

# Ratings encoding

In the table below, each row represents a user’s ratings of movies: ✓ (check) indicates the person liked the movie, ✗ (x) that they didn’t, and • (dot) that they didn’t rate it one way or another (neutral rating or didn’t watch). Can encode these ratings numerically with 1 for ✓ (check), −1 for ✗ (x), and 0 for • (dot).

| Person | Fyre | Frozen II | Picard | Ratings written as a 3-tuple |
|--------|------|-----------|--------|------------------------------|
| $P_1$  | ✗    | •         | ✓      |                              |
| $P_2$  | ✓    | ✓         | ✗      |                              |
| $P_3$  | ✓    | ✓         | ✓      |                              |
| $P_4$  | •    | ✗         | ✓      |                              |

# Defining sets

*To define sets:*

To define a set using **roster method**, explicitly list its elements. That is, start with  $\{$  then list elements of the set separated by commas and close with  $\}$ .

To define a set using **set builder definition**, either form “The set of all  $x$  from the universe  $U$  such that  $x$  is ...” by writing

$$\{x \in U \mid \dots x \dots\}$$

or form “the collection of all outputs of some operation when the input ranges over the universe  $U$ ” by writing

$$\{\dots x \dots \mid x \in U\}$$

We use the symbol  $\in$  as “is an element of” to indicate membership in a set.

**Example sets:** For each of the following, identify whether it’s defined using the roster method or set builder notation and give an example element.

$$\{-1, 1\}$$

$$\{0, 0\}$$

$$\{-1, 0, 1\}$$

$$\{(x, x, x) \mid x \in \{-1, 0, 1\}\}$$

$$\{\}$$

$$\{x \in \mathbb{Z} \mid x \geq 0\}$$

$$\{x \in \mathbb{Z} \mid x > 0\}$$

$$\{\text{A, C, U, G}\}$$

$$\{\text{AUG, UAG, UGA, UAA}\}$$

## Defining functions ratings

Recall our representation of Netflix users' ratings of movies as  $n$ -tuples, where  $n$  is the number of movies in the database. Each component of the  $n$ -tuple is  $-1$  (didn't like the movie),  $0$  (neutral rating or didn't watch the movie), or  $1$  (liked the movie).

Consider the ratings  $P_1 = (-1, 0, 1)$ ,  $P_2 = (1, 1, -1)$ ,  $P_3 = (1, 1, 1)$ ,  $P_4 = (0, -1, 1)$

Which of  $P_1$ ,  $P_2$ ,  $P_3$  has movie preferences most similar to  $P_4$ ?

One approach to answer this question: use **functions** to define distance between user preferences.

For example, consider the function  $d_0$  :  
given by

→

$$d_0( (x_1, x_2, x_3), (y_1, y_2, y_3) ) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$

*Extra example:* A new movie is released, and  $P_1$  and  $P_2$  watch it before  $P_3$ , and give it ratings;  $P_1$  gives ✓ and  $P_2$  gives ✗. Should this movie be recommended to  $P_3$ ? Why or why not?

*Extra example:* Define a new function that could be used to compare the 4-tuples of ratings encoding movie preferences now that there are four movies in the database.