TabEE: Tabular Embeddings Explanations

RONI COPUL, Tel Aviv University, Israel
NAVE FROST, eBay Research, Israel
TOVA MILO, Tel Aviv University, Israel
KATHY RAZMADZE, Tel Aviv University, Israel

Tabular embedding methods have become increasingly popular due to their effectiveness in improving the results of various tasks, including classic databases tasks and machine learning predictions. However, most current methods treat these embedding models as "black boxes" making it difficult to understand the insights captured by the models. Our research proposes a novel approach to interpret these models, aiming to provide local and global explanations for the original data and detect potential flaws in the embedding models. The proposed solution is appropriate for every tabular embedding algorithm, as it fits the black box view of the embedding model. Furthermore, we propose methods for comparing different embedding models, which can help identify data biases that might impact the models' credibility without the user's knowledge. Our approach is evaluated on multiple datasets and multiple embeddings, demonstrating that our proposed explanations provide valuable insights into the behavior of tabular embedding methods. By making these models more transparent, we believe our research will contribute to the development of more effective and reliable embedding methods for a wide range of applications.

CCS Concepts: • **Information systems** → *Data management systems*; • **Computing methodologies** → **Knowledge representation and reasoning**; **Machine learning**.

Additional Key Words and Phrases: Explainable AI, Interpretability, Tabular Representation Learning, Embedding Models

ACM Reference Format:
Roni Copul, Nave Frost, Tova Milo, and Kathy Razmadze. 2024. TabEE: Tabular Embeddings Explanations. *Proc. ACM Manag. Data* 2, 1 (SIGMOD), Article 72 (February 2024), 26 pages. https://doi.org/10.1145/3639329

1 INTRODUCTION

Tables are fundamental data structures in a wide range of domains, from scientific research to business applications, and they serve as a critical means for storing and organizing data. In recent years, researchers have proposed a novel approach to analyzing tables and dataframes using tabular embedding techniques, containing richer representations of the data. Tabular embedding aims to represent the structural and semantic information of dataframes in a low-dimensional vector space, facilitating its effective analysis and processing. If the embedding is row-wise (column-wise), then each row (column) of a dataframe is represented as a vector. In whole tabular embeddings, each table is represented by a single vector.

Authors' addresses: Roni Copul, Tel Aviv University, Israel, Tel Aviv, ronycopul@mail.tau.ac.il; Nave Frost, eBay Research, Israel, Netanya, nafrost@ebay.com; Tova Milo, Tel Aviv University, Israel, Tel Aviv, milo@cs.tau.ac.il; Kathy Razmadze, Tel Aviv University, Israel, Tel Aviv, kathyr@mail.tau.ac.il.

ACM 2836-6573/2024/2-ART72
https://doi.org/10.1145/3639329

Variety of tabular embedding models. Tabular embedding techniques have advanced vastly in recent years, providing several options for generating vector representations of tabular data. These techniques include static and contextualized word embedding methods, graph-based approaches, and neural network-based methods. Static embedding methods (i.e., [10]) approaches based on word2vec (i.e., [31]) generate embeddings by analyzing the co-occurrence statistics of words in a corpus. Graph-based approaches, such as EmbDI [11], model dataframes as graphs and learn embeddings by capturing the interactions between dataframe columns and rows. More recent techniques, including TabNet [5], TransTab [46], VIME [52] and SubTab [41], leverage neural network architectures like Transformers (i.e., [43]) and Auto-Encoders (i.e., [23]) to generate embeddings that consider the surrounding context of each word in a sentence. Each of these approaches has its own strengths and weaknesses, making it necessary to choose the most appropriate technique depending on the specific task and data being analyzed.

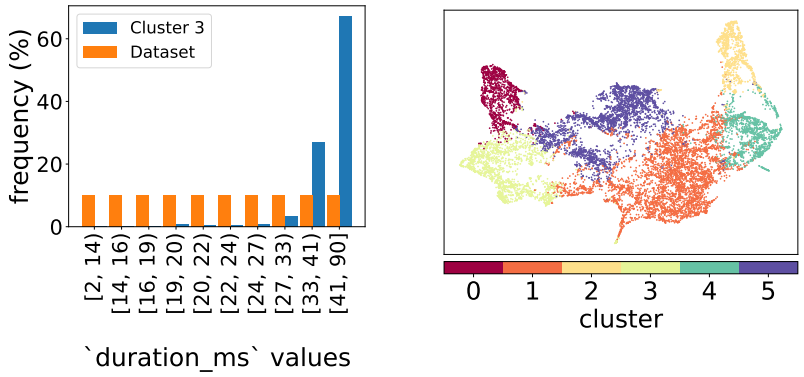
Importance of tabular embedding models. Tabular embedding has been extensively used in numerous applications, including table retrieval, entity linking, table completion, and classification tasks. In table retrieval, embeddings are used to retrieve tables based on their semantic similarity to a given query, enabling efficient and accurate table search. For entity linking, embeddings can be used to link entities in tables to external knowledge graphs or ontologies, facilitating tasks like named entity recognition and disambiguation. In table completion, embeddings can be used to predict missing values in a table, allowing for more complete and accurate data analysis. Finally, in classification tasks, embeddings can be used to classify either the entire dataframe or individual rows, enabling more effective machine learning models for different tasks. The ability to generate tabular embeddings has thus opened up a range of possibilities for the analysis and processing of tabular data, and continues to be an active area of research.

Motivation. The field of tabular embedding techniques encompasses numerous works, but there is no principled method for choosing an appropriate embedding for a specific task. Even the simplest tabular embedding model produces vector representations for rows or columns in the original dataframe that are derived from the embedding space, creating a disconnection between the original dataframe and the embedding space. Columns and rows cannot be directly associated with specific axes in the embedding space, and the dimensionality of the vectors may differ from the original attributes.

As tabular embedding models become more complex, understanding and analyzing their inner workings becomes increasingly challenging. Nevertheless, such efforts are of paramount importance for exploring learned patterns, comparing different embedding models, performing an efficient hyperparameter tuning, conducting error analysis for the improvement of different downstream tasks, identifying bias, and investigating the transformed data space.

Remarkably, despite the significance of explainability, to the best of our knowledge no attempts have been made to develop black-box interpretations specifically tailored to tabular embedding models. Addressing this research gap would contribute to the field's advancement by providing interpretable and understandable insights into the behavior of these embedding models, unlocking their full potential for various applications.

Inadequacy of existing approaches. One possible approach to addressing the interpretability challenge in the context of tabular embedding models is to leverage existing and commonly employed methods for black box ML explanation. Various techniques have been developed to explain the decision-making process of ML models, particularly in the field of classification, as will be elaborated further in Section 2. Recent advancements in interpretable algorithms have also emerged, catering to NLP tasks (e.g., question answering [50]) and tabular data analysis (e.g., [5]).



(a) Single cohort's explanation (b) Cohorts visualization

Fig. 1. Part of the created explanation for a trained tabular embedding model

However, these interpretations are intricately tailored to specific embedding and classification models, making them less adaptable to other algorithms. Such approaches lack the capacity to capture intrinsic patterns of the embedding model independently and do not provide users the flexibility to explore the internal workings of the trained embedding model autonomously, apart from the ML classification model and its associated downstream tasks, which can exhibit variation.

Proposed solution. Our Tabular Embeddings Explanations system (TabEE) uses an explanation method that operates on a dataset and its corresponding embedding model. Our method aims to identify cohorts of data points that are embedded closely together, allowing us to uncover common patterns within these groups. Specifically, we focus on detecting cohorts in the embedding space with, among other parameters, the most significant distribution shifts across one of the original attributes. This is done using a novel metric that takes into consideration different parameters of a "good" explanation from previous works defined as interestingness, sufficiency and diversity as detailed in Section 4. This enables us to explain the patterns captured by the tabular embedding model.

Our explanation technique is inspired by the concept of measuring changes in distribution between two data groups, similar to the approach presented in [17]. However, instead of solely measuring distribution shifts, our method assesses the focus that the embedding model has given to a group of attributes and their representation within the embedding model. By analyzing these shifts and attribute representations, we are able to provide meaningful explanations that shed light on the patterns captured by the embedding model. The following example illustrates the helpfulness of such explanations:

Example 1.1. Suppose we have a music dataset sourced from [2] and a trained tabular embedding model that produces a vector representation for each row in the dataset. Our objective is to gain insights into the patterns captured by the embedding model and their potential implications for various tasks, such as genre classification. Figure 1 showcases a portion of the explanation generated by TabEE for the given embedding model and dataset. The cohorts (shown in Figure 1b) are computed using our designated metric and are selected based on maximizing the overall score of the provided explanations for the embedding model. Figure 1a, illustrates an explanation for one of the cohorts. Concretely, the orange bars represent the distribution of the attribute *duration_ms* in the original dataset, demonstrating an even distribution across its range. Conversely, the blue

bars represent the distribution of the same attribute in the embedding space. Notably, for this specific cohort of points in the embedding space (Cluster 3), their corresponding original points are predominantly drawn from the higher range of the *duration_ms* attribute. This insight provides valuable information about how the embedding model represents the attribute *duration_ms* for this particular cohort. It highlights a distinct pattern captured by the embedding model, suggesting that songs with longer duration are grouped together within this cohort. Such findings enable us to better understand the underlying patterns and characteristics captured by the embedding model in relation to the music dataset.

Specifically, domain experts can observe that certain music genres often exhibit characteristic duration ranges. For instance, trance genre songs tend to be longer compared to Hip-Hop or pop songs. The explanation highlights that, in the embedding space, songs with similar durations are clustered together, diverging from the original data space (pre-embedding), as illustrated by the orange bars in Figure 1a. This snippet of the explanation suggests the potential utility of this embedding for predicting song genres. Additionally, assessing the correlation between the *duration_ms* and *genre* attributes can serve as a valuable measure of the embedding's compatibility with the downstream genre classification task.

As mentioned, given the limited prior work in this area, we provide several definitions and metrics to address these concerns:

(1) How can we measure that an explanation has been successfully applied to an embedding (i.e., the model has genuinely learned a given pattern)?
(2) How can we measure the clarity of an explanation to the user (despite it being what the model has "understood")?
(3) How can we measure the non-triviality of an explanation, or its level of interest (i.e., whether it can be applied for data analysis and provide insights about the data beyond the model itself)?
(4) Can we compare two embedding models based on the explanations provided for them? Is there a basis for mutual comparison?

These metrics are crucial for evaluating the efficacy and interpretability of tabular embedding models, and we will explore them in further detail in our proposed methodology.

Main approach. As outlined in [15], the quality of an explanation or interpretation of a model is determined by the extent to which a human observer can comprehend the rationale behind a given decision or prediction made by the model. Thus, we propose a novel approach for generating a global explanation for a tabular embedding model through a set of cohorts of examples, which are represented as clusters or a dense sets of points in the embedding space. Building on the insights of [17], we measure the difference in the layout of the vectors representing the rows relative to their original values, in each one of the cohorts. To maximize the effectiveness of this approach, we have also defined a methodology for utilizing the explainability of embedding models.

Contributions. This work makes several contributions to the field of explainable embeddings, including:

(1) A formal definition of what constitutes an explanation of a tabular embedding model.
(2) Defining a metric to measure the quality of the explanation of the embedding model. This includes adapting the definition of faithfulness of a local explanation from [16] to the case of global explanations, and the diversity metric from [53] to include this types of explanations, and adapted interestingness measure from [17] to fit better to various attributes types.
(3) Proposing an algorithm for generating explanations that capture the main patterns learned by an embedding model.
(4) Implementation of the system as an open-source Python library.

(5) Providing experimental results that demonstrate the value of the explanations generated by our system within a reasonable time.
(6) Conducting user studies that demonstrate the effectiveness of our system in enabling analysts to derive useful insights.

Organization. This paper is organized as follows. In Sec. 2 we present the preliminaries of tabular embeddings, explaining AI models and other related work. Sec. 3 presents our algorithm for creating explanations for tabular embedding. Sec. 4 defines the metrics used to measure the quality of the created explanation. In Sec. 6, we describe our experimental study. We conclude in Sec. 7.

2 RELATED WORK

In our review of previous related work, we first examine tabular embedding methods. We then explore general methods for explainable AI, which provide valuable concepts applicable to our approach. Additionally, we investigate approaches focused on explanations, summarization, and exploration specifically tailored to tabular data.

Tabular embedding models. Tabular embedding is a powerful technique in database systems that maps relational tables to dense vector representations, captures the semantic relationships between columns and rows, enabling tasks like database retrieval, entity resolution, and knowledge graph completion. Various approaches have been proposed, including graph-based methods (e.g., EmbDI [11]), matrix factorization techniques (e.g., [24]), and deep learning-based models (e.g., Tabnet [5], SubTab [41], VIME [52], TransTab [46]). Recent research has demonstrated the superiority of tabular embedding models over traditional methods in tasks like database retrieval and entity resolution [11]. However, the field currently lacks sufficient explanations for the insights derived from these models, which is concerning given the potential biases and pitfalls that flawed models can introduce. Existing studies have not adequately addressed this lack of explanation or provided satisfactory justifications for the results obtained from tabular embedding models.

Explaining Machine Learning models. Explainability is a crucial aspect of ML systems, especially in domains where decisions have significant impacts. Explanations can be generated either locally, providing insights into specific predictions, or globally, explaining patterns learned by the model. Approaches for generating explanations include intrinsic generation using interpretable models like decision trees [25], and extrinsic generation treating the model as a black box, utilizing methods such as LIME [8] or SHAP [28]. These techniques assign feature contributions based on inclusion or exclusion values, leveraging concepts from cooperative game theory.

Example 2.1. The study in [20] provides an illustrative example for a computer vision embedding model, often characterized by its complexity. In this domain, each image is represented by a vector (embedding representation) utilized for various tasks, such as image classification. In the example, the embedding space of this image domain is organized into clusters based on the vectors' representation of the images within that domain. As expected in a well-designed representation, images with similar content or recognizable features exhibit more similar vectors, resulting in their proximity within the embedding space. For instance, the red cluster contains nature-oriented images like deer, parks, leaves, and forests, positioned closely to each other compared to images from different categories, such as fire trucks or ships. However, translating this explanation to tabular data is not straightforward, given that tabular data involves multiple attributes with diverse types and values. Merely illustrating rows that are close together lacks meaning without a deeper understanding of which attributes and values contributed to this proximity in the embedding space.

Providing explanations to patterns in the data. The FEDEx approach [17] focuses on identifying the most interesting rows or sets of rows within a dataframe based on their contribution to the overall interestingness of different columns. While our work draws inspiration from this concept, it differs significantly. We concentrate on explaining the latent space by identifying optimal separation into cohorts, uncovering meaningful patterns and insights. Our approach considers the collective insights from the entire set of explanations, going beyond metrics per cluster and providing a comprehensive understanding of the patterns captured by the embedding space.

Data summarization. Data summarization aims to derive compressed forms or high-level summaries of data, and includes dimension reduction [14], data sketches [13], and aggregation-focused AQP methods [21, 54]. While approaches like [12] generate summaries using association rules, they are computationally expensive for interactive exploration. In contrast, TabEE efficiently generates explanations specifically for tabular embeddings.

Data exploration and discovery tools. Various tools have been developed to facilitate data exploration for users with varying levels of expertise. For non-programmer users, works such as [6, 7, 38, 39] propose simplified exploration interfaces that enable data manipulation without requiring explicit query writing. Moreover, visualization recommender systems like Voyager [51], DeepEye [29], and LUX [26] automatically generate suitable visualizations for input datasets. Other solutions like TopK-Insight [40] and QuickInsights [18] provide aggregative insights and patterns from data. While these approaches produce useful visualizations and insights, they focus on specific patterns mined from the original attributes. TabEE complements these systems by providing explanations for tabular embedding models rather than patterns in the original data.

3 EXPLAINING TABULAR EMBEDDING MODELS

This section introduces a novel algorithm for generating explanations tailored to tabular embedding models. Our approach focuses on capturing the learned patterns within the embedding model and providing valuable insights into their potential impact on subsequent tasks. The algorithm, presented in Algorithm 1, incorporates a clustering process to guide the generation of explanations. For each cohort, Algorithm 2 generates cohort explanations that pertain to the relevant rows in the original dataframe represented by the cluster in the embedding space. To ensure the effectiveness of our algorithm and evaluate the quality of the explanations, we propose a metric in Section 4. This metric serves two purposes: determining the optimal number of cohorts and assessing the quality of the explanations produced by our algorithm. By quantifying the effectiveness of our approach, this metric offers a reliable measure to evaluate the algorithm's performance. In Section 6, we present the experimental results, showcasing the utility of our proposed metric and evaluating the algorithm's performance.

3.1 Formal Definition

We consider a dataset represented as a dataframe d, consisting of a list of rows over a schema $\mathcal{A}(d)$. Each row $r \in d$ represents a tuple, where $A \in \mathcal{A}(d)$ specifies a dataframe attribute. We denote the dataframe column associated with attribute A as $d[A]$, which represents the multiset of all rows' values associated with attribute A in the dataframe d. Given an embedding model M, we denote the representation created for the dataset as $M(d)$, also referred to as $\hat{d}$. The embedding representation $\hat{d}$ has a different schema than the original dataset d, such that $\mathcal{A}(\hat{d}) \neq \mathcal{A}(d)$. For each datapoint in the original dataset r, there is a corresponding embedding representation $\hat{r}$ in the embedding space. We define a cohort of size n as a set of rows $C = \{r_1, r_2, ..., r_n \in d\}$ that are grouped closely together in the embedding space $\hat{C} = \{\hat{r_1}, \hat{r_2}, ..., \hat{r_n} \in \hat{d}\}$.

Cohort explanation. Given a cohort $C = \{r_1, \ldots, r_n\} \in d$, we aim to measure the significance of the change in the distribution of values of a specific attribute A between the original dataset and the cohort C. The deviation, with respect to attribute $A \in \mathcal{A}(d)$, is defined as the difference between the values in $d[A]$ and $C[A]$, which represent the column associated with attribute A in the original dataframe d and the cohort C, respectively. To quantify this deviation, we utilize the Jensen-Shannon (JS) distance [27], a statistical test that measures the difference between two distributions, including categorical values.

First, we define the column probability distribution $Pr(d[A])$ based on the relative frequency of its values (i.e., for each value $v \in d[A]$, $Pr(v)$ is the probability to choose v at random). We then calculate the deviation in the embedding space in this attribute: $D(A, C, d) := JS(Pr[d[A], Pr(C[A]]))$.

Suppose attributes $[A_1, \ldots, A_m]$ exhibit the m largest deviations in the embedding space for cohort C. We treat the cohort explanation candidates as m distributions, each of attributes A_j (for $j \in [1, m]$) in cohort $C[A_j]$ alongside the original distribution $d[A_j]$, denoted as $Exp(C; A_j)$ (as illustrated in Figure 1). For each cohort we can define all its explanation candidates by $Exp(C_i) = \{Exp(C_i; A_{i,j})\}_{j=1}^{m}$. Given all cohort explanations candidates $Exp(C_1), \ldots, Exp(C_k)$ for k cohorts, we evaluate their quality using the metric defined in Section 4. Based on the metric score, we then determine the best (metric maximizing) combination of explanations across all evaluated numbers of cohorts k, and across all combinations of candidates (with a single candidate per cohort).

3.2 Creating Cohorts' Explanations

As the created vectors by the embedding model are from the embedding space, and there is no direct connection that could be drawn from the original dataframe, we have to connect these two spaces, and measure the changes in them. Inspired by [17], we provide an explanation for a cluster of vectors (cohort), based on their contribution to a pre-defined measure. The Algorithm 1 takes as input an embedding model M and the original dataset D, and produces a global explanation $E = [e_1, e_2, ..., e_k]$ comprising k cohort explanations.

The algorithm begins by setting the maximum number of cohorts K_{max} to 10 (As will be shown in Section 6 as a reasonable choice). It also initializes two empty lists, a list for explanations (denoted as $Explain$), and a list for the score (denoted as $Score$). In both of the lists, each entry in index k corresponds to the explanation or score (correspondingly) that will be created when the number of cohorts is k. It then iterates over values of k from 2 to K_{max} and performs the following steps for each k value:

(1) It applies a clustering method (default: K-means), denoted as $clusteringMethod(M, k)$, to generate the cohorts (a comparison between different clustering methods presented at Section 6).
(2) For each cohort i (ranging from 1 to k), it generates a cohort explanation, that at this stage is m candidates for explaining this cohort. Those candidates are denoted as $[e_{i,1}, \ldots, e_{i,m}]$ using the $ClustTopExp$ (elaborated in Algorithm 2).
(3) Then, we evaluate the quality of the candidate explanations created for each cohort together as a single explanation for the embedding model by $EvalExp$ function, denoted as $EvalExp([e_{i,1}, \ldots, e_{i,m}], cohorts)$ (elaborated in Section 4). Briefly, for each of the cohorts we pick the best candidate explanation such that their combined score will be the highest. This score is appended to the list $Score$ and the explanation to $Explain$.

After generating the explanations for all k options, the algorithm selects the value k^* that maximizes the score by identifying the argument that maximizes $Score(k)$. Finally, the algorithm returns the explanation $Explain(k^*)$ associated with the optimal k^*.

Algorithm 1 TabEE Algorithm - Explaining Tabular Embeddings

Input: Embedding model M, Original Dataset D, number of candidates per cohort m, maximal number of cohorts K_{max}
Output: $E = [e_1, e_2, ..., e_k]$ global explanation consisting of k cohort explanations

```
Score, Explain = []
for k ∈ [2, K_max] do
    cohorts ← clusteringMethod(M, k)
    for i ∈ [1, k] do
        cohort = cohorts(i)
        [e_{i,1}, ..., e_{i,m}] = cohortTopExp(D, cohort, m)
    end for
    Score[k] = max_{j_1,...,j_k} EvalExp([e_{1,j_1}, ..., e_{k,j_k}], cohorts)
    Exp[k] = arg max_{j_1,...,j_k} EvalExp([e_{1,j_1}, ..., e_{k,j_k}], cohorts)
end for
k* = arg max (Score)
return Explain[k*]
```

3.3 Explaining a Single Cohort

In this section, we present the algorithm responsible for generating explanation candidates for the cohorts discovered by Algorithm 1. Building upon the interestingness metric and notion introduced in previous works [22, 37], and further used in [17], we utilize the concept of attribute drift to identify the attributes that capture the most significant patterns within a cohort. Given the original dataset D and a specific cohort *cohort*, Algorithm 2 generates a list of explanation candidates $[e_{i,1}, \ldots, e_{i,m}]$. The algorithm begins by initializing an empty list called *distances*. For each attribute *attr* in D, the following steps are performed:

- Calculate the JS distance [27] between the attribute value distributions within the cohort and the original dataset.
- Append the calculated distance to the *distances* list.

Finally, the algorithm selects the m attributes with the largest distances, indicating substantial differences in distribution between the cohort and the original dataset. These selected attributes provide insights into the distinct patterns captured by the cohort. The algorithm outputs the comparative distributions of attribute values for both the cohort and the original dataset, as illustrated in Figure 1.

4 METRICS FOR EVALUATION

After developing an algorithm for generating explanations for tabular embedding models, we introduce a novel approach to evaluate their effectiveness. We view a single global explanation of an embedding model as a collection of cohort explanations that capture different patterns within it. Recognizing that a complete explanation should address various aspects of the data, we believe that multiple cohort explanations are required to provide a comprehensive understanding of the entire model. The utility of an explanation algorithm lies in its ability to assess the quality of the generated explanations, allowing for comparison and selection among different methods or candidate explanations. As there is no universally accepted measure for evaluating explanations, we incorporate guidelines from prior research in explainable AI into our framework. Concretely, we consider the following aspects when evaluating the explanations:

Algorithm 2 $cohortTopExp(D, cohort, m)$ - Generating top-m explanations for a cohort of the representation

Input: Original Dataset D, $cohort \subseteq D$, number of candidates m
Output: $[e_{i,1}, \ldots, e_{i,m}]$ m candidates for explanation

```
distances = []
for attr ∈ D.attributes do
    distance = JS_distance(cohort[attr], D[attr])
    distances.append(distance)
end for
top_m_attributes = arg sort_top_m{distances}
return {Distributions(cohort[attr*], D[attr*])
        for attr* ∈ top_m_attributes}
```

- Sufficiency: The extent to which an explanation captures the patterns learned by the embedding model.
- Interestingness: The level of interest elicited by the explanations.
- Diversity: The variation and distinctiveness among all the explanation's cohort components within an embedding.

To comprehensively address these aspects, we propose a metric that combines these dimensions and incorporates adjustable parameters to modify the weighting of different metrics. In Section 6, we validate the effectiveness of our metric through ablation studies, demonstrating its significance in evaluating explanations for tabular embedding models. Furthermore, we conduct a user study in Section 6.4 to establish a correlation between high metric scores and the usefulness of explanations for analysts. This study provides an empirical evidence of the value and practicality of our evaluation metric in assessing the quality of explanations.

4.1 Measuring the Quality of a Single Explanation for a Cohort

We propose a method to measure the alignment between a given explanation and a closely grouped set of points in the embedding space, referred to as a cohort. Each point in the cohort represents an individual row in the original dataset. This measure aims to assess the accuracy of the explanation in capturing the distinct characteristics of the cohort compared to other sets of points in the embedding space (referred to as **Sufficiency**), as well as its significance with respect to the original distribution of the data (referred to as **Interestingness**).

Sufficiency. Our motivation behind this metric is to ensure that a given explanation is relevant only to its associated cluster, capturing the concept of sufficiency. Building on the framework proposed in [16], we define the sufficiency metric as follows: Let x be a data tuple from the original dataset d, with its corresponding embedding vector $\hat{x}$. We denote the cluster assigned to $\hat{x}$ as $C_{\hat{x}}$, and the explanation assigned to the cluster as $Exp(C_{\hat{x}})$. The explanation is represented as a probability distribution over the values of attribute $A \in \mathcal{A}(d)$.

To evaluate the sufficiency of an explanation, we define the relation $R(x, e)$ as the probability that the instance x is drawn from the distribution of the explanation e. This relation quantifies the extent to which the explanation holds for the instance x, taking into account the probability value. This metric is inspired by the sufficiency metric presented in in [16], where the binary relation $A(x, \pi)$ is used to asses whether an explanation π applies to x. In order to extend this definition to probabilistic explanations, we define the extended relation $R(x, e) = \Pr_e[x \sim e]$.

Given an instance x and its explanation π, the original sufficiency metric captures the probability a random sample that the explanation π holds for, is assigned the same prediction as x. Thus, the original sufficiency metric can be approximated empirically by:

$$m^s(x) = \frac{\sum_{x'} \mathbb{1}[f(x') = f(x)] \cdot A(x', \pi)}{\sum_{x'} A(x', \pi)}$$

In order to extend this definition to our setting, we can use the clustering model as the classification model in the formula, and use the extended relation $R(\cdot, \cdot)$:

$$m^s(x) = \frac{\sum_{x'} \mathbb{1}[\hat{x'} \in C_{\hat{x}}] \cdot R(x', Exp(C_{\hat{x}}))}{\sum_{x'} R(x', Exp(C_{\hat{x}}))}$$

Now we want to use a more explicit form of $R(\cdot, \cdot)$:

$$R(x', Exp(C)) = \Pr[x' \sim Exp(C)] = \Pr[C|x'] = \frac{\Pr[x'|C] \cdot \Pr[C]}{\Pr[x']}$$

Plugging this expression in the formula yields:

$$m^s(x) = \frac{\sum_{x'} \mathbb{1}[\hat{x'} \in C_{\hat{x}}] \cdot \sum_{x'} Pr_{x' \sim Exp(C_{\hat{x}})}[x']/\Pr[x']}{\sum_{x'} Pr_{x' \sim Exp(C_{\hat{x}})}[x']/\Pr[x']}$$

To obtain the global sufficiency, we calculate the local sufficiency for each example and average it across all examples.

Interestingness. The interestingness metric aims to assess the significance of an explanation by comparing the distribution of a specific attribute within a cluster to its distribution in the original dataset. Given an explanation e_i for a cluster C_i, we denote A_i as the attribute used to explain the cluster, and $Pr(d[A_i])$ as the distribution of attribute A_i in the original dataset d. The interestingness of the single cluster explanation is defined as the distribution distances between the distribution of $Pr(C_i[A_i])$ in the cluster and its distribution in the original dataset. Formally, we have:

$$interestingness(e_i) = dist(Pr(C_i[A_i]), Pr(d[A_i]))$$

In this formula, $dist(\cdot, \cdot)$ represents the metric used to compare the distributions. We utilize the JS-divergence metric, which can handle both numeric and categorical histograms. The global interestingness of an explanation $E = (e_1, \ldots, e_n)$ is then calculated by averaging the interestingness metrics across all the cohorts:

$$interestingness(E) = \frac{1}{n} \sum_{i=1}^{n} interestingness(e_i) =$$

$$= \frac{1}{n} \sum_{i=1}^{n} dist(Pr(C_i[A_i]), Pr(d[A_i]))$$

This metric provides a measure of how distinct the distribution of the attribute used for explanation is within its corresponding cluster compared to the original distribution of the attribute in the dataset. It captures the level of interestingness of the explanation in terms of its JS-divergence from the overall data distribution.

4.2 Measuring the Quality of Several Explanations as a Whole

Diversity. To assess the overall quality of multiple explanations for different cohorts, we introduce the **Diversity** metric, inspired by [53]. It quantifies the amount of new information gained from each cohort's explanation, by measuring the marginal gain of knowledge provided by each new explanation compared to the previously seen explanations. Given a set of explanations $e_1, \ldots, e_k$, where e_i represents the i-th explanation, we define the diversity as the difference between the knowledge gained from e_{k+1} and the cumulative knowledge gained from $e_1, \ldots, e_k$. If e_{k+1} introduces a new attribute, the diversity is maximal, represented by a value of 1. If it introduces a different distribution for an already known attribute, it contributes the distance between the explanations as a measure of how much of a new knowledge we have gained from the additional explanation. Conversely, if e_{k+1} provides the same explanation as a previous one, the diversity is minimal, represented by a value of 0.

To account for the ordering of explanations, we calculate the average diversity value across all permutations of the orderings, considering only explanations that share the same attribute. This metric enables us to identify explanations that significantly contribute to our understanding of the model, even in scenarios with a large number of explanations. Formally, we first define the following sets:

$$ExpBy(A, E) = \left\{C | \text{A is the explained attribute in } Exp(C) \text{ in } E\right\}$$

Now, using the notation of $perm(S)$ as the set of all ordered permutations of a set S, we can define the diversity metric:

$$diversity(E) = \sum_{A \in \mathcal{A}(d)} \sum_{p \in perm\left(ExpBy(A,E)\right)} \frac{PermDiversity(p, A)}{|ExpBy(A, E)|!}$$

$$\text{where } PermDiversity(p, A) = \sum_{i=1}^{|p|} \min_{j<i} D(A, p[i], p[j])$$

Where for any permutation, we fixate $p[0]$ to be a constant such that $D(p[1], p[0]) := 1$.

4.3 Combining All Metrics Together

The total score combines all the individual metrics (as a weighted sum) to provide an overall assessment of the quality of the explanations. Formally,

$$Score(E) = \alpha \cdot sufficiency(E) + \beta \cdot interestingness(E) +$$

$$\gamma \cdot diversity(E)$$

The default values for the parameters are $\alpha = \beta = \gamma = \frac{1}{3}$, indicating equal weightage given to each metric. These values can be adjusted based on specific requirements or preferences. By combining multiple metrics, the total score provides a comprehensive evaluation that considers various aspects of the explanations, including their sufficiency, interestingness and diversity. This holistic approach enables us to select the most informative and relevant explanations from a set of candidates. In Section 6, we conduct ablation experiments to verify the importance of each metric component.

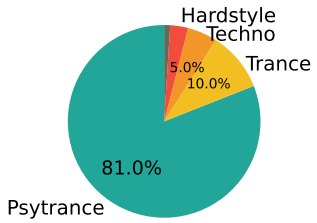


Fig. 2. Visualization of the labels in cluster 3

Remark. Another metric that has been considered is the Silhouette score [35], as a way to evaluate the goodness of fit of the clustering model to the embedding space. However, our ablation experiments resulted in it being redundant with relation to the combination of all other metric components, therefore, omitted from the final score.

5 USE CASES AND GUIDELINES FOR TABULAR EMBEDDING EXPLANATION

Next, we present a range of use cases aimed at showcasing the advantages of TabEE in practical data analysis scenarios. Each of them is accompanied by actionable guidelines on how to effectively integrate TabEE into data exploration workflows to derive valuable insights. Many of these use cases and associated guidelines have been informed by the valuable feedback provided by participants in our user study (Section 6.4). These examples serve as illustrative demonstrations of how our approach can streamline diverse data analysis tasks, making them accessible to individuals with varying levels of expertise. By leveraging TabEE, users can readily identify critical patterns and relationships within their data, ultimately enhancing the efficiency of their analyses. While space constraints prevent us from presenting an exhaustive list of use cases, those featured here underscore the efficacy of our approach in majorly reducing the time required to accomplish these tasks. Moreover, users are encouraged to adapt and expand these guidelines to suit their specific research goals. In Section 6.2, we address less complex use cases, such as gaining an intuitive understanding of the embedding space.

***General guidelines in using* TabEE.** One of the fundamental principles for effectively utilizing our system is to explore the clusters formed within the embedding space, as visually depicted in Figure 1b. This exploration entails a careful examination of the explanations generated by TabEE. The relative sizes of the clusters in comparison to each other, as well as the extent of their separation, can provide valuable insights into the significance of the patterns they represent within the entire dataset. Users are encouraged to delve into the explanations created for each cluster alongside the quality score of the explanation, across various embedding models. For instance, in Figure 1a, which received a high score of 0.85, we observe that within one of the top three clusters, the attribute most prominently represented is *duration_ms*, particularly in its higher range. This understanding serves as a starting point for numerous inquiries that analysts might want to explore. These could include questions regarding the relevance of this attribute to the target variable they seek to predict, the potential presence of bias in this label, and the nature of the information encapsulated within this particular attribute.

Error analysis. In the task of error analysis, our objective is to differentiate between errors made by the downstream model and errors made by the embedding model. Typically, an embedding model is trained on the original data to create a more expressive representation, which is then used to train an ML model for genre classification. During the inference phase on the test data,

we may encounter misclassifications by the model, raising concerns about potential issues in either the embedding representation or the classification model. This analysis helps us identify the areas that require improvement in the machine learning pipeline. For example, if we have a well-trained embedding model that accurately maps each downstream label to its own cluster in the embedding space, but the classification model performs poorly, our focus should be on improving the classification model. Conversely, if we have a flawed embedding model that fails to separate classes clearly, no matter which classification model we use, the performance will be limited, and we should concentrate on improving the embedding model.

For a misclassified example x, which receives a prediction of l_1 instead of l_2, the user should initially identify the cohort to which the example belongs. By closely inspecting the explanation provided for this cohort, the user can gain insights into the dominant attribute A and their associated range $[A_1, A_2]$. After confirming that the example was correctly placed in this cohort, by ensuring that the values in A align with the specified range, the user can explore the labels of other examples within the same cohort. This exploration may reveal that a significant proportion of examples within the cohort are labeled as l_1, leading the model to correlate examples with values falling within the range $[A_1, A_2]$ in A with label l_1 based on the representations of the examples in the embedding space. If achieving precise classification of such examples is imperative, prioritizing improvements to the embedding model becomes essential. Potential measures to address this issue include training a different embedding model, employing oversampling techniques, or considering the removal of attribute A from the embedding process.

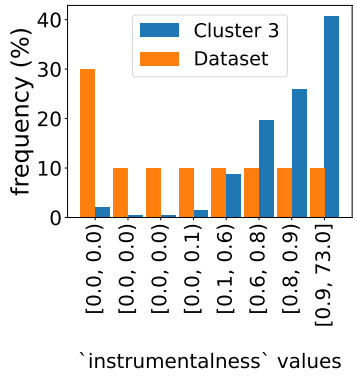
To illustrate this process, consider a genre classification task using the *Spotify* dataset. During error analysis, we select a misclassified song, such as one originally categorized as *Trance* but classified by the model as *Psytrance*. In the embedding space, this song is assigned to cohort c_3, with explanations provided in Figure 1. Upon closer examination of the song's duration, we observe that it falls within the higher range. Further investigation reveals that the majority of songs in c_3 are categorized as *Psytrance* (depicted in Figure 2). Thus, the misclassified song is positioned closely to the points representing *Psytrance* songs in the embedding space, posing a challenge for the classification model to distinguish between the two genres. This scenario suggests that the embedding model encounters difficulty in accurately representing *Trance* songs with longer durations.

Overfit analysis. Evaluating the potential for overfitting in downstream ML models is a common practice in model assessment. Overfitting arises when a model becomes excessively tailored to the training data, resulting in diminished generalization performance when applied to unseen data. Detecting overfitting in downstream models has been addressed through various techniques. However, when dealing with downstream task models trained on data transformed by an embedding model, it becomes challenging to discern whether overfitting originates from the downstream task model or the embedding model. To our knowledge, there have been no specific studies addressing overfitting in tabular embedding models, particularly when it lacks an explicit connection to a downstream task. Given that the embedding model relies solely on the training data, there is potential for overfitting, especially when dealing with complex models and limited data.

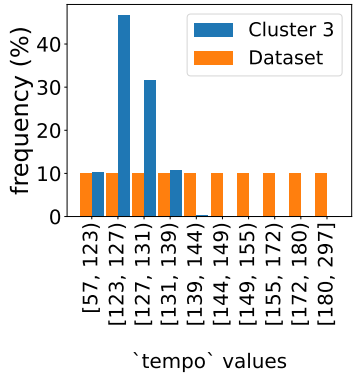
Outlined below are guidelines for initiating an overfit analysis. Firstly, we employ our system to generate explanations separately for the patterns captured by the embedding model in both the training and validation datasets. By comparing these explanations between the two sets, we can assess the robustness of the captured patterns. If an explanation is present in the training data (i.e. a range of certain attribute is highly represented in the embedding's cluster), but does not hold in the validation data (i.e. this pattern was not captured in the embedding representation), it suggests that the pattern may not be genuinely representative of the data, hinting at potential overfitting in

	duration_ms	instrumentalness	tempo	energy	danceability	acousticness
0	384000	0.90	124.99	0.36	0.61	0.70
1	609867	0.87	140.00	0.97	0.60	0.00
2	96062	0.00	139.99	0.58	0.77	0.14
3	304010	0.00	179.81	0.46	0.46	0.00
4	224855	0.00	189.94	0.56	0.47	0.83
5	220112	0.26	154.46	0.59	0.33	0.00

(a) A sub-table with patterns of the embedding model



(b) Row #0 cluster's first explanation.



(c) Row #0 cluster's second explanation

Fig. 3. Explaining embedding models for EDA sub-tables

the embedding model. Conversely, if the explanations remain consistent across the training and validation datasets, it becomes more plausible that the overfitting has occurred in the downstream task model rather than originating from the embedding model.

Explanations for EDA. In the context of Exploratory Data Analysis (EDA), data scientists often scrutinize subsets of extensive datasets to uncover underlying data patterns. Prior research, as exemplified in [34], has employed tabular embedding models to generate informative sub-tables, although these lacked interpretability. Our explanation method can complement these sub-tables by offering insights into the patterns discerned by the tabular embedding model. Specifically, our approach enables users to inspect particular attributes and their corresponding values for each row within a sub-table, facilitating an understanding of the learned patterns. For instance, consider Figure 3a, which represents the sub-table created by the SubTab embedding model for the *Spotify*. In this example, the highlighted cells in the first row elucidate the primary pattern captured in cluster 1 within the embedding space. These patterns are further elucidated in Figures 3b with respect to the *instrumentalness* attribute range, in Figure 3c concerning the *tempo* attribute range, and in Figure 1a in relation to the *duration_ms* attribute ranges. Through the exploration of these explanations, analysts can gain valuable insights into the patterns discovered by the embedding model, providing essential context for data exploration and analysis during EDA.

Evaluation of an embedding model and a comparison between embedding models. As detailed in Section 2, many embedding methods using different approaches and architectures were released, presenting significance over others in performing a variety of tasks. We assert that our

explainability method is highly valuable for comparing different embeddings. To the best of our knowledge, no previous work has introduced a method for directly comparing embedding models, aside from analyzing their performance on downstream tasks. Our explanation method provides a unique approach for evaluating and contrasting different models based on their interpretability.

A productive approach to initiate the analysis on why a specific embedding model might be more suitable for a downstream task is to meticulously examine the distinct attributes and distributions used for generating explanations for the predominant clusters in each embedding. It's noteworthy that, in certain instances, the same attribute may emerge as a crucial factor for multiple clusters, each capturing a distinct range or facet of this attribute. This observation, combined with the significance of these clusters within the embedding space, underscores the importance of this attribute in the embedding representation. Subsequently, users can leverage the insights gained above and compute the correlation between the label attribute and the attributes highlighted as significant in the explanations.

Our experiments have revealed that embedding models may emphasize attributes that are less relevant to a specific task (particularly in cases where embeddings are learned in a self-supervised manner rather than a supervised manner tailored to a specific task). This can result in important task-specific attributes being underrepresented in the embedding space. Such disparities are readily discernible through explanations generated by TabEE, significantly assisting users in making informed decisions concerning model selection and evaluation. Furthermore, we argue that our explanation method is also applicable for parameter tuning and optimization of embedding models. Different parameter settings within the same model can be treated as distinct models, and our method can be used to analyze and optimize their performance (presented at Section 6.4).

6 EXPERIMENTS

When evaluating explanations generated by ML models, the standard approach typically falls into two categories [45]: objective evaluations and human-centered evaluations. Objective evaluations involve using metrics and automated methods to assess explanation quality, while human-centered evaluations collect feedback from humans to evaluate usefulness and interpretability. In our experiments, after describing the experimental setup and system parameters in Section 6.1, we divide our evaluations into three parts:

(1) Objective Evaluation: This part centers on objective metrics and automated approaches. We consider default parameter values, examine the trade-off between running time and explanation quality, assess the alignment of explanations with the underlying embedding model using both synthetic and real datasets, and evaluate the helpfulness of the explanations for downstream tasks. We use SHAP values [28] as an indication of relevant patterns and assess how well the explanations support these tasks. **(2) Baselines Comparison:** Our proposed method lacks a direct baseline for comparison as, to the best of our knowledge, it is currently the only explanation method for tabular embeddings. However, we explored alternative methods for creating explanations for the embedding space, including Association Rules, cluster representatives, and other explainability algorithms designed for downstream tasks, such as SHAP. Utilizing the methods of objective evaluation, we compare our method to these algorithms and report the results. **(3) Human-Centered Experiments:** In this part, we conduct a user study to validate our metric for measuring explanation quality and its overall value in the data science exploration process. We also include experiments measuring application-grounded metrics [33] to evaluate the quality of machine-produced explanations compared to standard data science tools. By assessing the increase in user productivity when following machine-produced explanations versus using traditional data analysis and data science tools, we can gauge the effectiveness of our explanations.

Summary of the results. The results of our experiments demonstrate that the explanations generated by TabEE effectively capture the patterns learned by the embedding model. Notably, there is a strong correlation between high scores obtained by our metric and the actual quality of the explanations, as observed in both objective empirical experiments and subjective user opinions from our user study. Furthermore, our user study indicates that users who utilized our system achieved improved objective performance, such as the ability to identify data patterns and understand the distinctions between different embedding models for subsequent analysis. These findings validate the effectiveness and utility of TabEE in enhancing data analysis and model comprehension.

6.1 Experimental Setup and General Parameters

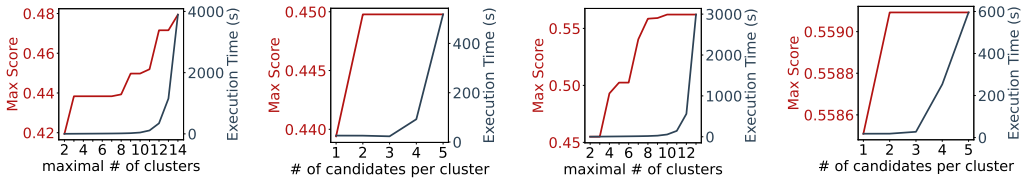
System setup. Our system is implemented in Python 3.8 as a local Python library that is given as input a trained embedding model as a black box, alongside with the original dataset, and therefore can be used in common exploratory data analysis (EDA) environments, such as Jupyter notebooks. As requirements we are using standard Python data analysis libraries - Scikit Learn, Numpy, Pandas, Matplotlib and UMAP. The experiments were run on an Intel Xeon CPU-based server with 24 cores and 96 GB of RAM.

Embedding models & datasets. We extensively assessed various tabular embeddings derived from different methods designed for diverse tasks, aiming to demonstrate the effectiveness and versatility of our explanation method across a wide range of scenarios. This evaluation highlights our method's robustness in providing meaningful explanations for different tabular embedding models (briefed at Section 2), including: SubTab (W2V) [34], Vime [52], EmbDI [11], TransTab [46], TabNet [5], SubTab (latent space) [42].

The datasets used in our experiments are as follows: *(1) Spotify* (*SP*) [2]: Contains music-related features with 42K rows, including numerical and categorical attributes. Used for genre classification. *(2) CoverType* (*CT*) [9]: Information about forest cover types with 581K rows. Includes numerical and categorical attributes. *(3) Flights* (*FL*) [1]: Data about airline flights with 6M rows. Consists of numerical and categorical attributes. *(4) HiggsBoson* (*HB*) [49]: Used in particle physics experiments with 11M rows. Contains only numerical attributes related to particle energy and momentum.

These datasets cover diverse domains and attribute types, allowing comprehensive evaluation of our proposed method. We conducted experiments on all combinations of embeddings and datasets, but present results for representative combinations due to space limitations. Other combinations exhibit similar trends.

System parameters optimization. We optimized the system parameters through ablation experiments, considering their impact on explanation quality and running time. Default values were chosen based on effectiveness, but users can customize these parameters to their specific needs. The following parameters were evaluated: *(1) Number of candidates per cluster*: The default is three, balancing quality and efficiency. Increasing candidates improves quality but increases running time. *(2) Range of clusters examined*: The default range is 2 to 10. It captures underlying patterns while considering running time. *(3) Number of explanations per cluster*: By default, one explanation is output per cluster in the global explanation. Multiple explanations can be chosen for a more comprehensive understanding, but complexity may increase. *(4) Importance of metric components*: by default equal weights are assigned. *(5) Distributions distance metric*: Jensen-Shannon (JS) metric is used, handling both numeric and categorical values effectively. *(6) Binning methods*: We use FEDEX's [17] binning method for discretizing numeric attributes, ensuring equal distribution within each bin for better comparison and distance calculation. *(7) Clustering method*: The default is KMeans, chosen for its simplicity and fast runtime.



(a) Spotify, VIME (b) Spotify, TabNet

Fig. 4. Quality and execution time analysis

Balancing running time versus quality. To ensure a reasonable running time for interactive user experience while maintaining good quality explanations, we evaluated the system's performance with default parameter settings on various datasets and embedding models. Figure 4 illustrates the relationship between explanation quality (red line) and running time in seconds (blue line) for the two main parameters of the system: the range of the number of clusters and the number of candidates examined per cluster, for general parameters optimization. In each pair of figures, the left figure presents score and execution time versus maximal number of clusters, and the right figure presents score and execution time versus number of candidates per cluster. Dataset names and embedding types denoted under each couple of figures. Based on these results and the analysis of other embeddings and datasets, we set the default maximum number of clusters to 10 and the number of candidates per cluster to 3. These values strike a balance between maximizing the metric score for each task and maintaining reasonable execution time, making them suitable for exploratory data analysis (EDA) purposes.

6.2 Objective Evaluation

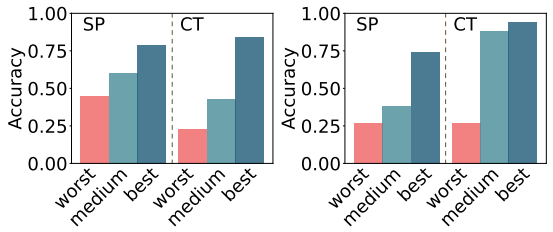
We perform experiments to objectively assess the quality and relevance of our explanations. These experiments involve analyzing patterns in the embedding space using synthetic data, checking the consistency between explanations and embeddings, and validating our metric (Sec. 4) through ablation studies and a downstream task. These assessments offer a comprehensive evaluation of the explanations and their alignment with the underlying embedding model.

Capturing patterns in the embedding space. To validate TabEE's robustness and effectiveness,we performed experiments with synthetic datasets and embedding models. These experiments assessed the fundamental properties of our algorithm and simulated real-world scenarios. Given that our algorithm hinges on its ability to segment the embedding space into cohorts based on the quality of the explanations provided, this experiment aimed to demonstrate the algorithm's resilience when faced with different numbers of "correct" clusters in the embedding space, across varying dimensionalities.

We generated synthetic datasets and embedding representations that were created using the identity function. Both datasets and embeddings were generated as clusters centered around randomly chosen points within an embedding space of specified dimensionality. For each configuration of synthetic data, denoted by the number of clusters (rows in Figure 5), and the length of the embedding vector (columns in Figure 5), we entrusted our system to determine the optimal number of clusters, with the aim of maximizing the quality of explanations. Then, we compared the number of clusters suggested by TabEE to the original number of clusters. This entire process was repeated 10 times for each configuration to establish statistical significance. In Figure 5 we report the average error in choosing the number of clusters based on our method compared to the actual number. The results are indicative of the remarkable accuracy (as the error is mostly 0) of TabEE in selecting

# clusters	dim = 10	dim = 23	dim = 56	dim = 133
5	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
10	-0.4 ± 0.7	0.0 ± 0.0	0.0 ± 0.0	-0.2 ± 0.6
15	-1.2 ± 1.7	0.0 ± 0.4	-0.2 ± 0.4	-0.1 ± 0.3

Fig. 5. Average error in selecting the number of cohorts (rows) for embedding vector dimensions (columns) - synthetic data



(a) TabNet (b) VIME

Fig. 6. Explanation-Clustering correspondence results

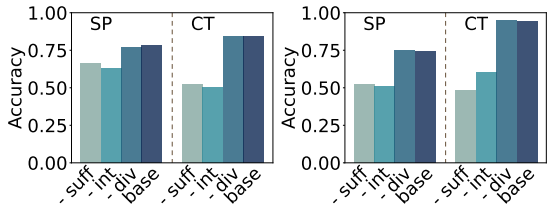
the exact number of clusters across a wide range of configurations. This attests to the robustness of our method in effectively capturing the inherent structure of both data and embeddings, even when confronted with intricate scenarios characterized by varying numbers of clusters and dimensionalities.

Correspondence between explanations and embeddings. In the following experiments, we aim to illustrate through simulations how our explanations help users gain intuition regarding the relationship between the latent representations and the original data.

(1) Overall correspondence. We utilize the explanations generated by our algorithm to identify the cluster whose explanation best fits the sampled tuple. Specifically, we calculate the probability $\Pr[x \sim Exp(C_i)]$, as formulated in Section 4, for each cluster C_i, and choose the cluster i' with the highest probability. Then, we compare it with the actual cluster the embedding of that sample was assigned by the clustering method. We report the accuracy results of these tests in Figure 6. Note that for each bar, the score was normalized according to the number of clusters, using Min-Max normalization[1]. In all of the experiments, the presented values for the best metric score significantly surpass the random baseline (zero score). This indicates that the generated explanations effectively align with the way the original attributes divide the embedding space.

(2) Metric correspondence. We run 3 experiments per dataset and embedding, differing by the generated explanation for analysis. For the first we have used the best explanation (i.e. the explanations we show the user in our algorithm), for the second a medium-scored explanation, and for the third the worst explanation across our possible candidates. The results, presented in Fig 6, demonstrate a strong correspondence between the values of the metric and the accuracy score in the experiments. These results support the claim that explanations with higher metric scores fit better the relations between the original attributes and the latent representation learned by the embedding model, and therefore justifies our use of the defined metric.

[1]The motivation for the normalization is based on the fact that given k clusters, the probability to randomly assign a correct cluster is $\frac{1}{k}$, and therefore making it easier to gain a higher score when the explanation consists of less cohorts.



(a) TabNet (b) VIME

Fig. 7. Metric components ablation experiments results

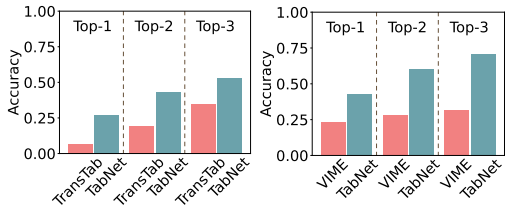
	JS-Divergence	KS Score	TVD
TabNet	0.78 ± 0.00	0.78 ± 0.00	0.78 ± 0.02
VIME	**0.82 ± 0.05**	0.63 ± 0.04	**0.81 ± 0.05**
TransTab	**0.99 ± 0.01**	0.42 ± 0.03	**0.99 ± 0.01**

Fig. 8. Correspondence scores of distance metrics (*SP*)

Quality metric ablation studies. In accordance with [45], there is no universal consensus regarding the definition of AI explanations and how to assess their validity and reliability. Nevertheless, we selected three metrics that capture various aspects of the calculated explanation, which have been previously employed in similar works. Using the *overall correspondence* score, we perform experiments to evaluate each component of our metric and explore the impact of other potential components that we ultimately chose to exclude from the final metric. The results of the ablation experiment are illustrated in Figure 7. In each experiment, we nullify a specific part of the metric and observe how the correspondence task score changes.

It is evident that for the Sufficiency (*suff*) and Interestingness (*int*) metrics, each plays a significant role in selecting meaningful explanations. The removal of these components from the metric leads to a notable decrease in the accuracy score in the experiment. We also examined commonly used clustering quality metrics, such as the Silhouette score [36] (*sil*) and Dunn score [19], when included in a similar manner, instead of our metrics (specifically instead of sufficiency). As depicted in Figure 7, the replacement of sufficiency in Silhouette score results in a decrease of the correspondence score. Same result is obtained when Dunn Index is used, and was omitted from the figure due to lack of space. Regarding the Diversity metric, removing it from the overall scores does not affect the performance on this task, but can generate different explanations patterns. As discussed in Section 6.4, some users prefer higher diversity score, so we fixate a zero weight of the diversity in order to improve execution time, but allow the users to change it upon demand.

Distributions distance metric comparisons. We chose JS divergence [27] as our metric due to its widespread applicability across various domains and its effectiveness in handling both categorical and numerical distributions. To justify this, we conducted ablation experiments, using *overall correspondence* score, exploring different distance measures, including the total variation distance (TVD) [44] and the Kolmogorov–Smirnov (KS) test [30], across all datasets and embedding models (*SP* dataset results summarized in Figure 8). In all settings, the JS metric either outperforms or performs equally well compared to the other metrics when examined by the correspondence score. However, due to the fact TVD metric showed comparable performance, we provide flexibility for users to choose their preferred distance metric.



(a) Spotify (b) Higgs Boson

Fig. 9. SHAP as an automatic verification of our method

Downstream task patterns correspondence. We use *SHAP* to validate patterns identified by TabEE, by using SHAP's importance scores for each attribute across the dataset in the context of the downstream task. While SHAP cannot replace our method, it proves beneficial when the embedding is used for downstream tasks, especially for capturing valuable patterns assisting a classification model. To account for potential issues with multicollinearity in SHAP explanations, we exclude correlated columns from the experiment.

We train embedding and classification models for the datasets' respective tasks. TabEE generates the explanations and SHAP values are calculated for the end-to-end task, combining the embedding and classification model. This enables us to obtain SHAP values on the original attributes, even if they are not directly input into the classification model. Despite SHAP's longer runtime, even on a small sample of points, TabEE remains significantly faster.

Next, we compare the explained attribute from TabEE's explanation to the attribute with the highest SHAP score for each example in the dataset, calculating the correspondence rate. We also assess the correspondence between TabEE's attribute and one of the top two attributes of each datapoint, the top three, and so on (presented in Figure 9). In both of the datasets, TabNet achieves higher classification results (based on accuracy and F1 metrics), and in the Figure 9 it is prominent that TabNet achieves correspondingly higher SHAP-correlation scores. The correspondence results are generally favorable in all settings, providing evidence of our method's performance. Furthermore, the results indicate that the correspondence scores improve as the embedding model's performance on the downstream task increases, suggesting that our explanations unveil strong, useful patterns that align with the patterns used by the classification model.

Handling multicollinearity. Various interpretability methods, such as SHAP, struggle with multicollinear datasets, leading to inaccurate explanations such as incorrectly dividing the importance of highly correlated attributes. Our experiments showcase the resilience of TabEE to multicollinearity, providing reliable explanations. To assess this, we trained the embedding model on the original datasets with the addition of attributes highly correlated with some of the original attributes. These correlated attributes were generated by introducing noise until achieving a Pearson correlation of 0.9. To evaluate the stability of TabEE, we applied the algorithm to the generated embeddings twice—once with the augmented dataset and once with the original dataset. We measured the average sufficiency score of the original attributes in all cohorts in both scenarios and compared the outcomes. On average, the relative change in the average score in the augmented dataset setting compared to the original dataset setting was 2%. In 83% of cases, the change was less than 1%, indicating minimal to no change. In the remaining 17%, the average relative change was 11%, suggesting that even in this scenario, the shifts were not substantial. Furthermore, TabEE demonstrated consistency in the number of cohorts in 90% of the runs.

	Medoids	Centroids	DIFF-2	DIFF-2.5	TabEE
TabNet	0.36 ± 0.01	0.38 ± 0.13	0.73 ± 0.00	0.5 ± 0.02	**0.78±0.00**
VIME	0.31 ± 0.11	0.23 ± 0.13	0.56 ± 0.23	0.41 ± 0.12	**0.82±0.05**
TransTab	0.52 ± 0.16	0.75 ± 0.11	0.12 ± 0.01	0.02 ± 0.12	**0.99±0.00**
EmbDI	0.07 ± 0.00	0.07 ± 0.02	0.76 ± 0.01	0.76 ± 0.02	**0.80±0.03**

Fig. 10. Baselines correspondence scores on *SP* dataset

6.3 Baselines Comparisons

In order to effectively benchmark our system against potential baselines, we executed experiments aimed at quantifying the *overall correspondence* between the cluster assigned by the clustering model and the accompanying explanation (detailed in Section 6.2). The results emphasise that TabEE's explanations exhibit greater robustness and coherence in capturing patterns learned by the embedding model compared to the alternatives. To explore these alternatives we conducted experiments in two distinct phases. Initially, as certain methods necessitate clustering the embedding space, we performed clustering solely based on the Silhouette score. Next, the explanations were created based on the clustering, as detailed bellow.

Cluster representatives. To consider cluster explanations as an alternative, we adopted the practice of selecting the medoid (or centroid) for each cluster in the embedding space, utilizing its original attributes to represent the entire cluster. This means that for each data point in the original dataset, there is now another data point representing it in the embedding space, without knowledge of which attributes and values in the original space contributed to their proximity in the embedding space. Essentially, when a row in the original space serves as the representative of a collection of other rows, it becomes challenging to highlight the reasons for this representation. Different attributes are represented differently, with varying weights in the embedding model. Furthermore, as the embedding space often contains more attributes than the original data, the proximity of points in the embedding space could be due to combinations of specific values of attributes in the original space.

To measure the correspondence of these explanations, we adapted the *overall correspondence* score, and instead of measuring the probability distribution, we assess how similar the representative is to other points in the cluster in the original space. Given that the original data includes various attributes of different data types, we normalize numeric attributes and perform one-hot encoding for categorical attributes. If the representative row accurately represents a set of other rows, we expect it to be similar in values to the other data points in the original space, compared to other clusters' representatives. Next, we measure the proportion of data points for which the representatives (chosen by the clusters in the embedding space) are indeed the most similar cluster in the original space. The results are presented in the *medoids* and *centroids* columns in Figure 10. In the *medoids* baseline, the cluster representatives are the medoids of each cluster (the example with smallest sum of distances to the rest of the cluster), and in the *centroid* baseline the cluster representatives are the closest datapoints to the centroid in the embedding space of each cluster. As observed, both medoids and centroids failed to accurately capture the patterns of the embedding space.

DIFF operator. We explored the DIFF [4] operator as an alternative approach to shifting distributions in TabEE. Instead of focusing on distributional changes, this method analyzes changes in the patterns of examples both within and outside a cluster. Concretely, it quantifies changes in the

support of association rules inside the cluster compared to the entire dataset by the values defined as *risk*.

We evaluated the DIFF as a baseline by conducting experiments with various *risk* values ranging from 1.5 to 2.5. In order to measure the *overall correspondence* in this setting, for a given example of the original attributes, we calculated the proportion of rules that apply within each cluster. The example was then assigned to the cluster with the highest proportion. Next, we compared the assigned cluster to the actual cluster designation and calculated the proportion of correct assignments across the entire dataset. The results presented in Figure 10 in the columns labeled $DIFF - 2$ and $DIFF - 2.5$, represent the use of DIFF with the specified *risk* values. Although the original paper employing this method used a *risk* of 3, in our experiments, no such examples were found. Even when considering the lowest *risk* of 1.5, which implies that the representation of rules captured within the cluster is only slightly overrepresented compared to the rest of the datapoints outside of the cluster, there were very few association rules, and they failed to capture significant patterns.

Downstream task explanation methods. TabEE helps to distinguish between errors in the embedding model and downstream model errors, a feat unattainable by current global explainability methods like global surrogate models [32] We tested two scenarios: using a well-trained embedding model with random labels and a random embedding model with correct labels. TabEE explained the embedding in both scenarios, and we used the entire end-to-end process of embedding and prediction with a global surrogate model, in each scenario we sample multiple random states of the system for the statistical significance, and present results for VIME trained over *CT* as a representative. TabEE's explanation scores significantly differed between the two scenarios. In the random embeddings scenario, the average score was 0.16 ± 0.00, while in the actual embeddings with random labels, the average score was 0.69 ± 0.02. This noticeable distinction was also validated in the user study (Section 6.4).

However, global surrogate feature importance maps failed to clearly distinguish between the scenarios. In both cases, the importance scores were close to zero and exhibited random values, highly dependent on the model's random state. The average *Rank-Biased Overlap* (RBO) [47] score between rankings based on different random states in the first and second scenarios were 0.69 ± 0.09 and 0.71 ± 0.06 respectively. Comparing rankings from the first scenario with those from the second scenario resulted in an average *RBO* of 0.63 ± 0.08, indicating that the difference within each scenario is similar to the difference between rankings of different scenarios. Thus, users could not differentiate between the scenarios based on feature importance rankings. For reference, in a scenario of using a correct embedding and model, different random states create rankings with an average *RBO* of 0.86 ± 0.04. The average *RBO* when comparing a ranking from this third scenario with a ranking from one of the two random scenarios is 0.69 ± 0.07 on average. The results introduce similar trends for local explainability methods such as SHAP and LIME [8], which, although not providing a global explanation like TabEE, can produce broader explanations by separately creating explanations for a sample of the data. Those methods, as opposed to TabEE, could not differentiate between the described scenarios.

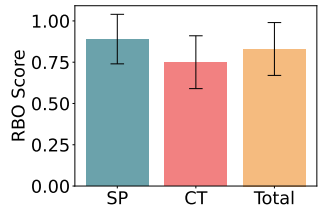
6.4 Human-Centered Experiments - User Study

Next, we use extrinsic quality indicators that involve real users, to validate (1) our system helpfulness by independent experiments; and (2) the adequacy of our intrinsic metrics by testing the correlation of our metric with extrinsic usefulness metrics.

Exploration task analysis. We recruited 20 participants with varying levels of expertise in data science (junior to expert), from academic and industrial backgrounds. The user study consisted of

	Default	TabEE
# of additional steps	9.3 ± 4.4	**2.7 ± 1.1**
# of accurate insights	0.2(17%) ± 0.2	**1.5 (44%) ± 0.5**
% users with no insights	33%	**0%**

(a) Exploration task analysis results



(b) Metric verification task results

Fig. 11. User study results

two groups, one which used TabEE and one which did not. Each group was assigned two notebooks, each contained a dataset and two embedding methods, TabNet and VIME, used for a classification task (Genre and Cover Type predictions). In both notebooks, the model based on TabNet embeddings performed better than the model based on VIME. The task was to understand the gap in performance between the methods, by exploration of the datasets and embedding models, within a time limit of 15 minutes per notebook. This constraint was used to challenge the participants in performing the task, while allowing them sufficient time to gain valuable insights. During the user study, participants were instructed to focus on uncovering patterns and relationships learned by the embedding and classification models, using a data-driven exploration approach. They were explicitly instructed not to focus on the model architecture of the embedding methods or attempt to retrain the models or embeddings. To evaluate the insights generated by the participants, we compared their responses to a pre-defined set of ground truth insights (accessible in [3]). For instance, explaining that TabNet divide the embeddings space (to high values and low values) based on an attribute that is more correlated to the label than the attribute VIME uses to divide the embeddings space. A few additional insights were identified by some participants during the experiment and were further verified through a rigorous evaluation process, which involved cross-referencing with existing knowledge and consulting domain experts when necessary. The first group (denoted *Default*) was provided with only the datasets, embedding vectors, and a classification model, while the second group (denoted *TabEE*) had access to TabEE. Both groups were instructed to use any data exploration tools available to them, which were indeed used by both groups. The *Default* group served as a rich baseline, allowing for a comprehensive comparison of our method to other available approaches.

Table 11a depicts the averaged results achieved by participants, with industry and academic experience, and reflects the inherent challenges of the task, caused by the complexity of the task and enhanced by the 15 minutes time constraint. It includes the number and percentage of correct insights, the percentage of users who did not derive insights at all, and the total number of insights averaged across all users and datasets. Note that when using TabEE, users derived an average of almost 2 correct insights per dataset, which is 5X more in average than users without TabEE (*Default*). As for incorrect insights, the *Default* group reached false or irrelevant conclusions since there is no ad-hoc tool for comparison two embedding models, and the existing tools for different purposes might be misleading. Finally, observe that when using TabEE almost 100% of users

were able to successfully finish the analysis task and derive at least one correct insight, which was not true for the other baselines. The performance of the *TabEE* group demonstrated the efficacy of our tabular embedding explanations in facilitating efficient and effective analysis of a complex task, even in highly time-constrained scenarios. This outcome underscores the critical importance of providing explanations for tabular embedding models, as they enable users to gain valuable insights in a timely manner.

Metric verification. The second user study we have conducted focused on assessing the correlation of our metric with users' preferences, in order to prove that our algorithm outputs the most preferable explanation. The participants in the study were provided with a set of three explanations and were asked to rank them. The participants were instructed to rank highest the explanations that they believed would be most useful for investigating the performance gap between the embedding methods, as introduced in the first user study. The explanations presented to the users were carefully chosen, including one with a high score, one with a medium score, and one with a low score, from all possible candidates for explaining the embedding and dataset. After collecting the rankings from the participants, we used the Rank-Biased Overlap (RBO) [48] similarity measure to compare the agreement between the participants' rankings and our metric. The results, presented in Figure 11b, demonstrate a general level of agreement across datasets and participants.

Interestingly, there was unanimous agreement among participants that the low-score explanation was considered the worst. This suggests a general consensus that explanations with lower scores may be less preferred or less valuable when investigating the performance gap. Regarding the comparison between the medium and high-score explanations, the majority of users preferred the high-score explanation for both datasets. This indicates a preference for explanations that score higher in terms of our metric, and provides an empirical human-centered justification for using the suggested metric.

However, note that 33% of the users preferred the medium-score explanation over the high-score explanation, due to higher diversity in the medium-scored explanation compared to the high-scored. This suggests that there might be some variability in user preferences, and some users may prioritize different aspects of explanations, such as diversity or other factors, over higher scores. Following these results, and the ablation experiments reported in Section 6.2, we allow the users using our module to change the weight of the diversity metric (or any other metric component) according to their preferences.

7 CONCLUSION AND FUTURE WORK

In conclusion, our system offers a framework for interpreting tabular embedding models, facilitating the discovery of data patterns and improving model understanding. By utilizing cohorts in the embedding space and generating explanations, we enable comprehensive analysis. The proposed metric effectively evaluates explanation quality, as demonstrated through extensive experiments and a user study that validates the system's value in complex tasks. Future research directions include adaption to addressing fairness and biases, and exploring new challenging use cases. These advancements will enhance interpretability and broaden the system's applicability.

ACKNOWLEDGMENTS

The protocol conforms to the ethical requirements of Human Research enforced by *Tel Aviv University* Ethics committee.
This research was partly supported by BSF - the US-Israel Binational Science foundation (grants No. 2018194) and iSF - the Israel Science foundation (grant 2707/22 of the Breakthrough Research Grant (BRG) Program).

REFERENCES

[1] 2015. Flights Dataset. https://www.kaggle.com/usdot/flight-delays?select=flights.csv.
[2] 2020. Spotify Dataset. https://www.kaggle.com/datasets/mrmorj/dataset-of-songs-in-spotify.
[3] 2023. TabEE git repository. https://github.com/KathyRaz/TabEE.
[4] Firas Abuzaid, Peter Kraft, Sahaana Suri, Edward Gan, Eric Xu, Atul Shenoy, Asvin Ananthanarayan, John Sheu, Erik Meijer, Xi Wu, et al. 2021. DIFF: a relational interface for large-scale data explanation. *The VLDB Journal* 30 (2021), 45–70.
[5] Sercan Ö Arik and Tomas Pfister. 2021. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6679–6687.
[6] Zhifeng Bao, Yong Zeng, HV Jagadish, and Tok Wang Ling. 2015. Exploratory keyword search with interactive input. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. 871–876.
[7] Ramon Bespinyowong, Wei Chen, HV Jagadish, and Yuxin Ma. 2016. ExRank: An exploratory ranking interface. *PVLBD* 9, 13 (2016), 1529–1532.
[8] Przemyslaw Biecek and Tomasz Burzykowski. 2021. Local interpretable model-agnostic explanations (LIME). *Explanatory Model Analysis; Chapman and Hall/CRC: New York, NY, USA* (2021), 107–123.
[9] Jock Blackard. 1998. Covertype. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C50K5N.
[10] Rajesh Bordawekar and Oded Shmueli. 2019. Exploiting latent information in relational databases via word embedding and application to degrees of disclosure. In *CIDR*.
[11] Riccardo Cappuzzo, Paolo Papotti, and Saravanan Thirumuruganathan. 2021. Embdi: generating embeddings for relational data integration. In *29th Italian Symposium on Advanced Database Systems (SEDB), Pizzo Calabro, Italy*.
[12] Jieying Chen, Jia-Yu Pan, Christos Faloutsos, and Spiros Papadimitriou. 2013. TSum: fast, principled table summarization. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*.
[13] Graham Cormode. 2017. Data sketching. *Commun. ACM* 60, 9 (2017).
[14] John P Cunningham and Zoubin Ghahramani. 2015. Linear dimensionality reduction: Survey, insights, and generalizations. *J. Mach. Learn. Res.* 16, 1 (2015).
[15] Hoa Khanh Dam, Truyen Tran, and Aditya Ghose. 2018. Explainable software analytics. In *Proceedings of the 40th International Conference on Software Engineering: New Ideas and Emerging Results*. 53–56.
[16] Sanjoy Dasgupta, Nave Frost, and Michal Moshkovitz. 2022. Framework for evaluating faithfulness of local explanations. In *International Conference on Machine Learning*. PMLR, 4794–4815.
[17] Daniel Deutch, Amir Gilad, Tova Milo, Amit Mualem, and Amit Somech. 2022. FEDEX: An Explainability Framework for Data Exploration Steps. *arXiv preprint arXiv:2209.06260* (2022).
[18] Rui Ding, Shi Han, Yong Xu, Haidong Zhang, and Dongmei Zhang. 2019. Quickinsights: Quick and automatic discovery of insights from multi-dimensional data. In *Proceedings of the 2019 International Conference on Management of Data*. 317–332.
[19] Joseph C Dunn. 1974. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics* 4, 1 (1974), 95–104.
[20] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. 2014. A multi-view embedding space for modeling internet images, tags, and their semantics. *International journal of computer vision* 106 (2014), 210–233.
[21] Joseph M Hellerstein, Ron Avnur, Andy Chou, Christian Hidber, Chris Olston, Vijayshankar Raman, Tali Roth, and Peter J Haas. 1999. Interactive data analysis: The control project. *Computer* 32, 8 (1999).
[22] Robert J Hilderman and Howard J Hamilton. 2013. *Knowledge Discovery and Measures of Interest*. Vol. 638. Springer Science & Business Media.
[23] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
[24] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37. https://doi.org/10.1109/MC.2009.263
[25] Sotiris B Kotsiantis. 2013. Decision trees: a recent overview. *Artificial Intelligence Review* 39 (2013), 261–283.
[26] Doris Jung-Lin Lee, Dixin Tang, Kunal Agarwal, Thyne Boonmark, Caitlyn Chen, Jake Kang, Ujjaini Mukhopadhyay, Jerry Song, Micah Yong, Marti A Hearst, et al. 2021. Lux: always-on visualization recommendations for exploratory dataframe workflows. *PVLDB* 15, 3 (2021), 727–738.
[27] J Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory* 37, 1 (1991), 145–151.
[28] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems* 30 (2017).
[29] Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018. DeepEye: Towards Automatic Data Visualization. ICDE.
[30] Frank J Massey Jr. 1951. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association* 46, 253 (1951).
[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*.

[32] Christoph Molnar. 2020. *Interpretable machine learning*. Lulu. com.
[33] Alun Preece. 2018. Asking 'Why'in AI: Explainability of intelligent systems–perspectives and challenges. *Intelligent Systems in Accounting, Finance and Management* 25, 2 (2018), 63–72.
[34] Kathy Razmadze, Yael Amsterdamer, Amit Somech, Susan B Davidson, and Tova Milo. 2022. SubTab: Data Exploration with Informative Sub-Tables. In *Proceedings of the 2022 International Conference on Management of Data*. 2369–2372.
[35] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.
[36] Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7
[37] Sunita Sarawagi, Rakesh Agrawal, and Nimrod Megiddo. 1998. Discovery-driven exploration of OLAP data cubes. In *EDBT*.
[38] Manish Singh, Michael J Cafarella, and HV Jagadish. 2016. DBExplorer: Exploratory Search in Databases. *EDBT* (2016).
[39] Arjun Srinivasan, Steven M Drucker, Alex Endert, and John Stasko. 2018. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 672–681.
[40] Bo Tang, Shi Han, Man Lung Yiu, Rui Ding, and Dongmei Zhang. 2017. Extracting top-k insights from multi-dimensional data. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 1509–1524.
[41] Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. 2021. Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems* 34 (2021), 18853–18865.
[42] Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. 2021. Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems* 34 (2021), 18853–18865.
[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[44] Sergio Verdú. 2014. Total variation distance and the distribution of relative information. In *2014 Information Theory and Applications Workshop (ITA)*. IEEE, 1–3.
[45] Giulia Vilone and Luca Longo. 2021. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion* 76 (2021), 89–106.
[46] Zifeng Wang and Jimeng Sun. [n. d.]. TransTab: Learning Transferable Tabular Transformers Across Tables. In *Advances in Neural Information Processing Systems*.
[47] William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)* 28, 4 (2010), 1–38.
[48] William Webber, Alistair Moffat, and Justin Zobel. 2010. A Similarity Measure for Indefinite Rankings. *ACM Trans. Inf. Syst.* 28, 4, Article 20 (nov 2010), 38 pages. https://doi.org/10.1145/1852102.1852106
[49] Daniel Whiteson. 2014. Higgs Boson Dataset. https://archive.ics.uci.edu/dataset/280/higgs.
[50] Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics* 8 (2020), 183–198.
[51] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *TVCG* (2016).
[52] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. 2020. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems* 33 (2020), 11033–11043.
[53] Brit Youngmann, Sihem Amer-Yahia, and Aurelien Personnaz. 2022. Guided exploration of data summaries. *Proceedings of the VLDB Endowment (PVLDB)* 15, 9 (2022), 1798–1807.
[54] Kai Zeng, Sameer Agarwal, Ankur Dave, Michael Armbrust, and Ion Stoica. 2015. G-ola: Generalized on-line aggregation for interactive analysis on big data. In *SIGMOD*.

Received July 2023; revised November 2023; accepted December 2023