arXiv:2003.04792v1 [cs.AI] 10 Mar 2020

Metafeatures-based Rule-Extraction for Classifiers on Behavioral and Textual Data

A Preprint

Yanou Ramon[*]
Dpt. of Engineering Management
University of Antwerp

David Martens
Dpt. of Engineering Management
University of Antwerp

Theodoros Evgeniou
INSEAD Paris

Stiene Praet
Dpt. of Engineering Management
University of Antwerp

March 11, 2020

***K*eywords** Explainable Artificial Intelligence, Interpretable Machine Learning, Metafeatures, Comprehensibility, Global explanations, Rule-extraction, Classification, Big Behavioral data, Textual data, High-dimensional data, Sparse data, Decision tree

Abstract

Machine learning using behavioral and text data can result in highly accurate prediction models, but these are often very difficult to interpret. Linear models require investigating thousands of coefficients, while the opaqueness of nonlinear models makes things even worse. Rule-extraction techniques have been proposed to combine the desired predictive behaviour of complex "black-box" models with explainability. However, rule-extraction in the context of ultra-high-dimensional and sparse data can be challenging, and has thus far received scant attention. Because of the sparsity and massive dimensionality, rule-extraction might fail in their primary explainability goal as the black-box model may need to be replaced by many rules, leaving the user again with an incomprehensible model. To address this problem, we develop and test a rule-extraction methodology based on higher-level, less-sparse "metafeatures". We empirically validate the quality of the rules in terms of fidelity, explanation stability and accuracy over a collection of data sets, and benchmark their performance against rules extracted using the original features. Our analysis points to key trade-offs between explainability, fidelity, accuracy, and stability that Machine Learning researchers and practitioners need to consider. Results indicate that the proposed metafeatures approach leads to better trade-offs between these, and is better able to mimic the black-box model. There is an average decrease of the loss in fidelity, accuracy, and stability from using metafeatures instead of the original fine-grained features by respectively 18.08%, 20.15% and 17.73%, all statistically significant at a 5% significance level. Metafeatures thus improve a key "cost of explainability", which we define as the loss in fidelity when replacing a black-box with an explainable model.

1 Introduction

Technological advances have allowed storage and analysis of large amounts of data and have given industry and government the opportunity to gain insights from thousands of digital records collected about individuals each day [50]. These "big, behavioral data"—characterized by large volume, variety, velocity and veracity and defined as data that capture human behavior through the actions and interactions of people [61]—have led to predictive modeling applications in areas such as fraud detection [70, 71], financial credit scoring [46, 20, 66], marketing [73, 50, 11] and political science [55]. Sources of behavioral data include, but are not limited to, transaction records, search query data, web browsing histories [52], social media profiles [55, 42], blog posts, online reviews, and smartphone

[*] Corresponding author.

sensor data (e.g., GPS location data) [50]. Textual data are also increasingly available and used. Example text-based applications are automatic identification of spam emails [5], objectionable web content detection [48] and legal document classification [13], just to name a few. Behavioral and textual data are very high-dimensional compared to traditional data, which is primarily structured in a numeric format and is relatively low-dimensional [52, 19, 50]. Consider an example to illustrate these characteristics. We want to predict health or personality traits [42] of users based on the Facebook pages they have "liked". A user is represented by a binary feature for each unique page, which results in an enormous feature space. However, each user only has liked a small number of pages, which results in an extremely sparse data matrix (almost all elements are zero).

Learning from behavioral and textual data can result in highly accurate models [38, 19]. A drawback of models, such as classifiers, learned from such ultra-big, ultra-sparse data, however, is that they can become very complex. The complexity arises from either the learning technique (e.g., deep learning) or the data, or both. It is essentially impossible to interpret classifications of nonlinear techniques such as nonlinear SVMs or deep neural networks. For linear models or decision trees, the most common approach to globally understand the model is to examine the estimated coefficients or to inspect the paths from root to leaf nodes. In the context of big, sparse data, however, even linear models are not straightforward to interpret because of the large number (thousands) of features each with their corresponding coefficient [48]. Moreover, one may question the comprehensibility of decision trees with thousands of leaf nodes. Alternatively, for linear models, we could inspect only the features with the highest weights. For sparse data, however, the coverage of the top-weighted features is extremely low, such that only a relatively small fraction of the classified instances are actually explained [52, 48]. [42], for example, explain models that predict personality traits using over 50,000 Facebook "likes" by listing the pages that are most related to extreme frequencies of the target classes. For example, the best predictors for high intelligence include Facebook pages "*The Colbert Report*", "*Science*" and "*Curly Fries*" [42]. However, because of the extreme sparsity of the data (in this sample, a user liked on average 170 out of 55,814 possible pages), one may question the usefulness of such feature listings for interpreting model predictions.

Explainability has emerged recently as a key business and regulatory challenge for Machine Learning adoption. The relevance of global interpretability of classification models is well-argued in the literature [45, 46, 37, 23, 3].[2] In the process of extracting knowledge from data, the predictive performance of classification models *alone* is not sufficient as human users need to understand the models to trust them, accept them and also improve them [69]. Both the US and the European Union are currently pushing towards a regulatory framework for trustworthy Articial Intelligence, and all global organizations such as the OECD and the G20 aim towards a human-centric approach [25]. In high-stake application domains, explanations are often legally required. In the medical domain, for example, computer-aided diagnosis can influence a patient's treatment, and for this reason, explanations are required by doctors to validate such systems. Also in the credit scoring domain, legislation such as the Equal Credit Opportunity Act in US Federal Law [68] prohibits creditors from discrimination and requires reasons for rejected loan applications. Also in lower-stake applications, such as (psychologically) targeted advertising [50, 52] or churn prediction [72], explanations are managerially relevant. Global interpretability allows to verify the knowledge that is encoded in the underlying models, known as "*knowledge verification*" [3, 34]. Models trained on big data may learn incorrect trends or may perpetuate social biases [11]. Furthermore, explanations give users more control of their virtual footprint. To many, privacy invasions via inferences are at least as troublesome as privacy invasions based on personal data [11]. [49] argue that insight into what data is being collected and the inferences that can be drawn from it, allows users to make more informed privacy decisions [49]. Lastly, global model explainability can help to induce new insights and to generate hypotheses or theories [61, 3].

Rule-extraction algorithms have been proposed in the literature to generate global explanations by distilling a comprehensible set of rules from complex models. However, rule-extraction in the context of ultra-high-dimensional and sparse data can be challenging, and, to our knowledge, has thus far received scant attention. Because of the data characteristics, rule-extraction might fail in their primary task (providing insight in the black-box model) as the complex model is only replaced by a set of hundreds or even thousands of rules, leaving the user again with an incomprehensible model.

This paper addresses the challenge that rule-extraction explanation models using fine-grained input data can have poor performance. Instead of focusing on rule-extraction techniques themselves, this paper leverages alternative higher-level feature representations—which we refer to as "metafeatures"—to improve the fidelity (approximation of the underlying model), explanation stability (same explanations for different training sessions - a concept we introduce, which for simplicity we will be calling just stability) and accuracy (correct predictions of the original observations) of the extracted rules. We propose desired properties of such metafeatures and study their relationship with the quality of the extracted rules. For simplicity, we only focus on classification. Our main claim is that metafeatures are more appropriate,

[2] Model explanations vary in scope: the method either generates global explanations or instance-level explanations [48, 58] We focus on global explanations that give insight in the model's predictions over all possible feature values and for all instances. Instance-level explanations, on the other hand, are tailored to one classification. For example, an *evidence counterfactual* explanation shows a minimal set of features such that, when removing these features, the predicted class changes [48, 58].

in specific ways we discuss, for extracting rules from classifiers on high-dimensional, sparse data than the original fine-grained features.

This paper's main contributions are threefold: (1) we propose a novel methodology for rule-extraction by exploring how higher-level feature representations (metafeatures) can be used for explaining a classification model learned from high-dimensional, sparse data; (2) we define a set of quantitative criteria to assess the explanation quality of rule sets in terms of fidelity, stability and accuracy; a key contribution is to also empirically study the trade-offs between these; and lastly, (3) we perform an in-depth empirical evaluation of the rule explanation quality with metafeatures using a set of behavioral and textual data and benchmark their performance against the rules extracted with the high-dimensional input data. One important finding is that metafeatures can improve a key "cost of explainability", which we define as the loss in fidelity[3] when replacing a black-box with an explainable model.

2 Related work

2.1 Rule-extraction

In the Explainable Artificial Intelligence (XAI) literature, rule-extraction falls within the class of post-modeling explainability methods. Post-hoc interpretability allows users to analyze a trained model in order to provide insights into the learned relationships [53]. The idea of rule-extraction is to train a comprehensible model (the *white-box*) to mimic the predictions of a more complex, underlying *black-box* model [23].[4] We define a black-box model as a nontransparent complex model from which it is not straightforward for a human interpreter to understand how predictions are made. In this paper, we consider all models (linear, rule-based or nonlinear) trained on high-dimensional data as black-box models (because of the high-dimensional and sparse nature of the data, see Section 2.2); which is different from existing research on rule-extraction that only considers highly-nonlinear models as black-boxes [3, 46, 45, 23]. In the machine learning literature, decision trees and rule learners have been argued to yield the most comprehensible models [69, 28], making them the good candidates to use as white-box models.

Rule-extraction can be used for two purposes. First and foremost, one may be interested in knowing the rationale behind decisions made by a classification model and verify whether the results make sense in practice. The goal is to extract comprehensible rules that closely mimic the black-box, a difference that is measured by what is called "fidelity". Alternatively, the goal can be to improve the "accuracy", namely, the generalization performance of decision tree or rule-based learners by approximating the black-box (e.g., by removing noise from the data) [46, 45, 34]. In this study, we report most results using fidelity instead of accuracy as our focus is on developing explainable models that "best mimic the black-box" – but all our analyses can be done also using accuracy as the main metric. We focus on so-called pedagogical rule-extraction methods that do not rely on the inner workings of the underlying classification model, and only make use of the input-output mapping defined by the model [3, 46, 34]. The idea behind this approach is that the similarity between the black-box and white-box model can be substantially improved by presenting the labels predicted by the black-box model $\hat{Y}=\{\ y_i^{BB}\ \}_{i=1}^n$ to the white-box algorithm, instead of the original labels $Y=\{y_i\}_{i=1}^n$ associated with the training data set [37].

2.2 Challenges of rule-extraction for high-dimensional data

The vast majority of the rule-extraction literature has focused on improving the explanatory power and scalability of rule learners. However, despite some very impressive and promising work [3, 46, 37, 23], the rule-extraction techniques are mostly validated on low-dimensional, dense data sets, such as the widely-used set of benchmark data coming from the UCI Machine Learning repository [6]. These data sets have feature dimensions going up to 50 features. We identify at least three challenges regarding rule-extraction from classifiers on high-dimensional, sparse data:

1. *Complexity of the extracted rules.* In the context of large, sparse data, rule-extraction might to provide insight in the black-box model as the black-box model is only replaced by a large set of rules [34, 46]. [64] applied rule-extraction on (real-world) high-dimensional text data and show that rule learners can closely approximate the underlying model, but at the cost of being very complex (hundreds of rules) [64, 35].
2. *Computational complexity.* It is not straightforward for every existing rule learning algorithm to be used for high-dimensional data, because the learning task might become computationally too demanding [3, 64, 45]. Some rule algorithms, such as Ripper [16], are not able to computationally deal with problem instances with large input data [64].

[3] As the main goal of explainability methods is to best mimic "black-boxes", we focus on fidelity instead of, say, accuracy, as discussed below. Our methodology and analysis can be adapted to also study the "accuracy cost of explainability".

[4] We will interchangeably refer to this as the black-box model or the underlying model.

3. *Fine-grained feature comprehensibility.* [23] questions the usefulness of extracted rules from models trained on high-dimensional data. For example, when rules are learned from a model initially trained on a "bag-of-words" representation of text documents, the antecedents in a rule include individual words out of context. This may reduce the sematic comprehensibility of these rules. Also for other digital traces one leaves behind, we may question the interpretability of a single action (e.g., a single credit card transaction, a single Facebook "like") taken out of context, that is, the collection of all behavioral actions of an individual.

Because of the above challenges, it is questionable *whether fine-grained behavioral and textual features are the best representation for extracting rules in order to achieve the best explanation quality (e.g., fidelity, stability, accuracy).* This motivates our approach to use a metafeatures representation instead. However, it is not clear a priori whether such a representation can improve the explanation quality. This is a key empirical question we study.

3 Metafeatures as Explanation Drivers

3.1 Motivation

We consider *explanation drivers* as the set of factors whose impact on a target (e.g., the predicted label for classification) is described by an explanation (e.g., a set of rules). The most common drivers are the original input features of the classification model, however, these are not necessarily the best explanation drivers. Behavioral and textual data suffers from sparsity and, for this reason, the original fine-grained features (individually) may exhibit less discriminatory power to explain the black-box model [12]. A sparse feature that contains a specific behaviour or word in the training data may not be aligned with out-of-sample features in the test data [12]. Moreover, because of the low coverage that characterizes such sparse features, a single feature is not expected to "explain" much of the classifications of the underlying model. The feature will only be active for a small fraction of all data instances, and therefore, the coverage of the rule is likely to be low [63, 64].[5]

We address the data sparsity by mapping the fine-grained, sparse features onto a higher-level, less-sparse feature representation, which we refer to as "metafeatures". Metafeatures have been used by others, for example, in the context of natural language processing tasks [12] to cope with data sparsity. In [12], metafeatures are derived by clustering similar features by their frequency in large data sets. Metafeatures have also been used for explaining image classification: using input pixels as explanation drivers are not straightforward to interpret, hence researchers have proposed to use a patch of similar pixels (super-pixels) for generating explanations of the model's image classifications [75]. [40] also argue that explanations in the domain of image classification should use higher-level, human-friendly concepts rather than the original pixels.

3.2 Desired properties

In this section, we propose some desired properties for the metafeatures. We define a mapping function $h(\mathbf{x})$: $X_{FG} \rightarrow X_{MF} \subset \mathbb{R}^k$, where X_{MF} is a space of metafeatures and k is the dimensionality of this new representation. We propose the following set of properties for engineering metafeatures:

1. *Low dimensionality.* We want the dimensionality k of the metafeatures to be smaller than the dimensionality m of the original input space: $k << m$. A lower feature dimension may lead to more stable explanation rules [2]. Moreover, the computational burden for extracting rules with metafeatures is likely to be much lower compared to rule-extraction with high-dimensional data [3, 64].
2. *High density.* This property relates to the coverage of a metafeature, which we want to be higher compared to the coverage of fine-grained features [12]. In other words, there should be more instances for which a metafeature is *active* (nonzero value) compared to the fine-grained features. Higher density of the features may increase fidelity and accuracy of cognitively simple explanations because of the higher coverage of rules.
3. *Faithfulness.* This is in line with prior research suggesting that the representation of the original data instances in terms of metafeatures should preserve relevant information to discriminate between the predicted labels $\hat{Y}$ [2]. It is important that the extracted rules using metafeatures can reach a high level of faithfulness with respect to the true predictions being made, because this may result in a better approximation of the underlying model.
4. *Mutual exclusivity.* In prior research, this property indicates that features should be representable with only few non-overlapping metafeatures (referred to as the "*diversity*" property by [2]). In other words, a data instance

[5] The coverage of a feature is defined as the number of data instances that have a nonzero value for this feature, whereas the coverage of a rule is defined as the number of instances that are classified by this rule. For sparse data sets, both feature and rule coverages tend to be low.

that is described by 100 behavioral features should only be described by a few mutually exclusive concepts. In our Facebook running example, this implies that a user that has liked 100 Facebook pages can be equivalently represented by a smaller set of Facebook categories that the user liked (e.g., "*Entrepreneurship*").

5. *Semantic comprehensibility.* This property is in line with the "*grounding*" property proposed by [2], that is, metafeatures should have a human-comprehensible interpretation. For example, Facebook "likes" can be grouped into specific categories (e.g., "*Female Fashion*", "*Entrepreneurship*") and GPS location data can be categorized into different venue types (e.g., "*Sports venues*", "*Concert halls*"). This property is rather subjective in nature and depends on the application domain and the expectations of the users [34, 35, 9, 76].

In Section 6 we evaluate the performance of rules extracted with such metafeatures and discuss links between the empirical results and the first three properties (dimensionality, density and faithfulness). However, we do not explicitly touch upon the properties of mutual exclusivity and semantic comprehensibility. The latter would require experimentation with people, a direction to explore if indeed metafeatures improve the quality of explanations in the other dimensions we study here, a first step we focus on.

4 Metafeatures-based Rule-extraction

We introduce and validate a methodology for global rule-extraction from a complex model learned from large-scaled, sparse data. The steps of the rule-extraction methodology are summarized in **Fig. 1** and are discussed in the sections below.

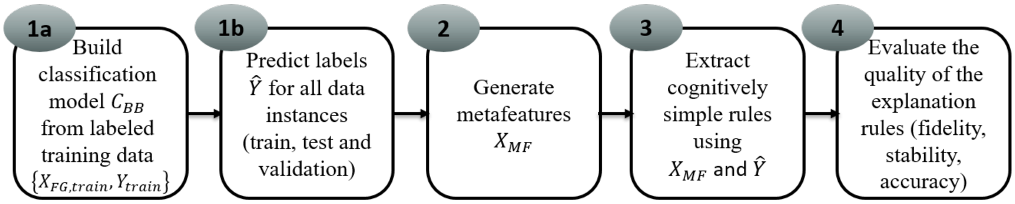

1a
Build classification model C_{BB} from labeled training data $\{X_{FG,train}, Y_{train}\}$
1b
Predict labels $\hat{Y}$ for all data instances (train, test and validation)
2
Generate metafeatures X_{MF}
3
Extract cognitively simple rules using X_{MF} and $\hat{Y}$
4
Evaluate the quality of the explanation rules (fidelity, stability, accuracy)

Figure 1: Proposed rule-extraction methodology using metafeatures.

4.1 Model building and predicting new labels

From the original fine-grained behavioral data we train and test the underlying black-box model (C_{BB}). The model is trained on a part of the data (the training set) and hyperparameters are optimized using a separate holdout set (the validation set). Finally, the generalization performance of the black-box model is evaluated on an unseen part of the data (the test set). The trained black-box model is used to make predictions $\hat{Y}$ for all instances in our dataset (training, validation and test data), which will thereafter be used to train our white-box model (C_{WB}).

4.2 Generating metafeatures

We need to specify a feature transformation process to group features with similar properties together in "metafeatures" [12] and that satisfies the desired properties defined in Section 3.2. There are various approaches for generating metafeatures from the original input data, either by manually generating them using domain knowledge [53] or by automatically obtaining them by means of data-driven feature engineering techniques, such as (un)supervised dimensionality reduction.[6]

Domain-based metafeatures One way of generating metafeatures from the original input features is to group features together in *domain-based* categories manually crafted by experts [53, 2]. For example, for the Facebook "Like" data set, individual Facebook pages can be grouped together in predetermined categories, for example, pages related to "*Machine Learning*" or "*Entrepreneurship*". This human-selected set of metafeatures can then be used to extract simple rules to explain model predictions, which tests the relative importance of such domain-based metafeatures.

Data-driven metafeatures Alternatively, metafeatures can be generated via a more data-driven approach, such as dimensionality reduction. The idea is to represent the data in a lower dimensional space without too much loss of information. An important assumption we make is that the resulting data-driven metafeatures are semantically comprehensible.

[6] We use DomainMF, DDMF and FG as abbreviations for domain-based metafeatures, data-driven metafeatures and fine-grained features respectively.

A variety of dimensionality reduction techniques exist, varying from supervised to unsupervised techniques [53]. Well-established unsupervised methods are Non-negative Matrix Factorization (NMF), Principal Component Analysis (PCA) and Singular Value Decomposition (SVD). In this paper, we use NMF as the data-driven approach to obtain metafeatures because of its property to produce feature vectors with only non-negative values.[7] In most real-life applications, negative components or subtractive combinations in the representation are physically meaningless. Incorporating the nonnegativity constraint thus facilitates the interpretation of the extracted metafeatures in terms of the original data [74]. More specifically, NMF decomposes the original data matrix $X_{n \times m}$ with n unique instances and m unique features into two non-negative matrices $U_{n \times k}$ and $W_{k \times m}$ such that: $X \approx UW$. The objective function is to minimize $||X - UW||^2$ with respect to U and W, subject to the constraints U, $W \geq 0$ [43]. Several algorithms exist for solving this optimization problem [74], we will make use of the *Scikit-learn* (Python) implementation based on coordinate descent.[8]

NMF is a popular technique because it groups together related features. The quality of the components depends on the number of extracted metafeatures k: a value of k that is set too high results in many highly-similar topics, whereas a low value of k tends to generate overly-broad metafeatures. The intended goal to generate metafeatures in this paper is using them for rule-extraction, and consequently, we optimize the number of k such that the out-of-sample fidelity of the rules is maximal (we use a validation set to finetune the value of k). We consider values of k from 10 up to 1000. Note that we should not be concerned about generating too many metafeatures because we only need to interpret the ones that are part of the final explanation rules.

For generating metafeatures based on the NMF method we first approximate the original training data X_{FG} by two non-negative matrices U and W for a given number of metafeatures k (**step 1** in **Fig. 13**). Matrix W maps each metafeature to the original fine-grained features. To ensure mutual exclusivity (see Section 3.2) we will transform W into a binary matrix W_{binary}, where 1 represents the maximum element for each column (fine-grained feature) of W and all other elements are 0 (**step 2** in **Fig. 13**). Moreover, working with the binary matrix W_{binary} led to better results than using the original, non-binary matrix W. Next, we map the original matrix X_{FG} to X' by multiplying X_{FG} with the transposed binary matrix W_{binary} (**step 3** in **Fig. 13**). Finally, matrix X' is normalized over the original number of features per instance to become matrix X_{DDMF} which represents the metafeatures per instance (**step 4** in **Fig. 13**). Again, we found that the normalized matrix X_{DDMF} produced better results than utilizing the original matrix X' or even a binary matrix derived from X'.

4.3 Cognitively simple rule-extraction

Both rule and decision tree induction methods can be used for pedagogical rule-extraction. Since the trees can be converted into rules, we consider tree algorithms as rule-extraction techniques [46, 35, 47]. Among the most wide-spread algorithms for tree and rule induction are C4.5 [56], CART [7], ID3 [57], CN2 [14], RIPPER [16], AQ [51], 1R [32] and CHAID [39]. A full review of these techniques is beyond the scope of this paper, but we will shortly describe CART [9], as this is the tree induction technique used in this study. It should be noted that both algorithmic complexity and software availability have impacted the choice of the rule-extraction method used in this study [3, 37].

CART can be used for both classification and regression problems and uses information theoretic concepts, more specifically, it uses as splitting criterion the Gini index, which measures the impurity of a node. The best split is the one that reduces the impurity the most. The Gini index is an impurity measure that is based on the divergences between the probability distributions of the target classes. For a binary target, the Gini impurity at node t is defined as follows,

$$Gini(t) = \sum_{i=1}^{2} p(i,t)(1 - p(i,t)) \tag{1}$$

where $p(i,t)$ is the fraction of instances belonging to class i at node t. We apply CART to the data where the output is changed to the black-box predicted class label $\hat{Y}$.

The size of explanations can be used as a proxy for comprehensibility, based on the general assumption in the literature that smaller models or rule sets are more comprehensible than larger ones [31, 35]. Restricting the rule set is also motivated by research on how people make decisions - for example, based on relatively simple rules to avoid excess

[7] It is not our research objective to investigate which data-driven approach is best to obtain metafeatures for rule-extraction. Doing so would multiply the complexity of our analysis without necessarily changing our main conclusions directionally. Of course, other approaches can be investigated and may generate better results, making this a research direction that is worth to explore.

[8] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html

[9] CART is readily available from the *Scikit-learn* library (Python).

cognitive effort [29, 31] due, for example, to cognitive limitations [65]. In the context of consumer decision-making for example, [31] argue that decision rules should incorporate "cognitive simplicity": rule sets should consist of a limited number of rules, each with a small number of antecedents. Interestingly, in the machine learning literature, these constraints (e.g., limited number of rules) are used for complexity control of the model to minimize in-sample overfitting and improve out-of-sample inferences [31]. Finally, it is important to note that the concept of comprehensibility comprises many different aspects, such as the size of the explanation, but also the specific application context and subjective opinion and expectations of the end user, which makes it more difficult to measure comprehensbility in a generic way [35, 9, 76]. In line with cognitive simplicity arguments [31], we restrict the rule sets to have up to 32 rules each consisting of (at most) 5 antecedents (equivalent to a tree depth of 5).[10]

4.4 Evaluation of extracted rules

As discussed, we evaluate the quality of the extracted explanation rules using a number of criteria, which we now explain in detail.

Fidelity First and foremost, the extracted rules are evaluated on how well they approximate the classification behaviour of the underlying model. Fidelity measures the ability of the extracted rules to mimic the model's classification behaviour from which they are extracted. Let $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$ represent the labeled data instances and y_i^{WB} and y_i^{BB} respectively the white-box and black-box predictions. Fidelity is expressed as the fraction of instances for which y_i^{WB} and y_i^{BB} are the same [34]:

$$fidelity^{WB} = Prob(y_i^{BB} = y_i^{WB} | \mathbf{x}_i \in X) \tag{2}$$

While most of our analysis is using, for simplicity, fidelity, we can extend the fidelity to "*f-score fidelity*" (*f-fidel*), which may be a more appropriate evaluation metric to measure how well the rules approximate the underlying classification model when there is a (predicted) class imbalance. The *f-fidel* is defined as the harmonic mean between $precision$ and $recall$ (w.r.t. the predicted labels y^{BB} rather than the true labels y) and may be a more suitable performance metric because of the unbalanced nature of the explicit predictions. More precisely, the formula of *f-fidel* is $\frac{2 \cdot precision \cdot recall}{precision + recall}$, where the $precision$ of the classifier is the fraction of positively-predicted instances that is correctly classified and the $recall$ refers to the fraction of positive instances that is correctly classified as a positive. (Note once more that here we use the predicted labels as the target variable).

Explanation stability A second important factor is *explanation stability* - which for simplicity we will call just stability. Users, businesses, or regulators may have a hard time accepting explanations that are unstable (meaning small changes in the data lead to large change in explanations of the predictions made by the model), even if the explanation model has shown to be very accurate and comprehensible [69]. Stability is closely related to the consistency criterion proposed by [3], who define a rule set as consistent if under different training sessions, the rule sets produce the same classifications on unseen examples. Prior research has shown that explanation methods that rely on high-dimensional input data tend to be less robust compared to techniques that operate on higher-level features [2]. For this reason, we expect that the extracted rules with metafeatures are more stable over different training sessions compared to the rules with fine-grained features. [67] distinguishes two types of stability: syntactic and semantic stability. Semantic stability is often measured by estimating the probability that two models learned on different training sets, will give the same prediction to an instance. On the other hand, syntactic similarity measures how similar two explanations are (e.g., the overlap in the features used in two different explanation rule sets), and is more specific to a particular explanation representation [67]. We argue that the latter is the most relevant type of stability in the context of explaining predictive models. To the best of our knowledge, it remains an open question how to measure syntactic stability for different explanation representations, such as rules and trees. We propose the following method to measure the stability of extracted rules from decision trees:

- Step 1: Generate 10 subsamples from the training data using bootstrapping.
- Step 2: Train a decision tree from each bootstrap and the corresponding predicted labels $\hat{Y}$. Obtain 10 decision trees and keep track of the features that are part of the decision tree in a set.

[10] **Fig. 10** (in Appendix) shows an example decision tree extracted from a classification model that predicts gender using *Facebook* data [55]. **Fig. 11** shows a decision tree that is extracted from the same classification model, but it uses domain-based metafeatures (the *Facebook* categories already available in the data) to explain how predictions are made. **Fig. 12** shows a decision tree using data-driven metafeatures. An illustration of how we interpreted the metafeatures is provided in **Table 4**.

- Step 3: Make $\frac{n!}{k!(n-k)!} = \frac{10!}{2!(8)!} = 45$ pairwise comparisons of the decision trees using the Jaccard coefficient. For two feature sets A and B (respectively from decision trees DT_A and DT_B), the Jaccard coefficient is defined as: $J(A,B) = |A \cap B|/|A \cup B|$. The Jaccard coefficient will be equal to 1 if the sets are equal and 0 if they are disjoint.
- Step 4: Compute the average Jaccard coefficient to approximate the stability of the explanation rules.

Accuracy Rule-extraction has also been used to increase the predictive performance of intrinsically-interpretable (white-box) models, measured by *accuracy*. [46] have shown that rules that mimic the behaviour of an underlying, better-performing model can become more accurate compared to the rules learned from the original data with ground-truth labels. Accuracy is defined as the fraction of correctly classified observations [34]:

$$accuracy^{WB} = Prob(y_i = y_i^{WB} | \mathbf{x}_i \in X) \tag{3}$$

5 Experimental setup

The experiments in this paper explore the performance of extracted rules with the original features versus higher-level metafeatures. We evaluate the performance on a suite of classification tasks using high-dimensional behavioral and textual data sets. **Fig. 2** summarizes the experimental procedure.

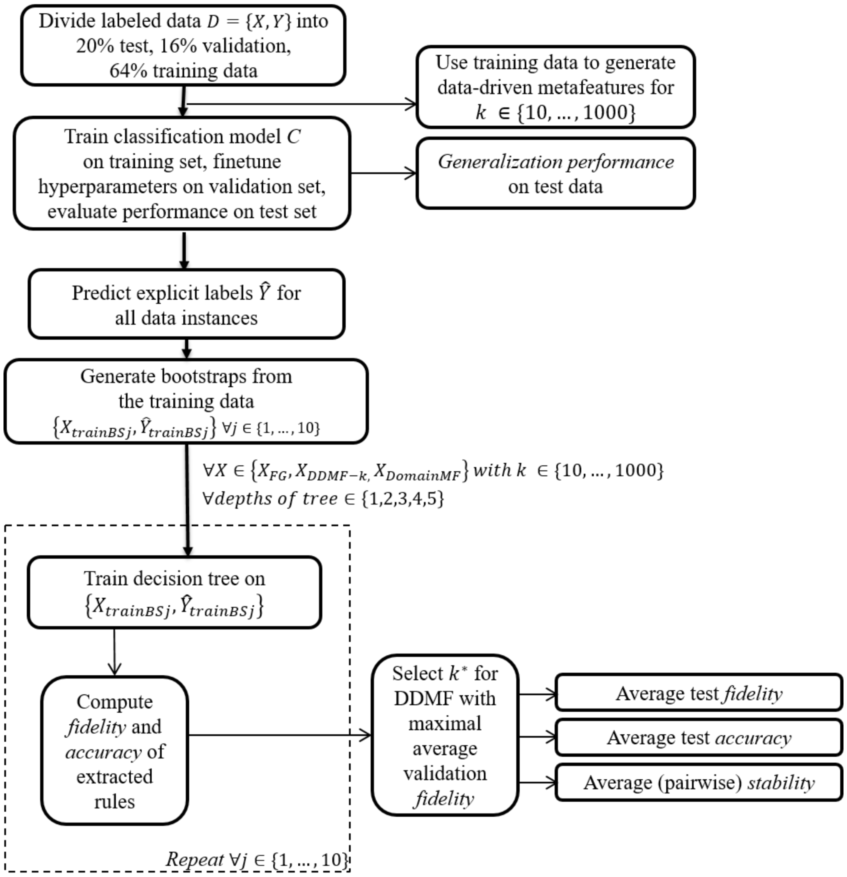

Divide labeled data $D = \{X, Y\}$ into 20% test, 16% validation, 64% training data
Use training data to generate data-driven metafeatures for $k \in \{10, \ldots, 1000\}$
Train classification model C on training set, finetune hyperparameters on validation set, evaluate performance on test set
Generalization performance on test data
Predict explicit labels $\hat{Y}$ for all data instances
Generate bootstraps from the training data $\{X_{trainBSj}, \hat{Y}_{trainBSj}\}$ $\forall j \in \{1, \ldots, 10\}$
$\forall X \in \{X_{FG}, X_{DDMF-k,} X_{DomainMF}\}$ $with\ k \in \{10, \ldots, 1000\}$
$\forall depths\ of\ tree \in \{1,2,3,4,5\}$
Train decision tree on $\{X_{trainBSj}, \hat{Y}_{trainBSj}\}$
Compute *fidelity* and *accuracy* of extracted rules
Repeat $\forall j \in \{1, \ldots, 10\}$
Select k^* for DDMF with maximal average validation *fidelity*
Average test *fidelity*
Average test *accuracy*
Average (pairwise) *stability*

Figure 2: Experimental procedure of evaluating explanation rules using different feature representations (fine-grained features (FG), data-driven metafeatures (DDMF) and domain-based metafeatures (DomainMF)), using different rule set complexity settings, over 10 bootstrapped samples of the training data.

5.1 Data sets and Models

Our experimental data comprise 7 behavioral and 2 textual data sets. The data sets are summarized in **Table 1**. The Facebook "like" data collected by [55] (*Facebook*) contains likes from over 6,000 individuals in Flanders (Belgium)

Table 1: Characteristics of the data sets: data type (Type: behavioral/textual), classification task (Target), number of instances (Instances), number of features (Features), number of domain-based metafeatures (DomainMF), balance of the target b (fraction of instances with a "positive" class label), and sparsity of the data p (fraction of zero feature values in the data matrix).

Dataset	Type	Target	Instances	Features	DomainMF	b	p
Facebook	B	gender	6,733	5,357	50	32.42%	98.19%
Movielens1m	B	gender	6,040	3,883	18	28.29%	95.76%
Yahoomovies	B	gender	7,642	11,915	n.a.	71.13%	99.76%
Movielens100	B	gender	943	1,682	n.a.	71.05%	93.69%
Tafeng	B	gender	31,640	23,719	n.a.	45.23%	99.90%
Libimseti	B	gender	137,806	166,353	n.a.	44.53%	99.93%
20news	T	topic	18,846	41,356	n.a.	4.24%	99.87%
Airline	T	sentiment	14,640	5,183	n.a.	16.14%	99.82%
Flickr	B	comments	100,000	190,991	n.a.	36.91%	99.99%

Table 2: Performance of black-box classification models: accuracy, f-score, precision and recall. The last column shows the optimal hyperparameter value (regularization parameter C for L2-LR).

Dataset	accuracy	f-score	precision	recall	$\mathbf{HP_{opt}}$
Facebook	85.97%	78.35%	79.91%	76.85%	0.01
Movielens1m	78.06%	61.31%	60.69%	61.95%	0.01
Yahoomovies	76.78%	83.51%	82.70%	84.33%	0.1
Tafeng	67.69%	64.98%	67.59%	62.55%	0.1
Libimseti	93.05%	92.53%	99.97%	86.11%	0.001
Movielens100	73.55%	81.48%	82.71%	80.29%	0.1
20news	96.66%	61.11%	60.74%	61.49%	100
Airline	89.58%	66.96%	64.51%	69.59%	1
Flickr	81.22%	75.36%	79.61%	71.54%	10

and is used to predict gender. The *Movielens1m* and *Movielens100* [30] data sets contain movie ratings from users of the MovieLens website. We focus on the task of predicting the gender of these users. The *Yahoomovies* data [78] has a similar setting, and also from these movie ratings, we predict gender. Another behavioral data set is the *Libimseti* data [8], which contains data about profile ratings from users of the Czech social network Libimseti.cz. The prediction task is, again, the gender of the users. The *Tafeng* data [33] consists of fine-grained supermarket transactions, where we predict the age of customers from the products they have purchased. The *Flickr* data set [10] contains millions of Flickr pictures and users have marked them as favorite or not. The target variable is the popularity (number of comments) of a picture. The *20news* data [77] contains about 20,000 newsgroup documents. For this data, the task is to predict whether a document belongs to the topic *Atheism*. Lastly, the *Airline* data [1] contains Twitter data about U.S. airlines, and the task is to predict positive sentiment.

All data have a high-dimensional feature space up to hundreds of thousands of features. *Movielens_1m, Movielens_100k* and *Airline* have lower-dimensional feature spaces compared to the other data sets. The large sparsity values p for all data indicate that the number of active features is very small compared to the total number of possible active features.

We train Logistic Regression models with $l2$-regularization (L2-LR)[11] using the *Scikit-learn* library (Python). For training the classification model, we use 80% of the data and 20% of the data is used for testing purposes. For finetuning hyperparameters (HP) of the model, we use a validation data set. More specifically, the regularization parameter C of the L2-LR model is selected by using a separate hold-out set (20% of the training data), and by selecting the C for which the accuracy on the validation set is maximal.

Measuring accuracy in practice requires discrete class label predictions, which we obtain by comparing the predicted probabilities to a threshold value t and assigning instances with a predicted probability that exceeds this threshold a positive predicted label. In practice, the choice of the threshold value t depends on the costs associated with false positives and false negatives. In our study, the exact misclassification cost are unknown, and for this reason we compute the threshold value t such that the fraction of instances that are classified as positive equals the fraction of positives in

[11] Logistic Regression has proven to be the best-performing state-of-the-art classification method for very-high-dimensional, sparse data [19].

the training data set [44]. **Table 2** indicates the generalization performance of all models for each data set. Note that we also report the f-score, precision and recall.

To extract decision trees using the CART algorithm, we use the *DecisionTree* model of the *Scikit-learn* library in Python. For controlling the complexity of the extracted rules, or equivalently, the depth of the tree, we play around with the max_depth parameter. We let the depth of the tree vary from 1 to 5 such that the extracted rule sets remain cognitively simple (which we motivated in Section 4.3).

6 Experimental results

In this section, we compare rules extracted using the original features with those extracted using metafeatures, across different classification tasks, data sets and evaluation criteria. As mentioned, our main goal is to better understand how metafeatures affect these different criteria as well as their trade-offs. Specifically, we address the following questions:

6.1 First, our central question is how do rules extracted using metafeatures to explain black-box classification models on high-dimensional, sparse data compare to rules extracted using the original features across the different criteria (fidelity, stability, accuracy)?

6.2 Second, how does the "cost of explainability", defined as the loss in fidelity when we restrict the complexity of the explainable model, vary over different complexity settings?

6.3 Finally, to what extent do both the fidelity and stability of rules extracted with metafeatures depend on a key parameter of our methodology, namely the number of generated metafeatures k?

6.1 Are data-driven metafeatures a better alternative than fine-grained features for explaining models learned from high-dimensional, sparse data?

Table 3 shows the fidelity (on test data), stability and accuracy (on test data) of rules with FG features and rules with DDMF. **Fig. 3** further summarizes these results by showing (a) the difference in test fidelity, (b) the difference in test f-fidel, (c) the difference in stability and (d) difference in test accuracy of rules with FG features and the DDMF representation for every data set.

One of the first key questions related to the performance of the rules is "what is the cost of explainability", which we measure as $100\% - fidelity$, as we want our explanation rules to mimic the black-box as closely as possible. Overall, our results indicate that the cost is lower for DDMF than for the FG-based rules. The rules with DDMF achieve a higher number of wins for both the fidelity (8 in contrast to 1) and f-fidel (7 in contrast to 2). We reach the same conclusion about fidelity when we use a one-tailed Wilcoxon signed-rank test [22] to make a statistical comparison between the fidelity of rules with FG vs DDMF features. The test is performed with a sample size of 9 data sets and for the results in **Table 3**. We find a test statistic T=2 (which is smaller than the critical value T_c=8), hence the difference in fidelity between DDMF and FG is statistically significant at a 5% significance level. In other words, we conclude that the "cost of explainability" of using metafeatures for rule-extraction is, on average across data sets, lower compared to using fine-grained features.

One exception is the *20news* data, as the fidelity values for this data are very high, whereas the f-fidel results are relatively low. This is probably because of the severe class imbalance (4.24%) compared to the other data sets. This may make the fidelity criterion less suitable for this specific data set. Instead, we could have optimized the depth of the tree and the k of the DDMF on the f-fidel as measured on the validation set.[12] If we perform a Wilcoxon signed-rank test without the *20news* data, we have a test statistic of T=0 for the differences in fidelity and T=3 for f-fidel, which makes the difference in both fidelity and f-fidel between DDMF and FG rules statistically significant at the 5% level. The *Flickr* data suffers from a similar issue: there is a large difference in fidelity and f-fidel between the rules with DDMF and FG, while the f-fidel for the rules with FG is only 0.49%. This may be because the explanation rules classify only 12 test instances (of over 190,000 instances) as positive, which results in a recall (and f-fidel) of approximately 0.

Moving to the stability of the explanations over different bootstraps, we observe from **Table 3** that, overall, the rules with DDMF are also more stable (8 wins versus 1)[13], and that this difference in stability is statistically significant at a 5% significance level. This may be expected - but whether it is true is an empirical question - to be due to the higher coverage of the DDMF compared to the FG features. Moreover, the dimensionality of the DDMF is relatively much lower, which together with the higher coverage may explain why over different bootstrapped samples the same DDMF

[12] However, for simplicity, we only used fidelity for all data sets.

[13] The relatively large difference in stability for the *Movielens100* data is most likely because of the difference in optimal tree depth (1 for FG and 3 for DDMF).

Table 3: Quality of extracted rules for explaining L2-LR model using fine-grained features (FG) and data-driven metafeatures (DDMF) with optimal number of generated metafeatures k in parentheses. The best performance values (FG vs DDMF) are indicated in bold. Rules are extracted from 10 bootstrapped training samples and the average fidelity, f-fidel and accuracy on the test data are reported. The best complexity setting (tree depth) is shown in the last column. For *Facebook* and *Movielens1m*, we also report results for the domain-based metafeaturs (DomainMF).

Complexity: $\#rules \leq 32$

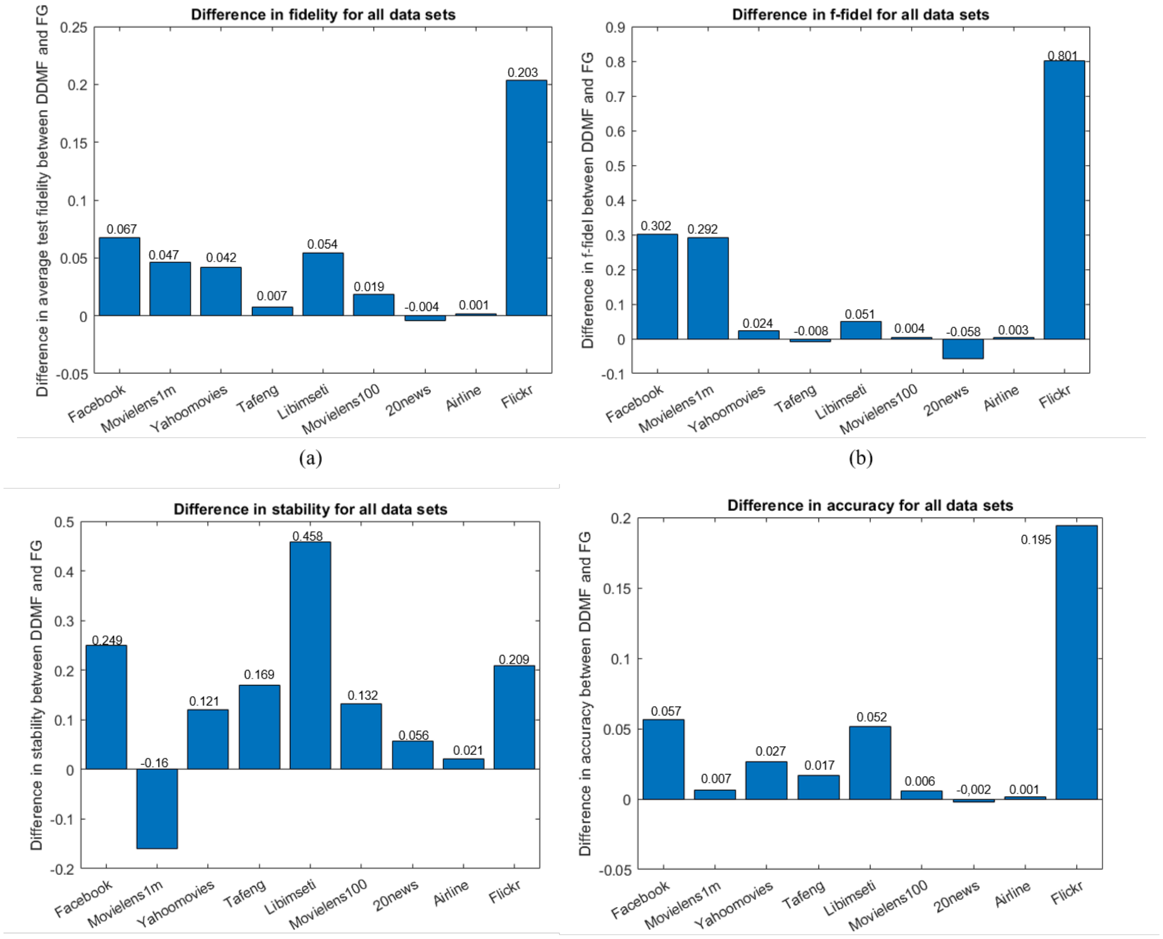
Dataset	Representation	fidelity	f-fidel	stability	accuracy	optimal depth
Facebook	FG	75.39%	42.00%	13.56%	74.11%	5
	DDMF (50)	**82.15%**	**72.19%**	**38.09%**	**79.76%**	4
	DomainMF	*73.77%*	*53.01%*	*45.88%*	*71.62%*	*5*
Movielens1m	FG	75.37%	32.14%	**60.33%**	73.60%	2
	DDMF (40)	**80.02%**	**61.38%**	44.29%	**74.26%**	5
	DomainMF	*71.24%*	*32.34%*	*49.28%*	*70.19%*	*4*
Yahoomovies	FG	76.34%	84.48%	20.55%	71.23%	5
	DDMF (100)	**80.54%**	**86.89%**	**32.60%**	**73.88%**	5
Tafeng	FG	67.32%	**60.73%**	22.35%	58.04%	5
	DDMF (50)	**68.06%**	59.98%	**39.28%**	**59.73%**	5
Libimseti	FG	87.71%	87.49%	28.64%	94.55%	5
	DDMF (10)	**93.11%**	**92.58%**	**74.43%**	**99.70%**	5
Movielens100	FG	69.63%	81.68%	8.89%	70.58%	1
	DDMF (100)	**71.48%**	**82.09%**	**22.13%**	**71.16%**	3
20news	FG	**96.19%**	**30.99%**	15.41%	**95.86%**	5
	DDMF (300)	95.78%	25.21%	**21.05%**	95.62%	5
Airline	FG	90.58%	64.58%	25.35%	87.79%	5
	DDMF (700)	**90.71%**	**64.85%**	**27.50%**	**87.92%**	5
Flickr	FG	64.01%	0.49%	21.64%	59.87%	3
	DDMF (30)	**84.34%**	**80.63%**	**58.01%**	**79.32%**	5
	#wins DDMF-FG	$8-1$	$7-2$	$8-1$	$8-1$	
	Average difference DDMF-FG	4.85%	15.69%	15.63%	5.08%	

are likely to appear in the global explanation. For the FG data, where the coverage of each feature is relatively low, the most informative features that are selected may vary more over the different bootstrapped samples.

When we compare the accuracy between the rules with DDMF and FG, we observe that the metafeatures-based rules result in more accurate predictions regarding the true labels Y (8 wins versus 1). Using a Wilcoxon signed-rank test, we find that the difference in accuracy is statistically significant at a 5% level. This shows that the metafeature approach outperforms the fine-grained one on all three (fidelity, accuracy and stability) measures.

Lastly, **Table 3** also shows that there is an average decrease of the loss in fidelity, accuracy, and stability from using metafeatures instead of the original fine-grained features by respectively 18.08%, 20.15% and 17.73%, all statistically significant at a 5% significance level.

In order to better understand what may drive some of the differences in the cost of explainability between DDMF and FG-based rules, we consider the Gini impurity reduction for different features, which we plot in **Figures 6** and **7**. The results in **Figures 6**, **7** and **3** indicate (visually) that the ratio in Gini impurity reduction of the best (or generally, the top k for some small k) metafeature and the best FG feature may relate to the difference in fidelity between rules using DDMF and FG features. For example, take the *Flickr* data set, for which the explanation rules with the metafeatures achieve a fidelity of 20.3 percentage points higher compared to the fine-grained rules. From **Fig. 6** we observe that the top DDMF hold much more information (larger Gini impurity reduction) than the top fine-grained features, which may



(a)

(b)

(c)

(d)

Figure 3: Difference in average (a) test data fidelity, (b) test data f-fidel, (c) stability and (d) test data accuracy in percentage points between extracted rules with data-driven metafeatures and fine-grained features.

explain the large difference in fidelity between the extracted rules. Indeed, the correlation coefficient between this ratio and the difference in fidelity between rules using DDMF vs FG from **Table 3** is 0.929.

Finally, **Fig. 4** plots the difference in test fidelity between rules with data-driven metafeatures and rules with fine-grained features against the maximum tree depth. Points above the horizontal line (where y=0) are data sets for which the rules with DDMF perform better. The graph clearly shows that for the majority of data, and over varying complexity settings, the DDMF representation performs better than the FG one (differences larger than 0). Only for the *Tafeng* and *20news* data sets, the differences are sometimes not positive, indicating that for these complexity settings, the average test fidelity for the rules with FG features is best. In general, from this plot, we can conclude that the findings of **Table 3** hold for varying complexity settings, and that the cost of explainability is generally lower for the DDMF representation compared to the FG representation.

6.2 How does the "cost of explainability" vary over different complexity settings?

Fig. 5 plots the average test data fidelity over 10 bootstrapped (training) samples against the maximal allowed decision tree depth for both DDMF and FG features-based rule-extracted models. We observe that, as one would expect, for all data sets, there is generally an increasing cost of explainability (or decreasing test fidelity) when we decrease the depth of the decision tree. For example, take the *Libimseti* data set using the FG data: here the increasing cost of explainability is very prominent (e.g., going from a depth of 1 to 2, the test fidelity increases with 15 percentage points). Interestingly, for some data sets, this fidelity-complexity trade-off is less severe. For example, for the *20news* and *Movielens100* data, the slopes of the curves are relatively flat. This also indicates that in some cases, there may not be much to gain by using

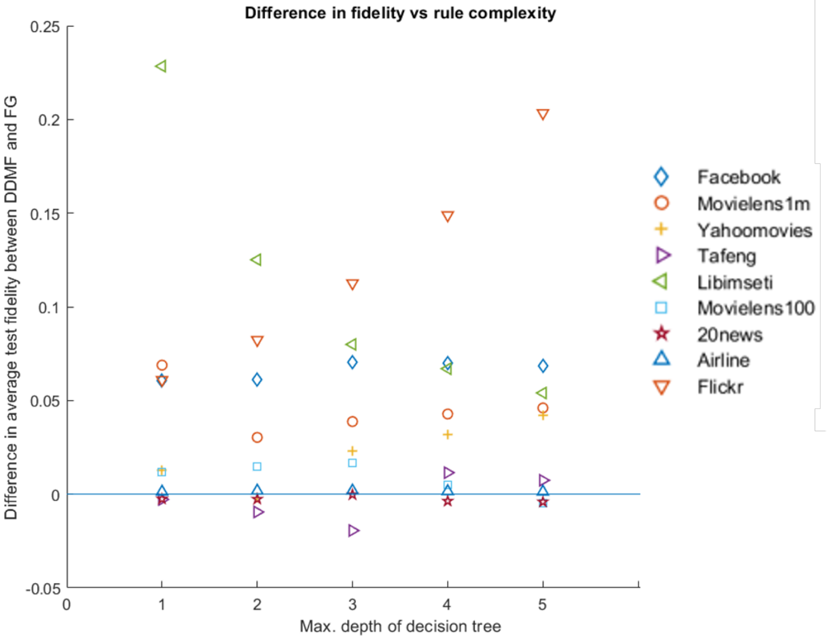
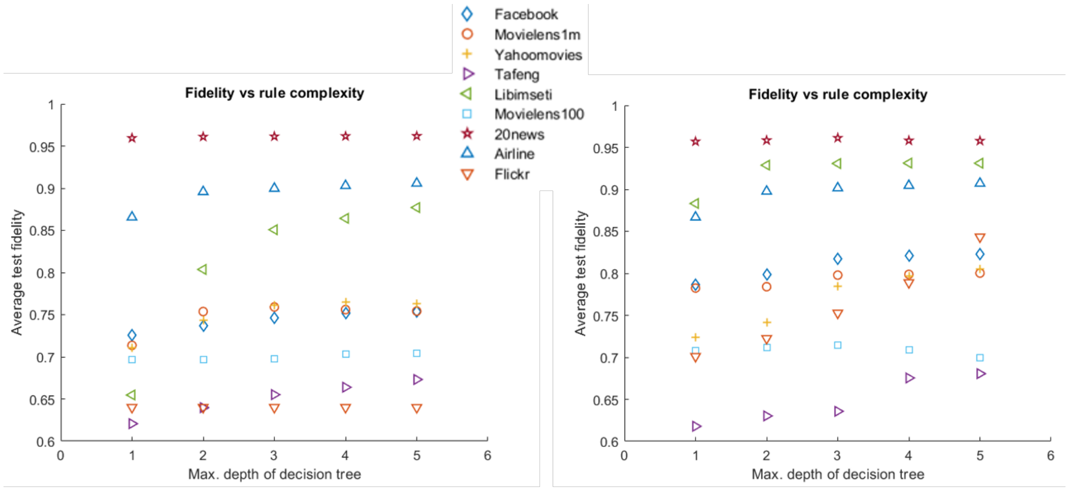


Figure 4: Difference in average test data fidelity of rules with DDMF and rules with FG features in percentage points for varying complexity settings (tree depths from 1 to 5).

a relatively "more complex" explainable model. Therefore, once one is willing to trade-off fidelity for explainability, in some cases, one might as well choose an "extremely" simple model. This is not the case for all data, however, as the example of *Libimseti* data set indicates.



(a)
FG

(b)
DDMF

Figure 5: Average test fidelity of rules with (a) FG features and (b) DDMF for varying complexity settings (depths from 1 to 5).

Finally, instead of generating metafeatures using a data-driven method, we can also rely on domain-based metafeatures, crafted by experts. The prominent advantage of this approach is that the resulting metafeatures are (by design) very comprehensible. However, these may not always be available. For example, we have such metafeatures for only 2 out of our 9 data sets, the *Facebook* and *Movielens1m* ones. When comparing DDMF with domain-based metafeatures for these two data sets, we see again that the cost of explainability is higher for the domain-based metafeatures compared to the data-driven metafeatures (**Table 3** shows that the rules with these domain-based metafeatures achieve, at best, test fidelity values of 73.77% for *Facebook* and 71.24% for *Movielens1m*), providing further support for using DDMF when developing explainable models for black-boxes.

6.3 How does the number of generated data-driven metafeatures impact fidelity and stability?

A key parameter in our methodology is the number of DDMF (k) used. We have been selecting the k that maximizes the fidelity of the explainable model on the validation data. As this k may be an important parameter that defines the dimensionality of the space where rule-extraction methods operate (and their performance), we also investigate to what extent the quality - both fidelity and stability - of rules extracted using DDMF depends on this parameter.[14] Although fidelity can be considered the most important evaluation criteria, in practice, one may wish to tune parameters such as k on a desired combination of fidelity, stability and accuracy[15] depending on the context.

Figures 8 and **9** show the fidelity (on the test data) and the stability (on different bootstrap samples of the training data) as the number of k of metafeatures used and the maximum rule complexity (maximum tree depth) allowed vary. Firstly, we note that for all data, the fidelity increases with a higher number of metafeatures up until a certain point, after which fidelity it decreases again. This turnover point varies per data set, and also depends on the complexity setting. Therefore, an important implication is to select the optimal number of metafeatures on a separate validation set, as we also do. Interestingly, fidelity behaves similarly to how (out-of-sample) accuracy typically does as complexity increases: for both measures there is some sort of "over-fitting" to the black-box training data in case ofwhen including too many features.

On the other hand, for stability, we observe that, overall, the stability of the extracted rules *decreases* with a higher number of k, especially when allowing for a larger maximum tree depth. For a lower value of k, the dimensionality of the metafeatures is lower and coverage is higher, making the same metafeature more likely to appear over several bootstrap samples (as also explained in Section 6.1). Interestingly, the stability of rules with FG features – also shown in the figures – tends to also decrease more steeply compared to the rules with metafeatures, another possible advantage of the latter. What we can also take away from these figures is that there is an important *fidelity-stability* trade-off. While fidelity generally increases at first with rule set complexity, stability does not. This may also impact the "optimal" number of generated metafeatures k, or any parameter selection for any explainability methodology.

7 Conclusion

The fine-grained level of the features that are typically observed in behavioral and textual datasets are of great value for predictive modeling. Dimensionality reduction techniques to come to a reduced set of "metafeatures" have been shown in the literature to lead to lower accuracies [15, 38]. On the other hand, we have shown empirically using a number of datasets that these metafeatures are of great value to *explain* the complex predictive models built on the fine-grained features. The rule sets built on data-driven metafeatures are able to better mimic the black-box models than those extracted using the fine-grained features. As such, metafeatures help to reduce the cost of explainability: small trees/rule sets that explain a large(r) percentage of the black-box's predictions (higher fidelity) can be obtained. Performance in terms of the stability, accuracy and complexity of the obtained trees is also shown to improve.

Our empirical results also show important trade-offs between the quality measures of the extracted rules that we considered. For example, more complex models tend to lead to higher fidelity but lower stability.

An interesting implication of our empirical findings is that one should carefully finetune any parameters of their explainability method, such as the number of generated metafeatures in our methodology, in order to obtain the desired trade-offs. In our case, increasing the number of generated metafeatures has shown to result in lower stability of the extracted rules, whereas the impact on fidelity is not straightforward and depends on the data set and the complexity setting.

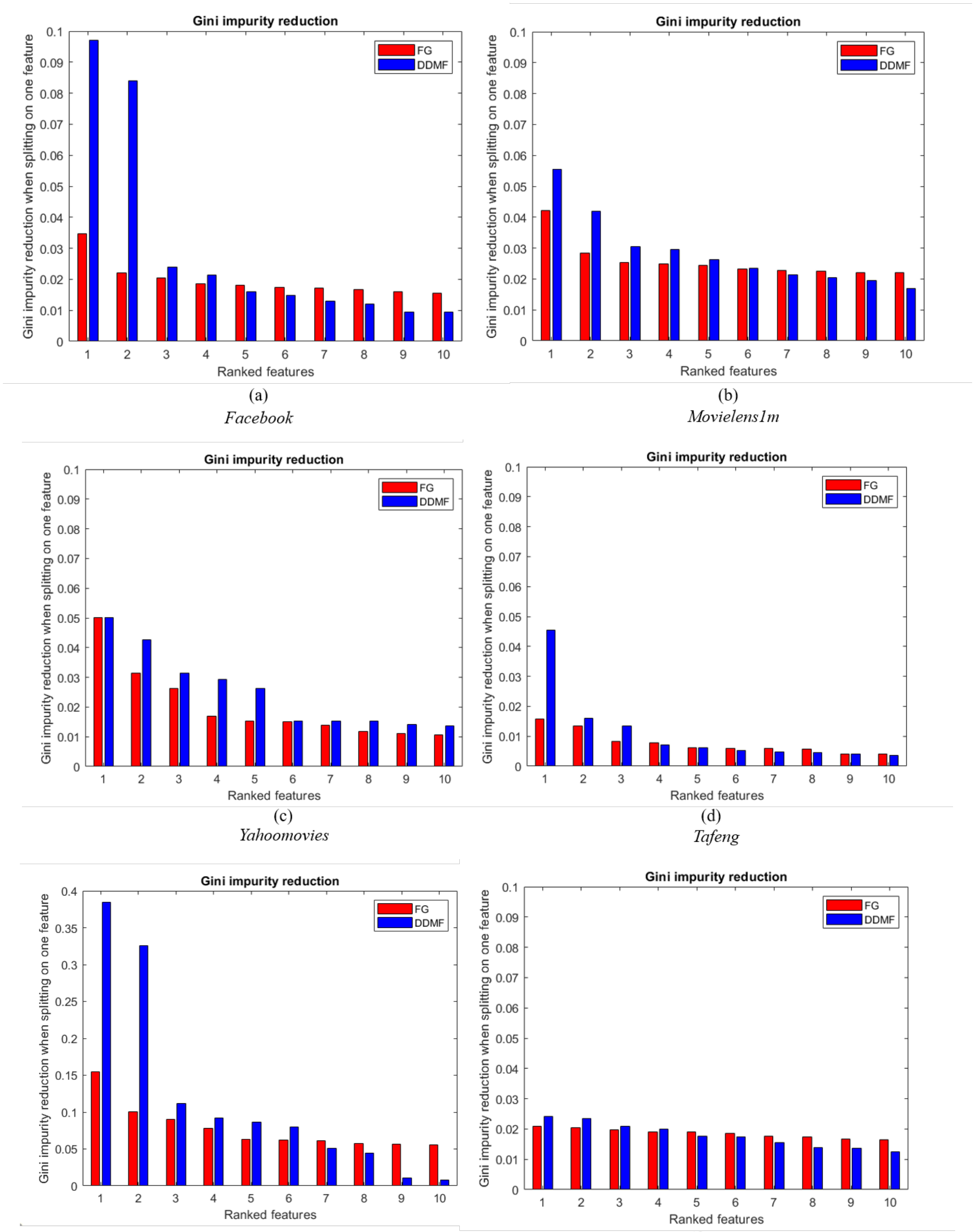
In this work we have defined a *key* "cost of explainability" based on the fidelity: explaining the black-box models leads to comprehensible rules but these do not mimic the black-box model completely. There are other types of costs of explainability, which we did not explicitly define: the computational cost to achieve such models, the cost of having a

[14] One can do such an analysis for other parameters, too, in general.

[15] As mentioned earlier, for simplicity, we focus on fidelity - namely how well we can mimic the black-box - instead of accuracy. All analyses can be done for either of the two - or for both – although trade-off decisions become more complex when one uses many criteria.

rule set with an accuracy that is lower than that of the black-box model, or the cost of presenting only one rule set, while other rule sets with similar fidelity and accuracy might also exist (related to the stability metric we discuss). Although these aspects are implicitly addressed in our paper, a more qualitative study on how these costs are perceived by users can be an interesting issue for future research. On a methodological level, this study could spur future research on the use of other feature engineering techniques to be used in rule-extraction. One interesting approach is to include the fidelity, accuracy, stability and complexity measures explicitly when defining the metafeatures.

Finally, our explanation approach for high-dimensional, sparse data has important practical implications for any setting where such data is available and explainability is an important requirement, be it for model acceptance, validation or model improvement. This paper could therefore potentially lead to a wider use of valuable behavioral and textual data in domains such as credit scoring and medical diagnosis among others.



(a)
Facebook

(b)
Movielens1m

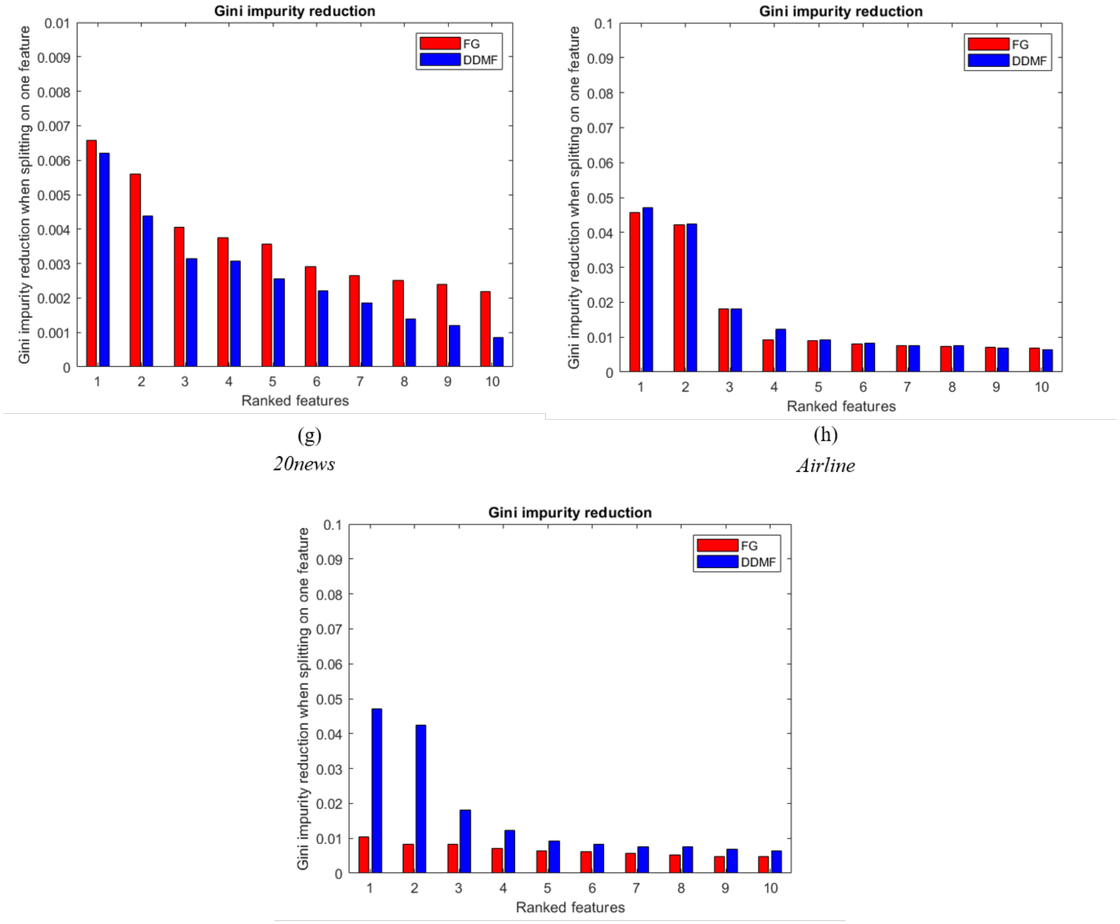
(c)
Yahoomovies

(d)
Tafeng

(e)
Libimseti

(f)
Movielens100

Figure 6: Top-ranked features with highest Gini impurity reduction for each data representation for the L2-LR model as the underlying black-box model.



(g)

20news

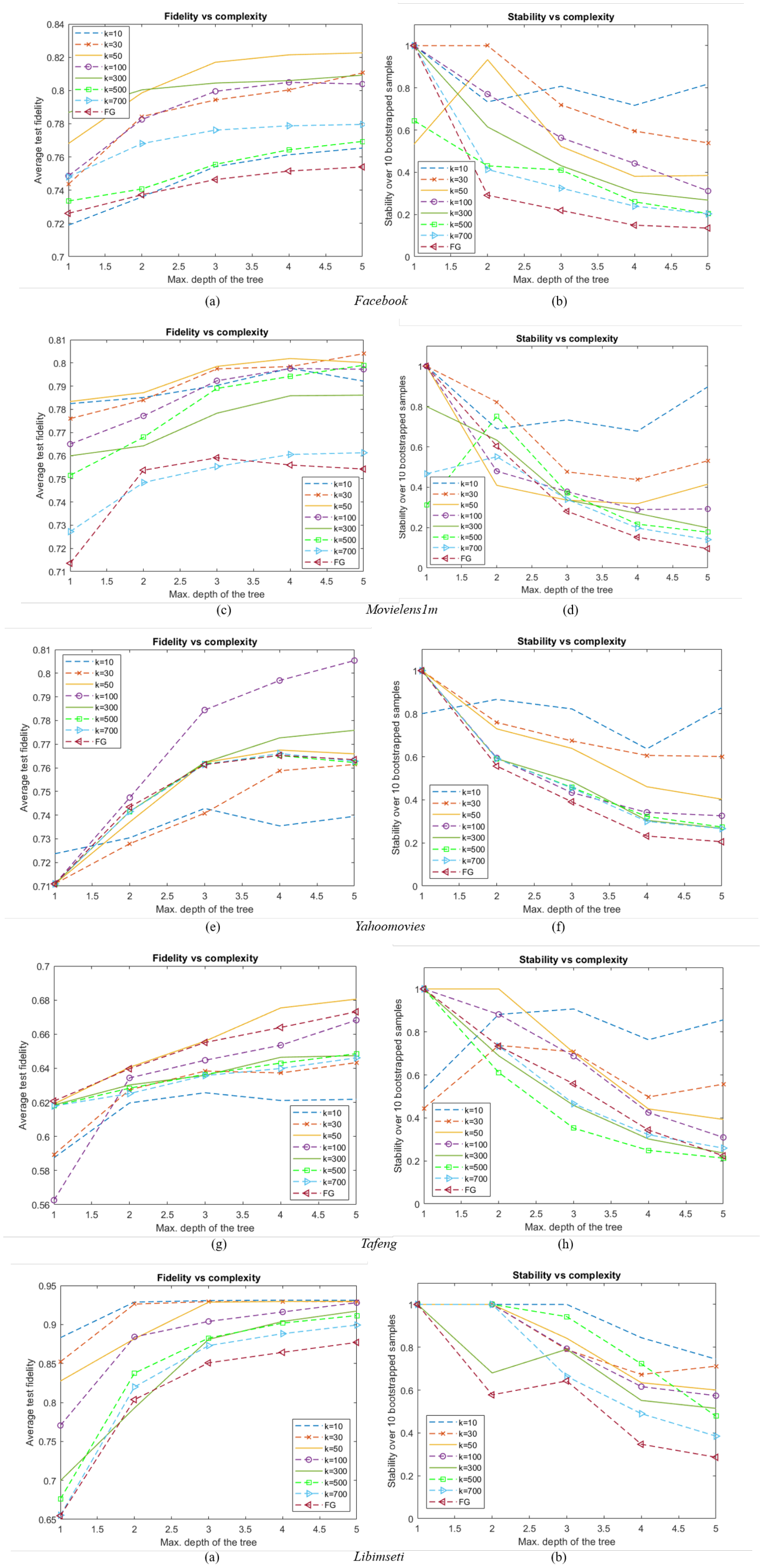
(h)

Airline

(g)

Flickr

Figure 7: Top-ranked features with highest Gini impurity reduction for each data representation for the L2-LR model as the underlying black-box model.



(a) *Facebook* (b)

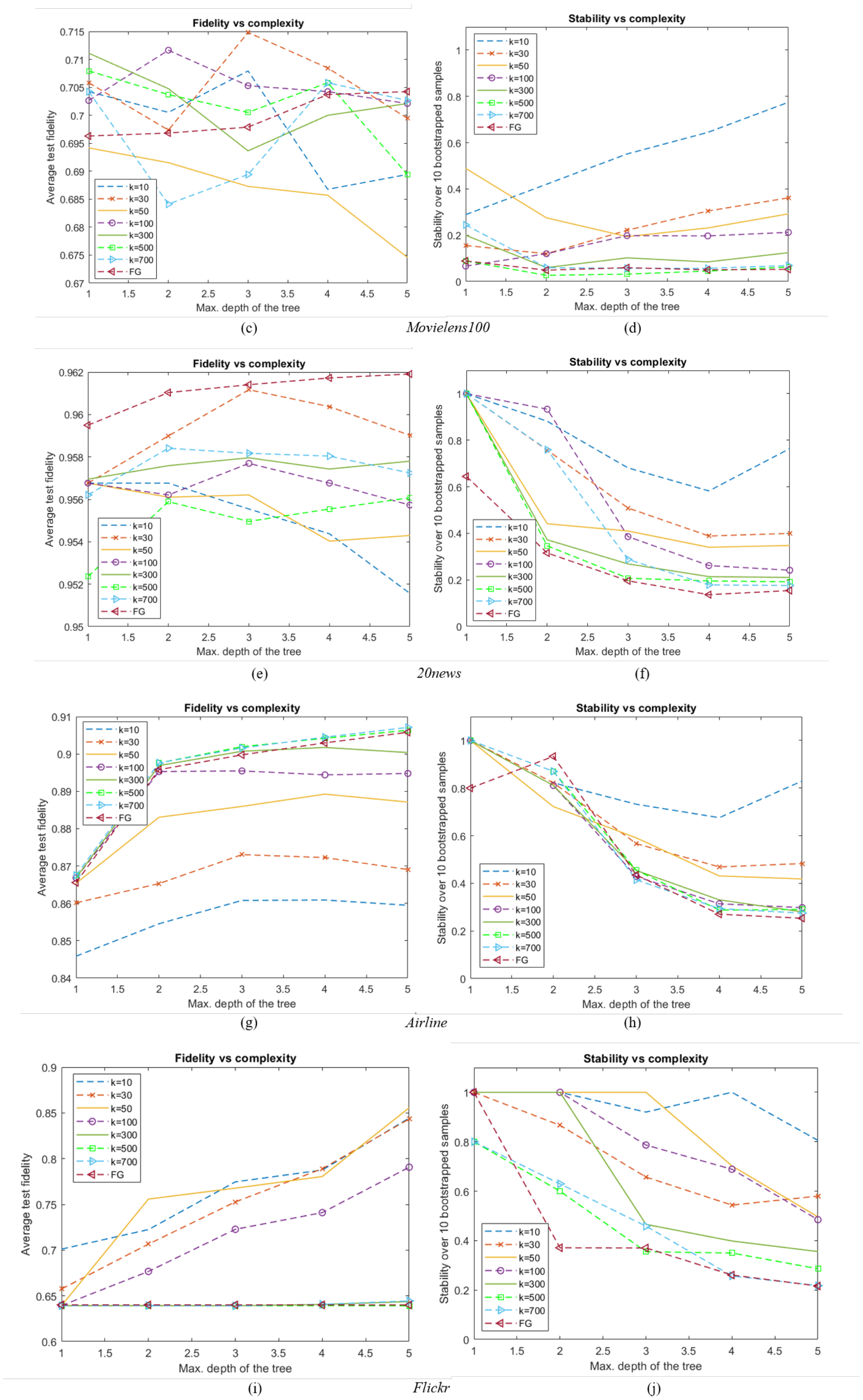
(c) *Movielens1m* (d)

(e) *Yahoomovies* (f)

(g) *Tafeng* (h)

(a) *Libimseti* (b)

Figure 8: Average test fidelity and stability for rules with data-driven metafeatures for varying number of generated metafeatures k for data sets *Facebook*, *Movielens1m*, *Yahoomovies*, *Tafeng* and *Libimseti*.



(c) *Movielens100* (d)

(e) *20news* (f)

(g) *Airline* (h)

(i) *Flickr* (j)

Figure 9: Average test fidelity and stability for rules with data-driven metafeatures for varying number of generated metafeatures k for data sets *Movielens100*, *20news*, *Airline* and *Flickr*.

References

[1] Airline Twitter Sentiment data, https://www.figure-eight.com/data-for-everyone/

[2] Alvarez-Melis, D, Jaakkola, TS, Towards Robust Interpretability with Self-Explaining Neural Networks, arxiv paper (2018)

[3] Andrews, R, Diederich, J, Survey and critique of techniques for extracting rules from trained artificial neural networks, Knowledge-Based Systems (1995)

[4] Arras L, Horn F, Montavon G, Müller K-R, Samek W,"What is Relevant in a Text Document": An Interpretable Machine Learning Approach, PLoS ONE, 12(8) (2017)

[5] Attenberg J, Weinberger K, Smola Q, Dasgupta A, Zinkevich M, Collaborative Email-Spam Filtering with the Hashing-Trick, Proceedings of the 6th Conference on Email and Anti-Spam, (2009)

[6] Bache, K, Lichman, M, UCI machine learning repository, School Inf. Comput. Sci., Univ. California, Irvine, CA, USA. Available: http://archive.ics.uci.edu/ml

[7] Breiman, L, Friedman, J, Olshen, R, Stone, C, Classification and Regression Trees, Chapman & Hall, New York (1984)

[8] Brozovsky L, Petricek V, Recommender System for Online Dating Service, Proceedings of Conference Znalosti, VSB, Ostrava, Czech Republic, (2007)

[9] Campbell, D, Task Complexity: a review and analysis, Academy of Management Journal, 13(1), pp 40–52 (1988)

[10] Cha M, Mislove A, Gummadi KP, A measurement-driven analysis of information propagation in the flickr social network, Proceedings of the 18th International World Wide Web Conference, doi:10.1145/1526709.1526806, (2009)

[11] Chen, D, Fraiberger, SP, Moakler, R, Provost, F, Enhancing Transparency and Control When Drawing Data-Driven Inferences About Individuals, Big Data, 5(3), pp 197–212 (2017)

[12] Chen, W, Zhang, M, Zhang, Y, Duan, X, Exploiting meta features for dependency parsing and part-of-speech tagging, Artificial Intelligence, 230, pp 173–191 (2016)

[13] Chhatwal R, Gronvall P, Huber N, Keeling R, Zhang J, Zhao H, Explainable Text Classification in Legal Document Review: a case study of explainable predictive coding, CoRR, abs/1904.01721, (2019)

[14] Clark, P, Niblett, T, The CN2 induction algorithm, Machine Learning, 3(4), pp 261–283 (1989)

[15] Clark J, Provost F, Dimensionality Reduction via Matrix Factorization for Predictive Modeling from Large, Sparse Behavioral Data, Data Min Knowl Disc, 33(4), pp 871–916 (2015)

[16] Cohen, WW, Fast effective rule induction, in A. Prieditis and S. Russell, editors, Proc. of the 12th International Conference on Machine Learning, pp 115–123, Tahoe City, USA, Morgan Kaufmann (1995)

[17] Craven, M, Shavlik, J, "Rule extraction: where do we go from here?" in Proc. Mach. Learn. Res. Group Working Paper, pp 1–6 (1999)

[18] Dalessandro, B, Chen, D, Raeder, T, Perlich, C, Williams, MH, Provost, F, Scalable hands-free transfer learning for online advertising, in Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, 99, pp 721–730 (2014)

[19] De Cnudde, S, Martens, D, Evgeniou, T, Provost, F, A benchmarking study of classification techniques for behavioral data, International Journal of Data Science and Analytics (2019)

[20] De Cnudde, S, Moeyersoms, J, Stankova, M, Martens, D, What does your Facebook profile reveal about your creditworthiness? Using alternative data for microfinance, Journal of the Operational Research Society, 70(10), pp 1–10 (2018)

[21] De Cnudde, S, Ramon, Y, Martens, D, Provost, F, Deep Learning on Big, Sparse, Behavioral Data, Big Data, 7(4), pp 286–307 (2019)

[22] Demsar, J, Statistical Comparisons of Classifiers over Multiple Data Sets, JMLR, 7(1), pp 1–30 (2006)

[23] Diederich, J, Rule Extraction from Support Vector Machines (2008)

[24] Drechsler, L, Sánchez, JCB, The Price Is (Not) Right: Data Protection and Discrimination in the Age of Pricing Algorithms, European Journal of Law and Technology, 9(3) (2018)

[25] European Commission White Paper. On Artificial Intelligence - A European approach to excellence and trust (2020)

[26] European Union, Counil Directive 2004/113/EC, art.3; European Union, Council Directive 2004/43/EC of 29 June 2000 implementing the principle of equal treatment between persons irrespective of racial or ethnic origin, OJ L 180 (19 July 2000), art.3.

[27] Fletcher, S, Islam, MZ, Comparing sets of patterns with the Jaccard index, Australasian Journal of Information Systems, 22 (2018)

[28] Freitas, AA, Comprehensible classification models: a position paper, ACM SIGKDD Explorations, 15(1) (2013)

[29] Gigerenzer, G, Goldstein, DG, Reasoning the fast and frugal way: models of bounded rationality, Psychological Review, 103(4) pp 650–669 (2016)

[30] Harper FM, Konstan JA, The MovieLens Datasets: History and Context, ACM Trans. Interact. Intell. Syst., 5(4) (2015)

[31] Hauser, JR, Toubia, O, Evgeniou, T, Befurt, R, Silinskaia, D, Disjunctions of Conjunctions, Cognitive Simplicity and Consideration Sets, Journal of Marketing Research (2009)

[32] Holte, RC, Very simple classification rules perform well on most commonly used datasets, Machine Learning, 11(1), pp 63–90 (1993)

[33] Hsu C-N, Chung H-H, Huang, H-S, Mining Skewed and Sparse Transaction Data for Personalized Shopping Recommendation, Machine Learning, 57(1), pp 35–59 (2004)

[34] Huysmans, J, Baesens, B, Vanthienen, J, Using Rule Extraction to Improve the Comprehensibility of Predictive Models, SSRN Electronic Journal, doi:10.2139/ssrn.961358 (2006)

[35] Huysmans, J, Dejaeger, K, Mues, C, Vanthienen, J, Baesens, B, An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models, Decision Support Systems, 51(1), pp 141–154 (2011)

[36] Joachims, T, Text Categorization with support vector machines: learning with many relevant features, Springer (cit. on pp 19,104,108,123,132) (1998)

[37] Junqué de Fortuny, E, Martens, D, Active Learning-Based Pedagogical Rule Extraction, IEEE Transactions on Neural Networks and Learning Systems, 26(11), pp 2664–2677 (2015)

[38] Junqué de Fortuny, E, Martens D, Provost F, Predictive modeling with big data: is bigger really better?, Big Data, 1(4), pp 215–226 (2013)

[39] Kass, G, An exploratory technique for investigating large quantities of categorical data, Applied Statistics, 29, pp 119–127 (1980)

[40] Kim, B, Wattenberg, M, Gilmer, J, Cai, C, Wexler, J, Viegas, F, Sayres, R, Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV), ICML 2018, arXiv:1711.11279 (2018)

[41] Knijnenburg, BP, Kobsa, SM, Jin, H, Counterfacting the negative effect of form auto-completion on the privacy calculus, in Thirty Fourth International Conference on Information Systems, Milan, Italy, pp 1–21 (2013)

[42] Kosinski M, Stillwell D, Graepel T, Private traits and attributes are predictable from digital records of human behavior, National Academy of Sciences, 110(15), pp 5802–5805 (2013)

[43] Lee, DD, Seung, HS, Algorithms for non-negative matrix factorization, In Advances in neural information processing systems, pp 556–562 (2001)

[44] Lessman, S, Baesens, B, Seow, HV, Thomas, L, Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research, EJOR (2015)

[45] Martens, D, Building Acceptable Classification Models for Financial Engineering Applications, Doctoral Thesis (2008)

[46] Martens, D, Baesens, B, Van Gestel, T, Vanthienen, J, Comprehensible credit scoring models using rule extraction from support vector machines, EJOR, 183, pp 1466–1476 (2007)

[47] Martens, D, Huysmans, J, Setiono, R, Vanthienen, J, Baesens, B, Rule Extraction from Support Vector Machines: An Overview of Issues and Application in Credit Scoring, Studies in Computational Intelligence (SCI), 80, pp 33–63 (2008)

[48] Martens, D, Provost, F, Explaining Data-Driven Document Classifications, MIS Quarterly, 38(1), pp 73–99 (2014)

[49] Matz, SC, Appel, R, Kosinski, M, Privacy in the Age of Psychological Targeting, Current Opinion in Psychology (2019)

[50] Matz, SC, Netzer, O, Using Big Data as a Window Into Consumer Psychology. Current Opinion in Behavioral Science, 18, pp 7–12 (2017)

[51] Michalski, R, On the quasi-minimal solution of the general covering problem, in Proceedings of the V International Symposium on Information Processing (FCIP 69), pp 125–128 (1969)

[52] Moeyersoms, J, d'Alessandro, B, Provost, F, Martens, D, Explaining classification models built on high-dimensional sparse data. In Workshop on Human Interpretability, Machine Learning: WHI 2016, June 23, 2016, New York, USA/Kim, Been [edit.], pp 36–40 (2016)

[53] Murdoch, WJ, Singh, C, Kumbier, K, Abbasi-Asl, R, Yu, B, Interpretable machine learning: definitions, methods, and applications, http://arxiv.org/abs/1901.04592 (2019)

[54] Oskarsdottir, M, Bravo, C, Sarraute, C, Vanthienen, J, Baesens, B, The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics, Applied Soft Computing, 74, pp 26–39 (2019)

[55] Praet, S, Van Aelst, P, Martens, D, I like, therefore I am : predictive modeling to gain insights in political preference in a multi-party system, Research paper, University of Antwerp, Faculty of Business and Economics, pp 1–34 (2018)

[56] Quinlan, JR, C4.5 Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)

[57] Quinlan, JR, Induction of Decision Trees, Machine Learning, 1, pp 81–106 (1986)

[58] Ramon, Y, Martens, D, Provost, F, Evgeniou, T, Instance-level explanation algorithms SEDC, LIME, SHAP for behavioral and textual data: a counterfactual-oriented comparison, Forthcoming in Advances in Data Analysis and Classification (2019)

[59] Ribeiro, MT, Singh, S, Guestrin, C, Why should I trust you? Explaining the predictions of any classifier, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 1135–1144 (2016)

[60] Shmueli G, Analyzing Behavioral Big Data: Methodological, practical, ethical, and moral issues, Quality Engineering, 29(1), pp 57–74 (2017)

[61] Shmueli, G, To Explain or To Predict?, Statistical Science, 25(3), pp 289–310 (2010)

[62] Simonyan, K, Vedaldi, A, Zisserman, A, Deep inside convolutional neural networks: visualising image classification models and saliency maps, Computing Research Repository, arXiv:1312.6034 (2013)

[63] Sommer, E, An approach to quantifying the quality of induced theories. In C. Nedellec, editor, Proceedings of the IJCAI Workshop on Machine Learning and Comprehensibility (1995)

[64] Sushil, M, Suster, S, Daelemans, W, Rule induction for global explanation of trained models, arXiv:1808.09744 (2018)

[65] Sweller, J, Cognitive load during problem solving: Effects on learning, Cognitive Science, 12(2), pp 257–285 (1988)

[66] Tobback, E, Martens, D, Retail credit scoring using fine-grained payment data, Journal of the Royal Statistical Society (2019)

[67] Turney, P, Technical Note: Bias and the Quantification of Stability, Machine Learning, 20, pp 23–33 (1995)

[68] US Federal Trade Commission, Your Equal Credit Opportunity Rights, Consumer Information (2003)

[69] Van Assche, A, Blockeel, H, Seeing the Forest Through the Trees: Learning a Comprehensible Model from an Ensemble (2007)

[70] Vanhoeyveld, J, Martens, D, Peeters, B, Customs fraud detection: assessing the value of behavioural and high-cardinality data under the imbalanced learning issue, Pattern analysis and applications, ISSN 1433–7541, New York, Springer, pp 1–21 (2019)

[71] Vanhoeyveld, J, Martens, D, Peeters, B, Value-added tax fraud detection with scalable anomaly detection techniques, Applied Soft Computing, ISSN 1568–4946, 86 (2020)

[72] Verbeke, W, Dejaeger, K, Martens, D, Hur, J, Baesens, B, New insights into churn prediction in the telecommunication sector: A profit driven data mining approach, European Journal of Operational Research, 218(1), pp 211–229 (2012)

[73] Verbeke, W, Martens, D, Mues, C, Baesens, B, Building comprehensible customer churn prediction models with advanced rule induction techniques, Expert Systems with Applications, 38 (2011)

[74] Wang, YX, Zhang, YJ, Nonnegative matrix factorization: A comprehensive review, IEEE Transactions on Knowledge and Data Engineering, 25(6), pp 1336–1353 (2012)

[75] Wei, Y, Chang, MC, Ting, T, Lim, SN, Lyu, S, Explain Black-box Image Classifications Using Superpixel-based Interpretation, IEEE, 24th International Conference on Pattern Recognition (ICPR) (2018)

[76] Wood, R, Task complexity: defintion of the construct, Organizational Behavior and Human Decision Processes, 37, pp 60–82 (1986)

[77] 20 newsgroups data, http://people.csail.mit.edu/jrennie/20Newsgroups/

[78] Yahoo Labs Movie Rating data, https://webscope.sandbox.yahoo.com

8 Appendix

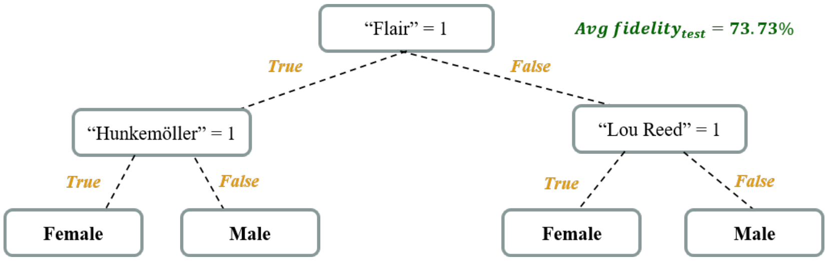


Figure 10: Example decision tree (depth=2) using the **fine-grained feature representation** to explain predictions of the underlying L2-regularized logistic regression model for *Facebook* data.

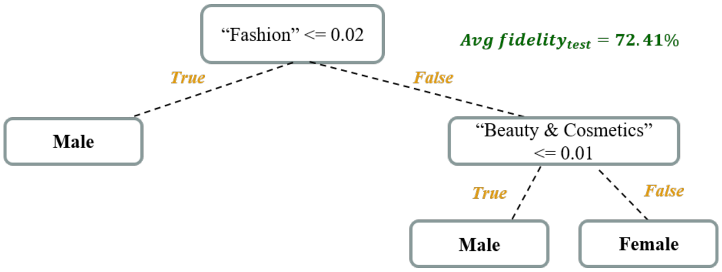

"Fashion" <= 0.02
$Avg\ fidelity_{test} = 72.41\%$
True
False
Male
"Beauty & Cosmetics" <= 0.01
True
False
Male
Female

Figure 11: Example decision tree (depth=2) using the **domain-based metafeatures** to explain predictions of the underlying L2-regularized logistic regression model for *Facebook* data.

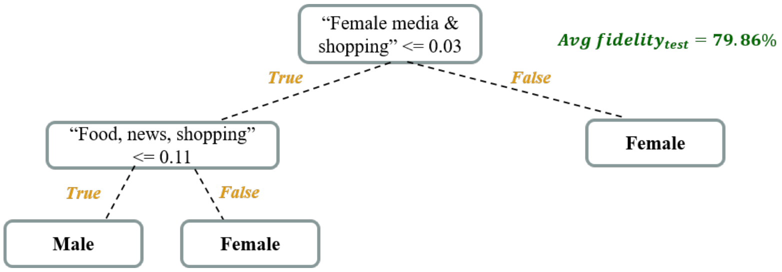

"Female media & shopping" <= 0.03
$Avg\ fidelity_{test} = 79.86\%$
True
False
"Food, news, shopping" <= 0.11
Female
True
False
Male
Female

Figure 12: Example decision tree (depth=2) using the **data-driven metafeatures** (k=50) to explain predictions of the underlying L2-regularized logistic regression model for *Facebook* data.

Table 4: Top-20 features with highest coefficient for two data-driven metafeatures that are part of the explanation rules (depth=2, see Fig. 12) for the L2-LR model on *Facebook* data. The “cluster names” at the bottom show our interpretation of these metafeatures, based on the largest coefficients.

Metafeature 1	Metafeature 2
Flair	IKEA
H&M	Dagelijkse kost
Gossip Girl	Decovry.com
ZARA	ZOO Planckendael
Tasty	Standaard Uitgeverij
ELLE België	Sandra Bekkari
Ben & Jerry’s	Lidl Belgium
Bokken voor bij het blokken	Vente-Exclusive.com
Jamie’s World	Lekker van bij ons
CHANEL	Radio 1
VIJF	Pascale Naessens
Route du Soleil	Natuurpunt
The Notebook	Ish Ait Hamou
Starbucks	Eén
Abercrombie & Fitch	Mme Zsazsa
UGent Confessions	Alpro
Pretty Little Liars	Libelle.be
Adele	Veritas
Hunkemöller	Marie Jo Lingerie
Beyoncé	Child Focus
Female media, shopping	Food, news, shopping

Step 1: NMF

$$\underset{\substack{X_{FG}\\(n\times m)}}{\begin{array}{cc} & \begin{array}{cccc}1 & 2 & \dots & m\end{array}\\ \begin{array}{c}1\\2\\\vdots\\n\end{array} & \begin{pmatrix}1 & 0 & \dots & 0\\ 1 & 0 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & 1\end{pmatrix}\end{array}} = \underset{\substack{U\\(n\times k)}}{\begin{array}{cc} & \begin{array}{cccc}1 & 2 & \dots & k\end{array}\\ \begin{array}{c}1\\2\\\vdots\\n\end{array} & \begin{pmatrix}u_{11} & u_{12} & \dots & u_{1k}\\ u_{21} & u_{22} & \dots & u_{2k}\\ \vdots & \vdots & \ddots & \vdots\\ u_{n1} & u_{n2} & \dots & u_{nk}\end{pmatrix}\end{array}} \times \underset{\substack{W\\(k\times m)}}{\begin{array}{cc} & \begin{array}{cccc}1 & 2 & \dots & m\end{array}\\ \begin{array}{c}1\\2\\\vdots\\k\end{array} & \begin{pmatrix}w_{11} & w_{12} & \dots & w_{1m}\\ w_{21} & w_{22} & \dots & w_{2m}\\ \vdots & \vdots & \ddots & \vdots\\ w_{k1} & w_{k2} & \dots & w_{km}\end{pmatrix}\end{array}}$$

Step 2: Binarization

$$W \xrightarrow{\forall i \in 1,\dots,k, \forall j \in 1,\dots,m:\; w_{ij}^{bin} = \begin{cases}1, & \text{if } w_{ij} = \max\limits_{1\le i\le k} w_{ij}\\ 0, & \text{otherwise}\end{cases}} W_{binary}$$

$$\underset{\substack{W_{binary}\\(k\times m)}}{\begin{array}{cc} & \begin{array}{cccc}1 & 2 & \dots & m\end{array}\\ \begin{array}{c}1\\2\\\vdots\\k\end{array} & \begin{pmatrix}0 & 0 & \dots & 0\\ 1 & 0 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & 1\end{pmatrix}\end{array}}$$

Step 3: Mapping to the metafeature space

$$\underset{\substack{X'\\(n\times k)}}{\begin{array}{cc} & \begin{array}{cccc}1 & 2 & \dots & k\end{array}\\ \begin{array}{c}1\\2\\\vdots\\n\end{array} & \begin{pmatrix}0 & 13 & \dots & 0\\ 0 & 0 & \dots & 6\\ \vdots & \vdots & \ddots & \vdots\\ 2 & 0 & \dots & 8\end{pmatrix}\end{array}} = \underset{\substack{X_{FG}\\(n\times m)}}{\begin{array}{cc} & \begin{array}{cccc}1 & 2 & \dots & m\end{array}\\ \begin{array}{c}1\\2\\\vdots\\n\end{array} & \begin{pmatrix}1 & 0 & \dots & 0\\ 1 & 0 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & 1\end{pmatrix}\end{array}} \times \underset{\substack{W_{binary}^T\\(m\times k)}}{\begin{array}{cc} & \begin{array}{cccc}1 & 2 & \dots & k\end{array}\\ \begin{array}{c}1\\2\\\vdots\\m\end{array} & \begin{pmatrix}0 & 1 & \dots & 0\\ 0 & 0 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & 1\end{pmatrix}\end{array}}$$

Step 4: Normalization

$$X' \xrightarrow{\forall i \in 1,\dots,n, \forall j \in 1,\dots,k:\; x_{ij}^{DDMF-k} = x'_{ij} / \Sigma_{j=1}^{k} x'_{ij}} X_{DDMF-k}$$

$$\underset{\substack{X_{DDMF-k}\\(n\times k)}}{\begin{array}{cc} & \begin{array}{cccc}1 & 2 & \dots & k\end{array}\\ \begin{array}{c}1\\2\\\vdots\\n\end{array} & \begin{pmatrix}0 & 0.3 & \dots & 0\\ 0 & 0 & \dots & 0.2\\ \vdots & \vdots & \ddots & \vdots\\ 0.05 & 0 & \dots & 0.2\end{pmatrix}\end{array}}$$

Figure 13: Procedure for generating metafeatures using nonnegative matrix factorization.