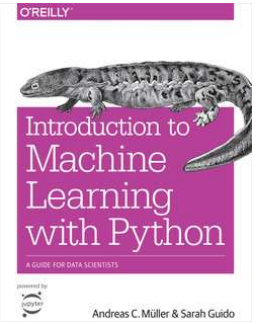
Applied Machine Learning
Cross Validation

BSc course Informatiekunde 2026
https://staff.fnwi.uva.nl/a.visser/education/AML

Arnoud Visser
Intelligent Robotics Lab & Computer Vision Lab
Informatics Institute
Universiteit van Amsterdam
A.Visser@uva.nl

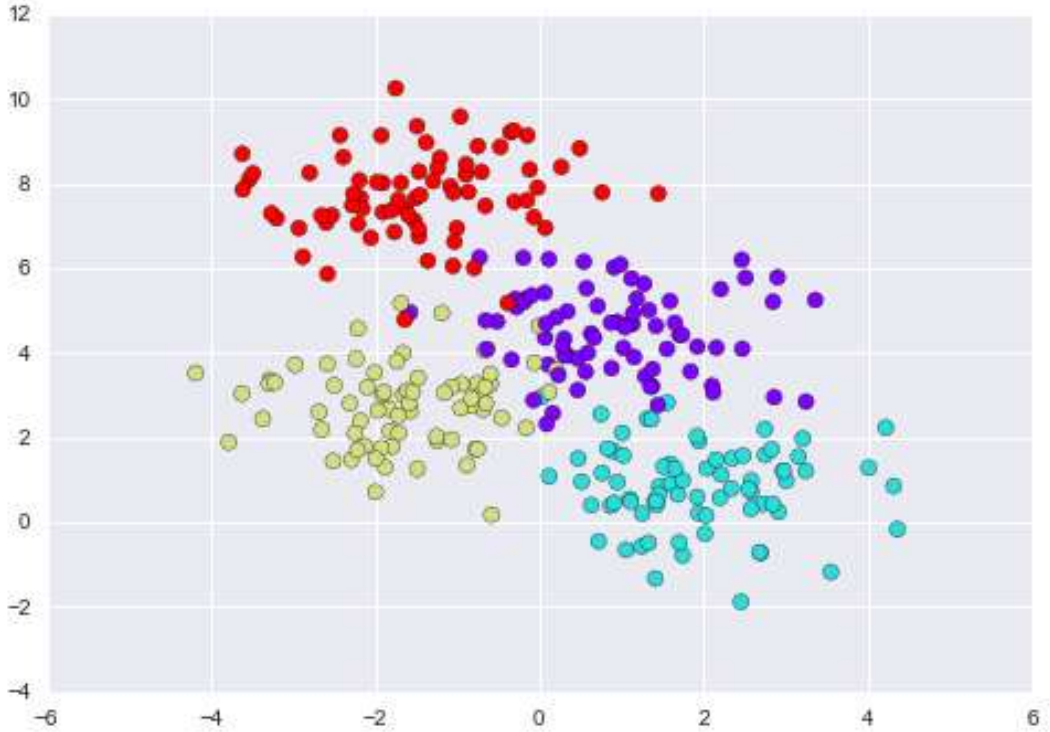
Illustrations courtesy of Maarten Marx, Sarah Guido, Yolanda Hagar, and many others.



Section 5.1

Randomness in model

- Decision Trees will iteratively split the data

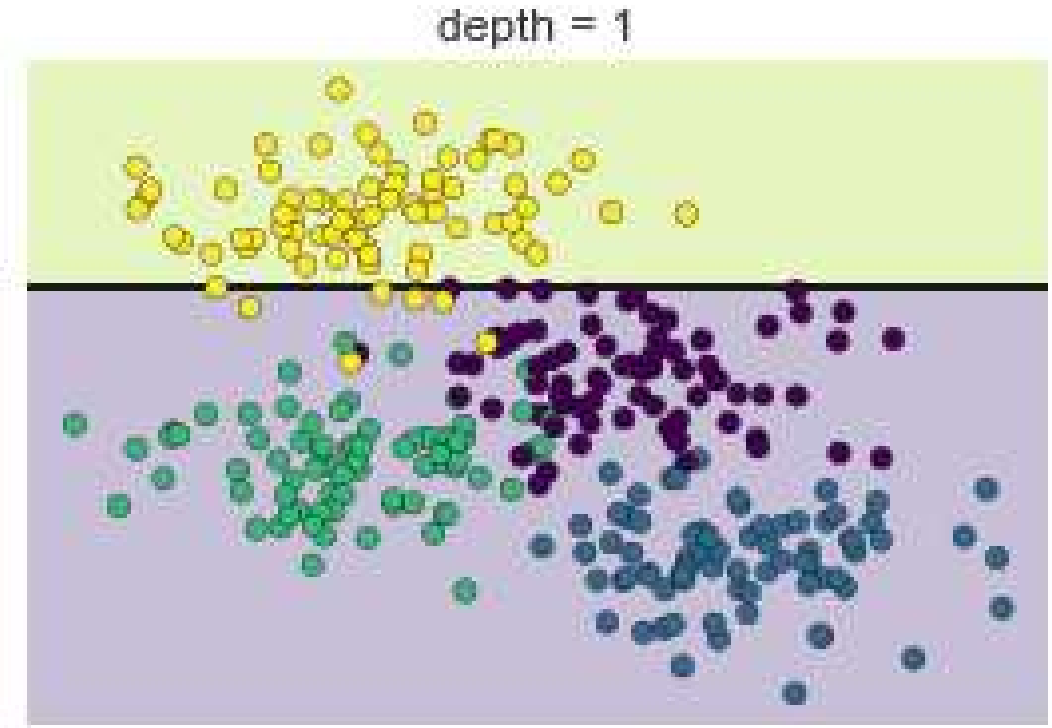


4 Classes

Page 422

Randomness in model

- Decision Trees will iteratively split the data

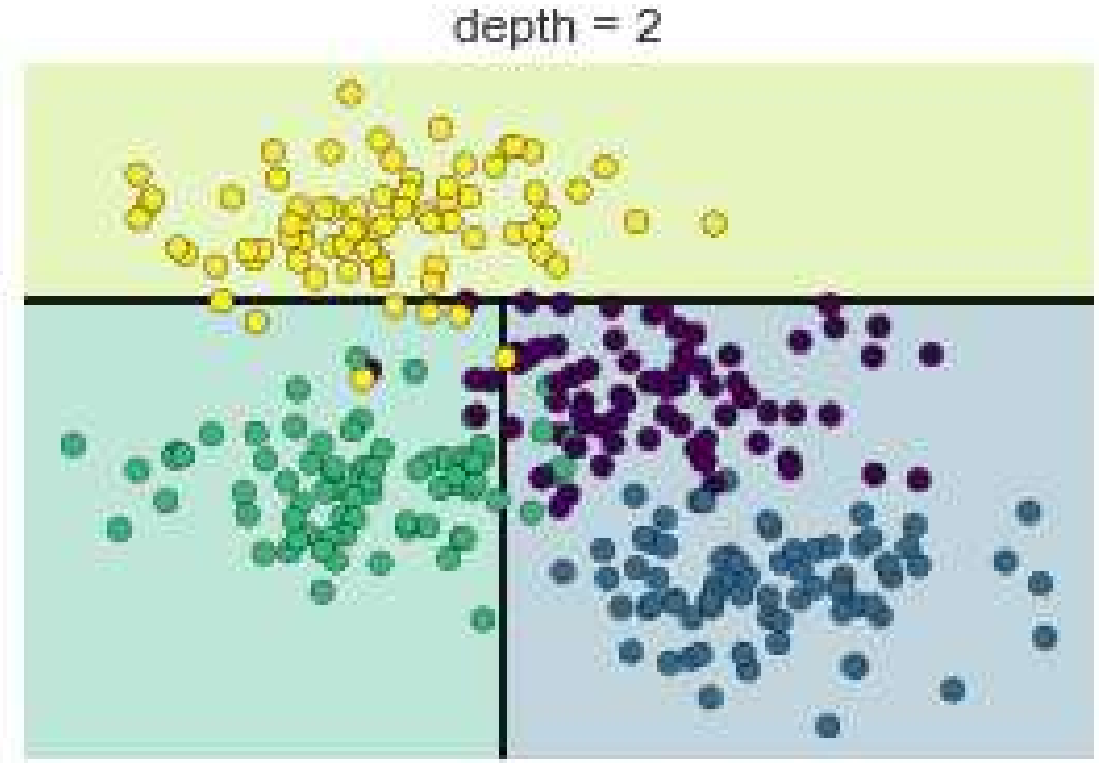


4 Classes

Page 423

Randomness in model

- Decision Trees will iteratively split the data

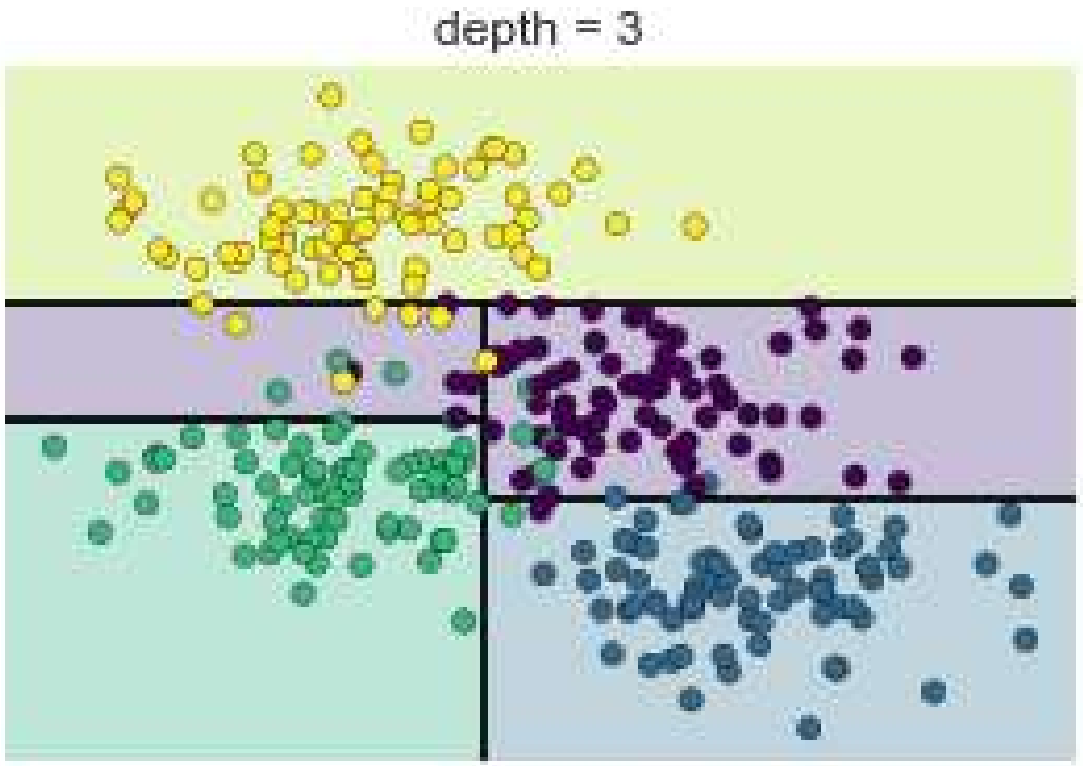


4 Classes

Page 423

Randomness in model

- Decision Trees will iteratively split the data

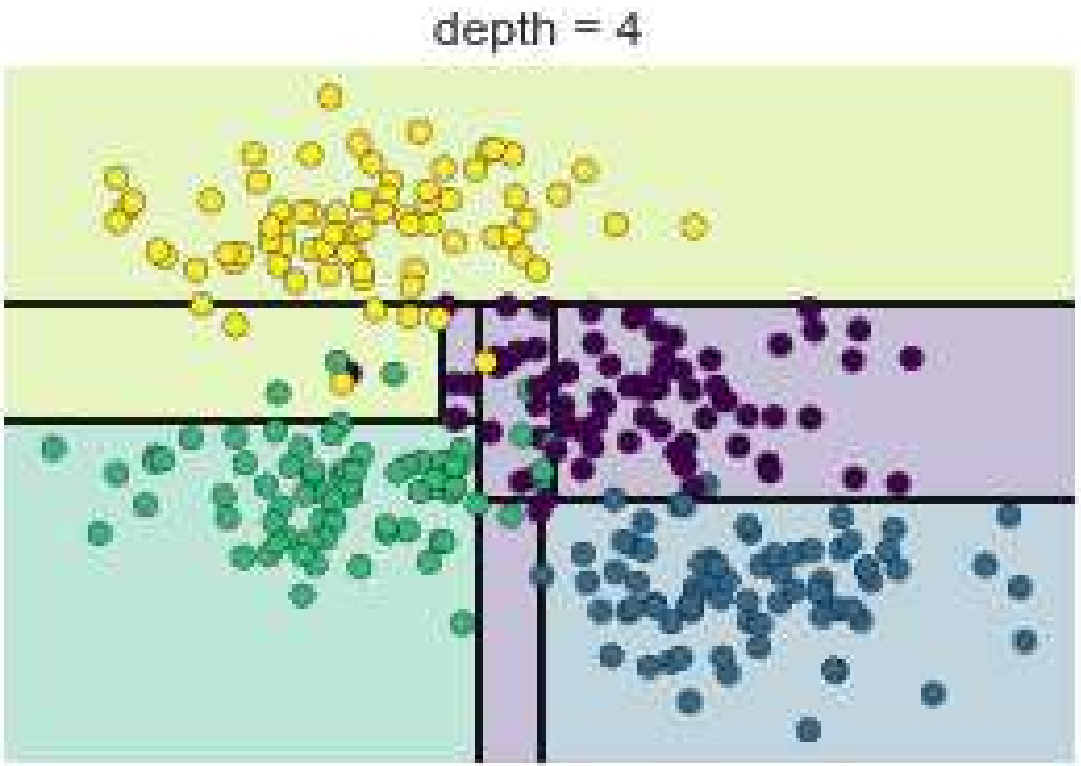


4 Classes

Page 423

Randomness in model

- Decision Trees will iteratively split the data

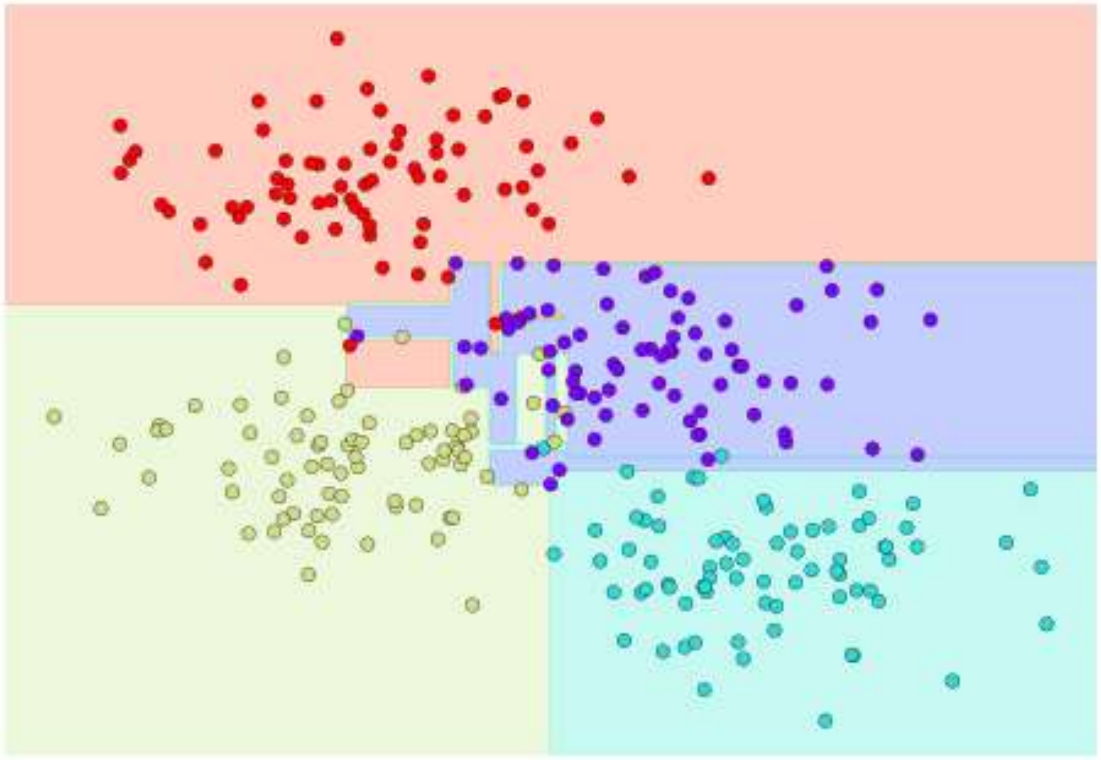


4 Classes

Page 423

Randomness in model

- When the depth increases, strange regions appear

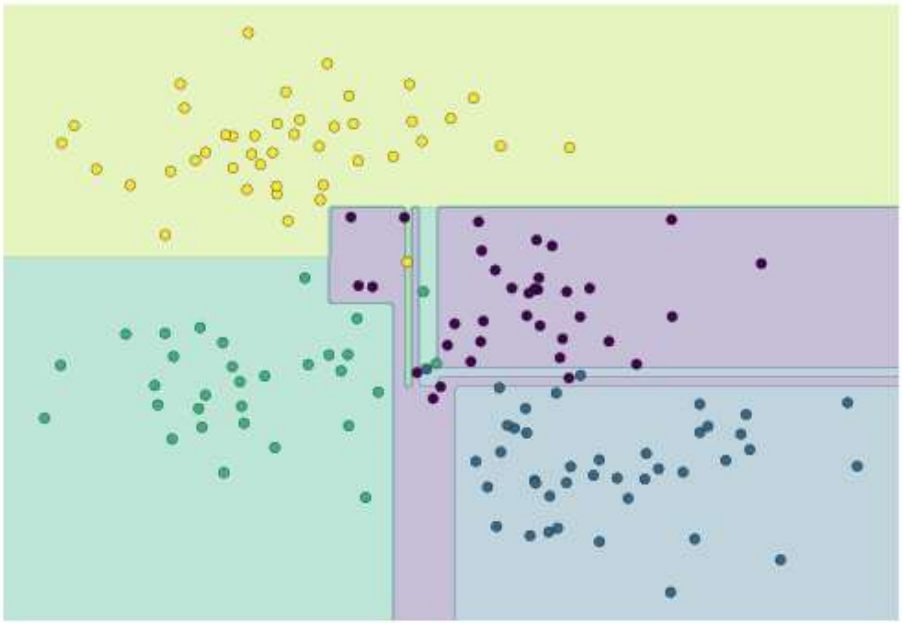


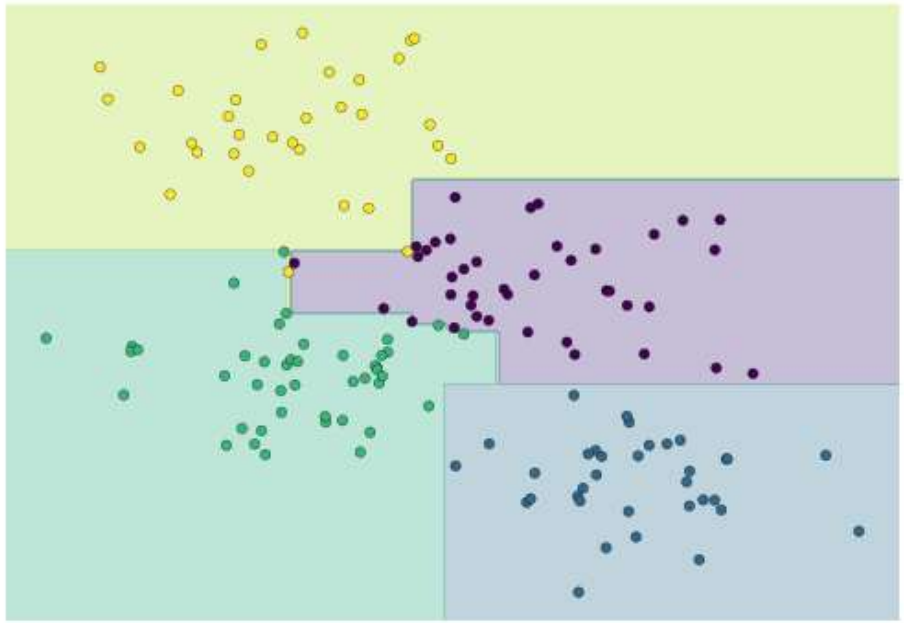
4 Classes

Page 424

Randomness in model

- When the split the data in two,
  two different decision trees are built.



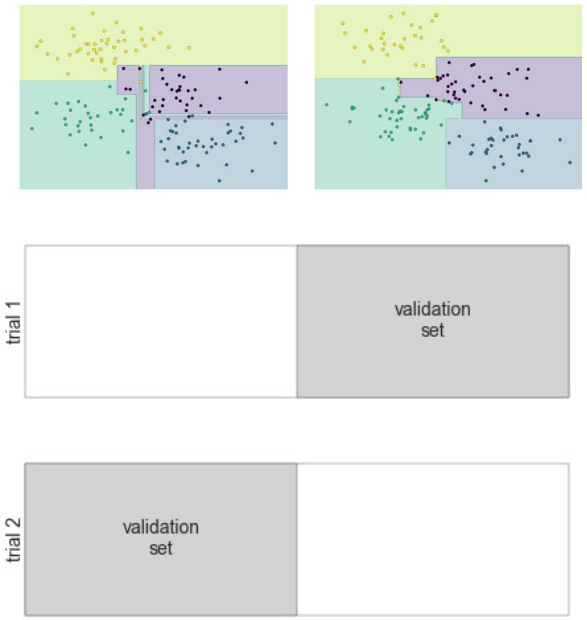


4 Classes

Holdout sets

- split the data in two, train on one, test on the other, and vise versa

Two trails




```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

model = DecisionTreeClassifier()

y2_model = model.fit(X1, y1).predict(X2)
y1_model = model.fit(X2, y2).predict(X1)

accuracy_score(y1, y1_model), accuracy_score(y2, y2_model)
```

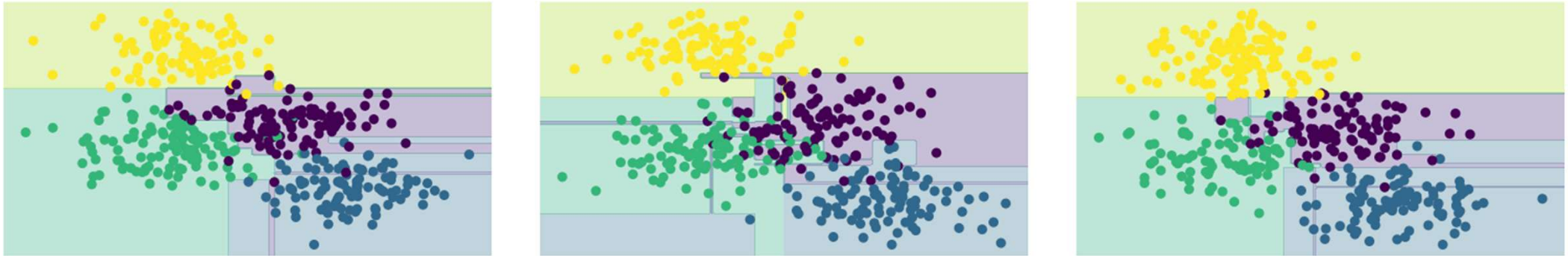
```
(0.92, 0.87)
```

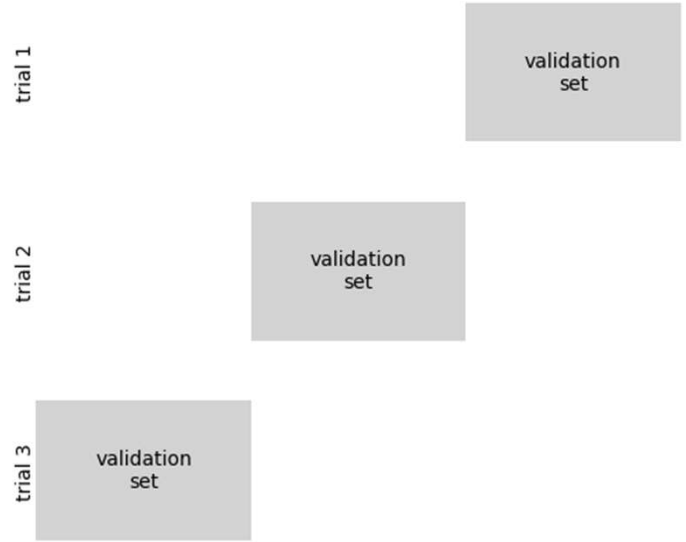
Page 361

K-fold cross-validation

- split the data in three, train on two, test on the one hold-out, and repeat

Three trails






```
from sklearn.model_selection import cross_val_score

model = DecisionTreeClassifier()

scores = cross_val_score(model, X, y, cv=3)
```

```
(0.8675, 0.8925, 0.9275)
```

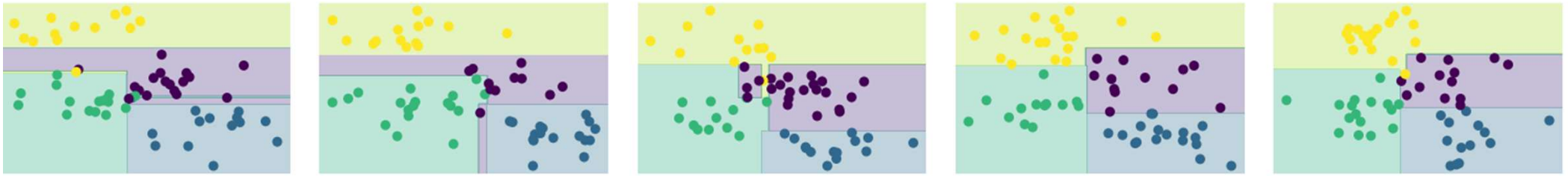
```
print("accuracy = {:.2f} \u00B1 {:.3f}".format(scores.mean(),
                                               scores.std()))
```

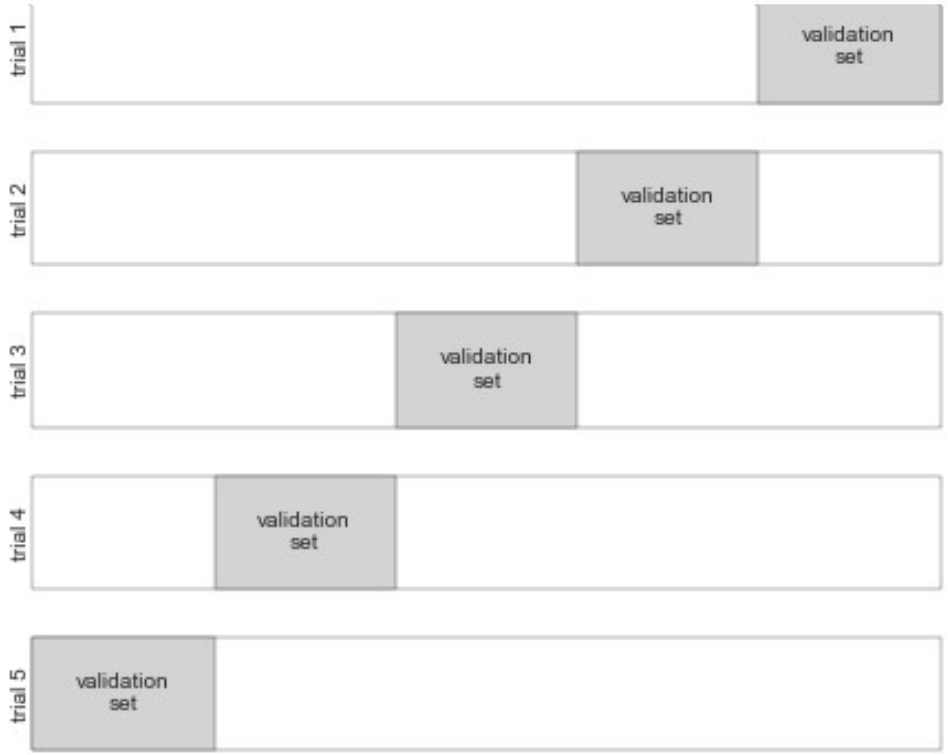
accuracy = 0.90 ± 0.025

K-fold cross-validation

- split the data in five, train on four, test on the one hold-out, and repeat

Five trails






```
from sklearn.model_selection import cross_val_score

model = DecisionTreeClassifier()

scores = cross_val_score(model, X, y, cv=5)
```

```
(0.88, 0.88, 0.90, 0.90, 0.90)
```

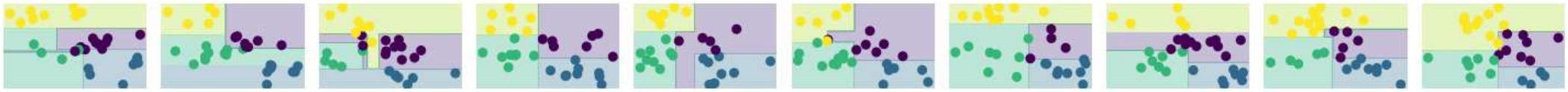
```
print("accuracy = {:.2f} \u00B1 {:.3f}".format(scores.mean(),
                                               scores.std()))
```

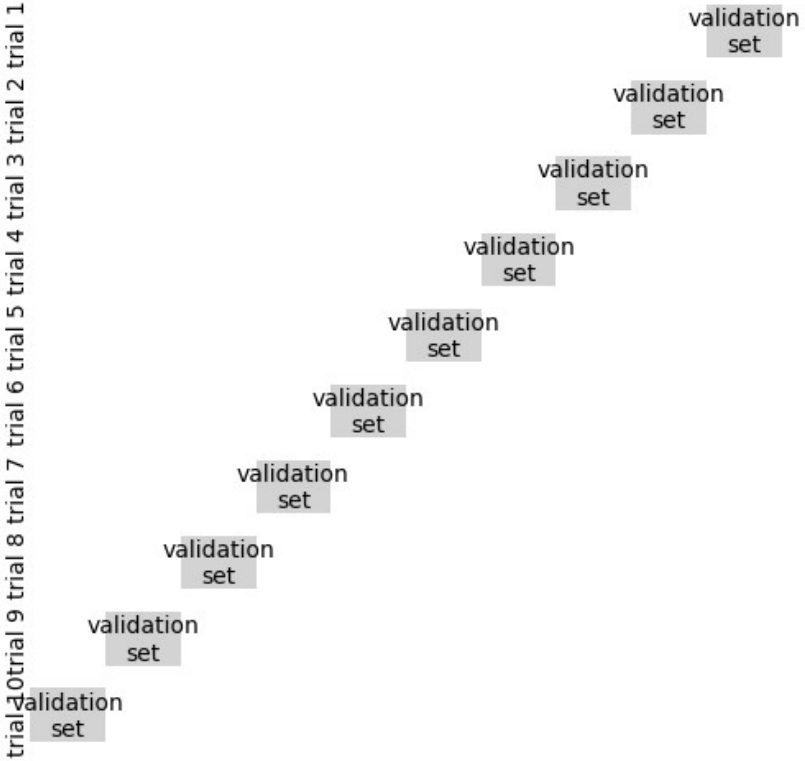
accuracy = 0.89 ± 0.008

K-fold cross-validation

- split the data in ten, train on nine, test on the one hold-out, and repeat

Ten trails






```
from sklearn.model_selection import cross_val_score

model = DecisionTreeClassifier()

scores = cross_val_score(model, X, y, cv=10)
```

```
(0.87, 0.90, 0.93, 0.87, 0.93, 0.97, 0.83, 0.87, 0.97, 0.87)
```

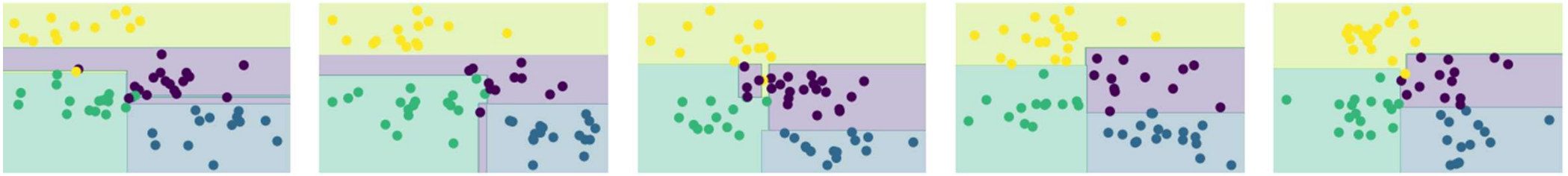
```
print("accuracy = {:.2f} \u00B1 {:.3f}".format(scores.mean(),
                                               scores.std()))
```

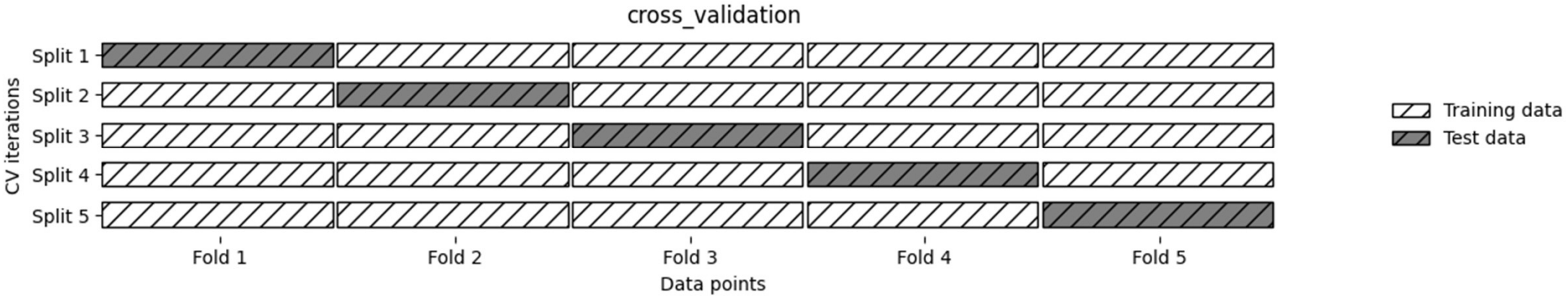
accuracy = 0.90 ± 0.045

5-fold cross-validation

- split the data in five, train on four, test on the one hold-out, and repeat

Five trails

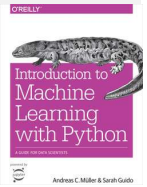





```
from sklearn.model_selection import cross_val_score

model = DecisionTreeClassifier()

scores = cross_val_score(model, X, y, cv=5)
```



accuracy = 0.89 ± 0.008

Page 258

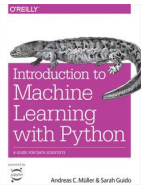
Stratified k-fold cross-validation

- Sometimes your dataset is ordered

Iris Setosa, Versicolor and Virginica

```
from sklearn.datasets import load_iris
iris = load_iris()
print("Iris labels:\n{}".format(iris.target))
```

```
Iris labels:
[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 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 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 2 2 2
 2 2]
```



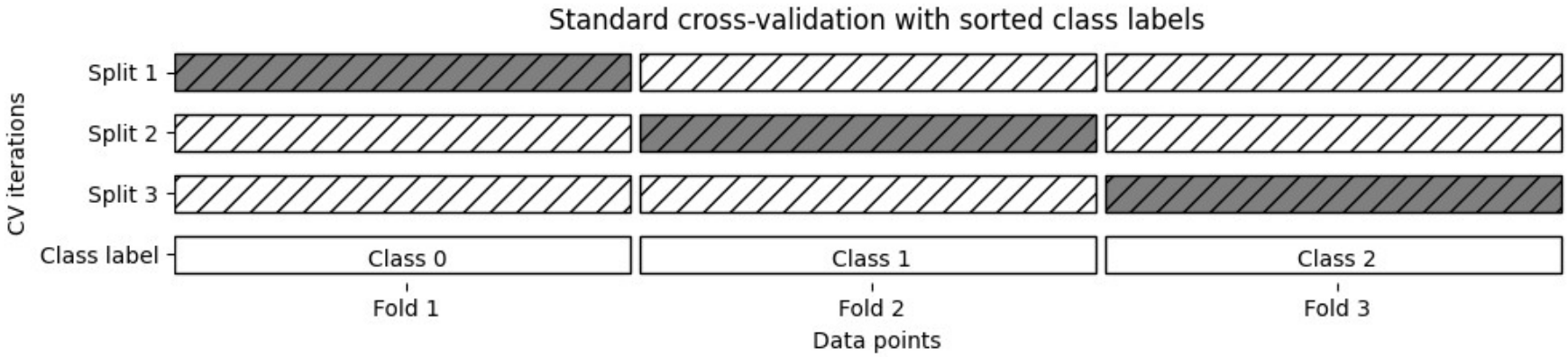
Stratified k-fold cross-validation

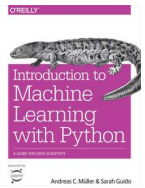
- Sometimes your dataset is ordered

```
from sklearn.datasets import load_iris
iris = load_iris()

model = DecisionTreeClassifier()

scores = cross_val_score(model, X, y, cv=3)
```





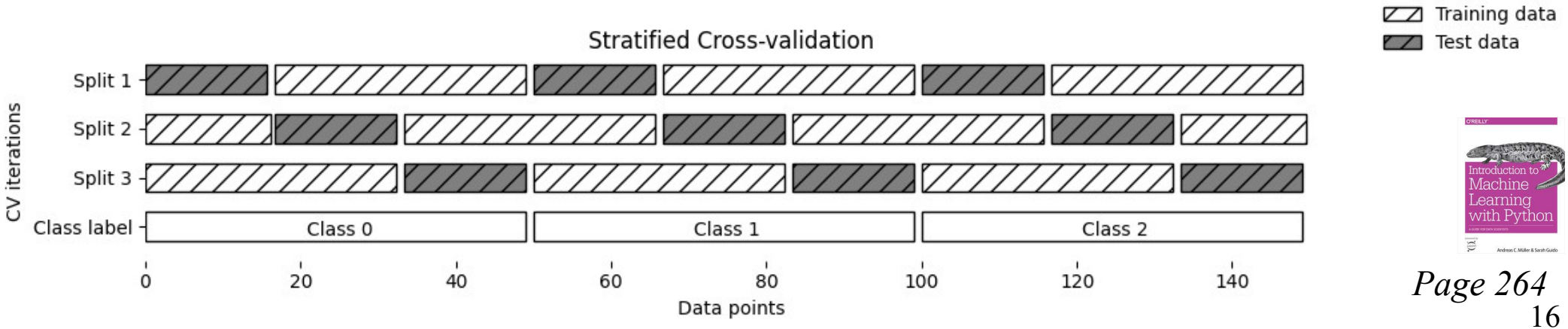
Stratified k-fold cross-validation

- Sometimes your dataset is ordered

```
from sklearn.datasets import load_iris
iris = load_iris()

model = DecisionTreeClassifier()

scores = cross_val_score(model, X, y, cv=3)
```



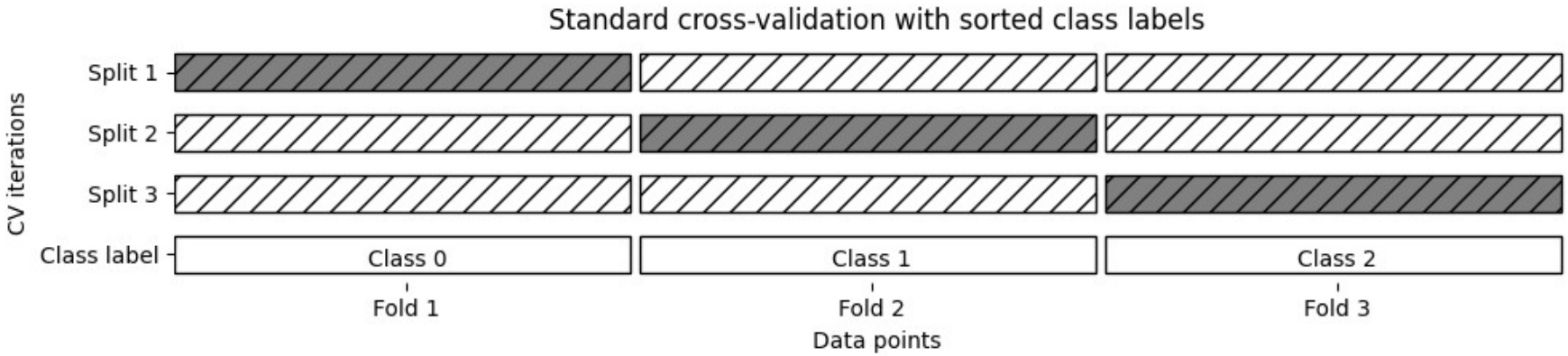
Stratified k-fold cross-validation

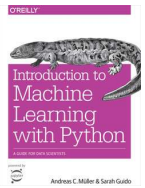
- Sometimes your dataset is ordered

```
from sklearn.model_selection import KFold
kfold = KFold(n_splits=3)

scores = cross_val_score(model, X, y, cv=kfold)
```

```
[0.0, 0.0, 0.0]
```





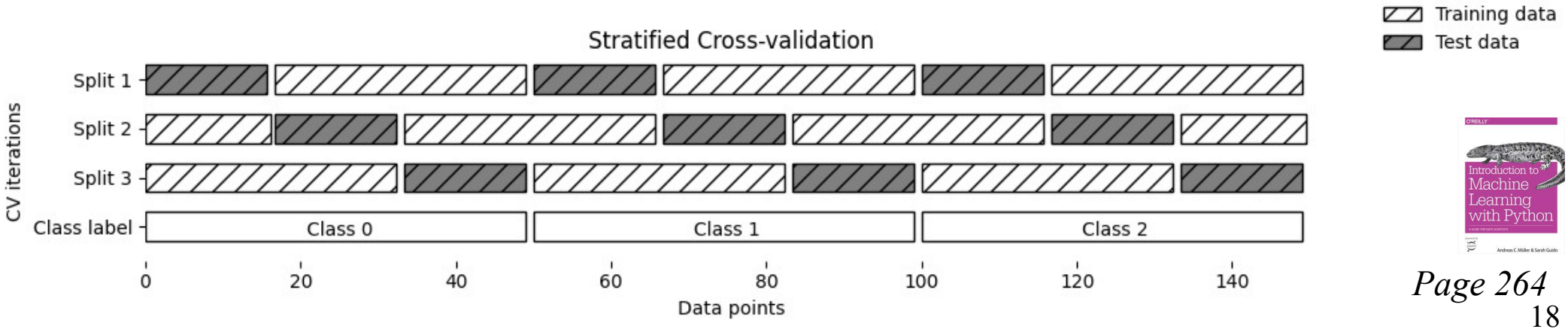
Stratified k-fold cross-validation

- Sometimes your dataset is ordered

```
from sklearn.model_selection import KFold
kfold = KFold(n_splits=3, shuffle=True)

scores = cross_val_score(model, X, y, cv=kfold)
```

```
[0.98 0.96 0.96]
```



Shuffle-split cross-validation

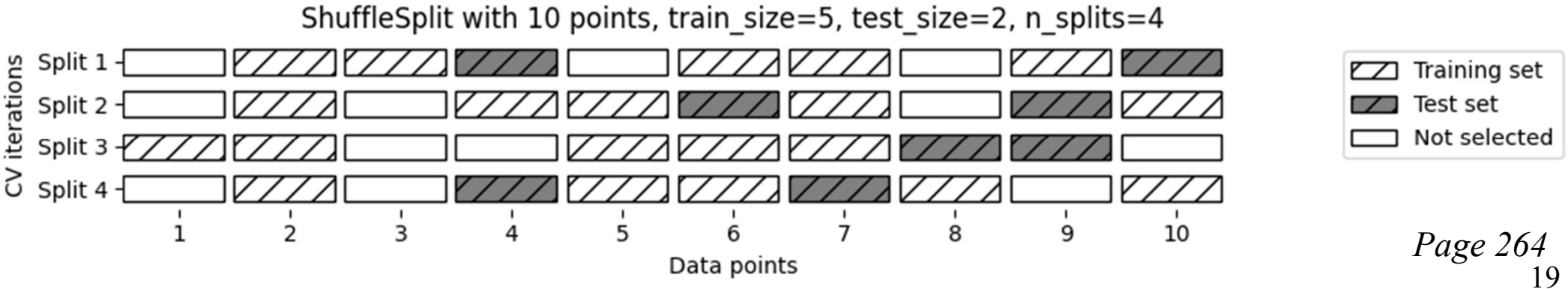
- Leave a portion out ← good for robustness

```
from sklearn.model_selection import ShuffleSplit
shuffle_split = ShuffleSplit(test_size=.5, train_size=.5, n_splits=10)

scores = cross_val_score(model, X, y, cv=shuffle_split)
```

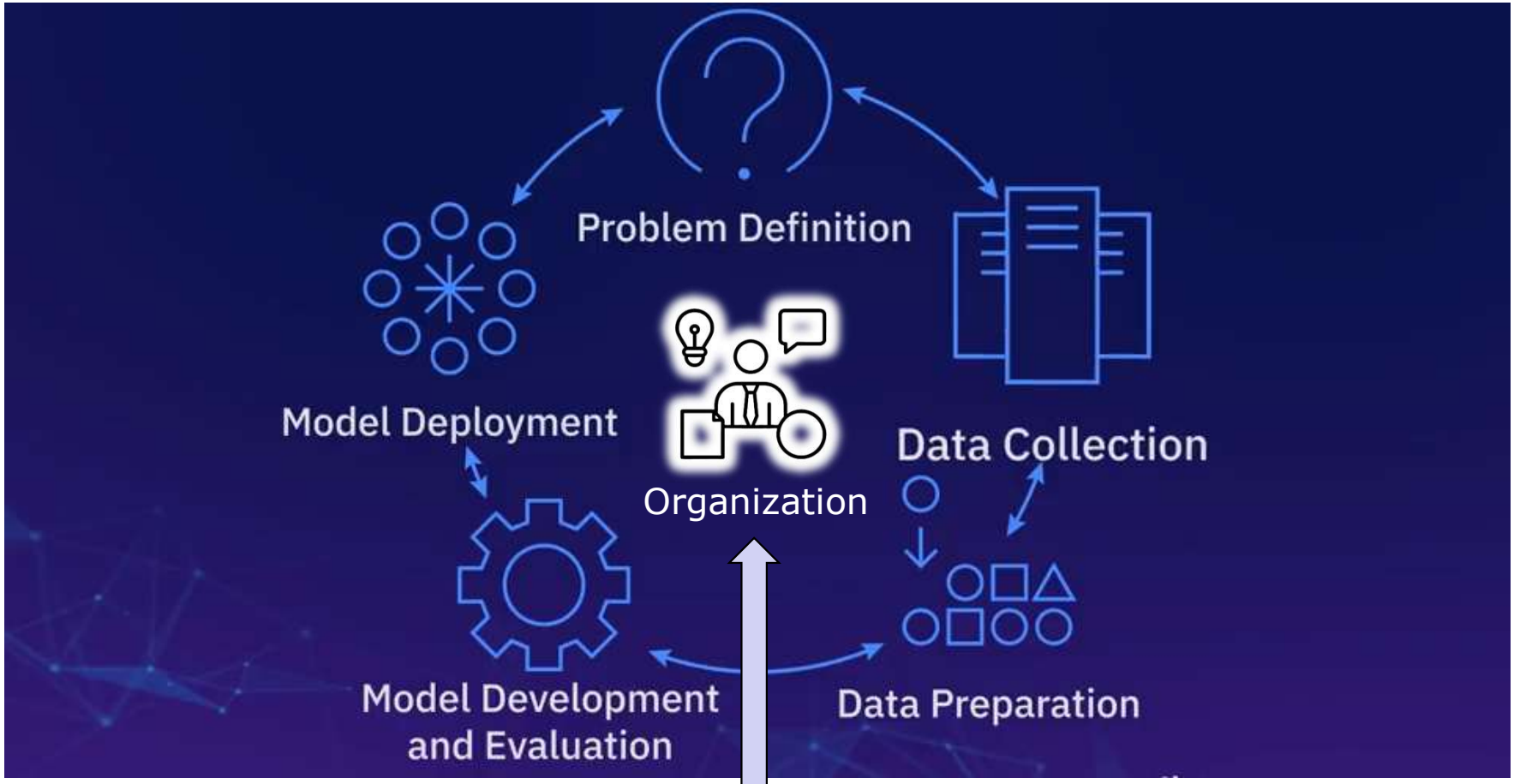
```
[0.96 0.93 0.97 0.97 0.96 0.92 0.97 0.97 0.97 0.92]
```

accuracy = 0.96 ± 0.022



Introduction to Machine Learning

Building a predictive model is a circular process.

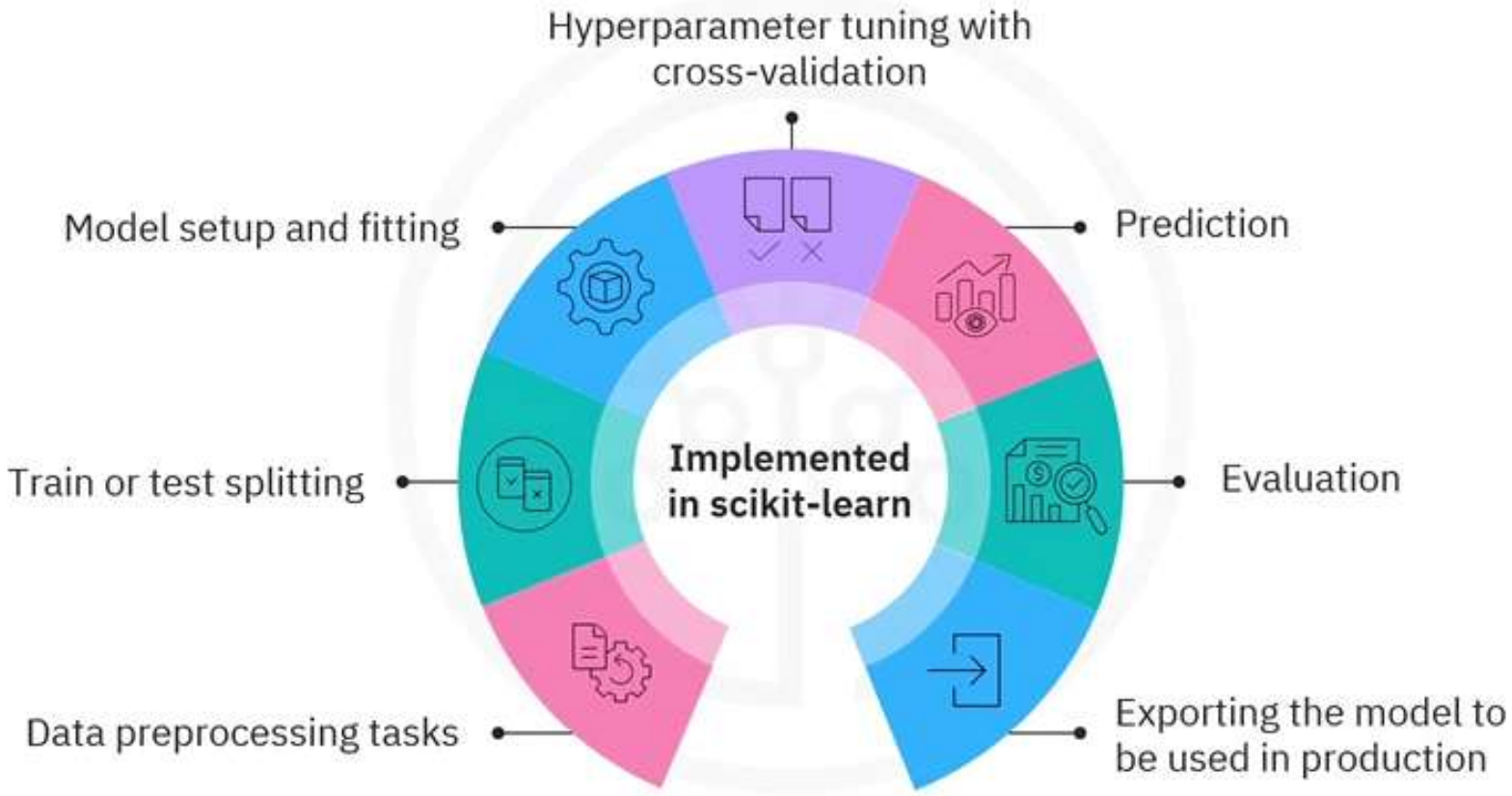


Expertise

- Depends on the organization
- Interpret the information in the right way

Introduction to Machine Learning

Building a predictive model is a circular process.

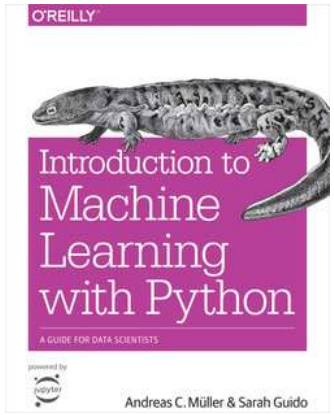


Joseph Santarcangelo & Joss Grossman, Machine Learning with Python, IBM Skills network, 2025

Introduction to Machine Learning

- 5.1 Cross-Validation

- Strengths:
  - A reliable estimate of the performance
  - No better performance of a model by ‘luck’
- Weakness:
  - Another hyperparameter to control
  - You have to know your data



Andreas C. Müller, Sarah Guido, Introduction to Machine Learning with Python, O’Reilly Media, October 2016

Conclusion

Learning outcomes of this course covered today

- Most work is in Data Preparation and Model Evaluation
- Beter do Model Evaluation the correct way!