Neural Data Science with **Python**

L10 : Classification and Clustering

Michael Graupner

SPPIN – Saint-Pères Institute for the Neurosciences

Université Paris Cité, CNRS

Free tutorial suggestions (next week, Nov 29th)

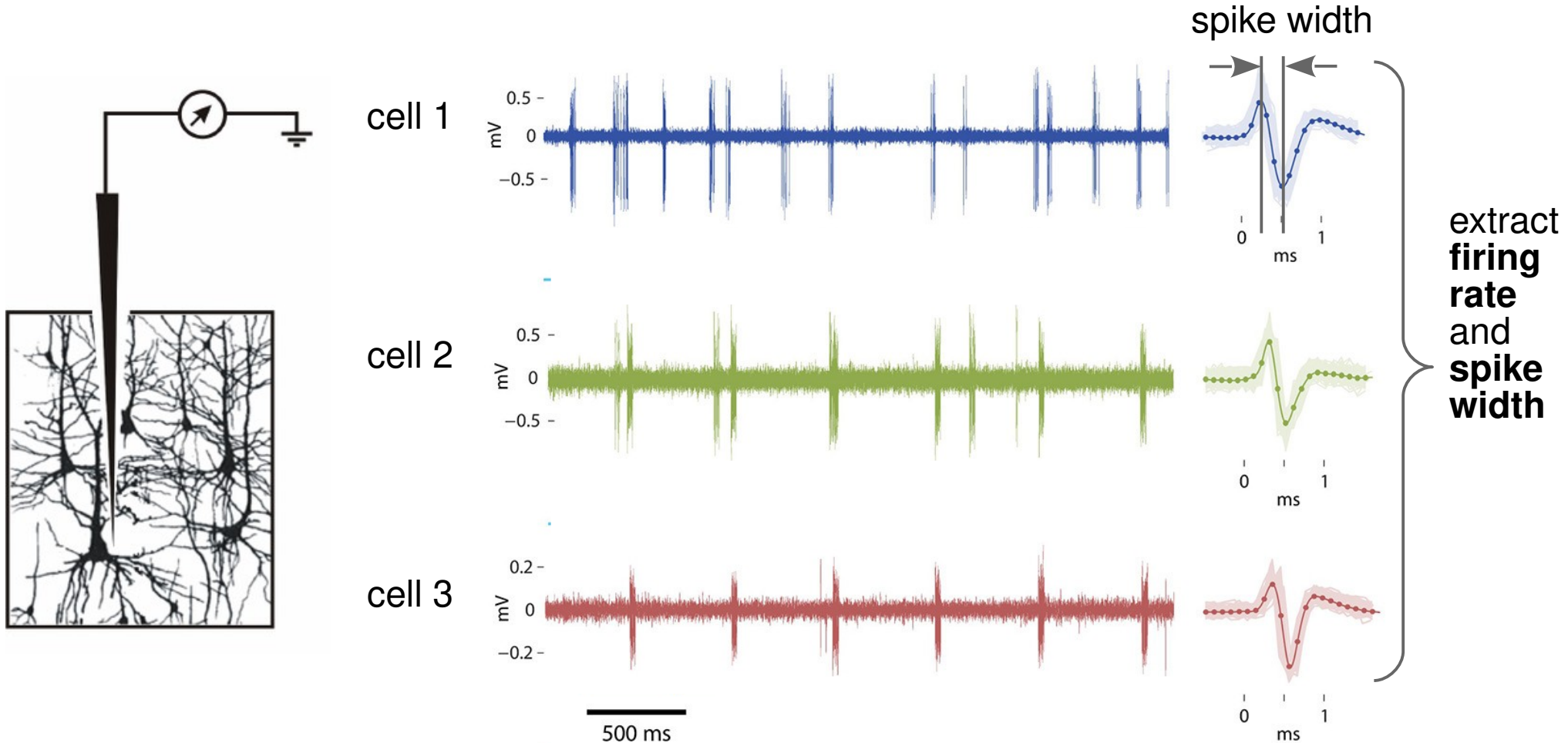1) Program a game in python : "Rock, Paper, Scissors"

2) Resolve typical programming errors (bugs) in form of a quiz

3) Re-work on programming concept/analysis concept suggested by YOU.
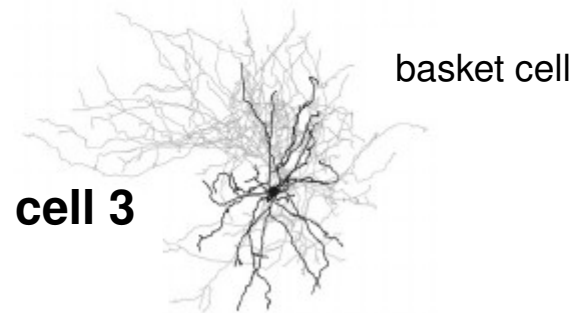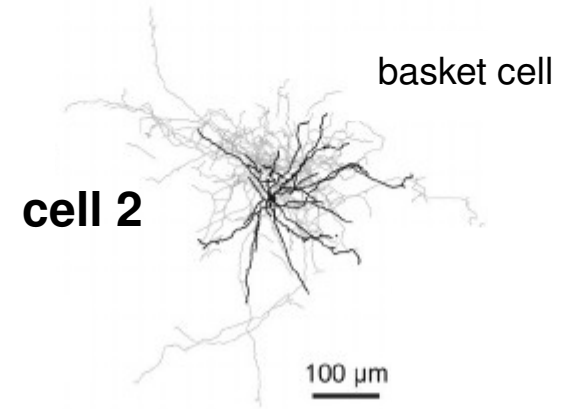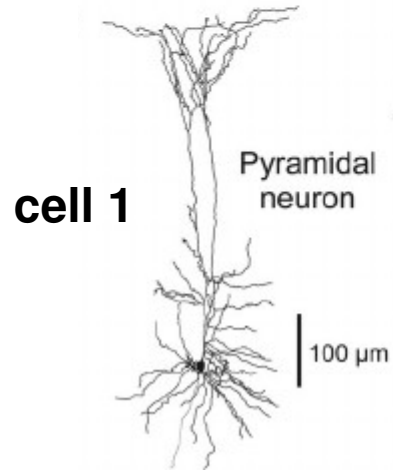
4) Start to work on a "fictive" end-of-course project : discuss approach, tips, tricks, typical problems

Slido Link : https://app.sli.do/event/pZ7KRRPsGkfTD8KioD56Y8/embed/polls/73a60122-8717-4539-8ba8-d25cd314451f
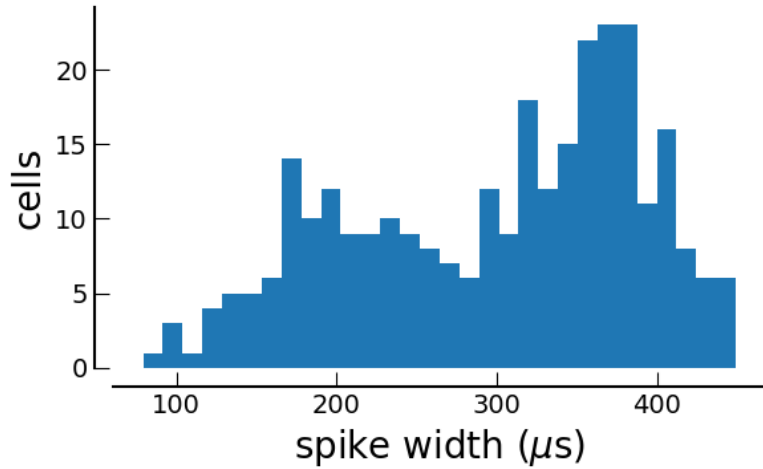
Example : extracellular single unit recording

Example : single unit recording + reconstruction

cell 1

Pyramidal neuron

100 µm

cell 2

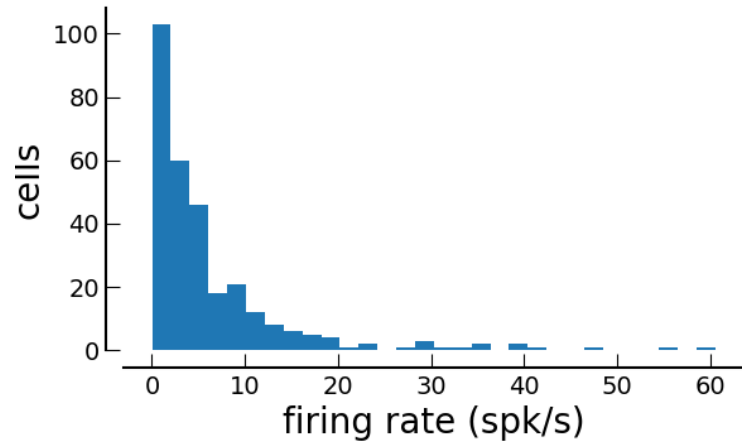basket cell

100 µm

cell 3

basket cell

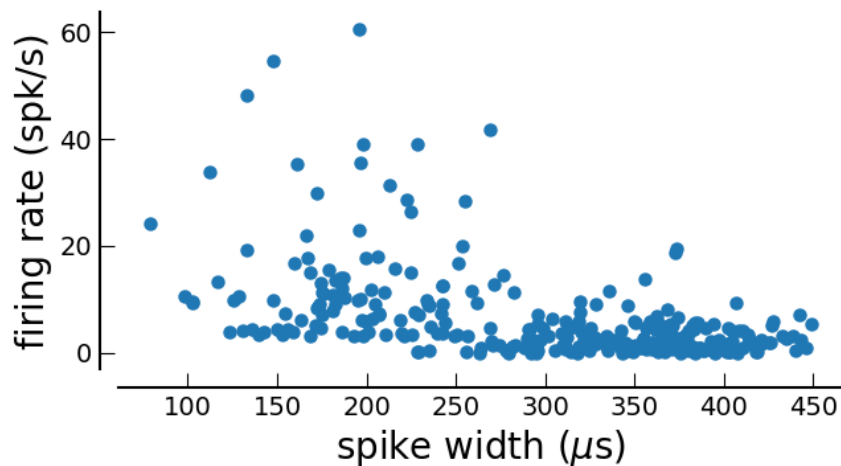# Example: ephys summary of recordings from all cells

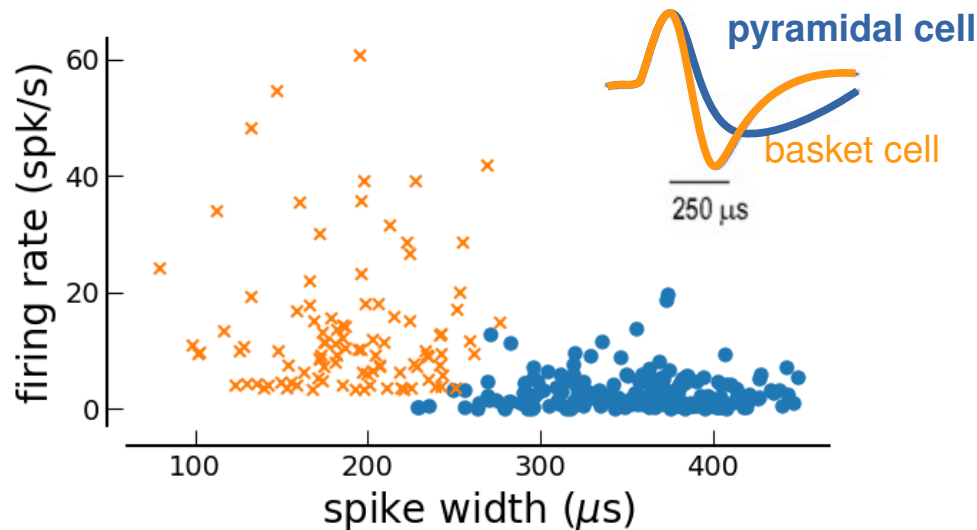

histogram of spike widths

histogram of firing rates

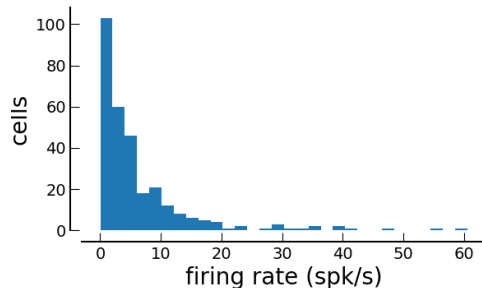# Example: summary of recordings from all cells
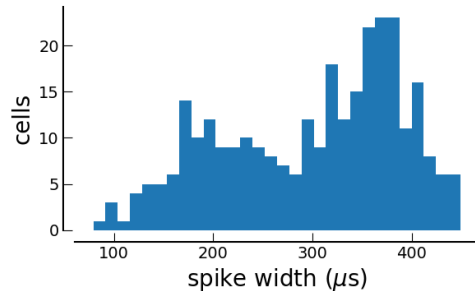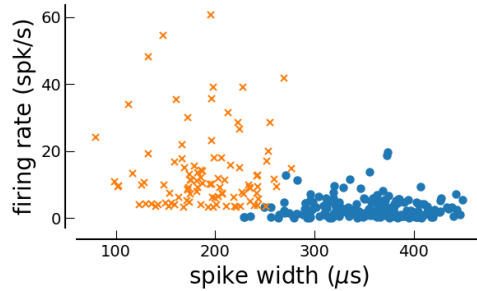
spike widths over firing rates
(all cells)

spike widths over firing rates :
using additional information of cell
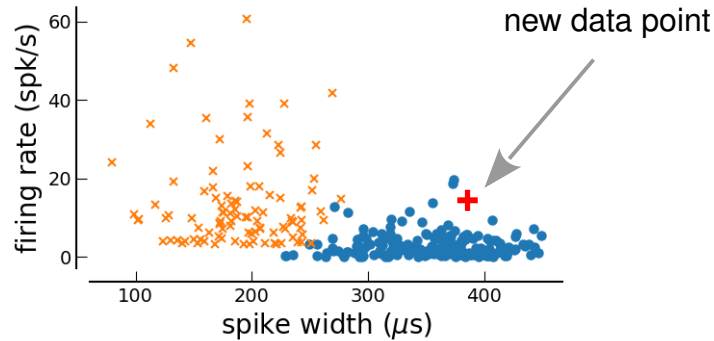identity from reconstruction

Averaging throws away specific information:

- ok if one averages over members of the same group
- problematic if not, looking for groups before averaging

➡ need to identify sub-groups in order to refine analysis and obtain deeper insights

Aim : classification and clustering

Classification is the task of predicting the group of new data points based on past observations.
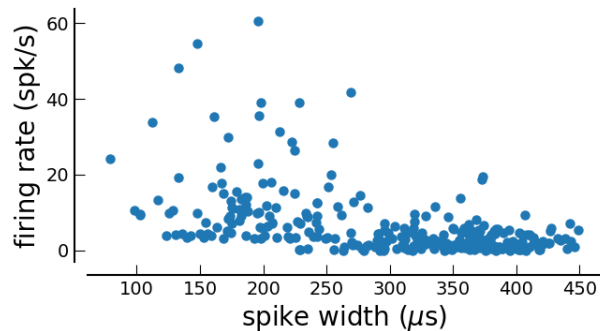
Clustering is the task of dividing the data into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters.

Classification vs. Clustering

Example : 2 classes with labels "pyramidal cells" and "basket cells"
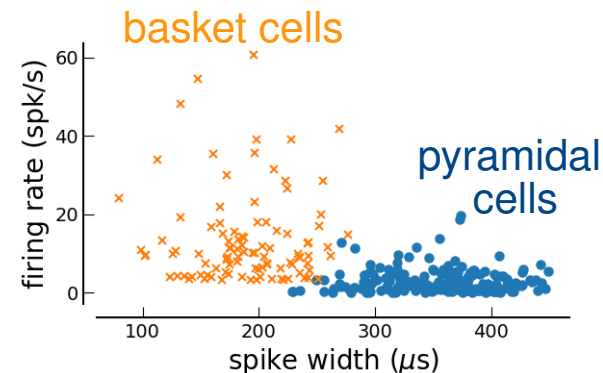
Classification	Clustering
supervised learning technique (labels are known)	unsupervised learning technique (labels are not known)
known number of classes	unknown number of classes
used to classify future observations	used to understand (explore) data structure

Classification: supervised learning

classes (output variables)

input variables

$$Y = f(X)$$

mapping function

Classification is where you have input variables (X) and output category/class (Y) and you use an algorithm to learn the mapping function (f) from the input to the output.

input variables: X		mapping function	classes: Y
spike width (µs)	firing rate (spk/s)	classifier	neuron type
335	5		Pyramidal neuron
167	34		Basket cell
212	43		Basket cell

⋮

Classification: supervised learning

input variables: X		mapping function	classes: Y
spike width (µs)	firing rate (spk/s)	classifier	neuron type
335	5	⟶	Pyramidal neuron
167	34	⟶	Basket cell
212	43	⟶	Basket cell

⋮

**Classification aim :** Learn mapping to predict neuron type of new recordings.

412	12	⟶	**????**

# Classification method 1: Logistic regression
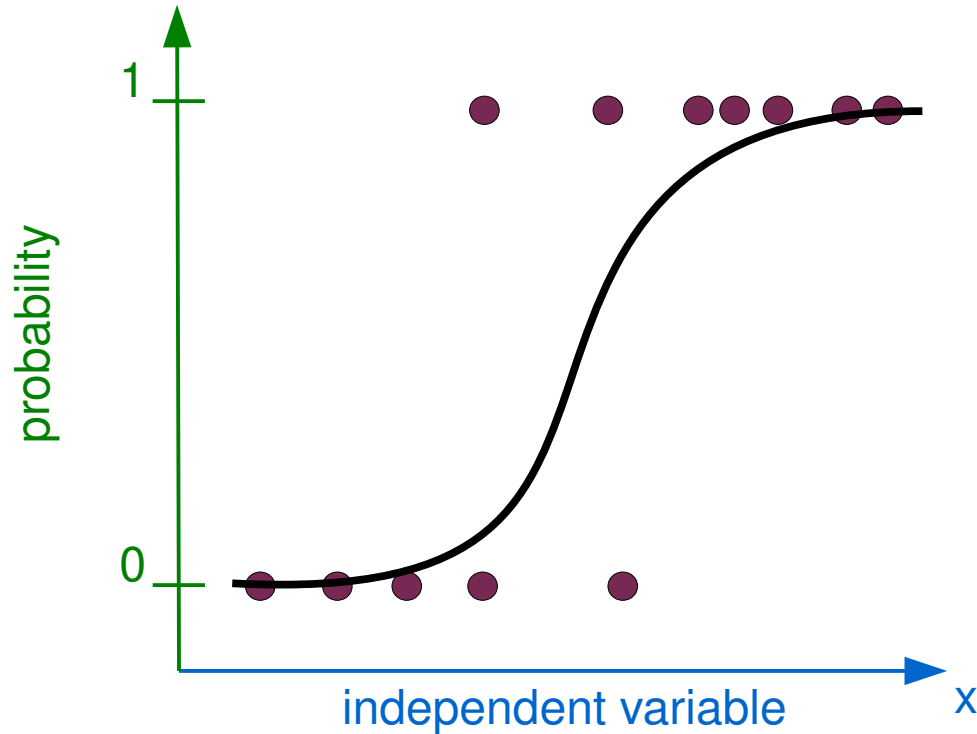
Logistic regression is a nonlinear model to link predictors and outcomes through a *sigmoidal* function. It gives the *odds* that an outcome happens – vs. it not happening for a given value of the independent variable (predictor value).
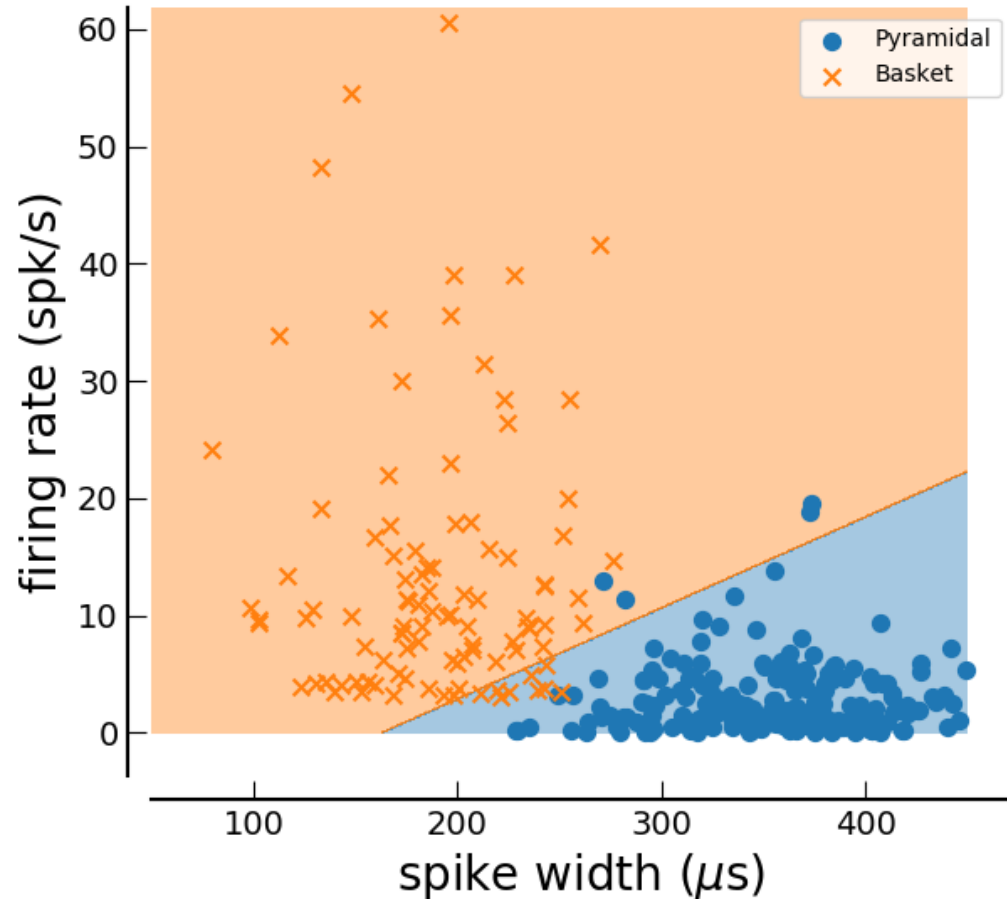
Logistic regression and our example
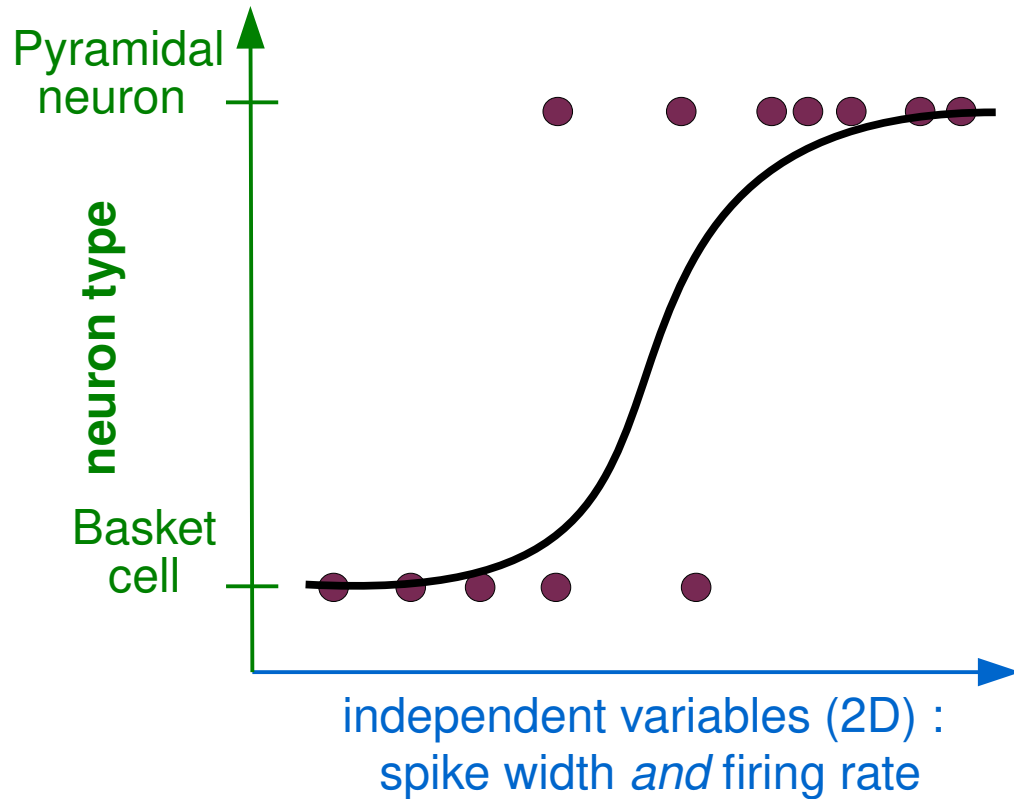
Task : fit the sigmoidal function to predict likelihood of neuron type.

Logistic regression and our example
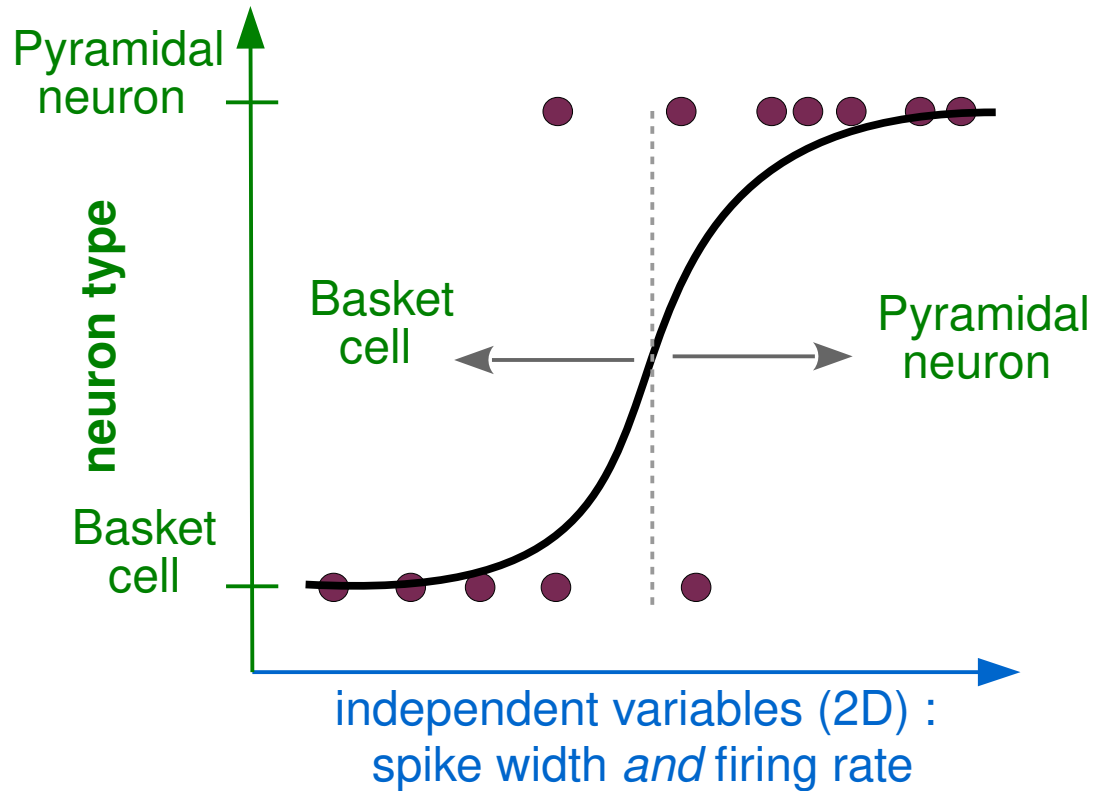
**Task :** fit the sigmoidal function to predict likelihood of neuron type.

Logistic regression on recording example

- does a pretty good job of separating the two neuron groups based on spike width and firing rate
- logistic regression give probability of being in either group (here we plot which of the groups the points are more likely to belong to)
- one-dimensional *linear* classifier

Linear vs. nonlinear separation of groups

Linear
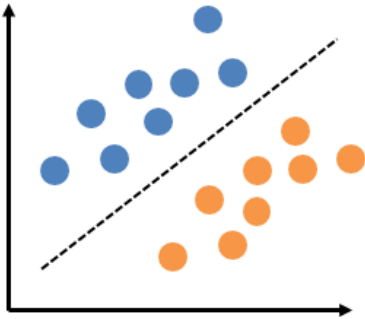
Nonlinear
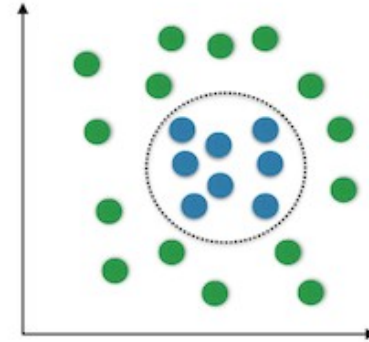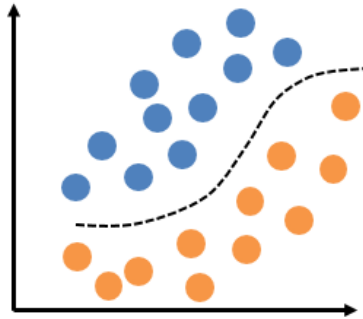
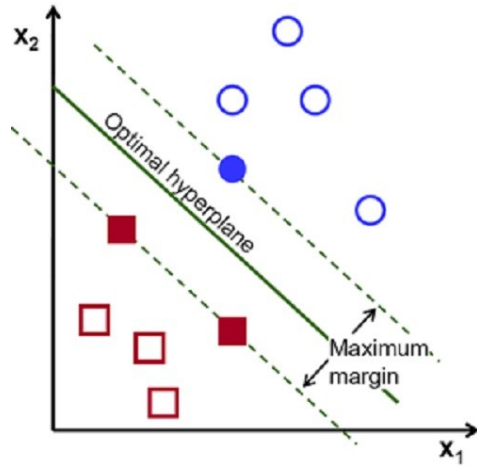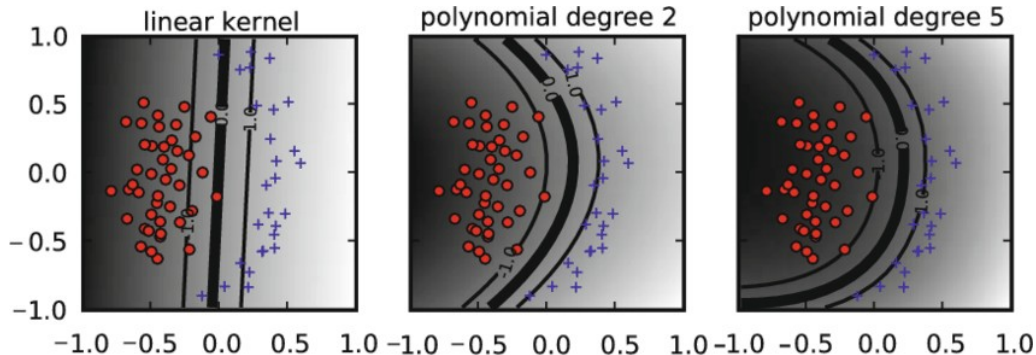# Classification method 2: support-vector machine

Support vector machine (SVM) : find the separating line (in higher dimensional space : hyperplane) that is equidistant to the two classes of datapoints, such that it maximizes the distance btw. the hyperplane and the nearest datapoint on each side.

The SVM classifier can be linear (finding a hyperplane) or non-linear using different kernels (e.g. polynomials).

Classification method 2: support-vector machine
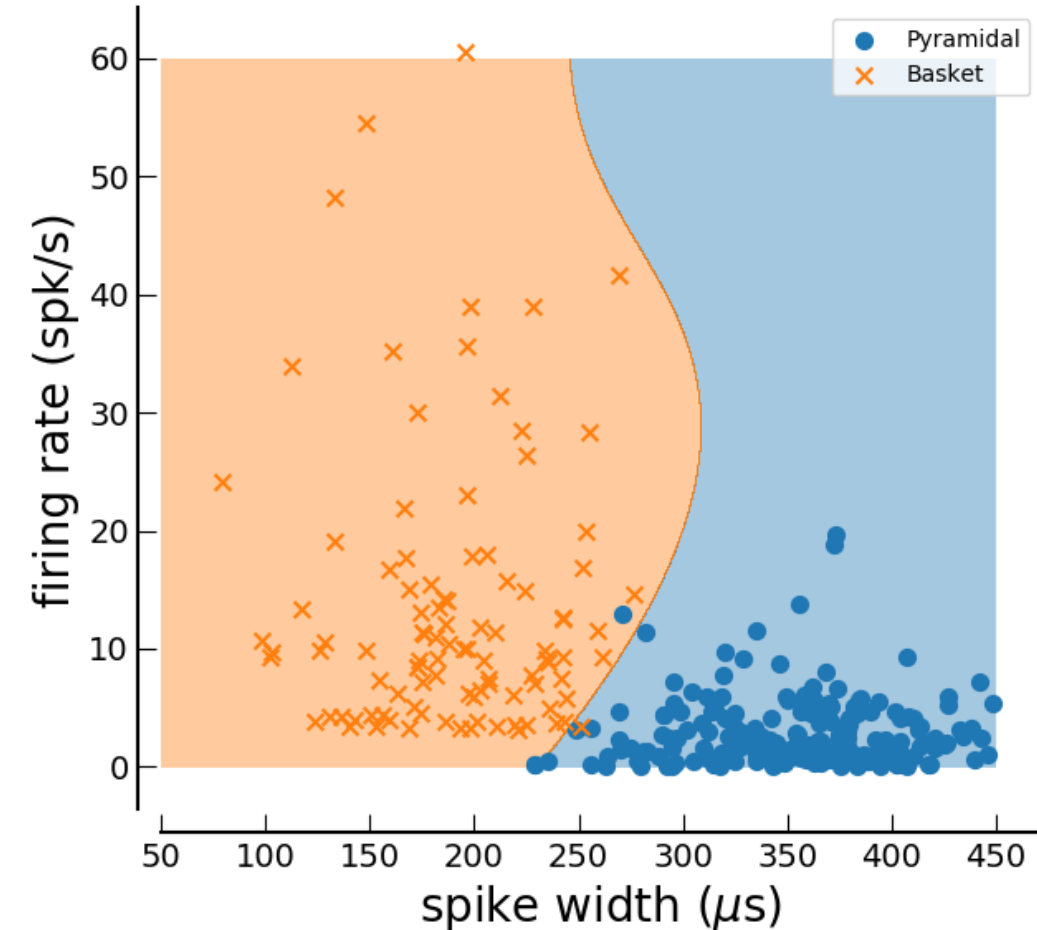
SVM classification : we use *polynomial kernel* which finds a higher-order polynomial that separates the classes of points.

Note : SVM classification can be linear in which case the algorithm finds a line to separate the classes.

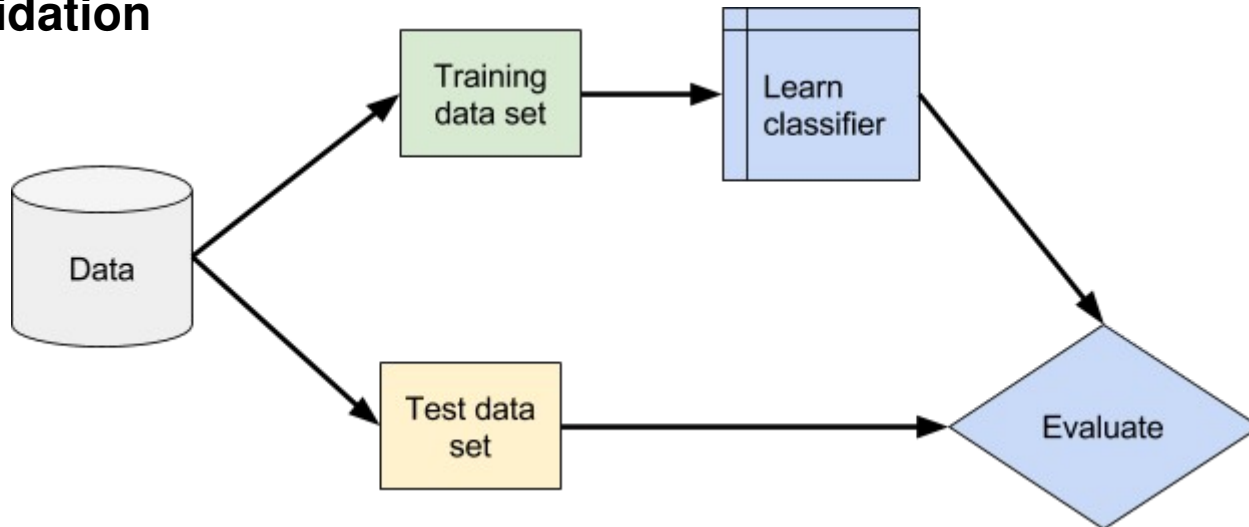# Predictions, validation and cross-validation

- How good is the classification ?
- How do we know that the separation is true ?
- How do we know that the classifier didn't just over-fit the data ?
  (over-fitting : the production of an analysis that corresponds too closely or
  exactly to a particular set of data, and may therefore fail to classify
  additional data reliably)

→ **Cross-validation**

Predictions, validation and cross-validation

- data is split in training (90 %) and test (10 %) set
- training set is used to build classifier
- test set (never seen by classifier) is used to check whether prediction is correct
- this process is typically repeated multiple times with different splits between training and test data
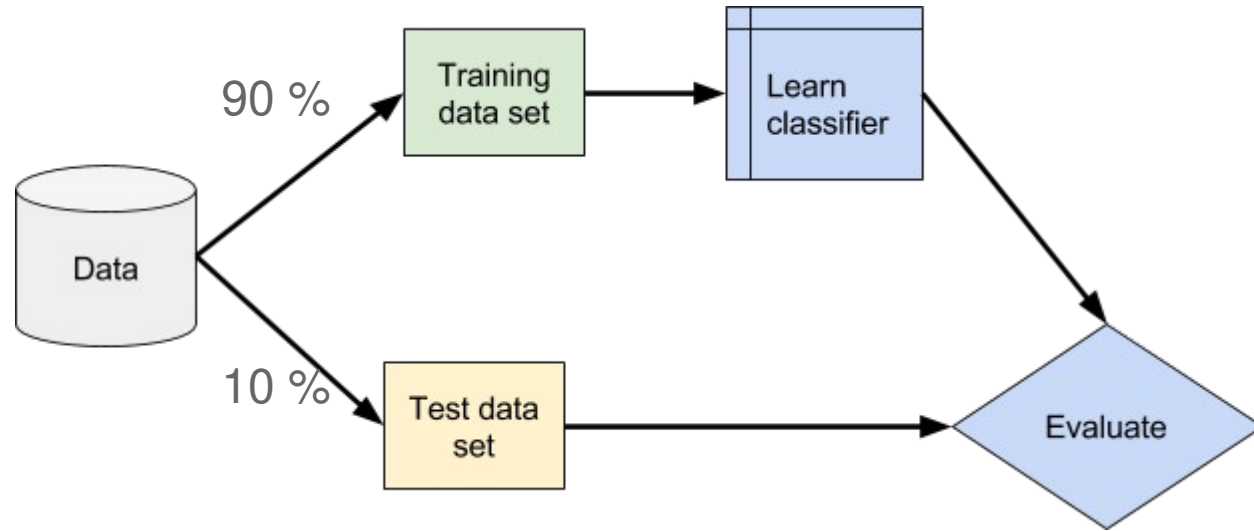
Clustering

| Classification	Clustering
supervised learning technique (labels are known)	unsupervised learning technique (labels are not known)
known number of classes	unknown number of classes
used to classify future observations	used to understand (explore) data

We know the classes.   Have to figure out classification.

Clustering method: k-means clustering

k-means on our example (k=2)

basket cell
pyramidal cell
miss-classified
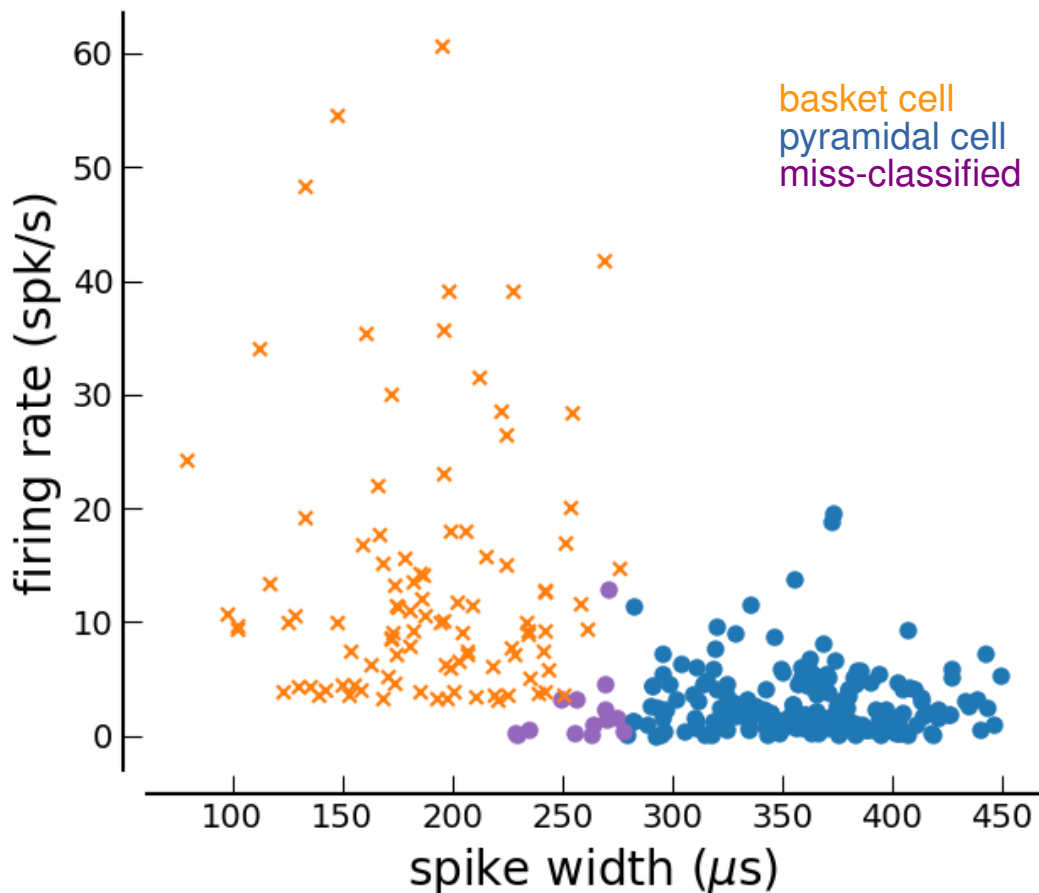
- **k-means :** algorithm is an iterative algorithm that tries to partition the dataset into *K pre-defined* distinct non-overlapping clusters
- each data point belongs to only one cluster
- It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum

k-means clustering: how many clusters to use ?

k-means on our example (k=2)

basket cell
pyramidal cell
miss-classified
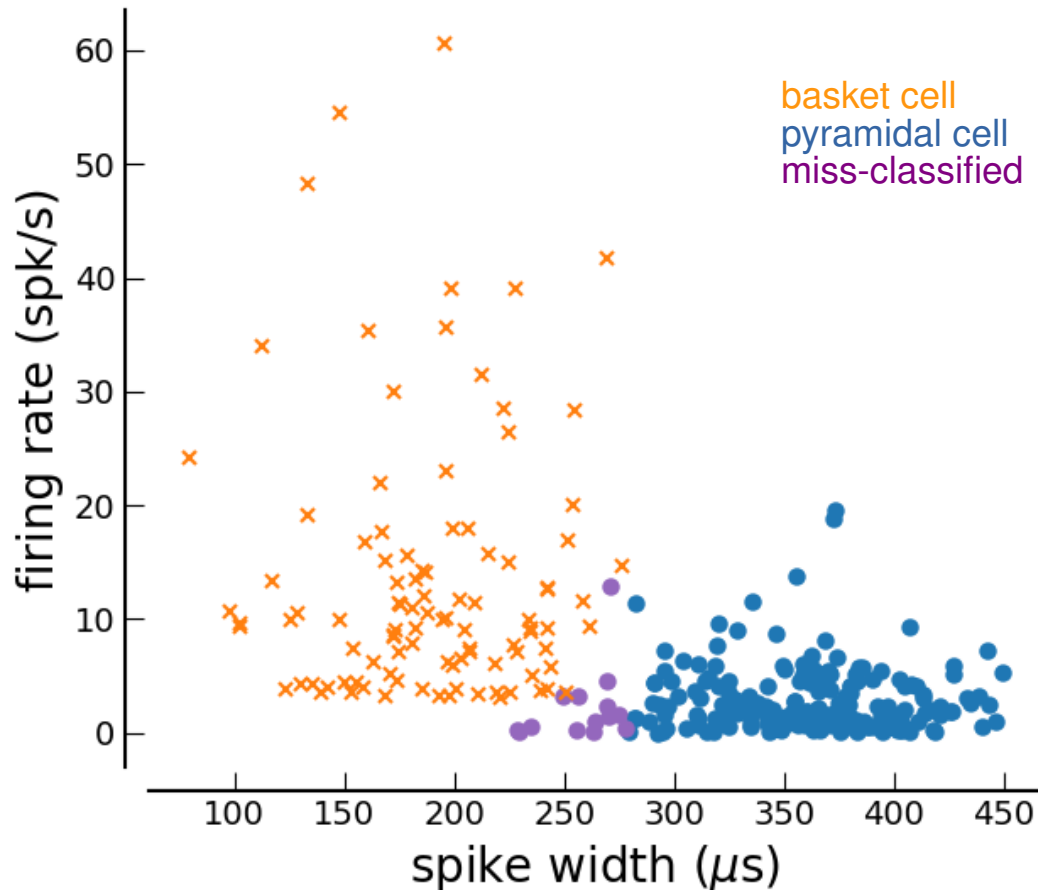
- **k-means :** tells you were the cluster centers (centroids) are but NOT how many there are

- **problem :** in actual data, you don't know the number of clusters

- **solution :** *silhouette analysis* to determine how many clusters are likely to be valid

k-means clustering : silhouette analysis
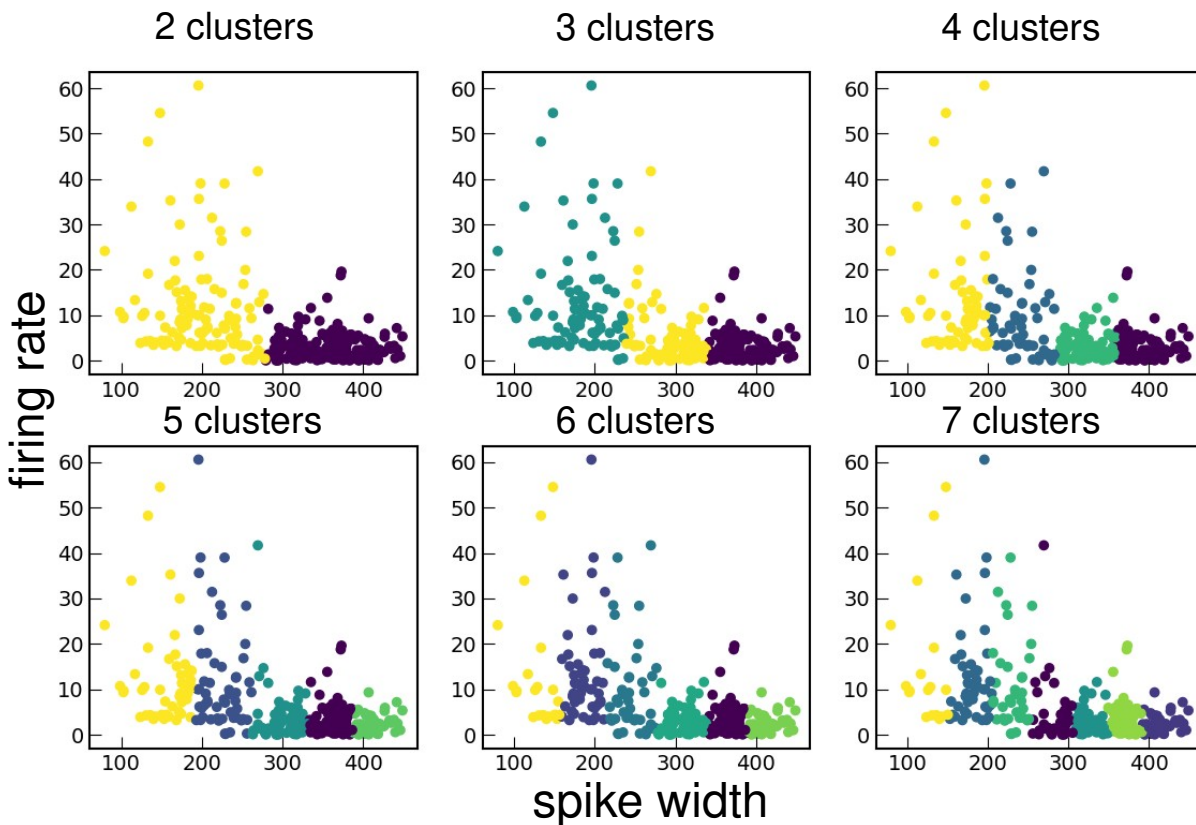
- Silhouette analysis gives a perspective into the density and separation of the formed clusters

- measures of how close each point in one cluster is to points in the same cluster compared to the neighboring clusters

- values range between [-1, 1] :
  - '1' good separation between clusters
  - '0' classification is almost arbitrary
  - '-1' indicates missclassification

k-means clustering : silhouette analysis
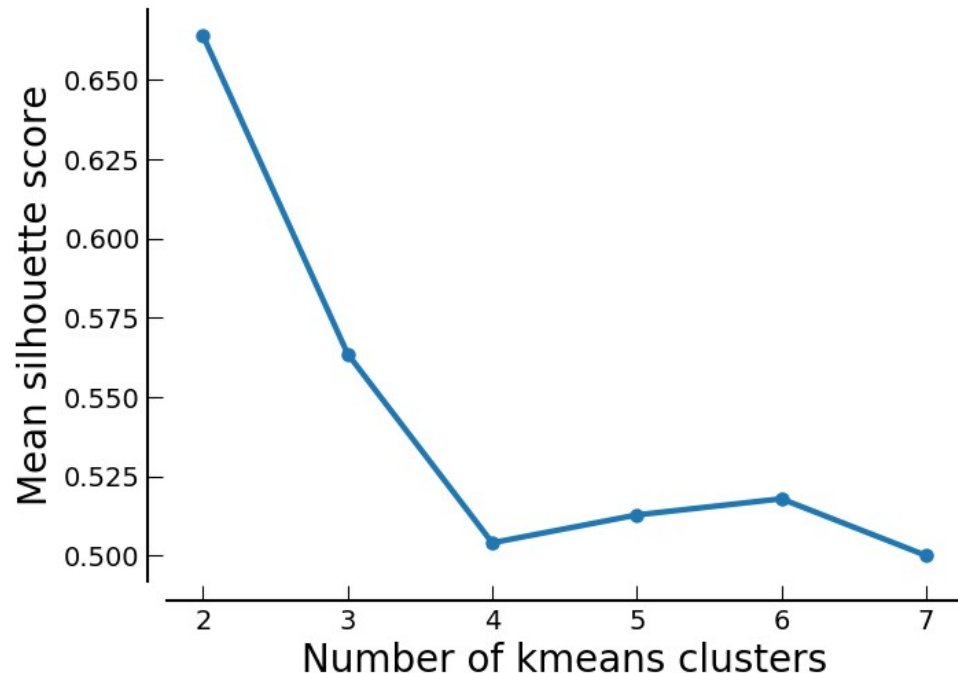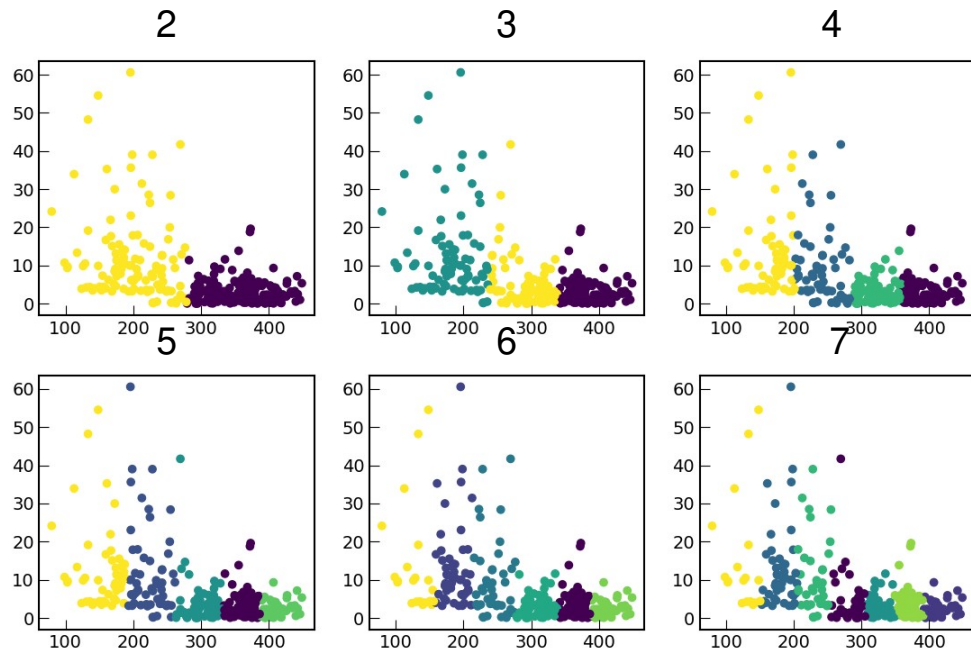
- for our example : 2 clusters yield the highest, average silhouette score
  → best separation with 2 clusters