



SAPIENZA
UNIVERSITÀ DI ROMA

Explaining Machine Learning Models with Subgroup Analysis

Faculty of Information Engineering, Informatics, and Statistics
Department of Statistical Sciences
PhD Thesis in Methodological Statistics

Flavia Tagliafierro

ID number 951271

Supervisor

Prof.^{ssa} Donatella Firmani

Academic Year 2025/2026

Thesis defended on May 2026
in front of a Board of Examiners composed by:
Prof.ssa Daniela Marella (chairman)
Prof. Pierfrancesco Alaimo Di Loro
Prof.ssa Maura Mezzetti

Explaining Machine Learning Models with Subgroup Analysis
Sapienza University of Rome

© 2026 Flavia Tagliaferro. All rights reserved

This thesis has been typeset by L^AT_EX and the Sapthesis class.

Author's email: flavia.tagliaferro@uniroma1.it

*Dedicated to my family and my supervisor, who always supported and encouraged
me, even during difficult times*

Abstract

Explanations in classification models play an important role in making machine learning methods more transparent, particularly those models that are not clearly defined, such as black-box models. The algorithm described in this thesis, belonging to the class of Automatic Subgroup Detection techniques (ASD), aims at giving an insight into the way results are obtained. ASD methods try to discover population subsets where model behavior is significantly different from the baseline. While traditional ASD methods often focus on identifying problematic subgroups, such as those with high divergence in false positive rates, this research introduces a novel non-perturbative method specifically designed to identify the largest possible data subsets that achieve an accuracy exceeding a predefined threshold. The methodology extends the class of ASD techniques by providing the ability to identify subspaces rather than subgroups, defining these regions through feature ranges for both numerical and categorical variables. This algorithm is particularly effective at identifying subsets that respond best to specific actions, ensuring a predefined accuracy threshold is maintained for prediction tasks. Specifically, the methodology focuses on finding subspaces characterized by a significantly better or worse performance in terms of classification accuracy without requiring the transformation of numerical attributes. The proposed algorithm can be evaluated against others in the same class, despite significant methodological differences. The benchmarking process involves comparing, across various datasets, execution times and subset cardinality, while maintaining consistent levels of accuracy and support. To delve deeper into our method, we create a grid search space to decompose the algorithmic complexity and present different versions of the algorithm designed to operate within each specific data partition. Across all experiments, the algorithm is evaluated to demonstrate its effectiveness in identifying valid subspaces as the accuracy threshold becomes increasingly stringent. This research was inspired by practical needs arising from the analysis of criminal recidivism data and, more broadly, from the examination of judicial records. The argument could serve as a primary application context for the proposed methodology and is examined in depth in a dedicated chapter of this dissertation.

Contents

1	Introduction	1
1.1	Overview of the Thesis and Main Contributions	2
1.2	Background and Motivations of the Thesis	3
2	Explainability in the Presence of Feature Interactions	4
2.1	Model Explainability	4
2.2	Classification Models and Subgroup Representativeness	5
2.2.1	Subspace Analysis and the Notion of Coverage	6
2.3	Metrics and Validation of Subspace Representations	6
2.4	The Explanation Methods: the Lens of Transparent Models	7
3	Related Work	9
3.1	Automatic Subgroup Detection Techniques	10
3.2	Inadequate Representation Methods	12
3.3	Explainable AI Domain Techniques	13
4	Subspace Classification OPTimization and Explanation	16
4.1	Preliminaries and Problem Definition	16
4.2	Subspace Classification OPTimization and Explanation	17
4.2.1	The Mixed Integer Optimization Problem	18
4.2.2	Data Pre-Processing for the Subspace Optimization Problem	19
4.3	The SCOPE Algorithm	20
4.3.1	Introduction	20
4.3.2	Demonstration Scenarios	23
4.3.3	Conclusions	24
5	The SCOPE Grid Algorithm	26
5.1	Improving Efficiency with Grid Search	26
5.2	Experimental Setup	28
5.3	Comparison of SCOPE and DivExplorer	29
5.4	Technical Details about the Multidimensional Grid Search Space	32
5.5	Application of TeraStat2 for Job Execution with Large Datasets	35
5.6	Analysis of the Time Complexity and Efficiency of Scope	36
5.7	Algorithms in Action	37

6	Administrative Datasets and Evidence Based Decision Making in the Public Sector	44
6.1	Model Explainability and Pattern Discovery in Judicial Records . . .	44
6.2	Predicting modeling. A Critical Approach to Compas Recidivism Classification	46
6.3	Predictive Models for Criminal Recidivism	47
6.3.1	Introduction	48
6.3.2	Data and Methods	48
6.3.3	Outcomes	50
6.3.4	Conclusions and Future Perspective	53
7	Conclusions and Future Work	57
7.1	Conclusions	57
7.2	Limitations and Future Work	59
	Bibliography	61
	Acknowledgments	64
	Appendix	65
.1	Entropy of the Clustering Scheme	65
.2	Main Tables	65

List of Figures

2.1	Quality metrics by classes for the optimal subspace identified by our algorithm on the Titanic dataset with threshold for accuracy = 0.95.	7
4.1	Definition of the Subspace Optimization Problem for finding the largest subset of instances with accuracy $\geq \beta$	18
4.2	High-accuracy subspace in two dimensions. The space is separated in two regions according to the green line. Data points to the left are classified as red (+) and data points to the right are classified as blue (-). The ground-truth class of each instance is represented by the color of each point.	21
4.3	Main demonstration functionalities.	22
5.1	Scope optimization when applied to a grid search: accuracy level and hyper-rectangular subspaces.	27
5.2	The percentage of points in two Automatic Subgroup Detection techniques for the Titanic, Compas and Credit dataset: a comparison of Scope (standard, Grid Sum, Grid Max) and DivExplorer.	31
5.3	Accuracy and size of the cells (grid search) for the Titanic, Compas and Credit dataset.	33
5.4	The execution times in two Automatic Subgroup Detection techniques for the Titanic, Compas and Credit dataset: a comparison of Scope (standard, GS SERIAL, GM PARALLEL) and DivExplorer.	34
5.5	The percentage of points across the Titanic, Compas and Credit dataset: a comparison of classification models in Scope.	36
5.6	Shapley item contributions of the second divergent pattern (left) and overall impact of an item for the Titanic dataset (right) with support = 0.10.	39
5.7	SHAP values of the individuals in the optimal subspace found with Scope for the Titanic dataset and threshold for accuracy = 0.95 (left) and SHAP values for the entire dataset (right).	39
5.8	Shapley item contributions of the first divergent pattern (left) and overall impact of an item for the Compas dataset (right) with support = 0.10.	41
5.9	SHAP values of the individuals in the optimal subspace found with Scope for the Compas dataset and threshold for accuracy = 0.85 (left) and SHAP values for the entire dataset (right).	42

5.10	Shapley item contributions of the first divergent pattern (left) and overall impact of an item for the Credit dataset (right) with support = 0.10.	43
5.11	SHAP values of the individuals in the optimal subspace found with Scope for the Credit dataset and threshold for accuracy = 0.95 (left) and SHAP values for the entire dataset (right).	43
6.1	SHAP values of the individuals in the optimal subspace found with Scope for the Recidivism dataset and threshold for accuracy = 0.85 (left) and SHAP values for the entire dataset (right).	46
6.2	Classification report, main metrics for each class (negative, positive). Logistic Regression, Neural Networks, Random Forest models; COMPAS dataset.	51
6.3	Feature importance based on the mean absolute SHAP values of the Random Forest model for the COMPAS dataset.	52
6.4	Confusion matrix: comparison between actual (rows) and predicted values (columns). Logistic Regression, Neural Network and Random Forest models; COMPAS dataset.	53
6.5	Silhouette score and number of clusters for COMPAS dataset.	54
6.6	Accuracy of the Random Forest model calculated for each cluster identified using the K-Means method ($k = 9$) on the COMPAS dataset.	55
6.7	Classification report, main metrics for each class, cluster n.3 (left side) and whole dataset COMPAS (right side) based on the Random Forest model.	55
6.8	Random Forest model's confusion matrix. Comparison between actual (rows) and predicted values (columns). Cluster n.3 (left) and the COMPAS dataset (right).	56
6.9	Random Forest model's SHAP values for cluster n.3 (left) and COMPAS dataset (right).	56

List of Tables

5.1	Qualitative comparison of two Automatic Subgroup Detection techniques: Scope and DivExplorer.	31
5.2	Top 5 divergent patterns with respect to accuracy for the Titanic dataset with support = 0.10 and redundancy pruning.	37
5.3	Optimal subspace obtained with Scope for the Titanic dataset, with threshold for accuracy = 0.95.	38
5.4	Top 5 divergent patterns with respect to accuracy for the Compas dataset with support = 0.10 and redundancy pruning.	40
5.5	Optimal subspace obtained with Scope for the Compas dataset, with threshold for accuracy = 0.85.	41
5.6	Top 5 divergent patterns with respect to accuracy for the Credit dataset with support = 0.10 and redundancy pruning.	42
5.7	Optimal subspace obtained with Scope for the Credit dataset, with threshold for accuracy = 0.95.	43
.1	A comparison of number of points (percentage) in two automatic subgroup detection techniques: Scope (standard, Grid Sum, Grid Max) and DivExplorer.	66
.2	A comparison of execution times (seconds) in two Automatic Subgroup Detection techniques: Scope (standard, GS SERIAL, GM PARallel) and DivExplorer.	67

Chapter 1

Introduction

Classification models have become ubiquitous in a wide range of applications, from healthcare to finance. These models, powered by sophisticated algorithms, can play a pivotal role in decision-making processes. The standard de-facto metric for assessing model performances is *global accuracy*, that is, the fraction of correctly classified data points on a test set. However accuracy can vary widely depending on the characteristics of a certain dataset, and there are two main reasons for which global accuracy alone can be misleading: high global accuracy and modest global accuracy. Intuitively, breaking down global accuracy would build confidence in using the model for real-world decisions. Knowing where the model excels and where it does not would allow users to make informed decisions about its applicability and limits, guiding data collection, feature engineering, and model tuning.

SCOPE (Subspace Classification OPTimization and EXplanation) introduces a novel approach to the field of Automatic Subgroup Detection techniques (ASD). By solving a mixed integer optimization problem, it defines hyper-rectangular subspaces rather than discrete itemsets, a collection of one or more items that appear together in a dataset, eliminating the need to discretize numerical features. This algorithm is particularly effective at identifying subsets that respond best to specific actions, ensuring a predefined accuracy threshold is maintained for prediction tasks. The primary objective of our algorithm is to identify accuracy variations across different subspaces within the dataset, moving beyond the standard approach of evaluating the dataset as a monolithic set. Its goal is to identify the largest possible subsets of data points that achieve an accuracy value exceeding a predefined threshold, regardless of the specific prediction model used for the task.

Scope can be considered as a detection method for problematic high accuracy subspaces, capable of discovering subspaces with accuracy higher than a given threshold. However, analyzing these partitions is crucial to understand the drivers of such high accuracy. Indeed, this performance may stem from a substantial number of both positive and negative predictions relative to the total population, as would be expected in a robust model. However, it may also result from biased behavior of the classifier. For example, an imbalance in the true labels within a subgroup can lead to biased training in that subgroup, resulting in a prevalence of predictions for the majority class. In this situation, the recall for the majority class is higher than that for the minority class, and, as a consequence, the overall accuracy is also high

even if the minority class is not correctly predicted.

1.1 Overview of the Thesis and Main Contributions

The primary contribution of our algorithm is the definition of the range query, which offers a compact, human-readable representation for distinguishing data points; specifically, those within the defined ranges exhibit high accuracy, while those outside signify low accuracy. SCOPE, standing for Subspace Classification OPTimization and EXplanation, serves as an interpretability tool, as the identification of high accuracy ranges effectively maps and explains a model's underlying classification patterns, also highlighting the feature importance in the high accuracy area.

The thesis is organized as follows. Some of the results have already been published or are currently undergoing peer review in international conference proceedings and journals.

Chapter 2 introduces the fundamental principles of explainability and the issues of classification models in learning from data with partially represented interactions and propose a novel method (at the core of this thesis) to explore subgroups in which predictive accuracy varies relative to the baseline, aimed at explaining the different behavior of classifiers across subgroups in classification tasks.

Chapter 3 describes several model-agnostic algorithms sharing similarities with our method and generally falling under the category of "Automatic Subgroup Detection techniques". Other approaches focus on uncovering lack of adequate coverage and underrepresented groups within a dataset. However, the identification of interesting data subgroups in these "Inadequate representation methods" is independent of a specific classification task. The "Explainable AI domain techniques" encompass models that aim to identify the factors influencing individual predictions for each feature.

Chapter 4 introduces a Subspace Optimization Problem (SOP) formulated with a *mixed-integer mathematical program* as the problem of finding the subspace containing the largest number of points and respecting a minimum level of accuracy, decided by the user. The algorithm is defined as Scope (Subspace Classification and OPTimization and EXplanation). This chapter presents the article "Breaking Down Accuracy with Subspace Optimization"⁽⁵⁾, published in the International Conference on Extending Database Technology Proceedings in 2024, introducing the Python toolkit for identifying high accuracy portions of the test dataset.

Chapter 5 presents the Scope Grid algorithm, the optimized version of Scope, and an experimental comparative analysis with the Scope algorithm described in Section 4.2 and the DivExplorer algorithm described in Section 3.1. These results are included in the paper "Efficient Explanation of Classification Models with Accuracy Ranges Optimization", that is under review at the Journal of Data and Information Quality in 2026. This chapter evaluates our algorithm comparing it with other similar methods in the same class to find analogies and verify performance in different datasets. To delve deeper into this method, a grid search space is created to decompose the algorithmic complexity and achieve efficiency and scalability. Afterward, a test to demonstrate Scope's efficiency in finding subspaces as the accuracy threshold increases is executed, calculating both the execution times and

the percentage of data points in the subset. Subsequently, a description of the High-Performance Computing (HPC) cluster supporting the computational challenges in the optimization process is provided. Finally, a case study is presented illustrating the different approaches to feature contribution and subgroup definition in Scope and DivExplorer.

Chapter 6 describes how processing Administrative Datasets for statistical analysis or classification tasks can therefore be demanding. Nevertheless, recent advances in data-driven methodologies have made it increasingly feasible to address these challenges. Despite this progress, great care must be taken to properly understand and explain analytical outcomes in order to avoid misinterpretations or, even worse, unintended consequences. This chapter presents the article "Criminal Recidivism. Towards Reliable and Transparent Predictive Models"⁽¹⁾ published on the Italian Journal of Economic, Demographic and Statistical Studies in 2026.

Chapter 7 presents the main conclusions and future research directions arising from this dissertation.

1.2 Background and Motivations of the Thesis

I am a senior statistician at the Ministry of Justice and my interest in exploratory analysis, explainability, and solving data-driven problems stems from my experience in producing official statistics and in supporting data communication processes. I leverage data science and statistical methodologies to derive meaningful insights and support evidence-based decision-making, particularly using administrative data. My expertise lies in prison statistics and research on criminal recidivism, quality of life within prisons and overcrowding issues.

My research focus is motivated by the practical necessity of experimenting with new data-driven techniques to explore administrative archives, aiming to identify key information and the underlying relationships worthy of investigation. Furthermore, in this phase of expanding AI applications within the public sector, it is essential to govern processing workflows and output generation mechanisms. This ensures the ability to identify methodological weaknesses and provide a sound rationale for data-driven conclusions.

A primary example in this field is the analysis of criminal recidivism: while other countries employ AI tools to support sentencing or the granting of alternative measures, these are not used for punitive purposes in our country. However, one could envision their application for rewarding or rehabilitative purposes, or simply as predictive tools to estimate trends within the phenomenon.

Chapter 2

Explainability in the Presence of Feature Interactions

This chapter defines the most important concepts of Explainability and introduces the novel method at the core of this thesis, aimed at explaining the different behavior of classifiers across subgroups in classification tasks. Understanding feature interactions is essential for Explainability, to characterize their contribution to predictive outcomes, as model behavior not only depends on additive effects between features. This method is able to compute high accuracy ranges in order to interpret the model's classification patterns.

2.1 Model Explainability

Explainability focuses on the outputs of the classification process by providing explanations of the results after training (post hoc). In the presence of sophisticated models such as Deep Learning, often referred to as black boxes, explanations are fundamental for understanding why a model makes a particular decision, thereby guessing the underlying reasoning and logic of the process. Explainability differs from interpretability in that interpretability refers to models whose internal structure and decision-making process are inherently transparent, thus not requiring post hoc explanations. In general, interpretability is often associated with a trade-off in predictive accuracy, which is why more complex black-box models are frequently preferred. Consequently, explainability techniques become necessary to support users by providing insights into the decisions made by these complex models. Many tools addressing this challenge are already available in the literature and are discussed in the following chapters.

However, this thesis introduces a different perspective on explainability by focusing on the varying behavior of classifiers when applied to subgroups, specifically subspaces defined by feature interactions. This approach enables the identification of subspaces characterized by distinct feature interactions and allows for the discovery of cases in which test set accuracy is higher or lower than a predefined threshold β .

2.2 Classification Models and Subgroup Representativeness

Representation Bias happens when the training data under-represents some parts of the target population and fails to generalize well to them⁽⁹⁾: it can be identified and resolved in structured and unstructured types of data from a data-centric point of view. Historical Bias, due to the socio-technical issues in the world, underlying distribution skew and sampling/selection bias are the origins of under-representation. Selection bias is introduced to the data when one fails to ensure proper randomization in selecting people, groups, or tuples of data for analysis.

Some metrics are used as representation indicators, for example Representation Rate, a metric defined to identify representation bias as the base rate. Base rate, also known as “prior probability”, refers to the class probability unconditioned on any observation. An equal base rate is defined as having an equal number of objects for different subgroups in the data set. In other words, the objects in the selected set should have an equal chance of belonging to each subgroup. Another metric to measure representation bias is Data Coverage, which at a high level refers to having enough similar entries for each object in a data set.

Representation indicators are often defined for structured data. Depending on the type of the attributes of interest, the techniques can be divided into two classes based on whether they target the problem for nominal or ordinal/numerical attributes. In the discrete space of attributes of interest representation bias can be defined using the discrete notion of coverage measure. For cases where attributes of interest are non-ordinal categorical, coverage is defined as having “enough” entries in the data set matching a particular pattern. A pattern is a string that specifies a subgroup that matches possible values over a subset of attributes of interest. Coverage is usually discussed for groups given by the conjunction of attribute-value assignments. A constant value is considered as the threshold for coverage, meaning that a minimum number of entries equal to the threshold value should exist from a subpopulation to be covered. In discrete data sets, there are multiple attributes each having multiple possible values that form a combinatorial number of possible patterns. Since patterns are the combination of some or all attributes-values, they can have multiple children.

Binning can transform the continuous space into discrete. However, it may lead to coarse groupings that are sensitive to the thresholds chosen. Techniques in this category assume data with continuous-valued attributes and propose solutions for identifying representation bias in such data sets. The notion of coverage can be extended to a continuous space for identifying representation bias. The problem of identifying representation bias using the continuous notion of coverage is defined as follows: given a data set D with tuples and attributes, a vicinity radius and a coverage threshold identify the uncovered region. A query point in a continuous data space is covered if there are enough data points in its vicinity neighborhood (the circle centered at the query point with a given radius). The uncovered region is demarcated by the collection of all the uncovered query points in the space.

2.2.1 Subspace Analysis and the Notion of Coverage

The representation problem is evaluated as a primary key for understanding the classification behavior of models in data subgroups. The focus of this research is the identification of the largest subset of instances having an accuracy above the threshold: in this way data coverage, as introduced in this section, is ensured in order to have enough similar entries in the subspace and to avoid under representation. This is an obvious requirement to obtain good classification results, even if accuracy is higher than the threshold: the subspace has to be representative in order to generalize well to the target population.

When dealing with subspaces, the notion of coverage is inherently different from the methods described in Chapter 3, where itemsets are represented as the combination of some or all attribute-values: the notion of coverage in this case refers to a continuous space. A representation of a continuous space is described in Section 2.2, where a data point is analyzed within its local neighborhood (the circle centered at the query point with a given radius). In this thesis, however, a different approach is adopted: subspaces are defined through specific feature ranges, which identify hyper-rectangular areas containing the maximum number of instances.

Subspaces can be simply created by finding the best lower and upper bounds. In the following chapters, we detail our algorithm and its theoretical foundations, emphasizing their non-perturbative nature and their capacity to find exact solutions. Our approach centers on global optimization and exact analytical derivation to identify the true global maximum and ensure an exact solution of the problem.

2.3 Metrics and Validation of Subspace Representations

In the algorithm introduced in this thesis, a subspace is identified by upper and lower bounds for the features: the classification performance within this continuous space can be analyzed using usual metrics as recall and precision for the class values. Label balance (the proportion of the classes) can also be considered, as it can explain the amount of accuracy. Ground-truth labels within a subgroup can be imbalanced even when the labels in the overall dataset are balanced. This may be due to poor representation of the specific subgroup and can lead to improper behavior of the classifier in that partition. The implication is that the classifier performs well on the majority class but poorly on the minority class. The F1 score can confirm this hypothesis, if there are differences between F1 values of the classes. To evaluate subspace optimization, metrics and the imbalance of true labels can be helpful. Accuracy is useful if classes are balanced, but it can mask the importance of minority class, because it focuses on overall correctness rather than the ability to predict a specific class.

Recall and precision help in identifying high performance on the target feature: recall considers false negatives, while precision controls false positives. Optimizing for recall means a lower decision threshold and optimizing for precision implies a higher decision threshold (a trade off exists between recall and precision). In our examples the classes are quite balanced, so accuracy works well: we aim to find overall correctness but recall, precision and the F1 score can also be evaluated as indicators of the optimal subspace quality and representativeness. A classification report for

the subspace can be compared with the overall results to assess the classification task within that subgroup. For example, we might observe high accuracy alongside lower values of recall and precision for the target class, if it is not the majority class. This can depend on the high weight of the majority class in the subgroup, even if labels are balanced in the overall dataset, meaning that optimizing for accuracy can lead to selecting a subspace where only the majority class is well predicted.

Our algorithm can be considered as a detection method for problematic high accuracy subspaces. This method can discover subspaces with accuracy higher than a given threshold, but we need to analyze these partitions to understand why the accuracy is so high. In fact, this may depend, as we hope, on a large number of both positive and negative predictions relative to the total. However, it may also result from biased behavior of the classifier. For example, an imbalance in the true labels within a subgroup can lead to biased training in that subgroup, resulting in a prevalence of predictions for the majority class. In this situation, the recall for the majority class is higher than that for the minority class, and, as a consequence, the subgroup accuracy is high even if the minority class is not correctly predicted.

2.4 The Explanation Methods: the Lens of Transparent Models

Explainability in our method can be defined as an attempt to interpret a model's classification pattern. A fundamental in depth analysis involves examining subspaces and comparing their performance metrics to those of the overall reference dataset. This is crucial, because optimizing solely for overall accuracy is not the only measure of model quality. For instance, figure 2.1 shows a heatmap representing the metrics for the subspace obtained from the Titanic dataset through our approach, having accuracy higher than a threshold equal to 0.95 and the largest number of points. As we can observe, all metrics (F1 score, precision and recall) for this specific subgroup are superior to those of the overall reference dataset. This indicates that the identified subspace not only meets the accuracy threshold but also satisfies other quality objectives related to each target value.

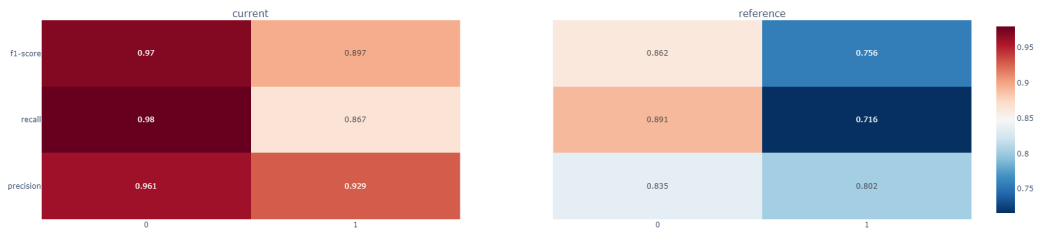


Figure 2.1 Quality metrics by classes for the optimal subspace identified by our algorithm on the Titanic dataset with threshold for accuracy = 0.95.

Furthermore we observe a disparity in performance across classes, likely due to the process preference for the majority class, as discussed in Section 2.3. Specifically

metrics for label 0 (the majority class) are higher than those for label 1 (the minority class) within this subspace.

Chapter 3

Related Work

This thesis presents novel automatic techniques designed to analyze interesting data subgroups. The goal is to identify the largest possible subsets of data points that achieve an accuracy value exceeding a predefined threshold, regardless of the specific prediction model used for the task.

While several model-agnostic algorithms share similarities with our methods, notable differences exist. These methods generally fall under the category of "Automatic Subgroup Detection techniques"^{(17) (11)}. Other approaches focus on uncovering lack of adequate coverage and underrepresented groups within a dataset⁽¹⁵⁾. However, the identification of interesting data subgroups in these "Inadequate representation methods" is independent of a specific classification task. The "Explainable AI domain techniques" encompass models that aim to identify the factors influencing individual predictions for each feature, such as Shapley values⁽²²⁾.

Key differences with the closest methods. A comparison between our algorithms and the closest methods, belonging to the class of the Automatic Subgroup Detection techniques (ASD), reveals distinct operational differences despite their shared objective of model-agnostic metric decomposition. Unlike other approaches, ours are non-perturbative methods specifically designed to identify the largest possible subsets of data points that achieve an accuracy value exceeding a predefined threshold, while other methods focus on the discovery of problematic subsets. For instance they identify a subgroup that exhibits the highest divergence in the false positive rate compared to the baseline, subject to a minimum support threshold.

The primary differences lie in the handling of numerical features and the role of the threshold. Our algorithms eliminate the need for discretization by directly generating high accuracy ranges from numerical data. Furthermore, the optimization logic is inverted: in our methods, accuracy serves as the primary constraint while the algorithms maximizes support. In contrast, traditional methods in this class utilize support as a threshold and subsequently rank the resulting patterns by accuracy. Ultimately, the novel algorithms described in this thesis provide users with a multidimensional subspace defined by continuous feature intervals, whereas the other methods produce itemsets that represent discrete subgroups of data points

3.1 Automatic Subgroup Detection Techniques

DivExplorer. In our opinion DivExplorer⁽¹¹⁾ is the method most closely to our algorithms. It analyzes the varying classification behavior within data subgroups relative to the overall behavior, based on different outcome functions (e.g., false positive rate and accuracy). The notion of divergence enables the estimation of differences between data subgroups relative to the overall behavior. Divergence exploration can help in finding the subgroups where the model achieves better performance. Itemset divergence can be defined as follows:

$$\Delta_f(I) = f(I) - f(D) \quad (3.1)$$

where an itemset I consists of attribute–value equalities (items), each associated with a distinct attribute, f is the outcome function and D the dataset. An item’s contribution to the divergence of an itemset is defined based on the Shapley value of a player within a coalition and represented as $\Delta(\alpha|I)$. The effect of an individual item α on classification can be estimated by adding it to other itemsets, providing a global measure. This definition is, once again, grounded in the concept of Shapley values. Owing to the computational complexity, an approximate definition of an item’s global contribution can be derived, obtaining the global divergence $\tilde{\Delta}^g(\alpha, s)$ depending on the support threshold chosen for analysis (s).

DivExplorer focuses only on relevant subsets comprising a sufficient portion of the dataset, considering only frequent itemsets as those whose support exceeds a given threshold s . Numerous methods exist in data mining for computing frequent itemsets and DivExplorer can utilize them.

DivExplorer requires a nominal scale for features to create itemsets, necessitating the discretization of numerical attributes, which can potentially affect the final results. H-DivExplorer⁽⁷⁾ offers a hierarchical approach to anomalous subgroup discovery, exploring the optimal granularity for each numerical attribute, but we do not consider it further at this time. Notably, with our algorithms, features do not require transformation during preprocessing and for nominal features we can perform a binary transformation of the categories (one-hot encoding).

For a fixed threshold, our methods yield a unique solution. In contrast, DivExplorer allows the exploration of all frequent itemsets with a support size above a given threshold, followed by redundancy pruning, and considers the top divergent patterns. The performance of DivExplorer is dependent on the chosen Frequent Pattern Algorithm (FPM); after extraction frequent itemsets can be ranked by divergence or support. The final output for our users is a multidimensional subspace defined by feature intervals, whereas DivExplorer provides itemsets corresponding to subgroups of data points.

While our approach shares a foundation in Shapley-based coalitional game theory with DivExplorer, the two methods diverge in application: we utilize it to identify factors influencing individual predictions, whereas DivExplorer employs it to estimate the contribution of specific attribute values to the subgroup divergence. This contribution, termed Global item divergence, quantifies the increase in divergence when the item is added to the model, capturing associated contributions.

Slice Finder. Slice Finder⁽¹⁷⁾ is similar to DivExplorer in that it identifies slices

of data points where performance is critical by comparing the classifier loss on the slice to the loss on the rest of the data. However, Slice Finder does not exhaustively search for slices, although it can produce overlapping and interpretable ones. It is an interactive framework that automatically identifies large data slices where the model performs poorly, defined as “problematic” slices. Two approaches for automated data slicing are considered: decision trees and lattice searching, which are more effective than simple clustering for finding data points with similar behavior.

A top-down lattice search identifies the top-k slices of interest, guided by statistical techniques that measure the significance and magnitude of performance discrepancy on subgroups. To identify large and interpretable subgroups, the search is halted if the considered data group is already statistically significant and the model performs sufficiently poorly. However, since the metrics used for assessing the model’s performance on subgroups are typically non-monotonic, the degree of discrepancy of a group does not indicate the behavior of its super/sub-groups. Slice Finder sorts slices by increasing number of literals, decreasing slice size and effect size (above an interactive threshold T) for finding the top k slices. The algorithm stops when the k slices with a sufficiently large effect size are found and no larger itemsets are considered.

Slice Finder and DivExplorer can be used for preliminary fairness analysis by identifying problematic slices. It is crucial for prediction models to perform equally well across all features to avoid discrimination. Our methods also enable users to find the optimal subspace with accuracy below the fixed threshold, effectively performing the same objective as a pre-processing step for fairness.

Other methods in this category include Errudite⁽¹⁴⁾, FairVis⁽¹⁶⁾ and Div-Rank⁽³⁾.

Errudite. Errudite facilitates data grouping for Natural Language Processing error analysis. It employs an expressive domain specific language and enables systematic grouping of relevant instances with filtering queries. It then supports the investigation of error causes through automated counterfactual rewriting. Errudite is an interactive system, with an interface for the analysis process and additional exploration support such as visualizing data distributions, suggesting potential queries and presenting the grouping and rewriting results. This can be highly beneficial for users to formulate and inspect their hypotheses in real time.

At the core of Errudite is an expressive domain-specific language (DSL) for precisely querying instances based on linguistic features. The DSL formalizes unambiguous error hypotheses, allows grouping to scale to all instances, and enables rewriting for counterfactual testing. For example, it simplifies the creation of a precise group containing all instances where the ground truth and the prediction share an entity type. Data groups and rewrite rules can be easily saved and shared for replication or for analysis of different models using the same groups and rules. Our work deals with structured data and we automatically slice the dataset based on the actual attribute domains.

FaiVis. FairVis is able to discover intersectional bias, a form of discrimination that arises from the interaction of multiple personal characteristics, such as age, race and gender. It is a visual analytics system that integrates a novel subgroup discovery technique to enable users to audit the fairness of machine learning models. Users can generate and investigate known subgroups and explore suggested and similar subgroups, ranging from a high-level overview of subgroup performance to a

detailed investigation. FairVis helps uncover biases in machine learning models and demonstrates the utility of interactive visualization in this task.

To generate suggested subgroups, FairVis uses a clustering-based technique, to identify groups with significant statistical similarity that can be described by a few dominant features using feature entropy. Performance metrics are evaluated on the identified clusters and presented to the user. In FairVis, suggested groups are created by clustering instances in the neighborhood using a distance-based algorithm such as K-means. A preprocessing step to transform nominal features into binary variables is required in order to compute distances, while this is not necessary for numerical or ordinal variables. Once a user identifies an interesting subgroup, exploring similar subgroups can reveal more general patterns. Similarity is defined in terms of statistical divergence, comparing how closely related groups are. Consequently, similarity between subgroups can be viewed as the statistical distance between their feature distributions.: the more values two subgroups share, the more similar we consider them. Unlike FairVis, our approach directly identifies data subgroups by slicing attribute domains, resulting in subspaces that are inherently interpretable. Clusters of similar instances can help identify variations in accuracy (i.e., subgroups with better predictions) depending also on imbalance in the target classes.

Div-Rank. Fairness requires “enough” individuals from protected groups in the top-k positions, for a ranking to be considered valid. Div-Rank addresses this issue by automatically identifying the disadvantaged groups in the data and mitigating their disparate representation in the final ranking. Subgroups with negative divergence experience a disadvantage and this method try to re-ranking instances to maximize the minimum subgroup divergence, while maintaining the new ranking as close as possible to the original one. Div-Rank is based on identifying the divergent subgroups and applying a re-ranking procedure with the goal of maximizing the minimum divergence.

3.2 Inadequate Representation Methods

DeepDiver. DeepDiver⁽¹⁵⁾ belongs to the class of methods for discovering a lack of adequate coverage in a dataset and underrepresented groups. Inadequate representation may cause errors in predictions and undesirable outcomes, such as algorithmic bias. Uncovered patterns are introduced to identify attribute space regions not adequately covered by the data, and data subgroups are identified by attribute value combinations. In DeepDiver the target attributes and classification outcome are not considered in the coverage problem, whereas we explicitly consider classification performance and identify subgroups in which a classification model performs differently with respect to the overall population. Unlike DeepDiver, we consider subgroups with adequate representation, selected by the objective function of the optimization problem.

DeepDiver analyzes the coverage of a given dataset across multiple categorical attributes, with efficient techniques for traversing the value combinations to identify any regions of attribute space not adequately covered by the data. Then it suggests the minimum amount of additional data that must be obtained to resolve this lack

of adequate coverage. DeepDiver formalizes the novel notion of maximal uncovered patterns (MUPs), to show the lack of coverage with regard to multiple attributes. The identification of MUPs can be solved in different ways, but efficient search and pruning strategies are available for it with DeepDiver, a search algorithm that tends to quickly find MUPs and use them to limit the search space by pruning the nodes both dominating and dominated by the discovered MUPs.

3.3 Explainable AI Domain Techniques

This class includes models for understanding the reasons behind model predictions, explaining the factors influencing an individual prediction for a given instance, like SHAP⁽²²⁾, LIME⁽²⁴⁾, LACE⁽¹⁹⁾ and Anchors⁽²¹⁾. This group also includes a novel non-perturbative method based on Relative Keys⁽⁴⁾. These explanations can be applied to classification across a diverse set of domains (tabular, text, and images). Individual prediction explanations can be used to select between models, alongside accuracy. In fact an algorithm with higher accuracy on the validation set can actually be much worse, a fact that is easily observed when explanations are provided, due to human prior knowledge, but difficult to discern otherwise. Furthermore, there is frequently a mismatch between the metrics that we can compute and optimize (e.g. accuracy) and the actual metrics of interest such as user engagement and retention.

SHAP. In SHAP explanation tool the Shapley value is used to compute the contribution of each feature value to the prediction for a single instance, in LIME instances are weighted according to their proximity to x (they have a high weight if near or a low weight if far away from x). Although the original model may be too complex to explain globally, LIME presents a linear explanation that is locally faithful. Anchors exhibit high precision and clear coverage, crucial for interpretable explanations of a model’s local behavior. Anchors creates rule-based model-agnostic explanations called anchors and highlights the part of the input that is sufficient for the classifier to make the prediction, making them intuitive and easy to understand.

In SHAP, any explanation of a model’s prediction is considered as a model itself, the explanation model. It belongs to the class of additive feature attribution methods, which includes LIME. The explanation model is simpler than the original and can be defined as any interpretable approximation of the original model. Additive feature attribution methods have an explanation model that is a linear function of binary variables. Methods with explanation models matching this definition attribute an effect to each feature, and summing the effects of all feature attributions approximates the output $f(x)$ of the original model, as shown in equation 3.2.

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i \quad (3.2)$$

where $z \in \{0, 1\}^M$, M is the number of simplified input features, $\phi_i \in R$, f is the original prediction model to be explained and g the explanation model. We focus on local methods designed to explain a prediction $f(x)$ based on a single input x : explanation models often use simplified inputs x' through a mapping function $x = h_x(x')$. Local methods try to ensure $g(z') \simeq f(h_x(z'))$ whenever $z' \simeq x'$.

SHAP uses classic equations from cooperative game theory to compute explanations of model predictions. In the class of additive feature attribution methods there is a single unique solution with three desirable properties (local accuracy, missingness, consistency): SHAP values provide the unique additive feature importance measure that adheres to these properties and uses conditional expectations to define simplified inputs. SHAP values are a unified measure of feature importance, values of a conditional expectation function of the original model as shown in formulation 3.3.

$$E[f(x)|X_s = x_s] \quad (3.3)$$

The exact computation of SHAP values is complicated because they are NP-hard in general. We can approximate them: feature independence and model linearity are two optional assumptions simplifying the computation of the expected values. The partial dependence plots of linear models have a close connection to SHAP values because each feature in the model is handled independently of every other feature (the effects are just added together). For linear models, with input feature independence, SHAP values can be approximated directly from the model's weight coefficients. Given a linear model $f(x) = \sum_{j=1}^M w_j x_j + b$:

$$\phi_0(f, x) = b$$

and

$$\phi_j(f, x) = w_j(x_j - E[x_j])$$

LIME. LIME belongs to the class of additive feature attribution methods, like SHAP: the black-box model's complex decision function f is unknown to LIME and cannot be approximated well by a linear model. LIME samples instances, gets predictions using f , and weights them by their proximity to the instance being explained. The learned explanation is locally (but not globally) faithful. The local linear explanation model that LIME uses, as in SHAP, is an additive feature attribution method, a linear function of binary variables. In LIME the effect of each feature can be found minimizing an objective function based on a squared loss, a problem that can be solved using penalized linear regression as follows:

$$\xi(x) = \operatorname{argmin}_{g \in G} L(f, g, \pi_x) + \Omega(g) \quad (3.4)$$

where g is an explanation, $\Omega(g)$ is a measure of complexity (as opposed to interpretability, for example in decision trees it may be the depth of the tree), $\pi_x(z)$ is a proximity measure between an instance z to x , $L(f, g, \pi_x)$ is a measure of how unfaithful g is in approximating f in the locality defined by $\pi_x(z)$.

LACE. LACE is a novel rule-based method that explains the prediction of any classifier on a specific instance by analyzing the joint effect of feature subsets on the classifier prediction. The relevant subsets are identified by learning a local rule-based model in the neighborhood of the prediction to explain. While local rules give a qualitative insight of the local behavior, their relevance is quantified by using the concept of prediction difference. Despite the approximation introduced by the local model, the explanations provided by LACE are effective in detecting the effects of attribute correlation.

Anchors. Anchors is a model-agnostic system that explains the behavior of complex models with high-precision rules called anchors, representing local, “sufficient” conditions for predictions. These explanations can be computed by an algorithm for any black-box model with high-probability guarantees, explaining different models for different domains and tasks. Anchors enable users to predict how a model would behave on unseen instances with less effort and higher precision, as compared to existing linear explanations or no explanations. Anchors is a novel family of model-agnostic explanations based on if-then rules, called anchors. An anchor explanation is a rule that sufficiently “anchors” the prediction locally, such that changes to the rest of the feature values of the instance do not matter. In other words, for instances on which the anchor holds, the prediction is (almost) always the same.

Relative Keys. While formal feature explanations maintain perfect conformity, they are often computationally intractable. Conversely, heuristic methods offer significantly higher speeds but may produce unreliable explanations due to a lack of conformity guarantees. To bridge this gap, we propose Relative Keys, a novel non-perturbative method that combines the strengths of both approaches. By associating feature explanations with a specific set of instances as context, Relative Keys ensure perfect conformity over that context—similar to formal methods, while remaining orders of magnitude faster and applicable to complex black-box models. Leveraging this framework, a prototype that computes explanations was developed (CCE), without requiring model access.

Feature importance in our algorithms. While our algorithms are distinct from the class of local explanation methods, we include a discussion of these techniques, specifically SHAP, as they provide a valuable framework for understanding feature importance within subspaces. A key focus of this analysis is the interplay between feature importance and range boundaries, which offers deeper insight into why specific solutions are prioritized during the optimization process.

Chapter 4

Subspace Classification Optimization and Explanation

4.1 Preliminaries and Problem Definition

We define a Subspace Optimization Problem (SOP) formulated as a *mixed-integer mathematical program*, as the problem of finding the subspace containing the largest number of points and respecting a minimum level of accuracy, decided by the user.

SOP takes as input the threshold accuracy β and returns lower and upper bounds for each feature (l^k and u^k for the k -th feature) representing a **range query** for identifying a high-accuracy subspace. Intuitively, the fact that there is a subspace with higher performances means two things: on the one hand, we can be more confident of the results if they lay in the subspace, and on the other hand, we may show the model is unbalanced for certain regions. In the rest of this section we describe the main components of the problem: sets, decision variables, parameters, constraints and objective function.

Sets. The main sets are:

- $I = \{1, \dots, P\}$ represents the set of indices of the points in the dataset.
- $(x_i, y_i)_{i \in I}$ represents the dataset, where $y_i \in R$ is the label, $x_i \in R^K$ is the vector of features.

The number of features in the dataset is given by the integer $K > 0$, and x_i^k is the k -th feature of x_i .

Decision variables. The decision variables are:

- $v_i, i \in I$, are binary indicator variables indicating the instances to be included in the final ranged problem. We use the notation v for the vector of all the variables v_i . The solution will have $v_i = 1$ if the i -th point belongs to the optimal subspace, and $v_i = 0$ otherwise.
- $p_i^k, i \in I$ and $k = 1, \dots, K$, are binary indicator variables indicating if the i -th instance belongs inside the k -th range. The solution will have $p_i^k = 1$ if the i -th point belongs to the k -th optimal range, and $p_i^k = 0$ otherwise.

- s_i^k , $i \in I$ and $k = 1, \dots, K$, are auxiliary binary indicator variables indicating to which disjunction the i -th instance belongs when it is not inside the k -th range.
- $u^k \in R$ for $k = 1, \dots, K$, are continuous variables representing the upper bounds of the ranges. We indicate with u the vector of all the variables u^k .
- $\ell^k \in R$, $\ell^k \leq u^k$ for $k = 1, \dots, K$, are continuous variables representing the lower bounds of the ranges. We indicate with ℓ the vector of all the variables ℓ^k .

Parameters and additional sets. In the following, we describe the main parameters and additional sets:

- $\beta \in (0, 1]$ is the threshold value for the accuracy. It is decided interactively by the user. Increasing β will lead to progressively higher accuracy of the optimal subspace found.
- r_k for $k = 1, \dots, K$ are the size of the ranges in the entire dataset.
- c_i , for $i \in I$, indicates if the label returned by the classification model is correct or not. In other words, $c_i = 1$ is the model ϕ correctly classifies the i -th vector x_i , i.e. $c_i = 1$ if $\psi(x_i) = y_i$, and $c_i = 0$ otherwise.
- $M^k = 1 + \max_{i \in I} \{|x_i|\}$, for $k = 1, \dots, K$ are the big-M values used to activate the disjunctive constraints for the lower and upper bounds following the methodology in⁽²⁵⁾.
- $K_\ell^a \subseteq \{1, \dots, K\}$ is the set for which there exists an upper bound a_ℓ^k on ℓ^k specified by the user
- $K_\ell^b \subseteq \{1, \dots, K\}$ is the set for which there exists a lower bound b_ℓ^k on u^k specified by the user.
- $\hat{K}_\ell^a \subseteq \{1, \dots, K\}$ is the set for which there exists a fixed value $\hat{a}_\ell^k$ for ℓ^k decided by the user.
- $\hat{K}_\ell^b \subseteq \{1, \dots, K\}$ is the set for which there exists a fixed value $\hat{b}_\ell^k$ for u^k decided by the user.
- $\epsilon > 0$ is a small value.

4.2 Subspace Classification OPTimization and Explanation

In the following we will present our method SCOPE (Subspace Classification OPTimization and Explanation), for computing high-accuracy ranges in order to interpret the model's classification patterns. Due to the difficulties in solving the mixed-integer problem with large datasets, this method can be optimized through a grid search within cells obtained via a binning operation. This simple mechanism effectively

$$\begin{aligned}
\text{SOP}(\beta) &= \max_{v, \ell, u} \sum_{i \in I} v_i - \frac{1}{K+\epsilon} \sum_{k=1}^K \frac{u_k - l_k}{r_k} & (1) \\
\text{s.t. } & \sum_{i \in I} c_i v_i - \beta \sum_{i \in I} v_i \geq 0 & (2) \\
& -M^k(1 - p_i^k) + \ell^k - x_i^k p_i^k \leq 0, \quad \forall i \in I, k = 1, \dots, K & (3) \\
& x_i^k p_i^k - u^k - M^k(1 - p_i^k) \leq 0, \quad \forall i \in I, k = 1, \dots, K & (4) \\
& M^k(p_i^k + 1 - s_i^k) - \epsilon(1 - p_i^k) + \ell^k \geq x_i^k, \quad \forall i \in I, k = 1, \dots, K & (5) \\
& -M^k(p_i^k + s_i^k) + \epsilon(1 - p_i^k) + u^k \leq x_i^k, \quad \forall i \in I, k = 1, \dots, K & (6) \\
& K v_i - \sum_{k=1}^K p_i^k \leq 0, \quad \forall i \in I & (7) \\
& v_i - \sum_{k=1}^K p_i^k \geq 1 - K, \quad \forall i \in I & (8) \\
& \ell^k \geq a_\ell^k, \quad k \in K_\ell^a & (9) \\
& u^k \leq b_u^k, \quad k \in K_\ell^b & (10) \\
& \ell^k = \hat{a}_\ell^k, \quad k \in \hat{K}_\ell^a & (11) \\
& u^k = \hat{b}_u^k, \quad k \in \hat{K}_\ell^b & (12)
\end{aligned}$$

Figure 4.1 Definition of the Subspace Optimization Problem for finding the largest subset of instances with accuracy $\geq \beta$

reduces the complexity, allowing us to find solutions even for high thresholds, where the ordinary problem typically results in a timeout. In 4.2.1 the mixed integer optimization problem is defined in its ordinary form, whereas the optimized version of the algorithm is described in 5.1.

4.2.1 The Mixed Integer Optimization Problem

In order to compute the sub-space partitioning, we design a mixed integer optimization problem and use a state-of-the-art optimizer⁽⁶⁾ to find the hyper-rectangular sub-space that has maximum size and accuracy above threshold. The size of a sub-space is simply defined as the number of enclosed data points.

The computed result is returned to the user as a *range query*, which provides a compact and human readable way to identify high-accuracy data points and low-accuracy data points as respectively those inside and outside the ranges.

Constraints and objective function. Figure 4.1 formally defines $\text{SOP}(\beta)$ as the problem of finding the ranges of the largest subset of instances having an accuracy $\geq \beta$.

The high-level description of the main constraints follows.

- (2) indicates that the accuracy of the subspace should not be lower than β .
- (3)–(4)–(5)–(6) are disjunctive constraints indicating that the k -th component of the i -th point, x_i^k , belongs to the interval $l^k \leq x_i^k \leq u^k$ or it is outside, i.e. $x_i^k < l^k$ or $x_i^k > u^k$.
- (7) are constraints indicating that if the i -th point is selected then it must belong to all the K ranges, i.e. $v_i = 1 \iff p_i^k = 1$ for $k = 1, \dots, K$.
- (8) are constraints indicating that if a point belongs to all the K ranges then it has to be selected, i.e. if $p_i^k = 1$ for $k = 1, \dots, K$ then $v_i = 1$.

- (9) – (10) are bound constraints limiting the choice of the ranges below the levels specified externally.
- (11) – (12) are fixing constraints, blocking the ranges to the levels specified externally.

The objective function (1) is obtained according to the results in⁽¹³⁾, as a transformation by scalarization of the lexicographic optimization problem:

$$\text{lex max}_{v,\ell,u} \left(\sum_{i \in I} v_i, -\frac{1}{K+\epsilon} \sum_{k=1}^K \frac{u_k - l_k}{r_k} \right)$$

by noticing that the first objective is a γ -function. The objective function maximizes the number of selected points, while introducing a regularization term that penalizes excessively large ranges. The second term of the expression is introduced to compensate for varying range scales, thereby ensuring comparability.

Implementation details. We solve SOP with the Gurobi Optimizer Python library with academic free license⁽⁶⁾. The optimizer eventually returns the optimal solution. We also include a parametric time budget, which by default is set to 10 minutes, after which the current best solution is returned. We call our tool SCOPE, standing for **S**ubspace **C**lassification **O**ptimization and **E**xplanation. We slightly abuse the term “explanation” because computing high-accuracy ranges can be broadly thought as an attempt to interpret the model’s classification patterns. Users of the demo will be able to use our tool either standalone or together with the SHAP⁽²²⁾ explanation tool to show the feature importance in the high-accuracy area.

4.2.2 Data Pre-Processing for the Subspace Optimization Problem

To find the best solutions for the MIP problem called SOP (Subspace Optimization Problem), satisfying the property of monotonicity as accuracy increases, a pre-processing phase is needed: each feature has to be normalized with min-max scaling, where the lower and upper original values become 0 for the minimum and 1 for the maximum. This ensures that ranges have the same breadth, making optimization more efficient. Consequently, the optimal lower and upper bounds of the analyzed features are within the range of 0 to 1, but a de-normalization step can recover original values. Min-max scaling ensures that upper and lower bounds are always non negative, even when original values are negative, which is a desirable property for Scope. The range of values of raw data can vary widely and objective functions do not work properly without normalization, necessitating scaling to neutralize varying ranges and ensure comparability.

Many classifiers calculate the distance between two points using the Euclidean distance, and if one of the features has a broad range of values, the distance will be dominated by that particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance. Another reason for applying feature scaling is that gradient descent converges much faster with it than without. It is also important to apply feature scaling if regularization is used as part of the loss function (so that coefficients are penalized appropriately).

The limitation of min-max scaling, in contrast with standardization that centers data around the mean (0) and scales it according to the standard deviation (1), is

its sensitivity to outliers. Mean normalization can successfully adjust for outliers in some scenarios, but other techniques might not be as effective. In general, normalization techniques do not handle the outlier problem as effectively as standardization because standardization explicitly relies on both the mean and the standard deviation. Normalization is widely used in distance-based algorithms like K-Nearest Neighbors (KNN), where features must be on the same scale to ensure accuracy in distance calculations. Standardization, on the other hand, is vital for gradient-based algorithms such as Support Vector Machines (SVM) and is frequently applied in dimensionality reduction techniques like PCA, where maintaining the correct feature variance is important.

When normalization is applied to the data, the Subspace Optimization Problem is simplified because only continuous and binary attributes are considered, all in the range between 0 and 1: integer values are 0 and 1, while all others are positive decimal numbers less than 1.

4.3 The SCOPE Algorithm

This section presents our algorithm for Subspace Classification OPTimization and EXplanation (SCOPE) that has been described in the article "Breaking Down Accuracy with Subspace Optimization"⁽⁵⁾, published in the International Conference on Extending Database Technology Proceedings in 2024. The following subsections report the introduction, the demonstration scenarios and the conclusions of the published paper presented exactly as submitted, whereas the formulation of the Subspace Optimization Problem is defined in Sections 4.1 and 4.2.

4.3.1 Introduction

Classification models have become ubiquitous in a wide range of applications, from healthcare to finance. These models, powered by sophisticated algorithms, can play a pivotal role in decision-making processes. The standard de-facto metric for assessing model performances is *global accuracy*, that is, the fraction of correctly classified data points on a test set. However accuracy can vary wildly depending on the characteristics of a certain dataset, and there are two main reasons for which global accuracy alone can be misleading:

- *High global accuracy $\neq$ good model.* High-accuracy models can exhibit bias, performing well for over-represented groups or types of data, but not for others that are equally important.
- *Modest global accuracy $\neq$ bad model.* Different applications can tolerate errors in sub-spaces of the data that are not critical.

Intuitively, breaking down global accuracy would build confidence in using the model for real-world decisions. Knowing where the model excels and where it does not would allow users to make informed decisions about its applicability and limits, guiding data collection, feature engineering, and model tuning.

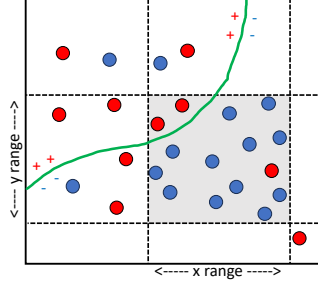


Figure 4.2 High-accuracy subspace in two dimensions. The space is separated in two regions according to the green line. Data points to the left are classified as red (+) and data points to the right are classified as blue (-). The ground-truth class of each instance is represented by the color of each point.

Unfortunately, to the best of our knowledge, there are no standard approaches to perform accuracy break down. In this paper, we initiate this study by demonstrating our current solution, which is based on a simple - yet effective - break-down policy.

Our break-down policy is illustrated in Figure 4.2. In the figure, each point represents a different instance of a two-dimensional test set and the global accuracy is $\frac{25-6}{25} = 0.76$. Given a threshold such as 0.9 to quantify high accuracy, our tool can compute a partition of the evaluation’s scope in two parts: a high-accuracy sub-space, in grey, and a low-accuracy sub-space, in white. Accuracy of the two areas is respectively $accuracy(grey) = \frac{14}{15} \approx 0.93$, above the threshold, and $accuracy(white) = \frac{5}{10} = 0.5$.

In order to compute the sub-space partitioning, we design a mixed integer optimization problem and use a state-of-the-art optimizer⁽⁶⁾ to find the hyper-rectangular sub-space that has maximum size and accuracy above threshold. The size of a sub-space is simply defined as the number of enclosed data points.

The computed result is returned to the user as a *range query*, which provides a compact and human readable way to identify high-accuracy data points and low-accuracy data points as respectively those inside and outside the ranges.

Contribution. We call our tool **SCOPE**, standing for **S**ubspace **C**lassification **O**ptimization and **E**xplanation. We slightly abuse the term “explanation” because computing high-accuracy ranges can be broadly thought as an attempt to interpret a model’s classification patterns. Users of the demo will be able to use our tool either standalone or together with the SHAP⁽²²⁾ explanation tool to show the feature importance in the high-accuracy area.

SCOPE is available on GitHub¹. The high-level architecture of the system consists of four modules:

1. *Dataset module.* A module to load/select datasets and features. We provide a collection of pre-loaded datasets. Users can bring their own datasets by uploading files in a dedicated folder.
2. *Classification module.* A module to select/train a classification model, make predictions and compute accuracy. We include by default three classification

¹<https://github.com/Flaviatagliafiero/Scope>

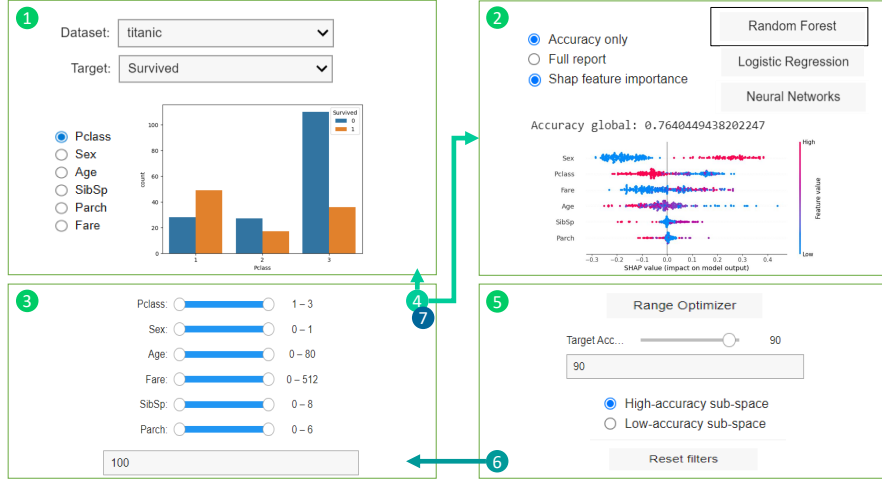


Figure 4.3 Main demonstration functionalities.

models, dubbed Random Forests, Logistic Regression, and feedforward Neural Network. Users can bring their own models by implementing a Python class.

3. *Optimization module.* A module that takes as input the dataset, the features and the classification model as described in the previous modules. Then, it solves a mixed integer optimization problem to finds the largest sub-space with a certain minimum level of accuracy desired. The module adopts GUROBI⁽⁶⁾ as an optimizer and returns a range query that partitions the test data space into an high-accuracy and a low-accuracy sub-space.
4. *Display module.* A graphical interface to interact with the automatically computed ranges, adjust them manually and observe changes in the accuracy and feature importance.

The existing literature is rich with applications where high-accuracy classification models deviate from the desired behaviour in unexpected ways. One popular example is a classifier for images of huskies and wolves⁽²⁴⁾, unable to recognize pictures of huskies without snow in the background. Another example is a 95% accuracy Entity Matching model unable to recognize two different songs with the same duration⁽¹⁸⁾. Range queries are a popular tool in data analysis. We mention the recent work⁽¹⁰⁾ where ranges are automatically computed to identify *fair* sub-spaces, according to different fairness constraints⁽²⁰⁾.

Overview. In Section 4.2 we formalize the sub-space optimization problem at the heart of our approach. In Subsection 4.3.2 we illustrate different demo scenario using publicly available datasets. Finally Subsection 4.3.3 provides conclusive remarks.

4.3.2 Demonstration Scenarios

In Figure 4.3, we illustrate the user interface with a simple dataset from Kaggle, dubbed `Titanic`². The dataset contains 889 records representing Titanic passengers and their survival status. Features include the traveling class (“PClass”), the cost of their ticket (“Fare”), demographics data such as “Sex” and “Age”, the size of the group, divided into siblings/spouses (“SibSp”) and parents/children (“ParCh”). For people traveling alone and we have that “SibSp”+“ParCh”=0.

Initial set-up. In step ① the user can choose a dataset and specify a target variable for the classification task, e.g., “Survived”. Upon a new selection, features are shown below and the user is able to select them and inspect the corresponding distribution of the datapoints in the test set, split by the “Survived” class.

Users can bring their own datasets by uploading two files in a dedicated folder: a CSV file with all the instances and a metadata file with the type of each feature among **natural-ordering**, **ordinal-feature** and **no-ordering**. Examples of features of the first type are numerical features, although there can be textual features that make sense to sort lexicographically. Features of the second type include categorical features that support a domain-based ordering relation, such as “Degree”. Features of the third type include features that have no natural or domain ordering such as “Country”. The latter type, i.e., **no-ordering**, cannot be added to the optimization problem but can still be used here, by the classifier and the explanation tool if the user wants to.

Users can optionally upload a third file to specify how to split the dataset into training and test. If such a file is not provided, **SCOPE** does a random 70% – 30% split.

Classification results. In step ② the user can choose a classification model. We provide three options for the demonstration: Random Forest, Logistic Regression, and Feedforward Neural Network. Upon a new selection, that is, “Random Forest” in Figure 4.3, the tool trains a new classification model and then generates a classification results report. The user may customize the report choosing to show a combination of accuracy (“Accuracy only”), confusion matrix (“Full report”), and distribution of the feature importance values. By default, the latter is computed with the popular SHAP⁽²²⁾ explanation toolkit (“Shap feature importance”). Users cannot bring new models during the live demo. However, they can bring their own models when using **SCOPE** off-line, by implementing a class in a dedicated Python module. Pre-trained models are also supported.

Manipulating ranges. In ③, the user can interact with range sliders to decide to set up bounds that will be included in the optimization module. Below the ranges, a textbox displays the size of the selected sub-space as a percentage. In Figure 4.3, we show the initial configuration of the sliders, with the textbox showing the value 100, indicating the entire test set will be included.

Ranges can be manually adjusted (step ④) by simply moving the sliders. Upon adjustments, the value of the selected percentage of the test set changes, and the results in ① and ② are updated by restricting the scope to the selected range. For instance, by restricting the “Age” range to 18-35, plot in ① would show the “PClass” distribution by considering only data points of the test set that have an age between

²<https://www.kaggle.com/competitions/titanic>

18 and 35 years. Analogously, the accuracy in ② would become the accuracy of the selected sub-space.

The button “Reset Filters” in part ⑤ resets all the ranges.

Sub-space optimization. In step ⑤, the user can set the accuracy threshold and run the optimization module to solve the SOP problem defined in Section 4.2, initialized with the bounds in ③. As soon as optimal ranges are ready, the slider buttons are updated (step ⑥) and therefore the visualizations in ① and ② (step ⑦, analogously to step ④). With the setup in Figure 4.3, that is no initial range restriction and target accuracy equals to 90%, the solution found has size 23% of the test set and corresponds to “Pclass”=[2, 3], “Age”=[17, 30] and “Fare”=[7, 13]. The accuracy achieved on this subspace is 0.90625, just above the threshold. Such sub-space can be intuitively described as young passengers with cheap tickets, travelling in second and third class.

During the live demo, users will be able to inspect the full report in ② (omitted from the figure for space constraints) and discover that most of those passengers did survive³ and that even a simple classifier was able to learn this pattern with exceptional accuracy.

Other scenarios. During the live demo, users can change dataset, classification model, lower and upper ranges, try different accuracy thresholds are run the optimizer with different settings. They will be able to highlight strengths and weaknesses of selected classifiers and learn more about the selected dataset. Finally, all the visualizations concern by default the high-accuracy sub-space found by the optimizer. However, the user can select the “Low-accuracy sub-space” option in part ⑤ and inspect the part outside the ranges.

4.3.3 Conclusions

This paper presented SCOPE, a Python toolkit to enhance the evaluation of classification models by providing a more granular analysis of model performance. Our approach diverges from the traditional reliance on global accuracy and represents a significant improvement over manual inspection of accuracy variation, thanks to a mixed integer optimization algorithm for automatic identification of high-accuracy sub-spaces within the test set.

During the live demo, users will be able to explore these sub-spaces by running the optimization algorithm and inspecting results. Furthermore, they will be able to bring their own datasets and uncover common characteristics among data points that are correctly or incorrectly classified, thereby narrowing the scope for in-depth examination of classification patterns.

Intuitive graphical elements such as slider buttons make the toolkit accessible to a diverse range of users, regardless of their technical expertise, enabling them to gain a deeper understanding of how changes in data can impact model accuracy and to learn to identify possible biases that affect performance variations.

Future works. Future works will focus on expanding the toolkit with other performance metrics like precision and recall and adding constraints to solution sizes, further increasing its applicability. Other envisioned expansions include compact

³This might be surprising for those remembering the *Titanic* movie by James Cameron.

representations alternatives to hyper-rectangles, such as cluster of data points which are correctly / incorrectly labeled. Finally we plan to explore efficient methods to compute approximate solutions over large datasets.

Chapter 5

The SCOPE Grid Algorithm

This chapter presents our scalable Scope Grid algorithm and an experimental comparative analysis with the Scope algorithm described in Section 4.2 and the DivExplorer algorithm described in Section 3.1. These results are included in the paper "Efficient Explanation of Classification Models with Accuracy Ranges Optimization", that is under review at the Journal of Data and Information Quality in 2026.

In this chapter we evaluate our method Scope comparing it with other similar methods in the same class to find analogies and verify performance in different datasets. To delve deeper into our method, we create a grid search space to decompose the algorithmic complexity. Afterward, we test our algorithm to demonstrate its efficiency in finding subspaces as the accuracy threshold increases. We calculate both the execution times and the percentage of data points in the subset. Subsequently, we describe the High-Performance Computing (HPC) cluster supporting the computational challenges in the optimization process. Finally, we present a case study illustrating the different approaches to feature contribution and subgroup definition in Scope and DivExplorer.

5.1 Improving Efficiency with Grid Search

The Scope algorithm performs well only under limited conditions, specifically when the number of instances and features is small. Complexity of the algorithm is an important limit in the Subspace Optimization Problem (SOP) and efficient methods for computing approximate solutions over large datasets can be very helpful. The SOP is formulated as a mixed integer mathematical program, as the problem of finding the subspace containing the largest number of points and respecting a minimum level of accuracy, and the execution time depends heavily on the number of data points. To overcome these limitations, we introduce a grid approach in order to simplify the analysis and reduce complexity. This optimized version of the algorithm proves to be more powerful than the classical one, thanks to a simple breakdown of the original dataset into cells. In order to achieve this result we construct multidimensional grids to partition the space based on an equi-depth binning operation starting from the median. This is a classic binning approach, with a quantile based discretization function to account for variable skewness. the number

of elements in each bin is roughly the same, but interval widths are differently sized. On the other hand, we can have the bin edges equal sized with a different number of elements in each bin or group. In an equi-depth binning we specify the number of quantiles, telling to cut the variable column into equal quantiles. We first consider datasets with a sufficiently large number of instances (such as the Compas dataset) to ensure adequate sample size in every quadrant. This approach is justified by the fact that the method exhibits certain limitations precisely when applied to larger datasets.

It's interesting to note that Scope encounters significant issues with large datasets, which contains thousands of instances. Conversely, in a grid search space, we can find optimal solutions even with a large number of points by simply employing a binning method. The use of a grid search space ensures that the entire dataset can be analyzed efficiently while remaining within established time constraints.

Figure 5.1 illustrates how Scope optimization operates when applied to a grid search. The partitioning into cells, resulting from the equi-depth binning of features, is shown alongside a color gradient representing the accuracy level of each cell. Consequently, Scope identifies subspaces with broader coverage where accuracy is high, and a larger number of instances where cell density is greater. In the figure, the symbols 'X' and 'O' represent correct and incorrect predictions in a binary classification task for some illustrative cells. This allows for a more precise description of the differences between cells, which do not share the same level of accuracy. For instance, in cell 7, the accuracy is 0.5 (5 correct predictions out of 10 instances), while in cell 9, it is 0.7 (7 correct predictions). The latter cell is characterized by a higher density of correct predictions; consequently, finding a solution with Scope is more straightforward and results in a higher number of points.

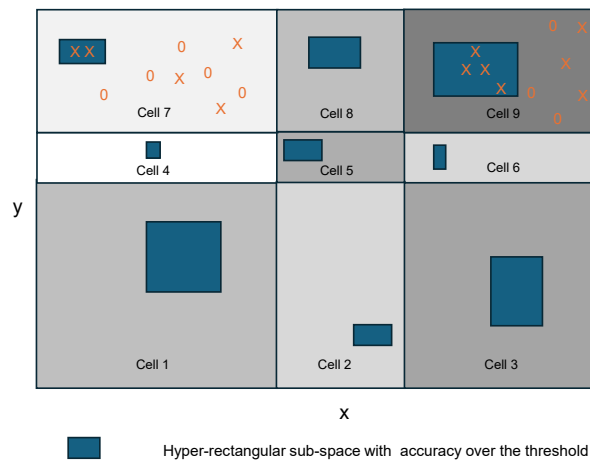


Figure 5.1 Scope optimization when applied to a grid search: accuracy level and hyper-rectangular subspaces.

As a consequence, the standard version of Scope is enriched with its grid-optimized

variants, defined as Grid Sum and Grid Max. These variants can reduce the standard algorithm’s complexity by operating within the discrete cells generated through partitioning.

In this configuration, Scope Grid Sum (Scope GS) calculates solutions of the optimization problem in each partition; the union of the resulting ranges creates a multi-subspace consisting of disjoint ranges for each feature. Since the subspaces do not overlap, we can sum the points obtained in each cell while maintaining a minimum level of accuracy and the total coverage is the sum of all contributions from each partition. Another version of the grid algorithm, called Scope Grid Max (Scope GM), is based on the maximum number of points obtained by Scope across the cells. It might seem that the standard solution would outperform Scope GM; however, this is not always true, especially when dealing with high accuracy levels in large-scale datasets.

Regarding execution times, Scope GS can be conceptualized as a serial computation where processing times are recorded cumulatively. In contrast, Scope GM can be described as a parallel computation, in which cells are processed simultaneously, resulting in a total execution time equivalent to the maximum duration recorded among the cells.

5.2 Experimental Setup

Datasets. We use datasets from three different domains and with different sizes, where the target is binary and the number of features is small for the sake of simplicity in demonstrating our results. Features can be numerical and categorical, but for nominal ones binary transformation is needed.

We illustrate our method with a simple dataset from Kaggle, dubbed *Titanic* ¹. The dataset contains 889 records representing Titanic passengers and their survival status. Features include the traveling class (“PClass”), the cost of their ticket (“Fare”), demographics data such as “Sex” and “Age”, the size of the group, divided into siblings/spouses (“SibSp”) and parents/children (“ParCh”). For people traveling alone and we have that “SibSp”+“ParCh”=0.

We also utilize the well-known Compas dataset ² (Correctional Offender Management Profiling for Alternative Sanctions) containing information of defendants. This dataset, published by a non-profit investigative journalism organization, includes individual recidivism risk score, the actual recidivism over a two-year period, demographic variables and criminal history, such as age, race, sex, offence type and the number of prior offences.

The Credit dataset ³ classifies people described by a set of attributes as good or bad credit risks. The selected attributes for this dataset are: sex, age, job, credit amount and duration.

The datasets are diverse in nature, spanning real-world or synthetic data, with variations in scale and differing levels of accuracy for classification models. We

¹<https://www.kaggle.com/competitions/titanic>

²<https://www.kaggle.com/datasets/mlethics/compas>

³<https://www.kaggle.com/datasets/uciml/german-credit>

compute first of all the accuracy, the fraction of correctly classified data points on a test set, but also the F1 score, precision, and recall.

Metrics. Label balance (the proportion of the classes) can explain the amount of accuracy. If the balance of the subgroup differs significantly from the complete dataset, the implication is that the classifier performs well on the majority class but poorly on the minority class. The F1 score can confirm this hypothesis, if there are differences between F1 values of the classes. To evaluate subspace optimization, metrics and the imbalance of true labels can be helpful. Accuracy is useful if classes are balanced, but it can mask the importance of minority class, because it focuses on overall correctness rather than the ability to predict a specific class.

Recall and precision help in identifying high performance on the target class: recall considers false negatives, while precision controls false positives. Optimizing for recall means a lower decision threshold and optimizing for precision implies a higher decision threshold (a trade off exists between recall and precision). In our examples the classes are quite balanced, so accuracy works well: we aim to find overall correctness but recall, precision and the F1 score can also be evaluated as indicators of the optimal subspace quality and representativeness.

To find the best solutions for the Mixed Integer Programming (MIP) problem called SOP (Subspace Optimization Problem), satisfying the property of monotonicity as accuracy increases, a pre-processing phase is needed: each feature has to be normalized with min-max scaling, where the lower and upper original values become 0 for the minimum and 1 for the maximum. This ensures that ranges have the same breadth, making optimization more efficient. Consequently, the optimal lower and upper bounds of the analyzed features are within the range of 0 to 1, but a de-normalization step can recover original values. Min-max scaling ensures that upper and lower bounds are always non negative, even when original values are negative, which is a desirable property for Scope.

Classification models. Classification models and their sophisticated algorithms can play an important role in decision making processes. We train three models on the training set in order to predict target feature in the test set, Random Forest, Logistic Regression, and Feedforward Neural Networks, and generate a confusion matrix, a classification results report and a distribution of the feature importance values. By default, we compute them with the popular SHAP⁽²²⁾ explanation toolkit (“Shap feature importance”). To demonstrate that Scope is strictly model-agnostic, we evaluate it using various classification algorithms. Since Scope’s logic relies primarily on predicted labels rather than internal model architecture, the specific classifier used does not influence the detection process. Furthermore, to ensure compatibility, specific Shapley value explainers are provided in Python for each implemented method.

5.3 Comparison of SCOPE and DivExplorer

To compare Scope with the most similar Automatic Subgroup Detection method (ASD), we consider DivExplorer, which analyzes classification behavior in subgroups using different outcome functions. Details about the Scope toolkit and DivExplorer are available in the previous sections, where we discussed their similarities and

differences. Slice Finder is similar to DivExplorer in that it identifies slices of data points where performance is critical by comparing the classifier loss on the slice to the loss on the rest of the data. This method automatically finds large data slices where the model performs poorly, defined as “problematic” slices. Although the Slice Finder method shares a similar scope, DivExplorer proves more effective at identifying problematic itemsets. Consequently, our comparative analysis focuses exclusively on the latter.

In the following, we will describe the application of the two methods to different datasets. For all datasets numerical attributes (such as age) must be discretized before performing the DivExplorer method. For nominal features, discretization is not required because they have already been transformed into binary values according to the requirements imposed by Scope.

Experimental choices. Regarding the discretization strategy, features are binned according to their distributions. For all datasets, Age is categorized into ‘up to 25’, ‘25–45’, and ‘over 45’. Specifically for the Titanic dataset, Fare is discretized into intervals of ‘up to 10’, ‘10–20’, ‘20–30’, and ‘over 30’. In the Compas dataset, Decile Score is grouped into ‘1–4’, ‘5–7’, and ‘8–10’, while Priors Count is divided into ‘0’, ‘1–3’, and ‘over 3’. Finally, for the German Credit dataset, Duration is binned into ‘up to 25’, ‘25–50’, ‘50–75’, and ‘over 75’, and Amount into ‘up to 5’, ‘5–10’, ‘10–15’, and ‘over 15’. In all our analyses, the support is fixed at 0.10, as this value represents the required minimum threshold for the Scope optimization problem. An exception is made exclusively for the Compas dataset, where the minimum support is set to 0.01, aligning with the subspace coverage obtained in Scope. For DivExplorer, we select, among all extracted frequent itemsets, those whose accuracy exceeds the threshold used in Scope and report the subgroup with the largest size. To obtain a comprehensive set of accuracy values for comparison, we process all itemsets without the redundancy pruning approach proposed in⁽¹¹⁾ to have a more compact representation.

Table 5.1 presents the most important information regarding the two methods to highlight their similarities and differences. This information is useful for understanding the analysis in this section, where real datasets are trained and tested. The experiments were performed in a Google Colab environment (Google Compute Engine Python 3 backend), with a system configuration providing 12.67 GB of RAM. The optimization process was conducted using the Gurobi Optimizer Python library under a free academic license. Reproducing the results requires Python and the Gurobi optimization solver.

The function "DivergenceExplorer" is available in Python for obtaining the divergence in different subsets: we can choose accuracy as an outcome, noting that in Scope accuracy serves as the constraint in our optimization problem. In this way itemsets should not be considered "critical" but "optimal", as DivExplorer tries to maximize the divergence of outcomes⁽¹¹⁾.

Experimental results. In table .1 and table .2 (in Appendix), we indicate the chosen accuracy level, the execution times, the number of points in the subgroup, and their percentage relative to the total for three different binary classification models.

We generally observe that, for all models considered, as the threshold increases, the Scope method produces subspaces with a greater number of points compared

Topic	Scope	DivExplorer
method	gurobi	frequent pattern mining
maximum	number of data points	divergence in outcome functions
user interaction	threshold for accuracy	threshold for support, metric
solutions	unique	multiple
metrics	accuracy	accuracy and other metrics
model agnostic	YES	YES
feature scale	interval scale	nominal scale
source code	Python	Python
Shapley theory	important factors	contribution to divergence

Table 5.1 Qualitative comparison of two Automatic Subgroup Detection techniques: Scope and DivExplorer.

to DivExplorer. This trend is also demonstrated in figure 5.2. One contributing factor to this difference in the results is that DivExplorer requires the discretization of numerical attributes to generate itemsets, as features must be represented on a nominal scale. This preprocessing step reduces the granularity of the data and affects the extraction process of subgroups and thus the final results. In contrast, Scope operates directly on continuous features without requiring the transformation of numerical attributes. As a result, Scope preserves the original information and searches directly in the continuous feature space.

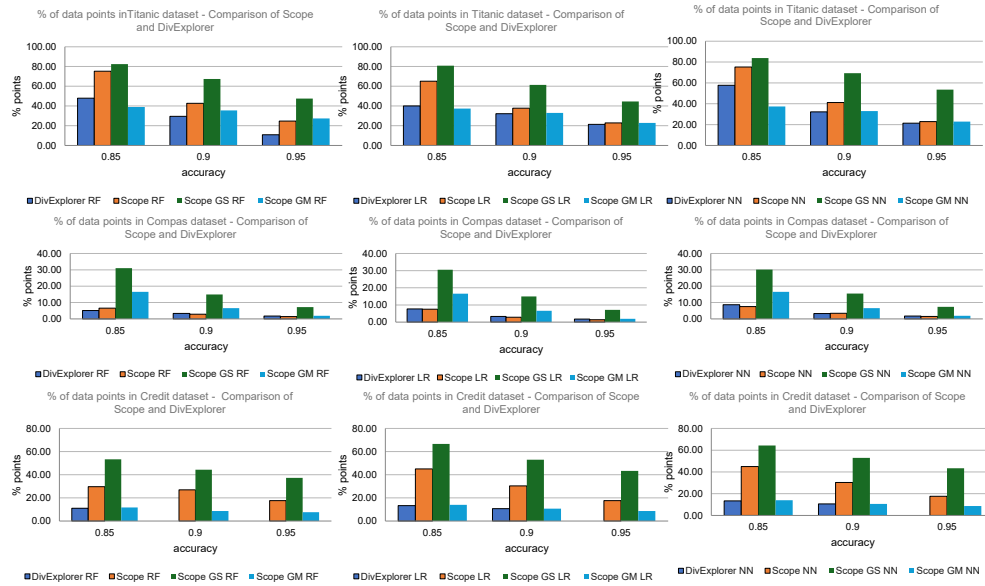


Figure 5.2 The percentage of points in two Automatic Subgroup Detection techniques for the Titanic, Compas and Credit dataset: a comparison of Scope (standard, Grid Sum, Grid Max) and DivExplorer.

Another important difference concerns how accuracy is handled. Scope produces a solution for any given input accuracy threshold, as accuracy is enforced as a constraint in the optimization problem. In DivExplorer, instead, accuracy is an

outcome of the method: subgroups are extracted under a support constraint, and then evaluated in terms of accuracy and divergence with respect to the overall dataset. Therefore, not all accuracy levels can be obtained. This is reflected in the missing values reported in table .1 (in Appendix), where no subgroup satisfying the desired accuracy threshold is found.

It is evident, however, that the execution times for Scope are longer than those for DivExplorer, due to the complexity of the mixed integer optimization problem. Conversely, DivExplorer is based on the extraction of frequent itemsets, which are then ordered by divergence or support. Frequent Pattern Mining (FPM) techniques have a limited impact on the computational complexity of the problem.

Figure 5.2 compares the standard version of Scope with its grid-optimized counterpart (Grid Sum, Grid Max). The latter reduces complexity by partitioning the problem into discrete cells. In this configuration, Scope Grid Sum (Scope GS) solutions consist of disjoint ranges for each feature, allowing for the identification of more complex subspaces. Figure 5.2 reports for Scope GS the total coverage for these disjoint subgroups. The results show that this grid-optimized variant demonstrates a clear advantage in identifying optimal subspaces within larger and higher-dimensional datasets like Compas dataset. Indeed, the previous figures present results for the complete dataset, whereas the standard Scope method requires an estimation when dealing with the full scale version and an increase of trees in the Random forest classifier, as described in Section 5.5.

In figure 5.2, we present another version of the grid algorithm based on the maximum number of points among the cells, Scope Grid Max (Scope GM). It may seem that the standard solution is better than the Scope GM; however, this is not always true, especially regarding high accuracy levels in large-scale datasets.

Generally, Scope achieves higher quality results than Scope GM in terms of the percentage of points in the identified slice. This is expected since Scope is not limited to a single cell; by considering the entire search space, it can identify larger ranges with higher sample counts. The only exception is the Compas dataset. Due to its high dimensionality and volume, the optimization fails to find an ideal solution within the full search space. In this instance, Scope GM allows for larger ranges with more points, demonstrating the clear benefit of grid optimization in complex scenarios. It is worth noting that DivExplorer shows no limitations in processing large datasets.

5.4 Technical Details about the Multidimensional Grid Search Space

The clustering coefficient is a measure in graph theory and network science that quantifies the degree to which nodes in a graph tend to cluster together. Calculating the clustering coefficient for a dataset after transforming it into a similarity graph provides deep insights into the local and global topology of the data. The Global Clustering Coefficient reveals whether the data inherently tends to cluster and form well defined groups, even if it is not an inherent property of a table. In this context, this coefficient is useful to determine whether a partitioning approach is effective.

The performance of the grid approach appears to be good even when the clustering

coefficient is lower. The Titanic and Compas datasets, where the points are more connected to each other, exhibit coefficients of 0.62 and 0.63, respectively, compared to 0.40 for the Credit dataset. But we observe that even for the Credit dataset, the optimized version of the algorithm improves execution times and finds a greater number of optimal points, as for the other datasets.

In the grid search process for the Titanic dataset we obtain 4 cells, for the Compas dataset, in the full scale version, this process yields 8 cells, while for the Credit dataset, we obtain 11. The number of cells depends on the variable types, as continuous features result in a larger number of cells during the binning procedure.

Figure 5.3 illustrates the minimum and maximum levels of accuracy across the datasets cells and the minimum and maximum size of the cells, suggesting that this grid-based approach facilitates the clustering of units into homogeneous groups. We see for example that for the Compas dataset the accuracy ranges from 0.40 to 0.82 and the sample size ranges from 3.33% to 19.49% of the test set. Appropriate cell sizing ensures that our mixed integer optimization method results in a significant number of instances within the target subspace.

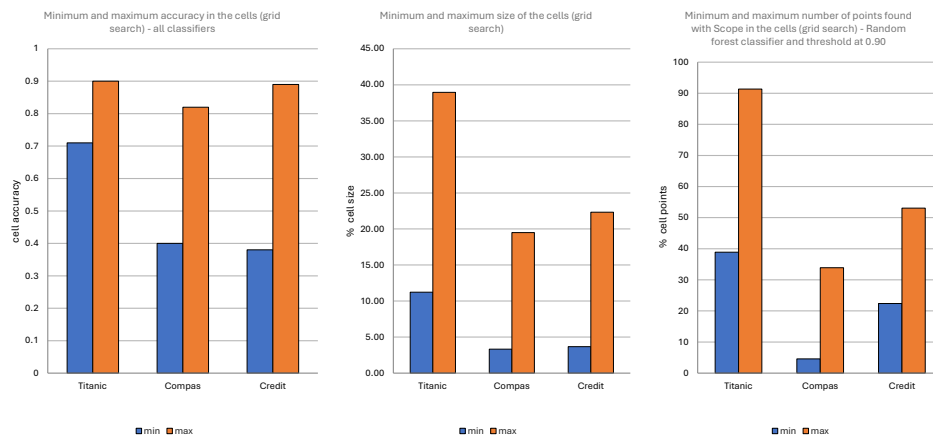


Figure 5.3 Accuracy and size of the cells (grid search) for the Titanic, Compas and Credit dataset.

For each cell, we calculate the number of points in the optimal subspace obtained with Scope, evaluating the results according to various accuracy thresholds and classifiers. The number of points in the optimal subspace depends on the size of the cell and on the accuracy. In figure 5.3 we find varying percentages of data points for the same threshold and classifier. By way of example for the Titanic dataset classified with a Random forest model at a threshold of 0.90, the percentages range from 38.89% to 91.35%, whereas the corresponding percentage for the entire dataset is 42.70%.

In the grid search, we can compute with Scope GS an aggregated subspace, which is formed by the union of the points in each cell's subset. All the subspaces

found in every cell for a certain threshold generates a new subspace (the union of hyper-rectangular subspaces) that satisfies the same condition for the accuracy. The subspaces are not overlapping, so we can sum the points obtained in each cell, satisfying a minimum level of accuracy. This aggregated number is not the same as the one calculated for the entire dataset without the grid search. The aggregated subspace depends on the grid search approach, which offers flexibility in the research.

For the datasets considered, this aggregated solution always contains a greater number of elements than the subspace obtained with the standard method. The performance in terms of execution time is better for the aggregated solution, as is the case of the number of points. This is clearly shown in figure 5.2 and figure 5.4, where the grid-based solution results in the best performance in terms of coverage and execution speed.

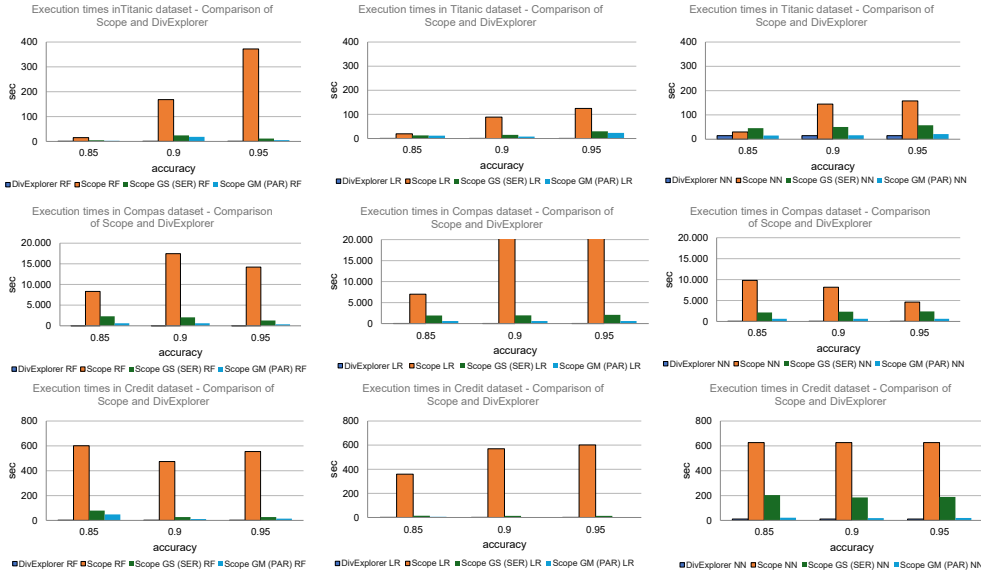


Figure 5.4 The execution times in two Automatic Subgroup Detection techniques for the Titanic, Compas and Credit dataset: a comparison of Scope (standard, GS SERial, GM PARallel) and DivExplorer.

When we consider the maximum number of points obtained among the cells (Scope GM) instead of the aggregated subspace, we observe that the former is usually worse than the standard solution, with the exception of the Compas dataset. In this case, due to the larger scale of the data, the standard solution is not always better than the maximum among the cells. Scope GM can outperform the standard formulation at high accuracy thresholds, where the latter may struggle to converge.

Regarding execution times, Scope GS can be conceptualized as a serial computation where times are recorded cumulatively: the final duration is the sum of the processing times of each individual cell. Conversely, Scope GM can be described as a parallel computation, where cells are processed simultaneously; in this case, the total execution time corresponds to the maximum duration recorded among the cells. For this reason, in Figure 5.4, Scope GS is referred to as Scope GS (SER), while Scope GM is labeled as Scope GM (PAR). This distinction highlights the difference

between serial and parallel computing approaches.

5.5 Application of TeraStat2 for Job Execution with Large Datasets

In the implementation details provided in Section 4.2, we describe how the SOP can be solved using the Gurobi Optimizer Python library under an academic free license⁽⁶⁾. The optimizer returns the optimal solution within a parametric time budget, set to 10 minutes by default, after which the current best solution is returned. To evaluate the performance of our method beyond this time constraint, we executed our jobs on TeraStat2, the supercomputing cluster hosted by the Department of Statistical Science.

TeraStat2 (TS2) is the general-purpose supercomputing cluster of the Department of Statistical Sciences, designed for solving mathematical and statistical models on Big Data. As a high-capacity system, the infrastructure is accessed via batch job submission. Users provide a formatted description of their experiment along with all necessary materials (e.g., input data). This description is submitted to a queuing system on TS2, which manages resource allocation and tracks consumption. Once sufficient resources become available, the job is executed automatically. Upon completion, the user is notified via email and can reconnect to TS2 to retrieve the results.

The system consists of 12 compute nodes running Linux, each equipped with two AMD Epyc 7452 processors and 256GB of RAM. Additionally, there are 12 compute nodes featuring two 8-Core Intel Xeon E5-2630 processors (16 virtual cores) and 64GB of RAM; notably, one of these latter nodes is equipped with 512GB of RAM. In total, the system provides 1,920 cores. The cluster also includes 76TB of shared storage and an InfiniBand QDR fiber-optic network, supporting a bandwidth of 40Gb/s both between nodes and between nodes and storage.

For sample sizes exceeding one thousand instances, we conducted benchmarks to test execution times across a limited set of features. As a use case, we again consider the Titanic, Compas and Credit datasets. For the Compas dataset, the standard Scope method requires an estimation when dealing with the full scale version and an increase of trees in the Random forest classifier. The supercomputing cluster allows us to overcome time constraints encountered when solving complex mixed integer optimization problems. Specifically for the Compas dataset, we design custom workflows to bypass the algorithmic difficulties inherent in the standard tools defined in Section 5.3.

The Scope algorithm requires several hours to achieve high levels of accuracy on the Compas dataset; these results, obtained via the High-Performance Computing (HPC) cluster are detailed in table .2 (in Appendix) and figure 5.4. As shown in these results, execution times for the Compas dataset rise from 1 to 11 hours, highlighting the need for a scalable approach, as discussed in Section 5.4.

5.6 Analysis of the Time Complexity and Efficiency of Scope

In the previous sections we introduced the Scope technique⁽⁵⁾ defining it as an ASD method comparable to DivExplorer⁽⁷⁾ and Slice Finder⁽¹⁷⁾. This section provides a detailed analysis of Scope's performance and efficiency, following the evaluation standards established for the aforementioned methods.

Figure 5.4 presents the execution times for each classifier across various datasets.

As shown in table .2 (in Appendix) classification accuracy varies across different models even when applied to the same dataset. The execution times reported in the table exhibit significant variance, influenced by the specific dataset, the chosen classifier, and the accuracy threshold. Generally, looking at figure 5.2 and figure 5.4 a clear trend emerges: as the accuracy threshold increases, the size of the identified subspace (the number of enclosed points) tends to decrease. Conversely, execution times typically increase with higher thresholds, although some exceptions occur.

Figure 5.5 illustrates the relationship between the coverage of data points (the percentage of the total sample) and the required accuracy threshold across different classification models and datasets.

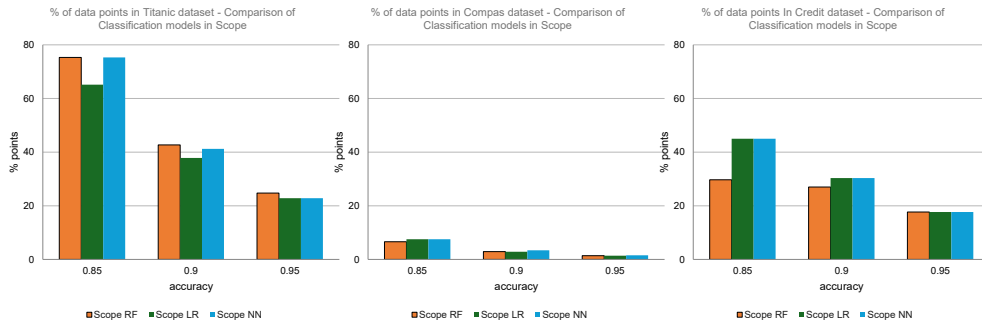


Figure 5.5 The percentage of points across the Titanic, Compas and Credit dataset: a comparison of classification models in Scope.

A notable observation is the consistent trend in coverage relative to the classification models for both the Compas and Credit datasets. Specifically, at any given accuracy threshold for these two datasets Random Forest yields the lowest coverage of data points, Logistic Regression results in a higher coverage than Random Forest, Neural Networks achieve the highest percentage of enclosed data points. The Titanic dataset follows a similar pattern regarding Random Forest and Neural Networks. However, Logistic Regression deviates from the previous trend by producing a lower coverage on this specific dataset. Furthermore, the overall proportion of data

points varies significantly depending on the dataset. These variations reflect both the dataset's size and the inherent complexity of the classification task, as higher accuracy requirements naturally constrain the size of the identifiable subspaces.

Figure 5.4 shows that execution times vary widely across the results, and no clear patterns emerge. For example, in the Titanic dataset, Random Forest exhibits the worst performance in terms of time at the highest accuracy levels. Conversely, in the Compas dataset, Logistic Regression records the longest execution times, while the Credit dataset shows no significant differences among the classifiers.

While the grid-based version of Scope offers significant enhancements in point density and execution speed, DivExplorer retains a performance advantage in computational time, due to the specific frequent pattern mining (FPM) approach.

5.7 Algorithms in Action

We define the ASD techniques as model-agnostic. Consequently a Random Forest classifier is chosen for training after splitting the original data to obtain predictions for the target variable in the test dataset. Our goal is to compare the specific outcomes of the two subgroup detection techniques, aiming to discover whether they identify congruent groups of instances relative to the accuracy threshold. Furthermore, we intend to conduct a comparison of the available tools of the two methods to find the most useful ones for characterizing the identified instances. Consistent with the previous section, we proceed to apply the methods to our datasets, which vary according to the number of data points, accuracy, and items. Subsequently, we identify the optimal subspace with Scope corresponding to the subsets discovered by DivExplorer.

For comparing Scope and DivExplorer discretization of numerical features is required: as an example we consider Titanic dataset and transform Age and Fare in intervals in order to obtain a symmetrical and balanced distribution: many techniques can be used in this case, but for the moment we do not focus on optimal granularity⁽⁷⁾.

Table 5.2 displays the top five divergent patterns with respect to accuracy (ac) for the Titanic dataset, where the support threshold is fixed to 0.10. Divergence, length of itemsets, relative support and support count are also shown. A redundancy pruning approach is performed post exploration.

support	itemset	length	count	accuracy	acc. div.
0.11	(Sex=1, Fare=over 30)	2	29	1.00	0.18
0.13	(Pclass=2, SibSp=0)	2	35	0.94	0.12
0.12	(Pclass=2, Sex=0)	2	32	0.94	0.11
0.21	(Fare=up to 10, Age=25-45)	2	57	0.93	0.11
0.20	(Pclass=2)	1	54	0.93	0.10

Table 5.2 Top 5 divergent patterns with respect to accuracy for the Titanic dataset with support = 0.10 and redundancy pruning.

The first itemset of table 5.2 includes female individuals having paid an high Fare: in this subset there are 29 data points and the support is 0.11, which is over the 10% threshold. The accuracy is 1, and the divergence is 0.18 (the difference between the

subset's accuracy and the overall accuracy). The second itemset includes individuals in medium class and without siblings or spouses. In this subset the accuracy is 0.94 and the divergence is 0.12, while the support is 0.13 corresponding to 35 data points. The third itemset refers to 32 males in a medium class, accuracy is 0.94, the divergence is 0.11, while the support is equal to 0.12, corresponding to 32 data points.

As we illustrated in the previous section one of the most important differences between DivExplorer and Scope is the discretization of numerical attributes in the preprocessing phase, which assumes only a finite, small set of values. Scope doesn't require transformations for numerical attributes and permits an interval scale. Discretization appears to worsen performances of the Scope toolkit, which is based on a mixed integer optimization method⁽⁵⁾.

In table 5.3 we show the optimal hyper rectangular subspace obtained with Scope for the Titanic dataset using an accuracy threshold of 0.95. The maximum number of data points is 66, with 63 correctly predicted. We choose this solution due to its similarity in accuracy to the second itemset of DivExplorer, even though finding many analogies seems difficult.

attributes	original lower	lower bounds	upper bounds	original upper
Pclass	1	1	3	3
Sex	0	0	1	1
Age	0	21	29	80
Fare	0	7	15	512
SibSp	0	0	8	8
Parch	0	0	1	6

Table 5.3 Optimal subspace obtained with Scope for the Titanic dataset, with threshold for accuracy = 0.95.

In the Scope solution represented in table 5.3 passengers' Age is between 21 and 29 years. The interval found by Scope for Fare is between 7 and 15 dollars while the number of parents and children is either none or only one. The comparison between the Divexplorer pruned solution and the Scope solution, at the accuracy threshold of 0.95, does not show analogies in the resulting optimal subsets.

In 3.1 Shapley coalition game theory is mentioned as a means to find factors influencing an individual prediction (in Scope) and to estimate the contribution of each attribute value to the subgroup divergence (in DivExplorer). Methods for attributing the divergence of an itemset to its items and for estimating the overall impact of an item are available in DivExplorer.

In figure 5.6 we represent the items influencing the divergence of the second itemset of table 5.2, ordered by importance: these contributions are responsible for the divergence of a specific itemset, but they refer to a single item (for example a medium passenger class). The medium passengers' class is the most important factor contributing to divergence (0.10), and being without siblings or spouses is the second (0.02). All the contributions sum up to the total divergence of the second itemset (0.12). Also Global Shapley values are represented (overall impact of an item).

In figure 5.7 SHAP values found in the optimal subspace with Scope for Titanic dataset are displayed. For comparison the summary plot of SHAP values for the

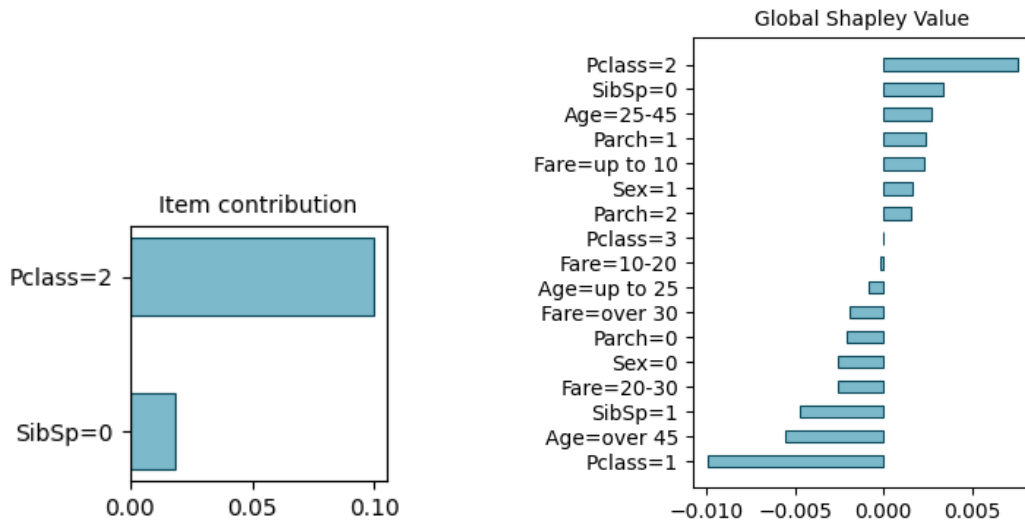


Figure 5.6 Shapley item contributions of the second divergent pattern (left) and overall impact of an item for the Titanic dataset (right) with support = 0.10.

entire dataset is reported, illustrating the global ranking of features. It is evident that within the optimal subspace, Fare emerges as a more significant factor; indeed, it serves as a defining feature for the ranges within this subspace. It can be difficult to compare these values with the contribute of an item in DivExplorer, because Scope evaluates the interval scale of attributes to find the individual impact of a single variable on predictions whereas in DivExplorer Shapley contributions represent a specific item belonging to an itemset.

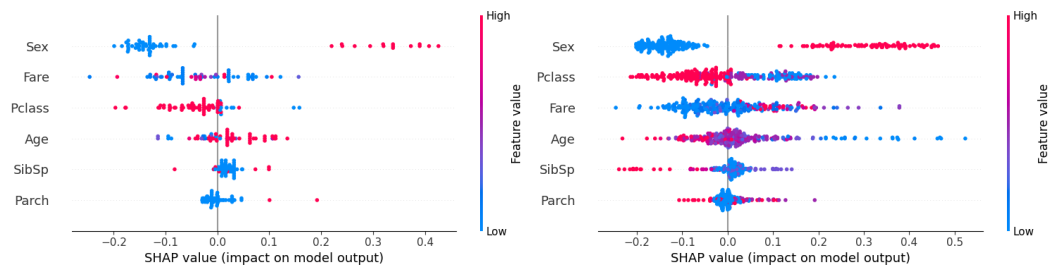


Figure 5.7 SHAP values of the individuals in the optimal subspace found with Scope for the Titanic dataset and threshold for accuracy = 0.95 (left) and SHAP values for the entire dataset (right).

In the plot we can see that features are ordered by their importance, higher values of the attribute Sex lead to an increase of the prediction value while higher values of Pclass cause a decrease of the prediction value. The ranking of feature importance reflects the set of features utilized to define the subspace in the Scope method. This knowledge represents a valuable resource in Scope, while DivExplorer aims only at finding critical or optimal subsets and does not provide information about the individual impact on predictions as can be obtained with individual SHAP values.

We proceed with the application of the methodologies to our datasets, selecting the well known Compas dataset containing criminal history and demographic information of defendants. For the Compas dataset a random sample without replacement is extracted for reducing the size and for avoiding difficulties in managing datasets with thousands of instances. The sampling process is necessary to compare Scope with DivExplorer, as the latter is unable to achieve an accuracy above 0.85 when processing the entire dataset, aside from the use of TeraStat2, described in Section 5.5. This dataset contains criminal history and demographic information of defendants. For performing DivExplorer, discretization of numerical features is required to transform age, the number of priors offences and decile score in intervals.

Table 5.4 displays the top 5 divergent patterns with respect to accuracy identified within the Compas dataset. This analysis is conducted by fixing the support threshold for the support at 0.10. For each pattern the table reports the computed divergence metric, the accuracy, the length of the itemsets, the relative support and the support count. A redundancy pruning approach is performed post exploration.

support	itemset	length	count	accuracy	acc. div.
0.11	(race_african=0, decile_score=1-4, age=over 45)	3	35	0.86	0.20
0.11	(age=up to 25, race_african=1, c_charge_degree=1)	3	35	0.80	0.14
0.15	(decile_score=1-4, age=over 45)	2	49	0.80	0.14
0.13	(race_african=0, age=over 45)	2	43	0.79	0.14
0.11	(sex=0, age=up to 25, race_african=1)	3	36	0.78	0.12

Table 5.4 Top 5 divergent patterns with respect to accuracy for the Compas dataset with support = 0.10 and redundancy pruning.

The analysis of the divergent patterns in table 5.4 reveals that the first itemset characterizes individuals whose decile score is between 1 and 4, whose race is not African and whose age is over 45 years. This subset contains 35 data points, resulting in a support of 0.11, which exceeds the 10% threshold. The accuracy within this subset is 0.86, yielding a divergence of 0.20 (calculated as the difference between the subset's accuracy and the overall accuracy). The second itemset describes African individuals whose age is below 25 years and whose charge type is felony. The accuracy within this subset is 0.80 with a divergence of 0.14. The support is 0.11 corresponding to 35 data points. The third itemset includes individuals with a low individual score (1-4) and an age over 45 years. This subset comprises 49 data points, resulting in a support of 0.15, again exceeding the 10% threshold. The accuracy is 0.80 and the divergence is 0.14.

Table 5.5 presents the optimal hyper rectangular subspace obtained using Scope for the Compas dataset, subject to an accuracy threshold of 0.85. The chosen optimal solution involves a maximum of 42 data points, of which 36 are correctly predicted. This solution is selected specifically to facilitate a comparison with the first itemset discovered by DivExplorer, in an attempt to identify potential analogies or overlaps between the two methods.

The subspace obtained by the Scope methodology is defined by the following constraints: age is between 22 and 36 years, the number of priors offences is between 5 and 22 and the decile score is between 4 and 9. These parameter ranges are not directly comparable with the first itemset discovered by DivExplorer (table 5.4),

attributes	original lower	lower bounds	upper bounds	original upper
age	19	22	36	96
sex	0	0	1	1
priors_count	0	5	22	33
c_charge_degree	0	0	1	1
race_african	0	0	1	1
decile_score	1	4	9	10

Table 5.5 Optimal subspace obtained with Scope for the Compas dataset, with threshold for accuracy = 0.85.

where the criteria are: age over 45 and a decile score between 1 and 4. We can only observe that Age and Decile score appear to be important factors influencing the outcomes of both methods. Curiously the number of priors offenses is not included as a defining feature in the DivExplorer solution, but is a primary factor in the Scope solution.

Figure 5.8 illustrates the items influencing the divergence of the first itemset from table 5.4, ordered by their contribution magnitude. These contributions indicate the isolated impact of a single item (e.g., age over 45) on the total divergence of the specific itemset. Age is the most significant factor, contributing 0.14 to divergence. The low decile score contributes 0.04. The sum of all individual contributions equals to the total divergence of the first itemset (0.20). Also Global Shapley values are represented (overall impact of an item).

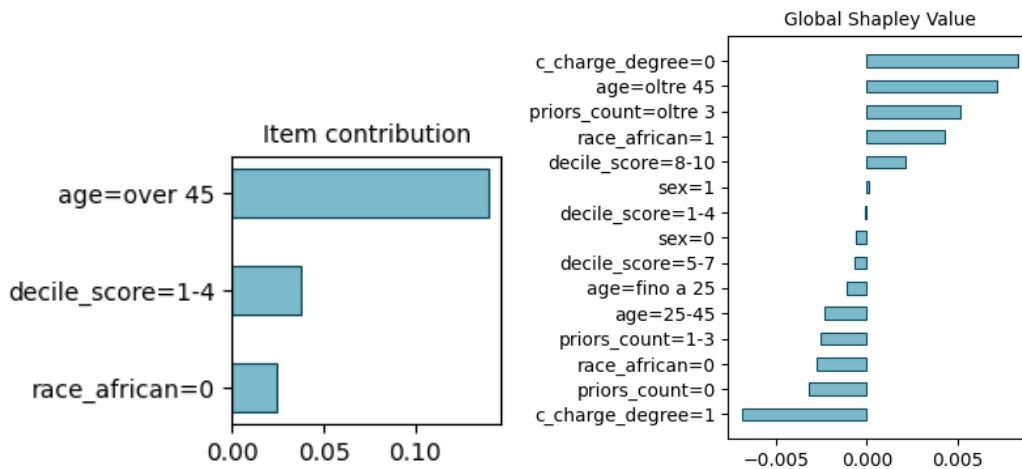


Figure 5.8 Shapley item contributions of the first divergent pattern (left) and overall impact of an item for the Compas dataset (right) with support = 0.10.

Figure 5.9 presents the SHAP values for the optimal subspace identified by Scope within the Compas dataset. It is important to note that these values represent the individual impact of a single variable on specific predictions; in contrast, Shapley contributions in DivExplorer refer to the contribution to the divergence of a specific item within an itemset. The feature importance ranking aligns with the set of features utilized to define the subspace in the Scope method. This result represents a key departure from DivExplorer, where contributions do not provide direct insight

into the impact on predictions in the same way individual SHAP values do. To facilitate comparison, a summary plot of SHAP values for the entire dataset is provided, illustrating the global ranking of features. The ranking remains the same for the most important features; however, race is ranked higher than sex, which falls to the last position. Features are ordered by importance, for instance a higher number of priors correlates with an increase in the predicted value.

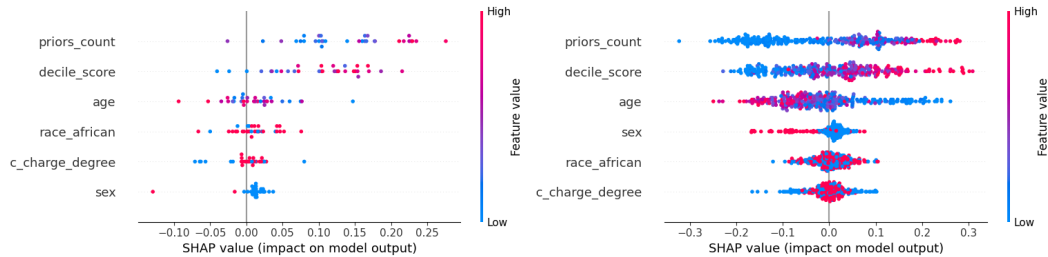


Figure 5.9 SHAP values of the individuals in the optimal subspace found with Scope for the Compas dataset and threshold for accuracy = 0.85 (left) and SHAP values for the entire dataset (right).

The same analysis is reported for the Credit dataset. Table 5.6 displays the top 5 divergent patterns with respect to accuracy identified within the Credit dataset. This analysis is conducted by fixing the support threshold for the support at 0.10. For each pattern the table reports the computed divergence metric, the accuracy, the length of the itemsets, the relative support and the support count. A redundancy pruning approach is performed post exploration.

support	itemset	length	count	accuracy	acc. div.
0.11	(Job=3, Duration=up to 25)	2	33	0.85	0.17
0.10	(Sex=0, Job=3, Age=25-45)	3	30	0.83	0.15
0.13	(Duration=up to 25, Amount=up to 5, Age=over 45)	3	40	0.83	0.15
0.15	(Duration=up to 25, Age=over 45)	2	46	0.80	0.12
0.15	(Job=3)	1	46	0.80	0.12

Table 5.6 Top 5 divergent patterns with respect to accuracy for the Credit dataset with support = 0.10 and redundancy pruning.

Table 5.7 presents the optimal hyper rectangular subspace obtained using Scope for the Compas dataset, subject to an accuracy threshold of 0.95. This solution is selected specifically to facilitate a comparison with the first itemset discovered by DivExplorer, in an attempt to identify potential analogies or overlaps between the two methods.

Figure 5.10 illustrates the items influencing the divergence of the first itemset from table 5.7, ordered by their contribution magnitude. These contributions indicate the isolated impact of a single item on the total divergence of the specific itemset. Also Global Shapley values are represented (overall impact of an item).

Figure 5.11 presents the SHAP values for the optimal subspace identified by Scope within the Compas dataset. It is important to note that these values represent the individual impact of a single variable on specific predictions; in contrast, Shapley contributions in DivExplorer refer to the contribution to the divergence of a specific

attributes	original lower	lower bounds	upper bounds	original upper
Age	19	35	59	75
Sex	0	0	1	1
Job	0	0	3	3
Amount	0	1	4	18
Duration	4	9	30	72

Table 5.7 Optimal subspace obtained with Scope for the Credit dataset, with threshold for accuracy = 0.95.

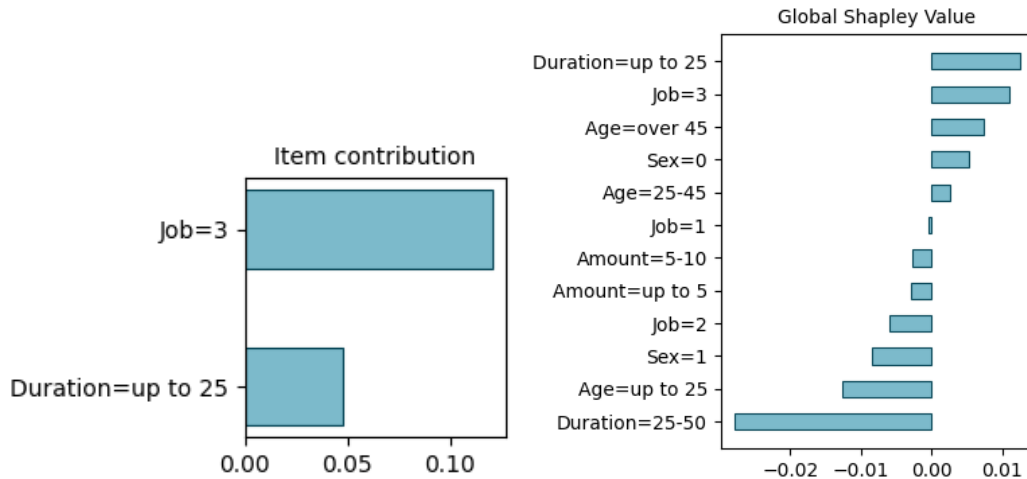


Figure 5.10 Shapley item contributions of the first divergent pattern (left) and overall impact of an item for the Credit dataset (right) with support = 0.10.

item within an itemset. To facilitate comparison, a summary plot of SHAP values for the entire dataset is provided, illustrating the global ranking of features.

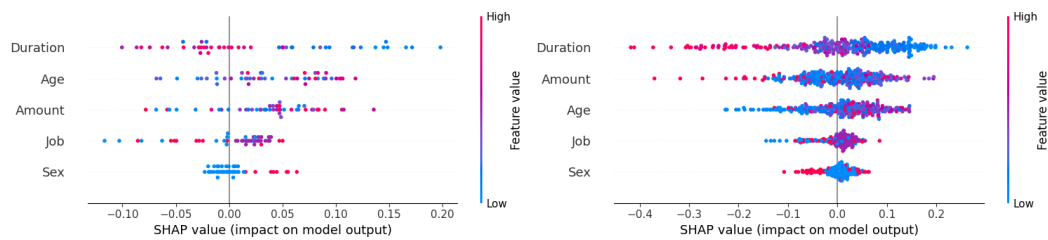


Figure 5.11 SHAP values of the individuals in the optimal subspace found with Scope for the Credit dataset and threshold for accuracy = 0.95 (left) and SHAP values for the entire dataset (right).

Chapter 6

Administrative Datasets and Evidence Based Decision Making in the Public Sector

Administrative datasets contain strategic and highly valuable information about key real-world phenomena. Capturing this knowledge is both challenging and extremely useful for decision-making, as well as for researchers and the public at large. However, extracting meaningful insights from such data is often difficult, since many administrative archives are not designed for analytical purposes but solely for administrative use. As a result, they frequently contain missing values, approximate or unstructured information, and data quality may be insufficient. Processing these datasets for statistical analysis or classification tasks can therefore be demanding. Nevertheless, recent advances in data-driven methodologies have made it increasingly feasible to address these challenges. Despite this progress, great care must be taken to properly understand and explain analytical outcomes in order to avoid misinterpretations or, even worse, unintended consequences. Indeed, data-driven decision-making can have a significant impact on real-world outcomes, particularly in sensitive and critical domains.

In Section 6.1 an example of a sensitive classification task is presented, concerning judicial data on re-admissions to penal institutions after release. Section 6.2 describes a critical approach to Compas recidivism classification and Section 6.3 introduces the article on Criminal Recidivism produced in collaboration with Istat researchers.

6.1 Model Explainability and Pattern Discovery in Judicial Records

This thesis aims to identify novel methods and techniques for discovering interpretable patterns in administrative data and for formalizing the concept of model explainability. To this end, a sample extracted from an Italian criminal recidivism dataset is analyzed. The dataset is proprietary and confidential, subject to social uncertainty, and its details cannot be disclosed. Recidivism is the target variable and is represented as a one-hot encoded feature, while the remaining features are

either integer-valued or one-hot encoded, as the dataset contains both numerical and categorical features. The most important features in the analysis of criminal history (re-admissions after release) include age at entry, number of previous incarcerations, duration of imprisonment, citizenship and treatment of prisoners. Other relevant factors are gender, as well as the type and location of the penal institution.

Individual-level predictions are not considered a final objective due to the inherent challenges in classifying social behavior and the risks associated with false positive detections. Currently, only general information on re-incarceration rates is analyzed, incorporating personal, territorial, and judicial features. Consistent with the previous chapters, the focus in this is placed on explainability and exploratory analysis, acknowledging that judicial data present inherent limitations and ethical concerns. Although such data are often affected by anomalies and missing values, in this application they have already been preprocessed to improve data quality.

The recidivism data were split into training and test sets, as is common practice in supervised learning. Statistical and machine learning methods are useful for computing feature importance, which provides a summary of the information conveyed by the dataset. Such information can support public policy decisions; however, in the social sciences, defining reliable models is challenging, and making accurate predictions is often difficult. The Random Forest model achieved an accuracy of 0.65 on the criminal behavior test set. However, the recall for recidivism is relatively low (0.38), whereas the recall for non-recidivism is 0.78. This indicates that the algorithm is significantly less effective at identifying recidivist individuals than non-recidivist ones, a discrepancy stemming from a distributional imbalance within the target labels. Indeed, the data shows that recidivists represent 32.59% of the sample, whereas non-recidivists constitute the remaining 67.41%. This finding contrasts with the practical use of predictive models in judicial contexts.

The number of previous offenses exhibits a right-skewed distribution, with a low mode. Most individuals do not have previous offenses, while only a small fraction have one or more. Additionally, the duration of imprisonment exhibits a right-skewed distribution, with a mode concentrated at the lower end. Women constitute only a minority of the population, whereas foreign prisoners represent a relatively large proportion. Accuracy does not vary substantially across classifiers: it is equal to 0.70 for both the Logistic Regression and Neural Network models, with only negligible differences. Both the features and the target exhibit skewness and imbalance, leading to poor performance across all classification models, regardless of the algorithm used.

An application of Scope to recidivism data. For this case study, Scope was applied to the real-world dataset on recidivism in Italian penal institutions. Predicting recidivism is a complex challenge. This approach enables us to evaluate the inherent difficulty of forecasting criminal behavior across specific subgroups. Scope identifies a subspace with accuracy exceeding the 0.85 threshold (while maximizing the number of points) when age and duration are high and the number of entries is low. Conversely, when accuracy falls below 0.50, age and duration are low and the number of entries is high. The analysis shows, for instance, that when the number of previous incarcerations is high while age at entry and duration of imprisonment are low, accuracy decreases, but recall for recidivists improves compared to the baseline. When the number of previous incarcerations is low and both age at entry and

duration of imprisonment are high, accuracy increases, and recall for not recidivists improves.

Regarding feature importance, calculated using Tree SHAP values, the number of entries and age at entry are the most significant predictors, followed by citizenship and duration of imprisonment. However, these rankings may vary when analyzing specific subgroups, for example in high accuracy subgroups like the one reported in figure 6.1.

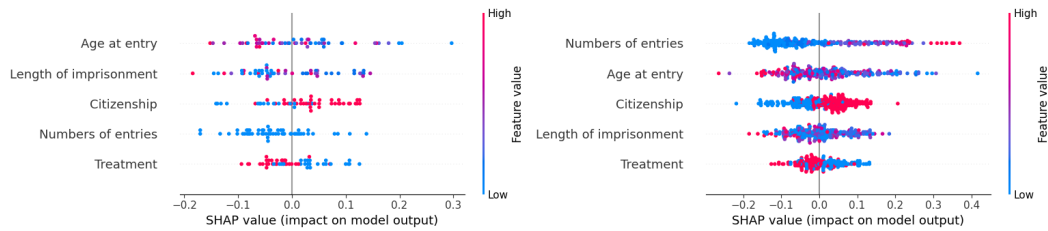


Figure 6.1 SHAP values of the individuals in the optimal subspace found with Scope for the Recidivism dataset and threshold for accuracy = 0.85 (left) and SHAP values for the entire dataset (right).

Scope can be viewed as a method for detecting a strong treatment effect. This algorithm is particularly effective at identifying subsets that respond best to specific actions, ensuring a predefined accuracy threshold is maintained for prediction tasks. For example, in the context of criminal recidivism, it is crucial to understand the effect of rehabilitation programs, including cultural, social, vocational activities and prisoner employment, on re-incarceration rates. By applying this algorithm and setting a specific accuracy threshold, we are able to identify the subgroups that demonstrate the strongest response to these interventions.

6.2 Predicting modeling. A Critical Approach to Compas Recidivism Classification

The Compas algorithm utilizes a risk score to inform judicial decisions regarding the pretrial release or detention of defendants. Within this framework, recidivism is defined as a new criminal offense resulting in a jail booking that occurs after the initial assessment. The judicial decision relies on a predictive probability: what is the likelihood that the defendant will commit a sufficiently serious crime if released before trial? While it is statistically true that individuals with extensive prior criminal histories are, on average, more likely to reoffend, predictive models are prone to significant errors

Automated Subgroup Detection techniques provide deep insights into data, helping to mitigate errors that carry profound real-world consequences, such as those found in recidivism prediction. Disparities in false positive rates across specific subgroups were analyzed using FairVis⁽¹⁶⁾, a visual analytics system designed to discover intersectional bias in machine learning. This analysis focuses on a classification model where the predicted labels align with the Compas scoring system (categorized into high and medium-low risk classes). The model incorporates features such as age, sex, race, duration of stay, criminal charges and prior offense counts.

In the Compas dataset, instances represent defendants assigned a probability of recidivism, while the target variable is a binary classification of actual recidivism over a two-year observation period. As described in⁽¹⁶⁾, a high false positive rate for this model indicates that many low-risk individuals (who might be good candidates for release on bail) are labeled as high-risk. If this model is used to inform decisions on release, these false positives correspond to low-risk candidates who might be denied bail.

The analysis conducted using FairVis reveals that, due to imbalances in the true labels, subgroup bias according to race can be observed. For example, if the model were to make a single prediction for the entire subgroup of African-American males, labeling the subgroup as positive (predicting high recidivism risk) could result in higher overall accuracy for this subgroup than for others. Even less extreme scenarios may hold: to maximize accuracy for this subgroup, the model may assign a disproportionately higher number of positive labels than negative labels. Further studies⁽¹²⁾ suggest that racial bias may be strongly linked to age within the algorithm, as these factors are generally interconnected. This highlights the need for greater transparency and accountability in the deployment of such systems, to understand, for example, how age interacts with other factors to produce biased outcomes.

Our approach to the Compas dataset. In our analysis on Compas dataset in Chapter 5, a data-driven model is used to investigate how, and to what extent, recidivism may depend on features such as age, sex, race, charge, decile score, and prior counts. The model aims to predict recidivism based on the available information while evaluating feature importance. Accordingly, the Compas score is treated as one feature among others, contributing to the predictive task rather than being considered in isolation. Our focus is not on individual predictions but on identifying the factors that influence the target outcomes. Real-world datasets such as Compas are challenging to analyze because predictive accuracy is often limited: the relationship between features and the target is relatively weak. Understanding which instances are predicted correctly can be highly informative, but this is often affected by imbalances in the true labels across subgroups.

Interpretable models can often perform approximately as well as black-box models in various recidivism prediction tasks, as described in⁽²⁾ and⁽⁸⁾. Using interpretable models does not cause practical loss in accuracy, while providing significant benefits in terms of transparency and interpretability. Recidivism prediction models constructed using data from a single location do not generally perform as well when applied to another location. This suggests that models should be trained on data from the location where they are intended to be used and should be updated periodically over time.

6.3 Predictive Models for Criminal Recidivism

This section presents the article "Criminal Recidivism. Towards Reliable and Transparent Predictive Models"⁽¹⁾, published on the Italian Journal of Economic, Demographic and Statistical Studies in 2026. The following subsections report the introduction, the data and methods, the outcomes and the conclusions of the accepted paper presented exactly as submitted.

6.3.1 Introduction

Recidivism is broadly defined as the tendency of previously convicted individuals to reoffend, serving as a key indicator of risk, social dangerousness, and the effectiveness of penal and rehabilitative measures. Identifying the factors associated with reoffending as well as understanding their interactions can support prevention strategies, resource allocation, and evidence-based decision-making in the justice system.

Definitions and measurements of recidivism vary across jurisdictions. Some systems consider only new convictions while others include arrests or reports. In Italy, recidivism is regulated by articles 99–105 of the Penal Code which influence sentencing and enforcement. International organizations such as the United Nations Office on Drugs and Crime and the Council of Europe adopt different indicators depending on whether the focus is on policy evaluation, reintegration, or risk assessment.

Initial predictive efforts relied on static models using demographic and criminal history data. Since the 1970s, attention shifted toward dynamic models that incorporate contextual and modifiable variables. In the last two decades, the development of machine learning (ML) has enabled more complex, yet less transparent, systems⁽²⁾ and⁽⁸⁾, underscoring the need for interpretable tools to guarantee transparency for legal actors and the public.

From a legal perspective, there is a clear divergence between common law and civil law systems. In Italy, Article 220 of the Code of Criminal Procedure prohibits expert assessments on personality or criminal propensity during trial, thereby restricting the use of AI tools to post-sentencing phases. Conversely, in common law systems such as the U.S., tools like COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) are routinely integrated into judicial decision-making in several States. Common law jurisdictions are increasingly adopting principles of fairness, accountability, and transparency, whereas civil law countries require a more cautious, rights-based approach. These institutional differences shape how ML systems are introduced into the justice sector and raise important questions about balancing predictive efficiency with the protection of individual rights.

The European Union’s AI Act (Regulation EU 2024/1689)¹, adopted on August 1, 2024, establishes a harmonized framework based on four levels of risk. Recidivism prediction tools categorized as “high risk” are subject to rigorous standards concerning transparency, reliability, fairness, and continuous oversight. While the AI Act seeks to facilitate the European digital single market, it also allows national authorities to tailor implementation to local legal traditions, ensuring respect for fundamental rights.

6.3.2 Data and Methods

This study relies on the COMPAS dataset published by a non-profit investigative journalism organization², which contains information on more than 7,000 individuals arrested in Broward County, Florida. It includes demographic data, criminal history,

¹<https://eur-lex.europa.eu/eli/reg/2024/1689/oj>

²<https://www.kaggle.com/datasets/mlethics/compas>

risk scores and a binary recidivism outcome within two years. We focused on key features: age, decile score, prior offenses, ethnicity (origins), gender, and offense type. Moderate linear correlations are observed between age and decile score (0.39), prior offenses and decile score (0.43), decile score and recidivism (0.35). The decile score, a composite risk indicator of recidivism correlates with reoffending as expected. The dataset showed issues of racial bias⁽²³⁾ and misclassification risks⁽²⁾. Instead of predicting individual risk scores, our aim is to investigate the relative importance of features and their contribution to the overall decision-making process. Despite ethical and legal constraints⁽¹²⁾, we argue that ML can elucidate complex crime patterns in data on convicts/detainees and guide targeted interventions.

Our approach combines a data-driven methodology using machine learning models like Random Forests and Neural Networks for predictive pattern detection, alongside interpretable models such as Logistic Regression to enhance transparency. Our goal is to critically evaluate the reliability and interpretability of these models. All models were implemented in Python using specific packages: *sklearn.preprocessing*, *modelselection*, *sklearn.metrics*, *sklearn.linearmodel*, *sklearn.ensemble*, *keras*, *sklearn.cluster*, *kmodes.prototypes*, *shap*.

Because the dataset contains features with different units of measure, variables were standardized to ensure comparability. In our preprocessing, we applied min-max normalization. Given potential heterogeneity in model performance, maximizing accuracy supports the identification of local explanations and the ranking of feature importance.

For the Logistic Regression model, we used default parameters from the linear model package. The Random Forest model employed 100 trees from the "ensemble" package, with other hyperparameters left unchanged; increasing this number could improve performance but would increase model complexity. For example, a randomized search ("RandomizedSearchCV") might identify a model with similar accuracy using 385 trees. The Neural Network is sequential, with three layers consisted of 16, 16, and 1 neuron(s); the first two layers used Rectified Linear Unit (ReLU) activation, while the final layer used sigmoid function. The optimizer was Adam, and the loss function was binary-crossentropy. The same test set served as validation.

We split the COMPAS dataset into training and test sets to evaluate model performance and to compute accuracy on the test data, in order to reduce the risk of overfitting. We focused on supervised learning approach for classification, where input features (labelled data) were mapped to discrete outcome categories. We used the test set to assess model capacity in the population (Hold-out procedure). We fitted our data with Random Forest, a non-parametric supervised learning method for non-linear relationships defined as an ensemble model, the result of aggregating a set of Decision Trees; it avoids local optima and correlation, Decision Trees limitations, through bagging and features selection. We also applied Neural networks (thousands of simple nonlinear models that work together, very difficult to interpret) and Logistic regression for comparing classification results, as they are representative in the trade-off between interpretability and accuracy⁽⁸⁾.

With a binary target we analysed the classification report and the confusion matrix, focusing on the most important metrics: accuracy, recall, precision, F1-score and specificity. The confusion matrix is created by setting an assignment threshold. The decision threshold or cutoff point, generally set at 0.5, is a critical value used

to convert the output of a classification model into a class prediction, since many algorithms return a probability score indicating the likelihood that an input belongs to the positive class, the threshold determines the cutoff point for classification. We obtained feature importance through the SHAP values related to each classifier, a powerful method for explaining the predictions of any machine learning model representing how much each feature contributes to a particular prediction for a given instance, given the expected baseline. We also analysed the classifiers behaviour in subgroups, to verify if accuracy is homogeneous in these subsets. To this end, we applied unsupervised classification to group the N units into clusters, to ensure that units within the same cluster displayed homogeneity.

We employed K-means to partition observations by minimizing within-cluster variance and maximizing separation between groups. To avoid instability associated with random initialization, we adopted the more robust K-means++ method. This approach improves convergence and clustering quality by selecting initial centroids with probabilities proportional to their squared distance from those already chosen, ensuring more diverse and informative starting points.

In order to select the optimal number of clusters K, we computed the Silhouette Score 6.1, which evaluates how well each unit fits its assigned cluster (cohesion) compared to the nearest alternative cluster (separation). The score ranges from -1 to 1 , with higher values indicating better-defined clusters.

$$s(i) = b(i) - a(i) / \max \{b(i), a(i)\} \quad (6.1)$$

where $a(i)$ is the average distance of each unit i to all the other units in its assigned cluster, and $b(i)$ is the average distance of unit i to the units in the nearest cluster to which it was not assigned. The optimal K was selected by maximizing the silhouette score. A hierarchical solution with Ward's method confronted at the same K produced similar values, supporting robustness, while K-means++ achieved slightly higher silhouette values, indicating improved stability and separation. By clustering similar instances, we identified subgroups differentiated by accuracy, and we analysed each cluster in terms of label imbalance, evaluation metrics, and SHAP-based feature importance ⁽²²⁾. SHAP provides a linear explanation model in which binary variables attribute an effect to each feature; the sum of these contributions approximates the original model's output $f(x)$ 6.2.

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i \quad (6.2)$$

where $z \in \{0,1\}^M$, M is the number of simplified input features, $\phi_i \in R$, f is the original prediction model to be explained and g the explanation model. We compared subgroup metrics with the overall situation on the entire test set to underline differences and define each subgroup peculiarity: the idea is that the defined subgroup may highlight regions where the problem is more or less accurate.

6.3.3 Outcomes

The models show (Table 6.2) accuracies of 0.68 (Logistic Regression), 0.67 (Neural Networks), and 0.63 (Random Forest), with the latter performing worst. To assess

overfitting, performance variability was evaluated via Cross-Validation (CV), which revealed a standard deviation (std) of 0.02 across all models, thus excluding significant overfitting.

Models		Precision	Recall	F1-score	Support
Logistic Regression	negative	0.69	0.78	0.73	1,189
	positive	0.68	0.57	0.62	976
	Accuracy			0.68	2,165
	Macro avg	0.68	0.67	0.67	2,165
	Weighted avg	0.68	0.68	0.68	2,165
Neural Networks	negative	0.66	0.82	0.73	1,189
	positive	0.69	0.50	0.58	976
	Accuracy			0.67	2,165
	Macro avg	0.68	0.66	0.66	2,165
	Weighted avg	0.68	0.68	0.67	2,165
Random Forest	negative	0.66	0.69	0.67	1,189
	positive	0.60	0.56	0.58	976
	Accuracy			0.63	2,165
	Macro avg	0.63	0.63	0.63	2,165
	Weighted avg	0.63	0.63	0.63	2,165

Figure 6.2 Classification report, main metrics for each class (negative, positive). Logistic Regression, Neural Networks, Random Forest models; COMPAS dataset.

All models show higher F1-scores for the negative class, as recall is greater for non-recidivists. Conversely, precision exceeds recall for the positive class, indicating better control of false positives than false negatives. Given the higher risk of false positives in criminal recidivism, prioritizing precision reduces the chance of mislabelling non-recidivists as recidivists. In the COMPAS tool's trade-off, avoiding missed recidivists is considered more acceptable than penalizing non-recidivists. The Receiver Operating Characteristic-Area under the Curve (ROC-AUC) values (0.74 for Logistic Regression and Neural Networks, 0.68 for Random Forest) support these findings; 0.74 is considered excellent for complex classification tasks. Selecting an optimal threshold requires analysing the confusion matrix to balance FP and TP where T and F denote true and false predicted values, respectively.

We obtained feature importance through the SHAP values (Figure 6.3), in general model agnostic while the actual computation for different model types often uses specific approximations or exact algorithms. The explainer for Random Forest is specifically designed for tree-based models and ensembles of trees (model specific) and can calculate SHAP values exactly or with high accuracy and efficiency, taking into account all possible paths. Due to the nature of tree models, Tree SHAP is particularly effective at capturing and quantifying interaction effects among features.

The confusion matrix (Figure 6.4) shows consistent patterns across models, except for Random Forest. It has a higher proportion of FPs (16.95%) and a lower TNs (37.97%), indicating a greater risk of overestimate the recidivism (false positive errors). In complex scenarios characterized by weak feature-target relationships or nearly indistinguishable classes, an AUC of 0.68 (as achieved by the Random

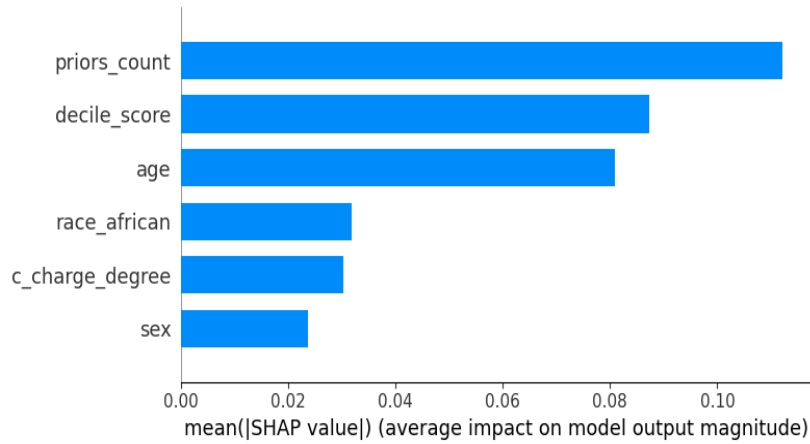


Figure 6.3 Feature importance based on the mean absolute SHAP values of the Random Forest model for the COMPAS dataset.

Forest model) may still be acceptable. Further analysis is required: evaluation of feature selection, hyperparameters optimization and classifier behaviour across specific subsets.

We grouped similar units, selecting a nine-cluster solution based on the peak Silhouette Score ($k=9$, 0.57), a value considered reasonable (Figure 6.5).

We compared K-means and K-prototypes for clustering numerical and binary features from the COMPAS dataset. The results showed no significant differences in classification accuracy across clusters, an essential focus of this study, nor in overall clustering quality ($k=8$, silhouette score=0.60). We selected K-means, which produced the widest accuracy range, and applied a Random Forest classifier to the resulting clusters (6.6). Cluster centroids and sizes were computed. Accuracy ranged from 0.54 (cluster 6) to 0.73 (cluster 3). Comparing the global metrics with those of cluster 3 with the highest accuracy, patterns emerged that helped identify factors potentially enhancing classification performance (Table 6.7).

The confusion matrix for cluster 3 versus the whole dataset (Figure 6.8) indicates improved recall for the negative class but decreased recall for the positive class, reflecting class imbalance (>70% non-recidivists). Consequently, FPs decrease (6.61 vs. 16.95), TNs increase (64.46 vs. 37.97), while TPs decline (8.26 vs. 25.40).

SHAP values in the most accurate clusters identified key features influencing model behaviour. In cluster 3, gender (specifically female) was more important than age, while origin was not relevant; this cluster mainly include Caucasian women with minor offenses and low risk scores (Figure 6.9). Cluster 8 (not shown), with similar profiles but of African origin, also demonstrated high accuracy (0.71). Conversely, cluster 6 (not shown): Caucasian men with serious offenses and high-risk scores, had the lowest accuracy (0.54), likely due to recidivism imbalance. Importantly, *ceteris paribus*, individuals of African origin (cluster 0) outperformed their Caucasian

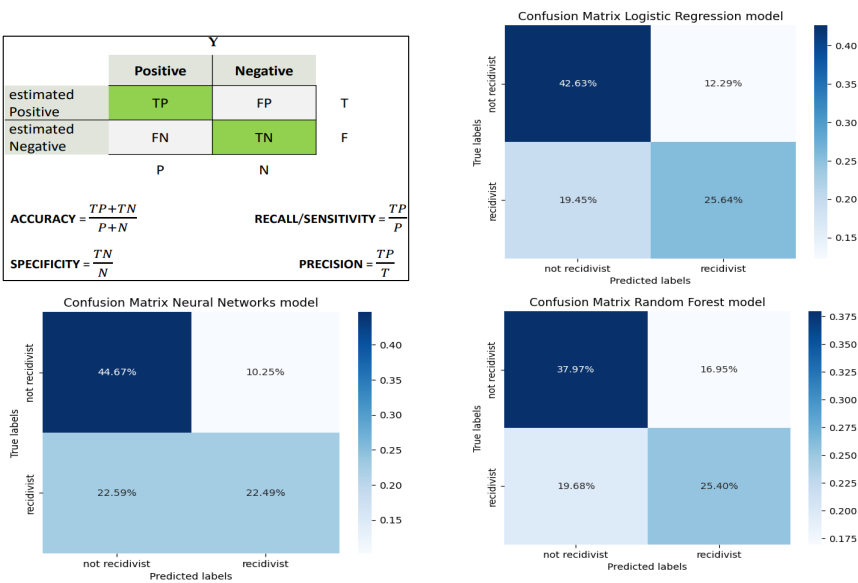


Figure 6.4 Confusion matrix: comparison between actual (rows) and predicted values (columns). Logistic Regression, Neural Network and Random Forest models; COMPAS dataset.

counterparts: accuracy increased from 0.54 to 0.62 (not shown).

6.3.4 Conclusions and Future Perspective

This study examines the predictive potential of ML models in assessing criminal recidivism risk, using the COMPAS dataset as a case study. Three supervised classification models – logistic regression, neural networks, and random forest – were tested and evaluated in terms of accuracy, interpretability, and subgroup variability. Although the overall predictive performance was moderate (maximum accuracy 0.68), logistic regression and neural networks outperformed random forest, particularly with respect to the area under the ROC curve (AUC) and F1-score.

This study shows that clustering reveals substantial accuracy heterogeneity in the COMPAS data: K-means++ identified stable subgroups with distinct performance levels, a structure confirmed by a Ward hierarchical solution and supported by slightly higher silhouette scores. SHAP values clarified why accuracy varies, with high-performing clusters showing consistent feature contributions and low-performing ones affected by heterogeneity and class imbalance. These results show that overall accuracy can hide substantial performance differences across subgroups. Cluster-based evaluation and interpretability methods therefore provide a more reliable and transparent assessment of model behaviour in criminal-risk prediction.

From a legal perspective, the study highlights the divergence between civil and common law systems: while tools like COMPAS are used in the U.S., In the Italian legal system, predictive assessments of individual risk are restricted by procedural norms (Article 220 of the Italian Code of Criminal Procedure), effectively limiting the use of AI tools to the post-sentencing phase. The EU AI Act (Regulation 2024/1689) classifies such tools as high-risk, subjecting them to strict requirements

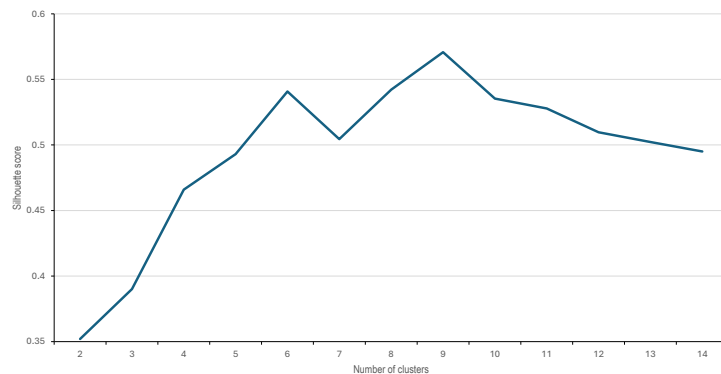


Figure 6.5 Silhouette score and number of clusters for COMPAS dataset.

of transparency, reliability, and protection of fundamental rights

Despite their current limitations, ML models could serve a complementary but important role in public policy and correctional planning. Their use should not be confined to making decisions about alternative sanctions or sentencing. Instead, they may be strategically employed to identify individuals at higher risk of reoffending, thereby enabling targeted interventions and the efficient allocation of social support services. We suggest adjusting the classification threshold; in this context, prioritizing a slightly higher false positive rate may be ethically acceptable if it serves to initiate rehabilitative or generally supportive interventions for the individual – rather than punitive measures.

In conclusion, while machine learning holds promise for enhancing our understanding of recidivism patterns and informing data-driven policies, its application in the criminal justice domain must remain grounded in legal safeguards, methodological transparency, and a strong commitment to human rights. Future research should further explore the integration of richer contextual variables, investigate algorithmic fairness across demographic groups.

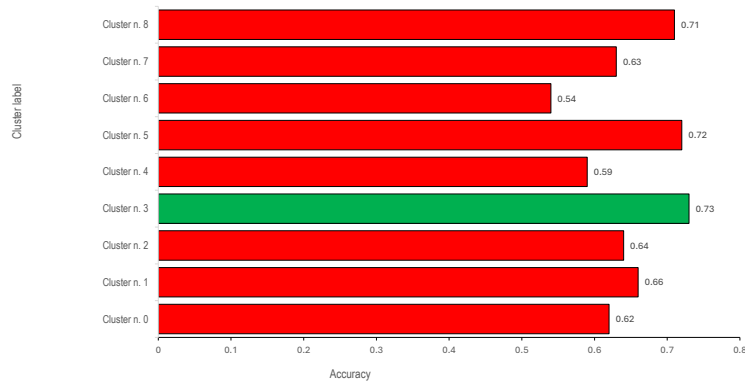


Figure 6.6 Accuracy of the Random Forest model calculated for each cluster identified using the K-Means method ($k = 9$) on the COMPAS dataset.

Random Forest (cluster n.3)						Random Forest (whole dataset)				
Random Forest	Precision	Recall	F1-score	Support	Balance	Precision	Recall	F1-score	Support	Balance
negative	0.76	0.91	0.83	86	0.71	0.66	0.69	0.67	1,189	0.55
positive	0.56	0.29	0.38	35	0.29	0.60	0.56	0.58	976	0.45
Accuracy			0.73	121				0.63	2,165	
Macro avg	0.66	0.60	0.60	121		0.63	0.63	0.63	2,165	
Weighted avg	0.70	0.73	0.70	121		0.63	0.63	0.63	2,165	

Figure 6.7 Classification report, main metrics for each class, cluster n.3 (left side) and whole dataset COMPAS (right side) based on the Random Forest model.

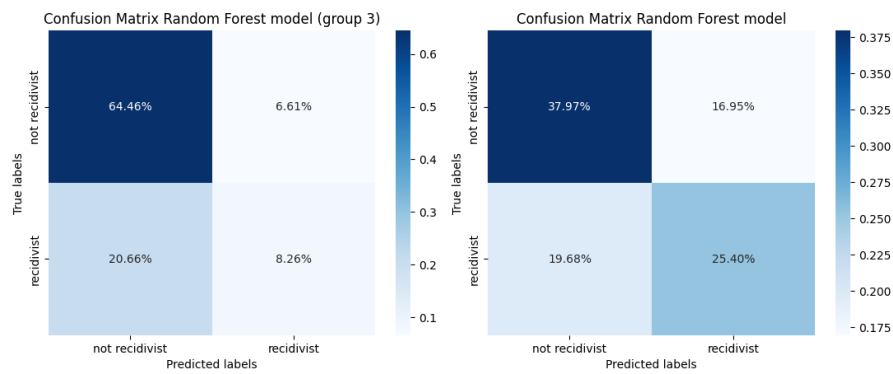


Figure 6.8 Random Forest model's confusion matrix. Comparison between actual (rows) and predicted values (columns). Cluster n.3 (left) and the COMPAS dataset (right).

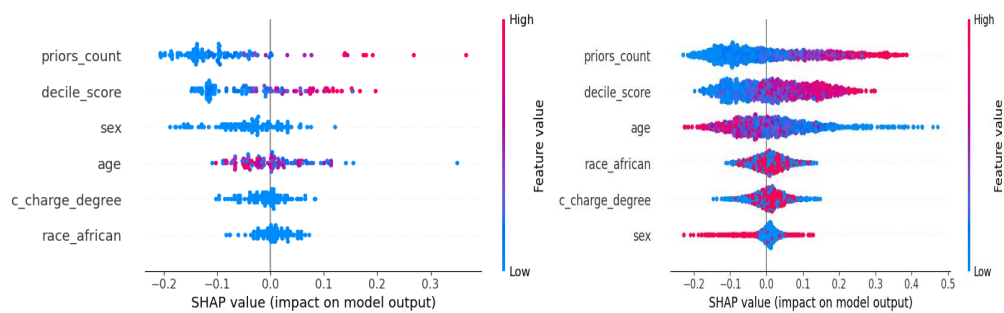


Figure 6.9 Random Forest model's SHAP values for cluster n.3 (left) and COMPAS dataset (right).

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis presents Scope, a Python toolkit to enhance the evaluation of classification models by providing a more granular analysis of model performance. Our approach diverges from the traditional reliance on global accuracy and represents a significant improvement over manual inspection of accuracy variation, thanks to a mixed integer optimization algorithm for automatic identification of high accuracy subspaces within the test set. The primary contribution of our algorithm lies in the definition of the range query, which provides a compact and human-readable representation for partitioning data points. Specifically, points within these defined intervals exhibit high accuracy, whereas those outside signify low accuracy. The theoretical framework of the Mixed Integer Programming (MIP) enables us to establish rigorous bounds for features through these queries, which consist of conjunctions of range constraints that filter objects based on whether their attributes fall within the specified intervals. Users will be able to explore these subspaces by running the optimization algorithm and inspecting results. This allows for the identification of common characteristics among correctly and incorrectly classified data points, thereby narrowing the scope for subsequent in-depth classification analysis.

A comparison between our algorithm and related methods within the field of Automatic Subgroup Discovery (ASD) reveals fundamental operational distinctions, despite their common goal of achieving model-agnostic metric decomposition. Unlike other approaches, such as DivExplorer, Scope is a non perturbative method specifically designed to identify the largest possible subsets of data points that achieve an accuracy value exceeding a predefined threshold, while other methods focus on the discovery of problematic subsets. Scope differs in both its formulation and output. While DivExplorer identifies frequent itemsets and ranks them by divergence, Scope directly optimizes for coverage under an accuracy constraint. It identifies an optimal subspace by solving a MIP problem, allowing it to operate on continuous feature spaces without the need for discretization. This approach preserves the original information of the data, at the cost of increased computational complexity inherent to mixed-integer optimization. Experimental results demonstrate that Scope effectively identifies extensive, high-accuracy subspaces across various datasets and models, consistently outperforming existing subgroup discovery methods in terms of

coverage, except when dealing with higher thresholds.

Scope enables users to obtain an optimal subspace for datasets containing numerical and binary features by solving the MIP problem with Gurobi. This method preserves information on interval-scale feature without discretization, thereby overcoming the inherent limitations of competing ASD approaches that rely on data binning. For instance, DivExplorer requires features to be on a nominal scale to generate itemsets; this necessitates the discretization of numerical attributes, which may introduce binning artifacts and potentially bias the final results. Notably, Scope requires only feature normalization during the preprocessing phase. For categorical variables, a binary transformation via one-hot encoding is applied to integrate nominal data into the mixed integer optimization framework.

Scope leverages SHAP values to provide feature-level explanations within the identified subspaces. This additional interpretability layer increases the tool’s practical utility, offering deeper insights into the model’s decision-making process. While DivExplorer’s framework is also rooted in Shapley coalitional game theory, the applications diverge significantly: Scope utilizes it to identify factors influencing specific predictions, whereas DivExplorer employs it to estimate the contribution of each attribute value to subgroup divergence. In the latter, the Global item contribution to divergence quantifies the marginal increase in divergence when an item is introduced, thereby capturing its overall impact on the model.

To enhance scalability, we introduce grid-based variants that partition the feature space to mitigate computational costs. This approach utilizes binning to divide the dataset into grids, creating clusters of homogeneous instances. By executing the optimization task within these localized clusters, the problem becomes more tractable, enabling the identification of solutions in large-scale scenarios where standard methods might converge poorly or fail. Two strategies are proposed: Scope Grid Sum (Scope GS) and Scope Grid Max (Scope GM). Scope GS computes solutions for the optimization problem within each partition; the union of these local solutions forms a complex subspace composed of disjoint regions across the feature space. Alternatively, Scope Grid Max (Scope GM) identifies the single optimal subspace by selecting the cell that yields the maximum number of points as determined by the Scope algorithm. These variants offer a practical trade-off between computational efficiency and solution quality, especially under high-accuracy thresholds.

Following the experimental evaluations, several conclusions can be drawn about our method. The novel algorithm is particularly effective at identifying subsets that respond best to specific actions, ensuring a predefined accuracy threshold is maintained for prediction tasks. Furthermore Scope can be defined as a detection method for problematic high accuracy subspaces, capable of identifying partitions that exceed a specific accuracy threshold. However, further analysis of these partitions is essential to understand the underlying drivers of such high performance. For these reasons, the proposed methodology could enhance the analysis of criminal recidivism data and contribute to the exploration of administrative records. This approach facilitates the discovery of patterns, provides actionable insights, and supports public decision-making, while remaining mindful of the limitations detailed in this dissertation.

7.2 Limitations and Future Work

Despite its effectiveness, Scope presents certain limitations that suggest directions for further investigation. First, the method relies on feature normalization; consequently, it sometimes fails to identify the global optimum when applied to raw, unscaled data. Restricting the search space to a $[0, 1]$ range may not only obscure the physical meaning of the bounds but also make the transformation sensitive to outliers, potentially failing to capture true data variability. Future research could refine the penalty term within the objective function (figure 4.1) to facilitate the exploration of optimal bounds using original data scales, specifically introducing a regularization component that penalizes excessively broad ranges ensuring comparability.

Second, the computational complexity of the Subspace Optimization Problem (SOP) remains a key limitation, particularly for large datasets. We optimize the method through a grid search within cells obtained via a binning operation. By effectively reducing complexity, this mechanism allows us to find solutions in scenarios where the ordinary problem would otherwise time out. Despite this efficiency, the results remain sensitive to the binning strategy. Replacing binning with a cluster analysis approach, such as K-means, would allow for a more nuanced grouping of units based on their similarities, potentially refining the analysis of their effects on classification. The SOP is formulated as a mixed-integer programming problem aimed at identifying the subspace that maximizes point coverage while satisfying a minimum accuracy constraint. Consequently, efficient methods for computing approximate solutions over large datasets are essential for maintaining computational tractability. Since execution time heavily depends on the cardinality of the dataset, such approximations are crucial for large-scale applications.

Furthermore, the clustering efficiency resulting from the binning procedure can be unsatisfactory; when measured as the entropy of the clustering scheme relative to the correct predictions distribution (Appendix .1), the values obtained for the considered datasets are excessively high. However, the clusters' shapes closely resemble hyper-rectangular subspaces, making the method highly intuitive, straightforward to implement, and closely aligned with the logic of the optimization problem.

The current implementation focuses on binary classification tasks for clarity. Yet, Scope can also solve multiclass tasks, as the problem formulation employs the notion of c_i , that indicates if the label returned by the classification model is correct or not, whatever the class could be. Similarly, while accuracy is used as the primary constraint, other metrics such as recall, precision or F1 score could be incorporated. The first constraint in figure 4.1 refers to accuracy over the threshold. To evaluate other metrics, we have to replace c_i with c_i^j , which indicates if the label returned by the classification model is correct or not for the class j . In this way we can compute quantities such as false and true positives, false and true negatives as well, enabling the optimization to target different evaluation criteria. In some cases metrics other than accuracy are more relevant. For instance, in the Compas dataset, false positives represent individuals predicted to be at high risk of recidivism despite having a low actual risk. Specifically, black defendants are frequently assigned higher risk scores than their actual behavior warrants, making the control of the false positive rate particularly critical.

Another significant area for future work involves benchmarking our algorithm against other Automatic Subgroup Detection techniques (ASD). DivExplorer is the most closely related method in this class; however, since it primarily targets the identification of critical data subgroups, a direct comparison remains challenging. For instance, this method identifies a subgroup that exhibits the highest divergence in the false positive rate compared to the baseline, subject to a minimum support threshold, ranking the resulting patterns. Scope identifies high accuracy subspaces, where accuracy serves as the primary constraint while the algorithm maximizes support.

Future research could explore the integration of interpretability tools, such as SHAP, to elucidate how features are prioritized when defining subspaces. Studying the interplay between feature importance and range boundaries could provide further insight into why specific solutions are prioritized during the optimization process.

Finally, applying this methodology to additional real world datasets, particularly within the judicial domain, could facilitate the development of specialized tools tailored to the exploration and interpretation of such complex data.

Bibliography

- [1] Flavia Tagliafierro and Claudio Caterino. Criminal recidivism. towards reliable and transparent predictive models. *Italian Journal of Economic, Demographic and Statistical Studies*, Vol. LXXX-2 April-June 2026, 2026.
- [2] Neeraj Venkataraman. *Quantitative Criminology Handbook*. Educohack Press, 2025.
- [3] Eliana Pastor and Francesco Bonchi. Intersectional fair ranking via subgroup divergence. *Data Min. Knowl. Discov.*, 38(4):2186–2222, May 2024.
- [4] Shuai An and Yang Cao. Relative keys: Putting feature explanation into context. *Proc. ACM Manag. Data*, 2(1), March 2024.
- [5] Donatella Firmani, Giorgio Grani, Flavia Tagliafierro, et al. Breaking down accuracy with subspace optimization. In *EDBT*, pages 790–793, 2024.
- [6] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.
- [7] Eliana Pastor, Elena Baralis, and Luca de Alfaro. A hierarchical approach to anomalous subgroup discovery. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2647–2659, 2023.
- [8] Caroline Wang, Bin Han, Bhrij Patel, and Cynthia Rudin. In pursuit of interpretable, fair and accurate machine learning for criminal recidivism prediction. *Journal of Quantitative Criminology*, 39(2):519–581, 2023.
- [9] Nima Shahbazi, Yin Lin, Abolfazl Asudeh, and H. V. Jagadish. Representation bias in data: A survey on identification and resolution techniques. *ACM Computing Surveys*, 55:1 – 39, 2022.
- [10] Suraj Shetiya, Ian P Swift, Abolfazl Asudeh, and Gautam Das. Fairness-aware range queries for selecting unbiased data. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 1423–1436. IEEE, 2022.
- [11] Eliana Pastor, Luca de Alfaro, and Elena Baralis. Looking for trouble: Analyzing classifier behavior via pattern divergence. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD ’21, page 1400–1412, New York, NY, USA, 2021. Association for Computing Machinery.
- [12] Cynthia Rudin, Caroline Wang, and Beau Coker. The Age of Secrecy and Unfairness in Recidivism Prediction. *Harvard Data Science Review*, 2(1), mar 31 2020. <https://hdsr.mitpress.mit.edu/pub/7z10o269>.

- [13] Marianna De Santis, Giorgio Grani, and Laura Palagi. Branching with hyperplanes in the criterion space: The frontier partitioner algorithm for biobjective integer programming. *European Journal of Operational Research*, 283(1):57–69, 2020.
- [14] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. Errudite: Scalable, reproducible, and testable error analysis. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 747–763, Florence, Italy, July 2019. Association for Computational Linguistics.
- [15] Abolfazl Asudeh, Zhongjun Jin, and H. V. Jagadish. Assessing and remedying coverage for a given dataset. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 554–565, 2019.
- [16] Ángel Alexander Cabrera, Will Epperson, Fred Hohman, Minsuk Kahng, Jamie Morgenstern, and Duen Horng Chau. Fairvis: Visual analytics for discovering intersectional bias in machine learning. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 46–56, 2019.
- [17] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. Automated data slicing for model validation: a big data - ai integration approach, 2019.
- [18] Vincenzo Di Cicco, Donatella Firmani, Nick Koudas, Paolo Merialdo, and Divesh Srivastava. Interpreting deep learning models for entity resolution: an experience report using lime. In *Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management @ SIGMOD*, 2019.
- [19] Eliana Pastor and Elena Baralis. Explaining black box models by means of local rules. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC '19*, page 510–517, New York, NY, USA, 2019. Association for Computing Machinery.
- [20] Arvind Narayanan. Translation tutorial: 21 fairness definitions and their politics. In *Proc. conf. fairness accountability transp., new york, usa*, volume 1170, page 3, 2018.
- [21] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [22] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.
- [23] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias, May 2016.

-
- [24] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International conference on knowledge discovery and data mining*, pages 1135–1144, 2016.
 - [25] Dimitris Bertsimas and Robert Weismantel. *Optimization over integers*. Dynamic Ideas, 2005.

Acknowledgments

I wish to convey my sincere thanks to Professor Donatella Firmani for her unwavering support and the precious guidance shared during the preparation of this thesis. Deep gratitude is extended to G. Grani for her exceptional contribution to the definition of the Subspace Optimization Problem. Her original insights were instrumental to this analysis. I would also like to express my deep gratitude to Eliana Pastor for her invaluable inspiration and insightful feedback throughout the development of this thesis.

Appendix

.1 Entropy of the Clustering Scheme

To support our approach, we define the clustering entropy as the weighted average of the entropy within each cluster, based on its relative size. Entropy measures impurity and is used to quantify the state of disorder within data, evaluating how mixed a distribution is. We are interested in measuring the impurity of the correct predictions distribution across the dataset and within the groups created by the binning procedure to verify if this process effectively reduces impurity. For example, we generate predictions using a Random Forest classifier and compare the global entropy against the weighted average entropy of the binned groups, which provides a comprehensive overview of the overall clustering scheme. Across all datasets, we observe a decrease in the original entropy value up to 5.33%, indicating that the binning procedure effectively reduces global impurity.

Global entropy is consistently greater than or equal to the clustering (weighted) entropy. While global entropy measures the total uncertainty of the dataset, the latter represents the average residual uncertainty once a data point’s cell within the grid partition is known. Conditioning on the cell or cluster can only decrease or, at most, leave unchanged the uncertainty regarding the class, but never increase it. This principle, derived from Jensen’s inequality for a concave function, is defined by the following equation .1, where H denotes the entropy, P_i is the probability of correct predictions, p is the total number of partitions, and w_i represents the weight assigned to each partition:

$$H\left(\sum_{i=1}^p w_i P_i\right) \geq \sum_{i=1}^p w_i H(P_i) \quad (.1)$$

.2 Main Tables

model	dataset	n test	accuracy	beta	Scope	Scope GS	Scope GM	DivExplorer
Random forest	titanic	267	0.82	0.85	75.28	82.40	38.95	47.94
Random forest	titanic	267	0.82	0.9	42.70	67.42	35.58	29.59
Random forest	titanic	267	0.82	0.95	24.72	47.57	27.34	10.86
Random forest	compas	2165	0.69	0.85	6.61	30.99	16.58	5.13
Random forest	compas	2165	0.69	0.9	2.91	14.92	6.61	3.37
Random forest	compas	2165	0.69	0.95	1.39	7.21	1.94	1.76
Random forest	credit	300	0.68	0.85	29.67	53.33	11.67	11.00
Random forest	credit	300	0.68	0.9	27.00	44.33	8.67	-
Random forest	credit	300	0.68	0.95	17.67	37.33	7.67	-
Logistic regression	titanic	267	0.8	0.85	65.17	80.90	37.45	40.07
Logistic regression	titanic	267	0.8	0.9	37.83	61.42	32.96	32.21
Logistic regression	titanic	267	0.8	0.95	22.85	44.57	22.85	21.35
Logistic regression	compas	2165	0.68	0.85	7.53	30.53	16.58	7.67
Logistic regression	compas	2165	0.68	0.9	2.86	14.97	6.61	3.28
Logistic regression	compas	2165	0.68	0.95	1.39	7.11	1.94	1.76
Logistic regression	credit	300	0.71	0.85	45.00	66.67	14.00	13.33
Logistic regression	credit	300	0.71	0.9	30.33	53.00	10.67	10.67
Logistic regression	credit	300	0.71	0.95	17.67	43.33	8.67	-
Neural networks	titanic	267	0.81	0.85	75.28	83.90	37.45	57.68
Neural networks	titanic	267	0.81	0.9	41.20	69.29	32.96	32.21
Neural networks	titanic	267	0.81	0.95	22.85	53.56	22.85	21.35
Neural networks	compas	2165	0.67	0.85	7.53	30.12	16.58	8.64
Neural networks	compas	2165	0.67	0.9	3.42	15.47	6.61	3.28
Neural networks	compas	2165	0.67	0.95	1.52	7.39	1.94	1.76
Neural networks	credit	300	0.71	0.85	45.00	64.33	14.00	13.33
Neural networks	credit	300	0.71	0.9	30.33	53.00	10.67	10.67
Neural networks	credit	300	0.71	0.95	17.67	43.33	8.67	-

Table .1 A comparison of number of points (percentage) in two automatic subgroup detection techniques: Scope (standard, Grid Sum, Grid Max) and DivExplorer.

model	dataset	n test	accuracy	beta	Scope	Scope SER	Scope PAR	DivExplorer
Random forest	titanic	267	0.82	0.85	15.17	4.41	2.38	0.58
Random forest	titanic	267	0.82	0.9	168.33	24.49	18.37	0.58
Random forest	titanic	267	0.82	0.95	372.04	11.65	5.14	0.58
Random forest	compas	2165	0.69	0.85	8334.06	2314.07	601.82	12.46
Random forest	compas	2165	0.69	0.9	17458.87	2064.82	602.79	12.46
Random forest	compas	2165	0.69	0.95	14226.64	1270.04	358.11	12.46
Random forest	credit	300	0.68	0.85	600.91	78.79	48.26	0.47
Random forest	credit	300	0.68	0.9	474.13	27.05	10.84	0.47
Random forest	credit	300	0.68	0.95	554.71	26.78	14.49	0.47
Logistic regression	titanic	267	0.8	0.85	19.25	13.01	11.81	0.45
Logistic regression	titanic	267	0.8	0.9	88.76	14.99	7.80	0.45
Logistic regression	titanic	267	0.8	0.95	124.8	29.68	23.29	0.45
Logistic regression	compas	2165	0.68	0.85	7014.65	1931.01	600.91	5.36
Logistic regression	compas	2165	0.68	0.9	39452.26	1971.57	600.74	5.36
Logistic regression	compas	2165	0.68	0.95	24058.61	2082.40	601.02	5.36
Logistic regression	credit	300	0.71	0.85	359.64	14.94	6.18	0.23
Logistic regression	credit	300	0.71	0.9	569.48	13.17	3.06	0.23
Logistic regression	credit	300	0.71	0.95	600.98	13.28	3.19	0.23
Neural networks	titanic	267	0.81	0.85	29.77	45.77	15.04	14.49
Neural networks	titanic	267	0.81	0.9	144.59	50.55	16.44	14.49
Neural networks	titanic	267	0.81	0.95	157.65	57.41	21.14	14.49
Neural networks	compas	2165	0.67	0.85	9811.20	2142.69	628.23	45.07
Neural networks	compas	2165	0.67	0.9	8159.96	2332.22	622.31	45.07
Neural networks	compas	2165	0.67	0.95	4617.93	2384.78	621.74	45.07
Neural networks	credit	300	0.71	0.85	626.17	203.26	21.32	10.54
Neural networks	credit	300	0.71	0.9	625.84	185.12	18.57	10.54
Neural networks	credit	300	0.71	0.95	625.87	188.50	19.37	10.54

Table .2 A comparison of execution times (seconds) in two Automatic Subgroup Detection techniques: Scope (standard, GS SERial, GM PARallel) and DivExplorer.