Learning challenge

Last updated: 2019-11-20

For this assignment you will participate in groups of 3 students in a learning challenge. You will need to submit your predictions, as well as a report describing your solutions and the code which you used to generate the predictions.

This assignment is worth 30% of your course grade.

The assignment grade will be based on the quality of your work as judged by the instructor based on your report and code. Additionally, you will get a bonus based on your ranking on the leaderboard of the shared task.

Specifically:

- if your rank first, you will receive bonus 2 points;
- if your score is no better than the provided baseline your will receive no bonus;
- for intermediate ranks the bonus points will be linearly interpolated.

The performance of the baseline solution is shown on Codalab (submitted by account `gchrupala`.)

Report

Your report should be **2 page maximum**, in **PDF format** and should include the following:

Page 1

- Description of your computational learning experiments, including:
  - feature engineering
  - learning algorithm(s) tried
  - parameter tuning
  - discussion of the performance of your solution

Page 2

- The name of the account under which you submitted your results to the competition on Codalab (see below)
- Detailed specification of the work done by group members
- References or appendices (if applicable)

Code

Your code should be a **plain Python script** (`.py`, not a notebook) which can be run to generate your predictions. You should not include any data files.

Codalab submission

You will need to submit your prediction file to the competition server. The competition is hosted on https://competitions.codalab.org. The team leader will need to get a codalab account (using a `uvt.nl` or `tiburguniversity.edu` email), and will be responsible for submitting your solution. Indicate the name of this account in your report. For further details about the format of the prediction file, see section **Submission to Codalab**.

IN SUMMARY: submission consists of the following items:

1. Report (`.pdf`, Canvas)
2. Code (`.py`, Canvas)
3. Prediction file (`.zip`, Codalab)

Group work

Your report needs to contain a detailed description of who did what, so make sure to keep track of this information.

Note: it is **not acceptable** to just say *All members worked together and contributed equally.*

If there are any problems with collaboration, such as serious disagreements, a group member not contributing, or a group dissolving, make sure inform the course coordinator as soon as possible via email.

Code reuse rules

Remember this assignment is group work. You are **not allowed** to collaborate or share code with students outside your group. **Submissions will be checked for plagiarism.**

If you are found breaking the above rules you will be reported to the Board of Examiners for fraud.

You **are allowed** to use:

- code examples provided by the instructor during the course, or as part of the competition
- open source libraries available for Python;

- code found on Github, Stackoverflow or similar websites, as long as it is credited in your script with a link to the source.

Dataset

Speech classification

In this challenge the task is to learn to recognize which of several English words is pronounced in an audio recording. This is a multiclass classification task.

Data files

The dataset is available for download on Canvas. It contains the following files:

- `wav.tgz`: a compressed directory with all the recordings (training and test data) in the form of `wav` files.
- `feat.npy`: an array with Mel-frequency cepstral coefficients extracted from each `wav` file. The features at index `i` in this array were extracted from the `wav` file at index `i` of the array in the file `path.npy`.
- `path.npy`: an array with the order of `wav` files in the `feat.npy` array.
- `train.csv`: this file contains two columns: `path` with the filename of the recording and `word` with word which was pronounced in the recording. This is the training portion of the data.
- `test.csv`: This is the testing portion of the data, and it has the same format as the file `train.csv` except that the column `word` is absent.

You can load the files `npy` using the function `numpy.load`, and the CSV files using the `csv` module or the `pandas.read_csv` function.

Evaluation metric

The evaluation metric for this task is classification accuracy (the proportion of correct predictions).

Method

There are three important restrictions on the method used:

- the method should be fully automatic, that is, by re-running your code it should be possible to re-create your prediction file;
- every software component used should be open-source and possible to install locally. This means that you cannot use propritary closed-source speech recognition software, or access a web service to carry out any data processing;

- the method should not use any external dataset which overlaps with the provided data. If you wish to make use of external data in your solution, ask the instructor via the course forum to confirm that this data is allowed.

Some hints:

- You can use the provided MFCC features for the spoken utterances, or you can extract your own features from the `wav` files.
- Use part of the provided training data as a validation set. Only submit to Codalab after validating your results on this your validation data.

Submission format

The submission format is the same as the file `test.csv` with the added column `word` with your predicted word.

The competition is hosted on Codalab at the following URL: http://bit.ly/2QByqLy

You can submit your results in the **Participate** link.

Over the course of the competition you can make 7 submissions.

Note that if you submission fails for some reason such as incorrect format, this is still counted as one of the 7 submissions.

The results from all the participating teams will be displayed in the **Results** tab.

The submission file should be a `.zip` file with a file named `result.csv` in it. (Make sure there are not additional subdirectories in the zip file.) Your file needs to use a valid CSV format. It is recommended to use the Python `csv` library or the `pandas.to_csv` function to create the file.