

Machine Learning Project

Fabrice Rossi

The goal of the machine learning project is to build a predictive model from a realistic data set. The construction of the model can be implemented either in Python (using Scikit-learn and other packages) or in R (with caret or with the tidymodel packages).

1 Data files

The data studied in this project are stored in several CSV files documented below. They concern 99985 French persons (from Metropolitan France). The goal the project is to build a predictive model for the target variable. This variable represents an undisclosed characteristic related to the person's socio-economical status taken in a broad sense.

The data set has been split into a learning set with 49993 persons and a test set with 49992 persons. Each set is described by four csv files:

- `learn_dataset.csv` and `test_dataset.csv` contain the main description of the persons in the data set, with the `target` variable only in `learn_dataset.csv`;
- `learn_datatset_job.csv` and `test_datatset_job.csv` describe the jobs of the persons with an employee status;
- `learn_dataset_Emp_contract.csv` and `test_dataset_Emp_contract.csv` give the type of job of each person with a job (including non employee);
- `learn_dataset_Club.csv` and `test_dataset_Club.csv` contain information about persons who are registered to sport clubs.

Details about the file are provided below.

1.1 Persons

Persons are described by the following variables:

- `Id`: primary key (unique identifier);
- `AGE_2019`: age;
- `sex`: sex;
- `household`: family type;
- `Highest_degree`: highest diploma;
- `ACTIVITY_TYPE`: activity type;
- `is_student`: true if the person is a student;
- `OCCUPATION_42`: socio-professional category (PCS 2003 norm, see below);
- `insee_code`: INSEE code of the city of residence.

1.2 Job

The current jobs of persons with an employee status are described with the following variables:

- **Id**: foreign key to the person table;
- **Eco_sect**: economic sector of the job;
- **Terms_of_emp**: work contract type;
- **job_category**: type of job (regular, internship, etc.);
- **WORK_CONDITION**: job terms (full-time, part-time, etc.);
- **Working_hours**: total annual working hours (this variable has missing values);
- **employer_category**: type of employers;
- **EMPLOYEE_COUNT**: size of the company;
- **Job_dep**: department in which the job is located;
- **work_description**: description of the job according to the PCS-ESE 2017 norm (see below);
- **EMOLUMENT**: annual salary of the person.

1.3 Job type

The job type of all persons with a job is given by the **Emp_contract** variable in the associated csv file. The link with the person file is provided by the foreign key **Id**. Notice that persons with a job do not necessarily have an employee status.

1.4 Sport

When a person is registered in a sport club, the corresponding club is described by a **Club** variable in the associated csv file. The link with the person file is provided by the foreign key **Id**.

1.5 Categorical variables and geography

Most variables are categorical. The possible values are listed and documented in CSV files named after the variables (e.g. **code_Highest_degree.csv** for the **Highest_degree** variable). Notice that those files have been produced by INSEE and are written in French. The PCS-ESE 2017 INSEE norm is described by the following files:

- **code_work_description.csv** contains the association between codes and profession;
- **code_work_description_map.csv** contains a mapping between the complete codes (N3) used in the data set and two coarser representations (N1 and N2);
- **code_work_description_n2.csv** contains the association between codes and profession groups at the N2 level;
- **code_work_description_n1.csv** contains the association between codes and profession groups at the N1 level.

The PCS 2003 is a complementary norm which adds modalities to the N2 level of the PCS-ESE 2017. Codes are given in `code_OCCUPATION_42.csv`

Geographical and administrative information about metropolitan French cities is contained in several files:

- `city_adm.csv` contains administrative information:
 - `Nom de la commune`: city name
 - `insee_code`: INSEE code of the city;
 - `dep`: code of the department of the city;
 - `town_type`: city type (modalities are administrative city category);
- `city_loc.csv` contains geographical information, the GPS coordinates of the cities expressed in the WSG 84 system¹ as well as in the Lambert-93 projection². The Lambert-93 coordinates can be used to compute distances (in meters) between cities with a reasonable precision in metropolitan France. Attributes:
 - `insee_code`: INSEE code of the city;
 - `Lat`: latitude;
 - `LONG`: longitude;
 - `X`: X Lambert coordinate;
 - `Y`: Y Lambert coordinate;
- `city_pop.csv` contains population information:
 - `insee_code`: INSEE code of the city;
 - `Inhabitants`: population of the city.
- `deparments.csv` contains departments information:
 - `Nom du département`: department name;
 - `dep`: code of the department;
 - `REG`: code of the region to which the department belongs.
- `regions.csv` contains region information (from 2018):
 - `Nom de la région`: region name;
 - `REG`: code of the region.

Sport clubs are affiliated to sport federations which are themselves sorted into several broad categories, as document in `code_Club.csv`

¹https://en.wikipedia.org/wiki/World_Geodetic_System

²https://en.wikipedia.org/wiki/Lambert_conformal_conic_projection

2 Expected results

The goal of the project is to build a predictive model for the **target** variable given the other variables. More precisely, you are expected to

- build a predictive model using the learning data;
- estimate the future performances of the model on new data;
- provide the prediction of your model on the test set.

The following rules must be obeyed:

- at least two different machine learning algorithms must be compared. It is recommended to include a linear model and a random forest among them;
- the use of a resampling technique to select the best model is mandatory (this can be for instance v-fold cross-validation for general models, and leave-one-out or out-of-bag estimates for specific ones);
- the main meta-parameters of the machine learning algorithms must be selected via a resampling technique;
- observations with missing data cannot be removed from the test set.

In addition, it is recommended:

- to debug your program on a sub-sample of the learning set, given its relatively large size;
- to complement the core data set (i.e. the general description of the persons) by information contained in other files (jobs, sports, geography, etc.);
- to use category simplification/grouping if needed for categorical variables. In particular, most predictive models will have difficulties with the **work_description** variable if used directly. It is acceptable to use of external data to simplify the categories;
- external data can be used to complement the features.

3 Project submission

The results of the project must be submitted as a single zip file containing:

- a report on the predictive analysis (exclusively in pdf format, other format will be discarded): this report should be short and very precise. It should outline the methodology used to construct the chosen model. Do not include anything that could be considered as lecture notes (for instance, I happen to have a precise idea of what is a v-fold cross-validation and I do not need to be reminded of the definition). More precisely the report should answer to the following questions:
 - What external data were added beyond the one provided as part of the project?
 - What data were used to build the models? (this can depend on the model)
 - What pre-processing were conducted? (this can also depend on the model)
 - What models were tested and what was the grid of meta-parameters used to tune each of them (advanced meta-parameters optimisation techniques can be used instead of a grid search)?

- What **resampling method** was used to select the meta-parameters and the final model?

The report must contain an **estimation of the expected** quality of the predictions on new data reported in an adapted form (for instance the **coefficient of determination**, a.k.a. the **r-squared**, and the **mean-squared error for regression problems**). Additional assessment of the model and of its predictions will be appreciated (**this can include an analysis of important variables, graphical representation of the predictions, etc.**);

- a file named **predictions.csv** with the predicted values of **target** on the test set, using the following convention
 - the file must contain two columns in this order
 1. **Id**: the foreign key that links a prediction to the person in the test set;
 2. **target**: the prediction itself;
 - the file must be in CSV format, with commas as the separation character;
 - decimal numbers must use the standard US representation (e.g. 2.5).
- the full code used to perform the analysis. I should be able to run the code without any modification by simply unzipping the file and adding the data you were provided in the same directory as the code (or in a subdirectory specified in the report). You may include the original data in the submission and you must include any external data.

Notice that no manual editing of the data files via e.g. excel is permitted. In particular, if data files must be combined, this has to be done with R/Python. **It is strongly recommended to produce the report in a reproducible way, using rmarkdown, quarto or jupyterbook.** Notice that the pdf/html output produced by jupyter-notebook are horrendous and do not achieve the minimal presentation quality requested for the project.

In addition, the quality of the predictions will play an important part in the marking of the project. This quality will be automatically computed from the **predictions.csv** file. If the file is not named correctly, if it does not follow the format specified above or if some predictions are missing, this part of the project will be considered as failed.