Applying ML Algorithms on DStreams

Janani Ravi
CO-FOUNDER, LOONYCORN
www.loonycorn.com

Overview

Understand how the k-means clustering algorithm is used to find patterns

Understand the nuances of applying k-means clustering to streaming data

Implement the algorithm in Python on a real world dataset using

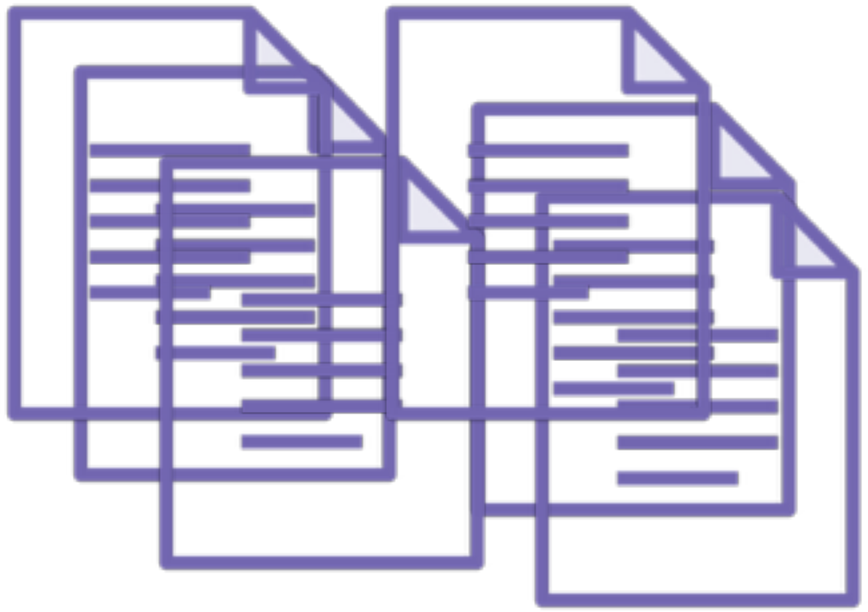
- Spark streaming
- MLLib

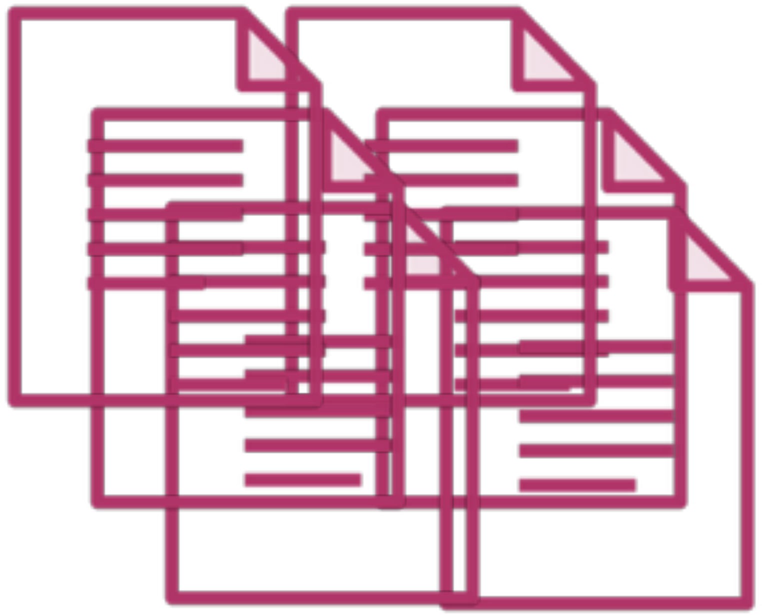
Patterns in Data

Patterns in Data

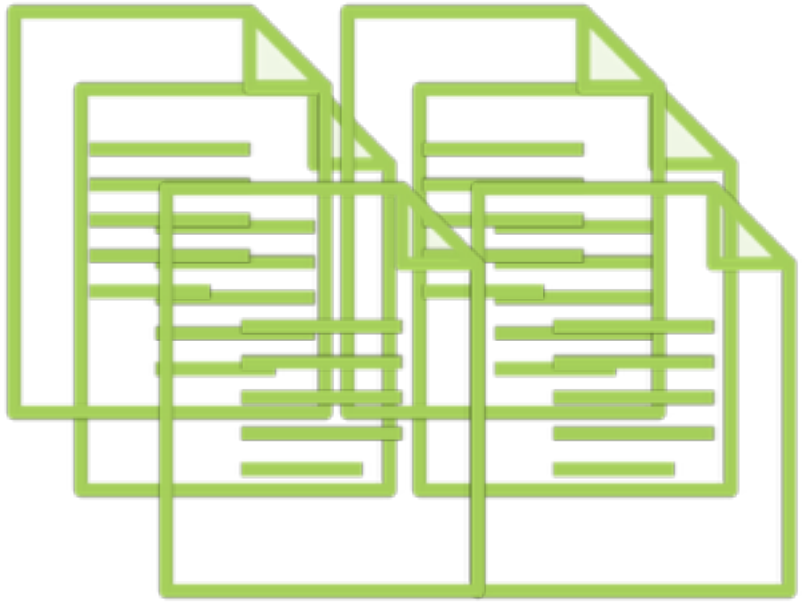
How do you make sense of this?

Patterns in Data



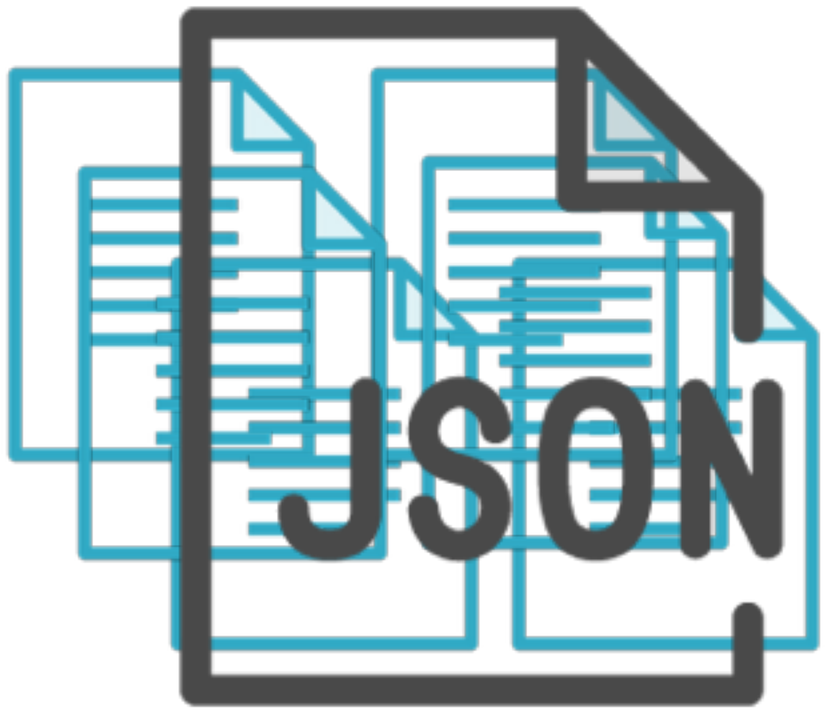


Group them based on some common attributes



Patterns in Data

Clustering



Clustering

Products sold on Amazon

People on Facebook

Websites indexed by Google

What if you want to group more complex entities?

Clustering

Products sold on Amazon

People on Facebook

Websites indexed by Google

Too many entities, too many attributes per entity

Huge complexity

Clustering

Anything can be represented by a set of numbers

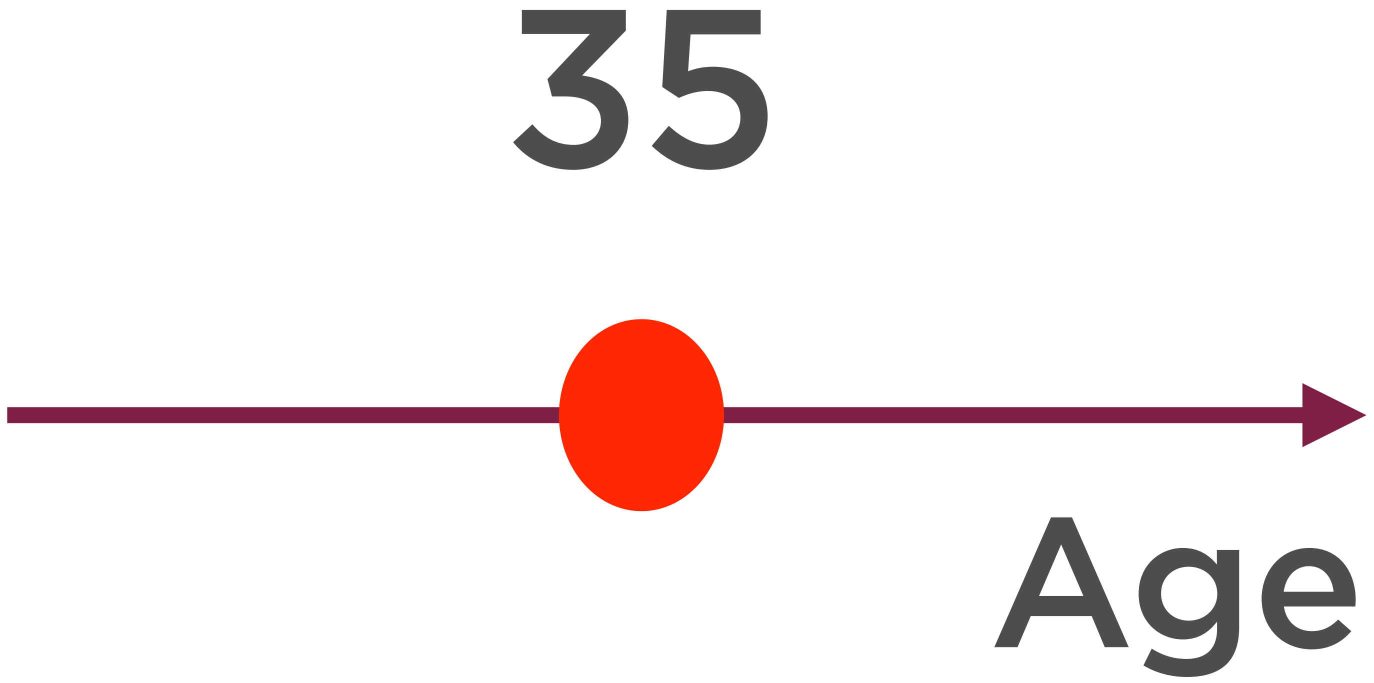
Clustering

Product ID,
Timestamp, Amount

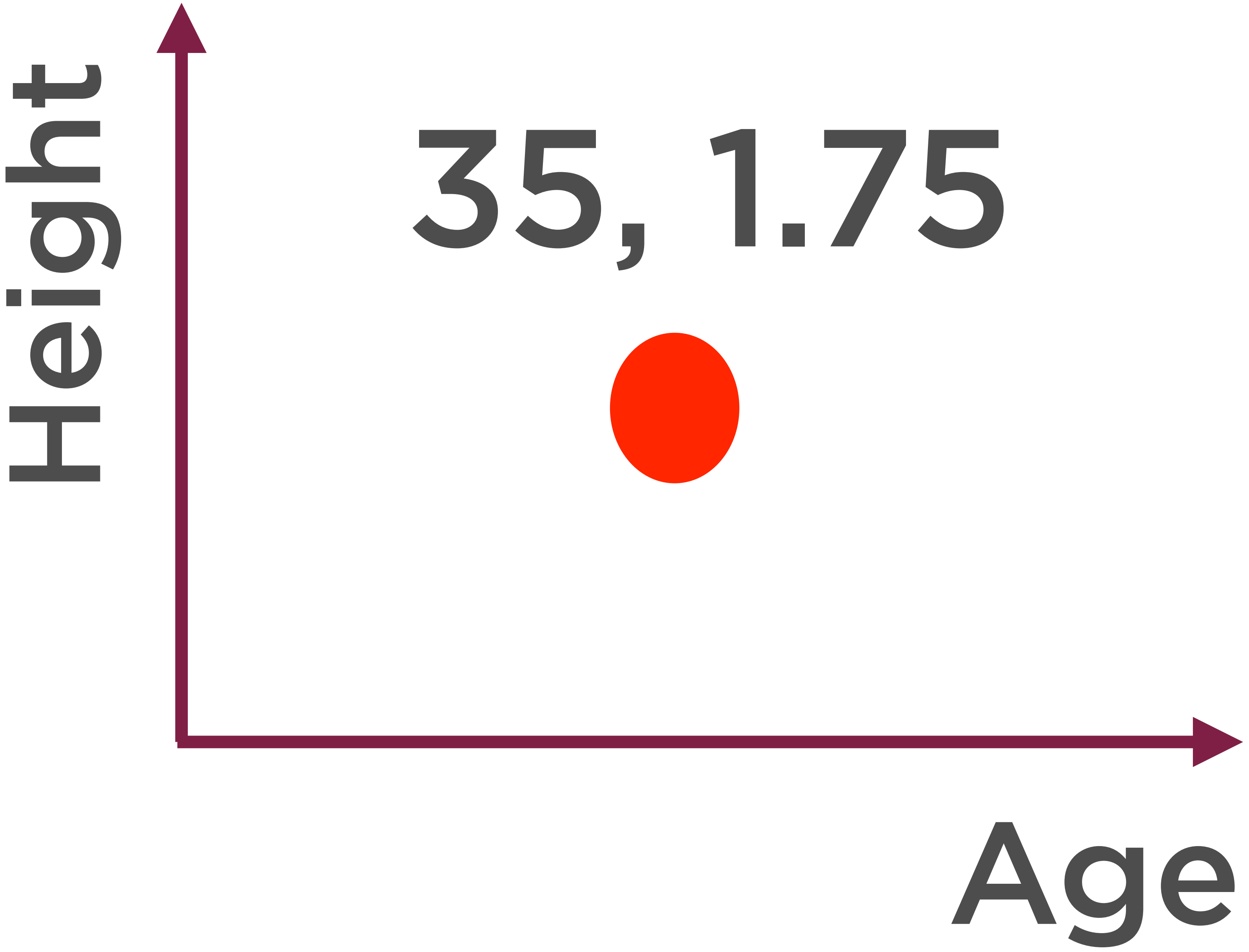
Age, Height, Weight

Length, word frequencies

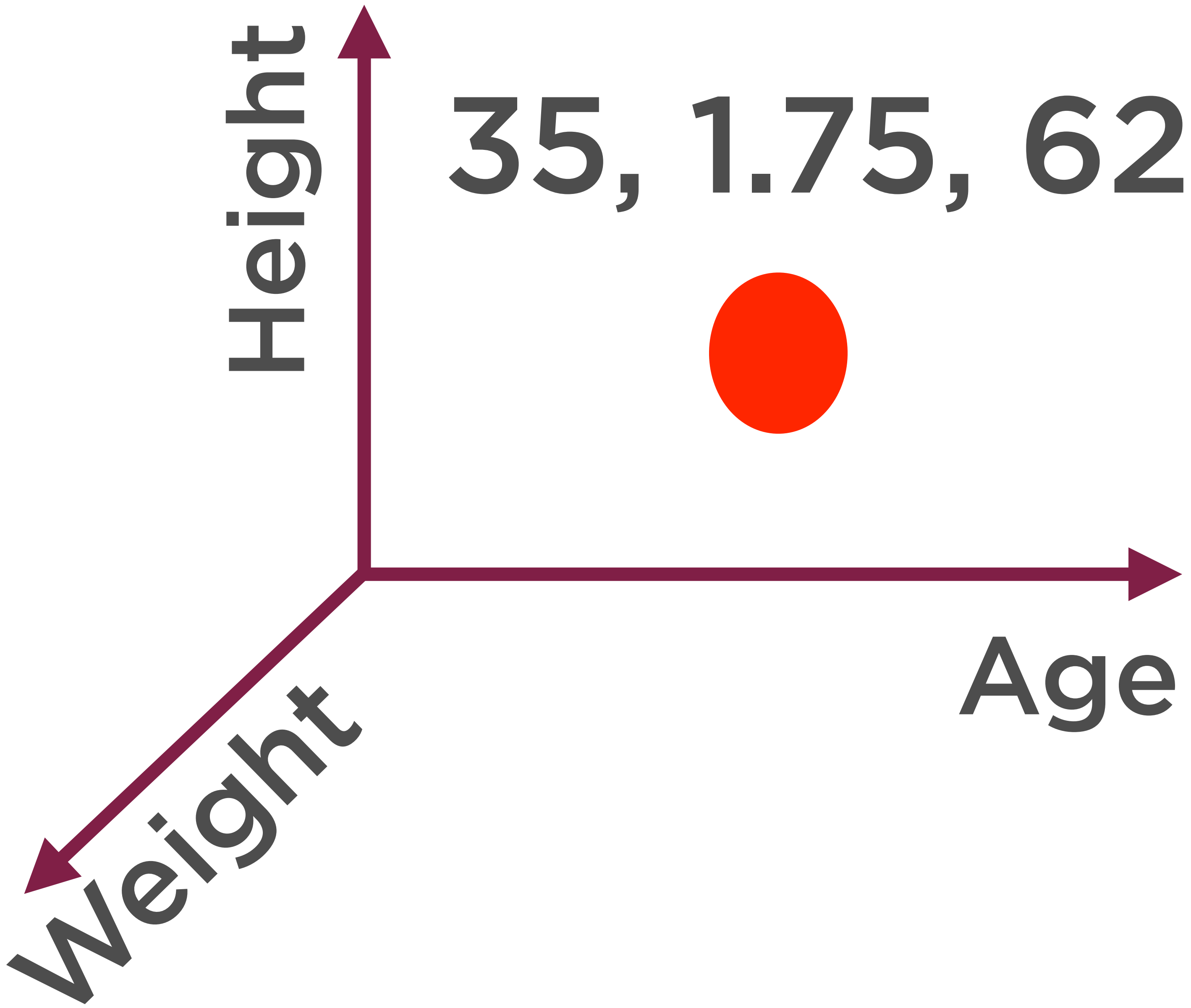
Age, Height, Weight



Age, Height, Weight



Age, Height, Weight



A set of N numbers represents a point in an **N-dimensional Hypercube**

The K-Means Clustering Algorithm

Clustering

A set of points, each representing a Facebook user

Clustering

Same group = similar

Different group = different

Clustering

Same group = similar

Different group = different

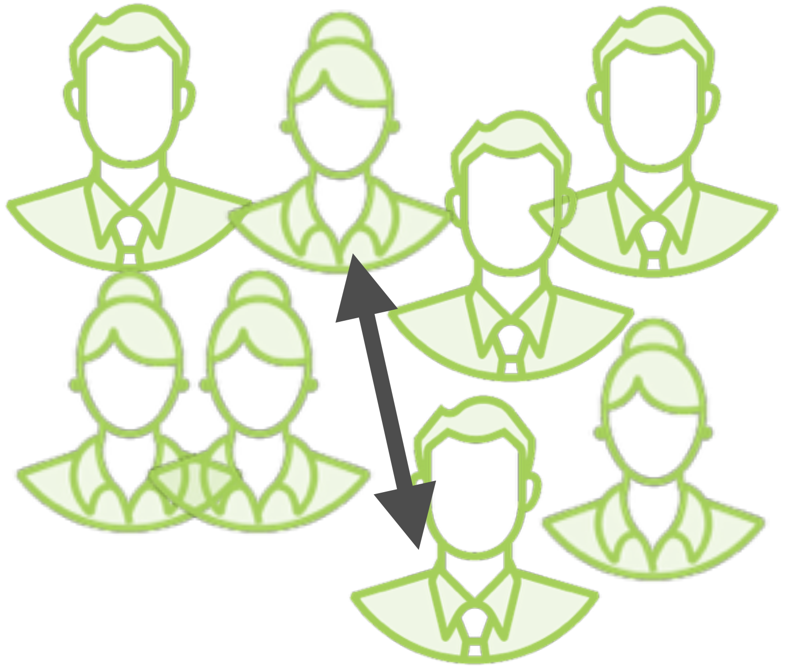
Users in a Cluster

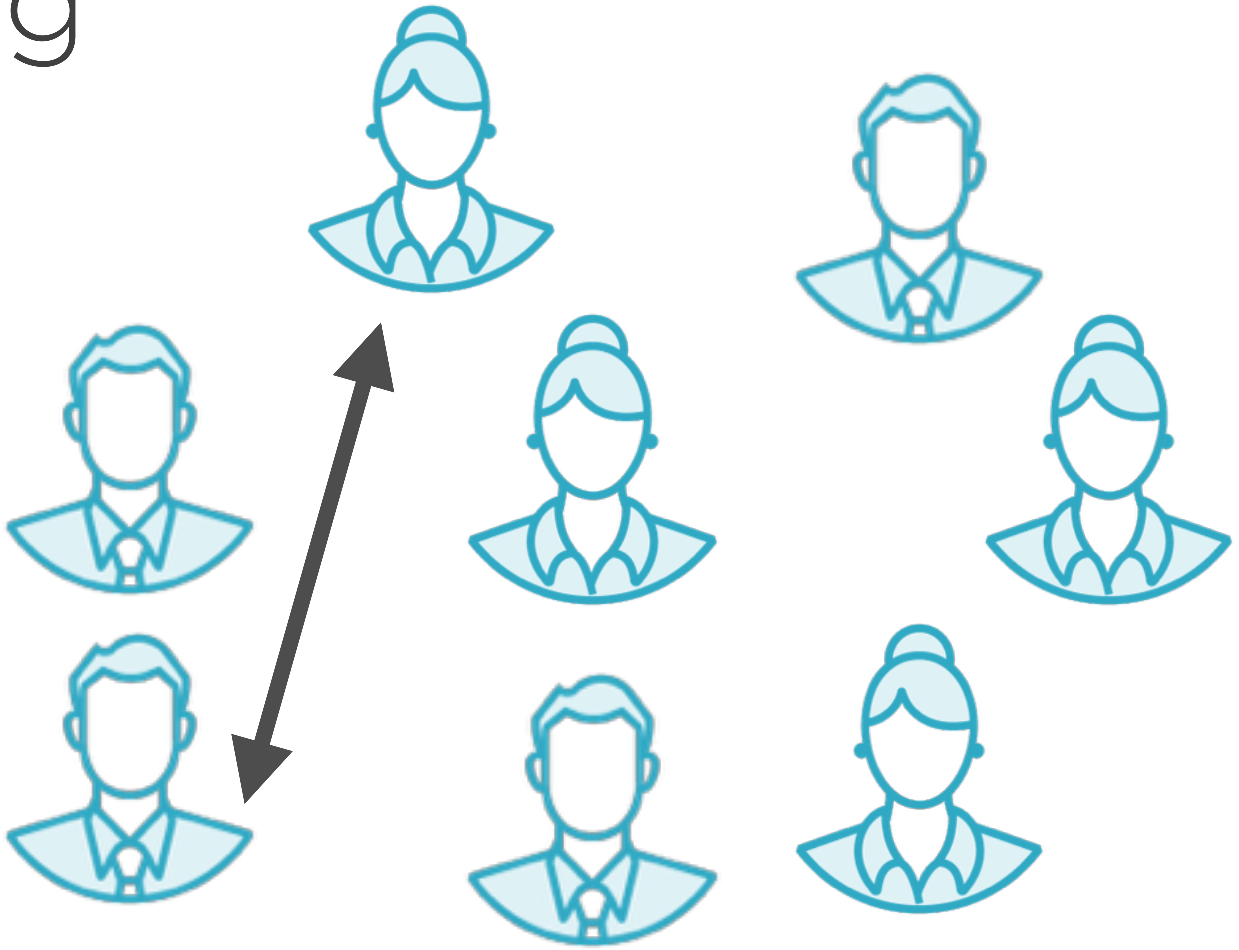
May like the same kind of music

May have gone to the same high school

May have kids of the same age

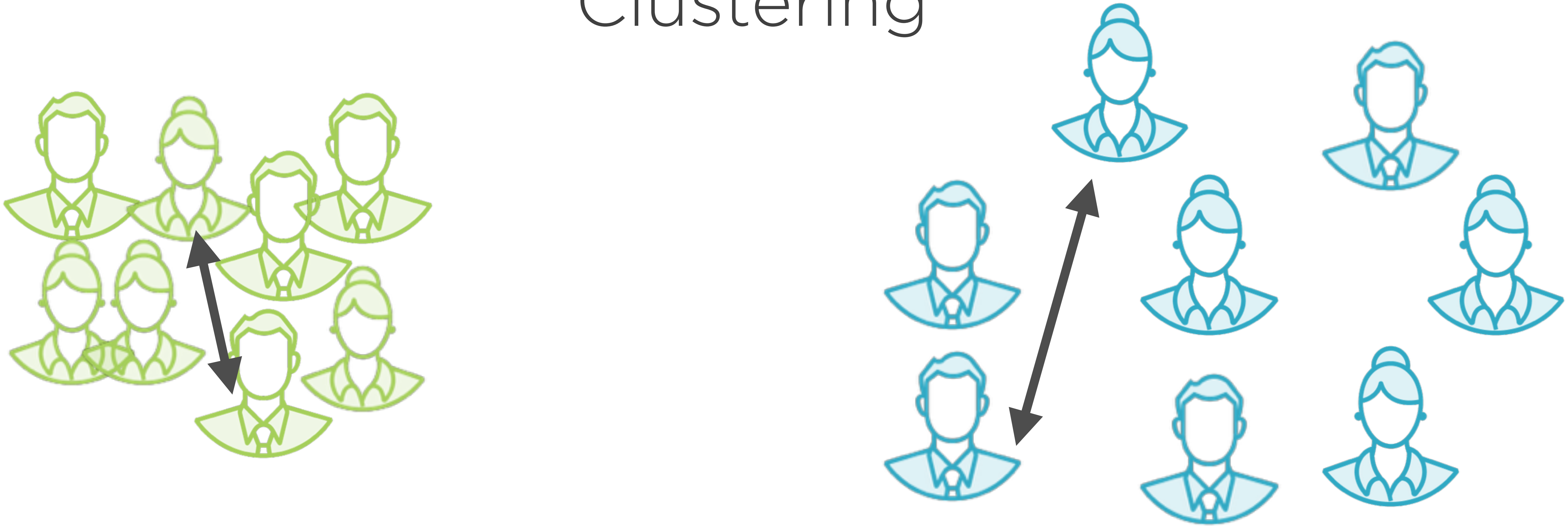
Clustering





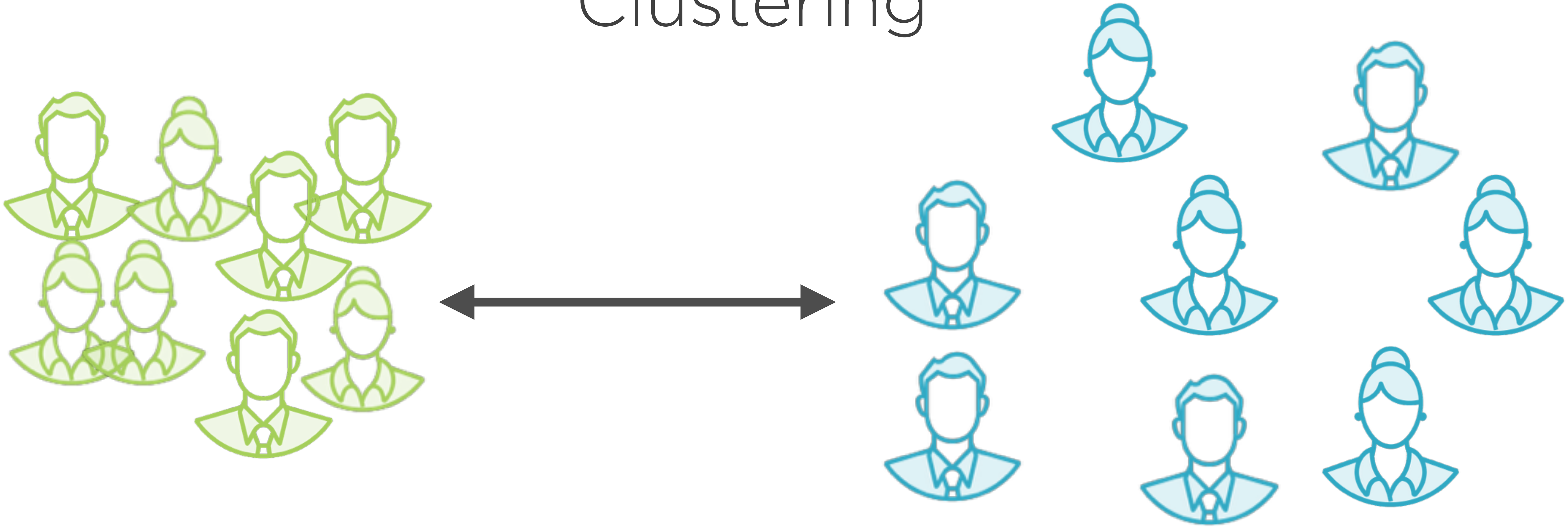
The **distance** between users in a cluster indicates how **similar** they are

Clustering



Maximize intra-cluster similarity

Clustering



Minimize inter-cluster similarity

Clustering Objective

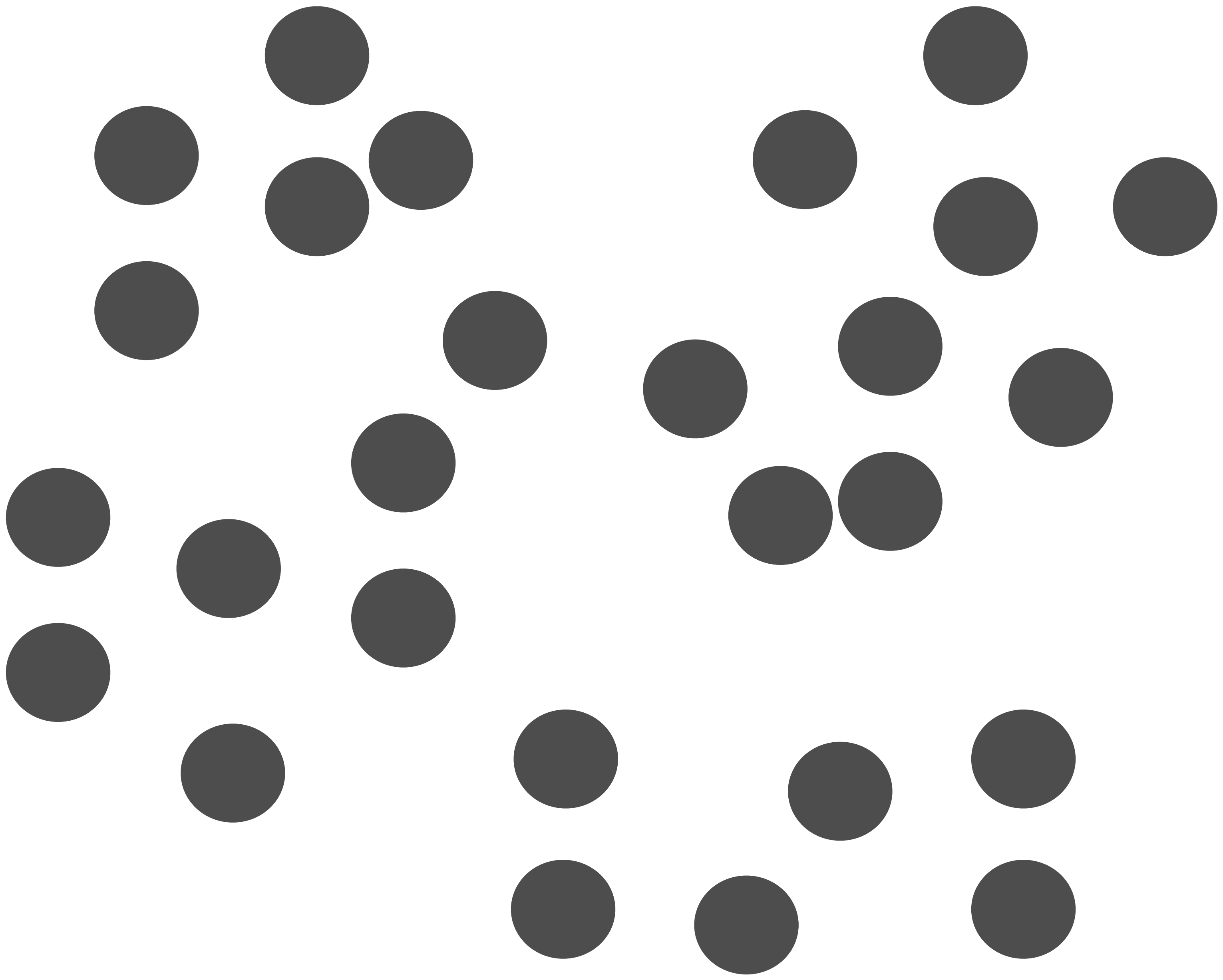
Maximize intra-cluster similarity

Minimize inter-cluster similarity

The **K-Means Clustering** algorithm is a famous Machine Learning algorithm to achieve this

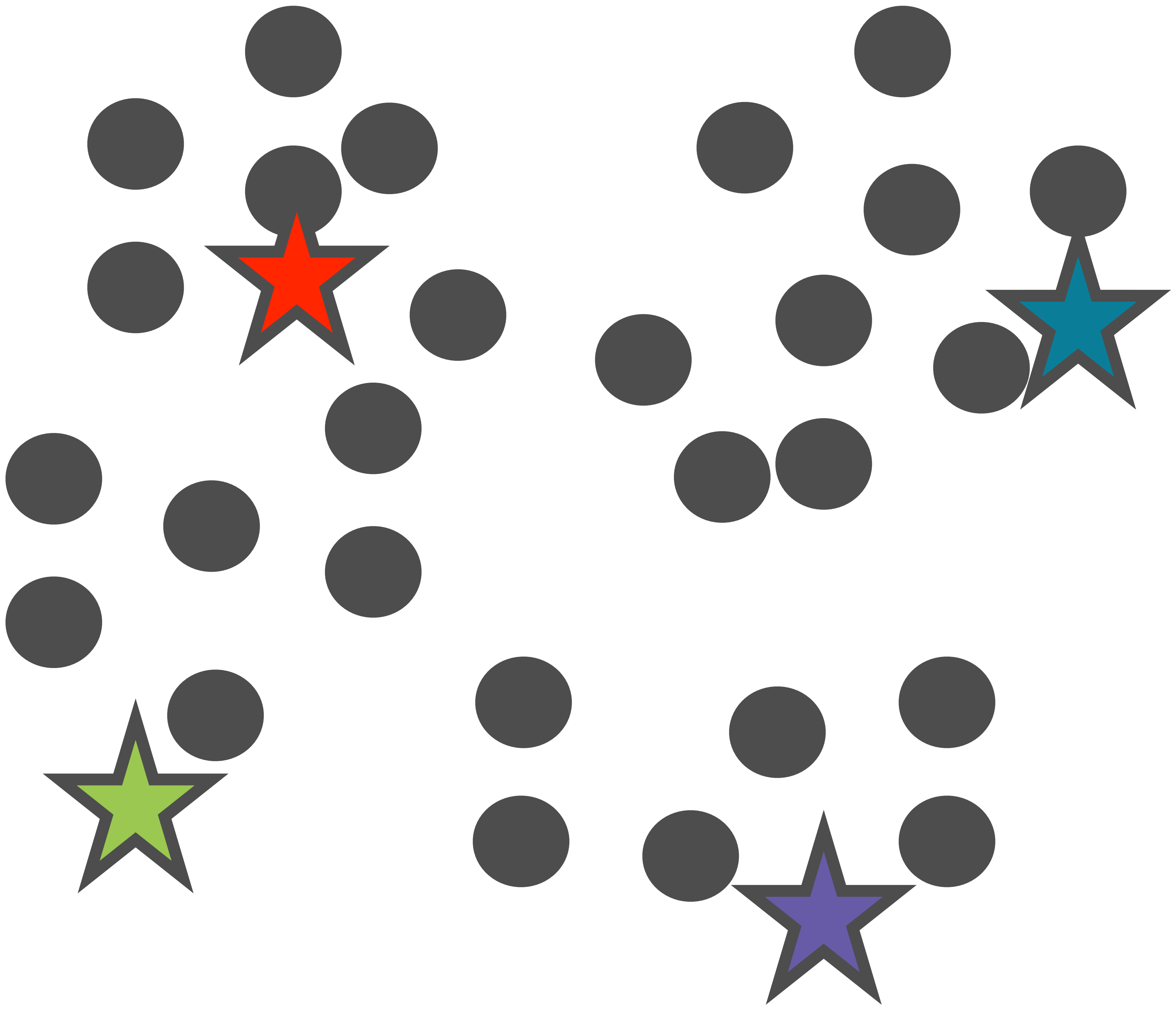
K-Means Clustering

Initialize K centroids i.e means



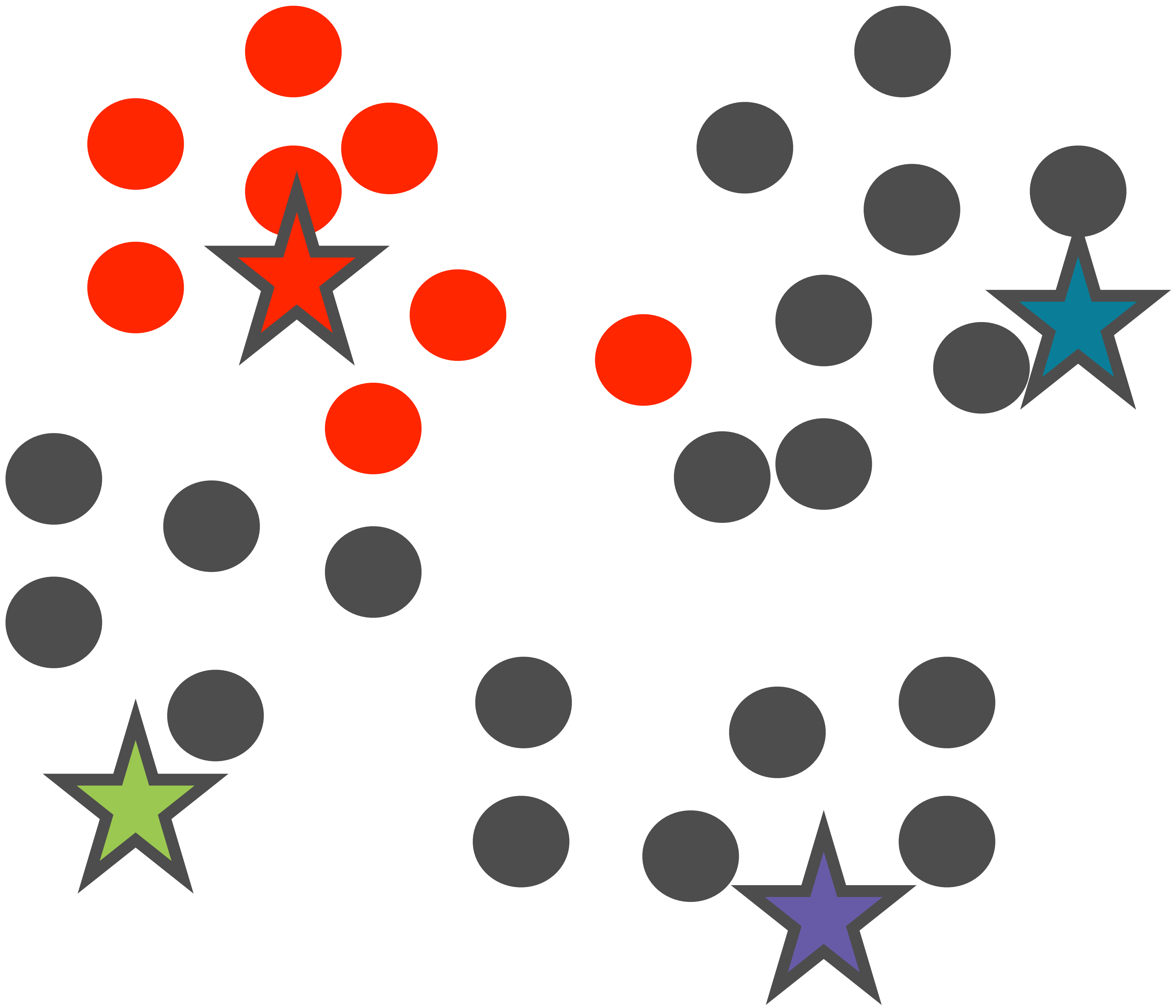
K-Means Clustering

Assign each point to a cluster



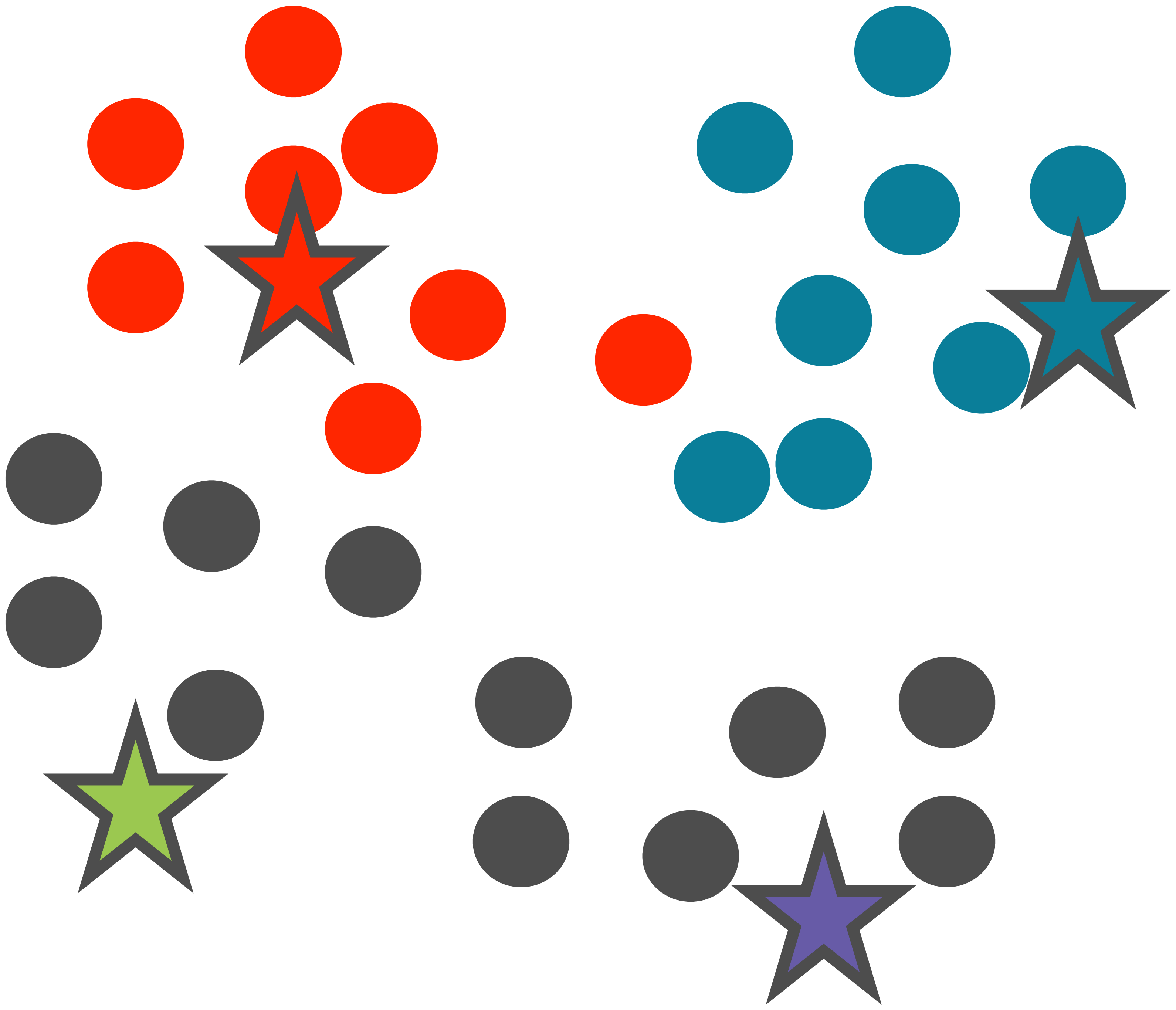
K-Means Clustering

Assign each point to a cluster



K-Means Clustering

Assign each point to a cluster



K-Means Clustering

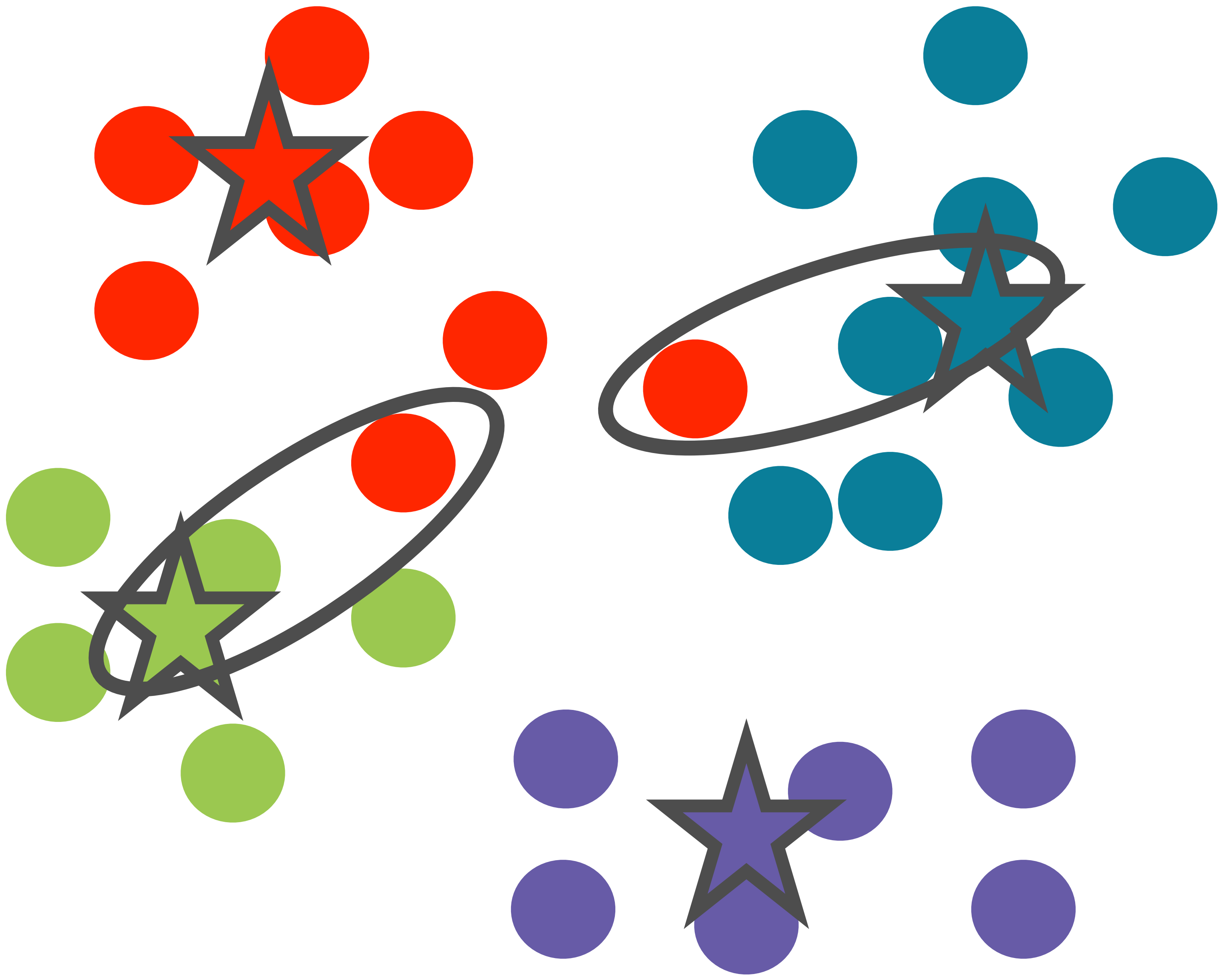
Assign each point to a cluster

K-Means Clustering

Recalculate the mean for each cluster

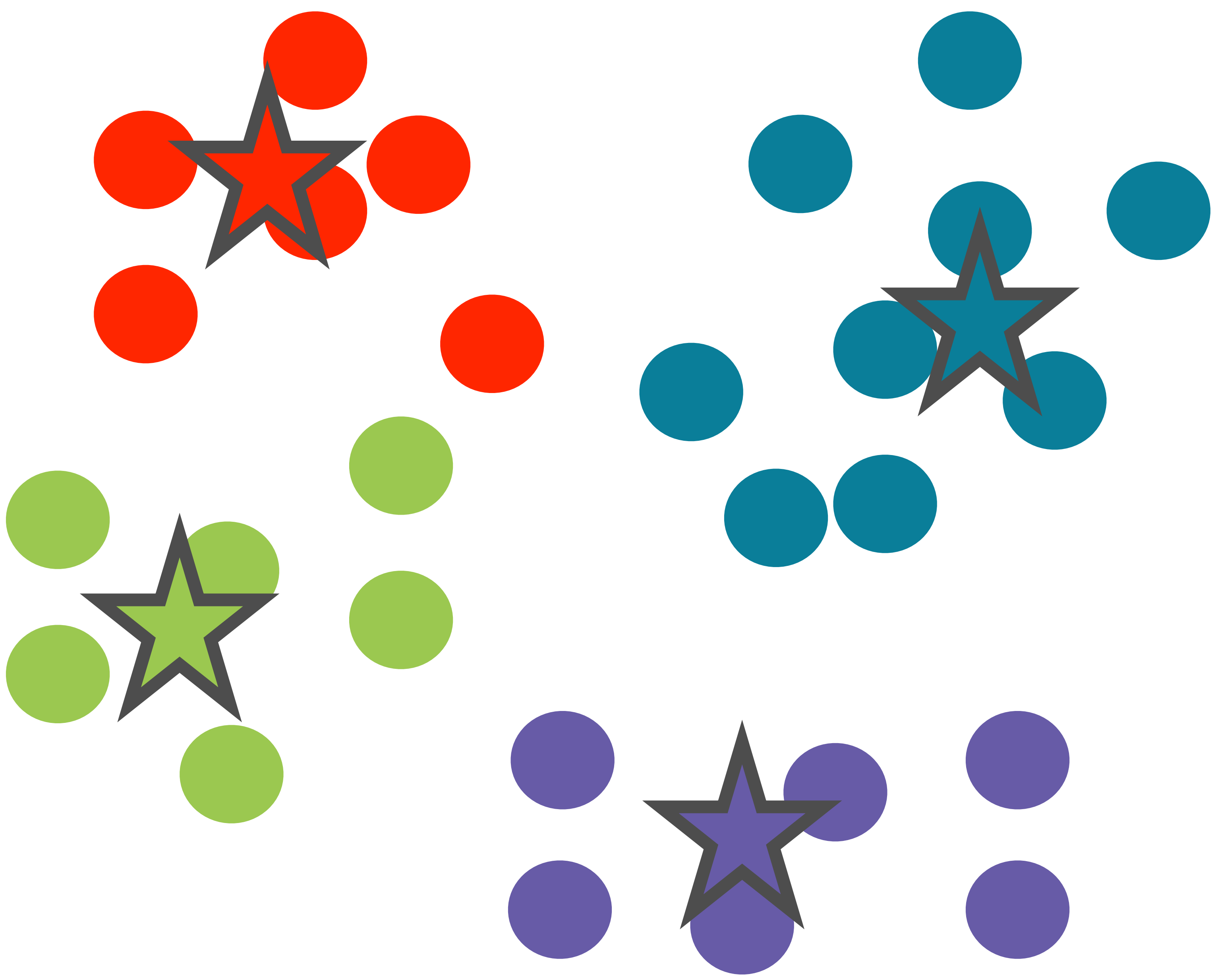
K-Means Clustering

Re-assign the points to clusters

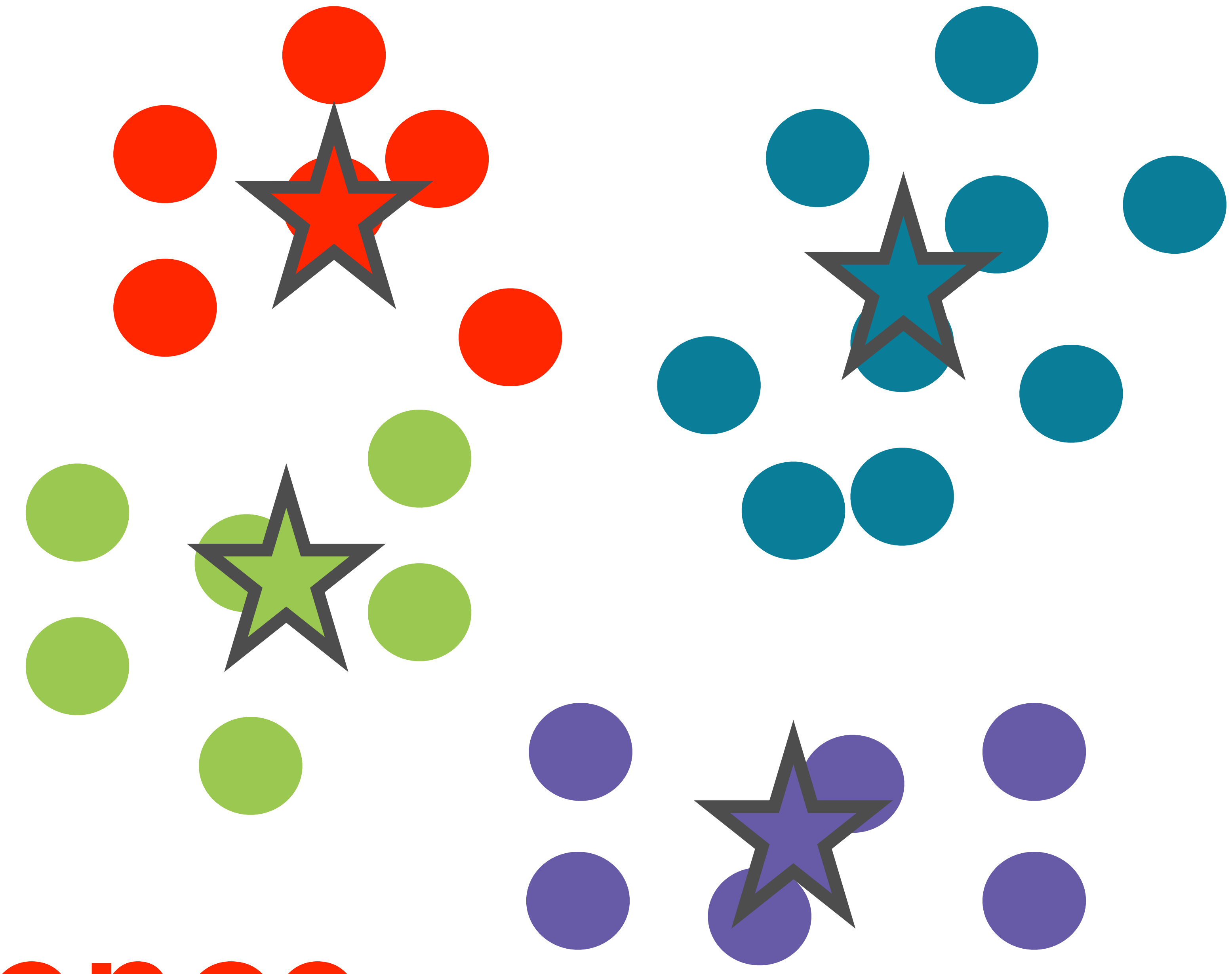


K-Means Clustering

Iterate until points are in their final clusters



K-Means Clustering

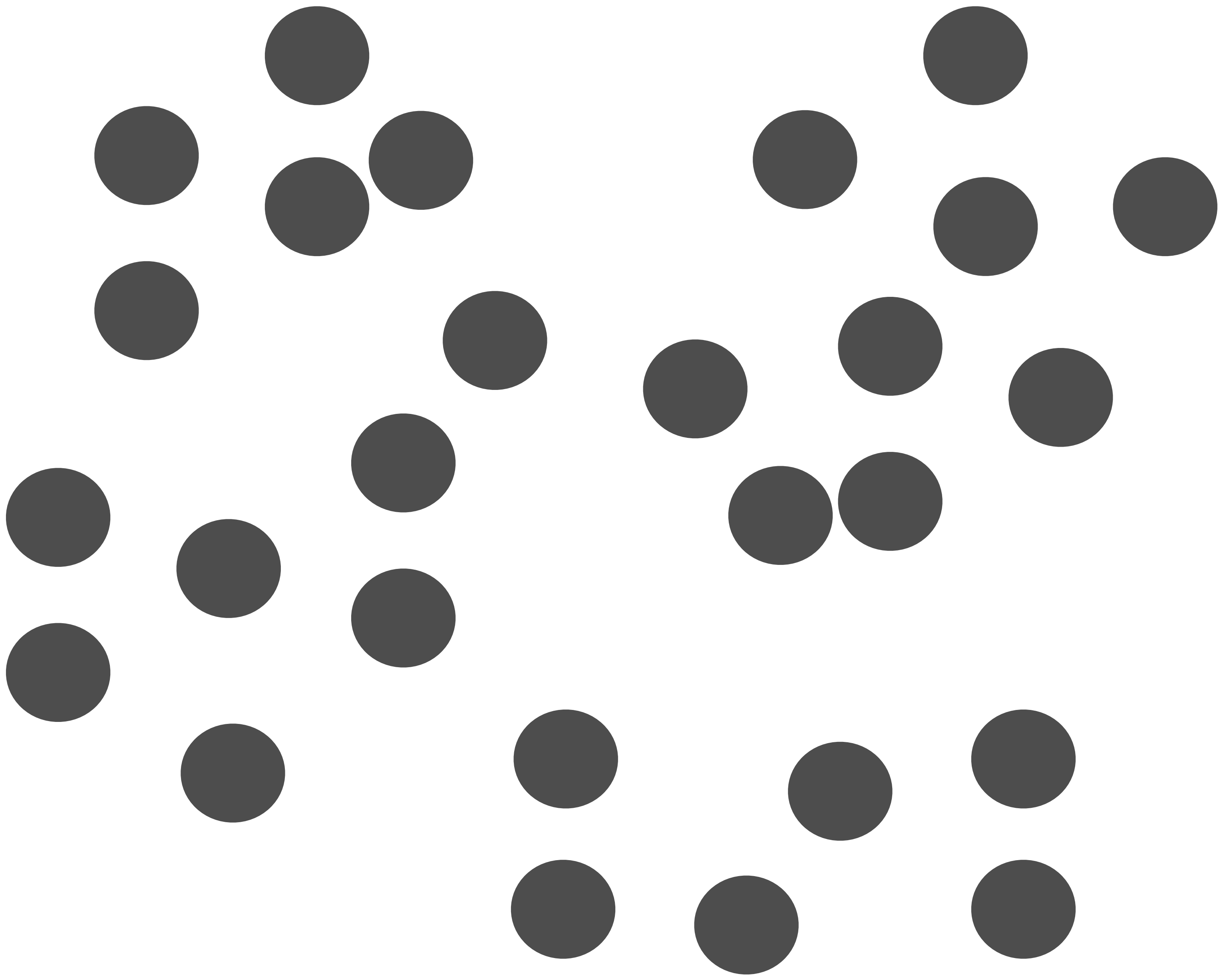


Convergence

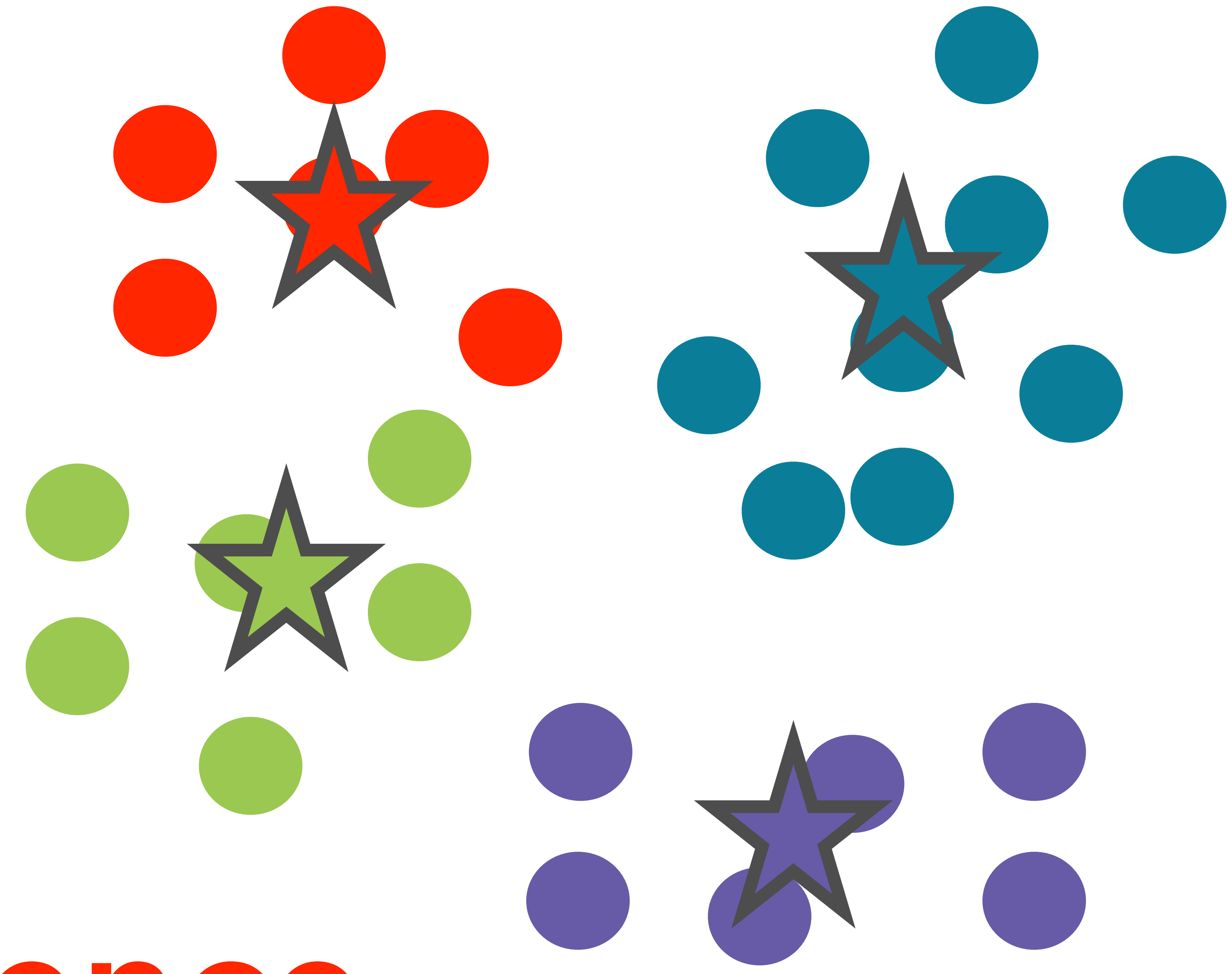
Applying K-Means Clustering to Streams

K-Means Clustering on Streams

First batch of data



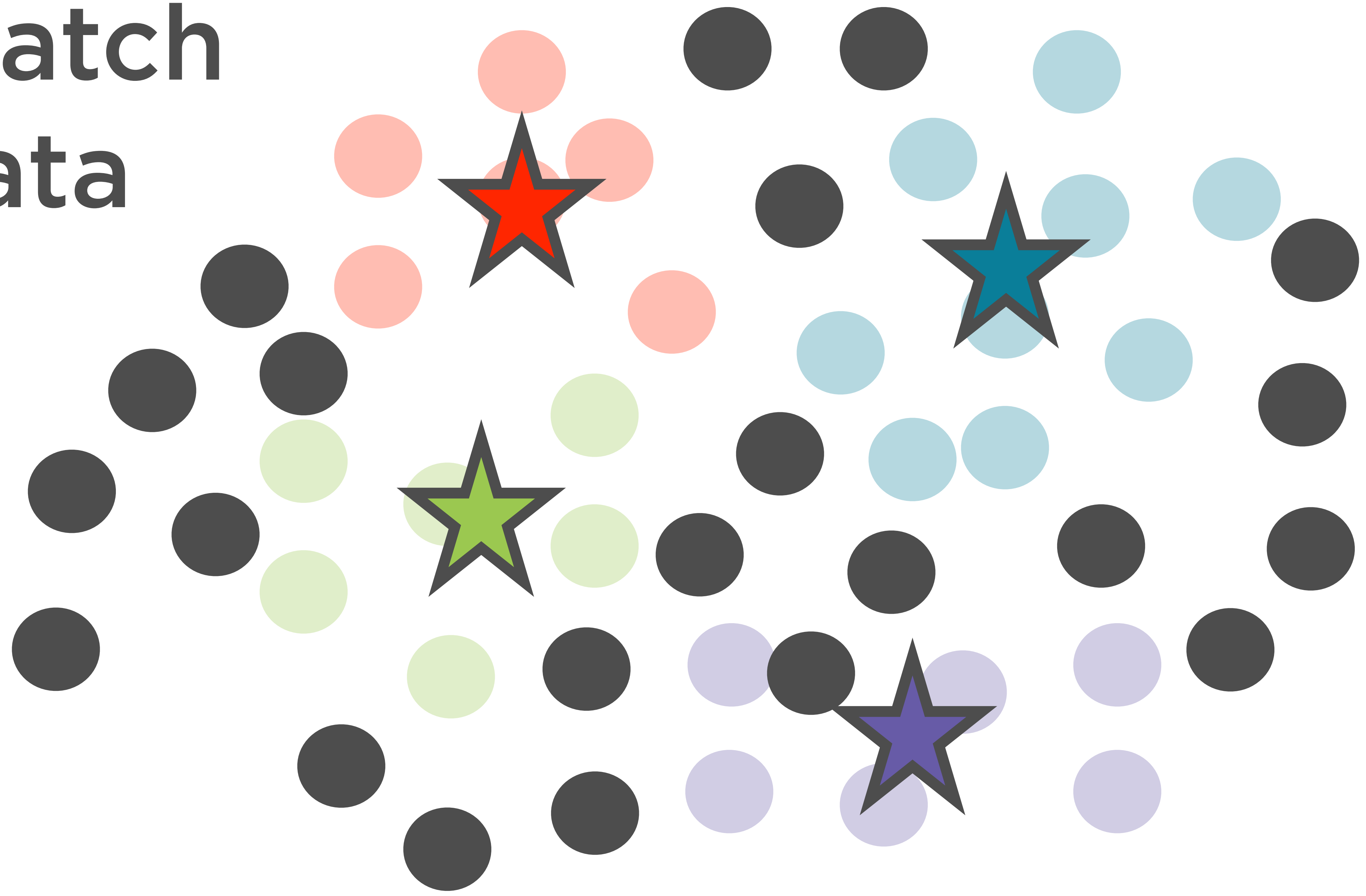
K-Means Clustering on Streams



Convergence

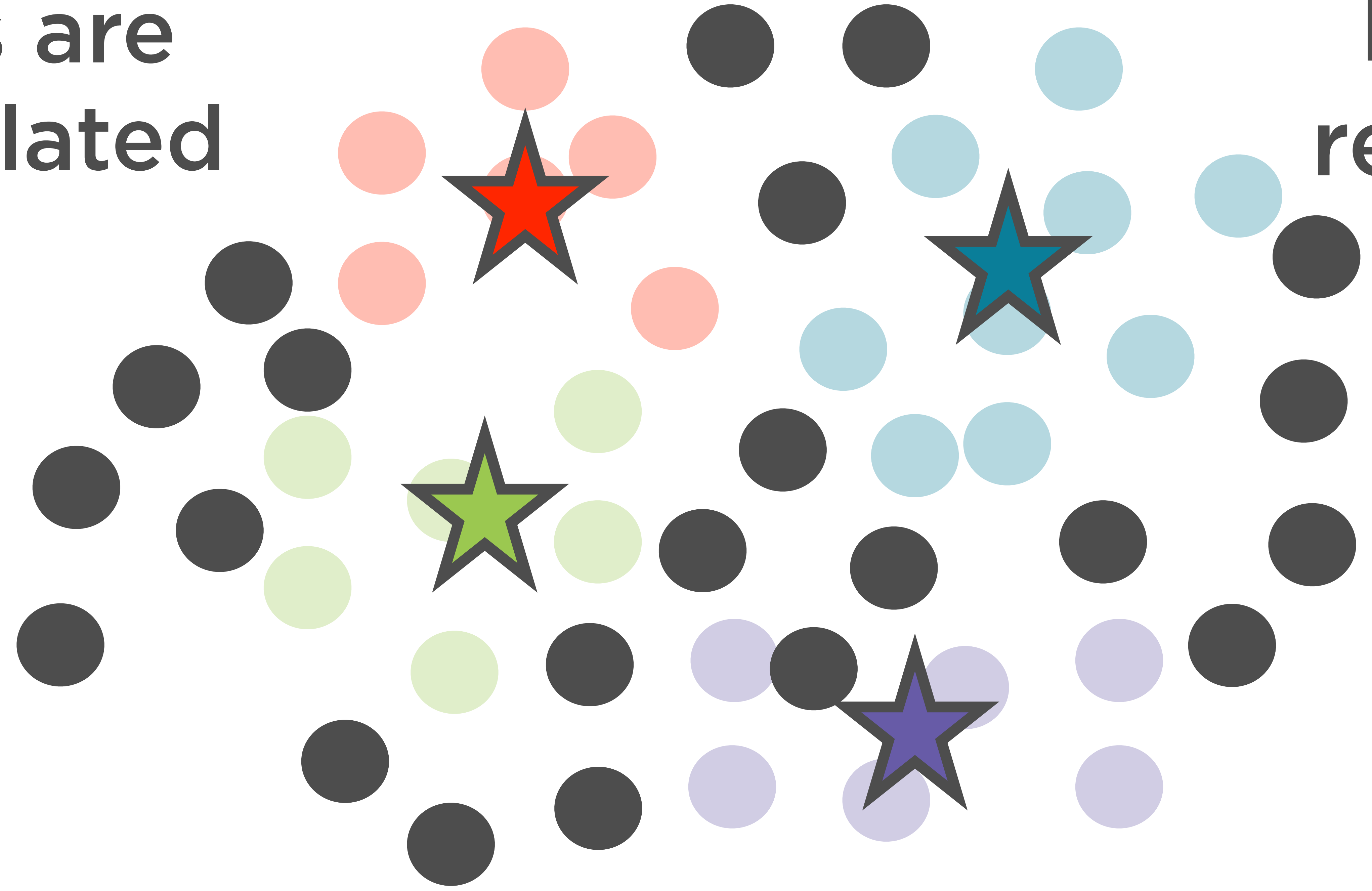
K-Means Clustering on Streams

New batch of data



K-Means Clustering on Streams

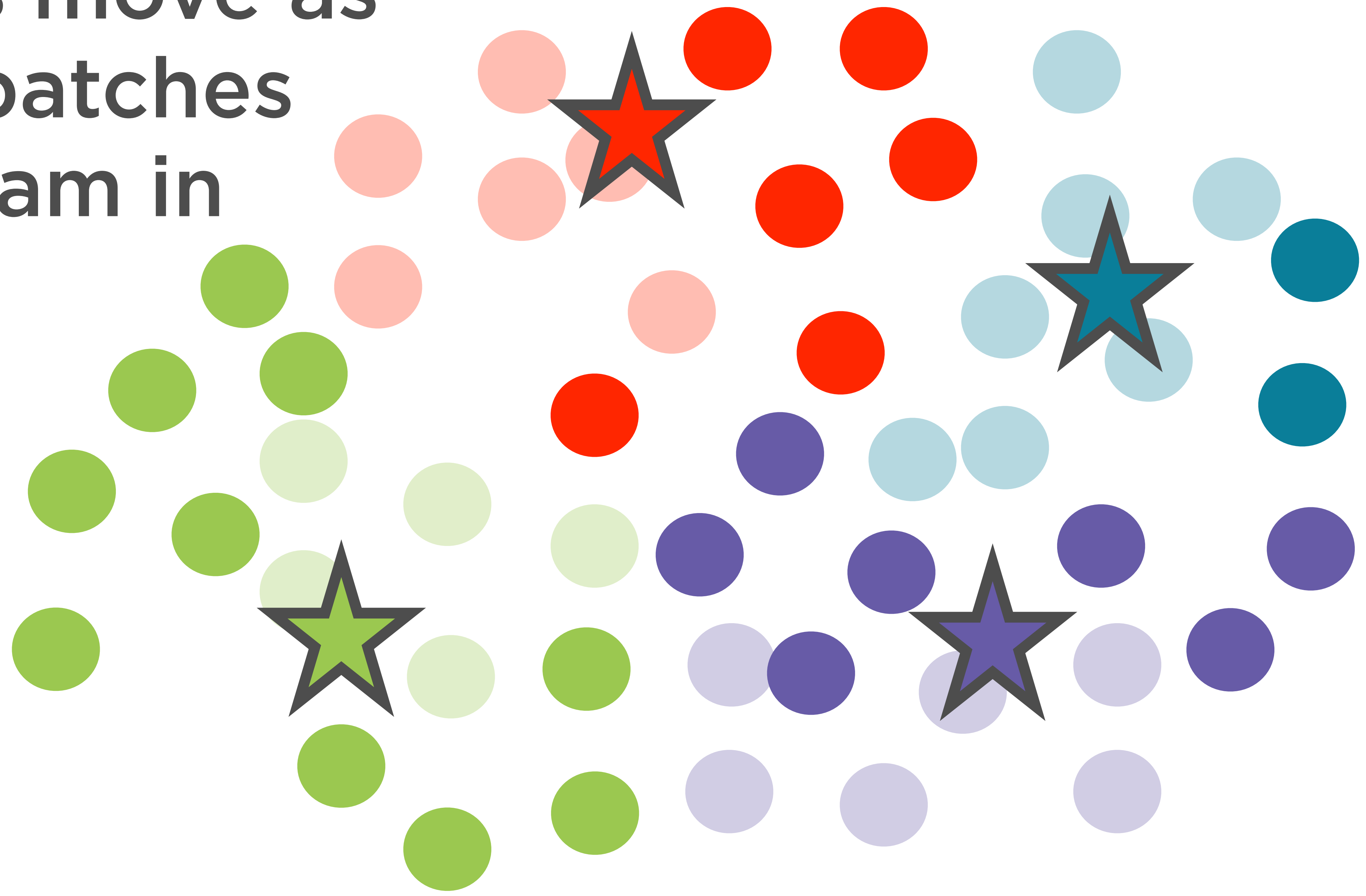
Means are recalculated



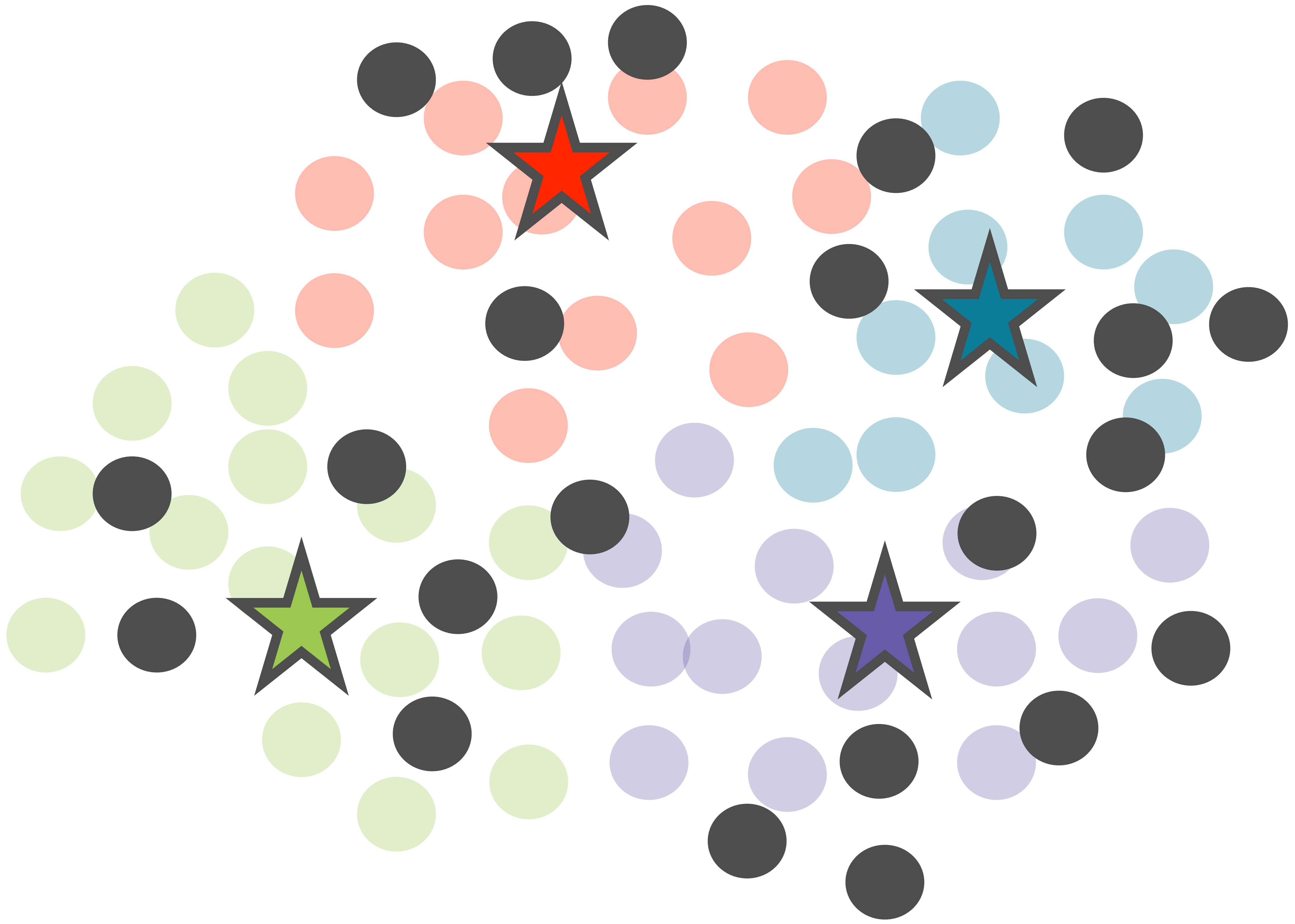
Points are re-classified

K-Means Clustering on Streams

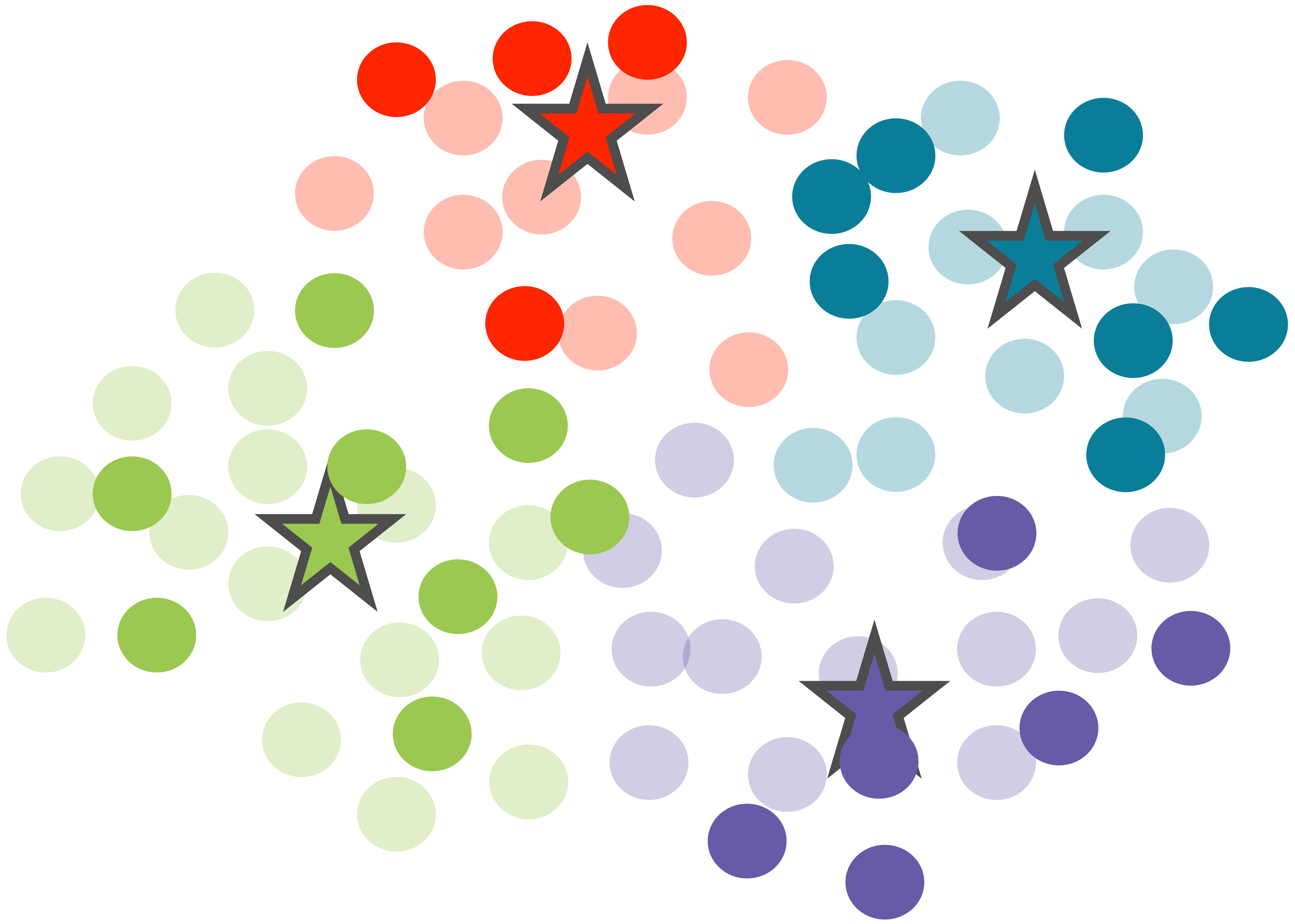
Centers move as new batches stream in



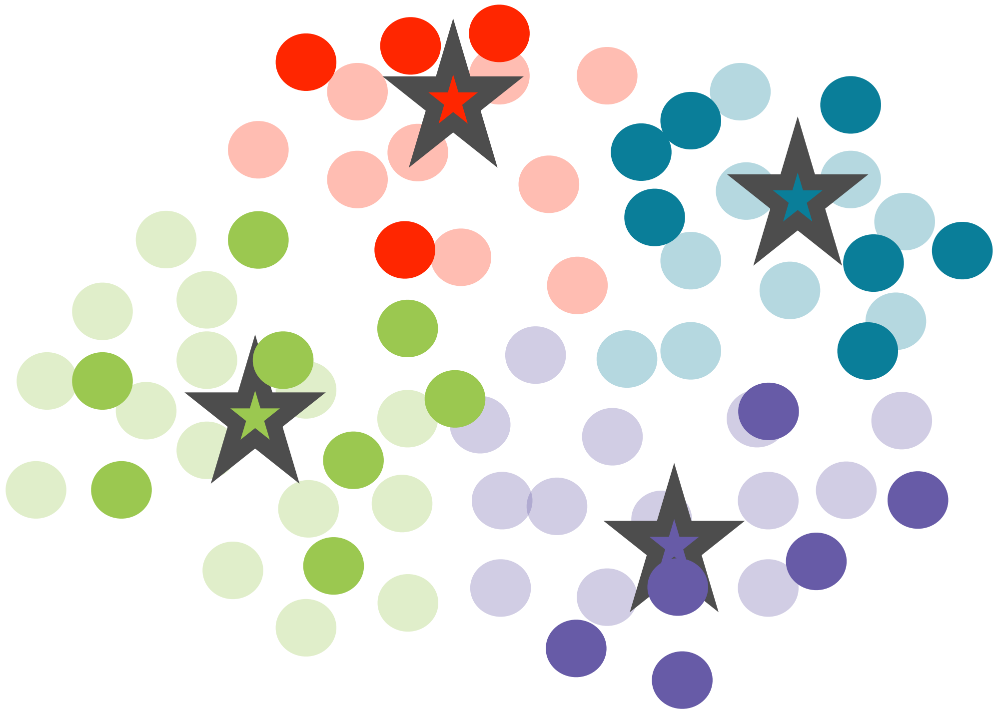
K-Means Clustering on Streams



K-Means Clustering on Streams



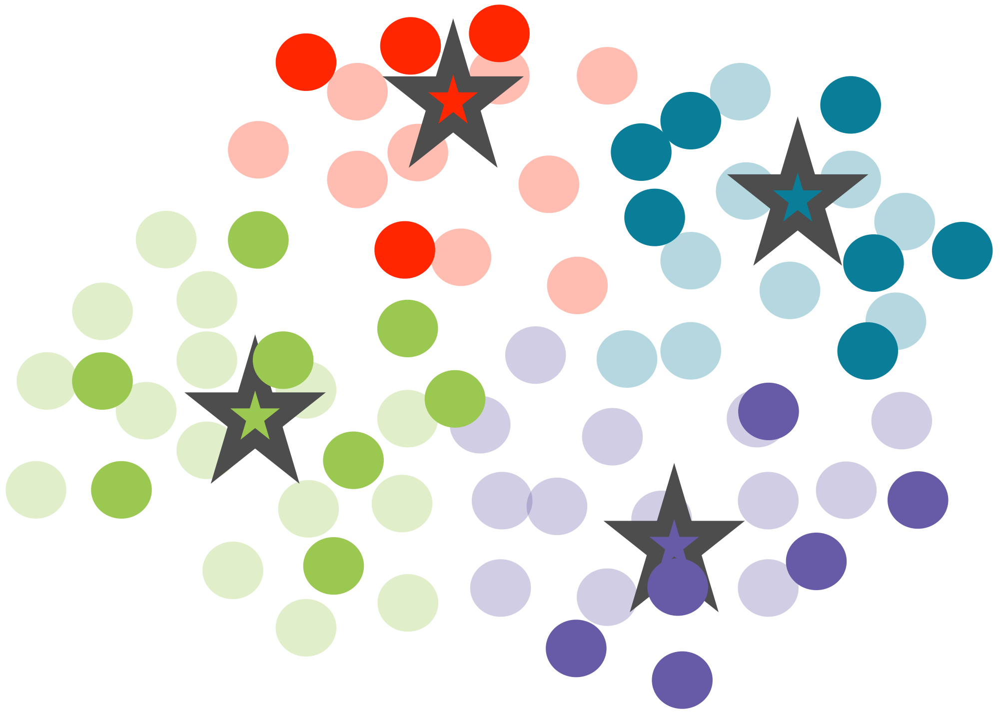
Is new data more **relevant** than older data?



Recent Data More Relevant

Trending topics on Twitter

- Tweets from a year ago are useless today
- The clustering should be entirely based on recent tweets



All Data Equally Relevant

Most active users by location

- Active users from a year ago are still relevant
- Newer users should have a higher weight

A **forgetfulness** metric can be specified on the **Streaming** K-Means Clustering algorithm

Forgetfulness in Streaming K-Means Clustering

How Does One Forget Older Data?

Recent data **Older data**

10	9	8	7	6	5	4

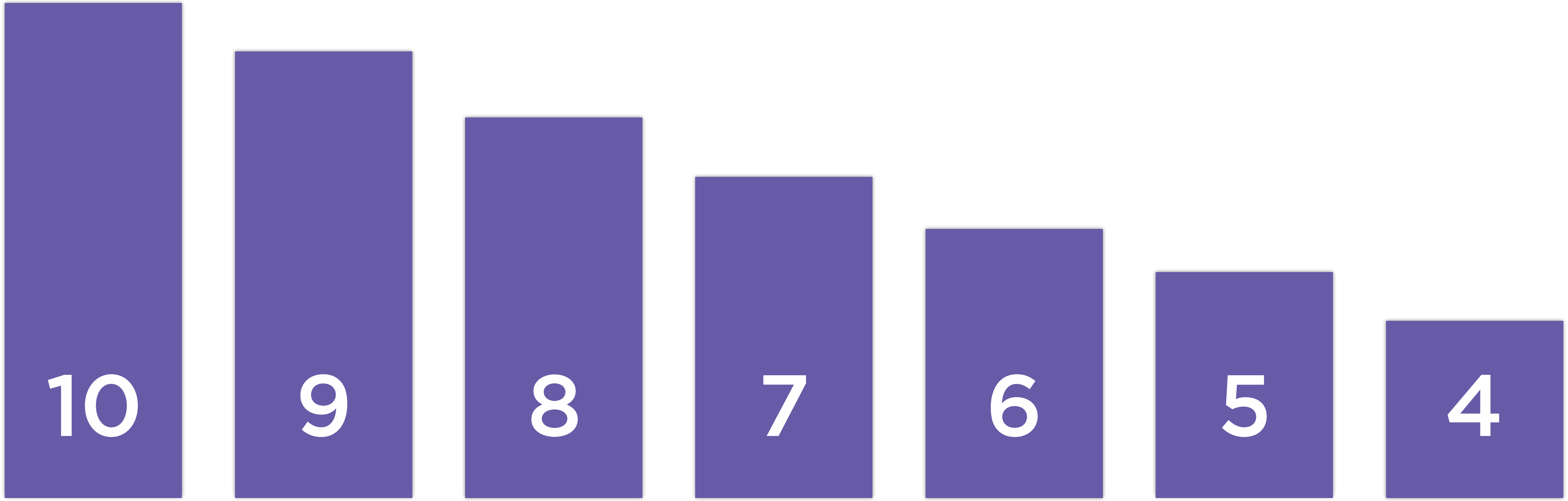
An average of numbers in this stream = 7

How Does One Forget Older Data?

10 9 8 7 6 5 4

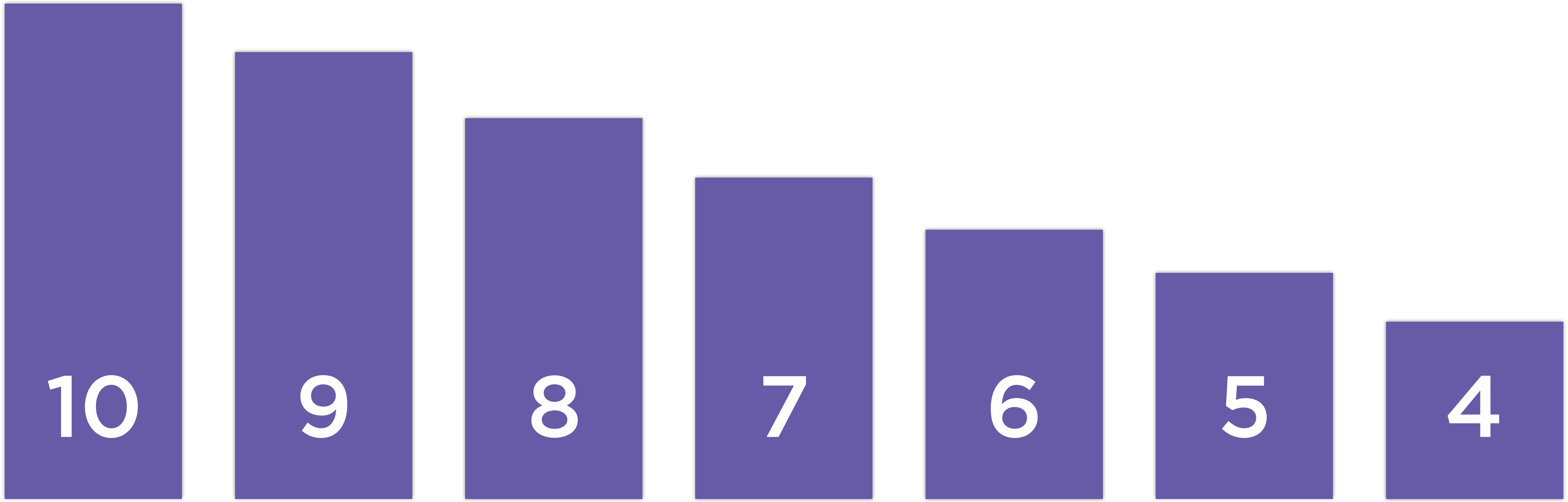
What if the most recent integers are more important?

How Does One Forget Older Data?



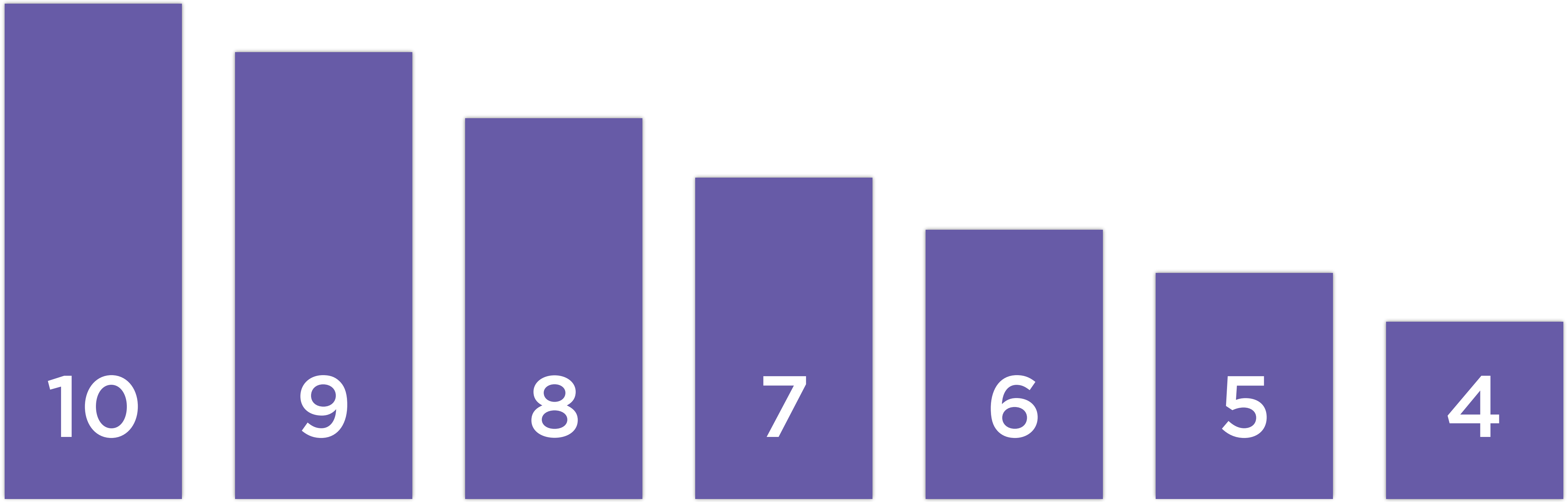
Apply higher weights to recent numbers in the stream

How Does One Forget Older Data?



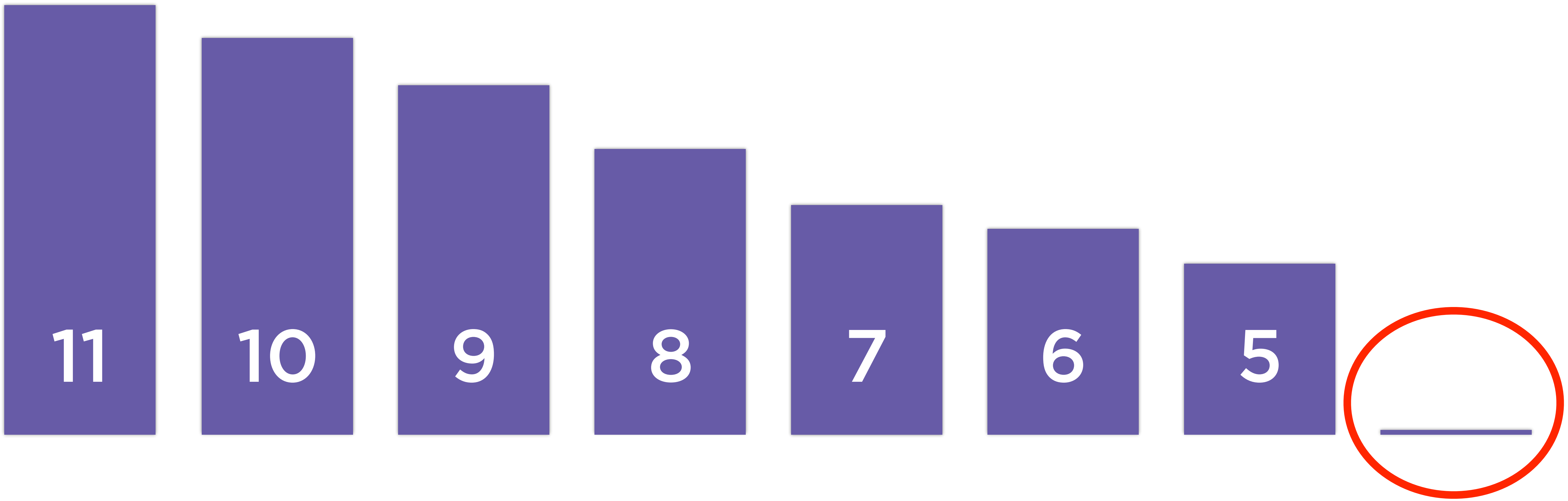
The weighted average of numbers in this stream will be >7

How Does One Forget Older Data?



The weight for older data can be progressively reduced to 0

How Does One Forget Older Data?



Older data will be forgotten

Forgetfulness

Forgetfulness

Makes the algorithm adaptive
to changing datasets

Forgetfulness

Balance the importance of new data versus old

- All data from the beginning of time treated equally

OR

- Use only the most recent data, discard the rest

Forgetfulness

Decay factor

A scalar quantity which determines how much of the old data is considered

Half-life

A time at which old data contributes to only half the model

Decay Factor

Value ranges from 0 to 1

- Decay factor = 1: Use all data from the beginning
- Decay factor = 0: Use only the most recent data

Decay Factor

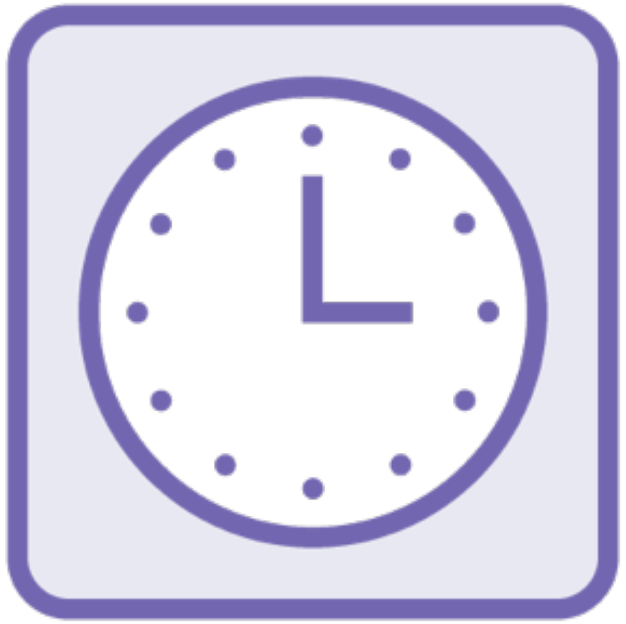
The values 0 and 1 make sense

Other values in the range are not intuitive

Forgetfulness

Decay factor

A scalar quantity which determines how much of the old data is considered



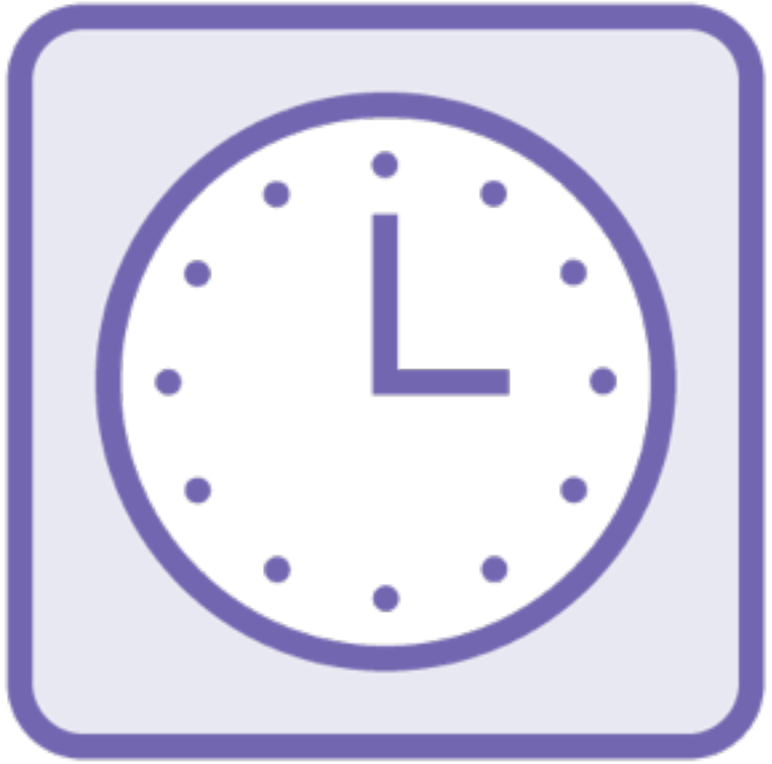
Half-life

A time at which old data contributes to only half the model

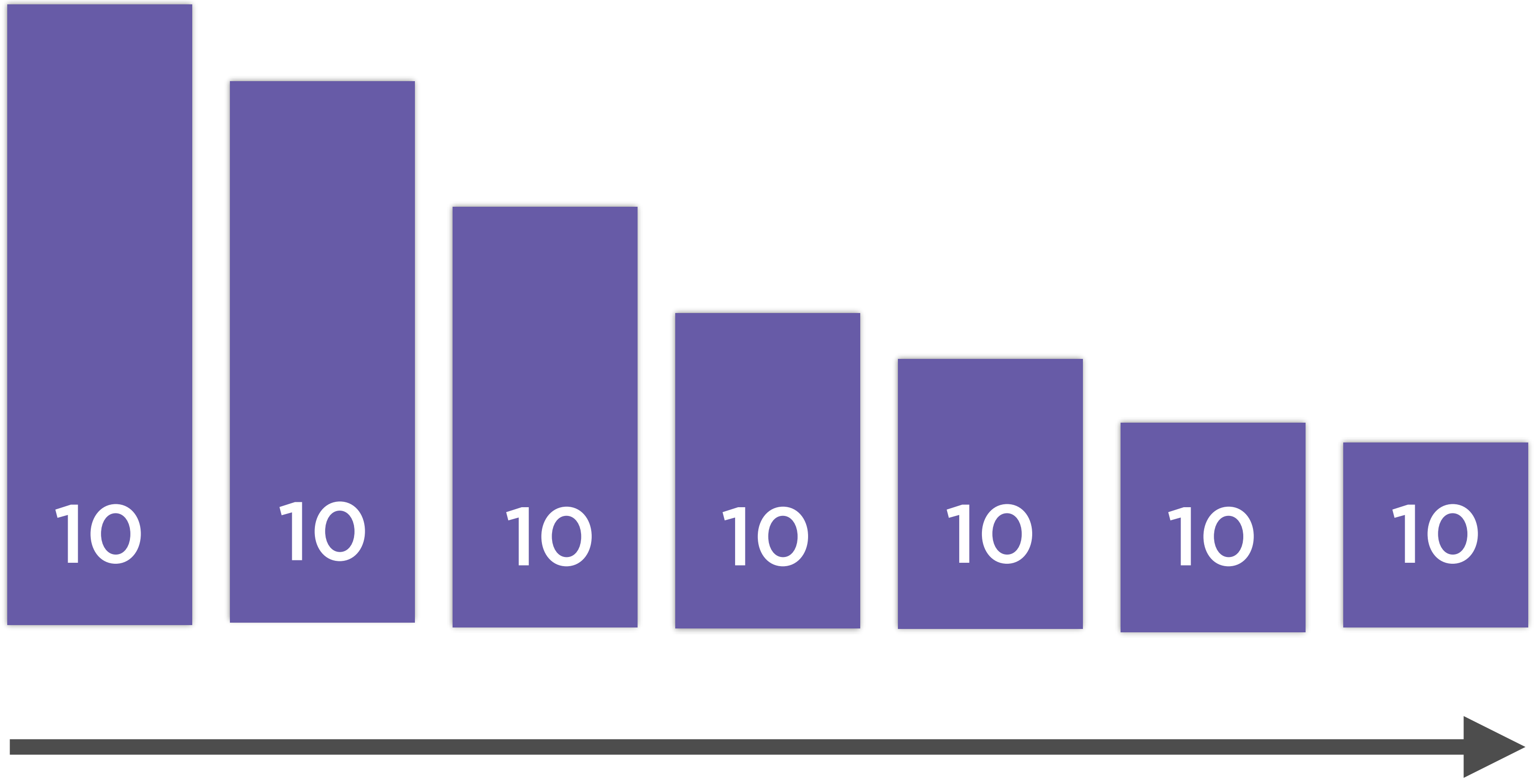
Half-life

Past data should progressively contribute less to the current model

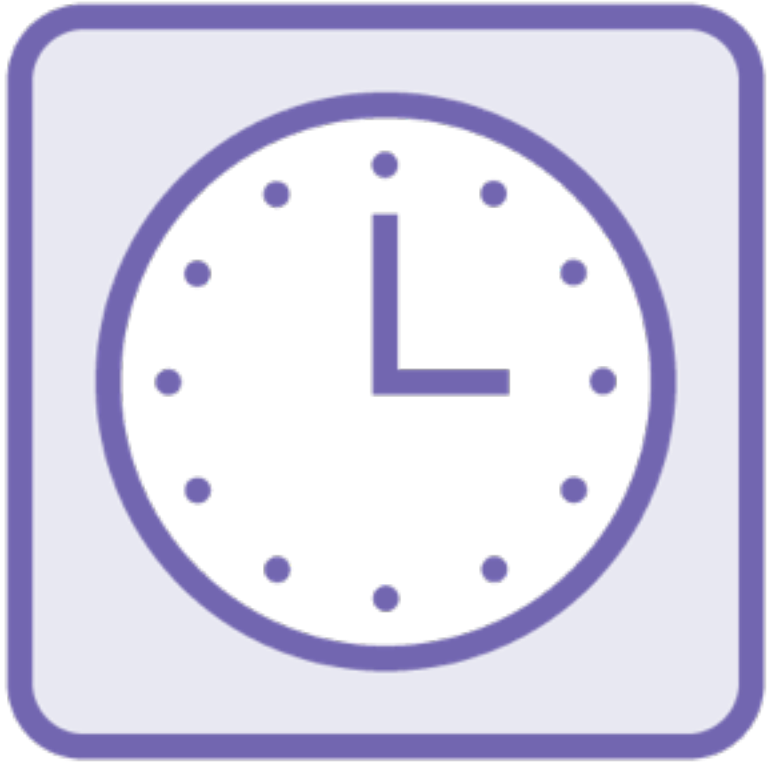
Define a time at which a batch contributes to only half of the current model

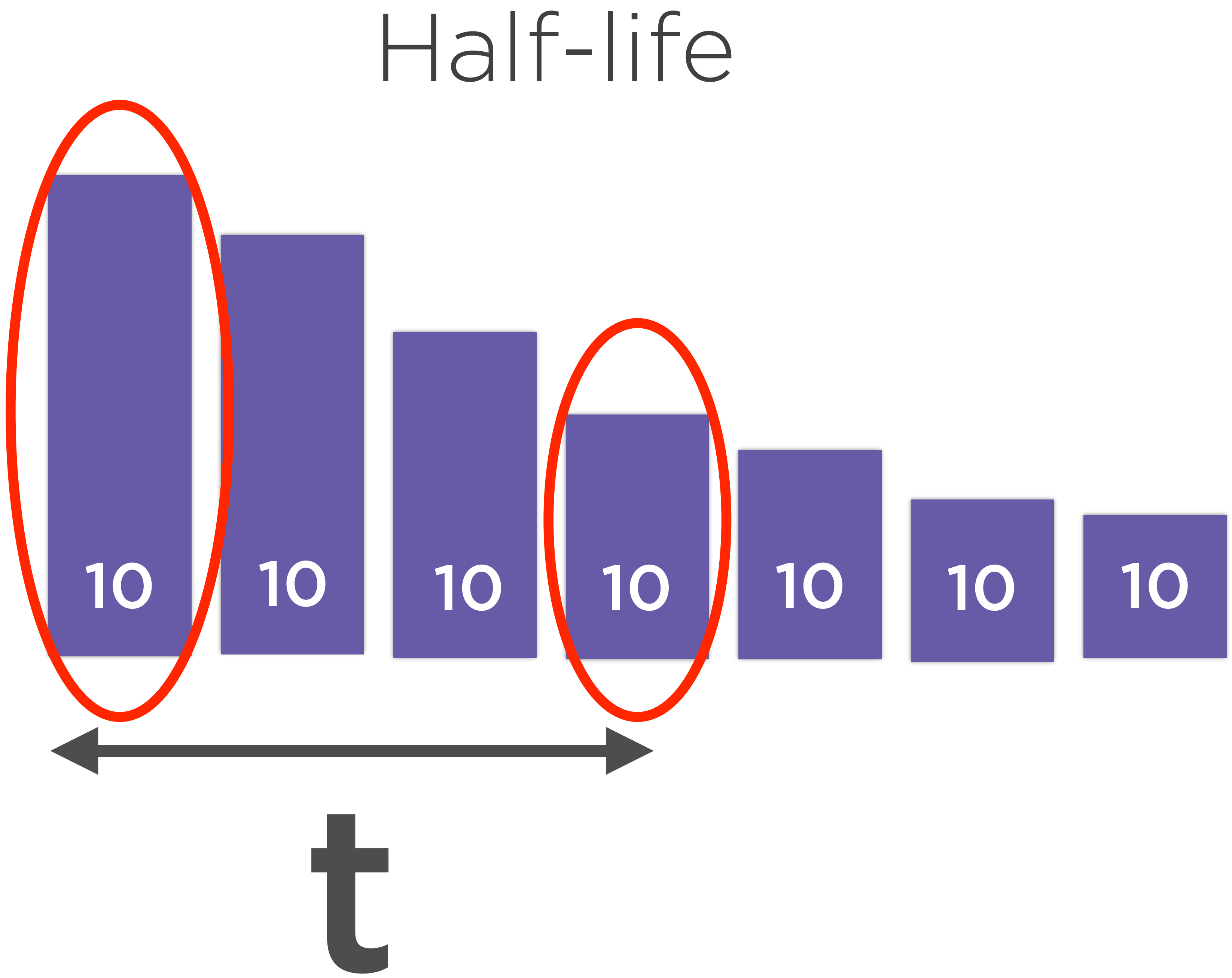


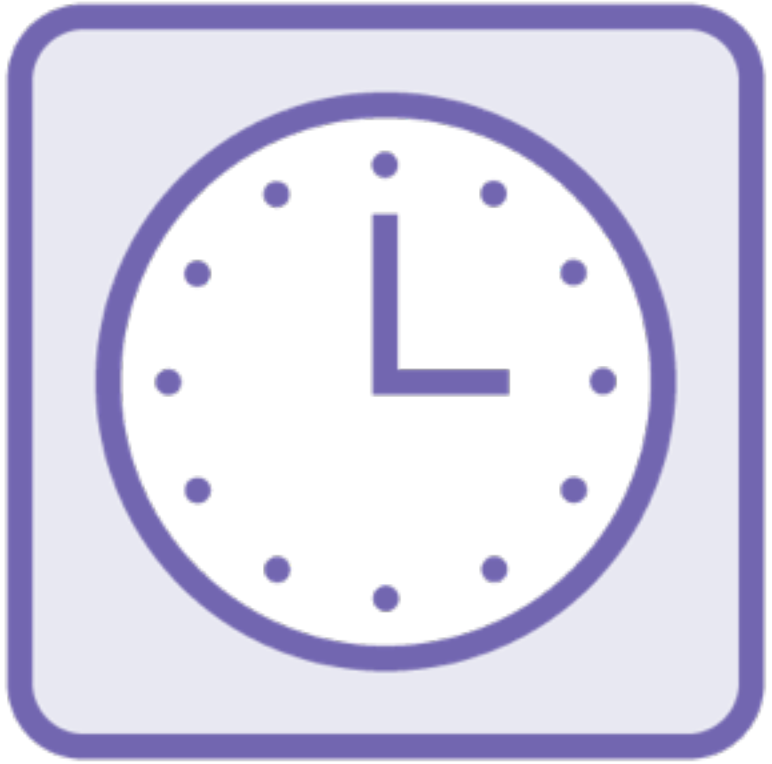
Half-life



The same element gets older




Half-life
10
10
10
10
10
10
10
t



Half-life

The half-life can be specified in one of 2 ways

- number of batches (each batch is a fixed unit of time)
- number of data points (each batch has a variable number of data points)

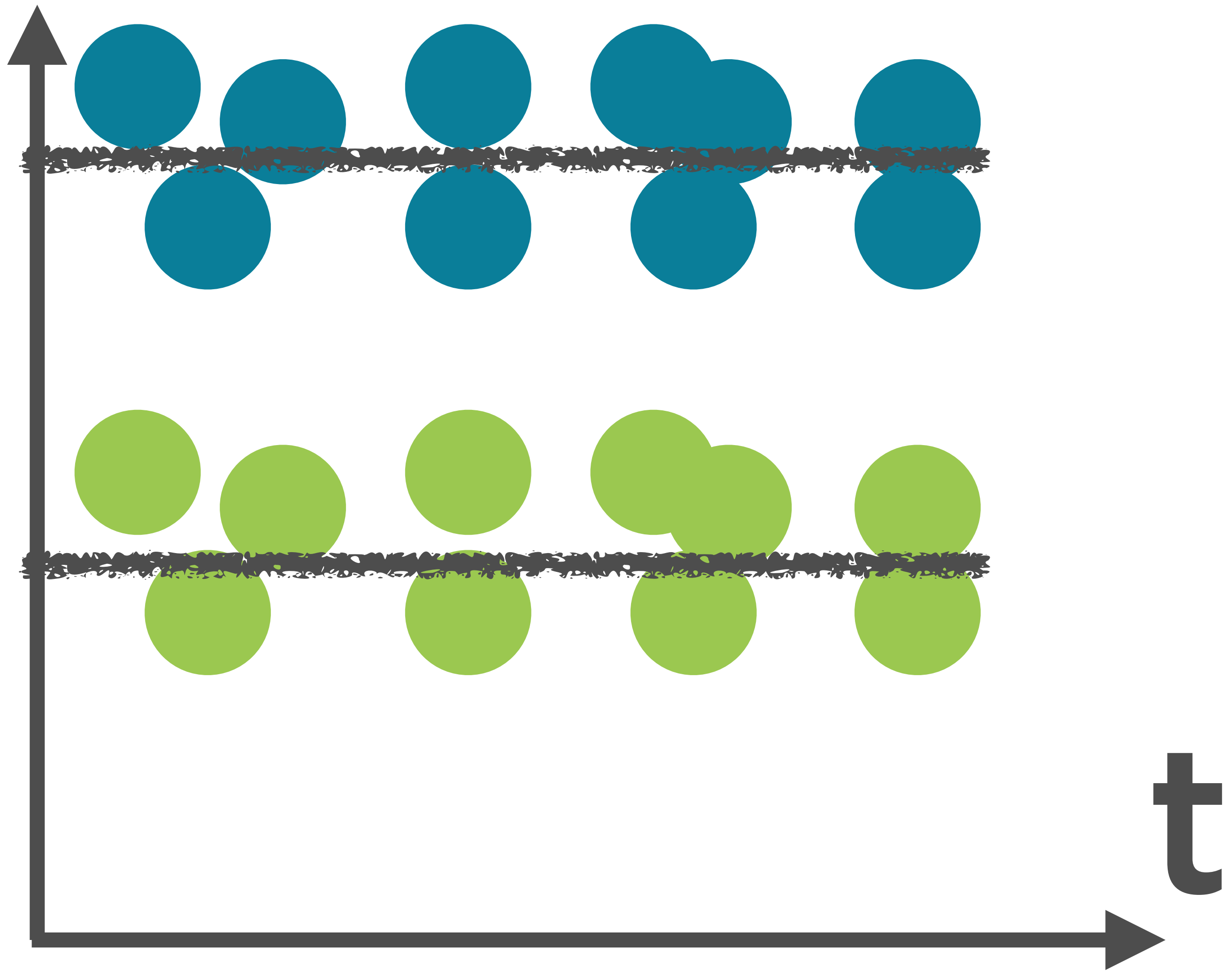
Forgetfulness

Forgetfulness

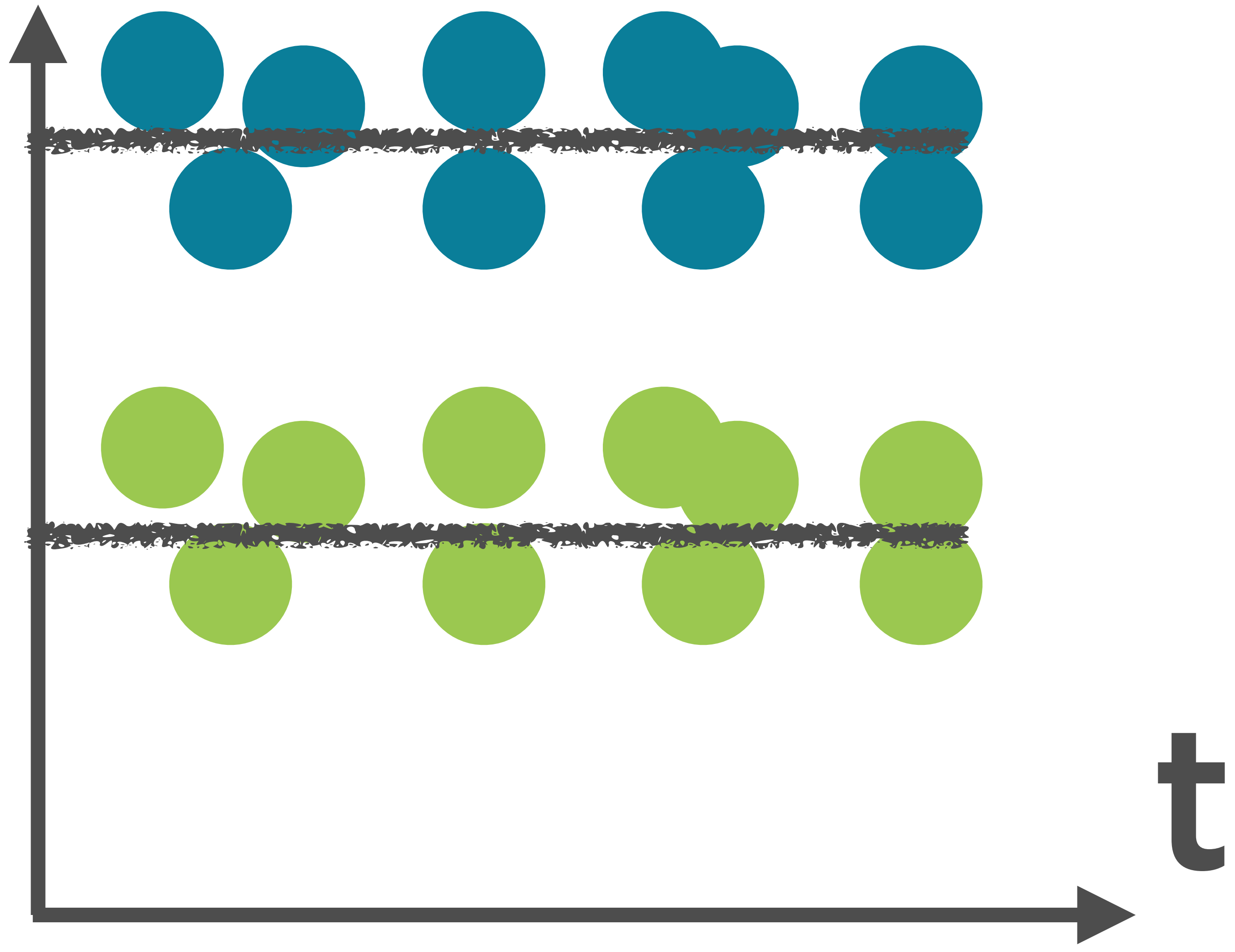
Makes the algorithm adaptive
to changing datasets

K=2

Effect of Half-life

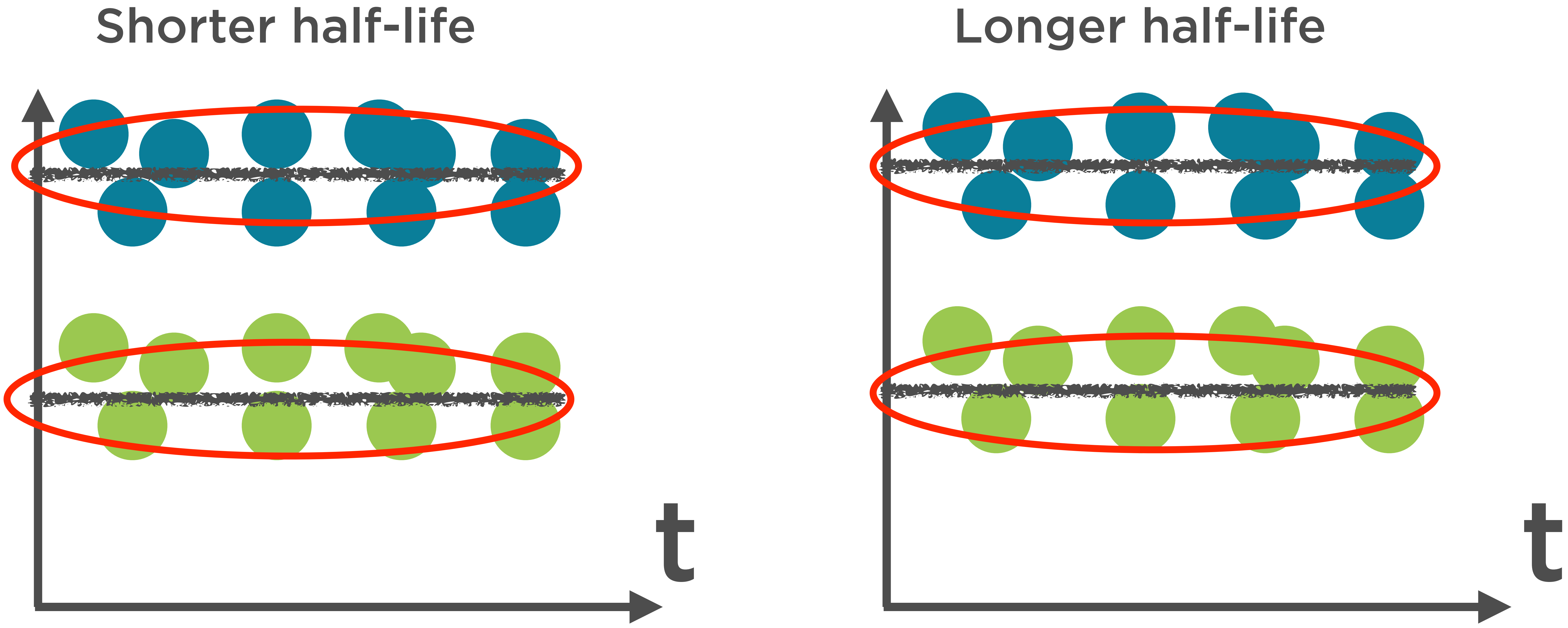


half-life = 1 batch



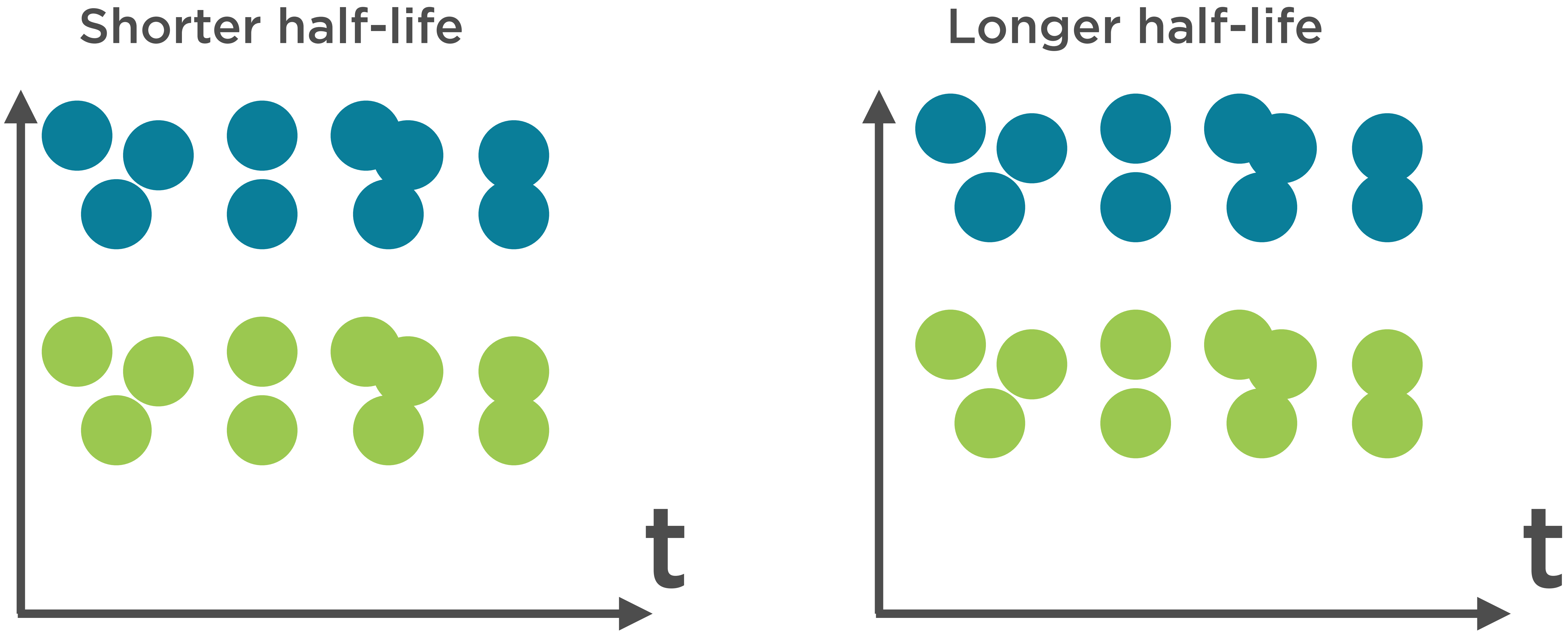
half-life = 5 batches

Effect of Half-life

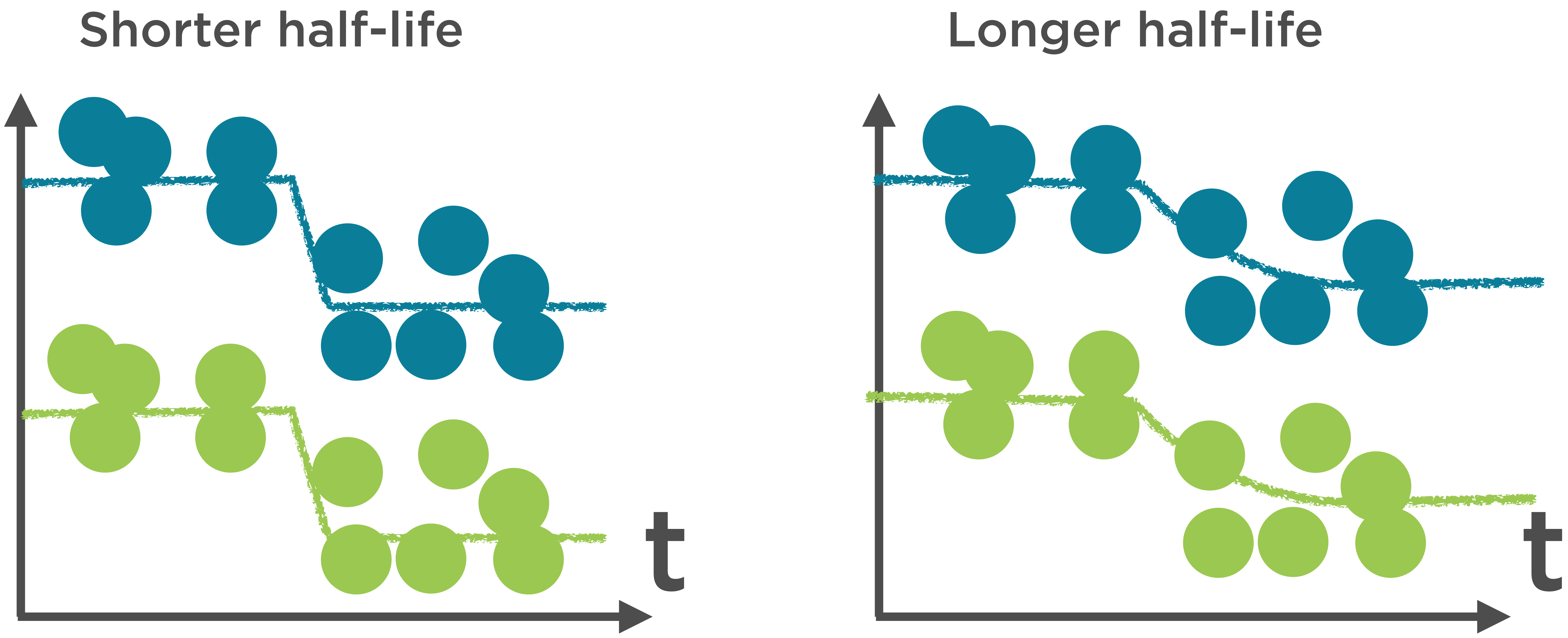


Cluster centers

Effect of Half-life

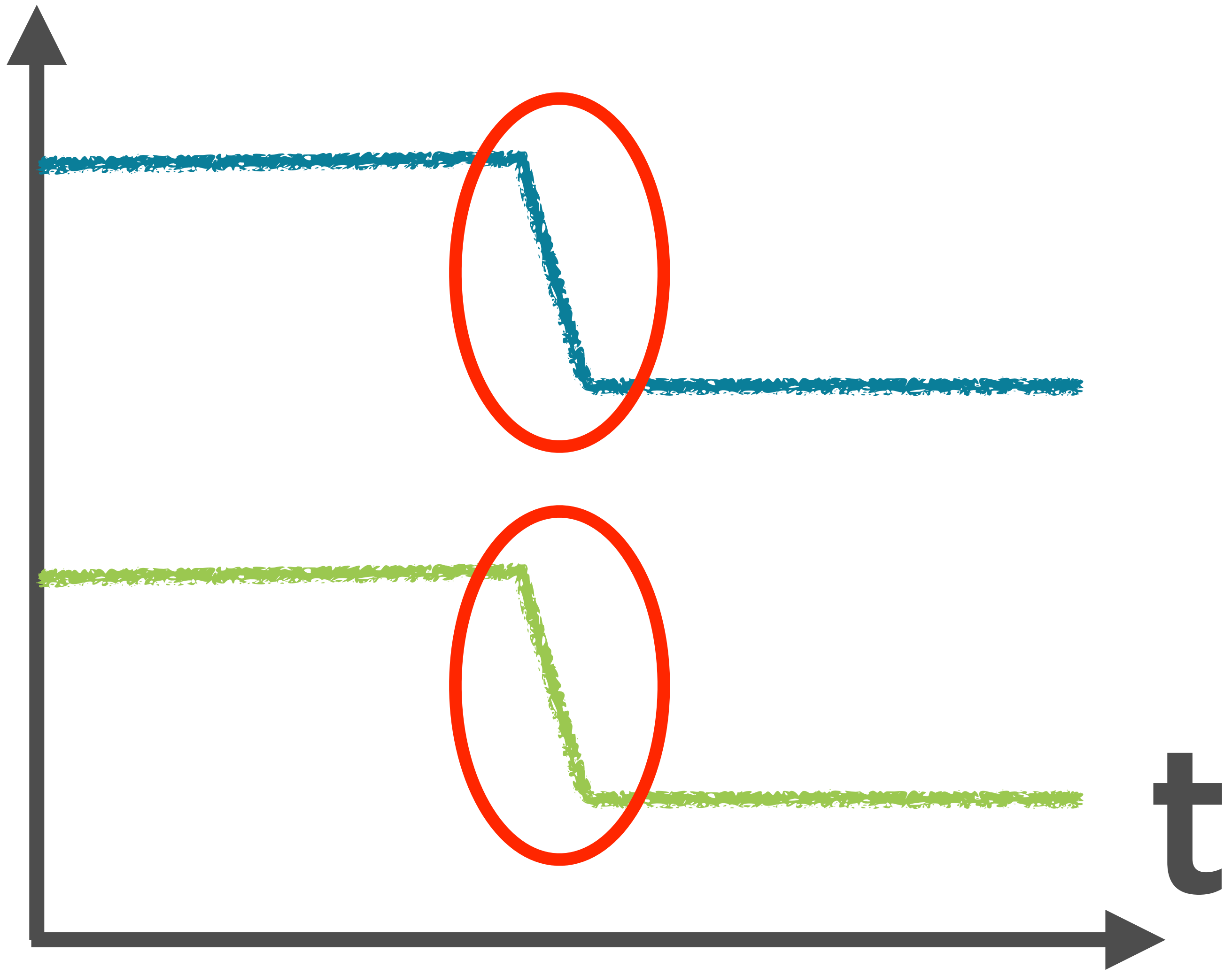


Effect of Half-life

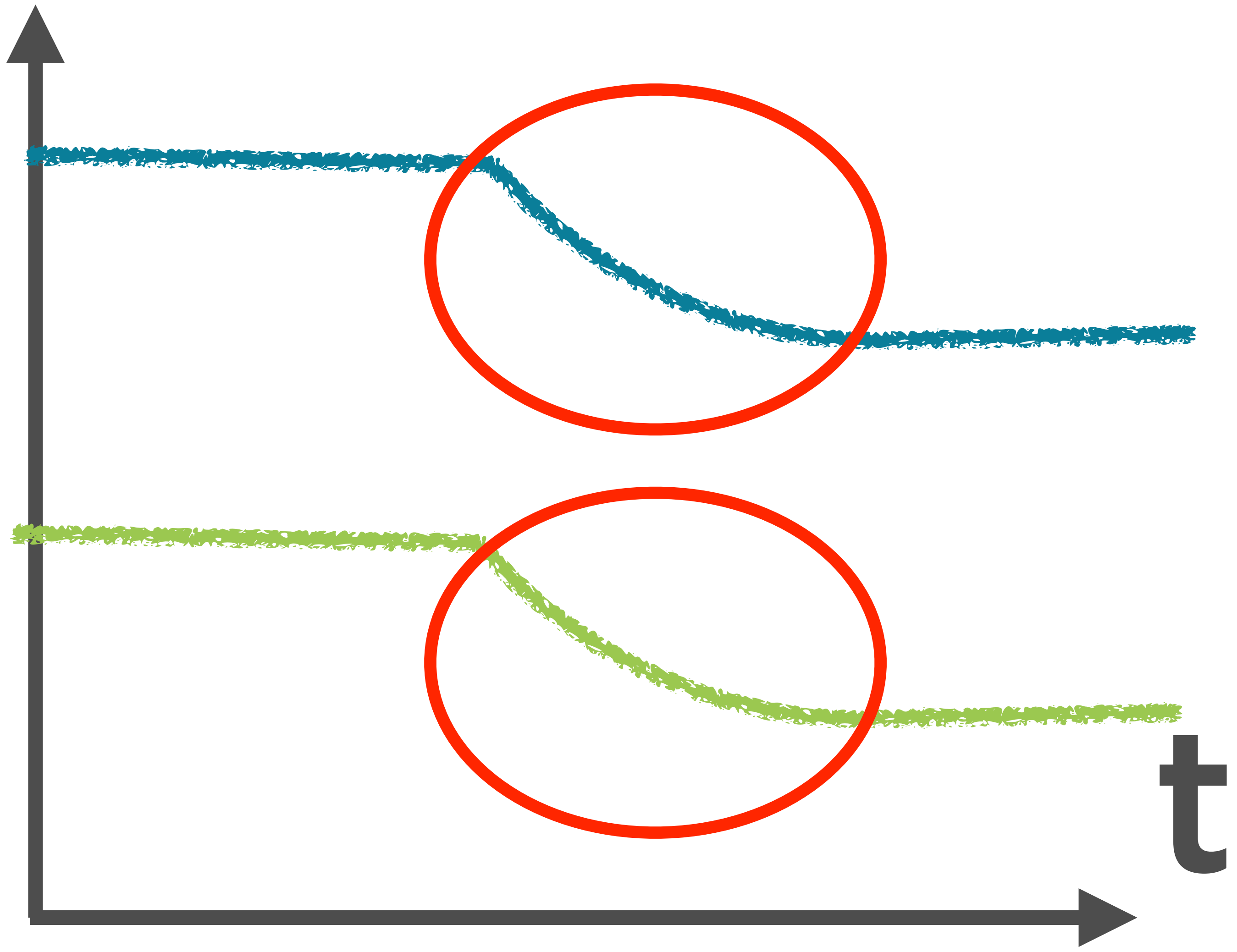


Effect of Half-life

Shorter half-life



Longer half-life



The algorithm adapts **faster** to changes in data when **half life is shorter**

Demo

Work with streaming data in the form of files saved in a directory

Apply the Streaming K-Means Clustering algorithm to find where tweets come from i.e. location clusters

Demo

Tweak the decay factor to see how the cluster centers change

Overview

Understood the basic k-means clustering algorithm and how it works on streaming data

Understood the decay factor and half-life which let you tweak the forgetfulness of the algorithm

Implemented the streaming k-means algorithm on a real world Twitter dataset to determine tweet location patterns