

Introduction to Machine learning with scikit-learn

Important ideas

What is machine learning?

- extracting knowledge from data
- closely related to statistics and optimization.
- What distinguishes machine learning is that it is very focused on prediction.

Types of Machine Learning

- Supervised
- Unsupervised
- Reinforcement
-

- **Supervised** learning: Models that can predict labels based on **labeled data**
 - Classification: Models that predict labels as two or more discrete categories
 - Regression: Models that predict continuous labels
- **Unsupervised** learning: Models that identify structure in unlabeled data
 - Clustering: Models that detect and identify distinct groups in the data
 - Dimensionality reduction: Models that detect and identify lower-dimensional structure in higher-dimensional data.

$$(x_i, y_i) \propto p(x, y) \text{ i.i.d.}$$

$$x_i \in \mathbb{R}^p$$

$$y_i \in \mathbb{R}$$

$$f(x_i) \approx y_i$$

- Given an array of test results from a patient, does this patient have diabetes?

The x_i would be the different test results, and y_i would be diabetes or no diabetes.

- Given a piece of a satellite image, what is the terrain in this image?

Here x_i would be the pixels of the image, and y_i would be the terrain types.

$$x_i \propto p(x) \text{ i.i.d.}$$

Learn about p .

- discovering topics in news articles or on twitter, or grouping data into clusters for easier analysis.
- outlier detection, where you ask “does this data look normal” which is important for fraud detection and security systems.

Classification

- target y discrete
- Is this patient sick?

Regression

- target y continuous
- How long will it take for the patient to recover?

- Not only $f(x_i) \approx y_i$ for the data seen/used
- also for new data: $f(x) \approx y$

Supervised learning

What is the relationship between input and output variables?

IQ	ClassesAttended	CatsYouOwn	YourFinalGrade
110	14	0	73
105	28	2	99
107	26	1	95

- Input variables
 - Independent variables, predictors, input, features
- Output variables
 - Dependent variables, response, output

The relationship between X and Y

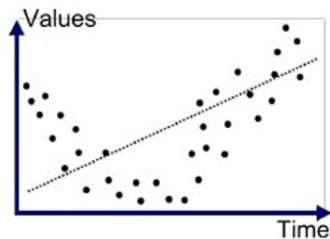
The true relationship is $Y = f(X)$

Our goal is to come up with an *estimate* \hat{f} of the true f

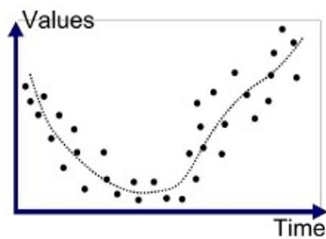
$$\hat{Y} = \hat{f}(X)$$

- Why?
 - So we can plug in values of X and see what \hat{Y} are produced
 - Think of it like a machine.

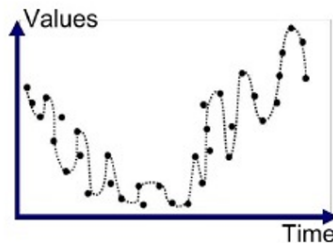
Overfitting



Underfitted



Good Fit/Robust



Overfitted

- Worst sin : overfitting

bcos the model works well for data that is seen but does poor job for unseen data

- Split data into Train-Test.

Representing Data

one sample

$$X = \begin{pmatrix} 1.1 & 2.2 & 3.4 & 5.6 & 1.0 \\ 6.7 & 0.5 & 0.4 & 2.6 & 1.6 \\ 2.4 & 9.3 & 7.3 & 6.4 & 2.8 \\ 1.5 & 0.0 & 4.3 & 8.3 & 3.4 \\ 0.5 & 3.5 & 8.1 & 3.6 & 4.6 \\ 5.1 & 9.7 & 3.5 & 7.9 & 5.1 \\ 3.7 & 7.8 & 2.6 & 3.2 & 6.3 \end{pmatrix}$$

one feature

$$y = \begin{pmatrix} 1.6 \\ 2.7 \\ 4.4 \\ 0.5 \\ 0.2 \\ 5.6 \\ 6.7 \end{pmatrix}$$

outputs / labels

Training and Test Data

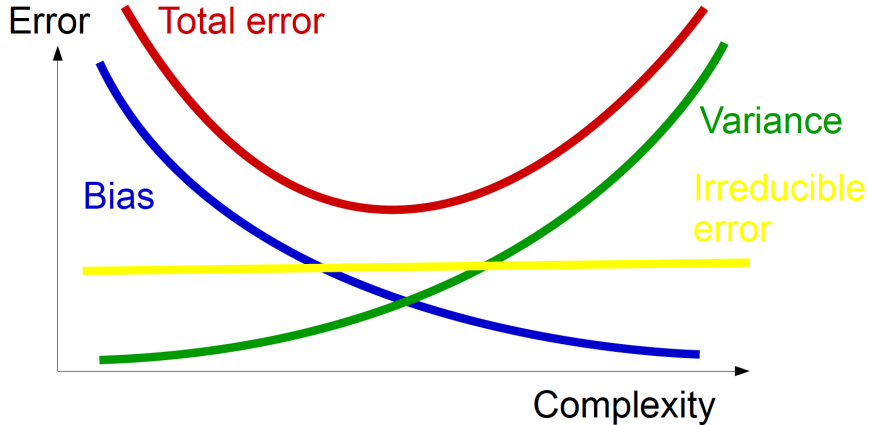
$$X = \begin{array}{c} \text{training set} \\ \begin{pmatrix} 1.1 & 2.2 & 3.4 & 5.6 & 1.0 \\ 6.7 & 0.5 & 0.4 & 2.6 & 1.6 \\ 2.4 & 9.3 & 7.3 & 6.4 & 2.8 \\ 1.5 & 0.0 & 4.3 & 8.3 & 3.4 \\ 0.5 & 3.5 & 8.1 & 3.6 & 4.6 \\ 5.1 & 9.7 & 3.5 & 7.9 & 5.1 \\ 3.7 & 7.8 & 2.6 & 3.2 & 6.3 \end{pmatrix} \\ \text{test set} \end{array}$$

$$y = \begin{array}{c} \begin{pmatrix} 1.6 \\ 2.7 \\ 4.4 \\ 0.5 \\ 0.2 \\ 5.6 \\ 6.7 \end{pmatrix} \end{array}$$

Model evaluation and selection

- train, test
- fit on train
- predict on test
- score the model by how well it does on test

- All data science in one slide: **bias variance tradeoff**



bias variance tradeoff

want the model with least error. As complexity increases, bias goes down but variance increases

- Every estimator has its advantages and drawbacks.
- Its generalization error can be decomposed in terms of bias, variance and noise.
 - The bias of an estimator is its average error for different training sets.
 - The variance of an estimator indicates how sensitive it is to varying training sets.
 - Noise is a property of the data.

Datasets

- Breast Cancer
- Wine
- Iris
- Parkinsons

Team Exercise 1

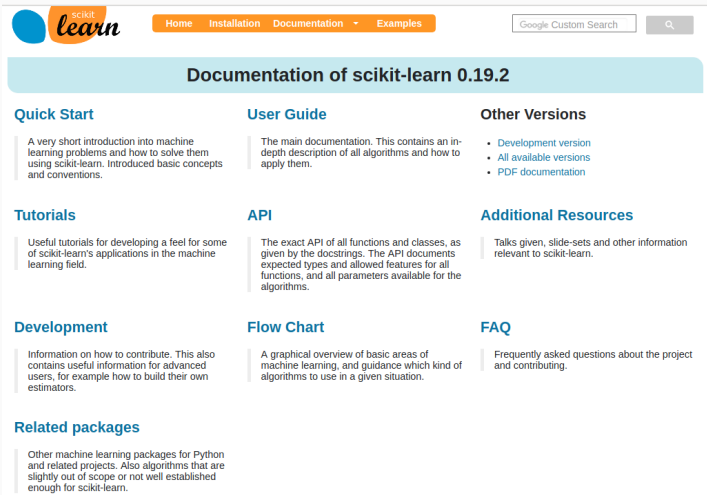
Take a look at the data

Team Exercise 2

Plot the data

Present : Explain your data set

- Explain the problem you are trying to solve
- Give some preliminary visualizations that are indicative of the difficulty of the problem and possible approaches



The screenshot shows the scikit-learn documentation homepage for version 0.19.2. At the top, there is a navigation bar with links for Home, Installation, Documentation (selected), and Examples. A search bar labeled 'Google Custom Search' is also present. Below the navigation bar, the main heading is 'Documentation of scikit-learn 0.19.2'. The page is organized into a grid of links and descriptions for various documentation sections.

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

- [Development version](#)
- [All available versions](#)
- [PDF documentation](#)

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.

Development
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.

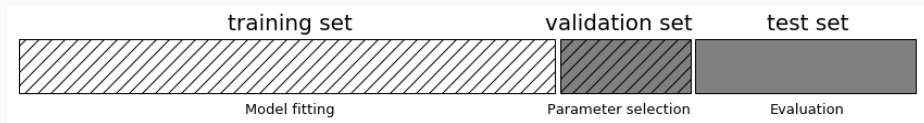
FAQ
Frequently asked questions about the project and contributing.

Related packages
Other machine learning packages for Python and related projects. Also algorithms that are slightly out of scope or not well established enough for scikit-learn.

Team Exercise 3

Prepare Rmd slides and present one of the methods below

Threefold split



We use the training set for model building, the validation set for parameter selection and the test set for a final evaluation of the model.

pro: fast, simple

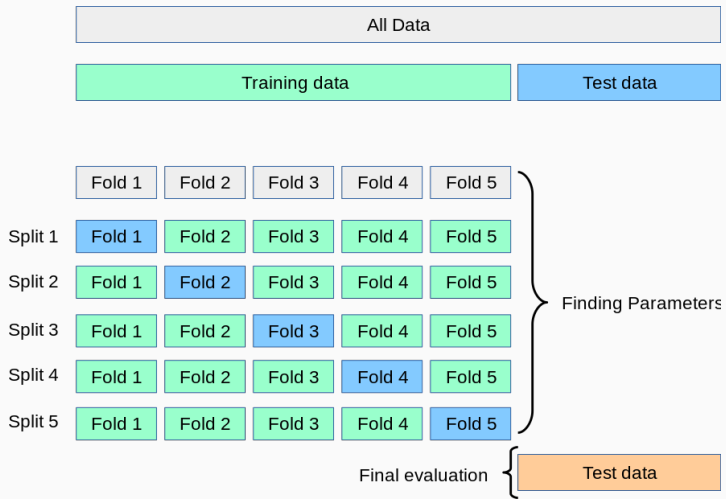
con: high variance, bad use of data

Implementing threefold split

```
X_trainval, X_test, y_trainval, y_test = train_test_split(X, y)
X_train, X_val, y_train, y_val = train_test_split(X_trainval, y_trainval)

val_scores = []
neighbors = np.arange(1, 15, 2)
for i in neighbors:
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train, y_train)
    val_scores.append(knn.score(X_val, y_val))
print("best validation score: {:.3f}".format(np.max(val_scores)))
best_n_neighbors = neighbors[np.argmax(val_scores)]
print("best n_neighbors:", best_n_neighbors)
```

Cross-validation + test set



.padding-top[

