Exercise 1 - Language Identification with sklearn

From Linear to Deep :-)

Deadlines

Deadline for Exercise 1 is **14.10.2019, 12:00 (MESZ)**.
Deadline for the peer review is **21.10.2019, 12:00 (MESZ).** You will find instructions for the peer review process at the end of this document.
Deadline for feedback to your peer reviewers is **25.10.2019 (MESZ).**

Learning goals

This exercise consists of two parts: the first part aims at deepening your understanding of linear models. The second part will already target a simple kind of multi-layered network; the multilayer perceptron (MLP). Don't worry if you don't know anything about MLPs when you read this. We will cover all you need to know to do the second part of this exercise next week in class and in the tutorial session (hopefully). By completing this exercise you should …

- … understand linear models and use them for multiclass classification tasks.
- … be able to implement different machine learning models, including MLPs, in scikit-learn.
- … understand the role of hyper-parameters, regularisation and handling class imbalance.
- … perform an error analysis of machine learning models.

Please keep in mind that you can always consult and use the exercise forum if you get stuck (note that we have a separate forum for the exercises).

Deliverables

Please hand in your code separately for part one and two. You will also have to write a lab report. Hand in the following files and name them exactly in the following fashion:

- ex01_lc.py
- ex01_mlp.py
- ex01_labreport.pdf

The .py files contain your well documented AND EXECUTABLE code. We assume that the data is in the same folder as the scripts, e.g.

- exercise01
    - ex01_lc.py
    - ex01_mlp.py
    - hydrated.json
    - Uniformly_sampled.tsv

Keep in mind that we are working with peer review, so your peers need to be able to run your code. If it does not work, you will not be able to obtain the maximum number of points.
Please submit the lab report in PDF format. The lab report should contain a detailed description of the approaches you have used to solve this exercise. Please also include results. We highlight places where we expect a statement about an issue in your lab report.
DO NOT submit the data files! Please zip your submission folder.

Data

For both parts of this exercise, you will work with the same data. The goal is to classify the language of Tweets. This is an extension of the problem described in Goldberg, chapter 2. However, we will work with more languages than just six and the text segments we need to classify are much shorter. Download the data from the material folder in the exercise section of OLAT. The folder contains the files "tweets.json", "labels-train+dev.tsv" and "labels-test.tsv". The first contains the tweets along with the Tweet-IDs, the latter two contain labels along with the Tweet-IDs. To prepare the data, you first need to match the language codes to the Tweets (there are more labels than tweets because some tweets have been deleted in the meantime). Then inspect the data and see how it is distributed.

- What are the properties of the data? Describe the most important of distributional properties of this data set.

Create a training set and a development set. Use a 90/10 split. Of course it is forbidden to peek into the test data ;-). You should only do this when you evaluate your model that performs best on the dev set.

Part 1 - Language identification with linear classification

Scikit-learn is a useful Python library for all kinds of machine learning tasks. In the following, you will train several models in sklearn to solve this task. The aim is to become acquainted with a few different classifiers, as well as with the basic functionality of sklearn.

1. Create a suitable pipeline in sklearn to preprocess the data. Think about extending the feature space. What other features could you use to determine the language? You're supposed to not only use bigrams for this task.
2. Train the following classifiers: SGDClassifier and Multinomial Naïve Bayes
3. In order to find the optimal hyperparameter settings for both classifiers, use sklearn's GridSearchCV. Especially with the SGDClassifier you are supposed to experiment with the following hyperparameters:
   a. Loss function
   b. Regularisation
   c. # of iterations
   d. Dealing with class imbalance

   Report the hyperparameter combination for your best performing model on the test set.

Also, compare the outputs of the best models for the two different classifiers. Which classifier scores higher on the test set? Do you have an idea, why this might be? What is the advantage of grid search cross validation? Use a confusion matrix to do your error analysis and summarise your answers in your report.

Part 2 - Your first Multilayer Perceptron (MLP)

Let's see if you can beat your best linear model you've trained with sklearn with an MLP.

1. Train an MLP classifier. You can also use GridSearchCV. Play around with different layer sizes, activation functions, solvers, iterations, momentum, and report your best hyperparameter combination. Important: use the same data splits as for Part 1.

If you need help on that, Raschka's [2015] chapter 2 provides an introduction to MLPs. The Google Machine Learning Crash Course also offers good material.

Peer Review Instructions

First: go to www.peergrade.io/join and join the class with the code **DY5RGZ**. Important: Register with the E-mail address you use for OLAT.

As soon as the deadline for handing in the exercise expires you will have time to review the submissions of your peers. Every student needs to do **3 reviews** in order to get the maximum number of points for this exercise.

Here some more rules:

- If you do not submit 3 reviews, the maximum number of points you can achieve is 0.5 (from a total of 1).
- Please use full sentences when giving feedback.
- Be critical, but fair!
- **Side note: all reviews are anonymous.**
- You must also give your reviewers feedback. The same criteria as above apply. Not giving feedback to your reviewers will also result in a deduction of points.
- Good reviewers (based on the feedback of their peers) will receive an extra point for the exercises, which means you can obtain at most 7 points for the exercises and presentation section. Ways to obtain points are thus the following:
  - 4 exercices = 4 points
  - 1 presentation or research paper dissection = 2 points
  - good review = 1 point