Looking for Trouble: Analyzing Classifier Behavior via Pattern Divergence

# Looking for Trouble: Analyzing Classifier Behavior via Pattern Divergence

Eliana Pastor
Politecnico di Torino, Italy
eliana.pastor@polito.it

Luca de Alfaro
UC Santa Cruz, USA
luca@ucsc.edu

Elena Baralis
Politecnico di Torino, Italy
elena.baralis@polito.it

## ABSTRACT

Machine learning models may perform differently on different data subgroups, which we represent as itemsets (i.e., conjunctions of simple predicates). The identification of these critical data subgroups plays an important role in many applications, for example model validation and testing, or evaluation of model fairness. Typically, domain expert help is required to identify relevant (or sensitive) subgroups.

We propose the notion of divergence over itemsets as a measure of different classification behavior on data subgroups, and the use of frequent pattern mining techniques for their identification. A quantification of the contribution of different attribute values to divergence, based on the mathematical foundations provided by Shapley values, allows us to identify both critical and peculiar behaviors of attributes. Extended experiments show the effectiveness of the approach in identifying critical subgroup behaviors.

## CCS CONCEPTS

• **Information systems → Data mining**; • **Mathematics of computing → Exploratory data analysis**.

## KEYWORDS

classifier validation; fairness in machine learning; Shapley value; bias detection; machine-learning model debugging

## 1 INTRODUCTION

The evaluation of classification models generally focuses on overall performance, estimated over all the data. However, the overall estimation provides no indication if differences in the model behavior exist across subsets of data.

In this paper, we introduce the notion of *divergence* to estimate the different classification behavior in data subgroups with respect to the overall behavior. A subgroup is a subset of the data characterized by a set of attribute values, also referred to in the paper as *patterns* or *itemsets*. The *divergence* of the subgroup measures the difference in statistics such as false positive and false negatives on the subgroup compared to the entire dataset.

The identification of data subgroups in which a machine learning model performs differently is relevant in many applications such as model validation and testing [8], model comparison [8, 15], error analysis [26] and evaluation of model fairness [7, 8]. Divergence exploration can reveal in which subgroups a model performs poorly, helping data scientists in model debugging. Moreover, the analysis of divergent subgroups provides indication of model behavior and hence can be a tool for model understanding. It may also reveal if divergence from the overall behavior occurs for sensitive attributes.

| Itemset | |
|---|---|
| age=25-45, #prior>3, race=African-Am, sex=Male | $FPR$=0.308 |
| age>45, race=Caucasian | $FNR$=0.929 |
| race=African-Am, sex=Male | $FPR$=0.150 |
| race=African-Am, sex=Male, #prior>3 | $FPR$=0.267 |
| race=African-Am, sex=Male, #prior=0 | $FPR$=0.097 |

**Table 1: Example of patterns in the *COMPAS* dataset, along with their FPR or FNR. The overall FPR and FNR are 0.088 and 0.698.**

As an example, consider the *COMPAS* dataset [3] containing demographic information and criminal history of defendants. For each criminal defendant, the *COMPAS* score of recidivism risk assesses the defendant's likelihood of committing another offense in a period of two years. *COMPAS* scores are determined by a proprietary algorithm and we do not have access to its inner workings. We compare the predicted recidivism rate with the actual one. The overall false positive (FPR) and false negative (FNR) rates are 0.088 and 0.698, where the positive class indicates being recidivist. However, the rates are different when subgroups are considered (see Table 1). The subgroup identified by pattern (*age=25-45, #prior>3, race=African-American, sex=Male*) has a FPR equal to 0.308. Instances belonging to this data subset tend to be wrongly assigned to high risk of recidivism with a rate higher than the dataset overall. On the other hand, the FNR of the pattern (*age>45, race=Caucasian*) is 0.929, indicating that caucasians with age greater than 45 tend to be wrongly labelled with a low risk of recidivism more than in the dataset overall.

Several existing approaches that explore differences in subgroup performance [5, 15] require users to specify the attributes or attribute values of interest. This requires human expertise, and hinders the identification of unexpected and previously unknown critical subgroups. Instead, our approach belongs to automatic subgroup detection techniques. Differently from existing methods [7, 8, 14], we introduce algorithms that allow us to efficiently estimate the divergence in classifier behavior for all subgroups with the condition of being sufficiently represented in the dataset. Furthermore, our approach is *model agnostic*. Hence, it treats the classification model as a black box, without knowledge of its internal working.

The contributions of this paper are both theoretical and algorithmic, and are implemented in the DivExplorer tool [20]. On the theoretical side, we introduce the notion of *divergence* over itemsets, and we provide a way of measuring its statistical significance that is informed by Bayesian statistics. Next, we introduce the use (and generalization) of Shapley values to analyze the contribution of atomic patterns (single-attribute patterns such as *sex=Male*) both within larger patterns, and overall in the dataset.

Recall our example dataset *COMPAS*. Once we determine that the pattern (*age=25-45, #prior>3, race=African-American, sex=Male*) has high divergence, one might wonder about the relative contribution to divergence of the four members of the pattern, to which we refer as *items*. The problem of measuring individual contributions to a collective outcome has been considered in game theory, and the celebrated notion of Shapley value [24] answers precisely this question. We propose to apply Shapley values to divergence analysis. This will enable us to determine that the item contributing most to the divergence is *#prior>3*, followed by *race=African-American*, with *sex=Male* giving only marginal contribution (see Figure 2).

In a dataset such as *COMPAS*, one is often interested not only in analyzing particular patterns where divergence is high (of which there are many; see Table 2 and Figure 7), but also in understanding what is the role of each item in leading to divergence across all patterns. The simplest approach is to measure the *individual* divergence of the item in isolation. We propose to extend the notion of Shapley value to measure the contribution to divergence of each item, in the context of all other items. The result, which we call *global* divergence, measures how much an item contributes to increasing the divergence when added to patterns. We prove that our generalization satisfies the fundamental axioms of Shapley values, stated in our modified context. Individual and global item divergence have different properties. We argue that among the two, global divergence is often a better measure of the effect of an item on divergence. In fact, individual item divergence is often unable to capture divergence that results from the association of multiple items. Global item divergence captures such associated contributions, due to its basis in the team-analysis underlying Shapley values (see Figure 4).

The second class of contributions of the paper are algorithmic, and they rest on the realization that item and pattern divergences can be computed efficiently by augmenting well known *frequent pattern mining* algorithms [25]. This enables us to efficiently compute the divergence of *all* patterns whose support (frequency in the dataset) is above a specified threshold. A boundary on support is reasonable, as patterns with small support are less relevant, due to the few data instances they affect, and measurements on them are more affected by statistical fluctuations. We provide experimental results on multiple real-world datasets showing that our algorithms enable full exploration up to the support threshold typically in a matter of seconds (see Figure 6).

The need for a complete exploration derives from the consideration that the considered metrics to estimate differences in classification performance are not monotone. Therefore, from the divergence of a pattern we cannot make assumptions on the divergence of the patterns that are contained in it. Let $G$ and $H$ be two data subsets of dataset $D$, with $H \subset G \subset D$. The divergence of $H$ can be higher, equal or lower than that of its superset $G$.

Previous approaches, such as [8], adopted heuristics to prune the search, stopping when divergence reaches sufficient values, or when a prescribed number of divergent patterns has been found. Our complete exploration not only enables the measurement of metrics such as global divergence, but it also makes visible phenomena that might be invisible under pruning. One of the most intriguing is the notion of *corrective items*, which are items that *reduce* the divergence when added to patterns.

Summarizing, our main contributions, implemented in the DivExplorer tool, are as follows.

- *Divergence.* We introduce the notion of divergence and we characterize each relevant data subgroup by its divergence. We estimate the local contribution of each attribute value to the subgroup divergence through the notion of Shapley value.
- *Global item divergence.* We generalize the notion of Shapley value to estimate the global contribution to divergence of each attribute value.
- *Corrective attribute value.* We introduce the notion of corrective attributes values, which tend to renormalize the divergence.
- *Bayesian treatment of statistical significance.* We propose a way to measure the statistical significance of the results that can be applied to black-box classifications.
- *Divergence computation algorithm.* We propose an efficient algorithm to automatically extract and explore all divergent subgroups with sufficient support.

After a discussion of related work, Section 3 gives our main definitions of items, itemsets, divergence and statistical significance. Section 4 uses the notion of Shapley value to define local and global item contribution to divergence. Section 5 introduces our algorithm, and Section 6 presents experimental results on several real-world datasets, reporting running times and divergence results.

## 2 RELATED WORK

Data grouping solutions often rely on domain experts to identify the relevant subgroups of interest. In TensorFlow Model Analysis (TFMA) [2, 5], the users specify the input features on which to partition the data used for classification performance evaluation. MLCube [15] is an interactive and explorative visualization technique that estimates aggregate statistics and performance metrics over subgroups defined by users. For model fairness, the diagnosis concentrates on evaluating if results are dependent on certain sensitive or protected attributes (e.g., gender, ethnicity, sexual orientation) [6, 23]. Several works [10, 11, 16, 19] consider fairness for an intersection of multiple sensitive attributes, known as intersectional fairness. For intersectional fairness diagnosis, protected attributes are generally specified a priori [19].

The identification of problematic attributes might not be straightforward. Hence, to limit the required human intervention, several automatic subgroup detection techniques to identify interesting data subgroups have been recently proposed [7, 8, 14, 26]. These works are close to our approach.

Slice Finder [8, 9] is an interactive framework that automatically identifies large data slices in which the model performs poorly, defined as "problematic" slices. A top-down lattice search finds top-k slices of interest in a breadth-first traversal. The lattice search is controlled by statistical techniques that measure the significance and magnitude of performance discrepancy on subgroups. To identify large and interpretable subgroups, the breadth-first traversal does not proceed if the considered data group is already statistically significant and the model performs sufficiently poorly. However, since the metrics used for assessing model performance on subgroups are typically non-monotone, the grade of discrepancy of a group

provides no indication on the behavior of its super/sub-groups. We propose a more thorough exploration of the lattice, considering all data slices, identified by itemsets having support greater than a given threshold. Frequency constraints allow us to identify data subsets (i.e., slices) large enough to be of interest. To improve interpretability, concepts of coalition game theory are exploited to characterize subgroup divergence. Since the work in [8] is the closest to DivExplorer, a more detailed comparison is performed later in the paper.

FairVIS [7] is a visual analytics system to discover intersectional bias in machine learning models that integrates a clustering-based generation technique to identify subgroups. Groups with significant statistical similarity are then described by a few dominant features using feature entropy. Performance metrics are evaluated on the identified clusters. Differently from FairVis, we identify data subgroups directly by slicing attribute domains. As a result, identified subgroups are already interpretable.

Errudite [26] exploits data grouping for NLP error analysis. A domain specific language is proposed to systematically group instances. Despite the system suggestions and guidance to formulate group queries, data grouping highly depends on users. Differently from [26], we deal with structured data and we automatically slice the dataset with respect to the actual attribute domains.

DeepDiver [4] addresses the lack of adequate coverage in a dataset. Inadequate representation may cause errors in predictions and undesirable outcomes such as algorithmic racism. Uncovered patterns are introduced to identify attribute space regions not adequately covered by the data. Data subgroups, as in our work, are identified by attribute value combinations. However, [4] addresses a different problem, because the target attributes and classification outcome are not considered in the coverage problem, while we explicitly consider classification performance and identify subgroups in which a classification model performs differently with respect to the overall population. Furthermore, differently from [4] which considers underrepresented groups, we consider subgroups with adequate representation selected by a frequency threshold.

Many techniques have been proposed for understanding the reasons behind model predictions in the explainable AI literature [12]. LIME [21], Anchor [22] and SHAP [18] focus on explaining the factors influencing an individual prediction for a given instance. Differently, our work focuses on studying the statistical behavior of a classifier on the entire dataset. Rather than understanding individual predictions, we provide an analysis of the entire dataset by characterizing the subgroups in which a different behavior than overall is observed. As SHAP [18], we exploit the concept of Shapley value from coalition game theory. In SHAP, the Shapley value is used to compute the contribution of each feature value to the prediction for a single instance. In our work, the notion of Shapley value is adopted to characterize the contribution of each attribute value to the subgroup divergence, hence characterizing a subgroup rather than a specific instance. Furthermore, we generalize the notion of Shapley value to estimate the global contribution to divergence of each attribute value over the entire dataset.

# 3 ITEMSET DIVERGENCE

In this section, we first review basic concepts of frequent pattern mining, and we then define the notion of *divergence* that will be used in the paper.

## 3.1 Dataset and Itemsets

An $n$-dimensional dataset $D$ consists of a set of instances over a set $A$ of attributes (i.e., with schema $A$), with $|A| = n$. We assume that every attribute $a \in A$ can take a discrete, finite set $\mathcal{D}_a$ of values, and we let $m_a = |\mathcal{D}_a|$. An *instance* $x \in D$ assigns value $x(a) \in \mathcal{D}_a$ to every attribute $a \in A$. We only consider *discretized* attributes; continuous-valued attributes are discretized before our analysis techniques are applied.

An *item* $\alpha$ is an attribute equality $a = c$ for $a \in A$ and $c \in \mathcal{D}_a$ (e.g. *sex=Male*). We say that an instance $x$ is covered by the item $\alpha : a = c$, written $x \models \alpha$, if $x(a) = c$. Thus, covering is the equivalent of the logical notion of satisfaction. We denote with $\text{attr}(\alpha)$ the attribute to which an item refers, so that $\text{attr}(a = c) = a$.

An *itemset* is a set of items $I = \{\alpha_1, \ldots, \alpha_k\}$ that refer each to a distinct attribute, i.e., such that $\text{attr}(\alpha_i) \neq \text{attr}(\alpha_j)$ for all $1 \leq i < j \leq k$. An itemset $I = \{\alpha_1, \ldots, \alpha_k\}$ can be represented as the conjunction $\alpha_1 \wedge \cdots \wedge \alpha_k$ of its items (e.g. {*sex=Male, #prior=0*}). An instance $x \in D$ is covered by an itemset $I$, written $x \models I$, if $x \models \alpha_i$ for $1 \leq i \leq k$. Itemsets can also be depicted as *data cubes*.

The *support-set* $D(I) = \{x \in D \mid x \models I\}$ of an itemset $I$ consists of the instances that satisfy $I$, while the support of $I$ is given by $sup(I) = \frac{|D(I)|}{|D|}$.

The *length* of an itemset is the number of items contained in it, that is, the number of conjuncts. The length can range between 0, for the empty itemset, and $n$, the number of attributes. We denote by $\text{attr}(I) = \bigcup_{\alpha \in I} \text{attr}(\alpha)$ the set of attributes included in an itemset. For a subset of attributes $B \subseteq A$, we write $\mathcal{I}_B = \{I \mid \text{attr}(I) = B\}$ for the itemsets over attributes $B$. In particular, the set $\mathcal{I}_A$ consists of the itemsets that contain all attributes of our dataset.

## 3.2 Outcome Function and Itemset Divergence

Consider a dataset $D$ with schema $A$, alongside a function $f : 2^D \mapsto \mathbb{R}$. The function $f$ represents a statistics that can be computed over (subsets of) the dataset, such as the false positive or negative classification rates. For an itemset $I$, we write for brevity $f(I)$ for $f(D(I))$, denoting $f$ evaluated on the set of instances that satisfy $I$.

We define the *$f$-divergence* over an itemset $I$ as the difference between the statistics $f$ as measured on $I$, and as measured on the complete dataset.

**Definition 3.1.** (*itemset divergence*). Let $I$ be an arbitrary itemset in dataset $D$ and $f : 2^D \mapsto \mathbb{R}$ a function defined over subsets of the dataset. The *$f$-divergence* of itemset $I$ is:

$$\Delta_f(I) = f(I) - f(D) \tag{1}$$

We do not provide $f$ directly to DivExplorer. Rather, we specify $f$ as the *outcome rate* of an *outcome function*. This will be instrumental in allowing the efficient algorithmic computation of itemset divergences.

**Definition 3.2.** (*outcome function and positive outcome rate*). Given a dataset $D$, an *outcome function* is a function $o : D \mapsto \{\text{T}, \text{F}, \bot\}$.

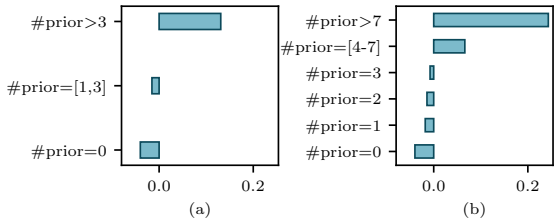


**Figure 1: Individual item divergence for false-positive rate of *prior* attribute value of the *COMPAS* dataset where the attribute is discretized in 3 (a) and 6 (b) intervals ($s$=0.05.)**

The *positive outcome rate* $f_o(X)$ of $o$ over a set of instances $X \subseteq D$ is defined as

$$f_o(X) = \frac{|\{x \in X \mid o(x) = \text{T}\}|}{|\{x \in X \mid o(x) \neq \bot\}|} \tag{2}$$

Thus, instances $x$ with $o(x) = \bot$ are not considered in the computation of the positive rate. In this paper, we will concern ourselves mostly with classifiers, and the outcome $o(x)$ will indicate whether $x$ is a false-positive, or false-negative, instance in the classification. Specifically, if $v : D \mapsto \{\text{T}, \text{F}\}$ is the ground truth, and if $u : D \mapsto \{\text{T}, \text{F}\}$ is the classification outcome, to study the false-positive rate we use

$$o(x) = \begin{cases} \text{T} & \text{if } u(x) \wedge \neg v(x); \\ \text{F} & \text{if } \neg u(x) \wedge \neg v(x); \\ \bot & \text{if } v(x). \end{cases}$$

An outcome function reflecting the false-negative rate can be similarly defined. If we wish to study the positive rate of the ground truth, we can obviously set $o(x) = v(x)$ for $x \in X$. The classification outcome $u$ can be the output of a generic classification model, making the approach model agnostic.

We will refer to $\Delta_f(I)$ as the *$f$-divergence* of $I$. When $o$ is the false positive outcome, we will call this the *false positive divergence* of $I$, and so forth. When $f$ is generic or can be understood from the context, for brevity we omit it by using the notation $\Delta(I)$.

By relying on a Boolean outcome function, we can apply DivExplorer to classifiers as black boxes, without need for accessing their internal loss or classification probability, as would be needed for real-valued outcome functions. As we shall see in Section 6.1, the focus on Boolean outcome functions also allows efficiently exploring itemsets and measuring their divergence.

The DivExplorer tool supports multiple metrics to assess classifier performance, such as accuracy, misclassification error, positive predictive value, true positive and negative rates, false discovery and false omission rates. We will mostly focus on false positive and false negative rates as our measures $f$ of interest. The $f$-divergence satisfies the following property.

**Property 3.1.** (*divergence is not hidden by finer discretization*). Let $X$ be a set of instances, and let $X_1, .., X_n$ be a partition of $X$, so that $\bigcup_{i=1}^{n} X_i = X$ and $X_i \cap X_j = \emptyset$ for $1 \leq i < j \leq n$. For any $f$-divergence measure, there is at least one subset $X_i$, $1 \leq i \leq n$, with $f$-divergence equal or greater than $X$ in absolute value.

The property holds because the divergence of $X$ is simply the weighted average of $X_1, \ldots, X_n$, where the weight of $X_i$ is the number of instances with non-bottom outcome function in $X_i$, for $1 \leq i \leq n$. The property has an important implication for the discretization of continuous-valued attributes. If we refine a discretization, for every divergent itemset in the coarser discretization there is at least one finer itemset that has equal or greater divergence. In other words, a finer discretization never hides divergence. This is illustrated in Figure 1: when the item *#prior>3* is split into the two finer ones *#prior=[4-7]* and *#prior>7*, the finer *#prior>7* has greater divergence than *#prior>3*.

## 3.3 Statistical Significance

Once an itemset with high divergence is identified, the question arises as to whether the divergence is statistically significant, or whether it originates from statistical fluctuations due to the finite size of the itemset. We can exploit the fact that the outcome function is Boolean, and follow an approach based on Bayesian statistics.

Our aim is to estimate the precision in the knowledge of the positive rate. We reason as follows. Consider a Bernoulli trial (a coin toss) with outcomes T with probability $Z$, and F with probability $1 - Z$. In our setting, the Bernoulli trial is the evaluation of the outcome function $o$ over an instance in the itemset, and $Z$ is the positive rate in the itemset. Before any trial is carried out, $Z$ is not known, and it is natural to assume a uniform prior, which is the least information prior, $\Pr(Z = z) = 1$ for $0 \leq z \leq 1$. If we then perform trials and we observe $k^+$ T outcomes and $k^-$ F ones, that is, if

$$k^+ = |\{x \mid x \models I \wedge o(x) = \text{T}\}|\,, \quad k^- = |\{x \mid x \models I \wedge o(x) = \text{F}\}|$$

we can use Bayes' rule to obtain the posterior distribution for the positive rate $Z$:

$$\Pr(Z = z) = \kappa\, z^{k^+}(1-z)^{k^-} = \text{Beta}(k^+ + 1,\, k^- + 1)(z)\,,$$

where $z \in [0, 1]$, $\text{Beta}(\alpha, \beta)(z) = \kappa\, z^{\alpha-1}(1-z)^{\beta-1}$ is the Beta distribution with parameters $\alpha, \beta$, and $\kappa$ is a normalization constant ensuring the distribution's integral in $[0, 1]$ is 1. This states the well-known fact that the Beta distribution is the posterior distribution that results from carrying Bernoulli trials starting from a uniform prior. We can then measure the mean and variance of our positive rate $Z$ via the mean $\mu_I$ and standard deviation $\nu_I$ of the Beta distribution:

$$\mu_I = \frac{k^+ + 1}{k^+ + k^- + 2} \qquad \nu_I = \frac{(k^+ + 1)(k^- + 1)}{(k^+ + k^- + 2)^2\,(k^+ + k^- + 3)} \tag{3}$$

Once mean and variance are known, we can compare the positive rate on $I$ to the positive rate on the whole dataset using Welch's t-test:

$$t = \frac{|\mu_I - \mu_D|}{\sqrt{\nu_I + \nu_D}}\,.$$

The advantage of the form (3) with respect to simply considering the mean and variance of the outcome function includes numerical stability when $k^+ + k^- = 0$, which happens when the outcome function is $\bot$ on the itemset (e.g., if we are measuring the FPR in an itemset where all instances have ground truth $v = \text{T}$).

| Itemset | Sup | $\Delta_{FPR}$ | $t$ |
|---|---|---|---|
| age=25-45, #prior>3, race=Afr-Am, sex=Male | 0.13 | 0.22 | 7.1 |
| age=25-45, #prior>3, race=Afr-Am | 0.15 | 0.211 | 7.4 |
| age=25-45, charge=F, #prior>3, race=Afr-Am | 0.11 | 0.202 | 6.2 |
| | Sup | $\Delta_{FNR}$ | $t$ |
| age=25-45, stay<week, #prior=0 | 0.15 | 0.236 | 12.1 |
| charge=M, stay<week, #prior=[1,3] | 0.10 | 0.233 | 12.2 |
| age>45, race=Cauc | 0.10 | 0.231 | 10.3 |
| | Sup | $\Delta_{ER}$ | $t$ |
| age<25, stay<week, race=Afr-Am | 0.10 | 0.098 | 4.7 |
| age<25, stay<week, sex=Male | 0.13 | 0.095 | 5.2 |
| age<25, race=Afr-Am, sex=Male | 0.11 | 0.090 | 4.5 |
| | Sup | $\Delta_{ACC}$ | $t$ |
| stay<week, #prior=0, race=Cauc | 0.12 | 0.141 | 8.4 |
| charge=M, stay<week, #prior=0 | 0.15 | 0.133 | 8.6 |
| charge=M, #prior=0 | 0.16 | 0.129 | 8.5 |

**Table 2: Top-3 divergent patterns with respect to FPR, FNR, error rate (ER) and accuracy (ACC) for the *COMPAS* dataset. The support threshold is $s = 0.1$.**

### 3.4 Frequent Itemsets and DivExplorer

The number of itemsets in a dataset is exponential in the number of attributes. Many itemsets may have very small or empty support, and these itemsets are of lesser interest for divergence analysis, for two reasons. First, in itemsets with small support, the measure of the positive rate of $o$ will be affected by statistical fluctuations, as discussed. Second, it is reasonable to assume that divergence affecting a larger portion of the dataset is more consequential than divergence affecting only a smaller portion of it. For these reasons, DivExplorer will only consider *frequent* itemsets, that is, itemsets $I$ whose support size $sup(I)$ is above a given threshold $s$ specified at the outset of the exploration.

The problem of finding all frequent itemsets in a dataset is a fundamental one in data mining, and much effort has been devoted to developing efficient algorithms for this task; see, e.g., [1, 13]. DivExplorer will leverage those algorithms, augmenting them so that the performance statistics $f$ can be computed for all frequent itemset. The detailed algorithms are presented in Section 5.

### 3.5 Summarizing divergent itemsets

To provide a compact representation of pattern divergence, we present a post-exploration pruning approach. A pattern $I$ is pruned if there exists an item $\alpha \in I$ whose absolute marginal contribution is lower than a threshold $\epsilon$, i.e. $|\Delta_f(I) - \Delta_f(I \setminus \{\alpha\})| \leq \epsilon$. The pattern $I \setminus \alpha$ captures the divergence of pattern $I$, since the inclusion of item $\alpha$ only slightly alters the divergence (slightly with respect to the threshold $\epsilon$). In Section 6.3, the impact of the $\epsilon$ input parameter on the number of resulting itemsets is studied.

### 3.6 Our Running Example: *COMPAS*

As a running example to illustrate the previous definitions, we again consider the *COMPAS* dataset. We compare the predicted recidivism rate with the actual rate, defined as the new occurrence of a misdemeanor or felony offense over a two-year period. For an instance (a person) $x$, we let $v$ be the ground truth, with $v(x) = \text{T}$ iff recidivism occurred, and $v(x) = \text{F}$ if none occurred. The classification outcome $u(x)$ corresponds to the output of the *COMPAS* system. We let $u(x) = \text{T}$ if *COMPAS* classifies person $x$ as being at high recidivism risk, and F otherwise.

Table 2 shows the most divergent patterns with respect to the false positive rate (FPR), false negative rate (FNR), error rate (ER) and accuracy (ACC) for a support threshold $s$=0.1. The pattern with highest false-positive rate divergence is $I_1$ = (*age=25-45,#prior>3, race=African-American, sex=Male*) with $\Delta_{FPR}(I_1)$=0.220. The model tends to be biased towards African-Americans with age in the range 25-45 that have a high number of prior offenses. These three items are shared for all the top-3 FPR divergent patterns. Another influencing item is having been convicted of a felony (charge=F).

The results also indicate that the model has a higher false negative rate for people with fewer than 3 prior offenses, short stays in jail (stay<week) and having a prior conviction of a misdemeanor charge (charge=M) rather than a felony. Caucasians with age greater than 45 also have higher-than-overall FNR. We note that the model has a higher error rate for African-American defendants with age lower than 25 and short stays in jail. The model tends to be more accurate for Caucasian defendants with short stays in jail and no prior offenses.

## 4 ITEM CONTRIBUTION TO DIVERGENCE

Once itemsets with large divergence are identified, such as those of Table 2, the question arises as to which of the items appearing in the itemsets are most responsible for the divergence. We introduce methods both for attributing the divergence of an itemset to its items, and for estimating the overall impact of an item on divergence.

### 4.1 Item Contribution to Itemset Divergence

Our definition of the item contribution to itemset divergence is based on the notion of the *Shapley value* of a player in a coalition.

The *Shapley value* [24] is defined in the context of a cooperative $N$-player game. Let $v(\sigma)$ be the value that can be attained by a coalition $\sigma \subseteq \{1, \ldots, N\}$ of players. When all players cooperate, they can achieve the value $v(\{1, \ldots, N\}) = v^*$. The Shapley value measures the contribution $\hat{v}(i)$ of each player to $v^*$, in such a way that $\sum_{i=1}^{N} \hat{v}(i) = v^*$. The Shapley value of player $i$, for $1 \leq i \leq n$, is given by:

$$\begin{aligned} \hat{v}(i) &= \sum_{\sigma \in \pi(1,\ldots,N)} v(\sigma[:i]^+) - v(\sigma[:i]^-) \qquad (4) \\ &= \sum_{\phi \subseteq \{1,\ldots,N\} \setminus \{i\}} \frac{|\phi|!(N - |\phi| - 1)!}{N!} \left[ v(\phi \cup \{i\}) - v(\phi) \right], \end{aligned}$$

where $\pi(1, \ldots, N)$ is the set of permutations of $1, \ldots, N$, and where, for a permutation $\sigma$, $\sigma[:i]^+$ is its prefix up to $i$ included, and $\sigma[:i]^-$ is its prefix up to $i$ excluded. The notion of Shapley value directly yields a way to measure the (local) contribution of an item to the divergence of an itemset.

**Definition 4.1.** (*item contribution to itemset divergence*). Given an itemset $I$ and an item $\alpha \in I$, the contribution $\Delta(\alpha \mid I)$ of $\alpha$ to the

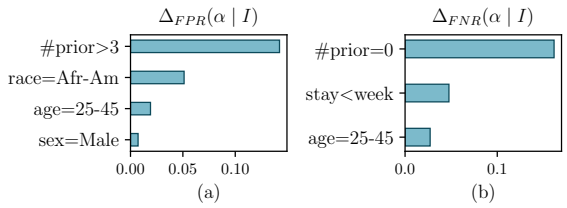


**Figure 2: Contributions of individual items to the divergence of the *COMPAS* frequent patterns having greatest false-positive and false-negative divergence.**

divergence of $I$ is:

$$\Delta(\alpha \mid I) = \sum_{J \subseteq I \setminus \{\alpha\}} \frac{|J|!(|I| - |J| - 1)!}{|I|!} \left[\Delta(J \cup \alpha) - \Delta(J)\right]. \quad (5)$$

In (5), if $I$ is a frequent itemset, then all the itemsets appearing in the formula are frequent, being subsets of $I$. Therefore, we can compute the local item contributions to frequent itemsets on the basis of the exploration performed by DivExplorer, which is limited to frequent itemsets.

Consider again the *COMPAS* dataset. Figure 2 gives the item contributions to the divergence of the frequent itemsets with largest false-positive and false-negative divergence. The item with the greatest influence on the false-positive divergence of the itemset is whether the person has at least 3 prior criminal charges. This is followed by belonging to the African-American race. The *sex=Male* item gives only minor contribution. For the divergence in false-negative rate, the greatest contribution is given by not having prior convictions. In general, we see that the items' contribution to itemset divergence can be quite different.

We note that the Shapley value tends to under-estimate the contribution to divergence of correlated items appearing jointly. For example, consider two fully-correlated items $\alpha$ and $\beta$ in itemset $I \cup \{\alpha, \beta\}$. When appearing jointly, $\alpha$ and $\beta$ are attributed only half of the contribution to divergence that they receive in isolation: if $c = \Delta(\alpha \mid I \cup \{\alpha\}) = \Delta(\beta \mid I \cup \{\beta\})$, we have $\Delta(\alpha \mid I \cup \{\alpha, \beta\}) = \Delta(\beta \mid I \cup \{\alpha, \beta\}) = c/2$.

This effect is intrinsic to the way in which the Shapley value attributes contribution symmetrically. In DivExplorer, users can explore the lattice around any divergent itemset (see Section 6.4). The lattice would show that the divergence of $I \cup \{\alpha, \beta\}$ was already present in $I \cup \{\alpha\}$ and $I \cup \{\beta\}$. Users can appreciate the contribution of the items $\alpha$ and $\beta$ by looking at their contributions to these shorter itemsets. Furthermore, this situation is mitigated by the redundancy pruning described in Section 3.5. According to this pruning, the itemset $I \cup \{\alpha, \beta\}$ is omitted from the output, since it is no more divergent than its subsets $I \cup \{\alpha\}$ and $I \cup \{\beta\}$.

## 4.2 Corrective Items

Divergence is not monotonic: $I \subseteq J$ does not imply $\Delta(I) \leq \Delta(J)$ for itemsets $I, J$. We call items that decrease divergence when added to an itemset *corrective items*.

**Definition 4.2.** (*corrective item and corrective factor*). Given an itemset $I$ and an item $\alpha \notin I$, we say that $\alpha$ is a *corrective item* for $I$ if $|\Delta(I \cup \alpha)| < |\Delta(I)|$. The *corrective factor* of $\alpha$ w.r.t. $I$ is $|\Delta(I)| - |\Delta(I \cup \alpha)|$.

| I | corr. item | Δ(I) | Δ(I ∪α) | c_f | $t$ |
|---|---|---|---|---|---|
| *FPR* | | | | | |
| race=Afr-Am, sex=Male | #prior=0 | 0.062 | 0.009 | 0.053 | 2.8 |
| race=Afr-Am | #prior=0 | 0.051 | -0.001 | 0.051 | 3.4 |
| stay<week, #prior=0 | race=Afr-Am | -0.044 | -0.003 | 0.041 | 3.1 |
| *FNR* | | | | | |
| charge=F, race=Afr-Am, sex=Male | #prior=[1,3] | -0.123 | -0.011 | 0.112 | 3.8 |
| charge=F, race=Afr-Am | #prior=[1,3] | -0.113 | 0.004 | 0.109 | 4.3 |
| race=Afr-Am, sex=Male | charge=M | -0.090 | -0.001 | 0.089 | 3.3 |

**Table 3: Top corrective items for FPR and FNR of *COMPAS* dataset.**

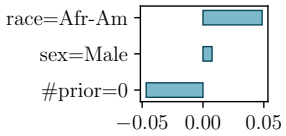


**Figure 3: An itemset where an item has a negative divergence contribution.**

By performing an exhaustive exploration of all frequent itemsets, DivExplorer can identify the corrective items.

Table 3 shows the top corrective items for false-positive and false-negative divergence in *COMPAS*. The FPR-divergence of itemset $I_3$=(*race=Afr-Am, sex=Male*) drops from 0.062 to 0.009 when item *#prior=0* is included, with a corrective factor of 0.053. The absence of prior convictions tends to lower the wrong assignments of African-American male defendants to the recidivist class to a similar FPR rate to the overall.

Figure 3 shows that the corrective effect of having no prior convictions is also reflected in the item contributions to the divergence of the corrected itemset, measured according to the Shapley value.

## 4.3 Global Item Divergence

Given an individual item $\alpha$, there are two ways of measuring the effect of $\alpha$ on divergence. One is via its divergence $\Delta(\alpha)$ defined in (1). This *individual* measurement is the most common way of measuring the effect of an item on divergence. For example, when studying the effect of *race = African-American* on classification, we can measure the false-positive or false-negative divergence for this item, to see if the classifier behaves differently for people in the support-set of the item compared to people at large.

Another way of measuring the effect of an item on classification is to consider the effect of adding the item to other itemsets. As this measures the effect of the item on all itemsets, it provides a *global* measurement of the effect of the item. Roughly, the *global divergence* of an item will tell us whether the item $\alpha$ tends to skew the classification in every possible context. The definition is based

on the notion of Shapley value, adapted to account for the fact that only items for different attributes can be part of the same itemset.

**Definition 4.3.** (*global itemset divergence*). Let $D$ be a dataset with schema $A$, and let $\Delta$ be the divergence of its itemsets measured for a given outcome function. We define the *global divergence* $\Delta^g(I)$ of an itemset $I$ of $D$ as:

$$\Delta^g(I) = \sum_{B \subseteq A \setminus \text{attr}(I)} \frac{|B|!(|A| - |B| - |I|)!}{|A|! \prod_{b \in B \cup \text{attr}(I)} m_b} \sum_{J \in \mathcal{I}_B} \left[\Delta(J \cup I) - \Delta(J)\right] . \tag{6}$$

The definition parallels (4), except for the additional factor $1/(\prod_{b \in B \cup \text{attr}(I)} m_b)$, which is necessary to normalize the sums, accounting for the number of different itemsets with given attributes. The following theorem gives the properties of the above notion of global divergence. Together, these properties formalize the fact that (6) is the generalization of Shapley value to the itemset case.

Theorem 4.1. (properties of global divergence). *Consider a dataset $D$ with set $A$ of attributes, alongside a divergence $\Delta$ for its itemsets. The global divergence defined as in (6) satisfies the following properties:*

- Efficiency:
$$\sum_{a \in A} \sum_{c \in \mathcal{D}_a} \Delta^g(a = c) \; = \; \frac{1}{|\mathcal{I}_A|} \sum_{I \in \mathcal{I}_A} \Delta(I) \; . \tag{7}$$
- Null items: *if there is an attribute $a \in A$ such that, for all $c \in \mathcal{D}_a$ and all itemsets $I \in \mathcal{I}$ with $a \notin attr(I)$, $\Delta(I) = \Delta(I \cup \{a = c\})$, then $\Delta^g(a = c) = 0$. Furthermore, under the above hypotheses, removing $a$ from $A$ does not affect the value of $\Delta^g(I)$ for any itemset $I$ not containing $a$.*
- Symmetry: *if for two itemsets $I, I'$ we have $\Delta(J \cup I) = \Delta(J \cup I')$ for all $J \in \mathcal{I}$ with $attr(J) \cap attr(I) = \emptyset$ and $attr(J) \cap attr(I') = \emptyset$, then $\Delta^g(I) = \Delta^g(I')$.*
- Linearity: *If two notions of divergence $\Delta_1, \Delta_2 : 2^X \mapsto \mathbb{R}$ are combined into a single one $\Delta = \gamma_1\Delta_1 + \gamma_2\Delta_2$ via a linear combination, for every item $I$ we will have $\Delta^g(I) = \gamma_1\Delta_1^g(I) + \gamma_1\Delta_2^g(I)$, where $\Delta^g$ is computed from $\Delta$, and $\Delta_1^g$, $\Delta_2^g$ from $\Delta_1, \Delta_2$, respectively.*

These properties are the generalization of the corresponding properties of Shapley values. The difference in the forms is due to the fact that there is more than one complete itemset.

*Accounting for support lower bound.* In DivExplorer we cannot use (6) directly, as it involves the consideration of all itemsets. Rather, we opt for an approximation of (6), in which we limit the summation to frequent itemsets, whose support is at least $s$. Let $\mathcal{I}_B^\star$ be the set of frequent itemsets with attributes B. We define the global divergence approximated to support $s$ via:

$$\widetilde{\Delta}^g(I, s) = \tag{8}$$
$$\sum_{B \subseteq A \setminus \text{attr}(I)} \frac{|B|!(|A| - |B| - |I|)!}{|A|! \prod_{b \in B \cup \text{attr}(I)} m_b} \sum_{J : J \cup I \in \mathcal{I}^\star_{B \cup \text{attr}(I)}} \left[\Delta(J \cup I) - \Delta(J)\right].$$

If $I$ is frequent, the summations can be computed in terms of frequent itemsets only, and the approximation can be computed on the basis of the output of DivExplorer, which only outputs frequent itemsets.

## 4.4 Global vs. Individual Item Divergence

For an item $\alpha$, we can measure both the *individual divergence* $\Delta(\alpha)$ defined by (1), and the *global divergence* $\widetilde{\Delta}^g(\alpha, s)$ defined by (8). The individual divergence $\Delta(\alpha)$ is independent of support threshold (provided the item itself is above the threshold). The global divergence $\widetilde{\Delta}^g(\alpha, s)$, on the other hand, depends on the support threshold $s$ chosen for its analysis.

The value of global divergence lies in its ability to highlight the role of items in giving rise to divergence via association with other items. For instance, assume that in a dataset there are two items $\alpha, \beta$ that cause divergence in the itemset $\{\alpha, \beta\}$, but less so in isolation. The individual divergence of $\alpha$ and $\beta$ may be low, masking the effect of the items when jointly present. On the other hand, global divergence is able to capture the effect of $\alpha$ and $\beta$ on divergence, provided the itemset $\{\alpha, \beta\}$ has support above the threshold. We make this observation precise via a theorem, and via an example on an artificial dataset.

Theorem 4.2. (individual and global divergence do not coincide). *There is a dataset $D$ with schema $A$, a minimum support $s > 0$, and items $a = c$ for $a \in A$, $c \in \mathcal{D}_a$, such that $\Delta(a = c) = 0$ but $\widetilde{\Delta}^g(a = c, s) \neq 0$.*

To illustrate how global item divergence is able to capture the role of items that cause divergence when joint with other items, we constructed an artificial 10-dimensional dataset, denoted *artificial*, with 50,000 instances and attributes $a, b, c, \ldots, j$ with domain $\{0, 1\}$.

We construct the dataset, and a classifier, so that the itemsets $a = b = c = 1$ and $a = b = c = 0$ are divergent. To this end, we create the instances by setting each of their attributes $a, \ldots, j$ randomly and independently to values 0 and 1, with equal probability. We first train a classifier with respect to a class label that is T when $a = b = c$ and F otherwise. Then, to simulate classification errors, during test, we flip the class label for half of the instances in $a = b = c$ (without retraining the classifier).

The global and individual item divergence for the false positive rate, analyzed with minimum support $s = 0.01$, are given in Figure 4. We see that individual item divergence is unable to capture the role of $a, b, c$, together, to cause high divergence. The divergence of $a = b = c$ is completely masked by statistical fluctuations in the overall dataset, to the point that unrelated items such as $g = 0, g = 1, h = 0, h = 1$ have much larger individual divergence that items for attributes $a, b, c$. On the other hand, the global divergence is clearly able to identify the attributes $a, b, c$ as those causing divergence when appearing together.

For the *COMPAS* dataset, Figure 5 compares the global and individual false-positive divergence for items. Global divergence assigns more importance to racial factors: for instance, being African American introduces almost as much bias to an itemset as having been convicted more than 3 times. This indicates how race plays a role jointly with other factors in creating highly divergent itemsets.

# 5 THE DIVEXPLORER ALGORITHM

The DivExplorer algorithm extracts frequent subsets of attribute values and estimates their divergence. The computation is embedded in the frequent pattern extraction process, and DivExplorer can leverage any frequent pattern mining (FPM) technique [25] to

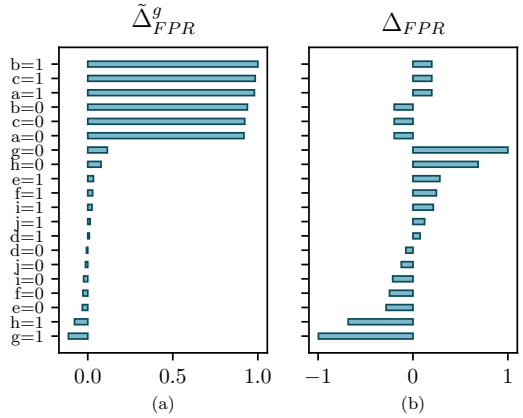


**Figure 4: Relative magnitudes of $\widetilde{\Delta}^g(\cdot, s)$ and individual item divergence, for false-positive rate in the artificial dataset. The attributes $a, b, c$ give raise to divergence when appearing together: global divergence captures this.**

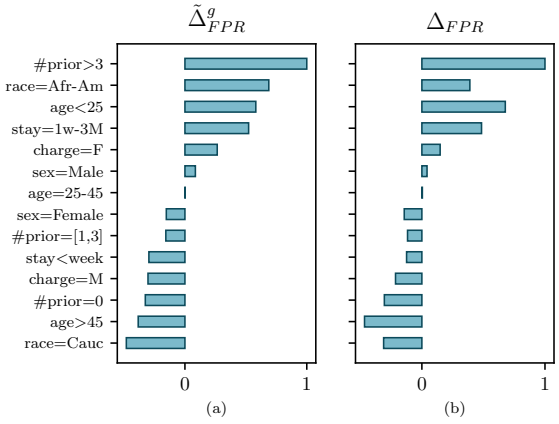


**Figure 5: Relative magnitudes of global Shapley value and individual item divergence, for false-positive rate in the *COMPAS* dataset with $s = 0.1$**

extract frequent subsets. More specifically, when the support of an itemset is estimated, the outcome function $o$ and outcome rate $f$ are also computed. Hence, the performance of DivExplorer directly depends on the efficiency of the selected FPM algorithm, because the dataset is accessed as many times as the selected underlying FPM method does.

FPM algorithms require discrete data. Thus, continuous attributes (if any) are firstly discretized. This discretization is only performed after the classification process. In particular, DivExplorer does not require the classification algorithm to rely on discretization.

Algorithm 1 outlines the main steps of DivExplorer. Given the input data set $D$, the ground truth $v$, the classification outcome $u$, the outcome function $o$ and outcome rate of interest $f$, the algorithm returns the divergence coefficient for all frequent itemsets. The

**Algorithm 1:** The DivExplorer algorithm.

**Input:** $D, u, v, o, f, s$
**Output:** FP divergence $FP_\Delta$

```
1  o(D) = computeOutcomeFunction(D, o, f);
2  T̂_D, F̂_D, ⊥̂_D = OutcomeOneHotEncoding(o(D));
3  FI_withOutcomes = [];
4  for step_i in Frequent Pattern Mining steps do
5      I_step_i = extractItemsets(D, step_i);
6      for I in I_step_i do
7          T_I, F_I, ⊥_I = cardinalityOutcomesI(I, (T̂_D, F̂_D, ⊥̂_D));
8          if (T_I + F_I + ⊥_I)/len(D) ≥ s then
9              FI_withOutcomes.append((I, (T_I, F_I, ⊥_I));
10         end
11     end
12 end
13 FP_f = evaluateFunctionf(FI_withOutcomes, f);
14 FP_Δf = divergence(FP_f, f(D));
15 return FP_Δf
```

algorithm requires the definition of the minimum support threshold $s$ as its (single) input parameter.

The first step (Line 1) of Algorithm 1 computes the outcome function on all instances $x \in D$ (see Section 3.2). The outcome function results are then input to the *OutcomeOneHotEncoding* function that maps each outcome to a one-hot representation. More specifically, for each instance $x \in D$, $\mathrm{T}_x$, $\mathrm{F}_x$ and $\bot_x$ are estimated, with $\mathrm{T}_x$ equal to 1 if $o(x) = \mathrm{T}$ and 0 otherwise ($\mathrm{F}_x$ and $\bot_x$ are computed analogously). This representation enables us to tally the outcome function values simply by adding the one-hot representations. The results are $\hat{\mathrm{T}}_D$, $\hat{\mathrm{F}}_D$ and $\hat{\bot}_D$ one-hot representations of outcome function $o(x)$ for dataset $D$.

Next, for each $step_i$ of a generic FPM technique, itemsets are extracted (Line 5). The general function *extractItemsets* extracts the itemsets to be evaluated for support threshold at $step_i$ and varies depending on the FPM algorithm of choice. For example, $step_i$ could be level $i$ iteration in level-wise approaches such as Apriori [1], or the recursive step performed by FP-growth [13] on the FP-tree compressed representation. We implemented both an Apriori-based and an FP-growth-based version of DivExplorer.

The cardinalities $\mathrm{T}_I$, $\mathrm{F}_I$, $\bot_I$ of each itemset $I$ extracted at $step_i$ are then estimated, with $\mathrm{T}_I = |\{x \mid x \models I \wedge o(x) = \mathrm{T}\}|$, $\mathrm{F}_I = |\{x \mid x \models I \wedge o(x) = \mathrm{F}\}|$ and $\bot_I = |\{x \mid x \models I \wedge o(x) = \bot\}|$. Note that function *cardinalityOutcomesI* does not require access to the dataset $D$, because it is integrated in the FPM algorithm. $\mathrm{T}_I$, $\mathrm{F}_I$ and $\bot_I$ are computed as the sum of the $\hat{\mathrm{T}}_D$, $\hat{\mathrm{F}}_D$ and $\hat{\bot}_D$ terms that satisfy $I$. The sum of $\mathrm{T}_I$, $\mathrm{F}_I$ and $\bot_I$ represents the support count of itemset $I$, i.e. $| D(I) |$. Hence, it is exploited to estimate if $I$ is frequent, i.e. with a support greater or equal than $s$ (Line 8). Frequent itemsets and their cardinality outcomes are stored in *FI_withOutcomes* (Line 9).

Once all frequent itemsets are extracted, the outcome rate of outcome function $f$ is estimated for all frequent itemsets with *evaluateFunctionf* (Line 13). Finally, the $f$-divergence (Equation 1) of all frequent itemsets is computed (Line 14) and returned (Line 15). The extracted frequent itemsets can be ranked according to many

| dataset | $\|D\|$ | $\|A\|$ | $\|A\|_{cont}$ | $\|A\|_{cat}$ |
|---|---|---|---|---|
| *adult* | 45,222 | 11 | 4 | 7 |
| *bank* | 11,162 | 15 | 6 | 9 |
| *COMPAS* | 6,172 | 6 | 2 | 4 |
| *german* | 1,000 | 21 | 7 | 14 |
| *heart* | 296 | 13 | 5 | 8 |
| *artificial* | 50,000 | 10 | 0 | 10 |

**Table 4: Dataset characteristics. $A_{cont}$ is the set of continuous attributes, $A_{cat}$ of categorical ones.**

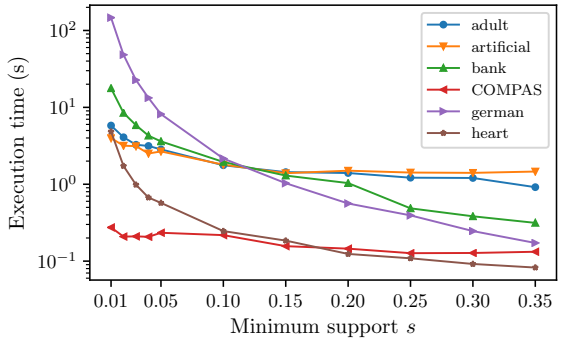


**Figure 6: DivExplorer execution time when varying the minimum support threshold.**

different metrics, such as their statistical significance, support, or $f$-divergence. In this paper, we rank itemsets according to $f$-divergence, to identify subgroups where the behavior diverges strongly. Users can choose their preferred ranking according to the problem, and to their desired analysis goals.

It is straightforward to extend Algorithm 1 to efficiently compute the $f$-divergence of multiple outcome functions simultaneously.

Theorem 5.1. *(*Soundness and completeness*) Algorithm 1, called with minimum support $s$, is sound and complete:*

- Sound: *If Algorithm 1 outputs an itemset $I$ along with $f$-divergence $\Delta_f(I)$, then there is an itemset $I$ in the dataset with support above $s$ and with divergence $\Delta_f(I)$.*
- Complete: *If there is an itemset $I$ with support above $s$ and with divergence $\Delta_f(I)$, the itemset $I$ along with its divergence will be part of the output.*

We note that completeness does not hold for Slice Finder, since the search for problematic itemsets is pruned whenever sufficiently problematic itemsets are found, so that longer (more specific) itemsets, even if more problematic, can be missed. This will be illustrated later in Section 6.5.

# 6 EXPERIMENTAL RESULTS

We present here results on the running time of DivExplorer, on its ability to extract and summarize divergence information on real-world datasets, and on the visualizations and explorations that can be created on the basis of its output. We also outline the main differences between our approach and Slice Finder [8].

The main features of the datasets used in our experiments are reported in Table 4. The cardinalities are reported after standard preprocessing steps (e.g., removing instances with missing values). For most of our experiments, we used the *COMPAS* dataset [3], already introduced in Section 3.6, and the *adult* dataset [17]. The *adult* dataset includes census data and the prediction of individual incomes, divided in two classes "≤50K" and "'>50K". In our analysis we used the age, workclass, education, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week features. For the performance experiments, we also used the *German Credit Data*, *Bank Marketing*, and *heart* datasets [17]. The *German Credit Data* (*german*) dataset is devoted to the prediction of an individual's credit risk using loan application data, according to attributes as age, sex[1], checking_account, credit_amount, duration, purpose, etc. The *Bank Marketing* (*bank*) dataset contains information related to a direct marketing campaign of a Portuguese banking institution. The *heart* dataset contains data to detect the presence of a heart disease in patients. Its features describe the demographic and health information (as serum cholesterol, resting blood pressure) of patients. Finally, we also used the *artificial* dataset already described in Section 4.4.

DivExplorer has been developed in Python. The source code and all the datasets used in our experiments are available [20], together with the description of all performed preprocessing steps. In all the reported experiments, DivExplorer is coupled with FP-growth as frequent pattern mining technique to extract frequent itemsets [13].

## 6.1 Performance analysis

We evaluated the efficiency of DivExplorer by measuring the execution time required to (i) extract all frequent itemset and (ii) estimate their divergence and statistical significance. We repeated each experiment 5 times and reported the average execution time. The experiments were performed on a PC with Ubuntu 16.04.1 LTS 64 bit, 16 GB RAM, 2.40GHz×4 Intel Core i7. For all the datasets, except *COMPAS* and the *artificial* dataset (for which the class label is already provided), we used a random forest classifier with default parameters to provide the classification outcome $u$.

Figure 6 shows DivExplorer execution time as a function of the support threshold. The higher the support threshold, the lower the running time. Note that the execution time depends on the FPM algorithms used for the extraction of the itemsets (FP-growth in the reported experiments). For all considered datasets, except *german*, the execution time is below 20s, even for minimum support thresholds as low as 0.01. For the *german* dataset the worst case execution time is anyway lower than 150s. The execution time required to compute itemset divergence and statistical significance is negligible (<7%) compared to the time required for itemset extraction.

The number of frequent itemsets extracted by DivExplorer when varying the minimum support is reported in Figure 7. For low

[1]From the the original features, we derived "sex" and "civil-status" from the "personal-status" attribute.

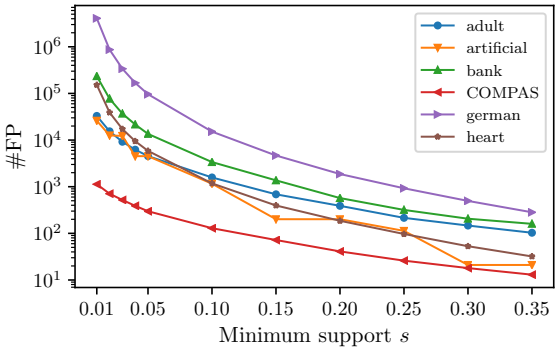


**Figure 7: Number of frequent itemsets when varying the minimum support threshold.**

| Itemset | Sup | $\Delta_{FPR}$ | $t$ |
|---|---|---|---|
| gain=0, status=Married, occup=Prof, race=White | 0.05 | 0.469 | 25.8 |
| gain=0, loss=0, status=Married, occup=Prof | 0.05 | 0.462 | 26.6 |
| loss=0, status=Married, occup=Prof, race=White | 0.06 | 0.458 | 25.3 |
| | Sup | $\Delta_{FNR}$ | $t$ |
| age≤28, gain=0, hoursXW≤40, status=Unmarried | 0.17 | 0.61 | 21.8 |
| gain=0, loss=0, edu=HS, hoursXW≤40, status=Unmarried | 0.14 | 0.61 | 28.2 |
| gain=0, loss=0, status=Unmarried, relation=Own-child | 0.12 | 0.61 | 18.9 |

**Table 5: Top-3 divergent itemsets for FPR and FNR. *adult* dataset, $s = 0.05$.**

support thresholds, the number of extracted patterns for the *german* dataset is very high, thus impacting the execution time, as shown in Figure 6. For this dataset, a support threshold equal to 0.01 is rather low, as it corresponds to 10 records only (see Table 4). Nevertheless, the ability of DivExplorer to find divergent itemsets with very low levels of support enables the analysis of under-represented group behavior in the dataset.

## 6.2 Exploring dataset divergence

In this section, we demonstrate the capability of DivExplorer to (a) detect the itemsets that mostly contribute to misclassifications, (b) provide a "drill-down" analysis to highlight most influential items in an itemset divergence, and (c) explore the global contribution of single items to divergence. We focus on the *adult* dataset, as similar results for *COMPAS* have been presented throughout the paper. A complete report of the experimental outcome for all the datasets under analysis is available [20].

Table 5 shows the top divergent itemsets for *adult*, both for the FPR and FNR rate, with $s = 0.05$. The reported itemsets show some degree of overlap, which will be discussed in Section 6.3. Figure 8 reports the item contributions to the top divergent itemsets of Table 5. Figure 8(a) shows that the most relevant items which contribute to the higher-than-overall misclassification rate for the high income class are *being married* and *working as a professional*.

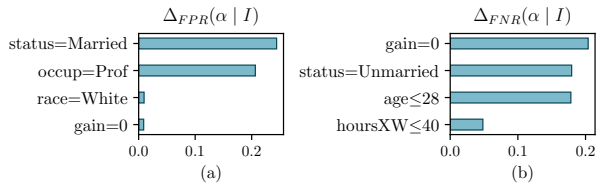


**Figure 8: Contributions of individual items to the divergence of the adult frequent patterns having greatest FPR (Line 1 of Table 5) and FNR (Line 4 of Table 5) divergence.**

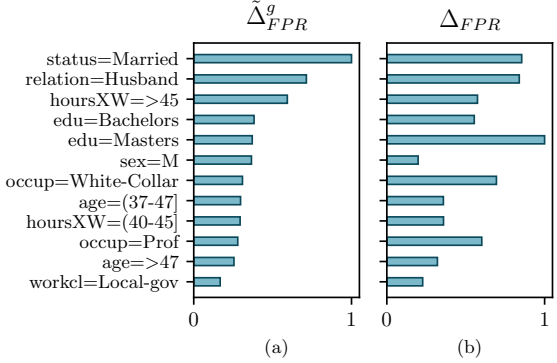


**Figure 9: Relative magnitude of global Shapley value (a) and individual item divergence (b), for FPR, adult dataset, $s = 0.05$. Top 12 global item positive contributions are reported.**

| Itemset | Sup | $\Delta_{FPR}$ | $t$ |
|---|---|---|---|
| status=Married, occup=Prof | 0.07 | 0.434 | 26.1 |
| occup=Prof, relation=Husband | 0.06 | 0.423 | 23.4 |
| edu=Bachelors, status=Married | 0.09 | 0.413 | 29 |

**Table 6: Top-3 divergent itemsets for FPR with redundancy pruning. *adult* dataset, $\epsilon = 0.05$, $s = 0.05$.**

The items *gain=0* (capital gain) and *race=White* have instead a very small influence. For the top FNR itemset (Figure 8(b)), we observe that *age* ≤ *28*, *capital gain = 0*, and *being unmarried* are the most important items, while *number of hours per week* ≤ *40* provides a limited contribution.

Figure 9 shows the relative magnitude of global and individual item contribution to FPR divergence, again for *adult*; for conciseness, only the 12 items with largest positive contribution are shown. Consider the item *edu = Masters*. While its individual divergence is the highest overall, its global divergence is markedly lower, indicating its limited role in giving rise to divergence via association (in longer itemsets). Indeed, *edu = Masters* does not appear in the top divergent itemsets of Table 5.

## 6.3 Summarizing divergent itemsets

As seen in Table 5, the top divergent itemsets often include some level of redundancy. DivExplorer can reduce such redundancy

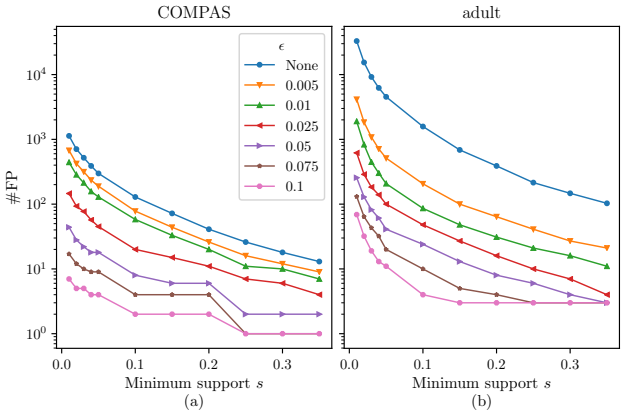


**Figure 10: Number of frequent itemsets varying redundancy pruning threshold $\epsilon$ for FPR divergence of *COMPAS* and *adult* datasets.**

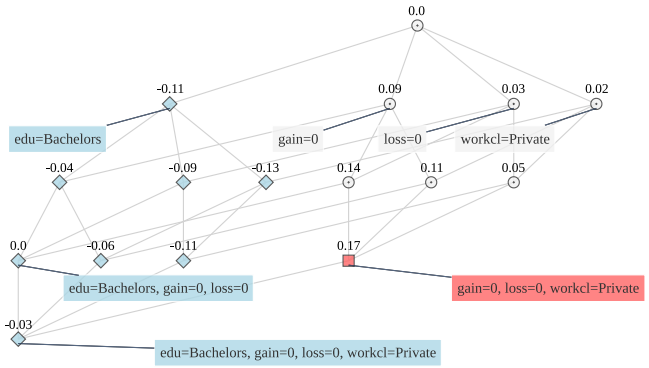


**Figure 11: Lattice showing a corrective phenomenon for FNR divergence on the *adult* dataset. Nodes showing a corrective phenomenon appear as rhombus in light blue. Nodes with FNR-divergence $\geq T = 0.15$ are squares in red.**

via the heuristic pruning approach discussed in Section 3.5, which eliminates itemsets that are not significantly more divergent than their shorter subsets. We report in Table 6 the top FPR-divergent itemsets for the *adult* dataset when applying a redundancy threshold $\epsilon = 0.05$. Comparing this result with Table 5, we note how pruning helps in presenting more diverse, and thus relevant, information. The most FPR-divergent itemset in Table 6 is *status=Married, occup=Prof* (occupation=Professional), with a slightly lower divergence and similar statistical significance. The importance of these two items was already shown by Figure 8(a), in which they were providing the most relevant contribution to the itemset divergence. On a global scale, for the FPR-divergence, the total number of extracted itemsets drops from 4534 to just 40.

Figure 10(a) and 10(b) report a quantitative evaluation of the impact of the pruning parameter $\epsilon$, and minimum support $s$ on the number of divergent itemsets returned, for FPR-divergence in *COMPAS* and *adult*. We see how the heuristic post-pruning, even with relatively small values of $\epsilon$, leads to an effective summarization of divergent itemsets.

## 6.4 Lattice visual exploration

DivExplorer allows the interactive exploration of divergent patterns by means of a visual representation of the itemset lattice. In this lattice, nodes correspond to frequent itemsets and edges to subset relationships between itemsets. Given a divergent pattern of interest $I$, the itemset lattice shows all its subsets and their divergence coefficient. The root represents the empty subset (with $\Delta_f$=0 by definition) and the last level the pattern $I$ itself. Figure 11 reports a portion of the lattice for the *adult* dataset. The itemset lattice may be actively navigated. The visualization allows the identification of the items driving divergence increases, i.e., items that, when added to a subset, increase the divergence. Furthermore, the user may interactively select a divergence threshold $T$. The lattice nodes with divergence coefficient larger than the threshold are highlighted.

The itemset lattice can also be exploited to explore corrective behaviors. The visualization highlights the subsets (i.e., nodes in the lattice) in which a corrective phenomenon is observable. An example of corrective phenomenon visualization is reported in Figure 11. The example shows the lattice for the FNR-divergence of itemset $I_x$=(*edu=Bachelors, gain=0, loss=0, workclass=Private*) in the *adult* dataset. Item *edu=Bachelors* is a corrective item for pattern $I_y$=(*gain=0, loss=0, workclass=Private*). The FNR-divergence drops from 0.17 for itemset $I_y$ to -0.03 for itemset $I_x$ when the item *edu=Bachelors* is included. Besides pattern $I_x$, item *edu=Bachelors* introduces a corrective effect for all the itemsets including it in the lattice. Hence, the exploration of the itemset lattice also allows a deeper and more comprehensive analysis of corrective behaviors.

## 6.5 Comparison with Slice Finder

Slice Finder [8] is the closest approach to DivExplorer. It identifies slices of the data, denoted by conjunctions of literals, i.e., itemsets, in which the model performs poorly. Slice Finder measures how "problematic" a slice is by comparing the classifier loss on the slice, and on the *remainder* of the dataset. This notion is similar to our notion of divergence, with two differences. First, Slice Finder measures classifier loss, while DivExplorer is based on an outcome function that encodes metrics such as FPR and FNR. Second, Slice Finder measures the difference between an itemset and its complement, while DivExplorer measures the difference between the itemset and the *whole* dataset. The main difference between Slice Finder and our approach, however, is that Slice Finder's search is not exhaustive: the exploration of an itemset is stopped (no larger itemsets are considered) when sufficiently large deviation is found, and the overall exploration stops once a prescribed number of itemsets has been found. We can afford to perform an exhaustive search due to our reliance on efficient frequent pattern mining algorithms. Our exhaustive search allows us to study item contribution to individual itemsets and global divergence. The exhaustive search also enables the identification of corrective items.

It is difficult to provide a comparison of Slice Finder and DivExplorer on a general dataset, because the two tools drive their exploration differently (effect size and bound on result size for Slice Finder, support size and divergence for DivExplorer). For this reason, we compare them on the *artificial* dataset of Section 4.4, where the divergent itemsets $a = b = c = 0$ and $a = b = c = 1$ are well

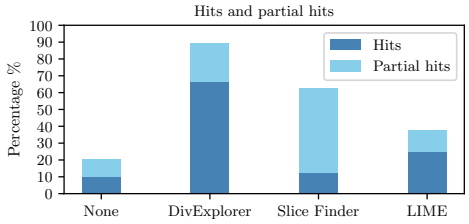


**Figure 12: User study results. Percentage of hits for the injected bias according to the provided information.**

characterized and drive both explorations equally. Unless differently specified, we executed Slice Finder with its default parameters. We use the predicted and class labels as inputs to DivExplorer. We used a Random Forest classifier with default parameters to provide the loss function required by Slice Finder. DivExplorer minimum support is set to 0.01. For Slice Finder, we set *degree* to 3 to obtain itemsets of length 3.

Since DivExplorer does not enforce parallel execution, for a fair comparison we turned it off in Slice Finder. In this case, DivExplorer mean execution time is 4s, 4.5 times faster than Slice Finder. If parallel execution is turned on (max-workers=4), DivExplorer is 3.5 times faster than Slice Finder.

DivExplorer successfully identifies (*a=0,b=0,c=0*) and (*a=1,b=1,c=1*) as the itemsets with the highest FPR divergence. Slice Finder finds all 6 subsets of length 2 of ( *a=0,b=0,c=0*) and (*a=1,b=1,c=1*). These subsets are already highly "problematic" (in our terms, they have high divergence). Hence, Slice Finder's search stops. The stopping criterion based on "problematicity" fails to identify the two itemsets that are the true source of divergence, returning their many subsets instead. If the threshold of the effect size is increased to 1.65, Slice Finder identifies the true source of divergence. This search requires 18s with 1 worker.

### 6.6 User study

We conducted a user study to assess how useful is the information provided by DivExplorer in helping users identify data subgroups with anomalous behavior. The study compared the information provided by DivExplorer, Slice Finder, and LIME [21], the latter as a relevant representative method from the Explainable AI domain. For the study, we considered the *COMPAS* dataset. We performed a controlled experiment in which we artificially injected bias in a subgroup, and we measured how well the information provided by the different tools allowed users to identify the injected bias. Specifically, in the training set we injected bias in the subgroup characterized by the pattern {age_cat>45, charge_degree=M}, changing all outcomes to recidivate, and we trained a (biased) multi-layer perceptron neural network on such modified dataset. We then analyzed the misclassifications of such a biased classifier on the (unmodified) testing dataset with DivExplorer, Slice Finder, and LIME.

The study involved 35 undergraduate computer science students, who had some knowledge of the notions of classifiers, and false positive and negative errors. We divided the users in four groups. Group 1 was shown examples of correctly and mis-classified instances drawn uniformly at random. The other groups were shown the same information as group 1, and in addition:

- Group 2: the top 6 itemsets and their $\Delta_{FPR}$ computed by DivExplorer with $s = 0.05$, and the global item divergence.
- Group 3: the itemsets computed by Slice Finder and their impact factors, with *degree=3* and default parameters.
- Group 4: LIME explanations for 8 correctly classified and 8 mis-classified instances drawn uniformly at random.

The amount of information received by groups 2, 3, and 4, was similar, amounting to a couple of pages in PDF format. We asked the participants to select the top 5 itemsets that are most affected by errors. We consider the following metrics in evaluating the user answers: *hit* and *partial hit*. The metric *hit* $\in \{0,1\}$ is 1 if the user included the injected bias itemset {age_cat>45, charge_degree=M}, and 0 otherwise. The metric *partial hit* $\in \{0,1\}$ is 1 if the user included the items {age_cat>45} or {charge_degree=M}, and 0 otherwise.

Figure 12 summarizes the percentage of hits and partial hits for each user group. The information provided by DivExplorer was the one that led the users most directly to identify the injected bias, with a combined hit rate of 88.89%. In group 1, 20% of the users completely or partially identified the biased subgroups by carefully inspecting the misclassified instances. In group 3 (Slice Finder), most of the users only partially selected the biased itemset. Slice Finder with default parameters identifies the two items composing the itemset as already highly 'problematic', and prunes the search. Finally, in group 4 the explanations provided by LIME led to a combined hit rate of 37.5%. Interestingly, LIME had more full hits than Slice Finder, in spite of LIME's goal being to provide classification explanations, rather than identifying critical subgroups.

## 7 CONCLUSIONS AND FUTURE WORK

In this paper, we propose the notion of divergence over itemsets as a measure of different classification behaviors in subsets of a given dataset. A solid theoretical foundation, based on Shapley values, is proposed to quantify divergence contributions, both for pattern and dataset. The concept of divergence also allows capturing interesting item behaviors, for example corrective items. An efficient algorithm for divergence computation is provided and an extended experimental evaluation shows the effectiveness and efficiency of the proposed approach.

Given the generality of the divergence notion, as future work we plan to study its extension to other data science tasks, including, e.g., the preprocessing tasks.

## ACKNOWLEDGMENTS

This work has been partially supported by the SmartData@PoliTO center on Big Data and Data Science.

## REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 487–499.

[2] TensorFlow Model Analysis. 2018. Introducing TensorFlow Model Analysis: Scaleable, Sliced, and Full-Pass Metrics. https://medium.com/tensorflow/introducing-tensorflow-model-analysis-scaleable-sliced-and-full-pass-metrics-5cde7baf0b7b.

[3] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine Bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[4] Abolfazl Asudeh, Zhongjun Jin, and H.V. Jagadish. 2019. Assessing and Remedying Coverage for a Given Dataset. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. 554–565. https://doi.org/10.1109/ICDE.2019.00056

[5] Denis Baylor, Eric Breck, Heng-Tze Cheng, Noah Fiedel, Chuan Yu Foo, Zakaria Haque, Salem Haykal, Mustafa Ispir, Vihan Jain, Levent Koc, Chiu Yuen Koo, Lukasz Lew, Clemens Mewald, Akshay Naresh Modi, Neoklis Polyzotis, Sukriti Ramesh, Sudip Roy, Steven Euijong Whang, Martin Wicke, Jarek Wilkiewicz, Xin Zhang, and Martin Zinkevich. 2017. TFX: A TensorFlow-Based Production-Scale Machine Learning Platform. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada) *(KDD '17)*. Association for Computing Machinery, New York, NY, USA, 1387–1395. https://doi.org/10.1145/3097983.3098021

[6] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, Seema Nagar, Karthikeyan Natesan Ramamurthy, John T. Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. 2019. AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development* 63, 4/5 (2019), 4:1–4:15.

[7] Ángel Alexander Cabrera, Will Epperson, Fred Hohman, Minsuk Kahng, Jamie Morgenstern, and Duen Horng Chau. 2019. FairVis: Visual analytics for discovering intersectional bias in machine learning. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 46–56.

[8] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. 2019. Automated Data Slicing for Model Validation: A Big data - AI Integration Approach. *IEEE Transactions on Knowledge and Data Engineering* (2019). https://doi.org/10.1109/TKDE.2019.2916074

[9] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. 2019. Slice Finder: Automated Data Slicing for Model Validation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. 1550–1553.

[10] Cynthia Dwork and Christina Ilvento. 2018. Group fairness under composition. In *Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency (FAT* 2018)*.

[11] James R Foulds, Rashidul Islam, Kamrun Naher Keya, and Shimei Pan. 2020. An intersectional definition of fairness. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1918–1921.

[12] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.* 51, 5, Article 93 (Aug. 2018), 42 pages. https://doi.org/10.1145/3236009

[13] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining Frequent Patterns without Candidate Generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, Weidong Chen, Jeffrey F. Naughton, and Philip A. Bernstein (Eds.). ACM, 1–12. https://doi.org/10.1145/342009.335372

[14] Zhongjun Jin, Mengjing Xu, Chenkai Sun, Abolfazl Asudeh, and H. V. Jagadish. 2020. MithraCoverage: A System for Investigating Population Bias for Intersectional Fairness. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 2721–2724. https://doi.org/10.1145/3318464.3384689

[15] Minsuk Kahng, Dezhi Fang, and Duen Horng Chau. 2016. Visual exploration of machine learning results using data cube analysis. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 1–6.

[16] Michael J. Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*. PMLR, 2569–2577. http://proceedings.mlr.press/v80/kearns18a.html

[17] M. Lichman. 2013. UCI Machine Learning Repository.

[18] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4765–4774. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[19] Giulio Morina, Viktoriia Oliinyk, Julian Waton, Ines Marusic, and Konstantinos Georgatzis. 2019. Auditing and Achieving Intersectional Fairness in Classification Problems. arXiv:1911.01468 [cs.LG]

[20] Eliana Pastor, Luca de Alfaro, and Elena Baralis. 2021. DivExplorer project page. https://divexplorer.github.io/

[21] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 1135–1144. https://doi.org/10.1145/2939672.2939778

[22] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[23] Pedro Saleiro, Benedict Kuester, Abby Stevens, Ari Anisfeld, Loren Hinkson, Jesse London, and Rayid Ghani. 2018. Aequitas: A Bias and Fairness Audit Toolkit. *arXiv preprint arXiv:1811.05577* (2018).

[24] Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games* 2, 28 (1953), 307–317.

[25] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. *Introduction to Data Mining (2nd Edition)* (2nd ed.). Pearson.

[26] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2019. Errudite: Scalable, Reproducible, and Testable Error Analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 747–763. https://doi.org/10.18653/v1/P19-1073