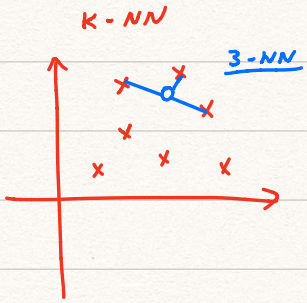
1) For the following dataset:

	apple	*ibm*	*lemon*	*sun*	CLASS
	TRAINING INSTANCES				
A	4	0	1	1	FRUIT
B	5	0	5	2	FRUIT
C	2	5	0	0	COMPUTER
D	1	2	1	7	COMPUTER
	TEST INSTANCES				
T_1	2	0	3	1	?
T_2	1	2	1	0	?

a) Using the **Euclidean distance** measure, classify the test instances using the 1-NN method.

Euclidean (2-norm): vectors A & B

$$d_E(A,B) = \sqrt{\Sigma_K (a_K - b_K)^2}$$



1-NN for T_1: $\langle 2, 0, 3, 1 \rangle$

$$d_E(T_1, A) = \sqrt{(4-2)^2 + (0-0)^2 + (1-3)^2 + (1-1)^2}$$

$$= \sqrt{4+0+4+0}$$

$$= \sqrt{8}$$

$$d_E(T_1, B) = \sqrt{(5-2)^2 + (0-0)^2 + (5-3)^2 + (2-1)^2}$$

$$= \sqrt{9+0+4+1}$$

$$= \sqrt{14}$$

$$d_E(T_1, C) = \sqrt{35}$$

$$d_E(T_1, D) = \sqrt{49}$$

$\Rightarrow$ Classify T_1 as Fruit.

For T_2:

$$d_E(T_2, A) = \sqrt{14}$$

$$d_E(T_2, B) = \sqrt{40}$$

$$d_E(T_2, C) = \sqrt{11}$$

$$d_E(T_2, D) = \sqrt{19}$$

$\Rightarrow$ Classify T_2 as computer

b) Using the **Manhattan distance** measure, classify the test instances using the 3-NN method, for the three weightings we discussed in the lectures: *majority class*, *inverse distance*, *inverse linear distance*.

$$d_M(A,B) = \Sigma_k |a_k - b_k|$$

$$d_M(T_1, A) = |4-2| + |0-0| + |1-3| + |1-1|$$

$$= 2+0+2+0$$

$$= 4$$

$$d_M(T_1, B) = 6$$

$$d_M(T_1, C) = 9$$

3-NN : A, B, C

$$d_M(T_1, D) = 11$$

	apple	*ibm*	*lemon*	*sun*	CLASS
	TRAINING INSTANCES				
A	4	0	1	1	FRUIT
B	5	0	5	2	FRUIT
C	2	5	0	0	COMPUTER
D	1	2	1	7	COMPUTER
	TEST INSTANCES				
T_1	2	0	3	1	?
T_2	1	2	1	0	?

Classification : (3 weightings)

① Majority Class (equal weights)

2 Fruit, 1 Computer

⇒ Classify as Fruit

② Inverse distance ($w = \frac{1}{d+\varepsilon}$) ← avoid $\frac{1}{0}$

Let $\varepsilon = 1$

for A (fruit) : $\frac{1}{4+1} = 0.2$

for B (fruit) : $\frac{1}{6+1} = 0.14$

for C (comp) : $\frac{1}{9+1} = 0.1$

"score" for fruit = 0.2 + 0.14 = 0.34

0.34 (fruit) > 0.1 (comp) ⇒ Classify as fruit

③ Inverse linear distance ($w_j = \frac{d_3 - d_j}{d_3 - d_1}$) (rescaling dist) ⇒ $w_1 = \frac{d_3 - d_1}{d_3 - d_1} = 1$

d_3 ← furthest, d_1 ← nearest (in NN)

$w_3 = \frac{d_1 - d_1}{d_3 - d_1} = 0$

⇒ $w_j \in [0, 1]$

3-NN: $d_M(T_1, A) = 4$, $d_M(T_1, B) = 6$, $d_M(T_1, C) = 9$, ~~$d_M(T_1, D) = 11$~~

$\Rightarrow$ For A (fruit): $\frac{9-4}{9-4} = 1$

For B (fruit): $\frac{9-6}{9-4} = \frac{3}{5} = 0.6$

For C (comp): $\frac{9-9}{9-4} = 0$

$1 + 0.6 = 1.6$

1.6 (fruit) > 0 (comp) $\Rightarrow$ Classify as fruit.

c) Can we do weighted k-NN using **cosine similarity**?

Yes, easier than distance, use cos similarity as weights directly.

(score)
weight: cosine similarities.

All predictions:

Inst	Measure	k	Weight	Prediction
T_1	d_E	1	-	FRUIT
		3	Maj	FRUIT
		3	ID	FRUIT
		3	ILD	FRUIT
	d_M	1	-	FRUIT
		3	Maj	FRUIT
		3	ID	FRUIT
		3	ILD	FRUIT
	cos	1	-	FRUIT
		3	Maj	FRUIT
		3	Sum	FRUIT
T_2	d_E	1	-	COMPUTER
		3	Maj	FRUIT
		3	ID	FRUIT
		3	ILD	COMPUTER
	d_M	1	-	COMPUTER
		3	Maj	COMPUTER
		3	ID	COMPUTER
		3	ILD	COMPUTER
	cos	1	-	COMPUTER
		3	Maj	FRUIT
		3	Sum	FRUIT

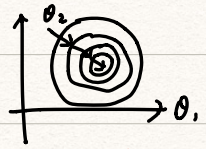
2.

~~1.~~ What is **gradient descent**? Why is it important?

GD: an iterative optimization algorithm (step-by-step)

→ find the params corresponding to optimal points of a target function (e.g. min loss, max likelihood, ...) step-by-step.

→ Start with initial param values, incrementally modify these values in the way that leads to largest improvement. (take derivatives!)



Important: for optimization problem with no closed from solution.

3.

~~6.~~ [OPTIONAL] (a) What is **Regression**? How is it similar to **Classification**, and how is it different?

(b) Come up with one typical classification task, and one typical regression task. Specify the range of valid values of y (results) and possible valid values for x (attributes).

(a) Both supervised learning methods, use labelled train'g dataset to make predictions.

Nominal target → Classification

Numeric target → Regression

(b) Regression: house price prediction

attributes: location, size, age, ...

class (y): real value price (positive)

Classification: Sentiment analysis of movie reviews

attributes: set of words, author ID, length of review, ...

class (y): Weak (1), Not bad (2), Good (3), Great (4), Master piece (5)

4. What is **Discretisation**, and where might it be used?

Continuous attribute ⟶ nominal (or ordinal) attribute.

Some learners, e.g. Decision tree, work better with nominal attributes

Some datasets inherently have grouping of values

⇒ treating them as an equivalent might make it easier to discern underlying patterns.

5. Discretise the following dataset according to the (unsupervised) methods of **equal width** and **equal frequency**.

ID	A (°C)	B (mm)	C (hPa)	CLASS
1	22.5	4.6	1021.2	AUT
2	16.7	21.6	1027.0	AUT
3	29.6	0.0	1012.5	SUM
4	33.0	0.0	1010.4	SUM
5	13.2	16.4	1019.5	SPR
6	14.9	8.6	1016.4	SPR
7	18.3	7.8	995.4	WIN
8	16.0	5.6	1012.8	WIN

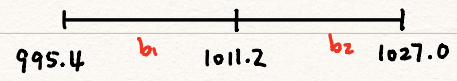
① Equal width

Divide the "range" of possible values seen in training set into equally-sized subdivisions (regardless of # instances in each division)

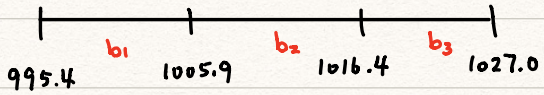
Example:

Attribute C: min = 995.4 max = 1027.0 range = 31.6

2 buckets: bucket width = $\frac{31.6}{2} = 15.8$



3 buckets: bucket width = $\frac{31.6}{3} \approx 10.5$



② Equal frequency

Divide the "range" of possible values seen in training set, s.t. (roughly) the some number of instances appear in each bucket

Example:

Attribute C:

Sort (Asc):

995.4 (7) < 1010.4 (4) < 1012.5 (3) < 1012.8 (8) < 1016.4 (6) < 1019.5 (5) < 1021.2 (1) < 1027.0 (2)

2 buckets:

b1: 7, 4, 3, 8

b2: 6, 5, 1, 2

E.g. $b_1 \rightarrow [995.4, \frac{1012.8+1016.4}{2})$

$b_2 \rightarrow [\frac{1012.8+1016.4}{2}, 1027.0]$