university of groningen / faculty of science and engineering

Machine learning systems rationale analysis

Bachelor's Project Thesis

Timo Wahl, s3812030, t.h.wahl@student.rug.nl
Daily Supervisor: C.C. Steging & Second Assessor: H.B. Verheij

Abstract: Bench-Capon (1993) showed that high accuracies on an open texture classification task does not necessarily mean that the reasoning of the model making the prediction is sound. In that paper it was shown that an in-depth analysis of the decision making process of an MLP can indicate the discrepancies in the models understanding, or rationale, of the domain that was trained on. This paper first replicated the original paper by Bench-Capon and then extended that paper with a more in-depth analysis of the domain, new machine learning systems and improved hyperparameter optimization. The MLP, Random Forest classifier and XGBoost classifier were all compared in their performance and rationales on the same domain. The conclusions from the paper by Bench-Capon were reaffirmed, with the primary conclusion from the extension being that the Random Forest and XGBoost classifiers are not capable of learning completely sound rationales for the welfare domain, despite being more optimized and being trained on larger datasets.

1 Introduction

1.1 Theoretical background

Neural networks, as explained in Zou, Han, and So (2008), are used for a wide array of applications in which they are often used as black box models that predict outcomes when given a set of input variables. In such black box systems, the information inside of the model, which its decisions are based on, is often unknown. The outcome and performance are frequently only of importance, whereas the decision making process behind a decision is not. This is especially the case in deep learning models. However in some fields the outcome actually has to be explained, such as when AI systems are used in law. The paper by Atkinson, Bench-Capon, and Bollegala (2020) gives an overview of explainable AI in law, concerning itself with identifying many different approaches towards explaining model outcomes. Explanation can be achieved by taking a look at the decision making process of a black box system to find out why a neural network is making decisions with regards to the output; the decision making process henceforth being called a rationale.

The paper by Bench-Capon (1993) assesses the soundness of the rationale given by a neural network when trained on an open texture domain. The neural network, specifically a multi-layer-perceptron (MLP), was trained and tested on a dataset which was based on six rules from the welfare domain, resulting in a boolean outcome variable. The welfare domain concerns itself with defining if a person visiting a patient is eligible for a welfare benefit. Specifically, the results were analyzed to see if the MLP was capable of forming a sound rationale. The rationale was extracted from the MLP through the use of graphs. Critically, the accuracies of the MLP were often high but Bench-Capon found that it was not capable of perfectly recognizing rules that consisted of non-straightforward combinations of factors. Which showed that the MLP was not capable of forming a sound rationale with regards to the defined open texture system. The paper concludes by warning against usage of neural networks which report high accuracies because it does not equal a sound rationale.

The primary concern of the machine learning models in this paper is the formation of sound rationales when learning for a classification task. In general a sound rationale consists of a model being capable of forming a sound decision making process with regards to the outcome produced. A decision making process can be analyzed in different ways, but will primarily be analyzed through graphs and accuracies in this paper. It is important to note that high accuracies do not equate to a sound rationale, as was found by Bench-Capon: A machine learning model can output the right results, but for the wrong reasons. In the case of the research by Bench-Capon a sound rationale for the entire welfare domain could only be formed when all rules were learned correctly by the MLP.

To formulate model decision making, Lundberg and Lee (2017) introduced SHapley Additive exPlanations (SHAP): a framework for interpreting the results of a machine learning system when trained on a learning task. In the paper it is first proposed that the model itself is always the best explanation for the results, but sometimes, such as in deep learning or in ensemble models, the models are too complex be used as the explanation. SHAP explains the outcome of a model by showing the different features and in how much they contribute to the outcome. It does this by forming a power set of the features that determine the outcome. All possible combinations of features are considered as models. The connections between the models, can give information about the marginal contributions of their features. For example by comparing a model with only one variable to a model that has that variable in combination with another variable. To form a final conclusion about the importance of a feature, all marginal contributions of a feature are combined and compared to that of a single marginal contribution. In doing so the prediction is compared to that of the null model, showing the effect of a specific feature compared to a baseline. SHAP attempts to solve the tension between accuracy and accountability by providing easy to understand explanations that are widely applicable in machine learning systems. SHAP is especially relevant to this paper as it has the same goal of explaining decision making in machine learning models.

Several others have replicated the paper by Bench-Capon in different settings. Možina, Žabkar, Bench-Capon, and Bratko (2005) attempted to find a sound rationale by use of inductive logic programming. Možina et al. used the same welfare domain as used by Bench-Capon which they used to show that their rationale found four of the six rules that defined the result of the outcome variable. Another approach was taken by Wardeh, Bench-Capon, and Coenen (2009), who employed a case based reasoning system to form a dialogue between two agents that would determine the final result and more importantly, which rules had an effect on the final result. Their system, called PADUA, was able to get an accuracy of over 90%.

1.2 Extended rationale analysis

This paper will extend the research by Bench-Capon by performing the same analysis but with two alternative machine learning systems: the Random Forest classifier and the XGBoost classifier. These two machine learning systems are chosen because they usually perform well in classifying decision processes and to analyze if this improved performance helps in forming a better explanation of the decision making process.

Random Forests (Breiman, 2001) consist of a number of decision trees (Kamiński, Jakubczyk, and Szufel, 2018) which are first formed randomly. The decision trees in the forest are trained by supervised bootstrapping; sub-sampling parts of the training data to decision trees. The training works well because in each decision tree, a node does not take the best predictor, but a random predictor, which establishes more diversity. After the initial set of trees is made and the model is trained, the Random Forest can predict a classification task by averaging the result of the different trees in the forest. The averaging is why it deals with the core features of the prediction task well. Random Forests can be used in a wide variety of learning settings, such as regression or classification. One of the distinguishing characteristics of a Random Forest is how well it adapts to new or unexpected data; the averaging causes new training data to have little effect which leads to robustness.

XGBoost (Chen and Guestrin, 2016) is a Random Forest based algorithm centered around extreme gradient boosting. The XGBoost models are trained by iteratively adding new trees to the forest with their leaves and nodes based on those of combined previous models. The new leaves and nodes are based on minimizing predicted residual error from those previous models. New models keep being added until there is no more improvement in the performance, resulting in a final model; with the new models minimizing the loss by using the gradient descent algorithm. Besides being very fast compared to other machine learning systems due to its distributed computing and parallelization, it is also widely applicable in many different classification tasks. This wide applicability is due to the scalability of the boosting of each model and its basis in supervised learning.

1.3 Research description

This paper will first replicate the analysis done by Bench-Capon to reaffirm the conclusions of that paper. The research by Bench-Capon will then be extended. The extension includes new specialized datasets, that can be used to further analyze separate rules in the welfare domain. The machine learning system used in that paper, the MLP, will also be improved by optimizing parameters through halving grid search. The MLPs performance will also be compared with two new machine learning systems: the Random Forest classifier and the XG-Boost classifier. Both of these new machine learning systems will also have improved parameter search. It is then particularly interesting to analyse the rationales of these different machine learning systems, which leads to the research question:
How do the rationales of different machine learning systems, with similar performance, compare in terms of soundness?

2 Methods

2.1 Welfare domain

For the replication of the paper by Bench-Capon, the artificial welfare domain is used. The domain is concerned with defining if a person is allowed to get a welfare benefit to pay for the trip of visiting a spouse in the hospital. There are six rules that defined the domain:

1. The person should be of pensionable age (60 for a woman, 65 for a man)
2. The person should have paid contributions in four out of the last five relevant contribution years
3. The person should be a spouse of the patient
4. The person should not be absent from the UK
5. The person should have capital resources not amounting to more than 3000 pounds
6. If the relative is an in-patient the hospital should be within a certain distance: if an out-patient, beyond that distance

It is important to note that some rules are more complex than simply stating a boolean condition. For example rule one changes the boolean condition based on the gender of the visitor. It is also possible to translate the rules for the welfare domain to simple predicate logic. Rules 1-6 can be found in Table 2.1 as rules R1(x)-R6(x). It should be noted that only the combination of all the rules being true leads to eligibility for a welfare benefit.

2.2 Dataset generation

Variable	**Type**	**Range**
Age	Numerical	0-100
Resource	Numerical	0-10000
Distance	Numerical	0-100
Contribution 1	Numerical	0-1
Contribution 2	Numerical	0-1
Contribution 3	Numerical	0-1
Contribution 4	Numerical	0-1
Contribution 5	Numerical	0-1
Residency	Boolean	Yes | No
Spouse	Boolean	Yes | No
Type	Boolean	In | Out
Gender	Boolean	Male | Female
Noise	Numerical	0-100

Table 2.2: Dataset generation definition.

Datasets are generated to provide training and testing data for the experiment. In total there are twelve variables resulting from the domain. Each rule in the welfare domain concerns itself with a number of boolean and/or numerical variables, a definition of all the variables in the domain is given in Table 2.2. Some things that should be noted

Eligible(x)	$\Leftrightarrow$ R1(x) $\wedge$ R2(x) $\wedge$ R3(x) $\wedge$ R4(x) $\wedge$ R5(x) $\wedge$ R6(x)
R1(x)	$\Leftrightarrow$ (Gender(x) = Female $\wedge$ Age(x) $\geq$ 60) $\vee$ (Gender(x) = Male $\wedge$ Age(x) $\geq$ 65)
R2(x)	$\Leftrightarrow$ | Con1(x), Con2(x), Con3(x), Con4(x), Con5(x) | $>$ 3
R3(x)	$\Leftrightarrow$ Spouse(x)
R4(x)	$\Leftrightarrow$ Residency(x)
R5(x)	$\Leftrightarrow$ Resource(x) $\leq$ 3000
R6(x)	$\Leftrightarrow$ (Type(x) = In $\wedge$ Distance(x) $\leq$ 50) $\vee$ (Type(x) = Out $\wedge$ Distance(x) $>$ 50)

Table 2.1: Rules for the welfare domain expressed logically.

when studying Table 2.2: the resource variable is always set in steps of ten and the noise variable is always included 52 times for each datapoint with each separate noise variable having a new randomly generated value. It should also be noted that the in or out patient distance threshold is set to 50, all numerical variables are always integers and any numerical variable has a randomly generated value based on a uniform distribution of the range. If that variable is supposed to fail, then the range is adapted and the value is also generated based on a uniform distribution over the failing range.

In general for the replication, training of the MLPs is done on datasets comprising 2400 datapoints with 50 percent eligibility, each datapoint having 64 variables; 52 noise variables, and the 12 domain variables. Testing is done on datasets comprising 2000 datapoints, with the same characteristics as the training datasets.

Four different types of datasets are generated in the replication to train the MLPs and test them. The standard dataset is the multiple fail dataset (A). In this dataset the datapoints would not be eligible due to multiple rules being false, instead of just one rule. This dataset is used because it is the most common version of the welfare domain, as most visitors will fail on multiple rules from the domain, not just one. The amount of rules that a datapoint fails on is randomly decided, with a minimum of 2 and a maximum of 6. The rules that are failed on are also randomly decided, with no rule being selected multiple times.

Another dataset that is used to test and train the MLPs is the single fail dataset (B), which was made in response to the multiple fail dataset. It was made in response to the multiple fail dataset, because that dataset got relatively high accuracies, as shown by Bench-Capon; the single fail dataset could give an indication as to if the MLP was capable of understanding all the rules of the domain separately. In the single fail dataset, the datapoints would not be eligible, but only because it failed on a single rule from the domain. This rule is randomly picked from the six rules, for each datapoint. Two more datasets were introduced as rationale evaluation datasets for the replication.

2.3 Rationale evaluation datasets

There are six different types of datasets that can be used to evaluate the rationale of the system. Two of those are used in the replication, with the other four being introduced in the extension. Specifically, datasets are generated in the replication to test rules 1 and 6 only, whereas in the extension the datasets to evaluate the remaining rules are added as well. These rationale evaluation datasets are introduced because they can tell a lot more about if a machine learning system is capable of learning a specific rule. If the machine learning system is capable of performing well on a rationale evaluation dataset, it is capable of learning that rule from the welfare domain. In the rationale evaluation datasets, the datapoints will only fail on the specified rule, with all the other rules evaluating to true. Which ensures that each rationale evaluation dataset tests a machine learning system's ability to learn only one specific rule, without being influenced by the other rules.

The two datasets used in the replication, the age/gender (1) and distance/type (6) datasets are designed to further analyze the inner workings of the MLPs. Datapoints in the age/gender dataset are designed to fail on only the first rule from the domain. In essence this means that the age of every datapoint is randomly generated to be a number between 0 and 100. It is important to note that this meant that the eligibility was only 37.5%, contrasting datasets A and B, which both had 50% eligibility rates. The resulting eligibility of 37.5% is due to the fact that only 40% of all females and 35% of all males will be eligible with the uniformly randomized age values; With there being a 50% split of males and females.

The distance/type dataset that is used to only test the MLPs is the second of the two rationale evaluation datasets from the replication. This dataset is set to only fail on the sixth rule from the domain, for every datapoint. The datapoint has a distance value randomly set between 0 and 100. The in or out patient distance threshold for rule six is defined to be 50, hence the eligibility for this type of dataset is 50%.

With the definition of the rationale evaluation datasets used in the replication, it is then possible to define all the different training and testing datasets used for the MLPs in the replication, as is visible in Table 2.3.

Name	Type	Size	Eligibility
A	Train	2400	50%
A	Test	2000	50%
B	Train	2400	50%
B	Test	2000	50%
1	Test	2000	37.5%
6	Test	2000	50%

Table 2.3: Dataset definition for the replication.

To extend the work of Bench-Capon, several new testing datasets are included. Rules 2-5 did not have rationale evaluation datasets in the original experiment, so new rationale evaluation datasets are generated to analyze those rules. All the extended datasets have an eligibility percentage of 50. The other 50 percent will fail on a specific rule from the welfare domain.

There are four new testing datasets introduced in the extension. For the contribution dataset (2), the datapoint will fail on two or more of the contribution variables. For the spouse dataset (3), the datapoint will fail on the spouse variable. For the residency dataset (4), the datapoint will fail on the residency variable. For the resource dataset (5), the datapoint will fail on the capital resource variable, with it being 3000 or lower.

In the replication, the training datasets have a size of 2400 datapoints and the testing datasets have a size of 2000 datapoints. For the extension the sizes of those datasets are increased. The training and testing datasets in the extension have a size of 50000 datapoints. For the extension it is then also possible to give an overview of the datasets that are used, as is visible in Table 2.4.

Name	Type	Size	Eligibility
A	Train	50000	50%
A	Test	50000	50%
B	Train	50000	50%
B	Test	50000	50%
1	Test	50000	37.5%
2	Test	50000	50%
3	Test	50000	50%
4	Test	50000	50%
5	Test	50000	50%
6	Test	50000	50%

Table 2.4: Dataset definition for the extension.

2.4 Multi-layer perceptrons

The neural networks used, as defined by Bench-Capon, are triangle structured MLPs. All MLPs consist of an input layer consisting of 64 nodes, an amount of hidden layers with a different number of nodes in each one and an output layer consisting of a single node, which signifies the boolean outcome of assessing the input variables, for a specific datapoint. There were three different types of MLPs used by Bench-Capon, the amount of nodes in each hidden layer for each MLP is specified as follows:

1. One hidden layer with 12 nodes
2. Two hidden layers with 24 and 6 nodes
3. Three hidden layers with 24, 10 and 3 nodes

Version 0.24.2 of the scikit-learn (Pedregosa et al. 2011) Python library is used to implement the MLPs for the replication. As it was not defined in the paper by Bench-Capon, the hyperparameters for the three different MLPs were found empirically. The hyperparameters for the three MLPs are the same: 3000 max iterations, a logistic activation, learning rate initialized to 0.001 and a batch size of 50. Any parameters that need to be set for a scikit-learn MLP but are not named, are kept at their defaults, which can be reviewed in the documentation.

2.5 Learning systems

Three different types of learning systems are trained on different training datasets, A or B, and then compared in their accuracies and the rationales that they form in determining a result for a testing dataset. The first type of learning systems is the same one as from the replication, the MLP. In the extension there will only be one version of the MLP used: the MLP with three layers. This MLP version is used because it had the best and most consistent performance over all the datasets. A halving grid search algorithm is used to find the optimal hyperparameters, within computational limits. Halving grid search entails that first all possible candidate sets of hyperparameters are evaluated with a low amount of resources and on each new iteration the best candidates get more resources The hyperparameters found for the MLP in the extension through the halving grid search algorithm are as follows: logistic activation, an alpha regularization value of 0.00008, a learning rate initialization of 0.008, a batch size of 26 and a maximum number of iterations of 3000. Any hyperparameters not named were kept at their defaults.

The second type of learning system is a Random Forest classifier. For this learning system the halving grid search algorithm is applied as well, to find the best hyperparameters. The resulting hyperparameters are as follows: 16 estimators, max depth of 19, 17 as the maximum number of leaf nodes, a minimum samples split of 6 and a random state of 0. Other hyperparameters are kept at their default values and are not computed due to irrelevance to the classification task or a lack of computational power.

The third type of learning system is an XGBoost classifier. The halving grid search algorithm is again applied to find the best hyperparameters. The resulting hyperparameters are as follows: 16 estimators, a max depth of 7, an objective based on minimizing the squared error, a learning rate of 0.25 and a gamma value of 0.5. Any hyperparameters not named were again kept at their default values due to the same reasons as for the Random Forest classifier. As opposed to the Random Forest classifier and the MLP classifier, the XGBoost classifier was not developed by scikit-learn so a separate library was used for it (Chen and Guestrin, 2016).

For all learning systems, the same type of preprocessing is used. The boolean variables are first converted to numerical variables and all the variables are then scaled using a min max scaler. The min max scaler scales every value in each datapoint in the dataset to a value between 0 and 1.

2.6 Experimental setup

Over both the replication and the extension, the machine learning systems are trained on datasets A and B. In the replication there are only four testing datasets, the same dataset types A and B, and two special rationale evaluation datasets: 1 and 6. In the extension four more rationale evaluation testing datasets are added: the datasets for rules 2-5. An overview of the scenarios can be found in Table 2.5.

Tested on	MF trained	SF trained
Multiple Fail	AA	BA
Single Fail	AB	BB
Age Gender	A1	B1
Contributions	A2	B2
Spouse	A3	B3
Residency	A4	B4
Resource	A5	B5
Patient Distance	A6	B6

Table 2.5: Scenario overview. The first letter is the dataset trained on, and the second letter/number is the dataset tested on.

For the results, in both the replication and the extension, all results are averaged over 100 iterations. On each iteration the models are retrained and tested on newly generated datasets.

	Model Trained on A			**Model Trained on B**		
Tested on	*MLP*	*Random Forest*	*XGBoost*	*MLP*	*Random Forest*	*XGBoost*
A	99.81 ± 0.11	98.32 ± 0.34	99.92 ± 0.02	99.62 ± 0.56	99.91 ± 0.05	99.97 ± 0.02
B	88.23 ± 2.54	72.80 ± 3.13	90.74 ± 0.15	98.72 ± 0.52	90.56 ± 0.55	91.53 ± 1.44
1	76.76 ± 6.79	58.71 ± 13.93	99.53 ± 0.61	99.17 ± 1.21	97.07 ± 2.46	99.99 ± 0.05
2	97.96 ± 1.56	82.51 ± 2.46	94.99 ± 0.69	99.64 ± 0.55	96.81 ± 0.59	97.59 ± 0.92
3	98.38 ± 2.77	62.09 ± 9.67	100.0 ± 0	99.64 ± 0.56	99.06 ± 2.59	99.99 ± 0
4	98.29 ± 2.68	99.94 ± 0.17	100.0 ± 0	99.63 ± 0.55	99.99 ± 0	99.99 ± 0
5	83.02 ± 4.89	77.07 ± 12.92	100.0 ± 0	98.21 ± 0.81	99.85 ± 0.59	99.99 ± 0
6	72.87 ± 4.97	50.02 ± 0.22	50.02 ± 0.22	97.28 ± 1.28	50.02 ± 0.22	51.86 ± 8.43

Table 3.2: Mean accuracies (%) for all learning systems across all scenarios of the extension.

3 Results

3.1 MLP Accuracies

In the replication, the performance of the three different MLPs is measured in the form of accuracies and graphs. The accuracies for all the scenarios used in the replication are visible in Table 3.1.

3.2 Rationale Evaluation

In the replication, scenarios 1 and 6 are analysed in more detail through the use of graphs. These graphs are important because they can be used to see if a rule that consists of a combination of factors can be learned by the MLPs. Figures 3.1a-3.1b depict the resulting graphs from scenario 1, while Figures 3.1c-3.1d depict the resulting graphs from scenario 6.

3.3 Extended Accuracies

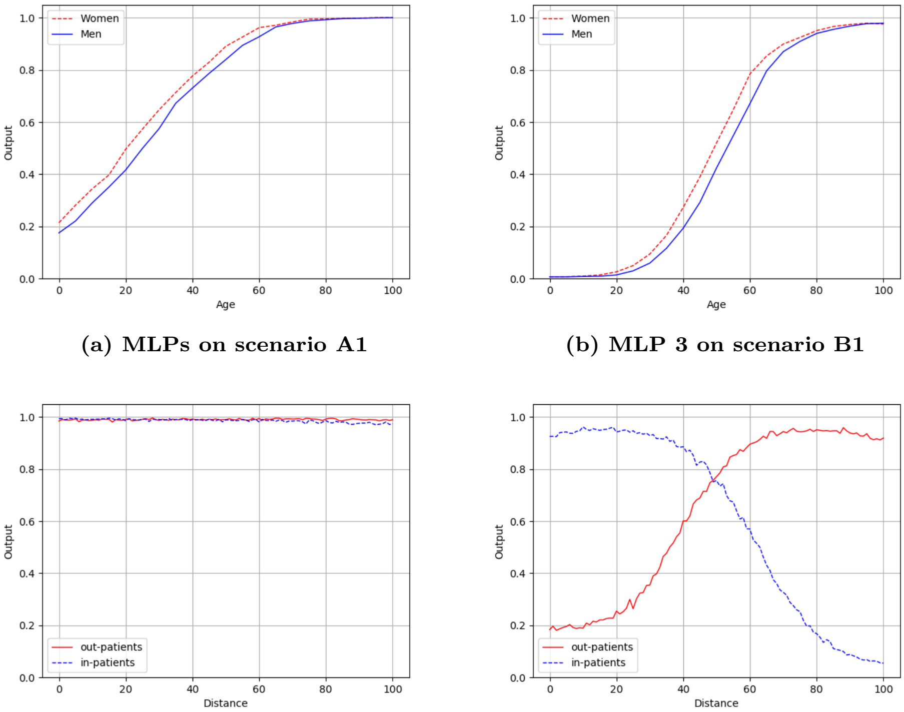
As was mentioned in section 2.3, the experiment by Bench-Capon is extended with four new testing datasets. The three layer MLP, with optimized parameters, the Random Forest classifier and the XGBoost all have their accuracies measured to assess their performance over the 16 different scenarios. For scenarios 1, 5 and 6 the results are also expressed as graphs to assess their respective rationales. The accuracies for these three learning systems are visible in Table 3.2.

3.4 Extended Rationale Evaluation

For the extension, scenarios 1, 5 and 6 are analyzed in more detail. Scenario 5 is added to see how the machine learning systems would perform when tested on learning a single numerical variable with a cutoff point, arguably an easier thing to learn than scenarios 1 and 6, but important nonetheless. Figure 3.2 shows all the resulting graphs when the models are trained on either dataset A or B and tested on the age/gender dataset. Figures 3.3 and 3.4 show the same but when the machine learning models are tested on the resource dataset and the distance/type dataset respectively.

Sc.	**MLP 1**	**MLP 2**	**MLP 3**
AA	98.33 ± 0.42	98.48 ± 0.39	98.14 ± 0.42
AB	76.67 ± 1.39	78.66 ± 1.45	77.87 ± 1.57
BA	96.32 ± 0.7	95.08 ± 5.22	94.23 ± 8.37
BB	91.00 ± 0.74	89.95 ± 0.82	84.17 ± 10.88
A1	61.30 ± 5.11	66.59 ± 5.39	63.29 ± 6.37
B1	86.35 ± 1.56	85.26 ± 1.46	80.44 ± 12.16
A6	50.35 ± 0.54	50.25 ± 0.29	50.16 ± 0.34
B6	85.67 ± 1.19	83.34 ± 2.73	67.58 ± 11.02

Table 3.1: Mean accuracies (%) for the three MLPs across all scenarios of the replication.



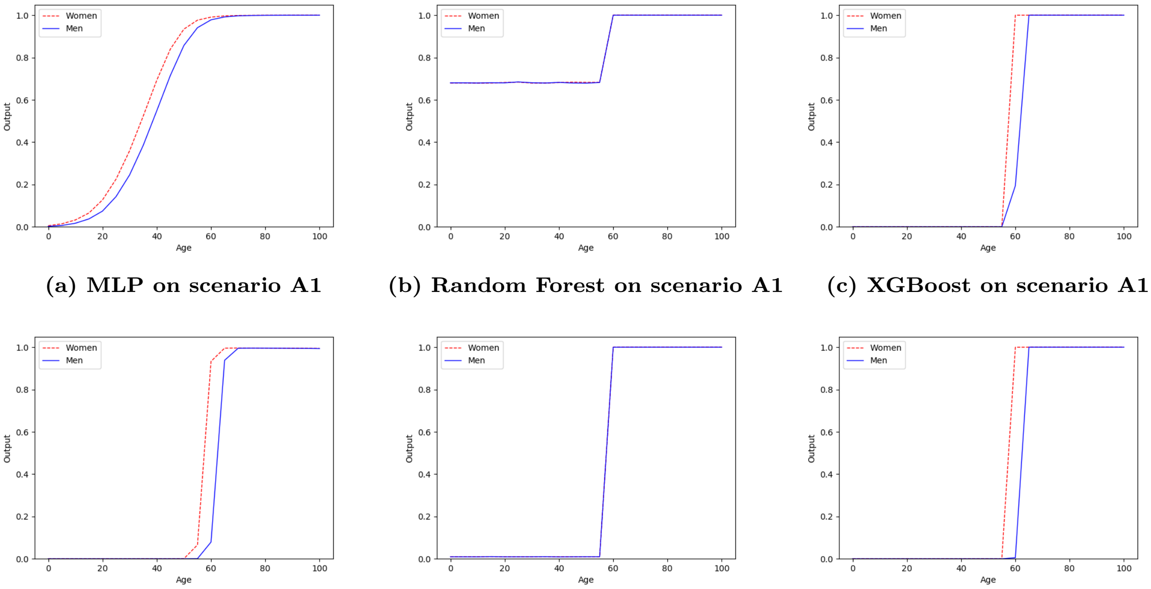
(a) MLPs on scenario A1

(b) MLP 3 on scenario B1

(c) MLPs on scenario A6

(d) MLPs on scenario B6

Figure 3.1: Replication graphs for rationale evaluation.



(a) MLP on scenario A1

(b) Random Forest on scenario A1

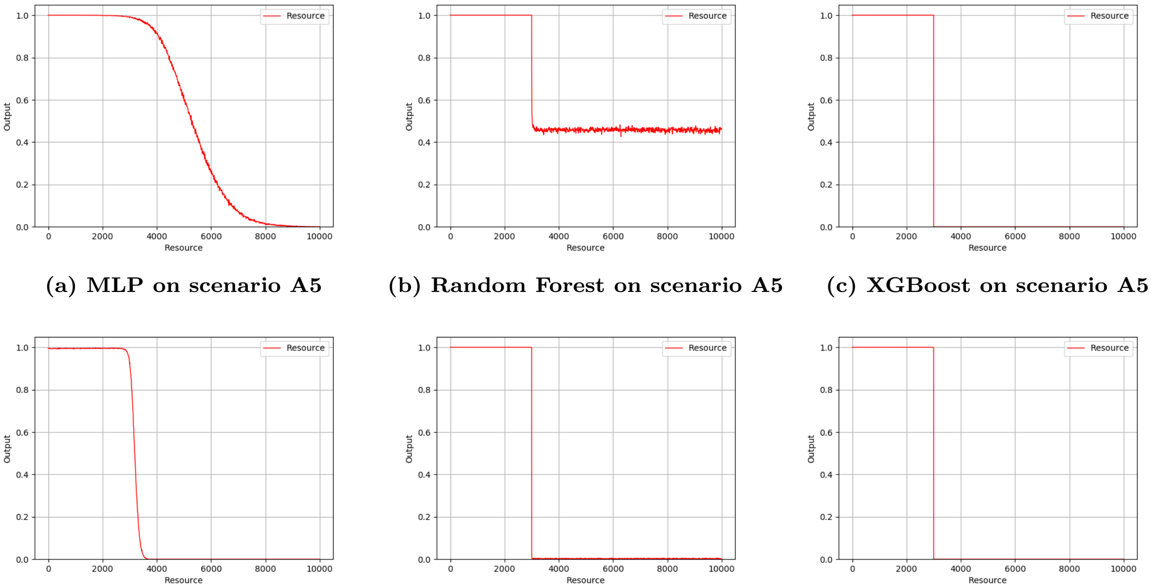
(c) XGBoost on scenario A1

(d) MLP on scenario B1

(e) Random Forest on scenario B1

(f) XGBoost on scenario B1

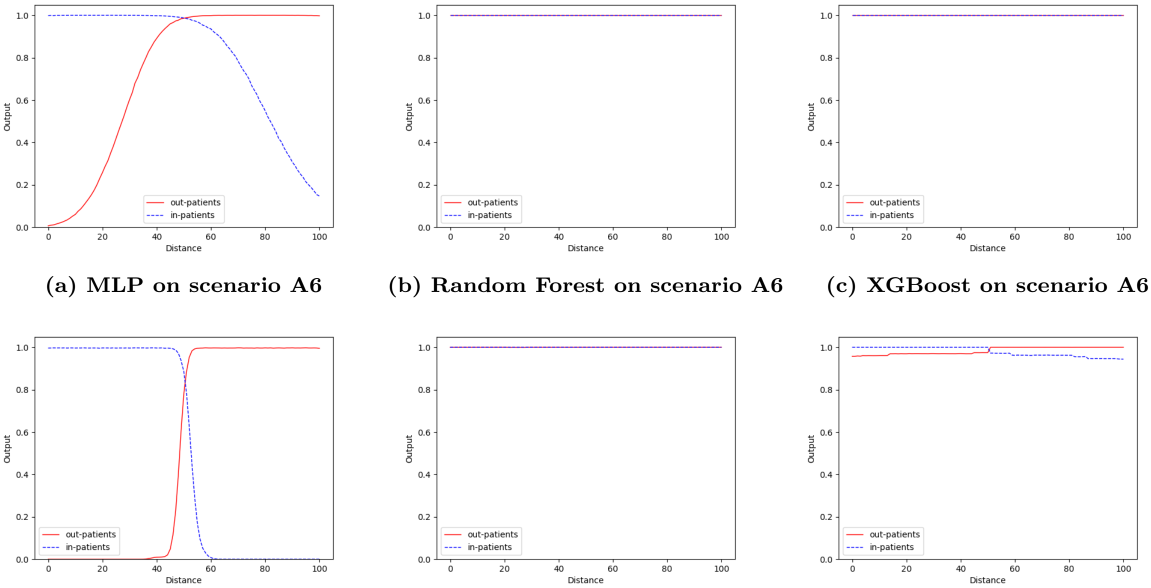
Figure 3.2: All extension machine learning performance graphs for scenario 1.



(a) MLP on scenario A5 (b) Random Forest on scenario A5 (c) XGBoost on scenario A5

(d) MLP on scenario B5 (e) Random Forest on scenario B5 (f) XGBoost on scenario B5

Figure 3.3: All extension machine learning performance graphs for scenario 5.



(a) MLP on scenario A6 (b) Random Forest on scenario A6 (c) XGBoost on scenario A6

(d) MLP on scenario B6 (e) Random Forest on scenario B6 (f) XGBoost on scenario B6

Figure 3.4: All extension machine learning performance graphs for scenario 6.

4 Discussion

4.1 Replication

The results from the replication are similar to those from the original paper by Bench-Capon, as shown in Table 3.1. It was reaffirmed that all three types of MLPs have high accuracies in scenario AA. However, the accuracies are much lower when those same MLPs, trained on dataset A, are tested on dataset B; scenario AB. This is exactly the same as in the original paper by Bench-Capon and serves to testify to the validity the replication. The high accuracies of scenario AA can be explained by the MLPs only having to know a subset of the rules to assess a datapoint correctly; the MLPs only learn a number of rules but learning those is enough to get the right output very often. This is due to the fact that if a datapoint is ineligible it is due to a combination of rules, e.g. for a datapoint to be ineligible only one of the six rules has to fail, some of which are easier to learn than others. Which entails that the MLP does not have to learn the complex rules. In scenario AB this is different, as the MLPs have a harder learning task, which is evident from the lower accuracies. The learning task is much more complicated because when training on dataset A, it can be difficult for the MLP to assess which rules the datapoint is failing on, due to the combination of rules that the datapoint fails on. Finally, scenarios BA and BB show that the MLPs are better capable of learning each individual rule when trained on dataset B. Opposed to dataset A, in dataset B it is much easier for the MLP to recognize each individual rule. This is because each datapoint only fails on one rule, leading to a much clearer division of the rules. Which is why the resulting accuracies for those two scenarios are higher.

The results for scenarios A1 and B1 are different when compared to those of the original paper by Bench-Capon. The graph in Figure 3.1b shows that for scenario B1 the three layer MLP is close to learning rule 1 correctly. However, despite relatively high accuracies in this scenario, as shown in Table 3.1, the lines are far from perfect. Ideally the line would be at 0, go up and stay at 1 at an age of 60 for women and 65 for men. However, the graphs show that the MLP has a hard time mimicking this. From Table 3.1 it can also be concluded that the MLPs have a hard time learning rule 1 when trained on dataset A. The graph in Figure 3.1a confirms this finding as the lines go up too early with a much too low gradient. It is important to note however, that both results of scenarios A1 and B1 seem to attempt to make a difference in output between men and women, with women having a higher chance of being eligible. Which would be a desired effect, considering the difference in eligibility age. However, in both scenarios the MLPs have a very hard time of learning this difference between men and women correctly.

The results for scenarios A6 and B6 are also different compared to those found by Bench-Capon; the accuracies in Table 3.1 and the resulting graphs in Figure 3.1d are less ideal compared to the original paper. The graph in Figure 3.1d shows that the MLPs make a solid attempt towards learning rule 6 correctly, however the resulting lines are far from perfect. Ideally the line for the in-patients would stay at 1 until the distance threshold of 50 is reached, at that point it should go down in a straight line and take on a value of 0 for the rest of the range. The line for the out-patients should ideally be exactly the opposite compared to that of the in-patients. Figure 3.1d shows that the line for both the in patients and the out patients begin to decrease/increase much too early and also much too slowly. It is also important to point out the different minimum values between the two lines, with the out-patients having a minimum of about 0.2 while the in patients have the almost correct minimum value of close to 0. An odd result, as it would be more logical for the two lines to mimic each other exactly. The accuracies in Table 3.1 show that for scenario A6 the accuracies are much too low for the MLP to have correctly learned the rule. The graph in Figure 3.1c shows why that is; the MLPs assume rule 6 to always be correct in scenario A6, hence causing the accuracy of 50 percent.

4.2 Testing on extended datasets

For the extension, the three layer MLP, as used by Bench-Capon is optimized with better hyperparameters, which results in better accuracies overall. When the accuracies in Table 3.2 are reviewed, it can be concluded that the MLP is not able to learn all the rules when trained on dataset A. Despite the optimized hyperparameters and a larger training dataset, the accuracies in Table 3.2 show that only some rules are learned almost perfectly. Specifically, rule 1, 5 and 6 are not learned correctly. Figure 3.2a shows this in more detail, the lines are far from ideal, specifically, the lines go up way too early; detailing the inability of the MLP to learn the rule correctly. Which is confirmed by Figures 3.3a and 3.4a. The MLP seems to attempt to learn a rule, but seems unable to with the given hyperparameters and training dataset: not forming a sound rationale. When reviewing Table 3.2, it can be concluded that the MLP is able to learn each rule quite well when trained on dataset B, as opposed to being trained on dataset A. The standard deviations are low as well, meaning that over the 100 iterations, the MLP is often able to find the rule. The graphs in Figures 3.2d and 3.4d confirm this finding as for both of them the lines are close to ideal. Figure 3.3d also shows a close to ideal line, the ideal line being 1 for a resource value from 0 to 3000 and 0 for a resource value beyond the 3000 threshold.

For the Random Forest classifier the resulting accuracies in Table 3.2 show a limited understanding of most of the rules, when trained on dataset A. The accuracies show that only rule 4 is learned nearly correctly. For all the other rules, the rule is either learned somewhat correctly, for example rules 2 and 5, or not at all, which includes the rules left over. Figure 3.2b shows that the Random Forest classifier is unable to perceive that the eligibility value can never be positive when the age value is lower than 60. Figure 3.3b shows a correct understanding of the rule, up until the threshold point of 3000, after which the output seems to converge to 0.5. It is important to note the high standard deviations on rules 1 and 5, which signify that the Random Forest classifier is sometimes able to get better and worse results, depending on differences in the training dataset. It is also important to note the differences in accuracies when comparing rules 3 and 4. They are logically the same rule, both with a boolean value that should be true for the datapoint to be eligible. However, the accuracies differ greatly, which is an unexpected result. When trained on dataset B, the Random Forest classifier performs a lot better. This is evident from the accuracies in Table 3.2. However, not all rules are learned well, for example rule 6 is not learned well at all. The graph for that scenario, in Figure 3.4e, shows that the rule is always assumed to be satisfied; which shows a complete misunderstanding of the rule. On the other hand, all the other rules are learned quite well when trained on dataset B.

For the XGBoost classifier the resulting accuracies in Table 3.2 show promising results when trained on the dataset A. Rule 2 is learned well, but somewhat worse than the other machine learning systems and the classifier is unable to learn rule 6 correctly, with the output being very far from the ideal line. Figure 3.4c shows an interesting result when compared to the result from the model trained on dataset B; The rule is not learned at all, the classifier is unable to see the pattern of the rule, with the output always set to 1. It is also important to note the relatively low standard deviations of the XGBoost classifier. Which means that in all scenarios, except A6, the XGBoost classifier is able to find the correct rule with a nearly equal accuracy over all the iterations. When trained on dataset B, the results are not very different compared to when the classifier is trained on dataset A. Only rule 6 is not learned well at all, while the other rules are learned close to perfection. The graph in Figure 3.4f shows that the classifier is making an attempt at learning the rule, but does not succeed in learning the rule perfectly. It can be seen that the lines are moving along the same trend as they are supposed to, as visible in Figure 3.4d, but to a much smaller degree and at a too high output value. This result could be explained by the XGBoost classifier very rarely making a much better attempt at learning rule 6, judging from the increased standard deviation. But since this is so rare, it has a small effect on the final accuracy value, considering the amount of iterations that were run.

4.3 Comparison between rationales extracted by the machine learning systems

The three different machine learning systems tested in the extension all form rationales for deciding on eligibility for the welfare domain. When comparing the rationales that are formed when the machine learning systems are trained on dataset A to those that are formed when they are trained on dataset B, the results are almost always better when they are trained on dataset B. This is an expected result, as the division of rules in dataset B allows the machine learning systems to learn the rules better, as opposed to the combination of rules in dataset A. However, it is important to note that in the case of the XGBoost classifier, this difference in rationale between training on dataset A and training on dataset B, is quite minimal compared to the Random Forest classifier and the MLP. This might be due to the underlying principles of the XGBoost classifier, causing it to learn the rules better. Specifically in comparison to the rationale extracted by the Random Forest classifier, when trained on dataset A, the effect of extreme gradient boosting combined with tree selection proves very effective.

Figures 3.2 - 3.4 are useful in determining the differences in the rationales extracted from the datasets. Figures 3.2d - 3.2f show the difference in the decision making regarding rule 1, when the machine learning models are trained on dataset B. Essentially, the MLP is able to infer a difference between male and female gender, whereas the Random Forest classifier is not. The XGBoost classifier performs better than the both of them by assessing a difference between male and female and also getting close to perfect threshold points. Which it is only approximately 5 years removed from for both male and female genders. From Figures 3.3d - 3.3f it can be seen that the difference between the three machine learning systems is less obvious in those graphs. All of them are able to get the threshold value, or at least very close to it; testifying to their ability of extracting a well formed rationale. However, only the MLP does not have the desired straight line, showing that in this case it does not learn the rule 5 completely correctly. Figures 3.3a - 3.3f show an entirely different story with regards to rationale extraction. The Random Forest and XGBoost classifiers are unable to extract a reasonable rationale. Only the MLP has a somewhat decent performance, and in doing so is able to get very close to a sound rationale for rule 6.

4.4 Weak performance on rule 6

In general, the performance on rule 6 is a lot worse than initially expected. Across all the machine learning systems, in both the replication and the extension, the accuracies for both scenarios A6 and B6 are lower than expected. It is possible that with a different setup of hyperparameters, the results for the different machine learning systems could be better on rule 6. This is a possibility because for the MLP it was found empirically that a lower alpha regularization parameter lead to better results. An accuracy value that was previously close to 50 percent would be improved to around 70 percent with the lowered alpha regularization hyperparameter, with both MLPs trained on dataset A. It is possible that the same could be true for both the Random Forest classifier and the XGBoost classifier, with a lower regularization parameter leading to the systems being able to learn rule 6 more easily, both when trained on dataset A and B. However, better performance does not necessarily mean that the machine learning systems would learn a sound rationale.

4.5 Comparison with replication

In comparing the replication to the extension, essentially the effects of optimization of the hyperparameters for the three layer MLP and the extension of dataset sizes is measured. One of the primary things that is important to note is that the accuracies are higher in general in the extension due to the larger datasets used, as is evident when comparing Table 3.1 to Table 3.2. When comparing Figure 3.2a to Figure 3.1a it also becomes clear that in the extension, the rationale extracted by the MLP is better, as the lines lie closer to the ideal line. When trained on dataset B the MLP also has a better rationale in the extension compared to the replication, as is visible when comparing Figure 3.2d to Figure 3.1b. When comparing the figures for rule 6, the conclusion is essentially the same, with the lines being closer to the ideal line for the extension when compared to the replication. Higher accuracies and better rationales prove the importance of dataset sizes and model hyperparameters.

4.6 Conclusion

This paper aims to analyze the rationales that different machine learning systems form when trained on the same domain. The paper by Bench-Capon has been replicated; the results are different in some cases, but the overall conclusions are the same. This paper introduces some extensions, which include larger datasets, new machine learning systems and improved hyperparameter optimization. Based on the results a few things can be concluded:

- Larger datasets and hyperparameter optimization lead to higher accuracies for the welfare benefit classification problem.
- The Random Forest classifier and the XGBoost classifier, like the MLP, both do not suffer from weaker performance when noise variables are included.
- The Random Forest classifier is not capable of forming a sound rationale for rules consisting of a combination of factors, such as rules 1 and 6 from the welfare domain. The XGBoost performs better, only being incapable of forming a sound rationale for rule 6.
- None of the three machine learning systems are capable of forming sound rationales for the welfare domain when trained on dataset A. But when trained on dataset B, the MLP comes quite close to a perfect rationale for the welfare domain.
- The three machine learning systems are capable of forming sound rationales for some of the separate rules from the welfare domain, when either trained on dataset A or B, but not all.

Code availability

The code that was written for both the replication and the extension is publicly available on GitHub. The repository can be found at the following link: `https://github.com/Sparvriend/Bachelor-Project-2021`.

References

K. Atkinson, T. Bench-Capon, and D. Bollegala. Explanation in ai and law: Past, present and future. *Artificial Intelligence*, page 103387, 2020.

T. Bench-Capon. Neural networks and open texture. In *Proceedings of the 4th international conference on Artificial intelligence and law*, pages 292–297, 1993.

L. Breiman. Random forests. *Machine learning*, 45 (1):5–32, 2001.

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

B. Kamiński, M. Jakubczyk, and P. Szufel. A framework for sensitivity analysis of decision trees. *Central European journal of operations research*, 26(1):135–159, 2018.

S. Lundberg and S. Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

M. Možina, J. Žabkar, T. Bench-Capon, and I. Bratko. Argument based machine learning applied to law. *Artificial Intelligence and Law*, 13 (1):53–73, 2005.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

M. Wardeh, T. Bench-Capon, and F. Coenen. Padua: a protocol for argumentation dialogue using association rules. *Artificial Intelligence and Law*, 17(3):183–215, 2009.

J. Zou, Y. Han, and S. So. Overview of artificial neural networks. *Artificial Neural Networks*, pages 14–22, 2008.