Python for finance and optimization
Part II: Machine Learning with Scikit-Learn

The goal of this lecture is to discover Scikit-Learn (sklearn) and to test a few classification techniques.

Discovering the MNIST database

- Discover through Google what is the MNIST database.

- Get the data in Python by typing the following lines:

```
from sklearn.datasets import fetch_openml
mnist = fetch_openml('mnist_784', version=1)
```

- What is the type of mnist? What is the difference with a Python dictionary?

- Read thoroughly the description in `mnist.DESCR`.

- Get the images in a variable `X` and the corresponding digits in a variable `y`.

- Plot a few images to check your understanding of the data structure.

Recognizing the digit* 2 *(Logistic Regressions)

- Define the learning set and the test set following MNIST documentation.

- Try to use a Logistic Regression (`LogisticRegression` is in `sklearn.linear_model`) to predict whether an image represents a 2. What happens? Read well what is suggested.

- Try to increase the number of iterations (**max_iter**) and scale the data using a preprocessing tool: `StandardScaler` is in `sklearn.preprocessing`.

  Remark: be sure you understand the logic of the following methods: `fit()`, `transform()`, `fit_transform()`.

- Analyze the results using a confusion matrix and classical classification metrics: accurary, precision, recall and F1 score (find them by yourself).
- Have you really carried out a Logistic Regression as taught in an Econometrics 101 class? Test the different penalties.

Recognizing the digit* 2 *(Logistic Regressions and SVM with stochastic gradient descent)

- Try to use a stochastic gradient descent for fitting a Logistic Regression and a linear SVM (`SGDClassifier` is in `sklearn.linear_model`) and predict whether an image represents a 2. Read the documentation thoroughly before you code.

Recognizing digits

- Use a One vs. Rest methodology to classify images and recognize any digit (`OneVsRestClassifier` is in `sklearn.multiclass`).