

A TOOL FOR MACHINE LEARNING BASED DASYMETRIC MAPPING APPROACHES IN GRASS GIS

C. Flasse^{1*}, T. Grippa¹, S. Fennia¹

¹ Department of Geoscience, Environment and Society (DGES), Université Libre de Bruxelles, Avenue Franklin Roosevelt 50, Bruxelles 1050, Belgium – (tais.grippa, charlotte.flasse)@ulb.be; saf.a.fennia@gmail.com

KEY WORDS: GRASS GIS, Dasymetric Mapping, Machine Learning, Random Forest.

ABSTRACT:

Socio-economic and demographic data is often released at the level of census administrative units. However, there is often a need for data available at a higher spatial resolution. Dasymetric mapping is an approach that can be used to disaggregate such data into finer levels of detail. It relies on the assumption that proxies available at a higher spatial resolution, along with knowledge of an area, can be used to produce weights in order to spatially reallocate the data to a finer scale layer. The power and efficiency of machine learning (ML) approaches can be taken advantage of when producing weighted layers for dasymetric mapping. Less advanced users, however, may find these approaches too complex. To encourage a wider uptake of such approaches, easy-to-use tools are necessary. GRASS GIS is a free and open-source GIS software that contains many modules for processing geographic data. The existing GRASS GIS add-on “v.area.weigh” already makes the dasymetric mapping approach more accessible, however users must provide their own weighted layer. This paper presents the development of a GRASS GIS add-on, “r.area.createweight”, which provides a simple and convenient tool to facilitate the implementation of a ML-based approach to produce weighted layers for dasymetric mapping. The tool will be available on the official GRASS GIS add-on repository to encourage a more widespread uptake of these approaches.

1. INTRODUCTION

Socio-economic and demographic data is usually collected at the individual or household level, and numbers are then aggregated and released at the level of administrative units (Su *et al.*, 2010). The spatial extent of many phenomena, however, do not correspond to any existing administrative limits, making them difficult to exploit. Additionally, geospatial information has started to be available at more and more detailed spatial resolutions, thanks to progress made using high-resolution earth observation (EO) data. Consequently, scientists often aim to perform spatial analyses at a fine resolution, but face issues related to the fact that the spatial resolution of administrative units, on which socio-economic and demographic data are aggregated, is too coarse and does not fit their needs and specifications.

The spatial disaggregation of data from administrative units (i.e., from aggregated data), often assumes a uniform distribution, however this is unlikely to reflect real world patterns, as a large part of human activity is spatially heterogeneous. In recent decades, the dasymetric mapping approach (Wu *et al.*, 2005; Langford, 2007) has increasingly received attention in order to exploit socio-economic and demographic data for spatially detailed analysis and/or to explore spatial phenomena that do not follow existing administrative units (the modifiable areal unit problem - MAUP). This modelling technique relies on the assumption that the knowledge of an area can be used to unequally spatially disaggregate (or redistribute) socio-economic data provided at the administrative level, to create a more realistic, finer scale, gridded layer of disaggregated socio-economic data (Su *et al.*, 2010). This is often done through the use of proxy indicators, in order to create the weights necessary for the distribution of the administrative data into finer levels of detail. Consequently, the major challenge in dasymetric mapping resides in determining the spatial distribution of the socio-economic or demographic

variable within the administrative units, based on a set of ancillary geoinformation data.

In the example of human population counts, these weighted layers are typically produced using surrogate information provided by land cover (LC) and land use (LU) maps derived from EO data. For a long time, these weights have been subjectively determined based on expert knowledge (Mennis, 2003), where higher weights are attributed to urban areas, slightly lower weights to suburban or rural areas, and a weight of zero for forest areas or water bodies. Recent research, however, has shifted this paradigm by taking advantage of the power and the efficiency of machine learning (ML) algorithms to create weighting layers for dasymetric mapping without any a priori knowledge and in a data driven framework.

The WorldPop project, for example, uses the random forest (RF) algorithm as a flexible means to predict the weights for reallocation of population into grid layers, improving upon “existing, freely available population mapping approaches” (Stevens *et al.*, 2015). Grippa *et al.*, (2019) published a similar, replicable approach also using the RF algorithm for the creation of a weighting layer. The related computer code (Grippa, 2019) allows replicating the method, but it is very specific to the experiments presented in the paper and may not fit the needs of other scientists. Moreover, since it is computer code, potential users not skilled enough in the programming languages used (Python and R) could be reluctant to use it.

An important step of the approach has already been implemented in a GRASS GIS add-on, “r.zonal.classes” (Grippa, GRASS Development Team, 2019), which consists of the zonal extraction of class proportions from categorical raster data, that are the main proxies used in the ML approach. Further developments were still needed, however, to make the approach more accessible to a larger audience. While there are clear benefits to using improved disaggregation methods, the

development of simple and convenient methods or tools are necessary to encourage a more widespread uptake of these approaches by potential user communities, for example by remote sensing and GIS scientists.

2. AIMS AND OBJECTIVES

The dasymetric mapping approach has already been made more accessible with an existing GRASS GIS add-on “v.area.weigh” (Metz, GRASS Development Team, 2013), available on the official GRASS GIS add-on repository. It provides a tool for dasymetric mapping, however it requires that the user provide their own weighted layer. The aim of this work is to provide a ready-to-use tool, accessible to non-programmer users, to facilitate the implementation of a ML-based approach to produce weighting layers for dasymetric mapping of variables based on ancillary spatial layers. In the presented example, we make use of LC and LU maps to disaggregate population count data, as a proof of concept.

The “r.area.createweight” add-on presented in this paper has been developed as a GRASS GIS add-on to achieve this aim. The proposed layout of the add-on in GRASS GIS is shown in Figure 1. Along with the existing “r.zonal.classes” add-on (Figure 2), it implements the approach of Grippa *et al.*, (2019) in a more generic and user-friendly manner, that better fits the needs of different potential users. It allows users unfamiliar with coding or ML algorithms to use these advanced approaches and create a weighted layer which can then be used as input to the existing “v.area.weigh” add-on for dasymetric mapping.

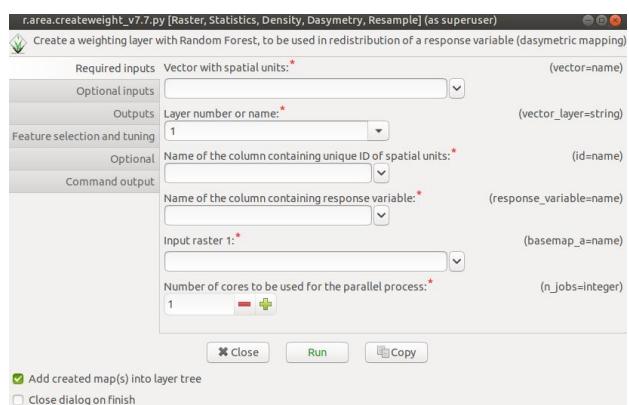


Figure 1. Proposed layout of the “r.area.createweight” add-on in GRASS GIS

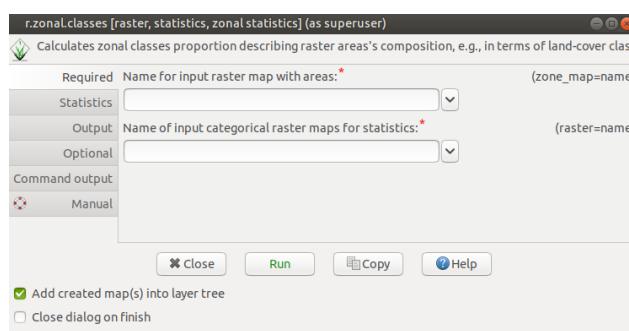


Figure 2. Part of the processing step has already been implemented in a user friendly click-button tool called “r.zonal.classes”

This tool will be available under a complete open-source license and will be accessible on the official GRASS GIS add-on repository. In this way, the tool will be able to reach a large and international audience in GIS and EO research. To our knowledge, there is no other existing open-source and ready-to-use tool, with a Graphical User Interface (GUI) for creation of dasymetric mapping weighting layers, using a ML approach.

3. THE GRASS GIS PROJECT

GRASS GIS (GRASS Development Team, 2020) is a powerful free and open-source GIS software. It was developed in the 80’s and has since been maintained and regularly improved thanks to its active international developer team. It is characterized by its ability to efficiently process raster data, in addition to manipulating vector and 3D formats, simple to advanced spatial analysis and modelling, and connecting to spatial databases.

GRASS GIS is built using a system of modules, and its’ standard distribution currently contains over core 500 modules which enable the processing of geographic data. As GRASS is a Free and Open-Source Software (FOSS), this enables the wider community to both benefit from, and also contribute to software development. Indeed, the GRASS developer team encourage users to create their own tools, in addition to contributing to the improvement of existing tools. As such, there also exists over 300 extensions, or addons, in the official GRASS addons Github repository (<https://github.com/OSGeo/grass-addons>), not to mention those that are currently under development.

This “r.area.createweight” add-on contributes, therefore, to the development of the GRASS GIS software. The source code is consequently also free and open source. This is essential for allowing any user to refer to it for more details on the functioning of the code, or even to develop future improvements. It will be available and maintained through the official GRASS GIS add-ons repository.

4. METHOD

4.1 General workflow

The major steps of the workflow are provided in Figure 3. The add-on first pre-processes the input data then statistics are then calculated at two levels, one at the level of spatial units in order to first calibrate and train a Random Forest (RF) regressor, then at the level of the output grid to create a weighted layer using the trained RF model. This weighting layer can then be used for redistribution of a response variable (e.g., population count) known at a spatial unit level (e.g., administrative units), into a raster grid with a finer spatial resolution (dasymetric mapping).

4.2 Initial data preparation

The add-on requires two types of datasets to be provided. Firstly, a vector of spatial units with attribute table containing, for each unit, the information to be disaggregated (the response variable e.g., population count), and secondly at least one categorical raster (a.k.a. basemap) that provides information (such as LC or LU) which is associated with the response variable and can be used to predict the spatial disaggregation of the response variable. The add-on allows the user to optionally include a second categorical raster, and/or a raster of continuous numerical values (e.g., distance to the nearest point of interest, temperatures, etc.).

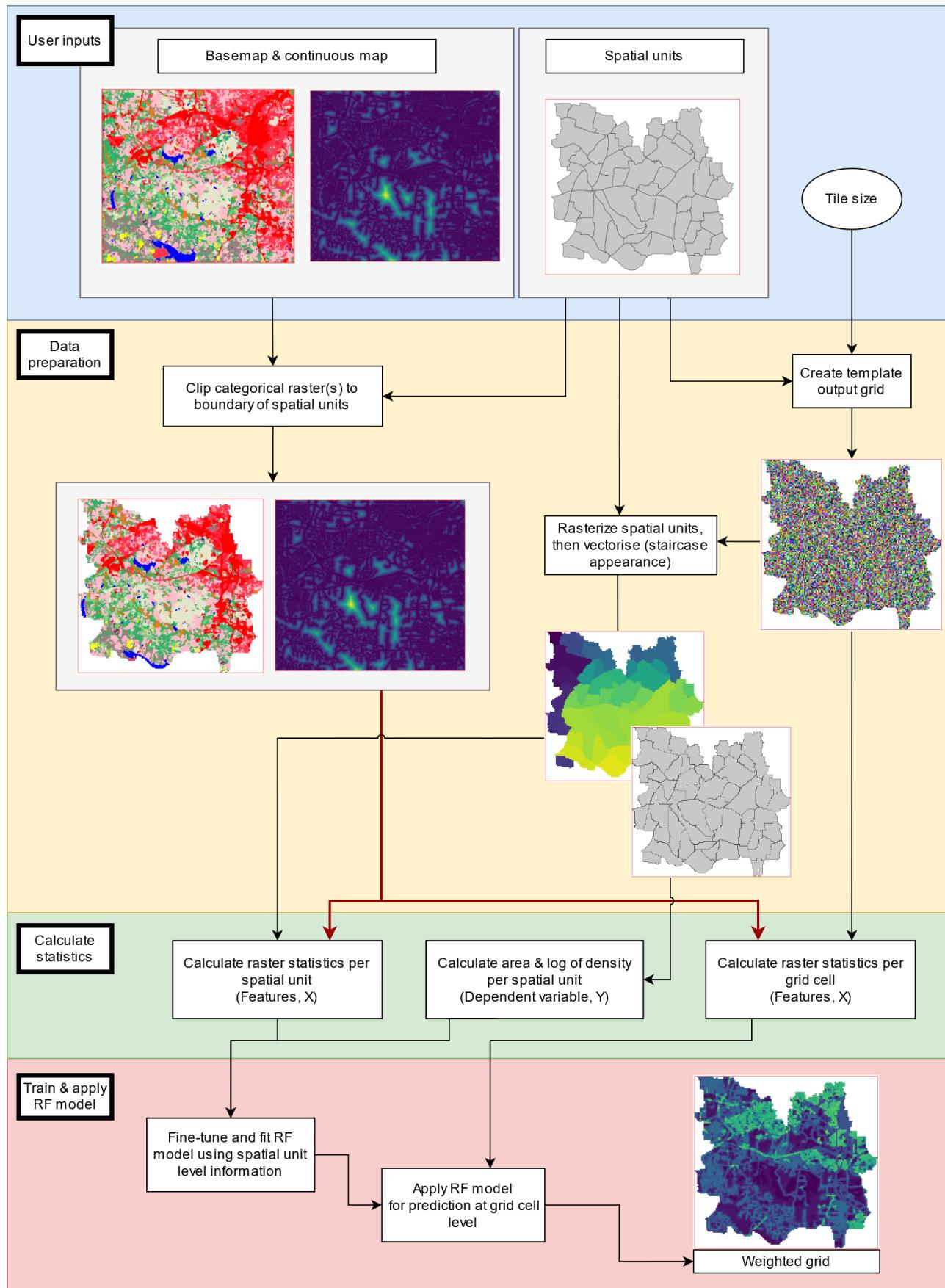


Figure 3. General workflow

A template of the output grid is also created, using the user-defined spatial resolution, or pixel size, and the spatial coverage of the spatial units provided. The pixel size should be coarser than that of the input raster layers to avoid spatial scale mismatches. Predicting a RF model using raster layers that have a coarser spatial resolution than the output predicted layer would reduce the predictive accuracy and reliability of the model. As such, the add-on will produce an error if the user-defined pixel size is of a higher spatial resolution than the first categorical raster provided.

The input spatial units are rasterised to the extent, cell size and alignment of the output weighted grid, and are then vectorised (Figure 4). This results in spatial units whose boundaries will have a 'staircase' appearance, ensuring that each pixel of the output weighted grid will be contained in only one spatial unit. This can result in some polygons disappearing if the original vector contains small (or very narrow) polygons, and the desired tile-size is too large. In this case, the add-on will also produce an error to the user about the loss of observations.

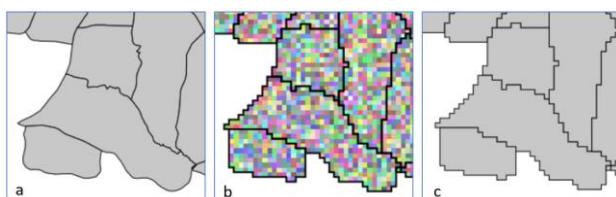


Figure 4. Spatial units a) input, b) raster grid overlaid with final gridded vector, c) final gridded vector (data adapted from GRASS GIS sample data North Carolina, USA).

4.3 Dasymetric framework

Statistics are calculated from all the input rasters. For each of the categorical rasters (a.k.a basemaps), the proportion of each of the classes present is calculated. For the continuous raster map, if included, the average values are calculated. These extracted covariates are then used to train the RF model. By default, all present classes will be taken into account, however, the user has the option to define a list of specific classes should they not wish to take all classes into account.

These statistics are calculated at two levels: first at the level of the spatial units, and second for each pixel of the output grid layer. The model is first trained and fitted at the level of the spatial units. The model is then used to predict the response variable for each pixel of the output grid.

Prior to fitting the RF model, the variable to be predicted (e.g., population count) is first transformed. The density per spatial unit is calculated (e.g., population density) and then the natural log is taken. Previous research has suggested that the quality of the prediction of weights is improved when log-transformation is applied to the response variable prior to RF model fitting (Stevens *et al.*, 2015). The response variable of the RF model is therefore the log of the variable density.

Once the RF model has been trained and fitted, it predicts the response variable (log of variable density) for each pixel of the output grid. The log is then back transformed, to retrieve the variable density (e.g., population density), producing the final weighted grid. For each cell of the output grid, a weighted prediction will only be made if statistics are available for that cell from each of the input raster layers. This means that any data gaps, or no-data cells in any one of the input raster layers will result in a no-data value for that cell in the output weighted

layer. The weighting layer can then be used for standard dasymetric mapping with the “v.area.weigh” add-on.

4.4 Random Forest model

RF is a non-parametric supervised ML algorithm. It is an ensemble method that can be used both for regression (as in this add-on), or classification. The RF algorithm builds an ensemble of decision trees (CART). Each decision tree is trained using bootstrapped samples, i.e., random sampling with replacement of the training data. At each node of the tree, a random subset of features is analysed, and the node uses the feature that optimally splits the observations. Each tree in the forest provides a predicted value. In the case of regression, the average value of all the trees in the forest is calculated, to produce the final prediction. Individual decision trees are sensitive to the data they are trained on and are very prone to overfitting. The combined use of many trees, bootstrapped samples, and random selection of features, however, ensures that the RF model is relatively resistant to overfitting and able to ensure a strong degree of generalization.

The RF model is also advantageous, as it only has a few (hyper)-parameters which can be optimised to tune the model and improve its’ performance. The most important parameters are the number of trees in the model, and the number of randomly selected features at each tree node.

“r.area.createweight” makes use of the Python sci-kit learn (sklearn) ML library (Pedregosa *et al.*, 2011) to train, fine-tune and use the RF model. A RF model is first trained using the default sklearn parameters, with 500 trees. The importance of features is estimated and the add-on, by default, removes the features with an importance of less than 0.5%. This allows the elimination of features that have little or no impact on the prediction. The user can, however, optionally choose to keep all features regardless of their importance.

The RF model is then fine-tuned for its parameters. This is facilitated by using a grid-search algorithm where every combination of a range of given parameters is used to train different models. The performance of each model is assessed using a k-fold cross-validation. This splits the data into k-sections (folds) and trains the model k-times, each time leaving out 1-fold of samples. The performance is assessed with the out-of-bag (OOB) accuracy, a pseudo-independent validation measure that is known to be reliable to assess model performance. It uses the ‘left-out’ samples to test the accuracy of the model’s predictions and select the best performing model. In “r.area.createweight”, a pre-defined default range of parameters and a 5-fold cross validation are used. The add-on, however, allows more advanced users to optionally define a custom range of hyper-parameters, and/or a different number of folds for the cross validation. This allows a greater flexibility for users who are more familiar with RF.

The resulting fine-tuned RF model is then used to predict the response variable for each pixel of the output weighted grid. Each tree in the RF model makes a prediction, and the average of these predictions is taken to provide the final response variable prediction. The add-on also provides the user with a log file containing details such as the selected features, the hyper-parameters tested, the optimally selected parameter combination, and measures of exactitude (OOB score). A graph of the features used, and their importance, is also provided to the user. For more information about RF, please refer to the original publication (Breiman, 2001).

4.5 Methodological considerations

The method used contains a number of assumptions and methodological elements to be taken into consideration.

Regarding the input data, the spatial coverage of the spatial units by the ancillary data layers will affect the quality of the RF model. Best practice is for the spatial units to be completely covered by each of the ancillary data layers, which will result in a better RF model, and better predictions. In the case that the spatial units are not completely covered, the RF algorithm may be missing important data for training the model. In addition, predicting the response variable weights is only possible for pixels where there is coverage from all the ancillary data layers. Therefore, when spatially reallocating the value given in the initial spatial units, values will only be predicted for cells where a weight exists and it will be as if the non-covered area is weightless, resulting in overestimated values in the dasymetric process. There may be cases where some missing data is acceptable, as will be shown with the case study. The add-on leaves it up to the user to decide what levels of coverage are acceptable for their analysis, and to remove any spatial units with insufficient coverage, prior to analysis.

A key assumption of the RF approach used here, is that there exists an association between the covariates (e.g., land cover) and the response variable (e.g., population). The RF model is strongly dependent on the quality of the input data, as such its' predictive accuracy will only be as good as the data it is given. In addition, as the RF model is trained and predicted at different spatial scales, there is also the assumption that the relationship between the covariates and the response variable stays constant at both the spatial unit level, and the level of the output grid cells. In reality, this is unlikely to quite be true, and depends strongly on the afore mentioned MAUP, i.e. the information at a specific spatial scale depends on the shape, but also the scale at which data is aggregated. Indeed, Grippa *et al.*, (2019) found that the best covariates for predicting distribution vary from one spatial scale to another.

Also as a result of training at one spatial scale, but prediction at another, is the need to be cautious when using the OOB-score to measure the performance of the model for reallocation. When training the RF model with k-fold cross validation, the RF internal validation computes the OOB score, or error. This accuracy assessment metric measures the ability of the model to predict the response variable on data left out of the training dataset but does so for a specific spatial level (the level of the aggregated spatial units). The model predicts, however at another spatial scale. (Grippa *et al.*, 2019) found that as a result, the OOB score may not be very effective for measuring the performance of the model in reallocation.

We must also mention the constraints linked to the use of the natural log. While using the natural log increases the accuracy

of the prediction, this requires all input spatial units to have a non-zero value for the variable to be predicted. This, in combination with the fact that the RF algorithm can only predict values within the range of values on which it was trained, means that the RF model will be unable to make a prediction of zero. The add-on has been designed to force a value of zero in the weighting layer if the predicted weight is smaller than 0.000000001 obs./m².

Lastly, the user should keep in mind that RF is a predictive modelling tool, and that it can only provide an approximation, albeit a useful approximation, of reality. “*All models are wrong; some models are useful*” (Box *et al.*, 2005) (p.440).

5. CASE STUDY

Population distribution is often highly correlated with LC and LU. This example uses very high-resolution LC and LU data (0.5 m and 5m respectively) to produce a gridded weighted layer, i.e., predicted population density, at a resolution of 100 m for the city of Ouagadougou. This weighted layer is then used for dasymetric mapping, redistributing the population count per administrative unit to a more spatially detailed grid.

The LC and LU data are publicly available in scientific repositories (Grippa, Georganos, 2018) and (Grippa, Georganos, 2019) respectively. The LU dataset is provided in vector format and was rasterized to 5 m resolution. Similar datasets for the city of Dakar have previously been used for population redistribution (Georganos *et al.*, 2019; Grippa *et al.*, 2019). The administrative units containing population counts were obtained from the Institut National de la Statistique et de la Démographie of Burkina Faso (INSD, 2012). The add-on “r.area.createweight” was run using these three inputs and specifying an output grid resolution of 100 m. The default parameters were used, although 16 cores were used for parallel processing of these large datasets.

The population density was calculated for each administrative unit, then was log-transformed, to obtain the response variable of the RF model. The add-on calculated the proportion of available classes within the LC and LU layers to train the RF model at the level of the administrative units. Once trained and fine-tuned, the RF model predicted the response variable (i.e., log of the population density) for each pixel of the output grid. The add-on then back transformed the predictions, to retrieve population density values. The final weights consist precisely of these gridded predicted population density values. The inputs and the final weighted grid are presented in Figure 5. The output plot of feature importances is shown in Figure 6 and shows that the features with the most importance for prediction are the ‘Low buildings’ of the LC basemap and the ‘Planned’ residential of the LU basemap. The OOB score for the model was 0.725.

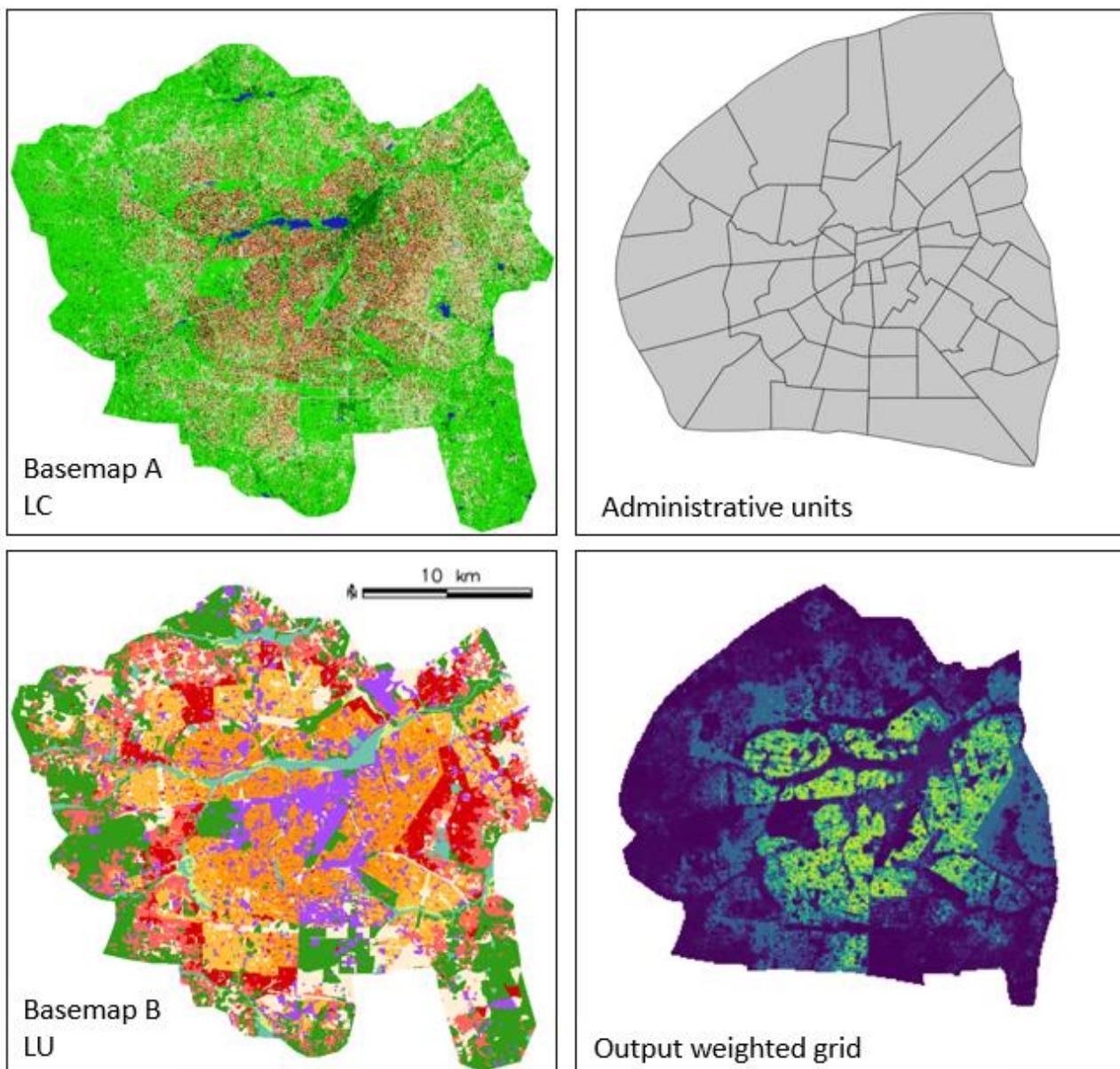


Figure 5. Creating a weighted layer for disaggregation of population count

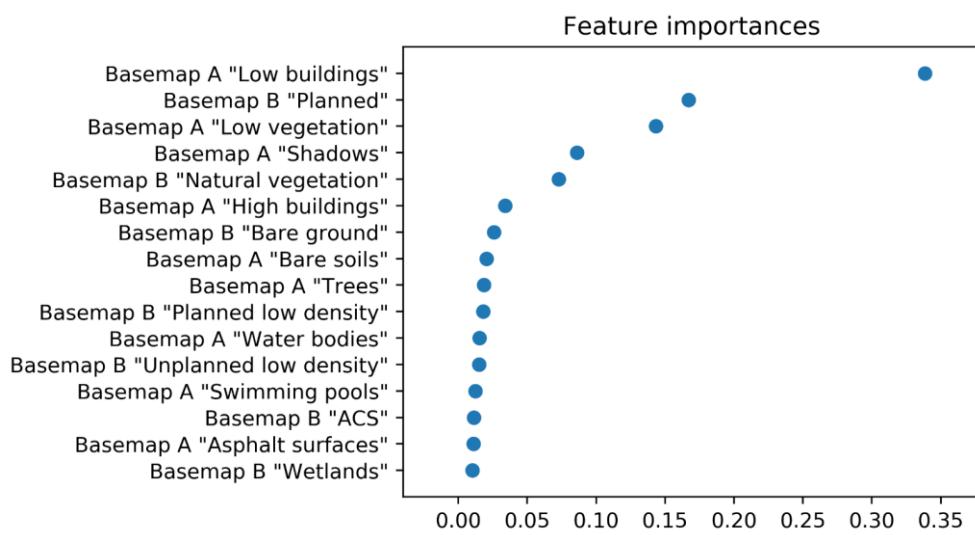


Figure 6. Feature importances

The weighted grid was then used with the “v.area.weigh” add-on to predict population count per grid cell. Figure 7 compares the weighted grid and the final population density for a section of Ouagadougou. The two classes with greatest feature importance were the “Low buildings” (LC map) and “Planned” (LU map). We can see that the unplanned area (LU class 4) has lower predicted population counts than the planned area (LU class 7). By looking at the LC map this can be understood by the higher proportion of high buildings (LC class 111) in the planned class than the unplanned class. Vegetated zones (LC

classes 30 and 31) and Administrative, Commercial, Services areas (ACS, LU class 5) have low predicted population counts. From our knowledge of where people reside, this is as we could expect.

This example shows, therefore, the ease with which the add-on can be used, with no programming knowledge, and where with few inputs and using default parameters, dasymetric mapping for the disaggregation of population count in Ouagadougou was carried out using ML techniques.

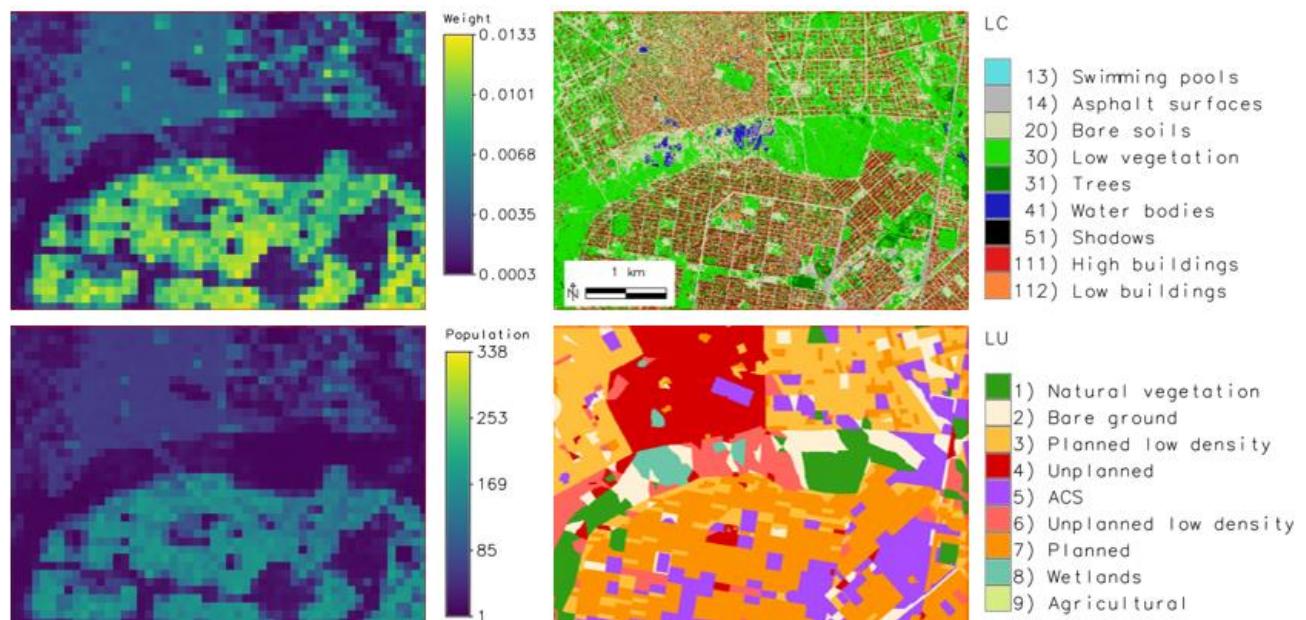


Figure 7. Comparing the LC, LU, weighted layer and final predicted population counts.

We mentioned previously that it is good practice for the spatial units to be completely covered by each of the ancillary data layers. As can be seen in Figure 8, this is not the case here and several of the spatial units are not completely covered by the LC and LU data (they have the same coverage). While, generally speaking, this should be avoided, it can in some situations can be acceptable. Our aim, in this example, is to disaggregate population count, using LC and LU ancillary data. Through additional expert knowledge of the area, we know that the zones not covered by the LC and LU maps are in fact zones with sparse or no population, meaning therefore that the population present in the area covered by the LC and LU data for a spatial unit, will approximate the total population for that entire spatial unit. In addition, it is useful to be reminded that dasymetric mapping only provides a model of population distribution, i.e., is only an approximation of reality, and so will in any case contain some errors. As such, it is assumed in this case that the impact of this non-complete coverage on RF training and prediction is minimal.

If a spatial unit were to not be completely covered by the ancillary data, but that the non-covered zone in fact contained a high presence of population, then population estimates in the area covered by the ancillary data for that spatial unit would strongly overestimate population numbers.

With this add-on, while higher coverage of ancillary data is preferable, it is up to the user to choose the level of coverage acceptable for their analysis. If in doubt, it is better for the user to first remove the spatial units with low coverage, prior to using the add-on.

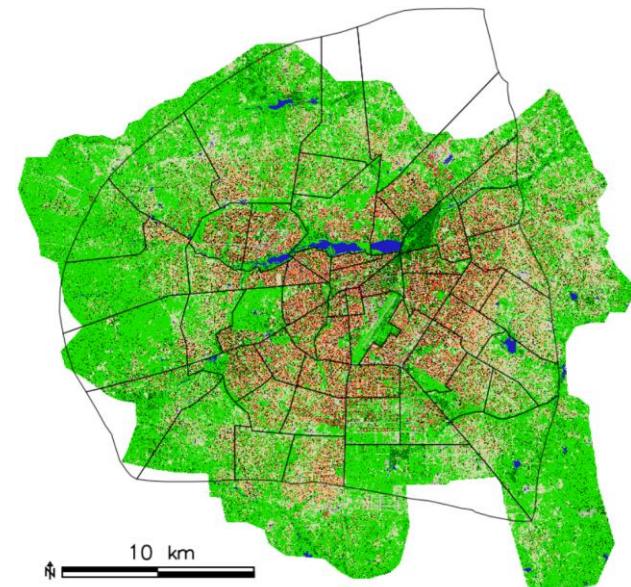


Figure 8. Spatial coverage of the administrative zones by the LC dataset, Ouagadougou

6. CONCLUSIONS & FUTURE CONCEPTS

The “r.area.createweight” GRASS GIS add-on presented in this paper provides a user-friendly tool, accessible to non-programmer users, which greatly facilitates the implementation of a ML-based approach to produce weighting layers for

dasymetric mapping. In addition, the add-on provides more advanced users with the option to define different hyper-parameters for fine-tuning the RF model. As a GRASS GIS add-on, the approach is easily and freely accessible, allowing for increased uptake and use. The add-on therefore will allow users to rapidly and readily perform dasymetric mapping, creating datasets that can be used in many applications such as the example given disaggregating population data.

ACKNOWLEDGEMENTS

This work was funded by the Belgian Federal Science Policy Office (BELSPO) (Research Program for Earth Observation STEREO III, contract SR/11/205) as part of the DASYWEIGHT project). The authors are grateful to the MAUPP (<http://maupp.ulb.ac.be>) and REACT (<http://react.ulb.be>) projects, funded by BELSPO, for the publicly available LC and LU datasets used for the case study, and to the Institut National de la Statistique et de la Démographie (INSD) of Burkina Faso for the population data at administrative level, accessed via the MAUPP project. The authors sincerely thank Stefanos Georganos for his proof-reading and advice.

REFERENCES

- Box, G. E. P., Hunter, W.H., Hunter, S. 2005. *Statistics for Experimenters: Design, Innovation, and Discovery*. 2nd edn. Wiley Series in Probability and Statistics; Wiley-Interscience: Hoboken, NJ, USA.
- Breiman, L. 2001. Random Forests, *Machine Learning*, 45, 5–32. doi.org/10.1023/A:1010933404324.
- Georganos, S., Grippa, T., Niang Gadiaga, A., Linard, C., Lennert, M., Vanhuysse, S., Mboga, N., Wolff, E., Kalogirou, S. 2019. Geographical random forests: a spatial extension of the random forest algorithm to address spatial heterogeneity in remote sensing and population modelling, *Geocarto International*, 36(2), 121–136. doi.org/10.1080/10106049.2019.1595177.
- GRASS Development Team, 2020. Geographic Resources Analysis Support System (GRASS) Software. Open Source Geospatial Foundation. <https://grass.osgeo.org> (1 June 2021).
- Grippa, T., Linard, C., Lennert, M., Georganos, S., Mboga, N., Vanhuysse, S., Gadiaga, A., Wolff, E. 2019. Improving urban population distribution models with very-high resolution satellite information, *Data*, 4(1), 1–17. doi.org/10.3390/data4010013.
- Grippa, T., Georganos, S. 2018. *Ouagadougou very-high resolution land cover map (v1.0)* [Dataset]. Zenodo. doi.org/10.5281/zenodo.1290654.
- Grippa, T., Georganos, S. 2019. *Ouagadougou land use map at street block level (v2.0)* [Dataset]. Zenodo. doi.org/10.5281/zenodo.3238302.
- Grippa, T., GRASS Development Team, 2019. Addon r.zonal.classes. Geographic Resources Analysis Support System (GRASS) Software, Version 7.8. Open Source Geospatial Foundation. <https://grass.osgeo.org/grass78/manuals/addons/r.zonal.classes.html> (7 June 2021)
- INSD Institut National de la Statistique et de la Démographie of Burkina Faso, 2012. *Ouagadougou Census Data, 2012, urban sectors* [Dataset].
- Langford, M. 2007. Rapid facilitation of dasymetric-based population interpolation by means of raster pixel maps, *Computers, Environment and Urban Systems*, 31(1), 19–32. doi.org/10.1016/j.compenvurbsys.2005.07.005.
- Mennis, J. 2003. Generating surface models of population using dasymetric mapping, *The Professional Geographer*, 55(1), 31–42.
- Metz, M., GRASS Development Team, 2013. Addon v.area.weigh, Geographic Resources Analysis Support System (GRASS) Software, Version 7.8, Open Source Geospatial Foundation. <https://grass.osgeo.org/grass78/manuals/addons/v.area.weigh.html> (7 June 2021)
- 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 {P}ython’, *Journal of Machine Learning Research*, 12, 2825–2830.
- Stevens, F. R., Gaughan, A. E., Linard, C., Tatem, A. J. 2015. Disaggregating census data for population mapping using Random forests with remotely-sensed and ancillary data, *PLoS ONE*, 10(2), 1–22. doi.org/10.1371/journal.pone.0107042.
- Su, M. D., Lin, M. C., Hsieh, H. I., Tsai, B. W., Lin, C. H. 2010. Multi-layer multi-class dasymetric mapping to estimate population distribution, *Science of the Total Environment*, 408(20), 4807–4816. doi.org/10.1016/j.scitotenv.2010.06.032.
- Wu, S. S., Qiu, X., Wang, L. 2005. Population estimation methods in GIS and remote sensing: A review, *GIScience and Remote Sensing*, 42(1), 80–96. doi.org/10.2747/1548-1603.42.1.80.