

Distributional Framework for the Shapley Value in Context of Data Valuation Methods for Machine Learning Applications

Bachelor Thesis

by

Alexander Munker

Degree Course: Industrial Engineering and Management

Institute for Applied Informatics and Formal
Description Methods (AIFB)

KIT Department of Economics and Management

Advisor:	Prof. Dr. Ali Sunyaev
Second Advisor:	Prof. Dr. Andreas Oberweis
Supervisor:	Konstantin Pandl
Submitted:	26. October 2021
Matriculation number:	1953511

Abstract

This thesis focusses on data valuation of a medical classification data set for a machine learning application. Recent research has come up with data valuation methods, which use a notion from the cooperative game theory, the Shapley value, to evaluate the worth of data points within a data set.

A completely new approach, the Distributional Framework, has been developed in 2020. The method is based on the Shapley value as well and provides new properties. It is the goal of this thesis to evaluate its empirical effectiveness in performing data valuation. Furthermore, as this new approach seems to be able to influence the way data can be traded, it may stimulate future research concerning one of the most important challenges for machine learning and Artificial Intelligence: to find a way to get sufficient data for the training of machine learning models.

After implementing the Distributional Framework on the “AIforCOVID” data set for a binary classification task, several experiments are performed. In conclusion, the Distributional Framework seems to be able to value data successfully regarding value assignment. Still, the Distributional Framework faces some challenges concerning runtime and applicability, which can be seen as potential goals for research to solve in the future.

Table of Content

List of Abbreviations.....	IV
List of Figures	V
List of Tables.....	VI
1. Introduction	1
1.1. Problem Definition	1
1.2. Research Objectives	1
1.3. Method and Structure	2
1.4. Related Work.....	2
2. Background on Machine Learning	3
2.1. Overview	3
2.2. Logistic Regression	5
2.3. Neural Network	7
3. Background on Data Valuation.....	9
3.1. Overview	9
3.2. Data Shapley.....	11
3.3. Distributional Framework for the Shapley Value	13
3.4. Data Marketplaces and Implications	14
4. Structure for Analyzing Data Valuation Methods	16
4.1. Model Architecture.....	16
4.2. Dataset and Training.....	17
4.3. Implementation of the Data Valuation Methods	19
5. Applications and Experiments	21
5.1. Distribution of the Estimations.....	21
5.2. Runtime Comparison.....	22
5.3. Point Removal Experiment.....	22
6. Discussion of Experiments.....	24
6.1. Principal Findings.....	24
6.2. Implications for Research and Practice	24
6.3. Limitations and Future Research.....	25
7. Conclusion.....	26
Declaration about the Thesis	28

List of Abbreviations

AUC	Area under the ROC
CNN	Convolutional Neural Network
DF	Distributional Framework for the Shapley value
FN	False Negative
FP	False Positive
KNN-Shapley	Estimation procedure for computing Shapley values based on K-nearest neighbors method
LOO	Leave-one-out method
ML	Machine Learning
NaN	Not a number
SV	Shapley value
TN	True Negative
TP	True Positive
ROC	Receiver Operating Characteristic Curve
TMC-Shapley	Estimation procedure for computing Shapley values based on truncated Monte-Carlo approximations

List of Figures

Figure 1: Receive Operating Characteristic	5
Figure 2: Sigmoid Function.....	6
Figure 3: Feed Forward Neural Network	7
Figure 4: Convolutional Neural Network.....	8
Figure 5: Ingredients for Data Valuation.....	9
Figure 6: Ingredients for Data Valuation with the Distributional Framework.....	13
Figure 7: ROC for Logistic Regression.....	19
Figure 8: Histograms of the Shapley Values.....	21
Figure 9: Runtime Comparison of the Distributional Framework and the TMC-Shapley	22
Figure 10: Point Removal Experiment - Results	23

List of Tables

Table 1: Confusion Matrix	4
Table 2: Features of "AIforCOVID" Data Set	18
Table 3: Confusion Matrix for a Logistic Regression	19

1. Introduction

1.1. Problem Definition

In recent years, machine learning (ML) has become a promising technology in the healthcare sector, for example, in context of classification tasks. Since large data sets are indispensable to train the ML models, data becomes the fuel driving technological and economic growth. Data marketplaces can enable the collection of such large datasets. For example, in healthcare and consumer markets, it has been suggested that individuals should be compensated for the data that they generate, but it is not clear what is an equitable valuation for individual data. Thus, a fundamental challenge is how to quantify the value of data in algorithmic predictions and decisions. Several data valuation methods have been developed and tested on a number of classification tasks in the recent past. Especially data valuation methods based on the Shapley value (SV) [1] seem to be promising. However, a key limitation of the traditional methods is that their estimations are dependent on the data set on which they are trained on. They do not consider statistical features of the data [2]. To overcome these weaknesses, a new concept of data valuation has been developed, the Distributional Framework for the Shapley value (DF). The value of a data point, in this method, is defined in context of an underlying data distribution. In practice, this means that the DF is capable to value data outside the dataset [2] and therefore privacy concerns which are especially present in the field of healthcare could be overcome. These properties could help to make a step towards data marketplaces, where data can be traded fairly for the use of machine learning applications.

However, it remains uncertain, how well this approach is applicable in practice and how this method performs on an extensive medical data set.

In my thesis, I want to evaluate the practicability and the advantages of the DF in comparison to other data valuation methods by applying it on a medical classification task. Hence, the following research question is asked:

How efficient is the Distributional Framework in valuing medical data sets for machine learning applications?

1.2. Research Objectives

As this thesis focuses on the evaluation of the empirical effectiveness of the DF, it is the goal to successfully implement the DF on a medical classification task. Therefore, the work concentrates on a binary classification task on the “AIforCOVID” data set.

It is the goal to calculate Shapley values by the DF and the TMC-Shapley to understand if the DF is able to value data points correctly. In addition, it is intended to classify the runtime performance of the DF. Lastly, it is the goal to understand how good, or even better than the TMC-Shapley, the DF is able in

valuating data. These objectives help to evaluate the empirical effectiveness of the DF. A different learning approach on another data set brings further insights about the applicability of the DF. Furthermore, it is the target to understand more about data marketplaces and the possible role that the DF could take on in this context.

1.3. Method and Structure

Three ingredients are required to perform data valuation on a data set: a train data set, a learning algorithm, and an evaluation metric (Chapter 3.1). Therefore, for the understanding of this thesis, it is important to describe these first. In Chapter 2, the basics of ML and the model that is used during the practical work get introduced. Afterwards, the basics of data valuation and the most common methods as well as the DF get presented in Chapter 3. The pre-processing steps and the implementation of the data valuation methods get explained in Chapter 4 before carrying out various experiments in Chapter 5. Lastly, the results are discussed in Chapter 6 before concluding the thesis in Chapter 7. All the calculations of the ML model are performed with the programming language Python and Tensorflow as a library source. To increase the CPU and GPU performance as well as the storage capacities for the calculations, the entire work is carried out on the GPU computing nodes of the bwUniCluster 2.0, a high-performance computing cluster of the universities of Baden-Württemberg.

1.4. Related Work

The Shapley value is an important component in this study. It is a meaningful instrument in the field of game theory which belongs to the science of economics [3]. It has been applied to a variety of problems including resource allocation, voting, and bargaining [4]. More interesting for this thesis is the development and research in context of data valuation. Here, literature focusses on the scientific question on how to equally value data [5]. Several methods have been introduced so far. Next to the data valuation method, which is based on the Shapley value, various approximation methods have been developed [6,7]. More recently, a new method has been presented that faces the limitations of the other data valuation methods. The method is called the Distributional Framework for the Shapley value [2]. As previous work focuses on reducing the computational effort to value data, the latest work in this scientific area aims on data valuation, which is independent from the data set, on which the ML model is trained on [2]. This brings significant advantages, which will be discussed later. Furthermore, research has advanced in the design and implementation of data marketplaces [8]. Since the DF has beneficial properties in context of privacy concerns [2], future data marketplaces may be influenced by the DF. Hence, the development and design of data marketplaces may seek more intention in the future, especially in the field of healthcare.

2. Background on Machine Learning

2.1. Overview

To understand how data valuation works, it is necessary to describe the basics of ML first. ML can be assigned to Artificial Intelligence, a growing field in computer science. In the recent past, ML has become one of the most researched topics. Scientists of almost all sectors are interested in this technology and aim to use ML specifically for their research now or in the future.

ML means that a computer can automatically learn, find patterns, and improve certain tasks from experience. Experience, in this context, can be translated to data [9]. To make a computer learn certain tasks for itself, it needs large amounts of data. Therefore, data is essential for the success of ML. Before taking a closer look at the mechanisms of machine learning, a vast overview of the sub methods of ML will be presented. There are three common types of ML:

- Supervised machine learning
- Unsupervised machine learning
- Reinforcement machine learning

Supervised Learning needs a labelled dataset to learn a task. “Labelled” means that the data set consists of input/output pairs (x_i, y_i) . It is the goal to develop a function that can predict the right outcome when giving the input data to the function. Supervised learning is often applied to data classification and regression tasks. Unsupervised learning uses, in contrast to supervised learning, only unlabelled data. Data clustering, for example, is a common task on which this approach is applied. Reinforcement learning uses a reward and punishment strategy. Using the reward, a utility function gets approximated. The agent, in fact, learns a strategy by himself. The focus in this work lies on supervised learning as the dataset, which is used, is labelled. A supervised machine learning model consists of 4 components [10]:

1. A feature representation of the input. For each input observation this will be a vector of features.
2. A classification function that computes y_i , the estimated class. Sigmoid or SoftMax are common tools for this computation (Chapter 2.2.).
3. An objective function for learning, usually involving minimizing the error on training examples. The cross-entropy loss function will get introduced in Chapter 2.2.
4. An algorithm for optimizing the objective function (for example the gradient descent algorithm).

With these components, a typical ML process can be divided into several basic steps. Firstly, the data gets infiltrated into the model where calculations are performed by using component 1 and 2. The model tries to develop a pattern on how to solve the task. Secondly, a loss function gets implemented which measures the difference between the result of the model and the actual outcome (component 3). The loss

function tries to calculate how bad the miss is on each particular guess. The last component is the optimization process, where the algorithm takes the miss and then updates the parameters to achieve a better guess, so that the miss is smaller than before (component 4).

As mentioned before, data plays a major role in every machine learning process. Most of the times, the data must get pre-processed before running it into the model. Furthermore, the data set get split into a training and test data set. A small part of the training data will be used to validate the training phase. The training data set is used to train the model on a specific task. The test data set will be used to test the performance of the model on new, unseen data. Quantitative performance metrics aim to analyse the model's performance. As this thesis focusses on a binary classification task, further explanations focus especially on this task. The performance metric is measured on the test data set. A common metric for a binary classification task is the confusion matrix (Table 1), which is a tabular visualization of the truth labels versus the model predictions [11]. To build this matrix, 4 key figures are necessary. The True Positive number signifies how many positive samples the model predicted correctly. The True negative number states how many negative samples have been predicted correctly. The False positive and False negative numbers signify how many class samples have been predicted incorrectly.

	P' (predicted)	N' (predicted)
P (actual)	True Positive	False Negative
N (actual)	False Positive	True Negative

Table 1: Confusion Matrix

With the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values further metrics for binary classification tasks can be calculated. An important metric is the accuracy. The accuracy gets calculated as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad \text{True Positive Rate} = \frac{TP}{TP+FN} \quad \text{False Positive Rate} = \frac{FP}{FP+TN}$$

With these performance indicators, the ROC (receiver operating characteristic curve), another evaluation metric, curve can be plotted (Figure 1). The curve plots two parameters, the True Positive rate, and the False Positive rate and shows the trade-off between hits and costs. Above the straight line, two potential ROC curves have been plotted.

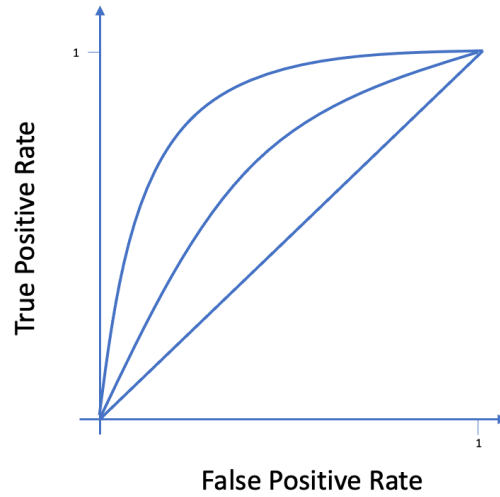


Figure 1: Receive Operating Characteristic

Building on the ROC, another evaluation metric for this study can be calculated, the AUC (Area under the ROC Curve). It measures the entire two-dimensional area underneath the ROC curve from (0,0) to (1,1). It is used in Chapter 4.2. The AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0. The AUC is used as a summary of the ROC curve and should be above 0.5 for the model's performance to be useful.

2.2. Logistic Regression

Logistic Regression is a supervised ML classification algorithm [10]. It is used to assign observations to a discrete set of classes. It has its origin in the field of statistics. There are three types of Logistic Regressions: the binary Regression model, which has only two possible types of dependant variables, in fact pass or fail. The Multi-Logistic Regression model has 3 or more possible unordered types of dependant variables. The ordinal Logistic Regression model is similar to the Multi-Logistic Regression model. However, it can decide between ordered types of dependant variables (for example “low”, “medium” or “high”).

$$z = (\sum_{i=1}^n w_i x_i) + b$$

Equation 1

$$y = \sigma(z) = \frac{1}{1+e^{-z}}$$

Equation 2

Logistic Regression solves the classification task by learning, from a train data set, a vector of weights and a bias term which are shown in Equation 1. Each weight w is a real number and associated with one of the input features x_i . The weight represents how important each input feature is for the classification task. Its value can be negative or positive. The bias b is another value, which is added to the weighted

inputs. First, the input gets multiplied by the weight and then the bias gets added. Z represents that value and lies between 0 and 1. To create a probability, z gets passed through the Sigmoid function [10], which is shown in Figure 2. Equation 2 shows the mathematical formula of the Sigmoid function.

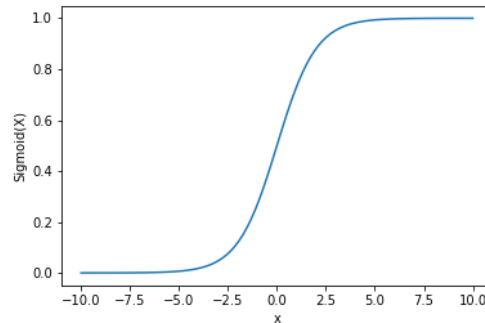


Figure 2: Sigmoid Function

Furthermore, a decision boundary is needed as a component for the Logistic Regression. The boundary is typically 0.5, so that $y > 0.5$ leads to class 1 and $y \leq 0.5$ leads to class 2. However, the threshold value can also be assigned independently (for example 0.7). Below that threshold point, values get classified into class 1. Every value which is above the threshold gets assigned to class 2.

The Logistic Regression has two phases. In the first phase, the training phase, the system gets trained using stochastic gradient descent and the cross-entropy loss [10]. The model learns the parameters (weights w and bias b) during this phase. In the second phase the trained model gets tested. Therefore, given a test sample, the probability gets computed and the higher probability, label $y = 1$ or $y = 0$ gets returned (class 1 or 2).

The system produces \hat{y} , which is the system's estimate of the true y . The learning process requires two components, a loss function, and an optimizer. The first metric measures how close the estimation \hat{y} is from the actual outcome y . A loss function which is often used for a Logistic Regression is the cross-entropy loss. Furthermore, an optimizer is needed which is in the case of a Logistic Regression the stochastic gradient descent algorithm. The goal of the gradient descent algorithm is to find the optimal weights. Hence, the algorithm minimizes the loss function. The loss function of a Logistic Regression is convex [10]. This means that the function only has one minimum which makes it simple to find it.

In conclusion, Logistic Regression forms the basis of machine learning along with other algorithms [12]. Neural networks, which will be introduced in Chapter 2.3., were developed on top of Logistic Regressions. Logistic Regressions can be used for a variety of problems and are therefore of great interest for this thesis. Recent research showed that it is especially the field of healthcare, where Logistic Regressions have gained more and more impact [13].

2.3. Neural Network

A neural network is an ML architecture, which is inspired by the structure of neurons in the brain. In general, a neural network breaks down the input data into layers of abstraction. The classic neural network is the feedforward model. Using this architecture, the basic structure of a neural network will be explained in the following.

A typical feedforward neural network consists of three types of layers, which is shown in Figure 3. The input layer brings the initial data into the system. There must be always one input layer in a neural network. The hidden layer is located between the input and the output layer. When there is more than one hidden layer placed in the neural network, it is referred to as deep learning [14]. The hidden layers can be seen as the real computational engine of a neural network. The output layer is responsible for producing the final output. An output layer is necessary for every neural network. The output layer is responsible to release information to the outside world. As one can see in Figure 3, information can only move forward in this neural network as there are no cycles or loops. However, there are also neural network architectures where loops exist. These neural networks are often more complex and used for more difficult tasks.

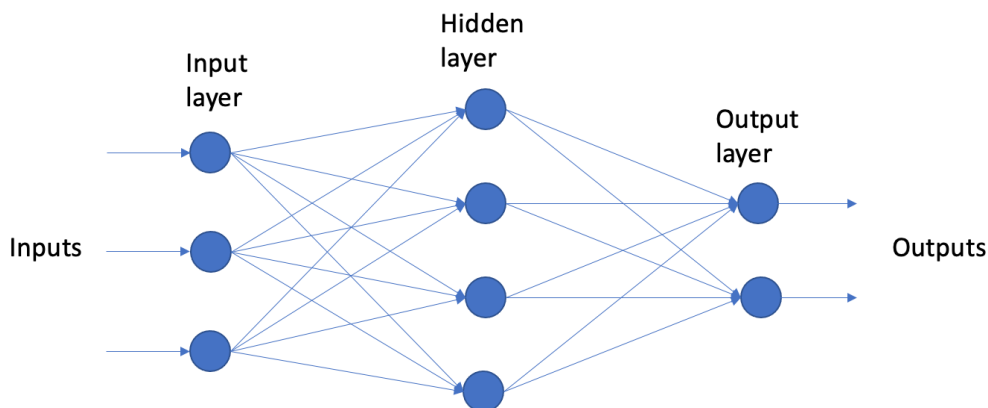


Figure 3: Feed Forward Neural Network

Nodes are the basic unit of computation in a neural network. In Figure 3, the nodes are presented as blue dots. They receive input and pass it on to the next layers. Weights w are also important in the structure of a neural network. They can be seen as parameters which transform the input data as it get passed through the hidden layers. Weights are allocated within the nodes. The value of a weight gets multiplied with the input data. Often, next to the weights, also a bias b is placed inside the node to further transform the input data. In comparison to the weight, the bias gets added to the input data [14].

To produce an output-signal an activation function is used for every node. The activation function f can be linear or non-linear. The purpose of non-linear activation functions is to bring non-linearity into the system and make the network more complex as real-world problems often occur to be also complex.

Common activation functions are the Sigmoid-function, the Tanh-function, and the ReLu-function. In total, the term in a node looks like this: $f(\sum_i w_i x_i + b)$ [14].

There are different learning algorithms for the neural network to improve its performance in accomplishing its task. An example for a learning mechanism is the gradient descent method, which iteratively updates all learnable parameters, for example the weights, to minimize the loss.

A Convolutional Neural Network (CNN) is a type of deep learning, which imitates how the visual cortex of the brain processes images. Since, next to the Logistic Regression, it is intended to apply the DF on a medical image classification task in addition, it seems important to introduce the basics of a CNN. A CNN is especially efficient in processing data that has a grid pattern, such as images. In digital images, pixel values are stored in a two-dimensional grid. A CNN typically consists of three types of layers: a convolutional layer, a pooling layer, and a fully connected layer [15]. This structure is shown in Figure 4. A CNN has an optimizable feature extractor, which makes it possible to identify specific features everywhere in the image. The benefits of a CNN have been used in a variety of areas so far. However, CNNs seem to have great potential especially in the field of healthcare, since they are especially efficient in classification tasks. In context of this thesis, this means that recognizing whether a certain disease is present when looking at lung scans is the same as classifying the image into two classes, one for “disease” and one for “no disease”. However, the CNN does not directly use the images as this would lead to poor results. The images get processed as they get passed through the different layers.

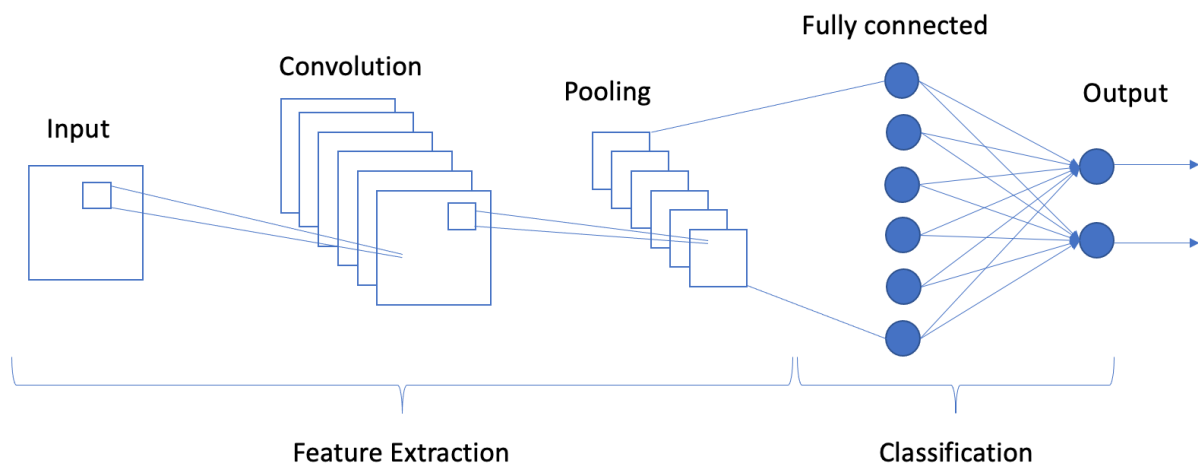


Figure 4: Convolutional Neural Network

The convolutional layer contains filters which convert the image. These convolutional filters are two-dimensional matrices (for example 5x5 or 3x3 matrices). The values of the filter matrices get determined during the training stage. The next layer after the convolutional layer is the pooling layer. Its task is to reduce the image size by combining the neighboring pixels of a specific area of the image to a single representative value. The value could be for example the maximum value (max pooling) or the mean value (mean pooling) [15]. In comparison to classic neural networks there is no learnable parameter in

any of the pooling layers. However, stride, filter size and padding are parameters which are similar to the convolution operations. In the last block of layers, often referred to as the fully connected layer, the output of the last convolutional layer or pooling layer gets transformed into a one-dimensional array of numbers. It is connected to the fully connected layer, where the final output of the network is generated. This could be for example the probability for each class in an image classification task. There is also a last layer activation function which targets class probabilities [15]. The implementation of a CNN on the “CheXpert” data set is discussed in Chapter 4.1.

3. Background on Data Valuation

3.1. Overview

Data valuation focusses on how to evaluate the worth of a single data point in a given data set. In fact, the question which is set to be asked is what impact one single data point has on the performance of an ML model. This thesis focusses on data valuation for a supervised ML task. Hence, the basics of data valuation will be explained in context of a supervised ML task only.

But why is data valuation important and seeking increasingly more interest in context of ML? As it has been mentioned before, ML requires large data sets to learn specific tasks. It has been shown that a big problem why ML often fails or will not get implemented is because of the lack of data.

Latest research focuses on the idea of implementing data marketplaces, where data buyers and sellers can come together and buy/sell their data [8]. However, to give each dataset a price, the value of each data instance must get determined first, which can be achieved by several data valuation methods. Next to that, data valuation can be useful to improve the model’s performance as it can identify low value data points. A point removal experiment, where this is explained more, is carried out in Chapter 5.

Generally, three ingredients are crucial to perform data valuation on a data set, which is shown in Figure 5. The figure is inspired by the illustrations used by Amirata Ghorbani in his lectures about the Distributional Framework.

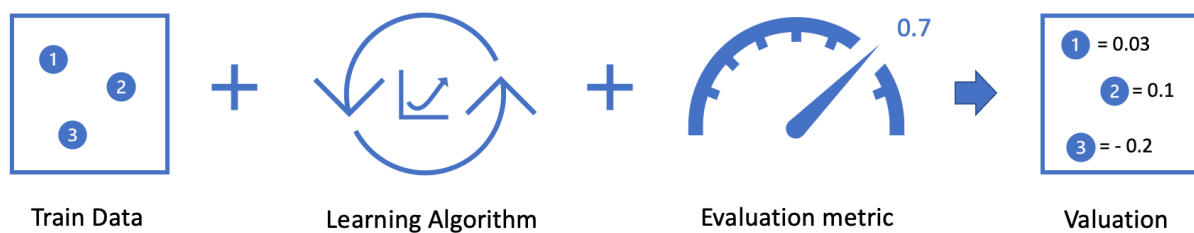


Figure 5: Ingredients for Data Valuation

The first ingredient is the train data $B = \{z_i\}_{i=1}^N$, which consists of all the training points z_i . The training points are composed of feature-label pairs (x_i, y_i) . At this point, the data is fixed without distributional

assumptions about B . The next ingredient is a learning algorithm A . It is necessary for the ML model to learn the task. The learning algorithm consists, among other modules, of a cost function and an optimizer, which have been introduced in Chapter 2.2. in context of a Logistic Regression. They transform the model's weights and biases so that its performance reaches its maximum. The last ingredient is an evaluation metric U . As discussed in Chapter 2.1., an evaluation metric measures the total performance of the model. The score $U(S, A, B_{val})$ of the evaluation metric is calculated on the test set and measures the performance of the train set. S represents a subset from the dataset B ($S \subseteq B$) and A stands for a learning algorithm. With these three components it is possible to calculate a value for the contribution of each training point towards achieving an overall performance score.

The value for each data instance is described in Equation 3. B_{val} is another variable, that represents the validation data set.

$$v(z_i, B, A, B_{val}, U), \text{ short } v(z_i).$$

Equation 3

For a fair data valuation, three axioms have been developed [7]. They are called Shapley axioms. The 3 axioms get introduced in the following paragraph.

1. Group rationality

$$U(B) = \sum_{z_i \in B} v(z_i)$$

2. Fairness

- a) $U(S \cup \{z_i\}) = 0$, for all $S \subseteq B \setminus \{z_i\}$, then $(v(z_i) = 0)$
- b) $\varphi(z) = v(z)$ if $U(S \cup \{z\}) = U(S \cup \{z\})$

3. Additivity

$$v(z, U_1 + U_2) = v(z, U_1) + v(z, U_2), \text{ for all } z \in B$$

The first axiom says that the contributions must add up to the difference of prediction for x and the average. The fairness axiom states that the contributions of two features must be the same if they contribute equally to all possible coalitions. Furthermore, when a feature does not change the predicted value, then it should have a SV of zero. The last axiom says that for a game with combined payouts, the respective overall SV is the sum of each single SV contributed by each player. These principles are based on the mathematical properties of the SV [1,3]. It is important to understand that the SV and the data valuation methods based on the SV satisfy these axioms.

In the following section different data valuation methods get presented. First, the Leave-one-out method (LOO) is introduced as a classic data valuation method. The method is not based on the Shapley value and therefore also does not suffice the Shapley axioms, which have been presented above

[6]. However, it is still worth showing the principle to understand how data valuation can be performed without the use of concepts of the SV in general. The LOO is an uncomplicated way to value data. The model gets trained on the train data set. Afterwards, the performance metric will be estimated on the test data set. To value a specific data point, the model gets re-trained on a dataset without this data instance. The performance then gets re-evaluated. Finally, the LOO-score is calculated by comparing the performance metric scores as can be seen in the following formula [6].

$$v_{LOO} = U(B) - U(B \setminus \{z_i\})$$

Equation 4

The method can be applied to every other data point within the train data. Unfortunately, the LOO faces two limitations. It does not satisfy the properties for an equitable valuation scheme as it is described above, and it produces high computational costs. For instance, when there are N data points in a dataset, to value every data point, the model needs to get retrained N times. Hence, especially for large data sets, the model seems to be unsuitable for a data valuation task [6].

3.2. Data Shapley

The Data Shapley is a data valuation method based on the SV to create a fair value for every single data point in a given data set as this is desirable as discussed in 3.1. (Shapley axioms) [5]. To understand the mechanism of the Data Shapley it is helpful to discuss the concept of the SV first.

The Shapley value is a solution concept in cooperative game theory for dividing a total payoff/ cost between players, assuming that they all collaborate [1]. The allocation for player i is proportional to the player's contribution for the game, i.e., how much value player i creates. It can also be used in the context of ML to assign a value to every data point in context of the overall performance of a supervised ML model. Each data instance is modeled as a player in a coalitional game. The value of a specific data point from any subset of contributors is measured by a utility function.

As described in Figure 5, the Data Shapley needs three ingredients to perform data valuation for a machine learning task: a fixed training data set of points; a learning algorithm; and a performance metric that measures the overall value of a trained model.

The general concept of the Data Shapley can be summed up as follows. In the first step, all the possible subsets from B that do not include i , the data point which is intended to value, are needed. For each of these subsets, the incremental value that arises when i is added back in, gets calculated. Lastly, all the value increments get summed up and divided by the number of subsets to yield the average value increment for a subset of the training data when a data point i is added to it. The following formula describes this method in a mathematical notation ($|B| = m$).

$$v_{\text{shap}}(z; U, B) \triangleq \frac{1}{m} \sum_{k=1}^m \frac{1}{\binom{m-1}{k-1}} \sum_{\substack{S \subseteq B \setminus \{z\}: \\ |S|=k-1}} (U(S \cup \{z\}) - U(S))$$

Equation 5

A big advantage of the Shapley value is that it fulfills the desired properties: fairness, rationality and additivity as described above. This leads to an equitable valuation scheme.

The data Shapley value of a point $z \in B$ is a weighted empirical average over subsets $S \subseteq B$ of the marginal potential contribution of z to each S [2]. Generally, the Data Shapley showed in various experiments its capability in successfully valuating data on real data sets [5, 6].

In comparison to the LOO method, empirical experiments have shown that Shapley-based methods are more effective in identifying valuable data points [6]. However, the classic Data Shapley method faces serious disadvantages when implementing it on a large data set. Evaluating the exact SV involves computing the marginal contribution of each training point to all possible subsets, whose complexity is $O(2^N)$. The term of the Data Shapley has an exponentially large number of terms, and this implies that computing the SV exactly is NP-hard in general [7]. Solving NP-hard problems means that the computational effort is exponential high which is unsatisfactory in terms of practicability. This problem is especially present when the data sets get bigger.

To solve this problem, approximation methods of the SV have been developed, which aim to value data as good as the classic method but are designed to reduce time and computational effort [5]. One method is called the TMC-Shapley and it combines two tools to minimize the computational and timing effort. When using Monte-Carlo simulations, the value of sample i can be estimated only by looking at some subsets of the data. These subsets are randomly sampled and can provably approximate the rest of the data. This setting is combined with the concept of truncation. This approach is to end the training early if the marginal value of the ML model is not increasing significantly any further. This is especially helpful for large datasets. It has been shown in a variety of experiments that the TMC-Shapley is significantly more successful in valuating data than the LOO-method [5]. Hence, the TMC-Shapley is used as a benchmark in this thesis to evaluate the performance of the DF.

Another approximation method is the K-nearest neighbors data valuation method (KNN-Shapley), which is also based on the Shapley value [16]. Since this method is not the core of this work, it will not be discussed further in this chapter.

Even though the Data Shapley method and the TMC-Shapley as an approximation method satisfy the Shapley axioms, they face practical disadvantages. In fact, the valuation reacts very sensitive to a change in B , that means when the data set changes. Given another dataset B' ($B' \neq B$), where $z \in B \cap B'$, the value $v_{\text{shap}}(z; U, B)$ can vary a lot in comparison to $v_{\text{shap}}(z; U, B')$. Generally, this means that when a new point $z' \notin B$ is added to B , the procedure must be repeated, and the data Shapley value needs to get

calculated for all points in $B \cup \{z'\}$ [2]. This dependency on a data set seems to be inefficient, especially when dealing with a large data set.

3.3. Distributional Framework for the Shapley Value

In the last chapter a brief overview over the current data valuation methods has been given. The Data Shapley and its approximations methods seem to look promising from a theoretical perspective. However, practical limitations exist, which have been discussed 3.1. and 3.2.

The latest work on this topic introduced a method, which is called the Distributional Framework for the Shapley value (DF), which addresses these limitations [2]. Moreover, the method seems to be promising to have an impact on the design of data marketplaces.

The Data Shapley and its approximation methods have the limitation that the value of every point depends on every other point in the data set. In context of privacy aspects, this means that the data Shapley has a significant drawback as the estimate of $v_{shap}(z; U; B)$ for a point $z \in B$ reveals information about the other points in B as well. The DF aims to reduce the dependency on the data set and therefore opens the door to value data without restricting privacy concerns. In Figure 6, the change of the first ingredient to a data distribution instead of a whole data set is shown.

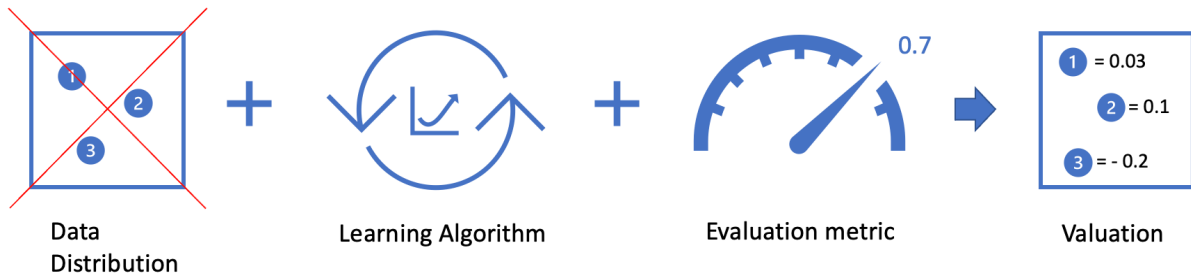


Figure 6: Ingredients for Data Valuation with the Distributional Framework

Instead of a fixed train data set, as it is shown in Figure 5, the DF uses a data distribution D . The DF, still, inherits many of the desirable properties of Shapley value, including the Shapley axioms and an expected efficiency property. The data distribution can be seen as a statistical element. The mathematical formula then looks like this:

$$v_{dshap}(z; U, D, m) = E_{\substack{k \sim [m] \\ S \sim D^{k-1}}} [U(S \cup \{z\}) - U(S)]$$

Equation 6

To understand the functionality, it is worth it to compare the DF with the Data Shapley, which has been introduced in Chapter 3.2. The Data Shapley depends on a specific draw of data from the distribution D and can therefore be seen as a random variable. In contrast, the DF is the expectation of this random

variable, which prevents instability caused by the variance of $v_{shap}(z; U; B)$ [2]. This connection between the Data Shapley and the DF is shown in Equation 7. It is claimed that this stability aspect is advantageous in comparison to the Data Shapley since it still fulfills the Shapley axioms. Secondly, the stability aspect brings significant advantages in practice. It has been shown that the DF satisfies the stability concept from Lipschitz in most of the cases [2]. Since it is possible to value data points outside the data set B as well, it is interesting to see if two points z and z' have similar Shapley values, when added to the data set. This is true when the utility function satisfies the Lipschitz stability concept.

$$v_{dshap}(z; U, D, m) \triangleq E_{B \sim D^{m-1}} [v_{shap}(z; U, B \cup \{z\})]$$

Equation 7

Furthermore, an efficient procedure for performing the DF, the D-Shapley, has been developed [2]. However, the runtime performance of the D-Shapley has not been evaluated yet on a real-world application. In context of performing data valuation on real world applications, they also hint that there is an upper bound on how big the data set should be to assure accurate Shapley values calculated by the DF. It would go beyond the scope of this thesis to show the proof of these features of the DF.

3.4. Data Marketplaces and Implications

It seems important to shortly describe the fundamentals of a data marketplace to understand the benefits of the DF in comparison to the other data valuation methods in this context.

Generally, a data marketplace is used to describe a place, where third-party data can be bought and sold. It is often thought of as a platform-based business that seeks to leverage scale or network effects.

A marketplace provides a clear ongoing benefit for everyone to take part instead of bypassing it and trading directly with other participants. Furthermore, a data marketplace should provide “thickness” so that buyers and sellers have the opportunity to trade with a wide range of potential partners [8]. In addition, especially when trading with health data, privacy policies must be obeyed.

Having introduced the key components of a data marketplace, it is now of great interest to discuss the impact of the data valuations methods, especially the DF, for the design of data marketplaces in the future.

In common data marketplaces, the data sets, which are put up for sale, get a certain value. However, as it has been described in Chapter 3.1, the value of the data also depends on the learning algorithm and the score of the performance metric (Figure 5). Taking this as an approach, it means, that for the same data set, its value might change when training a different task with it. In other words, this means that the learning effect of a ML model depends strongly on the task it will be trained on. Therefore, it seems to be a too short-sighted view to assign a general value for every data set.

Furthermore, when trading data, often privacy concerns occur. This problem is especially present in the field of healthcare. The DF has practical advantages in comparison to the other data valuation methods

when facing these privacy concerns. To understand its ability, a practical experiment setting will be thought through. Let's consider a hospital that desires to use machine learning in some way, for example for a binary classification task based on health features. However, as it has been described in Chapter 2, the hospital needs large amounts of data to train its machine learning model. As the hospital itself does not generate enough data to train the model successfully, the hospital must get more data from a different source that can be added to its own data. At this point, a data marketplace can be expedient to help the hospital to get a suitable data set for its model. The data marketplace makes it possible for buyers and sellers to get together to trade data fairly.

As it has been presented in Chapter 3, with the help of data valuation, every data instance in a data set gets a value assigned which represents its contribution to the performance of the model which has been trained on this data set. The classic data valuation methods, however, face a practical issue, which makes them not fitting as a tool for data valuation in some sort of a data marketplace scenario. To value the data, the buyer needs full access to the data set to evaluate its worth for the training of the model, since the train data is one crucial ingredient for data valuation. The data gets added to the data that the hospital already has and data valuation will be performed on the whole set.

In a typical trade act, it does not make sense to give the product to the buyer before selling it. There are multiple reasons not to give the data away before selling it. The DF overcomes these limitations, as the data set which gets evaluated only needs the underlying statistical distribution. The buyer does not need to get full access to the data set to evaluate its worth for the task. The mentioned privacy concerns can therefore be overcome when using the DF. In summary, it seems promising that data marketplaces can be important for data exchange henceforth and the DF can be a promising component for that.

4. Structure for Analyzing Data Valuation Methods

It is the goal of this thesis to examine the efficiency of the DF in performing data valuation on a medical data set. It has been described that for performing data valuation, three ingredients are necessary. The DF still needs three ingredients as well. However, it only needs a distribution D instead of the whole dataset B [2]. Thus, in 4.1., a learning algorithm and an evaluation metric get picked. In 4.2. a data set/ data distribution on which the learning algorithm gets trained on will be selected and pre-processing steps carried out. Lastly, in Chapter 4.3., the process of implementing the data valuation methods gets described before discussing the experiments and results in Chapter 5.

4.1. Model Architecture

There are two ML models, which gained interest in the field of healthcare recently. The first one is the Logistic Regression, which has already been discussed in Chapter 2.2. It is mostly used for classification tasks. A fitting data set is the “AIforCOVID” data set, further information is given in Chapter 4.2. It contains CXR images and 44 health features of people in Italy who have been tested positive for COVID19. The data set is labelled, in fact the label is either death (1) or no death (0). Only the health labels are of interest for this data set.

The second model architecture is the CNN, which is being used increasingly in radiology and medical imaging (Chapter 2.3.). In context of this thesis, it is intended to use an extensive data set of lung scans (labelled) and predict a specific disease. The data set is called “CheXpert” and provides 224,316 chest X-rays [17]. It has been used especially for studying the efficiency of the Data Shapley in valuating medical images [17]. The DenseNet-121 is a sub-model architecture of the CNN. It has been chosen as an appropriate model architecture, since it has turned out to be the best alternative for this task in comparison to other models [18].

Both model architectures have been implemented and trained on their specific sets of data, the Logistic Regression on the “AIforCOVID” data set and the Dense-Net121 on the “CheXpert” data set. Afterwards, the DF has been attempted to implement and to run on both data sets. Unfortunately, several errors occurred during the implementation process of the DF on the DenseNet-121. One error message stated that a ‘get_params’ method is not implemented. This is only one example as there have been a handful more errors. Since for this model architecture the method seemed to be not fully implemented yet, it was unrealistic to solve these problems in context of this thesis. However, the implementation of the Logistic Regression worked without any major issues. Hence, for this thesis, the Logistic Regression together with the “AIforCOVID” data set seems to be a good choice to evaluate further the efficiency of the DF as the method can be run flawlessly on it and it is often used in healthcare applications for diagnostic and prognostic predictions [13].

After two of the three needed ingredients for data valuation are present at this point, the “AIforCOVID” data set and the Logistic Regression as an ML model, a matching performance metric is the last ingredient to find. The Logistic Regression in this scenario is used for a binary classification task. Therefore, it seems logical to use an evaluation metric, which is suitable for this task. Possible candidates are the accuracy, the recall or the ROC curve and the AUC, introduced in Chapter 2.1. Unlike the threshold metrics, the AUC value reflects the overall ranking performance of a classifier. The AUC has proven to be theoretically and empirically better than the accuracy metric for evaluating the classifier’s performance and to find an optimal solution during the training [19].

As it has been described in Chapter 2.2., a Logistic Regression has several parameters, which must be determined first. Therefore, the decision boundary is set to 0.5 and the “Softmax” function is used as a cost function. Lastly, the Logistic Regression uses the “lbfgs” algorithm as an optimizer. “lbfgs” stands for “Limited-memory Broyden-Fletcher-Goldfarb-Shanno”. With these parameters set, the model is ready for the training phase. Before that, the data set is presented in more detail in the following chapter.

4.2. Dataset and Training

COVID-19, also known as the corona virus, has led to a pandemic since 2020. The pandemic had and still has a strong influence on the people’s everyday life. There have also been shocking results for the economy, as the gross domestic product decreased in Germany by 4,9 % in 2020. More important, the pandemic has cost many lives as there have been around 4,55 Mio. deaths worldwide so far. As this thesis focuses on analysing data valuation methods for machine learning applications in context of healthcare, the “AIforCOVID” data set seems to be appropriate for this task.

During the first wave of COVID-19 in Europe from March to June 2020, Italy was hit hard by the pandemic. Italy had to deal with many infections and deaths during that period. Especially the situation in the hospitals was severe as they were completely overcrowded. During that time a lot of data has been generated during clinical activity. The primary purpose was to manage the patient’s health development within the daily practice. Retrospectively, the data was reviewed and collected after anonymization.

As a result, a data set has been generated. The data has been collected from 6 hospitals in Italy and provides data from 820 patients. The data set includes CXR images, several clinical attributes and clinical outputs. Each instance of the data set has various labels (total: 44 labels). Some of the labels are binary (0 or 1) and the rest have numeric values (for instance age). Some have NaN (not a number) as a value, since not for every patient every feature has been observed. In total, 1103 data instances are available.

In Figure 7, the features have been listed. They consist of characteristic health values (for example age, obesity, Glucose value) as well as specific diseases. This includes cardiovascular disease, dementia, diabetes and so on. In addition, also therapy measures have been recorded (for example “Therapy_anti-

inflammatory” and “Therapy_antiviral”). Overall, 39 features are suitable for the ML task. The output label decides between death (1) and no death (0).

Hospital	Therapy_eparine	CardiovascularDisease
Age	WBC	IschemicHeartDisease
Sex	RBC	HeartFailure
Positivity at admission	Fibrinogen	Ictus
Temp_C	Glucose	HighBloodPressure
DaysFever	PCT	Diabetes
Cough	INR	Dementia
Difficulty in Breathing	D-dimer	BPCO
Therapy-anti-inflammatory	Ox_percentage	Cancer
Therapy_Tocilizumab	PaO2	Chronic Kidney Disease
Therapy_Anakinra	SaO2	Respiratory Failure
Therapy_hydroxychloroquine	PaCO2	Obesity
Therapy_antiviral	pH	Position

Table 2: Features of "AlforCOVID" Data Set

First, to be able to work with the data set, it has get uploaded to python, more precisely to the Jupyter notebook. The data set is available as a csv file. In the first step of pre-processing, the file gets converted to a “data frame”, which represents a table of data with rows and columns and belongs to the opensource library “pandas”. Because the focus lies on a Logistic Regression, the CXR images can be neglected. Instead, only the clinical attributes will be used as features. From the total 44 labels, only 39 labels seem to be suitable for the regression. A few patients occur more than ones in the data set. Therefore, to create a better learning process for the Logistic Regression model, the data set got adapted in a way that every patient only occurs once in the data set.

Another important question is how to deal with the NaN entries. For some patients, not every health feature has been documented. Therefore, occasionally not every patient has all the features. One possibility is to change all NaN values to “0”. Another possibility is to change them to “1”. It is also possible to delete all rows which own one or more NaN values. This does not seem to be efficient as the number of rows, which have a NaN value is quite high. Hence, all NaN entries have been converted to “0”. In the next step, all the values have been normalized. That means, that every entry now has a value between “0” and “1”. The binary values stay the same.

After the pre-processing steps the model gets trained on the test data set. Therefore, the “data frame” gets converted to a “NumPy” array. “NumPy” is another opensource library, especially useful for array

operations. This “NumPy” array with the specific features gets split into a training and test data set. For our data split, the hold-out method was applied. This means that a proportion is held back for the testing of the model. In this setting a 75:25 split is used, which means 75 % of the data is used for training and 25 % is used for testing the model.

During the training phase, the Logistic Regression gets trained on the training set. Afterwards, the model is able to predict values from the test set. Based on the results, the confusion matrix has been plotted as a graph.

	P' (predicted)	N' (predicted)
P (actual)	226	4
N (actual)	32	14

Table 3: Confusion Matrix for a Logistic Regression

In Table 3 the confusion matrix of the Logistic Regression for the “AIforCOVID” data set is shown. The True positive rate with 226 values is relatively high in comparison to the other values. It is noteworthy, that the proportion of False Positive values is quite high, which is not a positive feature for the model’s performance. Still, based on Table 3, the performance of the Logistic Regression convinces overall.

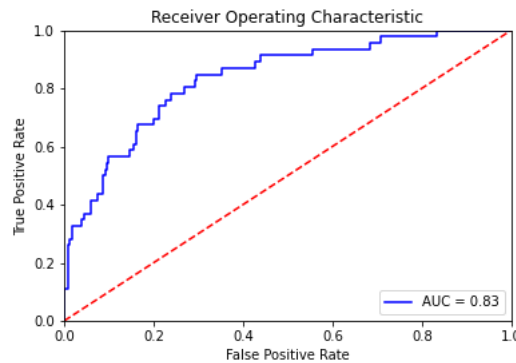


Figure 7: ROC for Logistic Regression

In Figure 8 the ROC curve and the AUC score is presented. With an AUC score of 0.83 (the maximum is 1.0), the model is successful in correctly predicting the right label, death or no death.

4.3. Implementation of the Data Valuation Methods

All three ingredients for data valuation, which have been discussed in Chapter 3, are now present. A data set/ distribution, a learning algorithm and a performance metric are available at this point. The implementation steps are based on the source code from Amirata Ghorbani which has been published in context of the scientific paper which introduces the DF [2].

Because it is desired to study the empirical effectiveness of the DF, it seems useful to compare it with a different data valuation method to assess the performance. In Chapter 3, various methods based on the Shapley value have been presented. It has been shown that the TMC-method is superior in comparison to many other methods [5].

However, when comparing it to the KNN-Shapley, a different data valuation approximation method, the TMC-Shapley performs worse, especially on large data sets [6]. It still seems a reasonable choice to compare the DF to the TMC-Shapley instead of the KNN-Shapley, since the data set is not very extensive with 1000 data points and the experiments that the source code provides were not carried out with the KNN-Shapley method. Therefore, the TMC-method is used in this thesis as a comparison method.

The following assumptions are relevant for the applications and experiments explained in Chapter 5. First, both methods get implemented on a set of 320 train data points and 100 test data points. Although the “AIforCOVID” data set contains about 1000 data instances, a train set of 320 data points seemed to be reasonable to investigate the model performance since its runtime is fairly high (more about this in Chapter 5.2). The data valuation process is designed to calculate the values iteratively. The number of samples to take at every iteration is set to 10. In addition, a stopping criterion is set that determines a deviation limit for the change of the values in the past 100 iterations. In this thesis, the stopping criterion is set to 0.01, which is the default value. The calculated values are stored in an array. The data points get assigned a value between -1 and 1. In general, the values are positive when there is a positive contribution towards the model’s performance and negative when the data point reduces the model’s performance.

5. Applications and Experiments

In the following chapter, the structure, and the results of two experiments are described. In 5.1., the Shapley values have been plotted in a histogram in comparison to another data valuation method (TMC-Shapley). Afterwards, a runtime comparison is performed, which is the first experiment. Lastly, a point removal experiment is carried out to understand how successful the DF is in valuating data so that the research question can be answered adequately.

5.1. Distribution of the Estimations

In Figure 8 the histogram on the left represents the Shapley values calculated by the DF and on the right side the values have been calculated by the TMC-Shapley. The train size for both evaluations is 320. The remaining parameters are the same compared to the implementation parameters described in Chapter 4.3. It is the goal to understand if the DF calculated the values of the data points correctly. A negative value implies that the data instance has a negative influence on the model's performance. Often, low values capture outliers and corruptions. High values, in comparison, signify a positive impact on the model's performance. The first observation is, that the results of both methods are quite close to each other, when examining the two graphs in Figure 8. The minimum value of the methods is -0.0224 (DF) and -0.0226 (TMC-Shapley) which are very similar. The highest values are 0.0297 and 0.0235 respectively. Furthermore, the rank correlation between the DF and TMC-Shapley values is 0.933. This shows that the two methods correlate strongly. The absolute percentage error is -11.4 %. In total, this strengthens the first observation.

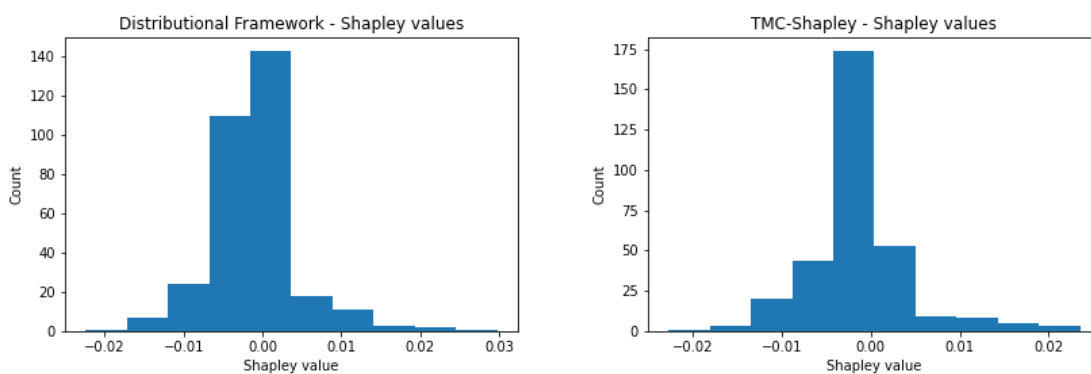


Figure 8: Histograms of the Shapley Values

The second observation is, that the distribution of the values of both methods are not distributed evenly. When looking at the values calculated by the DF, only 11% percent of the values are positive, and 89% percent of the values are negative. The results of the TMC-Shapley are close to each other as 24% percent of the values are positive and 76% negative. The sum of the bins of the DF is 0.04 and of the

TMC-Shapley is 0.004. Hence, the TMC values the training points from the “AIforCOVID” dataset slightly more positively.

5.2. Runtime Comparison

The first experiment is a runtime experiment to see how the duration of the DF develops in comparison to the TMC-Shapley method. Therefore, the two methods have been run on different training sizes. The training size have been as follows: 20, 40, 80, 160, 320 and 450. There is a time limitation for the jupyter notebook (4 hours). Therefore, 450 was set to be the maximum number for training points since greater sets would not finish within 4 hours. The train and test data points were randomly picked from the training and test set, respectively.

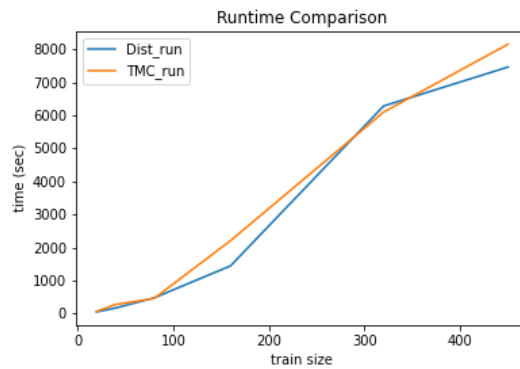


Figure 9: Runtime Comparison of the Distributional Framework and the TMC-Shapley

Figure 9 plots the train size on the x-axis and the time in seconds on the y-axis. It shows that the time needed for running the DF and the TMC-Shapley is similar. It is no clear tendency recognizable. It could be determined that the DF needs about two hours for valuating 450 values. This seems to be a long runtime for the size of the data set, since in practice the size of data sets for training quickly exceeds a hundred thousand.

5.3. Point Removal Experiment

The point removal experiment has been used by Ghorbani and Zou [5]. It is the goal to show how good a method is in evaluating data, especially in comparison to a different method. The experiment works as follows:

First, after the model has been trained and the performance metric calculated, the data valuation is performed on the whole train data set. As a result, every training point gets a value assigned. Then, the data points get ranked according to its values. After that, a specific number of points get removed from the training data set iteratively. Then, the impact of the removal gets evaluated on the performance of the

test data set. Lastly, the change in performance is measured by comparing the evaluation metrics (before and after the removal of points).

There are two common ways in performing data removal. Either high values or low values can be removed. It is expected that removing high values would lead to a decrease in performance of the model. Vice versa, when removing data points with low values the performance of the model should increase. In this experiment setting, low values have been removed from the training set. Three ways of removing data points have been tried. The first two ways have been done by performing data valuation and point removal with the DF and the TMC-Shapley method. In addition, also randomly picked data points have been removed to see how this would impact the model's performance. Both data valuation methods have been trained on 320 training data points. The test set size is 100.

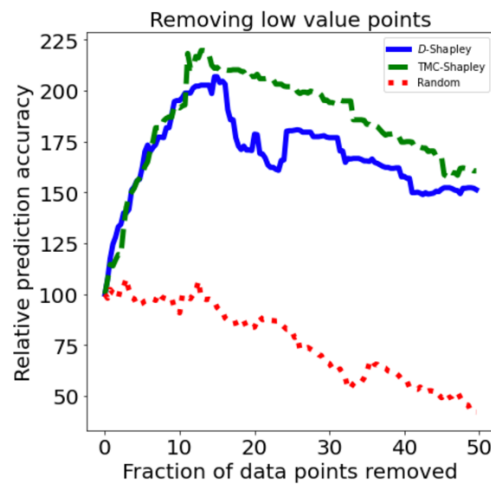


Figure 10: Point Removal Experiment - Results

Figure 11 plots the results from the point removal experiment, which has been explained above. On the x-axis the fraction of data points, which are removed from the data set, is shown. On the y-axis, the relative prediction accuracy is presented.

The TMC-Shapley and the DF respond similar to the removal of “bad” data points from the training set. The prediction accuracy clearly goes up at first. At a certain point the accuracy drops again. This seems consequential, since at a certain point, the “bad” data points have all been removed and the algorithm begins to remove data points which have a positive value and therefore are positive for the model's performance.

The removal of randomly picked data points does not have a significant influence on the performance of the model at first. This seems logical as the algorithm should pick positive and negative values in a way that in the mean a value around zero should be reached. However, as the algorithm increases to remove points, the same phenomenon as with the other data valuation methods can be observed.

The removal of low value points, in conclusion, leads to a significant improvement in performance when using the DF. The impact of removing randomly picked data points is, in contrast, significantly worse.

6. Discussion of Experiments

6.1. Principal Findings

The obtained results should be studied in context of the goal of this thesis: To evaluate the empirical effectiveness of the DF.

It is assumed that the TMC-Shapley values the data points correctly, since it has been used in the example and its empirical correctness has been proven in research. When comparing the results with the values estimated by the TMC-Shapley method, it can be concluded that the DF calculated the values correctly, too.

The runtime comparison was performed second. In terms of efficiency, the DF was not able to outperform the TMC-Shapley regarding runtime. Even though the TMC-Shapley is successful in valuating data, it has practical disadvantages since the model must get retrained continuously during the valuation process. The DF, hence, also faces the limitation of inefficiency in time consumption. Different approximation methods perform significantly faster (for example the KNN-Shapley) and seem superior in this context.

The point removal experiment then showed that the DF performs similar in discriminating between good and bad values on the training data set. Generally, the graph showed that both, the DF and the TMC-Shapley, are competent in identifying low value data points in the train data set. Removing bad data points from the data led to a significant increase in performance. This effect can clearly be seen in the graph (Figure 10). To validate the performance of the two methods, points were also removed which have been picked randomly. The results were clearly worse, which implies that the DF is capable of successfully valuating data.

The following finding is not directly related to the experiments; however, it seems noteworthy to discuss the finding as it seems important for future work. In Chapter 4.1. two different model architectures, a Logistic Regression and a CNN have been evaluated for the data valuation task. The CNN, as discussed in Chapter 2.3., is especially suitable for image classification tasks. Hence, the CNN has been trained on the “CheXpert” data set (Chapter 4.1.) and the DF has been tried to apply on this setting. In comparison to the Logistic Regression, the DF failed to evaluate the train data set. Multiple errors occurred and it seemed that not every method was fully implemented for this problem. Hence, it can be cautiously assumed that the DF is not fully applicable for every model architecture yet and further implementation might be necessary.

6.2. Implications for Research and Practice

The DF encourages to make a further step towards a more practicable and more independent way to perform data valuation. The DF was successfully applied on a real medical classification data set. The

DF showed that it is successful in classifying the data into high and low value data points. The method is another approach to perform data valuation on medical data sets.

Since the DF has brought the exciting feature that data valuation can be performed without the dependency on a fixed data set, it strongly stimulates further research to solve the problem of insufficient data. More precisely, the DF looks promising to transform the way data can be traded in the future. It can boost the research and development of data marketplaces, since common data privacy concerns, which led to several limitations, are surmount.

The already known data valuation methods showed that they can improve the quality of data sets by removing low value data. Since, the DF showed in the experiments that it performs as good as the TMC-Shapley regarding quality, it seems convincing that the DF is equally able to do so.

6.3. Limitations and Future Research

Following the implications on research and practice, this method has brought promising findings for theoretical and practical developments in this research area.

The dependency of data valuation methods on a fixed dataset has been reduced by the development of the DF. However, the method is still dependent on the model architecture. The effort of implementing the DF on different model architectures can differ quite significantly depending on the model. This thesis showed that applying the DF on a medical image classification task, when a CNN is used, can produce several obstacles. This brings the following consequence for research: Future work should seek for ways to reduce the dependency of the DF or a similar method on the model's architecture.

In our setting, the DF was not able to perform better regarding time consumption than the TMC-Shapley, which has been proven to be time relatively inefficient when it comes to extensive data sets. Hence, this knowledge may stimulate research, to increase the DF's applicability concerning its time performance on greater data sets.

7. Conclusion

As data has become the fuel driving technological and economic growth, the valuation of data in algorithmic predictions and decisions has become a fundamental challenge. Several data valuation methods have been developed. In this work I have investigated how the recently released method of the Distributional Framework for the Shapley Value (DF) performs on a real-world medical data set. By comparing the calculated SVs to the TMC-Shapley in performed experiments as runtime and point removal experiments, it could be demonstrated that the DF is an efficient instrument for data valuation with advantageous properties in comparison to common data valuation methods.

It is very likely that data valuation in general will gain more interest due to the developments of methods like the DF and the growing importance of data in general. The DF has great potential for a variety of applications, from improving data sets to the implementation of data marketplaces where data can be traded fairly.

However, there is still the need to investigate the performance of the DF even further. For example, it could be useful to see how well the method performs when implementing it on a convolutional neural network since this also has been tried during this work, but errors occurred. Furthermore, research may focus on evaluating the possibilities of improving the model's efficiency in context of runtime for large data sets.


Nevertheless, as this thesis seems to be one of the first studies to deal with this new method, applying the DF on a real-world medical data set has stimulated future work and the gained results have motivated future work, both for researchers and practitioners.

References

1. Peters, H., *Shapley Value*, in *Game Theory – A multi-leveled Approach*, 2008. P.125-126
2. Ghorbani A., Kim M.P. & Zou J., *A Distributional Framework for Data Valuation*. 2020.
3. Shapley, L., *A Value for N-Person Games*, in *Contributions to the Theory of Games*. 1953. P. 307-317
4. Roth, A., *The Shapley Value*, in *Essays in Honor of Lloyd S. Shapley*. 1988: Cambridge University Press.
5. Ghorbani, A. and Zou, J., *Data Shapley: Equitable Valuation of Data for Machine Learning*. *Proceedings of the 36th International Conference on Machine Learning, 2019*. Long Beach, California, USA.
6. Ruoxi, J., et al., *Scalable vs. Utility: Do we have to sacrifice one for the other in Data Importance Quantification?*
7. Ruoxi, J., et al., *Towards Efficient Data Valuation Based on the Shapley Value*. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*. 2019. Naha, Japan.
8. Koutroumpis, P., *The unfulfilled Potential of Data Marketplaces*. 2017. ETLA Working Papers No 53. <http://pub.etla.fi/ETLA-Working-Papers-53.pdf>
9. Jo, T., *Simple Machine Learning Algorithms*, in *Machine Learning Foundations*. 2021. P. 69-90
10. Jurafsky, D., Martin, J., *Logistic Regression in Speech and Language Processing*. 2021
11. Hossin, M., Sulaiman, M.N., *A Review on Evaluation Metrics for Data Classification Evaluations in International Journal of Data Mining & Knowledge Management Process*. 2015
12. Kleinbaum, T., Klein, M., *Introduction to Logistic Regression in Logistic Regression – A Self-Learning Text*. P. 2-37
13. Zamare, R., et al., *Implementation of Logistic Regression in Healthcare in International Research Journal of Education and Technology*. 2020.
14. Ertel, W., *Neuronale Netze in Grundkurs Künstliche Intelligenz*. P. 285 – 333.
15. Kim, P., *MATLAB Deep Learning*. Seoul, Korea. P. 121-147.
16. Jia, R., et al., *An empirical and Comparative Analysis of Data Valuation with Scalable Algorithms*. arXiv preprint: arXiv:1911.07128v1, 2019.
17. Irvin, J., et al., *CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison in The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*. 2019.
18. Tang, S., et al., *Data Valuation for Medical Imaging Using Shapley Value: Application on A Large-scale Chest X-ray Dataset*. 2021.
19. Ling, C., et al., *AUC: A Better measure than Accuracy in Comparing Learning Algorithms in Conference of the Canadian Society for Computational Studies of Intelligence*. 2003. P. 329-341.

Declaration about the Thesis

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.



Karlsruhe, den 26. October 2021

VORNAME NACHNAME