PyData
Chicago 2016

Learning scikit learn

– An Introduction to Machine Learning in Python

PyData Chicago 2016
Chicago, The University of Illinois • August 26, 2016

Sebastian Raschka

Links & Info

Tutorial Material on GitHub:

https://github.com/rasbt/pydata-chicago2016-ml-tutorial

Contact:

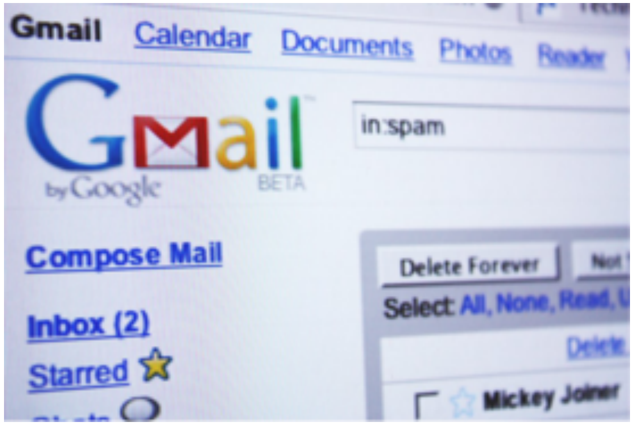
- E-mail: mail@sebastianraschka.com
- Website: http://sebastianraschka.com
- Twitter: @rasbt
- GitHub: rasbt

Let’s not stress!

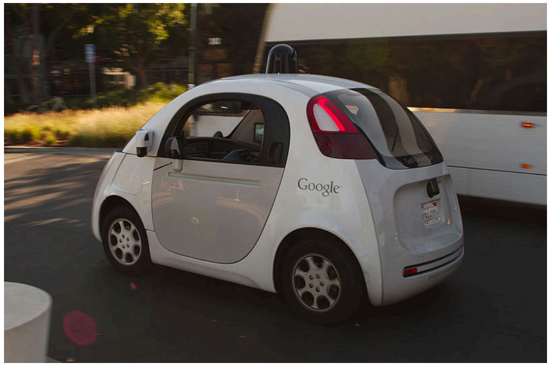
This is an introductory tutorial, and we are here to learn!

Please ask questions!

What can machine learning do for us?



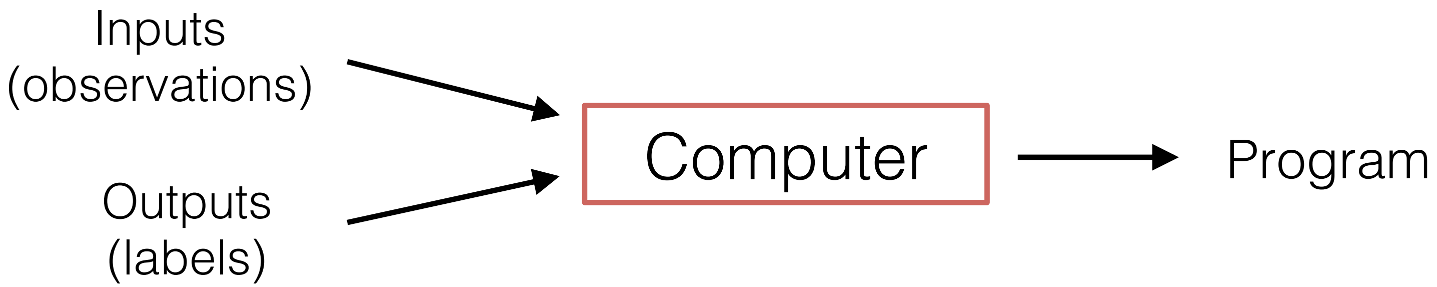
https://flic.kr/p/5BLW6G [CC BY 2.0]



https://commons.wikimedia.org/wiki/
File:Google_self_driving_car_at_the_Googleplex.jpg

Photo by Michasel Shick, CC BY-SA 4.0 lit

What is machine learning?

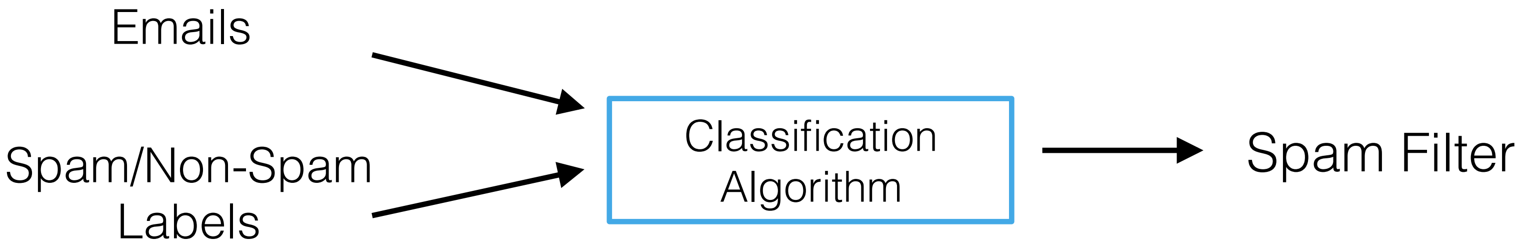


Inputs (observations)

Outputs (labels)

Computer

Program



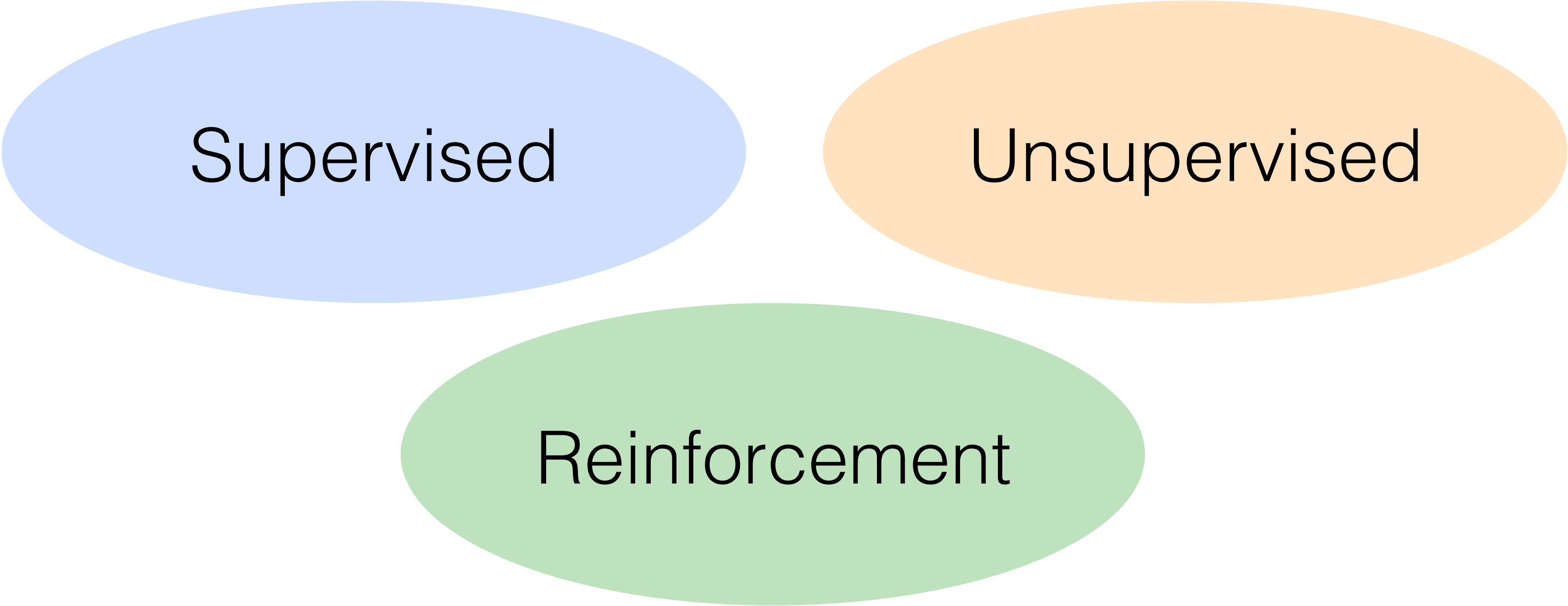
Emails

Spam/Non-Spam Labels

Classification Algorithm

Spam Filter

3 types of learning

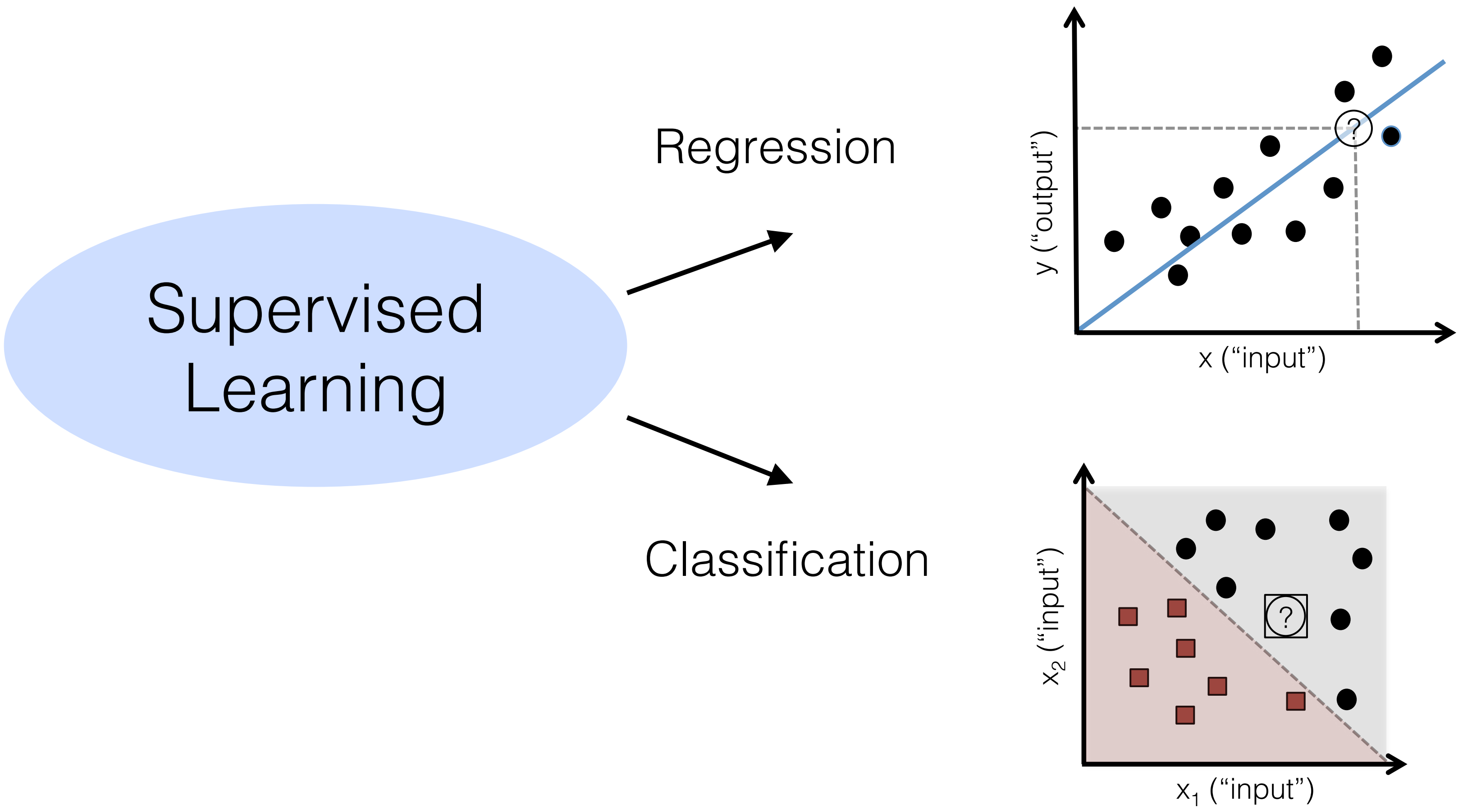


Supervised

Unsupervised

Reinforcement

Working with labeled data

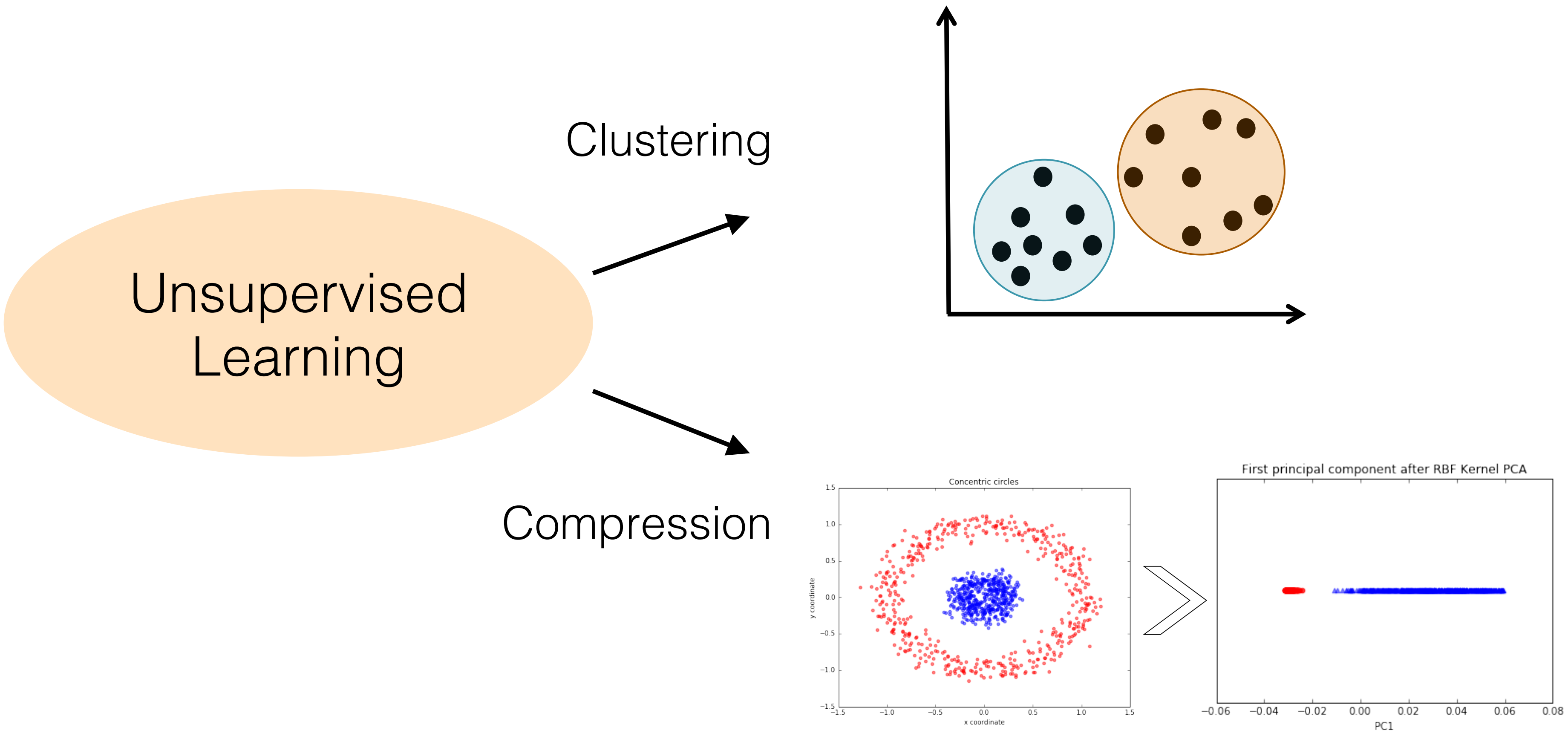


Supervised Learning

Regression

Classification

Working with <u>unlabeled</u> data



Unsupervised Learning

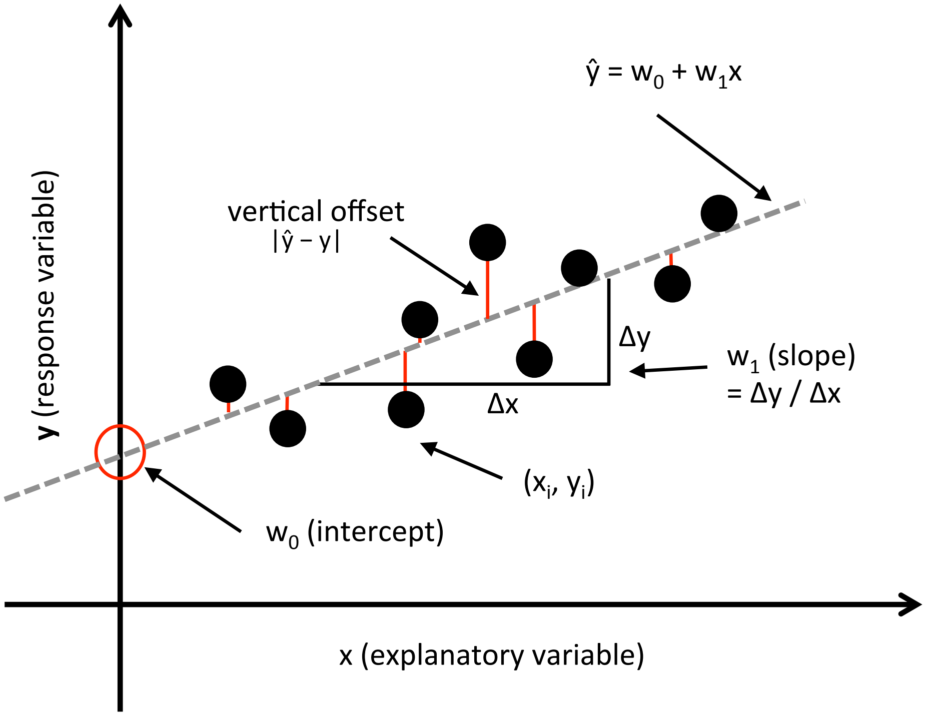
Clustering

Compression

Topics

1. Introduction to Machine Learning
2. **Linear Regression**
3. Introduction to Classification
4. Feature Preprocessing & scikit-learn Pipelines
5. Dimensionality Reduction: Feature Selection & Extraction
6. Support Vector Machine Classifiers
7. Model Evaluation & Hyperparameter Tuning
8. Tree-based Methods
9. Unsupervised Learning: Clustering

Simple Linear Regression

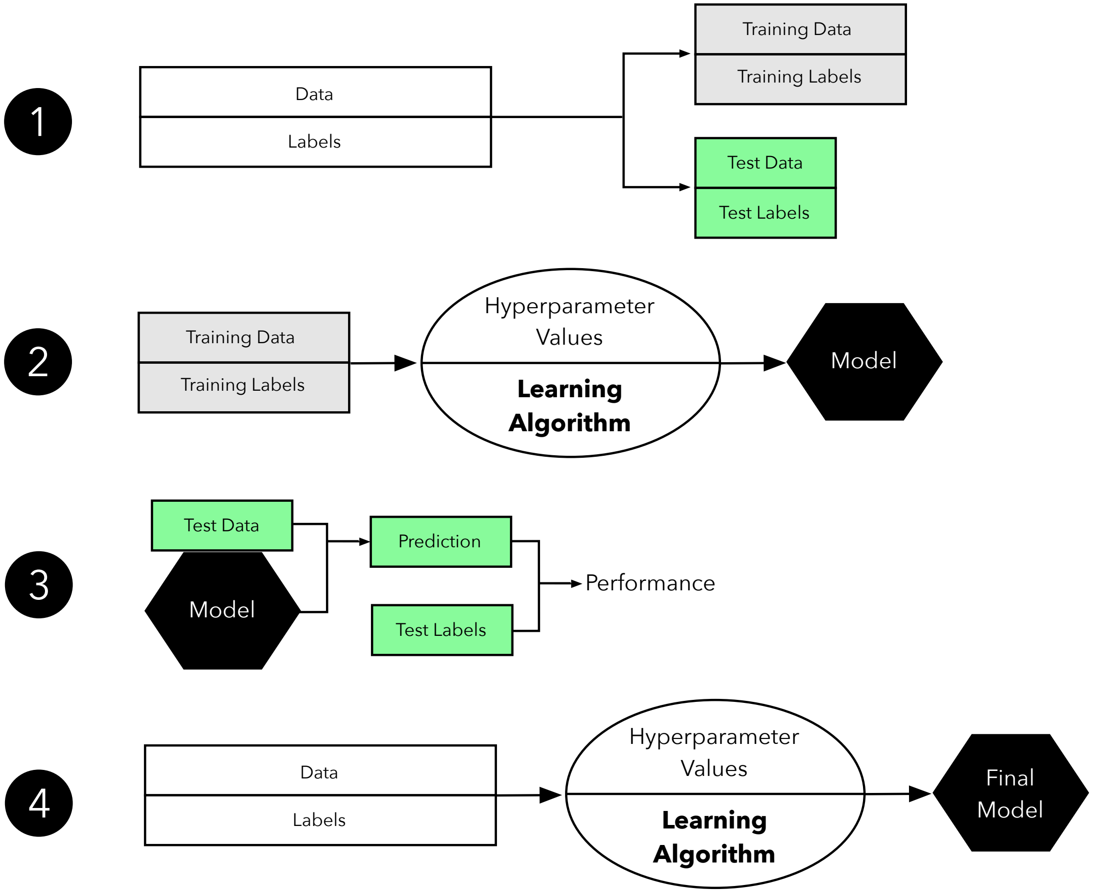


Data representation

$$\mathbf{X} = \overbrace{\begin{array}{|c|c|c|c|} \hline \mathbf{x_0} & \mathbf{x_1} & \dots & \mathbf{x_m} \\ \hline x_{0,0} & x_{0,1} & & \\ x_{1,0} & x_{1,1} & & \\ x_{2,0} & x_{2,1} & & \\ x_{3,0} & x_{3,1} & & \\ . & & & \\ . & & & \\ . & & & \\ \hline x_{n,0} & x_{n,1} & \dots & x_{n,m} \\ \hline \end{array}}^{\text{features (columns)}} \left.\vphantom{\begin{array}{c} x \\ x \\ x \\ x \\ x \\ x \\ x \\ x \end{array}}\right\} \text{samples (rows)}$$

$$\mathbf{y} = \begin{array}{|c|} \hline y_0 \\ y_1 \\ y_2 \\ y_3 \\ . \\ . \\ . \\ \hline y_n \\ \hline \end{array}$$

“Basic” Supervised Learning Workflow



Coding Example

➡ Jupyter Notebook

Topics

1. Introduction to Machine Learning
2. Linear Regression
3. **Introduction to Classification**
4. Feature Preprocessing & scikit-learn Pipelines
5. Dimensionality Reduction: Feature Selection & Extraction
6. Support Vector Machine Classifiers
7. Model Evaluation & Hyperparameter Tuning
8. Tree-based Methods
9. Unsupervised Learning: Clustering

Scikit-learn API

```
class SupervisedEstimator(...):
    def __init__(self, hyperparam, ...):
        ...
    def fit(self, X, y):
        ...
        return self
    def predict(self, X):
        ...
        return y_pred
    def score(self, X, y):
        ...
        return score
    ...
```

Iris dataset

Iris-Setosa

Iris-Setosa

Iris-Versicolor

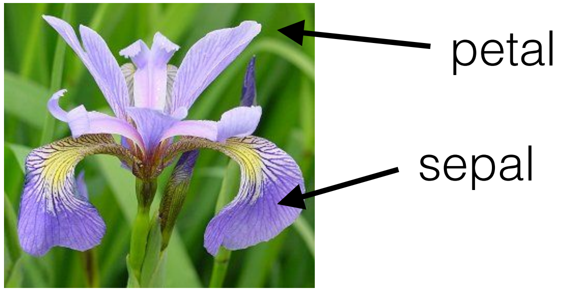
Iris dataset

features (columns)

X=

	sepal length [cm]	sepal width [cm]	petal length [cm]	petal width [cm]
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
50	6.4	3.5	4.5	1.2
	.			
	.			
	.			
150	5.9	3.0	5.0	1.8

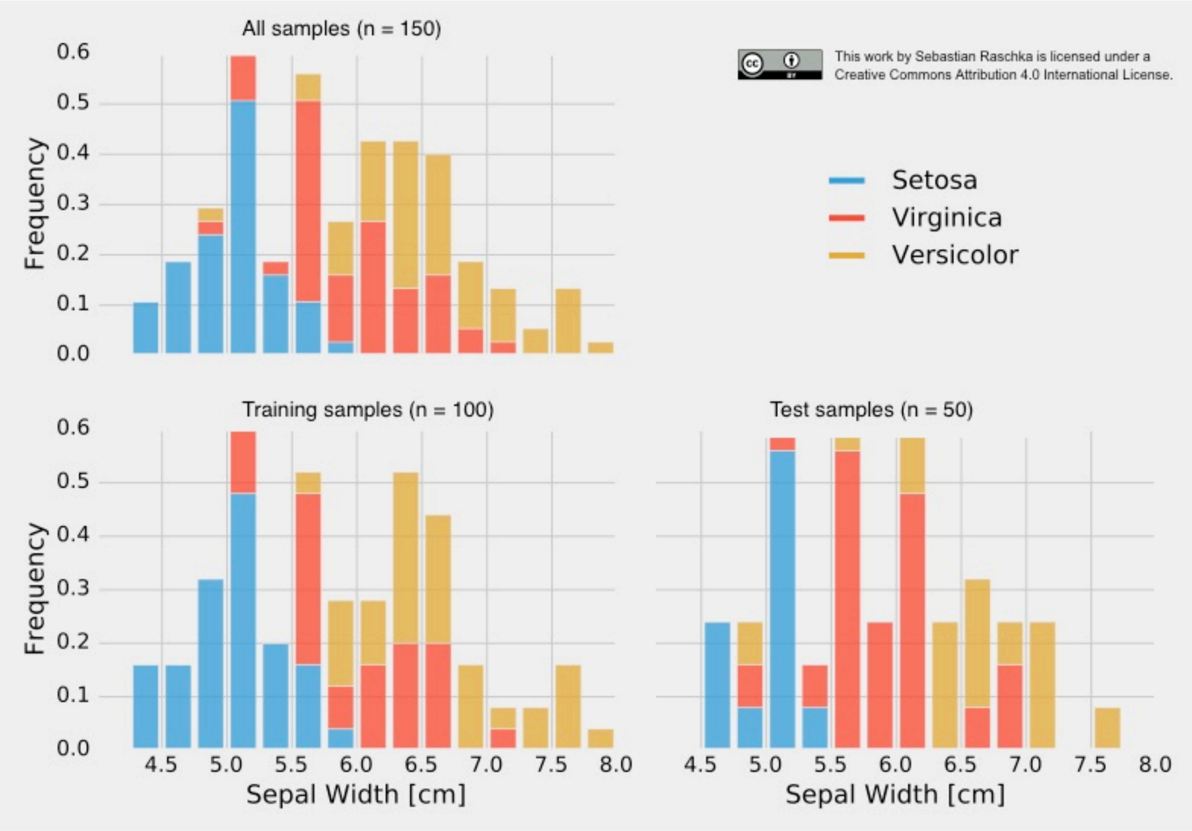
samples (rows)



y=

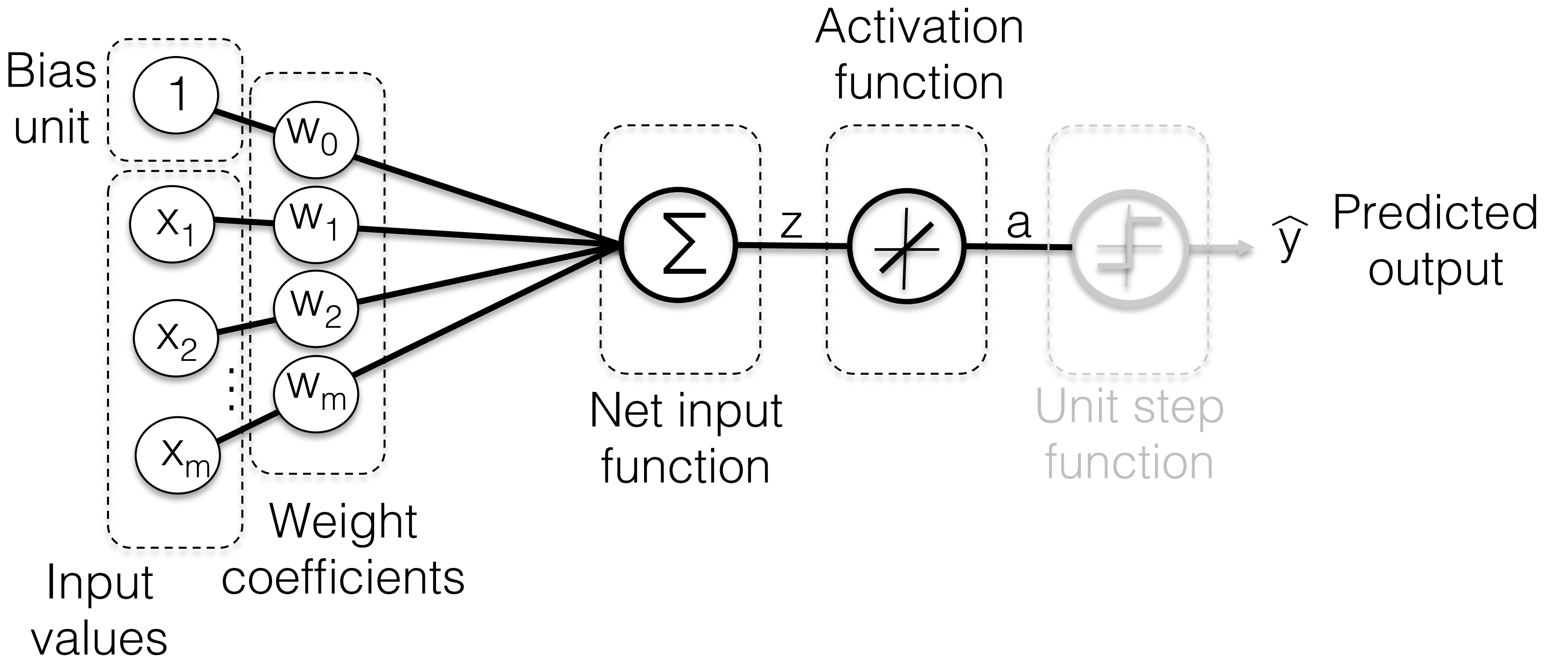
setosa
setosa
versicolor
.
.
.
virginica

Note about non-stratified splits

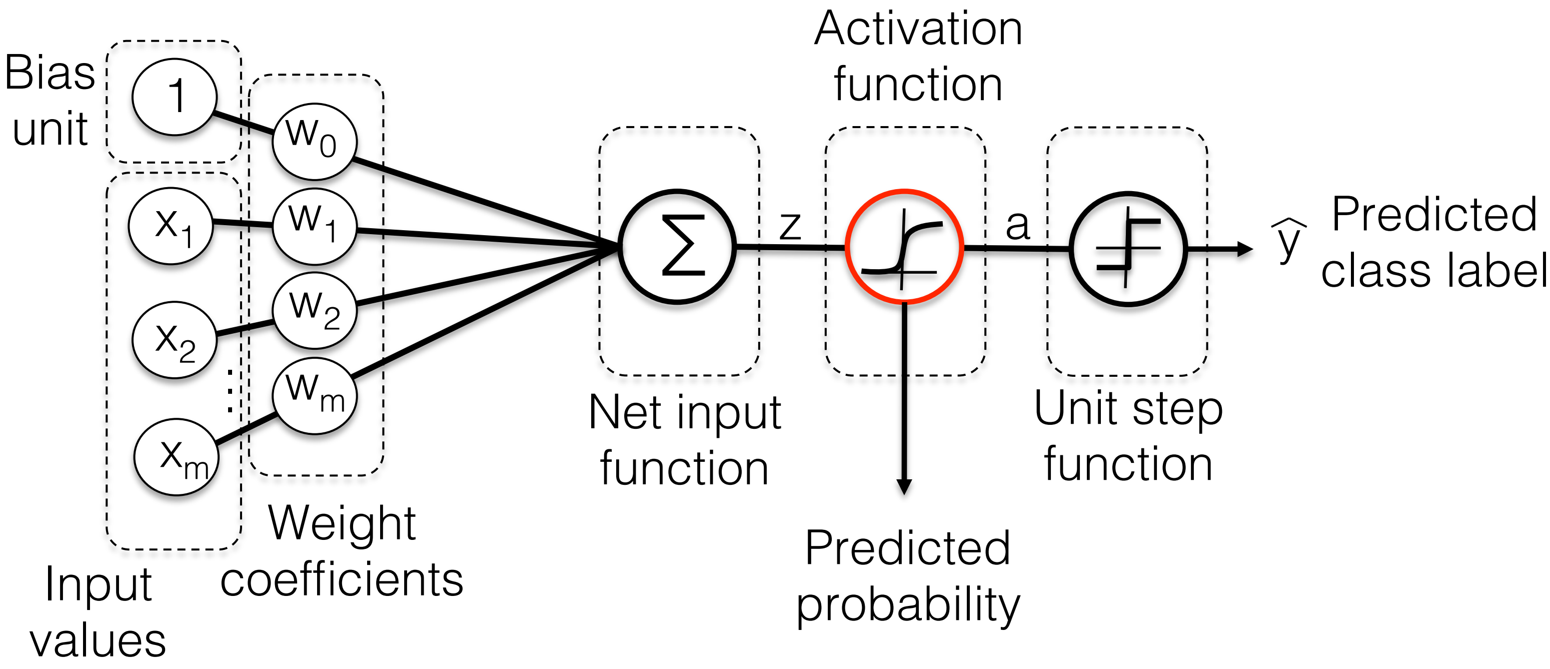


- training set → 38 x Setosa, 28 x Versicolor, 34 x Virginica
- test set → 12 x Setosa, 22 x Versicolor, 16 x Virginica

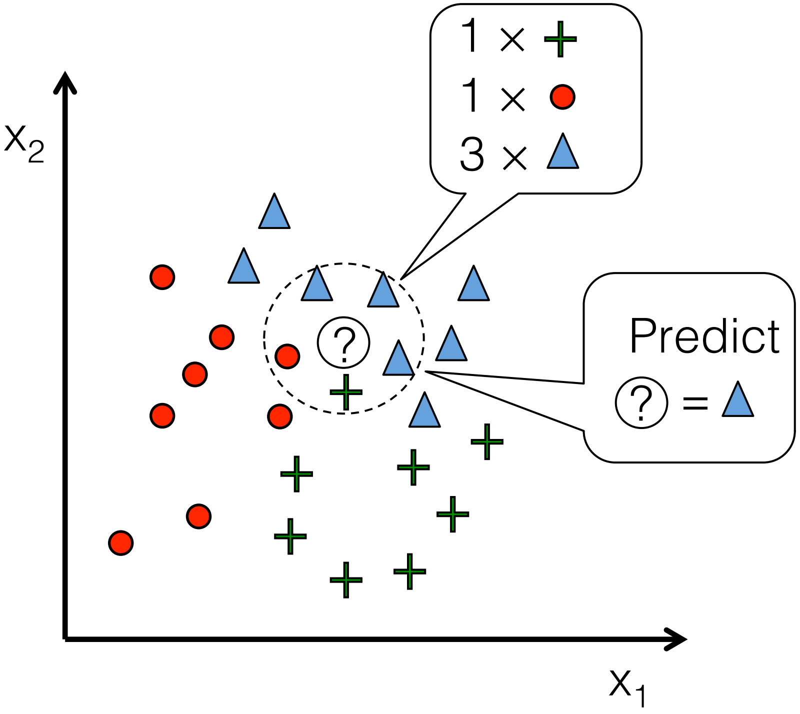
Linear Regression Recap



Logistic Regression, a generalized linear model



A “lazy learner:” K-Nearest Neighbors classifier



1 × +
1 × ●
3 × ▲

Predict
? = ▲

Coding Example

➡ Jupyter Notebook

Topics

1. Introduction to Machine Learning
2. Linear Regression
3. Introduction to Classification
4. **Feature Preprocessing & scikit-learn Pipelines**
5. Dimensionality Reduction: Feature Selection & Extraction
6. Support Vector Machine Classifiers
7. Model Evaluation & Hyperparameter Tuning
8. Tree-based Methods
9. Unsupervised Learning: Clustering

Scikit-learn API

```
class UnsupervisedEstimator(...):
    def __init__(self, hyperparam, ...):
        ...
    def fit(self, X, y):
        ...
        return self
    def predict(self, X):
        ...
        return y_pred
    def score(self, X, y):
        ...
        return score
    ...
```

Topics

1. Introduction to Machine Learning
2. Linear Regression
3. Introduction to Classification
4. Feature Preprocessing & scikit-learn Pipelines
5. **Dimensionality Reduction Feature Selection & Extraction**
6. Support Vector Machine Classifiers
7. Model Evaluation & Hyperparameter Tuning
8. Tree-based Methods
9. Unsupervised Learning: Clustering

Topics

1. Introduction to Machine Learning
2. Linear Regression
3. Introduction to Classification
4. Feature Preprocessing & scikit-learn Pipelines
5. Dimensionality Reduction: Feature Selection & Extraction
6. **Support Vector Machine Classifiers**
7. Model Evaluation & Hyperparameter Tuning
8. Tree-based Methods
9. Unsupervised Learning: Clustering

Topics

1. Introduction to Machine Learning
2. Linear Regression
3. Introduction to Classification
4. Feature Preprocessing & scikit-learn Pipelines
5. Dimensionality Reduction: Feature Selection & Extraction
6. Support Vector Machine Classifiers
7. **Model Evaluation & Hyperparameter Tuning**
8. Tree-based Methods
9. Unsupervised Learning: Clustering

Topics

1. Introduction to Machine Learning
2. Linear Regression
3. Introduction to Classification
4. Feature Preprocessing & scikit-learn Pipelines
5. Dimensionality Reduction: Feature Selection & Extraction
6. Support Vector Machine Classifiers
7. Model Evaluation & Hyperparameter Tuning
8. **Tree-based Methods**
9. Unsupervised Learning: Clustering

Topics

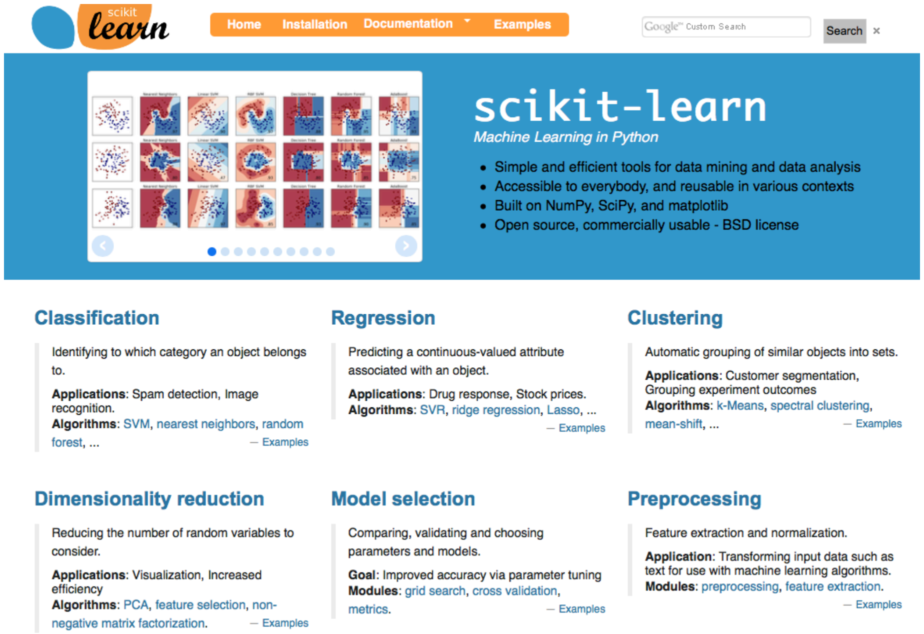
1. Introduction to Machine Learning
2. Linear Regression
3. Introduction to Classification
4. Feature Preprocessing & scikit-learn Pipelines
5. Dimensionality Reduction: Feature Selection & Extraction
6. Support Vector Machine Classifiers
7. Model Evaluation & Hyperparameter Tuning
8. Tree-based methods
9. **Unsupervised Learning: Clustering**

Topics

1. Introduction to Machine Learning
2. Linear Regression
3. Introduction to Classification
4. Feature Preprocessing & scikit-learn Pipelines
5. Dimensionality Reduction: Feature Selection & Extraction
6. Support Vector Machine Classifiers
7. Model Evaluation & Hyperparameter Tuning
8. Tree-based methods
9. Unsupervised Learning: Clustering

Further Resources

Documentation:
http://scikit-learn.org



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ...
— Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ...
— Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ...
— Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization.
— Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics.
— Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction.
— Examples

Mailing list:
https://mail.python.org/mailman/listinfo/scikit-learn

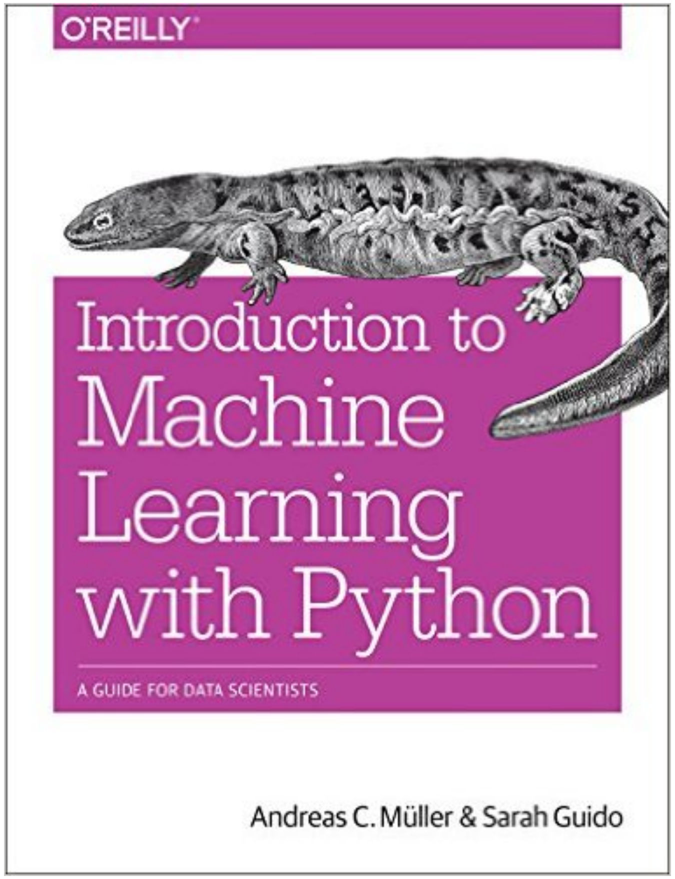
Further Resources

Great "math-free," practical guide to machine learning with scikit-learn

By Andreas Mueller (scikit-learn core developer) and Sarah Guido

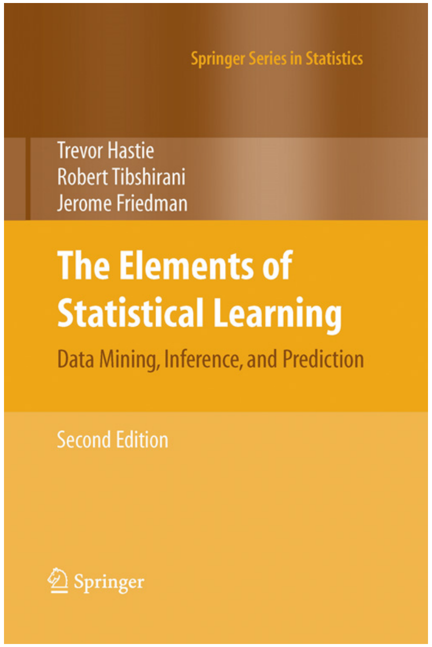
http://shop.oreilly.com/product/0636920030515.do

Estimated release: October 20, 2016


O'REILLY
Introduction to Machine Learning with Python
A GUIDE FOR DATA SCIENTISTS
Andreas C. Müller & Sarah Guido

Further Resources

My favorite machine learning "math & theory" books



Springer Series in Statistics

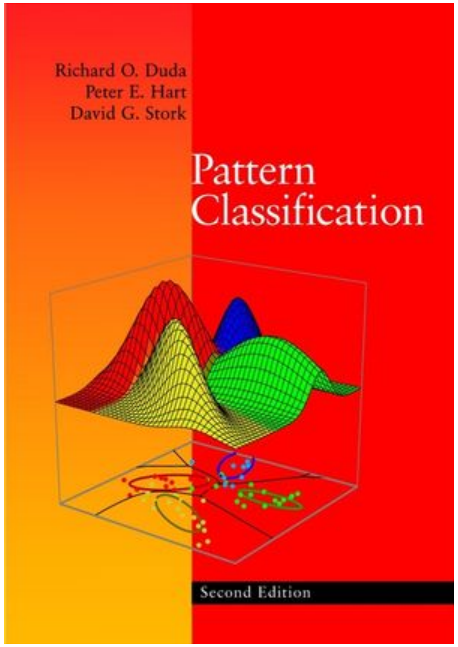
Trevor Hastie
Robert Tibshirani
Jerome Friedman

The Elements of Statistical Learning

Data Mining, Inference, and Prediction

Second Edition

Springer



Richard O. Duda
Peter E. Hart
David G. Stork

Pattern Classification

Second Edition

http://statweb.stanford.edu/~tibs/ElemStatLearn/ (free PDF)

http://www.wiley.com/WileyCDA/WileyTitle/productCd-0471056693.html

Further Resources

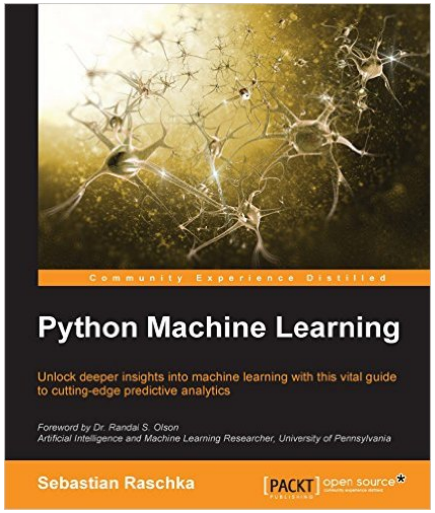
My own book, math, from-scratch code, and practical scikit-learn code:

GitHub repository:

https://github.com/rasbt/python-machine-learning-book

Amazon link:

https://www.amazon.com/Python-Machine-Learning-Sebastian-Raschka/dp/1783555130/


Community Experience Distilled
Python Machine Learning
Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics
Foreword by Dr. Randal S. Olson
Artificial Intelligence and Machine Learning Researcher, University of Pennsylvania
Sebastian Raschka
PACKT open source