Predihood: an open-source tool for predicting neighbourhoods' information

Nelly Barret[1], **Fabien Duchateau**[1], **and Franck Favetta**[1]

1 LIRIS UMR5205, Université Claude Bernard Lyon 1, Lyon, France

DOI: 10.21105/joss.02805

Software

- Review
- Repository
- Archive

Editor: Gabriela Alessio Robles

Reviewers:

- @jdalzatec
- @omshinde
- @nuest
- @martinfleis

Submitted: 21 September 2020
Published: 09 May 2021

Introduction

Neighbourhoods are a widespread concept in studies from diverse domains such as health, social sciences, or biology. For instance, Japanese researchers investigated the relationships between social factors and health by taking into account behavioral risks and housing and neighbourhood environments (Takada et al., 2014). In a British study, authors describe how living areas impact physical activities, from which they determine a walkability index at the neighbourhood level for improving future urban planning (Frank et al., 2010). Another survey describes the luxury effect, i.e., the impact of wealthy neighbourhoods on the surrounding biodiversity (Leong et al., 2018). Several works focus on qualifying neighbourhoods using social networks. For instance, the Livehoods project defines and computes neighbourhoods dynamics (Cranshaw et al., 2012), while the Hoodsquare project detects similar areas based on Foursquare check-ins (Zhang et al., 2013). Crowd-based systems are interesting but may sometimes be biased, and they require technical skills to extract relevant data. DataFrance is an interface that integrates data from several sources, such as indicators provided by the National Institute of Statistics (INSEE), geographical information from the National Geographic Institute (IGN), and surveys from newspapers for prices (L'Express). DataFrance enables the visualization of hundreds of indicators but makes it difficult to judge the main characteristics of a neighbourhood and is limited to France. Despite all of these existing applications, there is no simple tool to visualize and predict insights about neighbourhoods.

The Predihood tool fills this gap by defining neighbourhoods, their characteristics, and variables to be predicted. It also includes a cartographic interface for searching and displaying information about neighbourhoods. Domain experts can provide a few examples of annotated neighbourhoods, and Predihood provides a configuration interface for using popular machine-learning algorithms in order to predict variables for the remaining neighbourhoods. One of the most recent applications of Predihood was measuring the impact and influence of a neighbourhood's environment on the decision-making process when people move to another city (Barret et al., 2020).

Predihood mainly targets non-programmers users (e.g., researchers in social sciences or history) due to its simplicity for running and configuring predictive algorithms. It can be extended to other application domains: measuring the pollution degree in neighbourhoods, determining whether a certain neighbourhood is suitable as a stopover for migratory birds, predicting neighbourhood evolution based on historical data, etc. This paper describes the main features of Predihood and how to extend them.

Methodology

Predihood provides the following functionalities:

- adding new neighbourhoods and indicators to describe them;
- predicting variables of a neighbourhood by configuring and using predefined algorithms;
- adding new predictive algorithms.

To facilitate understanding, we describe and illustrate these functionalities based on a simple example that aims to evaluate which neighbourhood is preferable for migratory birds to make a temporary stop. We only include three indicators per neighbourhood: the percent of greens, the percent of buildings and the degree of human pressure. A single variable `migration zone` qualifies a neighbourhood from *favorable* to *unfavorable*.

Adding new datasets

As Predihood is a general-purpose application, it enables contributors to add their own datasets. The key concept of a dataset is the neighbourhood, which is represented as a GeoJSON object including:

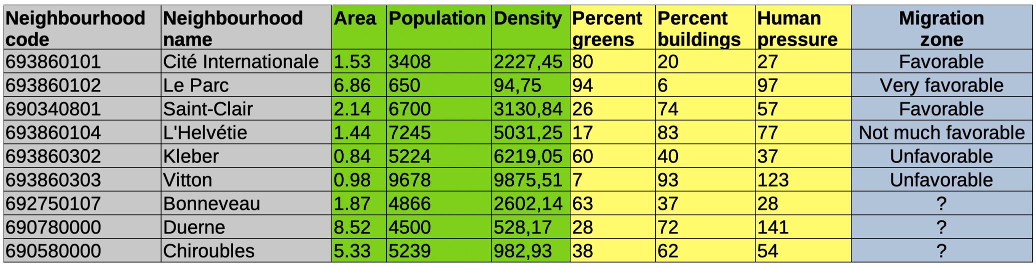
- a geometry (multi-polygons), which describes the shape of the neighbourhood. This crucial data is not only useful for cartographic visualization but also enable automatic calculations such as area;
- properties, which are divided into two categories. Descriptive information (e.g., name, city postcode) mainly aims to improve display while a set of quantitative indicators is used to predict the values of the variable.

Besides, some neighbourhoods have to be manually annotated, a task typically performed by domain experts. To add a new dataset, it is necessary to store them as GeoJSON and make them accessible by Predihood, for instance, in a document-oriented database.

To build a dataset for the *bird migration* example, it is necessary to collect and integrate data sources about neighbourhoods located in the studied geographic area. Values for the three indicators should be provided, and a value to the `migration zone` variable should be assigned to a few neighbourhoods.

Predicting

Machine learning algorithms require data preparation by grouping relevant properties and variables. We illustrate this step on the *bird migration* dataset, as shown in Figure 1. Predihood produces a table composed of the identifier and the name of the neighbourhood (grey columns), its indicators (yellow columns) that could be normalized by factors such as density of population (green columns), and optionally the assessment of researchers for the `migration zone` variable (blue column). The objective of Predihood is to automatically fill question marks for neighbourhoods that are not yet assessed.

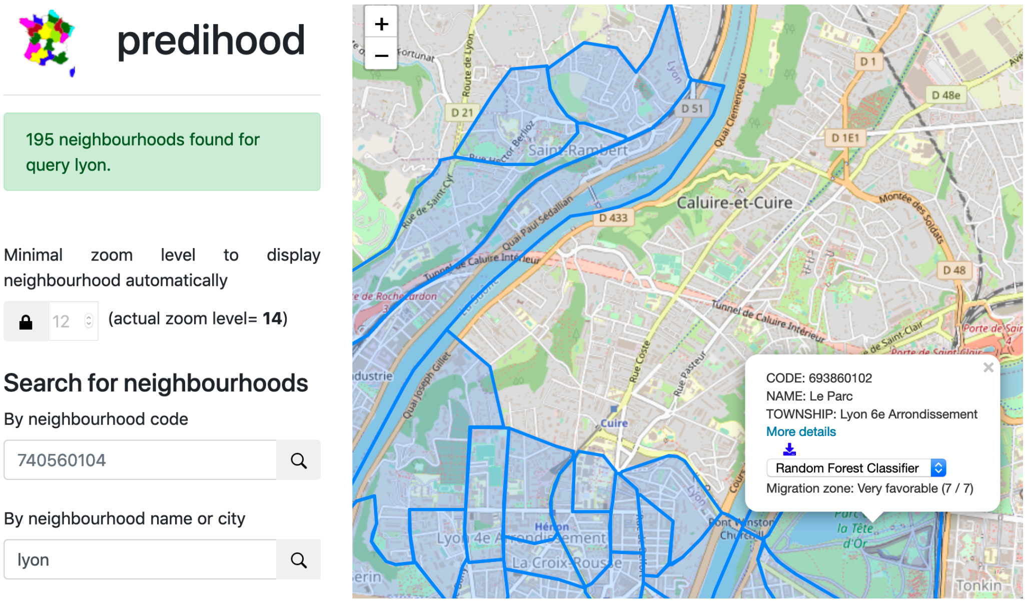


Neighbourhood code	Neighbourhood name	Area	Population	Density	Percent greens	Percent buildings	Human pressure	Migration zone
693860101	Cité Internationale	1.53	3408	2227,45	80	20	27	Favorable
693860102	Le Parc	6.86	650	94,75	94	6	97	Very favorable
690340801	Saint-Clair	2.14	6700	3130,84	26	74	57	Favorable
693860104	L'Helvétie	1.44	7245	5031,25	17	83	77	Not much favorable
693860302	Kleber	0.84	5224	6219,05	60	40	37	Unfavorable
693860303	Vitton	0.98	9678	9875,51	7	93	123	Unfavorable
692750107	Bonneveau	1.87	4866	2602,14	63	37	28	?
690780000	Duerne	8.52	4500	528,17	28	72	141	?
690580000	Chiroubles	5.33	5239	982,93	38	62	54	?

Figure 1: A subset of the *bird-migration* dataset.

To perform prediction, a selection process first selects subsets of relevant indicators, since too many indicators may degrade performance. The tool automatically reduces the number of indicators, e.g., by removing indicators with a unique value or those highly correlated using Spearman coefficient. Then, Predihood generates 7 lists of indicators, containing from 10 to 100 indicators. The current version of Predihood includes 8 predictive algorithms from scikit-learn (e.g., Random Forest, KNeighbours) (Pedregosa et al., 2011), depending on the algorithm, small or large lists of indicators may be more effective.

Predihood provides a cartographic web interface based on Leaflet and Open Street Map, as shown in Figure 2. For this example we use the search query "*lyon*" (left panel) and all neighbourhoods containing this query in their name or their city names are shown in blue on the map. We select the neighbourhood *Le parc* and run the Random Forest classifier: migration is considered *very favorable* in this area (for the seven lists of indicators). Indeed, this park seems relevant for bird migration as it has nice green areas for birds and it is a healthy environment for them.



predihood

195 neighbourhoods found for query lyon.

Minimal zoom level to display neighbourhood automatically

12 (actual zoom level= **14**)

Search for neighbourhoods

By neighbourhood code

740560104

By neighbourhood name or city

lyon

CODE: 693860102
NAME: Le Parc
TOWNSHIP: Lyon 6e Arrondissement
More details

Random Forest Classifier

Migration zone: Very favorable (7 / 7)

Figure 2: Screenshot of the cartographic interface of Predihood.

Adding new algorithms

Because prediction is a complex task, testing specific algorithms tuned with different parameters and comparing their results may help increase the overall quality. In order to facilitate this task, Predihood proposes a generic and easy-to-use programming structure for machine learning algorithms, based on Scikit-learn algorithms. Thus, experts can implement hand-made algorithms and run experiments in Predihood. Adding a new algorithm only requires 4 steps:

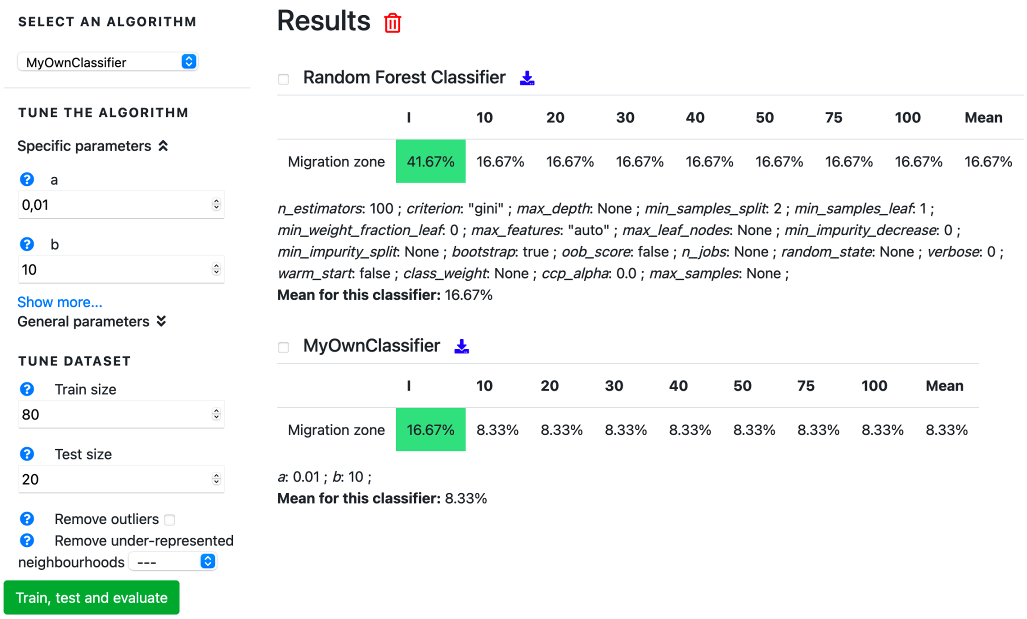
1. Create a new class that represents your algorithm, e.g. `MyOwnClassifier`, which inherits from `Classifier`;
2. Implement the core of your algorithm by coding the `fit()` and `predict()` functions. The `fit` function aims at fitting your classifier on assessed neighbourhoods while the `predict` function aims at predicting variable(s) for a given neighbourhood;
3. Add the `get_params()` function, which returns a dictionary of parameters along with their value, in order to be compatible with the Scikit-learn framework;

4. Comment your classifier with the Numpy style (Harris et al., 2020) so that Predihood automatically extracts its parameters and enables their tuning in the dedicated interface.

Below is a simple example to illustrate the aforementioned steps. Note that new algorithms are directly loaded into Predihood.

```
# file ./algorithms/MyOwnClassifier.py
from predihood.classes.Classifier import Classifier


class MyOwnClassifier(Classifier):
  """Description of the classifier.
  Parameters
  ------------
  a : float, default=0.01
    Description of a.
  b : int, default=10
    Description of b.
  """

  def __init__(self, a=0.01, b=10):
    self.a = a
    self.b = b

  def fit(self, X, y):
    # do stuff here

  def predict(self, df):
    # do stuff here

  def get_params(self, deep=True):
    # suppose this estimator has parameters "a" and "b"
    return {"a": self.a, "b": self.b}
```

To facilitate experiments, Predihood provides an interface for easily tuning and testing algorithms on a dataset, as shown in Figure 3. The left panel stands for the selection of an algorithm and the tuning of its parameters and hyper parameters, such as training and test sizes. Note that two options enable to remove outliers and under-represented neighbourhoods (for a given variable) without directly modifying the dataset. On the right, the table illustrates the accuracies obtained for each list of indicators (generated during the selection process) and each variable. Results can be exported in XLS with the blue download icon. Here, we notice that the new algorithm *MyOwnClassifier* has been chosen, and its parameters (*a* and *b*) can be configured. We have performed 2 runs, the former with the Random Forest classifier and the latter with MyOwnClassifier. Best predictive results are achieved with all indicators (green cells).



SELECT AN ALGORITHM

MyOwnClassifier

TUNE THE ALGORITHM

Specific parameters

a

0,01

b

10

Show more...

General parameters

TUNE DATASET

Train size

80

Test size

20

Remove outliers □

Remove under-represented neighbourhoods ---

Results

□ Random Forest Classifier

	I	10	20	30	40	50	75	100	Mean
Migration zone	41.67%	16.67%	16.67%	16.67%	16.67%	16.67%	16.67%	16.67%	16.67%

n_estimators: 100 ; *criterion*: "gini" ; *max_depth*: None ; *min_samples_split*: 2 ; *min_samples_leaf*: 1 ; *min_weight_fraction_leaf*: 0 ; *max_features*: "auto" ; *max_leaf_nodes*: None ; *min_impurity_decrease*: 0 ; *min_impurity_split*: None ; *bootstrap*: true ; *oob_score*: false ; *n_jobs*: None ; *random_state*: None ; *verbose*: 0 ; *warm_start*: false ; *class_weight*: None ; *ccp_alpha*: 0.0 ; *max_samples*: None ;
Mean for this classifier: 16.67%

□ MyOwnClassifier

	I	10	20	30	40	50	75	100	Mean
Migration zone	16.67%	8.33%	8.33%	8.33%	8.33%	8.33%	8.33%	8.33%	8.33%

a: 0.01 ; *b*: 10 ;
Mean for this classifier: 8.33%

Figure 3: Screenshot of algorithmic interface of Predihood.

Current applications of Predihood

Our Predihood tool was presented during the DATA conference (Barret et al., 2020) posing the evaluation of whether people choose a similar neighbourhood environment when moving elsewhere as the main research challenge. The tool is bundled with data from France using the mongiris project (in which unit divisions named IRIS stand for neighbourhoods), this dataset contains about 50,000 neighbourhoods with 640 indicators (about population, shops, buildings, etc.). Six environment variables have been defined (*building type*, *building usage*, *landscape*, *social class*, *morphological position* and *geographical position*), and 270 neighbourhoods were annotated by social science researchers (one to two hours per neighbourhood to investigate building and streets pictures, parked cars, facilities and green areas from services such as Google Street View). Prediction results achieved by Predihood using 6 algorithms from Scikit-learn range from 30% to 65% accuracy depending on the environment variable, and designing new algorithms could help improving these results.

The open-source project is available here: https://gitlab.com/fduchate/predihood.

Acknowledgements

This work has been partially funded by LABEX IMU (ANR-10-LABX-0088) from Université de Lyon, in the context of the program "Investissements d'Avenir" (ANR-11-IDEX-0007) from the French Research Agency (ANR). In addition to Scikit-learn and Numpy, Predihood relies on other dependencies, namely Pandas (team, 2020), seaborn (Waskom, 2021) and matplotlib (Hunter, 2007).

References

Barret, N., Duchateau, F., Favetta, F., & Bonneval, L. (2020). Predicting the environment of a neighborhood: A use case for france. *International Conference on Data*

Management Technologies and Applications (DATA), 294–301. https://doi.org/10.5220/0009885702940301

Cranshaw, J., Schwartz, R., Hong, J. I., & Sadeh, N. (2012). The livehoods project: Utilizing social media to understand the dynamics of a city. *International AAAI Conference on Weblogs and Social Media*, 58. https://ssrn.com/abstract=2168428

Frank, L. D., Sallis, J. F., Saelens, B. E., Leary, L., Cain, K., Conway, T. L., & Hess, P. M. (2010). The development of a walkability index: Application to the neighborhood quality of life study. *British Journal of Sports Medicine*, *44*(13), 924–933. https://doi.org/10.1136/bjsm.2009.058701

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Leong, M., Dunn, R. R., & Trautwein, M. D. (2018). Biodiversity and socioeconomics in the city: A review of the luxury effect. *Biology Letters*, *14*(5), 20180082. https://doi.org/10.1098/rsbl.2018.0082

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Takada, M., Kondo, N., & Hashimoto, H. (2014). Japanese study on stratification, health, income, and neighborhood: Study protocol and profiles of participants. *Journal of Epidemiology*, *24*(4), 334–344. https://doi.org/10.2188/jea.JE20130084

team, T. pandas development. (2020). *Pandas-dev/pandas: Pandas 1.1.4* (Version v1.1.4) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.4161697

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

Zhang, A. X., Noulas, A., Scellato, S., & Mascolo, C. (2013). Hoodsquare: Modeling and recommending neighborhoods in location-based social networks. *2013 International Conference on Social Computing*, 69–74. https://doi.org/10.1109/socialcom.2013.17