Data Science and Machine Learning Lab
Lab 07 - Classification

Politecnico di Torino

Intro

The main objective of this laboratory is to put into practice what you have learned on classification techniques. You will mainly work on audio signals. In particular, you will try to build a classification model that is able to identify which digit was uttered analyzing the content of short audio samples from different speakers.

Details on the competition

For what concerns this laboratory, you are encouraged to upload your results to the competition we launched on our platform (Data Science Lab Environment, or DSLE), even if the submission will not count on your final exam mark. Please refer to the course website for a complete guide on DSLE.

The personal key. To submit your solution, you must use the personal key you have received via email. If you do not have a key yet, please contacts lorenzo.vaiani@polito.it. Refer to Section 3 to read more about submitting your work.

Duration. The competition will last until the **Due date**. Then, DSLE will freeze the final leaderboard. However, once the games are over, you may want to test further solutions before the next laboratory, so we will leave the competition open for a few more days. Be aware that solutions uploaded within this window will not affect the leaderboard. All the dates are reported below for better clarity.

The DSLE platform is available at http://trinidad.polito.it:8888

Competition dates

Start date: November 16, 2025 at 10:00 AM (CET)
Due date: November 22, 2025 at 11:59 PM (CET)

Due date is a **strict deadline**.

1 Preliminary steps

1.1 Datasets

1.1.1 Free Spoken Digit Dataset

The dataset for this laboratory has been inspired by the Free Spoken Digit Dataset.

It is composed of 2,000 recordings by 4 speakers of numbers from 0 to 9 with English pronunciation. Thus, each digit has a total of 50 recordings per speaker. Each recording is a mono `wav` file. The sampling

rate is 8 kHz. The recordings are trimmed so that they have near minimal silence at the beginnings and ends.

The data has been distributed uniformly in two separate collections:

- Development (dev): a collection composed of 1500 recordings **with** the ground-truth labels. This collection of data has to be used during the development of the classification model. Each file in this portion of the dataset is a recording named with the following format `<Id>_<Label>.wav`.
- Evaluation (eval): a collection composed of 500 recordings **without** the labels. This collection of data has to be used to produce the submission file containing the labels predicted for each evaluation recording, exploiting the previously built model. Each file in this portion of the dataset is a recording named with the following format `<Id>.wav`.

So far, you should be used to work, developing your models, with training, validation and test sets. In this case, the Development data must be used to tune your hyper-parameters while you should consider the Evaluation portion as the actual test set.

1.1.2 Dataset tree hierarchy

The dataset archive is organized as follows:

- `dev`: the folder that contains the labeled recordings.
- `eval`: the folder that contains the unlabeled recordings. Use this data to produce the submission file containing the predicted labels.
- `sample_eval_sumbission.csv`: a sample submission file.

You can download the dataset at:

`https://github.com/dbdmg/data-science-lab/raw/master/datasets/free-spoken-digit.zip`

2 Exercises

In this laboratory you have a single classification task to carry out.

2.1 Free Spoken Digit classification

In this exercise you will build a complete data analytics pipeline to pre-process your audio signals and build a classification model able to distinguish between the classes available in the dataset. More specifically, you will load, analyze and prepare the Free Spoken Digit dataset to train and validate a classification model. Finally, you will be able to upload your classification results and participate to the lab competition.

1. Load the dataset from the root folder. Here the Python's os module comes to your help. You can use the `os.listdir` function to list files in a directory. Furthermore, you can use the `wavefile` module from SciPy to read a file in wav format. You can read more about it on the official documentation.

2. Focus now on the data preparation step. You should have noticed that `wavfile` gives you an array of floating point values, plus the sampling rate. Before continuing, take you your time to answer these questions:

   - what do these numbers represent?
   - were the audios recorded under the same conditions? (e.g. recording volume, noise, etc.)
   - do the arrays have an equal length? How different lengths could impact on your pre-processing solution? If it was needed, could you figure a way out to align them to the same length?

   Now, in order to train your model, you are required to design and build a vector representation. This mainly involves extracting several features out of your initial representation. Bear in mind that, since you are dealing with audio signals, you can work either on the time domain or the frequency one. In the former case, you might opt, for example, to split the signal into chunks and characterize them by means of statistical measures (e.g. mean, standard deviation). In the latter case, you can base your features on the frequencies contained in the signal. This involves reshaping the signal using a transformation function (e.g. Fourier transform) and work on its spectrum of frequencies (e.g. spectogram). Data preparation on frequencies can be hard to carry out. To know more about it, please refer to Camastra and Vinciarelli 2015 and Oppenheim and Schafer 2014, and Presti and Neri 1992.

   Identify a set of possible feature candidates and transform your data using them.

3. Once you have your vector representation, choose one classification algorithm of those you know. Then, perform the classic training-validation pipeline on the Development dataset to identify the best set of hyper-parameters for your model. As you can read in section 3.3, we will evaluate your results on the mean F1 score. Hence, it is a reasonable option trying to optimize it on the Development dataset [1].

   Info: the mean F1 score, also known as *macro average*, calculates the F1 for each label, and computes their unweighted mean. This does not take label imbalance into account.

4. Assign a classification label (i.e. the spoken digit) to each recording in the Evaluation dataset.

5. Submit your results to DSLE. Head to section 3 to know more about it.

6. Compile your final report and upload it to the "Portale della Didattica" as described in section 3.2.

[1] Actually, since your task does not present class imbalance, optimizing the classification accuracy would have been equally correct.

3 Submitting you work

For this laboratory, the submission consists of two files to be uploaded to two different web sites. The first file contains the classification results, the second file contains a report on the experiments you carried out. The following sections provide further details on that.

3.1 Submit your classification results

In order to get you results evaluated, you have to upload a result file on DSLE. The submission file has to be a `.csv` file formatted as follows:

```
Id,Predicted
0,0
1,0
2,0
3,1
4,1
...
```

As you can see, it must contain an header line and a row for each recording in the Evaluation collection. Each row must have two fields:

- the `Id` of the recording, as an integer number
- the `Predicted` label, as an integer number

3.2 Upload your report (optional)

For those interested, it is possible to submit a report describing the proposed solution for this laboratory.



Warning: Although for this laboratory no mandatory structure is required, at the end of the course you will have to write a report following a specific template **to pass the DSL exam**. So, we strongly suggest you to use the following LaTeX template:

- IEEE Conference Template - LaTeX

This template is similar to the one you will have for the **final exam** and it will help you get familiar with the LaTeX markup language producing more professional documents.

You can use the web based editor Overleaf to create and compile LaTeX documents with ease.

If you need a tutorial to start writing LaTeX documents you can have a look at Learn LaTeX in 30 minutes.

Later in the course, we will have a thorough lecture on the best practices for composing a technical report on your work.

The report submission should consist of a single PDF file. If you have developed your solution on a Jupyter notebook, you can export it as a PDF and use it for the submission. However the file should be sufficiently commented. Please respect the following requirements:

- state clearly which pre-processing step characterized your final solution;
- describe which classification algorithm you used;
- describe which validation strategy you adopted and which are the best hyper-parameters you found on the Development set.

You may upload the report to "Portale della Didattica", under the Homework section of the course. Please use the following as the description: `report_lab_7`.

If you would like to receive feedback about your work, please contact lorenzo.vaiani@polito.it before November 22, explicitly indicating that your report has already been submitted and that you are interested in receiving comments on the report.

Info: This is your first trial in producing a scientific report about your work. Try to be as focused as possible, provide a complete description of your pipeline, and highlight the relevant facts that drove your decisions. As previously stated, this is not a required submission and it will have no effect on your final grade. However, writing this document will help you understand how to organize your work into a report (a skill that you will put to use when writing the graded report for the exam).

3.3 Evaluation

Your classification results will be evaluated in terms of mean F1 score.

References

[1] Francesco Camastra and Alessandro Vinciarelli. *Machine learning for audio, image and video analysis: theory and applications*. Springer, 2015.

[2] Alan V Oppenheim and Ronald W Schafer. *Discrete-time signal processing*. Pearson Education, 2014.

[3] Letizia Lo Presti and Fabio Neri. *L'analisi dei segnali*. CLUT, 1992.