k-NN
k-Nearest Neighbours

k-NN – k-Nearest Neighbours - 1

- A simple algorithm used in classification
- Based on **similarity functions**
- Uses **distance functions** to determine the nearest neighbours
- Classification (prediction) is done using majority voting
- Does not assume anything about the data, other than a distance measure can be calculated consistently between any two instances.
- As such, it is called ***non-parametric*** or ***non-linear*** as it does not assume a functional form

- For **classification**, a case / record / observation is assigned to the class most observed among k nearest observations (neighbours), measured by the distance function
- For **regression**, the average of the predicted attribute is returned
- If k=1, then the case (record) is the same as the nearest neighbour

k-NN – k-Nearest Neighbours - 2

Strength

- Simple to implement and use
- Comprehensible and easy to explain
- Robust to noisy data – averages out the neighbours

Weakness

- One drawback of using distance measure calculation is when the dataset has mixed data types or different measurement scales.
    - ❑ **Eg: x1=Income, x2=years of experience**
        - ✓ x1 will have a greater influence on the distance calculated
        - ✓ May lead to incorrect predictions
        - ✓ Solution: Standardization of Data

Calculating the distance

For continuous variables

Euclidean

$$= \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$

Manhattan

$$= \sum_{i=1}^{n}|p_i - q_i|,$$

Minkowski

$$\left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p}$$

For categorical variables

Hamming

$$= \sum_{i=1}^{n}|p_i - q_i|,$$

p=q -> D=0; p≠q -> D=1

The Hamming Distance is a number used to denote the difference between two strings

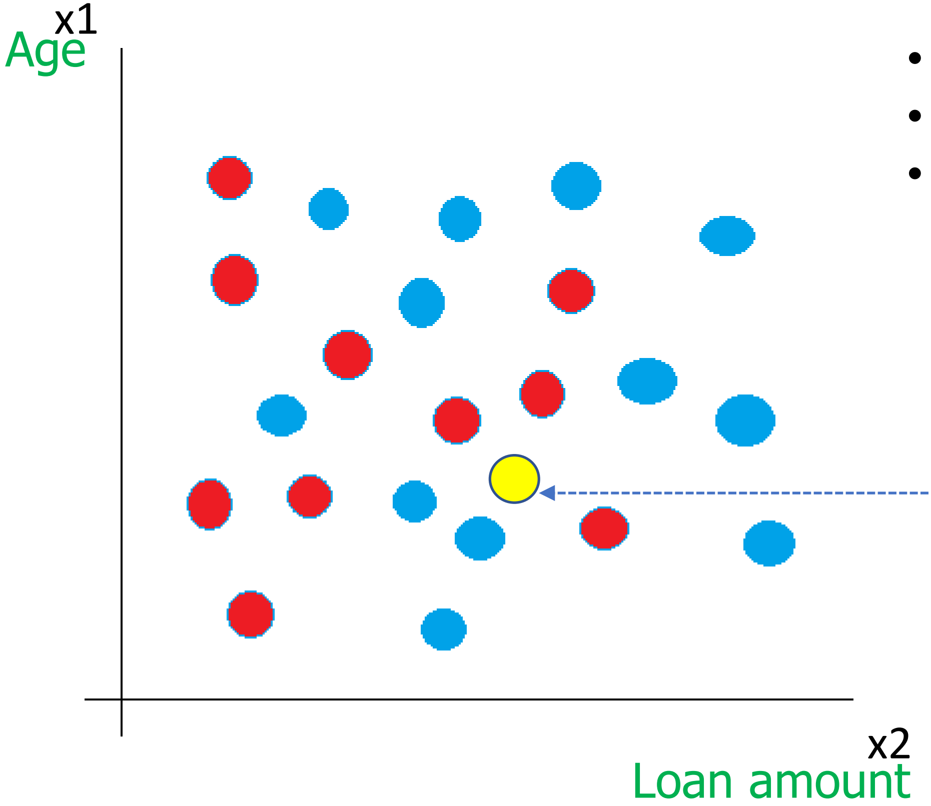
In case of mixed datatypes in dataset

- Standardize the data
- Typical standardization techniques are :

❑	Z-score	(x-μ) / σ
❑	min-max()	[x-min(x)] / [max(x)-min(x)]
❑	Logit	[1 / 1 + e^{x}]

example

k-NN – an illustration



- Let X1 and X2 be two variables eg: Age and Loan amount
- Red represents “loan defaulter”
- Blue represents “loan paid”

Predict the yellow record

loan defaulter / **loan paid**

Selecting the number of neighbours – optimum value for ‘k’

- Inspect the data
- A large **‘k’** value gives a better result
- Perform cross-validation with different ‘k’ values to get the best **‘k’**
- Industry standard / Best practice methodology for the value of **k** is:

  $3 <= k <= 10$
- Select an odd ‘k’ to avoid tie and random selection

Exercise

Given the following dataset, predict if the given record of a customer will default the loan or not?
Take ‘Neighbours’ = 5

$$\sqrt{(48-25)^2 + (91-40)^2}$$

Obs	Age	Loan_amt	Default	Distance
1	25	40	N	55.95
2	27	45	Y	50.57
3	35	70	N	24.70
4	37	68	Y	25.50
5	29	55	N	40.71
6	40	14	N	77.41
7	43	67	N	24.52
8	52	90	Y	**4.12**
9	34	58	Y	35.85
10	38	77	Y	17.20
11	37	85	Y	**12.53**
12	33	79	Y	19.21
13	27	20	N	74.04
14	28	16	N	77.62
15	26	10	N	83.93
16	41	90	Y	**7.07**
17	53	55	Y	36.35
18	49	80	N	**11.05**
19	47	67	N	24.02
20	40	77	Y	**16.12**
21	**48**	**91**	**????**	

Neighbours (k)	Default
4.12	Y
7.07	Y
11.05	N
12.53	Y
16.12	Y
17.20	
19.21	
24.02	
24.52	
24.70	
25.50	
35.85	
36.35	
40.71	
50.57	
55.95	
74.04	
77.41	
77.62	
83.93	

Prediction (48,91)
Y

Data Standardization

Before standardizing values

age (x)	loan (y)	Default	knn	default	knn
25	40000	n	102000	y	8000
35	60000	n	82000	n	22000
45	80000	**n**	22000	y	42000
20	20000	n	122000	y	47000
35	120000	**n**	22000	n	62000
52	18000	n	124000	y	78000
23	95000	**y**	47000	y	80000
40	62000	y	80000	n	82000
60	100000	**y**	42000	n	102000
48	220000	y	78000	n	122000
33	150000	**y**	8000	n	124000
48	142000	y			

Y	N	default
3	2	Y

max(age)	60
min(age)	20
max(loan)	220000
min(loan)	18000

Y	N	default
3	2	Y

After standardizing values

age (x)	loan (y)	default	knn	default	Knn
0.125	0.11	n	0.765	n	0.316
0.375	0.21	n	0.520	n	0.343
0.625	0.31	n	0.316	y	0.365
0	0.01	n	0.925	y	0.377
0.375	0.50	n	0.343	y	0.386
0.8	0.00	n	0.622	y	0.444
0.075	0.38	y	0.667	n	0.52
0.5	0.22	y	0.444	n	0.622
1	0.41	y	0.365	y	0.667
0.7	1.00	y	0.386	n	0.765
0.325	0.65	y	0.377	n	0.925
0.7	0.61	**n**			