Algorithms for Data Analysis

Julien Tissier

November 24, 2017

Overview

1. Introduction to Data Analysis
   - Definition
   - Tools and libraries
   - Numpy
   - Pandas
   - Matplotlib
2. Machine Learning
   - Definition
   - Supervised vs. Unsupervised
   - Training, test & validation sets
   - Overfitting/Underfitting
3. Supervised Learning
   - k-Nearest Neighbors
   - Linear Models

What is data analysis?

Data

Pieces of information (measurement, values, facts...) that can be :

- structured (matrices, tabular data, RDBMS, time series...)
- unstructured (news articles, webpages, images/video...)

Data Analysis

Process of preparing, transforming and using models to find more information from data, as well as visualizing results.

How to analyze data?

There are usually two steps in data analysis. The first one is to find and **develop models** that can extract useful information from data (with languages like R or MATLAB). The second one is to **develop programs** that can be used in production systems (with languages like Java or C++).

With a growing popularity among scientists as well as the development of efficient libraries (numpy, pandas), **Python** became a great tool for data analysis. Python has many advantages :

- great for string/data processing
- can be used for both prototyping/production
- has a lot of existing libraries
- can easily integrate C/C++/FORTRAN legacy code
- easy to read/develop

Libraries

This course will be based on Python 3.5 (or above) and the following libraries :

- IPython (6.2+) : enhanced Python shell
- numpy (1.13+) : fast/efficient arrays and operations
- pandas (0.20+) : data structures (Series/DataFrame)
- matplotlib (2.1.0+) : plots and 2D visualization
- scipy (0.19+) : scientific algorithms
- scikit-learn (0.19+) : machine learning algorithms

Jupyter Notebook will also be used to give you samples of code, as they provide a more interactive way to learn and discover how these libraries work.

Numpy

Numpy (**Num**erical **Py**thon) is a high performance scientific computing library that can be used for matrices computations, Fourier transforms, linear algebra, statistical computations...

The main type of data in Numpy is the **ndarray** :

- n-dimensional array
- fixed size
- homogeneous datatypes
- similar to C arrays (continuous block of memory)

Why is Numpy efficient?

As a high level language, Python is slow to do any heavy computations, especially if very large arrays are involved. Numpy solves this problem thanks to the ndarray datatypes :

- efficient memory management (continuous block)
- use C loops instead of Python loops for computations on array
- vectorized operations (computations are done block by block, not element by element)
- rely on low-level routines for some operations (BLAS/LAPACK)

Numpy

Example

```
import numpy as np

a = np.array([1, 2, 3, 4])
b = np.array([6, 7, 8, 9])

c = a * b

d = np.array([[1, 2], [3, 4]])
```

Pandas

Pandas is a high-performance Python library used to work with data and analyze them. It contains many pre-implemented methods to read and parse data, as well as common statistical computations (mean, variance, correlation...).

Pandas has two main datatypes:

- Series : one-dimensional container. Indexes can be integers (like an array) or other objects (string, date...)
- DataFrame : tabular data, like a spreadsheet. It contains multiple rows and multiple columns. It can be seen as a collection of Series.

Pandas

Example (Series)

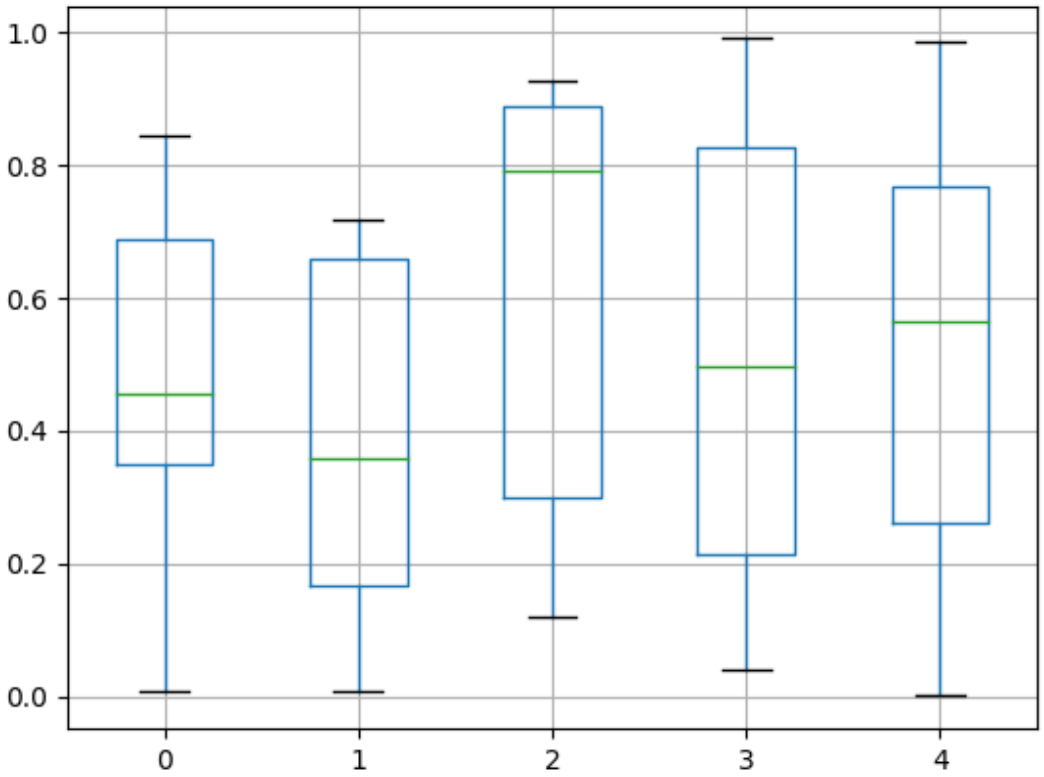
```
from pandas import Series

s1 = Series([4, 7, 9, -1])
s2 = Series([12, 3.2, "John"],
            index=["mark", "gpa", "name"])
```

Example (DataFrame)

```
from pandas import DataFrame
df = DataFrame({"key1": [1, 2, 3],
                "key2": [4, 5, 6]})
```

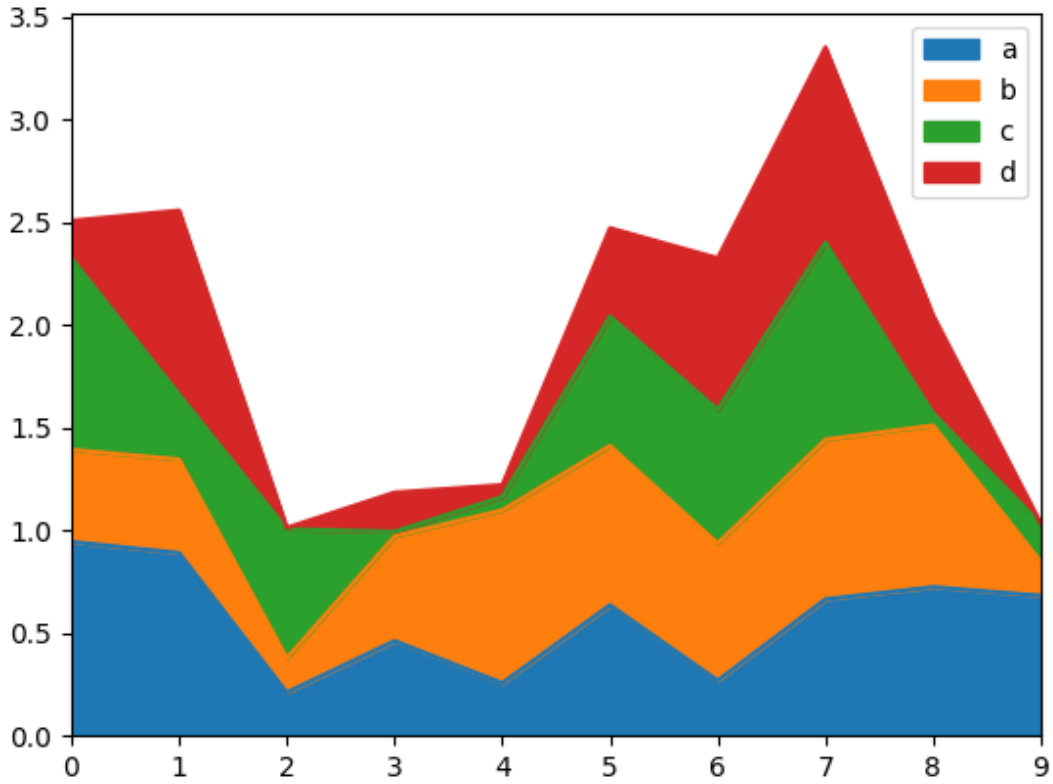
Matplotlib

Matplotlib is a plotting library used to visualize data and create graphics. Pandas directly uses matplotlib for representation.



Matplotlib

Matplotlib is a plotting library used to visualize data and create graphics. Pandas directly uses matplotlib for representation.



What is Machine Learning?

Machine Learning (ML)

ML is when a computational system can **automatically learn** and extract **knowledge from data** without being explicitly programmed. ML is at the intersection of statistics, artificial intelligence and computer science.
ML is now everywhere:

- automatic friend tagging in your Facebook photos
- music recommendation for Spotify/YouTube
- self-driven cars
- medical diagnosis in a hospital
- and many more...

ML can be **supervised** or **unsupervised**.

What is Machine Learning?

Brief history

At the beginning, many decision-making applications (like spam detection) were built with a lot of manually programmed if/else conditions. These methods have some drawbacks :

- it requires the **knowledge of an expert** to design the rules
- it is **very specific** to a task

With ML, you only need to feed a model with your data. The algorithm will find the rules by itself.

Supervised Learning

In supervised learning, you give the algorithm two kinds of data :

- **inputs** (an image, information about a person...)
- **outputs** (name of the object, the salary...)

The algorithm learns to find a way to **produce the output given the input**. The algorithm is then capable of producing the output of an input it has never seen before.

Supervised learning can be used to :

- recognize handwritten digits (input=image, output=digit)
- identify spams (input=mail, output=yes/no)
- predict a price (input=house infomation, output=price)

Supervised Learning

The inputs are in general real value vectors (sometimes it can be binary vectors). Each dimension of this vector is called a **feature**.

There are two main problems that can be solved with supervised learning :

- **classification** : the output is the class the item belongs to. It can be a binary classification problem (yes/no) or a multi-class classification problem (class 1, class 2, class 3...)
- **regression** : the output is a real value number

Unsupervised Learning

In unsupervised learning, there are no known output, so you only give the algorithm the inputs. For example, if you want to regroup similar clients according to their customer habits, you do not know in advance what are the groups.

The main problems that can be solved with unsupervised learning are :

- **clustering**
- **dimensionality reduction**

How to evaluate a ML algorithm?

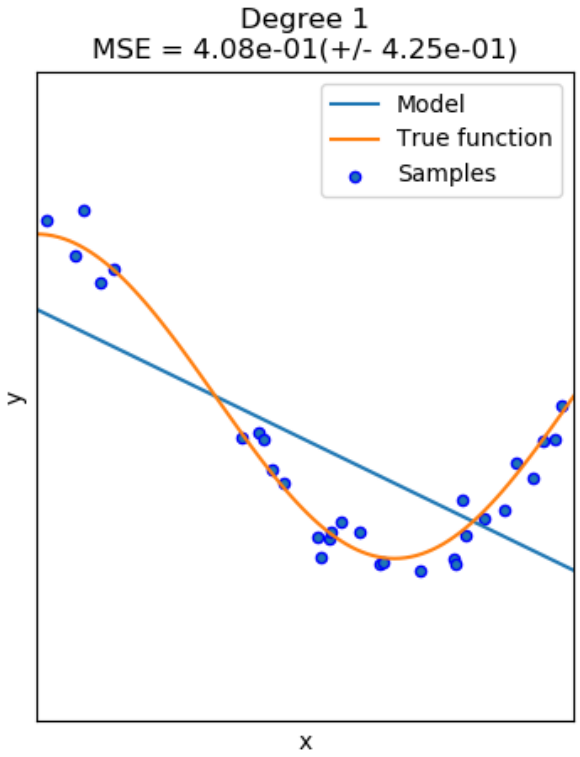
After training a ML algorithm with data, we need to evaluate its performance. One way would be to feed the algorithm with the same data, compute the output and compare it with the original output.

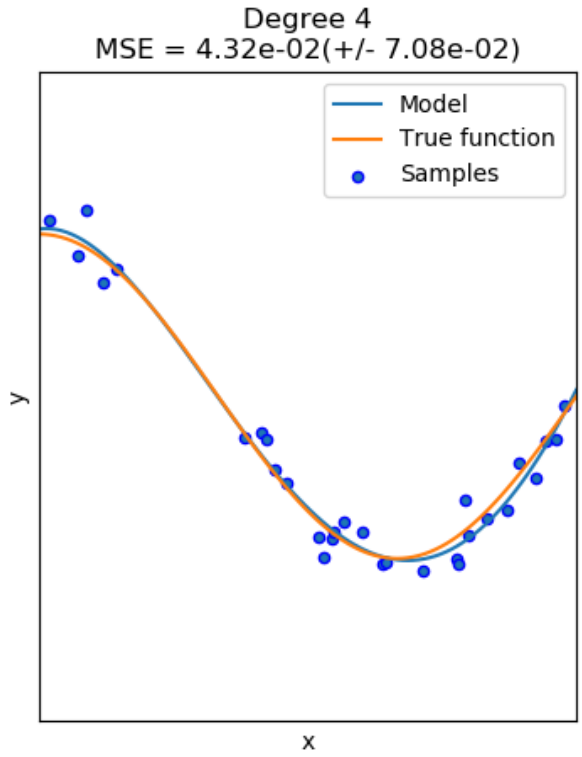
This method is bad because it does not tell us the capacity of the algorithm **to generalize** (is the algorithm able to perform well on unseen data ?). One way to solve this problem is to separate the data into 3 sets :

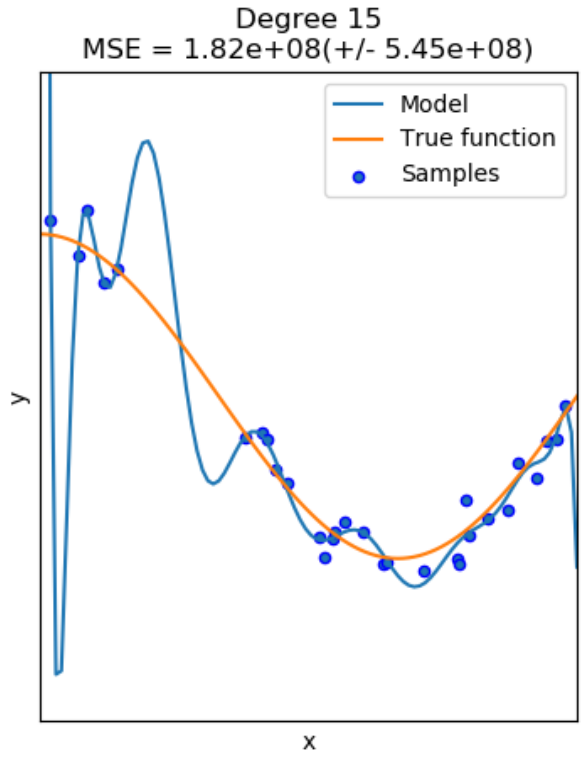
- training set : used to train the algorithm
- validation set : used to adjust the algorithm
- test set : used to measure the performance of the algorithm

Problems of ML

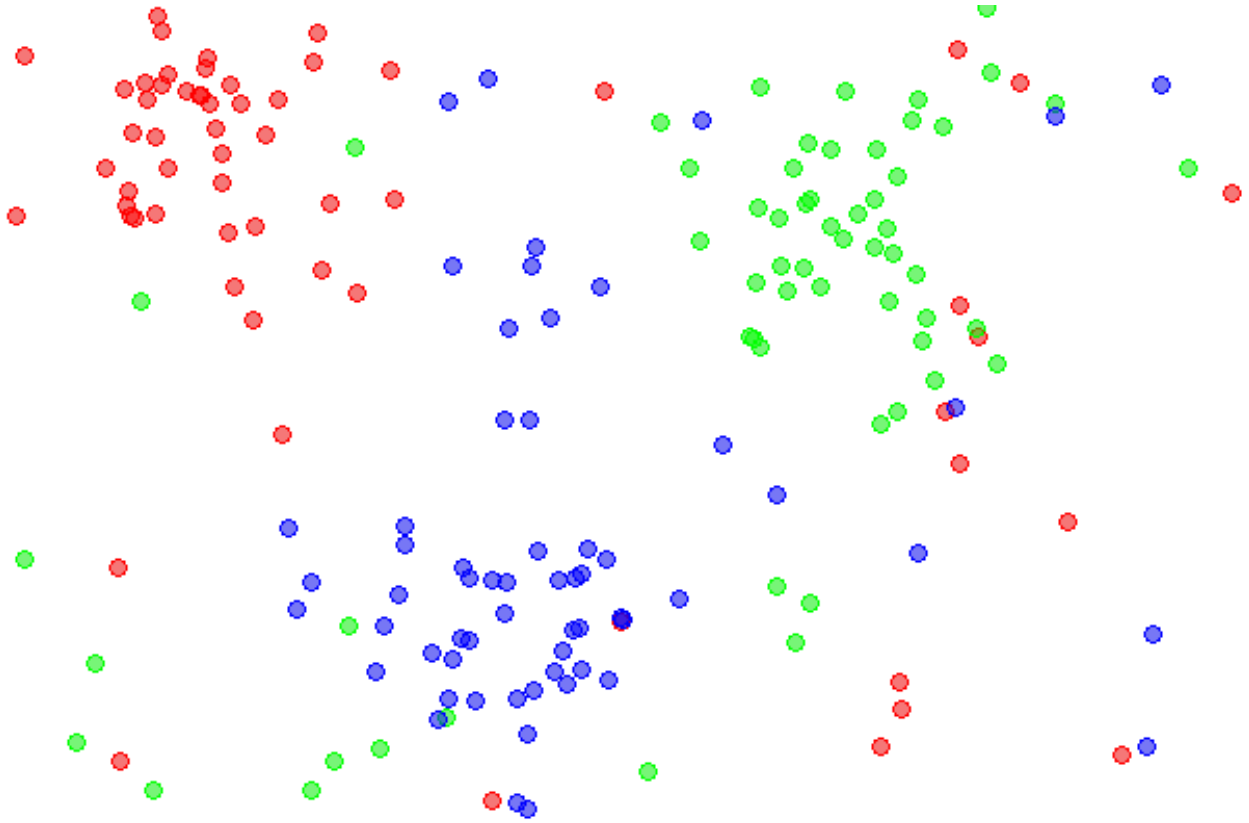
- **underfitting**: when the model is too simple. The training score and the test score are both bad (left)
- **overfitting**: is when the model is too complex (right). The training score is good but the test score is bad. The model is not able to generalize well.







k-Nearest Neighbors



Linear Models