Explainable Machine Learning in Cardiovascular Diagnostics

Alexander Gutell and Ludvig Skare

***Abstract*—The major challenges in implementing machine learning models in medical applications stem from ethical and accountability concerns, which arise from the lack of insight and understanding of the models' inner workings and reasoning. This opaqueness has resulted in the emergence of a new subfield of machine learning called *Explainability*, which aims to develop and deploy methods to gain insight into how input data is weighted and propagated through a machine learning algorithm. This paper aims to examine the viability of certain explainability methods when applied to cardiovascular diagnostics. The machine learning models that were implemented and subsequently evaluated include Logistic Regression, Decision Trees, and Random Forests. Methods such as Feature Importance plots, Lasso Regularization (L1 norm), and Sequential Feature Selection were applied to achieve better interpretation of these models. The results indicate that different models and forms of regularization prioritize various input features more heavily than others, even when trained on identical data. A consistent finding across all models, except for Logistic Regression with Lasso regularization, was the ability to significantly reduce the dimensionality of the input feature space without substantial loss in model test performance. This allows for the isolation of specific features, thereby enhancing insight into and improving a model's interpretability. Systolic and diastolic blood pressure along with cholesterol values were the two main input features that determined a patients cardiovascular diagnosis.**

***Sammanfattning*—De största utmaningarna vid implementering av maskininlärningsmodeller i medicinska tillämpningar härrör från etiska och ansvarsfrågor, som uppstår på grund av bristande insikt och förståelse för modellernas inre funktion och resonemang. Denna opakhet har resulterat i framväxten av en ny underkategori inom maskininlärning kallad *Förklarbarhet* (Explainability), som syftar till att utveckla och tillämpa metoder för att få insikt i hur indataviktning och spridning sker genom en maskininlärningsalgoritm. Denna studie ämnar undersöka genomförbarheten av vissa förklarbarhetsmetoder när de tillämpas på kardiovaskulär diagnostik. De maskininlärningsmodeller som implementerades och utvärderades inkluderade logistisk regression, beslutsträd och random forests. Metoder såsom Feature Importance-diagram, Lasso-regularisering (L1-norm) och sekventiell egenskapsurval tillämpades för att uppnå bättre tolkning av dessa modeller. Resultaten visar att olika modeller och former av regularisering prioriterar olika indatavariabler mer än andra, även när de tränas på identiska data. En konsekvent upptäckt för alla modeller, förutom logistisk regression med Lasso-regularisering, var förmågan att avsevärt minska dimensionaliteten i indatavariabelrymden utan betydande förlust i modelltestprestanda. Detta möjliggör isolering av specifika variabler, vilket förbättrar insikten och tolkningsbarheten för en modell. Systoliskt och diastoliskt blodtryck samt kolesterolvärden var de två huvudsakliga indatavariablerna som avgjorde en patients kardiovaskulära diagnos.**

***Index Terms*—Explainability, Machine learning, XML, Healthcare diagnostics, Cardiovascular Disease**

Supervisors: Ragnar Thobaben

TRITA number: TRITA-EECS-EX-2023:184

I. Introduction

Machine Learning (ML) is both an established and expanding field with many implementations in both theoretical and applied applications. Fundamentally, the goal is to create and deploy mathematical models and train them on collected data, with the help of algorithms, to make predictions and classifications of new data points. The difference, and what makes ML distinct from traditional statistical methods, is the ability for a model to update itself based on new data, without human interference and effort, which can be seen as a form of learning, hence the field name of *Machine Learning*. Overall, ML is capable of transforming and improving many different type of fields in society, especially where there are larger amounts of data and a desire to predict or classify outcomes. According to MIT some common implementations of ML in today's society are chat bots, predictive text (auto completion), language translation, media exposure on social media platforms, as well as self-driving cars and medical diagnostics [1]. In this article we will be exploring how ML can be implemented in healthcare and medical diagnostics as well as addressing the ethical and practical issues that limit its full scale implementation.

II. Background

As ML models grow in complexity they allow us to solve more difficult problems with increasing precision. An example of this would be using a Neural Networks model to correctly classify images of different diseases. More complex and accurate models usually come at the cost of insight and understanding of how the implemented model works [2]. Insight and understanding are defined as *Explainability* and *Interpretablility* and constitutes the heart of Explainable AI (XAI) and Explainable Machine Learning (XML). XML aims to improve the understanding of how different ML models work and how they reason. In other words, how they process the input data in order to e.g. make a classification of whether a medical image contains a disease or not.

XML is not always necessary in all applications or fields as sometimes the only important factor is correct classifications. However when ML models are applied to fields that directly affect human beings, their lives and their health, explainable machine learning becomes an absolute necessity according to Burkart [3]

A. Need for explainability

Explainability is necessary when we aim to implement ML models in healthcare for several different reasons. The essence of healthcare is to give patients a suitable and relevant treatment that in turn aims to make them a healthier individual, capable of living a life not inhibited by sickness och health problems. As healthcare can have substantial effect on the patients life, whether they are sick or not, it is imperative that we provide the best possible conditions for diagnosis and treatment. To meet these conditions, analysis and evaluation of the involved processes is essential, both for productive, societal and ethical reasons. Therefore it is important that ML addresses these problems accordingly. In the following paragraphs a selection of important topics are listed.

1) Trust

According to Burkart [3] humans have an innate need of understanding a decision or at a minimum an explaination for certain decisions. This is due to the fact that humans do not trust blindly as Burkart [3] states. Burkart [3] continues by defining that trust and acceptance of a system is necessary in its deployment where knowing the models strengths and weaknesses are a key prerequisite for human trust and understanding. If we do not trust the model then what is the point of deploying and implementing it? Its prediction and effect will be obsolete regardless of outcome if the results are not trusted.

2) Correlation

Correlation encompasses a predictable input-output relationship between patient data and their diagnosis. Diseases are most often linked to certain biometrics. An example of this is diabetes that can be linked to insufficient insulin production and varying levels of blood glucose levels according to the United States National Institutes of Health [4]. If a model was trained for diabetes prediction and consistently had large weights for insulin and blood glucose levels when predicting if a patient had diabetes then we would feel there is a strong correlation between these parameters. Thus resulting in greater trust in the model. If instead the model prioritized different biometric values for different patients this would result in an unclear and unreliable relationship between data inputs and their diagnostic outputs. An example of this would be a model that predicts patient A has diabetes due to high blood sugar levels but predicts that patient B also has diabetes due to their large feet and patient C does not have diabetes due to their hair color. Would we trust and implement a model with such an unclear behaviour? Most likely not regardless of the accuracy. Explainability is once again a method of evaluating such performance.

3) Ethics and fairness

Equality and non-discriminatory classifications are also of great importance. If a model works very well for a certain type of age group or ethnicity but is less accurate and unreliable for remaining groups then the model is not general and cannot be applied in all circumstances. Thus explainability insight is of utmost importance to identify eventual biases and built-in discrimination when it comes to medical diagnostics and treatment. Assuming that eventual biases are not able to be reduced due to intrinsic properties of the available training data, then insight into the models inner workings grants us the knowledge of to which medical groups the model is applicable. Burkart et al. [3] expands on these proxy functionalities where we can examine a model based on other important factors such as safety, fairness, privacy and robustness.

4) Informativeness

Informativeness is described by Burkart [3] as the necessity to evaluate if the model actually serves its intended purpose. If this can be verified through the ability to observe the inner workings of the model the credibility of the model increases.

5) Accountability

Accountability and responsibility are cornerstones of the judicial aspect of implementation of ML models in various sectors. Are ML models allowed to be implemented yet exempted from accountability if e.g. a life is lost due to an incorrect diagnosis? Is it a doctors responsibility or the company that developed the ML model diagnostics tool that is to be held responsible in the case of a tragedy? In order for an individual or program to be held accountable it must first be proven to be capable of producing consistent non-random decisions. One way of doing this is implementing Explainability methods in order to understand the strengths and weaknesses of the model.

6) Adjustments

Another reason for explainability in general that also applies to healthcare is the ability to make adjustments. Burkart [3] talks about how understanding a models inner workings allows a domain expert to compare the prediction model to current domain knowledge. This can be utilized in order to make adjustments to the model that better fits the existing domain knowledge which in turn can improve the models performance and potentially the acceptance of its implementation.

III. Explainability

A. The subject of explanation and its definition

This project is restricted to supervised learning (SL) and binary classification problems. To formulate an SL problem, let $\mathbf{x} \in \mathcal{X} \subseteq R^n$ be a datapoint represented by features, and $y \in \mathcal{Y} \subseteq R$ be a label representing the outcome based on $\mathbf{x}$, then approximate $h(\mathbf{x}) = y$ by fitting $h(\mathbf{x})$ to the training data $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$. The main goal is to approximate $h(\mathbf{x})$ in such a way that unseen examples of $\mathcal{X}$ are mapped to the correct values in $\mathcal{Y}$.

B. Gaining Interpretability

Burkart [3] suggests three main ways of gaining interpretability for $h(\mathbf{x})$. They are presented briefly below to

demonstrate the idea and concept of how the field of explainability reasons and discusses generating explainability of models.

1) Explanation Generation

Explanation generation defines a function that generates some sort of explanation belonging to the set of all possible explanations. The vagueness of this definition stems from the wide variety of explanations. Explanation generation can be thought of as being presented with an explanation to a query, such as: *Why was I diagnosed with having a heart disease?* A generated explanation could then be presented in the form of an answer on a screen, with the text *because of high cholesterol in combination with low activity*. Another generated explanation might show the patient, represented as a data point, grouped with other patients with similar genetics who also have heart disease. Yet another example could be an audio explanation filled with medical terms that a layperson might not understand, but which a doctor would accept as a perfect explanation. The importance of how we put an explanation together is discussed in the next subsection C. Components of an explanation.

2) Surrogate Model Fitting

Model surrogate learning aims to approximate a complex and hard-to-interpret model, such as a deep neural network, with a more interpretable model in order to achieve a deeper understanding of the inner workings.

3) Learning Interpretable Models

Learning interpretability models refers to $h(\mathbf{x})$ being an interpretable model, which is considered relatively easy to understand.

Explanation generation and model surrogate learning can be used to gain interpretability for both local and global approaches. A local approach tries to explain how a prediction was made for a single data point while a global approach attempts to explain the model as a whole. For this project, both explanation generation and learning interpretable models will be relevant, and we will not be implementing surrogate model learning.

In summary, these methods of gaining interpretability lacks detail due to the numerous ways they can be achieved for various models. Nevertheless, these five methods serve as a starting point.

C. Components of an explanation

According to Burkart [3] an explanation can be constructed with three components, *What*, *How* and *Whom*. The *What* refers to the content of the explanation, the *How* pertains to the medium through which the explanation is communicated, and the *Whom* designates the level of expertise the explanation should cater to, i.e., the prerequisites of the target group. These components are the core to determining what constitutes a good explanation and will be discussed in the conclusion.

IV. Model selection

There are many different models $h(\mathbf{x})$ we can deploy to solve an SL problem, and depending on the problem at hand, some alternatives will be more suitable than others. In this project we aimed to predict cardiovascular disease in patients, which means our model will make high-stakes predictions. Consequently, accuracy alone is not the most crucial criterion for our model, we also require interpretability. As mentioned in the "Gaining Interpretability" section, we will utilize "Learning Interpretable Models", which involves selecting a interpretable model $h(\mathbf{x})$. Three such models, that are popular and widely used within ML include Logistic Regression, Decision Tree and Random Forest.

V. The Models

A. Logistic Regression

Logistic Regression (LR) is a classifier and a parametric model that predicts outcomes based on conditional probabilities associated with data points. In essence, LR takes a data point, estimates the probability of a particular outcome, and then assigns a class depending on a predefined probability threshold as is exemplified in Figure 1. The reason for stating that LR is a parametric model highlights that the probability estimation's shape and form are essentially assumed to be correct, thereby limiting the hypothesis space to a single model family. To fit the probability function to the data, maximum likelihood estimation (MLE) is adopted to estimate the parameters.
The probability estimation for LR consists of the Sigmoid function:

$$\hat{p}(y_i = 1|X_i) = \frac{1}{1 + exp(-X_i w - w_0)} \tag{1}$$

where X_i is the datapoint, w is the weight vector and w_0 is the bias constant.

The cost function derived from MLE is defined as

$$\underset{w}{\operatorname{argmin}} = -C \cdot \sum_{i=1}^{n} (y_i \log(\hat{p}) + (y_i - 1) \log((1 - \hat{p})) + r(w) \tag{2}$$

where C is a constant, y_i is the label of the datapoint, $\hat{p}$ is the datapoints probability and $r(w)$ is a regularization function.

1) Engineering for interpretability and performance

In order to make our model more interpretable we want to avoid multicollinearity between the features [5], and this can be achieved by making the model more sparse i.e removing features. However, the constraints of sparsity can have an effect on performance, which needs to be taken into consideration when modeling. In turn, there are many metrics for measuring performance of a model and we will make use of some of the most established and popular ones that are used in practice.

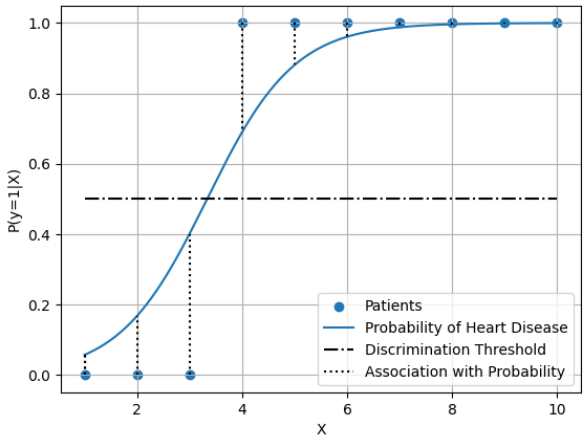


Fig. 1. Example of a Logistic Regression curve fitted to a toy set of data. The true labels are either 1 (presence of heart disease) or 0 (no presence of heart disease) and have corresponding arbitrary feature values X.

2) Feature Selection

As previously mentioned, a sparse model with low multicollinearity is desirable for interpretability. Multicollinearity arises when the features in a linear regression model correlate with each other, meaning that a change in one feature entails a change in other correlated features. This may not impact the model's accuracy but can result in some features being statistically insignificant even if they are significant individually. In other words, prediction is fine, but interpretation is not. For example, suppose two features, smoking and degree of physical activity is highly correlated in our data, and our goal is to predict lung cancer. The model might fit itself to the data in such a way that the prediction relies heavily on activity while having a minimal impact from smoking. Should we interpret this as smoking not being harmful or that being very active eliminates the risk of lung cancer from smoking? An expert might argue that these interpretations are not in line with empirical studies of lung cancer, suggesting that it might be better to remove one of these features for improved interpretation.

Several remedies exist to multicollinearity, but we will focus on sparsity which indirectly reduces multicollinearity by making the feature space smaller. In this report we will utilize Sequential Feature Selection (SFS) and Lasso Regularization to achieve this. Another method is Principle Component Analysis where the input data is projected onto a feature space of lower dimensionality. The data is projected so that the maximum amount of variance and information is preserved.

Lasso Regularization puts a L_1-norm constraint on the cost function, $r(w) = ||w||_1$, and has the effect of prioritizing weights that are small or zero in order to minimize the cost function, thus making the model more sparse.

SFS begins by selecting the optimal feature to train the model on. Once this is done, SFS will keep adding one feature at a time in combination with the former one/ones, in order to keep optimizing for a given metric. In our case this is the AUC-value, and the algorithm will stop adding features when the increment of the AUC (Area under the ROC Curve) is below a certain threshold.

B. Decision Tree

A Decision Tree is a non-parametric model that provides probabilities of outcomes. It employs recursion to divide the feature space into partitions. The model is trained in a manner that optimizes the partitions to best represent the empirical data distribution. When given an unseen data point, the model determines the probabilities associated with the partition containing the data point, thereby generating a prediction.

A notable advantage of Decision Trees, in the context of explainability, is that the resulting partitions corresponds to simple questions that can be visualized in a tree-like structure, with each question represented by a node (see example in Fig. 2). By following these questions, one can arrive at a final question that leads to a prediction. This makes Decision Trees easy to interpret, as long as the tree depth is manageable. Entropy and Information Gain are the primary methods for obtaining the optimal partitioning question for each node. Gini Impurity is used to evaluate the leaf nodes purity and subsequently the trees performance.

Decision Trees also perform automatic feature selection [5] and can be beneficial as a tool for comparing the model's selection with one's own. However, a negative aspect of Decision Trees is their tendency to overfit and their sensitivity to small variations in the data when undergoing training. This can result in very different trees being generated based on such variations [5].

1) Gini Impurity

Gini impurity is a commonly used method to measure the impurity of decision trees leaf nodes, i.e. how many wrongly classified data points are in a respective leaf node. This impurity measure is defined as [6]:

$$\text{Gini impurity} = 1 - \sum_{i=1}^{K} p_i^2 \tag{3}$$

where K is the number of classes that data points can be classified as. p_i are the probabilities of the different class labels for each leaf node in the decision tree.

If a leaf node is pure, i.e. only has data points of one class, then the p_i^2 = 1. This entails a Gini impurity of 0 for a perfectly classified leaf node. A fully imbalanced leaf node, for two classes, where the frequency of each class in the leaf node is 0.5 respectivly results in a Gini impurity of 0.5 as $\frac{n_k}{n_{tot}}$ = 0.5 for both classes of the leaf node. n_k is the frequency of a given class and n_{tot} is the total amount of datapoints in the leaf node. A gini impurity of 0 entails a perfect classification for the leaf node and a gini impurity of 0.5 is the worst possible classification

.

2) Entropy and information gain

In information theory entropy is a measurement of how mixed or disorganized a group or set or information is. The entropy of a set of information is defined as [7]:

$$\text{Entropy: } H(T) = -\sum_{i=1}^{K} P(X_i) log_b P(X_i) \tag{4}$$

The most commonly implemented logarithm is base 2 where we work with bits of information. A homogeneous group of data results in a low entropy with an entropy of zero if all data points are of the same type. A group consisting of data points of different types results in a higher entropy. Entropy is an essential implementation in decision trees as a form of determining the best partitioning question that results in the largest decrease in entropy, so called Information Gain [8].

$$\text{Information gain: IG} = H(\text{previous}) - H(\text{current}|\text{partition}) \tag{5}$$

A decrease in entropy is equivalent to our dataset being partitioned into more homogenous subgroups, i.e classifying the datapoints into different groups or classes making the groups more pure. The goal of a decision tree is to parition a set of data into homogeneous groups and is a greedy algorithm entailing that it aims to find a partitioning question and value that maximually reduces the overall entropy of the dataset each layer.

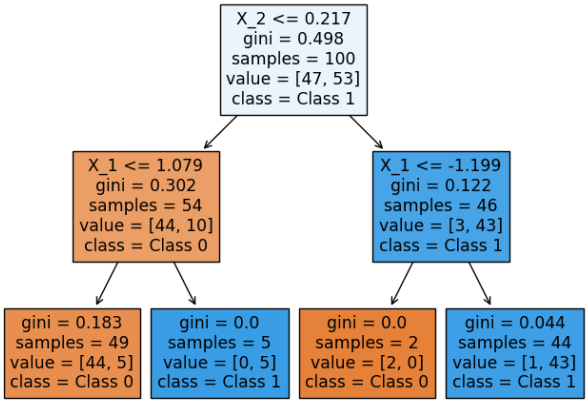


Fig. 2. Example of a Decision Tree fitted to a toy set of data. The true labels are either 1 (presence of heart disease) or 0 (no presence of heart disease) and have two corresponding arbitrary feature values X_1 or X_2.

3) Feature Importance

In order to determine a feature's importance for a decision tree, the total decrease of the weighted impurity for that feature is calculated and normalized [9].

The weighted impurity decrease equation:

$$\frac{N_t}{N} \cdot (I - \frac{N_{t,R}}{N_t} \cdot I_R - \frac{N_{t,R}}{N_t} \cdot I_L), \tag{6}$$

where N_t is the number of data points at the current node, $N_{t,R}$ and is the number of data points in the right child, $N_{t,L}$ is the number of data points in the left child, and N is the total number of data points. I is the impurity, I_R is the right impurity and I_L is the left impurity.

C. Random Forest

Random forest is an ensemble machine learning classifier model that consists of numerous randomly initiated decision trees which intuitively gives the model its name. A random forest classifier consists of n decision trees all trained on the same data yet initialized and composed of different partitioning questions. This results in every decision tree being unique where unknown data points runs through unrepeated partitioning processes. Each tree in the random forest can be seen as an individual decision tree model trained on the training data (a random forest is thus an ensemble of many separate decision trees), see example in Fig. 3. The random forest classifies new data points by computing the respective outputs for each decision tree and later taking the most frequent class attributed to the data point [10]. A related example to our healthcare approach would be training a random forest model with 100 decision trees to predict if a patient has CVD. Each datapoint is passed through every decision tree and thus obtains 100 independent predicted outcomes. If 51 or more trees predict that the patient has CVD then the random forest predicts the patient has CVD and vice versa. A larger consensus among the trees can be seen as a more convincing prediction.

Ensemble random forest models have several performance advantages when compared to decision trees. The main advantage being that random forests prevent overfitting of the data as numerous independent decision trees act as a built in regularization [10]. The regularization aspect emerges internally as each decision tree has a unique composition of partitioning questions and values that prevents the model from "focusing" too much on a certain feature och parameter value. Random forests are also more resilient to outliers in the data as the model averages numerous predictions from all decision trees.

A primary shortcoming of random forests models is the loss of explainaiblity compared to a single decision tree. As random forests consists of *n* decision trees it is increasingly difficult to visually interpret the models behaviour as *n* grows larger. Hundreds or thousands of decision trees are impossible to simultaneously examine and be understood by human beings, let allow visualize side by side on a screen or piece of paper. This is somewhat mitigated by the ability to analyse individual decision trees to evaluate if the trees of the random forest model are behaving as predicted. Another possibility is observing the individual predictions and obtaining a visual and explainable overview of the frequencies for the different classes. Random forests are also expected to be more expensive to produce with respect to time and computation power.

1) Feature Importance

The feature importance for a random forest model is calculated in the same way as for the decision tree in Eq. (6)

D. Model Performance

In terms of medical terminology a detection of disease is classified as true positive (TP), while a correct prediction of a healthy individual is classified as true negative (TN). However, for any diagnostic test, there is a probability that

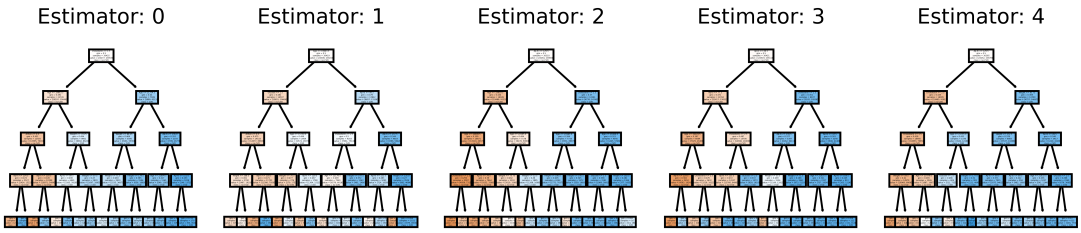


Fig. 3. 5 decision trees from a trained random forest with 300 decision tree estimators with a maximum depth of 4.

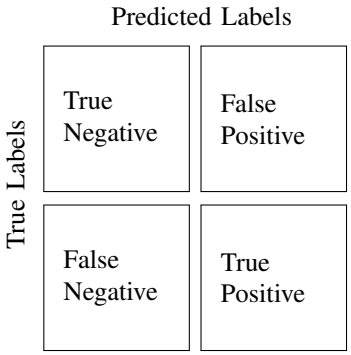
some predictions will be false, and so the metrics false negative (FN) and false positive (FP) are defined. These metrics are often represented in a confusion matrix to provide an overview of the performance. Depending on the specific problem, different relationships between these metrics may be desirable to analyze. In the following equations TP, TN, FP, and FN are all quantifiable amounts of patients with respective classification.

$$Sensitivity = \frac{TP}{TP + FN} \tag{7}$$

$$Specificity = \frac{TN}{TN + FP} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$F_1 = \frac{2 \cdot Precision \cdot Sensitivity}{Precision + Sensitivity} \tag{10}$$



When assessing performance, there may be various incentives for creating a model that performs particularly well on a combination of these metrics. For instance, when diagnosing a disease, it might be desirable to minimize FN, therefore emphasizing sensitivity as a metric. Conversely, there may also be an economic aspect to the testing, such as the treatment or operation is costly, making it desirable to minimize FP by focusing on precision. Another metric that is commonly used is the F_1-score, which is the harmonic mean between recall and sensitivity. Ideally, we want zero FPs and FNs, but this is not a realistic expectation for a model. Bearing this in mind, feature selection should be conducted in a manner that prudently considers these incentives.

In this report, we primarily focused on ROC (Receiver Operating Characteristic) and AUC (Area under the ROC Curve) to determine the model performance. ROC and AUC are two commonly used metrics in binary classification problems. The ROC curve is plotted as the True Positive Rate (TPR) versus the False Positive Rate (FPR) for different discrimination thresholds. These thresholds determine whether a test result is positive or negative (i.e. has a presence or absence of heart disease). In essence, the ROC curve helps us visualize the trade-off between TPR and FPR.

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

$$FPR = \frac{FP}{FP + TN} \tag{12}$$

Due to the relationship of TPR and FPR, the ROC curve will lie between zero and one on both axes. A high AUC value indicates good overall performance across all possible thresholds, which is desirable for a classifier when seeking a good balance between TPR and FPR. The points on the ROC curve closest to (TPR, FPR) $= (1, 0)$ is called the optimal point and corresponds to the best possible threshold value in terms of TPR and FPR.

VI. Method

A. Overview

In order to analyse the explainability of a specific machine learning model, i.e. how the model processes inputs and generates an outcome or prediction, we first need a developed model. Developing an accurate model consists of several stages which all aim to improve the performance of the model. These stages are described below and can be summarised as:

- Find or produce relevant data to train your model on.
- Process data to reduce sampling errors and human errors. Such as eliminating data points with missing values or balancing the data set to reduce overrepresentation of a certain class
- Feature analysis and dimensionality reduction where we investigate the possibility of eliminating certain data features that have little effect on the models prediction power. Dimensionality reduction can be used to simply the datas complexity by projecting it onto a lower dimensional plane. The aim is to simply the data as to improve model interpretability and insight.

- Training of model through choice of model class and tuning of possible hyperparameters.

B. Acquiring a dataset

The data that was used in this project was the cardiovascular dataset *Cardiovascular Disease dataset* found on the open source machine learning dataset website Kaggle [11]. The dataset consists of 70 000 data points. Each data point consists of an id value, 11 feature values and a binary classification of cardiovascular disease. Each data point consists of the features listed in Table I

TABLE I:
FEATURES AND COMPONENTS OF EACH DATA POINT.

Label	Feature type	Unit
Age	Objective	Days (integer)
Height	Objective	cm (integer)
Weight	Objective	kg (float)
Gender	Objective	Binary (0-1)
Systolic BP (ap-hi)	Examination	mm Hg (integer)
Diastolic BP (ap-lo)	Examination	mm Hg (integer)
Cholesterol	Examination	1: normal, 2: above normal, 3: well above normal
Glucose	Examination	1: normal, 2: above normal, 3: well above normal
Smoking	Subjective	Binary(0-1)
Alcohol intake	Subjective	Binary (0-1)
Physical activity	Subjective	Binary (0-1)
Presence of cardiovascular disease	Target variable	Binary (0-1)

C. Data preprocessing

Initially plotting our unprocessed data with the help of boxplots, see Figure 4, allows us to identify and target non-physiological anomalies.

After observing the different boxplots we identified three features that contained anomalies of extraordinary nature. The three features being height, systolic blood pressure and diastolic blood pressure. As can be observed in the height boxplot we have a datapoint with a height of 250 cm which is extremely rare. It is so rare that we deemed it statistically impossible for it to occur in a small sample size of 70 000 data points. We defined a reasonable higher limit for statistically reasonable heights of 220 cm. After dropping all data points with a height larger than 220 cm our new maximum height is 207cm which is a more reasonable data set.

According to the CDC [12] normal values for systolic and diastolic blood pressure is less than 120 mm Hg and 80 mm Hg respectively. Similarly they define high blood pressure as more than 140 mm Hg and 90 mm Hg respectively. This information gives us the insight to conclude that blood pressures in the thousands and even hundreds as seen in the extreme data points of the systolic and diastolic blood pressure box plots in Figure 4 are unrealistic. Based on this and the fact that people with cardiovascular diseases often have elevated blood pressures we set an upper limit for systolic and diastolic blood pressure as 250 mm Hg and 200 mm Hg respectively. We also identified data points with negative blood pressure values and thus set a

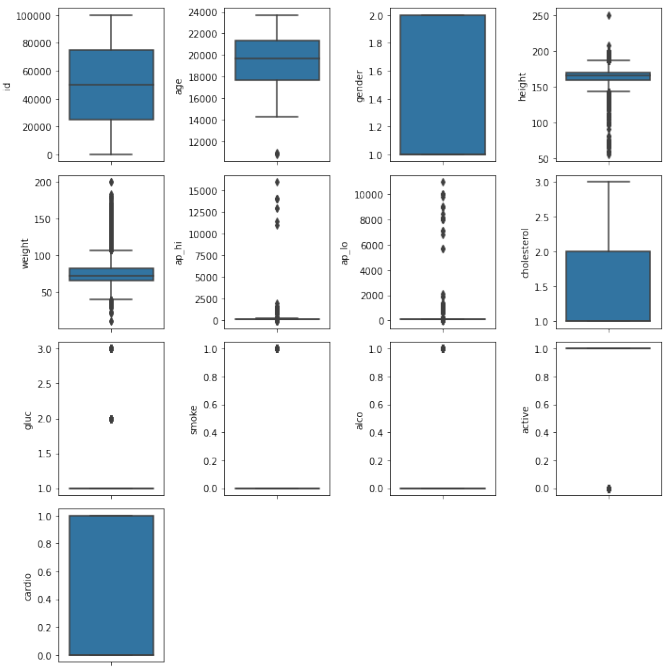


Fig. 4. Boxplots for the unprocessed dataset from Kaggle

lower limit as 40 mm Hg for both systolic and diastolic blood pressure.

As an effect of this data processing we removed 1242 data points from the data set which corresponds to 1.77% of the number of original data points. Leaving us with 68 758 usable data points.

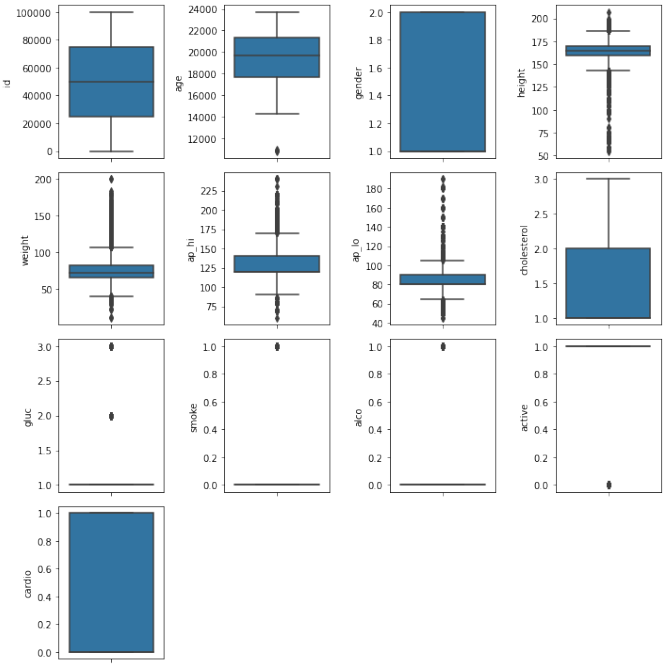


Fig. 5. Boxplots for the processed dataset

Another important component of dataprocessing is as previously stated to ensure that the data is balanced in the sense that no class is overrepresented.

Our first implementation of the *scikit-learn* library was to divide our now processed data into 3 distinct sets using two iterations of the *scikit-learn* function [13] *test_train_split*. Each use of *test_train_split* partitions a numpy 64int array [14] into two smaller partitions of specified size. We used the random seed 1234 which pseudo-randomized the partitioning process of the data points entailing a random split of the data that could be reproduced by using the same seed. Our training data set of 48 131 data points consists of the variables x_train and y_train. Our validation data set of 10 314 data points consists of the variables x_val and y_val. Our test data set of 10 313 data points consists of the variables x_test and y_test. The x variables are $n \times m$ dimensional Numpy 64int arrays [14] that have a length corresponding to the number of n data points and where each row has m columns corresponding to all the input features that each data point has. E.g x_train is a 48131×11 Numpy 64int array [14] which represents a 48131×11 matrix where all 48 131 number of rows contain the 11 features that each data point has.

D. Logistic Regression

1) Training the model

The logistic regression model was trained with the help of the machine learning library *scikit-learn* [15]. In order to get a benchmark of how logistic regression could perform, we began by training the model without regularization. The performance metrics, trained weights and produced plots were then used for comparison. We were interested in:

- Maximizing AUC
- Imposing sparsity
- Prioritize feature selection backed by science

The workflow for each model is displayed below.

Workflow for LR with SFS:

1) train model on training data with regularization parameter C
2) apply SFS
3) check what features got choosen
4) train model on the transformed training data
5) compute AUC on training and validation set
6) store results

This was iterated over C = 0.0001, 0.001, 0.01, 1, 10, 100 and without regularization. The SFS-tolerance was set to 0.001 (roc_auc_score) and the results were then compared in order to choose the best considered model. The ROC-curve was then plotted and the optimal threshold value calculated.

Workflow for LR with lasso regularization:

1) train model on training data with regularization parameter C
2) check feature values
3) compute AUC on training and validation set
4) store results

This was iterated over C = 0.0001, 0.001, 0.01, 1, 10, 100 and without regularization. The results where compared in order to pluck the best considered model. The ROC-curve was then plotted and the optimal threshold value calculated.

E. Decision Tree

The decision tree model was trained using sci-kit learns decision tree model which chose the decision questions and values that gave the lowest Gini Impurity in the leaf nodes and that reduced the entropy for each layer. The decision tree was trained on the training data and its performance tested on our validation set. The only form of regularization that was tested and implemented was testing different number of layers. We iterated through layer depths between two and ten, and then picked the model with the highest AUC. For specific performance metrics, used parameters and comparison, see Table II.

F. Random Forest

Our Random Forest model was trained using sci-kit learns random forest model which implemented and varied the decision questions as described above and also generated *n* number of decision trees. The regularization parameters that were manually adjusted were the depth of each tree as well as the the number of trees. The reasoning and choice of these parameters was finding the fewest number of trees (lower training computation costs) and lowest depth (increased interpretability) that gave the highest AUC. We iterated through decision tree depths between two and ten and iterated through 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 and 1000 trees in the random forest. For specific performance metrics, used parameters and comparison, see Table II.

VII. Results

A. Logistic Regression

Out of the LR SFS models, the chosen model had the best AUC score of 0.79 together with a reduction down to five features.

Out of the LR lasso models the chosen model had the next best AUC score of 0.78 along with a reduction down to ten features. The reason for not picking the model with the best AUC score was due the chosen model putting more importance on examination features and less on subjective ones.

For specific performance metrics, used parameters and comparison, see Table II. A comparison between the ROC curves generated with the validation data and test data can be seen in Fig 8 (SFS) and in Fig 9 (Lasso).

In Fig. 6 the feature importance for logistic regression with SFS is displayed as a bar plot. Bars on the positive side of the log odds ratio implies that an increment of the corresponding feature value will increase the risk of diagnosis of heart disease. The bars on the negative side implies that a increment of the corresponding feature value will do the opposite. For the LR SFS, in Fig 6 we can observe that high cholesterol, high blood pressure and weight contributes to a positive CVD diagnosis meanwhile being active contributes to a negative CVD diagnosis. In Fig. 7 the same type of plot is shown for logistic regression with lasso regularization. We can observe in Fig 7 that lasso regularization attributes high cholesterol values, high blood pressure, weight and gender to contributing to a positive CVD diagnosis. Meanwhile being active, having

higher glucose values, and having a larger height contributes to a negative CVD diagnosis. The confusion matrices for LR SFS and LR lasso regularization can be observed in Fig. 10 (SFS) and Fig 11 (Lasso) respectively.

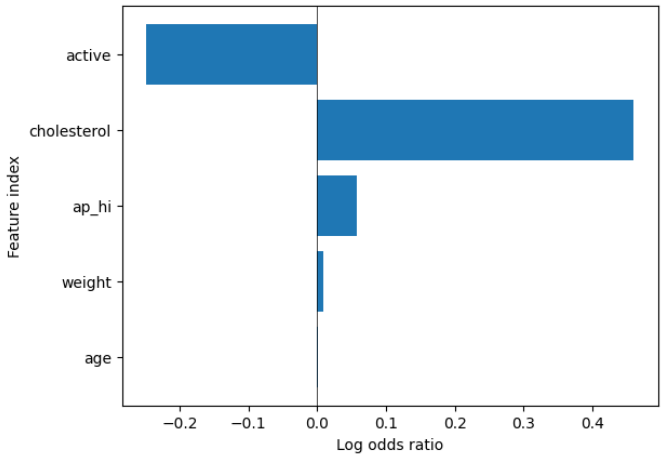


Fig. 6. Feature Importance from logistic regression with implemented SFS.

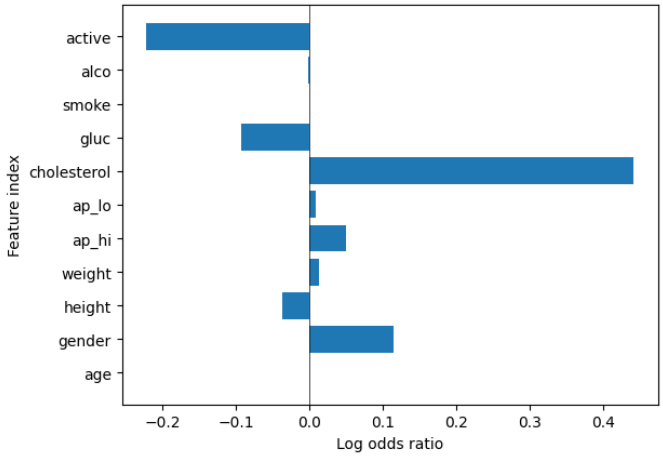


Fig. 7. Feature Importance from logistic regression with implemented lasso regularization.

TABLE II:
PARAMETER VALUES AND DIFFERENT METRIC VALUES FOR DIFFERENT MODELS EVALUATED ON THE TEST DATA

Metrics and Parameters	LR SFS	LR Lasso	DT	RF
AUC	0.79	0.78	0.78	0.80
Sensitivity	0.70	0.68	0.71	0.67
Specificity	0.75	0.77	0.74	0.78
Precision	0.74	0.75	0.73	0.76
F_1-score	0.72	0.71	0.72	0.71
C (LR) / Depth (DT and RF)	10	0.01	4	10
Threshold	0.46	0.48	0.52	0.49

B. Decision Tree

The decision tree had an AUC of 0.78 on the test set. The ROC comparison between validation data and test data can be seen in Fig 13. The feature importance generated by the tree is shown in Fig 12 where it can be observed that high blood pressure, age, high cholesterol values, and high glucose values are the prominent features in that order. To observe the confusion matrix see Fig 14.

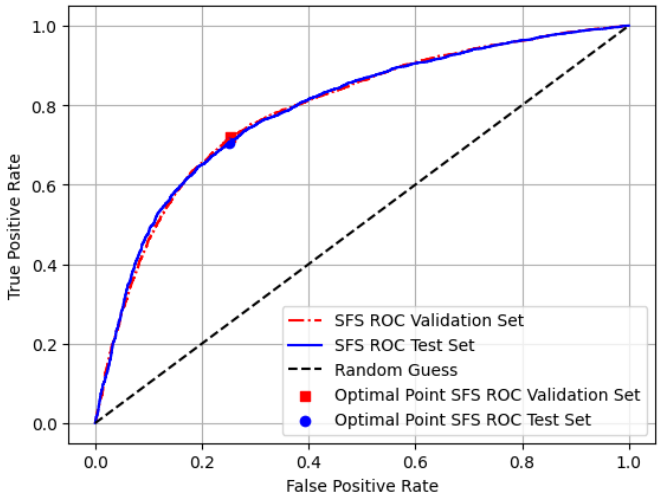


Fig. 8. ROC-curve over test- and validation data. Model: Logistic regression with implemented SFS.

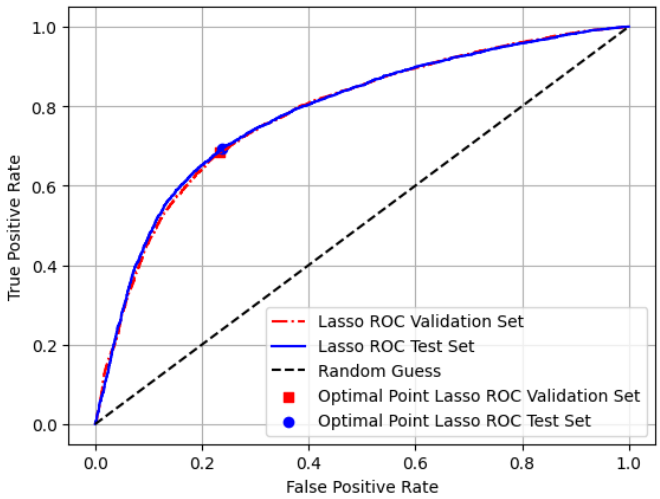


Fig. 9. ROC-curve over test- and validation data. Model: Logistic regression with implemented lasso regularization.

C. Random Forest

The Random Forest model achieved an AUC of 0.80. The ROC comparison between validation data and test data can be seen in Fig 16. The feature importance generated by the model is shown in Fig 15 where it can be observed that high blood pressure, age, high cholesterol values, weight and height were the most prominent features and had noticable effects on the data sets impurity. Meanwhile the remaining features being active, drinking alcohol, smoking, and gender had minimal effects on the impurity of the partitioning. To observe the confusion matrix see Fig 17.

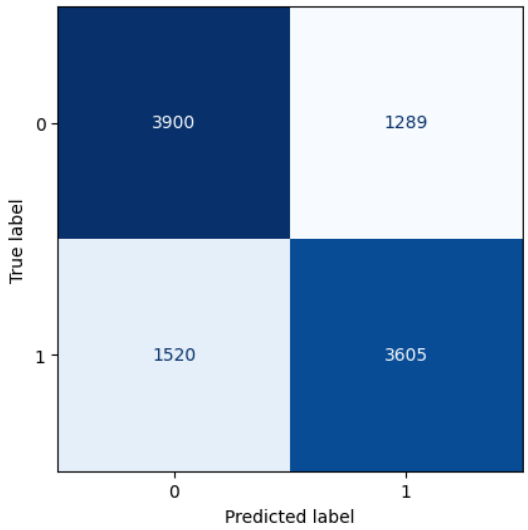


Fig. 10. Confusion Matrix over test data. Model: Logistic regression with implemented SFS.

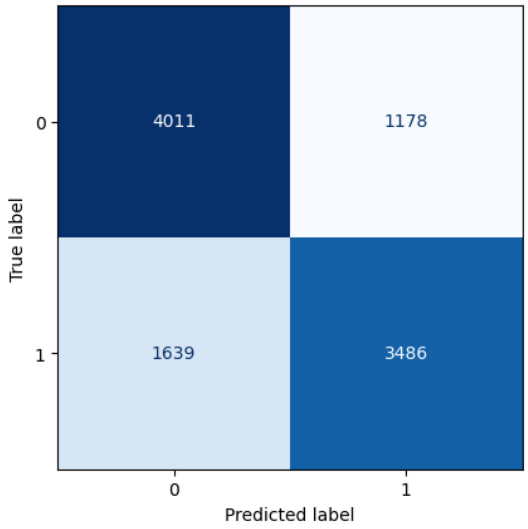


Fig. 11. Confusion Matrix over test data. Model: Logistic regression with implemented lasso regularization.

VIII. Discussion

Regarding the AUC, the models demonstrated relatively similar results (between 0.78 and 0.80). The order from best to worst was as follows: Random Forest, Logistic Regression SFS and Logistic Regression Lasso and Decision Tree having identical results. Performance is undoubtedly crucial but must be balanced with good explainability. In this case, since the performance metrics were quite similar, the choice of models should be made primarly based on their interpretability. However, the recurrent question still remains *what constitutes a good explanation?*. There is currently no absolute truth to this, and we believe there never will be. As stated in *Components of an explanation* III-C, the core components of an explanation are the *what*, *how*, and *whom*, and the answer to what constitutes a good explanation or interpretability depends on these components. The relevant *what*, *whom* and *how* for

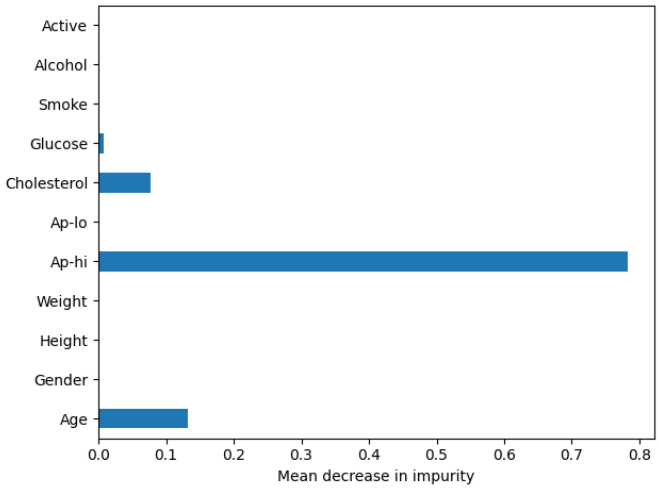


Fig. 12. Feature Importance from decision tree.

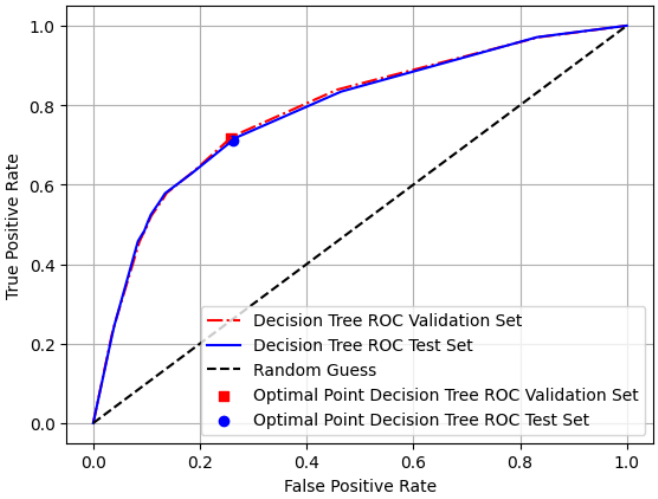


Fig. 13. ROC-curve over test- and validation data. Model: Decision tree.

the models in this report was:

- Content type (what): Global explanation (explanation of the model as a whole), and local explanation (explanation for individual patients).
- Communication (how): Text and visualization
- Target group (whom): AI-developers and medical staff with prerequisites for novice AI-development.

From the specified target group's perspective there are no questions regarding whether or how the models in this project work, which was motivated by the type of models employed being *Interpretable Models* III-B3. If the models were to be implemented, they could be done so with a strong sense of trust, causality, ethics and equality, informativeness and adjustment (see the description of these topics under *Background* II). Although this is the case for all the models in this project, there were still significant differences between them regarding feature importance, which should be considered before implementation. Firstly, the feature importance between the Random Forest/Decision Tree and Logistic Regression is not directly comparable due to their definitions, see Sec. V. Secondly,

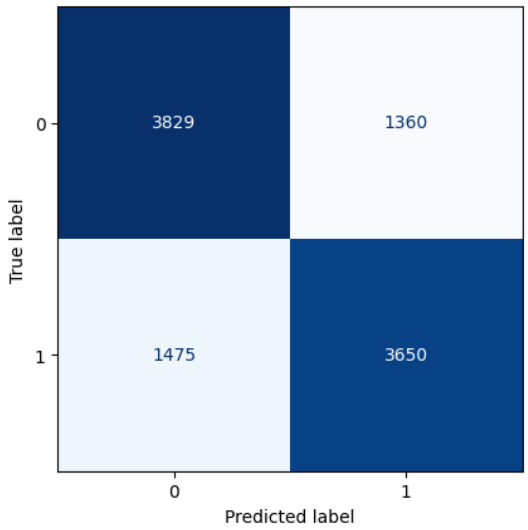


Fig. 14. Confusion Matrix over test data. Model: Decision tree.

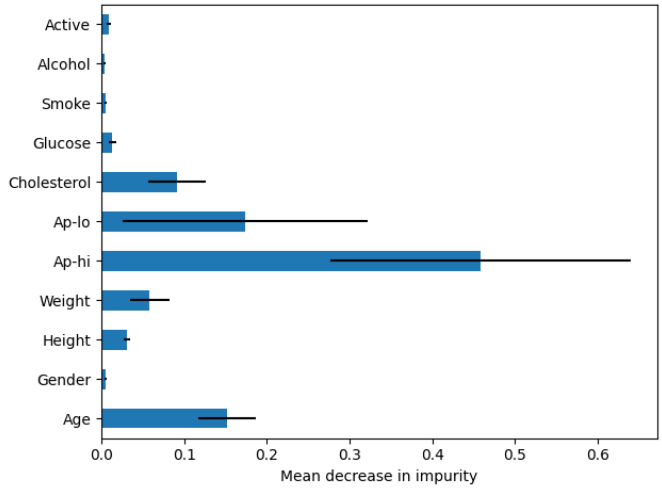


Fig. 15. Feature Importance from random forest. The black lines are displaying the variance for decrease in impurity.

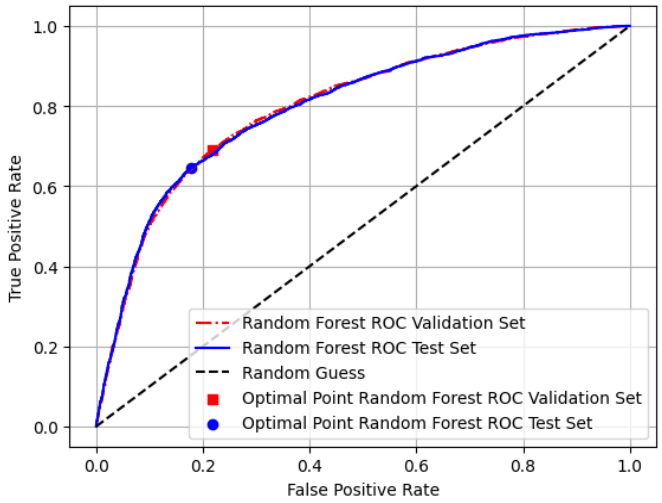


Fig. 16. ROC-curve over test- and validation data. Model: Random forest.

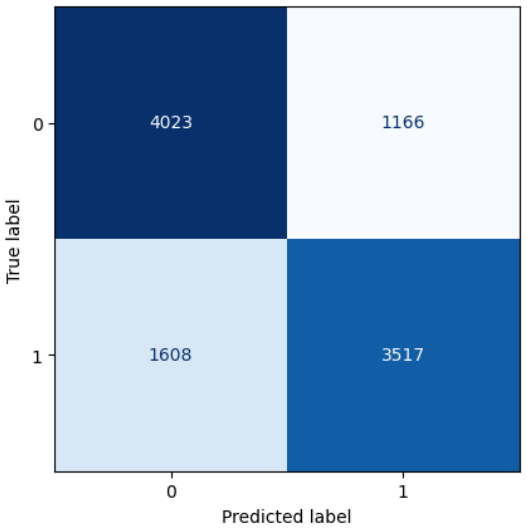


Fig. 17. Confusion Matrix over test data. Model: Random forest.

the intriguing difference between Logistic Regression SFS and Logistic Regression Lasso, was the difference in feature selection. While they both showed a big impact on cholesterol and being active, SFS made decisions based on fewer features, and also had a higher AUC, making it a better choice for implementation in this case. Thirdly, both the Random Forest and the Decision Tree generated the highest impact on prediction based on Ap-hi (systolic blood pressure). Although the Random Forest had a higher AUC, the Decision Tree selected far fewer features, which is desirable for explainability. The visual interpretation of how the prediction is made is also much easier to follow in a Decision Tree, as looking at one tree rather than many is more preferable and easier to understand, see examples in Fig 18 and Fig 3. We would therefore suggest the Decision Tree as the better choice for implementation between the two in this case.

A. *Potential improvements with choice of regularization parameters.*

An important aspect to mention regarding the Logistic Regression is that both SFS and Lasso yielded different feature selections with different regularization (values for parameter C). We discussed our reasoning behind our parameter choices in the results Sec. VII, but this could potentially have been further improved and tuned towards real-life implementation with the assistance of a heart disease expert. To be clear, different regularization resulted in different feature selections. The answer to which feature selection would be most desirable in terms of interpretation should be consulted with the expert before determining what model to use.

B. *Future studies*

The focus of this project has been to study and analyse the explainability of interpretable by design ML models. The ML models were trained on data resembling healthcare patient data with the purpose of evaluating the feasability of implementing

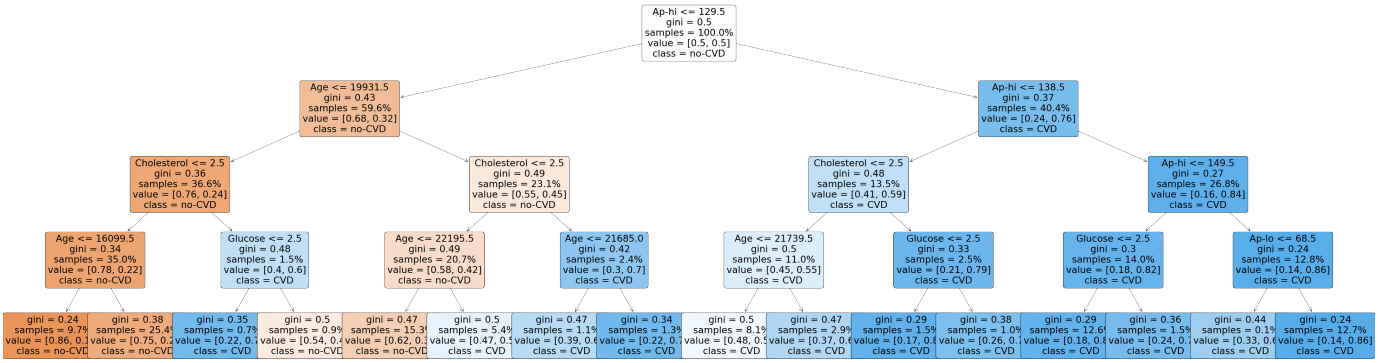


Fig. 18. Trained decision tree with a depth of four layers.

such ML models in a healthcare environment. A key component to this is good data. For future studies we would strive to acquire more data with more numerical features. Instead of having binary subjective features it would be better and more detailed if all input features are objective measurements as this would better distinguish between patients who for example smoke a cigarette a day or two packs a day. Currently both these patients are classified as only smokers (1) even if their behaviour and risks are significantly different.

An area of future study where the need for explainability has become more self evident over the time of writing this article are models that are not interpretable by design. More specifically deep learning neural networks of different forms and complexity. Neural networks are able to be trained more thoroughly and have greater adaptation to input features. These networks have more practical applications as they are known to have much greater accuracy, learn more complex connections, and can perform vastly more complex tasks compared with simpler ML models studied in this article [1]. An example of this is computer vision and classification of complex images of diseases that pose a greater engineering challenge but, if successful, can greatly aid doctors and healthcare institutions. Healthcare implementations of neural networks could result in faster diagnostics of disease but also better aid in educating doctors and help them perform at their fullest potential. All of this, hopefully, directly benefiting patients health and lives. Current techniques that we would like to study are local surrogate models that aim to approximate neural networks and through that introduce a form of interpretablility. Local Interpretable Model-Agnostic Explanations (LIME) shows great promise in this regard.

Another interesting area that has so far not been discussed or even mentioned are the tolerances and trade offs that hospitals and healthcare would be willing to accept in order to implement ML models in their practices. Some questions of discussion or study would be: Do the ML models need to perform better than human doctors and nurses?, If so, by how much?, Is performance or explainability more important?, How much interpretability and explainability is necessary? Do we need to be able to track the decision-making process for each patient data point or simply understand the general reasoning of the model?, Which areas of healthcare can ML models be implemented in and which are strickly forbidden?

A last area of study that essentially has the final say in regards to implementing ML models in healthcare are patients trust and acceptance. Are patients willing to be diagnosed by an ML algorithm? How can a ML models explainability and interpretablility be explained to a patient without a background and zero knowledge of machine learning?

IX. Conclusion

The aim of this project was to analyse the *Explainability* of different machine learning algorithms when applied to a healthcare setting. The major experiments that were implemented consisted of different feature importance methods and anlysis. The results found that different ML algorithms and feature importance methods resulted in different emphasis of features even when trained on identical data. The most interesting result being the fact that many features could be omitted without great losses in predictive accuracy. The remaining features being those that we often correlate cardiovascular diseases with such as high blood pressure and cholesterol values. This is quite exciting as it proves the possibility of consolidating more complex models into ones that are easier to understand yet still effective. Future studies on the subject of *Explainability* include gathering better numerical data and examining Neural Networks.

Acknowledgment

The authors would like to sincerely thank their supervisor Ragnar Thobaben for his steady support, guidance and counselling along with his curiosity invoking attitude towards the subject of machine learning and its implementations. This has not only been vital for the project but also helped cultivate a greater interest and passion for the subject which we are grateful for.

References

[1] Sara Brown. (2021, Apr) Machine learning, explained. Massachusetts Institue of Technology. [Online]. Available: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

[2] L.-V. Herm, K. Heinrich, J. Wanner, and C. Janiesch, "Stop ordering machine learning algorithms by their explainability! a user-centered investigation of performance and explainability," *International Journal of Information Management*, vol. 69, p. 102538, Apr 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S026840122200072X

[3] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *Journal of Artificial Intelligence Research*, vol. 70, pp. 245–317, Jan 2021. [Online]. Available: https://doi.org/10.1613%2Fjair.1.12228
[4] National Institute of Diabetes and Digestive and Kidney Diseases. (2016, Dec) Symptoms causes of diabetes. Bethesda, Maryland. [Online]. Available: https://www.niddk.nih.gov/health-information/diabetes/overview/symptoms-causes
[5] L. A. Celi, *Leveraging Data Science for Global Health*, 1st ed. Cham: Springer Nature, 2020.
[6] Wikipedia. (2023, Mar) Decision tree learning. [Online]. Available: https://en.wikipedia.org/wiki/Decision_tree_learning
[7] ——. (2023, Apr) Entropy(information theory). [Online]. Available: https://en.wikipedia.org/wiki/Entropy_(information_theory)
[8] ——. (2022, Sep) Information gain (decision tree). [Online]. Available: https://en.wikipedia.org/wiki/Information_gain_(decision_tree)
[9] scikit-learn. (2023, Apr) sklearn.tree.decisiontreeclassifier. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier.feature_importances_
[10] Towards AI. (2022, Jan) Why choose random forest and not decision trees. [Online]. Available: https://towardsai.net/p/machine-learning/why-choose-random-forest-and-not-decision-trees
[11] Svetlana Ulianova. (2019, May) Cardiovascular disease dataset. [Online]. Available: https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset
[12] Centers for Disease Control and Prevention. (2021, May) High blood pressure symptoms and causes. [Online]. Available: https://www.cdc.gov/bloodpressure/about.htm
[13] scikit-learn. (2023, Apr) sklearn.model_selection.train_test_split. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
[14] NumPy. (2023, Mar) numpy.array. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.array.html
[15] scikit-learn. (2023, Mar) Machine learning in python. [Online]. Available: https://scikit-learn.org/stable/