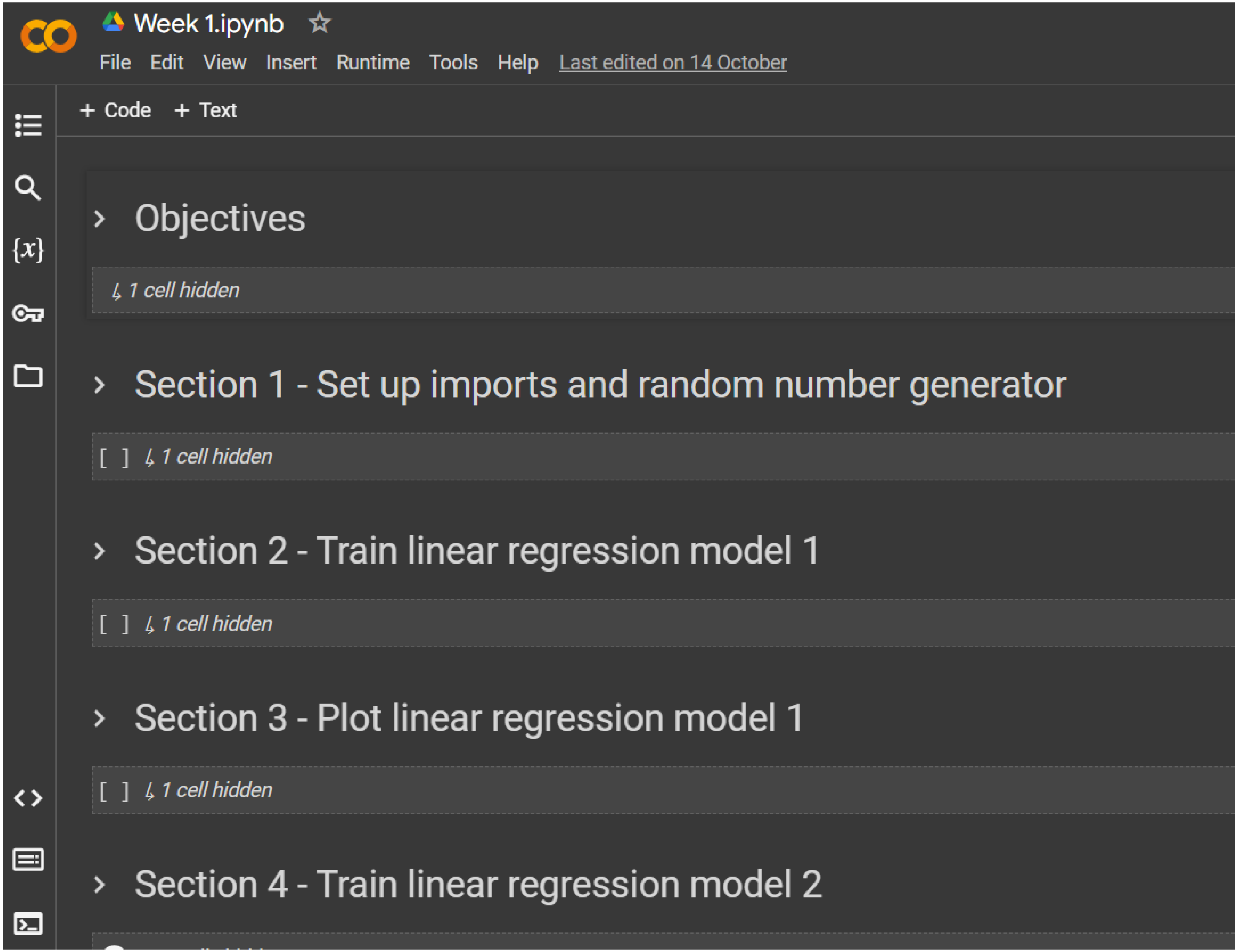


# Summary: Classification

1. Classification is to categorical labels as regression is to real valued labels.
2. An activation function allows the basic linear model to be used for classification.
3. Classification loss functions are typically different from regression loss functions but differentiability is an important for both.

# Week 1 lab: Read the objectives!



# Week 1 ungraded quiz

## Quiz: The basic linear model

Questions

- Question 1
- Question 2
- Question 3
- Question 4
- Question 5
- Question 6

## Quiz instructions

**Question 1** — **10 pts**

In a basic machine learning task, the aim is to

- ○ minimise the loss function
- ○ update model parameters
- ○ fit a line to the data
- ○ predict a numerical target/label given a feature instance

Next ▸