# Machine Learning 2014: Project 3 - Text Classification Report

anguyen@student.ethz.ch
spark@student.ethz.ch
frenaut@student.ethz.ch

December 19, 2014

## Experimental Protocol

Our approach can be divided in two fundamental steps:

- String Preprocessing
- Classification

Further details will be given in the next sections.

# 1 Tools

For the project we used python with the scikit-learn library.
In particular we used text mining tools for feature extraction (Tf-idf features) and multi-class SVM for classification.

# 2 Algorithm

## 2.1 String Preprocessing

In this phase, we create a dictionary of key words that will be used as features for the classification process.
To do so, we map *similar* words to the same key word.
The notion of string similarity is based on the levenshtein distance, that we slightly modified to better handle the problem. More in details, we assign a weight to the characters of the string so that the ones in the last (the rightmost) positions have less influence (e.g. 'g' and 'd' are more distant than 'yuiqosidnq' and 'yuiqosidnz').

### 2.2 Classification

The feature matrix is built using the key words weighted according to the tf-idf statistics.
For the learning process, we used Support Vector Classification implemented in the scikit-learn library with radial basis functions as kernel.

## 3 Features

As mentioned before, we used as features adequately preprocessed keywords weighted according to their frequency.

## 4 Parameters

The most important parameters to tune were a threshold value in the preprocessing phase (to decide if two words are similar enough to be considered the same) and the paramaters related to the classifier (in our case SVC with rbf kernel).
Since classification required a considerable amount of time, all parameters were manually tuned over a very restrictive yet reasonable range of values.

## 5 Lessons Learned

We tried to classify the data in two different phases: first by country code and then, given the country, by city code. This approach performed worse. A reason could be that the sample cannot be easily classified country (country code misclassification leads directly to a city code misclassification).
Moreover, we tried different classifiers. For example KNN, results were acceptable but not good enough. This might be related to the curse of the dimensionality: the final model uses some thousand of feature words.