CSC311 Introduction to Machine Learning

Supervised Learning and Nearest Neighbours

Alice Gao

UNIVERSITY OF TORONTO

DEFY GRAVITY

Learning Outcomes (Supervised Learning)

By the end of this lecture, you will be able to

- Distinguish artificial intelligence, machine learning, and deep learning.
- Distinguish supervised, unsupervised, and reinforcement learning.
- Formulate a problem as a supervised learning problem in formal notation.
- Explain the purposes and uses of the training, validation, and test sets.

Learning Outcomes (KNN)

By the end of this lecture, you will be able to

- Classify a new data point by applying KNN to a small data set.
- Draw the decision boundaries of KNN for a small data set.
- Analyze the impact of varying k on KNN's behaviour.
- Explain the common pitfalls of KNN.

Outline

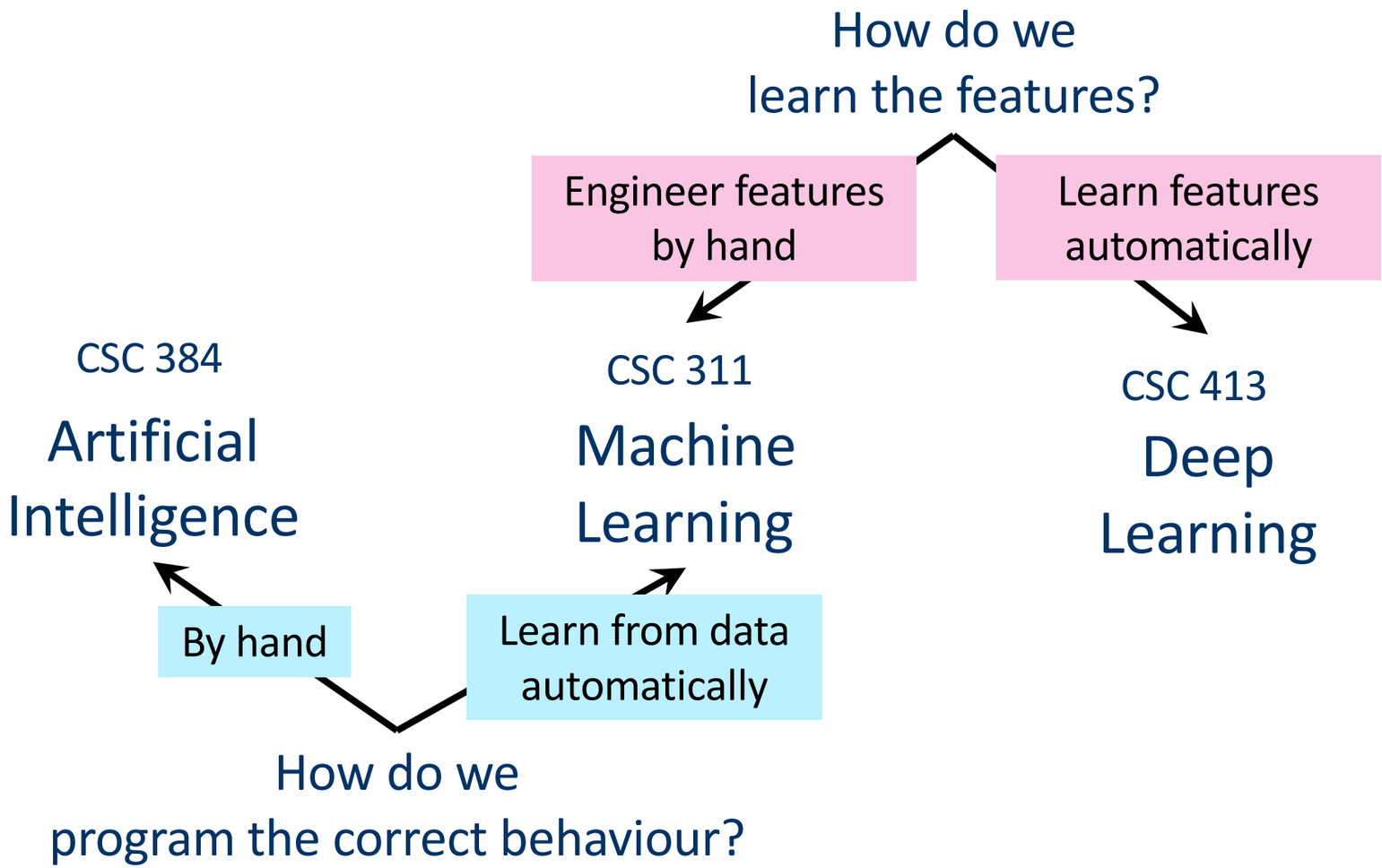
- Introduction to Machine Learning
- Nearest Neighbours
- Decision Boundaries
- K Nearest Neighbours
- Training, Validation and Test Sets
- Limitations of KNN

Introduction to Machine Learning

UNIVERSITY OF TORONTO

DEFY GRAVITY

Artificial Intelligence vs. Machine Learning vs. Deep Learning



Supervised learning -> predict ground truth labels of new data
Unsupervised learning -> trying to find patterns in unlabeled data

Supervised vs. Reinforcement vs. Unsupervised Learning

Supervised Learning

- data with ground truth labels
- predict labels given inputs

Reinforcement Learning

- receives reward signal and
- tries to maximize the reward signal

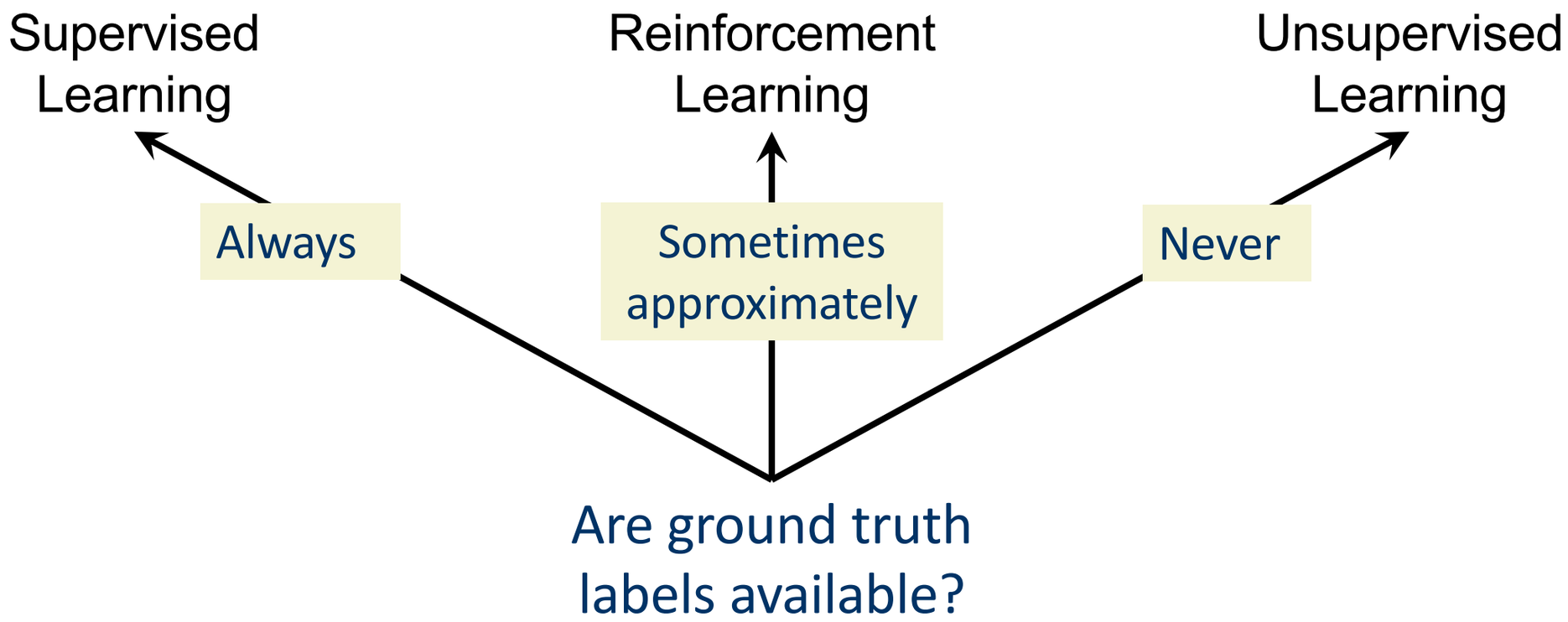
An agent is trying to maximize a reward signal

Unsupervised Learning

- data without ground truth labels
- find pattern in data

Supervised vs. Reinforcement vs. Unsupervised Learning

Trying to find out which decisions maximize the reward



Supervised Learning

Supervised Learning

hypothesis/model → function mapping input data to a label

Objective:

Learn a function that maps an input to an output

based on given input-output pairs

training set, test set

Given the training set (a set of labeled examples)

Produce a hypothesis or model (a function that maps inputs to outputs)

To perform predictions in the test set (unlabeled examples)

Supervised Learning Setup: Input

Input is a vector $\mathbf{x} \in \mathbb{R}^D$:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \cdots & x_D \end{bmatrix}^{\mathrm{T}}$$

Feature: each scalar x_i in $\mathbf{x}$ describes a characteristic of the input.

Many types of data (e.g., text, image, audio) can be represented as vectors.

Vectors are useful because we can use linear algebra.

Supervised Learning Setup: Output

Prediction output

Target output

Goal of Supervised learning: Create a hypothesis/model

Input: a vector $\mathbf{x} \in \mathbb{R}^D$

Target output: $t \in \mathbb{R}$, the ground-truth value that we want to predict.

Prediction output: $y \in \mathbb{R}$, the value generated by our model.

types of labels: classification, regression, structured

Types of output:

- Classification: t is discrete (categorical, a value in a finite set $\{1, \ldots, C\}$).
- Regression: t is continuous (a real number).
- Structured prediction: t is structured (e.g., text, image).

Supervised Learning Setup: Training Set

The training data set has N labeled examples:

$$\left\{\left(\mathbf{x}^{(1)}, t^{(1)}\right), \left(\mathbf{x}^{(2)}, t^{(2)}\right), \cdots, \left(\mathbf{x}^{(\mathrm{N})}, t^{(N)}\right)\right\}$$

The i^{th} data point has feature vector $\mathbf{x}^{(\mathrm{i})}$ and target $t^{(i)}$.

Notation:

- Superscript = index of the data point in the dataset.
- Subscript = index of the feature in the input vector.

Question: What does $x_{12}^{(35)}$ represent?

→ the 12^{th} feature of input vector at index 35

Example: Handwritten Digit Recognition

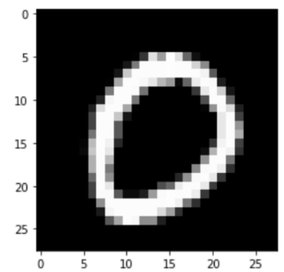
Famous Dataset: MNIST

Problem: Given an image of a handwritten digit, determine which digit the image represents.

- Dataset: MNIST, a large collection of handwritten digits commonly used to benchmark machine learning methods
- Input: a greyscale 28×28 pixel image
- Output: a digit 0-9

An Image is a Matrix

```
In [18]: plt.imshow(pixels, cmap="gray")
Out[18]: <matplotlib.image.AxesImage at 0x7f9390e68bb0>
```



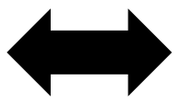

```
In [19]: print(pixels)
[[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   2  53 225 157 255 224 101  16   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0  24 123 252 252 252 253 252 252 212 138   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0  22 201 252 252 252 252 253 252 252 252 243  89   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   2 168 252 252 248 230 182 154  10 102 237 252 249  30   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0 119 252 252 250 115   0   0   0   0   0  56 233 252 206  22   0   0   0   0   0]
 [  0   0   0   0   0   0   0  35 246 252 252 117   0   0   0   0   0   0   0  79 252 252  99   0   0   0   0   0]
 [  0   0   0   0   0   0  32 221 252 252 166   2   0   0   0   0   0   0   0   5 146 251 220  32   0   0   0   0]
 [  0   0   0   0   0   0  67 252 252 199  22   0   0   0   0   0   0   0   0   0   0 231 252  66   0   0   0   0]
 [  0   0   0   0   0   0  67 252 252  88   0   0   0   0   0   0   0   0   0   0   0 231 252 147   0   0   0   0]
 [  0   0   0   0   0   0 115 252 252  88   0   0   0   0   0   0   0   0   0   0   0 231 252 175   0   0   0   0]
 [  0   0   0   0   0   5 187 253 241  39   0   0   0   0   0   0   0   0   0   0  51 244 253  66   0   0   0   0]
 [  0   0   0   0   0  10 199 252 230   0   0   0   0   0   0   0   0   0   0  15 179 252 241  49   0   0   0   0]
 [  0   0   0   0   0   0 176 252 187   0   0   0   0   0   0   0   0   0   0 146 252 252 118   0   0   0   0   0]
 [  0   0   0   0   0   0 176 252 121   0   0   0   0   0   0   0   0   0 136 249 252 237  47   0   0   0   0   0]
 [  0   0   0   0   0   0 105 252 144   0   0   0   0   0   0   0  39 146 249 252 233  55   0   0   0   0   0   0]
 [  0   0   0   0   0   0  67 252 230   0   0   0   0   0  78 104 236 252 252 233  59   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0  64 250 250  88  12  12  31 155 250 253 252 252 237  55   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0 142 252 252 252 252 252 252 252 253 241 118  47   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0  26 245 252 252 252 252 252 219 161  49   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0  89 166 252 247 142 142  33   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]]
```

UNIVERSITY OF TORONTO

From a Matrix to a Vector

Image

0	0	0
0	2	53
24	123	252



Vector

0
0
0
0
2
53
24
123
252

Input Dimension for MNIST Dataset

Handwritten Digit Recognition Problem:

- Input: a greyscale 28×28 pixel image
- Output: a digit 0-9

Question: What is the dimension of an input vector $\mathbf{x}$?

$x \in \mathbb{R}^{784}$

$x \in \mathbb{R}^D$, vector of scalar features

Summary: Supervised Learning Terminologies

target: $t \in \mathbb{R}$ ground truth value we want to predict

Input	$\mathbf{x} \in \mathbb{R}^D$, a vector of input features
Feature	$x_j \in \mathbb{R}$, a scalar value describing a characteristic of the input
Target	$t \in \mathbb{R}$, a scalar ground-truth value that we aim to predict
Prediction	$y \in \mathbb{R}$, a scalar predicted value generated by a model
Data Point	$(\mathbf{x}, t)$, a pair of input-output values
Data Set	$\left(\mathbf{x}^{(1)}, t^{(1)}\right), \ldots, \left(\mathbf{x}^{(\mathrm{i})}, t^{(i)}\right), \ldots, \left(\mathbf{x}^{(\mathrm{N})}, t^{(N)}\right)$, a set of labeled examples

Learning: tds $\{x^{(i)}, t^{(i)}\}_{i=1}^{N}$
→ ground truth labels

Supervised Learning: Learning versus Inference

A hypothesis or model is a function f mapping input $\mathbf{x}$ to output prediction y.

Inference → using a model to predict value y

Learning

Find a model f that maps from inputs to outputs given a labeled training dataset.

Training set $\{(\mathbf{x}^{(i)}, t^{(i)})\}_{i=1}^{N}$ → Model f

training the model to make a y prediction

Inference

Use a model f to predict an output y given a new example $\mathbf{x}$

Input $\mathbf{x}$, Model f → Prediction y

UNIVERSITY OF TORONTO

Supervised Learning: Model Family

A model is a function f mapping input $\mathbf{x}$ to output prediction y.

Model family -> set of models that share same structure

A model family is a set of models that share the same structure

(e.g., linear models, decision trees, neural networks)

but differ in their parameters.

parameters / features of the data themselves

Supervised Learning: Evaluation

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \mathbb{I}[t^{(i)} = y^{(i)}]}{N}$$

- Evaluation is assessing how well a model performs.
- Evaluation is generally independent of the model family.

Accuracy is measured as fraction of correct predictions in a data set

For classification, we can measure the model accuracy (or error) as the fraction of correct (or incorrect) predictions in a data set.

ground truth label

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \mathbb{I}[t^{(i)} = y^{(i)}]}{N}, \qquad \text{Error} = 1 - \text{Accuracy}$$

where $\mathbb{I}[e] = 1$ if condition e is true, and 0 otherwise.

Nearest Neighbours

Training Set and New Point

Training Data

$\mathbf{x}^{(i)}$	$t^{(i)}$
	8
	5
	2
	0
	7
	1

New Data

$\mathbf{x}$	y	
	?	1

Distance Between First Training Point and New Point

Training Data

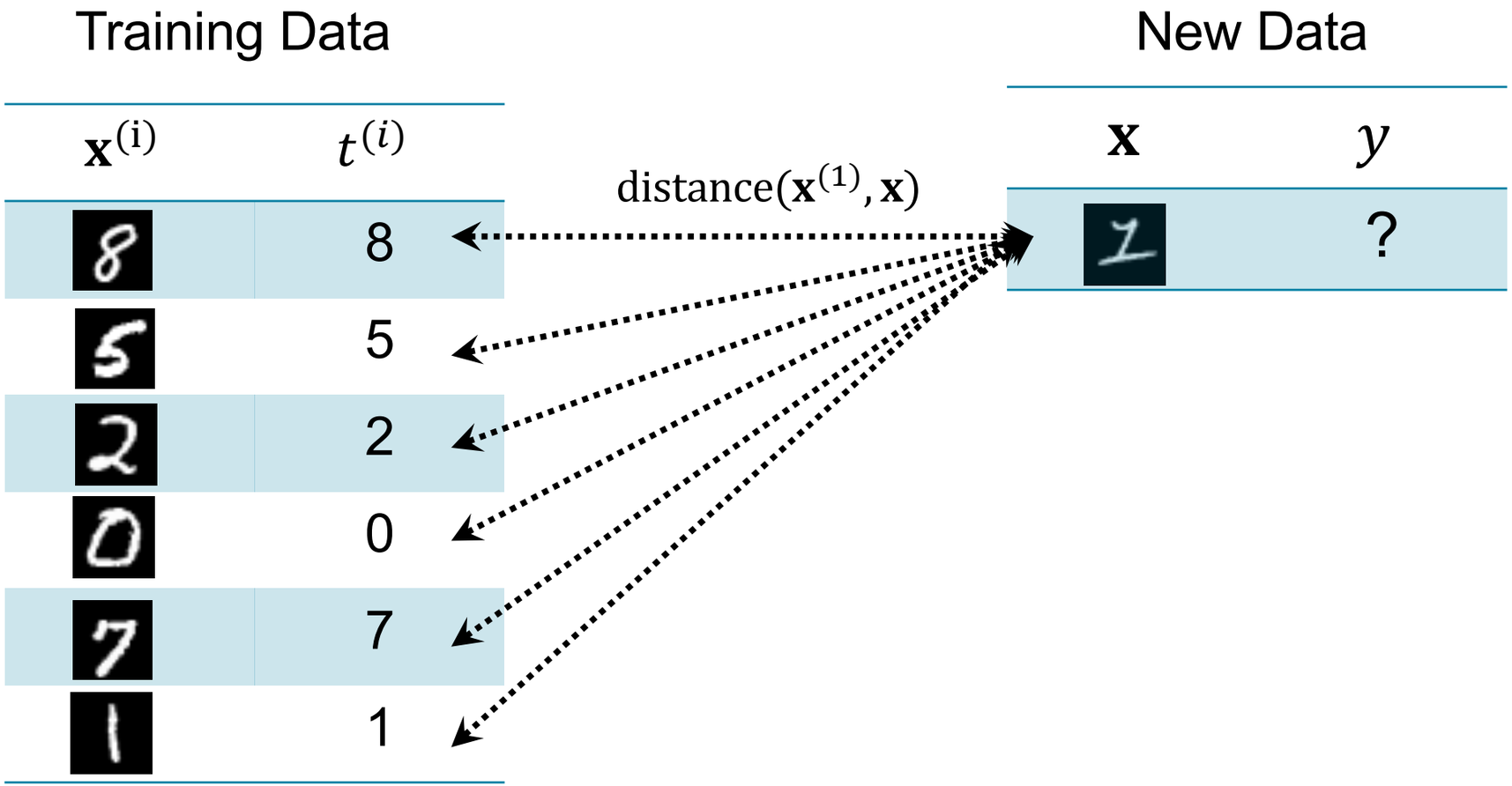
$\mathbf{x}^{(i)}$	$t^{(i)}$
	8
	5
	2
	0
	7
	1

$\text{distance}(\mathbf{x}^{(1)}, \mathbf{x})$

New Data

$\mathbf{x}$	y
	?

Compute Distances to All Training Points

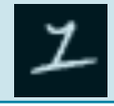


Identify Closest Neighbour

Training Data

$\mathbf{x}^{(i)}$	$t^{(i)}$	
	8	$\text{distance}(\mathbf{x}^{(1)}, \mathbf{x})$
	5	$\text{distance}(\mathbf{x}^{(2)}, \mathbf{x})$
	2	$\text{distance}(\mathbf{x}^{(3)}, \mathbf{x})$
	0	$\text{distance}(\mathbf{x}^{(4)}, \mathbf{x})$
	7	$\text{distance}(\mathbf{x}^{(5)}, \mathbf{x})$
	1	$\text{distance}(\mathbf{x}^{(6)}, \mathbf{x})$

New Data

$\mathbf{x}$	y
	?

Predict by Copying Closest Neighbour's Label

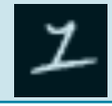
Training Data

$\mathbf{x}^{(i)}$	$t^{(i)}$
	8
	5
	2
	0
	7
	1

Closest training data point

Copy its label

New Data

$\mathbf{x}$	y
	7

Copied label

Pick the closest training data point, and
Copy its label (target)

Nearest Neighbours (Formal Description) 1/2

1. Calculate the distance between every training data point $\mathbf{x}^{(\mathrm{i})}$ and the new data point $\mathbf{x}$.

$$\text{distance}(\mathbf{x}^{(\mathrm{i})}, \mathbf{x})$$

2. Find training data point $(\mathbf{x}^{(\mathrm{c})}, t^{(c)})$ that is closest to the new data point $\mathbf{x}$.

$$\text{distance}(\mathbf{x}^{(\mathrm{c})}, \mathbf{x}) = \min_{i \in 1,\ldots,N} \text{distance}(\mathbf{x}^{(\mathrm{i})}, \mathbf{x})$$

3. Predict $y = t^{(c)}$ as the label for the new data point $\mathbf{x}$.

distance $(x^{(i)}, X)$

Nearest Neighbours (Formal Description) 2/2

1. Calculate the distance between every training data point $\mathbf{x}^{(\mathrm{i})}$ and the new data point $\mathbf{x}$.

$$\mathrm{distance}(\mathbf{x}^{(\mathrm{i})}, \mathbf{x})$$

2. Find training data point $(\mathbf{x}^{(\mathrm{c})}, t^{(c)})$ that is closest to the new data point $\mathbf{x}$.

$x^{(c)} = \text{argmin distance}(x^{(i)}, X)$, $x^{(i)} \in$ training set

$$\mathbf{x}^{(\mathrm{c})} = \underset{\mathbf{x}^{(\mathrm{i})} \in \text{training set}}{\arg\min} \ \mathrm{distance}(\mathbf{x}^{(\mathrm{i})}, \mathbf{x})$$

(compare with previous notation: $\mathrm{distance}(\mathbf{x}^{(\mathrm{c})}, \mathbf{x}) = \min_{i \in 1,\ldots,N} \mathrm{distance}(\mathbf{x}^{(\mathrm{i})}, \mathbf{x})$)

3. Predict $y = t^{(c)}$ as the label for the new data point $\mathbf{x}$.

How Do We Measure Distance?

Manhattan distance
$\|x^{(a)} - x^{(b)}\|_2$

Must choose a distance measure, depending on the problem we are solving.

Euclidean distance (L^2 distance)

a reasonable default choice.

other L^p distances also reasonable.

$x^{(c)} = \arg\min_{x^{(i)} \in \text{training set}} \text{distance}(x^{(i)}, x)$

$$\|\mathbf{x}^{(a)} - \mathbf{x}^{(b)}\|_2 = \sqrt{\sum_{j=1}^{D} \left(x_j{}^{(a)} - x_j{}^{(b)}\right)^2}$$

Cosine similarity

measures angle between two vectors.

$\text{cosine}(x^{(a)} - x^{(b)}) = \frac{x^{(a)} \cdot x^{(b)}}{\|x^{(a)}\| \|x^{(b)}\|}$

$$\text{cosine}\left(\mathbf{x}^{(a)} - \mathbf{x}^{(b)}\right) = \frac{\mathbf{x}^{(a)} \cdot \mathbf{x}^{(b)}}{\|\mathbf{x}^{(a)}\|_2 \|\mathbf{x}^{(b)}\|_2}$$

data point $x \in \mathbb{R}^2$

Example: Citrus Fruit Prediction

Classify whether a fruit is an **orange** or a **lemon** based on its **weight** and **height** using the training data.

Weight	Height	Label
6.8	7.4	Orange
7.1	7.5	Orange
7.6	8.2	Orange
7.2	7.2	Orange
7.2	9.3	Lemon
7.3	9.5	Lemon
7.2	9.0	Lemon
6.9	7.0	Lemon

Classifying a New Fruit

Weight	Height	Label
6.8	7.4	Orange
7.1	7.5	Orange
7.6	8.2	Orange
7.2	7.2	Orange
7.2	9.3	Lemon
7.3	9.5	Lemon
7.2	9.0	Lemon
6.9	7.0	Lemon

A new fruit has weight 7.4 and height 8.5.
How should we classify it using Nearest Neighbours?

The new data point is $\mathbf{x} =$ $\begin{bmatrix} 7.4 \\ 8.5 \end{bmatrix}$

Calculate distance to each training point:

$$\text{distance}\left(\mathbf{x}^{(1)}, \mathbf{x}\right) = \sqrt{(7.4-6.8)^2 + (7.4-8.5)^2}$$

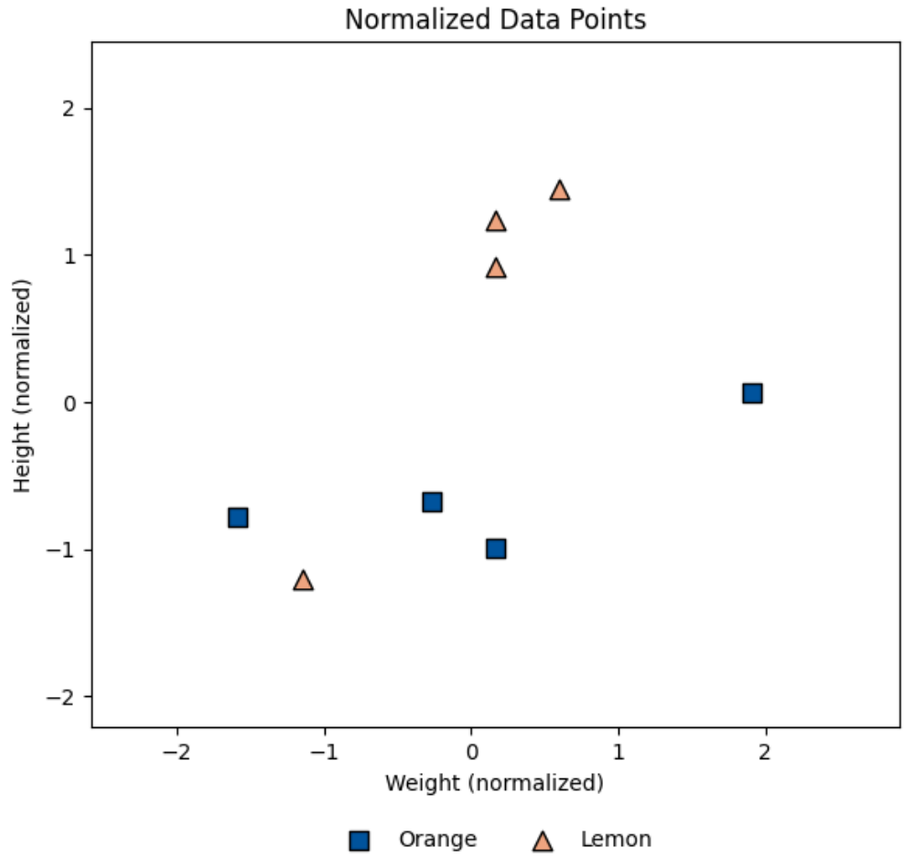
$$= \_\_\_\_\_$$

Similarly, calculate $\left(\mathbf{x}^{(i)}, \mathbf{x}\right)$, for $i = 2, \ldots, 8$.

Copy the label of the closest training example.

Nearest Neighbours Decision Boundaries

Visualize Decision Boundaries



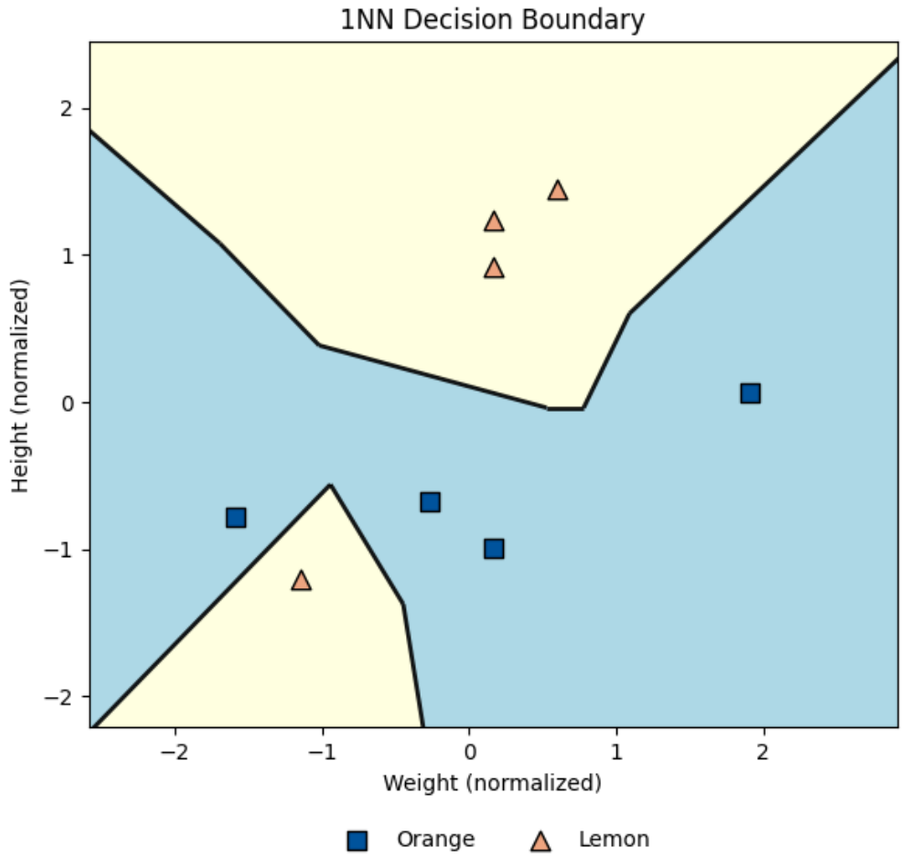
Height is typically lemon

Weight	Height	Fruit
6.8	7.4	Orange
7.1	7.5	Orange
7.6	8.2	Orange
7.2	7.2	Orange
7.2	9.3	Lemon
7.3	9.5	Lemon
7.2	9.0	Lemon
6.9	7.0	Lemon

Nearest Neighbours Decision Boundary

A decision boundary is

- a region where the classifier output is ambiguous
- a dividing line/surface where the classifier switches from predicting one class to another

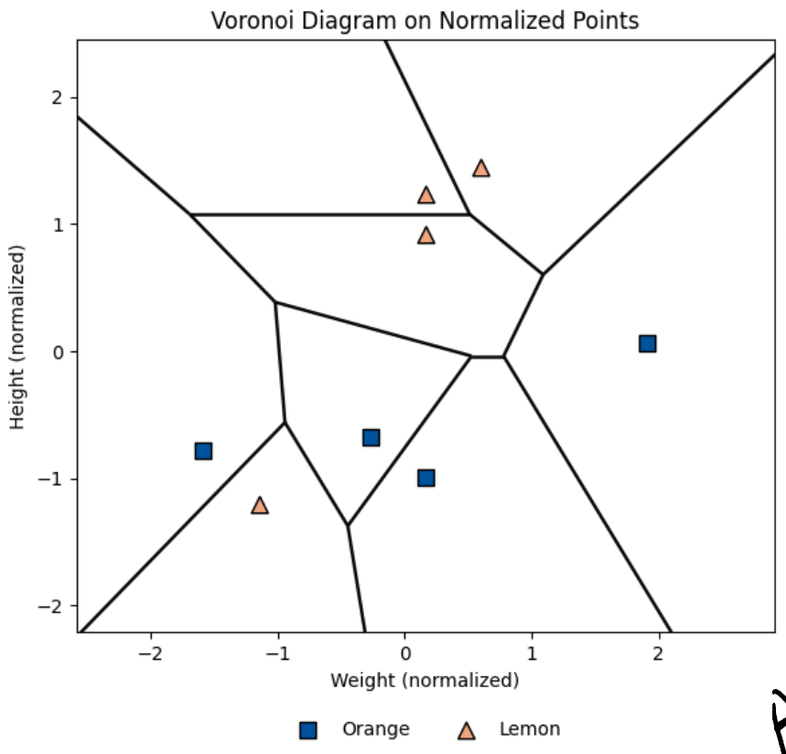


1NN Decision Boundary via Voronoi Diagram

Voronoi Diagram

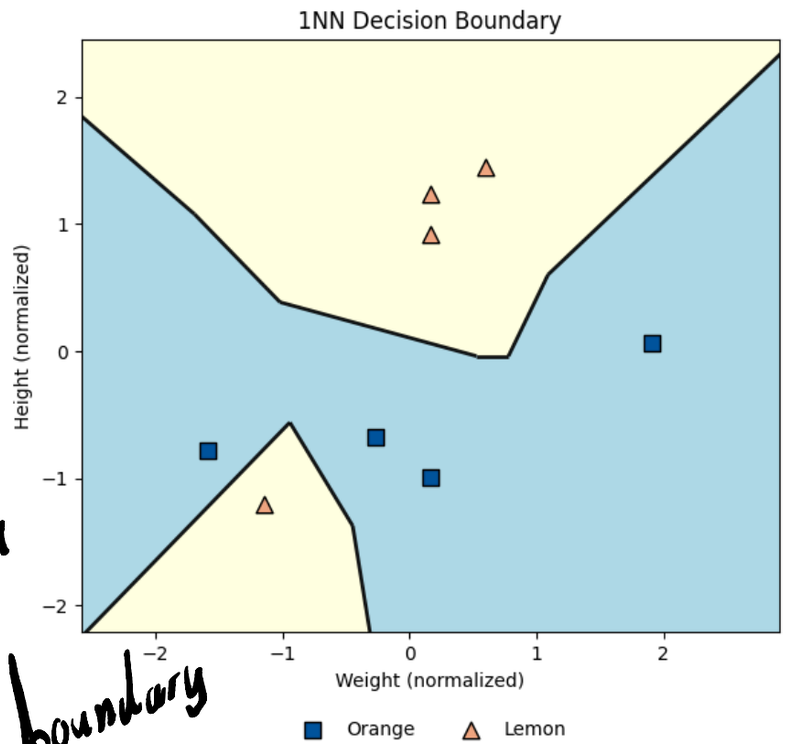
Perpendicular bisections

1NN Decision Boundary

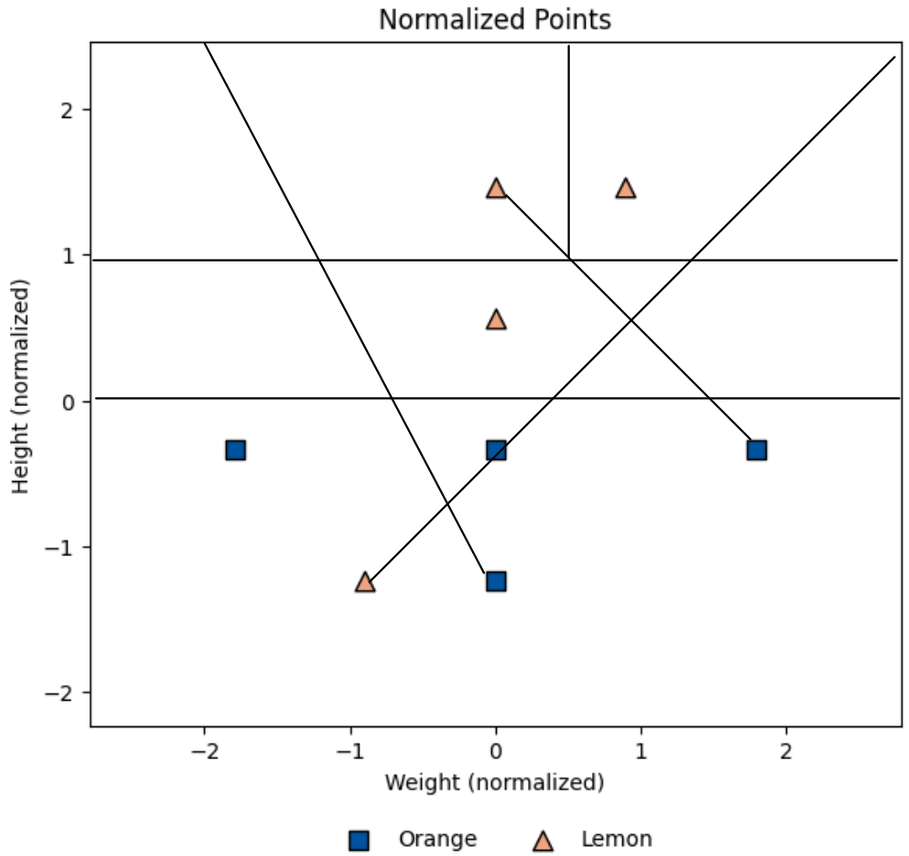


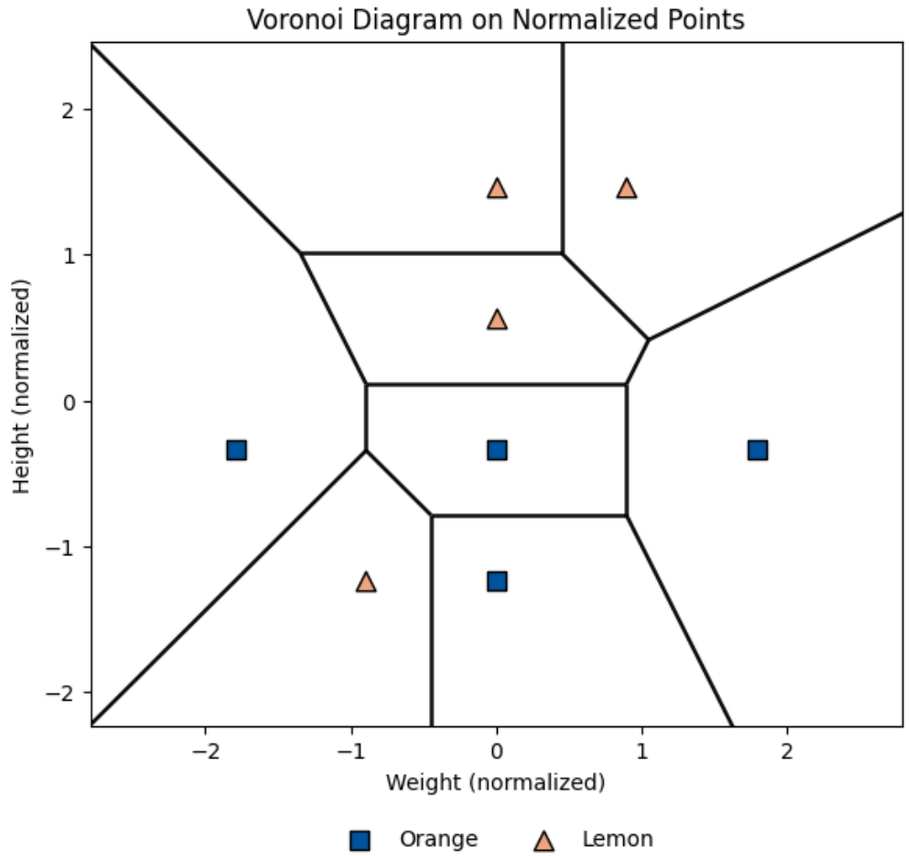
Difference between Voronoi and 1NN:

Voronoi creates regions for each point, 1NN creates decision boundary where classifiers change



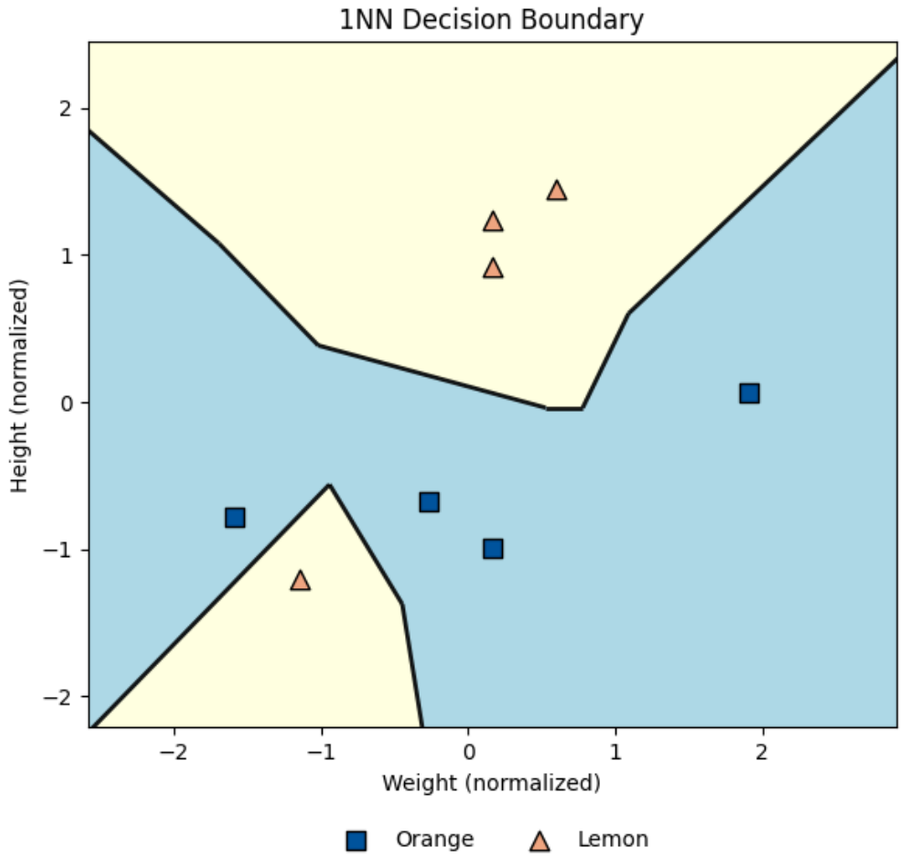
Sketch Voronoi Diagram





K Nearest Neighbours

Problem with Nearest Neighbours



The NN model is sensitive to noise or mis-labeled data.

Which area may have unreasonable predictions?

k nearest neighbours

K Nearest Neighbours (KNN)

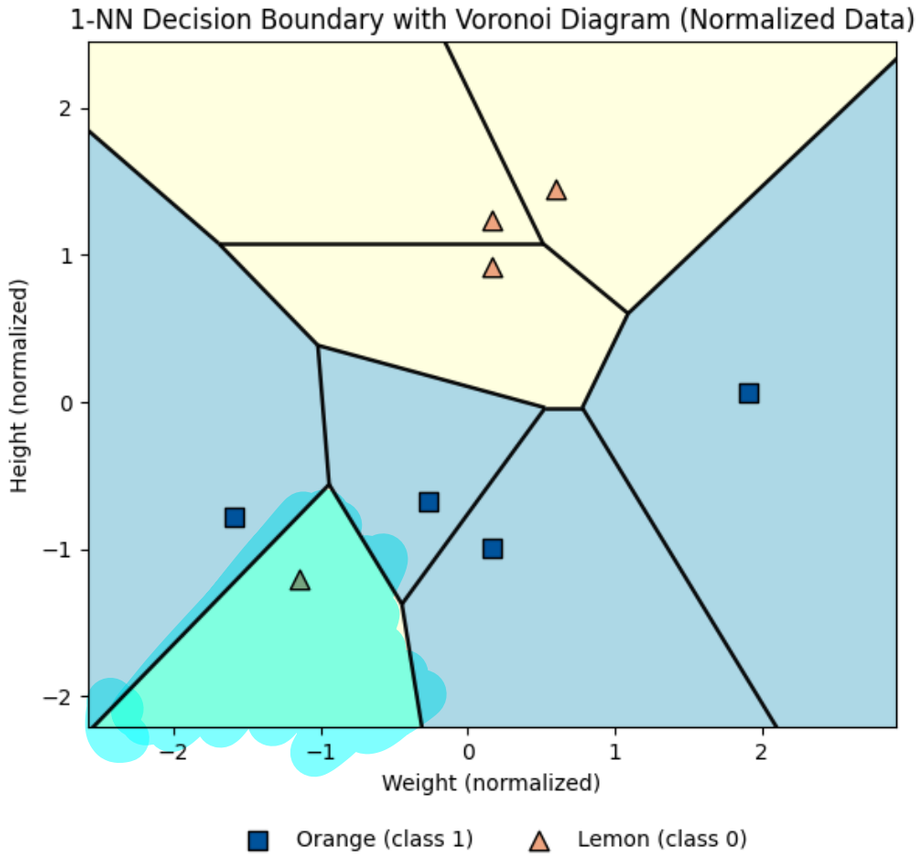
Instead of using the closest point, use K closest points for prediction.

K Nearest Neighbours:

- Find the K closest training points to new data point $\mathbf{x}$.
- Predict the label by majority vote among K nearest neighbours.

3NN Decision Boundary (Optional Exercise)

Find a region in the data space where 1NN and 3NN models predict different labels.



K is a Hyperparameter of KNN

learning:
training set $\{x^{(i)}, t^{(i)}\}_{i=1}^{N}$ → model
inference
new data x, model ——> Prediction label

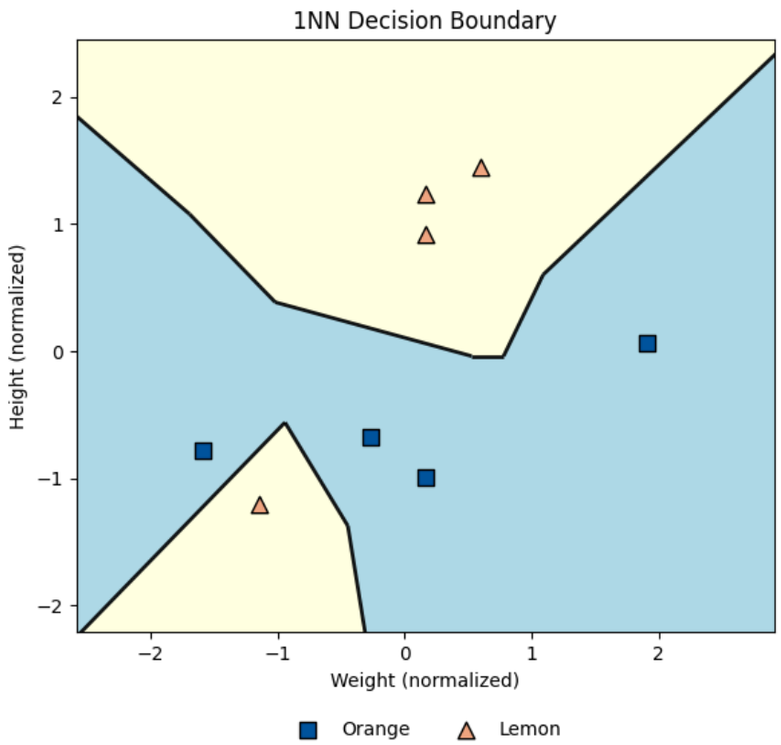
A hyperparameter is

- A setting or configuration external to the model
- Chosen before training → Creating the model itself / what you need to know about the model
- Governs how the model is learned
- Strongly influences model behaviour and performance

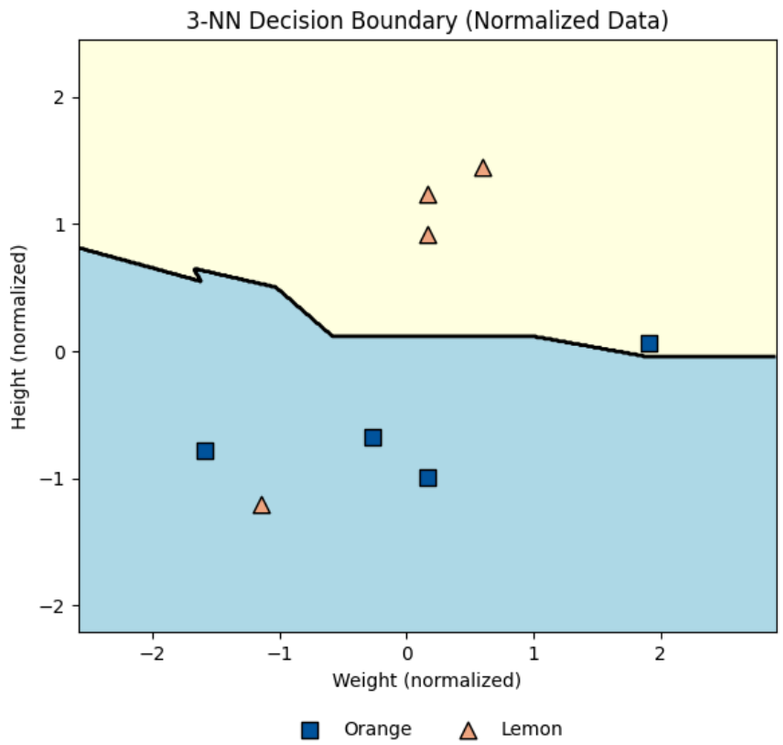
model example: for kNN, choosing K determines how the function works

Effects of Varying K

$K = 1$

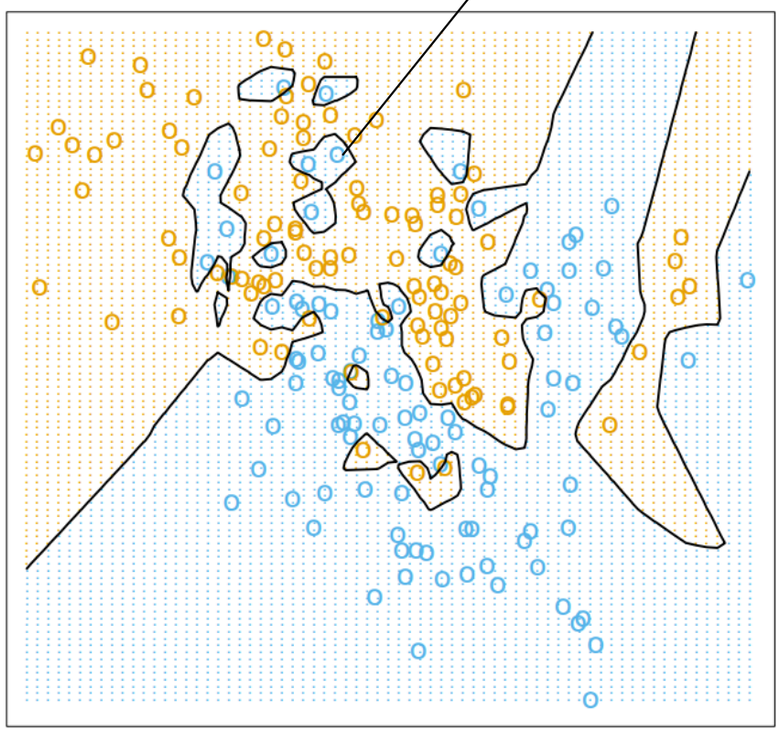


$K = 3$

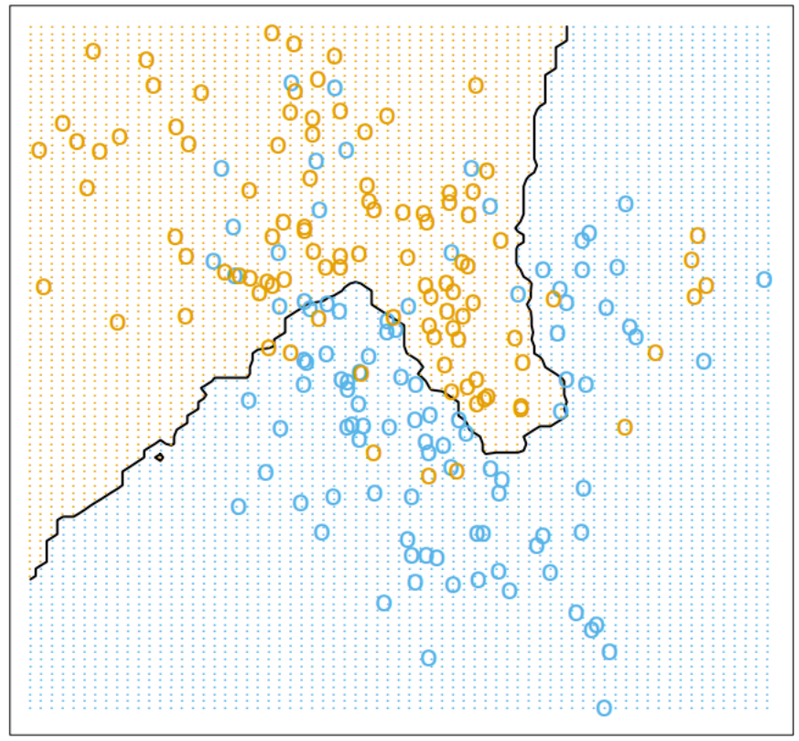


Effects of Varying K

$K = 1$ Overfits



$K = 15$



Tradeoffs in Choosing K

K is small => overfitting (high variance)
K is large => underfitting (high bias)

K is too small

- Decision boundary very complex
- Captures fine-grained patterns
- Highly sensitive to noise

May **overfit (high variance)**

K is too large

- Decision boundary very smooth
- Misses important/local patterns
- Produces stable predictions by averaging many neighbours

May **underfit (high bias)**

Tuning Hyperparameters

Choosing K in KNN

General approach

No single "best" choice of K

Depends on dataset size N

Common heuristic: $K < \sqrt{N}$

Use a validation set

(or cross-validation)

to select hyperparameters

Training, Validation, and Test Sets

Why Split the Data?

What do we need to do with our data?

- Train a model.
- Choose hyperparameters before training a model.
- Evaluate our model on unseen data.

Solution:

- Split available data into training, validation, and test sets.

$$\frac{\sum \mathbb{I}\left(f^{(i)} = y^{(i)}\right)}{N}$$

Roles of Training, Validation, and Test Sets

Citrus fruit example: width and height of various fruit

- Training set is used to learn model parameters.
- Validation set is used to tune hyperparameters and choose between models.
- Test set is used only once at the end to evaluate how well our model generalizes to unseen data.

Training Set	Validation Set	Test Set
Learn model parameters	Tune hyperparameters	Measure generalization error once, at the end

Workflow with KNN

$k \in \{1, 2, 3\}$

Divide data into training, validation, and test sets.
Train kNN model on the training set
find out which k hyperparameter is most accurate
Find the accuracy of the most accurate on new training set

1. Divide the dataset into training, validation and test sets.
2. For each $K \in \{1, 2, 3\}$, train a KNN model on the training set.

What happens at step 2?

3. Compute accuracy of each KNN model on the validation set.

What happens at step 3?

4. Select value of K with best validation accuracy. Assume $K = 3$ is best.
5. Compute accuracy of the KNN model with $K = 3$ on the test set.

Exercise: Which should you not do with the test set?

A. Perform data exploration.

B. Perform feature engineering.

C. Learn model parameters.

D. Tune hyperparameters.

E. Report performance on the set as the final unbiased result.

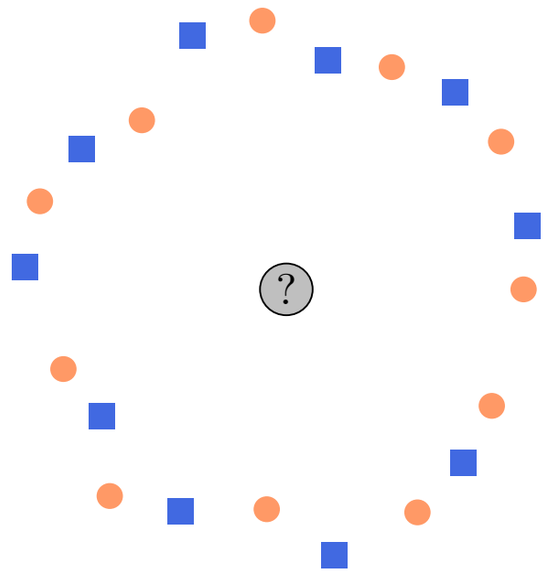
Limitations of KNN

Curse of Dimensionality

As dimensions grow, the data space expands exponentially, and points becomes very far apart.

If the nearest neighbour is as far as any other point,

it is not very useful for prediction.



Problem: KNN is not suitable for high-dimensional data.

Possible solutions: add more data or reduce dimensionality.

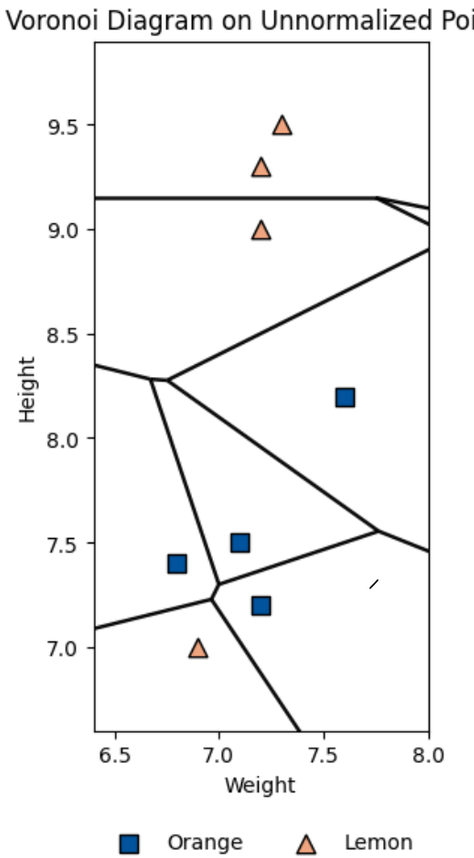
The Problem - Sensitivity to Scale

KNN is sensitive to ranges of features.

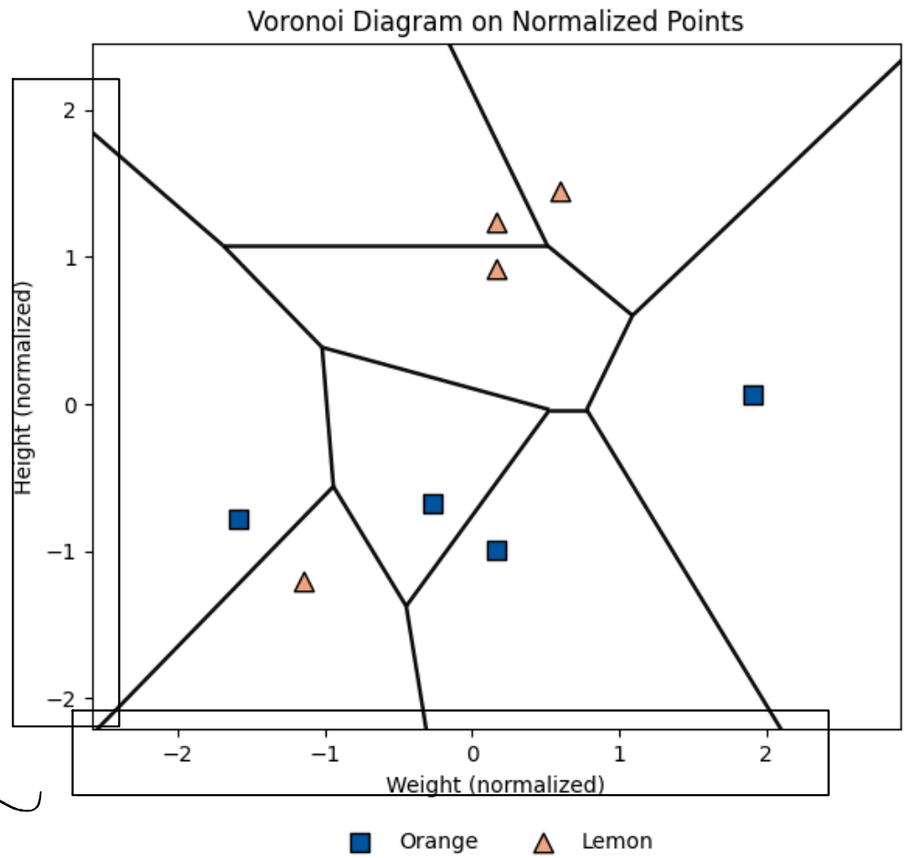
Why:

- KNN relies on distance calculations.
- A feature with a larger range dominates distance calculations.

Predictions for Unnormalized vs. Normalized Data



All normalized



The Solution - Normalization

$N(\mu=0, \sigma^2=1)$

$x_{norm} = \frac{x-\mu}{\sigma}$

Problem: A feature with a larger range dominates distance calculations.

Solution: Normalize features to mean $= 0$ and variance $= 1$.

$$x_{\text{norm}} = \frac{x - \mu}{\sigma}$$

where $\mu =$ mean and $\sigma =$ the standard deviation.

KNN: Advantages and Limitations

Advantages

- Simple and intuitive
- No training required
- Works for both classification and regression
- Adapts easily to new data

Limitations

- Struggles in high dimensions
- Sensitive to feature scaling
- Slow for large datasets
- Must store all training data