This part looks alike this: identifying important parts of explained instances and prototypes

Jacek Karolczak[1,*], Jerzy Stefanowski[1]

[1]Poznan University of Technology, Institute of Computing Science, ul. Piotrowo 2, 60-695 Poznań, Poland

Abstract

Although prototype-based explanations provide a human-understandable way of representing model predictions they often fail to direct user attention to the most relevant features. We propose a novel approach to identify the most informative features within prototypes, termed alike parts. Using feature importance scores derived from an agnostic explanation method, it emphasizes the most relevant overlapping features between an instance and its nearest prototype. Furthermore, the feature importance score is incorporated into the objective function of the prototype selection algorithms to promote global prototypes diversity. Through experiments on six benchmark datasets, we demonstrate that the proposed approach improves user comprehension while maintaining or even increasing predictive accuracy.

Keywords

prototype-based explanation, feature importance, user attention guidance, local and global explanations

1. Introduction

Research on explaining black-box machine learning methods, which have been intensively developing in recent years, has led to the introduction of a great number of various explanation methods; see, e.g. [1]. Since prototypes correspond to training data, they are easier for humans to understand compared to more complex explanation methods [2]. Prototypes can serve as a *local explanation* by associating predictions with similar examples or as a *global explanation* to illustrate model decision boundaries using a limited number of representative instances.

Although in general prototypes can be applied to different types of data, in this paper we focus on tabular data, i.e., the description of examples in the form of vectors of (feature , value) pairs. However, their interpretation may be a challenge, especially when there are too many features [2]. For local explanations in particular, human users may encounter difficulties in assessing which features are most important for the prediction of the considered instance. Furthermore, it can be expected for global explanations that the discovered prototypes are not only well spread over the learning data space but are simultaneously characterized by quite diversified subsets of the most important features.

Recall that similar expectations have been examined for other data modalities. For images, *prototypical parts networks* were introduced to identify characteristic patches instead of complete images [3]. However, for tabular data, the decomposition into meaningful parts remains underexplored. To bridge this gap, we introduce the identification of the most important features in prototypes. This is achieved by applying an agnostic explanation method for computing the feature importance of the black-box model, and offers a more refined perspective than existing techniques. Such subsets of features can be exploited for local or global approaches and support users in better interpreting the provided explanations.

Our approach uses feature importance in two ways. First, we identify alike parts by highlighting the most informative overlapping features between an instance and its nearest prototype, directing the user's attention to a limited number of key features when interpreting a model prediction. Second, we incorporate feature importance into the prototype selection objective function to promote diversity,

Late-breaking work, Demos and Doctoral Consortium, colocated with The 3rd World Conference on eXplainable Artificial Intelligence: July 09–11, 2025, Istanbul, Turkey

*Corresponding author.

jacek.karolczak@cs.put.poznan.pl (J. Karolczak); jerzy.stefanowski@cs.put.poznan.pl (J. Stefanowski)
0000-0001-5414-960X (J. Karolczak); 0000-0002-4949-8271 (J. Stefanowski)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

which aids in identifying alike parts. These strategies balance interpretability and diversity, enhancing both local explanations and prototype selection. The methods are evaluated on benchmark datasets, with source code available on GitHub[1].

2. Related work

A dataset $\mathcal{S}$ consists of n instances (learning examples), expressed as $\mathcal{S} = (\mathbf{x}_i, y_i)_{i=1}^{n}$, where each $\mathbf{x}_i \in \mathbb{X}^d$ represents a d dimensional feature vector, and $y_i \in \mathcal{Y}$ denotes its corresponding label. In this work, we consider tabular data in a feature-value format. We assume the presence of a classifier h trained in $\mathcal{S}$, which serves as a black-box model to make predictions. The classifier h maps an input instance $\mathbf{x}_i$ to a predicted label $y_i : h(\mathbf{x}_i) \mapsto \hat{y}_i$.

From our perspective, a *prototype* is a representative instance selected from the dataset, i.e., an element $(\mathbf{x}_j, \hat{y}_j)$, where $\hat{y}_j$ denotes the class assignment of the instance made by the classifier h. Typically, the set of prototypes, denoted as $\mathcal{P}$, is a subset of $\mathcal{S}$, such that $\mathcal{P} = \{(\mathbf{x}_j, y_j)\}_{j=1}^{m}$, where $m \ll n$ ($\mathcal{P} \subset \mathcal{S}$), ensuring that the number of prototypes is much smaller than the total size of the training dataset.

Some prototype selection methods use kernel functions and vector quantization [4], while KNN-based methods share similar principles. IKNN_PSLFW [5], for example, partitions data into class-specific subsets and selects prototypes farthest from other classes. However, most methods rely on standard distance measures in the original attribute space, requiring a similarity definition that supports diverse data types (binary, numerical, categorical) and is robust to scaling differences. However, most of these algorithms exploit standard distance measures in the original attribute space, which requires the definition of similarity that supports different data types and is immune to different scales.

More recent proposals mitigate these distance limitations by considering the proximity of instances in the new space, referring to predictions of the black-box model; see the tree-space prototypes developed for explaining ensembles. The first algorithm, SM-A, introduced in [6], searches for prototypes – medoids in this space. However, it requires the user to specify the expected number of prototypes. This limitation was later addressed by A-PETE [7], which automates prototype selection.

Although numerous methods have been proposed to assess feature influence for black-box model predictions, they have not been widely applied in conjunction with prototype-based explanations. Popular techniques such as SHAP [8] as a local explanation yield a vector of length equal to the number of features, where each value attributes the importance score of individual features, helping to understand the behavior of the model for specific instances.

Despite multiple studies on prototypes for tabular data, only a few papers discuss how prototypes should be presented to end users. In [9], some prototype visualizations are provided, such as 2D scatter plots or self-organizing maps; however, they are suitable only for low-dimensional data and ultimately do not focus user attention on specific parts.

3. Method

In Section 3.1 we will first present our proposal to support the local explanation of the example predication by the nearest prototype. Then, in Section 3.2 we will generalize it to create a diverse global set of prototypes.

3.1. Identifying important parts

Following [2], for many features, a prototype as a whole can be difficult to comprehend and therefore make it difficult to explain the prediction of a black-box model. Some features within the prototype may be of high importance, while others may have low importance to the specific prediction that is being explained.

[1] https://github.com/jkarolczak/important-parts-of-prototypes

Table 1
Finding alike parts for the instance and its prototype from Apple Quality dataset. The first two rows present the feature importance values for the instance and its prototype, respectively. The third row shows the computed weights, obtained as the element-wise product of normalized feature importance scores (Formula 2). The bottom row indicates the binary mask, which selects the most relevant shared features-those with weights above the mean - denoted by '1'

	Size	Weight	Sweetness	Crunchiness	Juiciness	Ripeness	Acidity
Instance	-2.77	-1.08	-1.72	1.38	0.19	3.65	0.31
Prototype	-0.97	-0.20	-3.07	0.00	-0.52	3.16	-0.52
Weights	0.18	0.02	0.27	0.00	0.00	0.51	0.00
Mask	1	0	1	0	0	1	0

Therefore, we propose a method that identifies the most informative features shared between an instance and its prototype, guiding the user's attention to a concise subset of features. Further, we refer to them as *alike parts*, where the importance of features within the alike part is similarly high in both the instance and its nearest prototype.

To explain the instance $\mathbf{x}_i$ by its nearest prototype $\mathbf{p}_j$, we first identify the alike parts by computing feature importance scores for each feature $l \in 1, \ldots, d$ in the classification of $\mathbf{x}_i$ and $\mathbf{p}_j$ using the classifier h, denoted as $\phi(h, \mathbf{x}_i^l)$ and $\phi(h, \mathbf{p}_j^l)$, respectively. We use the SHAP method [8] to quantify the influence of each feature, as it is one of the most widely used methods for feature importance estimation. However, any feature importance method can be applied in this context. To ensure comparability, the raw importance scores are normalized, as they can vary in magnitude. We treat both positive and negative scores equally by squaring them, which avoids cancellations and enables the identification of similarities and differences between the instance and its prototype:

$$\hat{\phi}(h, \mathbf{x}_i^l) = \frac{(\phi(h, \mathbf{x}_i^l))^2}{\sum_{k=1}^{d} (\phi(h, \mathbf{x}_i^k))^2}, \quad \hat{\phi}(h, \mathbf{p}_j^l) = \frac{(\phi(h, \mathbf{p}_j^l))^2}{\sum_{k=1}^{d} (\phi(h, \mathbf{p}_j^k))^2}\,. \tag{1}$$

To quantify the alignment of feature importance between the instance and the prototype, we define a weight for each feature as the product of its normalized importance scores:

$$w_l = \hat{\phi}(h, \mathbf{x}_i^l) \cdot \hat{\phi}(h, \mathbf{p}_j^l)\,. \tag{2}$$

These weights are used to determine the degree to which each feature highly influences the prediction of the model for both the prototype and the explained instance. Various operators can achieve this - here we propose to select a subset of the most influential features – by defining a binary mask $\mathbf{m} \in \{0, 1\}^d$, where these features with weights above the mean of all values are retained:

$$m_l = \nVdash \left(w_l > \frac{1}{d} \sum_{k=1}^{d} w_k \right) \tag{3}$$

Table 1 illustrates the identification of a subset of important features.

3.2. New definition of optimization problem

The prototype selection algorithms discussed in this paper, such as [6, 7], define the task of identifying representative data points as a k-medoids problem, which is solved using a greedy approximation algorithm. Typically, the k-medoids problem minimizes a distance function d between each training example $\mathbf{x}_i$ and its nearest prototype $\mathbf{p}_j$. This is expressed as follows:

$$f(\mathcal{P}) = \sum_{i=1}^{|\mathcal{S}|} \min_{\mathbf{p}_j \in \mathcal{P}} d\left(\mathbf{x}_i, \mathbf{p}_j\right), \tag{4}$$

Table 2
An example of an instance and its alike parts identified from the nearest prototype using the A-Pete algorithm [7]. The selection is based on two optimization problem definitions: the original (raw) and the Feature Importance (FI)-informed approach. Parts alike between the explained instance and prototype in the FI-informed approach are bolded, while those alike in the original (raw) strategy are underlined.

type	Pregnancies	Glucose	BloodP.	SkinT.	Insulin	BMI	PedigreeF.	Age
instance	6	**102**	82	0	0	30.8	**0.18**	**36**
prototype (FI)	7	**125**	86	0	0	37.6	**0.30**	**51**
prototype (Raw)	7	62	78	0	0	32.6	0.39	41
instance	**8**	**100**	74	40	215	39.4	**0.66**	43
prototype (FI)	**9**	**152**	78	34	171	34.2	**0.89**	33
prototype (Raw)	9	171	110	24	240	45.4	0.72	54

where the notation $|\mathcal{S}|$ refers to the cardinality of the training set. The choice of the distance function d varies between different algorithms. In neural network-based approaches, it can be a dot product between trainable embeddings [10], or in tree ensembles, a specialized tree distance metric [6, 7].

To strengthen diversification in feature importance, we propose extending the objective function by including an additional feature importance component fi defined as the product of normalized feature importance of l-th feature of instance $\mathbf{x}_i$ and its nearest prototype $\mathbf{p}_j$:

$$fi(\mathbf{x}_i, \mathbf{p}_j) = \sum_{l=1}^{d} \frac{(\phi(h, \mathbf{x}_i^l))^2}{\sum_{k=1}^{d}(\phi(h, \mathbf{x}_i^k))^2} \cdot \frac{(\phi(h, \mathbf{p}_j^l))^2}{\sum_{k=1}^{d}(\phi(h, \mathbf{p}_j^k))^2}. \tag{5}$$

The fi scores can be calculated once for all $\mathbf{x}_i$ prior to optimization and cached for efficiency. The revised function incorporates both the minimization of the distance between each instance and its nearest prototype and an additional term weighted by β to account for the feature importance score. The first term promotes that each instance in the dataset is well represented by a prototype, promoting compact coverage of $\mathcal{S}$ by assigning each instance to its closest prototype, while the second encourages diversification in the feature importance across prototypes. The revised function is formally defined as:

$$f(\mathcal{P}) = \sum_{i=1}^{|\mathcal{S}|} \min_{\mathbf{p} \in \mathcal{P}_j} \left(d\left(\mathbf{x}_i, \mathbf{p}_j\right) + \beta \cdot fi\left(\mathbf{x}_i, \mathbf{p}_j\right) \right), \tag{6}$$

This modification enables a more nuanced global prototype selection, with β balancing distance and feature importance. The updated formulation improves prototype selection for identifying alike parts. The proposed method is robust to missing values, assuming that the selected components can handle them. In this paper, we used prototype selection algorithms [6, 7] based on RF [11], and SHAP, both of which natively support missing values. Therefore, the method does not require additional preprocessing for missing data.

4. Experiments

As discussed in Section 3.2, the proposed optimization method can be adapted to various algorithms. We applied this modification to prototype selection algorithms optimizing tree distance: A-PETE, SM-A, and G-KM [6, 7], to explain the Random Forest (RF) ensemble [11]. All use greedy medoid selection, with key differences: G-KM selects an equal number of prototypes per class (greedy k-Medoid approximation computed within classes); SM-A [6] selects the prototype providing the greatest improvement across all classes; and A-Pete [7] automates this by stopping based on relative improvements (see [7] for pseudo codes). For evaluation, we use four benchmark datasets that have a subset of globally important features:

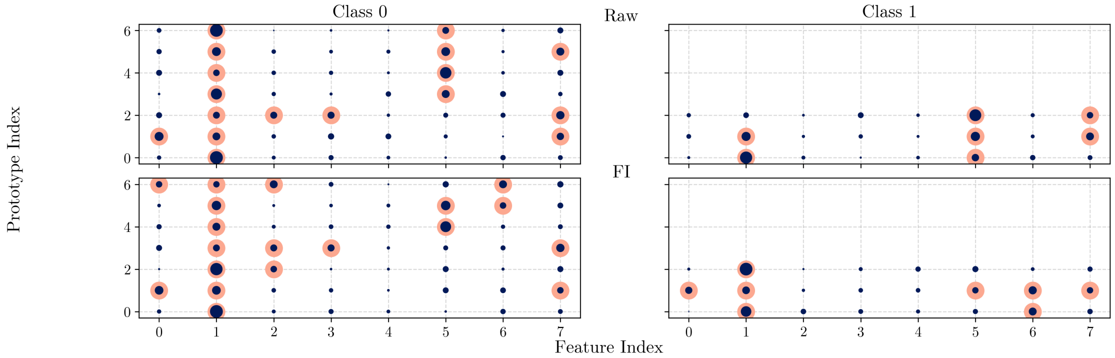

Figure 1: Comparison of prototypes (x-axis – prototype index) and important features (y-axis – feature index) for the Diabetes dataset. The top row displays prototypes generated using the original raw algorithm, while the bottom row incorporates an extended target function with feature importance (FI). The size of the inner circle represents feature importance, and pink highlights features identified as important for a given prototype.

Australia Rain[2], Breast Cancer[3], Diabetes[4], and Passenger Satisfaction[5]; and two: Apple Quality[6] and Wine Quality[7], which exhibits high feature importance across all features.

The experiments are organized as follows: Section 4.1 presents examples of alike parts identification on real data and how extending the optimized function improves this process. Section 4.2 aims to quantify the quality of the proposed improvements by comparing our modified with the original prototype selection methods, highlighting the impact of our changes. Section 4.3 presents an ablation study that analyzes the contribution of the β factor to algorithm performance.

4.1. Studying the methods in action

Finding alike parts on real data is shown in Table 1, illustrating how feature importance for both the instance and prototype is used to compute weights. Table 2 compares how alike parts of an instance and its nearest prototype are selected using the original (raw) and FI-informed versions of the A-Pete for the Diabetes dataset. Incorporating feature importance into A-Pete's optimization led to different selections than the raw algorithm when generating prototypes from black-box RF [11].

For example, when using the prototype from raw A-PETE, only the *Glucose* is highlighted as the feature important for both the instance and prototype. Meanwhile, the FI-informed algorithm also highlights *Diabetes Pedigree Function*, and *Age* which aligns with established medical knowledge on diabetes risk factors [12]. This demonstrates the potential of our method to facilitate the identification of more meaningful relationships between instances and prototypes.

A visual comparison of the globally generated sets of prototypes and selected important attributes for the Diabetes dataset is presented as Figure 1. The figure contrasts prototypes generated using the original (raw) A-Pete algorithm with those generated using the FI-informed approach. The figure demonstrates that the FI-informed algorithm yields more diversified prototypes that highlight parts varying between prototypes – the sixth feature was selected as important only when FI was included in the target function. A similar phenomenon was observed for Australia Rain and Breast Cancer – certain features were considered significant only when using the FI-informed version of the algorithm.

The proposed approach was validated on the test subset of each dataset to quantitatively compare the frequency of features identified as important. In the Figure 2, presenting results, one can observe that the frequency of highlighting each feature differs between the original and FI-informed strategies. This difference is especially noticeable for the G-KM algorithm: when prototypes are selected using the FI-informed strategy, certain features are highlighted that were not emphasized by the raw algorithm.

[2] https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package
[3] https://www.kaggle.com/datasets/rahmasleam/breast-cancer
[4] https://www.kaggle.com/datasets/mathchi/diabetes-data-set
[5] https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction
[6] https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality
[7] https://www.kaggle.com/datasets/taweilo/wine-quality-dataset-balanced-classification

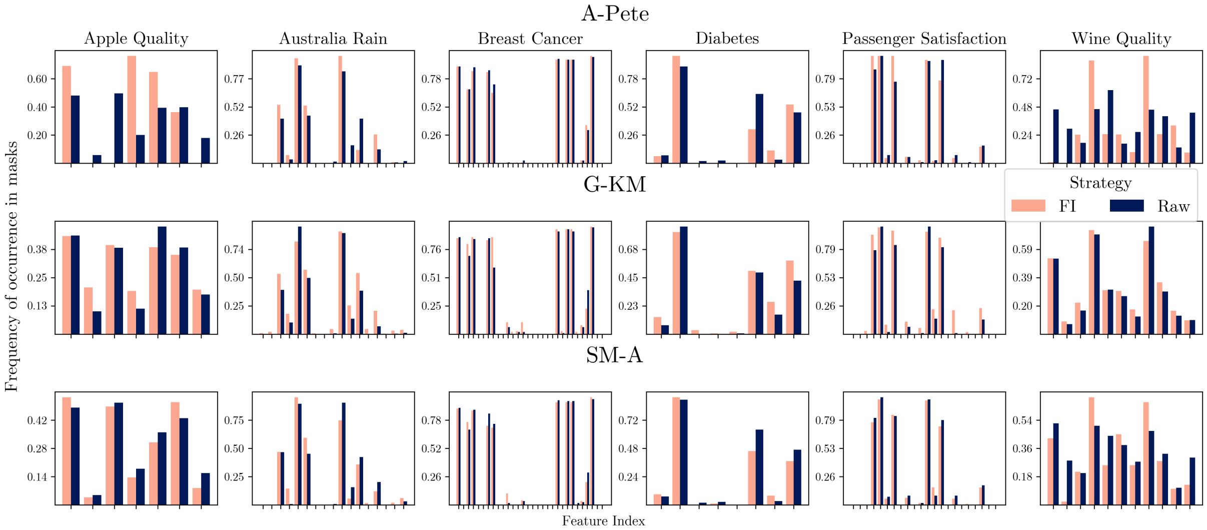

Figure 2: The comparison of the frequency of feature highlighting between the original (raw) and Feature Importance (FI)-informed strategies across different benchmark datasets. The results are shown for three prototype selection algorithms: A-Pete, G-KM, and SM-A.

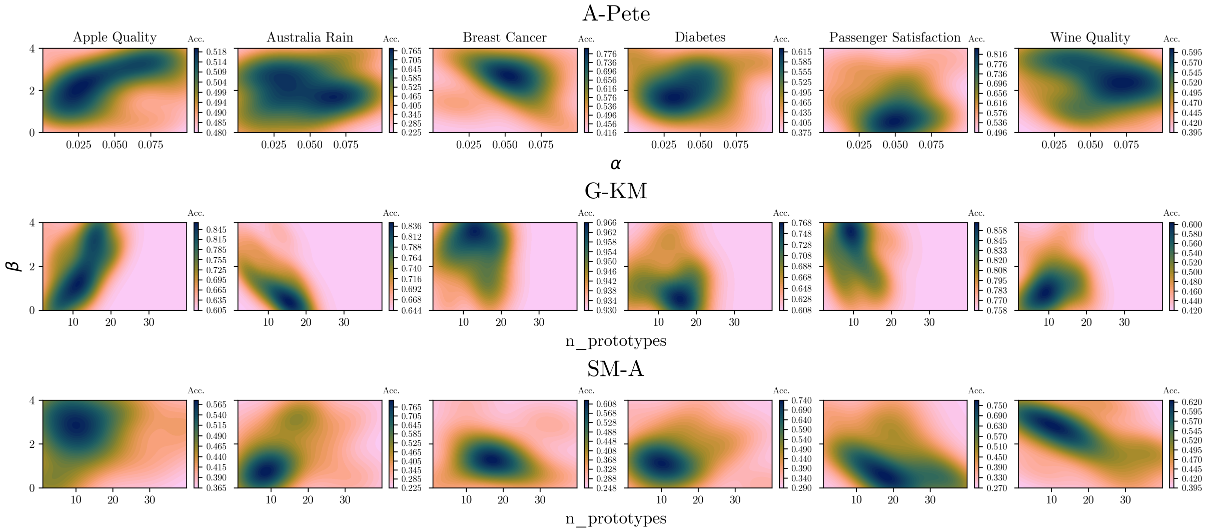

Figure 3: The comparison of accuracy (hue) achieved by A-Pete, G-KM, and SM-A algorithms across benchmarks against algorithm-specific hyperparameters (x-axis) and β (y-axis). Note that the bottom line of each subfigure ($\beta = 0$) represents the original definition of the algorithms, where only the tree distance is minimized.

4.2. Predictive performance in comparison to original versions of the algorithms

This section compares the accuracy achieved by a surrogate model based on prototypes, as it was done in [6, 7]. The surrogate model uses a 1-nearest neighbor (1-NN) search within the set of selected prototypes and is evaluated on classifying instances from a test set. We specifically examine the impact of our modified prototype selection method, which incorporates feature importance.

Figure 3 illustrates how algorithm-specific hyperparameters and the weighting factor β influence prototype selection and consequently impact accuracy, with β controlling the extent to which feature importance is incorporated into the optimization function. The results show that the modified approach maintains or improves predictive performance with respect to main parameters. Similar information is presented in Table 3 where the values corresponding to the accuracy optima found are presented for the original and the FI-incorporated algorithms.

Table 3
Comparison of accuracy achieved by A-Pete, G-KM, and SM-A across benchmarks. The hyperparameters selected for the Feature Importance-informed version of the algorithm correspond to the maxima of accuracy in Figure 3.

Algorithm	Objective function	Dataset					
		Apple Quality	Australia Rain	Breast Cancer	Diabetes	Passenger Satisfaction	Wine Quality
A-Pete	FI	0.520	0.767	0.798	0.623	0.837	0.605
	Raw	0.487	0.424	0.488	0.427	0.783	0.438
G-KM	FI	0.861	0.843	0.965	0.766	0.865	0.602
	Raw	0.785	0.822	0.939	0.739	0.781	0.541
SM-A	FI	0.571	0.809	0.623	0.734	0.779	0.624
	Raw	0.461	0.625	0.344	0.492	0.712	0.448

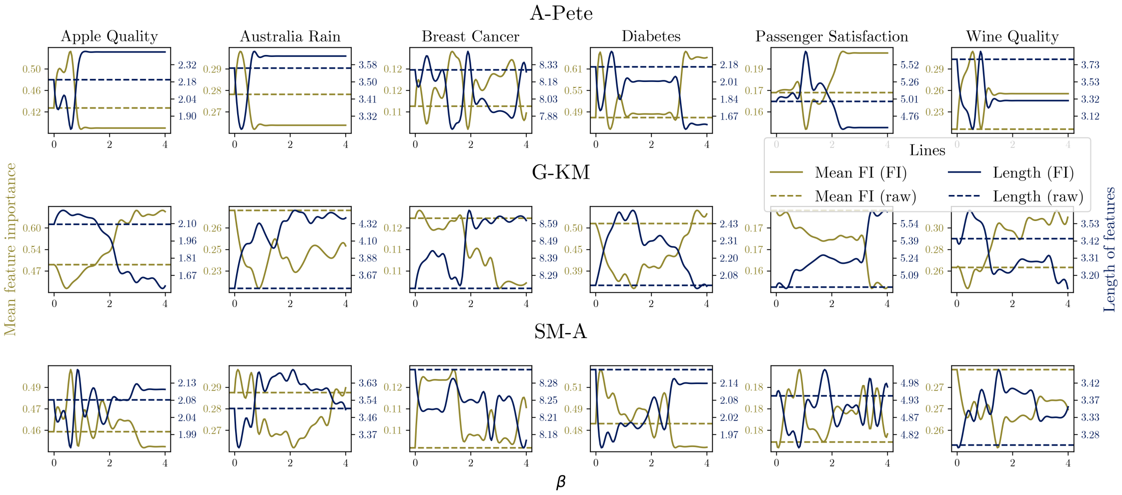

Figure 4: The comparison of mean feature importance of the features included in alike parts (left y-axis) and the length of the vector identified as alike parts between the explained instance and the prototype (right y-axis). The plot illustrates these two values tested against different β values (x-axis).

4.3. Ablation study

Here, we analyze the impact of the parameter β on the selection of the prototype by examining how it influences the alikeness between an explained instance and its prototype. Figure 4 shows how mean feature importance and alike-part length vary with β. The results indicate that as β increases, the mean feature importance similarity tends to rise, suggesting that high β encourages the selection of prototypes that align more closely with important features of the explained instance. However, this trend is not strictly monotonic and careful tuning is required, with $\beta \leq 2.0$ often providing a good balance, although the optimal value depends on the dataset. To determine the optimal value of β, grid search or Bayesian optimization can be used to tune β and other algorithm-specific parameters, aiming to maximize the accuracy of a surrogate 1-NN model.

5. Discussion

This work introduces an innovative approach to prototype-based explanations, enhancing their interpretability by directing user attention to the most important features of both the prototype and the classified instance, the so-called alike parts. By incorporating feature importance into the prototype selection, our proposal bridges a gap in the literature where these two aspects were previously considered separately. The experimental results suggest that this integration improves the clarity of the explanation while preserving and, in some cases, even improving the predictive accuracy (see Section 4.2). Incorporating feature importance leads to selecting prototypes with different, often more meaningful, alike parts. This was shown with the Diabetes dataset, where our method identified features such as *Age* and *Pedigree Function* as crucial, aligning with established medical knowledge (see Section 4.1). Moreover, it can extend beyond the tested algorithms, G-KM, SM-A, A-PETE, and a black-box RF. Importantly, Section 4.3 shows that adjusting the weighting factor β fine-tunes the balance between feature importance and distance minimization, highlighting adaptability to different tasks. Future research should explore its effectiveness from the user perspective, assessing whether these explanations enhance human understanding of model decisions. Furthermore, evaluating the approach on non-tabular modalities, such as images and text, is necessary to assess its broader applicability.

Acknowledgments

This research was funded in part by National Science Centre, Poland OPUS grant no. 2023/51/B/ST6/00545 and in part by PUT SBAD 0311/SBAD/0752 grant.

Declaration on Generative AI

The author has not employed any Generative AI tools.

References

[1] F. Bodria, F. Giannotti, R. Guidotti, F. Naretto, D. Pedreschi, S. Rinzivillo, Benchmarking and survey of explanation methods for black box models, Data Mining and Knowledge Discovery 37 (2023) 1719–1778. doi:10.1007/s10618-023-00933-9.

[2] O. Menis Mastromichalakis, G. Filandrianos, J. Liartis, E. Dervakos, G. Stamou, Semantic prototypes: Enhancing transparency without black boxes, in: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 1680–1688. doi:10.1145/3627673.3679795.

[3] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, C. Rudin, This looks like that: deep learning for interpretable image recognition, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2019.

[4] F.-M. Schleif, T. Villmann, B. Hammer, P. Schneider, Efficient kernelized prototype based classification, International Journal of Neural Systems 21 (2011) 443–457.

[5] X. Zhang, H. Xiao, R. Gao, H. Zhang, Y. Wang, K-nearest neighbors rule combining prototype selection and local feature weighting for classification, Knowledge-Based Systems 243 (2022) 108451. doi:10.1016/j.knosys.2022.108451.

[6] S. Tan, M. Soloviev, G. Hooker, M. T. Wells, Tree space prototypes: Another look at making tree ensembles interpretable, in: Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference, FODS '20, 2020, p. 23–34. doi:10.1145/3412815.3416893.

[7] J. Karolczak, J. Stefanowski, A-PETE: Adaptive prototype explanations of tree ensembles, in: Progress in Polish Artificial Intelligence Research, volume 5, Warsaw University of Technology, 2024, pp. 2–8. URL: https://pages.mini.pw.edu.pl/~estatic/pliki/PP-RAI_2024_proceedings.pdf.

[8] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., 2017, p. 4768–4777. doi:10.5555/3295222.3295230.

[9] M. Biehl, B. Hammer, T. Villmann, Prototype-based models in machine learning, WIREs Cognitive Science 7 (2016) 92–111. doi:10.1002/wcs.1378.

[10] O. Li, H. Liu, C. Chen, C. Rudin, Deep learning for case-based reasoning through prototypes: a neural network that explains its predictions, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI Press, 2018. doi:10.5555/3504035.3504467.

[11] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32. doi:10.1023/A:1010933404324.

[12] A. Kautzky-Willer, J. Harreiter, G. Pacini, Sex and gender differences in risk, pathophysiology and complications of type 2 diabetes mellitus, Endocrine Reviews 37 (2016) 278–316.