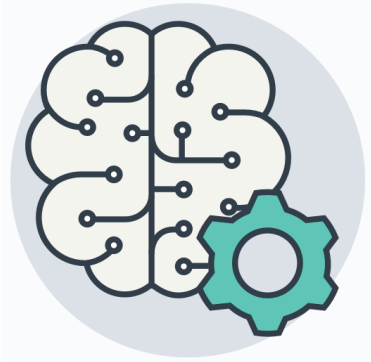
Aprendizado de Máquina

Validação / Split



Prof. Regis Pires Magalhães

regismagalhaes@ufc.br - http://bit.ly/ufcregis

Flower Classification

Iris-Setosa

Iris-Versicolor

Iris-Setosa

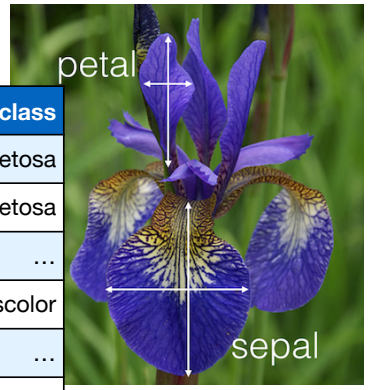
Data Representation

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



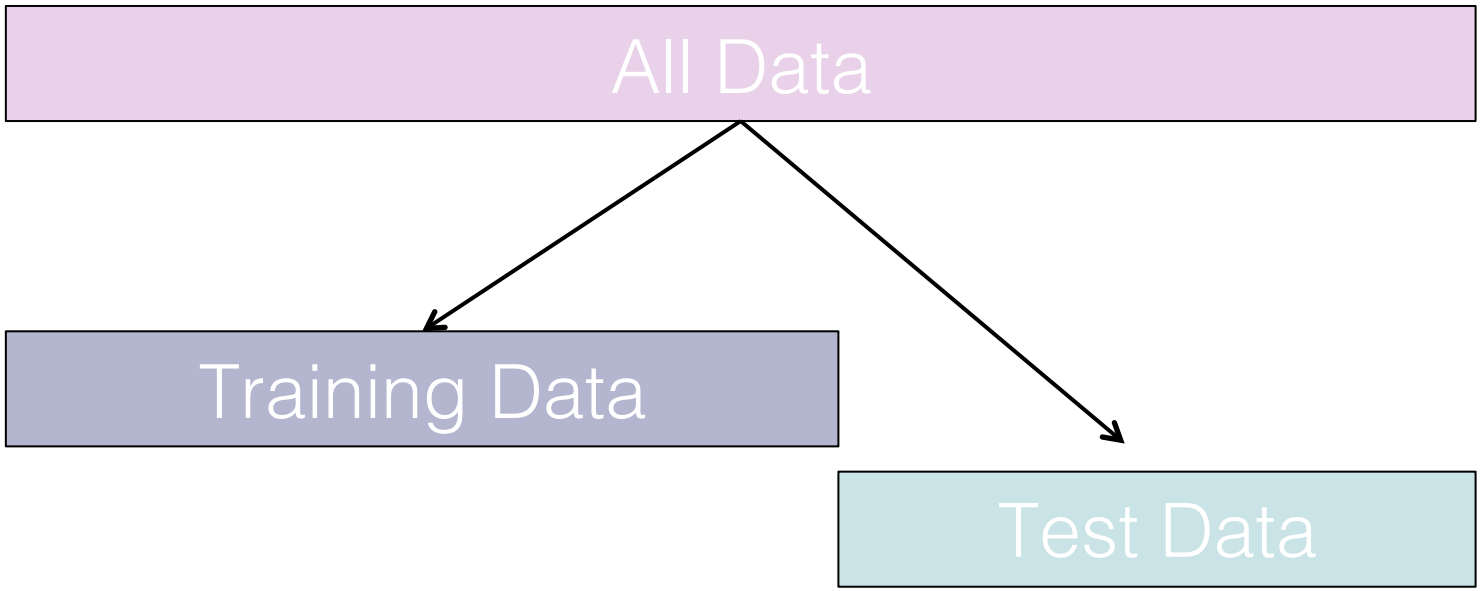
Features (attributes, dimensions)

Classes (targets)

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

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

Training & Test Data

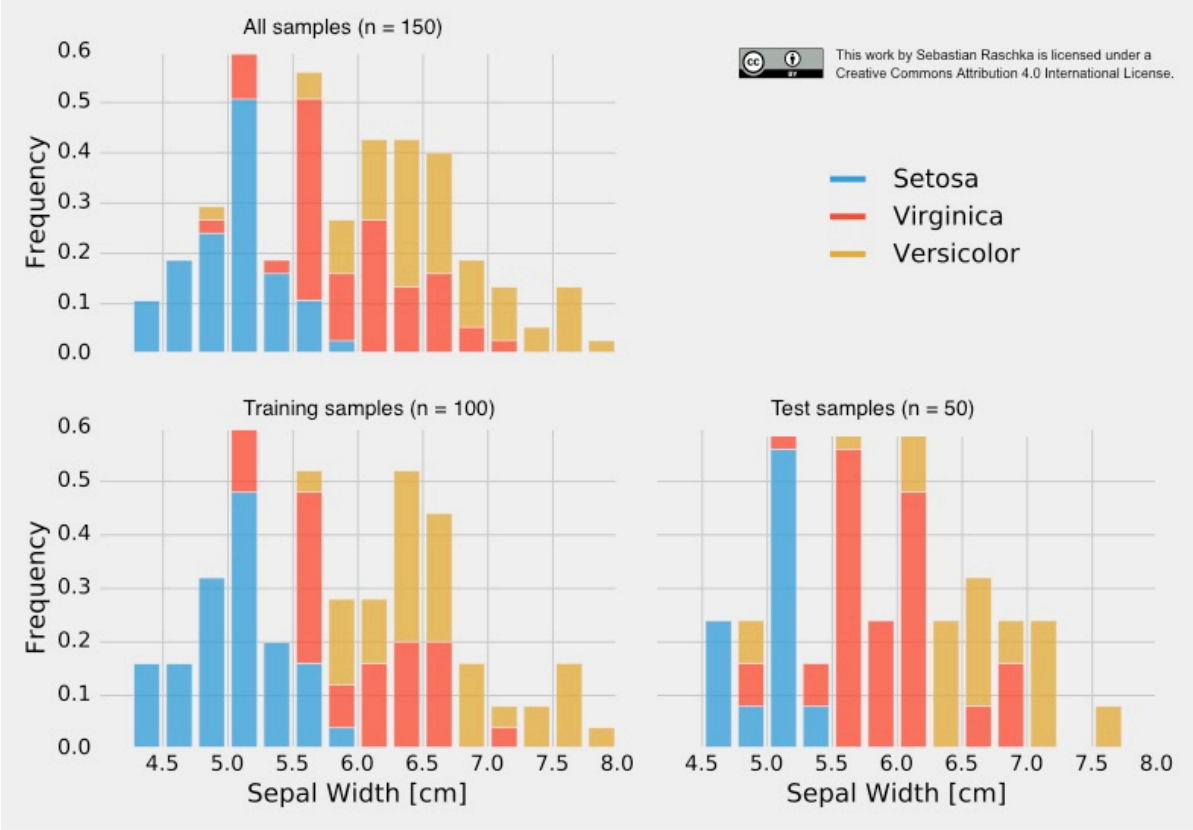


Typically:

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

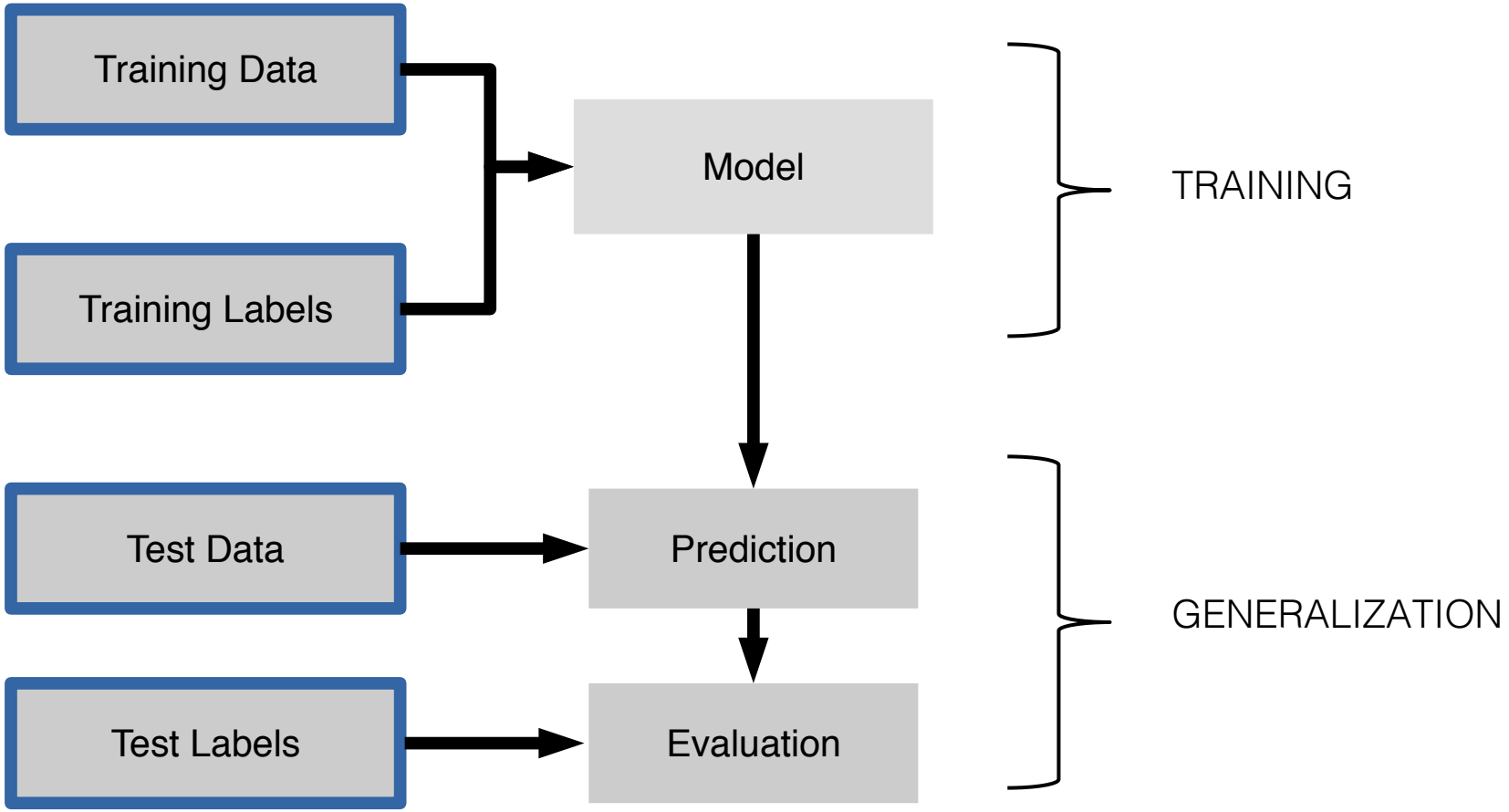
Stratification

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

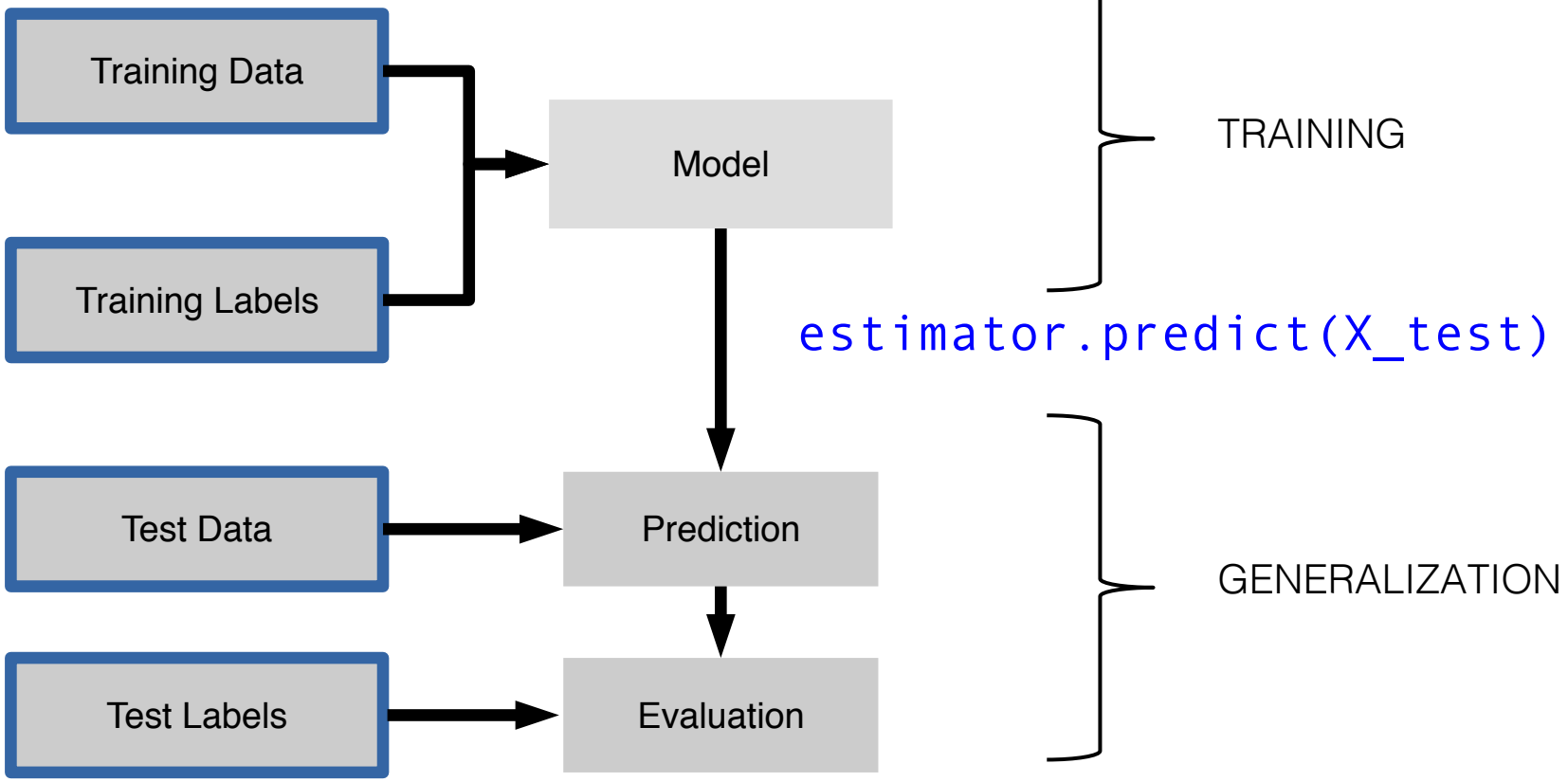
Supervised Workflow



- Fit model on all data after evaluation

Supervised Workflow

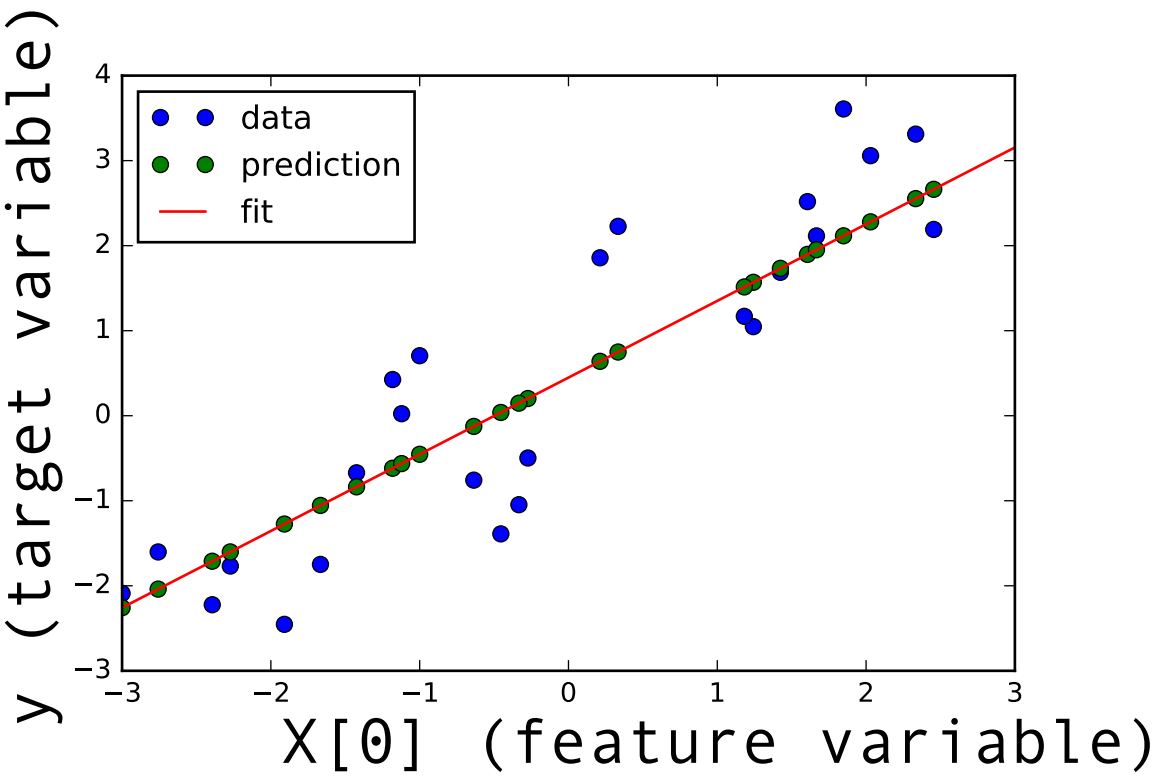
```
estimator.fit(X_train, y_train)
```



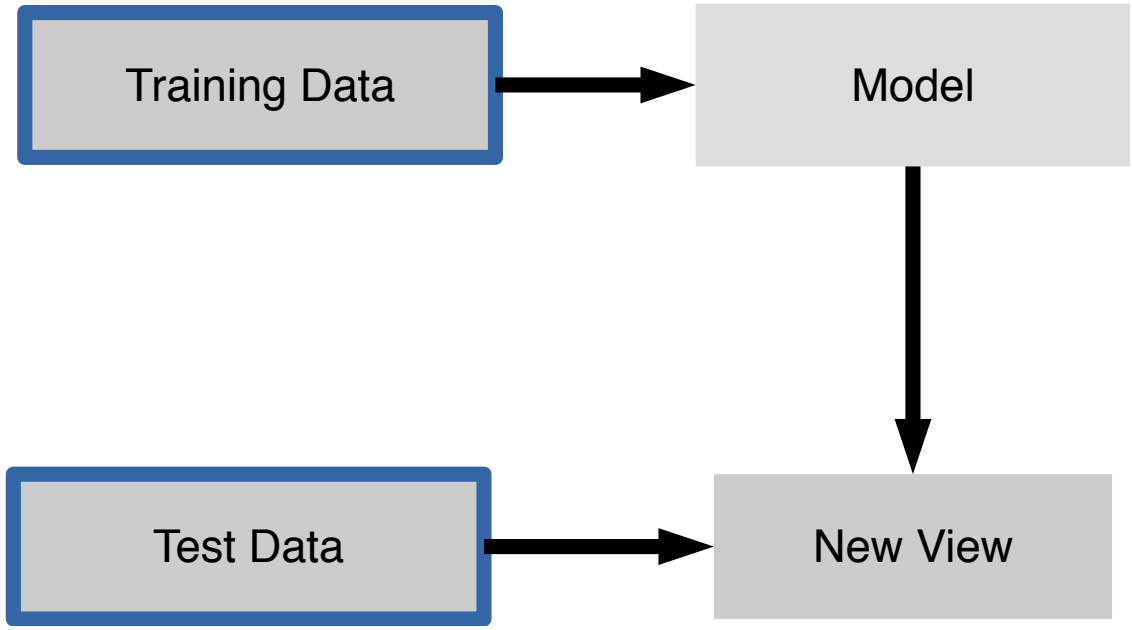

```
accuracy_score(y_test, y_pred)
```

Linear Regression

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

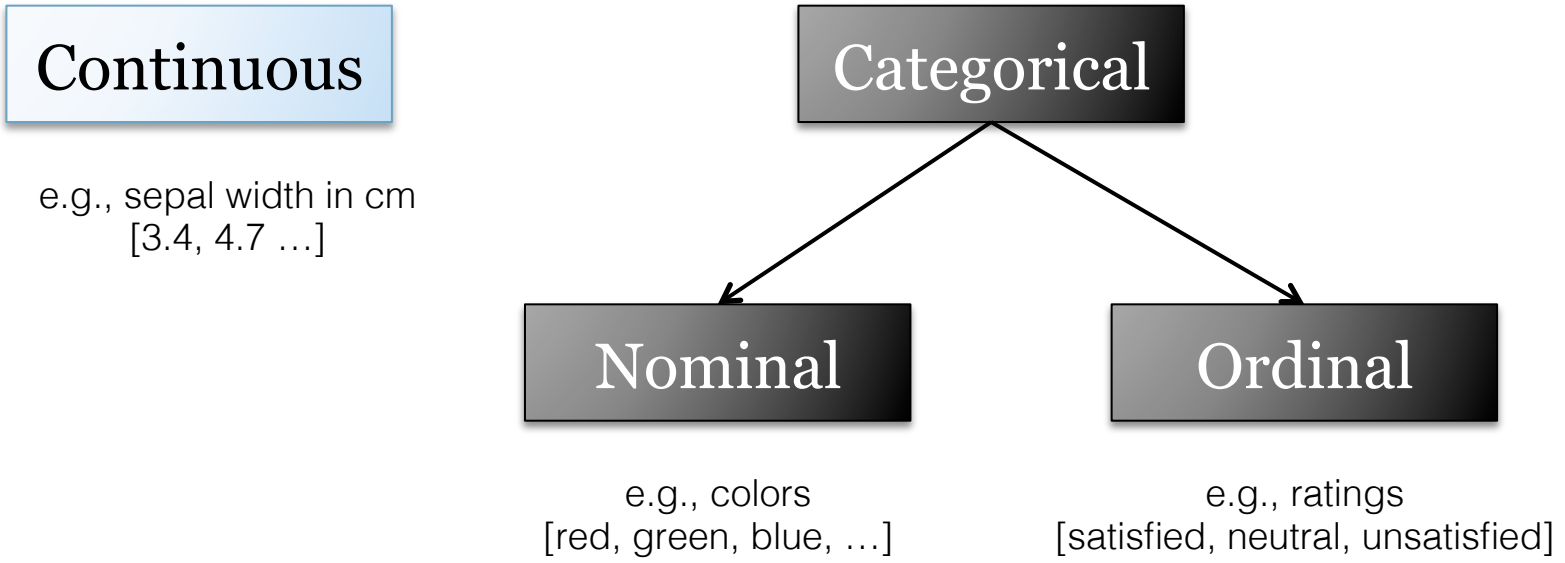


Unsupervised Transformers



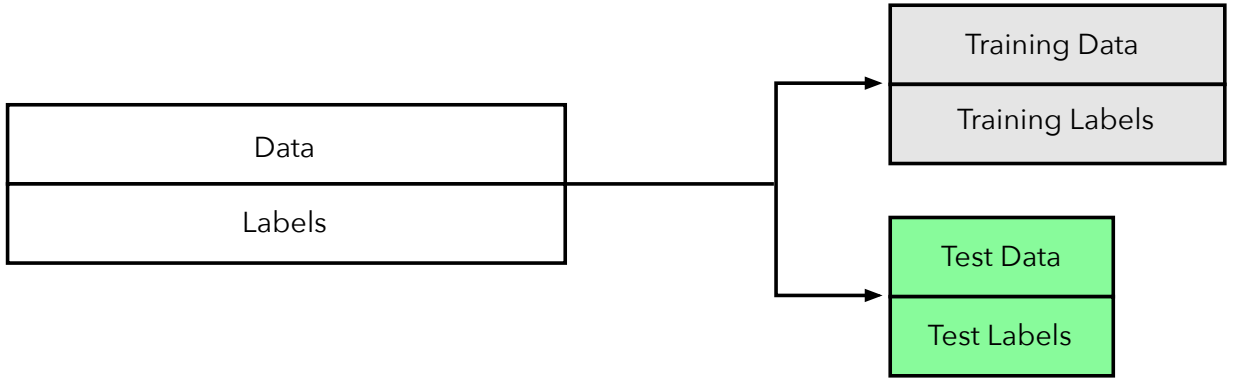
① `transformer.fit(X_train)`
② `X_train_transf = transformer.transform(X_train)`
③ `X_test_transf  = transformer.transform(X_test)`

Continuous & Categorical Features

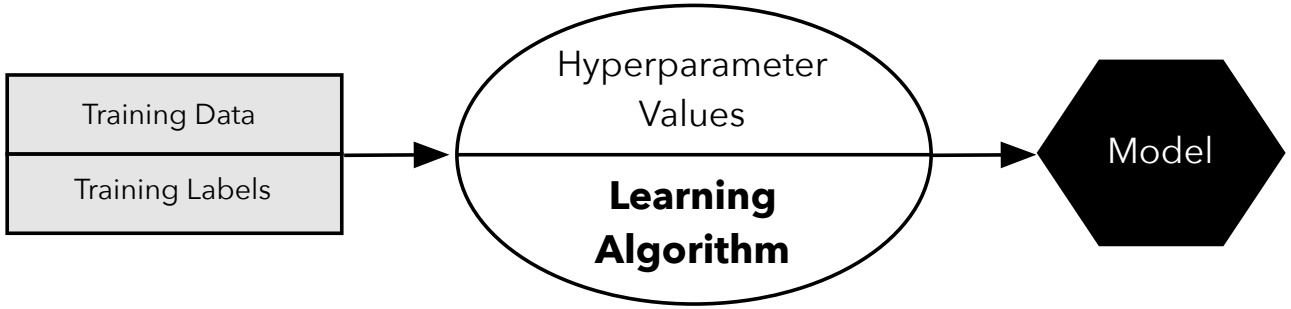


Holdout Evaluation I

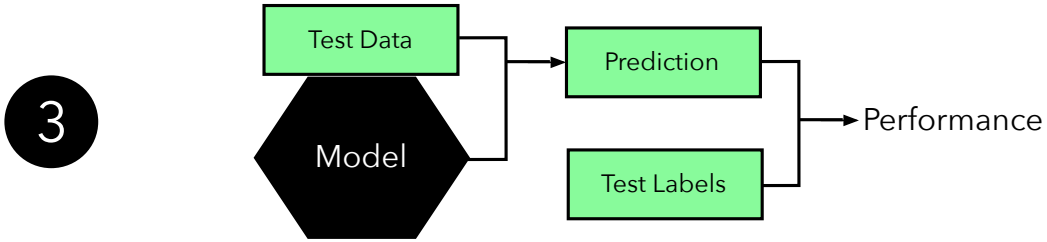
1

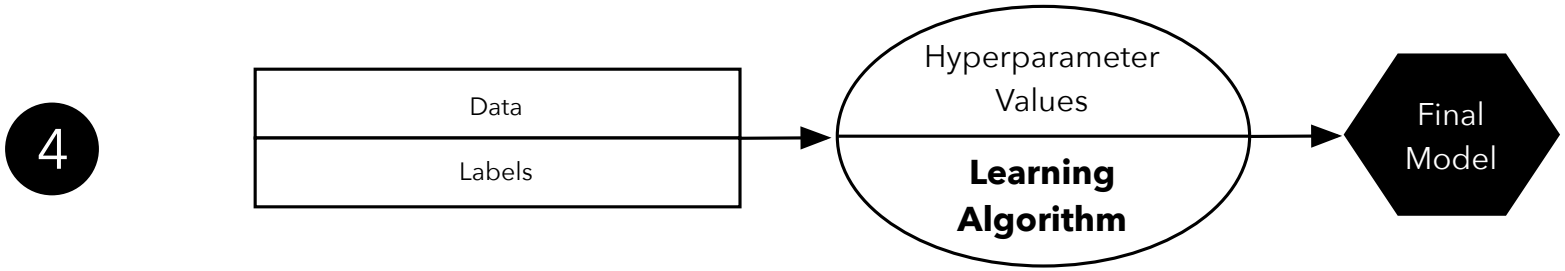


2

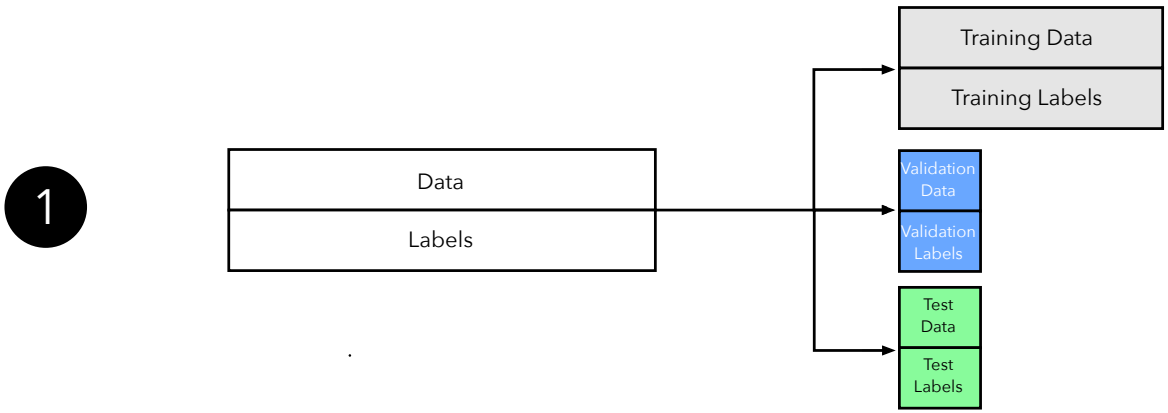


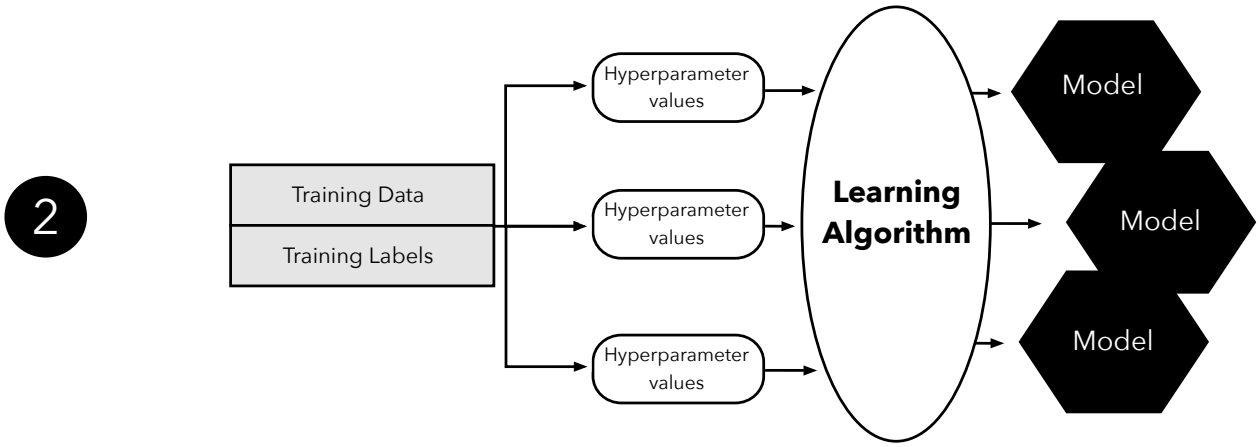
Holdout Evaluation II





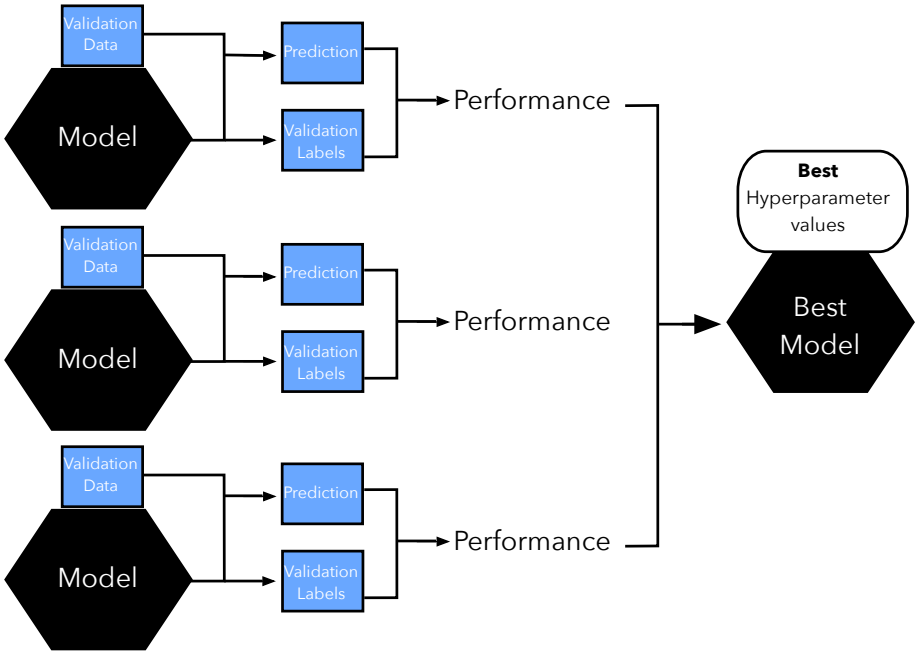
Holdout Validation I



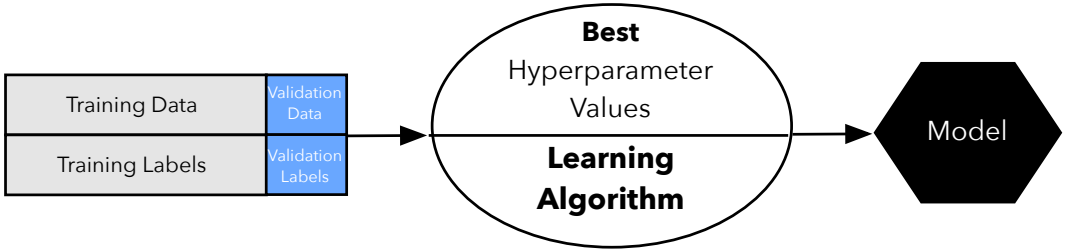


Holdout Validation II

3

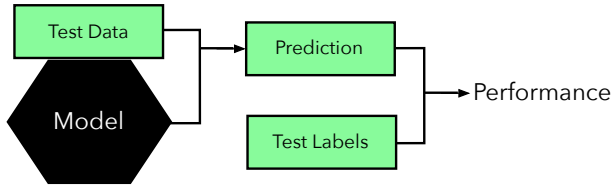


4

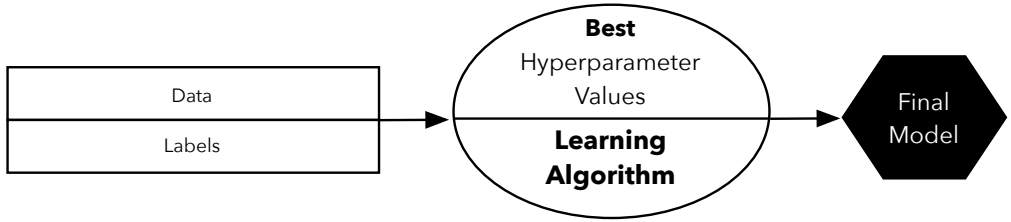


Holdout Validation III

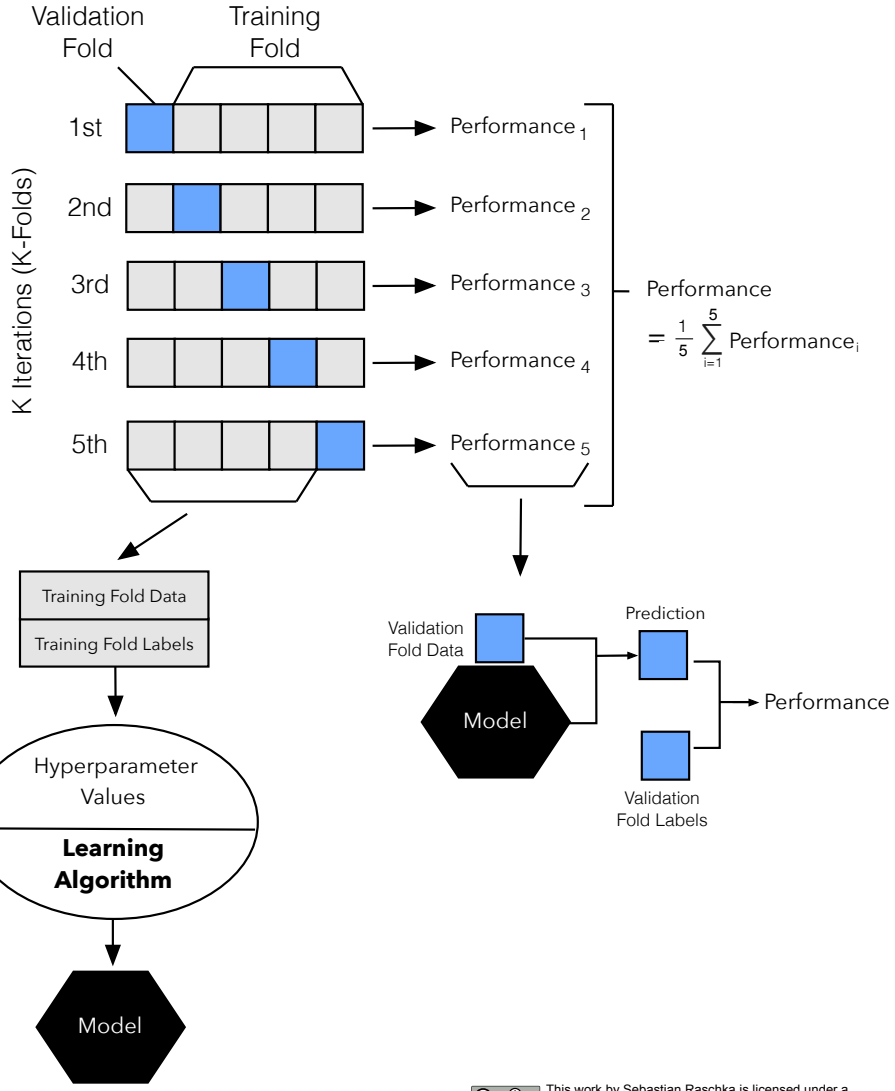
5



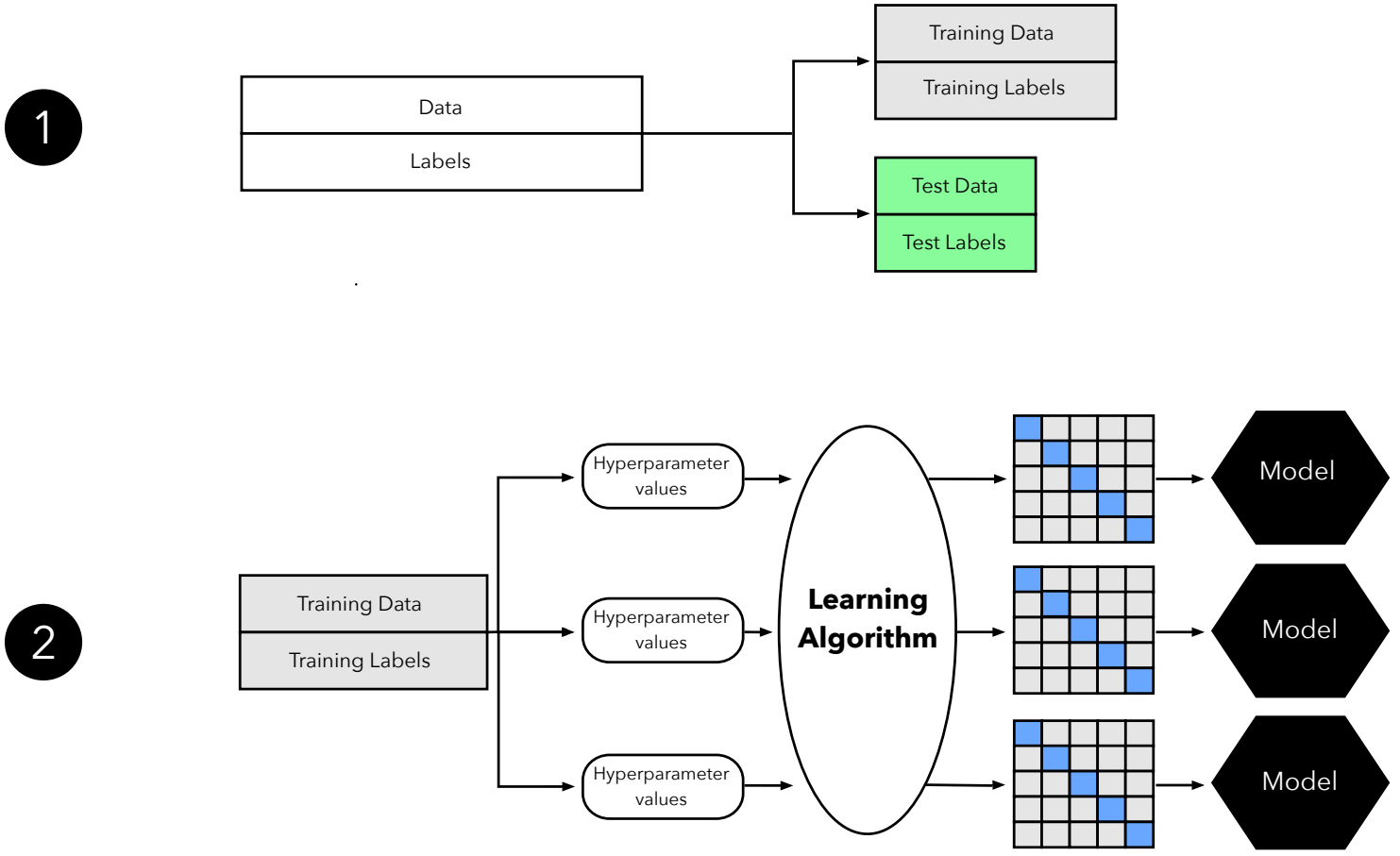
6



K-fold Cross-Validation

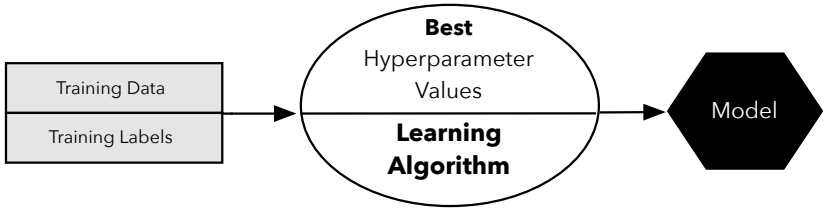


K-fold Cross-Validation Pipeline I

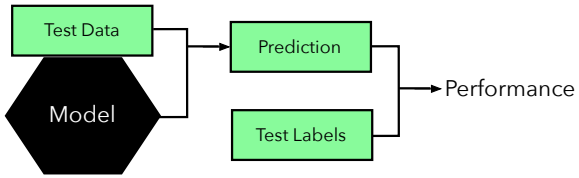


K-fold Cross-Validation Pipeline II

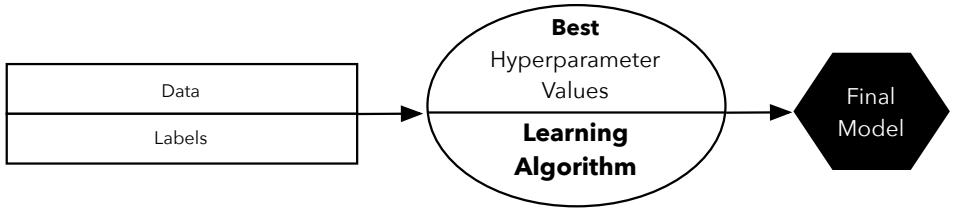
3



4



5



Stratification Scikit Learn

```
from sklearn.model_selection import StratifiedShuffleSplit, StratifiedKFold
from sklearn import datasets

splits = 3

tx = range(12)
ty = [0] * 6 + [1] * 6

print("KFold")
kfold = StratifiedKFold(n_splits=splits, shuffle=True, random_state=42)
for train_index, test_index in kfold.split(tx, ty):
    print("TRAIN:", train_index, "TEST:", test_index)

print("\nShuffle Split")
shufflesplit = StratifiedShuffleSplit(n_splits=splits, test_size=1/3,
random_state=42)
for train_index, test_index in shufflesplit.split(tx, ty):
    print("TRAIN:", train_index, "TEST:", test_index)
```

Stratification Scikit Learn

```
KFold
TRAIN: [ 2  3  4  5  7  8 10 11] TEST: [0 1  6  9]
TRAIN: [ 0  1  3  4  6  9 10 11] TEST: [2 5  7  8]
TRAIN: [ 0  1  2  5  6  7  8  9] TEST: [3 4 10 11]

Shuffle Split
TRAIN: [ 0  1  6  2  9  8  5  7] TEST: [10 11  3  4]
TRAIN: [ 6  5  2  8 11  0 10  4] TEST: [1   3  7  9]
TRAIN: [11  4  9 10  3  1  0  7] TEST: [8   6  2  5]
```

Learning Curves

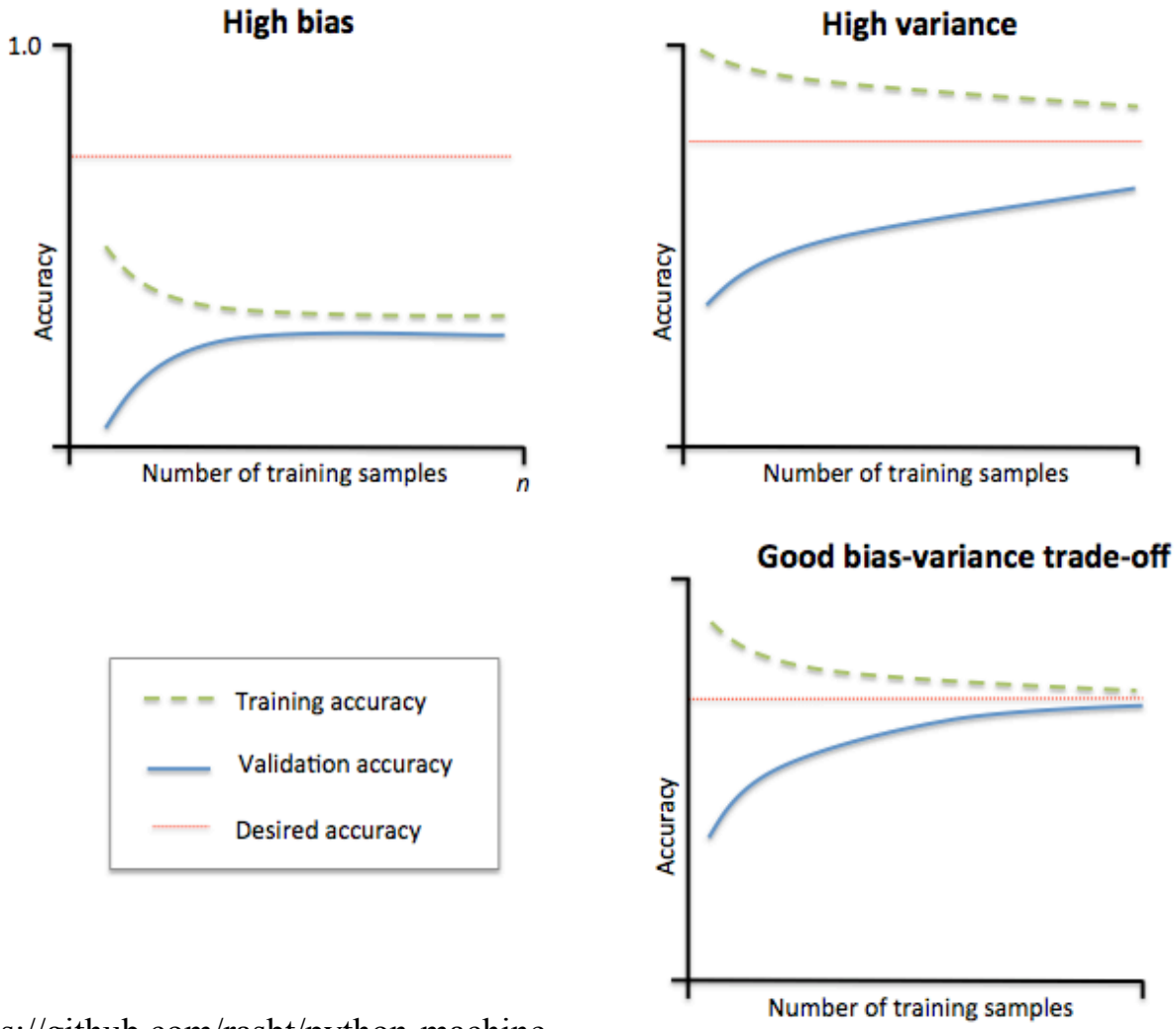
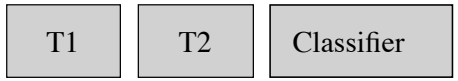


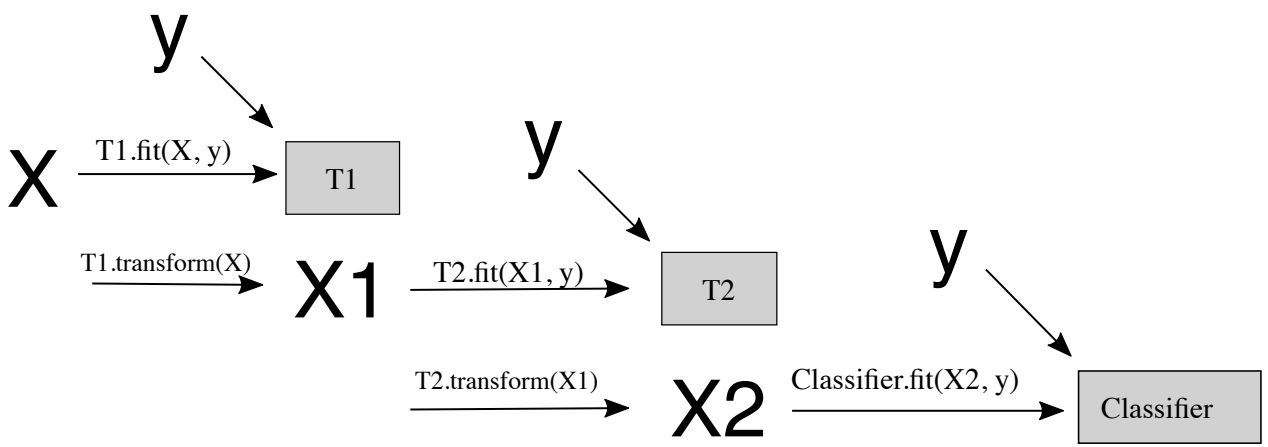
Image source: https://github.com/rasbt/python-machine-learning-book/blob/master/code/ch06/images/06_04.png

Pipelines

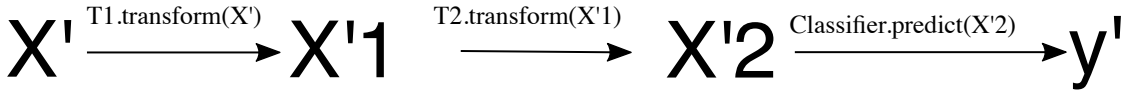
```
pipe = make_pipeline(T1(), T2(), Classifier())
```



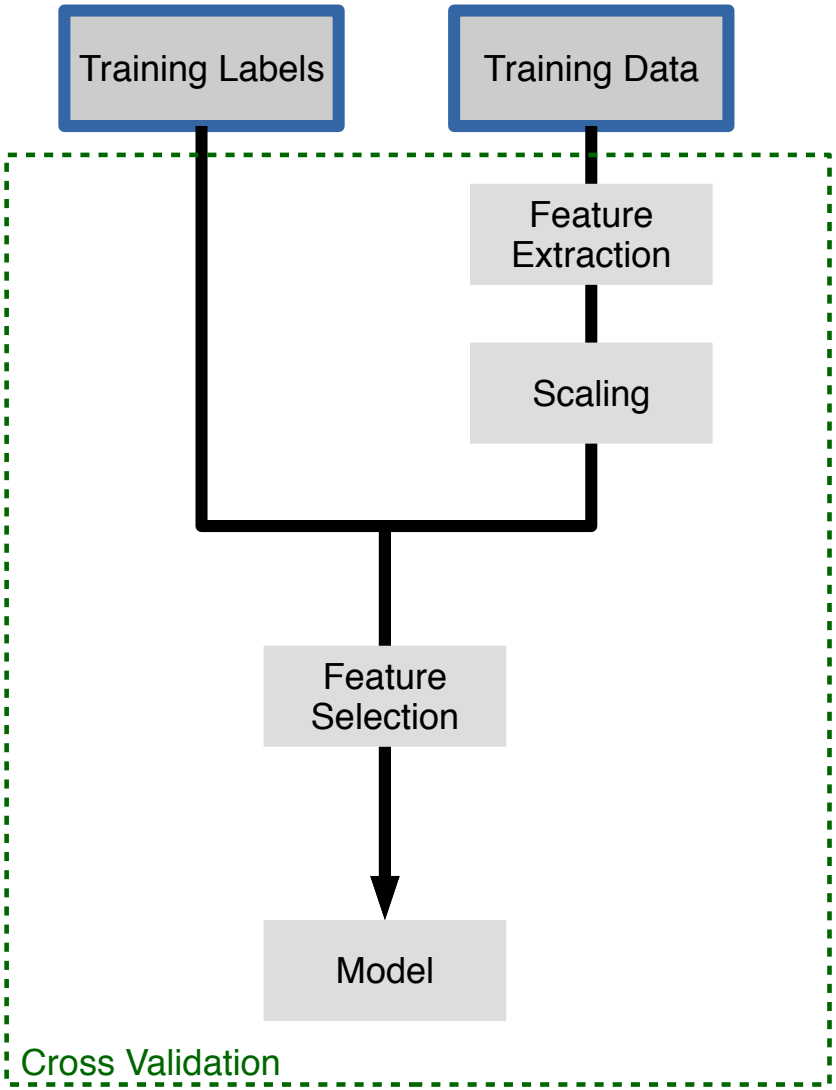

```
pipe.fit(X, y)
```




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

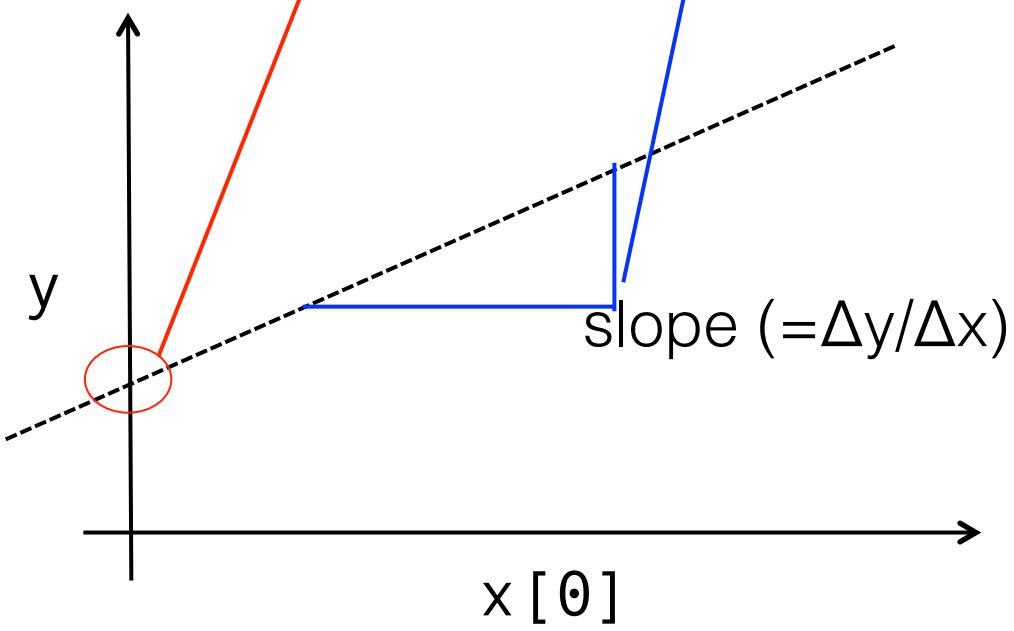


Pipelines & Cross Validation

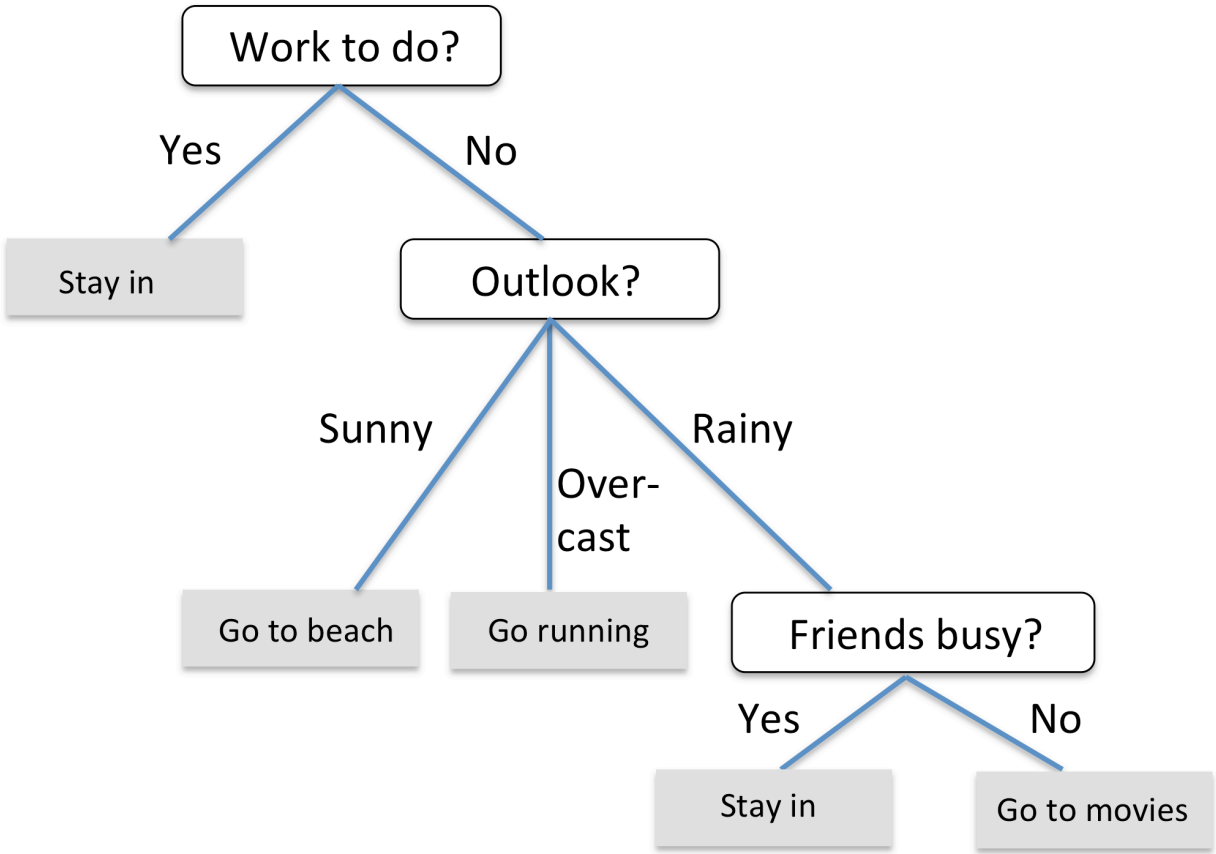


Linear models for regression

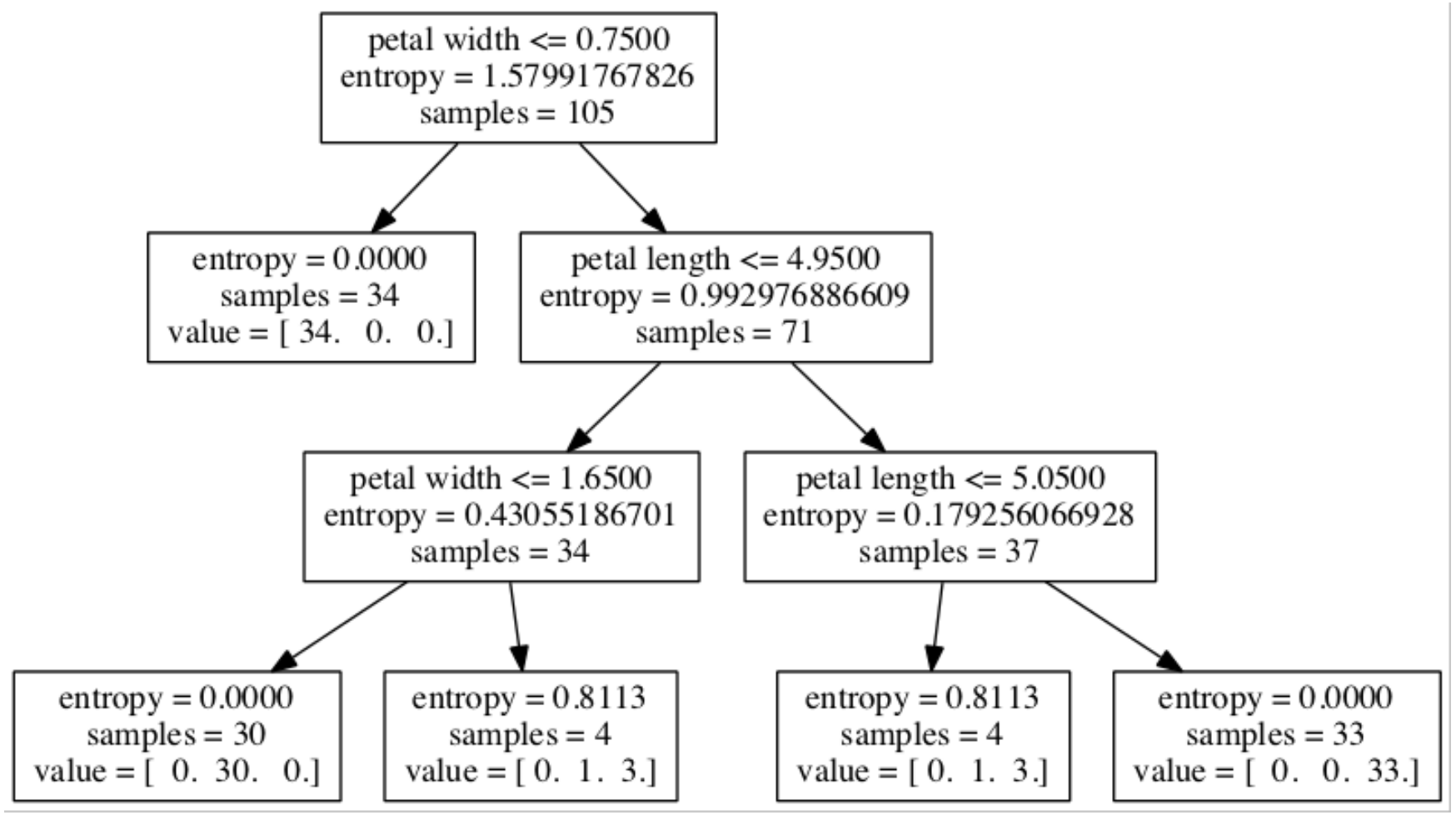
```
y_pred =   x_test[0] * coef_[0] + ...
         + x_test[n_features-1] * coef_[n_features-1]
         + intercept_
```



Decision Trees



Classification w. Continuous Features



Obrigado!

Dúvidas, comentários, sugestões?

Regis Pires Magalhães
regismagalhaes@ufc.br

UFC