Tutorial Q1

- Three examples from a system for predicting whether a person is over or under the drink driving limit.
  - Gender, Weight, Amount of alcohol in units, Meal type, Duration of drinking session.

Example x1

Gender	female
Weight	60
Amount	4
Meal	full
Duration	90
Class	over

Example x2

Gender	male
Weight	75
Amount	2
Meal	full
Duration	60
Class	under

Query example

Gender	male
Weight	70
Amount	1
Meal	snack
Duration	30
Class	???

a. Normalise all numeric features to the range [0,1].

b. Propose an appropriate global distance function for comparing examples such as the above.

c. Use your proposed distance function to calculate the distances between the query example and the two labelled examples. Which class label would a 1NN classifier assign to the query based on the distances?

Q1a

a. Normalise all numeric features to the range [0,1]

- Min-max normalisation:
  Use min and max values for a given feature to rescale to the range [0,1]

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

  - Weight: numeric range [50,150]
  - Amount: numeric range [1,16]
  - Duration: numeric range [20,230]

Example x1

Weight	(60-50)/(150-50) = 0.1
Amount	(4-1)/(16-1) = 0.2
Duration	(90-20)/(230-20) = 0.333

Example x2

Weight	(75-50)/(150-50) = 0.25
Amount	(2-1)/(16-1) = 0.067
Duration	(60-20)/(230-20) = 0.19

Example x3

Weight	(70-50)/(150-50) = 0.2
Amount	(1-1)/(16-1) = 0
Duration	(30-20)/(230-20) = 0.048

Q1b

b. Propose an appropriate distance function for comparing the examples.

Ordinal features: the distance can be the absolute difference between the two positions in the ordinal list of possible values.

- Meal: {None, Snack, Lunch, Full} = {1, 2, 3, 4}

e.g. d(Snack,Full) = |2-4| = 2

In practice, we often normalise with respect to ordinal list length though this is not absolutely necessary

e.g. |2-4|/4 = 0.5

Gender	Categorical	Overlap / Hamming function
Weight	Numeric	Absolute difference (after normalisation)
Amount	Numeric	Absolute difference (after normalisation)
Meal	Ordinal {None, Snack, Lunch, Full}	Absolute relative rank difference (norm)
Duration	Numeric	Absolute difference (after normalisation)

Q1c

c. Use your proposed distance function to calculate the distances between the query example and the two labelled examples.

Sum over local distance on each feature:
Gender + Weight + Amount + Meal + Duration

We used Manhattan Distance here, but you could also use Euclidean or your custom distance function. As long as it works…

D(x1,q)

Gender	1
Weight	|0.1-0.2| = 0.1
Amount	|0.2-0| = 0.2
Meal	|2-4|/4 = 0.5
Duration	|0.333-0.048| = 0.285

D(x1,q) = 1 + 0.1 + 0.2 + 0.5 + 0.285 = 2.085

D(x2,q)

Gender	0
Weight	|0.25-0.2| = 0.05
Amount	|0.067-0| = 0.067
Meal	|2-4|/4 = 0.5
Duration	|0.19-0.048| = 0.142

D(x2,q) = 0 + 0.05 + 0.067 + 0.5 + 0.142 = 0.759

➞ Label **q** with same class as **x2** ('under')

Q2a

- Pairwise distances between 9 labelled training examples and a new query example **q**, for the system described in Question 2.

a. What class would a 3-NN classifier assign to **q**?

Example	Class	Distance to q
x1	over	1.5
x2	under	2.8
x3	over	1.8
x4	under	2.9
x5	under	2.2
x6	under	3.0
x7	under	2.4
x8	over	3.2
x9	over	3.6

Example	Class	Distance to q
x1	over	1.5
x3	over	1.8
x5	under	2.2
x7	under	2.4
x2	under	2.8
x4	under	2.9
x6	under	3.0
x8	over	3.2
x9	over	3.6

Sort by distance, smallest first

- Over = 2 votes
- Under = 1 vote

→ Label **q** as 'over'

Q2b

- Pairwise distances between 9 labelled training examples and a new query example **q**, for the system described in Question 1.

b. What class would a 4-NN classifier assign to **q**?

Example	Class	Distance to q
x1	over	1.5
x2	under	2.8
x3	over	1.8
x4	under	2.9
x5	under	2.2
x6	under	3.0
x7	under	2.4
x8	over	3.2
x9	over	3.6

Example	Class	Distance to q
x1	over	1.5
x3	over	1.8
x5	under	2.2
x7	under	2.4
x2	under	2.8
x4	under	2.9
x6	under	3.0
x8	over	3.2
x9	over	3.6

Sort by distance, smallest first

- Over = 2 votes
- Under = 2 votes

➡ Tie!

Note top-ranked examples are both 'over'

Q2c

- Pairwise distances between 9 labelled training examples and a new query example **q**, for the system described in Question 2.

c. What class would a <u>weighted</u> 4-NN classifier assign to **q**?

Example	Class	Distance to q	Weight
x1	over	1.5	1/(1.5)^2 = 0.444
x3	over	1.8	1/(1.8)^2 = 0.308
x5	under	2.2	1/(2.2)^2 = 0.207
x7	under	2.4	1/(2.4)^2 = 0.117
x2	under	2.8	...
x4	under	2.9	...
x6	under	3.0	...
x8	over	3.2	...
x9	over	3.6	...

Sort by distance, smallest first.
Calculate weight as square of inverse distance.

- Over = 0.444 + 0.308 = 0.752
- Under = 0.207 + 0.117 = 0.424

➞ Label **q** as 'over'

Q3

- Two examples for estimating the price of second-hand cars are described by 6 features:

Example 007

Manufacturer	Ford
Model	Fiesta
Engine Size	1,100
Fuel	Petrol
Mileage	65,000
Bodywork	Excellent
Price	€3,100

Example 014

Manufacturer	Citroen
Model	BX
Engine Size	1,800
Fuel	Diesel
Mileage	37,000
Bodywork	Fair
Price	€4,500

a. Normalise all numeric features to the range [0,1]. Assume that the feature ranges are: Engine Size 1,000 to 3,000; Mileage 1,000 to 100,000.

b. Propose a suitable distance function. Assume that Bodywork is an ordinal feature that has the possible values {Poor, Fair, Good, Excellent},

c. Use this measure to calculate the distance between the two examples above.

Q3a

a. Normalise all numeric features to the range [0,1]. Note that you can assume that the feature ranges for: Engine Size is 1,000 to 3,000; Mileage is 1,000 to 100,000.

- Min-max normalisation:
  Use min and max values for a given feature to rescale to the range [0,1]

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

Example 007

Manufacturer	Ford
Model	Fiesta
Engine Size	**(1100-1000)/(3000-1000) = 0.05**
Fuel	Petrol
Mileage	**(65000-1000)/(100000-1000) = 0.646**
Bodywork	Excellent

Example 014

Manufacturer	Citroen
Model	BX
Engine Size	**(1800-1000)/(3000-1000) = 0.4**
Fuel	Diesel
Mileage	**(37000-1000)/(100000-1000) = 0.364**
Bodywork	Fair

Q3b

Engine Size	Numeric	Absolute difference (after normalisation)
Fuel	Categorical	Hamming / Overlap
Mileage	Numeric	Absolute difference (after normalisation)
Bodywork	Ordinal {Poor, Fair, Good, Excellent}	Absolute rank difference (normalised)

Sum over distance on each feature:
Engine Size + Fuel + Mileage + Bodywork

Note: Any distance function can be used! All you need to do is make sure it works

Q3c

Example 007 (Normalised)

Engine Size	0.05
Fuel	Petrol
Mileage	0.646
Bodywork	Excellent

Example 014 (Normalised)

Engine Size	0.4
Fuel	Diesel
Mileage	0.364
Bodywork	Fair

Dist(Case 007, Case 014)

Engine Size	|0.05-0.4| = 0.35
Fuel	1
Mileage	|0.646-0.364| = 0.282
Bodywork	|4-2|/4 = 0.5

Dist = 0.35 + 1 + 0.282 + 0.5 = 2.132

* subject to rounding

Note: You can include Manufacturer and Model as categorical features as well if you think they contain information. Feature selection!!