SeerCuts: Explainable Attribute Discretization

Eugenie Lai
eylai@mit.edu
MIT

Inbal Croitoru
inbal.cr@campus.technion.ac.il
Technion

Noam Bitton
bitton.noam@campus.technion.ac.il
Technion

Ariel Shalem
ariel.shalem@campus.technion.ac.il
Technion

Brit Youngmann
brity@technion.ac.il
Technion

Sainyam Galhotra
sg@cs.cornell.edu
Cornell

El Kindi Rezig
elkindi@cs.utah.edu
University of Utah

Michael Cafarella
michjc@csail.mit.edu
MIT

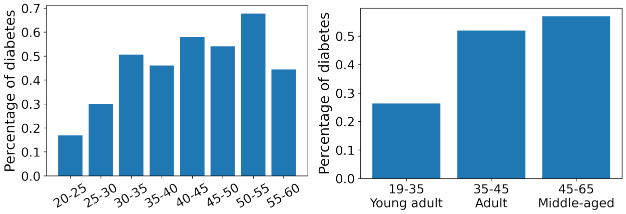
ABSTRACT

This demonstration showcases SeerCuts — a tool that suggests useful and semantically meaningful discretization strategies (partitions) for numerical attributes. SeerCuts is a generic, interactive framework where users specify attributes to discretize and their utility measure for a downstream task of choice. It uses GPT-4 to assess the semantic meaningfulness of candidate partitions and employs an efficient search strategy to explore the vast space of discretization options. With hierarchical clustering to group related partitions and a multi-armed bandit policy to identify useful partitions with only a few samples, SeerCuts quickly finds meaningful and useful partitions. In the demo, we will provide an overview of SeerCuts and allow the audience to explore various datasets and tasks, including data visualization and comprehensive modeling. The users will be able to evaluate how SeerCuts identifies meaningful discretization strategies and compare the tradeoff between different discretization options. A companion video for this submission is available at [1].

1 INTRODUCTION

Data discretization is the process of converting numerical attributes into categorical ones by grouping values into intervals (bins). For example, transforming a continuous age variable into bins representing decades or life stages. This transformation is widely useful in tasks such as data mining, data visualization, and causal inference [5]. Discretizing attributes enables symbolic data mining (e.g., decision trees, association rule mining) on continuous variables more easily, enhancing interpretability for analysis. Data-sharing projects are a priority in scientific and governance communities, but standard databases often result in untargeted design decisions. As a result, databases require adaptation for each specific use case. However, choosing an unsuitable discretization strategy can lead to low utility and unexplainable results. If the bins are not well-designed, the discretized data may fail to capture important patterns, reducing model performance. Additionally, bins without semantic meaning are hard to interpret and complicate insight communication.

Existing discretization techniques fall into two broad categories: (I) statistical methods, such as equal-width, are commonly used but often ignore domain knowledge, (II) domain-specific approaches, that often rely on handcrafted bins—such as medical categories for health-related attributes—which are interpretable but may lead to decreased utility in a downstream task. Despite the abundance of discretization techniques [5], striking a balance between statistical utility and semantic explainability remains an open challenge.



(a) Age **binned by equal width.** **(b)** Age **binned by life stages.**

Figure 1: (a) user's attempt; (b) SeerCuts output in Ex.1.

We present a system named SeerCuts that bridges the gap by offering an automatic tool to discover *useful and explainable* discretization strategies (referred to as partitions). In SeerCuts, the user inputs: (1) a database with the numerical attributes to be discretized and a target outcome attribute, and (2) a downstream task with a corresponding utility measure (e.g., model accuracy for tree-based modeling). SeerCuts outputs a set of partitions that aim to optimize both utility and interpretability. The following examples demonstrate how SeerCuts facilitates analysts in two example downstream tasks:

Example 1. Alex, an analyst assisting a research team, examines a diabetes dataset (Table 1) to understand the percentage of diabetes diagnosis w.r.t age for people between 21 and 60. She hopes to find an explainable partition that reveals interesting trends in the data. Here, the utility of the visualization task is measured by the correlation between age and the probability of having diabetes. Alex starts with a standard equal-width method to bin Age as: $(20, 25, 30, \ldots, 60)$ (Fig. 1a). She sees no clear trend and finds it difficult to meaningfully label the bins. Alex knows there are countless ways to bin Age and wonders if at least one is satisfactory. Verifying her work without extensive trial and error is impossible.

Now consider Alex uses SeerCuts. SeerCuts suggests a life-stage-based partition $(18, 35, 45, 65)$ with labels ("Young adult," "Adult," "Middle-aged") (Fig. 1b), a common approach in medical literature [10]. Alex quickly notices a sharp jump from young adult to adult, with a milder increase to middle-aged. She is pleased that her visualization is both interesting and explainable. □

Example 2. Alex is now training a tree-based model for diabetes diagnosis, with the partitioned Age, BMI, and Glucose attributes,

Table 1: A snapshot of the diabetes dataset

Glucose	BMI	Age	Outcome
148	23.1	50	0
139	33.3	35	1
89	28.3	21	1
137	33.6	33	1
183	22.7	46	0
130	30.1	27	0

in the hope of finding interpretable and useful patterns in the data to guide diabetes diagnosis. She starts with an automated machine learning package and finds that the highest accuracy partition for Age is $(20, 24, 30, 33, 42, 43, 47, 48, 54, 61)$ by ChiMerge [8]. After consulting with a diabetes expert, Alex realizes that this high-accuracy partition does not make much sense in medical contexts. She wonders if she can quickly find other partitions that maintain accuracy while improving the interpretability of the model.

Consider Alex uses SeerCuts. SeerCuts returns partitions that balance both goals: Age $(19, 35, 45, 60)$ ("Young adult," "Adult," "Middle-aged"); BMI $(18.5, 25, 30, 68)$ ("Healthy," "Overweight," "Obese"); and Glucose $(0, 140, 200)$ ("Normal," "Impaired"). This results in a model that is both accurate and explainable. Fig. 2 shows some example suggested partitions for this scenario. Here, BMI has only one medical-aware partition, and SeerCuts finds it even though Alex is unaware of this domain-specific. □

Challenges. SeerCuts takes a dataset with numerical attributes to be binned and a utility measure of the downstream task. It outputs a set of partitions that has high utility and explainability for each attribute. To obtain such a system, there are several challenges to overcome: (C1) Efficiently supporting multiple downstream tasks; (C2) Defining how to measure the semantic meaningfulness of the partitions; (C3) Navigating the vast search space (which is exponential in the number of attributes to bin) of potential partitions.

Contributions. To address (C1), we designed a generic framework that allows users to specify their utility measure for their task. In our demonstration, we will showcase SeerCuts's operation across multiple tasks, including data visualization, comprehensive modeling, and data imputation. To address (C2), SeerCuts uses GPT-4o [12] to measure the semantic meaningfulness of partitions, leveraging its comprehensive knowledge and ability to capture contexts. To address (C3), a search-based strategy is proposed. SeerCuts first curates the search space using commonly used discretization methods and ones suggested by LLMs to capture domain-specific candidates. SeerCuts expresses partitions as distributions over the data and applies a hierarchical clustering technique to group similar partitions. Then it uses a multi-armed bandit policy [3] for navigating among clusters. The key observation is drawing a sample partition from a cluster yields evidence about the other partitions within this cluster. Our search strategy is informed by this evidence to find useful partitions with only a few samples.

Related Work. There are numerous statistical data discretization techniques. [5] evaluates the effect of more than 90 techniques on the accuracy of learning models. These techniques range from the widely considered ones (e.g., equal width, ChiMerge) to more sophisticated ones (e.g., cluster-based or distance-based discretizers). SeerCuts considers commonly used techniques to identify partitions that are explainable and achieve high utility. Related work includes automated data visualization (AutoViz) and automated machine learning, which streamline visual analysis [14] and model training [7], enabling high-utility visualizations and models with less effort. SeerCuts focuses on data discretization, providing semantically meaningful partitions that enhance utility across various tasks, including visualizations and symbolic modeling. Semantic-aware binning methods in AutoViz often use historical data to recommend semantic bins for unseen attributes [13]. SeerCuts is a generic framework that applies to various downstream tasks and lets users adjust the trade-off between utility and explainability.

2 TECHNICAL BACKGROUND

2.1 Problem Formulation

We consider a database over a schema $\mathbb{A}$. Each attribute $A \in \mathbb{A}$ is associated with a domain $Dom(A)$, which can be categorical or continuous. A database instance D, populates the schema with a set of tuples $t = (a_1, \ldots, a_n)$ where $a_i \in Dom(A_i)$. We denote by $\mathbb{A}_{\text{num}} \subseteq \mathbb{A}$ the set of continuous variables to be discretized. Given an attribute A and a partition P of A into m bins, we denote by A_P the discretized variable A. We adopt the definition of bins in [4]. A *partition* is a list of m bins denoted as $P = (b_1, \ldots, b_m)$. A bin is written as $b = ([p, q]; n)$, where: (1) $p \leq b < q$ is the range of b for the first $m-1$ bins, (2) $p \leq b \leq q$ is the range of b_m, (3) n is the number of elements in the bin. P is a *valid partition* of A if: (1) every two bins in P have disjoint ranges, and (2) P covers all the values in $Dom(A)$.

2.1.1 Measuring Goal One: Semantic Meaningfulness. Our first goal is to assess the semantic meaningfulness of a given partition w.r.t. a numerical attribute. Developing a semantic measure for attribute partitioning is challenging due to the lack of a standardized approach. The subjective nature of semantic value necessitated the creation of a novel and interpretable method. We leverage the observation that the number of appearances in the literature or available code online can serve as a proxy for semantic meaningfulness, assuming that frequently referenced partitions reflect accumulated domain expertise and practical utility. However, counting references for a particular partition is not straightforward, as attributes may use different synonym names, and oftentimes the partitioning appears only in source code. We therefore leveraged LLMs due to their comprehensive knowledge base and ability to capture domain-specific nuances. We investigated two models: GPT-4o [12], known for its superior domain-specific knowledge, and Perplexity AI [2], selected for its capability to ground responses in referenced sources.

We conducted a 1-4 scale for the semantic value of a given partitioning method for an attribute where 1 represents poor semantic value (not used at all) and 4 represents excellent semantic value (very commonly used). We then normalized these scores to a 0-1 scale. When multiple attributes are binned, the semantic value scores are averaged between attributes. The specific prompts used are given in our Github[1] repository. To validate our approach, we conducted a user study involving 54 participants recruited by the Prolific platform[2]. The participants were given 26 questions, where each question consisted of an attribute description and a possible

[1] https://github.com/noambitton/Demo

[2] https://www.prolific.com/

Table 2: Example utility definitions for downstream tasks considered. D_B represented the binned data.

Task	Utility Definition	Explanation
Visualization	Spearman correlation coefficient [6] One-way ANOVA [9]	For Spearman - higher correlation indicates better utility. For ANOVA - a higher F-value suggests significant differences among the bins.
Comprehensive modeling	$\mathcal{M}(\mathcal{D}_{\mathcal{B}})$	The accuracy or parity of a prediction model $\mathcal{M}$ with the discretized data.
Data imputation	$\mathcal{M}(\texttt{imputed}(\mathcal{D}_{\mathcal{B}}))$	The model accuracy using imputed data, assuming the data is first binned, then a data imputation algorithm is employed, finally the prediction model is used.

partitioning strategy, and were asked to rank the partition's semantic value using the same 1-4 scale applied to LLMs. The form is given in our repository. We measured inter-rater agreement using Krippendorff's Alpha, obtaining a score of 0.2, highlighting the task's complexity, yet indicating moderate agreement among the participants. Using Spearman's Rank correlation, we compared human ratings with LLM assessments and found a strong correlation of 0.9 (p-val $< .05$) for GPT-4o and 0.83 (p-val $< .05$) for Perplexity. These high correlations motivate the interpretability we sought, demonstrating our method's potential to capture semantic nuance. We selected GPT-4o for its higher correlation with the participants.

2.1.2 Measuring Goal Two: Statistical Quality. With a database D and a downstream task $\mathcal{T}$, we denote the outcomes of applying $\mathcal{T}$ to D as $\mathcal{T}(D)$. We denote the utility of the task $\mathcal{T}$ computed over D as $utility(\mathcal{T}(D))$. The utility of each task is determined based on its specific objective. SeerCuts currently supports the following tasks: visualization, comprehensive modeling, and data imputation (details in Table 2). Future work would extend SeerCuts to support other tasks, including pattern mining, and causal analysis.

2.1.3 Problem Definition. Given a dataset and a set of numerical attributes to be binned $\mathbb{A}_{\text{num}}$, we aim to find a set of valid partitions for $\mathbb{A}_{\text{num}}$, s.t. the partitions are semantically meaningful and the utility of the downstream task is maximized. The user provides the following: (1) A **database** D with attributes to be binned $\mathbb{A}_{\text{num}}$ and an outcome attribute O (used to measure utility), (2) A **utility measure** of the downstream task $\mathcal{T}$. Our goal is to bin the attributes $\mathbb{A}_{\text{num}}$ to obtain the binned version of D, $D_{\mathbb{P}}$, that maximizes:

$$\alpha \cdot sem(\mathbb{P}) + (1-\alpha) \cdot utility(\mathcal{T}(D_{\mathbb{P}}))$$

where $sem(\mathbb{P})$ is the aggregate semantic value score of the partitions $\mathbb{P}$ (as explained in Section 2.1.1) and $0 \leq \alpha \leq 1$ is a system parameter.

Because the balance between semantic value and utility varies by scenario, SeerCuts generates a Pareto curve, as shown in Figure 2. Each point on this curve represents a partition set for all attributes in $\mathbb{A}_{\text{num}}$, ensuring that no other point outperforms it in both utility and semantic meaningfulness with respect to α.

2.2 The SeerCuts System

Candidate curation. We consider 11 common discretization techniques (e.g., ChiMerge, equal-width) to curate the search space. We also use GPT-4o [12] to suggest other commonly used discretization options based on domain knowledge and common practice. Our framework can be easily generalized to include other discretizers.

Partition distribution. We represent partitions as distributions over the data. Inspired by [14], we apply a partition P to its corresponding attribute A to obtain a histogram and use the histogram to compute the value distribution of P over A. For example, when binning the Age attribute, we get a partition $P = (b_1, b_2, b_3)$, where $b_1 = ([19, 35] : 33), b_2 = ([35, 45] : 57), b_3 = ([45, 60] : 24)$. We then get its distribution $(0.289, 0.5, 0.211)$ as a vector.

Identifying similar partitions. Our key insight is that candidate partitions that bin an attribute in a similar way would have similar utility and semantic meaningfulness. We then use this observation to reduce the search space by grouping together partitions that have a similar distribution. We use a hierarchical clustering method [11] due to its ability to detect the hierarchical structure of the clusters, which is leveraged in our next step. Since we are measuring the dissimilarity between distributions, we use the earth mover's distance (EMD) as the distance measure.

Estimating optimal partitions. Given clusters of candidate partitions, we use the Upper Confidence Bound (UCB) algorithm [3] for deciding among clusters. The key observation is drawing a sample partition (i.e., calculating its utility and semantic measure) from one cluster yields evidence about the entire cluster. We set the reward as the mean of the utility and semantic score of the drawn sample. Specifically, our search algorithm starts by randomly drawing one partition from each cluster and then ranks the clusters by its UCB. For the rest of the budget, at every iteration, SeerCuts draws a sample from the highest-ranked cluster, updates the UCB of that cluster, and re-ranks the clusters. This way, SeerCuts finds useful partitions in a large search space with a limited budget.

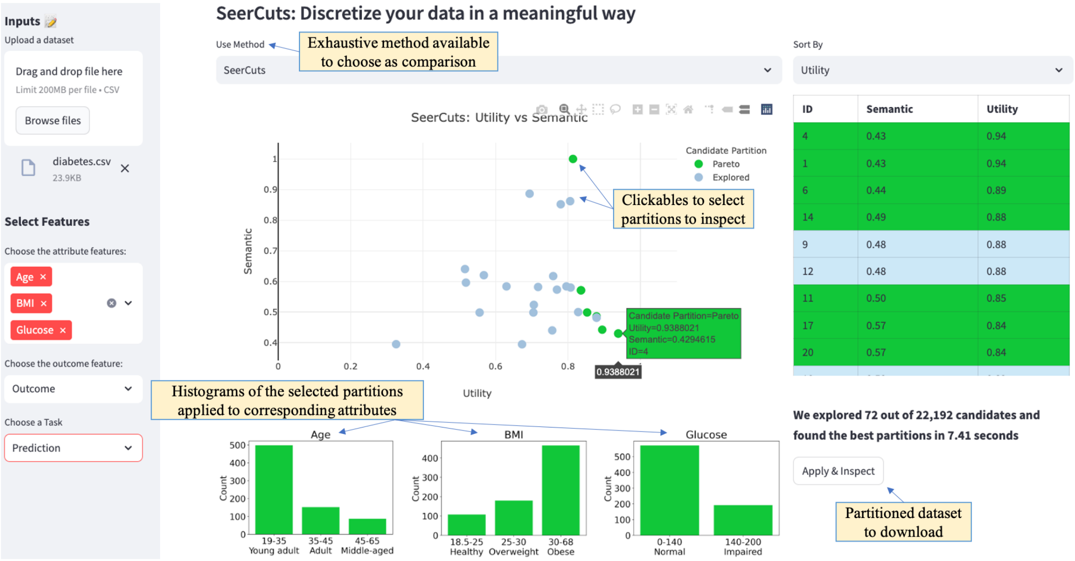
3 DEMONSTRATION

SeerCuts was implemented using Python and Streamlit[3] to provide an intuitive and expressive interface for analysts. Our code and datasets are available in our Github repository. Utility metrics are tailored to the specific task such as prediction (where utility is the model accuracy), as described in Section 2.1.2.

In this demonstration, we highlight how users can interact with SeerCuts and evaluate its quality in balancing semantic and utility-driven binning strategies to streamline this process for various downstream tasks. Participants will take on the role of analysts seeking to discretize their data to enhance the explainability and performance of their downstream tasks. We will allow them to choose one of the pre-loaded datasets. **Titanic**: historical passenger information from the Titanic. **Medical**: Clinical measurements in females for the detection of diabetes. **Spotify**: A dataset containing audio and artist features that reflect song popularity.

Overview of SeerCuts: To demonstrate the usability of SeerCuts, in this part, we will walk through illustrative examples of the usage of SeerCuts. We go through two common data science tasks: prediction and visualization. For the prediction task, we begin by uploading the Medical dataset, aiming to predict if a patient has diabetes. We select using the UI (left-hand side of Fig. 2) the attributes to bin: 'Age', 'BMI', 'Glucose', and specify 'Outcome' as the target outcome. We then choose "Prediction" from the available task options. Here, we fix the prediction model and utility function parameters, though users can configure them.

[3] https://streamlit.io/


SeerCuts: Discretize your data in a meaningful way

ID	Semantic	Utility
4	0.43	0.94
1	0.43	0.94
6	0.44	0.89
14	0.49	0.88
9	0.48	0.88
12	0.48	0.88
11	0.50	0.85
17	0.57	0.84
20	0.57	0.84

We explored 72 out of 22,192 candidates and found the best partitions in 7.41 seconds

Candidate Partition=Pareto
Utility=0.9388021
Semantic=0.4294615
ID=4

Figure 2: UI of SeerCuts (example usability for comprehensive modeling).

SeerCuts offers several options for exploring the recommended binning strategies. SeerCuts produces a graph illustrating the trade-off between utility (prediction accuracy in our case) and semantic scores for the recommended binning strategies (see the middle part of Fig. 2). SeerCuts presents only the sampled binning strategies as described in Section 2.2. Users can hover over any point in the graph to reveal its utility and semantic scores. Alongside the graph, SeerCuts provides a corresponding ranking table that lists the binning strategies shown on the graph for easy comparison between the strategies: Each strategy is represented by a unique Id corresponding to the points on the graph. The green colored points are the the Pareto points (recommended) and blue for suboptimal points. The table enables to rank the binning strategies from highest to lowest, with the default sorting based on utility. However, users can choose to sort by the semantic scores if desired. Clicking on a point in the graph brings up a detailed visualization of the binning strategy for the corresponding attributes as histograms, as shown in the bottom part of Fig 2. The user can apply a selected binning strategy and inspect the binned data in detail by clicking on the ”Apply & Inspect" button. The inspection view will replace the current graph, and a return button will allow users to switch back to the graph view. Additionally, they can download the binned data for further analysis.

For the visualization task, we demonstrate the process using the Spotify dataset, binning the ‘Loudness’ attribute and examining its correlation with the ‘Popularity’ attribute as the outcome. The workflow remains similar, but now the utility score is the correlation coefficient, which reflects how well the chosen binning method preserves trends in loudness distribution w.r.t. popularity.

Binning Demonstration: Next, attendees would be invited to interact with SeerCuts to test its capabilities with their specific use cases. Users can choose one of the datasets supported by SeerCuts or upload any structured dataset containing continuous variables and experiment with different combinations of utility measures and tasks. This simulates a real-world scenario where users can leverage SeerCuts across various domains and tasks, showcasing its versatility and effectiveness in different use cases.

Looking Under the Hood: SeerCuts performance can be compared with an exhaustive (naive) approach that exhaustively evaluates all possible binning strategies. While the naive approach computes utility and semantic scores for every possible binning strategy combination (one for each attribute), SeerCuts employs an efficient algorithm to achieve comparable results in significantly less time. The interface visualizes this comparison through a dual-panel display, showing how SeerCuts identifies high-quality solutions without the computational overhead of exhaustive search.

Finally, users interested in the prompts used for semantic scoring can reach out for detailed documentation and will be allowed to modify the prompts to observe the effect on the performance.

REFERENCES

[1] [n. d.]. Google Drive. https://drive.google.com/drive/folders/18ImrmCzu3vzDxNVgTarQrzYNnmsctfj2?usp=sharing.

[2] Perplexity AI. 2021. https://www.perplexity.ai/. Accessed: January 12, 2025.

[3] P Auer. 2002. Finite-time Analysis of the Multiarmed Bandit Problem.

[4] Rachel Behar and Sara Cohen. 2020. Optimal Histograms with Outliers.. In *EDBT*. 181–192.

[5] Salvador Garcia, Julian Luengo, José Antonio Sáez, Victoria Lopez, and Francisco Herrera. 2012. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *TKDE* (2012), 734–750.

[6] Jan Hauke and Tomasz Kossowski. 2011. Comparison of values of Pearson’s and Spearman’s correlation coefficients on the same sets of data. *Quaestiones geographicae* (2011), 87–93.

[7] Yuval Heffetz, Roman Vainshtein, Gilad Katz, and Lior Rokach. 2020. DeepLine: AutoML Tool for Pipelines Generation Using Deep Reinforcement Learning and Hierarchical Actions Filtering. In *KDD ’20*.

[8] Randy Kerber. 1992. Chimerge: Discretization of numeric attributes. In *AAAI*. 123–128.

[9] Tae Kyun Kim. 2017. Understanding one-way ANOVA using conceptual figures. *Korean journal of anesthesiology* 70, 1 (2017), 22.

[10] Morris L Medley. 1980. Life satisfaction across four stages of adult life. *The International Journal of Aging and Human Development* (1980), 193–209.

[11] Daniel Müllner. 2011. Modern hierarchical, agglomerative clustering algorithms. arXiv:1109.2378 [stat.ML] https://arxiv.org/abs/1109.2378

[12] OpenAI. 2024. GPT-4o. arXiv:2410.21276 [cs.CL] https://arxiv.org/abs/2410.21276

[13] Vidya Setlur, Michael Correll, and Sarah Battersby. 2022. Oscar: A semantic-based data binning approach. In *IEEE VIS*. IEEE, 100–104.

[14] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya Parameswaran, and Neoklis Polyzotis. 2015. Seedb: Efficient data-driven visualization recommendations to support visual analytics. In *PVLDB*. NIH Public Access.