Interpretability of Machine Learning Models on Medical Tabular Data

Rubinić, Ivan

UNIVERSITY OF RIJEKA
FACULTY OF ENGINEERING
Graduate University Study of Computing

Master's thesis

Interpretability of Machine Learning Models on Medical Tabular Data

Rijeka, September 2023.

Ivan Rubinić
0069085793

UNIVERSITY OF RIJEKA

FACULTY OF ENGINEERING

Graduate University Study of Computing

Master's thesis

Interpretability of Machine Learning Models on Medical Tabular Data

Mentor: prof. dr. sc. Ivan Štajduhar

Rijeka, September 2023.

Ivan Rubinić
0069085793

Umjesto ove stranice umetnuti zadatak
za završni ili diplomski rad

Statement on independent creation of thesis

I declare that I created this thesis independently.

Rijeka, September 2023.

____

Ivan Rubinić

Acknowledgements

I thank my mentor, prof. dr. sc. Ivan Štajduhar for all the help provided regarding this thesis. I also thank asist. dr. sc. Franko Hržić for all the help, guidance, many meetings and numerous advices. I'd also like to thank my family and friends for support during the studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the past several years, machine learning techniques have been incorporated into the healthcare sector at an impressive rate. Whether the mentioned techniques predict medical conditions, time to recovery or give out the best medical procedures for treating patients [3, 4, 5], one of their key requirements is transparency, trust and prediction reasoning [6]. There are a lot of state-of-the-art machine learning models that show excellent results in aforementioned tasks, but lack interpretability or are not explainable, which, in turn, hinders the mentioned requirements [7].

This thesis evaluates interpretability and explainability methods on the case study of detecting issues related to appendicitis. The goal is to research machine learning models that could be useful for this case study, as well as to research which techniques are applicable in order to achieve greater levels of explainability. The motivation behind each machine learning model's selection, as well as its advantages and disadvantages have to be described. After that, detailed analysis of the used data has to take place and the selected models have to be fitted to the mentioned data. Lastly, appropriate explainability techniques need to be applied and discussed.

In order to evaluate different machine learning algorithms, as well as to make the recommendations, a medical tabular dataset provided by Medical University of Graz, Department of Radiology was used. The dataset consists different features related to appendicitis and the outcome that needed to be predicted – preoperative diagnosis for appendicitis. The dataset itself is further discussed and analyzed in subsection

2.1 on page 5.

Once the models' performance was evaluated, different explainability techniques were utilized to produce human-understandable reasoning behind their predictions.

1.1 The importance of interpretability and explainability

In the era of using ever more complex machine learning models in ever broader scope of tasks, a need for providing insight into models' inner workings and its decision has significantly risen [8].

Interpretability in a model implies that these models' internal logic and inner workings are understandable to humans which makes it possible to verify, interpret and understand the reasoning of the system and how a particular decision was made – without any additional methods that provide such interpretations [9]. For example, decision trees are interpretable – one can exactly follow the model's train of thought. In contrast, convolutional neural networks are not interpretable – one can not understand why the model predicted a certain outcome. Model's interpretability is usually inverse to its complexity, as shown in figure 1.1.

On the other hand, explainability aims to take a machine learning model, explain its behaviour and reveal the connection between input data and model outputs [10]. For instance, how one input feature is related to the final model's prediction. Model's predictions can be explained using various explainability techniques, some of which are model-agnostic – applicable to any machine learning model. Certain model-agnostic explainability techniques are described and demonstrated in chapter 2.

Interpretability and explainability can help with increasing the transparency, accountability, trust, understanding and prediction reasoning in machine learning models. These perks are especially instrumental in medicine and healthcare to ensure that the recommendations made by intelligent systems are sound, correct, justifiable and enable the care providers to make better decisions and infer new knowledge, insights or discovery [11].

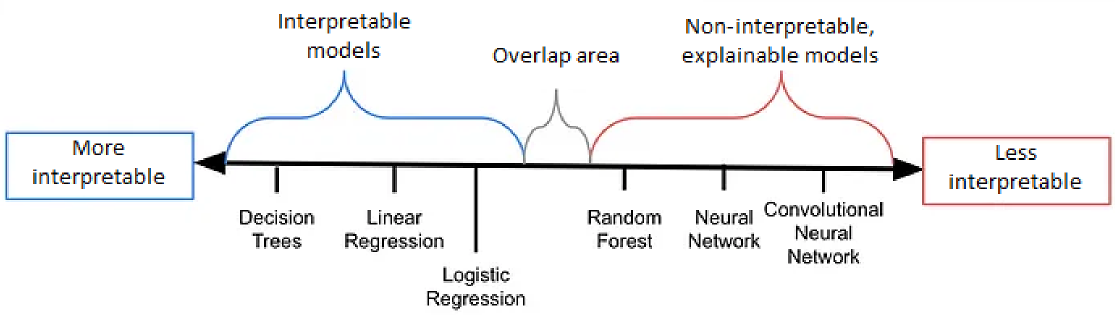


Figure 1.1 Model interpretability versus model complexity. One can notice that as model complexity increases, its interpretability decreases. There is an overlap area between interpretable and non-interpretable models since the judgement on the interpretability can be unclear for some models. It is important to note that non-interpretable models can still be explained using explainability techniques [1].

1.2 Related Work

Since the main aim of the task studied in this thesis is classifying patients' data into appendicitis categories from histological and postoperative viewpoints, similar and related work was researched. The following paragraphs deliver short descriptions of the mentioned.

A Diagnostic Testing for People with Appendicitis using Machine Learning Techniques paper focuses on a similar topic. The paper presents a machine learning (ML) technique to predict appendix illness, i.e. whether it is acute or subacute. The authors compared predictive results of logistic regression, naive Bayes, generalized linear, decision tree, support vector machine, gradient boosted tree and random forest machine learning models [12].

The Use of Machine Learning Approaches for the Diagnosis of Acute Appendicitis paper aims to develop an easy, fast, and accurate estimation method for early acute appendicitis diagnosis using machine learning algorithms [13].

Application of Machine Learning to the Prediction of Postoperative Sepsis after Appendectomy paper focuses on applying various machine learning algorithms to a large national dataset to model the risk of postoperative sepsis after appendectomy. They found that machine learning methods can be used to predict the development of sepsis after appendectomy with moderate accuracy [14].

A novel and simple machine learning algorithm for preoperative diagnosis of acute appendicitis in children paper used machine learning algorithms to detect appendicitis and diferentiate uncomplicated from complicated cases [15] which is very similar to the topic of this paper.

Using Machine Learning to Predict the Diagnosis, Management and Severity of Pediatric Appendicitis paper investigated the use of machine learning (ML) classifiers for predicting the diagnosis, management and severity of appendicitis in children [16].

Machine learning prediction model for postoperative outcome after perforated appendicitis developed and validated a machine learning prediction model for postoperative outcome of perforated appendicitis [17].

Acute appendicitis in the elderly: risk factors for perforation aims to identify the risk factors of perforation in elderly patients who presented with acute appendicitis [18].

Other related work includes early detection of appendicitis using smart wearables [19], developing a scoring system to distinguish uncomplicated from complicated appendicitis [20] and big data in medicine [21].

However, whilst most of the mentioned related works deal with machine learning algorithms in the field of medicine and healthcare, it is important to note that none of then focus on explainabiltiy and interpretability which is the key topic of the research conducted in this tehsis.

Chapter 2

Methodology

As it was mentioned before, this paper includes work with different machine learning algorithms and appropriate explainability techniques in order to highlight the most favourable algorithm in this use-case. This chapter brings details and analysis of the used dataset, algorithms and explainability techniques.

Prior to applying the machine learning algorithms to the selected medical problem, all of the algorithms were tested on exemplary tasks ensuring the proper implementation of the selected algorithms and explainability methods.

2.1 Dataset

Initial dataset provided by Medical University of Graz, Department of Radiologly had 9 features, 787 instances, and 4 different columns of output predictions (`Histology Report`, `Histology Report Binary`, `Postoperative Diagnosis`, `Postoperative Diagnosis Binary`). Four instances of dataset were made, with respect to their output predictions which are described in more detail in subsection 2.1.1. It is important to note that all of the dataset instances were significantly imbalanced which made classification process much more challenging.

All instances in the initial dataset which had missing values have been removed. Thus, the resulting dataset consisted of 696 instances. The majority of removed instances had `Diameter Appendix [mm]` values missing.

2.1.1 Dataset instances

Available data was used to create four different datasets:

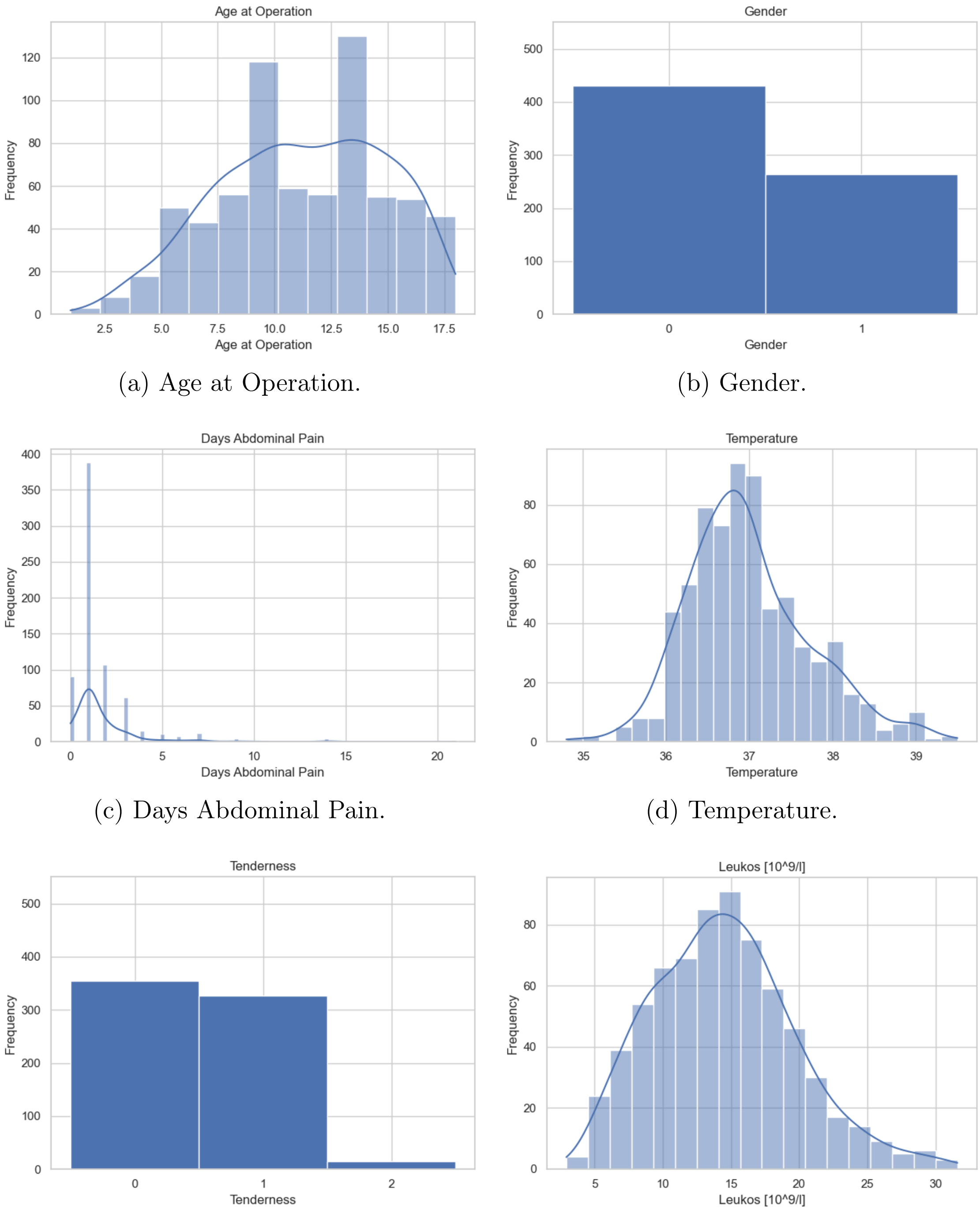
1. histology binary classification dataset (**HBD**) – whether the appendicitis is present from a histological viewpoint,
2. histology classification dataset (**HD**) – which type of appendicitis (no appendicitis, simple appendicitis, complex appendicitis) is present from a histological viewpoint,
3. postoperative binary classification dataset (**PBD**) – predicting postoperative diganosis to simple and complex,
4. postoperative classification dataset (**PD**) – predicting postoperative diganosis in three categories:
    (a) simple,
    (b) complex and complex with abscess,
    (c) perforated and gangranos.

The idea behind dividing the initial dataset into four different datasets is making the eventual classifier's task easier. Instead of predicting both postoperative diagnosis and histology report simultaneously, those predictions were set apart – making two separate datasets (HBD and PD). Each of these multinomial classification datasets were then intepreted as binary classification datasets and adapted to binary classification regarding their desired output values (HBD and PBD) – making a total of four datasets.

2.1.2 Dataset distribution

In order to undestand feature values and output classes distribution, histograms of each feature is provided in figure 2.1.

Multiple imbalanced output classes can be noticed from the given histograms. It can be observed that all output classes, displayed in subfigures 2.1j, 2.1k, 2.1l and 2.1m, have highly imbalanced number of instances per class which substantially compromises the learning process, since most of the standard machine learning algorithms expect a balanced class distribution [22]. This imbalance can be a problem since most classification algorithms pursue to minimize the error rate over all classes, i.e. they ignore the difference between types of misclassification errors [23]. They implicitly assume that all misclassification errors cost equally [23].



(a) Age at Operation.

(b) Gender.

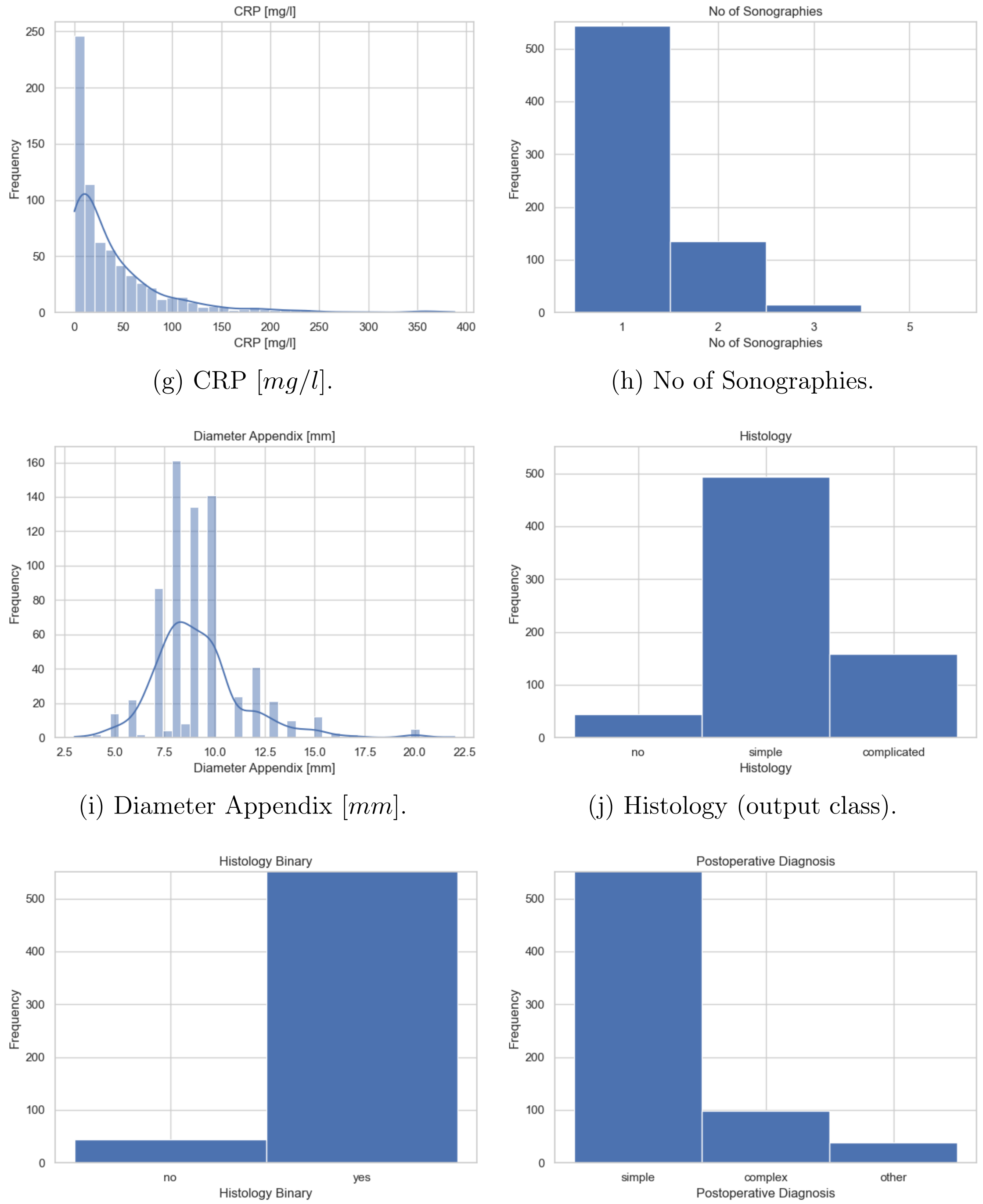
(c) Days Abdominal Pain.

(d) Temperature.

(e) Tenderness.

(f) Leukos $[10^9/l]$.

Figure 2.1 Dataset features histograms – each feature's values are visualized through the use of histograms.



(g) CRP $[mg/l]$.

(h) No of Sonographies.

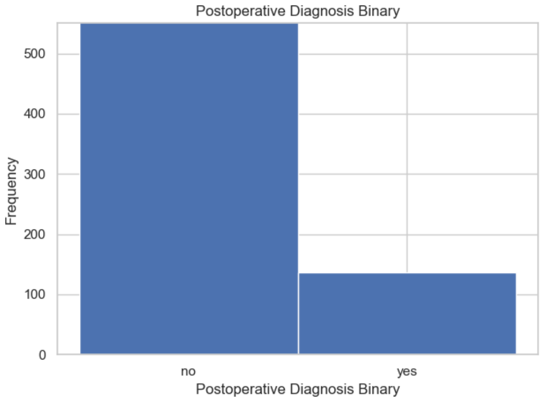
(i) Diameter Appendix $[mm]$.

(j) Histology (output class).

(k) Histology Binary (output class).

(l) Postoperative Diagnosis (output class).

Figure 2.1 Dataset features histograms – each feature's values are visualized through the use of histograms.

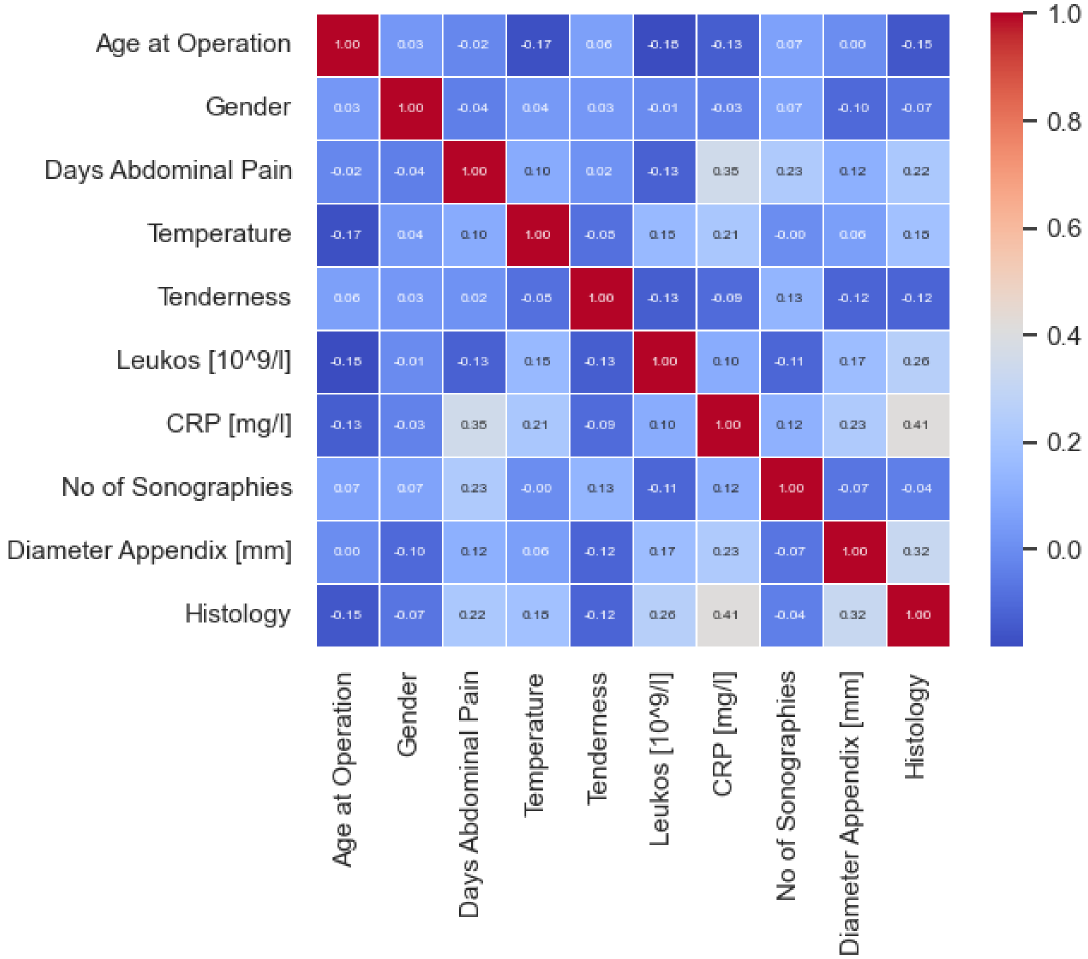


(m) Postoperative Diagnosis Binary (output class).

Figure 2.1 Dataset features histograms – each feature's values are visualized through the use of histograms.

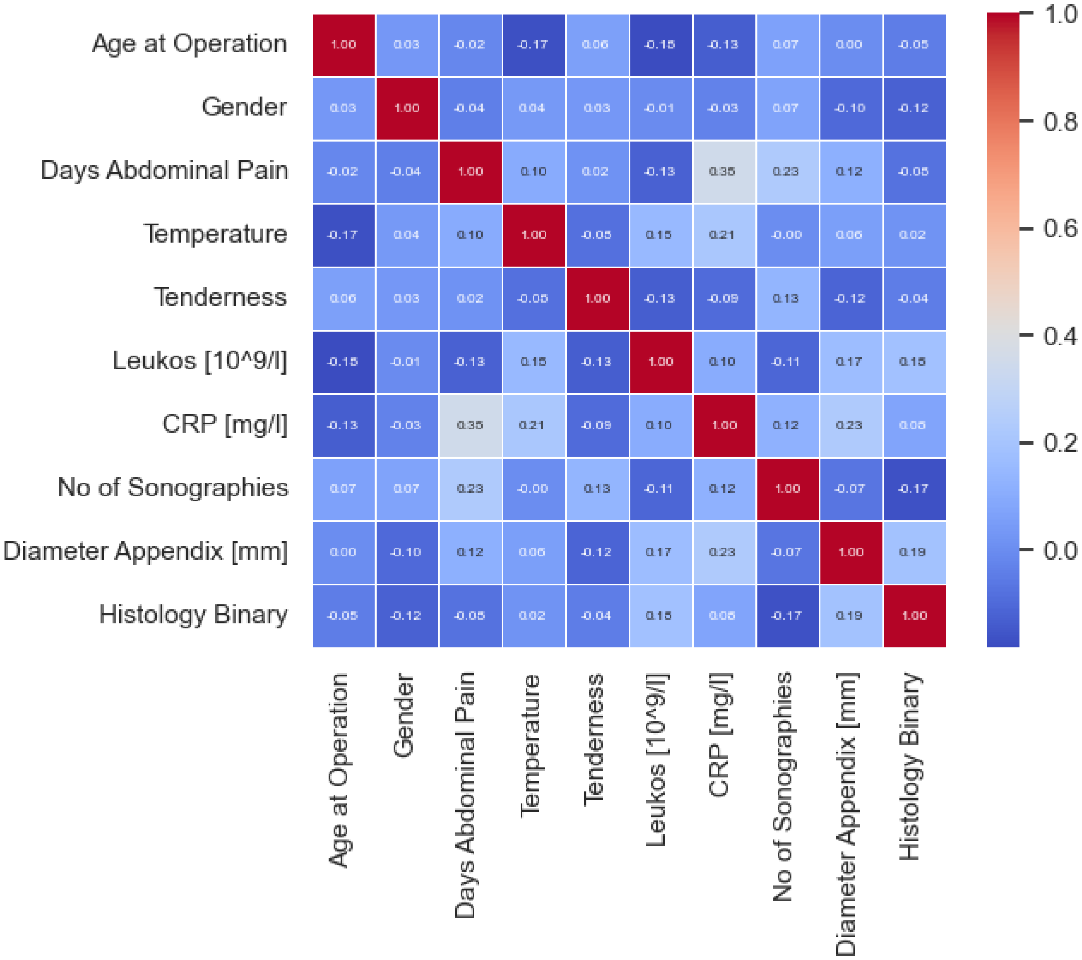
2.1.3 Correlation

In order to decide which explainability techniques are valid to use on this dataset, correlation between features and appropriate output predictions must be looked into. Thus, a correlation matrix for each instance of the dataset was created and is shown in figure 2.2.



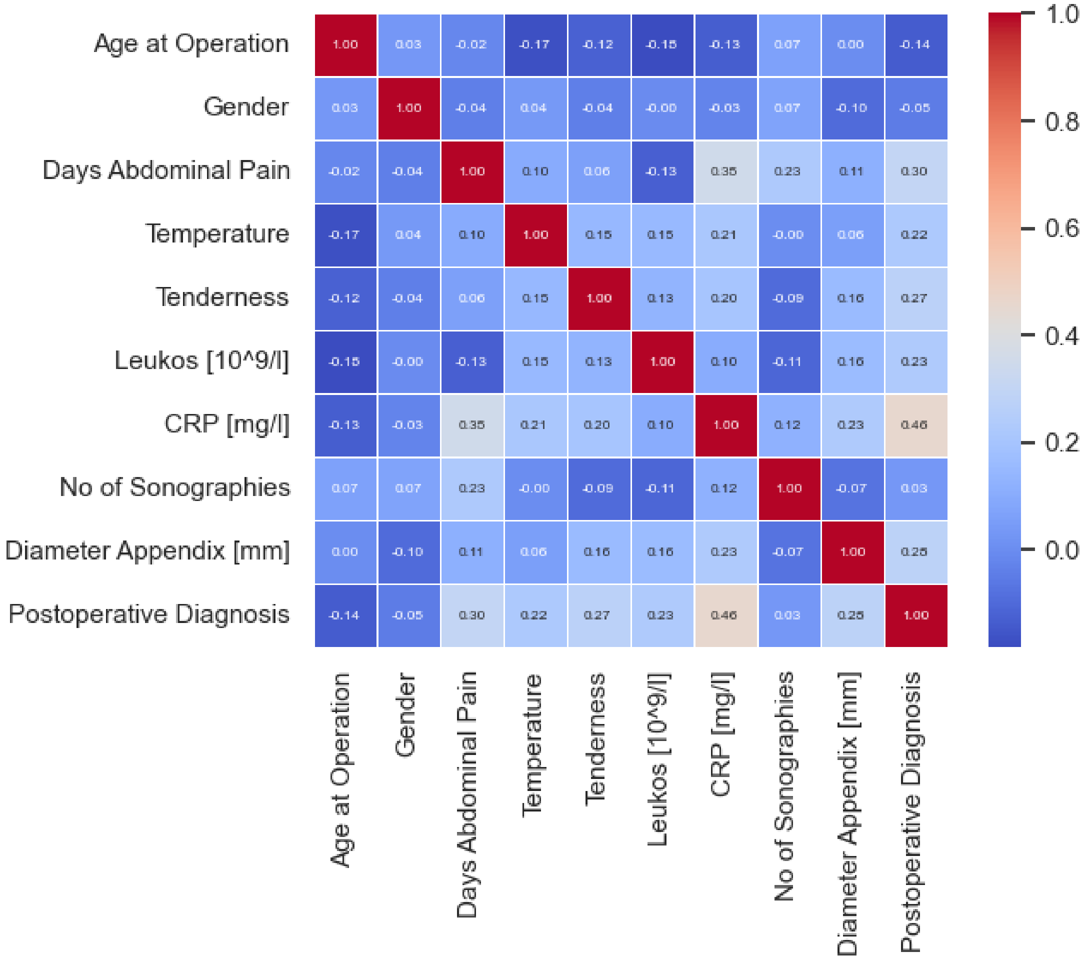
(n) Histology report dataset.

Figure 2.2 Dataset features and output correlation. Correlation for each combination of features is shown. No significant correlation can be noticed.



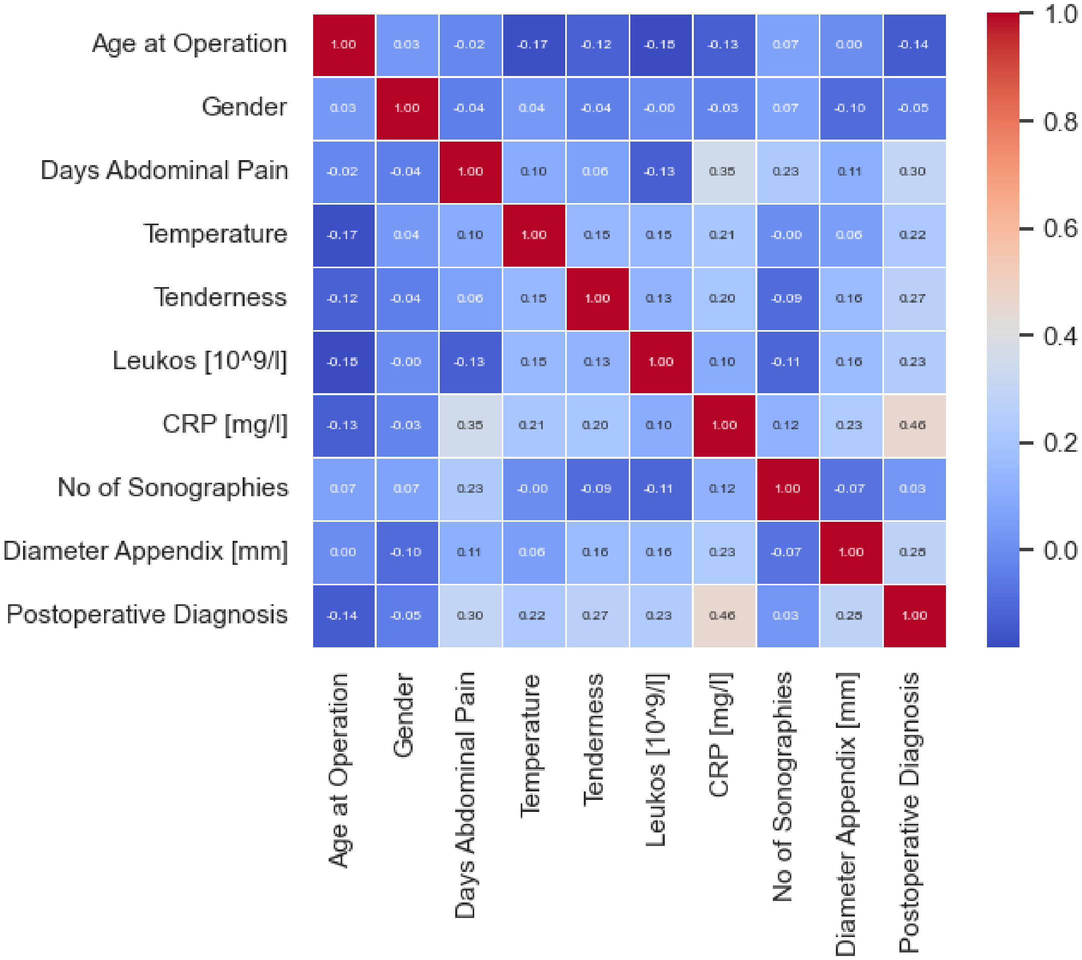
(o) Binary histology report dataset.

Figure 2.2 Dataset features and output correlation. Correlation for each combination of features is shown. No significant correlation can be noticed.



(p) Postoperative diagnosis dataset.

Figure 2.2 Dataset features and output correlation. Correlation for each combination of features is shown. No significant correlation can be noticed.



(q) Binary postoperative diagnosis dataset.

Figure 2.2 Dataset features and output correlation. Correlation for each combination of features is shown. No significant correlation can be noticed.

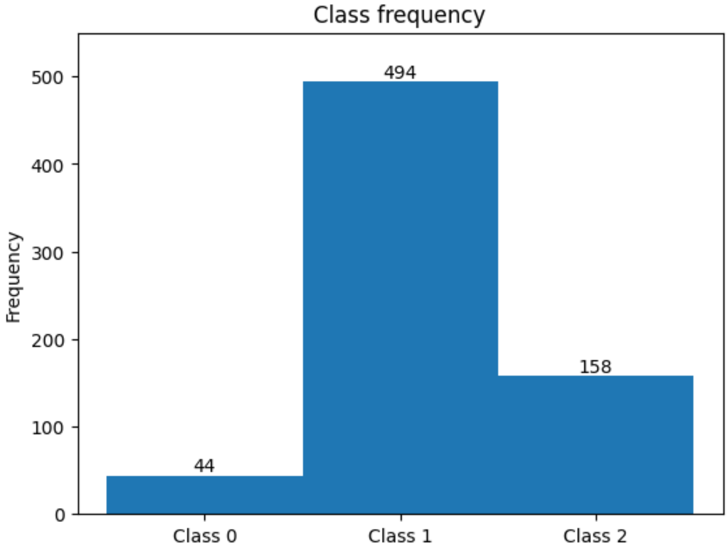
As the correlation coefficients range from negligible correlation to low levels of moderate correlation [24], we suppose there is no significant correlation between features and the output and features internally that could hamper the explainability techniques' reliability.

2.1.4 Oversampling

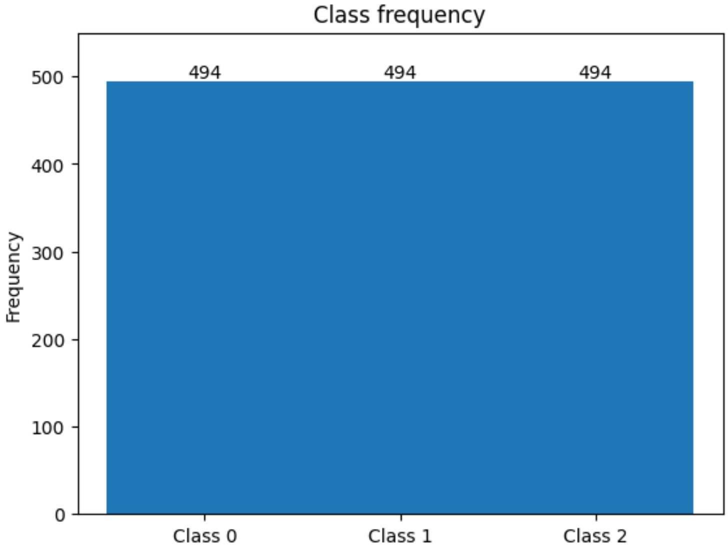
Due to the imbalanced dataset, oversampling on each of the available versions of datasets was also conducted. Only neural network model used such, oversampled dataset in its training due to lack of noticeable changes in performance metrics with other models when using this technique. Oversampling is a technique that adjusts the ratio between the different classes [25]. By performing oversampling on the dataset, existing minority samples were replicated in order to increase the size of the minority class [25].

Figure 2.3 illustrates the effect of the used oversampling method. Duplicates of samples of classes 0 and 2 were appended to the dataset while the frequency of their respective classes leveled with the most frequent class (in this case, class 1). Duplicates were selected sequentially meaning that all the samples were duplicated if the class did not become more frequent than the most frequent class after the duplication. If that was not possible, number of appended duplicates was $MFC_{freq} - CC_{freq}$ where MFC_{freq} and CC_{freq} are the frequencies of the most frequent and current classes respectively.

Oversampling was only conducted in combination with neural networks and only on their respective training data subsets [26, 27]. Validation and test data was not oversampled.



(r) Class frequencies of the initial dataset.



(s) Class frequencies of the oversampled dataset.

Figure 2.3 Oversampling illustration on HD. One can notice that the oversampled dataset has perfectly balanced number of classes' samples due to replication of existing samples of less frequent class.

2.1.5 Prototypes and criticisms

Prototypes and criticisms is an explainability technique that is used to calculate and visualize the dataset's prototypes and criticisms.

While a prototype should be and instance that is representative of all the data, a criticism should be its counterpart – an instance that is not well represented by the set of prototypes. The purpose of this technique is to provide insights to the data, especially the criticisms which are interesting for further analysis [2].

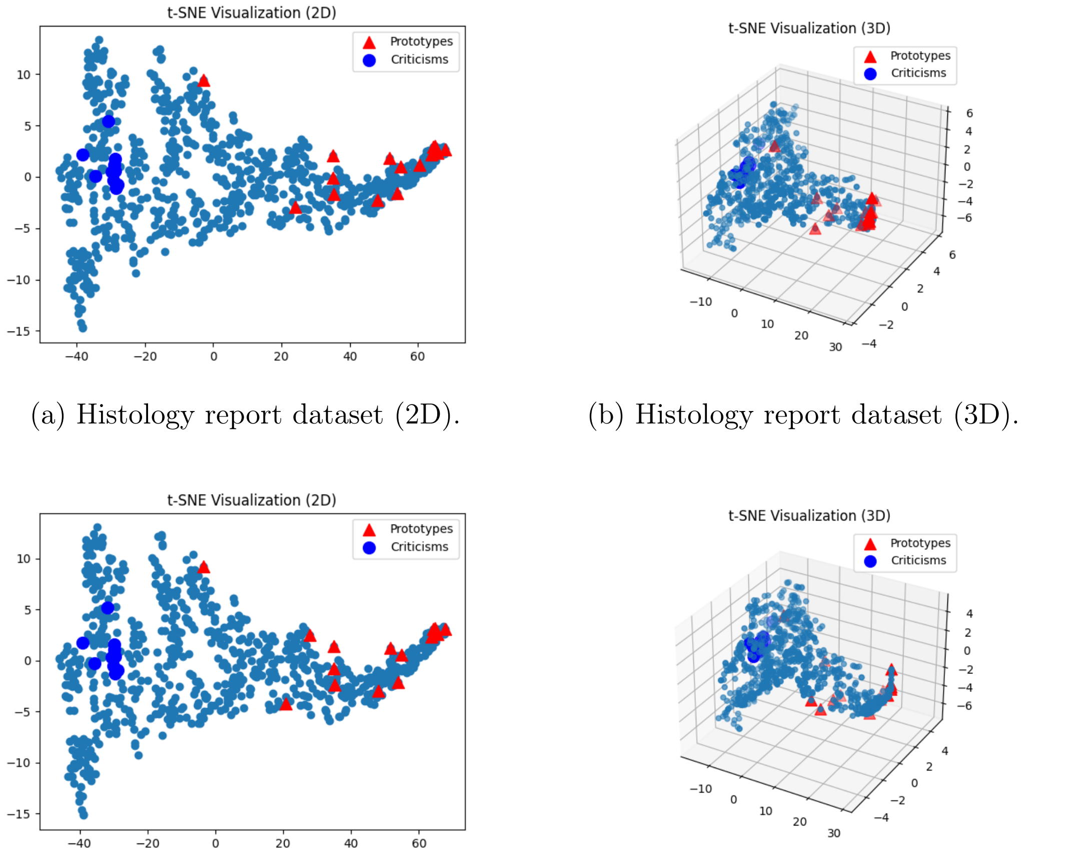
Prototypes and criticisms on the available datasets were calculated with different sets of hyperparameters displayed in table 2.1. The clearest visualizations came when parameter `gamma` experimetntaly determined to be 0.2, number of prototypes to 15 and number of criticisms to 10.

It is important to note that some criticisms and some prototypes may seem oddly chosen. Namely, as humans are able to easily perceive only up to three dimensions, the initial nine-dimensional data had to be put through dimensionality reduction process. The process consisted of applying the t-SNE technique that visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map whilst trying to preserve as much of the significant structure of the high-dimensional data as possible in the low-dimensional map [28].

To return to the previously stated observation – the choice of prototypes and criticisms may seem odd. The reason for it may lay in the dimensionality reduction process. Whilst the process tries to preserve as much of the structure of the data that is present in the data's own, initial hyper-dimensional space, some structure is bound to degrade as the dimensionality is reduced. That is why some prototypes and criticisms visualizations make much more sense in 3D than 2D displays – for example, figures 2.4e and 2.4f.

Hyperparameter	Used values in analysis
`gamma`	[0.2, 0.4, 0.6, 0.8]
`n_prototypes`	[5, 10, 15, 20]
`n_criticisms`	[5, 10, 15, 20]

Table 2.1 Analysed parameters during prototypes and criticisms calculation. The best combination of parameters was chosen through inspecting the visualizations created.



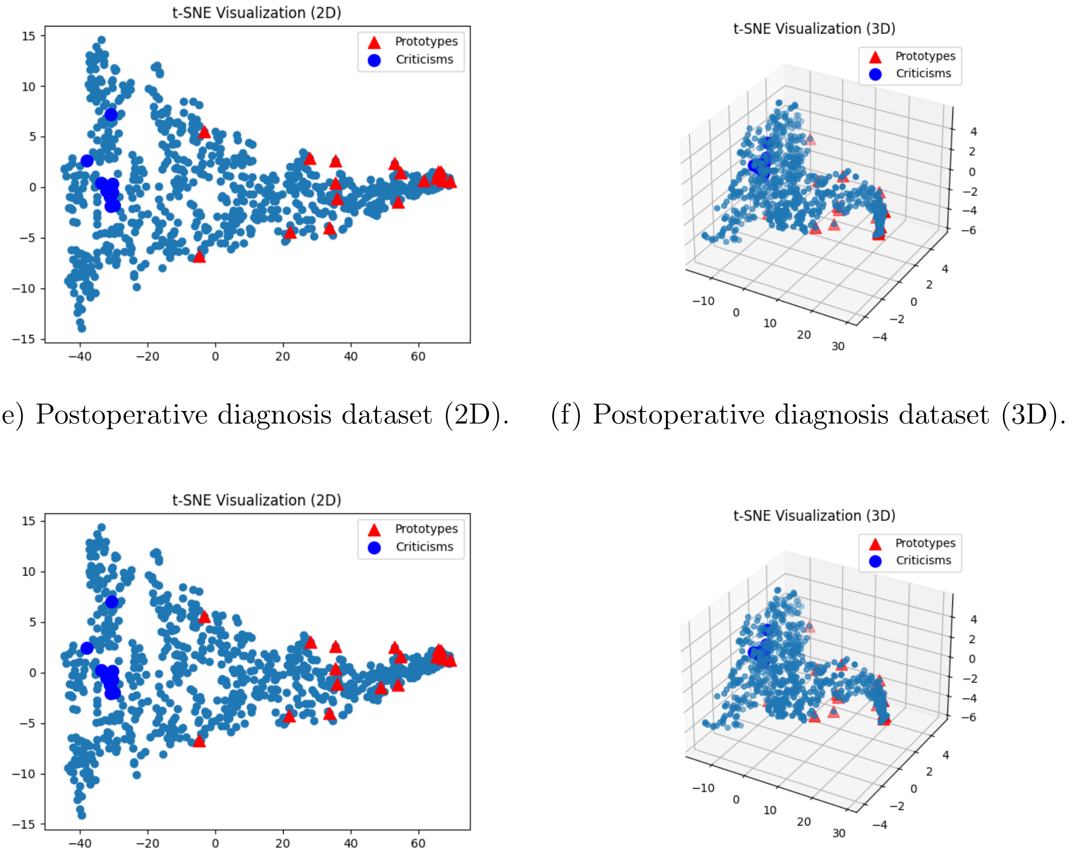
(a) Histology report dataset (2D).

(b) Histology report dataset (3D).

(c) Binary histology report dataset (2D).

(d) Binary histology report dataset (3D).

Figure 2.4 Generated visualizations for prototypes and criticisms. Prototypes are located in areas where there is higher density of data samples since they focus on describing the most data in the best way possible. Criticisms, on the other hand, are usually located in less densely populated data space.



(e) Postoperative diagnosis dataset (2D).

(f) Postoperative diagnosis dataset (3D).

(g) Binary postoperative diagnosis dataset (2D).

(h) Binary postoperative diagnosis dataset (3D).

Figure 2.4 Generated visualizations for prototypes and criticisms. Prototypes are located in areas where there is higher density of data samples since they focus on describing the most data in the best way possible. Criticisms, on the other hand, are usually located in less densely populated data space.

2.2 Utilized machine learning algorithms

In this chapter, the utilized machine learning algorithms will be preseneted along with their advantages and disadvantages. Algorithms taken into account are:

1. decision tree,
2. OneR,
3. sequential covering,
4. random forest and
5. neural network (multilayer perceptron).

2.2.1 Decision tree

Decision trees are one of the most popular interpretable machine learning models. Due to their interpretability, the decision-making process is completely transparent which is important in high-risk environment such as medicine. A simple decision tree is shown in figure 2.5 in order to illustrate how a prediction is made in a decision tree.

The decision tree in this experiment was trained using the popular CART (Classification and Regression Tree) algorithm. The algorithm splits the available data based on feature k and its threshold t_k (in figure 2.5, k is *temperature*, and t_k is 5). Parameters k and t_k are chosen in a way so that the newly created subsets of initial data are as homogeneous as possible, i.e. have the lowest impurity. After that, the algorithm recursively repeats the process for both subsets of data. The process goes on until the stopping criteria is met.

This experiment used scikit-learn's implementation of the decision tree – DecisionTreeClassifier. It made training process very straightforward and the output model was easy to visualize via the built-in plotting functions. The parameters for maximum depth and minimum samples in each leaf node of the resulting tree were used as stopping criterium that should limit overfitting of the model. To sum up, the main advantages of this model are [29]:

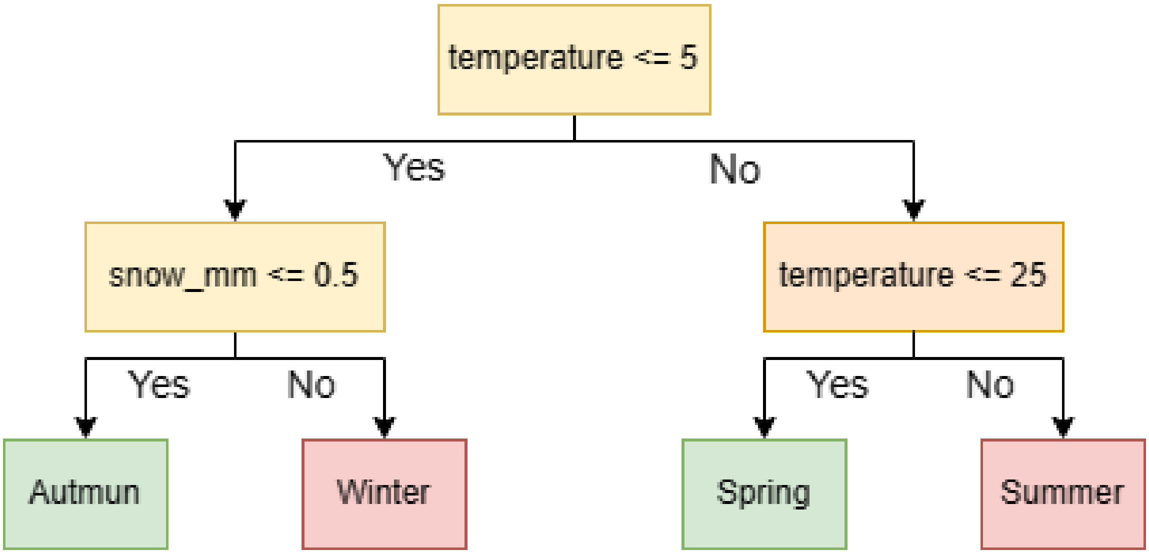


Figure 2.5 Decision tree's prediction making. Suppose a day in which the temperature is 3°C and there is 70 centimeters of snow is used as an input. The decision tree will choose the left subtree of the root node because the temperature is lower than 5°C and then the right leaf node because there is some snow. The input day's season will be classified as winter.

- simple to understand and interpret, easy to visualize,
- require little to none data preparations,
- able to handle multi-output problems – important for the dataset used in this paper,
- possible to validate using statistical tests – possible to account for the reliability of the model,
- compputational cost of training and using the model is low,
- white box – meaning we can interpret its internals as well as its predictions.

On the other hand, the leading disadvantages of this model include [29]:

- overfitting (mechanisms such as minimum number of samples required in a leaf node or maximum tree depth lower the possibility of overfitting),
- predictions are neither smooth nor continuous,

- some concepts are hard to learn,
- can create biased trees if some classes dominate – important for the dataset used in this paper.

To sum up, the decision tree model can achieve satisfying results if trained on appropriate data with appropriate anti-overfitting mechanisms. A trained decision tree is highly interpretable which makes it especially popular among medical scientists – perhaps because it mimics the way a doctor thinks [30].

2.2.2 OneR

OneR, short for One Rule is a decision rule algorithm which is characterized by its high levels of interpretability. It resembles a natural way of making conclusions, predictions and decisions as the algorithm produces IF-THEN statements. OneR selects a feature that holds the most information about the outcome of interest and creates decision rules from that feature [2]. For example, a possible decision rule list learned by this algorithm could be: *IF temperature = high THEN fever; IF temperature != high THEN not_fever.*

The algorithm is simple and can be described in a couple of steps [2]:

1. discretize the continuous features,
2. for each feature:
    - create a cross table between the feature values and categorical outcome,
    - for each value of the feature, create a rule which predicts the most frequent class of the instances that have this feature value,
    - calculate the total error of the rule for the feature,
3. select the feature with the smallest total error.

Basically, the algorithm calculates a rule based on a feature for whose values it can predict the outcomes with the highest accuracy. The produced rules are fully interpretable and transparent. OneR algorithm benefits from such interpretability and simplicity both in training and in use. On the other hand, it does not have the capacity to make predictions with as high level of accuracy as other, more complex algorihms. Despite that, it can often achieve some unexpectedly good results [31].

2.2.3 Sequential covering

Sequential covering is a general procedure that repeatedly learns a single rule to create a decision list that covers the entire dataset rule by rule [2]. It is somewhat more complicated to train and use than the previously described algorithm, OneR. The main idea behind this algorithm is to cover the dataset part by part – first, a learn a rule that covers an adequate part of the data instances, remove them from the training set, and then repeat – until all data instances are covered. The process of learning one rule can vary [32, 33].

More formally, the sequential covering algorithm for binary classifications can be trained in a few steps [2]:

1. initialize an empty rule list
2. learn a rule r – can be computed by calculating the path to the purest decision tree's node or using other similar algorithms,
3. while the positive examples are not yet covered:
    - add rule r to rule list
    - remove all data instances covered by rule r,
    - learn another rule on the remaining data,
4. return the rule list.

The process of training the sequential covering algorithm on binary classification dataset is shown in figure 2.6.

For multi-class classification, the classes are ordered by increasing frequency. The algorithm then starts with the least common class which is considered a positive class in contrast to all other classes which are considered as members of the negative class. After the rule list for the least common class has been calculated, classes with higher frequency get marked as a positive class. This process is also reffered to as one-

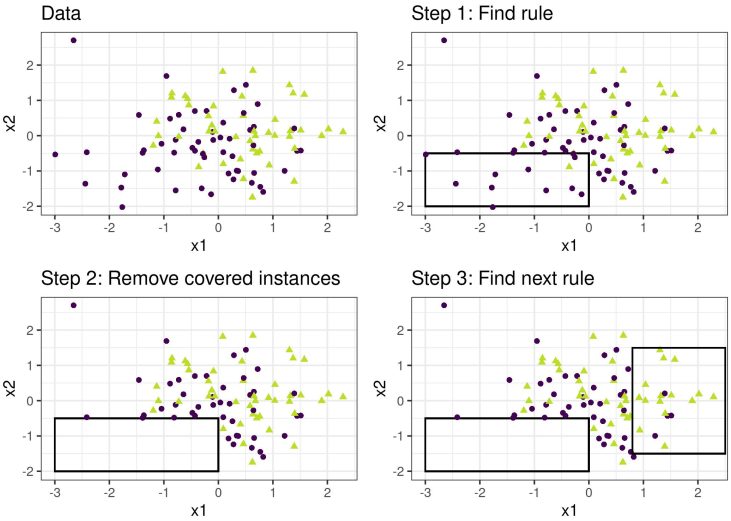


Figure 2.6 Visualization of sequential covering algorithm training on binary classification dataset [2]. One can notice the steps the algorithm takes in one iteration to "cover" some data instances and remove the covered instances in the next iteration.

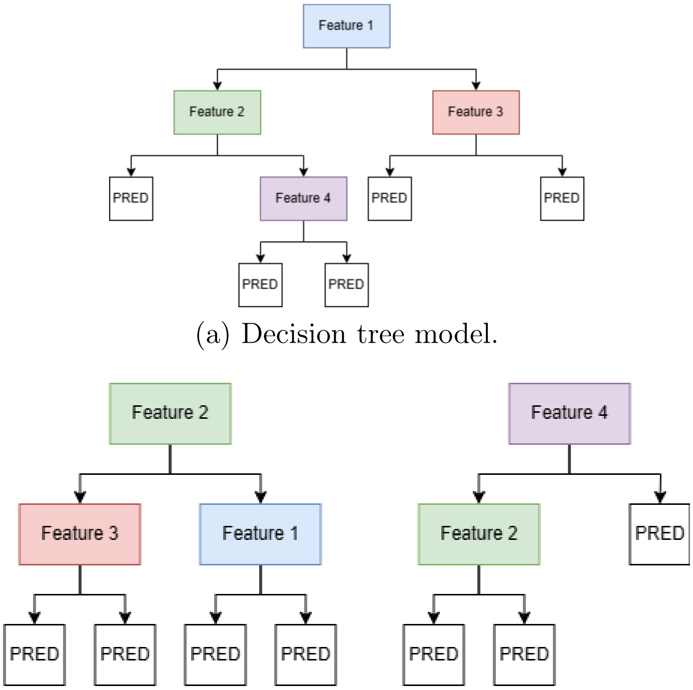
versus-all strategy in classification [2].

2.2.4 Random forest

Random forest is an ensemble model. Its goal is to pool a set of not necessarily optimal predictors instead of seeking to optimize a sole predictor [34]. In other words, forest in random forest comes from a collection of decision trees participating in predictions, and random comes from the fact that each of the trees is assigned a random permutation of the training data.

For example, lets consider a random forest model that consists of k decision trees and is trained on dataset consisting of n instances. Each of k decision trees will be trained on a random sample S_k of the original training data, usually of size n. As each sample taken from the original training set is selected at random, each tree's training set S_k will probably have copies of some instances, while, in contrast, some other instances will not be present at all (known as bagging). Different training sets for each decision tree result in trees that focus on different specifics of the original dataset. Therefore, the trees will use different features to make their predictions.

To sum up, random forests use bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual trees [35]. Figure 2.7 displays the key difference between decision tree and random forest models.



(a) Decision tree model.

(b) Random forest model.

Figure 2.7 Comparison of decision tree and random forest models. Random forest model trained using bagging technique incorporates different feature across its decision trees therefore making a more generalized predictor.

The biggest advantages of random forest model are:

- the ability to work well with both classification and regression tasks [36],
- increased generalization in contrast to decision trees due to its randomness in feature selection over its decision trees,
- ability to handle non-linear relationships [36].

The disadvantages of random forest model include:

- not interpretable – due to usually high number of decision trees, the model's decision path for making a prediction is not understandable,
- it can be computationally intensive for large datasets [36].

2.2.5 Neural network

The last machine learning model used in this experiment is neural network, or, more specifically, multilayer perceptron (MLP). MLP is defined as a fully connected feedforward neural network with at least three layers (input layer, at least one hidden layer and an output layer) [37].

Neural network in general is a model inspired by the human brain and the way it functions. It consists of input and output layers and an arbitrary number of hidden layers. Each layer consists of neurons – activation functions that map weighted inputs to outputs. While hidden layers are connected to their respective previous and next layers, input and output layers are only connected to one, appropriate, hidden layer. Each connection between some layers' neurons has its belonging weight that also serves as a ponder to its neuron [37].

Neural networks are trained through the processes of forward-propagation and backward-propagation. Forward-propagation implies several steps [37]:

1. propagating the input instances from the dataset to generate output values from the neural network,
2. comparing the produced output values with the ground-truth values.

After that, the process of backward-propagation takes place [37]:

1. calculating the error at the output units,
2. backward-propagating the error one layer at a time using gradient descent, until the input layer is reached,
3. updating weights using the calculated gradients and the defined learning rate.

Figure 2.8 shows a multilayer perceptron that has two hidden layers. Input layer consists of three neurons, hidden layers of three and two neurons respectively, and the output layer has one neuron. The network is fully connected, meaning that each neuron from i-th layer is connected to each neuron from $(i + 1)$-th layer.

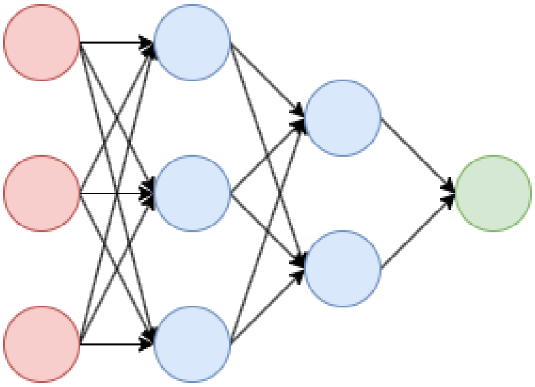


Figure 2.8 Neural network illustration. Input, hidden and output layers are highlighted in red, blue and green colours respectively.

The biggest advantages of neural networks are capability to learn complex, nonlinear models, and to learn models in real-time. Disadvantages includes requirement to tune a number of different hyperparameters (number of layers, neurons, iterations etc.), a possibility for different random weight initializations to lead to different accuracy due to non-convex loss function, possibility of overfitting, and sensitivity to feature scaling [38].

2.3 Utilized explainability techniques on non-interpretable models

2.3.1 Partial dependence – PD

PDP (Partial Dependence Plot) shows the marginal effect of one or two features on the predicted result of the machine learning model [39]. PD plots are thus named because they show how the model's predictions partially depend on values of the input variables of interest [40]. The value of the PD function is determined by averaging the prediction of the machine learning model when the values of the observed feature of all instances in the dataset are replaced by the desired value. The expression which describes the value of the PD function is obtained for a desired value of the observed feature S is $\hat{f}_S(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_S, x_C^{(i)})$ [2], and is usually calculated through all or part of the domain of the observed feature S.

The advantages of PDP are its intuitiveness and ease of implementation. Despite this, the PDP has a big problem with features that are correlated – for example, for the calculation of PDP in the figure 2.9, the dependence of temperature on the season is completely ignored. Thus, for the calculation of PDP for the value of high temperatures even the dataset's specimens describing winter days are included in the calculation of the average prediction value (by setting their temperature to such high values) which creates a distorted interpretation. Furthermore, since PDP involves averaging of the calculated values, opposing influences will cancel each other out and thus may go unnoticed. Another disadvantage of PDP is its inability to visualize the influence of more than two features simultaneously [2].

PDP example is shown in figure 2.9. By reviewing the PDP, one can notice that as the temperature rises, so does the number of rented bicycles.

2.3.2 Individual Conditional Expectation – ICE

ICE plots are equivalent to PDP, except that they show each instance of the dataset as a separate function curve. In other words, for each instance of the data set $\{(x_S^{(i)}, x_C^{(i)})\}_{i=1}^{N}$, the curve $\hat{f_S^{(i)}}$ is displayed against $x_S^{(i)}$ (replaced by values from feature S's domain space), while $x_C^{(i)}$ remains fixed [2].

The advantages of the ICE display, as with the PDP, are the intuitiveness and ease of calculation, but also the possibility of detecting opposing influences that may go unnoticed with the PDP due to the averaging of the calculated values. In addition to the disadvantages of PDP regarding the neglect of dependencies between features and the number of features that can be displayed simultaneously, ICE representations are generally not fully interpretable due to the number of instances within a dataset. Despite that, they can still provide excellent insight into machine learning models

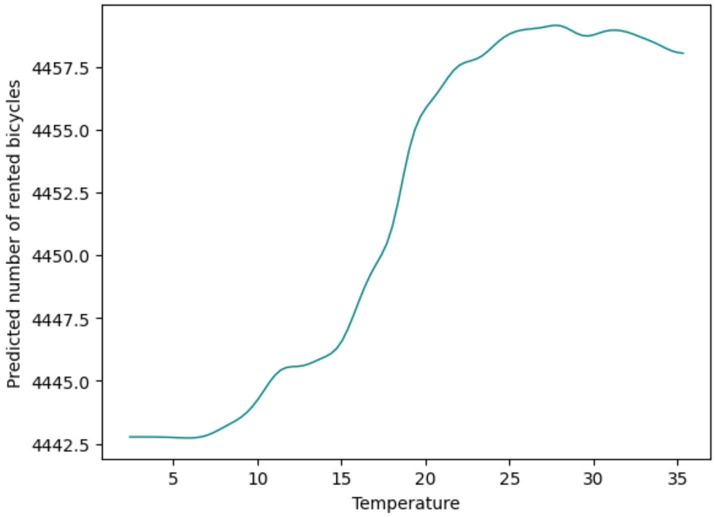


Figure 2.9 Partial Dependence Plot for the feature describing the temperature. One can notice that the rise in temperature in average scenario results in higher number of rented bicycles.

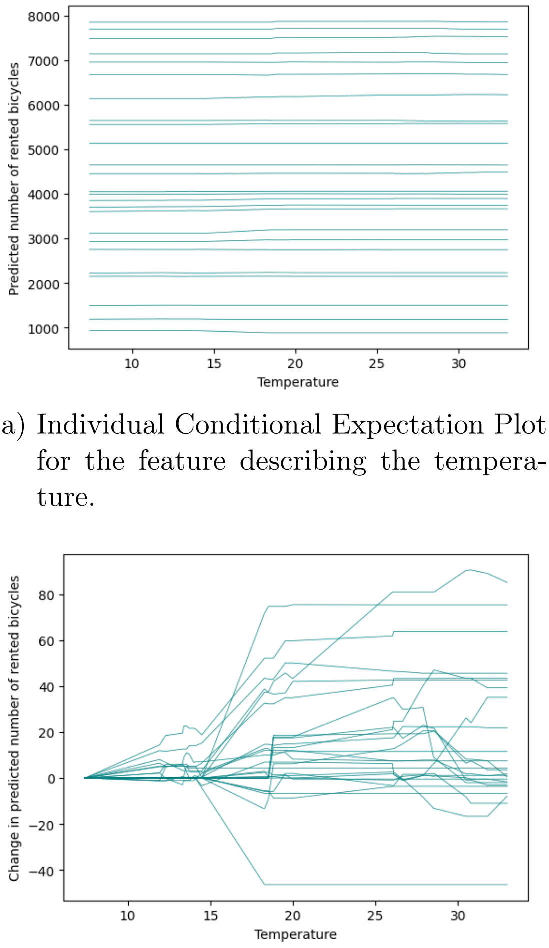
related to medical use-cases [41], but in general as well [42].

An example of ICE is shown in the figure 2.10. A review of the ICE shows that there are no instances within the dataset that behave significantly differently from the average, and that the PDP illustrates the average situation well. Figure 2.10 also shows the Centered ICE view, which often makes it easier to compare dataset samples with regard to the observed feature.

2.3.3 Accumulated Local Effects – ALE

ALE describes how features affect a machine learning model's prediction on average. They are model agnostic so they can be used on any supervised machine learning model [43]. Unlike PD and ICE, ALE only takes the samples with realistic combinations of features into account [2]:

- PD – what the model predicts on average when the observed feature is set to the desired value for each instance,



(a) Individual Conditional Expectation Plot for the feature describing the temperature.

(b) Centered Individual Conditional Expectation Plot for the feature describing the temperature.

Figure 2.10 Individual Conditional Expectation Plot. One can notice the difference relative to Partial Dependence Plots based on multiple instances that are displayed through the whole feature value range.

- ALE – how model predictions change in a narrow feature region around a desired value for dataset instances within that region.

The ALE value is calculated according to the expression

$$\hat{\tilde{f}}_{j,ALE}(x) = \sum_{k=1}^{k_j(x)} \frac{1}{n_j(k)} \sum_{i:x_j^{(i)} \in N_j(k)} \left[f(z_{k,j}, x_{\setminus j}^{(i)}) - f(z_{k-1,j}, x_{\setminus j}^{(i)}) \right]$$

which can be divided into three parts:

1. **e**ffect – the red part of the expression – the difference in predictions (in order to only take the observed feature's effect on the prediction into account) over instances where the feature j is set to the value z_k, i.e. z_{k-1} which marks the feature j's edge values of the currently observed region,
2. **l**ocal – the green part of the expression – calculation of the average difference in the predictions of all instances in the dataset within the observed region (i.e. only of realistic instances – the difference in relation to PD),
3. **a**ccumulated – the blue part of the expression – the summation of the effect across all defined areas.

The values obtained by the given expression are centered to show the deviation from the average prediction – $\hat{f}_{j,ALE}(x) = \hat{\tilde{f}}_{j,ALE}(x) - \frac{1}{n}\sum_{i=1}^{n} \hat{\tilde{f}}_{j,ALE}(x_j^{(i)})$ [2].

Ultimately, the ALE value should be understood as the impact of a feature at a particular value compared to the average prediction of the model. For example, the ALE value 15 for $x_j = 11$ means that at such a value of the feature j, the prediction is for 15 units higher than the average prediction of the model.

ALE plots bring several advantages:

- also work when there is interaction between dataset features,
- calculation is faster compared to PD displays,
- interpretation is clear and intuitive.

Despite the great advantages, the disadvantages of the ALE view are manifested in the form of determining the number of used areas during the calculation and much more complex implementation compared to the PD and ICE views [2].

An example of the ALE display is given in the figure 2.11. By examining the ALE plot, it can be determined that temperature has the greatest influence, and wind speed the least influence, on the average model prediction.

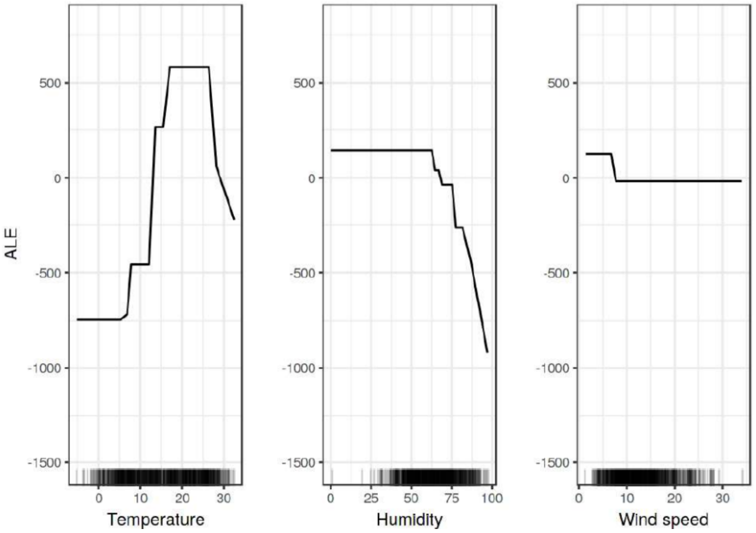


Figure 2.11 ALE plot for the features that represent temperature, humidity and wind speed [2]. Although the plot looks similar to the PDP, only realistic combination of feature values is taken into account.

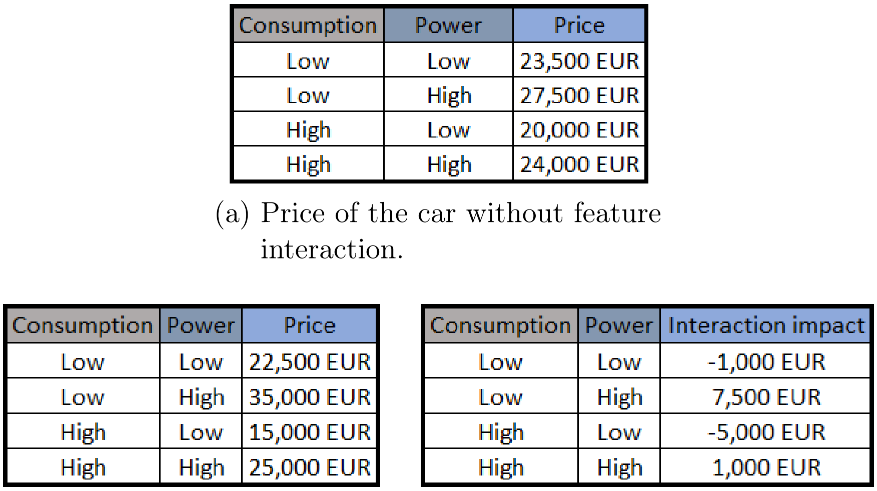
2.3.4 Feature interaction

The interaction between two features is the change in the model prediction that results from a specific combination of feature values after taking the individual contributions of the features into account [2].

For example, if a machine learning model trained on a dataset where the price of a car is determined by consumption and power characteristics (figure 2.12) is being explained:

- with the absence of interaction between the features of consumption and power, the total price of the car will be the sum of the initial price of the car (20000 EUR) and individual contributions to the price based on consumption (high -1000 EUR, low +2500 EUR) and power (high +5000 EUR, low +1000 EUR) (subfigure 2.12a),

- with the interaction of the consumption feature with the power feature, the total price of the car will be the sum of the initial price of the car, the individual contributions of both features and the contribution of their interaction in a certain combination – for example, a car with high power with low consumption contributes to the price with +7500 EUR (subfigure 2.12b).



Consumption	Power	Price
Low	Low	23,500 EUR
Low	High	27,500 EUR
High	Low	20,000 EUR
High	High	24,000 EUR

(a) Price of the car without feature interaction.

Consumption	Power	Price
Low	Low	22,500 EUR
Low	High	35,000 EUR
High	Low	15,000 EUR
High	High	25,000 EUR

Consumption	Power	Interaction impact
Low	Low	-1,000 EUR
Low	High	7,500 EUR
High	Low	-5,000 EUR
High	High	1,000 EUR

(b) Price of the car with feature interaction.

Figure 2.12 Feature interaction. The tables show the interaction between features on each possible value combination.

The level of feature interaction is calculated in a simple manner using the H-statistic. It is possible to calculate the level of interaction of a certain feature with respect to all other features or the level of interaction between two features.

If the feature j does not interact with any other feature, the prediction of the model can be expressed as a sum of PD functions, one of which depends only on the feature j, and the other on all other features:

$$\hat{f}(x) = PD_j(x_j) + PD_{-j}(x_{-j})$$

. In the next step, the difference between the obtained outputs from the model and the assumed function is observed, which reveals the level of interaction [2]. In a similar way, the level of interaction between a pair of features is obtained.

The advantages of this method are that all types of interactions are detected, and the result is a number within the interval $[0, 1]$, which makes the method comparable between several different models and features. Despite this, the method is computationally demanding and includes the estimation of marginal distributions, which contributes to the instability of the results [2].

An example of the interactions of features with all the remaining features is given in the figure 2.13.

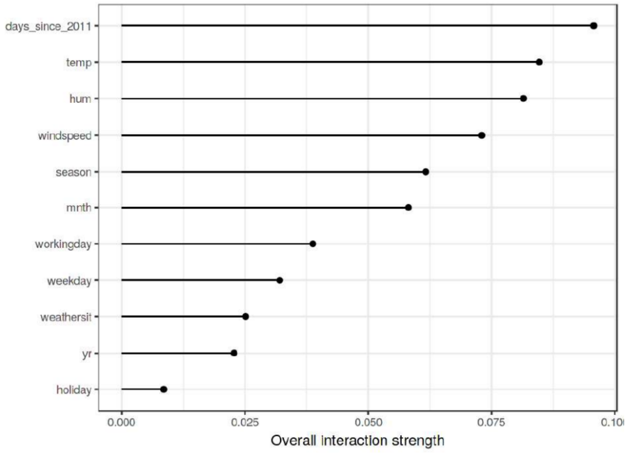


Figure 2.13 Feature interactions on daily rented bicycles dataset [2]. The plot is very interpretable – the longer the line, the higher the interaction.

2.3.5 Feature importance

The feature importance level is the increase in the error of the machine learning model after permuting the feature value, which imitates the effect of destroying the connection between the feature and the actual result. The way the importance level of a particular feature is calculated is by looking at the error of the machine learning model that occurs after said permutation – the more important the feature, the higher the model's error after permuting its values [2].

The advantages of this method are that it is easily interpretable, comparable across different types of problems and takes into account all types of interactions with other features. The biggest downside is that it is not completely clear whether it should be carried out on the test or training dataset, and that there are differences in the results with regard to the way of permuting the feature values within the chosen dataset [2].

An example of features' importance is given in the figure 2.14. One can conclude that, in accordance with the mentioned procedure, the used machine learning model gave greatest importance to the feature that describes the temperature.

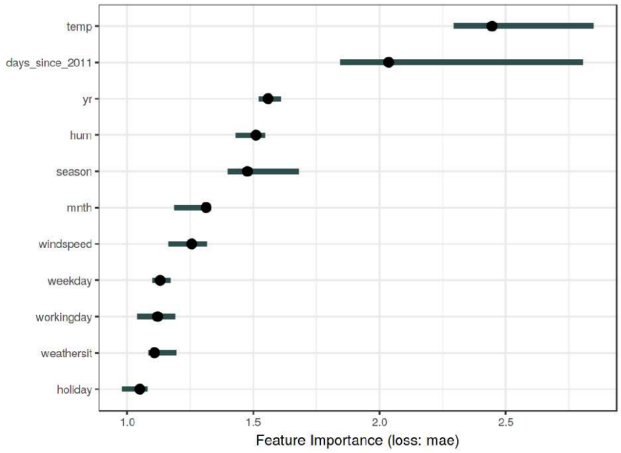


Figure 2.14 Feature importance on daily rented bicycles dataset [2]. One can notice that the `temp` feature is the most important while `holiday` feature is the least important.

2.3.6 Shapely values

The Shapely value expresses how much a particular feature contributes to a particular result of a machine learning model [44].

The goal of the Shapely value is to explain the difference between the average predicted value of the model and the predicted value of the model, that is, to explain which feature had what influence in that difference [2].

The Shapely value is determined for a particular feature by calculating its average contribution using all combinations of all other features.

An example of the display of Shapely values is given in the figure 2.15 and it can be concluded that the most negative impact on the model's prediction was brought by bad weather and high humidity.

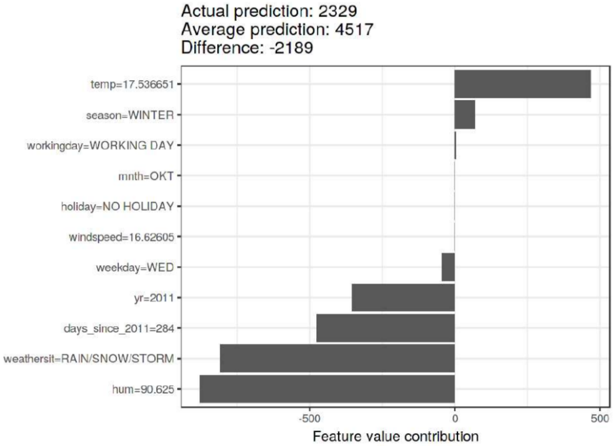


Figure 2.15 Shapely values for a prediction from daily rented bicycles dataset [2]. One can observe that the temperature feature has the most positive influence on the prediction value, whilst humidity lowers the predicted value the most.

Chapter 3

Conducted Experiments

This chapter discusses the experiments conducted on the aforementioned dataset using a selection of machine learning algorithms and explainability methods, all of which were previously described in chapter 2. All machine learning models included in the scope of this paper – decision tree, OneR, sequential covering, random forest and neural network – were trained on each instance of the dataset. Neural network model was also trained on oversampled versions of the dataset instances, whilst other models did not use the process of oversampling due to unnoticeable changes in their performance metrics when using the mentioned process. The explainability techniques described in section 2.3 were used to explain the predictions and background of non-interpretable models – random forest and neural networks.

3.1 Splitting the dataset

The dataset was split in the same way for each instance of the dataset. Each dataset was divided to cross-validation data and test data, in usual ratio of 80%-20%. K-Fold cross-validation with 5 folds was used. In cases where models used oversampling, the technique was implemented on each fold's training set only.

All models used this process to split the datasets except for OneR model which did not use cross-validation data for K-fold cross-validation but for regular training process since there were no hyperparameters to be tuned.

3.2 Decision tree

The decision tree machine learning algorithm was created and trained to make predictions on all instances of the original dataset – HBD, HD, PBD and PD. Data on its performance for each task was gathered through various metrics. As for explainability methods, only feature importance was visualized since decision tree is an interpretable machine learning model thus does not need additional working using explainability techniques.

Decision tree's hyperparameters were chosen using experimental analysis – each combination of the hyperparameters was evaluated using the average accuracy during cross-validation. Hyperparameters used during experimental analysis, except the default ones, are shown in table 3.1. Best parameters for each dataset instance's decision tree are noted in chapter 4.

Hyperparameter	Used values in analysis
`max_leaf_nodes`	[10, 15, 20]
`max_features`	[2, 5, 8]
`max_depth`	[6, 8, 10]
`min_samples_split`	[10, 20]
`min_samples_leaf`	[5, 10]

Table 3.1 Analysed parameters during decision tree model training.

Gathered predictions on test set were analysed and various metrics were calculated – precision, recall, F1-score and accuracy. Plots that visualize mentioned metrics and confusion matrix were also generated.

Additionally, feature importances were calculated and visualized using `sklearn`'s `permutation_importance` function.

3.3 OneR

As the previous machine learning model, OneR algorithm was also used with all available dataset instances – HBD, HD, PBD and PD. Various model's performance metrics were gathered. Explainability techniques for this machine learning model were not used as the algorithm produces rules for a sole feature which makes it as interpretable as possible.

There were no hyperparameters that could be used so none were analysed and cross-validation was not used. Model's performance metrics, namely precision, recall, F1-score and accuracy were presented through various bar plots and a confusion matrix display.

3.4 Sequential covering

Sequential covering machine learning algorithm is another and the last of the interpretable machine learning algorithms presented in this paper. Despite its interpretability, its resulting decision list is often long and consists of many rules, especially in multinomial classification. Because of that, it can benefit of feature importance explainability technique that will sum up the rules' influences and display them in an interpretable way.

As the sequential covering algorithm decides on an individual rule by fitting a decision tree classifier to the data and selecting the purest node to deduct a rule, there is quite a number of hyperparameters that can be tuned. Hyperparameters available to tune are actually inherited from the decision tree (specifically `sklearn`'s `DecisionTreeClassifier`).

Hyperparameters were chosen using experimental analysis based on the average accuracy during cross-validation. Hyperparameters used during experimental analysis, are shown in table 3.2. Best parameters for each dataset instance's sequential covering model are noted in chapter 4.

The results the model accomplished on the test set were displayed in form of confusion matrix and already mentioned metrics – precision, recall, F1-score and accuracy.

Aforementioned feature importances were also calculated and plotted and are displayed in 4.

Hyperparameter	Used values in analysis
`max_leaf_nodes`	[10, 15, 20, 25, 30]
`max_features`	[2, 5, 8, 11]
`max_depth`	[3, 6, 9, 12]
`min_samples_split`	[2, 5, 8]
`min_samples_leaf`	[1, 6, 11]

Table 3.2 Analysed parameters during sequential covering model training.

3.5 Random forest

Random forest machine learning algorithm is an introduction to non-interpretable models of this chapter. Like all other machine learning algorithms used in this paper, random forest was used in combination with all instances of datasets – HBD, HD, PBD and PD.

Random forest model's pipeline did not differ from the usual pipeline described on the previous, simpler machine learning models. Tuneable hyperparameters were analysed with respect to average validation accuracies during cross-validation of each hyperparameter combination. Hyperparameters used during experimental analysis, except the default ones, are shown in table 3.3. Best parameters for each dataset instance's random forest model are noted in chapter 4.

Hyperparameter	Used values in analysis
`n_estimators`	[25, 50, 75, 100]
`max_leaf_nodes`	[10, 20, 30, 40]
`max_features`	[2, 5, 8, 11]
`max_depth`	[6, 8, 10]
`min_samples_split`	[1, 11, 21]
`min_samples_leaf`	[1, 6, 11]

Table 3.3 Analysed parameters during random forest model training.

Visualizations of performance metrics – precision, recall, F1-score, accuracy and confusion matrix were generated. Furhtemore, since the model is not interpretable, additional plots that offer explainability were generated:

1. partial dependence plot,
2. individual conditional expectation plot,
3. accumulated local effects plot,
4. feature importance,
5. SHAP.

Visualizations are presented in chapter 4.

3.6 Neural network

Neural network machine learning model is another non-interpretable model studied in this paper. It was used for all instances of the initial dataset – HBD, HD, PBD and PB – but also for their oversampled counterparts. Chapter 4 brings the results which show whether the oversampling helped. Oversampling technique was exclusive to neural network models since using the technique did not show any noticeable changes in other models' performance metrics.

Previously mentioned 5-fold cross-validation was used to analyse hyperparameter combinations which are shown in table 3.4. Best parameters for each dataset instance's neural network model are noted in chapter 4.

Hyperparameter	Used values in analysis
`learning_rate`	[0.1, 0.01, 0.001, 0.0001, 10^{-5}]
`optimizer`	['Adam', 'SGD', 'AdamW', 'Adafactor', 'Nadam']
`loss` (binary/multinomial)	['binary_crossentropy', 'categorical_crossentropy'] ['sparse_categorical_crossentropy', 'categorical_crossentropy']
`dropout`	[0.05, 0.1, 0.15, 0.2]

Table 3.4 Analysed parameters during neural network model training.

Neural network model always used the same architecture shown in figure 3.1. It consists of an input layer, hidden layers which include 512, 512, 256 and 128 neurons respectively and an output layer. All hidden layers apart the first one are followed by a dropout layer.

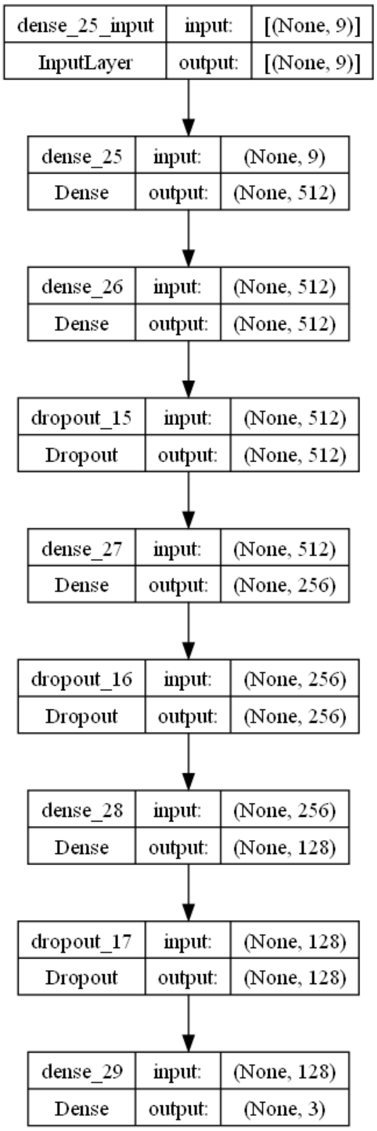


Figure 3.1 Neural network model architecture used in this research.

As for the other models, model performance metrics were generated and visualized appropriately. Range of explainability techniques used is identical to the one described in the previous section – 3.5.

Chapter 4

Results

This chapter states the performance metrics of each machine learning algorithm trained on the dataset. Furthermore, results of explainability techniques are presented in this chapter.

4.1 Decision tree

The decision tree machine learning model was trained on all dataset variations, as noted in 3.2. The best parameters gathered through experimental analysis are noted in 4.1.

Hyperparameter	HD	HBD	PD	PBD
`max_leaf_nodes`	20	10	10	10
`max_features`	2	5	2	2
`max_depth`	8	6	6	6
`min_samples_split`	20	20	20	10
`min_samples_leaf`	5	5	5	5

Table 4.1 Best parameters from experimental analysis for decision tree models.

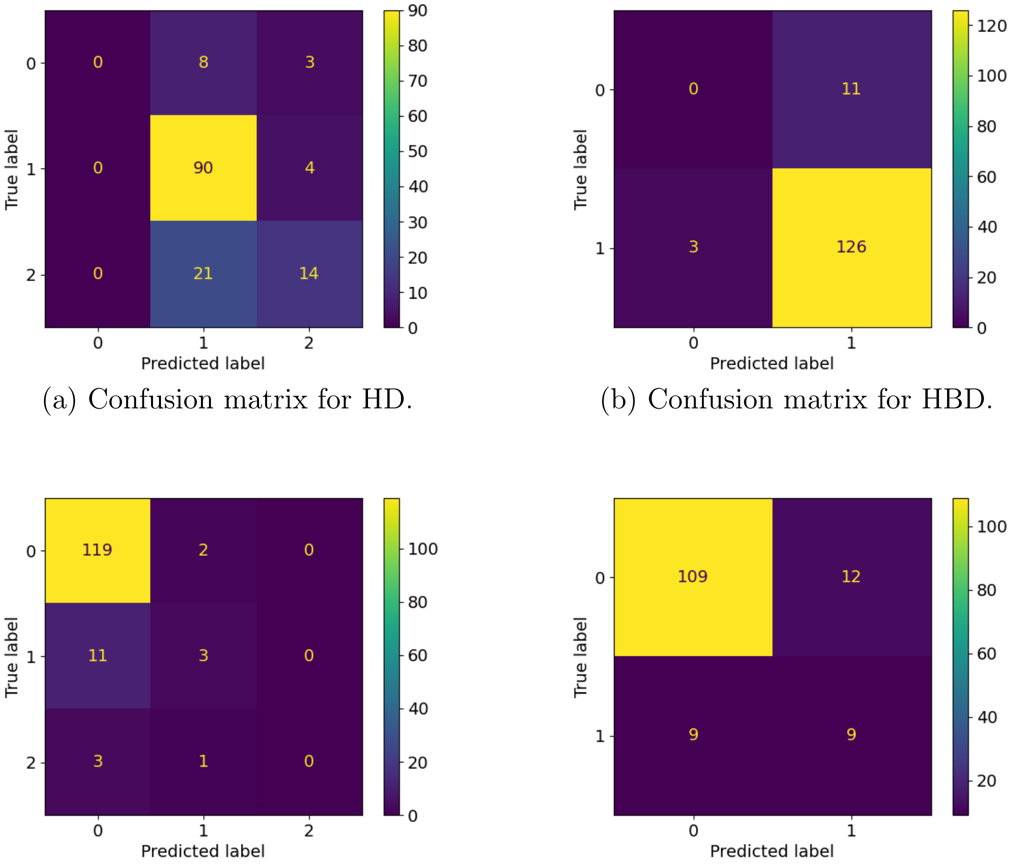
Various performance metrics gathered for each dataset instance's decision tree model on test set and during cross-validation are displayed in table 4.2.

Table 4.2 Performance metrics from each dataset instance's trained model.

Dataset	Class	Precision	Recall	F1-score	Weighted F1-score	Accuracy	Validation accuracy
HD	0	0.00	0.00	0.00	0.69	0.74	0.77
	1	0.76	0.96	0.85			
	2	0.67	0.40	0.50			
HBD	0	0.00	0.00	0.00	0.87	0.90	0.94
	1	0.92	0.98	0.95			
PD	0	0.89	0.98	0.94	0.85	0.88	0.84
	1	0.50	0.21	0.30			
	2	0.00	0.00	0.00			
PBD	0	0.92	0.90	0.91	0.85	0.85	0.85
	1	0.43	0.50	0.46			

Furthermore, confusion matrix that outlines number of true/false positives/negatives is given in figure 4.1.

Additionally, feature importance plots shown in figure 4.2 further raise the interpretability of the trained models.



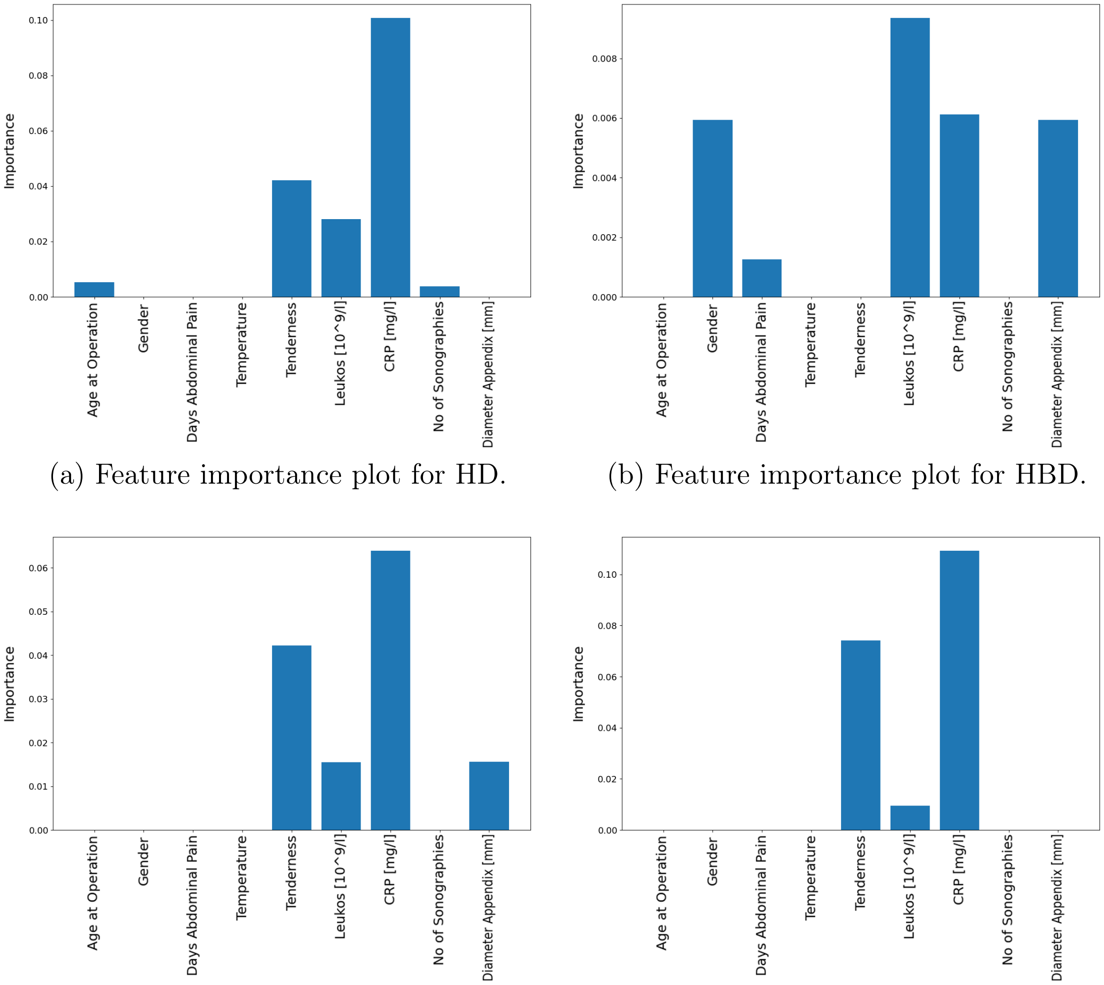
(a) Confusion matrix for HD.

(b) Confusion matrix for HBD.

(c) Confusion matrix for PD.

(d) Confusion matrix for PBD.

Figure 4.1 Decision trees confusion matrices.



(a) Feature importance plot for HD.

(b) Feature importance plot for HBD.

(c) Feature importance plot for PD.

(d) Feature importance plot for PBD.

Figure 4.2 Decision trees feature importance plots. They show that CRP and leukocytes features are dominant concerning the feature importance.

4.2 OneR

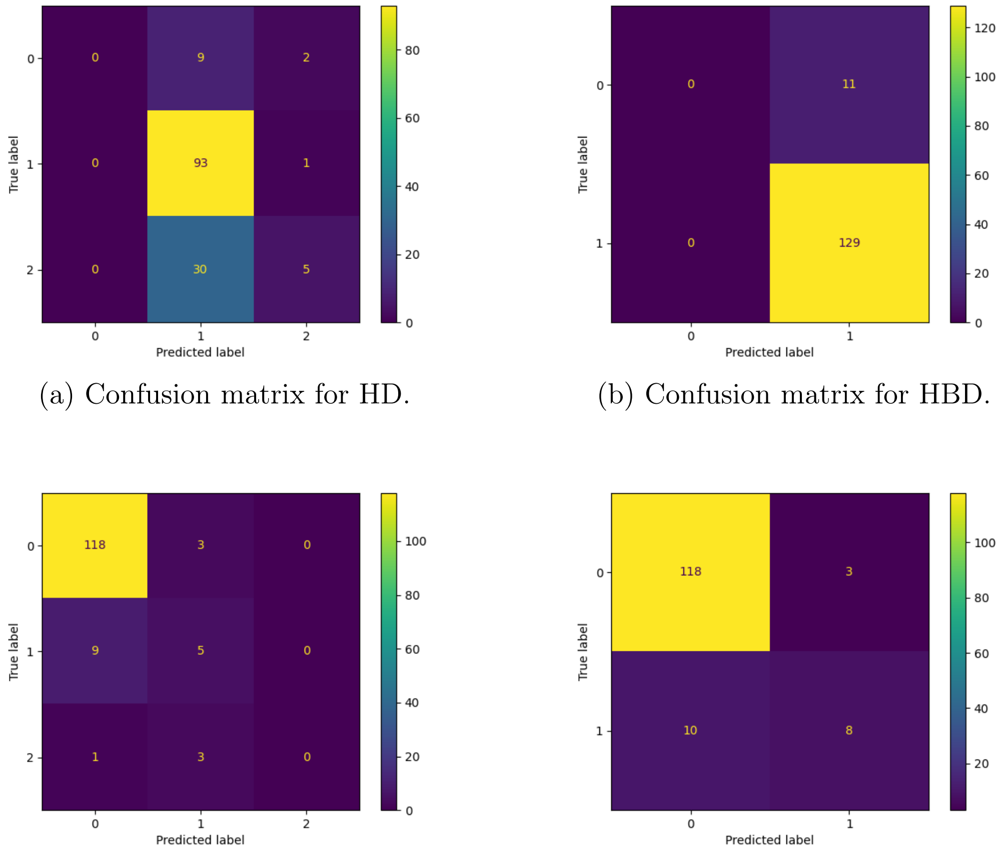
Although OneR was also trained on all of the available dataset instances, there was not any need for hyperparameter tuning since it has none. Gathered performance metric on test set are displayed in table 4.3.

Dataset	Class	Precision	Recall	F1-score	Weighted F1-score	Accuracy
HD	0	0.00	0.00	0.00	0.61	0.70
	1	0.70	0.99	0.82		
	2	0.62	0.14	0.23		
HBD	0	0.00	0.00	0.00	0.88	0.92
	1	0.92	1.00	0.96		
PD	0	0.92	0.98	0.95	0.87	0.88
	1	0.45	0.36	0.40		
	2	0.00	0.00	0.00		
PBD	0	0.92	0.98	0.95	0.90	0.91
	1	0.73	0.44	0.55		

Table 4.3 Performance metrics from each dataset instance's trained model.

Confusion matrices for trained OneR models are given in figure 4.3.

No explainability methods were used to explain OneR models' predictions since they are as interpretable as possible.



(a) Confusion matrix for HD.

(b) Confusion matrix for HBD.

(c) Confusion matrix for PD.

(d) Confusion matrix for PBD.

Figure 4.3 OneR confusion matrices.

4.3 Sequential covering

Sequential covering algorithm was also run on all datasets, and in contrast to OneR, has several tuneable hyperparameters which were optimised through experimental analysis. The hyperparameters that proved to be optimal for a specific dataset instance are presented in table 4.4.

Results of the sequential covering models in view of precision, recall, F1-score and accuracy metrics are shown in table 4.5.

As when reporting results of the previous models, confusion matrices for sequen-

Hyperparameter	HD	HBD	PD	PBD
`max_leaf_nodes`	10	20	10	10
`max_features`	5	2	2	8
`max_depth`	3	12	3	9
`min_samples_split`	2	8	2	2
`min_samples_leaf`	11	1	11	1

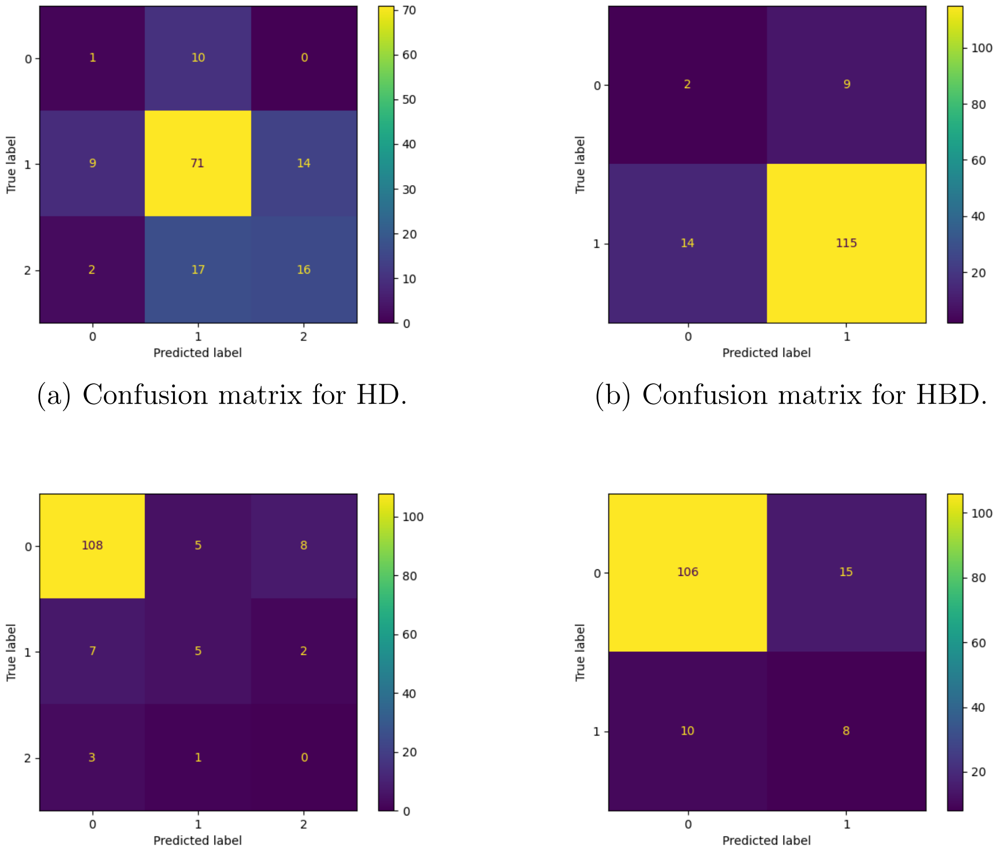
Table 4.4 Best parameters from experimental analysis for sequential covering models.

Dataset	Class	Precision	Recall	F1-score	Weighted F1-score	Accuracy	Validation accuracy
HD	0	0.08	0.09	0.09	0.63	0.63	0.71
	1	0.72	0.76	0.74			
	2	0.53	0.46	0.49			
HBD	0	0.12	0.18	0.15	0.85	0.84	0.91
	1	0.93	0.89	0.91			
PD	0	0.92	0.89	0.90	0.83	0.81	0.77
	1	0.45	0.36	0.40			
	2	0.00	0.00	0.00			
PBD	0	0.91	0.88	0.89	0.83	0.82	0.83
	1	0.35	0.44	0.39			

Table 4.5 Performance metrics from each dataset instance's trained model.

tial covering models trained on the available dataset variations are given in figure 4.4.

Furthermore, as sequential covering algorithm outputs decision lists for classifying each class separately, they can easily become long and cumbersome. Due to mentioned, feature importance analysis is presented as a form of increasing models' explainability.



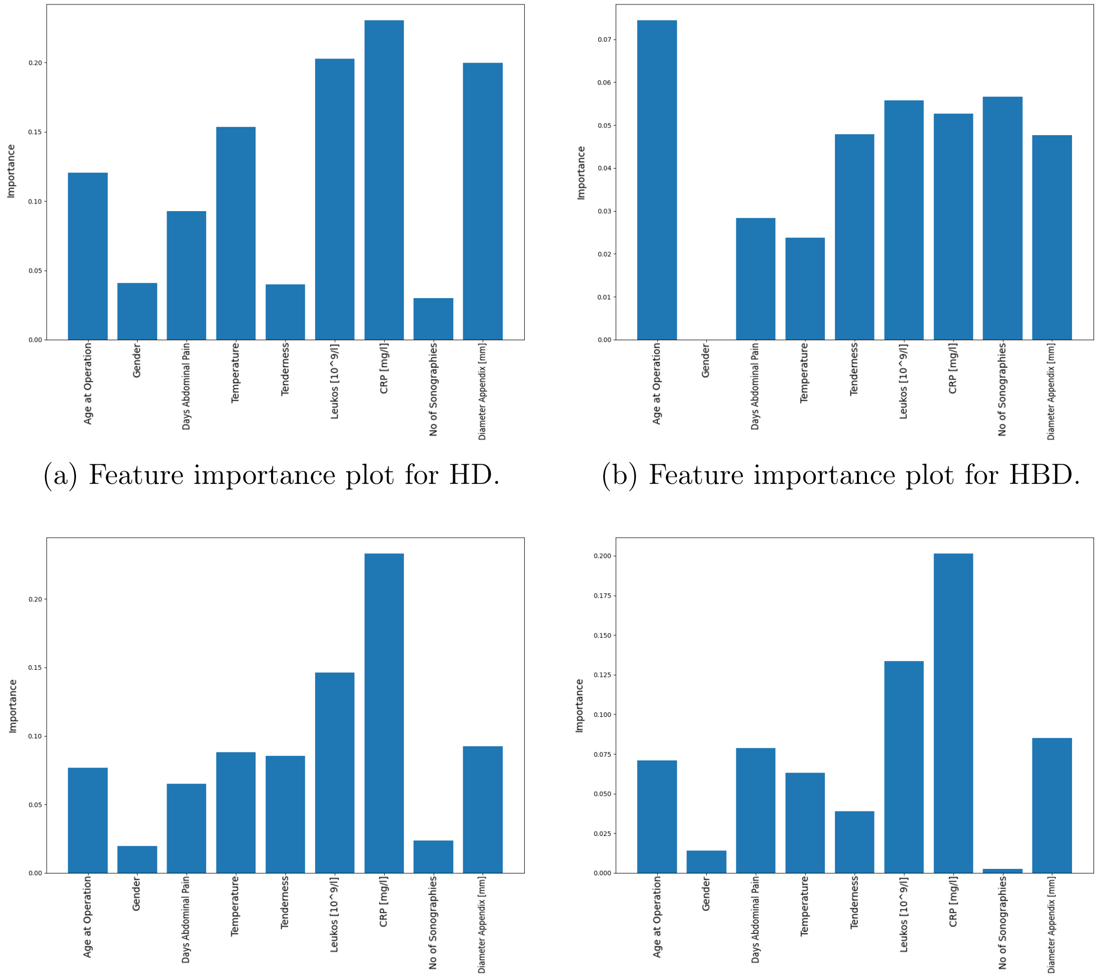
(a) Confusion matrix for HD.

(b) Confusion matrix for HBD.

(c) Confusion matrix for PD.

(d) Confusion matrix for PBD.

Figure 4.4 Sequential covering confusion matrices.



(a) Feature importance plot for HD.

(b) Feature importance plot for HBD.

(c) Feature importance plot for PD.

(d) Feature importance plot for PBD.

Figure 4.5 Sequential covering models' feature importance plots. CRP feature is the most important except in HBD where age of the patient is considered the most important.

4.4 Random forest

Random forest was, as is already usual, trained and tested on all of the available dataset instances. Optimal combinations of tuneable hyperparameters found through the process of experimental analysis are shown in table 4.6.

Hyperparameter	HD	HBD	PD	PBD
`n_estimators`	25	75	100	25
`max_leaf_nodes`	20	10	20	30
`max_features`	2	2	2	11
`max_depth`	6	6	10	10
`min_samples_split`	11	11	21	11
`min_samples_leaf`	1	1	1	1

Table 4.6 Best parameters from experimental analysis for random forest models.

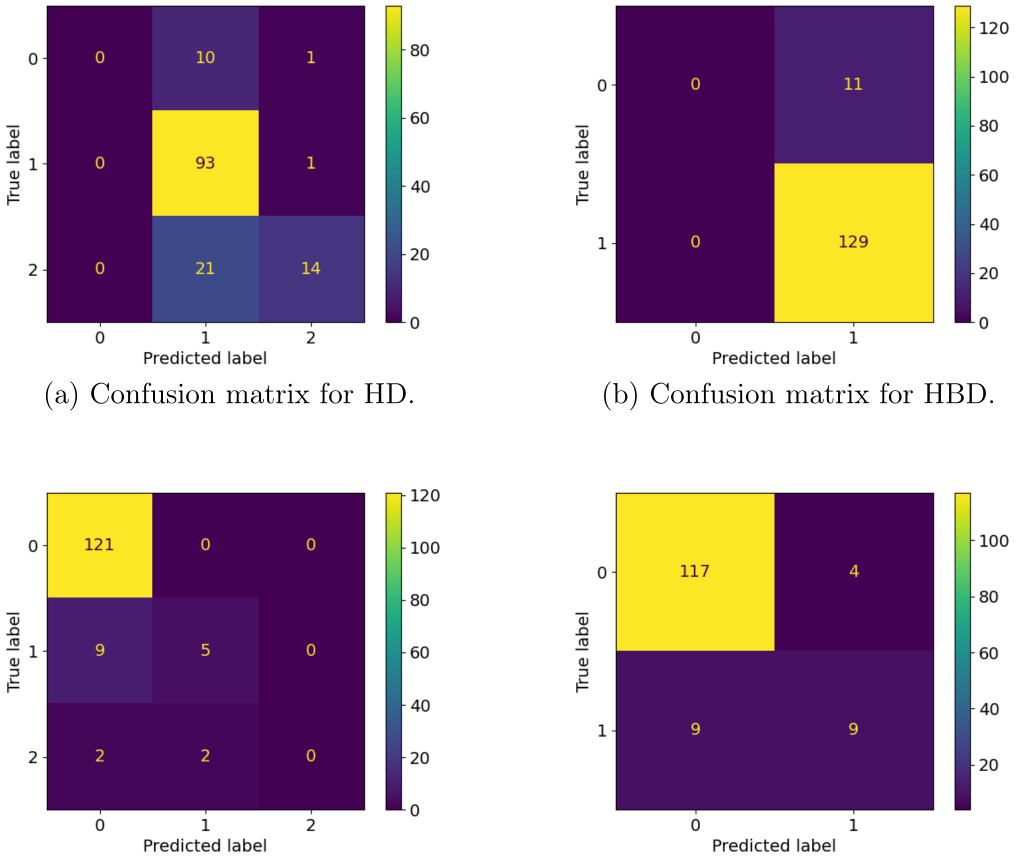
Performance metrics of the trained random forest models are available in table 4.7.

Dataset	Class	Precision	Recall	F1-score	Weighted F1-score	Accuracy	Validation accuracy
HD	0	0.00	0.00	0.00	0.71	0.76	0.79
	1	0.75	0.99	0.85			
	2	0.88	0.40	0.55			
HBD	0	0.00	0.00	0.00	0.88	0.92	0.94
	1	0.92	1.00	0.96			
PD	0	0.92	1.00	0.96	0.88	0.91	0.85
	1	0.71	0.36	0.48			
	2	0.00	0.00	0.00			
PBD	0	0.93	0.97	0.95	0.90	0.91	0.87
	1	0.69	0.50	0.58			

Table 4.7 Performance metrics from each dataset instance's trained model.

Confusion matrices for the trained models are shown in figure 4.6 and show additional insight into the models' performance.

Feature importance analysis of random forest models is presented in figure 4.7.



(a) Confusion matrix for HD.

(b) Confusion matrix for HBD.

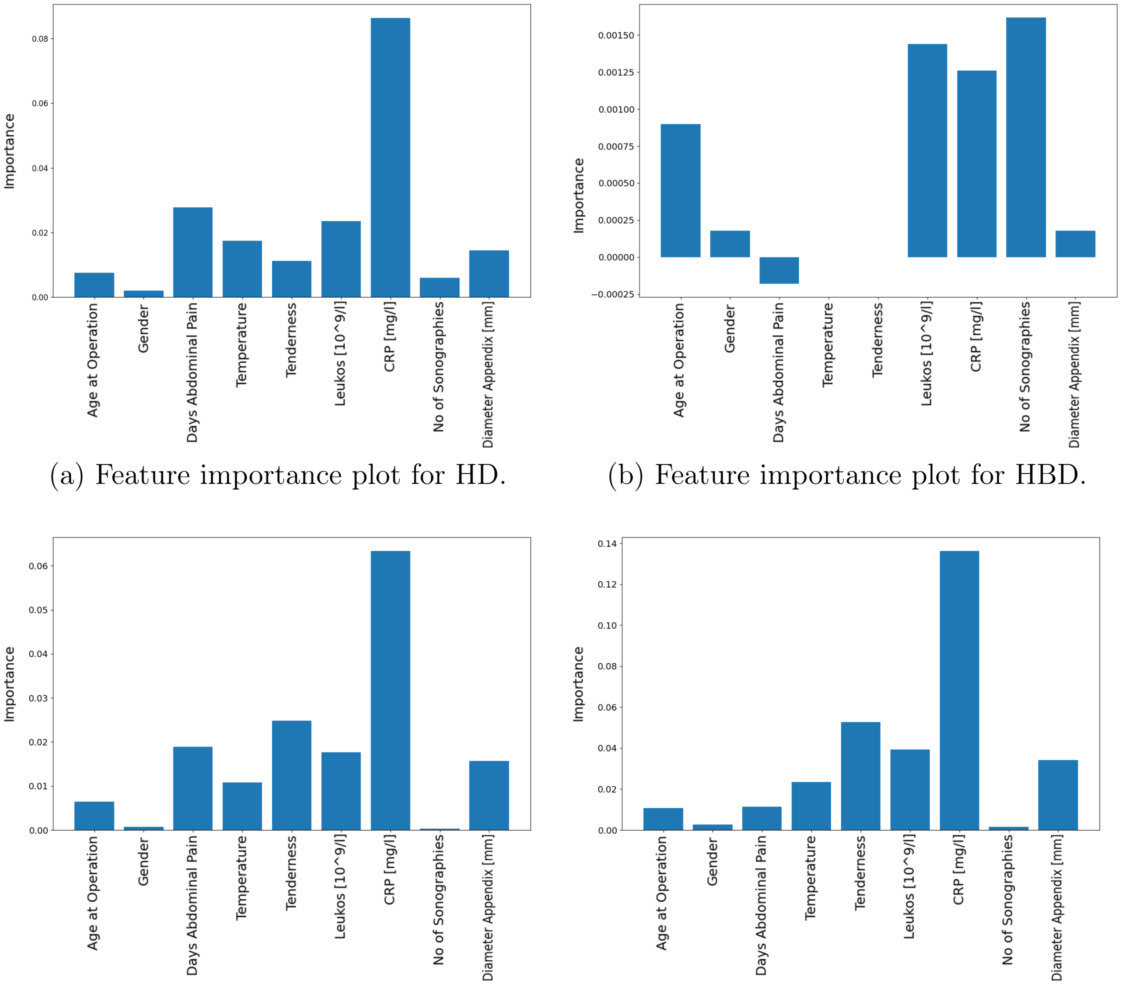
(c) Confusion matrix for PD.

(d) Confusion matrix for PBD.

Figure 4.6 Random forest confusion matrices.

As random forest models are not interpretable, much more explainability techniques were run using random forest models with respect to all the previously mentioned models in this chapter. Since many explainability visualizations were created (2 or 3 PD/ICE plots, 9 ALE plots and 1 SHAP plot for each of the four random forest models), this paper will only focus on explanations for only one random forest model trained on HD, only one output class and only on one feature – CRP – which was marked as the most important feature in the subfigure 4.7a. All the explanations and visualizations can be accessed in a way described in 4.6.

PD/ICE plot is provided in figure 4.8, ALE plot is presented in figure 4.9 and



(a) Feature importance plot for HD.

(b) Feature importance plot for HBD.

(c) Feature importance plot for PD.

(d) Feature importance plot for PBD.

Figure 4.7 Random forest models' feature importance plots. CRP feature is the most important except in HBD where the number of sonographies performed is considered the most important.

SHAP plot is given in figure 4.10.

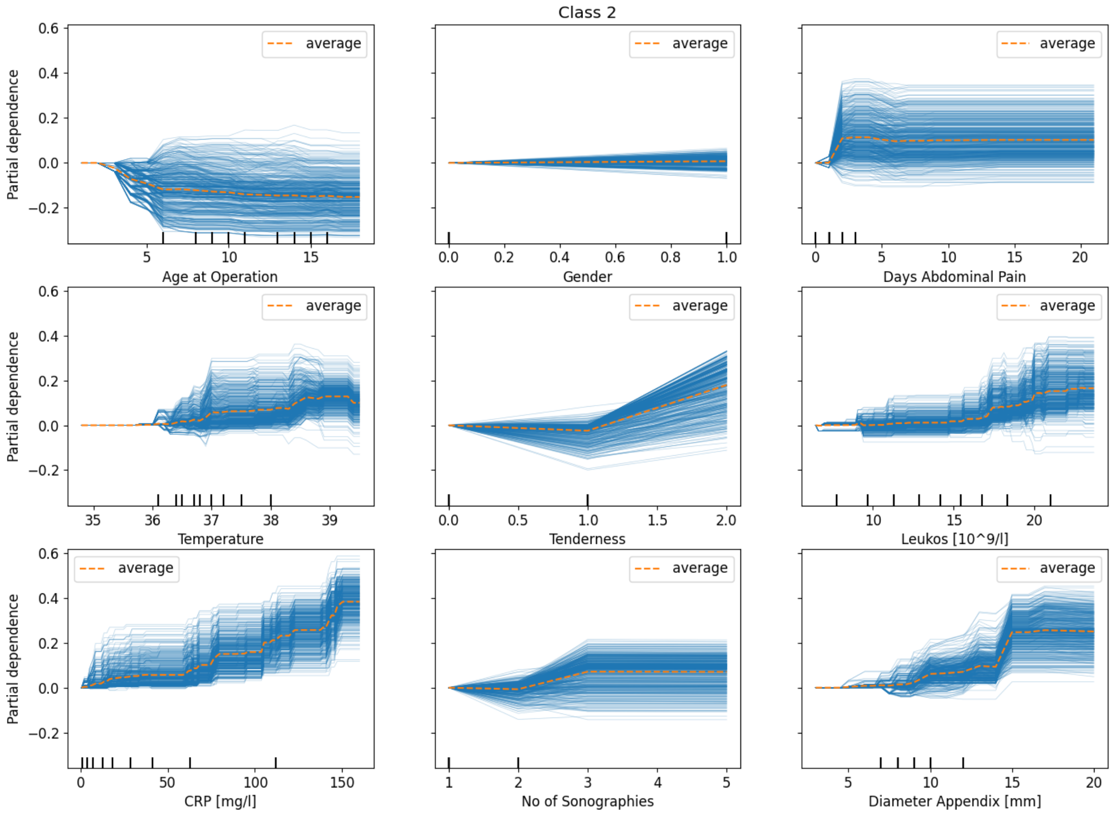


Figure 4.8 PDP/ICE centered plot for class 2 of random forest model trained on HD.

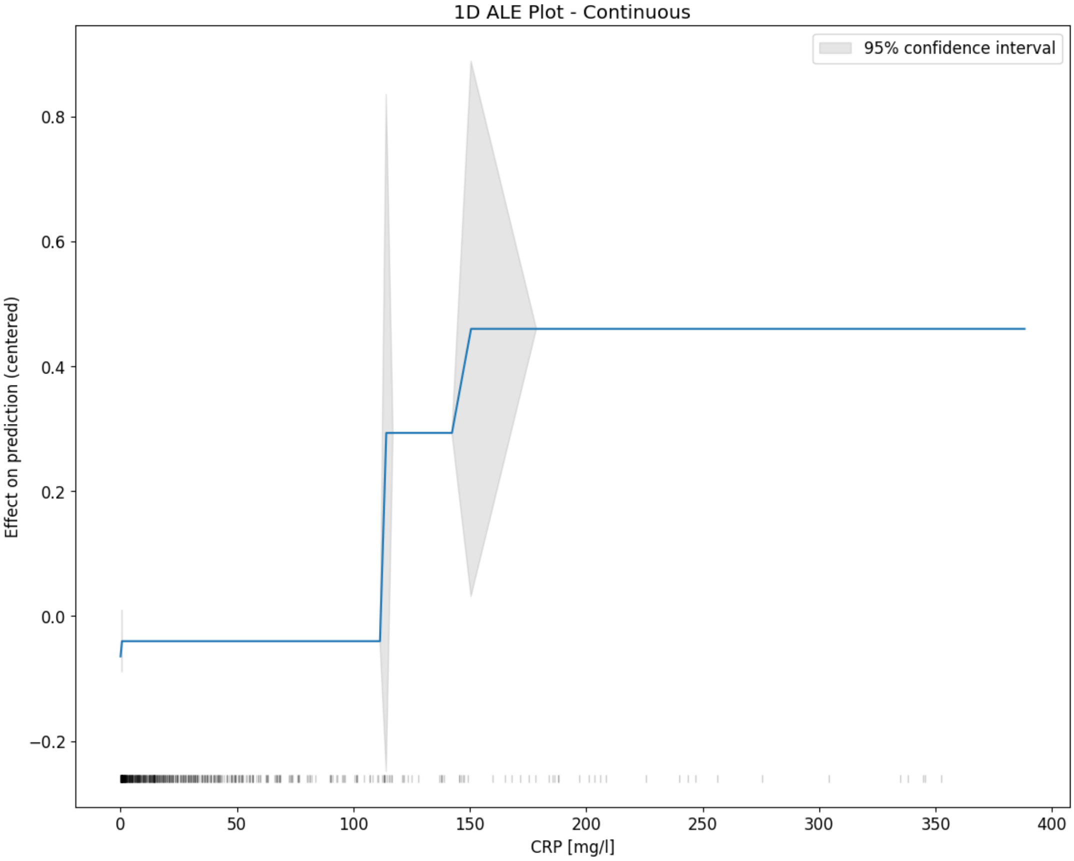


Figure 4.9 ALE plot for CRP feature of random forest model trained on HD.

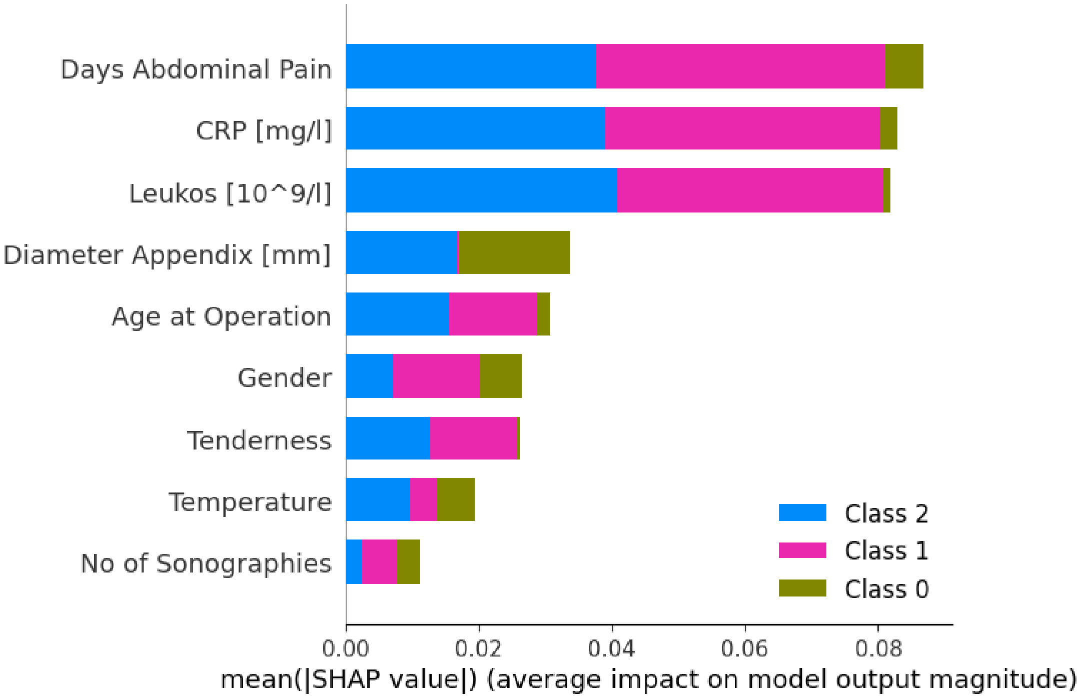


Figure 4.10 SHAP plot for random forest model trained on HD.

4.5 Neural network

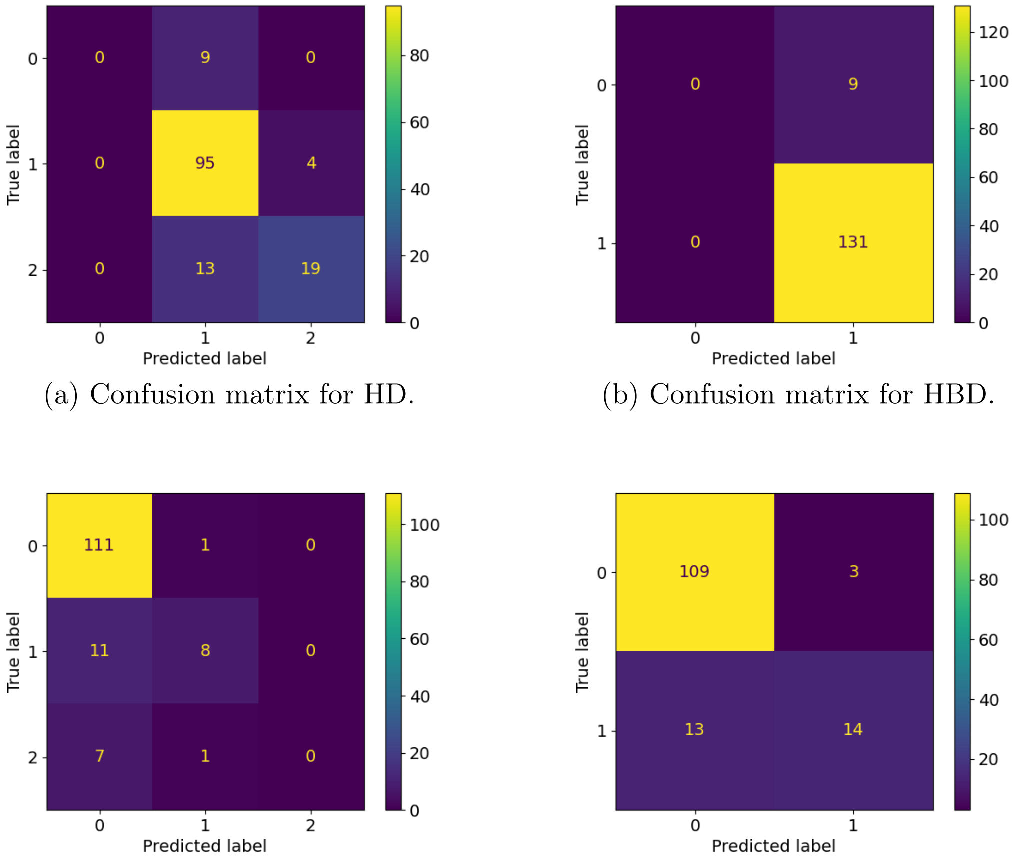
Work on neural network training was the most extensive. Instead of four dataset instances, neural network was trained on eight dataset instances due to oversampling variations of each of the available datasets. Optimal hyperparameters chosen for each dataset through experimental analysis are available in table 4.8.

After the neural network models were trained using their respective optimal parameters, performance metrics were gathered using the test set. The gathered data is shown in table 4.9.

Each model's confusion matrix and feature importance plot are shown in figures 4.11 and 4.12 respectively.

As neural network models are not interpretable, all of the available explainability techniques were used to increase the explainability of the models (as with random forest). Because of the same fact as with random forest models, only one model's explanations on one output class and one dataset instance will be displayed. That is because of a big number of visualizations generated during the process of processing explainability techniques (2 or 3 PD/ICE plots, 9 ALE plots and 1 SHAP plot for each of the eight trained neural network models). Displayed explanations will, as with the random forest model, explain neural network model's predictions on HD, class 2 and feature CRP. All the explanations and visualizations can be accessed in a way described in 4.6.

PD/ICE plot is provided in figure 4.13, ALE plot is presented in figure 4.14 and SHAP plot is given in figure 4.15.



(a) Confusion matrix for HD.

(b) Confusion matrix for HBD.

(c) Confusion matrix for PD.

(d) Confusion matrix for PBD.

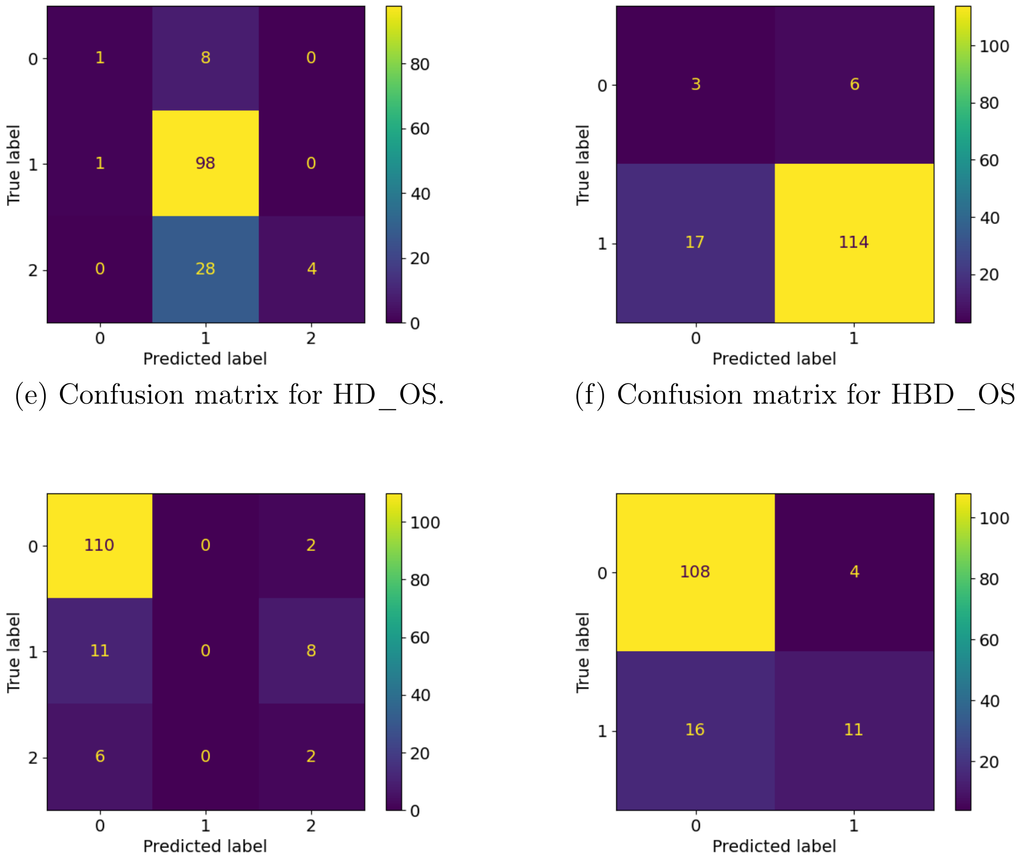
Figure 4.11 Neural network models confusion matrices.

Dataset	learning_rate	optimizer	loss	dropout
HD	0.1	SGD	categorical_crossentropy	0.2
HBD	1e-05	Nadam	binary_crossentropy	0.2
PD	0.1	SGD	sparse_categorical_crossentropy	0.05
PBD	0.1	SGD	categorical_crossentropy	0.15
HD_OS	0.1	SGD	categorical_crossentropy	0.2
HBD_OS	0.1	SGD	categorical_crossentropy	0.2
PD_OS	0.1	SGD	categorical_crossentropy	0.2
PBD_OS	0.1	SGD	categorical_crossentropy	0.05

Table 4.8 Best parameters from experimental analysis for random forest models.

Dataset	Class	Precision	Recall	F1-score	Weighted F1-score	Accuracy	Validation accuracy
HD	0	0.00	0.00	0.00	0.78	0.81	0.77
	1	0.81	0.96	0.88			
	2	0.83	0.59	0.69			
HBD	0	0.00	0.00	0.00	0.90	0.94	0.94
	1	0.94	1.00	0.97			
PD	0	0.86	0.99	0.92	0.82	0.86	0.84
	1	0.80	0.42	0.55			
	2	0.00	0.00	0.00			
PBD	0	0.89	0.97	0.93	0.87	0.88	0.90
	1	0.82	0.52	0.64			
HD_OS	0	0.50	0.11	0.18	0.66	0.74	0.74
	1	0.73	0.99	0.84			
	2	1.00	0.12	0.22			
HBD_OS	0	0.15	0.33	0.21	0.86	0.84	0.94
	1	0.95	0.87	0.91			
PD_OS	0	0.87	0.98	0.92	0.75	0.81	0.84
	1	0.00	0.00	0.00			
	2	0.17	0.25	0.20			
PBD_OS	0	0.87	0.96	0.92	0.84	0.86	0.89
	1	0.73	0.41	0.52			

Table 4.9 Performance metrics from each dataset instance's trained model.



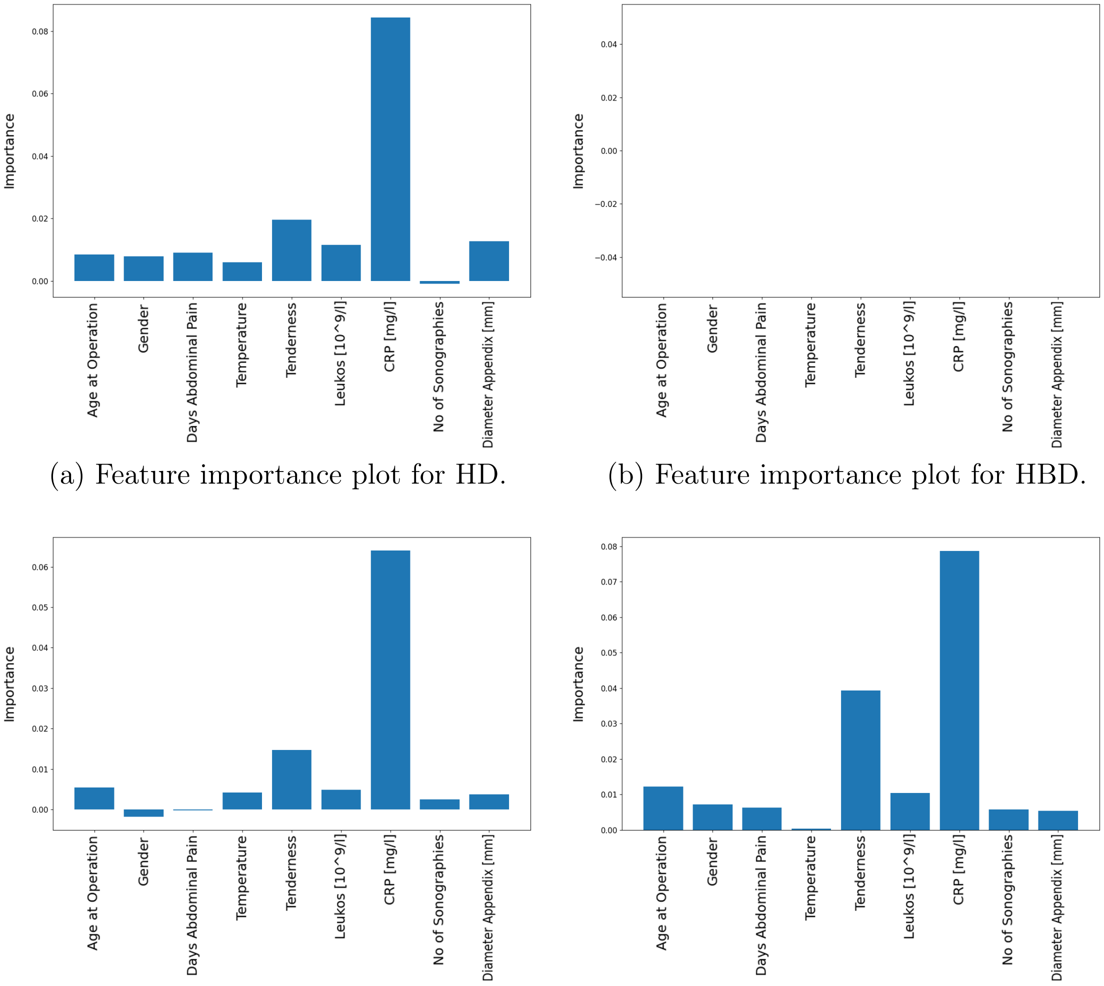
(e) Confusion matrix for HD_OS.

(f) Confusion matrix for HBD_OS.

(g) Confusion matrix for PD_OS.

(h) Confusion matrix for PBD_OS.

Figure 4.11 Neural network models confusion matrices.



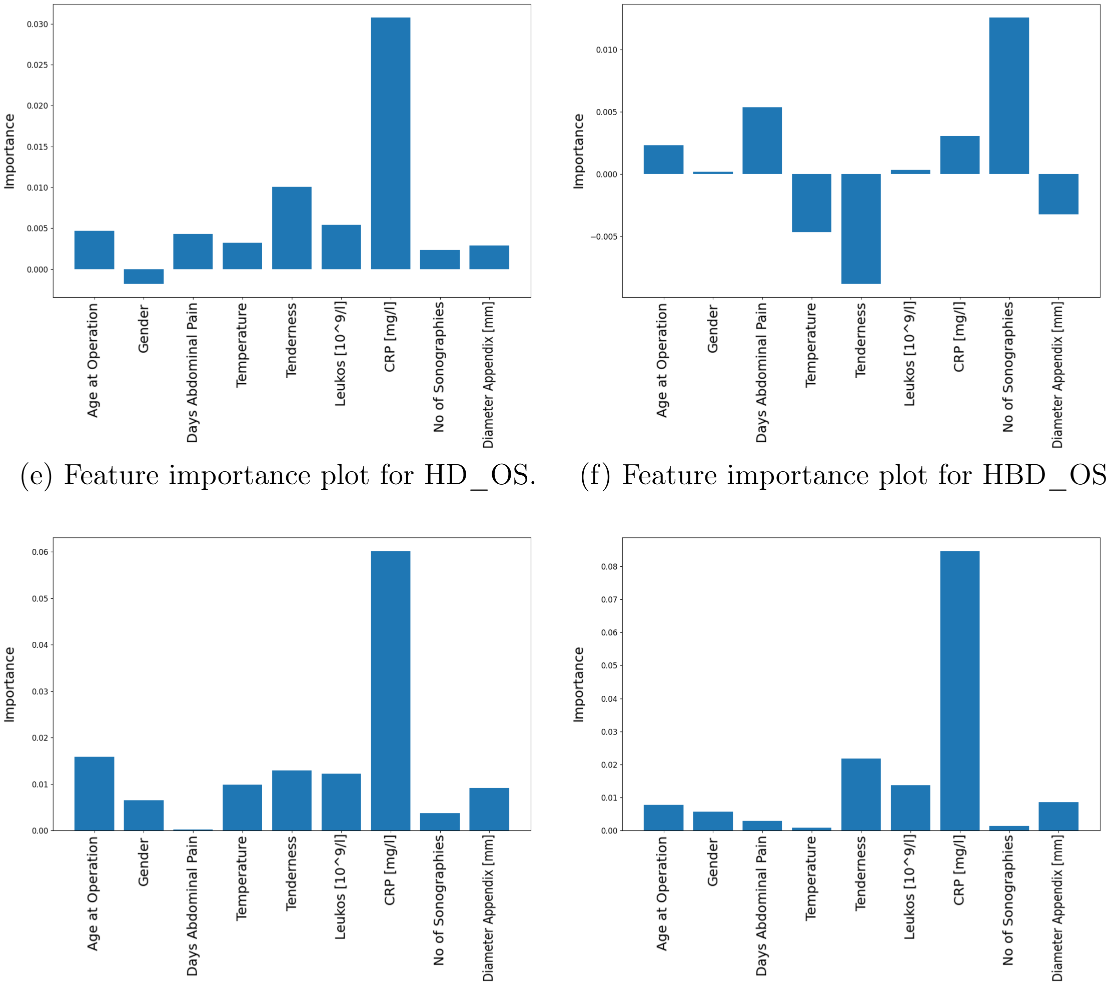
(a) Feature importance plot for HD.

(b) Feature importance plot for HBD.

(c) Feature importance plot for PD.

(d) Feature importance plot for PBD.

Figure 4.12 Neural network models' feature importance plots.



(e) Feature importance plot for HD_OS.

(f) Feature importance plot for HBD_OS.

(g) Feature importance plot for PD_OS.

(h) Feature importance plot for PBD_OS.

Figure 4.12 Neural network models' feature importance plots. One can observe that the CRP feature was the most important in all datasets but the HBD_OS and HBD where no feature is important since the model always predicts the same outcome.

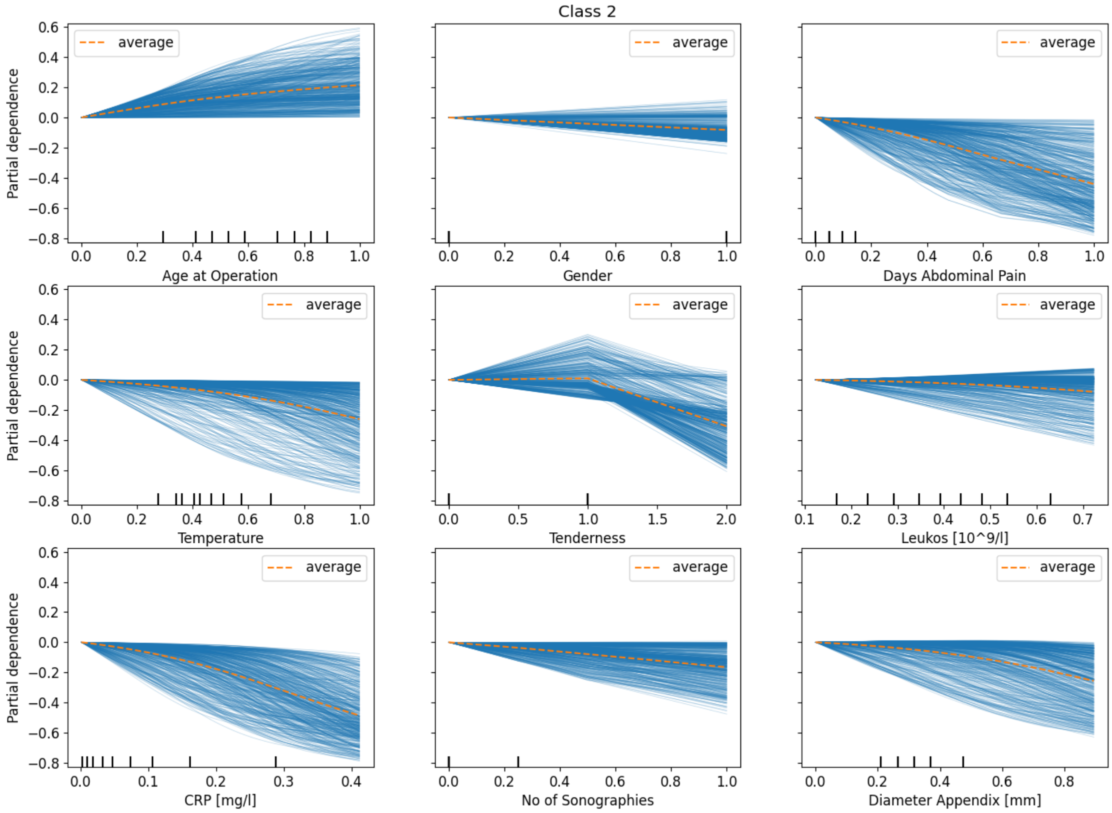


Figure 4.13 PDP/ICE centered plot for class 2 of neural network model trained on HD.

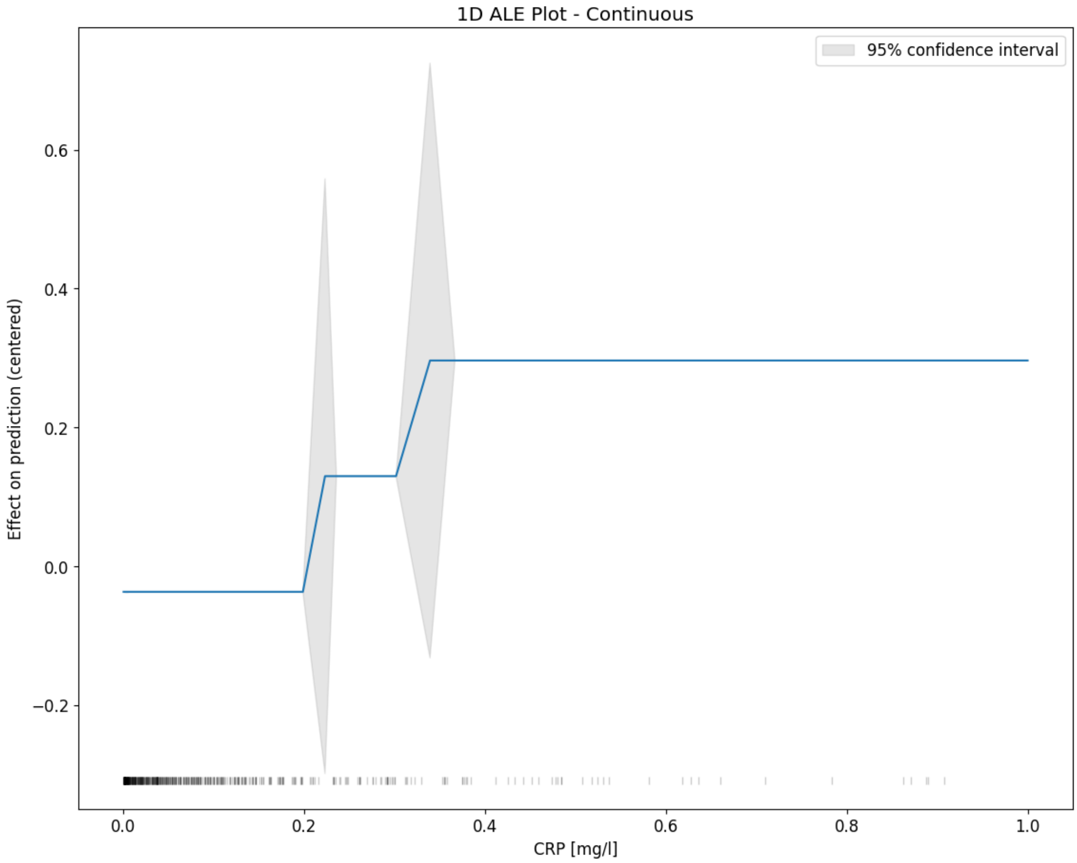


Figure 4.14 ALE plot for CRP feature of neural network model trained on HD.

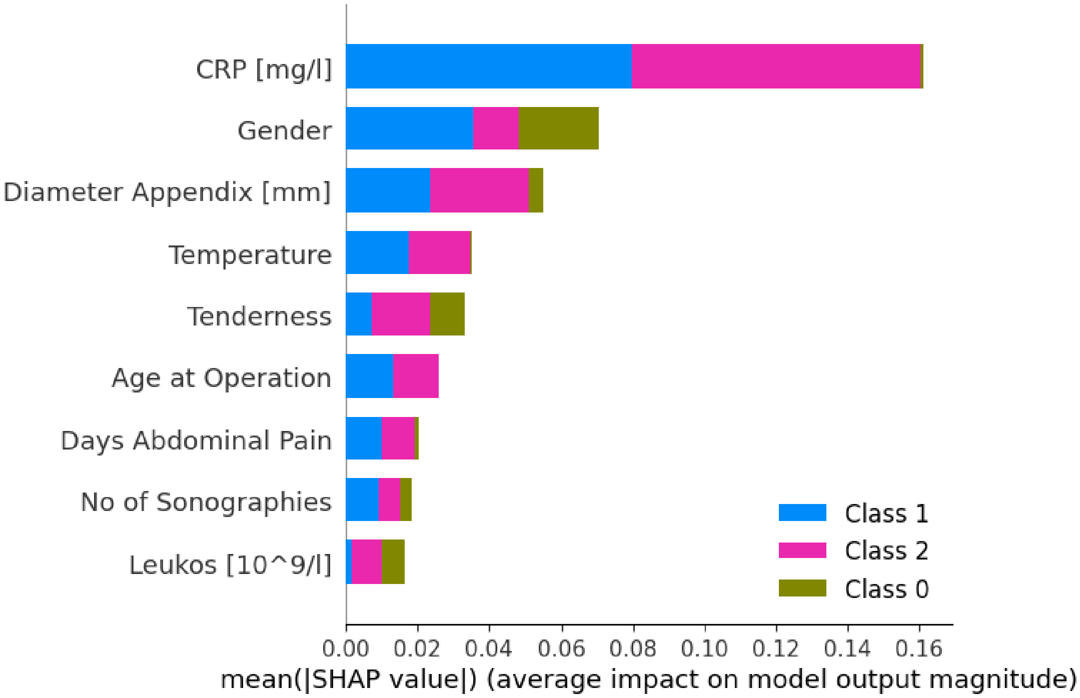


Figure 4.15 SHAP plot for neural network model trained on HD.

4.6 Additional explainability data

As it is not possible to display all gathered explainability visualizations and plots in this paper, the mentioned has been uploaded to the author's GitHub page [45]. The visualizations are available in `report` folder which consists of `NO_OS` and `OS_NN` folders which designate models trained on non-oversampled and oversampled data respectively. Each folder inside of the two mentioned folders contains four folders – one for each type of classified data – histology report, binary histology report, postoperative diagnosis and binary postoperative diagnosis. Each folder contains performance data and visualizations of machine learning models, with `dt_`, `oner_`, `seqcov_`, `rf_` and `nn_` prefixes marking the file's affiliation to decision tree, OneR, sequential covering, random forest and neural network model respectively.

Chapter 5

Discussion

This chapter describes aforementioned results. It tries to highlight the best machine learning model for each of the available datasets based on their performances, but based on their interpretability as well.

5.1 Performance

This section discusses the gathered performance metrics of all machine learning models combined with all available dataset variations. The problem of imbalanced data is present with all dataset variations and it represents the main factor that lowers the models' performances.

5.1.1 Histology multinomial classification

The model which proved to work best on histology multinomial classification dataset is neural network. It reached a respectable 81% in terms of accuracy in general on the test set while decision tree, OneR, sequential covering and random forest models reached somewhat lower accuracy of 74%, 70%, 63% and 76% respectively.

It is important to note that all models failed to classify any of the `Class 0` instances from the test set, except the sequential covering model which managed to classify only one true positive of the class (probably a result of its possibility to overfit in small regions of the feature space). However, when oversampling is introduced on this dataset instance, neural network model manages to devote itself to instances of `Class 0` a bit more, but in turn hampers its F1-score on instances of the remaining classes.

To sum up, the model that highlights among others is the neural network model without using oversampling technique during training. Furthermore, instances of `Class 0` proved to be extremely difficult to classify and the models would likely benefit if more instances of that, least frequent class, were added to the dataset.

5.1.2 Histology binary classification

Interesting thing happens when analysing histology binary classification dataset. Out of all trained models, only the neural network model in combination with oversampling and sequential covering model succeed in classifying some instances of `Class 0`. On the other hand, other models ignore those instances completely as it does not hamper their accuracy dramatically due to a very imbalanced dataset, as shown in figure 2.2o.

It is important to note that while both of the aforementioned models succeed in classifying some `Class 0` samples, their performance in classifying `Class 0` samples is not astonishing, but only better when put head to head with other models.

All things considered, the model that can be highlighted as the most appropriate for histology binary classification is neural network model trained on oversampled data. Sequential covering model's performance is close to the higlighted model's performance due to its possibility to overfit locally, but lacks some performance points when classifying `Class 0`. All the models would likely perform significantly better if the dataset was not imbalanced in such proportions.

5.1.3 Postoperative diagnosis multinomial classification

Most of the researched models fail completely while trying to classify this dataset's `Class 2` samples. To be more specific, all models except neural network model in combination with oversampling do so, but the mentioned model in turn fails completely when trying to classify `Class 1` samples. The problem is again probably caused by the imbalanced dataset – class distribution available in figure 2.1l.

Of the trained models, the one that stands out the most is random forest machine learning model. It achieves the best combination of accuracy in general and F1-scores per class.

Having said that, depending on one's interests, the optimal model could well be one of the interpretable ones – decision tree, OneR or sequential covering. They almost match random forest's performance metrics and are easily understood which is especially important in medicine-related predictions.

5.1.4 Postoperative diagnosis binary classification

Performance metrics on this dataset are somewhat better than the ones on the previous dataset. No models ignore samples of any class completely.

The model that can be highlighted is neural network model without oversampling during training – it has the best combination of F1-scores and its accuracy is respectable as well. In this case, oversampling attempt fails completely since all the performance metrics get worse.

Random forest model also achieves respectable results – while it is slightly more accurate than the neural network model without oversampling, its F1-scores are marginally worse.

All the interpretable models achieve respectable results as well. While sequential covering and decision tree models are closely matched, OneR stands out in a good way. While it achieves the same levels of performance metrics as the aforementioned non-interpretable models and exceeds the possibilites of other interpretable models, one has to ask himself whether he wants only one feature value to decide on his medical condition.

5.2 Model interpretability

The models highlighted based on their performance metrics in the previous section were mainly non-interpretable models – neural networks and random forest. Although they do not offer a dramatic improvement over interpretable models, their performance metrics are better in general which was expected. In cases where model intepretability is of high value, one would have to satisfy himself with an interpretable model of a lower performance. Instead, one could try to overcome the lack of interpretability by using explainability techniques which are the topic of this section.

5.2.1 PD and ICE plots

PD and ICE plots were visualized for neural network and random forest models for `Class 2` of the HD. Both plots were visualized in the same graph.

Random forest model's PD and ICE plots (figure 4.8) show that rise in CRP feature causes rise in prediction – model is more certain that class 2 should be predicted. ICE plots are used to confirm the individual effects were not cancelled out due to the fact that PD plot shows the averaged effects. This PD plot represents the situation on average well.

On the other hand, the influence of the CRP feature on neural network model's prediction (figure 4.13) is reversed relative to the CRP's influence in random forest model. As the CRP rises, the model is less certain the sample should be classified as a member of `Class 2`. PD plots show the averaged effect well.

5.2.2 ALE plots

ALE plots were visualized for neural network and random forest models for CRP feature. ALE plots should show a more faithful representation of the feature's influence on the prediction since no unrealistic data samples are taken into account while calculating ALE values.

Random forest model's ALE plot (figure 4.9) shows the similar situation as its PD and ICE counterparts. On the other hand, neural network model's ALE plot (figure 4.14) shows inverse effect of the CRP feature relative to the PD and ICE plot counterparts. That could mean that unrealistic data samples were created during the calculation of PD and ICE plots which resulted in unrealistic explanations in form of PD and ICE plots.

5.2.3 Feature importance

Generated feature importance plots were shown for all models except the OneR model since feature importance plot is not applicable to OneR (it only deals with one feature).

Interpreting feature importance plots is as simple as it can be – the higher the bar of a feature, the more important that feature is and vice versa. For example, the most important feature for random forest model trained on the HD is by far the CRP feature (figure 4.7a). Likewise, the most important feature for neural network model trained on the same dataset is also the CRP feature (figure 4.12a).

An interesting observation is a feature importance plot where all importances are zero (figure 4.12a) – present in neural network model trained on HBD without oversampling. One possible interpretation of such importances lies in the fact that the model always predicted the same class (can be seen in 4.11b). Because of that, no feature was important since the model will always predict the same class.

Feature importance plots can also be used to identify models that overfitted or do not make their predictions reasonably. For example, sequential covering model trained on HBD values the feature describing the age of the patient the most which should probably not be the case. Although medical expert's knowledge is required to evaluate that observation, feature importance explainability technique did its part in identifying possible problems in the model.

5.3 SHAP

SHAP plots were generated for random test set instances of neural network and random forest models. The plot shows each feature's influence on predicting the sample is a member of a given class.

For example, random forest model's SHAP plot (figure 4.10) shows that CRP feature has a bigger influence in possibility of predicting the given sample is a member of `Class 1` than of `Class 0`. Likewise, neural network model's SHAP plot (figure 4.15) shows that CRP feature influences the eventual `Class 1` and `Class 2` predictions, but has no influence on classifying the sample as a member of `Class 0`.

Chapter 6

Conclusion

Machine learning models solve a wide range of problems that are otherwise not solvable. Despite their impressive capabilities, we face the problem of lack of interpretability while using the machine learning models ever more often. It is crucial to understand and overcome interpretability based problems when the machine learning models are being used in areas where incorrect predictions can have enormous consequences. Some of these areas include medicine and healthcare, autonomous driving, finance, environmental science and in quality control.

This paper focuses on evaluating both model interpretability and performance on selected machine learning models – decision tree, OneR, sequential covering, random forest and neural network. All models are evaluated using the appendicitis dataset with four different available classification tasks. The paper summarizes each of the used machine learning models and explainability techniques. Furthermore, it highlights the optimal models and interpretability versus performance trade-off in each dataset variation.

Future work should include obtaining more data and explore additional oversampling techniques since the biggest issue for the models was unbalanced classes.

To conclude, this paper provides a perspective on the significance of interpretability in machine learning models and effectively addresses the balance between interpretability and performance.

Bibliography

[1] C. O'Sullivan, "Interpretable vs Explainable Machine Learning — towardsdatascience.com," https://towardsdatascience.com/interperable-vs-explainable-machine-learning-1fa525e12f48, 2020, [Accessed 27-02-2023].

[2] C. Molnar, *Interpretable Machine Learning*, 2nd ed., 2022. , s Interneta, https://christophm.github.io/interpretable-ml-book

[3] A. Garg and V. Mago, "Role of machine learning in medical research: A survey," *Computer Science Review*, vol. 40, p. 100370, May 2021. , s Interneta, https://doi.org/10.1016/j.cosrev.2021.100370

[4] J. A. Nichols, H. W. H. Chan, and M. A. B. Baker, "Machine learning: applications of artificial intelligence to imaging and diagnosis," *Biophysical Reviews*, vol. 11, no. 1, pp. 111–118, Sep. 2018. , s Interneta, https://doi.org/10.1007/s12551-018-0449-9

[5] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358, Apr. 2019. , s Interneta, https://doi.org/10.1056/nejmra1814259

[6] A. Vellido, "The importance of interpretability and visualization in machine learning for applications in medicine and health care," *Neural Computing and Applications*, vol. 32, no. 24, pp. 18 069–18 083, Feb. 2019. , s Interneta, https://doi.org/10.1007/s00521-019-04051-w

[7] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, vol. 2, no. 6, pp. 305–311, Jun. 2020. , s Interneta, https://doi.org/10.1038/s42256-020-0186-1

[8] E. Sorantin, M. G. Grasser, A. Hemmelmayr, S. Tschauner, F. Hrzic, V. Weiss, J. Lacekova, and A. Holzinger, "The augmented radiologist: artificial intelligence

in the practice of radiology," *Pediatric Radiology*, vol. 52, no. 11, pp. 2074–2086, Oct. 2021. , s Interneta, https://doi.org/10.1007/s00247-021-05177-7

[9] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, Jul. 2019. , s Interneta, https://doi.org/10.3390/electronics8080832

[10] "Interpretability versus explainability - Model Explainability with AWS Artificial Intelligence and Machine Learning Solutions — docs.aws.amazon.com," https://docs.aws.amazon.com/whitepapers/latest/model-explainability-aws-ai-ml/interpretability-versus-explainability.html, [Accessed 23-08-2023].

[11] A. Shaban-Nejad, M. Michalowski, and D. L. Buckeridge, "Explainability and interpretability: Keys to deep medicine," in *Explainable AI in Healthcare and Medicine*. Springer International Publishing, Nov. 2020, pp. 1–10. , s Interneta, https://doi.org/10.1007/978-3-030-53352-6_1

[12] M. M. Mijwil and K. Aggarwal, "A diagnostic testing for people with appendicitis using machine learning techniques," *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 7011–7023, Jan. 2022. , s Interneta, https://doi.org/10.1007/s11042-022-11939-8

[13] O. F. Akmese, G. Dogan, H. Kor, H. Erbay, and E. Demir, "The use of machine learning approaches for the diagnosis of acute appendicitis," *Emergency Medicine International*, vol. 2020, pp. 1–8, Apr. 2020. , s Interneta, https://doi.org/10.1155/2020/7306435

[14] C. Bunn, S. Kulshrestha, J. Boyda, N. Balasubramanian, S. Birch, I. Karabayir, M. Baker, F. Luchette, F. Modave, and O. Akbilgic, "Application of machine learning to the prediction of postoperative sepsis after appendectomy," *Surgery*, vol. 169, no. 3, pp. 671–677, Mar. 2021. , s Interneta, https://doi.org/10.1016/j.surg.2020.07.045

[15] E. Aydin, İ. U. Türkmen, G. Namli, Ç. Öztürk, A. B. Esen, Y. N. Eray, E. Eroğlu, and F. Akova, "A novel and simple machine learning algorithm for preoperative diagnosis of acute appendicitis in children," *Pediatric Surgery International*, vol. 36, no. 6, pp. 735–742, Apr. 2020. , s Interneta, https://doi.org/10.1007/s00383-020-04655-7

[16] R. Marcinkevics, P. R. Wolfertstetter, S. Wellmann, C. Knorr, and J. E. Vogt, "Using machine learning to predict the diagnosis, management and severity of pediatric appendicitis," *Frontiers in Pediatrics*, vol. 9, Apr. 2021. , s Interneta, https://doi.org/10.3389/fped.2021.662183

[17] R. M. Eickhoff, A. Bulla, S. B. Eickhoff, D. Heise, M. Helmedag, A. Kroh, S. M. Schmitz, C. D. Klink, U. P. Neumann, and A. Lambertz, “Machine learning prediction model for postoperative outcome after perforated appendicitis,” *Langenbeck's Archives of Surgery*, vol. 407, no. 2, pp. 789–795, Feb. 2022. , s Interneta, https://doi.org/10.1007/s00423-022-02456-1

[18] A. H. Omari, M. R. Khammash, G. R. Qasaimeh, A. K. Shammari, M. K. B. Yaseen, and S. K. Hammori, “Acute appendicitis in the elderly: risk factors for perforation,” *World Journal of Emergency Surgery*, vol. 9, no. 1, Jan. 2014. , s Interneta, https://doi.org/10.1186/1749-7922-9-6

[19] H. M. K. Ghomrawi, M. K. O’Brien, M. Carter, R. Macaluso, R. Khazanchi, M. Fanton, C. DeBoer, S. C. Linton, S. Zeineddin, J. B. Pitt, M. Bouchard, A. Figueroa, S. Kwon, J. L. Holl, A. Jayaraman, and F. Abdullah, “Applying machine learning to consumer wearable data for the early detection of complications after pediatric appendectomy,” *npj Digital Medicine*, vol. 6, no. 1, Aug. 2023. , s Interneta, https://doi.org/10.1038/s41746-023-00890-z

[20] J. J. Atema, C. C. van Rossem, M. M. Leeuwenburgh, J. Stoker, and M. A. Boermeester, “Scoring system to distinguish uncomplicated from complicated acute appendicitis,” *British Journal of Surgery*, vol. 102, no. 8, pp. 979–990, May 2015. , s Interneta, https://doi.org/10.1002/bjs.9835

[21] H. M. Krumholz, “Big data and new knowledge in medicine: The thinking, training, and tools needed for a learning health system,” *Health Affairs*, vol. 33, no. 7, pp. 1163–1170, Jul. 2014. , s Interneta, https://doi.org/10.1377/hlthaff.2014.0053

[22] H. He and E. Garcia, “Learning from imbalanced data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009. , s Interneta, https://doi.org/10.1109/tkde.2008.239

[23] V. Ganganwar, “An overview of classification algorithms for imbalanced datasets,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, pp. 42–47, 01 2012.

[24] P. Schober, C. Boer, and L. A. Schwarte, “Correlation coefficients: Appropriate use and interpretation,” *Anesthesia & Analgesia*, vol. 126, no. 5, pp. 1763–1768, May 2018. , s Interneta, https://doi.org/10.1213/ane.0000000000002864

[25] P. V. K. S. Miss. Mayuri S. Shelke, Dr. Prashant R. Deshmukh, “A review on imbalanced data handling using undersampling and oversampling technique,” *International Journal of Recent Trends in Engineering and Research*, vol. 3,

no. 4, pp. 444–449, May 2017. , s Interneta, https://doi.org/10.23883/ijrter.2017.3168.0uwxm

[26] "Imbalance dataset: Test and validate resampled tabular data — dagshub.com," https://dagshub.com/blog/imbalance-dataset-test-and-validate-resampled-tabular-data/, [Accessed 03-09-2023].

[27] R. Sharma, "Oversampling/Undersampling only train set only or both train and validation set — datascience.stackexchange.com," https://datascience.stackexchange.com/a/63255, [Accessed 03-09-2023].

[28] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[29] scikit learn, "1.10. Decision Trees — scikit-learn.org," https://scikit-learn.org/stable/modules/tree.html, [Accessed 30-08-2023].

[30] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* Springer New York, 2009. , s Interneta, https://doi.org/10.1007/978-0-387-84858-7

[31] A. Alzu'bi, H. Najadat, W. Doulat, O. Al-Shari, and L. Zhou, "Predicting the recurrence of breast cancer using machine learning algorithms," *Multimedia Tools and Applications*, vol. 80, no. 9, pp. 13 787–13 800, Jan. 2021. , s Interneta, https://doi.org/10.1007/s11042-020-10448-w

[32] A. Aneeshkumar and C. J. Venkateswaran, "Reverse sequential covering algorithm for medical data mining," *Procedia Computer Science*, vol. 47, pp. 109–117, 2015. , s Interneta, https://doi.org/10.1016/j.procs.2015.03.189

[33] M. M. MALIK and H. HAOUASSI, "Efficient sequential covering strategy for classification rules mining using a discrete equilibrium optimization algorithm," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 7559–7569, Oct. 2022. , s Interneta, https://doi.org/10.1016/j.jksuci.2021.08.032

[34] R. Genuer and J.-M. Poggi, *Random Forests with R.* Springer International Publishing, 2020. , s Interneta, https://doi.org/10.1007/978-3-030-56485-8

[35] T. Yiu, "Understanding Random Forest — towardsdatascience.com," https://towardsdatascience.com/understanding-random-forest-58381e0602d2, 2019, [Accessed 30-08-2023].

[36] J. Singh, "Random Forest: Pros and Cons — medium.datadriveninvestor.com," https://medium.datadriveninvestor.com/random-forest-pros-and-cons-c1c42fb64f04, 2020, [Accessed 30-08-2023].

[37] A. Subasi, "Machine learning techniques," in *Practical Machine Learning for Data Analysis Using Python.* Elsevier, 2020, pp. 91–202. , s Interneta, https://doi.org/10.1016/b978-0-12-821379-7.00003-5

[38] "1.17. Neural network models (supervised) — scikit-learn.org," https://scikit-learn.org/stable/modules/neural_networks_supervised.html, [Accessed 30-08-2023].

[39] J. H. Friedman, "Greedy function approximation: A gradient boosting machine." *The Annals of Statistics*, vol. 29, no. 5, pp. 1189 – 1232, 2001. , s Interneta, https://doi.org/10.1214/aos/1013203451

[40] R. Wright, "Interpreting black-box machine learning models using partial dependence and individual conditional expectation plots," *Exploring SAS® Enterprise Miner Special Collection*, vol. 1950, 2018.

[41] Y. Nohara, Y. Wakata, and N. Nakashima, "Interpreting medical information using machine learning and individual conditional expectation," *Stud. Health Technol. Inform.*, vol. 216, p. 1073, 2015.

[42] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation," *Journal of Computational and Graphical Statistics*, vol. 24, no. 1, pp. 44–65, Jan. 2015. , s Interneta, https://doi.org/10.1080/10618600.2014.907095

[43] T. Danesh, R. Ouaret, P. Floquet, and S. Negny, "Interpretability of neural networks predictions using accumulated local effects as a model-agnostic method," in *Computer Aided Chemical Engineering.* Elsevier, 2022, pp. 1501–1506. , s Interneta, https://doi.org/10.1016/b978-0-323-95879-0.50251-4

[44] E. Winter, "Chapter 53 the shapley value," in *Handbook of Game Theory with Economic Applications.* Elsevier, 2002, pp. 2025–2054. , s Interneta, https://doi.org/10.1016/s1574-0005(02)03016-3

[45] I. Rubinić, "GitHub - ivancrg/mlp: Diplomski — github.com," https://github.com/ivancrg/mlp, 2023, [Accessed 09-09-2023].

Abstract

This paper includes work on classification of appendicitis from the point of view of histological analysis and postoperative diagnosis. The paper provides an overview of various machine learning models – decision trees, OneR, sequential covering, random forests and neural networks. Four different versions of the dataset were used during the experiments. Uninterpretable models were identified and their respective visualizations were created using explainability techniques which serve to provide explanations on their decision making.

Keywords **— appendicitis, machine learning, interpretability**

Sažetak

Ovaj rad se bavi klasifikacijom upala slijepog crijeva s gledišta histološke analize i postoperativne dijagnoze. Rad obuhvaća pregled raznih modela strojnog učenja – stabla odluke, OneR, uzastopnog pokrivanja, nasumičnih šuma te neuronskih mreža. Tijekom provođenja eksperimenata korištene su četiri različite inačice skupa podataka. Neinterpretabilni modeli su identificirani te su za njih tehnikama za postizanje objašnjivosti modela stvorene vizualizacije koje služe za pojašnjenje njihovog rada.

Ključne riječi **— upala slijepog crijeva, strojno učenje, interpretabilnost**