ML Poverty: Using Machine Learning to estimate poverty rates in Switzerland at the canton level

Authors: Yara Abu Awad, Christopher Sulkowski, Matthias Widmer, Lancelot Marti, Raphael de Fondeville, Stephan Haeni

Introduction: The SILC Survey and Poverty Indices

The SILC survey is an annual survey in over 30 European countries[1] which Switzerland began conducting in 2007[2]. The sampling plan uses a proportional, stratified design which is structured around seven major geographical regions and is currently based on the population register. The sample size typically consists of about 8'500 households made up of about 18'000 persons enrolled in a 4-year rotating panel design. Interviews are conducted via telephone (although starting in 2023, individual questionnaires can be filled online) and participants are asked to complete a household questionnaire (which one person answers for the entire household and takes 10 to 15 minutes) and an individual questionnaire (which is answered by all persons aged 16 or over and takes 25 to 30 minutes). Weights are estimated to adjust for non-response, loss to follow-up and for calibration to the reference population.

In combination with linked register and income data, survey responses are used to estimate multiple poverty indices including: Absolute Poverty, Relative Poverty and, Material and Social Deprivation. More information regarding these indices can be found in an official report entitled "Poverty Measurement in Switzerland"[3]. The results we will be presenting in this paper are only concerned with the "Absolute Poverty Rate"; a needs-based definition of a social subsistence level that not only guarantees physical survival but also a minimum level of participation in social life. People are considered as poor if they do not have the means to buy goods and services that are necessary for a socially integrated life. Disposable household income is compared to cost of: basic needs (food, clothing, personal care, transport, entertainment, education), housing and, other necessary expenditures such as liability insurance. This rate is estimated at the personal level according to the poverty status of the household. For the remainder of this paper, when we use the terms poverty / poor / poverty rate, we are referring to this absolute poverty variable.

Challenge

The sample size of SILC allows estimates at the level of grand regions (NUTS 2), but it is too small to generate robust estimates at canton level (there are 26 cantons in Switzerland which vary greatly in terms of population size). However, a sufficient increase in sample size for cantonal evaluations would be very expensive.

Approach of Statistics Austria

We studied the approach taken by colleagues at Statistics Austria who implemented a proof-of-concept study named LEARN4SDGis[4] which aimed to answer the questions: "How can important indicators be estimated for small areas? How can valid information be provided, for example, for poverty risk at any regional level - such as grid, census district, municipality, district or NUTS level?".

---

[1] EU statistics on income and living conditions - Microdata - Eurostat (europa.eu)
[2] Swiss Federal Statistical Office - Poverty and deprivation
[3] Poverty Measurement in Switzerland
[4] LEARN4SDGis - Machine Learning for Sample Data Geographic information systems

Supervised machine learning algorithms were trained (Random Forest, Boosting, Support Vector Machines and Neural Networks) to predict relative poverty using SILC survey responses linked with administrative and geographic data.

Considerations for Complex Survey Design

We wanted to extend the work of our Austrian colleagues while taking into account the complex survey design of the Swiss SILC survey, namely: the non-independence of observations since persons are clustered within household, the use of stratified sampling and the use of survey weights. The latter were considered quite important as they adjusted for potential biases within the data. A recent paper by MacNell et al.[5] (2023) concluded: "Failing to account for sampling weights in gradient boosting models may limit generalizability for data from complex surveys."

For this reason, we opted to use weights for model fitting. However, because our poverty variable was not balanced (a lot of zeroes and few ones), in order to ensure better model fitting, we calculated balancing weights by multiplying the weights of those with status poor = 1 by a constant equal to the ratio of: sum of weights among those with poor = 0/ sum of weights among those with poor = 1. This results in a weighted population where the probability of being poor equals the probability of not being poor.

Original weights were used in the calculation of model evaluation metrics since the quantity we are interested in predicting is the weighted poverty rate. Upon evaluating our models, we found that the use of weights in estimating metrics led to the choice of models with different hyperparameters.

Finally, the complex survey design affected the creation of folds for cross validation and testing. We used the advice given by Wieczorek et. al. (2022)[6] to sample at the level of primary sampling unit (PSU – in this case, the household) "so that the folds are a random partition of PSUs rather than of elementary sampling units". Sampling strata (in this case, grand region) were also taken into account when creating the folds.

Implementation Steps

We created reproducible pipelines to process data, train algorithms and estimate metrics using the R language on the Renku platform.

Step 1: Merge SILC survey data from the years 2018, 2019 and 2020 to register data (population, buildings and income) and geographic variables. This led to some loss of observations due to missingness. We evaluated whether or not this was differential with the use of Chi squared tests (accounting for survey design using the R survey package[7]) comparing poverty status within the missing and non-missing subsets of data in each year and concluded that the missingness could be ignored.

Step 2: Data preparation. This step involved data concatenation for models using all 3 years of data (a this step we did not account for the panel structure as we would be estimating rates for each year individually although we plan to account for it when estimating standard errors), the creation of dummy variables for categorical features and the creation of 10 folds using the criteria described above.

---

[5] MacNell, Nathaniel, et al. "Implementing machine learning methods with complex survey data: Lessons learned on the impacts of accounting sampling weights in gradient boosting." Plos one 18.1 (2023): e0280387.
[6] Wieczorek, J., Guerin, C., & McMahon, T. (2022). K-fold cross-validation for complex sample surveys. Stat, 11(1), e454.
[7] Lumley, Thomas. "Package 'survey'." *Available at the following link: https://cran. r-project. org* (2020).

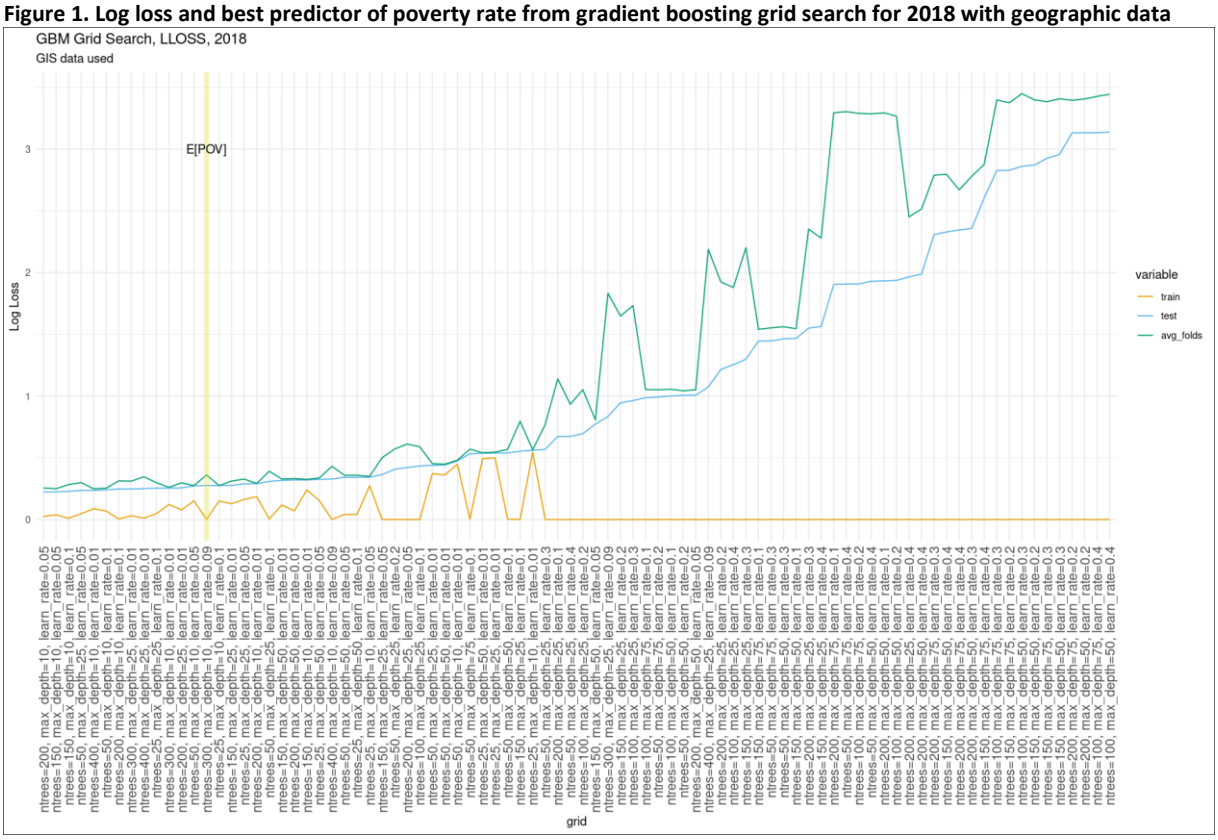
Steps 3 & 4: Run a grid search for each algorithm and generate metrics for each model. Although we initially explored neural networks, random forest and gradient boosting as potential models, we opted to drop neural networks as the other algorithms were performing comparably. The h2o package[8] was used as it simplified model fitting and the extraction of results. Additionally, models were fit to individual observations rather than households following the approach of Statistics Austria (we found that the variance of predictions within household was low so at first glance this does not appear to be an issue). The folds were used as follows during model fitting and evaluation; 9 were included in cross validation and 1 was held out as a test set. The grid search was an iterative process whereby hyperparameters were selected and then, based on the model metrics, more hyperparameters were explored etc. We also explored several strategies including: fitting one algorithm to all 3 years at once vs fitting each year separately and including vs excluding geographic data.

Model Selection

We focused on two metrics for model selection: the first was the log loss to provide a clear indicator of model quality relative to the data, and the second was the difference between the predicted and observed poverty rate in the test set to account for the model's ability to specifically provide predictions at the aggregate level.

Figure 1 illustrates our approach to choosing the best model. On the x-axis, hyperparameters of the gradient boosting grid search are displayed in order of lowest to highest log loss in the test set and the log loss is shown on the y axis. The log loss in the training and test sets and the mean log loss across the 9 cross validation folds (avg_folds) for 2018 SILC data merged to all predictors (including geographic data) are plotted. A vertical yellow line with the label E[POV] indicates which combination of hyperparameters yielded predictions that were closest to observed data. Based on this plot, we would select this model as the average log loss across folds is still quite low.

Figure 1. Log loss and best predictor of poverty rate from gradient boosting grid search for 2018 with geographic data



[8] H2O.ai. (2022) h2o: R Interface for H2O. R package version 3.38.0.2. https://github.com/h2oai/h2o-3.

Overall, gradient boosting models performed better than random forest and models fit to individual years and those that used geographic data also did better.

Next Steps

- Generating predictions at the population level: this will enable us to carry out post-hoc sensitivity analyses and better understand the behavior of the prediction model
- Estimating the uncertainty of the predictions: first using bootstrapping to estimate model uncertainty and then propagating this uncertainty into the Bernoulli approach illustrated by the Austrian team while accounting for clustering by household
- Use of privacy preserving techniques for dissemination of information: due to the potential of disclosure of these results at a smaller resolution (i.e. smaller than canton), we would like to use differential privacy when doing so and are looking into what this means for geographic data in collaboration with the OpenDP project.