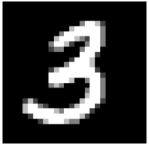
k-nearest neighbors (kNN)

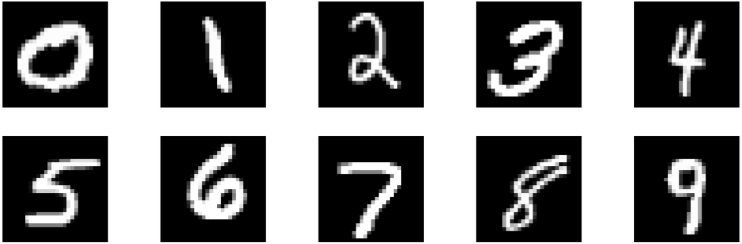
The problem we'll solve today

Given a 28x28 image, guess which digit it is

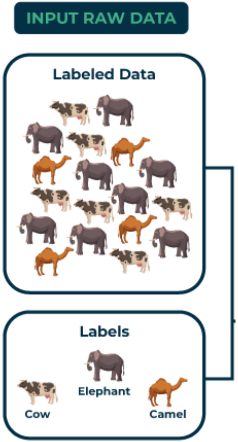


$\Longrightarrow$ 3

Some more examples:



Supervised learning



INPUT RAW DATA

Labeled Data

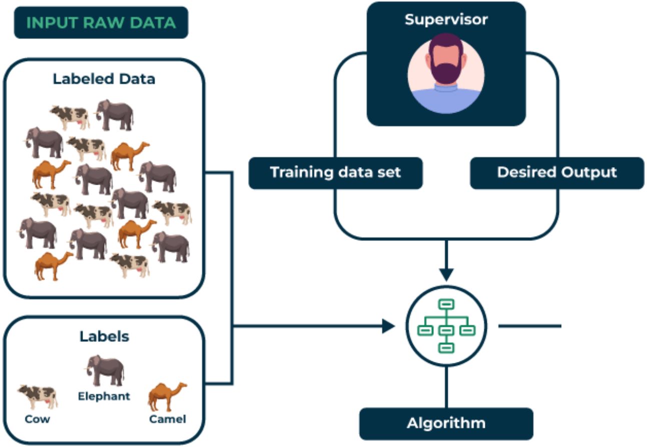
Labels

Cow

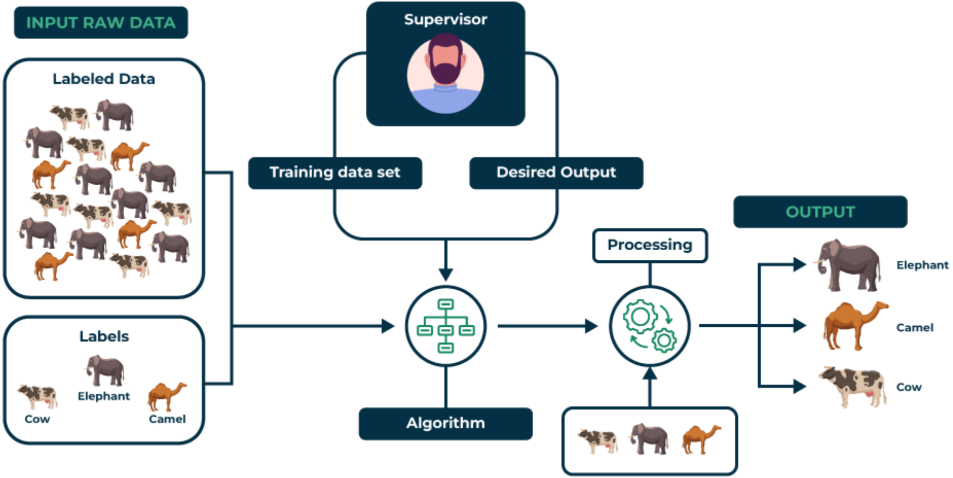
Elephant

Camel

Supervised learning



Supervised learning

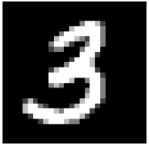


Common supervised learning models

- Logistic regression
- k-nearest neighbor (k-NN) $\longleftarrow$ **Today**
- Naïve Bayes
- Decision tree
- Random forest
- Support vector machine (SVM)

Notations: Data point

$x^{(i)} \quad \Longrightarrow$



$y^{(i)} \quad \Longrightarrow \quad 3$

MNIST dataset

- Training set of 60,000 images and their labels
- Test set of 10,000 images and their labels

Nearest neighbor classification

- **Training** images $x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(60000)}$
- Labels $y^{(1)}, y^{(2)}, y^{(3)}, \ldots, y^{(60000)}$ are numbers from $0 - 9$

Nearest neighbor classification

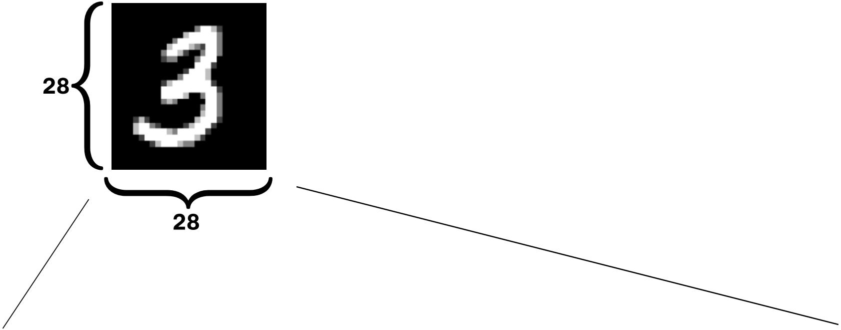
- **Training** images $x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(60000)}$
- Labels $y^{(1)}, y^{(2)}, y^{(3)}, \ldots, y^{(60000)}$ are numbers from $0-9$

How to classify a new image x?

- Find its nearest neighbor amongst the $x^{(i)}$
- Return $y^{(i)}$

Data as vectors

How to measure the distance between images?



0	0	0	0	0	0	0	0	·	·	·	·	1	1	1	1	0	0	1	1	1	·	·	·	0	0	0

Stretch each image into a vector with 784 coordinates

$$x^{(1)} = (0, 0, 0, \ldots, 0.6, 1, 1, 1, 0, 0, 1, 1, 0.8, \ldots, 0, 0, 0)$$

$$y^{(1)} = 6$$

The distance function

Euclidean distance in two dimensions is

Euclidean distance in higher dimension

Two images a and b:

$$a = (a_1, a_2, a_3, \ldots, a_{784})$$
$$b = (b_1, b_2, b_3, \ldots, b_{784})$$

The Euclidean distance between a and b is

$$\begin{aligned} \|a - b\|_2 &= \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_{784} - b_{784})^2} \\ &= \sqrt{\sum_{i=1}^{784} (a_i - b_i)^2} \end{aligned}$$

Nearest neighbor classification

Training images $x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(60000)}$ Labels $y^{(1)}, y^{(2)}, y^{(3)}, \ldots, y^{(60000)}$

Nearest neighbor classification

Training images $x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(60000)}$ Labels $y^{(1)}, y^{(2)}, y^{(3)}, \ldots, y^{(60000)}$

To classify a new image x

- Find its nearest neighbor **in Euclidean distance**, say $x^{(i)}$
- Return $y^{(i)}$

Accuracy of Nearest Neightbor on MNIST

Predictions on all points in the **Training set**

Question: What is the accuracy?

Accuracy of Nearest Neighbor on MNIST

Predictions on all points in the **Test set**

Question: What is the accuracy?

Test accuracy and Test error

Test set of 10,000 points

- 309 are misclassified

$$\text{Test accuracy} = \frac{\#\ \text{correct classification}}{\#\ \text{all points}}$$

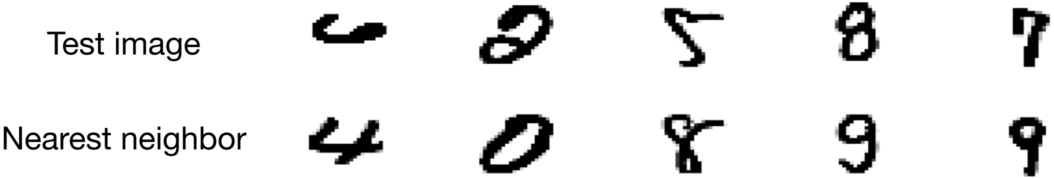
$$\text{Test error} = \frac{\#\ \text{incorrect classification}}{\#\ \text{all points}}$$

Examples of errors

Test set of 10,000 points

- 309 are misclassified

Examples of errors:



Test image

Nearest neighbor

How to improve?

k-nearest neighbor classification

To classify a new point:

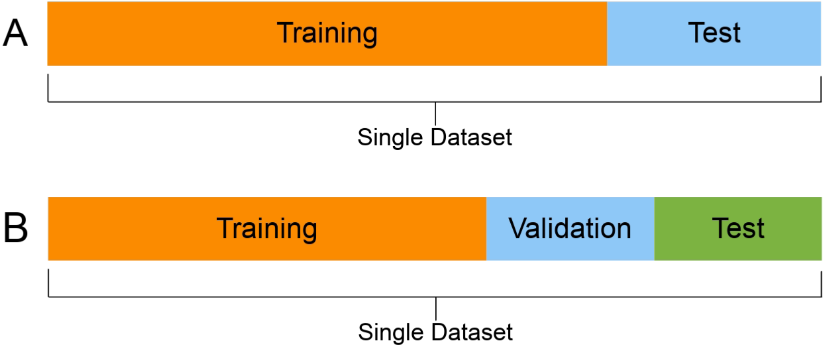
- Find the k **nearest neighbors** in the training set
- Return the most common label amongst them

MNIST:

k	1	3	5	7	9	11
Test error (%)	3.09	2.94	3.13	3.10	3.43	3.34

need to find k before final eval on the test set

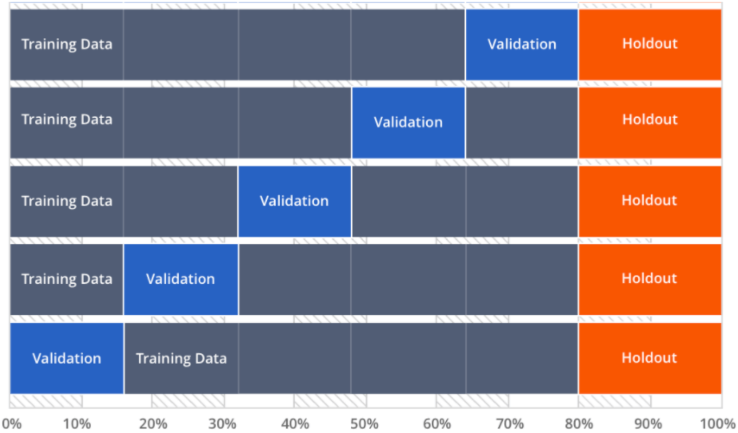
Validation



Train on Training set with $k = 1 \implies$ Evaluate on the Validation set
Train on Training set with $k = 3 \implies$ Evaluate on the Validation set
Train on Training set with $k = 5 \implies$ Evaluate on the Validation set

$\vdots$

Cross-validation



Other distance function

$$a = (a_1, a_2, \ldots, a_m) \qquad b = (b_1, b_2, \ldots, b_m).$$

- Cosine similarity

$$d_{\cos}(a, b) = \frac{a \cdot b}{\|a\|_2 \|b\|_2} = \frac{a}{\|a\|_2} \cdot \frac{b}{\|b\|_2}$$

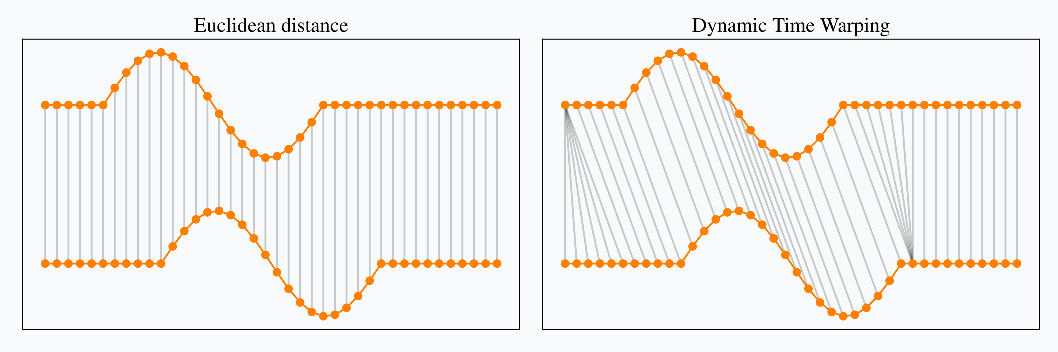
measures the angle between vector a and b.

$$-1 \leq d_{\cos}(a, b) \leq 1.$$

Examples

$$a = (1, 2, 2) \qquad b = (3, 4, 0)$$

Distance between time series



use **dynamic time warping**

k-NN regression

y is continuous $\quad x =$ test data

$(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ are k-nearest neighbors of x.

Prediction:

$$\hat{y} = \frac{1}{k} \sum_{i=1}^{k} y_i$$

Example of 5-NN

Training set:

$$\begin{array}{ccc} (2,1) & (5,3) & (10,6) \\ (4,9) & (1,3) & (8,2) \\ (5,8) & (7,8) & (1,4) \end{array}$$

New point: $x = 6$. The 5-NN prediction is