Implementation of an Explainable AI classifier for Heart Disease Detection

Saad Ejaz
School of Mechanical and Manufacturing Engineering
National University of Sciences and Technology, H-12
Islamabad, Pakistan
ejazsaad0332@gmail.com

***Abstract*—The exponential growth of AI does come with a downside. Increasingly complex algorithms demonstrate superior performance, but do not provide an explanation to their predictions. Explainable AI caters to bridge this knowledge gap so that users from a variety of sensitive fields can trust AI predictions. This paper mainly focuses on implementing the proposed probabilistic method by *Fan et al.* on a sensitive dataset aimed at predicting the presence of heart disease. The algorithms involved in the method are implemented in Python and Gurobi is used as the linear solver. Performance and explainability, both are analyzed, and results visualized. The results conclude age to be the most decisive factor in predicting heart diseases; however, this provides little information without interference from a medical professional.**

Keywords—Explainable AI, probabilistic methods, heart disease, medicine

I. Introduction

With the rapid growth of research in Machine Learning (ML), especially in Deep Learning (DL) with millions of hidden parameters, prediction performance is increasing regardless of the complexity of the problem. However, with this advancement, the reason behind a specific prediction is becoming increasingly unpredictable. This has come to a point where one cannot positively trust a prediction [1]. While this may not be significant for most AI problems where good accuracy is success enough [2], other industries and professions (such as medicine, risk assessment, financial analysis, etc.) might want to associate a trust factor with their predictions. Explainable AI is a means to determine what features (involved in a model) are responsible for a specific prediction [3]. While some traditional models are inherently interpretable (such as k-nearest neighbors, decision trees, and Naïve Bayes classifiers), these models either perform poorly, or do not present a complete explanation. These are compared with more model agnostic approaches where the model that explains classification is detached from the model that classifies. The classifier implemented in this paper is one that is proposed by *Fan et al.* in 2020 [4]. It uses probabilistic logic inference to *both* explain and classify predictions. The basic structure of the method proposed in [4] is revisited. The method is implemented in Python on a dataset from Kaggle[1]. The details of the dataset succeed the method summary. Moreover, the results from the implementation are compared with another probabilistic method (perhaps the simplest one: Naïve Bayes Classifier) and several other methods (Decision Trees, Random Forest, and Artificial Neural Networks). The procedure and results of this exercise are mentioned later in the paper.

A. Method Summary

Knowledge base is at the heart of this method. A knowledge base is defined as a list of clauses, with each having a probability measure, indicating the chance of occurrence of that clause. Such logical statements have been used prior to [4], but never in conjunction with probabilities. The method proposes algorithms and intuition for the following steps:

1) Constructing a knowledge base: Reference [4] proposes two distinct method of constructing a knowledge base. The first uses a tree method where a decision tree is traversed to generate clauses, and counts are stored for probability calculation. The knowledge base generated through this method is denoted by $\mathcal{K}_T$. The second methods employs is know as the direct method and employs a more procedural algorithm to construct all possible clauses and their probabilities. The knowledge base generated from the direct method is denoted by $\mathcal{K}_D$. The direct method gives a more decent performance that the tree method, since the direct method allows for a richer knowledge base. This is because $\mathcal{K}_\mathrm{D} \subseteq \mathcal{K}_T$.

2) Querying a knowledge base (Classification): A linear system of constraints and objectives is set based on previous work. Such a system was not new to the research literature, however, it had a very high computational complexity that exponentially grew with increase in features and data. The system construction proposed by [4] relaxes some of these constraints to allow a linear solver to compute probabilities in polynomial time. To further reduce computation time, an algorithm was suggested that extracts clauses relevant to the query only, before being fed to the linear program. Solving the system results in probability of the query being positive, hence classifying the query.

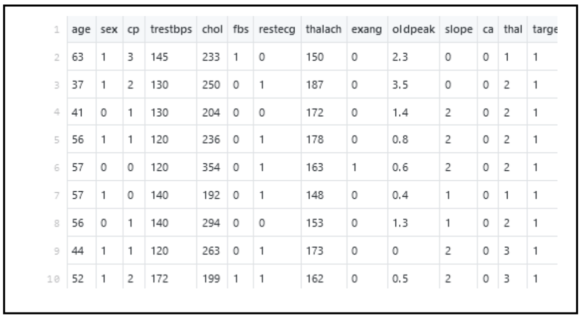
3) Explanation: For each query, one can iterate sub-queries over the program to select those that yield probability closest to the ground truth (0 for negative and 1 for positive). This way, one can identify which path/clause produces the most decisive result, concluding that features in that clause are the decisive features (that *explain* the prediction of the said query).

B. Dataset

The dataset consists of 303 rows of data each with 10 features related to cardiac therapy, and a target that specifies whether the person has a heart disease or not. The details of the features are listed below, while Fig. 1 displays a sample of the raw data.

[1] https://www.kaggle.com/ronitf/heart-disease-uci

1. **age:** The age of the person
2. **sex:** Gender with 0 representing females and 1 representing males.
3. **cp:** The type of chest pain. (Integer values ranging from 0 to 3, inclusive)
4. **trestbps:** Resting blood pressure.
5. **chol:** Serum cholesterol measured in mg/dl.
6. **fbs:** A binary variable indicating whether the fasting blood sugar level exceeds 120 mg/dl (positive) or not (negative).
7. **restecg:** Results from resting electrocardiography. (Integer values ranging from 0 to 2, inclusive)
8. **thalach:** Heart rate (maximum value, continuous).
9. **exang:** A binary variable denoting exercise induced angina (0 for NO and 1 for YES)
10. **oldpeak:** ST depression induced by exercise relative to rest (continuous)
11. **slope:** the slope of the peak exercise ST segment (Integer values ranging from 0 to 2, inclusive)
12. **ca:** major vessels colored by fluoroscopy. (Integer values ranging from 0 to 3 inclusive)
13. **thal:** (Integer values ranging from 0 to 3 inclusive)



1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targe
2	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
3	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
4	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
5	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
6	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
7	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
8	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
9	44	1	1	120	263	0	1	173	0	0	2	0	3	1
10	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1

Fig. 1. Example rows from the raw dataset

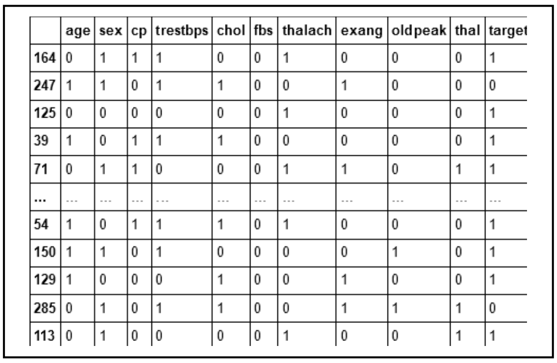
The column target provides the ground truth for whether the person in question has a heart disease or not. The objective for this binary classification problem is that provided basic test results (*restecg, exang, ca, etc*.), one could determine whether (*and/or what*) test results performed depict a heart disease.

II. Procedure

The data was first cleaned/pre-processed before running through the steps stated at the start of this paper. Two metrics were considered for performance (accuracy and F1 score), while explanations were checked against feature importance provided by CART, a decision tree algorithm. These results were the visualized. Baseline algorithms were also used to classify the problem, and all results were tabulated.

A. Pre-processing

One of the main shortcomings of the method proposed in [4] was that the features used were binary. Although the math works out fine for any type of categorical features, the computational complexity that would entail is infeasible. So, for the sake of simplicity, not only were continuous variable categorized, but they were also categorized as binary variables i.e., any variable having a value below the median would be denoted by 0 and a value above the median would be denoted by 1. However, this meant that the data would lose its *energy*, resulting in lower performance. Moreover, not all features nicely fit into this binary categorization. This coupled with the fact that 13 features will account for thousands of clauses justified the exclusion of three features. The features excluded from the dataset were *slope*, *restecg*, *ca*. Part of the resulting data is shown in Fig. 2. Following this, the data was split into train and test set with the ratio 70-30.



	age	sex	cp	trestbps	chol	fbs	thalach	exang	oldpeak	thal	target
164	0	1	1	1	0	0	1	0	0	0	1
247	1	1	0	1	1	0	0	1	0	0	0
125	0	0	0	0	0	0	1	0	0	0	1
39	1	0	1	1	1	0	0	0	0	0	1
71	0	1	1	0	0	0	1	1	0	1	1
...	...	...	...	...	...	...	...	...	...	...	...
54	1	0	1	1	1	0	1	0	0	0	1
150	1	1	0	1	0	0	0	0	1	0	1
129	1	0	0	0	1	0	0	1	0	0	1
285	0	1	0	1	1	0	0	1	1	1	0
113	0	1	0	0	0	0	1	0	0	1	1

Fig. 2. Example rows from cleaned dataset

B. Classification

The algorithms from [4] were implemented. The knowledge base was generated using the direct method on the training set only. Each query in the test set was used to develop constraints in the linear system which was solved using the *Gurobi*[2] solver. Finally, both accuracy and F1 score were measured. The dataset was also run through baseline ML algorithms such as the Naïve bayes classifier, CART, Random forests, and an ANN. The *probabilistic* Naïve Bayes model required no parameter (much like [4]), random forests were built with a maximum depth of 4, and the ANN had two hidden layers with 32 and 16 neurons, respectively. It is important to note that for the random forests and ANN algorithm, the data was categorized but not converted to binary. All results are tabulated in Table I. The code is uploaded in Github[3].

TABLE I. Classification Performane

Model Name	Accuracy	F1 Score
Direct Method [4]	0.80	0.84
Naïve Bayes	**0.85**	0.87
Random Forest	0.82	0.83
ANN / MLP	-	**0.97**
Cart	0.74	0.76

[2] https://www.gurobi.com/

[3] https://www.github.com/saadejazz/explain_prob

C. *Explanation*

Algorithm 6 was implemented from [4] to trace important features pertinent to a query. This however, resulted in several sub-queries being suggested as the optimal path. Therefore, a counting algorithm was adopted to keep track of the most *popular* feature among the list of sub-queries all having the optimal probability. Finally, this algorithm was implemented to reveal important feature-value pairs for three difference cases: (1) positive target (composition displayed in Fig. 3.), (2) negative target (composition displayed in Fig. 4.), and (3) all targets (composition displayed in Fig. 5.). The third type of explanation can be seen as feature importance and is thus compared to important features as revealed by CART (see Fig. 6.).

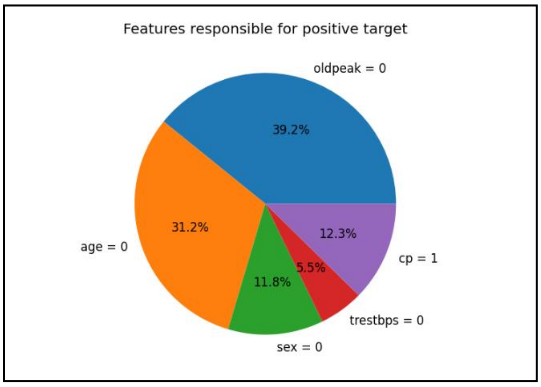


Fig. 3. Most import features and their values (for positive classifications)

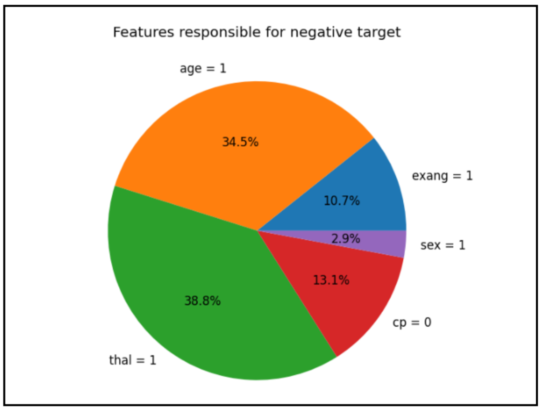


Fig. 4. Most import features and their values (for negative classifications)

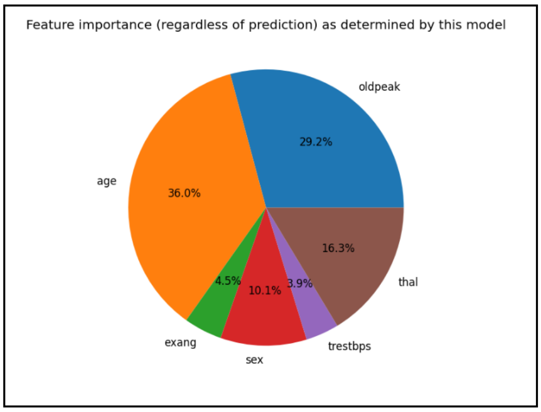


Fig. 5. Most import features and their values (for negative classifications)

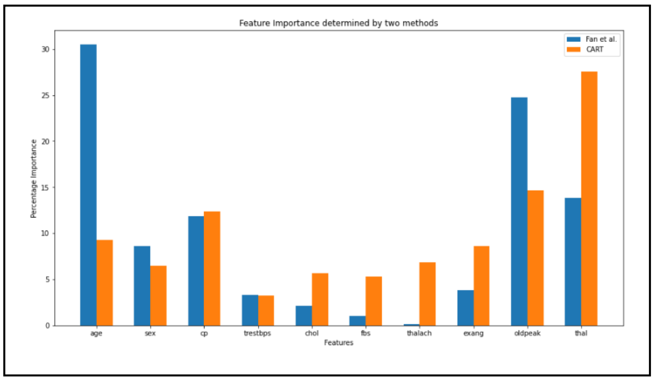


Fig. 6. A comparison of the most important features

From Fig. 3. We can deduce that *age* and *oldpeak* are the primary reason for the model to predict the presence of a heart disease. For the negative classification, *thal* and *age* dominate the cart in Fig. 4. For the overall feature importance, the algorithm in [4] gives more weightage to *age.* Finally, when compared with feature importance deduced by CART, there are considerable differences. Unfortunately, there is no concrete explainability metric to determine better explanations. That being said, it may be noted that the direct method performed slightly better than CART, and if performance influences explanation, the direct method has better chances.

III. Conclusion

The explainable AI model suggested by *Fan et al.* works well on smaller datasets such as the one under discussion in this paper. An explainability/feature importance analysis revealed age to be the most decisive factor in prediction heart disease. However, this is still based on correlation instead of causation, and advanced method and/or human intervention is required to establish ground truth and more concrete metrics for estimating the performance of and explainable AI model.

References

[1] Castelvecchi, Davide. "Can we open the black box of AI?." *Nature News* 538, no. 7623 (2016): 20.

[2] Holzinger, Andreas, Georg Langs, Helmut Denk, Kurt Zatloukal, and Heimo Müller. "Causability and explainability of artificial intelligence in medicine." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9, no. 4 (2019): e1312.

[3] Doran, Derek, Sarah Schulz, and Tarek R. Besold. "What does explainable AI really mean? A new conceptualization of perspectives." *arXiv preprint arXiv:1710.00794* (2017).

[4] Fan, Xiuyi, Siyuan Liu, and Thomas C. Henderson. "Explainable AI for Classification using Probabilistic Logic Inference." *arXiv preprint arXiv:2005.02074* (2020).