GBEx – towards Graph-Based Explainations

Paweł Mróz[1,3], Alexandre Quemy[1,2],
Mateusz Ślażyński[3], Krzysztof Kluza[3], and Paweł Jemioło[3]

[1] IBM Krakow Software Lab, Cracow, Poland
pawel.j.mroz@ibm.com, aquemy@pl.ibm.com
[2] Faculty of Computing, Poznań University of Technology, Poznań, Poland
[3] AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Krakow, Poland
{mslaz,kluza,pjm}@agh.edu.pl

Abstract. This paper proposes a new concept of Graph-Based Explanations (GBEx), which represent explanations using graphs. The approach presents a view on explanations in the form of a graph, where nodes represent arguments, and edges represent connections. The value of a graph node accounts for the influence of a given argument while the value of a graph edge accounts for the influence of a given connection. GBEx requires data to be converted to a binary form and provides the visualization in the form of a graph. Moreover, arguments and connections can be analyzed separately in a different manner as well, i.e. nodes can be presented on a bar chart and edges can be reflected with a heatmap.

Keywords: explainable artificial intelligence, XAI, graphs, features

1 Introduction

Nowadays, the most advanced machine learning algorithms are black-box models. That is, there is no easy way to comprehend what is happening inside a model. Due to a lack of understanding of the processes inside algorithms, there is no straightforward way to answer questions such as:

- Why, under given circumstances, an algorithm made such a decision?
- What is the most important feature in the dataset?
- Does the model take into consideration irrelevant data?

To answer the above-mentioned questions, scientists and researchers came up with a few ideas to provide the tools that make Artificial Intelligence (AI) more reliable [1,2,3]. With more complex and efficient models, there is huge potential to start applying machine learning algorithms to new fields. Nevertheless, sometimes the accuracy or other effectiveness indicators are not enough to trust a model. There are already attempts to apply AI in domains having a significant impact on some people's lives like medicine [4] or the judicial system [5]. In these cases, it is crucial not only to have a good model but also to be able to see on what basis the algorithm makes its predictions.

In this paper, we propose yet another method enabling interpretability of the black-box models. The main contribution of the paper is a new approach for artificial intelligence systems enhancing their transparency. Our solution relies on knowledge graphs for presenting explanations. We compare our method to the selected state of the art methods and provide validation on a real-life example.

The rest of the paper is composed as follows. In Section 2, we present the current state of the art concerning the methods for explainability. Moreover, we focus on their advantages and drawbacks. Section 3 describes our algorithm enabling interpretable explanations with graphs. In Section 4, we provide validation of our ideas. The paper ends with Section 5, where we summarize our work and outline our future research directions.

2 State of the art

According to a vast review on the subject [1], understandability is a key concept of explainable AI methods. Intelligible model is a model, which functionality can be comprehended by humans. This particular characteristic is closely tied to interpretability and transparency. As the authors suggest, the first factor is a measure of the degree to which an output can be understood by persons. The second trait, in turn, refers to the inherent internal features of the specific models, i.e. possibility to easily follow the process leading to generate the output or ease in the presentation of results.

This capability can be observed notably among primary Machine Learning techniques like linear regression or tree and rule-based models. Nevertheless, with Deep Neural Networks gaining its popularity among researchers and companies, transparency is no longer an asset of the AI systems. Using black-box models created a gap between performance and explainability of models that are used in learning [6].

When talking about more sophisticated Machine Learning methods, post-hoc explainability is usually applied. It means that some additional techniques are added to the model to make its decisions justified and understandable. According to [1], there are two types of adapted strategies: model-agnostic which can be used to any type of Machine Learning model, and model-specific, which must be applied in correspondence with a selected learning tool.

Local Interpretable Model-AgnosticExplanation (LIME) [7] is one of the most widely-used explainable methods. It focuses on building linear models which approximates and simplifies the unintelligible outputs of primary solutions.

The main advantage of this approach is that it is possible to explain every black-box model with a variety of interpretable models, often with good accuracy. It is very flexible in terms of choosing algorithms. For example, linear regression could explain Xboost, and a decision tree can elucidate the structure of the neural network.

One of the problems with LIME is its locality. Although it usually works well and is good enough to be applicable, there is an issue of trust. The more delicate task, the more global explanation might be required. Another problem

is the definition of neighborhoods. With some datasets, it may be a difficult task to come up with such a way to alter the sample, that it definitely could be seen as its neighbor.

Additionally, there are SHapley Additive exPlanations (SHAP) [8], which is the measure of certainty calculated for each prediction basing on features relevance in terms of the task. This approach has many advantages. It is based on a well known and acknowledged mathematical theory, i.e. game theory. Opposite to LIME, it is fairly distributed among the whole dataset, not only locally. Moreover, as it was shown in [8], it satisfies four properties that are required for a model to be able to produce a fair payout: efficiency, symmetry, dummy and additivity.

Nonetheless, there are also some disadvantages of this method. In practical use, the biggest issue with this approach is the computational time needed to produce the explanation. The number of calculations required for real-life data is huge, and it grows rapidly with a growing number of features. With contemporary computers, it is often not possible to compute the output in the required time. In some cases, especially in problems with a large number of features, produced explanations might be confusing for people, and there is no way to reduce the explanations as it happens in LIME. Finally, generating all possible coalitions might lead to predicting some absurd or impossible cases.

3 GBEx – Graph-Based Explanations

The main objective of Graph-Based Explanations (GBEx) is to represent explanations in a graph such that:

1. the value of a node in a graph represents the influence of a given argument,
2. the value of an edge in a graph represents the influence of a given connection.

Therefore, it forms an interpretable surrogate model that approximates output from a black-box AI algorithm.

GBEx is made of two levels of abstraction. Most of the explanation models are focused on giving a certain value of importance to the specific argument. Sometimes also considering the influence of different level interactions like SHAP and sometimes not, like for example simple tree-based explanations. In this approach, the goal is to add another layer of abstraction, which is to give a value of importance to a connection between specific arguments.

The basic formula would look like this:

$$\hat{y} = W^1\mu^1 + W^2\mu^2 + \beta \tag{1}$$

where:

- $\hat{y}$ – the vector to approximate from the AI model that is explained.
- W^1 – the matrix of inputs. Every case is represented by one row. 0 states for the absence of an argument and presence is valued 1 divided by the number of present arguments in a specific case.

- μ^1 – the vector of node importance. Each value marks the influence of the corresponding argument to the output.
- W^2 – the matrix of connections. Similarly to inputs, each row represents some case. 0 means absence of connection and 1 divided by the number of present connections means that both arguments are occurring in a given case.
- μ^2 – the vector of edge importance, also similarly as for nodes this variable holds information about the influence of given connection to the output.
- β – the base value. When no arguments given, this is the predicted value.

There were quite a few challenges that needed to be faced when working with GBEx. First two of them are specific to our approach. They focus on ways to prepare data to be suitable for the algorithm and how to solve the Equation 1. The last two issues are connected with problems that are common to almost every method used for extending interpretability, namely, the time that is needed to compute explanations and how to present it in a human-friendly way.

3.1 Binarization and clustering

Input matrix W^1 contains the information about the presence or absence of an argument in given cases. However, most datasets contain non-binary features. Typically there can be distinguished two types of data: categorical and floating.

Categorical elements could be transferred to the one-hot vector, where each position in the vector represents one of the classes [9].

Floating values are the more challenging ones. The simplest way would be to treat them as categorical features, where every appearing value is considered as a different category. However, such an approach has some severe disadvantages. The number of arguments and what follows the number of connections could be pretty huge. What follows is that the number of intersection between cases would be very low, which makes it hard to find relevant statistics. That would cause an increase in the time needed to complete computations. Also, interpretability would be seriously strained with the growing number of arguments to take into account.

One of the possible approaches to this problem is to cluster the features so that similar values are treated as the same. There are a few different ways that clustering could be done. The first example is rounding the numbers up. It is quick and easy, but it requires knowledge of how the data looks so there might be a chosen point to which these numbers are rounded. Another approach could be k-means. This method is more complex and timely but does not need to have knowledge about features, and the clustering is done really well.

Doing clustering is connected with a certain trade-off. Simple linear or logistic regression assigns one weight to the given feature, which means that on the whole range that values of this column might change, the effect of the same difference will always have the same impact. This is not the case after binarization. Each group that will be created with clustering has its independent value assigned. This gives an algorithm more liberty when it comes to finding the best weights.

Nevertheless, giving up floating values also has its cost in a mean of losing information about the distance between certain points. Naturally, the centers of the clusters are known, but the information about differences between the points in one group is gone, and between points form different clusters, it is reduced to the distance between centers.

It is important to note not to create redundant edges while prepossessing data. That is to omit connections between nodes that represent the same feature. Thanks to that size of w^2 matrix is not unnecessarily bigger.

3.2 Solving equation

Solving equation $y = Ax+b$ is fairly simple, either using a direct matrix inversion or, if the matrix is too large, an iterative approach. Linear regression does a pretty good job of minimizing an error. Furthermore, logistic regression works well for classification problems. Adding another layer of abstraction complicates the situation. In the Equation 1, variables μ should approximate the output $\hat{y}$ as good as possible, but without losing the attribute of interpretability.

One approach is to separate the task into two simple ones. First, linear regression solves a simple equation like the following:

$$\hat{y} = W^1\mu^1 + \beta^1 \tag{2}$$

Then the error that is left $e = \hat{y} - W^1\mu^1 - \beta^1$ is approximated in the same way by the second part:

$$e = W^2\mu^2 + \beta^2 \tag{3}$$

Assuming that $\beta = \beta^1 + \beta^2$, it agrees with basic Formula 1.

Although this rather trivial approach gives pretty good results, the interpretability of connection is something that is still in question. Another downside of this method is that it is hard to apply it for classification problems, where logistic regression should be used.

An approach that would not put in doubt its interpretability is one that could learn all the features at one time. That is possible with some heuristic algorithms like genetic or tabu-search. These methods have their drawbacks as they do not guarantee to find the best possible value.

Another approach that would solve the equation at once is converting this problem into a simple one linear regression task. That could be done by creating a matrix W^0 in the following way:

$$W^0 = [W^1 \;\; W^2] \tag{4}$$

Similarly, vector μ^0 would look like:

$$\mu^0 = \begin{bmatrix} \mu^1 \\ \mu^2 \end{bmatrix} \tag{5}$$

And finally, the Equation 1 could be transformed to the following form:

$$\hat{y} = W^0 \mu^0 + \beta \tag{6}$$

That approach guarantees to find the best possible solution while being able to solve the task without splitting the main task into two. Unfortunately, this has also its disadvantages in computational cost as it requires to invert the matrix W^0 of a huge size.

3.3 Presenting results

Presenting results is a crucial part of the explanations. The whole point of creating interpretable algorithms is to be able to show results in a human-friendly way. One of the primary goals of creating GBEx is to be able to present results in a graph form. Graphs posses an extremely important feature, that is an easiness to extract relevant pieces of information by simply looking at the graph, while still being able to hold much knowledge inside the structure.

Graph structures are widely used in many areas of science and are well known mathematical objects. Additionally, using graphs allows for adding the second level of abstraction, the connections between arguments, without losing clarity.

Taking into account these arguments, we decided to present explanations with arguments as nodes and relationships between these arguments as edges.

- Size – this parameter, applied to node or edge, is representing the importance of an argument or connection. The direction is not taken into account.
- Color – to complement lack of direction from size, the color of node or edge is showing which output this particular argument or connection is supporting. The difference in intensity of color points out how strong this support is.

A few conclusions might be drawn from this enhancement. First, the intensity of colors and size of elements are redundant, which means they point out the same quality of explanation. Depth of color, however, is significant because it is good to be able to look at just one quality and tell which of elements is supporting more and to which output. Moreover, it is easier to compare the importance of two opposing arguments by looking at size rather than how intense some red is in contrast to the strength of other green. This is why we decided to keep this redundancy.

The second conclusion might be that, depending on the number of clusters, this graph can get huge. We thus propose two ways of presenting explanations:

1. General – the explanation is focused on the whole dataset and every possible argument. To comprehend the amount of data presented, data from the same node are merged. Thus, if one feature was clustered into five groups, then the average of the importance of these fractions is taken. The same happens for binary or multiclass features. In that case, the knowledge about support for the given output is not really meaningful, and for that reason, it is omitted.

2. Specific – the explanation is focused on one certain case. In this scenario, only presented data is the same as for the whole dataset, but only with nodes and edges that appear in this specific case. The values are refactored. The importance is measured for the given case and not for the whole dataset.

A graph seems the best way to present the importance of both arguments and connections in one picture. However, there are also other methods that could better display the parameters of the model separately.

Significance of an argument can also be presented at a simple bar chart. Such a structure shows clearly which feature is most important. It could easily plot the direction to which this argument is directing and how strongly. To plot only absolute importance, a pie chart could be used as well. This method shows visibly how strong impact on the final decision each argument has.

The same methods could be used to present explanations on a connection level. A clear and easy method to plot the dependency of connections is a heatmap. This structure was chosen because of its scalability. Even big connection matrices could be visualized by heatmap in a very straightforward form.

3.4 Relation to state of the art

As calculating Shapley values is computationally extremely expensive, SHAP uses other well-known methods like LIME or DeepLift. The huge downside of the first one is its locality. This means that explanation would only be valid in the neighborhood of explained case. The effect of this quality is that plenty of counter-examples could be found. This would undermine the meaningfulness of the explanation. GBEx is able to produce a model that is accurate on the whole data space.

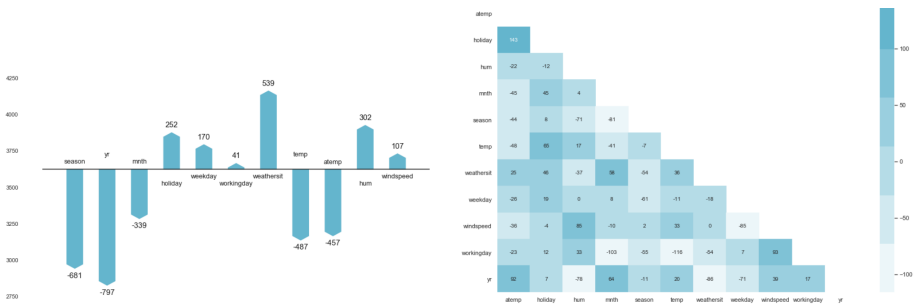
Another upside of GBEx is that while classic Shapley values take into account all possible coalitions between arguments, there is no information on how much of the contribution of feature comes from the feature itself and how much comes from the interactions. GBEx has a separate layer just for connections. The higher level of abstractions could also be considered.

4 Validation

For this research, we decided to use a well-known example, namely *Bikerental* dataset [10] for regression task for the first two cases. The last test was performed on *skin* dataset [10] for classification.

4.1 One case explainations

In the first example, we used Multi-Layer Perceptron (MLP) Regressor. The task of GBEx was to mirror the output of the MLP and give parameters needed for interpretation. The first approach is using a specific case and checking what contributed to the prediction in terms of features and connections that exist between them. Parameters that are helpful to understand the explanation are the following:



(a) Feature. (b) Connection.

Fig. 1: Influence to the final prediction in chosen case

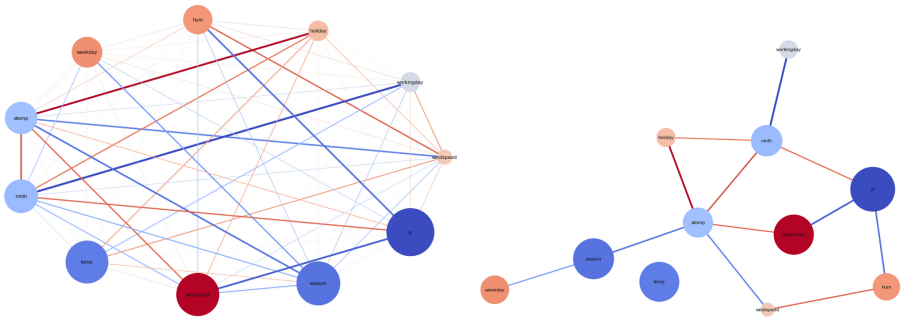
- Real value = 1011.
- Value predicted by MLP Regressor = 2026.
- Base value = 3624.
- Value produced by GBEx = 1939.
- Value produced only by nodes from GBEx = 2274.

The first conclusion is that MLP Regressor has not worked that well, given the difference between its prediction and real value. Nevertheless, the job of GBEx is not dependent on the quality of this model. The influence of arguments to the final prediction is shown in Figure 1a. Connections were not taken into consideration in this example. The starting point of this plot is a base value which is set to be 3624 bikes rented. According to the plot, the most influential features are 'Season' and 'Year'.

Complementary to the arguments, there are also structures created to present the influence of connections between given nodes. As was mentioned the best structures to do so are heatmaps. Continuing the previous example, the heatmap presented in Figure 1b shows how a given coalition is contributing to the final prediction.

One of the goals to create GBEx was to be able to present an explanation in a graph form. The previous example showed how a case could be presented separately for arguments and connections. The explanations were clear and easy to understand. Exact values are visible to be able to tell precisely how they influence the final output. Its drawback is that they have to be presented separately. It is not simple to grasp general knowledge of the whole explanation. That is why in Figure 2a another solution is presented. The same case is used as in the previous example.

With a huge amount of nodes, there are many more connections. To simplify the graph, there is a possibility to cut off less important edges and present only those whose influence is most significant. In Figure 2b is showed the simplified graph, that still contains most of the useful information and is much clearer and more readable.

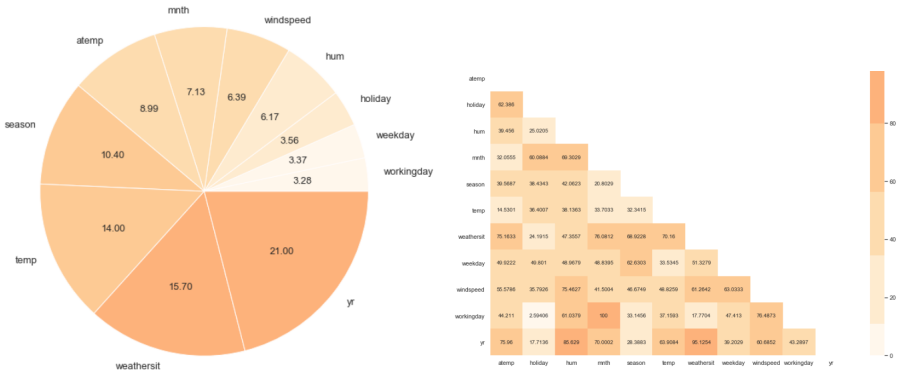


(a) Full. (b) Simplified.

Fig. 2: Explanation graph, showing general importance of arguments and connections.

4.2 General explanation

On advantage of GBEx is that explanations are made not only one a specific case level, but also it contains general pieces of information about the whole problem. As was already mentioned, to create this model, there is a need to merge the arguments coming from the same feature. Merging is done by taking the average absolute value of the influence of arguments.



(a) Argument. (b) Edges.

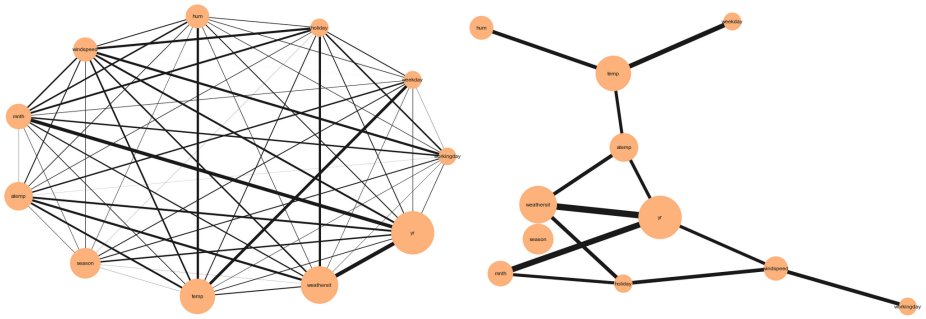
Fig. 3: Importance in general model.

Continuing example from the previous section of *bike sharing* dataset, in Figure 2b there is shown pie chart presenting mean importance of given argument over the whole dataset. The plot is intuitive and simple to read. It shows the average distribution of features contribution to the whole space. The disadvan-

tage of this approach is the inability to show the direction in which arguments are supporting.

Results from these two examples seem to reasonably explain how the predictions are made. As would be expected weather arguments have a lot of influence. Given that the popularity of the bike-sharing business is growing with time, it is also understandable that year was the most important feature. Least important are columns related to the day of week or holiday, which is reasonable given that those are not key factors when deciding whether to rent a bike, compared to weather conditions.

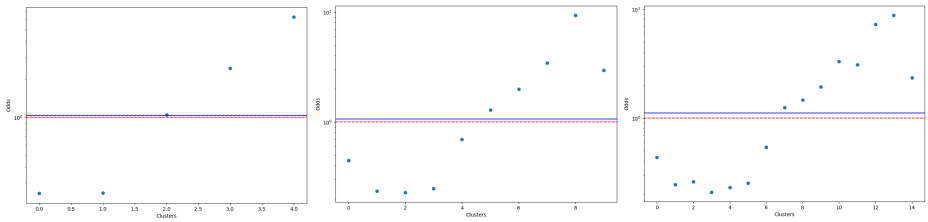
It is much harder to argue which connections should or should not have an influence on the final decision. Taking meaning out of this structure is a challenging task. The strength of a connection is representing the situation when both arguments combined are not enough to make a correct prediction, and they have to be supported with an additional parameter. Given that, it is difficult to judge how well the GBEx performed in this task. The conclusion is that if a model has a high score of metrics, it is fair to trust its explanations.



(a) Full. (b) Simplified.

Fig. 4: Explanation graph, showing general importance of arguments and connections.

General explanations also provide a possibility to present data in a graph form. But given a lack of ability to provide discriminatory support, the color of nodes does not hold any information. The strength of influence is represented by the size of nodes as well as edges. In Figure 4a there are showed explanations, coming from the previous example, but presented in a graph form. This representation clearly indicates which arguments and connections are the most important. For better visibility, there is also a simplified version with fewer edges showed in Figure 4b.



(a) 5 clusters. (b) 10 clusters. (c) 15 clusters.

Fig. 5: Support of third feature from *skin* dataset.

4.3 Feature analysis

Apart from classic interpretation and explanation, there is also a matter of analyzing each feature. Binarization and clustering may cause the loss of information about the exact distance between two values from the same column. Nevertheless, in return, there is a possibility to create a deeper and more complex model. Feature analysis is focused on how the support is changing within different clusters. However, the crucial questions that needed an answer were how the number of groups that the division is made might influence interpretability.

In Figure 5a, there is presented the influence of each of 5 clusters made from the third feature. The group with the higher number represents the bigger values. The red dashed line marks the neutral point that means the feature supports neither positive nor negative outcome. As this is the classification task, the odds are presented on a logarithmic scale. The blue line points out the base value generated by the logistic regression. The dependency seems to be clear: the greater the variable, the more support of a positive outcome. On the other hand, in Figure 5b, there are shown quite different results. The middle ones of 10 clusters present similar behavior as for a smaller number of groups, but on the sides, there can be observed reversed behavior. These are those more complex dependencies that could not be observed in traditional approaches. In spite of the increasing number of variables, interpretation of this structure is still pretty clear. In Figure 5c there is shown support of the same feature divided into 15 clusters. The shape created by these values is similar to the previous ones but is more complex. That shows that even if the structure could produce a better projection of AI algorithms, its interpretability might decrease with too many variables to analyze.

This example is an argument to the conclusion that choosing the number of clusters is the binarization part in one of the most important decisions. Also, it clearly shows the tradeoff between the interpretability and complexity of a model. Naturally, not all features would have that kind of easy dependency at 10 clusters. Sometimes that limit might be quite larger, but more probably it could be lower. Arguably, 4 or 5 groups seem to be the maximal number of divisions that could almost always be easy to interpret.

5 Conclusions and Future Work

This paper aimed to propose Graph-Based Explanations as a new concept for creating a surrogate model, that can interpret decisions made by some algorithm. This approach presents a view on explanations in the form of a graph, where arguments are represented as nodes and connections are represented by edges. Although it is possible to make the visualization in the form of a graph, arguments and connections can be analyzed separately in a different manner. Nodes can be presented on a bar chart, while edges may be reflected with a heatmap.

Right now the algorithm does not scale well, so it is crucial to increase its efficiency. Given that the solutions use other computationally demanding algorithms, it would not be possible to decrease the complexity to linear or better, but there is a chance of improving its performance. It would be beneficial to develop and include methods for deciding the optimal number of clusters. For now, it is a matter of choice for data analysts. It is doubtful that it is possible to find a universal solution to all scenarios. Nevertheless, more thorough research might help to find some patterns that show how to look for the optimal value for a given problem.

References

1. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.: Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion (2019)
2. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access **6** (2018) 52138–52160
3. Molnar, C.: Interpretable machine learning: A guide for making black box models explainable (2018)
4. Wang, L., Wong, A.: COVID-Net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest radiography images. arXiv preprint arXiv:2003.09871 (2020)
5. Deeks, A.: The judicial demand for explainable artificial intelligence. Columbia Law Review **119**(7) (2019) 1829–1850
6. Došilović, F.K., Brčić, M., Hlupić, N.: Explainable artificial intelligence: A survey. In: 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO), IEEE (2018) 0210–0215
7. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should i trust you?: Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, ACM (2016) 1135–1144
8. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems. (2017) 4765–4774
9. Vasudev, R.: What is one hot encoding? Why and when do you have to use it? (2017) `https://medium.com/hackernoon/what-is-one-hot-encoding-why-and-when-do-you-have-to-use-it-e3c6186d008f`.
10. Dua, D., Graff, C.: UCI machine learning repository (2017) `http://archive.ics.uci.edu/ml`.