KNN

K- Nearest Neighbours

(1) Proximity :

	x_1	x_2
Two observations:	$(a_{11}$	$a_{21})$
	$(a_{12}$	$a_{22})$

Euclidean distance (in vectorized notation)

Let $\underset{\sim}{x_1} = (x_{11}, x_{21})^T$ and $\underset{\sim}{x_2} = (x_{12}, x_{22})^T$

be two vectors.

Euclidian distance (in general)

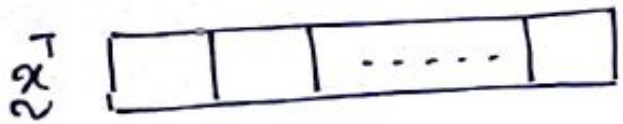
	X_1	X_2	.....	X_p
$\underset{\sim}{x}_1^T$	x_{11}	x_{21}	---	x_{p1}
$\underset{\sim}{x}_2^T$	x_{12}	x_{22}	---	x_{p2}
.....				

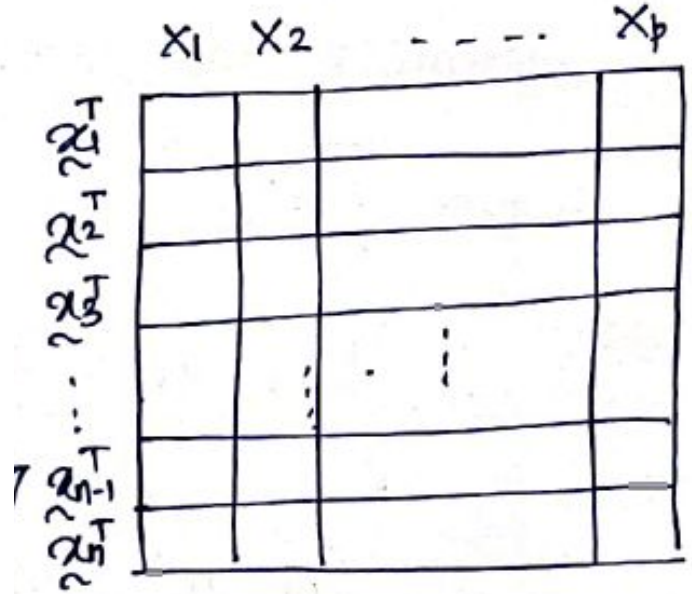
The euclidean distance between $\underset{\sim}{x}_1$ and $\underset{\sim}{x}_2$ can be written as:

$$\text{dist} = \sqrt{(x_{11}-x_{12})^2 + (x_{21}-x_{22})^2 + \cdots + (x_{p1}-x_{p2})^2}$$

$$= \sqrt{(\underset{\sim}{x}_1 - \underset{\sim}{x}_2)^T (\underset{\sim}{x}_1 - \underset{\sim}{x}_2)}$$

(2) finding the nearest neighbour :





1-NN

Step 1: calculate the dist. of $\underset{\sim}{x}$ from $\underset{\sim}{x_i}$, for all $i = 1(1)n$

Step 2: NN = the obs. that corresponds to the shortest distance.

Simplest approach (1-NN):

```
1: min_dist = inf
2: for i in range(1, len(train)):
3:     d = dist. of the test obs. from the ith obs. in train-data
4:     if d < min_dist:
5:         min_dist = d    #updati min_dist
6:         nearest_obs_index = i
7:     end if
8: end for
9: nearest_obs = train[i, ]
```

From 1 to len(train)

(3) 1-NN algorithm for classification :

Training data :

	X_1	X_2	---	X_p	Y
$\underset{\sim}{x}_1^T$	x_{11}	x_{21}	---	x_{p1}	y_1
$\underset{\sim}{x}_2^T$	x_{12}	x_{22}	---	x_{p2}	y_2
⋮	⋮	⋮		⋮	⋮
$\underset{\sim}{x}_{n_1}^T$	x_{1n_1}	x_{2n_1}	---	x_{pn_2}	y_n

Test data :

	X_1	X_2	---	X_p
$\underset{\sim}{x}_1^{*T}$	x_{11}^*	x_{21}	---	x_{p1}^*
$\underset{\sim}{x}_2^{*T}$	x_{12}^*	x_{22}	---	x_{p2}^*
⋮	⋮	⋮		⋮
$\underset{\sim}{x}_{n_2}^{*T}$	$x_{1n_2}^*$	x_{2n_2}	---	$x_{pn_2}^*$

n_1 = no. of obs. in training data

n_2 = no. of obs. in test data

- function 1: function to calculate the eucledian distance.

  distance (x, y) → returns euclidean dist. between the vectors $\underset{\sim}{x}$ and $\underset{\sim}{y}$.

- function 2: function to calculate the (or get the) NN for one test observation.

  NN (train_X, train_y, x^*) → finds the nearest obs. to x^* in the training data and get the value of Y for that obs.

□ function 3':

1NN (train_X, test, train y) → returns the values of y for all the obs. in the test data corresp. to the nearest neighbors in the train data.