Exceptional Model Mining on Model Residuals: Balancing Interpretability and Expressiveness in Rich Description Languages

Interpretable Insights into Black-Box Failures

Aniket Mishra
a.mishra3@student.tue.nl
Eindhoven University of Technology
Eindhoven, The Netherlands

Cristiana Cărbunaru
c.carbunaru@student.tue.nl
Eindhoven University of Technology
Eindhoven, The Netherlands

Abstract

ToDo

Keywords

Exceptional Model Mining, Subgroup Discovery, Residual Analysis, Interpretability

ACM Reference Format:
Aniket Mishra and Cristiana Cărbunaru. 2026. Exceptional Model Mining on Model Residuals: Balancing Interpretability and Expressiveness in Rich Description Languages: Interpretable Insights into Black-Box Failures. In *KDD '26: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, August 09–13, 2026, Jeju, Korea.* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Overview

This paper explores **Exceptional Model Mining (EMM)** and **Subgroup Discovery (SD)** [8, 14] for analyzing black-box regression models through their residuals [2, 16, 24, 25]. The goal is to discover interpretable subgroups where the model exhibits systematic over- or under-prediction, revealing regions of poor generalization or bias. We further investigate how different description languages, such as conjunctive, polynomial, tree-based, and symbolic, affect the balance between subgroup expressiveness and interpretability [23].

2 Introduction

Recent advances in deep learning and ensemble models have produced highly accurate predictive systems, but their opacity makes diagnosing model errors difficult. EMM and SD offer a framework to localize regions in the data where a model behaves unexpectedly. This paper explores residual-based EMM, where the residuals of a black-box model serve as the target variable for discovering interpretable subgroups that explain systematic model underperformance or bias. We further study how the choice of description language influences the trade-off between interpretability and expressiveness. By comparing conjunctive, polynomial, tree-based, and symbolic descriptions across energy and wind-turbine case studies, we aim to identify which representations best reveal meaningful, human-understandable failure patterns. **ToDo More Later**

3 Theoretical Background

3.1 Model Residuals + Rich Descriptions

Deep learning and other black-box models [24, 25] are widely used in regression tasks such as housing price estimation and music-year prediction. By analyzing residuals as the target [15], we can identify interpretable subgroups where the model consistently over- or under-predicts, revealing systematic biases or feature combinations that cause poor predictions/local non-linear patterns.

Classical SD typically employs simple conjunctions of attribute–value conditions [14] (e.g., $f_1 > a \land f_2 = b$). Mampaey et al. (2012) [20] introduced the notion of *richer descriptions*, where the focus lies on constructing more complex *atomic predicates* (e.g., numeric intervals, nominal disjunctions) within a conjunction, while maintaining the same overall logical structure.

In contrast, our work explores the use of **richer description languages**, such as shallow decision trees, low-degree polynomial predicates, and compact symbolic expressions [2, 19].

These languages expand beyond pure conjunctions, enabling the discovery of nonlinear or hierarchical failure patterns while preserving human interpretability and aiding in model fairness analysis [16].

The subsequent part, Section 4, formalizes this intuition. We cast the task as EMM on residuals, specify the optimization problem, and derive research questions about the trade-off between model-agnostic subgroup quality and interpretability across different description languages.

ToDo More Later

3.2 Related Work

Research on EMM originates from Leman et al. (2008) [18], who formalized the general framework of comparing local model parameters within subgroups to a global reference model. Subsequent works explored black-box model inspection through randomization and visualization [12, 13], and model class–specific EMM variants such as SCaPE [9] and Conformalized EMM [7]. Our work extends this line of research to residuals of regression and forecasting models, a setting that has received limited attention so far.

KDD '26, Jeju, Korea

ACM ISBN 978-1-4503-XXXX-X/2026/06
https://doi.org/XXXXXXX.XXXXXXX

Regarding description languages, most SD and EMM approaches rely on conjunctions of simple predicates [14]. Mampaey et al. (2012) [20] enhanced these predicates to handle numeric and nominal data more expressively, but the logical structure remained conjunctive. We generalize this idea by comparing multiple description languages-conjunctive, polynomial, tree-based, and symbolic to study their impact on both subgroup quality and interpretability. For handling numeric attributes, we adopt insights from Meeng and Knobbe [21], who proposed efficient local discretization strategies.

[**FOR RQ1**] Several strands of EMM research have addressed classifier exploration and model-based targets. Leman et al. (2008) [18] first formalized EMM, while later works such as [7, 9, 12, 13] focused on explaining classification models through model-induced targets or randomization strategies. However, residual-based EMM for regression and forecasting models remains largely unexplored. Our approach bridges this gap by explicitly modeling residuals as the exceptional target, enabling systematic analysis of model performance regions in continuous domains.

4 Problem Statement

Insights: Aniket’s experience. Aniket explains: “In one of my earlier projects on a solar power plant, all our models suddenly started flagging every single device for having a "critical health score", meaning major failure within the next 3 to 6 months. After investigating it for a while, we realised that it was happening because the average ambient temperature increased from 50 to 60 + degrees Celsius, triggering our bad data filter, causing the model to misbehave.

If we had a subgroup-based, drift-sensitive monitoring system, it could have flagged something like: “Subgroup of devices with ambient temperature $>$ 60°C shows exceptional deviation in predicted healthscore.” “

This idea is connected to the solar plant case described earlier. If we had analyzed model residuals directly, we might have quickly spotted that the high errors were confined to subgroups with unusually high ambient temperatures. That would have saved significant debugging time and clarified that the problem was with model generalization, not the solar panels.

Another example would be a downtime detection system Aniket built in a past project. By what he explained to the group, initially, it was a hard-coded chain of `elif` ladders running over aggregated time-series data. Later, Aniket made it configurable so that asset owners could define their own conditions like “Gen Speed $>$ 300,” “Wind Speed $\geq$ 3", and "Wind Speed $\leq$ 25,” and “Pitch Angle $\geq$ 75”. It worked (to an extent?), but writing and maintaining these rules was painful and heavily dependent on domain experts. If subgroup discovery could automatically find/learn such complex relationships from data, especially using richer description languages, it could help uncover operational rules directly, reducing manual effort and improving coverage. It could also aid in prescriptive analysis if we imagine it correctly.

While these industrial examples originally motivated this research, we rely on two benchmark regression datasets for empirical evaluation (see Section 6.1.1), which allow controlled experiments on model residuals under varying data dimensionality.

4.1 Research Questions and Hypotheses

Therefore, inspired by the aforementioned real-life issues, the following research questions are proposed, with RQ1 representing the main question:

RQ1: Can we identify interpretable subgroups where black-box model residuals exhibit exceptional patterns, revealing systematic model failures or biases [15, 26]?

RQ2: Which description languages balance interpretability and expressiveness for capturing non-linear model failure patterns? Which description languages $\mathcal{L}$ (conjunctions, polynomial predicates, shallow decision trees, and symbolic expressions) optimize

$$\max_{d\in\mathcal{L}} q(d) = \text{effect size}(r|d)$$

subject to interpretability constraints $|\text{desc}(d)| \leq L$ [1, 3, 5, 6, 10, 22, 23]?

RQ3: How does the expressiveness of the description language $\mathcal{L}$ influence the balance between subgroup quality and interpretability across datasets of varying complexity/domains, and which configurations are optimal [3, 17, 26]?

For **RQ2**, it is important to keep in mind that these description languages form a partial hierarchy of expressiveness (i.e., $\mathcal{L}_{\text{conj}} \subset \mathcal{L}_{\text{tree}}$ when both are bounded by the same depth). To avoid trivial dominance, we constrain each language by comparable complexity limits (e.g., at most five predicates for conjunctions, maximum depth of three for trees, and degree two for polynomials), so that improvements in subgroup quality $q(d)$ cannot be attributed solely to unconstrained expressive power.

Based on the research questions, hypotheses were composed as follows:

H1: Richer languages ($\mathcal{L}_{\text{poly}}, \mathcal{L}_{\text{tree}}, \mathcal{L}_{\text{symb}}$) will tend to increase $q(d)$ but decrease $I(d)$.

H2: Shallow trees or low-degree polynomials may achieve the best balance for practical domains (e.g., energy forecasting and fault analysis).

We will map the Pareto front $\{(q(d), I(d))\}$ per $\mathcal{L}$ to identify domain-appropriate operating points.

5 Solution Approach

ToDo

6 Experimental Results

6.1 Methodology

This section describes the methodology to solve our research questions. We introduce the datasets, formalize the problem, define the description languages for subgroup discovery, introduce the quality and interpretability measures, formulate the trade-off objective, and describe the optimization strategy.

6.1.1 Used Datasets and Their Overview. For now, we conduct experiments on two standard regression datasets: Boston Housing [11] and Year Prediction MSD [4], which differ in dimensionality and data complexity:

- **Boston Housing** (506 samples, 14 features): predicts median home value in \$1000s from socioeconomic and housing attributes.
  Features example of Boston dataset: RM - average number of rooms per dwelling; LSTAT - % lower status of the population; MEDV - Median value of owner-occupied homes in \$1000's; TAX - full-value property-tax rate per \$10,000.
- **Year Prediction MSD** (515,345 samples, 90 features): predicts the release year of a song from audio timbre statistics.

For both datasets, we train a black-box regressor (Gradient Boosted Trees and a simple neural network, respectively), compute residuals, and perform Exceptional Model Mining on the residuals to identify interpretable subgroups of systematic over- or under-prediction.

6.1.2 Code. All datasets and code are available in a **GitHub repository**.

6.1.3 Descriptions of Research Questions.

RQ1: *Residual-based Exceptionalness:*

Given a trained black-box model $f : \mathcal{X} \rightarrow \mathbb{R}$ and a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, with $n = |D|$ (i.e., cardinality), and $S \subseteq D$, which we treat as a *multiset* to allow duplicate or repeated observations, let residuals be defined as:

$$r_i = |y_i - f(x_i)| \quad \text{or} \quad r_i = (y_i - f(x_i))^2$$

or some other form of loss calculation.

We assume a user-specified description language $\mathcal{L}$ that determines the form of allowable subgroup descriptions $d \in \mathcal{L}$; subsequent sections investigate several instantiations of $\mathcal{L}$.

We aim to find subgroups [8, 14] $S = \{x \in \mathcal{X} : d(x) = 1\}$, where the subgroup description d maximizes the residual exceptionalness score:

$$q_{\text{residual}}(S) = \frac{|\mathbb{E}_{x \in S}[r(x)] - \mathbb{E}_{x \in D}[r(x)]|}{\sqrt{\text{Var}_{x \in D}[r(x)]/|S|}}$$

This measure quantifies how **exceptional** the average residual in subgroup S is compared to the overall dataset D:

- $\mathbb{E}_{x \in S}[r(x)]$: average residual within subgroup S;
- $\mathbb{E}_{x \in D}[r(x)]$: average residual across the entire dataset;
- The numerator $|\mathbb{E}_{x \in S}[r(x)] - \mathbb{E}_{x \in D}[r(x)]|$ measures the magnitude of deviation between the subgroup and the global average;
- The denominator $\sqrt{\text{Var}_{x \in D}[r(x)]/|S|}$ acts as a normalization factor, adjusting for global variance and subgroup size, similar to a *standard error*;
- The $q_{\text{residual}}(S)$ behaves like a z-score, identifying subgroups where residuals deviate substantially beyond what is expected by chance.

A high $q_{\text{residual}}(S)$ would show a subgroup where the model systematically underperforms (large residuals) or overperforms (small residuals), showing bias (underfitting) or overfitting. In our experiments, we will explore both directions separately: one list of subgroups where the model systematically *underperforms* (large positive residuals), and another where it *overperforms* (small or negative residuals).

RQ2: *Expressive Descriptions:*

We consider several hypothesis language classes for interpretable subgroup discovery:

(1) **Conjunctive (baseline):**

$$\mathcal{L}_{\text{conj}} = \{f_1\, \theta_1\, v_1 \;\wedge\; \cdots \;\wedge\; f_k\, \theta_k\, v_k\}$$

This language uses simple conjunctions of attribute–value conditions. It is the most interpretable form, expressing subgroups as a set of jointly satisfied predicates.
Example (Boston Housing):

$$(\text{RM} > 7) \wedge (\text{LSTAT} < 5)$$

identifies a subgroup of expensive houses with many rooms and low lower-status population.

(2) **Polynomial:**

$$\mathcal{L}_{\text{poly}} = \left\{ a_0 + \sum_i a_i f_i + \sum_{i,j} a_{ij} f_i f_j \;\leq\; \tau \right\}$$

To maintain tractability and interpretability, we restrict polynomial predicates to degree at most two. Coefficients are estimated via least squares over candidate feature pairs, and thresholds are quantized to a fixed grid. This limits the hypothesis space to $O(p^2)$ candidate monomials for p features, avoiding combinatorial explosion.

The language captures nonlinear relationships through polynomial combinations of features [5]. It allows expressing curved decision boundaries or interaction effects.
Example (based on the Boston Housing dataset):

$$0.6 \cdot \text{RM}^2 - 0.4 \cdot \text{LSTAT} < 30$$

identifies non-linear relation conditions between room count and status ratio that predicts higher prices

(3) **Decision Tree:**

$$\mathcal{L}_{\text{tree}} = \{\text{shallow trees with depth } \leq 3\}$$

Subgroups are represented as paths in a decision tree of limited depth. Each path corresponds to a conjunction of conditions learned from data, offering a balance between interpretability and expressiveness [10].
Example path (based on the Boston Housing dataset):

$$\text{LSTAT} < 10 \longrightarrow \text{RM} > 6 \longrightarrow \text{Residual} > 2$$

defines a subgroup where the model tends to under-predict for affluent neighborhoods with large houses.

To handle numeric attributes efficiently, we follow ~~recommendations by Meeng and Knobbe (2021)~~ [21], ~~who systematically evaluated algorithms for dealing with numeric features in SD~~. We adopt a local discretization approach, determining cut points only where statistically justified within each branch, to avoid the explosion of candidate intervals.

(4) **Symbolic Expressions:**

$$\mathcal{L}_{\text{symb}} = \{\text{combinations of <= and logical operations}\}$$

This language allows flexible symbolic expressions (e.g., ratios, differences, aggregates) generated through symbolic regression [1, 6], providing interpretable yet expressive formulas.

Example (based on Boston Housing):

$$\frac{\text{RM}}{\text{LSTAT}} > 1.5$$

indicates houses where the ratio of room count to lower-status percentage is high, showing systematic over-prediction.

The goal is to optimize the following objective:

$$d^* = \arg\max_{d \in \mathcal{L}} \lambda \cdot q(d) + (1 - \lambda) \cdot I(d)$$

where:

- $q(d)$: subgroup quality (effect size on residuals) [22]
- $I(d)$: interpretability score (inverse of description complexity) [3]
- λ: trade-off parameter

Constraints: The symbolic language $\mathcal{L}_{\text{symb}}$ allows combinations of maximum of 3 (we choose 3, jlt) arithmetic and logical operations built from a restricted operator set $\{+, -, \times, \div, \leq, >, =\}$ and up to two distinct features per description/expression. We permit simple ratio or difference formulas such as $\frac{f_i}{f_j} \leq \tau$ or $(f_i - f_j) > \tau$, which cover many interpretable relationships (e.g., efficiency ratios or deviations). Thresholds τ are quantized to a predefined resolution. This keeps the search space finite and the resulting expressions human-readable.

RQ3: *Interpretability–Quality Trade-off:*

Using RQ2's languages ($\mathcal{L}_{\text{conj}}, \mathcal{L}_{\text{poly}}, \mathcal{L}_{\text{tree}}, \mathcal{L}_{\text{symb}}$), we formalize a trade-off between:

- $q(d)$: **quality** of a description d (e.g., residual exceptionalness / effect size on residuals);
- $I(d)$: **interpretability** of d, modeled as a decreasing function of its complexity.

Interpretability model. Let complexity(d) aggregate simple, auditable counts:

$$\begin{aligned} \text{complexity}(d) = {} & w_1 \cdot \#\text{predicates}(d) \\ & + w_2 \cdot \#\text{operators}(d) \\ & + w_3 \cdot \text{depth}(d) \\ & + w_4 \cdot \text{precision}(d) \end{aligned}$$

where precision(d) penalizes overly precise thresholds (e.g., many decimal places). We instantiate $I(d)$ as

$$I(d) = \frac{1}{1 + \text{complexity}(d)} \quad \text{or} \quad I(d) = \exp\bigl(-\beta \cdot \text{complexity}(d)\bigr),$$

with $\beta > 0$ controlling how sharply interpretability decays with complexity [17].

Complexity components: We define each component so that interpretability scoring is auditable and intuitive:

- **#predicates(d): number of atomic conditions:** Counts the number of simple boolean tests (e.g., $f > \tau$, $f = v$). Example: (RM > 6.5) ∧ (LSTAT < 10) ⇒ #predicates = 2, representing homes with many rooms in affluent areas.
- **#operators(d)**: arithmetic/logical operators used: Includes $\{+, -, \times, \div, \wedge, \vee, \leq, >, =\}$. Example: (RM > 7) ∧ (TAX < 300) ⇒ #operators = 3 (>, <, ∧).
- **depth(d)**: maximum path length from the root of a decision tree to any leaf, measuring the deepest reasoning chain required. Example: LSTAT $< 10 \longrightarrow$ RM $> 6 \longrightarrow$ Residual $> 2 \Rightarrow$ depth = 2, indicating a simple two-step rule that identifies underpredicted wealthy areas.
- **#splits(d)**: total number of internal decision nodes across the entire tree, reflecting global structural complexity. Example: a decision tree with three internal splits on `RM`, `LSTAT`, and `TAX` has #splits = 3.
- **precision(d): threshold resolution penalty:** Rather than normalizing all numeric attributes, we penalize thresholds that are overly precise relative to the natural measurement scale of each feature. Let each numeric attribute f_j have observed range $R_j = \max(f_j) - \min(f_j)$ and measurement resolution Δ_j (the smallest meaningful increment in the data). For every numeric condition in a description d, we define

$$\text{precision}(d) = \sum_{j \in \text{num}(d)} \log_{10}\left(\frac{R_j}{\Delta_j}\right).$$

  This term grows when thresholds use unrealistically fine granularity compared to how the variable is measured. For example, if RM ranges from 3 to 9 (so $R = 6$) and is recorded to the nearest 0.1, a threshold such as RM > 6.2 incurs a small penalty, while RM > 6.23 incurs a larger one because it uses unnecessarily fine precision.
- **#terms(d): number of monomials in a polynomial:** Example: $a_0 + a_1 f_1 + a_2 f_2 + a_{12} f_1 f_2 \Rightarrow$ #terms = 4. This measures how many interaction components are combined in the polynomial rule.
- **degree(d): polynomial degree:** Example: $0.6 \cdot \text{RM}^2 - 0.4 \cdot \text{LSTAT} \Rightarrow$ degree = 2, capturing non-linear housing relationships.
- **#tokens(d): symbolic token count:** Total of operands (variables, constants) and operators. This metric quantifies the symbolic compactness of an expression. Example: $\frac{\text{RM}}{\text{LSTAT}} > 1.5 \Rightarrow$ #tokens = 5 (RM, LSTAT, 1.5, /, >). [26]

We then compute:

$$\begin{aligned} \text{complexity}(d) = {} & w_1 \cdot \#\text{predicates}(d) \\ & + w_2 \cdot \#\text{operators}(d) \\ & + w_3 \cdot \text{depth}(d) \\ & + w_4 \cdot \text{precision}(d) \end{aligned}$$

and derive interpretability:

$$I(d) = \frac{1}{1 + \text{complexity}(d)}.$$

Language-specific operationalization.

$$\begin{aligned} \mathcal{L}_{\text{conj}} &: \ \text{complexity}(d) = \#\text{predicates}(d). \\ \mathcal{L}_{\text{tree}} &: \ \text{complexity}(d) = \text{depth}(d) + \#\text{splits}(d). \\ \mathcal{L}_{\text{poly}} &: \ \text{complexity}(d) = \#\text{terms}(d) + \text{degree}(d). \\ \mathcal{L}_{\text{symb}} &: \ \text{complexity}(d) = \#\text{tokens}(d) \ \text{(operands + operators)}. \end{aligned}$$

Trade-off objective. We search for descriptions that balance quality and interpretability:

$$d^* = \arg\max_{d \in \mathcal{L}} \ \lambda \cdot q(d) + (1 - \lambda) \cdot I(d), \quad \lambda \in [0, 1].$$

The trade-off parameter $\lambda \in [0, 1]$ is user-controlled: $\lambda \to 1$ emphasizes subgroup quality, while $\lambda \to 0$ prioritizes interpretability.

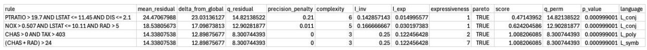
Constraints (readability guards).

- #predicates$(d) \leq 5$ (concision);
- depth$(d) \leq 3$ for trees; polynomial degree$(d) \leq 2$;
- thresholds quantized to sensible units (e.g., $1000 increments for MEDV, 0.1 rooms for RM), to ensure interpretability and prevent overfitting to measurement noise.

At each iteration, candidate subgroups are scored by $q(d)$ and $I(d)$, and Pareto-optimal solutions are retained to visualize the interpretability–quality frontier.

6.2 Preliminary Results

In Figure 1, the Pareto front results of languages conjunctive, polynomial, and symbolic expressions can be found.



rule	mean_residual	delta_from_global	q_residual	precision_penalty	complexity	I_inv	I_exp	expressiveness	pareto	score	q_perm	p_value	language
PTRATIO > 19.7 AND LSTAT <= 11.45 AND DIS <= 2.1	24.47067988	23.03136127	14.82138522	0.21	6	0.142857143	0.014995577	1	TRUE	0.47143952	14.82138522	0.000999001	L_conj
NOX > 0.507 AND LSTAT <= 10.11 AND RAD > 5	18.53805673	17.09873813	12.90281877	0.011	5	0.166666667	0.030197383	1	TRUE	0.624204586	12.90281877	0.000999001	L_conj
CHAS > 0 AND TAX > 403	14.33807538	12.89875677	8.300744393	0	3	0.25	0.122456428	2	TRUE	1.008206085	8.300744393	0.000999001	L_poly
(CHAS + RAD) > 24	14.33807538	12.89875677	8.300744393	0	3	0.25	0.122456428	7	TRUE	1.008206085	8.300744393	0.000999001	L_symb

Figure 1: The Pareto Results: Comparison of discovered rules with key residual and interpretability metrics.

7 Conclusions

ToDo

References

[1] Guilherme Seidyo Imai Aldeia and Fabricio Olivetti de Franca. 2024. Interpretability in Symbolic Regression: a benchmark of Explanatory Methods using the Feynman data set. arXiv:2404.05908 [cs.LG]

[2] Jakob Bach. 2024. Using Constraints to Discover Sparse and Alternative Subgroup Descriptions. *arXiv (Cornell University)* (2024). doi:10.48550/arxiv.2406.01411

[3] Pablo Barceló, Mikaël Monet, Jorge Pérez, and Bernardo Subercaseaux. 2020. Model Interpretability through the Lens of Computational Complexity. arXiv:2010.12265 [cs.AI]

[4] T. Bertin-Mahieux. 2011. Year Prediction MSD. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C50K61.

[5] Aleksey Buzmakov. 2020. Towards Polynomial Subgroup Discovery by means of FCA. In *CEUR Workshop Proceedings*, Vol. 2729. 1–12.

[6] F. O. de Franca, M. Virgolin, M. Kommenda, et al. 2023. Interpretable Symbolic Regression for Data Science: Analysis of the 2022 Competition. arXiv:2304.01117 [cs.LG]

[7] Xin Du, Sikun Yang, Wouter Duivesteijn, and Mykola Pechenizkiy. 2025. Conformalized Exceptional Model Mining: Telling Where Your Model Performs (Not) Well. arXiv:2508.15569 [cs.LG] doi:10.48550/arXiv.2508.15569

[8] Wouter Duivesteijn, Ad J. Feelders, and Arno Knobee. 2016. Exceptional Model Mining. *Data Mining and Knowledge Discovery* 30, 1 (2016), 47—-98. doi:10.1007/s10618-015-0403-4

[9] Wouter Duivesteijn and Julia Thaele. 2014. Understanding Where Your Classifier Does (Not) Work - The SCaPE Model Class for EMM. *Proceedings - IEEE International Conference on Data Mining, ICDM* (01 2014), 809–814. doi:10.1109/ICDM.2014.10

[10] Jack H. Good, Torin Kovach, Kyle Miller, and Artur Dubrawski. 2023. Feature Learning for Interpretable, Performant Decision Trees. In *Advances in Neural Information Processing Systems*, Vol. 36. 1–11.

[11] David Harrison and Daniel L. Rubinfeld. 1978. Hedonic Housing Prices and the Demand for Clean Air. Journal of Environmental Economics and Management, 81–102 pages. Original data.

[12] Andreas Henelius, Kai Puolamäki, Henrik Boström, Lars Asker, and Panagiotis Papapetrou. 2014. A peek into the black box: exploring classifiers by randomization. *Data Mining and Knowledge Discovery* 28, 5–6 (Sept. 2014), 1503–1529. doi:10.1007/s10618-014-0368-8

[13] Andreas Henelius, Kai Puolamäki, Isak Karlsson, Jing Zhao, Lars Asker, Henrik Boström, and Panagiotis Papapetrou. 2015. GoldenEye++: a Closer Look into the Black Box. In *Statistical Learning and Data Sciences. SLDS 2015. Lecture Notes in Computer Science*, Vol. 9047. 96–105. doi:10.1007/978-3-319-17091-6_5

[14] Francisco Herrera, Cristóbal José Carmona, Pedro González, and María José Del Jesus. 2010. An overview on subgroup discovery: foundations and applications. *Knowledge and Information Systems* 29, 3 (2010), 495—-525. doi:10.1007/s10115-010-0356-2

[15] Kimia Kamal and Bilal Farooq. 2022. Ordinal-ResLogit: Interpretable Deep Residual Neural Networks for Ordered Choices. arXiv:2204.09187 [cs.LG]

[16] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. An Empirical Study of Rich Subgroup Fairness for Machine Learning. arXiv:1808.08166 [cs.LG] https://arxiv.org/abs/1808.08166

[17] John P. Lalor and Hong Guo. 2022. Measuring algorithmic interpretability: A human-learning-based framework and the corresponding cognitive complexity score. arXiv:2205.10207 [cs.AI]

[18] Dennis Leman, Ad Feelders, and Arno Knobbe. 2008. Exceptional Model Mining. In *Data Mining and Knowledge Discovery*, Vol. 24. 1–16. doi:10.1007/978-3-540-87481-2_1

[19] Lu Liu. [n. d.]. Causal Learning for Heterogeneous Subgroups Based on Nonlinear Causal Kernel Clustering. https://arxiv.org/html/2501.11622v1#abstract. 2025.

[20] Michael Mampaey, Siegfried Nijssen, Ad Feelders, and Arno Knobbe. 2012. Efficient Algorithms for Finding Richer Subgroup Descriptions in Numeric and Nominal Data. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining (ICDM '12)*. IEEE Computer Society, USA, 499–508. doi:10.1109/ICDM.2012.117

[21] Marvin Meeng and Arno Knobbe. 2021. For real: a thorough look at numeric attributes in subgroup discovery. *Data Mining and Knowledge Discovery* 35 (01 2021), 1–55. doi:10.1007/s10618-020-00703-x

[22] Cristian Munoz, Kleyton da Costa, Bernardo Modenesi, and Adriano Koshiyama. 2023. Evaluating Explainability in Machine Learning Predictions through Explainer-Agnostic Metrics. arXiv:2302.12094 [cs.LG]

[23] W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. 2019. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences* 116, 44 (2019), 22071–22080. doi:10.1073/pnas.1900654116

[24] João Pimentel, Paulo J. Azevedo, and Luís Torgo. 2022. Subgroup mining for performance analysis of regression models. *Expert Systems* 40, 1 (2022). doi:10.1111/exsy.13118

[25] Youcef Remil, Anes Bendimerad, Marc Plantevit, Céline Robardet, and Mehdi Kaytoue. 2021. Interpretable Summaries of Black Box Incident Triaging with Subgroup Discovery. arXiv:2108.03013 [cs.AI] https://arxiv.org/abs/2108.03013

[26] Marco Virgolin, Andrea De Lorenzo, Eric Medvet, and Francesca Randone. 2020. Learning a Formula of Interpretability to Learn Interpretable Formulas. In *Parallel Problem Solving from Nature – PPSN XVI*. Springer, 79–93.

Received ?? October 2025; revised ????; accepted ????