Machine Learning Project

Fabrice Rossi

The objective of this machine learning project is to develop a predictive model based on a realistic dataset. Model construction can be implemented using either Python (with libraries such as Scikit-learn) or R (with packages mlr3 or tidymodels).

1 Data files

This project involves building a predictive model to estimate an undisclosed characteristic related to the socio-economic status of 100090 individuals in Metropolitan France. The `target` variable for this prediction is only available in the learning set.

1.1 Data structure

- The data is distributed across several CSV files, divided into learning and test sets (50046 and 50044 individuals, respectively).
- **Main Datasets:** `learn_dataset.csv` and `test_dataset.csv` contain primary individual information, including the `target` variable in the learning set.
- **Job-Related Datasets:**
  - the type of job of all working individuals, including non employee such as independent contractors, is given in `learn_dataset_EMP_TYPE.csv` and `test_dataset_EMP_TYPE.csv`.
  - `learn_datataset_job.csv` and `test_datataset_job.csv` describe in details the jobs of individuals with employee status.
- **Retirement-Related Datasets:**
  - `learn_dataset_retired_former.csv` and `test_dataset_retired_former.csv` provide information on the last working conditions of retired individuals (including non employee).
  - `learn_dataset_retired_jobs.csv` and `test_dataset_retired_jobs.csv` describe in details the last jobs of retired individuals who were previously employed.
  - `learn_dataset_retired_pension.csv` and `test_dataset_retired_pension.csv` indicate the pension amount for each retired individual.
- **Club Membership Dataset:** information about individuals registered with sports clubs are provided in `learn_dataset_Club.csv` and `test_dataset_Club.csv`.

Details on the content of the files, particularly regarding the nature of the variables, are given below.

1.2 Data Handling Considerations

This comprehensive dataset has a relatively complex structure. Some of the subsets contain missing values; for instance, the number of working hours in a job can be unknown. In addition, some information may be entirely missing for a given person. For instance, we may not know the last job of a retired person. These patterns of missingness are completely different: while imputation techniques may work for filling in a single missing value in the description of a person, imputing a large number of variables may lead to incorrect models. For instance when a person is not retired the most likely reasons for them to be missing from the job dataset are that they are not employee. They could be unemployed, they could also be independent contractors, etc. Considering their job data as "missing" would be completely wrong in this situation.

In addition, the datasets may exhibit minor inconsistencies, such as a job position being presented as non-permanent in the job dataset but as permanent in the type of job dataset. It is recommended to implement some minimal sanity checks to verify the consistency of the data. If inconsistency corrections are needed, the main dataset should be considered more reliable than the job type dataset, which is, in turn, more reliable than the full job description dataset.

Extreme care must be exercises when loading the data. In particular, some software may consider the INSEE city code (`insee_code`) as an integer and drop the leading 0 of some codes (e.g. turn 01001 into 1001). This may lead to an incorrect model.

1.3 Persons

Persons are described by the following variables:

- `primary_key`: primary key (unique identifier);
- `age_2020`: age;
- `sex`: sex;
- `Family_type`: family type;
- `Qualification`: 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.

1.4 Job

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

- `primary_key`: foreign key to the person table;
- `eco_sect`: economic sector of the job;
- `Type_of_contract`: work contract type;
- `JOB_CATEGORY`: type of job (regular, intersnship, etc.);

- Work_condition: job terms (full-time, part-time, etc.);
- working_hours: total annual working hours (this variable has missing values);
- EMPLOYER_TYPE: type of employers;
- Employee_count: size of the company;
- job_dep: department in which the job is located;
- job_desc: description of the job according to the PCS-ESE 2017 norm (see below);
- Remuneration: annual salary of the person.

The last job of a retired person is described with almost the same variable:

- the Remuneration is not given;
- an additional variable Last_dep specifies the department where the retired person lived when they were employed on this last job.

1.5 Job type

The job type of all persons with a job is given by the EMP_TYPE variable in the associated csv file. The link with the person file is provided by the foreign key primary_key. Notice that persons with a job do not necessarily have an employee status. It is therefore normal to find persons with a job type but with no description of this job.

1.6 Former job type

The former job type of retired persons is described by the following variables in the files learn_dataset_retired_former.csv and test_dataset_retired_former.csv:

- primary_key: foreign key to the person table;
- LAST_EMP_TYPE: the type of job (as for working persons);
- LAST_OCCUPATION_42: socio-professional category (PCS 2003 norm, see below);
- retirement_age: the age at which the person retired.

1.7 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 primary_key. This dataset is expected to be exhaustive: if a person is not listed in the file, they are not member of a sport club.

1.8 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_Qualification.csv` for the `Qualification` 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_job_desc.csv` contains the association between codes and profession;
- `code_job_desc_map.csv` contains a mapping between the complete codes (N3) used in the data set and two coarser representations (N1 and N2);
- `code_job_desc_n2.csv` contains the association between codes and profession groups a the N2 level;
- `code_job_desc_n1.csv` contains the association between codes and profession groups a 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;
  - `city_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 system[1] as well as in the Lambert-93 projection[2]. 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;
  - `Community_size`: population of the city.
- `deparments.csv` contains departments information:

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

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

  - `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`

These documentation files can be useful to understand better the structure of the data. They can also be used for feature engineering, for instance to reduce the number of categories for variables with a large number of them.

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 obeyed:

- you can use AI agents to help you write the code of the project, including for visualisation purposes (for instance to avoid wasting your time trying to understand matplotlib "documentation"). You cannot use AI to write any part of the report. Do not try to bling the report with AI generated images;
- 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). As the dataset is relatively large, a simple split sample strategy could be the only acceptable solution on small computers;
- 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, these recommendations may be helpful:

- you should debug your program on a sub-sample of the learning set, given its relatively large size;
- you should complement the core data set (i.e. the general description of the persons) by information contained in other files (jobs, sports, geography, etc.). In fact, it is very unlikely to get good predictive performances by using only the core dataset;

- random forests should be used as the reference algorithm. Testing the K-nearest neighbour algorithm is probably a waste of computational resources. Linear models may provide acceptable performances, but you have to be careful considering the nature of the data (potential high correlation between variables, very large number of modalities, etc.). There is no particular reason to test decision trees, as they are a particular cases of boosting. Boosting should provide the best performances, once properly tuned;
- you should use category simplification/grouping for categorical variables with a large number of modalities (large being more than 100 for instance). For instance some predictive models will have difficulties with the `job_desc` variable if used directly. It is acceptable to use of external data to simplify the categories;
- you may use external data to complement the features (by external, I mean data that are not part of the provided bundle);
- notice that a "model" is not simply, say, a random forest. Firstly, the model should include all preprocessing steps and secondly, owing to the complex structure of the data, the final model may well be a combination of several models applied conditionally. For instance, you may have one model for persons with a precise job description and another model for persons without such additional data.

3 Project submission

The results of the project must be submitted on Moodle 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 am already familiar with the definition of v-fold cross-validation and do not need to be reminded of it). More specifically, 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 was 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 include an explicit 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. R-squared, and the mean squared error for regression problems). I expect to read in the report something like: "The expected performances of our final model on new data are xxx".

  The report must include a full assessment of the predictions of the model using adapted tables and visualisations. For instance, in a classification setting, I expect to see a confusion matrix but also some score based assessment (like the ROC curve if this is possible).

Additional assessments include variable importance analysis, extremely wrong predictions exploration, 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. `primary_key`: 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 outputs 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.