THESIS WORK FOR DUAL MASTER'S DEGREE

ITBA Mag. in Energy and Environment

KIT M.Sc. in Mechanical Engineering

MACHINE LEARNING FOR SPATIAL DISAGGREGATION OF REGIONAL TRANSPORT DATA IN THE EU

Juan R. Fernandez
Chemical Engineer
Universidad Tecnológica Nacional

Tutor
Dr.-Ing Cihan Ates, KIT

Co-Tutor
Shruthi, Patil, Forschungszentrum Jülich

Examiners
Prof. Dr.-Ing. H.-J. Bauer, Ord. KIT / ITS
Dr.-Ing. Rainer Koch KIT/ ITS

Karlsruhe
23/06/2023

I assure that I have written the work independently and have only used the sources and aids indicated. The places that have been adopted literally or in terms of content are clearly marked. I have observed the statutes of the Karlsruhe Institute of Technology (KIT) for ensuring good scientific practice in their current version.

Karlsruhe, June 23, 2023

Acknowledgments

I would like to take this opportunity to express my deepest gratitude to those who have been instrumental in the completion of this thesis.

Firstly, I want to thank my family for their unwavering support throughout this journey. A special mention goes to my mother, Maymara Brugnoli, for her constant encouragement and belief in me. Her resilience and determination have been an inspiration, shaping my values and giving me the strength to persevere through challenges. Without her dedication and sacrifices, my pursuit of education would not have been possible.

I also extend my appreciation to my grandparents - Erica, Norma, Hualpa, and Juan. Their enduring presence and profound influence have been pillars of strength, shaping me into the person I am today.

My heartfelt thanks go to the Kantor family - Lisa, Todd, Ethan, Ross, and Drew. They have played a significant role in my life in recent years. Their support was invaluable, enabling me to study and travel to Europe. Their humble and respectful nature fostered a strong sense of belonging within me.

I would like to express my profound gratitude to Forschungszentrum Jülich for providing the opportunity to work on this important project. The experience and knowledge gained have been invaluable, and it was a privilege to collaborate with such a respected institution. A special acknowledgment goes to Shrithi Patil, my FZJ supervisor, who was fundamental in my learning process. Her teachings, extending from Git control to the best principles of coding, have significantly contributed to my development. She invested a lot of time in me and worked diligently, for which I am deeply grateful.

I am deeply thankful to my supervisor and mentor, Professor Cihan Ates, who provided indispensable guidance throughout this journey. His invaluable advice, support, and encouragement were instrumental in the completion of this thesis. Our lengthy discussions in his office were not only enlightening but also inspiring. I couldn't have accomplished this without his motivation and collaboration.

Lastly, this thesis is dedicated to my seven younger siblings - Cora, Lucrecia, Juan Ignacio, Salvador, Juana, Antonia, and Valentino. I hope this work serves as a testament to the power of hard work, determination, and the support of loved ones.

Thank you all for being a part of my journey.

Contents

Nomenclature

The Acronyms and Nomenclature section provides an essential reference for the reader. In this section, abbreviations and specific symbols used throughout the document are clearly defined. The objective is to enhance comprehension and accessibility of the content, ensuring it can be understood by readers from diverse backgrounds.

0.1 General Abbreviations and Acronyms

Abbreviations	
EU	European Union
GHG	Greenhouse Gas
EEA	European Environment Agency
LAU	Local Administrative Units
NUTS	Nomenclature of Territorial Units for Statistics
Eurostat	Statistical Office of the European Union
DLR	German Aerospace Center
IfV	Institute for Transport Studies
KIT	Karlsruhe Institute of Technology
FZJ	Forschungszentrum Jülich
NACE	Nomenclature générale des Activités économiques dans les Communautés Européennes (General Classification of Economic Activities in the European Communities)
GDP	Gross Domestic Product
GVA	Gross Value Added
CSV	Comma-Separated Values
ICE	Internal Combustion Engine
EV	Electric Vehicle
NaN	Not a Number
API	Application Programming Interface
GIS	Geographic Information System
GME	Generalized Maximum Entropy
IDM	Intelligent Dasymetric Mapping
PC	Population Census
CORINE	Coordination of Information on the Environment

Acronyms	
COVID-19	Coronavirus Disease 2019
OSM	OpenStreetMap
OSMnx	OpenStreetMap NetworkX
ETISplus	European Transport Policy Information System Plus
IEK-3	Techno-economic Systems Analysis Institute

0.2 Machine Learning and Statistics Abbreviations and Acronyms

Abbreviations

PCA	Principal Component Analysis
UMAP	Uniform Manifold Approximation and Projection
SSR	Sum of Squares for Regression
SSE	Sum of Squares for the Residual Error
MI	Mutual Information
RSS	Residual Sum of Squares
OLS	Ordinary Least Squares
L2	L2 Regularization (Ridge Regression)
CV	Cross-Validation

Acronyms

ML	Machine Learning
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
t-SNE	t-Distributed Stochastic Neighbor Embedding
F-Regression	F-Statistic Regression
MLP	Multi-Layer Perceptron
R-squared	Coefficient of determination
ANN	Artificial Neural Network
RF	Random Forest
SVM	Support Vector Machine
CNN	Convolutional Neural Network
ReLU	Rectified Linear Unit
NN	Neural Network
RBF	Radial Basis Function
SVR	Support Vector Regression
XGBoost	Extreme Gradient Boosting
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
R2	R-squared
GCN	Graph Convolutional Network

Machine Learning and Statistics Terms

SSR	Sum of Squares for Regression
SSE	Sum of Squares for the Residual Error
RSS	Residual Sum of Squares
MI	Mutual Information
F-Regression	Univariate Feature Selection: F-Regression
Mutual Information	Univariate Feature Selection: Mutual Information
Ridge	Feature Selection using Ridge (L2 Regularization)
Random Forest Importance	Feature Selection using Random Forest Importance

List of Figures

List of Tables

1 Introduction

The European Union (EU) is actively working to combat climate change and promote sustainable development by reducing greenhouse gas (GHG) emissions. The transport sector, a major contributor of GHG emissions, was at the forefront of these initiatives. After experiencing steady growth from 2013 until 2019, there was an abrupt decrease in 2020 due to the COVID-19 pandemic. However, preliminary estimates indicated a rebound of 7.7% for transport emissions in 2021, according to the [Agency (2021)]. Nonetheless, further research is necessary in order to devise effective strategies for regional decarbonization within this challenging sector.

An analysis of the transport sector in Europe reveals significant disparities in emission trends across different regions. According to [Eurostat (2021)], Western European countries have generally experienced greater decreases in transport emissions compared to Central and Eastern European nations, which have made slower progress. Furthermore, the European Environment Agency [Agency (2021)] points out that urban areas tend to have higher emissions due to higher population densities and greater demand for transportation. These discrepancies underscore the necessity for spatial disaggregation when developing tailored decarbonization strategies for different regions.

To address the intricacies of regional decarbonization potentials, this research aims to apply machine learning techniques to enhance the accuracy of estimating transport-related metrics at a regional level. This, in turn, will facilitate the identification of decarbonization opportunities within the transport sector. More precisely, the study seeks to establish a framework that utilizes machine learning methodologies for spatial disaggregation, a critical process for understanding the factors that influence emissions on a regional scale and devising efficient mitigation strategies.

1.1 Motivation

1.1.1 Problem Definition

The key challenge in estimating regional decarbonization potentials in transport is the accurate spatial disaggregation of transportation-related data. Without ground truth data at each target resolution, especially at the district level, the precision and accuracy of machine learning techniques are limited. The vast geographic area of Europe and the diversity of its data sources further complicate the acquisition of comprehensive and precise information across all member countries.

The goal of this research is to identify the most accurate machine learning method for disaggregating data from the country level to the district level. This method needs to take into account heterogeneous data, spatial dependencies, and interactions between transport-related information and ancillary data. The aim is to discover a model capable of providing reliable transport-related estimates across all European Union member states - specifically 1170 NUTS3 regions - by investigating various machine learning algorithms, assessing their performance, and selecting the most precise one for each region.

1.1.2 Proposed Solution

To tackle the challenge of spatial disaggregation, this study proposes creating a Master Model for each nation based on data from 26 other EU members. This approach utilizes the hybrid disaggregation method developed by [Monteiro et al. (2020)], as explained in the Chapter 3. This technique employs self-supervised regression techniques, which draw insights directly from available data while iteratively refining outcomes through training a regression model, thereby improving the accuracy of results.

The Master Model will be trained using a Self-Supervised deep learning algorithm capable of detecting complex patterns and relationships in large datasets. This algorithm considers nonlinear relationships between variables as well as spatial autocorrelation in geospatial data. By including features that encapsulate spatial relationships, the regression model can generate precise estimates for continuous variables, even without complete or accurate ground truth data at every resolution.

The accuracy of the Master Model will be tested using the number of charging and train stations in three countries: Portugal, Germany, and Austria. This evaluation will determine whether the self-supervised methodology can outperform alternative methods that lack ground truth data, and offer more precise spatial disaggregation solutions.

1.2 Research Goal

The goal of this research is to enhance the accuracy and efficiency of spatial disaggregation techniques for transport-related data, with potential applications in the Localised project. The primary focus of this study is on exploring machine learning-based dasymetric weighting schemes combined with ancillary information. The objective is to achieve accurate disaggregation and estimation across NUTS-3 regions in the European Union.

Dasymetric weighting, a method of spatial disaggregation, redistributes aggregated data from coarser spatial units (e.g., administrative boundaries) to finer spatial units (e.g., grid cells) using ancillary information [Eicher und Brewer (2001a)]. This method improves the accuracy of disaggregated data by taking into account underlying variables such as population density or land use that are correlated with the target variable, in this case, transport-related data.

NUTS (Nomenclature of Territorial Units for Statistics) is a hierarchical classification system used by Eurostat [Eurostat (2021)], the statistical office of the European Union, to collect and publish regional statistics across Europe. NUTS divides each EU country into several levels: from NUTS-0 at the country level down to LAUs or Local Administrative Units. Each member state defines its own LAU territorial units, but these must be compatible with NUTS, as this system has been adopted by all member states within the European Union.

This thesis examines the number of regions within 27 European Union member countries, each of which encompasses a different number of units in each region.

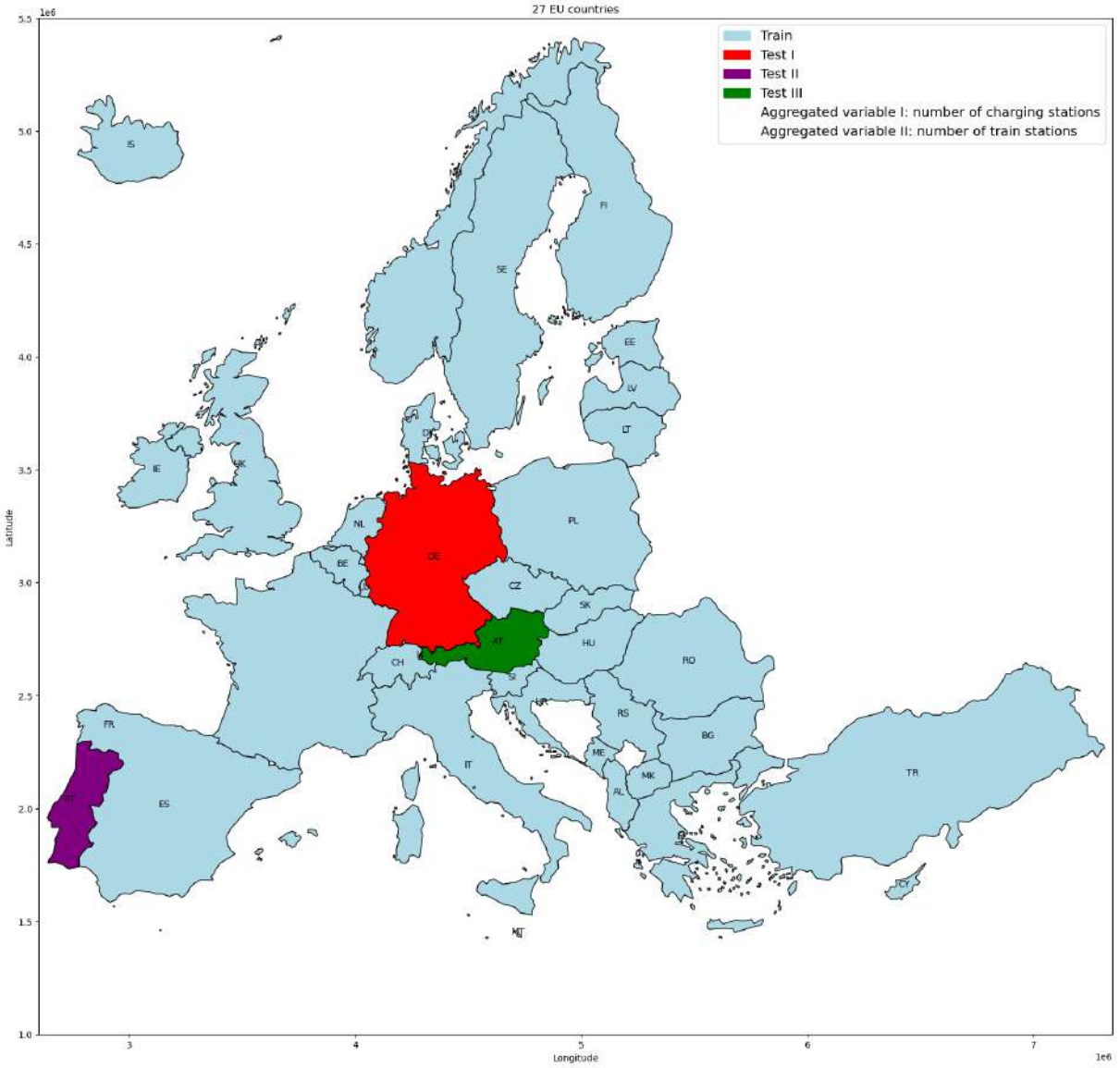


Figure 1.1: Training and Testing Data.

Spatial Resolution Unit	Number of Units
LAU	95743
NUTS-3	1170
NUTS-2	240
NUTS-1	92
NUTS-0 (Countries)	27

Table 1.1: Number of units per region in the EU.

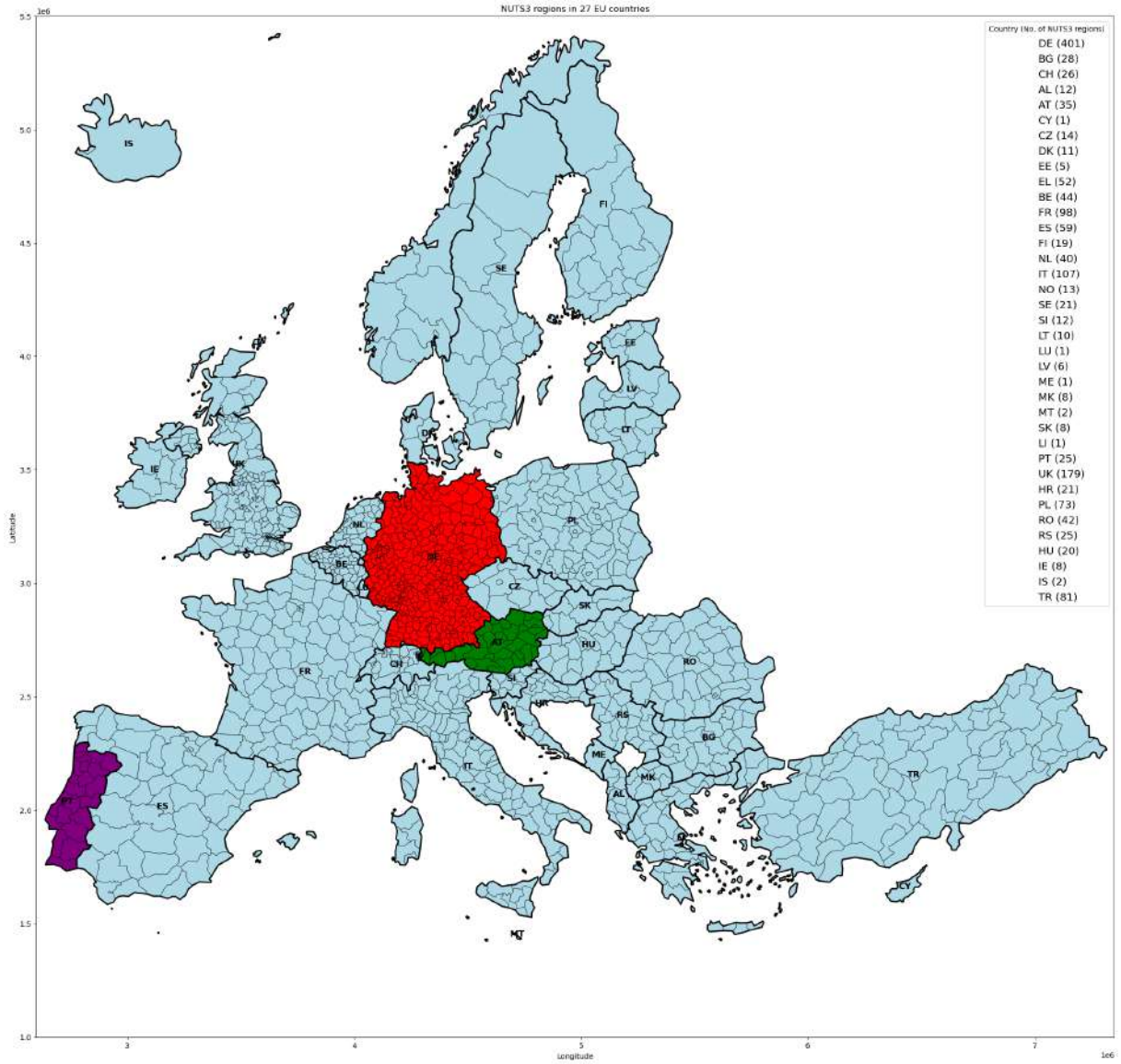


Figure 1.2: NUTS3 regions representation

This research seeks to answer the following questions:

1.Is it possible for machine learning-based dasymetric weighting schemes that utilize ancillary information on transport, energy, socio-economic, industry, demographic, and environmental factors at the regional level in the EU, along with other modern sources of information at NUTS-3-resolution, provide accurate disaggregation and estimation of transport-related data across NUTS-3 regions?

In other words, can a machine learning model learn the relationship between an aggregated variable and the ground truth variables at the target level of spatial resolution to estimate the added value at its disaggregated level?

2. Can the final estimate, derived from a machine learning-based dasymetric weighting scheme that incorporates ancillary information, yield a more accurate and precise result than the initial

estimate obtained from a basic disaggregation technique, such as population-weighted interpolation?

1.3 Outline

The thesis commences with an introduction in chapter **Chapter 1** that defines the problem and proposes a solution, while also presenting an alternative solution. In chapter **Chapter 3**, it delves into related work, scrutinizing the Master Model that employs a Self-Supervised Deep Learning algorithm, the alternative clustering approach, and a comparison between the clustering methodology and the Master Model.

Advancing to chapter **Chapter 4**, the methodology is meticulously examined, emphasizing the procedural steps involved in conducting a systematic exploration of a machine learning model. In chapter **Chapter 5**, the experimental results obtained throughout the thesis work are discussed, encompassing both the Master Model and clustering approaches.

Finally, chpater **Chapter 6** concludes the thesis by presenting the key findings and suggesting areas for future work to improve the self-supervised hybrid regression method in terms of accuracy.

2 Literature review

2.1 Literature Review

2.1.1 Traditional Approaches to Spatial Data Disaggregation

Spatial data disaggregation, an essential technique in the field of geospatial analysis, has its roots in traditional methodologies like areal weighting and pycnophylactic interpolation [Goodchild und Lam (1980)].

The areal weighting method, one of the earliest techniques in spatial disaggregation, involves redistributing the aggregated data from a source zone to multiple target zones based on the proportion of the area of the source zone that overlaps with each target zone [Goodchild und Lam (1980)]. This method, though simple and intuitive, relies heavily on the assumption of homogeneity, meaning that it presumes the distribution of the variable of interest within each source zone to be evenly spread. While this assumption simplifies the computation and the interpretation of results, it can also lead to inaccuracies when the true distribution of the variable is uneven.

On the other hand, the pycnophylactic interpolation method, introduced by Tobler in 1979, sought to improve the accuracy of spatial data disaggregation by preserving the total quantity of the variable of interest across the source and target zones [Tobler (1979)]. This method generates a smooth surface across the study area, ensuring that the sum of the disaggregated data matches the original aggregated total. The underlying assumption of pycnophylactic interpolation is that changes between zones are gradual rather than abrupt, leading to a smooth surface representation [Fisher und Langford (1997), Tobler (1979)]. Despite being a more sophisticated method compared to areal weighting, pycnophylactic interpolation still doesn't account for possible abrupt changes in spatial variables, which might lead to inaccuracies in the disaggregated data.

Both areal weighting and pycnophylactic interpolation, despite their limitations, laid a strong foundation for the field of spatial data disaggregation. These techniques catalyzed the development of more advanced methodologies, including dasymetric mapping and intelligent dasymetric mapping, that incorporate ancillary data to improve the disaggregation process [Eicher und Brewer (2001b), Mennis und Hultgren (2006)].

As the field of spatial data disaggregation continues to evolve, these traditional methods still serve as important benchmarks and reference points. Their strengths and weaknesses have guided and continue to guide the development of new methodologies that strive to balance computational feasibility, interpretability, and accuracy [Ghosh und Fung (2020)].

2.1.2 Modern Techniques and Machine Learning in Spatial Data Disaggregation

While traditional methodologies like areal weighting and pycnophylactic interpolation set the foundation for spatial data disaggregation, advancements in computational capabilities and

the availability of rich and diverse data sources have allowed for the development of modern, sophisticated techniques that deliver enhanced accuracy and precision.

Central to these modern techniques are machine learning (ML) and deep learning (DL) methodologies. These data-driven approaches have the potential to capture complex, nonlinear relationships between variables and account for spatial heterogeneity and autocorrelation, leading to improved disaggregation results [Goodfellow et al. (2016)].

Ensemble models, one of the robust machine learning techniques, combine multiple algorithms or multiple instances of the same algorithm to optimize predictive performance. They can capture the strengths of individual models and reduce the bias and variance, making them particularly valuable in the context of spatial data disaggregation [Polikar (2006)].

Further driving the advancement of spatial data disaggregation are deep learning techniques. Deep learning, a subset of machine learning, utilizes artificial neural networks with multiple layers (deep structures) to model high-level abstractions in data. Convolutional Neural Networks (CNNs), a class of deep learning models, have proven to be particularly beneficial in spatial data disaggregation due to their ability to effectively handle grid-like topology data, like images or spatial grids. CNNs can recognize and extract hierarchical features in data, making them powerful tools for capturing complex spatial patterns and relationships [LeCun et al. (2015)].

An exemplar contribution in this field is the work of Monteiro et al. (2019), who proposed an innovative hybrid regression disaggregation method integrating CNNs and random forest methodologies [Monteiro et al. (2019)]. This approach leverages the strength of deep learning's ability to learn abstract, intricate spatial patterns through CNNs and the robustness and interpretability of random forest, a traditional machine learning method. The combination results in an effective and versatile methodology that can address the complexities and challenges associated with spatial data disaggregation. This method, being self-supervised, further reduces the need for excessive labeling, making it a practical solution in scenarios with limited labeled data [Monteiro et al. (2019)].

In conclusion, the shift from traditional to modern machine learning and deep learning techniques represents a significant evolution in the field of spatial data disaggregation. This transition has provided the field with robust, sophisticated tools that have the potential to manage the complexities inherent in spatial data and deliver precise, reliable results. The developments underscore the vibrant and dynamic nature of spatial data disaggregation as it continues to adapt and evolve in response to new technologies and methodologies.

2.1.3 Data Preprocessing in Spatial Data Disaggregation

Data preprocessing is a crucial step in spatial data disaggregation, as it significantly affects the performance and accuracy of subsequent spatial analysis. It involves several stages, including data cleaning, standardization, dimensionality reduction, and feature selection, each serving a specific purpose to ensure the quality and usefulness of the spatial data [Pedregosa et al. (2011)].

Data cleaning is the first step in the data preprocessing pipeline. It involves the removal of inconsistencies, errors, or outliers in the dataset that might skew the analysis results. In spatial

data disaggregation, such errors could result from various sources, such as sensor noise, faulty measurements, or human errors during data collection. These inaccuracies need to be identified and addressed promptly to avoid inaccurate disaggregation outcomes.

The next step, standardization, is crucial in bringing all variables to a similar scale. Spatial data often consist of different types of data collected from various sources, each with its own scale of measurement. Standardization ensures that the magnitude of a variable does not influence the model disproportionately, thus improving the comparability of different variables.

Dimensionality reduction is another vital step, particularly when dealing with high-dimensional spatial data. High-dimensional datasets, while rich in information, often suffer from the 'curse of dimensionality,' where the space's dimensionality becomes a hindrance to effective learning due to the sparsity of high-dimensional spaces. Techniques such as Principal Component Analysis (PCA) are often used to reduce the data's dimensionality, thereby simplifying the model's complexity and enhancing its interpretability [Jolliffe (2002)].

Finally, feature selection is crucial in identifying the most relevant variables that contribute significantly to the spatial disaggregation model's predictive power. By selecting only the essential features, computational efficiency can be enhanced, and the likelihood of model overfitting can be reduced. Moreover, it helps in improving the model's interpretability by focusing on a subset of meaningful features instead of an overwhelming number of variables [Guyon und Elisseeff (2003)].

Through these preprocessing steps, spatial data are transformed into a more manageable, efficient, and meaningful format for analysis. This is critical as the quality of data preprocessing can significantly impact the efficacy and reliability of the spatial disaggregation models developed later, especially when using machine learning methods [Pedregosa et al. (2011)].

2.1.4 Limitations and Future Directions

Spatial data disaggregation techniques have revolutionized how we understand, analyze, and manipulate spatial data. However, it is important to note that they are not without their limitations, which present both challenges and opportunities for future research and development.

One major limitation is the "modifiable areal unit problem" (MAUP), which arises from the imposition of artificial units of spatial reporting on continuous geographical phenomena [Openshaw (1984)]. The MAUP can lead to substantial variations in statistical results, depending on the size and shape of the chosen spatial units. Furthermore, most disaggregation techniques inherently assume homogeneity within these units, which may not always hold true.

Additionally, while modern machine learning-based techniques provide advanced capabilities for spatial disaggregation, they are not entirely immune to problems such as overfitting and noise sensitivity. Overfitting occurs when models capture random noise in the training data, leading to poor performance on unseen data. Noise sensitivity, on the other hand, highlights the vulnerability of models to errors or inconsistencies in the input data Goodfellow et al. (2016).

Moreover, the reliance on extensive computational resources can also be a limiting factor, particularly for methods such as deep learning models that require significant training data and computational power. This limitation may not only impede the application of these techniques in resource-constrained settings but may also affect their efficiency and scalability [Goodfellow et al. (2016)].

Looking ahead, several areas warrant further research and improvement. Firstly, developing methods to robustly handle the MAUP could significantly improve the accuracy of spatial data disaggregation. Secondly, more focus could be placed on creating models that are both noise-resistant and less prone to overfitting. This might involve developing novel regularization techniques, improving model interpretability, or integrating ensemble methods for more robust predictions [Monteiro et al. (2020)].

Furthermore, improvements in data preprocessing could aid in mitigating some of the current limitations. For instance, advanced techniques for handling missing data, outlier detection, and dimensionality reduction could lead to more reliable and efficient disaggregation processes.

Finally, the integration of additional data sources, including remote sensing data, social media data, and more, could enhance the richness and accuracy of spatial data disaggregation. Such data, when combined with advanced machine learning models, could open up new avenues for high-resolution, real-time spatial data analysis [Monteiro et al. (2020)].

In conclusion, while the current state-of-the-art in spatial data disaggregation is quite advanced, there are numerous opportunities for future research, ranging from addressing inherent methodological issues to integrating diverse data sources and optimizing machine learning techniques.

3 Objective of the Study

Working towards more sustainable transportation requires estimating the potential for reducing carbon emissions in different regions. As mentioned in the literature review (Chapter 2), this task is deeply connected to spatial disaggregation of transport data, a complex field that currently faces significant barriers due to the lack of detailed ground truth data, especially at district levels. Overcoming this challenge could enhance the accuracy of machine learning techniques used for spatial disaggregation, thereby providing more precise information across Europe's varied landscapes. The groundbreaking efforts by [Monteiro et al. (2019)] have been pivotal in this regard. They proposed a hybrid spatial disaggregation technique, where machine learning methods are employed to enhance the process of disaggregating historical census data into high-resolution grids. Specifically, they utilized mass-preserving areal weighting, Pycnophylactic interpolation, and dasymetric mapping in combination with machine learning. This innovative approach has provided insights into changes in geographical population over time and improved the integration of this data with other geographic information system (GIS) layers. Such pioneering research is opening new paths for future studies focused on overcoming the inherent challenges of spatial disaggregation of transport data, setting the stage for improved estimations of regional potential for decarbonization within the transport sector, thereby contributing to sustainable transportation strategies.

In response to the challenges identified in the literature review (Chapter 2), this research aims to discern the most accurate machine learning method for disaggregating data from the country level to the district level, considering heterogeneous data and spatial dependencies, along with interactions between transport-related information and ancillary data. To this end, a Master Model is proposed for each nation, crafted from data sourced from 26 other EU members, employing a hybrid disaggregation approach. This model is based on a self-supervised deep learning algorithm that addresses nonlinear relationships between variables and spatial autocorrelation in geospatial data, enhancing the estimation accuracy of transport-related metrics [Monteiro et al. (2020)]. The main aim is to uncover a model that offers reliable transport estimates across the 1170 NUTS3 regions spanning all European Union member states, through the investigation and performance assessment of various machine learning algorithms. This data-driven approach, tailored to each region, aims to expand the current understanding and provide practical solutions for real-world applications.

3.1 Scientific Hypothesis

This work is premised on the assumption that map data, extracted at country level (NUTS0) and also at other degree of spatial resolution, like group of states or provinces (NUTS2) and disaggregated to the district resolution level (NUTS3), can provide a granular understanding of transport-related dynamics across Europe. The assumption is rooted in the belief that the spatial disaggregation of transport data can reveal intricate patterns and relationships that are otherwise obscured at a higher level of aggregation. This approach is particularly relevant given

the diversity of data sources across Europe and the need for precision in estimating regional decarbonization potentials in the transport sector.

It is proposed that transport-related dynamics are embedded or encoded in the mapped data. To extract these dynamics, a data-driven model is developed to identify multivariate correlations between features. These correlations are then used to iteratively disaggregate unknown features. This proposition is based on the methodology where self-supervised learning is explained. The model employs a variety of machine learning techniques, including ensembles of dimensionality redction, clustering and deep learning algorithms, to enhance the accuracy of spatial data disaggregation.

To test this hypothesis, data collected from 26 European nations is employed as training data for the remaining country under examination (PT, DE, AT). This strategy ensures that the model captures the comprehensive nature of the EU, rather than just conforming to a specific subset of data. As such, the significance lies not in the extensive size of the testing data, but in its particularity to one country. This is in contrast to a randomly shuffled portion which might be more representative of the entire dataset. Furthermore, if a well-known subset of data at the NUTS3 level exists for the country being tested, this specificity could be beneficial, aiding the model in discerning meaningful relationships. This distinct approach is essential in generating a robust and reliable model, capable of disaggregating data for all 27 member countries of the European Union.

4 Methodology

4.1 Introduction

This chapter in-depth examines the methodology employed to address research questions raised in **Chapter 1**. In **Chapter 4**, **Section 4.1** is provided an outline of the procedural steps involved in conducting a systematic investigation of a machine learning model.

This thesis encompasses a comprehensive machine learning workflow, aiming to provide a solution to the challenge outlined in the subsection Problem Definition (1.1.1) of the Introduction chapter (1). The objective of the research is defined as finding the most accurate machine learning method capable of disaggregating data from the country level to the district level within the complex and diverse geographical and data landscapes of Europe.

Datasets relating to the transport sector at different NUTS-X levels, including OpenStreetMap (OSM) data, Synthetic European Road Freight Transport Flow data, and Vehicle Stock data (for Germany), are collected and processed in the initial stages of the workflow.

This data is then meticulously explored and prepared. Visualizations such as bar charts and histograms are utilized to provide insights into the distribution of transportation infrastructure data and highlight potential trends or anomalies that may influence the performance of future machine learning models.

Subsequent steps involve the rigorous cleaning of the dataset, where anomalies like zero-values and missing values are appropriately addressed. Data standardization is applied to ensure all the features are at the same scale, enabling more effective processing by machine learning algorithms.

Following this, the high-dimensionality of the collected datasets is managed through the application of dimensionality reduction techniques such as Uniform Manifold Approximation and Projection (UMAP). The goal of these techniques is to reduce the number of dimensions while maintaining the data's structure.

Then, feature selection, a critical step in building successful machine learning models, is performed. Important variables are identified, and irrelevant or redundant ones are discarded. Several feature selection methods are employed in this stage to choose the most relevant features for the subsequent steps of the workflow.

The last stages of the workflow focus on hyperparameter tuning and a decision about the potential application of clustering. A critical measure used in this decision-making process is the Cross-Validation (CV) score. In machine learning, cross-validation is a technique used to assess how well a model will generalize to an independent data set. It involves training the model on a subset of the data and then testing it on the rest. The CV score is a performance metric calculated from the cross-validation process and indicates the predictive accuracy of the model. Higher CV scores represent better model fit and more accurate predictions.

If the CV score with the selected features is found to be lower than the CV score with all features, adjustments are made to the feature selection parameters, in a process known as

hyperparameter tuning. Clustering methodology might be applied, followed by model selection, training, evaluation, and validation for each cluster. Conversely, if clustering is not applied, model selection, training, evaluation, and validation are carried out for the entire dataset.

Various machine learning models known for their potential in enhancing spatial disaggregation techniques, including Artificial Neural Networks (ANNs), Random Forests (RFs), and Support Vector Machines (SVMs). Cross-validation techniques are used to compare the performance of these models, with the most effective model selected for use.

The overarching goal of this workflow is to provide a comprehensive solution to the challenge of accurately spatially disaggregating transport-related data across Europe. Each stage in the workflow builds upon the previous one, ultimately aiming to improve the accuracy of the final model.

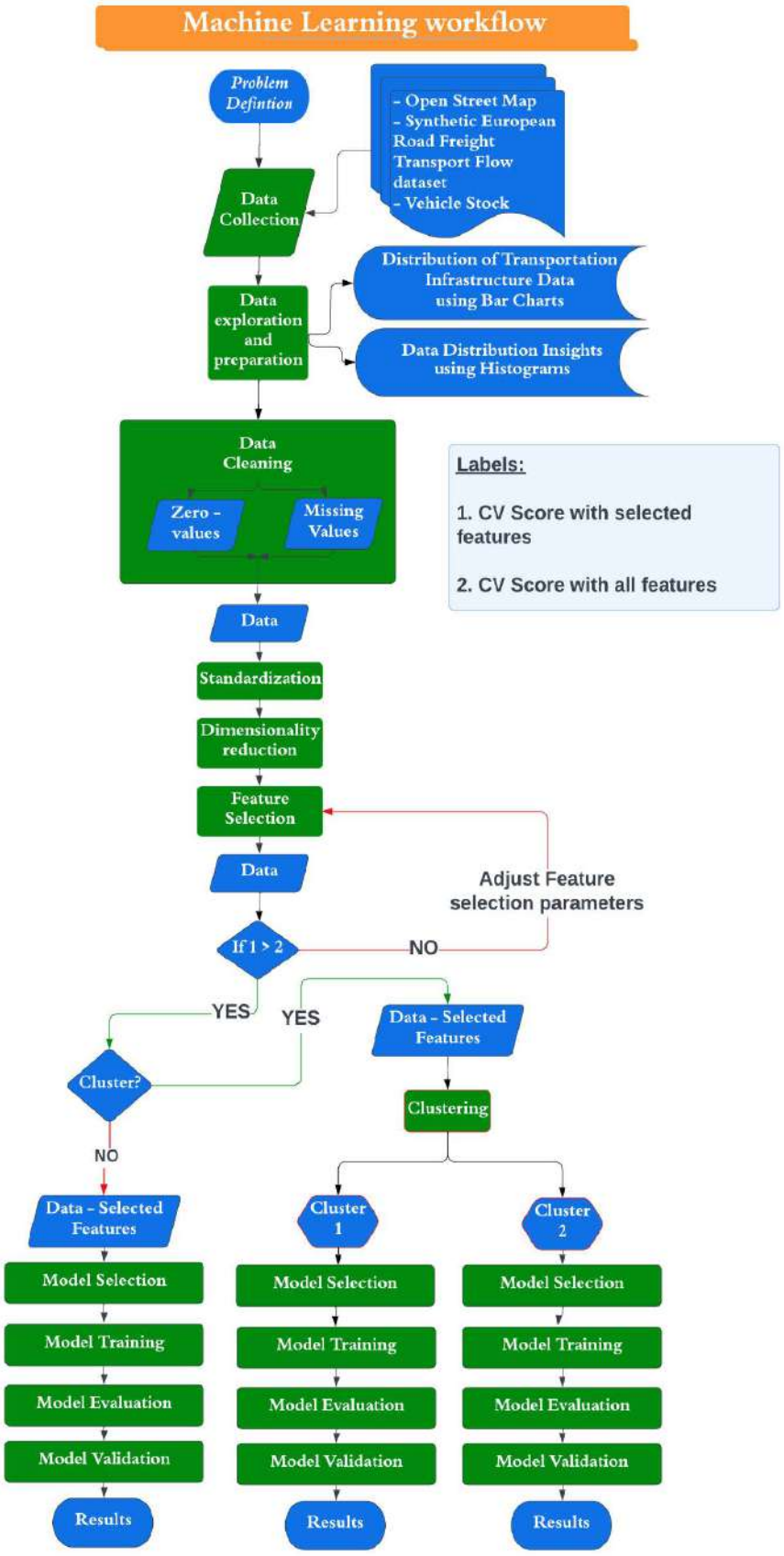


Figure 4.1: Machine Learning workflow.

4.2 Developing a Self-Supervised Hybrid Regression Method for Spatial Data Disaggregation

As discussed in the Problem Definition and Proposed Solution subsections, spatial disaggregation presents challenges due to a lack of ground truth data. A hybrid method is proposed that utilizes self-supervised regression techniques for learning from existing data while potentially refining outcomes iteratively. This approach may improve accuracy without access to ground truth information. A Master Model that integrates weighted population methods and regression-based dasymetric mapping is developed in order to address these complexities and achieve precise spatial disaggregation.

The methodology of this work involves a number of steps, as detailed below. Please refer to Figure 4.2 for a visual overview.

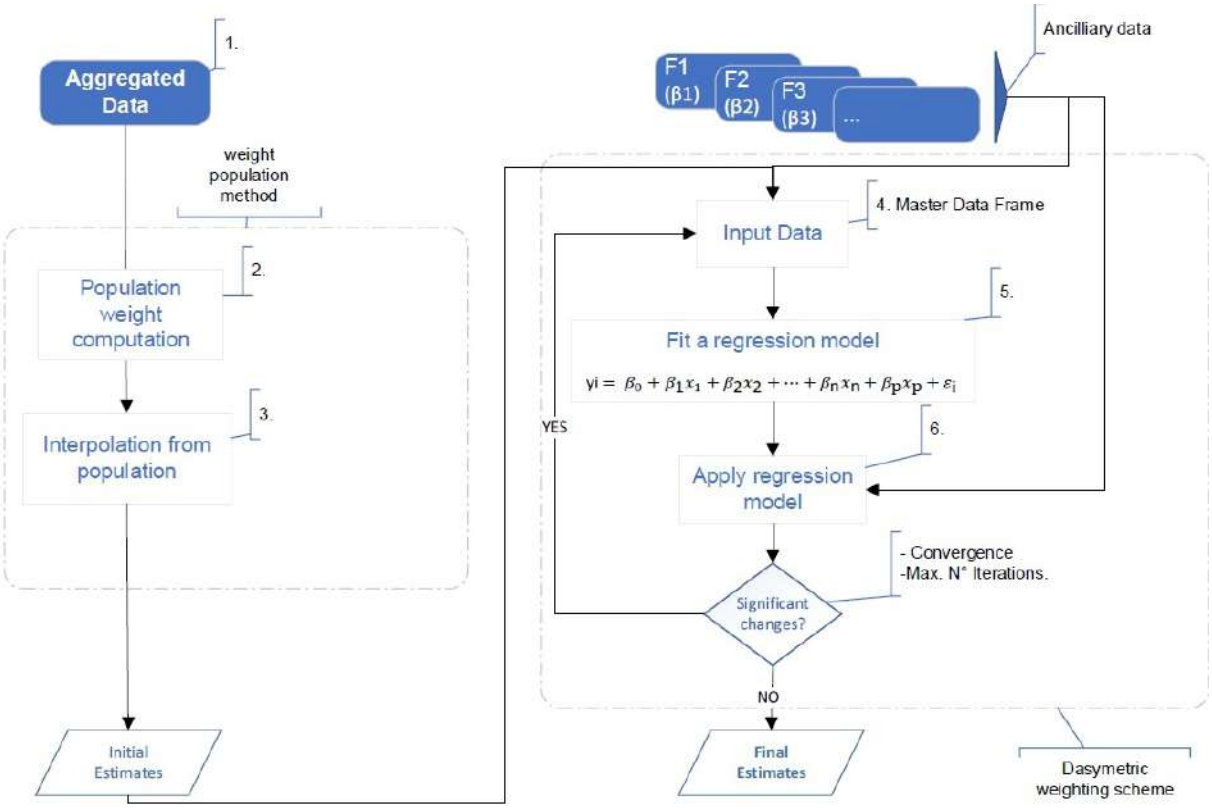


Figure 4.2: Spatial Disaggregation using a hybrid disaggregation Regression Method- Self Supervised Approach.

The general steps of the procedure can be seen below:

1. **Aggregated data at country level**: In the first step of the methodology, aggregated data at the country level is handled. As explained in the proposed solution subsection, a variable, such as charging stations or train stations, is selected through the feature selection procedure and aggregated from NUTS3 level to country level. This step, depicted in Figure 4.2, ensures that the selected variable contains sufficient information for accurate and effective disaggregation results.

Moreover, this allows for the validation of the method through comparison of the disaggregated values with the actual values at the country level.

2. **Population weights computation**: The second step in the methodology, as illustrated in Figure 4.2, involves the computation of population weights. Population density values for each NUTS3 region are calculated. This involves dividing the population of each NUTS3 region by the total population of the corresponding country. The resulting values represent the population density for each NUTS3 region. The equation is given as:

$$W^{i}_{\text{pop,NUTS3}} = \frac{\text{NUTS3}^{i}}{\text{NUTS0}^{k}} \tag{4.1}$$

In this formula:

$W^{i}_{\text{pop,NUTS3}}$ represents the population density for the i-th NUTS3 region. NUTS3^{i} denotes the total population for the i-th NUTS3 region. NUTS0^{k} signifies the total population for the k-th country.

3. **Initial estimation of the aggregated variable**: As depicted in step 3 of Figure 4.2, the initial estimation of the aggregated variable is performed. In this step, the aggregated variable at the country level is estimated utilizing the Weight Population Method, a straightforward heuristic disaggregation procedure. This method is selected due to its effective use of population as an initial estimate for spatial disaggregation, attributed to the high correlation between population and other geographic data.

InitialEsNUTS3i- refers to the initial estimate for the aggregated variable in the i-th NUTS3 region. It is computed by multiplying the total population for the i-th NUTS3 region by the population density for the same region. This relationship can be expressed mathematically as:

$$\text{InitialEsNUTS3}^{i} = \text{NUTS3}^{i} \times W^{i}_{\text{pop,NUTS3}} \tag{4.2}$$

4. In step 4 of the process as illustrated in Figure 4.2, the development of a **comprehensive European Union-wide data frame for model training is undertaken**:

While the methodology employed in this thesis is grounded in the literature review of [Monteiro et al. (2020)], this thesis introduces a new Master Data Frame structure, which is a proposed idea to enhance the interrelationship between variables. The proposed structure seeks to facilitate spatial disaggregation by providing an efficient representation of data across the entirety of the EU.

This data frame consists of appended data frames, with each data frame consisting of a different predictive ancillary covariate data (a pseudo value of an ancillary feature), remaining ancillary data at its true NUTS3 value, initial estimate data and true NUTS3 target value for each predictive covariate as the dependent target variable. This unique structure helps capture complex relationships between aggregated variables and ancillary features more accurately and precisely; ultimately leading to improved precision and accuracy when disaggregating results. It is important to note that the predictive ancillary covariate data (a pseudo value of an ancillary

feature) is calculated by multiplying the real pseudo variable by the population weight for each NUTS3 region.

AF pseudo value NUTS3i - This represents the pseudo value of an ancillary feature i for the NUTS3 region. It is calculated as the product of the total population for the i-th NUTS3 region and the population density for the same region.

$$\text{AF pseudo value NUTS3}^i = \text{NUTS3}^i \times W^i_{\text{pop,NUTS3}} \tag{4.3}$$

During the creation of the data frame for model training, it is crucial to consider that an aggregated variable (such as charging stations) is replicated 26 times, corresponding to each concatenated training data frame. This repetition is attributed to the inclusion of 26 countries in the training dataset, ensuring that the model captures the relationships and patterns present within the diverse set of data. This implies that the aggregated variable serves as input data, while each of its ancillary variables acts as its predictive variables or output data. This setup would allow the model to learn the complex relationships between the aggregated variable and all the ancillary features, ultimately leading to accurate predictions. By identifying the variables that are most relevant to the problem at hand and using them to improve the disaggregation results, the model would make informed decisions based on all the available information, instead of relying on a simplistic population heuristic.

5. In step 5 of the process as illustrated in Figure 4.2, the **model is fitted**: To fit the regression model, the Master Data Frame is split into two segments: one country (e.g., Germany, Portugal, or Austria) serves as the testing data, while the remaining 26 countries form the training data. During this process, the model identifies patterns and relationships between input and output variables. Evaluating the model's performance on both the training and cross-validation datasets is crucial to ensure that it does not overfit the training data and can generalize well to unseen data. Overfitting occurs when the model is too complex and fits the training data too closely, leading to poor performance on new data.

The relatively high testing data size is intentional, this approach ensures that the model is representative of the entire EU, rather than overfitting to a specific subset of the data. Therefore, despite the high testing data size, this approach is crucial for producing a robust and reliable model that can be used for disaggregating data for the 27 countries of the European Union.

The Master Data Frame for training and testing the model on Germany, is seen in Figure 4.3 and Figure 4.4 respectively.

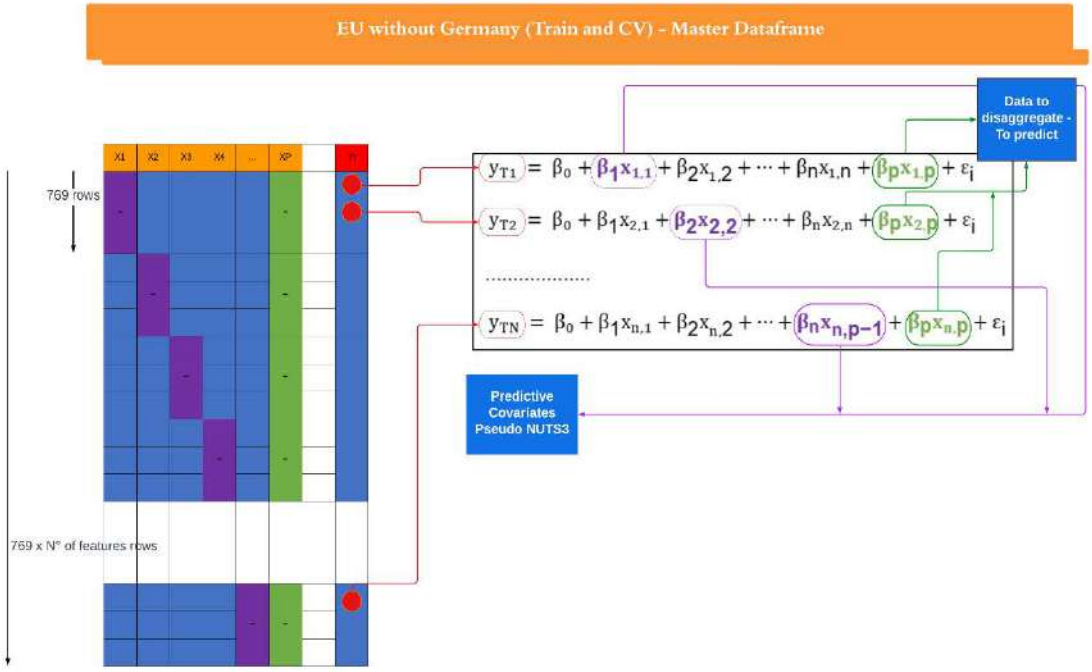


Figure 4.3: Master Data frame used in the training of the regression algorithm with Germany as the testing country.

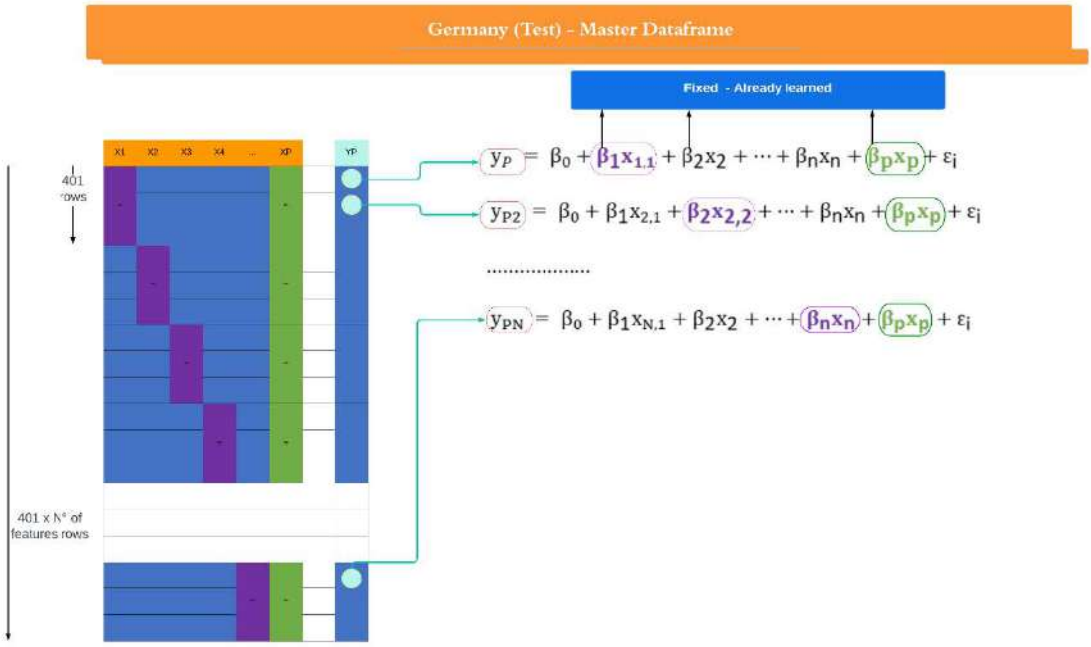


Figure 4.4: Master Data frame used in the Testing of the regression algorithm on Germany.

x1,2; x1,2 ... x1,n - These are the ancillary features for the NUTS3 region, representing NUTS3 values.

x1,1; x2,2 ... xn,p-1 - These are the ancillary features' pseudo values.

x1,p; x2,p ... xn,p - These values represent the data to be disaggregated. They include both initial estimates and refining estimates.

β_0 - This is the intercept or the constant term of the model.

$\beta_1, \beta_2, \beta_3, \ldots, \beta_n$ - These are the weights or coefficients that are learned during the training process of the model.

YT - This is the ground truth variable or the target variable that the model is trying to predict.

Yp - This is the predicted variable or the model's output.

6. **Applying the Model**: By taking into account weights acquired during training, this model can make accurate predictions for new estimates.

When applying the model, predictive covariates are excluded. Doing this ensures that the model has already captured the relationships between aggregated variables and the target variable, allowing it to make informed inferences about subsequent estimates with confidence.

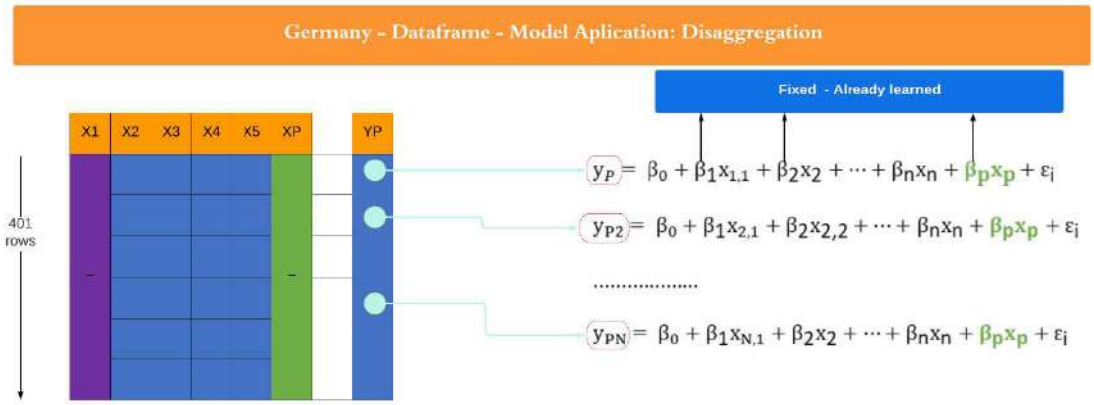


Figure 4.5: Data frame used in the model application in Germany, Data disaggregation.

This application can be observed in the data frame shown in Figure 4.5 which demonstrate the model application in disaggregating data for Germany.

7. **Repeat steps 5 to 7 iteratively**: The model is trained again with the new estimates until the estimates converge to a tolerance value or until a maximum settled threshold number of iterations is achieved.

4.3 Data Collection

The data collection procedure for this problem of spatial disaggregation involves obtaining and processing several datasets at the NUTS3 level.

The main collected datasets related to the transport sector for this thesis include:

4.3.1 OpenStreetMap (OSM) data

OSM is a collaborative project to create a free and open-source map of the world [contributors (2021)].

The OSM data collected comprises length information in meters for various features such as bicycle lanes, bus routes, railways, major roads, and shipping routes, in addition to the count of mapped stations in each NUTS3 region. The mapped stations include fuel stations, charging stations, bicycle stations, bus stations, airport stations, railway stations, train stations, subway stations, light rail stations, shipping stations, and helicopter stations.

Various tools are utilized to query and process the Open Street Maps (OSM) data, such as the OSMnx Python package, Nominatim API, Overpass API, TagFinder, and OpenStreetMap Data in Layered GIS Format Documentation [Boeing (2017), Contributors (2021a,b), TagFinder (2021)]. Initially, the OSM data is queried at the NUTS2 level to reduce computation time and accelerate the query process. Despite this initial collection at a higher level, the data can still be mapped to the NUTS3 level by overlapping each geospatial feature over its corresponding NUTS3 polygon using geographic coordinates.

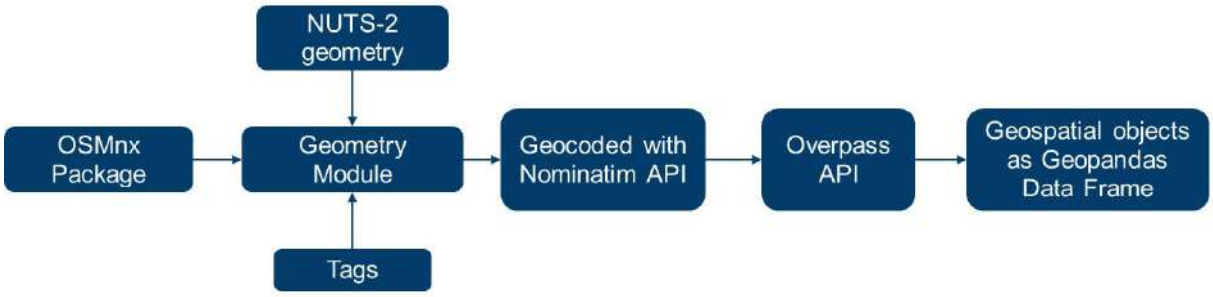


Figure 4.6: OSM data collection at NUTS2 level using the OSMnx package.

The road, bicycle, bus, railways networks in Open Street Maps (OSM) are represented physically by a line string, which is a combination of cardinal points (longitude and latitude). To accurately measure the length of these networks, each line string is queried at NUTS2 level and then intersected over each NUTS3 region. This intersection cuts the network into smaller pieces that can be accurately measured using a Python interface to PROJ, which calculates the geodesic length of the shapely geometry in meters. However, computing the difference between each latitude and longitude point being mapped and summing them up for the calculation can be a time-consuming process.

To overcome this challenge and speed up the calculation process, parallelization techniques are applied, which allow the workload to be divided among multiple processors or cores. By utilizing these techniques, the process of measuring road lengths is significantly accelerated.

4.3.2 Synthetic European Road Freight Transport Flow

This dataset is based on the publicly available ETISplus project from 2010, which was a joint project between the European Commission and EU Member States [Flötteröd und Lückenkötter (2018)]. The ETISplus project is a collection of Europe-wide freight volumes of calibrated origin-destination matrices with real-world traffic flows. This dataset contains updated results of the ETISplus project that incorporates current Eurostat data and a forecast up to 2030. The updated dataset provides a synthetically generated truck traffic volume for each road section. It was developed by researchers from the German Aerospace Center (DLR) and the Institute for Transport Studies (IfV) at the Karlsruhe Institute of Technology (KIT).

To conduct the analysis for this thesis, the origin of each of the 1,514,573 road freight traffic trajectories within the dataset is mapped to its corresponding NUTS3 region. Mapping the origin of each road freight traffic trajectory to its corresponding NUTS3 region provides a comprehensive overview of road freight traffic at the NUTS3 level.

4.3.3 Vehicle Stock

This is a dataset for Germany that provides the number of motor vehicles and their trailers by municipality for January 1, 2023, and is sourced from the "Kraftfahrt-Bundesamt" [Kraftfahrt-Bundesamt (2023)], which is the Federal Motor Transport Authority in Germany. It includes information on motorcycles, agricultural tractors, buses, passenger vehicles, load force wagons, and trailers. The dataset is structured to allow for the mapping of the data to NUTS3 regions. To achieve this, the postal codes in the dataset were referenced to merge with an existing Eurostat dataset that maps postal codes to NUTS3 regions.

4.3.4 Other Metrics

The Techno-economic Systems Analysis (IEK-3) Institute at Forschungszentrum Jülich (FZJ) has a rich collection of datasets that are highly applicable to this thesis. Some of these datasets are already mapped at the NUTS3 level, while others are aggregated from LAU to NUTS3 spatial resolution level. The following is a list of these available datasets:

Transportation and Infrastructure:

• Vehicle stock and buildings of Poland.

• Railway length.

Economic Metrics:

• Employment, gross domestic product, and gross value added in various NACE sectors.

• Number of businesses.

Socio-demographic Metrics:

• Deaths, live births, and quality-of-life index.

• Population, total area, and number of buildings.

Land Use and Environment:

• Pixels quantity (agriculture, forests, urban areas, and water bodies).

• Non-residential footprint area.

Energy and Industry:

• Industry electricity demand.

• Industry fuel demand.

• Industry generation capacity.

• Number of industry plants.

• Residential energy demand (energy demand, heat demand, and footprint area for residential buildings).

For all these data, the Forschungszentrum Jülich (FZJ) provides excellent resources to facilitate their processing and interpretation in a disaggregation context.

4.4 Data exploration and preparation

This section involves exploring and analyzing the data to understand the type of problem being solved. The data is stored in a CSV file and has been processed using Python programming language.

4.4.1 Distribution of Transportation Infrastructure Data using Bar Charts

Initially, bar charts were developed for the three collected datasets seen above. The values are plotted at country level to gain relevant insights into each feature and their combination for countries. The bar charts revealed that Italy has a significantly higher number of fuel stations than other countries in the EU, with approximately 23,000 in total. However, Italy has a relatively low number of charging stations, with only around 4,000. In contrast, Germany has around 20,000 fuel stations and approximately 20,000 charging stations, while for example the Netherlands has around 4,000 fuel stations and 5,000 charging stations. These findings provide valuable insights into the development of infrastructure for electric and traditional vehicles in each country [Agency (2021)].

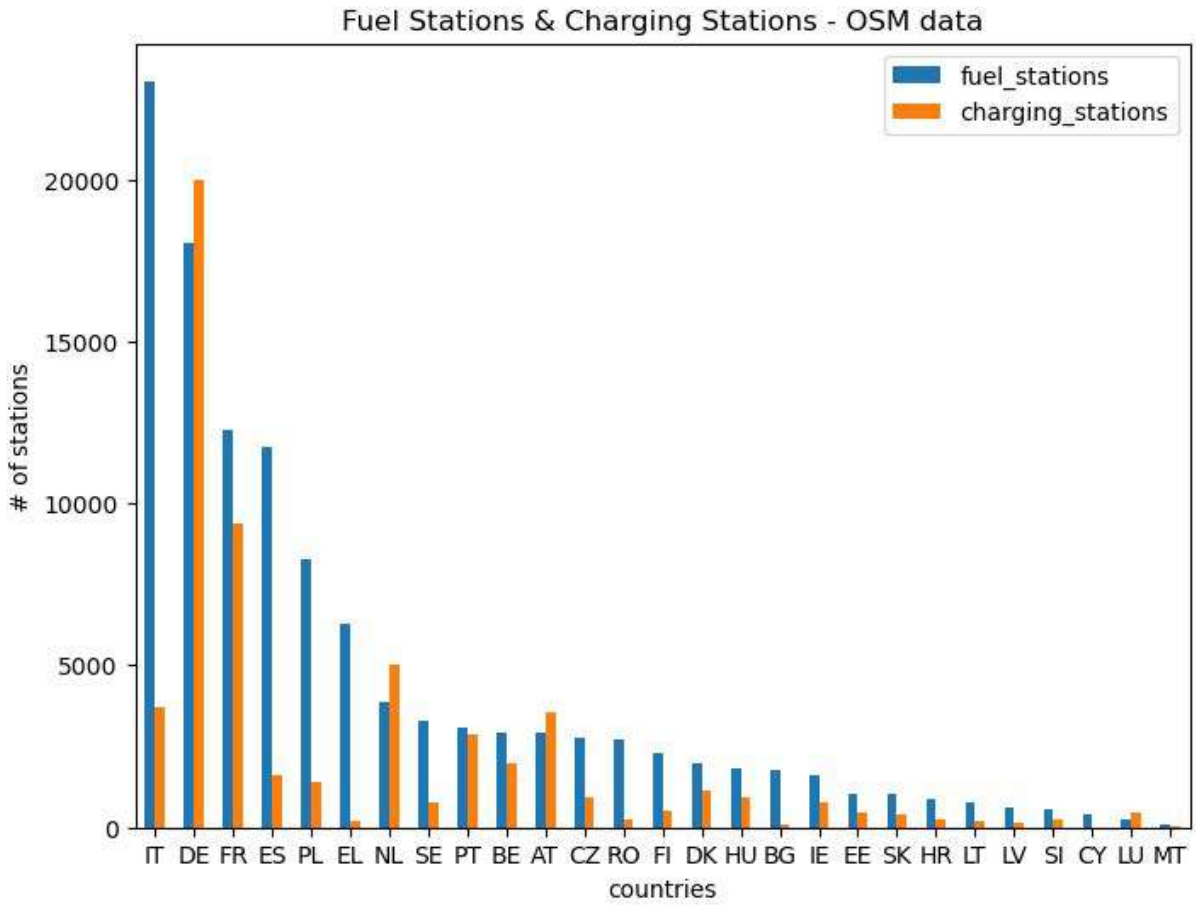


Figure 4.7: Number of fuel and charging stations by country. OSM data.

The lack of sufficient charging infrastructure development in Italy and other countries might be one of the primary reasons for the slow adoption of electric vehicles and the limited growth of EV penetration [Eurostat (2021)]. Insufficient charging infrastructure leaves consumers uncertain about the availability of charging stations, which can make them hesitant to switch from ICE vehicles to EVs. Thus, it is essential for policymakers and industry leaders to prioritize investments in charging station infrastructure to accelerate the adoption of electric vehicles and facilitate the transition towards a more sustainable transportation system.

Although charging infrastructure may not be directly relevant to the thesis topic, it is essential to take into account its implications on transportation in Europe. Acknowledging how inadequate charging infrastructure can affect electric vehicle adoption helps us better comprehend how factors interact and identify potential areas for improvement. Thus, taking this information into account when analyzing data and formulating strategic recommendations is vital.

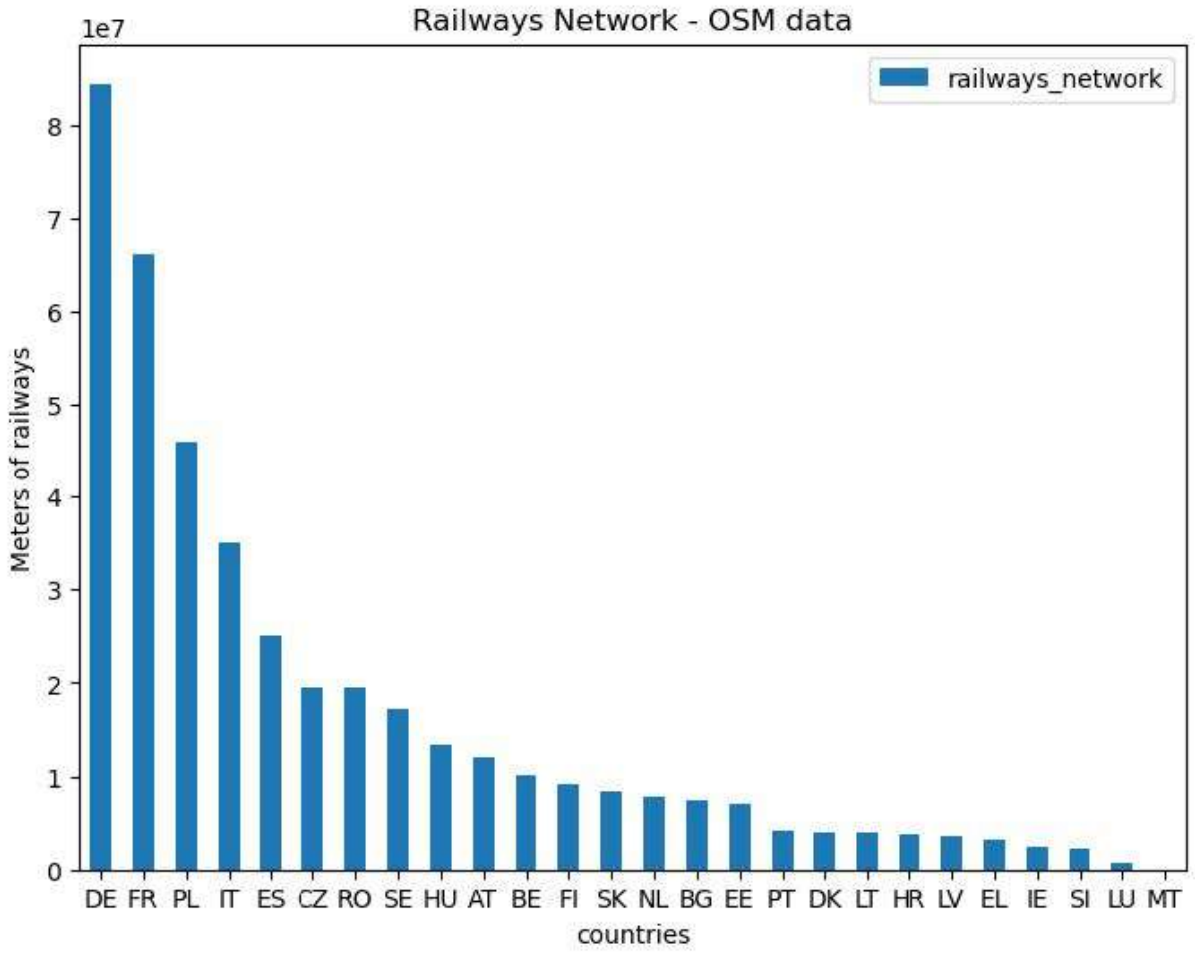


Figure 4.8: Meters of railways network per country. OSM data.

Figure 4.8 shows the total meters of railway network for each country. It is apparent that Germany, France, Italy, and Poland have the highest railway infrastructure.

4.4.2 Data Distribution Insights using Histograms

Histograms provide a visual representation of the data's distribution. It is important to analyze the histograms to determine how continuous the data for each feature is.

Data distribution across NUTS3 regions shows heterogeneity, with many regions having similar values while only a limited amount of data remains in a few NUTS3 areas. Acknowledging and managing this heterogeneity is essential for effectively training machine learning models to recognize underlying patterns and relationships [Hastie et al. (2009)].

The histogram analysis for airport stations' shows that its range value is the most frequently distributed feature across all NUTS3 EU regions. Specifically, the histogram displays that 230 NUTS3 regions have approximately between 1 and 3 airports, while 40 NUTS3 regions have between 9 and 11 airports. It is worth mentioning that the airport stations' range of value includes both commercial airports and private aerodromes.

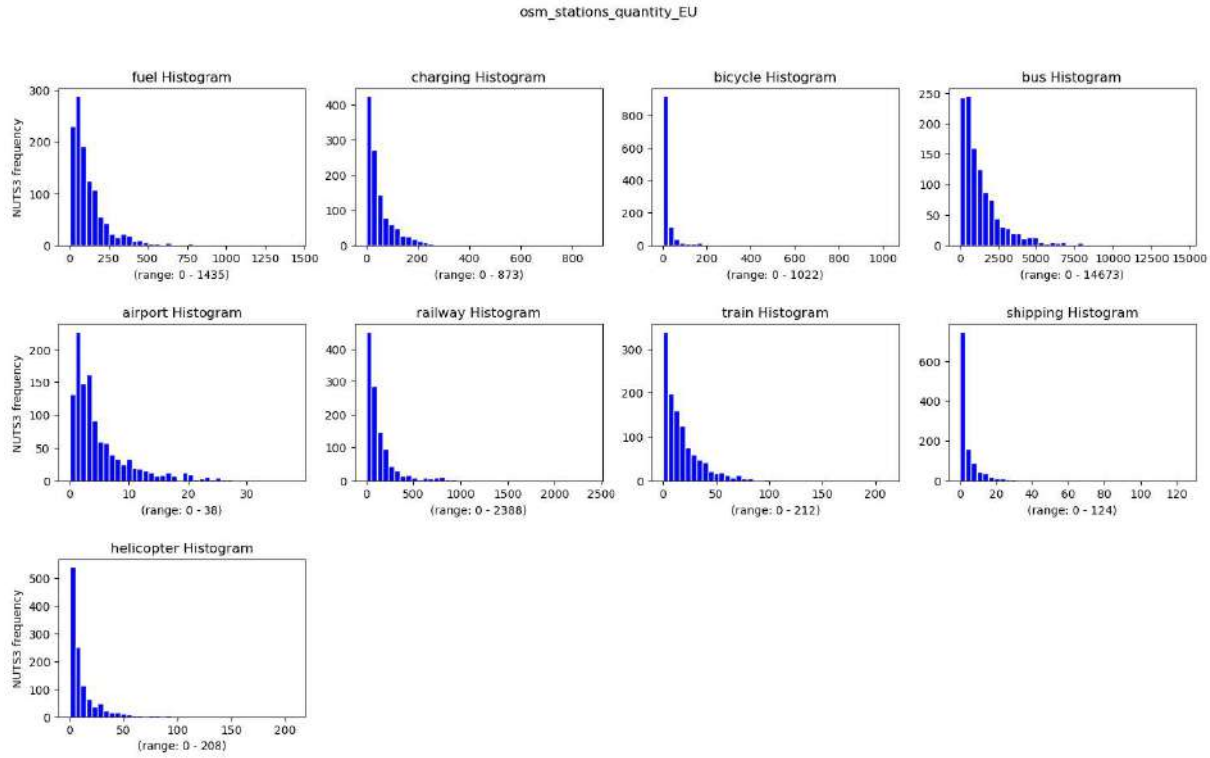


Figure 4.9: OSM stations frequency distribution at NUTS3 level.

Figure 4.9 depicts the distribution of OSM data for physical stations across NUTS3 EU regions [contributors (2021)]. It becomes apparent that there is an absence of well-distributed values across many NUTS3 regions, suggesting a discontinuity in the data as it has been discretized into only specific values within each range. Conversely, some smaller subsets within NUTS3 regions show varying range values indicating possible outliers that could significantly affect model training.

Analyzing other features, it can be seen similar distributions of the data over the European NUTS3 regions.

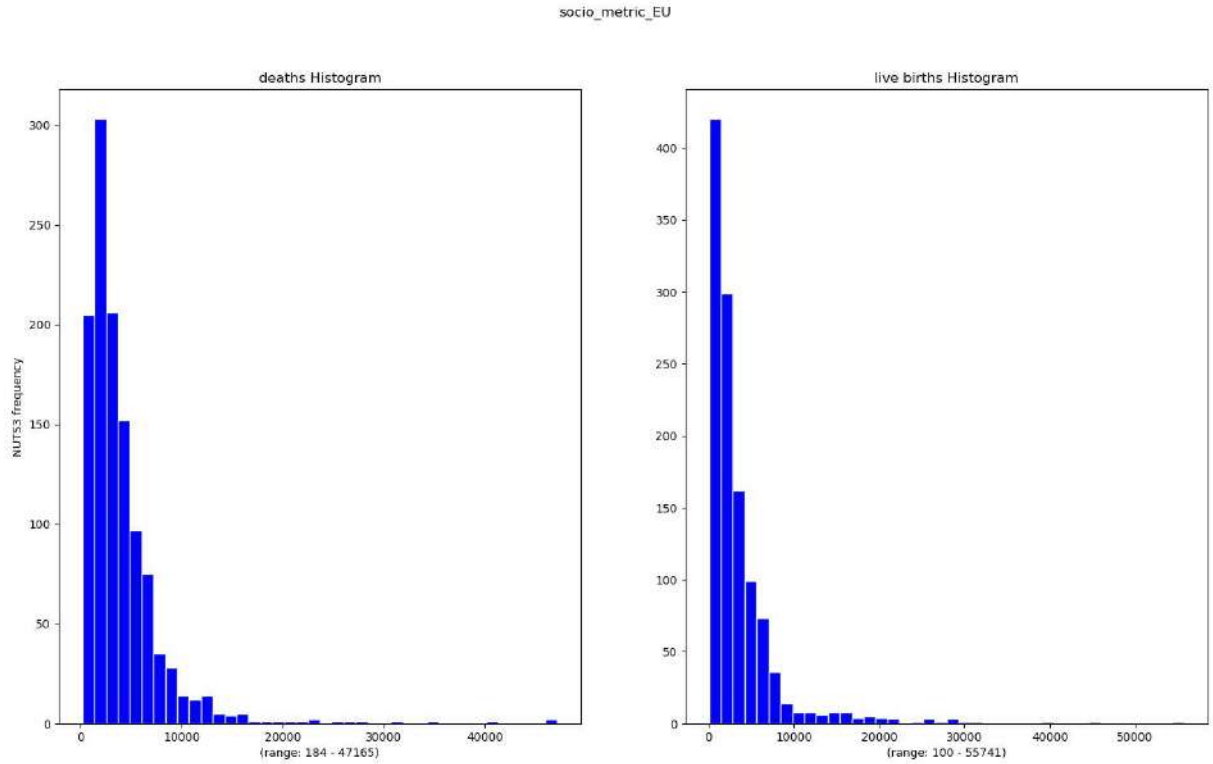


Figure 4.10: Socio-demographic frequency distribution at NUTS3 level.

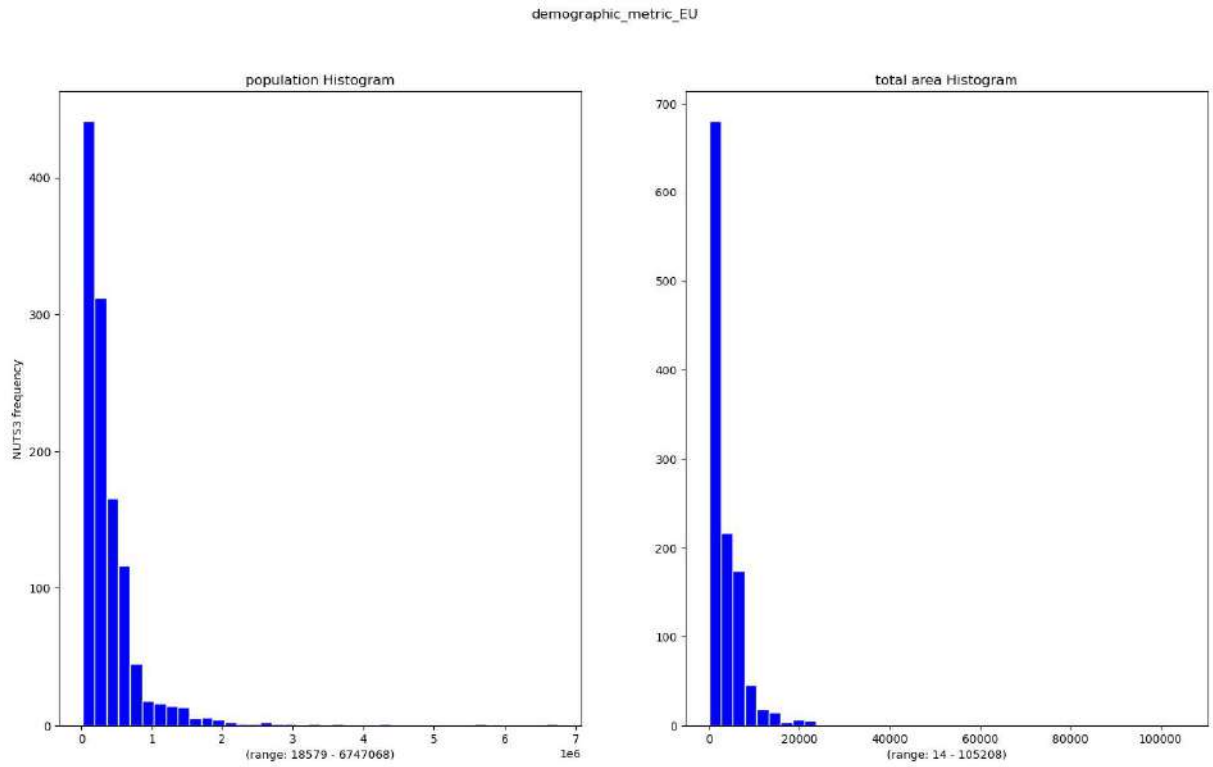


Figure 4.11: Socio-demographic metrics frequency distribution at NUTS3 level.

The analysis of histograms reveals the presence of heterogeneous data with widely varying scales and distributions, including discrete data like OSM data for physical stations across NUTS3 EU regions. Machine learning models, such as Neural Networks struggle to capture relationships between features with varying scales, so normalization, feature selection and feature engineering techniques are applied to address this issue [Goodfellow et al. (2016)]. Addressing heterogeneous data can improve models' ability to capture underlying patterns and relationships.

4.4.3 Data Cleaning

Missing values

In this subsection, the handling of missing values in a dataset is discussed. Missing values can significantly impede machine learning models' performance and result in biased and unreliable results; thus, it's essential to address these issues appropriately during data preprocessing.

For each numeric column in the dataset, the number of missing values (NaNs) is calculated. If this proportion exceeds a predefined threshold (e.g., 10%), the entire column is dropped from the dataset; this ensures that remaining data isn't affected by an excessive number of NaNs which might introduce bias during analysis. On the contrary, if there's less than or equal to 10% missing values per column, those missing values are replaced with their mean value; an imputation technique widely used to maintain overall distribution while dealing with missing values [Goodfellow et al. (2016)].

Metric	Value
Total number of features	177
Total number of dropped features	16
Percentage of dropped columns	9.04 %
Number of remaining features	161

Table 4.1: Handling Missing values.

Zero values

In this subsection, the handling of zero values in the dataset is examined. Zero values can potentially complicate analyses due to their representation as missing or incomplete data or an absence of a feature within an observation. It's essential to take into account how zero values affect data distribution and machine learning model performance when working with zero values.

Management of zero values in features necessitates setting a threshold (e.g., 80%) and analyzing the dataset to calculate how many features contain zeros. Features with percentages equal to or

higher than this threshold should be removed in order to guarantee sufficient non-zero values and minimize negative impacts on analysis [Bishop (2006)].

Prior to removal, a summary of features' zero-value percentages is presented which helps in setting an appropriate threshold. Subsequently, the number of remaining features is reported so that one can assess how much data reduction has been accomplished and its effects on the analysis.

In conclusion, handling zero values involves setting a threshold, computing zero values in features, and removing those exceeding it. This improves the reliability of analyses and machine learning models derived from processed datasets.

Metric	Value
Number of initial features	161
Percentage of columns with all zero values	13.04%
Percentage of columns with at least 95% zero values	31.68%
Percentage of columns with at least 90% zero values	37.89%
Percentage of columns with at least 80% zero values	49.07%
Percentage of columns with at least 70% zero values	56.52%
Percentage of columns with at least 60% zero values	60.87%
Percentage of columns with at least 50% zero values	64.60%
Percentage of features removed due to high percentage of zeros	49.07%
Number of remaining features	82

Table 4.2: Zero-value analysis.

4.4.4 Standardization

Standardization is a fundamental step in prepping data for machine learning models, as it guarantees all features are of the same scale. It is essential for optimal performance of many machine learning algorithms, as features with larger values can dominate the model and lead to overfitting, ultimately affecting its performance. This study employed the StandardScaler from the scikit-learn library in Python, which transforms features to have a mean of zero and standard deviation of 1, thereby providing a Gaussian representation [Goodfellow et al. (2016)]. It should be noted that standardization may lead to negative values; these will be crucial when defining hyperparameters for the machine learning model.

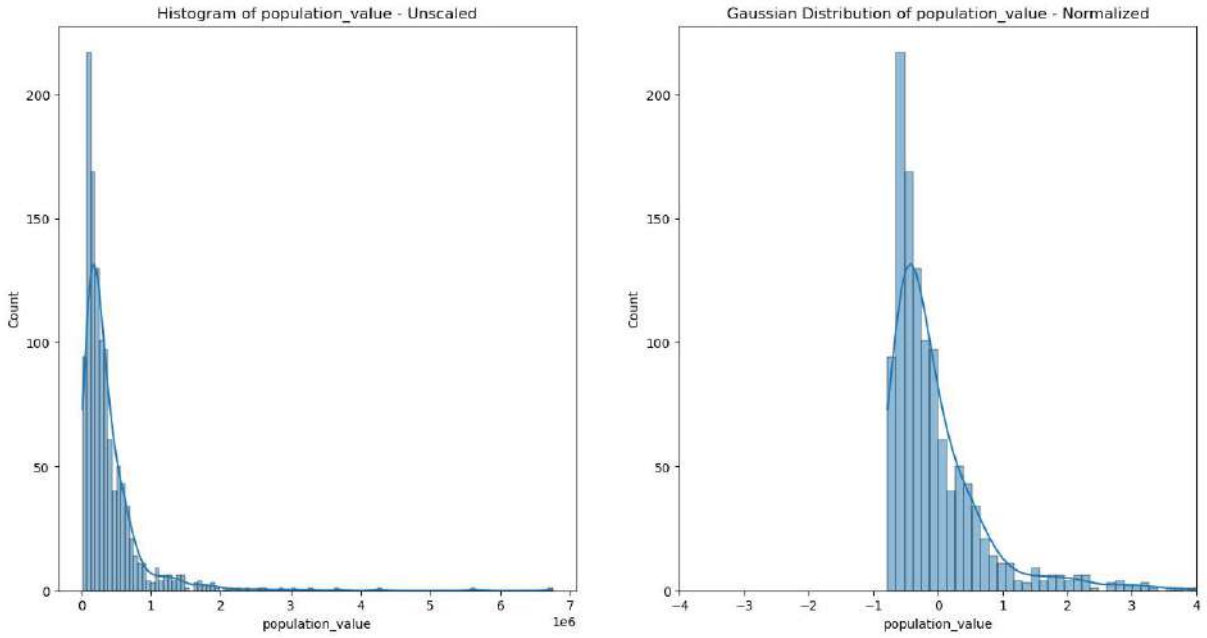


Figure 4.12: Standardization of population.

In Figure 4.12, it can be clearly observed the distinction between unscaled and scaled population data.

4.4.5 Dimensionality Reduction

Dimensionality reduction techniques are essential for managing high-dimensional datasets,which often suffer from the curse of dimensionality and hinder machine learning algorithms.

Popular methods include Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP), all capable of reducing dimensionality while maintaining data structure [McInnes et al. (2020)]. In this study, UMAP was selected due to its superior ability to handle heterogenous and nonlinear data more effectively than other methods.

UMAP-based Dimensionality Reduction

UMAP, an effective dimensionality reduction technique, is ideal for heterogeneous and nonlinear data due to its ability to preserve both local and global structures [McInnes et al. (2020)]. Unlike PCA which primarily focuses on linear transformation or t-SNE which only preserves local elements, UMAP's versatility made it the best fit for the current dataset which consisted of features with various ranges and distributions. After applying UMAP to this dataset, the transformed features could then be further analyzed.

In the methodology employed, specific features from the original dataset were retained while new, dimensionally reduced features were generated. UMAP was applied to training and testing

data, reducing its dimensions to a specified number of components. This transformation sought to capture and preserve structure and patterns within high-dimensional data while simplifying it. To maintain balance between original and reduced features, certain columns from the original dataset were interspersed with reduced UMAP components; then this combined dataset was used for further analysis.

To demonstrate the interrelationships among components in a dimensionally reduced dataset, two-dimensional UMAP plots were generated to display random component combinations.

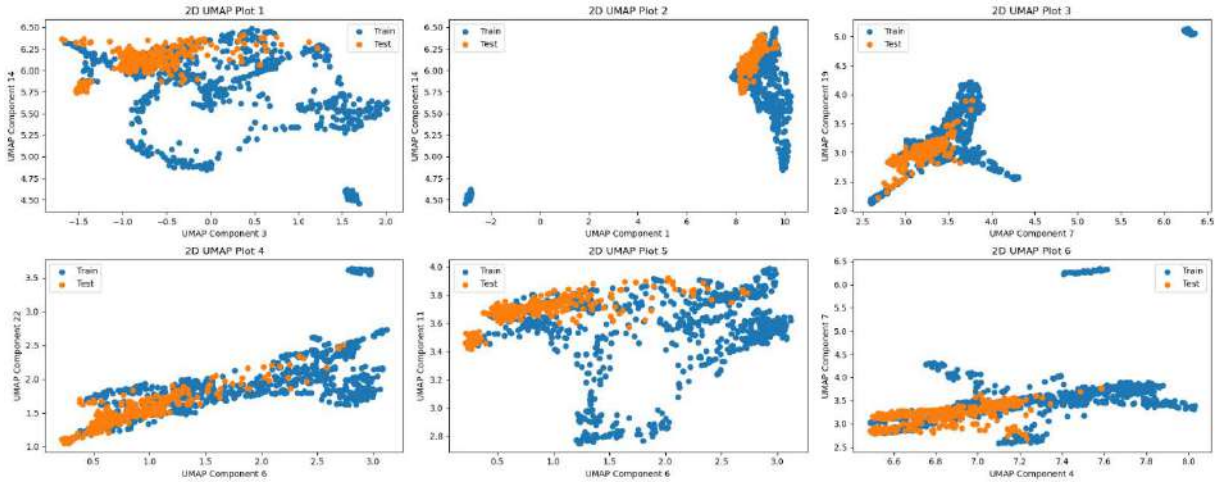


Figure 4.13: Two-dimensional UMAP plots with random component combinations.

The coherence observed between the training and testing data in the 2D UMAP plots suggests that the training data is representative of the testing data. This means that the patterns and structures present in the training data adequately capture the characteristics of the testing data, ensuring the model's generalization capabilities [Goodfellow et al. (2016)]. The similarity between the training and testing data distributions further indicates that the selected features and dimensionality reduction technique have effectively preserved essential information, potentially enabling the model to perform well on unseen data.

Box Plot Analysis of Reduced Features

After UMAP-based dimensionality reduction, box plot analysis was conducted to examine both the distribution of reduced features and original ones [McInnes et al. (2020)]. Box plots provide a visual representation of central tendency, dispersion, and potential outliers in data [Goodchild und Lam (1980)]. By analyzing both box plots for reduced and non-reduced features together, any issues such as extreme values or skewed distributions can be identified and addressed accordingly.

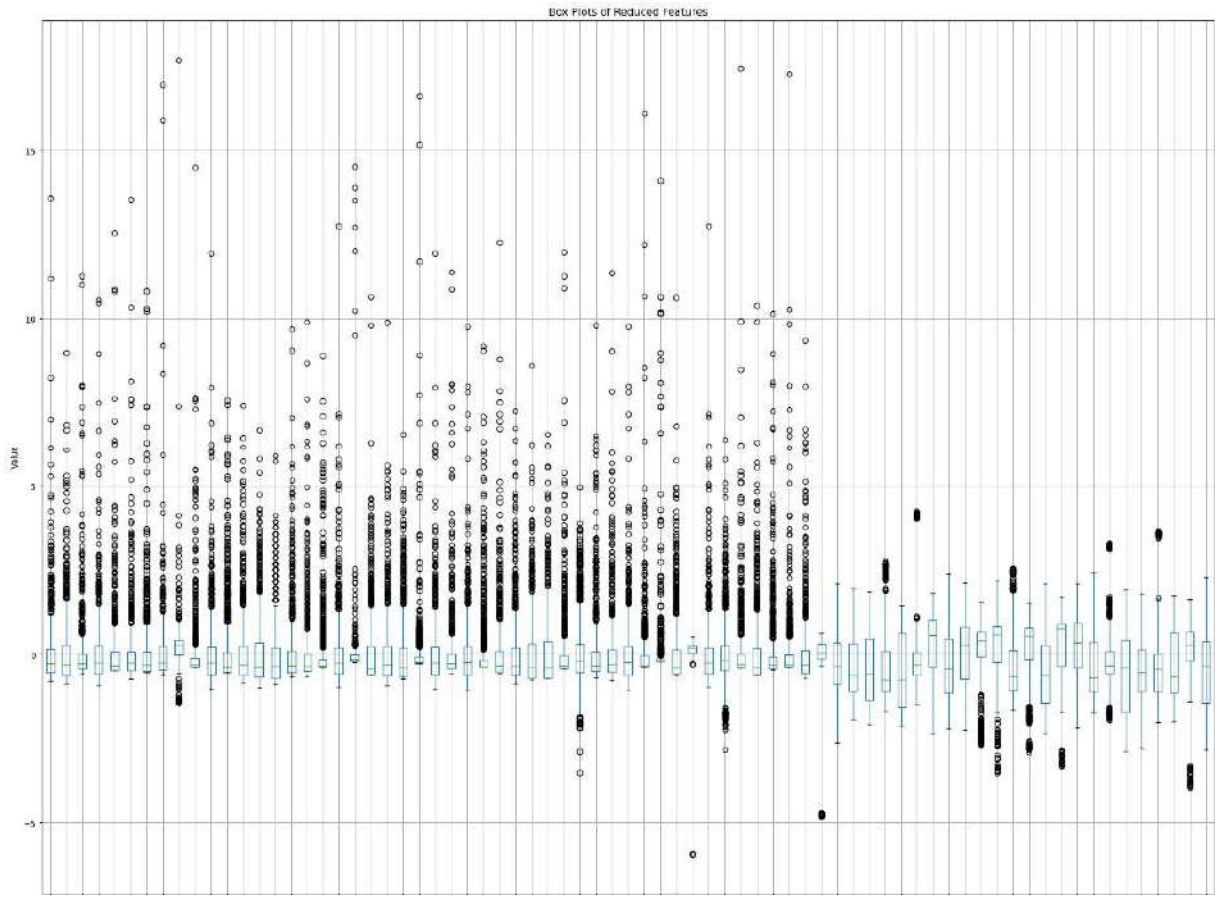


Figure 4.14: Box plots of reduced features.

In the case of the 25 UMAP-generated features, represented by the last 25 box plots in Figure 4.14, the box plots reveal that these features tend to be more centered around zero and exhibit fewer extreme outliers compared to the non-reduced features. This observation suggests that the UMAP dimensionality reduction technique has successfully transformed the original high-dimensional data into a more compact representation while preserving essential information [McInnes et al. (2020)]. It is important to note that both the reduced and non-reduced features were standardized using a Gaussian distribution. This preprocessing step ensures that all features have a mean of zero and a standard deviation of one, which is crucial for many machine learning algorithms that are sensitive to the scale of input features.

Note: After applying dimensionality reduction, the total number of remaining features is 73.

4.4.6 Feature Selection

Feature selection is a critical step in building successful machine learning models, as it helps identify important variables while eliminating irrelevant or redundant ones. In this study, several feature selection methods were utilized to select relevant features for Germany as the Test country and charging station number as the variable to disaggregate, as discussed previously.

The mathematical foundations and practical applications of these feature selection methods will be explored further below.

Correlation Matrix-based Feature Selection

The correlation matrix is an invaluable tool for discovering relationships between variables. It calculates pairwise correlation coefficients between all pairs of features in a dataset, producing a square matrix. Features with correlation coefficients above a pre-set threshold are considered highly correlated; to reduce redundancy, only those features above this level are retained in the dataset.

The Pearson correlation coefficient serves as the foundation of the correlation matrix method, quantifying the linear relationship between two variables [Rodgers und Nicewander (1988)]. This coefficient can range from -1 (perfect negative correlation) to 1 (perfect positive correlation), with zero signifying no correlation.

In order to calculate the Pearson correlation coefficient, the following formula is used:

$$r = \frac{\Sigma(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\Sigma(X_i - \bar{X})^2(Y_i - \bar{Y})^2}} \tag{4.4}$$

where r represents the Pearson correlation coefficient, X_i and Y_i are individual data points for variables X and Y, $\bar{X}$ and $\bar{Y}$ are the mean values of variables X and Y, and Σ denotes the summation. The numerator of the formula represents the covariance between variables X and Y, while the denominator contains the product of the standard deviations of variables X and Y. This formula effectively standardizes the covariance, making the Pearson correlation coefficient dimensionless and bounded between -1 and 1 [Draper und Smith (1998)].

This approach helps identify the strength of association between variables by taking into account both positive and negative relationships. It assists in recognizing pairs with strong linear connections within the data structure, which helps inform feature selection decisions. A threshold of 0.5 was set so that only features with Pearson correlation coefficients higher than 0.5 would be selected; as can be seen in figure 4.15, only five features exhibit this level of correlation.

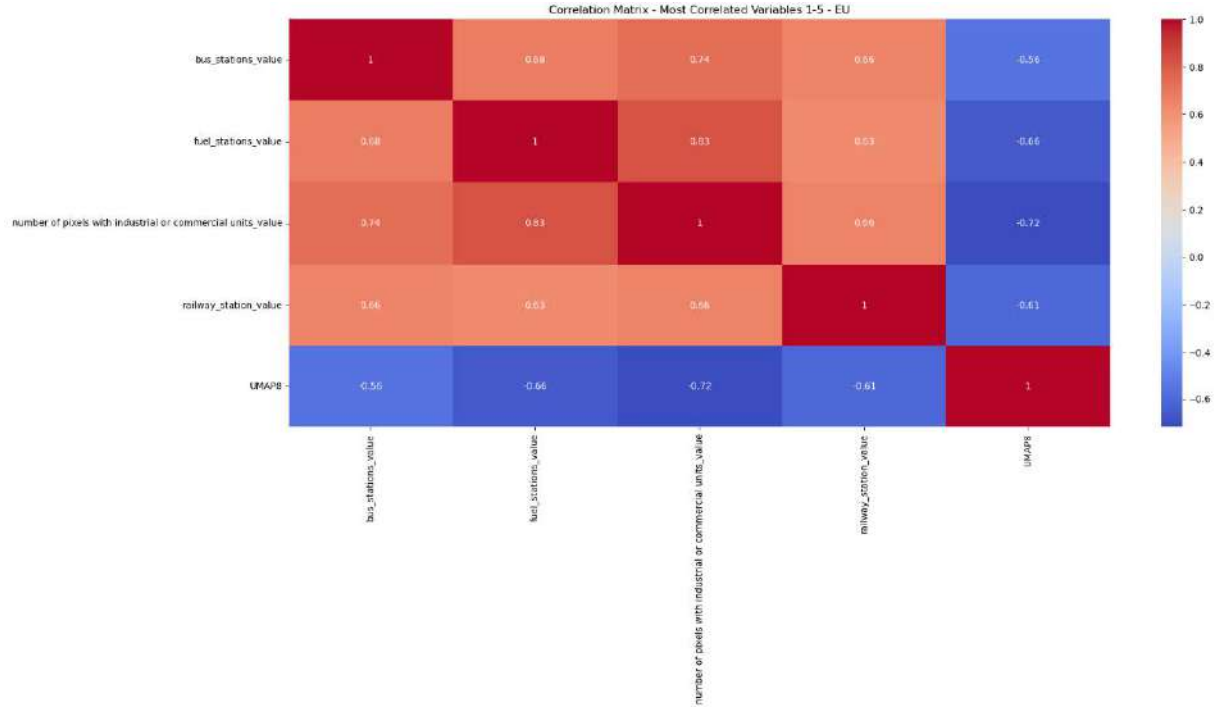


Figure 4.15: Matrix Correlation.

Univariate Feature Selection: F-Regression

Univariate feature selection using F-regression involves computing the F-statistic for each feature separately, in order to assess the strength of the relationship between it and a target variable. The formula for performing F-regression can be found at [Draper und Smith (1998)], as follows:

$$F = \frac{\frac{SSR}{k}}{\frac{SSE}{n-k-1}} \tag{4.5}$$

where F represents the F-statistic, SSR is the sum of squares for regression, SSE is the sum of squares for the residual error, k is the number of independent variables (features), and n is the total number of observations.

The sum of squares for regression (SSR) and residual error (SSE) are statistical measures used to evaluate the performance of a linear regression model. They help identify what portion of total variability in a dependent variable can be explained by the model, as well as what remains unexplained.

The sum of squares for regression (SSR) measures the variation in a dependent variable explained by its independent variables in a model. It represents the difference between the predicted values from the model and the mean of the dependent variable. A larger SSR indicates a stronger relationship between the independent variables and the dependent variable, suggesting that the model is effectively capturing the underlying pattern in the data.

The sum of squares for the residual error (SSE) represents the unexplained variation in the dependent variable. It is the difference between the actual values of the dependent variable and

the predicted values from the model. A smaller SSE indicates that the model's predictions are closer to the actual values, suggesting better model performance.

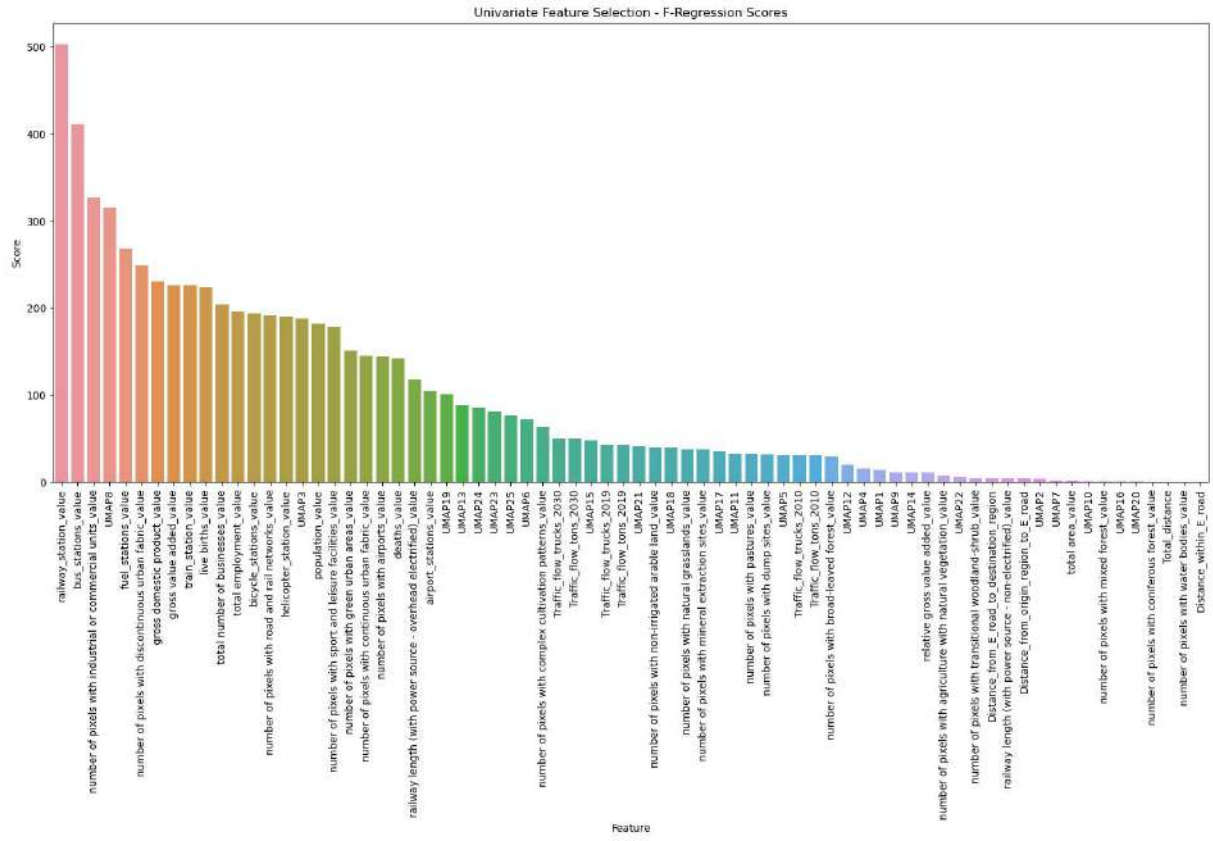


Figure 4.16: Univariate Feature Selection. F - Regression Score.

Figure 4.16 displays the F-regression scores for features in the dataset. By observing these scores, it becomes evident that certain features have a stronger linear relationship with the target variable than others. The top 5 features with highest F-regression scores are selected for further analysis as they have the strongest association with this variable. This approach provides an easy means of recognizing key characteristics within a dataset that have an established linear connection to the desired variable.

Univariate Feature Selection: Mutual Information

Univariate feature selection using mutual information is a technique that calculates a score for each feature independently to assess the strength of the association between the feature and a target variable. Unlike F-regression, which focuses on linear relationships, mutual information captures both linear and non-linear dependencies between variables. This measure quantifies the reduction in uncertainty about one variable when the value of another variable is known, indicating the extent to which information about one variable can be gained by observing the other variable [Cover und Thomas (2006)].

The mutual information (MI) between two variables X and Y can be defined as:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{4.6}$$

where $p(x,y)$ is the joint probability distribution of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distributions of X and Y, respectively [Cover und Thomas (2006)]. Higher MI values indicate a stronger association between the feature and the target variable, making them more relevant for further analysis.

Mutual information can capture both linear and non-linear relationships between variables, making it a versatile metric for feature selection in various problem domains. It is particularly useful in cases where the relationship between the feature and the target variable is not necessarily linear.

The top five features with the highest mutual information scores are chosen as they demonstrate the strongest association with the target variable. This method provides a straightforward means of recognizing key attributes within a dataset that have an established mutual dependence with another variable, whether this relationship is linear or non-linear.

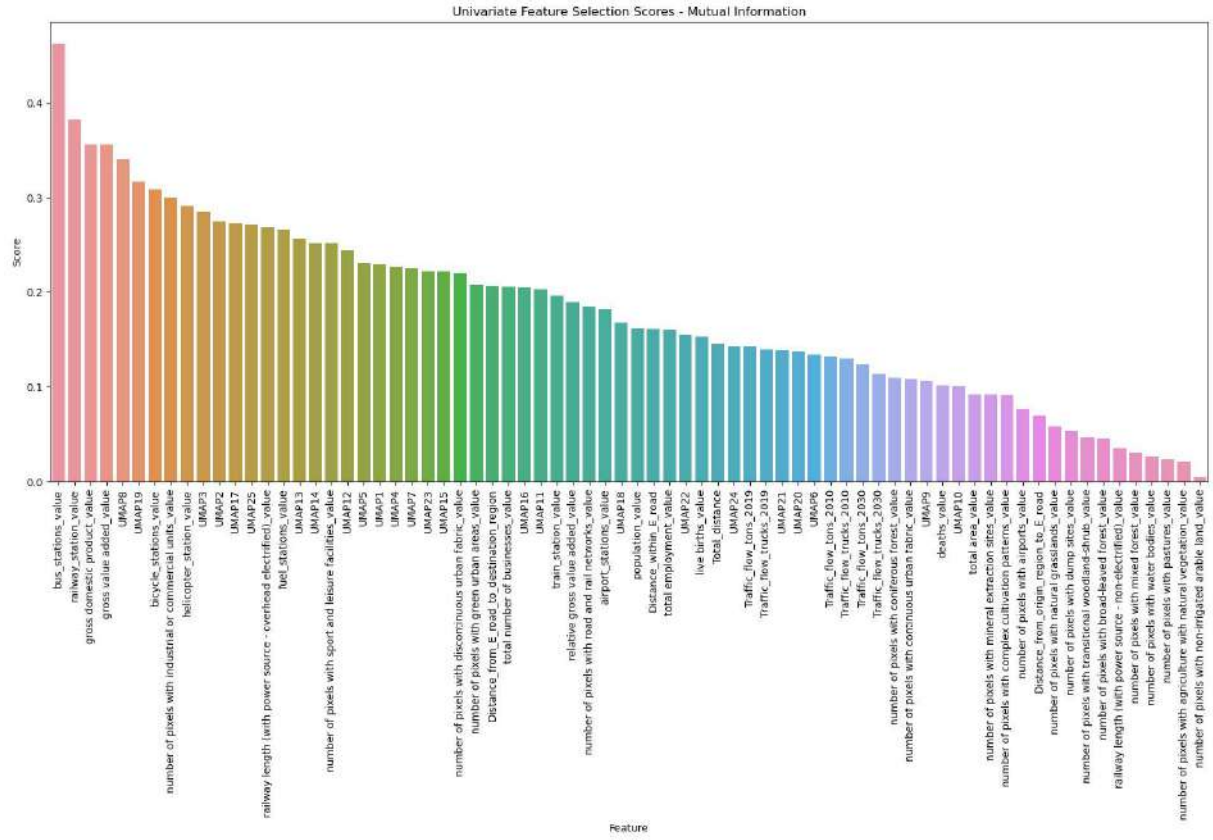


Figure 4.17: Univariate feature selection. Mutual information.

Figure 4.17 displays the mutual information scores for features in the dataset.

Feature Selection using Ridge (L2 Regularization)

Feature selection using Ridge regression (L2 regularization) is a technique designed to identify important features in a dataset by analyzing their relationship with an objective variable through linear regression. The Ridge regression model introduces a penalty term into the ordinary least squares (OLS) linear regression equation, helping control model complexity, prevent overfitting, and implicitly perform feature selection [Hoerl und Kennard (1970)].

Ridge regression is particularly useful when features in a dataset are correlated, as it helps distribute the weights of features more evenly, making it easier to identify the most significant ones. In this context, "Ridge" refers to an added penalty term in the model.

The Ridge regression model can be defined as follows:

$$\text{Ridge} = \text{RSS} + \lambda \sum_i \beta_i^2 \tag{4.7}$$

where Ridge represents the Ridge regression model, RSS denotes the residual sum of squares, λ is the regularization parameter, and β_i are the model coefficients. The residual sum of squares (RSS) is a measure of the discrepancy between the actual values of the dependent variable and the predicted values from the model, similar to the sum of squares for the residual error (SSE) mentioned earlier.

Higher values of λ lead to stronger regularization and further shrinkage of the coefficients while lower values produce weaker regularization and less shrinkage [Hoerl und Kennard (1970)]. The shrinkage serves as an implicit feature selection process, since less important features will see their coefficients reduced closer to zero, signifying their lower significance within the model.

The regularization parameter λ is essential in striking a balance between model complexity and performance. Selecting an optimal value for λ requires cross-validation, which involves fitting the model onto different subsets of data and evaluating its performance across these subsets.

The top eight features with the highest Ridge regression coefficients are chosen for further analysis, as they demonstrate the strongest relationship with the target variable. This approach helps identify key attributes within a dataset that have an important linear relationship to this desired variable while also taking into account regularization's impact on feature importance.

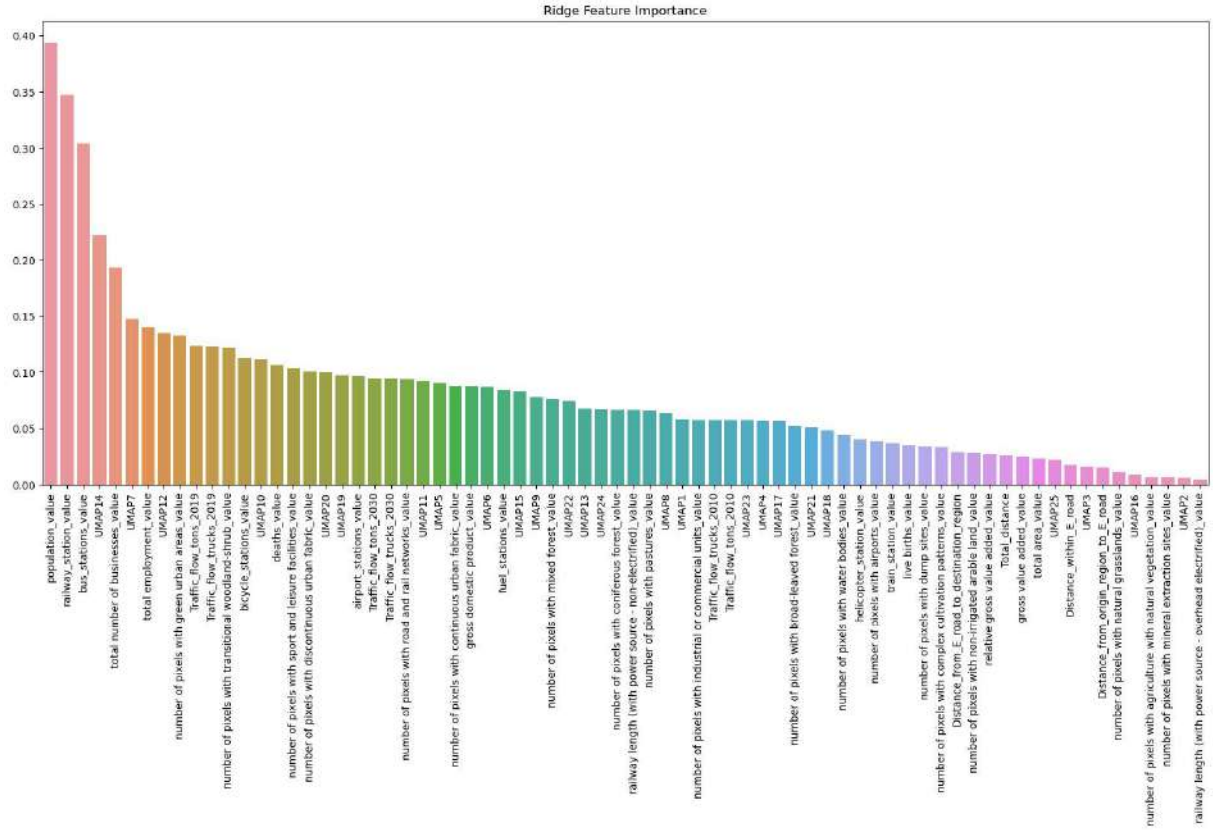


Figure 4.18: Ridge Regression. Feature Selection.

Figure 4.18 displays a Ridge regression model showing feature coefficients.

Feature Selection using Random Forest Importance

Random Forest Importance Feature Selection is a technique that recognizes important features in a dataset by analyzing their relation to an objective variable using the Random Forest model [Breiman (2001)], an ensemble learning approach based on decision trees. Random Forests create multiple decision trees during training and then combine their individual predictions for more precise and reliable final predictions.

A Random Forest model's importance can be assessed through either a decrease in Gini impurity or the mean decrease in accuracy when that feature is used for splitting nodes [Breiman (2001)]. A higher importance score suggests the feature has more relevance within the model and contributes more accurately to predicting the target variable accurately.

The process of determining feature importance in a Random Forest model involves the following steps:

1. Train a Random Forest model on the dataset.

2. Calculate the average decrease in the Gini impurity or the mean decrease in accuracy for each feature across all decision trees in the ensemble:

3. Average decrease in Gini impurity: Gini impurity is a measure of how often a randomly chosen element from a set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. The Gini impurity for a node in a decision tree can be calculated as:

$$\text{Gini Impurity} = 1 - \sum_i p_i^2 \tag{4.8}$$

where p_i is the proportion of samples in class i at that node.

To compute the average decrease in Gini impurity for a feature, the following steps are taken:

a. Calculate the Gini impurity for each internal node that splits on the feature.

b. Compute the decrease in Gini impurity for each split, which is the difference between the Gini impurity of the parent node and the weighted sum of the Gini impurities of the child nodes.

c. Sum the decreases in Gini impurity for all splits on the feature across all decision trees in the ensemble.

d. Divide the sum by the total number of decision trees to get the average decrease in Gini impurity for the feature.

Mean decrease in accuracy: The mean decrease in accuracy for a feature is calculated by following these steps:

a. For each decision tree in the ensemble, randomly permute the values of the feature and use the resulting dataset to make predictions.

b. Calculate the decrease in accuracy for the feature in each tree, which is the difference between the model's accuracy on the original dataset and its accuracy on the dataset with the permuted feature.

c. Sum the decreases in accuracy for the feature across all decision trees in the ensemble.

d. Divide the sum by the total number of decision trees to get the mean decrease in accuracy for the feature.

e. Rank the features based on their importance scores.

Random Forest Importance is especially useful for datasets with complex relationships between features and target variables, as it can capture both linear and nonlinear patterns. Furthermore, its robustness to overfitting and ability to handle multicollinearity among features make it a versatile method for feature selection across many problem domains.

The top seven features with the highest Random Forest importance scores are chosen for further analysis, as they exhibit the strongest association with the target variable. This approach helps identify key attributes within a dataset that have an important connection to a desired variable through ensemble learning and feature importance ranking provided by the Random Forest algorithm.

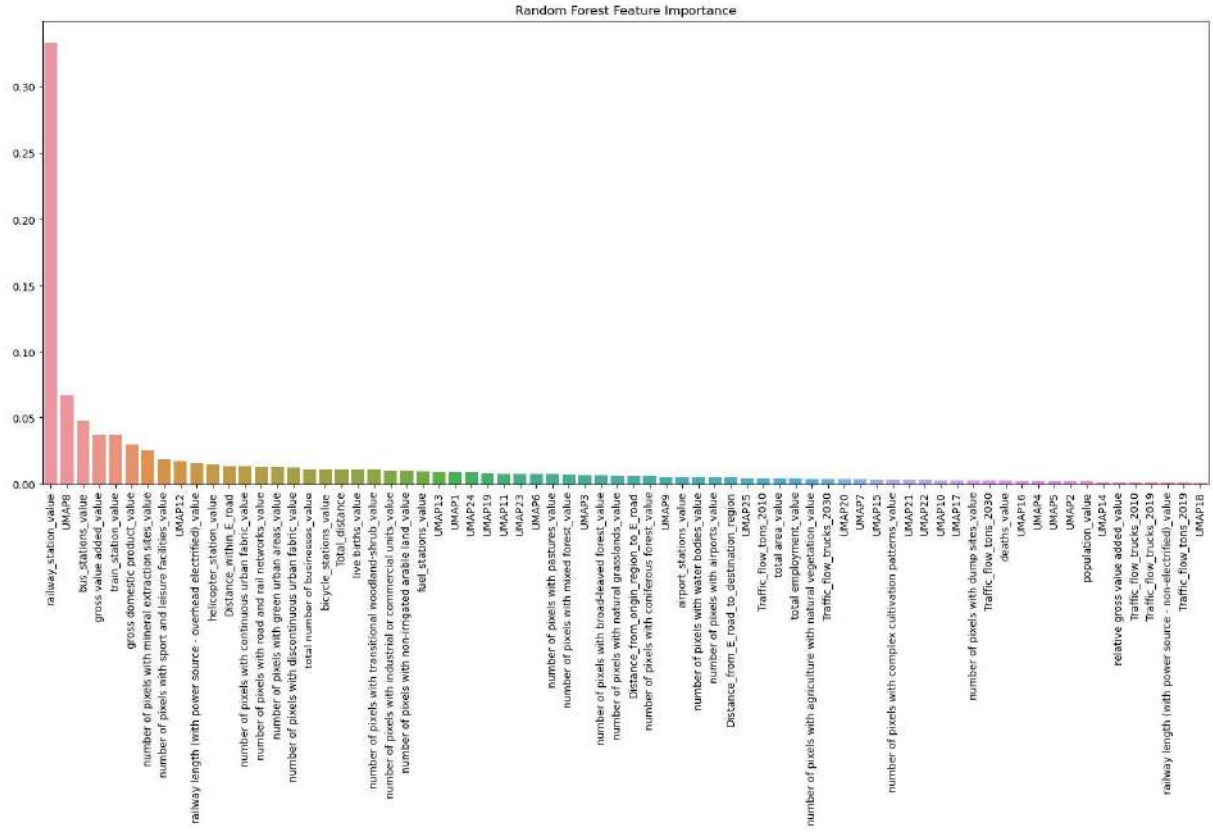


Figure 4.19: Random Forest. Feature Selection.

Figure 4.19 displays the feature importance scores generated from a Random Forest model for the given dataset.

Combining Selected Features from Different Methods

By combining the selected features from different methods, the model benefits from the strengths of each individual method, resulting in a more robust and diverse feature set.

The final set of selected features is obtained by taking the union of the features selected by each method.

The total number of selected features is 19. The selected features from each method are presented in Table A.1 in section A.1.

Note: Features named UMAPX are the ones created with the dimensionality reduction approach. In this case, UMAP3, UMAP7, UMAP8, UMAP12, UMAP14, and UMAP19 are the features obtained through dimensionality reduction. Out of the initial 73 features, 19 were selected, representing a reduction of approximately 74% in the number of features. The 25 UMAP features make up about 34% (25 out of 73) of the initial features. The presence of UMAP features in the selected list accounts for approximately 32% (6 out of 19) of the total selected features, indicating that the dimensionality reduction approach provides valuable insights and contributes to the overall feature selection process. This highlights the effectiveness of combining

traditional feature selection methods with dimensionality reduction techniques to extract relevant information from high-dimensional data.

Performing Cross-Validation with All and Selected Features for Each Model

This section evaluates the performance of various regression models using both their full set of features and a reduced set of selected ones. The goal is to assess whether feature selection has an impact on models' predictive abilities and see if enhanced features do indeed enhance them. The models included in this analysis include:

1. Linear Regression

2. Decision Tree

3. Random Forest

4. Gradient Boosting

5. Support Vector

6. Ridge Regression

7. Lasso Regression

8. Elastic Net Regression

9. K-Nearest Neighbor Regression.

10. MLP Regressor

A 5-fold cross-validation technique is employed to estimate the performance of each model. As mentioned previously, cross-validation is an indispensable approach for testing machine learning models' generalization ability by splitting the dataset into multiple subsets and iteratively using one subset for testing and the others for training purposes.

The cross-validation score calculated for each model represents its performance in predicting the target variable. Mathematically, the score is the average of the model's performance on each of the folds during cross-validation. Conceptually, a higher score indicates a better fit of the model to the data, meaning that the model can generalize well to unseen data. It is important to note that the score's interpretation depends on the metric used. In this case, the default scoring metric for regression models in scikit-learn is the coefficient of determination (R-squared), which ranges from $-\infty$ to 1. A higher R-squared value indicates a better fit between the model's predictions and the actual target values [Draper und Smith (1998), Rodgers und Nicewander (1988)].

Table [4.3] displays the cross-validation scores for each model using both all features and selected features, as well as an average score across all models.

The average cross-validation score improved to 38.8% with selected features compared to all features, indicating successful feature selection. Models such as Linear Regression and MLP Regressor showed significant improvements, suggesting reduced noise and more informative representations; however, some models displayed decreased performance which may need further tuning or adaptation for optimal performance.

Model	All Features Score	Selected Features Score
Linear Regression	-0.0651	0.2528
Decision Tree	-0.1728	-0.1706
Random Forest	0.3737	0.3317
Gradient Boosting	0.3352	0.2741
Support Vector	0.3702	0.3799
Ridge Regression	0.2807	0.2459
Lasso Regression	0.1096	0.2463
Elastic Net Regression	0.2809	0.2459
K-Nearest Neighbors	0.2851	0.3033
MLP Regressor	0.1567	0.3398
Average	**0.1497**	**0.2079**

Table 4.3: CV Scores with selected features.

Overall, feature selection can improve various models' performances by focusing on relevant predictors, leading to precise results, reduced overfitting, and lower computational costs. It is essential to take into account each model's individual characteristics when applying feature selection as some may require further adjustments with a smaller feature set.

4.5 Clustering

Clustering, an unsupervised machine learning method, groups data points into clusters based on their similarities. Various clustering algorithms, including Agglomerative Clustering, DBSCAN, and K-means, each offer unique advantages and drawbacks. The choice of the most suitable algorithm depends on the specific problem and the available data.

In the analysis, Agglomerative Clustering, DBSCAN and K-means algorithms were utilized to investigate different clustering patterns within the data. Each method is chosen for its unique advantages so as not to bias results towards one type of clustering technique.

Agglomerative Clustering provides a hierarchical structure which allows data exploration at various granularities [Rousseeuw (1987)], while DBSCAN efficiently handles heterogeneous and nonlinear datasets by identifying clusters of various shapes and sizes as well as noise points [Ester et al. (1996)]. K-means is well known for its simplicity and efficiency when dealing with large datasets making it an attractive choice when dealing with computationally demanding tasks.

To optimize the clustering process, a search is conducted to find the parameters that would yield the highest scores and most appropriate number of clusters. For each method, multiple parameter

values are tested, then its performance assessed using two popular evaluation metrics: silhouette score and Calinski-Harabasz score.

4.5.1 Silhouette Score

The silhouette score is a commonly-used metric to assess clustering results, measuring how well-defined the clusters are within a dataset [Rousseeuw (1987)]. Calculated individually for each data point, this score ranges from -1 to 1. A higher score indicates that a point has good match with its own cluster and poorly matches neighboring ones; conversely, a negative value could suggest incorrect assignment of that same data point into another cluster.

Mathematically, the silhouette score for a data point i is determined by applying the following formula:

$$s(i) = \frac{b(i) - a(i)}{\max a(i), b(i)} \tag{4.9}$$

where $a(i)$ represents the average distance between data point i and all other data points in the same cluster, and $b(i)$ is the minimum average distance between data point i and all data points in any other cluster. The overall silhouette score is obtained by averaging the silhouette scores of all data points in the dataset.

The silhouette score is valuable because it can be easily visualized, allowing for intuitive interpretation and validation of clustering results.

4.5.2 Calinski-Harabasz score

The Calinski-Harabasz score, commonly referred to as the Variance Ratio Criterion, is another metric used to assess clustering results [Caliński und Harabasz (1974)]. This score measures the ratio between cluster dispersion and within-cluster dispersion; higher scores indicate denser and more separated clusters.

Mathematically, the Calinski-Harabasz score is defined as:

$$CH(k) = \frac{B/(k-1)}{W/(n-k)} \tag{4.10}$$

where B is the between-cluster dispersion, W is the within-cluster dispersion, k is the number of clusters, and n is the number of data points in the dataset. The between-cluster dispersion is the sum of squared distances between cluster centers and the overall data mean, while the within-cluster dispersion is the sum of squared distances between data points and their corresponding cluster centers.

The Calinski-Harabasz score can be particularly useful when the number of clusters is unknown a priori, as it helps identify the optimal number by maximizing its score.

4.5.3 Optimal Hierarchical Clustering with Agglomerative Clustering

Agglomerative Clustering is a hierarchical clustering method that creates a tree-like structure by iteratively merging clusters until all data points belong to one group [Jain und Dubes (1988)]. The algorithm starts with each data point as its own singleton cluster and, at each step, merges the closest pair of clusters, thus decreasing their number by one. This process continues until either the desired number of clusters is achieved, or some termination criterion is met.

Based on the silhouette score [Rousseeuw (1987)], two clusters with a score of 0.59 were found to be optimal parameters; similarly, using Calinski-Harabasz index [Caliński und Harabasz (1974)] results revealed 2 clusters at 475.84.

Cluster analysis revealed the shape of the first cluster as (1077, 26), while the second had a shape of (93, 26). The smaller size of this second cluster - comprising 7.94% of total data points - indicates a distinct subgroup within the dataset. This small cluster could potentially represent some unique pattern or characteristic that sets it apart from its larger counterpart, and further investigation could be conducted to establish its significance and relevance within the problem domain.

Three figures are generated to visualize and support the clustering results.

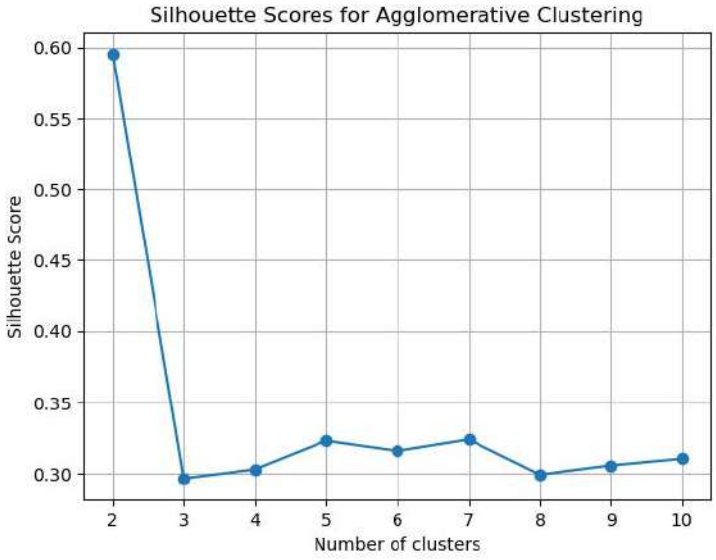


Figure 4.20: Silhouette Score Plot for Hierarchical Clustering.

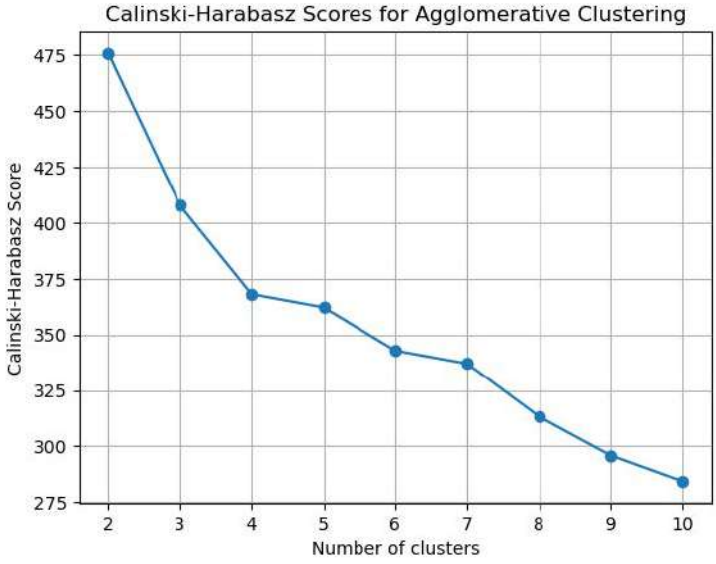


Figure 4.21: Calinski-Harabasz Index Plot for Hierarchical Clustering.

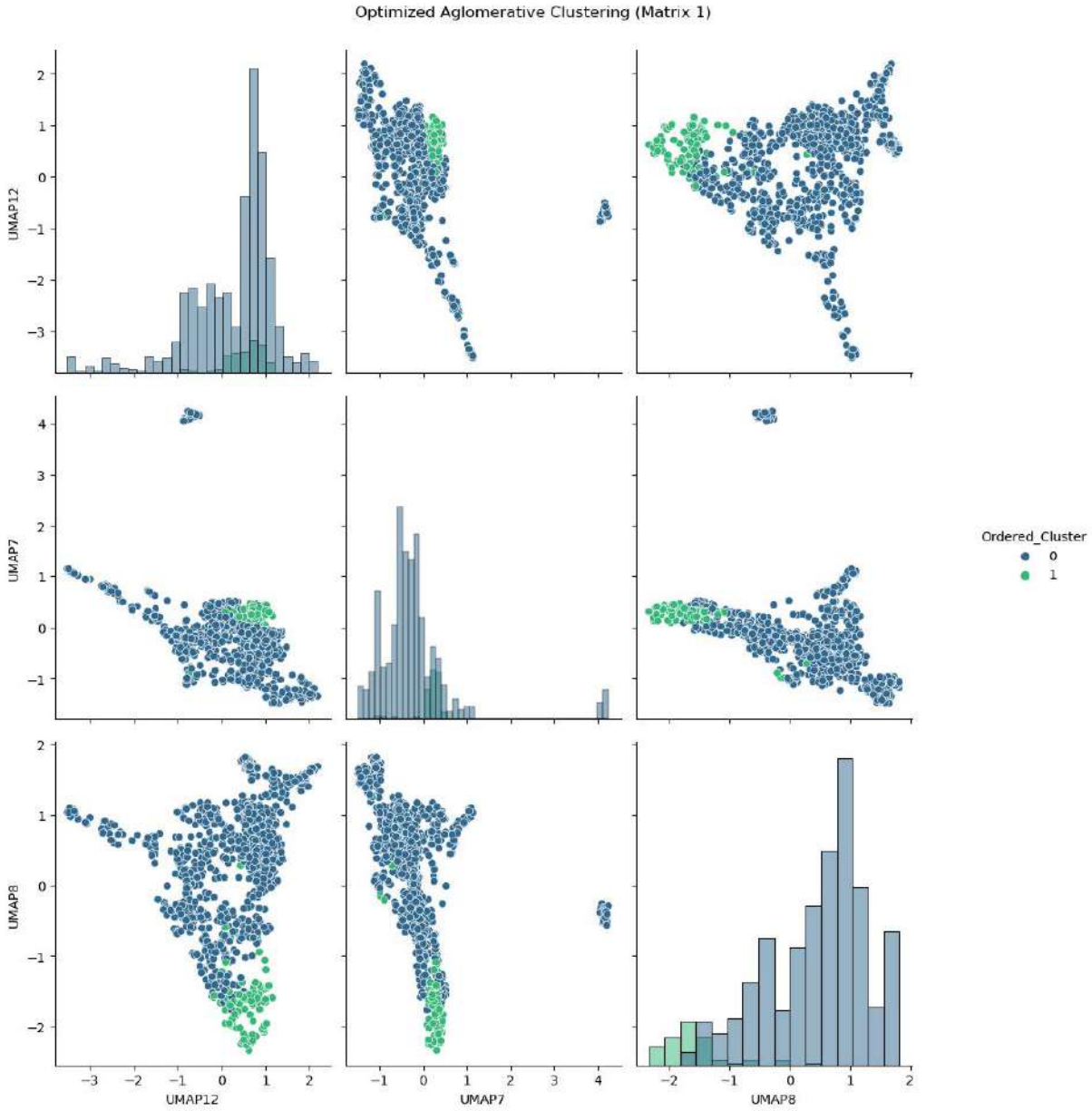


Figure 4.22: Optimized Agglomerative Clustering Pairwise Feature Scatterplot Matrices.

Figure 4.20 depicts silhouette scores for various cluster counts while Figure 4.21 displays Calinski-Harabasz scores according to cluster counts. These plots helped to select an optimal number of clusters for applying Agglomerative Clustering algorithm.

Figure 4.22, Optimized Agglomerative Clustering Pairwise Feature Scatterplot Matrices, displays scatterplot matrices of pairwise relationships among randomly selected features for the clusters.

4.5.4 Optimal Density-Based Clustering with DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a widely used clustering algorithm that works based on data point density in a feature space [Ester et al. (1996)].

Unlike partition-based methods like K-means or hierarchical methods like Agglomerative Clustering, DBSCAN does not require predefined clusters but instead recognizes them as regions of high point density separated from lower densities. Furthermore, DBSCAN is capable of detecting noise - an invaluable feature when working with datasets containing outliers.

The two primary parameters used in DBSCAN are epsilon (ε) and min_samples. ε measures the maximum distance between data points to qualify them as neighbors, while min_samples indicate the minimal number of required points to form a dense region or cluster. When determining optimal values for these parameters, silhouette score can be used because it accurately assesses cluster quality by evaluating separation between clusters and compactness within each cluster.

In this example, an epsilon value of 9.0 and a minimum sample size requirement of 2 are chosen, leading to an optimal silhouette score of 0.81. While the Calinski-Harabasz index suggested 3 clusters with a score of 315, the silhouette score was prioritized in selection because it provides more insight into intra-cluster cohesion and separation between clusters [Kaufman und Rousseeuw (2009)], thus helping determine an optimal number of clusters.

Cluster analysis revealed the shape of the first cluster as (1167, 26), while the second had a shape of (3, 26). Although only 0.26% of total data points reside in this second cluster, its small size could suggest noise or outliers within the dataset. Therefore, further investigation of this small cluster is recommended in order to understand its significance and potential effect on overall outcomes.

Three figures are created to visualize and support our clustering results. Figure 4.23 depicts the silhouette score plot.

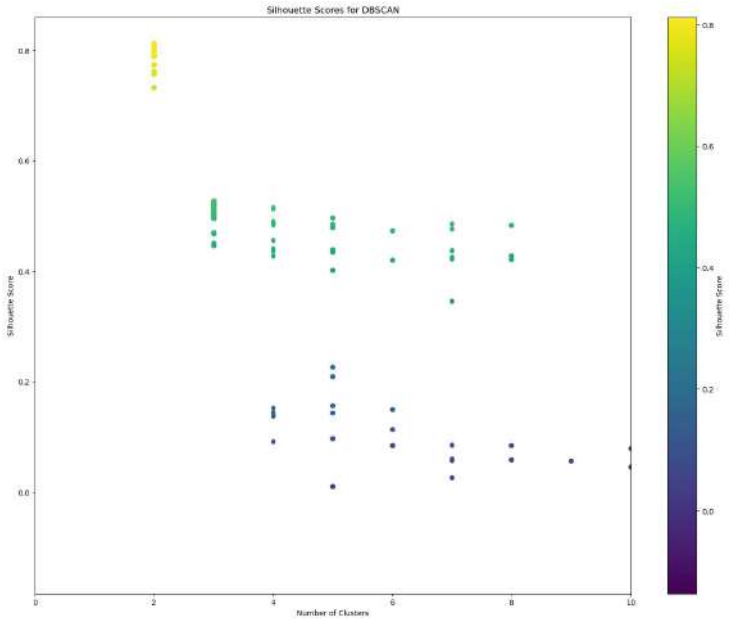


Figure 4.23: DBSCAN Silhouette Score Plot.

Additional visuals, like the Calinski-Harabasz scores plots for different parameter combinations (Figure A.2 in the Appendix), guided optimal parameter selection for the DBSCAN algorithm.

The DBSCAN Pairwise Feature Scatterplot Matrices, exhibited in Figure A.1 in the Appendix, illustrate pairwise relationships among selected features for the clusters.

4.5.5 Optimal Partitioning-Based Clustering with K-Means

K-Means is a widely employed partitioning-based clustering method [MacQueen (1967)]. Its goal is to partition the data into a pre-specified number of clusters (k) by minimizing the within-cluster sum of squared distances. The algorithm iteratively assigns data points to the closest cluster center (centroid) and updates the centroids based on the mean of the points within each cluster until convergence is achieved.

The mathematical objective of the K-Means algorithm can be expressed as:

$$\min \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2 \tag{4.11}$$

Where k is the number of clusters, C_i represents the i^{th} cluster, x denotes the data points, and μ_i is the centroid of the i^{th} cluster.

Three figures were generated to visualize and support the clustering results.

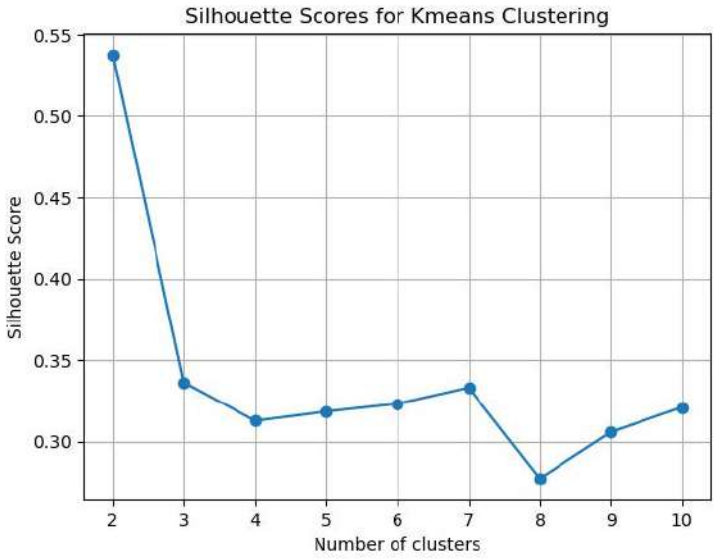


Figure 4.24: K-Means Silhouette Score Plot.

The silhouette score identified two clusters as optimal, yielding a score of 0.5376. The shape of the clusters were (1026, 26) and (144, 26), respectively, with the latter's smaller size indicating a distinct subset, comprising 12.34% of total data points.

The optimal cluster count for the K-Means algorithm was informed by the Calinski-Harabasz scores for different numbers of clusters, as visualized in Figure A.4 in section A.2.

Visual representation of pairwise relationships among random features in the clusters was facilitated by the K-Means Pairwise Feature Scatterplot Matrices in Figure A.3 in section A.2.

4.5.6 Clustering Results Summary

This section summarizes the findings from three clustering techniques used in this analysis: Hierarchical, K-Means, and DBSCAN. Based on the best number of clusters, best silhouette scores, and best Calinski-Harabasz scores; cluster shapes and percentages of data points per cluster are presented in a comparative manner.

Technique	Best Number of Clusters	Silhouette Score	Cluster Shapes	Percentage of Data Points per Cluster
Hierarchical	2	0.59	(1077, 26), (93, 26)	91.31%, 8.69%
K-Means	**2**	**0.54**	**(1026, 26), (144, 26)**	**87.69%, 12.31%**
DBSCAN	2	0.81	(1167, 26), (3, 26)	99.74%, 0.26%

Table 4.4: Clustering Results with Selected Features.

Each clustering technique revealed distinct patterns and structures within the data. Hierarchical and K-Means clustering both identified two clusters with slightly different distributions of data points; DBSCAN detected a highly concentrated main cluster and an extremely small secondary one.

4.6 Model Selection

The literature review indicates that machine learning (ML) algorithms, such as Artificial Neural Networks (ANNs) [Bishop (2006)] and Random Forests (RFs) [Breiman (2001)], have significant potential to enhance spatial disaggregation techniques. ANNs, which are deep learning algorithms inspired by the structure and function of the human brain, consist of interconnected nodes or "neurons" processing and transmitting information [Goodfellow et al. (2016)]. ANNs possess a great flexibility and can recognize patterns in large datasets, making them suitable for a range of spatial modeling tasks.

RFs are an ensemble-based algorithm capable of handling large datasets and capturing complex nonlinear relationships between input and output variables. They consist of multiple decision trees, each trained on a different subset of data to produce a more accurate final prediction than any single decision tree could.

In addition to ANNs and RFs, Support Vector Machines (SVMs) could also be considered as another option for spatial disaggregation [Cortes und Vapnik (1995)]. SVMs are supervised learning models with the ability to perform both linear and nonlinear regression and classification tasks. They are known for their ability to handle high-dimensional data and to find optimal separation between classes or regression targets, making them suitable for spatial modeling applications.

Monteiro et al. [Monteiro et al. (2019)] employed a Convolutional Neural Network (CNN) as they worked with data that included a spatial dimension - tensors. Tensors are mathematical objects that can be represented as multidimensional arrays; when applied to spatial data such as satellite imagery or remote sensing data, these dimensions could include height, width, and spectral bands.

Contrarily, in this study, tabular data was utilized, which is better suited for ANNs [Bishop (2006)]. Tabular data consists of rows and columns of numerical or categorical information, such as the number of charging stations or socioeconomic variables.

One significant advantage of deep learning-based models such as ANNs, CNNs, RFs, and SVMs for spatial disaggregation is their ability to learn complex relationships between input and output variables without requiring prior knowledge of underlying physical processes. This makes them exceedingly useful in spatial modeling and analysis.

4.6.1 Fundamentals of Neural Networks for Regression

Single-layer perceptron

A single-layer perceptron, illustrated in Figure 4.25, represents the most rudimentary form of a neural network employed for regression problems [Rosenblatt (1958)]. It consists of a single input layer and an output node, with the number of input nodes being equal to the length of the feature vector X. Each input node has an associated weight value (wi) and a bias value (b). Generally, each training sample takes the form (X, y), where X = [x1, ..., xd] is the feature vector and y are the ground truth value. During training, the weight matrix is iteratively updated using a loss function that gauges the difference between the predicted and ground truth values. The output signal is generated through two computational steps: first, the weight matrix is dot-product with the feature matrix; then, the aggregated signal is passed through an activation function (Act) to yield the output value y:

$$y = Act(w \cdot X + b) \tag{4.12}$$

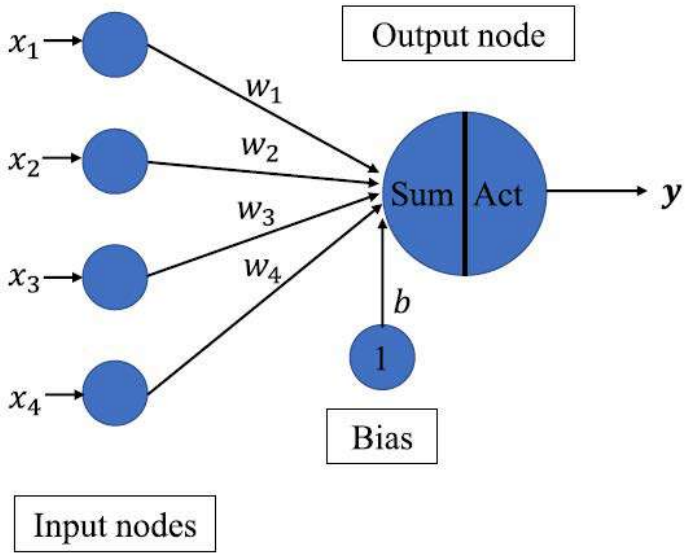


Figure 4.25: Perceptron architecture.

Common activation functions, such as ReLU (Rectified Linear Unit), htan, or sigmoid functions, introduce non-linearity to the model and are crucial for optimal performance since many real-world systems exhibit highly non-linear behavior [Goodfellow et al. (2016)]. In neural network (NN) regression problems, the goal is to predict a continuous output value based on a set of input features. Unlike classification problems with binary or categorical outputs, regression problems necessitate the model to generate a continuous value accurately representing the output variable [Bishop (2006)]. Consequently, the output activation function must be linear in nature to ensure the output is directly proportional to the input features and enable precise prediction of the continuous output variable.

The above Latex equation represents the computation step in a single-layer perceptron where the weight matrix is dot-product with the feature matrix, then the aggregated signal is passed through an activation function (Act) to yield the output value y.

Multi-layer perceptron

A Multi-layer Perceptron (MLP) is an advanced neural network architecture composed of multiple interconnected layers, as shown in Figure 4.26. The input layer connects to one or more hidden layers, which in turn link to the output layer. Compared to single-layer perceptrons, MLPs can capture more intricate relationships between input and output variables. Training occurs using the backpropagation algorithm; this is an iterative process that adjusts weights in order to minimize a loss function [Goodfellow et al. (2016)]. Generally, hidden layers employ nonlinear activation functions like ReLU, while output layers use linear activation functions.

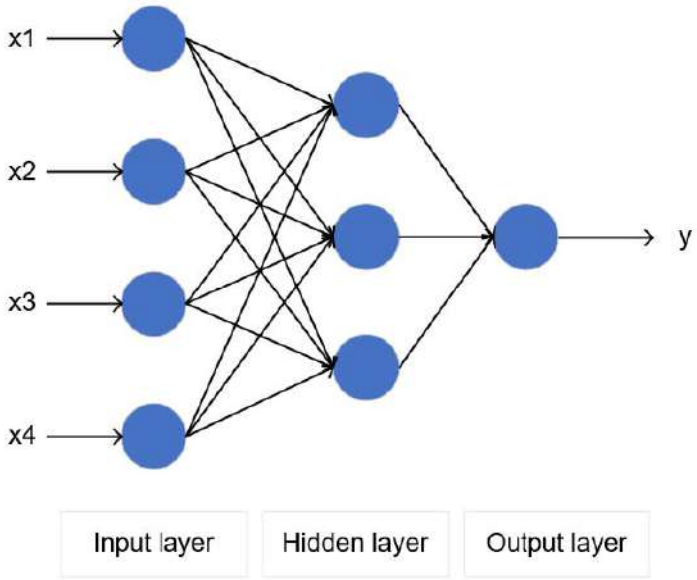


Figure 4.26: Multi-layer perceptron architecture used for regression.

This advanced architecture provides better model complexity compared to a single-layer perceptron, allowing the model to learn and predict more complex data patterns.

4.6.2 Random Forests for Regression

Random Forest Regression is a widely-used supervised learning algorithm that employs ensemble learning for regression tasks [Breiman (2001)]. Ensemble learning combines predictions from multiple machine learning algorithms to deliver more accurate predictions than a single model. In Random Forest Regression, numerous decision trees are constructed during training, and the final prediction is the mean of the output from these trees. This architecture enables trees to run in parallel, independent of each other, as illustrated in Figure 4.27.

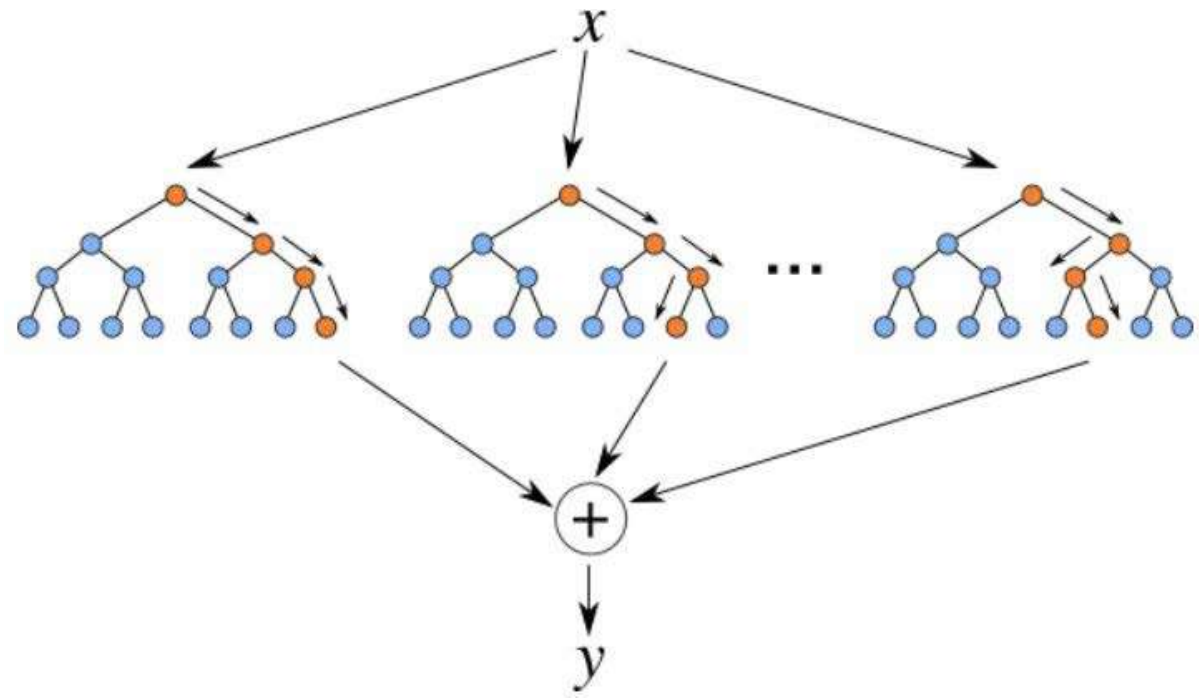


Figure 4.27: Random Forest architecture.

The Random Forest algorithm for regression progresses in the following steps [Breiman (2001)]:

1. Bootstrap Sampling: k data points are randomly selected from the training set with replacement, enabling each decision tree to be trained on a different dataset.

2. Random Feature Selection: The algorithm also randomly selects a subset of features for training each decision tree, preventing overfitting and enhancing model accuracy.

3. Building Decision Trees: Decision trees are created for each data subset, recursively dividing the data into smaller subsets based on feature values. The goal of each tree is to maximize information gain at each split, predicting the output variable as accurately as possible. Impurity of a node is usually measured using Gini impurity or entropy. This procedure is repeated N times to create N decision trees.

4. Combining Decision Trees: After constructing all decision trees, their outputs are merged to yield a final prediction. For a new data point, each of the N decision trees predicts the value of y for that point. The predicted y values are then averaged across all N trees to generate the final prediction.

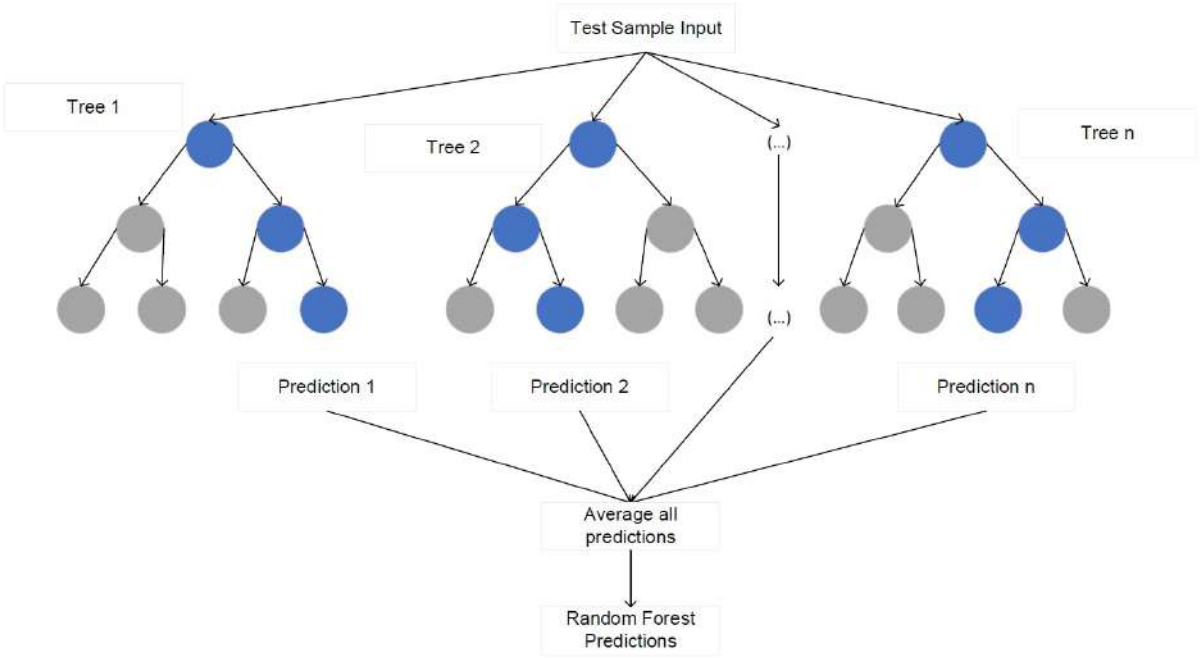


Figure 4.28: Random Forest Regression prediction.

Ensemble learning enhances the performance of decision trees and prevents overfitting by amalgamating predictions from several trees into a more precise final prediction, as depicted in Figure 4.28. Random Forest Regression trains each decision tree on a different subset of data and features, weighting each tree according to its accuracy; more accurate trees receive higher weights in the final prediction. Random Forest Regression is notable for its ability to manage high-dimensional datasets, accurately reflect complex nonlinear relationships between input and output variables, and avoid overfitting [Breiman (2001)]. Despite its efficiency in numerous regression problems involving non-linear feature relationships, drawbacks include lack of interpretability, the necessity to determine the number of trees in a model, and potential overfitting if not carefully tuned.

In conclusion, Random Forest Regression is an effective and precise algorithm for regression tasks.

4.6.3 Support Vector Machines for Regression

Support Vector Machine (SVM) Regression is an effective supervised learning algorithm commonly used for regression tasks. SVMs uniquely handle high-dimensional datasets, striving for optimal separation between classes or targets while minimizing prediction errors. In performing regression tasks, SVMs attempt to identify the best-fitting hyperplane that minimizes prediction errors [Drucker et al. (1997)], offering valuable insights.

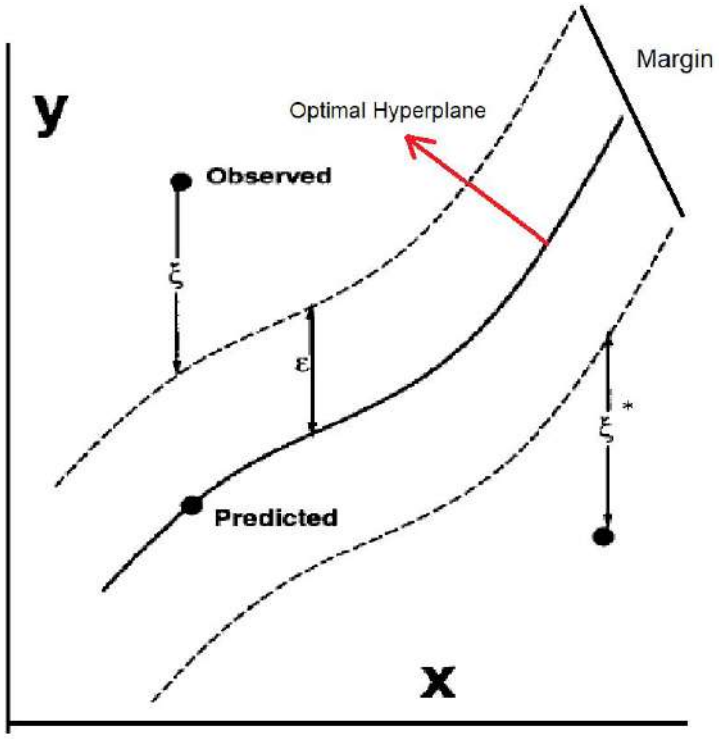


Figure 4.29: Support Vector for non-linear Regression.

Support Vector Regression (SVR), the algorithm for SVM regression, involves several steps:

1. Kernel Selection: A kernel function is chosen to transform input data into a higher-dimensional space, enabling the algorithm to better capture complex and nonlinear relationships between input and output variables. Common kernel functions include linear, polynomial, and radial basis function (RBF) kernels.

2. Determining the Epsilon Tube (ε): An epsilon value is determined to set the margin of error allowed for the regression model. The model strives to minimize prediction error within this epsilon-tube, containing the majority of training data points. The larger the epsilon value, the wider the margin around the regression hyperplane, allowing for more errors to fall within the epsilon tube.

3. Fine Tuning the Model: The SVM is trained by identifying the optimal hyperplane that separates data points with maximum margin while minimizing prediction error within an epsilon tube. Techniques such as quadratic programming or gradient descent may be employed depending on which approach is taken.

4. Predicting Output Values: With new input data points, the SVM utilizes its learned hyperplane to estimate corresponding output values.

Support Vector Machine Regression offers several advantages, such as its capacity for handling large datasets, capturing complex nonlinear relationships between input and output variables, and producing sparse models with excellent generalization performance.

5 Experimental Results

This chapter delves into the experimental results obtained throughout this thesis work. Both Master Model and clustering approaches are presented and analyzed. Preliminary results are discussed initially, followed by a more comprehensive examination.

5.1 Preliminary Results

An initial comparison was conducted between various advanced regression models, such as Support Vector Machine (SVM) Regression, Multi-Layer Perceptron (MLP), and Extreme Gradient Boosting (XGBoost). The primary goal was to identify the top-performing model, which would then receive fine-tuning and optimization to fully utilize self-supervised methodology for more precise spatial disaggregation solutions. The preliminary analysis focused on the data for Germany, with the aggregated target variable being the number of charging stations.

To assess the performance of each model, several metrics were employed, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R2) [Chai und Draxler (2014)]. MSE measures the average squared difference between predicted and actual values, emphasizing the impact of large errors. Unfortunately, since the error term is squared, it cannot be interpreted alongside its original data unit. Therefore, RMSE is used to address this limitation by measuring model prediction error using identical units as the original data.

MAE, on the other hand, calculates an average absolute difference between predicted and actual values, making it simpler to interpret. Unlike MSE or RMSE, MAE is less sensitive to outliers and large errors. This measure can be particularly useful when error distributions are non-symmetrical, and a more reliable measure of performance is needed [Chai und Draxler (2014)].

R-squared (R2) is a commonly used metric to estimate the proportion of variance in a dependent variable that can be explained by its independent variables. R2 values range between 0 and 1, with higher numbers indicating a better fit.

Comparing the performance of models against these metrics simplifies the process of identifying which one is best suited for further customization and analysis.

5.1.1 Multi-Layer Perceptron (MLP)

A preliminary MLP model with a simple architecture was used for the initial assessment. The architecture comprised multiple dense layers with a 'relu' activation function and He-normal initialization [Glorot et al. (2011)], along with batch normalization layers to ensure stable training. The model, built with the Sequential API and optimized using the Adam optimizer, exhibited a learning rate of 0.001.

A 5-fold cross-validation approach evaluated the model's performance. The results are summarized in Table 5.1.

Metric	**Train**	Test	Disaggregation
Mean Train RMSE	**1.9607**	-	-
Mean CV RMSE	**1.3240**	-	-
Mean Train MAE	**0.8014**	-	-
Mean CV MAE	**0.7982**	-	-
Mean Train R2	**-3.0394**	-	-
Test R2	-	-0.1376	-
Test MSE	-	1.3317	-
Test RMSE	-	1.1540	-
Disaggregation R2	-	-	-0.2599
Disaggregation MSE	-	-	4.7770
Disaggregation RMSE	-	-	2.1856

Table 5.1: MLP preliminary results.

The negative R2 value during training signifies that the model was unable to learn any significant patterns from the data. In such cases, it is advisable to reduce the data's heterogeneity and increase the model complexity. However, increasing the model complexity might result in higher computational cost.

Despite being capable of handling non-linear data, MLPs may struggle with heterogeneous data, which often includes diverse types, scales, and distributions, making it challenging for MLPs to manage effectively [Bishop (2006)]. Although MLPs can model complex nonlinear relationships, their architecture, comprising dense layers and fixed activation functions, may struggle to adapt to the varied characteristics of heterogeneous data sets. Additionally, MLPs can be prone to overfitting when dealing with high-dimensional input features common in heterogeneous data sets. Thus, alternative models or ensemble approaches that address the specific challenges posed by heterogeneous data might be more suitable for such applications.

5.1.2 Support Vector Machine for Regression

An initial SVR model was utilized for the preliminary evaluation. The model was trained using a radial basis function (RBF) kernel, which is capable of modeling complex nonlinear relationships. The SVR's performance was assessed using a 5-fold cross-validation approach.

The results are summarized in Table 5.2:

Metric	Train	Test	Disaggregation
Mean Train RMSE	**1.1337**	-	-
Mean CV RMSE	**1.1329**	-	-
Mean Train MAE	**0.6664**	-	-
Mean CV MAE	**0.6668**	-	-
Mean Train R2	**-0.0534**	-	-
Test R2	-	-0.0284	-
Test MSE	-	0.4548	-
Test RMSE	-	0.6744	-
Disaggregation R2	-	-	-0.0060
Disaggregation MSE	-	-	0.6362
Disaggregation RMSE	-	-	0.7976

Table 5.2: SVR preliminary results.

The negative R2 value during training indicates that the model struggled to learn any significant patterns from the data. SVR models have the potential to handle nonlinear data effectively. Nevertheless, their performance may be impacted when dealing with heterogeneous data. In such cases, alternative models or ensemble approaches that address the specific challenges posed by heterogeneous data could be more appropriate for these applications.

5.1.3 XGBoost Regression Model

An initial XGBoost model was employed for the preliminary assessment. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It is particularly effective for handling heterogeneous data due to its decision tree-based structure. The XGBoost model's performance was evaluated using a 5-fold cross-validation approach.

The results are summarized in Table 5.3:

Metric	Train	Test	Disaggregation
Mean Train RMSE	**0.2057**	-	-
Mean CV RMSE	**0.6641**	-	-
Mean Train MAE	**0.1367**	-	-
Mean CV MAE	**0.3656**	-	-
Mean Train R2	**0.9692**	-	-
Test R2	-	0.4255	-
Test MSE	-	0.1194	-
Test RMSE	-	0.3455	-
Disaggregation R2	-	-	0.5968
Disaggregation MSE	-	-	0.5050
Disaggregation RMSE	-	-	0.2690

Table 5.3: XGBoost preliminary results.

The positive R2 value during training demonstrates that the XGBoost model was able to learn meaningful patterns from the data. This model outperformed both the MLP and SVR models in handling the heterogeneous data, as evidenced by the significantly improved R2, MSE, and RMSE values. The decision tree-based structure of XGBoost enables it to effectively adapt to the varied characteristics of heterogeneous data sets, making it a more suitable option for this particular application. The preliminary analysis employed a non-tuned XGBoost model, with parameters selected based on default settings and general recommendations, leading to a high degree of overfitting.

5.2 Results of the Self-Supervised Hybrid Regression Method for Spatial Data Disaggregation - Master Model

Results of a self-supervised hybrid regression method for spatial data disaggregation are demonstrated using Portugal, Germany and Austria as test countries. Analysis centers around answering research questions regarding accuracy in disaggregating data by the model as well as its performance compared with population weighted disaggregation. Results also explore self-supervised approach's ability to retrain prior iterations models with success as well as R2 results calculation across each country; all this helps inform future decisions regarding disaggregation effectiveness vs population-based disaggregation approaches with conclusions being drawn accordingly.

5.2.1 Charging Stations results validation

In this subsection, the validation results of disaggregating the charging stations variable for the Master model is analyzed.

Country	Iteration	Disaggregation Method	RMSE	MAE	R2
Portugal	-	Population-based	8.26	4.37	-16.98
Portugal	1	Model-based	1.24	0.69	**0.59**
Portugal	2	Model-based	1.08	**0.88**	0.28
Portugal	3	Model-based	1.15	0.92	0.18
Portugal	4	Model-based	1.18	0.99	0.14
Portugal	5	Model-based	**1.03**	0.82	0.34
Germany	-	Population-based	0.67	0.33	0.28
Germany	1	Model-based	0.57	0.28	0.49
Germany	2	Model-based	0.58	0.29	0.47
Germany	3	Model-based	0.58	0.29	0.47
Germany	4	Model-based	**0.57**	**0.29**	0.49
Germany	5	Model-based	0.56	0.29	**0.51**
Austria	-	Population-based	2.48	1.32	-3.57
Austria	1	Model-based	0.88	0.62	0.42
Austria	2	Model-based	1.10	0.90	0.10
Austria	3	Model-based	**0.99**	0.79	0.28
Austria	4	Model-based	0.99	0.79	0.28
Austria	5	Model-based	0.91	**0.64**	**0.38**

Table 5.4: Spatial Disaggregation results of Charging Stations variable for Germany, Portugal and Austria.

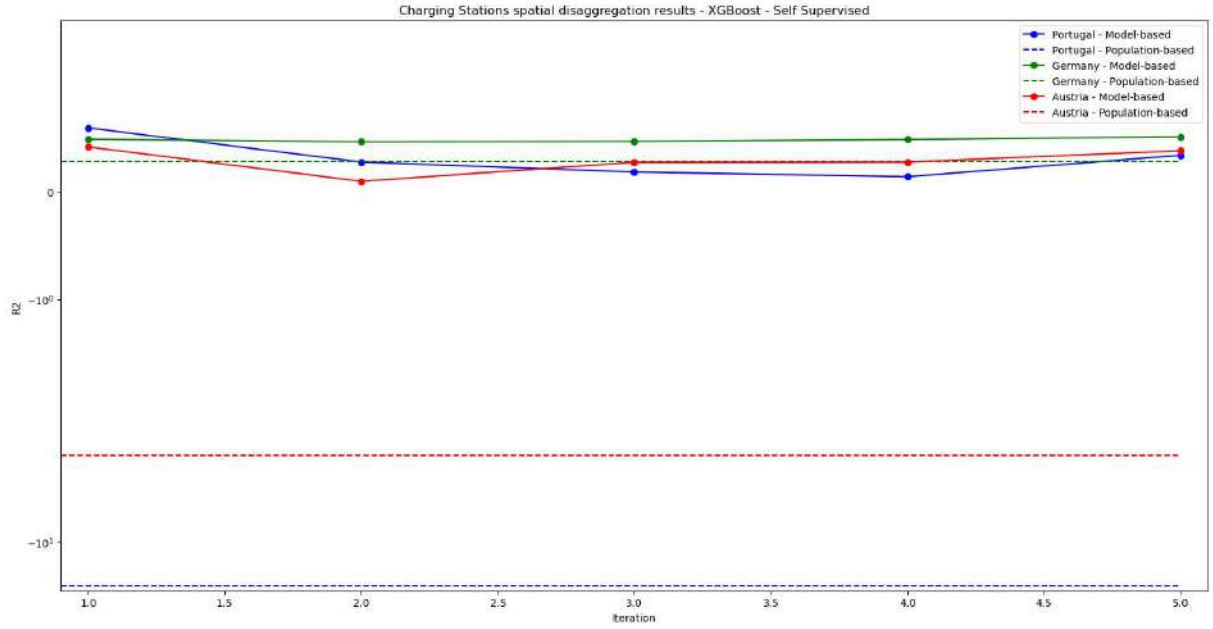


Figure 5.1: R squared result for the Spatial disaggregation of Charging Stations variable. Self - Supervised learning. A population weighted and model-based comparison.

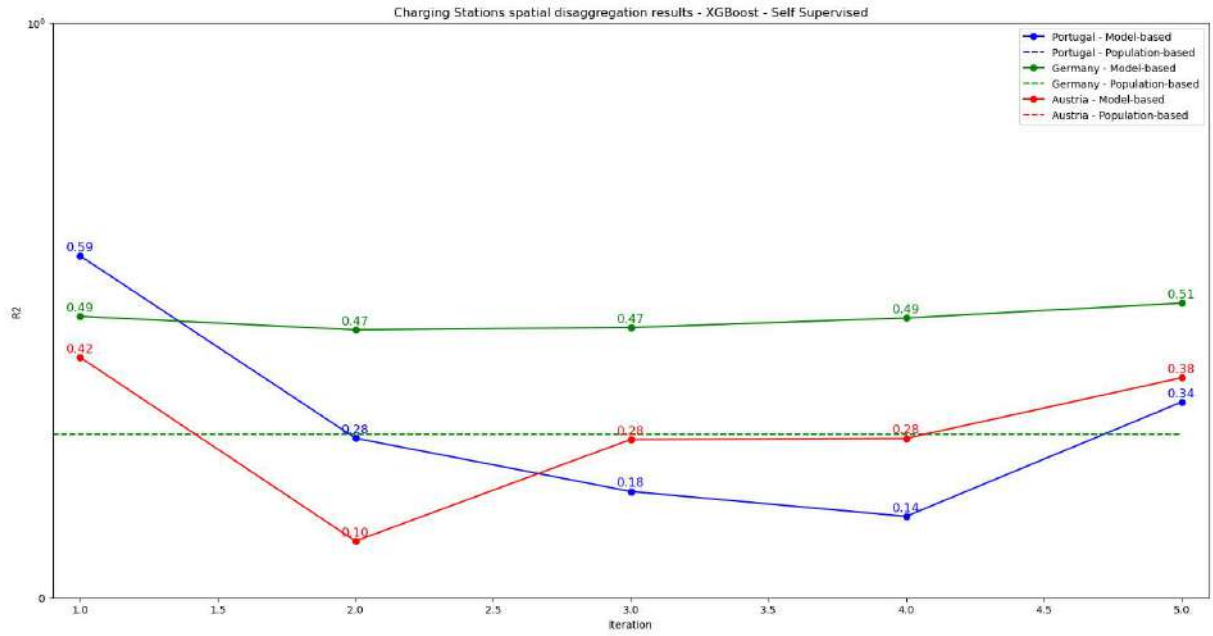


Figure 5.2: R squared result for the Spatial disaggregation of Charging Stations variable. Self - Supervised comparison.

Table 5.4 presents the disaggregation results for the charging stations in Germany, Portugal, and Austria, using the hybrid regression model. The model's performance for each country is evaluated by iteration and disaggregation method, with the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-squared (R2) values as performance indicators.

Figure 5.1 shows the R-squared results for the spatial disaggregation of charging stations using self-supervised learning. The graph provides a comparative analysis between population weighted and model-based approaches.

The disparities between the population weighted and model-based approaches for disaggregating the charging stations data become evident through the R-squared results. For instance, in Portugal, the model-based method shows a substantial improvement in accuracy compared to the population-weighted method. In contrast, the population-weighted approach gives a negative R2 value, indicating a poor fit. Similar observations can be made for Germany and Austria, where the model-based method outperforms the population-weighted method.

Figure 5.2 offers an analysis of whether re-training the model across iterations leads to improved accuracy. Notably, in Portugal, the model's performance decreases after the first iteration, reaching its lowest R2 value at the fourth iteration. However, the fifth iteration shows a rebound in accuracy. In Germany, the model exhibits consistent improvement across iterations. Austria, on the other hand, demonstrates fluctuating R2 values throughout the iterations.

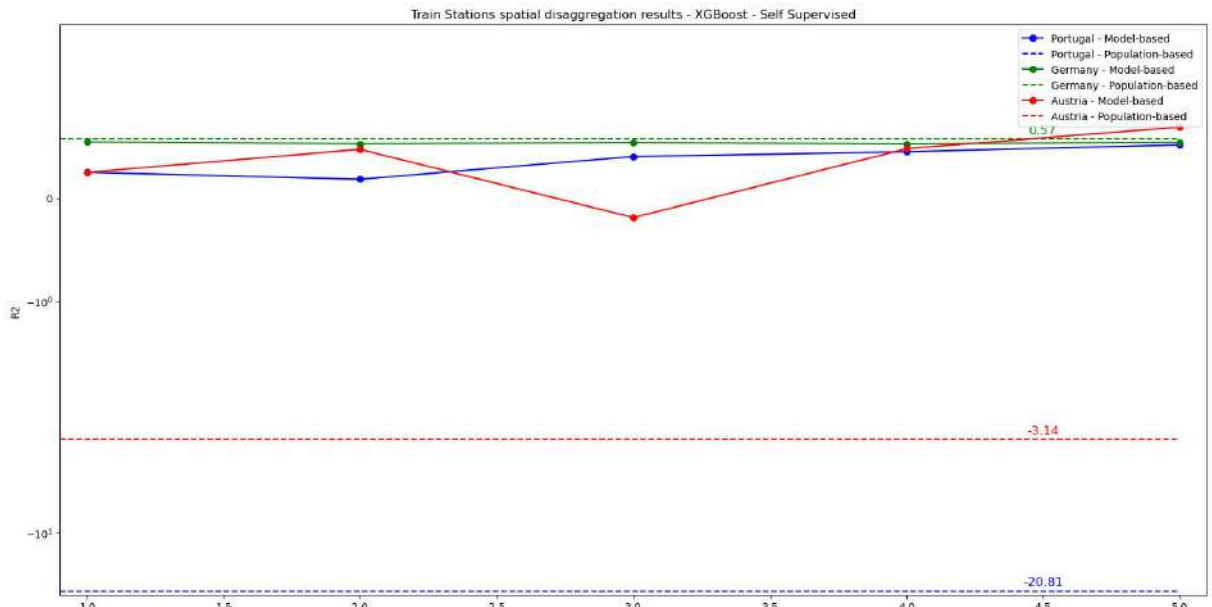
In conclusion, self-supervised hybrid regression proves to be advantageous for model-based disaggregation over population weighted methods. While Germany showed a consistent increase in performance across iterations, Portugal and Austria had less linear improvements, resulting in fluctuating accuracy rates during iteration periods.

5.2.2 Train Stations results validation

In this subsection, the validation results of disaggregating the train stations variable for the Master model are analyzed for the three test countries: Portugal, Germany, and Austria. The results are shown in Table 5.5.

Country	Iteration	Disaggregation Method	RMSE	MAE	R2
Portugal	-	Population-based	3.25	1.72	-20.81
Portugal	1	Model-based	0.60	0.44	0.25
Portugal	2	Model-based	0.63	0.44	0.19
Portugal	3	Model-based	0.54	0.42	0.41
Portugal	4	Model-based	0.52	0.41	0.45
Portugal	5	Model-based	**0.48**	**0.39**	**0.52**
Germany	-	Population-based	**0.41**	**0.27**	**0.58**
Germany	1	Model-based	0.42	0.30	0.55
Germany	2	Model-based	0.43	0.31	0.53
Germany	3	Model-based	0.43	0.29	0.54
Germany	4	Model-based	0.43	0.31	0.53
Germany	5	Model-based	0.43	0.29	0.55
Austria	-	Population-based	2.56	1.31	-3.14
Austria	1	Model-based	1.09	0.79	0.25
Austria	2	Model-based	0.91	0.73	0.48
Austria	3	Model-based	1.37	0.75	-0.19
Austria	4	Model-based	0.91	0.73	0.48
Austria	5	Model-based	**0.70**	**0.59**	**0.69**

Table 5.5: Spatial Disaggregation results of Train Stations variable for Germany, Portugal and Austria.



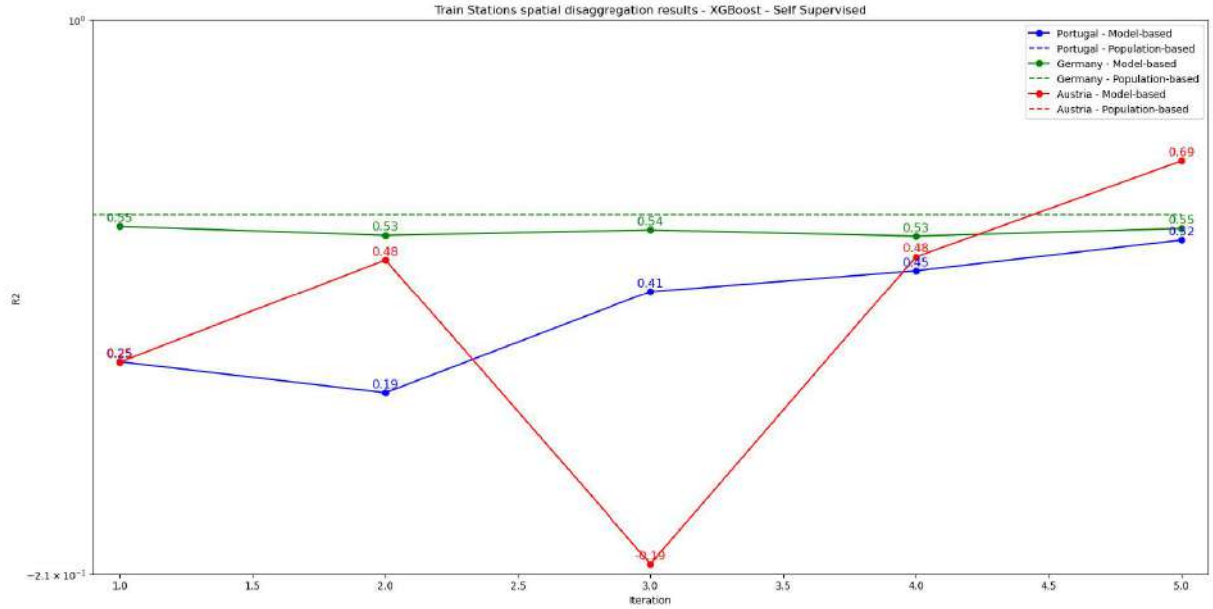


Figure 5.4: R squared result for the Spatial disaggregation of Train Stations variable. Self - Supervised comparison.

The self-supervised hybrid regression method for spatial data disaggregation of train stations shows significant differences between population-weighted and model-based approaches (Figure 5.3). For Portugal, the model-based method outperforms the population-weighted approach, which has a negative R2 value. The model-based R2 values for Portugal range from 0.1851 to 0.5197. In Germany, the model-based method exhibits similar performance to the population-weighted approach, with R2 values between 0.5300 and 0.5494, while the population-weighted method has an R2 value of 0.575. In Austria, the model-based method also demonstrates improvement over the population-weighted approach, which has a negative R2 value. The model-based R2 values for Austria vary from -0.1883 to 0.6924.

Figure 5.4 emphasizes the varying effectiveness of the self-supervised learning approach across iterations. For Portugal, the model's performance exhibits a non-linear progression, achieving the highest R2 value at the fifth iteration. This inconsistency in the R2 values may be attributed to factors such as data quality or differences in the distribution of train stations. Furthermore, the model's parameters and hyperparameters could also impact the performance across iterations. In Germany, the model demonstrates relatively stable performance across iterations, with only minor fluctuations in R2 values. This suggests that the model may be better suited to the specific characteristics of the German dataset. In the case of Austria, the model's performance experiences considerable variation across iterations, culminating in its highest R2 value at the fifth iteration.

In conclusion, the self-supervised hybrid regression method for spatial data disaggregation of train stations highlights the advantages of employing a model-based approach over a population-weighted method for all test countries. The self-supervised learning approach yields diverse results across iterations, with some cases indicating improvements in accuracy, while others

display fluctuations in performance. To ensure consistent improvements in accuracy across different scenarios, further research is necessary to optimize the self-supervised learning process, including refining the model's parameters and hyperparameters and addressing potential data quality issues.

5.3 Results of the Self-Supervised Hybrid Regression Method for Spatial Data Disaggregation - Cluster Model

This subsection presents and analyzes the clustering results for the spatial disaggregation of Germany's charging station variable. Based on the findings in subsection 4.5.6, it was observed that while the Silhouette scores for K-means clustering were relatively smaller in comparison to DBSCAN and Hierarchical clustering, the K-means method demonstrated a more favorable distribution of data. Consequently, the K-means clustering technique has been chosen for further analysis.

Cluster	Iteration	Disaggregation Method	RMSE	MAE	R2
Cluster 1	-	Population-based	**0.4233**	**0.2555**	**0.2589**
	1	Model-based	0.4279	0.2504	0.2429
	2	Model-based	0.4626	0.2764	0.1152
	3	Model-based	0.4720	0.2708	0.0786
	4	Model-based	0.4817	0.2773	0.0406
	5	Model-based	0.4733	0.2707	0.0739
Cluster 2	-	Population-based	**24.250**	**20.233**	**0.3395**
	1	Model-based	26.737	21.076	0.1971
	2	Model-based	28.172	23.656	0.1086
	3	Model-based	28.119	23.667	0.1119
	4	Model-based	27.644	22.180	0.1417
	5	Model-based	28.668	23.815	0.0769

Table 5.6: Spatial Disaggregation results of Charging Stations variable for Germany for each EU cluster.

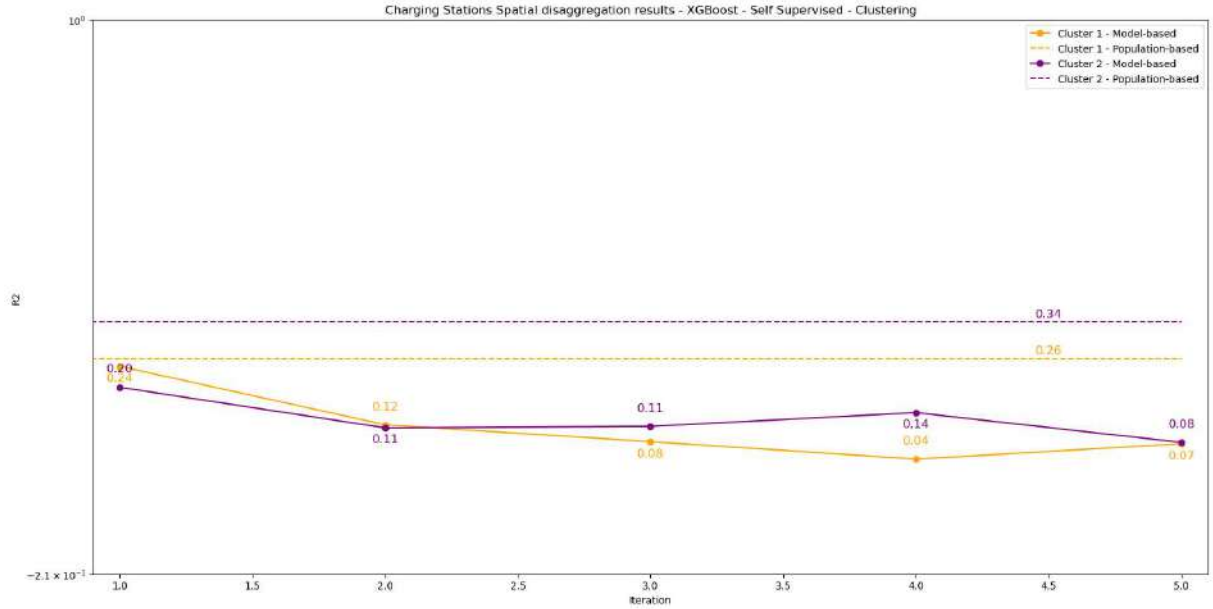


Figure 5.5: Spatial disaggregation R squared result of Charging Stations variable. Self - Supervised learning. A population weighted and model-based comparison.

The spatial data disaggregation of charging stations within each cluster, carried out using the self-supervised hybrid regression method and an XGBoost model, shows significant variance between the population-weighted and model-based approaches. This variation is seen in the R2 values, as evident from the data in Table 5.6. For Cluster 1, the model-based method shows lower performance than the population-weighted approach, which has an R2 value of 0.2589, with the model-based R2 values ranging from 0.0406 to 0.2429. Likewise, in Cluster 2, the model-based method shows poorer performance, with R2 values ranging from 0.0769 to 0.1971, compared to the population-weighted approach's R2 value of 0.3395.

As part of its training process, the model demonstrated robust learning in each cluster. For Cluster 1, the mean train and CV MAE values increased by 10.255% (R2 value of 0.7004), while remaining within reasonable bounds. This suggests that no overfitting occurred due to the small differences between MAE values across both train and CV. In Cluster 2, the mean train MAE was 0.6376 while the CV MAE increased by 23.09%, with an R2 value of 0.7414, suggesting successful generalization. However, the higher MAE values may be due to factors such as increased spatial heterogeneity or regional variations in adoption rates.

No matter how carefully the hyperparameter tuning was carried out, self-supervised learning did not produce consistent effectiveness across iterations due to considerable variation in results from iteration to iteration. This could be due to several reasons including data quality issues, differences in charging station distribution patterns or variations between models' parameters and hyperparameters.

Conclusions of self-supervised hybrid regression method for charging station disaggregation show an inconsistent result when comparing model-based to population-weighted approaches in two EU Union clusters. Self-supervised learning produces variable results across iterations

despite hyperparameter tuning being performed, suggesting additional research needs to optimize this learning process, refine model parameters/hyperparameters and address potential data quality issues; further techniques like including external data sources could lead to more consistent and accurate results.

6 Conclusion and Future Work

6.1 Conclusions

In this thesis, the main goal was to develop and evaluate a self-supervised hybrid regression method for spatial data disaggregation using machine learning models. Three countries, Portugal, Germany, and Austria were studied to see how well the proposed approach compared to population-weighted disaggregation. From the study's results, a few key conclusions can be drawn.

Firstly, it was found that the model-based disaggregation approach consistently performed better than the population-weighted method for both charging and train stations across all test countries. This result emphasizes the importance of using advanced machine learning techniques for spatial data disaggregation tasks, especially when the distribution of the target variable isn't solely dependent on population density.

Secondly, the use of clustering techniques, particularly K-means clustering, was shown to be helpful in dividing the dataset into more homogeneous groups. This step made it easier to understand the spatial distribution of the data and enabled more accurate disaggregation at the cluster level. However, the model-based approach didn't consistently produce better results than the population-weighted method when applied to individual clusters, indicating that more optimization is needed.

Thirdly, mixed results were seen with the self-supervised learning approach across different iterations and test countries. While some cases showed improvements in accuracy, others experienced fluctuations in performance. These inconsistencies could be due to factors like data quality issues, differences in the distribution of the target variable, or changes in the model's parameters and hyperparameters.

Additionally, the differences in the size and distribution of the testing data for each test country had an impact on the R2 results. Portugal, Germany, and Austria made up about 2.14%, 34.27%, and 2.99% of the total testing data, respectively. This unevenness in the testing data might have affected the model's ability to generalize and accurately disaggregate the target variables, particularly in countries with fewer regions, such as Portugal and Austria.

In conclusion, this study demonstrated the potential of a self-supervised hybrid regression method for spatial data disaggregation tasks. The model-based approach was able to deliver more accurate results compared to population-weighted disaggregation methods in most cases. However, the performance of the self-supervised learning approach was inconsistent across different iterations and test countries, showing that more research and optimization are necessary.

6.2 Future work

Based on this study's conclusions, there are a several different paths to take in future research that could help make the self-supervised hybrid regression method even better when it comes to accuracy and sturdiness:

1. It's a good idea to dig into and fix any data quality issues, particularly in cases where the model's performance is a bit shaky across iterations or strays from the population-weighted method.

2. Tuning the model's parameters and hyperparameters could help improve its ability to generalize and achieve more consistent results in various situations.

3. Adding external data sources, like other socio-economic factors or spatial information, might provide extra context and help improve the disaggregation accuracy.

4. It's worth exploring other machine learning algorithms like other ensemble techniques to find models that might be better suited for spatial data disaggregation tasks.

5. May be include more countries or regions in the analysis to see how well the proposed approach scales and works in different settings.

In addition to these suggestions, some other improvements can be considered:

6. Integrate mass-preserving areal weighting to maintain consistency with the aggregated variable.

7. If possible, increase the number of clusters to cut down on heterogeneity and apply a Multi-Layer Perceptron (MLP) or any other suitable model to each cluster, increasing the complexity of the model.

8. Get creative with dimensionality reduction techniques, feature selection, and feature engineering to boost the quality of input data and the model's performance.

9. Investigate alternative clustering algorithms or techniques that could do a better job of capturing the spatial distribution of the data.

10. Evaluate the model's performance with different spatial resolutions, potentially identifying the optimal scale for disaggregation tasks.

By tackling these areas of future work, the self-supervised hybrid regression method can be fine-tuned and optimized, ultimately leading to more accurate and dependable spatial data disaggregation outcomes.

Bibliography

Agency, European Environment (2021): *Greenhouse gas emissions from transport in Europe*.

Bishop, Christopher M. (2006): *Pattern Recognition and Machine Learning*. Springer.

Boeing, Geoff (2017): *OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks*. Computers, Environment and Urban Systems, Bd. 65, S. 126–139.

Breiman, Leo (2001): *Random Forests*. Machine Learning, Bd. 45, S. 5–32.

Caliński, T. und Harabasz, J. (1974): *A dendrite method for cluster analysis*. Communications in Statistics-theory and Methods, Bd. 3, S. 1–27.

Chai, Tianfeng und Draxler, Roland R. (2014): *Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature*. Geoscientific Model Development, Bd. 7, S. 1247–1250.

Contributors, OpenStreetMap (2021a): *Nominatim API*. `https://nominatim.org/`. Accessed: yyyy-mm-dd.

Contributors, OpenStreetMap (2021b): *Overpass API*. `https://wiki.openstreetmap.org/wiki/Overpass_API`. Accessed: yyyy-mm-dd.

contributors, OpenStreetMap (2021): *Planet dump*.

Cortes, Corinna und Vapnik, Vladimir (1995): *Support-vector networks*. Machine learning, Bd. 20, S. 273–297.

Cover, Thomas M. und Thomas, Joy A. (2006): *Elements of Information Theory*. Wiley-Interscience, 2. Aufl.

Draper, Norman und Smith, Harry (1998): *Applied Regression Analysis*. Wiley, 3. Aufl.

Drucker, Harris, Burges, Christopher J.C., Kaufman, Linda, Smola, Alex J. und Vapnik, Vladimir (1997): *Support Vector Regression Machines*. In: *Advances in Neural Information Processing Systems*, Bd. 9, S. 155–161.

Eicher, C. L. und Brewer, C. A. (2001a): *Dasymetric mapping and areal interpolation: Implementation and evaluation*. Cartography and Geographic Information Science, Bd. 28, S. 125–138.

Eicher, Carla L und Brewer, Cynthia A (2001b): *Dasymetric mapping and areal interpolation: implementation and evaluation*. Cartography and Geographic Information Science, Bd. 28, S. 125–138.

Ester, M., Kriegel, H. P., Sander, J. und Xu, X. (1996): *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, S. 226–231.

Eurostat (2021): *Greenhouse gas emissions by aggregated sector*.

Fisher, Peter F und Langford, Mitchell (1997): *Improving the precision of areal interpolation*. Geographical Analysis, Bd. 29, S. 217–230.

Flötteröd, Gunnar und Lückenkötter, Johannes (2018): *A synthetic European road freight transport flow dataset*. Data in Brief, Bd. 21, S. 1810–1814.

Ghosh, Sankhana und Fung, Brian C. M. (2020): *Spatial disaggregation of complex disaster and climate risk indicators: A temporal consistency comparison between methods*. International Journal of Digital Earth, Bd. 13, S. 1117–1134.

Glorot, Xavier, Bordes, Antoine und Bengio, Yoshua (2011): *Deep sparse rectifier neural networks*. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, S. 315–323.

Goodchild, Michael F. und Lam, Nina Siu-Ngan (1980): *Areal interpolation: A variant of the traditional spatial problem*. Geo-Processing, Bd. 1, S. 297–312.

Goodfellow, Ian, Bengio, Yoshua und Courville, Aaron (2016): *Deep Learning*. MIT Press.

Guyon, Isabelle und Elisseeff, André (2003): *An Introduction to Variable and Feature Selection*. Journal of Machine Learning Research, Bd. 3, S. 1157–1182.

Hastie, Trevor, Tibshirani, Robert und Friedman, Jerome (2009): *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2. Aufl.

Hoerl, Arthur E. und Kennard, Robert W. (1970): *Ridge Regression: Biased Estimation for Nonorthogonal Problems*. Technometrics, Bd. 12, S. 55–67.

Jain, A. K. und Dubes, R. C. (1988): *Algorithms for Clustering Data*. Prentice-Hall, Inc.

Jolliffe, Ian (2002): *Principal component analysis*. Springer Series in Statistics. ISBN 0-387-95442-2.

Kaufman, Leonard und Rousseeuw, Peter J (2009): *Finding groups in data: an introduction to cluster analysis*, Bd. 344. John Wiley and Sons.

Kraftfahrt-Bundesamt (2023): *Federal Motor Transport*.

LeCun, Yann, Bengio, Yoshua und Hinton, Geoffrey (2015): *Deep learning*. Nature, Bd. 521, S. 436–444.

MacQueen, J. (1967): *Some Methods for Classification and Analysis of Multivariate Observations*. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Bd. 1, S. 281–297. University of California Press.

McInnes, Leland, Healy, John und Melville, James (2020): *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. ArXiv, Bd. abs/1802.03426.

Mennis, J. und Hultgren, T. (2006): *Intelligent Dasymetric Mapping and Its Application to Areal Interpolation*. Cartography and Geographic Information Science, Bd. 33, S. 179–194.

Monteiro, J., Martins, B., Murrieta-Flores, P. und Pires, J. M. (2019): *Spatial Disaggregation of Historical Census Data Leveraging Multiple Sources of Ancillary Information*. Journal of Historical Geography.

Monteiro, J. M., Martins, B., Costa, M. und Pires, J. M. (2020): *A Co-Training Approach for Spatial Data Disaggregation*. In: *Proceedings of the 28th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, S. 481–490. Universidade de Lisboa.

Openshaw, Stan (1984): *The modifiable areal unit problem*. Concepts and techniques in modern geography, Bd. 38, S. 60–69.

Pedregosa, Fabian, Varoquaux, Gaël, Gramfort, Alexandre, Michel, Vincent, Thirion, Bertrand, Grisel, Olivier et al. (2011): *Scikit-learn: Machine learning in Python*. Journal of machine learning research, Bd. 12, S. 2825–2830.

Polikar, Robi (2006): *Ensemble based systems in decision making*. IEEE Circuits and Systems Magazine, Bd. 6, S. 21–45.

Rodgers, Joseph Lee und Nicewander, W Alan (1988): *Thirteen ways to look at the correlation coefficient*. The American Statistician, Bd. 42, S. 59–66.

Rosenblatt, Frank (1958): *The perceptron: A probabilistic model for information storage and organization in the brain*. Psychological Review, Bd. 65, S. 386–408.

Rousseeuw, P. J. (1987): *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis*. Journal of Computational and Applied Mathematics, Bd. 20, S. 53–65.

TagFinder (2021): *OpenStreetMap Data in Layered GIS Format Documentation*.

Tobler, Waldo R (1979): *Smooth pycnophylactic interpolation for geographical regions*. Journal of the American Statistical Association, Bd. 74, S. 519–530.

Appendix

A.1

Method	Selected Features
Number of Features	bus_stations_value, fuel_stations_value
	number of pixels with industrial or commercial units_value
	railway_station_value, UMAP8
Highly Correlated Features	railway_station_value, bus_stations_value
	number of pixels with industrial or commercial units_value
	UMAP8, fuel_stations_value
F-regression	railway_station_value, bus_stations_value
	number of pixels with industrial or commercial units_value
	UMAP8, fuel_stations_value
Mutual Information	bus_stations_value, railway_station_value
	gross domestic product_value, gross value added_value
	UMAP8, UMAP19, bicycle_stations_value
	number of pixels with industrial or commercial units_value,
	helicopter_station_value, UMAP3
Ridge Regression (L2)	population_value, railway_station_value
	bus_stations_value, UMAP14
	total number of businesses_value, UMAP7
	total employment_value, UMAP12
Random Forest Importance	railway_station_value, UMAP8, bus_stations_value
	gross value added_value, train_station_value
	gross domestic product_value
	number of pixels with mineral extraction sites_value
Combined Features	total employment_value, train_station_value
	UMAP8, bus_stations_value, UMAP14
	helicopter_station_value, bicycle_stations_value
	gross value added_value, fuel_stations_value
	UMAP3, UMAP12, gross domestic product_value
	number of pixels with mineral extraction sites_value
	number of pixels with industrial or commercial units_value
	UMAP19, total number of businesses_value
	railway_station_value, population_value, UMAP7

Table A.1: Selected Features.

A.2

The content located in the appendix corresponds to the "Methodology" chapter (chapter 4), more specifically to the "Clustering" section (section 4.5). This includes detailed elaboration on the subsubsection "Optimal Density-Based Clustering with DBSCAN" (subsection 4.5.4) as well as "Optimal Partitioning-Based Clustering with K-Means" (subsection 4.5.5). These sections provide an in-depth exploration of both DBSCAN and K-Means clustering and their respective optimization processes.

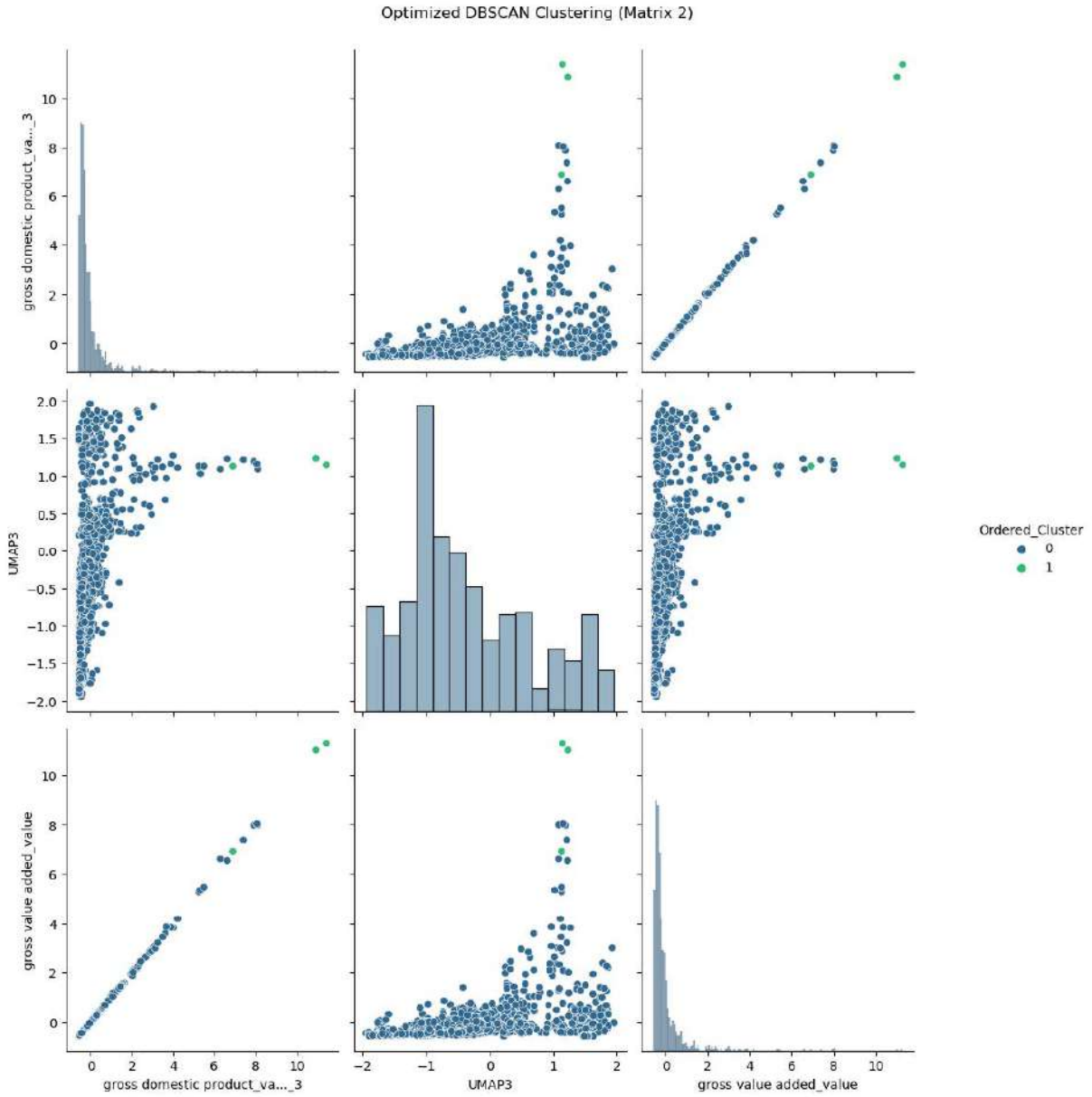


Figure A.1: DBSCAN Pairwise Feature Scatterplot Matrices.

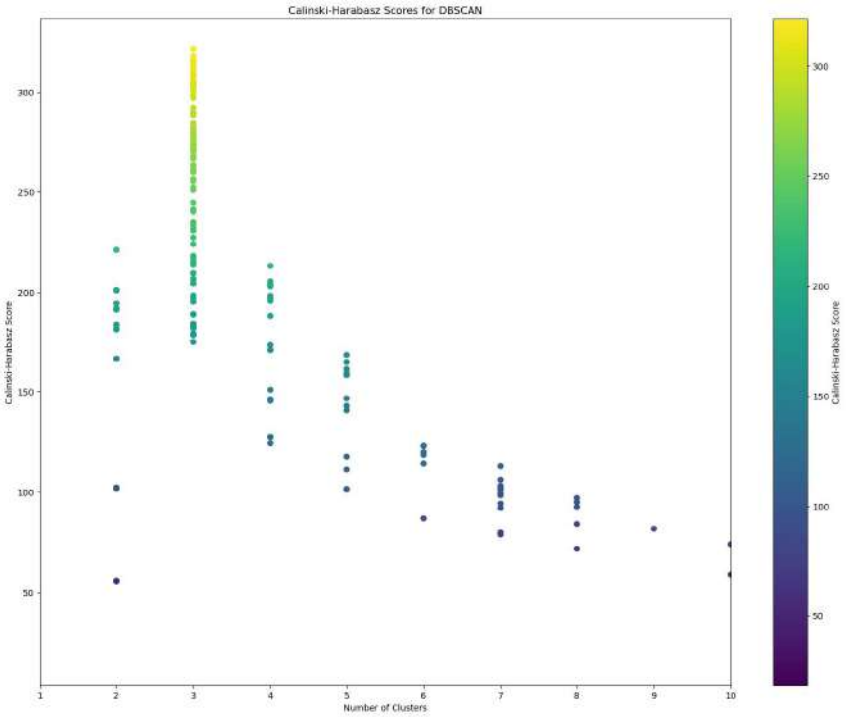


Figure A.2: DBSCAN Calinski-Harabasz Index Plot.

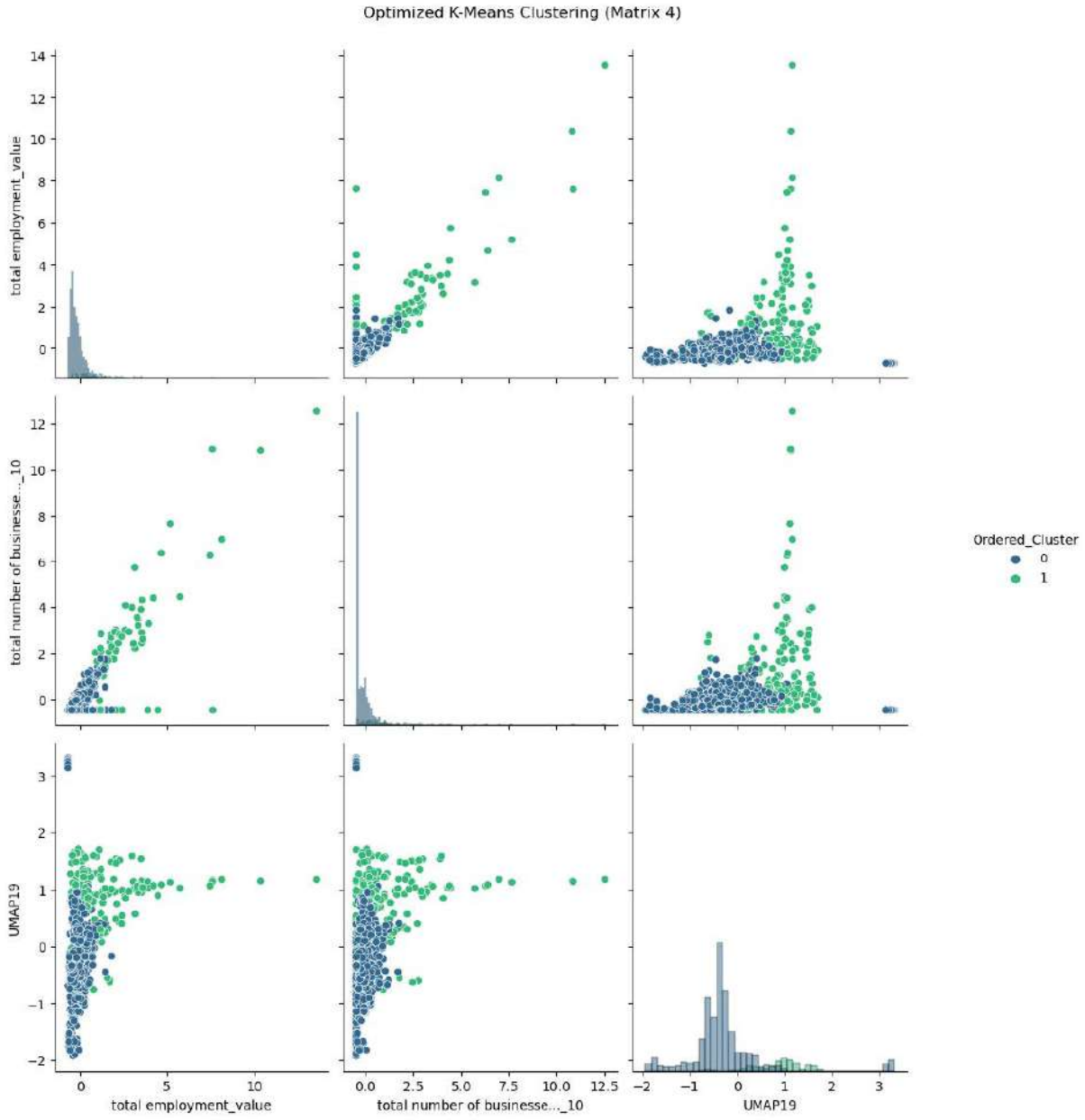


Figure A.3: K-Means Pairwise Feature Scatterplot Matrices.

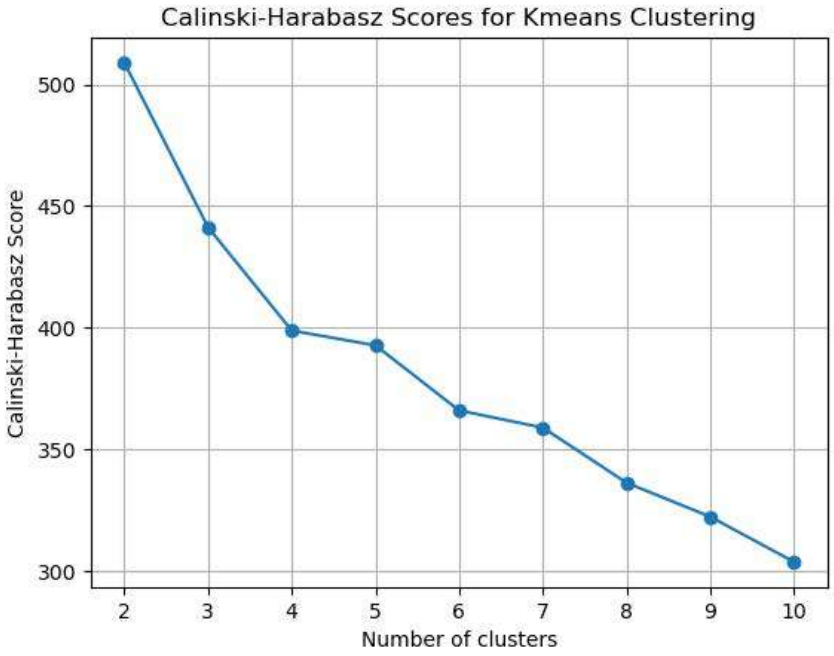


Figure A.4: K-Means Calinski-Harabasz Index Plot.