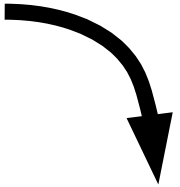
Machine Learning with Scikit-Learn

Andreas Mueller (NYU Center for Data Science, scikit-learn)

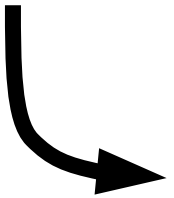
http://bit.ly/sklstrata

Me


universitätbonn
ais




Amazon
Machine
Learning




NYU
Center for
Data Science

Classification
Regression
Clustering
Semi-Supervised Learning
Feature Selection
Feature Extraction
Manifold Learning
Dimensionality Reduction
Kernel Approximation
Hyperparameter Optimization
Evaluation Metrics
Out-of-core learning
……

scikit learn

agramfort
Alexandre Gramfort

AlexanderFabisch
Alexander Fabisch

alextp
Alexandre Passos

amueller
Andreas Mueller

arjoly
Arnaud Joly

bdholt1
Brian Holt

bthirion
bthirion

chrisfilo
Chris Filo Gorgole…

cournape
David Cournapeau

duchesnay
Duchesnay

dwf
David Warde-Farley

fabianp
Fabian Pedregosa

GaelVaroquaux
Gael Varoquaux

glouppe
Gilles Louppe

jakevdp
Jake Vanderplas

jaquesgrobler
Jaques Grobler

jnothman

kastnerkyle
Kyle Kastner

kuantkid
Wei LI

larsmans
Lars

lucidfrontier45
Shiqiao Du

mblondel
Mathieu Blondel

MechCoder
Manoj Kumar

ndawe
Noel Dawe

NelleV
Varoquaux

ogrisel
Olivier Grisel

paolo-losi
Paolo Losi

pprett
Peter Prettenhofer

robertlayton
Robert Layton

ronw
Ron Weiss

satra
Satrajit Ghosh

sklearn-ci

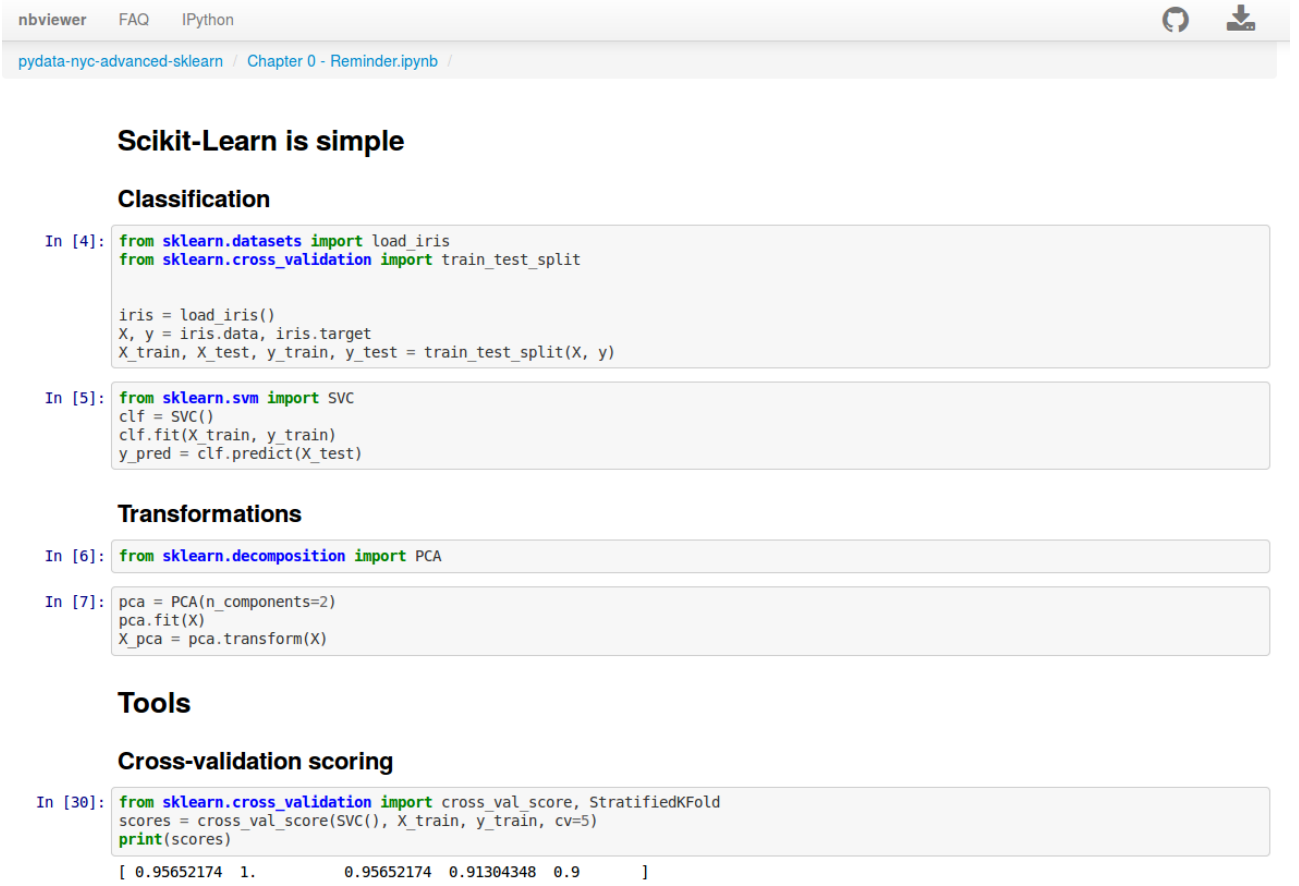
vene
Vlad Niculae

VirgileFritsch
Virgile Fritsch

vmichel
Vincent Michel

yarikoptic
Yaroslav Halchenko

Get the notebooks!



http://bit.ly/sklstrata

Hi Andy,

I just received an email from the first tutorial speaker, presenting right before you, saying he's ill and won't be able to make it.

I know you have already committed yourself to two presentations, but is there anyway you could increase your tutorial time slot, maybe just offer time to try out what you've taught? Otherwise I have to do some kind of modern dance interpretation of Python in data :-)
-Leah

Hi Andreas,

I am very interested in your Machine Learning background. I work for X Recruiting who have been engaged by Z, a worldwide leading supplier of Y. We are expanding the core engineering team and we are looking for really passionate engineers who want to create their own story and help millions of people.

Can we find a time for a call to chat for a few minutes about this?

Thanks

Hi Andy,

I just received an email fr… the first tutorial speaker, presenting … u, saying he's ill and won't be…

I know you have a… …urself to two presentations… …y you could increase you… …, maybe just offer time to try out what you've taught? Otherwise I have to do some kind of modern dance interpretation of Python in data :-)
-Leah

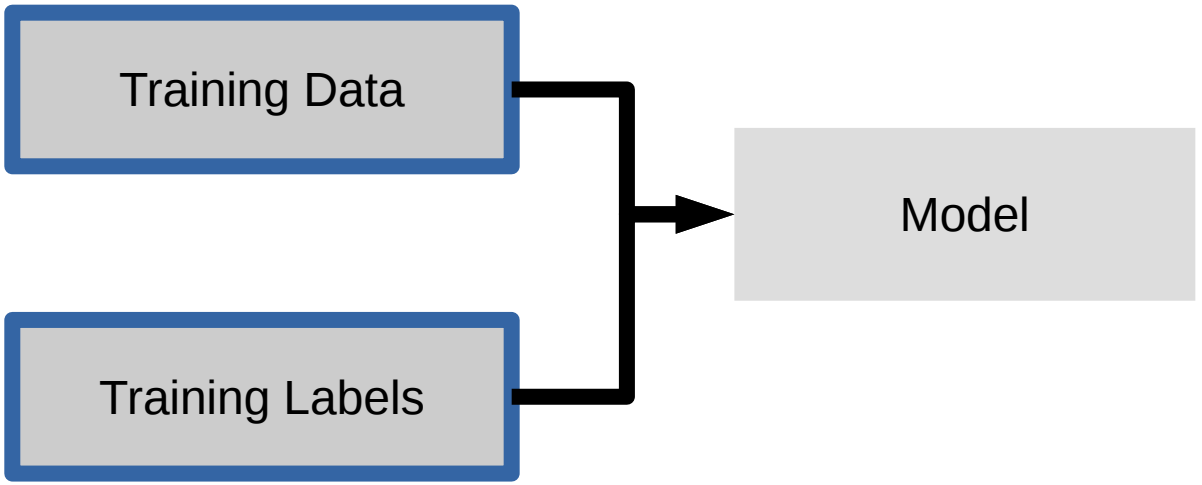
Hi Andreas,

I am very interested in your Machine Learning background. I work for X Recruiting who have been engaged by Z, a worldwide leading supplier of Y. We are expanding the core engineering team and we are looking for really passionate engineers who… …own story and help millions of…

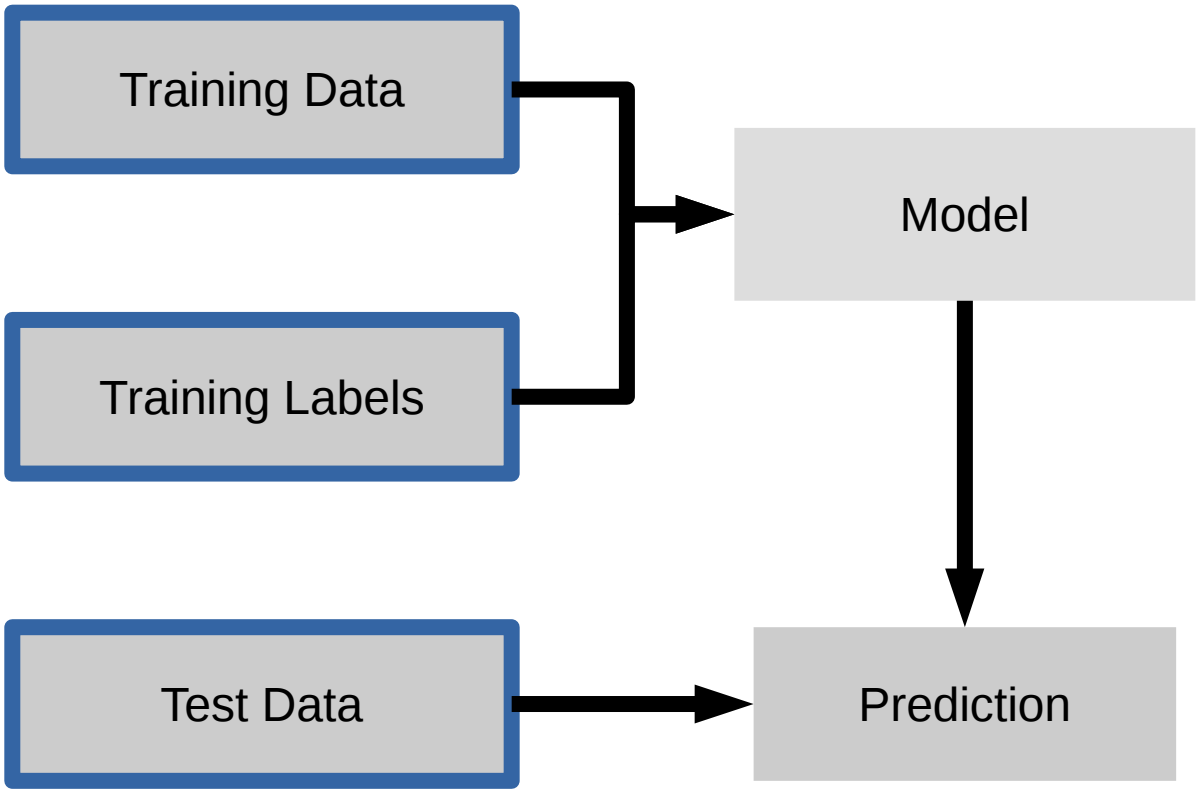
Can we find a ti… …for a few minutes about t…

Thanks

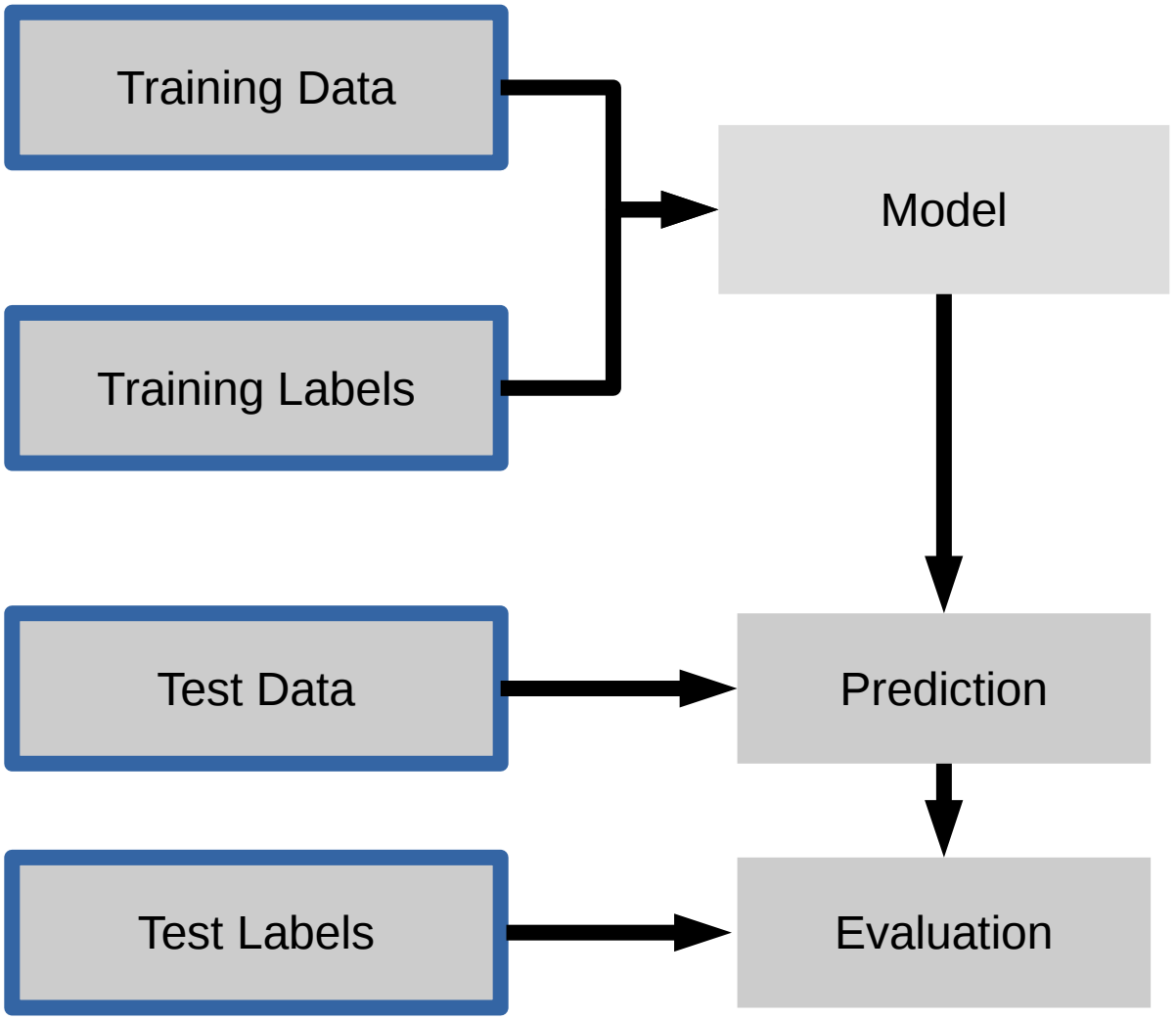
Supervised Machine Learning



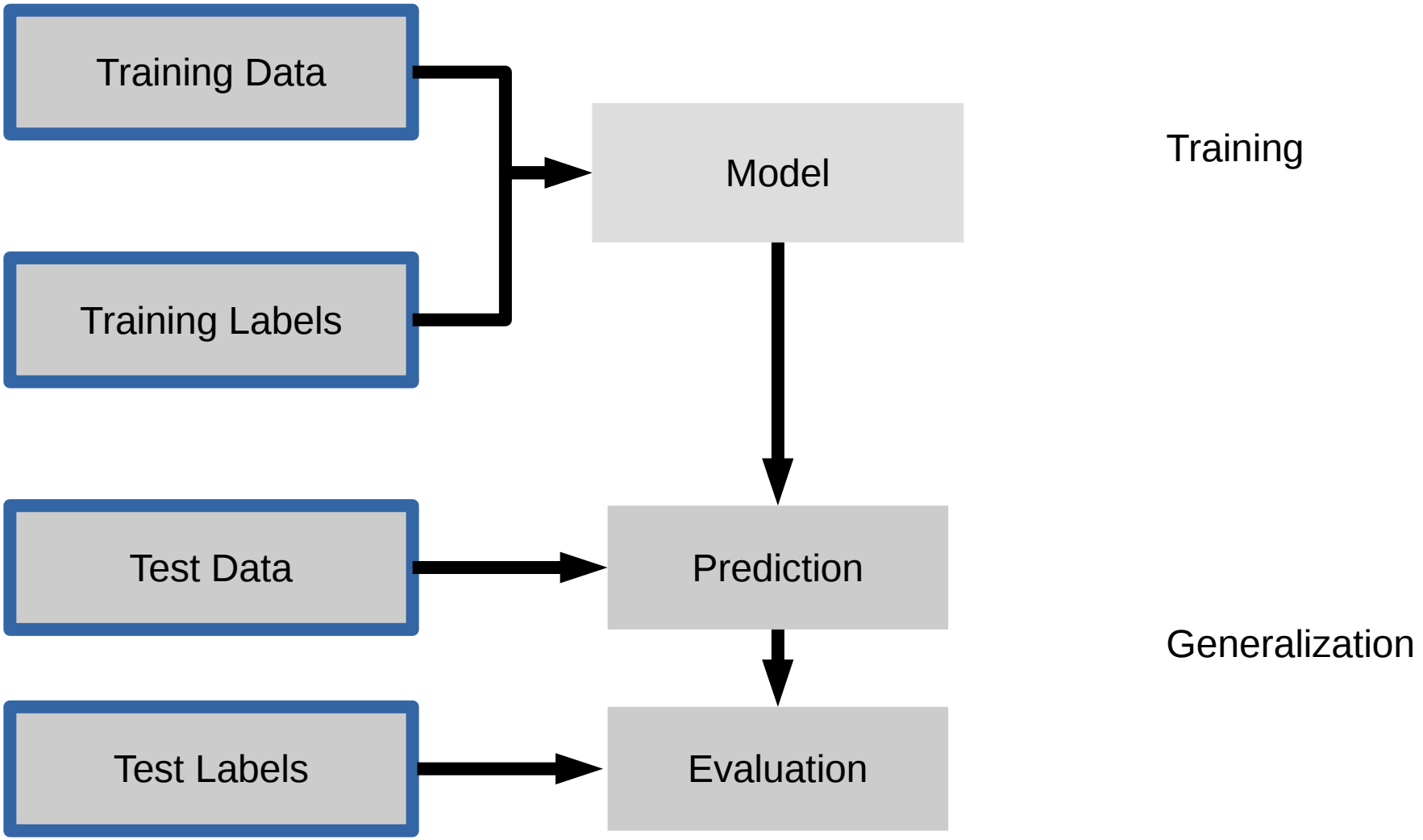
Supervised Machine Learning



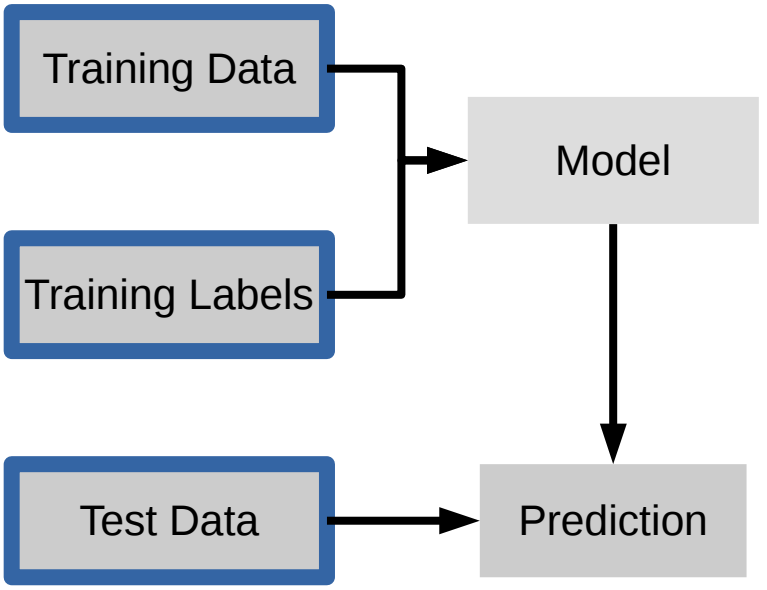
Supervised Machine Learning



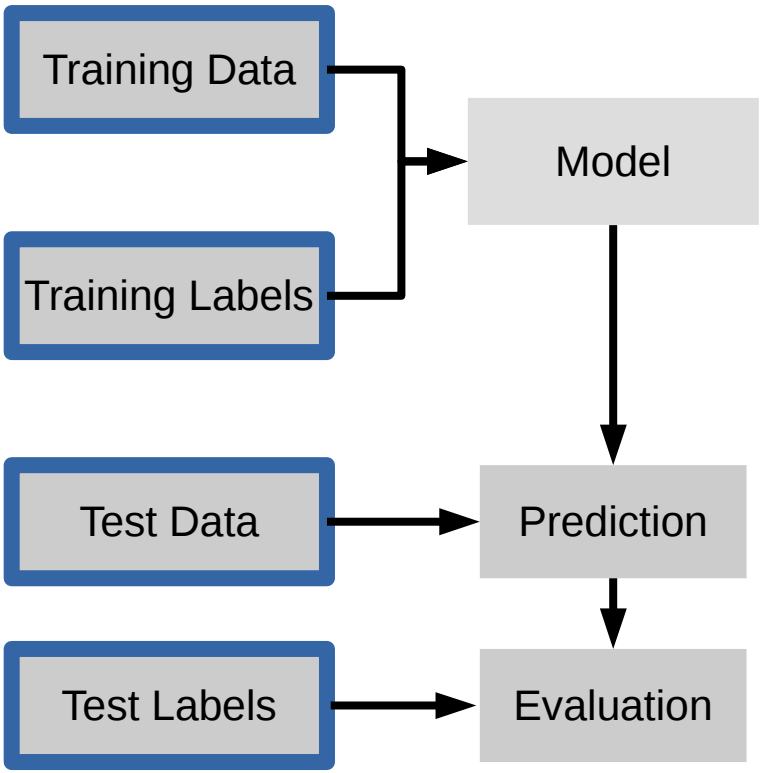
Supervised Machine Learning



```
clf = RandomForestClassifier()

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
```

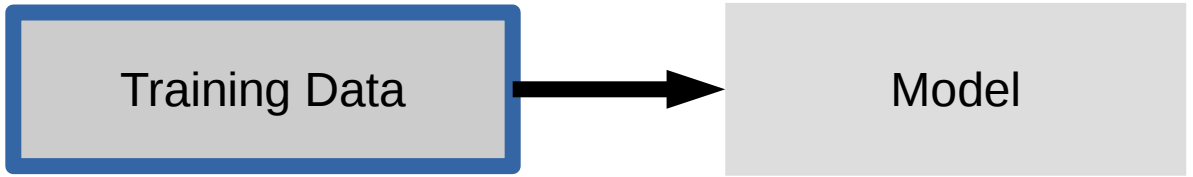


```
clf = RandomForestClassifier()

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

clf.score(X_test, y_test)
```

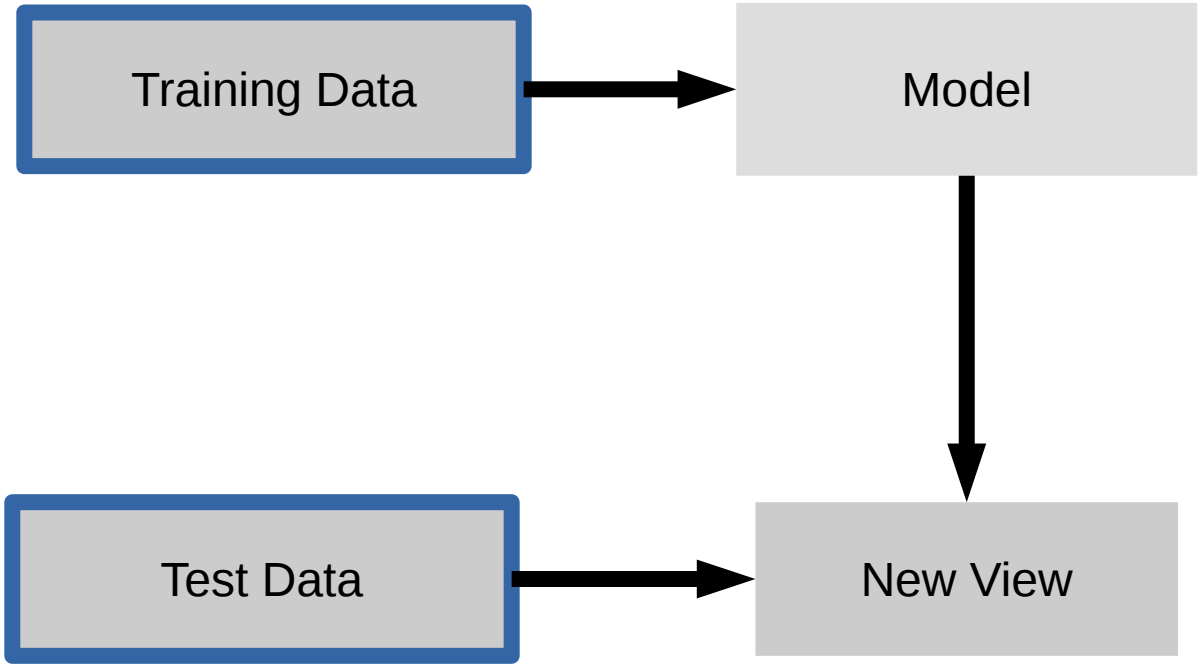


IPython Notebook:
Chapter 1 - Introduction to Scikit-learn

Unsupervised Machine Learning



Unsupervised Machine Learning

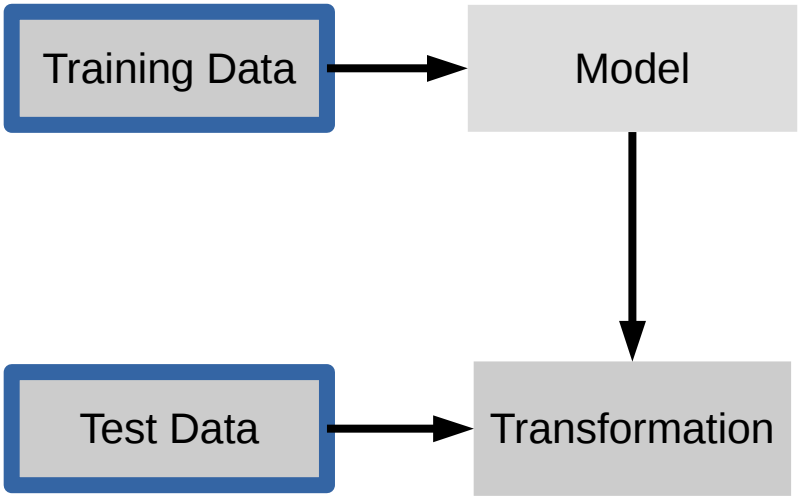


Unsupervised Transformations

```
pca = PCA()
```

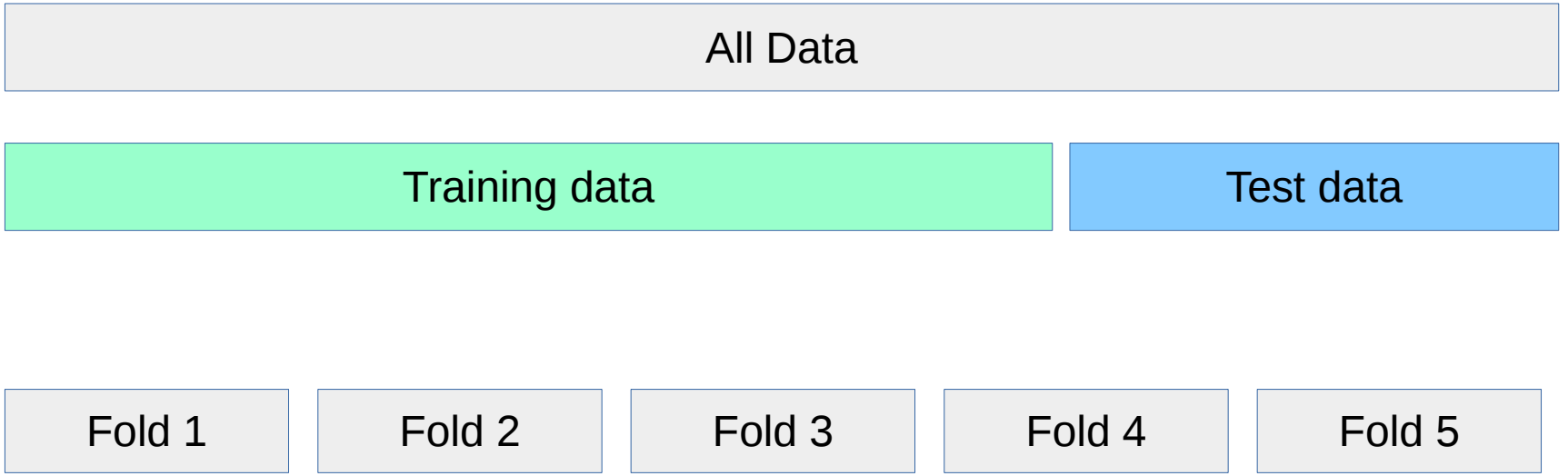
```
pca.fit(X_train)
```

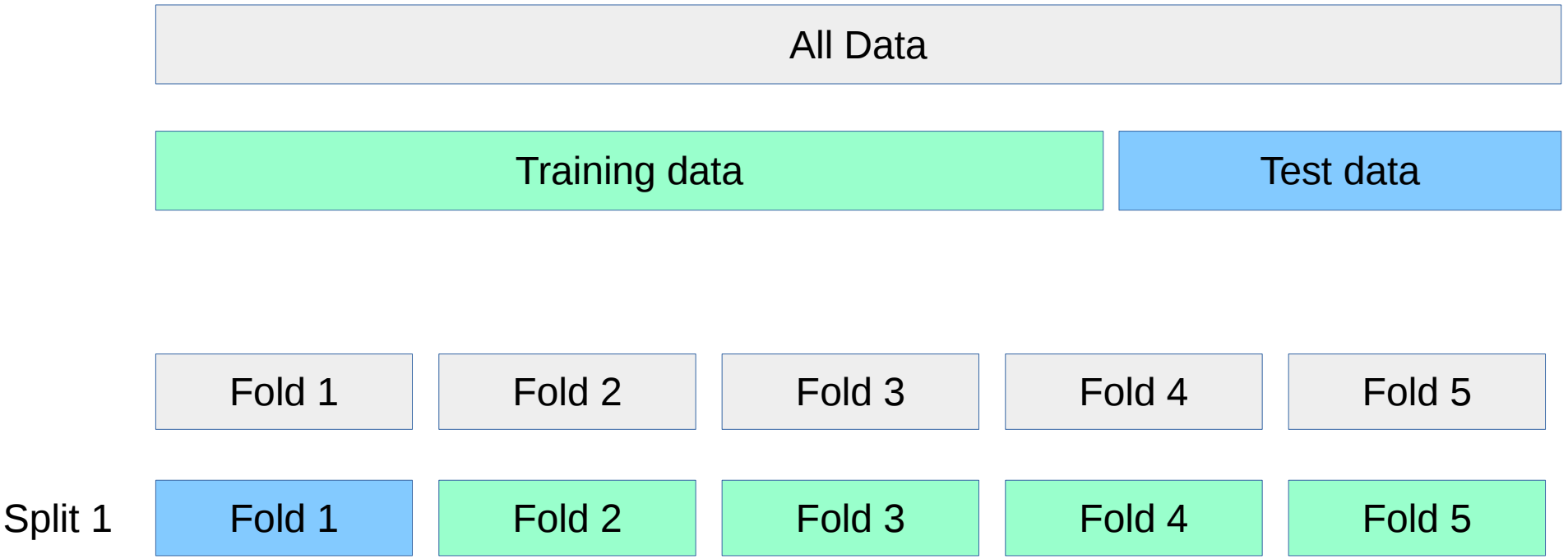
```
X_new = pca.transform(X_test)
```

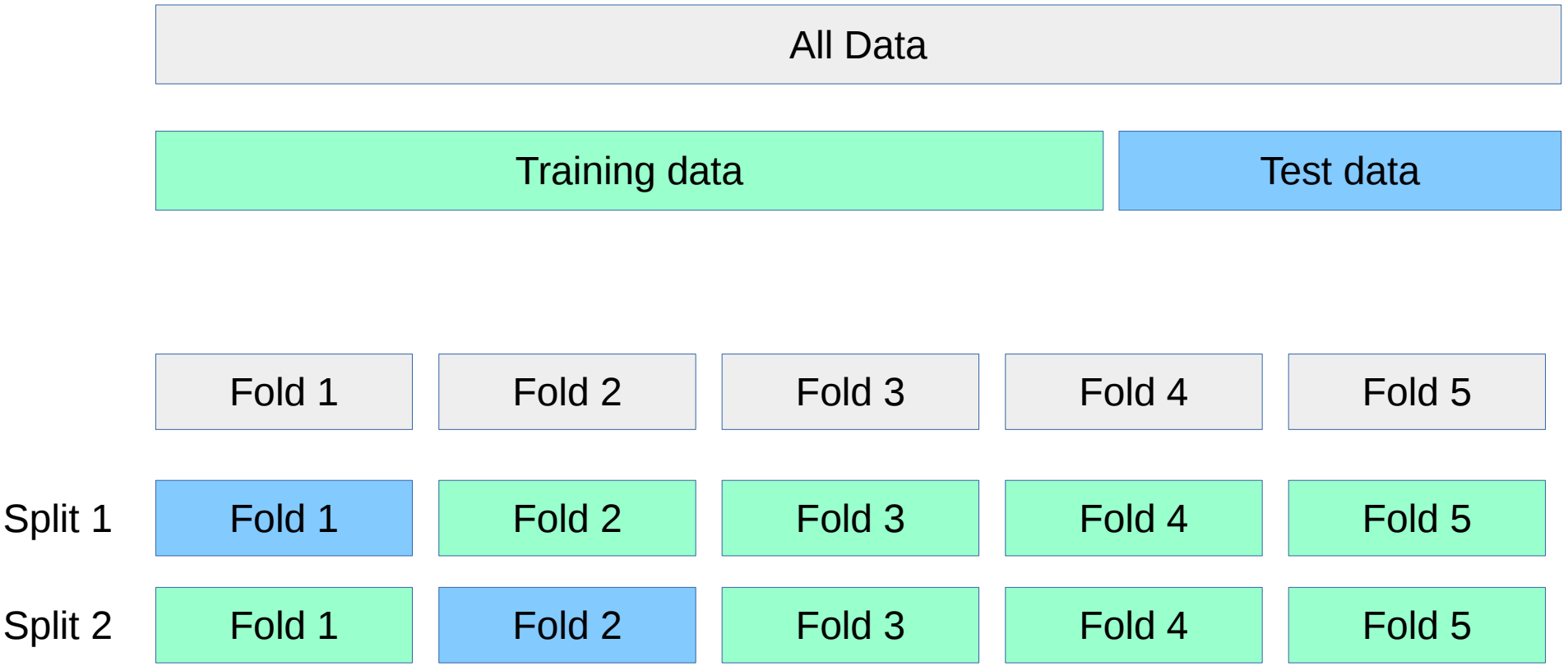


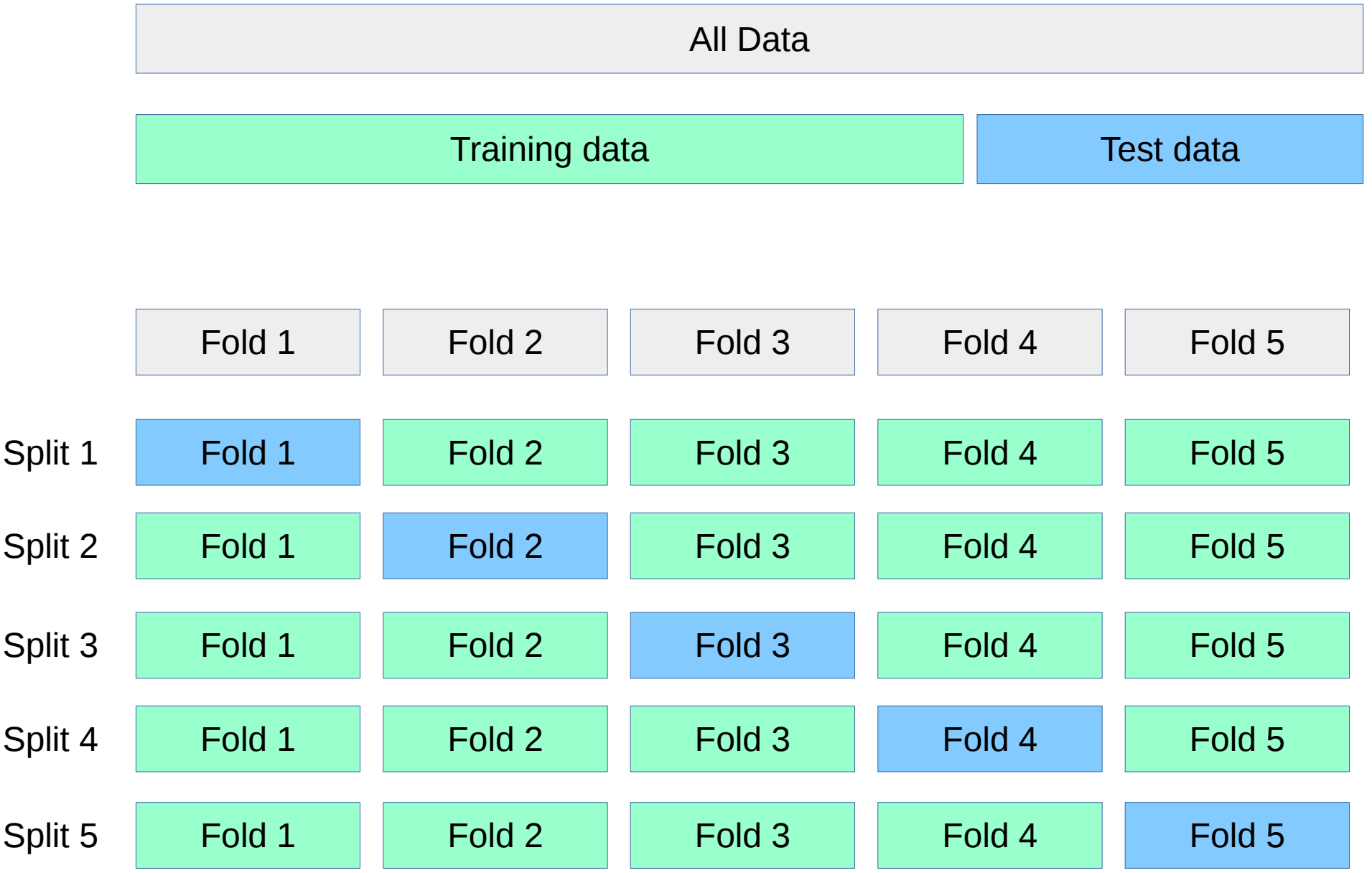
IPython Notebook:
Chapter 2 – Unsupervised Transformers


All Data
Training data
Test data


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5

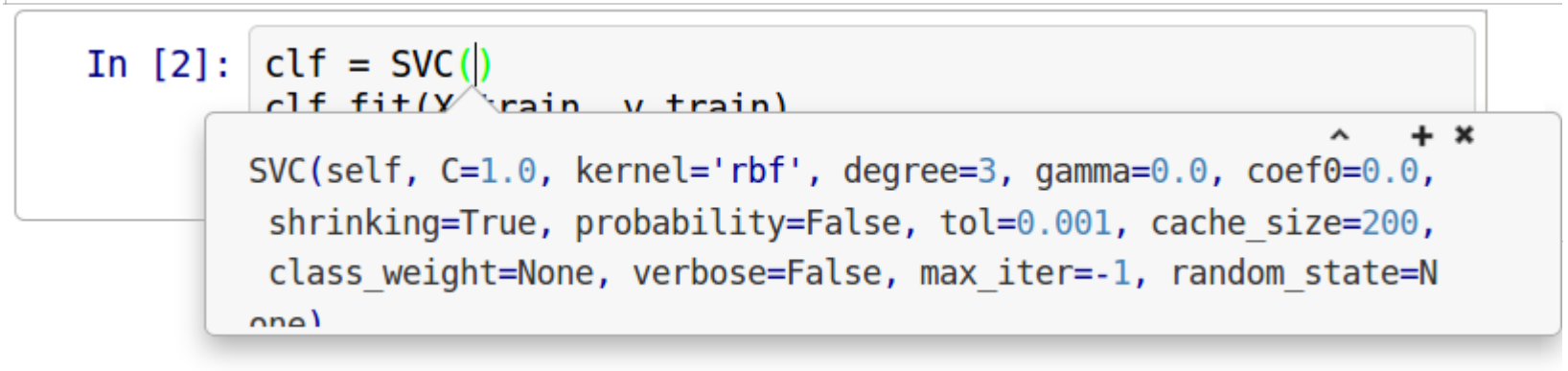

All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 2
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5

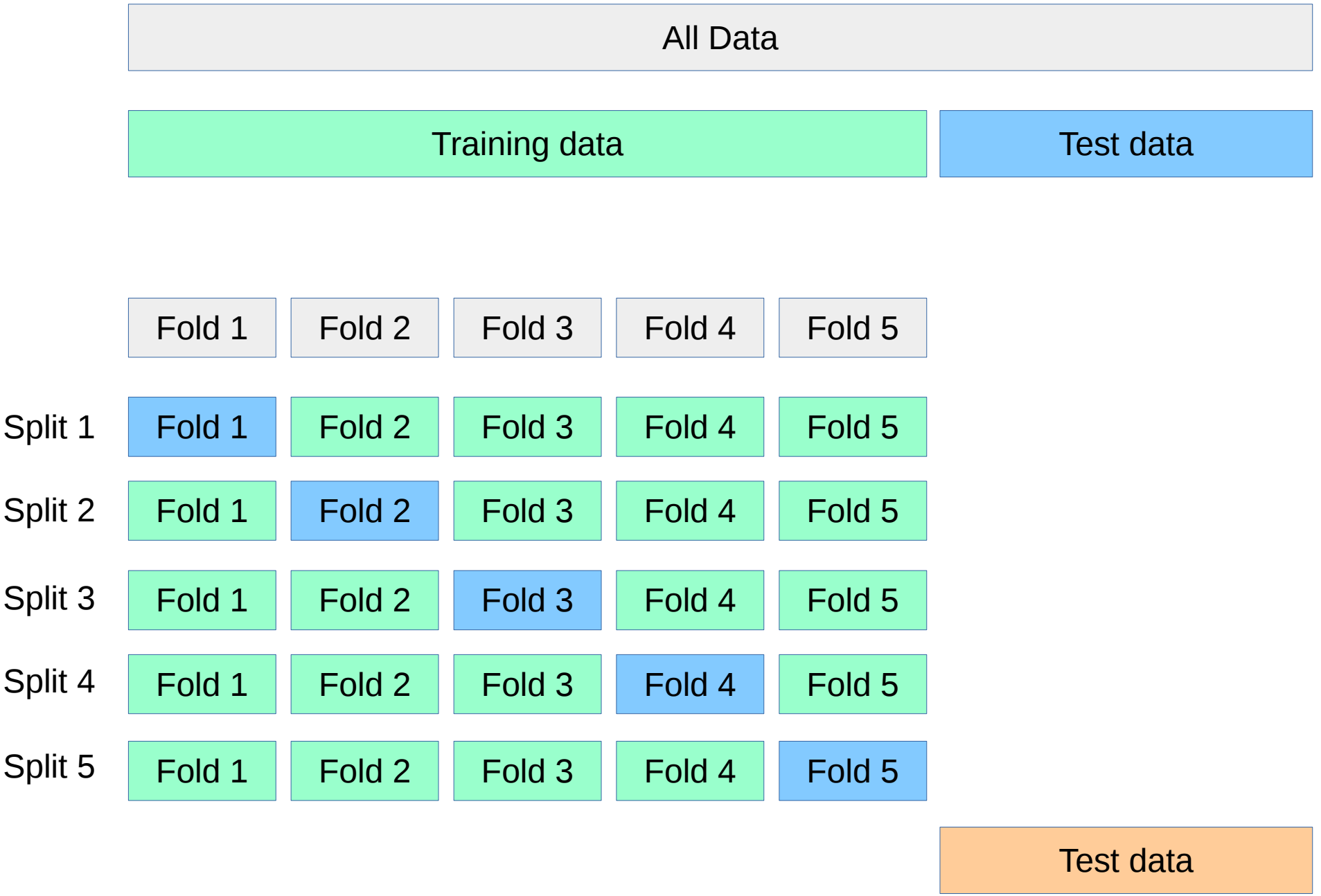

All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Split 2
Split 3
Split 4
Split 5

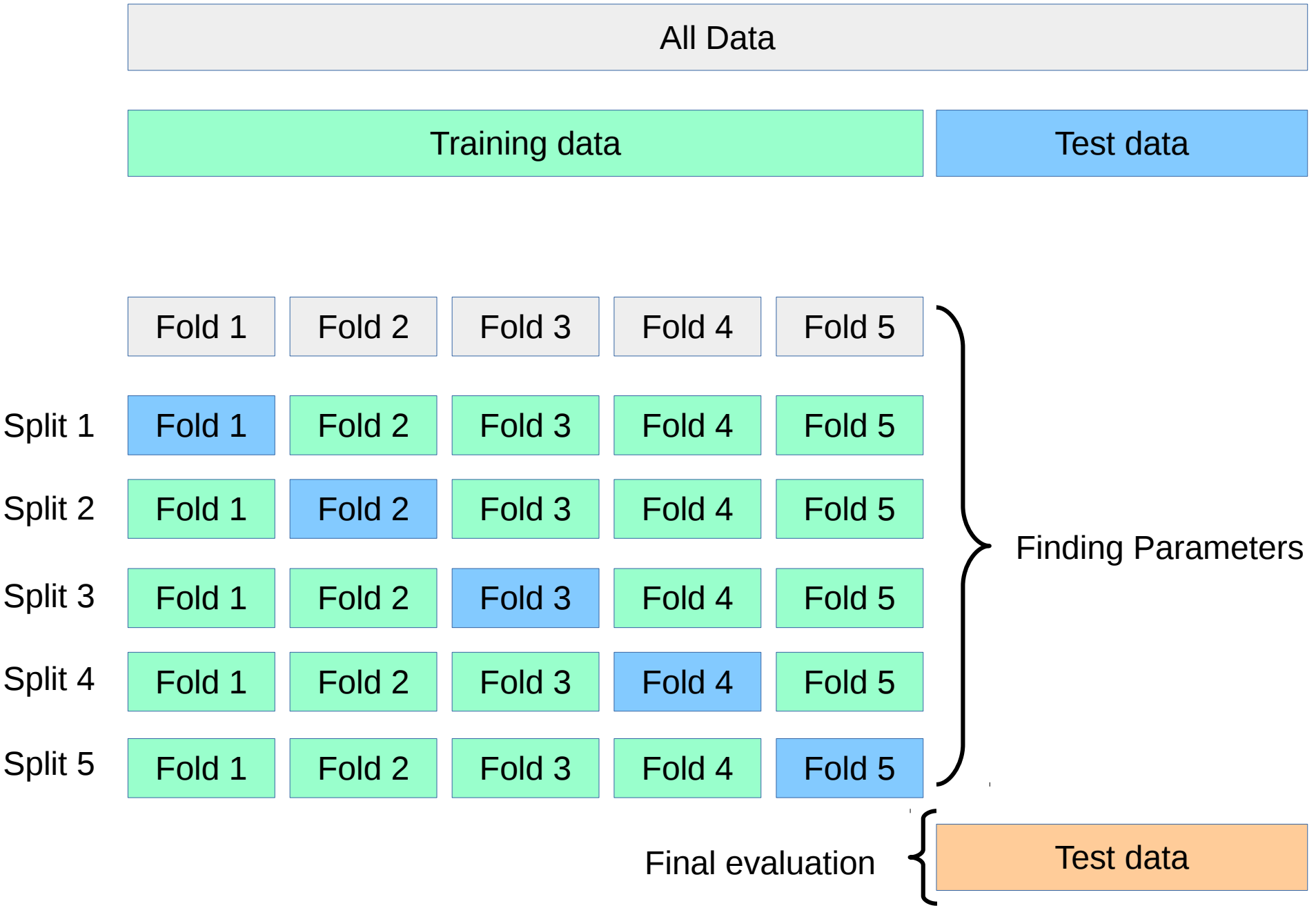
IPython Notebook:
Chapter 3 - Cross-validation

In [2]:
```
clf = SVC()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```


In [2]:
clf = SVC()
clf.fit(X_train, y_train)
SVC(self, C=1.0, kernel='rbf', degree=3, gamma=0.0, coef0=0.0,
shrinking=True, probability=False, tol=0.001, cache_size=200,
class_weight=None, verbose=False, max_iter=-1, random_state=N
one)


All Data
Training data
Test data


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Split 2
Split 3
Split 4
Split 5
Test data


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Split 2
Split 3
Split 4
Split 5
Finding Parameters
Final evaluation
Test data

SVC(C=0.001,
gamma=0.001)

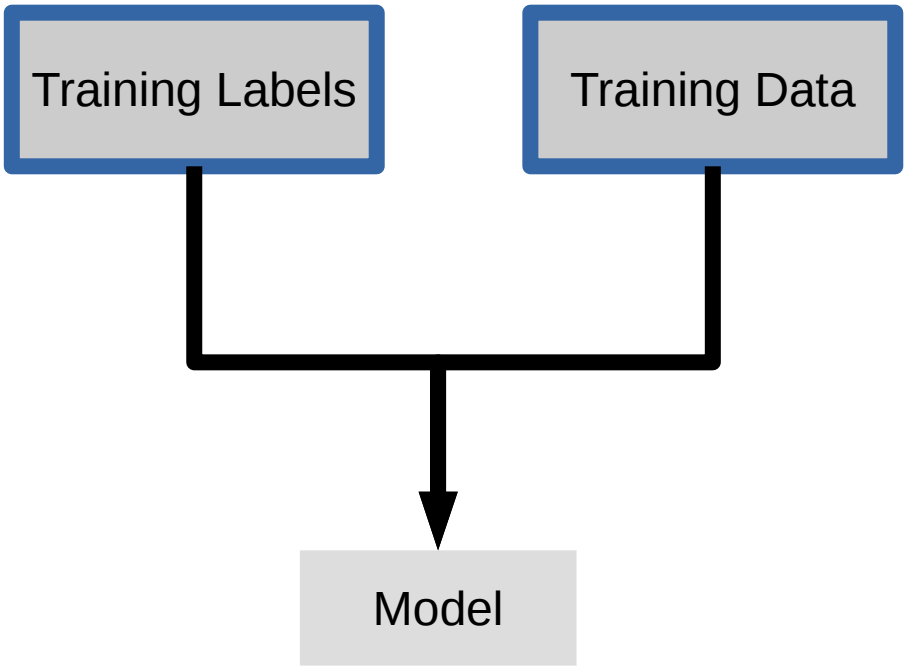
SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	SVC(C=0.1, gamma=0.001)	SVC(C=1, gamma=0.001)	SVC(C=10, gamma=0.001)

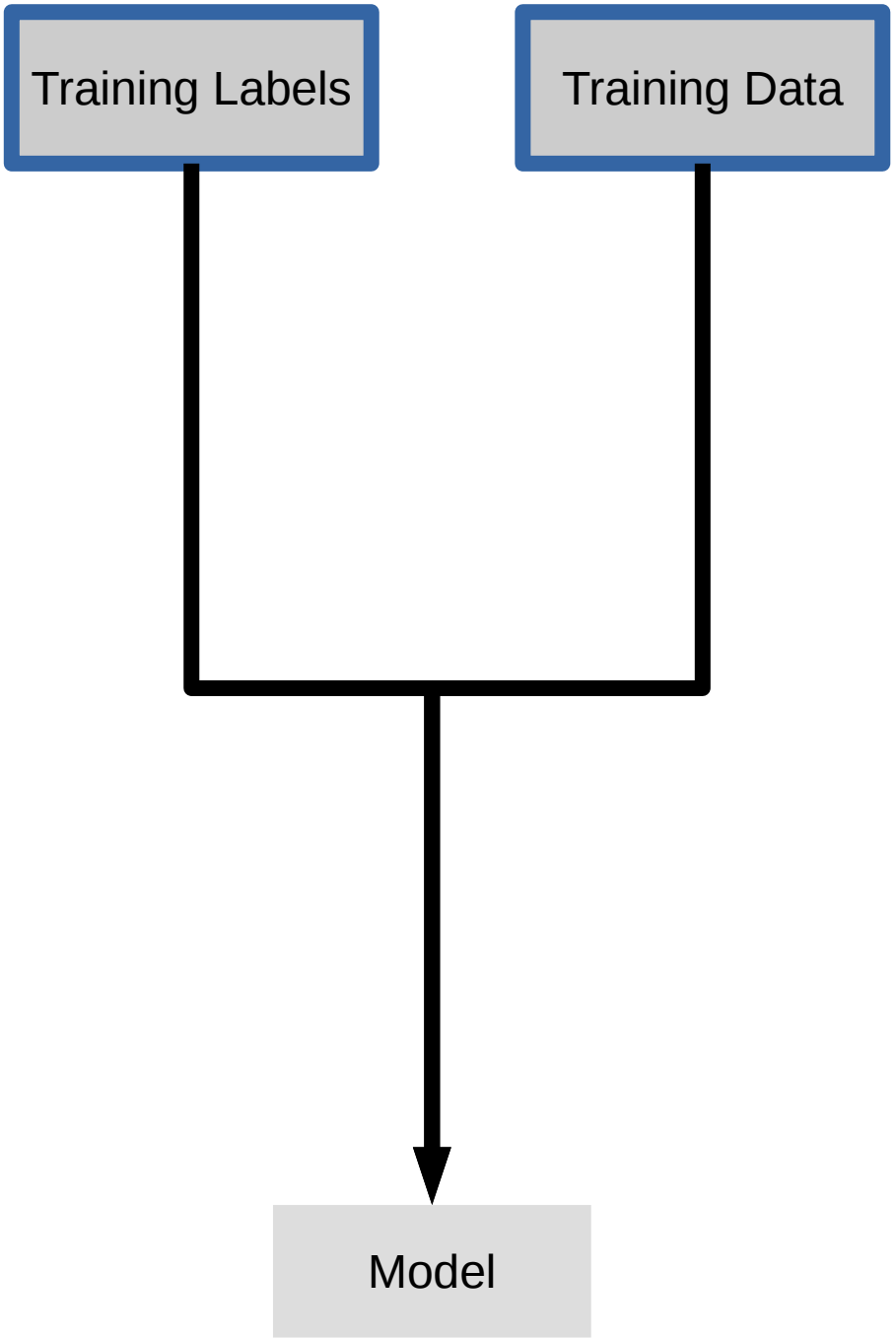
SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	SVC(C=0.1, gamma=0.001)	SVC(C=1, gamma=0.001)	SVC(C=10, gamma=0.001)
SVC(C=0.001, gamma=0.01)	SVC(C=0.01, gamma=0.01)	SVC(C=0.1, gamma=0.01)	SVC(C=1, gamma=0.01)	SVC(C=10, gamma=0.01)

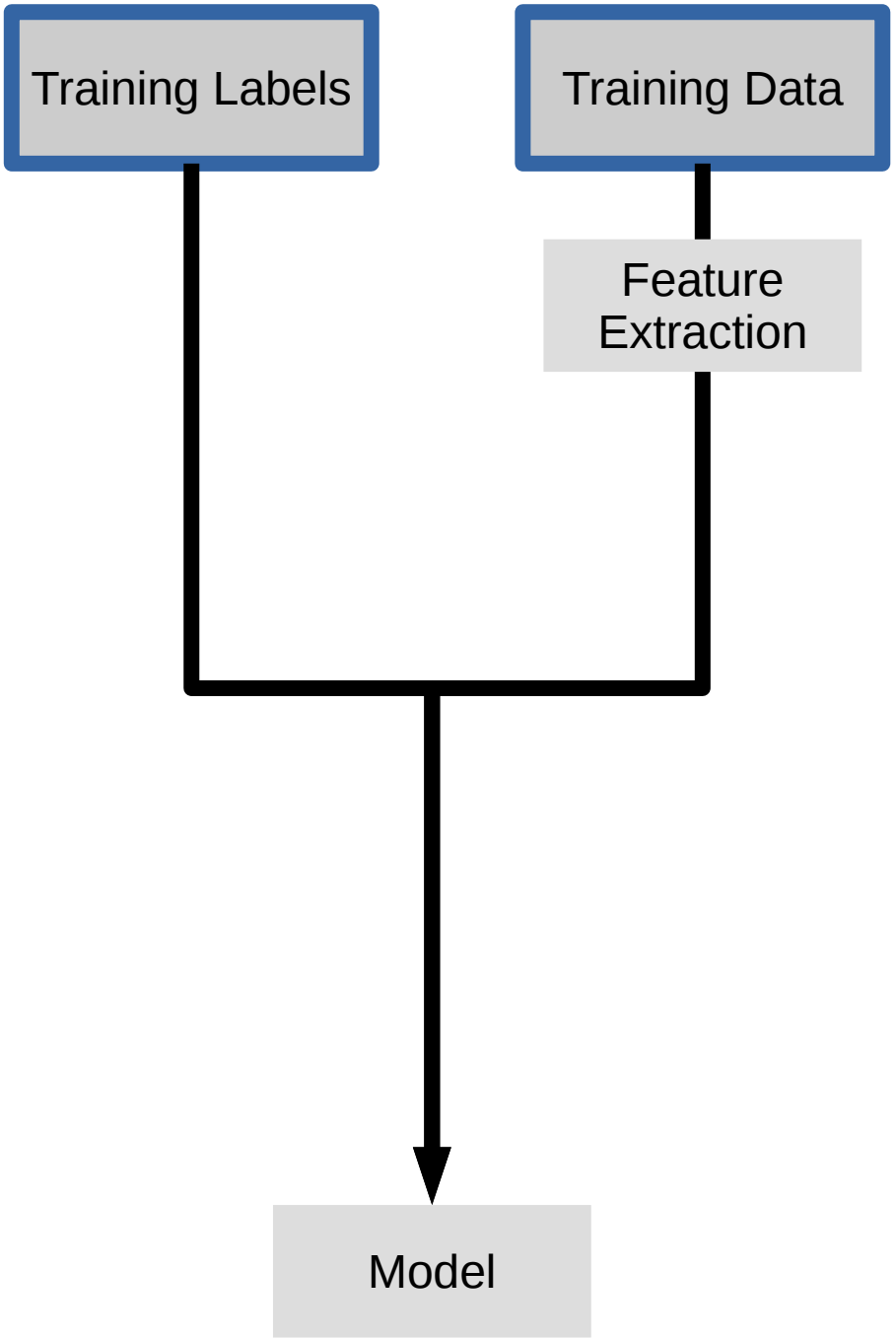
SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	SVC(C=0.1, gamma=0.001)	SVC(C=1, gamma=0.001)	SVC(C=10, gamma=0.001)
SVC(C=0.001, gamma=0.01)	SVC(C=0.01, gamma=0.01)	SVC(C=0.1, gamma=0.01)	SVC(C=1, gamma=0.01)	SVC(C=10, gamma=0.01)
SVC(C=0.001, gamma=0.1)	SVC(C=0.01, gamma=0.1)	SVC(C=0.1, gamma=0.1)	SVC(C=1, gamma=0.1)	SVC(C=10, gamma=0.1)

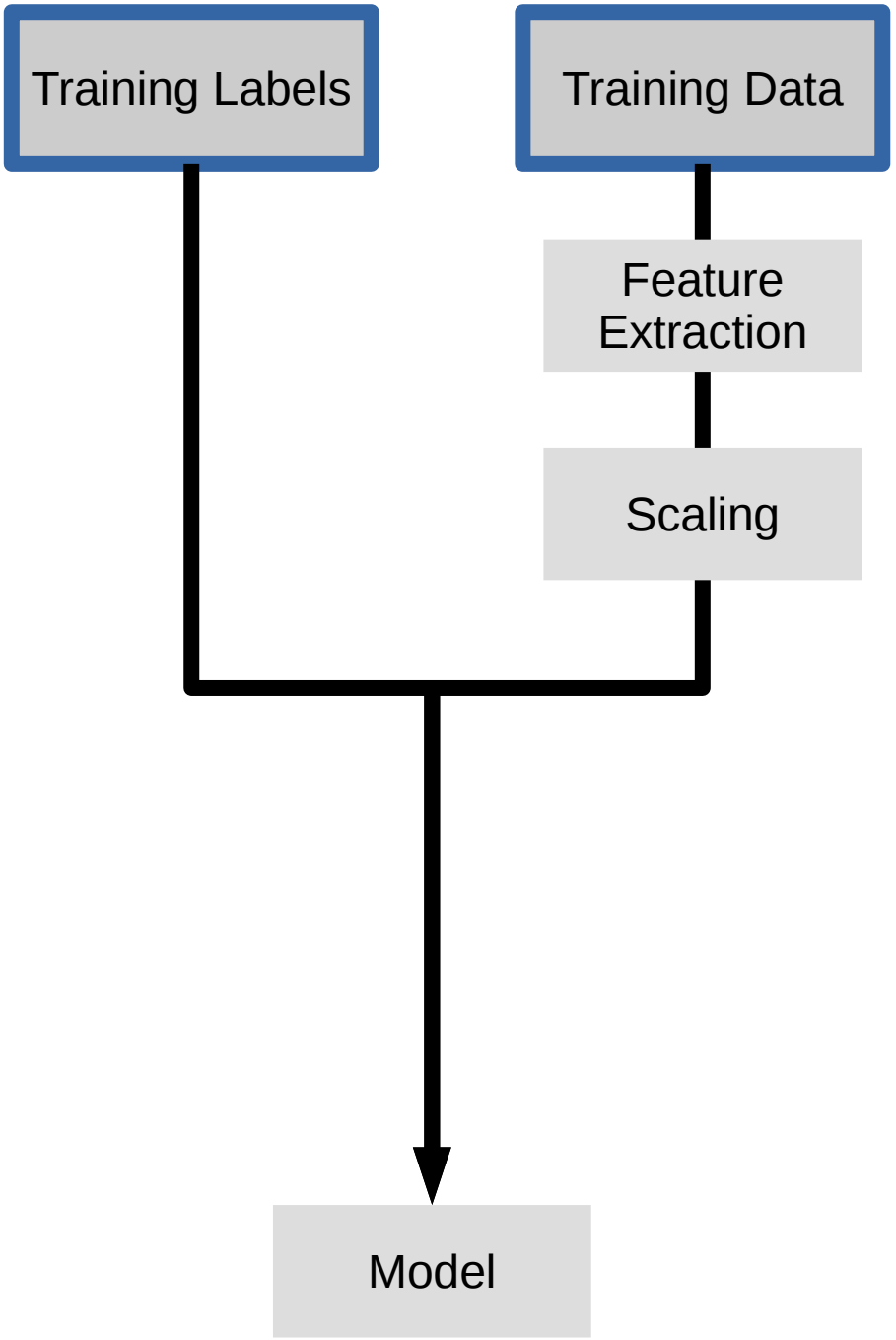
SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	SVC(C=0.1, gamma=0.001)	SVC(C=1, gamma=0.001)	SVC(C=10, gamma=0.001)
SVC(C=0.001, gamma=0.01)	SVC(C=0.01, gamma=0.01)	SVC(C=0.1, gamma=0.01)	SVC(C=1, gamma=0.01)	SVC(C=10, gamma=0.01)
SVC(C=0.001, gamma=0.1)	SVC(C=0.01, gamma=0.1)	SVC(C=0.1, gamma=0.1)	SVC(C=1, gamma=0.1)	SVC(C=10, gamma=0.1)
SVC(C=0.001, gamma=1)	SVC(C=0.01, gamma=1)	SVC(C=0.1, gamma=1)	SVC(C=1, gamma=1)	SVC(C=10, gamma=1)
SVC(C=0.001, gamma=10)	SVC(C=0.01, gamma=10)	SVC(C=0.1, gamma=10)	SVC(C=1, gamma=10)	SVC(C=10, gamma=10)

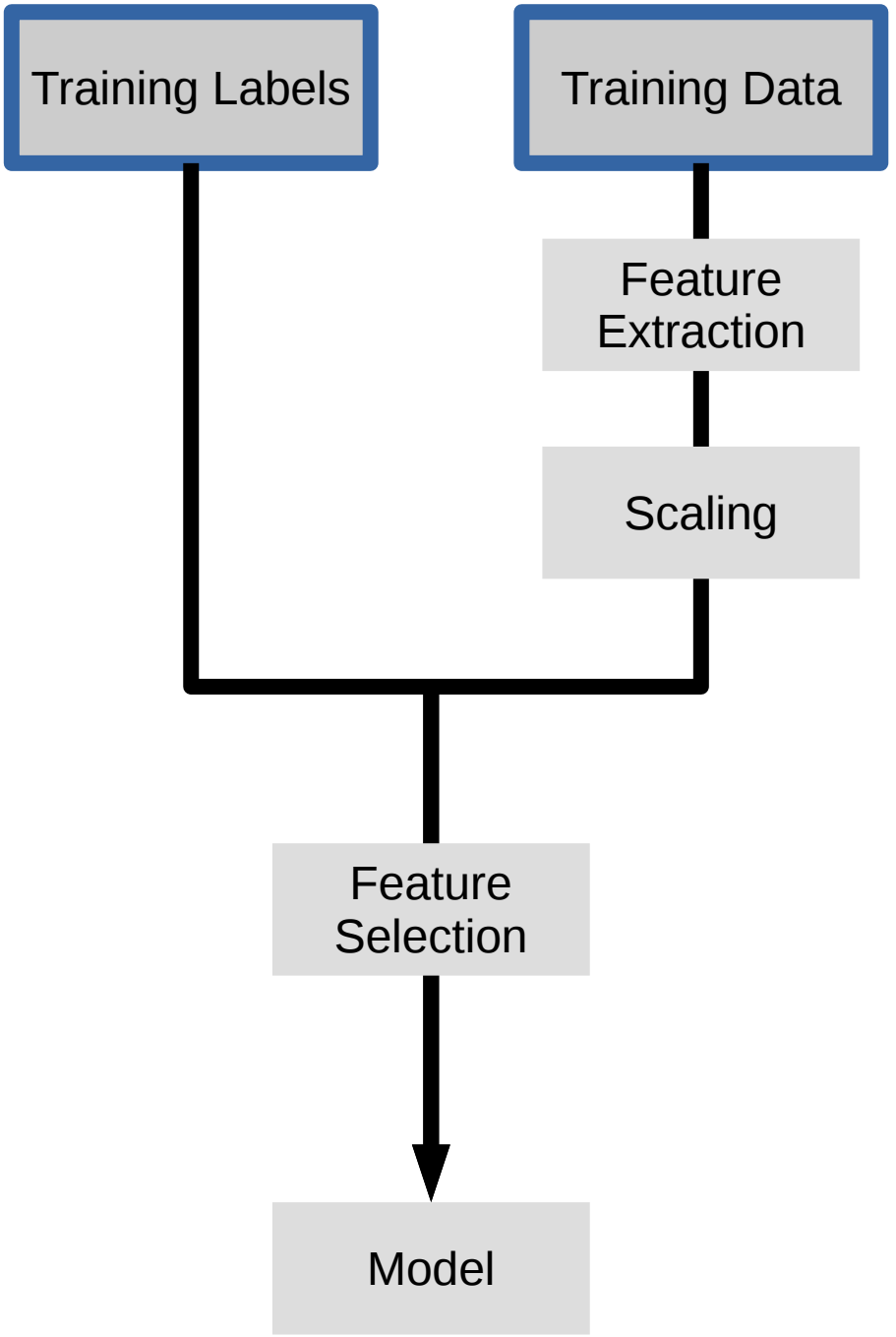
IPython Notebook:
Chapter 4 – Grid Searches

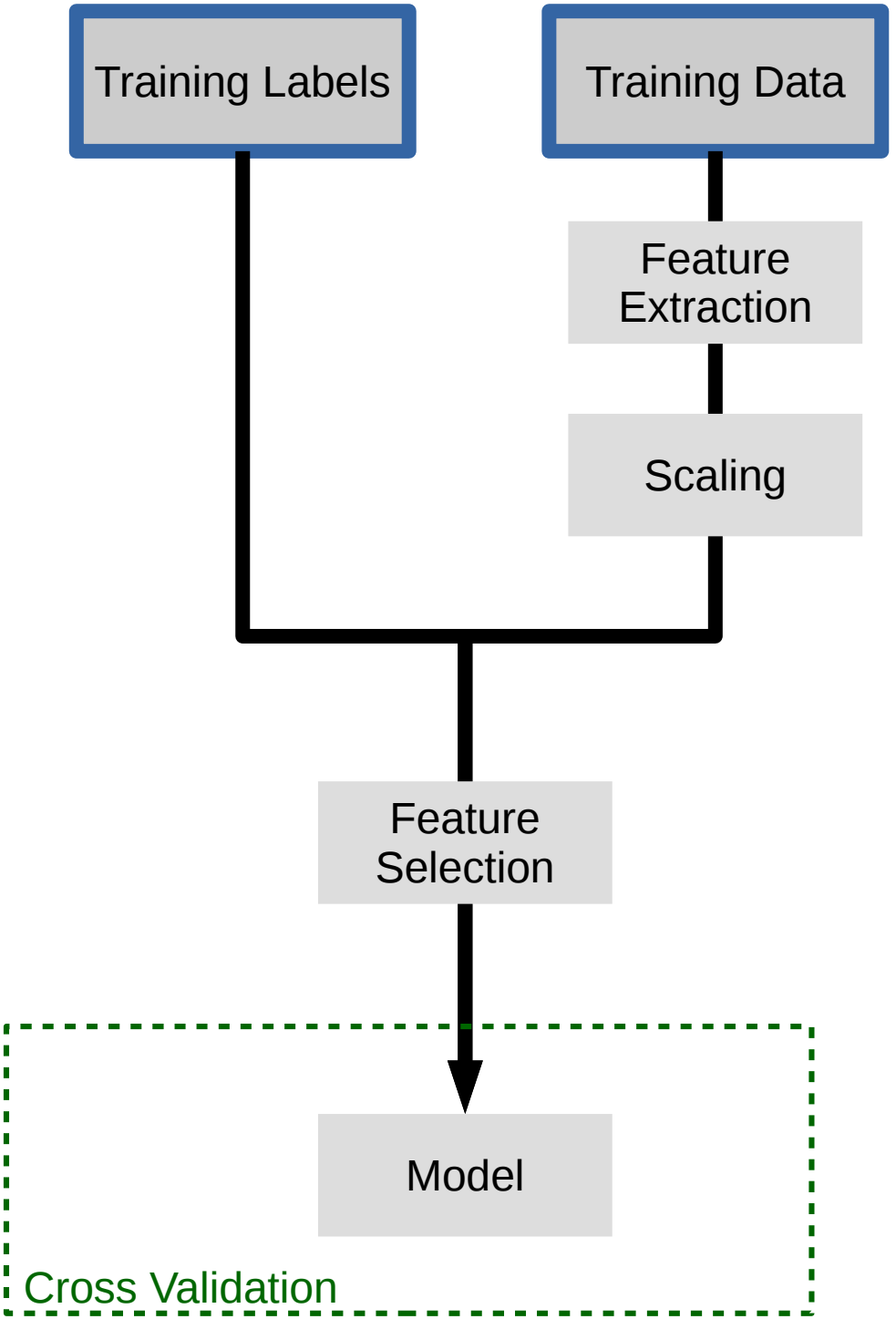

Training Labels
Training Data
Model

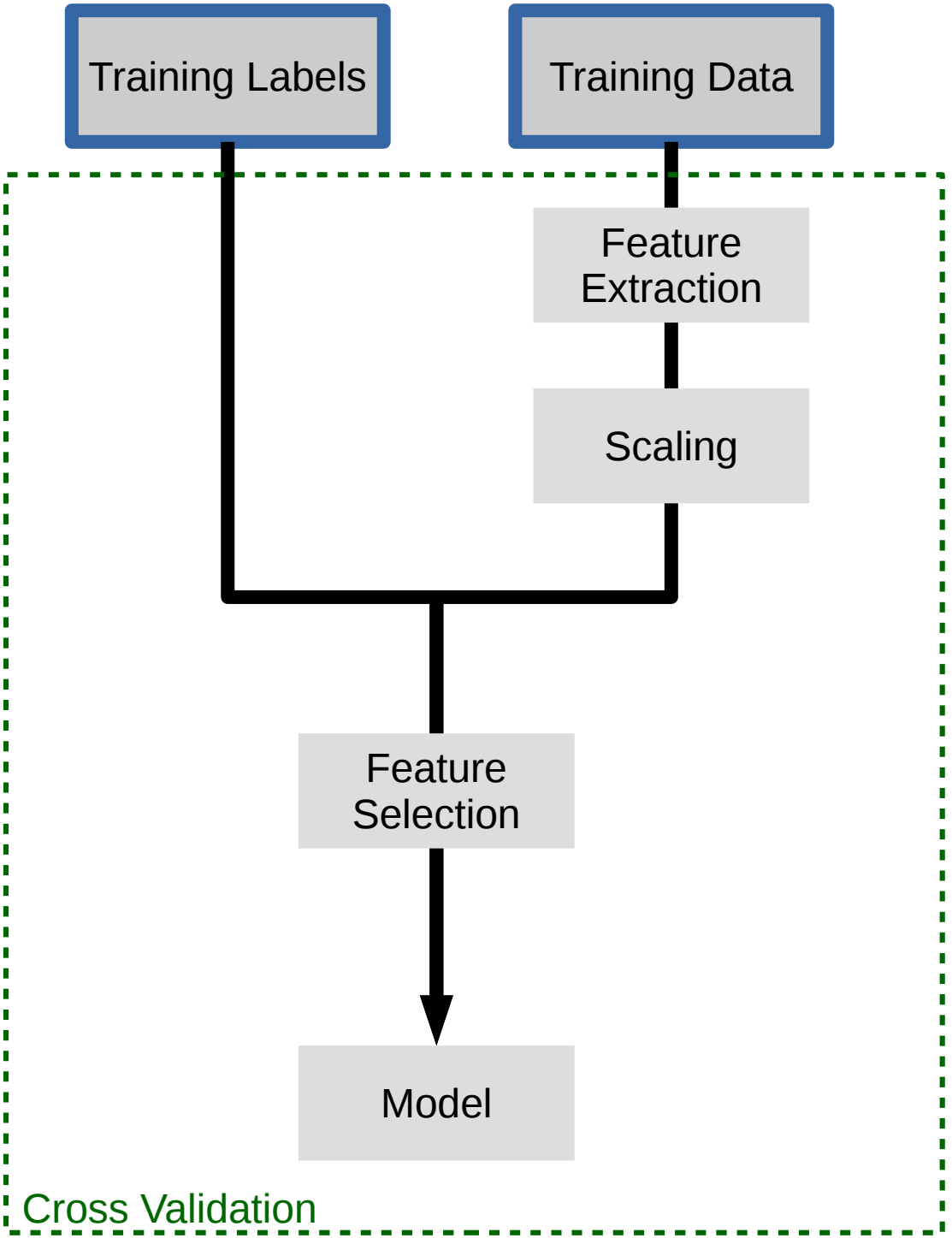

Training Labels
Training Data
Model


Training Labels
Training Data
Feature Extraction
Model


Training Labels
Training Data
Feature Extraction
Scaling
Model


Training Labels
Training Data
Feature Extraction
Scaling
Feature Selection
Model


Training Labels
Training Data
Feature Extraction
Scaling
Feature Selection
Model
Cross Validation


Training Labels
Training Data
Feature Extraction
Scaling
Feature Selection
Model
Cross Validation

IPython Notebook:
Chapter 5 - Preprocessing and Pipelines

Do cross-validation over all steps jointly.
Keep a separate test set until the very end.

Bag Of Word Representations

`CountVectorizer / TfidfVectorizer`

Bag Of Word Representations

`CountVectorizer / TfidfVectorizer`

`“You better call Kenny Loggins”`

Bag Of Word Representations

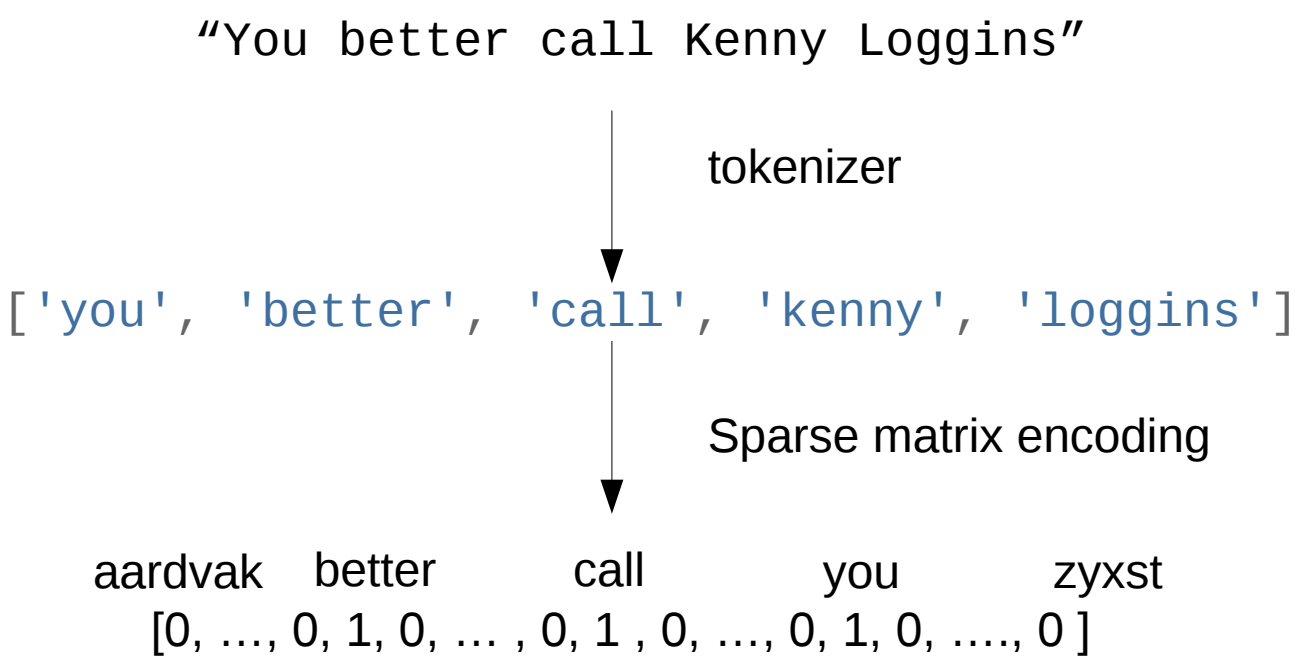
`CountVectorizer / TfidfVectorizer`

`“You better call Kenny Loggins”`


```
['you', 'better', 'call', 'kenny', 'loggins']
```

Bag Of Word Representations

`CountVectorizer / TfidfVectorizer`



Application: Insult detection

Application: Insult detection

```
i really don't understand your point. It seems
that you are mixing apples and oranges.
```

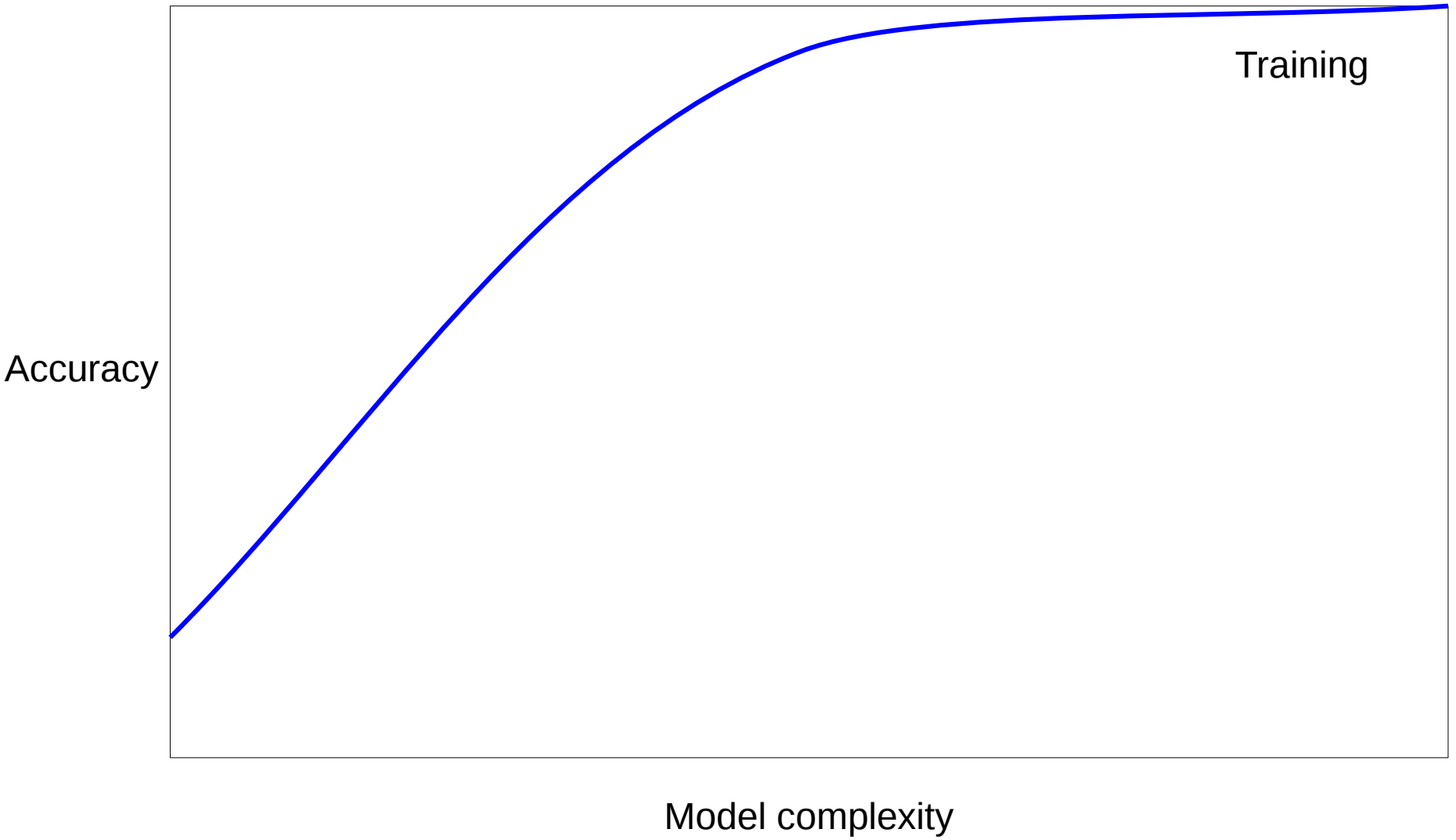
Application: Insult detection

i really don't understand your point. It seems that you are mixing apples and oranges.

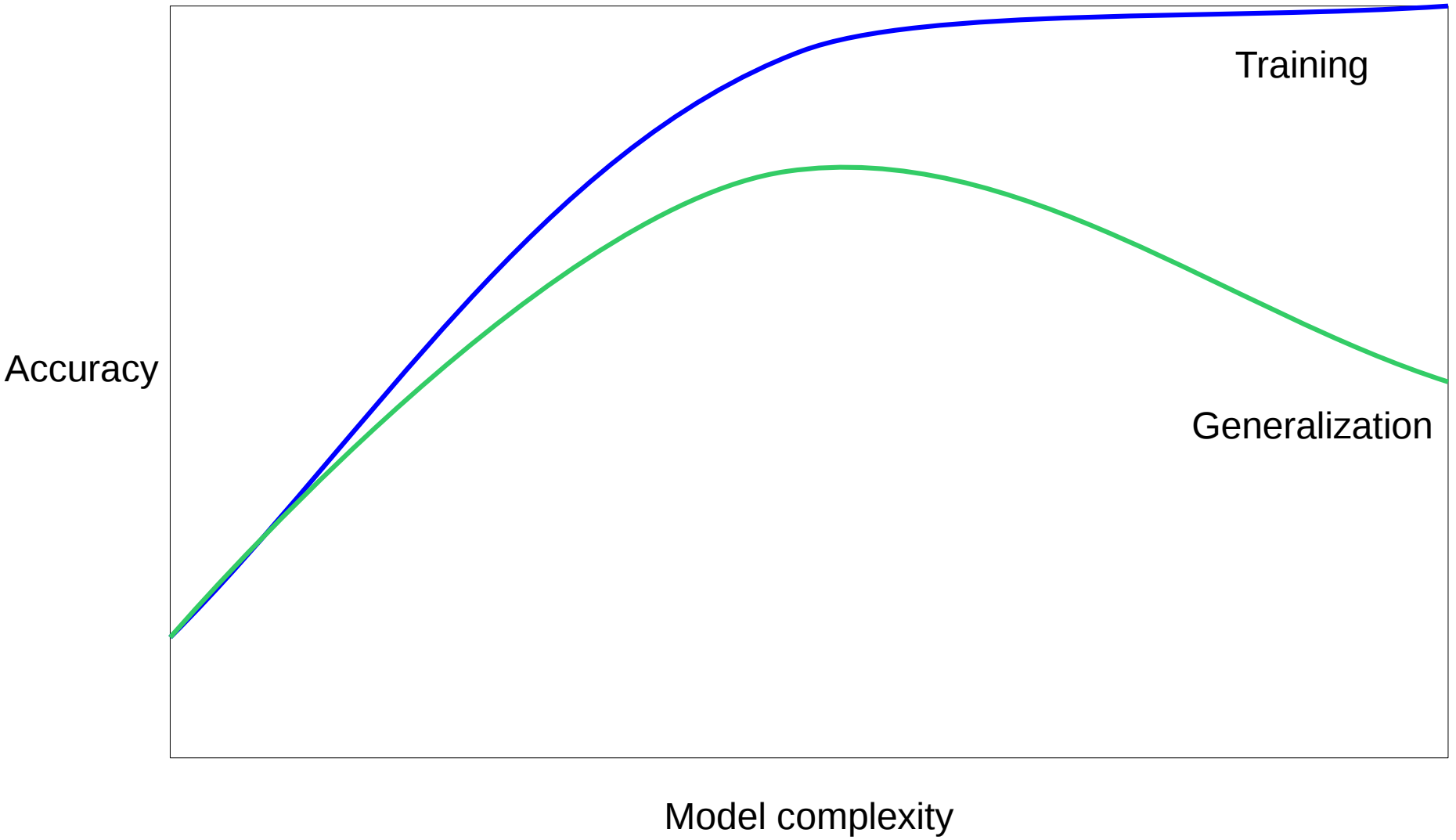
Clearly you're a fucktard.

IPython Notebook:
Chapter 6 - Working With Text Data

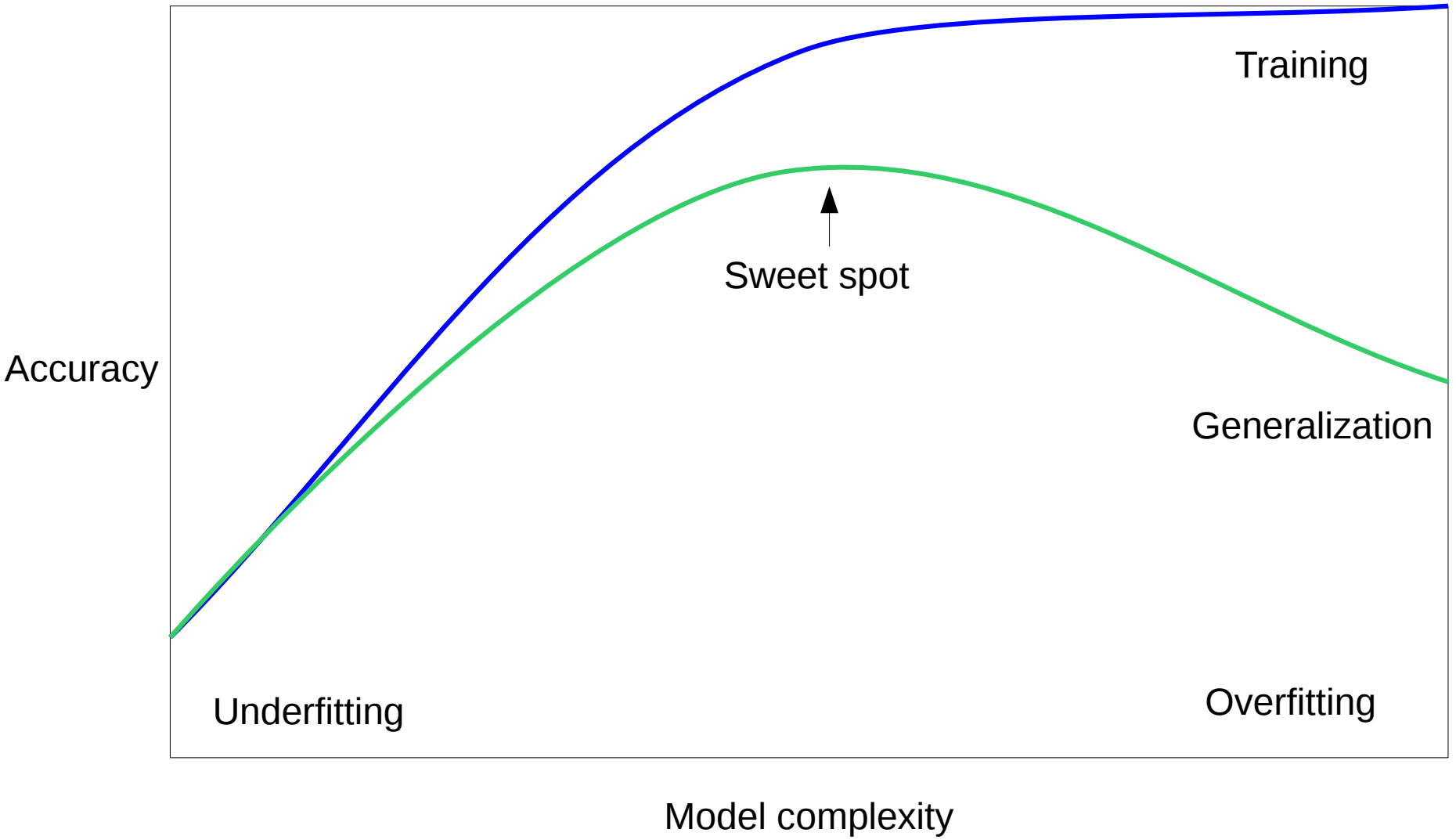
Overfitting and Underfitting



Overfitting and Underfitting



Overfitting and Underfitting

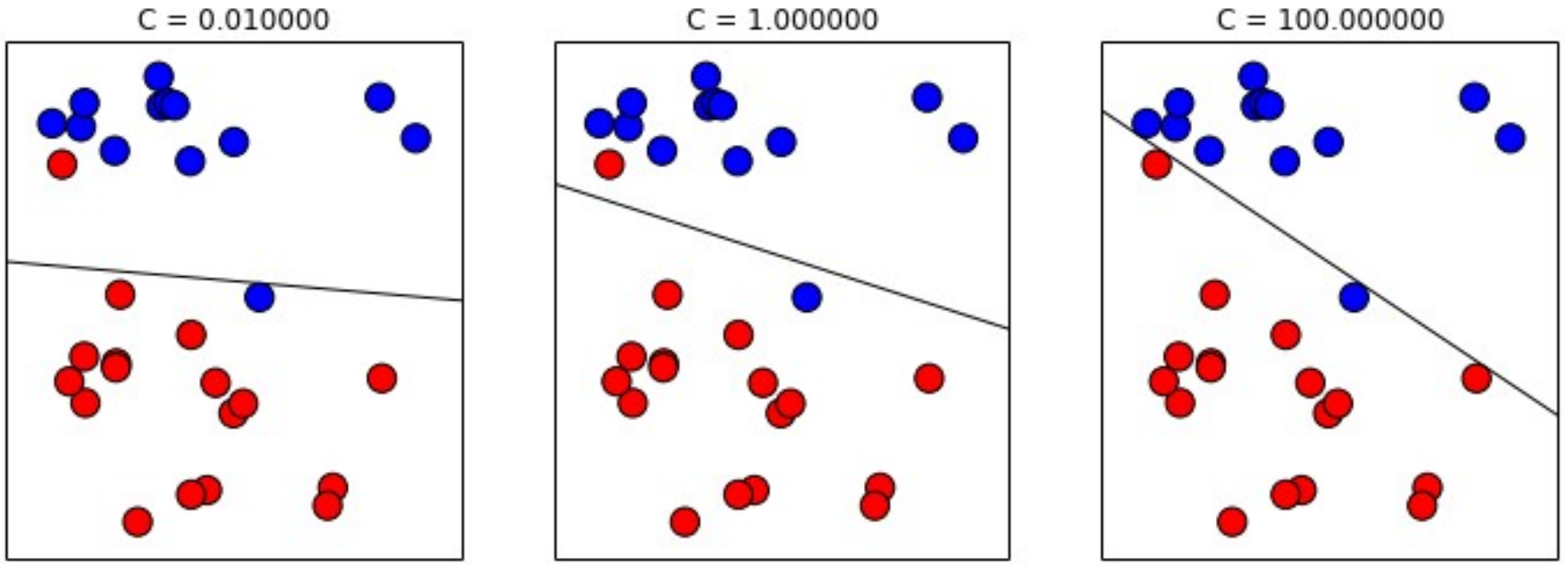


Linear SVM

$$\hat{y} = \text{sign}(w_0 + \sum_i w_i x_i)$$

Linear SVM

$$\hat{y} = \text{sign}(w_0 + \sum_i w_i x_i)$$



(RBF) Kernel SVM

$$\hat{y} = \text{sign}(\alpha_0 + \sum_j \alpha_j y_j k(\mathbf{x}^{(\mathbf{j})}, \mathbf{x}))$$

(RBF) Kernel SVM

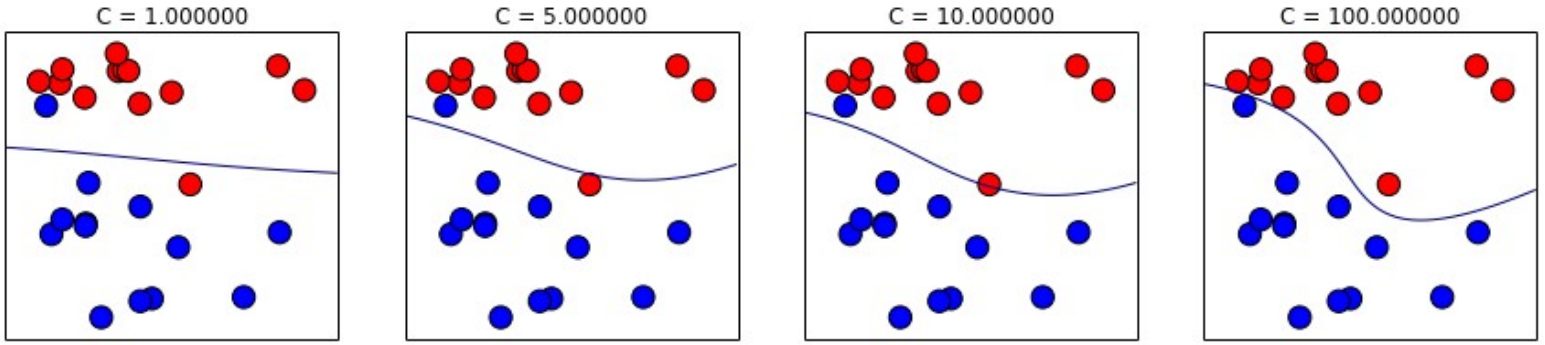
$$\hat{y} = \text{sign}(\alpha_0 + \sum_j \alpha_j y_j k(\mathbf{x^{(j)}}, \mathbf{x}))$$

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2)$$

(RBF) Kernel SVM

$$\hat{y} = \text{sign}(\alpha_0 + \sum_j \alpha_j y_j k(\mathbf{x}^{(\mathbf{j})}, \mathbf{x}))$$

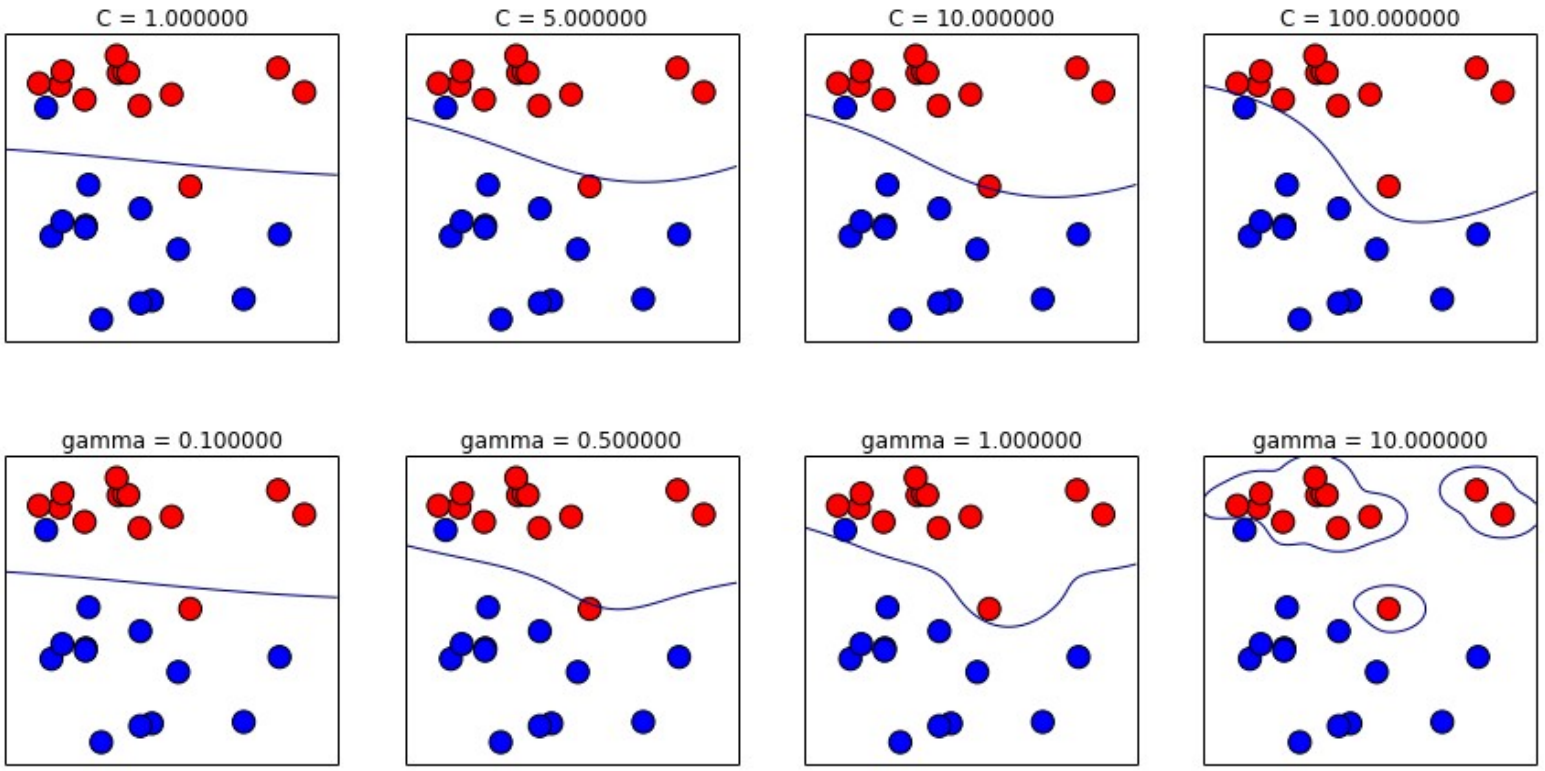
$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2)$$



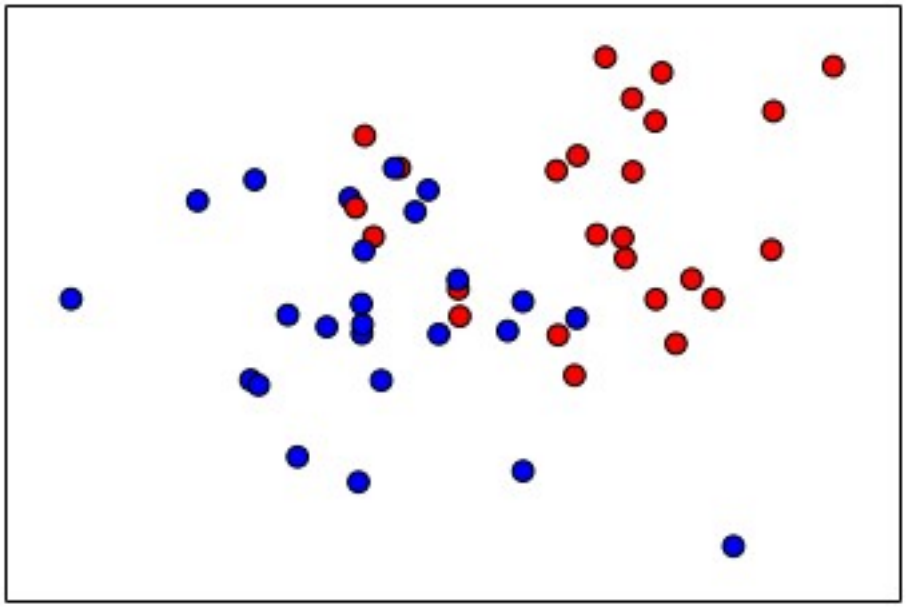
(RBF) Kernel SVM

$$\hat{y} = \text{sign}(\alpha_0 + \sum_j \alpha_j y_j k(\mathbf{x}^{(\mathbf{j})}, \mathbf{x}))$$

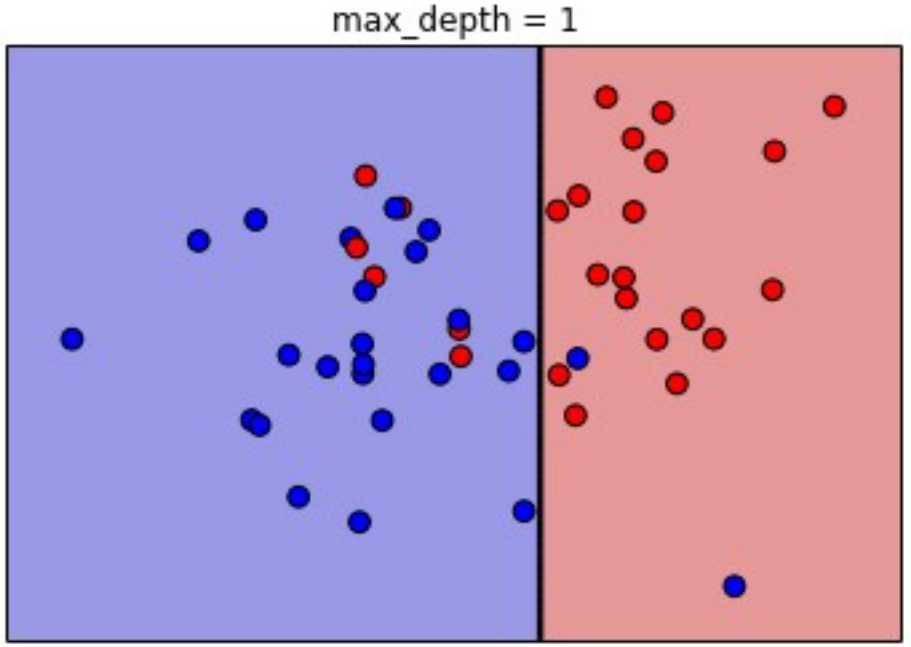
$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2)$$

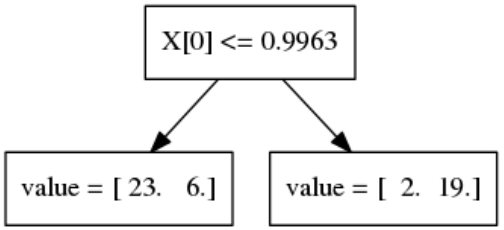


Decision Trees

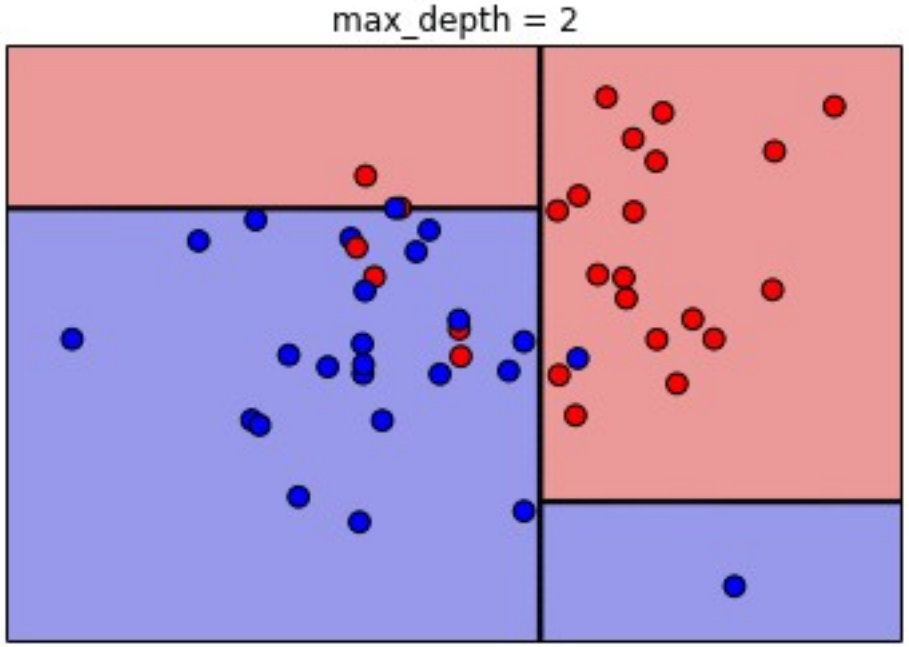


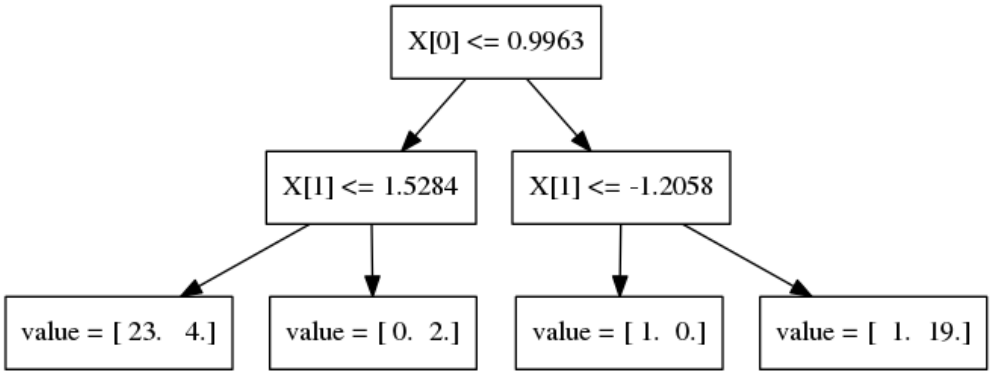
Decision Trees



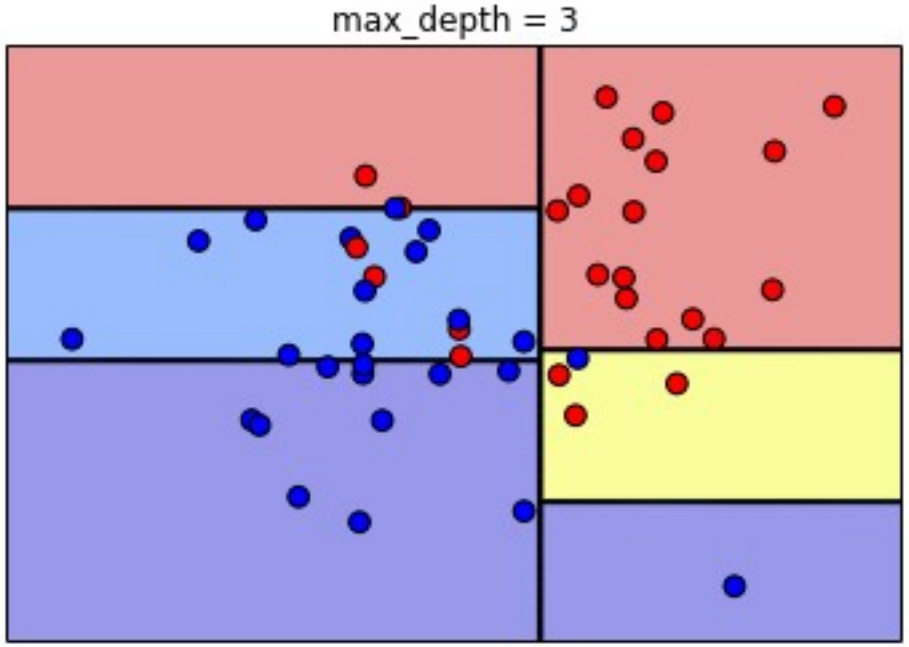


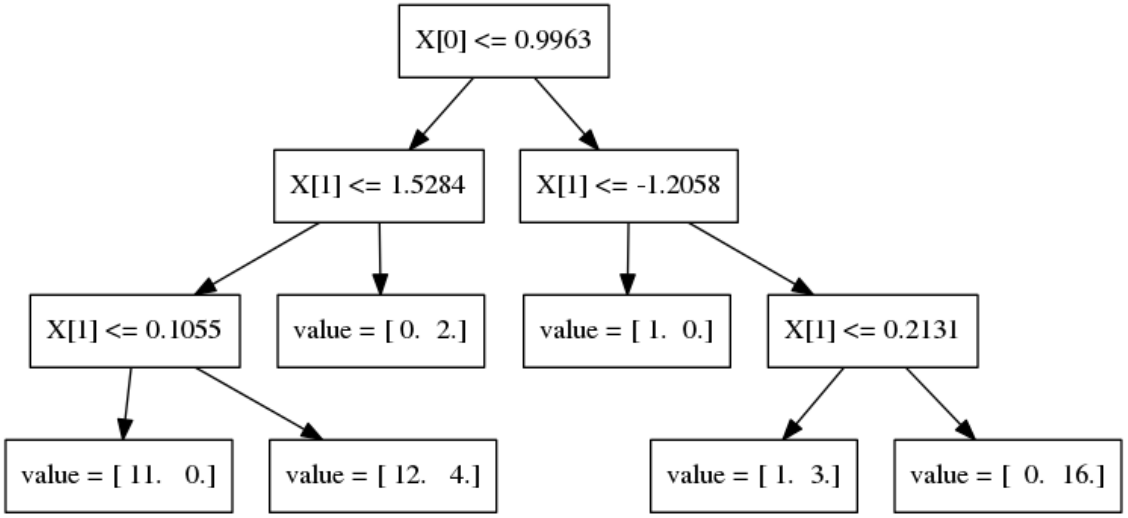
Decision Trees



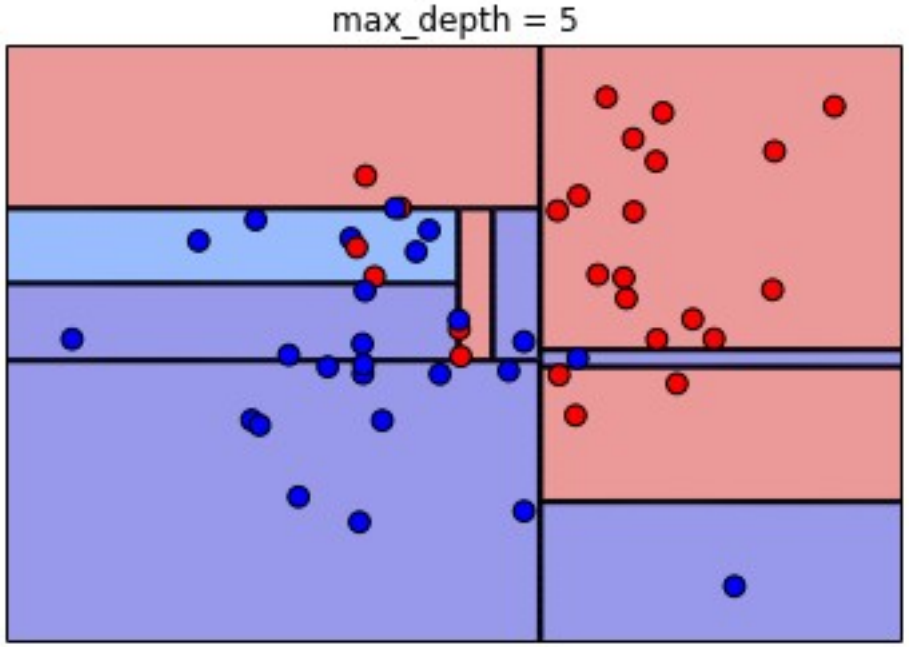


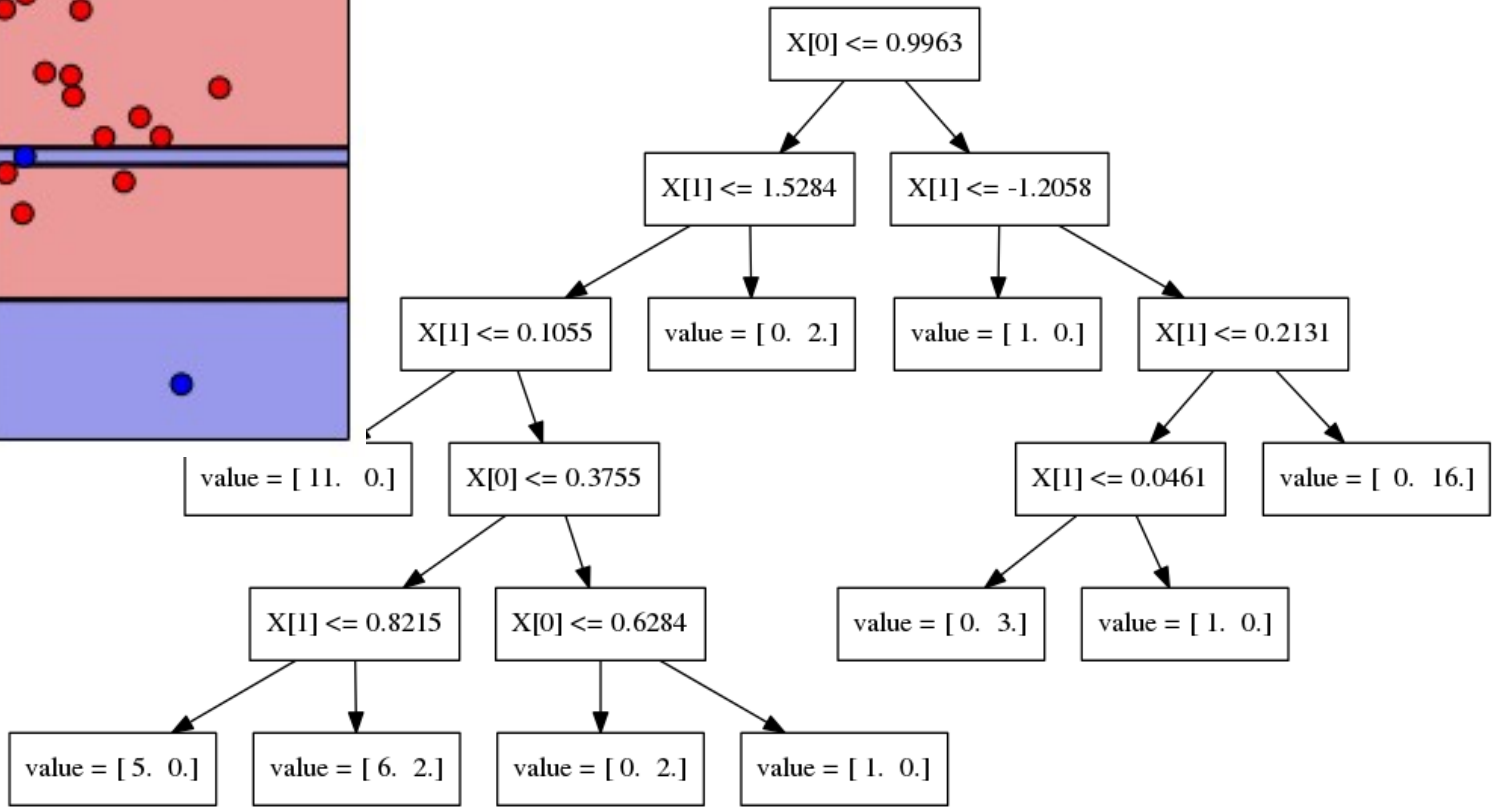
Decision Trees



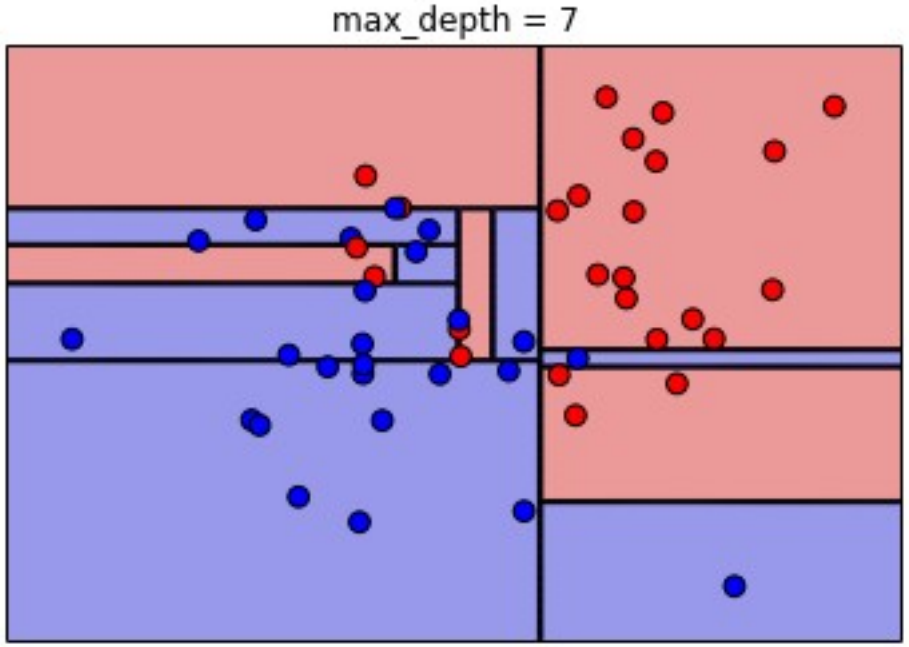


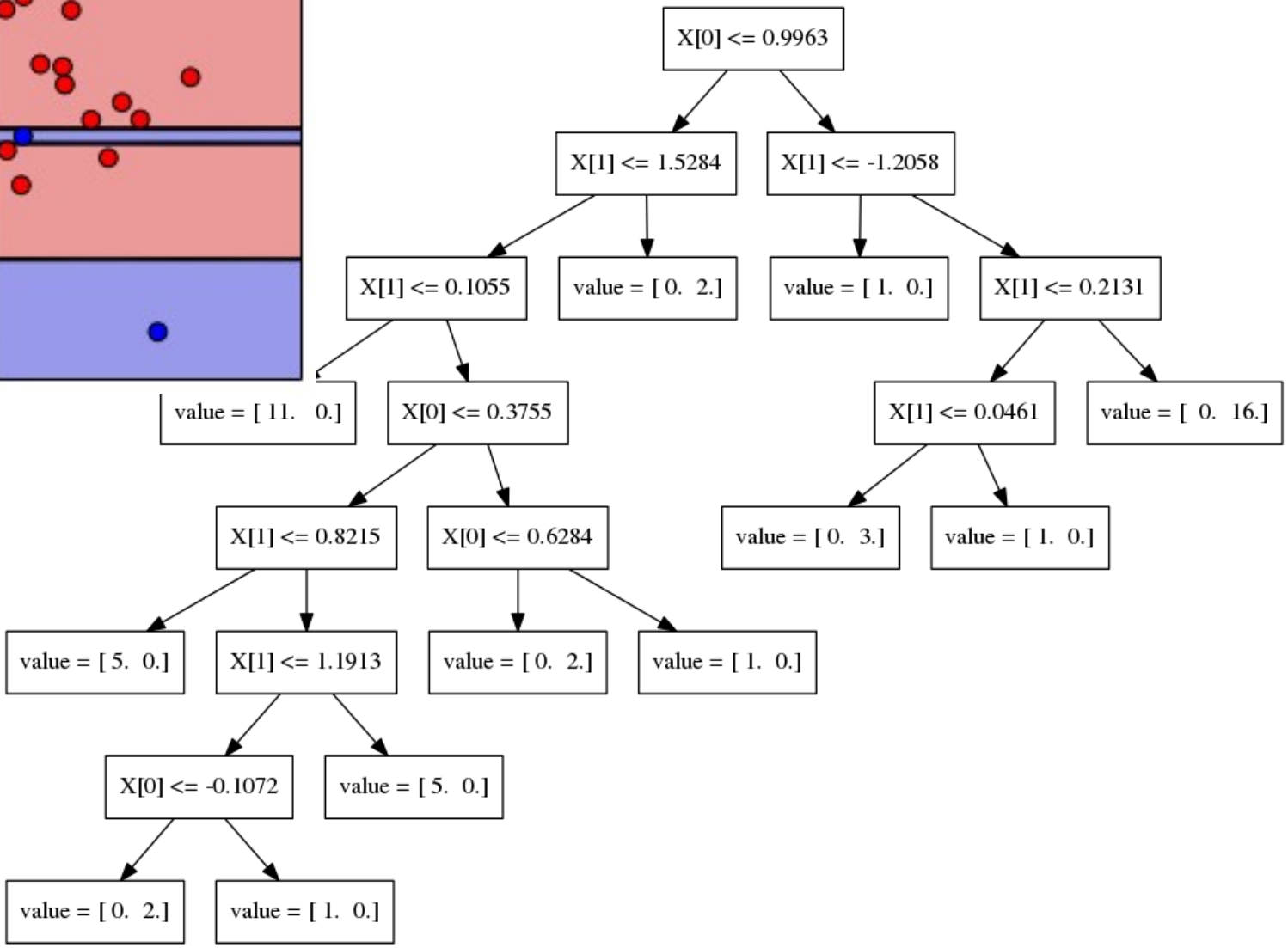
Decision Trees



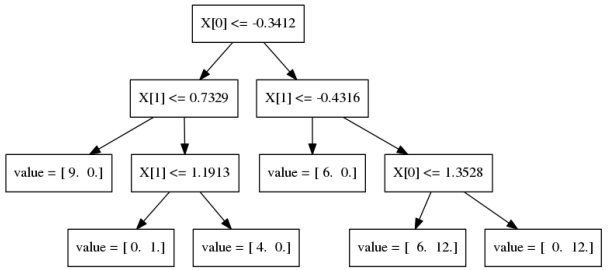


Decision Trees

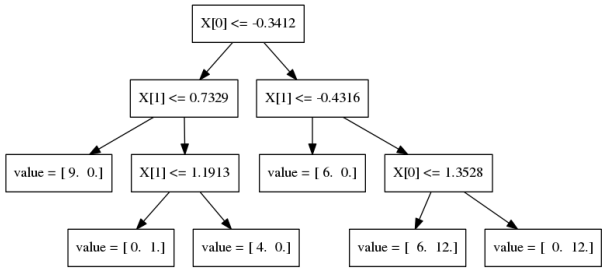


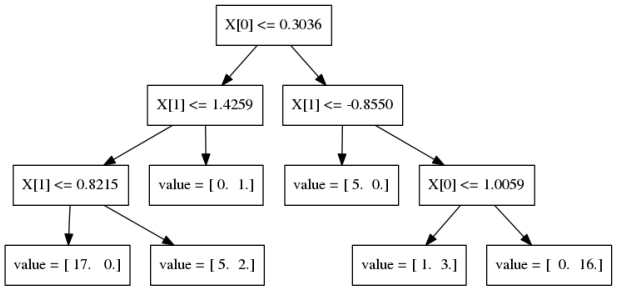


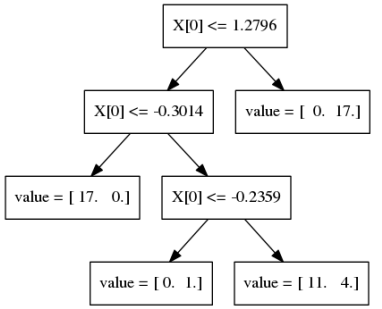
Random Forests



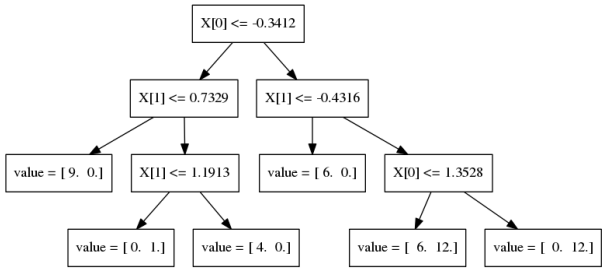
Random Forests

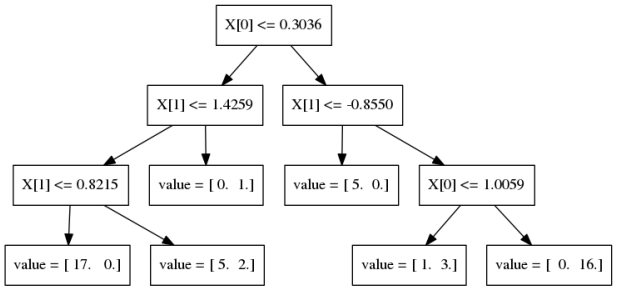


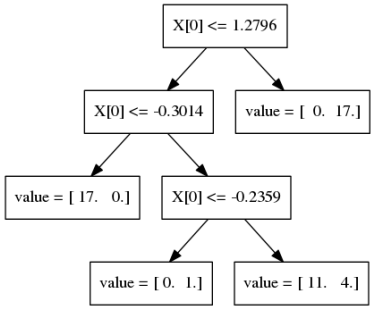


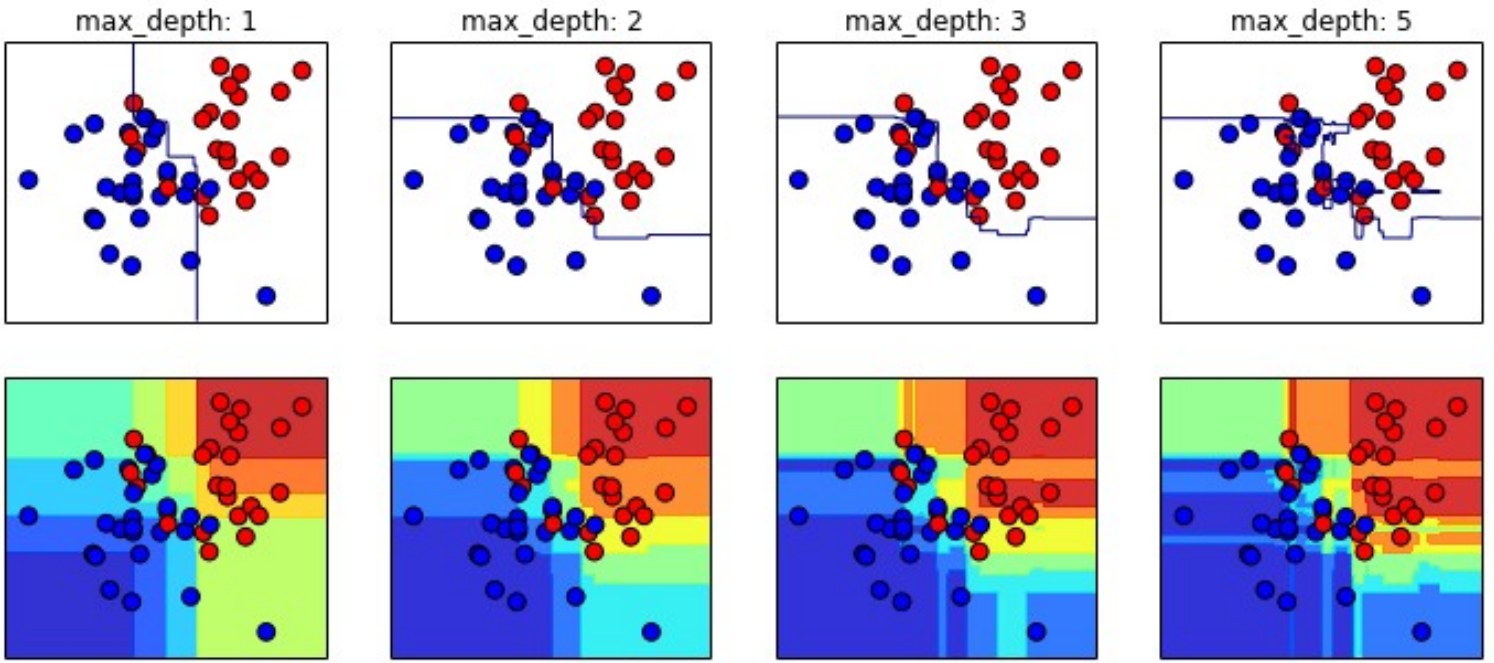


Random Forests









Documentation of scikit-learn 0.16-git

Quick Start

A very short introduction into machine learning problems and how to solve them using scikit-learn. Introduced basic concepts and conventions.

User Guide

The main documentation. This contains an in-depth description of all algorithms and how to apply them.

Other Versions

- scikit-learn 0.15 (stable)
- scikit-learn 0.16 (development)
- scikit-learn 0.14
- scikit-learn 0.13
- scikit-learn 0.12
- Older versions ▲

Tutorials

Useful tutorials for developing a feel for some of scikit-learn's applications in the machine learning field.

API

The exact API of all functions and classes, as given by the docstrings. The API documents expected types and allowed features for all functions, and all parameters available for the algorithms.

Additional Resources

Talks given, slide-sets and other information relevant to scikit-learn.

Contributing

Information on how to contribute. This also contains useful information for advanced users, for example how to build their own estimators.

Flow Chart

A graphical overview of basic areas of machine learning, and guidance which kind of algorithms to use in a given situation.

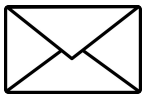
FAQ

Frequently asked questions about the project and contributing.

Thank you for your attention.

@t3kcit

@amueller



importamueller@gmail.com