ՇԻՐԱԿԻ Մ. ՆԱԼԲԱՆԴՅԱՆԻ ԱՆՎԱՆ ՊԵՏԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ
ШИРАКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ М. НАЛБАНДЯНА
M. NALBANDYAN STATE UNIVERSITY OF SHIRAK
УЧЕНЫЕ ЗАПИСКИ ԳԻՏԱԿԱՆ ՏԵՂԵԿԱԳԻՐ SCIENTIFIC PROCEEDINGS

Պրակ Ա
Выпуск А
Issue A

2021

№ 1

UDC 004.896
DOI 10.54151/27382559-2021.1a-29

INFORMATICS

COMPREHENSIVE EXPLANATIONS FRAMEWORK FOR MACHINE LEARNING MODELS TRAINED ON TABULAR DATA

Zakaryan A. G.

In the light of technological and scientific advances of the past decade machine learning models are becoming an inseparable part of many businesses. One of the shortcomings of ML models is the lack of transparency, which may result in a number of problems: hidden biases in the model, customer distrust, low adoption and usage etc. To increase the trust among the customers, the explainable techniques should find their way to the end customers in a digestible format. In this work, we will explore some ML explainability methods and provide a framework for presenting them in a comprehensive manner.

Keywords: explainable AI, SHAP, LIME, Machine Learning, Tabular Data.

1. Introduction. While the statistical methods are the foundation of the current machine learning algorithms have been around for centuries, the field itself didn't grow as fast as it did during the last couple of decades. A significant milestone in the development of this field was achieved in the 1990's, when most of the new discoveries were concentrated on the concept of "data-driven learning", instead of the existing "knowledge-based learning" [1]. This meant that scientists started developing algorithms for the computers to analyze large amounts of data and draw conclusions from the results. And with the advance of computers and computational power in general, the possibilities of machine learning increased exponentially. In today's world, it is hard to find an industry where machine learning and artificial intelligence aren't being applied and actively used. From healthcare to self-driving cars, from finance to law enforcement, every industry is trying to leverage the power of data to increase the efficiency of their businesses.

And as it gets more popular, it also receives a lot of backlash and resistance from many of the gatekeepers in those industries, and most of the time rightfully so. The reason for this is that most machine learning models, given the data, will learn the correlation between the input and the output, which usually is not enough to infer causation. And because of this, sometimes the models can contain biases and errors, which were not accounted for.

One example of such application of AI is Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) [2] used in many states of the US. Technically, it is a simple regression model that predicts the likelihood of a perpetrator to reoffend. While the creators of the system optimized their model for overall accuracy, the results showed that it was twice as likely to produce false positive results when the perpetrator was of African American descent.

Another example of bias is the conversational chatbot created by Microsoft, which was supposed to engage with people on twitter and was described as “casual and playful”[3]. Only a couple of hours after its release, Microsoft had to shut it down, as the contents of its tweets became extremely discriminatory.

These two examples showcase how negligence during the training process and the natural bias contained in the underlying training data can drive AI to bring more damage than to actually serve the intended purpose. And taking this as a basis, it is natural for many highly regulated fields to be resistant towards AI, as even the smallest mistakes can have destructive effects on human lives.

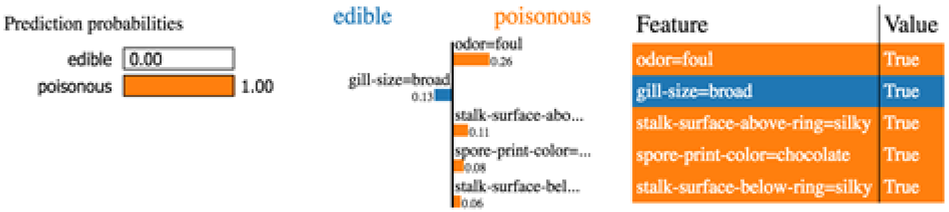
As a result of this resistance and the publicly perceived black-box image of AI systems, many consumers are still skeptical about AI powered products. This introduces the need of explainable AI.

Inherently, there are two approaches to adding explainability to machine learning models: designing the algorithm itself to be explainable (i.e. white-box models) or using approaches to explain the black-box model’s behaviour itself. While the former could be useful when the task at hand is not very complex, the latter is usually more relevant. This is because most advances in machine learning during the last decade are more result oriented and less transparency oriented. So in order to be able to take advantage of them, and still inspire trust to the users, adding explainability to black-box models is of higher interest.

However, in addition to making models explainable, researchers should be able to display the explanations in a comprehensive and preferably actionable way [4], as most of the end customers will not be tech savvy.

The aim of this paper is to describe a framework of generating model agnostic explanations that are both comprehensive and actionable.

2. Related Work. While development of white-box (i.e. linear or rule based) machine learning models was pretty common for a few decades now, adding explainability to black-box models is a relatively new area of research. An instance of a commonly used black-box ML model is Random Forest. Loupe et al. [5] was the first to introduce the concept of feature importances in his PhD Thesis, which provides global explainability to Random Forest based models. It is now widely used in one of the biggest machine learning packages in Python - scikit-learn. To take it one step further, Palczewska et al. [6] introduced the concept of interpretations of ML models, using a feature contributions method. The main idea of the work was to provide local explanations for each predicted instance. Over the next few years, Random Forest, which was considered one of the “strongest” algorithms at the time, started to get heavy competition from gradient boosted models like xgboost [11] and LightGBM [12], which in turn, introduced the need of creating a model agnostic explainability frameworks. For global explainability, one of the simplest explainability methods are Partial Dependence Plots (PDP) [14], which visualize the average prediction of the model for various values of a single feature, while other features are kept constant. A more robust version of PDP’s is Individual Conditional Expectation curves [15], which are the unpacked version of the PDP. That is, they show the model output for every instance for varying feature values, which gives more insights into the model. However, it’s rare that the relationship between a single feature and the target are easy to spot on these curves, so often times these plots are not enough. Another type of model agnostic explainability methods is the global surrogate technique [7], the main idea of which is to fit an explainable model (i.e. decision trees, linear models) to the predictions of the original, black-box model. This, again, will provide global interpretability to the models, but the quality of explanations will highly depend on the quality and stability of the surrogate model. Based on this idea, Ribeiro et al. [8] came up with LIME (Local Interpretable Model-Agnostic Explanations), which, for any predicted instance, augments the input data, and fits a linear model to the predictions, to estimate the impact of feature changes on the model output. This method is highly effective for researchers and developers to get more insights on how the models work, but it is not very effective in production environments (as shown in later sections).



Feature	Value
odor=foul	True
gill-size=broad	True
stalk-surface-above-ring=silky	True
spore-print-color=chocolate	True
stalk-surface-below-ring=silky	True

Figure 1: LIME Local explanations

In an attempt to unify the approaches described above, Lundberg et al.[9] developed SHAP (Shapley Additive exPlanations), which has its roots in the game theory (Shapley Scores). This method is more robust as it provides both global and local interpretations, while also efficient and stable enough to work in production environments.

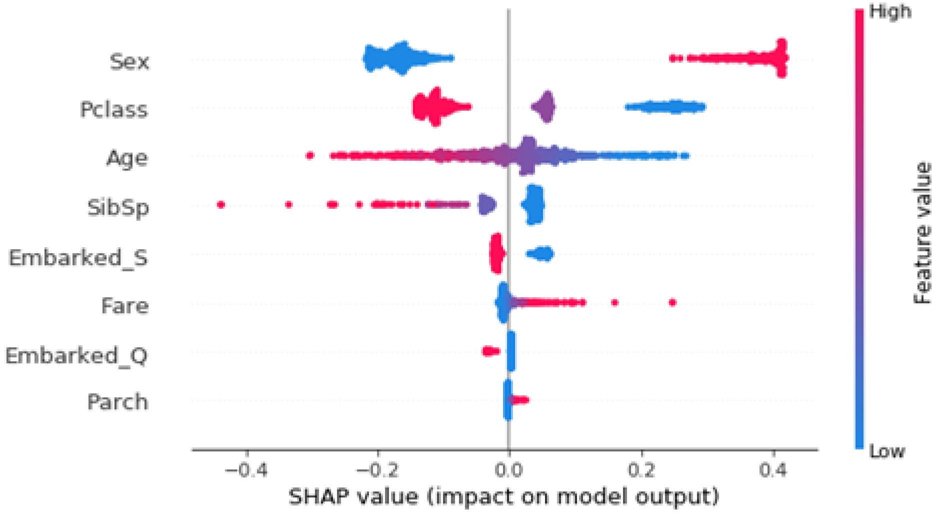


Figure 2: SHAP Global Explanations

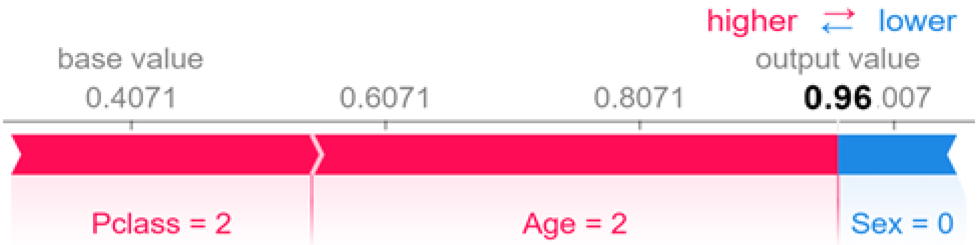


Figure 3: SHAP Local explanations

Some of the approaches described are also applicable to deep learning models, but as deep learning is sometimes separated from traditional machine learning, there are some explainability techniques that are only applicable to deep neural networks. Hendricks et al. [10] describe a framework on generating explanations for DNNs by passing the features found by DNNs to a recurrent neural network, which generates a sentence in plain English describing the features. This approach is pretty powerful in generating explanations in a consumable format, however it is pretty specific to image classification tasks (by DNNs). In the next section, some of the described approaches relevant to this work will be explored in depth.

3. Comprehensive Explanations Framework. Machine learning models can be used in various scenarios, from research environments to serving predictions to millions of customers. This work primarily focuses on the latter scenario, that is enabling the end customer to understand the reasoning behind the served predictions in a comprehensive manner. Additionally, to increase engagement and trust, the explanations should be actionable. So, we formally divide the framework into two layers: the *explainer* and the *translator*. The former will be responsible for extracting raw explanations from the models, while the latter will ensure that the explanations are comprehensive and actionable.

3.1. Explainer Layer. First, we fix the process of generating explanations. Based on the business case described earlier in the section, the framework requires local explainability, to provide customized experience to each individual user. The output of the layer should be a standardized list of <feature, impact> pairs, so the next layer will build the translations based on that. While in the scope of this work, we only apply and compare the most common local explainers - SHAP and LIME, the structure of the layer is explainer agnostic. This means that any algorithm, the output of which can be brought to the aforementioned format will be relevant and applicable.

3.2. Translator Layer. Consider a model that decides whether the end customer should get his/her credit application approved or not. The underlying model could use a number of features, which can be split into two categories: actionable and non-actionable. We define actionable features as features on which the end customer has direct or indirect control, while non-actionable features are things the end customer cannot change. In this case, features like *gender*, *age*, *ethnicity* will be considered non-actionable, while the credit score, the consistency of past payments, current salary will be considered actionable. While displaying explanations based on non-actionable features might be a requirement in many industries, usually it will have a discouraging effect on the end customer [13], and should be avoided whenever possible. The process of categorizing the features must involve both the data scientist and the product owner. In this way, we ensure that the selected features and generated explanations are in line with the end customer's understanding of the field, to avoid cognitive dissonance.

Showing the users actionable explanations will engage them more and feel more in control and confident in the predictions provided by the machine learning system. So, to comply with the framework, the final feature set that is being used for the model should clearly separate the actionable and non-actionable features. This way, the framework will provide a better user experience and increase customer engagement [13], which is not feasible by using the common explainability methods as is.

3.3. Limitations. In the context of this framework, human intervention is needed to split the input features of the model into actionable vs non-actionable features. As mentioned above, the human intervention should come from both a data scientist and a product owner, to include the customer's perspective. This naturally limits the usage of this framework in small companies, as they may not have the outreach yet, or not have designated product owners. In these cases, they may resort to the common explainability methods, until they reach the level of maturity that can employ the resources needed to accomplish the feature division.

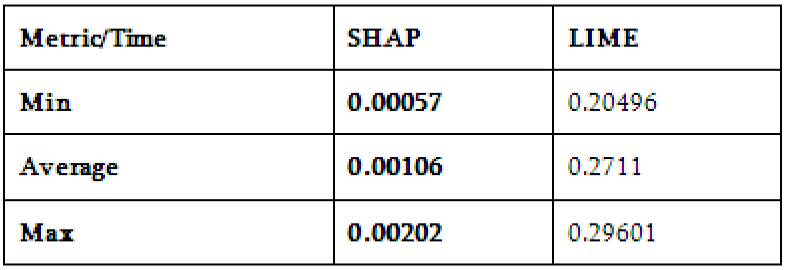
4. Comparison and Results. As mentioned in the previous section, in this work we compare two explainer algorithms powering the explainer layer - SHAP and LIME. The comparison is made based on the following 3 criteria: *stability*, *performance* and *actionability*.

4.1. Stability. Stability in our context refers to the consistency and "truthfulness" of explanations of the prediction. To compare this, we consider the fundamental difference between the two explainers: SHAP explains the predictions themselves, while LIME fits simpler linear models in the local neighborhood of an instance and explains the base model by using the linear model as a proxy. To add to that, LIME uses random augmentation technique to generate a sample of data points around the instance that needs to be explained. This process in itself also adds to the instability of this method. To make LIME more stable, the number of augmented data points be increased, which in turn affects the performance (see: next section). But even with providing a large enough sample of data points, LIME still has the risk of underfitting the surrogate linear model, and thus providing "untruthful" explanations.

4.2. Performance. Given the business context in which our framework is intended to be used, we define performance as the time it takes to generate a single explanation. In this experiment, we used a lightgbm regressor model fitted on a training data of size 300x5. The data itself is completely randomly generated in range [-100, 100], with the target variable being a non-linear combination of the features. All the experiments were done on a machine with an Intel Xeon E-2176M CPU 2.70Ghz and 32Gb RAM. After the training, we used both explainers to generate explanations for 100 samples and considered the average, minimum and maximum time it took run. The results showed that SHAP is significantly faster than its competitor. This is not surprising, as the basis of the algorithms are different. LIME fits a linear model for each prediction, which considerably slows it down, compared to SHAP, which uses the model itself to estimate local feature impact.

Table 1

Benchmarks of LIME and SHAP on a set of 100 explanations



Metric/Time	**SHAP**	**LIME**
Min	**0.00057**	0.20496
Average	**0.00106**	0.2711
Max	**0.00202**	0.29601

4.3. Actionability. We define the degree of actionability of the explanations as the answer to the question "What should I do to change the prediction in a specific direction?". In general, both considered methods are not suitable for answering this question, but in specific contexts, they can prove to be useful. For regression and binary classification tasks, the results of two techniques are pretty similar. In these cases, LIME slightly outperforms SHAP, by providing a little more context on feature impact. Besides only showing a specific value of a feature that affected the prediction, LIME also provides a range for that feature's values, within which the predictions won't change. This gives the end users a bit more insight on where the feature's value should be, for it to possibly affect the prediction.

For multiclass classification tasks, SHAP provides more actionability. In this context, LIME just outputs features' value ranges which drove the prediction towards that specific class. With SHAP, we also get insights on which feature values drove the prediction away from the other classes. So if users want their predictions to change classes, using SHAP's output will be more handy.

5. Conclusion. In the article we have described and implemented a framework for generating comprehensive explanations for machine learning models trained on tabular data. We proposed a modular, two-layer, abstract approach, that can, in theory, support any concrete explanation techniques and translation languages, which correspond to the format described above. We've also compared two implementations of the explainer layer to assess their "product readiness" in terms of 3 different criteria. For further research, we consider adding more explanation techniques to the framework, which will improve the actionability aspect. Also, building a language model similar to what was described in [10], to eliminate the human efforts in translating the explanations to human readable sentences.

ՀԱՄԱՊԱՐՓԱԿ ԲԱՑԱՏՐՈՒԹՅՈՒՆՆԵՐԻ ՀԱՄԱԿԱՐԳԸ ԱՂՅՈՒՍԱԿԱՅԻՆ ՏՎՅԱԼՆԵՐԻ ՎՐԱ ՎԱՐԺԵՑՎԱԾ ՄԵՔԵՆԱՅԱԿԱՆ ՈՒՍՈՒՑՄԱՆ ՄՈԴԵԼՆԵՐԻ ՀԱՄԱՐ

Զաքարյան Ա. Գ.

Անցած տասնամյակի տեխնոլոգիական և գիտական առաջընթացի հաշվին մեքենայական ուսուցման (Մ.Ու.) մոդելները դառնում են շատ բիզնեսների անբաժանելի մաս: Մ. Ու. մոդելների թերություններից մեկը թափանցիկության բացակայությունն է, որը կարող է հանգեցնել մի շարք խնդիրների՝ մոդելի թաքնված կողմնակալություն, հաճախորդի անվստահություն, քիչ օգտագործում և այլն: Հաճախորդների շրջանում վստահությունը բարձրացնելու համար բացատրելի տեխնիկաները պետք է հասկանալի ձևաչափով մատուցվեն հաճախորդներին: Այս աշխատանքում մենք կուսումնասիրենք ՄՈւ-ի բացատրելիության որոշ մեթոդներ և հիմք կստեղծենք՝ դրանց համապարփակ ձևով ներկայացնելու:

Բանալի բառեր. բացատրելի արհեստական բանականություն, մեքենայական ուսուցում, աղյուսակային տվյալներ:

КОМПЛЕКСНАЯ СТРУКТУРА ОБЪЯСНЕНИЙ ДЛЯ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ, ОБУЧЕННЫХ НА ТАБЛИЧНЫХ ДАННЫХ

Закарян А. Г.

В свете технологических и научных достижений последнего десятилетия модели машинного обучения становятся неотъемлемой частью многих предприятий. Одним из недостатков моделей машинного обучения является отсутствие прозрачности, что может привести к ряду проблем: скрытым предубеждениям в модели, недоверию клиентов, низкому уровню принятия и использования и т. д. Чтобы повысить доверие среди клиентов, объясняемые методы должны найти свой путь к конечным потребителям в удобоваримом формате. В этой работе мы исследуем некоторые методы объяснимости машинного обучения и дадим основу для их комплексного представления.

Ключевые слова: объяснимый ИИ, SHAP, LIME, машинное обучение, табличная дата.

REFERENCES

1. https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/?sh=6ecdb0c115e7 (17.02.2021)
2. Shadowen, Nicole A. "Ethics and Bias in Machine Learning: A Technical Study of What Makes Us "Good"" (2017). *CUNY* Academic Works. https://academicworks.cuny.edu/jj_etds/44 (17.02.2021)
3. In 2016, Microsoft's Racist Chatbot Revealed the Dangers of Online Conversation - IEEE Spectrum (17.02.2021)
4. Dykes B. Effective Data Storytelling: How to Drive Change with Data, Narrative and Visuals. Wiley. 2020. 336p.
5. https://www.researchgate.net/publication/264312332_Understanding_Random_Forests_From_Theory_to_Practice (17.02.2021)
6. https://www.researchgate.net/publication/303571296_Interpreting_random_forest_models_using_a_feature_contribution_method (17.02.2021)
7. https://www.researchgate.net/profile/Patrick_Hall20/publication/328160693_On_the_Art_and_Science_of_Machine_Learning_Explanations/links/5e668422a6fdcc37dd13954c/On-the-Art-and-Science-of-Machine-Learning-Explanations.pdf (17.02.2021)

8. https://www.researchgate.net/publication/305342147_Why_Should_I_Trust_You_Explaining_the_Predictions_of_Any_Classifier (17.02.2021)
9. https://www.researchgate.net/publication/317062430_A_Unified_Approach_to_Interpreting_Model_Predictions (17.02.2021)
10. https://www.researchgate.net/publication/336131051_Explainable_AI_A_Brief_Survey_on_History_Research_Areas_Approaches_and_Challenges (17.02.2021)
11. https://dl.acm.org/doi/pdf/10.1145/2939672.2939785 (17.02.2021)
12. https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf (17.02.2021)
13. https://www.jstor.org/stable/4166182 (23.03.2021)
14. https://www.jstor.org/stable/2699986 (23.03.2021)
15. https://arxiv.org/pdf/1309.6392.pdf (23.03.2021)

Information about the author

Zakaryan A. G. *- PhD Student, Institute of Informatics and Automation Problems*
Master: American University of Armenia (Computer and Information Systems) of NAS RA
E-mail: arman_zakaryan20@alumni.aua.am

Received by the editorial office on 17.02.2021
Reviewed on 23.03.2021