Data Management Project

Fabrice Rossi

The goal of the data management project is to conduct some of the main steps of a data oriented project, using in particular the Pandas Python library (including its graphical methods), as well as additional visualization libraries such as Seaborn. The main expected outcomes of the project are a Python script containing all the steps of the analysis and a report discussing the results. Both documents can be merged in a single Python Jupyter Notebook (this is the recommended solution).

1 Data files

The data studied in this project are stored in several CSV files documented below. The main file `dataset.csv` contains a description of persons enrolled in a marketing campaign together with the results of the campaign.

Persons are described by the following variables:

- `key`: unique key;
- `firstname`: first name;
- `l_name`: last name;
- `current_age`: age during the marketing campaign;
- `sex`: sex;
- `household`: family type;
- `Highest_degree`: highest diploma;
- `Studying`: student status (true or false);
- `ACT`: activity type (such as retired, unemployed, etc.);
- `occupation_8`: low resolution job type (8 classes);
- `Occupation_24`: high resolution job type (24 classes);
- `Insee`: INSEE code of the city of residence;
- `OUTCOME`: outcome of the marketing campaign.

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.

Additional information available only for a subset of the persons are available in separated files. Each file contains a `key` variable that enables to identify the person to whom the information relates.

- `dataset_contract.csv`: contains the employment type when that makes sense (such as permanent position, temporary position, etc.) in the `contract` variable.
- `dataset_CLUB.csv`: for persons who are members of a sports association, the `CLUB` variable contains the code of the association (codes are documented in the `code_CLUB.csv` file).

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`: 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 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`: INSEE code of the city;
  - `Lat`: latitude;
  - `Long`: longitude;
  - `X`: X Lambert coordinate;
  - `Y`: Y Lambert coordinate;
- `city_pop.csv` contains population information:
  - `Insee`: 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.

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

2 Expected results

2.1 Data loading and representation

The first step of the analysis consists in loading the data with Python and in performing all the needed operations to have a proper computer representation of them. In particular, categorical data are expected to be represented by Pandas categories[3]. Data frames describing the geographical and administrative structure of France may also benefit from using the indexing facilities of Pandas.

The report part of the project should document this phase from a data management point of view. Subjects to discuss include, if applicable, missing data, data encoding choice, data nature, etc.

As a bonus content, a conceptual model of the full data set (including data files) may be constructed, using the Entity-Relationship model[4]. Numerous tools can be used to draw ER diagrams, for instance the online ERDPlus[5] one.

2.2 Person level analysis

The data must be analyzed in order to discover possible relationships between the result of the marketing campaign and the characteristics of the persons in the data set, including information obtained from related data files (for instance information about their city of residence, among others). Two complementary points of view should be used:

- descriptive point of view: identify characteristics of the persons in the "success" set that differentiate them from the "failure" set. For instance, the "success" set might contain more women than men. Characteristics that are common to both groups should not be described;
- predictive point of view: identify characteristics of persons that can be used to predict the outcome of the marketing campaign. For instance being unemployed might induce a high probability of "success". The predictive quality is relative: if the expected success rate is 10% then observing that students have a success rate of 15% is very interesting, even if the final rate is still rather small.

Notice that the two analyses will of course agree but that the quality of the description may vary strongly. For instance, the "success" set may have more women than men because of a combination of a slightly higher probability of "success" for women, a larger number of women in the full data set and some dependencies between other variables. In the "success" set the gender rates can be quite different while the "success" probabilities conditionally on the gender can be less different.

The report must include a discussion about those two points of view supported by data manipulations and visualization implemented in Python. For instance, one might compute the rates of women in the two sub-populations with an appropriate split/apply/combine strategy. If the rates are sufficiently different, the resulting table should be included in the report with a discussion, and the corresponding code should be included in the python processing script (with an appropriate comment). Both can be combined if a notebook is used.

2.3 Grouped analysis

The geographical nature of the data enables one to perform group analyses at different aggregated level. For instance, one might use France administrative structure to perform departmental

[3] See https://pandas.pydata.org/pandas-docs/stable/user_guide/categorical.html
[4] https://en.wikipedia.org/wiki/Entity-relationship_model
[5] https://erdplus.com/

level analyses and regional level analyses. Distance based analyses can also be done using the Lambert-93 coordinates.

The report must include at least one type of grouped analyses. It is recommended to start with department level analyses. A possible approach consists in summarizing the population of each department with well chosen aggregates (such as the median age, among many other possibilities), keeping separated the "success" and "failure" population. Departments can then be compared, both in terms of population structure and in terms of local "success" explanations. For instance, in a given department the age might be a good predictor of "success" while in another this might be the gender.

As in the person level analyses, discussions must be based on statistics and figures computed from the data using python scripts (which must be in turn included in the global script/notebook).

3 Project submission

The results of the project must be submitted as a single zip file containing a report (exclusively in pdf format) and the full Python code used to produce the analysis summarized in the report. If the project is implemented with a notebook, the submission must contain: the notebook itself, the python code extracted from the notebook and a pdf rendering. All those files can be obtained using the file menu from Jupyter Notebook[6].

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 python. Submissions must be uploaded on Mycourse at the latest on Monday the 18th of January.

[6]It might by necessary to install pyppeteer for pdf export.