Polar Encoding: A Simple Baseline Approach for Classification with Missing Values

Oliver Urs Lenz, Daniel Peralta, and Chris Cornelis

***Abstract*—We propose polar encoding, a representation of categorical and numerical $[0, 1]$-valued attributes with missing values to be used in a classification context. We argue that this is a good baseline approach, because it can be used with any classification algorithm, preserves missingness information, is very simple to apply and offers good performance. In particular, unlike the existing missing-indicator approach, it does not require imputation, ensures that missing values are equidistant from non-missing values, and lets decision tree algorithms choose how to split missing values, thereby providing a practical realisation of the *missingness incorporated in attributes* (MIA) proposal. Furthermore, we show that categorical and $[0, 1]$-valued attributes can be viewed as special cases of a single attribute type, corresponding to the classical concept of barycentric coordinates, and that this offers a natural interpretation of polar encoding as a fuzzified form of one-hot encoding. With an experiment based on twenty real-life datasets with missing values, we show that, in terms of the resulting classification performance, polar encoding performs better than the state-of-the-art strategies *multiple imputation by chained equations* (MICE) and *multiple imputation with denoising autoencoders* (MIDAS) and — depending on the classifier — about as well or better than mean/mode imputation with missing-indicators.**

***Index Terms*—barycentric coordinates, classification, decision trees, fuzzy partitions, missingness incorporated in attributes, missing values, nearest neighbours, one-hot encoding.**

I. Introduction

MISSING values are a frequent issue in real-life datasets and a subject of ongoing research [1]–[3]. In the present paper, we consider what a good baseline approach is for handling missing values in the context of classification.

Missing values have been extensively studied in the context of statistical inference. For estimating a parameter value, the generally accepted approach is to perform *multiple imputation* [4], in which one models the posterior distribution of the values that are missing on the basis of the non-missing values. By drawing from this distribution, one obtains a sample of imputed datasets and a corresponding sample of the estimand, allowing one to estimate the true parameter value and determine the uncertainty of this estimate due to the missing values. Two popular multiple imputation proposals are *multiple imputation by chained equations* (MICE) [5] and *multiple imputation with denoising autoencoders* (MIDAS) [6].

The explicit assumption behind multiple imputation is that the distribution of missing values can be estimated on the basis of non-missing values (*missing at random* (MAR)). In contrast, the assumption behind the *missing-indicator* approach [7] is that missingness is potentially informative (*missing not at random* (MNAR)), and that this aspect of the data should be explicitly represented through binary indicator attributes, that record for each original attribute whether the value was missing. If one assumes that missing values are not part of the 'true' model, missing-indicators introduce bias [8], and for this reason they have generally been dismissed in the context of statistical inference.

In the context of machine learning, and of classification in particular, model bias is arguably less important than prediction performance. We have previously established, through the first large-scale evaluation of missing-indicators on real-life datasets, that these do generally increase classification performance [9].

For decision trees, missingness is also preserved by the *missingness incorporated in attributes* (MIA) approach [10], which stipulates that the tree construction algorithm should evaluate two versions of each split, with missing values included on either side. MIA has been shown to outperform imputation with or without missing indicators [11].

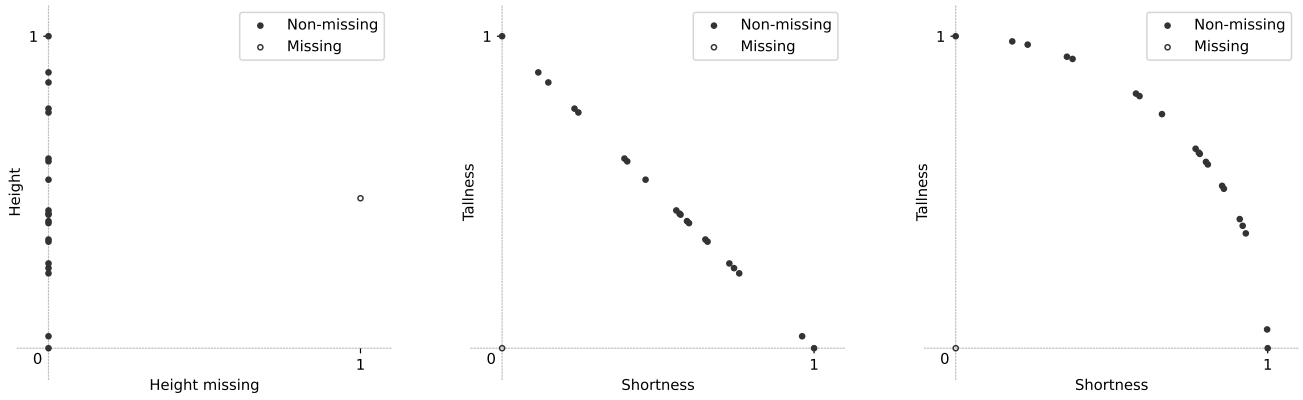
We conclude from this that missing values are an important part of a dataset, that should be made available for classifiers to learn from just like non-missing values. While missing-indicators can be used for this, there are two aspects that prevent them from being an ideal baseline approach towards missing values. Both stem from the fact that missing-indicators have to be combined with imputation. Firstly, this means that the practitioner still needs to make a choice — which imputation method to use. And secondly, while missing-indicators preserve missing values, we will see that to a certain extent, the imputation still induces the classifier to treat missing values like their imputed values.

To address this, we introduce in the present paper a new approach towards missing values called *polar encoding*, which can be used with categorical and $[0, 1]$-scaled numerical attributes. Polar encoding represents missing values without relying on imputation, leaving it completely up to the classifier how to learn from missing values. As we will see, polar encoding is a very simple proposal, but to the best of our knowledge, it has never been suggested before.

We will proceed by defining polar encoding, and comparing it against imputation and missing-indicators on the basis of four criteria that a good baseline approach towards missing values should satisfy (Section II). Next, we specifically explain why polar encoding is a good approach for distance-based (Section III) and decision tree (Section IV) classifiers. In Section V, we offer additional conceptual motivation for polar encoding by arguing that it can be seen as a fuzzification of

O. U. Lenz and C. Cornelis are with the Department of Applied Mathematics, Computer Science and Statistics, Ghent University, e-mail: {oliver.lenz,chris.cornelis}@ugent.be.

D. Peralta is with the Department of Information Technology, Ghent University – imec, e-mail: daniel.peralta@ugent.be



(a) Missing-indicator (b) Boscovich polar encoding (c) Euclidean polar encoding

Fig. 1. Illustrative example of a $[0, 1]$-valued attribute for height with missing value, with missing-indicator and polar encoding.

one-hot encoding.

We complement these theoretical arguments in Section VI with an experimental evaluation of the downstream classification performance of polar encoding, by comparing it against MICE and MIDAS, as well as against missing-indicators paired with mean/mode imputation, on the basis of twenty real-life datasets with missing values. Finally, we present our conclusions in Section VII.

II. Polar encoding as a good baseline approach

We will now present polar encoding[1], and discuss why it is a good baseline approach towards missing values. For comparison, one-hot encoding [12] is a simple baseline solution to the related problem of handling categorical attributes with algorithms that expect numerical input. It preserves the information encoded in categorical attributes and results in a dataset that can be fed to any numerical algorithm. Polar encoding is a similar solution, but for missing values.

For categorical attributes, polar encoding corresponds exactly to one-hot encoding, with missing values represented as zero vectors. Meanwhile, each $[0, 1]$-scaled numerical attribute is converted into a pair of features with the following map[2]:

$$\begin{aligned} x &\longmapsto \langle x, 1 - x \rangle \,, \\ ? &\longmapsto \langle 0, 0 \rangle \,, \end{aligned} \tag{1}$$

where x is any non-missing value, and $?$ a missing value. The resulting representation is illustrated by Fig. 1b, which contrasts with the representation produced by the missing-indicator approach (Fig. 1a).

We propose that in the context of classification, the qualities of a good baseline approach towards missing values are embodied by the following four criteria:

Modularity. The baseline approach should be self-contained. It should result in a complete, numerically encoded dataset, allowing classification algorithms to be agnostic about missing values.

Conservatism. The baseline approach should be a faithful representation of the original dataset. It should presuppose as little as possible about how missing values contribute to the learning task.

Simplicity. The baseline approach should be simple to apply. It should require a minimal amount of computational effort and no parameter choices by the user.

Performance. The baseline approach should enable good downstream prediction performance. It should perform well on average across real-life classification problems.

We may have to accept a certain trade-off between these criteria. For example, the simplicity and modularity of a good baseline approach may outweigh slightly lower downstream performance compared to a vastly more complicated solution. Conversely, we could accept a less conservative approach as a good baseline if it combined simplicity and modularity with superior performance. However, it turns out that in the context of supervised learning, conservatism, simplicity and performance appear to go somewhat hand in hand.

Imputation satisfies modularity, as it replaces missing values with estimates and the resulting complete dataset can be fed to any classification algorithm. However, by design, it is not a conservative approach towards missing values, since it predetermines the contribution of missing values towards the classification task by replacing them. Most imputation algorithms are not simple either, requiring substantial amounts of computation and user input.

Arguably the simplest form of imputation is imputation with the mean of numerical attributes and the mode of categorical attributes. Mean/mode imputation is not considered a good solution for statistical inference because it introduces bias. However, somewhat counterintuitively, it does not necessarily lead to worse prediction performance in the context of supervised learning [13]. The reason for this is that missing values can be informative, and it is precisely because mean/mode imputation

[1]We have chosen the name *polar encoding* as a loose analogy to polar coordinates, because values are encoded in relation to a number of poles: the origin and the unit vectors $\langle 1, 0 \rangle$ and $\langle 0, 1 \rangle$ (and higher-dimensional unit vectors for categorical attributes).

[2]This is the default form of polar encoding, to be used with (Boscovich) 1-distance and with algorithms not based on distance. We will propose a separate form of polar encoding to be used with (Euclidean) 2-distance in Subsection III-B.

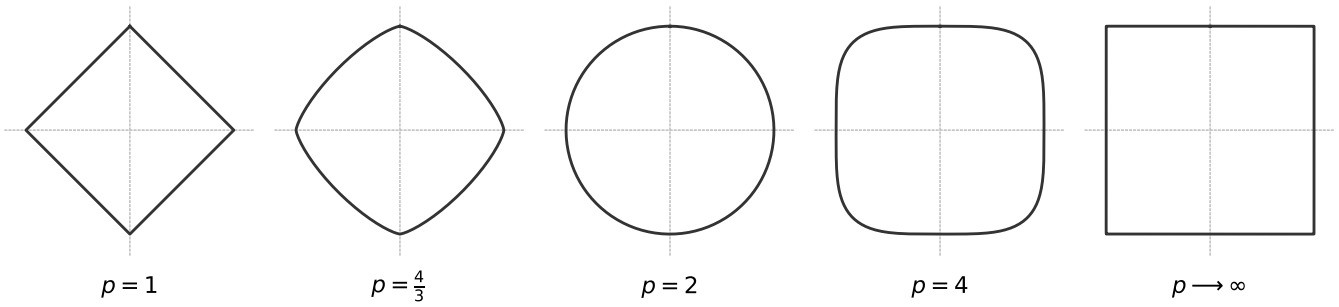


Fig. 2. Minkowski p-norm unit circles for various values of p.

fails to hide missing values well that this information remains partially available for prediction algorithms to learn from.

The missing-indicator approach is more conservative than imputation, because it preserves missing values. It also generally increases classification performance on real-life datasets [9]. However, the missing-indicator approach is not maximally conservative. As it has to be combined with imputation, it induces classification algorithms towards treating missing values like their imputed values. For algorithms that are based on distance, like nearest neighbours algorithms and support vector machines, the missing-indicator approach represents a missing value as being closer to its imputed value (e.g. the mean) than to other values (Fig. 1a). For decision tree algorithms, missing values will always split together with their imputed value when the algorithm splits on the original attribute.

Finally, because MIA lets decision trees choose how to split missing values, it is a conservative approach. However, it is not modular, since it requires an adaptation of the prediction algorithm itself, and because it can only be used with decision trees.

In contrast to these existing proposals, polar encoding satisfies all four criteria. Polar encoding is modular, since it results in a complete, numerical dataset that can be used with any classification algorithm. It is also simple. Being essentially a linear transformation of the data, it can be applied quickly and easily, without the need for any dedicated software. In the next two sections, we will show that polar encoding is conservative. In particular, we will argue that by making missing values equidistant from non-missing values, polar encoding does not presuppose their contribution to the classification problem, and that for decision trees, it is effectively a modular implementation of MIA. Finally, we will show in Section VI that the performance of polar encoding is also as good or better than the alternatives.

III. Polar encoding and distance-based classifiers

In this section, we will explain how polar encoding ensures that missing values are equidistant from all non-missing values, and present a variant proposal for Euclidean distance.

A. Boscovich distance

Recall the general definition of the Minkowski p-norm of a vector $x \in \mathbb{R}^m$, for $p \geq 1$:

$$|x|_p := \left(\sum_{i \leq m} |x_i|^p \right)^{\frac{1}{p}} . \tag{2}$$

The Minkowski p-distance between any two points $x, y \in \mathbb{R}^m$ is the p-norm of their difference. The p-norm unit sphere in $\mathbb{R}^m$ consists of all points with p-norm equal to 1. For $m = 2$, this gives us the p-norm unit circles (Fig. 2).

Two values of p are particularly often used in machine learning. When $p = 1$, we obtain the Boscovich norm[3], which reduces to $\sum_{i \leq m} |x_i|$, and when $p = 2$, we obtain the Euclidean norm[4].

Fig. 1b illustrates the application of polar encoding with a toy example. The key observation to make is that unlike the missing-indicator approach, the Boscovich distance between a missing value and any non-missing value is always 1. In fact, this is a simple consequence of the fact that polar encoding maps non-missing values onto the non-negative quadrant of the Boscovich unit circle.

Moreover, with polar encoding, the Boscovich distance between any two non-missing values $x, y \in [0, 1]$ becomes twice the original distance $|x - y|$. In other words, the distances between non-missing values remain essentially unchanged, except for a scaling factor of 2. The Boscovich distance between a missing value and non-missing values is 1, which is exactly half the maximum distance 2 between two non-missing values, reflecting the fact that we do not know what the 'true' value of a missing value is. This distance can be used directly, or transformed into a similarity value with $a \longmapsto 1 - a/2$. In this case, the similarity between a missing value and any non-missing value is always 0.5, exactly half the maximum similarity of 1.

This contrasts with the approach taken in [19], where the similarity between a missing value and any other value is

[3]Perhaps first used implicitly by Roger Joseph Boscovich (1711–1787) to minimise regression residuals [14]–[18]; also known as *city block*, *Manhattan*, *rectilinear* and *taxicab* norm.

[4]Also known as Pythagorean norm.

stipulated to always be 1. Similarly, the authors of the present paper have proposed [20] (based on previous work [21]) to propagate the uncertainty from missing values using interval-valued fuzzy sets. These interval values are bounded by an optimistic scenario, corresponding to the proposal in [19], and a pessimistic scenario, in which the similarity between a missing value and any other value (possibly also missing) is 0 (complete dissimilarity). In both cases the problem is that missing values are not more similar to each other than to non-missing values — missing values are not treated as a signal to generalise from. Moreover, in practice these similarity relations scale poorly to larger datasets, because they do not admit straightforward implementations in terms of an existing distance measure.

B. Euclidean distance

Based on the discussion in the previous subsection, a straightforward way to obtain polar encoding for Euclidean distance is to map non-missing values onto the non-negative quadrant of the Euclidean unit circle (Fig. 1c). We propose to do this with the following mapping, which establishes a linear correspondence between distance in $[0, 1]$ and arc length (scaling by a factor $\sqrt{2}$):

$$\begin{aligned} x &\longmapsto \left\langle \sin \frac{x \cdot \pi}{2}, \cos \frac{x \cdot \pi}{2} \right\rangle, \\ ? &\longmapsto \langle 0, 0 \rangle \, . \end{aligned} \tag{3}$$

Note that this map cannot preserve Euclidean distance. When encoding a $[0, 1]$-valued attribute in this manner, larger distances become relatively less large. However, the difference is relatively small and may not be problematic in practice. For instance, the Euclidean distance between the minimum and maximum values becomes $\sqrt{2} \approx 1.41$, which is slightly less than twice the distance between either value and the midrange:

$$\left(\left| \sin \frac{\pi}{4} - 0 \right|^2 + \left| \cos \frac{\pi}{4} - 1 \right|^2 \right)^{\frac{1}{2}} \approx 0.765.$$

Furthermore, this maximum distance between two non-missing values ($\sqrt{2}$), is now comparatively smaller than with Boscovich distance (2). This is completely consistent with the distance between two different one-hot encoded categorical values, which is likewise $\sqrt{2}$ for Euclidean distance and 2 for Boscovich distance.

For other values of p, there exist generalisations of $\sin$ and $\cos$ that could be used instead to parametrise the non-negative quadrant of the p-unit sphere [22], [23]. However, these functions are defined as the inverses of integrals, and so are not easy to apply in practice.

IV. Polar encoding and decision tree classifiers

Polar encoding also allows decision tree algorithms to learn from missing values. The two dimensions of a polar-encoded attribute induce identical splits on the data, except that missing values end up on either side of each split (Fig. 3). Therefore, decision trees are effectively offered a choice as to which side of each split missing values should be grouped with. Missing values can also be split off on their own by splitting on both dimensions of a polar-encoded attribute.

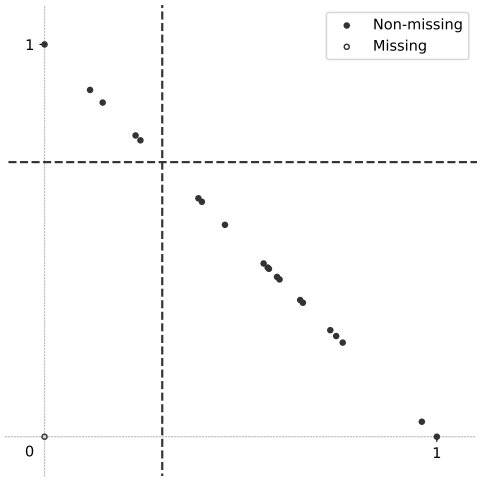


Fig. 3. Illustrative example of equivalent splits on a polar-encoded attribute, with missing values on either side.

This contrasts with the missing-indicator approach, where missing values either group together with their imputed value (when the tree splits on the original attribute), or alone (when the tree splits on the missing-indicator).

The effect of polar encoding on decision trees is very similar to the *missingness incorporated in attributes* (MIA) proposal [10] which stipulates that when splitting on an attribute with missing values, the algorithm should consider each potential split twice, with missing values on either side, and additionally a split that separates non-missing and missing values. MIA has been added to the scikit-learn [24] implementation of LightGBM [25], and a similar strategy is part of XGBoost [26]. The advantage of polar encoding is that it can be applied by the user, and combined with off-the-shelf implementations of decision tree algorithms that do not natively support MIA.[5]

The performance of MIA has mostly been evaluated on the basis of simulated data with informative missing values.

For decision trees, MIA performs better [10] than resolving missing values as a weighted combination of the two branches according to the prior probabilities of the non-missing values [27], and about as good as multiple imputation with expectation maximisation [28], which had emerged as the two best-performing strategies in a previous comparison [29].

For Bayesian additive regression trees, MIA has been shown to outperform random forest imputation [30]. Similarly, MIA has been shown to outperform mean imputation with missing-indicators and a handful of other strategies for regression with decision trees, Random Forest and XGBoost [11].

Finally the scikit-learn implementation of LightGBM mentioned above has also been evaluated on four large, real-life medical datasets, and MIA was found to produce somewhat to considerably better performance than the missing-indicator approach with various forms of imputation [13].

[5] A similar trick is suggested in [11]: repeat each attribute with missing features twice, and encode missing values alternatively as $-\infty$ and $+\infty$.

$$\left(\begin{array}{c|c|c} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{array}\right) \qquad \left(\begin{array}{ccc} 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 1 & 0 & 0 \\ \hline 0 & 0 & 1 \end{array}\right)$$

(a) Crisp partition (b) Categorical attribute

Fig. 4. Example illustrating the correspondence between crisp partitions and categorical attributes of a dataset X. Rows correspond to the records of X, columns to the partition classes and categories. The values 1 and 0 indicate membership and non-membership, respectively.

V. Polar encoding as representation of barycentric attributes

In this section, we will show how polar encoding can be seen as the representation of *barycentric* attributes, which generalise both categorical and $[0, 1]$-valued attributes. In particular, this explains how polar encoding generalises one-hot encoding. To begin with, we establish our working definitions of datasets, attributes and one-hot encoding.

A. Numerical and categorical attributes

A key difference between numerical and categorical attributes is that while the values of numerical attributes can be assumed to lie in $\mathbb{R}$, allowing us to construct machine learning models based on the arithmetic of $\mathbb{R}$, the set of values V of a categorical attribute is not assumed to have any relevant internal structure.

However, many algorithms are only defined for numerical data, and one popular solution, perhaps first documented by Suits (1957) [12] (but "not new" even then), is to transform a categorical attribute into a tuple of numerical features through *one-hot* encoding (or encoding with *dummy variables*).

Definition 1. Let V be a categorical attribute. For a chosen order $V = (v_1, v_2, \ldots, v_p)$, its *(redundant) one-hot encoding* is the map $V \longrightarrow [0, 1]^p$ that sends v_i to the standard basis vector $\mathbf{e}_i = \langle 0, \ldots, 0, 1, 0, \ldots, 0\rangle$ for all $i \leq p$, while its *compact one-hot encoding* is the map $V \longrightarrow [0, 1]^{p-1}$ that sends v_p to $\mathbf{0}$ and v_i for $i < p$ to $\mathbf{e}_i$.

Compact one-hot encoding is sufficient to ensure that all categorical values are linearly separable, but it also introduces an asymmetry that can be undesirable.

Remark 1. Binary attributes can be represented both as categorical attributes and as numerical attributes. In the latter case, a typical choice is to use the values 0 and 1. This numerical representation corresponds directly to a compact one-hot encoding of its categorical representation. We will exploit this correspondence to argue that barycentric attributes generalise not just categorical, but also $[0, 1]$-valued numerical attributes.

It is a classical observation that categorical attributes correspond to partitions [31]. Formally, a categorical attribute V induces a partition on a dataset X through the equivalence relation that equates elements of X with the same value in V. Conversely, if we have a partition $\mathcal{U}$ of X, we can derive a categorical attribute of X that takes, for each $x \in X$, the value U in $\mathcal{U}$ that contains y.

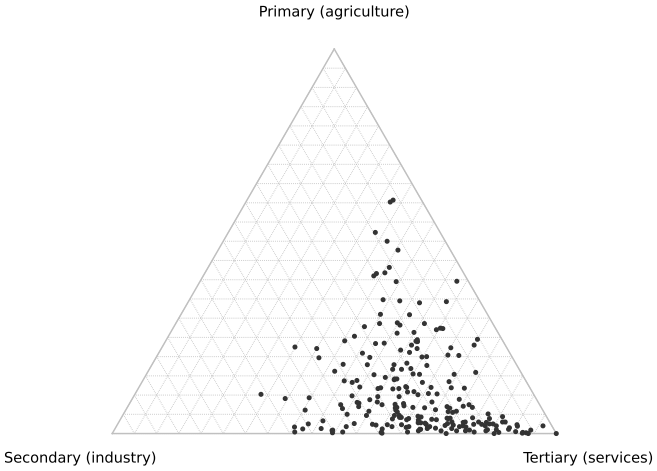


Fig. 5. Example of a ternary plot: distribution of GDP over economic sectors of countries and territories [32].

Both categorical attributes (through one-hot encoding) and partitions can be represented with a matrix of values in $\{0, 1\}$, with exactly one value equal to 1 on each row (Fig. 4). In Subsection V-C, we will extend this correspondence between categorical attributes and partitions to barycentric attributes and fuzzy partitions.

B. Barycentric attributes

Barycentric values (or *coordinates*; also known as *homogeneous* coordinates) are numerical values that sum to a fixed number (typically 1), or where only the relative proportions are considered important. The concept dates back to at least Möbius (1827) [33]–[35], who used it to express a point as the weighted sum (the *barycentre*, where the weights cancel each other out) of the vertices of a simplex. Barycentric values are also used to define the points that make up projective space. For the purpose of the present paper, we will assume that barycentric values are non-negative, and use the following formal definition:

Definition 2. An attribute is *barycentric* if it is equal to a copy of $\left(\mathbb{R}^m_{\geq 0} \setminus \{\mathbf{0}\}\right) / \sim$ for some $m \geq 1$, where $\sim$ is the equivalence relation defined by $\langle x_1, x_2, \ldots, x_m\rangle \sim \langle \lambda x_1, \lambda x_2, \ldots, \lambda x_m\rangle$ for all $\lambda \in \mathbb{R}_{>0}$. The *normalised representation* of a value $[x_1, x_2, \ldots, x_m] \in \left(\mathbb{R}^m_{\geq 0} \setminus \{\mathbf{0}\}\right) / \sim$ is the vector $\langle x_1/s, x_2/s, \ldots, x_m/s\rangle \in \mathbb{R}^m$, where $s := \sum_{i \leq m} x_i$.

Barycentric values are often encountered in the literature in the form of ternary plots (Fig. 5), which display the relative frequencies of three components. Recent examples include the composition of planets (core, mantle and hydrosphere) [36]–[38], seabed sediment [39], ternary mixtures of fluids [40], [41], ternary compounds [42], [43] and even human behaviour [44], [45].

In addition, some machine learning problems are typically approached by considering relative token frequencies. For

$$\left(\begin{array}{c|c|c} 0.2 & 0.6 & 0.2 \\ 0.1 & 0.9 & 0.0 \\ 0.4 & 0.1 & 0.5 \\ 1.0 & 0.0 & 0.0 \\ 0.1 & 0.9 & 0.0 \\ 0.2 & 0.2 & 0.6 \end{array}\right)$$

(a) Fuzzy partition

$$\left(\begin{array}{ccc} 0.2 & 0.6 & 0.2 \\ \hline 0.1 & 0.9 & 0.0 \\ \hline 0.4 & 0.1 & 0.5 \\ \hline 1.0 & 0.0 & 0.0 \\ \hline 0.1 & 0.9 & 0.0 \\ \hline 0.2 & 0.2 & 0.6 \end{array}\right)$$

(b) Barycentric attribute

Fig. 6. Example illustrating the correspondence between fuzzy partitions and fuzzy categorical attributes of a dataset X. Rows correspond to the records of X, columns to the partition classes and categories. Values are membership degrees.

instance, this can be part of the calculation of the cosine similarity between text records [46]–[48].

Finally, the confidence scores produced by a classification model (or some other estimate), when normalised to sum to 1, are also a natural example of barycentric values.

C. Barycentric attributes as fuzzified categorical attributes

Barycentric attributes generalise categorical attributes in the following way. If $\left(\mathbb{R}^m_{\geq 0} \setminus \{\mathbf{0}\}\right) / \sim$ is a barycentric attribute, then the subset V of values with only one non-zero coefficient forms a categorical attribute, and we will write $B(V) := \left(\mathbb{R}^m_{\geq 0} \setminus \{\mathbf{0}\}\right) / \sim$ and say that V is the set of categories of $B(V)$. In particular, the normalised representation of $B(V)$ reduces precisely to one-hot encoding when restricted to V.

This relationship can also be understood geometrically. The set of normalised representations of a barycentric attribute coincides with the standard $m-1$-simplex, which is spanned by m vertices, the one-hot encoded values of V.

Conversely, barycentric attributes can be understood as fuzzified categorical attributes, allowing us to give a fuzzy answer to the question of category membership:

Remark 2. Let $B(V)$ be a barycentric attribute with m categories. Then we can associate to each value in $B(V)$ with normal representation $\langle x_1, x_2, \ldots, x_m \rangle$ the fuzzy set in V with membership degrees $x_1, x_2, \ldots, x_m$. These are precisely the fuzzy sets in V with cardinality 1.

This is reinforced by the fact that barycentric attributes correspond to fuzzy partitions in the same way that categorical attributes correspond to crisp partitions (Subsection V-A). Recall the definition of a fuzzy partition [49], [50]:

Definition 3. Let X be a finite set. A *fuzzy partition* on X is a finite set $\mathcal{F}$ of fuzzy sets in X such that for each $x \in X$, we have $\sum_{F \in \mathcal{F}} F(x) = 1$.

To see that a barycentric attribute $B(V)$ on a dataset X contains the same information as a fuzzy partition on X, consider that both can be represented by a $|X| \times |V|$ matrix of values in $[0, 1]$, such that the rows sum to 1 [51]. The columns of such a matrix correspond to a fuzzy partition (Fig. 6a), whereas its rows correspond to the normalised values of a barycentric attribute (Fig. 6b).

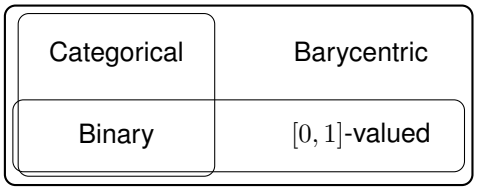


Fig. 7. Euler diagram of different attribute types. Barycentric attributes generalise both categorical and $[0, 1]$-valued attributes.

D. $[0, 1]$-valued attributes as barycentric attributes

Just as one-hot encoding is redundant and we can use compact one-hot encoding to represent the same information with one fewer value (Definition 1), so the normalised representation of a barycentric attribute $\langle x_1, x_2, \ldots, x_m \rangle$ is redundant, and we can encode it compactly as $\langle x_1, x_2, \ldots, x_{m-1} \rangle$. Together, these compactly encoded values form the $m-1$-simplex in $\mathbb{R}^{m-1}$ spanned by the standard $m-2$-simplex and the origin. Conversely, we can reconstruct the full representation from a compactly encoded value $\langle x_1, x_2, \ldots, x_{p-1} \rangle$ by appending the value $1 - \sum_{i \leq p-1} x_i$.

The compact encoding of a barycentric attribute with only two categories is a single value in $[0, 1]$. This leads us to the following observation:

Remark 3. Let A be a $[0, 1]$-valued attribute. Then the values of A are compactly encoded values of a barycentric attribute with two categories. We obtain the corresponding redundant representation with $x \longmapsto \langle x, 1 - x \rangle$. Thus, barycentric attributes generalise not just categorical attributes, but also $[0, 1]$-valued attributes (Fig. 7).

This redundant representation of $[0, 1]$-valued attributes generalises the categorical representation of binary attributes that we noted in Remark 1. We can illustrate this with an example. Suppose that we have a binary attribute denoting height, with two values, 'short' and 'tall'. Its compact encoding is as a single numerical attribute A with two values, 0 and 1, expressing 'tallness'. Its redundant encoding is as two numerical attributes, tallness (A) and shortness ($1-A$). Likewise, suppose that we have $[0, 1]$-valued attribute A' denoting height, then its redundant encoding $\langle A', 1 - A' \rangle$ consists of fuzzy expressions of 'tallness' and 'shortness'.

Of course, this redundant encoding of a $[0, 1]$-valued attribute is precisely the polar encoding that we propose in this paper (Fig. 1b).

E. Representing missing values

We now turn to the representation of missing values. Recall our example from the previous subsection: suppose that we have a barycentric attribute $B(V)$ denoting height, with V containing the two categories 'tall' and 'short', then a missing value does not convey positive information about either category. Therefore, we accommodate the possibility that a barycentric attribute can have a missing value by expanding the set $\left(\mathbb{R}^m_{\geq 0} \setminus \{\mathbf{0}\}\right) / \sim$ to $\mathbb{R}^m_{\geq 0} / \sim$, and by stipulating that the normalised representation of $[0, 0, \ldots, 0]$ is the zero vector $\mathbf{0}$.

TABLE I
CLASSIFICATION ALGORITHMS

Distance-based classifiers	
NN	Nearest Neighbours [54]
NN-D	Nearest Neighbours, distance-weighted [55]
FRNN	Fuzzy Rough Nearest Neighbours [56] with OWA [57]
SVM-G	Soft-margin Support Vector Machine [58] with Gaussian kernel
Decision tree classifiers	
CART	Classification and Regression Tree [59]
RF	Random Forest [60]
ERT	Extremely Randomised Trees [61]
ABT	Ada-Boosted Trees [62] with SAMME [63]
GBM	Gradient Boosting Machine [64]

This corresponds to the unique fuzzy set in V with cardinality 0 (the empty set).

Note that barycentric attributes with missing values can no longer be represented compactly, since doing so would also encode the non-missing value $[0, 0, \ldots, 1]$ as $\mathbf{0}$. It is precisely the redundancy of the redundant normal representation (in particular, redundant one-hot encoding) that enables us to encode missing values as zeroes. For $[0, 1]$-valued numerical attributes, this means that our proposed polar encoding is necessary if we want to represent missing values.

VI. EXPERIMENTAL EVALUATION

We now describe our experimental evaluation of using polar encoding for classification. Concretely, we ask whether it leads to better classification performance than the sophisticated imputation strategies MICE and MIDAS and mean/mode imputation with missing-indicators.

A. Setup

For MICE, we will use the recent *miceforest* implementation for Python [52], which employs LightGBM [25] to obtain predictions, while for MIDAS, we use the *MIDASpy* implementation for Python [53]. Since we want to obtain a single dataset that can be used as input for various classification algorithms, we perform single rather than multiple imputation. Otherwise, we use default hyperparameter values. We use our own implementations of mean/mode imputation with missing-indicators and polar encoding, in the latter case by manually applying the transformations (1) and (3) in Python.

We evaluate our selection of missing data approaches for two sets of classifiers: distance-based and decision tree–based algorithms (Table I). For the Support Vector Machine with Gaussian kernel that is based on Euclidean distance, we evaluate the Euclidean variant of polar encoding, while for the nearest neighbour algorithms that allow setting the distance measure as a hyperparameter, we evaluate both the standard and the Euclidean variant.

We use the same collection of twenty datasets from the UCI repository for machine learning [83] with naturally occurring missing values that we previously used in [9] (Table II). These datasets show great variation — they cover a number of different domains and contain between 155 and 76 000 records, between 4 and 590 attributes, between 2 and 21 decision classes and missing value rates between 0.0032 and 0.43. We rescale numerical attributes to $[0, 1]$, before applying polar encoding or imputation. In the latter case, we then also apply one-hot encoding to categorical attributes.

TABLE II
REAL-LIFE DATASETS WITH MISSING VALUES (ADAPTED FROM [9]).

Dataset	Records	Attributes	Missing rate	Source
adult	48 842	13	0.010	[65]
agaricus-lepiota	8124	22	0.014	[66]
aps-failure	76 000	170	0.083	[67]
arrhythmia	443	279	0.0032	[68]
bands	540	34	0.054	[69]
ckd	400	24	0.11	[70]
crx	690	15	0.0065	[71]
dress-sales	500	12	0.19	
exasens	399	7	0.43	[72]
hcc	165	49	0.10	[73]
heart-disease	1611	14	0.17	[74]
hepatitis	155	19	0.057	[75]
horse-colic	368	20	0.26	[76]
mammographic-masses	961	4	0.042	[77]
mi	1700	111	0.085	[78]
nomao	34 465	118	0.38	[79]
primary-tumor	330	17	0.039	[27]
secom	1567	590	0.045	[80]
soybean	683	35	0.098	[81]
thyroid0387	9172	23	0.069	[82]

We evaluate classification performance using the area under the receiver operator curve (AUROC) [84]. For each dataset, we perform five-fold stratified cross-validation, repeat this five times for different random divisions of the data, and take the mean of the resulting 25 AUROC scores. To establish whether the performance of polar encoding vis-à-vis imputation generalises to other (similar) datasets, we test for significance using one-sided Wilcoxon signed-ranks tests [85]. A p-value below 0.5 indicates that polar encoding performed better, while a p-value above 0.5 indicates that it performed worse.

For all classifiers we use the implementations provided by the Python library *scikit-learn* [24], except for FRNN, where we use our own implementation in *fuzzy-rough-learn* [86]. For our main experiment, we use default hyperparameter values, with three exceptions informed by the findings in [9]: with CART we perform cost complexity pruning ($\alpha = 0.01$), with ERT we set the number of trees to 1000, and with GBM we apply early-stopping.

We also perform a follow-up experiment in which we compare polar encoding against mean/mode imputation with missing-indicator for the same set of classifiers but with hyperparameter optimisation. For NN, NN-D and FRNN, we optimise k for all values in the range $[1, 50]$. For SVM, we optimise C and γ by randomly drawing 10 pairs of values from the exponential distribution $\frac{1}{\beta}e^{-\frac{1}{\beta}x}$, with, respectively, scale $\beta = 100$ and scale $\beta = \frac{1}{10}$. For the decision tree classifiers, we optimise the number of features that are considered at each split, as well as the minimum number of records required to continue splitting nodes, by randomly drawing 10 pairs of values from the interval $[0, 1]$, interpreted as share of the total number of features or records. For NN, NN-D and FRNN, we apply efficient leave-one-out validation, whereas for the

TABLE III
p-VALUES, POLAR ENCODING VS OTHER MISSING VALUE APPROACHES.

Distance	Classifier	Alternative			
		MICE	MIDAS	Mean/mode imputation with missing-indicators	
		Hyperparameter values			
		Default	Default	Default	Optimised
Boscovich	NN	0.011	0.024	0.18	0.074
	NN-D	0.011	0.068	0.19	0.16
	FRNN	0.0088	0.049	0.0024	0.0019
Euclidean	NN	0.0098	0.0070	0.14	0.19
	NN-D	0.018	0.024	0.15	0.086
	FRNN	0.0056	0.0056	0.0040	0.0021
	SVM-G	0.0027	0.0035	0.018	0.039
—	CART	0.13	0.085	0.031	0.058
	RF	0.012	0.23	0.40	0.57
	ERT	0.0063	0.20	0.14	0.23
	ABT	0.0045	0.054	0.054	0.77
	GBM	0.0017	0.099	0.61	0.50

other classifiers we apply stratified (nested) five-fold cross-validation,[6] selecting the hyperparameter values that result in the highest (mean) validation AUROC.

B. Results

Table III lists the p-values obtained from comparing the performance of polar encoding against the performance of MICE, MIDAS and mean/mode imputation with missing-indicators, in terms of the mean AUROC for each classifier and each dataset.[7]

The first thing to note is that with default hyperparameter values and for our selection of datasets, polar encoding generally increases classification performance, except for RF and GBM, where it leads to approximately the same performance as mean/mode imputation with missing-indicators. On the whole, the p-values for Euclidean distance are not higher than the p-values for Boscovich distance, which indicates that the relative advantage of polar encoding is not less with Euclidean distance. In Subsection III-B, we noted that the Euclidean variant of polar encoding introduces a slight distortion to the distances between non-missing values, but this does not appear to be harmful for classification performance.

Not all of the p-values are significant, which may be due to the small sample size (20 datasets). Overall, the advantage of polar encoding over pure imputation is more pronounced than the advantage of polar encoding over mean/mode imputation with missing-indicators, which agrees with our previous finding that missing-indicators increase performance because they preserve missingness-information. Nevertheless, it appears that the greater conservatism of polar encoding gives classifiers even more opportunity to learn from missing values. If we perform a clustered Wilcoxon signed-rank test [87] on the scores obtained for all datasets and all classifiers, clustered by dataset, we find that polar encoding performs significantly better than the alternative approaches, regardless of whether we use Boscovich or Euclidean distance (Table IV).

[6]For datasets with classes that contain only four records in the training set, we have applied four-fold cross-validation instead.

[7]The mean AUROC scores are provided as supplementary material.

TABLE IV
p-VALUES, POLAR ENCODING VS OTHER MISSING VALUE APPROACHES.

Distance used with NN, NN-D and FRNN	Alternative			
	MICE	MIDAS	Mean/mode imputation with missing-indicators	
	Hyperparameter values			
	Default	Default	Default	Optimised
Boscovich	0.0012	0.011	0.011	0.047
Euclidean	0.0012	0.0043	0.011	0.044

Table III also contains the p-values from our follow-up experiment, testing the performance of polar encoding against mean/mode imputation with missing-indicators in the context of hyperparameter optimisation. The p-values are essentially similar to the p-values obtained with default hyperparameter optimisation, except for ABT, where mean/mode imputation with missing-indicators now has a slight advantage. Across all classifiers, polar encoding still performs better (Table IV).

VII. Conclusion

In this paper we have presented polar encoding, a novel method to represent missing values of categorical and $[0, 1]$-valued attributes. We have argued that in the context of classification, it presents a good baseline approach for missing values because it is modular, conservative, simple and performant.

In particular, polar encoding is more conservative than the current baseline, mean/mode imputation with missing-indicators, because it does not just preserve the information from missing values, but also does not pre-suppose their contribution to the classification problem, as it avoids imputation altogether. For distance-based algorithms, it ensures that missing values are equidistant from all non-missing values. For decision tree algorithms, it allows missing values to be grouped on either side of each split. This latter behaviour corresponds to the existing MIA approach, with the crucial difference that polar encoding can be combined with all sorts of classification algorithms, which do not have to be adapted for this purpose.

We have provided further justification for polar encoding by showing that it can be viewed as a fuzzification of one-hot encoding, the standard approach for representing categorical attributes numerically. We did this by formalising the concept of barycentric attributes, which can be seen as both a fuzzification of categorical attributes and a generalisation of $[0, 1]$-valued attributes. Because one-hot encoding is slightly redundant, using one more dimension than strictly necessary, it allows us to represent missing values as zero vectors, symbolising the absence of information.

Having previously shown that missing-indicators improve classification performance on real-life datasets, in the present paper we conducted an experiment to test whether polar encoding works even better. We found that in the context of classification, polar encoding generally outperforms two sophisticated

imputation algorithms, MICE and MIDAS. Polar encoding also performs better than mean/mode imputation with missing indicators, although this difference is less pronounced, and mean/mode imputation may have a slight advantage with ABT if hyperparameter optimisation is applied.

In the future, we would like to extend polar encoding to numerical attributes that are scaled differently. For example, when an attribute is scaled by its standard deviation, polar encoding could be adapted to ensure that missing values are equidistant from all 'typical' non-missing values, namely those contained within one standard deviation of the mean.

Acknowledgments

The research reported in this paper was conducted with the financial support of the Odysseus programme of the Research Foundation – Flanders (FWO).

References

[1] C. Garcia, D. Leite, and I. Škrjanc, "Incremental missing-data imputation for evolving fuzzy granular prediction," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 10, pp. 2348–2362, 2020.

[2] W. Zhang, Z. Deng, T. Zhang, K.-S. Choi, J. Wang, and S. Wang, "Incomplete multi-view fuzzy inference system with missing view imputation and cooperative learning," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 8, pp. 3038–3051, 2022.

[3] D. Li, H. Zhang, T. Li, A. Bouras, X. Yu, and T. Wang, "Hybrid missing value imputation algorithms using fuzzy c-means and vaguely quantified rough set," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 5, pp. 1396–1408, 2022.

[4] D. B. Rubin, "Multiple imputations in sample surveys — a phenomenological Bayesian approach to nonresponse," in *Proceedings of the Survey Research Methods Section of the American Statistical Association*. American Statistical Association Alexandria, VA, USA, 1978, pp. 20–34.

[5] S. Van Buuren and K. Oudshoorn, "Flexible multivariate imputation by MICE," TNO Prevention and Health, Leiden, Tech. Rep. PG/VGZ/99.054, 1999.

[6] R. Lall and T. Robinson, "The MIDAS touch: Accurate and scalable missing-data imputation with deep learning," *Polit Anal*, vol. 30, no. 2, pp. 179–196, 2022.

[7] J. Cohen, "Multiple regression as a general data-analytic system," *Psychol Bull*, vol. 70, no. 6, pp. 426–443, 1968.

[8] M. P. Jones, "Indicator and stratification methods for missing explanatory variables in multiple linear regression," *J Am Stat Assoc*, vol. 91, no. 433, pp. 222–230, 1996.

[9] O. U. Lenz, D. Peralta, and C. Cornelis, "No imputation without representation," *arXiv preprint arXiv:2206.14254*, 2022. [Online]. Available: https://arxiv.org/abs/2206.14254

[10] B. E. Twala, M. Jones, and D. J. Hand, "Good methods for coping with missing data in decision trees," *Pattern Recognit Lett*, vol. 29, no. 7, pp. 950–956, 2008.

[11] J. Josse, N. Prost, E. Scornet, and G. Varoquaux, "On the consistency of supervised learning with missing values," *arXiv preprint arXiv:1902.06931*, 2020. [Online]. Available: https://arxiv.org/abs/1902.06931

[12] D. B. Suits, "Use of dummy variables in regression equations," *J Am Stat Assoc*, vol. 52, no. 280, pp. 548–551, 1957.

[13] A. Perez-Lebel, G. Varoquaux, M. Le Morvan, J. Josse, and J.-B. Poline, "Benchmarking missing-values approaches for predictive models on health databases," *GigaScience*, vol. 11, no. 1, giac013, 2022.

[14] R. J. Boscovich, "De litteraria expeditione per pontificiam ditionem," *De bononiensi scientarium et artium instituto atque academia commentarii*, vol. 4, pp. 353–396 (opuscula), 1757.

[15] ——, "De recentissimis graduum dimensionibus, et figura, ac magnitudine terræ inde derivanda," in *Philosophiæ recentioris,* B. Stay. Rome: Nicolaus et Marcus Palearini, 1760, vol. 2, pp. 406–426.

[16] I. Todhunter, *A History of the Mathematical Theories of Attraction and the Figure of the Earth, from the Time of Newton to that of Laplace*. London: Macmillan, 1873, vol. 1, ch. 14. Boscovich and Stay, pp. 331–332.

[17] C. Eisenhart, "Boscovich and the combination of observations," in *Roger Joseph Boscovich, S.J., F.R.S., 1711–1787: Studies of his Life and Work on the 250th Anniversary of his Birth*, L. L. Whyte, Ed. London: George Allen & Unwin, 1961, ch. 9, pp. 200–212.

[18] S. M. Stigler, *The History of Statistics: The Measurement of Uncertainty before 1900*. Cambridge, Massachusetts: The Belknap Press of Harvard University Press, 1986, ch. 1. Least Squares and the Combination of Observations, pp. 46–47.

[19] J. Dai, "Rough set approach to incomplete numerical data," *Inf Sci*, vol. 241, pp. 43–57, 2013.

[20] O. U. Lenz, D. Peralta, and C. Cornelis, "Adapting fuzzy rough sets for classification with missing values," in *IJCRS 2021: Proceedings of the International Joint Conference on Rough Sets*, ser. Lecture Notes in Artificial Intelligence, vol. 12872. Springer, 2021, pp. 192–200.

[21] R. Jensen and Q. Shen, "Interval-valued fuzzy-rough feature selection in datasets with missing values," in *FUZZ-IEEE 2009: Proceedings of the 18th IEEE International Conference on Fuzzy Systems*. IEEE, 2009, pp. 610–615.

[22] D. Shelupsky, "A generalization of the trigonometric functions," *Am Math Mon*, vol. 66, no. 10, pp. 879–884, 1959.

[23] P. Lindqvist and J. Peetre, "p-arclength of the q-circle," Lund University, Centre for Mathematical Sciences, Tech. Rep. Preprint 2000:21 LUNFMA-5014-2000, 2000.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *J Mach Learn Res*, vol. 12, no. 85, pp. 2825–2830, 2011.

[25] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *NIPS 2017: Proceedings of the Thirty-first Conference on Neural Information Processing Systems*, ser. Advances in neural information processing systems, vol. 30. NIPS Foundation, 2017, pp. 3146–3154.

[26] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[27] B. Cestnik, I. Kononenko, and I. Bratko, "ASSISTANT 86: A knowledge-elicitation tool for sophisticated users," in *EWSL 87: Proceedings of the 2nd European Working Session on Learning*. Sigma Press, 1987, pp. 31–45.

[28] J. L. Schafer, *Analysis of Incomplete Multivariate Data*, ser. Monographs on Statistics and Applied Probability. London: Chapman & Hall, 1997, vol. 72.

[29] B. Twala, "An empirical comparison of techniques for handling incomplete data using decision trees," *Appl Artif Intell*, vol. 23, no. 5, pp. 373–405, 2009.

[30] A. Kapelner and J. Bleich, "Prediction with missing data via Bayesian additive regression trees," *Can J Stat*, vol. 43, no. 2, pp. 224–239, 2015.

[31] J. R. Quinlan, "Induction of decision trees," *Mach Learn*, vol. 1, no. 1, pp. 81–106, 1986.

[32] CIA World Factbook, "GDP — composition, by sector of origin," 2022. [Online]. Available: https://www.cia.gov/the-world-factbook/field/gdp-composition-by-sector-of-origin/

[33] A. F. Möbius, *Der barycentrische Calcul: ein Hülfsmittel zur analytischen Behandlung der Geometrie*. Leipzig: Verlag von Johann Ambrosius Barth, 1827.

[34] R. E. Allardice, "The barycentric calculus of Möbius," *Proc Edinb Math Soc*, vol. 10, pp. 2–21, 1891.

[35] C. B. Boyer, *History of Analytic Geometry*. New York: Scripta Mathematica, 1956, ch. 9. The Golden Age, pp. 242–243.

[36] C. Huang, D. R. Rice, and J. H. Steffen, "MAGRATHEA: an open-source spherical symmetric planet interior structure code," *Mon Not R Astron Soc*, vol. 513, no. 4, pp. 5256–5269, 2022.

[37] M. G. MacDonald, L. Feil, T. Quinn, and D. Rice, "Confirming the 3:2 resonance chain of K2-138," *Astron J*, vol. 163, no. 4, 162, p. 162, 2022.

[38] J. Haldemann, V. Ksoll, D. Walter, Y. Alibert, R. S. Klessen, W. Benz, U. Koethe, L. Ardizzone, and C. Rother, "Exoplanet characterization using conditional invertible neural networks," *arXiv preprint arXiv:2202.00027*, 2022. [Online]. Available: https://arxiv.org/abs/2202.00027

[39] F. Wang, J. Yu, Z. Liu, M. Kong, and Y. Wu, "Study on offshore seabed sediment classification based on particle size parameters using XGBoost algorithm," *Comput Geosci*, vol. 149, no. 104713, 2021.

[40] S. Stemplinger, S. Prévost, T. Zemb, D. Horinek, and J.-F. Dufrêche, "Theory of ternary fluids under centrifugal fields," *J Phys Chem B*, vol. 125, no. 43, pp. 12 054–12 062, 2021.

[41] M. Tönsmann, D. T. Ewald, P. Scharfer, and W. Schabel, "Surface tension of binary and ternary polymer solutions: Experimental data of poly(vinyl acetate), poly(vinyl alcohol) and polyethylene glycol solutions and mixing rule evaluation over the entire concentration range," *Surf Interface*, vol. 26, no. 101352, 2021.

[42] W.-C. Chen, J. N. Schmidt, D. Yan, Y. K. Vohra, and C.-C. Chen, "Machine learning and evolutionary prediction of superhard bcn compounds," *npj Comput Mater*, vol. 7, no. 114, 2021.

[43] A. M. Nolan, E. D. Wachsman, and Y. Mo, "Computation-guided discovery of coating materials to stabilize the interface between lithium garnet solid electrolyte and high-energy cathodes for all-solid-state lithium batteries," *Energy Storage Mater*, vol. 41, pp. 571–580, 2021.

[44] M. Kim, J.-K. Choi, and S. K. Baek, "Win-stay-lose-shift as a self-confirming equilibrium in the iterated prisoner's dilemma," *Proc R Soc B*, vol. 288, no. 1953, 20211021, 2021.

[45] F. Molter, A. W. Thomas, S. A. Huettel, H. R. Heekeren, and P. N. Mohr, "Gaze-dependent evidence accumulation predicts multi-alternative risky choice behaviour," *PLoS Comput Biol*, vol. 18, no. 7, e1010283, 2022.

[46] R. Zhao and K. Mao, "Fuzzy bag-of-words model for document representation," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 794–804, 2018.

[47] Z.-P. Tian, R.-X. Nie, J.-Q. Wang, and R.-Y. Long, "Adaptive consensus-based model for heterogeneous large-scale group decision-making: Detecting and managing noncooperative behaviors," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 8, pp. 2209–2223, 2021.

[48] J. W. Sangma, Y. Rani, V. Pal, N. Kumar, and R. Kushwaha, "FHC-NDS: Fuzzy hierarchical clustering of multiple nominal data streams," *IEEE Trans. Fuzzy Syst.*, forthcoming.

[49] E. H. Ruspini, "A new approach to clustering," *Inf Control*, vol. 15, no. 1, pp. 22–32, 1969.

[50] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J Cybern*, vol. 3, no. 3, pp. 32–57, 1974.

[51] J. C. Bezdek and J. D. Harris, "Fuzzy partitions and relations; an axiomatic basis for clustering," *Fuzzy Sets Syst*, vol. 1, no. 2, pp. 111–127, 1978.

[52] S. V. Wilson, "miceforest: Fast, memory efficient imputation with LightGBM," 2020. [Online]. Available: https://github.com/AnotherSamWilson/miceforest

[53] R. Lall and T. Robinson, "Efficient multiple imputation for diverse data in python and r: Midaspy and rmidas," *J Stat Softw*, In press.

[54] E. Fix and J. Hodges, Jr, "Discriminatory analysis — nonparametric discrimination: Consistency properties," USAF School of Aviation Medicine, Randolph Field, Texas, Tech. Rep. 21-49-004, 1951. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA800276

[55] S. A. Dudani, "The distance-weighted k-nearest-neighbor rule," *IEEE Trans. Syst., Man, Cybern.*, vol. 6, no. 4, pp. 325–327, 1976.

[56] R. Jensen and C. Cornelis, "A new approach to fuzzy-rough nearest neighbour classification," in *RSCTC 2008: Proceedigns of the 6th International Conference on Rough Sets and Current Trends in Computing*, ser. Lecture Notes in Artificial Intelligence, vol. 5306. Springer, 2008, pp. 310–319.

[57] C. Cornelis, N. Verbiest, and R. Jensen, "Ordered weighted average based fuzzy rough sets," in *RSKT 2010: Proceedings of the 5th International Conference on Rough Set and Knowledge Technology*, ser. Lecture Notes in Artificial Intelligence, vol. 6401. Springer, 2010, pp. 78–85.

[58] C. Cortes and V. Vapnik, "Support-vector networks," *Mach Learn*, vol. 20, no. 3, pp. 273–297, 1995.

[59] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, ser. The Wadsworth statistics/probability series. Monterey, California: Wadsworth, 1984.

[60] L. Breiman, "Random forests," *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001.

[61] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach Learn*, vol. 63, no. 1, pp. 3–42, 2006.

[62] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *EuroCOLT '95: Proceedings of the Second European Conference on Computational Learning Theory*, ser. Lecture Notes in Computer Science, vol. 904. Springer, 1995, pp. 23–37.

[63] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class AdaBoost," *Stat Its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[64] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann Stat*, vol. 29, no. 5, pp. 1189–1232, 2001.

[65] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid," in *KDD-96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996, pp. 202–207.

[66] J. C. Schlimmer, "Concept acquisition through representational adjustment," Ph.D. dissertation, University of California, Irvine, 1987.

[67] C. Ferreira Costa and M. A. Nascimento, "IDA 2016 industrial challenge: Using machine learning for predicting failures," in *IDA 2016: Proceedings of the 15th International Symposium on Intelligent Data Analysis*, ser. Lecture Notes in Computer Science, vol. 9897. Springer, 2016, pp. 381–386.

[68] H. A. Güvenir, B. Acar, G. Demiröz, and A. Çekin, "A supervised machine learning algorithm for arrhythmia analysis," in *Proceedings of the 24th Annual Meeting of Computers in Cardiology*, ser. Computers in Cardiology, vol. 24. IEEE, 1997, pp. 433–436.

[69] B. Evans and D. Fisher, "Overcoming process delays with decision tree induction," *IEEE Expert*, vol. 9, no. 1, pp. 60–66, 1994.

[70] L. J. Rubini and P. Eswaran, "Generating comparative analysis of early stage prediction of chronic kidney disease," *Int J Mod Eng Res*, vol. 5, no. 7, pp. 49–55, 2015.

[71] J. R. Quinlan, "Simplifying decision trees," *Int J Man-Mach Stud*, vol. 27, no. 3, pp. 221–234, 1987.

[72] P. Soltani Zarrin, N. Röckendorf, and C. Wenger, "In-vitro classification of saliva samples of COPD patients and healthy controls using machine learning tools," *IEEE Access*, vol. 8, pp. 168 053–168 060, 2020.

[73] M. S. Santos, P. H. Abreu, P. J. García-Laencina, A. Simão, and A. Carvalho, "A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients," *J Biomed Inform*, vol. 58, pp. 49–59, 2015.

[74] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.-J. Schmid, S. Sandhu, K. H. Guppy, S. Lee, and V. Froelicher, "International application of a new probability algorithm for the diagnosis of coronary artery disease," *Am J Cardiol*, vol. 64, no. 5, pp. 304–310, 1989.

[75] B. Efron and G. Gong, "Statistical theory and the computer," in *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface*. Springer, 1981, pp. 3–7.

[76] M. McLeish and M. Cecile, "Enhancing medical expert systems with knowledge obtained from statistical data," *Ann Math Artif Intell*, vol. 2, no. 1–4, pp. 261–276, 1990.

[77] M. Elter, R. Schulz-Wendtland, and T. Wittenberg, "The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process," *Med Phys*, vol. 34, no. 11, pp. 4164–4172, 2007.

[78] S. E. Golovenkin, J. Bac, A. Chervov, E. M. Mirkes, Y. V. Orlova, E. Barillot, A. N. Gorban, and A. Zinovyev, "Trajectories, bifurcations, and pseudo-time in large clinical datasets: applications to myocardial infarction and diabetes data," *GigaScience*, vol. 9, no. 11, giaa128, 2020.

[79] L. Candillier and V. Lemaire, "Design and analysis of the nomao challenge: Active learning in the real-world," in *ECML-PKDD 2012: Active Learning in Real-world Applications Workshop*, 2012.

[80] M. McCann, Y. Li, L. Maguire, and A. Johnston, "Causality challenge: benchmarking relevant signal components for effective monitoring and process control," in *NIPS 2008: Proceedings of Workshop on Causality*, ser. Proceedings of Machine Learning Research, vol. 6. JMLR Workshop and Conference Proceedings, 2008, pp. 277–288.

[81] R. S. Michalski and R. L. Chilausky, "Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis," *Int J Policy Anal Inf Syst*, vol. 4, no. 2, pp. 125–161, 1980.

[82] J. R. Quinlan, P. J. Compton, K. A. Horn, and L. Lazarus, "Inductive knowledge acquisition: a case study," in *Proceedings of the Second Australian Conference on Applications of Expert Systems*. Turing Institute Press, 1986, pp. 157–173.

[83] D. Dua and C. Graff, "UCI machine learning repository," 2019. [Online]. Available: http://archive.ics.uci.edu/ml

[84] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Mach Learn*, vol. 45, no. 2, pp. 171–186, 2001.

[85] F. Wilcoxon, "Individual comparisons by ranking methods," *Biom Bull*, vol. 1, no. 6, pp. 80–83, 1945.

[86] O. U. Lenz, C. Cornelis, and D. Peralta, "fuzzy-rough-learn 0.2: a Python library for fuzzy rough set algorithms and one-class classification," in *FUZZ-IEEE 2022: Proceedings of the IEEE International Conference on Fuzzy Systems*. IEEE, 2022.

[87] B. Rosner, R. J. Glynn, and M.-L. T. Lee, "The Wilcoxon signed rank test for paired comparisons of clustered data," *Biometrics*, vol. 62, no. 1, pp. 185–192, 2006.

APPENDIX
FULL RESULTS

We list here the mean AUROC across five-fold cross-validation and five random states for each classifier, each dataset, and each missing value approach, for distance-based classifiers and decision tree classifiers with default hyperparameter values (Tables V and VI, respectively) and with optimised hyperparameter values (Tables VII and VIII, respectively). **MMI-I**: mean/mode imputation with missing indicators. **PE**: polar encoding.

TABLE V: Distance-based classifiers, default hyperparameter values. **Bold**: highest value per distance measure.

Classifier	Dataset	Boscovich distance				Euclidean distance			
		MICE	MIDAS	MMI-I	PE	MICE	MIDAS	MMI-I	PE
NN	adult	0.846	0.846	0.846	**0.846**	0.846	0.846	0.846	**0.846**
	agaricus-lepiota	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	aps-failure	0.908	**0.910**	0.910	0.909	0.895	0.897	0.902	**0.904**
	arrhythmia	0.756	**0.758**	0.757	0.757	0.720	0.706	**0.733**	0.723
	bands	0.778	0.777	0.800	**0.824**	0.770	0.766	0.794	**0.810**
	ckd	0.994	0.995	0.996	**0.999**	0.997	**0.999**	0.994	0.997
	crx	0.909	0.909	0.910	**0.912**	0.909	0.911	0.910	**0.911**
	dress-sales	0.535	0.545	**0.560**	0.552	0.536	0.545	**0.560**	0.548
	exasens	0.697	0.712	0.717	**0.719**	0.702	**0.719**	0.713	0.717
	hcc	0.718	0.703	**0.751**	0.717	0.696	0.687	**0.733**	0.699
	heart-disease	0.833	0.836	0.833	**0.841**	0.824	0.825	0.827	**0.832**
	hepatitis	**0.834**	0.826	0.815	0.818	**0.839**	0.827	0.815	0.828
	horse-colic	0.735	**0.754**	0.723	0.728	**0.735**	0.732	0.727	0.730
	mammographic-masses	0.820	0.822	**0.831**	0.830	0.820	0.821	**0.830**	0.830
	mi	0.551	0.559	**0.591**	0.575	0.553	0.557	**0.584**	0.583
	nomao	0.960	0.970	0.980	**0.982**	0.954	0.964	0.978	**0.980**
	primary-tumor	0.692	0.687	**0.719**	0.687	0.695	0.687	**0.718**	0.697
	secom	**0.623**	0.622	0.590	0.617	0.591	**0.598**	0.522	0.548
	soybean	0.976	0.988	0.988	**0.992**	0.973	0.986	0.987	**0.990**
	thyroid0387	0.797	0.819	0.833	**0.835**	0.785	0.804	0.827	**0.830**
NN-D	adult	0.825	0.825	0.826	**0.828**	0.827	0.826	**0.827**	0.827
	agaricus-lepiota	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	aps-failure	0.909	0.911	**0.911**	0.910	0.896	0.897	0.903	**0.905**
	arrhythmia	0.759	**0.760**	0.759	0.760	0.722	0.708	**0.735**	0.726
	bands	0.803	0.800	0.824	**0.851**	0.784	0.780	0.808	**0.825**
	ckd	0.994	0.995	0.997	**0.999**	0.997	**0.999**	0.996	0.997
	crx	0.905	0.905	0.906	**0.909**	0.906	0.906	0.906	**0.909**
	dress-sales	0.534	0.543	**0.564**	0.552	0.535	0.539	**0.563**	0.548
	exasens	0.694	**0.702**	0.636	0.637	0.699	**0.708**	0.632	0.634
	hcc	0.738	0.723	**0.762**	0.738	0.713	0.705	**0.744**	0.720
	heart-disease	0.835	0.839	0.837	**0.843**	0.828	0.829	0.832	**0.837**
	hepatitis	**0.828**	0.825	0.823	0.821	**0.833**	0.829	0.819	0.827
	horse-colic	0.750	**0.770**	0.747	0.754	0.745	**0.750**	0.745	0.749
	mammographic-masses	0.791	0.801	0.808	**0.808**	0.791	0.801	0.808	**0.808**
	mi	0.552	0.559	**0.592**	0.577	0.554	0.558	**0.586**	0.585
	nomao	0.961	0.971	0.981	**0.983**	0.954	0.965	0.979	**0.981**
	primary-tumor	0.676	0.681	**0.703**	0.679	0.676	0.680	**0.704**	0.688
	secom	**0.630**	0.628	0.594	0.624	0.594	**0.599**	0.526	0.549
	soybean	0.976	0.988	0.988	**0.992**	0.974	0.986	0.987	**0.990**
	thyroid0387	0.798	0.820	0.835	**0.837**	0.787	0.805	0.829	**0.832**
FRNN	adult	0.872	0.871	0.872	**0.878**	0.862	0.862	0.863	**0.867**
	agaricus-lepiota	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	aps-failure	**0.980**	0.965	0.943	0.952	**0.975**	0.964	0.962	0.968

Continued on next page

TABLE V: Distance-based classifiers, default hyperparameter values. **Bold**: highest value per distance measure.

Classifier	Dataset	Boscovich distance				Euclidean distance			
		MICE	MIDAS	MMI-I	PE	MICE	MIDAS	MMI-I	PE
	arrhythmia	0.882	0.882	**0.889**	0.887	0.856	0.855	0.868	**0.875**
	bands	0.812	0.810	0.832	**0.852**	0.796	0.795	0.819	**0.833**
	ckd	1.000	1.000	0.999	**1.000**	**1.000**	1.000	0.998	1.000
	crx	0.914	0.914	0.918	**0.921**	0.914	0.915	0.918	**0.920**
	dress-sales	0.562	0.583	**0.592**	0.577	0.558	0.566	**0.586**	0.572
	exasens	0.727	0.740	0.719	**0.745**	0.727	0.740	0.736	**0.749**
	hcc	0.775	0.778	0.784	**0.792**	0.777	0.771	0.769	**0.780**
	heart-disease	0.858	0.858	0.858	**0.863**	0.849	0.846	0.848	**0.854**
	hepatitis	**0.887**	0.884	0.882	0.884	**0.883**	0.878	0.879	0.880
	horse-colic	0.759	**0.794**	0.760	0.772	0.761	**0.792**	0.766	0.772
	mammographic-masses	0.800	0.813	0.816	**0.838**	0.806	0.816	0.824	**0.837**
	mi	0.668	0.680	0.674	**0.687**	0.658	0.662	0.670	**0.678**
	nomao	0.976	0.983	0.986	**0.990**	0.971	0.978	0.987	**0.989**
	primary-tumor	**0.794**	0.787	0.794	0.790	0.788	0.780	**0.791**	0.784
	secom	**0.693**	0.689	0.642	0.673	0.629	**0.630**	0.596	0.609
	soybean	0.992	0.997	**0.997**	0.997	0.991	0.996	0.997	**0.997**
	thyroid0387	0.871	0.872	0.888	**0.908**	0.875	0.875	0.891	**0.902**
SVM-G	adult					0.892	0.892	0.893	**0.900**
	agaricus-lepiota					**1.000**	**1.000**	**1.000**	**1.000**
	aps-failure					0.957	0.957	0.942	**0.974**
	arrhythmia					0.866	0.869	0.872	**0.878**
	bands					0.810	0.808	0.833	**0.843**
	ckd					**1.000**	1.000	0.999	1.000
	crx					0.918	0.920	0.920	**0.922**
	dress-sales					0.623	0.623	**0.632**	0.614
	exasens					0.760	0.767	0.768	**0.774**
	hcc					0.784	0.785	**0.800**	0.789
	heart-disease					0.860	0.862	0.861	**0.869**
	hepatitis					**0.861**	0.853	0.857	0.858
	horse-colic					0.759	0.786	0.776	**0.788**
	mammographic-masses					0.831	0.837	**0.845**	0.835
	mi					0.642	0.638	0.648	**0.655**
	nomao					0.981	0.986	0.990	**0.991**
	primary-tumor					**0.792**	0.782	0.781	0.789
	secom					0.702	**0.703**	0.678	0.696
	soybean					0.997	0.999	0.999	**0.999**
	thyroid0387					0.877	0.874	0.894	**0.922**

TABLE VI: Decision tree classifiers, default hyperparameter values. **Bold**: highest value.

Classifier	Dataset	MICE	MIDAS	MMI-I	PE
ABT	adult	0.915	0.915	0.915	**0.915**
	agaricus-lepiota	**1.000**	**1.000**	**1.000**	**1.000**
	aps-failure	0.986	0.986	**0.987**	0.987
	arrhythmia	0.633	0.634	0.634	**0.634**
	bands	0.805	0.806	0.806	**0.813**
	ckd	0.998	0.999	1.000	**1.000**
	crx	0.904	0.906	0.906	**0.908**
	dress-sales	0.585	**0.596**	0.581	0.583

Continued on next page

TABLE VI: Decision tree classifiers, default hyperparameter values. **Bold**: highest value.

Classifier	Dataset	MICE	MIDAS	MMI-I	PE
	exasens	0.700	0.712	0.720	**0.722**
	hcc	0.713	0.722	0.725	**0.729**
	heart-disease	0.849	0.854	**0.860**	0.859
	hepatitis	0.798	0.777	0.807	**0.809**
	horse-colic	0.740	**0.763**	0.756	0.756
	mammographic-masses	0.852	0.855	**0.857**	0.856
	mi	0.576	0.568	0.572	**0.582**
	nomao	0.979	0.985	0.987	**0.987**
	primary-tumor	**0.662**	0.662	0.660	0.648
	secom	0.673	**0.673**	0.668	0.663
	soybean	0.744	0.851	0.870	**0.892**
	thyroid0387	0.656	0.684	0.685	**0.685**
CART	adult	**0.844**	**0.844**	**0.844**	**0.844**
	agaricus-lepiota	0.991	**0.993**	0.992	0.992
	aps-failure	0.860	**0.867**	0.859	0.859
	arrhythmia	0.746	0.747	**0.748**	0.745
	bands	0.742	0.731	0.759	**0.768**
	ckd	0.980	**0.980**	0.975	0.977
	crx	**0.900**	0.897	0.896	0.897
	dress-sales	0.572	0.566	0.570	**0.574**
	exasens	0.721	0.715	0.732	**0.743**
	hcc	**0.617**	0.587	0.588	0.590
	heart-disease	0.778	**0.780**	0.777	0.774
	hepatitis	0.645	**0.661**	0.578	0.596
	horse-colic	0.702	0.699	**0.723**	0.718
	mammographic-masses	0.815	0.816	**0.823**	0.822
	mi	0.535	0.581	0.592	**0.607**
	nomao	0.917	**0.935**	0.916	0.916
	primary-tumor	0.707	0.700	0.707	**0.739**
	secom	**0.500**	**0.500**	**0.500**	**0.500**
	soybean	0.959	0.984	0.991	**0.993**
	thyroid0387	0.877	0.884	**0.909**	0.908
ERT	adult	0.847	0.847	0.847	**0.856**
	agaricus-lepiota	**1.000**	**1.000**	**1.000**	**1.000**
	aps-failure	0.991	0.991	0.991	**0.991**
	arrhythmia	0.897	0.899	**0.899**	0.899
	bands	0.879	0.888	0.890	**0.904**
	ckd	1.000	1.000	**1.000**	**1.000**
	crx	0.915	0.916	0.914	**0.916**
	dress-sales	0.572	0.579	**0.602**	0.575
	exasens	0.716	**0.740**	0.626	0.627
	hcc	0.778	0.791	**0.808**	0.803
	heart-disease	0.857	**0.864**	0.862	0.861
	hepatitis	0.874	**0.879**	0.873	0.857
	horse-colic	0.776	**0.803**	0.782	0.796
	mammographic-masses	0.789	0.793	0.802	**0.805**
	mi	0.680	0.690	0.695	**0.709**
	nomao	0.985	0.990	0.994	**0.994**
	primary-tumor	0.698	**0.727**	0.714	0.712
	secom	0.745	0.740	0.746	**0.747**
	soybean	0.997	0.999	0.999	**0.999**

Continued on next page

TABLE VI: Decision tree classifiers, default hyperparameter values. **Bold**: highest value.

Classifier	Dataset	MICE	MIDAS	MMI-I	PE
	thyroid0387	0.975	0.983	0.988	**0.991**
GBM	adult	0.927	0.927	**0.927**	0.927
	agaricus-lepiota	**1.000**	**1.000**	**1.000**	**1.000**
	aps-failure	0.988	0.987	**0.988**	0.987
	arrhythmia	0.848	0.848	**0.852**	0.851
	bands	0.846	0.859	0.857	**0.859**
	ckd	0.994	0.991	0.996	**0.996**
	crx	**0.934**	0.934	0.933	0.933
	dress-sales	0.592	**0.621**	0.614	0.608
	exasens	0.733	0.752	0.757	**0.757**
	hcc	0.734	**0.761**	0.745	0.751
	heart-disease	0.859	0.863	**0.871**	0.869
	hepatitis	**0.817**	0.804	0.810	0.798
	horse-colic	0.769	0.768	**0.784**	0.783
	mammographic-masses	0.850	0.852	**0.859**	0.856
	mi	0.642	0.636	0.637	**0.646**
	nomao	0.989	0.992	**0.994**	0.994
	primary-tumor	0.761	0.755	**0.767**	0.767
	secom	0.675	**0.694**	0.679	0.680
	soybean	0.997	0.998	0.999	**0.999**
	thyroid0387	0.885	0.899	0.918	**0.928**
RF	adult	0.890	0.890	0.890	**0.897**
	agaricus-lepiota	**1.000**	**1.000**	**1.000**	**1.000**
	aps-failure	0.988	0.988	0.989	**0.989**
	arrhythmia	0.885	**0.890**	0.887	0.885
	bands	0.885	0.894	**0.896**	0.894
	ckd	1.000	0.999	1.000	**1.000**
	crx	0.930	0.931	0.931	**0.931**
	dress-sales	0.585	**0.609**	0.606	0.576
	exasens	0.734	**0.753**	0.702	0.707
	hcc	0.794	0.806	**0.816**	0.813
	heart-disease	0.857	0.859	**0.864**	0.858
	hepatitis	0.880	**0.886**	0.886	0.875
	horse-colic	0.783	0.786	0.792	**0.798**
	mammographic-masses	0.812	0.812	0.822	**0.825**
	mi	0.664	0.679	0.686	**0.696**
	nomao	0.987	0.990	0.994	**0.994**
	primary-tumor	0.749	0.757	**0.758**	0.756
	secom	0.709	0.715	0.709	**0.722**
	soybean	0.997	0.999	0.999	**0.999**
	thyroid0387	0.978	0.987	**0.994**	0.992

TABLE VII: Distance-based classifiers, optimised hyperparameter values. **Bold**: highest value per distance measure.

Classifier	Dataset	Boscovich distance		Euclidean distance	
		MMI-I	PE	MMI-I	PE
NN	adult	0.886	**0.887**	0.886	**0.887**
	agaricus-lepiota	**1.000**	**1.000**	**1.000**	**1.000**
	aps-failure	**0.969**	0.969	0.966	**0.966**
	arrhythmia	**0.797**	0.796	0.785	**0.786**

Continued on next page

TABLE VII: Distance-based classifiers, optimised hyperparameter values. **Bold**: highest value per distance measure.

Classifier	Dataset	Boscovich distance		Euclidean distance	
		MMI-I	PE	MMI-I	PE
	bands	0.798	**0.814**	0.794	**0.803**
	ckd	0.996	**0.999**	0.994	**0.996**
	crx	**0.913**	0.913	0.909	**0.909**
	dress-sales	**0.612**	0.607	**0.614**	0.607
	exasens	0.728	**0.729**	**0.735**	0.729
	hcc	0.758	**0.761**	0.717	**0.744**
	heart-disease	**0.859**	0.858	0.847	**0.847**
	hepatitis	**0.860**	0.858	**0.862**	0.861
	horse-colic	0.762	**0.771**	0.757	**0.765**
	mammographic-masses	0.838	**0.839**	**0.841**	0.839
	mi	0.608	**0.626**	0.610	**0.632**
	nomao	0.985	**0.988**	0.983	**0.986**
	primary-tumor	**0.762**	0.736	**0.764**	0.749
	secom	0.637	**0.688**	**0.598**	0.594
	soybean	0.995	**0.996**	0.994	**0.995**
	thyroid0387	0.913	**0.915**	0.912	**0.913**
NN-D	adult	0.881	**0.885**	0.874	**0.878**
	agaricus-lepiota	**1.000**	**1.000**	**1.000**	**1.000**
	aps-failure	**0.971**	0.971	0.968	**0.969**
	arrhythmia	0.814	**0.815**	**0.807**	0.800
	bands	0.825	**0.856**	0.815	**0.828**
	ckd	0.997	**0.999**	0.996	**0.998**
	crx	0.913	**0.917**	0.911	**0.915**
	dress-sales	**0.604**	0.597	**0.601**	0.597
	exasens	**0.731**	0.729	**0.727**	0.726
	hcc	**0.776**	0.764	0.754	**0.771**
	heart-disease	**0.863**	0.863	0.852	**0.853**
	hepatitis	**0.864**	0.860	**0.862**	0.856
	horse-colic	0.771	**0.771**	0.773	**0.777**
	mammographic-masses	0.828	**0.830**	**0.825**	0.824
	mi	0.610	**0.627**	0.611	**0.633**
	nomao	0.988	**0.990**	0.987	**0.989**
	primary-tumor	**0.779**	0.745	**0.779**	0.761
	secom	0.662	**0.714**	0.589	**0.621**
	soybean	0.998	**0.998**	0.997	**0.998**
	thyroid0387	0.914	**0.921**	0.911	**0.913**
FRNN	adult	0.875	**0.881**	0.864	**0.869**
	agaricus-lepiota	**1.000**	**1.000**	**1.000**	**1.000**
	aps-failure	0.944	**0.952**	0.962	**0.968**
	arrhythmia	**0.879**	0.875	0.858	**0.868**
	bands	0.833	**0.870**	0.816	**0.832**
	ckd	0.998	**1.000**	0.997	**0.999**
	crx	0.917	**0.920**	0.919	**0.919**
	dress-sales	**0.613**	0.601	**0.608**	0.596
	exasens	0.741	**0.741**	0.739	**0.744**
	hcc	0.781	**0.781**	0.771	**0.775**
	heart-disease	0.856	**0.862**	0.846	**0.852**
	hepatitis	0.876	**0.881**	0.871	**0.878**
	horse-colic	0.757	**0.768**	0.766	**0.777**
	mammographic-masses	0.845	**0.853**	0.843	**0.847**

Continued on next page

TABLE VII: Distance-based classifiers, optimised hyperparameter values. **Bold**: highest value per distance measure.

Classifier	Dataset	Boscovich distance		Euclidean distance	
		MMI-I	PE	MMI-I	PE
	mi	0.648	**0.665**	0.648	**0.653**
	nomao	0.987	**0.991**	0.987	**0.990**
	primary-tumor	0.777	**0.778**	**0.784**	0.780
	secom	0.619	**0.667**	0.572	**0.585**
	soybean	**0.997**	0.997	0.996	**0.997**
	thyroid0387	0.898	**0.914**	0.894	**0.905**
SVM-G	adult			0.900	**0.906**
	agaricus-lepiota			**1.000**	**1.000**
	aps-failure			0.970	**0.972**
	arrhythmia			0.877	**0.882**
	bands			0.846	**0.872**
	ckd			**1.000**	**1.000**
	crx			0.916	**0.919**
	dress-sales			**0.634**	0.621
	exasens			0.779	**0.781**
	hcc			**0.793**	0.778
	heart-disease			0.864	**0.873**
	hepatitis			0.834	**0.841**
	horse-colic			0.787	**0.788**
	mammographic-masses			0.854	**0.855**
	mi			0.660	**0.669**
	nomao			0.992	**0.992**
	primary-tumor			**0.792**	0.786
	secom			0.672	**0.680**
	soybean			**0.999**	0.999
	thyroid0387			0.955	**0.961**

TABLE VIII: Decision tree classifiers, optimised hyperparameter values. **Bold**: highest value.

Classifier	Dataset	MMI-I	PE
ABT	adult	0.915	**0.915**
	agaricus-lepiota	**1.000**	**1.000**
	aps-failure	0.986	**0.987**
	arrhythmia	**0.741**	0.724
	bands	**0.812**	0.807
	ckd	1.000	**1.000**
	crx	**0.909**	0.909
	dress-sales	**0.576**	0.572
	exasens	**0.721**	0.716
	hcc	0.722	**0.735**
	heart-disease	**0.863**	0.862
	hepatitis	**0.803**	0.790
	horse-colic	0.753	**0.767**
	mammographic-masses	0.855	**0.856**
	mi	**0.600**	0.586
	nomao	0.987	**0.987**
	primary-tumor	**0.769**	0.757
	secom	**0.661**	0.657
	soybean	0.972	**0.990**

Continued on next page

TABLE VIII: Decision tree classifiers, optimised hyperparameter values. **Bold**: highest value.

Classifier	Dataset	MMI-I	PE
	thyroid0387	0.878	**0.888**
CART	adult	0.881	**0.882**
	agaricus-lepiota	0.996	**0.999**
	aps-failure	0.974	**0.975**
	arrhythmia	0.723	**0.727**
	bands	0.750	**0.752**
	ckd	**0.986**	0.981
	crx	**0.909**	0.902
	dress-sales	**0.596**	0.592
	exasens	0.733	**0.734**
	hcc	0.622	**0.627**
	heart-disease	0.798	**0.800**
	hepatitis	0.755	**0.767**
	horse-colic	0.709	**0.718**
	mammographic-masses	**0.838**	0.835
	mi	**0.632**	0.631
	nomao	**0.965**	0.964
	primary-tumor	**0.713**	0.712
	secom	0.642	**0.658**
	soybean	0.963	**0.964**
	thyroid0387	0.918	**0.920**
ERT	adult	0.893	**0.898**
	agaricus-lepiota	**1.000**	1.000
	aps-failure	**0.985**	0.985
	arrhythmia	**0.872**	0.867
	bands	0.848	**0.857**
	ckd	1.000	**1.000**
	crx	0.919	**0.922**
	dress-sales	**0.631**	0.618
	exasens	0.765	**0.765**
	hcc	0.772	**0.774**
	heart-disease	**0.860**	0.859
	hepatitis	0.864	**0.868**
	horse-colic	**0.793**	0.790
	mammographic-masses	0.858	**0.859**
	mi	0.680	**0.683**
	nomao	0.980	**0.981**
	primary-tumor	**0.791**	0.787
	secom	**0.729**	0.728
	soybean	0.998	**0.998**
	thyroid0387	0.967	**0.970**
GBM	adult	0.918	**0.919**
	agaricus-lepiota	**1.000**	1.000
	aps-failure	0.987	**0.987**
	arrhythmia	**0.905**	0.904
	bands	0.862	**0.862**
	ckd	1.000	**1.000**
	crx	0.937	**0.937**
	dress-sales	**0.609**	0.602
	exasens	0.783	**0.784**
	hcc	**0.794**	0.790

Continued on next page

TABLE VIII: Decision tree classifiers, optimised hyperparameter values. **Bold**: highest value.

Classifier	Dataset	MMI-I	PE
	heart-disease	0.874	**0.874**
	hepatitis	0.863	**0.873**
	horse-colic	0.788	**0.789**
	mammographic-masses	**0.866**	0.862
	mi	0.704	**0.709**
	nomao	0.990	**0.990**
	primary-tumor	**0.794**	0.792
	secom	**0.718**	0.713
	soybean	0.999	**0.999**
	thyroid0387	**0.955**	0.951
RF	adult	0.904	**0.907**
	agaricus-lepiota	1.000	**1.000**
	aps-failure	**0.985**	0.985
	arrhythmia	**0.864**	0.857
	bands	**0.860**	0.860
	ckd	0.999	**1.000**
	crx	0.930	**0.931**
	dress-sales	**0.640**	0.627
	exasens	**0.763**	0.757
	hcc	**0.818**	0.804
	heart-disease	**0.862**	0.859
	hepatitis	**0.872**	0.872
	horse-colic	0.781	**0.785**
	mammographic-masses	0.860	**0.862**
	mi	0.669	**0.678**
	nomao	**0.979**	0.979
	primary-tumor	0.787	**0.791**
	secom	0.709	**0.712**
	soybean	0.998	**0.998**
	thyroid0387	**0.975**	0.975