UNIVERSITY OF TWENTE.

Applications of eXplainable Artificial Intelligence in Public Employment Services Decision Support Systems

by

Julius Kooistra

A thesis submitted to the
Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
in partial fulfilment of the requirements for the degree of

MSc in Business Information Technology

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)

University of Twente

Enschede, Overijssel, The Netherlands

September 2024

ABSTRACT

The study explores the integration of eXplainable Artificial Intelligence (XAI) systems into the Decision Support Systems (DSS) of Public Employment Services (PES) to enhance the allocation of resources for reducing unemployment. The research focuses on profiling and clustering unemployed individuals in Switzerland using both supervised and unsupervised Machine Learning (ML) models. Traditional models like Logistic Regression and Decision Trees have been commonly used in PES but are often limited by their simplicity and limited interpretability. This study introduces more advanced models, such as XGBoost and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), to improve accuracy and clustering efficiency. The integration of XAI into these models ensures that predictions and decisions are interpretable, allowing caseworkers to make informed decisions. The XAI integration consists of three main components: a preprocessing pipeline, a clustering module, and a predicting module, which outputs explanations by design. The findings indicate that the XAI-enhanced models can provide actionable insights at the individual, cluster, and global levels, improving the efficiency of resource allocation and reducing long-term unemployment. Despite some limitations, including data quality and model performance, the study contributes to the literature by bridging the gap between AI in PES and XAI in DSS, and by implementing innovative clustering techniques within PES.

Keywords: eXplainable Artificial Intelligence (XAI); Decision Support System (DSS); Public Employment Services (PES); Machine Learning (ML); Unemployment Profiling, Clustering; Interpretability.

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Twente to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Twente to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Julius Kooistra

ACKNOWLEDGEMENTS

I would like to extend my deepest gratitude to those who have supported and guided me throughout this academic, professional, and personal journey.

First and foremost, I wish to express my sincere thanks to my supervisors Marcos Machado, João Rebelo Moreira, and Branka Hadji Misheva for steering me in the right direction and enriching my knowledge while allowing me to explore and have fun doing so. I would like to particularly thank Marcos for his helpful, comprehensive, and timely feedback, chasing me to pursue an academic publication, and always lending a sympathetic ear. My extensive gratitude also goes towards Branka, who allowed me to work on this interesting and challenging project with social importance, convinced me to work on the project from Switzerland, and supported me in becoming an expatriate working abroad for a foreign organization. I also want to thank Stephan Winzeler from the AVA Bern for providing us with this project, introducing me to the organization and its people and procedures, and always being available for any (domain-related) questions or having a good time.

I would also like to take this opportunity to thank my colleagues in the BFH team. Even though we were not working together directly on this project, our conversations have made me think about this research on a deeper level. It has been great to work in such a knowledgeable yet diverse team and your daily support has levitated my motivation to also work on my thesis instead of focusing on the practical implementation.

Last but not least, my heartfelt appreciation goes to my family for consistently encouraging me to push on despite the large physical distance, advising me on topics in the widest of spectra, and supporting my adventure abroad from the first ideas to the eventual extension. Without you, I wouldn't be where I am today, for which I'm eternally grateful. I would also like to thank my friends, both in the Netherlands and in Bern, for providing me with some moments of relaxation and diversion.

Thank you all for your personal contributions to and guidance during the past half-year in which I accomplished this significant milestone: writing this thesis and, with that, completing another step in my educational and personal journey. I hope you enjoy the reading!

CONTENTS

List of Figures

List of Tables

ABBREVIATIONS

AI	Artificial Intelligence.
AMM	Labor Market Measure.
AUC ROC	Area Under the Receiver Operating Characteristic Curve.
AVA	Office of Unemployment Insurance.
CART	Classification and Regression Trees.
CHI	Calinski–Harabasz Index.
CRISP-ML	CRoss Industry Standard Process for Machine Learning.
DBI	Davies–Bouldin Index.
DBSCAN	Density-based Spatial Clustering of Applications with Noise.
DSS	Decision Support System.
DT	Decision Tree.
FPR	False Positive Rate.
GBDT	Gradient Boosting Decision Trees.
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise.
LEAF	Local Explanation evAluation Framework.
LIME	Local Interpretable Model-agnostic Explanations.
LR	Logistic Regression.
LTU	Long-Term Unemployed.
MDS	Multidimensional Scaling.
ML	Machine Learning.
MSC	Mean Silhouette Score.
MST	Minimum Spanning Tree.
NLI	Necessary Level of Interpretability.
NN	Neural Network.
PES	Public Employment Services.
RAV	Regional Employment Centers.
RF	Random Forest.
ROC	Receiver Operating Characteristic.
SECO	State Secretariat for Economic Affairs.
SHAP	SHapley Additive exPlaination.
STES	Job Seekers.
TNR	True Negative Rate.
TPR	True Positive Rate.
TS	Target Statistics.

XAI eXplainable Artificial Intelligence.

1

INTRODUCTION

1.1. MOTIVATION

The global unemployment rate has been steadily declining since its peak at 7.0% after the 2008 global financial crisis and its peak of 6.6% after 2021's global pandemic. According to the World Bank, the global unemployment rate is 5.0% of the working population in 2023 [1]. This means that worldwide, around 181.5 million persons are unemployed. The World Bank estimates the unemployment rate in the European Union to be 6.0% in 2023 [1], meaning there are around 13 million unemployed in just the European Union. The unemployment rate in Switzerland, the country in which this research is conducted, is estimated at 4.0% [1], meaning that in the second quarter of 2024, just over 200.000 Swiss are unemployed [2].

On an individual level, unemployment can cause resource deprivation, social isolation, and reduced satisfaction and psychological well-being [3]. In addition, unemployment affects the direct environment, being the spouses, children and partners of an individual, but also has high societal costs in the form of benefits payment, loss of the production of workers and ultimately a reduction of the Gross Domestic Product (GDP) of a country [4]. Many Western nations have introduced the concept of a Public Employment Services (PES) to reduce unemployment rates, boosting economic growth and public well-being.

In today's Western labour markets, data is increasingly being used by PES to help deliver more efficient employment services to prevent long-term unemployment. This is done by, for instance, increasing the efficiency of its counselling process and the effectiveness of active labour market programs [5]. Statistical profiling tools have been found to ensure that costly and intensive resource allocation is targeted towards the groups most in need, allowing for tailored services more closely to their individual needs [6]. Statistical profiling of individuals is also employed in criminology [7], banking [8], and health care [9].

Profiling can have two inherent meanings [10, 11]:

1. the recording and analysis of a person's psychological and behavioural characteristics, so as to assess or predict their capabilities in a certain sphere. In this thesis, this is referred to as profiling.

2. the recording and analysis of a person's psychological and behavioural characteristics, to assist in identifying categories of people. In this thesis, this is referred to as clustering.

Only a few examples are known where the statistical profiling tool used to profile job seekers is an instance of Artificial Intelligence (AI). This research is done in Austria, Denmark, Flanders (Belgium), and Estonia [6, 12]. All examples employ simple, inherently interpretable Machine Learning (ML) models like Logistic Regression (LR) or Decision Tree (DT)s to profile job seekers. Currently, these simple statistical profiling tools help PES to offer more tailored services to individual job seekers based on the profiling and reduce the average cost of an unemployed, also referred to as optimizing their cost-efficiency. However, the accuracy of the simple ML performance can be insufficient.

Recently, more advanced ML models such as XGBoost have shown an improved accuracy over the simpler models in many cases, showcasing the potential success of more advanced ML models [13]. These models have one major disadvantage for profiling: having a black-box architecture, they are not inherently interpretable, requiring a separate explainer to allow for understanding why certain customers are profiled accordingly [14]. To our best knowledge, only the recently published work by Dossche et al. [15] shows the application of eXplainable Artificial Intelligence (XAI) in the Decision Support System (DSS) of a PES. Where the work of Dossche et al. focuses on the implementation and comparison of explainability methods in an existing and currently used profiling model, our work focuses on developing a new XAI system for an existing DSS.

The clustering of job seekers based on demographic and sociological factors is done in only a few known examples. For example, in Flanders youthly job seekers get a different treatment than the rest of the customers via a rule-based method [6]. To the best of our knowledge, there are no examples in PES literature cluster job seekers with respect to their mathematical similarity. This is what generally happens in unsupervised ML models, for example, in the Nearest Neighbour algorithm. The disadvantage of such methods is that they are expensive to compute in higher dimensions, produce inexplicit spherical groupings, and rely on mathematical distances in a multidimensional spectrum [16, p. 127].

The recently introduced ML models like Density-based Spatial Clustering of Applications with Noise (DBSCAN) and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) reduce the limitations of classical clustering methods like the Nearest Neighbours algorithm, allowing for non-spherical clusters and faster computation in higher dimensions [17, 18]. The analysis of the calculated clusters and their implications on policy level has to be performed manually. To avoid creating individual measures for people with very similar needs, assigning job seekers into clusters allows for the customization of employment measures to a

group level instead of an individual level, increasing the generalization and standardization of the measures.

1.2. MAIN CONCEPTS

This research considers Artificial Intelligence (AI) to adhere to the definition presented by Rai et al. [19, p.iii]: *"AI is [. . .] the ability of a machine to perform cognitive functions that we associate with human minds, such as perceiving, reasoning, learning, interacting with the environment, problem-solving, decision-making, and even demonstrating creativity"*.

eXplainable Artificial Intelligence (XAI) is most briefly defined as *"making AI understandable by people"*, however, as the focus of this research lies in the technological area, the definition is tightened to *"all technical means to making AI understandable by people, including direct interpretability, generating an explanation or justification, providing transparency information, etc."* [20, p.ii]. XAI is also sometimes called *interpretable AI*, of which the definition by Alangari et al. [21, p.iii] is used: *"reflect the capability to convey the trained model's output behaviours in a human understandable way"*. This research uses XAI to refer to technical contexts for the rest of this research, whereas interpretable AI is used for non-technical contexts.

In the multidisciplinary context of this study, the definition of an *explanation* slightly differs from what is generally considered to be an explanation: *"an interaction between two parties, the explainer and the explainee, with the goal of aligning mental models"* [22, p.i].

This research periodically uses the term Machine Learning (ML), defined as *"a type of artificial intelligence that allows machines to learn from data without being explicitly programmed. It does this by optimizing model parameters (i.e. internal variables) through calculations, such that the model's behaviour reflects the data or experience."* [23]. This study uses the broadest definition of a Decision Support System (DSS): *"an interactive computer-based system that helps people use computer communications, data, documents, knowledge, and models to solve problems and make decisions"* [24, p.xii].

Profiling is defined as "a systematic (qualitative and/or quantitative) assessment of the individual employment potential to identify and implement the most appropriate services that help the client through the whole integration chain" [25].

1.3. RESEARCH PRACTICAL CONTEXT

Office of Unemployment Insurance (AVA) Bern is the Office of Unemployment Insurance of the Swiss canton Bern and actively supports unemployed citizens, also referred to as customers, in their job search and with financial benefits. The Public Unemployment Insurance Fund covers the financial damage caused by unemployment with insurance benefits. The Regional Employment Centers (RAV) helps customers look for work with regular discussions and a wide range of offers to enable them to rapidly integrate permanently into the labour market. The AVA has many partnership-based collaborations with Bernese companies which contribute to the suc-

cess of finding jobs for their customers quickly. AVA Bern consists of a management layer, the Public Unemployment Insurance Fund Bern, and the ten RAVs [26] and reports directly to the State Secretariat for Economic Affairs (SECO) of the federal government.

The digital infrastructure of the AVA Bern is hosted by the SECO and is generalized throughout the 26 cantons of Switzerland. The main system in which all transactions and customer data are stored is called AVAM, a legacy system written in Cobol from the 1970s. Caseworkers must manually add all customer and transaction information to AVAM via a terminal window. Data is extracted from AVAM once a day, during nighttime, and stored in an external database system connected to MicroStrategy[1].

In MicroStrategy, dashboards can be made to inspect this data throughout the organisation, and RAV counsellors are provided with such a dashboard via their cantonal AVA. This allows every canton to have their own way of working, which is preferred in Switzerland due to strong regional differences in the labour markets. Currently, all data analysis is done via MicroStrategy reports. There is no known usage of AI profiling within the AVAs of Switzerland.

1.4. RESEARCH MOTIVATIONS AND OBJECTIVES

Over the last few years, AVA Bern has invested heavily in their data quality. With the improved data quality, the AVA Bern believes that it's ready to implement the first AI based DSS in the Swiss AVAs that improves their efficiency by supporting their counsellor's decision-making process and allowing them to assist a customer based on their socio-economic characteristics proactively. The objective is to reduce the average unemployment duration while reducing the required (human) resources required for a swift re-entry to the labour market for their customers. A swift re-entry to the labour market benefits both the unemployed and the AVA as it gives the unemployed confidence and self-satisfaction [3] and saves the AVA money in the form of payments they won't have to make. As the AVA is publicly funded, preventing long-term unemployment ultimately saves the Swiss population money. Every weekday, almost one million Swiss Francs flow through the AVA Bern in the form of paid benefits, showing the potential savings of this project. If the project reduces the average duration of unemployment by only 1%, the Canton of Bern saves 10.000 Swiss Francs per working day, which is more than 2.5 million Swiss Francs per year.

The primary goal of this study is to create an XAI tool that is integrated into an existing DSS using both supervised and unsupervised ML algorithms to profile and cluster unemployed people who might have difficulties finding a new job at the moment of registration at the RAV and create strategies to avoid this behaviour and ensure a rapid reintegration in society. Based on the databases in the AVAM system, appropriate features are selected for both the profiling and the clustering. This should be done interpretably to allow the counsellor to understand why the system returns a certain prediction.

[1] https://www.microstrategy.com/

1.5. RESEARCH QUESTIONS

Based on the research objectives mentioned in the previous section, we aim to provide answers to the following main research question (RQ):

> How to design and integrate an XAI system in an existing DSS to improve the allocation of resources in a PES?

From the main research question, several sub-questions have been drafted:

1. How to design an AI system for a PES?
   (a) What different AI models are available?
   (b) How to validate these AI systems?
2. How to integrate an XAI system into a DSS?
   (a) What different XAI technologies are available?
   (b) How can XAI technologies be used to understand predictions?
3. How to extract new knowledge from existing PES datasets using XAI technologies?
   (a) What patterns can be detected using profiling?
   (b) What patterns can be detected using clustering?

1.6. METHODS

This research uses the CRoss Industry Standard Process for Machine Learning (CRISP-ML) framework to ensure the Necessary Level of Interpretability (NLI) across the stages of the research. It applies Hierarchical clustering, DBSCAN and HDBSCAN unsupervised and the DT, Random Forest (RF), XGBoost, and Catboost supervised ML to a PES dataset. All models are validated using cross-validation. The unsupervised models are validated through the Calinski-Harabasz Index, Davies-Bouldin Index, and Silhouette Score. The supervised models are validated through the Area Under the Receiver Operating Characteristic Curve (AUC ROC), accuracy, F1-score, and Brier score. SHapley Additive exPlaination (SHAP) is employed as the explainability method of choice.

1.7. SUMMARY

The remaining sections are organized as follows. In chapter 2, we explore current AI applications in PES, finding the simple ML models are generally used and identified benchmarks for precision, accuracy, and AUC ROC scores. We highlight the required complexity of an AI profiling system to address the diverse challenges faced by job seekers, especially those with psychological issues or other barriers not typically recorded in PES data. Additionally, we explore how XAI can enhance DSS by making AI predictions understandable. We compare model-specific and model-agnostic explainability approaches and find that SHAP and Local Inter-

pretable Model-agnostic Explanations (LIME) are the most prevalent technologies. We emphasized the importance of model-agnostic explainers that can operate globally and locally, offering flexibility and clarity. mchapter 3, discusses the methodology used in this research and presents the CRISP-ML framework and several statistical methods implemented in this study. Next, chapter 4 presents an overview of the experimental set-up designed to develop state-of-the-art unsupervised and supervised ML models, of which the results obtained are presented in chapter 5, where we identify reproducible clusters of job seekers with different characteristics and needs. We apply supervised-learning methods and use SHAP values to interpret the produced models at global, cluster, and individual levels, revealing key profiling features like region, registration type, age, and language. Finally, the conclusions, limitations and future recommendations are presented in chapter 6, where we discuss how these insights could lead to better-targeted employment measures.

2

LITERATURE REVIEW

This chapter connects the two previously disconnected research areas of Artificial Intelligence (AI) in Public Employment Services (PES) and eXplainable Artificial Intelligence (XAI) in Decision Support System (DSS). We perform two structured literature reviews and attempt to integrate the knowledge to create a starting point for research on XAI in DSS in PES.

2.1. METHODOLOGY

This literature research employs the Scopus[1] abstract and citation database as the source to subtract high-quality scientific articles, mainly from peer-reviewed journals, selected according to a content coverage policy. Additionally, Scopus offers a wide range of features that can filter the body of literature to identify relevant literature. This research follows a structured approach in documenting the selection of literature comparable to Wijnhoven and Machado [27] and Amato et al. [28].

We executed the retrieval of documents using the advanced search bar, which allows researchers to insert several keywords and create custom search queries. As there exist multiple synonyms and abbreviations for these concepts, we have included these in the search query to ensure the concepts are well represented, resulting in the following query:

> *('Decision support systems' OR 'Decision support tools' OR 'DSS') OR ('Explainable artificial intelligence' OR 'explainable AI' OR 'XAI') AND ('Unemployment Insurance' OR 'Public Employment Service')*[2]

As this query produced only a several results, we aimed to investigate the application of AI in PES. Removing the explainability part, and thus replacing XAI with AI, from the query produced 23 results. In an attempt to create a nexus between the two, at the time of conceptualizing this

[1] https://www.scopus.com
[2] This query returns eight results

research, unconnected research areas of AI in PES and XAI applications in DSSs, we formulated a third and fourth query that separate the domains:

> *('Artificial Intelligence' OR 'AI') AND ('Unemployment Insurance' OR 'Public Employment Service')*[3]
>
> *('Decision support systems' OR 'Decision support tools' OR 'DSS') AND ('Explainable artificial intelligence' OR 'explainable AI' OR 'XAI')*[4]

2.1.1. INCLUSION AND EXCLUSION CRITERIA

Following Wijnhoven and Machado [27] and Amato et al. [28], inclusion and exclusion criteria have been set before the literature analysis to ensure a systematical, phase-based approach to identifying relevant and high-quality literature. Figure 2.1 visualizes the inclusion criteria and their results on the remaining body of literature.

As seen in Figure 2.1, the first six phases of the selection process reduced the relevant literature for this review. The final phase selects literature based on a thorough examination of the abstract of the research. From the AI in PES branch of the literature review, all literature was selected and taken as input for the analysis. From the branch of XAI in DSS much literature has been removed according the following reasoning. Duplicated literature and papers without access are removed from the selection, eliminating two documents from the body of literature. All literature in which the DSS is employed in a clinical context is removed, as the deeper goal of this literature review is to understand how XAI can be applied in a DSS in PES. The field of PES is highly dependent on data of a tabular nature, while clinical applications generally rely on image data [29], limiting the transferability of results between these fields, eliminating 51 papers from the remaining body of literature. Finally, from the title, keywords, and abstract, we judged the transferability of the literature to the field of PES, eliminating another 38 documents from the body of literature. Phase seven of this literature review results in 42 papers used as input for the analysis.

2.1.2. SNOWBALLING METHOD

As the AI in PES branch of the literature review resulted in four results, the authors decided to apply the snowballing method. In Berman et al. [30], a reference to an international comparison of statistical profiling in PES [6] was found. Further snowballing based on the abstracts of the literature referenced in that work resulted in 15 articles identified as of interest for this study and are thus reviewed in this chapter. One work, [12], was also found with the query from section 2.1 and is not reviewed again. In addition, a brand-new article about explanations for Machine Learning (ML)-based profiling in PES was found by examining the papers that referred to Desiere et al. [6] and was also added to the body of literature of this research. In total, this work reviews 57 articles.

[3] This query returns 20 results
[4] This query returns 406 results

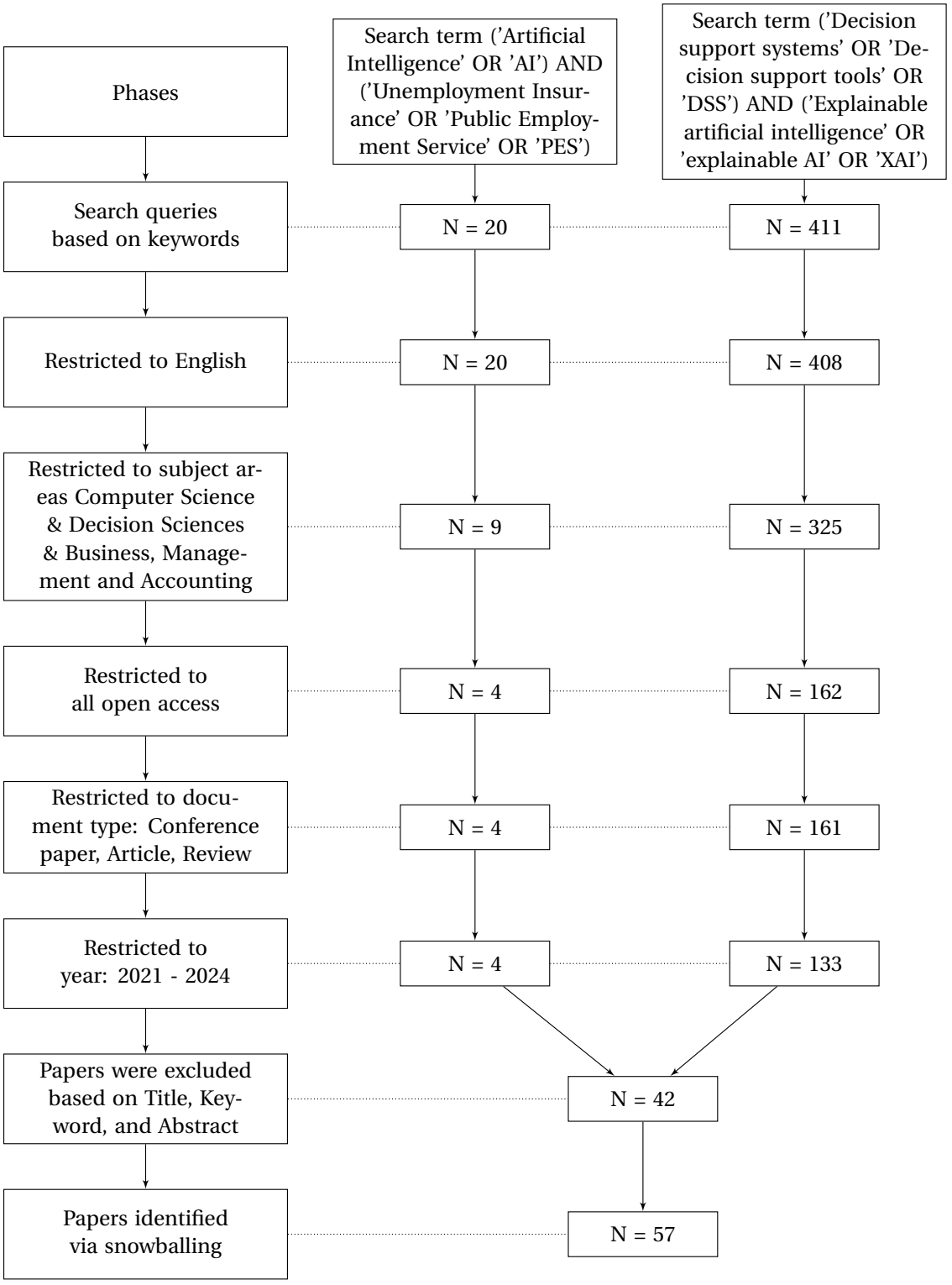


Figure 2.1: Stages of the study selection process for the literature review.

2.2. FINDINGS

The study selection process resulted in 57 articles which are categorized as either an article in the AI and PES domain or the XAI and DSS domain, respectively presented in Table 2.1 and

Table 2.2. The number of citations and Field-Weighted Citation Impact (FWCI) were scraped from the Scopus[5] database. Table 2.1 also contains literature gathered through a snowballing procedure and thus includes grey literature not published in a journal. In this case, the journal column contains the phrase '-'. The number of citations is unknown for literature without an entry in the Scopus database, and no Field-Weighted Citation Impact (FWCI) exists. The Field-Weighted Citation Impact[6] is the ratio of the document's citations to the average number of citations received by all similar documents over a three-year window. Each discipline contributes equally to the metric, eliminating differences in researcher citation behaviour. As the FWCI cannot be calculated for articles without citations, these cases are marked with '-'.

Table 2.1: The literature used for analysis of the AI and PES domain

Study	Journal	# Citations	FWCI
Allhutter, D.; Cech, F.; Fischer, F.; Grill, G.; Mager, Astrid [5]	Frontiers in Big Data	77	4.66
Andonovikj, V.; Boškoski, P.; Džeroski, S.; Boshkoska, B.M. [31]	Expert Systems with Applications	1	1.84
Bach, R.L.; Kern, C.; Mautner, H.; Kreuter, F. [32]	Data & Policy	1	0.46
Berman, A.; de Fine Licht, K.; Carlsson, V. [30]	Technology in Society	3	6.45
Braunsmann, K.; Gall, K.; Rahn, F.J. [33]	Historical Social Research	2	1.38
Considine, M.; Mcgann, M.; Ball, S.; Nguyen, P. [34]	Journal of Social Policy	21	4.14
Desiere, S.; Langenbucher, K.; Struyven, L. [6]	-	-	-
Desiere, S.; Struyven, L. [35]	Journal of Social Policy	25	2.85
Directorate-General for Employment, Social Affairs and Inclusion (European Commission); Scoppetta, A.; Johnson, T.; Buckenleib, A. [25]	-	-	-
Dossche, W.; Vansteenkiste, S.; Baesens, B.; Lemahieu, W. [15]	SN Computer Science	0	-
Haug, K.B. [36]	International Journal of Sociology and Social Policy	2	1.45
Kütük, Y.; Güloğlu, B. [37]	Journal of Research in Economics	-	-
van Landeghem, B.; Desiere, S.; Struyven, L. [11]	IZA World of Labor	-	-
Møller, N.H.; Shklovski, I.; Hildebrandt, T.T. [38]	ACM Other conferences	25	2.60
Mozzana, C. [39]	-	5	0.69
Niklas, J.; Sztandar, K.; Szymielewicz, K. [40]	-	-	-
Troya, I.M. de R. de; Moraes, L.O. [41]	-	9	-
Wan, C.; Belo, R.; Zejnilović, L.; Lavado, S. [42]	Communications in Computer and Information Science	0	-
Zhao, L. [43]	E3S Web of Conferences	4	1.00

[5] https://www.scopus.com
[6] https://service.elsevier.com/app/answers/detail/a_id/14894/supporthub/scopus/

Table 2.2: The literature used for analysis of the XAI and DSS domain

Study	Journal	# Citations	FWCI
Alangari, N.; El Bachir Menai, M.; Mathkour, H.; Almosallam, I. [21]	Information (Switzerland)	2	0.25
Amparore, E.; Perotti, A.; Bajardi, P. [14]	PeerJ Computer Science	48	4.53
Apostolopoulos, I.D.; Groumpos, P.P. [44]	Applied Sciences (Switzerland)	8	2.34
Aslam, M.; Segura-Velandia, D.; Goh, Y.M. [45]	IEEE Access	1	0.34
Barnard, P.; MacAluso, I.; Marchetti, N.; Dasilva, L.A. [46]	IEEE Int Conf Commun	5	2.93
Bayer, S.; Gimpel, H.; Markgraf, M. [47]	Journal of Decision Systems	18	3.21
Brown, K.E.; Talbert, D.A. [48]	Proc. Int. Fla. Artif. Intell. Res. Soc. Conf., FLAIRS	3	0.88
Cau, F.M.; Hauptmann, H.; Spano, L.D.; Tintarev, N. [49]	ACM Transactions on Interactive Intelligent Systems	4	0.77
Chen, V.; Liao, Q.V.; Wortman Vaughan, J.; Bansal, G. [50]	Proceedings of the ACM on Human-Computer Interaction	19	6.98
Christou, I.T.; Soldatos, J.; Papadakis, T.; Gutierrez-Rojas, D.; Nardelli, P. [51]	Proc. - Int. Conf. Distrib. Comput. Smart Syst. Internet Things, DCOSS-IoT	0	-
Cirqueira, D.; Helfert, M.; Bezbradica, M. [52]	Lect. Notes Comput. Sci.	17	5.64
Collenette, J.; Atkinson, K.; Bench-Capon, T. [53]	Artificial Intelligence	19	13.41
Da Silva Oliveira, F.R.; De Lima Neto, F.B. [54]	IEEE Access	0	-
Dandolo, D.; Masiero, C.; Carletti, M.; Dalle Pezze, D.; Susto, G.A. [55]	Expert Systems with Applications	9	2.44
Das, D.; Kim, B.; Chernova, S. [56]	Int Conf Intell User Interfaces Proc IUI	3	3.34
Delen, D.; Davazdahemami, B.; Rasouli Dezfouli, E. [57]	Information Systems Frontiers	2	4.3
Dreyling, R.M.; Tammet, T.; Pappel, I. [12]	Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications	0	-
Gajos, K.Z.; Mamykina, L. [58]	Int Conf Intell User Interfaces Proc IUI	41	11.61
Galanti, R.; de Leoni, M.; Monaro, M.; Navarin, N.; Marazzi, A.; Di Stasi, B.; Maldera, S. [59]	Engineering Applications of Artificial Intelligence	5	1.47
Garouani, M.; Ahmad, A.; Bouneffa, M.; Hamlich, M.; Bourguin, G.; Lewandowski, A. [60]	International Journal of Advanced Manufacturing Technology	29	3.44
Heider, M.; Stegherr, H.; Nordsieck, R.; Hähner, J. [61]	Artificial Life	1	0.46
Lee, H.-W.; Han, T.-H.; Lee, T.-J. [62]	IEEE Access	0	-
Liao, Q.V.; Zhang, Y.; Luss, R.; Doshi-Velez, F.; Dhurandhar, A. [63]	Proc. AAAI. Conf. Hum. Comput. Crowdsourcing.	27	8.98
Löfström, H.; Löfström, T.; Johansson, U.; Sönströd, C. [64]	Annals of Mathematics and Artificial Intelligence	1	0.36
Malandri, L.; Mercorio, F.; Mezzanzanica, M.; Seveso, A. [65]	Decision Support Systems	2	2.26
Mazzola, L.; Stalder, F.; Waldis, A.; Siegfried, P.; Renold, C.; Reber, D.; Meier, P. [66]	Commun. Comput. Info. Sci.	0	-
Mohiuddin, K.; Welke, P.; Alam, M.A.; Martin, M.; Alam, M.M.; Lehmann, J.; Vahdati, S. [67]	Int Conf Inf Knowledge Manage	0	-
Mucha, H.; Robert, S.; Breitschwerdt, R.; Fellmann, M. [22]	Conf Hum Fact Comput Syst Proc	17	2.73
Nguyen, A.; Foerstel, S.; Kittler, T.; Kurzyukov, A.; Schwinn, L.; Zanca, D.; Hipp, T.; Jun, S.D.A.; Schrapp, M.; Rothgang, E.; Eskofier, B. [68]	IEEE Access	6	1.88

Table 2.2: The literature used for analysis of the XAI and DSS domain (continued)

Study	Journal	# Citations	FWCI
Panagoulias, D.P.; Sarmas, E.; Marinakis, V.; Virvou, M.; Tsihrintzis, G.A.; Doukas, H. [69]	Electronics (Switzerland)	6	1.88
Rojo, D.; Htun, N.N.; Parra, D.; De Croon, R.; Verbert, K. [70]	Computers and Electronics in Agriculture	5	0.6
Senoner, J.; Netland, T.; Feuerriegel, S. [71]	Management Science	67	7.62
Shams, M.Y.; Gamel, S.A.; Talaat, F.M. [72]	Neural Computing and Applications	6	10.23
Steging, C.; Renooij, S.; Verheij, B. [73]	Proc. Int. Conf. Artif. Intell. Law, ICAIL	13	2.19
Sufi, F.K.; Alsulami, M. [74]	IEEE Access	21	1.87
Thuy, A.; Benoit, D.F. [75]	European Journal of Operational Research	2	4.6
Tiensuu, H.; Tamminen, S.; Puukko, E.; Röning, J. [76]	Applied Sciences (Switzerland)	3	0.26

2.2.1. YEAR-WISE REPORTING

Figure 2.2 shows the year-wise distribution of the publications assessed in phase seven of this literature review. Amongst the 57 papers, 11 have been published in both 2021 and 2022, while in 2023 more papers than the sum of the two years before it were published; 23. As the research was conducted in February of 2024, only five papers of this year have been assessed. Due to the inclusion and exclusion criteria discussed in subsection 2.1.1, all literature included before 2021 was identified through the snowballing method, represented by 'Snowball' in Figure 2.2. In the graph, the AI and PES domain is represented with 'Query 1', and the XAI and DSS domain is represented with 'Query 2'. The graph shows that AI applications in PES have been researched since 2015 and that there is a growing interest in XAI in DSSs, with a strong peak in 2023.

2.2.2. THEME-WISE REPORTING

Figure 2.3 shows the frequency of appearance of the keywords of the publications assessed in phase seven of this literature review after removing the words used in the search queries. The main concepts are multiple variants of 'machine learning', 'decision making' or 'decision support', and 'interpretability', reflecting the focus of this literature review. Interesting words that pop up are 'human computer interaction', 'human ai interaction', and 'learning systems', showing that this review goes beyond merely technical aspects of XAI in DSS, also reviewing the human factor. Figure 2.4 shows the frequency of words appearing in the title of the publications assessed in phase seven of this literature review. The most occurring words are 'decision', 'system', and 'support', followed by 'explainable', 'artifical', 'intellengence', 'machine', and 'learning', reflecting this literature review's focus.

Figure 2.5 visualizes the co-occurrence of the 25 most frequently appearing words in the titles of the selected research, given a context window of three words. Of interest are the co-occurrences between 'explanation' and 'trust', 'explaining' and 'predicting', 'risk' and 'explaining', 'intelli-

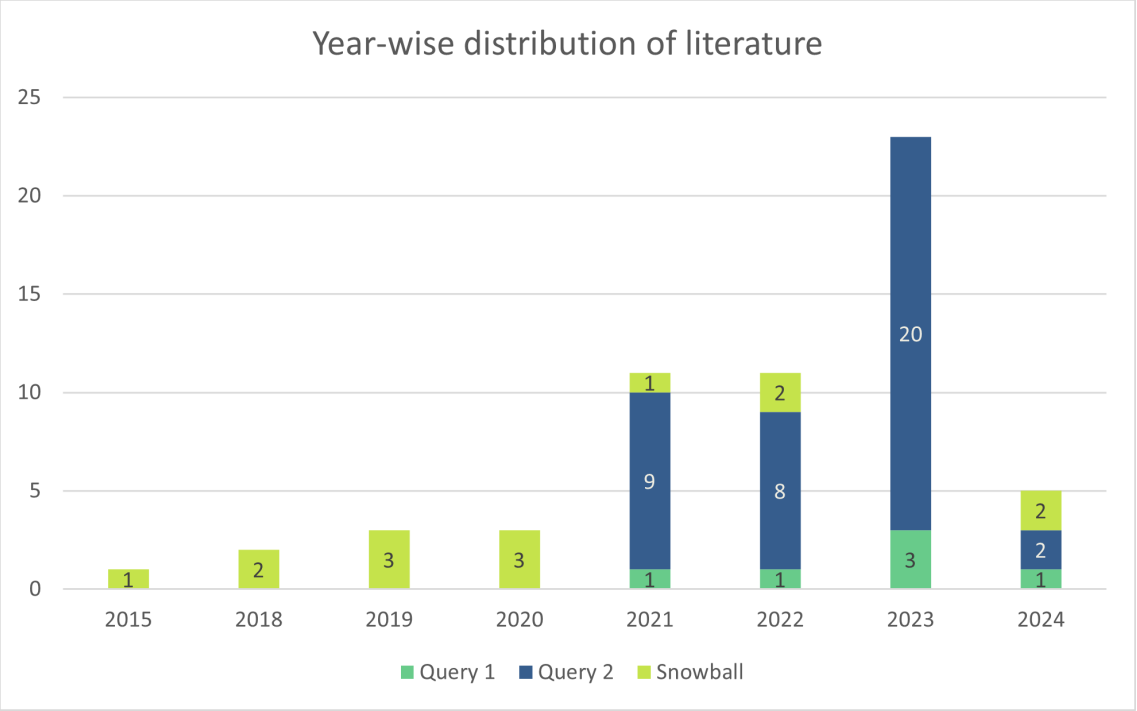


Figure 2.2: The distribution of selected research papers per year

Figure 2.3: A word cloud showing the different keywords of the selected papers

gent' and 'explanations', and 'intelligent' and 'decision'. The first four pairs might indicate relationships between the concepts, but the latter pair more likely refers to a name generally used for a DSS using explanations.

Figure 2.4: A word cloud showing the most appearing words in the titles of the selected papers

2.3. ALGORITHMIC PROFILING IN PUBLIC EMPLOYMENT SERVICES

This section focuses on presenting the results of the literature analysis of the works presented in Table 2.1. Of these works, three papers are reviews. The first review is published by the Organisation for Economic Co-operation and Development (OECD) [6], the second by the European Union [25], and the final one is academic [36]. Twelve papers describe the applications of AI profiling in the PES of a country [5, 30–33, 37, 39, 41, 43, 77], of which only three papers employ a method of explainability [41, 43, 77]. Four papers have been written from a Labour Economics point of view [11, 34, 35, 38], and the final paper employs a Human-Computer Interaction point of view [42].

2.3.1. APPLICATIONS OF ALGORITHMIC PROFILING IN PUBLIC EMPLOYMENT SERVICES

Algorithmic profiling in PES generally has one of three goals: (1) diagnose individual strengths and weaknesses to account for in the personal action plan for a client, (2) assess the risk of Long-Term Unemployed (LTU) of an individual, and (3) targeting the appropriate services, measures and programmes to the profile of the client [6, 25]. As these algorithms are reliant on statistics, they can only find the best predictors of LTU, but not the drivers behind it [36]. Accurate profiling tools improve the cost-efficiency of PES and can be used to guide planning and budget allocations [6, 25]. The high caseload of counsellors pushes further this need for risk profiling tools [25]. Focusing on profiling for early interventions is essential, as these have the highest cost-effectiveness and overall fiscal savings [25]. Additionally, caseworkers are more effective when they are supported by systematic tools [25]. Profiling appears to be a promising way to speed up the unemployment exit by guiding planning and budget allocations [36], providing systematic support to and reducing the caseload of counsellors [25], ultimately leading to a more cost-efficient PES [6].

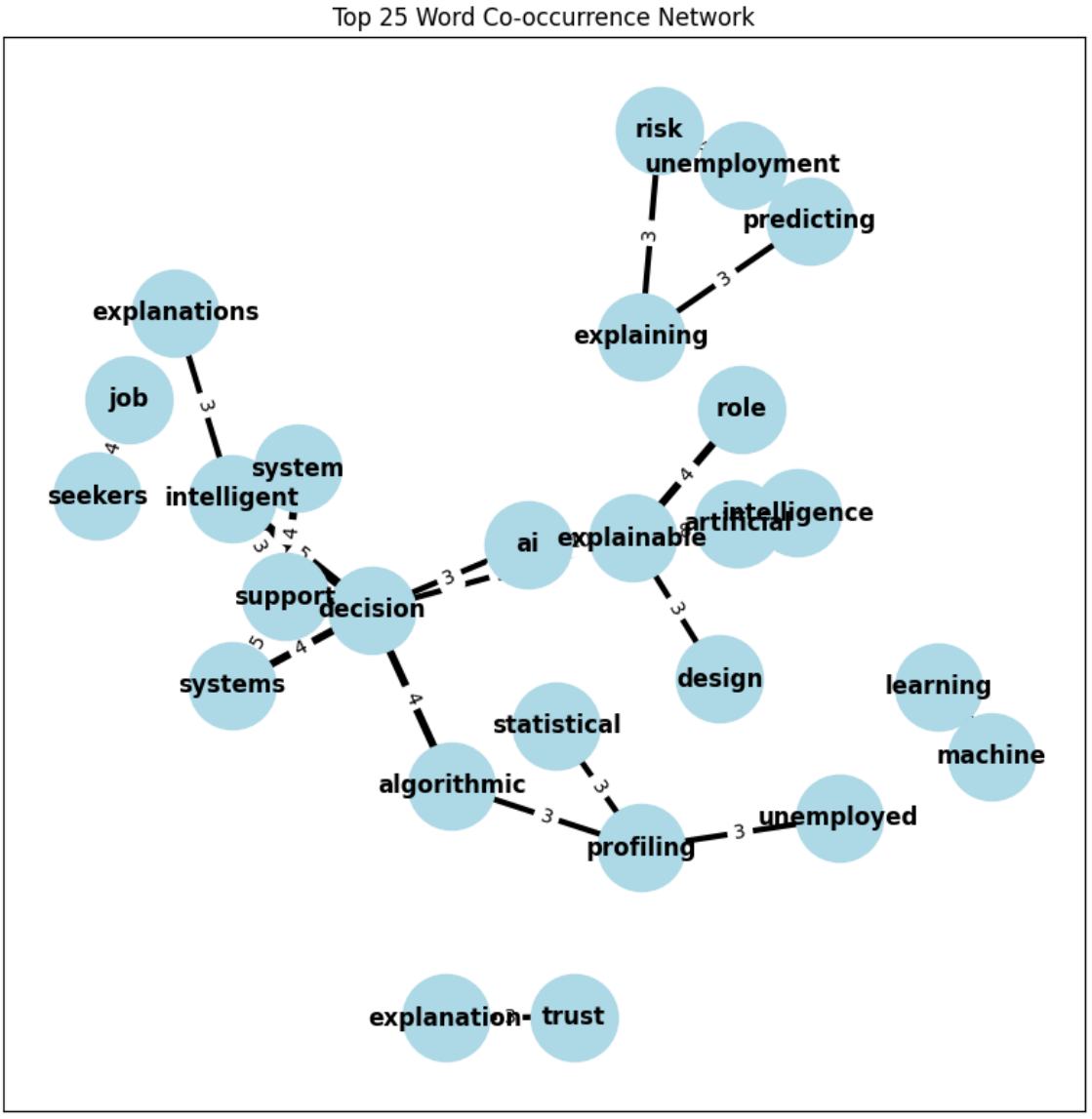


Figure 2.5: A research map showing the most co-occurring words in the titles of the selected papers

In some countries like Denmark and Sweden, the usage of algorithmic profiling is voluntary for both the case worker and the client and is used to support the decision-making process of the case worker [6]. Other countries like Australia and the USA use algorithmic profiling to determine the service streams jobseekers are assigned to automatically [6]. The actions the PES takes on as a result of the algorithmic profiling greatly differ across the world and greatly depend on the culture of the country it is being implemented in.

Desiere et al. [6] distinguish three types of profiling: rule-based profiling, caseworker-based profiling, and statistical (algorithmic) profiling, which are often combined in practice. The Directorate-General for Employment, Social Affairs and Inclusion (European Commission) adds data-assisted profiling as a fourth category [25] and defines this as a profiling method in which the caseworker has a significant role and is supported by quantitative data.

Most statistical profiling tools predict the probability that a jobseeker becomes LTU. The exact definition of LTU differs per country, as does the exact predictive value of the models [6]. Generally, the models calculate the probability of LTU where Denmark, Flanders, Sweden and the US use six months of unemployment as threshold [6, 77]. The cases in Australia, Ireland, Italy, Latvia, the Netherlands, Portugal and Türkiye consider 12 months of unemployment as LTU [6, 37, 43]. Another approach is to calculate a risk profile for an individual, which happens in Austria, Germany, Italy, Poland, Portugal, and Sweden [5, 30, 32, 39–41]. Slovenia is a unique case in which the model employed is a survival analysis which outputs the probability of unemployment as a 365-dimensional vector, with each dimension representing the probability of unemployment on the respective day in the year [31].

The models generally are statistical Logistic Regression (LR) models, see Table 2.3. These models are inherently, or ante-hoc, explainable, meaning that their inner workings can be deducted from parameters. More elaborate models, like, for instance, a Random Forest (RF), have not been widely applied yet [36], as a problem with these models is that their increased complexity not only improves the model performance but also increases model complexity which results in a lack of interpretability. To overcome this struggle post-hoc explainability methods like SHapley Additive exPlaination (SHAP) [31, 41, 43, 77], Local Interpretable Model-agnostic Explanations (LIME) [30] and the Aequitas[7] toolkit to audit bias and discrimination [36] have been applied.

The statistical profiling models employed throughout the OECD generally use a combination of four types of input variables: socio-economic characteristics of the jobseekers, motivation to look for a new job, job readiness, and opportunities [6]. The EU defines slightly different types of variables: personal characteristics, household characteristics, information on education and skills, the jobseeker's work history, health status, locality, and attitudes to job search and work [25]. Haug noted that profiling rarely includes unemployed people's needs, aspirations, or job quality metrics and can represent the harsh realities of structural labour market discrimination [36].

Challenges for algorithmic profiling in PES are to align the complexity of the profiling with the degree of Active Labour Market Policy (ALMP) in place in the country. For countries with minimal ALMP in place, the profiling is not useful, and for countries with strongly differentiated services, the profiling tools have to be complex [25]. Additionally, the people most in need might be hard to profile, as these can be people with psychological issues, debt, or drug addiction [25]. The information to assess this is provided by the clients themselves, either digitally or in a consult with a case worker, leading to the risk of missing key information for assigning a client to the proper profile [25], making the caseworkers crucial to the collection of data [36].

Profiling, therefore, does not come without risks and limitations. Different profiling tools may assign different profiles to the same individual, the accuracy of profiling tools can be lower in practice than in development, the link with service delivery may be missing, and casework-

[7] http://www.datasciencepublicpolicy.org/our-work/tools-guides/aequitas/

ers may not use the tool [6]. Different reasons for caseworkers not using the tool are that they might perceive profiling results as demotivating to jobseekers, find the tools useless and untrustworthy, ignore the tool, or be over-confident in their own experience [36]. In addition, the decisions of the profiling model are as good as the data used to train the model, the most thorough systems are still prone to statistical discrimination, and finally, difficulties in scrutinising or understanding the algorithm can result in a lack of transparency [6, 25]. On the other hand, publishing the algorithm introduces the risk of jobseekers manipulating their information to get assigned to the desired category [25]. Profiling only brings the desired results when there is no shortage of job offers and solely provide value to at-risk individuals [25].

The different applications of AI profiling in PES can be found in Table 2.3. The table reports the country of application, a reference to the source, the type of application, i.e., is it a live system developed and deployed by a government or a prototype build by researchers based on some acquired data source, the model interpretability methods and the performance reported. Some application cases do not report performance measures, as no ground truth is available, i.e., [39, 40].

Table 2.3: Comparison of PES (X)AI applications

Country	Source	Type	Model	Interpretability	Performance
Australia	[6]	Live	LR	Coefficient analysis	Precision = [0.80;0.91]
Austria	[5]	Live	Multivariate LR	Coefficient analysis	Precision = [0.80;0.91]
Denmark	[6]	Live	Decision Tree (DT) Classifier	-	Accuracy > 0.600
Flanders	[77]	Live	RF Classifier	TreeSHAP, TreeInterpreter	Area Under the Receiver Operating Characteristic Curve (AUC ROC) = 0.789
Germany	[32]	Prototype	LR, Penalized LR, RF Classifier, Gradient Boosting Decision Trees (GBDT) Classifier	-	AUC ROC = [0.700; 0.770], Accuracy = [0.837; 0.846], Precision = [0.328; 0.372], Recall = [0.256; 0.290]
Ireland	[6]	Live	Probit Regression	-	Accuracy = [0.700; 0.860]
Italy	[39]	Live	LR	-	-
Latvia	[6]	Live	Factor analysis	-	-
Netherlands	[6]	Live	LR	-	Accuracy = 0.7
New Zealand	[6]	Live	RF, GBDT	-	AUC ROC = [0.63; 0.83]
Poland	[40]	Live	-	-	-
Portugal	[41]	Prototype	LR, RF, XGBoost	SHAP	@10% of population: precision = [0.200; 0.230], recall = [0.640; 0;730]
	[43]	Prototype	LR, RF, XGBoost	SHAP	Accuracy = [0.740; 0.816], precision = [0.726; 0.828], Recall = [0.756; 0.806]. F1-score = [0.741; 0.808], AUC ROC = [0.736; 0.815]
Slovenia	[31]	Prototype	Semi-supervised multi-target regression	SHAP	AUC ROC = [0.76; 0.79]
Sweden	[30]	Live	Neural Network (NN)	LIME	Accuracy = 0.68
	[6]	Live	LR	-	-
Türkiye	[37]	Prototype	LR, RF, XGBoost, Shallow NN	-	Accuracy = [0.63; 0.67]
USA	[6]	Live	LR	-	-

2.3.2. LABOUR ECONOMICS PERSPECTIVE

Desiere and Stuyven [35] acknowledge that profiling models introduce an inherent tension between model accuracy and discrimination. They define accuracy as the share of jobseekers correctly classified and discrimination as the proportion of jobseekers who belong to a particular group and find a job ex-post but are misclassified as high-risk jobseekers relative to the proportion among the dominant group. They show that AI-based profiling increases accuracy compared to randomly selecting jobseekers and a rule-based approach, but also introduces 'statistical' discrimination. Improving accuracy thus comes at the cost of discrimination. Therefore, there exists an accuracy-equity trade-off. They find that jobseekers of foreign origin are 2.6 times more likely to be misclassified as high-risk jobseekers than jobseekers of Belgian origin. They calculate the model's accuracy and discrimination at different profiling thresholds and plot the accuracy versus discrimination at these different thresholds. This results in an inverse U-shaped relation, showing that more jobseekers are considered high-risk for higher thresholds, resulting in low accuracy and discrimination. For lower values of the threshold, fewer jobseekers are considered high-risk. Still, those who are, are mainly of foreign origin, resulting in a low accuracy and high level of discrimination. The visualization shows the trade-offs made, but policies ultimately decide the required levels of accuracy and discrimination, as the model's fairness is also given by the effects of the model and the perceived usefulness of the PES' support. Discrimination can be regarded as either 'negative' or 'positive' depending on the perceived usefulness or value of services offered.

Landeghem et al. [11] uncover that statistical models can reveal systematic patterns between socioeconomic and sociodemographic variables, direct research on the reason why some groups have a higher risk than others, give an indication of the potential duration of an individual's unemployment, and under some circumstances even reduce existing patterns of discrimination. On the contrary, the statistical models are only modestly more accurate than a lottery approach while misclassifying many individuals, introducing the risk of reinforcing existing patterns of discrimination, and only predicting outcomes while not revealing which programs work for whom. Variables found to have explanatory power are a person's region of residence and occupation and their employment history. Macroeconomic conditions, especially the local unemployment rate, can also have strong predictive power. In addition, they state that algorithms are more transparent than the minds of decision-making humans, who often are prone to unconscious biases against certain groups. In addition, statistical profiling models might help identify at-risk people, but they do not reveal which policy programs are effective for whom. Different regimes might prefer different sensitivity–specificity combinations, and the optimal combination depends on how profiling is intended to be used.

Considine et al. [34] explores the feasibility and limitations of replacing human interactions with automated systems, or 'machine bureaucracies', in welfare and employment services. They examine how digitalization can improve efficiency, accountability, and consistency in policy delivery but also consider the necessary trade-offs. While machine bureaucracies promise cost savings, policy fidelity, and reduced stigma for jobseekers, they may also create new forms of

exclusion, particularly for those lacking digital literacy or resources. The authors argue that while digitalization can streamline services, it might fail to address complex individual needs and ethical considerations better handled by empathetic human decision-making.

Møller et al. [38] discuss a research project involving data scientists, caseworkers, and system developers to develop algorithmic decision-support systems for job placement in Denmark. In this study, caseworkers emphasized the need for systems that support their professional responsibilities and maintain the dignity of those they serve rather than just enhancing efficiency [38]. The participatory approach revealed the importance of flexibility and discretion in algorithmic systems, leading to a focus on mitigating organizational contradictions instead of individual profiling aimed at creating more effective and humane decision-support systems. This highlighted the complexity of public service casework, making it difficult to establish clear value metrics for algorithmic tools.

2.3.3. HCI PERSPECTIVE

Wan et al. [42] employs an XGBoost classification model in the context of a European PES that predicts the probability of LTU of clients. They state that algorithms communicate with humans via causal representation and that human representation, based on prior beliefs, may collide with the representation of the algorithm, which is called the duet of representations. Their empirical field research investigates whether explanatory methods, like SHAP and LIME, introduce conflict between the algorithms and the human representations. They find that displaying college education as part of the explanation introduces conflict and decreases decision-making quality. Furthermore, they conclude that understanding the why behind the epistemic conflict should drive human-algorithm interaction. Additionally, they define four necessities to achieve communicative rationality: the human should understand the algorithm's epistemic standpoint, the algorithm should understand the human's epistemic standpoint, the human and algorithm should be able to argue with one another, and both the algorithm and the human should receive feedback on their judgement to improve it.

2.4. EXPLAINABLE ARTIFICIAL INTELLIGENCE IN DECISION SUPPORT SYSTEMS

This section presents the dominant themes in the assessed literature. The findings are classified into XAI Methods in DSSs, subsection 2.4.1, XAI Evaluation Methods in DSSs, subsection 2.4.2, XAI Applications in DSSs, subsection 2.4.3, and, finally, Human XAI Interaction in DSSs, subsection 2.4.4.

2.4.1. XAI METHODS IN DSSS

Many of the works [21, 46, 52, 57, 59, 60, 62, 67–71, 76] assessed by this research use SHAP, proposed by Lundberg and Lee [78], as their primary method of explainability. Another popular explainability method is LIME, proposed by Ribeiro et al. [79], used in [21, 22, 48, 49, 52, 64, 72].

Amparore et al. [14] compare the performance of LIME and SHAP and show that both have vulnerabilities in the form of a limited reiteration similarity, low conciseness or insufficient explanations for the correct label and state that explanations should always be paired with quality metrics.

The literature also contains studies that propose a new explainability method. Of these new explainability methods, two are somewhat similar to LIME and SHAP: Accelerated Modal-agnostic Explanations (AcME) [55], and ModEl constRastive expLanatIoN (MERLIN) [65]. First, AcME combines the computational simplicity of partial dependence plots with the versatility and visualizations of feature importances of SHAP. AcME supports both global and local interpretability and is model-agnostic for tabular data in both regression and classification problems, with the main advantage over SHAP being the improved speed and stability [55]. Second, MERLIN allows understanding how two textual or tabular classifier ML models work beyond their prediction performances. It shows that the explanations are crucial to understanding their behaviour, differences, and likeness [65].

Three of the new explainability methods are rule-based methods: Fuzzy Cognitive Maps (FCMs)[44], Minimum Converging Set [51], and Rationale Discovery [73]. These methods differ from the more traditional LIME and SHAP discussed in, respectively, section 3.4 and section 3.5. First, FCMs are models which infer causality derived from collective human knowledge between phenomena and features. Thus, FCMs are not intended to learn from data but rather to model knowledge and combine this with the results of ML to allow decision-makers to make more informed decisions, including causality to the reasoning [44]. Second, the goal of the minimum converging set is to find a minimal cardinality set of variables in the training dataset that are used by a (sub)set of the extracted rules that together cover the entire training set. For each instance in the training dataset, at least one rule exists in the (sub)set that satisfies the data instance [51]. Third, rationale discovery is a method that mimics unit testing in professional software development. It shows that with carefully designed test cases, a machine-learning engineer can assess if all of the rationale elements presented in the training set are correctly recognized and reproduced by the model [73].

Another new explainability method is Subgoal-Based Explanations for plan-based DSSs [56]. These explanations supplement traditional DSS output with information about the subgoal to which the recommended action would contribute and improve user task performance in the presence of DSS recommendations, improve user ability to distinguish optimal and suboptimal DSS recommendations, and are preferred by users [56]. One distinctive explainability method is AMLBID, an advanced AutoML tool that leverages *meta-learning*, estimating the performance of proposed ML models and hyperparameters to data characteristics based on characteristics of other datasets, to automatically select and tune the optimal machine-learning configurations tailored to specific problems, being the first AutoML tool to support different predictive performance measures and provide transparent, auto-explainable recommendations for machine-learning configurations [60].

The uncovered XAI methods in DSSs are classified according to the taxonomy of interpretability dimensions found in [21]. They distinguish between Stage, the nature of explainability, and Scope, whether the XAI can calculate local and/or global feature scores. The papers presenting XAI methods in DSSs assessed in this work are classified in Table 2.4. Several methods listed in the table were not initially identified in the literature review. They are incorporated as they were utilized in studies that were part of the reviewed literature. Of these 19 assessed methods, 11 support global explanations, and 11 support local explanations. Three of the methods support both global and local explanations. Most assessed methods are Model-Agnistic, whereas methods that aim to create a sense of causality require domain knowledge and are thus Model-Specific.

Table 2.4: XAI Methods in DSSs

XAI Method	Scope	Stage
SHAP [78]	Global & local	Model-Agnostic
LIME [79]	Local	Model-Agnostic
Fuzzy Cognitive Maps [44]	Global	Model-Specific
QARMA [51]	Global	Model-Agnostic
AcME [55]	Global & local	Model-Agnostic
Subgoal-Based Explanations [56]	Local	Model-Specific
MERLIN [65]	Global	Model-Agnostic
Test case based XAI [73]	Local	Model-Agnostic
Partial Dependence Plot [80] [a]	Global	Model-Agnostic
Accumulated Local Effects[81] [a]	Global	Model-Agnostic
Parallel Coordinates Plot [82] [a]	Global	Model-Agnostic
Natural Language Explanations [83] [a]	Global & local	Model-Agnostic
Counterfactual Explanations [83] [a]	Local	Model-Agnostic
Contrastive Explanations [83] [a]	Local	Model-Agnostic
Explainable Clustering [66] [a]	Global	Model-Specific
Anchors [84] [a]	Local	Model-Agnostic
Abstract Dialectical Framework [85] [a]	Global	Model-Specific
Inference Graphs [86] [a]	Local	Model-Specific
Layer-Wise Relevance Propagation [87] [a,b]	Local	Model-Agnostic

[a] This method was not originally included in the structured literature review.
[b] This is an image-specific explanation method.

2.4.2. XAI EVALUATION METHODS IN DSSS

The majority of this subsection leans on the output of a recent survey of evaluation methods for interpretable ML by Alangari et al. [21], which produces a taxonomy of interpretability evaluation methods, seen in Figure 2.6, and a summary of current evaluation methods, as seen in Table 2.5. Consecutively, a Local Explanation evAluation Framework (LEAF) has been presented by Amparore et al. [14], which uses different aspects of explanation quality partially aligned with the computational-based branch of the taxonomy presented by Alangari et al. [21] to evaluate explanations given by several explainability methods [14], showing that both SHAP and LIME are plagued with defects. Building on the computational-based branch, a way of model evaluation may be to communicate uncertainty. In this area, Thuy and Bernoit [75] claim that a prediction has two uncertainty values, as uncertainty can arise from two fundamentally different sources, data uncertainty and model uncertainty, allowing the human-in-the-loop expert

to adjust the prediction for uncertain observations. A model operator can still use a model performing less than optimally globally to make decisions about problem instances where the model is well-fitted [61]. A different approach is integrating domain knowledge into a visual support system and using this domain knowledge as ground truth, evaluating the overlap between model output and expert knowledge [70]. Rojo et al. [70] build forth on the work of Gil et al. [88], which suggested that information from domain knowledge is a potential strategy for evaluating and selecting a model. A disadvantage of this method is that human-based evaluation methods introduce subjectivity and biases to the evaluation as such a metric rewards alignment and similarity instead of faithfulness and correctness [21]. As XAI has different requirements in different usage contexts, the evaluation of XAI should not be static but rather differ between usage contexts. Liao et al. [63] investigate the relative importance of XAI evaluation criteria in five different usage contexts of XAI, summarized in Table 2.7. Furthermore, Löfström et al. [64] build on the idea of evaluation of explanations and investigate the impact of calibrating predictive models with Venn-Abers and Platt scaling before applying LIME on the quality of the explanations. They find that better-calibrated models result in better explanations, giving a more accurate representation of reality [64]. Generally, authors agree that good explanations are comprehensible, actionable, and interactive [21, 63, 75]

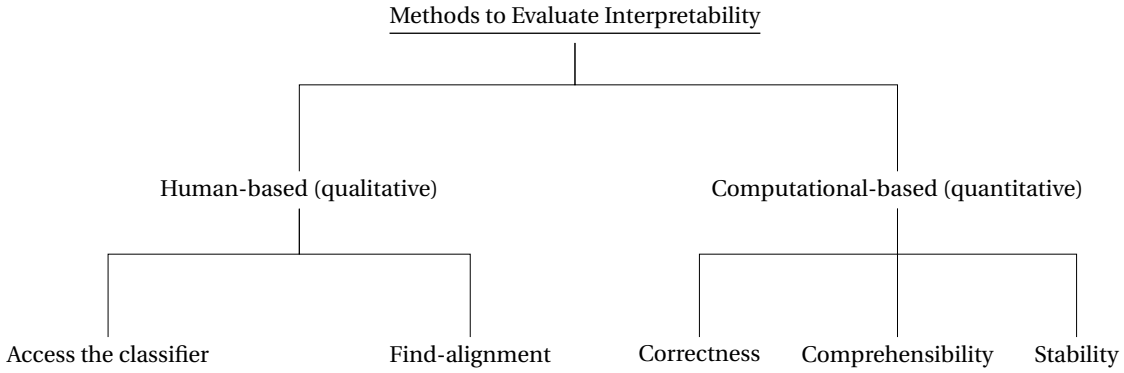


Figure 2.6: Methods to evaluate interpretability, adapted from [21]

Table 2.5: Lists of Computational- and Human-based evaluations methods, adapted from [21]

Computational-based	**Human-based**
Partial Dependency Plots (PDP)	Select the best classifier
SHapley Additive ExPlanations (SHAP)	Improve a classifier
Local Interpretable Model-agnostic Explanations (LIME)	Identify classifier irregularities
TREPAN	Forward Simulation
Feature weights/importance	Counterfactual explanations
Class Activation Mapping (CAM)	Describe the class characteristics
Saliency/heat maps	Alignment between humans and models
GoldenEye	Select the best explanation
	Verification
	Understand internal reasoning process
	Reconstruct target instance

Table 2.6: The explanation quality metrics as proposed in LEAF [14]

Metric	Description
Conciseness	The maximum number K of non-zero weights that are kept in the explanation presented to the user, while the other $F-K$ features are treated as non-relevant and are excluded
Local fidelity	Measures how good is the white-box g in approximating the behaviour of the black-box f for the target sample x around its synthetic neighbourhood $N(x)$.
Local Concordance	It measures how good g is in mimicking f for the sole instance x under the conciseness constraint.
Reiteration similarity	Measures the similarity of a set of explanations of a single instance x as a measure of similarity across multiple reiterations of the explanation process.
Prescriptivity	Measures how effective an LLE is when taken as a recipe to change the class of x.

Table 2.7: Requirements of XAI in different usage context, adapted from [63]

Usage context	Requirements
Model improvement	Explanations should be faithful and stable for users to inspect the true causes of model issues.
Capability assessment	Explanations should be easy to understand to help users make an efficient assessment, even at some cost of faithfulness.
Decision support	Explanations should communicate the uncertainty of model predictions and the limitations of explanations.
Understanding how the system works with one's data to make adjustments	Explanations should be transparent about the model uncertainty and explanation limitations, faithful and stable to allow users to discover and change the true causes of sub-optimal system behaviours.
Learning	Explanations should be stable, faithful and relatively complete to help people discover true patterns of the domain.
Model auditing	Explanations should be first and foremost faithful to the underlying model, transparent about the explanation limitations, and cover the decision components completely, to allow accurate and comprehensive auditing.

Additionally, the uncovered XAI Evaluation Methods in DSS are classified according to the taxonomy of Interpretability Evaluation Methods presented by Alangari et al. [21]. They distinguish between Human-based and Computational-based metrics, i.e., qualitative and quantitative, further branched into five specific categories. The classification of the XAI Evaluation Methods in DSSs is presented in Table 2.8. Most of the found evaluation methods are of a quantitative nature. These methods generally use an algorithm to assess the quality of an explainability and are thus objective. There are differences between the various quantitative explainability methods, but all four methods measure at least the correctness of an explanation. Two of the methods add the measurement of the comprehensibility and stability of an explanation to the quality of an explanation. Only one method combines qualitative and quantitative measures, and another relies solely on qualitative measures. Both of these methods use at least the find-alignment metric as a quality measure. The latter method also includes the access the classifier metric. The qualitative methods are prone to biases, as they rely on human judgement [21]. Additionally, Liao et al. [63] state that the classification is not necessarily binary, as it is possible to devise proxy measures to assess human-centred criteria. Proxy measures were not found in the assessed methods.

Table 2.8: XAI Evaluation Methods in DSSs

XAI Evaluation Method	Qualitative or Quantitative	Branch	Task/metric
Augmented Human Model Selection (AHMoSe) [70]	Qualitative	Access the Classifier	Select the best classifier
		Find-alignment	Alignment between humans and models
Local Explanation evAluation Framework (LEAF) [14]	Quantitative	Correctness	Local Concordance Prescriptivity
		Comprehensibility	Conciseness Local Fidelity
		Stability	Reiteration similarity
Contextualized Evaluation for Explainable AI [63]	Quantitative	Correctness	Faithfulness (Un)Certainty (communication) Translucence
		Comprehensibility	Completeness Compactness Comprehensibility
		Stability	Stability
	Qualitative	Find Alignment	Coherence
	None	None	Interactivity Actionability Novelty Personalization
Model calibration evaluation [64]	Quantitative	Correctness	Calibration
Uncertainty Estimation [75]	Quantitative	Correctness	(Un)Certainty (communication)

[a] This method was not originally included in the structured literature review.
[b] This is an image-specific explanation method.

2.4.3. XAI APPLICATIONS IN DSSS

From the body of literature, only one XAI application in DSSs is in unemployment insurance [15], the central subject of this literature review. However, many methods and frameworks applied to other industrial applications can be transferred to unemployment insurance. Therefore, this chapter analyses the different methods and frameworks throughout the many industrial applications of XAI in DSSs, starting with unemployment insurance.

The Estonian unemployment insurance fund has created OTT[8], a DSS showing counsellors the risk of unemployment of their clients with the aim to enhance effectiveness and increase efficiency in the unemployment counselling process. This system allows a counsellor to put more time and effort into the people who potentially need more services to get employed and additionally balances the workload between counsellors. The system uses a RF ML model with 40 input features to predict the chance that someone gains employment within 180 days and uses a human-in-the-loop approach to ensure quality decision-making [12].

Attrition[9] prevention is also an area where XAI is used in DSSs. This area is characterised by the object of the decision, a human, which brings all sorts of extra challenges, like privacy and discrimination, with it, aligning with the issues in unemployment insurance. Another issue

[8] https://e-estonia.com/ai-to-help-serve-the-estonian-unemployed/
[9] Voluntarily or involuntary leave

with human data found in the literature is noise in the data due to manual labelling, subjectivity in class labels, and class imbalance [57, 67]. In attrition prevention, the frameworks consist of data preprocessing, predictive models and explainable ML approaches in the form of SHAP [57, 67]. Delen et al. [57] aims to predict and mitigate freshman student attrition. They show that feature importance might be helpful for group-level insights, which allows for the adjustment of long-term strategies. These group-level insights should not be used to pursue a one-size-fits-all strategy on an individual level, but rather, an individual intervention should be designed. HR-DSS is a DSS that aims to predict employee attrition, helping the decision-maker to understand the relationship between specific features and the employee's decision to resign and design individual interventions with the help of a what-if analysis. This allows the user to change specific feature values of an individual to explore the impact on the probability of attrition, according to the ML model [67]. Nguyen et al. [68] leverage data from log and enterprise data-based sources to create an XAI customer support system to identify customers who might need special attention by calculating each customer's escalation probability. This allows customer support to efficiently allocate resources to improve customer satisfaction through more proactive customer relations. The authors state that in practice, an investment of 5 hours per week might lead to a reduction of nearly 50% of escalations yearly for the company in this case, and they propose an employee workflow to achieve this.

Major application areas of XAI in DSSs are manufacturing and quality control [61, 71, 76]. These use cases all rely on black-box ML models that use an XAI approach to allow human-in-the-loop decision-making. The ML methods used in this application area are Learning Classifier Systems (LCSs) [61] and GBDT [71, 76]. One of the main issues in quality management is finding the root cause of a problem. ML models cannot guarantee causality but rely on the process engineers to do so [71]. However, ML methods can produce outputs that can function as input to the decision-making process, exploiting the unique capability of computers to quickly process large amounts of data and extract information from it. Applications in this area include a DSS to inspect the predicted quality of the output before the production process has started [76] and finding the processes that need quality improvement and then selecting improvement actions [71]. By incorporating three reasonability aspects into decision systems: feasibility, rationality, and plausibility, similar reasonable decisions to the same problem can be generated, allowing the decision-maker to select one of these decisions with the help of a comprehensible and justifiable explanation [54].

Another application area with research into XAI in DSSs is agriculture [70, 72], using both white- and black-box models, complimented by either SHAP or LIME to allow for human-in-the-loop decision-making. Rojo et al. [70] apply regression models in a viticulture context and use SHAP in combination with domain knowledge to present and compare the output of different regression models in an interactive dashboard. Shams et al. [72] present XAI-CROP, aiming to bring farmers comprehensible crop recommendations. XAI-CROP uses a five-block architecture which can be transferred to other application areas: a data preprocessing module, a feature selection module, a model training module, an XAI integration module, and a validation mod-

ule.

Cyber security is another area in which XAI is integrated into DSSs. This area is characterised by high velocity and volume imbalanced data, which make human decision-making nearly impossible but are well suited for ML approaches [59, 62, 66]. One of the issues is that over-sensitive models are responsible for many false alarms, therefore, Lee et al. [62] employ false positive reduction in the form of an anomaly detection model based on DeepAID [89]. This is interpreted by the XAI module, which uses SHAP, contrastive explanation method, and counterfactual explanations to see if the false alarm reduction made a reasonable decision by finding the K-nearest neighbours. It then calculates the absolute difference with weighted distances, using the feature importance of the AI model, reducing the false positive rate. Security rule identification and validation also allows for the usage of AI to define uncommon patterns and security threats to be monitored and vetted with a noise-resistant density-based spatial clustering approach in the form of Density-based Spatial Clustering of Applications with Noise (DBSCAN) and anomaly detection in the form of a density-based algorithm that calculates a factor representing the degree of anomaly or novelty called Local Outlier Factor (LOF) [66]. Consecutively, the XAI system proposes a dimension based on the outliers to generate a new rule that can be, if accepted, integrated into the security rule engine. The current rule and its effects on a random set of data from the logs are then graphically displayed. In the subdomain process mining, Galanti et al. [59] compare Catboost and Long-Short Term Memory (LSTM) NNs on several real-life process predictive analytics tasks. They use SHAP to return both global and local explanations. Both models perform equally, however, the Catboost model is trained faster, which is preferred in real-time analytics. The authors [59] conduct a user study in which they gather feedback on the task difficulty of 18 tasks, which are found to be generally easy to execute, also for non-expert users, showing that predictions are explained in an effective and efficient form for process analysts.

Resource management scholars have also applied XAI in DSSs in energy management [69] and communication network slice reservation, the allocation of dedicated network resources for specific services within communication networks [46]. Where Barnard et al. [46] verify their AI with a mathematical model and employ SHAP to monitor and verify the real-time decisions of the model, revealing trends about the model's general behaviour and diagnosing potential harmful behaviours in advance of real-world deployment of the model, Panagoulias et al. [69] decide to employ a stacked NN with customised XAI analytics for the various groups of energy management stakeholders. To achieve this, they create a questionnaire, conduct a survey among stakeholders, and perform user clustering to define XAI systems tailored to user needs. They applied the Technology Acceptance Model (TAM) [90] and associated perceived usefulness with AI literacy and perceived ease of use with usability.

XAI has also been employed in legal DSSs, to evaluate the admission of claims to the European Court of Human Rights [53] and to evaluate rationales in tort law [73]. The models are both rule-based domain knowledge models requiring manual transcription of text to input features,

but where Collenette et al. [53] start with rules and then process data according to these rules, presenting the user with its verdict and a textual explanation, Steging et al. [73] do the opposite, creating carefully designed artificial training sets from a real-life legal setting with inherent logical rationale, aiming to extract five known knowledge structures in tort law. They conclude that NNs failed to learn one of the independent conditions despite high accuracy on the test set.

Sufi and Alsulami [74] present a fully automated media monitoring solution that automatically analyses unstructured global events, highlights significant events to users, and explains the root causes of abnormalities in plain English. The system automatically categorises entity information into twelve entity types, extracts the sentiments on events, compares the sentiments on a time-series and detects whether the event is anomaly or regular. The system can be used by policymakers to make evidence-based policy decisions or by news agencies and reporters as their eyes and ears for new articles.

Additionally, XAI can be used to understand epistemic uncertainty in deep NNs [48]. Brown et al. [48] use Bayesian dropout uncertainty estimation, distilled into regression-based ML models to which XAI is applied to evaluate the XAI-generated explanation of the model uncertainty by examining whether the explanation correlates with negative relevance in classification tasks, pinpointing specific features that lead to a data point being considered out-of-distribution, functioning as a guideline for human expert intervention.

In Table 2.9, all application domains of XAI in DSS are reviewed and classified according to the XAI method and XAI evaluation methods from, respectively, Table 2.4 and Table 2.8. Generally, SHAP is the preferred XAI method and is used in 11 of the 20 application domains, followed by LIME, which is used three times. Additionally, Natural Language Explanations are used twice, once in collaboration with SHAP. By providing a well-designed query with the SHAP values to a natural language model, the SHAP values can be presented to the human decision-maker in a textual manner instead of as a graph, allowing for more traditional, rule-based explanations. The evaluation methods found in this literature review were not explicitly applied to the XAI methods in practice. 11 out of 20 applications did not mention the evaluation of XAI, and the other nine mentioned it implicitly. Only one work used a quantitative approach, whereas eight used a qualitative one. The evaluation of calculated feature importance is the most popular XAI evaluation method found in the literature, being used three times.

Table 2.9: XAI application cases and their XAI Methods and XAI Evaluation Methods

DSS application case	Source	XAI Methods	XAI Evaluation Methods
Unemployment Insurance	[12]	Not mentioned	None
Manufacturing	[71]	SHAP	None
	[76]	SHAP, Partial Dependence Plot (PDP), Accumulated Local Effects (ALE), Parallel Coordinates Plot	User evaluation by comparison

Table 2.9: XAI application cases and their XAI Methods and XAI Evaluation Methods (continued)

XAI Method	Global Importance	Local Importance	Stage
	[61]	Learning Classifier Systems	None
Agriculture	[70]	SHAP	Agreement between model and expert knowledge
	[72]	LIME	None
Cyber Security	[62]	SHAP, Contrastive Explanation Method, Counterfactual Explanations	None
	[66]	Explainable Clustering	None
Attrition Prevention	[57]	SHAP	Evaluation of calculated feature importance
	[67]	SHAP, Natural Language Explanations	Evaluation of calculated feature importance
Resource Management	[46]	SHAP	Local Accuracy, Mask-Based Metrics (Keep Positive, Remove Negative, Remove Absolute, Remove Positive, Keep Negative, Keep Absolute)
	[69]	SHAP	None
Fraud Detection	[52]	LIME, Anchors, SHAP	Estimated user confidence based on Average Prediction Switching point (ASP) for instantiated explanation methods of Local Feature Importance (LFI), Global Feature Importance (GFI) and Feature Impact (FI)
Legal Reasoning	[53]	Abstract Dialectical Framework (ADF)	Accuracy compared to ground truth
	[73]	None	Detailed comparison of actual and expected outcomes
Predictive Process Analytics	[59]	SHAP	Evaluation of calculated feature importance
Customer Support	[68]	SHAP	None
Global Event Analysis	[74]	Natural Language Explanations	None
Candidate Solutions Generation	[54]	Inference Graphs	None
Understanding Uncertainty in Deep NNs	[48]	LIME, Layer-Wise Relevance Propagation (LRP)	None

2.4.4. HUMAN XAI INTERACTION IN DSSS

As explainability is a crucial tool for building user trust [69], this section aims to explain how the design of XAI systems influences the users' trust in the system. Additionally, the design of an XAI system can influence the capability of the user to learn from his interactions and cognitively understand why a decision has been made, allowing them to eventually reproduce the decisions and detect wrong predictions [58]. This section first introduces the concept of

trust, then we discuss the implications on the design of XAI in DSSs, introducing some design frameworks.

Trust can be divided into trusting beliefs, trusting intentions, and trusting behaviour, where trusting beliefs positively influence trusting intentions and trusting intentions positively influence trusting behaviour [47]. Bayer et al. [47] employ structured equation modelling and show that the user's expertise also plays an important role in this equation, as it negatively affects trusting beliefs, and additionally moderates the effect of the AI system's explanation on the trusting intention. For users with higher expertise, receiving an explanation increases the trusting intention, which is in line with the findings of Panagoulias et al. [69], highlighting the need for proper explanations in an expert DSS. In addition, Thuy and Bernoit [75] show that incorporating model uncertainty into the explanation increased the user's trust, as it enables the experts to distinguish uncertain and possibly incorrect predictions from more certain predictions allowing users to make decisions about problem instances where the model is well-fitted [61, 76, 91]. This also helps align the model with the user's expectations and point of view, as humans trust and are satisfied with a model's explanation if it matches their expectations and their point of view [21, 45], showing a potential threat to organizations. As humans are generally visually oriented, visualisation is key to increasing the user's trust in the system, as they help the users understand the results [76]. A last remark is that instability of interpretation limits its trust [21], highlighting the importance of choosing a stable explainability method and the relevance of the stability evaluation measure presented in subsection 2.4.2.

When it comes to the design of XAI systems, visualisation of the models is key to improving the user's understanding of the results, containing different visualisations for different user needs [76], delivering explanations that align closely with specific user expectations [45]. The expectations can be transcribed to requirements, which allow insight into the types of explanations and explanatory elements that are most useful and meaningful for the users, highlighting the need for explanations to align with the domain-level jargon [45]. Multiple authors [21, 49, 75] warn that humans might unquestioningly trust the XAI explanations. This overreliance is shown to be stronger for textual data than for tabular data [49]. Additionally, feature-based explanations increase the users' reliance on AI [91]. When users over-rely on the XAI prediction, they fail to recognize when the XAI decision might be wrong when a situation is not yet found in the training data distribution, accentuating the need for clear uncertainty communication by, for instance, indicating when an observation lies outside of the training data [75].

Therefore, the style of the explanation is also an aspect that needs to be carefully considered, as inductive-style explanations lead to an overreliance on XAI's advice, suggesting that abductive and deductive styles are better at preventing overreliance [49]. Additionally, feature-based explanations are found to increase participants' reliance on XAI regardless of whether the AI system is correct or incorrect. Example-based explanations helped achieve complementary human-AI performance in the study by increasing appropriate reliance when the AI system was correct while helping participants maintain their accuracy when the AI system was incor-

rect [91]. As overreliance reduces incidental learning due to the lack of cognitive engagement, XAI systems should be designed to minimize user overreliance while improving decision quality [58]. The visual design of an explanation affects the decision quality of the user [22]. The default LIME plot performs significantly better in giving an explanation than different plots containing the same information, showing the relevance of the exact form of visual representation of explanations [22]. Additionally, in plan-based models, Das et al. [56] show that subgoal-based explanations enable more robust user performance in the case of Intelligent DSS failure, showing the significant benefit of training users for an underlying task with subgoal-based explanations.

Multiple authors have presented design frameworks to guide an XAI designer through the design needs of an XAI system. These frameworks reach from an application-independent seven-question design framework, which assesses the fit between the use case and the XAI model [61] to a design principle framework specifically created for fraud detection usage contexts [52]. Another work uses mental models to obtain explainability requirements, showing that by embracing user-centric approaches and leveraging domain-specific scenarios XAI systems can be designed to deliver explanations that align closely with user expectations [45]. General design rules are that explanations should be aligned to human decision intuition and rationale [91], presented visually [22], exclude the AI's recommendation [58] and present the user with the AI's limitations [91].

2.5. GAPS IN THE LITERATURE

This research has investigated the application of AI in PES and the application of XAI in DSS to assess the possibilities of connecting these previously unconnected domains, addressing a gap in the current XAI in PES literature. From the AI in PES domain, we can transfer the risk, challenges and opportunities of AI applications in PES, as these are likely to translate to the XAI in DSS in PES domain this research aims to cover. From the XAI in DSS domain, we can transfer the technical aspects of an implementation. Namely, we can transfer the XAI methods, XAI evaluation methods, and user acceptance and design implications from the Human XAI Interaction in DSSs areas. We therefore conclude that we can combine the knowledge attained from the AI in PES and the application of XAI in DSS into an application of XAI in the DSS of a PES. This is further supported by the recent publication of Dossche et al.[15] in the exact research domain.

Further gaps found in the literature are:

- **Efficiency vs dignity of unemployed**: Much of the research acknowledges the fact that digitalization, and more specifically, advanced data analytics, has the potential to streamline services and mitigate organizational inconsistencies but might fail to address complex individual needs and ethical considerations better handled by empathetic human decision-making. In addition, statistical profiling models might help to identify at-risk people but do not reveal which policy programs are effective for whom. However, we did not identify any research on predicting effective policy programs or how the efficiency-

dignity tradeoff is made.

- **Influence of user expertise on trust in AI**: Multiple sources investigate the influence of user expertise on trust in AI. The sources all use mock scenarios with little at stake. Therefore, it would be interesting to understand how the expertise of a PES counsellor influences their trust in the AI system.
- **Uncertainty**: Literature suggests communicating model and data uncertainty to the user of XAI system to allow the user to adjust the prediction for uncertain observations without introducing subjectivity and biases to the evaluation. This is especially important as the datasets in the PES domain contain subjective and manually labelled data. However, no literature has been identified on objectively quantifying the data and model uncertainty.
- **Explainability methods**: The research assessed either uses SHAP or LIME as method of explainability in PES, while many other potential explainability methods have been created. In addition, many of the newly created explainability methods have not been applied to a real use case.
- **Visualization of explainability**: One of the articles assessed by this review has investigated the effect of the visualization of LIME explanations of the understanding of the user. As SHAP is generally the most used explainability method, it would be interesting to see how the visualization of SHAP values influences the understanding of the user.
- **Evaluation of explainability methods**: Most of the research that uses XAI methods do not evaluate the performance of these methods. At the same time, it has been proven that they tend to behave unreliably. In the scarce examples where XAI is evaluated, qualitative instead of quantitative evaluation is performed.

3

METHODOLOGY

3.1. CROSS-INDUSTRY STANDARD PROCESS FOR MACHINE LEARNING

This study used the CRoss Industry Standard Process for Machine Learning (CRISP-ML) process model [92], an updated version of the CRoss Industry Standard Process for Data Mining developed in 2019, to streamline the development process of the complex exercise of developing a ML model with the Necessary Level of Interpretability (NLI). The NLI is a combination of the accuracy of the ML model and the extent of understanding of the inputs, inner workings, outputs, user interface, and deployment aspects of the ML solution required to achieve project goals established and documented at the initiation stage of the project. A ML system is *sufficiently interpretable* when the NLI is achieved. An overview of the process model of the CRISP-ML methodology is shown in Figure 3.1.

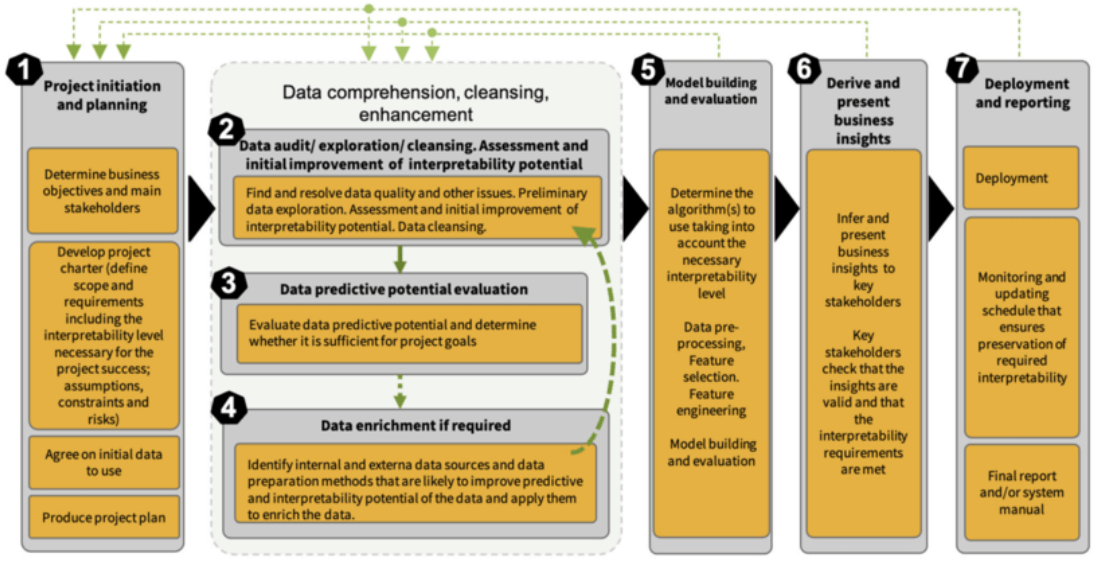


Figure 3.1: CRISP-ML Process Model, adapted from [92]

The CRISP-ML model accommodates modern ML techniques, creates the NLI through the whole ML solution creation process, and is industry-, tool-, and application-agnostic. Although the model looks like a linear process, each stage allows for revisiting previous stages. It consists of a total of seven stages [92]:

1. **Project initiation and planning**: Initially, the project team works together with the other stakeholders to establish project objectives, project scope, and assumptions. The project team also establishes and documents the NLI, including the minimal required accuracy and understanding of inputs, outputs, inner workings and implementation aspects specific to the project.

2. **Data audit, exploration and cleansing**: The next stage demonstrates that the data inputs into the solution are of adequate quality, make sense to the domain data specialists and represent the real-world data on which the solution will be deployed.

3. **Data predictive potential evaluation**: With the outputs of the previous stage, ML methods are now used to assess the predictive potential of the data quality.

4. **Data enrichment**: In this stage, additional internal and external data sources are identified and go through stage 2, adding them to the previously used data. Then, stage 3 is repeated to assess the predictive potential of the enriched data. The step is repeated until the required predictive potential is achieved.

5. **Model building and evaluation**: This stage includes the selection of ML techniques to be used in modelling, preprocessing the data accordingly, and building the models. Additionally, the performance of the models is evaluated, and the best model is chosen.

6. **Derive business insights**: Then, the executives and end-users of the system gain an understanding of the business insights the model produced. Additionally, this stage functions as a confirmation that the model insights are valid and valuable and trust in the insights can be built.

7. **Deployment and reporting**: Finally, the ML model is deployed, and a monitoring and updating schedule is prepared. Additionally, a technical report is created.

Considering the context of the PES in which the predictive model is created, the CRISP-ML methodology provides a structured and practical approach that ensures a proper level of NLI throughout the development process. The CRISP-ML methodology ensures that the ML model predictions about customers' length of unemployment are accurate, transparent, and presented understandably to the counsellors.

3.2. MACHINE LEARNING MODELS

In the ML field, there exists a clear distinction between two different types of classic ML models: unsupervised and supervised [93]. The unsupervised ML models do not require a ground truth for the training, but they rather rely on patterns in the provided data to define similar groups

within the dataset [93], also called clustering. The supervised ML models do require a ground truth for the training [93]. With this ground truth, they aim to find patterns in the data that can be exploited to predict the outcome of data without a ground truth. These models are generally called predictive models. This section will introduce the unsupervised and supervised models used in this research.

3.2.1. UNSUPERVISED MODELS

The unsupervised ML models presented in this section can be divided into two major categories, partitional and hierarchical clustering [94]. While partitional clustering determines all clusters at once, hierarchical clustering creates a hierarchy of clusters by merging or smaller clusters or dividing larger clusters in, respectively, a bottom-up or top-down process.

HIERARCHICAL CLUSTERING

Hierarchical Clustering was first described by Ward and Joe in 1963 [95]. The method forms hierarchical groups of k mutually exclusive subsets based on the similarity of X in n dimensions. Each subset initially consists of only one X. The method calculates all possible unions of k subsets to find the union with the lowest Error Sum of Squares from the $k(k-1)/2$ possible pairs that can be formed. The process can be repeated until all subsets are in one group. Hierarchical clustering can be employed with any distance metric and needs either the number of clusters to find or a linkage distance threshold above which clusters will not be merged. Hierarchical clustering can discover non-convex clusters. The hierarchical merging of clusters can be visualized as a hierarchical tree, also called a dendrogram, which can prove to be useful for understanding the structure of the data [96]. A different name for this algorithm is bottom-up (or agglomerative) clustering. Top-down (or divisive) clustering is also possible [94] but not regarded by this research.

DBSCAN

Density-based Spatial Clustering of Applications with Noise (DBSCAN) uses a density-based notion of clusters, designed to discover clusters of arbitrary shapes, is efficient for large spatial databases, can be applied in high dimensional feature spaces, and works with any distance function [17]. DBSCAN needs two parameters to work, Eps, a measure of distance in the feature space, and $MinPts$, the number of points to form a dense region. It works by labelling all the points of a dataset as core, border, or noise points. First, the algorithm labels all points as a core point when they have at least $MinPts$ within their Eps. Then, for every core point not assigned to a cluster, a new cluster is created for this point, and all density-connected points are added to that cluster. Then, for each cluster sequentially, points that are close to a cluster, but not labelled as a core point are added to the cluster but cannot be used to grow it. These points are called border points. All remaining points are labelled as noise. These points may lie close to clusters but are not part of the cluster as there are no core points within their Eps. While the time complexity of the algorithm may not strictly adhere to the $O(n \log n)$ as stated in the original paper, Schubert et al. [97] show that the time complexity is not completely unfeasible

and conclude that DBSCAN performs comparatively compared to newer methods, given the correct parameters are set.

HDBSCAN

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) is a clustering algorithm that extends DBSCAN by converting it into a hierarchical clustering algorithm [98]. It then extracts a flat clustering based on the stability of clusters. Unlike DBSCAN, it supports clusters of varying densities. HDBSCAN only needs one hyperparameter to be set, a classic smoothing factor m_{pts}. The HDBSCAN algorithm starts by calculating the mutual reachability distance between all possible sets of points. The mutual reachability distance between two points is based on the core distances of these two points, defined as $d_{core}(X_p)$ and is the distance from object X_p to its m_{pts}-nearest neighbour. The mutual reachability distance is defined as $d_{mreach}(X_p, X_q) = max\{d_{core}(X_p), d_{core}(X_q), d(X_p, X_q)\}$, or the maximum of the m_{pts}-nearest neighbour of X_p, X_q, and the Euclidian distance between X_p and X_q. Then, the algorithm builds a Minimum Spanning Tree (MST) connecting each data point to its nearest neighbour in the context of these new distances. The MST is then transformed into a hierarchical tree of connected components by progressively removing the longest edges from the MST. This splits the graph into increasingly smaller connected components. Then, HDBSCAN condenses the dendrogram and focuses on the most significant clusters, eliminating noise and branches too small to be meaningful. Finally, it calculates the stability of each cluster C_i by summing the difference between the density level at which object X_j is not part of cluster C_i and the minimum density level at which C_i exists. The algorithm then selects the most stable clusters from the hierarchy and regards points not belonging to a stable cluster as noise. One disadvantage of HDBSCAN is that it has a time complexity of $O(n^2)$. This is the result of a few subprocesses that have this time complexity, like distance calculations [99]. In their improved Accelerated HDBSCAN* implementation, McInnes and Healy solve this issue by replacing the processes with $O(n^2)$ time complexity and achieve and $O(n \log n)$ time complexity comparable to DBSCAN [99].

3.2.2. SUPERVISED MODELS

The supervised ML models used in this research are all decision tree based models. The advantage of decision tree based models is that they are easy to compute and capture non-linear relations in the data [100]. The major disadvantage is that their simplicity might lead to models not capturing the complete relations. This is where ensemble methods, both bagging and boosting, come into play.

DECISION TREE

A Decision Tree (DT) is a simple model that aligns well with humans' decision process by creating a tree-like structure where each node represents a decision point, each branch represents an outcome, and the final node in each branch has two leaves of which each leaf represent the decision of the model given the conditions through which it has passed in a branch. Over the

years, many different algorithms have been created, but generally, Classification and Regression Trees (CART) [100] is used, as it can handle both classification and regression problems. CART recursively partitions the dataset into subsets that are as homogeneous as possible for the target variable. The data is split at each node using criteria like Gini impurity or entropy (log-loss) for classification and mean squared error or half Poisson deviance for regression. The algorithm splits the data at the nodes until a stopping condition is met. This stopping condition can be the $MaxTreeDepth$, which is the maximum number of nodes from the root down to the furthest leaf node, or the $MinSamples$, which requires all leaf nodes to have at least x samples. DTs are generally easy to understand and visualize but are prone to overfitting for more complex datasets. Bagging is a technique that reduces the test set error by perturbing the training set repeatedly to create multiple predictors and combining their outputs by voting or averaging in, respectively, classification or regression contexts [101]. The DT algorithm is selected as part of this research as it has been reported to perform quite well on PES data [6] and is inherently explainable.

RANDOM FOREST

A Random Forest (RF) is a group of classifiers that consist of a collection of tree-structured classifiers, using the training set and a new independent random vector based on a distribution introducing randomness to the classifier construction similar to the bagging mechanism described Breiman introduced in [101], resulting in a diverse set of classifiers in the ensemble which casts unit votes for the most popular class for the given input [102]. Breiman also proposes that each classifier in the ensemble should use a set of k random features to increase the randomness within the ensemble further, reducing the bias and variance of the ensemble. The main hyperparameters for a RF are the $NoEstimators$, which corresponds to the number of decision trees the ensemble is built of, and $MaxFeatures$, which corresponds to the maximum number of random features that each of the classifiers in the ensemble will be trained on. We employ the RF algorithm as it has shown to perform well on PES data in previous studies [6, 32, 37, 41, 43, 77].

XGBOOST

XGBoost is presented in [50] and is an implementation of GBDT first introduced by Friedman [80], building forth on the idea of DT ensembles [101]. The core idea of GBDTs is to stochastically optimize the loss function of a model using gradient descent. To do so, it fits a DT and evaluates its errors. Based on the residual error, it produces another DT that decreases the residual error further. This procedure is repeated M times. In addition, GBDT has the parameter $LearningRate$, which implements regularization by shrinkage in the update rule of the gradient descent, controlling the degree of error reduction in each step. A lower $LearningRate$ value has empirically proven to present better results but relies on the number of iterations M. M should be as high as computationally feasible, and $LearningRate$ should be adjusted to minimize the Lack of Fit close to M. The XGBoost algorithm [50] optimizes the GBDT algorithm by making it more efficient, flexible and allowing a portable implementation. This is done by implementing justified weighted quantile sketch, which allows for approximations of

gradients instead of calculating them precisely, saving a lot of computational resources and, combined with the cache-aware block structure, allows for parallel computation of gradients. Additionally, the algorithm is improved to better handle sparse data, which is frequent in machine learning applications due to the one-hot-encoding technique in which categorical values are translated into binary features for each of the values of the feature [50]. XGBoost is proven to be an efficient and accurate model, for which the subsampling rate and the number of features selected during each split hyperparameters do not have to be adjusted as long as some randomization is used [103]. We employ the XGBoost algorithm as it has shown to perform well on PES data in previous studies [32, 37, 41, 43].

CATBOOST

CatBoost, for Catgorical Boosting, is introduced in Prokhorenkova et al. [104]. Like XGBoost in the previous paragraph, CatBoost relies on GBDTs. The main difference is in how CatBoost processes categorical features. Where XGBoost relies on an algorithm that improves its performance for sparse data, like one-hot-encoded categorical features, CatBoost uses Target Statistics (TS) to translate the categories into numerical values. It does so by introducing a sequence in artificial time through a random permutation of the training examples, which allows for ordering the TS. Thus, the name Ordered TS. This allows CatBoost to use all training data to learn the model without any target leakage. Additionally, CatBoost combines categorical feature combinations greedily to capture high-order dependencies, which are then converted to TS. One advantage of CatBoost is that the algorithm needs relatively view hyperparameters to be set and automatically estimates them, reducing the need for hyperparameter tuning. The *Iterations* hyperparameter is the only required hyperparameter and controls the number of boosting iterations the algorithm executes. CatBoost is proven to be the most efficient and accurate boosting model currently available, while only the learning rate and the depth of the trees' hyperparameters have to be adjusted to get the best results [103]. The literature review in chapter 2 uncovered a successful application of the CatBoost algorithm to predictive process analytics, where it was chosen over XGBoost for faster computational times [59]. Although CatBoost has not yet been applied in PES we believe that CatBoost is a very promising algorithm due to its similarities with XGBoost, which has a proven track record in the field of PES.

3.3. VALIDATION OF MODELS

3.3.1. CROSS-VALIDATION

The hyperparameter tuning process of the previously described models is possibly prone to a statistical error. To counter this error, cross-validation [105] can be used. Cross-validation is a model validation technique that assesses how the results of a statistical analysis, like the application of any ML model, will generalize to an independent dataset. In the context of ML application, it is used to assess the quality of a fitted model and the stability of its parameters [106].

The statistical error may have been introduced due to the random split in the data, resulting in

a coincidental performance of the ML model for the defined hyperparameters, as a differently chosen split could result in a better-performing model. This statistical error can be reduced by introducing multiple "folds" in the dataset. A fold is one of the random partitions when the dataset is split into k equally sized subsamples. Each fold is used as a validation set once, while the other folds are used for training. The procedure is repeated once for every fold k as illustrated in Figure 3.2 for $k = 5$, leading to the name k-fold cross-validation. Generally, we can say that there is no statistical error, and we can thus be sure of the model quality, if the scores are similar for each of the k folds.

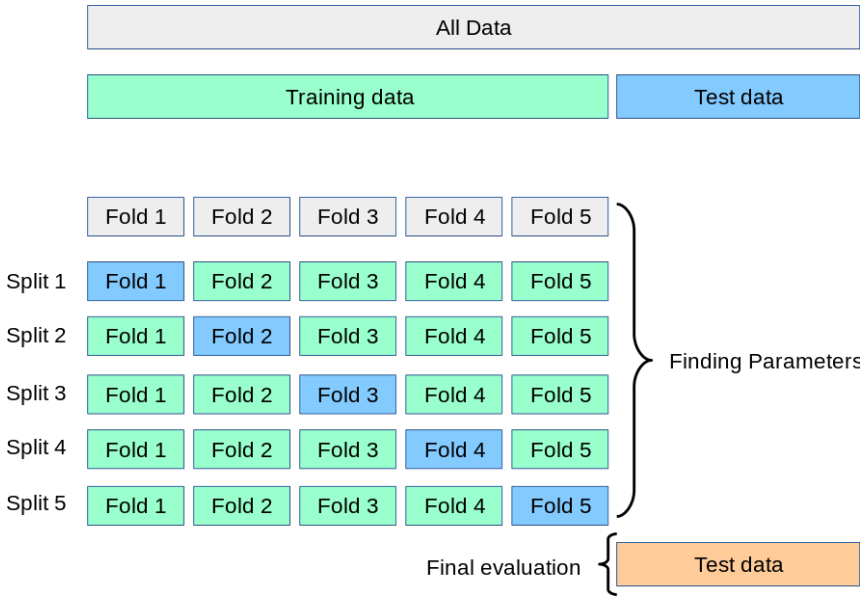


Figure 3.2: The k-fold cross-validation procedure, adapted from Scikit-learn[1].

3.3.2. METRICS OF VALIDATION FOR CLUSTERING MODELS

Where the supervised model evaluation metrics can all rely on the ground truth, say the given label of a test instance, unsupervised models do not have this access to a ground truth. Therefore, evaluation metrics for unsupervised models will generally rely on the distance between data points within clusters and distances between clusters [107–109].

CALINSKI–HARABASZ INDEX

The Calinski–Harabasz Index (CHI) score is introduced in [108] in 1974 by the eponymous authors and uses the dataset and clustering results to assess the clustering quality. It is defined as the ratio of the between-cluster separation (BCSS) to the within-cluster dispersion (WCSS), normalized by their degrees of freedom, see Equation 3.1. The BCSS is the weighted sum of squared Euclidean distances between each cluster mean and the overall data mean. The WCSS is the sum of squared Euclidean distances between the data points and their respective cluster means. The higher the CHI score, the better the clustering result.

[1] https://scikit-learn.org/stable/modules/cross_validation.html

$$CHI = \frac{\sum_{i=1}^{k} n_i \|\mathbf{c}_i - \mathbf{c}\|^2/(k-1)}{\sum_{i=1}^{k} \sum_{\mathbf{x}\in C_i} \|\mathbf{x} - \mathbf{c}_i\|^2/(n-k)} \tag{3.1}$$

Where k is the number of clusters $\{C_1, ..., C_k\}$, n is the number of points $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$, n_i the number of points in cluster C_i, $\mathbf{c}_i$ is the mean of cluster C_i, and $\mathbf{c}$ is the mean of the data.

The Calinski-Harabasz Index is used to validate cluster quality in, e.g., customer segmentation in finance [8] and retail [110].

DAVIES–BOULDIN INDEX

The Davies–Bouldin Index (DBI) is introduced in [109] in 1979 by the eponymous authors and is the average similarity measure of each cluster with its most similar cluster. Similarity, here, is defined as the ratio of within-cluster distances to between-cluster distances. The further apart and less dispersed the clusters are, the better the score. The minimum and optimal DBI is zero.

$$\begin{aligned} DBI &= \frac{1}{N} \sum_{i=1}^{N} \max_{j\neq i} \frac{S_i + S_j}{M_{i,j}}, \; where \\ S_i &= \left(\frac{1}{T_i} \sum_{j=1}^{T_i} \|X_j - A_j\|_p^q \right)^{\frac{1}{q}}, \; and \\ M_{i,j} &= \|A_i - A_j\|_p \end{aligned} \tag{3.2}$$

Where A_i is the centroid and T_i is the size of cluster C_i, q controls the root and the moment of the data and the mean, p controls the dimension and is 2 in the case of Euclidian distance. $M_{i,j}$ is a measure of separation between two clusters.

The DBI is used to validate cluster quality in customer segmentation in, e.g., finance [8] and retail [110].

MEAN SILHOUETTE SCORE

The Silhouette Score is introduced by Rousseeuw in [107] in 1987 and measures the similarity of an object to its own cluster ($a(i)$) compared to its dissimilarity to the closest other cluster ($b(i)$), as seen in Equation 3.3. It lies on $[-1, 1]$, and a higher value indicates a better match between the object and its cluster. Generally, a score of 0.7 is considered a good fit, 0.5 is a reasonable fit, and 0.25 is a weak fit between the object and its clusters. The Silhouette Score of all objects can be plotted to display the relative quality of the clusters, while the Mean Silhouette Score (MSC) evaluates clustering validity.

$$\begin{aligned} \forall_{k\in C} \forall_{i\in C_k}, \; s(i) &= \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \; where \\ a(i) &= \frac{1}{|C_k| - 1} \sum_{j\in C_k, i\neq j} d(i,j), \; and \\ b(i) &= \min_{m\neq k} \frac{1}{|C_m|} \sum_{j\in C_m} d(i,j) \end{aligned} \tag{3.3}$$

Where C is a set of clusters with k clusters, i and j are data points in cluster C_k, and $d(i, j)$ is an arbitrary distance function between points i and j.

The MSC is commonly used to validate cluster quality in specific applications of customer segmentation in finance [8] and retail [110].

3.3.3. METRICS OF VALIDATION FOR CLASSIFICATION MODELS

SENSITIVITY

In statistics and ML areas, Sensitivity, Recall or True Positive Rate (TPR) is regarded as the fraction of True Positive decisions by the actual positive cases [111]. Equation 3.4 shows the formula for Specificity.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.4}$$

SPECIFICITY

In statistics and ML areas, Specificity, or True Negative Rate (TNR), is regarded as the fraction of True Negative decisions by the actual negative cases [111]. Equation 3.5 shows the formula for Specificity.

$$Specificity = \frac{TN}{TN + FP} \tag{3.5}$$

FALSE POSITIVE RATE

The False Positive Rate (FPR) can be interpreted as the probability of a false alarm and is called a type I error in statistics [111]. The FPR is the ratio of False Positives divided by the total negatives, i.e., the sum of False Positives and True Negatives, and equal to the 1 - TNR or specificity, see Equation 3.6.

$$False\,Positive\,Rate = \frac{FP}{TN + FP} \tag{3.6}$$

RECEIVER OPERATING CHARACTERISTIC

The Receiver Operating Characteristic (ROC) is generally plotted as a curve between the TPR and the FPR, sometimes called the sensitivity vs (1 - specificity) plot. By comparing the TPR and the FPR under different threshold parameters in a binary classification, a plot can be generated [111]. The plot informs about the tradeoff between TPR and FPR under the different threshold parameters and compares them with a random coin flip, which is sometimes visualized as a 45 deg line through the plot. For any threshold value between 0 and 1, when a point of the curve lies under the line, the model performs worse than a random guess. All points above the curve are considered better predictions than a random guess.

AREA UNDER THE ROC CURVE

As a comparison between ROC curves is generally hard to do and does not allow for usage in automated optimizations, the AUC ROC can be used to compare different ROC curves. The AUC ROC loses the information about the thresholds which compose the ROC curve but is generally enough to find the better performing model of the two [111].

ACCURACY

Accuracy is the fraction of the study population that is correctly classified, which is the sum of the sensitivity and the specificity [112]. Equation 3.7 shows the formula for Accuracy.

$$\begin{aligned} Accuracy &= Sensitivity + Specificity \\ &= \frac{TP}{TP+FN} + \frac{TN}{TN+FP} \\ &= \frac{No.correctdecisions}{No.cases} \end{aligned} \tag{3.7}$$

PRECISION

Precision is the fraction of relevant instances in the set of retrieved cases [112]. When translated into the same jargon as the other metrics, this would be the ratio of True Positives amongst the Predictied Positives (True and False Positives), seen in Equation 3.8.

$$Precsion = \frac{TP}{TP+FP} \tag{3.8}$$

F-SCORE

The *F-score* is a family of metrics. The goal is to create a tradeoff between precision and recall [112]. The F_1-*score* is generally used in ML contexts and deems precision and recall equally important, thus creating a harmonic mean between the two. For example. the F_2-*score* can be used when recall is twice as important as precision. The general formula for F_β-*score* is seen in Equation 3.9. For the F_1-*score*, β would be set to 1, which simplifies to Equation 3.10.

$$F_\beta = (1+\beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall} \tag{3.9}$$

$$\begin{aligned} F_1 &= 2 \cdot \frac{precision \cdot recall}{precision + recall} \\ &= 2 \cdot \frac{\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}}{\frac{TP}{TP+FP} + \frac{TP}{TP+FN}} \\ &= 2 \cdot \frac{TP}{2TP+FP+FN} \end{aligned} \tag{3.10}$$

BRIER SCORE

The Brier score is a function that measures probabilistic predictions' accuracy [113]. It essentially calculates the mean squared error of the actual outcome $y \in \{0,1\}$ and the predicted probability estimate $p = Pr(y = 1)$, see Equation 3.11. This can theoretically be done for multiclass classifiers, however, the scikit-learn Python package[2] only allows this score to be calculated for binary classifiers.

$$Brier\ score = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} (y_i - p_i)^2 \tag{3.11}$$

3.4. LIME

Local Interpretable Model-agnostic Explanations (LIME) is an interpretability method capable of explaining any classifier or regressor by learning an interpretable model around the prediction that is locally faithful to the model [79]. LIME adheres to four important characteristics:

- **Interpretability**: LIME provides a qualitative understanding between the input variables and the response.
- **Local fidelity**: the explanation must correspond to how the model behaves near the predicted instance.
- **Model-agnostic**: LIME should be able to explain any model and achieves this by treating the original model as a black box.
- **Global perspective**: the model explanation is as important as an individual prediction. Therefore, LIME selects a few explanations to present to the user, such that they are representative of the model.

LIME is based on Equation 3.12. In this equation, $x \in \mathbb{R}^d$ is the original representation of an instance being explained, and $x' \in \{0,1\}^{d'}$ is the binary vector for the interpretable representation of x. G is a class of potentially interpretable models, like linear models or decision trees and can be presented visually to the user. g lies within the domain $\{0,1\}^{d'}$ and is a binary vector for the interpretable representation of the model. To punish models with high complexity, $\Omega(g)$ is included in the minimization objective and can be the number of non-zero weights in a linear model or the depth of the generated decision tree. The explained model is denoted as $f : \mathbb{R}^d \rightarrow \mathbb{R}$, and $f(x)$ is the probability that x belongs to a class in classification objectives. $\pi_x(z)$ is a proximity measure between an instance z to x and controls the locality around x. $\mathcal{L}(f, g, \pi_x)$ is a measure of the unfaithfulness of g in approximating f in the locality π_x. The formula can be used with any different explanation family G, fidelity function $\mathcal{L}$, and complexity measure Ω, which fulfils the model-agnostic characteristic.

[2] https://scikit-learn.org/stable/index.html

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{3.12}$$

LIME uses uniform random sampling of instances for local exploration around the binary vector for the interpretable representation x'. With this perturbed sample $z' \in \{0,1\}^{d'}$, the original representation $z \in \mathbb{R}^d$ is recovered and $f(z)$ is obtained and used as a label for the explanation model. With the dataset Z consisting of perturbed samples, Equation 3.12 is optimized to get the explanation $\xi(x)$. Objects near x get a higher weight than objects far away from x due to the proximity measure $\pi_x(z)$. This allows LIME to obtain an explanation for models that are too complex to explain globally.

The global perspective is tackled by constructing an explanation matrix W of $n \times d'$, where n is the number of explanations for a set of instances X. The number of instances to inspect is controlled by the user and is represented by a budget B that represents the number of explanations the user is willing to inspect to understand a model. The global importance I_j of each feature in the explanation space is given by $I_j = \sqrt{\sum_{i=1}^{n} W_{ij}}$ and in the case of text and tabular data $W_{ij} = |w_{g_{ij}}|$ can be regarded as the absolute value of the weight for feature j given explanation g_i. I_j should be larger for features that explain more different instances. The instances in budget B should be chosen according to a policy that minimizes redundancy in the components shown to the user. This is formalized in Equation 3.13, which picks a set of features based on their absolute value of feature weight for any possible set V from the set of instances X given budget B. This results in a set of instances and their explanations that cover the most informative explanations in the dataset based on their combined global feature importance.

$$Pick(W, I) = \underset{V, |V| \leq B}{\operatorname{argmax}} \sum_{j=1}^{d'} \mathbb{1}_{[\exists i \in V : W_{ij} > 0]} I_j \tag{3.13}$$

LIME has not yet been applied to the context of profiling in PES, but applications in the field of agriculture [72] and understanding uncertainty in neural networks [48].

3.5. SHAP

SHapley Additive exPlaination (SHAP), is an interpretability method that represents any explanation of a model's prediction as a model itself, called the explanation model. SHAP assigns an importance value for a particular prediction to each feature using an interpretable approximation of the original model [78]. SHAP adheres to three desirable properties:

- **Local accuracy**: the explanation model matches the output of the original model for the simplified input.
- **Missingness**: features missing in the original input to have no impact on, i.e., do not contribute to, the prediction.
- **Consistency**: if a model is altered so that the impact of a particular simplified input either

increases or remains unchanged, regardless of other inputs, the attribution assigned to that input should not be reduced.

SHAP calculates all Shapley values, which together form the output of the explainability model and is an additive feature attribution method that has an explanation model that is a linear function of binary variables. The explanation model g is defined in Equation 3.14

$$g(x') = \phi_0 + \sum_{i=0}^{M} \phi_i x'_i \tag{3.14}$$

In this equation, M is the number of features in the model $f(x)$, x' is the simplified input corresponding to the original input x. According to the local accuracy property, the original model $f(x) = g(x')$, and according to the missingness property, if $x'_i = 0 \Rightarrow \phi_i = 0$. ϕ_0 is the so-called base value, which is the mean output of the explainability model. Lundberg and Lee defined a theorem which states that there is only one possible explanation model g and define ϕ_i as Equation 3.15, which is a sum of the marginal contribution of all features when they are added to the subset of features weighted by the number of features, accounting for the different ways features can be added to the subset.

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!\left(M - |z'| - 1\right)!}{M!} \left[f_x(z') - f_x(z' \setminus i)\right] \tag{3.15}$$

Where $|z'|$ is the number of non-zero entries in the subset z' and $z' \subseteq x'$ represents all z' vectors where the non-zero entries are a subset of the non-zero entries in x'. The solution to this equation is a set of SHAP values, which is, in practice, approximated based on the model-type-specific kernels provided by the SHAP implementation [3].

SHAP has already been applied to the context of profiling in PES by Dossche et al. [77]. In addition, SHAP has been applied to prevent attrition among high school students [57] and employees [67].

[3] https://shap.readthedocs.io/en/latest/

4

EXPERIMENTAL SET-UP

4.1. EXPERIMENTAL SET-UP

The aim of this research is to implement an XAI application to predict LTU in the context of a Swiss PES. Additionally, this research aims to cluster customers into uniform groups with the goal of creating and assigning group-specific Labor Market Measure (AMM)s to the customers. The IT environment of this PES is provided by the State Secretariat for Economic Affairs (SECO) and is the same for all 26 cantons of Switzerland.

The SECO provides a financial administration system called ASAL in which all the data about the paid premiums is stored. The SECO also provides a customer administration system called AVAM in which all the data about the insured persons is stored. As both ASAL and AVAM are old systems, they depend on structured data storage, date back to the '70s, and are written in a proprietary programming language. Therefore, implementing business intelligence tools into these systems is nearly impossible. To foster the growing need for data analytics, a data intelligence system called LAMDA was developed. LAMDA runs on a MicroStrategy Workstation[1] environment and allows controllers to develop their own business intelligence tools by dragging and dropping components into boxes. Each canton has its own MicroStrategy environment and cannot access the data of any of the other cantons.

The implementation of an XAI module into the MicroStrategy environment comes with some complications:

- LAMDA, ASAL, and AVAM are not fully connected. Instead, they rely on an Extract, Transform, Load (ETL) server, which runs daily and ensures that updates to the data in each system are transferred to the other systems exactly once a day. This also means that upon the inscription of a new client in AVAM, it takes until the next morning for that client to

[1] https://www.microstrategy.com/enterprise-analytics/workstation

appear in LAMDA and ASAL. This delay also holds in any other direction of data flow. The ETL is capable of running Python[2] up to version 3.12 and allows all pip[3] packages to be used.

- Integrating custom build ML models on the LAMDA platform is impossible, as MicroStrategy also provides AI as a service, which allows only for limited customizability and comes with very high monthly fees. This challenge can be solved by integrating the predictive models in the ETL. This solution implies storing the output of the clustering and predictive models in a database so we can access it from the LAMDA platform.

To have a flexible and future-proof design, this research will incorporate a module design similar to [57, 67] and use separate modules for the different responsibilities of the various components. The implementation consists of four main components, visualized in Figure 4.1:

- **Mapping**: a JSON file[4] containing all the data transformations done in the Preprocessing step, allowing consistent feature encoding throughout the code. The end-users of the system, controllers, can easily change the mapping without any programming experience;
- **Preprocessing**: a pipeline responsible for cleaning and preparing the data;
- **Clustering**: a class containing the logic to calculate the clusters using an unsupervised learning model;
- **Predicting**: a class containing the logic for calculating the model output and explainability.

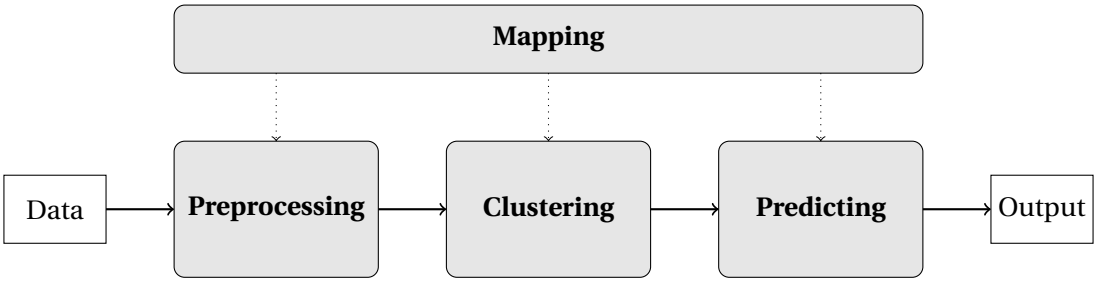


Figure 4.1: The modular design used in this study.

Both the Predicting and Clustering modules save their calculated models to a file. They can either be set to read their model from a file or retrained to allow for both the continuity in results and retraining. The explainability in the predicting module is tackled by a SHAP implementation, as it is model agnostic and thus allows for flexibility in model choice. It handles its outputs and explanations in a standardised manner, as opposed to LIME, which outputs different rules for each local prediction, increasing the complexity of storing the output in a structured database. All modules can be used independently and shared functions have been programmed module agnostic and are found in a utils file.

[2] https://www.python.org/
[3] https://pypi.org/
[4] See Appendix A

The technical implementation of the modules previously presented is a result of the non-technical experimental framework of this study, presented in Figure 4.2. The framework consists of three main phases and generally follows the CRISP-ML framework presented in section 3.1. The *Data Collection and Preprocessing* phase in the experimental framework maps to the *Data comprehension, cleansing, enhancement* phase of the CRISP-ML framework. The *Models Implementation* and *Models Validation* phases in the experimental framework map to the *Model building and evaluation* phase of the CRISP-ML framework. The first phase of the CRISP-ML framework will not be discussed in this thesis, although it has been conducted. The results of this research will be presented in chapter 5 and map to the *Derive and present business insights* and *Deployment and reporting* phases of the CRISP-ML framework.

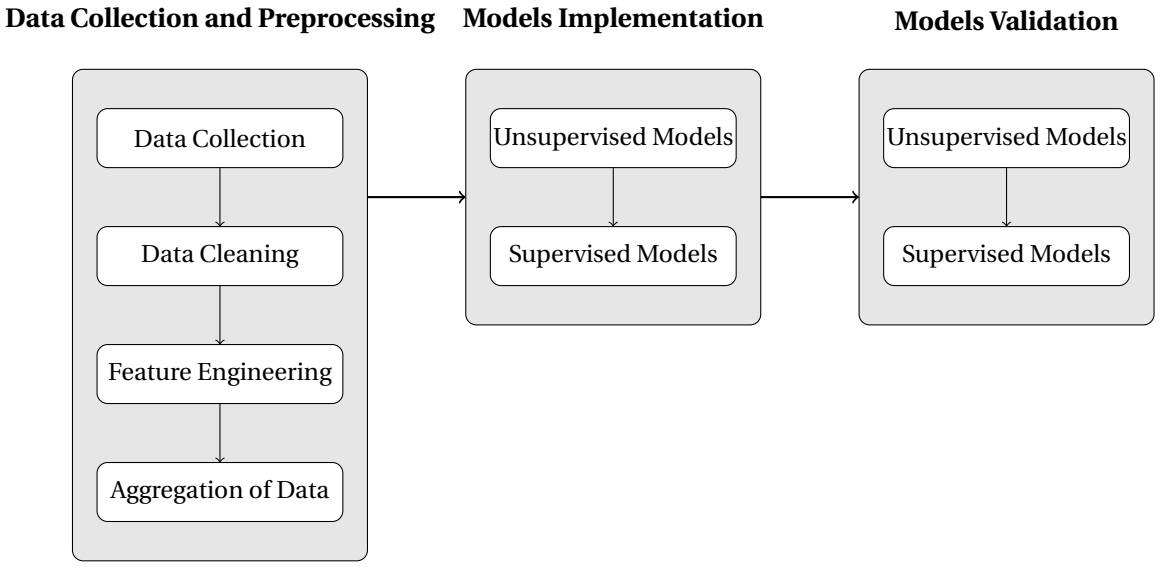


Figure 4.2: The experimental framework of this study.

Following the CRISP-ML framework, the predictive potential has been evaluated. We have explored solutions using regression, binary classification and multiclass classification, as the thesis's main problem can be solved through either of these methods. All three model types were experimented with using the default model hyperparameters. The regression solution attempted to predict the percentage of benefits taken and the days of benefits taken but for all supervised models discussed in chapter 3 and both problems, the explained variance (R^2) was negative, meaning that the models performed worse than always predicting the mean of the target value. The multiclass classification solution distributed the clients into one of six classes: the percentage of benefits taken between 0-19, 20-39, 40-59, 60-79, 80-99, or at 100. A note is that the classes were very imbalanced, with over half of the data within one of the 6 classes. This was tackled for the RF, XGBoost, and CatBoost models by setting class weights. Still, the F1-scores of all of the models remained far under 0.5, which is regarded as a bad-performing model. The results for a binary classification were more positive, with some of the models' F1-score reaching over 0.5. As the PES in question wants to distinguish people with potential problems to return to the labour market from those who can support themselves, the prob-

lem is fit for a binary classification model. Therefore, the rest of this thesis focuses on a binary classification problem.

4.2. DATA COLLECTION AND PREPROCESSING

4.2.1. DATA COLLECTION

This study uses SQL queries to collect data directly from the structured SECO databases. As a data quality improvement program started in 2021, this project uses the data from one of the Swiss Cantons from 2022 until 2024. The data is consolidated from two main sources: the database with the *financial administration of the paid premiums* (ASAL), the database with all the *information about the Job Seekers (STES)s* (AVAM), including tables containing AMMs, and sanctions. This data is then joined on the common factor, the STES-ID, to arrive at a data frame containing 56 dimensions, generally of a categorical nature, containing more than 85,000 rows. These rows contain many duplicates due to the multiple joins containing one to many relationships. In addition to duplicates resulting from joins, the dataset also contains duplications due to multiple values for a feature being applicable to a client. Each value is stored as a unique row, while the rest of the data is duplicated.

4.2.2. DATA CLEANING

First, the research goal is to assign a relevant measure as early in the unemployment stage of a client as possible. Therefore, features that only became known later in the unemployment process are dropped, of which many came from the Sanktionen and AMM databases, both representing measures that can be assigned to an unemployed during their process at a Regional Employment Centers (RAV). As the data comes from multiple scattered databases, some columns in the resulting data frame are exact duplicates, allowing the researcher to drop one of them. In addition, all columns containing dates were also dropped as the research goal required no information about time other than features already containing this information. All the features were dropped in agreement with the person responsible at the Office of Unemployment Insurance (AVA). At the end of the data cleaning process, 26 features were left in the dataset, including numerical financial data that can be used to engineer multiple predictor values. The initial SQL query was updated to reduce the data flow through the preprocessing module, speed up the process, and prevent any errors when column names are changed.

Second, the data frame contains incomplete information. For example, the age or education level of some clients is unknown. These clients are removed from the dataset, as imputing the values with either the mean or the mode might introduce errors or impossible combinations of values to the model. After completing this step, 78% of the original data is left.

Finally, all vowels with the ümlaut are replaced with their respective two-letter combination, i.e., ü becomes ue. This is necessary as the dataset's language is mainly German, and some algorithms are incompatible with special characters.

4.2.3. FEATURE ENGINEERING

One of the features that was causing duplicate rows due to multiple values being applicable to a client was the *mother tongue* of a client. As Switzerland is a multilingual country, it can be expected, and was found, that people have multiple mother tongues. More than 5000 clients are bilingual from the final dataset, some claiming to have four mother tongues. Therefore, a multilingual feature was created. In addition to this feature, a feature that holds information on whether someone speaks the local language, which can be either (Swiss-)German or French depending on the region, is engineered.

The features containing the branch the STES has worked in are stored differently for temporary work than for more permanent work. Both have many missing values that can be combined into one feature: the branch of the last employer. If there is a missing value in the permanent work feature, it is imputed with the value from the temporary work feature. The values contain a six-digit code; each digit has its own meaning, and there are hierarchies in the digits. In consultation with the data expert, the first two digits were found to provide enough details about the client's branch of work for this project. A 0 is imputed for clients with no value for the previous features, as it is not one of the branch codes.

Finally, the predictive feature standardizing the employment duration by calculating the paid percentage of total days assigned was engineered as in Switzerland, people get assigned benefits according to their contribution to the fund. The assigned benefits are generally one of the following: 0 days (extremely rare), 90 days, 200 days, 260 days, 400 days, 520 days, and 640 days. Given the large difference between these categories, we cannot define a long duration as x days of unemployment. Standardising the benefits as a percentage of total allowance allows for a fair comparison between groups.

4.2.4. AGGREGATION OF DATA

Data is aggregated on multiple levels. First, some features have values with only a few occurrences in the data frame. A good example is the feature mother tongue, which has 28 possible values, each representing less than 2% of the total data. These values are combined into a class called "other" according to the mapping. Second, the system codes are translated to meaningful strings, utilising the mapping of feature values. Some system codes are combined in this research to reduce the granularity of certain features. This was the case for the level of education and the residential status. A higher granularity may increase the required model complexity without improving the model performance or even lead to overfitting for groups with few data points. Multiple system codes with the same inherent meaning are combined to prevent this. For example, three system codes indicate that the highest level of completed education is primary education, and five system codes indicate that the client has a settlement permit. Third, numerical values are binned for the anonymization of the data. This happens, for instance, with age, which is binned into groups consisting of age groups of five years, starting at 16 all the way up to 65, the retirement age in Switzerland.

Finally, the remaining duplicate rows are aggregated according to an aggregation strategy. This strategy is created using domain knowledge and based on the content and cause of feature duplication. The strategy is, thus, feature-specific. Some duplications were caused by counsellors updating earlier mistakes or missing information in the system. The last value was chosen for these features as updating a record produces a new row beneath the old row in the data frame. The first value was taken for the feature 'mother tongue', as the first answer one would give to the question "What is your mother tongue?" is probably the preferred language and, therefore, regarded as the main mother tongue. For binary features, the max is taken.

4.3. MODELS IMPLEMENTATION

As much of the data in this project is of a categorical nature, dimensionality is one of the biggest issues for clustering. A subset of the available features is chosen for the clustering to reduce dimensionality. This subset is partially defined by running a predictive model on the data, finding the most important features to base the clustering on, and partly on experience from industry. All of the ordinal features are encoded using an ordinal encoder and the Gower distance is employed to calculate a matrix of pairwise distances for the clustering algorithms.

The Gower distance allows for the encoding of the ordinal data to retain the ordinal nature and combine this with the categorical nature of the other variables [114]. In contrast, most numerical clustering approaches use the more widespread Euclidian distance as the default distance metric but allow for computation with precomputed distance matrixes. As the pairwise distance matrix needs to be recalculated upon every new batch of clients, which is an expensive operation, a supervised machine learning model is trained on the input vectors and the resulting class of the unsupervised model to avoid recalculation.

All the hyperparameter tuning will be conducted using the python package Optuna[5] which uses a form of Bayesian optimization for hyperparameter tuning [115].

4.3.1. HIERARCHICAL CLUSTERING

This study employs the Scikit-learn implementation of Hierarchical Clustering [106]. As described in chapter 3, Hierarchical Clustering needs either the $LinkageDistanceThreshold$ or the $NumClusters$ as hyperparameters. We optimize the linkage distance threshold in the search space of (0, 10]. The search space follows a uniform distribution as the search is over a continuous range, and we have no knowledge of the best value for the hyperparameter.

Other hyperparameters that will be set are the:

- **Linkage**, which determines which distance to use between sets of observations. In this case, the Ward distance is chosen, as it minimises the merged clusters' variance and performs best for noisy data [106].
- **Number of clusters**, which must be set to None, as we're optimizing the linkage distance

[5] https://optuna.org/

threshold.

- **Metric**, which must be set to precomputed, as we're employing the Gower distance, which needs to be precomputed as it's not part of the Scikit-learn toolkit.

4.3.2. DBSCAN

This study employs the Scikit-learn implementation of DBSCAN [106]. As described in chapter 3, DBSCAN needs two hyperparameters to be initialised: Eps and $MinPts$. This study employs the Scikit-learn implementation of DBSCAN. The search space of Eps will be (0, 10] and follow a uniform distribution as the search is over a continuous range, and we have no knowledge of the best value for the hyperparameter, and the search space of $MinPts$ will follow a uniform discrete distribution [2, 20].

Other hyperparameters that will be set are the:

- **Metric**, which should be set to precomputed, as we're employing and precomputing the Gower distance, which is not part of the Scikit-learn toolkit.

4.3.3. HDBSCAN

As described in chapter 3, HDBSCAN needs only the $MinClusterSize$ hyperparameter to be initialised. However, as we employ the hdbscan[6] python package which uses the optimized HDBSCAN* algorithm presented by McInnes and Healy [99] compared to the Scikit-learn implementation which uses the algorithm as introduced by Campello et al. [98] to increase the computational efficiency, we can also set the hyperparameter $MinSamples$, which controls the clustering's conservativeness. The search space for the $MinSamples$ hyperparameter is set to [10, 100] and follows a uniform distribution as the search is over a continuous range, and we have no knowledge of the best value for the hyperparameter. The search space for $MinClusterSize$ is set to [50, 1000] and follows a uniform distribution as the search is over a continuous range, and we have no knowledge of the best value for the hyperparameter.

Other hyperparameters that will be set are the:

- **Metric**, which should be set to precomputed, as we're employing and precomputing the Gower distance, which is not natively supported by the hdbscan package.

4.3.4. DECISION TREE

The Scikit-learn implementation [106] of the CART algorithm that will be used does not support categorical features, in contrast to the original algorithm [100]. Therefore, the categorical features will have to be encoded. All ordinal features will be encoded using ordinal encoding, preserving the ordinal nature of the feature, and all other categorical features will be one-hot encoded.

As described in chapter 3, DT needs a stopping criteria to be set. This can either be $MaxTreeDepth$

[6] https://hdbscan.readthedocs.io/en/latest/index.html

or $MinSamples$. As the dataset is quite large, it can be expected that the tree will have to be rather deep to achieve good performance. Therefore, an optimization of the $MinSamples$ over the uniform discrete distribution of [1, 20] will be performed as the search is over a continuous range, and we have no knowledge of the best value for the hyperparameter. Furthermore, the $criterion$ will be optimized and can be either Gini impurity or entropy.

Other hyperparameters that will be set are the:

- **Class weight**, which will be set to $balanced$ to prevent the tree from being biased towards predicting the majority class by adjusting weights inversely proportional to class frequencies.
- **Max depth**, which controls the maximum depth of the DT and will be set to its default value of None to allow trees of any depth.

4.3.5. RANDOM FOREST

The Scikit-learn implementation of the RF algorithm that will be used does not support categorical features. Therefore, the categorical features will have to be encoded. All ordinal features will be encoded using ordinal encoding, preserving the ordinal nature of the feature, and all other categorical features will be one-hot encoded.

As a RF is essentially an ensemble of many DTs, the RF implementation also has access to the hyperparameters of the underlying DTs. Therefore, in addition to the hyperparameters mentioned in chapter 3, many more hyperparameters can be set.

The hyperparameters $MinSamplesSplit$ and $MinSamplesLeaf$ control, respectively, the number of samples required to split an internal node and the number of samples required to be at a leaf node. Both hyperparameters are used to control and prevent overfitting of the model and will be tested on the uniform discrete distribution of [1, 10].

Other hyperparameters that will be set are the:

- **NoEstimator**, which controls the number of trees and will be set to 100, its default value. Higher values than 100 generally do not massively improve the predictive performance.
- **MaxFeature**, which controls the number of features that will be considered for each of the individual trees in the ensemble. Generally, the square root of the total number of features performs best in classification problems [106]. Thus, this hyperparameter will be set to $sqrt$.
- **Criterion**, which will be set to the best-performing value from the previous DT hyperparameter tuning.
- **Class weight**, which will be set to $balanced$ to prevent the tree from being biased towards predicting the majority class by adjusting weights inversely proportional to class frequencies.

- **Max depth**, which controls the maximum depth of the DTs and will be set to its default value of None to allow trees of any depth.

4.3.6. XGBOOST

The implementation of the XGBoost algorithm[7] does not support categorical data. It is, however, optimized to handle one-hot encoded features, so all input features will be one-hot encoded.

As mentioned in chapter 3, the $LearningRate$ is the main hyperparameter of XGBoost and is used to prevent overfitting. It makes the model more robust by shrinking the weights on each step. The learning rate will be optimized on the interval [0.00001, 0.1] using a logarithmic scale.

Other hyperparameters that will be set are the:

- **NoIterations**, which controls the number of boosting rounds and will be set to 100, its default value. This will create a total of 100 boosted trees and create a model comparable complex as the RF.
- **Max depth**, which controls the maximum depth of the DTs and will be set to its default value of 6.

4.3.7. CATBOOST

Employing a CatBoost algorithm has two main advantages. The first is that it can work with categorical features out of the box and thus requires no further encoding. The second advantage is that it requires no hyperparameter tuning, as this is done automatically. The most important hyperparameter, the learning rate, is estimated based on the characteristics of the data. Therefore, we do not need to optimize this during the model selection stage. The $MaxDepth$ parameter is optimized on the interval [4, 10] using a uniform discrete distribution.

One hyperparameter that will be set to control the complexity of the algorithm is:

- **NoIterations** controls the number of boosting rounds and will be set to 100. This will create a total of 100 boosted trees and create a model comparable complex as the RF.

4.4. MODELS VALIDATION

4.4.1. UNSUPERVISED ML MODELS

The different clustering approaches are visually inspected using Multidimensional Scaling (MDS) to reduce the multiple dimensions to two dimensions and plot the data points and the clusters they are assigned to. In addition to the visual inspection, the clusters are evaluated based on the CHI, the DBI, and the MSC described in chapter 3. The computational comparisons are made using the pairwise distance matrix containing the Gower distances for all data points. The scores of the multiple validation metrics are compared over the multiple clustering meth-

[7] https://xgboost.readthedocs.io/en/stable/python/index.html

ods, as each of the metrics will likely lead to different hyperparameters for each of the models, leading to a different allocation of clusters. The similarity between the defined clusters is calculated for the different optimization metrics given each model and the different models given each optimisation metric. In addition, the individual clusters and their corresponding contents are explored and described for the best-performing clustering model.

4.4.2. SUPERVISED ML MODELS

The different supervised ML models are evaluated using the performance metrics presented in chapter 3. As the dataset is strongly imbalanced, the F1-score is the metric of most importance as it gives a fair representation of the model's performance despite class imbalance by considering both precision and recall. Cross-validation is employed to test the model's generalizability to an unknown dataset and detect overfitting. We employed five-fold stratified cross-validation on all models during hyperparameter tuning to allow a fair comparison between the different models. We use stratified cross-validation to ensure that both the positive and the negative classes are approximately represented according to the distribution of the complete dataset.

A sensitivity analysis is performed on the model output decision threshold. This threshold does not affect the model itself but when its output probabilities are transformed into positive or negative decisions. The threshold initially lies at 0.5, and thresholds on [0.40, 0.60] are tested and evaluated using the f1-score. The best-performing threshold is used for further experiments. Additionally, a sensitivity analysis is be conducted on the threshold of where to split the classes, i.e., the binary cutoff point for the percentage of benefits taken. We test thresholds between 10 and 90 with steps of 5, leading to 18 experiments evaluated on the aforementioned metrics. The binary cutoff point with the highest metric is used for further experiments.

Finally, this work adapts the calculated feature importance procedure evaluation from [57, 59, 67] as presented in subsection 2.4.2. This procedure generally consists of analysing the feature importance, or in this case, the SHAP values, for a set of selected instances and assessing the feature contributions with an expert to understand how accurate the explanation model is at distinguishing important features and feature interactions. This also allows for discovering currently uncovered relations between certain features at different aggregation levels [57, 67] or to evaluate the extent to which the evaluation method correctly identifies current organizational policies that might be present in the dataset [59, 67].

5

Results and Discussion

5.1. Exploratory Data Analysis

Understanding the input data for the ML models described in the previous chapters is important before examining their results. The dataset used in this research contained 84307 unique rows, many of which were duplicates. The original dataset contains 17103 unique STES-ids, and the many duplications are the result of multiple internal joins. The output of the preprocessing is a dataset with 17103 rows and 21 features. During preprocessing, the number of rows in the dataset is reduced by a factor of nearly five. Of the 21 features, one is the unique STES-id, fifteen are categorical features of the individual, one is the engineered predictive value of the percentage of benefits used, two are the numerical values used to create the predictive value, and the final two are created as part of the feature engineering and are both of a binary nature.

The dataset contains 29 individuals with a percentage of benefits used greater than 100%. These 29 individuals are the subjects of individual arrangements. They are removed from the dataset so as not to confuse the predictive models later on, resulting in a dataset of 17074 rows. The remaining distribution of the percentage of benefits used can be found in Figure 5.1. The graph follows a Pareto distribution, with a sharp spike at the maximum. The distribution is as expected by the experts of the PES, with the exception of the sharp spike at the maximum. This sharp spike indicates that there are many people who fully exhaust their benefits. As mentioned in chapter 1, one of the goals of this project in the PES is to detect these people and create strategies to avoid this behaviour and ensure a rapid reintegration in society.

As previously mentioned in chapter 4, this study predicts unemployment according to a binary classification objective. To get to a binary predictive value, the percentage needs to be binarized, which is done with a threshold. In consultation with the experts at the PES, this threshold has initially been set at 40%. As the mean allowance is 342 working days, 40% of that is 137 working days, or over half a year of unemployment, which is generally regarded as LTU

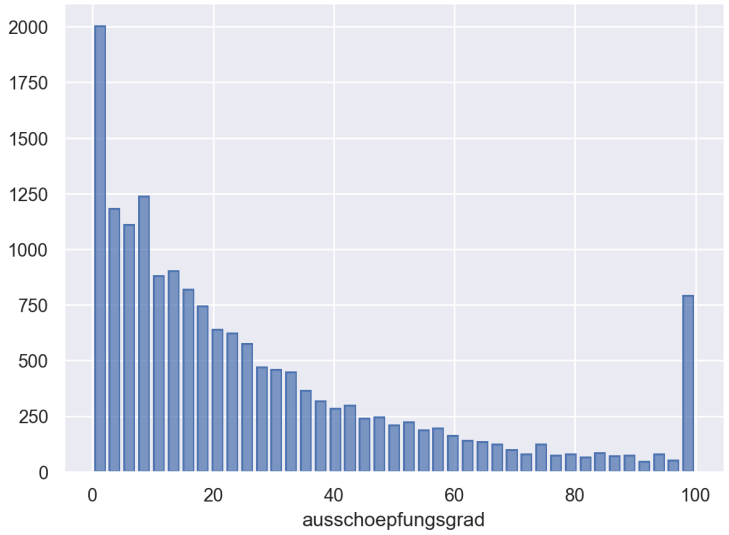


Figure 5.1: The distribution of the engineered predictive value.

across Europe[1]. The usage of a percentage allows for comparison across different degrees of allowances. The threshold will be optimized in a later stage, which is discussed in section 5.3. The initial distribution of the binary predictive value can be found in Figure 5.2. The Under 40%, or negative, class contains 12921 individuals, while the Over 40%, or positive, class contains a mere 4153 individuals. This means that 75.6% of individuals belong to the negative class, and only 24.4% of individuals belong to the positive class, indicating a strong imbalance in the dataset.

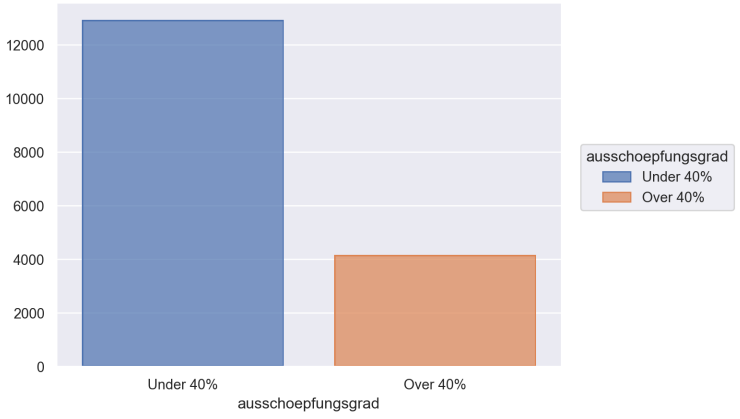


Figure 5.2: The binary distribution of the engineered predictive value with threshold 40%.

The assignment of the individuals into binary classes allows for an analysis of the distribution of the predictive value over the distributions of the input features. This is done for all input features except the last job and branche of the last employer, given the high cardinality of these categories. The analysis can be found in Figure 5.3 and Figure 5.4. We expect all of the input features to be distributed according to the 1 : 3 ratio, as also seen in the binary predictive value.

[1] See subsection 2.3.1

Any deviations from this might indicate a relationship between a feature and a lower or higher percentage of benefits used and will be reported below. As the dataset is in German, the translation of the feature name will be provided, followed by the name in the graph between brackets. Features of an increased interest due to a skewed distribution of the predictive value are:

- **Qualification (qualifikation_id)**: The learned value (gelernt) has slightly more individuals than expected for the Under 40% class. This might indicate that people who retrieved a professional qualification may find a job relatively easier than people who did not. The other values follow the distribution of the binary predictive value.
- **Professional Qualification (berufs_abschluss_id)**: The feature none (keiner) indicates that people did not receive a professional qualification. This value has a higher than expected number of individuals in the Over 40% class, which is in line with the findings for the feature Qualification.
- **Job Function (berufs_funktion_id)**: People with a management function (Kaderfunktion) are relatively less likely to use more than 40% of their benefits. In contrast, people with an apprentice function (Lehrling) are relatively more likely to use more than 40% of their benefits.
- **Mothertongue (sprache_id)**: Swiss German-speaking (CH-Deutsch) people are somewhat less likely to use more than 40% of their benefits in comparison with the average. People speaking French or Albanian (Franzosich or Albanisch) are relatively more likely to use more than 40% of their benefits.
- **Region (bezirksnummer)**: People in the regions Biel/Bienne, Interlaken-Oberhasli, Frutigen-Niedersimmental, and Obersimmental-Saanen are relatively more likely to use less than 40% of their benefits. People in the regions Emmental, and Jura Bernois are relatively more likely to use more than 40% of their benefits. This might indicate the existence of regional economic differences.
- **Offical Language (amt_sprache)**: People who speak the official language of their region (1) are less likely to use more than 40% of their benefits compared to people who do not speak it (0).
- **Level of Education (ausbildungs_niveau_id)**: People who finished their higher education, either professional or academic, are relatively less likely to use more than 40% of their benefits.
- **Age Category (ao_alter)**: People aged between 16 and 20 are slightly more likely to use more than 40% of their benefits. The age category 21 to 25 is the category for which people are most likely to use more than 40% of their benefits. The category after this, 26-30, is less likely to use more than 40% of their benefits. People aged between 51 and 55 are the least likely to use more than 40% of their benefits. After 61, so nearing the Swiss pension age of 65, people are more likely to use more than 40% of their benefits.

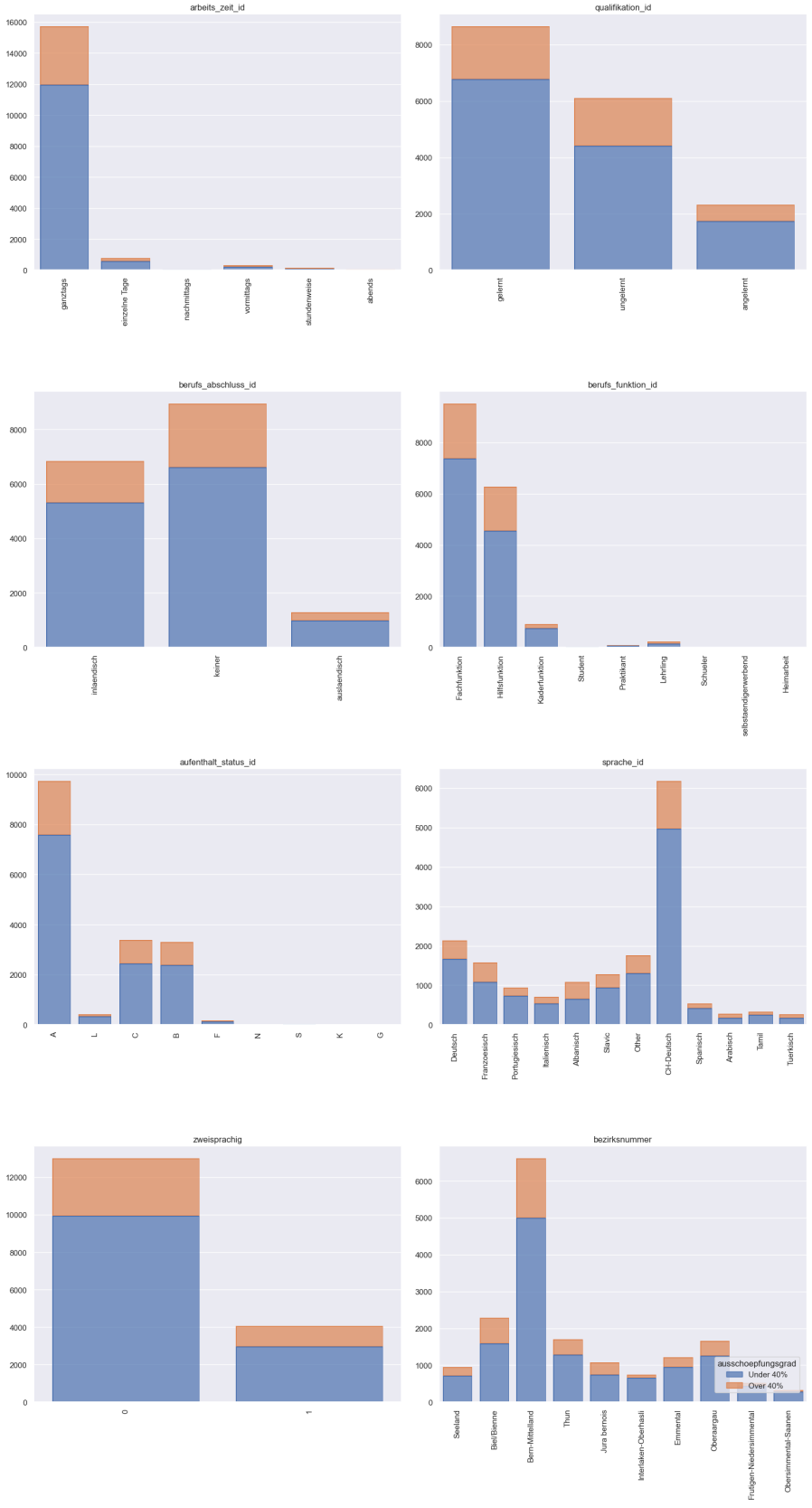


Figure 5.3: The distribution of input features over the binary predictive value.

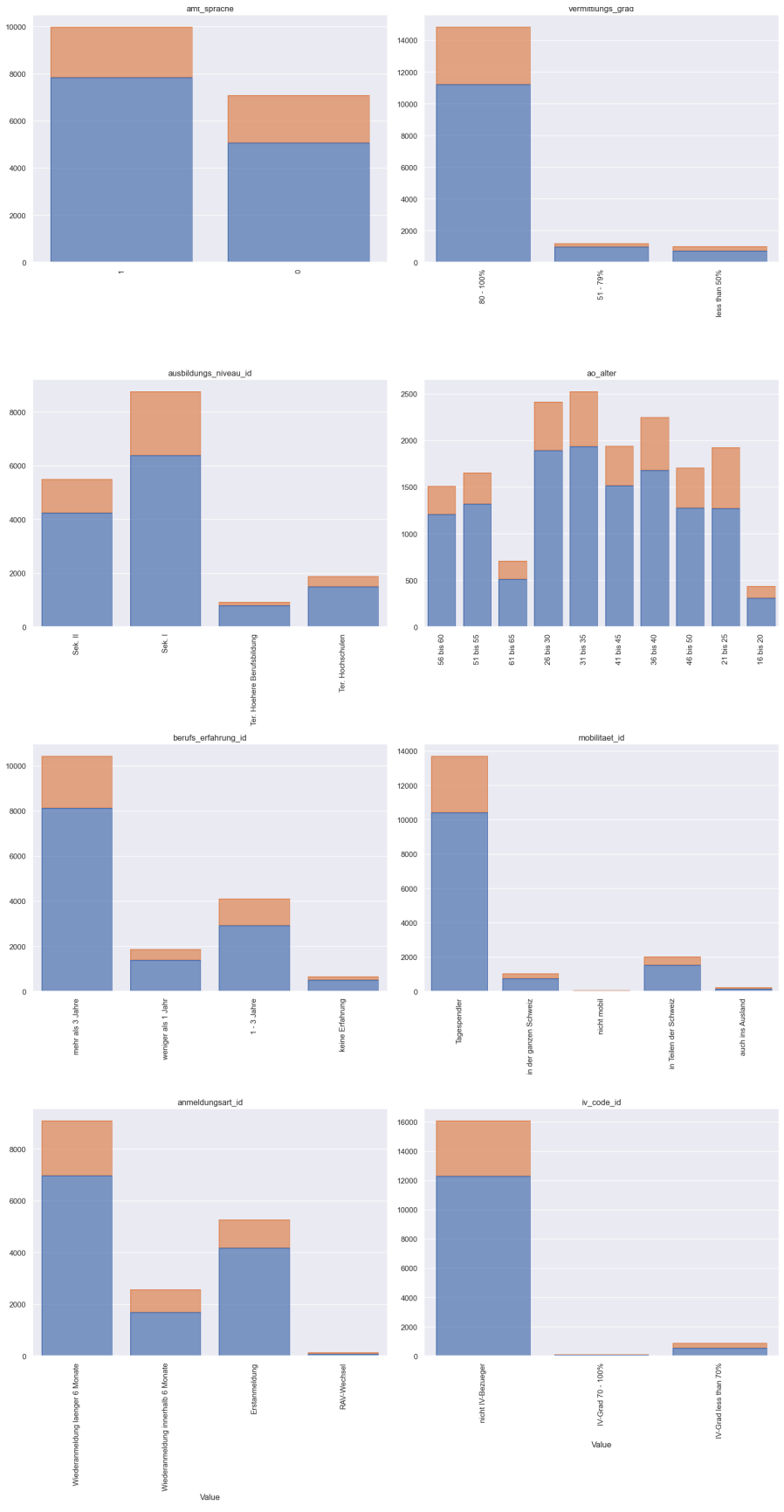


Figure 5.4: The distribution of input features over the binary predictive value (continued).

- **Experience (berufs_erfahrung_id)**: People with more than three years of experience (mehr als 3 Jahre) and people with no job experience (keine erfahrung) are slightly less likely to use more than 40% of their benefits, while people with 1-3 years of experience (1 - 3 Jahre) are more likely to use more than 40% of their benefits.
- **Registration Type (anmeldungsart_id)**: People who register for the first time (Erstanmeldung) at the PES are relatively less likely to use more than 40% of their benefits. People with subsequent registrations within half a year (Wiederanmelding innerhalb 6 Monate) are more likely to use more than 40% of their benefits.
- **Disability Rate (iv_code_id)**: People who have a disability rate fewer than 70% (IV-Grad less than 70%) are relatively more likely to use more than 40% of their benefits.

A chi-square goodness of fit test [116] has been performed over all possible feature combinations. The null hypothesis is that for each feature pair, the values of the first feature are independent of the values of the second feature. Assuming independence between the features, we would expect all values for the first feature to have an equal number of values for the second feature. The p-value of the chi-square test indicates if the value of the chi-square test statistic is large enough respective to the sample size and properties to reject the null hypothesis. To reject the null hypothesis, the p-value should be lower than the significance level, 5% in this case. If the p-value is lower than 5%, the null hypothesis is rejected, and the values of the first and second features are not independent, showing a correlation in the data. The correlation matrix in Figure 5.5 shows that most features are correlated. Therefore, we can select a subset of the features, which reduces the dimensionality of the clustering without excessive information loss.

The corresponding p-values of the chi-square tests are reported in Figure 5.5. It can be seen that most features are correlated with one another. Feature combinations without a statistical correlation are:

- Job Function (berufs_funktion_id) and Degree of Disability (iv_code_id),
- Bilingual (zweisprachig) and Employability Rate (vermittlungs_grad), and
- Experience (berufs_erfahrung_id) and Mobility (mobilitaet_id).

5.2. UNSUPERVISED MODELS RESULTS

As discussed in chapter 4, the Hierarchical Clustering, DBSCAN, and HDBSCAN unsupervised ML models are optimized with Optuna[2]. The results of the optimizations according to the hyperparameter search spaces also defined in chapter 4 are presented in Table 5.2. Additionally, the resulting clusters are visualized in two dimensions using MDS in Figure 5.6 and the distribution of the binary predictive value over the defined clusters is presented in Figure 5.7.

The results of the optimizations with respect to the three different metrics, CHI, DBI, and MSC,

[2]See footnote section 5

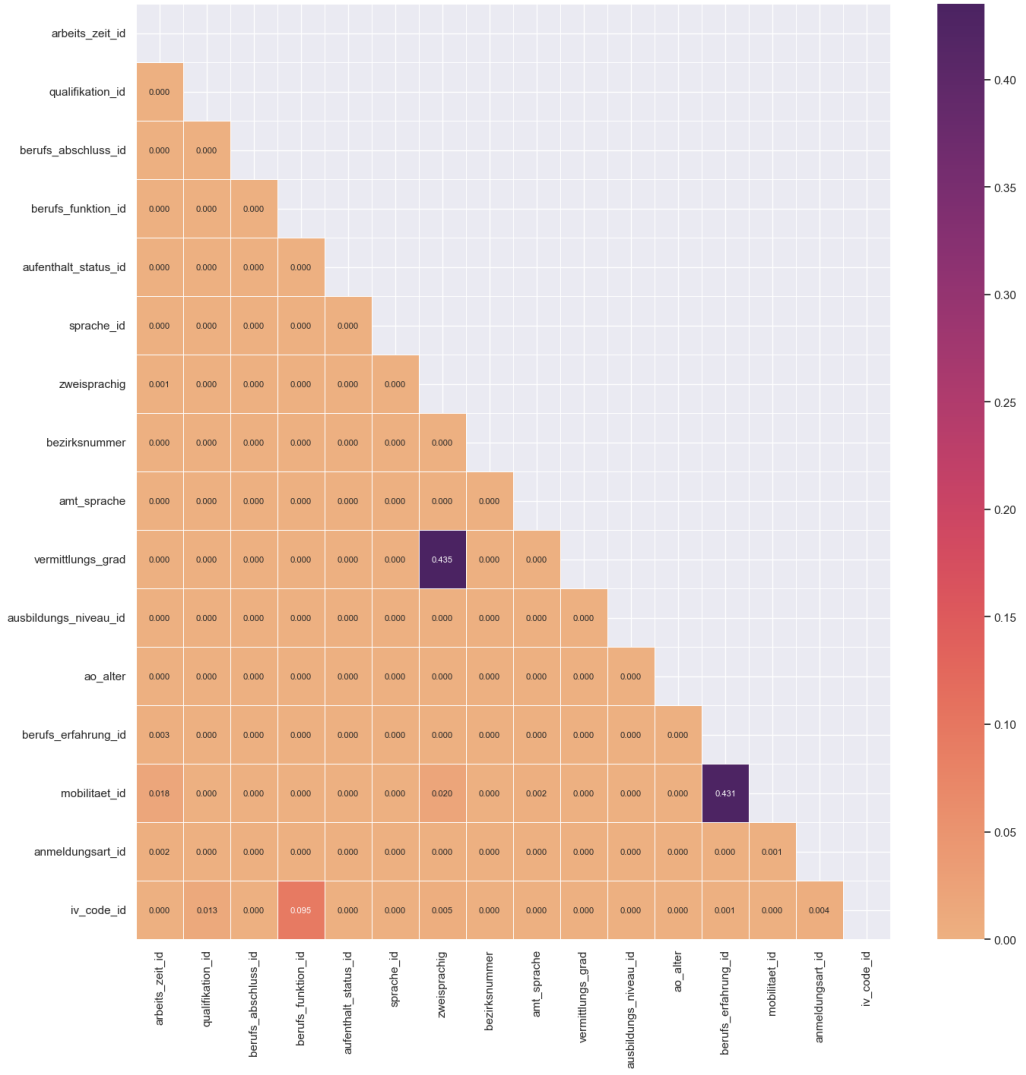


Figure 5.5: The correlation matrix of the input and target values.

are all in favour of the Hierarchical Clustering which has the highest score for CHI and MSC, and the lowest score for DBI. The DBSCAN came in at second place for all three metrics, while the HDBSCAN scored poorest on all three metrics. Although HDBSCAN is designed for applications on datasets with higher dimensions, it performed less than expected. This may have to do with the high correlations between many of the features.

Table 5.1: The results of the unsupervised learning optimization

Algorithm	**Metric**	**Score**	**Parameters**	**# Clusters**
Hierarchical Clustering	CHI	$\mathbf{5.76E+21}$	$LinkageDistanceThreshold$: 2.238	767
	DBI	$\mathbf{9.76E-06}$	$LinkageDistanceThreshold$: 8.923	223
	MSC	**0.992**	$LinkageDistanceThreshold$: 2.364	767
DBSCAN	CHI	$4.14E+4$	Eps : 6.448, $MinPts$: 3	68
	DBI	0.524	Eps : 1.140, $MinPts$: 14	263
	MSC	0.985	Eps : 1.464, $MinPts$: 14	263
HDBSCAN	CHI	$1.34E+4$	$MinClusterSize$: 541, $MinSamples$: 81	16
	DBI	0.598	$MinClusterSize$: 423, $MinSamples$: 24	19
	MSC	0.474	$MinClusterSize$: 763, $MinSamples$: 35	9

A visual inspection of the clustering results using Figure 5.6 shows that Hierarchical Clustering produces visually dispersed clusters, which is expected with the high number of clusters for all three metrics. The DBSCAN clustering approach produces many outliers for the DBI and MSC metrics while producing many clusters. The optimal solution for DBSCAN algorithm for the CHI metric and the solutions for the HDBSCAN algorithm produce well-ordered clusters. The optimal parameters for the HDBSCAN algorithm using the MSC metric only produce 9 clusters, which causes the whole bottom left part of the data to be considered as one cluster.

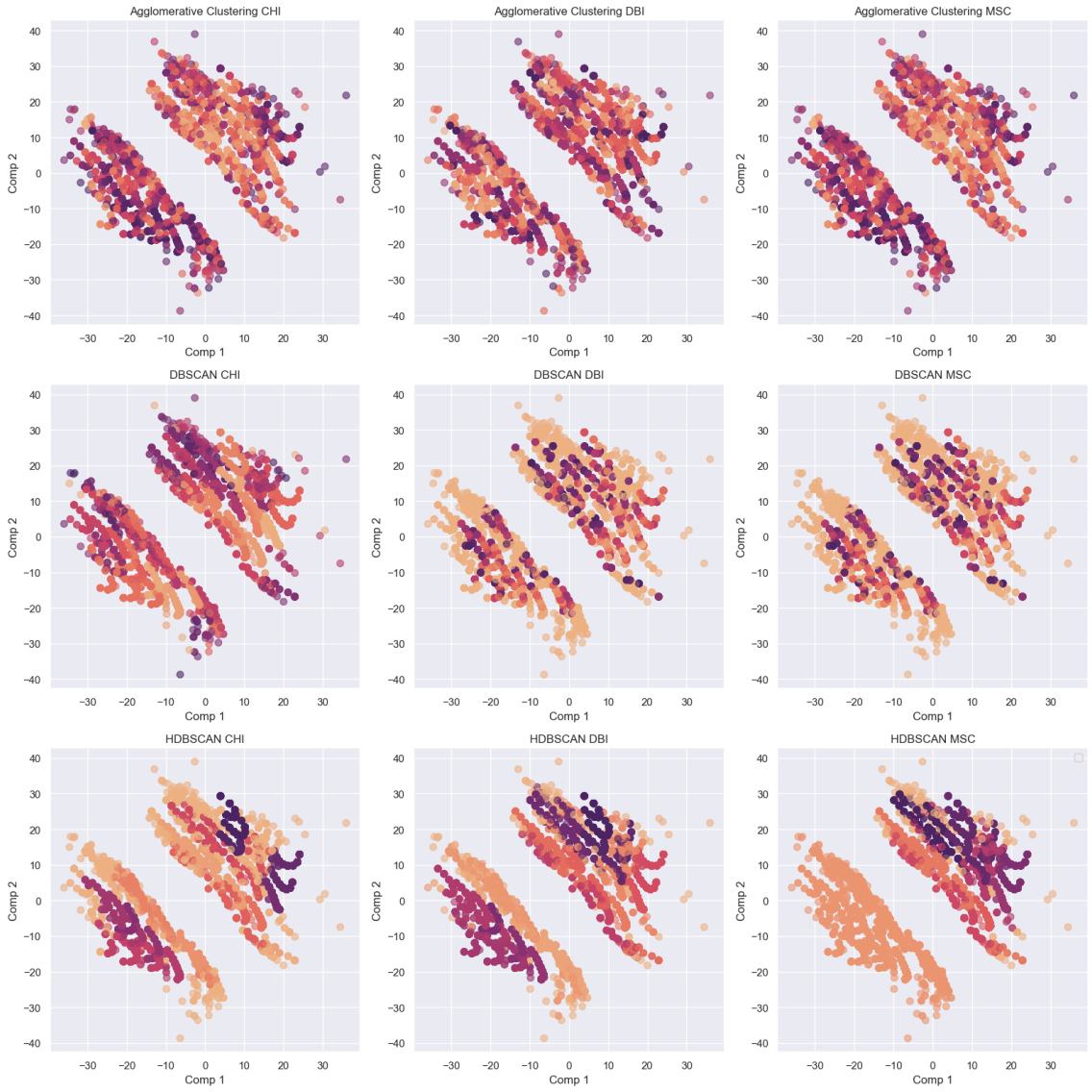


Figure 5.6: The clustering results of the different optimized unsupervised models.

Figure 5.7 shows the distribution of the binary predictive value over the clustering results of the different models, optimized towards different metrics. As the two top rows are hard to analyse due to the high amount of clusters, we will focus on the bottom row, presenting the HDBSCAN algorithm. The figure shows that some of the clusters, like, for instance, cluster 11 of the

HDBSCAN DBI or cluster 0 of the HDBSCAN MSC solutions, contain a higher proportion of instances of a benefits usage over 40%. Other clusters, like cluster 16 of the HDBSCAN CHI or cluster 18 of the HDBSCAN DBI solutions, contain a lower proportion of instances of benefits usage over 40%. This shows that the clustering can partially identify potentially problematic customer groups from those less likely to deplete their benefits.

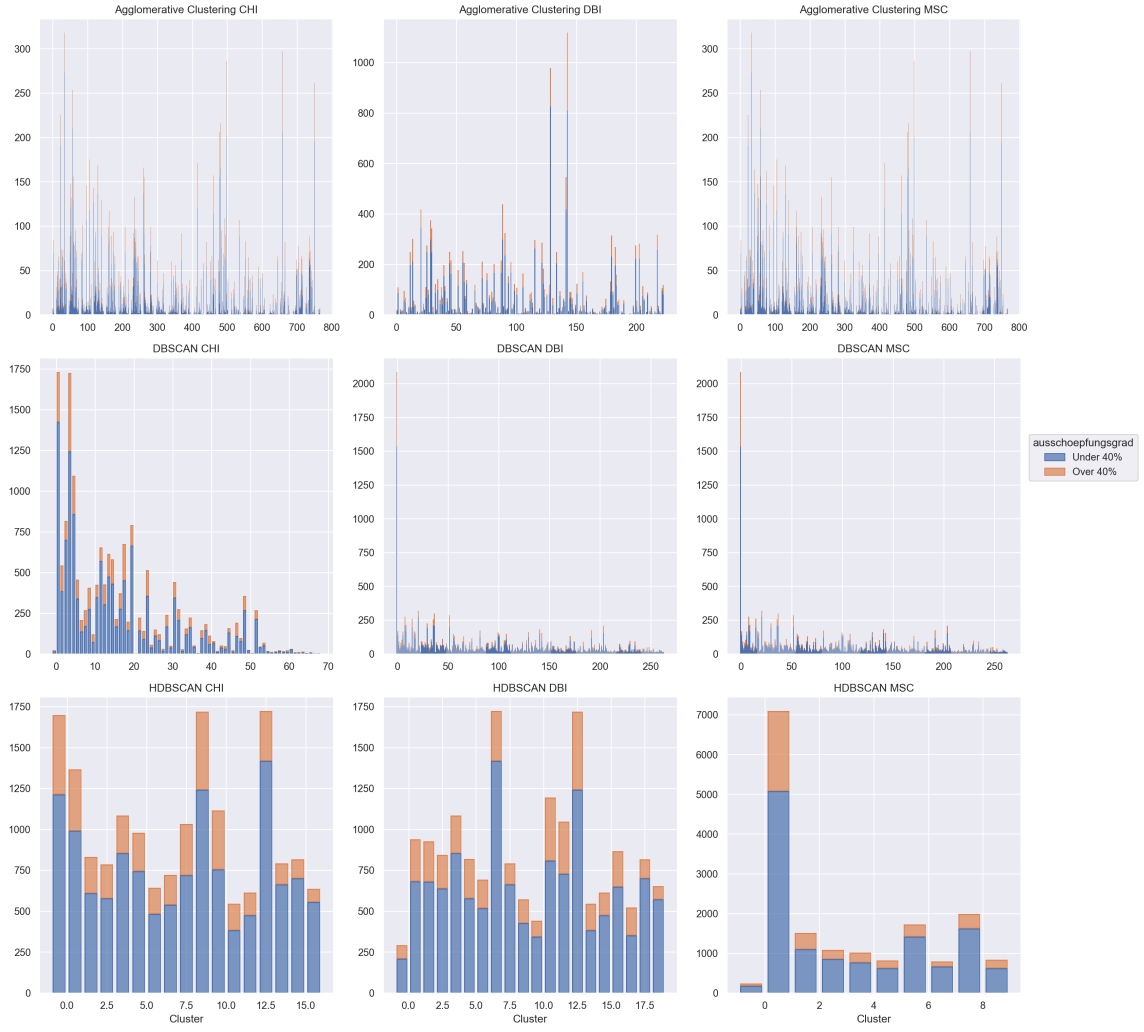


Figure 5.7: The distribution of the binary predictive value over the clustering results of the different optimized unsupervised models.

Remarkable is that the optimizations for the Hierarchical Clustering and DBSCAN presented different hyperparameters for all three metrics, but two of their results remained stable, producing respectively 767 and 263 clusters. The optimal parameters lie closely together for these cases, even though a different metric has been used. For the Hierarchical Clustering, the metrics CHI and MSC produce similar parameters and results, while for the DBSCAN the metrics DBI and MSC produce similar results. The optimization for the DBI metric for the Hierarchical Clustering produces a mere 223 clusters, as opposed to the 767 clusters that were result of the optimizations of the other metrics, and the optimization of the CHI metric resulted in 68 clusters in comparison with 263 clusters for the optimizations of the other metrics for the DBSCAN

algorithm. The optimizations for the HDBSCAN returned very different parameter sets for all three metrics.

However, as the PES is planning on using the results of the unsupervised ML models as an input of their initial response to the inscription of a recently unemployed, including presenting the counsellor with a set of possible metrics, it is practically undesirable to have over 20 clusters as the increasing complexity of the output cannot be grasped by humans. The complexity of the cluster output may be reduced by reducing the complexity of the input. In this case, the dimensionality of the clustering input could be reduced, as currently, the clustering is performed on all seventeen input features. Reducing the dimensionality of the clustering could potentially reduce the number of clusters defined by the different model configurations.

In consultation with the person responsible at the PES and with the variables causing a skewed distribution of the predictive value in mind as discussed in section 5.1, the features Registration Type (*anmeldungsart_id*), Age Category (*ao_alter*), Experience (*berufs_erfahrung_id*), Level of Education (*ausbildungs_niveau_id*), and Official Language (*amt_sprache*) are chosen as clustering features.

The results of the optimizations in a reduced dimensionality according to the hyperparameter search spaces also defined in chapter 4 are presented in Table 5.2. Additionally, the resulting clusters are visualized in two dimensions using MDS in Figure 5.8 and the distribution of the binary predictive value over the defined clusters is presented in Figure 5.9.

Looking strictly at the metrics presented in Table 5.2, the Hierarchical Clustering approach outperformed the two more recently developed density-based approaches in all three cases. The optimization of the density-based approaches resulted in an equal result for the CHI metric, but DBSCAN outperformed HDBSCAN on the other two metrics. The optimizations for the Hierarchical Clustering presented different hyperparameters for all three metrics, but their results remained stable, producing 767 clusters. The three optimizations for the DBSCAN model retrieved different hyperparameters and results, ranging from two to 625 clusters. The three different optimizations for the HDBSCAN model retrieved two completely identical solutions, of which the results align with the DBSCAN optimization using CHI as a metric. The HDBSCAN models resulted in either two or 33 clusters.

Table 5.2: The results of the unsupervised learning optimization

Algorithm	Metric	Score	Parameters	# Clusters
Hierarchical Clustering	CHI	$\mathbf{7.58E+21}$	$LinkageDistanceThreshold$: 2.555	767
	DBI	$\mathbf{9.80E-06}$	$LinkageDistanceThreshold$: 1.208	767
	MSC	**0.992**	$LinkageDistanceThreshold$: 0.098	767
DBSCAN	CHI	$4.14E+4$	Eps : 9.899, $MinPts$: 2	2
	DBI	0.909	Eps : 2.755, $MinPts$: 10	322
	MSC	0.985	Eps : 1.582, $MinPts$: 2	625
HDBSCAN	CHI	$4.14E+4$	$MinClusterSize$: 954, $MinSamples$: 98	2
	DBI	1.059	$MinClusterSize$: 202, $MinSamples$: 67	33
	MSC	0.580	$MinClusterSize$: 954, $MinSamples$: 98	2

Visually inspecting the clustering results presented in Figure 5.8 shows that Hierarchical Clustering produces visually dispersed clusters, which is expected with the high number of clusters for all three metrics. The DBSCAN clustering approach produces relatively fewer outliers for the DBI and MSC metrics compared to the higher dimension clustering presented in Figure 5.6. It does, however, still produce many clusters. The DBSCAN solution for the CHI metric and the HDBSCAN solutions for the CHI and MSC metrics produce the same solution, clearly splitting the data through the centre and producing two clusters. The optimal parameters for the HDBSCAN algorithm using the DBI metric produce 33 clusters with many outliers.

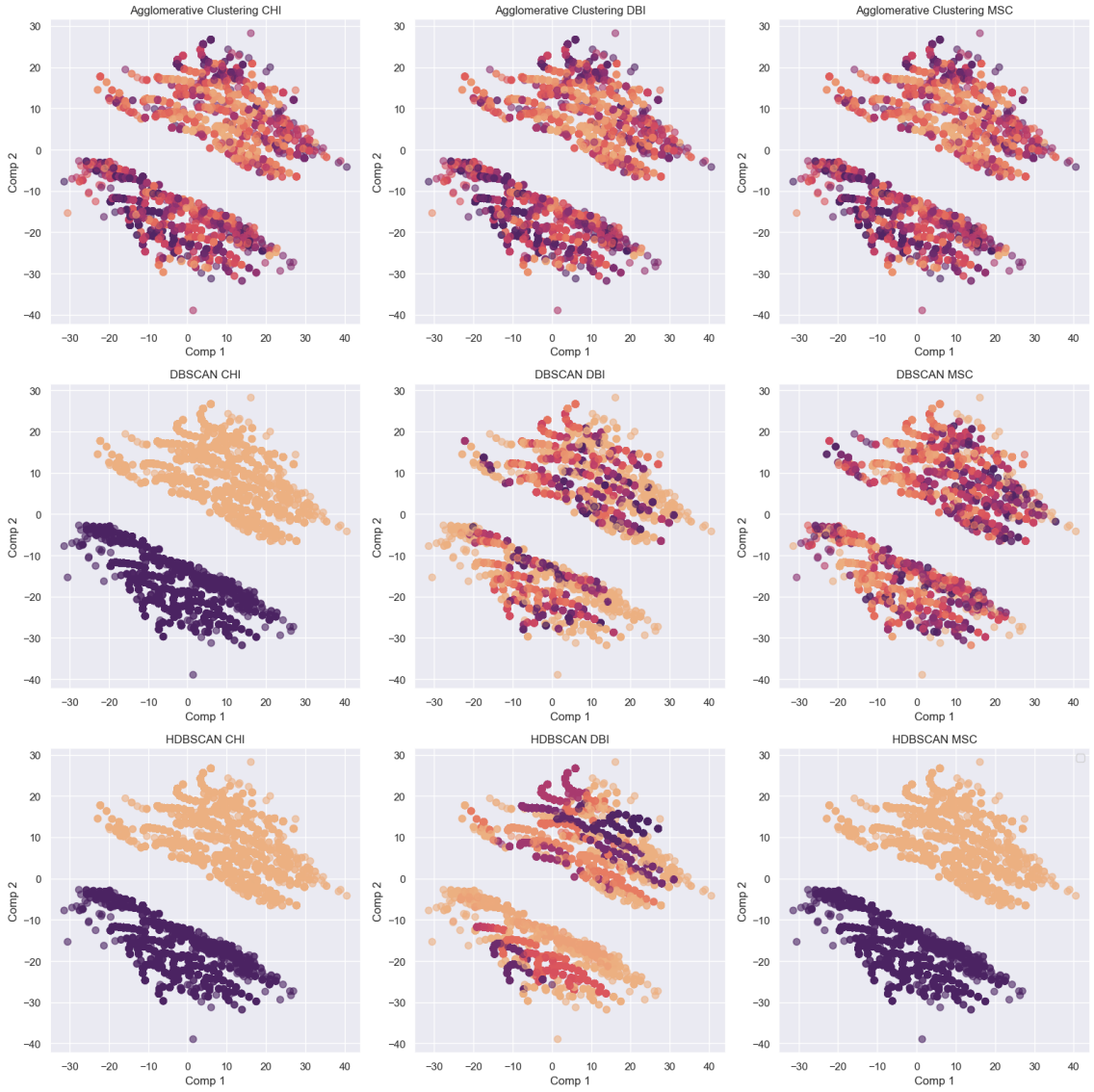


Figure 5.8: The clustering results of the different optimized unsupervised models.

Figure 5.9 shows the distribution of the binary predictive value over the clustering results of the different models, optimized towards different metrics. The instances that produce two clusters do not clearly distinguish between high or low benefits usage. Additionally, the HDBSCAN op-

timization of the DBI metric does not provide clusters with a distinction between high or low benefits usage as clear as the same optimization in the higher dimension solution presented in Figure 5.7. This hints that some information was lost, resulting in a less informative clustering solution.

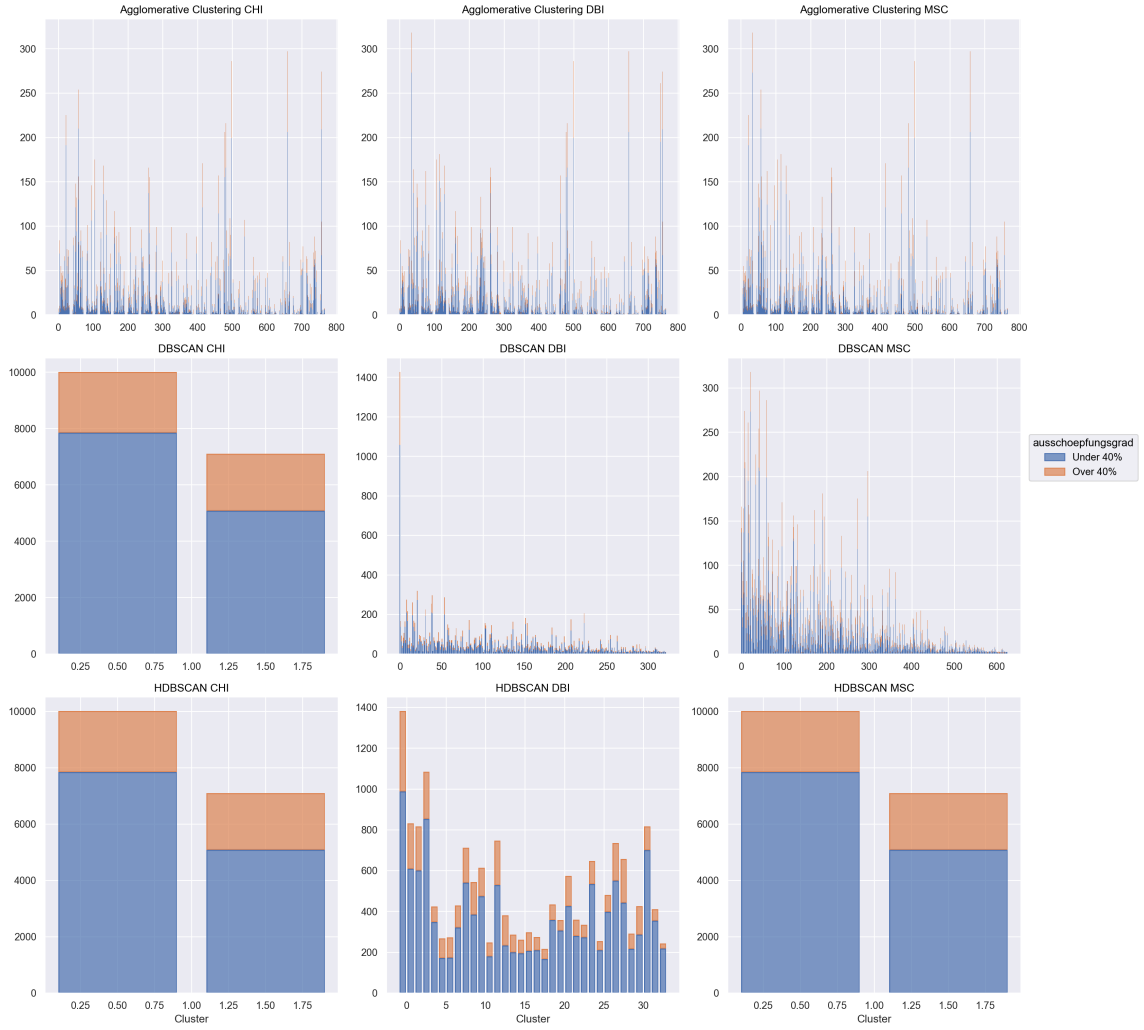


Figure 5.9: The distribution of the binary predictive value over the clustering results of the different optimized unsupervised models.

To conclude, the optimizations towards different metrics result in different parameters and clustering results over the three clustering models in both dimensionalities. Reducing the dimensionality did not necessarily lead to better clustering. The only model/metric combination for which the dimensionality reduction improves the clustering performance is the HDBSCAN MSC combination. The higher dimensionality counterpart produces an equal or better score for all other combinations. Therefore, we can conclude that there is no benefit in reducing the dimensions to just the features Registration Type (*anmeldungsart_id*), Age Category (*ao_alter*), Experience (*berufs_erfahrung_id*), Level of Education (*ausbildungs_niveau_id*), and Official Language (*amt_sprache*). Additionally, we found that solutions with many smaller clusters perform better in the scoring compared to a few larger clusters. This is not ideal for the

intended use case of the PES.

Moving forward, the original higher dimension HDBSCAN implementation using the parameters obtained from the optimization of the DBI is, in consultancy with the experts of the PES, chosen as the most informative model for policy creation. The complexity of the clustering solution is high enough to make informed decisions about specific groups but not so high that the groups are too specific, making human decision-making impossible. In addition, this solution shows clear differences in the predictive value between clusters of similar sizes while only disregarding a limited number of outliers.

The goal of the clustering is to distinguish different groups of people from one another, also referred to as profiling across literature. The cluster distributions across the feature values can be seen in Figure 5.10 and Figure 5.11. The several clustering profiles are summarised in Table 5.2.

Due to space limitations, only three clusters will be considered in detail, starting with cluster 5. Cluster 5 mainly contains unemployed Swiss nationals between 51 and 65 with more than three years of experience speaking at least two languages of either CH-German, German, or French, including the official local language of their region, without relevant professional education who have learned on the job, generally in specialist and auxiliary functions.

Cluster 7 mainly contains unemployed Swiss nationals between 31 and 65 with more than three years of experience speaking either CH-German, German, or French, which is the official local language of their region, with relevant Swiss higher education, generally in specialist and management functions, who have all reapplied to the PES within six months.

Cluster 16 mainly contains unemployed Swiss or foreign nationals between 16 and 65 with more than three years of experience speaking either either CH-German, German, or French, which is the official local language of their region, with relevant Swiss higher or professional education, generally in specialist functions, who have all reapplied to the PES within six months.

There are some slight differences between these clusters that can be relevant for finding a policy that works for the targeted individual, but this information is contained in a number rather than a textual description, allowing an easy mapping between target groups and policies. A difference between clusters 5 and clusters 7 and 16 is, for instance, that clusters 7 and 16 solely contain unemployed who are reapplying. One will reapply to the PES if they have lost their job, meaning that the people in these clusters are, according to some definitions seen across literature, LTU, requiring a different approach than people who apply to the PES for the first time. In addition, cluster 5 comprises people approaching the legal pension age. Finding a job is generally harder for them and requires a different approach than for someone who just finished their education.

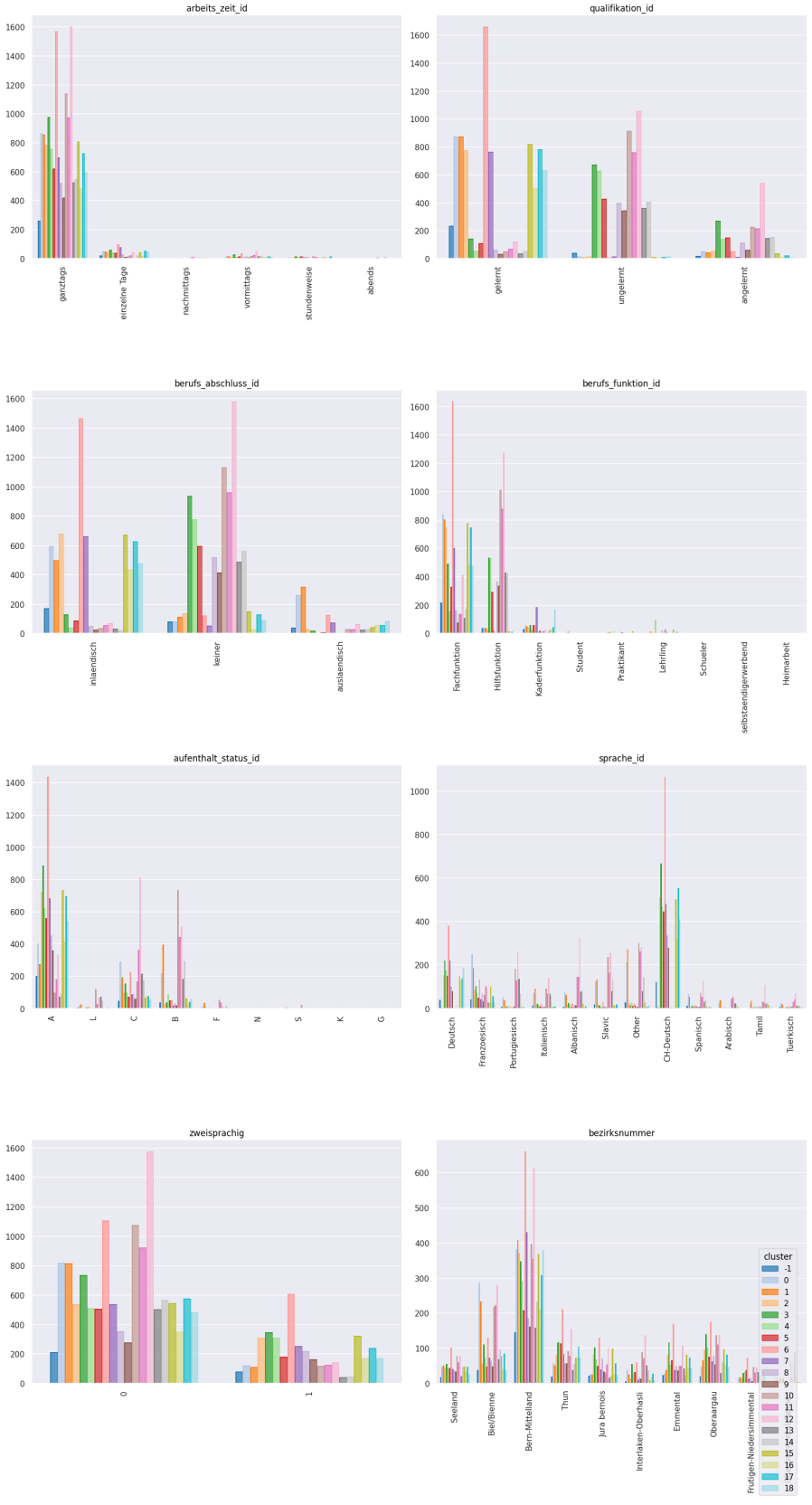


Figure 5.10: The distribution of input features over the clusters.

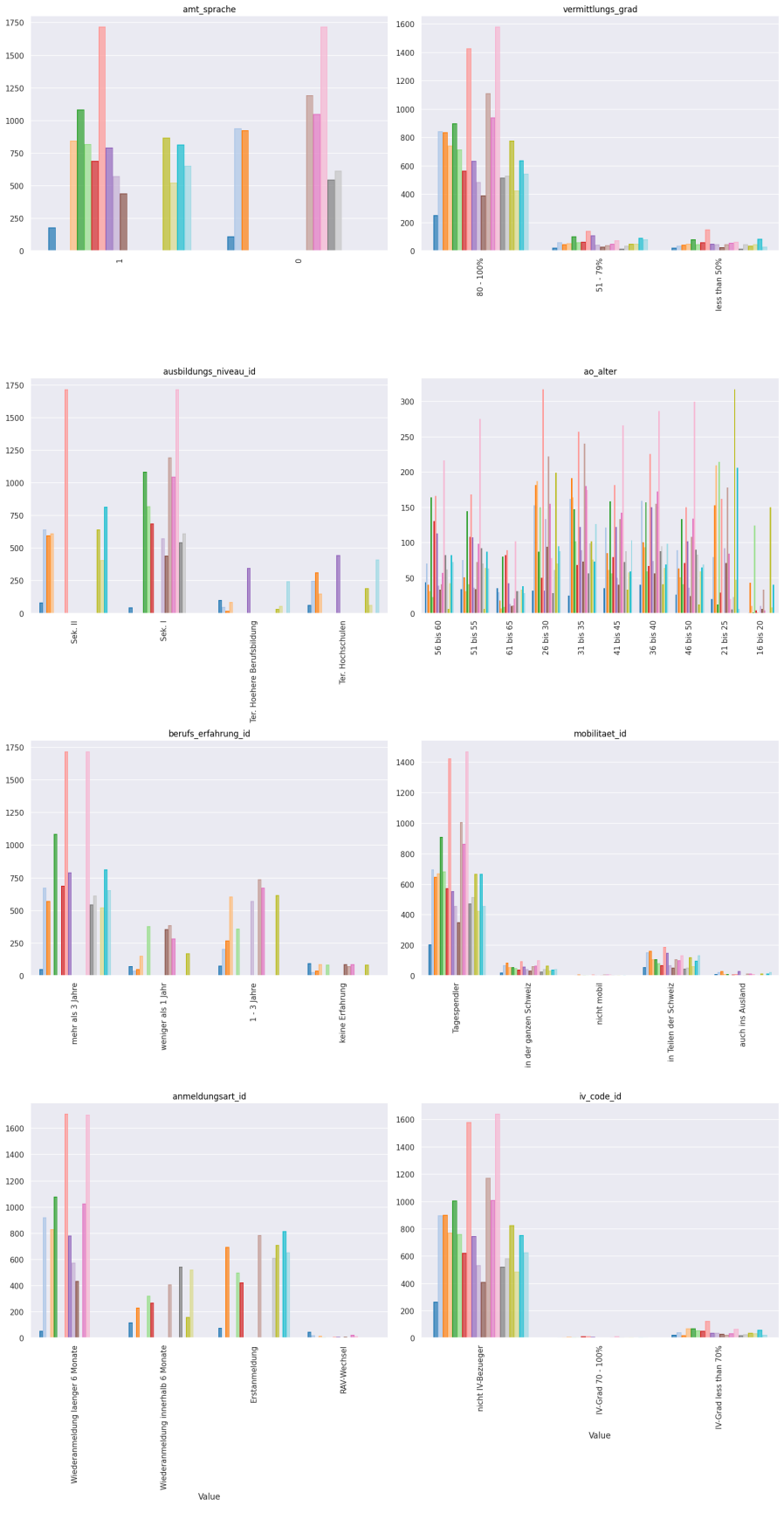


Figure 5.11: The distribution of input features over the clusters (continued).

Table 5.3: The profiles of the different clusters.

Cluster	qualifikation_id	berufs_abschluss_id	berufs_function_id	aufenthalt_status_id	sprache_id	zweisprachig
0	Relevant education	Swiss or foreign professional education	Specialists	A, B, C	French, other foreign languages	False
1	Relevant education	Swiss or foreign professional education	Specialists	A, B, C	French, other foreign language	False
2	Relevant education	Swiss professional education	Specialists	A, C	CH-German, German, French	True, False
3	No relevant education / learned on the job	No professional education	Specialists and auxiliaries	A, C	CH-German, German, French	True, False
4	No relevant education / learned on the job	No professional education	Auxiliaries and apprentices	A	CH-German, German, French	True, False
5	No relevant education / learned on the job	No professional education	Specialists and auxiliaries	A	CH-German, German, French	True
6	Relevant education	Swiss or foreign professional education	Specialists	A, C	CH-German, German, French	True, False
7	Relevant education	Swiss professional education	Specialists and managers	A	CH-German, German, French	False
8	No relevant education / learned on the job	No professional education	Specialists and auxiliaries	A	CH-German, German, French	False
9	No relevant education / learned on the job	No professional education	Auxiliaries	A	CH-German, German, French	False
10	No relevant education / learned on the job	No professional education	Auxiliaries	B, C	French, other foreign language	False
11	No relevant education / learned on the job	No professional education	Auxiliaries	B, C, L	French, other foreign language	False
12	No relevant education / learned on the job	No professional education	Specialists and auxiliaries	A, B, C, L	French, other foreign language	False
13	No relevant education / learned on the job	No professional education	Auxiliaries	B, C, L	French, other foreign language	False
14	No relevant education / learned on the job	No professional education	Auxiliaries	B, C, L	French, other foreign language	False
15	Relevant education	Swiss professional education	Specialists	A, B	CH-German, German, French	True, False
16	Relevant education	Swiss professional education	Specialists	A, B	CH-German, German, French	True, False
17	Relevant education	Swiss professional education	Specialists	A, B	CH-German, German, French	True, False
18	Relevant education	Swiss professional education	Specialists and managers	A, B	CH-German, German, French	True, False

Cluster	amt_sprache	ausbildungs_niveau_id	ao_alter	berufs_erfahrung_id	anmeldungsart_id
0	False	Sek. II, high. prof. educ., high. academic educ.	26 - 45	None, less than a year, 1 to 3 years, more than 3 years	Reapplication after 6 months
1	False	Sek. II, high. prof. educ., high. academic educ.	16 - 25	None, less than a year, 1 to 3 years, more than 3 years	Initial registration, reapplication within 6 months
2	True	Sek. II, high. prof. educ., high. academic educ.	21 - 35	None, less than a year, 1 to 3 years	Reapplication after 6 months
3	True	Sek. I	31 - 65	More than 3 years	Reapplication after 6 months
4	True	Sek. I	16 - 30	None, less than a year, 1 to 3 years	Initial registration, reapplication within 6 months
5	True	Sek. I	51 - 65	More than 3 years	Initial registration, reapplication within 6 months
6	True	Sek. II	21 - 35	More than 3 years	Reapplication after 6 months
7	True	high. prof. educ., high. academic educ.	31 - 65	More than 3 years	Reapplication after 6 months
8	True	Sek. I	21 - 60	1 to 3 years	Reapplication after 6 months
9	True	Sek. I	21 - 60	None, less than a year	Reapplication after 6 months
10	False	Sek. I	16 - 60	None, less than a year, 1 to 3 years	Initial registration, reapplication within 6 months
11	False	Sek. I	21 - 60	None, less than a year, 1 to 3 years	Reapplication after 6 months
12	False	Sek. I	26 - 65	More than 3 years	Reapplication after 6 months
13	False	Sek. I	26 - 65	More than 3 years	Reapplication within 6 months
14	False	Sek. I	26 - 65	More than 3 years	Initial registration
15	True	Sek. II, high. prof. educ., high. academic educ.	26 - 50	None, less than a year, 1 to 3 years	Initial registration, reapplication within 6 months
16	True	Sek. II, high. prof. educ., high. academic educ.	16 - 65	More than 3 years	Reapplication within 6 months
17	True	Sek. II	16 - 65	More than 3 years	Initial registration, reapplication within 6 months
18	True	high. prof. educ., high. academic educ.	26 - 65	More than 3 years	Initial registration, reapplication within 6 months

As the Gower distance matrix will have to be recalculated for every new inscription, and this is a rather expensive operation, a CatBoost supervised ML model will be trained on the clustering output. This model will be stored and queried upon new inscriptions. In addition, this also solves the issue that an unsupervised ML model may, over time, change its clustering output as new inscriptions may change the underlying densities. In this application, where a cluster corresponds to a clearly defined profile based on which measures are selected, this would be unacceptable and solving the issue would be labour-intensive. The clusters should, however, still be periodically reevaluated to prevent creep. It should be manually ensured that cluster-specific policies still are still valid for the calculated clusters.

The performance of the predictive model is evaluated using cross-validation over four-folds to ensure that the predictive model gets enough data points for each cluster to predict the cluster accurately. The cross-validation results can be found in Figure 5.12. In the four folds, respectively, 39, 41, 33, and 36 labels of the 4269 labelled individuals were wrongly predicted, resulting in an accumulated accuracy of 99.7%. All of these mispredictions were mistakes between the outlier cluster and any other cluster, showing that the predictive model has trouble with correctly predicting whether an instance belongs to any cluster or to the outliers. This can be caused by the sixteen-dimensional clustering and the difference between the handling of similar entries of the two different models. When the HDBSCAN model is not certain of a cluster, the individual is marked as an outlier, while the predictive model would return the most probable cluster with lower certainty. When an individual is more similar to individuals from any cluster than to the rest of the outliers, the CatBoost model will thus label this individual to be part of a cluster, while the difference, and thus distance to a dense area, may be large enough for the HDBSCAN model to assign this individual to the outlier class.

5.3. SUPERVISED MODELS

As discussed in chapter 4, the DT, RF, XGBoost and CatBoost supervised binary classification ML models are optimized with Optuna[3]. The results of the optimizations according to the hyperparameter search spaces also defined in chapter 4 are presented in Table 5.3. Additionally, the resulting confusion matrixes are visualized in Figure 5.13. To ensure the imbalance of the dataset is considered by the algorithms, the micro setting for the average parameter of the AUC ROC and F1-score metric is used.

Looking strictly at metrics, the CatBoost model scores best on AUC ROC, accuracy and the Brier score. The RF model scores best on the F1-score, followed by the DT, XGBoost, and CatBoost models respectively in second, third and fourth place. The XGBoost model has a slightly lower accuracy and Brier score than the CatBoost model and comes in at third place for AUC ROC. The performances of the optimized boosting algorithms are very similar, except for the AUC ROC score of the XGBoost model, which is significantly lower. The RF model scores second place for AUC ROC and third for accuracy and brier score. As expected, the only weak learning algorithm

[3]See footnote section 5

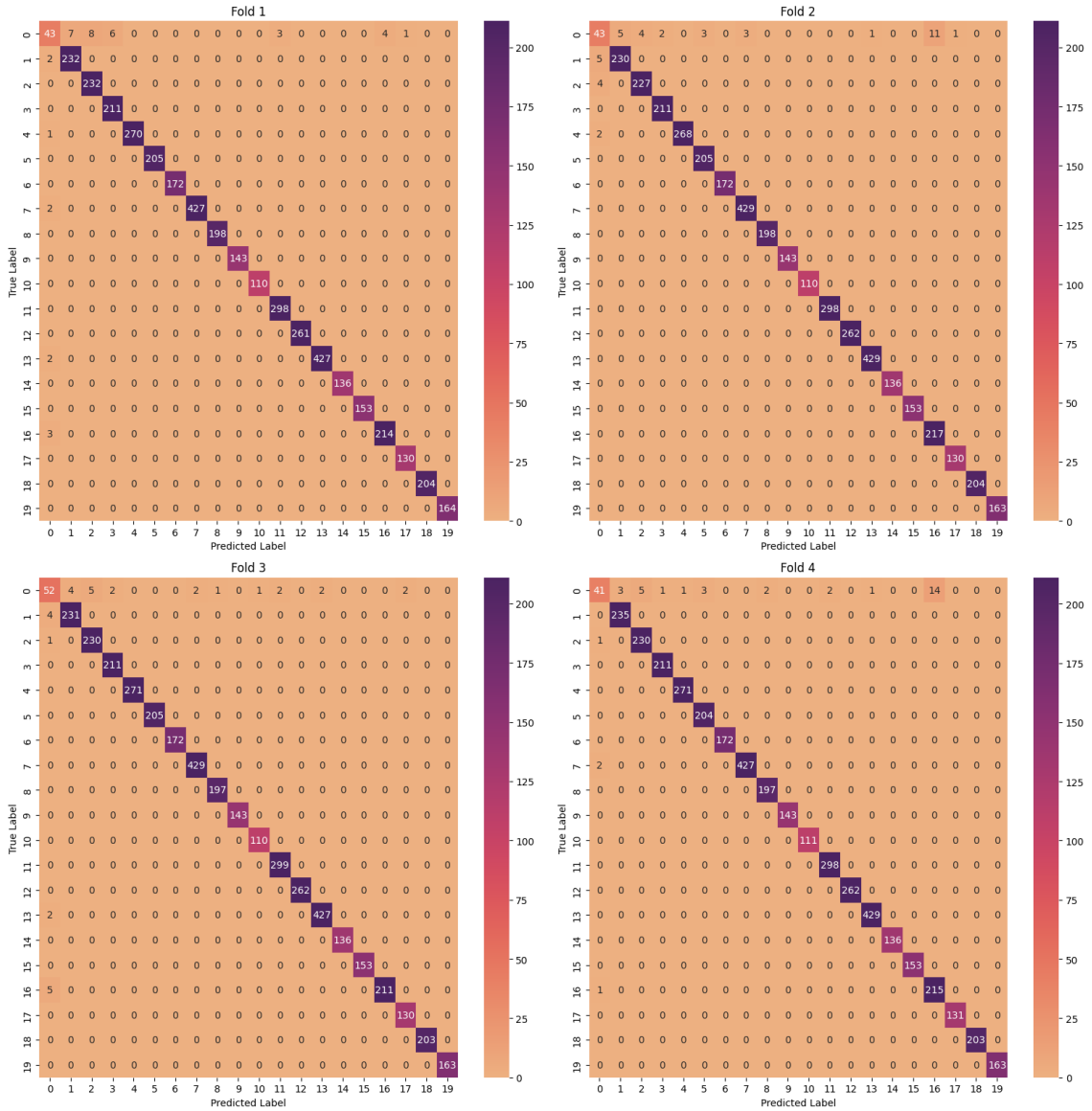


Figure 5.12: The cross-validation confusion matrices of the predictive clustering model.

used in this study scores lowest on all but one metric and is outperformed by the strong learners. The simple DT model comes in at fourth place for AUC ROC, accuracy, and Brier score. Surprisingly, the four F1-scores are quite similar and rather low, which can be further explained by examining the confusion matrices.

Table 5.5: The results of the supervised learning optimization. The best performance for each metric is shown in bold.

Algorithm	**Metric**	**Score**	**Parameters**
DT	AUC ROC	0.564	$MinSamplesSplit: 16, Criterion: gini$
	Accuracy	0.607	$MinSamplesSplit: 15, Criterion: gini$
	F1-score	0.532	$MinSamplesSplit: 13, Criterion: gini$
	Brier score	0.333	$MinSamplesSplit: 7, Criterion: entropy$
RF	AUC ROC	0.628	$MinSamplesSplit: 6, MinSamplesLeaf: 2$
	Accuracy	0.655	$MinSamplesSplit: 4, MinSamplesLeaf: 2$
	F1-score	**0.546**	$MinSamplesSplit: 9, MinSamplesLeaf: 10$

XGBoost	Brier score	0.217	$MinSamplesSplit: 4, MinSamplesLeaf: 2$
	AUC ROC	0.586	$LearningRate: 6.25E-5$
	Accuracy	0.746	$LearningRate: 0.001$
	F1-score	0.471	$LearningRate: 0.094$
	Brier score	0.182	$LearningRate: 0.020$
CatBoost	AUC ROC	**0.640**	$Depth: 6$
	Accuracy	**0.749**	$Depth: 6$
	F1-score	0.456	$Depth: 7$
	Brier score	**0.179**	$Depth: 4$

The confusion matrices of the sixteen different optimized models are seen in Figure 5.13. A confusion matrix allows one to better understand the effect of the parameters on the models and performance metrics, while the metrics allow for an easier comparison between different models. In addition, it can help with making the trade-off between false positives and false negatives. In this context, a false negative represents an individual that will be LTU but is not predicted as such, and a false positive represents an individual that will not be LTU but is predicted as such. As the goal of the PES is to help every individual as best as possible, false positives are preferred over false negatives.

From these confusion matrices, we can conclude that decreasing the $MinSamplesSplit$ parameter of the DT models decreases the number of false negatives, shifting slightly towards more false positives. The same happens for the $MinSamplesLeaf$ of the RF models. Decreasing this parameter again decreases the number of false negatives, shifting slightly towards more false positives and, generally, a more accurate classifier. For the boosting algorithms, we see similar patterns. If the $LearningRate$ of the XGBoost models is increased, the predictions shift more towards predicting the false class, decreasing the predicted positives. For the CatBoost models, we see that lowering the depth of the models shifts more predictions towards predicting the false class, decreasing the predicted positives, but this effect is minor.

Generally, all optimizations for the same model produce similar confusion matrices. Optimizing towards the brier score for the DT and RF models produces slightly more balanced classifiers, with classifier errors more divided over the two classes. The boosting models produce quite similar results and learn to classify all individuals in the negative class, which apparently produces better results than dividing the individuals over the classes. All models can be calibrated quite simply by changing the binary classification threshold. This will be discussed in more detail in the following paragraphs.

Following Bach et al. [32], the similarity matrix in Figure 5.14 has been created by comparing all the binary predictions of the different models with one another. A general pattern is that models optimized towards a different metric in the same model family return similar models. The DT optimized towards the Brier score and the RF model optimized towards F1-score are exemptions to this general rule. The two boosting algorithms predict almost identically, with similarities higher than 96%.

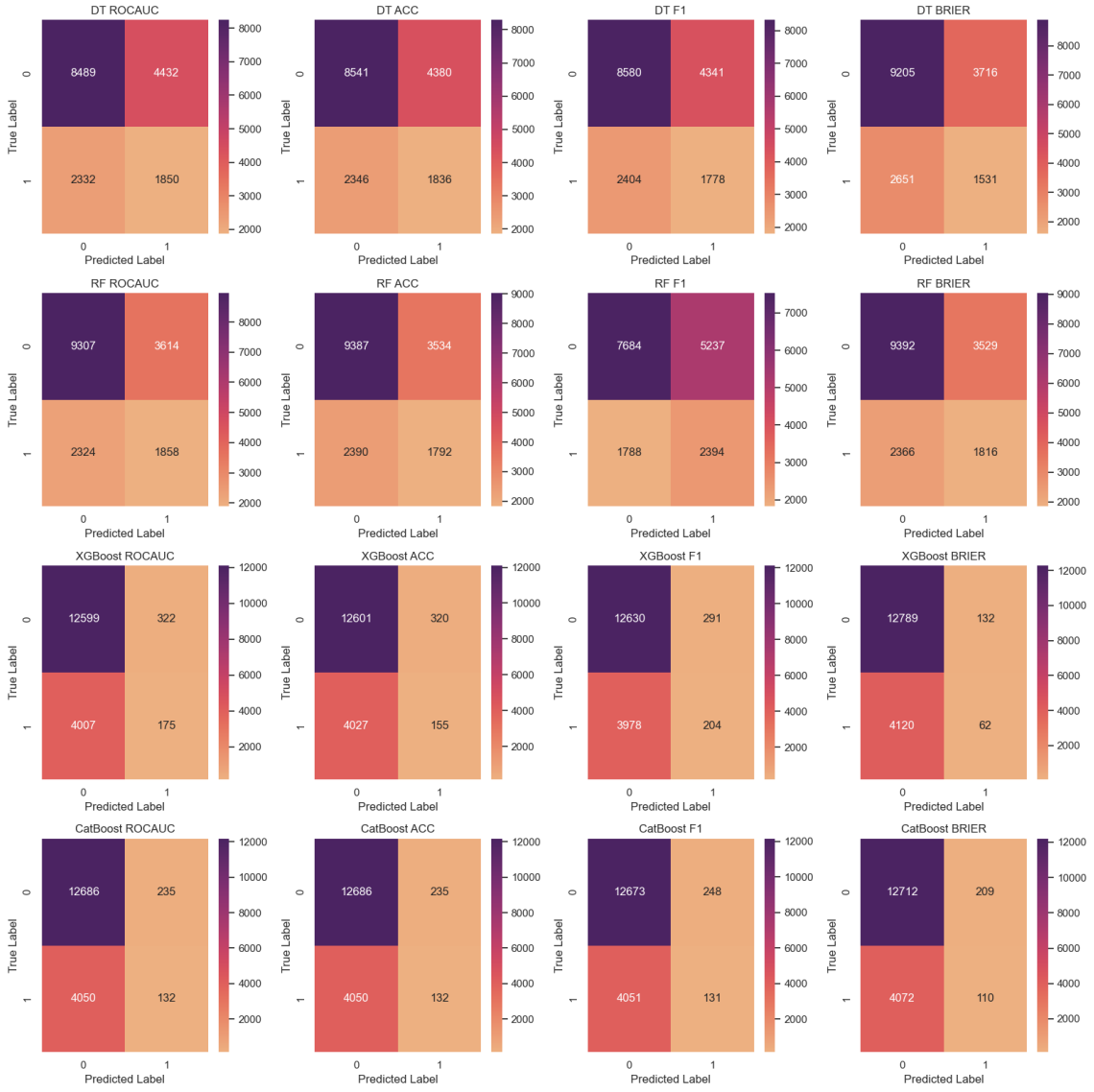


Figure 5.13: The confusion matrices of the different optimized supervised models.

From purely looking at the metrics, the CatBoost model is the clear winner of the comparison. However, when also consulting the confusion matrices, the boosting algorithms produce situations in which most customers would be predicted to be in the negative class or below 40% of benefits usage. In reality, this is not a very helpful result, but we may be able to increase the accuracy and F1 score by doing some further tuning of the decision threshold. In the confusion matrices, the RF produces promising results, which the same tuning may further improve. The rest of this chapter will proceed with the CatBoost and RF models with their optimized hyperparameters for the F1-metric, as this is the most balanced performance metric for unbalanced datasets available.

The optimum binary decision threshold is found with a visual inspection of the accuracy and f1-score for each possible threshold on the interval [0,1]. This is visualized in Figure 5.15. The minimum and maximum values for the accuracy are equal for both models. The minimum lies

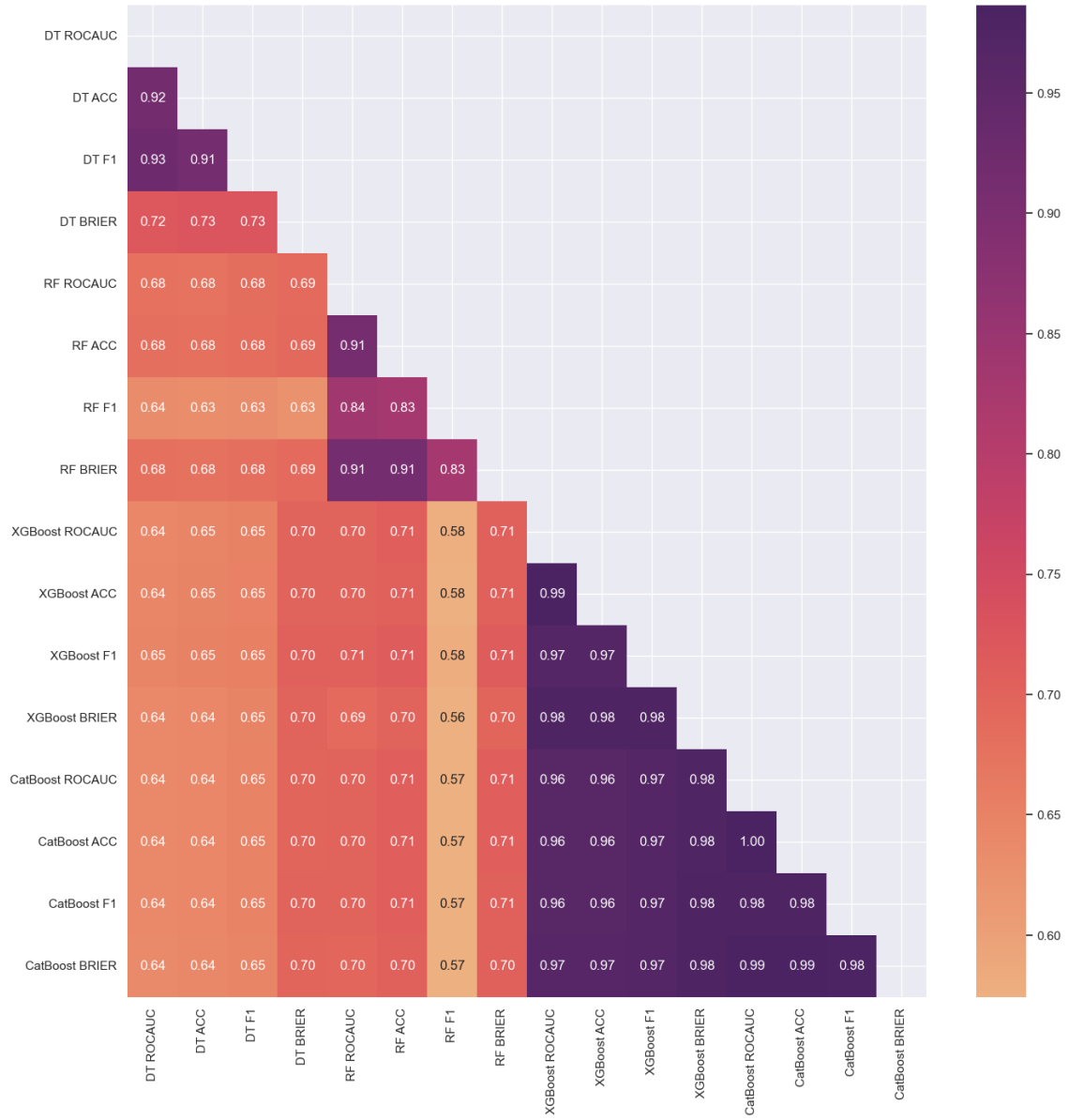


Figure 5.14: The similarity of prediction of the different optimized supervised models.

at 0.245, and the maximum at 0.755. For the F1-score, the minima are also equal at 0.196, but the maxima and the threshold for which the maximum value is valid differ. The maximum F1-score for the RF model is 0.560 at threshold 0.53, while the maximum of the CatBoost model is 0.572 at threshold 0.30. As the calibrated CatBoost model has a higher F1-score, we will regard this as the best-performing model

Another essential threshold of the model is the threshold that decides when an individual is regarded as long-term unemployed. This threshold has previously been set to 40% of benefits usage. This section will iterate over the possible threshold and create a new CatBoost model with the previously defined hyperparameters for each item of this interval. This is highly computationally expensive, so the threshold will be evaluated with a step size of 5. The same 5-fold

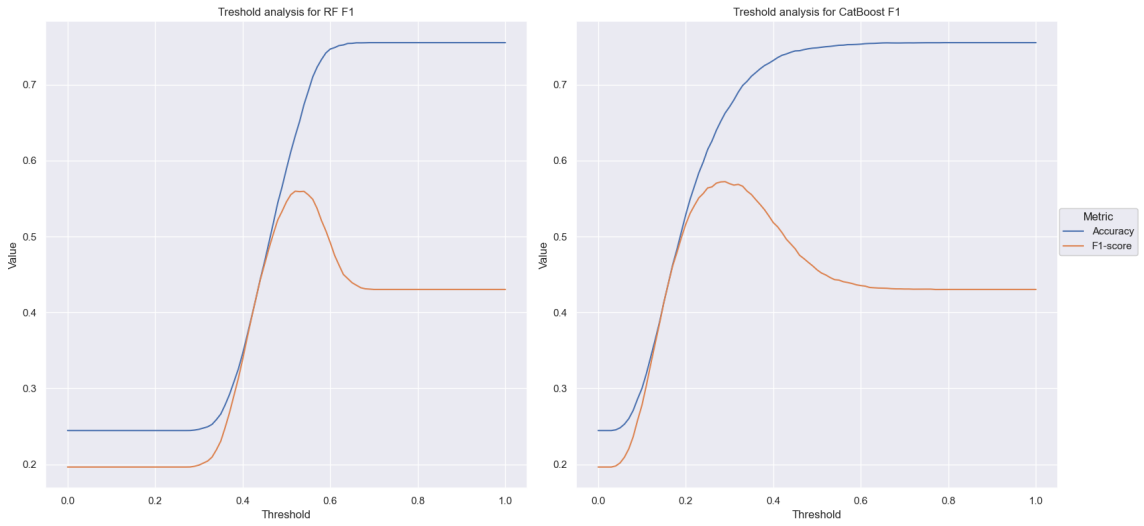


Figure 5.15: The threshold vs. accuracy and F1-score curves of the RF and CatBoost models.

cross-validation strategy as used previously will be employed, and the effect of the threshold on the different metrics is visualized in Figure 5.16.

From Figure 5.16, it becomes evident that no matter the LTU threshold, the model will always tend to predict all values in one class. This can be recognized by the low point in the line of the F1-score, which falls entirely together with the line of accuracy, at 0.5. The binary classes are almost equally distributed at the threshold for this value, 20. The near linear increase in accuracy and F1-score and decrease in Brier score is due to the distribution of the predictive value, also seen in Figure 5.1. This graph proves that there is no real optimal value for this threshold, and thus, the rest of this research will further consider a value of 40 for this threshold.

It is a proven fact that creating ensembles of weak learners generally tends to improve the model's accuracy while increasing the bias [101]. When ensembles of weak learners are made, bagging or boosting are techniques often used. However, the model considered in this research is not a weak learner but a strong learner, as a boosting model is technically an ensemble of weak learners. Creating ensembles of strong learners is also possible, generally called stacking algorithms. The advantage of this is that the errors and overfitting of one specific model might be cancelled by the errors and overfitting of the other models, just as with bagging.

Using the cluster information obtained from the unsupervised learning ML model, discussed in section 5.2, we have created multiple model configurations:

- **Predictive model**: This is the best-performing CatBoost model, as described in this section. It now includes the *cluster* feature, for which all values are set to -1, to prevent the model from using this feature for discovering patterns.
- **Predictive model clustering**: This is the best-performing CatBoost model, as described in this section, with the addition of the *cluster* feature as input feature.
- **Predictive model no outliers**: This is the best-performing CatBoost model, as described

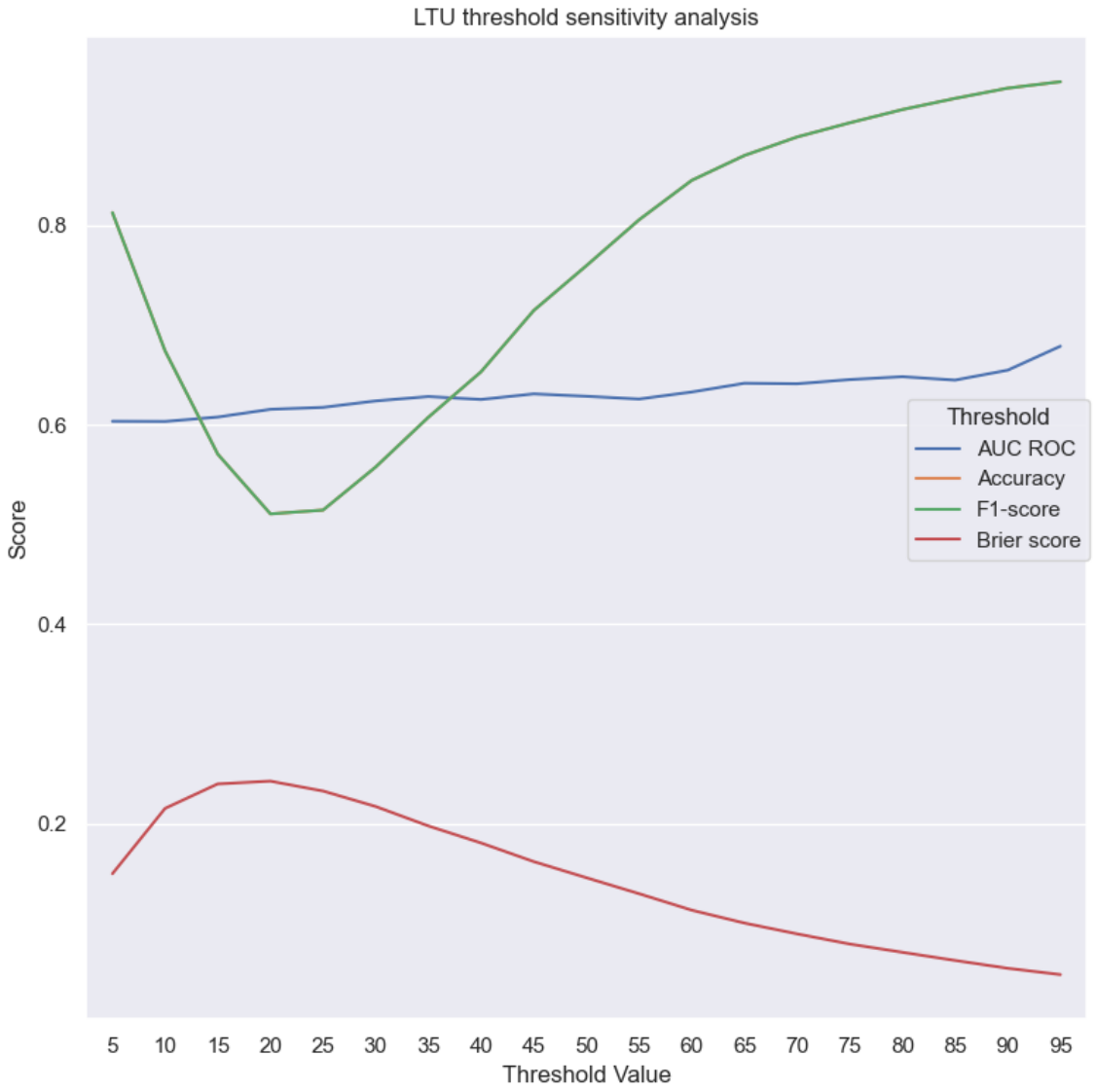


Figure 5.16: The long-term unemployed threshold vs. AUC ROC, accuracy, F1-score, and the Brier score curves of the optimized and calibrated CatBoost model.

in this section, with the addition of the *cluster* feature as input feature. However, it will not include the instances defined as outliers by the unsupervised learning ML model in the training set.

- **Predictive model cluster specific**: This is an ensemble of models trained specifically on each cluster's data. For each cluster, a separate model is trained and tested on instances from that cluster. The outputs of these models are then aggregated and presented through a common interface.

All ensembles are created using CatBoost models with the $depth$ parameter set to 7 and are evaluated using 5-fold cross-validation. The binary decision threshold needs to be recalibrated for every possible ensemble combination of the different model configurations, as every ensemble is essentially a completely new model. This is done using the same procedure as before,

plotting the F1-Score and accuracy for all possible thresholds. The resulting graph can be seen in Figure 5.17. All curves are very similar to the curves before, however, for the ensemble models, the F1-score does not increase above 0.50. The best-performing ensemble is the **Predictive model**, **Predictive model clustering**, and **Predictive model cluster specific** ensemble, with an F1-score of 0.499 at threshold 67, and an accuracy of 0.649.

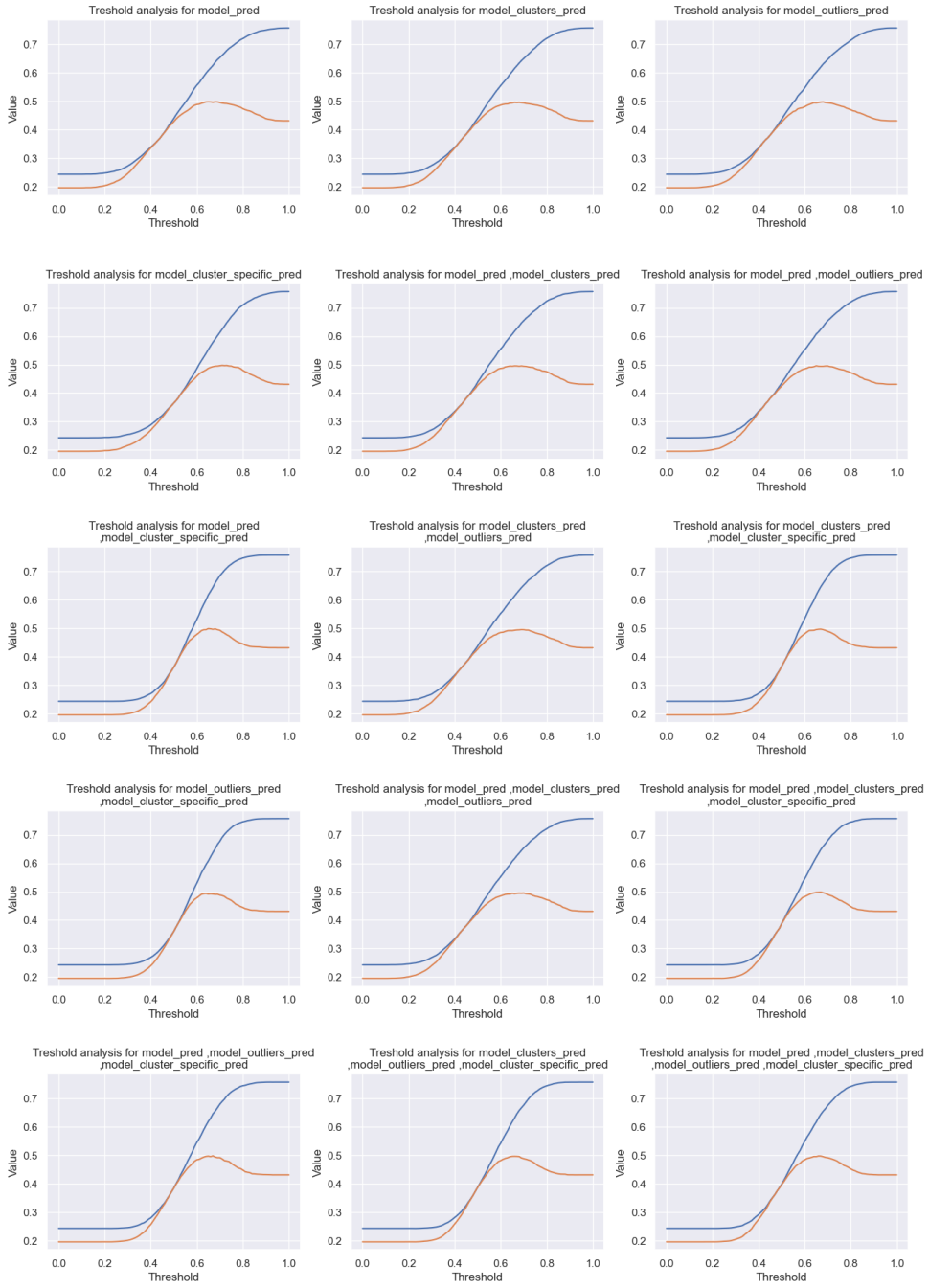
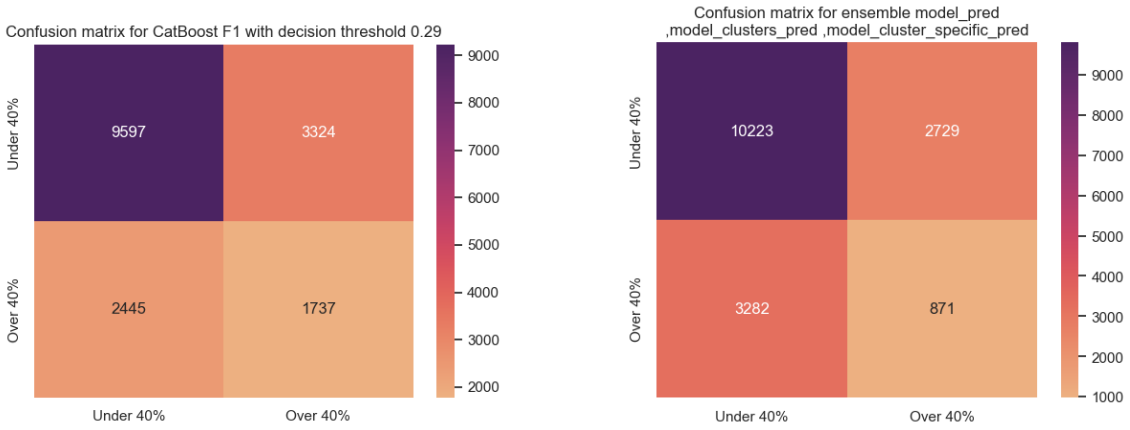


Figure 5.17: The threshold vs. accuracy and F1-score curves of the possible ensemble models.

Comparing the confusion matrices of the best regular predictive model and the best ensemble predictive model, as seen in Figure 5.18, leads to the conclusion that creating a complicated

ensemble from multiple CatBoost models with different configurations does not improve the overall predictive performance of the model. Therefore, we will resume the rest of this thesis with the initial hyperparameter-optimized and calibrated CatBoost model.



(a) Confusion matrix for the best CatBoost model

(b) Confusion matrix for the best ensemble model

Figure 5.18: The confusion matrices for the best regular and ensemble predictive models.

The distribution of the predictions over the different clusters is visualized in Figure 5.19. Some apparent differences are found when comparing this to the distribution of the predictive value over the clusters in Figure 5.7. For clusters 0, 4, 10, 11, 12, 13, 14, and 16, the model over-predicted the Over 40% class. The model predicted the same balance as the predictive value in clusters 1, 5, 8, 9, and 15. The model under-predicted the Over 40% class in clusters 2, 3, 6, 7, 17, 18.

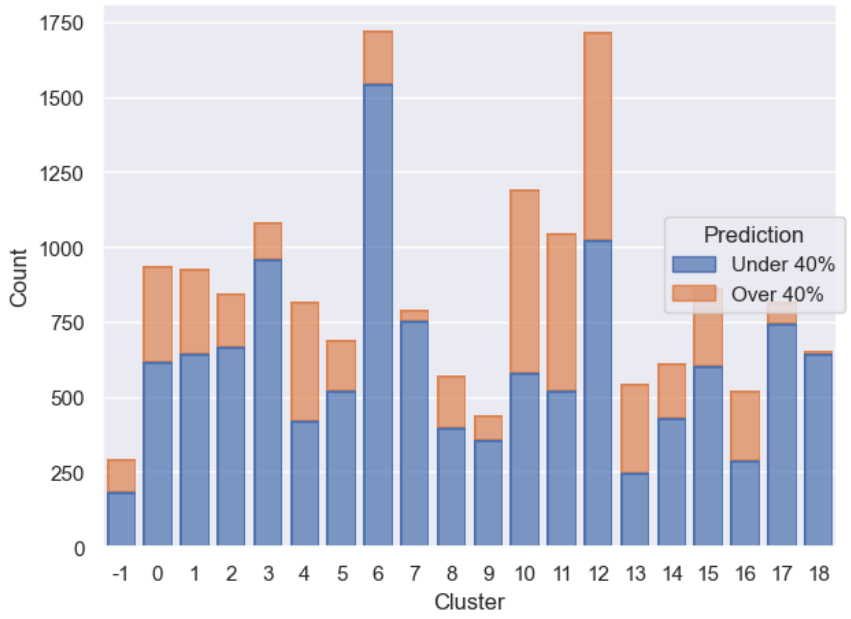


Figure 5.19: The distribution of the predictions over the different clusters.

5.4. SHAP ANALYSIS

After optimizing the model's predictive performance, we will analyse the SHAP-output of this model on different aggregation levels. The feature contributions will be analysed and discussed on the global, cluster, and individual levels. In contrast to the two higher-level analyses, a local analysis can visualize the direction of the effect given the input. As the original data is privacy sensitive, synthetic test entries have been generated manually and adhere to the clustering profiles described earlier. With these synthetic test entries, a what-if analysis will be conducted, showing the change in feature importance, and possibly the prediction, given a change in the synthetic input data.

Starting with the global summary statistics visualized in Figure 5.20, the five most important features of this model are the Region (*bezirksnummer*), Registration Type (*anmeldungsart_id*), Age (*ao_alter*), Language (*sprache_id*), and Job (*bezeichnung_m_de*). The five least important features Mobility (*mobilitaet_id*), Time of Labour (*arbeits_zeit_id*), Professional Qualification (*berufs_abschluss_id*), Job Function (*berufs_funktion_id*), and Employability Rate (*vermittlungs_grad*) are features with limited variance in the values. The most important features are the features containing the most variance, while the least important features are features without much variance, as we recall from Figure 5.3 and Figure 5.4. Remarkable is that the feature Degree of Disability (*iv_code_id*) also has a low variance but is more important for the model, showing the importance of this feature on the prediction.

The same patterns can be found on the cluster and global levels. However, the most contributing features for some clusters have more extreme values. This is the case for clusters 1, 4, 5, 11, 13, 15, 16, 17, and 18, where Registration Type (*anmeldungsart_id*) is the most contributing feature.

The most extreme case is cluster 16, which has a SHAP-value of 0.62 for Registration Type (*anmeldungsart_id*). Recall from Table 5.2 that this is a cluster with well-educated Swiss specialists capable of speaking the official local language of any age with more than three years of experience and reapplication to the PES within six months. The model predicts individuals in this cluster to have about a 50% probability of becoming LTU, while the actual distribution lies near a 33% probability. This might lead to a relative over-expenditure of PES resources on this cluster.

A what-if analysis on a synthetically created instance for this cluster's five most important features is found in Figure 5.24. The first waterfall plot shows the most important features and their values for the original instance, while in all other plots, the value of any of the five most important features is changed, keeping the others stable. When changing the application type to a first inscription (subplot b), the feature's contribution flips from negative to positive and the job is no longer one of the most important features but gets replaced by region. Changing the individual's age from 51 - 55 to 31 - 35 (subplot c) increases the job's and language's positive contribution and sum the other features. It also slightly decreases the contribution of the last active branch. Changing the individual's job from theatre tailor to painter (subplot d) re-

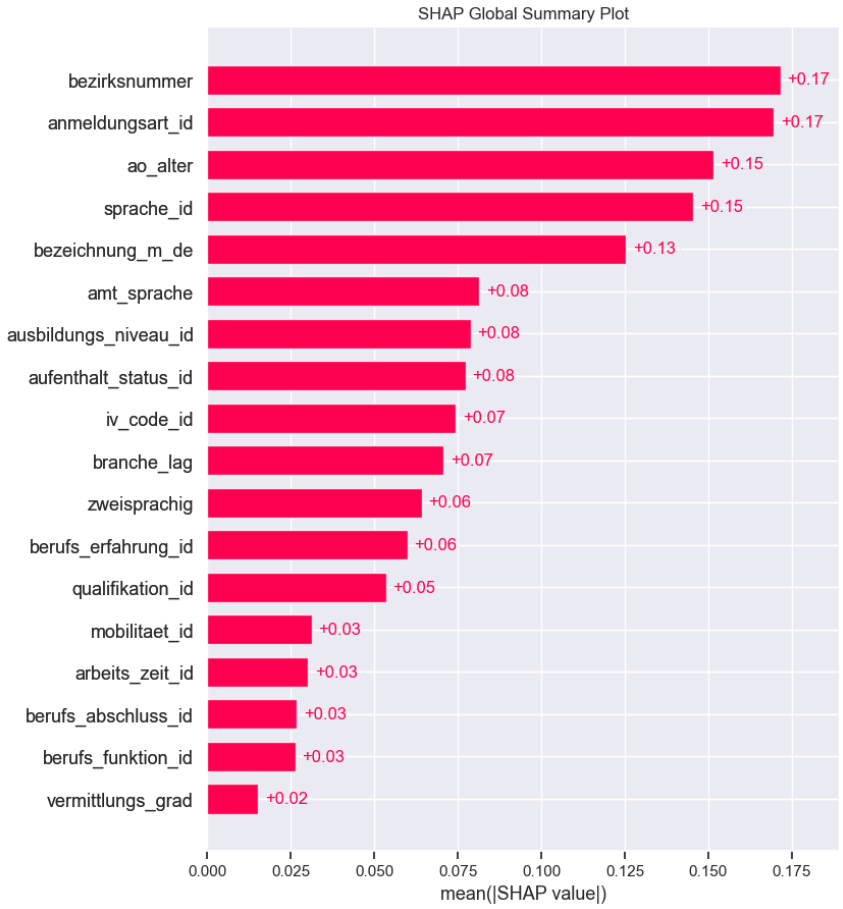


Figure 5.20: The summary statistics of the global SHAP values.

duces the positive contributions of the language and job and flips the prediction from under 40% to over 40% of benefits used. Changing the individual's region from Berner Mittelland to Emmental (subplot e) reduces the negative contribution of the branch and increases the positive impact of the other 14 features. This shows that this individual, according to the model, has a higher chance of finding a job in Emmental, indicating the possibility of a relocation. Finally, changing the individual's language from CH-German to German removes the positive contribution of this feature, flipping the prediction from under 40% to over 40% of benefits used. This shows that for this job, branch and location, it is essential to speak CH-German.

Another extreme case is cluster 5, with a SHAP-value of 0.36 for Registration Type (*anmeldungsart_id*). This cluster's actual probability of becoming LTU is nearly identical to the predicted distribution. This shows that more extreme SHAP-values for a cluster do not necessarily mean that the model performs better or worse than reality.

A what-if analysis on a synthetically created instance for this cluster's five most important features is found in Figure 5.25. The first waterfall plot shows the most important features and their values for the original instance, while in all other plots, the value of any of the five most important features is changed, keeping the others stable. When changing the application type from a

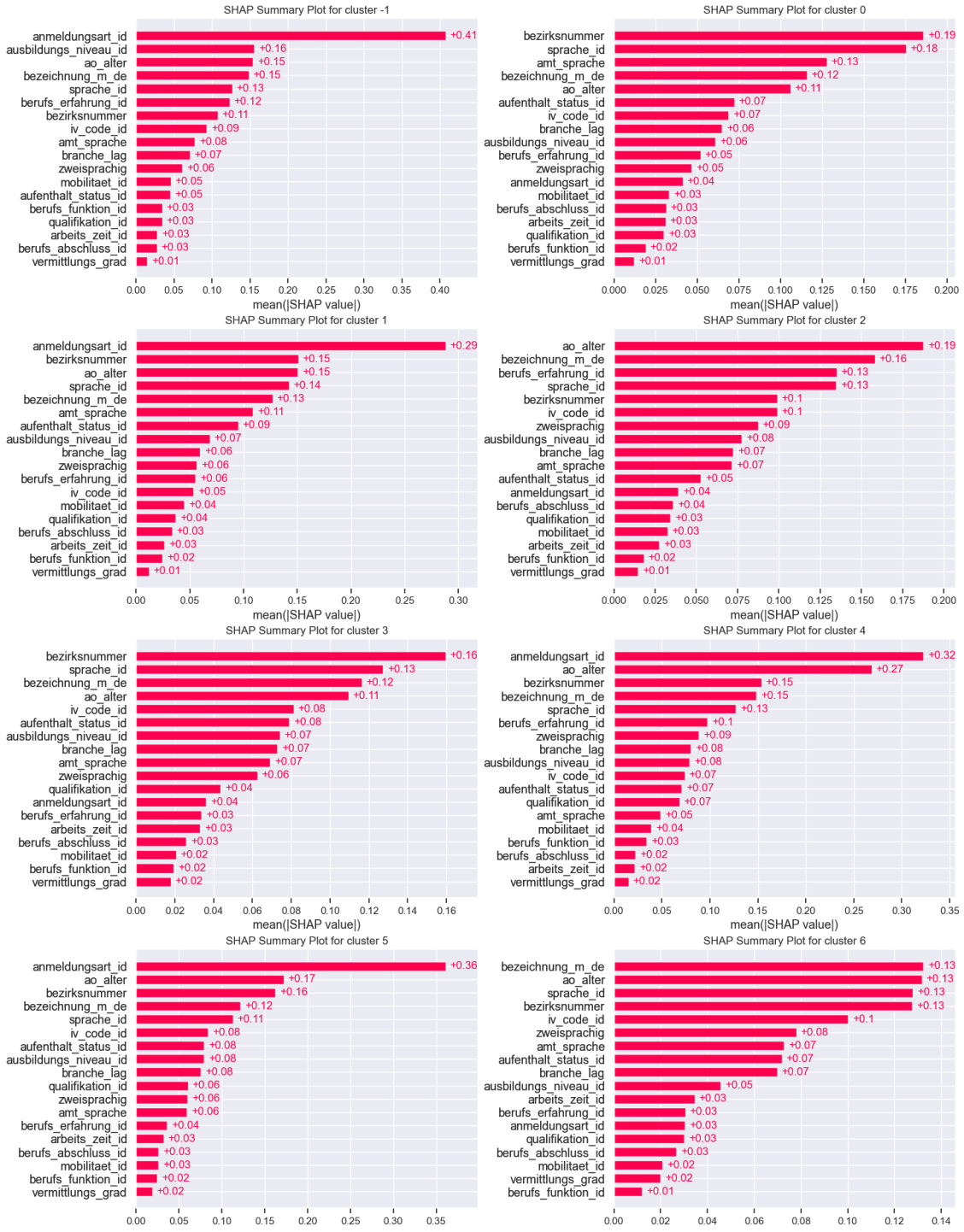


Figure 5.21: The summary statistics of the SHAP values per cluster (part 1).

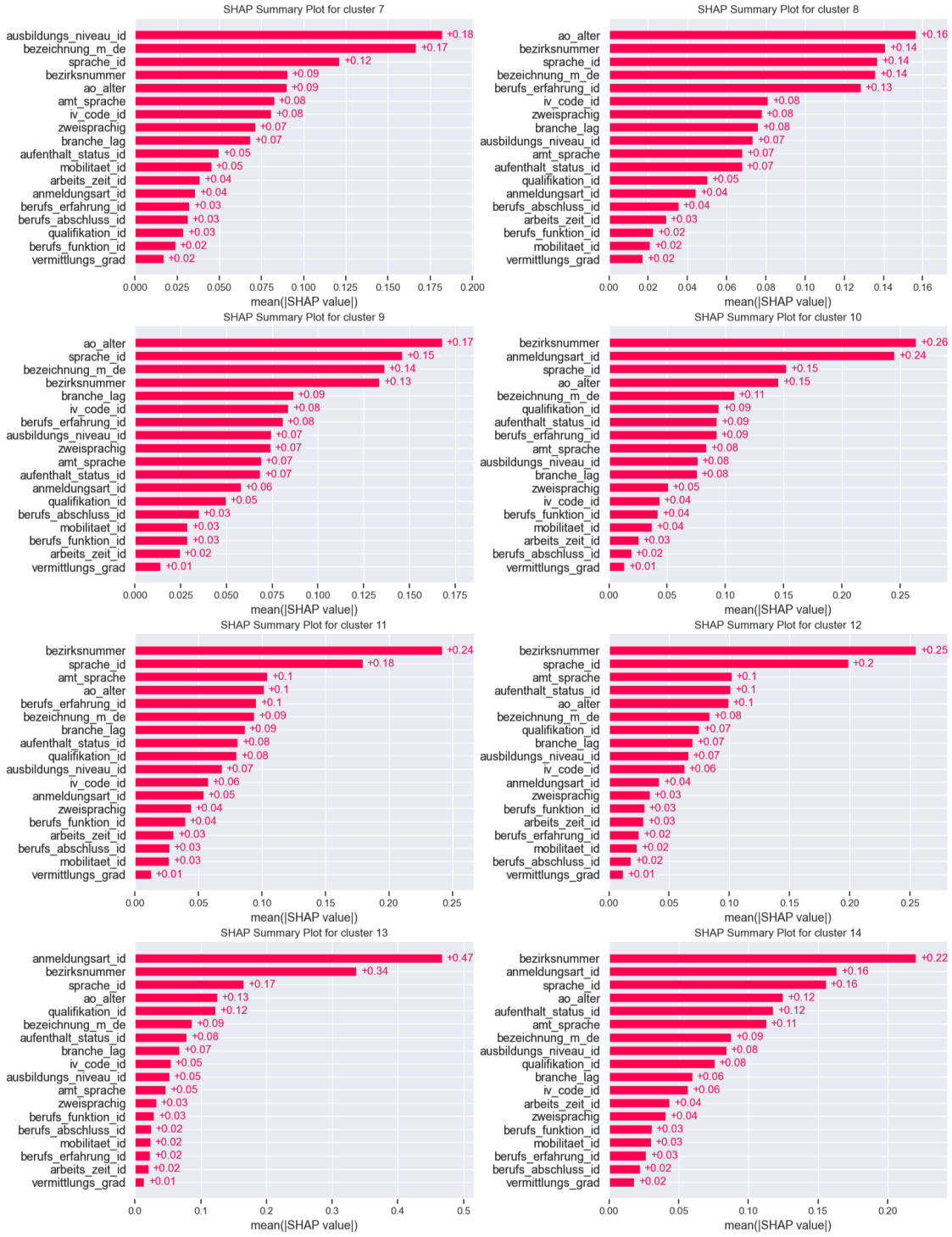


Figure 5.22: The summary statistics of the SHAP values per cluster (part 2).

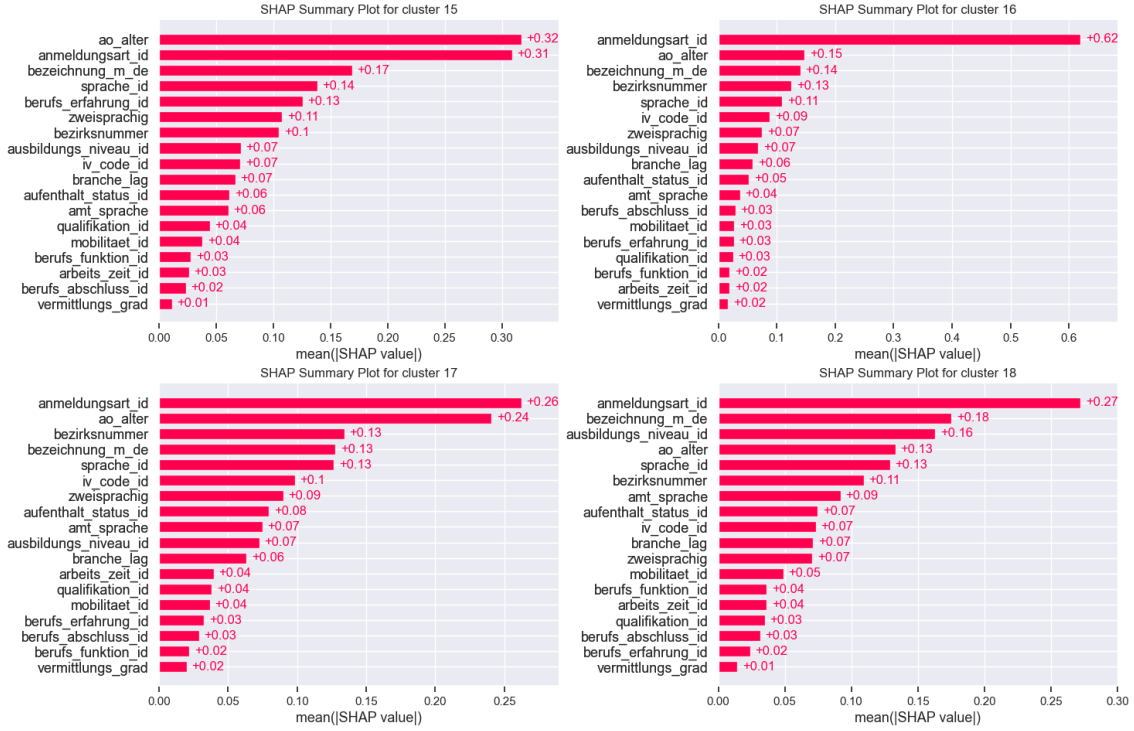
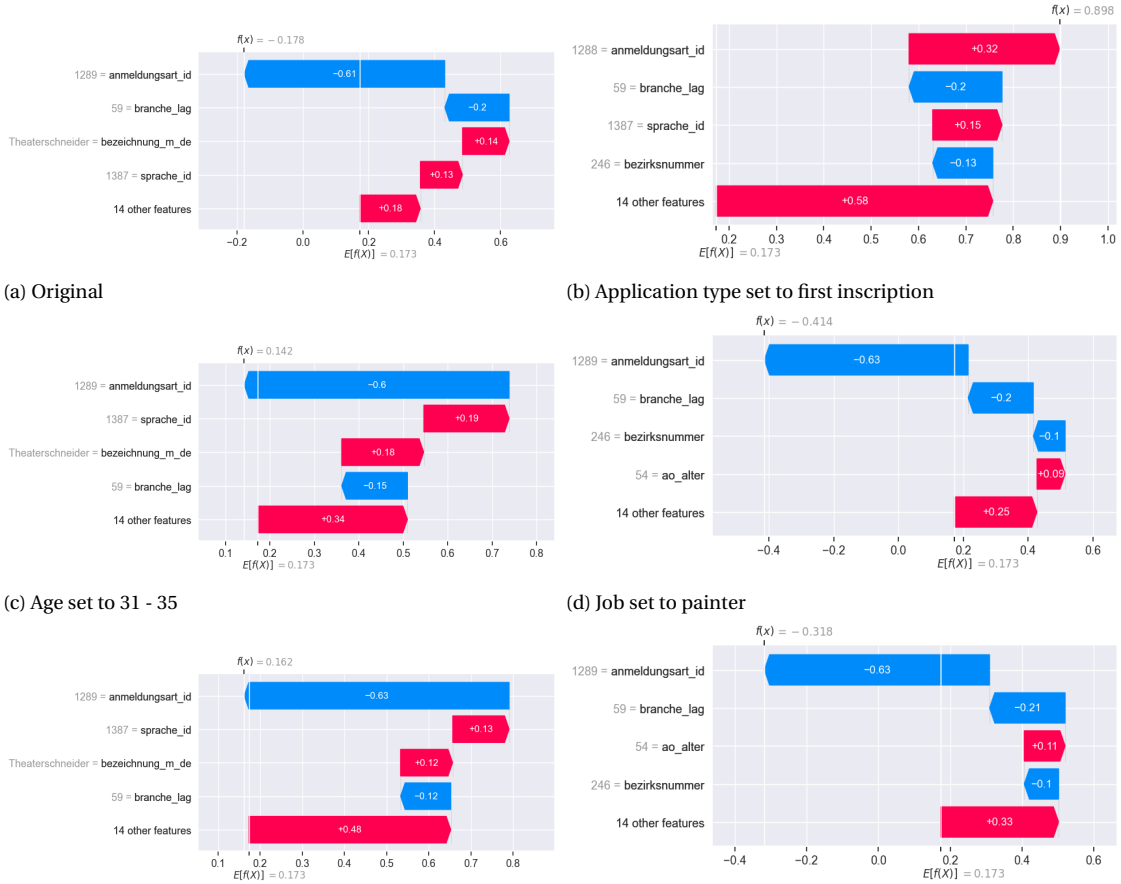


Figure 5.23: The summary statistics of the SHAP values per cluster (part 3).

first inscription to a reapplication within six months (subplot b), the feature's contribution flips from a slight positive to a strong negative contribution. Changing the individual's age from 35 - 40 to 21 - 25 (subplot c) introduces a strong negative contribution of this feature but reduces the contribution of the application type. This shows that somewhat older people with the same profile are less likely to become LTU. Changing the individual's job from host to salesman (subplot d) increases the application type's positive contribution and the negative contribution of the region and job. Changing the individual's region from Berner Mittelland to Thun (subplot e) increases the positive contributions of the language. Finally, changing the individual's language from CH-German to Other removes the positive contribution of this feature, and the fact that this individual speaks the official local language, which can be either CH-German, German, or French in this region, has a negative contribution to the prediction. The residency type A has a positive effect on the prediction. This means that the fact the individual was born in Switzerland only became relevant after changing the language.

An interesting case is cluster 7. For this cluster, unlike on the global level, the most important feature is Level of Education (*ausbildungs_niveau_id*) with a SHAP-value of 0.18. This is 0.10 higher than on a global level. This cluster represents Swiss nationals with relevant professional education in a specialist or management function capable of speaking the official local language aged between 31 and 65 with more than three years of experience and reapplication after six months. Their higher education is either of a professional or academic nature.

A what-if analysis on a synthetically created instance for this cluster's five most important features is found in Figure 5.26. The first waterfall plot shows the most important features and



(a) Original

(b) Application type set to first inscription

(c) Age set to 31 - 35

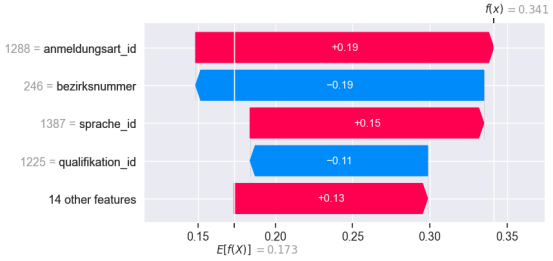
(d) Job set to painter

(e) Region set to Emmental

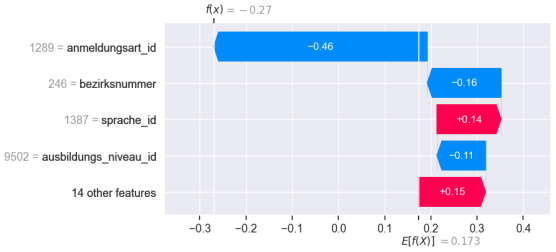
(f) Language set to German

Figure 5.24: What-if analysis using SHAP waterfall plots for one synthetic entry in cluster 16.

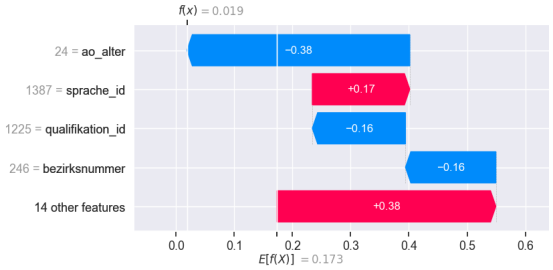
their values for the original instance, while in all other plots, the value of any of the five most important features is changed, keeping the others stable. The most important feature for this individual is the degree of disability with a value between 70 and 100%. This means that the individual is able to do less than 70% of work, showing that the model can find the relationship between disability and opportunities in the labour market. When changing the level of education from higher academic to higher professional (subplot b), the negative contribution of the job and the region and the positive contribution of education are reduced. Changing the job from head chef to team leader (subplot c) increases the positive contribution of education and application type and increases the negative contribution of age. Changing the language from CH-German to French (subplot d) reduces the negative effect of the degree of disability, which indicates a lack of training data for these instances. It increases the negative contributions of language and the official local language. The individual is from a region where both languages are spoken, so this is somewhat unexpected. Changing the region from Bern Mittelland to Jura increases the positive contribution of the education level and the negative contribution of the age and official local language. Decreasing the age from 61 - 65 to 46 - 50 increases the negative contribution of the degree of disability and the positive contribution of the education level.



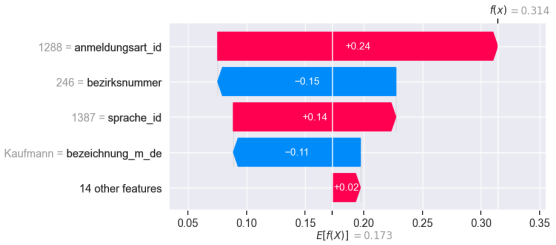
(a) Original



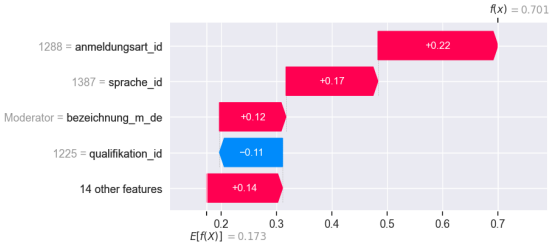
(b) Application type set to reapplication within six months



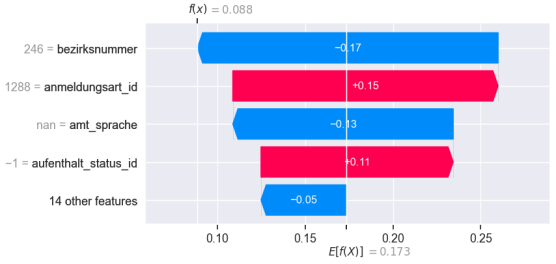
(c) Age set to 21 - 25



(d) Job set to salesman



(e) Region set to Thun



(f) Language set to Other

Figure 5.25: What-if analysis using SHAP waterfall plots for one synthetic entry in cluster 5.

(a) Original

(b) Education level set to Higher professional

(c) Job set to team leader

(d) Language set to French

(e) Region set to Jura

(f) Age set to 46 - 50

Figure 5.26: What-if analysis using SHAP waterfall plots for one synthetic entry in cluster 7.

6

CONCLUSION

6.1. LESSONS LEARNED

To conclude, this research consisted of an investigation into the integration of an XAI system in an existing DSS, in order to improve the allocation of resources in a PES. The main research question will be solved by answering the subquestions as presented in chapter 1.

RQ1: How to design an AI system for a PES?

Firstly, before shifting the focus to the mostly unexplored area of applying XAI in PES, the literature has been consulted for more general AI applications to transfer that knowledge. Most live systems are currently using simple LR and DT models, with three exceptions. A RF model is applied in the Belgian PES, Sweden employs a NN, and the New Zealand PES uses both RF and GBDT models. Only the Belgian and Swedish applications report an interpretability aspect. Precision, accuracy, and AUC ROC are used as validation measures in live profiling systems worldwide. Precisions between 0.8 and 0.91, accuracies between 0.6 and 0.86, and AUC ROC scores between 0.63 and 0.79 are found, setting the benchmark for the current research. The AI applications generally use as much information as possible, ranging from socio-economic characteristics of the jobseekers to motivation for finding a new job to job readiness and opportunities. However, people who need professional help most are generally hard to profile. These people might have psychological issues, debt, or drug addiction, which are generally not stored in the data of the PES. Finally, the AI profiling application should be as complex as the Active Labour Market Policy in place in the country.

RQ2: How to integrate an XAI system into a DSS?

Secondly, a thorough exploration of the XAI in DSS has been conducted. XAI predictions can be used on a global or local level to understand the contributions of features to the model's prediction. One way of achieving this is building the explainability directly into the model, called

model-specific explainability. Another option is to create a model-agnostic explainer, which needs the model input and its predictions and creates a simpler, interpretable model that allows the user to understand the model. Generally, it is desirable to have a model-agnostic model that allows for simple changes in models used and can explain both globally and locally, allowing for different levels of understanding. The most occurring XAI technologies are SHAP and LIME. Many other technologies have been identified, but these technologies cannot function globally and locally or are not model-agnostic. One technique that works both globally and locally and is model-agnostic is Natural Language Explanations, in which the numerical output of an explanation method, generally SHAP, is provided to a Large Language Model, which generates a textual explanation better suited for human interpreters. Another technique adhering to these properties is AcME, which is less popular in literature. This might be related to a defective Python package.

> *RQ3: How to extract new knowledge from existing PES datasets using XAI technologies?*

Finally, integrating the knowledge obtained from answering the previous subquestions, different AI and XAI models have been successfully applied to a Swiss PES dataset. Using multiple unsupervised ML methods, reproducible clusters have been established in the clientele of the PES. Within these clusters, a set of clusters has been defined in which the proportion of clients with a benefits usage of over 40% is higher than the population proportion. An example of this is cluster 11, characterized by performing an auxiliary function, a lack of professional education, a migrational background, not being able to speak the official local language, having less than three years of relevant experience, and reapplying to the RAV after at least six months of being unemployed. There are also examples of clusters where the proportion of clients with a benefits usage of over 40% is higher than the population proportion, like, for instance, cluster 18. This cluster contains clients who perform specialist and management functions, have Swiss higher education and a Swiss passport, speak the official local language, and have more than three years of relevant experience.

We are using hyperparameter tuning and model calibration to train a profiling model with an F1-score of 0.499 and an accuracy of 0.649. Based on the input and predictions for this model, the corresponding SHAP-values can be calculated. With these SHAP-values the models can be interpreted on global, cluster, and individual levels. For most models, the Region and Registration Type are the most important features, followed by Age and Language. Generally, the most important features are the features containing the most variance, while the least important features are those without much variance. These feature importances vary on cluster level. The most important profiling feature for individuals in cluster 7 is the Education Level. The individual level allows for understanding complex, multidimensional relationships between feature values.

6.2. PRACTICAL AND SCIENTIFIC CONTRIBUTIONS

From a practical perspective, the main contribution of this research is the development and deployment of an XAI system for profiling and clustering unemployed in a PES. The PES has a highly complex Active Labour Market Policy but previously used simple rule-based policies to assign labour market measures. With this new system, the PES can improve and possibly automate the assignment of labour market measures. The performed clustering allows for the design of more targeted labour market measures by tailoring the measures to the needs of the different groups. In addition, the output of the profiling model can be used to get an automated warning of individuals at risk of LTU, allowing counsellors to adjust their focus and assign labour measures to the customers who need it most. This LTU warning can be accompanied by the reason for the warning in the form of the features and values that contributed most to this prediction, allowing the counsellor to reassess the need for an intervention and adjust the labour market measure if necessary. The above innovations should lead to a more efficient allocation of human and financial resources.

From a scientific perspective, the main contribution is filling the existing gap in literature. Previously, AI in PES and XAI in DSS were completely distinct topics. To the best of our knowledge, there was no literature bridging the gap and applying XAI techniques in the DSS of PES. The literature review conducted in this research has bridged this gap, and the application proves the possibilities of transferring knowledge between the two research areas. Additionally, this literature, to the best of our knowledge, is the first to implement unsupervised ML techniques in PES, which allows for more group-level statistics and policies, improving the efficiency of policy creation. Multiple strong relationships between language, residency type, and the probability of LTU are found.

The generalizability of this research is reinforced by the widespread use of AI models in various Public Employment Services (PES) globally, showing the potential of the more complex GBDT algorithms compared to the more widely used DT and RF algorithms. The application of the SHAP XAI method is transferable due to its model-agnostic properties. These methods and the clustering approach used to profile clients based on features such as education, language proficiency, and work experience make the methodology applicable to other PES systems within the German-speaking part of Switzerland, as the input data is similar. Care must be taken in generalizing the research results in regions other than that, as the research uses a dataset from one of the Swiss cantons. This dataset might reflect inherent specific socio-economic and demographic factors that may not be applicable to other regions, even within German-speaking Switzerland, reducing the direct generalizability of the discovered feature interactions.

6.3. LIMITATIONS AND FUTURE RESEARCH RECOMMENDATION

Although this research shows many valuable insights, several limitations impacting the outcomes should be considered when interpreting the findings.

1. **Analysis of literature**: The literature review of this study has merely employed the Scopus

citation database to collect articles. As this database does not always contain all scientific literature published in Elsevier papers, this limits the findings of the literature review. The search criteria were composed, and relevant articles were selected based on the researcher's knowledge and judgement. Employing the snowballing procedure helped reduce these limitations. As mentioned in subsection 2.2.2, a DSS was connected to explanations with the word intelligent, showing that a possible area of interest might be intelligent DSSs, which was not included in the literature study.

2. **Involvement of end-users**: This study is performed with the assistance of the controlling division of the PES in question. The end-users, in this case the counsellors, have not been involved in the process of selecting data, training and validating models, and explaining the model outputs. This may lead to increased friction during the implementation process in the form of unwillingness to cooperate or nonunderstanding of the tool and its capabilities. The involvement of end users would also be helpful with model validation, as the tool mimics the intuition and experience of a counsellor by profiling unemployed as either a low- or high risk of unemployment.

3. **Data quality**: Even though the data quality has greatly at the PES has greatly improved over the years, the data quality is still not optimal. As with all ML applications, the quality of the input data is equal to the quality of the trained model. One of the main problems is that much of the data at the PES is collected manually by the counsellors and is thus prone to errors and inconsistencies. Furthermore, the data could be enriched by implementing labour market history, soft skills, and regional labour market data. This would also allow for the engineering of additional input features, which generally boost the performance of predictive models.

4. **Validation metrics employed and resulting optimizations**: The CHI, DBI, and MSC are generally higher for convex clusters rather than density-based clusters like those obtained through DBSCAN and HDBSCAN [106]. The optimizations for these algorithms may thus result in a non-optimal solution. However, Agglomerative clustering is a distance-based approach that can produce clusters of any shape. Therefore, this limitation also affects this algorithm, so this should not affect the outcomes of which algorithm to use. It might influence the optimal hyperparameters for each of the models. Solutions with many smaller clusters perform better in scoring than a few larger clusters, which is caused by including inter-cluster distance in all metrics. When disregarding this inter-cluster distance, larger clusters may be more desirable. In addition, the original notation of the DBI only allows for truthful applications in the Euclidian space. The clustering is performed in the Gower space as we employ Gower distances. The application is still possible, but we should question the validity of the results when comparing them to clustering performed in the Euclidean space.

5. **Model performance**: The model performance is somewhat lower than wished. The model performs poorly when compared to other models used throughout PES worldwide. Our

hypothesis for the slightly disappointing model performance is that the dataset contains many highly correlated features with low variance in the categories. This means a slight variance can already lead to a wrongly predicted label. Improving the accuracy might, however, include more discrimination in the model as per the accuracy-equity trade-off.

6. **Model evaluation**: The explainability method we have applied is reported to be unstable, meaning that it might not always report the proper feature contributions. However, the framework on which these claims are based did not come with runnable code, removing the possibility to verify these claims. Additionally, the main instability of SHAP is reiteration similarity, meaning that recalculation of the explainable models leads to different models. However, SHAP uses random sampling to construct these explainable local linear models, which produce the feature contributions. Therefore, some sort of randomness is to be expected, and the use of random number generator seeds is not mentioned. SHAP is still the industry-standard XAI methodology, and to the best of our knowledge, no other authors report similar issues. We thus conclude that it's better to have a potentially unstable measurement of feature contribution than none at all. Additionally, The explanations provided by SHAP are as good as the model. Therefore, the presented feature importances should be interpreted with caution.

7. **Explainability style**: Currently, the users of the model are presented with a SHAP feature contribution plot for a specific instance which includes the prediction of the model. Multiple sources, however, report that a better design would not include the model's prediction to prevent overreliance on the XAI system and keep users cognitively engaged allowing them to recognize when the model output is incorrect.

8. **Model explainability validation**: Currently, the model explanations are manually evaluated by looking at feature importances and spotting unexpected contributions. This is not scalable and leads to the fact that not all explanations are validated, but only a random subset. In addition, this method of validation includes human biases and can be problematic in spotting inconsistencies. Using numerical model explainability validation metrics would be preferred. However, to the best of our knowledge, the only Python package offering this is currently not working.

9. **Lack of external validation**: Although all models have been 5-fold cross-validated, the models have not yet been tested with a large external validation dataset and if the supervised model can hold up the accuracy and F1-score when used to predict many new, unseen data points. As the appearance of many new data points may shift the dense areas of the data, the so-called creep, it would be exciting to see if the unsupervised model presents different clusters after training with an extended dataset.

Future studies may include a more specific study on the explainability style to be used in XAI applications in PES. Also, a technique to remove correlation between features in a dataset while keeping the interpretability aspect, specifically for categorical features, would be a research

topic of interest. Furthermore, this research heavily leans towards the ML part in the non-symbolic school of AI. Future research may explore how symbolic AI, e.g., ontologies, can be implemented in PES. This study tried to maximize the impact of the AI profiling tool by incorporating it in the very beginning of the unemployment process, sacrificing additional information about the unemployed that is collected in a later stage of the process. An interesting research topic would be to compare the effectiveness of AI profiling at different stages of the unemployment process and find the balance in this information effectiveness trade-off. Additionally, this research has found some patterns between specific features which require additional research to fully understand.

Future case-specific research may include adding labour market history, soft skills, and regional labour market data to increase the input for the predictive model. Also, with this extra data, more informative features can be engineered, which may greatly enhance model performance. Additionally, the region of implementation could be extended. When changing the scope of this project from one Swiss canton to the whole country, much more data is available which might improve the performance of the predictive model. Additionally, a randomized controlled trial experiment may be conducted based on the current study in which the algorithm is extended to provide recommendations on labour market measures and is compared to the current way of working at the PES. This research would also allow for the testing of the organizational consistency of the PES. Finally, the latter can be combined with a cost-benefit analysis of the clustering and profiling tool to find if employing analytical methods increases the efficiency of the PES.

REFERENCES

[1] W. Bank, World bank open data, 2024. URL: https://data.worldbank.org.

[2] S. W. I. swissinfo.ch, Swiss unemployment rate jumps in second quarter of 2024, 2024. URL: https://www.swissinfo.ch/eng/workplace-switzerland/sharp-rise-in-swiss-unemployment-in-q2-fso/87266005.

[3] D.-G. for Internal policies, On the social consequences of unemployment, 2009. URL: https://www.europarl.europa.eu/RegData/etudes/note/join/2010/429996/IPOL-CRIS_NT(2010)429996_EN.pdf.

[4] Investopedia, The cost of unemployment to the economy, 2024. URL: https://www.investopedia.com/financial-edge/0811/the-cost-of-unemployment-to-the-economy.aspx.

[5] D. Allhutter, F. Cech, F. Fischer, G. Grill, A. Mager. Algorithmic Profiling of Job Seekers in Austria: How Austerity Politics Are Made Effective. Frontiers in Big Data (2020). doi:10.3389/fdata.2020.00005.

[6] S. Desiere, K. Langenbucher, L. Struyven, Statistical profiling in public employment services: An international comparison, Technical Report, OECD, Paris, 2019. doi:10.1787/b5e5f16e-en.

[7] L. Pas, Statistical criminal profiling, 2018. URL: https://www.universiteitleiden.nl/binaries/content/assets/science/mi/scripties/statscience/2017-2018/2018_07_25_masterthesis_pas.pdf.

[8] M. Machado, J. Osterrieder, A. Amato, Integrating Early Warning Systems with Customer Segmentation: An Information Management Approach to Identifying Business Opportunities for Commercial Customers in the Financial Industry, 2024. doi:10.2139/ssrn.4779632.

[9] C. L. Christiansen, C. N. Morris. Improving the statistical approach to health care provider profiling. Annals of Internal Medicine (1997). doi:10.7326/0003-4819-127-8_part_2-199710151-00065.

[10] O. E. Dictionary, Profiling, noun, 2007. URL: https://www.oed.com/dictionary/profiling_n, accessed: May 2024.

[11] V. Landeghem, B. Desiere, S. Struyven, Ludo. Statistical profiling of unemployed jobseekers. IZA World of Labor (2021). doi:10.15185/izawol.483.

[12] R. M. Dreyling, T. Tammet, I. Pappel, in: T. K. Dang, J. Küng, T. M. Chung (Eds.), Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications, Communications in Computer and Information Science, Springer Nature, Singapore, 2023, pp. 341–351. doi:10.1007/978-981-99-8296-7_24.

[13] P. Arni, A. Schiprowski, Die Rolle von Erwartungshaltungen in der Stellensuche und der RAV-Beratung, Technical Report, SECO, Staatssekretariat für Wirtschaft, 2016. URL: https://www.seco.admin.ch/seco/de/home/Publikationen_Dienstleistungen/Publikationen_und_Formulare/Arbeit/Arbeitsmarkt/Informationen_Arbeitsmarktforschung/Erwartungshaltungen_Stellensuche_RAV-Beratung.html.

[14] E. Amparore, A. Perotti, P. Bajardi. To trust or not to trust an explanation: using LEAF to evaluate local linear XAI methods. PeerJ Computer Science (2021). doi:10.7717/peerj-cs.479.

[15] W. Dossche, S. Vansteenkiste, B. Baesens, W. Lemahieu. Interpretable and Accurate Identification of Job Seekers at Risk of Long-Term Unemployment: Explainable ML-Based Profiling. SN Computer Science (2024). doi:10.1007/s42979-024-02884-4.

[16] C. Bishop, M., Pattern Recognition and Machine Learning, Information Science and Statistics, Springer Science+Business Media, LLC, 2006.

[17] M. Ester, H.-P. Kriegel, X. Xu, in: KDD-96 Proceedings, AAAI, 1996, pp. 226–231. URL: https://cdn.aaai.org/KDD/1996/KDD96-037.pdf.

[18] R. J. G. B. Campello, D. Moulavi, A. Zimek, J. Sander. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. ACM Transactions on Knowledge Discovery from Data (2015). doi:http://doi.org/10.1145/2733381.

[19] A. Rai, P. Constantinides, S. Sarker. Editor'S comments: next-generation digital platforms: toward human–AI hybrids. MIS Quarterly (2019).

[20] Q. V. Liao, K. R. Varshney. Human-Centered Explainable AI (XAI): From Algorithms to User Experiences. arXiv preprint arXiv:2110.10790v5 (2022). doi:10.48550/arXiv.2110.10790.

[21] N. Alangari, M. El Bachir Menai, H. Mathkour, I. Almosallam. Exploring Evaluation Methods for Interpretable Machine Learning: A Survey. Information (Switzerland) (2023). doi:10.3390/info14080469.

[22] H. Mucha, S. Robert, R. Breitschwerdt, M. Fellmann, in: Conf Hum Fact Comput Syst Proc, Association for Computing Machinery, 2021, pp. 1–6. doi:10.1145/3411763.3451759, journal Abbreviation: Conf Hum Fact Comput Syst Proc.

[23] I. S. Organziation, Machine learning: Everything you need to know, 2024. URL: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/news/insights/AI/machine-learning-need-to-know.evergreen.html.

[24] D. J. Power, Decision support systems: concepts and resources for managers, 1. publ ed., Quorum Books, Westport, Conn., 2002.

[25] A. Scoppetta, T. Johnson, A. Buckenleib, Tackling long-term unemployment through risk profiling and outreach: a discussion paper from the employment thematic network. Technical Dossier no. 6, May 2018, Technical Report, Directorate-General for Employment, Social Affairs and Inclusion (European Commission), 2018. doi:10.2767/524273.

[26] K. Bern, Amt für arbeitslosenversicherung, 2024. URL: https://www.weu.be.ch/de/start/ueber-uns/die-organisation/amt-fuer-arbeitslosenversicherung.html, accessed: May 2024.

[27] F. Wijnhoven, M. Machado, Writing & Presenting Academic Papers, 2024. doi:10.2139/ssrn.4798954.

[28] A. Amato, J. R. Osterrieder, M. R. Machado. How can artificial intelligence help customer intelligence for credit portfolio management? A systematic literature review. International Journal of Information Management Data Insights (2024). doi:10.1016/j.jjimei.2024.100234.

[29] A. M. Antoniadi, Y. Du, Y. Guendouz, L. Wei, C. Mazo, B. A. Becker, C. Mooney. Current Challenges and Future Opportunities for XAI in Machine Learning-Based Clinical Decision Support Systems: A Systematic Review. Applied Sciences (2021). doi:10.3390/app11115088.

[30] A. Berman, K. de Fine Licht, V. Carlsson. Trustworthy AI in the public sector: An empirical analysis of a Swedish labor market decision-support system. Technology in Society (2024). doi:10.1016/j.techsoc.2024.102471.

[31] V. Andonovikj, P. Boškoski, S. Džeroski, B. Boshkoska. Survival analysis as semi-supervised multi-target regression for time-to-employment prediction using oblique predictive clustering trees. Expert Systems with Applications (2024). doi:10.1016/j.eswa.2023.121246.

[32] R. L. Bach, C. Kern, H. Mautner, F. Kreuter. The impact of modelling decisions in statistical profiling. Data & Policy (2023). doi:10.1017/dap.2023.29.

[33] K. Braunsmann, K. Gall, F. Rahn. Discourse Strategies of Implementing Algorithmic Decision Support Systems: The Case of the Austrian Employment Service. Historical Social Research (2022). doi:10.12759/hsr.47.2022.30.

[34] M. Considine, M. Mcgann, S. Ball, P. Nguyen. Can Robots Understand Welfare? Exploring Machine Bureaucracies in Welfare-to-Work. Journal of Social Policy (2022). doi:10.1017/S0047279422000174.

[35] S. Desiere, L. Struyven. Using Artificial Intelligence to classify Jobseekers: The Accuracy-Equity Trade-off. Journal of Social Policy (2021). doi:10.1017/S0047279420000203.

[36] K. B. Haug. Structuring the scattered literature on algorithmic profiling in the case of unemployment through a systematic literature review. International Journal of Sociology and Social Policy (2022). doi:10.1108/IJSSP-03-2022-0085.

[37] Y. Kütük, B. Güloğlu. Prediction of Transition Probabilities from Unemployment to Employment for Turkey via Machile Learning and Econometrics: A Comparative Study. Journal of Research in Economics (2019).

[38] N. H. Møller, I. Shklovski, T. T. Hildebrandt, in: ACM Other conferences, pp. 1–12. doi:10.1145/3419249.3420149.

[39] C. Mozzana. A Matter of Definitions: The Profiling of People in Italian Active Labour Market Policies. Historical Social Research (2019). doi:10.12759/hsr.44.2019.2.225-246.

[40] J. Niklas, K. Sztandar, K. Szymielewicz, Profiling the Unemployed in Poland: Social and Political Implications of Algorithmic Decision Making, Technical Report, Fundacja Panoptykon, 2015. URL: https://www.ohchr.org/sites/default/files/Documents/Issues/Poverty/DigitalTechnology/LSE_appendix2.pdf.

[41] Í. Martínez de Rituerto de Troya, L. O. Moraes, in: Proceedings of the 32nd Conference on Neural Information Processing Systems, pp. 1–6.

[42] C. Wan, R. Belo, L. Zejnilović, S. Lavado, in: Explainable Artificial Intelligence, pp. 181–197. doi:10.1007/978-3-031-44067-0_10.

[43] L. F. Zhao. Data-Driven Approach for Predicting and Explaining the Risk of Long-Term Unemployment. E3S Web of Conferences (2020). doi:10.1051/e3sconf/202021401023.

[44] I. Apostolopoulos, P. Groumpos. Fuzzy Cognitive Maps: Their Role in Explainable Artificial Intelligence. Applied Sciences (Switzerland) (2023). doi:10.3390/app13063412.

[45] M. Aslam, D. Segura-Velandia, Y. M. Goh. A Conceptual Model Framework for XAI Requirement Elicitation of Application Domain System. IEEE Access (2023). doi:10.1109/ACCESS.2023.3315605.

[46] P. Barnard, I. MacAluso, N. Marchetti, L. Dasilva, in: IEEE Int Conf Commun, volume 2022-May, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1530–1535. doi:10.1109/ICC45855.2022.9838766.

[47] S. Bayer, H. Gimpel, M. Markgraf. The role of domain expertise in trusting and following explainable AI decision support systems. Journal of Decision Systems (2022). doi:10.1080/12460125.2021.1958505.

[48] K. Brown, D. Talbert, in: Bartak R., Franklin M., Keshtkar F. (Eds.), Proc. Int. Fla. Artif. Intell. Res. Soc. Conf., FLAIRS, volume 35, Florida Online Journals, University of Florida, 2022, pp. 1–6. doi:10.32473/flairs.v35i.130662.

[49] F. Cau, H. Hauptmann, L. Spano, N. Tintarev. Effects of AI and Logic-Style Explanations on Users' Decisions Under Different Levels of Uncertainty. ACM Transactions on Interactive Intelligent Systems (2023). doi:10.1145/3588320.

[50] T. Chen, C. Guestrin, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. doi:10.1145/2939672.2939785.

[51] I. Christou, J. Soldatos, T. Papadakis, D. Gutierrez-Rojas, P. Nardelli, in: Proc. - Int. Conf. Distrib. Comput. Smart Syst. Internet Things, DCOSS-IoT, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 562–567. doi:10.1109/DCOSS-IoT58021.2023.00092.

[52] D. Cirqueira, M. Helfert, M. Bezbradica, in: Lect. Notes Comput. Sci., volume 12797 LNAI, pp. 21–40. doi:10.1007/978-3-030-77772-2_2.

[53] J. Collenette, K. Atkinson, T. Bench-Capon. Explainable AI tools for legal reasoning about cases: A study on the European Court of Human Rights. Artificial Intelligence (2023). doi:10.1016/j.artint.2023.103861.

[54] F. Da Silva Oliveira, F. De Lima Neto. Method to Produce More Reasonable Candidate Solutions With Explanations in Intelligent Decision Support Systems. IEEE Access (2023). doi:10.1109/ACCESS.2023.3250262.

[55] D. Dandolo, C. Masiero, M. Carletti, D. Dalle Pezze, G. Susto. AcME—Accelerated model-agnostic explanations: Fast whitening of the machine-learning black box. Expert Systems with Applications (2023). doi:10.1016/j.eswa.2022.119115.

[56] D. Das, B. Kim, S. Chernova, in: Int Conf Intell User Interfaces Proc IUI, Association for Computing Machinery, 2023, pp. 240–250. doi:10.1145/3581641.3584055.

[57] D. Delen, B. Davazdahemami, E. Rasouli Dezfouli. Predicting and Mitigating Freshmen Student Attrition: A Local-Explainable Machine Learning Framework. Information Systems Frontiers (2023). doi:10.1007/s10796-023-10397-3.

[58] K. Gajos, L. Mamykina, in: Int Conf Intell User Interfaces Proc IUI, Association for Computing Machinery, 2022, pp. 794–806. doi:10.1145/3490099.3511138.

[59] R. Galanti, M. de Leoni, M. Monaro, N. Navarin, A. Marazzi, B. Di Stasi, S. Maldera. An explainable decision support system for predictive process analytics. Engineering Applications of Artificial Intelligence (2023). doi:10.1016/j.engappai.2023.105904.

[60] M. Garouani, A. Ahmad, M. Bouneffa, M. Hamlich, G. Bourguin, A. Lewandowski. Towards big industrial data mining through explainable automated machine learning. International Journal of Advanced Manufacturing Technology (2022). doi:10.1007/s00170-022-08761-9.

[61] M. Heider, H. Stegherr, R. Nordsieck, J. Hähner. Assessing Model Requirements for Explainable AI: A Template and Exemplary Case Study. Artificial Life (2023). doi:10.1162/artl_a_00414.

[62] H.-W. Lee, T.-H. Han, T.-J. Lee. Reference-Based AI Decision Support for Cybersecurity. IEEE Access (2023). doi:10.1109/ACCESS.2023.3342868.

[63] Q. Liao, Y. Zhang, R. Luss, F. Doshi-Velez, A. Dhurandhar, in: Proc. AAAI. Conf. Hum. Comput. Crowdsourcing., Association for the Advancement of Artificial Intelligence, 2022, pp. 147–159. doi:10.1609/hcomp.v10i1.21995.

[64] H. Löfström, T. Löfström, U. Johansson, C. Sönströd. Investigating the impact of calibration on the quality of explanations. Annals of Mathematics and Artificial Intelligence (2023). doi:10.1007/s10472-023-09837-2.

[65] L. Malandri, F. Mercorio, M. Mezzanzanica, A. Seveso. Model-contrastive explanations through symbolic reasoning. Decision Support Systems (2024). doi:10.1016/j.dss.2023.114040.

[66] L. Mazzola, F. Stalder, A. Waldis, P. Siegfried, C. Renold, D. Reber, P. Meier, in: Commun. Comput. Info. Sci., Springer Science and Business Media Deutschland GmbH, 2021, pp. 431–438. doi:10.1007/978-3-030-78642-7_58.

[67] K. Mohiuddin, M. A. Alam, M. M. Alam, P. Welke, M. Martin, J. Lehmann, S. Vahdati, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 4752–4758. URL: https://dl.acm.org/doi/10.1145/3583780.3615497. doi:10.1145/3583780.3615497.

[68] A. Nguyen, S. Foerstel, T. Kittler, A. Kurzyukov, L. Schwinn, D. Zanca, T. Hipp, S. Jun, M. Schrapp, E. Rothgang, B. Eskofier. System Design for a Data-Driven and Explainable Customer Sentiment Monitor Using IoT and Enterprise Data. IEEE Access (2021). doi:10.1109/ACCESS.2021.3106791.

[69] D. Panagoulias, E. Sarmas, V. Marinakis, M. Virvou, G. Tsihrintzis, H. Doukas. Intelligent Decision Support for Energy Management: A Methodology for Tailored Explainability of Artificial Intelligence Analytics. Electronics (Switzerland) (2023). doi:10.3390/electronics12214430.

[70] D. Rojo, N. Htun, D. Parra, R. De Croon, K. Verbert. AHMoSe: A knowledge-based visual support system for selecting regression machine learning models. Computers and Electronics in Agriculture (2021). doi:10.1016/j.compag.2021.106183.

[71] J. Senoner, T. Netland, S. Feuerriegel. Using Explainable Artificial Intelligence to Improve Process Quality: Evidence from Semiconductor Manufacturing. Management Science (2022). doi:10.1287/mnsc.2021.4190.

[72] M. Shams, S. Gamel, F. Talaat. Enhancing crop recommendation systems with explainable artificial intelligence: a study on agricultural decision-making. Neural Computing and Applications (2024). doi:10.1007/s00521-023-09391-2.

[73] C. Steging, S. Renooij, B. Verheij, in: Proc. Int. Conf. Artif. Intell. Law, ICAIL, Association for Computing Machinery, Inc, 2021, pp. 235–239. doi:10.1145/3462757.3466059.

[74] F. Sufi, M. Alsulami. Automated Multidimensional Analysis of Global Events with Entity Detection, Sentiment Analysis and Anomaly Detection. IEEE Access (2021). doi:10.1109/ACCESS.2021.3127571.

[75] A. Thuy, D. Benoit. Explainability through uncertainty: Trustworthy decision-making with neural networks. European Journal of Operational Research (2023). doi:10.1016/j.ejor.2023.09.009.

[76] H. Tiensuu, S. Tamminen, E. Puukko, J. Röning. Evidence-based and explainable smart decision support for quality improvement in stainless steel manufacturing. Applied Sciences (Switzerland) (2021). doi:10.3390/app112210897.

[77] W. Dossche, S. Vansteenkiste, B. Baesens, W. Lemahieu. Interpretable and Accurate Identification of Job Seekers at Risk of Long-Term Unemployment: Explainable ML-Based Profiling. SN Computer Science (2024). doi:10.1007/s42979-024-02884-4.

[78] S. M. Lundberg, S.-I. Lee, in: NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, volume 2017 Proceeding of *NIPS'17*, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 4768–4777. doi:10.5555/3295222.3295230.

[79] M. T. Ribeiro, S. Singh, C. Guestrin, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, San Francisco California USA, 2016, pp. 1135–1144. doi:10.1145/2939672.2939778.

[80] J. H. Friedman. Greedy function approximation: A gradient boosting machine. The Annals of Statistics (2001). doi:10.1214/aos/1013203451.

[81] D. W. Apley, J. Zhu. Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models. Journal of the Royal Statistical Society Series B: Statistical Methodology (2020). doi:10.1111/rssb.12377.

[82] A. Inselberg. The plane with parallel coordinates. The Visual Computer (1985). doi:10.1007/BF01898350.

[83] E. Cambria, L. Malandri, F. Mercorio, M. Mezzanzanica, N. Nobani. A survey on XAI and natural language explanations. Information Processing & Management (2023). doi:10.1016/j.ipm.2022.103111.

[84] M. T. Ribeiro, S. Singh, C. Guestrin. Anchors: High-Precision Model-Agnostic Explanations. Proceedings of the AAAI Conference on Artificial Intelligence (2018). doi:10.1609/aaai.v32i1.11491.

[85] G. Brewka, S. Ellmauthaler, H. Strass, J. P. Wallner, S. Woltran. Abstract Dialectical Frameworks. An Overview. IFCoLog Journal of Logics and Their Applications (2017).

[86] D. R. Schlegel, S. C. Shapiro. Inference Graphs: A Roadmap. Second Annual Conference on Advances in Cognitive Systems (2013).

[87] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. PLOS ONE (2015). doi:10.1371/journal.pone.0130140.

[88] Y. Gil, J. Honaker, S. Gupta, Y. Ma, V. D'Orazio, D. Garijo, S. Gadewar, Q. Yang, N. Jahanshad, in: Proceedings of the 24th International Conference on Intelligent User Interfaces, ACM, Marina del Ray California, 2019, pp. 614–624. URL: https://dl.acm.org/doi/10.1145/3301275.3302324. doi:10.1145/3301275.3302324.

[89] D. Han, Z. Wang, W. Chen, Y. Zhong, S. Wang, H. Zhang, J. Yang, X. Shi, X. Yin, in: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pp. 3197–3217. doi:10.1145/3460120.3484589.

[90] F. D. Davis. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly (1989). doi:10.2307/249008.

[91] V. Chen, Q. Liao, J. Wortman Vaughan, G. Bansal, in: Proceedings of the ACM on Human-Computer Interaction, pp. 1—-32. doi:10.1145/3610219.

[92] I. Kolyshkina, S. Simoff, in: Data Mining, pp. 156–170. doi:10.1007/978-981-15-1699-3_13.

[93] M. I. Jordan, T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. Science (2015). doi:10.1126/science.aaa8415.

[94] A. K. Jain, M. N. Murty, P. J. Flynn. Data clustering: a review. ACM Computing Surveys (1999). doi:10.1145/331499.331504.

[95] J. H. Ward Jr. Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association (1963). doi:10.1080/01621459.1963.10500845.

[96] Scikit-learn, 2.3. Clustering, 2024. URL: https://scikit-learn/stable/modules/clustering.html, accessed: June 2024.

[97] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, X. Xu. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. ACM Trans. Database Syst. (2017). doi:1author={Schubert,ErichandSander,JÃűrgandEster,MartinandKriegel,HansPeterandXu,Xiaowei},0.1145/3068335.

[98] R. J. G. B. Campello, D. Moulavi, A. Zimek, J. Sander. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. ACM Transactions on Knowledge Discovery from Data (2015). doi:10.1145/2733381.

[99] L. McInnes, J. Healy, in: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 33–42. doi:10.1109/ICDMW.2017.12.

[100] L. Breiman, J. Friedman, R. A. Olshen, C. J. Stone, Classification and Regression Trees, Chapman and Hall/CRC, New York, 1984. doi:10.1201/9781315139470.

[101] L. Breiman. Arcing Classifiers. The Annals of Statistics (1998). URL: http://www.jstor.org/stable/120055.

[102] L. Breiman. Random Forests. Machine Learning (2001). doi:10.1023/A:1010933404324.

[103] C. Bentéjac, A. Csörgő, G. Martínez-Muñoz. A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review (2021). doi:10.1007/s10462-020-09896-5.

[104] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, A. Gulin, CatBoost: unbiased boosting with categorical features, 2019. doi:10.48550/arXiv.1706.09516.

[105] D. M. Allen. The relationship between variable selection and data agumentation and a method for prediction. Technometrics (1974). doi:10.1080/00401706.1974.10489157.

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

[107] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics (1987). doi:10.1016/0377-0427(87)90125-7.

[108] T. Caliński, J. Harabasz. A dendrite method for cluster analysis. Communications in Statistics (1974). doi:10.1080/03610927408827101.

[109] D. L. Davies, D. W. Bouldin. A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence (1979). doi:10.1109/TPAMI.1979.4766909.

[110] R. Gupta, S. Subedi, A. Singh, S. K. Singh, in: 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1054–1059. doi:10.23919/INDIACom61295.2024.10498443.

[111] C. E. Metz. Basic principles of ROC analysis. Seminars in Nuclear Medicine (1978). doi:10.1016/S0001-2998(78)80014-2.

[112] N. Chinchor, in: Proceedings of the 4th conference on Message understanding, MUC4 '92, Association for Computational Linguistics, 1992, pp. 22–29. doi:10.3115/1072064.1072067.

[113] G. W. Brier. Verification of Forecasts Expressed in Terms of Probability. Monthly Weather Review (1950). doi:10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

[114] J. C. Gower. A general coefficient of similarity and some of its properties. Biometrics (1971). doi:10.2307/2528823.

[115] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2623–2631. doi:10.1145/3292500.3330701.

[116] W. G. Cochran. The χ^2 test of goodness of fit. The Annals of Mathematical Statistics (1952). doi:10.1214/aoms/1177729380.

A

APPENDIX A: OUTLINE OF THE JSON MAPPING


```
{
  "id_features": [
    "stes_id_avam"
  ],
  "remove_categories": {
    "anmeldungsart_id": [
      "NA"
    ],
    "arbeits_zeit_id": [
      "NA"
    ],
    "ausbildungs_niveau_id": [
      "NA"
    ],
    "berufs_funktion_id": [
      "NA"
    ],
    "ao_alter": [
      "NA"
    ],
    "bezirksnummer": [
      "NA"
    ],
```

```
    "mobilitaet_id": [
      "NA"
    ]
  },
  "intact_categorical_features": [
    "bezeichnung_m_de"
  ],
  "intact_numerical_features": [
    "abgerechnete_tgg_total"
  ],
  "branche_features": [
    "noga_code",
    "noga_code1"
  ],
  "categorical_mapping": {
    "anmeldungsart_id": {
      "mapping": {
        "1288": "Erstanmeldung",
        "1289": "Wiederanmeldung innerhalb 6 Monate",
        "1290": "Wiederanmeldung laenger 6 Monate",
        "1291": "RAV-Wechsel"
      },
      "na_value": "NA"
    },
    "arbeits_zeit_id": {
      "mapping": {
        "587": "ganztags",
        "588": "vormittags",
        "589": "nachmittags",
        "590": "stundenweise",
        "591": "einzelne Tage",
        "592": "abends"
      },
      "na_value": "NA"
    },
    "aufenthalt_status_id": {
      "mapping": {
        "-1": "A",
        "594": "B",
        "9678": "B",
        "595": "B",
```

```
      "597": "C",
      "596": "C",
      "598": "C",
      "599": "C",
      "9384": "C",
      "600": "F",
      "9679": "F",
      "602": "G",
      "1519": "K",
      "604": "L",
      "603": "L",
      "605": "N",
      "606": "S"
    },
    "na_value": "A"
  },
  "ausbildungs_niveau_id": {
    "mapping": {
      "9501": "Sek. I",
      "9502": "Sek. I",
      "9503": "Sek. I",
      "9504": "Sek. II",
      "9505": "Sek. II",
      "9506": "Sek. II",
      "9507": "Sek. II",
      "9508": "Sek. II",
      "9509": "Sek. II",
      "9512": "Ter. Hochschulen",
      "9513": "Ter. Hochschulen",
      "9516": "Ter. Hochschulen",
      "9511": "Ter. Hoehere Berufsbildung",
      "9510": "Ter. Hoehere Berufsbildung",
      "9514": "Ter. Hochschulen",
      "9515": "Ter. Hochschulen"
    },
    "na_value": "NA"
  },
  "berufs_abschluss_id": {
    "mapping": {
      "701": "keiner",
      "702": "inlaendisch",
```

```
      "703": "auslaendisch"
    },
    "na_value": "keiner"
  },
 "berufs_erfahrung_id": {
    "mapping": {
      "-1": "keine Erfahrung",
      "706": "keine Erfahrung",
      "707": "weniger als 1 Jahr",
      "708": "1 - 3 Jahre",
      "710": "mehr als 3 Jahre"
    },
    "na_value": "keine Erfahrung"
 },
 "berufs_funktion_id": {
    "mapping": {
      "713": "Fachfunktion",
      "714": "Hilfsfunktion",
      "712": "Kaderfunktion",
      "715": "Lehrling",
      "9006": "Praktikant",
      "718": "Student",
      "711": "selbstaendigerwerbend",
      "717": "Schueler",
      "716": "Heimarbeit"
    },
    "na_value": "NA"
 },
 "bezirksnummer": {
    "mapping": {
      "246": "Verwaltungskreis Bern-Mittelland",
      "242": "Verwaltungskreis Biel/Bienne",
      "245": "Verwaltungskreis Emmental",
      "249": "Verwaltungskreis Frutigen-Niedersimmental",
      "250": "Verwaltungskreis Interlaken-Oberhasli",
      "244": "Verwaltungskreis Oberaargau",
      "248": "Verwaltungskreis Obersimmental-Saanen",
      "243": "Verwaltungskreis Seeland",
      "247": "Verwaltungskreis Thun",
      "241": "Arrondissement administratif Jura bernois"
    },
```

```
    "na_value": "NA"
  },
  "iv_code_id": {
    "mapping": {
      "1629": "nicht IV-Bezueger",
      "1630": "IV-Grad less than 70%",
      "1631": "IV-Grad less than 70%",
      "1518": "IV-Grad less than 70%",
      "1642": "IV-Grad less than 70%",
      "1643": "IV-Grad less than 70%",
      "1644": "IV-Grad 70 - 100%"
    },
    "na_value": "nicht IV-Bezueger"
  },
  "mobilitaet_id": {
    "mapping": {
      "1172": "nicht mobil",
      "1173": "Tagespendler",
      "1174": "in Teilen der Schweiz",
      "1175": "in der ganzen Schweiz",
      "1176": "auch ins Ausland"
    },
    "na_value": "NA"
  },
  "qualifikation_id": {
    "mapping": {
      "-1": "ungelernt",
      "1224": "gelernt",
      "1225": "angelernt",
      "1226": "ungelernt"
    },
    "na_value": "ungelernt"
  },
  "sprache_id": {
    "mapping": {
      "1387": "CH-Deutsch",
      "1401": "Deutsch",
      "1386": "Other",
      "1367": "Tuerkisch",
      "1365": "Spanisch",
      "1366": "Portugiesisch",
```

```
"1402": "Franzoesisch",
"1388": "Slavic",
"1397": "Other",
"1385": "Albanisch",
"1370": "Slavic",
"1384": "Tamil",
"1403": "Italienisch",
"1375": "Arabisch",
"1395": "Other",
"1377": "Slavic",
"9578": "Other",
"1393": "Slavic",
"1369": "Other",
"9218": "Other",
"1389": "Slavic",
"1372": "Slavic",
"1390": "Slavic",
"1400": "Other",
"1391": "Slavic",
"1364": "Other",
"1382": "Slavic",
"1368": "Other",
"1371": "Slavic",
"1373": "Other",
"9978": "Slavic",
"9646": "Other",
"1399": "Other",
"9818": "Other",
"1381": "Slavic",
"9579": "Other",
"9798": "Other",
"1380": "Other",
"1398": "Other",
"1376": "Other",
"9658": "Other",
"1379": "Other",
"1378": "Other",
"1383": "Other",
"1363": "Other",
"1374": "Other",
"1394": "Other",
```

```
      "1392": "Other",
      "1396": "Other",
      "9998": "Other"
    },
    "na_value": "Other"
  }
},
"keep_few_mapping": {
  "bezirksnummer": {
    "keep": [
      "246",
      "242",
      "245",
      "249",
      "250",
      "244",
      "248",
      "243",
      "247",
      "241"
    ],
    "replace_value": "NA"
  }
},
"numerical_categories": {
  "ao_alter": {
    "values": [
      15,
      20,
      25,
      30,
      35,
      40,
      45,
      50,
      55,
      60,
      80
    ],
    "labels": [
      "16 bis 20",
```

```
      "21 bis 25",
      "26 bis 30",
      "31 bis 35",
      "36 bis 40",
      "41 bis 45",
      "46 bis 50",
      "51 bis 55",
      "56 bis 60",
      "61 bis 65"
    ],
    "nan_label": "NA"
  },
  "hoechstanspruch": {
    "values": [
      -10,
      0,
      145,
      230,
      330,
      460,
      580,
      1000
    ],
    "labels": [
      0,
      90,
      200,
      260,
      400,
      520,
      640
    ],
    "nan_label": 0
  },
  "vermittlungs_grad": {
    "values": [
      0,
      50,
      79,
      100
    ],
```

```
      "labels": [
        "less than 50%",
        "51 - 79%",
        "80 - 100%"
      ],
      "nan_label": "less than 50%"
    }
  }
}
```