SUBNATIONAL MAP OF POVERTY GENERATED FROM REMOTE-SENSING DATA IN AFRICA: USING MACHINE LEARNING MODELS AND ADVANCED REGRESSION METHODS FOR POVERTY ESTIMATION

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Industrial Engineering

by

Lionel N. Hanke

September 2021

©2021

Lionel Norbert Hanke

ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Subnational map of poverty generated from remote-sensing data in Africa: Using machine learning models and advanced regression methods for poverty estimation

AUTHOR: Lionel Norbert Hanke

DATE SUBMITTED: September 2021

COMMITTEE CHAIR: Tali Freed, Ph.D.
Professor of Industrial Engineering

COMMITTEE MEMBER: Reza Pouraghabagher, Ph.D.
Professor of Industrial Engineering

COMMITTEE MEMBER: Lizabeth Thompson, Ph.D.
Professor of Industrial Engineering

ABSTRACT

Subnational map of poverty generated from remote-sensing data in Africa: Using machine learning models and advanced regression methods for poverty estimation

Lionel N. Hanke

According to the 2020 poverty estimates from the World Bank, it is estimated that 9.1% - 9.4% of the global population lived on less than $1.90 per day. It is estimated that the Covid-19 pandemic further aggravated the issue by pushing more than 1% of the global population below the international poverty line of $1.90 per day (WorldBank, 2020). To provide help and formulate effective measures, poverty needs to be located as exact as possible. For this purpose, it was investigated whether regression methods with aggregated remote-sensing data could be used to estimate poverty in Africa. Therefore, five distinct regression frameworks were compared regarding their R^2 value and the mean absolute relative percentage error when estimating poverty from aggregated remote-sensing data in continental Africa. A total of 12 regression models were developed at the three poverty rates at the $1.90, $3.20, and $5.50 income level per day and can be divided into direct models, two-step models, and ensemble models. It was found that ensemble methods perform better than simpler models, with an R^2 value of 0.74 for the ensemble neural net and 0.80 for the ensemble xgboost model. The best performing one step model is the kernel ridge regression with an R^2 of 0.72, while the remaining frameworks of this type all perform worse. Bayesian ridge regression models consistently performed the worst compared to the other frameworks under investigation. It was found that it the model estimations were most stable at the daily income level of $1.90 and $3.20, which can be explained by the increasingly skewed distribution of target values for higher poverty thresholds. Overall, it was found that xgboost, kernel ridge regression and artificial neural networks perform better than the other models.

ACKNOWLEDGMENTS

I would like to thank my family and friends for their never ending support during college and my adventures far from home. Additionally, I want to thank DataHow AG for introducing me into the field of machine learning which ultimately started my passion for the subject and data analysis in general.

I would also like to thank my partners from the World Bank for making this project possible, for their assistance and guidance as well as the World Bank in general for their work making the world a better place for everyone and collecting and providing the data for this study.

Thank you to my advisor Dr. Freed who not only guided me during this study, but has also been a key constant during my time at Cal Poly in general.

Thank you Dr. Thompson, for your helpful feedback and agreeing to be part of the thesis committee on such a short notice.

Lastly, I want to thank Dr. Pouraghabagher for all the times I had the pleasure to be in one of your classes and the great experiences which further grew my passion for data, numbers, and analysis as well as your unbiased opinion as part of the thesis committee.

Table of contents

List of Tables

List of Figures

Introduction

Background

The World Bank is a global partnership of 189 member countries. It was initially founded to help rebuild European countries which have been partially ruined in World War II. Soon after, other entities took over financing and organization of the rebuilding process so the World Bank shifted its attention to funding infrastructure projects in Latin America (World Bank, 2020a).
Since the 1970s, World Bank's primary focus lays on poverty eradication and over time expanded to social development in general. Thus, the organization has a lot of experience of generating solution strategies that have a meaningful impact for the people affected by a problem. Nonetheless, help can only be provided if the problem can be located.
This is the underlying issue in this work, as poverty data is still scarce in coverage and expensive to obtain in the field. However, it is fundamental to understand the spatial distribution of poverty for poverty reduction programs. Only if the spatial resolution is high enough can poverty maps be efficiently employed by policy makers. Thus, the aim of this work is to build a model that predicts poverty within reasonable accuracy limits in existing countries, and where data is only scarce or not available at all. This would allow not only World Bank's poverty economists to formulate better solutions for affected regions in the world, but also aid national statistical offices, the international development research community, and policy makers. Ideally the solution produced in this work produces results that are consistent over time and will be used again whenever updated data becomes available.

Research Significance

The World Bank would like to generate detailed accurate spatial poverty maps consistent over time using a novel and robust system based on machine or deep learning. For this purpose, the World Bank suggests the use of the following

data:

- Subnational Poverty Estimates
- Small Area Estimations
- Household Survey Data
- Freely available remote-sensing data

Furthermore, the World Bank aims to improve and simplify existing poverty mapping approaches by offering an easily transferable service for map generation that can be employed by policy makers. Historically, poverty data has been sourced locally. For that reason, poverty data is often scarce, labor intensive to obtain and overall expensive (Xie, Jean, Burke, Lobell, & Ermon, 2015). However, disaster relief, food security and sustainable development can only really be achieved with precise, well available data for all regions of interest. The lack of reliable data mainly affects developing countries, which in turn prevents formulation of effective measures in those areas. There is an urgent need for a solution that can estimate poverty by analysis of readily available data. Nowadays, remote-sensing data such as high resolution satellite imagery can be easily obtained for little to no money, and machine learning models have shown that they can be used to extract features from seemingly unstructured input data. Additionally, there is an increasing number of preprocessed data sets which are readily available online. For the purpose of this work we want to capitalize on these new possibilities and explore regression approaches to estimate poverty on different administrative levels.

Research Questions

1. Which regression models are most promising to estimate poverty on sub-national administrative levels?
2. What data sets are best used in those models?
3. Which Features contribute most to the model's output?

Problem Description

The World Bank has been able to increase the frequency of reporting global poverty estimates in the past 20 years. Until 2008, estimates were presented every three years. Since then, implementation of new surveys, updates on existing surveys and additional data sources made a more frequent estimation of poverty possible (World Bank, 2018). Spatial poverty maps are primarily generated by small area estimation methods. This technique relies heavily on household survey data, where thousands of people are interviewed. This involves a lot of work and effort, however the sparsity of data coverage typically does not allow high levels of resolution when used without additional data sources (Bedi, Coudouel, & Simler, 2007). Only when detailed household survey data is combined with comprehensive data like from a national census or survey can the small area estimation method provide sufficiently accurate poverty estimates for policy makers.

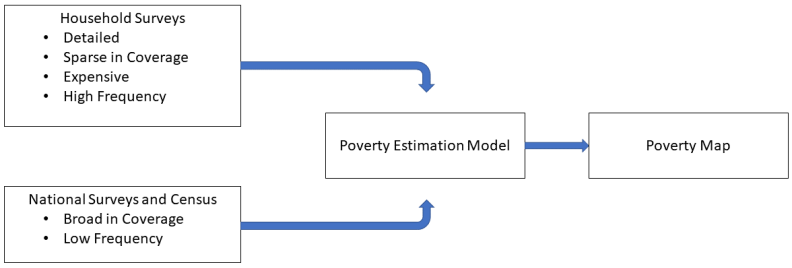
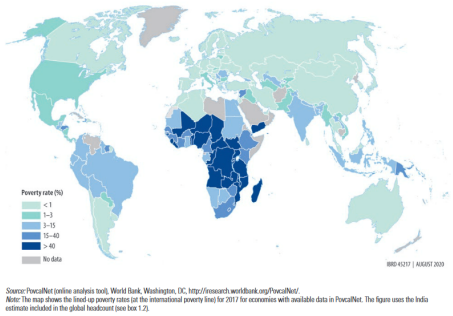


Figure 1. *Visualization of Small Area Estimation Approach*

Figure 1 visualizes the small area estimation models. Detailed surveys are combined with more general information from census allowing the model to make reasonably good estimations of poverty that can be visualized in poverty maps. Although at least two sources of data are considered, the power, flexibility and adaptability of these estimation models are limited. First, an

analyst uses multiple regression to build a model for household consumption from the survey data. However, that model can only use variables that are present in both datasources. This allows using the model's parameters on the census data to get a biggger picture for consumption in all households. Nonetheless, these estimations are not very accurate and simulation methods must be applied to account for the inaccuracy of the prediction model. The estimates are then presented in a GIS, where additional data can be overlayed for correlation analysis (Bedi et al., 2007). By 2018, data-sources and methodology have improved, however the core process of poverty mapping is still the same (World Bank, 2018).

(a) *National Poverty Map*



(b) *Sub-National Poverty Map*

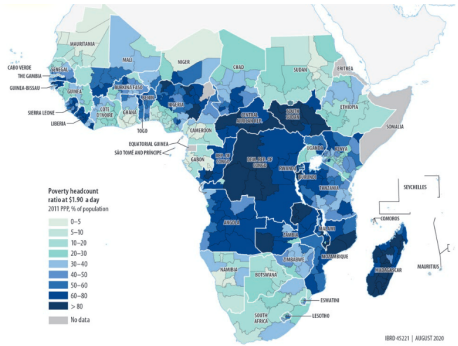


Figure 2. *Comparison of National and Sub-national Poverty Map.*

In figure 2a and figure 2b poverty estimates from World Bank (2020b) for the year 2018 are visually presented in poverty maps. While national maps are easier to create, they are not very valuable for policy makers and economists because the resolution is not high enough. sub-national maps on the other hand, are very valuable for targeted interventions and poverty reduction programs, as they localize poverty more accurately. However, this requires more data, additional analysis and more sophisticated models.

The lack of frequently updated data is still one of the major problems for generation of poverty maps (World Bank, 2020b). This is exemplified by India,

which could not release updated poverty estimates in 2017/2018. This left a big gap in understanding poverty in India but also affected broader areas due to the immense population and the geographical location in asia. This is worrisome, as poverty is still a big concern in that geographical area.

Literature Review

Conventional Poverty Estimation Methods

Disaster relief, food security and sustainable development can only really be achieved with precise, well available data in all countries of interest. The lack of reliable data complicates providing humanitarian aid, which unfortunately mainly effects developing countries. Poverty data is especially scarce because it is labor intensive and expensive to obtain and usually limited in coverage (Xie et al., 2015).

Historically, survey data has been combined with census data to generate spatial poverty maps (Hentschel, 2000). This allows calculation of poverty estimates on a sub-national level, however the lack of high-quality data leads to errors of high magnitude in the predictions. Olivia, Gibson, Smith, Rozelle, and Deng (2009) show how regression models for poverty estimation based on survey data can be negatively affected by spatial information. This could explain the large prediction errors. Such spatial error structures could introduce bias and could cause inference problems. In order to overcome this, Hentschel (2000) suggests overlaying other regional patterns such as road access with poverty maps to better understand correlations between such factors.

Henninger and Snel (2002) identified data availability and access as a major issue for map producers. This challenge is even more prominent when applying small area estimation techniques, where census data is combined with household surveys to generate high resolution poverty maps for areas of interest. Many governments hesitate providing independent agencies with sensitive data, and that for a good reason. Even when data is accessible, its quality is another frequent problem. Data sources can be unreliable or outdated to produce useful prediction models. Henninger and Snel (2002) came to the conclusion that there is a need for more sophisticated spatial analysis in the field of poverty mapping. Although techniques were not as advanced as today, poverty maps have been successfully used for developmental programs and by

policy makers in general. There is still need for more accurate maps and better tools and methods which is exemplified by the fact that of the 2015 millennium development goals, only 3.5 out of 18 targets were reached by the deadline (Ritchie & Roser, 2018).

Did we achieve the Millennium Development Goals (MDGs)?

Our World in Data

Summary of global progress of the United Nations' (UN) Millennium Development Goals (MDGs), which spanned the period 2000-2015. Shown are the Targets of the MDGs*, levels in the baseline year, the final target level and actual achieved level for each Target.

– Achieved Targets are marked in **green**;
– Missed Targets are marked in **red**.

Millennium Development Goal (MDG) Target	Baseline level	Target level	Achieved final level
MDG1.A: halve share of people living in extreme poverty (<$1.25 per day)	47% in developing regions	Reduce to 23.5%	Fell to 14%
MDG1.B: achieve full and productive employment, as well as decent work for all, including young people and women	62% global working-age population in employment	Full (100%)	Fell to 60%
MDG1.C: halve the proportion of individuals suffering from hunger	23.3% in developing regions	Reduce to 11.5%	Fell to 12.9%
MDG2.A: ensure that children universally – including both boys and girls – will be able to complete a full course of primary education	83% in developing regions	Universal (100%)	Increased to 91%
MDG3.A: eliminate gender disparity at all education levels	Developing regions: 0.87 in primary 0.77 in secondary 0.71 in tertiary	Gender parity index (GPI) between 0.97-1.03	Developing regions: 0.98 in primary 0.98 in secondary 1.01 in tertiary
MDG4.A: reduce the under-five mortality rate by two-thirds	90 per 1,000 live births	Reduce to 30 per 1,000	Fell to 43 per 1,000
MDG5.A: reduce the maternal mortality ratio by 75 percent	380 per 100,000 births	Reduce to 95 per 100,000	Fell to 210 per 100,000
MDG5.B: achieve universal access to reproductive health. *Pregnant women receiving adequate antenatal care visits*	35% in developing regions	Universal (100%)	Increased to 52%
MDG5.B: achieve universal access to reproductive health. *Women aged 15 – 49 in marriage/union, using contraceptives*	55% in developing regions	Universal (100%)	Increased to 64%
MDG6.A: halt and have started to reverse the spread of HIV/AIDS	3.5M new cases per year	0 new cases	2.1M new cases per year
MDG6.B: achieve global access to treatment for HIV/AIDS for those who need it by 2010	3% of people with HIV	100% of people with HIV	23% of people with HIV (2010) 45% of people with HIV (2015)
MDG6.C: ceased & started reversal of incidence of malaria & TB. *Incidence of malaria*	158 new cases per 1,000 at risk	Fewer than 158 new cases per 1,000 at risk	Fell to 94 new cases per 1,000 at risk
MDG6.C: ceased & started reversal of incidence of malaria & TB. *Incidence of tuberculosis (TB)*	172 new cases per 100,000 people	Fewer than 172 new cases per 100,000 people	Fell to 142 new cases per 100,000 people
MDG7.A: integrate principles of sustainable development into country policies & reverse loss of environmental resources			Multiple metrics (nearly all deteriorating)
MDG7.B: reduce biodiversity loss, achieving, by 2010, a significant reduction in the rate of loss			Red List Index shows continued biodiversity loss
MDG7.C: halve the proportion of the population without sustainable access to *safe drinking water*	24% without access to improved water source	Reduce to 12% without access	Fell to 9% without access
MDG7.C: halve the proportion of the population without sustainable access to *sanitation*	46% without access to improved sanitation	Reduce to 23% without access	Fell to 32% without access

*MDG8 (Global Partnership) does not have easily quantifiable targets and is therefore not included.

Source: United Nations (UN), the MDG Report (2015) & MDG Monitor.

The data visualization is available at OurWorldinData.org. There you will find further data on this topic.

Licensed under CC-BY-SA by the authors Hannah Ritchie & Max Roser.

Figure 3. *The 18 targets for the millennium development goals (MDG, 2000-2015).*

Figure 3 summarizes the circumstances after the deadline has been reached. Without putting any blame, it is evident that most of the millennium development goals have not been achieved by the deadline. It is especially concerning that for some targets the development went against the desired trend, for example MDG 1.B.

Technological Advancement in Poverty Mapping

The biggest issue is the lack of reliable and diverse data which is required to plan and execute effective development strategies aiming to ease poverty. In many countries, poverty and welfare indicator maps are used on a regional or national level for policy makers (Akinyemi, 2010). Current maps are often a

result of census data, household surveys and information from geographical information systems (GIS) (Bedi et al., 2007). Including GIS data like land use, water access, geographical isolation and other factors help not only localizing poverty more accurately but also understand important relationships between those factors. The capture and maintenance of geospatial data is still a manual process at most mapping agencies. Therefore, a lot of them currently pursue integration of AI in order automate workflow and enhance their value proposition (Murray et al., 2020). Using AI for image analysis can not only speed up the workflow for new images, but it can also be employed to identify previously overlooked features from older data and include those in current products.

Remote-Sensing Data

It has been shown that remote-sensing data can be used as an additional data source for poverty maps (Xie et al., 2015). In recent years, this data has become more available for little to no money. However, according to Xie et al. (2015), those data sets are unstructured and therefore extracting useful insights is a task that hasn't been automated yet. It has been shown that such data sets can be analyzed by various models and methods to obtain useful information on various aspects of interest (Ma et al., 2019). Applying machine learning (ML) methods to analyze remote-sensing data has been routine for a couple of years; however, the data sets and the purpose of its analysis have substantially evolved. In earlier applications, one usually analyzed multi-spectral data in a simple, restricted ML model to estimate various features, such as land coverage, water access, etc. In those restricted ML models, it is common to split up the selection of features important for solving the problem, then applying an algorithm to those features, followed by post-processing to produce its final output.

Splitting up these tasks make the models more predictable, simple, and easier to understand and implement (Quinn et al., 2018).

Machine Learning Algorithms

Machine learning is a type of algorithm that is capable of learning by analyzing important features in data. A well-trained algorithm can then be used to make decisions and thereby solve real world problems (Goodfellow, Bengio, & Courville, 2016). The identification, selection and extraction of the features for the model are imperative as the quality of the features is the determining factor for accuracy and performance of the resulting model (Lussier, Thibault, Charron, Wallace, & Masson, 2020). If the previously mentioned steps of feature identification, selection and extraction are controlled by the programmer, the algorithm is known as a supervised ML algorithm (Lussier et al., 2020). Supervised learning is useful whenever the inputs and outputs of the model are well-known and controlled. Deep learning models are a solution to the mentioned limitations, as DL models are capable of extracting the relevant features autonomously. Generally speaking, DL is a learning algorithm based on artificial neural networks (ANNs) (Schmidhuber, 2015). Those algorithms transform input data in one or more layer to output data while learning higher-level features. Layers between input and output are known as "hidden" layers. When the algorithm is characterized by multiple hidden layers, the network is considered a "deep" neural network, hence "deep learning" (Litjens et al., 2017). These algorithms can identify complex relationships and causalities from inputs in nodes in those hidden layers. The increased performance and complexity of DL algorithms allow analysis of images (Krizhevsky, Sutskever, & Hinton, 2017) and other more complicated inputs. This evolution introduced DL to many fields of sciences and thus helps solve real world problems (Lussier et al., 2020).

Prior to DL, remote-sensing relied first on ANNs but then shifted its attention

to support vector machine (SVM), random forest (RF) and decision trees (Ma et al., 2019). Since 2014, DL algorithms are implemented more frequently in order to analyze various variables such as land use, land cover, land classification and others. Various DL algorithms have been implemented for remote-sensing. The most well-known is the convolutional neural network (CNN) which is well-suited for image analysis when pixels are arranged regularly. Additionally, to CNNs, stacked autoencoders and decoders (AE, AD), restricted Boltzmann machine (RBM) and deep belief networks (DBNs) and generative adversarial networks (GANs) are commonly used for analysis of remote-sensing data. It has been shown that such models are generally more powerful and accurate for analysis of remote-sensing data compared to supervised ML algorithms, however they also require a substantial amount of training data and computational power (Ma et al., 2019).

Machine Learning and Poverty Estimation

Jean et al. (2016), Xie et al. (2015) and Tingzon et al. (2019) have shown how effective combining machine learning and satellite imagery can be for wealth estimation. Additionally, Tingzon et al. (2019) found that their models performed similarly when comparing open-source data against proprietary data from other sources. Ayush, Uzkent, Burke, Lobell, and Ermon (2020) finds that using cheaper low resolution satellite imagery in order to find regions to use high resolution satellite images is overall a better approach for poverty mapping when using policy networks as ML models. This information is useful because using high resolution imagery for everything is very expensive and because of the variability in the data sets a lot of computational power is required to train such a model.

The availability of poverty data depends heavily on the financial power of a region or country (Xie et al., 2015). For example, there are certain countries that haven't taken a census in over 10 years. The problem that follows from

that is that high profile initiatives from the United Nations often rely on poverty rates, infant mortality rates and other statistics to assess the effectiveness of their actions. Poverty measures are often collected in small scale field surveys which can never catch the full extent of the problem. These circumstances cause the issue that training data for regions where the results have the largest impact is very scarce. remote-sensing data has become so accurate that the temporal and spatial resolution can provide an incredible amount of data useful for sustainable development. Xie et al. (2015) states that CNNs and or deep learning models have been used to extract those data directly from remote-sensing imagery; however, this is not possible due to the lack of poverty data.

Transfer Learning in Poverty Estimation

Noe (2019) suggests using parameters from pre-trained models as initial guesses when training data is scarce. The technique is known as transfer learning and requires a model that uses the same input data and predicts something at least correlated to the target variable, in this case poverty. However, this does not work all the time and resulting models can be prone to over-fitting or biased predictions, characteristics that complicate finding a solution for the final model. If it works, less computational time is required for training and transfer learning can also decrease the risk of over-fitting and therefore the resulting model could react less sensitive to unseen data Murray et al. (2020).

Xie et al. (2015) have shown that nighttime light intensity correlates well with poverty. Thus, the lack of training data can be overcome by implementing transfer learning. In order to do so, one first needs to train a model that predicts nightlight intensities based on the remote-sensing data used for poverty mapping. The features in the first model could include structures, buildings, and farmland, and are generally important factors for nighttime light intensity.

Xie et al. (2015) demonstrates that those features are very useful for poverty mapping as well and can be used like survey data collected in the field.

The use of Aggregated Data and Regression for Poverty Estimation

Zhao et al. (2019) have taken a step back and use aggregated data on a 10 km x 10 km grid in a random forest regression model to estimate poverty in those grid cells. As independent variables nighttime light data, satellite images to extract terrain features, land cover data, road maps as well as geographic information regarding the distance to division headquarters were used.

Methods

Recently, a large amount of high-quality remote-sensing data has become available on platforms like google earth engine (GEE) and open street maps (OSM). Previous work from Xie et al. (2015) has shown that pre-trained deep learning algorithms like VGG-16, ResNet 50, Inceptionv3 or EfficientNet can be used for image classification and ultimately for poverty estimation at the local level using satellite images. In this study, a different approach is investigated, namely using readily available geospatial data, and using regression approaches to find relationships to better explain poverty at different administrative levels. Thus, the approach is similar to Zhao et al. (2019)'s approach, but instead of pre-defined grids the goal is to aggregate statistics to geopolitical boundaries at the admin 1 (state / province) and admin 2 (county / district) level. In this work, all data was extracted from GEE and subsequently analyzed with python. Additionally, an increased set of features and a much larger area was analyzed in this work compared to the previous approach.

Data Set Description

In this work, the possibility of building a general framework that is capable of estimating poverty on a global scale but at sub-national resolution is explored. For that reason, it was important to verify that most data sets had data available for most countries of the world with a high enough resolution. Additionally, the poverty data used for training was available for 2015 and 2018, therefore requiring matching data from GEE to train the models for past estimates. Finally, the data should be recent so the model can be used for poverty estimation. Table 1 lists all data sets used in this work and their availability in time. For a detailed explanation of every variable please refer to appendix 1.

Table 1. *Overview of used data sets in this work.*

ID	Name	Availability
VIIRS	Visible Infrared Imaging Radiometer Suite (VIIRS) Day/Night Band (DNB)	Since 2014
POP	WorldPop Global Project Population Data: Estimated Residential Population per 100x100m Grid Square	Since 2000
POL	Sentinel-5P NRTI NO2: Near Real-Time Nitrogen Dioxide: As an estimate for Air Pollution	Since 2018
LCT	Copernicus Global Land Cover Layers: CGLS-LC100 collection 3 Tree	2015 - 2019
LCU	Copernicus Global Land Cover Layers: CGLS-LC100 collection 4 Urban	2015 - 2019
LCG	Copernicus Global Land Cover Layers: CGLS-LC100 collection 5 Grass	2015 - 2019
LCS	Copernicus Global Land Cover Layers: CGLS-LC100 collection 6 Shrubs	2015 - 2019
LCC	Copernicus Global Land Cover Layers: CGLS-LC100 collection 7 Crops	2015 - 2019
LCB	Copernicus Global Land Cover Layers: CGLS-LC100 collection 8 Bare	2015 - 2019
GHM	CSP gHM: Global Human Modification: As an Estimate for the Level of Development	2016
FEW	FLDAS: Famine Early Warning Systems Network (FEWS NET) Land Data Assimilation System	Since 1982
SMD	TerraClimate: Monthly Climate and Climatic Water Balance for Global Terrestrial Surfaces, University of Idaho	Since 1958
GFS	Global Friction Surface 2019: Travel Speed	2019

As can be observed in table 1, the data sets from GEE stem from various different data sources, meaning that the units and measurements were not uniform and had different scales, meanings, and dimensions. The data is generally made available as image collections, with the information stored in grids, which can be made visible in pixels. In addition to everything else, the resolution of the base data can also vary, e.g. 100 m grid vs 1 km grid. For the purpose of this work this does not pose an issue, because the pixel data is collected and aggregated to the size of the administrative regions during the exporting step. During export, the aggregated mean, maximum, minimum, and standard deviation of the data in the area is stored in a csv file. Those four sub-features for every variable were then used as possible model inputs.
The three poverty rates describing the percentage value of people living with less than $1.90, $3.20, or $5.50 per day in the area in question were used as target values. Therefore, the target values were limited between 0 and 1. To increase the amount of training data, poverty rates were aggregated from admin level 2 to admin level 1 using the population data from the feature export whenever possible. The same logic was later employed to evaluate the predictions on admin level 2 data, where little to no training data was available.

Model Evaluation and Data Structure

To build models that produce stable outputs there are usually multiple evaluation steps involved. In this case, two dedicated training sets with historical poverty data from 2015 and 2018 were available. Each of these training tables were further expanded to build training sets on administrative level 1 and 2 respectively, resulting in three tables per year. Additionally, there are three tables in the same format for 2019 which can be used for testing, while one assumption must hold. Namely, it is assumed that the poverty rate from 2018 to 2019 has not changed dramatically in the observed areas. Therefore, we can assume that the model evaluation can be made with historical poverty data from 2018, combined with data from 2019, where available. These tables for the year 2019 are exclusively used to test and evaluate the models. To identify over-fitting and bias it was necessary to split

Table 2. *Overview of Tables used for Training and Testing.*

Use	Year	Level
Training	2015	Mixed
		Admin 1
		Admin 2
	2018	Mixed
		Admin 1
		Admin 2
Testing	2019	Mixed
		Admin 1
		Admin 2

the training tables into a separate training and testing set to allow model evaluation at every step of the process. This is useful to identify problems but is not a solution. However, it is possible to generate one or multiple validation

sets as another subset of data which allows optimization of model hyperparameters and being able to identify models with high bias at the same time.

Figure 4. *Visualization of the Model Building Procedure. The Models are trained using cross validated scores, which decreases the likelihood of over-fitting. Parts of this figure were adapted from:* Scikit Learn - Cross-validation: evaluating estimator performance *(n.d.)*

Figure 4 shows the experimental design used for model training. The hyperparameters were only optimized once, namely in the initial training step. Cross-validated scores were used for training. This set of parameters is not necessarily the best set of parameters to fit the training set, but it performs best on the internal test sets, which is a good indicator that it will also performs well in the final test set.

The model specialization is a step of transfer learning because the parameters and hyperparameters from the optimized initial model are used as initial guesses for the specialized models. It was found that this not only produces more stable outputs, but also decreases the time required for parameter optimization during specialization.

Model Overview

The following chapters introduce the models used for poverty estimation in this analysis.

XGB Classifier

XGBoost is a scalable, end to end tree boosting system. It was developed by Chen and Guestrin (2016) and now widely used and adapted thanks to its state-of-the-art results on many machine learning challenges. The framework allows both, classification, and regression. The XGB classifier was used in this work to differentiate a priori whether an area is thought to be below or above the median poverty rate at $1.90 per day in Africa. For that, no optimization was performed, since the default parameters resulted in an accuracy of over 90% in the testing set.

Kernel Ridge Regression

Kernel ridge regression combines linear regression and l2 regularization with the kernel method. L2 regularization means that the regression function receives an additional penalty equal to the sum of squares of the magnitude of

coefficients. This would be the cost function for a linear ridge regression (l2 regularization) without the kernel method:

$$Cost = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda \sum_{j=0}^{M} W_j^2 \quad (1)$$

With N being the sample size, M the number of features, W the weights and lambda the regularization parameter. The additional cost function with the sum of squares of model weights forces the overall size of coefficients to shrink and large coefficients are penalized more than smaller ones (Pedregosa et al., 2011). In addition to ridge regression, this model also utilizes the kernel trick, which allows solving problems in a high-dimensional space by the use of kernel functions. The combination of the above techniques can produce very powerful and accurate models, but the chance of over-fitting is high if the regularization strength or the kernel function is not chosen appropriately (Theodoridis, 2020a). Therefore, the regularization strength λ as well as the possible kernel functions are optimized during a grid search for hyperparameter optimization. The following table summarizes the parameters optimized during the search:

Table 3. *Parameter Grid for Kernel Ridge Regression*

Parameter	Grid			
Kernel	Linear	Polynomial	Radial Basis Fxn (RBF)	Sigmoid
Regularization	Logspace: $10^{-3}, \ldots$ (N=14) $\ldots, 10^{3}$			
Gamma*	Logspace: $10^{-3}, \ldots$ (N=14) $\ldots, 10^{3}$			
Gamma Definition	None	Regularization param. for the Kernel Function		

The regularization strength λ defines how much the parameter size affects the cost function. This means that with $\lambda = 0$ the problem would be an ordinary least squares regression with the kernel method. A model like that would be prone to over-fit and the resulting output function is less smooth than with a higher λ. However, very high values for the regularization strength might just

lead to a model that predicts an average value instead of accurate estimates. The gamma value is ignored for a linear kernel. For the other kernels the value defaults to the inverse of the feature size. However, the exact definition varies depending on the chosen kernel function. In general, a high value of gamma usually leads to over-fitted models, because the increased model complexity allows more paths to error minimization, while very small values of gamma in turn generalize too much, meaning that those models are not very useful for accurate predictions either.

Bayesian Ridge Regression

Bayesian ridge regression assumes probabilistic distributions instead of point estimates as target values in the regression problem and employs l2 regularization at the same time. The L2 regularization has been explained in equation 1 and is equal to a regularization constant (λ) times the square of the magnitude of the coefficients. This is unfavorable for large coefficients and the coefficient size to shrink.

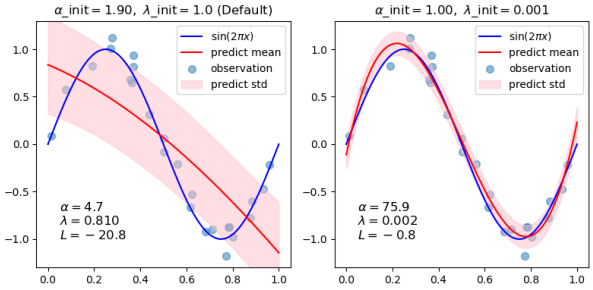


Figure 5. *Two examples of a Bayesian regression model output depending on chosen hyperparameters (*Curve Fitting with Bayesian Ridge Regression*, n.d.).*

Therefore, there is a trade-off between accuracy and generalization, with no regularization resulting in the same as a Bayesian regression model, and high regularization to a model that will tend to predict average values. In addition to the regularization, Bayesian ridge regression models assume a normal distribution of target values, thus they are more stable in regard to outliers and

ill posed problems (Theodoridis, 2020b).

Table 4. *Parameter Grid for Bayesian Ridge Regression*

Parameter	Grid
Tolerance	Logspace: $10^{-10}, \ldots$ (N=5) $\ldots$, 10^{-1}
Alpha 1	Logspace: $10^{-10}, \ldots$ (N=5) $\ldots$, 10^{3}
Alpha 2	Logspace: $10^{-10}, \ldots$ (N=5) $\ldots$, 10^{3}
Lambda 1	Logspace: $10^{-10}, \ldots$ (N=5) $\ldots$, 10^{3}
Lambda 2	Logspace: $10^{-10}, \ldots$ (N=5) $\ldots$, 10^{3}

As seen in figure 5, the selection of hyperparameters has a great effect on model performance, accuracy, and its ability to generalize. To avoid over-generalization while ensuring the ability to generalize, the optimal values for the hyperparameters are determined in the grid presented in table 4.
The tolerance defines the stopping criterion; thus the algorithm will stop if the error is smaller than the tolerance. The remaining four values are defaulted to $10^{-}6$ if none are provided. Alpha 1 and Lambda 1 are shape parameters for the Gamma distribution, while alpha 2 and lambda 2 are rate parameters.

Support Vector Regression

Support vector regression (SVR) is based on support vector machines (SVM). SVMs use the kernel trick, similar to kernel ridge regression, so highly nonlinear classification problems can be solved in a higher dimension than the original problem. The same principle can also be used for regression, but instead of fitting a separation curve between two classes, the SVR fits a curve to target values (Pisner & Schnyer, 2020).

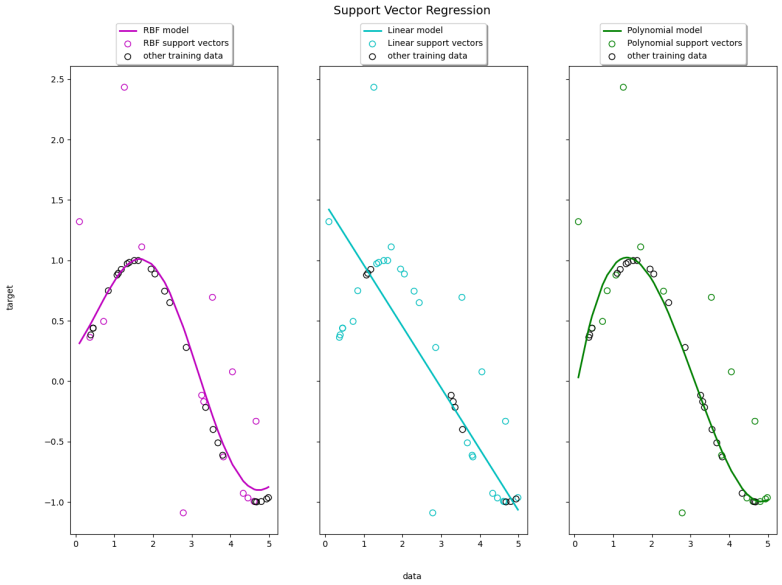


Figure 6. *Three examples of a support vector regression model output depending on chosen kernel functions. (*Support Vector Regression (SVR) using linear and non-linear kernels*, n.d.)*

As seen in figure 6, the correct choice of the kernel function and other hyperparameters has a significant effect on model performance, especially if the chosen model is incompatible with our data. To limit bias and the possibility of over-fits, all major parameters for the SVR model were optimized in a grid search using cross validated scores. The parameter grid is summarized below:

Table 5. *Parameter Grid for Support Vector Regression*

Parameter	Grid			
Kernel	Linear	Polynomial	Radial Basis Function (RBF)	Sigmoid
Gamma	Scale	Auto	Logspace: $10^{-3}, \ldots$ (N=5) $\ldots$, 10^{3}	
Regularization	Logspace: $10^{-2}, \ldots$ (N=12) $\ldots$, 10^{4}			

The four kernel functions are identical to those that were tested for the kernel ridge regression, gamma is an important parameter for the kernel function and the regularization strength reduces the parameter size.

XGB Regression

The XGB-regressor was optimized in a grid search to produce stable outputs and avoid bias. The following parameters were adjusted:

Table 6. *Parameter Grid for the XGB Regressor.*

Parameter	Grid			
Eta	0.01	0.05	0.1	0.5
Gamma	0	Logspace: $10^{-3}, \ldots$ (N=6) $\ldots, 10^{3}$		
Child Weight (min)	1, 3, 5, 7, 10			
Max Depth (Tree)	2	3	4	5

The eta parameter is a step size shrinkage operator that shrinks the feature weights after every boosting step. An ideal value of eta limits over-fits and leads to more stable estimations. Gamma is the minimum loss reduction required to add another leaf to the tree. This means that with large gamma, the algorithm is more conservative. The minimum child weight determines sum of (hessian) instance weights needed to partition the tree. Thus, with a high child weight, the algorithm is more conservative. The parameters gamma and minimum child weight both have the purpose to limit over-fitting by increasing generalization. The last parameter, the max depth, determines the depth of a tree. With increasing depth, the model can learn more and more detail but becomes very complicated quickly, which also increases the likelihood of over-fitting.

Artificial Neural Net Regression

In deep learning, artificial neural nets (ANN) are commonly used to tackle a wide variety of problems. Artificial neural nets are just that, they mimic our biological way of learning and connecting information to draw conclusions. They belong in the field of deep learning, because the extracted features and

relationships are not defined by the scientist, but developed by the algorithm itself.

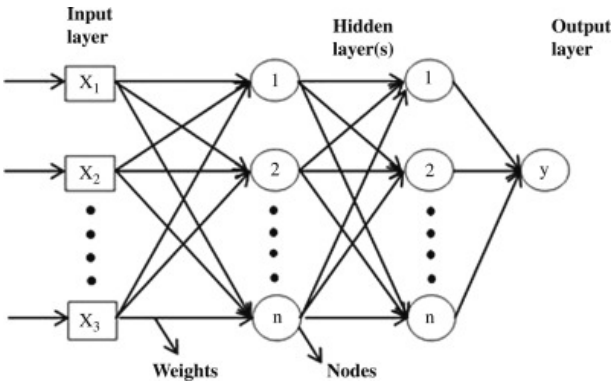


Figure 7. *Visualization of a generic Artificial Neural Net (ANN). The algorithm learns how to manipulate the information in the input layer by adjusting weights and activation functions until a stable output is produced. (Sairamya, Susmitha, Thomas George, & Subathra, 2019)*

Neural nets have obvious advantages since they can find features in the hidden layers that the scientist does not even know about. Nonetheless, this is also the algorithms biggest weakness, because it is difficult to follow a neural net's logic and calculate important parameters like feature importance or contribution to an output. ANNs are often described as black box models, where only the input and output are well-known but it is not easy explain why a neural net produces certain outputs. However, nowadays there are advanced analytical tools that use different techniques to estimate feature contribution and importance in neural nets (Lundberg & Lee, 2017). With the additional liberties in regard to model structure there is also an increased risk of bias and over-fitting. For that reason, a model building function was written which was able to adjust model hyperparameters without human input. The parameters were again optimized in a grid search, while conditional parameters were only assigned when necessary.

Table 7. *Possible Inputs for ANN building function*

Parameter	Grid				
Number of Layers	2	3	4	5	6
Regularize Layer		Yes		No	
Regularization Type	L1		L2	L1_L2	
Add Dropout		Yes		No	
Dropout Rate	0.01	0.02	...	0.19	0.2
Activation Function		Relu , Selu , Sigmoid , Tanh			

As presented in table 7, some parameters, like the dropout rate and the regularization type are only used conditionally. This makes a grid search more complicated, which is why a model building function was developed with the kerastuner package from (O'Malley et al., 2019). This allows a similar procedure to a grid search, where the best combination of model hyper-parameters can be identified by using cross validated scores, but conditional parameters and other more advanced functionalities are made possible.

Ensemble Methods

Ensemble methods describe models that use multiple independent sources of information to produce a final output. XGBoost is an example, as the algorithm relies on decision trees and regression methods and averages multiple sources of information to produce the final output. There are many other algorithms that are readily available which are based on ensemble methods. It has been shown that they generally produce more accurate results than a single model would, which is why there is so much effort going into improvement, development, and integration into well-known frameworks like Scikit-Learn from Pedregosa et al. (2011) and XGB from Chen and Guestrin (2016). For that reason, it was decided to not only include ensemble methods in the

analysis, but also see if it is possible to build a stacked ensemble model using the outputs from the other models used in this analysis. As inputs, the estimated poverty rates from the 5 models that were previously introduced were used.

Model Evaluation

The models were evaluated based on two scores, namely the R^2 value of prediction and the mean normalized absolute relative percentage error (mabsRPD). The R^2 value provides information on how well the model follows the trend in the target data, whereas the mabsRPD provides information about the error to the target value. The combination of the two metrics allows us to draw well founded conclusions about the models and their accuracy. In sociological models it is common to see lower R^2 values compared to scientific applications. Similar applications for poverty mapping from Zhao et al. (2019) on a 10 x 10 km grid in Bangladesh and Nepal resulted in R^2 0.70 and 0.61 respectively. To avoid over-fitting, which ultimately leads to sensitive model parameters that generally don't perform well when used to predict data that the model has not previously been trained with (Twin, 2021), all training steps were immediately followed by testing and analysis to ensure a similar performance for training and prediction. Furthermore, validation scores were the basis for model training whenever applicable.

Data Preprocessing

Since the feature data is composed of values from varying sources with different meanings and units, it was evident that a thorough analysis was necessary before any good model could be built. As can observed in the table 2, there are three tables with different data structures for every year. The mixed tables have been obtained through World Bank and contain a mixture of data at the country (admin 0), state (admin 1) as well as at the county or provincial level (admin 2). The other two tables for each year were obtained through google earth engine (GEE) and were produced by the Food and Agriculture

Organization of the United Nations (FAO). Because of the hierarchical nature of administrative regions, it is clear that with a higher administrative level the areas become smaller. Therefore, it was important to analyze the dependence of the features on the administrative level but also depending on time. The complete analysis can be found in appendix 2 and appendix 3.

By analysis of those tables, it became evident that the data varies only a little in time, but much in regard to the administrative level used for data export. This makes sense because of the previous stated fact regarding the average size of the areas used to aggregate the statistics. Nonetheless, this complicates preprocessing because the high variance in the data made normalization and most other non-linear transformations unfeasible, because the learned parameters in the training set would not produce the desired output in the testing set. For that reason, it was decided to limit the transformations to exclusively linear operations in preprocessing, which ultimately produced stable and reproducible outputs useful for machine learning models. For scaling, a minimum-maximum-scaler was chosen, which limits the range of all feature values between 0 and 1, based on the minimum and maximum values for every feature. These values are fairly constant year over year, so the learned parameters for scaling don't need to be updated frequently, which allowed training the parameters on the training set and using them without any adjustments in the test set. Another factor that was taken into consideration was the distribution of target values in the data. As presented in figure 8, the poverty rate at the daily income level of $1.90 is almost

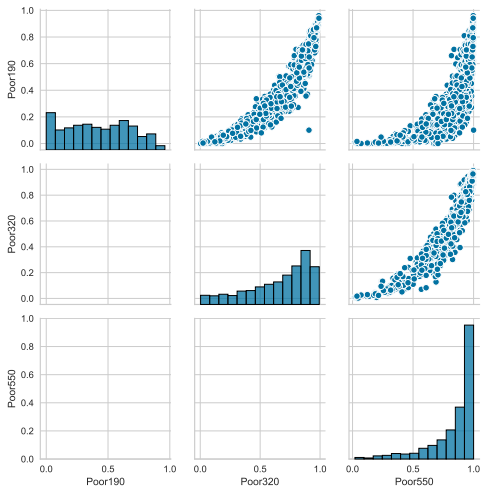


Figure 8. *Distribution of Target Values depending on Poverty Level.*

uniform in Africa, while the rate at the daily income level of $3.20 is already slightly skewed to the left, and the distribution of poverty rates at $5.50 per day is skewed heavily. These circumstances mean that it is simple for a model to produce results with small errors at the $5.50 daily income level, but the accuracy of these estimations must be questioned since the possibility of an over-fit is greater compared to poverty estimations at the lower income levels. In fact, if accurate estimates at the $5.50 level are desired, it would be best to explore whether box cox transformed, or possibly logarithmic target values are better suited for poverty estimation at that respective income level. For the purpose of this study, this has not been further investigated.
After scaling, correlated features were identified using the Spearman rank correlation method. While the Pearson correlation method requires; the data to be continuous, the presence of related pairs, the absence of outliers, normality of variables, linearity, and homoscedasticity, the Spearman correlation only requires the data to not be nominal and measures the monotonic correlation instead of the linear one.

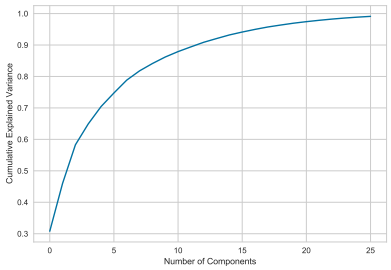
Feature Selection

To increase model efficiency and performance and decrease the likelihood of over-fitting and high bias, a lot of effort was put into the selection of features for the estimation models. Initially, the plan was to use the first N principle components explaining 95% of the variance in the data.

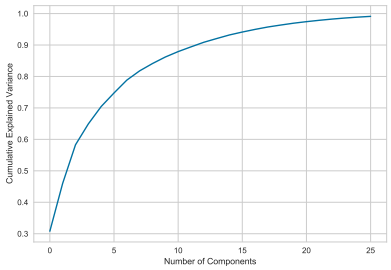
Principle Component Analysis

It was found that the PCA transformation is quite sensitive to changes in data year over year. Additionally, the PCA changes depending on administrative level, which again makes this approach unfeasible. Nonetheless, the PCA was still utilized to estimate the quantity of features required to explain majority of the variance in the data. This value ranges from 20 to 25, depending on the administrative level of the data as well as the current year of observance.

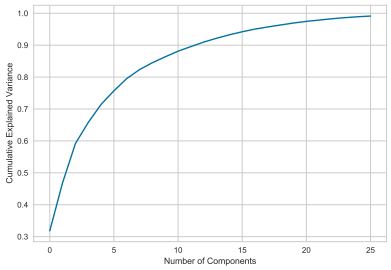
(a) *2015*



(b) *2018*



(c) *2019*



(d) *Admin 1*

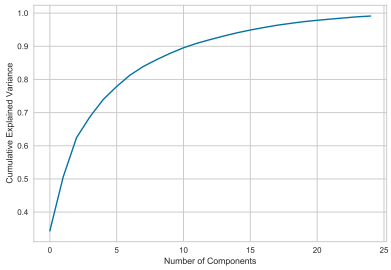


Figure 9. *PCA analysis for a selection of used data sets in this work. a) and b) correspond to training tables obtained from World Bank. c) is the same transformation for the test set. d) shows the PCA for the Admin 1 level, with a steeper shape compared to the other three examples.*

In the figures above, the cumulative explained variance depending on the number of principle components used in the analysis is presented. It was found that there is no need to include all 52 independent variables, half of them explain most of the variability in the test set, if they were all linearly independent.

Although the PCA is a useful tool to identify the amount of necessary features to explain the data, in this case it can't be used for feature selection because the transformation reacts sensitive to changes in administrative levels, meaning a learned PCA won't perform well on the test set. Nonetheless, the main takeaway from this analysis was still useful, namely that most of the data could be described by about 25 linearly independent variables. Therefore, a select few variables were removed from the data set because they showed little to no

variation across all the data. Those variables were land-cover minima and maxima values in the areas of interest, which were almost always equal to 0 and 1 respectively. This information is not useful to any model, which is why it is important to remove such data prior to training of machine learning models to reduce the number of parameters.

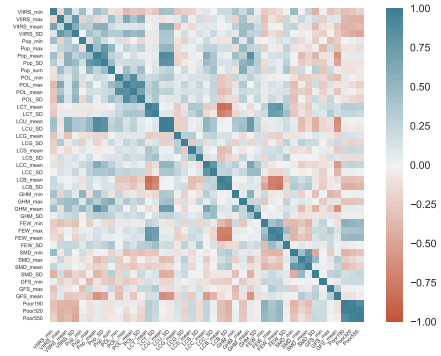
Recursive and Parallel Feature Selection

The removal of the land-cover maxima and minima values leaves us with 41 possible features as input values for the machine learning models. Ideally, the feature selection is run in parallel with model training and hyperparameter tuning. This is very troublesome to code, difficult to generalize and requires a lot of computational power and or time. The fact that the model frameworks have been obtained by three different sources meant that it was not feasible to write a function that performs recursive or parallel feature reduction while training. Instead, two sources of information were combined to perform the feature selection. First, a Spearman rank correlation analysis was used to identify strongly correlated features. The Spearman rank method was chosen because it does not require the data to be normal, as it uses bins to determine the level of correlation between two variables.

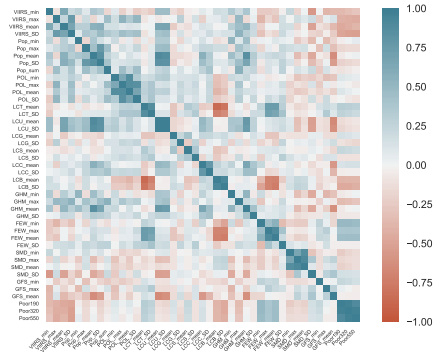
Correlation Analysis

The analysis for all data sets shown below are nearly identical. Notably, there is a strong correlation between the last three variables on the bottom right of the plot, which are the poverty rates used for training, therefore a strong correlation was expected and observed in figure 8.

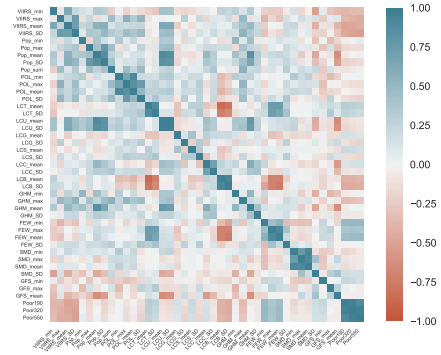
(a) *2015*



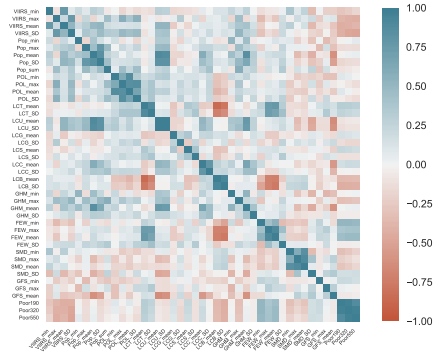
(b) *2018*



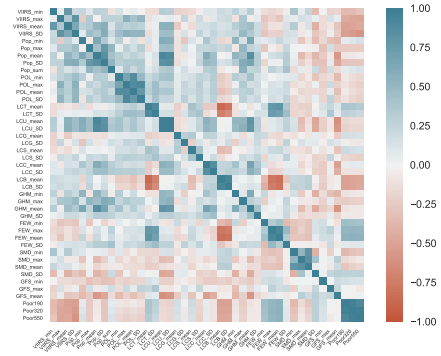
(c) *2019*



(d) *Training Set Combined*



(e) *Admin 1*



(f) *Admin 2*

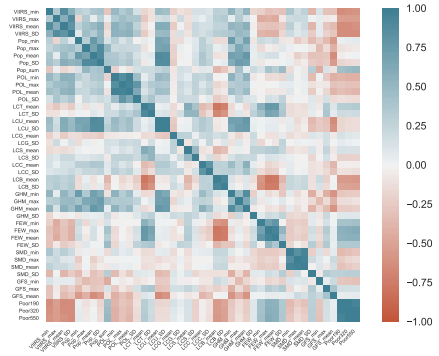


Figure 10. *As we can see in the figures above, the correlation does not depend on the administrative level.*

Other strong interactions have been identified between the urban land cover and the population, and between the brush and tree landcover. While these stronger interactions seem intuitive, there are some interactions that are difficult to explain, like a positive correlation between the tree cover and the famine early warning system. The complete discussion of this analysis can be found in

appendix 4, where full size images of the figures above are also presented.
The second information used for feature selection was an initial model using all 41 possible features as model inputs. It was found that the most important features in that model correlate well with a suggested subset of features based on the Spearman rank method. Therefore, the correlation analysis was used to select a subset of not strongly correlated features, with the limitation that all mean values are included in the set. This procedure resulted in 29 features which were later used as model inputs and are summarized in the table below.

Table 8. *Summary of all Features used as possible model inputs.*

Selected Features			
VIIRS av.	VIIRS minimum	LC Crops av.	LC Barren std. dev.
Human mod. av.	Human mod. min.	LC Barren av.	Famine std. dev.
Polution av.	Polution min.	Population av.	Soil Moisture std. dev.
LC Tree av.	Famine min.	Famine av.	VIIRS max.
LC Urban av.	Soil Moisture min.	Soil Moisture av.	Polution max.
LC Grass av.	VIIRS std. dev.	Travel Speed av.	Famine max.
LC Shrubs av.	LC Tree std. dev.	Travel Speed min.	Soil Moisture max.

Classification

After some initial model testing, it was found that it is difficult to build a model that accurately identifies and estimates poverty in "poor" areas and "rich" areas at the same time. Therefore, it was decided to include a classification algorithm in the preprocessing, which simply estimates whether the location is thought to be in either of the two areas. As a threshold, the median poverty rate in the training set at the poverty rate of $1.90 a day was chosen, which is about 0.3.
As a classification algorithm, the powerful XGBoost classifier from Chen and Guestrin (2016) was selected for its great versatility and stability. The

algorithm is based on an ensemble model of boosted trees. The information from this preprocessing step is later used to build specialized models receiving either areas that are thought to show high poverty rates or low poverty rates respectively. This is similar to using a discrete input in a regression model, but allows more flexibility, if there are issues with just a subset of models. In the following chapter the results from this procedure are summarized.

Solution Evaluation

Metrics

Twelve different models were developed in the course of this work. To allow a meaningful comparison, multiple metrics of evaluation where chosen. It was important to consider the relative range of the poverty rate depending on the level under investigation as well as the distribution of the poverty rates. Since the Area under investigation is continental Africa, a majority of Areas suffer from high poverty rates. This is especially true if the threshold for poverty is set high (like for the $5.50 per day poverty level). It follows that the likelihood of over-fitting depends heavily on the poverty level under investigation.

R^2-Value

The first metric used to compare the models and their accuracy was the coefficient of determination, or the R^2 value. Contrary to multiple misleading definitions, R^2 is not restricted between 0 and 1, but instead has a theoretical range from negative ∞ to its upper limit 1. Looking at the formula for the coefficient of determination it becomes evident that it is in fact not restricted by zero:

$$R^2 = 1 - \frac{SSE}{SST} \tag{2}$$

The term on the right in equation 2 is always positive, and only limited by SSE, since SST is given from the data. Therefore, for bad models R^2 can indeed be smaller than zero.

Mean Absolute Percentage Difference

As previously mentioned, the second evaluation metric was the mean normalized absolute relative percentage difference (mapsRPD). It was found that the relative score metric was better suited for comparison of model

performance at the three income thresholds because of their different distribution in the area of interest.

Distribution of Residuals

The last factor taken into consideration for model evaluation was the distribution of residuals. In most classical regression models, residuals must follow a normal distribution and follow an equal trend of variance (homoscedasticity) for mathematical reasons. This allows quite accurate estimation of the prediction error in the range of model training. However, these assumptions pose serious limitations for a lot of applications today. For a large sample size, even slight deviations from a Gaussian distribution means that there is a statistically significant deviation which will result in a failed normality test. Contrary to classic regression models, the frameworks used in this analysis do not rely on the assumption that the residuals are normally distributed for the calculation of their parameters. Nonetheless, residual analysis allows us to draw further conclusions about the accuracy, performance, and reliability of the models. Therefore, a normal distribution of residuals is still desirable and an indicator for a well-suited model for the problem at hand.

Training Results

The following sections summarize the results obtained during model training. First, the classifier is analyzed, followed by the initial base training results. Finally, the transfer learning results as well as the final testing data is presented.

Poor - Not Poor Classifier

The XGB classifier used as part of the preprocessing was not optimized in a grid search. Therefore, it is possible that more accurate classification results could be obtained by optimizing the hyperparameters of this model. However, the accuracy in the testing and training set were satisfactory for the purpose of this study. Additionally, the regions of interest in Africa remain the same, thus

one could always use historical poverty estimates instead of a classifier for this step. However, the purpose of this work was to estimate poverty without any historical poverty data, which is why this framework was used. The XGB model was trained on 80% of the data from 2015 and 2018 and tested on the remaining 20%. The average accuracy score during training was 92% (1%), and during testing the model predictions resulted in accuracy scores of 87% (1%). The results were obtained by using 5-fold cross validated testing scores which allows the additional calculation of an approximated standard deviation of the accuracy metric.

The small drop in prediction accuracy of the classifier in the testing set compared to the training set indicates a minor overfit. However, in this case this is not further concerning, since the accuracy in the testing set at 87% with 1% standard deviation indicates a good model fit, nonetheless. Additionally, the feature importance was analyzed, which can be seen in figure 11a.

(a) *Feature Importance* (b) *Feature Contribution*

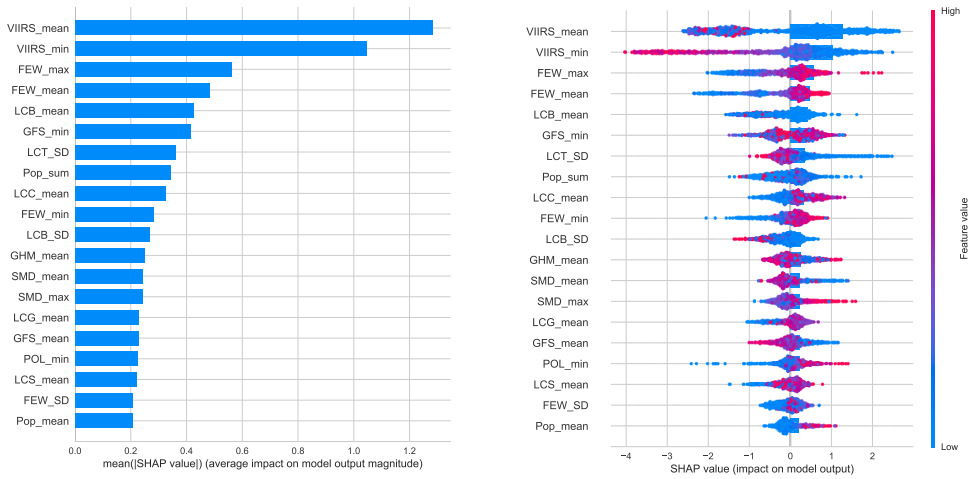


Figure 11. *Feature importance and contribution for the XGB Classifier used in preprocessing.*

The figure shows that the importance for the nighttime lights (VIIRS mean and minimum) are quite high compared to the other variables. This was expected, since Xie et al. (2015) has found that nighttime lights correlate well with

poverty. For further analysis, shapley values were used to analyze feature contribution in further detail (Lundberg & Lee, 2017).

Figure 11b shows the impact on model output depending on the value of the most important features. This so called beeswarm plot helps us better understand the estimated poverty rates, since the contribution of the most important features are visualized. Figure 11b shows that the feature values for the nighttime lights mean, and minima contribute to the model output in a similar way. For small feature values, the model usually produces high poverty estimates. This seems intuitive, since we have seen in figure 10 that the nighttime lights and poverty are negatively correlated. However, there are additional conclusions that we can draw from the analysis of the feature contribution. The first conclusion is that the relative size of the feature doesn't always correlate with the model output. A good example for this is the minimum value for the global friction surface (GFS_{min}), for which high and low feature values can have both, a positive and negative impact on the model output.

Initial Training

The test set results after initial base model training are presented in this section. This means, that these models are estimating poverty on various admin levels and are not meant for final poverty estimation. Instead, these models form the basis for the later models, where the model parameters are optimized in order produce stable and reasonable results on a certain administrative level. First, the R^2 values and mean normalized absolute relative percentage difference (mabsRPD) were calculated. The following figure shows the resulting values for all models built in this step. The numeric results used to produce figure 12 can also be found in appendix 5, table 16.

Three different types of models were built for this study. First, a set of "direct" models, which do not use the information from the classification in the preprocessing. This means that those models have the largest training set size,

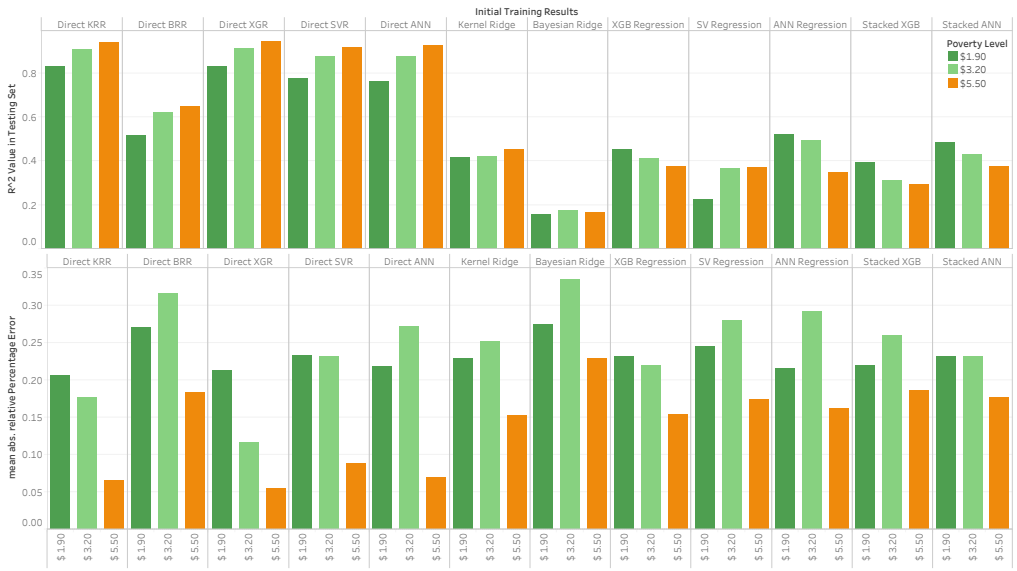


Figure 12. *Summary of Test Set Results after Initial Training. Top: Bar Graph of all R^2 Values. Bottom: Bar Graph of Corresponding mean absolute relative percentage Error.*

but it was found that they don't perform well for data in other years. The second type of model uses the information from the classifier and therefore the training set is about 50% compared to the first model type. The last models built were stacked ensemble models.

They use the estimated poverty values from the 5 distinct models of the second type as input and don't rely on the features as inputs, which means they have again about 50% of the training samples compared to the first model type, but they have less input features as well. It was found that those stacked models perform quite well for unseen data. From figure 12 we can conclude that Bayesian Ridge Regression (BRR) performs worse compared to all other models under investigation. Additionally, it was observed that the other models KRR, SVR, XGR and the ANN perform similar in their respective category, meaning that the performance of the models is similar for the same data. Looking at the mean absolute relative percentage error (mabsRPD) a trend following to the poverty level under investigation was found. This makes sense because of the distribution of poverty rates depending on the level which is

shown in figure 8. Those distributions have the effect that the relative error for a poverty rate at a lower income is bigger than the relative error for a poverty rate at a higher income for the same distribution of absolute errors. Therefore, the observed trend in the mean absolute relative percentage difference is not further concerning. Another trend is observed regarding the R^2 values. While R^2 for direct model increases with a higher income threshold for poverty classification, there is no clear trend like that for the remaining model frameworks. Although direct models seem to have an overall higher R^2 value compared to the other model types, the mabsRPD is not significantly smaller than for the other models. The main takeaway from figure 12 are that direct models with the largest training set size perform better compared to ensemble methods on a test set for the same year and that BRR is performing worse compared to the other models.

Admin 1 Specialization and Transfer Learning

The resulting models from the initial training were not ultimately used to estimate poverty for future years of interest. Instead, they were used as an initial guess to build specialized models which are only compatible with data from a certain administrative level. This makes the training data more uniform and ideally allows the model to find a better set of parameters for poverty estimation on that income and administrative level. In this chapter, we are analyzing the results for the admin level 1 specialization, which is optimized to estimate poverty at the first level of local administration.

Looking at figure 13 different trends compared to figure 12 are observed. Contrary to before, the direct models do not clearly outperform the other models, although they have the largest training set available. The stacked and direct ANN, the XGB Regression and the Kernel Ridge regression show comparably high R^2 values and comparably low mapsRPD which indicates a good model performance and reliability. Interestingly, the direct models for the admin level 1 estimations have lower average R^2 values and higher average

Figure 13. *Summary of Test Set Results after Transfer Learning from Initial Training to Admin Level 1. Top: Bar Graph of all R^2 Values. Bottom: Bar Graph of corresponding mean absolute relative percentage error.*

mabsPRD values compared to the results on the mixed data. Similar to before, the BRR model does not seem to be adequate for this type of data. On the other hand, the kernel ridge regression using classification information as well as the XGB model and the stacked models performed quite well.

Admin 2 Specialization and Transfer Learning

After building models for poverty estimation at the admin 1 level it was decided to try a similar procedure for the admin 2 level. It was expected that these models would have lower accuracy and higher errors compared to before, which is mainly due to the lack of available training data and the fact that the average size of a region is now much smaller compared to the initial training data.

In figure 14 a similar situation to the one in figure 12 was observed. The direct models apart from the BRR algorithm outperform all others, but the trends in R^2 values indicate a lot of bias for poverty estimates at high income levels. This means that future estimations with these models could be unstable or even unusable. To test that, we look at the resulting poverty rates for data obtained

Figure 14. *Summary of Test Set Results after Transfer Learning from Initial Training to Admin Level 2. Top: Bar Graph of all R^2 Values. Bottom: Bar Graph of corresponding mean absolute relative percentage error.*

from 2019, while it was assumed that the poverty rate has not changed from 2018 to 2019. This allows analysis evaluation of the models identical to the training steps.

Testing Results

The previous chapter summarized testing set results during training, this chapter summarizes results for the year 2019. This data is used exclusively to estimate poverty for testing. The evaluation was done assuming that poverty rates have not changed from 2018 to 2019.

World Bank Format 2019

The first testing set has an identical format to the table obtained by World Bank for the year 2018 which was used in training. However, the feature values in this set correspond to the year 2019, meaning the models have not yet been exposed to this data.

As seen in figure 15, the overall model performance is quite high for all models except for algorithms using Bayesian ridge regression. Interestingly, the R^2

Figure 15. *Results for table in mixed format using initially trained model with data from 2019. Top: Bar Graph of all R^2 Values. Bottom: Bar Graph of corresponding mean absolute relative percentage error.*

values for the direct models are similar to figure 12, while the other models actually perform better than in the testing set used during training. This can be explained by the similar data structure in this feature set, since it uses the exact format to 2018. For some models, like neural networks and boosted trees, eliminating over-fitting is basically impossible. Instead, the cross validation finds a model that is usually generalizing well but fits the training data still better than the testing set. This can explain the unexpectedly high performance of all models on this data set.

Admin 1 Format 2019

This set of data contains feature values corresponding to all administrative regions on the first admin level in Africa for the year 2019. The data was evaluated by comparing it to values from 2018, where data was aggregated when necessary. The aggregation was done according to the following formula,

$$AggregatedRate = \sum_{i=0}^{N} \frac{Rate_i * PopulationSum_i}{\sum_{i=0}^{N} PopulationSum_i}, \tag{3}$$

where N is the amount of subregions for which a poverty rate is available, i is a running index, the Rate_i is the estimated poverty rate from the model, and the PopulationSum_i is the sum of population obtained from GEE, which relies on estimated population densities in a 100 x 100m square grid. The use of these population sums is not optimal, as it was found that the accuracy of the data does not correspond 100% with population data from census data. This adds another possible source of error and should definitely be considered in future work.

For these models a similar performance compared to the mixed models was expected. The World Bank has put in a lot of effort to provide high resolution poverty maps, which is why there is an increasing amount of admin level 1 and 2 poverty estimates in every new publication. Considering the input data analysis from appendix 2, 3, and 4, we can observe that the data for administrative level 1 only differs a little bit from the mixed data used for initial model training. This indicates that the "average sample" is not much different to before, which should allow quite accurate poverty estimations for this data.

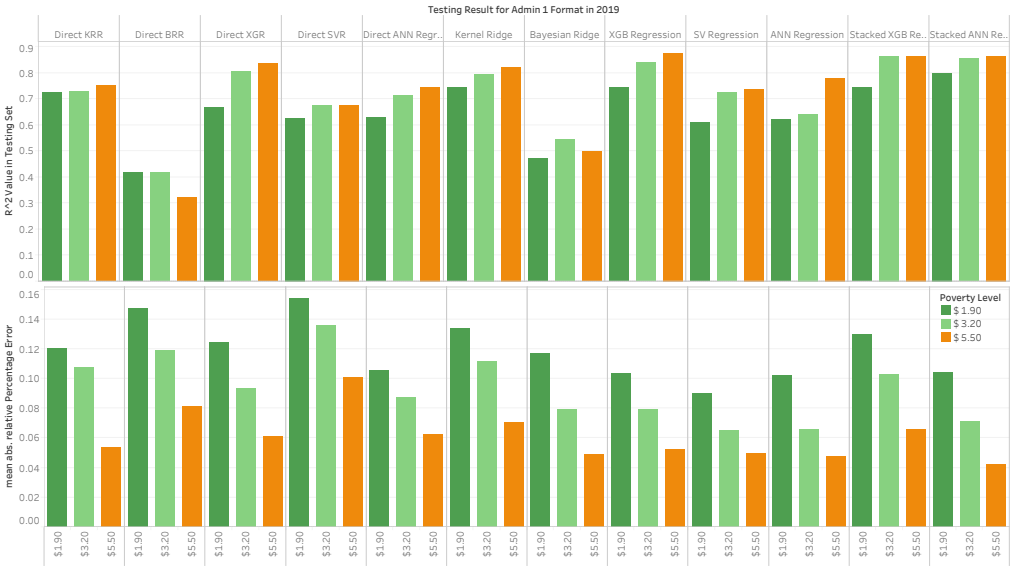


Figure 16. *Results for admin level 1 specialized models for 2019 data. Top: Bar Graph of all R^2 Values. Bottom: Bar Graph of corresponding mean absolute relative percentage error.*

As seen in figure 16, the situation is a little different compared to figure 15. The direct models are not as accurate as for the mixed data, whereas the models using the information from the classifier perform better, as well as the stacked models which use poverty estimates from the latter models to build their final prediction. Similar to all previous cases, the Bayesian ridge model does not perform as well as the other models, indicating that linear models are simply not able to capture the complex relationships required for poverty estimation without any poverty data. Another detail to point out is the R^2 value for the stacked models. Similar to figure 15, the stacked models have higher R^2 values than any of the models it relies upon for its predictions. This is a testament to the approach of ensemble models, since these stacked models are very basic examples thereof. There are well developed ensemble methods available that are much more stable, better integrated, and more sophisticated than the two examples used in this study. The fact that developed ensemble methods outperformed the other models anyway underlines the power and versatility of these approaches.

Admin 2 Format 2019

This testing set contains exclusively data for regions on the second administrative level. Most of the training data was available for the first administrative level, with some data only available on the national level. This means that there was little to no training data for data on the admin 2 available. Additionally, considering the information from the input data analysis in appendix 3, it becomes evident that the administrative level has a significant impact on the average value of a feature.

Figure 17 shows that the overall accuracy is not as high compared the previous two cases. For the direct Bayesian ridge model, the R^2 values were negative. This means that predicting the average poverty rate would be a more accurate estimation model and underlines earlier conclusions that Bayesian ridge regression is not suited for this kind of regression problem. On the other hand,

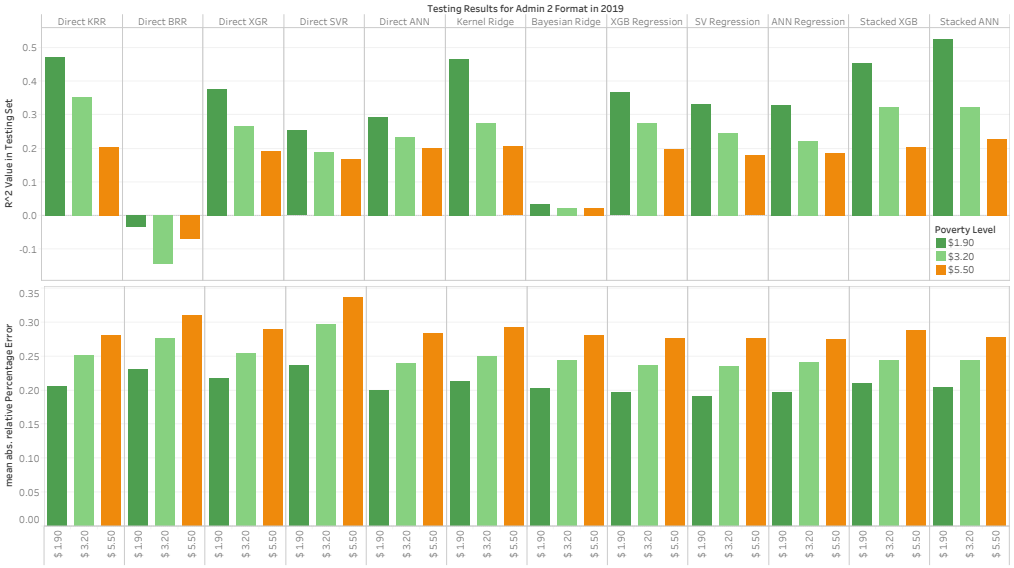


Figure 17. *Results for admin level 2 specialized models for 2019 data. Top: Bar Graph of all R^2 Values. Bottom: Bar Graph of corresponding mean absolute relative percentage error.*

the ensemble methods as well as the kernel ridge regression still have quite high R^2 values, but their mabsRPD are also large. Contrary to the previous cases, the errors now follow an opposed trend, with the largest errors for the poverty rate at a daily income of $5.50, which could be caused by the skewed distribution of those target values which was shown in figure 8.

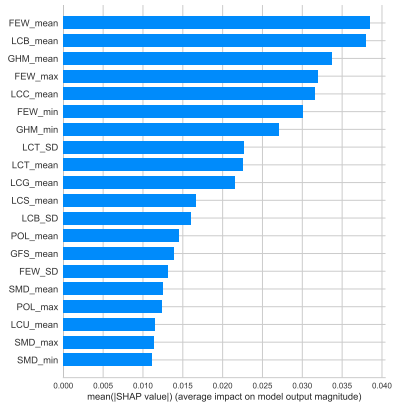
Optimized Parameters and Feature Importance

This chapter summarizes the optimized model parameters and structures which were obtained in an automated grid search in the training step. Additionally, the individual feature importance as well as the contribution to the output value are visualized using the python module shap. The feature importance in shap is calculated by analyzing the change of model output if a feature input is missing and are presented as shapley values (Lundberg & Lee, 2017).

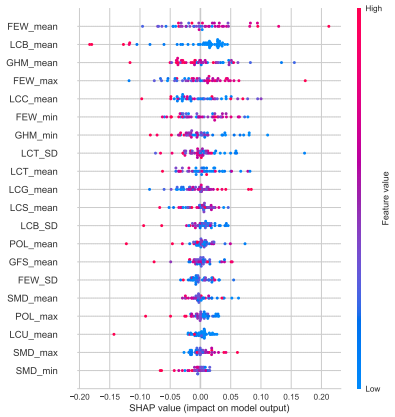
Kernel Ridge Regression

The first model under investigation is the kernel ridge regression. As observed in figure 18 in the right column, the kernel functions allow this regression model to develop nonlinear interactions to produce the final estimation value.

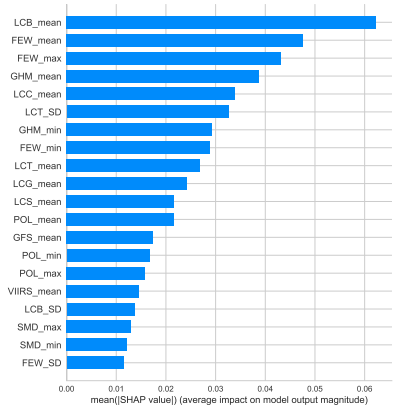
(a) *Feature Importance at $1.90*



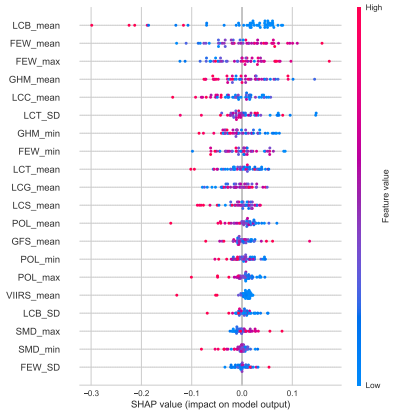
(b) *Feature Contribution at $1.90*



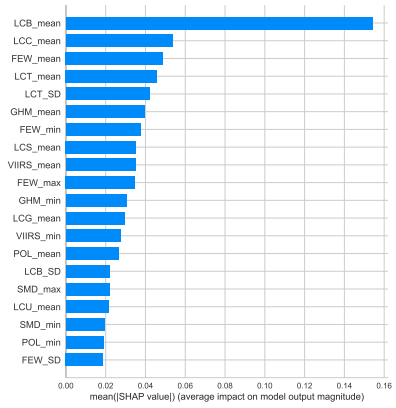
(c) *Feature Importance at $3.20*



(d) *Feature Contribution at $3.20*



(e) *Feature Importance at $5.50*



(f) *Feature Contribution at $5.50*

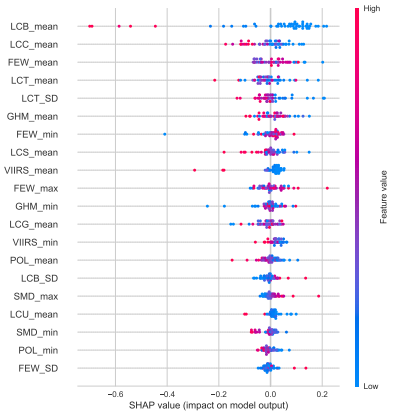


Figure 18. *Feature Contribution and Importance for the KRR Models*

The most important features are the famine early warning system, the barren land cover and the global human modification index. These are different observations made compared to the XGB classifier analysis regarding figure 11, where the most important features were the nighttime lights (VIIRS) followed by the crops land cover and the famine early warning system. Although the features contribute to the final estimation value in a different way compared to the XGB classifier, the important features are still reasonable and it is expected that the model is able to estimate poverty well and produce stable estimations for data from a different year.

The found parameters in the grid search are summarized in table 9 below.

Table 9. *The optimized parameters found in the grid search specified in table 3.*

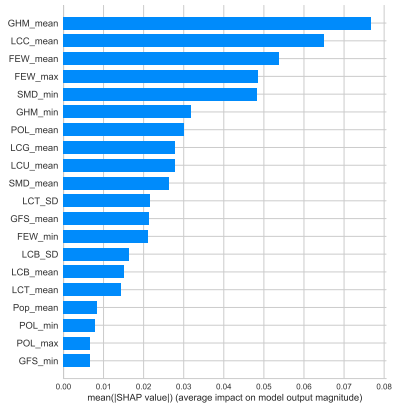
	Kernel Ridge Regression		
Poverty Level	Kernel	Regularization	Gamma
$1.90	rbf	0.024	4.92
$3.20	rbf	0.024	4.92
$5.50	rbf	0.0028	1.70

As can be seen in table 9, all models have chosen the radial basis function (rbf) as a kernel function. The rbf kernel requires the parameter gamma to be specified, which resulted in quite high values compared to the default value which would be the inverse of the feature size.
The first two poverty levels lead to an identical model structure, while the model for the poverty level at $5.50 per day has a lower regularization and a lower gamma value compared to the previous two. The lower regularization can also be observed in 18e. Compared to the other feature importance plots, the most important feature carries more weight compared to the following features. This could indicate an over-fit for this model, most likely caused by the skewed distribution of target values. This is further supported in the appendix 6 (figure 32), where the residual plots are presented, and a cone shape can be observed.

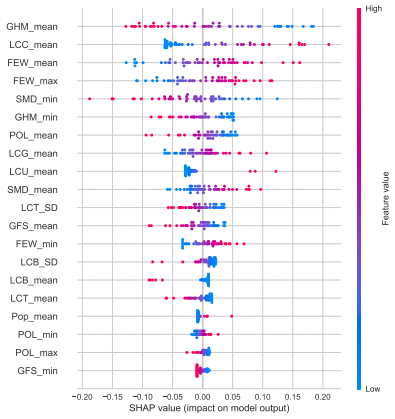
Bayesian Ridge Regression

Next, the BRR model was analyzed. The previous results have shown that the Bayesian ridge regression is not able to produce accurate poverty estimates from the chosen features. Since BRR is essentially a linear regression with some additional tweaks, it can be concurred that a linear model is not well-suited for this problem. Although the models with this algorithm don't perform particularly well, the individual feature importance correlates well compared to other models. This is another indicator that linear models are not able to capture the full complexity of the relationships between inputs and target values.

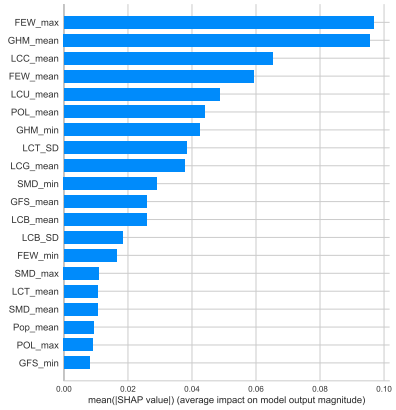
(a) *Feature Importance at $1.90*



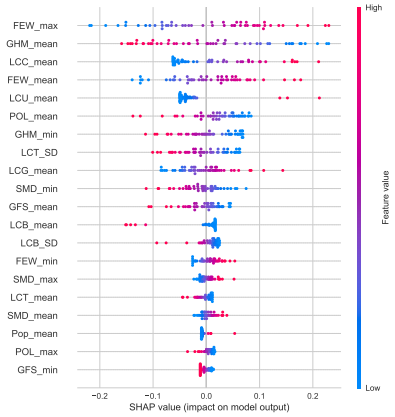
(b) *Feature Contribution at $1.90*



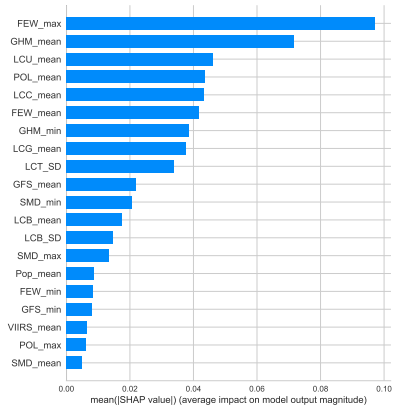
(c) *Feature Importance at $3.20*



(d) *Feature Contribution at $3.20*



(e) *Feature Importance at $5.50*



(f) *Feature Contribution at $5.50*

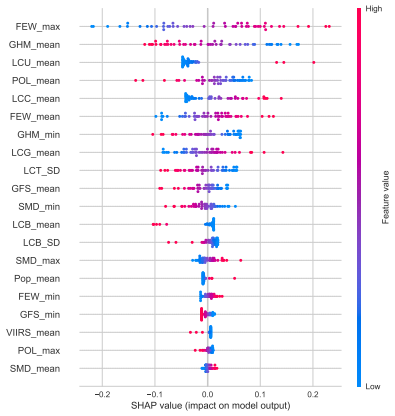


Figure 19. *Feature Contribution and Importance for the BRR Models*

As seen in figure 19, the Bayesian ridge regression only allows linear correlations between target values and features. This can be easily observed in the feature contribution plots on the previous page, where the feature value

always correlates either positively or negatively with the shapley value, which is used as an estimate for the impact on the model output.
Table 10 below summarizes the optimized parameters found in the grid search.

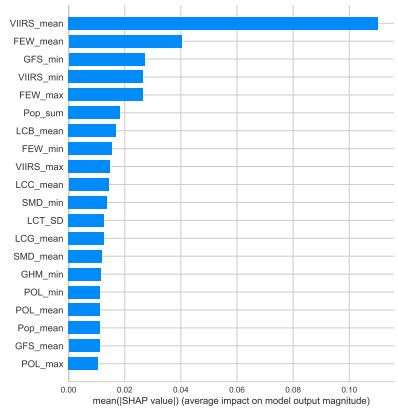
Table 10. *The optimized parameters found in the grid search specified in table 4.*

	Bayesian Ridge Regression			
Poverty Level	Alpha 1	Alpha 2	Lambda 1	Lambda 2
$1.90	1.00E-10	1000	1000	1000
$3.20	1000	1000	1.00E-10	0.56
$5.50	1000	1000	1.00E-10	0.56

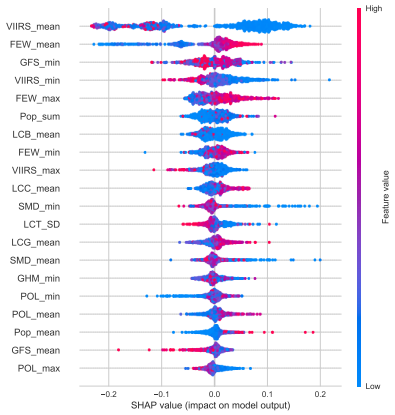
XGB Regression

The third model under investigation is the regressor built with the XGBoost framework. It is the only model using gradient boosting trees for its poverty estimation, and as for the classifier used in preprocessing the nighttime light intensity is the most important feature for this model. For no other framework was the VIIRS data of such great importance. Additionally, the global friction surface (GFS), which is used as an estimate for land travel speed, is of higher importance compared to other models. Although the important features seem reasonable, the high importance of the average VIIRS value is concerning, especially in figure 20a. In general, like all other models except BRR, nonlinear feature contributions are observed in the right column in the following figure.

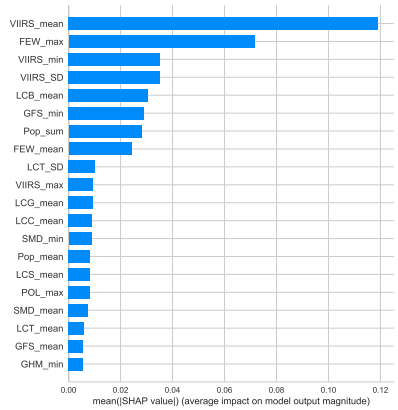
(a) *Feature Importance at $1.90*



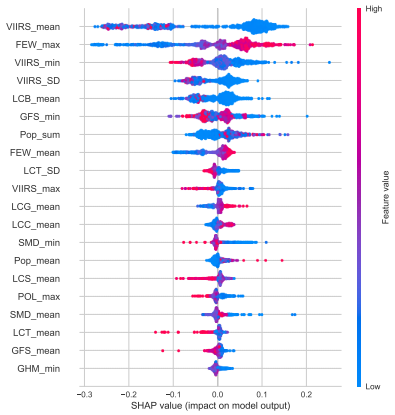
(b) *Feature Contribution at $1.90*



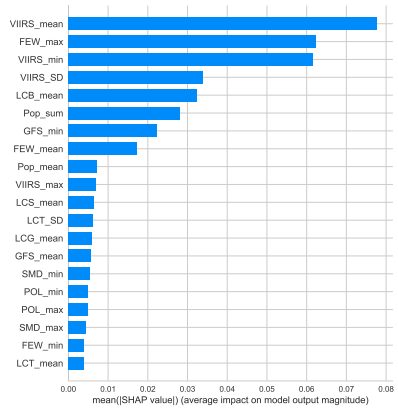
(c) *Feature Importance at $3.20*



(d) *Feature Contribution at $3.20*



(e) *Feature Importance at $5.50*



(f) *Feature Contribution at $5.50*

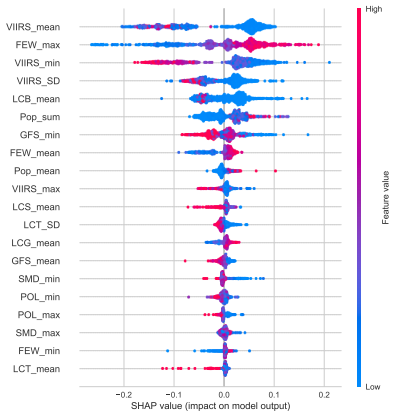


Figure 20. *Feature Contribution and Importance for the XGR Models*

In figure 20 we observe similar trends as for the classifier. As observed previously in figure11a, the nighttime lights data is a very important feature for the XGBoost models. Like other models, the famine early warning data as well

as the land cover information also show a comparably large feature importance. The most important parameters found in the grid search for the XGBoost

Table 11. *The optimized parameters found in the grid search specified in table 6.*

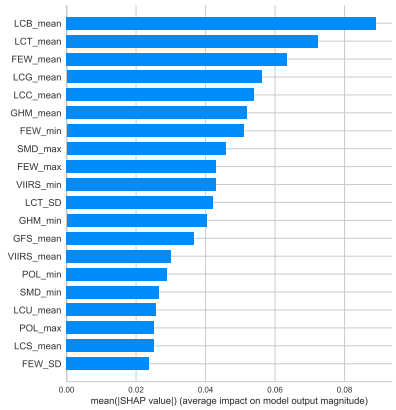
	XGB Regression			
Poverty Level	Eta	Gamma	Child Weight	Max Depth
$1.90	0.5	0	10	5
$3.20	0.1	0	5	5
$5.50	0.1	0.001	5	5

regressor are summarized in table 11. Surprisingly, the optimized parameter do not restrict the model much. Only the child weight hyper parameter stands out as a regularizing force.

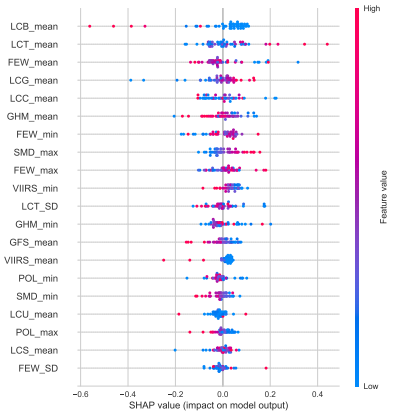
Support Vector Regression

The support vector regression is similar to the kernel ridge regression, as they both rely on kernel functions that allow them to solve problems in higher dimensions. Like all models except XGB, the VIIRS data is important, but not found at the top of the list. Instead, other environmental data, especially the land cover values (barren, tree, crops, and grass) and the famine early warning data is again of high importance. Interestingly, this is the only model where the three most important features are consistent along the three poverty rates which the models were evaluated upon.

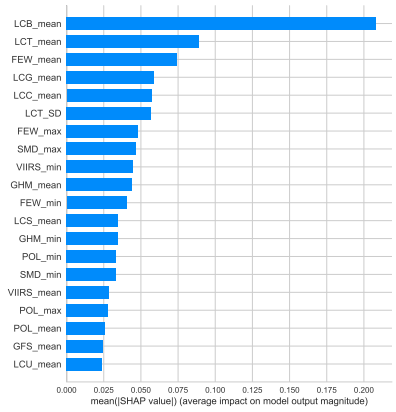
(a) *Feature Importance at $1.90*



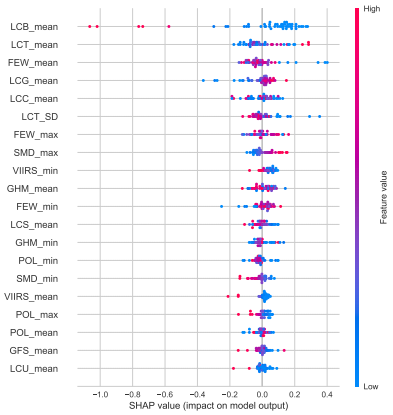
(b) *Feature Contribution at $1.90*



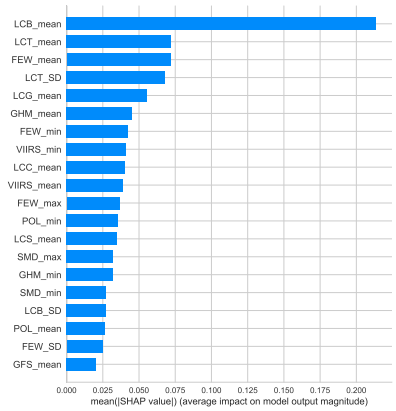
(c) *Feature Importance at $3.20*



(d) *Feature Contribution at $3.20*



(e) *Feature Importance at $5.50*



(f) *Feature Contribution at $5.50*

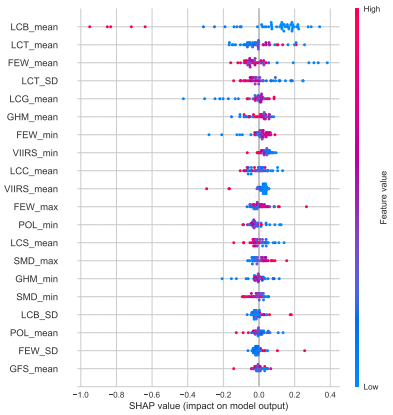


Figure 21. *Feature Contribution and Importance for the SVR Models*

Figure 21 shows how the first model has a wider distribution of the feature importance compared to the other two. However, contrary to the example for KRR in figure 18e, this is not caused by the different selection of

hyperparameters in the grid search. As we can see in the following table, the grid search resulted in equal parameters for all three models.

Table 12. *The optimized parameters found in the grid search specified in table 5.*

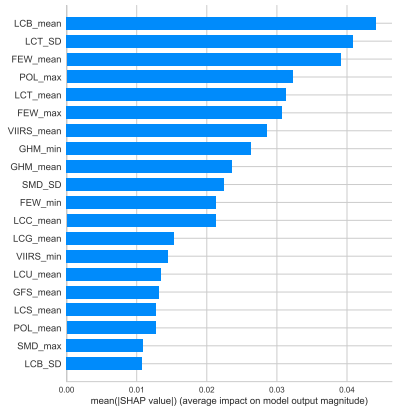
	SVM Regression		
Poverty Level	Kernel	Gamma	Regularization Strength
$1.90	rbf	scale	65
$3.20	rbf	scale	65
$5.50	rbf	scale	65

Artificial Neural Net Regression

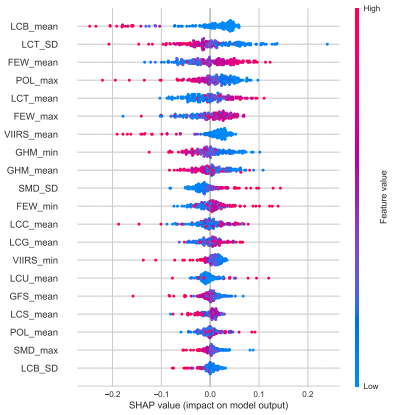
The last model under investigation is inherently different from all previous ones. For artificial neural networks, the calculation of feature importance and contribution is something that can usually only be done numerically. Since solutions that offer this have only become available recently, this is something quite new. However, the shap package which was also used for the previous models supports TensorFlow as well, which makes the analysis consistent along all frameworks.

In figure 22 on the following page, the nighttime lights, the land cover features as well as the famine early warning system are the dominant features for the neural networks used for poverty estimations at the three distinct levels analyzed in this work. It is good to see that the no single feature dominates the estimated value alone and that the important features are somewhat consistent to ones found in earlier models.

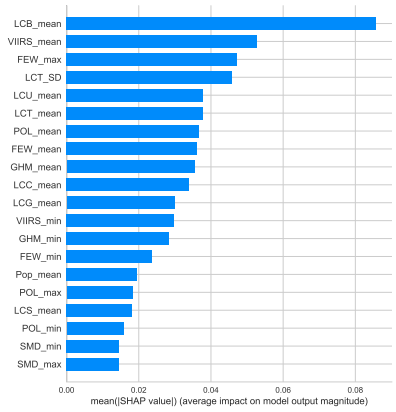
(a) *Feature Importance at $1.90*



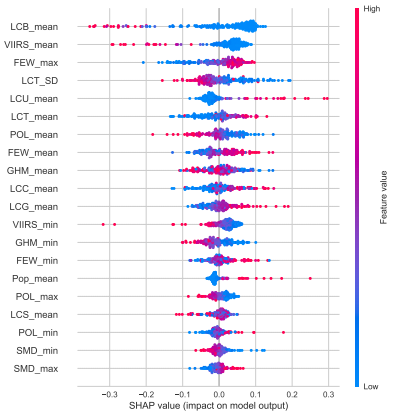
(b) *Feature Contribution at $1.90*



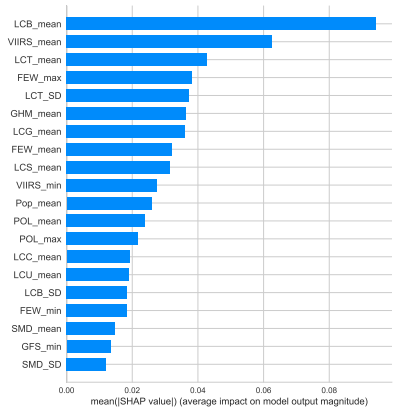
(c) *Feature Importance at $3.20*



(d) *Feature Contribution at $3.20*



(e) *Feature Importance at $5.50*



(f) *Feature Contribution at $5.50*

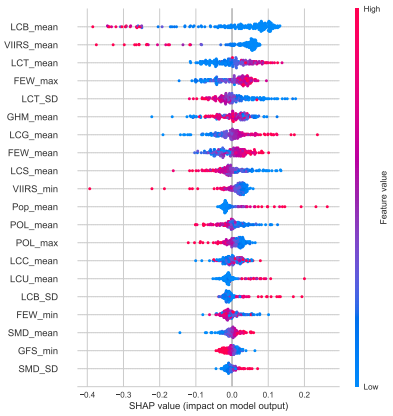


Figure 22. *Feature Contribution and Importance for the ANN Models*

The following table summarizes the neural networks developed with the model building function in a grid search like setting.

Table 13. *The optimized parameters found in the grid search from table 7.*

	Artificial Neural Net Regression			
Poverty Level	Layers	Regularization	Dropouts	Activation Function
$1.90	2	yes	1,2	sigmoid
$3.20	2	yes	2	sigmoid
$5.50	2	yes	None	sigmoid

The optimization lead to models that all use some regularization in at least one of their layers. This limits over-fitting and allows ANNs to make accurate predictions although the parameter size is similar and sometimes even larger than the amount of training samples. An additional feature that the grid search is taking advantage of is dropout layers. Like weight regularization, they limit over-fitting. However, they do so by blending out a part of inputs within the neural network during training, which limits over-fitting by forcing a level of generalization. The value in the column for dropouts stands for those layers which use a dropout function during training. All neural networks have a sigmoid activation function in their last layer, which is intrinsically limited between 0 and 1, which is the same range as the poverty rates. In appendix 6, figure 42, this has an effect on the shape of the residuals. For more information about the model structure, please refer to appendix 7, where a complete model summary is presented.

Conclusions

Summary

In this research, five different classes of algorithms have been tested in hope to find an adequate model framework that is able to accurately estimate poverty on a large area of interest to contribute to future poverty mapping approaches. For this purpose, a vast amount of data was extracted from different sources through the Google earth engine (GEE). The data was extracted in tables where each row contained distinct region of interest. This region can be a country (admin 0), a state or province (admin 1) or a county (admin 2). In the columns, the input variables were collected, which were aggregated using GEE's built-in functions for this purpose. Since this data contained information from different sources and corresponds to different sized regions a lot of effort was put into finding an adequate method of preprocessing. It was found that normalization and other nonlinear transformations react sensitive to changes caused by the investigation of different administrative levels. Therefore, it was decided to implement minimum - maximum scaling to limit all data between 0 and 1. The poverty rates were already limited within this range, which is why they were excluded from any preprocessing. However, especially the distribution of poverty rates at the income level of $5.50 a day is heavily skewed to the left in Africa. This issue was left unaddressed in this study, since the other two poverty rates which models were developed for showed nearly uniform distributions across the whole training data and finding a way to generalize the transformation of poverty rates across the different levels of administration was found to be difficult.

After some initial testing it was found that the likelihood of over-fits is high. Therefore, an extensive grid search using 5-fold cross validation was performed during training to find a good set of model hyper parameters. An additional factor for over-fitting in the first few tests was the high number of input features, which lead to complicated models with a lot of parameters. This

allowed the models to over-fit although they were trained using cross validated scores. Therefore, the feature size was reduced. Unfortunately, no solution was found to perform and generalize recursive or parallel feature selection for all models.

Initially, the idea was to use the first N principle components found during training as input features. However, it was found that the parameters required for the transformation which are determined during training react sensitive to changes in levels of administration and time. This meant that the information from this analysis was only used to estimate the required number of features for this study. The resulting value depends on the poverty level under investigation, generally speaking about 25 principle components explain around 99% of the variance in the data.

Therefore, a correlation analysis was performed to determine the level of correlation in the input features. Some interesting patterns were found, and an algorithm was chosen to select a set of relatively uncorrelated features from the possible feature set. This resulted in 29 features, which were used as input variables for the finally developed models. After initial testing it was found that linear and stepwise regression is not suited for this method of poverty estimation. It was concluded that more advanced frameworks are necessary to provide accurate estimations.

Therefore, the model selection was based on commonly implemented advanced regression methods which were thought to be better suited for this problem. Kernel ridge regression, Bayesian ridge regression and support vector regression frameworks for python have been obtained by sklearn (Pedregosa et al., 2011). Additionally, the TensorFlow environment and keras was used to build artificial neural networks (Abadi et al., 2015) and (Chollet et al., 2015). Finally, the xgboost framework provided gradient boosted decision tree ensembles which were used to build a classifier used in preprocessing and different regression models at the three poverty levels (Chen & Guestrin,

2016). The models were initially trained on data containing different levels of administration, followed by a transfer learning step in hopes to make the models more accurate for predictions with more uniform input data.

Discussion

In general, Bayesian ridge regression is not suited for poverty estimation with this kind of data. Although a lot of effort was put into finding unbiased parameters, the model tended to under-fit and simply predicted around the average poverty rate in the data. Additionally, it was the only model to produce negative R^2 values in the testing set, meaning that predicting the average value would be a better method. This indicates that linear models are simply not suited for poverty estimation at pre-defined administrative levels with this type of data.

The remaining frameworks, namely the kernel ridge regression, the support vector regression, the xgboost regressor and the neural net produced much more accurate results with smaller errors and were overall comparable to each other. Surprisingly, the developed ensemble models did not perform well in the training set but after transfer learning exceeded the scores of the remaining models during testing. Intuitively this makes sense, since one can imagine that many estimations will balance each other out and reduce the error, however the ensemble methods sometimes exceeded accuracy of the models they use as inputs, which is truly fascinating.

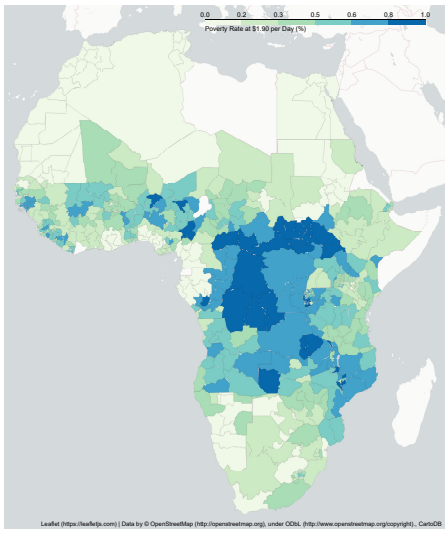
Another interesting pattern was observed regarding the use of classifiers in the preprocessing. While models that use classification in their preprocessing performed worse in the training set compared to models that did not, they performed better on average in the testing set. The classification was used to separate the "poor" areas from the "not poor" areas, which apparently reduced over-fitting since the accuracy increased during testing, but was lower during training.

Overall, no single model exceeded the performance of all others. However, the xgboost framework and kernel ridge regression performed very well with very little modification. Developing a neural network is more challenging and complicated compared to the previous models, and analysis is usually troublesome. Nowadays there are solutions to these limitations and adequate models can be developed fast and subsequently analyzed. Except for the ensemble method based on a neural network, the other ANNs performed similar to the support vector regression implemented in this study. Those two types of models produced lower R^2 values compared to XGB and KRR, but still adequate results which support further investigation.
In conclusion, it was found that ensemble methods perform better on average than simple model frameworks when using regression approaches for poverty estimation with aggregated values at the level of interest. Furthermore, models with R^2 values of 0.7 and higher were obtained in the testing set for the first level of administration for all models except BRR. This means that those models can be used to estimate poverty without any poverty data as inputs with quite high accuracy considering that this is not a scientific problem. Nonetheless, for accurate poverty maps more sophisticated models and approaches are required.

Poverty Maps

Generated poverty maps can be easily compared visually to maps generated with historical values. This allows a visual analysis of the poverty estimates which can be calculated with any of the models developed in this study. For comparison, only the stacked neural network is analyzed and compared to literature values from the year 2018:

(a) *Poverty Map with Literature Values from 2018*



(b) *Poverty Map with Estimated Values for 2019*

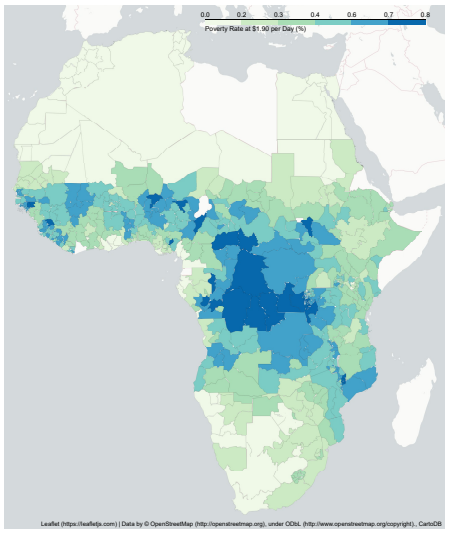


Figure 23. *Comparison of poverty maps with historical values (23a) and estimated values (23b).*

The two figures above do not appear much different from each other. The overall distribution of poverty seems similar, which is a good indicator for the model performance. A similar performance was observed for a data set containing exclusively admin 1 regions:

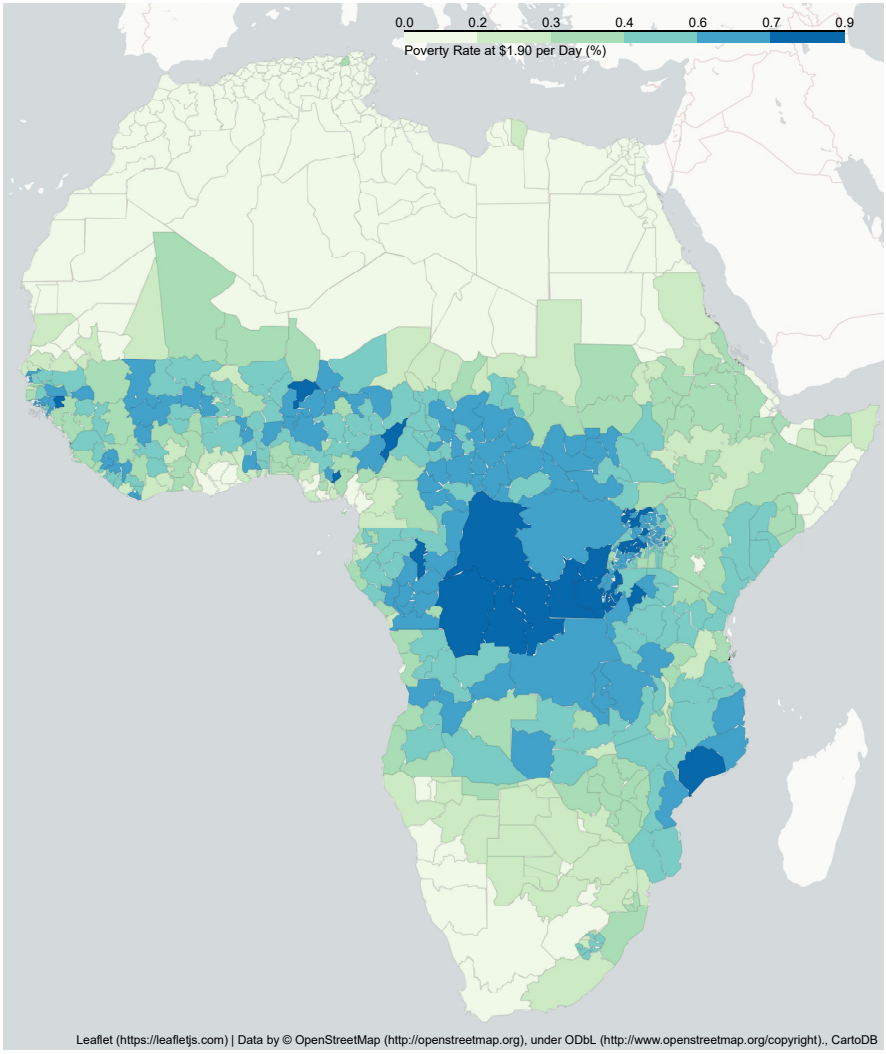


Figure 24. *Estimated poverty rates at the $1.90 level for the year 2019 for regions of the first administrative level.*

Fortunately, the distribution of poverty rates still appears similar to the literature data. Compared to before, the amount of samples has increased from 556 to 746. Nonetheless, the model performance is very similar.

The analysis for results at the second administrative level is not so straightforward:

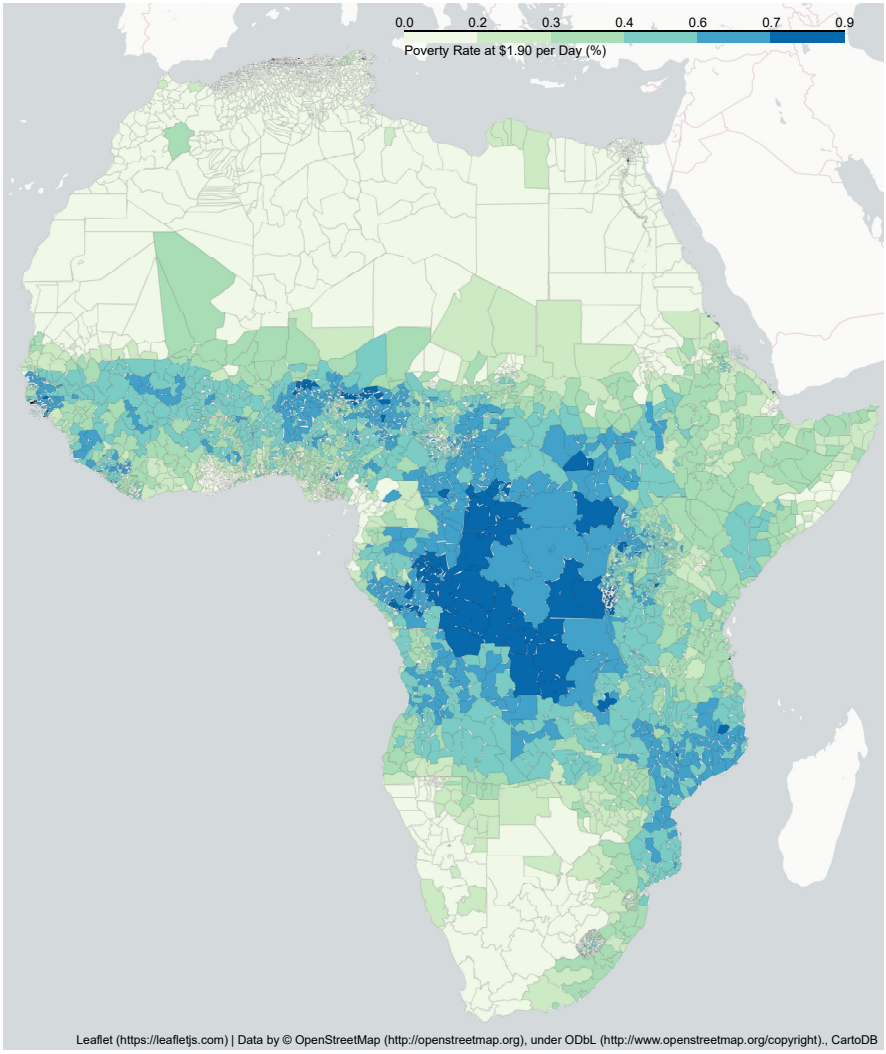


Figure 25. *Estimated poverty rates at the $1.90 level for the year 2019 for regions of the second administrative level.*

It was observed that the size of the administrative regions at the second level is more heterogeneous compared to the first administrative level. For example, Tunisia, located in northern Africa, is characterized by very small areas at this resolution, especially when compared to regions that are comparably large, like those found in Libya. The overall smaller area of each sample increased the sample size dramatically from 556 to 6364. Nonetheless, the poverty estimates still follow a similar distribution compared to the example from literature. This indicates that at least for the stacked neural network, the found parameters are useful for poverty estimation where there is a lack of data. Nonetheless, there

are also multiple limitations to this approach.

Limitations

The approach of using aggregated values of remote-sensing data in this work is much less data intensive compared to most current poverty mapping approaches which focus mainly on machine learning assisted small area estimation techniques. The lack of standardization regarding the extracted feature values also poses a serious limitation. The areas for which features were extracted are not the same size, which obviously has an effect on the aggregated feature values. This lack of standardization limited the possibilities for data preparation and preprocessing in this study. Instead, extraction and model development should ideally be based on aggregated values for a pre-defined grid of equal area size.

Another limitation is the range of availability in time for some features. If all features were available for each year, a model could be developed taking the year into consideration as an additional input. In this case this could not be done, and those features were instead used as a regularizing force, however this also increases risks of over-fitting and sensitive model parameters.

Furthermore, this analysis focused on Africa and was never tested in other regions of the world. I suspect that certain features, like the famine early warning system, have a significant importance in Africa, but might not carry as much weight in models developed for regions where poverty and famine are less of an issue.

Another limitation was the lack of training data for data at the second level of local administration. Detailed poverty information is still difficult to obtain, especially if the area of interest is large. This meant that the models which were exposed to the largest testing set had the smallest training size.

Finally, the inaccuracy of the population data used to calculate poverty rates for regions where data was available added another source of error. Although

implemented consistently, it would be better to work with population data from a different source.

Global Justice and Ethical Considerations

Global poverty is a significant challenge in maintaining global justice. Poverty rates have mainly decreased in the past, however there are indications that the consequences of Covid-19 have reversed some of the progress. Additionally, the effectiveness of poverty reducing efforts have been uneven. For these reasons, it is estimated that about 10% of the world's population lives in extreme poverty on less than $1.90 a day. This work aims to explore additional approaches to locate affected areas to formulate effective measures with today's possibilities. The data used in this work is freely available and does not contain any sensitive information. The model results should be treated as such, and not be mistaken as poverty estimates. The objective of this study was to contribute to future approaches of poverty estimation and to assist locating affected areas, so the people can be helped effectively.

Recommendations and Future Work

As suggested in the previous chapter, I strongly suggest development of a similar model framework for predefined grids of a much smaller size than the resolution of the poverty map itself. This allows a bottom-up approach, which is much less data intensive than most current machine learning assisted small area estimation techniques, but more data intensive compared to the larger aggregations and top-down approach implemented in this study.
I suggest labeling the areas in the grid with the corresponding identifiers for later aggregation to an administrative region of which a reliable poverty rate is available. This simplifies later aggregation steps and will make model evaluation much easier.
Relying on smaller grids of equal size also allows more sophisticated techniques during preprocessing. Increasing the level of standardization

simplifies transformations and other operations with the data which ultimately makes development of reliable models much easier. Additionally, this should decrease the risk of sensitive model parameters. Cleaner data and a high level of standardization is possibly the first step to a generally applicable poverty mapping solution that can be easily implemented and distributed.

For this purpose, some training data at the grid level would be helpful. This data can then be used to either train the model initially and use transfer learning approaches to build a model that is generally applicable, or the estimations at the grid level can be used to benchmark and test the developed models.

For a generally applicable small area estimation method, I suggest investigating whether convolutional neural networks similar to the ones used by Xie et al. (2015) and Jean et al. (2016) could be expanded to account for additional inputs that aren't available at the pixel level. This approach has been described by Rosebrock (2021) and seems very promising to improve current poverty mapping processes in general, but especially small area estimation methods, since they rely heavily on convolutional neural networks to analyze satellite image data. The increased amount of computational power should not pose a significant limitation, since the image analysis itself is very costly already. The additional high-quality data could be useful to develop a model framework that is not limited to a small geographical area but can be used on a global scale.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Retrieved from https://www.tensorflow.org/ (Software available from tensorflow.org)

Akinyemi, F. (2010). A conceptual poverty mapping data model. *Transactions in GIS*, *14*, 85–100. doi: 10.1111/j.1467-9671.2010.01207.x

Ayush, K., Uzkent, B., Burke, M., Lobell, D., & Ermon, S. (2020). *Efficient poverty mapping using deep reinforcement learning.*

Bedi, T., Coudouel, A., & Simler, K. (2007). *More Than a Pretty Picture : Using Poverty Maps to Design Better Policies and Interventions* (No. 6800). The World Bank. Retrieved from https://ideas.repec.org/b/wbk/wbpubs/6800.html doi: 10.1596/978-0-8213-6931-9

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2939672.2939785 doi: 10.1145/2939672.2939785

Chollet, F., et al. (2015). *Keras.* GitHub. Retrieved from https://github.com/fchollet/keras

Curve fitting with bayesian ridge regression. (n.d.). Retrieved from https://scikit-learn.org/stable/auto_examples/linear_model/plot_bayesian_ridge_curvefit.html

Goodfellow, I. J., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge, MA, USA: MIT Press. (http://www.deeplearningbook.org)

Henninger, N., & Snel, M. (2002). *Where are the poor? experiences with the*

development and use of poverty maps. World Resources Institute. Retrieved from https://www.wri.org/publication/

Hentschel, J. (2000, 02). Combining census and survey data to trace the spatial dimensions of poverty: A case study of ecuador. *World Bank Economic Review*, *14*, 147-65.

Jean, N., Burke, M., Xie, M., Davis, W. M., Lobell, D. B., & Ermon, S. (2016). Combining satellite imagery and machine learning to predict poverty. *Science*, *353*(6301), 790–794. doi: 10.1126/science.aaf7894

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. doi: 10.1145/3065386

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, *42*, 60 - 88. doi: https://doi.org/10.1016/j.media.2017.07.005

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon et al. (Eds.), *Advances in neural information processing systems 30* (pp. 4765–4774). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

Lussier, F., Thibault, V., Charron, B., Wallace, G. Q., & Masson, J.-F. (2020, Mar). Deep learning and artificial intelligence methods for raman and surface-enhanced raman scattering. *TrAC Trends in Analytical Chemistry*, *124*, 115796. doi: 10.1016/j.trac.2019.115796

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., & Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, *152*, 166 - 177. doi: https://doi.org/10.1016/j.isprsjprs.2019.04.015

Murray, J., Sargent, I., Holland, D., Gardiner, A., Dionysopoulou, K.,

Coupland, S., ... Atkinson, P. M. (2020). Opportunities for machine learning and artificial intelligence in national mapping agencies: Enhancing ordnance survey workflow. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLIII-B5-2020*, 185–189. doi: 10.5194/isprs-archives-XLIII-B5-2020-185-2020

Noe, K. (2019, Dec). *Deep learning for visual searches and mapping.* Towards Data Science. Retrieved from `https://towardsdatascience.com/deep-learning-for-visual-searches-and-mapping-89b85061ef9e`

Olivia, S., Gibson, J., Smith, A. D., Rozelle, S., & Deng, X. (2009). An empirical evaluation of poverty mapping methodology: Explicitly spatial versus implicitly spatial approach. (420-2016-26692), 42. Retrieved from `http://ageconsearch.umn.edu/record/47651` doi: 10.22004/ag.econ.47651

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al. (2019). *Keras tuner.* `https://github.com/keras-team/keras-tuner`.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pisner, D. A., & Schnyer, D. M. (2020). Chapter 6 - support vector machine. In A. Mechelli & S. Vieira (Eds.), *Machine learning* (p. 101-121). Academic Press. Retrieved from `https://www.sciencedirect.com/science/article/pii/B9780128157398000067` doi: https://doi.org/10.1016/B978-0-12-815739-8.00006-7

Quinn, J. A., Nyhan, M. M., Navarro, C., Coluccia, D., Bromley, L., & Luengo-Oroz, M. (2018, Aug). *Humanitarian applications of machine learning with remote-sensing data: review and case study in refugee*

settlement mapping. Retrieved from https://royalsocietypublishing.org/doi/full/10.1098/rsta.2017.0363

Ritchie, H., & Roser, M. (2018, Sep). *Now it is possible to take stock – did the world achieve the millennium development goals?* Retrieved from https://ourworldindata.org/millennium-development-goals

Rosebrock, A. (2021, Jul). *Keras: Multiple inputs and mixed data.* Retrieved from https://www.pyimagesearch.com/2019/02/04/keras-multiple-inputs-and-mixed-data/

Sairamya, N., Susmitha, L., Thomas George, S., & Subathra, M. (2019). Chapter 12 - hybrid approach for classification of electroencephalographic signals using time–frequency images with wavelets and texture features. In D. J. Hemanth, D. Gupta, & V. Emilia Balas (Eds.), *Intelligent data analysis for biomedical applications* (p. 253-273). Academic Press. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780128155530000136 doi: https://doi.org/10.1016/B978-0-12-815553-0.00013-6

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85 - 117. doi: https://doi.org/10.1016/j.neunet.2014.09.003

Scikit learn - cross-validation: evaluating estimator performance. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/cross_validation.html

Support vector regression (svr) using linear and non-linear kernels. (n.d.). Retrieved from https://scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html

Theodoridis, S. (2020a). Chapter 11 - learning in reproducing kernel hilbert spaces. In S. Theodoridis (Ed.), *Machine learning (second edition)*

(Second Edition ed., p. 531-594). Academic Press. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780128188033000222 doi: https://doi.org/10.1016/B978-0-12-818803-3.00022-2

Theodoridis, S. (2020b). Chapter 13 - bayesian learning: Approximate inference and nonparametric models. In S. Theodoridis (Ed.), *Machine learning (second edition)* (Second Edition ed., p. 647-730). Academic Press. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780128188033000258 doi: https://doi.org/10.1016/B978-0-12-818803-3.00025-8

Tingzon, I., Orden, A., Go, K. T., Sy, S., Sekara, V., Weber, I., ... Kim, D. (2019). Mapping poverty in the philippines using machine learning, satellite imagery, and crowd-sourced geospatial information. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLII-4/W19*, 425–431. doi: 10.5194/isprs-archives-xlii-4-w19-425-2019

Twin, A. (2021, May). *How overfitting works.* Investopedia. Retrieved from https://www.investopedia.com/terms/o/overfitting.asp

World Bank. (2018). *Piecing together the poverty puzzle*. World Bank. (License: Creative Commons Attribution CC BY 3.0 IGO) doi: 10.1596/978-1-4648-1330-6

World Bank. (2020a). *History.* Retrieved from https://www.worldbank.org/en/about/archives/history

World Bank. (2020b). *Poverty and shared prosperity 2020: reversing reversals of fortune*. World Bank. doi: 10.1596/978-1-4648-1602-4

WorldBank. (2020, Oct). Covid-19 to add as many as 150 million extreme poor by 2021. *World Bank*. Retrieved from https://www.worldbank.org/en/news/press-release/2020/10/07/covid-19-to-add-as-many-as-150-million-extreme-poor-by-2021

Xie, S. M., Jean, N., Burke, M., Lobell, D. B., & Ermon, S. (2015). *Transfer learning from deep features for remote sensing and poverty mapping* (Vol. abs/1510.00098). Retrieved from http://arxiv.org/abs/1510.00098

Zhao, X., Yu, B., Liu, Y., Chen, Z., Li, Q., Wang, C., & Wu, J. (2019). Estimation of poverty using random forest regression with multi-source data: A case study in bangladesh. *Remote Sensing*, *11*(4). Retrieved from https://www.mdpi.com/2072-4292/11/4/375 doi: 10.3390/rs11040375

Appendices

Appendix 1: Variable Description

The following pages summarize the variables in tabular format with links to the respective providers. For all variables, the minima, maxima, mean, and the sum was extracted. Only the population sum was used, and the landcover minima and maxima values were also removed from the data. This left 41 possible input features for the models.

The variable extraction process was repeated for each year and resolution (admin level) of interest. One extraction from GEE results in a csv file for every variable, which contains the aggregated statistics previously mentioned. This means that 12 csv files were obtained for every year, which were then combined to a single table for simpler handling.

The limited availability of high quality data sets for the whole region and time-frame of interest was a serious limitation. Nonetheless, the developed models showed an acceptable accuracy even with those limitations.

Abb.	Full Name	Provider	EE Name	Date Start	Date End	comment
VIIRS	Visible Infrared Imaging Radiometer Suite (VIIRS) Day/Night Band (DNB)	Earth Observation Group, Payne Institute for Public Policy, Colorado School of Mines	ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMSLCFG")	2014-01-01T00:00:00	2020-12-01T00:00:00	
Pop	WorldPop Global Project Population Data: Estimated Residential Population per 100x100m Grid Square	WorldPop	ee.ImageCollection("WorldPop/GP/100m/pop")	2000-01-01T00:00:00	2021-01-01T00:00:00	
POL	Sentinel-5P NRTI NO2: Near Real-Time Nitrogen Dioxide	European Union/ESA/Copernicus	ee.ImageCollection("COPERNICUS/S5P/NRTI/L3_NO2")	2018-07-10T10:05:44	2021-04-21T00:00:00	Since 2018
LCT	Copernicus Global Land Cover Layers: CGLS-LC100 collection 3 Tree	Copernicus	ee.ImageCollection("COPERNICUS/Landcover/100m/Proba-V-C3/Global")	2015-01-01T00:00:00	2019-12-31T00:00:00	

LCU	Copernicus Global Land Cover Layers: CGLS-LC100 collection 4 Urban	Copernicus	ee.ImageCollection("COPERNICUS/Landcover/100m/Proba-V-C3/Global")	2015-01-01T00:00:01	2019-12-31T00:00:01	
LCG	Copernicus Global Land Cover Layers: CGLS-LC100 collection 5 Grass	Copernicus	ee.ImageCollection("COPERNICUS/Landcover/100m/Proba-V-C3/Global")	2015-01-01T00:00:02	2019-12-31T00:00:02	
LCS	Copernicus Global Land Cover Layers: CGLS-LC100 collection 6 Shrubs	Copernicus	ee.ImageCollection("COPERNICUS/Landcover/100m/Proba-V-C3/Global")	2015-01-01T00:00:03	2019-12-31T00:00:03	
LCC	Copernicus Global Land Cover Layers: CGLS-LC100 collection 7 Crops	Copernicus	ee.ImageCollection("COPERNICUS/Landcover/100m/Proba-V-C3/Global")	2015-01-01T00:00:04	2019-12-31T00:00:04	
LCB	Copernicus Global Land Cover Layers: CGLS-LC100 collection 8 Bare	Copernicus	ee.ImageCollection("COPERNICUS/Landcover/100m/Proba-V-C3/Global")	2015-01-01T00:00:05	2019-12-31T00:00:05	
GHM	CSP gHM: Global	Conservation Science Partners	ee.ImageCollection("CSP/HM/GlobalHumanModification")	2016-01-01T00:00:00	2016-12-31T00:00:00	single image

	Human Modification					
FEW	FLDAS: Famine Early Warning Systems Network (FEWS NET) Land Data Assimilation System	NASA GES DISC at NASA Goddard Space Flight Center	ee.ImageCollection("NASA/FLDAS/NOAH01/C/GL/M/V001")	1982-01-01T00:00:00	2021-03-01T00:00:00	
SMD	TerraClimate: Monthly Climate and Climatic Water Balance for Global Terrestrial Surfaces, University of Idaho	University of California Merced	ee.ImageCollection("IDAHO_EPSCOR/TERRACLIMATE")	1958-01-01T00:00:00	2020-12-01T00:00:00	
GFS	Global Friction Surface 2019	Malaria Atlas	ee.Image("Oxford/MAP/friction_surface_2019")	2019-01-01T00:00:00 -	2020-01-01T00:00:00	

Appendix 2: Feature Dependence on Admin Level

The following table shows the change of feature values depending on the administrative level averaged over all years.

Feature	Admin Level	Mean	Median	St. Dev	St. Dev (%)
	Mixed	0.0%	0.0%	0.0%	0.0%
Poor190	Admin 1	-15.8%	-23.1%	4.4%	24.0%
	Admin 2	-30.2%	-39.9%	10.3%	58.0%
	Mixed	0.0%	0.0%	0.0%	0.0%
Poor320	Admin 1	-12.2%	-11.4%	19.1%	35.7%
	Admin 2	-28.3%	-25.7%	39.9%	95.2%
	Mixed	0.0%	0.0%	0.0%	0.0%
Poor550	Admin 1	-7.8%	-3.2%	39.0%	50.7%
	Admin 2	-20.6%	-10.5%	74.5%	119.9%
	Mixed	0.0%	0.0%	0.0%	0.0%
VIIRS_min	Admin 1	195.6%	948.9%	-53.4%	-84.2%
	Admin 2	1453.1%	1797.7%	756.6%	-44.8%
	Mixed	0.0%	0.0%	0.0%	0.0%
VIIRS_max	Admin 1	39.6%	29.4%	25.1%	-10.4%
	Admin 2	-68.2%	-59.7%	-54.5%	43.3%
	Mixed	0.0%	0.0%	0.0%	0.0%
VIIRS_mean	Admin 1	30.9%	94.3%	10.1%	-15.9%
	Admin 2	147.1%	127.3%	141.4%	-2.3%
	Mixed	0.0%	0.0%	0.0%	0.0%
VIIRS_SD	Admin 1	60.1%	13.2%	89.8%	18.6%
	Admin 2	37.0%	5.6%	144.0%	78.2%
	Mixed	0.0%	0.0%	0.0%	0.0%
Pop_min	Admin 1	-32.0%	31.3%	-20.2%	17.3%
	Admin 2	197.3%	703.1%	205.4%	2.7%

	Mixed	0.0%	0.0%	0.0%	0.0%
Pop_max	Admin 1	-5.6%	24.9%	-16.5%	-11.5%
	Admin 2	-67.7%	-58.4%	-71.6%	-12.0%
	Mixed	0.0%	0.0%	0.0%	0.0%
Pop_mean	Admin 1	-8.5%	-7.1%	-0.7%	8.5%
	Admin 2	42.4%	47.2%	115.0%	51.0%
	Mixed	0.0%	0.0%	0.0%	0.0%
Pop_SD	Admin 1	1.5%	-3.3%	4.6%	3.1%
	Admin 2	-21.0%	-18.2%	-16.4%	5.9%
	Mixed	0.0%	0.0%	0.0%	0.0%
Pop_sum	Admin 1	2.0%	22.8%	-18.9%	-20.5%
	Admin 2	-90.7%	-92.2%	-88.9%	19.5%
	Mixed	0.0%	0.0%	0.0%	0.0%
GHM_min	Admin 1	1.5%	16.7%	-6.1%	-7.5%
	Admin 2	168.5%	337.7%	61.0%	-40.1%
	Mixed	0.0%	0.0%	0.0%	0.0%
GHM_max	Admin 1	2.1%	2.3%	-11.7%	-13.5%
	Admin 2	-11.7%	-11.5%	34.7%	52.5%
	Mixed	0.0%	0.0%	0.0%	0.0%
GHM_mean	Admin 1	1.5%	-0.1%	5.5%	3.9%
	Admin 2	30.0%	49.2%	15.2%	-11.4%
	Mixed	0.0%	0.0%	0.0%	0.0%
GHM_SD	Admin 1	-0.8%	-0.5%	0.8%	1.6%
	Admin 2	-27.1%	-31.5%	-5.1%	30.2%
	Mixed	0.0%	0.0%	0.0%	0.0%
GFS_min	Admin 1	-2.9%	-11.1%	-15.5%	-13.0%
	Admin 2	18.6%	33.3%	116.8%	82.7%
	Mixed	0.0%	0.0%	0.0%	0.0%
GFS_max	Admin 1	-0.8%	1.9%	-6.7%	-6.0%

	Admin 2	-40.7%	-24.6%	-42.8%	-3.4%
	Mixed	0.0%	0.0%	0.0%	0.0%
GFS_mean	Admin 1	5.3%	2.4%	-3.0%	-7.8%
	Admin 2	-9.7%	-7.8%	3.4%	14.5%
	Mixed	0.0%	0.0%	0.0%	0.0%
GFS_SD	Admin 1	-3.1%	-1.4%	-11.2%	-8.4%
	Admin 2	-7.1%	-2.3%	-0.6%	6.9%
	Mixed	0.0%	0.0%	0.0%	0.0%
FEW_min	Admin 1	-11.4%	-24.8%	-4.4%	7.9%
	Admin 2	42.3%	127.0%	-4.2%	-32.7%
	Mixed	0.0%	0.0%	0.0%	0.0%
FEW_max	Admin 1	-2.3%	-3.5%	0.6%	3.0%
	Admin 2	-18.6%	-25.4%	-1.9%	20.6%
	Mixed	0.0%	0.0%	0.0%	0.0%
FEW_mean	Admin 1	-2.4%	-5.7%	-0.2%	2.2%
	Admin 2	-6.9%	-15.2%	-4.5%	2.5%
	Mixed	0.0%	0.0%	0.0%	0.0%
FEW_SD	Admin 1	0.6%	4.2%	-2.8%	-3.4%
	Admin 2	-49.0%	-66.2%	-7.2%	81.8%
	Mixed	0.0%	0.0%	0.0%	0.0%
POL_min	Admin 1	4.6%	2.3%	9.5%	4.7%
	Admin 2	58.9%	34.4%	123.0%	40.3%
	Mixed	0.0%	0.0%	0.0%	0.0%
POL_max	Admin 1	8.3%	6.5%	2.1%	-5.8%
	Admin 2	-9.7%	-9.0%	-30.6%	-23.1%
	Mixed	0.0%	0.0%	0.0%	0.0%
POL_mean	Admin 1	6.8%	3.5%	24.7%	16.7%
	Admin 2	27.1%	17.6%	77.0%	39.3%
	Mixed	0.0%	0.0%	0.0%	0.0%
POL_SD					

	Admin 1	17.6%	18.8%	12.9%	-4.0%
	Admin 2	-34.7%	-36.6%	-42.2%	-11.6%
	Mixed	0.0%	0.0%	0.0%	0.0%
SMD_min	Admin 1	14.7%	21.2%	23.5%	-44.9%
	Admin 2	50.1%	48.1%	-1.4%	-97.6%
	Mixed	0.0%	0.0%	0.0%	0.0%
SMD_max	Admin 1	1017.9%	160.5%	31.1%	114.3%
	Admin 2	-692.3%	-72.8%	-3.3%	87.8%
	Mixed	0.0%	0.0%	0.0%	0.0%
SMD_mean	Admin 1	31.1%	41.9%	22.9%	-78.3%
	Admin 2	35.7%	38.2%	9.9%	-70.9%
	Mixed	0.0%	0.0%	0.0%	0.0%
SMD_SD	Admin 1	2.3%	2.6%	9.2%	6.7%
	Admin 2	-63.9%	-70.3%	-43.4%	56.8%
	Mixed	0.0%	0.0%	0.0%	0.0%
LCB_mean	Admin 1	19.2%	22.1%	8.9%	-8.6%
	Admin 2	3.6%	116.4%	-1.3%	-4.7%
	Mixed	0.0%	0.0%	0.0%	0.0%
LCB_SD	Admin 1	10.2%	27.8%	2.5%	-7.0%
	Admin 2	-17.8%	39.9%	-27.2%	-11.4%
	Mixed	0.0%	0.0%	0.0%	0.0%
LCC_mean	Admin 1	2.1%	2.2%	1.9%	-0.2%
	Admin 2	34.6%	42.3%	29.1%	-4.0%
	Mixed	0.0%	0.0%	0.0%	0.0%
LCC_SD	Admin 1	6.1%	11.4%	4.2%	-1.8%
	Admin 2	3.8%	8.0%	1.8%	-2.0%
	Mixed	0.0%	0.0%	0.0%	0.0%
LCS_mean	Admin 1	-3.4%	-5.1%	-3.1%	0.3%
	Admin 2	-8.2%	-9.3%	-4.7%	3.8%

	Mixed	0.0%	0.0%	0.0%	0.0%
LCS_SD	Admin 1	0.9%	2.4%	1.5%	0.7%
	Admin 2	-7.3%	-7.1%	0.9%	8.9%
	Mixed	0.0%	0.0%	0.0%	0.0%
LCG_mean	Admin 1	-2.1%	-1.6%	-1.7%	0.4%
	Admin 2	-3.3%	-3.2%	2.5%	5.9%
	Mixed	0.0%	0.0%	0.0%	0.0%
LCG_SD	Admin 1	-1.1%	-0.9%	-1.5%	-0.4%
	Admin 2	-19.3%	-17.5%	-19.0%	0.4%
	Mixed	0.0%	0.0%	0.0%	0.0%
LCT_mean	Admin 1	0.9%	8.6%	2.7%	1.8%
	Admin 2	-13.7%	-16.9%	-4.3%	10.8%
	Mixed	0.0%	0.0%	0.0%	0.0%
LCT_SD	Admin 1	-0.4%	-1.5%	1.0%	1.4%
	Admin 2	-18.1%	-22.9%	-11.4%	8.1%
	Mixed	0.0%	0.0%	0.0%	0.0%
LCU_mean	Admin 1	-13.6%	-9.6%	-16.2%	-2.9%
	Admin 2	31.0%	34.9%	19.0%	-9.2%
	Mixed	0.0%	0.0%	0.0%	0.0%
LCU_SD	Admin 1	-2.1%	-4.8%	-2.0%	0.1%
	Admin 2	11.7%	13.2%	2.5%	-8.3%

Table 14. *Table summarizing the change of feature values depending on the administrative level of the table.*

Appendix 3: Feature Dependence on Time

The following table shows the change of feature values depending on the time used for feature extraction. As one can see, not all features were available for all years.

Feature	Year	Mean	Median	St. Dev	St. Dev (%)
	2015	0.0%	0.0%	0.0%	0.0%
Poor190	2018	-5.8%	-5.5%	0.2%	410.1%
	2019	-5.8%	-5.5%	0.2%	410.1%
	2015	0.0%	0.0%	0.0%	0.0%
Poor320	2018	-2.7%	-3.4%	0.8%	233.8%
	2019	-2.7%	-3.4%	0.8%	233.8%
	2015	0.0%	0.0%	0.0%	0.0%
Poor550	2018	-1.1%	-0.5%	2.3%	298.8%
	2019	-1.1%	-0.5%	2.3%	298.8%
	2015	0.0%	0.0%	0.0%	0.0%
VIIRS_min	2018	375.9%	13.9%	-26.4%	-119694.8%
	2019	207.9%	8.5%	-25.8%	-114150.3%
	2015	0.0%	0.0%	0.0%	0.0%
VIIRS_max	2018	-6.4%	-271.3%	-4825.5%	911.4%
	2019	-20.4%	-421.0%	-16071.9%	2988.3%
	2015	0.0%	0.0%	0.0%	0.0%
VIIRS_mean	2018	17.9%	14.1%	-29.1%	-9671.9%
	2019	14.7%	7.9%	-25.5%	-8338.1%
	2015	0.0%	0.0%	0.0%	0.0%
VIIRS_SD	2018	0.7%	-5.8%	85.6%	4060.6%
	2019	-4.5%	-5.7%	-24.3%	128.1%
	2015	0.0%	0.0%	0.0%	0.0%
Pop_min	2018	-17.6%	-0.1%	-33.2%	-6629.4%

	2019	-13.8%	-0.1%	-22.6%	-2986.0%
	2015	0.0%	0.0%	0.0%	0.0%
Pop_max	2018	7.2%	919.0%	2535.6%	-946.1%
	2019	12.0%	1295.0%	4457.3%	-1347.0%
	2015	0.0%	0.0%	0.0%	0.0%
Pop_mean	2018	13.2%	11.0%	110.3%	-836.5%
	2019	16.8%	13.0%	149.1%	-818.6%
	2015	0.0%	0.0%	0.0%	0.0%
Pop_SD	2018	12.7%	11.3%	264.3%	1758.8%
	2019	16.9%	17.2%	323.7%	1791.5%
	2015	0.0%	0.0%	0.0%	0.0%
Pop_sum	2018	0.9%	7521945.4%	13185766.0%	546.3%
	2019	4.2%	10054172.4%	24913292.5%	554.3%
	2015	0.0%	0.0%	0.0%	0.0%
GHM_min	2018	-1.2%	0.0%	-0.3%	-200.1%
	2019	-1.2%	0.0%	-0.3%	-200.1%
	2015	0.0%	0.0%	0.0%	0.0%
GHM_max	2018	-0.3%	-1.1%	-0.4%	-39.7%
	2019	-0.3%	-1.1%	-0.4%	-39.7%
	2015	0.0%	0.0%	0.0%	0.0%
GHM_mean	2018	1.6%	1.4%	0.1%	-55.3%
	2019	1.6%	1.4%	0.1%	-55.3%
	2015	0.0%	0.0%	0.0%	0.0%
GHM_SD	2018	0.3%	0.0%	0.1%	38.0%
	2019	0.3%	0.0%	0.1%	38.0%
	2015	0.0%	0.0%	0.0%	0.0%
GFS_min	2018	1.0%	0.0%	0.0%	54.2%
	2019	1.0%	0.0%	0.0%	54.2%
	2015	0.0%	0.0%	0.0%	0.0%
GFS_max					

	2018	1.2%	0.1%	0.0%	-35.6%
	2019	1.2%	0.1%	0.0%	-35.6%
	2015	0.0%	0.0%	0.0%	0.0%
GFS_mean	2018	-1.8%	0.0%	0.0%	74.4%
	2019	-1.8%	0.0%	0.0%	74.4%
	2015	0.0%	0.0%	0.0%	0.0%
GFS_SD	2018	-0.1%	0.0%	0.0%	-43.3%
	2019	-0.1%	0.0%	0.0%	-43.3%
	2015	0.0%	0.0%	0.0%	0.0%
FEW_min	2018	10.9%	0.0%	0.0%	-595.1%
	2019	11.9%	0.0%	0.0%	-437.7%
	2015	0.0%	0.0%	0.0%	0.0%
FEW_max	2018	6.6%	0.0%	0.0%	-372.0%
	2019	7.5%	0.0%	0.0%	-242.5%
	2015	0.0%	0.0%	0.0%	0.0%
FEW_mean	2018	7.9%	0.0%	0.0%	-369.7%
	2019	8.6%	0.0%	0.0%	-194.0%
	2015	0.0%	0.0%	0.0%	0.0%
FEW_SD	2018	5.7%	0.0%	0.0%	-208.5%
	2019	7.1%	0.0%	0.0%	-102.9%
	2015	0.0%	0.0%	0.0%	0.0%
POL_min	2018	-3.8%	0.0%	0.0%	236.8%
	2019	-9.3%	0.0%	0.0%	365.9%
	2015	0.0%	0.0%	0.0%	0.0%
POL_max	2018	-4.3%	0.0%	0.0%	87.2%
	2019	-17.6%	0.0%	0.0%	1470.5%
	2015	0.0%	0.0%	0.0%	0.0%
POL_mean	2018	-3.2%	0.0%	0.0%	25.2%
	2019	-13.0%	0.0%	0.0%	297.0%

	2015	0.0%	0.0%	0.0%	0.0%
POL_SD	2018	-4.4%	0.0%	0.0%	-16.5%
	2019	-22.1%	0.0%	0.0%	1873.2%
	2015	0.0%	0.0%	0.0%	0.0%
SMD_min	2018	27.3%	104.5%	143.9%	-8130.8%
	2019	-1.0%	14.1%	148.4%	-4077.8%
	2015	0.0%	0.0%	0.0%	0.0%
SMD_max	2018	1449.3%	108.8%	159.9%	407725.8%
	2019	1044.7%	103.0%	198.6%	431954.1%
	2015	0.0%	0.0%	0.0%	0.0%
SMD_mean	2018	50.6%	118.2%	140.6%	-27102.7%
	2019	9.8%	42.4%	175.2%	-11798.8%
	2015	0.0%	0.0%	0.0%	0.0%
SMD_SD	2018	-4.7%	-0.5%	-1.7%	218.6%
	2019	12.3%	19.3%	-10.0%	-2170.2%
	2015	0.0%	0.0%	0.0%	0.0%
LCB_mean	2018	-3.4%	4.2%	-39.7%	430.3%
	2019	-3.9%	-5.3%	-57.7%	338.5%
	2015	0.0%	0.0%	0.0%	0.0%
LCB_SD	2018	-6.8%	3.7%	-61.8%	81.4%
	2019	-8.5%	-4.5%	-87.5%	-102.0%
	2015	0.0%	0.0%	0.0%	0.0%
LCC_mean	2018	6.7%	164.1%	-2.7%	-629.2%
	2019	17.2%	442.3%	-8.9%	-1489.1%
	2015	0.0%	0.0%	0.0%	0.0%
LCC_SD	2018	-0.3%	11.4%	-31.7%	-214.4%
	2019	-2.5%	13.0%	-100.1%	-589.5%
	2015	0.0%	0.0%	0.0%	0.0%
LCS_mean	2018	-4.1%	-95.7%	-101.2%	-373.3%

	2019	-5.7%	-105.1%	-139.8%	-527.9%
	2015	0.0%	0.0%	0.0%	0.0%
LCS_SD	2018	-6.4%	-39.9%	-36.5%	-192.4%
	2019	-9.0%	-78.6%	-43.6%	-181.6%
	2015	0.0%	0.0%	0.0%	0.0%
LCG_mean	2018	-4.4%	-104.7%	-111.6%	-138.9%
	2019	-7.6%	-146.4%	-233.8%	-389.8%
	2015	0.0%	0.0%	0.0%	0.0%
LCG_SD	2018	-5.0%	-61.6%	-48.7%	-110.4%
	2019	-9.7%	-118.4%	-103.2%	-287.3%
	2015	0.0%	0.0%	0.0%	0.0%
LCT_mean	2018	5.0%	282.9%	-6.1%	-487.7%
	2019	4.3%	245.3%	-5.8%	-420.7%
	2015	0.0%	0.0%	0.0%	0.0%
LCT_SD	2018	2.6%	42.3%	-31.1%	-397.1%
	2019	2.3%	42.3%	-32.7%	-391.1%
	2015	0.0%	0.0%	0.0%	0.0%
LCU_mean	2018	-2.6%	0.5%	-89.5%	-1475.1%
	2019	-2.4%	2.3%	-89.1%	-1514.5%
	2015	0.0%	0.0%	0.0%	0.0%
LCU_SD	2018	0.0%	-13.3%	26.2%	262.7%
	2019	0.1%	-11.7%	25.8%	251.5%

Table 15. *Table summarizing the change of feature values depending on the year for which the data was extracted.*

Appendix 4: Correlation Analysis of Independent Variables

The following figures show the full size output shown in the feature selection chapter. In general, the correlation plots look very similar and only change little year over year. However, changing the administrative level does have a noticeable effect on the correlation analysis. A similar situation was encountered doing the principle component analysis (PCA). Those findings support the use of smaller grids and more standardization, since it would eliminate those issues and facilitate preprocessing and feature selection.

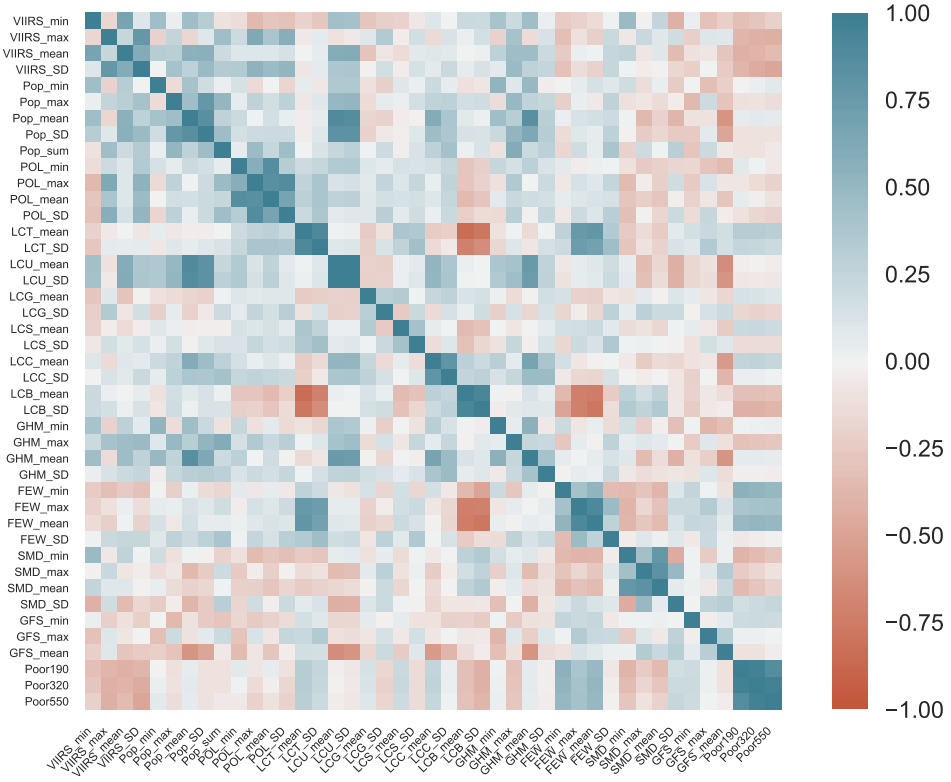


Figure 26. *Correlation Heatmap for the Data in 2015 in the mixed table obtained from World Bank.*

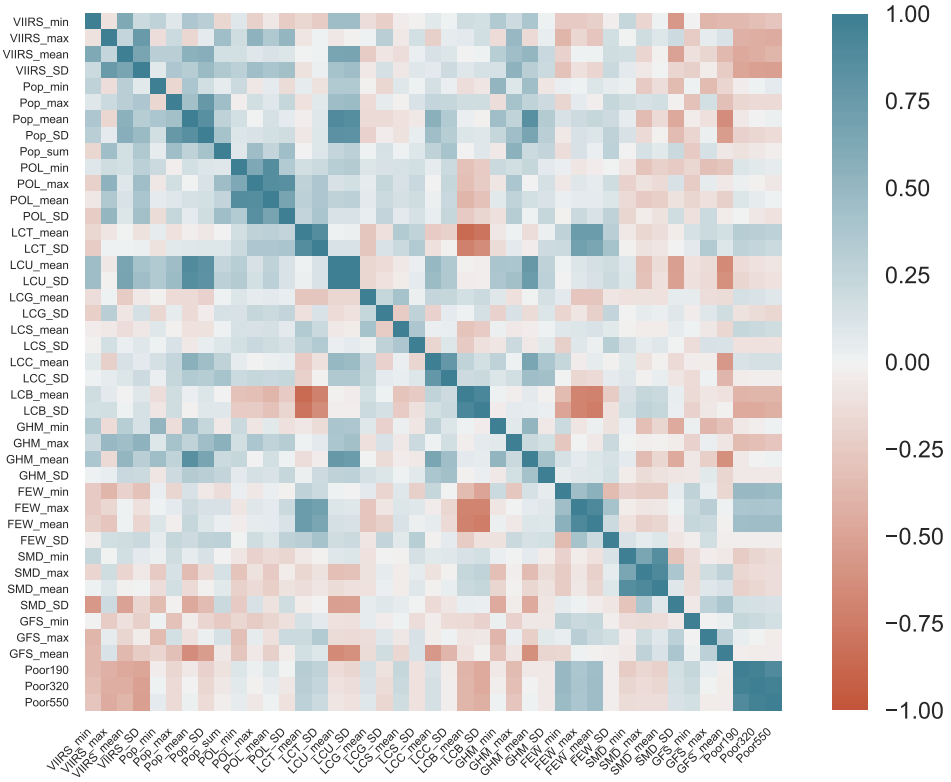


Figure 27. *Correlation Heatmap for the Data in 2018 in the mixed table obtained from World Bank.*

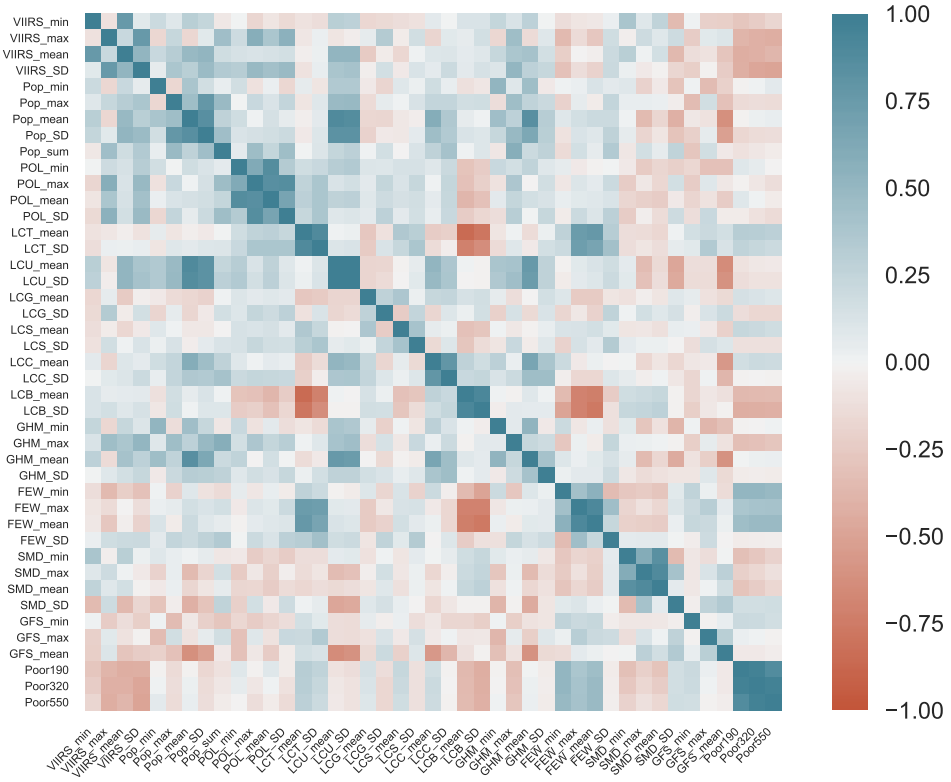


Figure 28. *Correlation Heatmap for the combined Data from 2015 and 2018 in the mixed tables obtained from World Bank.*

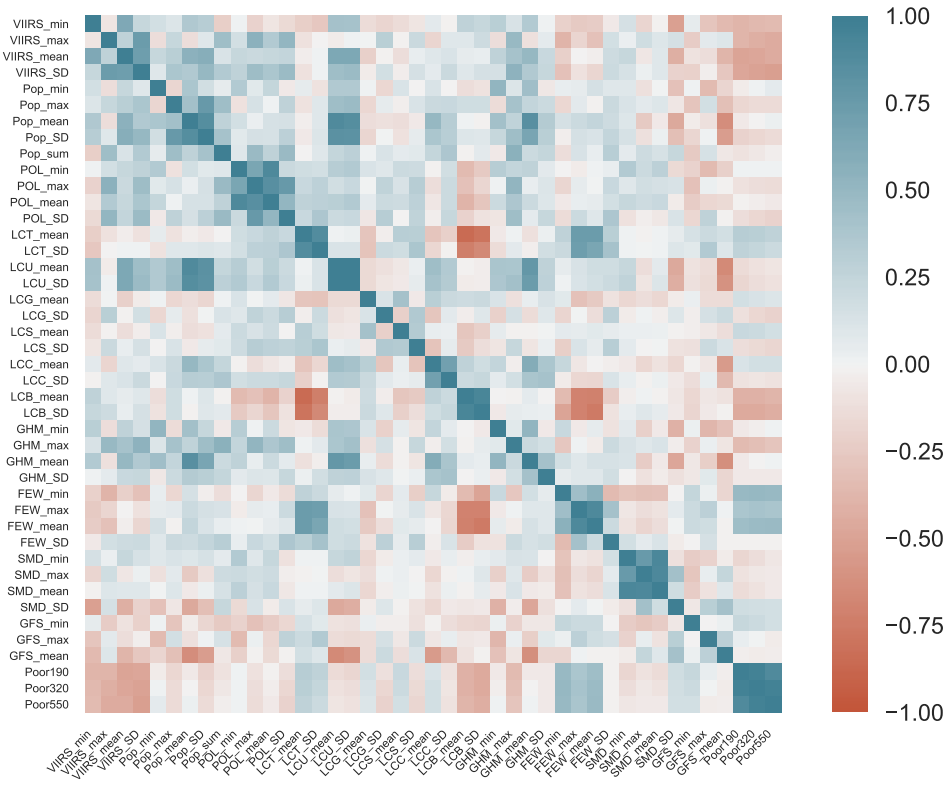


Figure 29. *Correlation Heatmap for the Data in 2019 in the mixed table obtained from World Bank.*

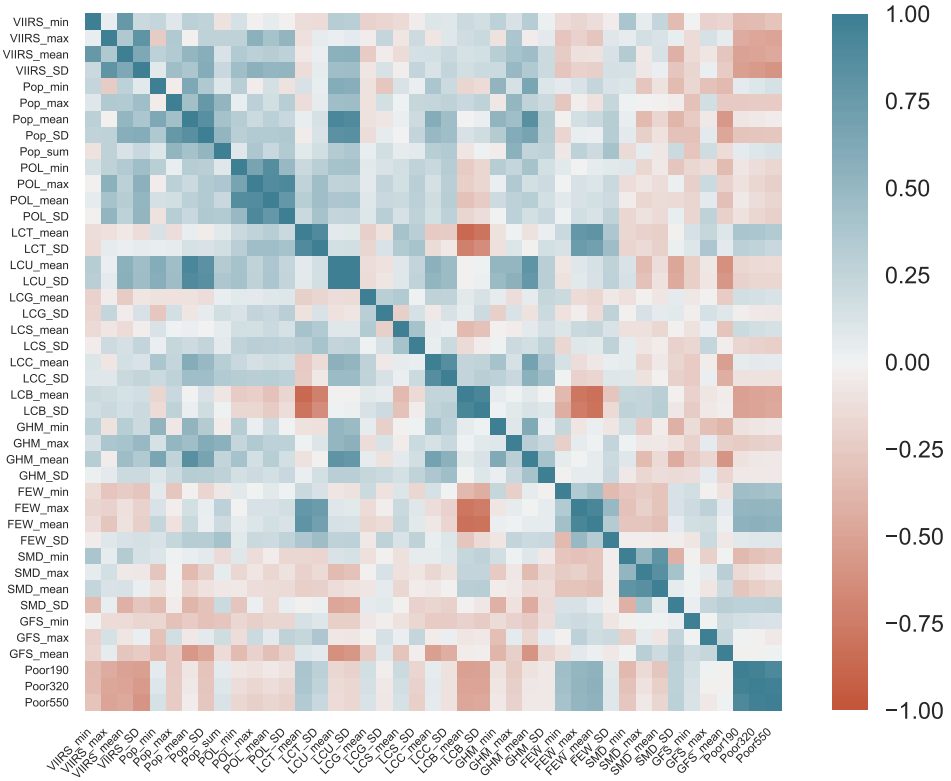


Figure 30. *Correlation Heatmap for the Admin 1 Data used in the Analysis.*

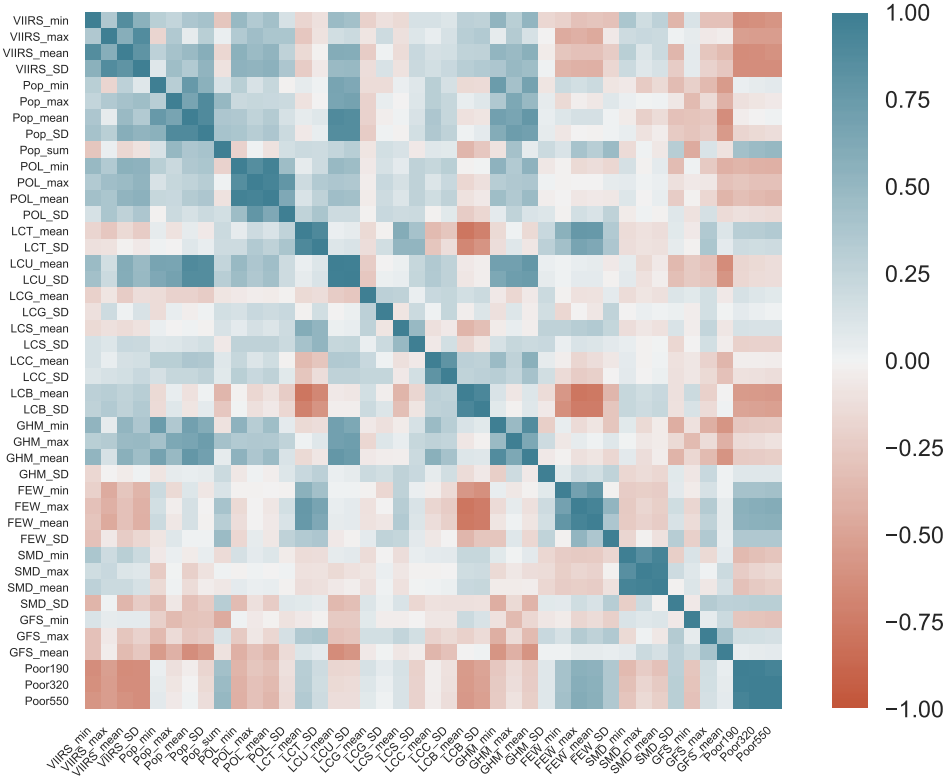


Figure 31. *Correlation Heatmap for the Admin 2 Data used in the Analysis.*

Appendix 5: Numeric Results

The following sections provide numerical results corresponding to the figures presented in the chapter solution evaluation.

Initial Results

The following table shows the testing results for the models after base model training and grid search.

Table 16. *Numeric Results in Testing Set after Initial Model Training*

	Poverty Level $1.90		Poverty Level $3.20		Poverty Level $5.50	
Model	R^2	MABS RPD	R^2	MABS RPD	R^2	MABS RPD
Kernel Ridge	0.42	0.23	0.42	0.25	0.45	0.15
Bayesian Ridge	0.16	0.27	0.17	0.33	0.16	0.23
XGB Regression	0.45	0.23	0.41	0.22	0.38	0.15
SV Regression	0.23	0.24	0.37	0.28	0.37	0.17
ANN Regression	0.52	0.22	0.49	0.29	0.35	0.16
Direct KRR	0.83	0.21	0.91	0.18	0.94	0.07
Direct BRR	0.52	0.27	0.62	0.32	0.65	0.18
Direct XGR	0.83	0.21	0.91	0.12	0.95	0.05
Direct SVR	0.77	0.23	0.88	0.23	0.92	0.09
Direct ANN	0.39	0.22	0.31	0.26	0.29	0.19
Stacked ANN	0.48	0.23	0.43	0.23	0.38	0.18

Transfer Learning to administrative level 1

The following table shows the numeric results for the testing set after transfer learning to improve poverty estimation at the first administrative level.

Table 17. *Numeric Results in Testing Set after Transfer Learning from Initially trained Models to Administrative Level 1 Predictions.*

	Poverty Level $1.90		Poverty Level $3.20		Poverty Level $5.50	
Model	R^2	MABS RPD	R^2	MABS RPD	R^2	MABS RPD
Kernel Ridge	0.87	0.09	0.88	0.08	0.82	0.07
Bayesian Ridge	0.36	0.19	0.40	0.20	0.43	0.20
XGB Regression	0.89	0.08	0.74	0.10	0.75	0.08
SV Regression	0.81	0.15	0.82	0.15	0.65	0.16
ANN Regression	0.61	0.15	0.61	0.17	0.71	0.11
Direct KRR	0.75	0.16	0.79	0.17	0.84	0.15
Direct BRR	0.49	0.20	0.61	0.20	0.62	0.20
Direct XGR	0.71	0.18	0.82	0.12	0.85	0.10
Direct SVR	0.69	0.17	0.77	0.16	0.75	0.16
Direct ANN	0.76	0.15	0.86	0.14	0.94	0.07
Stacked XGB	0.68	0.13	0.70	0.12	0.63	0.13
Stacked ANN	0.88	0.12	0.85	0.10	0.74	0.10

Transfer Learning to administrative level 2

The following table shows the numeric results for the testing set after transfer learning to improve poverty estimation at the second level of administration.

Table 18. *Numeric Results in Testing Set after Transfer Learning from Initially trained Models to Administrative Level 2 Predictions.*

	Poverty Level $1.90		Poverty Level $3.20		Poverty Level $5.50	
Model	R^2	MABS RPD	R^2	MABS RPD	R^2	MABS RPD
Kernel Ridge	0.48	0.23	0.43	0.25	0.46	0.15
Bayesian Ridge	0.22	0.29	0.21	0.35	0.18	0.23
XGB Regression	0.48	0.25	0.39	0.22	0.40	0.15
SV Regression	0.42	0.26	0.43	0.30	0.39	0.17
ANN Regression	0.59	0.22	0.47	0.31	0.49	0.15
Direct KRR	0.87	0.21	0.93	0.16	0.95	0.06
Direct BRR	0.51	0.28	0.63	0.33	0.65	0.18
Direct XGR	0.86	0.22	0.93	0.12	0.95	0.05
Direct SVR	0.81	0.25	0.90	0.24	0.92	0.09
Direct ANN	0.83	0.21	0.90	0.27	0.94	0.06
Stacked XGB	0.41	0.23	0.30	0.27	0.29	0.19
Stacked ANN	0.52	0.24	0.43	0.22	0.43	0.19

Testing Result on Mixed Table

This table shows testing results for 2019 assuming identical poverty rates to 2018.

Table 19. *Numeric Results in Testing Set of mixed format for the year 2019.*

	Poverty Level $1.90		Poverty Level $3.20		Poverty Level $5.50	
Model	R^2	MABS RPD	R^2	MABS RPD	R^2	MABS RPD
Kernel Ridge	0.72	0.12	0.81	0.09	0.79	0.06
Bayesian Ridge	0.40	0.17	0.50	0.14	0.47	0.09
XGB Regression	0.73	0.12	0.83	0.09	0.84	0.06
SV Regression	0.62	0.15	0.71	0.12	0.73	0.08
ANN Regression	0.61	0.13	0.65	0.13	0.78	0.06
Direct KRR	0.72	0.11	0.78	0.10	0.78	0.06
Direct BRR	0.36	0.17	0.45	0.15	0.42	0.10
Direct XGR	0.72	0.12	0.82	0.09	0.83	0.06
Direct SVR	0.61	0.15	0.70	0.13	0.71	0.08
Direct ANN	0.63	0.14	0.74	0.11	0.77	0.07
Stacked XGB	0.70	0.12	0.78	0.10	0.81	0.06
Stacked ANN	0.76	0.11	0.83	0.09	0.85	0.06

Testing Result on Admin 1 Table

This table shows the testing results for 2019 assuming identical poverty rates to 2018. Additionally, some data was aggregated to produce these results, meaning there are additional sources of errors.

Table 20. *Numeric Results in Testing Set of administrative level 1 format for the year 2019.*

	Poverty Level $1.90		Poverty Level $3.20		Poverty Level $5.50	
Model	R^2	MABS RPD	R^2	MABS RPD	R^2	MABS RPD
Kernel Ridge	0.74	0.10	0.80	0.08	0.82	0.05
Bayesian Ridge	0.47	0.15	0.54	0.12	0.50	0.08
XGB Regression	0.74	0.10	0.84	0.07	0.88	0.04
SV Regression	0.61	0.13	0.72	0.10	0.74	0.07
ANN Regression	0.62	0.12	0.64	0.11	0.78	0.05
Direct KRR	0.72	0.11	0.73	0.09	0.75	0.06
Direct BRR	0.42	0.15	0.42	0.14	0.32	0.10
Direct XGR	0.67	0.12	0.80	0.08	0.83	0.05
Direct SVR	0.63	0.13	0.67	0.11	0.68	0.07
Direct ANN	0.63	0.12	0.71	0.09	0.75	0.06
Stacked XGB	0.74	0.10	0.86	0.07	0.86	0.05
Stacked ANN	0.80	0.09	0.85	0.06	0.87	0.05

Testing Result on Admin 2 Table

This table shows the testing results for 2019 assuming identical poverty rates to 2018. Basically no training data was available for this kind of data. Most values are heavily aggregated, meaning there are additional sources of errors.

Table 21. *Numeric Results in Testing Set of administrative level 2 format for the year 2019.*

	Poverty Level $1.90		Poverty Level $3.20		Poverty Level $5.50	
Model	R^2	MABS RPD	R^2	MABS RPD	R^2	MABS RPD
Kernel Ridge	0.47	0.20	0.28	0.24	0.21	0.28
Bayesian Ridge	0.03	0.23	0.02	0.28	0.02	0.31
XGB Regression	0.37	0.20	0.28	0.24	0.20	0.28
SV Regression	0.33	0.21	0.25	0.24	0.18	0.29
ANN Regression	0.33	0.20	0.22	0.25	0.18	0.28
Direct KRR	0.47	0.20	0.35	0.24	0.20	0.28
Direct BRR	-0.03	0.24	-0.15	0.30	-0.07	0.34
Direct XGR	0.38	0.20	0.27	0.24	0.19	0.28
Direct SVR	0.25	0.21	0.19	0.25	0.17	0.29
Direct ANN	0.29	0.22	0.23	0.25	0.20	0.29
Stacked XGB	0.46	0.20	0.32	0.24	0.20	0.27
Stacked ANN	0.53	0.19	0.32	0.23	0.23	0.28

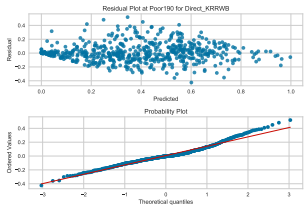
Appendix 6: Residual Analysis for all Testing sets

In this chapter residual plots for all developed models are presented. In general, the analysis of residuals depends on the poverty level. At the $1.90 level the assumption of approximately normally distributed residuals holds for some models. This is a good sign and allows us to give an estimate for the average prediction error. However, the higher the poverty level (higher income), the more likely it is to observe a cone shape in the residuals vs fits plot. This is caused by over-fitting and poorly distributed target values.

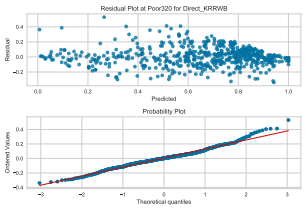
Kernel Ridge Regression

First, the KRR models using no additional information from the classifier are analyzed. As we can see in the figure 32, the KRR models for the original format and the administrative regions on the first level performed quite well. The Residuals at $1.90 and $3.20 are approximately normal distributed. However, for estimations at the second administrative level the assumption of homoscedasticity is clearly violated. There could be multiple reasons for this, most likely a combination of over-fitting, estimation outside of the calibration range and model sensitivity all play a part in the effects observed. Overall, KRR performed well. The estimation errors are among the smallest observed among all frameworks investigated and except for predictions at the second administrative level the approximately normal distribution of residuals and homoscedasticity support further investigation in this framework. Additionally, if the poverty estimates should be available at the $5.50 level, a transformation of target values might increase prediction accuracy and decrease the errors. At the least, it should help making the errors independent from the predicted value.

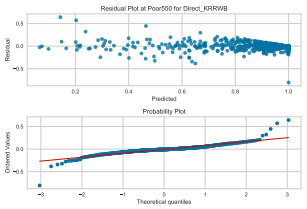
(a) *Original Format $1.90*



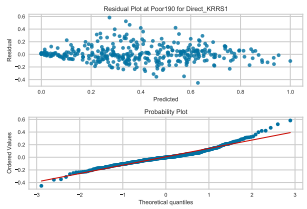
(b) *Original Format $3.20*



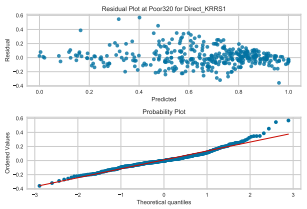
(c) *Original Format $5.50*



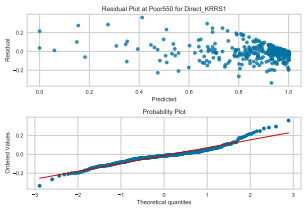
(d) *Admin 1 Format $1.90*



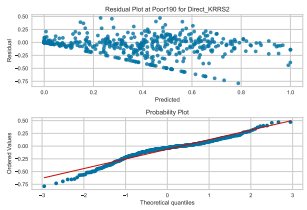
(e) *Admin 1 Format $3.20*



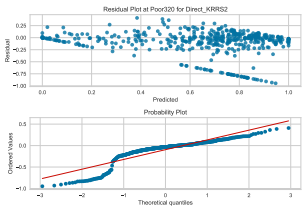
(f) *Admin 1 Format $5.50*



(g) *Admin 2 Format $1.90*



(h) *Admin 2 Format $3.20*



(i) *Admin 2 Format $5.50*

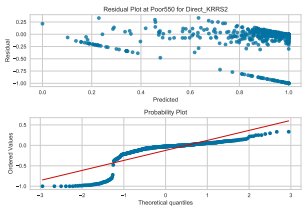
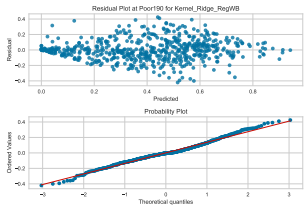


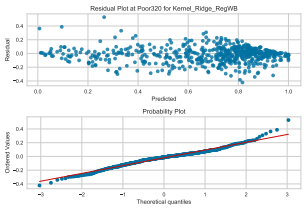
Figure 32. *Residual Analysis for the direct KRR Models*

Next, the model frameworks using the classifier information from preprocessing are analyzed. Compared to the previous analysis, the residuals show a little better homoscedasticity, however this again only holds for data at first administrative level and the format used for training. The level 2 predictions all have a clear cone shape, which surprisingly has a very particular line. This could be caused by numerical errors during the aggregation or an inadequate model, but the clear line shape indicates that it is not necessarily a trend that holds for all samples in the training set, but instead for a group of possible outliers.

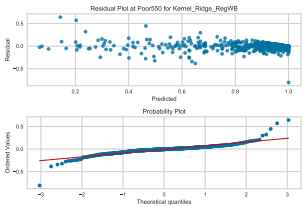
(a) *Original Format $1.90*



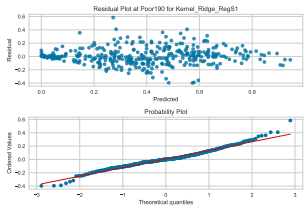
(b) *Original Format $3.20*



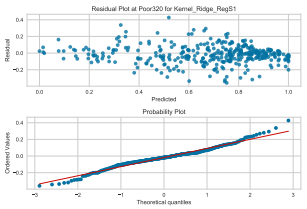
(c) *Original Format $5.50*



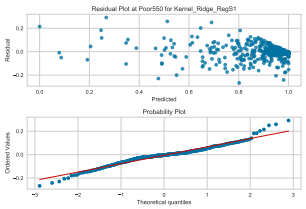
(d) *Admin 1 Format $1.90*



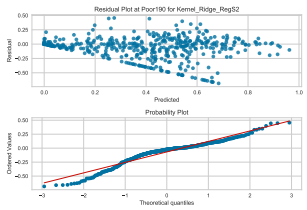
(e) *Admin 1 Format $3.20*



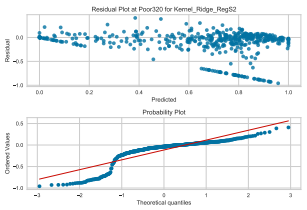
(f) *Admin 1 Format $5.50*



(g) *Admin 2 Format $1.90*



(h) *Admin 2 Format $3.20*



(i) *Admin 2 Format $5.50*

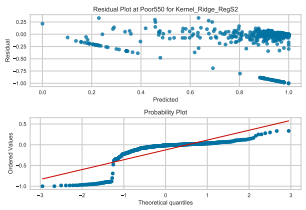
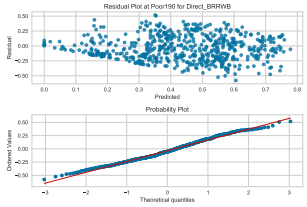


Figure 33. *Residual Analysis for the KRR Models*

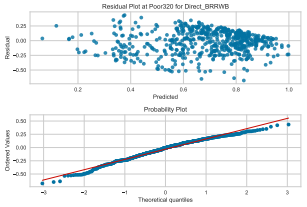
Bayesian Ridge Regression

During analysis of the Bayesian ridge regression models in the main body of the text it became evident that it is the least suited framework for poverty estimation with the tabular data investigated in this study. Still, in figure 34 the residuals for the testing sets in 2019 are presented. Contrary to the previous analysis, one can not assume approximately normal distribution of residuals or homoscedasticity for the examples presented below. This further underlines previous conclusions that BRR does not perform well and is ultimately not useful for this kind of problem.

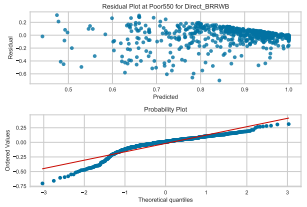
(a) *Original Format $1.90*



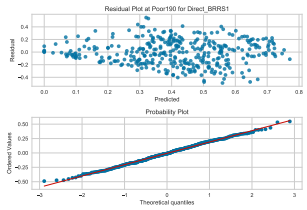
(b) *Original Format $3.20*



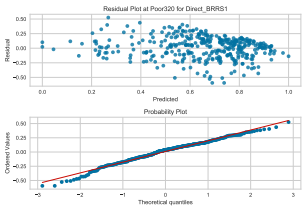
(c) *Original Format $5.50*



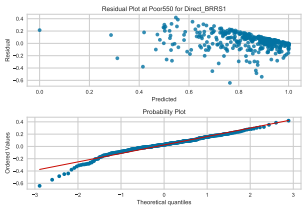
(d) *Admin 1 Format $1.90*



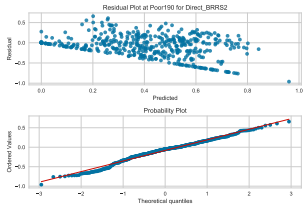
(e) *Admin 1 Format $3.20*



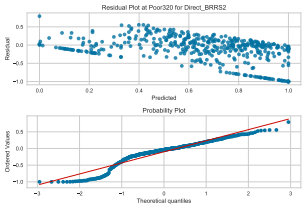
(f) *Admin 1 Format $5.50*



(g) *Admin 2 Format $1.90*



(h) *Admin 2 Format $3.20*



(i) *Admin 2 Format $5.50*

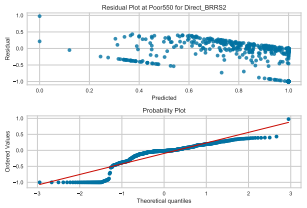
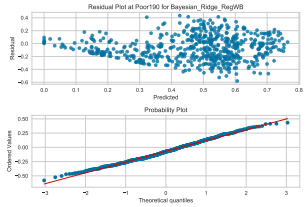


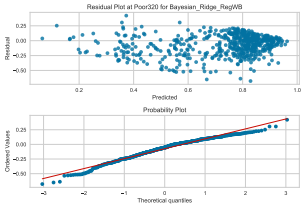
Figure 34. *Residual Analysis for the direct BRR Models*

Below, the identical analysis for models using information from the classifier is presented. We observe that the overall accuracy increases but the performance is not as good compared to any of the other models.

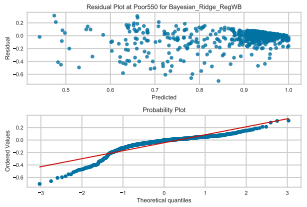
(a) *Original Format $1.90*



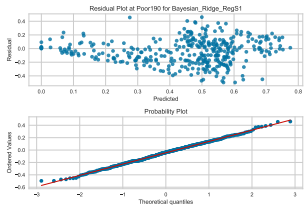
(b) *Original Format $3.20*



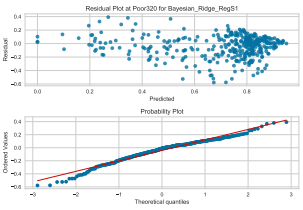
(c) *Original Format $5.50*



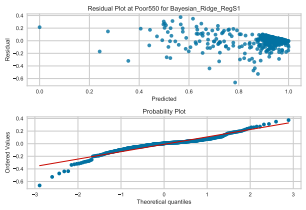
(d) *Admin 1 Format $1.90*



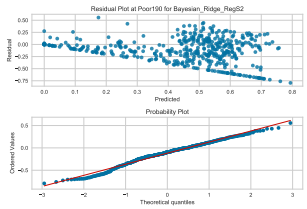
(e) *Admin 1 Format $3.20*



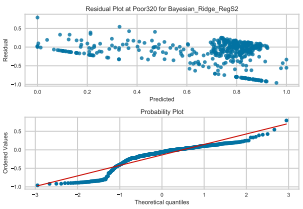
(f) *Admin 1 Format $5.50*



(g) *Admin 2 Format $1.90*



(h) *Admin 2 Format $3.20*



(i) *Admin 2 Format $5.50*

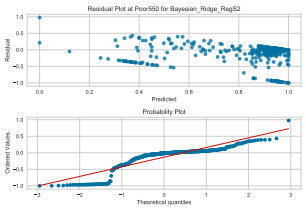
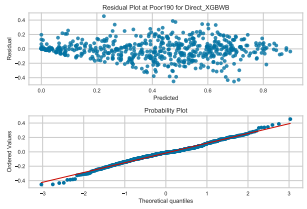


Figure 35. *Residual Analysis for the BRR Models*

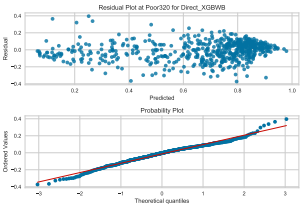
XGBoost Regression

The XGBoost framework was used for multiple steps in this study. First, the analysis of residuals for the model using all samples in its set is presented in figure 38:

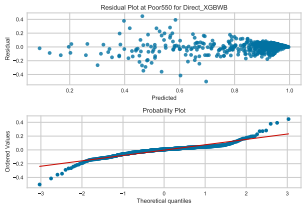
(a) *Original Format $1.90*



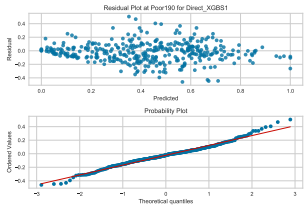
(b) *Original Format $3.20*



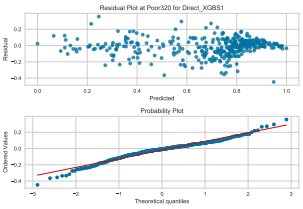
(c) *Original Format $5.50*



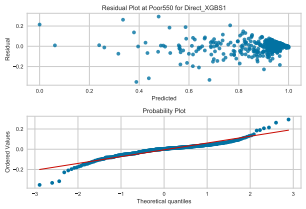
(d) *Admin 1 Format $1.90*



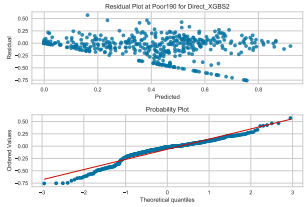
(e) *Admin 1 Format $3.20*



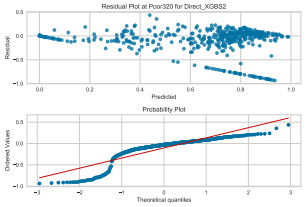
(f) *Admin 1 Format $5.50*



(g) *Admin 2 Format $1.90*



(h) *Admin 2 Format $3.20*



(i) *Admin 2 Format $5.50*

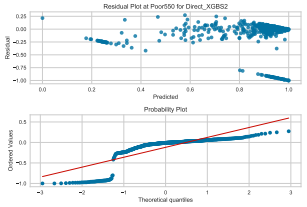
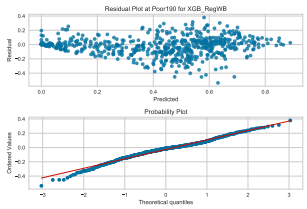


Figure 36. *Residual Analysis for the direct XGB Models*

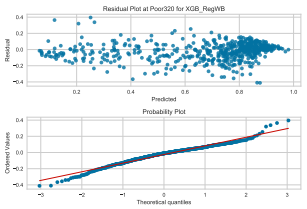
Similar to previous conclusions, the XGB regressor using no information from the classifier performs okay on the first administrative level but the accuracy is quite low for the second administrative level. Again, there seems to be a group of samples that the model can not explain.

Below, the identical analysis for models using information from the classifier is presented. We observe that contrary to previous examples the XGB framework does not perform better when using information from the classifier.

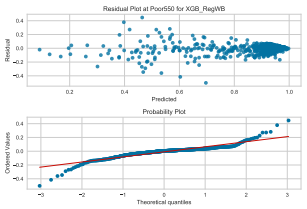
(a) *Original Format $1.90*



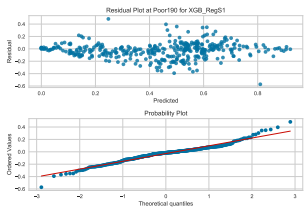
(b) *Original Format $3.20*



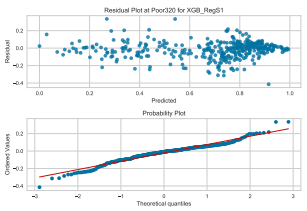
(c) *Original Format $5.50*



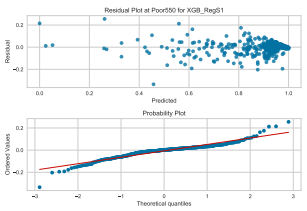
(d) *Admin 1 Format $1.90*



(e) *Admin 1 Format $3.20*



(f) *Admin 1 Format $5.50*



(g) *Admin 2 Format $1.90* (h) *Admin 2 Format $3.20* (i) *Admin 2 Format $5.50*

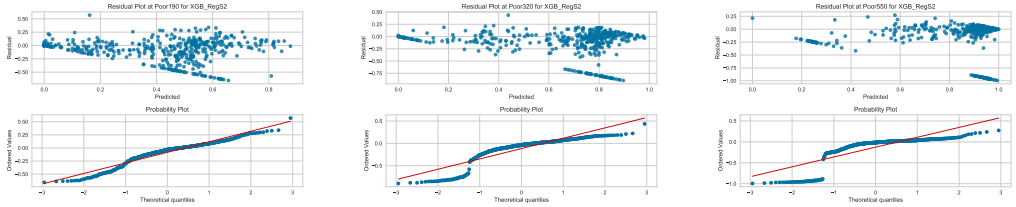


Figure 37. *Residual Analysis for the XGB Models*

Additionally to the previous models, the XGB framework was also used to build a stacked ensemble model. Below, the residuals for that analysis are presented.

(a) *Original Format $1.90* (b) *Original Format $3.20* (c) *Original Format $5.50*

(d) *Admin 1 Format $1.90* (e) *Admin 1 Format $3.20* (f) *Admin 1 Format $5.50*

(g) *Admin 2 Format $1.90* (h) *Admin 2 Format $3.20* (i) *Admin 2 Format $5.50*

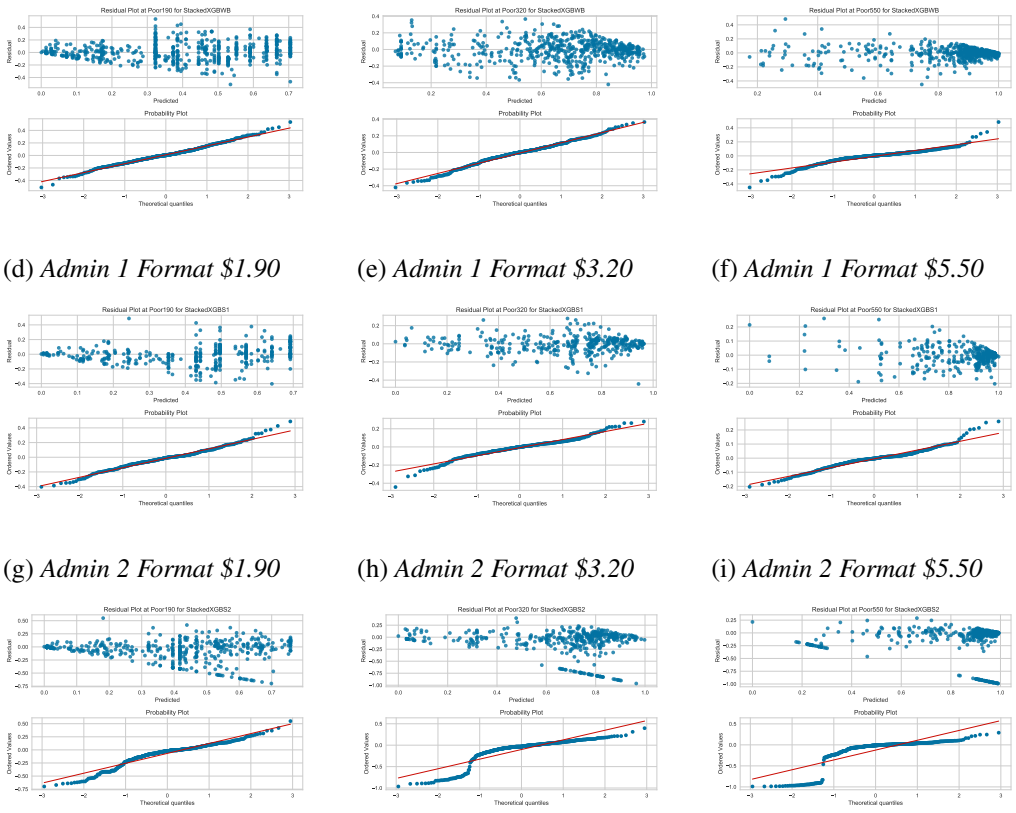


Figure 38. *Residual Analysis for the ensemble XGB Models*

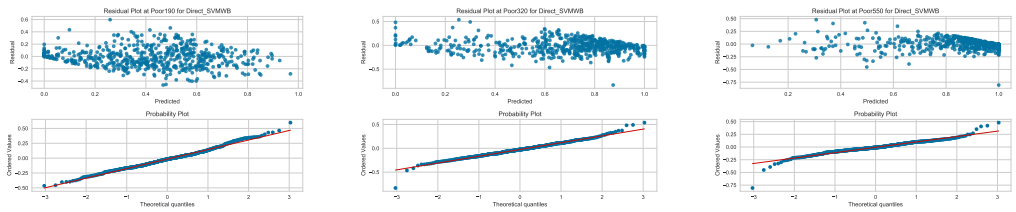
As we can see above, the ensemble model has overall smaller errors compared to the previous models. However, for the second administrative level we again observe the previously discussed trends, possibly caused by numerical errors

during aggregation or inadequate predictions. Additionally, there can a clear cone shape be observed, which makes error estimation and model fitting less accurate. For the other two models, there can be certain patterns observed, however there are only serious limitations for the estimations at the $5.50 level for the first administrative level.

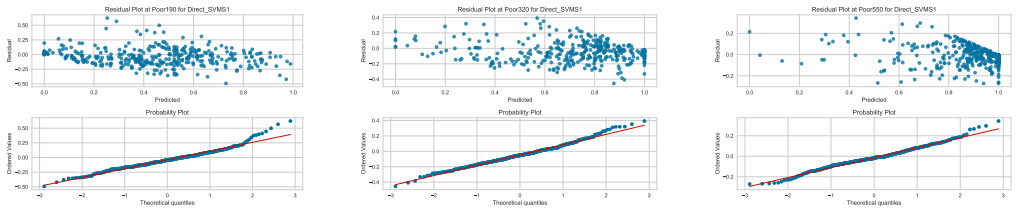
Support Vector Regression

Below, the residual analysis for all developped support vector regression models are presesented. The takeaways are the same as for KRR.

(a) *Original Format $1.90* (b) *Original Format $3.20* (c) *Original Format $5.50*



(d) *Admin 1 Format $1.90* (e) *Admin 1 Format $3.20* (f) *Admin 1 Format $5.50*



(g) *Admin 2 Format $1.90* (h) *Admin 2 Format $3.20* (i) *Admin 2 Format $5.50*

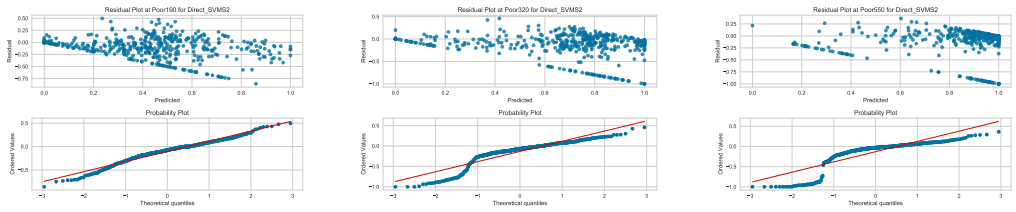
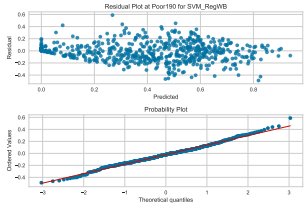


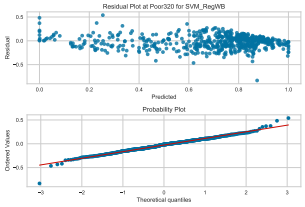
Figure 39. *Residual Analysis for the direct SVR Models*

Below, the analysis for the models relying on information from the classifier are presented. The errors are a little smaller compared to the direct models.

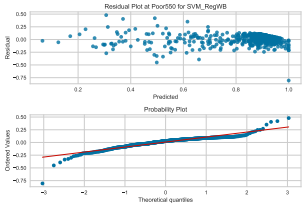
(a) *Original Format $1.90*



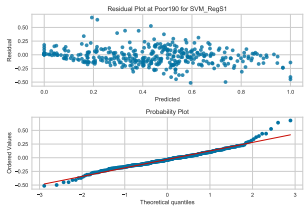
(b) *Original Format $3.20*



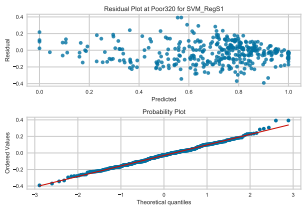
(c) *Original Format $5.50*



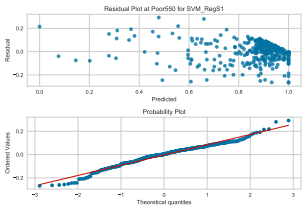
(d) *Admin 1 Format $1.90*



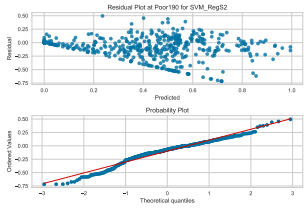
(e) *Admin 1 Format $3.20*



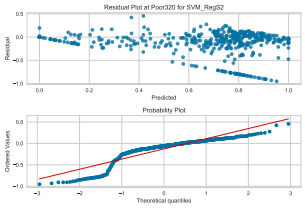
(f) *Admin 1 Format $5.50*



(g) *Admin 2 Format $1.90*



(h) *Admin 2 Format $3.20*



(i) *Admin 2 Format $5.50*

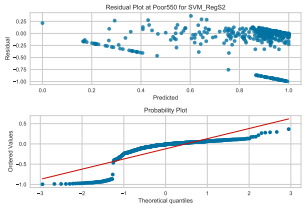


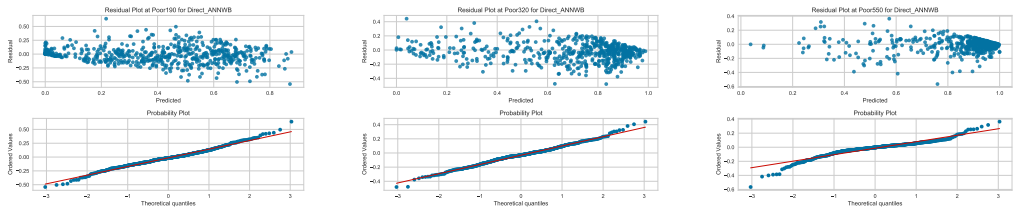
Figure 40. *Residual Analysis for the SVR Models*

Neural Network Regression

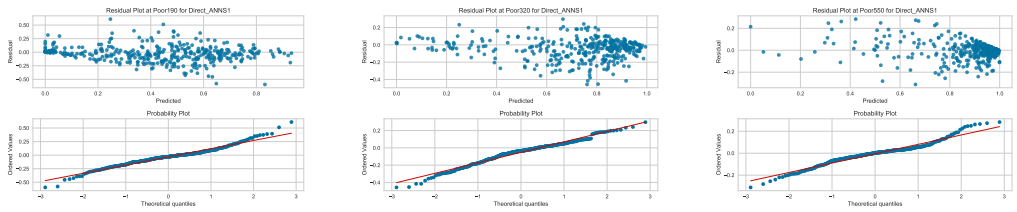
Lastly, we go through the residual analysis for the developed neural networks. Their method of estimation works completely different from the other examples, which has an effect on the residuals. All models use a sigmoid activation function in their output layer. This intrinsically limits the range of estimated poverty rates between 0 and 1. Therefore, at high values of y_{pred} the residuals can not be positive, and at small values they can not be negative. This limits the error but inherently makes normal distribution and homoscedasticity impossible. It was observed that the generalization for estimations at the second administrative level seems comparably high. Compared to other models there are less samples for which there is a distinct trend of error. Nonetheless, if those estimations are to be used for poverty maps there is still more work to be done. I suspect the main issue with the inaccuracies at the second level is a

lack of reliable way to clean the input data and identify samples that match with the training set. One could use deep learning frameworks like classifiers to improve upon these issues.

(a) *Original Format $1.90* (b) *Original Format $3.20* (c) *Original Format $5.50*



(d) *Admin 1 Format $1.90* (e) *Admin 1 Format $3.20* (f) *Admin 1 Format $5.50*



(g) *Admin 2 Format $1.90* (h) *Admin 2 Format $3.20* (i) *Admin 2 Format $5.50*

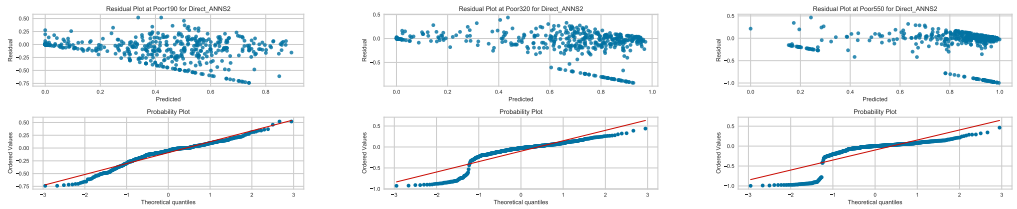
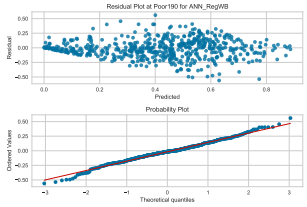


Figure 41. *Residual Analysis for the direct ANN Models*

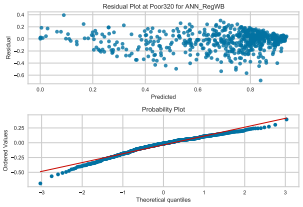
Contrary to other examples, the direct neural network did not produce very accurate estimations at the first administrative level at the poverty rate $5.50. There is a clear cone shape and the models seems to have over-fit and usually generates estimates around the mean value. This is further supported by the fact that the models for this poverty level do not use any dropout layers during training, which is a powerful regularization technique.

However, this situation changes for the models using the classifier from the preprocessing.

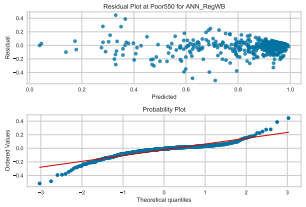
(a) *Original Format $1.90*



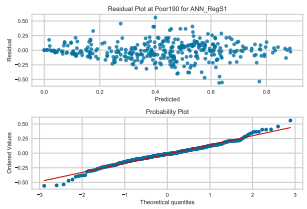
(b) *Original Format $3.20*



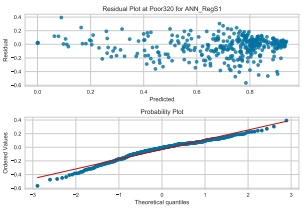
(c) *Original Format $5.50*



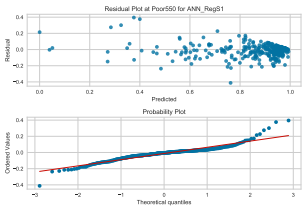
(d) *Admin 1 Format $1.90*



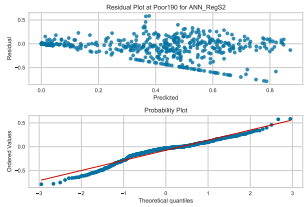
(e) *Admin 1 Format $3.20*



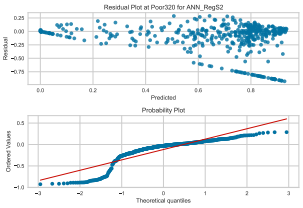
(f) *Admin 1 Format $5.50*



(g) *Admin 2 Format $1.90*



(h) *Admin 2 Format $3.20*



(i) *Admin 2 Format $5.50*

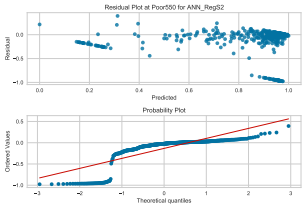
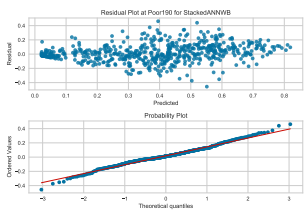


Figure 42. *Residual Analysis for the ANN Models*

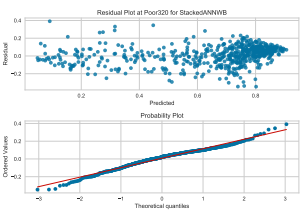
As we can see above, the accuracy has increased dramatically for estimations at the first administrative layer. Compared to before, there are also less outliers in the residuals. This does not hold for estimations at the second administrative level, however the estimations at the $1.90 level are better than for any other model.

Like the XGB framework, there was also an ensemble model developed which is analyzed below.

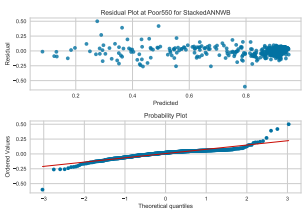
(a) *Original Format $1.90*



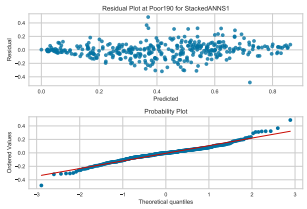
(b) *Original Format $3.20*



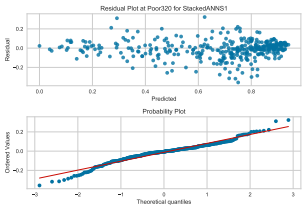
(c) *Original Format $5.50*



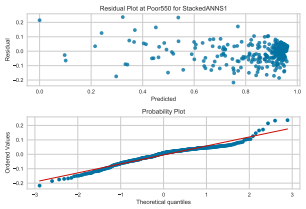
(d) *Admin 1 Format $1.90*



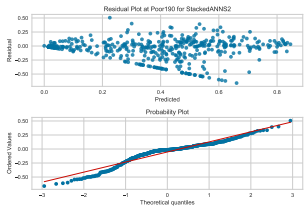
(e) *Admin 1 Format $3.20*



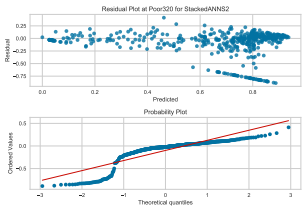
(f) *Admin 1 Format $5.50*



(g) *Admin 2 Format $1.90*



(h) *Admin 2 Format $3.20*



(i) *Admin 2 Format $5.50*

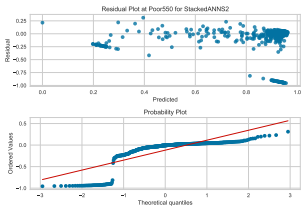


Figure 43. *Residual Analysis for the ensemble ANN Models*

The developed ensemble method produces quite accurate results. Compared to previously analyzed frameworks the estimations at the second administrative level are more accurate and there are fewer large and grouped residuals. This means that neural networks are possibly suited for a top-down poverty estimation method with a lack of training data.

However, I suggest looking into bottom-up approaches with aggregated data on grids, as further specified in future directions.

Appendix 7: ANN Model Information

This chapter contains the complete summary of all 15 neural network algorithms developed in the process of this study. All algorithms have one additional layer that is not mentioned in the tables. This layer contains only one node and uses a sigmoid activation to produce the final poverty estimate. This activation function is limited between 0 and 1, which is the same range as the poverty rates. First, the three direct models are analyzed.

Table 22. *Summary of the optimized hyperparameters for the direct neural networks.*

	Direct at $1.90	Direct at $3.20	Direct at $5.50
Layer0	24	24	28
Dropout	2%	-	-
Regularization	-	-	-
Activation	relu	relu	relu
Layer1	14	8	16
Dropout	18%	8%	-
Regularization	-	-	l1
reg_value1	-	-	0.001
Activation1	selu	selu	selu

All models above use some type of regularization. The model at $1.90 uses two dropout layers after both hidden layers. The model at the next higher income threshold has its dropout layer before the final node which generates the poverty estimation, and the model at the highest income level uses l1 regularization. Interestingly, the regularization strength decreases with higher income thresholds for the poverty level. This goes hand in hand with previous observations that the models at the higher income level have a much higher tendency towards biased results at the $5.50 level because of high poverty rates.

Next, the tables for the two-step models is presented. Contrary to the direct models, most two-step models use little to no regularization. This explains their worse performance compared to the direct models previously analyzed.

Table 23. *Summary of the optimized hyperparameters for the two-step neural network models in poor areas.*

	Poor at $1.90	Poor at $3.20	Poor at $5.50
Layer0	26	28	26
Dropout	-	-	-
Regularization	-	-	-
Activation	relu	relu	relu
Layer1	6	2	8
Dropout	-	-	-
Regularization	-	-	-
reg_value1	-	-	-
Activation1	selu	tanh	tanh

None of the models estimating poverty in poor areas use any type of regularization. Since the model structure is similar to the direct models, this indicates a possibility for sensitive model parameters and could explain the lower accuracy of predictions obtained with these models. Contrary to the models developed for the poor areas, the ones for non-poor areas all take advantage of some kind of regularization to limit over-fitting:

Table 24. *Summary of the optimized hyperparameters for the two-step neural network models in non-poor areas.*

	non-Poor at $1.90	non-Poor at $3.20	non-Poor at $5.50
Layer0	28	28	28
Dropout	9%	-	6%
Regularization	-	-	-
Activation	relu	relu	relu
Layer1	6	24	6
Dropout	-	15%	-
Regularization	-	l1_l2	l1_l2
reg_value1	-	0.1	0.1
Activation1	selu	sigmoid	selu

As we can see above, all models for non-poor areas use regularization. Since those algorithms were trained using the same optimization framework as before, this indicates that there was more variability in the non-poor areas. Therefore, the optimization framework developed models that are better at generalization for non-poor areas, while the models specialized for poor areas are given more freedom at the cost of a higher risk of sensitive model parameters. In any case, these models do not perform as well as others developed in the course of this work.

Ultimately, the stacked neural networks are analyzed. They take inputs from all two-step models and generate a final poverty estimate and are therefore an example of an ensemble method. The following table shows the stacked model parameters found during optimization which takes the estimates of areas that were classified as poor during preprocessing. Compared to the previous neural

Table 25. *Summary of the optimized hyperparameters for the stacked neural network models in poor areas.*

	st-Poor at $1.90	st-Poor at $3.20	st-Poor at $5.50
Layer0	9	12	12
Dropout	-	-	-
Regularization	-	-	-
Activation	selu	selu	selu
Layer1	3	10	-
Dropout	15%	15%	-
Regularization	l2	-	l1
reg_value1	1	-	1
Activation1	tanh	sigmoid	tanh

networks, these models are much smaller with less nodes and sometimes just a single layer. This makes sense, since only five input values are given to these models. Nonetheless, they all use regularization for better generalization ability.

The last models under investigation are the stacked neural networks for non-poor areas. Like before, smaller model architectures compared to the direct and two-step models are observed. Except for the model at the income level of $3.20 the models use regularization methods.

Table 26. *Summary of the optimized hyperparameters for the stacked neural network models in non-poor areas.*

	st-non-Poor at $1.90	st-non-Poor at $3.20	st-non-Poor at $5.50
Layer0	10	12	11
Dropout	-	-	-
Regularization	-	-	-
Activation	selu	selu	selu
Layer1	-	9	7
Dropout	-	-	14%
Regularization	l1_l1	-	-
reg_value1	0.01	-	-
Activation1	selu	sigmoid	relu