SHAP-based Explanations are Sensitive to Feature Representation

Hyunseung Hwang
KAIST
Daejeon, Republic of Korea
aguno@kaist.ac.kr

Andrew Bell
New York University
New York, USA
alb9742@nyu.edu

Joao Fonseca
New York University
New York, USA
jpm9748@nyu.edu

Venetia Pliatsika
New York University
New York, USA
venetia@nyu.edu

Julia Stoyanovich
New York University
New York, USA
stoyanovich@nyu.edu

Steven Euijong Whang
KAIST
Daejeon, Republic of Korea
swhang@kaist.ac.kr

Abstract

Local feature-based explanations are a key component of the XAI toolkit. These explanations compute feature importance values relative to an "interpretable" feature representation. In tabular data, feature values themselves are often considered interpretable. This paper examines the impact of data engineering choices on local feature-based explanations. We demonstrate that simple, common data engineering techniques, such as representing age with a histogram or encoding race in a specific way, can manipulate feature importance as determined by popular methods like SHAP. Notably, the sensitivity of explanations to feature representation can be exploited by adversaries to obscure issues like discrimination. While the intuition behind these results is straightforward, their systematic exploration has been lacking. Previous work has focused on adversarial attacks on feature-based explainers by biasing data or manipulating models. To the best of our knowledge, this is the first study demonstrating that explainers can be misled by standard, seemingly innocuous data engineering techniques.

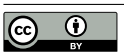
CCS Concepts

• **Human-centered computing**; • **Computing methodologies** → **Machine learning**; • **Information systems** → **Data management systems**; • **Social and professional topics** → **Sociotechnical systems**;

Keywords

Explainable AI, SHAP, Feature Representation

ACM Reference Format:
Hyunseung Hwang, Andrew Bell, Joao Fonseca, Venetia Pliatsika, Julia Stoyanovich, and Steven Euijong Whang. 2025. SHAP-based Explanations are Sensitive to Feature Representation. In *The 2025 ACM Conference on Fairness, Accountability, and Transparency (FAccT '25), June 23–26, 2025, Athens, Greece.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3715275.3732105



FAccT '25, Athens, Greece

ACM ISBN 979-8-4007-1482-5/25/06
https://doi.org/10.1145/3715275.3732105

1 Introduction

Explainable AI (XAI) is becoming increasingly critical for justifying the behavior of AI systems implemented in high stakes domains like education, lending, public employment, and healthcare [1, 10, 39]. One of the key components of XAI is the use of *local feature-based explanations*, which quantify the importance of an observation's features to an outcome (or some other quantity of interest). For example, these explanations are foundational for *algorithmic recourse*, where understanding *why* an individual is rejected for a loan by an AI-assisted system allows them to contest or reverse that unfavorable decision. Local feature-based explanations can also be used to surface unfairness in decision-making, for example, if a model is revealed to be making individual-level decisions on the basis of features like age, gender, or race, which may be illegal to use under the disparate treatment doctrine.[1] Further, these explanations are becoming an essential tool to fulfill legal and regulatory requirements, such as the European Union's AI Act, and General Data Protection Regulation's "right to explanation" [14].

The Shapley value framework [29], originally developed for dividing revenue in cooperative games, is widely used to quantify local feature importance in predictive classification. It underpins prominent explanation methods like SHAP (Shapley Additive Explanations) [23] and QII (Quantitative Input Influence) [11]. The framework explains the classification outcome for an observation by assessing how changes to a feature's value, individually or in combination with others, impact that outcome. This process simulates interventions, aligning with causal inference principles by isolating each feature's influence while controlling for others. A high Shapley value for a protected feature like age suggests its significant influence on the classifier's decision.

However, Shapley-value-based explanations have limitations: they can mislead users (intentionally or unintentionally) [17] and are vulnerable to adversarial attacks and manipulations [22]. **In this paper, we focus on the key observation that local feature-based explanations, derived from trained models and post-processed data, are susceptible to manipulations through feature engineering, which occurs upstream from classification in the machine learning pipeline.** We use SHAP [23], the most widely adopted implementation of the Shapley value framework, to show that local feature-based explanations are influenced by simple data engineering operations, such as transforming continuous values or encoding categorical values, which modify feature

[1] https://en.wikipedia.org/wiki/Disparate_treatment

	age	*edu*	*label*	*prediction*
Ann	30	BS	1	0
Bob	40	BS	1	1
Cat	50	HS	1	1
Dan	40	MS	0	1

	age	*edu*	*label*	*prediction*
Ann	<50	BS	1	0
Bob	<50	BS	1	1
Cat	≥50	HS	1	1
Dan	<50	MS	0	1

age SHAP ⇩
edu SHAP ⇧

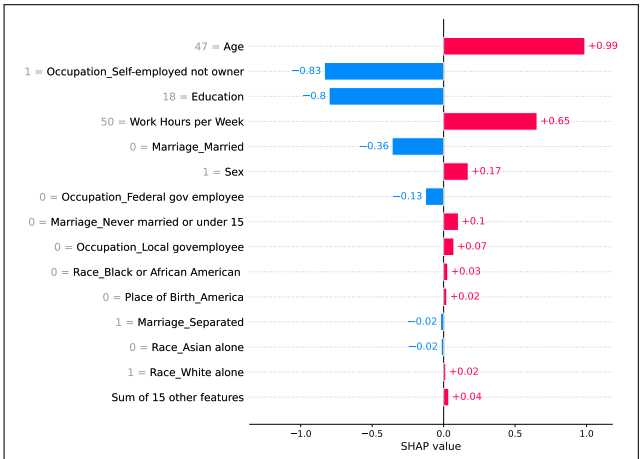
Figure 1: A hypothetical lending example: For a given classifier model, if we bucketize the *age* feature to generate a local explanation of the outcome with SHAP, then the importance of *age* for Ann decreases compared to using the raw value of the feature. Intuitively, this happens because $age = 30$ is infrequent—and low—in this hypothetical dataset, while $age < 50$ appears to be typical, both on its own and in combination with education.

representations. For instance, bucketization—a common method of grouping values into ranges—can make a feature value appear less (or more) important in terms of SHAP. We now provide an intuition through the example below:

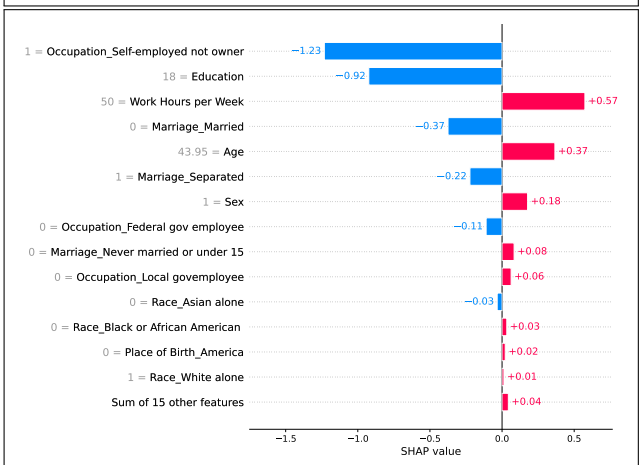
Example 1.1 (Motivating example). Consider a vendor — a financial institution that uses a binary classifier to approve loans (see Figure 1). Suppose that Ann applies for the loan and is incorrectly rejected (a false negative). The vendor would like to see if its model made this rejection decision based on Ann's age, and decides to compute feature importance using SHAP as part of its analysis. Indeed, when SHAP is run over the raw feature values, age appears to have high importance, likely because an age of 30 is comparatively low in the vendor's data. Worried about a potential lawsuit, the vendor attempts to generate a different explanation for the same classification outcome: they keep the classifier model fixed, but change the representation of age, "bucketizing" it into the ranges "below 50" and "50 and above". The vendor is relieved to see that this simple manipulation substantially diminishes the importance of age when explaining Ann's outcome.

Figure 2 demonstrates this very scenario for an individual in the ACS Income dataset, with the SHAP plot on the top showing an explanation on raw feature values, and the plot on the bottom showing an explanation after age is bucketized. Observe that the importance of age drops from rank 1 (most important) in Figure 2a to rank 5 (somewhat important) in Figure 2b, a decrease of 5 positions in terms of importance. Note also that, because of the efficiency property of Shapley values [29], a SHAP explanation can be used to reconstruct the outcome (by summing feature weights and returning the positive label if the sum is positive). In the example in Figure 2, both explanations are consistent with the classifier's prediction: they both predict that the individual would be rejected for the loan. If the vendor is worried about being challenged for using a protected feature like age to incorrectly reject applicants, it can look for an explanation that agrees with the prediction, but diminishes the importance of age. **In this paper, we refer to this kind of a manipulation as a data engineering attack.**

Contributions and roadmap. In this paper, we systematically investigate how SHAP-generated feature-based explanations are affected by simple data engineering choices, and how this sensitivity



(a) SHAP values before bucketization of age



(b) SHAP values after age is bucketized

Figure 2: SHAP values of features before (a) and after (b) bucketization for a fixed individual from the ACS Income dataset. Note that the classifier model and SHAP explainer remain fixed; the only modification to the individual's features from (a) → (b) was the bucketization of age. In (a), age is encoded as a continuous feature and is deemed most important by SHAP, with a rank of 1 and a feature weight of 0.99. In (b), the age feature was bucketized into 12 equi-width intervals over its active domain, using the median age to represent observations within each interval. This decreased the feature weight to 0.37, demoting age to the 5th rank in importance.

can be used to design a data engineering attack on SHAP. We discuss related work in Section 2, describe preliminaries in Section 3, and then present our contributions.

- As our first contribution, in Section 4, we empirically examine the impact of bucketization or binning on continuous features (e.g., age) and of different encoding methods on categorical features (e.g., race). We show that SHAP is highly sensitive to data engineering choices, with the importance of

age changing by as much as 20 rank positions in some cases. Further, in cases where age is the most important feature, its importance frequently drops by between 3 and 5 positions in the ranking. When using the race feature, we show that merging White and Asian individuals or White and Black individuals into a single category can reduce the importance of the race feature to nearly 0.

- As our second contribution, in Section 5, we design a feature engineering attack, demonstrating that sensitivity to seemingly benign data engineering choices can enable adversarial vendors to obscure the importance of protected features with minimal impact on predictions, allowing them to evade scrutiny without model retraining the model. For example, we demonstrate that our attack generally outperforms equi-width bucketization by substantially reducing the importance of the age feature without sacrificing explanation fidelity.

In Section 6, we highlight the need for a more robust framework for model explanations that evaluates not only accuracy and fairness, but also the impact of data engineering on local feature-based explanations. **Creating tools and guidelines to ensure that data engineering choices do not unduly influence reported feature importance should become standard practice in AI development.** We also acknowledge limitations and outline future directions. Finally, in Section 7, we summarize our insights.

All code is available at https://github.com/Aguno/Shap-Attack.

2 Related Work

The relationship between feature engineering and model explainability has been explored in previous works. For example, Ribeiro et al. [26] investigate how feature *selection* and engineering techniques impact model explanations, focusing on how feature importance is derived from global model behavior. Our research complements this approach by systematically demonstrating how bucketization and binning can impact model explanations.

Other studies have examined how data engineering operations like re-scaling, re-weighting, and re-sampling of features can either mitigate or exacerbate bias [21, 38]. These works demonstrate the unintended consequences of seemingly innocuous feature engineering decisions on AI systems. However, none of these studies explicitly address explainability, particularly in the context of SHAP, one of the most widely adopted explanation methods [5, 7].

More recently, Slack et al. [30] proposed an adversarial attack on SHAP by scaffolding a classifier that may be unfair on the input data, but appears fair on the rest of the data in terms of common statistical fairness criteria. In another recent approach, Baniecki and Biecek [2] generate synthetic data to manipulate SHAP. The authors use a genetic algorithm to manipulate the feature values towards certain SHAP targets. Finally, the Fool SHAP method by Laberge et al. [22] uses biased sampling to construct the background data such that the protected feature's importance is reduced, allowing the vendor to show false compliance during an algorithmic fairness audit. Here, an optimization problem is formulated to reduce the SHAP value of a feature without significantly altering the background data distribution relative to the original data. However, sampling data can be viewed as an explicit manipulation and may be prohibited. In comparison, our work does not focus on model fairness and only performs common data engineering manipulations that a data analyst could legitimately perform. We show that such seemingly benign operations can alter SHAP-computed feature importance and be used to design an attack.

Unlike prior work on predictive multiplicity [8], starting with the work on the "Rashomon Effect" [9], where different models may have comparable performance, our contribution is in showing that the same model may produce different explanations due to seemingly innocuous data preprocessing (i.e., feature engineering) choices. This exposes a critical weakness in SHAP that has not been shown in prior work and that, as we demonstrate, can enable an adversarial actor to manipulate explanations.

Finally, there are other works that interrogate explainability [3, 15, 16] (e.g., counterfactual explanations), but they do not identify data preprocessing vulnerabilities when using SHAP as we do.

3 Preliminaries

3.1 Local feature-based explanations with Shapley values

The Shapley value framework [29] is widely used to quantify local feature importance in predictive models [11, 23]. It does so by attributing a model's output for a given instance to individual input features, based on how their inclusion—alone or in combination with other features—affects the prediction.

The Shapley value for a feature i is formally defined as:

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left(f(S \cup \{i\}) - f(S)\right)$$

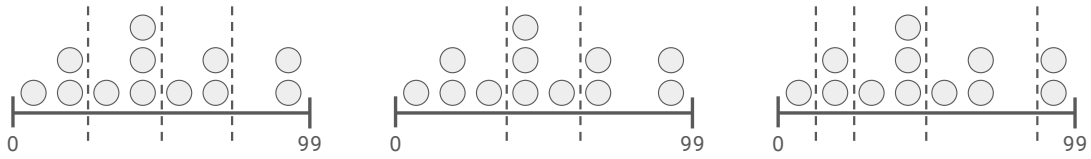
where N is the set of all players (features), $S \subseteq N \setminus \{i\}$ is a subset of players excluding player i, f is a value function defined on subsets of N (e.g., the expected model output conditional on the features in S), $\phi_i(f)$ is the Shapley value assigned to player i, representing their marginal contribution averaged over all possible coalitions.

In predictive classification, the players correspond to input features, and $f(S)$ is typically defined as the expected value of the model output conditional on the feature values in subset S. The vector of Shapley values assigned to an instance's features constitutes an *explanation*. Due to the efficiency property of Shapley values [29], the sum of these contributions exactly recovers the model output (minus a baseline), ensuring additive consistency.

By convention, a feature's importance is indicated by the absolute value of its weight (with higher values denoting greater importance), while the sign reflects the direction of its contribution toward a specific prediction (positive or negative). For example, in Figure 2(a), the *age* feature has a weight of -0.59: its high absolute value indicates importance, and the negative sign points toward the negative class label. Also by convention, visual explanations sort features in descending order of absolute weight, with the most important feature ranked first, regardless of sign, followed by the next most important, and so on.

In this work, we use SHAP [23], with its open-source implementation[2]. SHAP is used extensively by industry practitioners [18, 33], highlighting a critical need for its continued study, particularly

[2]https://pypi.org/project/shap/



(a) 4 equi-width buckets **(b) 3 equi-depth buckets** **(c) custom buckets per Section 5.1**

Figure 3: The figures above represent different ways to bucketize a continuous or high-dimensional ordinal feature like *age*. In each sub-figure, the feature is represented as a lineplot with values from 0 to 99. Each circle represents an age value for a single observation from the dataset, and the observations are the same across the sub-figures. For equi-width buckets (a), the domain is divided into buckets of equal width. For equi-depth buckets (b), buckets are created that all contain an approximately equal number of observations (this is equivalent to equi-width buckets over the percentile values of the feature). Sub-figure (c) shows how custom buckets may be created by the method described in Section 5.1 to manipulate the SHAP rank of a feature.

when it comes to surfacing vulnerabilities that could be abused by bad actors. We reveal the sensitivity of SHAP to feature engineering, adding to a robust body of work on studying the tool, surfacing issues, patching them, or expanding SHAP in other ways towards better implementation in practice [34, 36, 37].

Many researchers and practitioners use XAI for fairness auditing. Most relevant to our work, a high Shapley value for a protected feature like *age* or *race* suggests the feature has a significant influence on the classification outcome, which may raise ethical or legal concerns. Wexler et al. [37] present an approach for using SHAP and "what-if tools" to probe ML models for fairness. Vengroff [34] develops a toolkit based on SHAP that helps identify bias in both an ML system and the data used to train that system. Deck et al. [12] offer a nuanced perspective, noting that XAI tools are not an "ethical panacea," but are "one of many tools to approach the multidimensional, sociotechnical challenge of algorithmic fairness," along with other tools like those focused on bias auditing.

3.2 Representing features

In tabular data, features can be continuous, ordinal, or categorical. Continuous data can take any value within a range and is measurable, while ordinal data represents categories with a meaningful order or ranking. Categorical data consists of distinct groups or categories without an inherent ordering.

There are various ways to represent these data types in machine learning pipelines. For continuous or ordinal features, we may leave the data as is, or discretize the values using bucketization, where values are grouped into ranges. These buckets can then be one-hot encoded or treated as ordinal features. Categorical features can be encoded using methods like one-hot, ordinal, or target encoding. Additionally, scaling or normalization may be applied to adjust distributions or ranges of features.

Encoding continuous or ordinal features. Consider the feature *age*, which may be continuous or ordinal with integer values. For prediction or explanation purposes, *age* could be used in its raw form, bucketized into ordinal categories, or encoded into multiple buckets via one-hot encoding. Importantly, "upstream" feature representation choices affect the properties of the "downstream" model in a machine learning pipeline [31], influencing its accuracy [20], fairness, and explainability.

In this work, we focus on how bucketization [19]—grouping continuous or ordinal data into distinct value ranges—affects model predictions and SHAP explanations. We explore three methods for bucketizing continuous features, as illustrated in Figure 3. The first, *equi-width* in Figure 3a, creates buckets of equal feature value ranges. The second, *equi-depth* in Figure 3b, ensures each bucket contains an approximately equal number of data points (i.e., the buckets represent percentiles of the data). The third method, in Figure 3c, employs Bayesian Optimization to define bucket widths that optimize an adversarial objective, which we describe in Section 5.1. Note that the number of buckets can vary across methods.

Categorical features. We frame our discussion of encoding categorical features through the protected feature *race*. Whenever practitioners include *race* as a feature in machine learning models, they are implicitly making choices about how to encode that feature [6, 35]. For example, the ACS Income and ACS Public Coverage datasets[3], used in our experiments, include eight distinct race categories plus a null value. One approach is to represent all eight categories using one-hot encoding. However, due to small sample sizes in some categories, practitioners often create an "other" supercategory, leading to arbitrary groupings. Further, one category represents individuals identifying as "mixed race," but lacks information on which races they identify with. Intersectional encodings could also be considered.

In this work, we focus on six plausible encoding methods of the *race* feature, shown in Table 1. Four of these individuals into two race categories and two split them into three categories. While not exhaustive, these encodings are sufficient for exploring the sensitivity of SHAP to different race encodings.

Note: One-hot encoding can lead to counter-intuitive or redundant explanations. Consider, for example, a simple binary feature that denotes whether a person is a smoker. This feature would be represented by two one-hot-encoded features: *smoker=yes* (set to 0 for a non-smoker) and *smoker=no* (set to 1 for a non-smoker). An explanation of a medical diagnosis may redundantly assign high

[3] https://github.com/socialfoundations/folktables

importance to both *smoker=yes* and *smoker=no*: a person may be predicted to have a low likelihood of developing lung cancer both because they are a non-smoker (*smoker=no* is set to 1) and because they are not a smoker (*smoker=yes* is set to 0). An explanation may also be counter-intuitive, assigning high importance to *smoker=yes* being set to 0 and low importance to *smoker=no* being set to 1.

Returning to the one-hot representation of *race*: an explanation of a racially biased lending decision may assign a high positive weight to both *race=White* being set to 1 (the applicant is White), and to *race=Black* being set to 0 (the applicant is not Black). To estimate the total impact of race on the outcome, we sum up the weights of all one-hot-encoded components of the feature.

3.3 Experimental setup

We ran experiments over two real-world benchmark datasets with associated predictive tasks.

(1) ACS Income (Virginia, 2018) [13] is used to predict whether an individual's income is above $50K. It contains 46,144 observations comprised of 8 features, out of which 5 are categorical.
(2) ACS Public Coverage (Virginia, 2018) [13] is used to predict whether an individual is covered by public health insurance. It contains 25,524 observations comprised of 16 features, out of which 13 are categorical.

For both tasks, we treat *age* and *race* as protected features, and one-hot encode all categorical features, including *race*. We use XGBoost, a state-of-the-art ensemble classifier, with hyperparemeter tuning for overall accuracy.

Evaluating explanations. SHAP-based explanations can be evaluated using many different metrics [25, 27]. In this work, we use three metrics. An explanation is considered faithful if, for a given observation, the sum of its feature importance weights (before or after any preprocessing modifications) corresponds to the originally predicted outcome. The first metric, *fidelity*, refers to the proportion of observations for which a faithful explanation was generated, expressed as a ratio of those observations to the total number.

Additionally, we quantify how data representation choices affect both the absolute and the relative importance of a feature. We compute the *average SHAP value* (feature weight) and the *average rank* (by absolute value of feature weight) of the protected feature, and quantify the change in these metrics to compare explanations. Difference in average SHAP value quantifies the absolute change in feature importance, while difference in average rank quantifies the relative change in feature importance.

Sensitivity versus attack experiments. In the sensitivity experiments (Section 4), we alter the feature representation in both the training data (used to train the classifier) and the test data (used by the explainer). Here, we examine how explanations respond to feature bucketization, training a new model each time and applying the same representation to both the classifier and explainer.

In contrast, in the attack experiments (Section 5), we train the model on the original, non-bucketized data. We then keep the model fixed and only modify the representation of the inputs to the explainer. Here, we show that different explanations can be produced for the same model. In particular, this allows us to shift the apparent importance of a sensitive feature without retraining.

In both settings, the baseline applies no additional feature engineering beyond simple one-hot encoding for categorical features.

4 First Contribution: Sensitivity of SHAP to Feature Engineering

The experimental results reported in this section demonstrate that seemingly trivial feature engineering choices can lead to potentially harmful outcomes.

4.1 Continuous features (*age*)

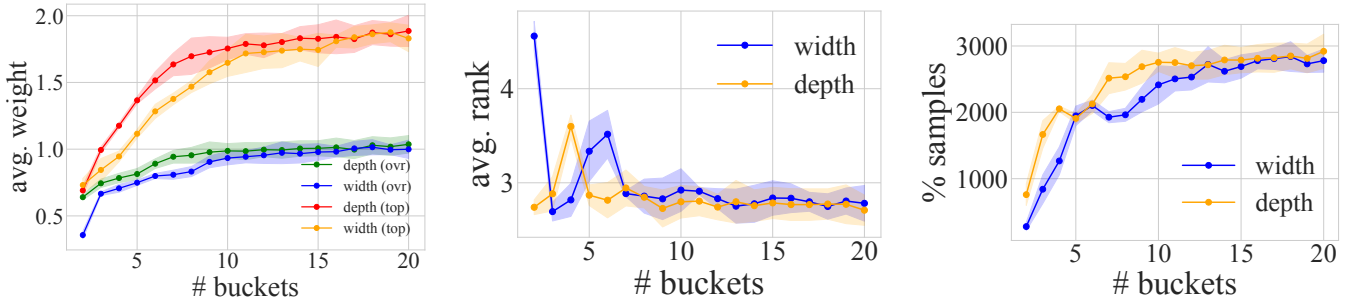
We evaluate SHAP's sensitivity to bucketization of the continuous *age* feature on the ACS Income dataset. Figure 4a shows that the feature importance of *age* increases with increasing number of buckets, both overall and for the observations for which *age* is the most important feature when the classifier is trained on raw (unbucketized) data. Figure 4b complements this result by showing that the average rank of *age* decreases (i.e., *age* moves closer to the top of the list) with increasing number of buckets. Figure 4c shows that the percentage of observations for which *age* is the most important feature increases substantially with increasing number of buckets, showing high sensitivity.

Intuitively, as the number of buckets increases, *age* becomes less obfuscated and thus plays a more important role in a model's prediction. Hence, even if bucketization itself is a standard operation, the importance of *age* , both in absolute terms (its weight) and in relative terms (its rank) can change drastically for a substantial portion of the observations. Similar to the sensitivity testing scenario, more buckets means the *age* is more fine-grained and has higher SHAP values across the entire dataset. However, the average SHAP value among the observations for which *age* is the most important feature remains similar regardless of the number of buckets.

Figure 5 shows the frequencies of rank changes of the *age* feature when using 5 or 10 equi-width or equi-depth buckets on ACS Income. For all scenarios, we observe that bucketization can have drastic impacts on a non-trivial portion of individuals, and that the importance of *age* may change by as much as 20 positions in the ranking.

In Figure 6, we vary the number of equi-width buckets and show the number of observations for which the rank of the *age* feature increased, decreased, or did not change. We also perform the same experiment for individuals for whom *age* is the highest-ranked (i.e., most important) feature, see Figure 11 in the Appendix. We can see in Figures 6a and 6b that the third bucket has the highest volatility. This demonstrates that bucketization can influence specific demographics more than others.

We also investigate how the SHAP ranking sensitivity changes based on the confusion matrix position of data points, i.e., whether an individual was a true positive (TP), false positive (FP), true negative (TN), or false negative (FN). This has important implications for fairness, as many fairness metrics are based on metrics derived from the confusion matrix of a model [28]. Interestingly, we find that changes are not uniform across these groups, as seen in Figure 12 in the Appendix, which quantifies the decrease in the importance of *age* relative to other features as the number of histogram buckets

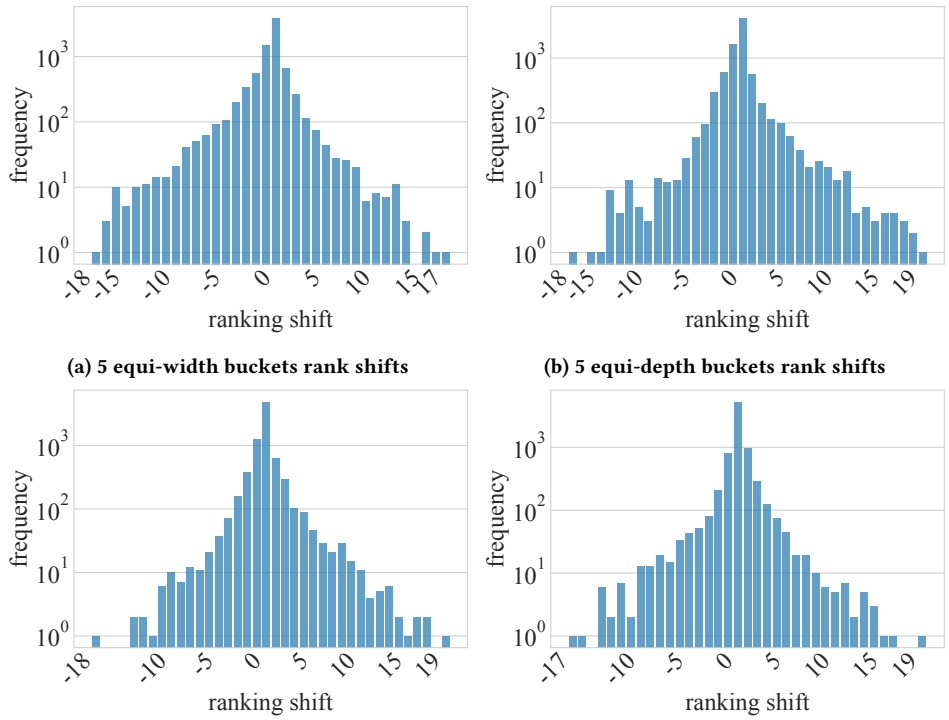


(a) Average SHAP value for the overall population (ovr) and subset of samples with *age* as the most important feature (top).

(b) Average rank of *age* in terms of feature importance.

(c) Percentage of samples for which *age* is the most important feature.

Figure 4: Number of buckets versus average feature importance weight and rank of the *age* feature on ACS Income. For each plot, we compare uniform (equi-width) and quantile (equi-depth) histograms.



(a) 5 equi-width buckets rank shifts

(b) 5 equi-depth buckets rank shifts

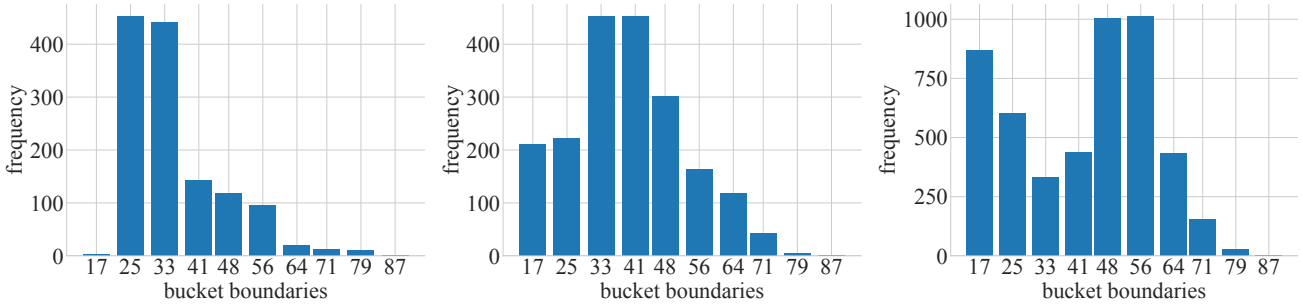
(c) 10 equi-width buckets rank shifts

(d) 10 equi-depth buckets rank shifts

Figure 5: Frequencies of rank shifts of the *age* feature when using (a) 5 equi-width, (b) 5 equi-depth, (c) 10 equi-width, or (d) 10 equi-depth buckets on ACS Income. The frequencies are shown on a log scale for better readability. Negative rank shifts represent "demotion" of *age* where the rank values increase (e.g., from Rank 1 to 10) making *age* less important, while positive shifts represent rank "promotion" where the importance of *age* increases.

increases. Depending on how much the adversary intends to lower the importance of a protected feature, the adversary can choose a specific number of buckets. For example, suppose that an adversary is using ACS Income and wants to make *age* look unimportant for false-negative observations using the data in Figure 12. Looking at Figure 12d, the adversary may decide to use 4 or 7 buckets where a large portion of the observations consider *age* to be unimportant ("rank 5 or below").

In summary, we showed that SHAP is highly sensitive to buckетization, with the relative importance of *age* changing by as much as 20 rank positions in some cases. Furthermore, when *age* is the



(a) demotions of the rank of *age* (b) promotions of the rank of *age* (c) the rank of *age* remains unchanged

Figure 6: Frequency plot for varying numbers of equi-width buckets on *age* on ACS Income. Bucket boundaries are shown on the x-axis.

Table 1: Bucketization strategies for the *race* feature. Base retains the original categories. OvR ("one vs. rest") creates two categories: one for the specified value and one for all others. The 2 buckets and 3 buckets strategies group values into 2 or 3 categories, respectively. Each row in the table corresponds to a strategy; buckets are separated by commas, and merged values are indicated with a plus sign (+).

Strategy	Buckets
Base	White, Black, Asian, Other
One vs. rest (OvR)	White, Rest Black, Rest Asian, Rest Other, Rest
2 buckets	White, Black + Asian + Other White + Black , Asian + Other White + Asian, Black + Other White, Other
3 buckets	White, Black, Asian + Other White, Asian, Black + Other

most important feature, its importance frequently drops by 3–5 positions in the ranking.

4.2 Categorical features (*race*)

We also evaluate SHAP's sensitivity to the representation of the categorical feature *race*. We apply two preprocessing strategies: one-vs-rest (OvR) and a combinatorial merging approach. In OvR, each *race* value is isolated while the remaining values are grouped, and a classifier is trained on the modified feature. The rest of the evaluation follows the procedure described earlier. The different merging strategies are shown and further described in Table 1.

Figures 7 (a)–(c) show how bucketization affects the *race* feature in the OvR case. In particular, we compare several settings: Base (no merging), White vs. others, Asian vs. others, Black vs. others, and Non-(White, Asian, Black) vs. others. Overall, the average SHAP value of the *race* feature decreases compared to Base for all settings as shown in Figure 7a. Even if the decrease seems minor for the White-versus-Rest strategy, the fraction of samples that have *race* as their most important feature drastically decreases, as shown in Figure 7b, demonstrating the effectiveness of bucketization. For Asian-versus-Rest, both of these values decrease significantly. We perform additional analyses in Figure 7c, which shows the average SHAP value of *race* only for observations with *race* as the most important feature. For Asian-versus-Rest, the average SHAP value is higher than that of Base, which means that these (few) observations are more likely to be discriminated against based on *race*.

In summary, we showed that SHAP is highly sensitive to bucketization for categorical features where the bucketization effectively shifts the importance from race to other features.

5 Second contribution: A feature engineering attack on SHAP

5.1 Deliberate manipulation of SHAP values

In Section 4, we demonstrated that post-hoc SHAP explanations are sensitive to the way features are encoded. Importantly, this sensitivity can be exploited to intentionally manipulate feature importance reported by SHAP.

Audit scenario. Similar to Laberge et al. [22], we consider a two-party audit scenario. The first party is a vendor that has full, white-box access to the data, the classifier model, and the data engineering and modeling pipelines. The vendor is able to make data engineering and modeling decisions and implement them into the respective pipelines. The second party is an auditor, who receives static copies of pre-processed data (i.e., the data that is prepared for modeling) and the model, to which it has black-box access. While the auditor cannot perform any engineering, it can generate feature-based explanations of model predictions for any observation in the dataset using SHAP. This allows the auditor to inspect how often the model appears to use protected features (according to SHAP explanations) to make classification decisions. Notably, the vendor has an adversarial goal: it wants to build models that make use of protected features, but it does not want those features to appear to have high importance according to SHAP when inspected by an auditor.

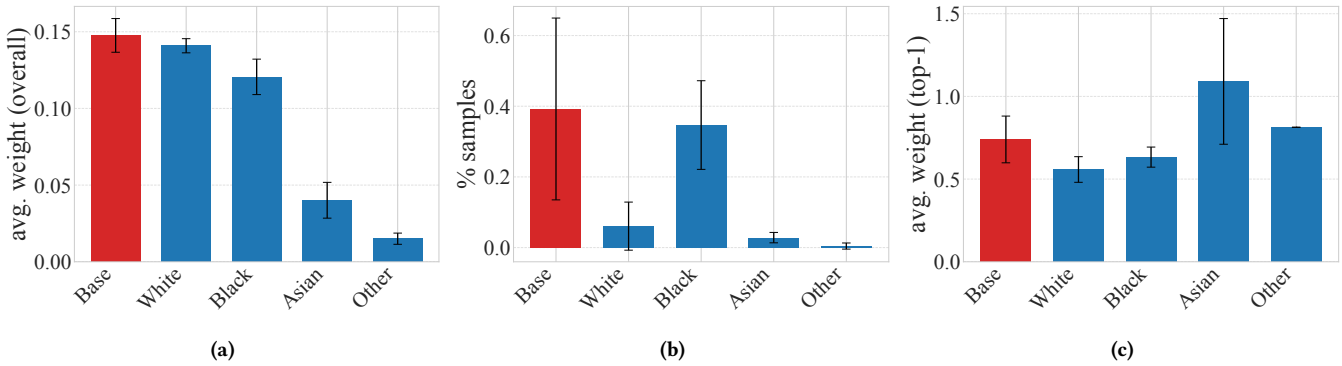


Figure 7: Effect on SHAP values using OvR bucketization for the *race* feature using the ACS Income dataset. We use each feature value (White, Asian, Black, or Other) as a category and assign the remaining values to a single category. (a) Average SHAP values of the *race* feature. (b) OvR setting versus the portion of observations where *race* is the most important feature. (c) OvR setting versus the mean SHAP value of *race* for the observations where *race* is the most important feature.

Formalization. The dataset D consists of $|D|$ training examples as tuples $(\mathbf{x}_i, y_i)$, where example i corresponds to the i-th individual in the dataset. Each $\mathbf{x} \in \mathbb{R}^n$, where n is the number of features, and that individual's outcome (or target) is given by $y \in \{0, 1\}$. We use a to denote the feature at a specific index $x_a \in \mathbf{x}$, which represents the protected feature for each individual. The dataset D is used to learn a machine learning classifier $f : \mathcal{X} \to [0, 1]$.

Recall from the description of the audit scenario, that the vendor is able to make data engineering decisions and implement them before training the classifier f. Let $\mathcal{T}$ be the set of all possible feature transformations that could be applied to the protected feature a. Each transformation $\tau \in \mathcal{T}$ is a function of the form $\tau : \mathcal{A} \to \mathcal{S}$, where $\mathcal{A}$ is the set of possible values for the protected feature a, and $\mathcal{S}$ is the set of all possible values for the transformed feature. We refer to $\tau(a)$ as the transformation of feature a. Then generally, the manipulation framework we propose uses the following objective:

$$\min_{\mathcal{T}} -\text{SHAP\_Rank}(a, f, D), \text{ s.t. } \lambda > \epsilon \tag{1}$$

where $\text{SHAP\_Rank}(a, f, D)$ is a function that returns the SHAP rank of feature a under model f and with data D, λ is the *fidelity* of the explanation (as defined in Section 3.3), and ϵ is a user-defined threshold for fidelity. The purpose of the fidelity constraint is to ensure that the feature transformation remains faithful to the original model (and, consequently, to the original explainer).

We now describe a particular instantiation of this framework for manipulating continuous features via the bucketization feature transformation. Suppose that $a \in \mathbb{R}^{\geq 0}$ is a continuous feature. We can define the feature transformation $\tau_K : \mathbb{R}^{\geq 0} \to \{0, 1, \ldots, k\}$ that discretizes a into k buckets in the following way:

$$\tau_K(a) = \begin{cases} 0, & b_0 < a < b_1 \\ 1, & b_1 < a < b_2 \\ \vdots & \\ \text{K}, & b_{k-1} < a < b_k \end{cases} \tag{2}$$

Where $b_0 < b_1 < \cdots < b_k$ are the upper-lower bounds for each bucket. Based on the choices for the upper-lower bounds used to define the cases of τ_k, applying the transformation $\tau_k(a)$ can be used to induce a particular set of k partitions over the data D. Let $\mathcal{P}_k$ be the set of all k-partitions over D. Note that this occurs upstream in our machine learning pipeline, so f will always be trained on data with the transformed feature. Then our objective is:

$$\min_{k \in \{0,1,\ldots,k\}, P \in \mathcal{P}_{\|}} -\text{SHAP\_rank}(a, f, D), \text{ s.t. } b_0 < \cdots < b_k, \lambda > \epsilon \tag{3}$$

We can solve this problem using Bayesian Optimization [24], which is commonly used to solve black-box optimization problems by constructing a posterior distribution of Gaussian functions that best describe the unknown function to optimize. Bayesian Optimization is particularly appropriate for this problem because it is effective when the objective function is expensive to evaluate, as is often the case with the function SHAP_rank.

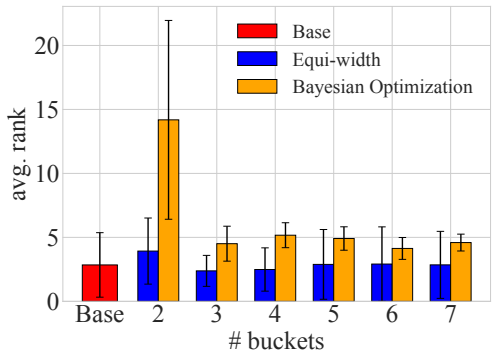
5.2 Bucketization attack experiments

We now perform the data engineering attacks described in the previous section, which use Bayesian Optimization (BO) to tune bucket boundaries when using *age*. Compared to equi-width bucketization, we observe sharper changes in the SHAP rank of *age*, but also a more substantial decrease in fidelity. (Although not shown here, the comparison with equi-depth bucketization is very similar.)

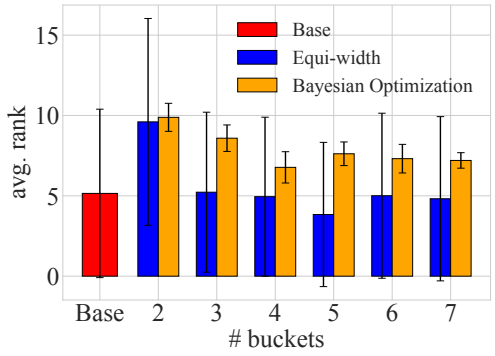
The experimental results in this section are computed using 5-fold cross-validation, with metrics averaged across folds. We report results for both ACS Income and ACS Public Coverage.

For BO, we tune four bucket boundaries between fixed minimum and maximum values. Using *age*, we set the minimum and maximum values to be 17 and 94 years old, respectively. We then perform 300 iterations where the objective function is the same as Equation (3). For the fidelity constraint, we require that the fidelity is at least as good as that in the equi-width setting. As a result, the BO attack significantly outperforms Base (no bucketization) and mostly outperforms the equi-width setting. We conclude that BO thus may be used to hide the contributions of *age* on model predictions.

Figure 8 shows how the importance rank of *age* changes when varying the number of buckets (and thus the number of BO parameters) using ACS Income and Public Coverage. As the number



(a) ACS Income ranks



(b) ACS Public Coverage ranks

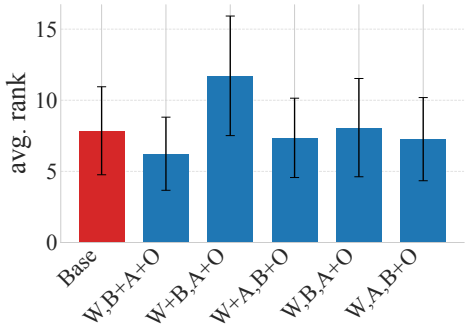
Figure 8: SHAP value rank trends of the *age* feature on ACS Income and ACS Public Coverage. Higher average rank means that feature has lower importance. Fidelity under attack is at least as high as under equi-width bucketization in each case.

of buckets increases, our attack consistently shows comparable or higher ranks compared to using the Base average rank, although the gap decreases. The gap decrease is due to the difficulty in solving high-dimensional BO problems. At the same time, the constraint on fidelity ensures that our attack is always at least as meaningful as the equi-width bucketization. Table 2 shows the fidelity values of the attacks on *age*, with bucketization of Figure 8. Even when the average rank of *age* drops dramatically, the fidelity values are at least 88%.

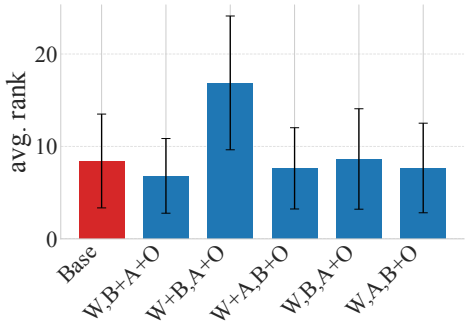
When using *race*, BO cannot be performed on categorical values, so we instead combine races for the bucketization attack as in Table 1, but consider more combinations. Figure 9 shows the how the rank of *race* changes on ACS Income and ACS Public Coverage. For the {White+Black, Asian+Other} bucketization on both tasks, the rank value of *race* is significantly higher (i.e., *race* is far less important) compared to the Base strategy. We suspect that *race=White* by itself is a strong signal when making predictions, but when combined with another race into the same bucket, its importance is diluted. Table 3 shows the fidelity values of the attacks on *race*

Table 2: Fidelity values of bucketizations on *age* for ACS Income and ACS Public Coverage.

Buckets	ACS Income	ACS Public Coverage
2	88.60 ± 0.09	94.38 ± 0.83
3	91.37 ± 0.26	95.99 ± 0.13
4	92.39 ± 0.21	96.95 ± 0.03
5	93.15 ± 0.13	96.85 ± 0.04
6	94.67 ± 0.08	97.25 ± 0.03
7	94.59 ± 0.04	97.43 ± 0.03



(a) ACS Income



(b) ACS Public Coverage

Figure 9: SHAP value rank trends of the *race* feature using the ACS Income and ACS Public Coverage datasets. Simply combining two races into a single bucket dramatically increases the average SHAP value rankings.

for the bucketizations of Figure 9. At least 98% of all explanations have perfect fidelity (i.e., reconstruct the true outcome).

In summary, we showed that, on ACM Income and Public Coverage, our BO attack mostly outperforms equi-width bucketization in terms of increasing the SHAP ranks of *age* without compromising fidelity. Our attack is also effective when using *race* where we combine races for bucketization. In particular, on ACS Public Coverage, combining White and Black individuals into a single category brings the importance of *race* close to zero.

Table 3: Fidelity value of bucketizations on *race* for ACS Income and ACS Public Coverage.

Buckets	ACS Income	ACS Public Coverage
W, B+A+O	98.56 ± 0.48	99.57 ± 0.09
W+B, A+O	98.10 ± 0.10	98.00 ± 0.19
W+A, B+O	99.16 ± 0.09	99.65 ± 0.10
W, B, A+O	99.72 ± 0.01	99.67 ± 0.10
W, A, B+O	99.65 ± 0.05	99.74 ± 0.09

6 Discussion

We note that any sociotechnical system that incorporates AI/ML is not a monolith: it impacts many stakeholders, including practitioners, compliance officers, auditors, affected individuals, and society at large. In this section, we state the implications of our work for practitioners and auditors seeking to ensure the responsible implementation of AI/ML systems.

Our experimental results show that SHAP, one of the most widely-used post-hoc explanation methods, is highly sensitive to upstream data engineering decisions. Furthermore, we demonstrate that a principled approach using Bayesian Optimization can exploit this sensitivity to manipulate SHAP. Below, we discuss the practical implications of this finding.

Practitioners. Our results further highlight the importance of thoughtful feature engineering when building machine learning pipelines, and of assessing how those decisions may impact the accuracy, fairness, and explainability of the systems being built and deployed [31]. Kamiran and Calders [21] and Zafar et al. [38] showed that varying feature representations can affect the fairness of a machine learning classifier. We expand on this result, showing that these effects extend to model explainability—particularly when using the post-hoc explanation method SHAP. Notably, this recommendation can also be framed as an opportunity. In line with earlier framing by Stoyanovich et al. [32], Bell et al. [4] showed that the explainability of a system depends on the context of use, the data, the underlying model type, the stakeholders, and the specific questions a human is trying to answer about the system. Practitioners could use these factors to inform how they encode features to *improve* the explainability of their pipelines.

Auditors. In this work, we showed how seemingly innocuous decisions—such as the bucketization of the *age* feature—can be used to manipulate the SHAP explainer. Our experiments demonstrate that one can successfully reduce a feature's apparent importance while *minimally impacting* the explanation's fidelity to the original model's outcomes. Importantly, such manipulation could be exploited by adversarial actors to obscure discrimination in their models. While developing a technical defense is beyond the scope of this paper, we offer the following recommendation to auditors: governance and audit frameworks for machine learning systems should be expanded to account for data engineering decisions.

Limitations. This work has several limitations. While we do perform an attack on categorical features, it relies on predefined, semantically meaningful groupings. More work is needed to develop methods for semi-automatically exploring the combinatorial space of groupings, particularly for high-cardinality features. A naïve approach—exhaustively enumerating all bucketings to find the one that most impacts the importance rank of *race* while preserving model fidelity—is computationally infeasible. Future work should develop principled techniques for navigating this space efficiently.

Another limitation is that we consider sensitive features in isolation. Our experiments explore the sensitivity of explanations to feature representations for *age* and *race* separately, with targeted attacks designed independently for each. In practice, sensitive attributes often interact—and their combined effects can influence model behavior and interpretability. Future work should develop holistic methods that jointly consider multiple sensitive features for explainability and robustness.

7 Conclusions and future work

We explored how common data engineering techniques affect local feature-based explanations from methods like SHAP, and showed that subtle preprocessing choices can significantly alter explanations. We also introduced a feature engineering attack that hides the importance of protected features with minimal impact on predictions. Finally, we called for more robust explanation frameworks that consider not only accuracy and fairness, but also the influence of data engineering—especially in transparency-critical settings.

While our work highlighted SHAP's sensitivity to common preprocessing operations and their potential for misuse, the same insights can be applied constructively. Instead of designing attacks, similar techniques can inform data engineering choices that improve the explainability and robustness of classification decisions.

For example, identifying preprocessing choices that consistently elevate the importance of semantically meaningful features may lead to models that better reflect human reasoning and are easier to audit. These insights could also support the development of classifiers with greater control over the distribution of feature importance. Future work should extend this approach to sets of features—continuous, categorical, or mixed—and investigate how thoughtful data engineering can help align model explanations with stakeholder expectations. Finally, future research should examine the impact of more complex data engineering transformations on SHAP explanations.

Acknowledgments

This work was supported in part by the NYU-KAIST Partnership and by the Institute of Information & Communications Technology Planning & Evaluation (IITP) with a grant funded by the Ministry of Science and ICT (MSIT) of the Republic of Korea in connection with the Global AI Frontier Lab International Collaborative Research. (No. RS-2024-00509258 and No. RS-2024-00469482). This work was also supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1A2C2004382), the US National Science Foundation (NSF) Awards No. 2312930 and 2326193, and by NSF GRFP DGE-2234660.

References

[1] Sarah Alwarthan, Nida Aslam, and Irfan Ullah Khan. 2022. An explainable model for identifying at-risk student at higher education. *IEEE Access* 10 (2022), 107649–107668.

[2] Hubert Baniecki and Przemyslaw Biecek. 2022. Manipulating SHAP via Adversarial Data Perturbations (Student Abstract). In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 12907–12908.

[3] Solon Barocas, Andrew D. Selbst, and Manish Raghavan. 2020. The hidden assumptions behind counterfactual explanations and principal reasons. In *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*. ACM, 80–89.

[4] Andrew Bell, Ian Solano-Kamaiko, Oded Nov, and Julia Stoyanovich. 2022. It's Just Not That Simple: An Empirical Study of the Accuracy-Explainability Trade-off in Machine Learning for Public Policy. In *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*. ACM, 248–266.

[5] Vaishak Belle and Ioannis Papantonis. 2021. Principles and practice of explainable machine learning. *Frontiers in big Data* 4 (2021), 688969.

[6] Sebastian Benthall and Bruce D Haynes. 2019. Racial categories in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*. 289–298.

[7] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José MF Moura, and Peter Eckersley. 2020. Explainable machine learning in deployment. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 648–657.

[8] Emily Black, Manish Raghavan, and Solon Barocas. 2022. Model Multiplicity: Opportunities, Concerns, and Solutions. In *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*. ACM, 850–863.

[9] Leo Breiman. 2001. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statist. Sci.* 16, 3 (2001), 199–231.

[10] Ahmad Chaddad, Jihao Peng, Jian Xu, and Ahmed Bouridane. 2023. Survey of explainable AI techniques in healthcare. *Sensors* 23, 2 (2023), 634.

[11] Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*. IEEE Computer Society, 598–617.

[12] Luca Deck, Jakob Schoeffer, Maria De-Arteaga, and Niklas Kühl. 2024. A Critical Survey on Fairness Benefits of Explainable AI. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency, FAccT 2024, Rio de Janeiro, Brazil, June 3-6, 2024*. ACM, 1579–1595.

[13] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring Adult: New Datasets for Fair Machine Learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. 6478–6490.

[14] Benjamin Fresz, Elena Dubovitskaya, Danilo Brajovic, Marco F Huber, and Christian Horz. 2024. How should AI decisions be explained? Requirements for Explanations from the Perspective of European Law. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, Vol. 7. 438–450.

[15] Leif Hancox-Li. 2020. Robustness in machine learning explanations: does it matter?. In *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*. ACM, 640–647.

[16] Leif Hancox-Li and I. Elizabeth Kumar. 2021. Epistemic values in feature importance methods: Lessons from feminist epistemology. In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*. ACM, 817–826.

[17] Xuanxiang Huang and Joao Marques-Silva. 2024. On the failings of Shapley values for explainability. *International Journal of Approximate Reasoning* (2024), 109112.

[18] Javier Camacho Ibáñez and Mónica Villas Olmeda. 2022. Operationalising AI ethics: how are companies bridging the gap between practice and principles? An exploratory study. *AI Soc.* 37, 4 (2022), 1663–1687.

[19] Yannis E. Ioannidis. 2003. The History of Histograms (abridged). In *Proceedings of 29th International Conference on Very Large Data Bases, VLDB 2003, Berlin, Germany, September 9-12, 2003*. Morgan Kaufmann, 19–30.

[20] Branka Jokanovic, Moeness G Amin, and Fauzia Ahmad. 2016. Effect of data representations on deep learning in fall detection. In *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*. IEEE, 1–5.

[21] Faisal Kamiran and Toon Calders. 2011. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.* 33, 1 (2011), 1–33.

[22] Gabriel Laberge, Ulrich Aïvodji, Satoshi Hara, Mario Marchand, and Foutse Khomh. 2023. Fooling SHAP with Stealthily Biased Sampling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

[23] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 4765–4774.

[24] Jonas Mockus. 1974. On Bayesian Methods for Seeking the Extremum. In *Optimization Techniques, IFIP Technical Conference, Novosibirsk, USSR, July 1-7, 1974 (Lecture Notes in Computer Science, Vol. 27)*. 400–404.

[25] Resmi Ramachandranpillai, Ricardo Baeza-Yates, and Fredrik Heintz. 2023. FairXAI - A Taxonomy and Framework for Fairness and Explainability Synergy in Machine Learning. doi:10.36227/techrxiv.24463945.v1

[26] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. AAAI Press, 1527–1535.

[27] Marko Robnik-Sikonja and Marko Bohanec. 2018. Perturbation-Based Explanations of Prediction Models. In *Human and Machine Learning - Visible, Explainable, Trustworthy and Transparent*. Springer, 159–175.

[28] Pedro Saleiro, Benedict Kuester, Loren Hinkson, Jesse London, Abby Stevens, Ari Anisfeld, Kit T Rodolfa, and Rayid Ghani. 2018. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577* (2018).

[29] Lloyd S Shapley et al. 1953. A value for n-person games. (1953).

[30] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. In *AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020*. ACM, 180–186.

[31] Julia Stoyanovich, Serge Abiteboul, Bill Howe, H. V. Jagadish, and Sebastian Schelter. 2022. Responsible data management. *Commun. ACM* 65, 6 (2022), 64–74.

[32] Julia Stoyanovich, Jay J. Van Bavel, and Tessa V. West. 2020. The Imperative of Interpretable Machines. *Nature Machine Intelligence* 2 (2020), 197–199. doi:10.1038/s42256-020-0171-8

[33] Umm-e-Habiba, Mohammad Kasra Habib, Justus Bogner, Jonas Fritzsch, and Stefan Wagner. 2025. How do ML practitioners perceive explainability? an interview study of practices and challenges. *Empir. Softw. Eng.* 30, 1 (2025), 18.

[34] Darren Erik Vengroff. 2024. Impact Charts: A Tool for Identifying Systematic Bias in Social Systems and Data. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency, FAccT 2024, Rio de Janeiro, Brazil, June 3-6, 2024*. ACM, 1187–1198.

[35] Angelina Wang, Vikram V Ramaswamy, and Olga Russakovsky. 2022. Towards intersectionality in machine learning: Including more identities, handling underrepresentation, and performing evaluation. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. 336–349.

[36] David S. Watson. 2022. Rational Shapley Values. In *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*. ACM, 1083–1094.

[37] James Wexler, Mahima Pushkarna, Sara Robinson, Tolga Bolukbasi, and Andrew Zaldivar. 2020. Probing ML models for fairness with the what-if tool and SHAP: hands-on tutorial. In *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*. ACM, 705.

[38] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. 2019. Fairness Constraints: A Flexible Approach for Fair Classification. *J. Mach. Learn. Res.* 20 (2019), 75:1–75:42.

[39] Leid Zejnilovic, Susana Lavado, Carlos Soares, Íñigo Martínez De Rituerto De Troya, Andrew Bell, and Rayid Ghani. 2021. Machine learning informed decision-making with interpreted model's outputs: A field intervention. In *Academy of Management Proceedings*, Vol. 2021. Academy of Management Briarcliff Manor, NY 10510, 15424.

A Local explanations

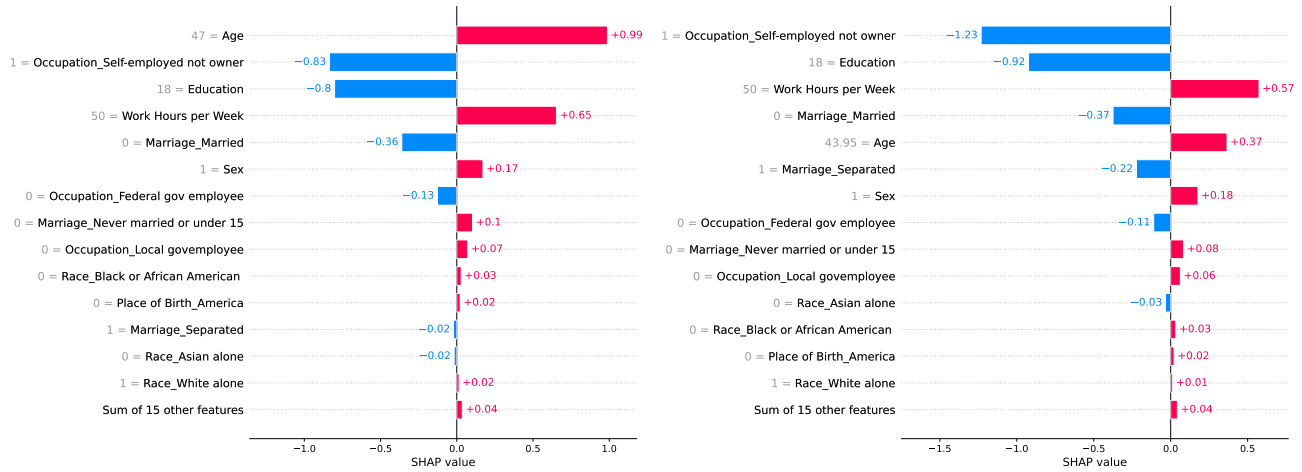
Continuing from Section 1, we perform individual analyses to understand how the SHAP ranking of age can decrease for a sample for the ACS Public Coverage dataset. Figure 10 shows how the age's SHAP ranking decreases for an individual. In (a), the age of the first individual is 22, which can be viewed as a small value. Once the age is bucketized into a range of ages, the value 22 is no longer special, and the age's SHAP value becomes negligible.

B Further analyses on ACS Income

Continuing from Section 4.1, we focus on individuals for whom *age* is the highest-ranked (i.e., most important) feature and show the number of observations for which the rank of the *age* feature increased, decreased, or did not change in Figure 11. Only a few samples in the third bucket have age as the first rank.

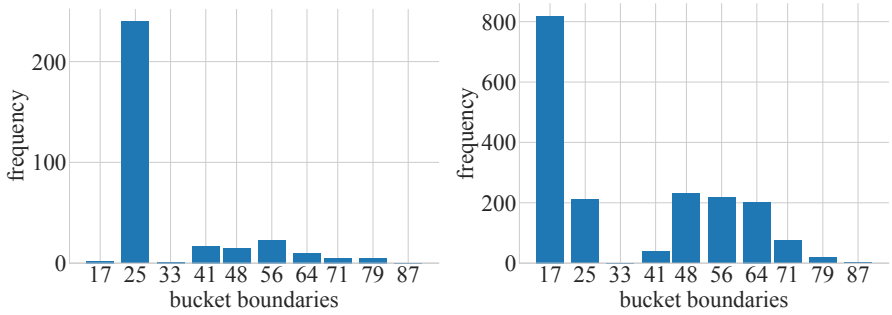
C SHAP Ranking Sensitivity

Continuing from Section 4.1, we investigate how the SHAP ranking sensitivity changes based on the confusion matrix position of data points. Figure 12 shows that changes are not uniform across these groups.



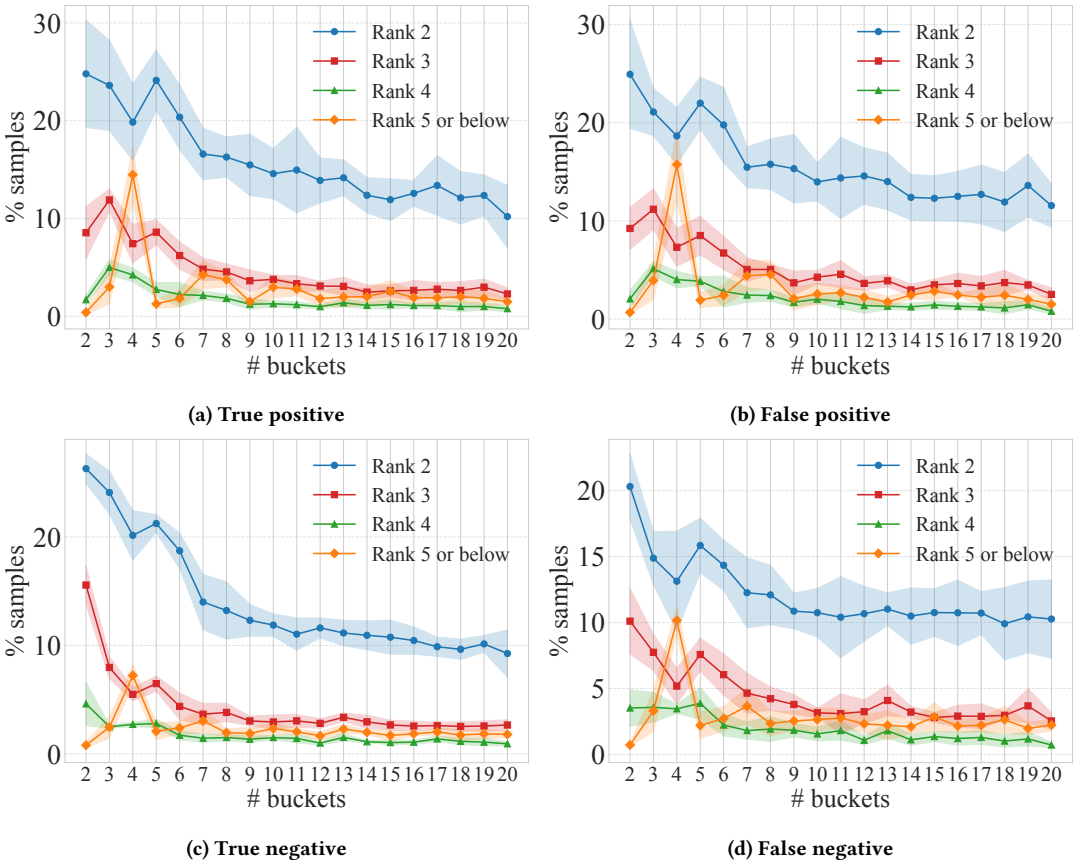
(a) SHAP values before bucketization **(b) SHAP values after bucketizing age into 10 equally sized buckets**

Figure 10: SHAP values of features before (a) and after (b) bucketization for a fixed observation from ACS Public Coverage.



(a) demotions in the rank of *age* **(b) no change in the rank of *age***

Figure 11: Frequency plot for different buckets of the *age* feature on ACS Income. Bucket boundaries are shown on the x-axis. (a) Number of observations where *age* was the most important feature before bucketization, and where the relative importance of *age* decreased (rank increased) after bucketization. (b) Number of observations where *age* was the most important feature both before and after bucketization.



(a) True positive

(b) False positive

(c) True negative

(d) False negative

Figure 12: SHAP ranking changes for the four outcomes where age was the highest-ranked feature as we change the number of buckets in the ACS Income dataset.