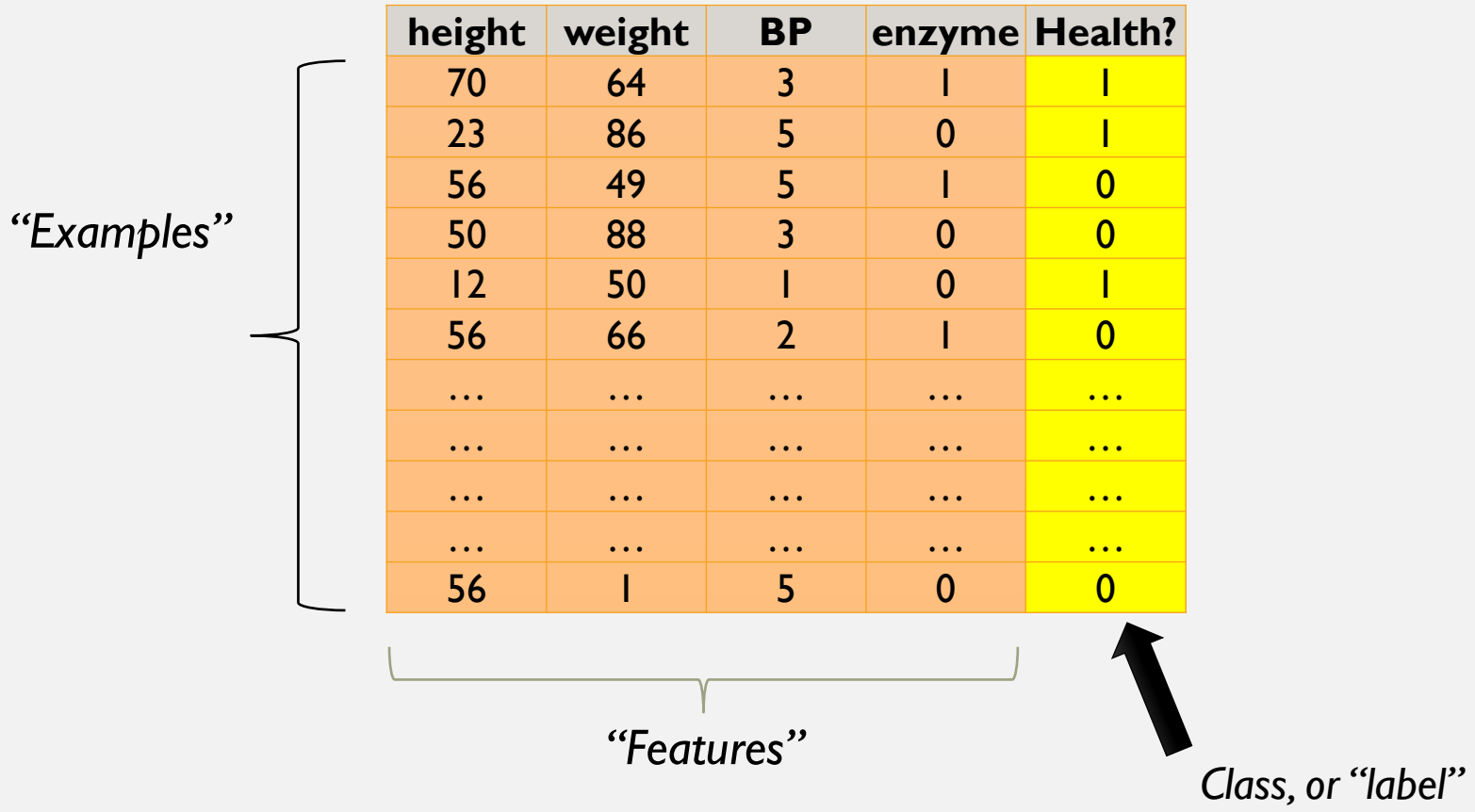
# MATHEMATICAL TOPICS IN MACHINE LEARNING

## (LECTURE 1 – RECAP OF COMP24112 TOPICS)

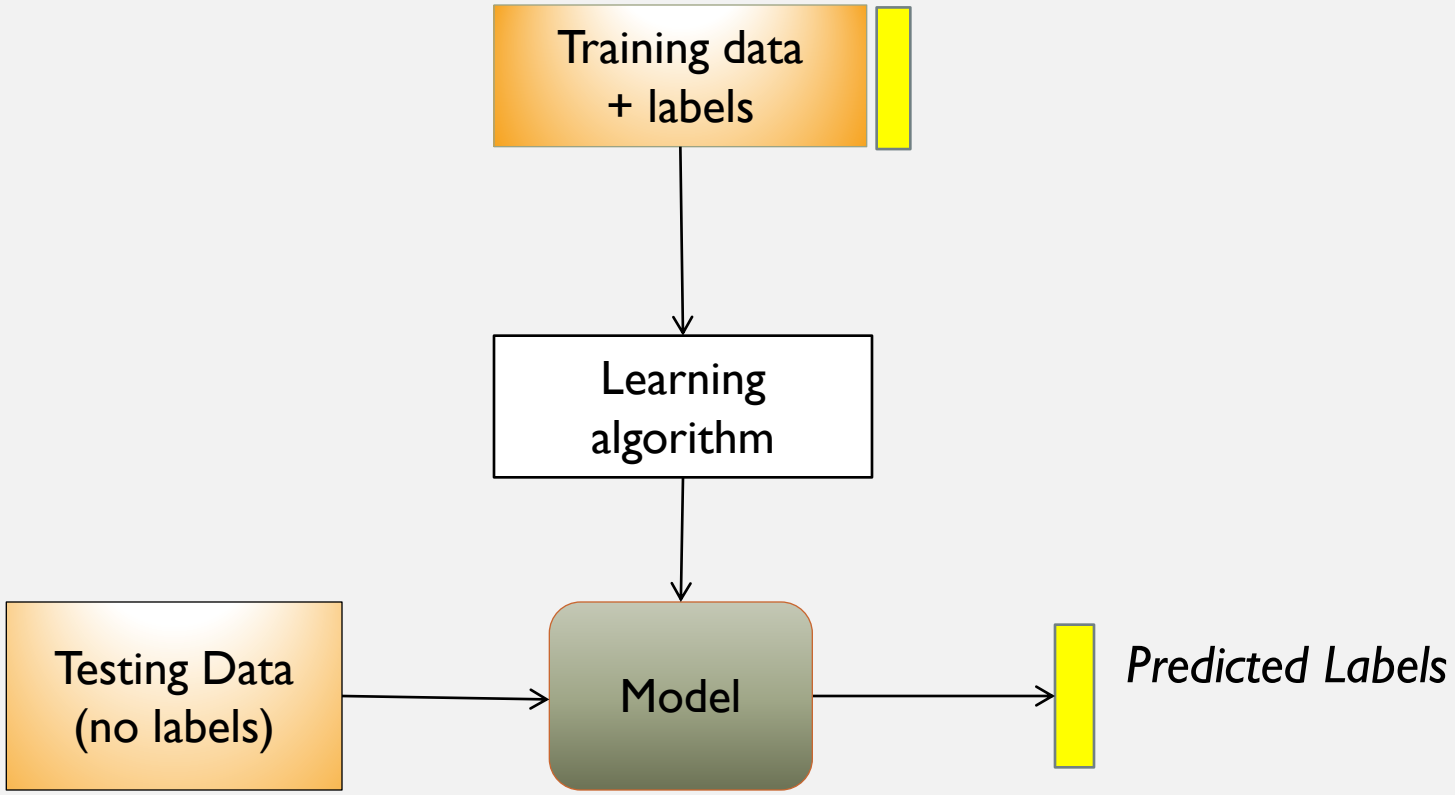
Professor Gavin Brown

# SUPERVISED LEARNING

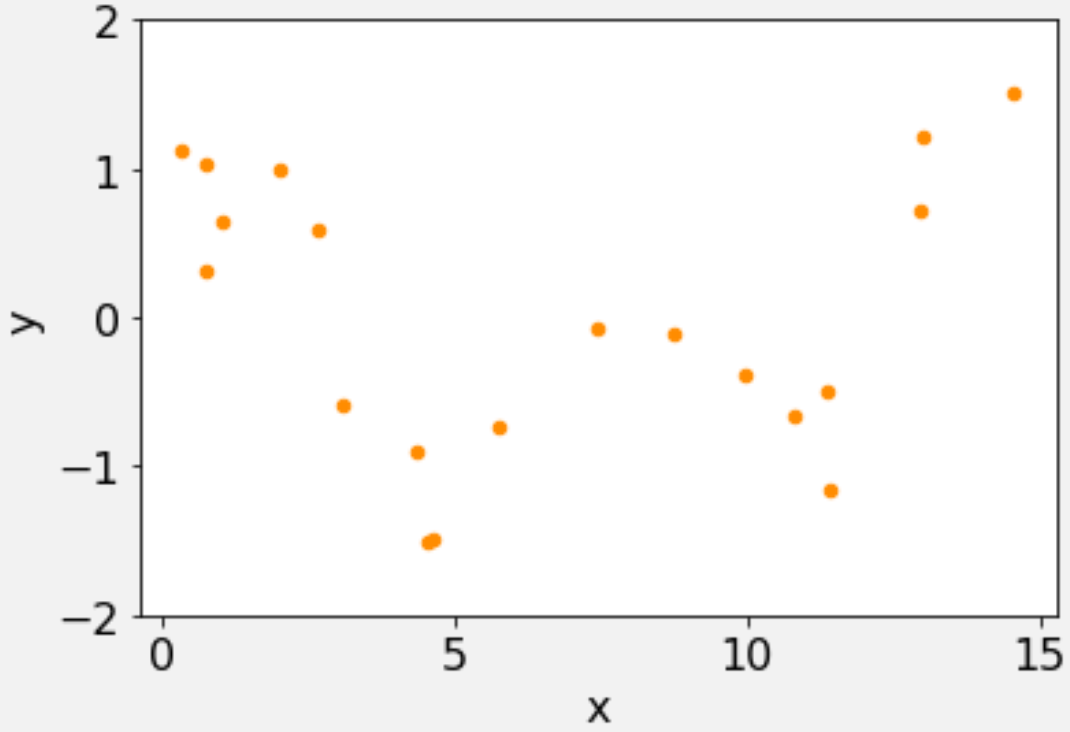



| height | weight | BP | enzyme | Health? |
|---|---|---|---|---|
| 70 | 64 | 3 | 1 | 1 |
| 23 | 86 | 5 | 0 | 1 |
| 56 | 49 | 5 | 1 | 0 |
| 50 | 88 | 3 | 0 | 0 |
| 12 | 50 | 1 | 0 | 1 |
| 56 | 66 | 2 | 1 | 0 |
| … | … | … | … | … |
| … | … | … | … | … |
| … | … | … | … | … |
| … | … | … | … | … |
| 56 | 1 | 5 | 0 | 0 |

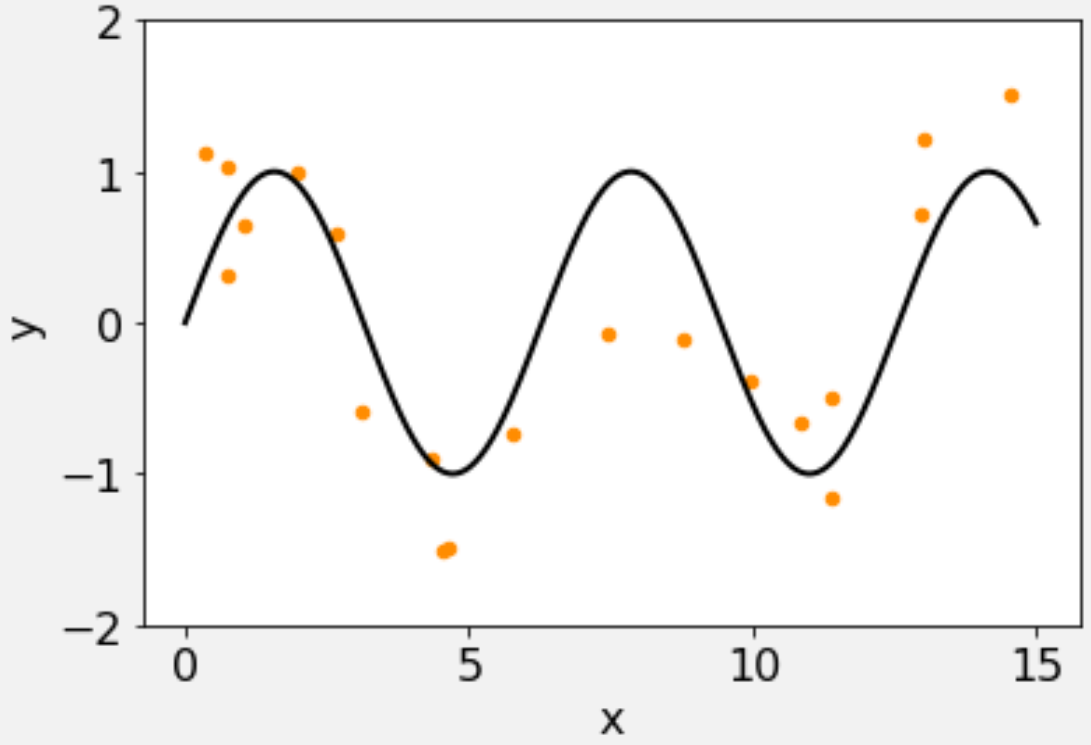
# SUPERVISED LEARNING



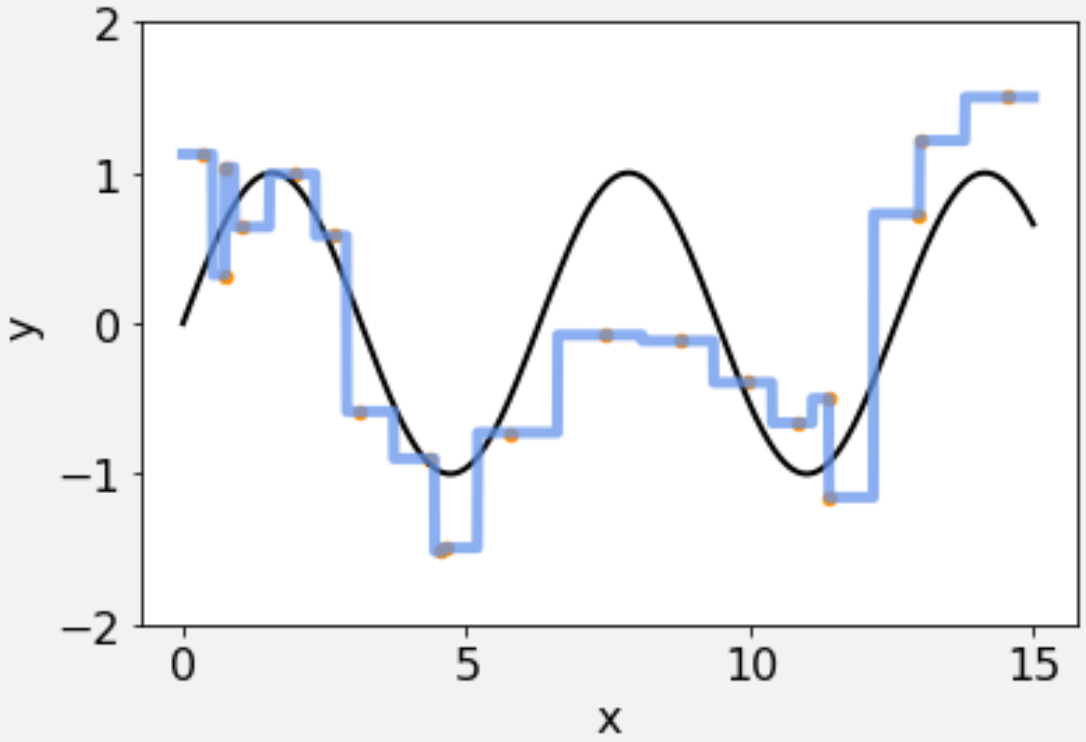
# A SIMPLE EXAMPLE – PREDICT **Y** FROM **X**



# THE UNDERLYING FUNCTION IS Y=SIN(X)

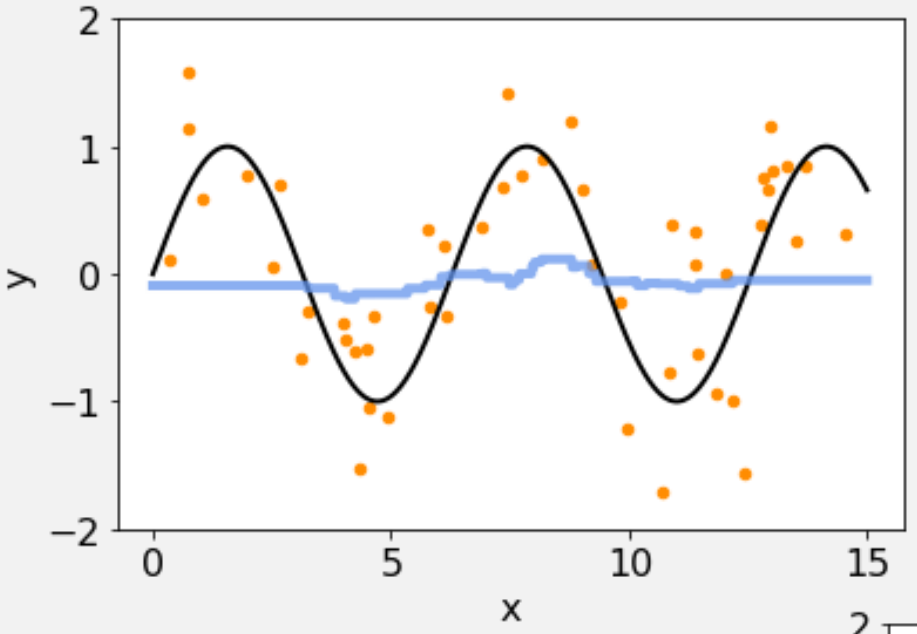


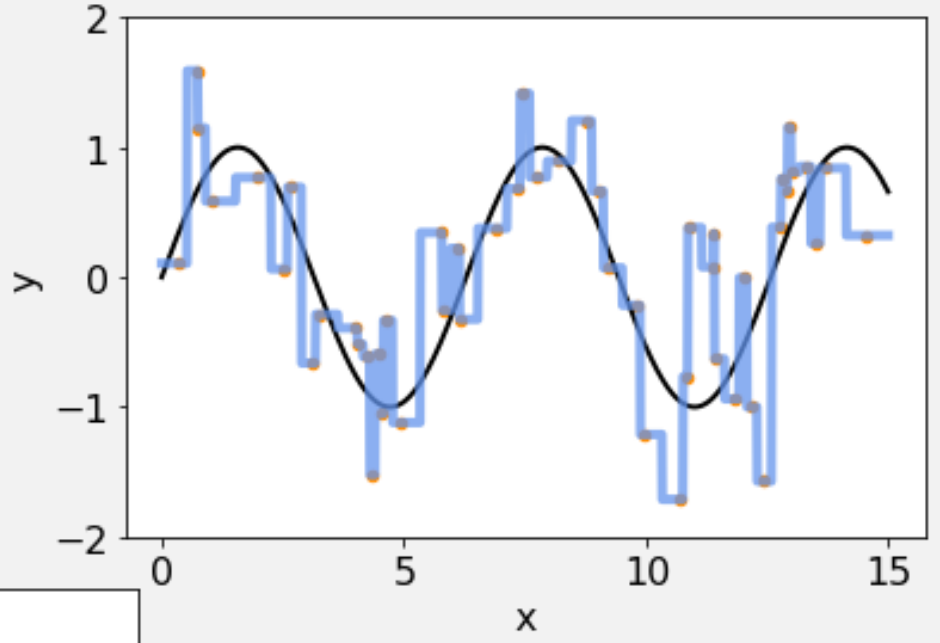
# K NEAREST NEIGHBOR REGRESSION (K=1)

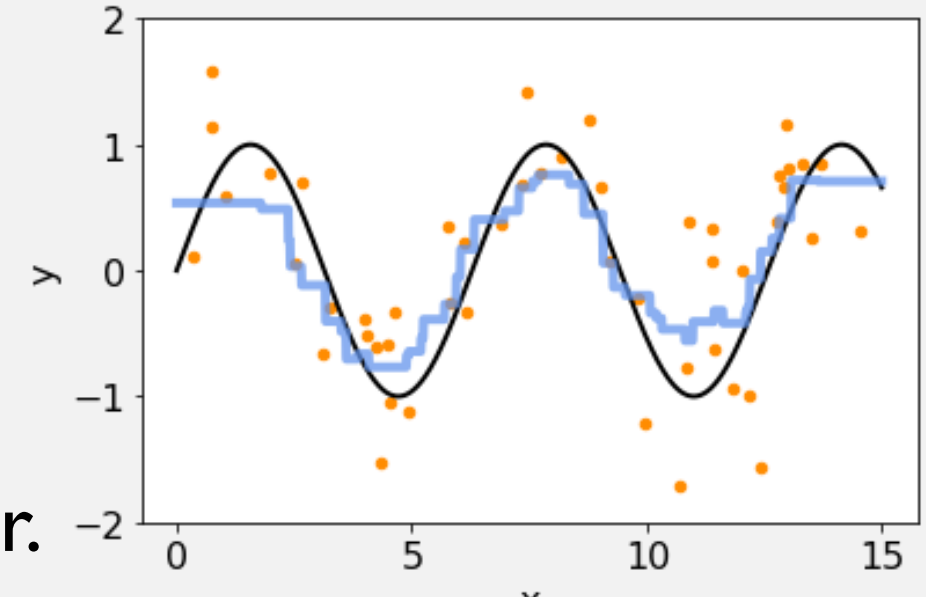



**Note** : k-NN regression was in COMP24112

# K NEAREST NEIGHBOR REGRESSION … K={1,8,20}

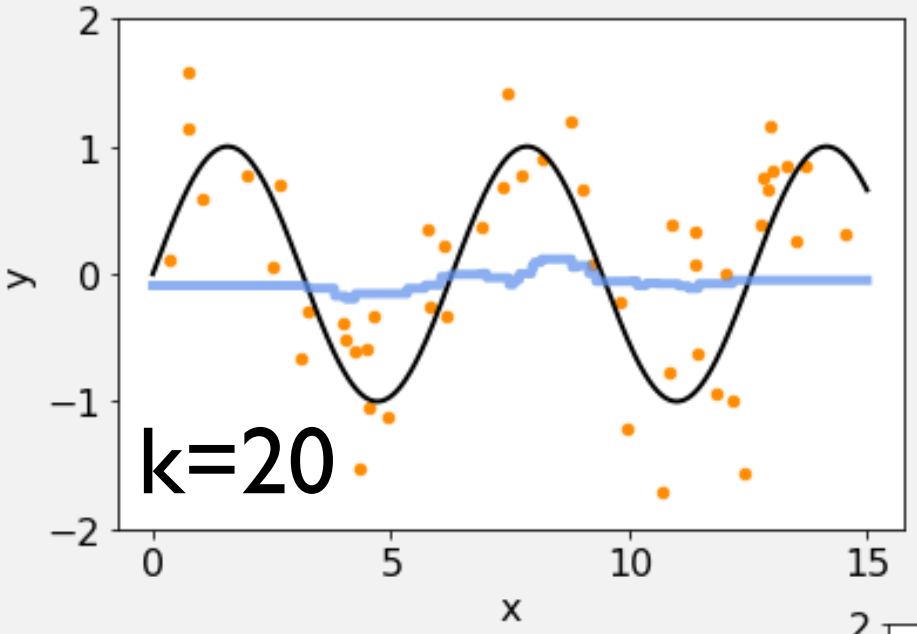


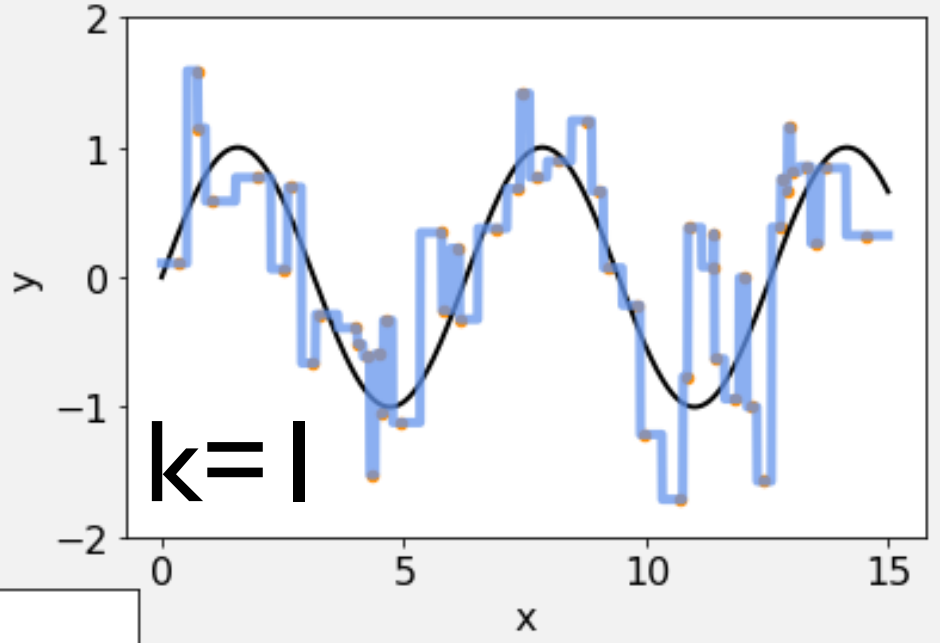



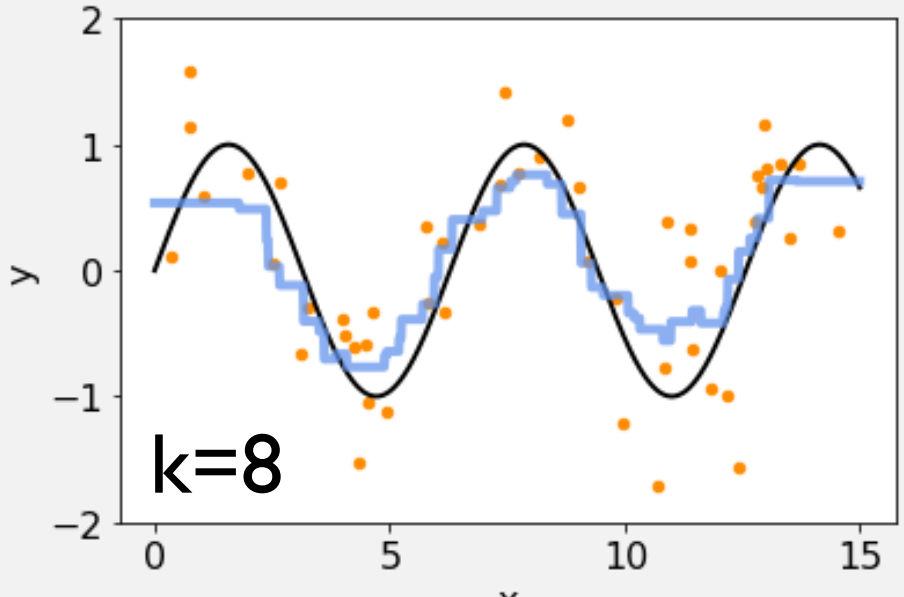




Which one is which?
Talk to your neighbor.

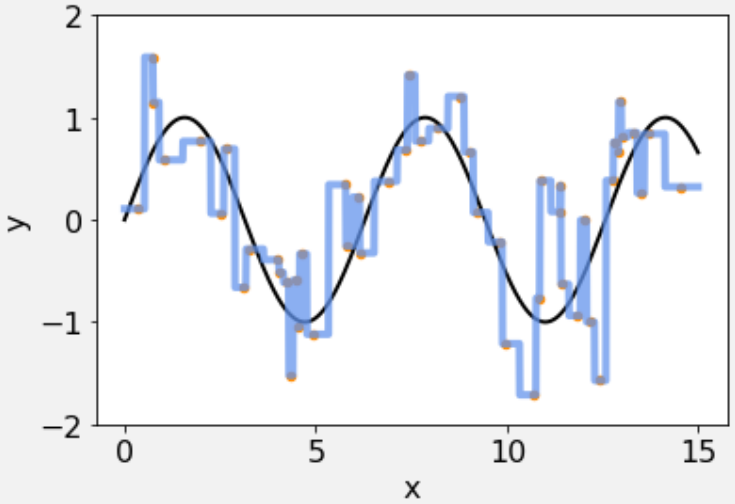
# K NEAREST NEIGHBOR REGRESSION … K={1,8,20}



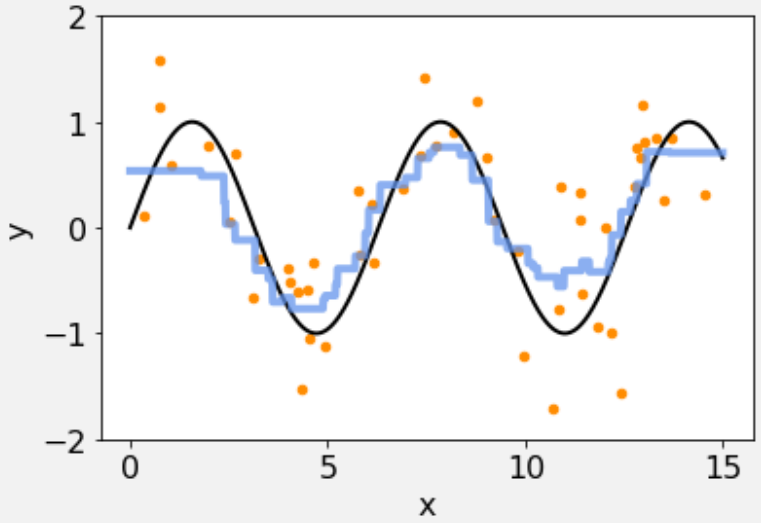



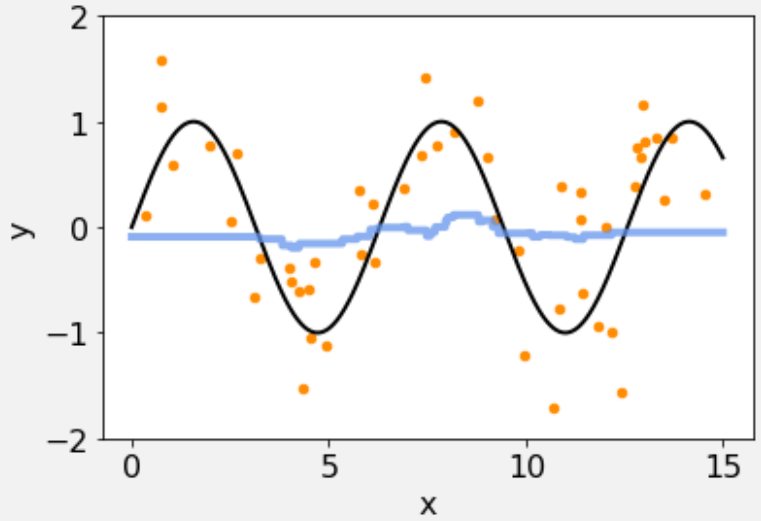
# K NEAREST NEIGHBOR REGRESSION … K={1,8,20}




k=1




k=8

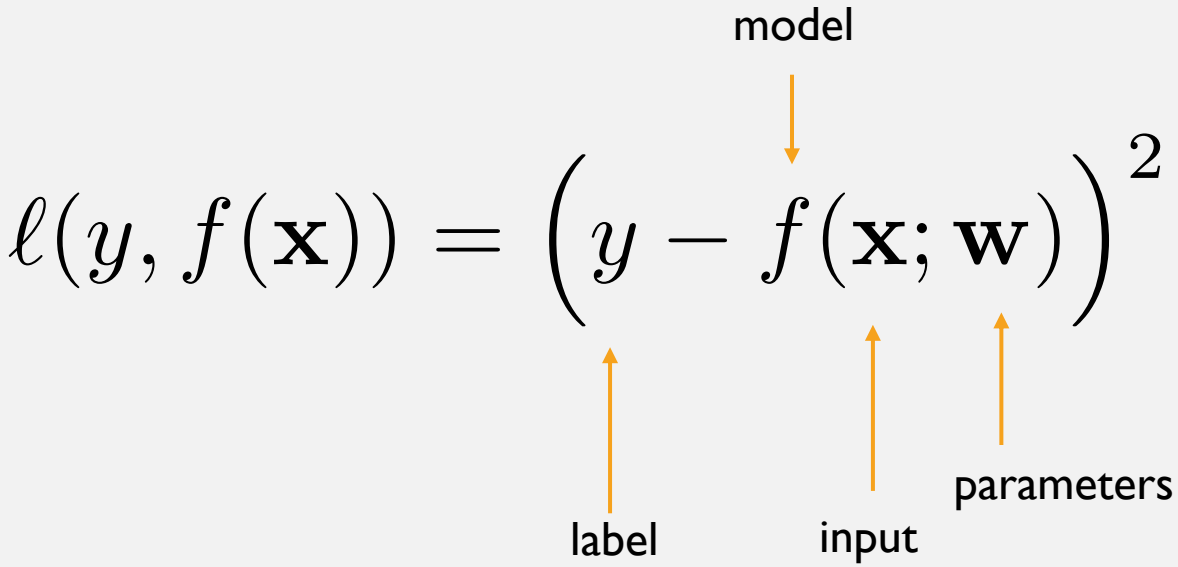



k=20

Complex function.
"**Overfitted**"

Simple function.
"**Underfitted**"

# LOSS FUNCTIONS : SQUARED LOSS




Loss for a single example (**x**,y) using a model ***f***

# LOSS FUNCTIONS : SQUARED LOSS

$$\ell_{train}(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\Big(y_i - f(\mathbf{x}_i;\mathbf{w})\Big)^2.$$

Training error for a model ***f*** on a training set $D_{train} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$

# LOSS FUNCTIONS : SQUARED LOSS

$$\mathbf{w}^* := argmin_{\mathbf{w}} \left[ \frac{1}{n} \sum_{i=1}^{n} \left( y_i - f(\mathbf{x}_i; \mathbf{w}) \right)^2 \right].$$

Our objective is to minimize training error by finding parameters **w***.

# LOSS FUNCTIONS : CROSS-ENTROPY

Squared loss is suitable for **regression,** which assumes $y \in \mathbb{R}$

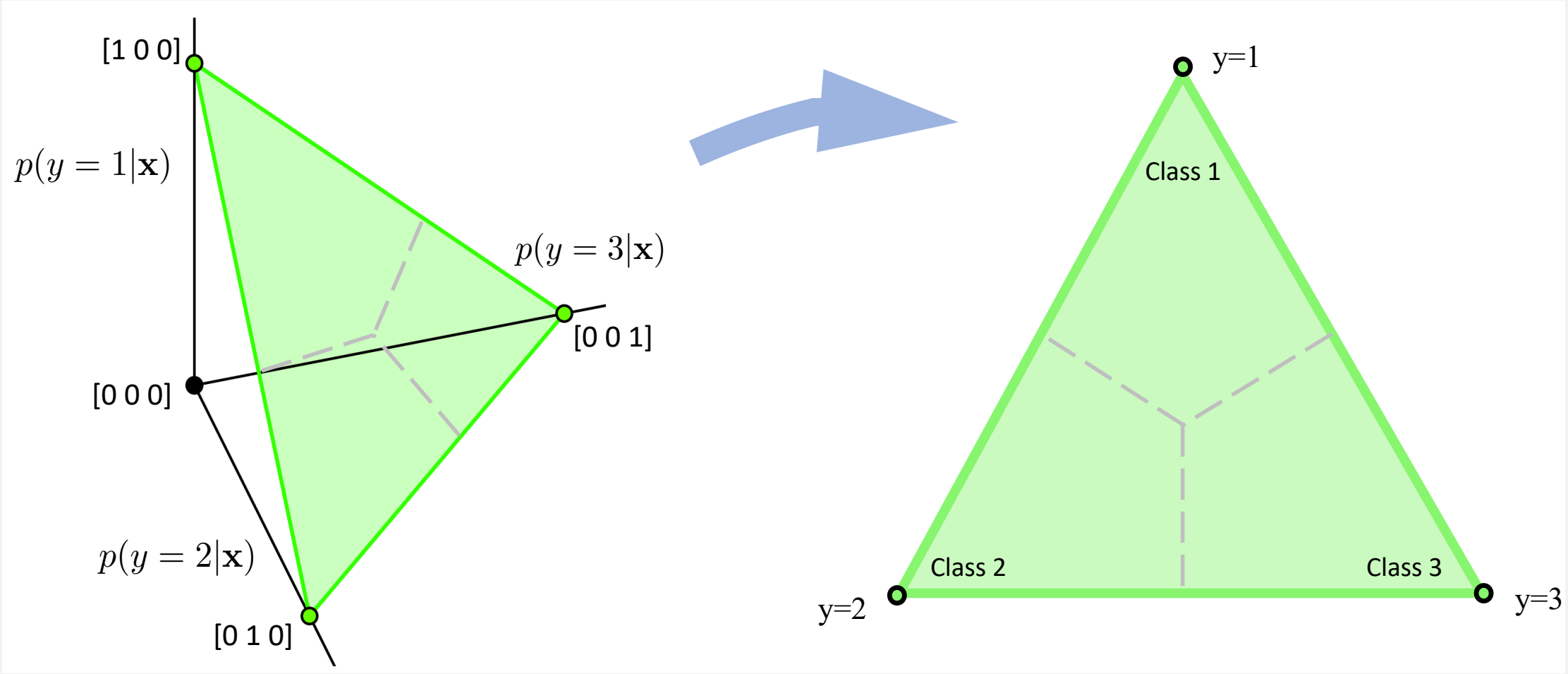
Many problems are instead one of **classification,** $y \in \{A, B, C, \ldots, \}$

*i.e. a class label*

We can tackle this by assuming ***f*** is a probability estimator, so $y \in \mathbb{R}^k_\Delta$

*i.e. a probability distribution over k classes*

# CLASSIFICATION IS A PREDICTION IN THE SIMPLEX



# LOSS FUNCTIONS : CROSS-ENTROPY

For binary classification, we have the binary cross-entropy loss…

$$\ell(y, f(\boldsymbol{x})) \;:=\; -\Big[y \ln f(\boldsymbol{x}) + (1-y)\ln(1-f(\boldsymbol{x}))\Big]$$

To understand this, remember that for any single training example (**x**,y), we have $y \in \{0, 1\}$

For multi-class classification, we have the general case:

$$\ell(\boldsymbol{y}, f(\boldsymbol{x})) \;:=\; -\Big[\boldsymbol{y} \cdot \ln f(\boldsymbol{x})\Big]$$

# LOSS FUNCTIONS : CROSS-ENTROPY

**Concept Check...**
I give you an example where the true label is $y = 1$.
ho My model predicts the probability of $y = 1$ as $f(\boldsymbol{x}) = 0.95$.

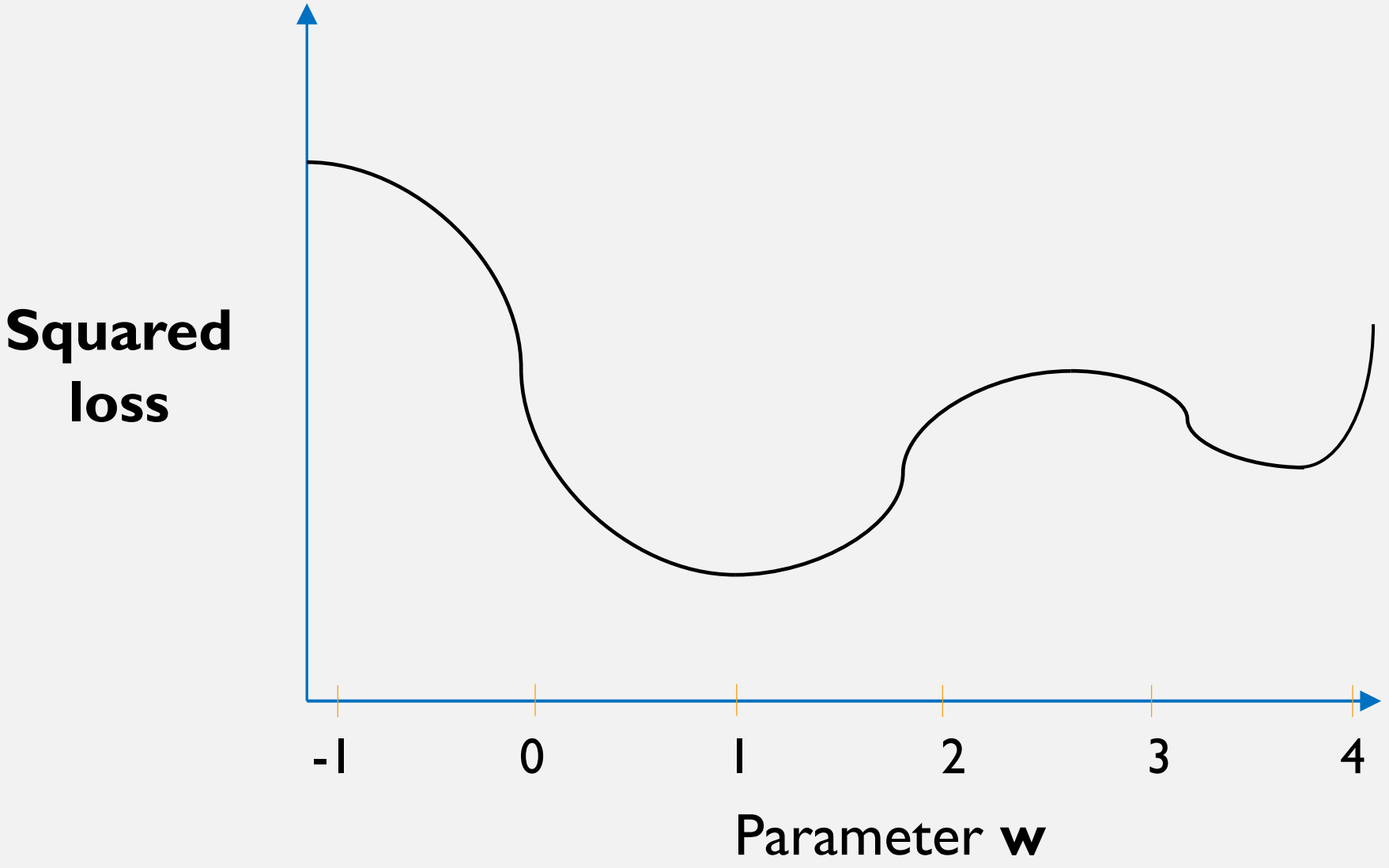
**Q1.** Calculate the cross-entropy loss. Remember to use the **natural** logarithm.

Now, I give you an example $\boldsymbol{x}$, where $\boldsymbol{y} = [1, 0, 0]$.
My model predicts $f(\boldsymbol{x}) = [0.3, 0.6, 0.1]$.

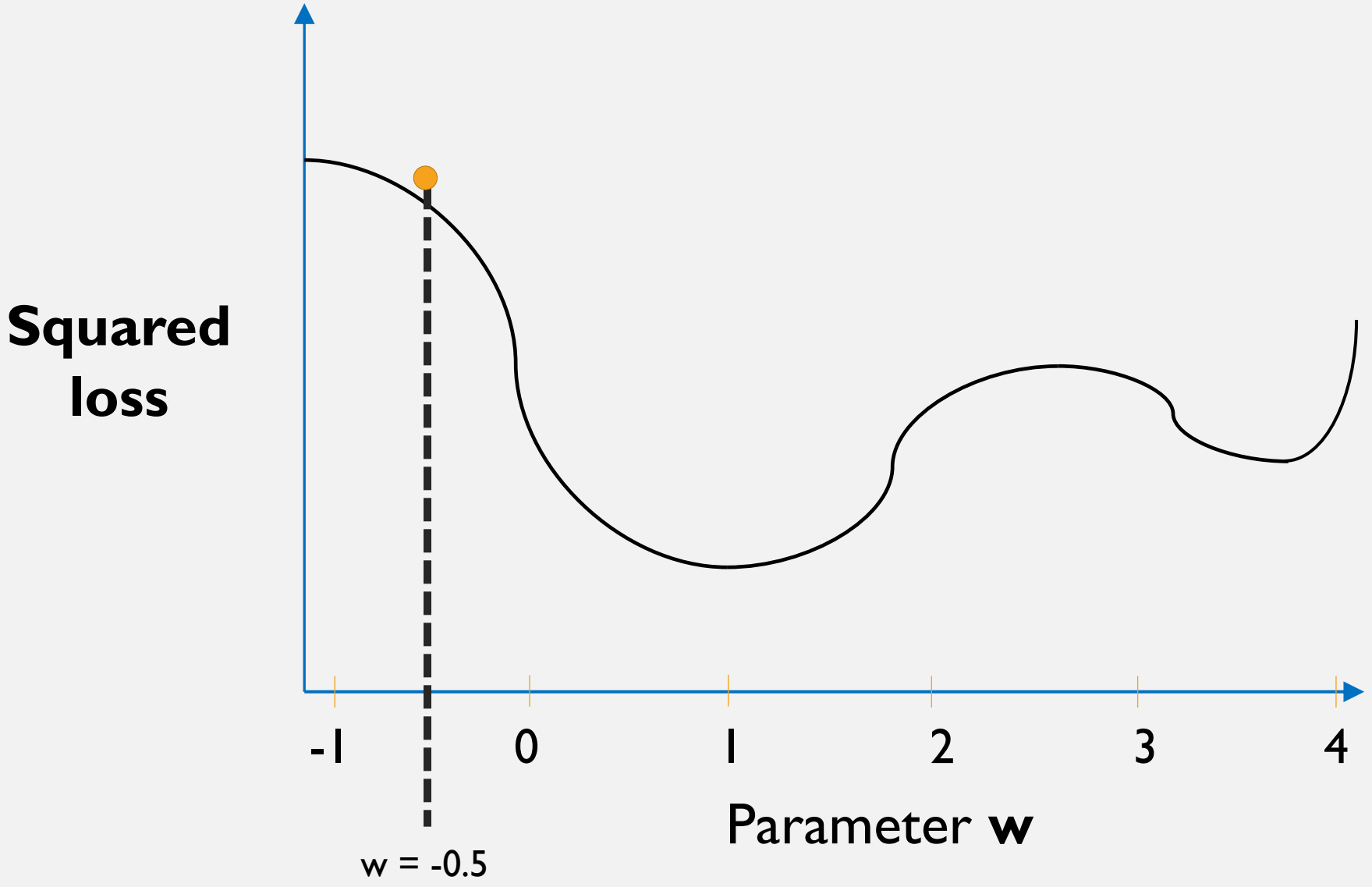
**Q2.** Calculate the cross-entropy loss again.

**Q3.** Roughly where in the simplex does this model sit?

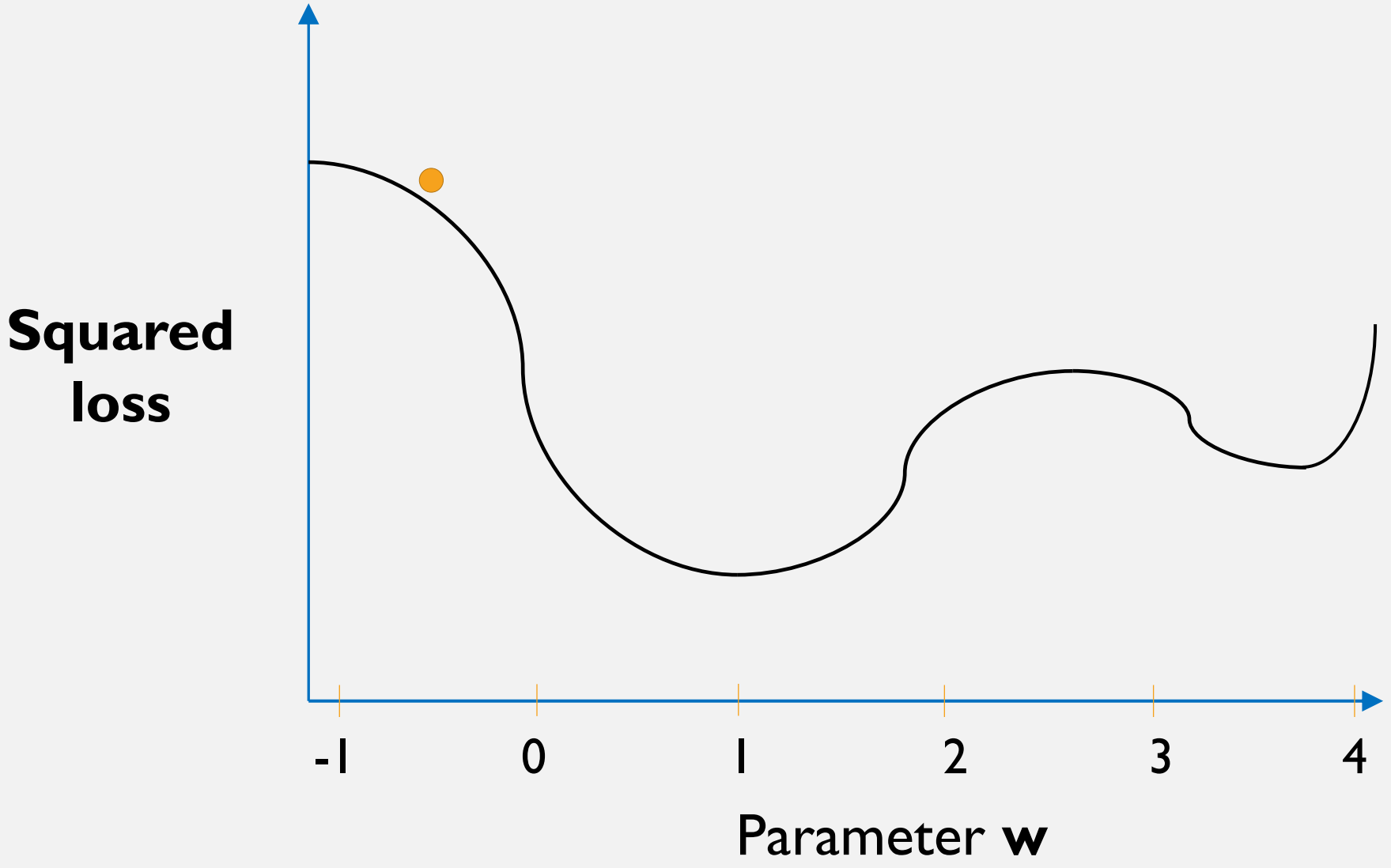
# LOSS FUNCTIONS : GRADIENT DESCENT



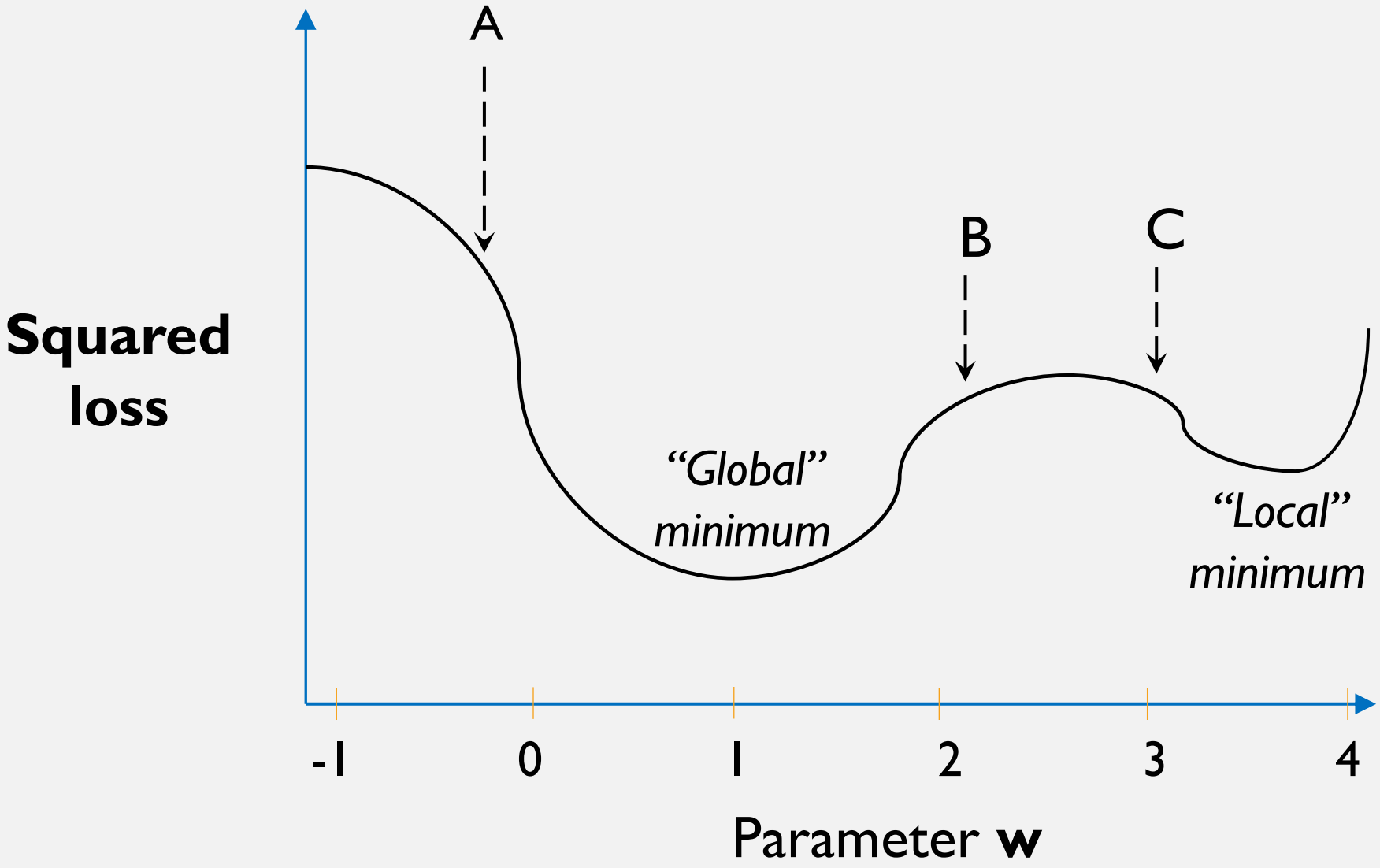
# LOSS FUNCTIONS : GRADIENT DESCENT



# LOSS FUNCTIONS : GRADIENT DESCENT



# LOSS FUNCTIONS : GRADIENT DESCENT



# LOSS FUNCTIONS : GRADIENT DESCENT

Loss

$$\ell(y, f) = \frac{1}{2}(y - f(\mathbf{x}; \mathbf{w}))^2$$

Gradient of loss

$$\frac{\partial \ell(y, f)}{\partial \mathbf{w}_j} = \Big(y - f(\mathbf{x}; \mathbf{w})\Big) \frac{\partial f(\mathbf{x}; \mathbf{w})}{\partial \mathbf{w}_j}$$

# LOSS FUNCTIONS : GRADIENT DESCENT

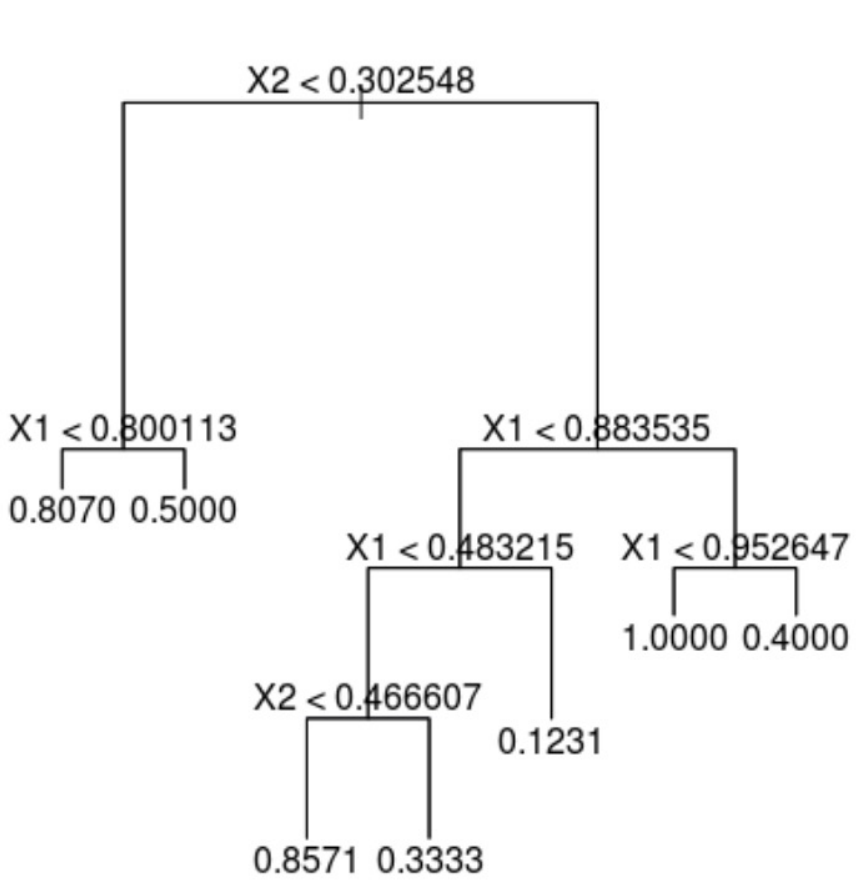
**Algorithm 1** Gradient Descent for weight vector $\mathbf{w}$ on a differentiable loss $\ell(y, f(\boldsymbol{x}; \mathbf{w}))$

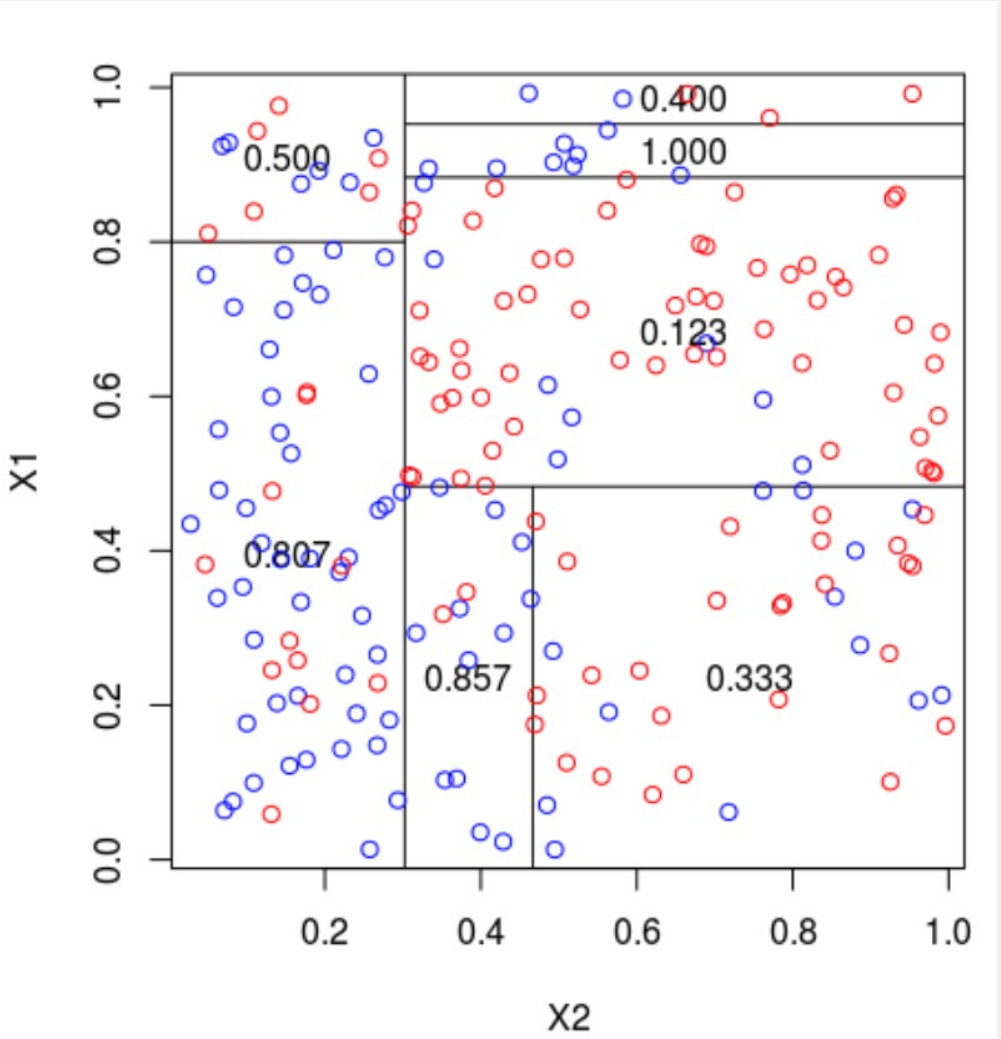
1: **Choose :** A value $T > 0$ for the number of steps to take.
2: **Choose :** A constant $\eta$, for the size of each step.
3: **Input:** An initial vector $\mathbf{w}$ of length $J$.
4: **for** $t = 1, \ldots, T$ **do**
5: $\quad \mathbf{w} \leftarrow \mathbf{w} - \eta \cdot \nabla \ell(y, f(\boldsymbol{x}; \mathbf{w}))$
6: **end for**
7: **Output :** $\mathbf{w}$

Notice that here we used the 'nabla' notation. $\nabla$. which denotes the fact that $\boldsymbol{w}$ is a vector. For a vector $\boldsymbol{w}$ of length $d$, we have the gradient vector:

$$\nabla \ell\Big(y, f(\boldsymbol{x}; \mathbf{w})\Big) = \left[ \frac{\partial \ell(y, f(\boldsymbol{x}; \mathbf{w}))}{\partial \mathbf{w}_1}, \frac{\partial \ell(y, f(\boldsymbol{x}; \mathbf{w}))}{\partial \mathbf{w}_2}, \ldots, \frac{\partial \ell(y, f(\boldsymbol{x}; \mathbf{w}))}{\partial \mathbf{w}_d} \right]^T. \tag{7}$$

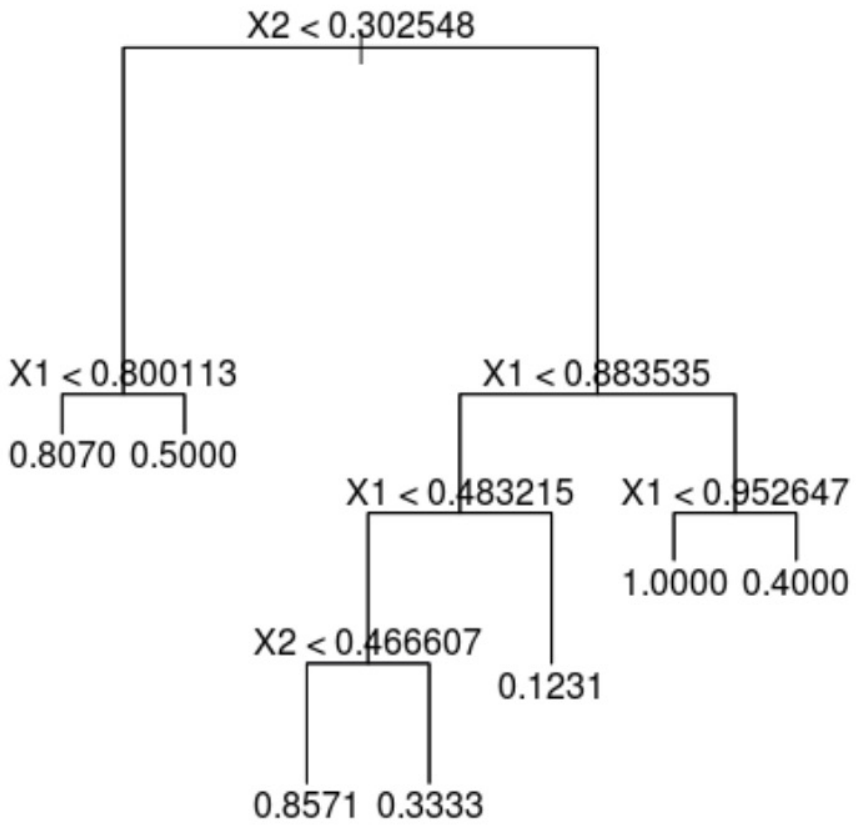
# AN IMPORTANT CLASS OF MODELS: DECISION TREES

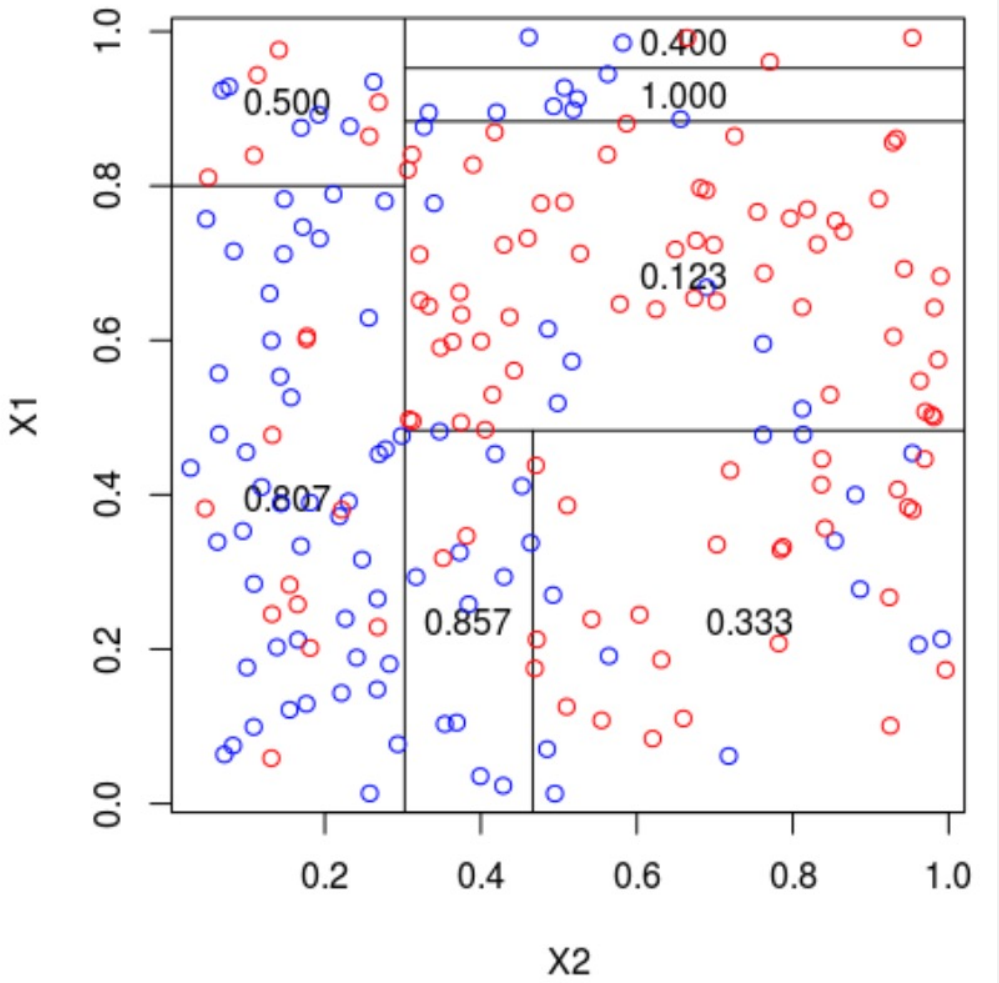


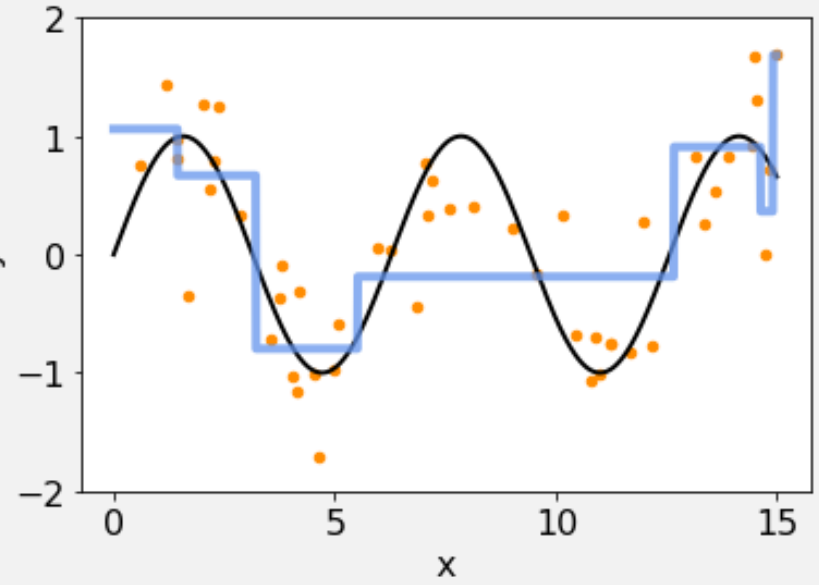
# AN IMPORTANT CLASS OF MODELS: DECISION TREES

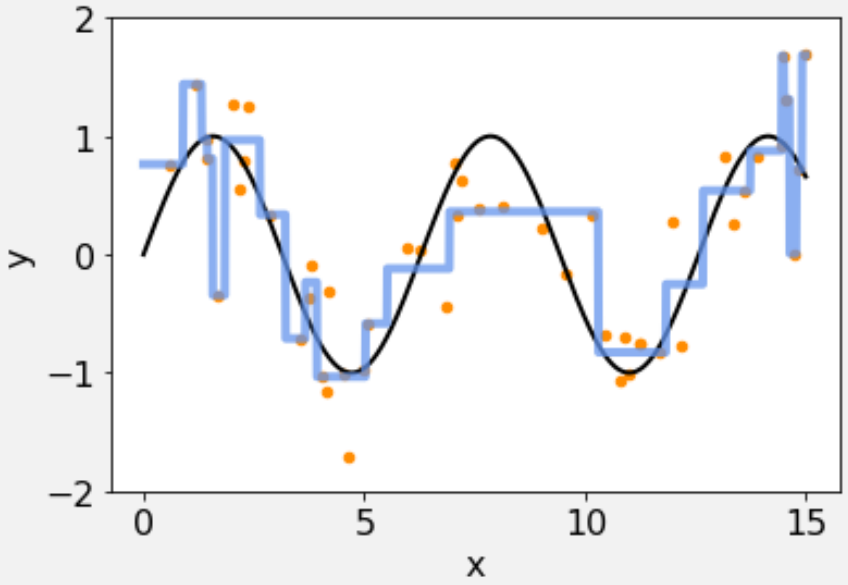
Tree depth

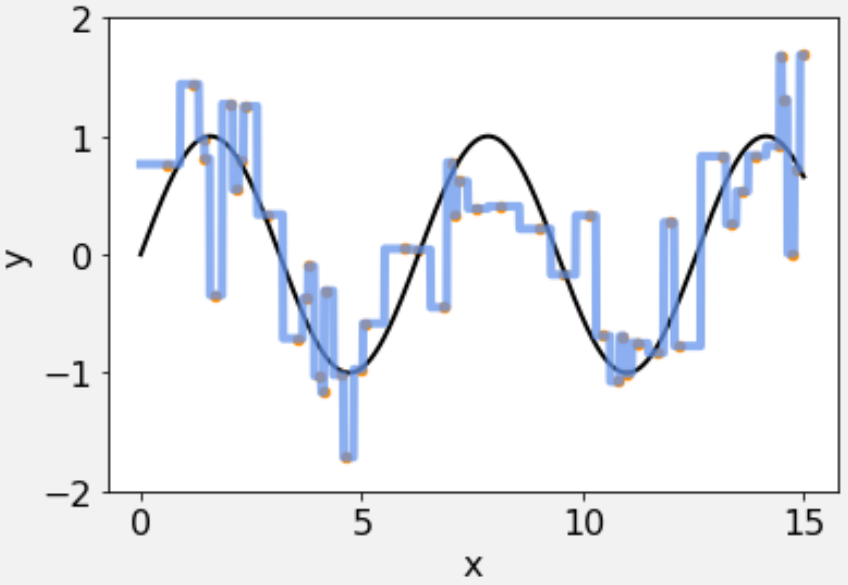


# AN IMPORTANT CLASS OF MODELS: DECISION TREES



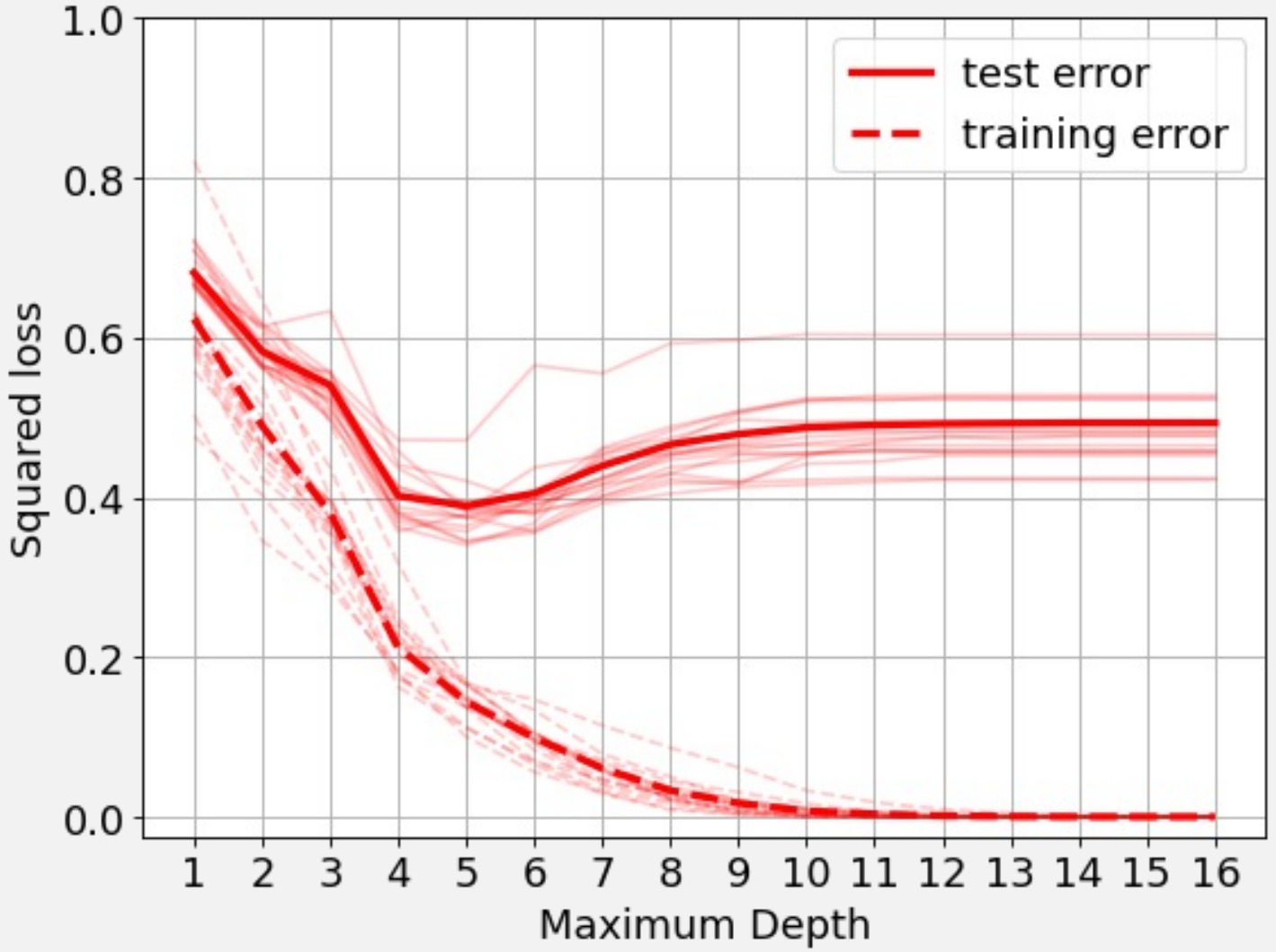




Depth = 3
Simple tree.
**Underfitted**.

Depth = 16
Complex tree.
**Overfitted**.

# AN IMPORTANT CLASS OF MODELS: DECISION TREES

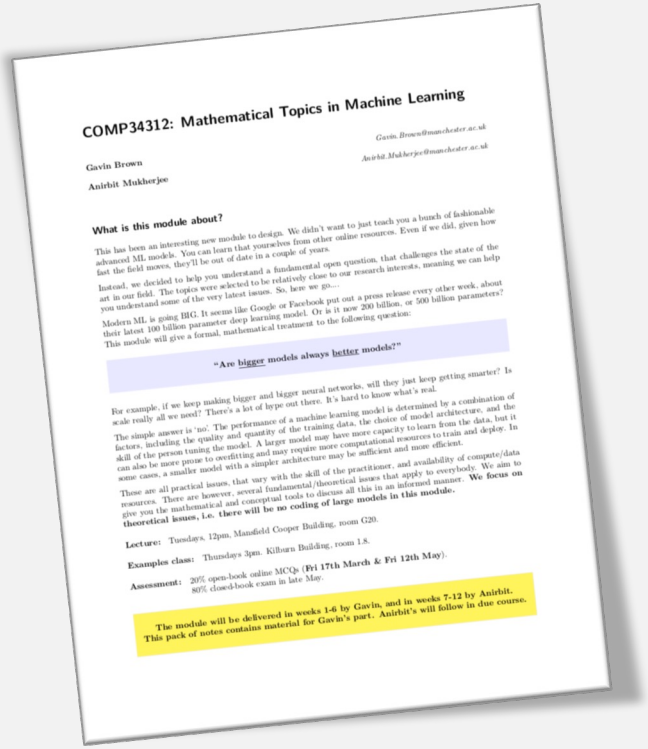


# TODAY HAS BEEN A (VERY QUICK) RECAP OF SUPERVISED LEARNING

- the training error, testing error
- overfitting versus underfitting
- the squared loss,
- the cross-entropy loss,
- the gradient of a loss, and global vs local minima,
- gradient descent versus stochastic gradient descent,
- regression trees, and the meaning/importance of tree ‘depth’.

# LEARNING RESOURCES

Notes pack also on BB.



COMP34312: Mathematical Topics in Machine Learning

Gavin Brown — Gavin.Brown@manchester.ac.uk
Anirbit Mukherjee — Anirbit.Mukherjee@manchester.ac.uk

**What is this module about?**

This has been an interesting new module to design. We didn't want to just teach you a bunch of fashionable advanced ML models. You can learn that yourselves from other online resources. Even if we did, given how fast the field moves, they'll be out of date in a couple of years.

Instead, we decided to help you understand a fundamental open question, that challenges the state of the art in our field. The topics were selected to be relatively close to our research interests, meaning we can help you understand some of the very latest issues. So, here we go....

Modern ML is going BIG. It seems like Google or Facebook put out a press release every other week, about their latest 100 billion parameter deep learning model. Or is it now 200 billion, or 500 billion parameters? This module will give a formal, mathematical treatment to the following question:

**"Are bigger models always better models?"**

For example, if we keep making bigger and bigger neural networks, will they just keep getting smarter? Is scale really all we need? There's a lot of hype out there. It's hard to know what's real.

The simple answer is 'no'. The performance of a machine learning model is determined by a combination of factors, including the quality and quantity of the training data, the choice of model architecture, and the skill of the person tuning the model. A larger model may have more capacity to learn from the data, but it can also be more prone to overfitting and may require more computational resources to train and deploy. In some cases, a smaller model with a simpler architecture may be sufficient and more efficient.

These are all practical issues, that vary with the skill of the practitioner, and availability of compute/data resources. There are however, several fundamental/theoretical issues that apply to everybody. We aim to give you the mathematical and conceptual tools to discuss all this in an informed manner. **We focus on theoretical issues, i.e. there will be no coding of large models in this module.**

**Lecture:** Tuesdays, 12pm, Mansfield Cooper Building, room G20.

**Examples class:** Thursdays 3pm. Kilburn Building, room 1.8.

**Assessment:** 20% open-book online MCQs (**Fri 17th March & Fri 12th May**).
80% closed-book exam in late May.

**The module will be delivered in weeks 1-6 by Gavin, and in weeks 7-12 by Anirbit. This pack of notes contains material for Gavin's part. Anirbit's will follow in due course.**

This week:

- study the slides and notes
- re-read your COMP24112 notes

Try to read ahead a little into week 2 if you want.

We will be accelerating the mathematical material **significantly**, especially after week 7.