SciPy2016

Scientific Computing with Python
Austin, Texas • July 11-17, 2016

Machine Learning with scikit learn

Sebastian Raschka & Andreas Mueller

Links

Tutorial Material on GitHub:

https://github.com/amueller/scipy-2016-sklearn

Contact Info:

Sebastian Raschka

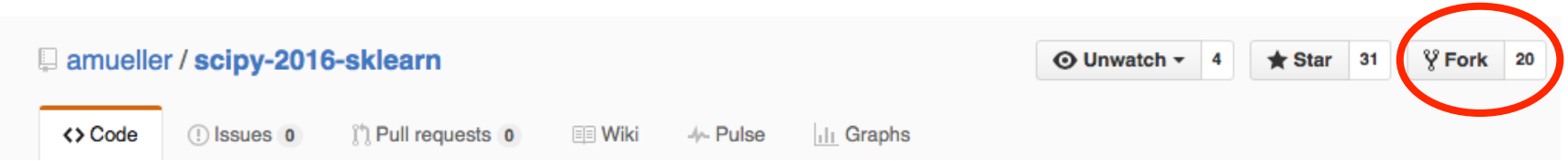
- mail@sebastianraschka.com
- http://sebastianraschka.com
- @rasbt

Andreas Mueller

- t3kcit+website@gmail.com
- http://amueller.github.io
- @amuellerml

Tutorial Setup I

a) Fork the Repository (if you haven't done so, yet):



b) Sync an older fork:

```
$ git remote add upstream https://github.com/amueller/scipy-2016-sklearn.git
$ git fetch upstream
$ git checkout master merge upstream/master
```

Tutorial Setup II

```
jupyter notebook check_env.ipynb
```

```
Using python in /Users/Sebastian/miniconda3
3.5.1 |Continuum Analytics, Inc.| (default, Jun 15 2016, 16:14:02)
[GCC 4.2.1 Compatible Apple LLVM 4.2 (clang-425.0.28)]

[ OK ] IPython version 4.2.0
[ OK ] numpy version 1.11.0
[ OK ] watermark version 1.3.1
[ OK ] matplotlib version 1.5.1
[ OK ] scipy version 0.17.1
[ OK ] yaml version 3.11
[ OK ] PIL version 1.1.7
[ OK ] sklearn version 0.17.1
[ OK ] pydot version 1.2.2
```

```
python fetch_data.py
```

~456 MB!!!

Our Agenda

- Morning Session: 8:00 AM - 12:00 PM (Room 105)
- Afternoon Session: 1:30 PM - 5:30 PM (Room 105)

Morning Session 8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

10 Preparing a real-world dataset (titanic)

11 Working with text data via the bag-of-words model

12 Application: SMS spam classification

Afternoon Session 1:30 PM - 5:30 PM

13 Cross-Validation

14 Model complexity and grid search for adjusting hyperparameters

15 Scikit-learn Pipelines

16 Supervised learning: Performance metrics for classification

17 Supervised learning: Linear Models

18 Supervised learning: Support Vector Machines

19 Supervised learning: Decision trees and random forests, ensemble methods

20 Supervised learning: feature selection

21 Unsupervised learning: Hierarchical and density-based clustering algorithms

22 Unsupervised learning: Non-linear dimensionality reduction

23 Supervised learning: Out-of-core learning

Morning Session 8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

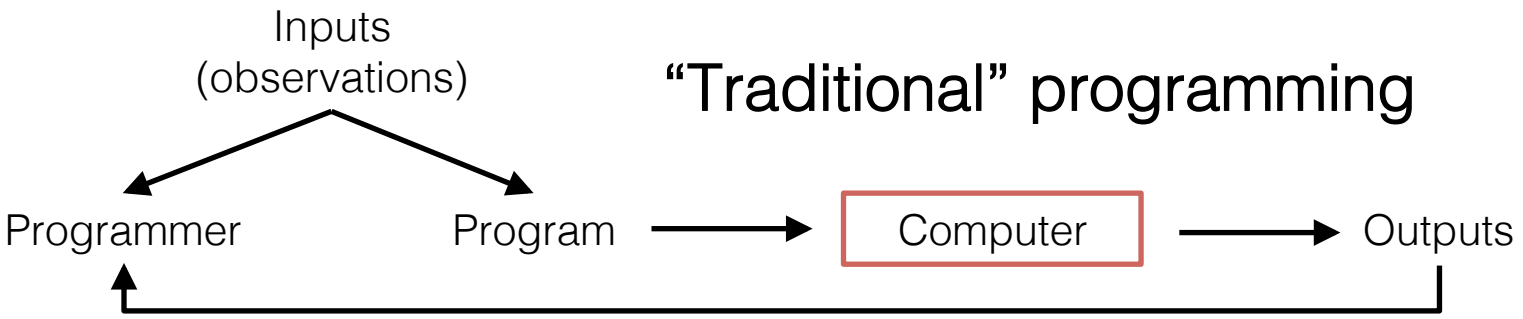
10 Preparing a real-world dataset (titanic)

11 Working with text data via the bag-of-words model

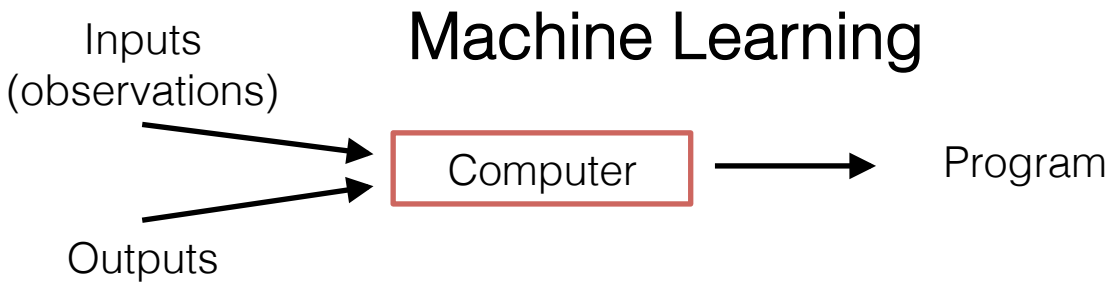
12 Application: SMS spam classification

S

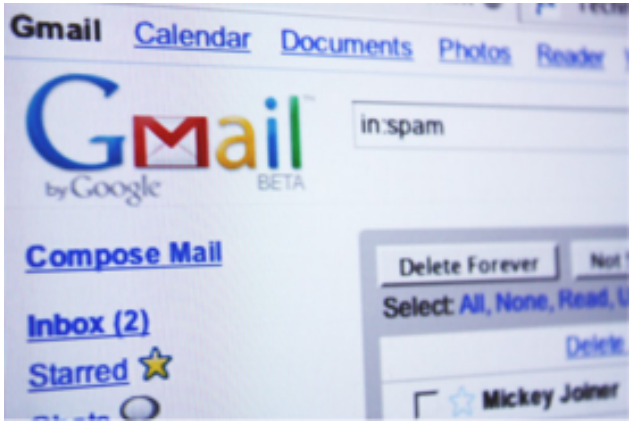
What is Machine Learning?



Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.
-- Arthur Samuel (1959)



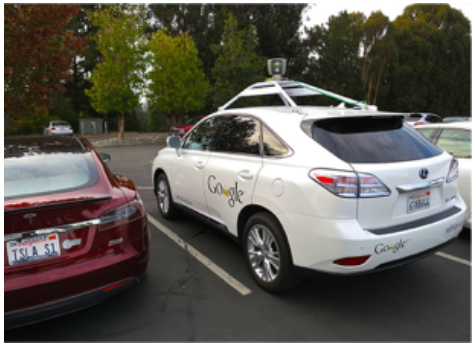
Examples of Machine Learning



https://flic.kr/p/5BLW6G [CC BY 2.0]

http://commons.wikimedia.org/wiki/File:American_book_company_1916._letter_envelope-2.JPG#filelinks [public domain]

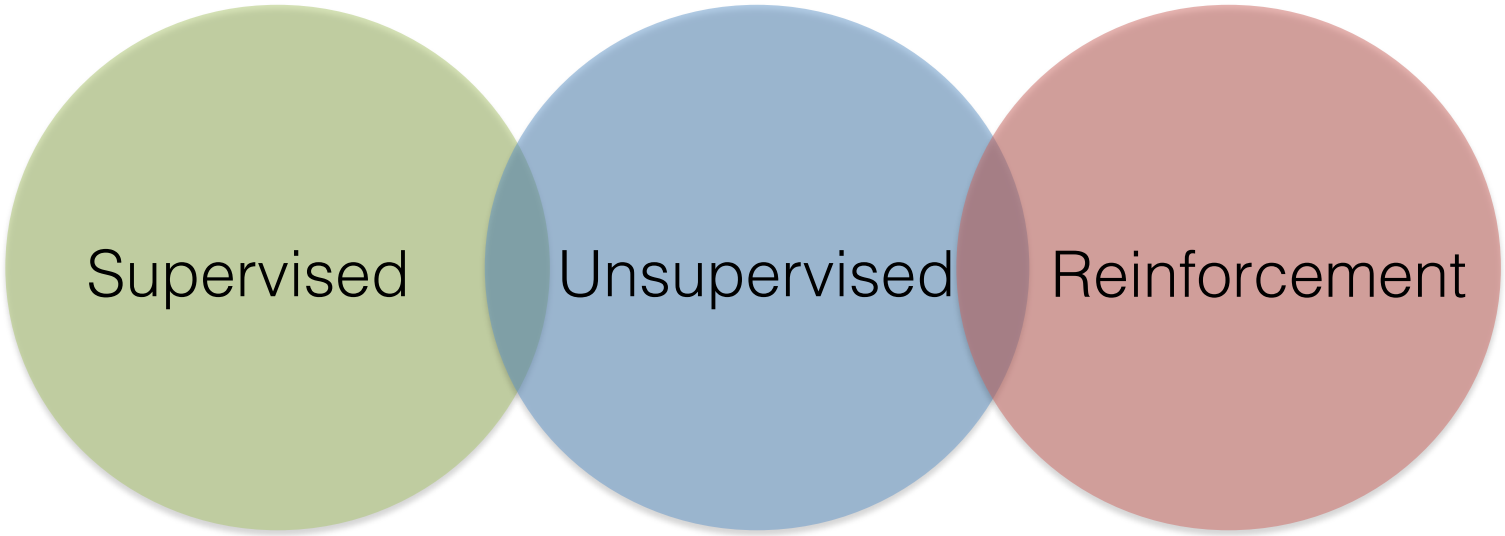
http://commons.wikimedia.org/wiki/File:Netflix_logo.svg [public domain]



By Steve Jurvetson [CC BY 2.0]

And many, many more …

3 Types of Learning

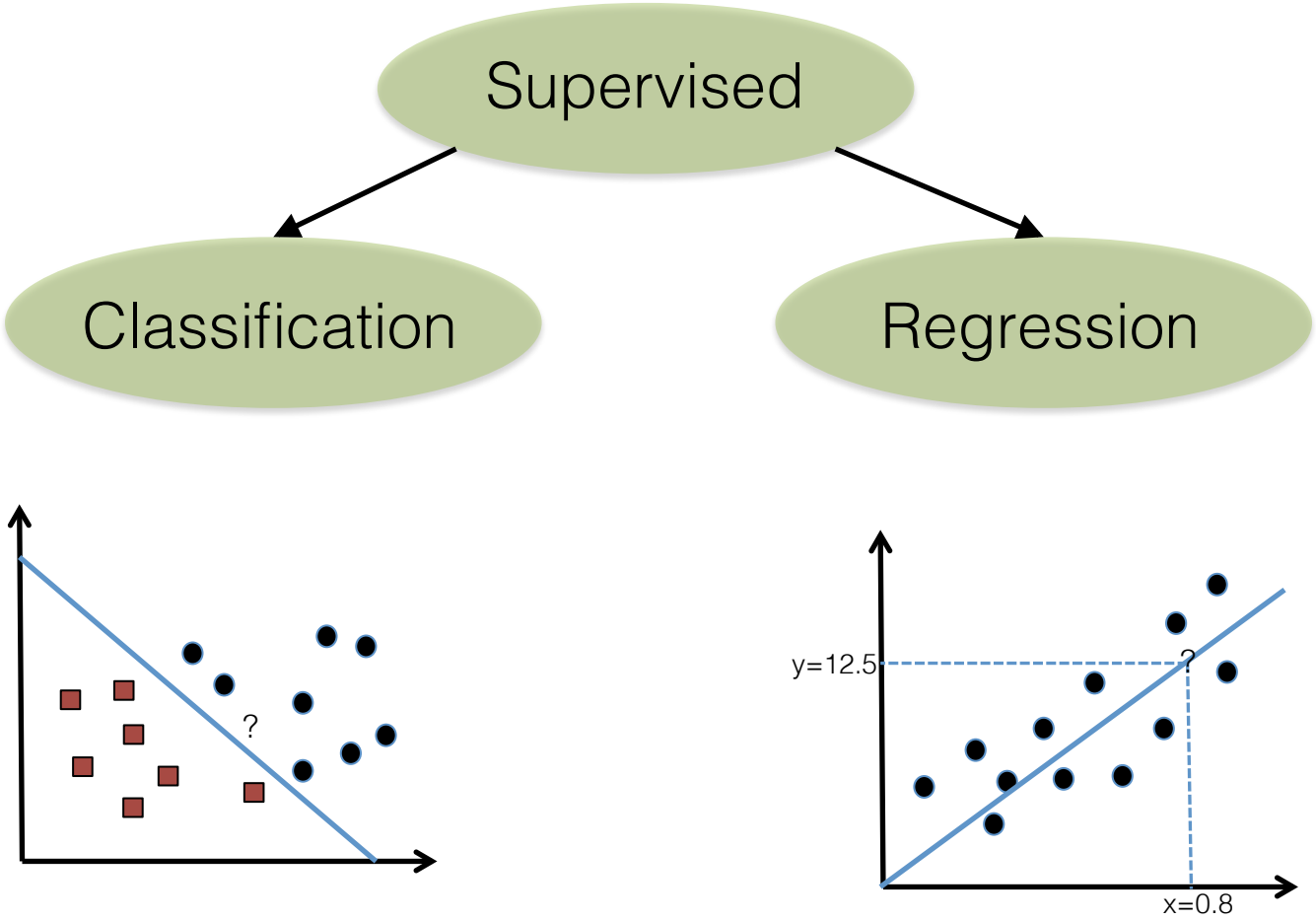


- Learning from labeled data
- E.g., Spam classification

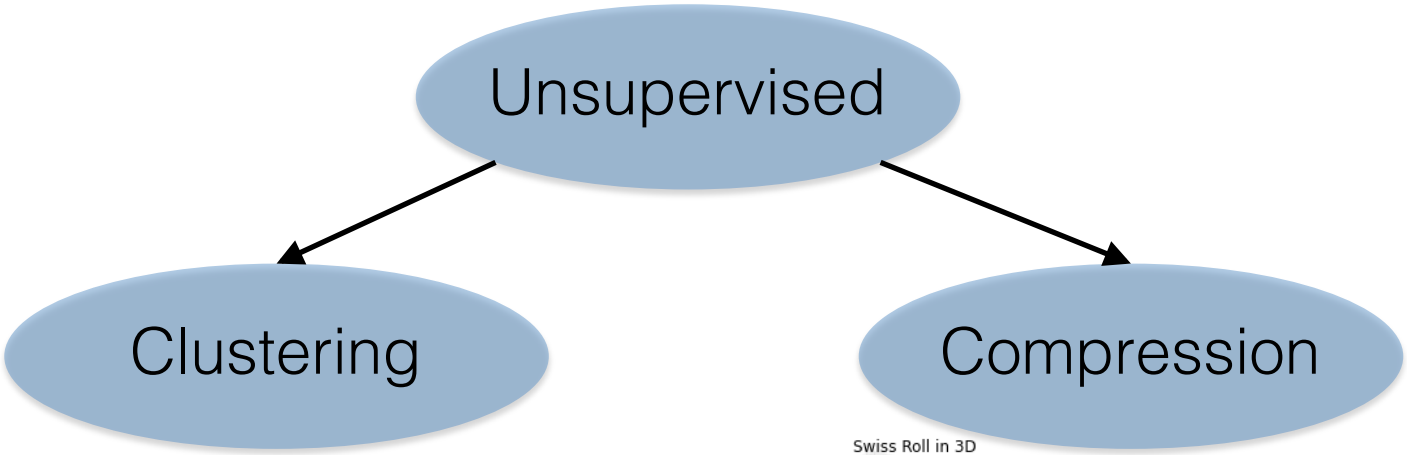
- Discover structure in unlabeled data
- E.g., Document clustering

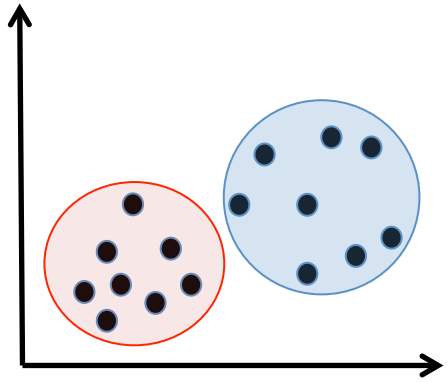
- Learning by "doing" with delayed reward
- E.g., Chess computer

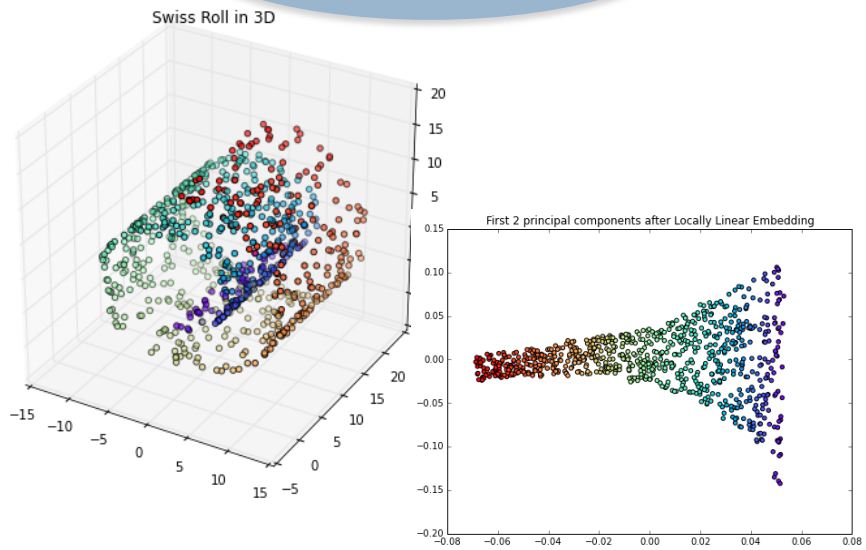
Supervised Learning



Unsupervised Learning







Flower Classification

Iris-Setosa

Iris-Versicolor

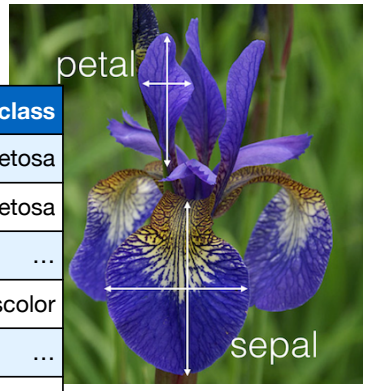
Iris-Setosa

Data Representation

IRIS

https://archive.ics.uci.edu/ml/datasets/Iris

Instances (samples, observations)



	sepal_length	sepal_width	petal_length	petal_width	class
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
...	...	...	...	...	...
50	6.4	3.2	4.5	1.5	veriscolor
...	...	...	...	...	...
150	5.9	3.0	5.1	1.8	virginica

Features (attributes, dimensions)

Classes (targets)

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ x_{31} & x_{32} & \cdots & x_{3D} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix}$$

$$\mathbf{y} = [y_1, y_2, y_3, \cdots y_N]$$

Morning Session

8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

10 Preparing a real-world dataset (titanic)

11 Working with text data via the bag-of-words model

12 Application: SMS spam classification

S

Jupyter Notebooks

jupyter 03 Data Representation for Machine Learning (unsaved changes)

File Edit View Insert Cell Kernel Help Python 3

Markdown CellToolbar

(Image: "Petal-sepal". Licensed under CC BY-SA 3.0 via Wikimedia Commons - https://commons.wikimedia.org/wiki/File:Petal-sepal.jpg#/media/File:Petal-sepal.jpg)

`scikit-learn` embeds a copy of the iris CSV file along with a helper function to load it into numpy arrays:

In [2]:
```
from sklearn.datasets import load_iris
iris = load_iris()
```

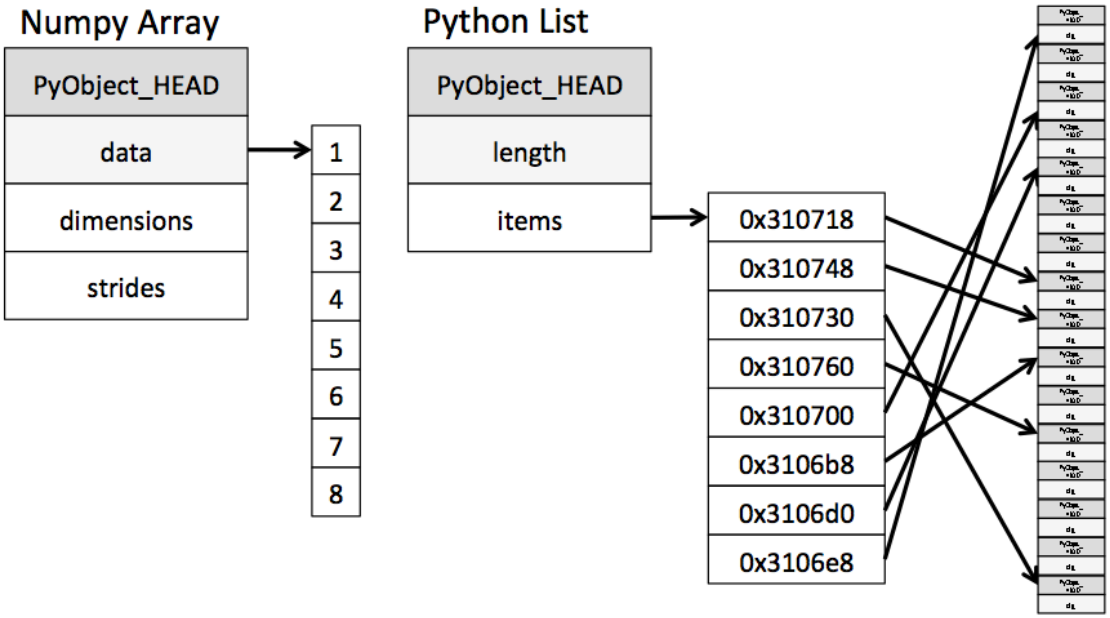
The resulting dataset is a `Bunch` object: you can see what's available using the method `keys()`:

In [3]:
```
iris.keys()
```

Out[3]:
```
dict_keys(['target_names', 'data', 'feature_names', 'DESCR', 'target'])
```

NumPy Arrays

- build around a C array with pointers to a contiguous data buffer of values
- Linear algebra functions
- Fancy indexing
- …




```
>>> import numpy
>>> ary = numpy.array([7, 8, 9, 10, 11])
>>> ary[[2, 4]]
array([ 9, 11])
>>> lst = list([7, 8, 9, 10, 11])
>>> lst[2, 4]]
>>> lst[[2, 4]]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: list indices must be
integers or slices, not list
```

Image source: "Why Python is Slow: Looking Under the Hood" by Jake VanderPlas
http://jakevdp.github.io/blog/2014/05/09/why-python-is-slow/

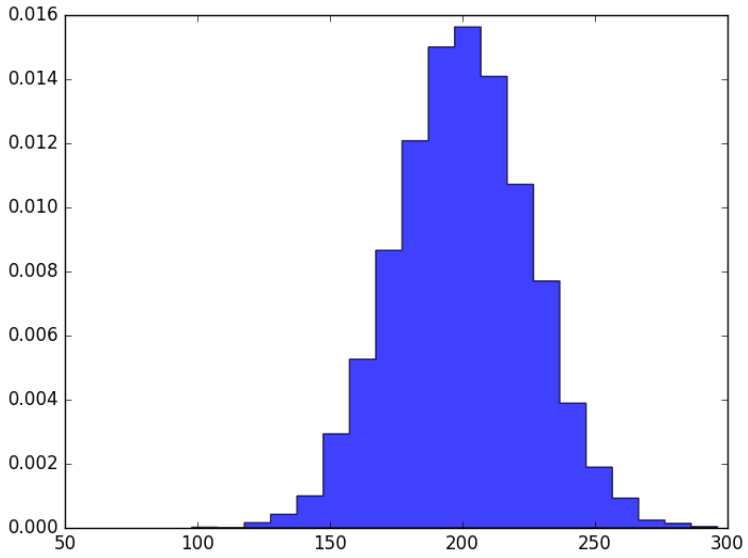
Scipy Sparse Matrices

List of Lists (LIL) example

```
>>> from scipy import sparse
>>> mtx = sparse.lil_matrix([[0, 1, 2, 0],
...                          [3, 0, 1, 0],
...                          [1, 0, 0, 1]])
>>> print(mtx)
  (0, 1)  1
  (0, 2)  2
  (1, 0)  3
  (1, 2)  1
  (2, 0)  1
  (2, 3)  1
>>> print(mtx.toarray())
[[0 1 2 0]
 [3 0 1 0]
 [1 0 0 1]]
```

Matplotlib

```
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>>
>>> mu, sigma = 200, 25
>>> x = mu + sigma*np.random.randn(10000)
>>> plt.hist(x, 20, normed=1,
...          histtype='stepfilled',
...          facecolor='b',
...          alpha=0.75)
>>> plt.show()
```



Morning Session

8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

10 Preparing a real-world dataset (titanic)

11 Working with text data via the bag-of-words model

12 Application: SMS spam classification

S

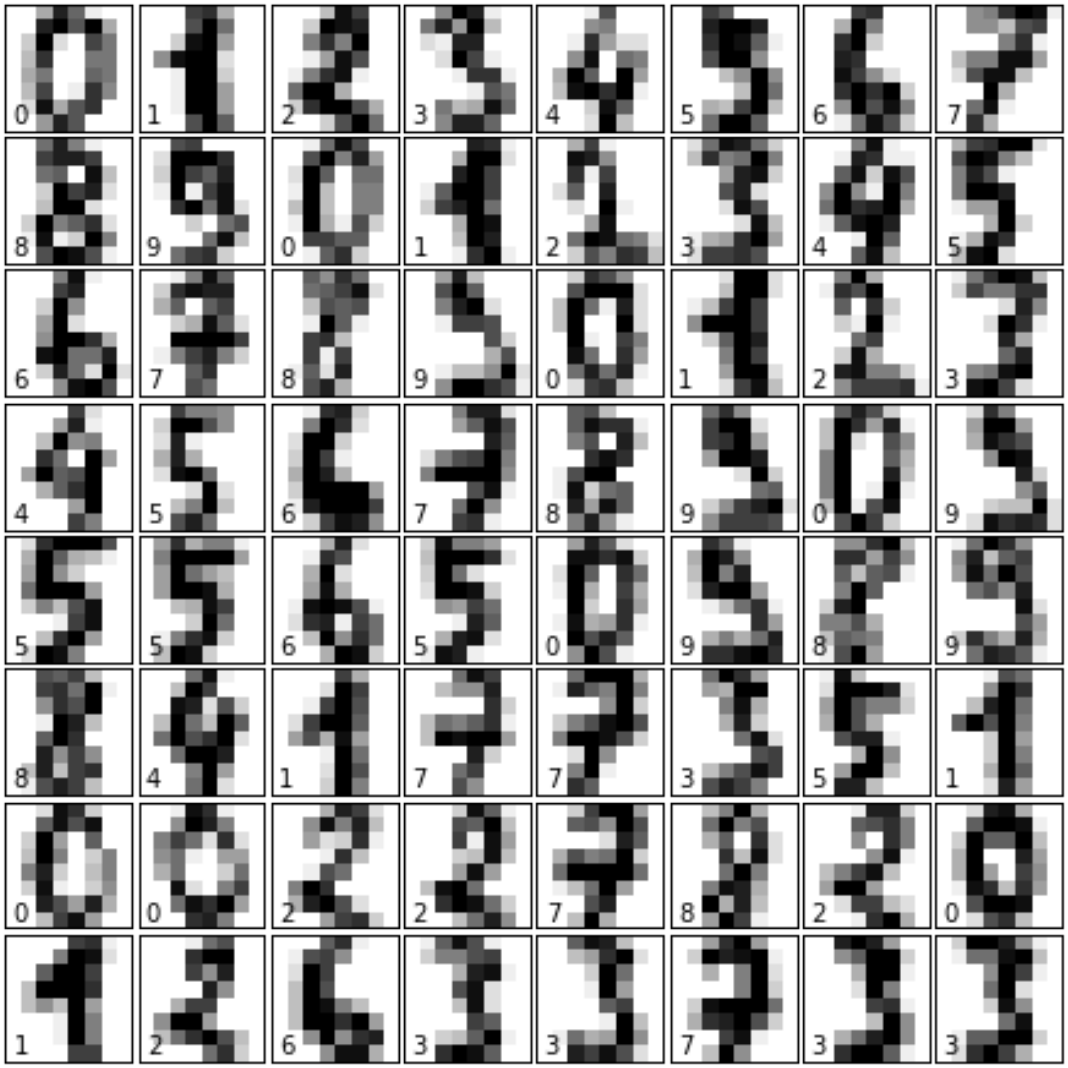
Iris

Iris-Setosa

Iris-Versicolor

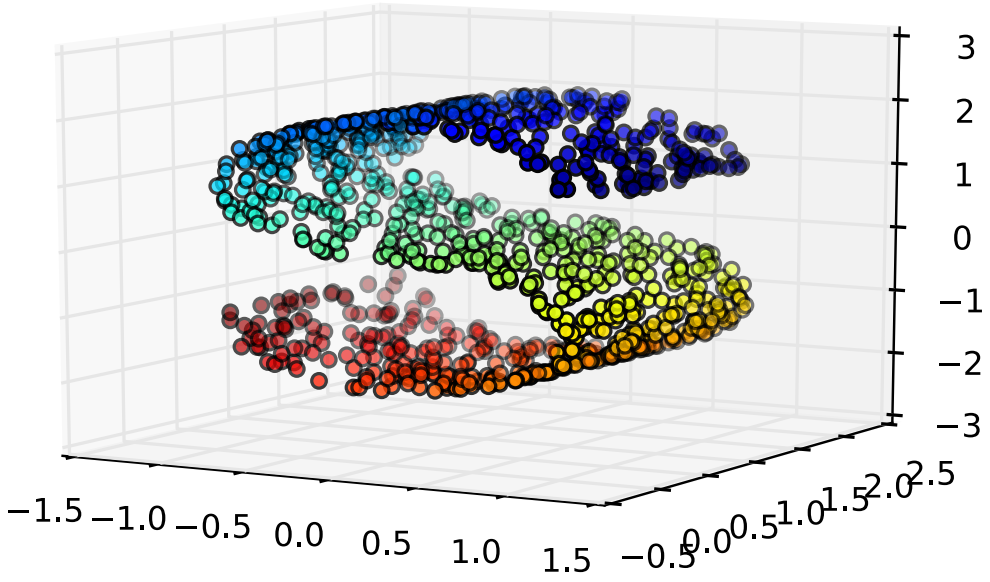
Iris-Setosa

Digits


0 1 2 3 4 5 6 7
8 9 0 1 2 3 4 5
6 7 8 9 0 1 2 3
4 5 6 7 8 9 0 9
5 5 6 5 0 9 8 9
8 4 1 7 7 3 5 1
0 0 2 2 7 8 2 0
1 2 6 3 3 7 3 3

Generating Synthetic Data

```
from sklearn.datasets import make_…
```



Morning Session

8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

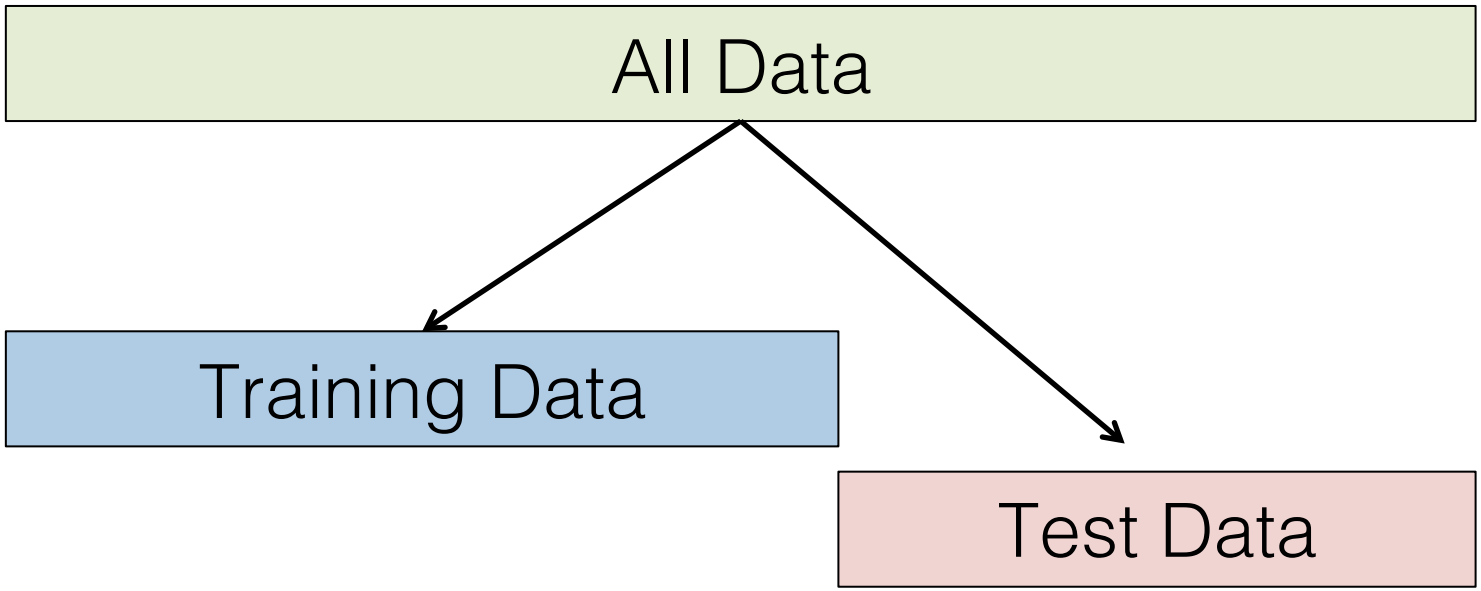
10 Preparing a real-world dataset (titanic)

11 Working with text data via the bag-of-words model

12 Application: SMS spam classification

A

Training & Test Data

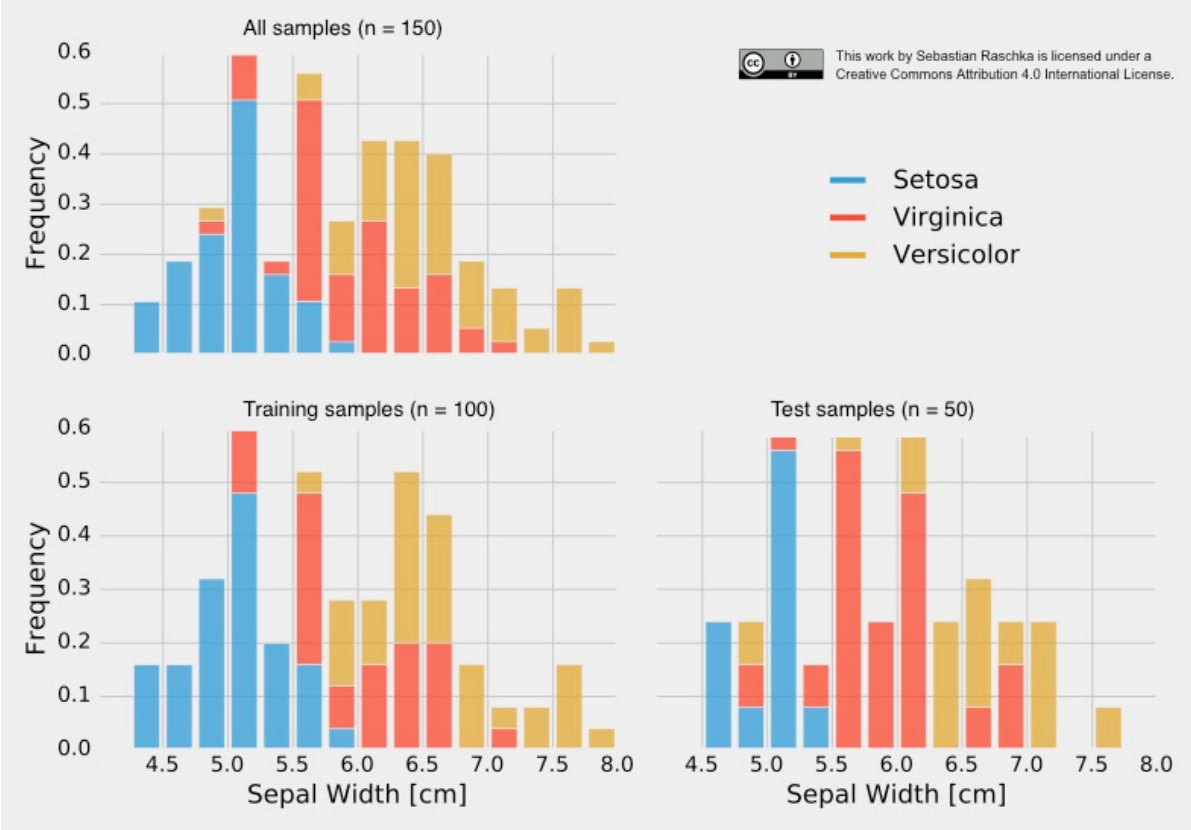


Typically:

- 75% : 25%
- 2/3 : 1/3

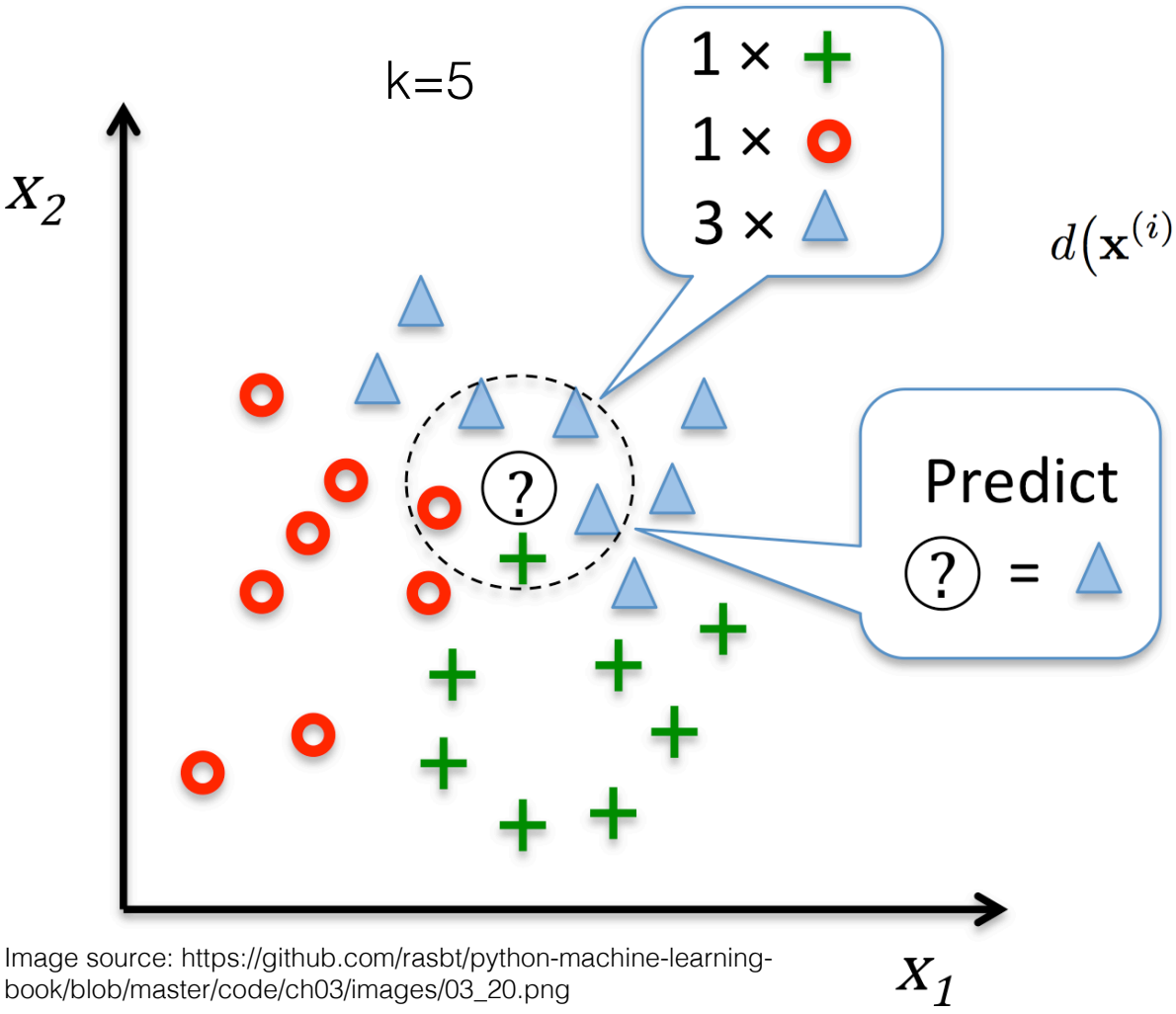
Stratification

Non-stratified split:



- training set → 38 x Setosa, 28 x Versicolor, 34 x Virginica
- test set → 12 x Setosa, 22 x Versicolor, 16 x Virginica

K-Nearest Neighbors



$$d\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \sqrt[p]{\sum_k \left|x_k^{(i)} - x_k^{(j)}\right|^p}$$

Image source: https://github.com/rasbt/python-machine-learning-book/blob/master/code/ch03/images/03_20.png

Morning Session 8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

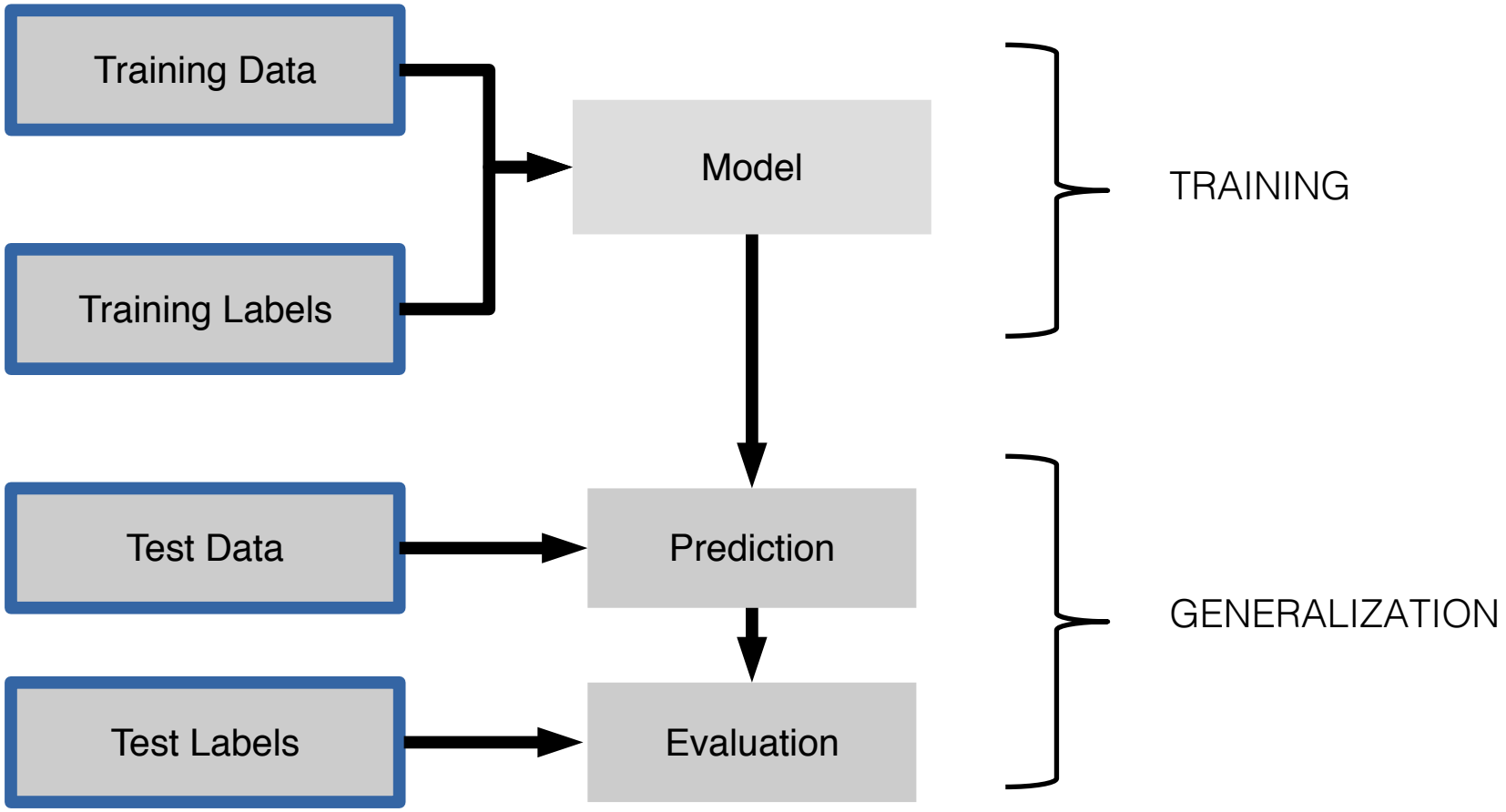
10 Preparing a real-world dataset (titanic)

11 Working with text data via the bag-of-words model

12 Application: SMS spam classification

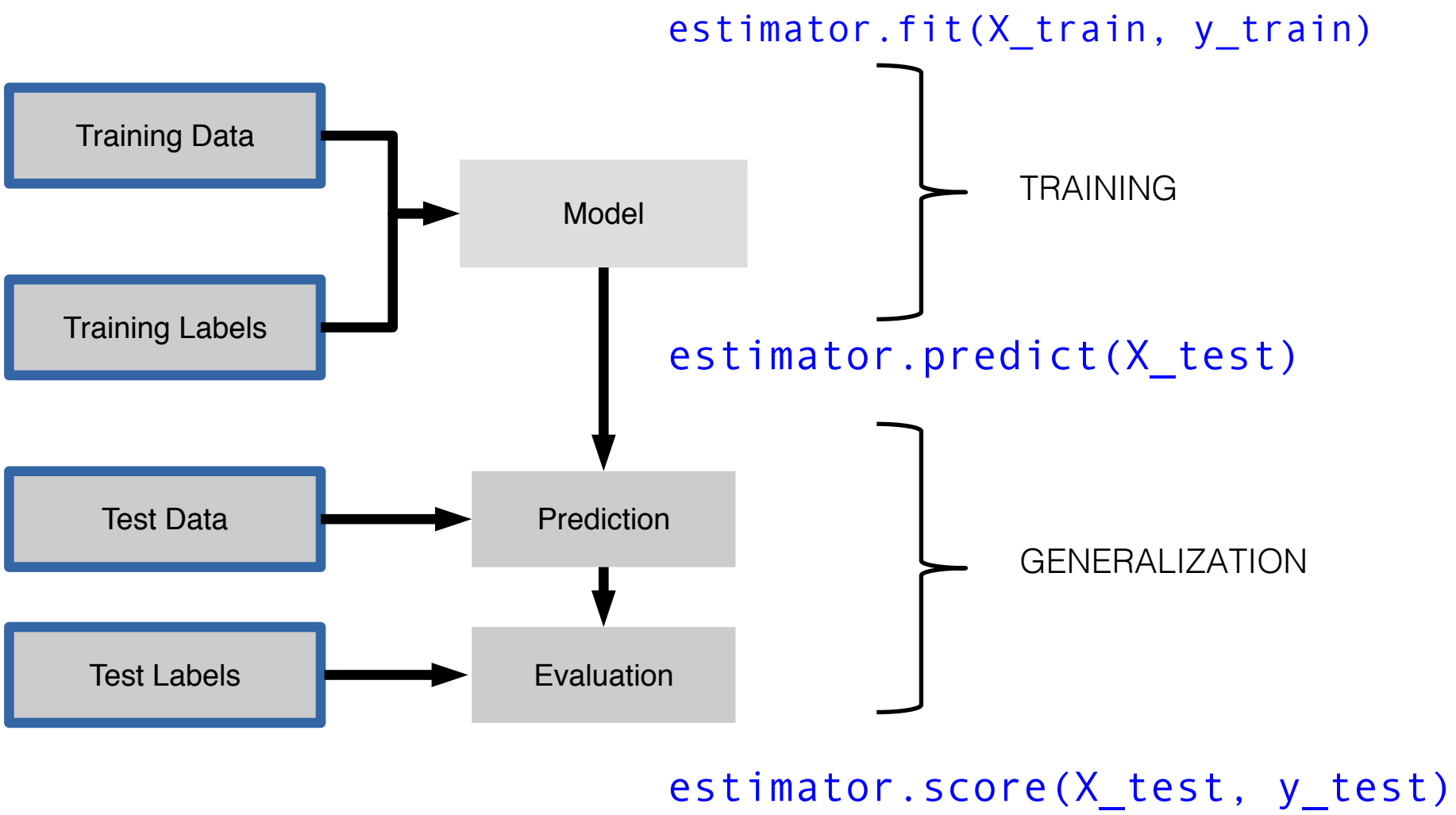
A

Supervised Workflow



- Fit model on all data after evaluation

Supervised Workflow



Morning Session 8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

10 Preparing a real-world dataset (titanic)

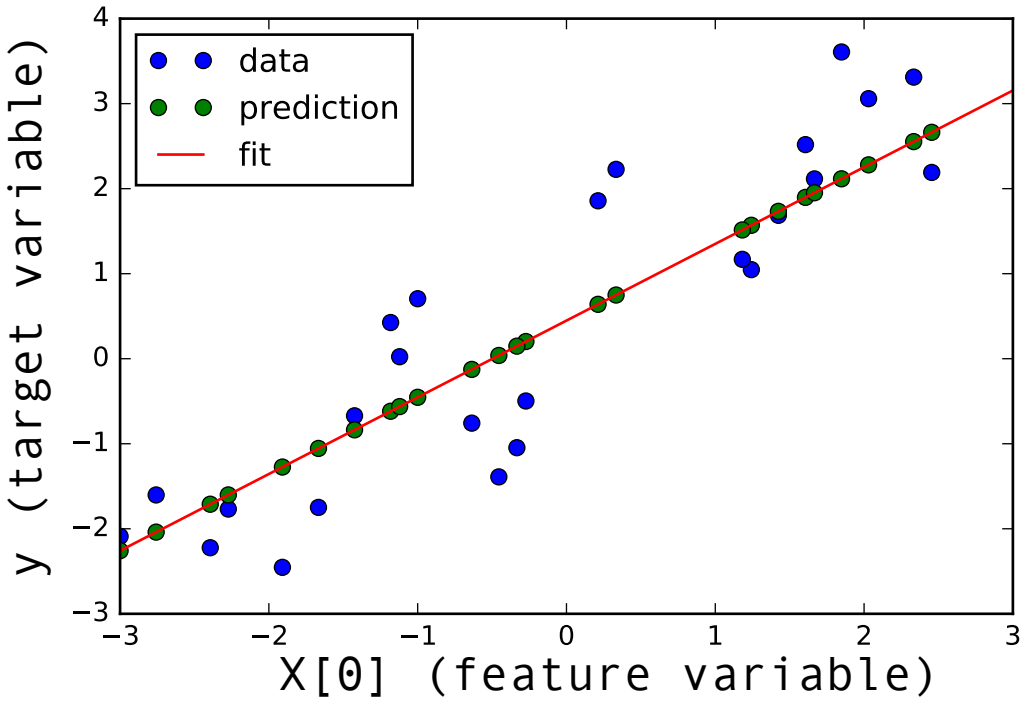
11 Working with text data via the bag-of-words model

12 Application: SMS spam classification

A

Linear Regression

```
y =coef_[0]*X[0] + intercept_
```



Morning Session

8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

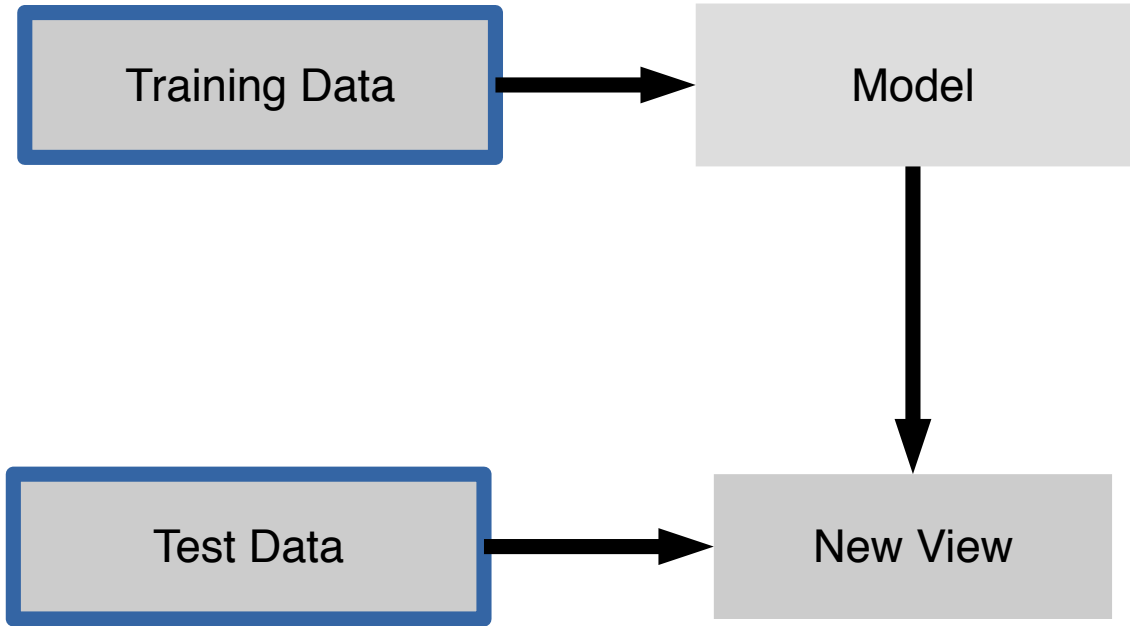
10 Preparing a real-world dataset (titanic)

11 Working with text data via the bag-of-words model

12 Application: SMS spam classification

S

Unsupervised Transformers



① `transformer.fit(X_train)`
② `X_train_transf = transformer.transform(X_train)`
③ `X_test_transf  = transformer.transform(X_test)`

Feature Scaling

$$x^{(i)}_{std} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

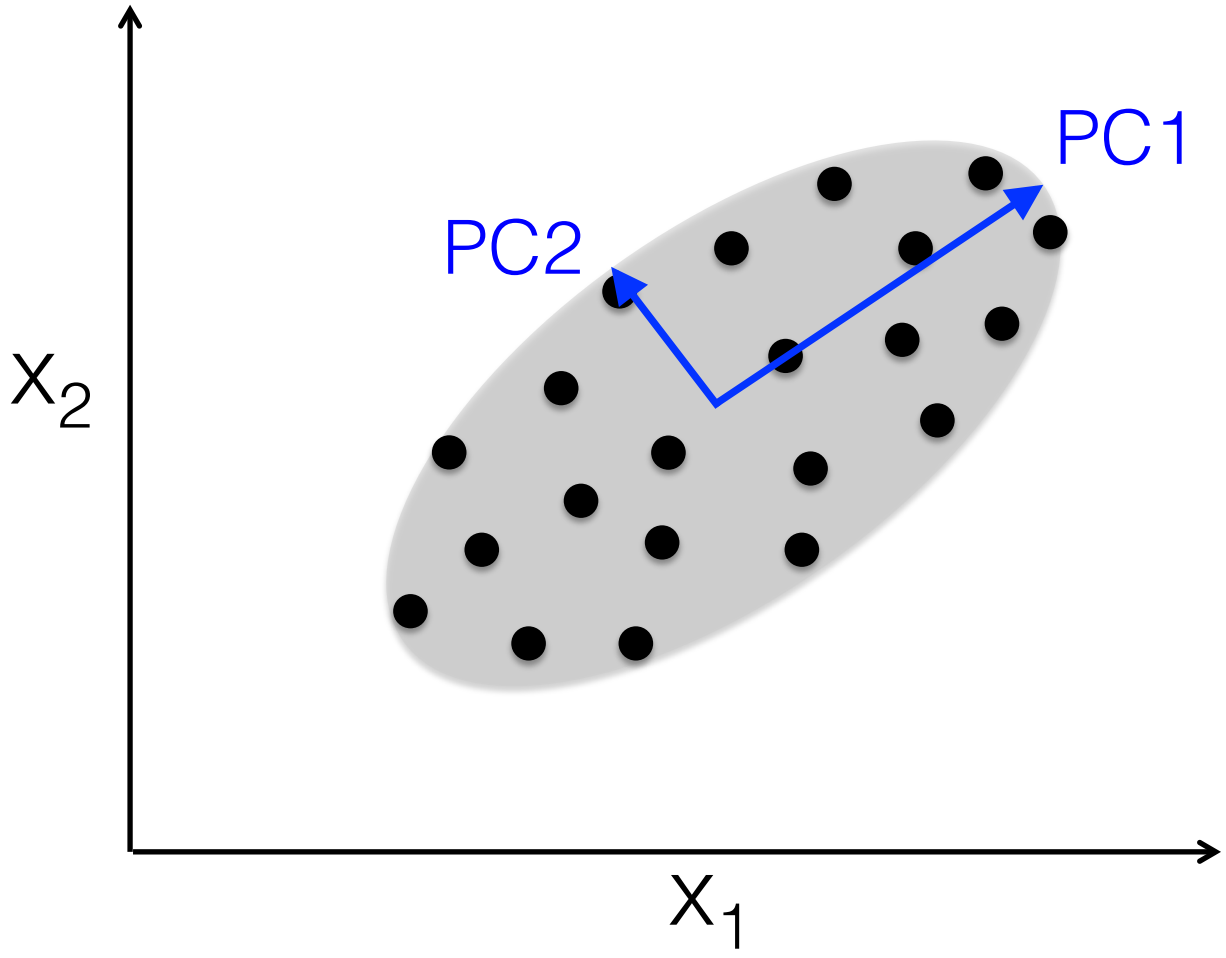
standardization

$$x^{(i)}_{norm} = \frac{x^{(i)} - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}}$$

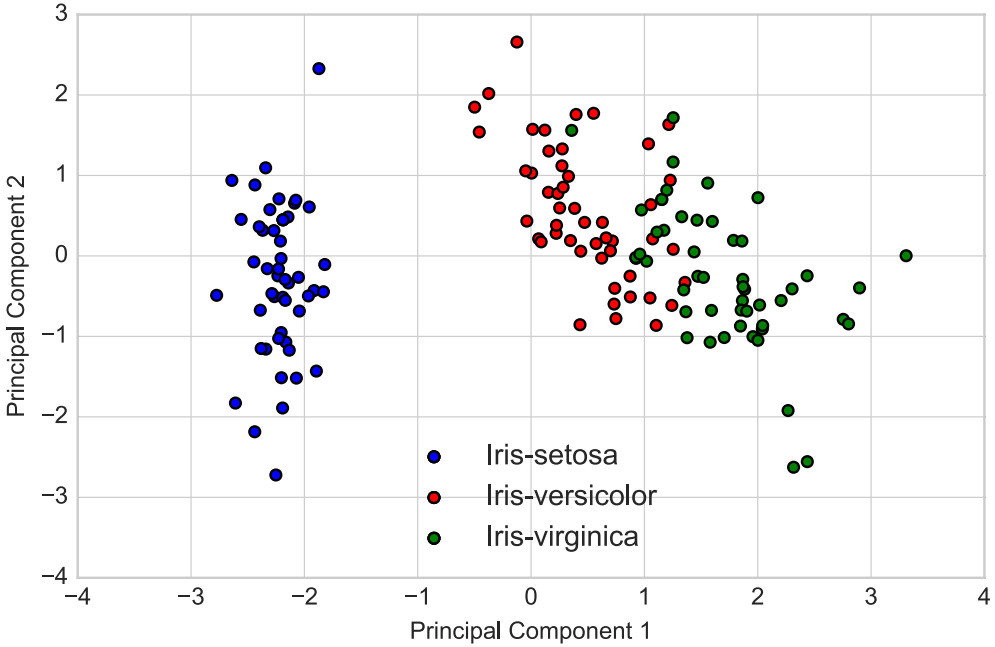
min-max scaling ("normalization")

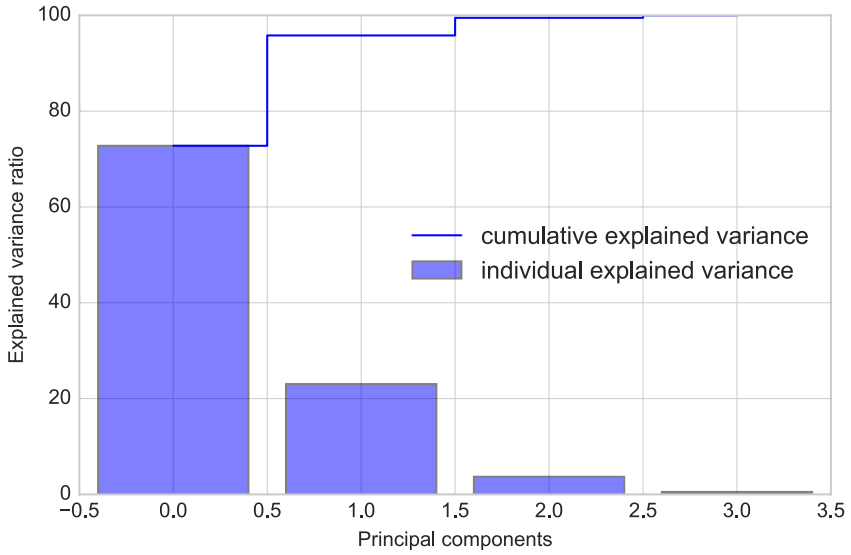
	input	standardized	normalized
0	0	-1.46385	0.0
1	1	-0.87831	0.2
2	2	-0.29277	0.4
3	3	0.29277	0.6
4	4	0.87831	0.8
5	5	1.46385	1.0

Principal Component Analysis



PCA for Dimensionality Reduction





Morning Session

8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

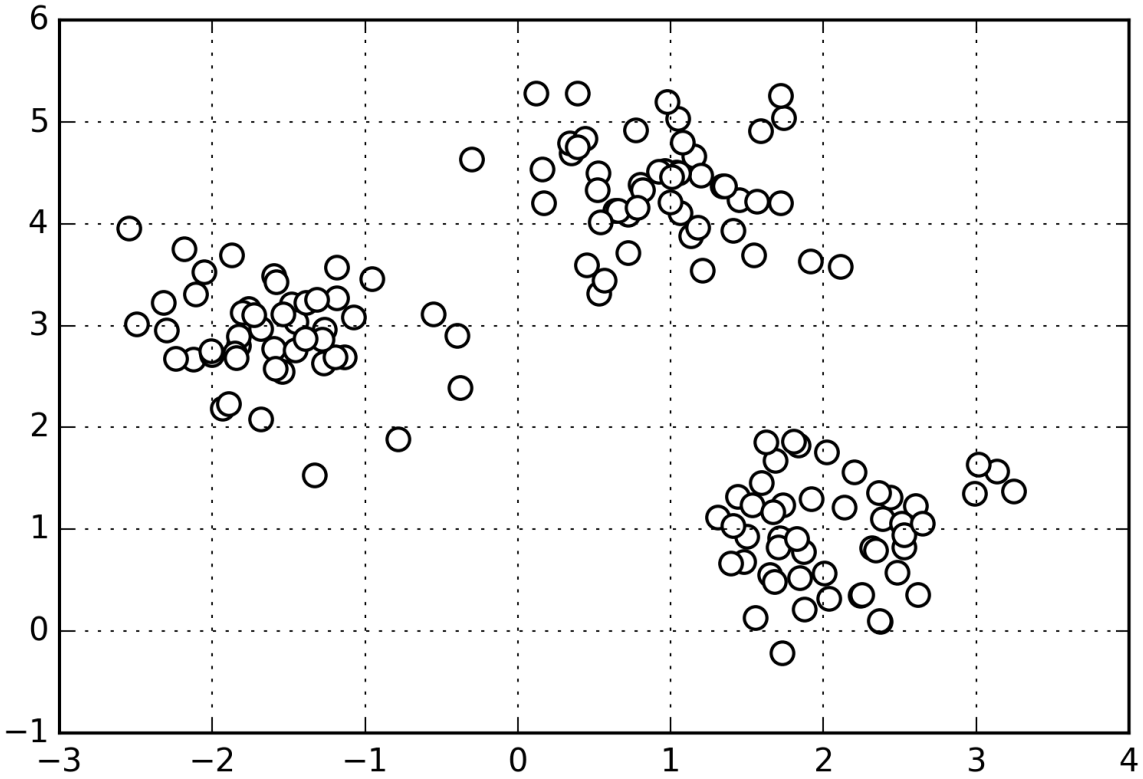
10 Preparing a real-world dataset (titanic)

11 Working with text data via the bag-of-words model

12 Application: SMS spam classification

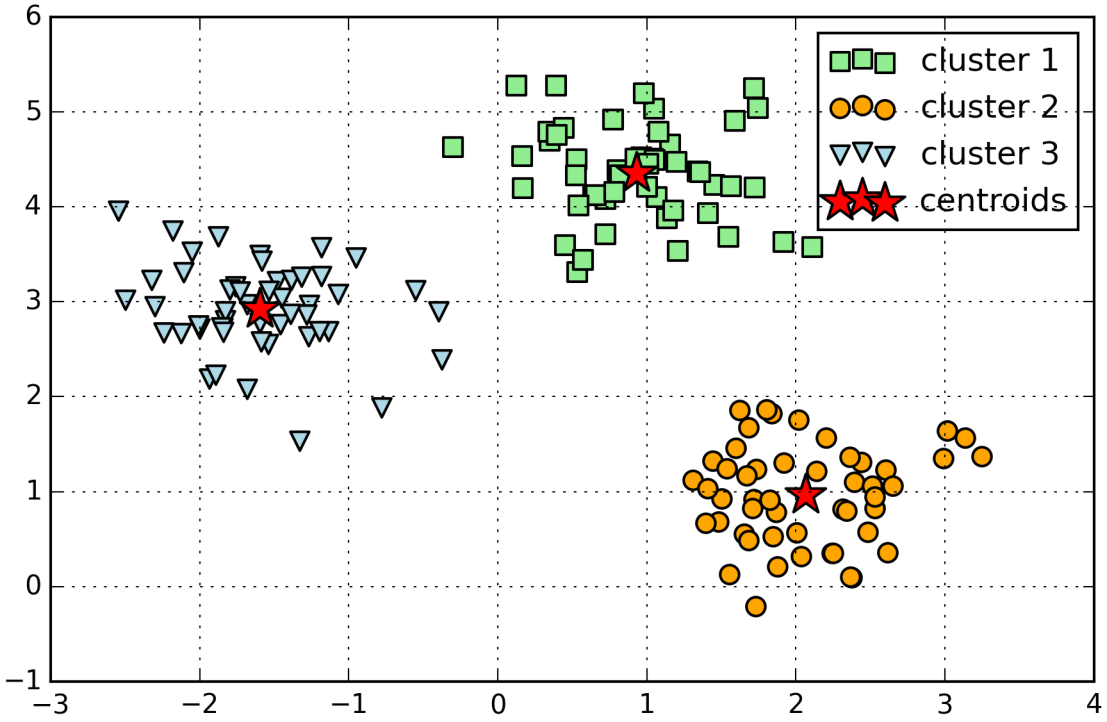
S

K-means Clustering



K-means Clustering

$$d(\boldsymbol{x}, \boldsymbol{y})^2 = \sum_{j=1}^{m} \left(x_j - y_j\right)^2 = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$



Morning Session 8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

10 Preparing a real-world dataset (titanic)

11 Working with text data via the bag-of-words model

12 Application: SMS spam classification

A

Scikit-learn API

`estimator.fit(X_train, [y_train])`	
`estimator.predict(X_test)`	**`estimator.transform(X_test)`**
Classification	Preprocessing
Regression	Dimensionality Reduction
Clustering	Feature Extraction
	Feature selection

Morning Session 8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

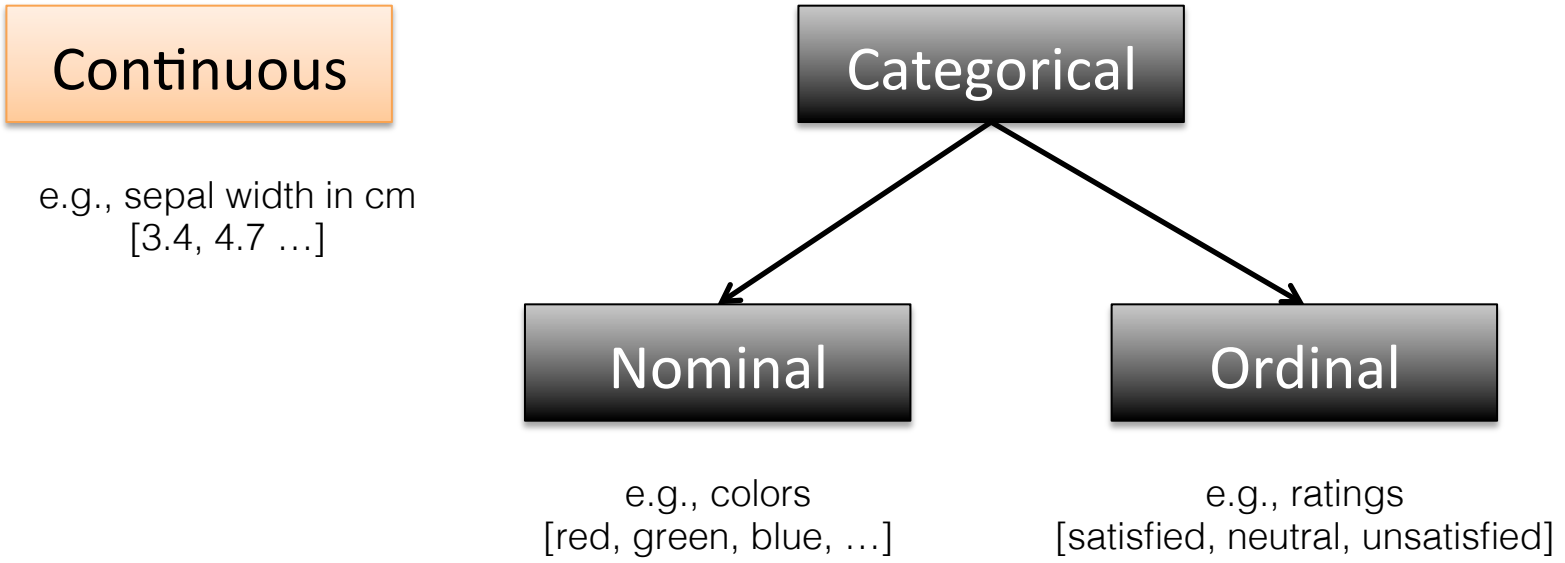
10 Preparing a real-world dataset (titanic)

11 Working with text data via the bag-of-words model

12 Application: SMS spam classification

A

Continuous & Categorical Features



Case Study - Titanic Survival

Morning Session

8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

10 Preparing a real-world dataset (titanic)

11 Working with text data via the bag-of-words model

12 Application: SMS spam classification

A

Bag of Words

- D1: ”Each state has its own laws.”
- D2: ”Every country has its own culture.”

V ={each:1, state:1, has:2, its:2, own:2,
laws: 1, every: 1, country: 1, culture: 1}

	each	state	has	its	own	laws	every	country	culture
$\mathbf{x}_{D1}$	1	1	1	1	1	1	0	0	0
$\mathbf{x}_{D2}$	0	0	1	1	1	0	1	1	1
$\sum$	1	1	2	2	2	1	1	1	1

Morning Session

8:00 AM - 12:00 PM

01 Introduction to machine learning with sample applications

02 Scientific Computing Tools for Python: NumPy, SciPy, and matplotlib

03 Data formats, preparation, and representation

04 Supervised learning: Training and test data

05 Supervised learning: Estimators for classification

06 Supervised learning: Estimators for regression analysis

07 Unsupervised learning: Unsupervised Transformers

08 Unsupervised learning: Clustering

09 The scikit-learn estimator interface

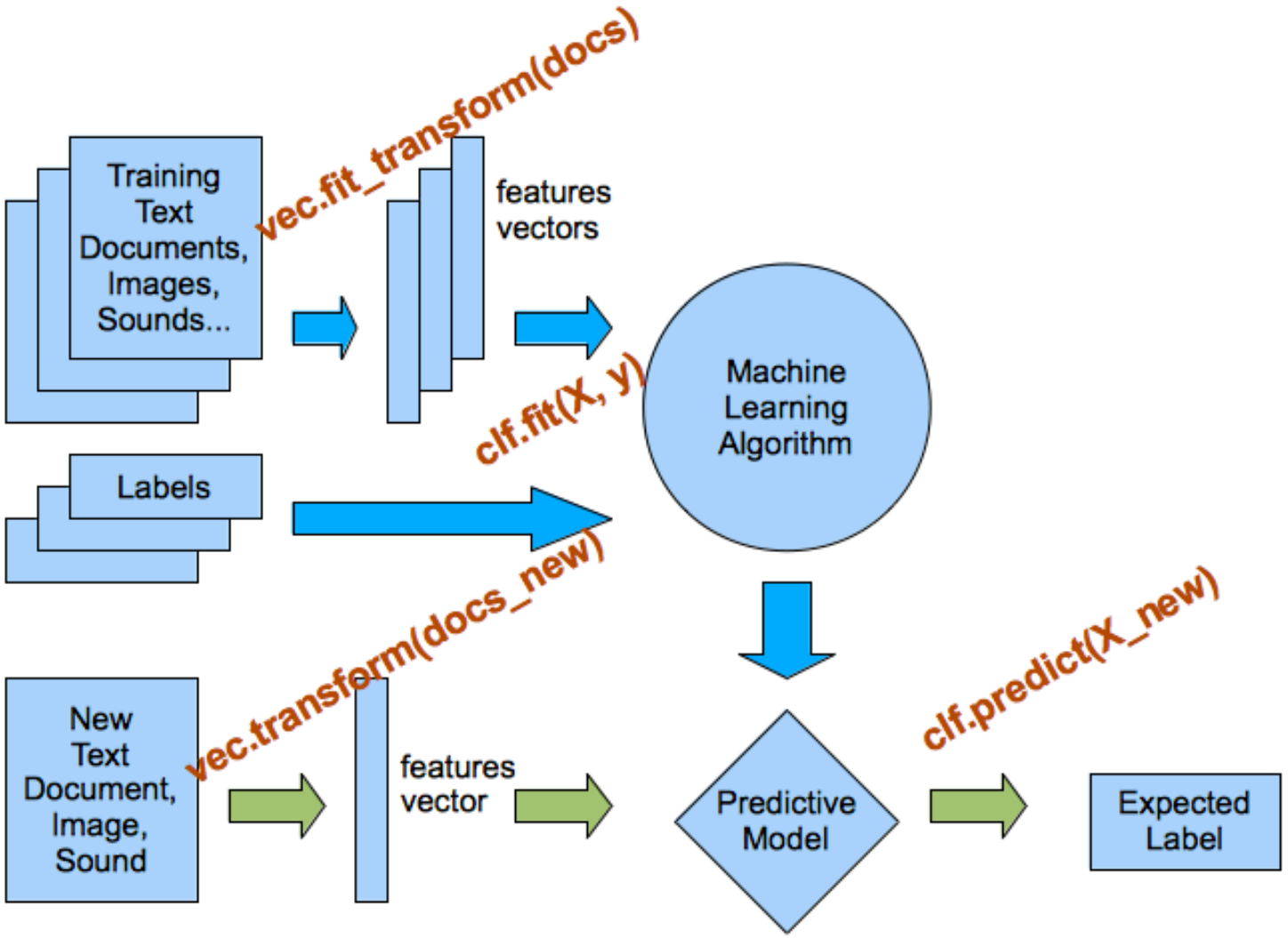
10 Preparing a real-world dataset (titanic)

11 Working with text data via the bag-of-words model

12 Application: SMS spam classification

A

Preprocessing & Classification Overview



Afternoon Session 1:30 PM - 5:30 PM

13 Cross-Validation

14 Model complexity and grid search for adjusting hyperparameters

15 Scikit-learn Pipelines

16 Supervised learning: Performance metrics for classification

17 Supervised learning: Linear Models

18 Supervised learning: Support Vector Machines

19 Supervised learning: Decision trees and random forests, ensemble methods

20 Supervised learning: Feature selection

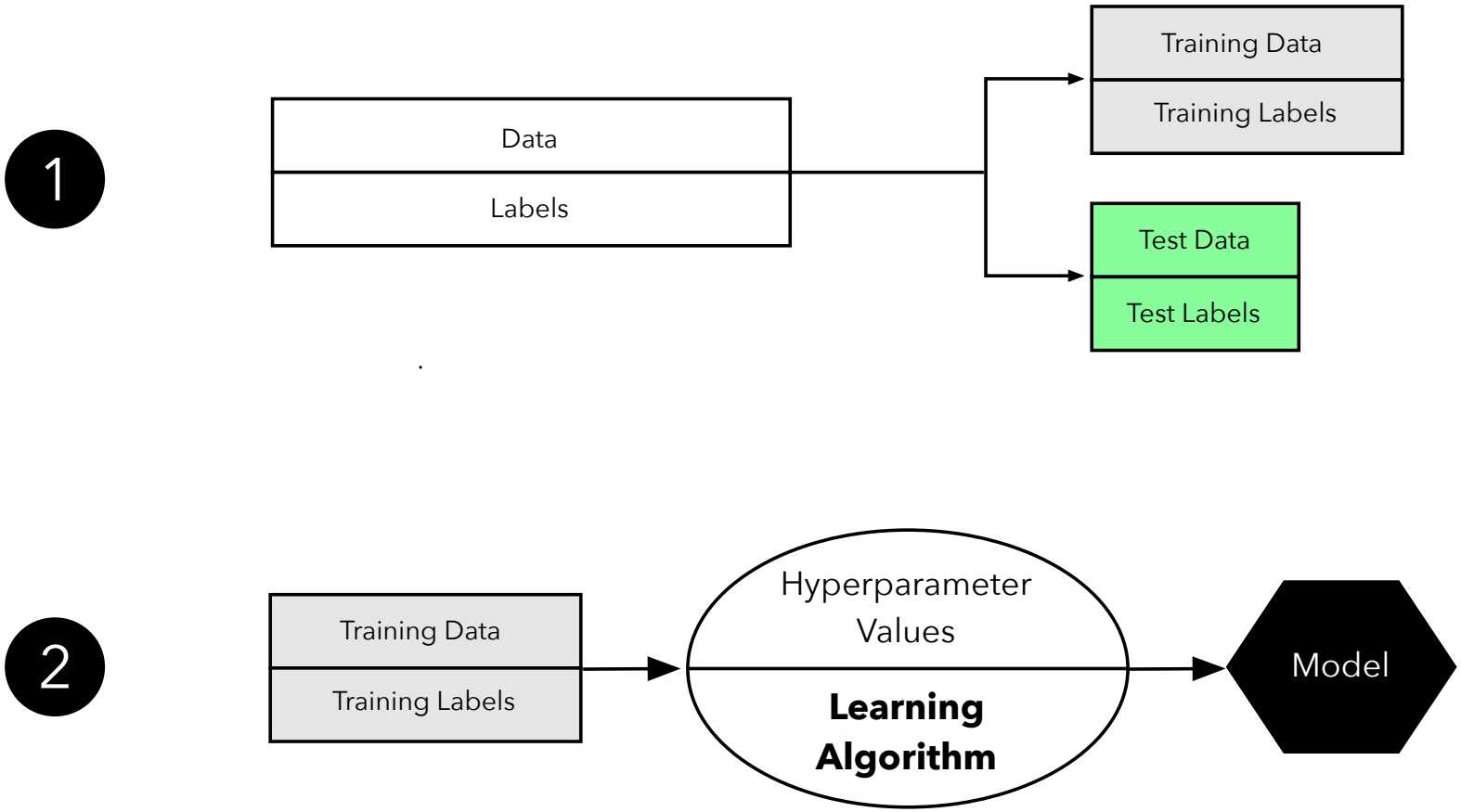
21 Unsupervised learning: Hierarchical and density-based clustering algorithms

22 Unsupervised learning: Non-linear dimensionality reduction

23 Supervised learning: Out-of-core learning

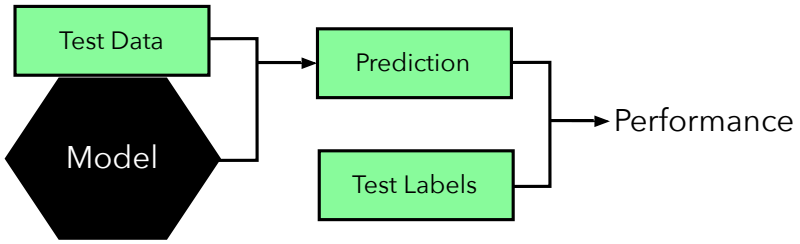
S

Holdout Evaluation I

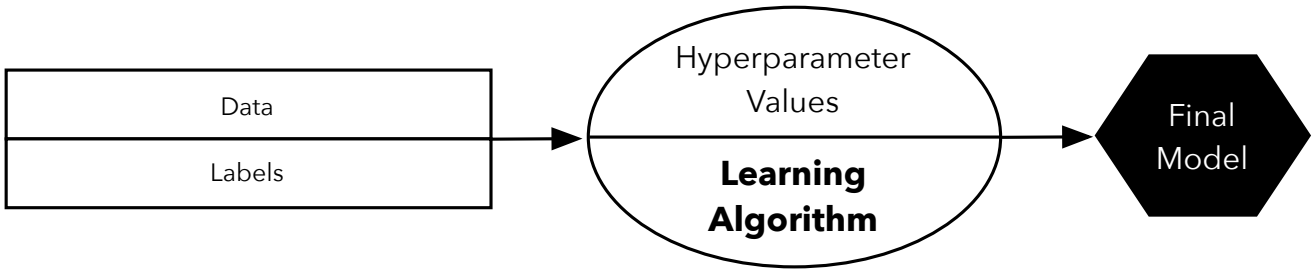


Holdout Evaluation II

3

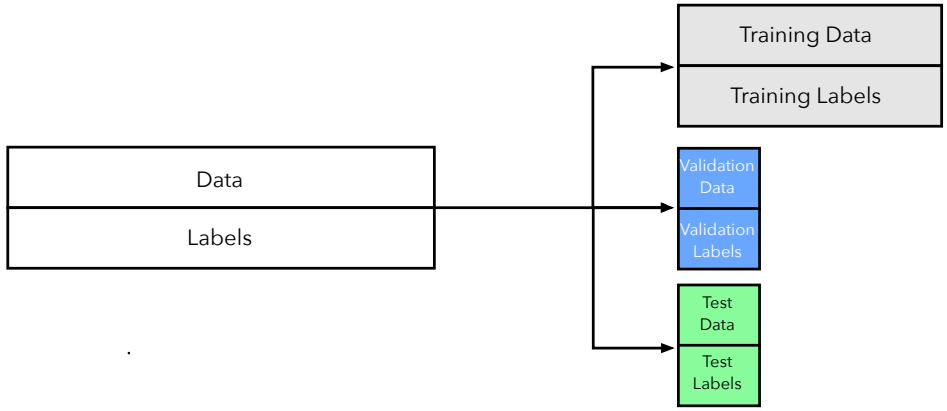


4

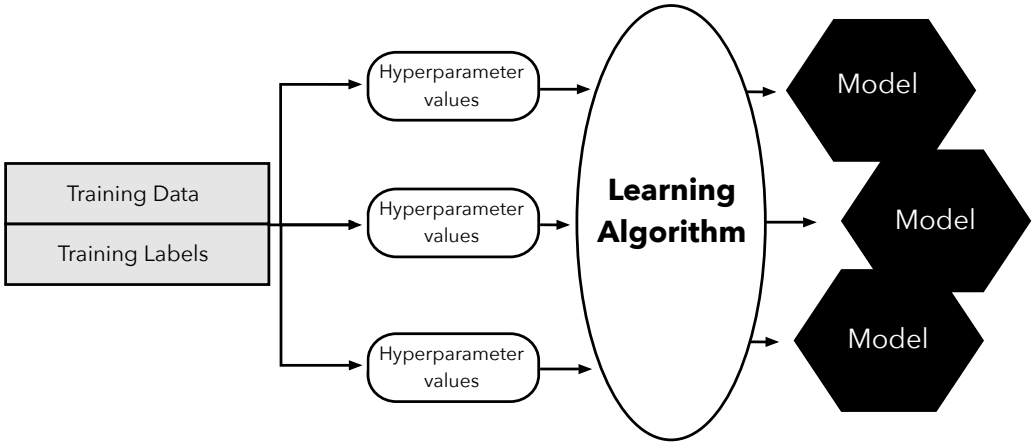


Holdout Validation I

1

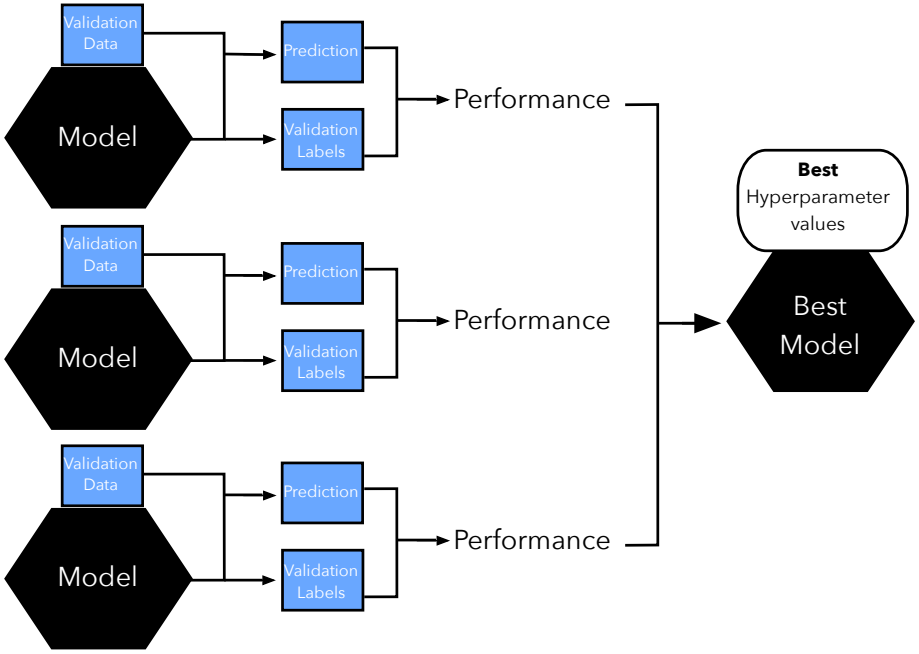


2

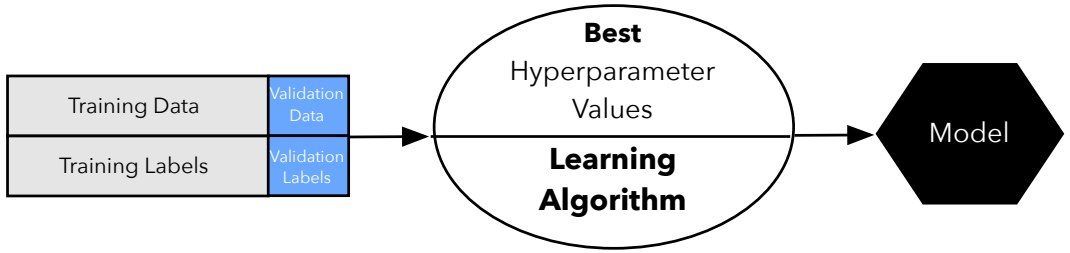


Holdout Validation II

3

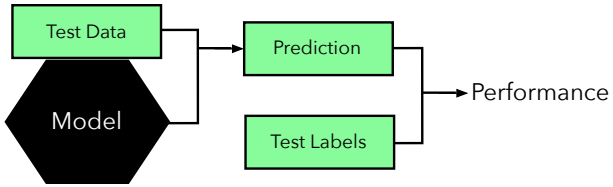


4

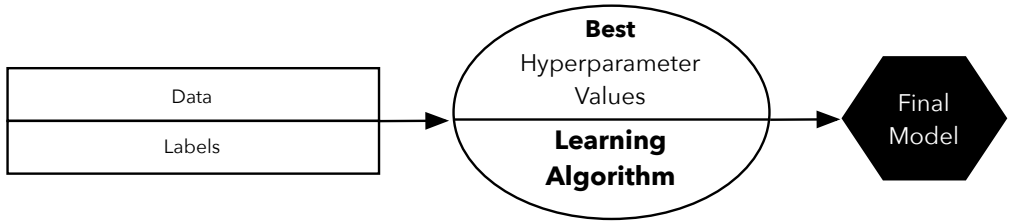


Holdout Validation III

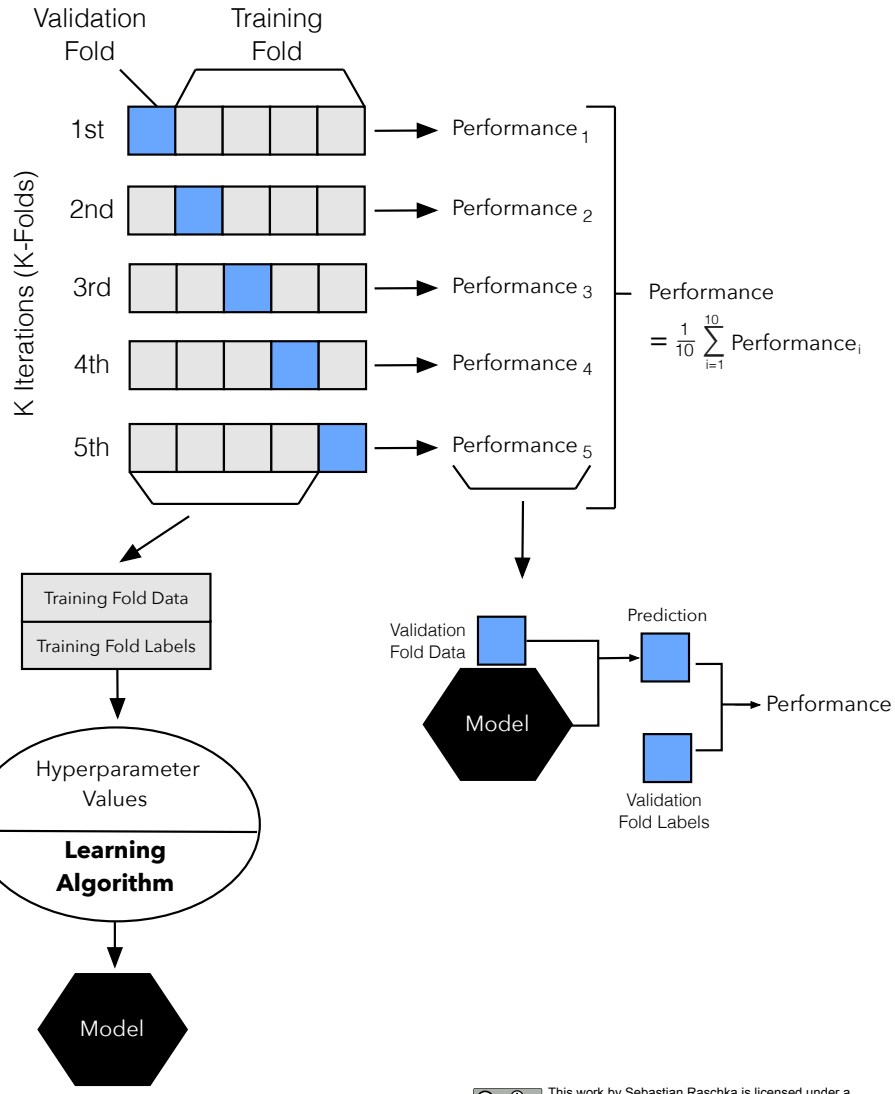
5



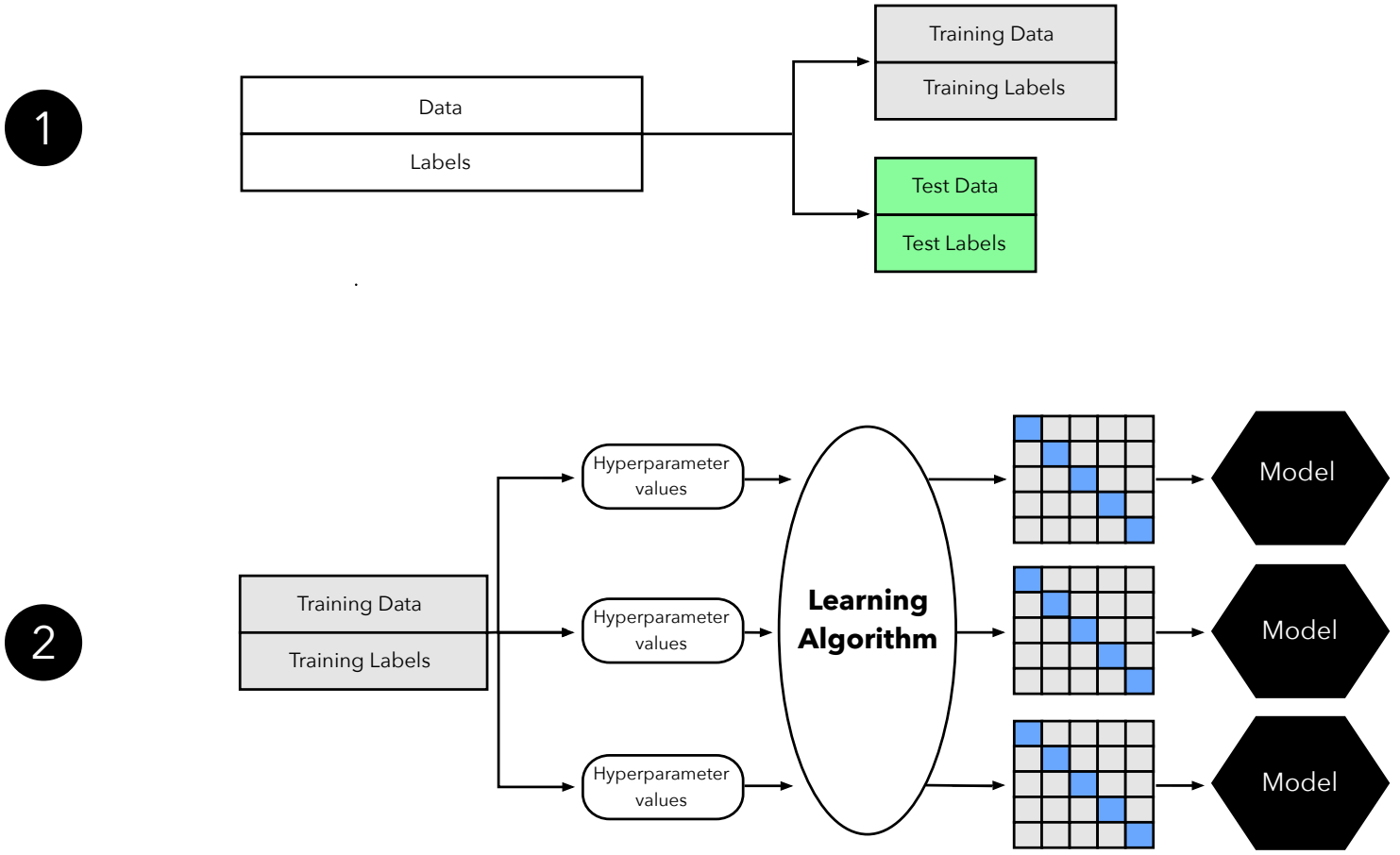
6



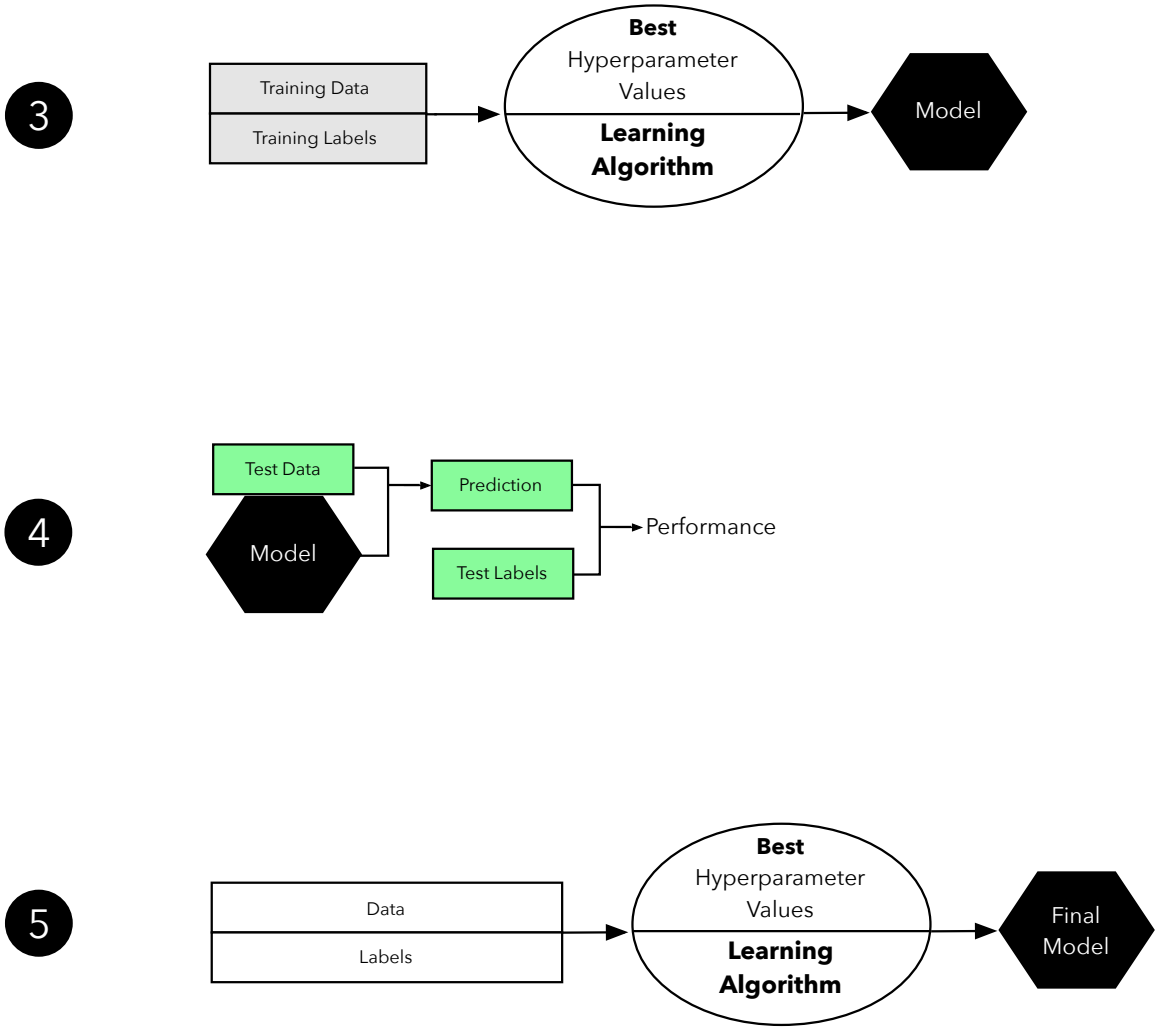
K-fold Cross-Validation



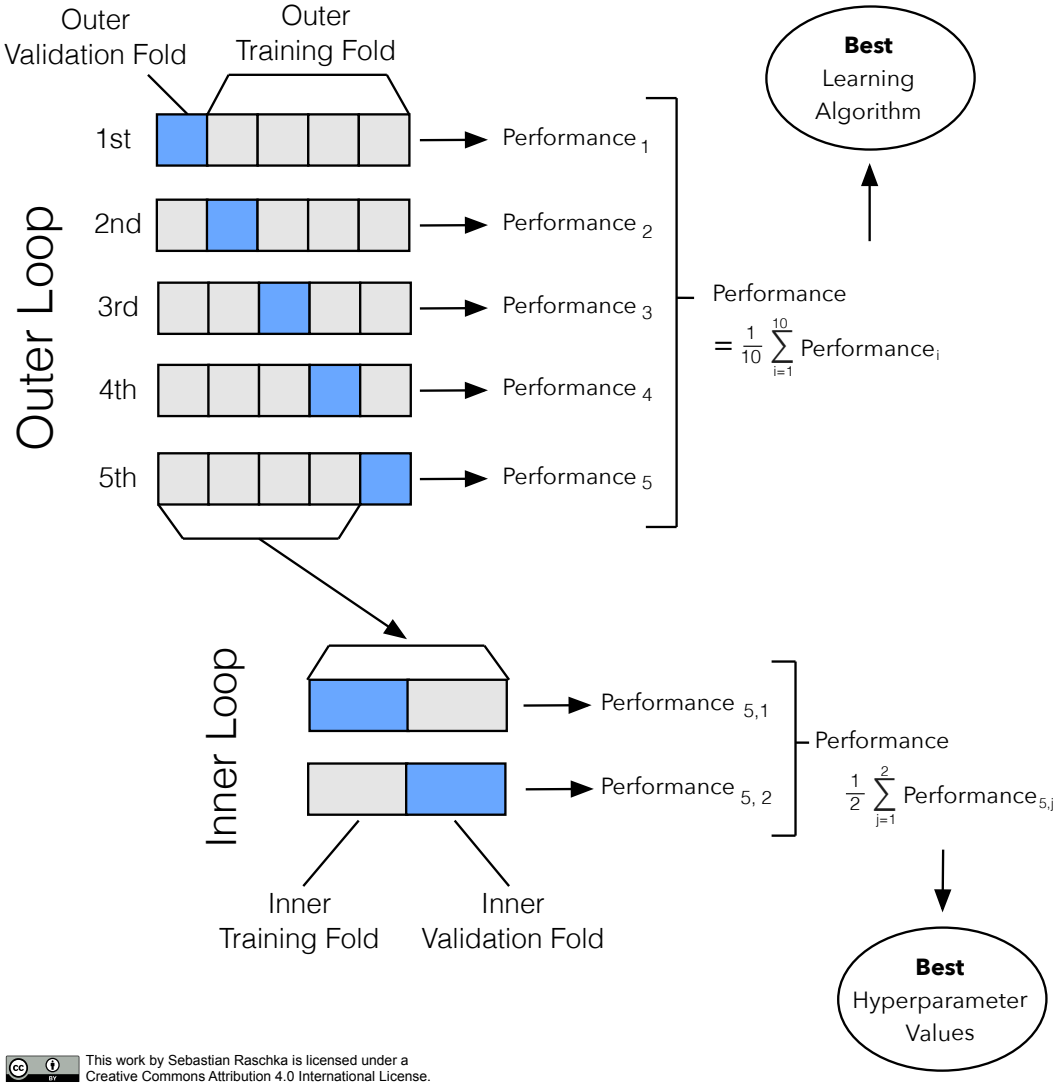
K-fold Cross-Validation Pipeline I



K-fold Cross-Validation Pipeline II



Nested CV



Afternoon Session 1:30 PM - 5:30 PM

13 Cross-Validation

14 Model complexity and grid search for adjusting hyperparameters

15 Scikit-learn Pipelines

16 Supervised learning: Performance metrics for classification

17 Supervised learning: Linear Models

18 Supervised learning: Support Vector Machines

19 Supervised learning: Decision trees and random forests, ensemble methods

20 Supervised learning: Feature selection

21 Unsupervised learning: Hierarchical and density-based clustering algorithms

22 Unsupervised learning: Non-linear dimensionality reduction

23 Supervised learning: Out-of-core learning

S

Learning Curves

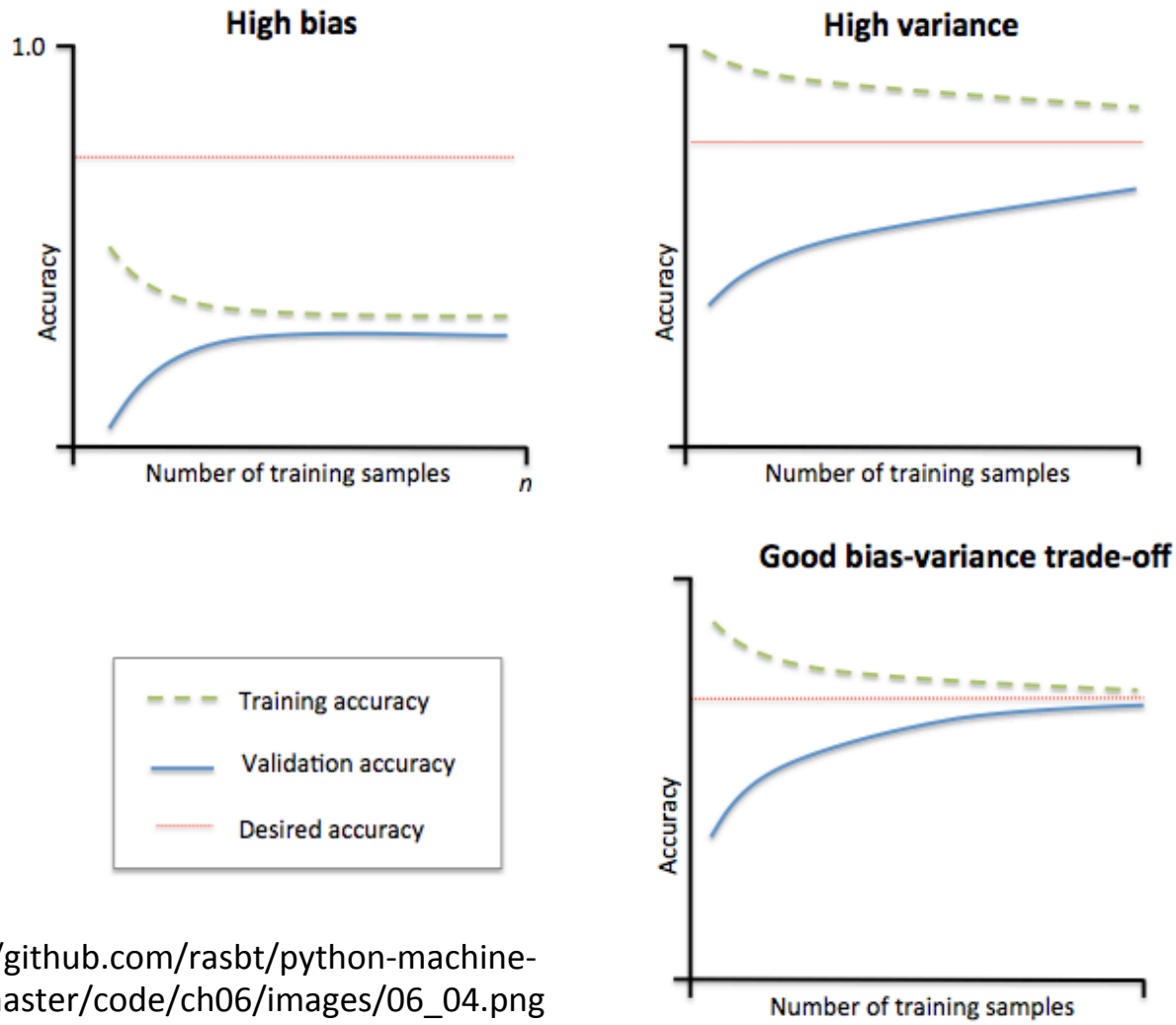
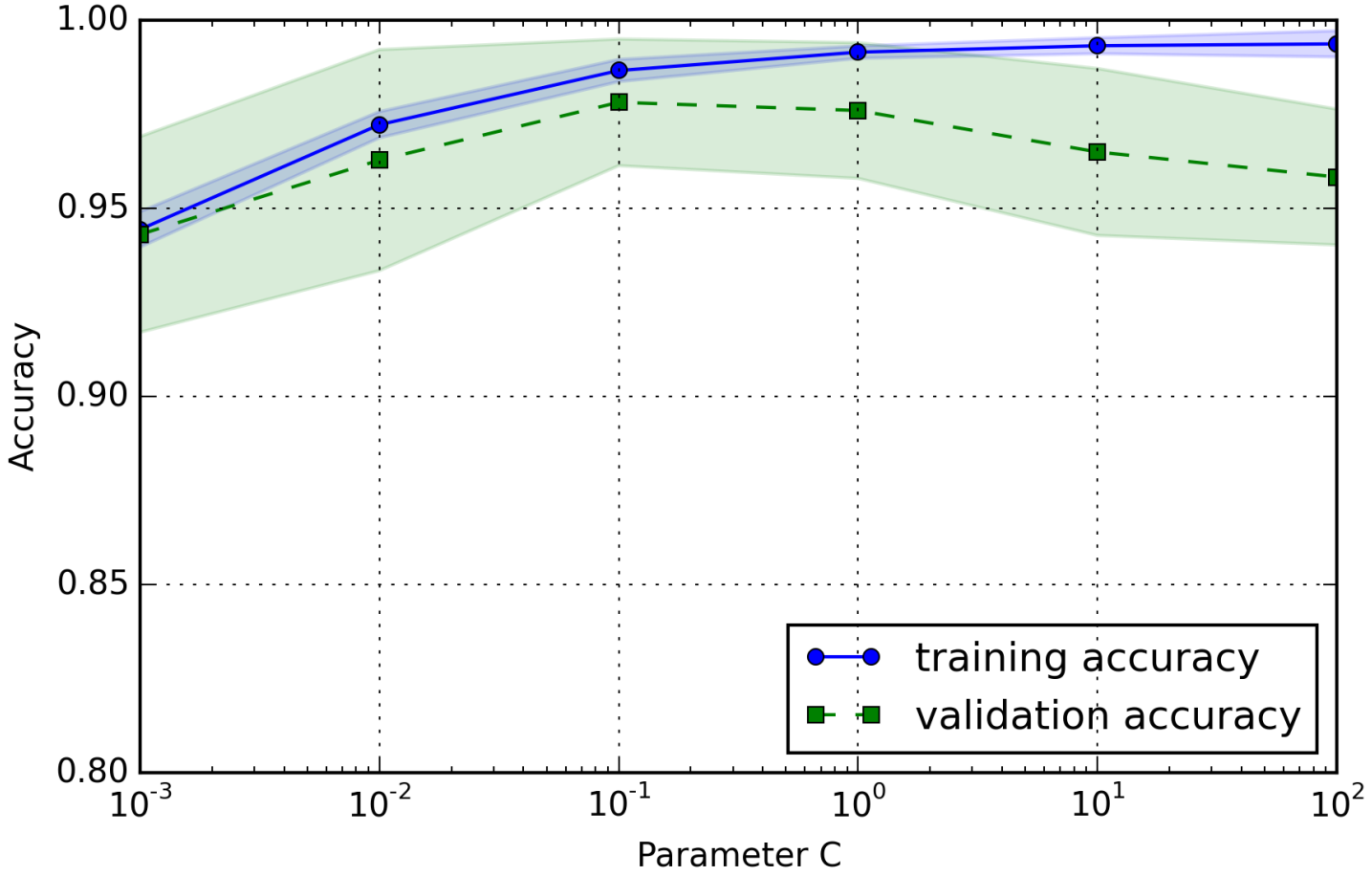
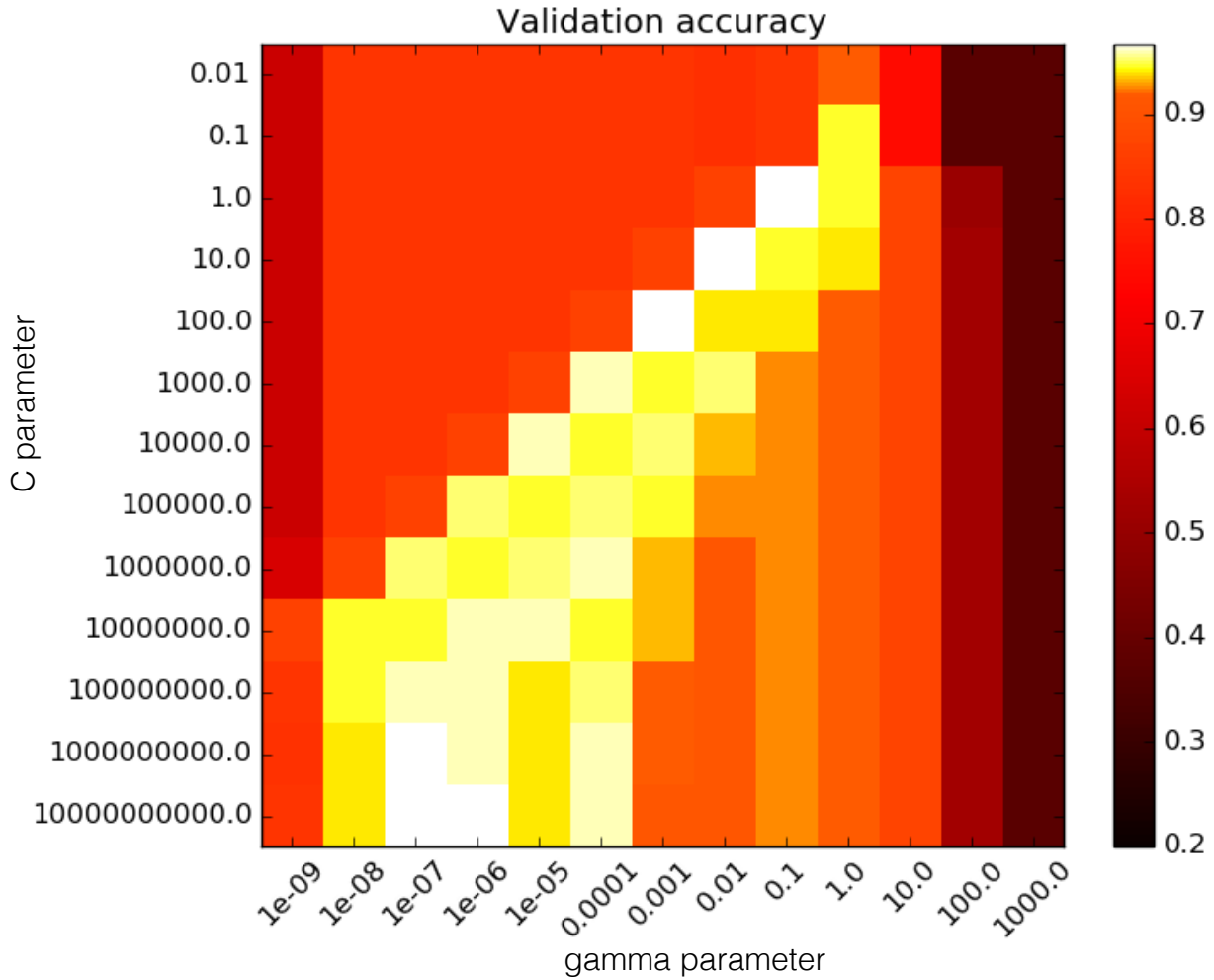


Image source: https://github.com/rasbt/python-machine-learning-book/blob/master/code/ch06/images/06_04.png

Model Complexity



Grid Search



Source: http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

Afternoon Session 1:30 PM - 5:30 PM

13 Cross-Validation

14 Model complexity and grid search for adjusting hyperparameters

15 Scikit-learn Pipelines

16 Supervised learning: Performance metrics for classification

17 Supervised learning: Linear Models

18 Supervised learning: Support Vector Machines

19 Supervised learning: Decision trees and random forests, ensemble methods

20 Supervised learning: Feature selection

21 Unsupervised learning: Hierarchical and density-based clustering algorithms

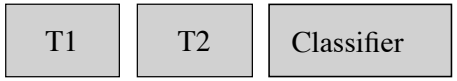
22 Unsupervised learning: Non-linear dimensionality reduction

23 Supervised learning: Out-of-core learning

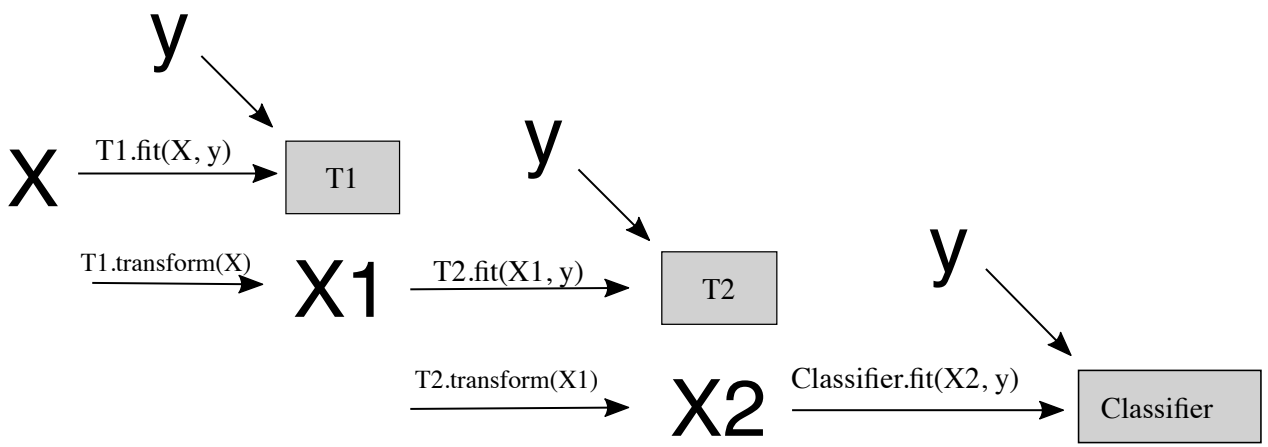
A

Pipelines

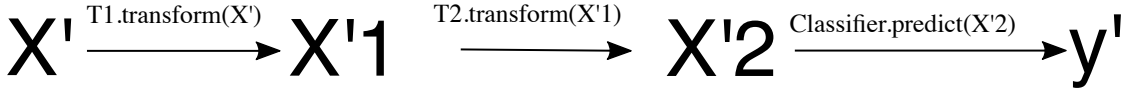
```
pipe = make_pipeline(T1(), T2(), Classifier())
```



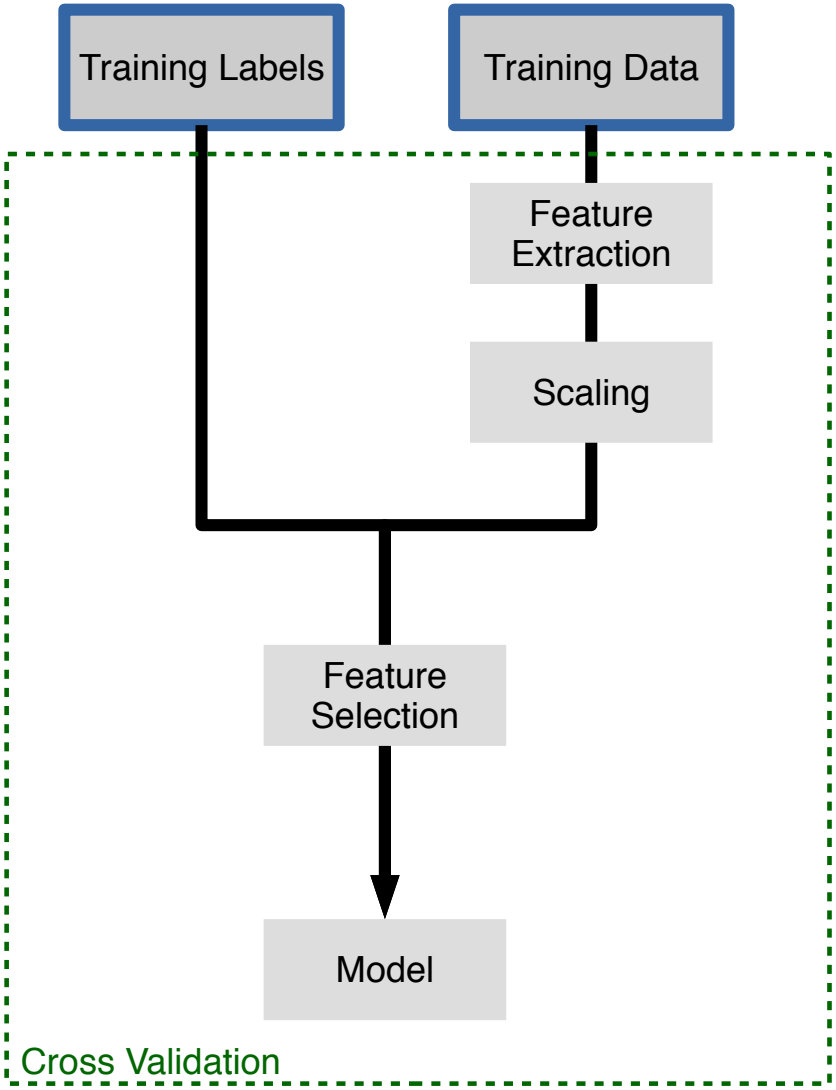

```
pipe.fit(X, y)
```




```
pipe.predict(X')
```



Pipelines & Cross Validation



Afternoon Session 1:30 PM - 5:30 PM

13 Cross-Validation

14 Model complexity and grid search for adjusting hyperparameters

15 Scikit-learn Pipelines

16 Supervised learning: Performance metrics for classification

17 Supervised learning: Linear Models

18 Supervised learning: Support Vector Machines

19 Supervised learning: Decision trees and random forests, ensemble methods

20 Supervised learning: Feature selection

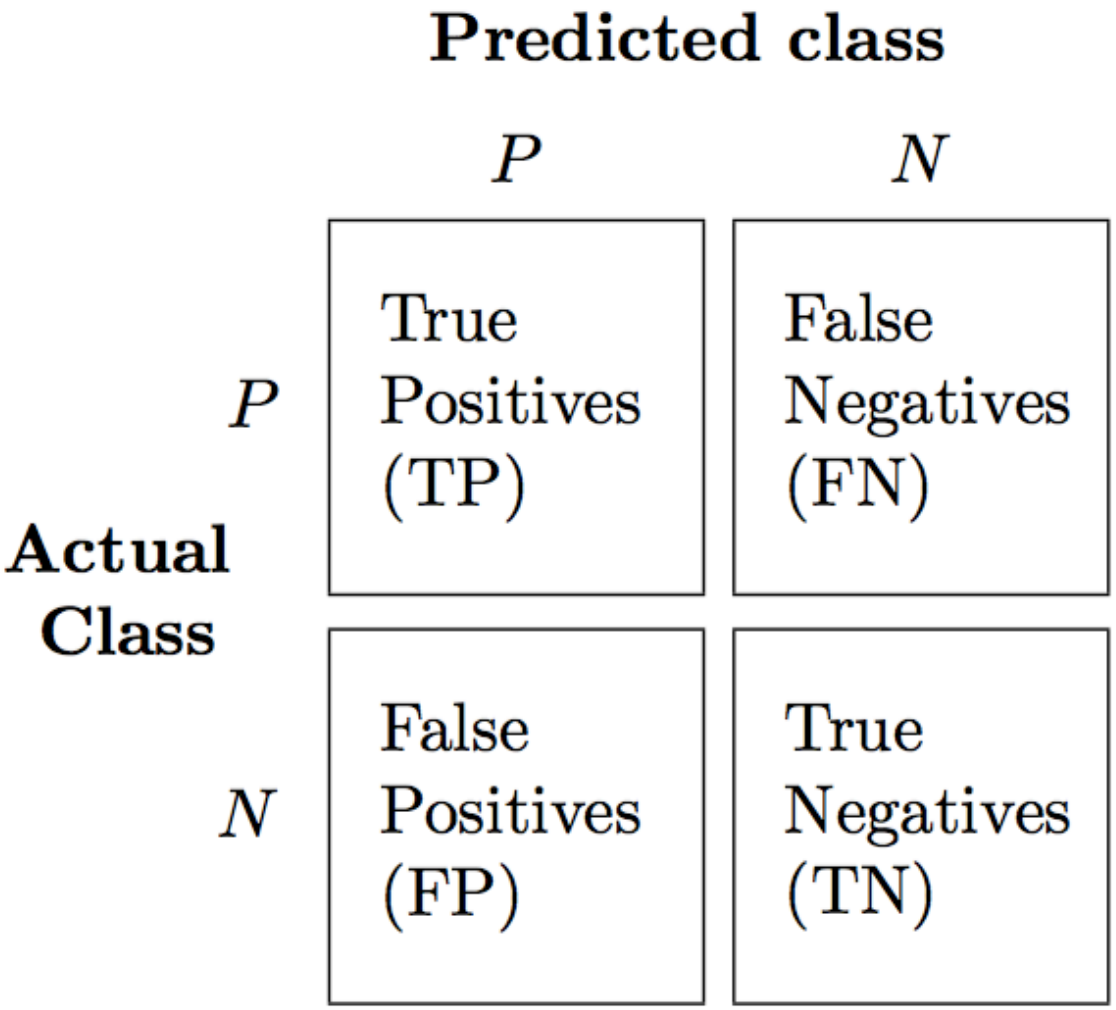
21 Unsupervised learning: Hierarchical and density-based clustering algorithms

22 Unsupervised learning: Non-linear dimensionality reduction

23 Supervised learning: Out-of-core learning

A

Confusion Matrix



Classification Metrics I

		Predicted class: P	Predicted class: N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

$$ERR = \frac{FP + FN}{FP + FN + TP + TN} = 1 - ACC$$

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} = 1 - ERR$$

Classification Metrics II

		Predicted class: P	Predicted class: N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN}$$

$$TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$$

Classification Metrics III

		Predicted class: P	Predicted class: N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

$$PRE = \frac{TP}{TP + FP}$$

$$REC = TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$$

$$F_1 = 2 \cdot \frac{PRE \cdot REC}{PRE + REC}$$

Receiver Operator Characteristic

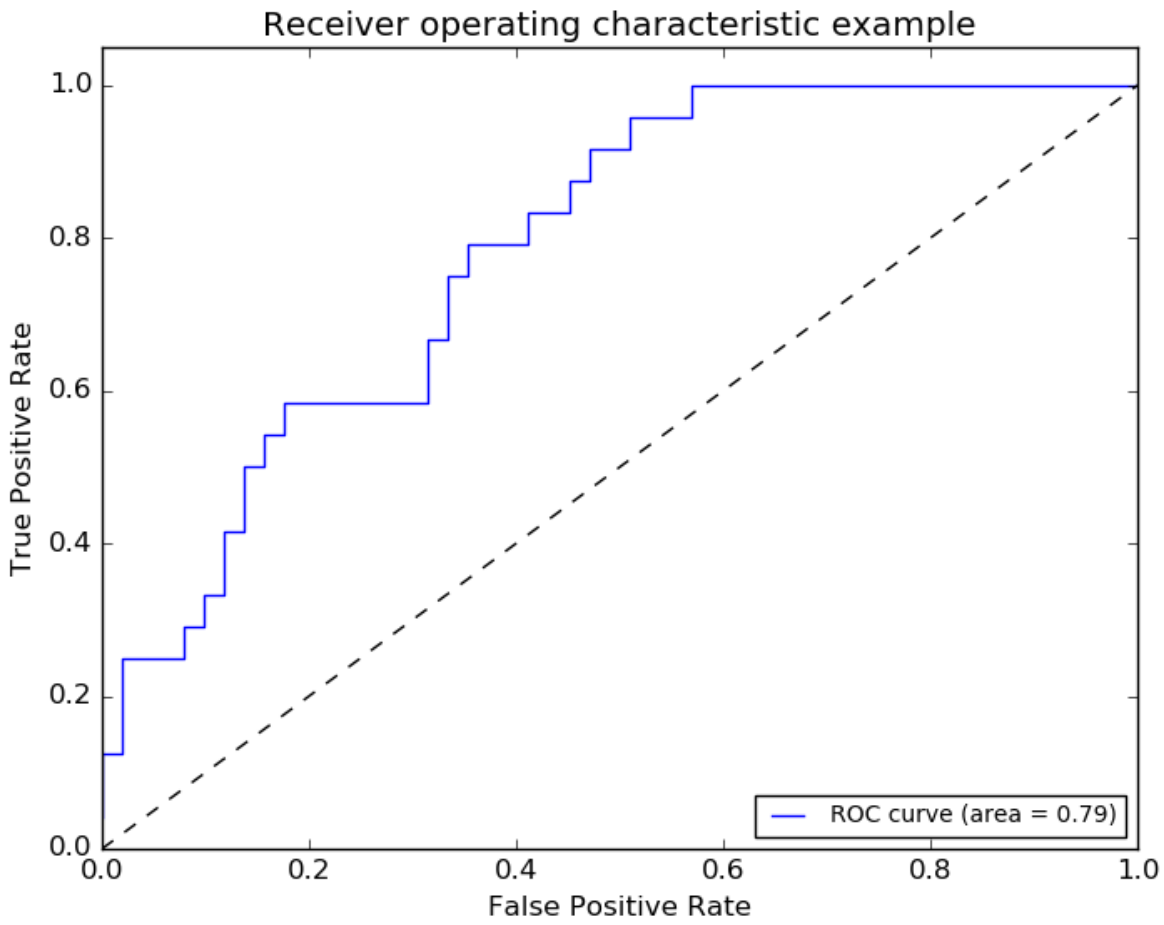


Image source: http://scikit-learn.org/stable/_images/plot_roc_001.png

Multi-Class

$$PRE_{macro} = \frac{PRE_1 + \cdots + PRE_k}{k}$$

$$PRE_{micro} = \frac{TP_1 + \cdots + TP_k}{TP_1 + \cdots + TP_k + FP_1 + \cdots + FP_k}$$

Afternoon Session 1:30 PM - 5:30 PM

13 Cross-Validation

14 Model complexity and grid search for adjusting hyperparameters

15 Scikit-learn Pipelines

16 Supervised learning: Performance metrics for classification

17 Supervised learning: Linear Models

18 Supervised learning: Support Vector Machines

19 Supervised learning: Decision trees and random forests, ensemble methods

20 Supervised learning: Feature selection

21 Unsupervised learning: Hierarchical and density-based clustering algorithms

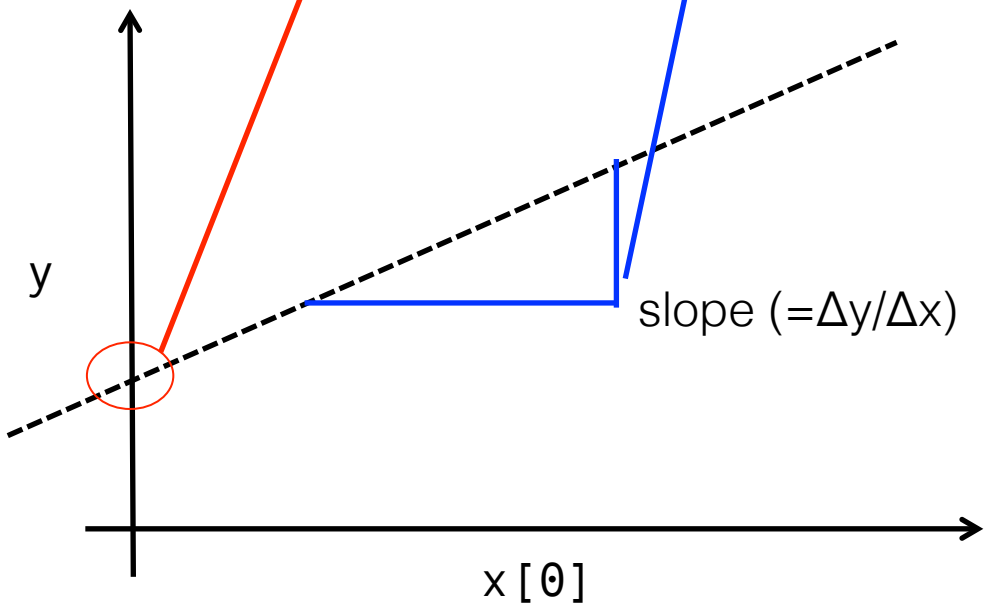
22 Unsupervised learning: Non-linear dimensionality reduction

23 Supervised learning: Out-of-core learning

A

Linear models for regression

```
y_pred =   x_test[0] * coef_[0] + ...
         + x_test[n_features-1] * coef_[n_features-1]
         + intercept_
```



Afternoon Session 1:30 PM - 5:30 PM

13 Cross-Validation

14 Model complexity and grid search for adjusting hyperparameters

15 Scikit-learn Pipelines

16 Supervised learning: Performance metrics for classification

17 Supervised learning: Linear Models

18 Supervised learning: Support Vector Machines

19 Supervised learning: Decision trees and random forests, ensemble methods

20 Supervised learning: Feature selection

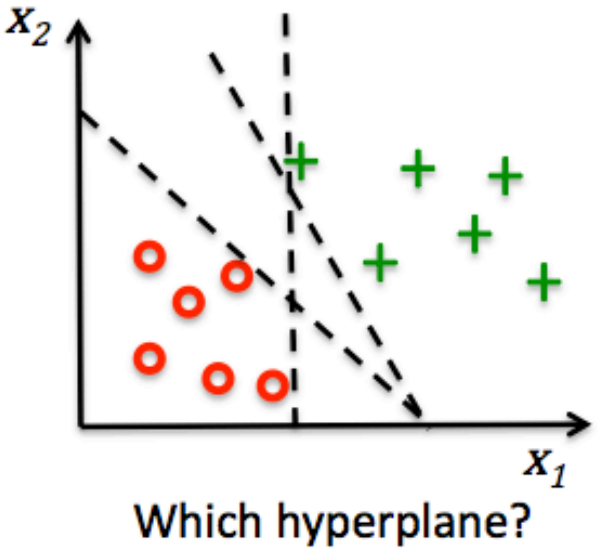
21 Unsupervised learning: Hierarchical and density-based clustering algorithms

22 Unsupervised learning: Non-linear dimensionality reduction

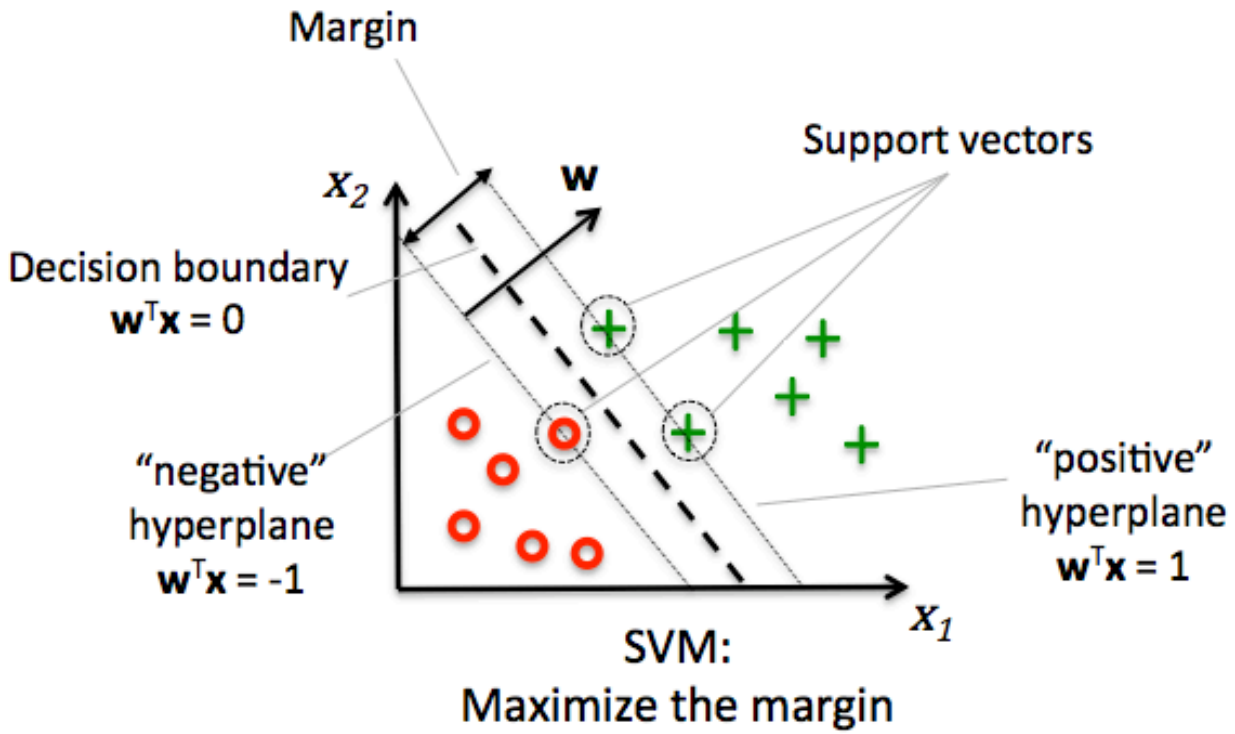
23 Supervised learning: Out-of-core learning

A

Support Vector Machines



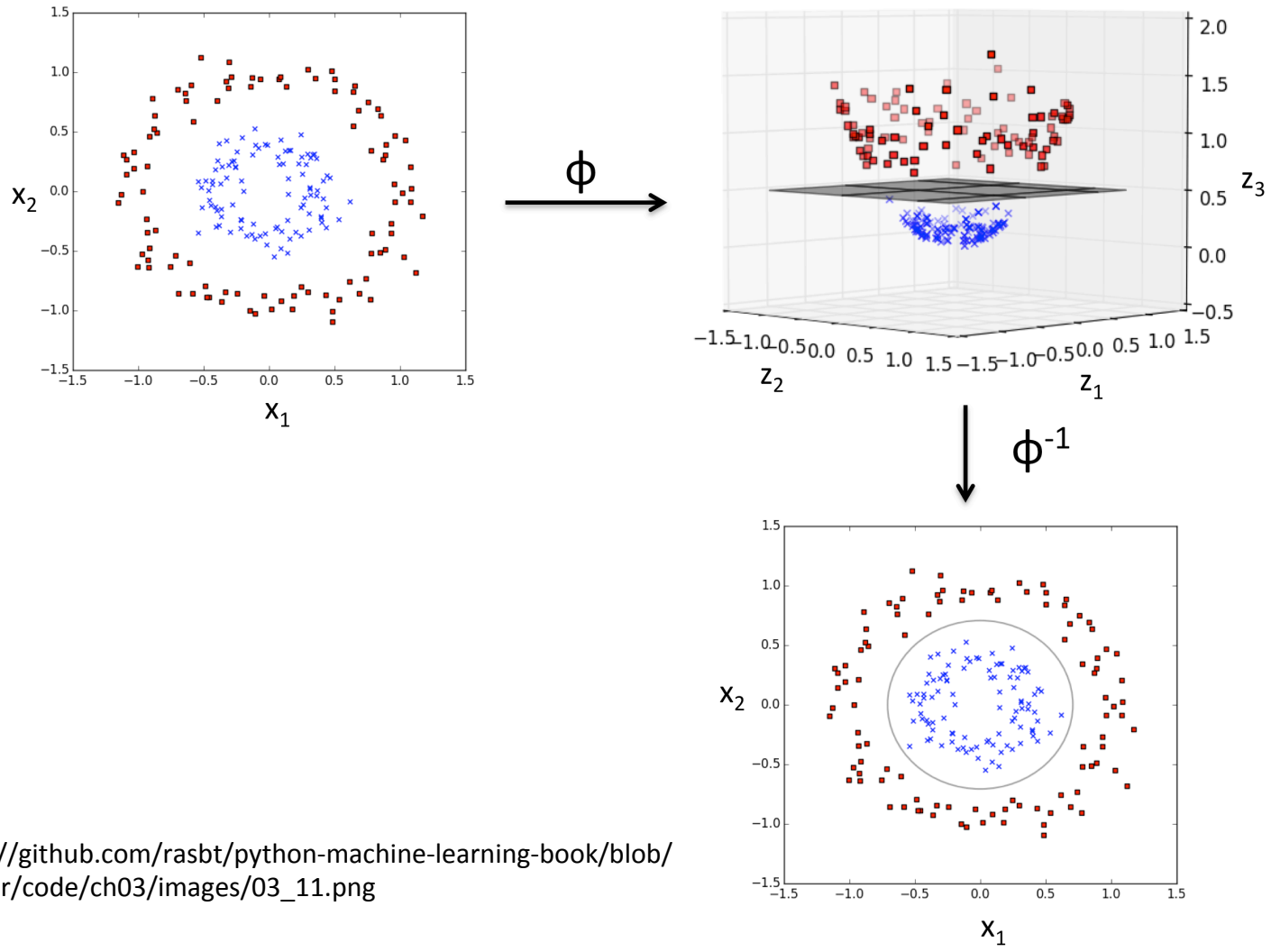
Which hyperplane?



SVM:
Maximize the margin

Image source: https://github.com/rasbt/python-machine-learning-book/blob/master/code/ch03/images/03_07.png

Kernel Trick



https://github.com/rasbt/python-machine-learning-book/blob/master/code/ch03/images/03_11.png

Afternoon Session 1:30 PM - 5:30 PM

13 Cross-Validation

14 Model complexity and grid search for adjusting hyperparameters

15 Scikit-learn Pipelines

16 Supervised learning: Performance metrics for classification

17 Supervised learning: Linear Models

18 Supervised learning: Support Vector Machines

19 Supervised learning: Decision trees and random forests, ensemble methods

20 Supervised learning: Feature selection

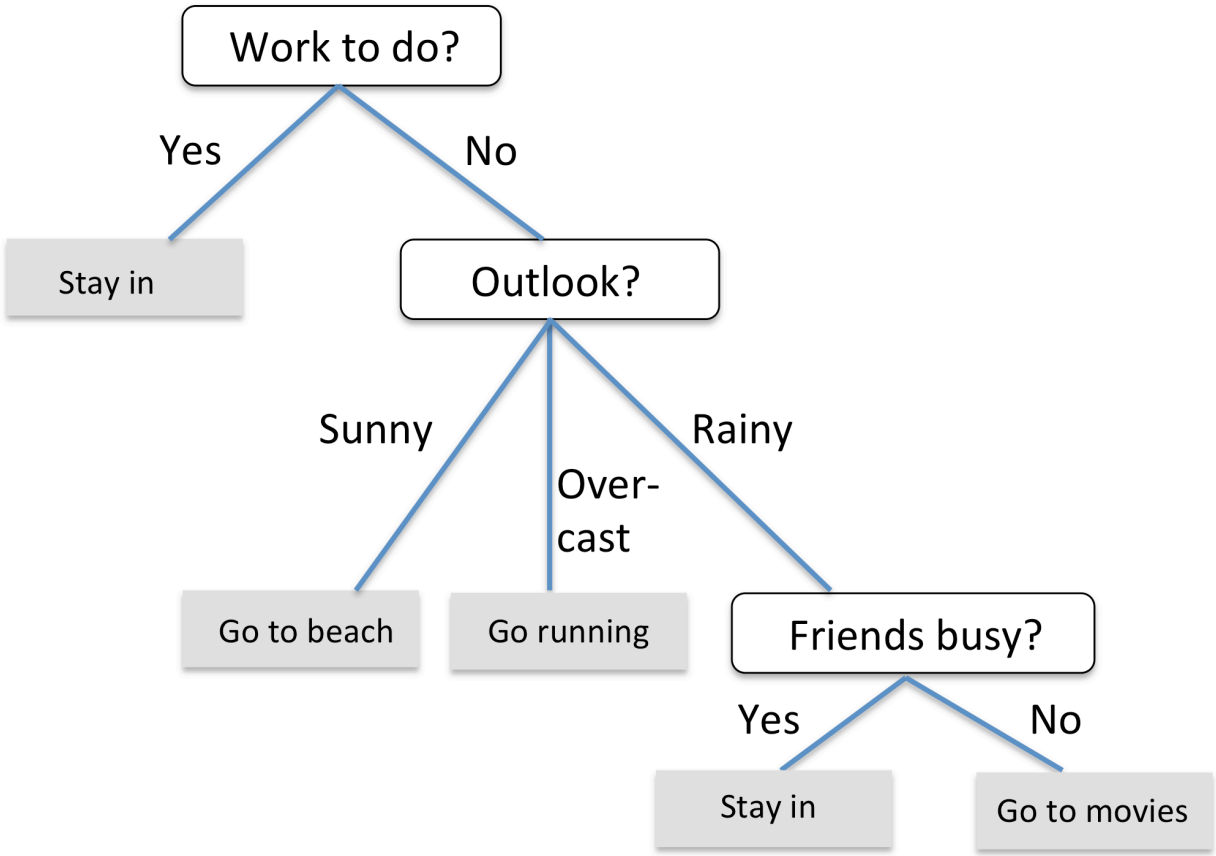
21 Unsupervised learning: Hierarchical and density-based clustering algorithms

22 Unsupervised learning: Non-linear dimensionality reduction

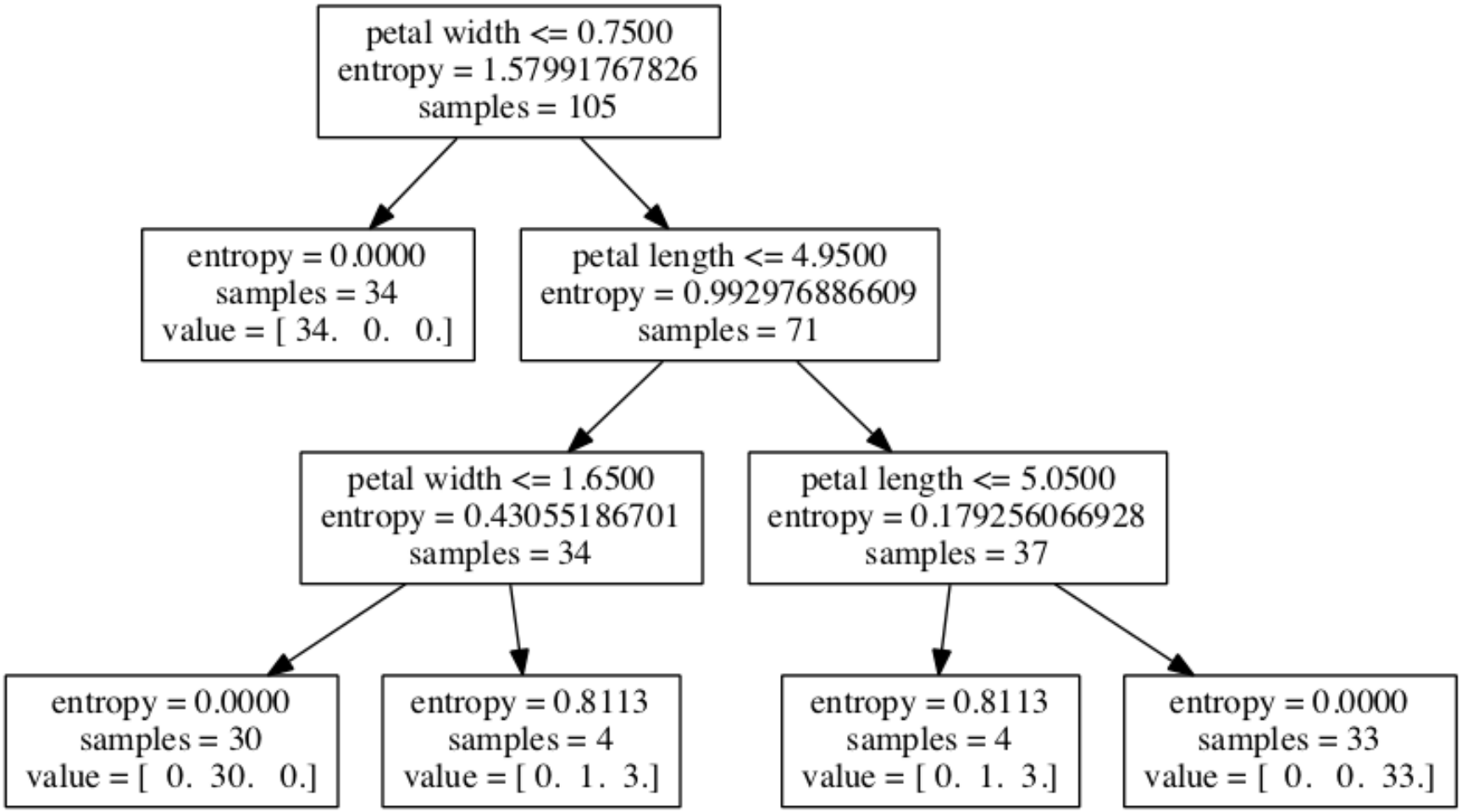
23 Supervised learning: Out-of-core learning

S

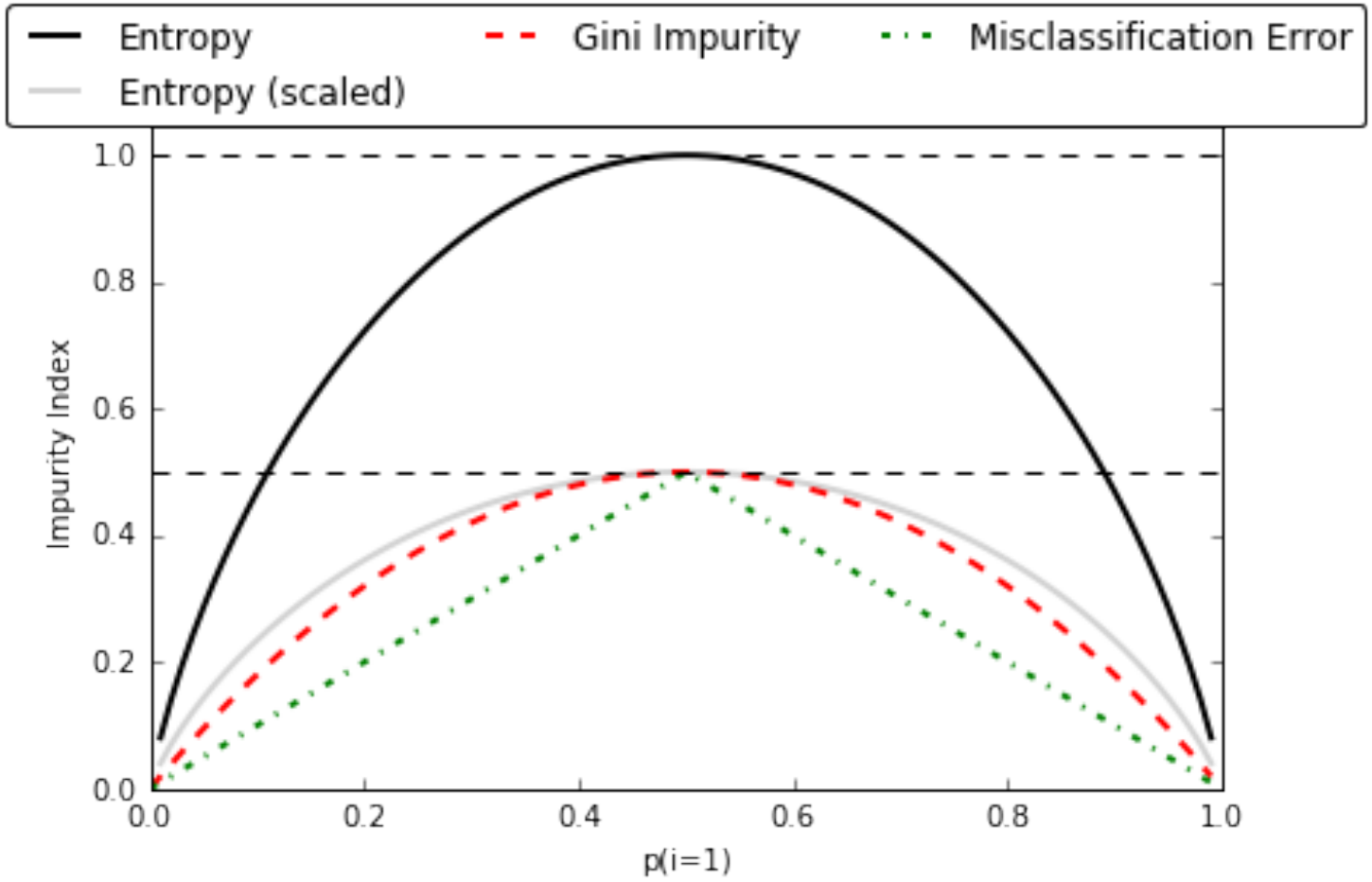
Decision Trees



Classification w. Continuous Features



Impurity measures



Ensemble Methods

Afternoon Session 1:30 PM - 5:30 PM

13 Cross-Validation

14 Model complexity and grid search for adjusting hyperparameters

15 Scikit-learn Pipelines

16 Supervised learning: Performance metrics for classification

17 Supervised learning: Linear Models

18 Supervised learning: Support Vector Machines

19 Supervised learning: Decision trees and random forests, ensemble methods

20 Supervised learning: Feature selection

21 Unsupervised learning: Hierarchical and density-based clustering algorithms

22 Unsupervised learning: Non-linear dimensionality reduction

23 Supervised learning: Out-of-core learning

S

Dimensionality Reduction

Afternoon Session 1:30 PM - 5:30 PM

13 Cross-Validation

14 Model complexity and grid search for adjusting hyperparameters

15 Scikit-learn Pipelines

16 Supervised learning: Performance metrics for classification

17 Supervised learning: Linear Models

18 Supervised learning: Support Vector Machines

19 Supervised learning: Decision trees and random forests, ensemble methods

20 Supervised learning: Feature selection

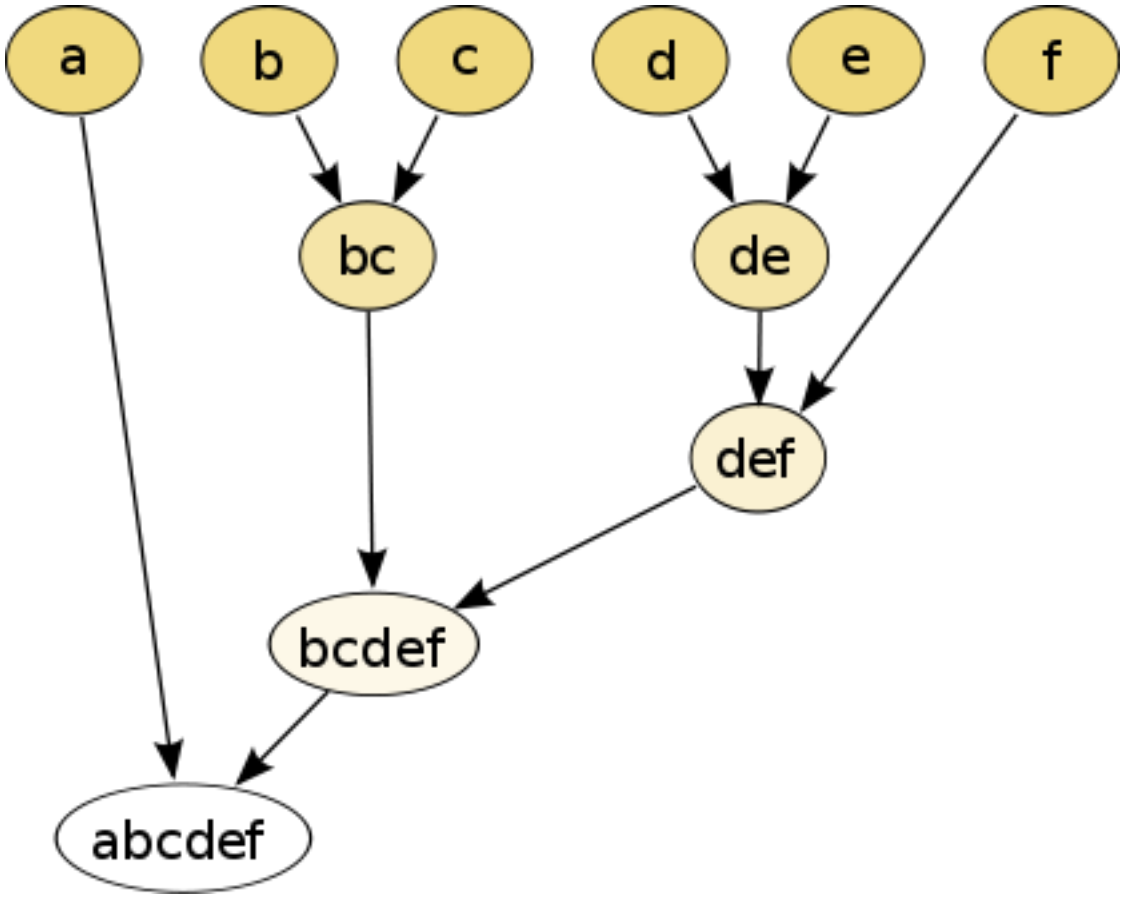
21 Unsupervised learning: Hierarchical and density-based clustering algorithms

22 Unsupervised learning: Non-linear dimensionality reduction

23 Supervised learning: Out-of-core learning

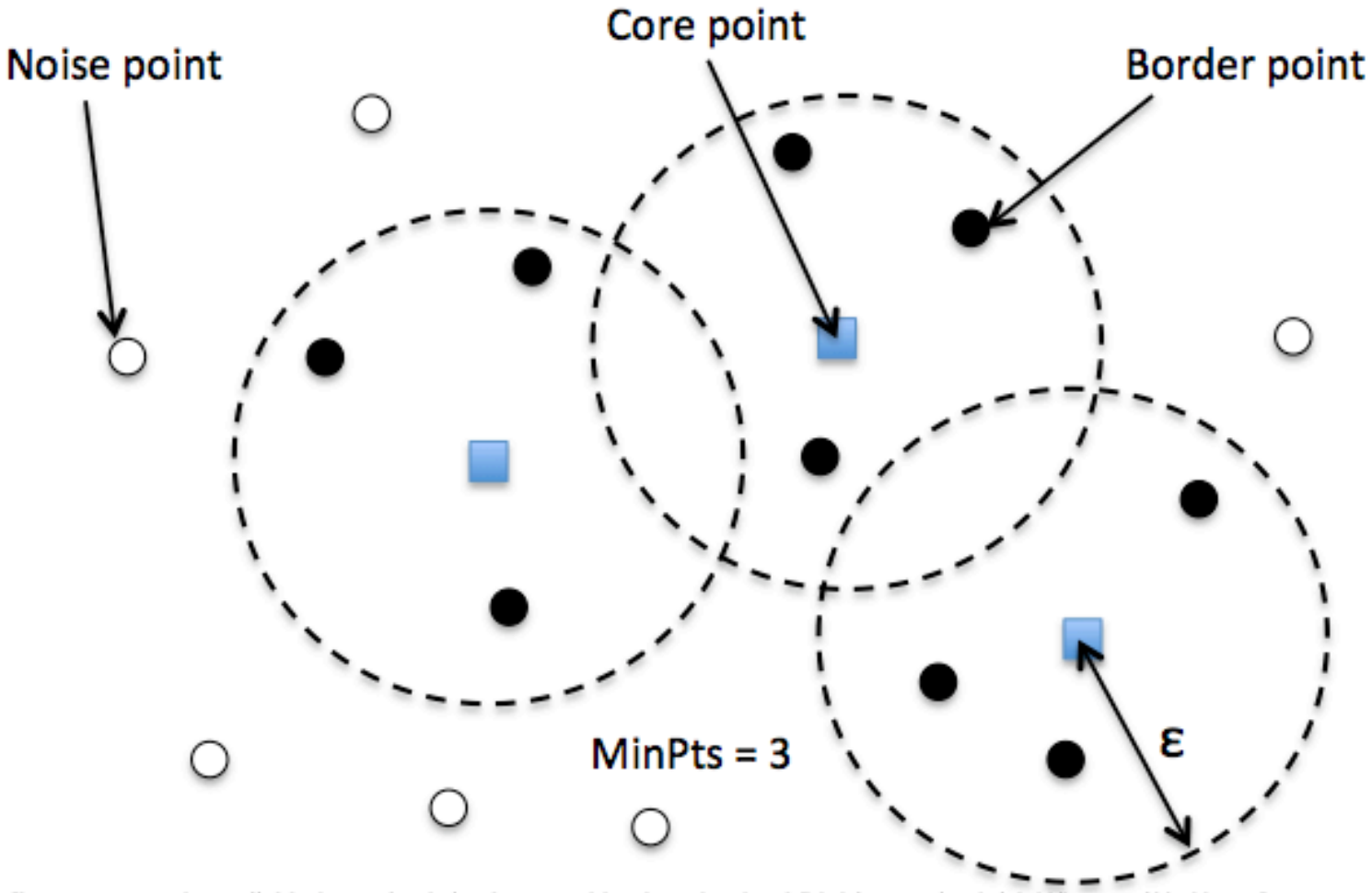
S

Hierarchical Clustering



https://en.wikipedia.org/wiki/Hierarchical_clustering#/media/File:Hierarchical_clustering_simple_diagram.svg [CC BY-SA 3.0]

DBSCAN



[Image source: https://github.com/rasbt/python-machine-learning-book/blob/master/code/ch11/images/11_11.png]

Afternoon Session 1:30 PM - 5:30 PM

13 Cross-Validation

14 Model complexity and grid search for adjusting hyperparameters

15 Scikit-learn Pipelines

16 Supervised learning: Performance metrics for classification

17 Supervised learning: Linear Models

18 Supervised learning: Support Vector Machines

19 Supervised learning: Decision trees and random forests, ensemble methods

20 Supervised learning: Feature selection

21 Unsupervised learning: Hierarchical and density-based clustering algorithms

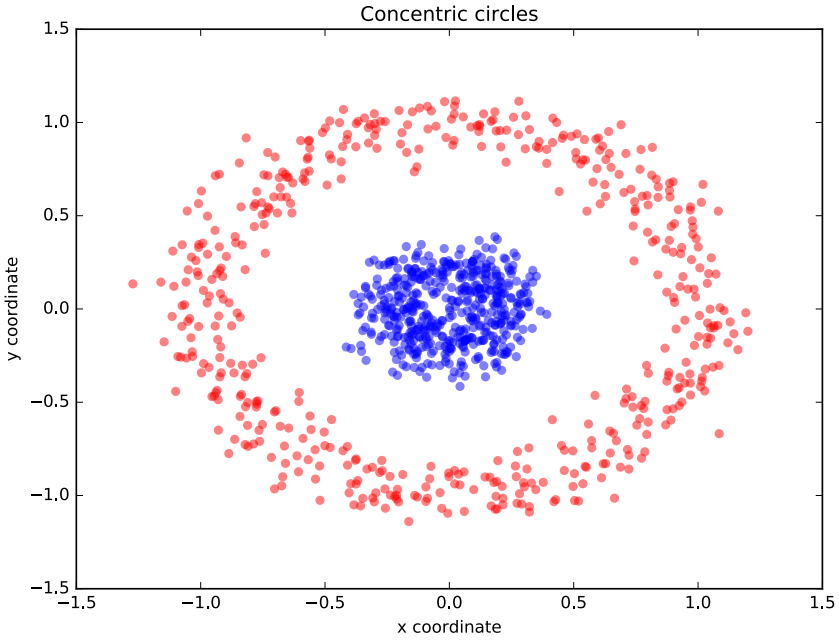
22 Unsupervised learning: Non-linear dimensionality reduction

23 Supervised learning: Out-of-core learning

S

PCA

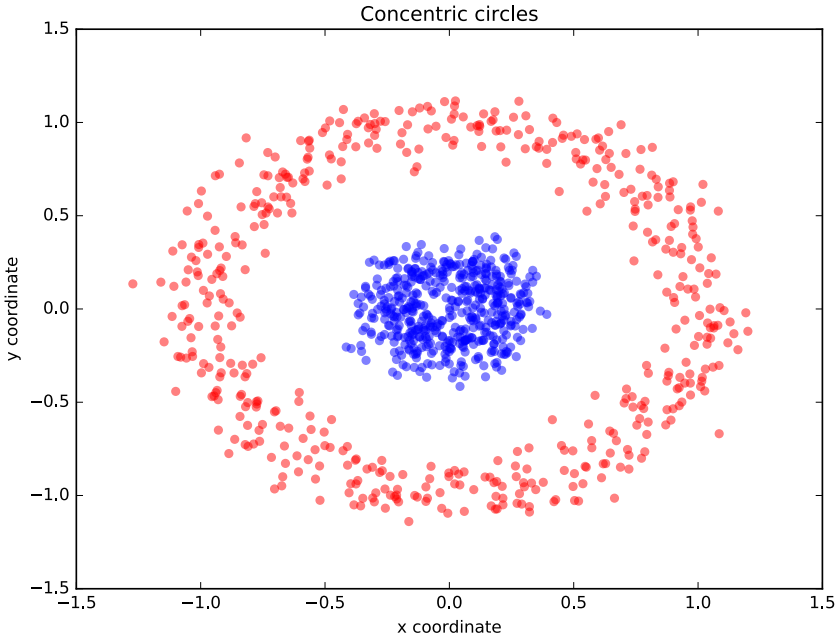
2D ⟶ 1D

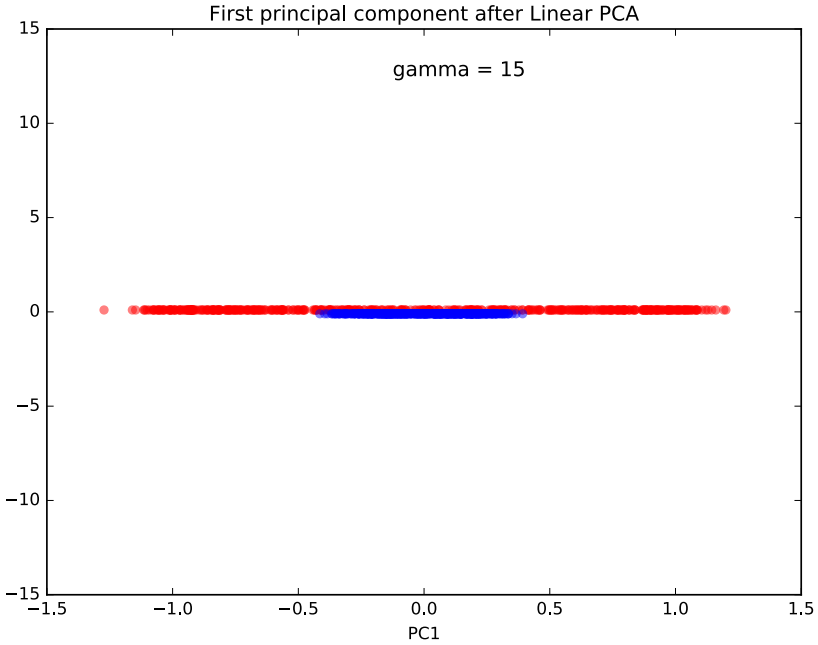

Concentric circles
y coordinate
x coordinate
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5

?

PCA

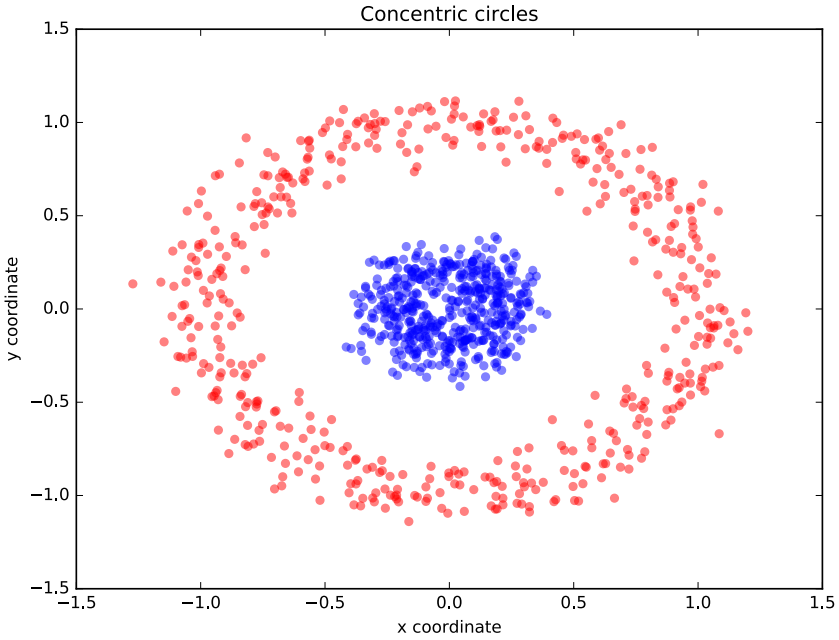
2D ⟶ 1D

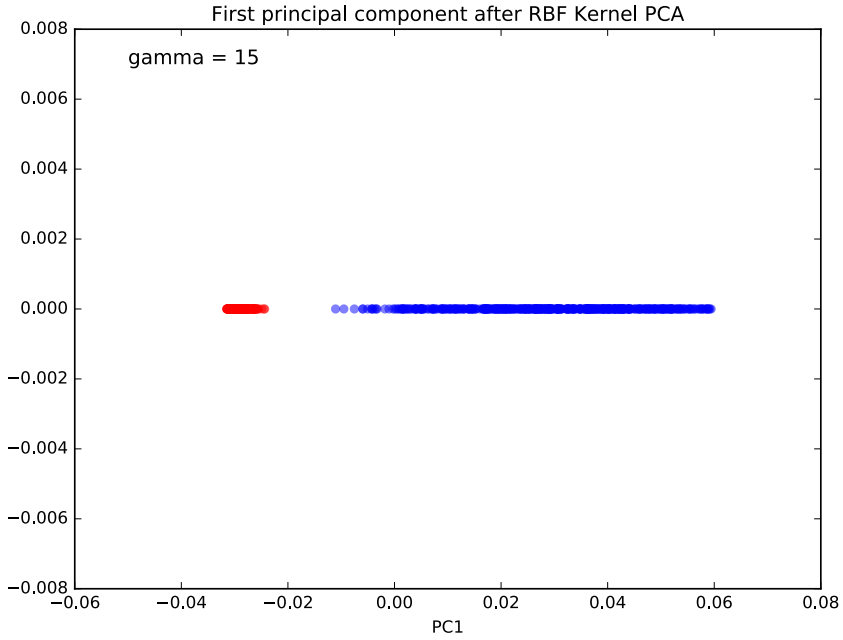

Concentric circles
y coordinate
x coordinate
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5


First principal component after Linear PCA
gamma = 15
PC1
15
10
5
0
−5
−10
−15
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

Kernel PCA

2D ⟶ 1D





Afternoon Session 1:30 PM - 5:30 PM

13 Cross-Validation

14 Model complexity and grid search for adjusting hyperparameters

15 Scikit-learn Pipelines

16 Supervised learning: Performance metrics for classification

17 Supervised learning: Linear Models

18 Supervised learning: Support Vector Machines

19 Supervised learning: Decision trees and random forests, ensemble methods

20 Supervised learning: Feature selection

21 Unsupervised learning: Hierarchical and density-based clustering algorithms

22 Unsupervised learning: Non-linear dimensionality reduction

23 Supervised learning: Out-of-core learning

S