Bradley Voytek, Ph.D.
UC San Diego
Cognitive and Neural Dynamics Laboratory

Department of Cognitive Science
Neurosciences Graduate Program
Halıcıoğlu Data Science Institute

bvoytek@ucsd.edu
@bradleyvoytek


HALICIOĞLU DATA SCIENCE INSTITUTE

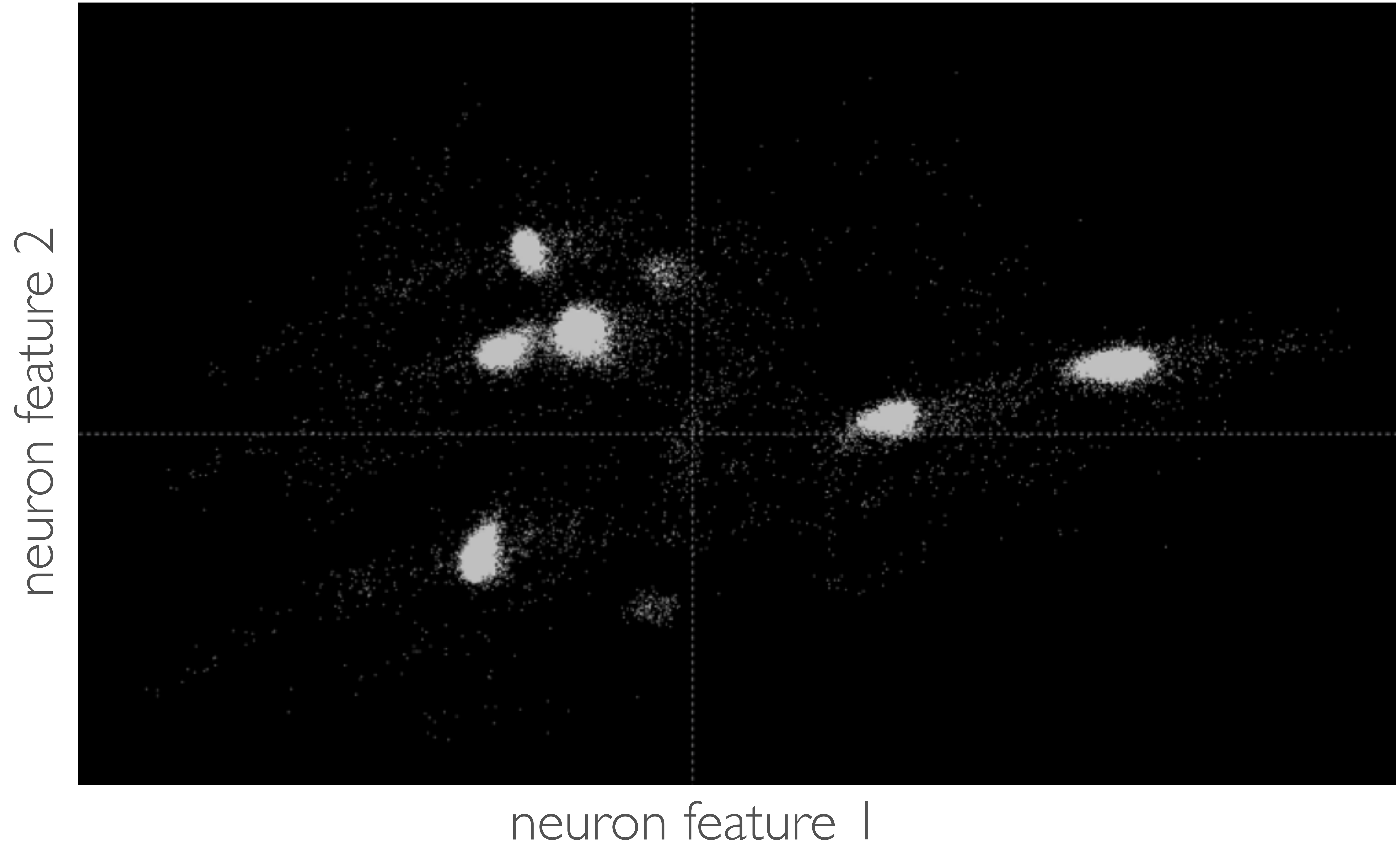

Bill
Nye
The Planetary Society
UC San Diego
Taner
Halicioglu
UC San Diego
UC San Diego

COGS 108
Data Science in Practice

Dimensionality reduction and clustering

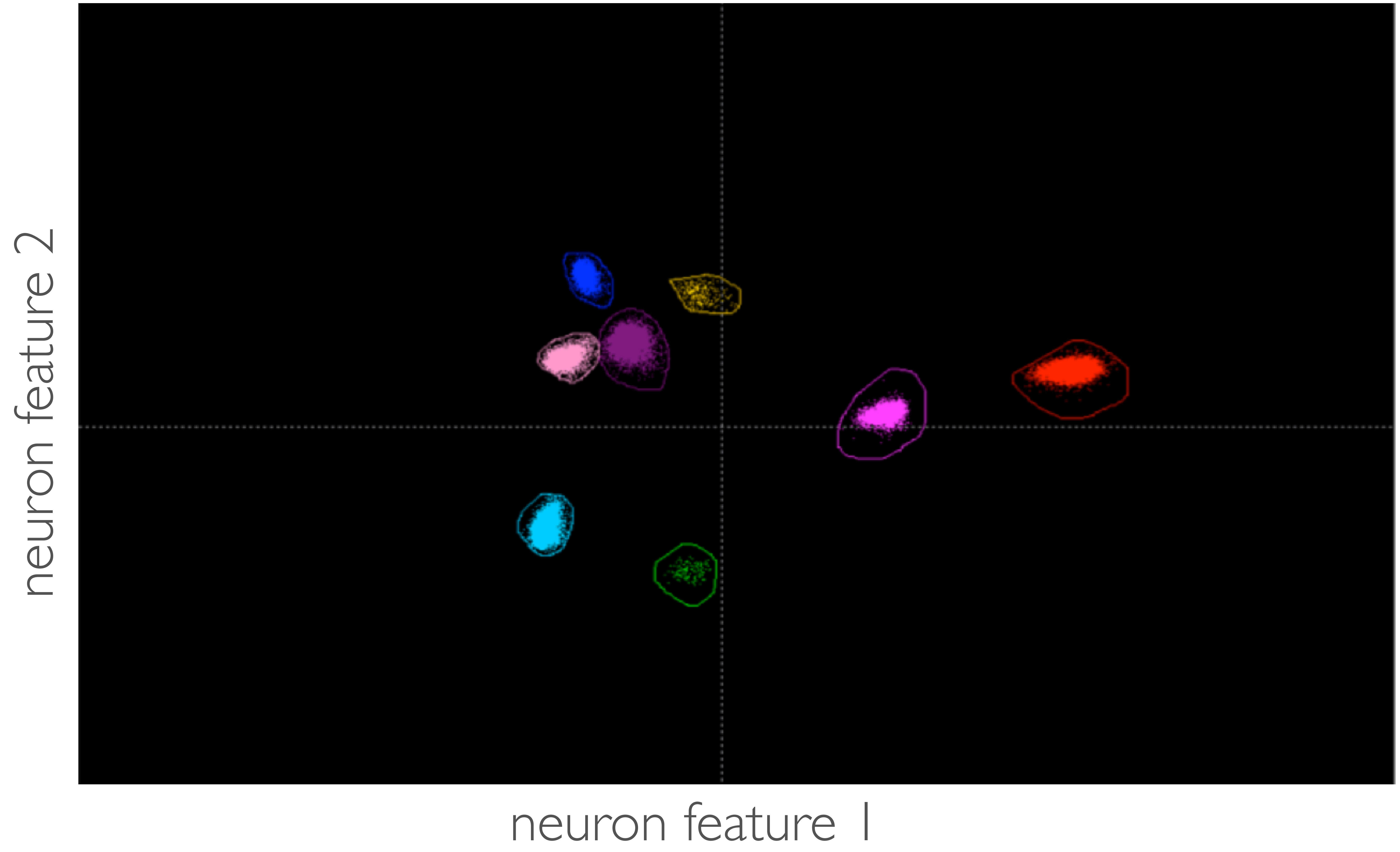
UC San Diego

Clustering



Source: Jake Olson

Clustering



Source: Jake Olson

Why clustering?

- Unlabeled data – unsupervised learning
    - If it is labeled, we'll use classification!
- Want to find groups!

Source: Jake Olson

Why clustering?

- Data Compression – We could want to reduce the dimensionality in the data.
    - Dimensionality – An aspect or a feature of something is a dimension.

Source: Jake Olson

How do we cluster?

- A lot of different ways we could, but we will start with K-Means.

Source: Jake Olson

UC San Diego

k-means

- A simple but effective clustering algorithm
- Partitions the data in to K disjoint sets (clusters)
    - disjoint - no overlap
- We then run an iterative batch algorithm
    - batch – to do to all of the items (batch of cookies)

Source: Jake Olson

k-means

- Start – Initialize a guess of k centers/"means"
  - notation – *j* simply refers to which of the *k* centers.

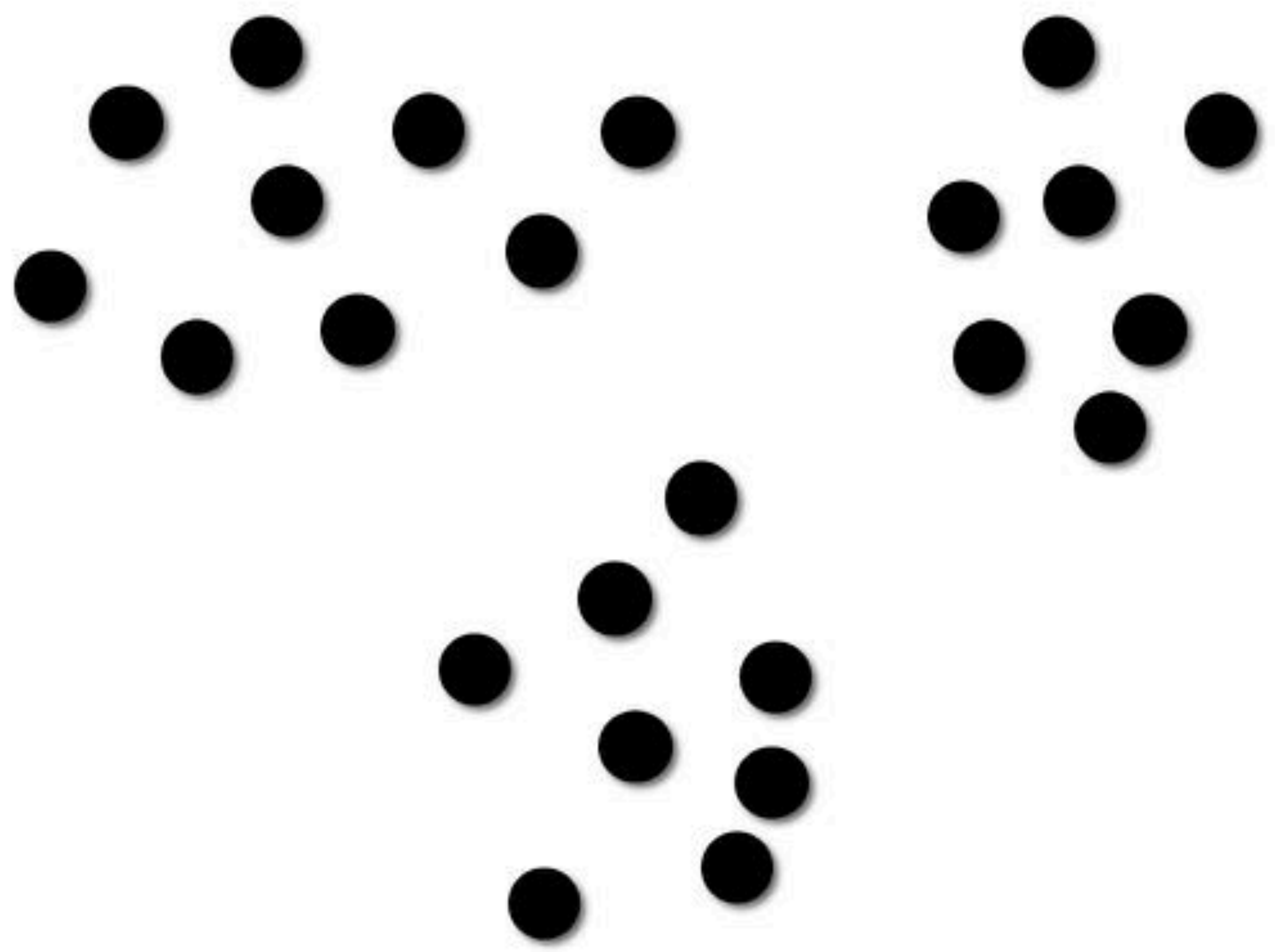
- $S^{(j)}$ is all points closest to $\boldsymbol{\mu}(j)$
  - So each "mean" gets all of the points close to it in its set.

- Update the means: $$\mu^{(j)} = \frac{1}{N_j} \sum_{n \in S^{(j)}} \mathbf{x}^{(n)}$$

- Continue until there is no change in the means or you decide to stop.

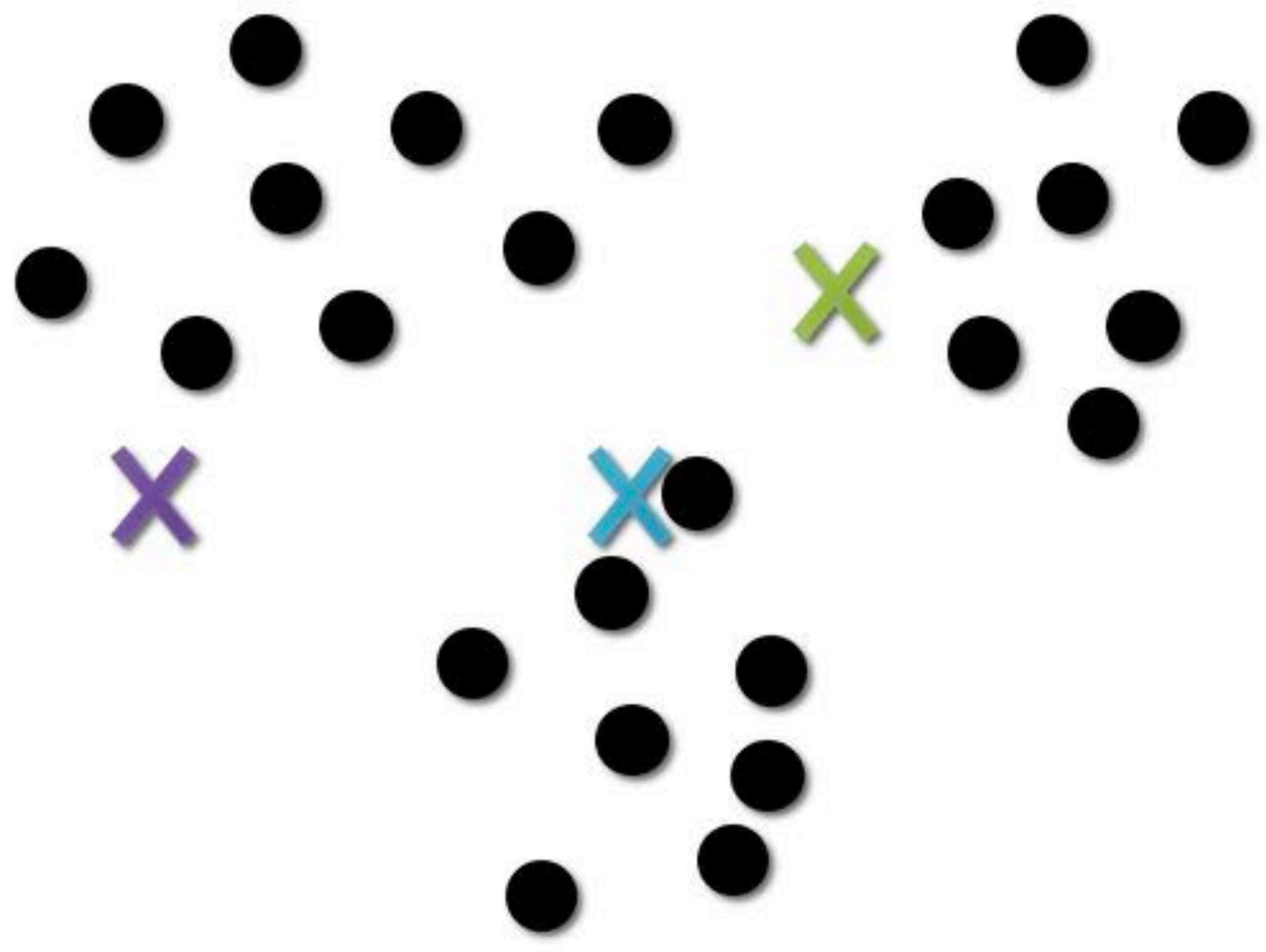
Source: Jake Olson

Initialize means



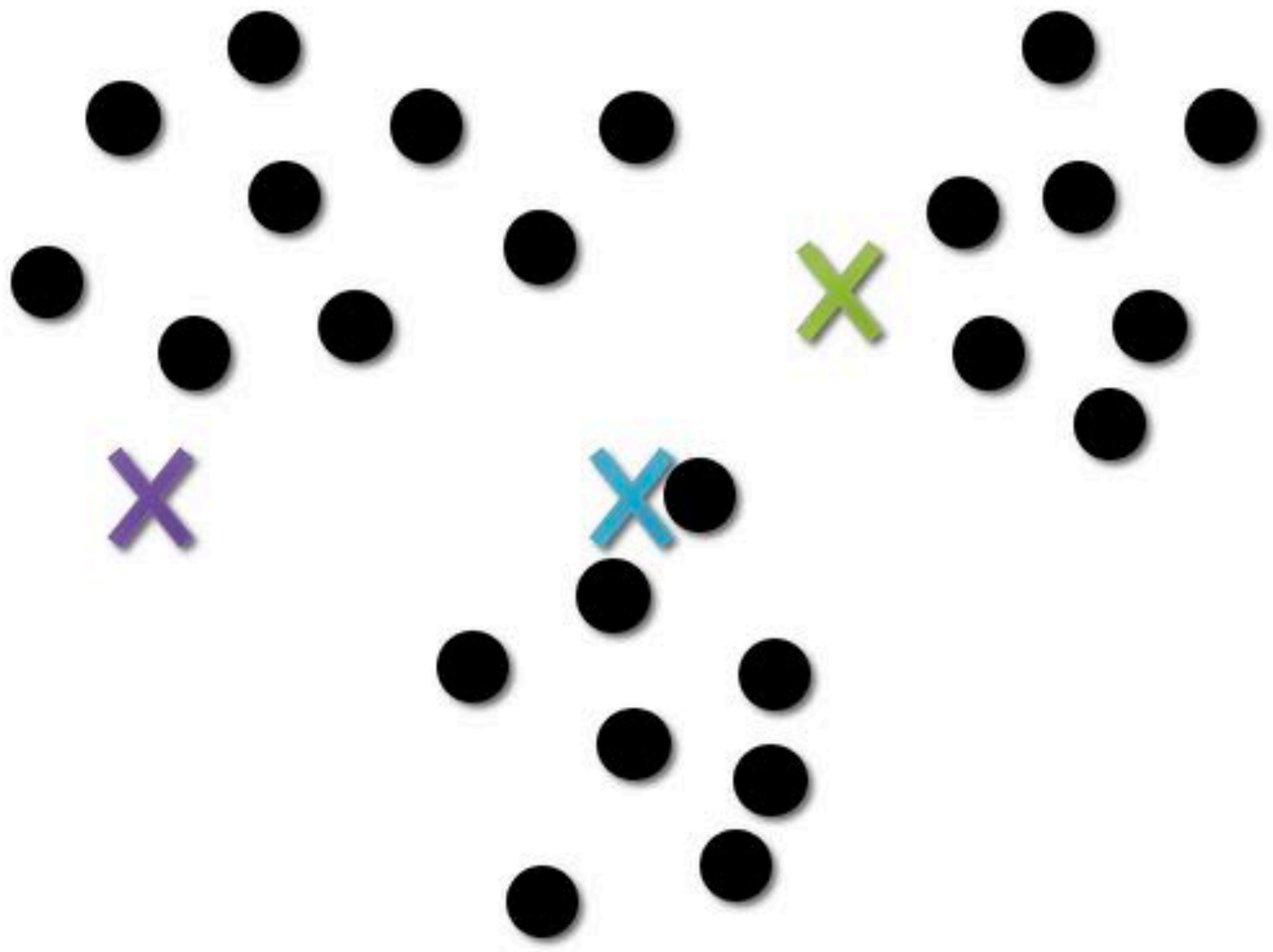
Source: Jake Olson

Initialize means

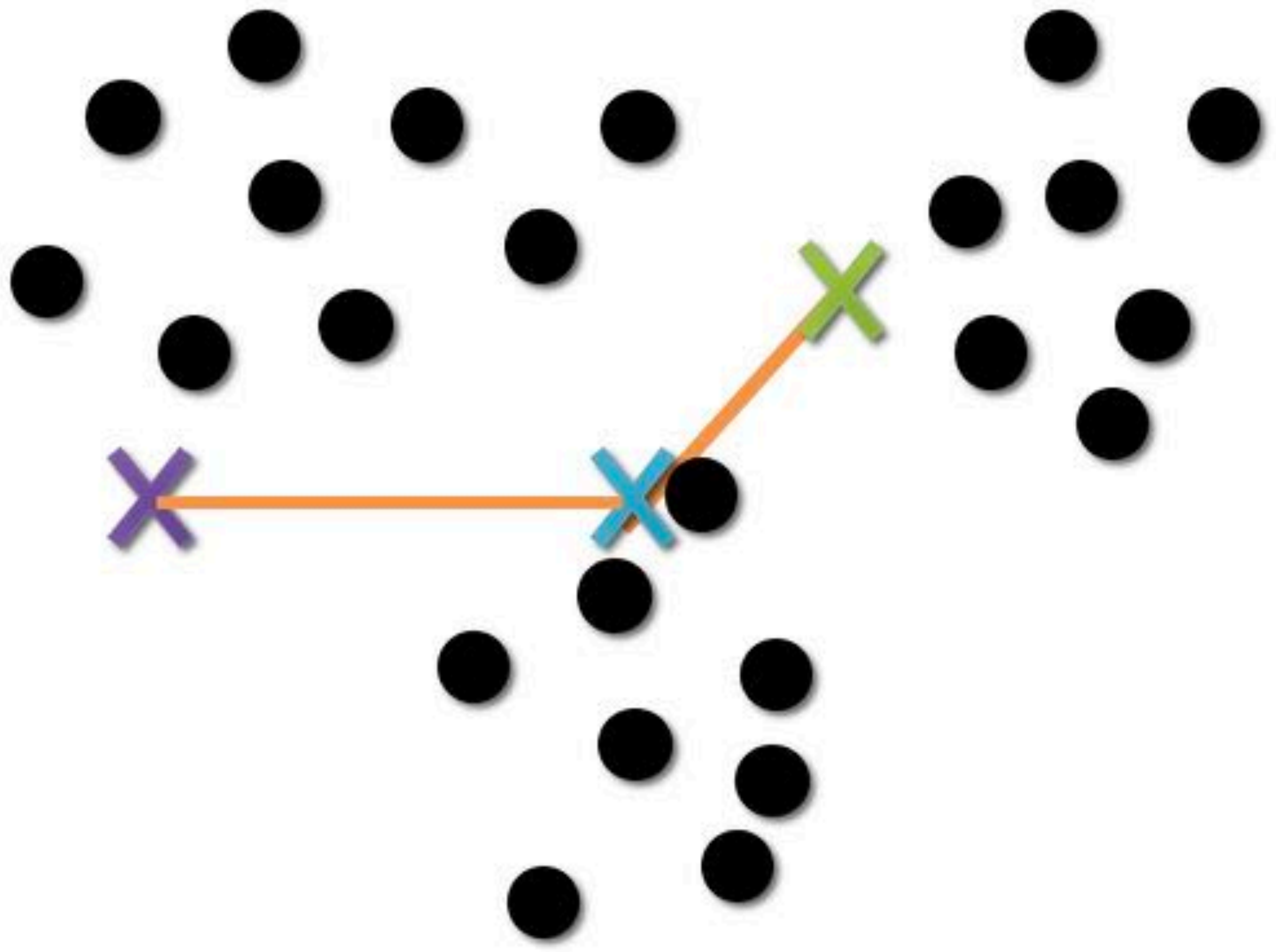


Source: Jake Olson

Label points

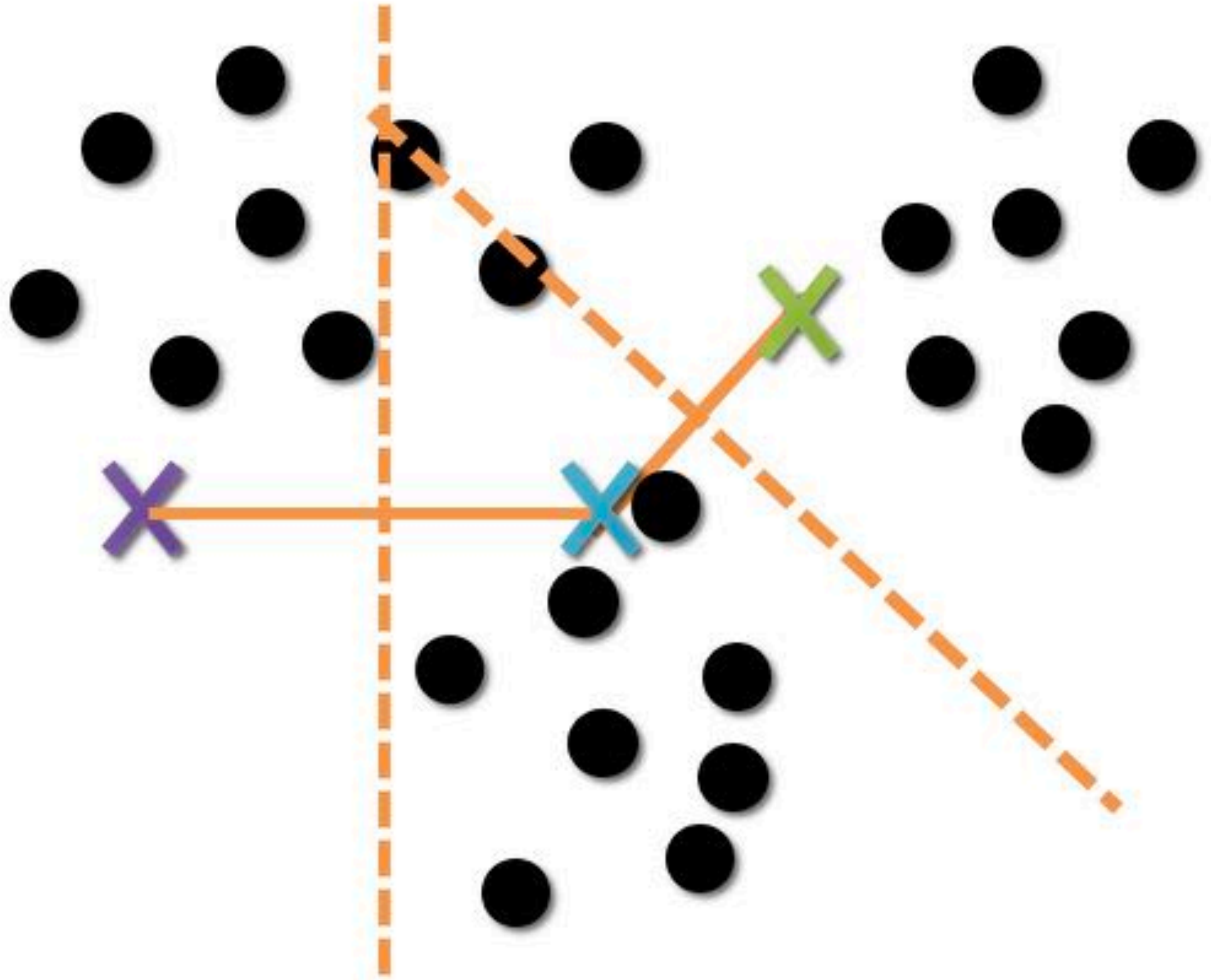


Connect adjacent means



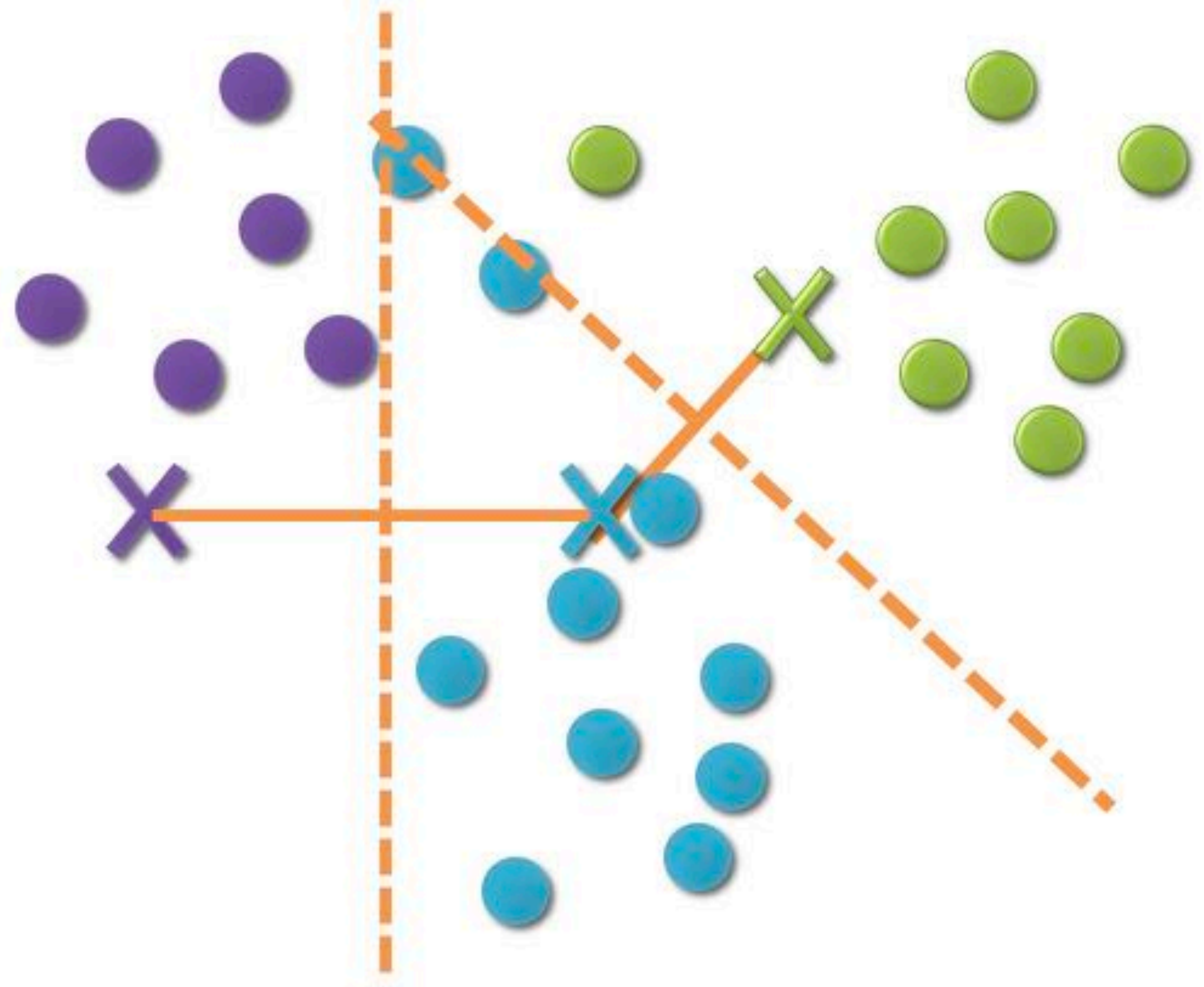
Source: Jake Olson

Bisect



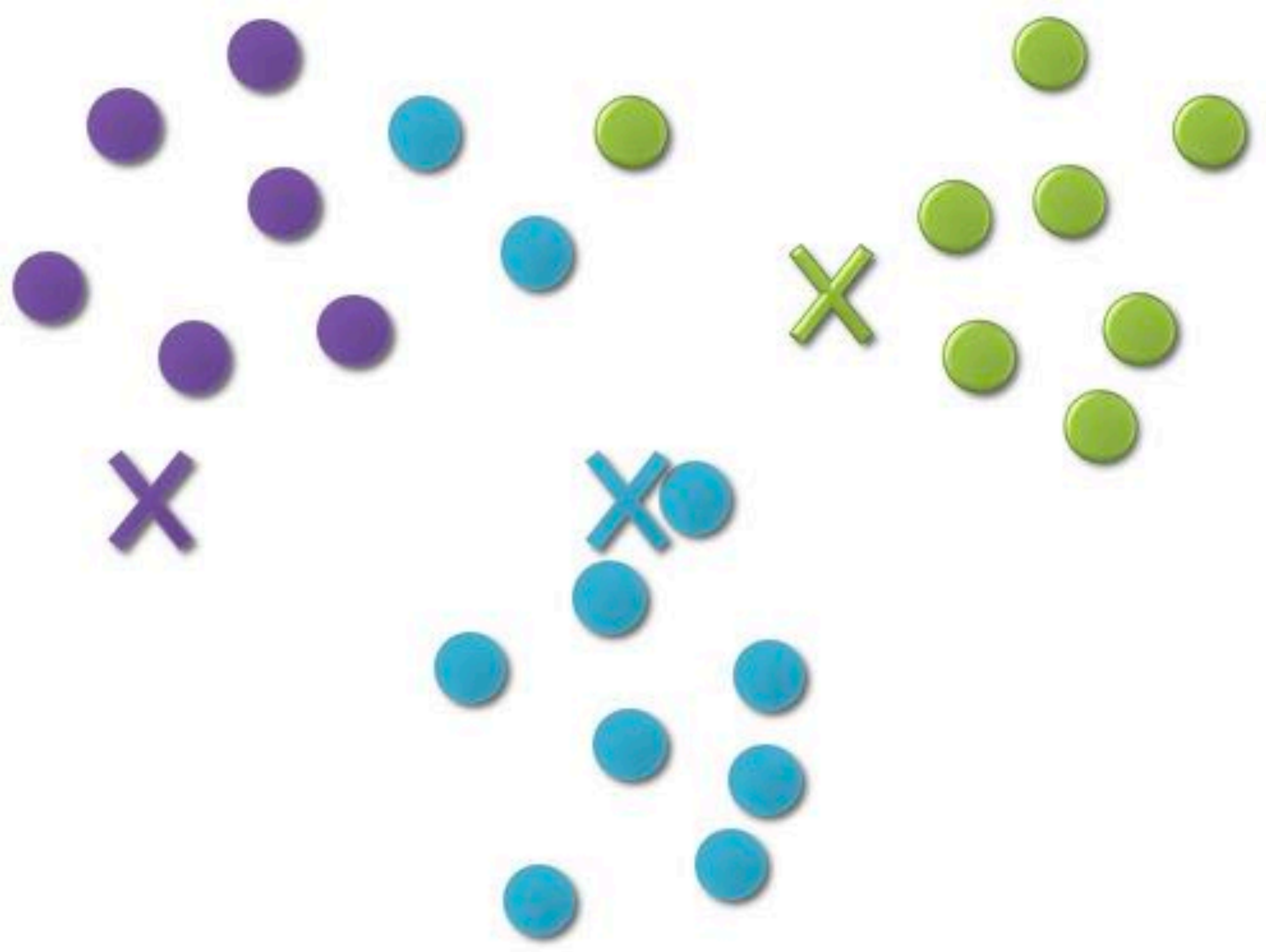
Source: Jake Olson

Label points



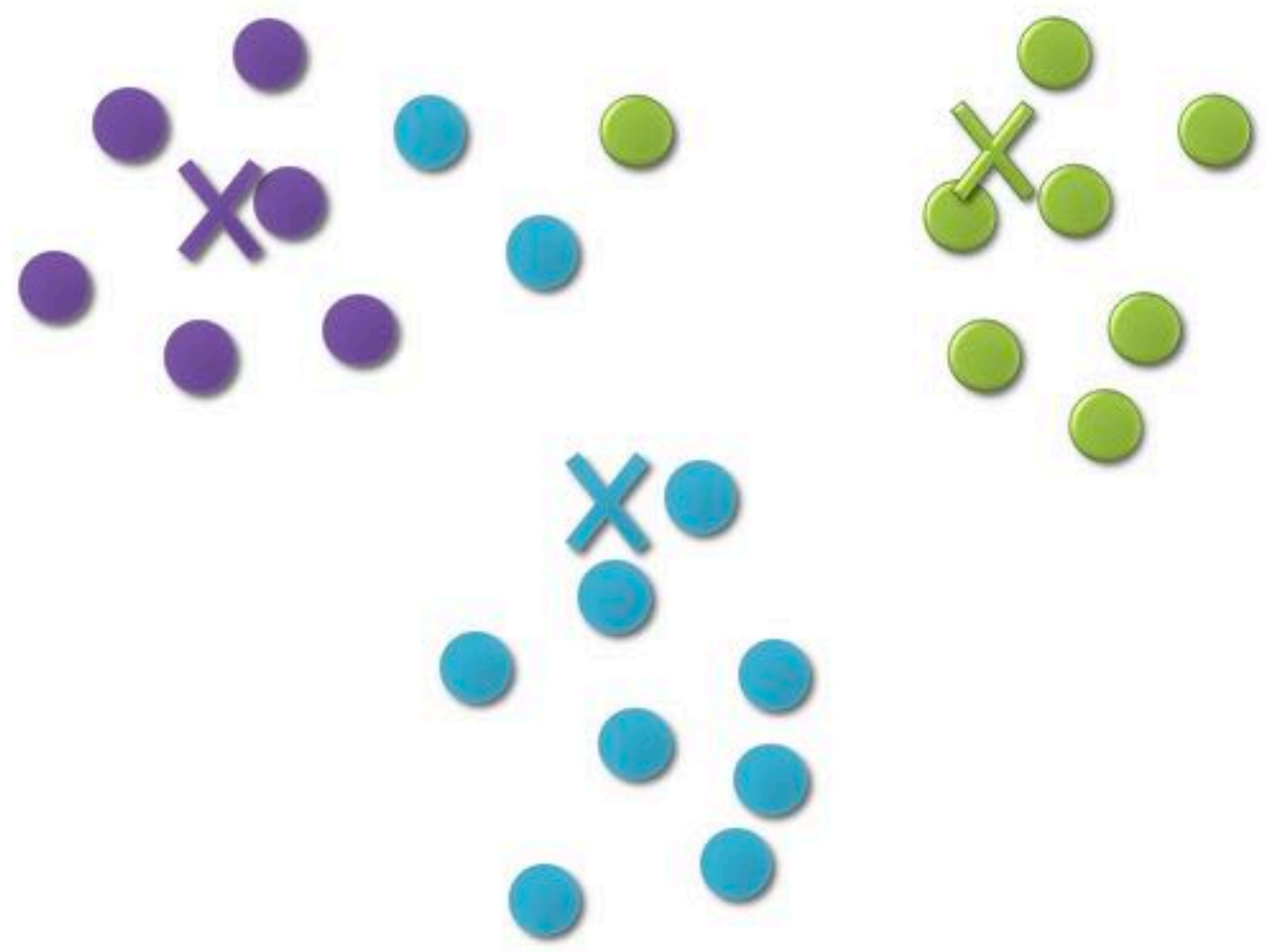
Source: Jake Olson

Move means to centroids



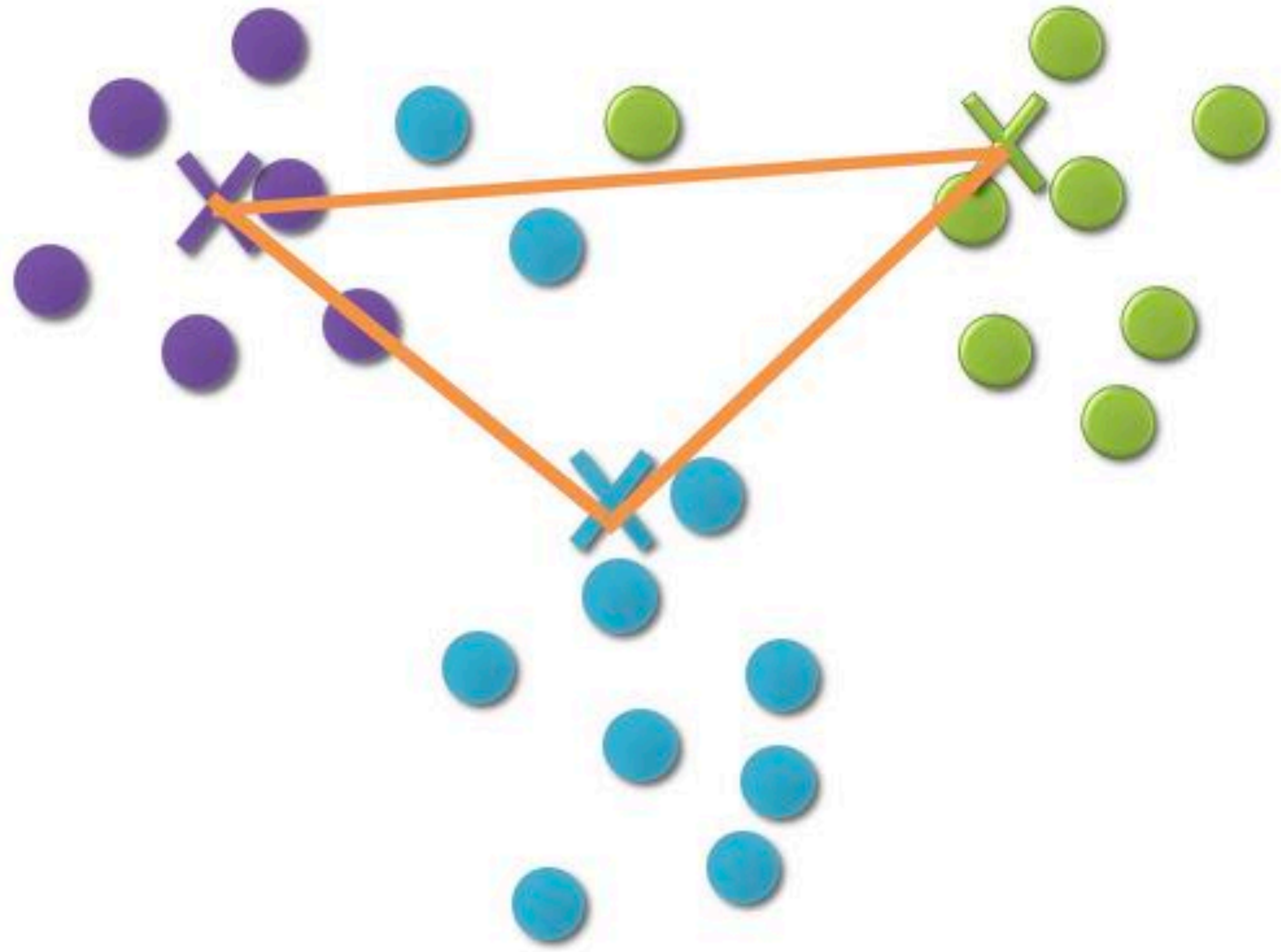
Source: Jake Olson

Move means to centroids



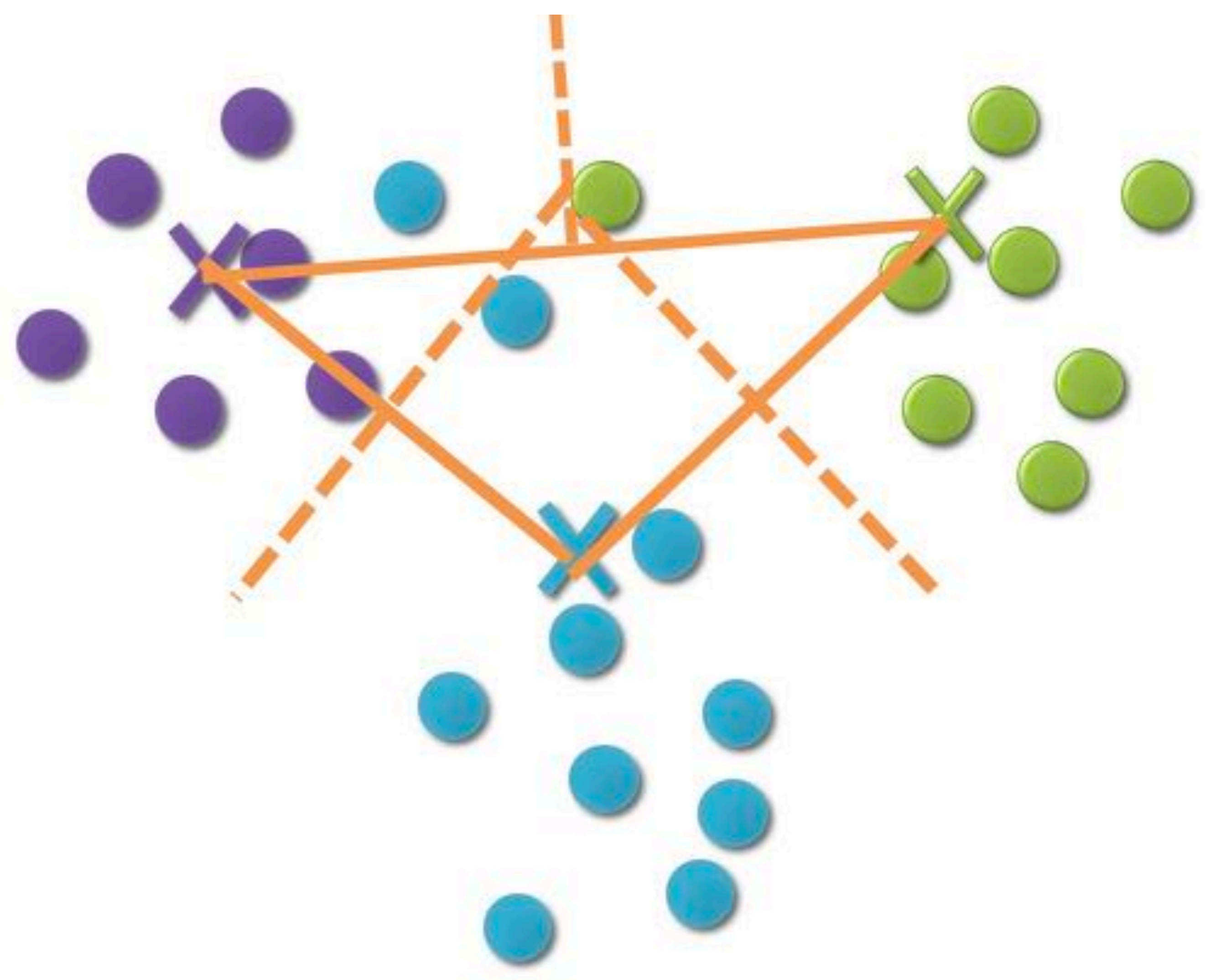
Source: Jake Olson

If means moved, repeat!



Source: Jake Olson

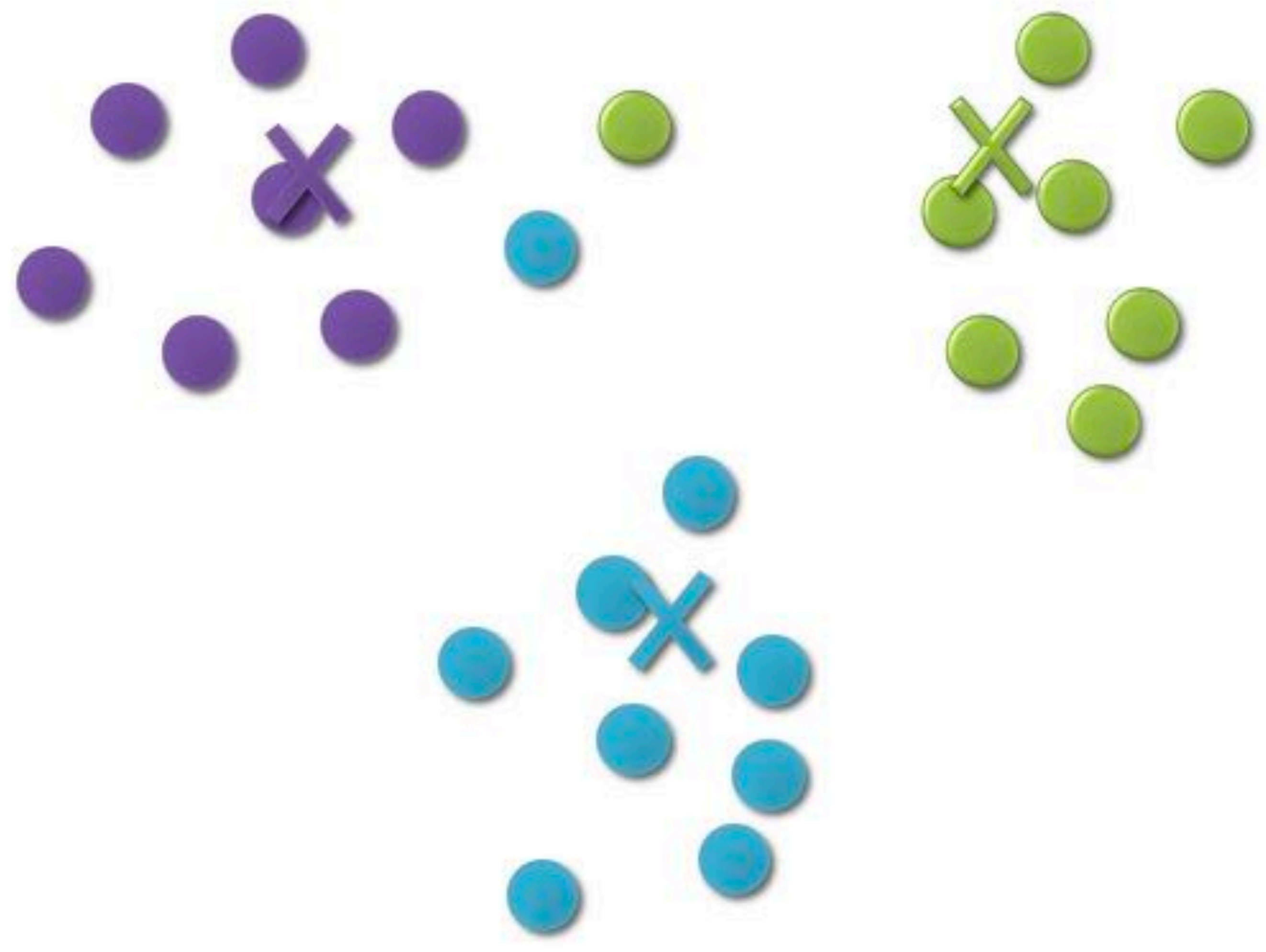
Bisect



Source: Jake Olson

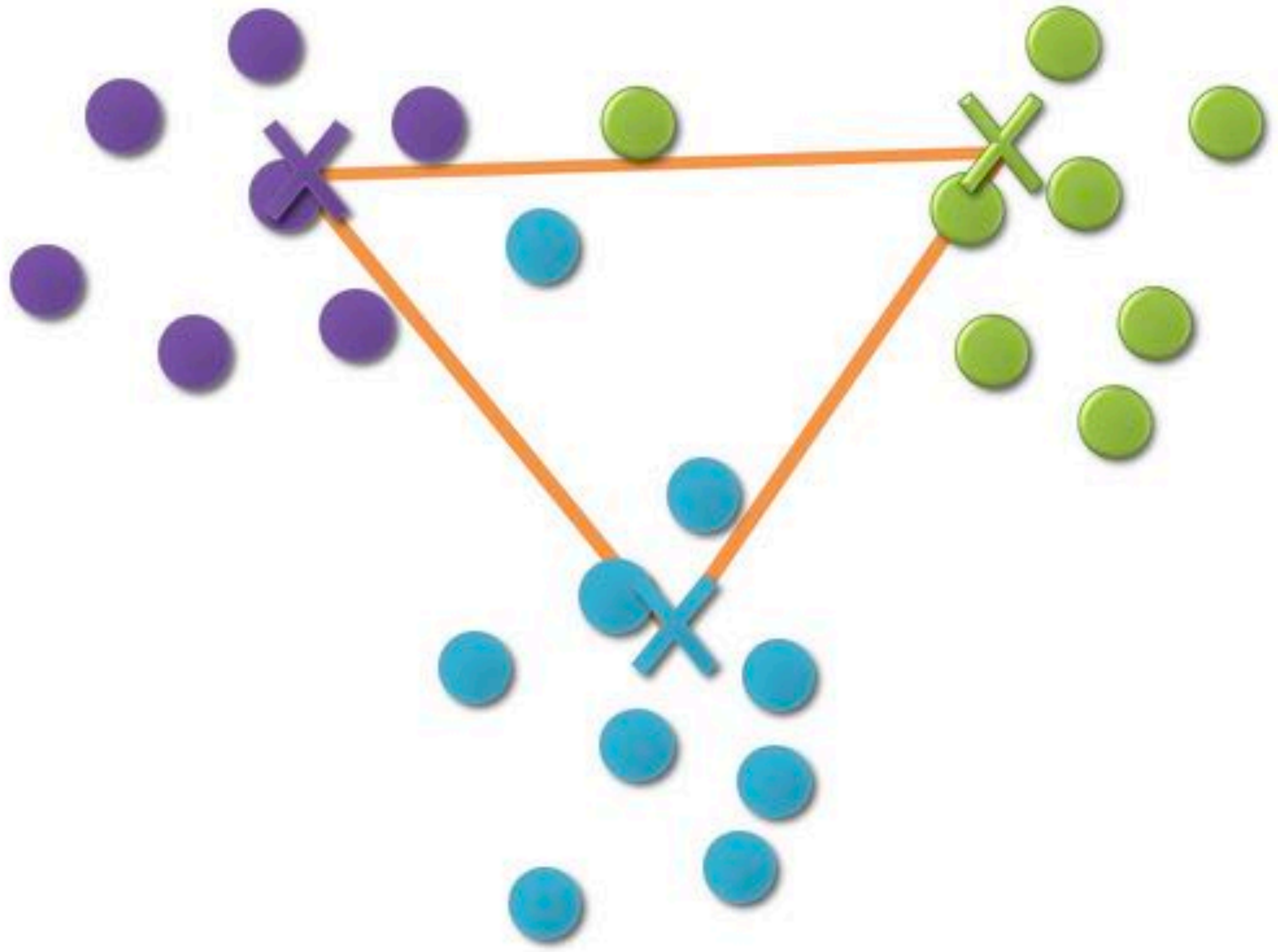
UC San Diego

Label points



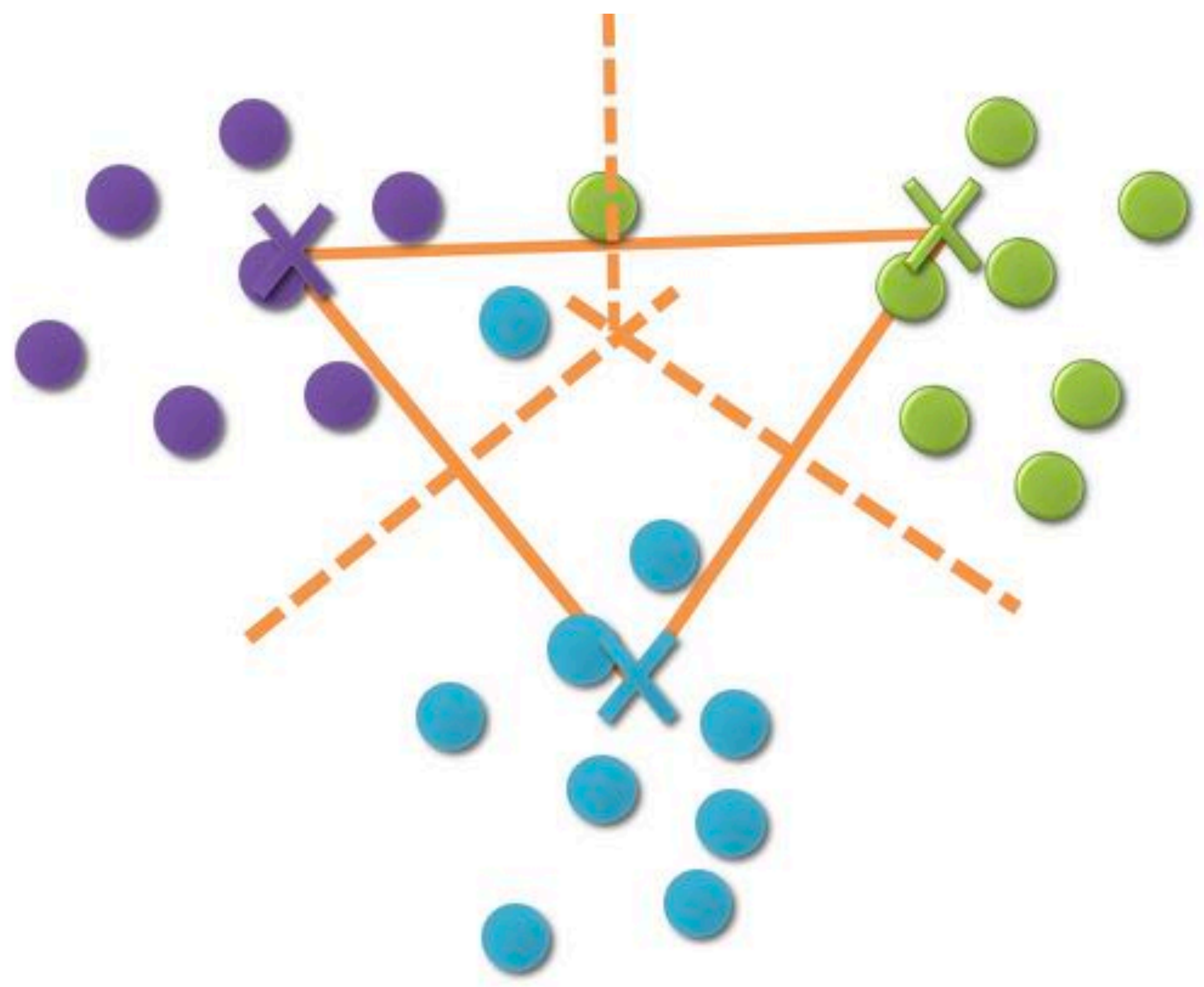
Source: Jake Olson

Connect adjacent



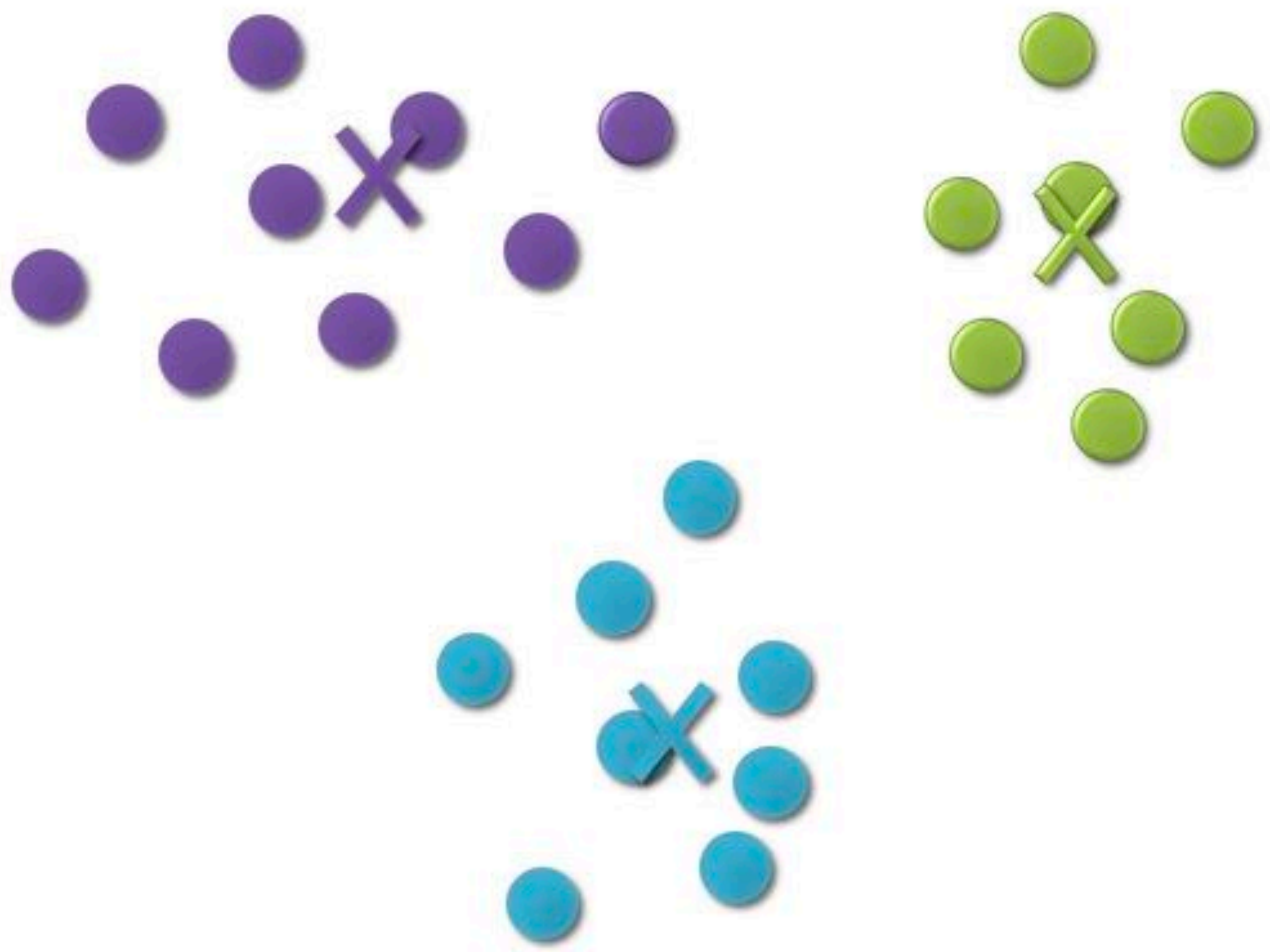
Source: Jake Olson

Bisect



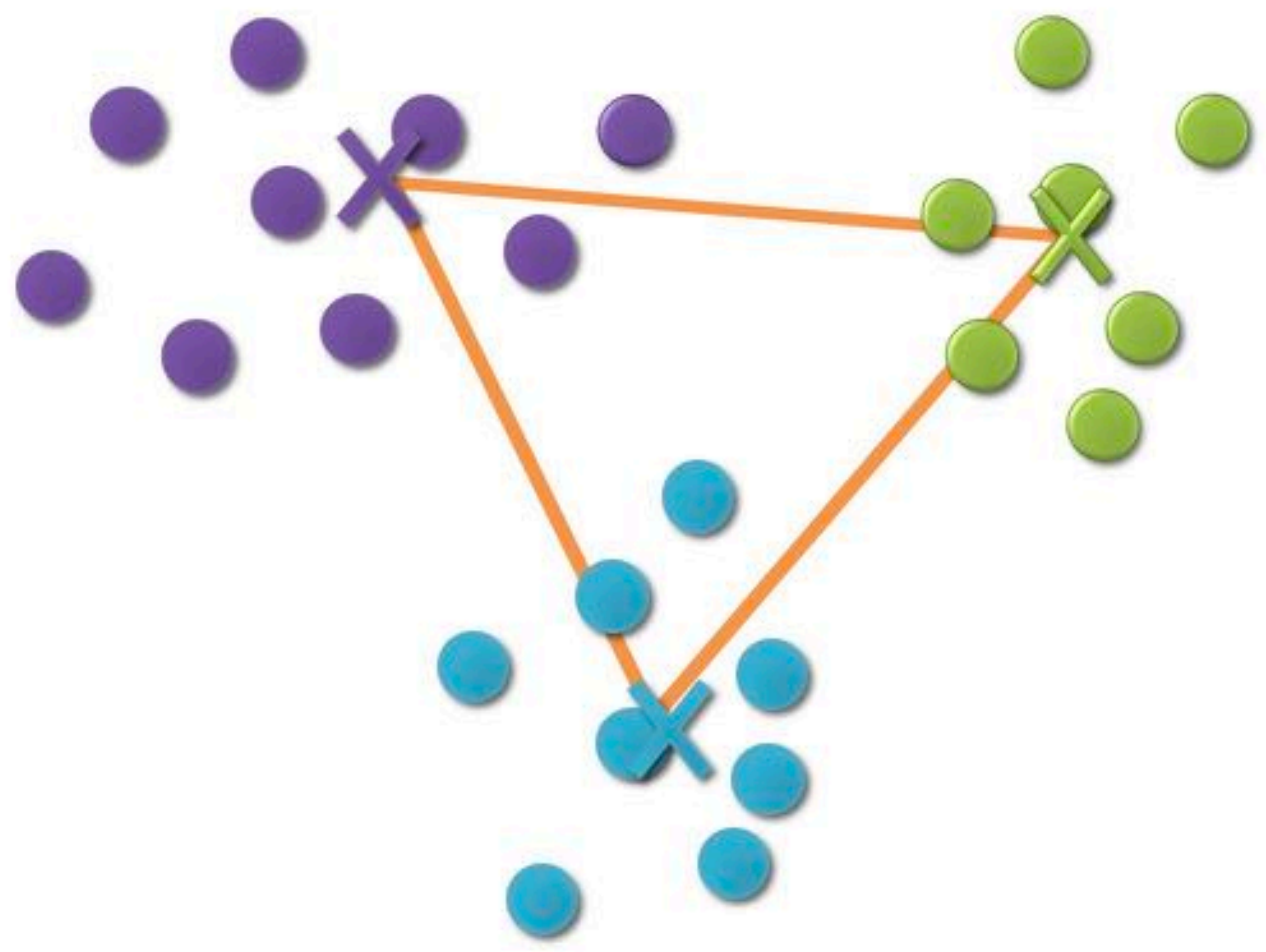
Source: Jake Olson

Label points

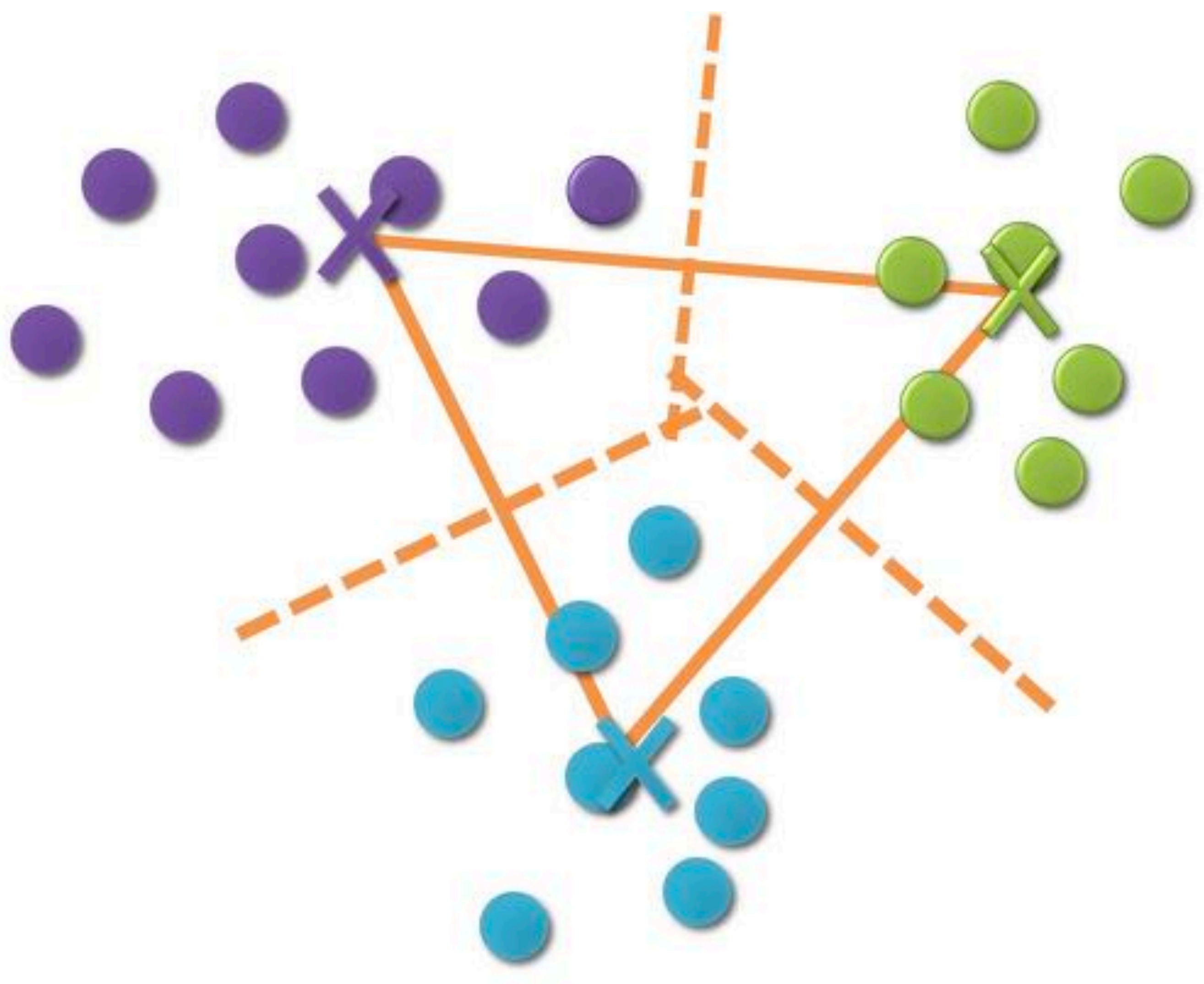


Source: Jake Olson

Connect adjacent

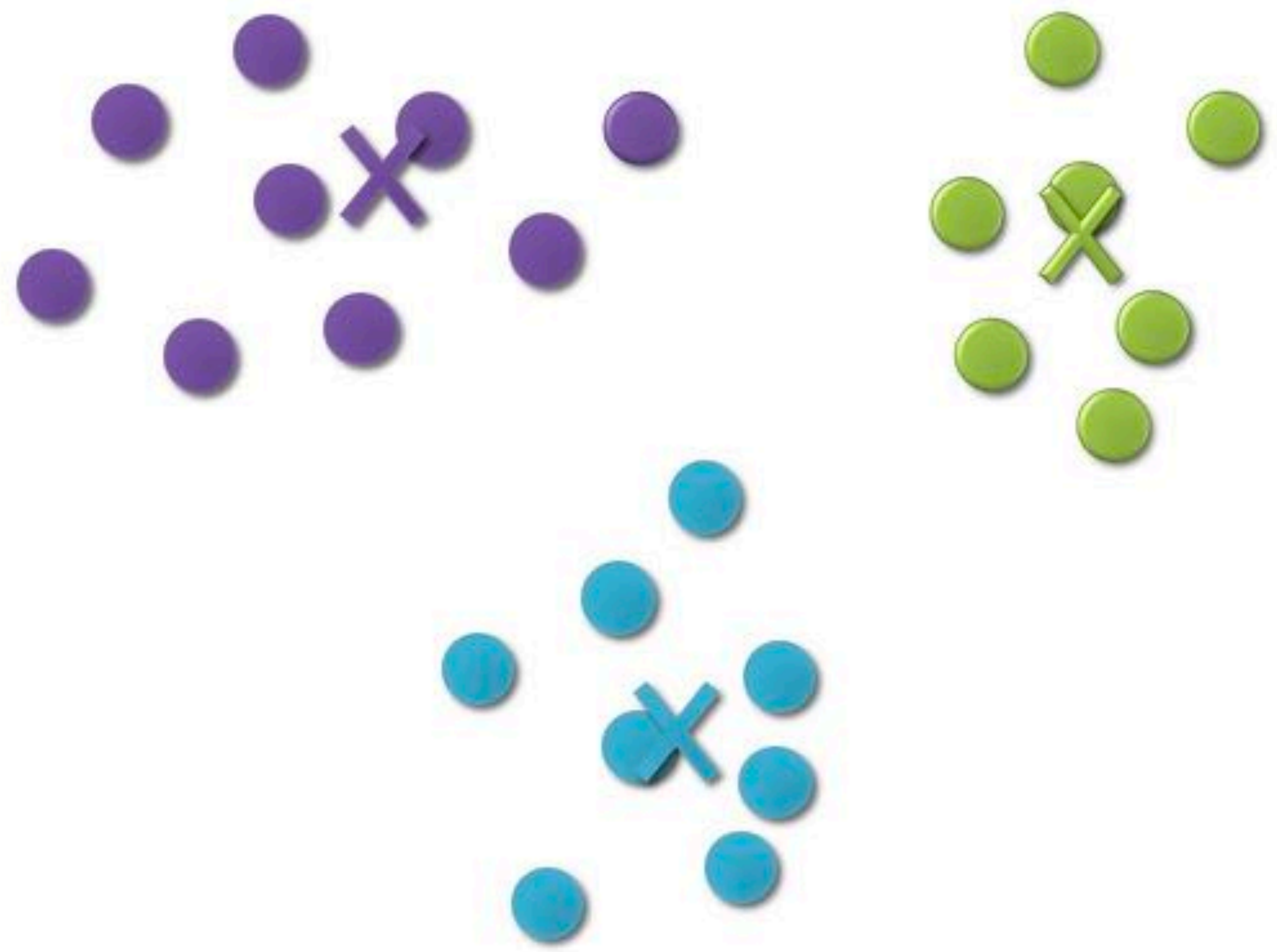


Bisect



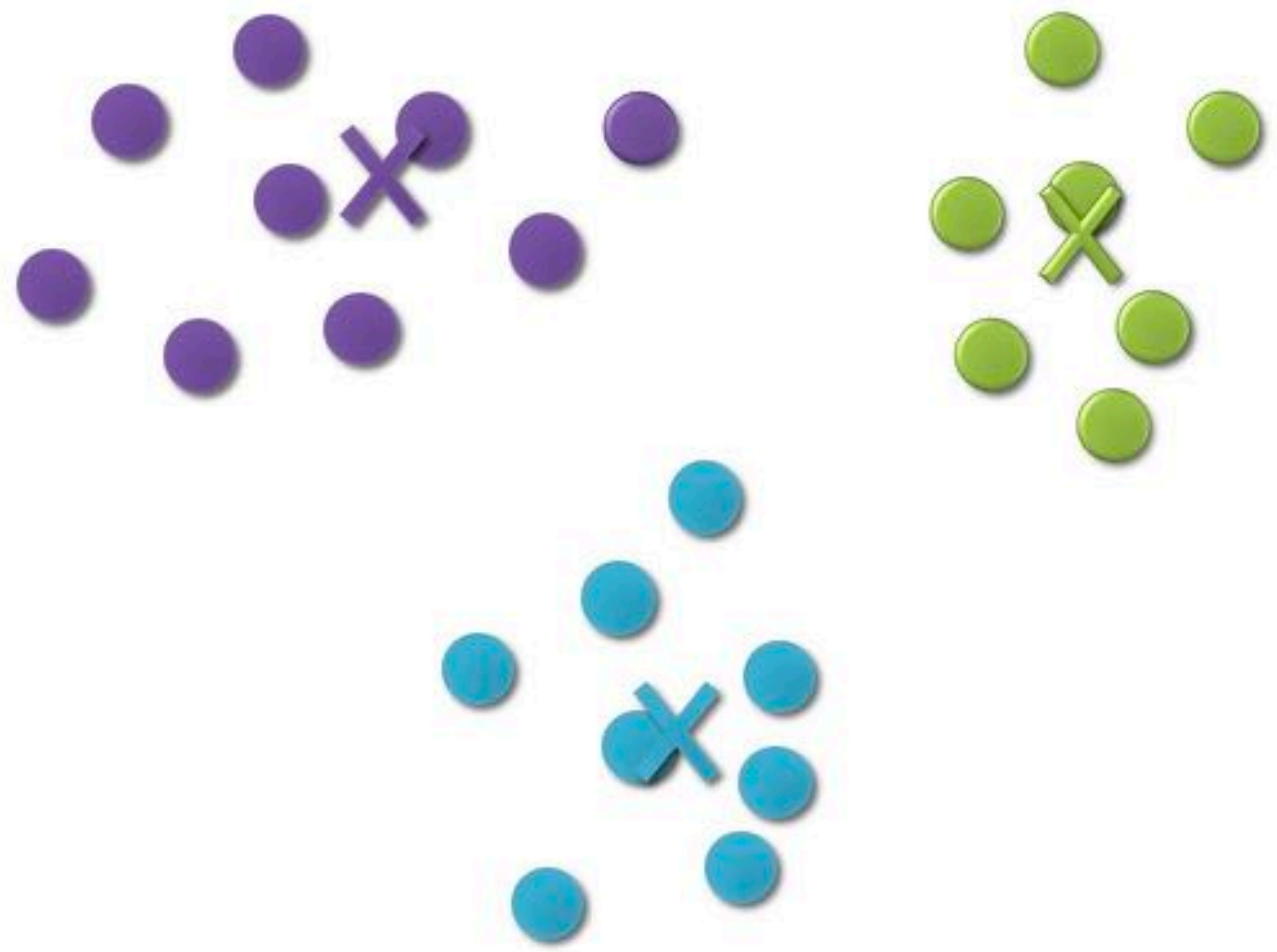
Source: Jake Olson

Label points



Source: Jake Olson

Finished!



Source: Jake Olson

When do we reduce dimensionality?

Alicia is researching the resistance of plants in different soil to bugs. She has 10 each of 5 different types of plants. She has counted the number of bugs on each once per week for 10 weeks. Should Alicia analyze her bug count data with a clustering analysis? Why?

Source: Jake Olson

When do we reduce dimensionality?

Pat is studying cell types and morhpologies in the basal forebrain of mice. They recorded the waveform amplitude, shapes, and firing patterns of 400 neurons. Should Pat run a clustering analysis?

When do we reduce dimensionality?

Linh is working at Apple. They have data on the accounts for all of their users, including what programs they use, how often, and for how long. They've been tasked with mining the data to find patterns that could affect marketing strategies. Should Linh run a clustering analysis?

COGS 108

Data Science in Practice

Support Vector Machines

SVM

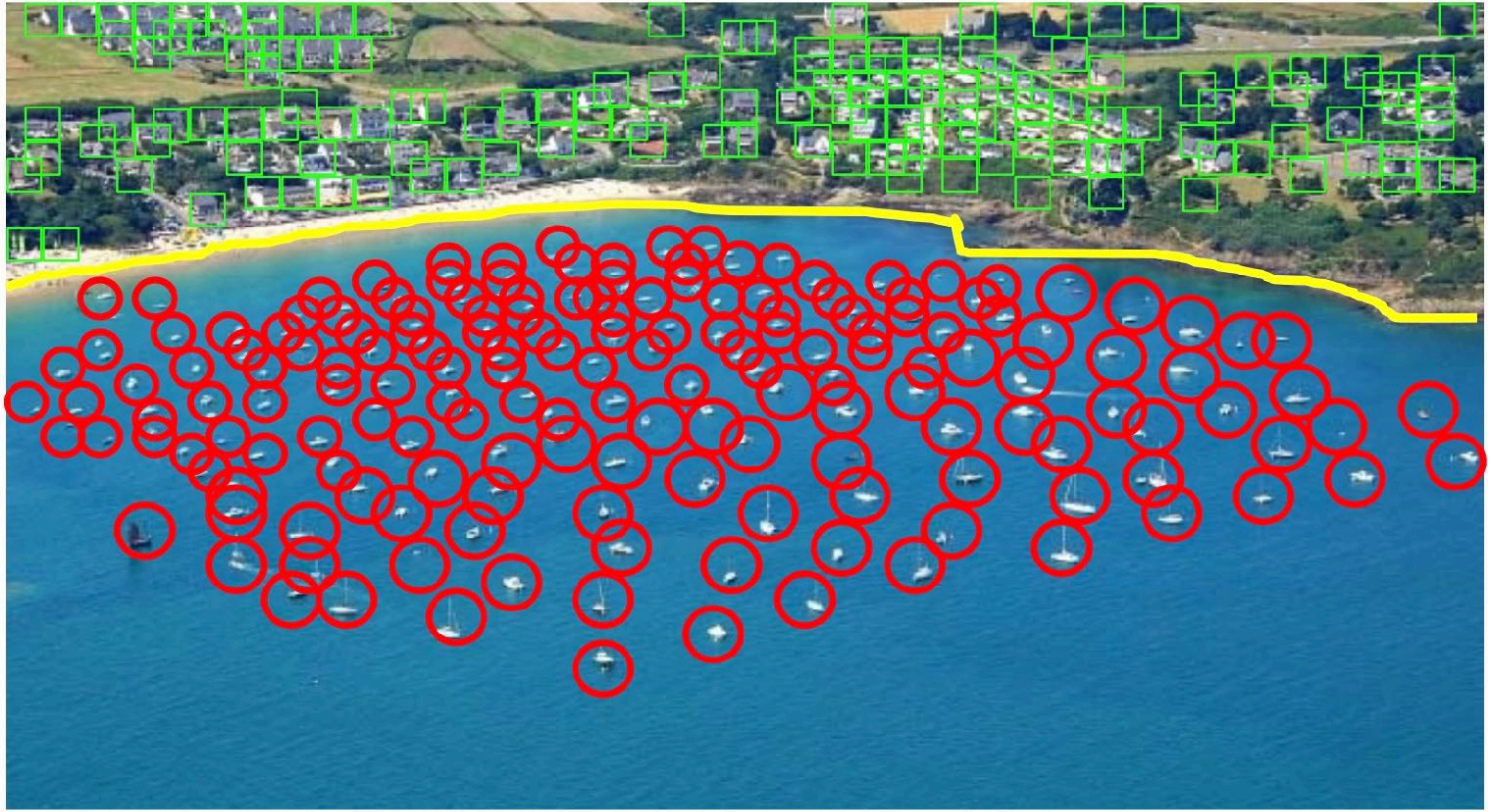
When classifying isn't easily linear,
make it linear.

SVM

UC San Diego

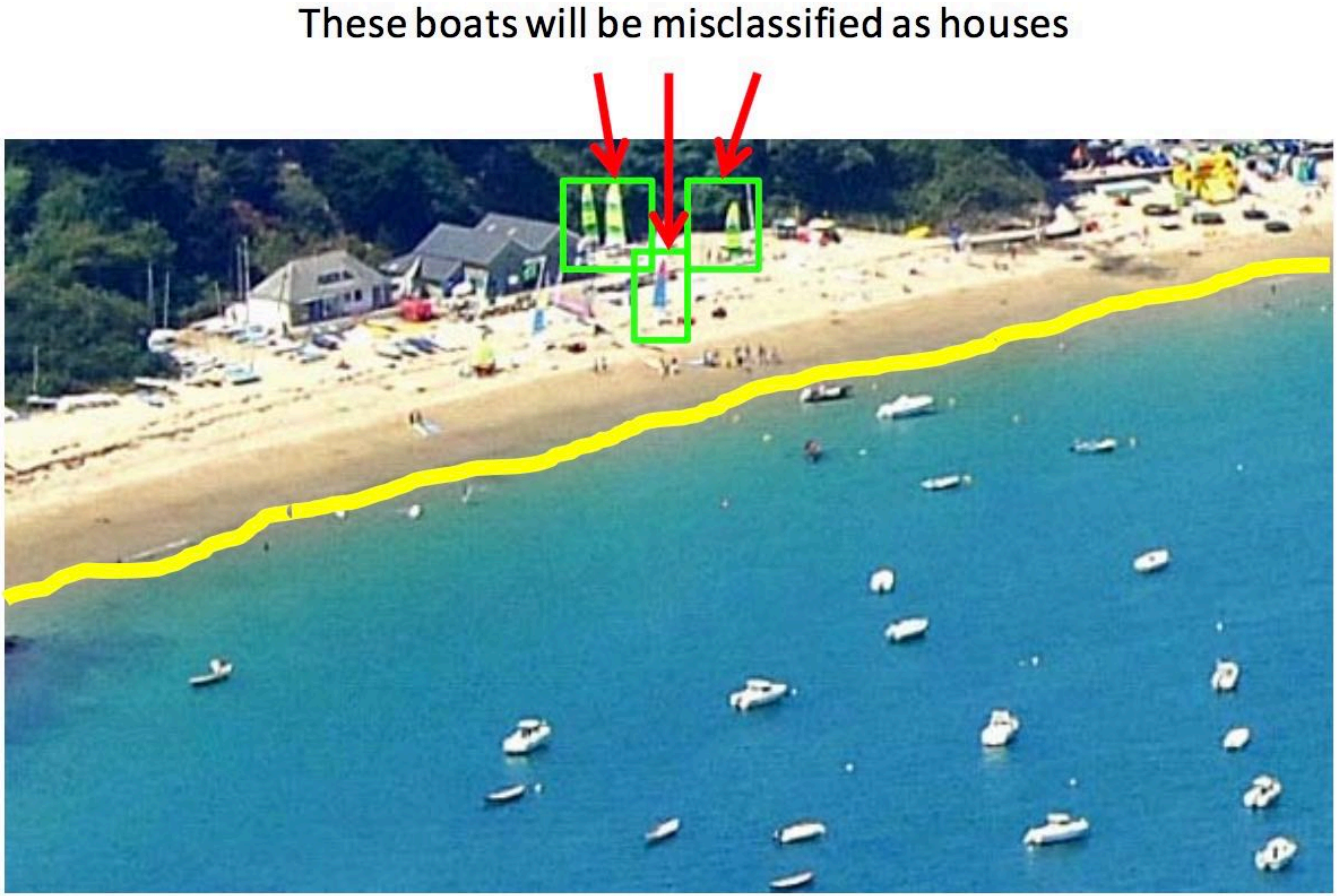
Source: Statnikov, Hardin, Guyon, Aliferis *AMIA* 2009

SVM



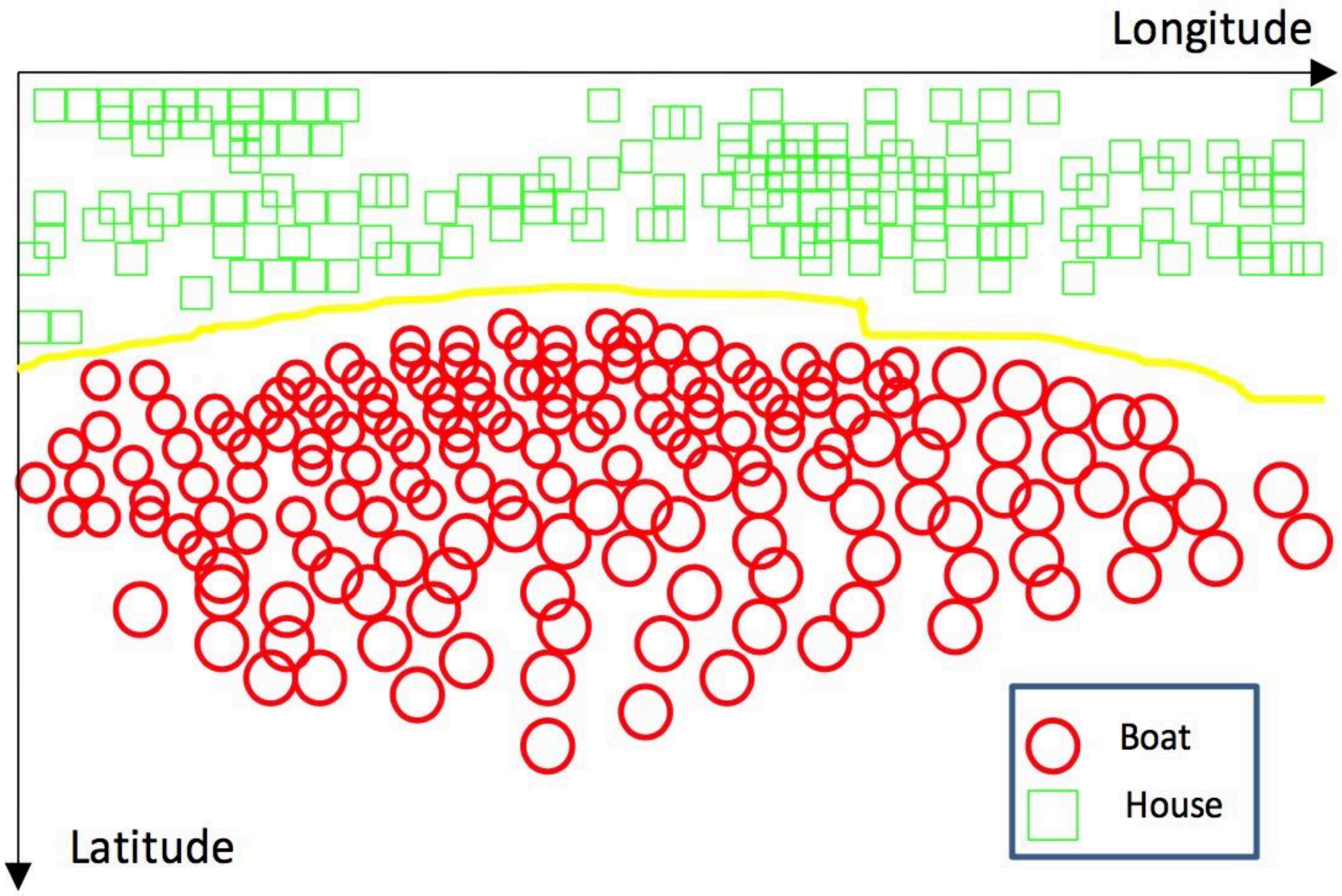
Source: Statnikov, Hardin, Guyon, Aliferis *AMIA* 2009

SVM



Source: Statnikov, Hardin, Guyon, Aliferis *AMIA* 2009

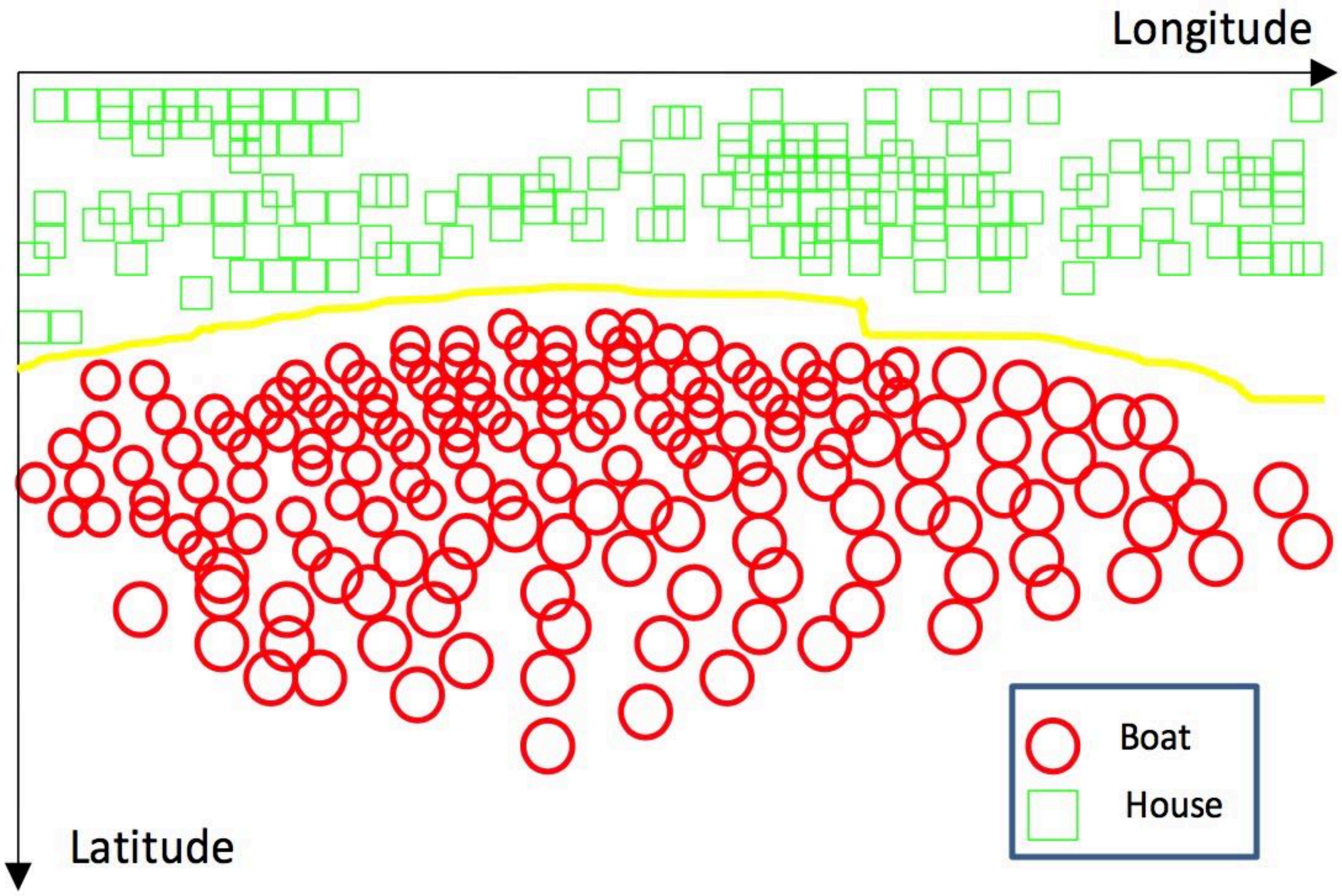
SVM



Shoreline is your decision boundary

Source: Statnikov, Hardin, Guyon, Aliferis *AMIA* 2009

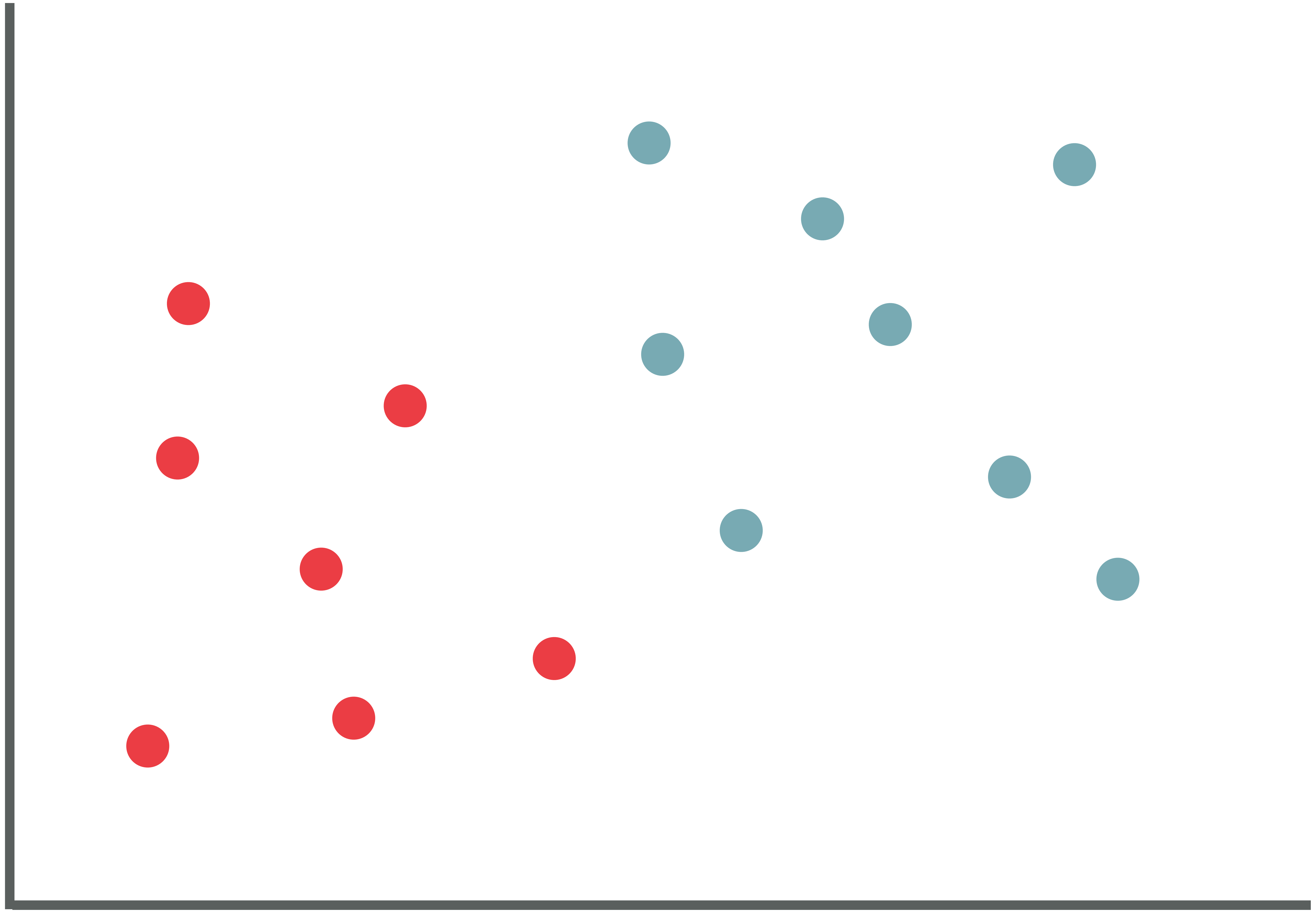
SVM



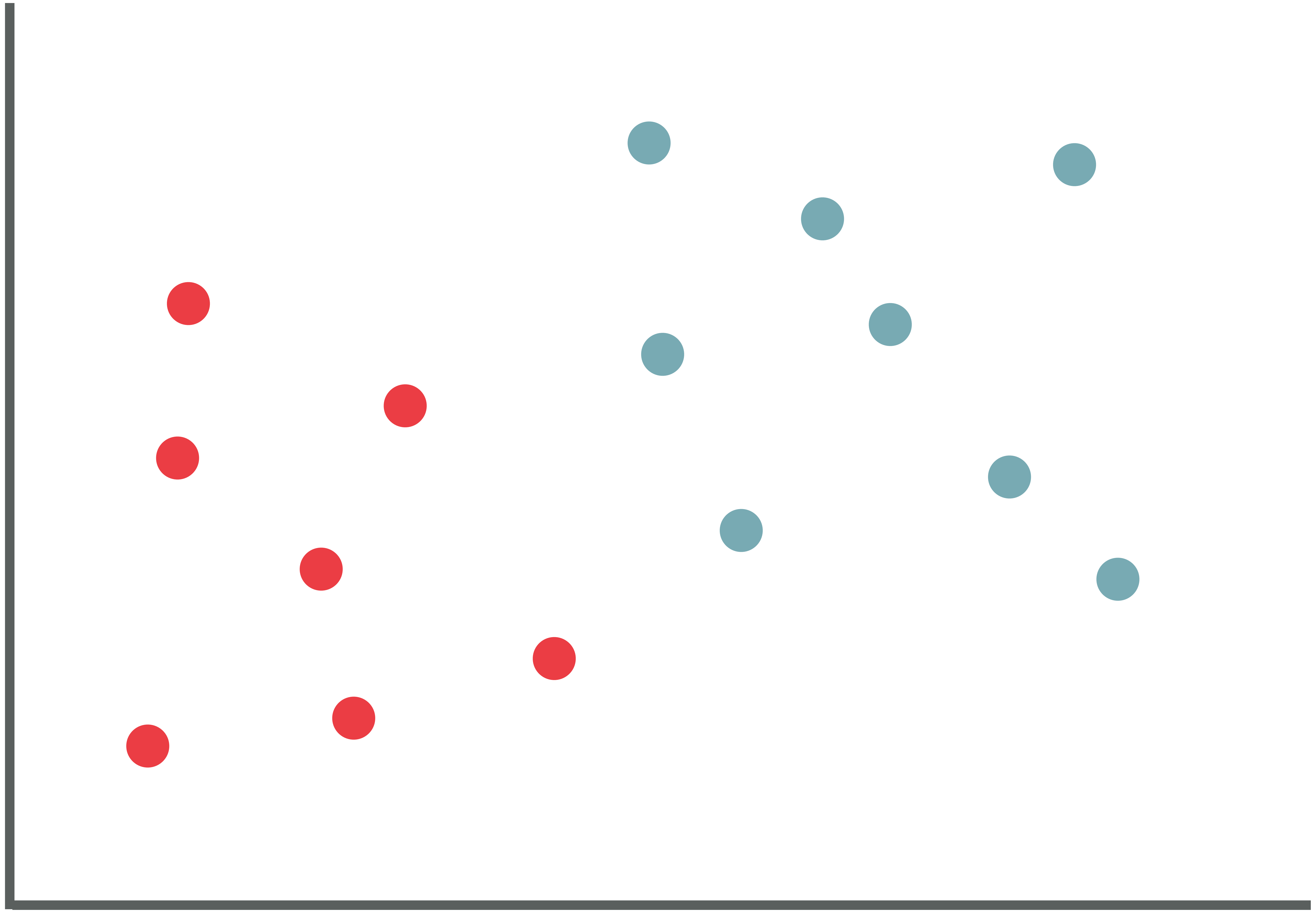
NOT LINEAR

Source: Statnikov, Hardin, Guyon, Aliferis *AMIA* 2009

SVM

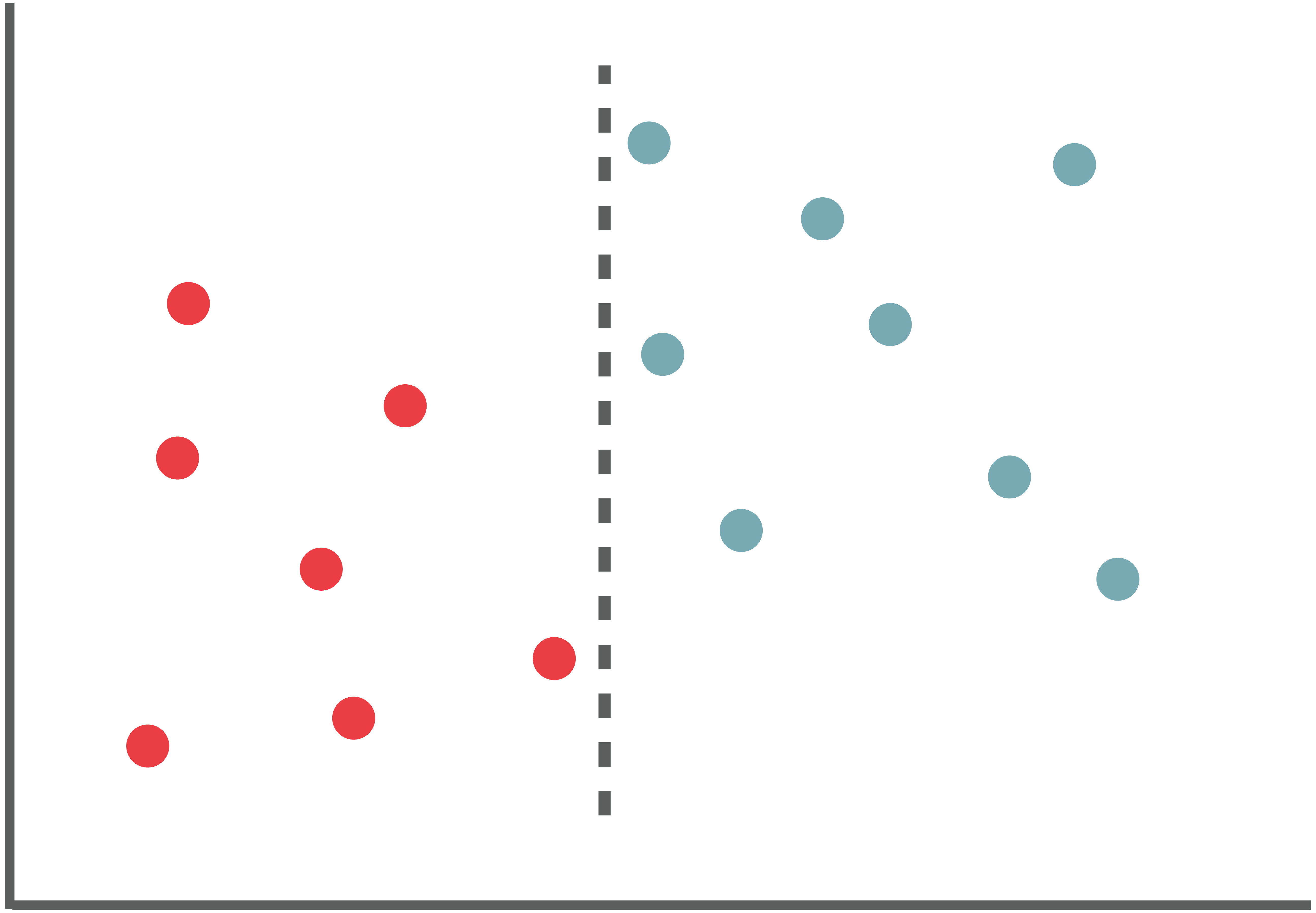


SVM



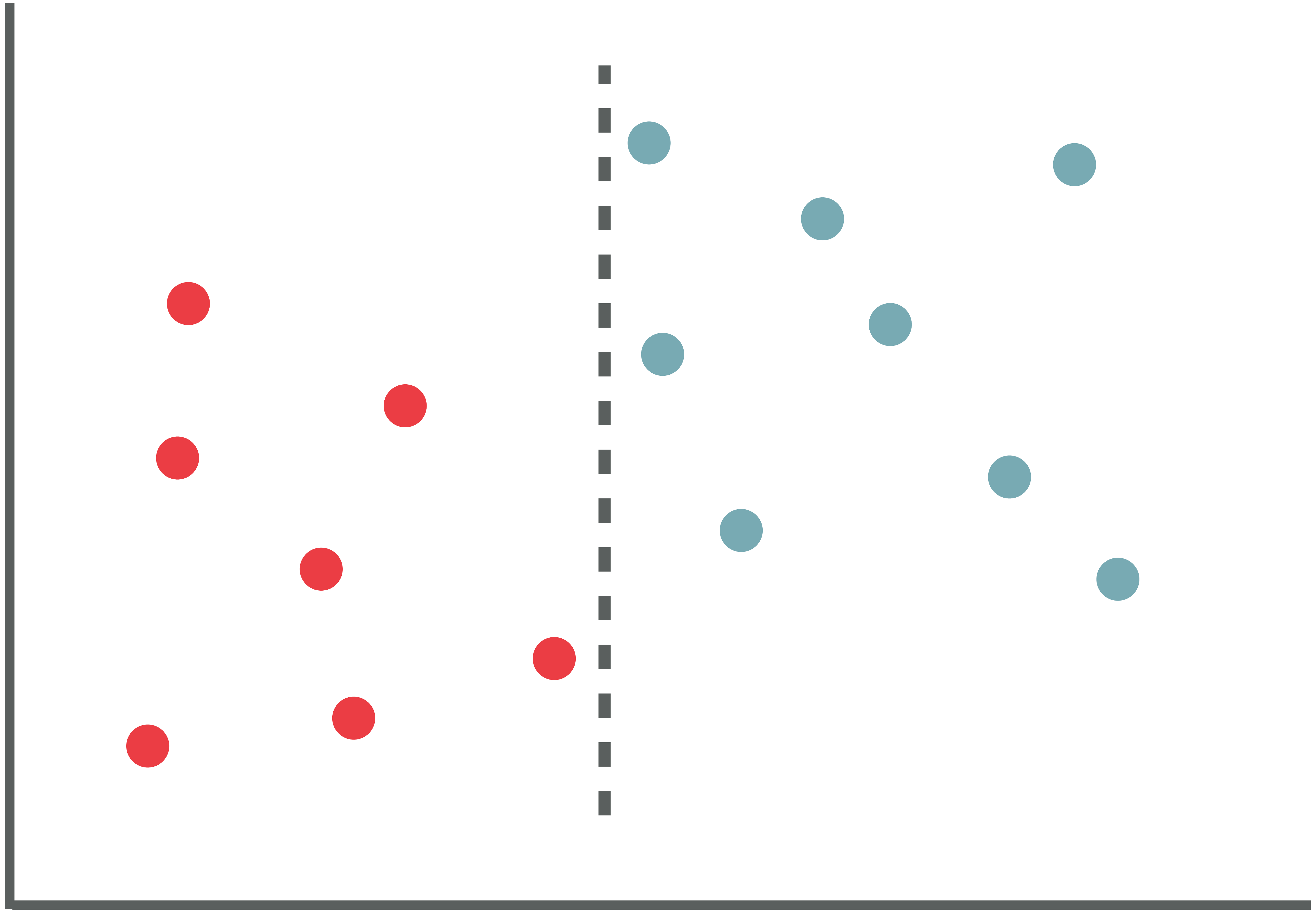
Classify!

SVM



Classify!

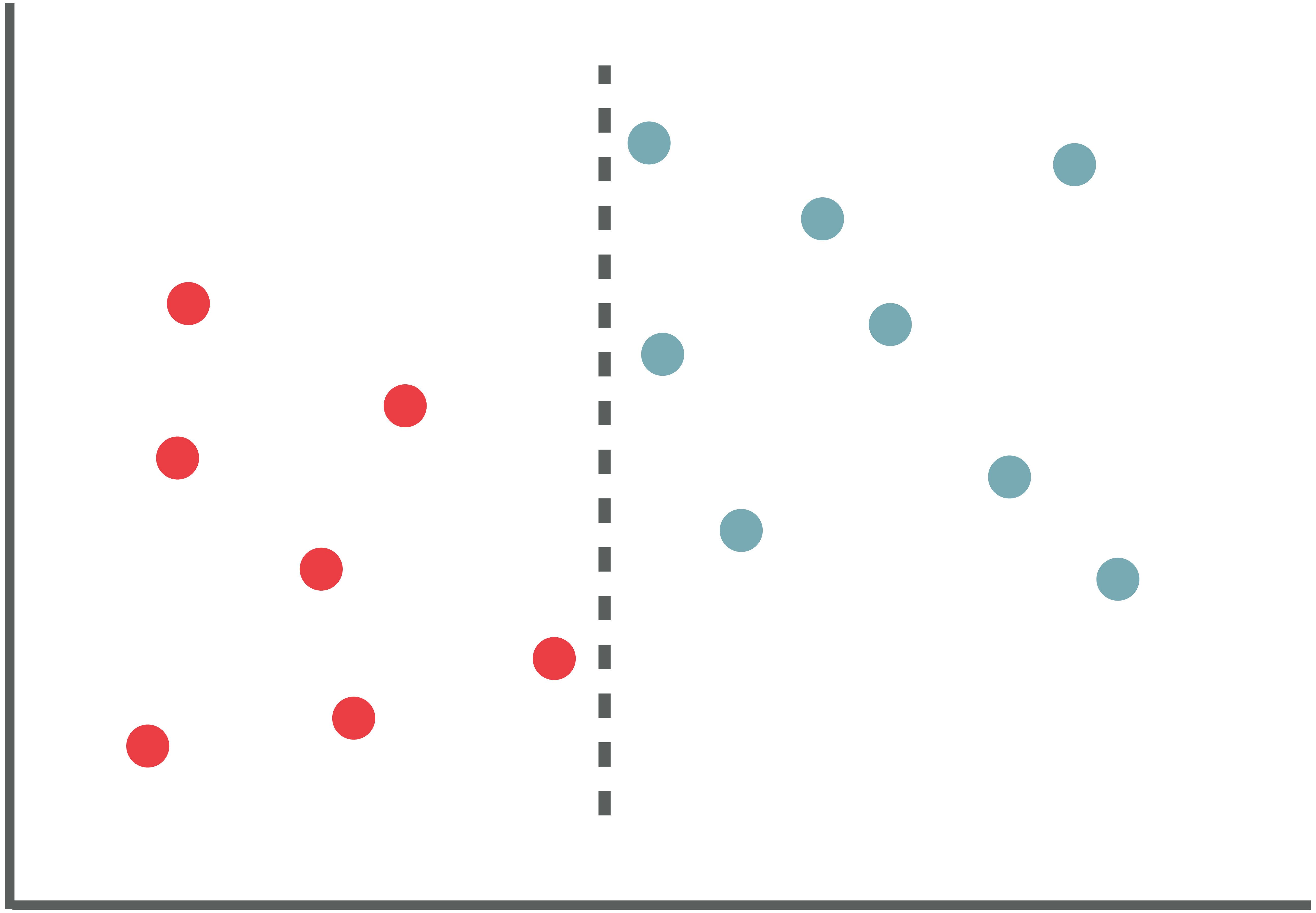
SVM



Classify!

DONE

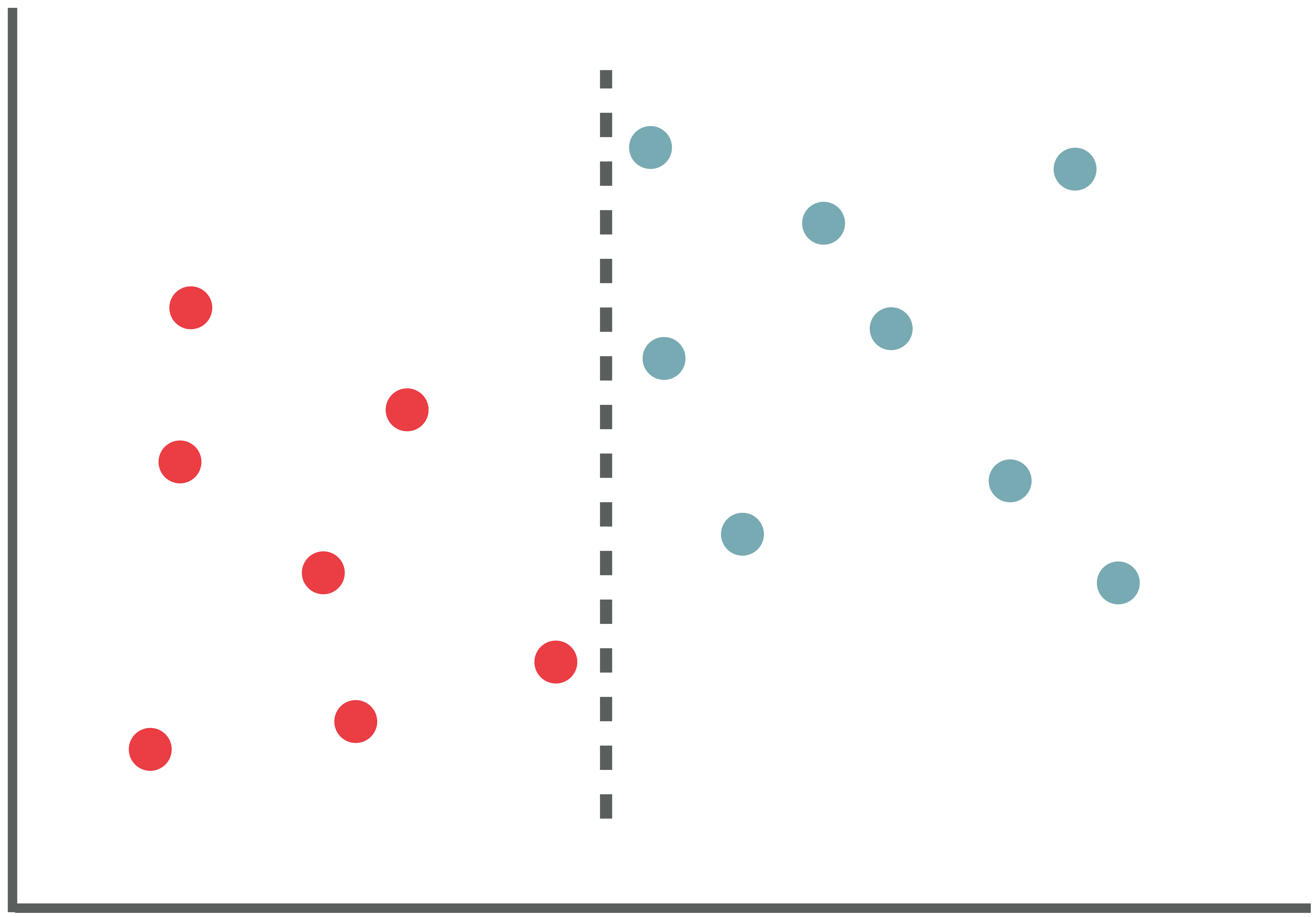
SVM



Classify!

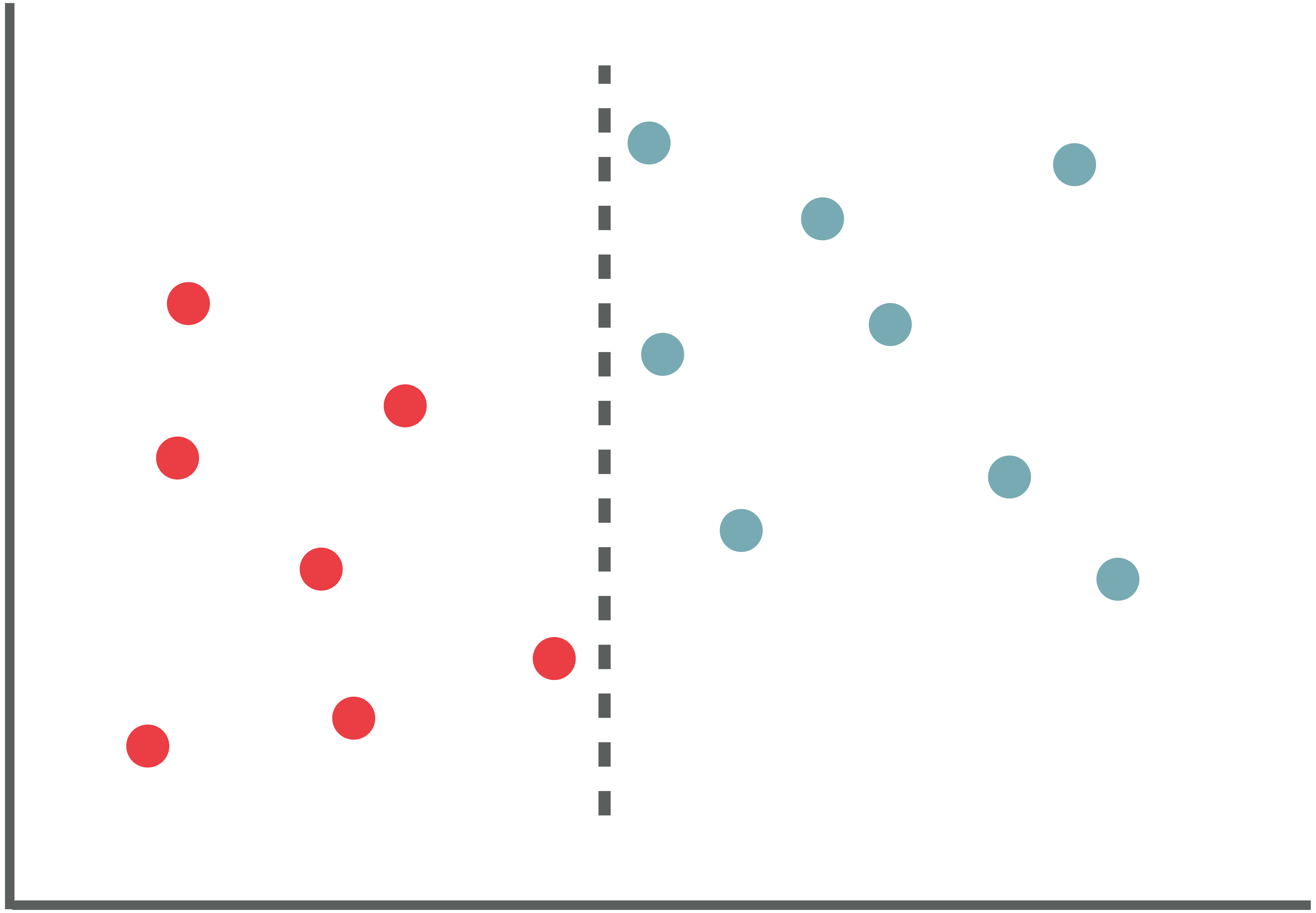
~~DONE~~

SVM



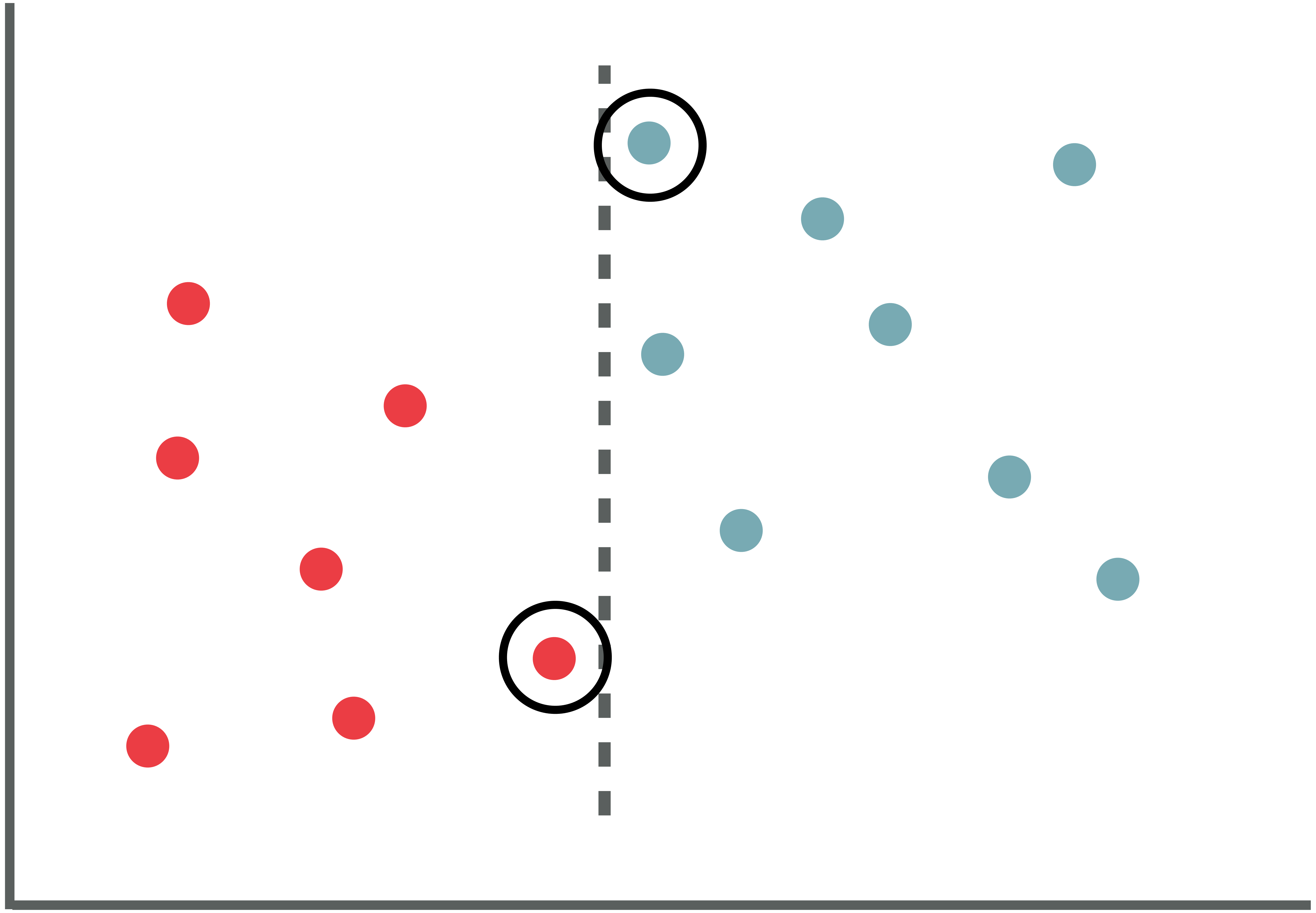
This is not the optimal separator

SVM



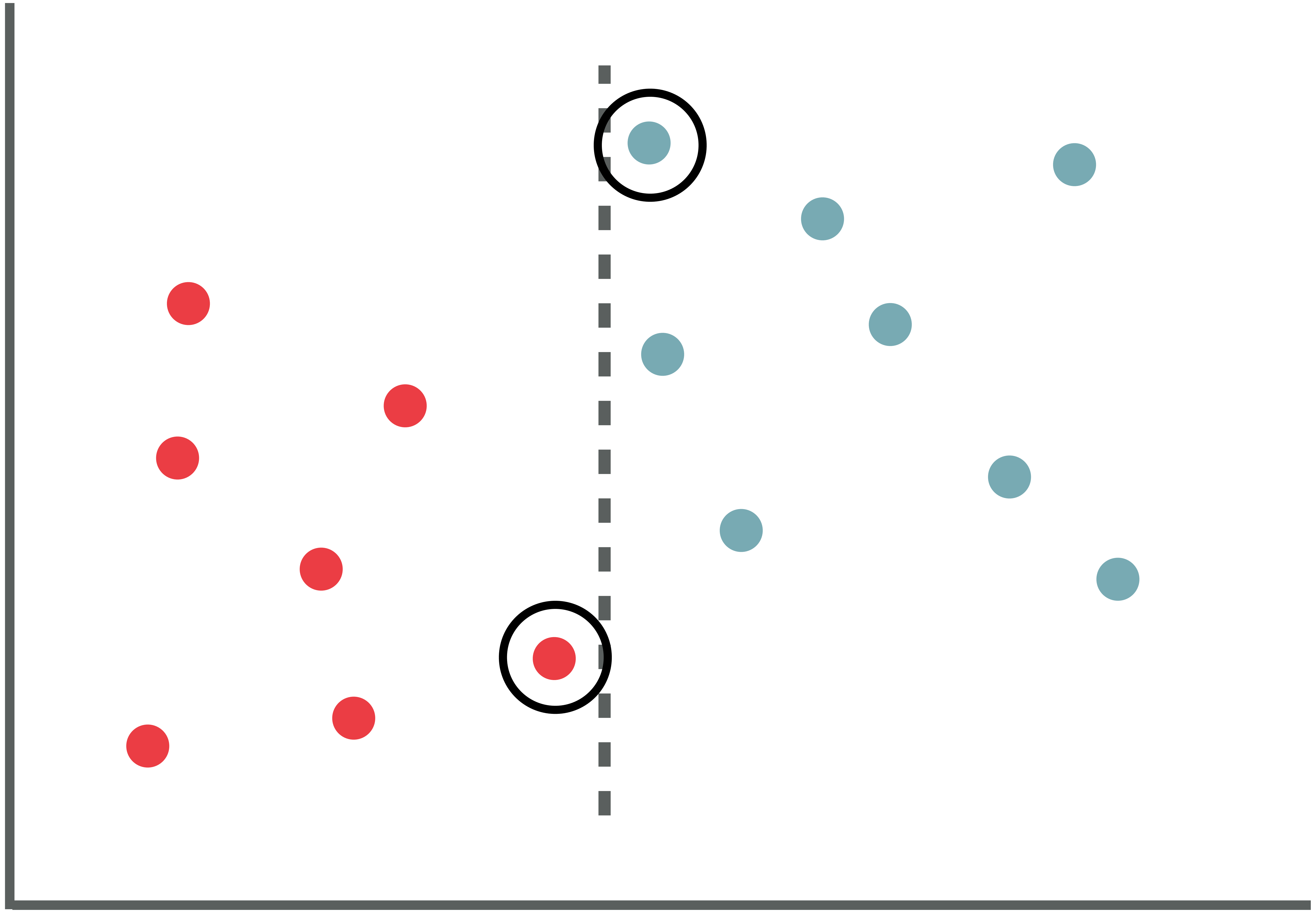
How do we find the "optimal"?

SVM



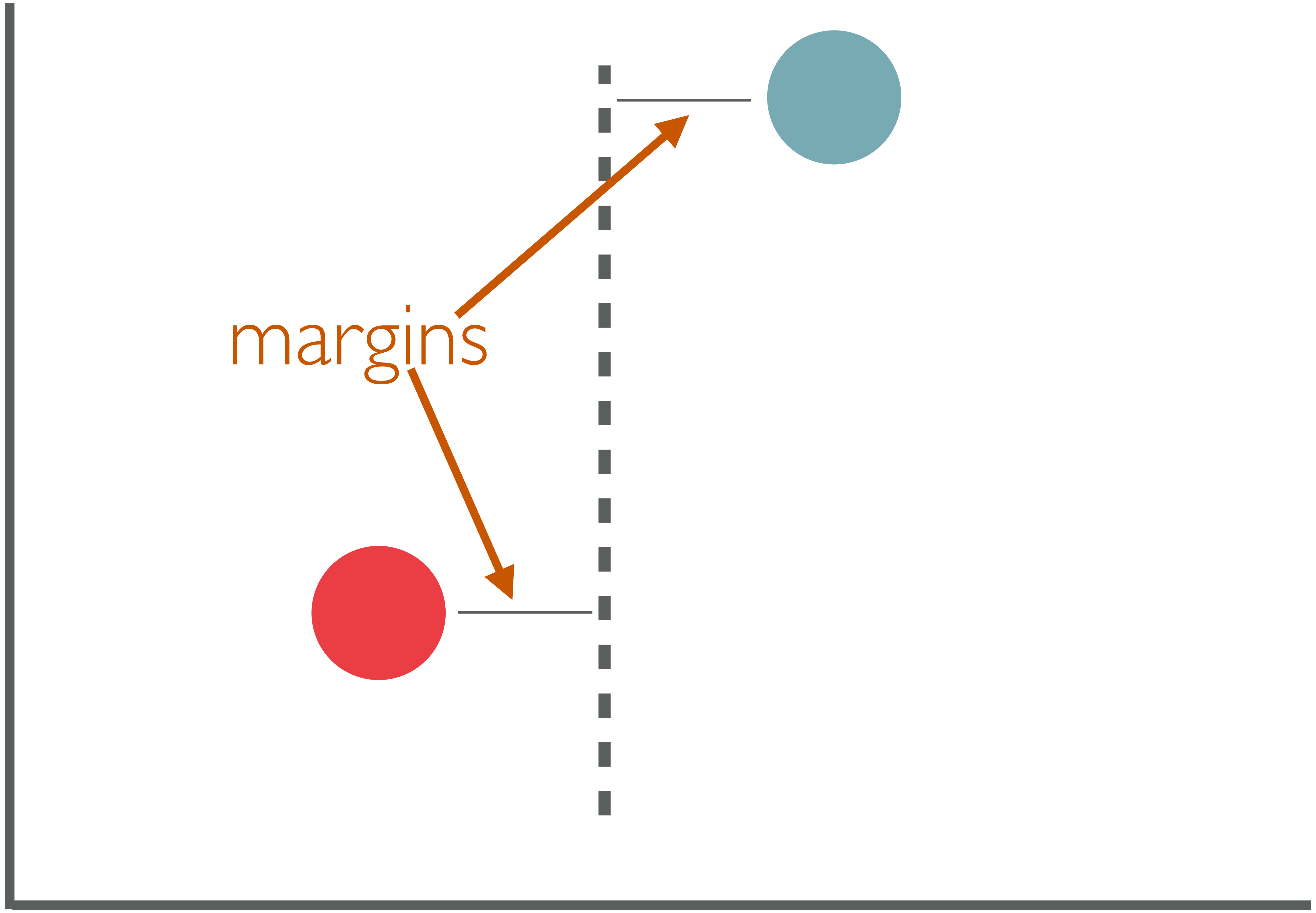
These are your "support vectors"

SVM



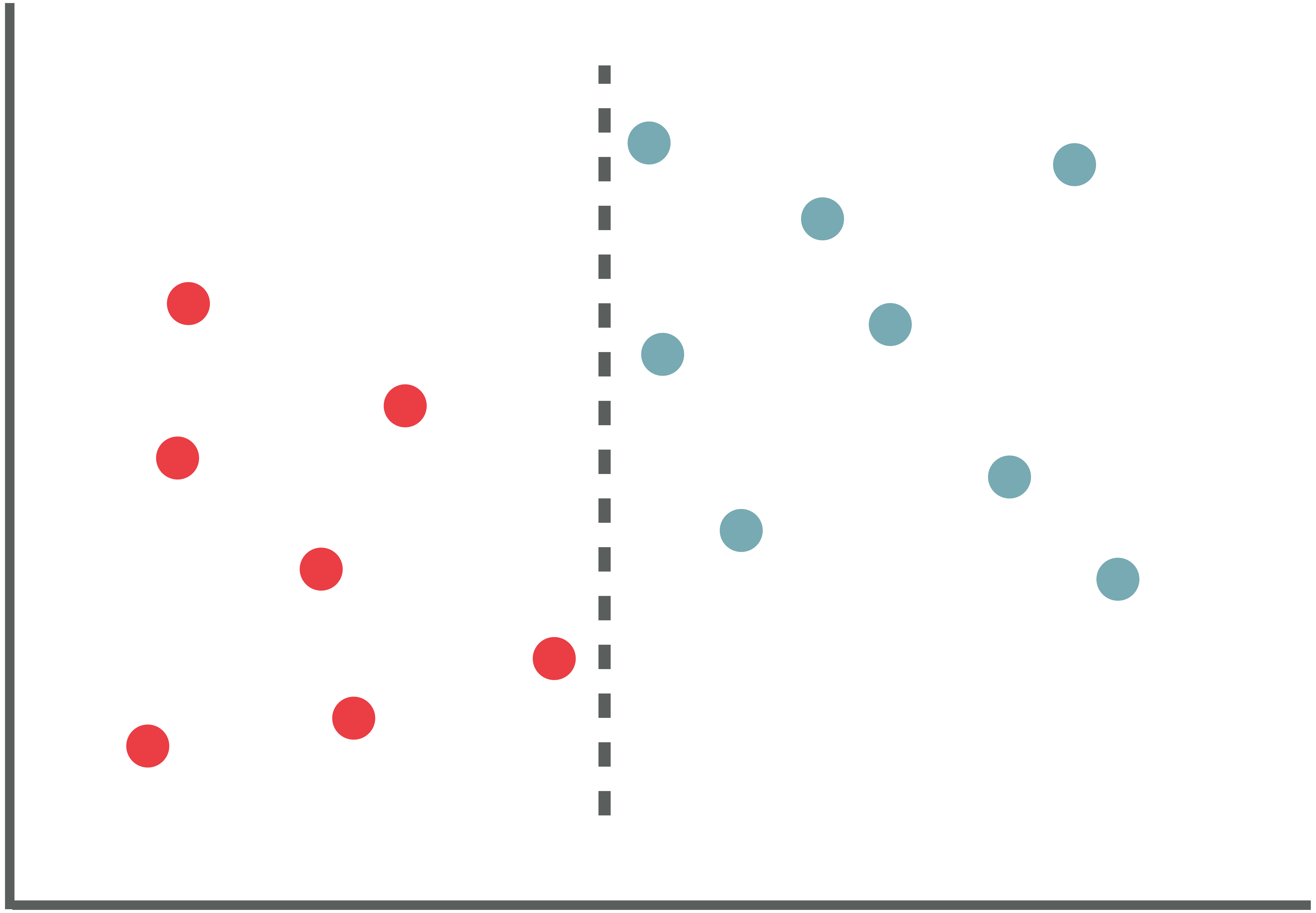
Support vectors are the points nearest to the plane

SVM



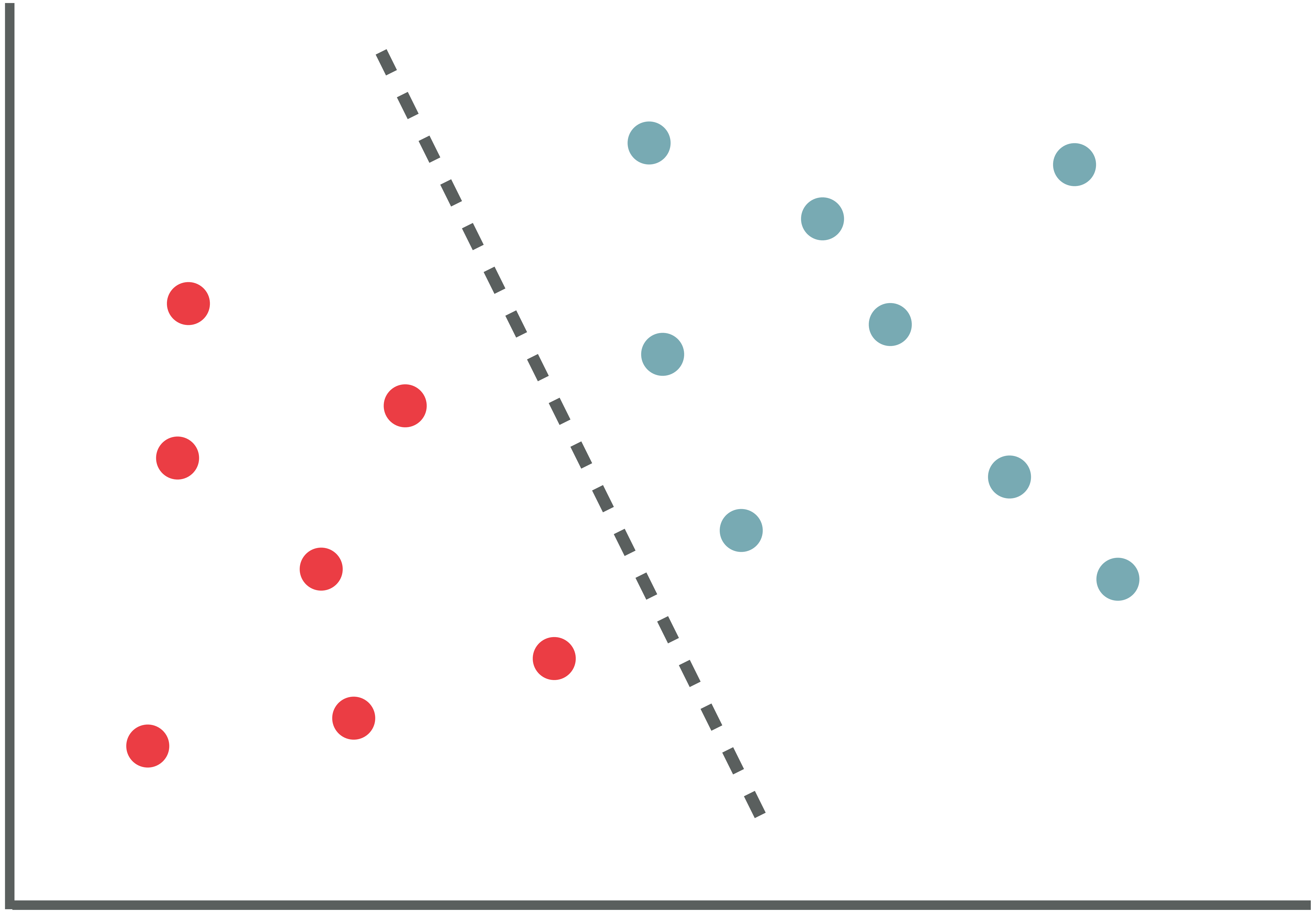
These are your margins

SVM



Goal: maximize the margins (*m*) such that the decision boundary is as far away from the data of both classes as possible

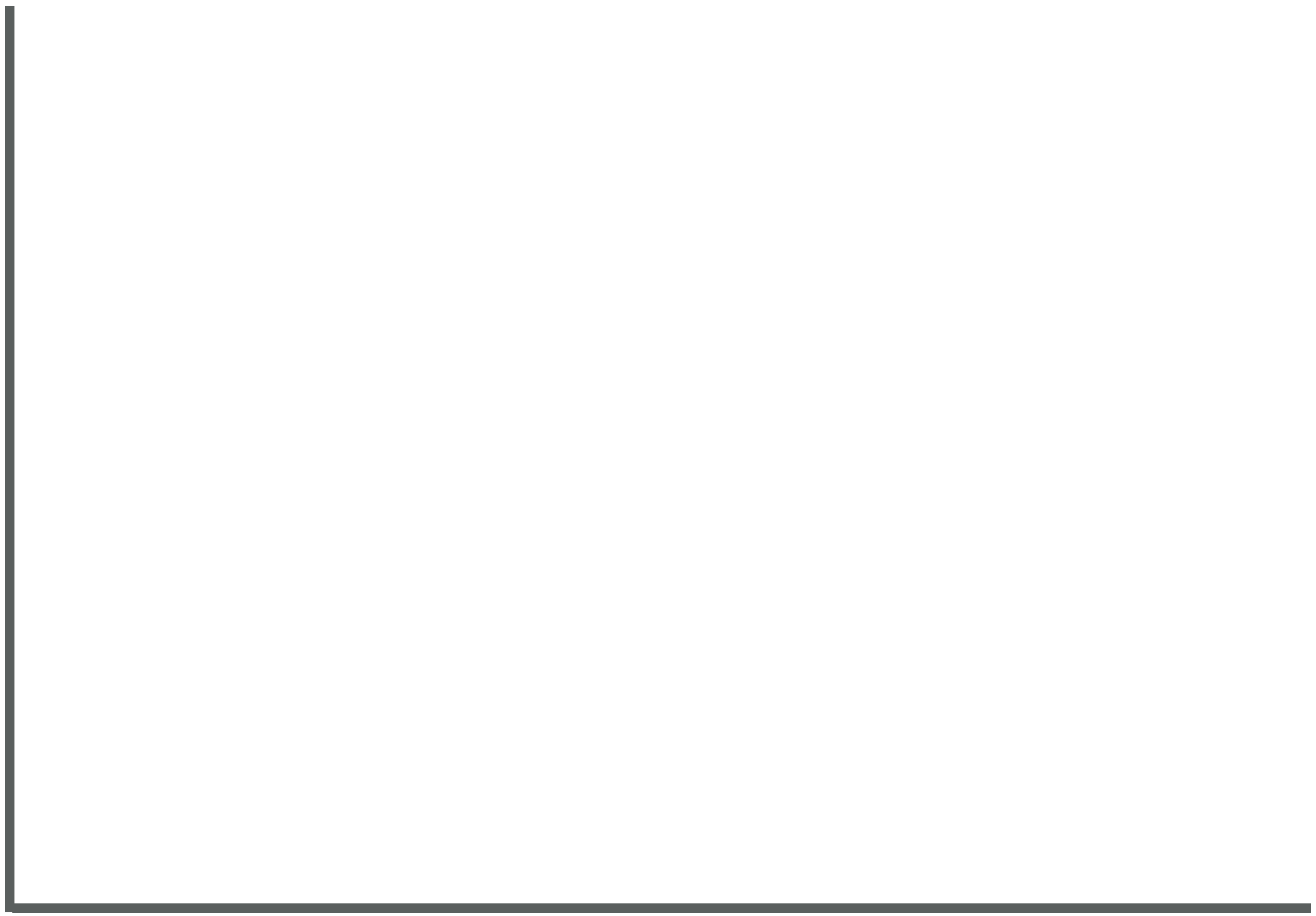
SVM



Classify!

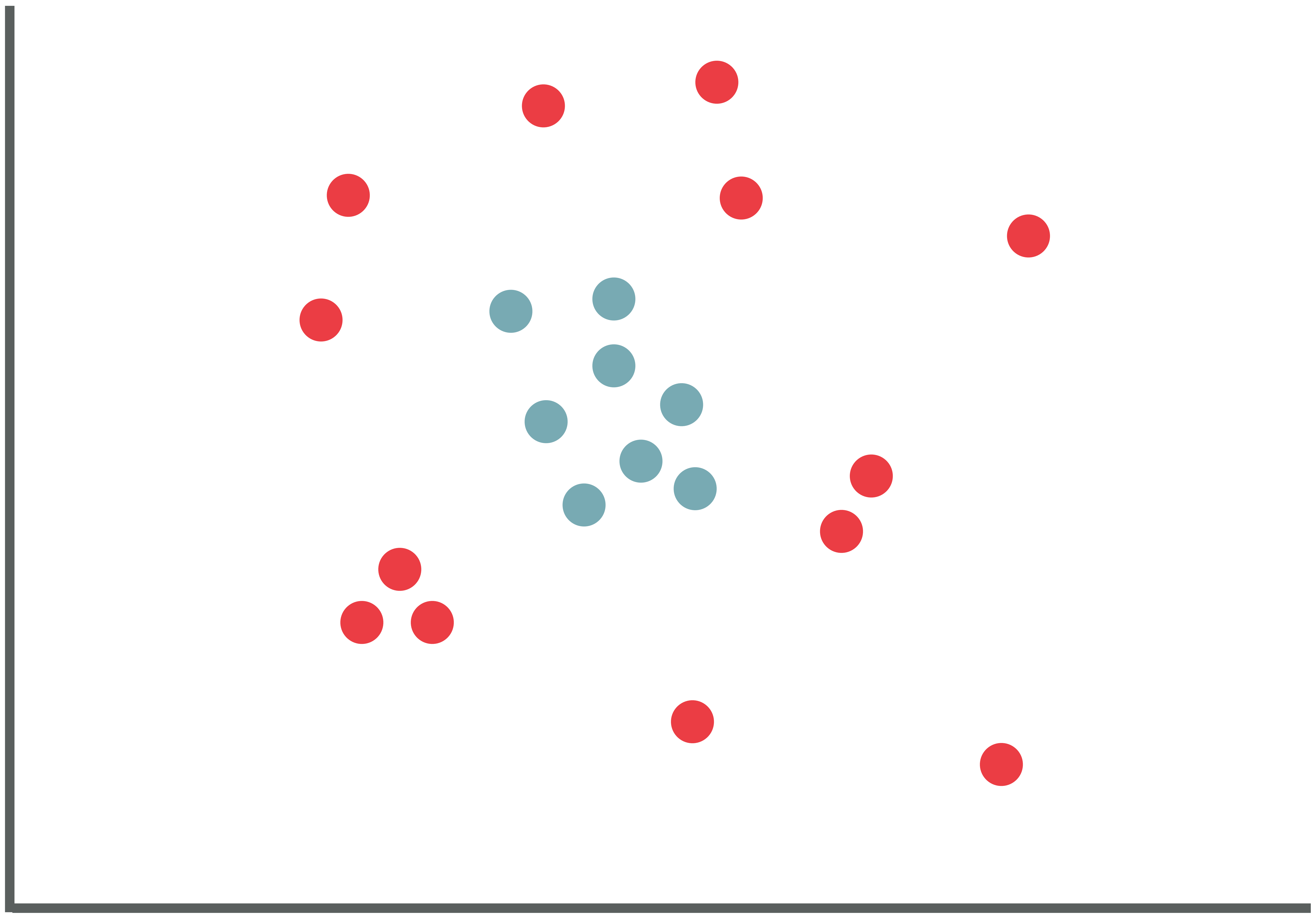
DONE

SVM



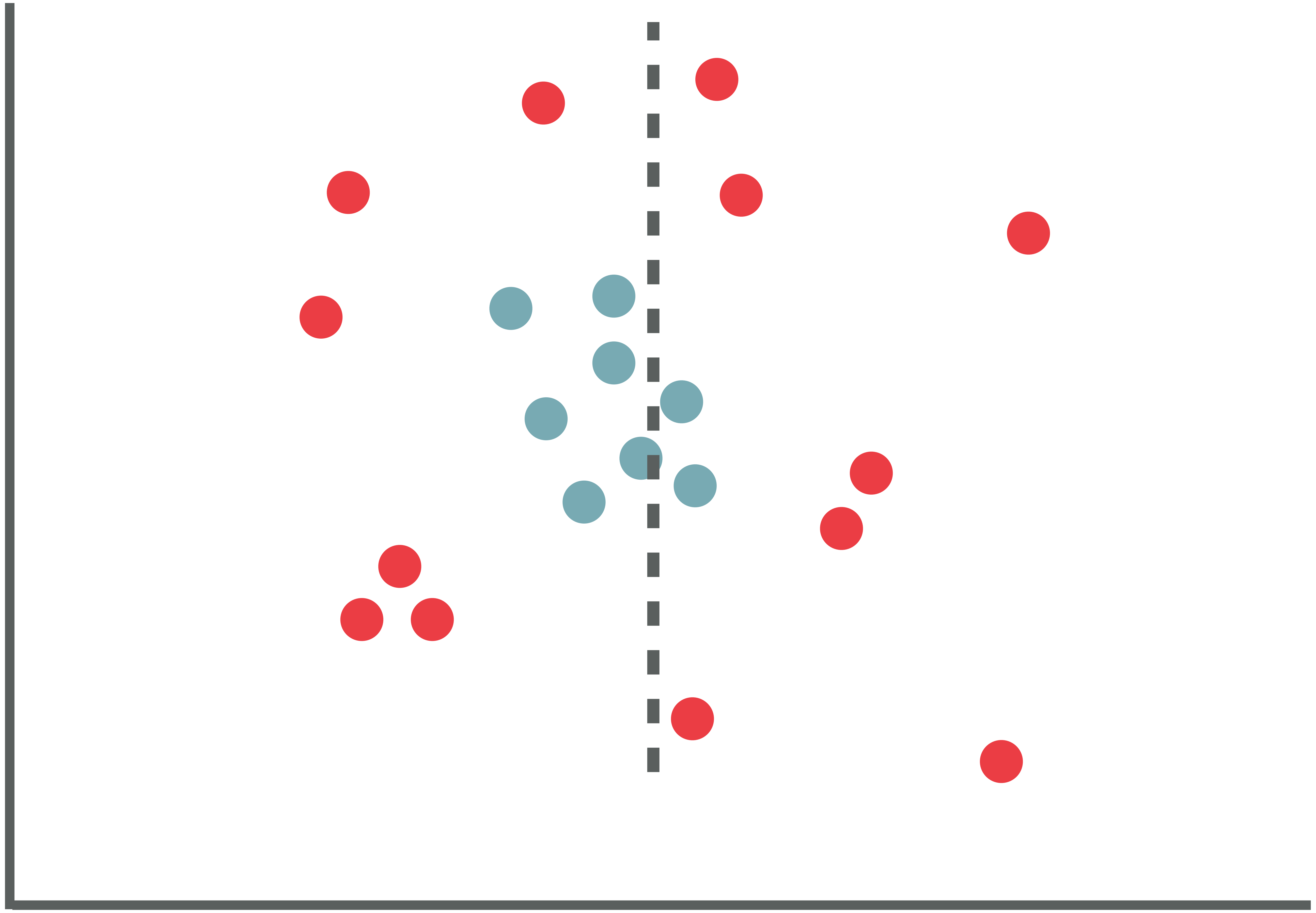
Okay… but…

SVM



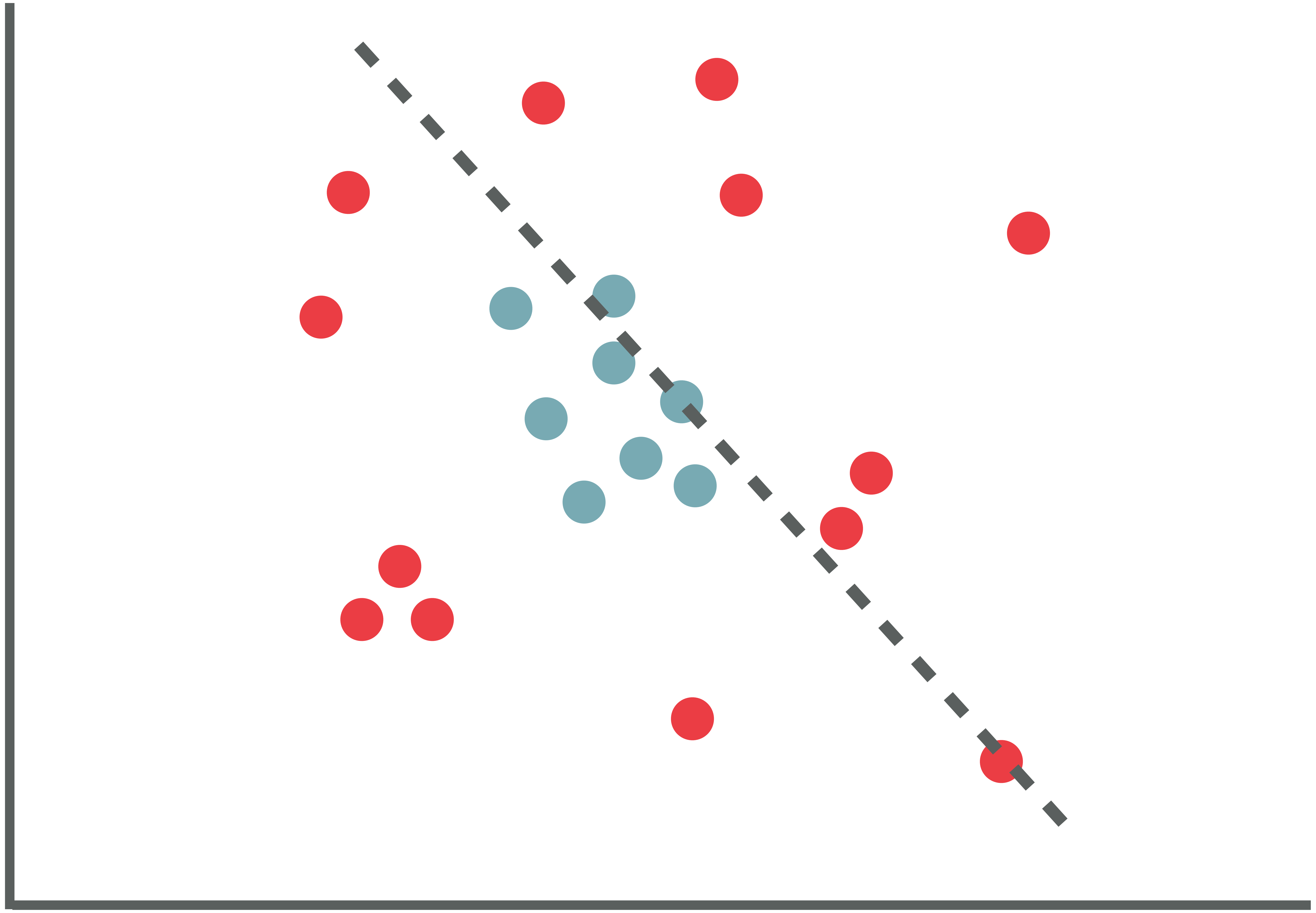
Classify!

SVM



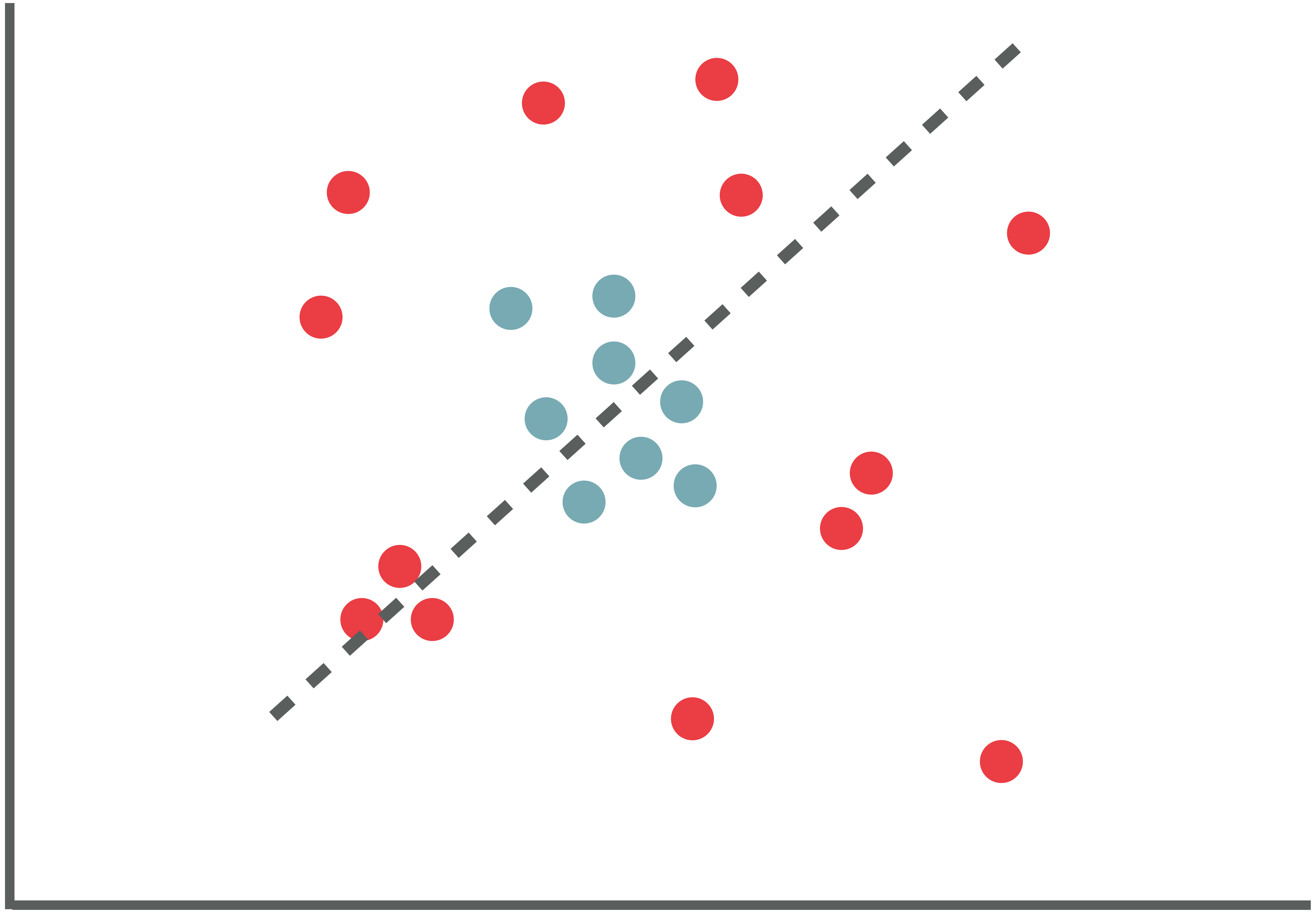
Classify!

SVM



Classify!

SVM

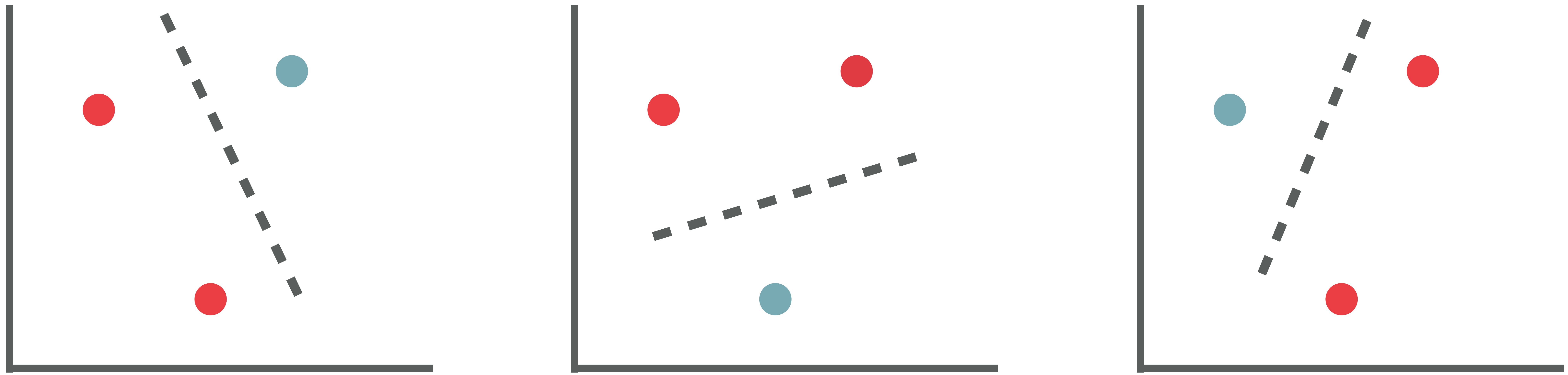


Classify!

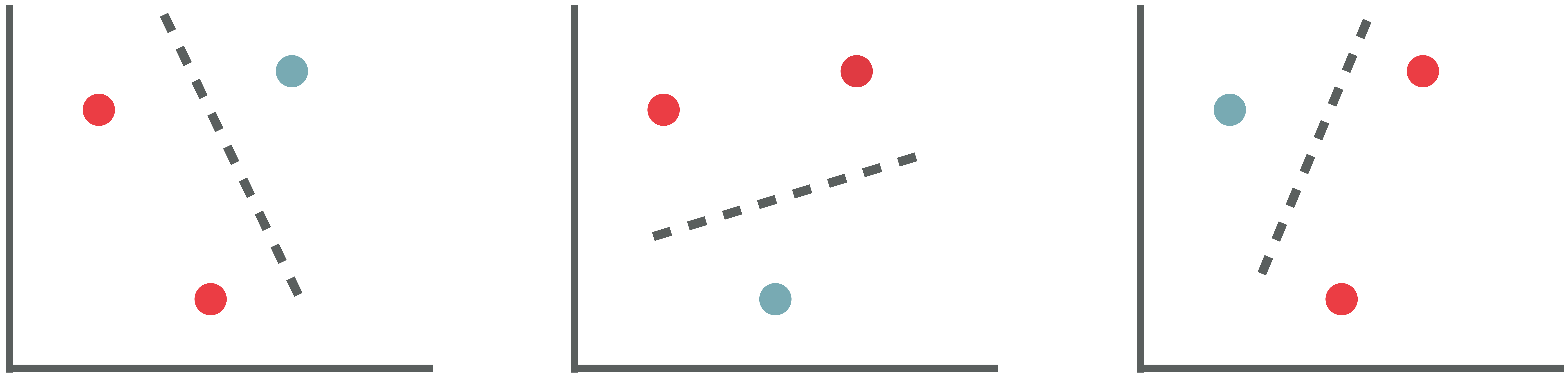
SVM

fy!

SVM



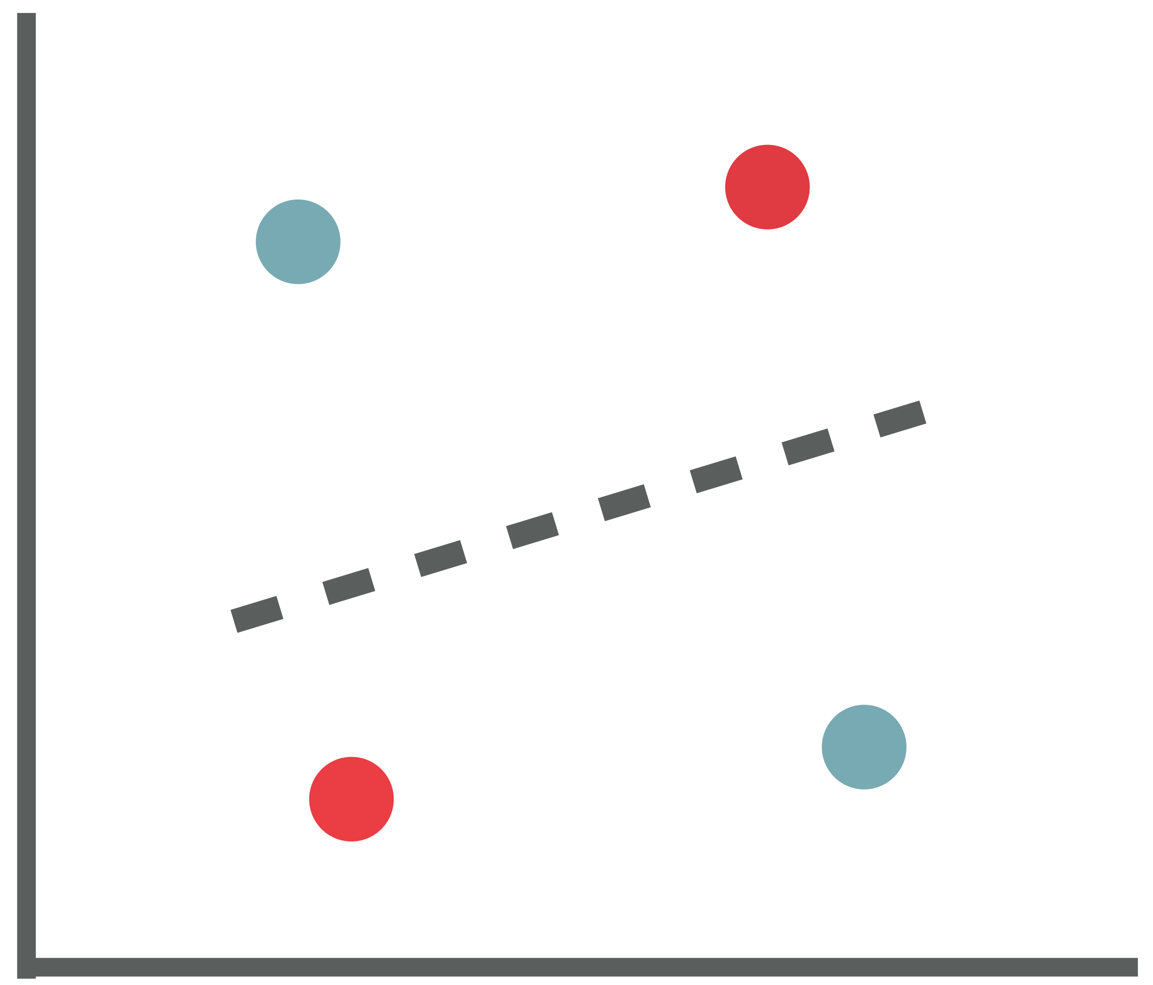
SVM



If we have 3 training points in 2D space, no matter how the points are labeled we can perfectly classify them using a linear algorithm

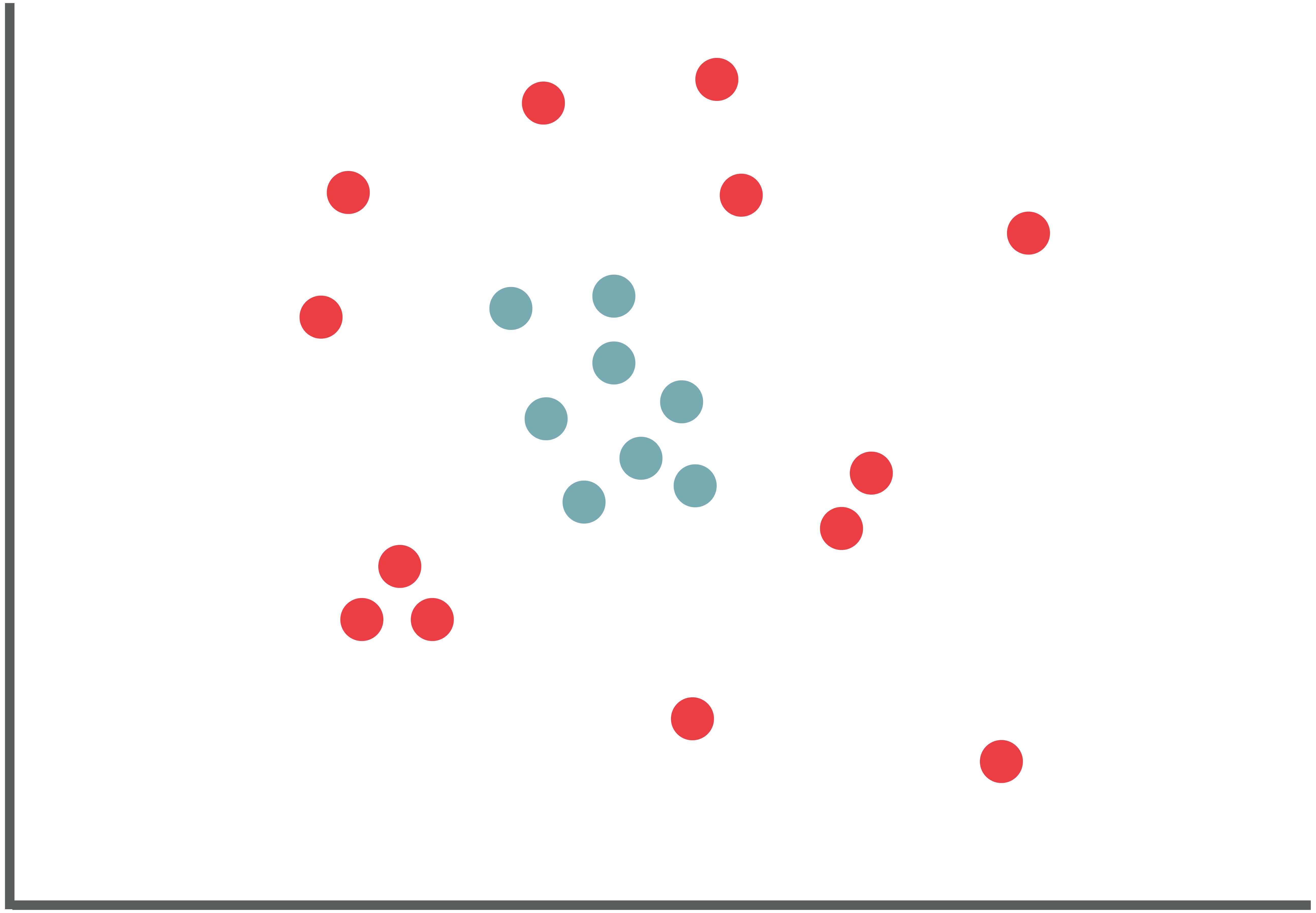
Source: Martin Law, MSU

SVM



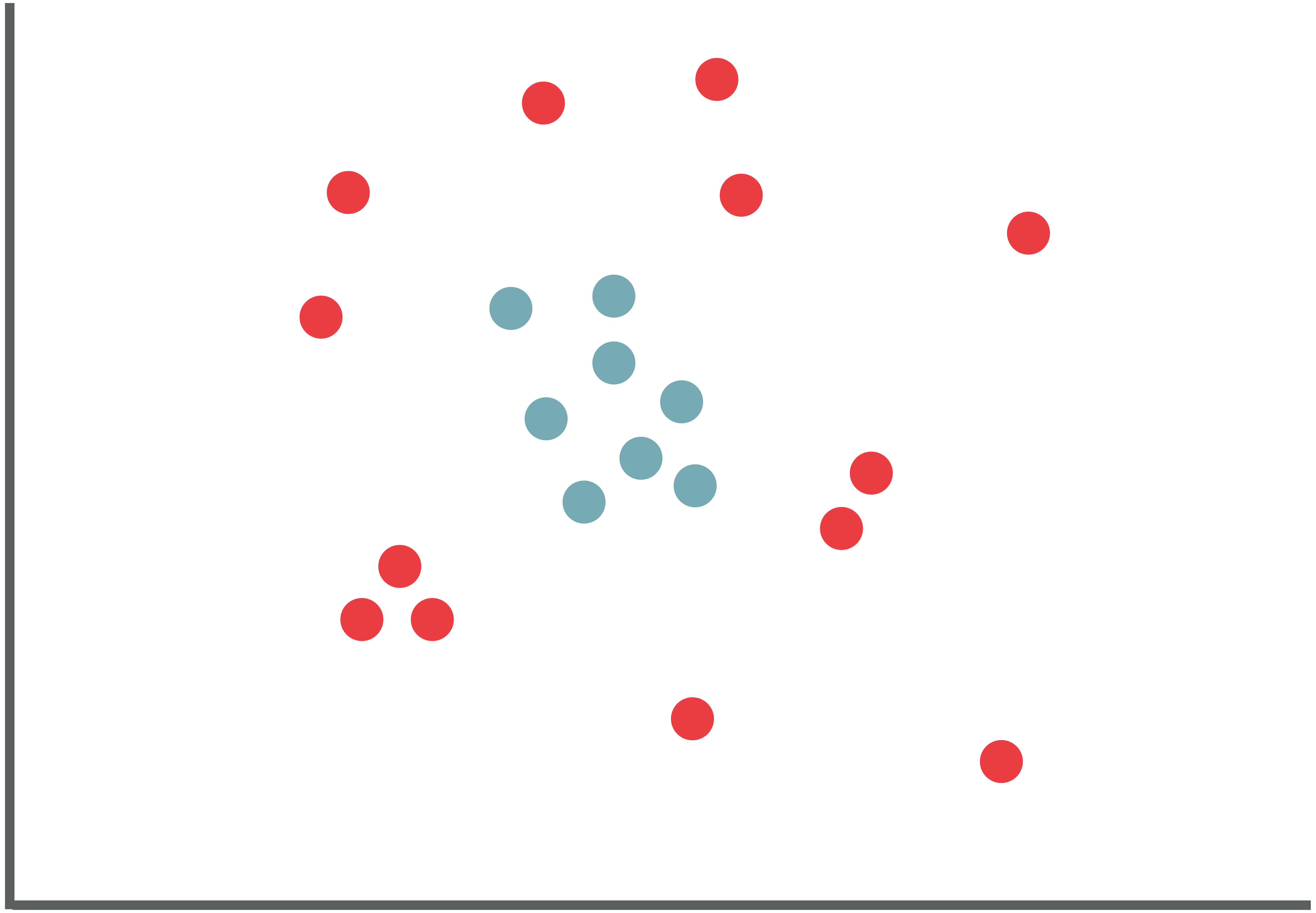
But add a fourth point, and this can fail

SVM



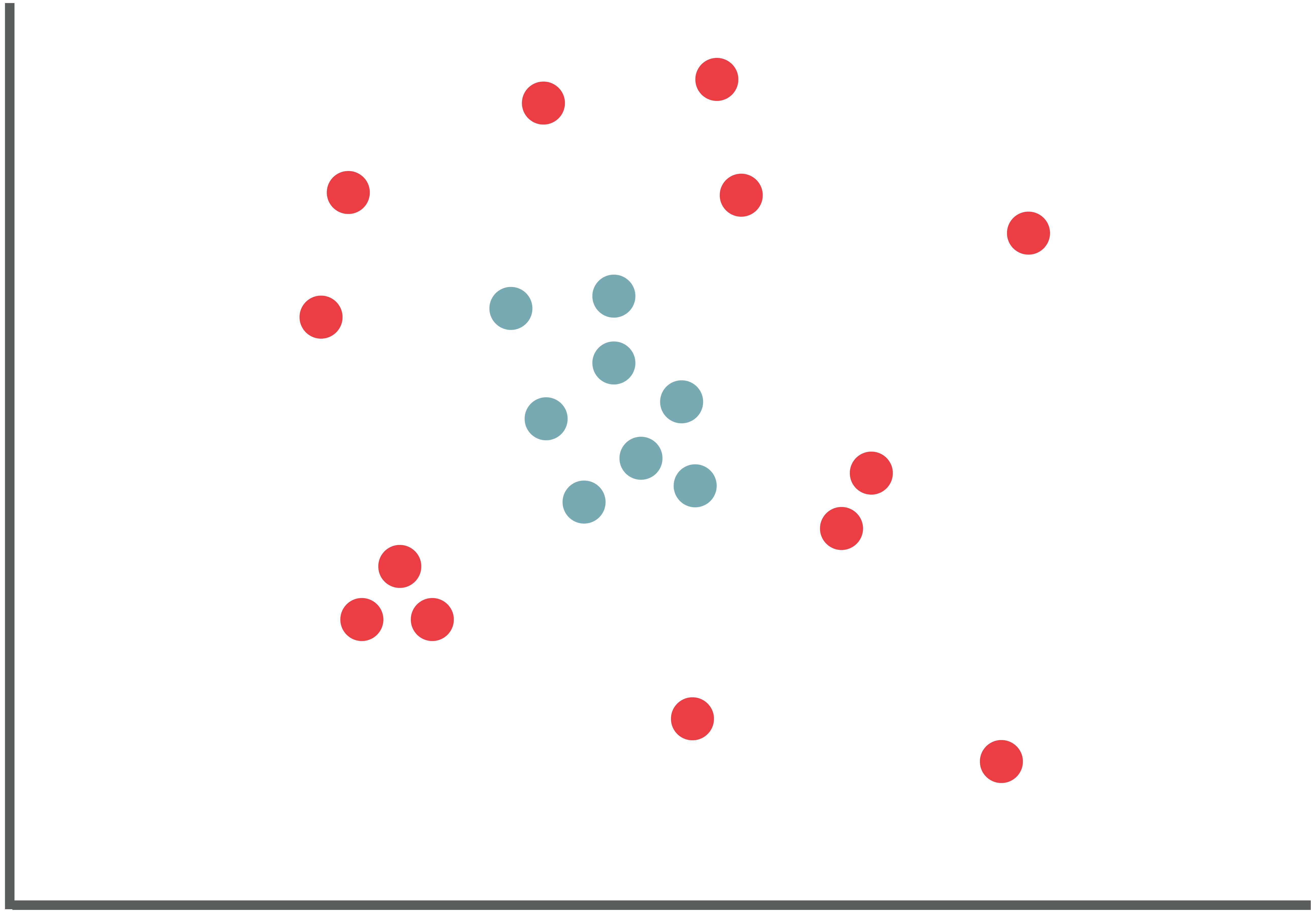
SVM: Total hack to turn data that are not optimally suited for linear classification into a linearly classifiable problem by warping them into higher dimensions

SVM



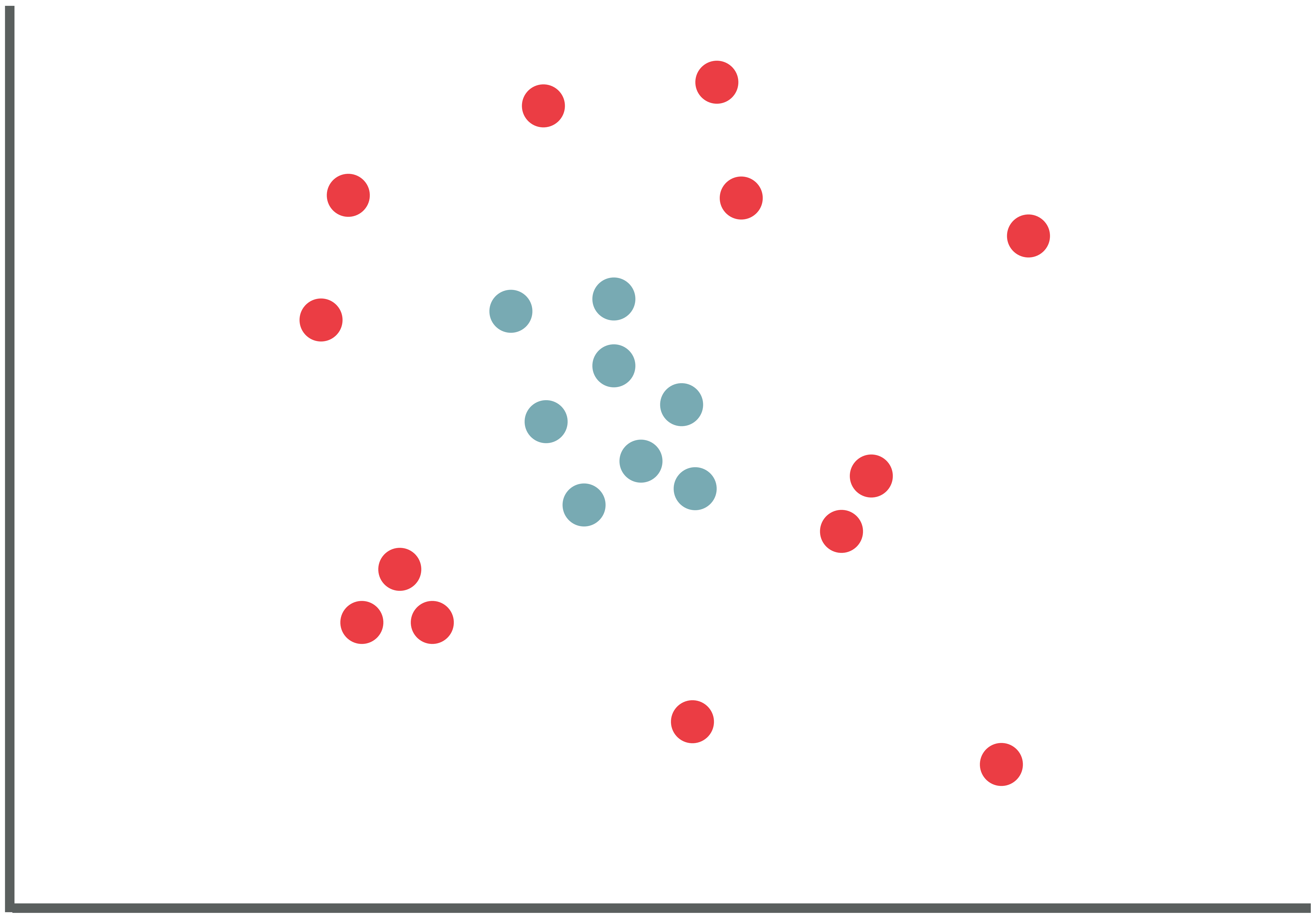
If we have N training points in $(N\text{-}1)$D space, no matter how the points are labeled we can perfectly classify them using a linear algorithm

SVM



But rarely do you need to go to *N*-1 space to do this

SVM



Classify!

Common SVM kernels

Examples:

$K(\vec{x}_i,\vec{x}_j)=\vec{x}_i\cdot\vec{x}_j$	Linear kernel
$K(\vec{x}_i,\vec{x}_j)=\exp(-\gamma\left\|\vec{x}_i-\vec{x}_j\right\|^2)$	Gaussian kernel
$K(\vec{x}_i,\vec{x}_j)=\exp(-\gamma\left\|\vec{x}_i-\vec{x}_j\right\|)$	Exponential kernel
$K(\vec{x}_i,\vec{x}_j)=(p+\vec{x}_i\cdot\vec{x}_j)^q$	Polynomial kernel
$K(\vec{x}_i,\vec{x}_j)=(p+\vec{x}_i\cdot\vec{x}_j)^q\exp(-\gamma\left\|\vec{x}_i-\vec{x}_j\right\|^2)$	Hybrid kernel
$K(\vec{x}_i,\vec{x}_j)=\tanh(k\vec{x}_i\cdot\vec{x}_j-\delta)$	Sigmoidal

Source: Martin Law, MSU

COGS 108
Data Science in Practice

Geospatial Analyses

Geocoding

Geocoding is the process of converting addresses (like a street address) into geographic coordinates (like latitude and longitude), which you can use to place markers on a map, or position the map.

Source: Google

Google Maps API

Client-side geocoding, which is executed in the browser, generally in response to user action. The Google Maps JavaScript API provides classes that make the requests for you.

Google Maps API

When to use client-side geocoding

The short answer is "almost always.'' The reasons are:

- Client-side request and response provide a faster, more interactive experience for users.
- A client-side request can include information that improves geocoding quality: user language, region, and viewport.

Google Maps API

HTTP server-side geocoding, which allows your server to directly query Google's servers for geocodes. The Google Maps Geocoding API is the web service that provides this functionality. Typically, you integrate this service with other code that is running server-side.

Google Maps API

When to use server-side geocoding

Server-side geocoding is best used for applications that require you to geocode addresses without input from a client. A common example is when you get a dataset that comes independently of user input, for instance if you have a fixed, finite, and known set of addresses that need geocoding. Server-side geocoding can also be useful as a backup for when client-side geocoding fails.

Google Maps API

The Google Maps Geocoding API has the following limits in place:

Standard Usage Limits

Users of the standard API:

- 2,500 free requests per day, calculated as the sum of client-side and server-side queries.
- 50 requests per second, calculated as the sum of client-side and server-side queries.

UC San Diego

Source: Google

Geocoding

Reverse geocoding is the process of converting geographic coordinates into a human-readable address. The Google Maps Geocoding API's reverse geocoding service also lets you find the address for a given place ID.

Source: Google

Geocoding

Jupyter Demo!

Administrative stuff

- **http://cape.ucsd.edu/**

- **COGS: https://goo.gl/dmujYA**
- The cognitive science department runs it's own evaluations. You can submit evaluations for the instructor, your section TA, and/or any TA you had enough interaction with to provide feedback. Please do fill these out - they are very useful for us to get feedback on our teaching, and provide feedback to the department, and this is especially useful for working on this new course.

Bradley Voytek, Ph.D.

UC San Diego
Cognitive and Neural Dynamics Laboratory

Department of Cognitive Science
Neurosciences Graduate Program
Halıcıoğlu Data Science Institute

bvoytek@ucsd.edu
@bradleyvoytek