Lab session : linear classification with scikit-learn (sklearn module)

1 Language identification

Detail how to build a classifier to identify the language of a text, with the following characteristics:

- the set of C possible languages is known in advance
- the classifier is learnt in a supervised way, using the perceptron algo
- the features to represent a text D are the number of occurrences of character bigrams, normalized by the total nb of occurrences of bigrams in D

In particular, what is the size of the vector representation of an input text?
What do you need to apply the perceptron?
What does the perceptron output?
How do you evaluate your classifier?

2 Linear regression with sklearn

This lab session is meant to introduce the use of the scikit-learn package, which implements various regression and classification algorithms. Note sklearn is much used for linear and log-linear models, but less used for deep learning. It also contains convenient methods to transform texts into vectors.

The classification and regression parts on the home page both point to the same general page on supervised learning:
https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

A lot of tutorials are available, in particular:

- http://scikit-learn.org/stable/tutorial/basic/tutorial.html
- https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

We will focus here on the « Ordinary least squares » method, which learns the parameters of a linear regressor using the least squares method.
We will use a non-NLP example:
https://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html#sphx-glr-auto-examples-linear-model-plot-ols-py which uses the LinearRegression class:
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

Import the **notebook** that is available at the bottom of the page and answer the following questions:

- recall what is linear regression
- what is the type and shape of diabetes_X, diabetes_y ?

- what is the semantics of each of the axis of diabetes_X, diabetes_y ?
- identify which are the methods for learning and prediction, and what do these take as input?
    - NB: these methods are always the same for all the sklearn models!
- Interpret what the line of code « diabetes_X = diabetes_X[:, np.newaxis, 2] » does.
- What is the evaluation metric?
- To go further: search for an explanation of r2_score

3 Modules to load and vectorize texts

See the specific notebook on sklearn modules to load texts
(sklearn_extract_features_from_text.ipynb)
cf. documentation of these modules:
https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_extraction.text
(in particular CountVectorizer and TfidfVectorizer, the latter being equivalent to CountVectorizer followed by TfidfTransformer)

Answer to the questions at the end of the notebook.

4 Linear classification with sklearn

We reuse the reuters dataset (small / medium / reuters for testing with various sizes), but we now use a « onedocperline » format.

CAUTION: a given document may be associated to several classes (comma-separated)
=> you will artificially duplicate these documents as many times as necessary in order to obtain one class per document (mono-label multiclass classification).

Write a program that :

- loads these documents into train and test matrices,
- learns on the training set using the perceptron
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html
- test it on a test set
- (study sklearn.metrics.accuracy_score to display the accuracy on test, and on train)

To go further:

- test other learning algos, in particular the SVM
    - http://scikit-learn.org/stable/supervised_learning.html#supervised-learning
    - (all the proposed classifiers have a `fit` and a `predict` method)
- implement a grid search for the hyperparameters of the SVM (see `sklearn.model_selection.GridSearchCV`)
    - `help(sklearn.model_selection.GridSearchCV)`
- see how to perform a cross-validation etc…