Learning Scikit-Learn: Machine Learning in Python

http://homepages.dcc.ufmg.br/~ramon.pessoa/

Contact

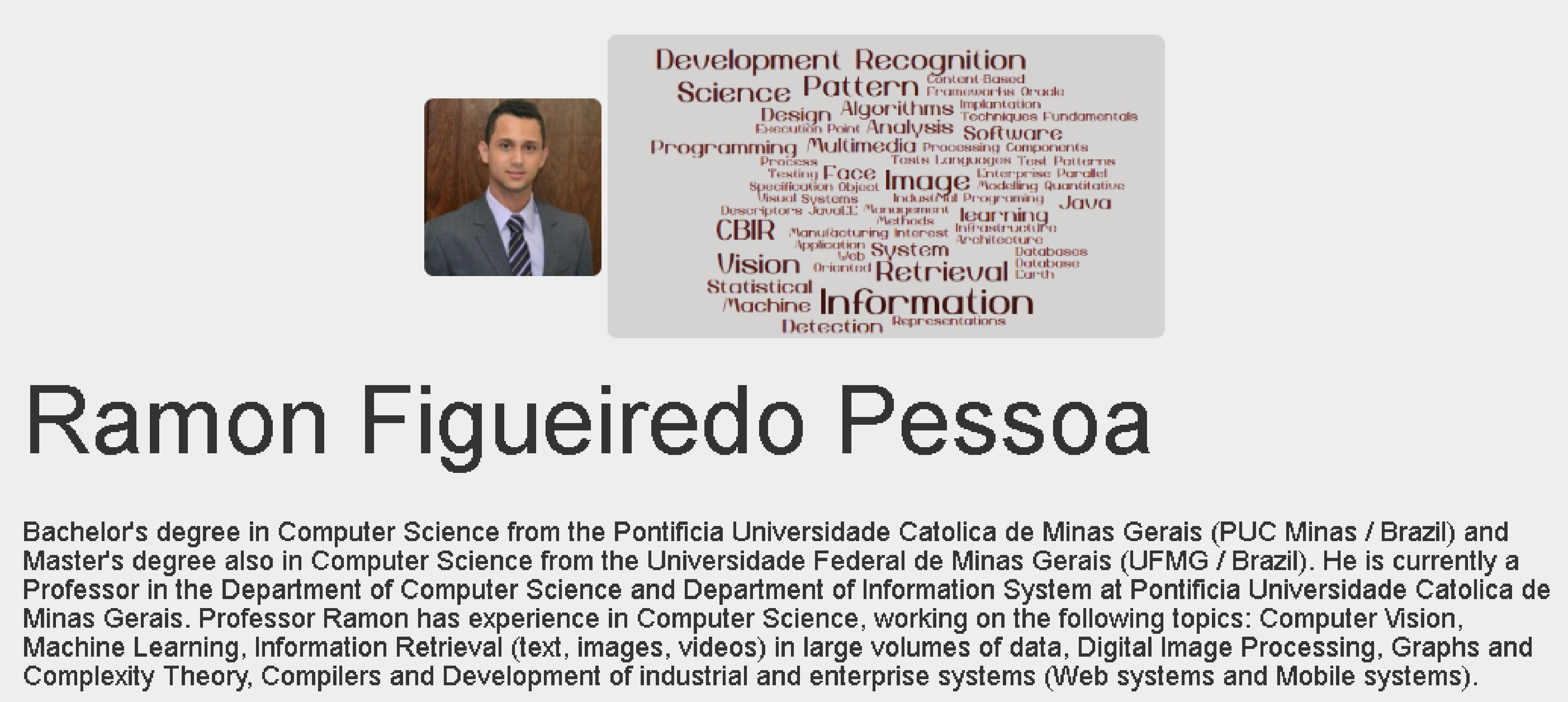
👤 Ramon Figueiredo Pessoa

📍 Av Dom José Gaspar, 500 Coração Eucarístico,
Belo Horizonte / MG 30535-610

✉ ramon.fgrd@gmail.com

Prof. Ramon Figueiredo Pessoa

http://homepages.dcc.ufmg.br/~ramon.pessoa/


Development Recognition
Science Pattern Content-Based
Frameworks Oracle
Design Algorithms Implantation
Techniques Fundamentals
Execution Point Analysis Software
Programming Multimedia Processing Components
Process Tools Languages Test Patterns
Testing Face Image Enterprise Parallel
Specification Object Modelling Quantitative
Visual Systems Industrial Programming Java
Descriptors JavaEE Management learning
Methods
CBIR Manufacturing Interest Infrastructure
Architecture
Application System
Web Databases
Vision Oriented Retrieval Database
Earth
Statistical
Machine Information
Detection Representations

Ramon Figueiredo Pessoa

Bachelor's degree in Computer Science from the Pontificia Universidade Catolica de Minas Gerais (PUC Minas / Brazil) and Master's degree also in Computer Science from the Universidade Federal de Minas Gerais (UFMG / Brazil). He is currently a Professor in the Department of Computer Science and Department of Information System at Pontificia Universidade Catolica de Minas Gerais. Professor Ramon has experience in Computer Science, working on the following topics: Computer Vision, Machine Learning, Information Retrieval (text, images, videos) in large volumes of data, Digital Image Processing, Graphs and Complexity Theory, Compilers and Development of industrial and enterprise systems (Web systems and Mobile systems).

Outline

- Introduction to Python
- Introduction to Machine Learning
- Introduction to Scikit-Learn
  - Scikit-Learn (http://scikit-learn.org/stable/)
  - Supervised Learning
  - Unsupervised Learning
  - Advanced Features

What you will learn (Scikit-Learn)?

1. Set up scikit-learn inside your Python environment
2. Classify objects (from documents to images) based on some of their features, using a variety of methods:
   - Support Vector Machines, Naïve Bayes
   - Decision Trees, Regression Techniques
   - K-Means and so on

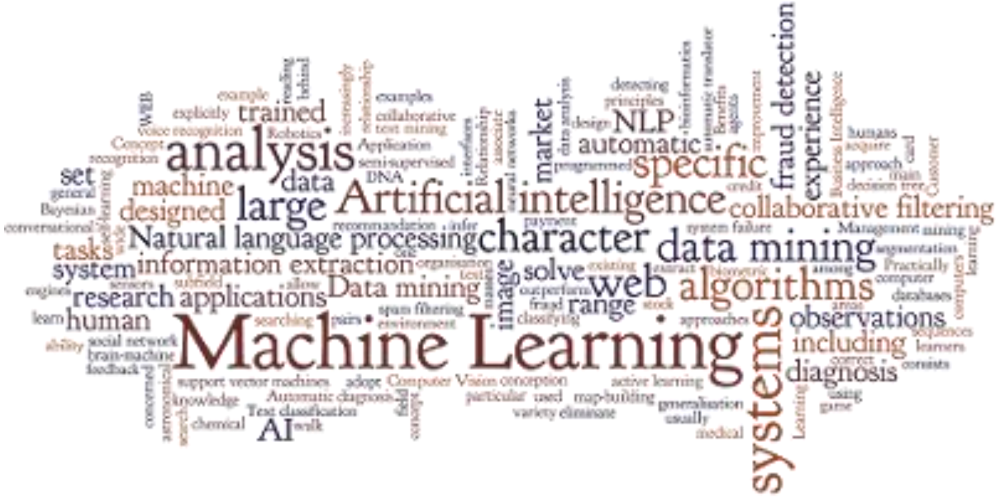
What you will learn?

(MeetUp – PUC Minas: June, 30 and July, 01)

3. Display and analyze groups in our data using dimensionality reduction **(MeetUp – PUC)**
4. Make use of different tools to preprocess, extract, and select the learning features **(MeetUp – PUC)**
5. Select the best parameters for our models using model selection **(MeetUp – PUC)**
6. Improve the way you build your models using parallelization techniques **(MeetUp – PUC)**

What is machine learning?

- Machine learning is a sub area of artificial intelligence which studies systems that can learn from data



What is machine learning?

- **Some examples**
  - Search on Google
  - Face Recognition (Facebook)
  - Classifier mail (Gmail)
  - Spam recognition in Emails
  - Robot Vision
  - Character Recognition (OCR)
  - Recommender Systems
  - Feelings Analysis

Problem

- The learning problem generally considers a set of ***n*** data samples and tries to predict an unknown sample
- The properties of a sample are generally called **features**
- They are categorized into:
  - Supervised Learning
  - Unsupervised Learning
- Note: There are other hybrid categories, such as semi-supervised learning

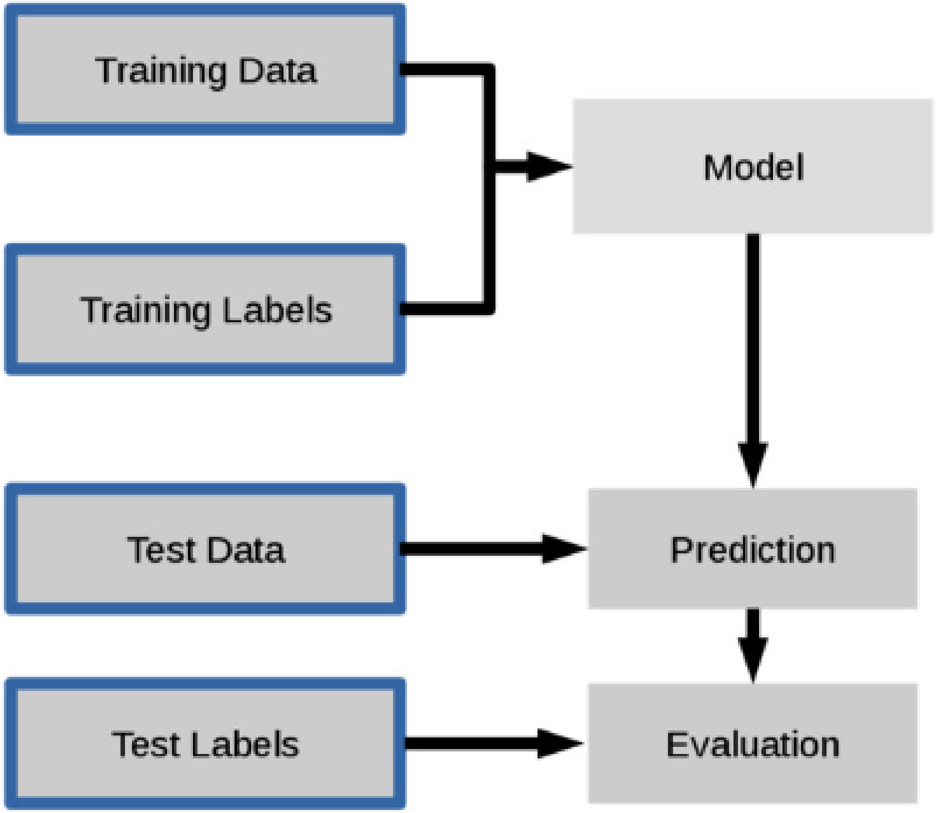
Supervised Learning

- In **supervised learning**, algorithms are trained with labeled data

- Example: Character Recognition, where the training is carried out with various samples of characters where each image contains a label (character)

Supervised Learning

scikit-learn
Machine Learning in Python



Training

Generalization

Supervised Learning

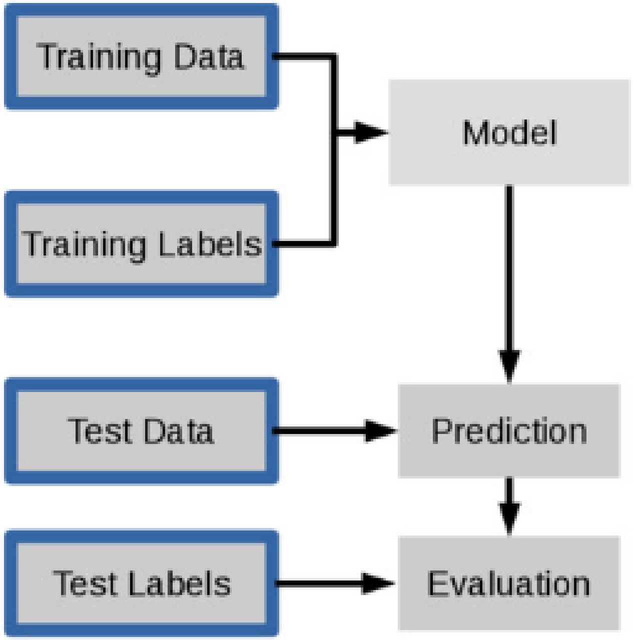
scikit-learn
Machine Learning in Python

```
clf = RandomForestClassifier()

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

clf.score(X_test, y_test)
```



Training Data

Training Labels

Model

Test Data

Prediction

Test Labels

Evaluation

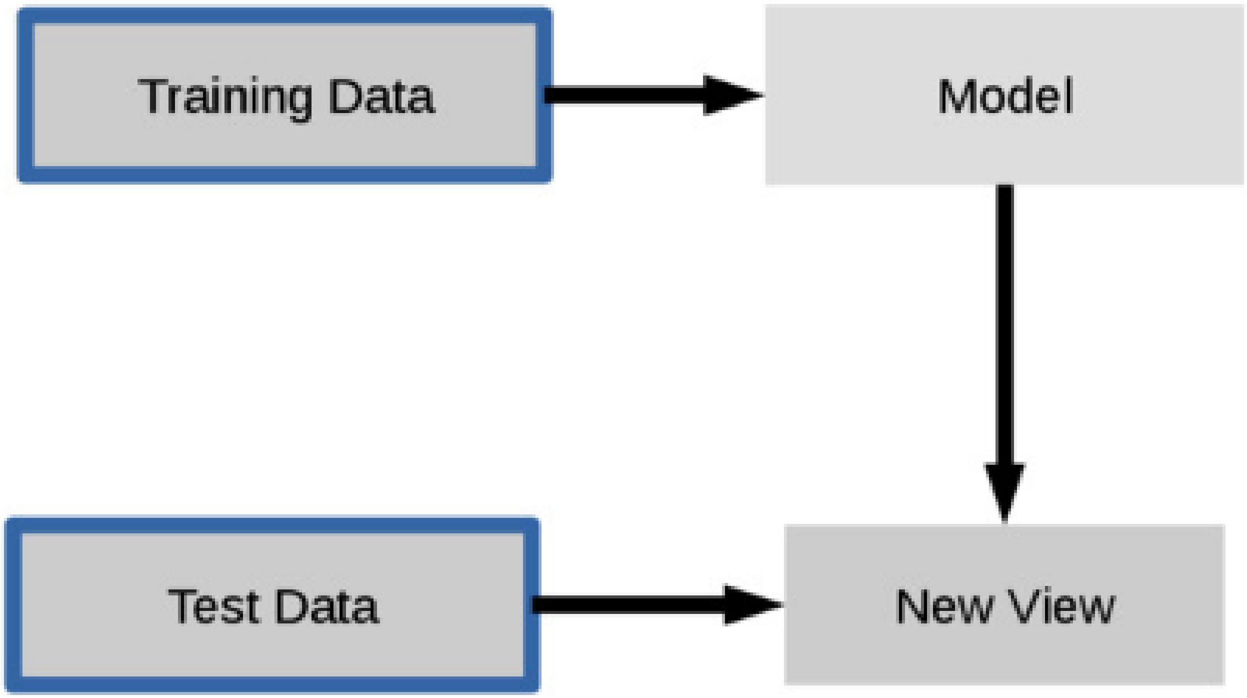
Unsupervised Learning

- In **unsupervised learning** algorithms operate on data unlabelled

- Example: clustering algorithm, where samples are grouped according to the level of similarity (To group similar images in an images dataset)

Unsupervised Learning

scikit-learn
Machine Learning in Python



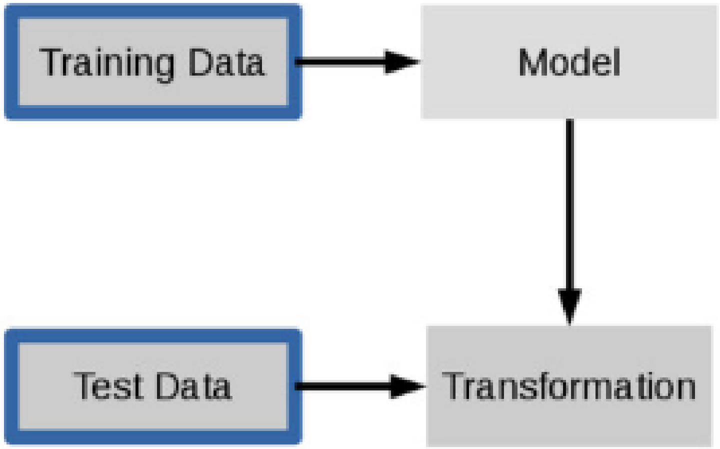
Unsupervised Learning

scikit-learn
Machine Learning in Python

```
pca = PCA()
```

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

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



Classification versus Regression

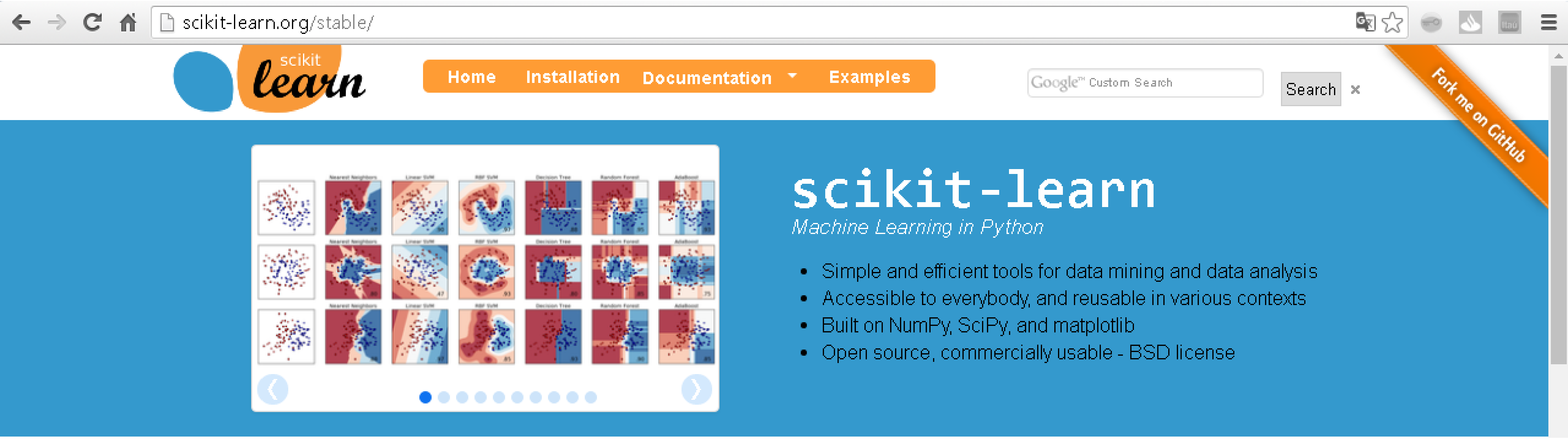
- Classification
  - The samples belonging to two or more classes (eg: span / non-span) and the goal is to learn from data already labeled what is the class of a new data not labeled
  - The classification can also be seen as a learning discrete values

Classification versus Regression

- Regression
  - If the expected output of the algorithm is one or more continuous variables, the problem is called regression
  - An example of regression is to predict the price of a house/apartment considering its features as size, room number, number of garages, etc

Scikit-Learn

- **Scikit-learn** é um framework open-source de machine learning escrito em Python utilizando as plataformas Numpy/Scipy e Matplotlib

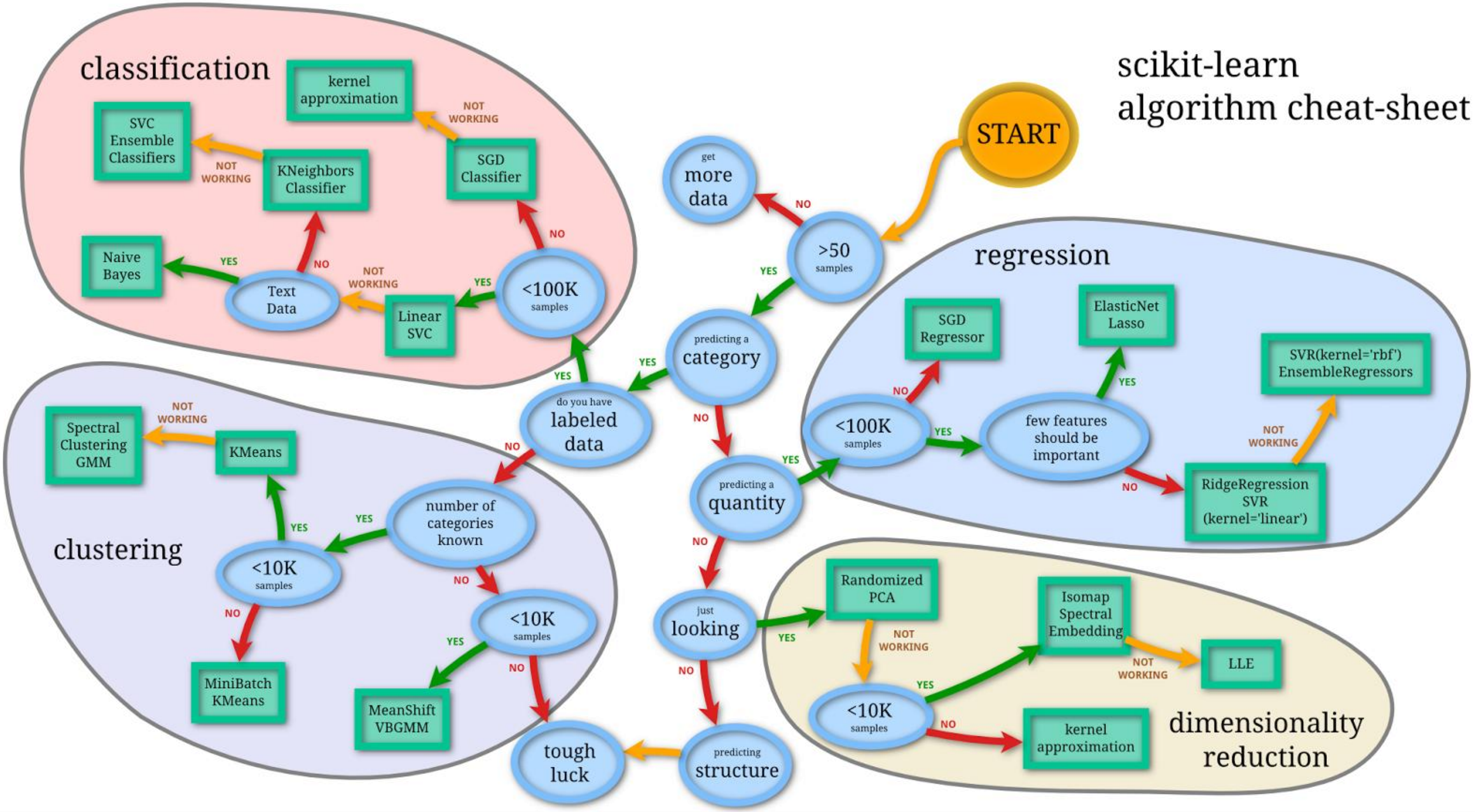


scikit-learn

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

Scikit-Learn



Installation on LINUX (Family based on Debian as Ubuntu)

1) To install standard packages using the command

```
sudo apt-get install build-essential python-dev python-numpy python-setuptools pythonscipy libatlas-dev python-pip
```

2) To install *matlibplot*

```
sudo apt-get install python-matplotlib
pip install libpng-dev libjpeg8-dev libfreetype6-dev
pip install matplotlib
```

3) To install *scikit-learn*

```
sudo pip intall ipython-notebook
```

4) To install IPython Notebook

```
sudo apt-get install ipython-notebook
pip install ipython
pip install tornado
pip install pyzmq
```

Matplotlib

We're updating the default styles for Matplotlib 2.0

Learn what to expect in the **new updates**

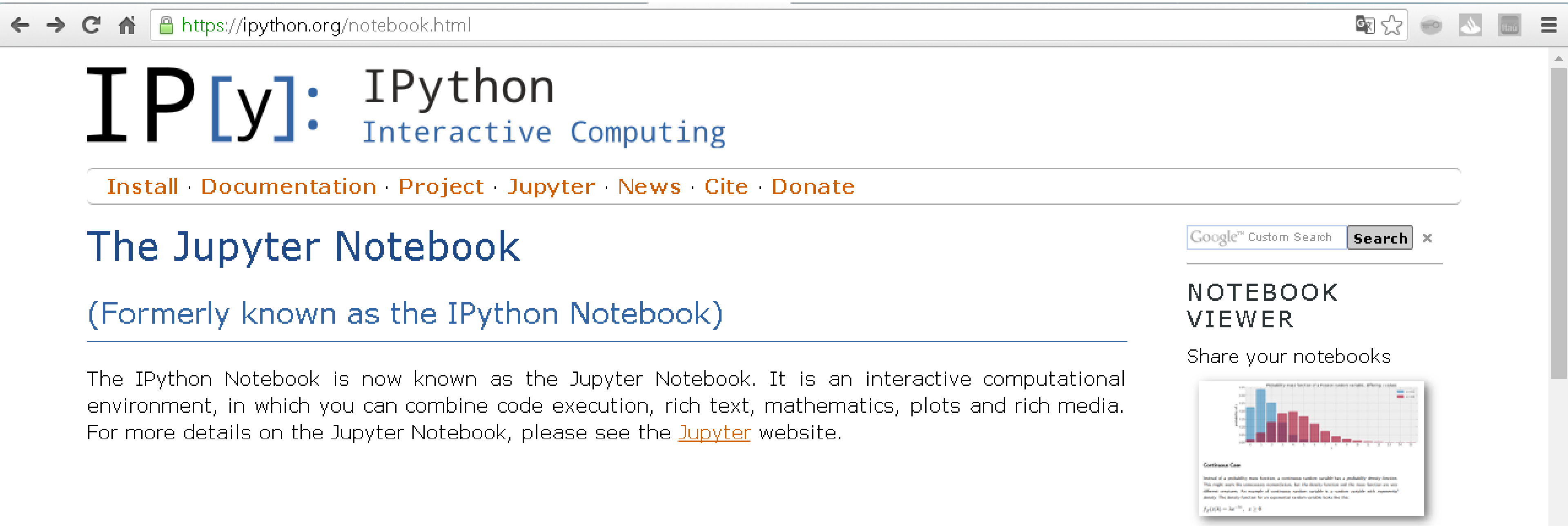
Introduction

matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. matplotlib can be used in python scripts, the python and ipython shell (ala MATLAB®* or Mathematica®†), web application servers, and six graphical user interface toolkits.

IPython Notebook

- Note: To run IPython Notebook in your browser, you must run the following command to open the:

```
ipython notebook
```



https://ipython.org/notebook.html
IP[y]: IPython
Interactive Computing
Install · Documentation · Project · Jupyter · News · Cite · Donate
Google™ Custom Search
Search

The Jupyter Notebook

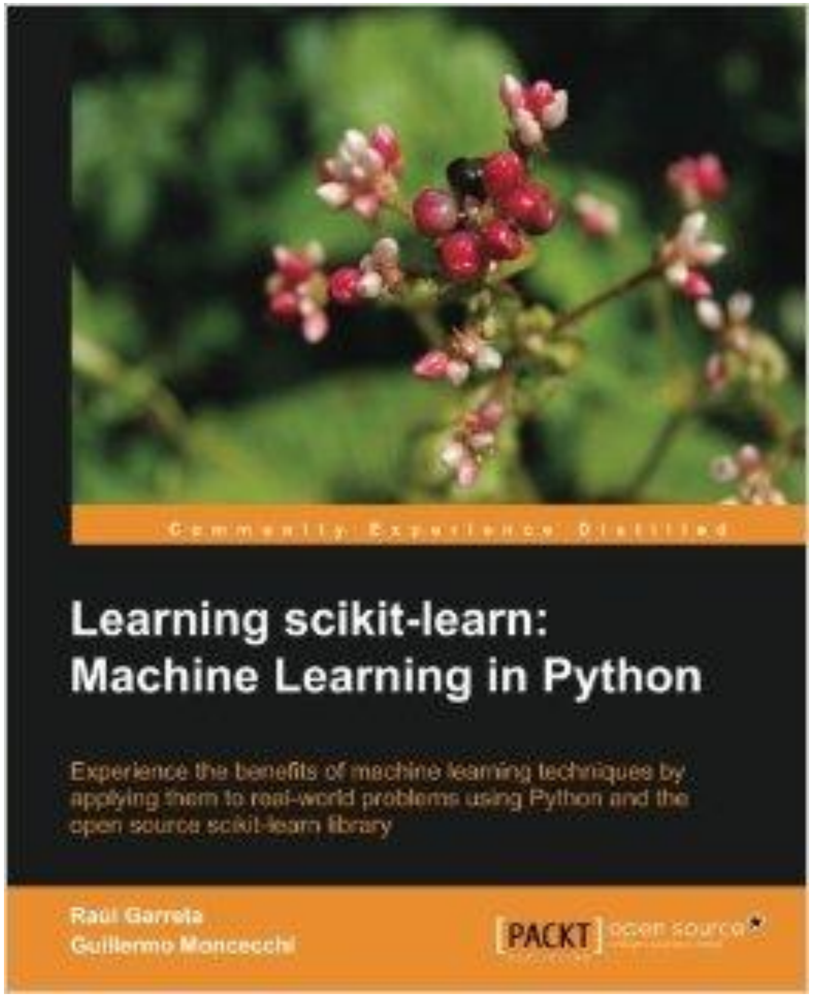
(Formerly known as the IPython Notebook)

The IPython Notebook is now known as the Jupyter Notebook. It is an interactive computational environment, in which you can combine code execution, rich text, mathematics, plots and rich media. For more details on the Jupyter Notebook, please see the Jupyter website.

NOTEBOOK VIEWER

Share your notebooks

Books



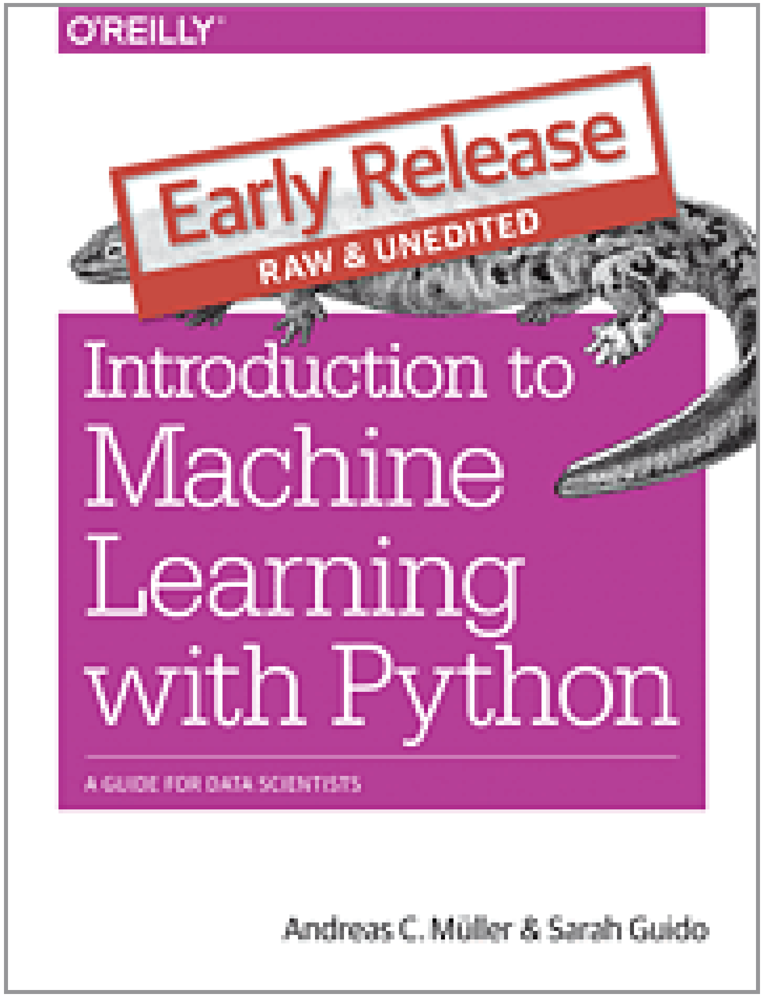
Community Experience Distilled

Learning scikit-learn: Machine Learning in Python

Experience the benefits of machine learning techniques by applying them to real-world problems using Python and the open source scikit-learn library

Raúl Garreta
Guillermo Moncecchi

PACKT open source



O'REILLY

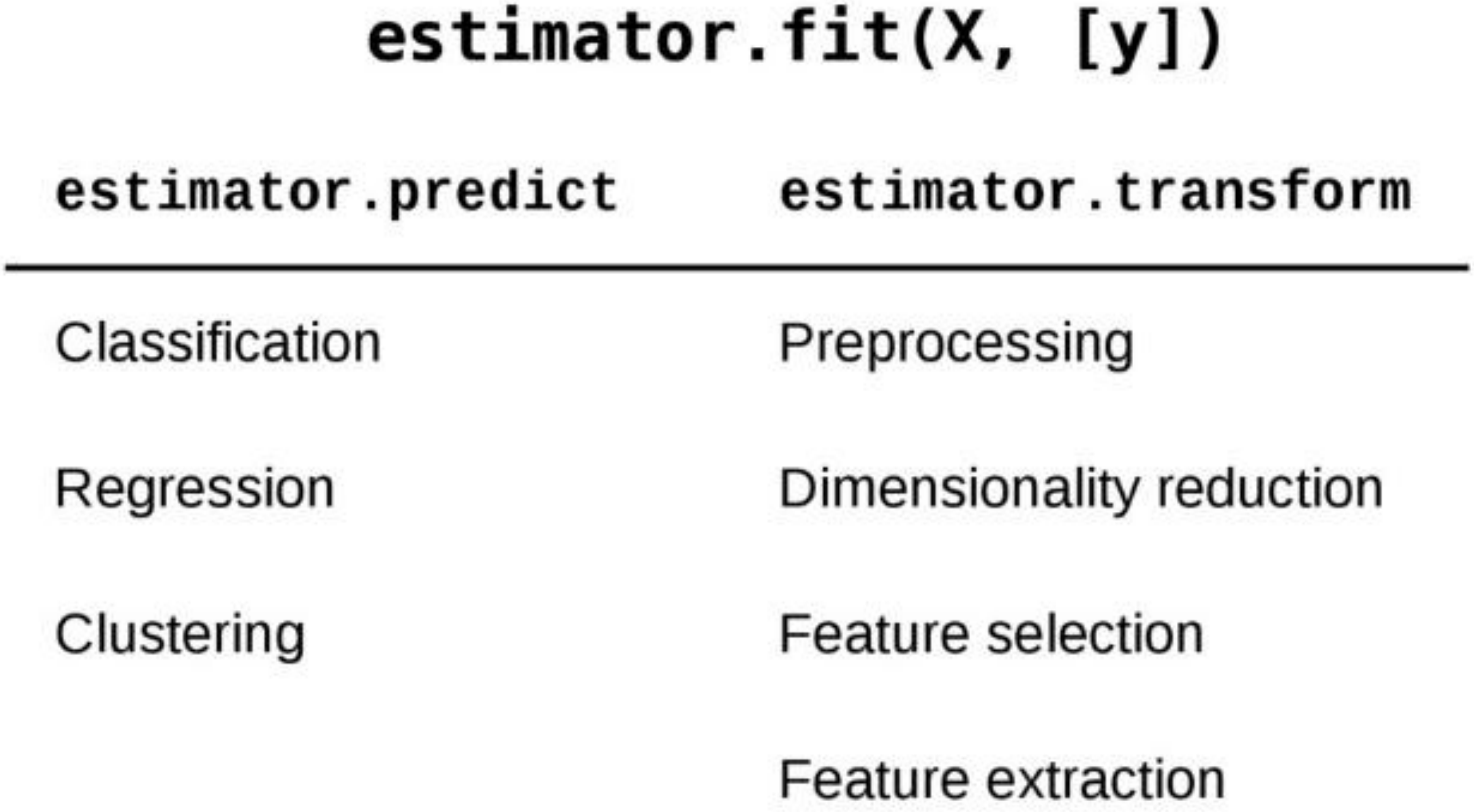
Early Release
RAW & UNEDITED

Introduction to Machine Learning with Python

A GUIDE FOR DATA SCIENTISTS

Andreas C. Müller & Sarah Guido

Basic API

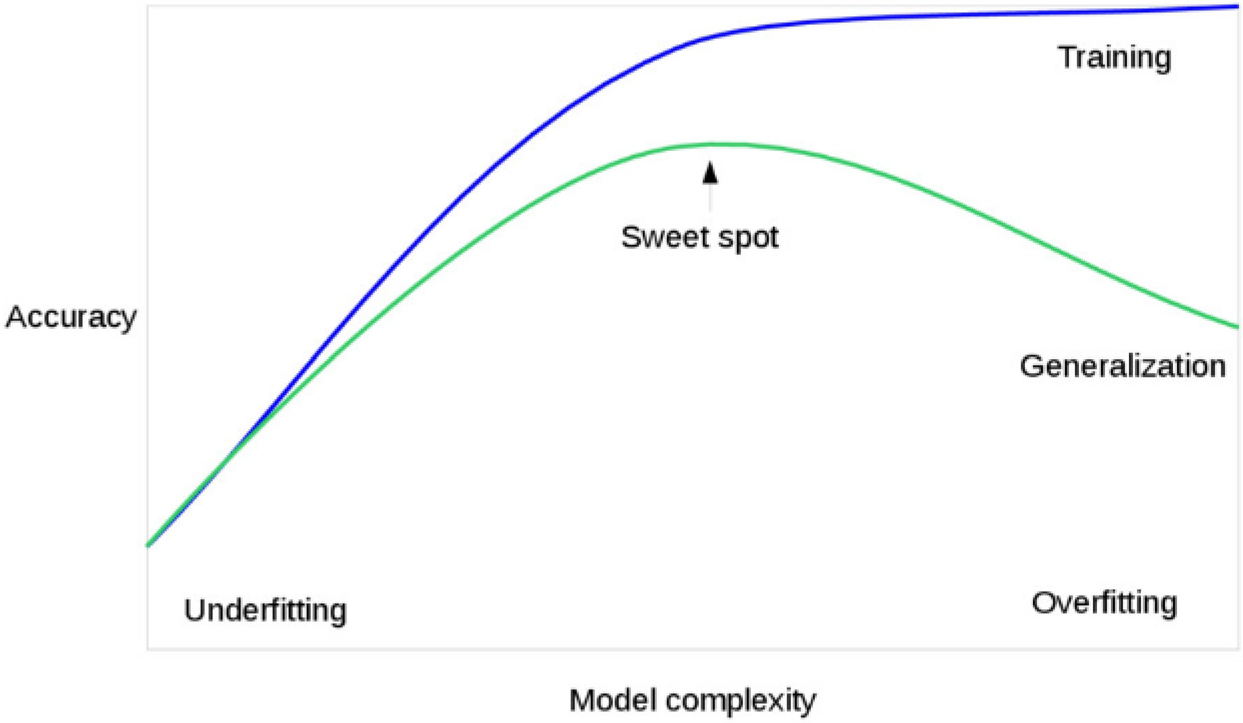


```
estimator.fit(X, [y])
```

estimator.predict	estimator.transform
Classification	Preprocessing
Regression	Dimensionality reduction
Clustering	Feature selection
	Feature extraction

Overfitting and Underfitting

- Overfitting and underfitting are the two biggest causes for poor performance of machine learning algorithms



Overfitting and Underfitting

- In **overfitting**, a statistical model describes random error or noise instead of the underlying relationship

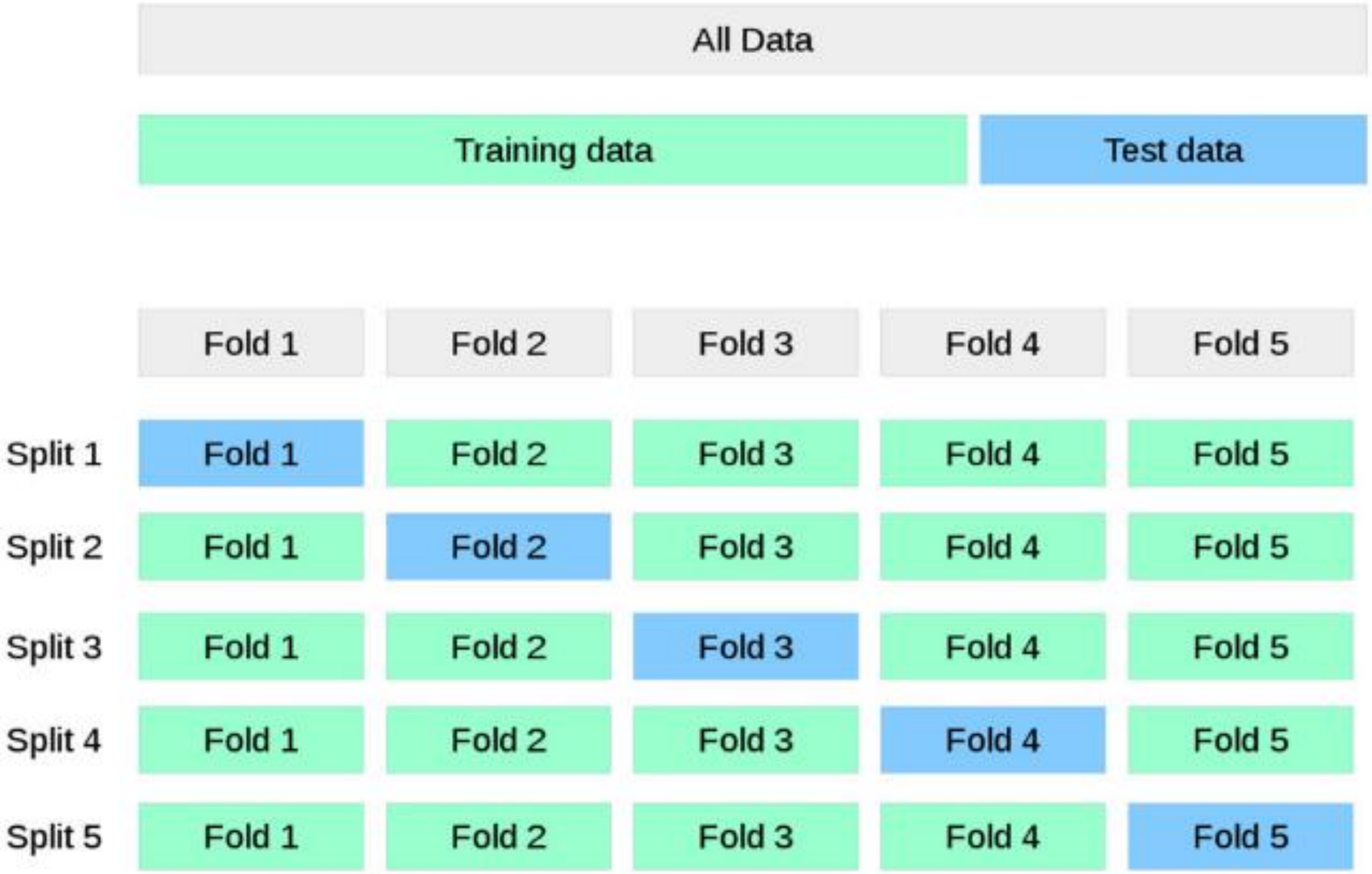
- Overfitting occurs when a model is excessively complex, such as having too many parameters relative to the number of observations

Overfitting and Underfitting

- **Underfitting** refers to a model that can neither model the training data not generalize to new data

- An underfit machine learning model is not a suitable model and it will have poor performance on the training data

Training data Vs Test data

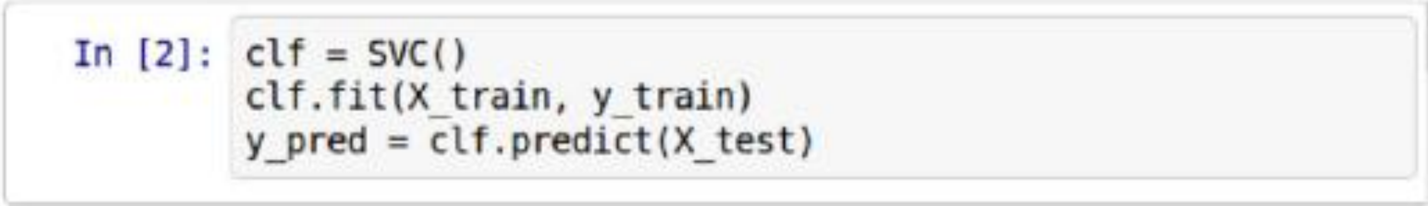


All Data

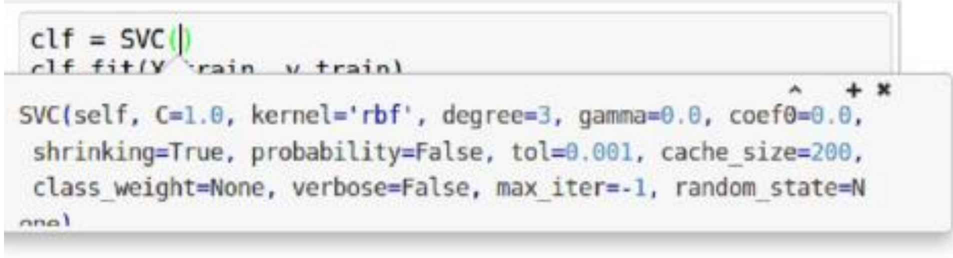
Training data | Test data

	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
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Cross-validation



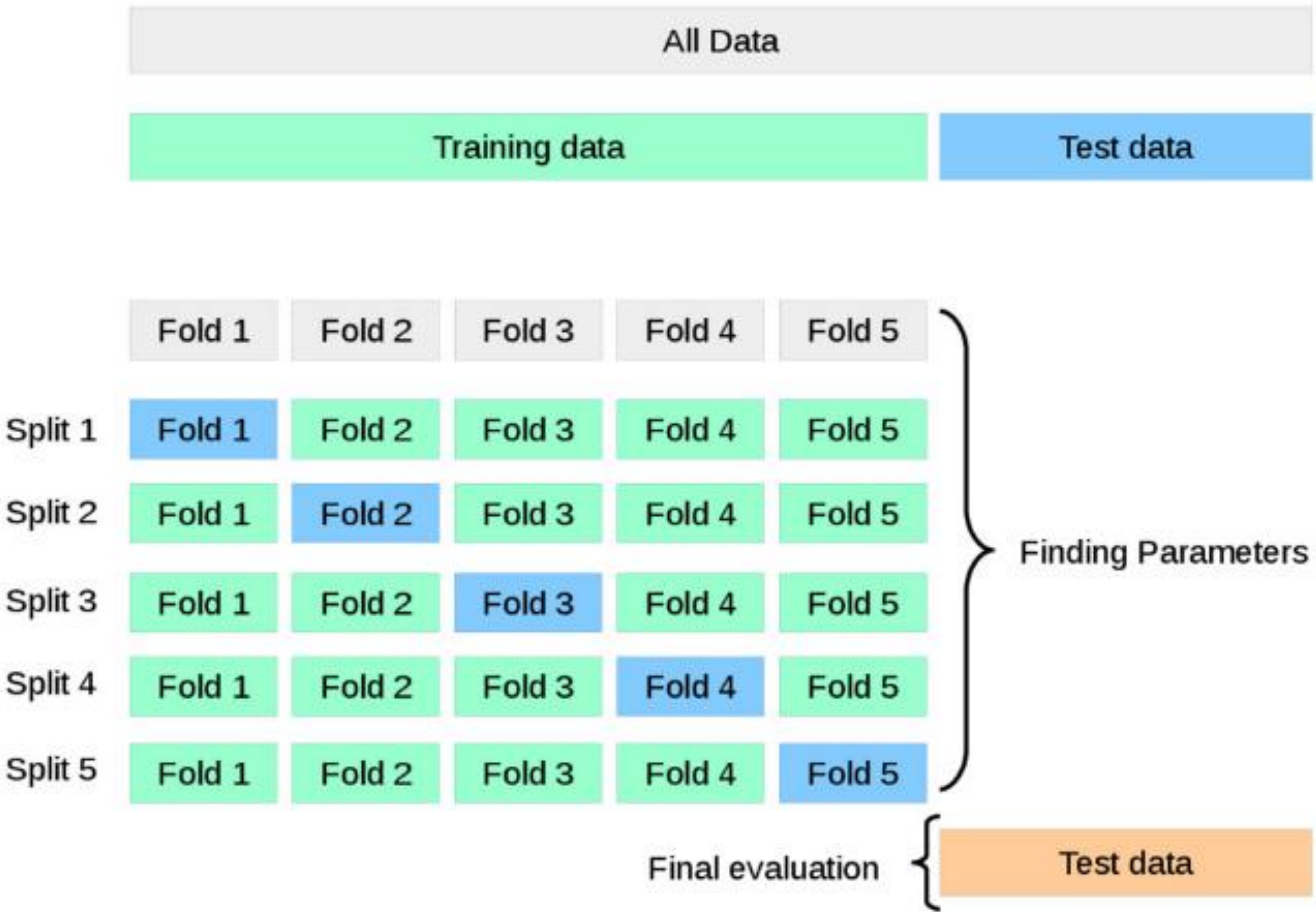
```
In [2]: clf = SVC()
        clf.fit(X_train, y_train)
        y_pred = clf.predict(X_test)
```



```
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)
```

Cross-validation



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
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Finding Parameters

Final evaluation

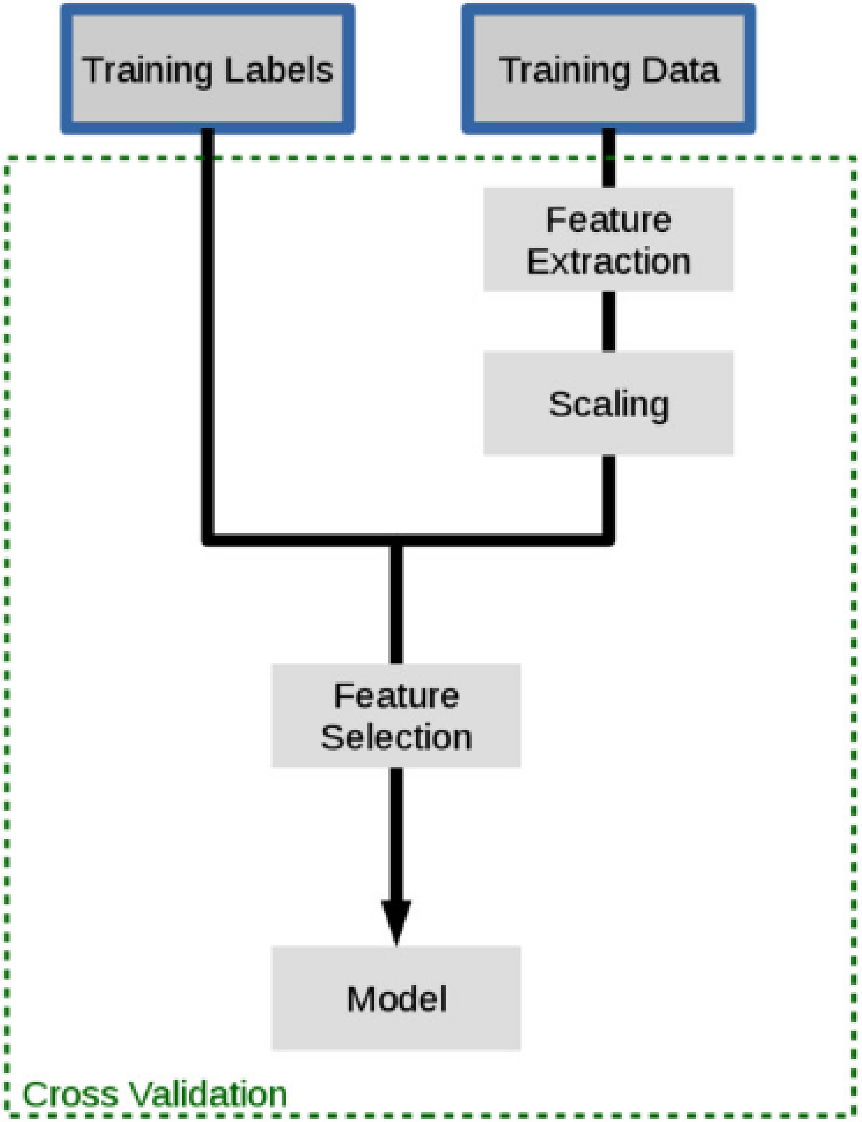
Test data

Grid search

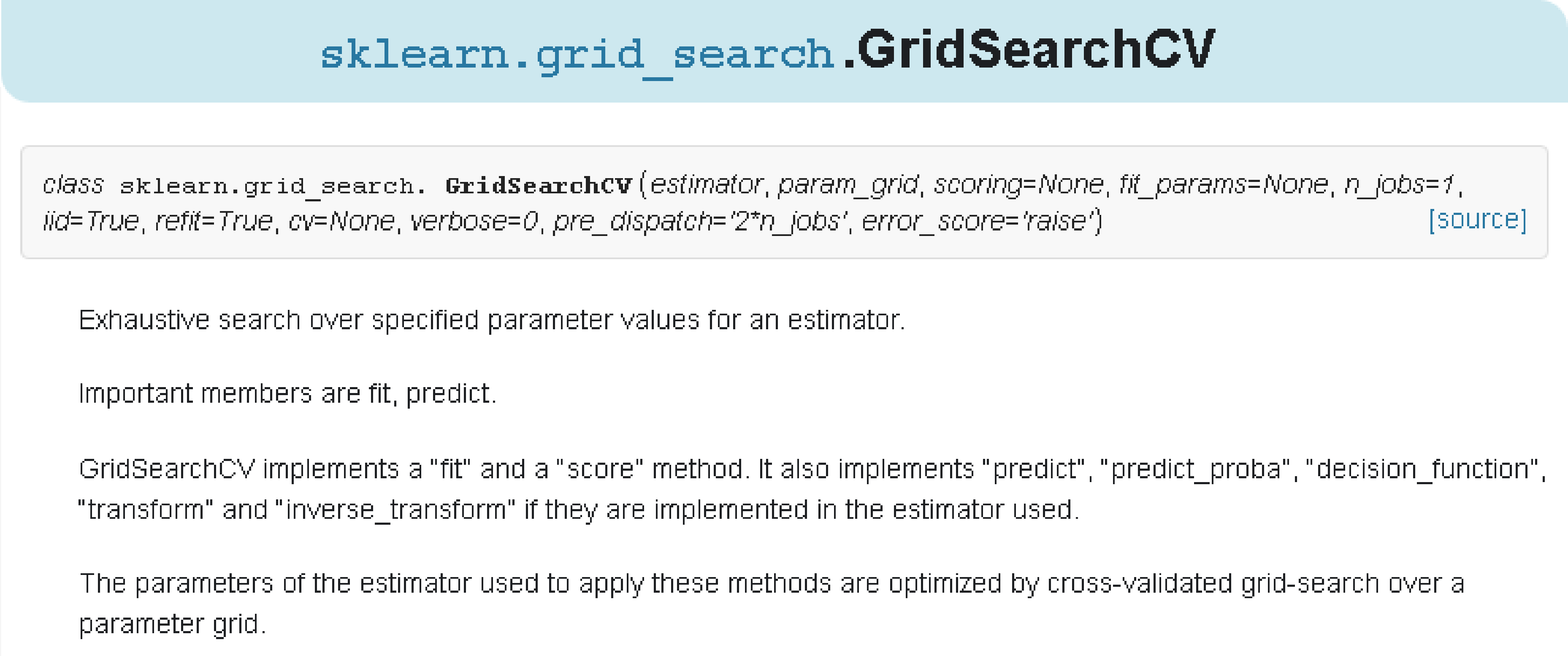
- How to Evaluate Machine Learning Models?

- Grid search picks out a grid of hyperparameter values, evaluates every one of them, and returns the winner

Grid search


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

Grid search



sklearn.grid_search.GridSearchCV

class sklearn.grid_search. **GridSearchCV** (*estimator*, *param_grid*, *scoring=None*, *fit_params=None*, *n_jobs=1*, *iid=True*, *refit=True*, *cv=None*, *verbose=0*, *pre_dispatch='2*n_jobs'*, *error_score='raise'*) [source]

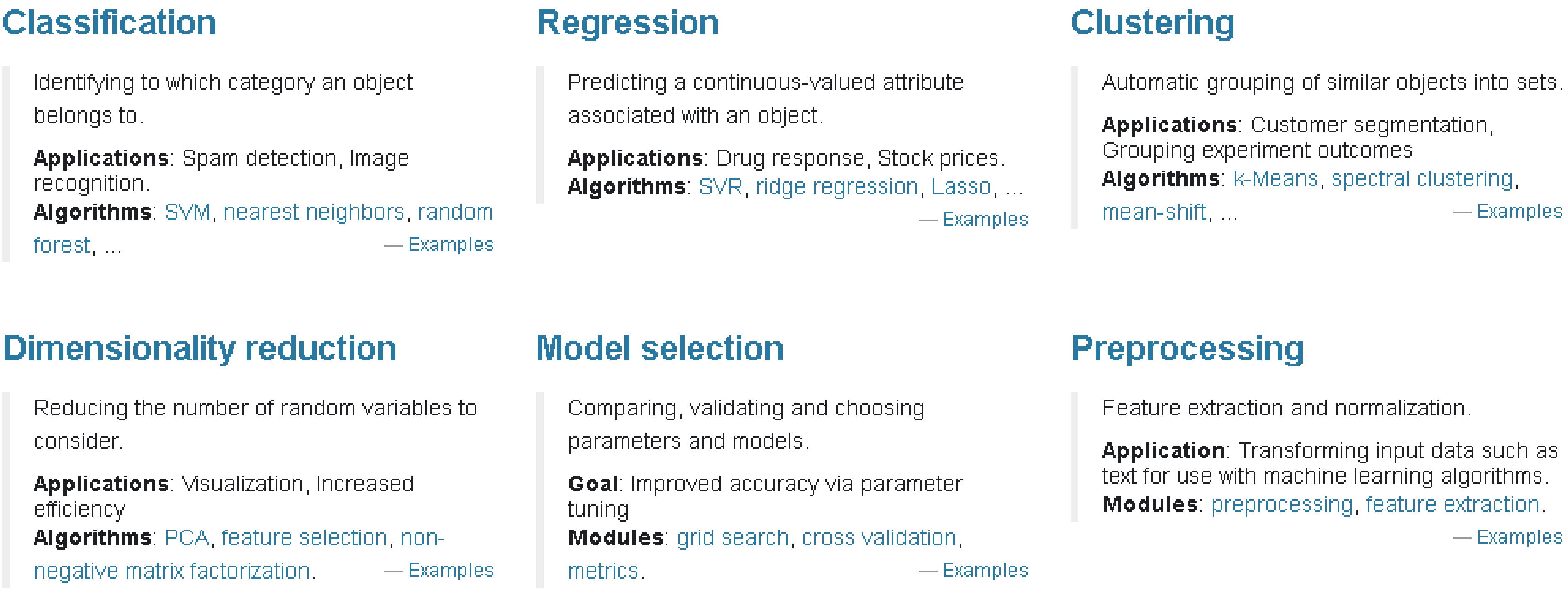
Exhaustive search over specified parameter values for an estimator.

Important members are fit, predict.

GridSearchCV implements a "fit" and a "score" method. It also implements "predict", "predict_proba", "decision_function", "transform" and "inverse_transform" if they are implemented in the estimator used.

The parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid.

Scikit-Learn: Supported Algorithms



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.

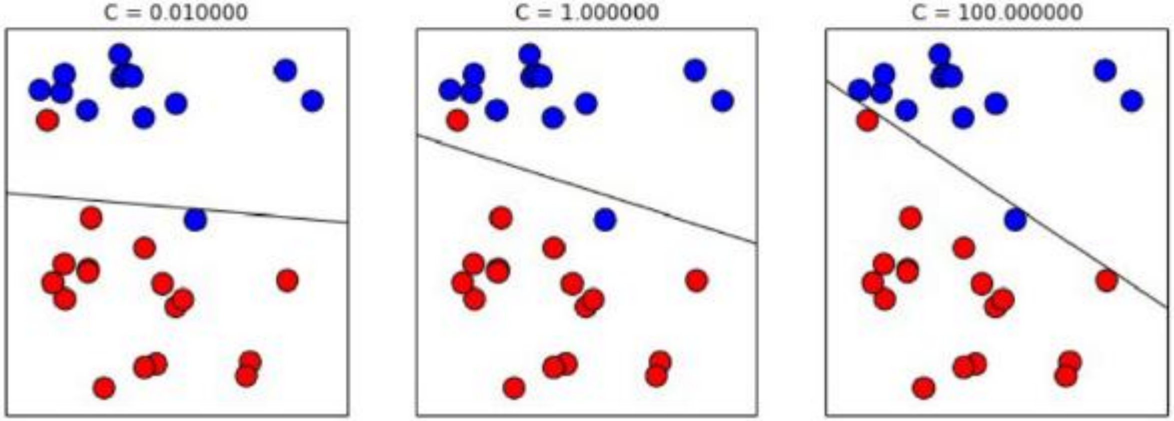
Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction.

— Examples

Scikit-Learn: Supported Algorithms

- Linear SVM

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

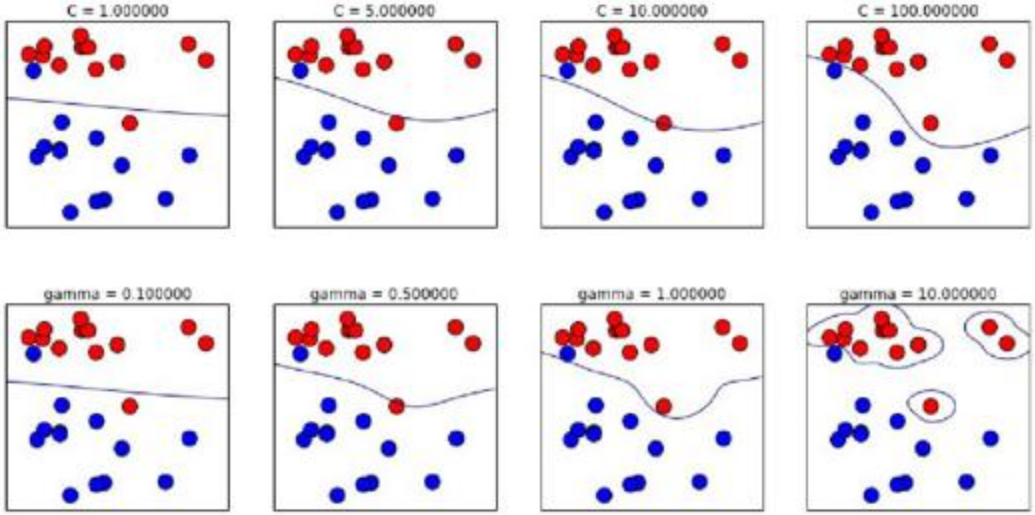


Scikit-Learn: Supported Algorithms

- (RBF) Kernel SVM

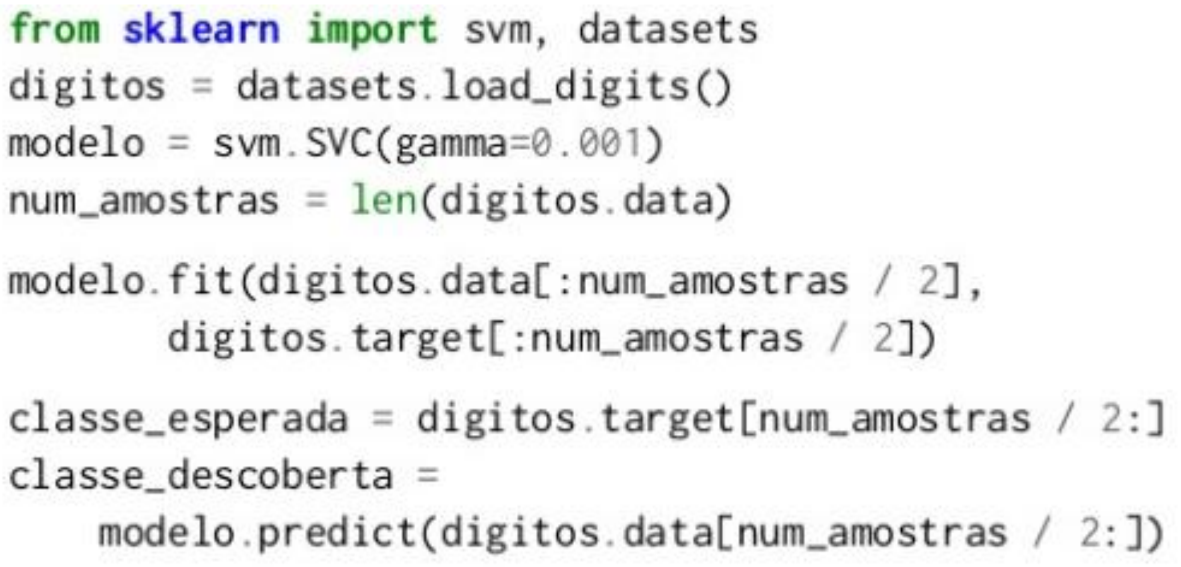
$$\hat{y} = \mathrm{sign}(\alpha_0 + \sum_j \alpha_j y_j k(\mathbf{x}^{(\mathbf{j})}, \mathbf{x}))$$

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



Scikit-Learn: Supported Algorithms

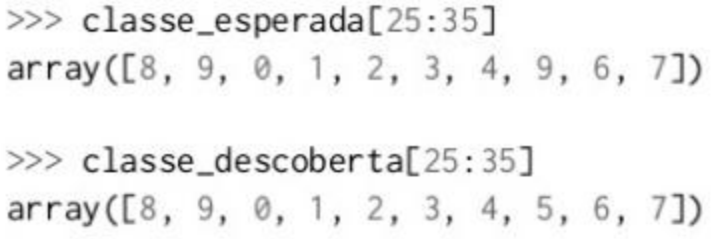
- Scikit-Learn + SVM



```
from sklearn import svm, datasets
digitos = datasets.load_digits()
modelo = svm.SVC(gamma=0.001)
num_amostras = len(digitos.data)

modelo.fit(digitos.data[:num_amostras / 2],
       digitos.target[:num_amostras / 2])

classe_esperada = digitos.target[num_amostras / 2:]
classe_descoberta =
    modelo.predict(digitos.data[num_amostras / 2:])
```

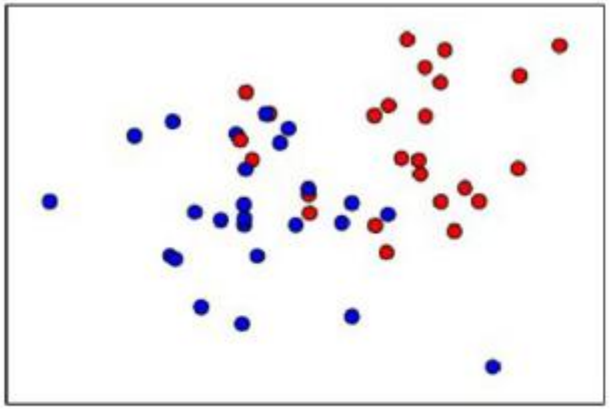


```
>>> classe_esperada[25:35]
array([8, 9, 0, 1, 2, 3, 4, 9, 6, 7])

>>> classe_descoberta[25:35]
array([8, 9, 0, 1, 2, 3, 4, 5, 6, 7])
```

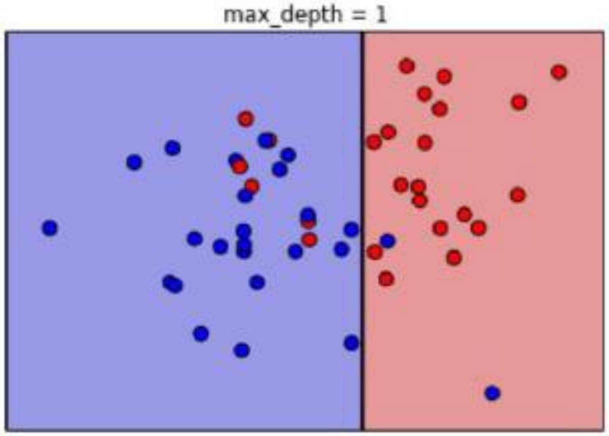
Scikit-Learn: Supported Algorithms

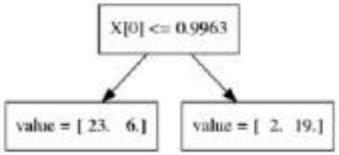
- Decision Trees



Scikit-Learn: Supported Algorithms

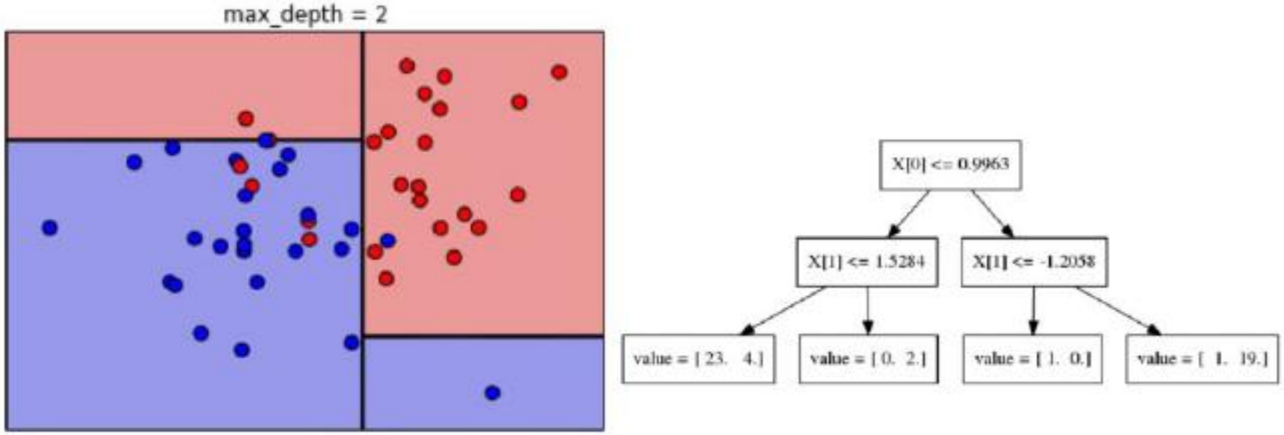
- Decision Trees





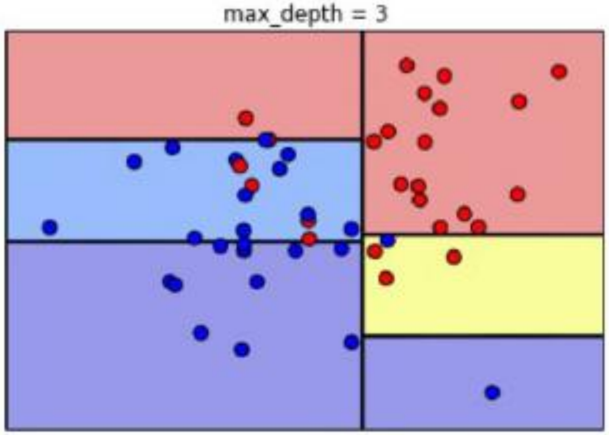
Scikit-Learn: Supported Algorithms

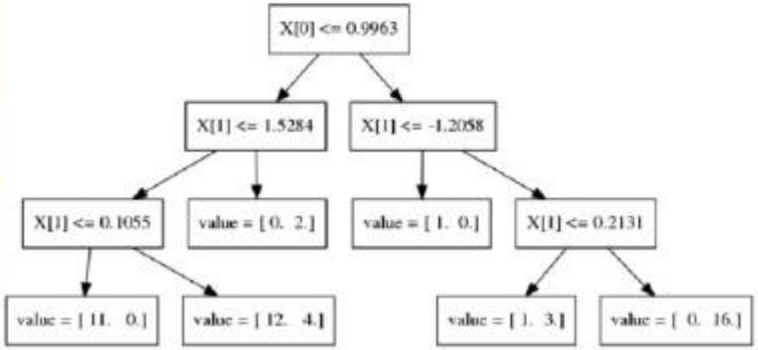
- Decision Trees



Scikit-Learn: Supported Algorithms

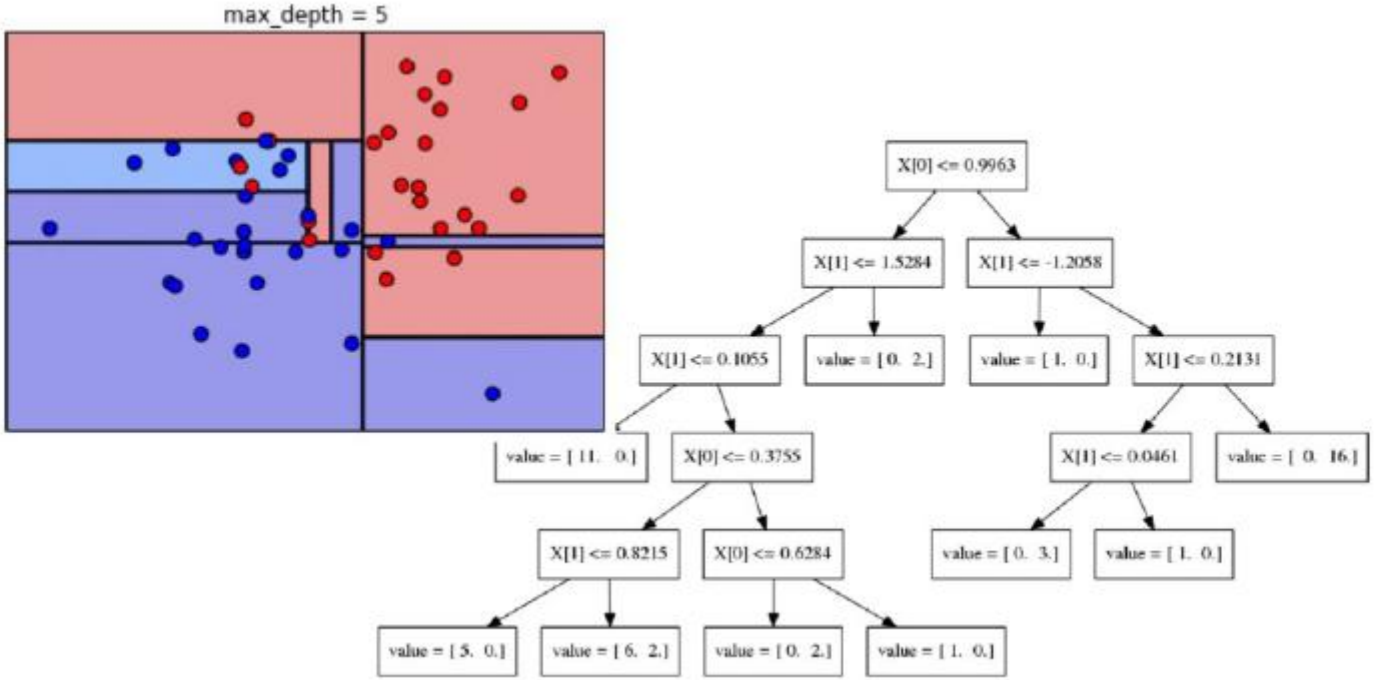
- Decision Trees





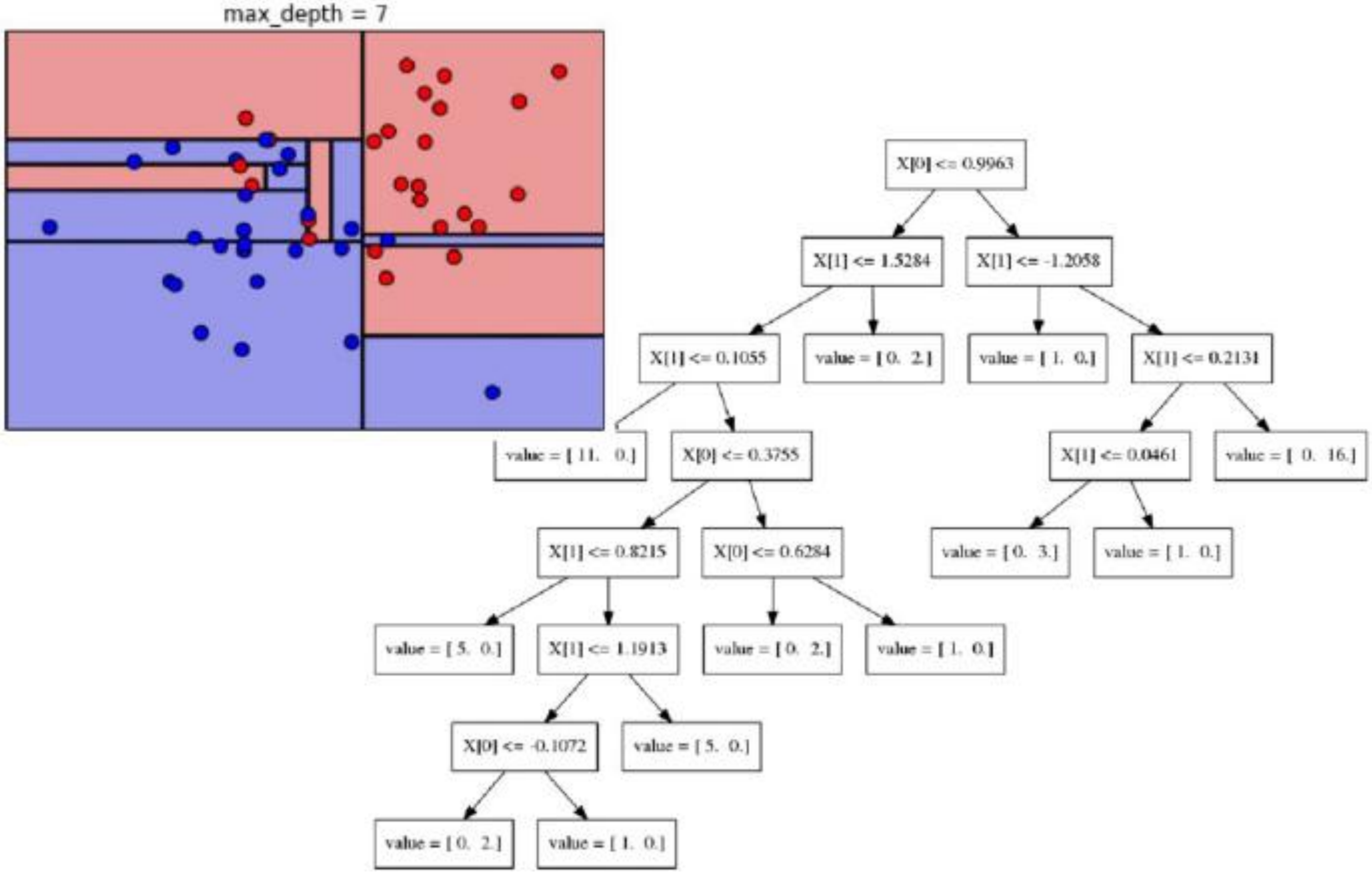
Scikit-Learn: Supported Algorithms

- Decision Trees



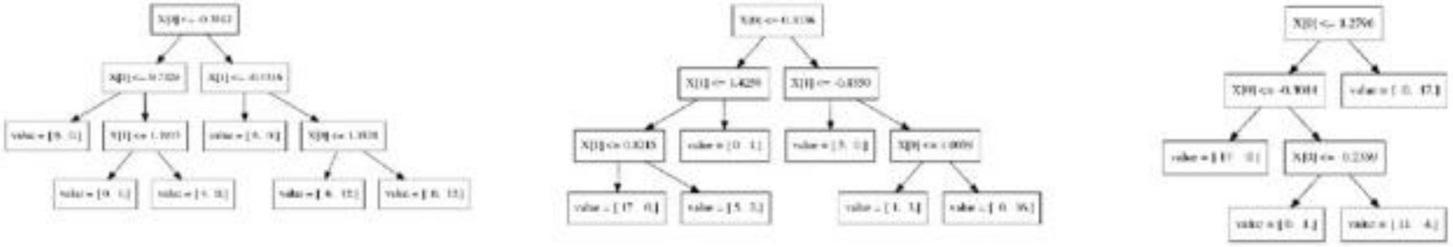
Scikit-Learn: Supported Algorithms

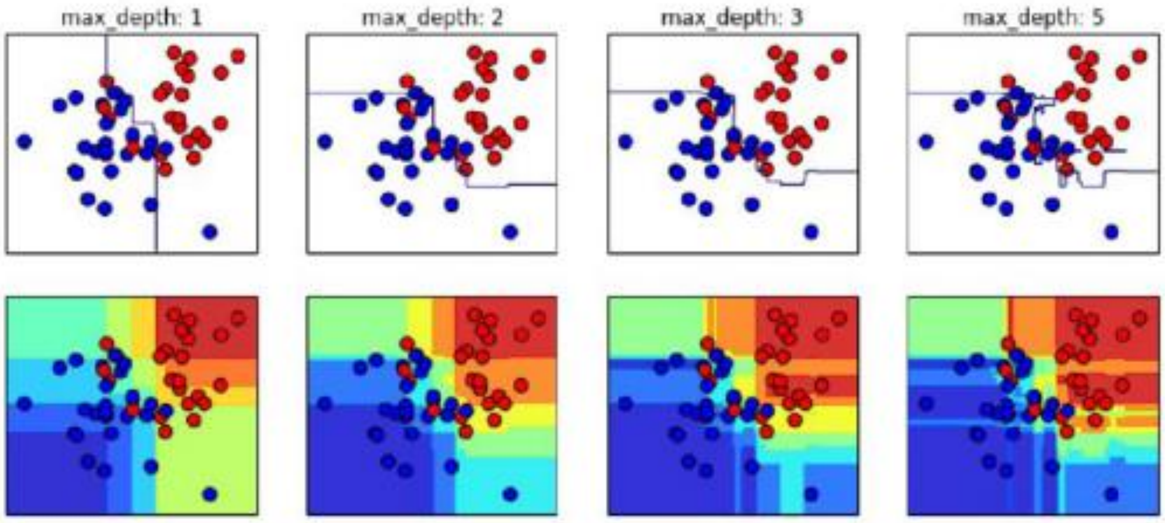
- Decision Trees



Scikit-Learn: Supported Algorithms

- Random Forests



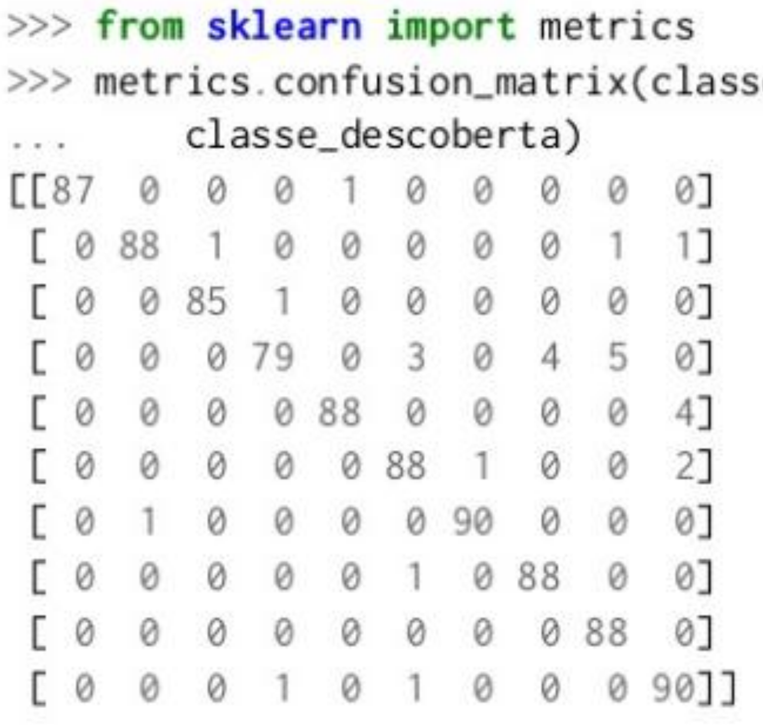


Scikit-Learn: Supported Algorithms

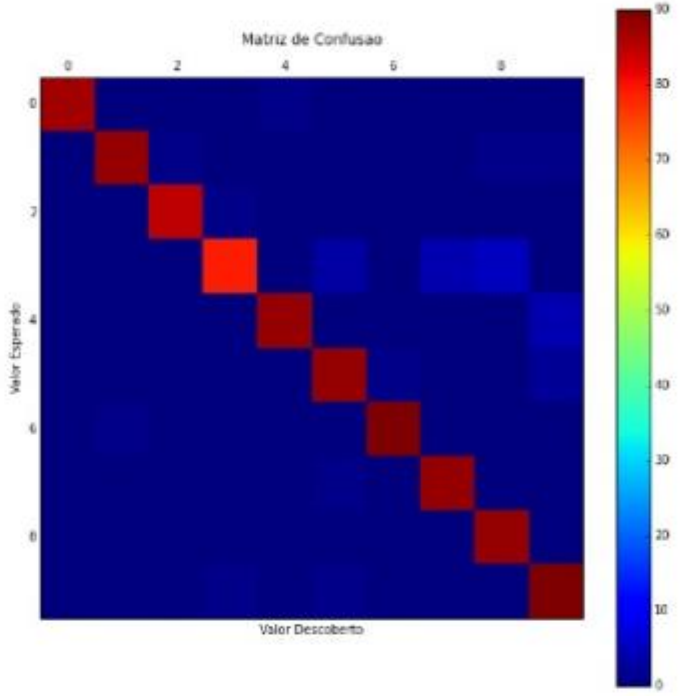
- Random Forests
  - Is a notion of the general technique of random decision forests that are an ensemble learning method for classification, regression and other tasks
  - Random Forest operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees

Scikit-Learn: Supported Algorithms

- Confusion Matrix
  - One way to evaluate how well a model behaves



```
>>> from sklearn import metrics
>>> metrics.confusion_matrix(classe_esperada,
...        classe_descoberta)
[[87  0  0  0  1  0  0  0  0  0]
 [ 0 88  1  0  0  0  0  0  1  1]
 [ 0  0 85  1  0  0  0  0  0  0]
 [ 0  0  0 79  0  3  0  4  5  0]
 [ 0  0  0  0 88  0  0  0  0  4]
 [ 0  0  0  0  0 88  1  0  0  2]
 [ 0  1  0  0  0  0 90  0  0  0]
 [ 0  0  0  0  0  1  0 88  0  0]
 [ 0  0  0  0  0  0  0  0 88  0]
 [ 0  0  0  1  0  1  0  0  0 90]]
```



Scikit-Learn: Supported Algorithms

Supervised learning	Unsupervised learning
1. Generalized Linear ModelsGeneralized Linear Models 2. Linear and Quadratic Discriminant Analysis 3. Kernel ridge regression 4. Support Vector Machines 5. Stochastic Gradient Descent 6. Nearest Neighbors 7. Gaussian Processes 8. Cross decomposition 9. Naive Bayes 10. Decision Trees 11. Ensemble methods 12. Multiclass and multilabel algorithms 13. Feature selection 14. Semi-Supervised 15. Isotonic regression 16. Probability calibration	1. Gaussian mixture models 2. Manifold learning 3. Clustering 4. Biclustering 5. Decomposing signals in components (matrix factorization problems) 6. Covariance estimation 7. Novelty and Outlier Detection 8. Density Estimation 9. Neural network models (unsupervised)

References

- Scikit-Learn:
  - http://scikit-learn.org/
- Machine Learning com Python e Scikit-learn
  - http://pt.slideshare.net/perone/intro-ml-slides20min
- Machine Learning With Scikit-Learn ODSC SF 2015
  - https://speakerdeck.com/amueller/machine-learning-with-scikit-learn-odsc-sf-2015
- Raul Garreta and Guillermo Moncecchi. **Learning scikit-learn: machine learning in python.** Packt Publishing Ltd, 2013
- Andreas C. Mueller and Sarah Guido. **Introduction to Machine Learning with Python. A Guide for Data Scientists.** By Publisher: O'Reilly Media, June 2016