Classification: Nearest Neighbors & Support Vector Machines

Spring 2018

Review

- Last week:
  - Classification Applications
  - Introduction to Probability
  - Decision Tree (Discriminative Model)
  - Naïve Bayes (Generative Model)
  - Evaluating Classification Models

- Assignments (Canvas)
  - Problem set 2 due yesterday
  - New lab assignment out and due next week

- Questions?

Today’s Topics

- Nearest Neighbor Classifier
- Support Vector Machines
- Evaluating Classifiers Using Cross-Validation
- Tuning Hyper-parameters
- Lab

Today's Topics

- Nearest Neighbor Classifier
- Support Vector Machines
- Evaluating Classifiers Using Cross-Validation
- Tuning Hyper-parameters
- Lab

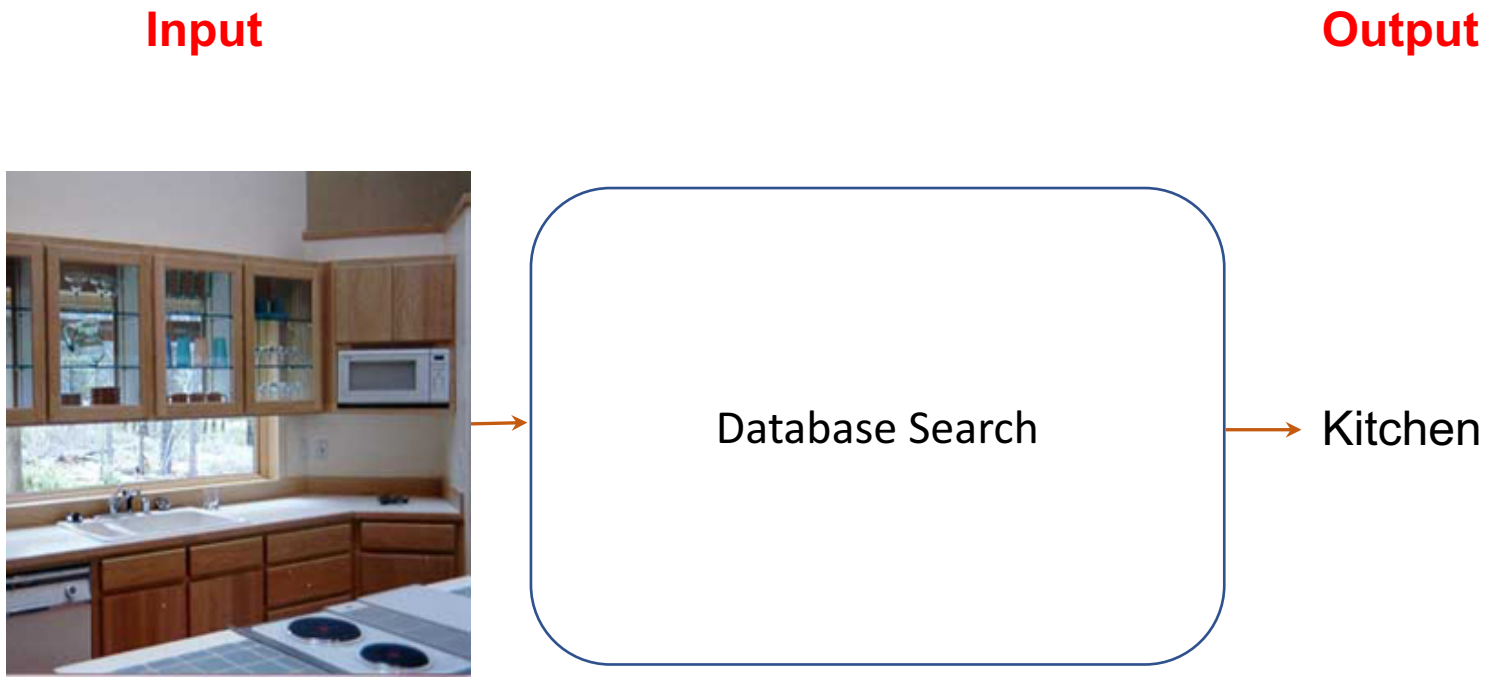
Recall: Instance-Based vs Model-Based Learning

Memorizes examples and uses a similarity measure to those examples to make predictions.

Build a model with the examples and use the model to make predictions.

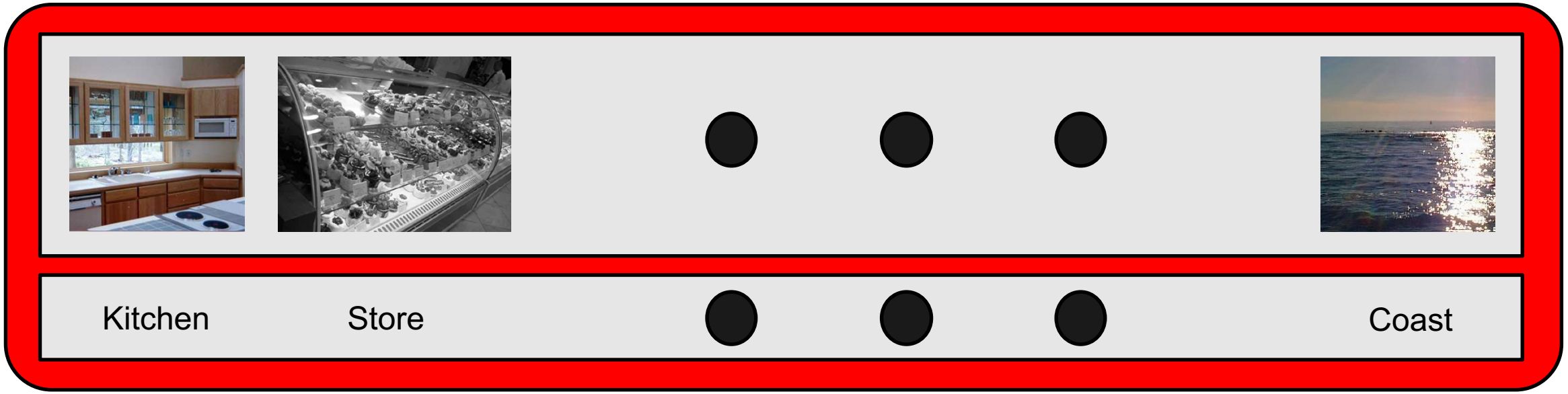
What is the difference between these learning styles?

e.g., Predict What Scene The Image Shows



Teaching Computers to Classify Scenes: Nearest Neighbor Classification

1. Create Large Database



Teaching Computers to Classify Scenes: Nearest Neighbor Classification

2. Organize Database so Visually Similar Images Neighbor Each Other

Adapted from slides by Antonio Torralba

Teaching Computers to Classify Scenes: Nearest Neighbor Classification

3. Predict Scene Using Label of Most Similar Image(s) in the Database

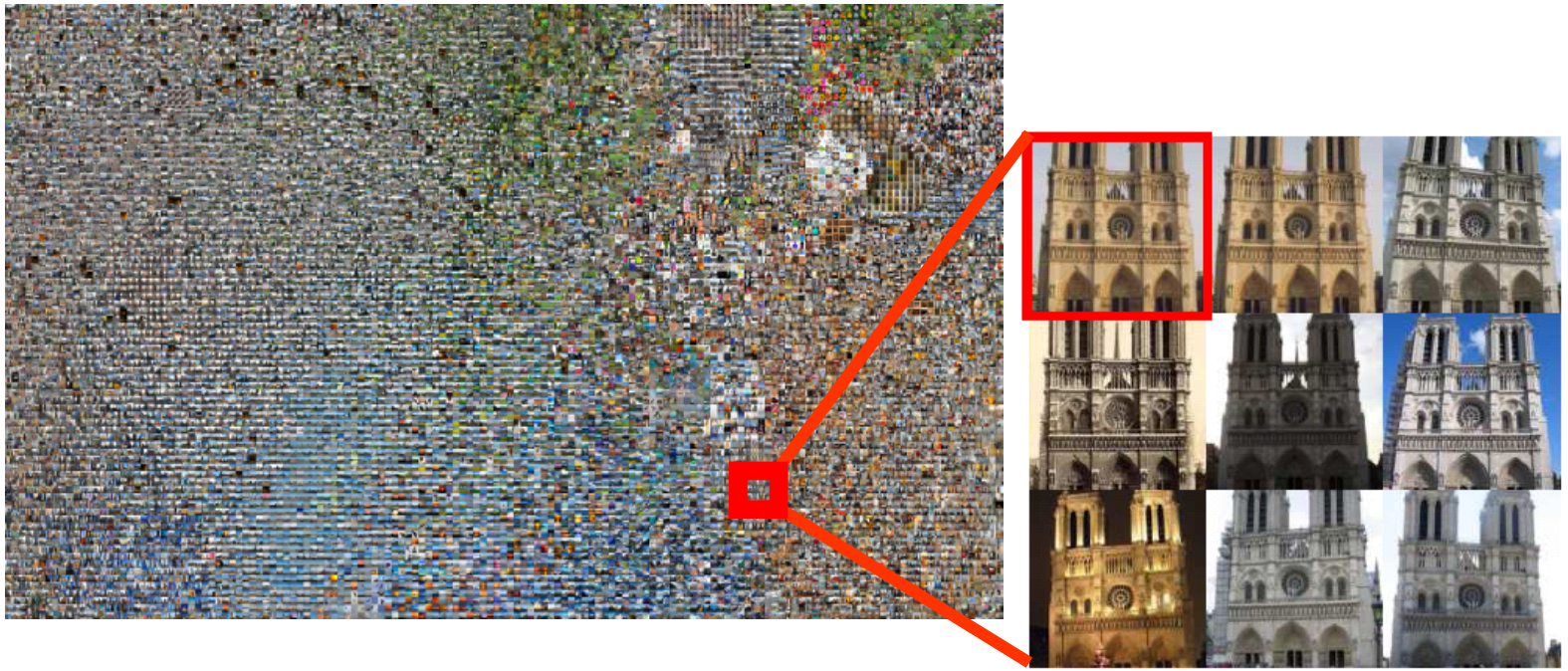
Input:

Label: ?

Adapted from slides by Antonio Torralba

Teaching Computers to Classify Scenes: Nearest Neighbor Classification

3. Predict Scene Using Label of Most Similar Image(s) in the Database



Input:

Label: Cathedral ✓

Adapted from slides by Antonio Torralba

Teaching Computers to Classify Scenes: Nearest Neighbor Classification

3. Predict Scene Using Label of Most Similar Image(s) in the Database

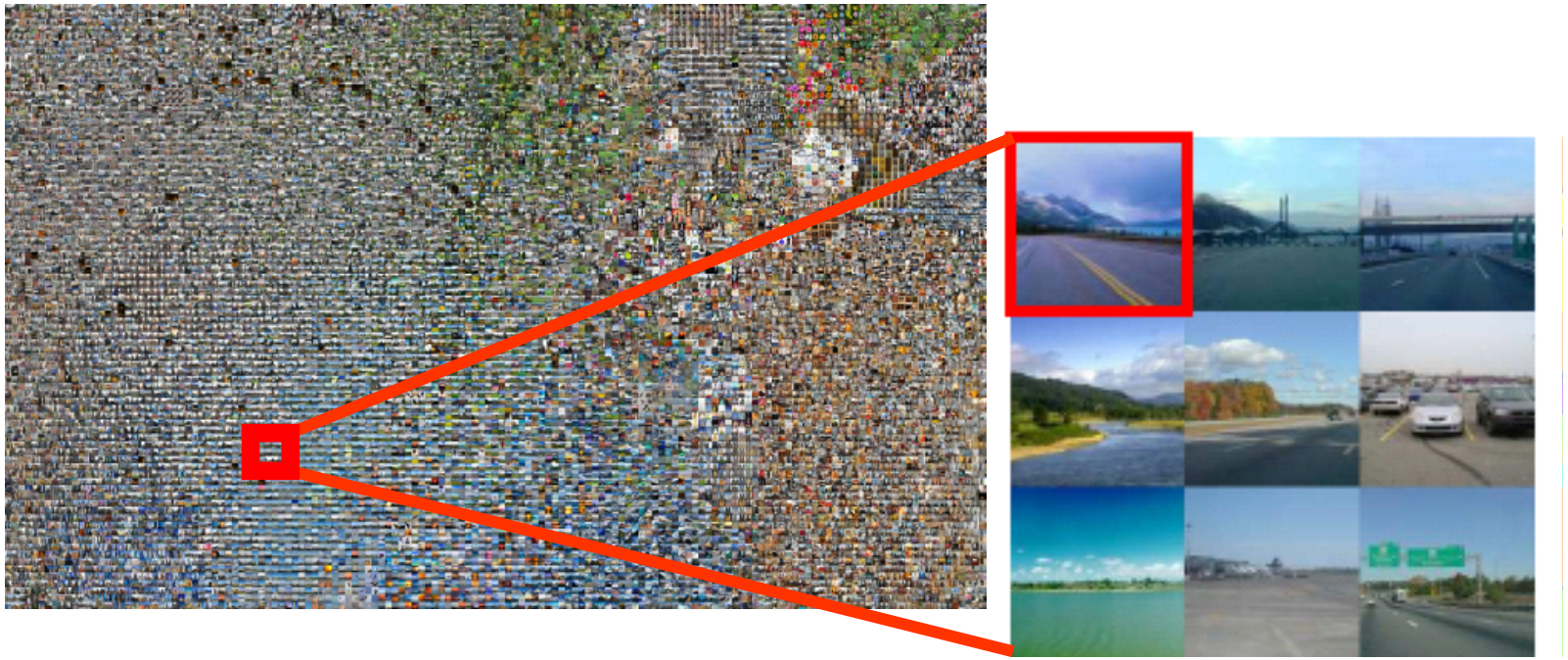
Input:

Label: ?

Adapted from slides by Antonio Torralba

Teaching Computers to Classify Scenes: Nearest Neighbor Classification

3. Predict Scene Using Label of Most Similar Image(s) in the Database



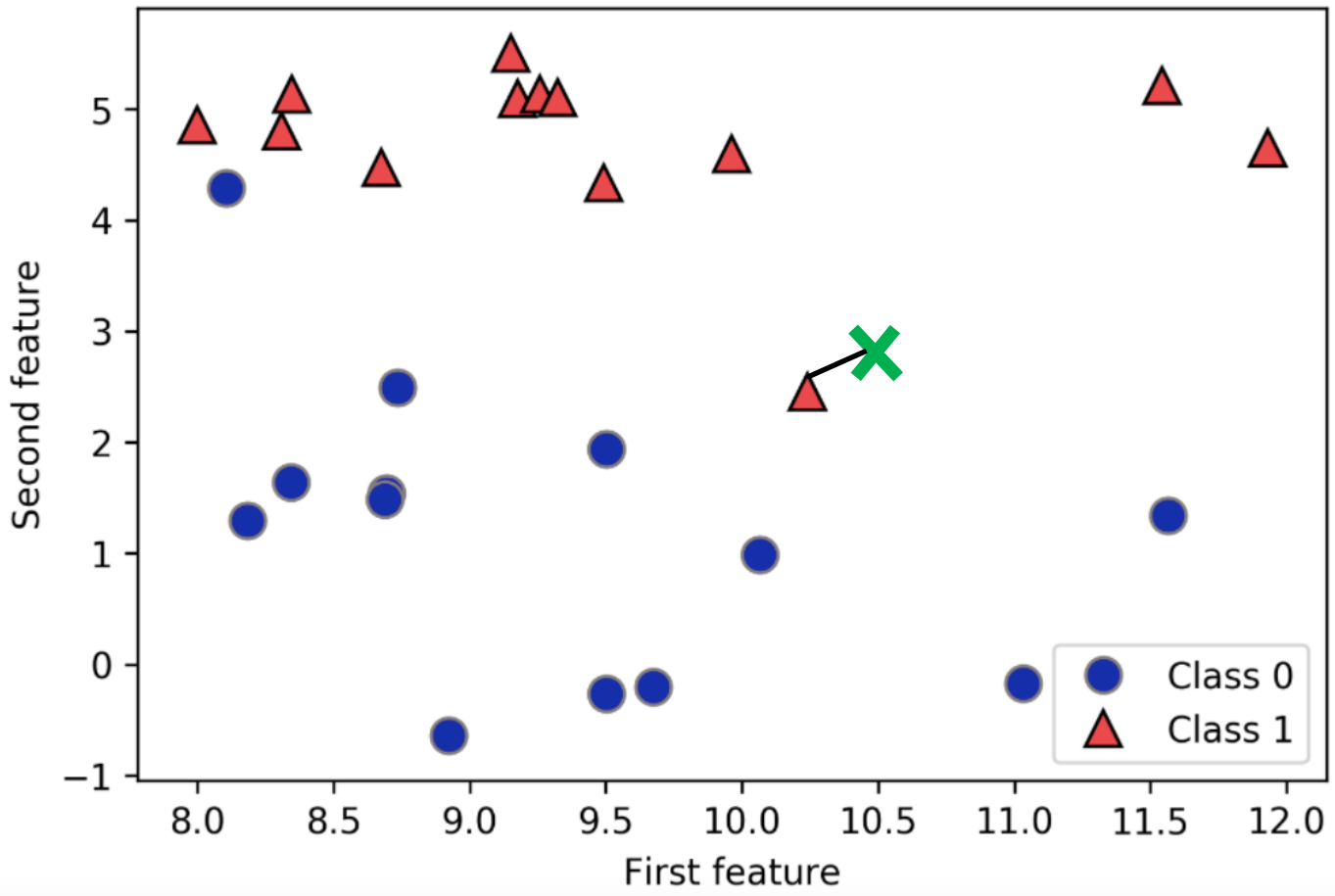
Input:

Label: Highway ✓

Adapted from slides by Antonio Torralba

K-Nearest Neighbor Classification

Training Data:



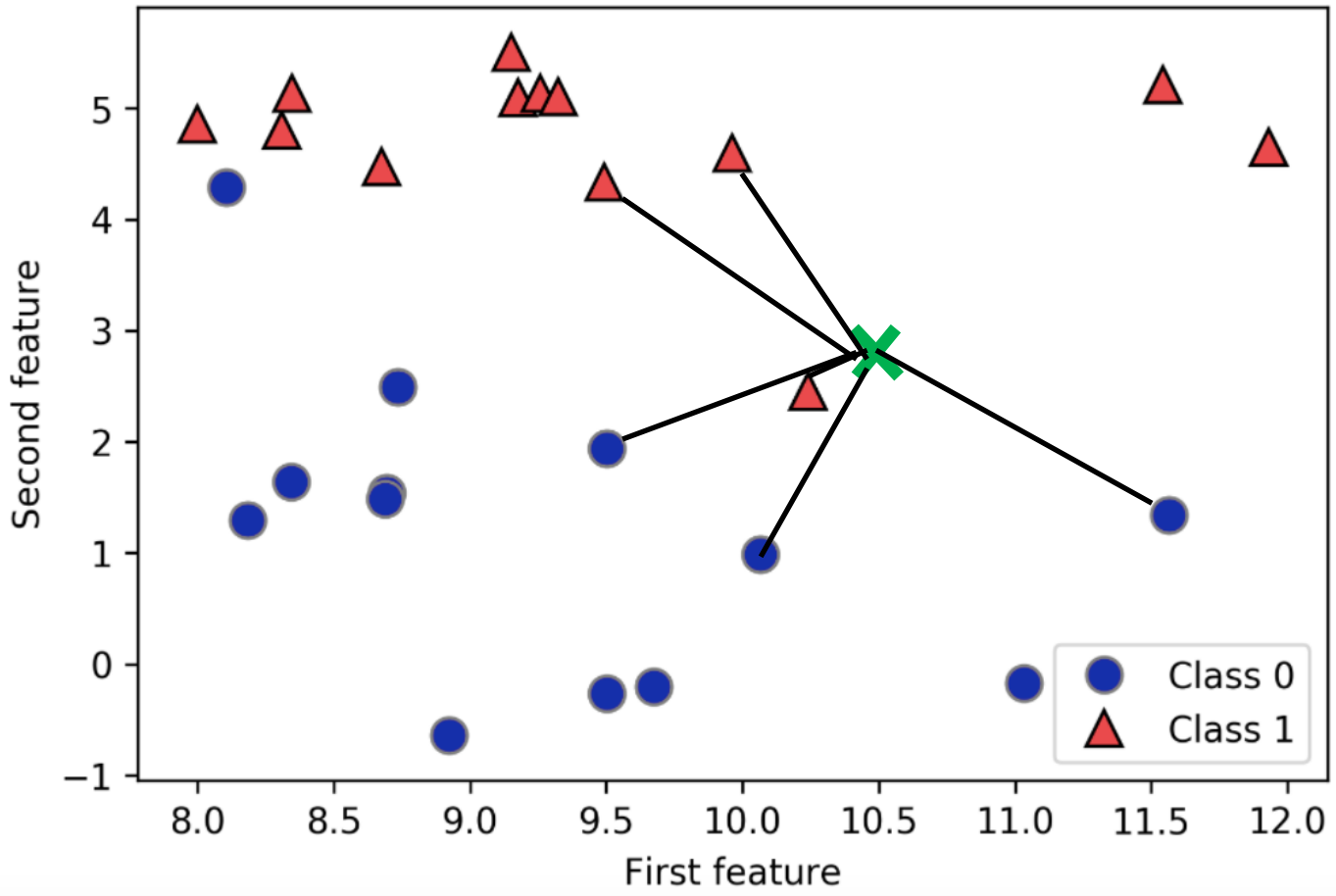
Novel Examples:

- Given:
  - **x = {10.5, 3}**
- Predict:
  - When k = **1:**

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification

Training Data:



Novel Examples:

- Given:
  - **x = {10.5, 3}**
- Predict:
  - When k = **1:**
    - Class 1
  - When k = **6:**
    - How to avoid ties?
      - Set “k” to odd value for binary problems
      - Prefer “closer” neighbors

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbors: Measuring Distance

How to measure distance between a novel example and test example?

- Commonly use, Minkowski distance:

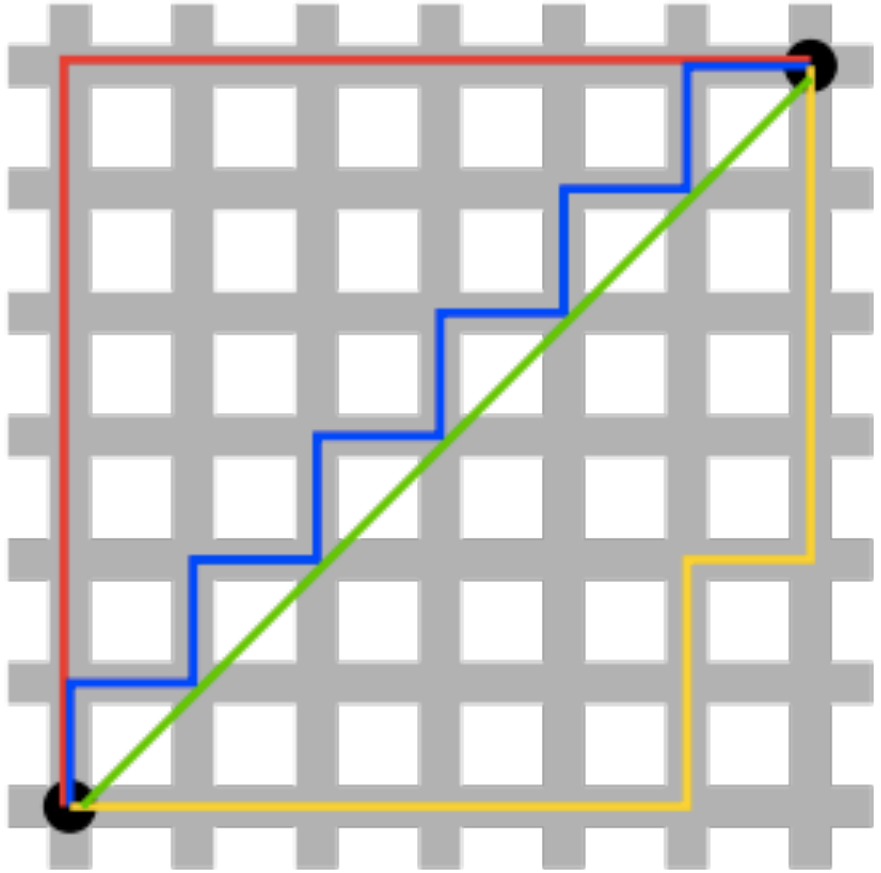
$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

  - When p = 2, Euclidean distance:

$$= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

  - When p = 1, Manhattan distance:

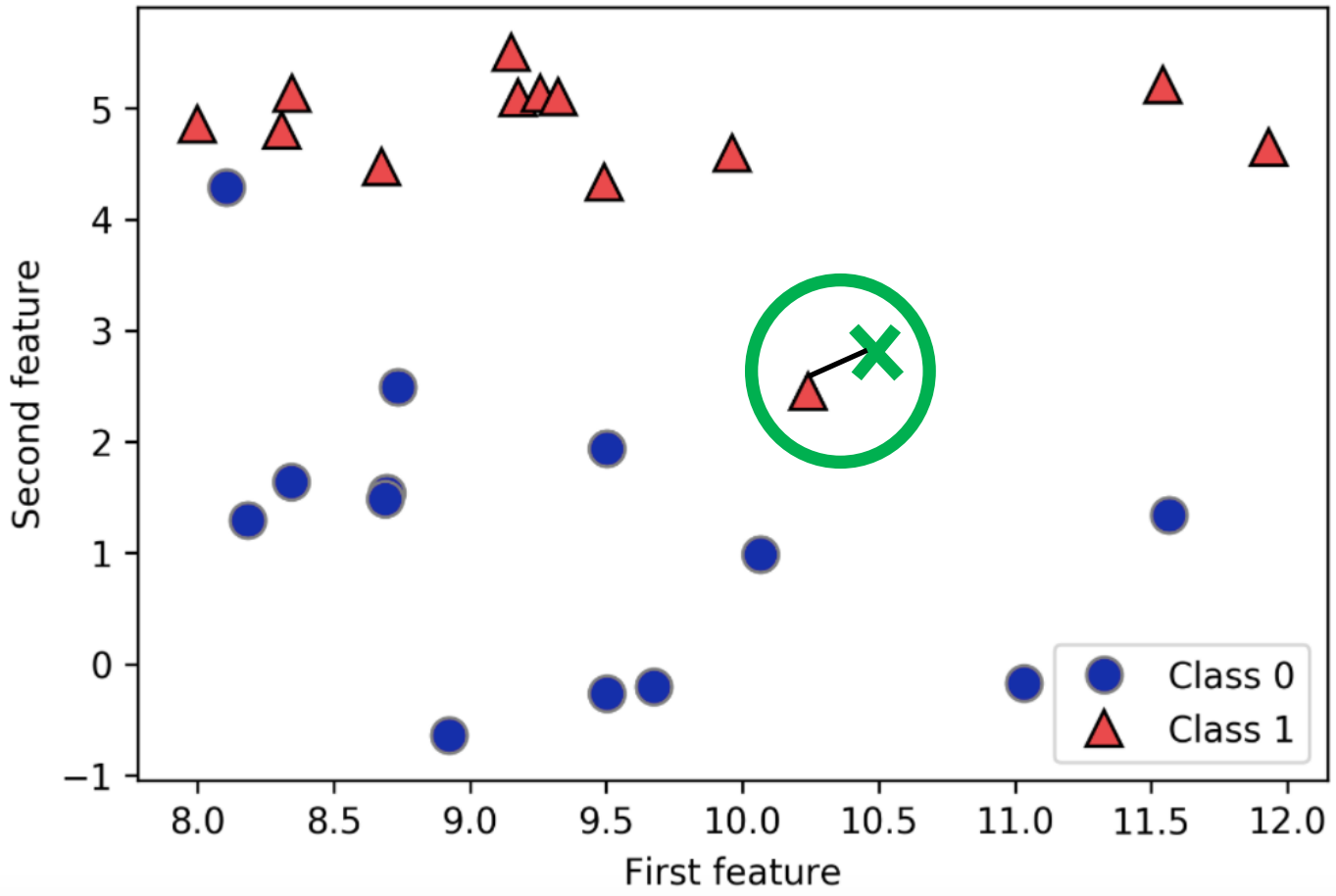
$$= \sum_{i=1}^{n} |x_i - y_i|$$



https://en.wikipedia.org/wiki/Taxicab_geometry

K-Nearest Neighbors: Measuring Distance

Training Data:



Euclidean Distance

- Given:
  - **x = {10.5, 3}**
- k =1:

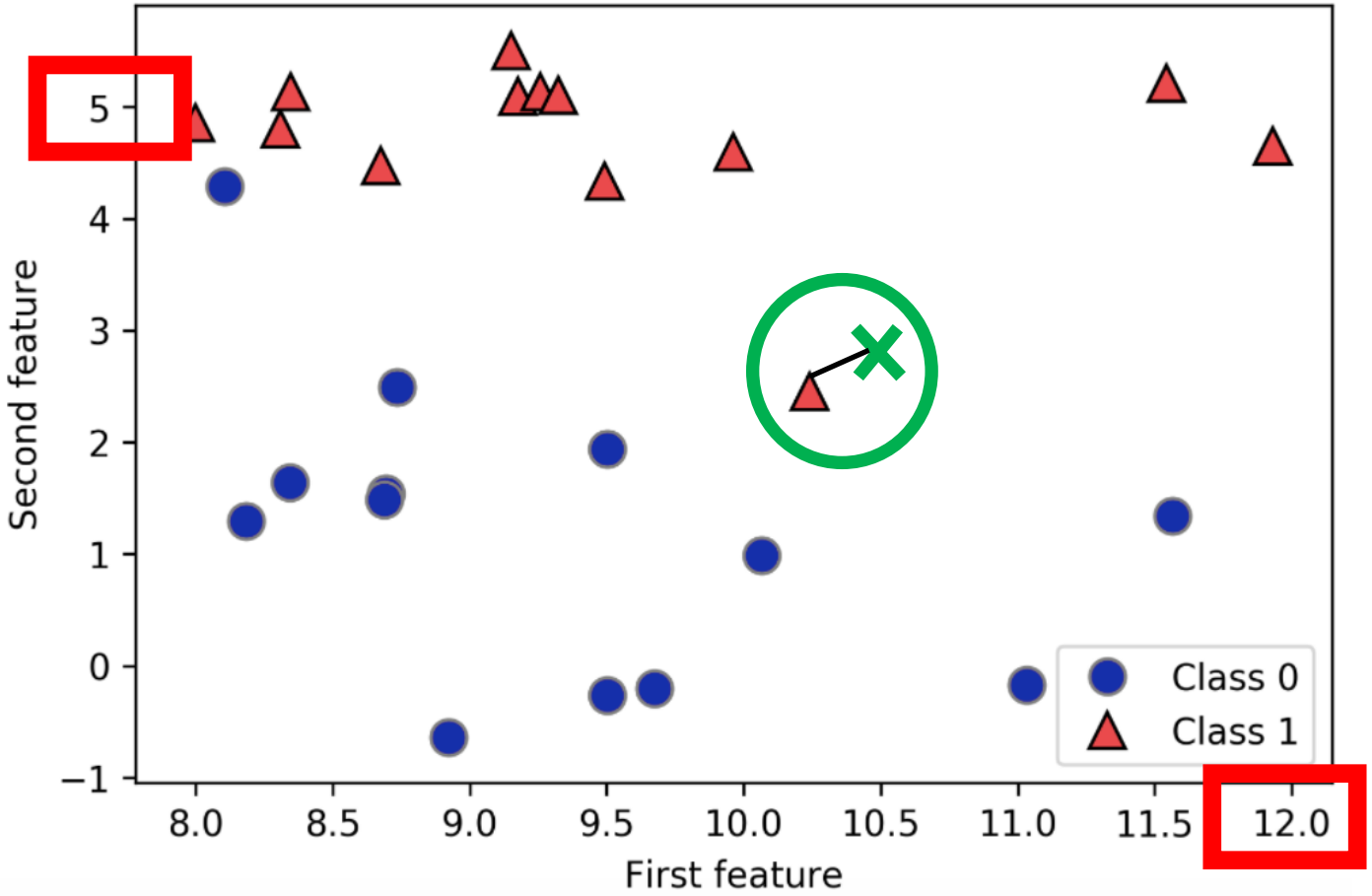
$$= \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$dist = \sqrt{(10.5 - 10.1)^2 + (3 - 2.3)^2}$$
$$dist = \sqrt{0.4^2 + 0.7^2}$$
$$dist = \sqrt{0.16 + 0.49}$$
$$dist = \sqrt{0.65}$$
$$dist = 0.81$$

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbors: Measuring Distance

Training Data:



Euclidean Distance

- Given:
  - **x = {10.5, 3}**
- k =1:

$$= \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$dist = \sqrt{(10.5 - 10.1)^2 + (3 - 2.3)^2}$$

$$dist = \sqrt{0.16 + 0.49}$$

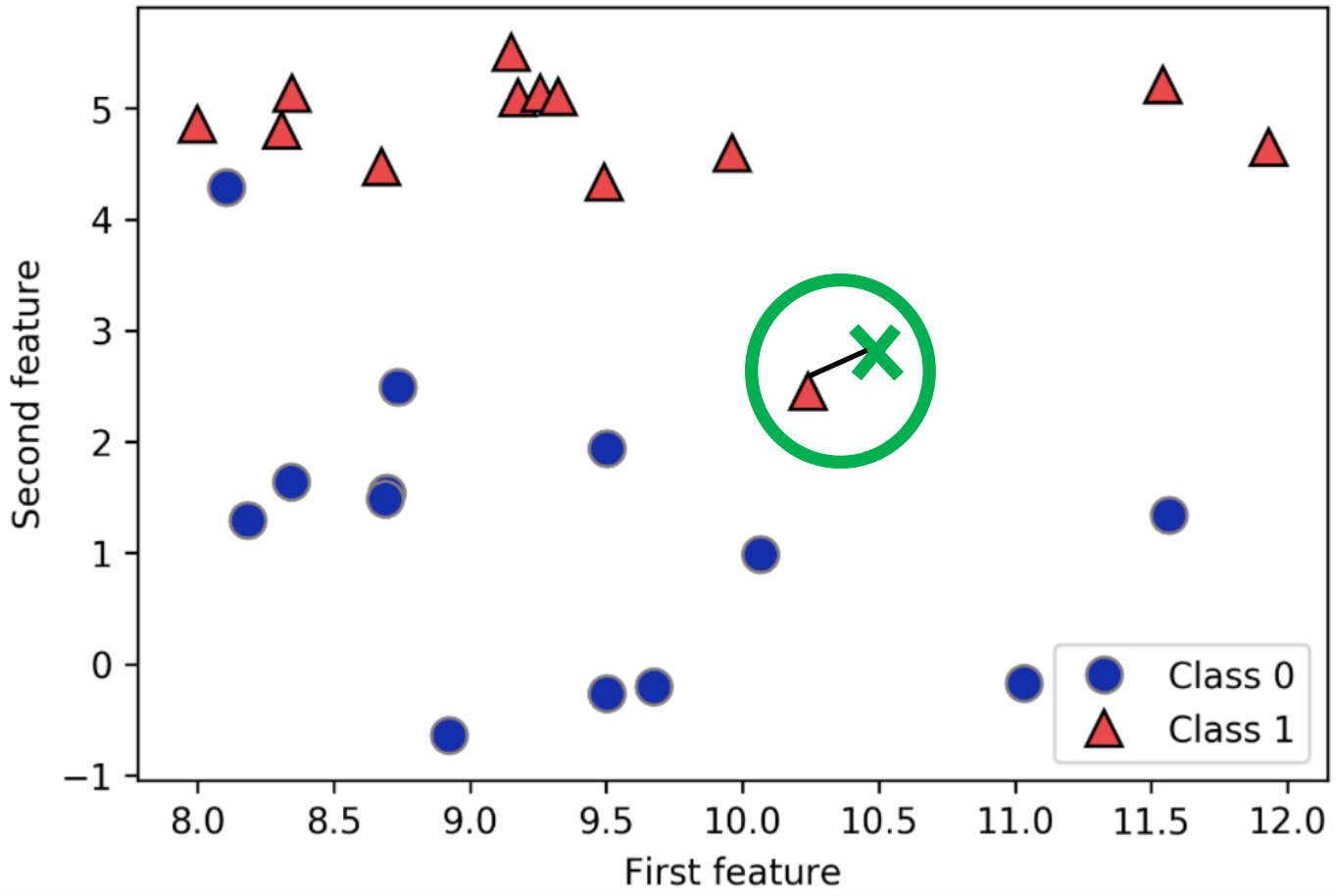
$$dist = 0.81$$

Note: Data should first be scaled to same range

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbors: Measuring Distance

Training Data:



Manhattan Distance

- Given:
  - **x = {10.5, 3}**
- k =1:

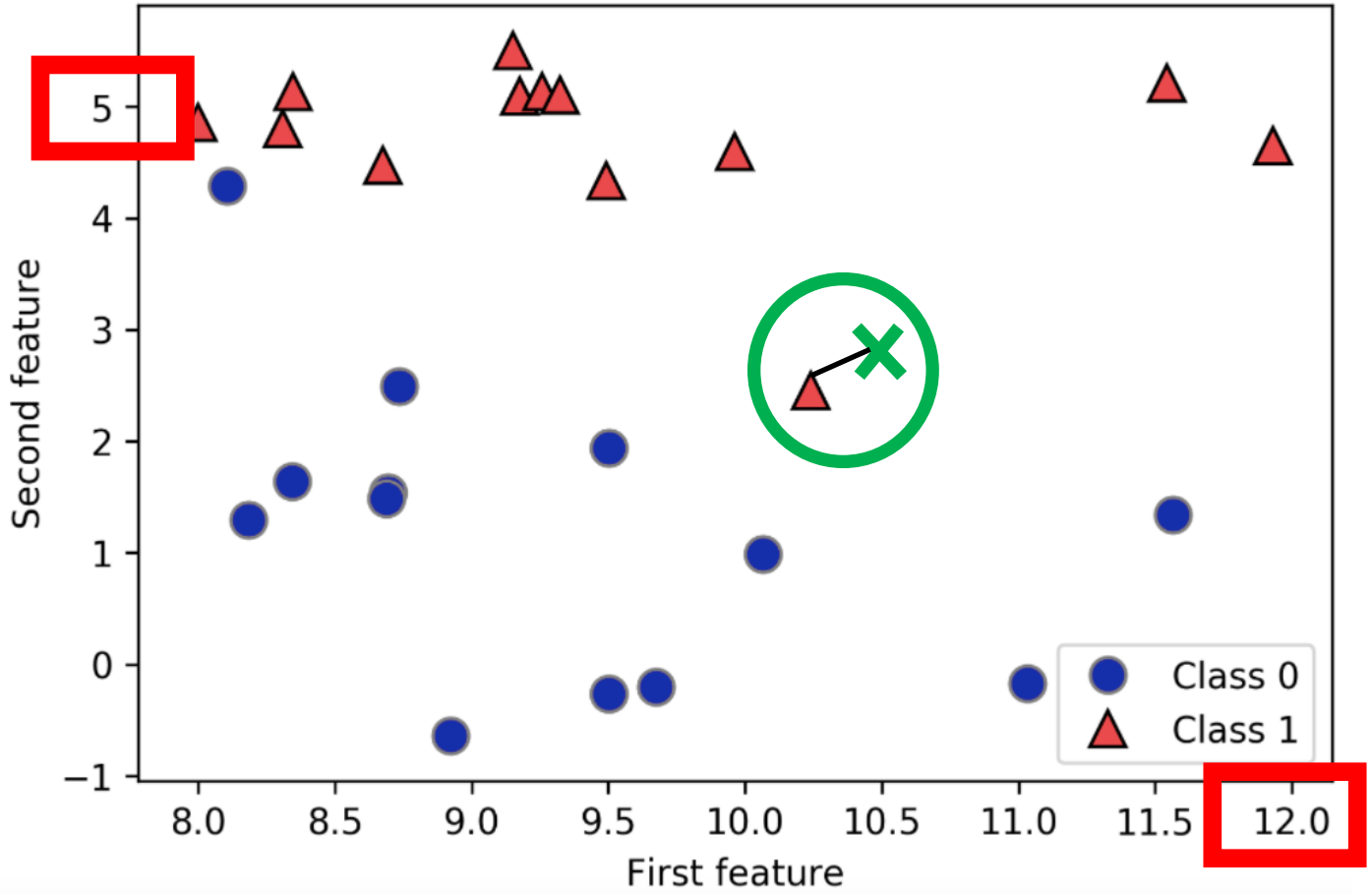
$$= \sum_{i=1}^{n} |x_i - y_i|$$

$$dist = |10.5 - 10.1| + |3 - 2.3|$$
$$dist = 0.4 + 0.7$$
$$dist = 1.1$$

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbors: Measuring Distance

Training Data:



Manhattan Distance

- Given:
  - **x = {10.5, 3}**
- k =1:

$$= \sum_{i=1}^{n} |x_i - y_i|$$

Note: Data should first be scaled to same range

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbors: Measuring Distance

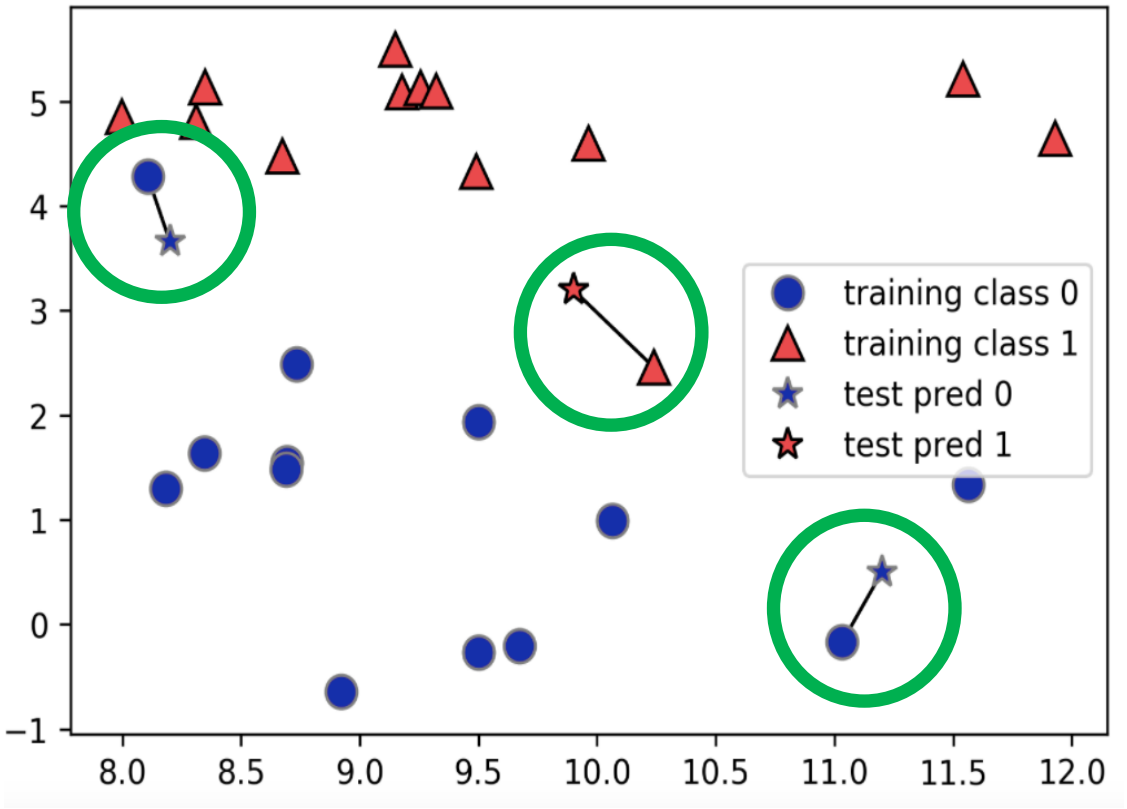
How to measure distance between a novel example and test example?

- For categorical data:
  - e.g., Train = blue
  - e.g., Test = blue; identical values so assign distance 0
  - e.g., Test = white; different values so assign distance 1

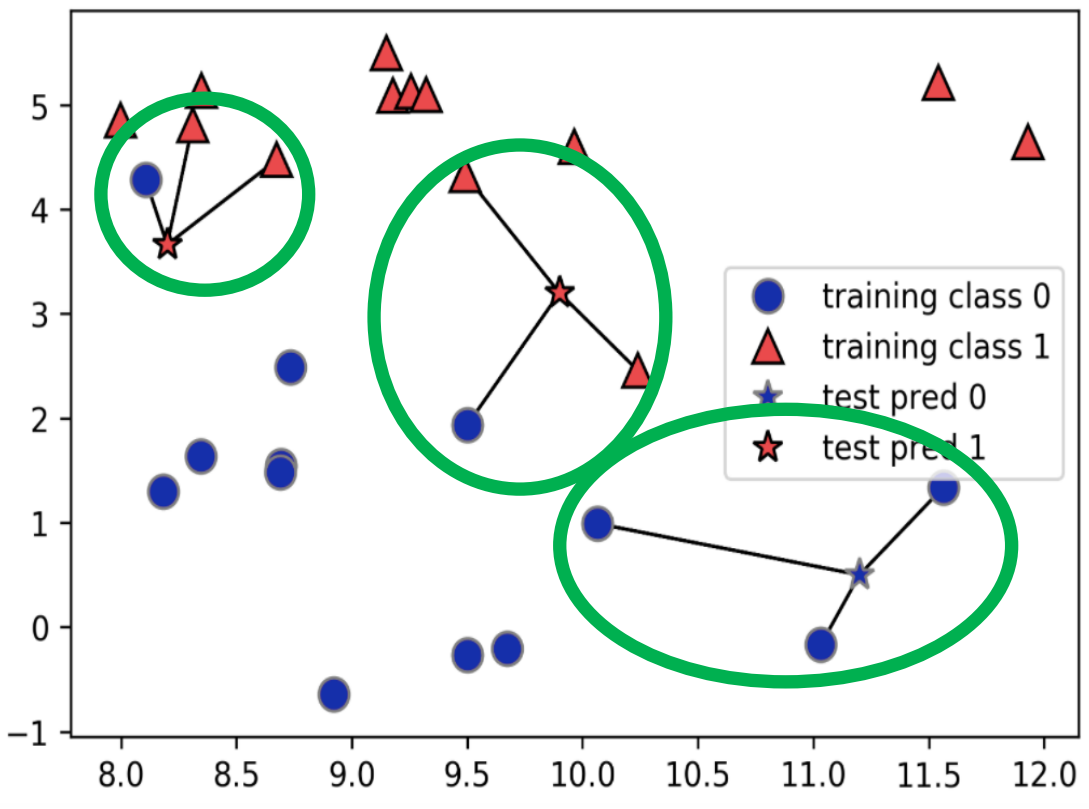
https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification: What "K"?

When K=1:



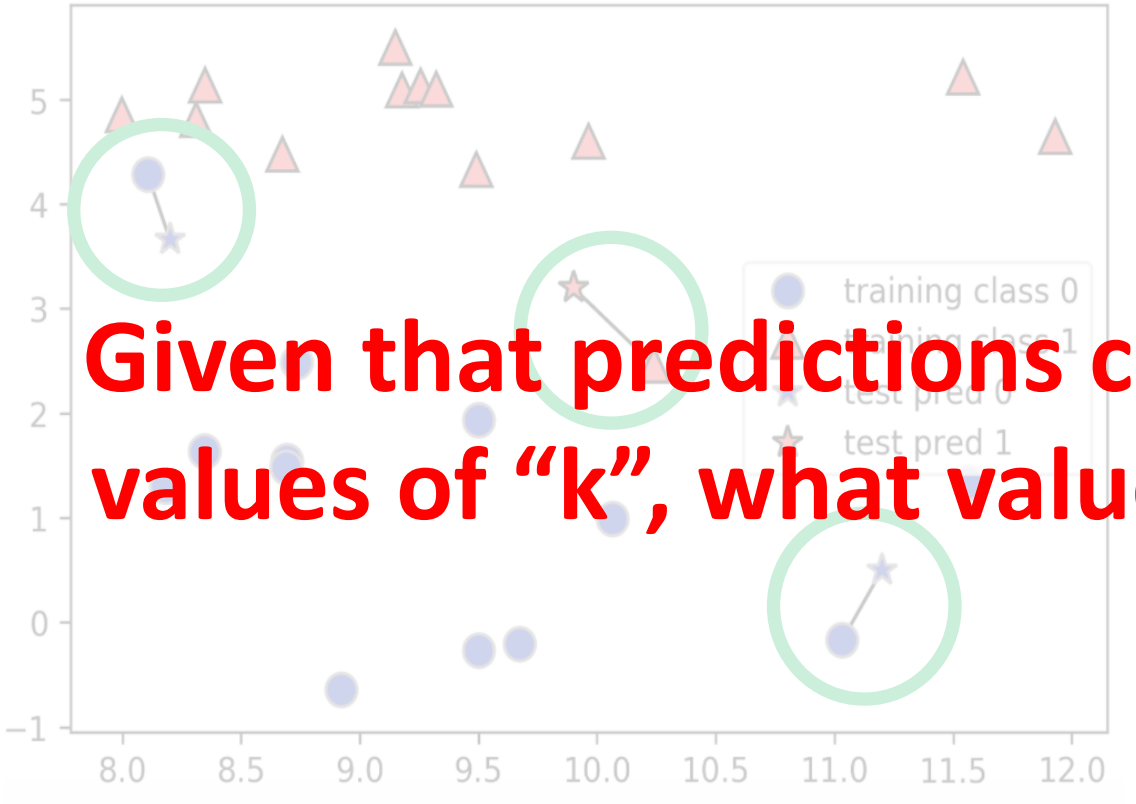
When K=3:



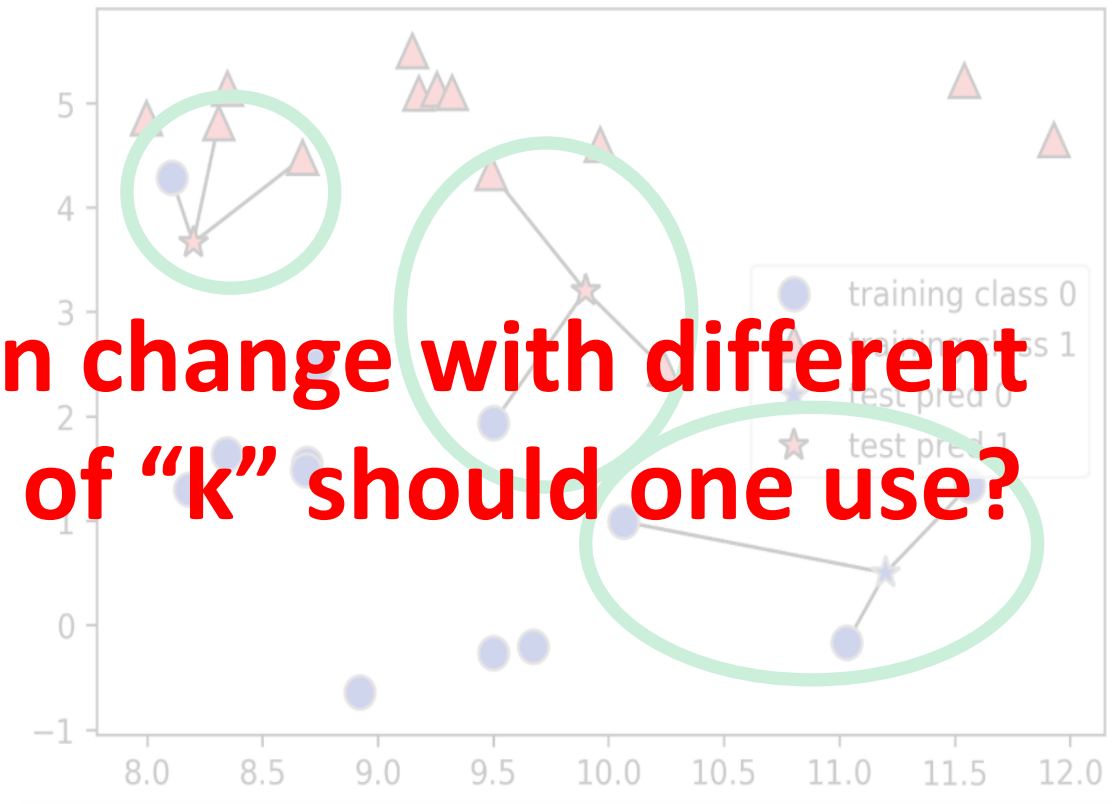
https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification: What “K”?

When K=1:



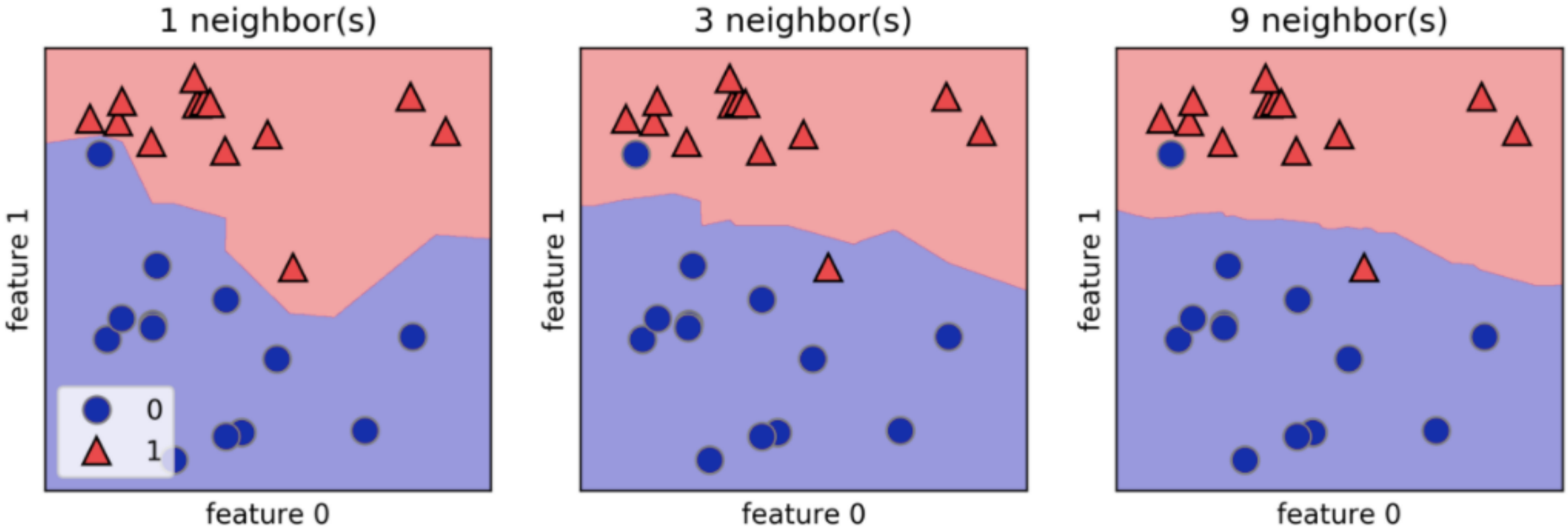
When K=3:



Given that predictions can change with different values of “k”, what value of “k” should one use?

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

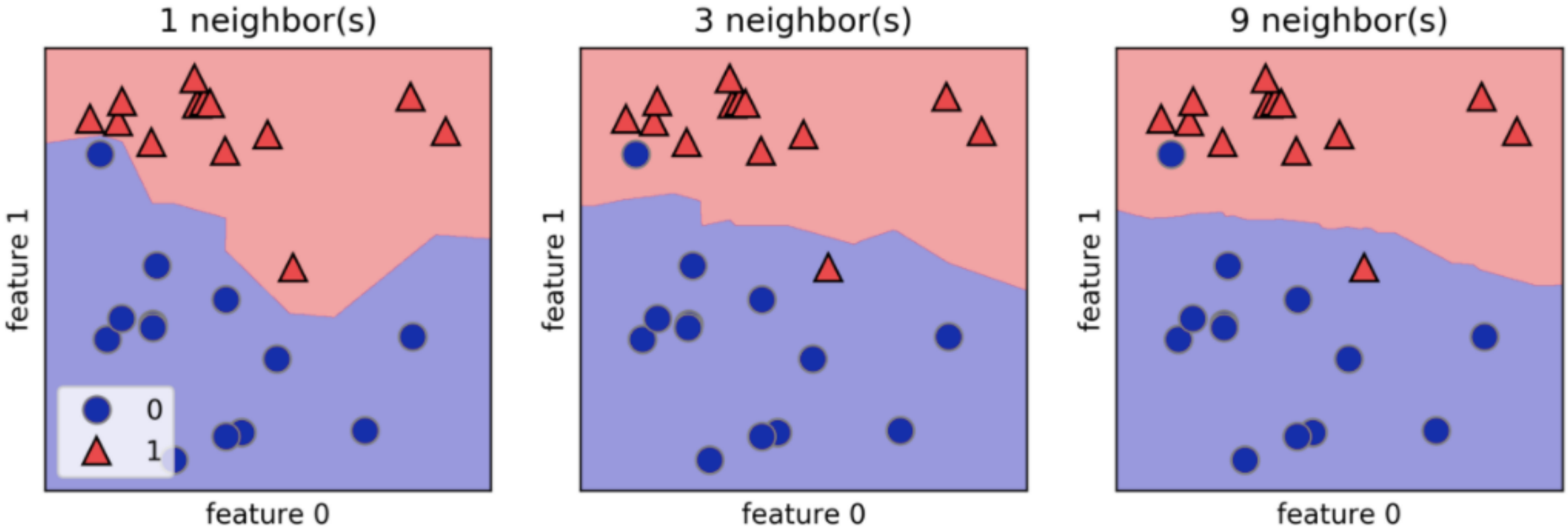
K-Nearest Neighbor Classification: What “K”?



What happens to the decision boundary as “k” grows?

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

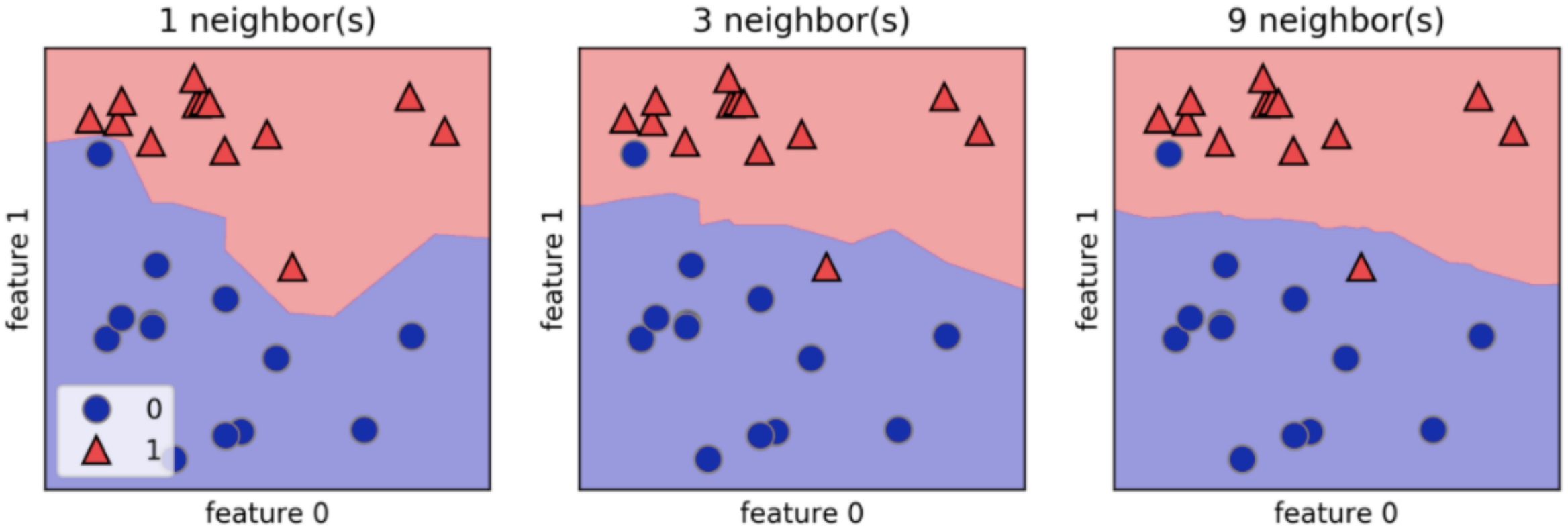
K-Nearest Neighbor Classification: What “K”?



What happens when “k” equals the training data size?

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification: What “K”?

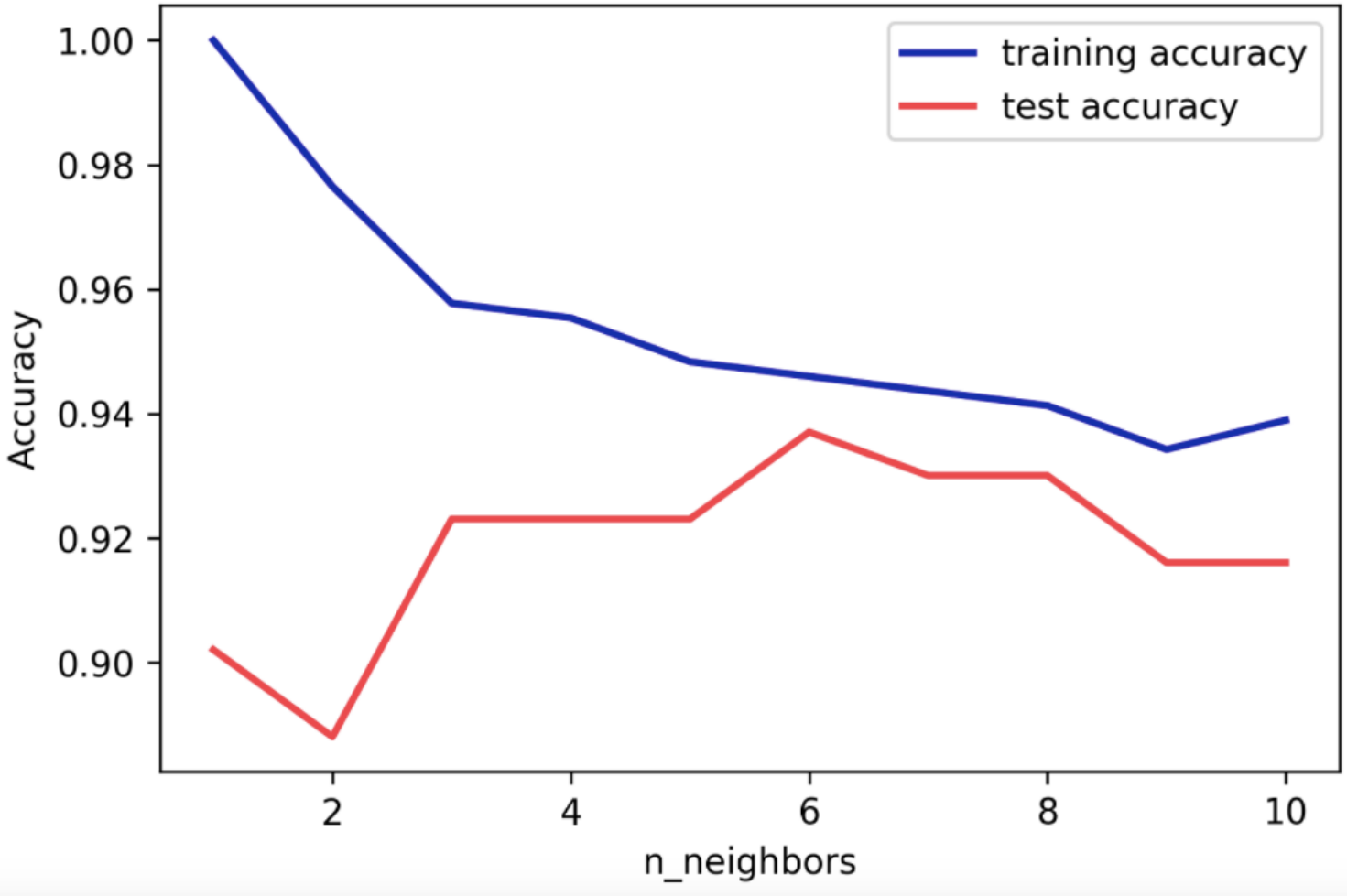


(Highest Model Complexity)

(Lower Model Complexity)

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification: What “K”?

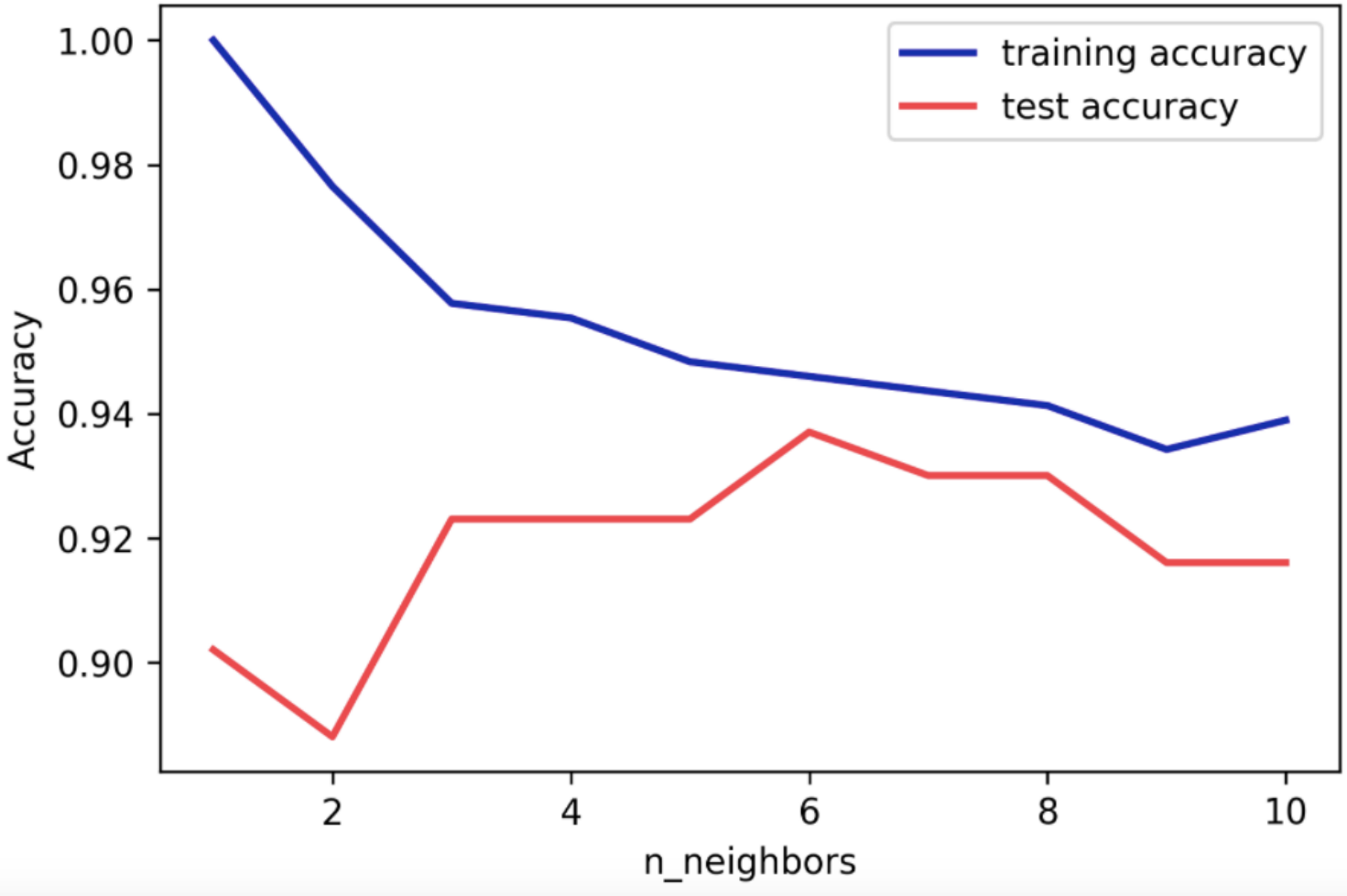


At what value for “k” is model **overfitting** the most?

k = 1

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification: What “K”?

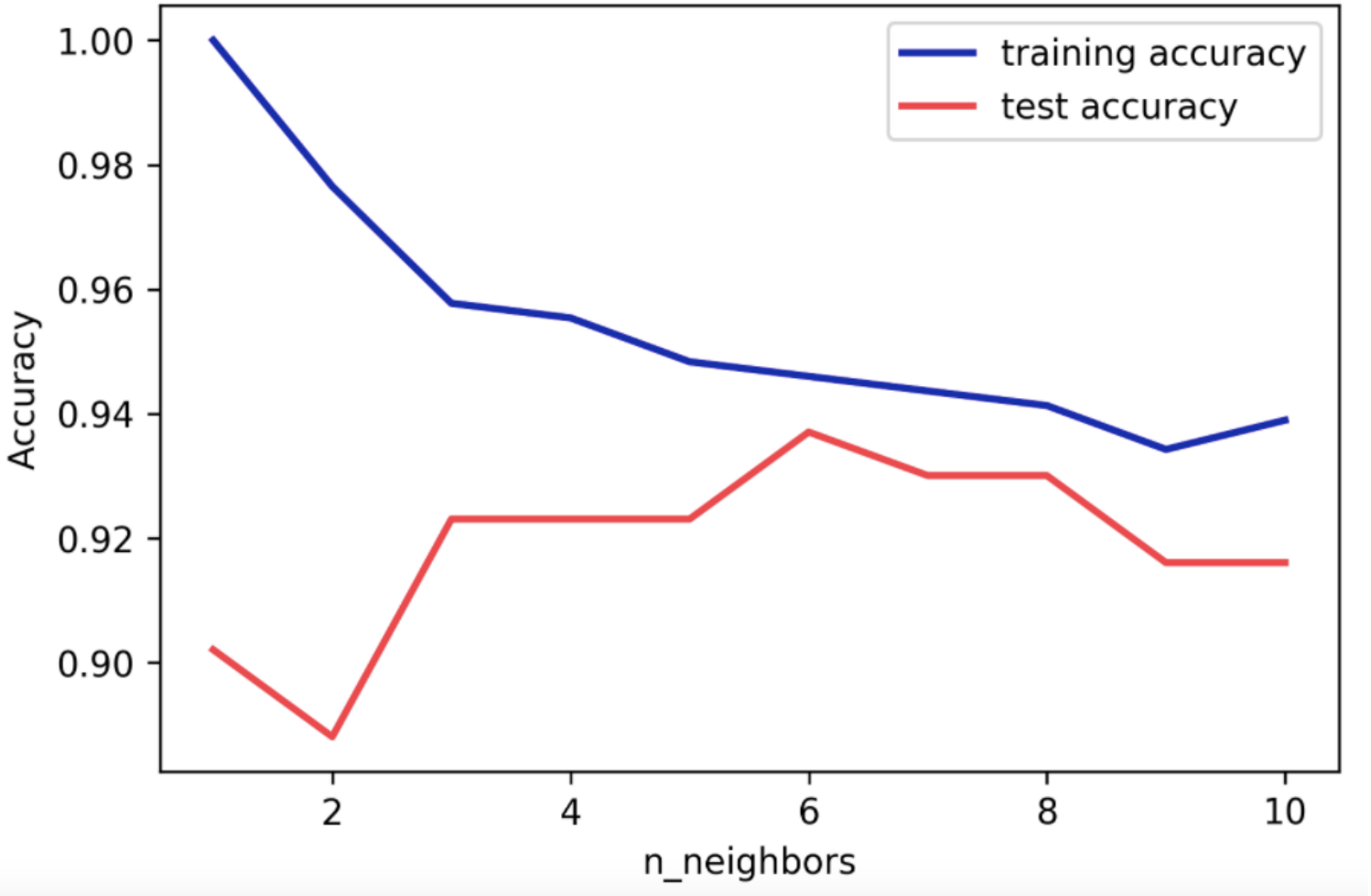


At what value for “k” is model **underfitting** the most?

k = 10

https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor Classification: What “K”?



What is the best value for “k”?

k = 6

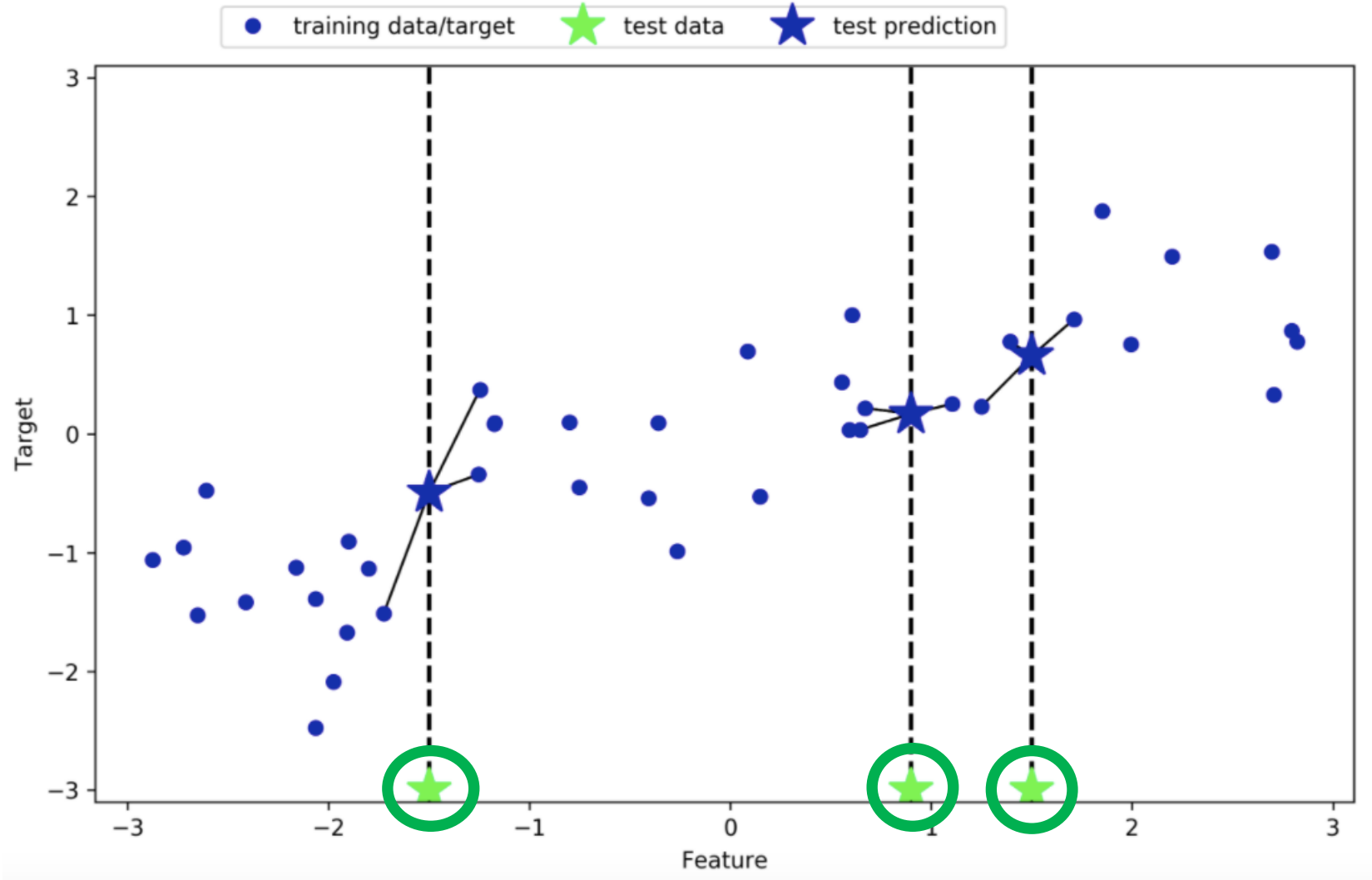
https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

K-Nearest Neighbor: How to Use to Predict More than Two Classes?

- Tally number of examples belonging to each class and again choose the majority vote winners

K-Nearest Neighbor: How to Use for Regression?

e.g.,



https://github.com/amueller/introduction_to_ml_with_python/blob/master/02-supervised-learning.ipynb

What are Strengths of KNN?

- Adapts as new data is added
- Training is relatively fast
- Easy to understand

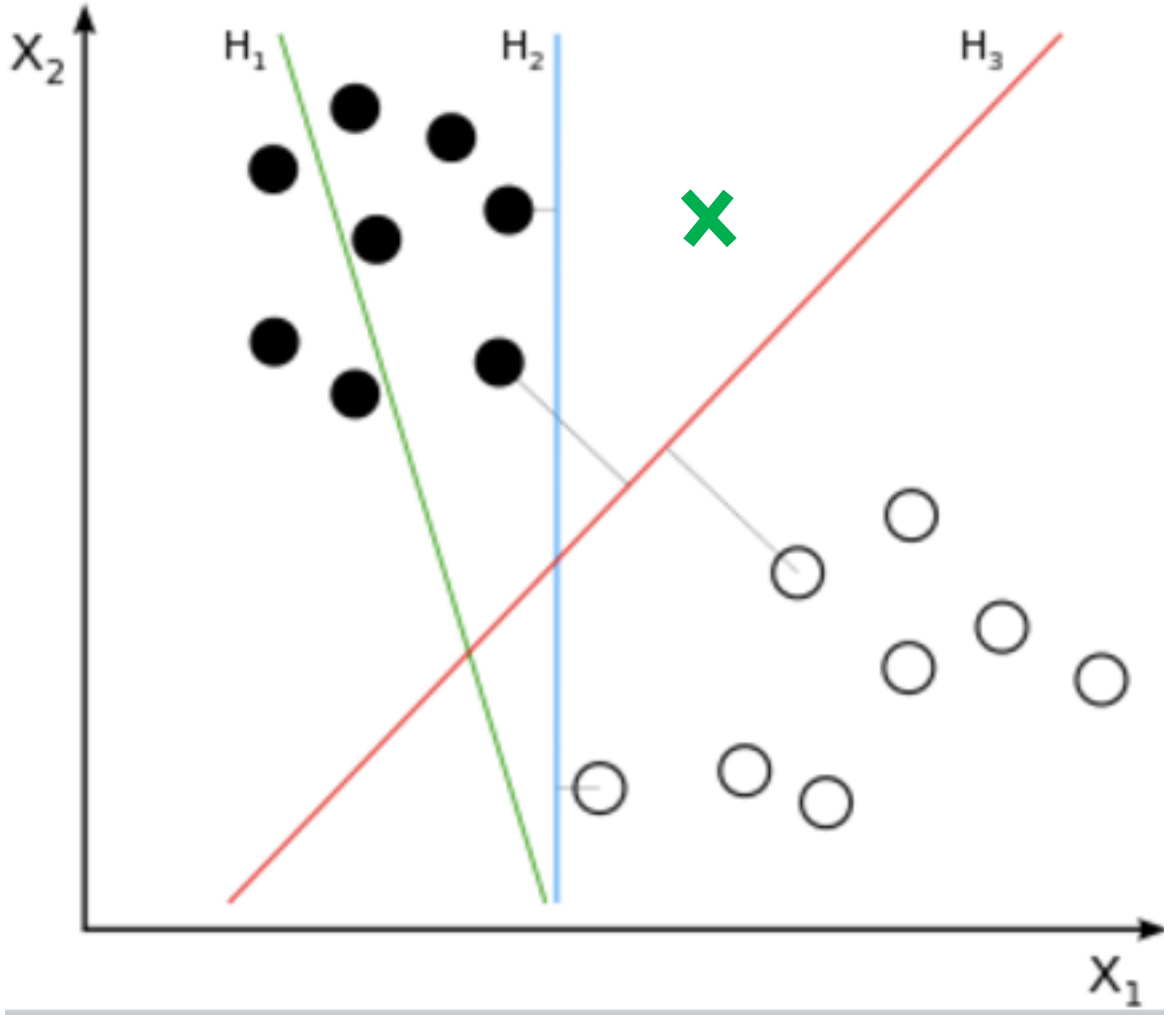
What are Weaknesses of KNN?

- For large datasets, requires large storage space
- For large datasets, this approach can be very slow or infeasible
  - Note: can improve speed with efficient data structures such as KD-trees
- Vulnerable to noisy/irrelevant examples

Today's Topics

- Nearest Neighbor Classifier
- Support Vector Machines
- Evaluating Classifiers Using Cross-Validation
- Tuning Hyper-parameters
- Lab

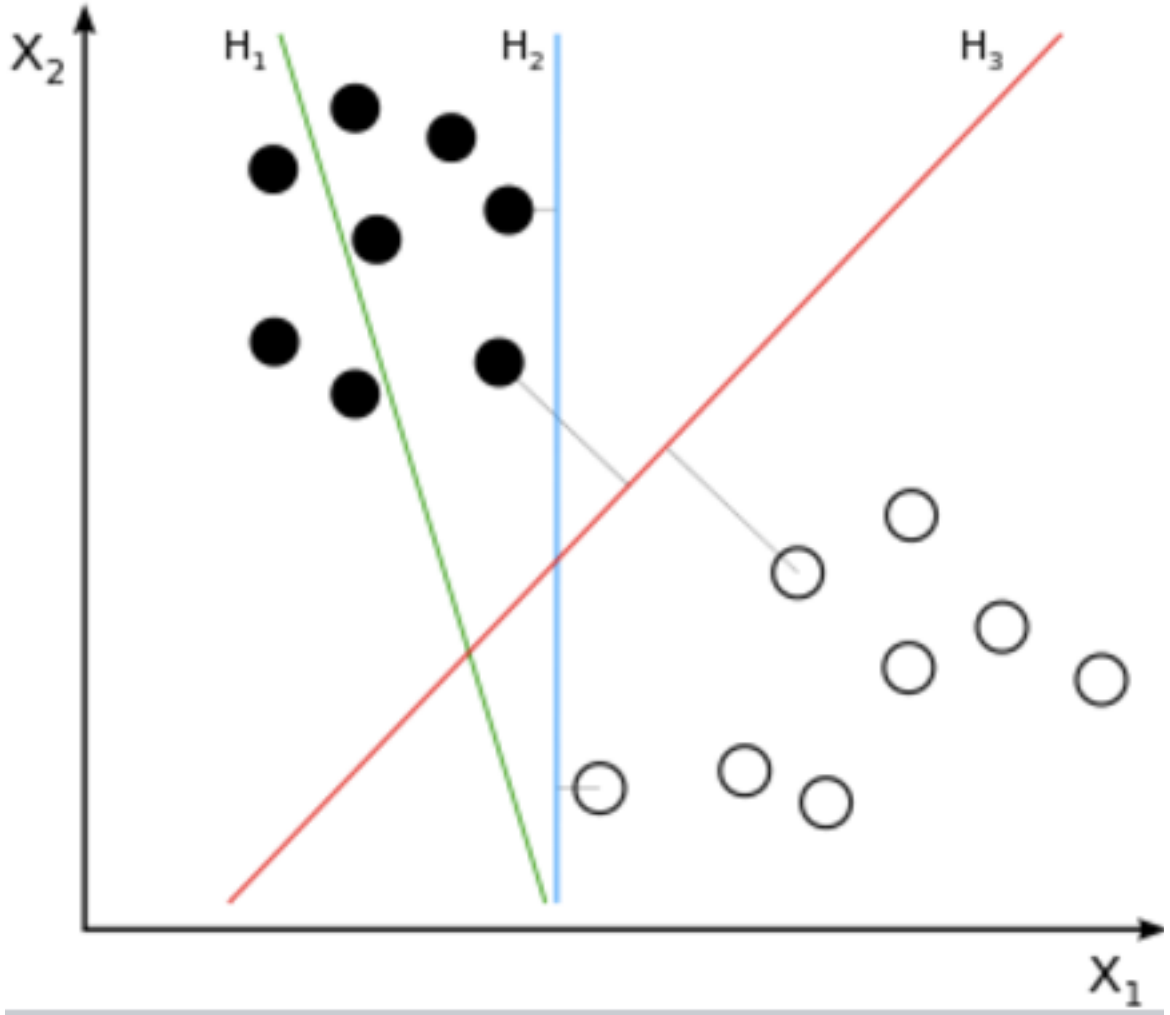
Support Vector Machine (SVM) Motivation



https://en.wikipedia.org/wiki/Linear_separability

To which class would each decision boundary assign the new data point?

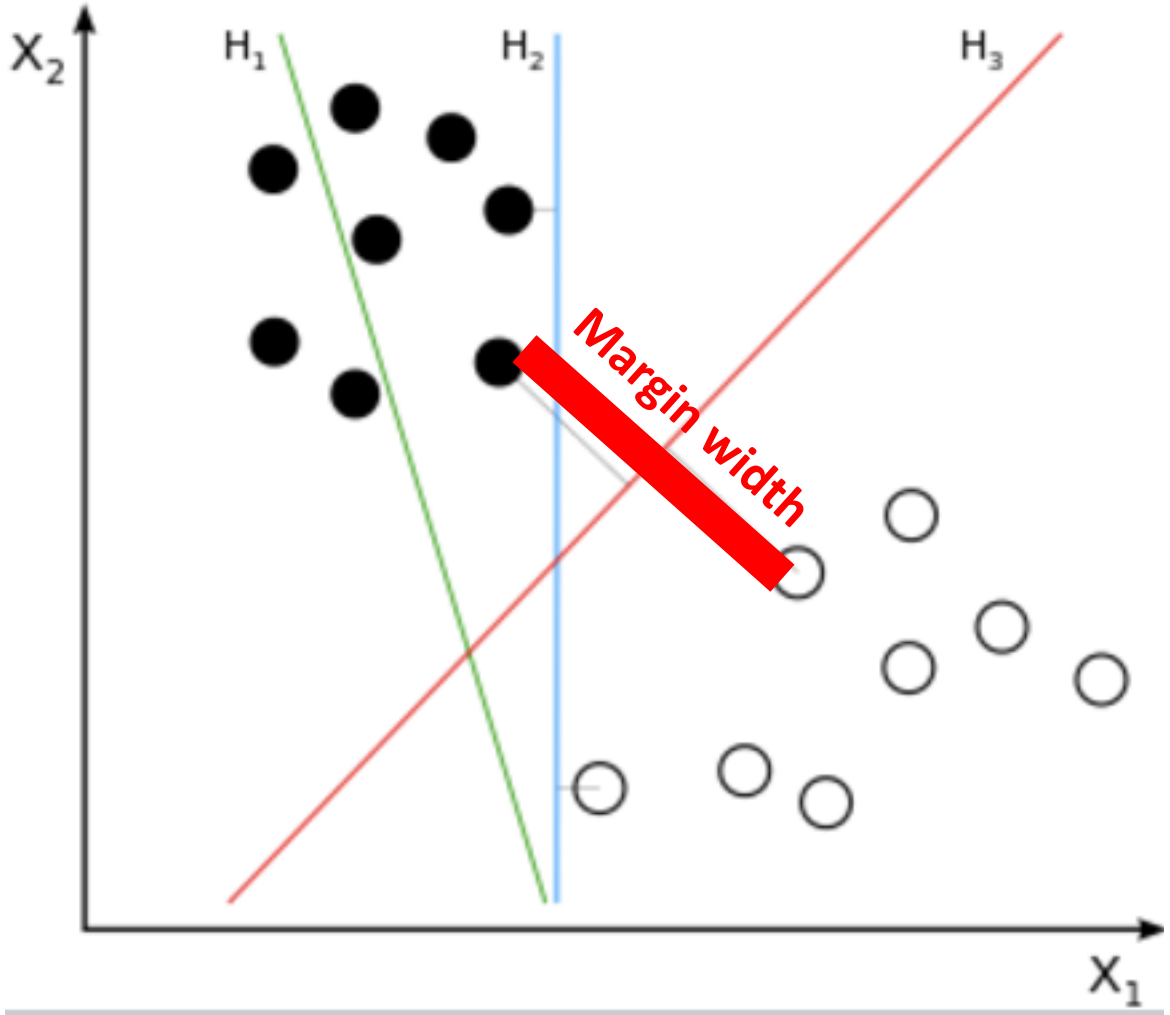
Support Vector Machine (SVM) Motivation



https://en.wikipedia.org/wiki/Linear_separability

Which decision boundary would you choose to separate data?

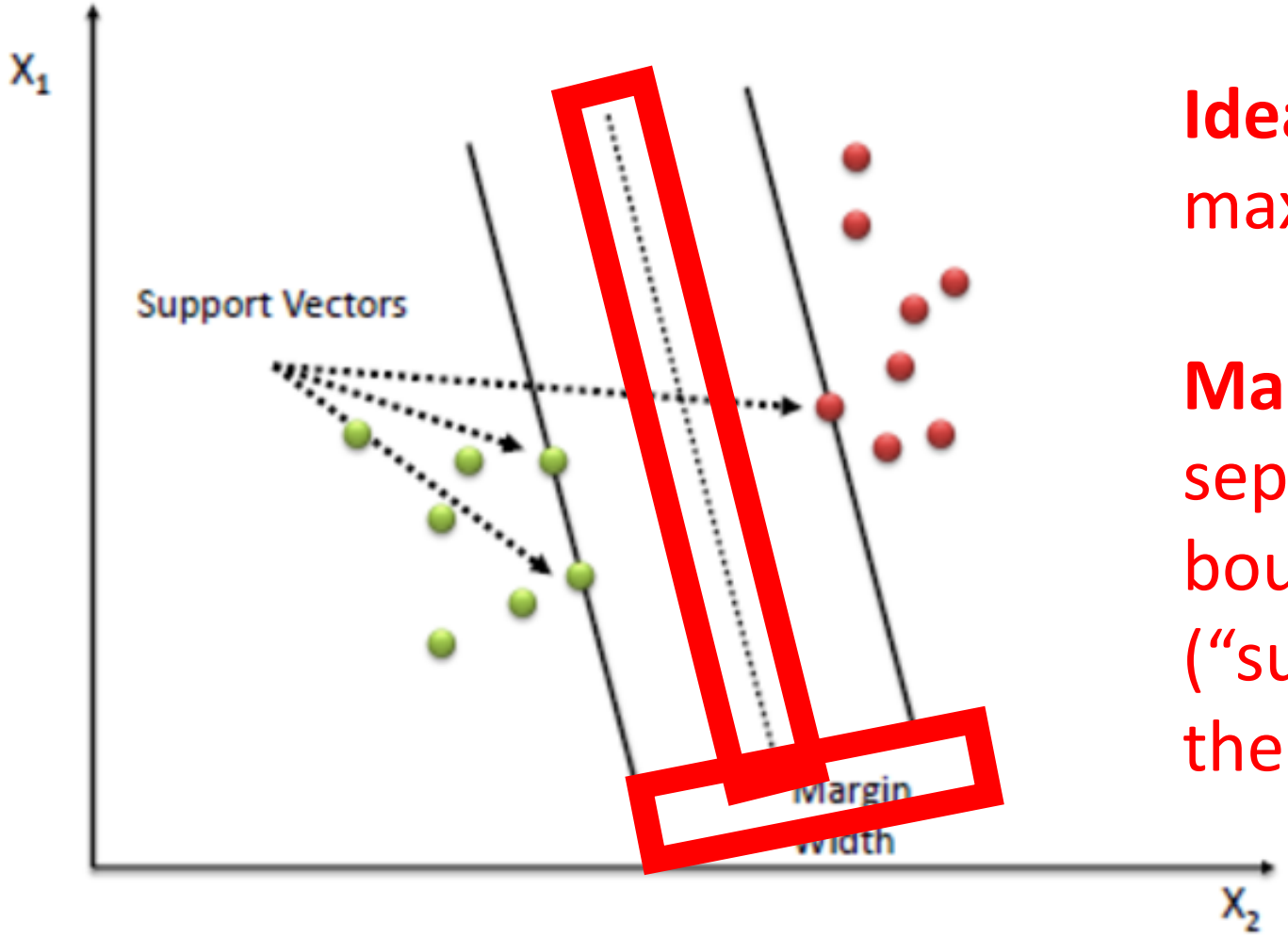
Support Vector Machine (SVM) Motivation



https://en.wikipedia.org/wiki/Linear_separability

Idea: choose hyperplane that maximizes the margin width.

Support Vector Machine (SVM) Motivation

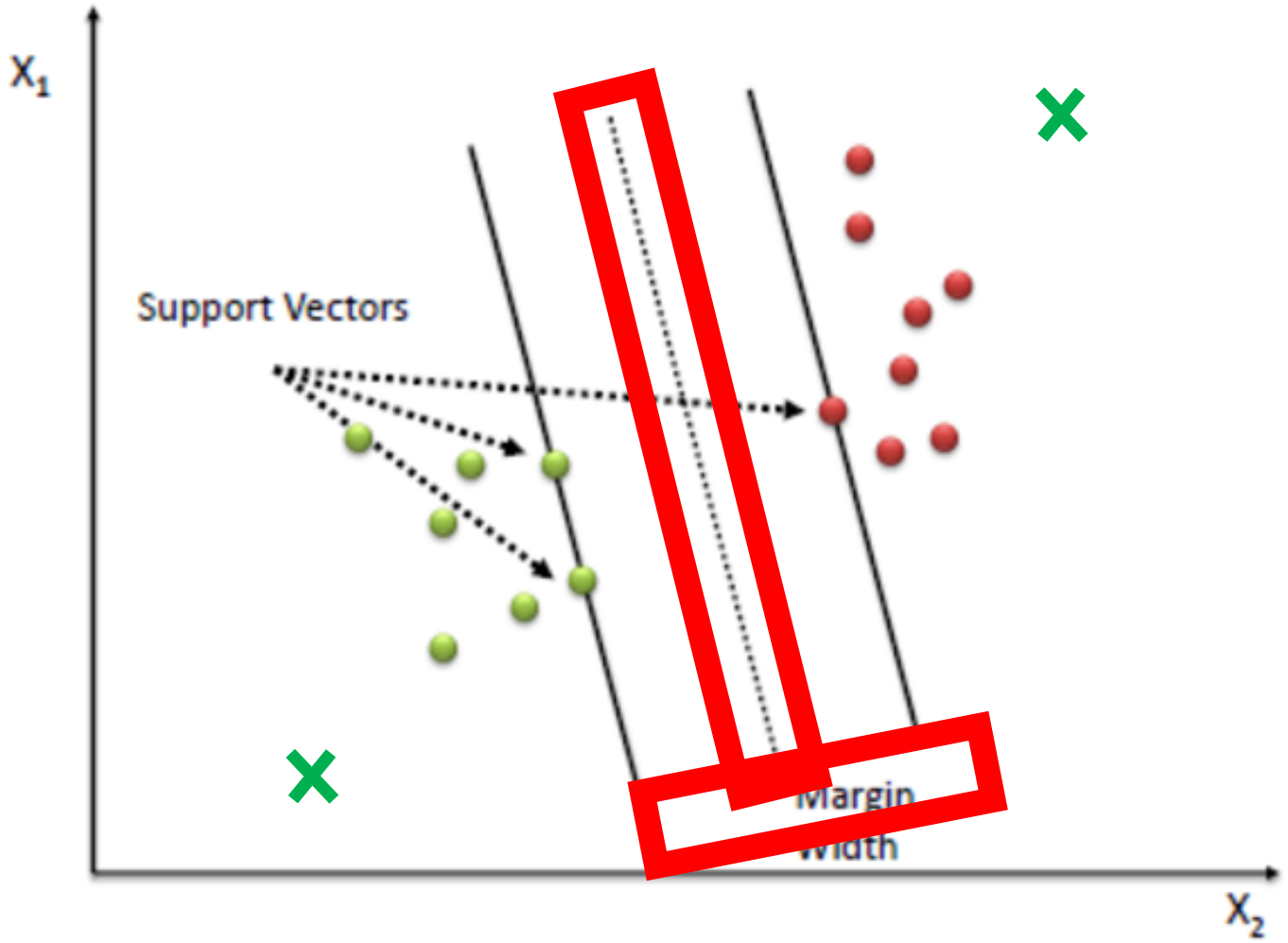


Idea: choose hyperplane that maximizes the "margin" width.

Margin: distance between the separating hyperplane (decision boundary) and training samples ("support vectors") closest to the hyperplane.

http://chem-eng.utoronto.ca/~datamining/dmc/support_vector_machine.htm

Support Vector Machine (SVM) Motivation

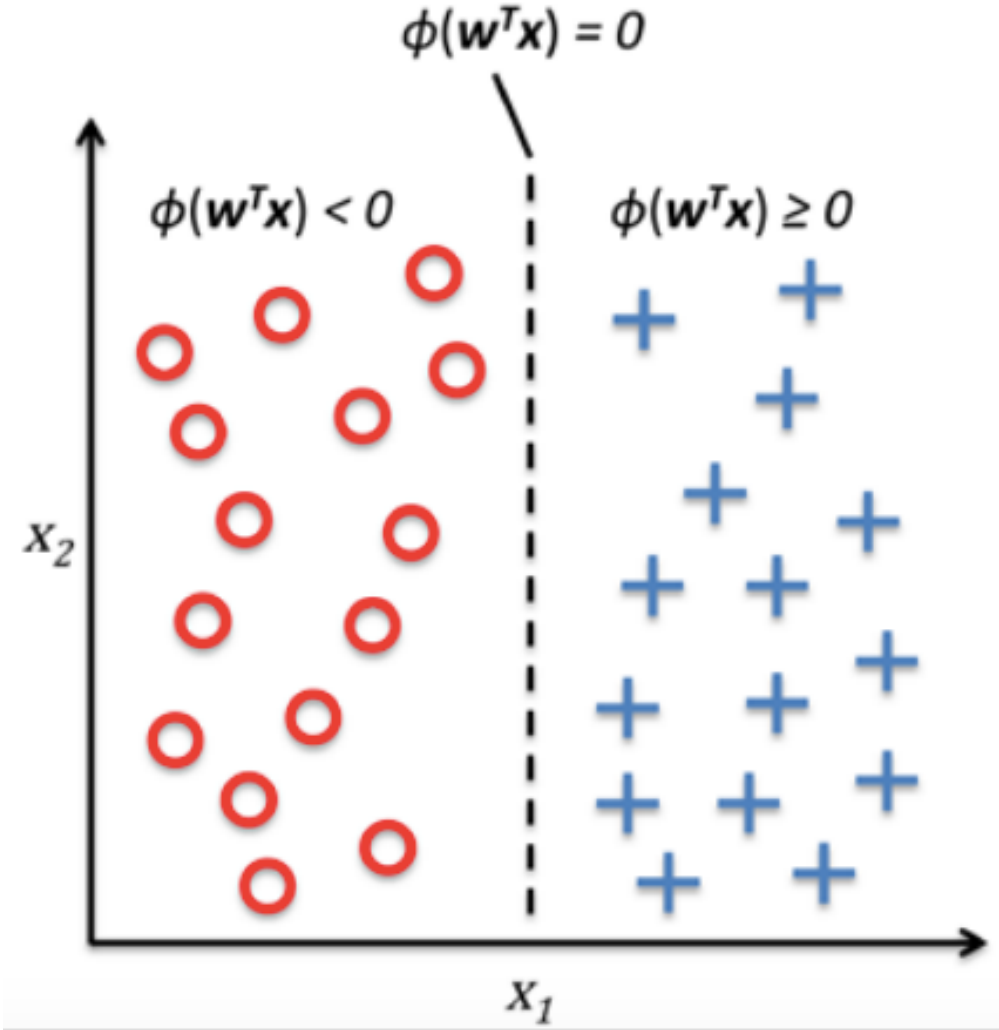


When trying to maximize the margin, what happens to the choice of line when you add outliers to the dataset?

http://chem-eng.utoronto.ca/~datamining/dmc/support_vector_machine.htm

Support Vector Machine (SVM): Linear Algebra Review

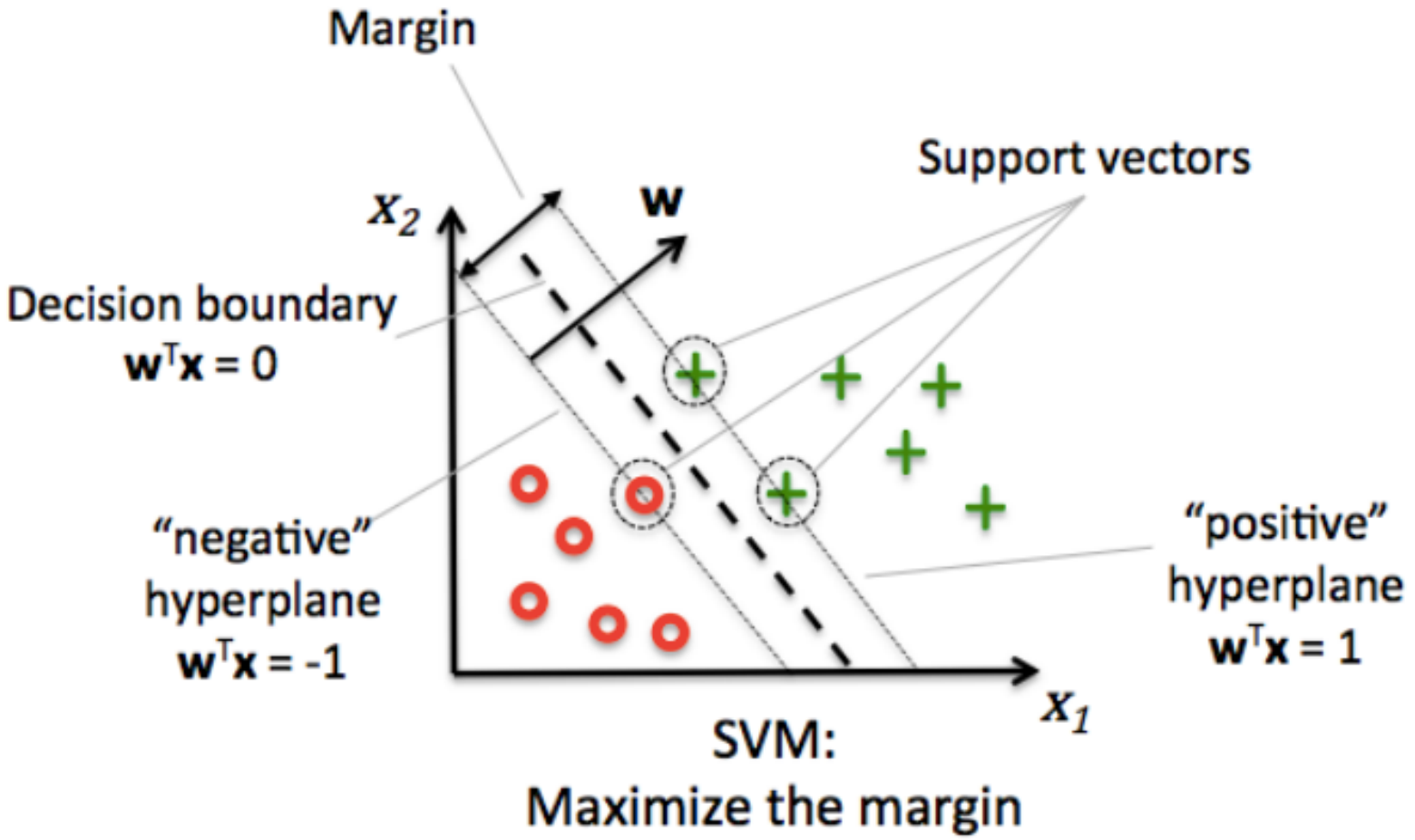
$$w_0 x_0 + w_1 x_1 + \cdots + w_m x_m = \sum_{j=0}^{m} x_j w_j = \boldsymbol{w}^T \boldsymbol{x}$$



(Tutorial: http://www.cs.cmu.edu/~zkolter/course/linalg/linalg_notes.pdf)

Python Machine Learning; Raschkka & Mirjalili

Support Vector Machine (SVM): Formalizing Definition



Python Machine Learning; Raschkka & Mirjalili

Support Vector Machine (SVM): Formalizing Definition

$$(\mathbf{w}^T\mathbf{x} + b)y \geq 1$$

$$y = \begin{cases} 1 & \text{if } \mathbf{w}^T\mathbf{x} + b \geq 1 \\ -1 & \text{if } \mathbf{w}^T\mathbf{x} + b \leq -1 \\ \text{Undefined} & \text{if } -1 \leq \mathbf{w}^T\mathbf{x} + b \leq 1 \end{cases}$$

http://www.cs.utoronto.ca/~fidler/teaching/2015/slides/CSC411/15_svm.pdf

Support Vector Machine (SVM): Training a Classifier

Same as finding parameters
(w, b) that maximizes margin

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2$$

$$s.t. \forall i \quad (\mathbf{w}^T\mathbf{x}^{(i)} + b)t^{(i)} \geq 1,$$

Constraint that enforces that all examples
fall outside of the margin space.

Support Vector Machine (SVM): Training a Classifier

Same as finding parameters (w, b) that maximizes margin

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2$$

$$s.t. \forall i \quad (\mathbf{w}^T \mathbf{x}^{(i)} + b)t^{(i)} \geq 1,$$

Derivation:

- Positive and negative hyperplanes are:

$$w_0 + \boldsymbol{w}^T \boldsymbol{x}_{pos} = 1 \qquad w_0 + \boldsymbol{w}^T \boldsymbol{x}_{neg} = -1$$

- Subtracting two equations above from each other gives:

$$\boldsymbol{w}^T \left(\boldsymbol{x}_{pos} - \boldsymbol{x}_{neg} \right) = 2$$

- Normalizing this equation by the length of **w** gives the margin distance:

$$\frac{\boldsymbol{w}^T \left(\boldsymbol{x}_{pos} - \boldsymbol{x}_{neg} \right)}{\|\boldsymbol{w}\|} = \frac{2}{\|\boldsymbol{w}\|}$$

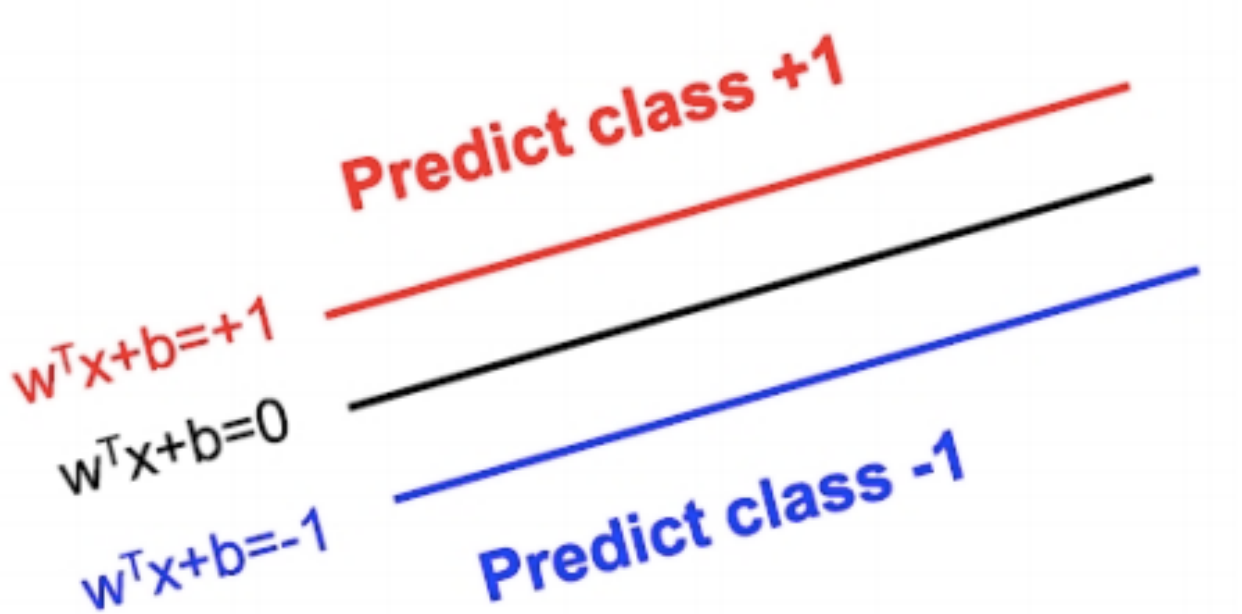
- Find maximum margin by finding the minimum of the reciprocal of the above term

Support Vector Machine (SVM): Training a Classifier

Same as finding parameters
(w, b) that maximizes margin

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2$$

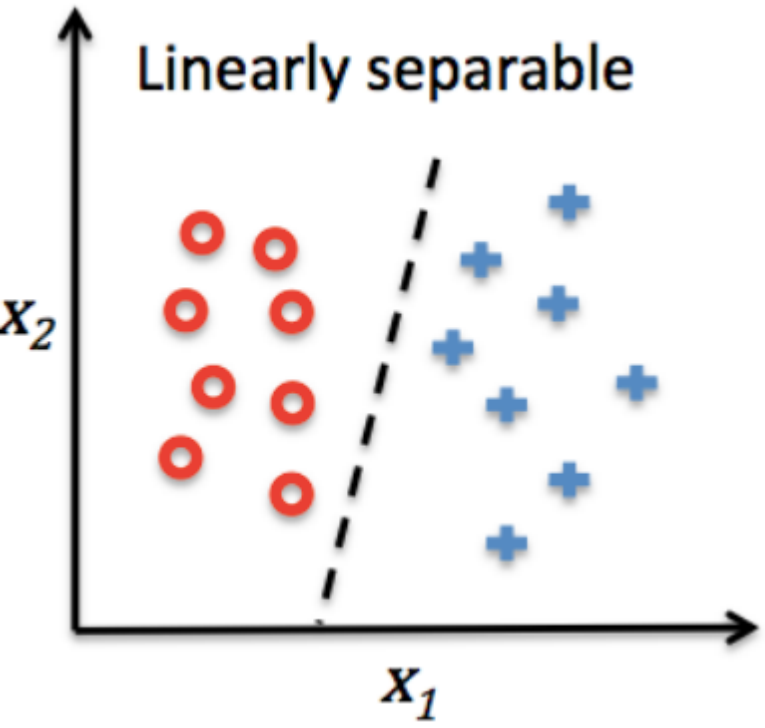
$$s.t. \forall i \quad (\mathbf{w}^T\mathbf{x}^{(i)} + b)t^{(i)} \geq 1,$$

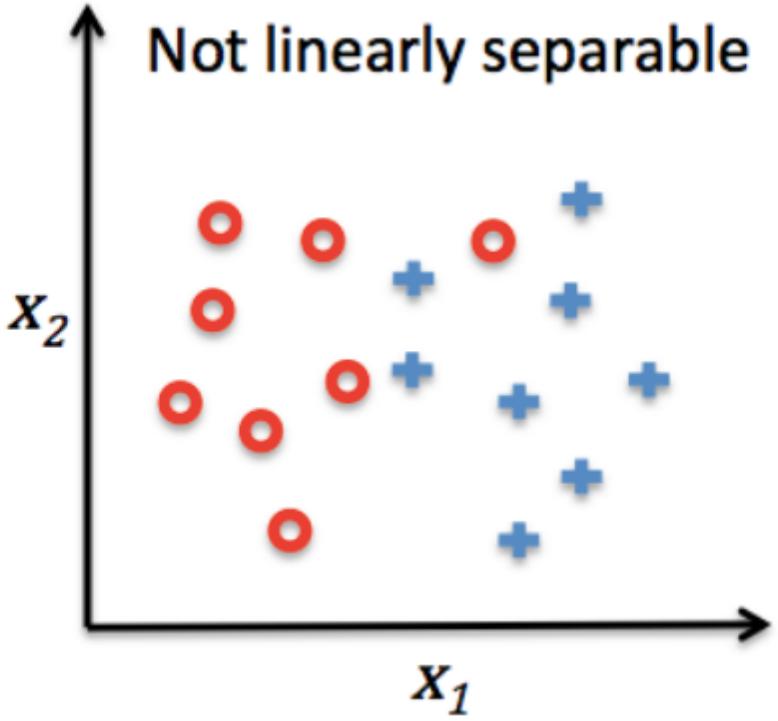


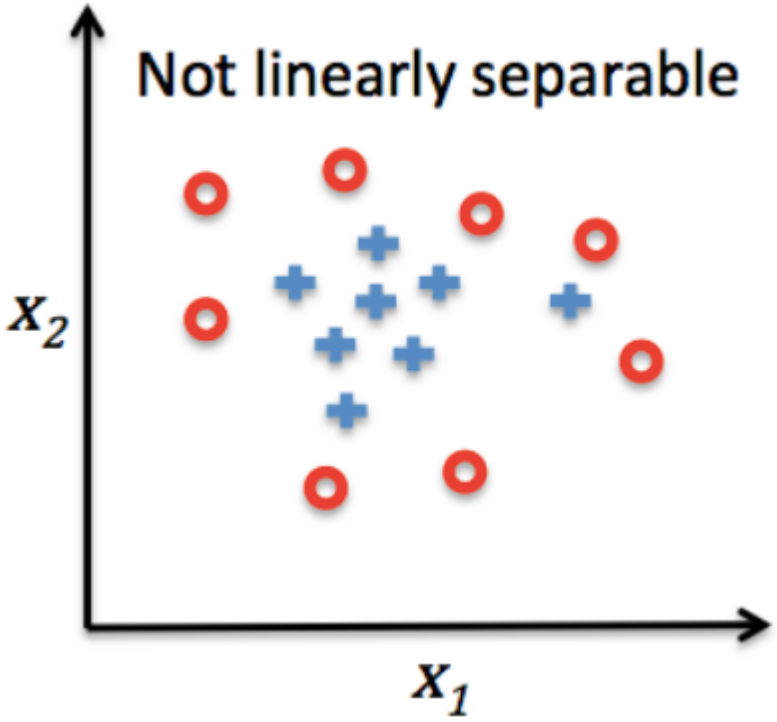
Constraint that enforces that all examples
fall outside of the margin space.

Can be solved with a quadratic programming solver… learn more about this at:
- "The Nature of Statistical Learning and Theory, by Vladimir Vapnik
- A Tutorial on Support Vector Machines for Pattern Recognition by Chris J. C. Burges'

What if the Decision Boundary is Not Linear?

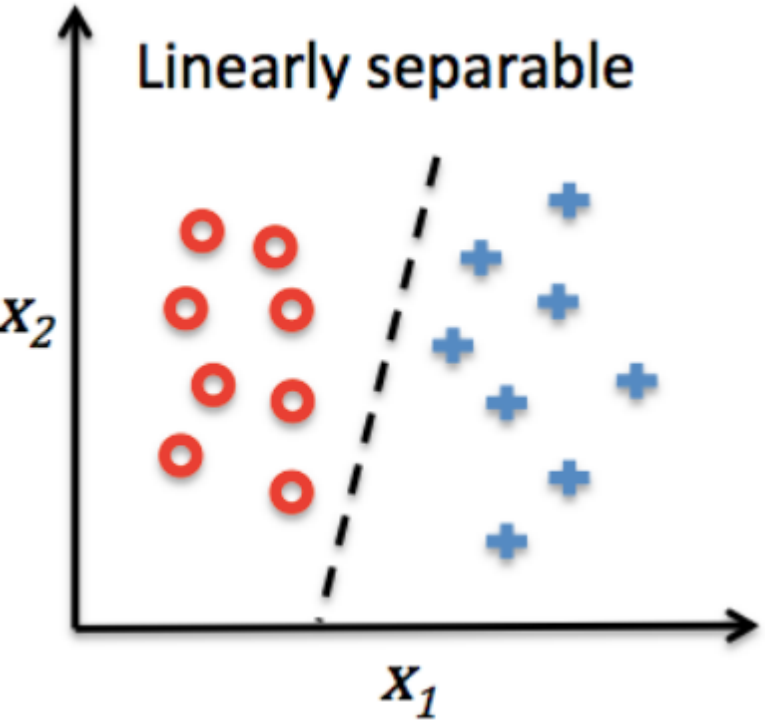




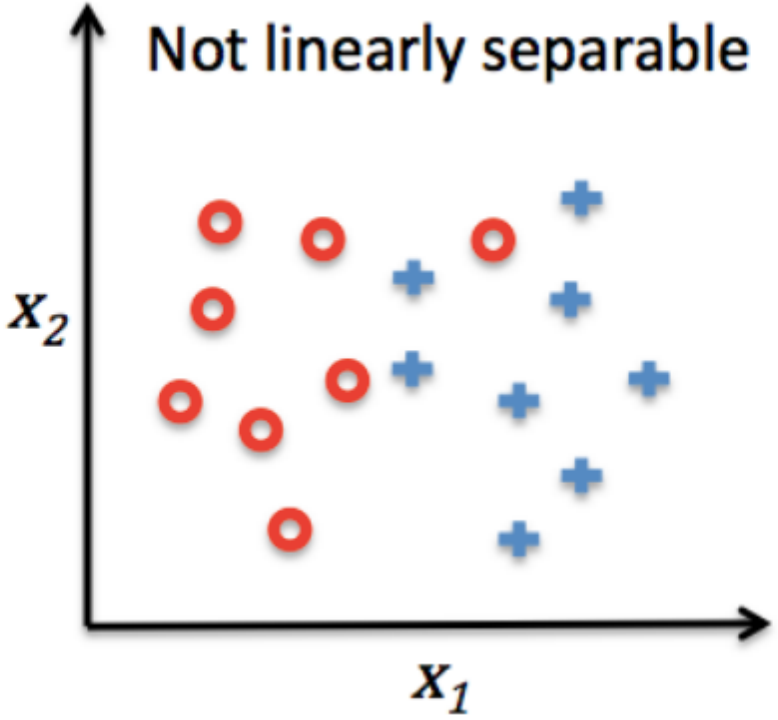


Python Machine Learning; Raschkka & Mirjalili

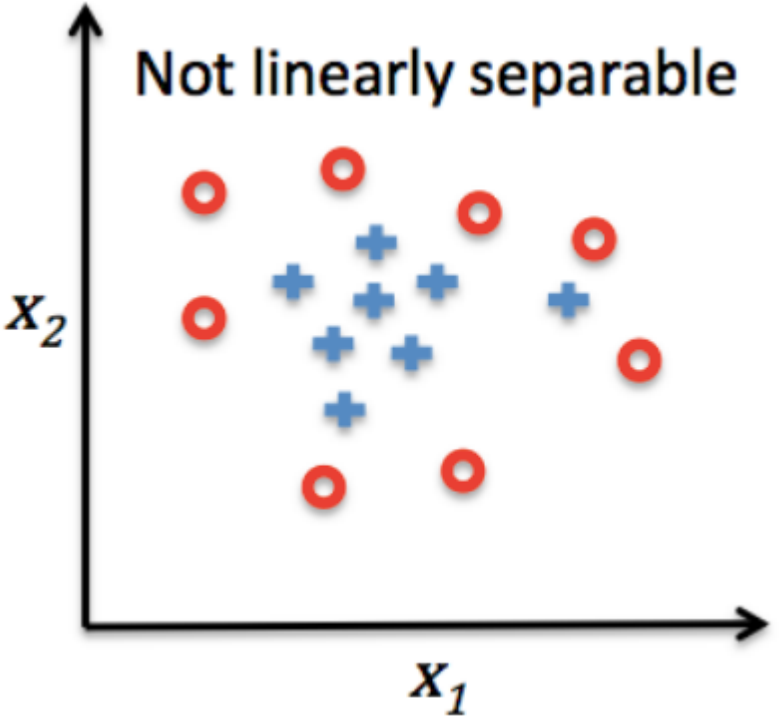
What if the Decision Boundary is Not Linear?



Hard-Margin Classification



Soft-Margin Classification



Python Machine Learning; Raschkka & Mirjalili

Soft-Margin Classification

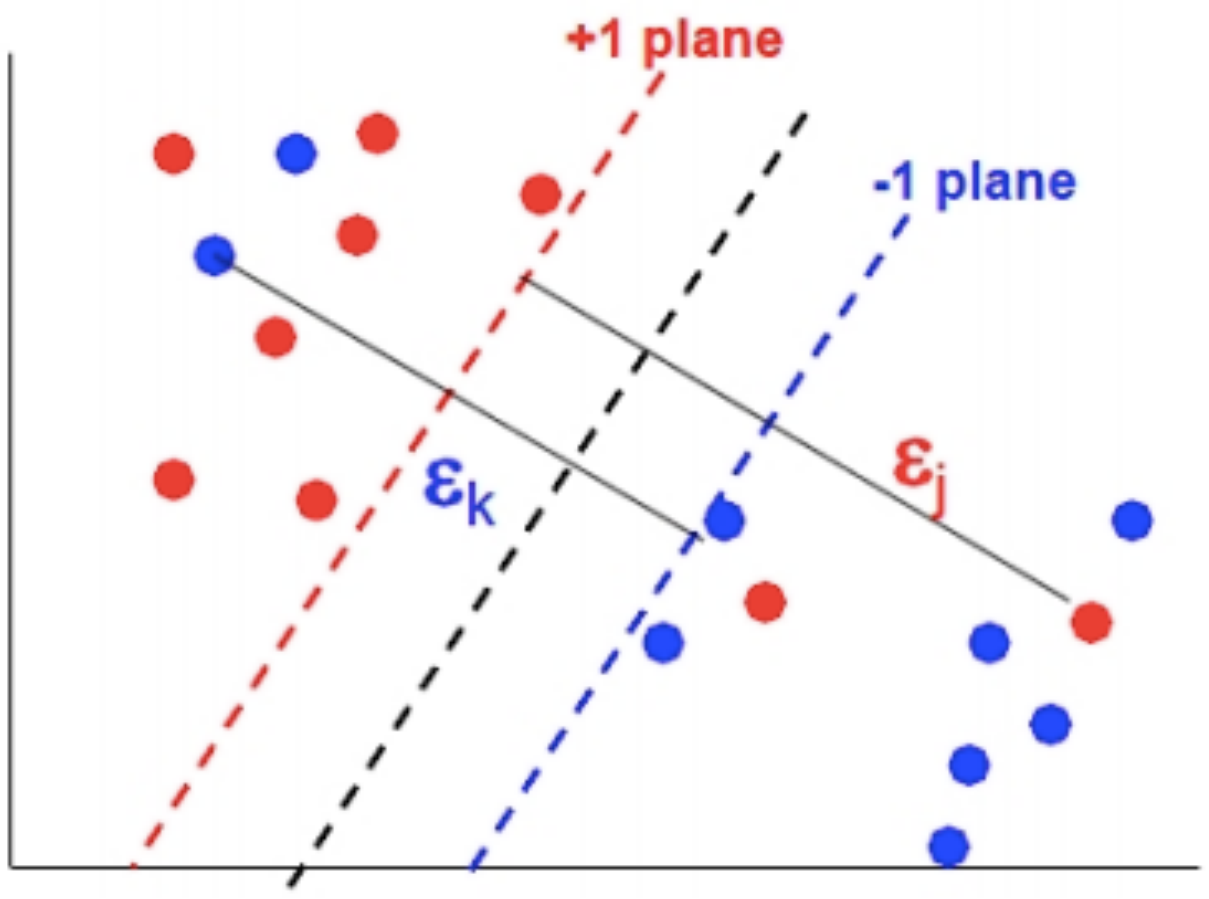
$$\min \frac{1}{2}||\mathbf{w}||^2 + \lambda \sum_{i=1}^{N} \xi_i$$

$$\text{s.t} \quad \xi_i \geq 0; \quad \forall i \quad t^{(i)}(\mathbf{w}^T \mathbf{x}^{(i)}) \geq 1 - \xi_i = 0$$

Introduce "slack" variable:

$$\boldsymbol{w}^T \boldsymbol{x}^{(i)} \geq 1 - \xi^{(i)} \text{ if } y^{(i)} = 1$$

$$\boldsymbol{w}^T \boldsymbol{x}^{(i)} \leq -1 + \xi^{(i)} \text{ if } y^{(i)} = -1$$

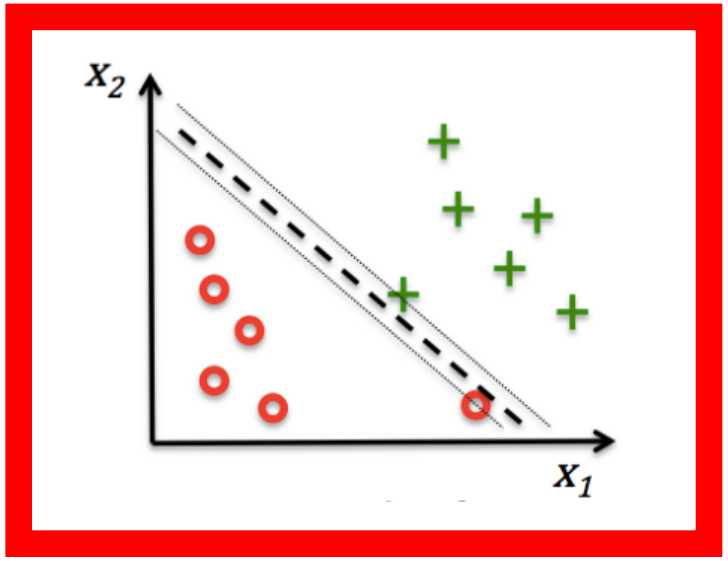


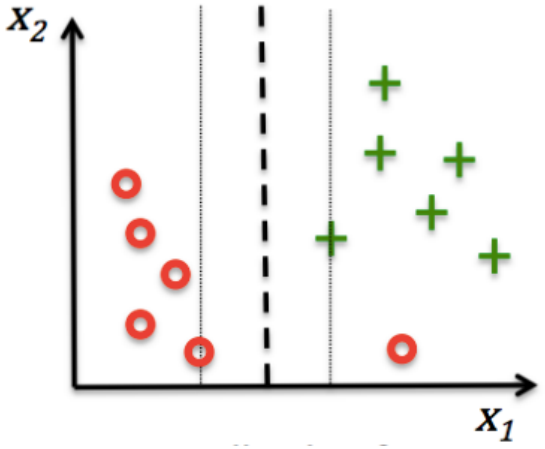
Python Machine Learning; Raschkka & Mirjalili

Soft-Margin Classification

$$\min \frac{1}{2}||\mathbf{w}||^2 + \lambda \sum_{i=1}^{N} \xi_i$$

$$\text{s.t} \quad \xi_i \geq 0; \quad \forall i \quad t^{(i)}(\mathbf{w}^T \mathbf{x}^{(i)}) \geq 1 - \xi_i = 0$$





(Increases priority placed on minimizing error so margin is smaller)

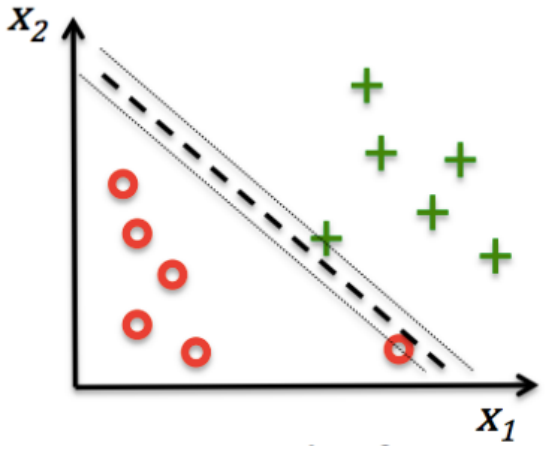
Which plot shows when the slack variable is **larger**?

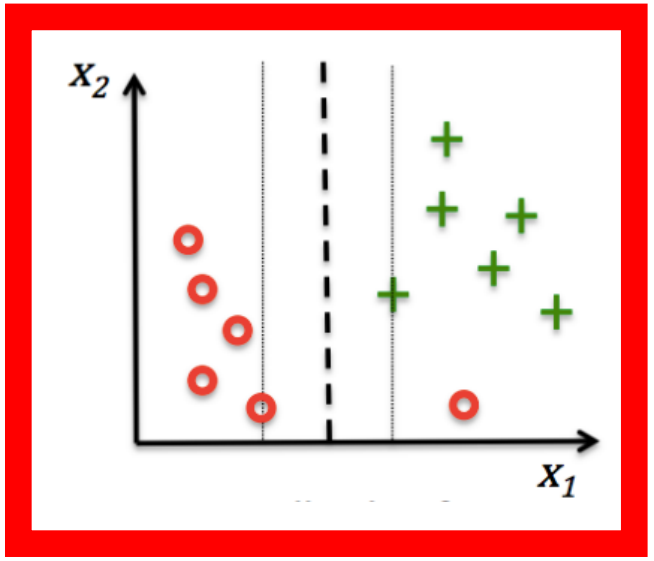
Python Machine Learning; Raschkka & Mirjalili

Soft-Margin Classification

$$\min \frac{1}{2}||\mathbf{w}||^2 + \lambda \sum_{i=1}^{N} \xi_i$$

$$\text{s.t} \quad \xi_i \geq 0; \quad \forall i \quad t^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)}) \geq 1 - \xi_i = 0$$



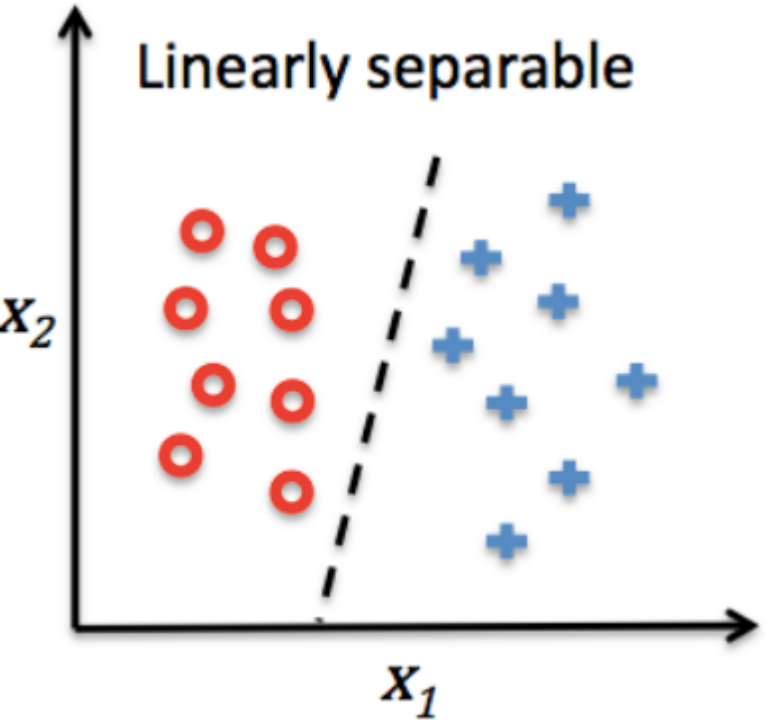


(Increases priority placed on maximizing margin so error is larger)

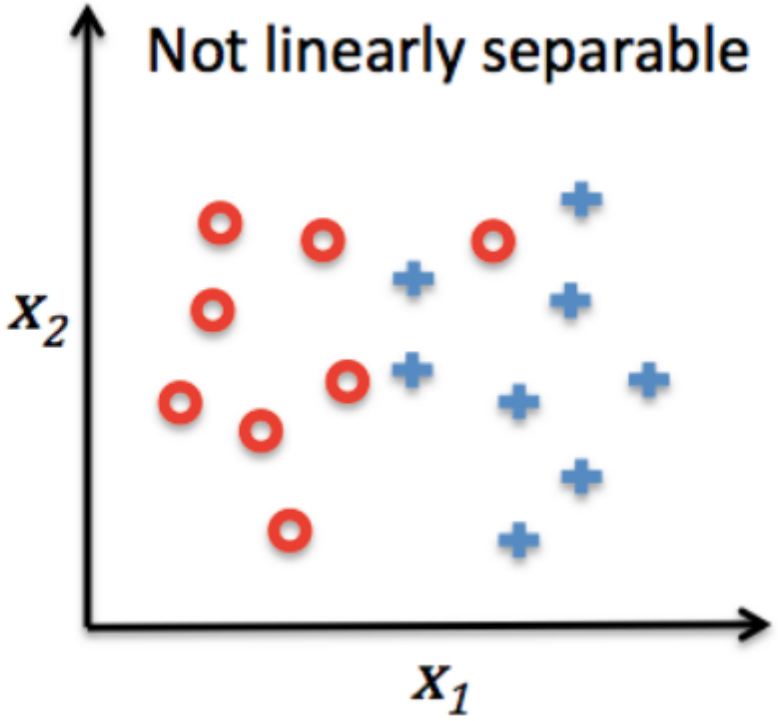
Which plot shows when the slack variable is **smaller**?

Python Machine Learning; Raschkka & Mirjalili

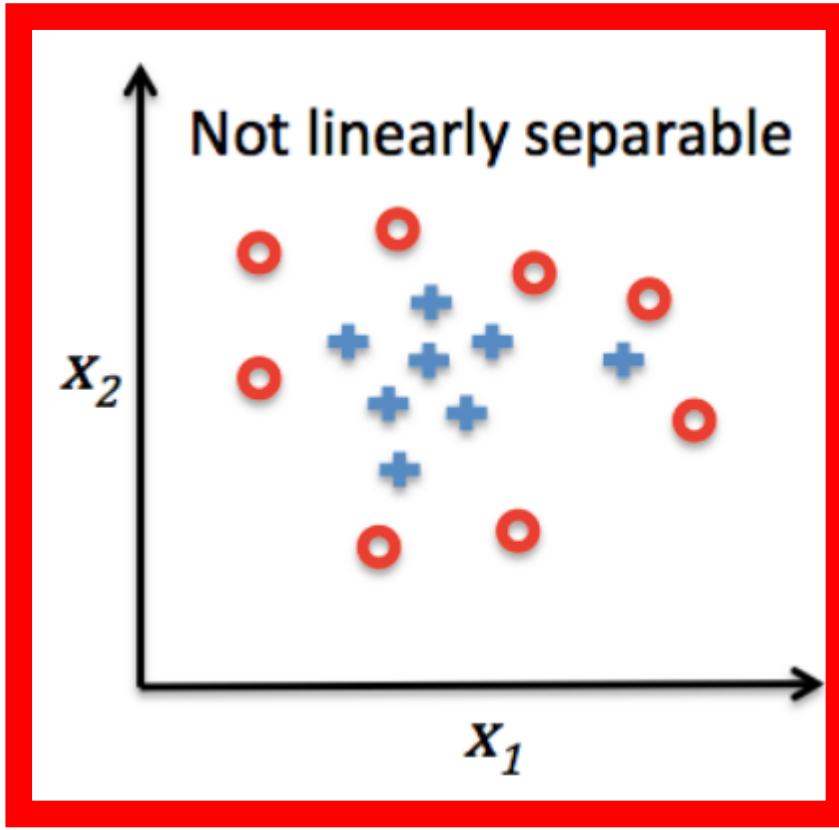
What if the **Decision Boundary** is Not Linear?



Hard-Margin Classification



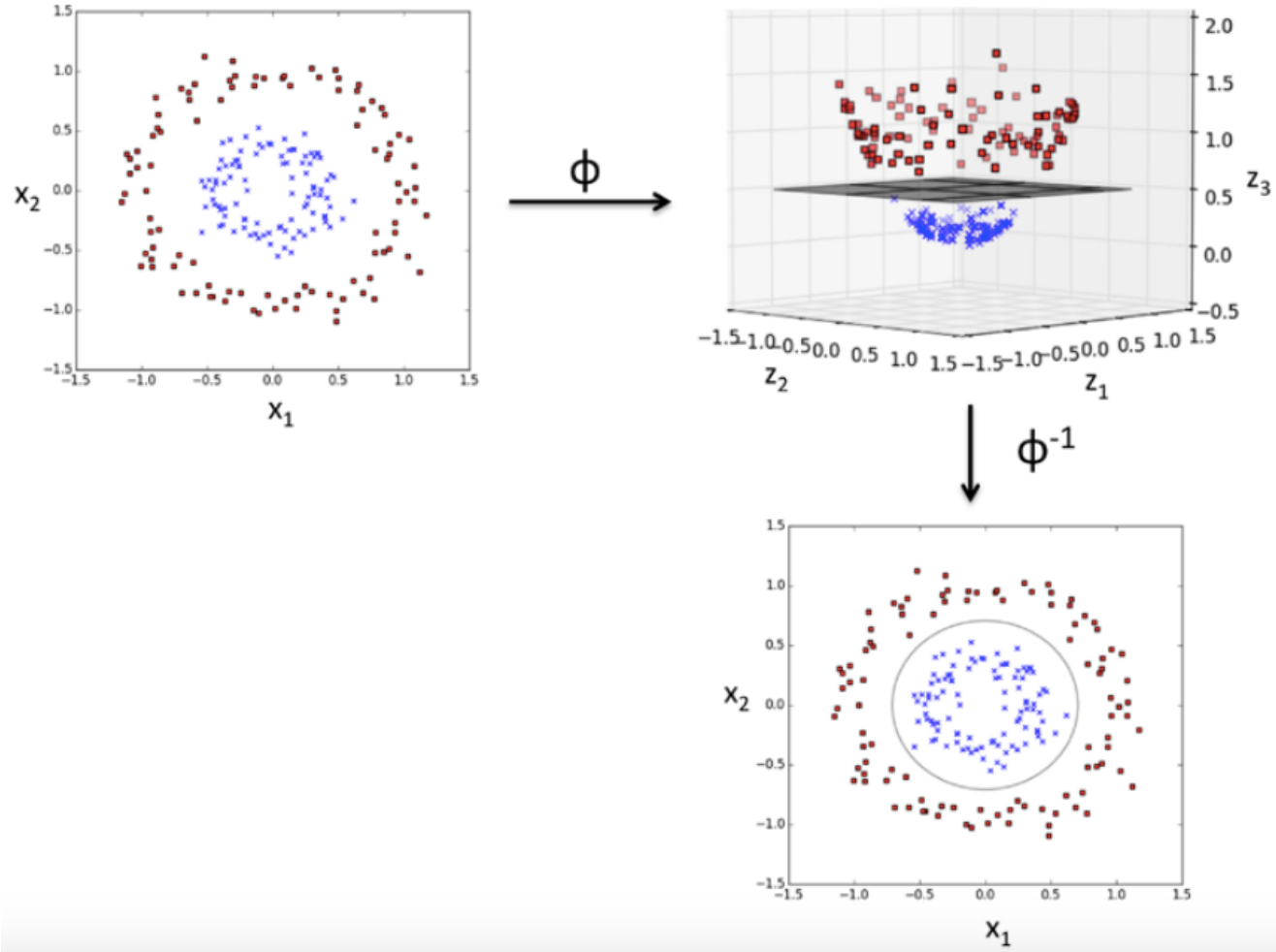
Soft-Margin Classification



Python Machine Learning; Raschkka & Mirjalili

Kernelized Support Vector Machines

- Recall polynomial regression?
  - Project features to higher order space



Python Machine Learning; Raschkka & Mirjalili

Kernelized Support Vector Machines

- Kernels project features to higher order spaces
- Key question: what functions to use when converting features? e.g.,
  - Polynomial kernel
  - Gaussian Radial Basis Function kernel

SVM for Regression

- Reverse the objective:
  - fit as many instances as possible on the street while limiting margin violations (instances of the street)

What are SVM's Strengths

- Insensitive to outliers (only relies on support vectors to choose dividing line)
- Require little memory (rely on a few support vectors)
- Once trained, prediction is fast
- Work well with high-dimensional data

What are SVM's Weaknesses

- Prohibitive computational costs for large datasets
- Performance heavily dependent on soft margin value
- Does not have a direct probabilistic interpretation

Today's Topics

- Nearest Neighbor Classifier
- Support Vector Machines
- Evaluating Classifiers Using Cross-Validation
- Tuning Hyper-parameters
- Lab

Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

Classifier **predicts well** when test data **matches** training data. Good luck?

Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

Classifier **predicts poorly** when test data **does not match** training data. Bad luck?

Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

How to know if good/bad evaluation scores happen from good/bad luck?

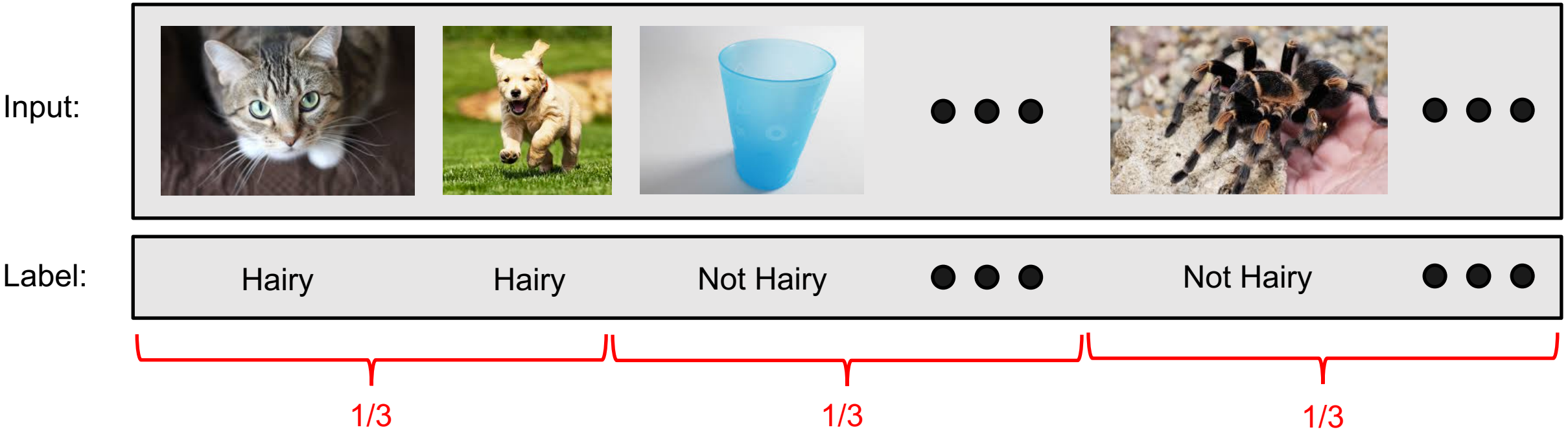
Evaluation of Classification Model

Cross-validation:

limit influence of chosen dataset split

Evaluation of Classification Model

e.g., 3-fold cross-validation



Cross-validation

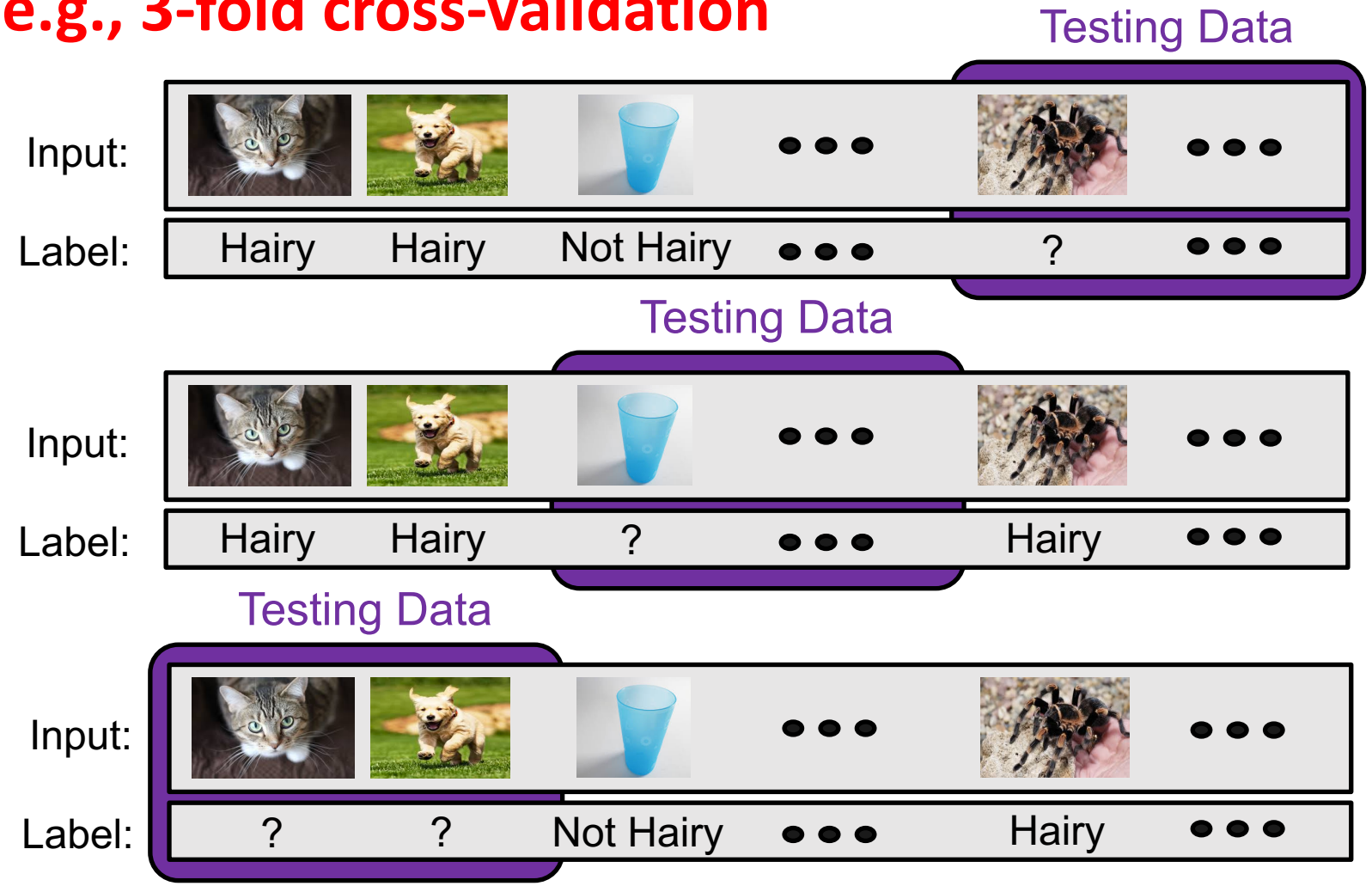
Evaluation of Classification Model

e.g., 3-fold cross-validation

Fold 1:
- train on k-1 partitions
- test on k partitions

Fold 2:
- train on k-1 partitions
- test on k partitions

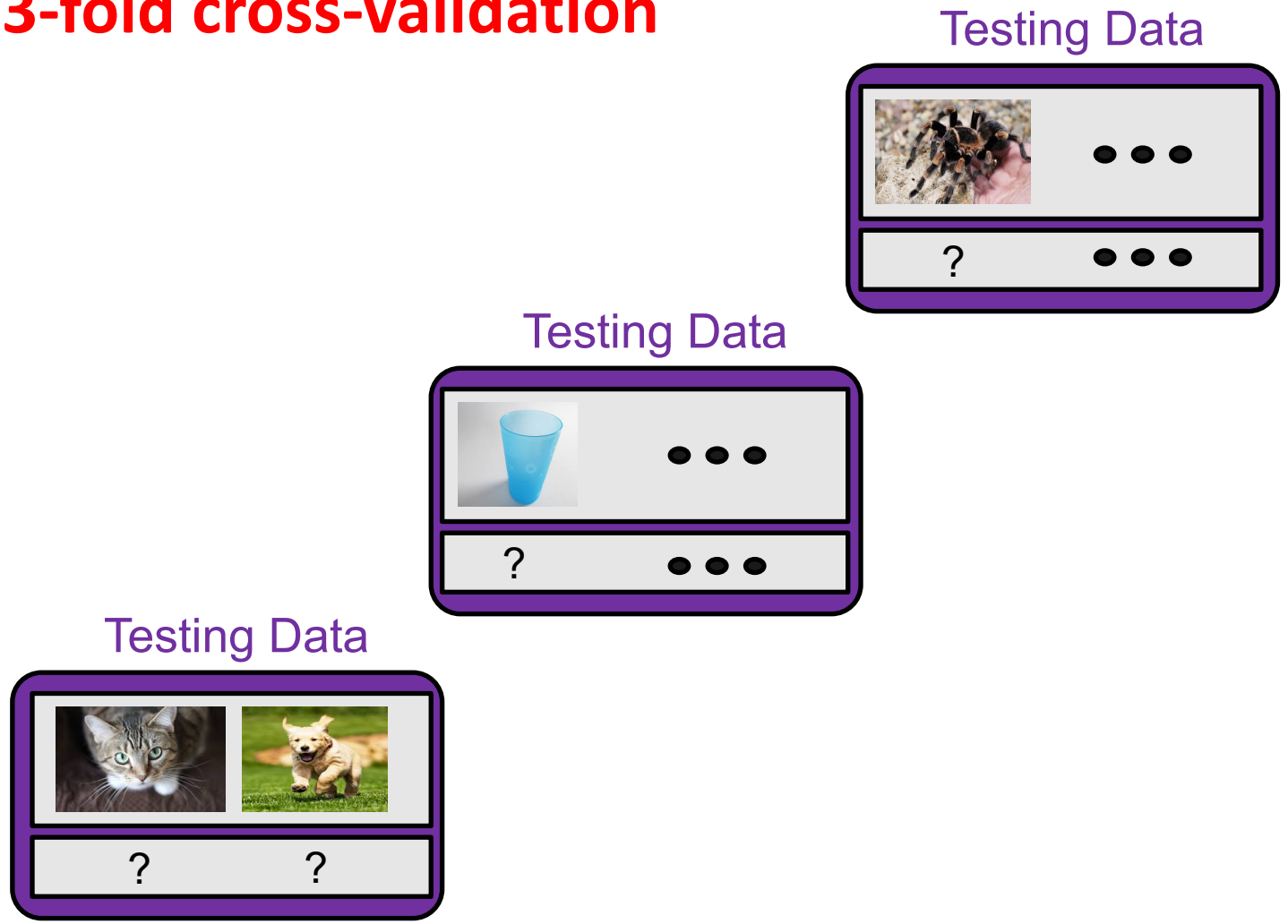
Fold 3:
- train on k-1 partitions
- test on k partitions



Evaluation of Classification Model

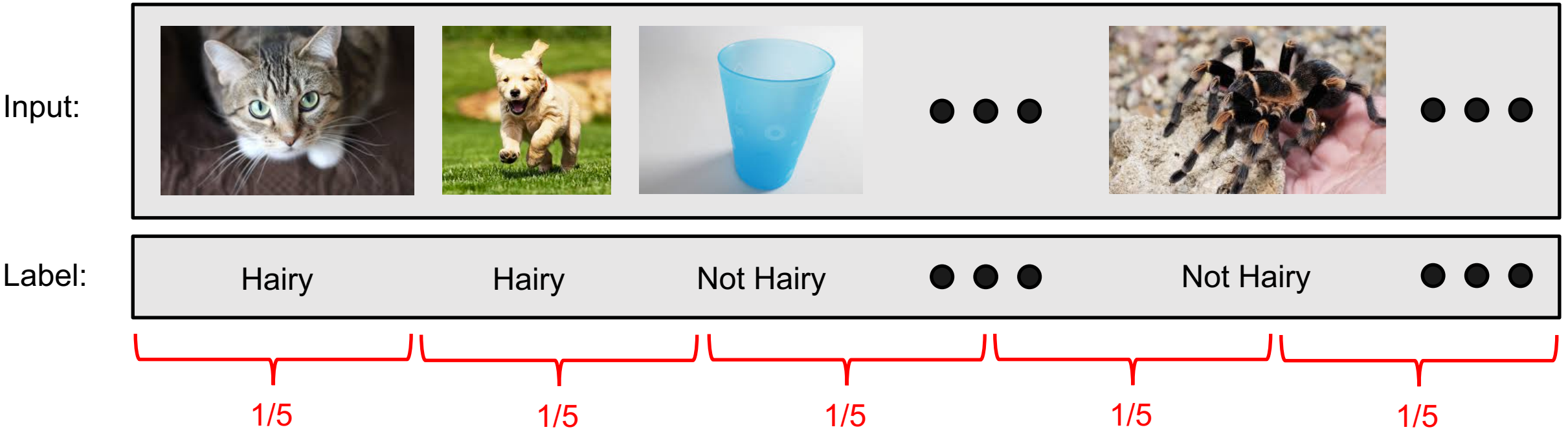
e.g., 3-fold cross-validation

Classifier accuracy: prediction accuracy across all folds of test data



Evaluation of Classification Model

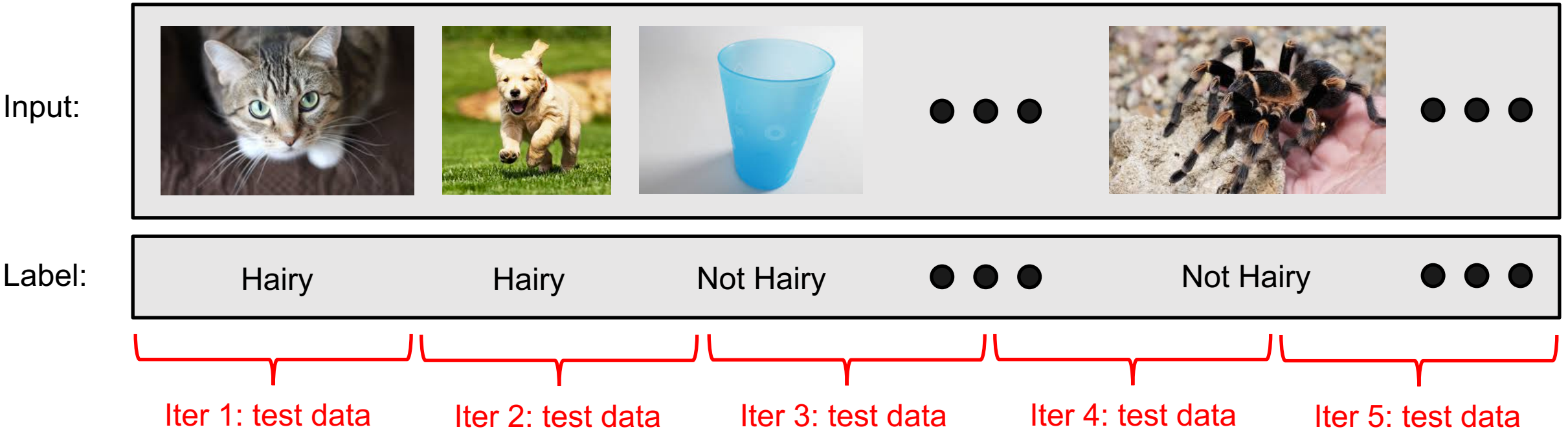
e.g., 5-fold cross-validation



How many partitions of the data to create?

Evaluation of Classification Model

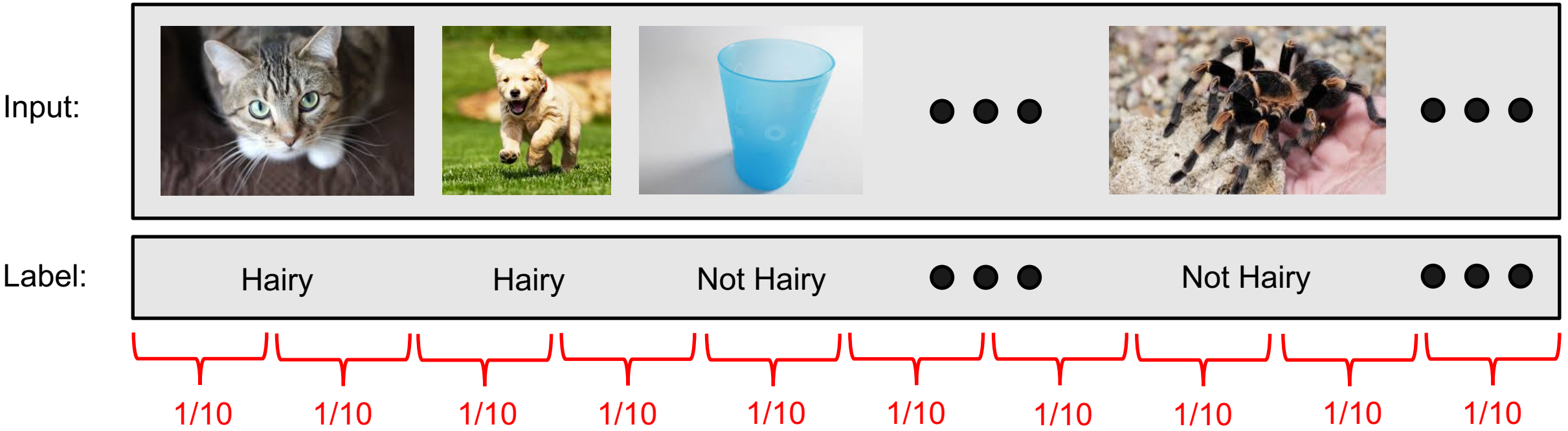
e.g., 5-fold cross-validation



How many iterations of train & test to run?

Evaluation of Classification Model

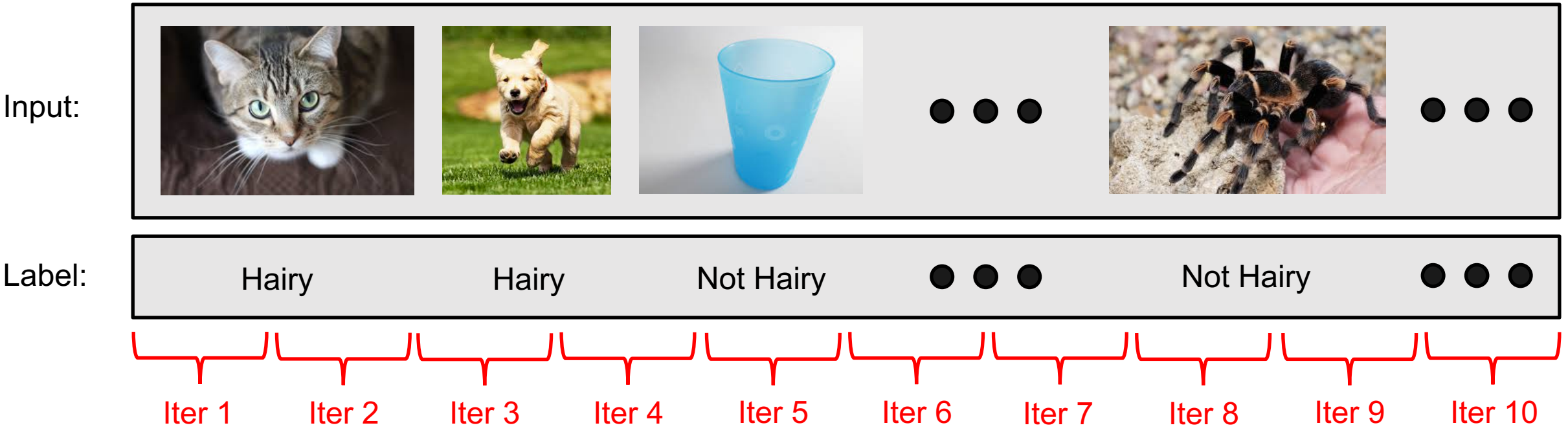
e.g., 10-fold cross-validation



How many partitions of the data to create?

Evaluation of Classification Model

e.g., 10-fold cross-validation



How many iterations of train & test to run?

Evaluation of Classification Model

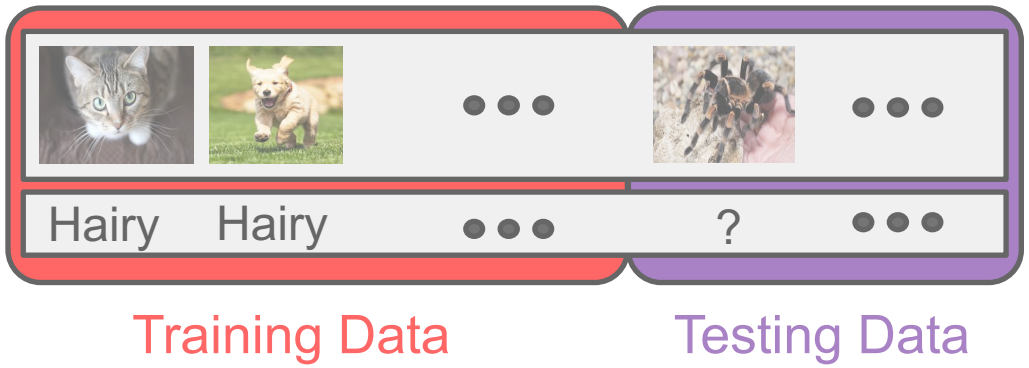
e.g., k-fold cross-validation

What are the (dis)advantages of using larger values for “k”?

Evaluation of Classification Model

Why would you choose percentage split over cross-validation?

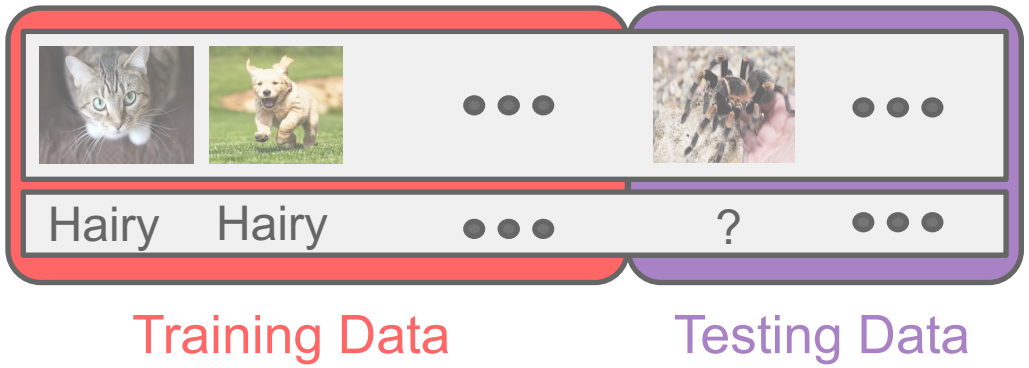
Percentage Split **VS** **Cross-validation (e.g., 3-fold)**



Evaluation of Classification Model

Why would you choose cross-validation over percentage split?

Percentage Split VS **Cross-validation (e.g., 3-fold)**



K-Fold Cross-Validation: How to Partition Data?

- e.g., 3-fold cross validation?

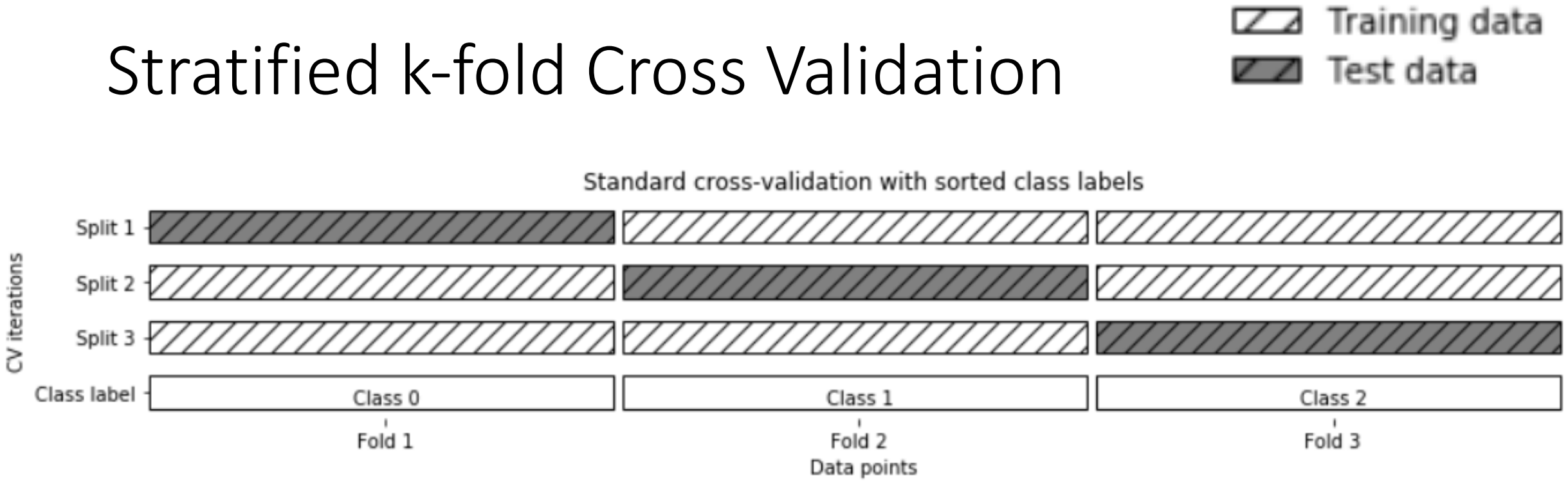
```
In [4]: iris.target

Out[4]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0,
              0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0,
              0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1,
              1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1,
              1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2,
              2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2,
              2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

Stratified k-fold Cross Validation

- e.g., 3-fold cross validation? Preserve class proportions in each fold to represent proportions in the whole dataset

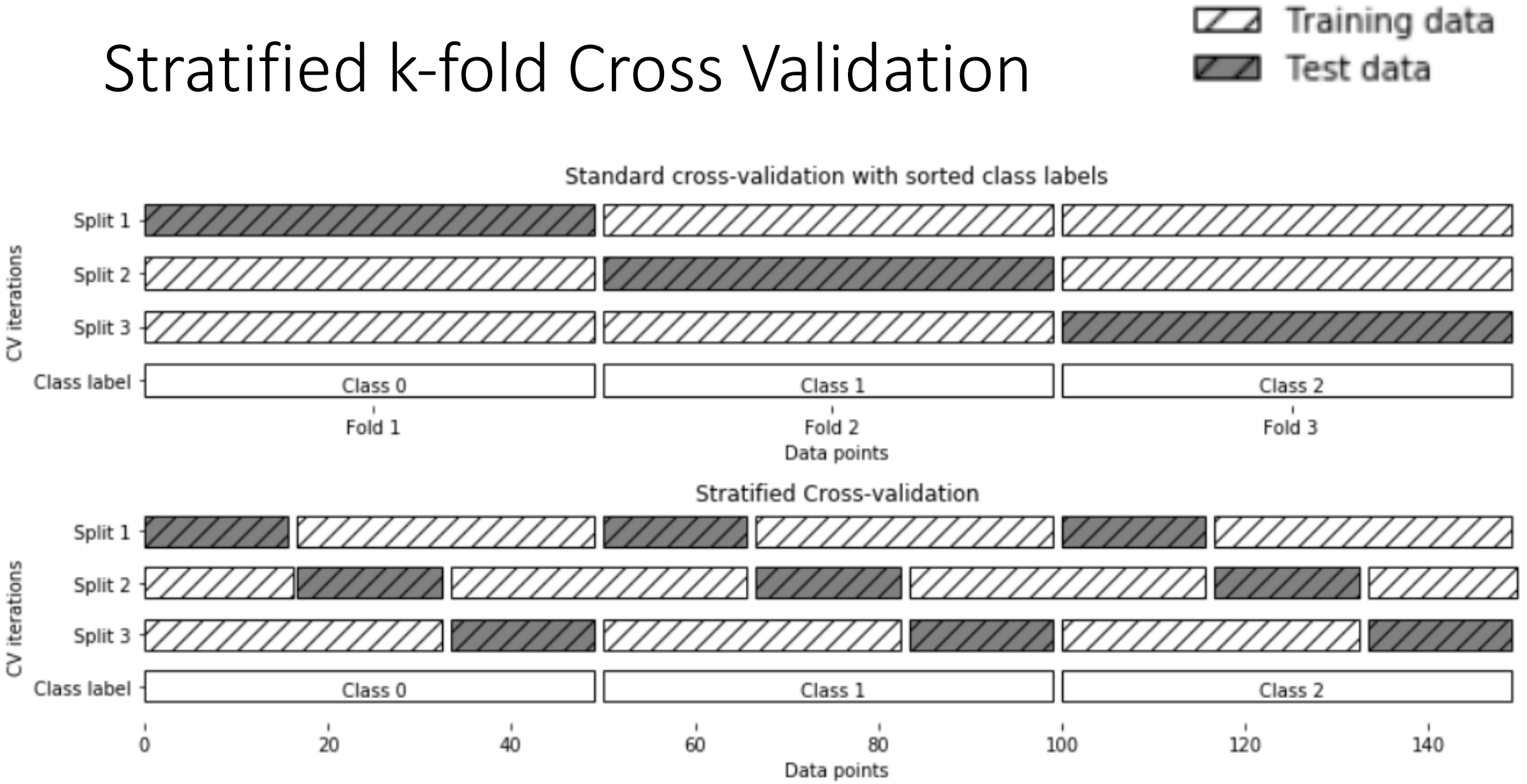
```
In [4]: iris.target

Out[4]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0,
              0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0,
              0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1,
              1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1,
              1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2,
              2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2,
              2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

Stratified k-fold Cross Validation



https://github.com/amueller/introduction_to_ml_with_python/blob/master/05-model-evaluation-and-improvement.ipynb

Stratified k-fold Cross Validation



https://github.com/amueller/introduction_to_ml_with_python/blob/master/05-model-evaluation-and-improvement.ipynb

Cross Validation: Additional Notes

- Implications of high variance of test accuracy between different folds?
  - Model is very dependent on folds used for training
  - Consequence of small dataset size

- Cross validation is NOT a way to build a model that can be applied to new data. It does not return a single model.

Today's Topics

- Nearest Neighbor Classifier
- Support Vector Machines
- Evaluating Classifiers Using Cross-Validation
- Tuning Hyper-parameters
- Lab

Recall: When Applying Supervised ML Model

1. Choose a model; e.g., ?
   - Linear Regression, Decision Tree, Naïve Bayes, Nearest Neighbors, SVM
2. Choose model parameters; e.g., ?
   - Linear Regression: regularization weight, Decision Tree: depth, Nearest Neighbors: # neighbors, SVM: kernel type

Hyperparameters

3. Fit the model to the training data
4. Use model to predict labels for new data

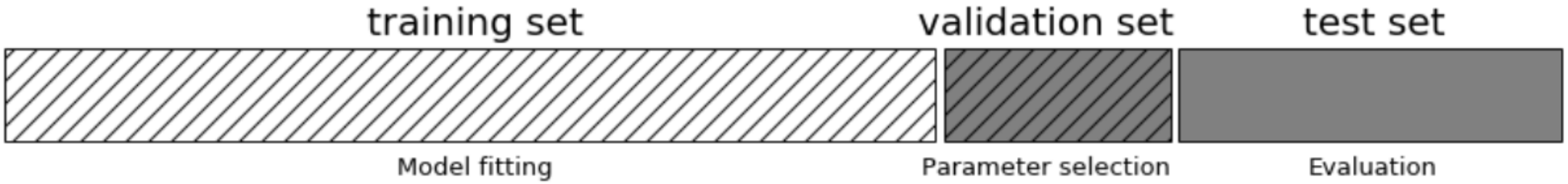
What are hyperparameters?

- Not parameters learned from the training data:
  - e.g., linear regression weights
  - e.g., decision tree splitting attributes
  - e.g., SVM weights

- Parameters we choose:
  - e.g., regularization parameters for Ridge and Lasso regression
  - e.g., “k” for K-NN;
  - e.g., “C” and “gamma” for SVM
  - e.g., split type or depth for Decision Tree

Tuning Hyperparameters

- Cannot evaluate model on the data used to tune hyperparameters
- On training data:
  1. Cross-validation
  2. Percentage split into “train” and “validation” datasets



- Algorithm: brute-force, exhaustive approach by evaluating every hyperparameter combination to find optimal hyperparameters

https://github.com/amueller/introduction_to_ml_with_python/blob/master/05-model-evaluation-and-improvement.ipynb

Today's Topics

- Nearest Neighbor Classifier
- Support Vector Machines
- Evaluating Classifiers Using Cross-Validation
- Tuning Hyper-parameters
- Lab

Microsoft Azure

- Set-up an account