Erasmus University Rotterdam
Erasmus School of Economics

Master Thesis Econometrics and Management Science
Programme: Business Analytics and Quantitative Marketing

A Quantitative Comparison of Local Explanation Methods

Name: Daniël de Bondt
Student ID : 416090

Supervisor: prof.dr. Ilker Birbil
Second Assessor: dr. Hakan Akyuz

Internship Viqtor Davis NL
Supervision: dr. Bram Bet

Abstract

Four different machine learning methods for providing local explanations are compared using cosine similarities. The Supersparse Linear Integer Model (SLIM) and Explainable Boosting Machine (EBM) are explored as inherently interpretable methods. A SHAP-explained XGBoost model is considered as a complex black-box model with a model agnostic explainer. Logit is included as a benchmark method. This comparison was done by computing the cosine similarity coefficient between models for individual explanations. A clear distinction was found between linear and nonlinear model formulations. This was further explored using synthetic data where dependent on the data relation either of the two sets of models produced better explanations. From an additional survey conducted among data professionals it was found that most respondents preferred SHAP-explained XGBoost, because of its high predictive performance and general popularity.

December 7, 2020

Contents

1 Introduction

Machine learning prediction models have seen much development over the past few years. More complex models such as deep neural networks and ensemble tree based methods have pushed the performance forward on popular classification data sets. One big drawback of these more advanced so called black-box models, where the inner workings are unobserved, is that they are often difficult to interpret due to their high level of complexity.

Interpretability however, is highly sought after as data modelling and machine learning find their way to many different applications like legal, medical and other business settings, all including their own specific requirements. Firms are in many cases legally required to support their client classifications or customer profiling with sensible undiscriminating relations, for which interpretability is necessary. Medical diagnosis predictions should be clear, robust and cannot be based on data biases, which interpretability helps detect and prevent. Many managers are increasingly interested in using data for performance measures and decision making. Involving complex models to make these data driven decisions would require them to be interpretable. One important aspect of consumer facing models would be trust. To gain a customer his or her trust on recommendations or decisions they need to be able to understand the underlying reasons, thus requiring a level of transparency.

The above examples lead in practice to a constant trade-off between interpretability and predictive performance which is provided by traditional rule-based or linear econometric methods and black box machine learning models, respectively. Two different paths can be explored to combine these two desirable properties: either try to explain the high-performing black-box models through explainer models, or try to improve and develop white box interpretable models, where the inner workings are exposed, aiming to increase their performance. These two pathways constitute the main topic of this thesis, focusing on four models in particular. SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017) provide model agnostic explanations, where the explanation method can be applied irrespective of which underlying complex model is used. Next, the Explainable Boosting Machine (EBM) (Nori et al., 2019) and the Supersparse Linear

Integer Model (SLIM) (Ustun et al., 2013) are both explored as inherently interpretable classification models. Additionally, the Logistic Regression is used as a baseline interpretable model. All of these methods provide a local explanation by means of a list of feature contributions that can be summed to reach a prediction. This similarity in explanations allows for a quantitative comparison not only in terms of performance but also interpretability. This comparison in performance and interpretability between the models will be the main focus of this thesis. Subsequently, an important subtask will be to find the most appropriate way to measure this interpretability quantitatively.

This research is structured as follows. Section 2 will present an overview of the existing literature on the topic of interpretable machine learning. Section 3 will dive into the specific methods considered: SHAP, EBM, SLIM and the Logistic Regression, as well as the used comparison metrics. Section 4 provides a short description of the data sets used and describes the creation of synthetic data. In Section 5, the main results from comparing the different explanations are presented as well as the results from the synthetic data. Section 6 describes the details and results of a survey among data scientists concerning interpretability. To conclude, Section 7 provides a conclusion and discussion of all findings.

2 Literature

One major introductory educational reference would be the book Interpretable Machine Learning (Molnar, 2019). This book serves as a summary of most recent developments in the field of interpretability, providing firstly an introduction into its relevance and terminology. Several aims are stated from Doshi-Velez and Kim (2017) that interpretability helps achieve: fairness, privacy, reliability, causality and trust. This is very similar to the FACT-AI framework employed by Viqtor Davis NL, where fairness is also present, privacy is replaced by confidentiality, reliability by accountability and causality combined with trust by transparency. These are all important criteria in business applications that can be checked and ensured by using model explanations.

Next, a vast range of inherently interpretable models and model agnostic interpretability methods are laid out in Molnar (2019). This first category consists of classical Generalized Linear Models (GLM) (Nelder and Wedderburn, 1972), Generalized Additive Models (GAM) (Hastie and Tibshirani, 1990) and tree- and rule-based methods. Classifying our inherently interpretable models of interest, Logit is an example of a GLM, the EBM formulation is based on GAMs and SLIM is a new, more obscure method not mentioned in the book. Out of the model-agnostic explanation methods SHAP (Lundberg and Lee, 2017) is mentioned as most promising by using Shapley values, grounded in Game Theory, to explain individual predictions in an additive framework. SHAP is thus well suited to include in our analysis. Another model-agnostic method, LIME, is also referenced in the book. LIME finds local surrogate models that aim to approximate the underlying complex model. Additionally, example based explanations are mentioned, but these will not be the focus of this paper.

Apart from this popular book, an exhaustive literature review on the topic has recently been performed by Adadi and Berrada (2018) which provides a more acadamic overview. Adadi and Berrada (2018) use the definition of responsible artificial intelligence from Dignum (2017), based on three main criteria: Accountability, Responsibility and Transparency (ART). They also mention the FAT-ML (fairness, accountability and transparency) community, whose goal is stated as: "to ensure that algorithmic decisions

as well as any data driving those decisions can be explained to end-users and other stakeholders in non-technical terms." According to Adadi and Berrada (2018) explainability is not always needed since sometimes more capable methods are preferred over insightful ones. In certain domains however, where the cost of making a wrong prediction is very high, there is great potential for explainable methods. Autonomous vehicles, healthcare, finance, defence and legal applications are mentioned as such domains. All references and interpretability methods are then categorized along the following axes. The first distinction is made between inherently interpretable models, which provide insightful predictions, and explainer models, that help explain existing complex models. Another split is made between local and global interpretability, where an explanation is global if it represents the logic of the entire model and local if it describes only a single prediction. Lastly, model specific methods, that can be applied to only one kind of models, are separated from model agnostic methods, that can be applied to any model. Inherently interpretable models are de facto model specific. The categorization using this framework of the four methods that will be researched can be found in Table 1. Decision Trees and LIME are also included as reference, but will not be included in the research. The Decision Tree does not follow the similar additive framework and fails to be able to be quantitatively compared for local explanations. LIME does provide additive local explanations, but the implementation of a Logistic Regression surrogate model was not readily available in python and developing one would be out of the scope of the research. While SLIM and the Logistic Regression or the decision tree seem similar they differ widely in their interpretability, especially for higher-dimensional input. Also keep in mind that the distinction between local and global explanations is not completely disjoint. Global

Table 1: The categorization of interpretation techniques along the different axes from Adadi and Berrada (2018).

	type	local/global	specific/agnostic
Logistic Regression	interpretable	global=local	specific
Decision Tree	interpretable	global	specific
SLIM	interpretable	global=local	specific
EBM	interpretable	local	specific
LIME	explainer	local	agnostic
SHAP	explainer	local	agnostic

explanations can just as easily explain single predictions for SLIM and the Logistic Regression. The local explanations of EBM and SHAP can also be aggregated to provide some global insight, although this is not as thorough as the true global methods. Note that SHAP is generally classified as a model-agnostic method, but it does have optimized algorithms, like TreeSHAP for example, that are model specific.

Adadi and Berrada (2018) also touch upon the notion of evaluation of explanation techniques. While they reach the conclusion that the field of quantifying explainability is still in its infancy, some references on the topic are stated. Out of these, Doshi-Velez and Kim (2017) is probably best defined, listing three different types of interpretation evaluation. The first is application-grounded, where the explanation is put into practice and tested by the end users, often domain experts. Secondly, they state human-grounded evaluation, which is similar as the former but with lay people instead of experts, providing a more general notion of explainability. Lastly functionally-grounded evaluation does not involve any humans, which saves time and costs in performing the experiment and getting it approved. However an existing notion of interpretability, for example the level of sparsity, is needed as a proxy for the explanation quality. The research aim of this paper, quantitatively comparing the explanation methods, can be classified as functionally grounded evaluation. The conducted survey provides some additional human-grounded evaluation.

Additionally Doshi-Velez and Kim (2017) argue that the need for interpretability arises from an incompleteness in the problem task. Examples of this incompleteness would be a lack of understanding or a need to guard against discrimination. Furthermore they state several latent aspects of interpretability with the aim of classifying and evaluating combinations of tasks and methods in terms of interpretability.

The field of decision sets has recently seen some development surrounding quantitative interpretability metrics. Lakkaraju et al. (2016) present a new framework for training decision sets optimizing both interpretability and accuracy. For metrics like rule length and number of rules this new method produced similar or more interpretable models compared to reference rule-based methods. Narayanan et al. (2018) have performed a

human-grounded evaluation of decision sets. They presented participants with decision set explanations varying in three ways: explanation length, amount of new information, and amount of repeated terms. From both recipe and clinical domains it was found that higher explanation complexity increased response time and dissatisfaction, but had little effect on the accuracy of understanding. These decision set based metrics however, do not easily translate to the additive framework of our proposed models, with the possible exception that explanation length could be measured as the amount of nonzero feature contributions.

Vaughan and Wallach (2020) call for a more human centered approach when it comes to evaluating intelligibility in machine learning systems. They argue for more extensive use of tools from social studies, more explicitly the field of human-computer interaction. This need is also acknowledged within the Machine Learning community by Kaur et al. (2020), who compared explanation techniques surrounding EBM and SHAP, two of our models of interest, and to what extent they helped the understanding of one specific stakeholder group, data scientists. Several semi-structured interviews and surveys were held that compare GAMS (EBM) and SHAP interpretability tools on three levels: local explanations, feature effects and total feature importance. This was done by means of semi-structured interviews as well as a survey to see if the interpretability tools could help users detect manipulated visualizations. It was found that EBM allows easier recognition of misspecified explanations, although slightly. Most importantly they found that people tend to have overconfidence in both interpretability techniques due to their popularity and public access without having a solid understanding of all their workings. This finding will be further explored in my own survey in Section 6.

3 Methodology

Four different explanation methods will be explored, three of which were developed within the last five years. SHAP provides explanations on top of complex black box models and will be applied alongside an XGBoost prediction model. Both EBM and SLIM are examples of machine learning models that are inherently interpretable and their explanations follow directly from the model structure itself. The baseline Logit model also has this interpretable property with inherent explanations. One limiting factor is that SLIM is developed as an exclusively binary classification method. Therefore, to be able to compare all models, the binary classification task will be the main scope of the research. Examples of this can be a medical diagnosis, or prediction of customer churn or credit card fraud occurrence. Thus for the main problem at hand, the dependent variable y_i, which we aim to predict, can hold two values, for example either 0 or 1. Here, $i \subset \{1, 2, .., N\}$ denotes the i'th observation from a dataset X of size N. This y_i will then be predicted by means of M different input features denoted by $\boldsymbol{x}_i$, a vector of size M. This leads to the following general model formulation:

$$L(P[y_i = 1]) = f(\boldsymbol{x}_i, \theta), \tag{3.1}$$

where $P[y_i = 1]$ denotes the estimated probability of an outcome of 1, f describes the model relation between $\boldsymbol{x}_i$ and y_i including possible unknown coefficients θ, and L represents the link function which maps model output to estimated probabilities.

One of the most used classification techniques is the Logistic Regression, or synonymously Logit model. For the Logit a simple linear relation is assumed in $f = \beta_0 + \boldsymbol{x}_i\beta$, where the constant β_0 combined with a vector of feature coefficients β of length M define the coefficients θ. As a link the logit function is used $L(x) = \text{logit}(x) = \log(\frac{x}{1-x})$. The Logit model is inherently interpretable through the linear coefficients β. A single prediction $P[y_i = 1]$ is uniquely determined through the monotonicity of the logit function by the so called log-odds $\ell_i = \text{logit}(P[y_i = 1])$ given by f. And these log-odds are equal to the sum of the constant β_0 and the set of feature contributions $x_{ij}\beta_j$ with $j \subset \{1, 2, .., P\}$.

Based on this additive property I have defined the contribution vector $\mathbf{c_i}$ of size $(M+1)$ to serve as explanation for observation i. It is formally defined as:

$$\mathbf{c_i}^\intercal = [c_0, c_{i1}, c_{i2}, ..., c_{iM}], \tag{3.2}$$

where for the logistic regression $c_0 = \beta_0$ and $c_{ij} = x_{ij}\beta_j$. This Logit model will be used as a baseline to compare the other methods, whose explanations will be shown to follow this same contribution vector representation as in Equation 3.2. This shared vector representation is what allows the research into a quantitative comparison between the methods. The other methods besides the Logit model will now be described in more detail.

3.1 SHAP

SHapley Additive exPlanations is a technique introduced by Lundberg and Lee (2017) that combines several existing explanation methods into one class of additive feature attribution methods. This additive feature attribution explanation method is a linear combination of M simplified binary input variables z'_{ij} for observation i given by the following equation:

$$g_i(\mathbf{z}') = \phi_0 + \sum_{j=1}^{M} \phi_{ij} z'_{ij}, \tag{3.3}$$

where $z_{ij} = 1$ if feature j is present and $z_{ij} = 0$ if it is not, ϕ_{ij} represents the Shapley value of feature j, and $g_i(\mathbf{z}')$ defines the explanation model approximating the actual model $f_i(x)$ for observation i. This representation follows the contribution vector as in Equation 3.2 by $c_0 = \phi_0$ and $c_{ij} = \phi_{ij}$. Given the requirement of three desirable properties; Local Accuracy, Missingness and Completeness (defined in more detail in Lundberg and Lee (2017) but outside the scope of this thesis), one unique solution is found within this class of feature attribution methods, borrowing coalition techniques from Game Theory:

$$\phi_{ij} = \sum_{S \subseteq X_i \setminus \{j\}} \frac{|S|!(M - |S| - 1)!}{M!} \left[f_i(S \cup \{j\}) - f_i(S) \right]. \tag{3.4}$$

Here, $X_i \setminus \{j\}$ is the set of all input features minus feature j and $f_i(S)$ is the model prediction for observation i, using feature set S. Subsequently, ϕ_0 is given by $f_i(\varnothing)$. SHAP values are defined as a unified measure of feature importance, given by this unique solution. The exact computation of these values involves iterating over all different 2^M coalitions of input features S, which thus takes exponential time and proves to be computationally challenging. Several approximation methods do exist however. Kernel SHAP is proposed as a more efficient model agnostic method as compared to direct calculation through Shapley values (Štrumbelj and Kononenko, 2014). Furthermore the following model specific methods are proposed: Linear SHAP, Low-Order SHAP and Max SHAP for linear models and Deep SHAP for neural networks. Closely following the original publication TreeSHAP is added by Lundberg et al. (2018), introducing an algorithm to compute SHAP values for tree-based models like decision trees, random forests and boosted trees, reducing the direct exponential computation to polynomial time.

This TreeSHAP implementation will also be used in our experiments, since it pairs well with the underlying XGBoost classification model that will be used. XGBoost, short for extreme gradient boosting, is a gradient tree boosting algorithm developed by Chen and Guestrin (2016) that is highly scalable and one of the most popular state-of-the-art machine learning techniques. However, since it consists of an ensemble of decision trees it is not very interpretable and would be classified as a black-box model. SHAP is therefore needed to provide explanations alongside the often highly accurate predictions and TreeSHAP is nicely tailored towards the XGBoost tree structure.

3.2 EBM

The Explainable Boosting Machine is an interpretable model formalized in Nori et al. (2019) as part of the open-source python package InterpretML. It is a generalized additive model including possible pairwise interactions coined GA^{2}M by Lou et al. (2013) with

the following formulation for the prediction $E[y_i]$ of class label y_i for observation i:

$$g(E[y_i]) = \beta_0 + \sum_{j=1}^{M} f_j(x_{ij}) + \sum_{j=1}^{M} \sum_{k \neq j} f_{jk}(x_{ij}, x_{ik}), \tag{3.5}$$

where g is the link function (logit in our case of binary classification), β_0 a constant mean prediction and the second sum covers the possible pairwise interactions. The functions f_j and f_{jk} are freely trained using bagging and gradient boosting, hence the name boosting machine follows. The input x_{ij} can be of any form, but continuous or categorical variables allow the creation of the visually intuitive GAM plots of f_j against the domain of x_{ij}, an example of which can be seen in figure 1. These plots are a major part of the interpretability of EBM, i.e., they make the boosting machine explainable.

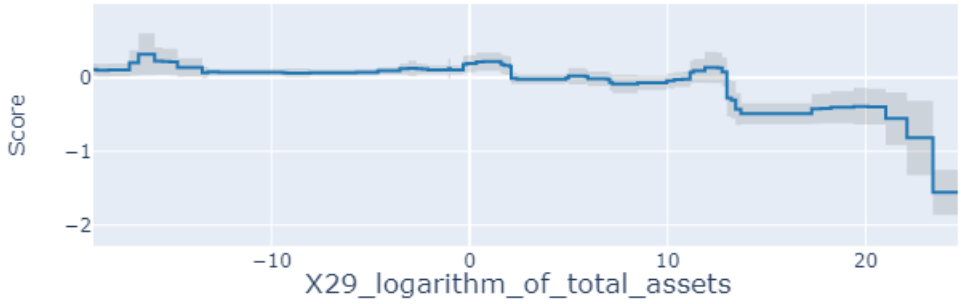


Figure 1: GAM plot of the effect of a feature log(total assets) on the prediction of bankruptcy for a data set of Polish firms. There is quite some variance in the relation, but overall higher assets clearly decrease the bankruptcy risk as would be expected.

While Nori et al. (2019) greatly praise the addition of pairwise interactions because it allows for higher levels of accuracy while maintaining interpretability, this benefit is not applicable to our research. Interaction terms would prevent the direct comparison of explanations with other models where they are not explicitly included and are thus left out for our experiments. This leaves the model formulation like:

$$g(E[y_i]) = \beta_0 + \sum_{j=1}^{M} f_j(x_{ij}) +, \tag{3.6}$$

with the individual contribution vector for observation i given by $c_0 = \beta_0$ and $c_{ij} = f_j(x_{ij})$.

3.3 SLIM

The Supersparse Linear Integer Model is introduced by Ustun et al. (2013) as an interpretable classification method. It creates scoring systems, an example of which can be seen in Figure 2, where predictions are made by means of simple addition and subtraction of several characteristics. These systems provide some unique requirements to keep them

PREDICT MUSHROOM IS POISONOUS IF SCORE > 3

1.	$spore\_print\_color = green$	4 points		.....
2.	$stalk\_surface\_above\_ring = grooves$	2 points	+	.....
3.	$population = clustered$	2 points	+	.....
4.	$gill\_size = broad$	-2 points	+	.....
5.	$odor \in \{none, almond, anise\}$	-4 points	+	.....
	ADD POINTS FROM ROWS 1–5	**SCORE**	=	.....

Figure 2: The scoring system for mushroom edibility produced by SLIM as displayed in Ustun and Rudin (2015)

simple and interpretable like integer coefficients and sparsity. This sparsity is required for applications where a large set of possible input variables is available while only some may prove significant for predicting the outcome. Classical linear methods perform poorly with regards to this sparsity and often create bias in rounding to integer coefficients. SLIM solves these issues by formulating the classification task as a discrete optimization problem. The following general strategy is laid out in Ustun et al. (2013):

$$\begin{aligned} \max_{f} \quad & \text{Accuracy}(f) + C \times \text{InterpretabilityScore}(f), \\ \text{s.t.} \quad & \text{InterpretabilityConstraints}(f) > 0. \end{aligned} \tag{3.7}$$

More formally, SLIM is defined as a special case of (3.7):

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[y_i \boldsymbol{x}_i^T \boldsymbol{\lambda} \leq 0] + C_0 ||\boldsymbol{\lambda}||_0 + C_1 ||\boldsymbol{\lambda}||_1, \\ \text{s.t.} \quad & \boldsymbol{\lambda} \in \mathcal{L}, \end{aligned} \tag{3.8}$$

where $\boldsymbol{x}_i$ denotes a vector of length $M + 1$, where the first value is a 1, followed by the feature values for observation i, with corresponding label $y_i \in \{-1, 1\}$. Here, $\boldsymbol{\lambda}$ represents a vector of coefficients of length $M+1$ corresponding with the vector of features $\boldsymbol{x}_i$, where

the first entry λ_0 represents the constant. The $\mathbb{1}[...]$ indicator function computes the 0-1 loss misclassification error. For correct classification y_i and $\boldsymbol{x}_i^T\boldsymbol{\lambda}$ should be both either negative or positive such that their product is positive. The regularization terms C_0 and C_1 are included in combination with the 0- and 1-norms of $\boldsymbol{\lambda}$ to control sparsity in the number of used features and limit the size of the feature coefficients respectively. The $\mathcal{L}$ solution space for $\boldsymbol{\lambda}$ is the set of all possible integers $\mathbb{Z}$. SLIM follows the contribution framework in Equation 3.2 as $c_0 = \lambda_0$ and $c_{ij} = \lambda_j x_{ij}$. This kind of discrete optimization problem can be solved using Mixed-Integer Programming (MIP). While these MIP problems are generally considered as NP-hard, a solution for medium sized data sets can generally be reached within reasonable time. This specific formulation allows for very versatile addition of constraints and optimized training. To summarize, SLIM is similar to Logit with its linear coefficient formulation, but these coefficients are restricted to integer values. Furthermore, the model is trained using a MIP solver whereas Logit is applied using other standard optimization techniques.

3.4 Comparison Metrics

As presented in the previous subsections, the explanations from any of the four methods used are thus all provided in an additive manner. For any given dataset of size $N \times M$, with N the number of observations and M the number of features, this gives a resulting three-dimensional matrix of explanations of size $N \times (M+1) \times P$, where a constant is added to the feature values and P is the number of models considered, four in our case. Each observation i is represented here by four different explanation vectors $\mathbf{c}_{1,i}, .., \mathbf{c}_{4,i}$ as described by Equation 3.2, one for each model. An example of this using models trained on the haberman data set can be viewed in Table 2. Here, the Logit misclassifies the observation as a 1, since the sum is positive, and all other methods correctly predict a 0. Note that wherever the term SHAP is used alongside the other models, an underlying XGBoost model that is explained using SHAP is implied. Several techniques are employed to look for relations between these explanations.

Table 2: An example of the different contribution vectors of each model for a specific prediction from the haberman data set

	Constant	Age	NumberOfNodes	YrsSinceFirstOperations	Sum
SLIM	80.00	-66.00	-39.00	3.00	-22.00
Logit	2.17	-0.70	-1.15	0.00	0.33
EBM	1.35	-0.17	-1.40	0.22	-0.00
SHAP	1.34	-0.56	-1.85	0.14	-0.93

The evaluation of predictive performance can be quite straightforward. Model predictions are either correct or incorrect in our binary case. Scoring the explanation quality however, is a much more subjective task. An attempt to define a quantitative similarity measure between the local explanation of different models could provide some insights. One challenge that immediately surfaces is that the different models do not provide explanations on a similar scale. Logit, EBM and SHAP produce log-odds values, but this does not hold for SLIM. This is especially present when feature values take on relatively high values, for example the above displayed Age variable that ranges from 30 to 83. Combine this with the SLIM integer coefficient constraint and it becomes obvious that both the constant threshold and feature contributions (integer coefficient times feature value) get very high to be able to distinguish between labels. Even though the regularizing $\mathcal{L}_1$ norm of SLIM should limit coefficient size, these coefficients multiplied with feature values are still much higher than the feature contributions of other models.

Another issue arises with EBM. Even though it produces log-odds values, which should be similar in terms of scale to Logit and SHAP, it is the only method with no way of explicitly controlling sparsity in the solution. More precisely, the EBM algorithm assigns a non-zero weight to every feature, regardless of the number of features present in the fitting task at hand, while all other methods leave out features that carry little information. This means that once tasks get increasingly more input features EBM will still be using all of these, while all of the other methods leave non important features out of the model entirely. This disparity is very pronounced for the mushroom data set. With 113 features, it has the most out of all explored data sets and it turns out that all methods except EBM dismiss about 80% of these features. With other methods setting

feature contributions to zero in many cases, EBM would become the odd one out if simple distance similarity measures would be used.

One last issue with comparing EBM is also the risk of overfitting. For data sets where prediction performance reaches perfection, EBM tends to set feature contributions very high. It pushes the model very high or low into the log-odds bringing the actual prediction ever that bit closer to 0 or 1, which in essence does not make a difference in predictive performance. However, these huge feature contributions are also very hard to directly compare to other methods. To conclude, if we are to develop some sort of similarity measure, it should be scale invariant to allow a fair comparison between explanations. Simply taking the absolute distance between two vectors $\mathbf{c}_{1,i}$ and $\mathbf{c}_{2,i}$ is not an option.

This need for scale invariance leads right into the idea of computing the Pearson correlation coefficient as a location and scale invariant comparison between explanation vectors $\mathbf{c}_{a,i}$ and $\mathbf{c}_{b,i}$. It would be given by:

$$\rho_{ab,i} = \frac{\sum_{j=1}^{M}(c_{a,ij} - \bar{c}_{a,i})(c_{b,ij} - \bar{c}_{b,i})}{\sqrt{\sum_{j=1}^{M}(c_{a,ij} - \bar{c}_{a,i})^2}\sqrt{\sum_{j=1}^{M}(c_{b,ij} - \bar{c}_{b,i})^2}}. \tag{3.9}$$

where, $\rho_{ab,i}$ represents the correlation between models a and b for observation i and $\bar{c}_{a,i}$ denotes the mean of $\mathbf{c}_{a,i}$. This correlation measure can subsequently be averaged across all N observations to provide a measure as to how similar the explanations of two different models are for a specific data set, along with a sampled standard deviation.

One remark regarding correlation coefficients could be its use of the mean of the contribution vector. This vector consists of contributions from different feature values, which are not expected to be drawn from some distribution. As such, an average over these different variables makes intuitively little sense. One way to get this mean out of the evaluation would be to use cosine similarities as an alternative. The cosine similarity is defined as the cosine of the angle between two vectors and ranges from $[-1, ..., 1]$, similar

to the correlation coefficient. It is formulated as:

$$\cos(\mathbf{c}_{a,i}, \mathbf{c}_{b,i}) = \frac{\sum_{j=1}^{M} c_{a,ij} c_{b,ij}}{\sqrt{\sum_{j=1}^{M} c_{a,ij}^2} \sqrt{\sum_{j=1}^{M} c_{b,ij}^2}}, \tag{3.10}$$

for explanation vectors $\mathbf{c}_{a,i}$ and $\mathbf{c}_{b,i}$. The definition is very similar to the Pearson correlation but the mean is not present in this equation.

For fitting the models, 5-fold cross validation is used. Each data set is split into five distinct subsets and each unique set is once taken as test set, where the other four make up the training set. Results are then averaged over these five different folds to yield a representative measure of out of sample performance. For SLIM a max timing is set to 1 hour to allow for a workable solution within reasonable time. The other methods are not time-constrained for the chosen data sets and are fully trained to convergence. AUC, the area under the ROC curve, is used as a performance measure since this allows for an evaluation of both recall, or true positive rate, and precision, false positive rate, for different cut-off values. Overall the AUC performance measure paints a broader picture than for example a hitrate accuracy statistic alone.

All the Jupyter Notebooks containing the code and scripts used for performing these evaluations can be found on my github `https://github.com/DanieldeBondt/Msc-Thesis-Explanations`.

4 Data

Many academic contributions in data science make use of publicly available data sets, often registered at the UCI Machine Learning Repository, to show their new methodology matches up to existing alternatives. This allows for easy replication, which is something that will also be very useful in comparing the different methods in my research. The binary classification data sets used by some of the referenced papers are given by:

- Breastcancer, haberman, internetad, mammo, spambase and tictactoe, all from UCI (Ustun et al., 2013).
- Spambase, insurance, magic, letter and adult, all from UCI (Lou et al., 2013).
- Breastcancer from sklearn, adult income from UCI, heart disease from Kaggle/UCI, credit card fraud from Kaggle and telecom from Kaggle (Nori et al., 2019).

Out of this list a collection of 8 representative data sets was selected with different sizes, shapes and feature types, the details of which are shown in Table 3. Four of these, heart, haberman, breastcancer, and mammo, concern a medical diagnosis task for which interpretability is of vital importance. Adult and mushroom are included to test the methods on data sets of increasing size and feature count. Bankruptcy and spambase provide interesting applications in finance and natural language processing respectively.

Table 3: Summary statistics of the different data sets used, where the balance denotes the percentage of positively labeled target observations.

name	size (N)	features (M)	balance	feature type
bankruptcy	250	6	0.57	discrete
heart	303	32	0.46	binary & continuous
haberman	306	3	0.74	continuous
breastcancer	683	9	0.35	continuous
mammo	961	14	0.46	binary
spambase	4601	57	0.39	continuous
mushroom	8124	113	0.48	binary
adult	32561	36	0.24	binary

4.1 Synthetic Data Creation

Comparing different explanation models to each other using the methods mentioned in Section 3.4 can provide some measure of explanation quality, but yields solely relative results. One way to overcome this would be to find a true or correct data relation that can be used to match the explanations of different models with. The model explanations can then be scored quantitatively to what extent they approximate this true relation. While the correct label of observations might be known for test and training data, this true or correct relation is however never known in practice, even for labelled data. Uncovering and explaining this relationship within the data is part of the aim of data modeling itself. Therefore, to score the methods in absolute terms it is necessary to construct the data by hand. This way, the original relation is known and the explanations from different methods can fairly be tested.

The general setting from the UCI Adult Income data set is used to keep the data understandable. To provide additional intelligibility, the synthetic data set is constrained to only four discrete explanatory variables X; Age, Education, (work) Experience and Hours. The number of generated data points is set to 5000. Age and Experience are uniformly generated integers between (18, 70) and (0, 20) respectively. Education and Hours are normal random variables with means 10 and 40, and deviations 2.5 and 2 respectively. These normal random variables are rounded to the nearest integer.

Next, the relation between these four variables and the label y (Income $\geq$ 50K) is set. Since a true contribution vector is needed, this relation can be defined as the relation between x_{ij} and its specific contribution c_{ij} for variable j and observation i as introduced in Equation 3.2. The simplest relation would be linear, similar to the the Logistic Regression model, with $c_{ij} = \beta_j x_{ij}$, where β_j is the coefficient determining the effect feature j has on the outcome. This is a simple formulation, but would yield unsurprising results. All models would produce excellent predictions and Logit and SLIM would give the best explanations because of their perfectly suited linear formulation. The strength of EBM and SHAP-explained XGBoost is their model complexity which allows them to capture nonlinearities in the data. Therefore a second relation $c_{ij} = g_j(x_{ij})$ is

considered, where g_j is a nonlinear function determining the effect of feature j. This nonlinear function could take any form, but is deliberately kept simple. For constructing the function, the domain of feature j is cut into five equal parts that all have a different constant effect. This nonlinear function for variable Age is displayed below in Figure 3 by means of the blue lines. The linear coefficient introduced above is also included as a straight, sloping red line in this figure. The variables Age and Education are given

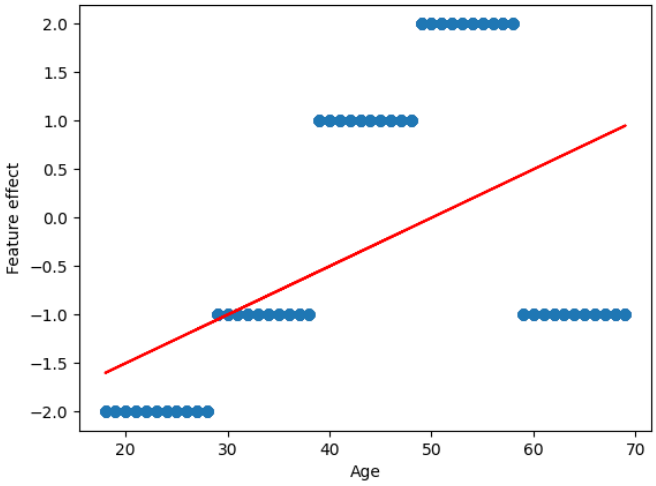


Figure 3: The constructed linear (red) and nonlinear (blue) feature effects for the constructed variable Age.

nonlinear effects of $[-4, -2, 2, 4, -2]$ and $[-4, -3, -1, 2, 6]$ respectively. The variables Experience and Hours are given linear effects with both coefficients equaling 0.4. These feature effects are added for every observation along with a random standard normal error. Subsequently the observations are labeled by means of a threshold c. The amount of nonlinearity in the data can be determined by rescaling the feature effects with a factor λ between 0 and 1. This defines the following data generation process:

$$y_i = \begin{cases} 1 & \text{if log-odds}_i \geq c, \\ 0 & \text{if log-odds}_i < c, \end{cases} \tag{4.1}$$

$$\text{log-odds}_i = \lambda[g_{\text{Age}}(x_{i\text{Age}}) + g_{\text{Edu}}(x_{i\text{Edu}})] + (1-\lambda)[\beta_{\text{Exp}} x_{i\text{Exp}} + \beta_{\text{Hrs}} x_{i\text{Hrs}}] + \epsilon_i, \tag{4.2}$$

where ϵ_i represents the normal error. For the threshold the median of all log-odds values

is chosen to produce balanced y labels. To summarize, a λ of zero yields a completely linear model and a λ of one a fully nonlinear one.

Alternatively, Equation 4.2 can be modified such that all four features contribute a combination of both linear and nonlinear effects. For this the linear coefficients for Age and Education are also set to 0.4 and the nonlinear effects for Experience and Hours are given by [-4, -1, 2, 4, 5] and [2, -4, -2, 2, 4] respectively. In a similar way to Equation 4.2 a λ coefficient is used to determine the level of nonlinearity, but now all features contribute in both ways described as:

$$\text{log-odds}_i = \lambda \left[\sum_{j \in D} g_j(x_{ij}) \right] + (1 - \lambda) \left[\sum_j \beta_j x_{ij} \right] + \epsilon_i, \tag{4.3}$$

where D is defined as the set of all four features {Age, Education, Experience, Hours}. The noise ϵ_i is kept at a standard normal error.

Within this synthetic data framework, a last comparison was also made to check to what extent the methods are robust to increasing levels of noise. This was done using the formulation in Equation 4.3 with a nonlinearity value kept constant at 0, while instead varying the standard deviation of ϵ_i.

5 Results

5.1 Visualizations

Since evaluating explanations mainly consists of subjective reasoning from the perspective of either an end user or a domain expert, an integral part of this evaluation is determined by the way the explanations are displayed. Both the introduction of SHAP and EBM (InterpretML) came with their own visualization tools. SHAP provides a force plot to explain a single observation, dependence plots to give insight into a single feature's effect on the output and summary plots to describe all feature effects simultaneously. The InterpretML package has similar graphs for local explanations and partial dependence plots. For comparing different methods however, it would be helpful to be able to compare them in one overview. A single predicted observation with different explanations can be summarized in a table, as for example Table 2. A more visually intuitive picture is achieved however, by modifying the SHAP decision plot. While it originally displays SHAP values for a single prediction, the code was altered to include multiple explanations with different start and end values and distinct colours. An example using the breastcancer data set can be viewed in Figure 4. SLIM is not included since it does not produce log-odds values to fit the Logit link used in the figure. All of the other three

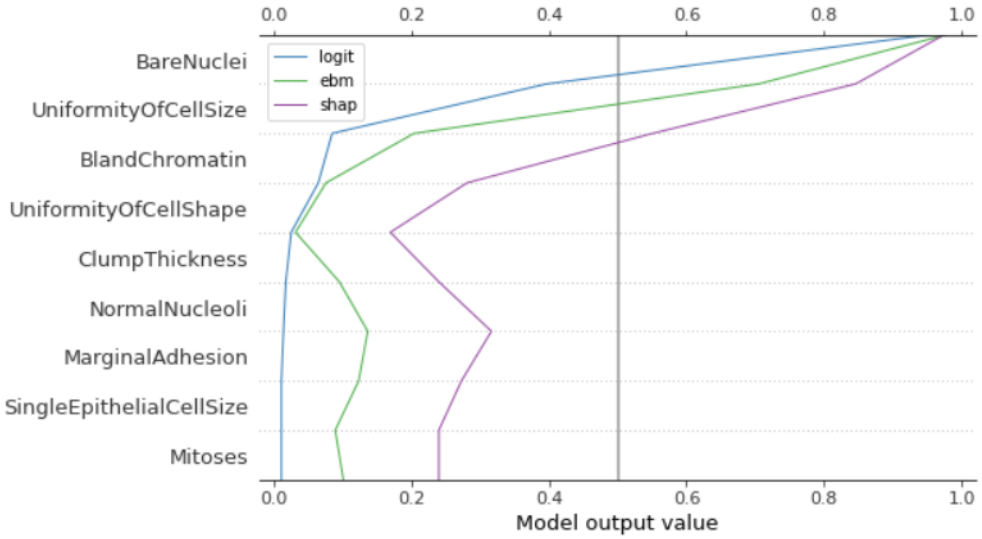


Figure 4: Decision plot showing prediction of the three models that use a Logit link function.

models start at the bottom left of the graph with a constant term favoring a zero prediction. It is clear however that the upper 4 features (BareNuclei, UniformityOfCellSize, BlandChromatin and UniformityOfCellShape) have strong predictive power towards a one prediction for all models, shifting the decision line to the right and resulting in a unanimous one prediction.

5.2 Predictive Performance

Historically, the predictive performance of machine learning models has been the main comparison benchmark. If a new technique outperformed other existing methods or at least matched them in terms of accuracy or AUC it would earn its recognition within the field. To draw the full picture between our different models it thus makes sense to first take a look at their performances, measured here as the AUC. Cross-validated average out-of-sample performance is displayed in Table 4. The in-sample results on the training sets can be found in Appendix A. From these training set results the SHAP-explained XGBoost model is clearly superior for all data sets, but this does not directly translate into the best test set performance, as can be seen in Table 4. This could be a sign of overfitting, which might be present since no explicit precautions were made to prevent it. The spambase and adult data sets are the only ones where SHAP still

Table 4: The AUC performance statistic for the test sets of different models and data sets with best performing models in bold. The standard deviation over the 5 different folds is also displayed.

model	EBM		Logit		SHAP		SLIM	
	AUC	std. dev.	AUC	std. dev.	AUC	std. dev.	AUC	std. dev.
dataset								
bankruptcy	**1.000**	0.001	**1.000**	0.000	**1.000**	0.000	0.993	0.007
heart	**0.908**	0.031	0.899	0.029	0.901	0.038	0.887	0.023
haberman	0.630	0.080	0.674	0.110	0.632	0.131	**0.731**	0.107
breastcancer	0.993	0.007	**0.995**	0.005	0.992	0.008	0.986	0.006
mammo	0.850	0.026	**0.855**	0.024	0.845	0.017	0.799	0.040
spambase	0.692	0.259	0.958	0.043	**0.974**	0.031	0.898	0.063
mushroom	0.971	0.066	**1.000**	0.000	0.998	0.005	0.930	0.118
adult	0.890	0.005	0.891	0.004	**0.893**	0.005	0.847	0.012

performs best on the test data. Surprisingly, the Logit seems to reach the best out of sample performance on the highest number of data sets. EBM reaches optimal or close to optimal performance on most sets with the exception of the spambase data set, probably caused by one or two folds significantly decreasing the average, which would also explain the high standard deviation. SLIM often fails to reach top level performance with the exception of the haberman data set where it reaches a competitive AUC of 0.731. For this data set however, all methods experienced a high variation among the different validation folds, so SLIM can not be said to be decisively superior.

5.3 Explanations

For all eight data sets, the average pearson correlations over the 5 folds between the explanations of different models are displayed in the tables in Appendix C. The test set correlation results are summarized in Table 5 with corresponding standard deviations. What can immediately be noted is that different data sets yield different results with respect to the models. It seems like it really depends on the underlying relation in the data whether some methods provide similar explanations. However when looking at the average over all data sets, a clear pairing can be observed. The two highest correlation values are the ones between SLIM and Logit (0.697), and between EBM and SHAP (0.622).

Table 5: The five-fold average test set correlations between explanations of different models and their corresponding standard deviations averaged over all eight data sets.

model	SLIM	Logit		EBM		SHAP	
		corr.	st. dev.	corr.	st. dev.	corr.	st. dev.
SLIM		0.697	(0.250)	0.383	(0.300)	0.505	(0.294)
Logit				0.450	(0.273)	0.511	(0.198)
EBM						0.622	(0.289)
SHAP							

Two important issues should be recognized with regards to the above results. Firstly this correlation analysis looks at all observations of a given test set, even ones where different models might disagree on the predicted value. One could question whether

it is sensible to compare explanations for two completely different predictions, one of which must be incorrect. A choice is made to remove these observations from the analyis, leaving only the explanations where all models agree on.

Furthermore, the correlation measure is undefined if one or more of the variables is a constant, which is the case for an all zero vector. This all zero vector can sometimes appear in explanations of SLIM or Logit when some features are not used and all others have zero value. These observations were also removed which resulted in the new correlation results as presented in Table 6, again extracted from the test set. The test set left after removal of observations was on average 43% smaller.

Table 6: The 5-fold average correlations between explanations of different models and their corresponding standard deviations for cleaned observations averaged over all eight data sets.

model	SLIM	Logit		EBM		SHAP	
		corr.	st. dev.	corr.	st. dev.	corr.	st. dev.
SLIM		0.689	(0.268)	0.365	(0.299)	0.465	(0.313)
Logit				0.451	(0.292)	0.546	(0.289)
EBM						0.618	(0.287)
SHAP							

These results do not defer very significantly from the raw results with many values changing by no more than a percentage point, except for the correlation between SHAP and SLIM or Logit. Also the deviation estimates reach similar results with a slight increase in general, which makes sense since the data set size is reduced. Of course this would be expected since for a large part the same explanations are being compared.

As introduced in Section 4, a more intuitive measure could be the cosine similarity. Results for this similarity measure for the test sets are displayed in Table 7. Note that training set results for both correlation and cosine similarity are summarized in Appendix B. The same general result can be derived from these cosine results. Namely, the similarity is highest for the two pairings of SLIM-Logit and EBM-SHAP. The difference between these similarities and the others is slightly more pronounced compared to the correlation results, with the two pairings achieving higher results and the other combinations all getting lower similarities.

Table 7: Table displaying the cosine similarity between explanations of different models on the test set averaged across several folds and different data sets .

model	SLIM	Logit		EBM		SHAP	
		cos.	st. dev.	cos.	st. dev.	cos.	st. dev.
SLIM		0.721	(0.232)	0.332	(0.317)	0.431	(0.373)
Logit				0.399	(0.372)	0.494	(0.396)
EBM						0.664	(0.316)
SHAP							

One aspect that may have had a great impact on the results could be the earlier mentioned sparsity in all methods except EBM. Some features may be removed from the model for these methods. An analysis of this sparsity for different data sets is given in Table 8. Note that EBM is not included since it will always use all features. It seems

Table 8: Number of features set to zero out of the original number of features p for different models trained on the first fold of different data sets.

data set	p	Logit		SHAP		SLIM	
bankruptcy	6	3	(50%)	1	(17%)	5	(83%)
heart	32	17	(53%)	7	(22%)	28	(88%)
haberman	3	0		0		0	
breastcancer	9	0		1	(11%)	3	(33%)
mammo	14	4	(29%)	2	(14%)	7	(50%)
spambase	57	5	(9%)	7	(12%)	39	(68%)
mushroom	113	93	(82%)	86	(76%)	106	(94%)
adult	36	4	(11%)	2	(6%)	30	(83%)

the only data set where all methods make use of all features is the haberman data set which had only three features to begin with. Overall Logit and SHAP are similarly selective whereas SLIM is very restrictive often removing over half or almost all of the features. This difference in sparsity can be further visualized by Figure 5 where feature importance on the bankruptcy data set is displayed for the different models. All models recognize CompetitiveRisk as the most important feature, where SLIM even solely uses it. SHAP and Logit seem to only take two other features into account. SHAP has another two nonzero features on paper, but their effects are very marginal and not visible in the figure. EBM on the other hand uses all five features and its contributions are on a way bigger scale. Luckily, this difference in scale is no issue when a comparison

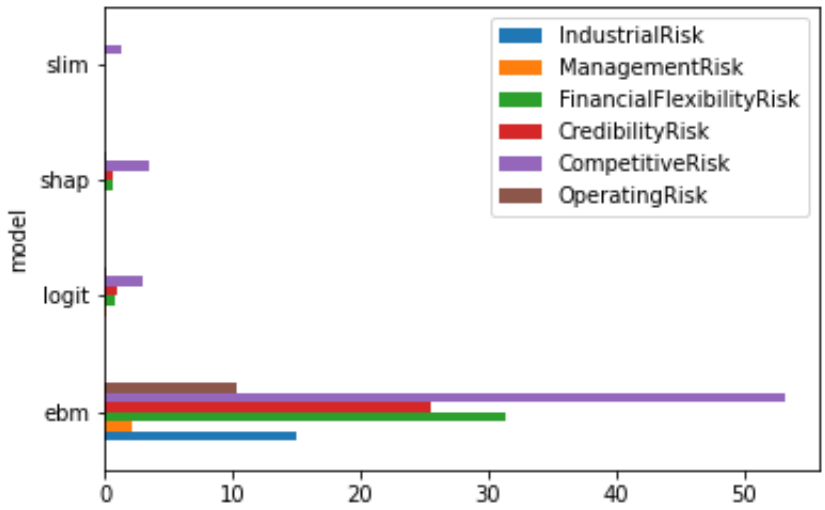


Figure 5: Absolute feature importances for different models trained on the bankrupcty data set

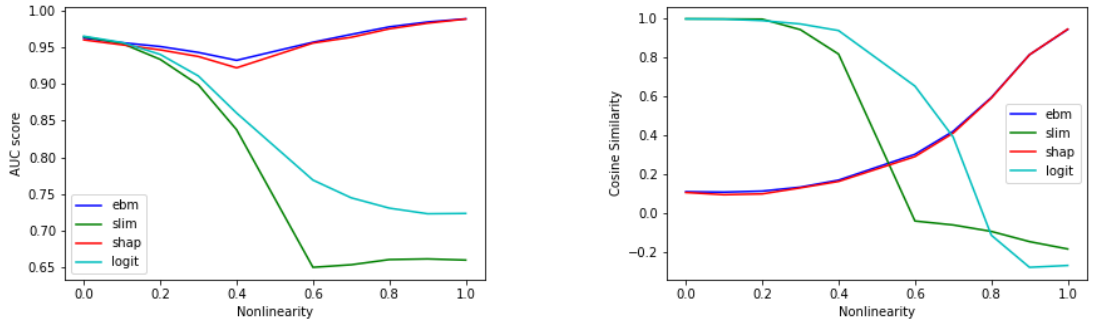
is made using scale-invariant correlation metrics. The sparsity difference could however give unreasonable results with many zero values in the correlation or cosine similarity computation.

This realisation also shines another light on the correlation results with regards to EBM. The two data sets with the most features and thus the most removed features by the other models are mushroom and spambase. It appears that for these two data sets the correlation of EBM with any other model is lower than between any two of the other three models as can be seen in Appendix C. This suggests the correlation or cosine similarity might not be a proper comparison method for bigger data sets, precisely because of this difference in sparsity. One way to overcome this would be to make EBM explicitly control the sparsity of the solution, but the InterpretML implementation did not support an option to handle this.

5.4 Synthetic Data

As described in Section 4.1, synthetic data is created to help define an absolute measure for scoring the explanations of different methods. Five thousand data points were created

for each different nonlinearity balance value from 0 to 1 with increments of 0.1, after which all models were trained. From these trained models, the test set AUC and the explanations similarity to the true relation is calculated as displayed in Figure 6. For this synthetic data experiment no cross validation was applied since the created data sets are already homogeneous and large in size.



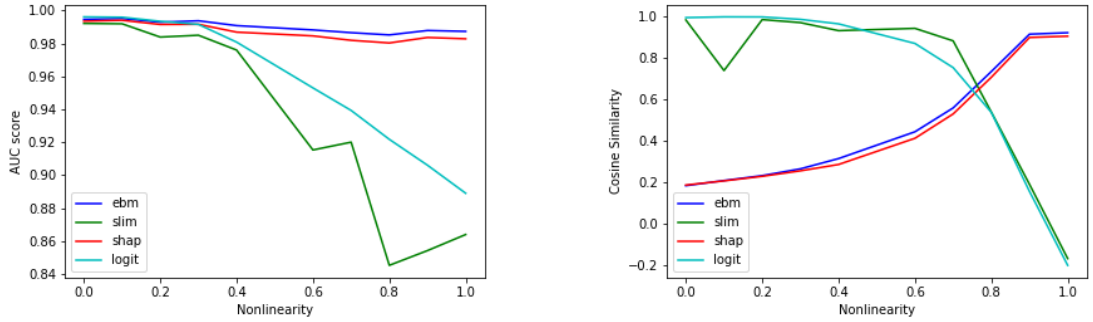
(a) AUC score (b) Cosine similarity of explanations

Figure 6: For different levels of nonlinearity in the synthetic data, the auc and cosine similarity is plotted for all four models.

What can be seen in the left figure of AUC scores is that for linear data ,i.e., low levels of nonlinearity, all models perform similarly. Zooming in on the case with 0 nonlinearity, SLIM and Logit even outperform the other two methods with a few percentage points, although hardly visible. Once some degree of nonlinearity is introduced however, the two linear models quickly drop as expected. SLIM performs similarly to Logit up to a balance of about 0.4, but drops even faster from there on.

A view at the cosine similarity shows a very interesting trade-off. For low nonlinearity models the linear models perfectly approximate the correct explanation, while SHAP and EBM are a long way off even though their AUC performance is similar. This reverses once the nonlinearity level rises above about 0.6. The decrease in AUC of the linear methods is clearly accompanied with a strong drop in explanation accuracy as well with again SLIM falling off earlier.

Additionally, instead of having distinct linear and nonlinear features these two data relation properties were also combined for every feature. This resulted in the following two pictures, shown in Figure 7. While the exact results are different from what

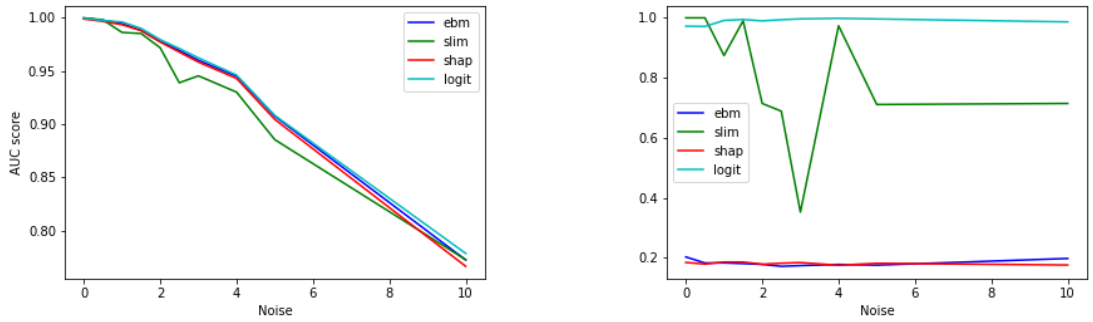


(a) AUC score (b) Cosine similarity of explanations

Figure 7: For different levels of nonlinearity in the synthetic data, the auc and cosine similarity is plotted for all four models.

was achieved with distinct features above, the general relation is very similar. The only exception would be that SLIM seems less consistent with some drops and peaks in both figures. It does however seem to be able to keep up with Logit in terms of cosine similarity, while for the AUC score it does still drop earlier.

Another experiment was performed to see how the different methods respond to different levels of noise. For this, the linearity was kept at 0 and the standard deviation of the random noise was set to different values between 0 and 10, instead of the constant 1 in the earlier experiment. Results can be found in Figure 8 below. It seems that an increase in noise leads to a decrease in AUC uniformly across all models. Furthermore, the statistics behind the Logistic Regression seem to hold up best in producing quality explanations in the special case of a large normal distributed error. Most surprisingly,



(a) AUC score (b) Cosine similarity of explanations

Figure 8: For different levels of noise in the synthetic data, the auc and cosine similarity is plotted for all four models, while keeping nonlinearity constant at 0.

SLIM yields very flaky results for increasing noise up to a certain level, but this settles after the standard deviation is increased to about 5 in this case.

Overall, it does seem that the simplicity of the created data set does not allow a clear distinction to be made between SHAP and EBM. Across all figures the blue and red line follow each other really closely. Maybe a more difficult data creation process, either through more features or a more complex data relation could help differentiate between the two.

6 Survey

The interpretation of predictions made by Machine Learning models will always involve a large human aspect. Quantitatively comparing explanations can lead to useful insights, but ultimately the level of understanding is defined from the perspective of the user. For this exact reason, an additional survey is conducted to provide some human-based evaluation to the explored explanation methods

6.1 Survey Structure

A similar survey as used in Kaur et al. (2020) was constructed with the aim to include Logit and SLIM into the evaluation. While Kaur et al. also look at global feature importance and dependency plots, the focus was held to only examine local explanations since this best aligns with the other parts of my research. The respondent group consisted of a limited number of data scientists within Viqtor Davis. Because of this limitation, all models were presented to everyone instead of showing each model to only a distinct subset of respondents. Similar to Kaur, The UCI Adult Income data set was used to train the models and ask interpretation questions about. First, a short introduction was provided into interpretable machine learning, binary classification and the Adult Income data set. Then respondents were asked to state their level of experience with machine learning and the specific data set. Next all four models were visited in the order of Logit, SLIM, EBM and SHAP-explained XGBoost. For each model, first a small introduction was provided into the model after which an explanation was displayed and the respondents were asked the following: (1) to state their familiarity with the model on a scale of 1-3; (2) a multiple choice true/false question about the explanation to test the respondents understanding; (3) to state their level of understanding on a scale of 1-7; (4) to state their confidence in explaining the method to stakeholders (1-7); and lastly (5) to state their confidence the model could be of value in practice (1-7). After all models were presented, the predictive performance of different models was displayed and the respondent was asked to choose one preferred explanation method and include their possible reasoning.

6.2 Survey Results

The survey got a total of 14 respondents. All had at least some experience with Machine Learning and 11 even stated their experience as considerable. Only 4 had seen the Adult Income data set before. Below in Table 9 the average answers to the model specific questions are displayed with the highest scoring model per question in bold. Additionally, the standard deviation is displayed to give a sense of the spread among responses.

Table 9: Average scores of survey respondents on model specific questions as well as corresponding standard deviations (SD). Highest scores in bold.

	Familiarity		Test	Understanding		Stakeholders		Practical Value	
	(1-3)	SD	% Correct	(1-7)	SD	(1-7)	SD	(1-7)	SD
Logit	**3.00**	0.00	78	5.21	1.25	4.57	1.50	5.36	0.93
SLIM	1.36	0.63	64	**5.79**	1.53	**5.71**	1.49	4.86	1.75
EBM	1.79	0.70	**86**	4.64	1.55	4.43	1.79	5.07	1.21
SHAP	2.36	0.63	71	5.43	1.09	5.00	1.24	**5.86**	0.77

It can be noted that Logit is by far the best known, which should come as no surprise as it is widely used in practice. Furthermore, it seems SHAP has gotten the most recent attention from the three newer models and SLIM is the most obscure. Examining the stated confidence levels, SLIM is the clear odd one out. While respondents state the method as the most easy to understand or explain to shareholders, their confidence of the practical value of the model is lowest out of all four. Overall, confidence spread was lowest for the most familiar models of Logit and SHAP, and the less known methods of SLIM and EBM got more variance in stated confidence levels. The percentage of correct responses to the True/False question is also displayed in the table, but from respondent feedback it was concluded that these questions had some ambiguities so it is difficult to draw clear conclusions from these percentage numbers.

Below in Table 10, the correlation is displayed between the stated familiarity and the different stated confidence measures. This correlation could be read as to what extent a model is understandable without prior knowledge, where a low number means it is understandable and a high number means that the model requires prior knowledge. Since all respondents stated a familiarity of 3 for the Logit model, these correlation were

Table 10: Correlation between stated familiarity and different stated confidence measures.

	Understanding	Stakeholders	Practical Value
Logit	-	-	-
SLIM	0.24	0.28	0.05
EBM	0.56	0.63	0.20
SHAP	0.43	0.00	0.11

undefined for the Logit. The EBM model has the strongest correlation between the familiarity and the stated confidence measures. This suggests it may be the hardest to understand at face value. SHAP has a strong correlation only for understanding, but less so for the other confidence measures. This leads to believe that it is also hard to understand at face value, but is trusted to some degree nonetheless. This could be because it is the most popular new explanation method and this general popularity could provide some legitimacy, which was also found by Kaur et al. (2020).

For the final question, where respondents were asked to state their preferred model, the majority (9) picked SHAP, followed by Logit (4) and SLIM (1). These results seem to follow the average scores for practical value as displayed in Table 9. The reasoning given behind SHAP was largely based around the strong predictive performance and the powerful combo with the easy to use python explainability package. One respondent summarizes this neatly as: *"The combination of a powerful classification ML algo with the possibility to 'explain' individual predictions by assigning contributions seems very powerful. The other 3 methods may lack predictive power when applied on complex data sets."* The Logit model was often stated as similar to EBM in terms of performance, but was superior in terms of explainability. For general business settings some preferred the Logit to SHAP, because SHAP, while being an explainer, was still perceived as hard to understand. The main and only given argument in favor of SLIM was the following: *"Converting the summation of coefficients (beta) via a sigmoid curve into probabilities is beyond most people. .. "*, i.e., the explanations of other methods are too mathematically challenging. The EBM was considered by some respondents, but was outperformed by SHAP and seen as less intelligible than Logit.

To conclude, this survey showed that the Logit model is by far the best known

and most used method. This familiarity makes for a strong case, but when predictive performance is considered, a SHAP explained XGBoost model is often favored. SLIM was perceived as very understandable and explainable, but lacked predictive performance. The EBM was not preferred in general.

7 Conclusion and Discussion

Four models and corresponding local explanations have been compared in terms of predictive performance and interpretability. Explanations of SLIM, Logit, EBM and SHAP have been evaluated using correlation and cosine similarity measures. In terms of performance, the SHAP-explained XGBoost model and the Logit generally achieved best results while SLIM underperformed, probably lacking some model complexity. From the explanation comparison, using both correlation and cosine similarity, it became clear that two pairings of similar models can be distinguished in SLIM-Logit and EBM-SHAP. This could be explained by the way SLIM and Logit are both fundamentally linearly formulated whereas EBM and SHAP can capture more complicated nonlinear relations. This difference is probably also reflected in the explanations.

Next, for comparing the explanations in absolute terms, synthetically generated data was created with a known relation containing different levels of nonlinearity. It was shown that EBM and SHAP achieved the highest predictive performance overall, but for very linear data relations their explanations were really different from the true relation. SLIM and Logit produced high predictive performance and explanation similarity for low nonlinearity levels, but both measures fell for both models as nonlinearity increased. SLIM performed considerably worse in both cases.

Finally, a survey was held among data scientists to test the general understandability of the different methods. SLIM came forward as easiest to understand and explain, but was generally not favored because of mediocre predictive performance. Most respondents gave preference for the SHAP because of good performance and some preferred Logit for its familiarity and simplicity.

The evaluation of the methods was performed only on UCI and synthetically generated data sets. Interpretability however, is also reliant on human understanding, which is best tested in practice. An extension of the research could thus be to include some real life application to test the different methods on. Another direction for further research could be a more extensive study on different synthetic data sets. The generated data

could be of bigger size or based on a different, more complex, data relation. The linearity nonlinearity trade-off already gives an interesting insight, but this could definitely be extended.

To conclude, all models provide explanations differently, but all within a similar interpretable additive framework. The clearest distinction can be made between either the simple linear models, SLIM and Logit and the more complex models, EBM and SHAP. For applications where predictice performance is similar, the data relation can be assumed to be linear and SLIM and Logit should be preferred, as they were shown to produce more correct explanations. However for more complex problems, the EBM and SHAP outperform the linear models in terms of both performance and explanation quality. In the end the human aspect also plays a big role for interpretable machine learning. It turns out experts will be more likely to trust methods that are widely known and accepted. This was found by Kaur et al. (2020) and confirmed in the performed survey.

In the end, the explanation value of different models depends on the application for which they are used. SLIM can be used for its great explainability, Logit is a solid simple and familiar method, SHAP-explained XGBoost provides high levels of accuracy and EBM can be used as a single model alternative to SHAP.

References

Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Dignum, V. (2017). Responsible autonomy. *arXiv preprint arXiv:1706.02513*.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized additive models*, volume 43. CRC press.

Kaur, H., Nori, H., Jenkins, S., Caruana, R., Wallach, H., and Wortman Vaughan, J. (2020). Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14.

Lakkaraju, H., Bach, S. H., and Leskovec, J. (2016). Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1675–1684.

Lou, Y., Caruana, R., Gehrke, J., and Hooker, G. (2013). Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631.

Lundberg, S. M., Erion, G. G., and Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774.

Molnar, C. (2019). *Interpretable Machine Learning*. `https://christophm.github.io/interpretable-ml-book/`.

Narayanan, M., Chen, E., He, J., Kim, B., Gershman, S., and Doshi-Velez, F. (2018). How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682*.

Nelder, J. A. and Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384.

Nori, H., Jenkins, S., Koch, P., and Caruana, R. (2019). Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*.

Štrumbelj, E. and Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665.

Ustun, B. and Rudin, C. (2015). Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, pages 1–43.

Ustun, B., Traca, S., and Rudin, C. (2013). Supersparse linear integer models for interpretable classification. *arXiv preprint arXiv:1306.6677*.

Vaughan, J. W. and Wallach, H. (2020). 1 a human-centered agenda for intelligible machine learning. *Machines We Trust: Ge ing Along with Artificial Intelligence*.

Appendix

A Performance Results

Table 11: The AUC performance statistic for the training sets of different models and data sets. The standard deviation over the 5 different folds is also displayed with best performing models in bold.

model	EBM		Logit		SHAP		SLIM	
	AUC	std. dev.	AUC	std. dev.	AUC	std. dev.	AUC	std. dev.
dataset								
bankruptcy	**1.000**	0.000	**1.000**	0.000	**1.000**	0.000	0.997	0.003
heart	0.954	0.021	0.936	0.009	**0.999**	0.000	0.894	0.009
haberman	0.823	0.026	0.705	0.018	**0.930**	0.018	0.688	0.029
breastcancer	0.999	0.001	0.996	0.001	**1.000**	0.000	0.991	0.003
mammo	0.860	0.006	0.860	0.006	**0.875**	0.004	0.811	0.018
spambase	0.699	0.268	0.978	0.005	**0.999**	0.000	0.914	0.024
mushroom	**1.000**	0.000	**1.000**	0.000	**1.000**	0.000	0.990	0.006
adult	0.891	0.001	0.891	0.001	**0.907**	0.003	0.847	0.006

B Summarized Training Set Similarity Metrics

Table 12: Table displaying the average over several data sets and train set folds for correlations between explanations of different models and their corresponding standard deviations for cleaned observations.

model	SLIM	Logit		EBM		SHAP	
		corr.	st. dev.	corr.	st. dev.	corr.	st. dev.
SLIM		0.672	(0.275)	0.336	(0.289)	0.436	(0.315)
Logit				0.456	(0.288)	0.547	(0.255)
EBM						0.626	(0.286)
SHAP							

Table 13: Table displaying the cosine similarity between explanations of different models on the training set averaged across several folds and different data sets .

model	SLIM	Logit		EBM		SHAP	
		cos.	st. dev.	cos.	st. dev.	cos.	st. dev.
SLIM		0.707	(0.239)	0.326	(0.325)	0.410	(0.375)
Logit				0.416	(0.361)	0.492	(0.369)
EBM						0.657	(0.308)
SHAP							

C Similarity Tables per Single Data Set

C.1 Test Set Results

Table 14: Bankruptcy average test set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.931	0.803	0.921
logit	0.931	0.000	0.902	0.944
ebm	0.803	0.902	0.000	0.879
shap	0.921	0.944	0.879	0.000

model	slim	logit	ebm	shap
slim	0.000	0.935	0.828	0.930
logit	0.935	0.000	0.920	0.954
ebm	0.828	0.920	0.000	0.905
shap	0.930	0.954	0.905	0.000

Table 15: Breastcancer average test set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.297	-0.120	-0.087
logit	0.297	0.000	0.156	0.225
ebm	-0.120	0.156	0.000	0.693
shap	-0.087	0.225	0.693	0.000

model	slim	logit	ebm	shap
slim	0.000	0.519	-0.165	-0.152
logit	0.519	0.000	-0.117	-0.076
ebm	-0.165	-0.117	0.000	0.857
shap	-0.152	-0.076	0.857	0.000

Table 16: Haberman average test set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.874	0.551	0.262
logit	0.874	0.000	0.358	0.101
ebm	0.551	0.358	0.000	0.774
shap	0.262	0.101	0.774	0.000

model	slim	logit	ebm	shap
slim	0.000	0.884	0.196	-0.025
logit	0.884	0.000	0.092	-0.108
ebm	0.196	0.092	0.000	0.878
shap	-0.025	-0.108	0.878	0.000

Table 17: Heart average test set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.713	0.386	0.421
logit	0.713	0.000	0.491	0.491
ebm	0.386	0.491	0.000	0.767
shap	0.421	0.491	0.767	0.000

model	slim	logit	ebm	shap
slim	0.000	0.721	0.415	0.450
logit	0.721	0.000	0.532	0.535
ebm	0.415	0.532	0.000	0.790
shap	0.450	0.535	0.790	0.000

Table 18: Mammo average test set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.880	0.535	0.723
logit	0.880	0.000	0.611	0.770
ebm	0.535	0.611	0.000	0.702
shap	0.723	0.770	0.702	0.000

model	slim	logit	ebm	shap
slim	0.000	0.888	0.586	0.749
logit	0.888	0.000	0.661	0.796
ebm	0.586	0.661	0.000	0.746
shap	0.749	0.796	0.746	0.000

Table 19: Mushroom average test set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.950	0.368	0.668
logit	0.950	0.000	0.395	0.713
ebm	0.368	0.395	0.000	0.359
shap	0.668	0.713	0.359	0.000

model	slim	logit	ebm	shap
slim	0.000	0.950	0.375	0.670
logit	0.950	0.000	0.402	0.714
ebm	0.375	0.402	0.000	0.371
shap	0.670	0.714	0.371	0.000

Table 20: Spambase average test set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.336	-0.002	0.325
logit	0.336	0.000	-0.003	0.712
ebm	-0.002	-0.003	0.000	0.016
shap	0.325	0.712	0.016	0.000

model	slim	logit	ebm	shap
slim	0.000	0.337	0.000	0.317
logit	0.337	0.000	-0.002	0.718
ebm	0.000	-0.002	0.000	0.002
shap	0.317	0.718	0.002	0.000

Table 21: Adult average test set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.529	0.402	0.488
logit	0.529	0.000	0.695	0.411
ebm	0.402	0.695	0.000	0.753
shap	0.488	0.411	0.753	0.000

model	slim	logit	ebm	shap
slim	0.000	0.532	0.419	0.509
logit	0.532	0.000	0.702	0.422
ebm	0.419	0.702	0.000	0.760
shap	0.509	0.422	0.760	0.000

C.2 Training Set Results

Table 22: Bankruptcy average training set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.926	0.791	0.904
logit	0.926	0.000	0.896	0.925
ebm	0.791	0.896	0.000	0.869
shap	0.904	0.925	0.869	0.000

model	slim	logit	ebm	shap
slim	0.000	0.931	0.820	0.918
logit	0.931	0.000	0.917	0.940
ebm	0.820	0.917	0.000	0.899
shap	0.918	0.940	0.899	0.000

Table 23: Breastcancer average training set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.275	-0.108	-0.079
logit	0.275	0.000	0.161	0.233
ebm	-0.108	0.161	0.000	0.704
shap	-0.079	0.233	0.704	0.000

model	slim	logit	ebm	shap
slim	0.000	0.503	-0.149	-0.138
logit	0.503	0.000	-0.095	-0.062
ebm	-0.149	-0.095	0.000	0.863
shap	-0.138	-0.062	0.863	0.000

Table 24: Haberman average training set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.830	0.240	0.152
logit	0.830	0.000	0.338	0.215
ebm	0.240	0.338	0.000	0.809
shap	0.152	0.215	0.809	0.000

model	slim	logit	ebm	shap
slim	0.000	0.855	0.073	-0.091
logit	0.855	0.000	0.156	-0.037
ebm	0.073	0.156	0.000	0.804
shap	-0.091	-0.037	0.804	0.000

Table 25: Heart average training set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.712	0.382	0.419
logit	0.712	0.000	0.488	0.486
ebm	0.382	0.488	0.000	0.773
shap	0.419	0.486	0.773	0.000

model	slim	logit	ebm	shap
slim	0.000	0.720	0.409	0.446
logit	0.720	0.000	0.527	0.528
ebm	0.409	0.527	0.000	0.795
shap	0.446	0.528	0.795	0.000

Table 26: Mammo average training set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.872	0.546	0.704
logit	0.872	0.000	0.607	0.748
ebm	0.546	0.607	0.000	0.703
shap	0.704	0.748	0.703	0.000

model	slim	logit	ebm	shap
slim	0.000	0.881	0.598	0.736
logit	0.881	0.000	0.659	0.779
ebm	0.598	0.659	0.000	0.749
shap	0.736	0.779	0.749	0.000

Table 27: Mushroom average training set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.941	0.430	0.615
logit	0.941	0.000	0.459	0.623
ebm	0.430	0.459	0.000	0.375
shap	0.615	0.623	0.375	0.000

model	slim	logit	ebm	shap
slim	0.000	0.941	0.437	0.619
logit	0.941	0.000	0.466	0.627
ebm	0.437	0.466	0.000	0.387
shap	0.619	0.627	0.387	0.000

Table 28: Spambase average training set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.290	0.001	0.288
logit	0.290	0.000	0.004	0.732
ebm	0.001	0.004	0.000	0.021
shap	0.288	0.732	0.021	0.000

model	slim	logit	ebm	shap
slim	0.000	0.295	0.000	0.286
logit	0.295	0.000	-0.002	0.740
ebm	0.000	-0.002	0.000	0.002
shap	0.286	0.740	0.002	0.000

Table 29: Adult average training set correlation (left) and cosine similarity (right) between explanation vectors of different models

model	slim	logit	ebm	shap
slim	0.000	0.527	0.401	0.488
logit	0.527	0.000	0.695	0.412
ebm	0.401	0.695	0.000	0.753
shap	0.488	0.412	0.753	0.000

model	slim	logit	ebm	shap
slim	0.000	0.530	0.417	0.509
logit	0.530	0.000	0.702	0.423
ebm	0.417	0.702	0.000	0.760
shap	0.509	0.423	0.760	0.000