Chapter 8: KNN Algorithm

- Knn (k Nearest Neighbors) is a supervised Machine Learning Algorithm.
- kNN is a simple algorithm that stores all available cases and predict the numerical target based on a similarity measure (e.g., distance functions).
- kNN can be used for:
  - **Classification:**
    - prediction task in which target variable is categorical.
  - **Regression:**
    - prediction task in which target variable is numeric.

KNN working:

- **Step1:**
- select the value of K Neighbors

- **Step2:**
- Find the K nearest data point/neighbors for our new data point based on Euclidean Distance.
  - **Euclidean Dist. : sqrt($(X_2 - X_1)^2 + (Y_2 - Y_1)^2$)**
  - **Euclidean Dist. : sqrt($(OV_1 - AV_1)^2 + (OV_2 - AV_2)^2$)**

- **Step3:**
- Combine those neighbors into a prediction for the new data point.

  - **For Classification:**
  - coz the target variable is categorical, prediction/class would be the value that occurs the most(its called mode)
    - Note: if there are multiple modes, there are multiple possible solutions, we can select any random or closest neighbors
  - **For Regression:**
  - coz the target variable is numeric, you can combine multiple neighbors into one prediction by taking the average of their values.

T-SHIRT CLASSIFIER

Person	Height(cm)	Weight(kg)	T-Shirt Size
P1	158	58	M
P2	158	59	M
P3	158	63	M
P4	160	59	M
P5	160	64	L
P6	165	65	L
P7	168	66	L
P8	170	64	L

Person	Height(cm)	Weight(kg)	T-Shirt Size
0	0	0	M
1	0	0.13	M
2	0	0.63	M
3	0.17	0.13	M
4	0.17	0.75	L
5	0.58	0.88	L
6	0.83	1	L
7	1	0.75	L

MinMaxScaler:

- Transform features by scaling each feature to a given range.

This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one.

X_std = X – min / max - min	
X_H_std = 158 – 158 / 170 – 158	
X_W_std = 58 – 58 / 66 - 58	
X_H_std = 158 – 158 / 170 - 158	
X_A_std = 59 – 58 / 66 - 58	

Person	Height(cm)	Weight(kg)	T-Shirt Size
0	0	0	M
1	0	0.13	M
2	0	0.63	M
3	0.17	0.13	M
4	0.17	0.75	L
5	0.58	0.88	L
6	0.83	1	L
7	1	0.75	L

Find the t-shirt size for person who got Height=161cm and Weight = 61Kg

Find the t-shirt size for person who got Height=0.25 and Weight = 0.36

We solve KNN using Euclidean Distance : sqrt($(OV_1 - AV_1)^2 + (OV_2 - AV_2)^2$)

1	sqrt($(0.25-0.00)^2 + (0.36-0.00)^2$)		
2	sqrt($(0.25-0.00)^2 + (0.36-0.13)^2$)		
3	sqrt($(0.25-0.00)^2 + (0.36-0.63)^2$)		
4	sqrt($(0.25-0.17)^2 + (0.36-0.13)^2$)		
5	sqrt($(0.25-0.17)^2 + (0.36-0.75)^2$)		
6	sqrt($(0.25-0.58)^2 + (0.36-0.88)^2$)		
7	sqrt($(0.25-0.83)^2 + (0.36-1.00)^2$)		
8	sqrt($(0.25-1.00)^2 + (0.36-0.75)^2$)		

HEALTH CLASSIFIER

Weight	Height	Class
51	167	Underweight
62	182	Normal
69	176	Normal
64	173	Normal
65	172	Normal
56	174	Underweight
58	169	Normal
57	173	Normal
55	170	Normal

Weight	Height	Class
0	0	Underweight
0.61	1	Normal
1	0.6	Normal
0.72	0.4	Normal
0.77	0.33	Normal
0.27	0.47	Underweight
0.39	0.13	Normal
0.33	0.4	Normal
0.22	0.2	Normal

MinMaxScaler:

❑ Transform features by scaling each feature to a given range.
This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one.

X_std = X – min / max – min	
X_W_std = 51 – 51 / 69 - 51	
X_H_std = 167 – 167 / 182 - 167	
X_W_std = 62 – 51 / 69 - 51	
X_H_std = 182 – 167 / 182 - 167	

Weight	Height	Class
0	0	Underweight
0.61	1	Normal
1	0.6	Normal
0.72	0.4	Normal
0.77	0.33	Normal
0.27	0.47	Underweight
0.39	0.13	Normal
0.33	0.4	Normal
0.22	0.2	Normal

Find the class for person who got weight = 57 and height=170

Find the class for person who got weight = 0.33 and height=0.2

We solve KNN using Euclidean Distance : sqrt($(OV_1 - AV_1)^2 + (OV_2 - AV_2)^2$)

1	$sqrt((0.33-0.00)^2 + (0.2-0)^2)$		
2	$sqrt((0.33-0.61)^2 + (0.2-1)^2)$		
3	$sqrt((0.33-1)^2 + (0.2-0.6)^2)$		
4	$sqrt((0.33-0.72)^2 + (0.2-0.4)^2)$		
5	$sqrt((0.33-0.77)^2 + (0.2-0.33)^2)$		
6	$sqrt((0.33-0.27)^2 + (0.2-0.47)^2)$		
7	$sqrt((0.33-0.39)^2 + (0.2-0.13)^2)$		
8	$sqrt((0.33-0.33)^2 + (0.2-0.4)^2)$		
9	$sqrt((0.33-0.22)^2 + (0.2-0.2)^2)$		

WEIGHT PREDICTOR

PERSON	HEIGHT	AGE	WEIGHT
1	6	40	60
2	6.1	26	55
3	5.9	30	56
4	5.8	32	58
5	5.3	33	75
6	5.6	34	78
7	5.5	35	80
8	5.7	33	70
9	5.8	32	72

PERSON	HEIGHT	AGE	WEIGHT
1	0.88	1.00	60
2	1.00	0.00	55
3	0.75	0.29	56
4	0.63	0.43	58
5	0.00	0.50	75
6	0.38	0.57	78
7	0.25	0.64	80
8	0.50	0.50	70
9	0.63	0.43	72

X_std = X – min / max - min	
X_H_std = 6 – 5.3 / 6.1 – 5.3	
X_A_std = 40 – 26 / 40 - 26	
X_H_std = 6.1 – 5.3 / 6.1 – 5.3	
X_A_std = 26 – 26 / 40 - 26	

PERSON	HEIGHT	AGE	WEIGHT
1	0.88	1.00	60
2	1.00	0.00	55
3	0.75	0.29	56
4	0.63	0.43	58
5	0.00	0.50	75
6	0.38	0.57	78
7	0.25	0.64	80
8	0.50	0.50	70
9	0.63	0.43	72

Find the weight for the person having height of 5.8 and age 37

Find the weight for the person having height of 0.63 and age 0.79

We solve KNN using Euclidean Distance : sqrt($(OV_1 - AV_1)^2 + (OV_2 - AV_2)^2$)

1	sqrt($(0.63-0.88)^2 + (0.79-1)^2$)		
2	sqrt($(0.63-1)^2 + (0.79-0)^2$)		
3	sqrt($(0.63-0.75)^2 + (0.79-0.29)^2$)		
4	sqrt($(0.63-0.63)^2 + (0.79-0.43)^2$)		
5	sqrt($(0.63-0)^2 + (0.79-0.5)^2$)		
6	sqrt($(0.63-0.38)^2 + (0.79-0.57)^2$)		
7	sqrt($(0.63-0.25)^2 + (0.79-0.64)^2$)		
8	sqrt($(0.63-0.5)^2 + (0.79-0.5)^2$)		
9	sqrt($(0.63-0.63)^2 + (0.79-0.43)^2$)		

kNN Advantages:

- ❑ It is simple to implement.
- ❑ It is robust to the noisy training data
- ❑ It can be more effective if the training data is large.

kNN Disadvantages:

- ❑ Always needs to determine the value of K which may be complex some time.
- ❑ The computation cost is high because of calculating the distance between the data points for all the training samples.