A Preliminary Look at Heuristic Analysis for Assessing Artificial Intelligence Explainability

KARA COMBS
Biomedical, Industrial & Human Factors Engineering
Wright State University
Dayton, OH USA

MARY FENDLEY
Biomedical, Industrial & Human Factors Engineering
Wright State University
Dayton, OH USA

TREVOR BIHL
Sensors Directorate
Air Force Research Laboratory
Wright Patterson AFB, OH USA

Abstract: - Artificial Intelligence and Machine Learning (AI/ML) models are increasingly criticized for their "black-box" nature. Therefore, eXplainable AI (XAI) approaches to extract human-interpretable decision processes from algorithms have been explored. However, XAI research lacks understanding of algorithmic explainability from a human factors' perspective. This paper presents a repeatable human factors heuristic analysis for XAI with a demonstration on four decision tree classifier algorithms.

Key-Words: - Artificial intelligence, classification and regression trees, decision trees, explainable AI, heuristic analysis, machine learning, XAI

Received: April 30, 2020. Revised: May 26, 2020. Accepted: May 30, 2020. Published: June 1, 2020.

1 Introduction

Analytics, machine learning (ML), and artificial intelligence (AI) have provided revolutionary advances in many disciplines, including aerospace, agriculture, business, cybersecurity, and the military. However, concerns exist when using AI solutions where decisions have impacts on lives, infrastructures, or processes with severe consequences; i.e. critical infrastructure applications. This concern is due to the complexity in AI decision-making processes which are often un-interpretable, unexplainable, or un-alignable from a human user perspective [1] [2].

In AI and automation, a user typically asks three questions when encountering unexpected or unexplainable results [3]:

- What is it doing?
- Why is it doing that?
- What will it do next?

Collectively, these questions get at a larger problem which precludes the wider adoption of AI algorithms, the failures of AI to address the "ilities" [4] [5] [6]: reliability, repeatability [7] [1], replicability [1], trust-ability [8], functionality [9], portability [9], usability [9], maintainability [9], and explainability [2]. Of particular interest in this paper, and to programs such as DARPA's eXplainable AI (XAI) project [2], is addressing the explainability of AI to reduce the opacity of algorithms and decrease the amount of unexpected or unexplained results [2].

While various approaches exist to "explain" an AI model [10], what is currently missing are approaches to evaluate the explainability of an AI model from a human factors' engineering perspective. The scope of this paper includes developing a repeatable approach for accessing the explainability of an algorithm. For this, the authors

leverage concepts from human factors studies, specifically heuristic analysis. To provide an example of the process, the authors consider decision trees as the AI/ML algorithm under investigation. Notably, decision trees already provide an explainable interface in how decisions are made in the form of a tree with explicit rules describing how data was divided. Thus, herein, the authors can focus on the mechanics of the XAI evaluation process and consider the relative differences between already explainable AI/ML approaches.

2 Background

The complexity in AI solutions introduces a challenge of opacity, which is then combined with the required "insider" knowledge to develop meaningful AI solutions. This leads to general misunderstandings and mistrust on the nature of AI algorithms [11]. This has brought about a vocal interest in explaining AI results and the development of XAI as a research domain.

2.1 Explainable AI

AI is typically used in the context of the pure machine power of computers to act "smart;" however, one off-shoot of this XAI. Notably, some contrast exists between explainable and accountable AI, c.f. [5] [12]. As noted in [13], not all AI solutions need to be explainable; for example, 1) trusted and trained algorithms are only useable within specified operating conditions and 2) well-studied conditions, and/or 3) no significant consequences for unacceptable results. Thus, accountable AI aims to develop trusted AI agents with known bounds of performance (akin to how decisions made by service dogs are unexplainable, but the dogs can be trusted) [5]. When human-interpretable understanding of black-box decision making is needed [12], XAI attempts to provide explainable interpretations for AI models throughout their operations

Prior work in the area of explaining complex algorithms includes neural network rule extraction methods, see [14], which began in the 1980s and 1990s by creating tree-based representations of neural network decision processes. However, as conceptualized in Figure 2, XAI extends beyond rule extraction approaches and includes human-computer interaction concerns as well as user-explanation concerns.

The result of an ideal XAI approach is conceptualized in Figure 3. Here, a contrast is presented between algorithms of today, which train an algorithm using data, outputs a declaration, and calculates the confidence on a new data sample's class. However, what is needed is an extension of this to provide human-interpretable knowledge structures from algorithms that further produce an understanding of the decision being made, i.e. this is a cat because it has whiskers, fur, cat-like ears, etc.

Fig. 1. General XAI goal compared with today's capabilities, adapted from [11]

While the general concept of XAI goals is seen in Figure 1, layers of explainability and infrastructure are needed to yield this. Work and reviews in XAI frequently mention human evaluation and interpretability criteria, c.f. [13] [15] [16], the approaches to doing so are less studied. Herein, we are not trying to provide the explainability, but rather to reduce opacity in algorithms by justifying the selection of algorithm A over B. The goal of this research endeavor is to include general software quality metrics for algorithms by understanding the quality of the interface, code, and results.

2.2 Academic Data for Consideration

Many data sets for classification or prediction (regression) tasks have data features that correspond to measurements. For this project, the Fischer Iris dataset [17] was used due to its simplicity and the general ability to understand its data features. Fisher Iris includes four data features: petal length, petal width, sepal length, and sepal width, with measures corresponding to three varieties of iris flowers: setosa, versicolor, virginica [17]. A total of 150 flowers, evenly distributed between the 3 varieties, were measured [17]. Fisher Iris, while small in nature, is one of the most used datasets for classification and was selected due to widespread familiarity with it in the machine learning community.

2.3 Classification and Regression Trees

Decision trees algorithms use various approaches to create a tree-like structure where data is divided and split into multiple categories based on other defining characteristics. [18]. Decision tree algorithms generally classify or regress (predict) depending on the scenario. The algorithm asks "questions" (as shown conceptually in Figure 2), which helps it divide each data point into predetermined groups based on other variables associated with that data point [18]. Colloquially,

such algorithms are often called "CART," for Classification and Regression Trees, and these algorithms differ based on their methodologies in splitting data and in stopping rules [18].

Fig. 2. Conceptualization of a decision tree with a classification objective of fruits based on different characteristics, adapted from Figure 8.2 of [18].

A wide variety of CART algorithms exists in literature. To provide a preliminary look at heuristic analysis for XAI, this study limited itself to four algorithms: ID3, C4.5, C&RT, and CHAID. These are briefly described as follows, with further details in [19], and the given algorithm references:

1. Iterative Dichotomiser 3 (ID3) - one of the first decision tree algorithms which computes the information gain associated with a data split and it continues splitting based on a splitting criteria based (information gain, entropy, or a Gini index) [20]
2. C4.5 (in the J48 Weka implementation of [22]) - an extension of the ID3 algorithm which uses a modified splitting criteria, a gain ratio of the ratio of information gain to intrinsic information, C4.5 further features pruning (removing unnecessary branches) and ability to account for missing data (as probability weights) [23]
3. Classification and Regression Trees (C&RT) – another early decision tree algorithm based on the Gini index impurity measure for splits, enables pruning and accounts for missing data (as surrogate splits on other values) [21]
4. Chi-squared Automatic Interaction Detection (CHAID) computes a Chi-squared test of association and then uses the p-values from the associated *F* statistics as the splitting criteria; CHAID does not support pruning. Missing values are imputed locally in a given split [24]

Briefly, these methods differ in both implementation, software, and metrics used to determine nodes and splitting. ID3 and J48 employ entropy and information gain [20] [22] [23], C&RT employs a Gini coefficient [20], and CHAID employs a Chi-squared test of association. Beyond these differences, the instantiations of the algorithms in software also provide for different native representations. For software implementation, ID3, C&RT, and CHAID were operated in Python (3.7.3) and J48 was operated in Weka (3.8). ID3 was taken from the sklearn and matplotlib libraries, C&RT was taken from the numpy, random, and csv libraries, and CHAID was taken from the sklearn, numpy, CHAID, and pandas libraries. Beyond these differences, the instantiations of the algorithms in software also provide for different native representations.

3 Heuristic Analysis

A heuristic evaluation is one of the tools most commonly employed by usability professionals [25]. While a heuristic evaluation won't uncover every usability problem, it is often quite effective when combined with other methods [26]. In a Heuristic Analysis, a human factors expert subjectively evaluates a system based on a set of predetermined criteria [27]. Measuring the "goodness" of a system is a non-empirical process than cannot be run through an algorithm that automatically "checks" the content, hence the need for heuristic evaluation [28]. The benefits of heuristic analysis include inexpensive cost, intuitiveness, no long-term planning required, and can be completed at any stage of the design process; disadvantages include the subjective-ness of the evaluation and the lack of suggestions to improve low-scoring usability-areas [28]. The goal of a heuristic analysis is to provide an evaluation of a system that identities all of the pros and cons based on a predetermined list of criteria [28]. Heuristic analysis was shown to provide the best insights at the lowest cost; however, it is essentially to have several qualified evaluators [29]. An initial set of usability principles was created [30], which developed into Nielsen's ten usability principles and have been widely adopted. Both the Gerhardt-Powals and Nielsen principles [30] [31] are effective in highlighting usability issues [32] [33], and have been expanded to effectively address new application domains [34]. Heuristic principles can also be modified [35] or used as the basis for a usability assessment scheme [26] which can be tailored to fit a particular need [36]. In this effort, each algorithm's output was evaluated according to a modified list from Nielsen's 10 Heuristic Principles [37] [38]. Nielsen's was chosen as the basis due to being well-established in the human factor's world and a common go-to for interface design. The list of tailored principles is defined in Table 1.

4 Decision Trees Applied to Fisher Iris

Applied in their native formats, the four algorithms present in Figures 3 – 6 are representations of decision trees for Fisher Iris. Due to its relatively small size (n = 150), the algorithms were able to

create each decision tree rather quickly as the set is simplistic in nature. When considering C&RT, in Python, we are presented with Figure 3. In Figure 3, each data feature, X1 to X4, is listed at a node along with the associated split. The inequality at the node indicates the value at which the data was split given a particular characteristic. This value is shown immediately after the inequality sign in the bracket a data feature, X1 or X4, appears.
The ID3 output, Figure 4, is presented as a multi-layer tree that shows the feature that is split at each node, the Gini coefficient associated with the data split, the number of samples in total, the value (the remaining samples for each class), and the class associated with each split. The value section shows three numbers within a set of brackets, referring to the distribution of samples by class. A total of five "layers" of the tree (noting that the top-most block is "Layer 0") are shown for this particular dataset as shown in Figure 4. Each node is shaded a different color, corresponding to the class and its concentration in that node.

Fig. 3. C&RT applied to Fisher Iris, with a minimum of 3 splits, where X1 is sepal length, X2 is sepal width, X3 is petal length, and X4 is petal width.

The output from the J48 algorithm is shown in Figure 5 and has been cropped from the full output of Weka. The full output showed detailed reports on running the algorithm, Figure 5 presents only the subset of the report that showed the "J48 pruned tree". The image in Figure 5 presents a text-based visual representation of the J48 tree embedded in the algorithm. The tree in Figure 5 clearly states the criterion for each "leaf," which also corresponds to its classification decision.

Fig. 4. ID3 applied to Fisher Iris

Fig. 5. J48 applied to Fisher Iris.

The raw CHAID output, Figure 6a, presents a more expansive display and seems to be organized by individual lines whose sections are divided by brackets. For clarity, part of the raw output is enlarged in Figure 6b. The lines show a value associated with a mean and standard deviation along with numerical groups. The algorithm's splits based on the sepal width and sepal length appear to be successful, but others were unsuccessful due to a violation of the minimum child or parent node criteria Some lines are succeeded by the text, "<Invalid Chaid Split>" indicating that further splitting would create nodes with less than the minimum child node size.

6a) Raw CHAID output

6b) Enlarged Subset
Fig. 6. CHAID applied to Fisher Iris, with a minimum of 3 splits, where X1 is sepal length, X2 is sepal width, X3 is petal length, and X4 is petal width

5 Heuristic Analysis Application
With an understanding of the XAI problem at hand, the heuristic analysis approach of Section III was applied to the C&RT results in Section IV from a human factors perspective.

5.1 Preliminary Study
To provide a preliminary look at using heuristic analysis for XAI, the first author applied the heuristic approach of Section III to subjectively analyze the results in Section IV. To evaluate the principles, for Phase 1 a binary decision (yes/no) was made and for Phase 2, a 5-point a Likert-scale is used with 1 being the worst and 5 being the best. For each principle, the algorithm is assumed to have 5 points, from which points will be deducted for each factor that decreases the quality of that particular output for which principle. Some values may be influenced by the evaluation of other algorithm outputs. The principles associated with the coding aspect will be evaluated at a binary level for all algorithms based on whether the principle is present or not present.

5.2 Heuristic Analysis of Algorithms
First, the algorithms were evaluated on the four principles associated with the code and documentation with a binary decision. If the statement applied to the algorithm, it was given a "yes" (positive) or a "no" (negative). This was based on the information receive prior to, while, or shortly after running the code. These results are summarized in Table 2. Regarding principle 3, all algorithms received a yes due to the user being able to change various parameters. Whereas some settings were particular to software instantiations, e.g. J48 and Weka, all algorithms were tailorable to some degree. On principle 5, only CHAID received a yes because of its GitHub source provided specific tips and error notices for setting up the algorithm whereas the other algorithms came built into their software and such notices were not readily apparent. Principle 9 was mostly not applicable or testable due

to the data being well structured and noting in the results that suggested there were errors when running the algorithm. However, CHAID receives a yes for principle 9 because it states if there was an attempted invalid CHAID split, i.e. if a minimum sample threshold was reached. On principle 10, only ID3 receives a no due to the lack of documentation for this instantiation of the algorithm whereas the other algorithms come with sufficient documentation.

Table 2. Phase 1 (Coding) Binary Assessment

Principle	Algorithm			
	ID3	***C&RT***	***J48***	***CHAID***
3	Yes	Yes	Yes	Yes
5	No	No	No	Yes
9	N/A	N/A	N/A	Yes
10	No	Yes	Yes	Yes

Table 3 presents results from the Phase 2 (Output) analysis. On principle 1, ID3 and J48 ranked highest because they both provided a clear understanding of the resultant tree and what occurred at each node and what is being presented. However, C&RT and CHAID performed poorly in this principle. C&RT performed poorly on principle 1 because only the feature and value associated with the split were presented whereas the other numbers were confusing; CHAID performed the worst due to its presentation, shown in Figure 6, which is extremely difficult to interpret and comprehend.

On the second principle, ID3 provided a decision tree with everything clearly marked and labeled. C&RT provided limited information and context, which caused it to perform poorly on principle 2. J48 provided a large amount of usable information but the text format of the presentation was confusing to interpret and does not match a typical mental model of a decision tree. CHAID performed poorly on principle 2 considering its output's long lines and a confusing organization of the data that does not resemble a decision tree initially.

When considering the fourth principle, ID3 and J48 perform highly because the layout is easy to understand and each node is presented consistently. C&RT also performs well in principle 4due to consistency, but it's unclear why certain sections have a number in brackets beneath it and other sections do not without searching through documentation. CHAID performs poorly on the 4th principle considering an overabundance of details that are provided for each node (the feature, statistical p-value, t-score, grouping values, and degrees of freedom), which is combined with inconsistent formatting to provide an appearance of many out-of-place details.

Looking at the sixth principle, most algorithms perform down the middle due to a general confusion on how values were computed (i.e. Gini in ID3), missing context in the numbers presented for C&RT, unknown conditions on what are "acceptable" values for J48, and a need to understand the context of various messages ("Invalid CHAID Split" for CHAID). Largely, this principle was difficult to judge because of a general lack of information to recognize and then, recall and an insufficient amount of context of the resulting information.

On the eighth principle, which relates to aesthetics and minimalist design, ID3 performs the highest due to its straightforward output. C&RT scores well due to its simple design, but it is notably too minimal to interpret. J48 performs well because it is easy to read, and the layout is aesthetic; however, a perfect score was not given due to the amount of data provided (which was cropped out of Figure 5). CHAID performs poorly again because, while the design of this output is rather plain, the way it is formatted and presented is not very aesthetic.

Principle 9 was again mostly not applicable or testable due to the data being well-structured and produced without errors. CHAID is the only algorithm that scores in this principle because it provides notices on splitting errors, but it does not provide any tips regarding how to fix those errors, assuming that is an actual error at all. One option to consider is that those notices are automatically built into the output as an explanation for why the tree didn't create another node and branch off.

On the additional principle of trustworthiness, ID3 scores in the middle because it seems to have reasonable results but doesn't provide an understanding of how it uses Gini. C&RT performs poorly because of a lack of context in its representation, i.e. the mystery numbers in square brackets. J48 scores highly due to the output presenting clear information which doesn't contradict any expectations. CHAID performs poorly because a user does not know how many data points were sorted at each node and considering the many aforementioned issues in the output's presentation.

Table 3. Phase 2 (Output) Qualitative Assessment. Values are Mean (N = 4) ± Standard Error

Principle	Algorithm

	ID3	*C&RT*	*J48*	*CHAID*
1	4.25 ±0.24	2.5 ±0.14	4.5 ±0.14	1.25 ±0.13
2	4.25 ±0.24	1.75 ±0.13	3.75 ±0.13	1 ±0
4	5 ±0	4.75 ±0.13	4.75 ±0.13	2.75 ±0.13
6	4.5 ±0.14	1.5 ±0.14	4.75 ±0.13	1.5 ±0.25
8	5 ±0	2.25 ±0.38	4.25 ±0.13	1.5 ±0.25
9	1 ±0 *	3 ±0.58	3 ±0.33	2.75 ±0.43
10	4.5 ±0.14	3.75 ±0.38	4.75 ±0.13	2.75 ±0.43
11	4.5 ±0.14	1.75 ±0.13	4.25 ±0.24	1 ±0

*For statistics calculations, these values had $N = 3$, one evaluator rated them as N/A

On the principle of user confusion, ID3 receives a 3 because while the graphic produces an understandable mental model of a decision tree, the uncertainty in the meaning of Gini invites confusion to inexperienced users. C&RT performs poorly as well due to missing details in how the splits occur, the number of samples at each split, and the context-less values presented. J48 scores highly on this measure since the user is presented with defined terms and statistics to draw conclusions from. Again, CHAID performs poorly because the output is very confusing to read, partially due to both the formatting and the way data is presented. In general, a user has no idea as to the classification aspect of CHAID.

5.3 Results Overview

Table 4 synthesizes the data found in Tables 2 and 3. Here, Phase 1 (coding aspects) are summarized by the total number of "yes" reports in Table 2, and Phase 2 results are presented as the means for each algorithm in Table 3. For the averages, the ID3, C&RT, and J48's total summations were divided by 7 due to Principle 9 not being applicable in their cases; CHAID's total summation was divided by 8. Overall, CHAID (or at least the implementation of CHAID considered) was the weakest of the group. This was followed by C&RT which had a much better interface than CHAID but lacked many vital details. However, both J48 and ID3 performed well and statistically the same in Phase 2. In general, if a user wants lots of details regarding their data set, ID3 could be sufficient; but J48 would be sufficient if the user doesn't need as much data and the assumption that the data set is rather small.

Table 4. Summary Results.

Principle	**Algorithm**			
	ID3	***C&RT***	***J48***	***CHAID***
Phase 1 Summary (number of yes's)	1*	2*	2*	4*
Phase 2 Summary (mean ± standard error)	4.23 ± 1.3	2.65 ± 1.13	4.29 ± 0.61	1.81 ± 0.8
Overall Ranking	1	3	1	4

*Only one participant, $N = 1$, evaluated Phase 1 conditions

6 Conclusions

This paper introduces the concept of heuristic analysis for assessing the understandability of eXplainable Artificial Intelligence (XAI) algorithms. XAI is a growing concern in AI and the general application of machine learning (ML) algorithms to a given task. XAI aims to address general criticism of AI in providing accurate, but not human-understandable logic in algorithmic decision making. In addition to this general XAI problem, a further issue exists in assessing and comparing the human interpretability of an XAI solution and this was primarily addressed by the authors.

The primary contribution of this paper is codifying and extending human factors' heuristic approaches to qualitatively and quantitatively assess the explainability of an AI algorithm and its results. As a demonstration of the developed heuristic approach, the authors applied this method to four decision tree classifiers. Decision tree classifiers were notably selected because they are intrinsically XAI in nature. To explore the assessment of XAI from a human factors' perspective, four decision tree algorithms were considered: J48, ID3, C&RT, and CHAID. These were each applied to a simple classification task and both their results (decision tree representation) and algorithmic coding were evaluated along 11 heuristic principles. These algorithms were considered in their native environment and thus, the differences in decisions being made were not assessed. The general conclusion is that details and cleanly designed representations are preferred along with simple representations that provide enough details to the user to interpret and understand, over a graphical result. Immediate future work would involve assessing identical-looking decision trees from these algorithms to assess the XAI aspects of the decisions being made. Further research involves XAI heuristic comparison of algorithms using identical visual representation of results to remove the coding approaches as factors. Additional extensions include evaluating multiple, different AI algorithms.

7 Acknowledgments

The views expressed herein are those of the authors and do not represent any position or view of the United States government, Department of Defense, or US Air Force. This work is cleared for unlimited release under case: 88ABW-2020-1624.

References:

[1] C. Drummond, "Replicability is not reproducibility: nor is it good science.," *International Conference on Machine Learning (ICML),* 2009.

[2] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced Research Projects Agency (DARPA),* 2017.

[3] D. D. Woods, "Decomposing automation: Apparent simplicity, real complexity," in *Automation and human performance: Theory and applications*, 1996, pp. 3-17.

[4] R. Potember, Perspectives on Research in Artificial Intelligence and Artificial General Intelligence Relevant to DoD, McLean, VA: JASON, The MITRE Corporation, 2017.

[5] T. J. Bihl and M. Talbert, "Analytics for Autonomous C4ISR within e-Government: a Research Agenda," *Hawaii International Conference on System Sciences (HICSS),* pp. 2218-2227, 2020.

[6] T. Bihl, J. Schoenbeck, D. Steeneck and J. Jordan, "Easy and Efficient Hyperparameter Optimization to Address Some Artificial Intelligence "ilities"," *Proceedings of the 53rd Hawaii International Conference on System Sciences (HICSS),* pp. 943-952, 2020.

[7] G. Zhang, "Avoiding pitfalls in neural network research," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 37, no. 1, pp. 3-16, 2007.

[8] A. Shadowen, Ethics and Bias in Machine Learning: A Technical Study of What Makes Us "Good", MS Thesis: CUNY John Jay College of Criminal Justice, 2017.

[9] A. Abran, A. Khelifi, W. Suryn and A. Seffah, "Consolidating the ISO usability models," *International Software Quality Management Conference ,* pp. 23-25, 2003.

[10] A. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins and R. Chatila, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion,* vol. 58, pp. 82-115, 2020.

[11] D. Gunning, "Explainable Artificial Intelligence (XAI), Proposers Day Slides," DARPA, Washington, DC, 2016.

[12] D. Dolk, D. Kridel, J. Dineen and D. Castillo, "Model Interpretation and Explainability towards Creating Transparency in Prediction Models," *Proceedings of the 53rd Hawaii International Conference on System Sciences (HICSS),* pp. 964-973, 2020.

[13] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608,* 2017.

[14] R. Andrews, J. Diederich and A. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowledge-based systems,* vol. 8, no. 6, pp. 373-389, 1995.

[15] L. Gilpin, D. Bau, B. Yuan, A. Bajwa, M. Specter and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," *IEEE 5th International Conference on data science and advanced analytics (DSAA),* pp. 80-89, 2018.

[16] W. Samek, T. Wiegand and K. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *arXiv preprint arXiv:1708.08296.,* 2017.

[17] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics,* vol. 7, no. 2, pp. 179-188, 1936.

[18] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, 2nd ed., John Wiley & Sons, 2001.

[19] W. Y. Loh, Classification and regression trees, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol. 1, no. 1, pp. 14-23, 2011.

[20] J. Quinlan, "Induction of decision trees," *Machine learning,* vol. 1, no. 1, pp. 81-106, 1986.

[21] D. Coppersmith, S. Hong and J. Hosking, "Partitioning nominal attributes in decision trees," *Data Mining and Knowledge Discovery,* vol. 3, no. 2, pp. 197-217, 1999.

[22] J. Quinlan, C4.5 : programs for machine learning, San Francisco: Morgan Kaufmann, 1993.

[23] I. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes and S. Cunningham, "Weka: Practical machine learning tools and techniques with Java implementations.," The University of Waikato, Hamilton, New Zealand, 1999.

[24] G. V. Kass, "An Exploratory Technique for Investigating Large Quantities of Categorical Data," *Applied Statistics,* vol. 29, no. 2, pp. 119-127, 1980.

[25] C. M. Barnum, Usability testing essentials: Read, set -- test!, Burlington, Massachusetts: Morgan Kaufmann, 2011.

[26] J. Horsky, K. McColgan, A. J. Melnikas, J. A. Linder, J. L. Schnipper and B. Middleton, "Complementary Methods of System Usability Evaluation: Surveys and Observations During Software Design and Development Cycles," *Journal of Biomedical Informatics,* vol. 43, no. 5, pp. 782-790, 2010.

[27] N. Stanton, P. Salmon, L. Rafferty, G. Walker, C. Baber and D. Jenkins, Human factors methods: a practical guide for engineering and design, CRC Press, 2017.

[28] J. Nielsen and R. Molich, "Heuristic Evaluation of user interfaces," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90),* pp. 249-256, 1990.

[29] R. Jeffries, J. R. Miller, C. Wharton and K. M. Uyeda, "User interface evaluation in the real world: A comparsion of four techniques," *Proceedings of the SIGCHI Conference on Human Factors in Computing System ,* pp. 119-124, 1991.

[30] R. Molich and J. Nielsen, "Improving a Human-computer Dialogue," *Communications of the ACM,* vol. 33, no. 3, pp. 338-348, 1990.

[31] J. Gerhardt-Powals, "Cognitive Engineering Principles for Enhaving Human-computer Performance," *Plastics, Rubber and Composites Processing and Applications,* vol. 8, no. 2, pp. 189-211, 1996.

[32] E. T. Hvannber, E. L. Law and M. K. Larusdottir, "Heuristic Evaluation: Comparing Ways of Finding and Reporting Usability Problems.," *Interacting with Computers,* vol. 19, no. 2, pp. 225-240, 2006.

[33] A. Karahoca, E. Bayraktar, E. Tatoglu and D. Karahoca, "Information System Design for a Hospitcal Emergency Department: A Usability Analysis of Software Prorotypes," *Journal of Biomedical Informatics,* vol. 43, no. 2, pp. 224-232, 2010.

[34] A. Alsumait and A. Al-Osaimi, "Usability Heuristics Evaluation for Child e-Learning Applications," *Journal of Software,* vol. 5, no. 6, pp. 654-661, 2010.

[35] A. Sivaji, A. Abdullah and A. G. Downe, "Usability Testing Methodology: Effectiveness of Heuristic Evaluation in E-government Website Development," *Asia Modeling Symposium,* pp. 68-72, 2011.

[36] E. De Kock, J. Van Biljon and M. Pretorius, "Usability Evaluation Methods: Mind the Gaps," *ACM International Conference Proceeding Series,* pp. 122-131, 2009.

[37] J. Nielsen, "Heuristic evaluation," in *Usability Inspection Methods*, New York, NY., John Wiley & Sons, 1994, pp. 25-62.

[38] J. Nielsen, "Enhancing the explanatory power of usability heuristics," *Proceedings of the SIGCHI conference on Human Factors in Computing Systems,* pp. 152-158, 1994.

Appendix

Table 1. Heuristic Principles

Heuristic Principle	Phase 1 – Coding Aspect	Phase 2 – Output Result	***Description***	***Heuristic Principle***	Phase 1 – Coding Aspect	Phase 2 – Output Result	***Description***
Visibility of system status (1)		**X**	Overall, a general evaluation of the algorithm. Can the viewer comprehend what the output is?	**Aesthetic and minimalist design (7)**		**X**	Are the necessary details included to determine the result without excess information? Does everything look pleasing or is it unorganized?
Match between system and real world (2)		**X**	Is the output similar to the user's mental model or are there differences?	**Help users recognize, diagnose, and recover from errors (8)**	**X**	**X**	Offers assistance if an input or section of code is incorrect within the shell as a result of the output. The principle focuses on the idea of how to fix errors once they occurred, instead of prevention as in principle 5. If there is an error in the output, does it attempt to offer solutions?
User control and freedom (3)	**X**		When running the algorithm, how much can the output or results be manipulated based on user input?	**Help and documentation (9)**	**X**		If errors occur, are there instructions (guide or FAQs) on how to reduce the errors or fix the errors?
Consistency and standards (4)		**X**	Is everything laid out in the same manner? Does anything appear "out of place?"	**Trustworthiness (10)**		**X**	Given the algorithm results, do they look reliable? Are there any contradictions? Are there any elements that causes the user to question the results? 1 – Not trustworthy at all 5 – Very trustworthy
Error Prevention (5)	**X**		While the algorithm is running or to be run, are there any tips as to how to prevent (potential) errors?	**User Confusion (11)**		**X**	Does the user feel comfortable explaining the results with confidence? 1 – Very confusing 5 – Not confusing at all
Recognition rather than recall (6)		**X**	Is everything in the output well-explained? Will the user need to research any statistics or rely on their memory?				

Figures 1 - 6

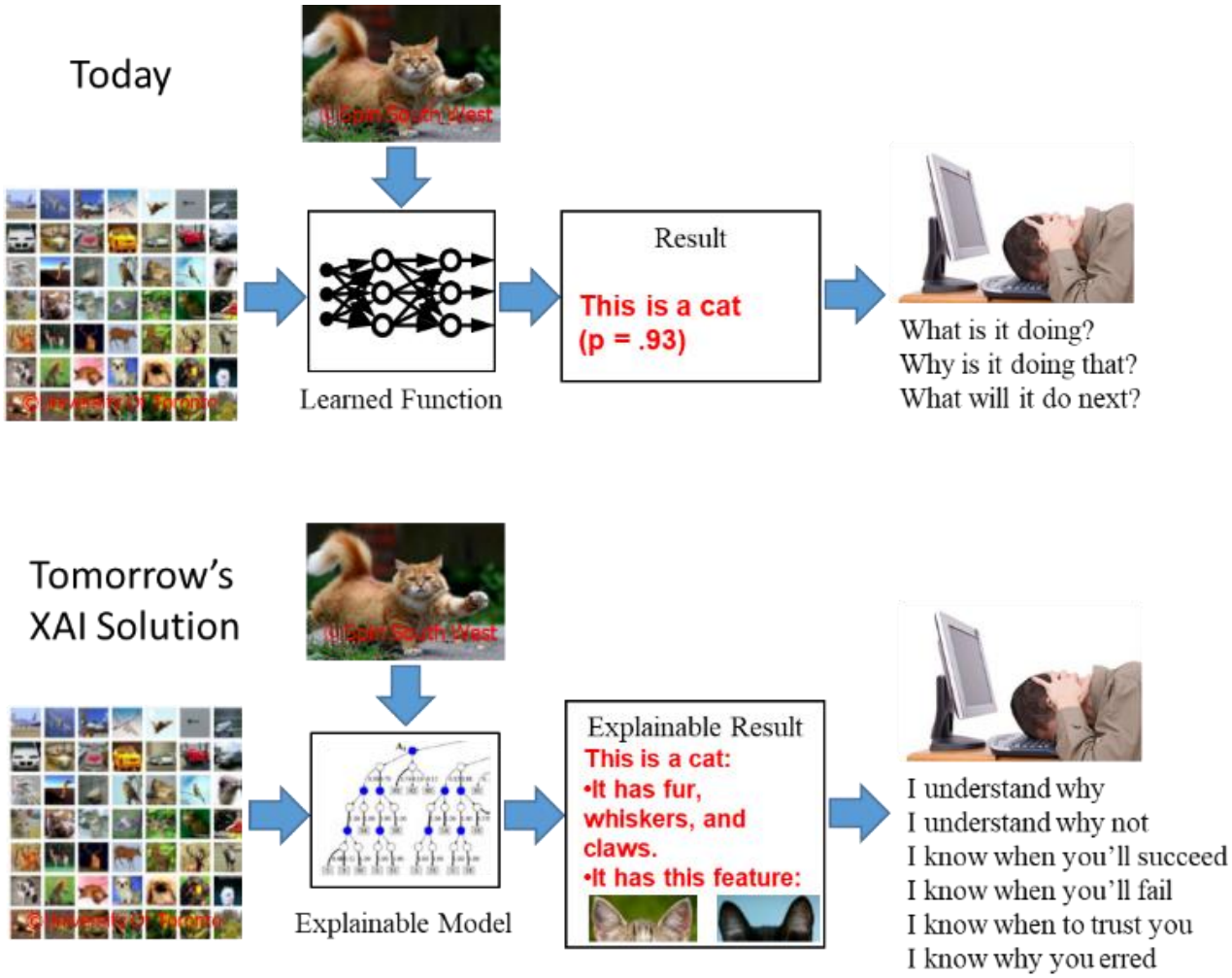


Fig. 1. General XAI goal compared with today's capabilities, adapted from [11]

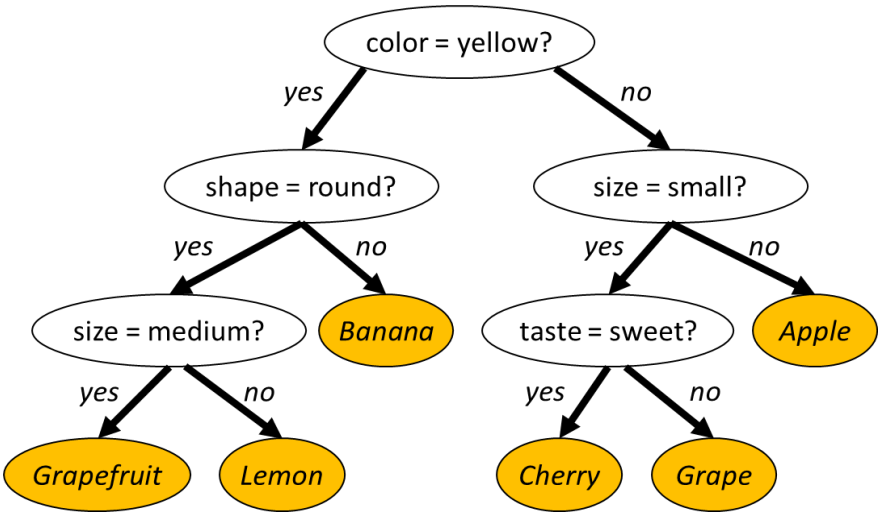


Fig. 2. Conceptualization of a decision tree with a classification objective of fruits based on different characteristics, adapted from Figure 8.2 of [18].

```
Split: [X3 < 3.000]
[X3 < 3.000]
 [X2 < 3.200]
  [X2 < 2.900]
   [0.3]
   [0.2]
  [X2 < 3.800]
   [0.2]
   [0.4]
 [X3 < 4.800]
  [X3 < 4.400]
   [1.3]
   [1.5]
  [X3 < 5.000]
   [1.8]
   [2.3]
```

Fig. 3. C&RT applied to Fisher Iris, with a minimum of 3 splits, where X1 is sepal length, X2 is sepal width, X3 is petal length, and X4 is petal width.

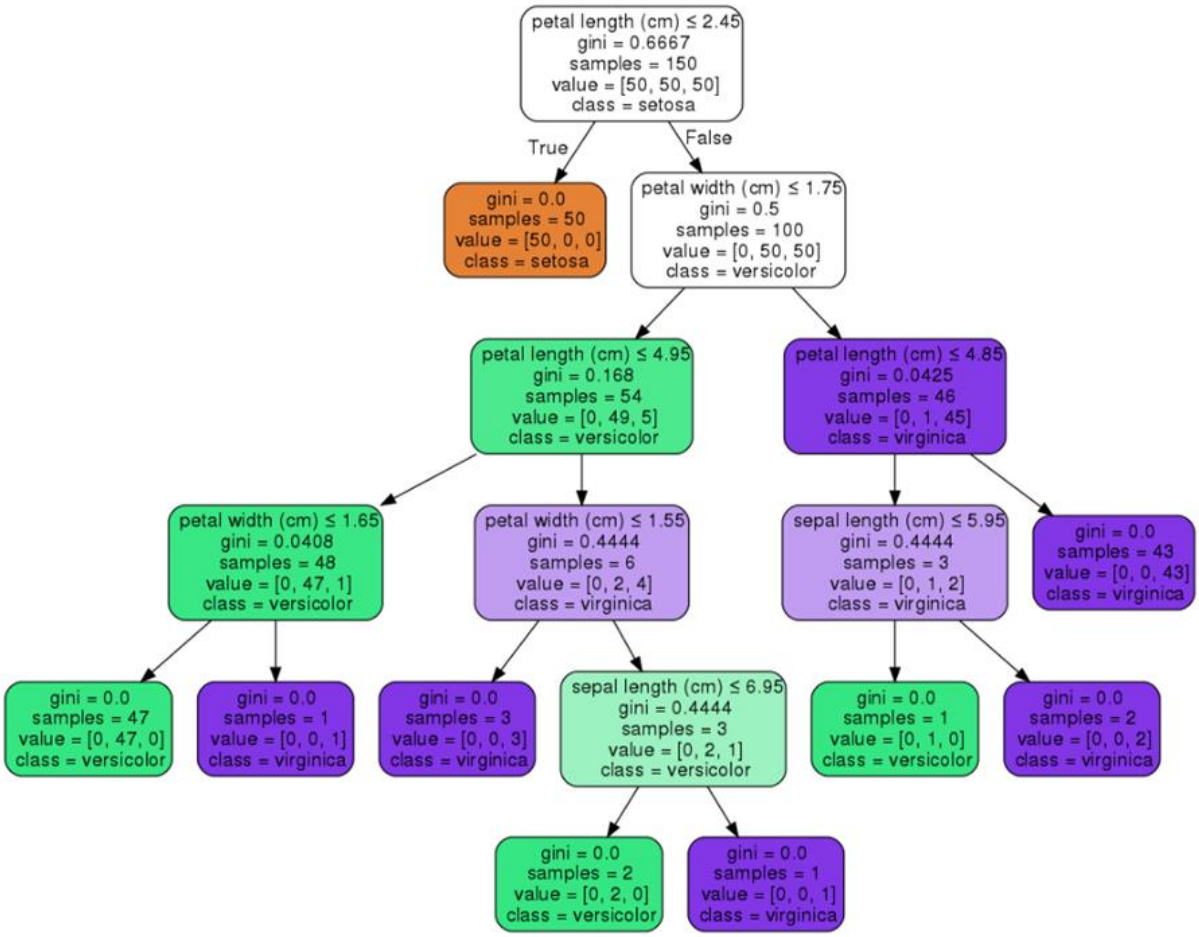
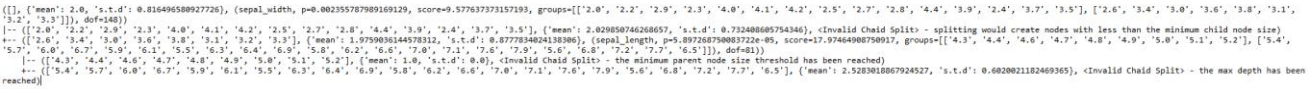


Fig. 4. ID3 applied to Fisher Iris

```
J48 pruned tree
------------------

petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (46.0/1.0)
```

Fig. 5. J48 applied to Fisher Iris.



a) Raw CHAID output

```
([], {'mean': 2.0, 's.t.d': 0.816496580927726}, (sepal_width, p=0.002355787989169129, score=9.577637373157193, groups=[['2.0', '2.2', '2.9', '2.3', '4.0', '4.1', '4.2', '2.5', '2.7', '2.8', '4.4', '3.9', '2.4', '3.7', '3.5'], ['2.6', '3.4', '3.0', '3.6', '3.8', '3.1',
'3.2', '3.3']]), dof=148))
|-- (['2.0', '2.2', '2.9', '2.3', '4.0', '4.1', '4.2', '2.5', '2.7', '2.8', '4.4', '3.9', '2.4', '3.7', '3.5'], {'mean':
+-- (['2.6', '3.4', '3.0', '3.6', '3.8', '3.1', '3.2', '3.3'], {'mean': 1.9759036144578312, 's.t.d': 0.8777834024138306}
'5.7', '6.0', '6.7', '5.9', '6.1', '5.5', '6.3', '6.4', '6.9', '5.8', '6.2', '6.6', '7.0', '7.1', '7.6', '7.9', '5.6', '6.8', '7.2', '7.7', '6.5']]), dof=81))
    |-- (['4.3', '4.4', '4.6', '4.7', '4.8', '4.9', '5.0', '5.1', '5.2'], {'mean': 1.0, 's.t.d': 0.0}, <Invalid Chaid Split>
    +-- (['5.4', '5.7', '6.0', '6.7', '5.9', '6.1', '5.5', '6.3', '6.4', '6.9', '5.8', '6.2', '6.6', '7.0', '7.1', '7.6', '7.9', '5.6', '6.8', '7.2', '7.7', '6.5'], {'mean': 2.5283018867924527, 's.t.d': 0.6020021182469365}, <Invalid Chaid Split>
reached)
```

b) Enlarged Subset

Fig. 6. CHAID applied to Fisher Iris, with a minimum of 3 splits, where X1 is sepal length, X2 is sepal width, X3 is petal length, and X4 is petal width