SVM, Decision Trees and Ensemble methods

1 SVM with scikit-learn

We use the object `sklearn.svm.SVC` :
`from sklearn.svm import SVC`

1. Use the website :
   `http://scikit-learn.org/stable/modules/svm.html`
   and the dataset `Iris`. Implement a classifier which classifies class 1 against class 2 of the dataset `iris` using the two first variables and a linear kernel. Use half of the dataset for training and half of the dataset for validation. To import the `iris` dataset, type

   ```
   from sklearn import datasets
   iris = datasets.load_iris()
   X = iris.data
   y = iris.target
   X = X[y != 0, :2]
   y = y[y != 0]
   ```

2. Split into train set and test set

3. Compare the result of a linear SVM with a SVM based on polynomial kernel. One can use the function `score` to compare the different models

4. Evaluate the impact of the choice of the kernel and the regularisation parameter C.

5. Improve the robustness of the results with cross validation
   `https://scikit-learn.org/stable/modules/cross_validation.html`

2 Decision trees and ensemble methods

This section aims at using Ensemble Methods on real and synthetic data using the package `scikit-learn`.

2.1 Data importation and preparation

1. Import the dataset `digits` from the library `datasets`

2. Define the data and the labels

3. Split the dataset into train set and test set

2.2 Decision Trees

1. Import `scikitlearn` and the library `tree` of `sklearn` appropriate for decision tree
2. Fit a decision tree on the train set
3. Performance of this model on the test set?

2.3 Bagging, Random Forest and Boosting

1. Import `BaggingClassifier`, `RandomForestClassifier` and `AdaBoostClassifier` from the library `sklearn.ensemble`
2. Compare the performance of these methods on the dataset `digits`