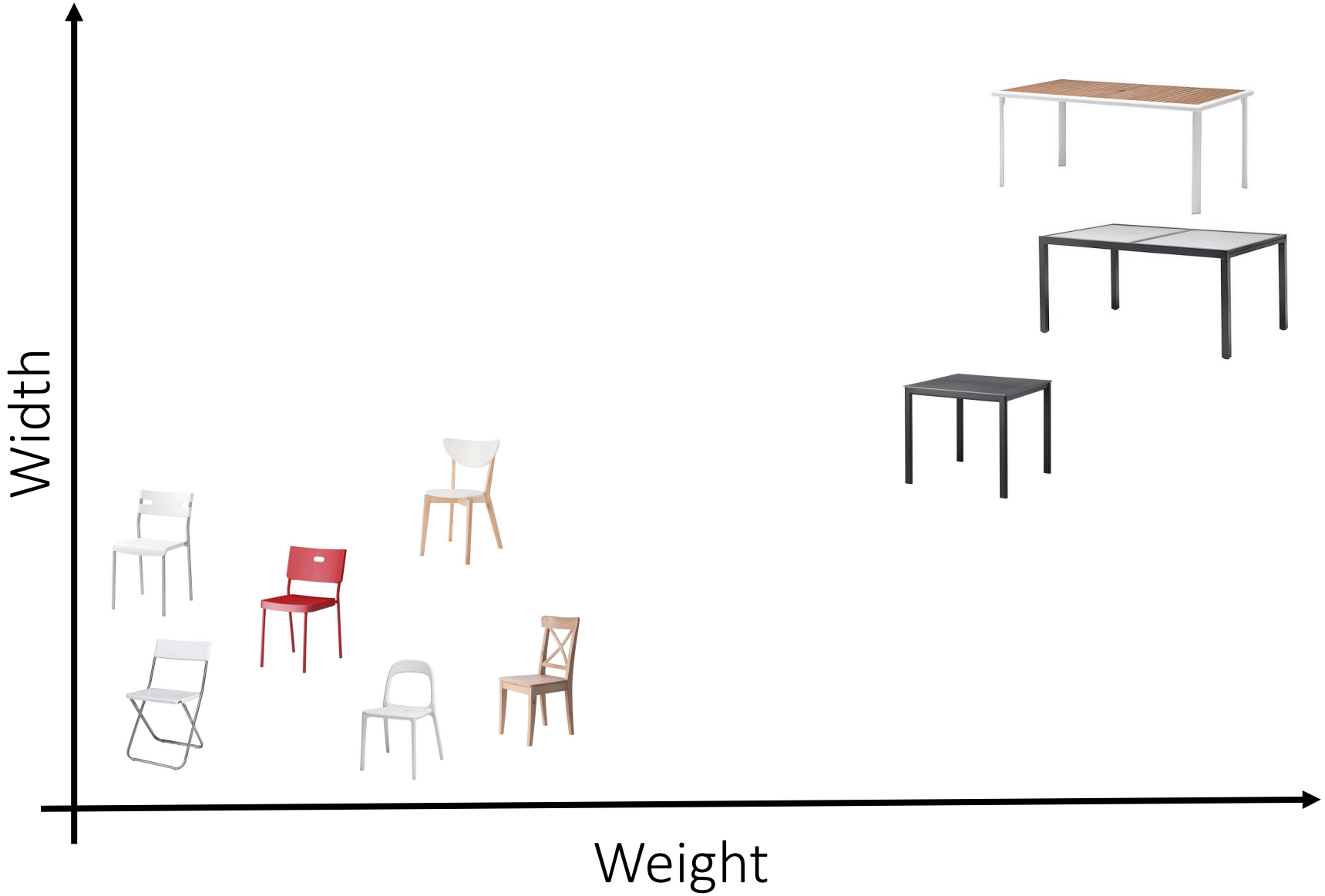
# Predicting the purpose of a piece of IKEA furniture

(Or: classification)

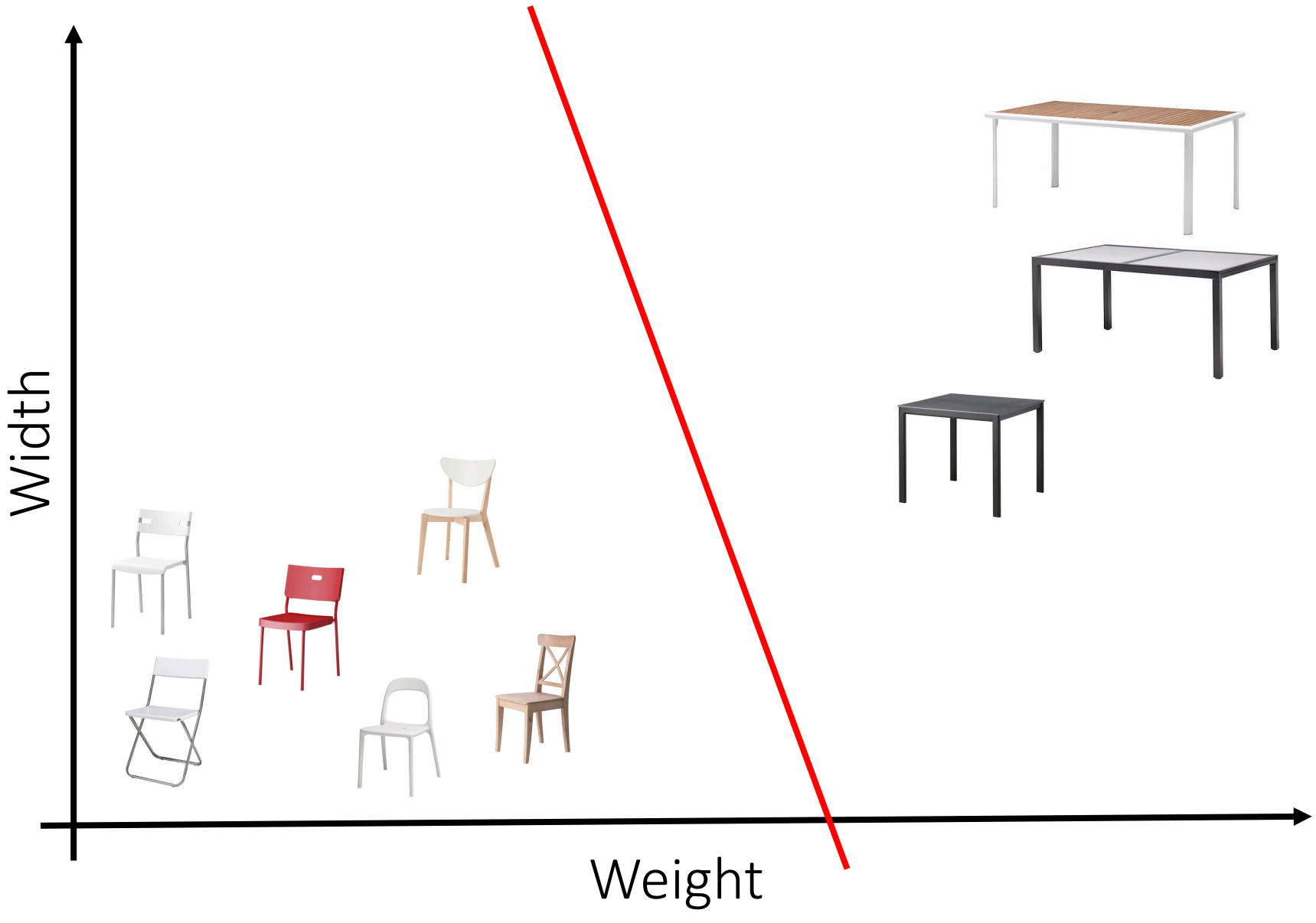
First thing we need:

# A model

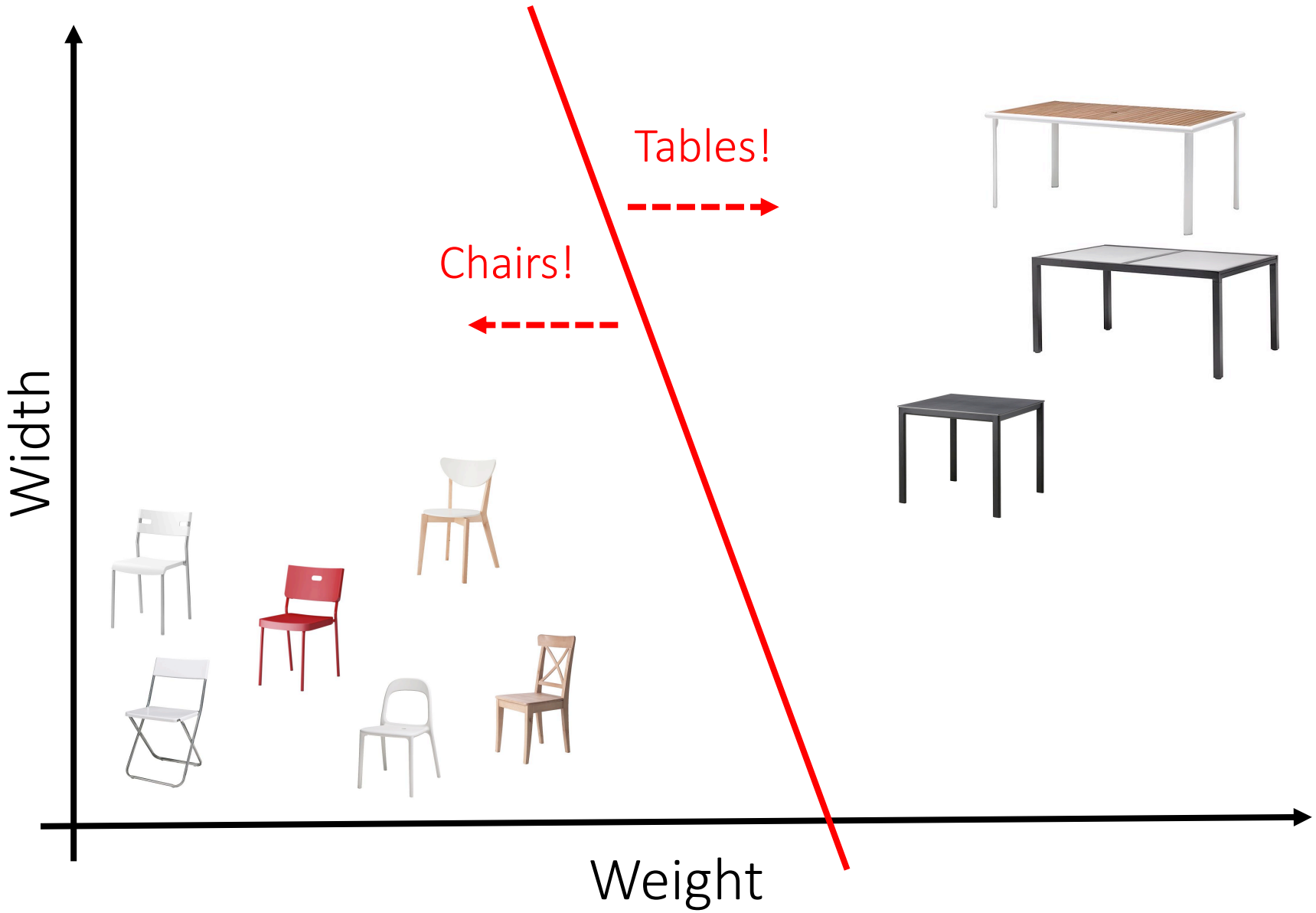
# Classification on features



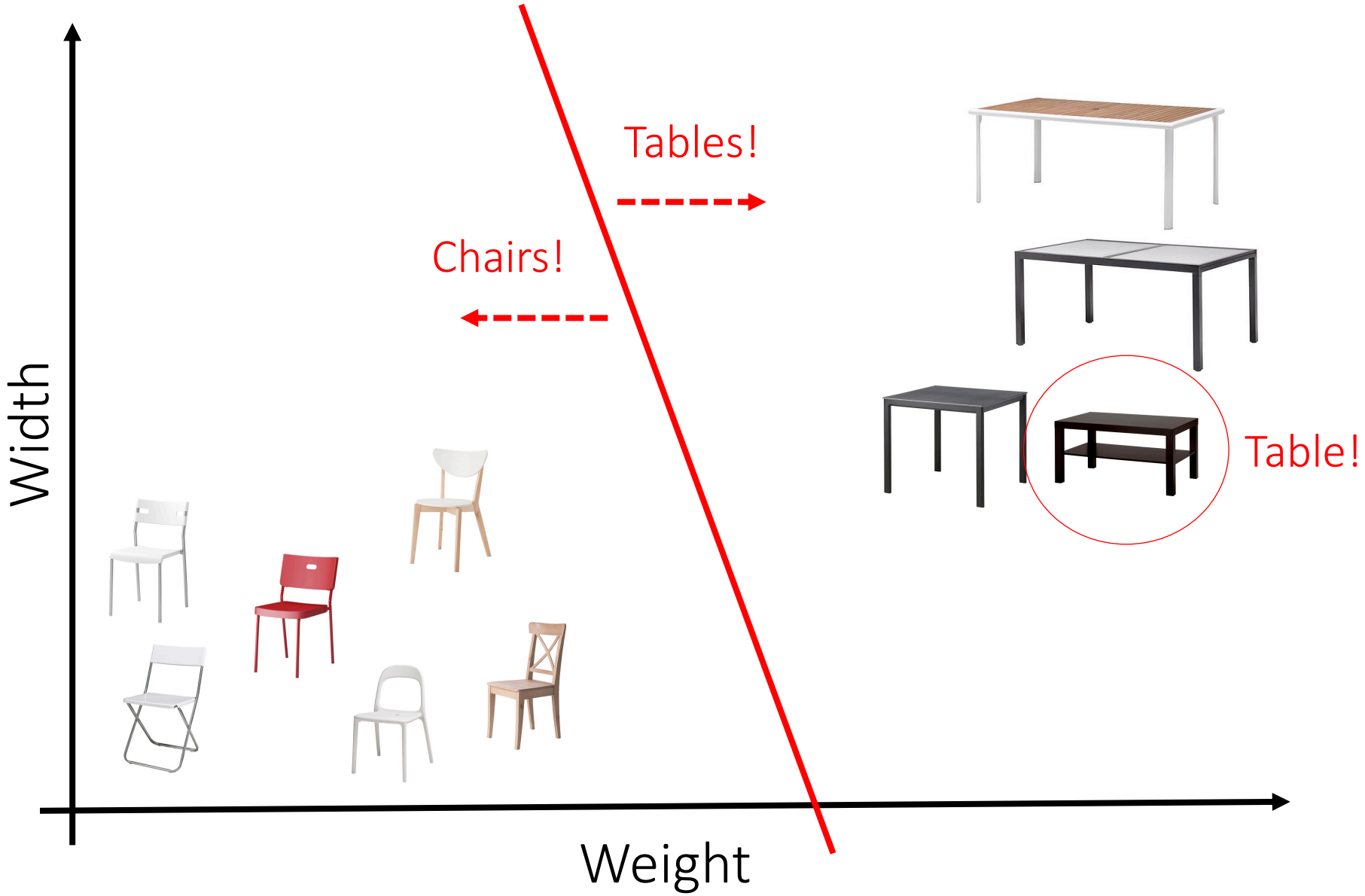
# Classification on features



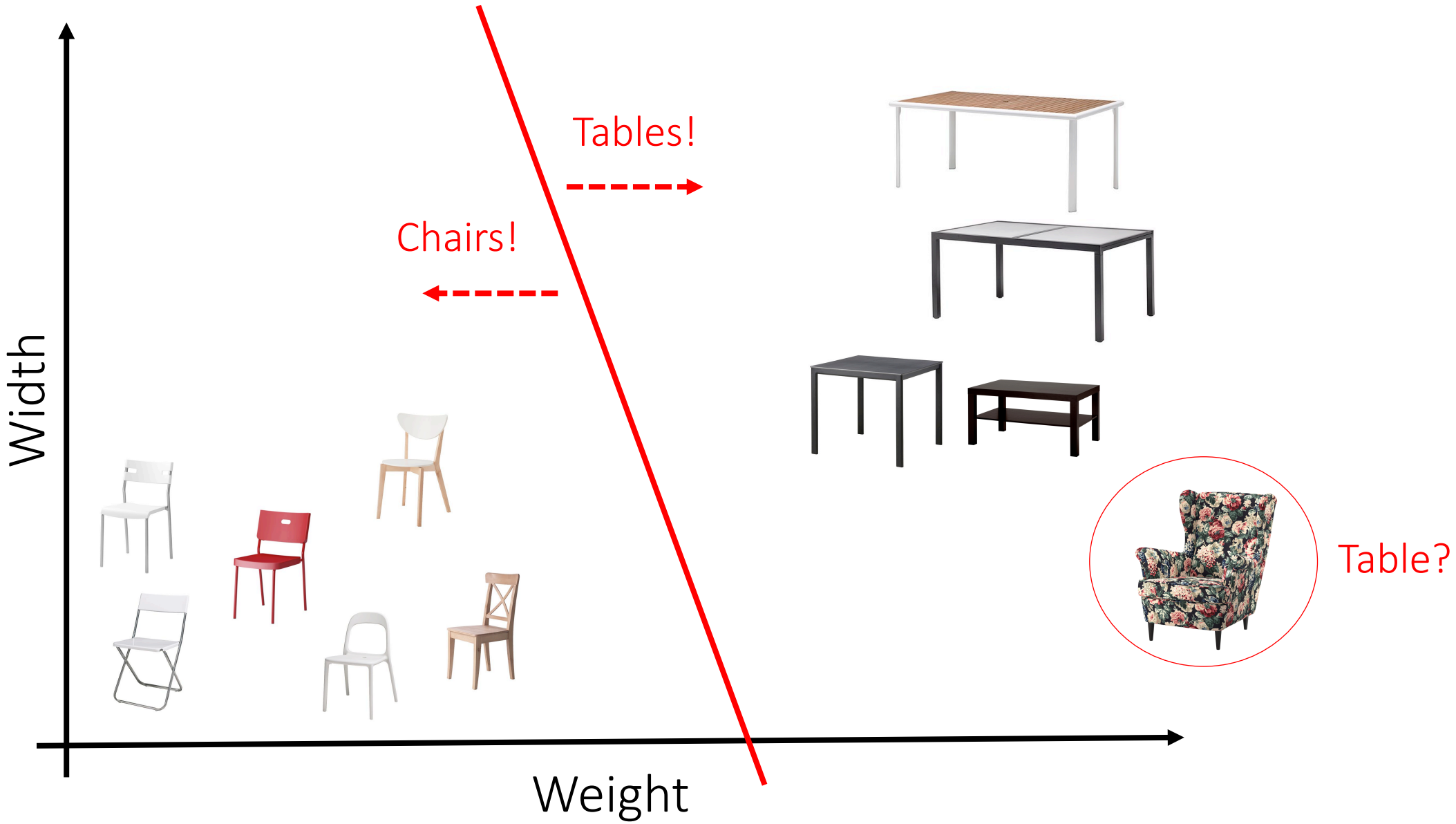
# Classification on features



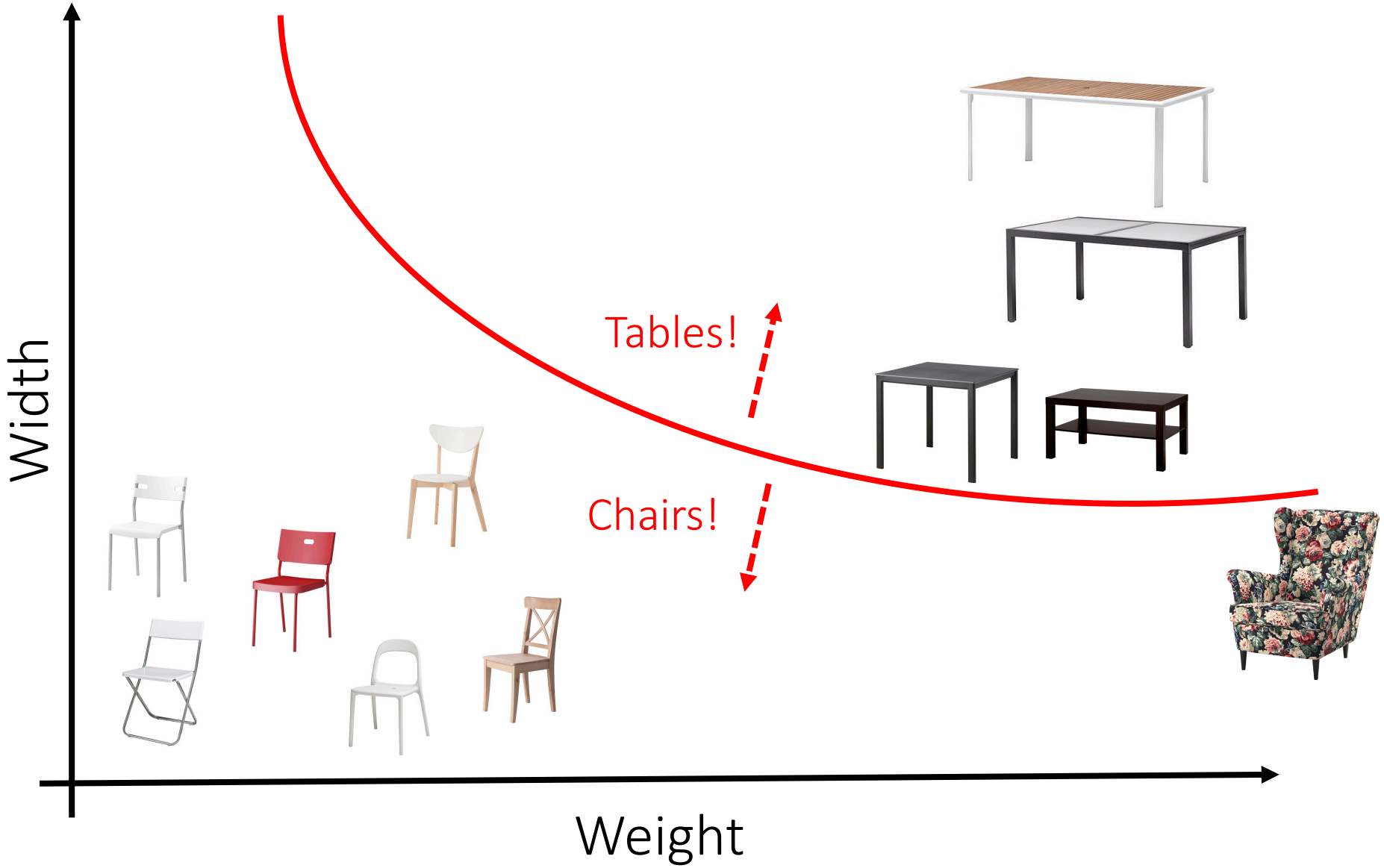
# Classification on features



# Classification on features



# Classification on features



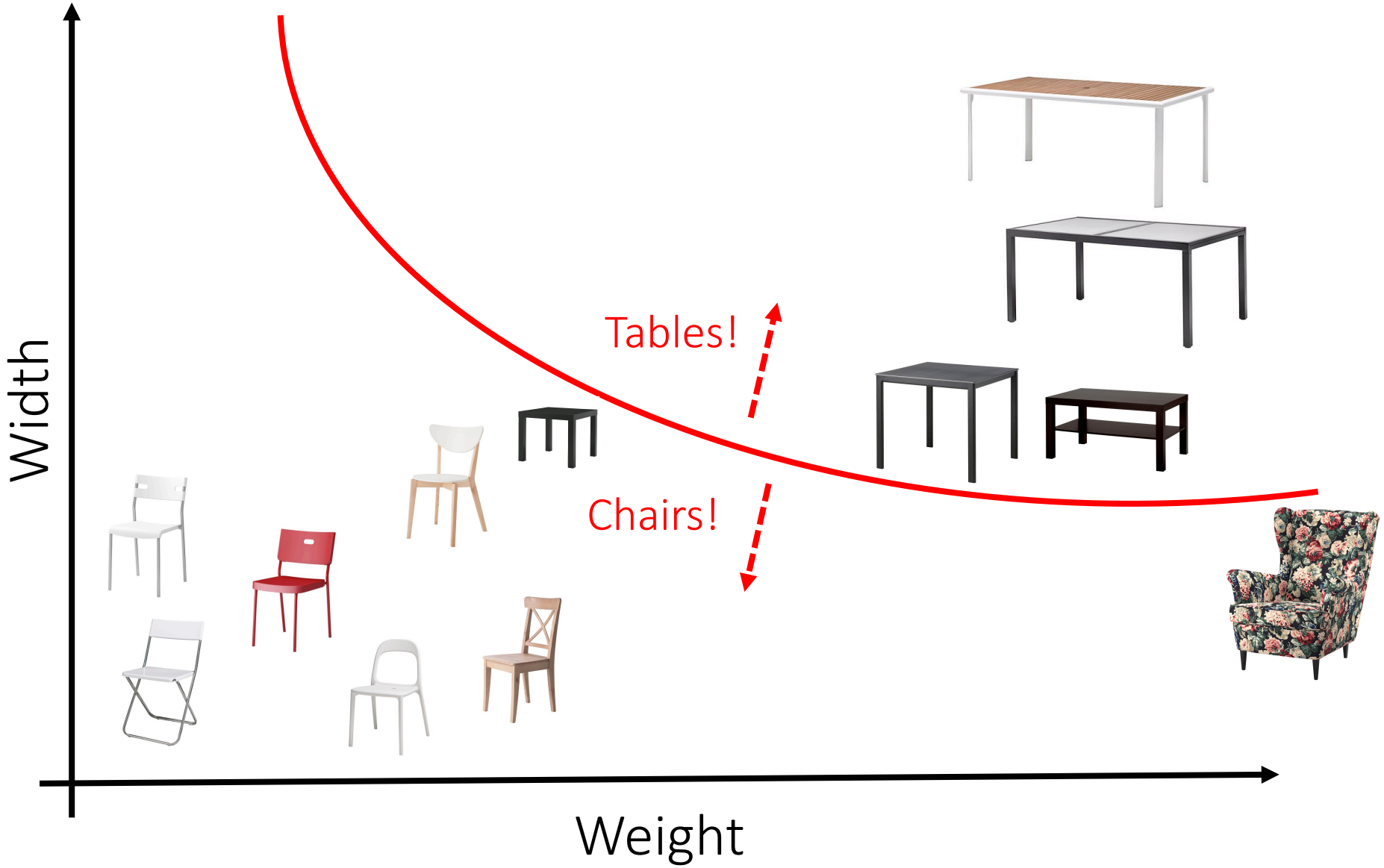
# *Now* we have the perfect model!

Right?

# Sure!

...except the model will break the next time
a slightly unusual piece of furniture comes along

# Classification on features



# Clustering on features

# Clustering on features

# Price + availability in local warehouse seem to be the perfect features!

Right?

# Sure!

…except the model will break when the price or availability changes,
which it probably will

# Feature selection is important!

To make a good model, we need:

- Enough observations to make sure that the model is generalizable
- Features that are consistently observed with a given piece of furniture

# Patient stratification

(Or: grouping patients responding similarly to a treatment)

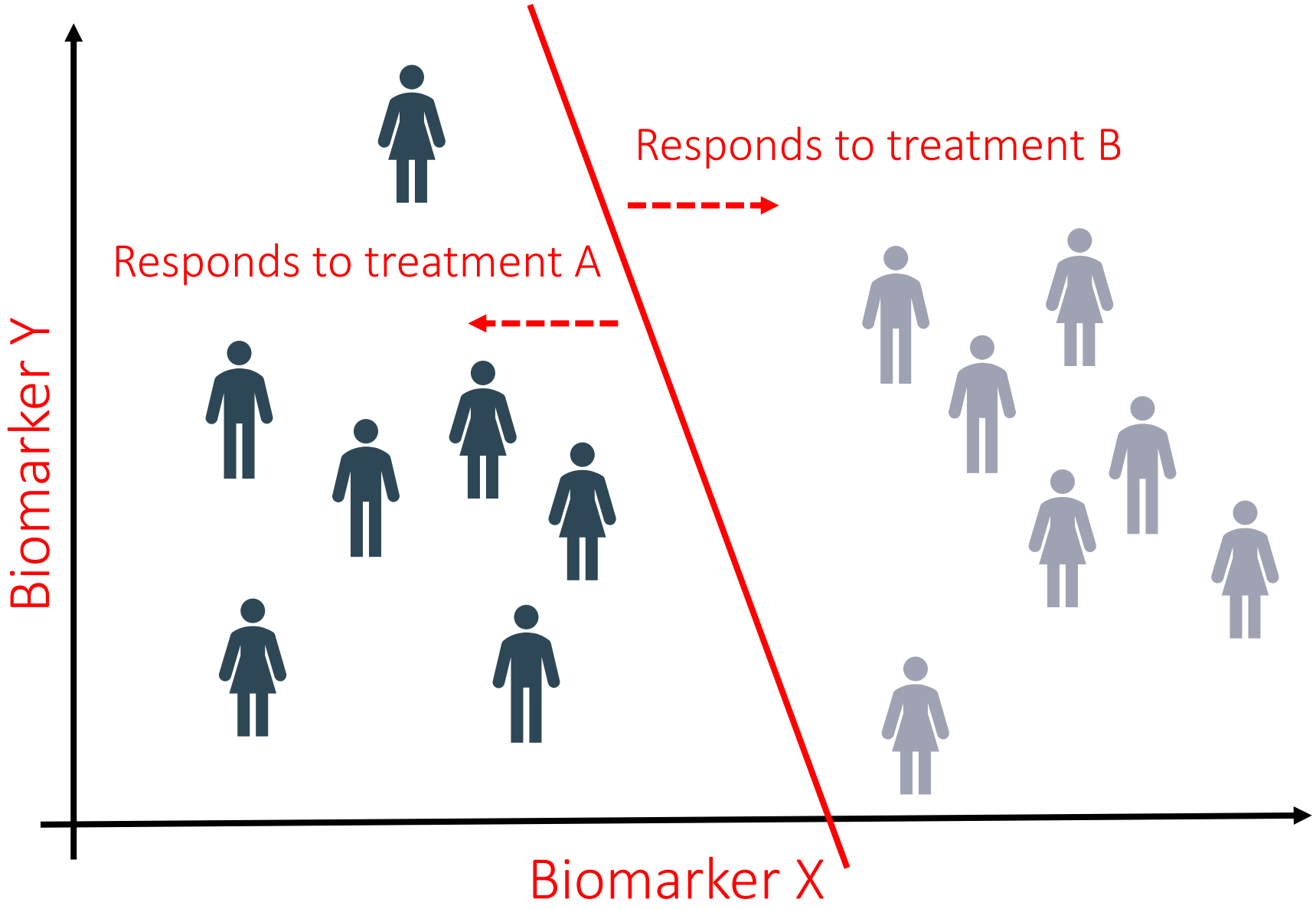
# Patient stratification

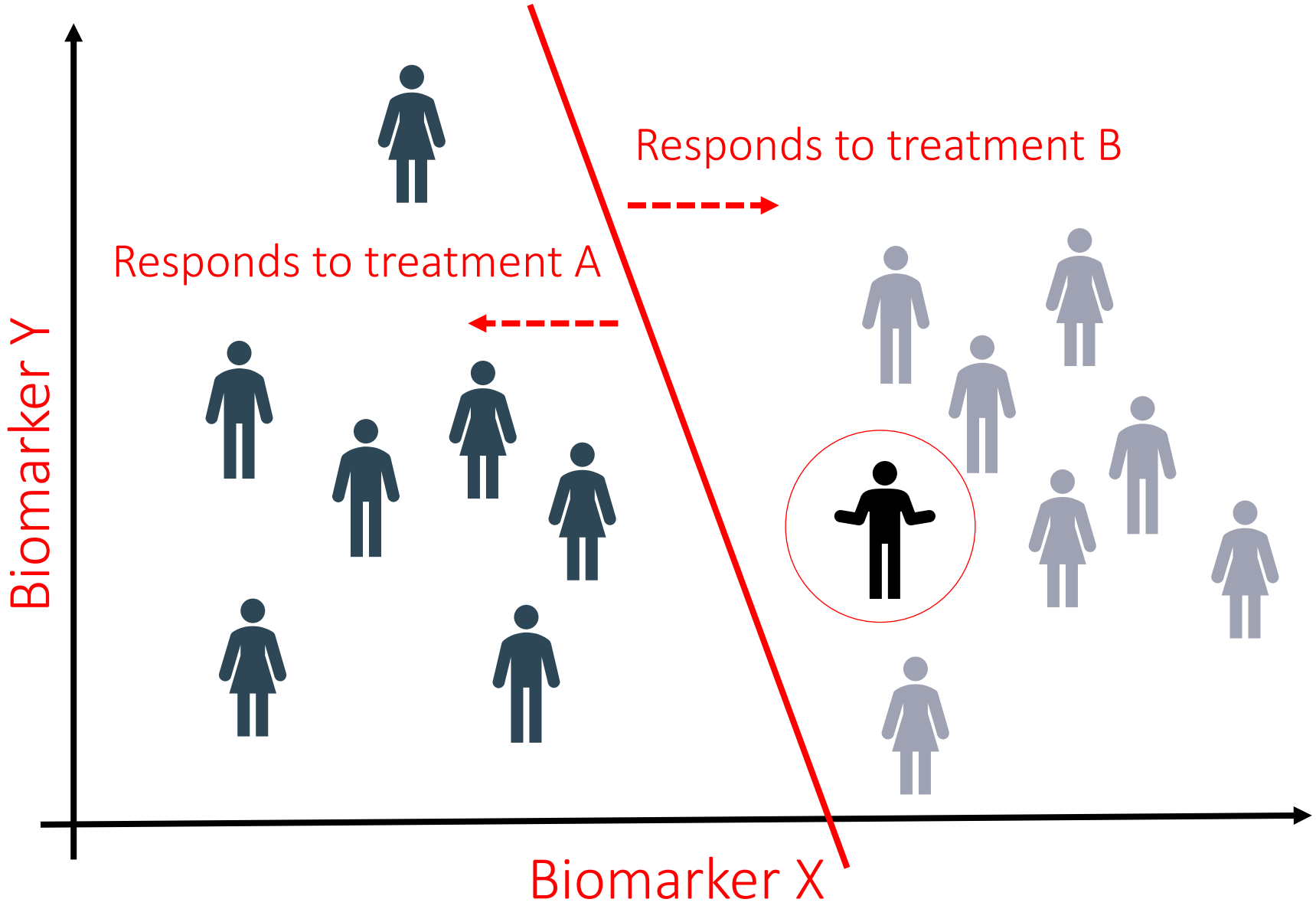
(Or: grouping patients responding similarly to a treatment)

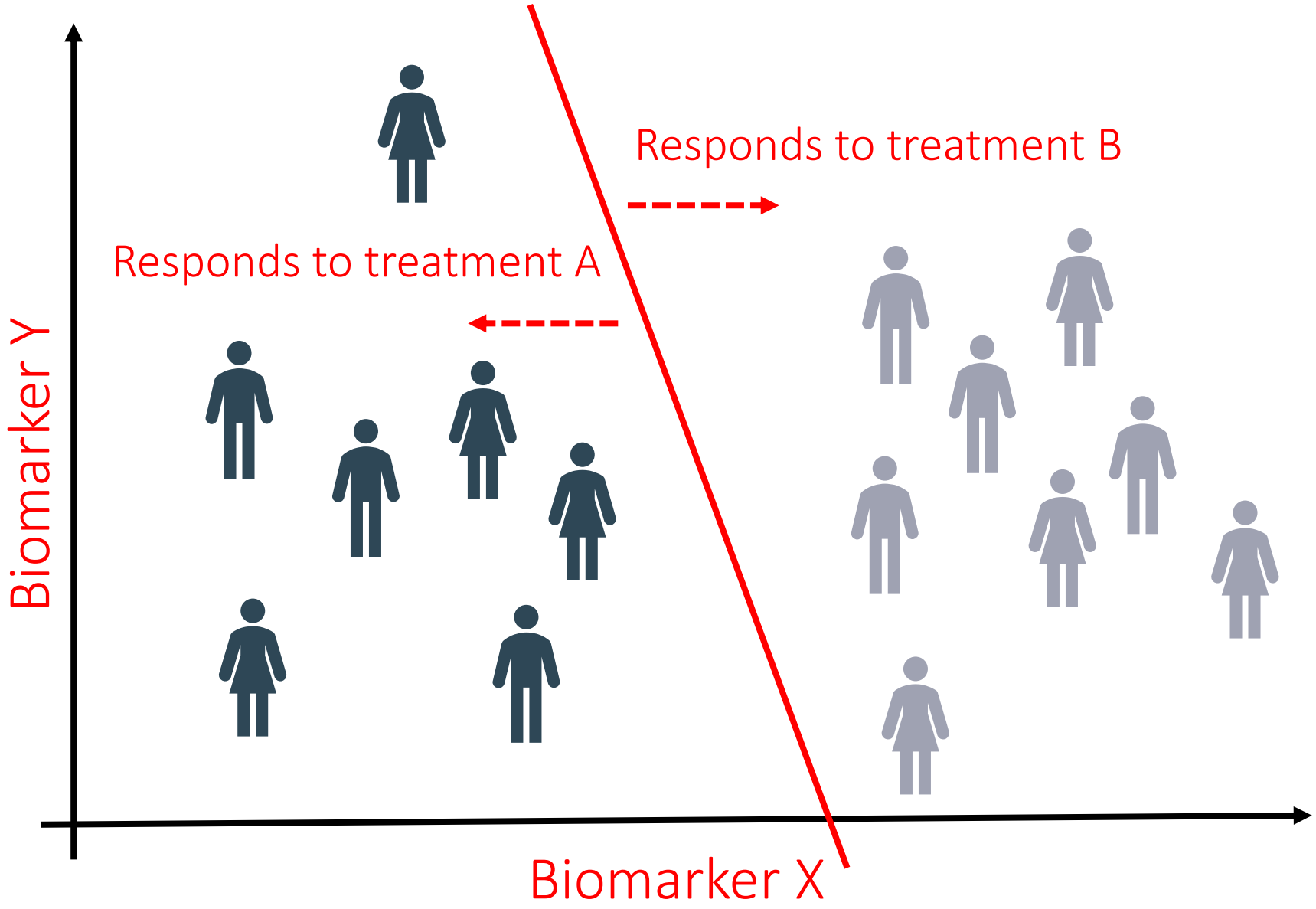
(Or: classification to clusters)

# For patient stratification we need:

1. A discovery cohort
2. Some biomarkers
3. A model (clustering)
4. A classification algorithm


Responds to treatment B
Responds to treatment A
Biomarker Y
Biomarker X


Responds to treatment B
Responds to treatment A
Biomarker Y
Biomarker X


Responds to treatment B
Responds to treatment A
Biomarker Y
Biomarker X

# Classification algorithms

# Algorithms we will cover in this course

- k nearest neighbor
- Distance to centroid

- There are many, many more...

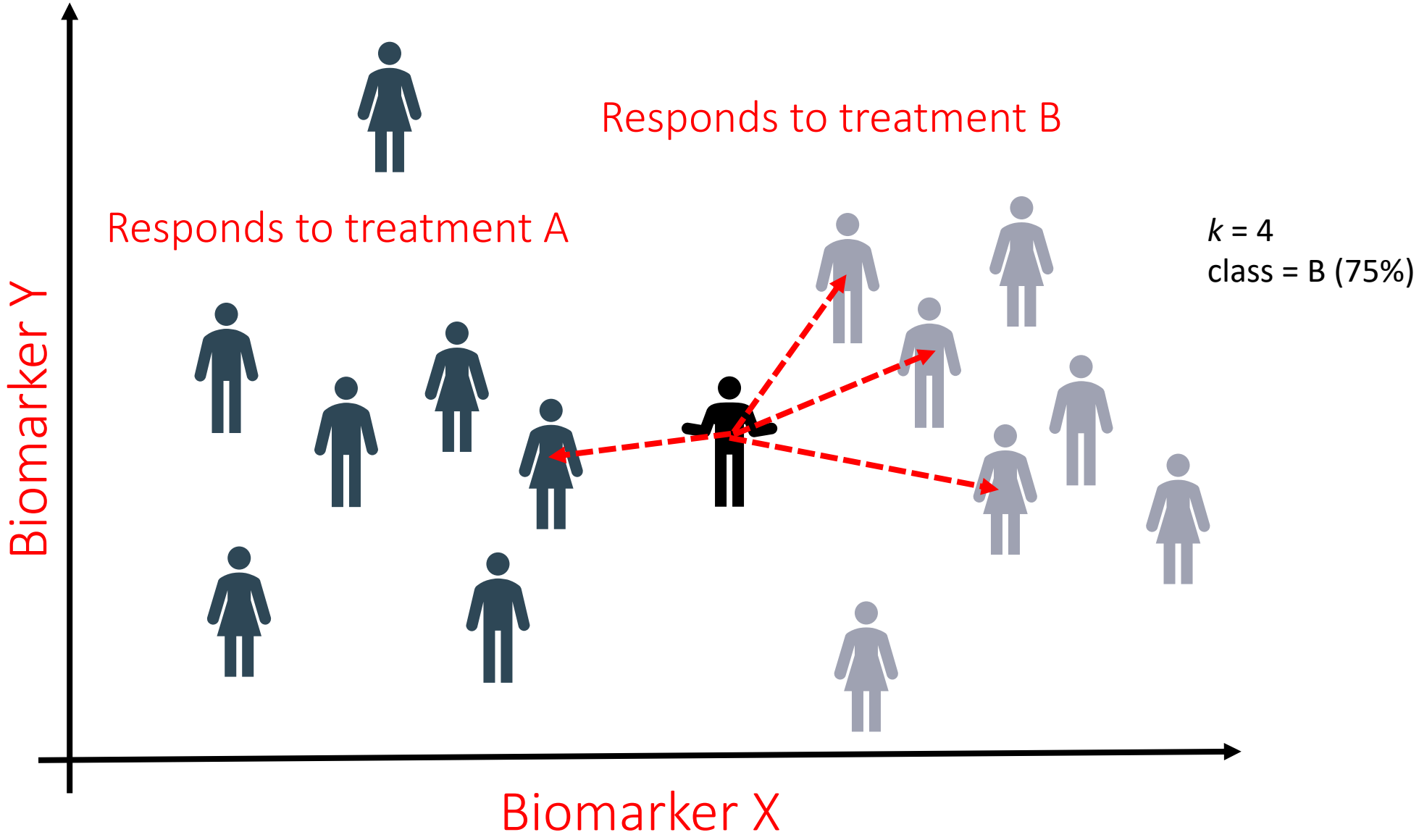
# *k* nearest neighbor

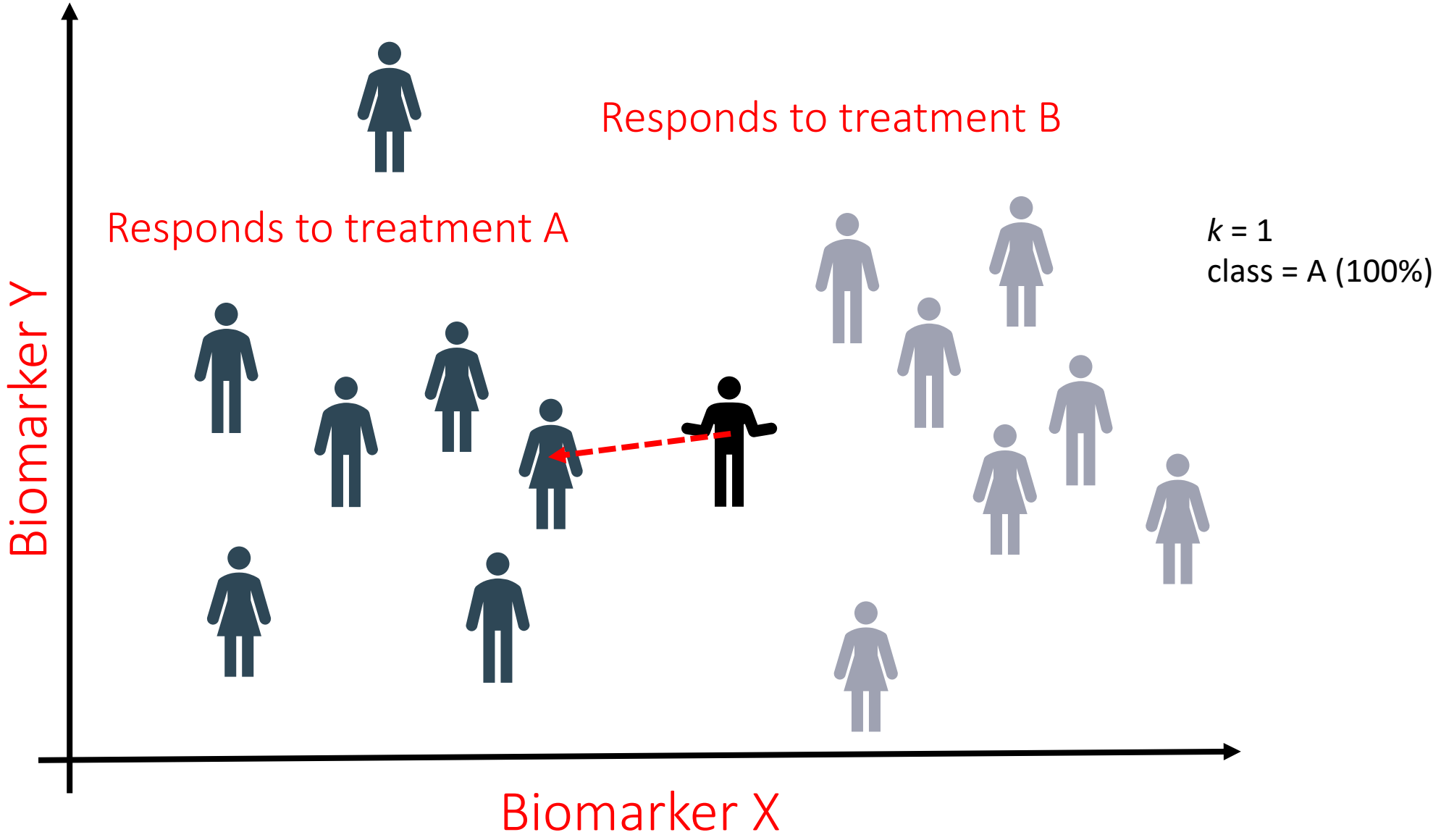
In this algorithm an observation is classified as belonging to a class, based on the class of *k* neighbors:

1. Calculate the distance between all samples (both the training and query samples)

2. For each query sample, find the *k* closest neighbors

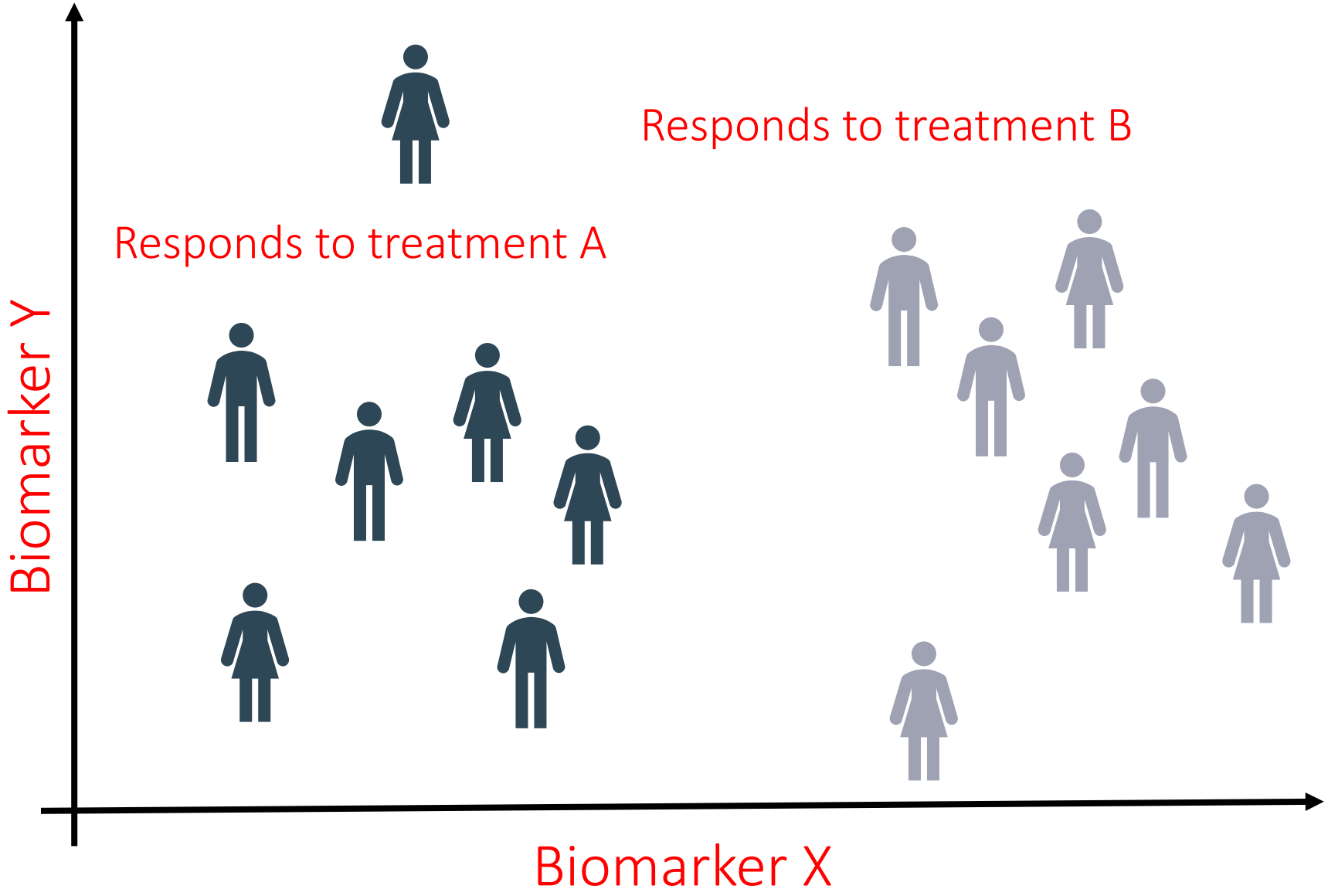
3. Class is then assigned as “winner-takes-all” or in a probabilistic fashion


Responds to treatment B
Responds to treatment A
k = 4
class = B (75%)
Biomarker Y
Biomarker X


Responds to treatment B
Responds to treatment A
k = 1
class = A (100%)
Biomarker Y
Biomarker X

# So which *k* is correct?

We estimate this using cross-validation on training cohort.


Responds to treatment B
Responds to treatment A
Biomarker Y
Biomarker X

Many of the algorithms have multiple parameters that can be tweaked independently of each other.

Theoretically, all combinations of these parameter settings must be tested.

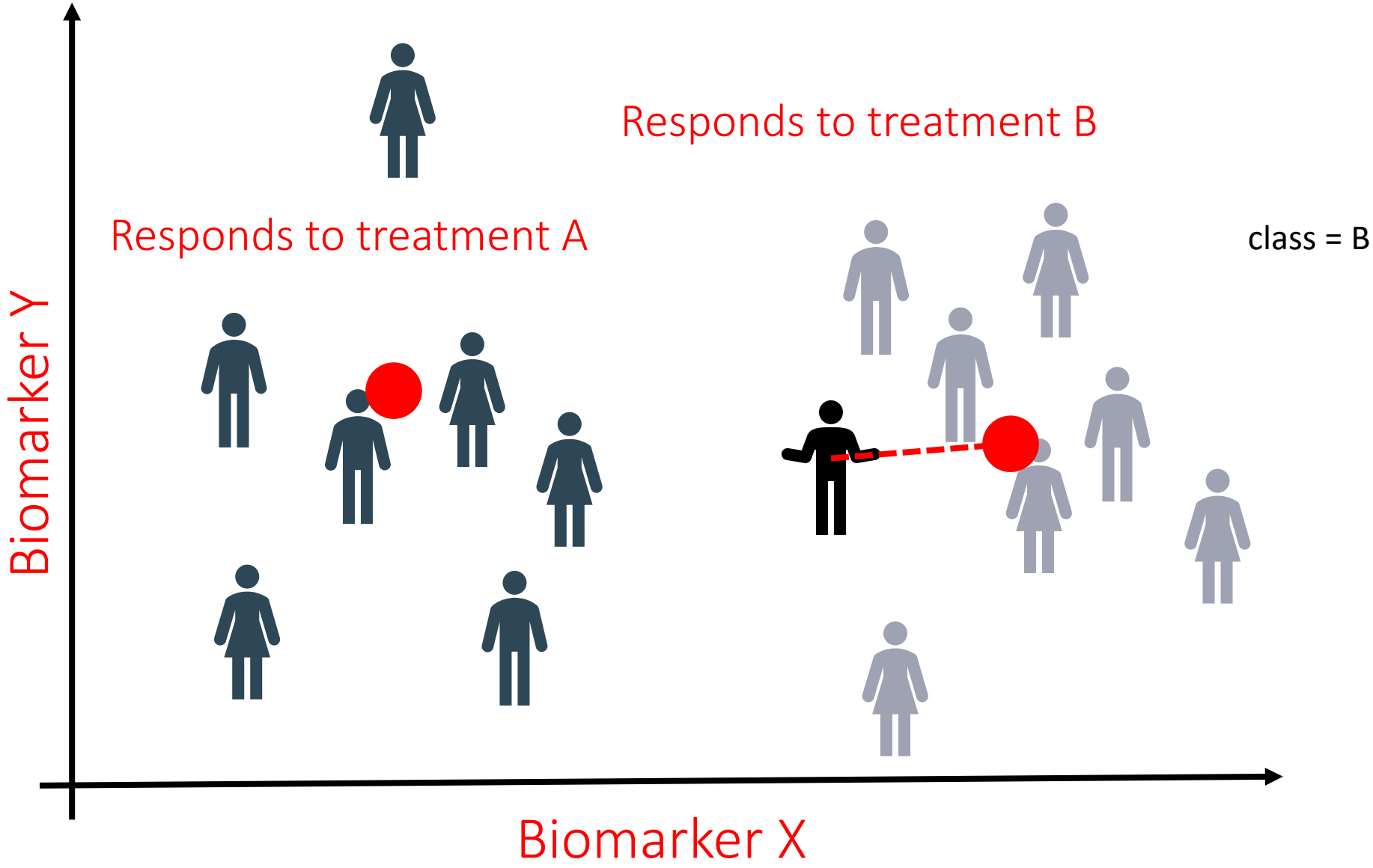
# Distance to centroid algorithm

In this algorithm class is assigned based on the closest class centroid.

1. The centroid for each class is calculated (typically the mean of all coordinates)

2. The distance from the query sample to each centroid is calculated

3. The sample is given the class of the closest centroid


Responds to treatment B
Responds to treatment A
class = B
Biomarker Y
Biomarker X

# Visualizing

- Everything shown today is two-dimensional, but gene expression data is often many thousand-dimensional.

- For these visualizations, we use principal component analysis.

# Exercise time!