Master thesis Artificial Intelligence

Unboxing the Black Box using Case-Based Argumentation

Rosa Ratsma
5518350

External supervisor: Dr. Sjors Broersen
First supervisor: Prof. dr. mr. Henry Prakken
Second examiner: Prof. dr. Floris Bex

Utrecht University
Department of Information and Computing Sciences

July 13, 2020

Deloitte.

Acknowledgments

First of all, I would like to thank Henry for his supervision during the project. He trusted me in bringing his own research proposal into practice and gave me a lot of freedom and responsibility along the way. Any time I would criticize his proposal or come up with new ideas, he would welcome it with enthusiasm, responding 'write about it!'. This gave me the confidence to develop the project in my way. I would also like to thank second examiner Floris for his instructional feedback at the beginning of the project.

I am grateful to A&C Deloitte for giving me the opportunity to conduct my research within their department. Although the circumstances did not allow me to finish my internship at their place physically, they were still always there to help me out. A special thanks to Edwin for making sure I had a good time at Deloitte and to Sjors for the supervision during the project. With his vision on translating ideas into actual applications, Sjors was a great support along the way. Finally, I would like to thank the GlassBox team, in particular Sebastiaan and Bojidar, for the interest in the project and thinking along.

In my personal circle, I would like to thank all that took the time to participate in the user experiment. I really appreciated the willingness to help, and your participation and comments were of great value to the study. Lastly, I would like to thank Rens, my family and friends for being there for me in other ways. A special thanks goes to Rens for the support and company during the a-typical past months. Without our daily walks and talks, the project would not have been the same.

Abstract

In search of the most accurate and stable predictors, machine-learning algorithms have been introduced that are so difficult to interpret that we metaphorically call them ‘black boxes’. Their lack of interpretability hinders their applicability in relevant domains, where it is often desired or even required to explain decisions. Recent work proposes case-based argumentation as a tool for justifying the predictions of black-box models. Case-based argumentation is a form of reasoning that draws analogies between new and previous cases. It fits naturally with machine learning, as input data can directly be used as cases. In this study, we bring the proposed justification system into practice. Based on the evaluation, we suggest a new argumentation framework. Besides justification, we examine the possibilities for replacing or monitoring black-box prediction models using case-based argumentation systems. The results of a user experiment hint at the suitability of a monitor approach.

Contents

1. Introduction

In recent years, machine learning has become one of the main promises of Artificial Intelligence. Although the field is by no means new, greater availability of data and improved computing power allowed for impressive recent results.

Machine learning works fundamentally different than rule-based systems. Instead of prescribing rules on how to *act*, programmers prescribe machine learning algorithms rules on how to *learn*, and provide them with learning experiences. Although we are technically still in control - we decide on how and from what the system learns - the resulting systems can learn such complex relations that we lose track of their inner workings. Algorithms such as Deep Neural Networks and Support Vector Machines have reached a level of complexity that makes them so difficult to interpret that we metaphorically call them 'black boxes'.

At the same time, machine learning applications are waiting to be applied in relevant domains such as healthcare and criminal justice, where they can make high-stake decisions in increasingly autonomous roles. In many of these domains, it can be desired or even required to explain the output of an algorithm. As a result, research towards the automated generation of explanations of machine-learning algorithms has recently attracted a lot of interest (Adadi and Berrada 2018; Guidotti et al. 2019; Robeer 2018).

1.1 Approach

In this research, we will investigate to which extent machine-learning models can be 'unboxed' using case-based argumentation. Case-based argumentation is

a form of reasoning that draws analogies between new and previous cases. This is a form of reasoning humans can easily relate to (Cunningham, Doyle, and Loughrey 2003; Gentner, Loewenstein, and Thompson 2003). Moreover, it fits naturally with machine learning, as input data can directly be used as cases.

As our starting point, we will implement and evaluate the proposal of Prakken (2020) to use case-based argumentation for explaining the predictions of black-box models. Based on this analysis, we will try to come up with suggestions for improvement. These suggestions will be implemented and tested using both computer- and user experimentation.

The research will be carried out within the Artificial Intelligence group at the Analytics and Cognitive (A&C) service of Deloitte in Amsterdam. At A&C, they help customers with modernizing their data and analytics environments and applying artificial intelligence. As complex machine-learning models have become one of their main tools to work with, they experience the growing demand for interpretability up close.

1.2 Structure

Before diving into the method investigated in this research, we will cover a brief overview of the field of explainable machine learning in Chapter 2, followed by a description of case-based reasoning in Chapter 3. In Chapter 4, we will zoom-in to the field of case-based argumentation and discuss existing work related to machine learning. At the end of this chapter, we will formulate our research questions.

The second part of the thesis starts with the evaluation of the proposal of Prakken (2020) in Chapter 5. In Chapter 6, a new argumentation framework is proposed. In addition, three directions for applying case-based argumentation are suggested: justifying, classifying and monitoring black boxes. Chapter 7 further investigates the classification possibilities. The justification- and monitor approach are tested with a user experiment, which is the topic of Chapter 8. We conclude with answering the research questions and making suggestions for future work in Chapter 9.

2. Explainable machine learning

2.1 What is machine learning?

Machine learning is a set of methods used by computers to make and improve predictions or behaviours based on data (Molnar 2019). It works fundamentally different than rule-based methods, in which all the instructions are given explicitly to the computer. In his book *The Discipline of Machine Learning* Mitchell (2006) provides a short formalism of what it means for a machine to learn:

> *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."*

The flexibility of machine learning allows us to use the defined experience and performance measure to train all kinds of algorithms on the task and stick with the one that performs best.

In this work, we will focus on supervised models. These models have the task to learn some function $f : X \text{ -> } Y$, which maps input (X) to labels (Y). The experience of the model is *supervised*, meaning that the data the model learns from - the *training data* - consists of labelled examples $\{x, y\}$ in which x is the input and y the realized outcome. More specifically, we will focus on *binary* prediction models. These models have only two possible prediction outcomes and are also called *classifiers*.

As our running example, we will consider a *Churn* prediction model for a Telecommunication company. The task of this model is to predict whether customers of

the company intend to continue their membership, or churn: cancel it. These predictions can help the service provider adapt its strategy to prevent customers from leaving. We will specify the Churn model in terms of the definition of Mitchell:

Task: The task of the model is to learn a function f : $customer_i$ -> $decision_i$, which can, given some customer, make a binary prediction about its decisions to stay or churn.

Experience: The experience of the model consists of pairs of data $\{x_i, y_i\}$, including information about a previous customer (x_i), and the decision - stay or churn - that customer made (y_i).

Performance measure: The performance of the model is measured by testing the model on new customers - the *test data* - and calculating the prediction accuracy: the number of correct predictions divided by the total number of predictions.

2.2 Why is there a need to explain?

A downside to machine-learning methods is that the insights gained about the data and the task they solve are usually hidden in complex models. Understanding how a model came to a decision is often an impossible task, even for the developers of the model. Models that are this opaque or that are proprietary are metaphorically called 'black boxes'. However, if we know that a machine-learning model performs well, can we not just ignore the 'why-question'?

There are algorithms for which we can. Suppose we have a model that recognizes handwriting on postcards with about perfect accuracy. The only information we need from the model is the postcode itself. Given the relatively small consequences of a mistake, we can trust it for this task based on its accuracy. Like in the example, explanations can be unnecessary when either the consequences of wrong results are insignificant or when we trust the decision process and do not need to know more than the outcome (Molnar 2019).

For other problems or tasks, the prediction alone may not be sufficient to apply a model or make optimal use of it. As Doshi-Velez and Kim (2017) describe, the need for interpretability arises from incompleteness in the problem formalization.

By that, they mean that to solve the problem, it is not sufficient to just obtain the prediction, the system must also be able to provide information about the context of the prediction.

Consider the classifier for the Churn prediction task. When this model is presented with a new customer, it would output either 'churn' or 'stay'. This answer, although relevant, is unsatisfactory. It demands full trust in the workings of the model, as it offers no insight into the grounds on which the decision is based. Moreover, it does not provide any starting points for improving the situation in case the model predicts the customer to churn. The prediction alone is an incomplete solution to the problem. Below five important motivators that drive the demand for interpretability will be discussed.

2.2.1 Building trust

An essential ingredient for the adaption of artificial intelligence by companies, individuals and society as a whole is trust. As long as users do not find the explanations of an AI system acceptable, they will either not use the system or refuse to comply with the decisions made (Guidotti et al. 2019).

People tend to attribute human traits to objects (Heider and Simmel 1944; Molnar 2019). Just like humans can receive our acceptance when we come to understand their reasoning or intentions, a machine-learning application will find more acceptance when it can explain itself.

2.2.2 Managing social interactions

Somewhat related to the goal of building trust, explanations can be used to manage social interactions. Both the impression of the *explainee* - the receiver of an explanation - of the performed behaviour and any future actions the explainer and explainee might perform together, could be managed through creating shared meanings of certain concepts (Malle 2006). Establishing shared meanings can be essential to enable a machine to interact with us or to achieve their intended goal (Molnar 2019).

2.2.3 Learning from the model

A model can contain valuable information that is not transferable through its outputs (Z. C. Lipton 2018). Explanations could help to transfer such knowledge from the model to humans. For example, when a black-box model is used in research, potential new scientific findings would remain hidden when all the information we receive from the model is a prediction. Using explanations, we could increase our scientific understanding and potentially detect causal relationships (Robeer 2018).

A related motivator for interpretable systems is the desire of humans to 'find meaning in the world'. We want to harmonize any contradictions or inconsistencies between elements of our knowledge structure that might appear when a machine-learning algorithm comes up with an unexpected prediction (Molnar 2019).

2.2.4 Debugging and safety measures

A machine-learning model is optimized to a specific objective, while there may be other criteria the model has to adhere to, such as objectivity. Molnar (2019) describes an example of a model trained for automatic approval or rejection of credit applications. Not to discriminate in this decision based on specific demographics is a constraint that may be part of the problem formalization. However, the loss function for which the machine-learning model was optimized does not cover it. Any biases that exist in the training data will be picked up by the model. Increased interpretability is crucial to detect those biases.

Explanations can also help provide insight into the risk profile of a model (Molnar 2019). Information about which features are most important for a given prediction can help detect cases in which the machine-learning model might fail. More directly, an explanation of a wrong prediction can help us understand what caused the error.

2.2.5 Ethics & regulation

Machine-learning algorithms are increasingly being used in applications where their decisions can have a severe impact on human lives. Whether someone is

allowed to receive a credit card or buy a house may be decided based on a black-box system, answering 'yes' or 'no'.

The concept of influencing human lives with opaque decision systems has received much criticism. The lack of transparency raises ethical concerns regarding the grounds on which the system bases its decisions. In many data sets, correlations which we do not wish to be used by the prediction model exist (Edwards and Veale 2017). For example, correlations related to 'protected characteristics', such as race or gender, are often unwanted and may not be allowed to be used in a decision-making process directly (Edwards and Veale 2017).

The General Data Protection Regulation (GDPR) includes clauses in order to protect people against the consequences of unfair automated decision-making. It allocates citizens, to some extent, the right to an explanation of the logic involved in an automated decision that impacts them (Guidotti et al. 2019). While this directive is not exceptionless, the law may reflect the future direction of legislation in this area (Rathi 2019).

2.3 Structure and key terminology

Different methods for generating explanations to machine-learning models exist. Figure 2.1 shows a structure in which we can classify these different methods. The alternative to explaining a black-box model is to replace the model by one that is interpretable by nature - a *white box*. An example of such a model is a linear regression model, which prediction is a weighted sum of the feature inputs (Molnar 2019).

The other option is to continue using a black-box model and try to explain it. This approach can also be divided into two directions. *Model-specific* explanation models are built for a specific model class and usually assume to have access to the inner working of the AI model. *Model-agnostic* explanation models generate an explanation without such access and can, therefore, be applied to different kinds of models. These methods have, by definition, no information about the decision process inside the model and usually base their explanations on analyzing input-output pairs (Molnar 2019).

The main aim of all of the methods is to increase the *interpretability* of the model. Interpretability can be defined as the degree to which a human can under-

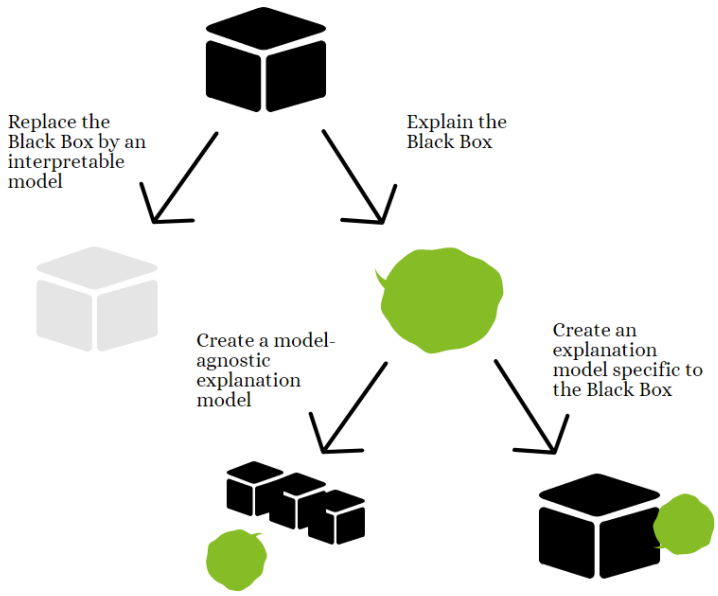

Figure 2.1: *Overview of the methods to create interpretable Machine Learning*

stand the reasons behind a decision (Miller 2019). The research field concerned with making machine-learning systems interpretable is sometimes called *Explainable AI*. An explanation can be used to pursue interpretability. Miller (2019) defines an *explanation* as the answer to a why-question.

An answer to the question of why a model in general works the way it does can be provided by a *global explanation*. Global explanations are generated to provide insight into the workings of a model as a whole. *Local explanations*, on the other hand, are used to explain a particular decision of the algorithm. They form an answer to the type of question 'Why did the model predict x in case y?'.

In this work, we will adopt a broad definition of *explanation*. We will consider any form of supplement to the prediction of a black box that can prove insightful to a user an explanation.

2.4 Design choices

As explained above, multiple approaches to pursuing interpretable AI are possible and have been tried. This section will elaborate on the choice between the different approaches.

2.4.1 Replacing or explaining

A fair question before starting to explain black boxes is whether those opaque models are necessary in the first place. Whereas some researchers seem to assume an insurmountable loss in accuracy when replacing a complex model with one that is interpretable, others argue this to be a misconception. One of the protagonists of interpretable models is Rudin (2019). She argues that for problems that have structured data with meaningful features, there is often no significant difference between the performance of black-box classifiers and much simpler models. Besides, according to Rudin, there are some good reasons to refrain from explaining AI.

To begin with, "explanations must be wrong" (Rudin 2019). Would the explanation be completely faithful to the computations of the original model, then the explanation model would be an interpretable model with which we can replace the original. This insurmountable fallibility can make it hard to trust an explanation, as there could be something wrong with the prediction, the explanation or both. Besides, currently used explanations of black boxes often do not make sense or lack the amount of information needed to understand the model truly. Furthermore, black-box models with explanations can have such complicated decision pathways that they are ripe for human error.

Using a model-agnostic approach may be forced in situations in which there is no access to the model, for example, when the model is proprietary. There can also be other motivations to refrain from opening the black box itself. Ribeiro, Singh, and Guestrin (2016) make a case for model-agnostic interpretability, as opposed to using interpretable models. Separating interpretability from the model creates the freedom to make the model as flexible as necessary for the tasks, enabling the use of sophisticated machine-learning approaches. Another advantage of a model-agnostic approach is that one can easily switch between models, which is not an uncommon operation in machine-learning pipelines

(Ribeiro, Singh, and Guestrin 2016). Furthermore, a model-agnostic approach enables the same techniques and representations to be used to explain different models; using the same explanation system makes it easy to compare them.

Interesting recent research investigated the trade-off between predictive accuracy and interpretability using Automated Machine Learning (Freitas 2019). Automated-Machine-Learning methods try to find the best fitting classification algorithm and the best configuration for a given data set (Yao et al. 2018). As Freitas (2019) describes, there is currently a strong focus on predictive accuracy in this area, which is often used as the only criterion to evaluate a classification model. He chose a different approach and compared black, grey (partly interpretable) and white boxes. With a 20-hour runtime limit, a white box model appeared to be most accurate for 7 of the 16 data sets. Furthermore, if we would accept a loss of 1% accuracy to use a white box, the white box would be a good fit for 10 of the 16 data sets. This finding supports the view that investigating the possibility to create an interpretable model, ideally by including such options in Automated-Machine-Learning methods, is worth trying and may for some applications even become a mandatory step to take.

Whether, in which cases, and to which extent a trade-off exists between accuracy and interpretability needs to be further investigated. The main criticism of Rudin seems to concern the practice of developers to simply assume that interpretability means a loss of accuracy. As supported by the research of Freitas (2019), this seems a premature assumption to make. On the other hand, black-box models offer advantages in flexibility and reach on specific tasks performances of which it is unclear whether an interpretable model will ever reach them. Therefore, it does not seem desirable to ban such methods, neither to stop finding solutions to create more insight into black-box models. This work will consider the suitability of CBA for both approaches. We will investigate the possibility of explaining black-box models, but also consider the potential of CBA to function as an independent white-box classifier.

2.4.2 Model-specific or model-agnostic

To keep the advantages of flexibility that come with a black-box, an explanation model should be model-agnostic. As model-agnostic methods do not look at the inner workings of a model, they are highly portable, meaning they can be applied to all kinds of AI models (Molnar 2019). The ignorance of the inner workings

also has the advantageous property that generating the explanation becomes independent of the complexity of the underlying system. Moreover, possibly sensitive information about the inner workings of a model does not need to be revealed.

A model-agnostic approach has zero-translucency, meaning that it does not rely at all on looking into the machine-learning model (Molnar 2019). It is, therefore, by definition, impossible to know the actual decision process of the model it tries to explain. Concretely, we will only assume to have access to:

1. The data the model is trained upon
2. The predictions of the model given input data

Further research must reveal whether that is enough information to provide high-quality explanations, and this work will try to contribute to that process.

2.4.3 Local or global

This research will focus on generating local explanations. A practical reason for this decision is that a global explanation of the workings of a black-box model can be too complicated, whereas 'zooming in' on the level of individual predictions can make the explanation task feasible (Ribeiro, Singh, and Guestrin 2018).

Furthermore, considering the motivational purposes for explaining AI, local explanations seem the most relevant. Research has shown that trust in AI applications is lost when users cannot understand traces of observed behaviour or decisions (Miller 2019). In such cases, a general description of the behaviour of the model does not seem suitable to either restore this trust or detect a mistake, whereas a local explanation would be helpful. Another example is the research of Wachter, Mittelstadt, and Russell (2017), which concluded that local, counterfactual explanations aptly suit the 'right to explanation' of the GDPR.

2.5 A good explanation

An essential question for our purposes is 'What constitutes a good explanation?'. Fortunately, this is not a new question; researchers in the fields of philosophy,

psychology and cognitive science have worked on the topic of explanation in the last decades. Miller (2019) provided an extensive review of papers from those fields to help the field of Explainable AI build on this existing research.

Explanations are meant for users; human beings with finite mental capacity and, most likely, limited time. Therefore, adjusting the explanation to human capabilities and preferences is desirable. Miller (2019) distinguishes four significant findings of the way humans explain that he argues should be taken into account in Explainable-AI models:

2.5.1 Explanations are contrastive

People do not explain the reasons for an event on their own; they explain them relative to some other event that did not occur (Miller 2019). In other words, when someone asks the question 'Why P?', what is usually meant by that person is 'Why P rather than Q?'. P. Lipton (1990) refers to P as the *fact*. Q represents a counterfactual contrast case that did not occur and is called the *foil* (P. Lipton 1990).

When one asks 'why will this customer leave our company?', the person is probably not interested in the complete causal chain that caused the customer to churn. Instead, the question he or she is presumably interested in is 'why will this customer leave our company instead of stay?', or 'why will this customer leave our company rather than that (similar) customer?'. An answer that distinguishes between cases is called a *contrastive explanation*. A *counterfactual* is the change in the input data necessary for the decision to change from fact to foil (Robeer 2018).

The finding that a human-friendly explanation should be contrastive forms both a challenge and an opportunity for the field of Explainable AI (Miller 2019). A challenge lies in the fact that the foil is usually unknown and needs to be determined. On the positive side, generating a contrastive explanation can be easier and less computationally extensive than providing a full causal attribution (P. Lipton 1990).

The suitability of contrastive explanations for users has stimulated some researchers to explore methods that try to adhere to this criterion. Rathi (2019) designed a model-agnostic method to create contrastive and counterfactual explanations by using SHAP. A limitation of their approach is that they process a

contrastive question, 'Why P instead of Q?', by splitting it into the two segments 'Why P?' and 'Why not Q?'. By doing this, part of the meaning of the original question is lost, as the answer no longer focuses on the relevant difference that made the fact and the foil result in a different outcome.

An interesting work of Robeer (2018) does address the foil relative to the fact. Using 'Foil Trees', a set of rules is detected that caused the prediction to be the actual outcome, instead of the foil. The research provides supporting evidence that contrastive explanations align better with the decision process of the user and are perceived as more understandable (Robeer 2018).

2.5.2 Explanations are selected

Humans are rarely looking for an explanation that consist of an actual and complete cause of an event. Presenting that much information can cause the less relevant parts of the chain to dilute those parts that are crucial to the particular question asked (Tetlock and Boettger 1989). Instead, humans prefer short explanations concerning one or two causes (Miller 2019).

A challenge here is to create an explanation that is both human-friendly and truthful. The world is often more complicated than humans can comprehend, so it seems a better approach to make them understand the general intuition than to dilute them with all the available information. In order to stay truthful, it seems essential to be transparent about what the explanation system presents to the user.

2.5.3 Explanations are social

An explanation can be seen as part of a conversation or interaction between the explainer and the explainee. The social context and previous knowledge of the explainee influence what the appropriate content of the explanation looks like. An explanation for the programmer of a model will look different than an explanation for the end user.

2.5.4 Probabilities are not as important as causal links

It is more effective to refer to causes, than to probabilities or statistical relationships. (Miller 2019). The conclusion Miller (2019) draws in his survey is that statistical generalizations are unsatisfying to explain why events occur unless they can be accompanied by an underlying causal explanation for the generalization itself.

Where possible, we will try to incorporate these insights into our approach. In the end, we will reflect on the extent to which we succeeded in designing our methods in accordance with these principles.

2.6 Related approaches

In this section, an overview of related work - approaches in which a model-agnostic local explanation is generated - is presented. An overview of the entire field of interpretable machine learning is beyond the scope of this work. Moreover, an excellent overview, consisting of 84 interpretable machine-learning methods, is recently published by Robeer (2018).

The existing methods can be differentiated according to their outputs (Molnar 2019). Some methods, including the method that will be used in this work, can be assigned to multiple of these result categories. Results of local, model-agnostic approaches can consist of: feature summary statistics, feature summary visualizations or data points. We will shortly discuss all of them.

2.6.1 Feature summary statistic

Many of the interpretation methods focus on explanation through presenting information about the importance of features (Molnar 2019). Typically this is done by calculating weights representing the importance per feature. In case of a regression model, feature importance is a direct representation of how the model operates; the predictions of the model are made by summing all weights. In the past years, multiple methods have been developed that try to establish which importance a black box model assigns to features. These methods calculate weights based on how the prediction model behaves given a variety of input combina-

tions. Lundberg and Lee (2017) recently showed that there are equivalences among the techniques used for obtaining explanations by feature summary statistics. In an attempt to unify the approaches, they introduced Shapley Additive Explanations (SHAP). SHAP is based on the game-theoretical concept of the Shapley Value and comes with theoretical guarantees, such as a fair distribution of the average prediction over the features.

2.6.2 Feature summary visualization

For most of the summary statistics, visualizations are used to present the information to users. For example, SHAP output, as shown in 2.2, can be easily communicated through visualization of the feature contributions.

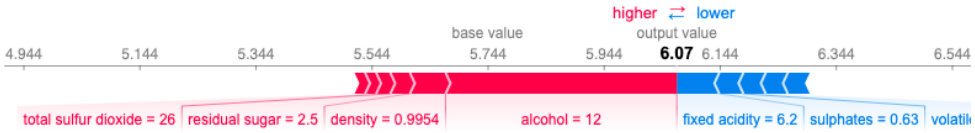

Figure 2.2: *Example of SHAP output on a single prediction. The size of a feature represents the strength of the contribution, the colour represents the direction.*

There are also feature summary methods that are only meaningful when presented visually. An example of such a method is the creation of a partial dependence plot (PDP). In a PDP, the marginal effect that one or two features have on the predicted outcome is shown (Friedman 2001). The information of this method can be communicated much more effectively by drawing a curve than by printing coordinates (Molnar 2019).

2.6.3 Data point

Another approach to increase the interpretability of a model is by returning either existing or newly created data points (Molnar 2019). The results of those methods are called *example-based explanations* (Adadi and Berrada 2018). A system taking this approach could, for example, show the data points that are most similar to the instance of interest. There are also methods that make a different selection of data points. *Prototypes and criticism* is an example-based method in which a collection of the most representative and the most exceptional

instances from the data is selected (Kim, Khanna, and Koyejo 2016). Another approach using data points is to look for counterfactual examples. A counterfactual explanation describes what the minimum conditions are that need to be met to change the outcome (Wachter, Mittelstadt, and Russell 2017).

2.6.4 Combining methods

Feature summaries - either textual or visualized - and example-based method both have different, complementary functions. Feature summaries provide insights into which aspects of an instance are of main importance. Example-based methods provide a context around individual instances. This context can help answer questions like: 'Which outcomes did similar instances receive?'. Moreover, example-based methods could be used to answer contrastive questions, such as: 'Why did this customer leave our company rather than that customer?'

A prerequisite for the interpretability of example-based approaches is that it is possible to interpret a single instance (Molnar 2019). This can be challenging for tabular data consisting of many features. To compare and visualize the instances in a meaningful way, we will combine an example-based approach with the usage of feature summaries. Considering the differences and similarities between the features of examples, we can reason about what an appropriate outcome of a new instance would be. The concept of using examples from the past to reason about new problems is known as *Case-Based Reasoning*. In the next chapter, we will further introduce this field of research.

3. Case-Based Reasoning

Case-based reasoning (CBR) means using experiences from the past to understand and solve new challenges (Kolodner 1992). The applications of the method can include adapting old solutions to meet new demands, using past cases to explain or interpret new situations or using past cases to criticize or justify new solutions (Kolodner 1992).

3.1 The potential of CBR for explanation

A CBR approach is a natural way to explain supervised machine-learning applications, as the input data can directly be used as cases (Prakken 2020). Every instance of the training data constitutes a single decided case, consisting of several properties - the input features - and an outcome. Together they make up the so-called *case base*: the collection of cases for which the results are known. New incoming instances are handled as undecided cases, for which one can reason about an appropriate outcome based on the knowledge in the case base.

Apart from the convenient technical fit, CBR seems a way of reasoning that is natural to humans. Explanation by analogy is shown to be a form of explanation to which people can easily relate (Cunningham, Doyle, and Loughrey 2003; Gentner, Loewenstein, and Thompson 2003). Cunningham, Doyle, and Loughrey (2003) empirically evaluated the usefulness of case-based explanation. They found that simply displaying a similar case along with the solution significantly improved the confidence the user had in the solution compared to showing just the solution or displaying a rule that was used to find the solution. A case-based approach also seems to offer a fruitful basis for taking into account some of the major social insights Miller (2019) found in his extensive review. CBR seems

especially suitable to create contrastive explanations, as it can be used to explain the case of interest relative to similar other cases that resulted in a different outcome.

Another advantage of CBR is that the method behind it has quite a transparent appearance. It is easy to comprehend the concept of searching for a similar case to solve the current problem (Sørmo, Cassens, and Aamodt 2005). The fact that the method is making use of 'real evidence', the items from the training data, promotes transparency.

3.2 Selecting cases

The usefulness of CBR is dependent on the ability of the user to understand the cases and to confirm the similarity assessment (Sørmo, Cassens, and Aamodt 2005). Adhering to those criteria becomes difficult when the structure of the cases is involved, or the similarity between the cases is far-fetched. A central topic within research on CBR concerns the issue of how to select an appropriate case.

Displaying the most similar case is the traditionally dominant form of explanation in CBR (Sørmo, Cassens, and Aamodt 2005). This practice was challenged by Doyle et al. (2004), who argued that the most similar case is not necessarily the most convincing case to display. Instead, they suggest selecting a case that is between the decision boundary and the case of focus. When trying to explain to a user that a customer will churn, it is more convincing to show an example of a customer who had a more favourable profile to the company and churned, even if a customer with a more negative profile is the closest match. Prakken (2020) takes a similar standpoint and distinguishes between *relevant differences* and other differences.

Independent of the decision to select the most similar or the arguably most convincing case, a second challenge remains in deciding which function to use to calculate the distance. A simple way to do this would be to determine the similarity between cases based on the number of features they have in common. However, when there is a wide variety in the contributions of different features to the outcome, this may lead to undesired results. A case that has a lot of relatively insignificant features in common with the case of focus may be considered less

similar to it than one that has few corresponding features that do have a substantial influence on the prediction.

As shown by Cunningham, Doyle, and Loughrey (2003), adding a similar case to the presentation of a solution already makes the user more accepting towards the solution. However, this approach seems somewhat limited (Karacapilidis, Trousse, and Papadias 1997). The user does not receive any help by comparing the relevant aspects of the cases. Besides, there is no possibility to adapt the explanation to the user or to let the user interact with the explanation system. Some researchers have tried to take CBR explanations to the next level by using argumentation for reasoning about the relevant similarities and differences between the cases.

3.3 Case-based argumentation

Case-based argumentation (CBA) is a sub-field within CBR, which originates from AI & law research on argumentation with cases. The research models how lawyers can refer to cases in the past and discuss their relevant similarities and differences. Case-based argumentation methods can be applied to a broader scope of problems. They are in particular suitable for problems that are not decided by a clear rule, but by weighting sets of relevant factors pro and con (Prakken 2020).

There is a small amount of existing work on using CBA to explain machine-learning algorithms. Cyras, Satoh, and Toni (2016) generated an explainable classifier using CBA. This system can make predictions for new cases and explain its own reasoning. Later, Čyras et al. (2019) reused this approach to use CBA for a related purpose; explaining outputs that were determined by humans. Prakken (2020) recently proposed a method which aims to generate explanations to the predictions of black-box prediction models. All these studies apply CBA to generate argumentative interpretations of predictions through a *dialogue* - an exchange of conflicting arguments - between two parties. In the next chapter, we will give an introduction to the formal argumentation theory they applied and discuss the two approaches.

4. Case-Based Argumentation

4.1 The one who laughs last laughs best

The approaches of Čyras et al. (2019) and Prakken (2020) rely on abstract argumentation frameworks, as introduced by Dung (1995). Dung was inspired by the observation that human argumentation relies on a simple principle: the one with the last word wins the argument. In the same way, a statement is considered believable when it can be successfully defended against any attacking arguments.

4.2 Formalization

Dung tried to formalize this notion into an abstract framework. An AA framework (AAF) is a pair AAF = $\langle A, attack \rangle$, where A is a set of arguments and $attack$ is a binary relation on A. Argumentation frameworks are usually presented in the form of a graph, in which the nodes represent the arguments, and the edges represent the attack relations between them. An argument A is treated as an abstract entity; the role of the argument is determined by its relation to other arguments, without considering the internal structure of the arguments.

For the sets of arguments $X, X' \subseteq A$ and the arguments $a,\ b \in A$, we say that:

- *a one-way attacks b* if *a attacks b* and *b* does not attack *a*
- *X attacks b* if there is an element $a \in X$ which attacks *b*
- *X attacks* X' if there is an element $b \in X'$ which is attacked by *X*
- *X* is *conflict-free* if *X* does not attack itself
- *X defends a* if for all *b* that *attack a* it holds that *X attacks b*
- *X* is *admissible* if *X* is *conflict-free* and *X defends* all elements in *X*.

The theory of Dung identifies different forms of admissible sets, called extensions. Čyras et al. (2019) and Prakken (2020) focus on the *grounded extension*. We call an extension *grounded* when it is admissible, contains all arguments it defends and is subset-minimal for those conditions. The grounded extension always exists, may be empty and is unique. It can be built incrementally by iterating over the following steps until no more changes to the argument graph can be made:

1. Adding all arguments that are not attacked to the grounded extension
2. Removing all arguments that are attacked by an argument in the grounded extension from the graph and returning to step 1

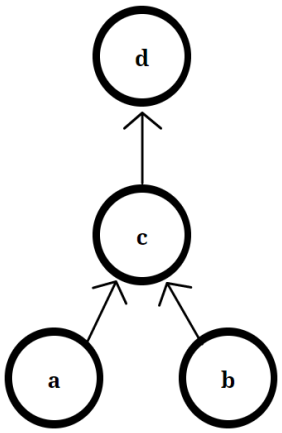

Figure 4.1: *AAF first example*

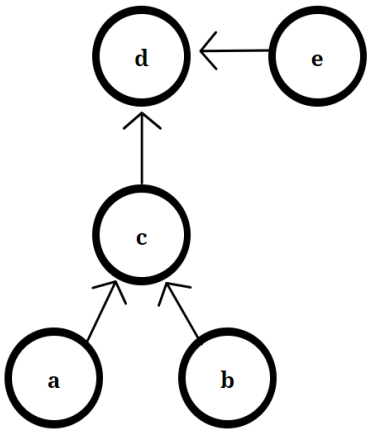

Figure 4.2: *AAF second example*

In Figure 4.1, we see an AAF with four arguments, represented as nodes. An outgoing arrow from x to y, represents that argument x attacks argument y. In our example, argument a and b are not attacked, so must be part of the grounded extension. As this set attacks the only attacker of d - argument c - d must be included in the grounded extension as well.

Another way to verify whether an argument is part of the grounded extension is to use an *argument game* (Modgil and Caminada 2009). An argument game is a formal dialogue between two players: a proponent and an opponent. The proponent begins the game by proposing an argument. Then the players take turns after each argument, in which:

- The opponent must attack the last argument of the proponent

- The proponent must one-way attack the last argument of the opponent

In correspondence with the principle derived from human argumentation - "the one who laughs last laughs best" - a player wins an argument game if its counterparty can no longer move. In our example in Figure 4.1, the proponent could start the game by playing argument d. The opponent has only one response: to attack d with c. After that, the proponent can win the game by either playing a or b, leaving the opponent out of moves. In Figure 4.2, we see an AAF that could lead to a different result. Instead of playing c, the opponent could now respond to d by playing e. Since e has no attackers, this would leave the proponent out of moves.

A way for a player to play against all possible moves of the opponent is called a *strategy*. We say that a strategy for a player is *winning* if that strategy makes the player win independently of what its opponent does. If the proponent has a *winning strategy* in a game for the proposed argument, the argument is called *justified* in the grounded extension. This makes argument d to be justified in 4.1 and unjustified in 4.2.

4.3 AA-CBR

Cyras, Satoh, and Toni (2016) proposed a method combining AA and CBR called AA-CBR. They aimed at generating an explainable classification system, which can establish outcomes for new cases independently. The method was later used in ANNA (Cocarascu, Cyras, and Toni 2018) and formed the basis of the approach of Čyras et al. (2019) to explain outcomes determined by humans.

In their method, they create an AAF to determine which binary outcome should be attributed to a new case, which we will call the *focus case*. They assume that there is a *default outcome*, d, which should be assigned to a case if there is no counter-evidence. Every *case* - a set of features together with an outcome - in the case base, the focus case and the default outcome make up the arguments in the AAF. Within the framework, an argument A attacks B if the following three conditions are met:

- A and B have different outcomes
- A is more specific than B; $features(B) \subset features(A)$
- A is as close as possible to B; there is no C such that $features(B) \subset features(C) \subset$

$features(A)$

Furthermore, the new case N attacks any case Y that has an element which is not in N; $Y \not\subseteq N$. Since N does not have an outcome yet, it cannot be attacked by other cases. The prediction for the new case is dependent on the default outcome. When the default outcome is part of the grounded extension, the new case is assigned this outcome. We can say that the default outcome is sceptically justified; the non-default outcome can only be reached if there is some justified counterargument against the default outcome. When the default outcome is unjustified in the argument graph, the system predicts the opposite outcome.

To illustrate this, and the next approach, we will consider a Churn scenario. In this scenario, we consider four binary features, which are presented in Figure 4.3.

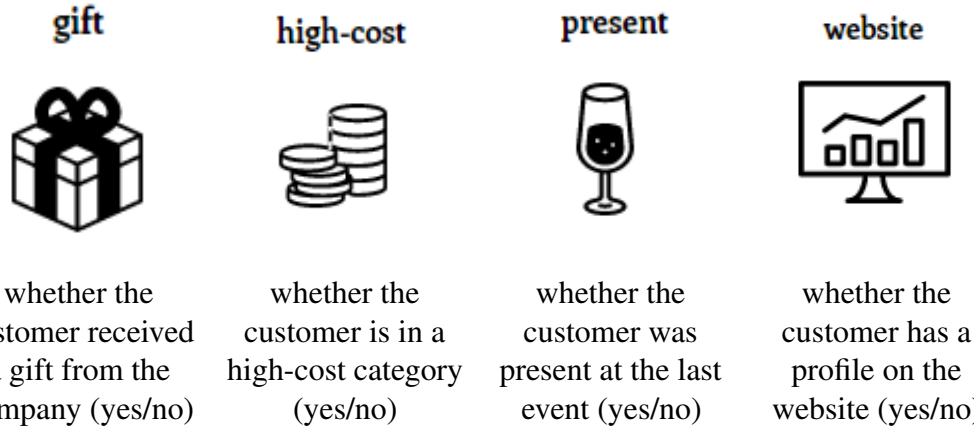

Figure 4.3: *Features Churn example*

Furthermore, we have a small data set with information about four previous customers (Table 6.1). The label - or outcome variable - 'churn' tells us whether the customer decided to leave the company or stayed. As our default outcome, we will assume a customer to stay. In the other four columns, a value of 0 represents that a feature was absent; a value of 1 that a feature was present for a customer.

We can transform the data set and default outcome into an AAF, as represented in Figure 4.4. Within the framework, there are four attack relations. *Nash* and *Dong* attack the default outcome, as they have different outcomes, are more specific and are as close as possible to the default case; there is no case which

customer	gift	high-cost	present	website	churn
Miss Hill	1	0	0	1	0
Mister Nash	1	1	1	0	1
Mister Wang	0	0	1	1	0
Miss Dong	0	0	0	1	1

Table 4.1: *Churn example data set*

features are both a proper superset of the attackee (the default case) and a proper subset of the features of *Nash* or *Dong*. For the same reasons, *Hill* and *Wang* attack *Dong*.

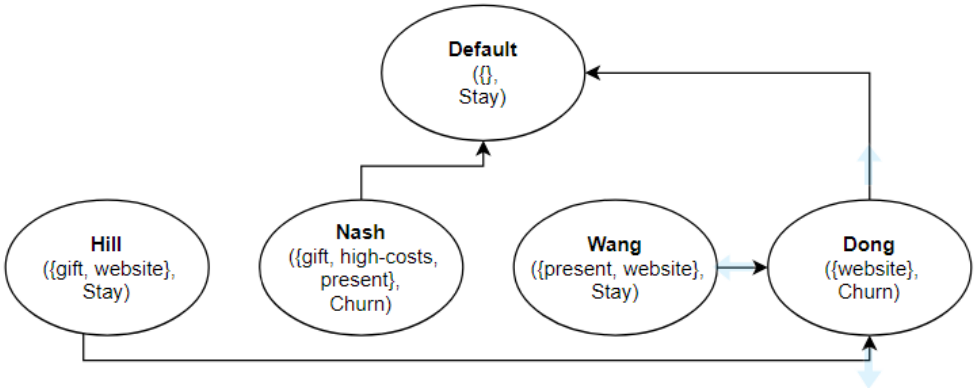

Figure 4.4: *The data set of the Churn example represented as an AAF*

Suppose that we have some new customer, Miss Jale (Table 6.2). Miss Jale is our focus case: we don't know her decision yet, and so she is the focus of our prediction task.

customer	gift	high-cost	present	website	churn
Miss Jale	0	0	1	1	?

Table 4.2: *Churn example focus customer*

To obtain the prediction for Jale, we add her case to the AAF. As figure 4.5 shows, *Jale* attacks both *Hill* and *Nash*. This is because both customers have a feature which *Jale* is missing. Now that *Nash* is defeated, the default outcome becomes part of the grounded extension. Therefore, the system predicts *Jale* will stay.

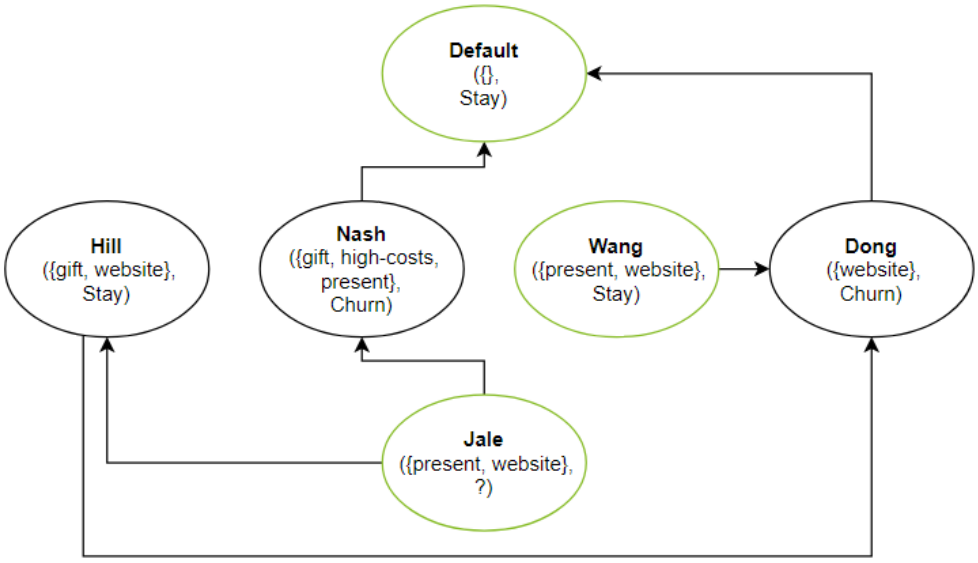

Figure 4.5: *Arguments AAF - Churn example AA-CBR*

To explain a prediction to a user, the authors use an argument game. As the outcome of the focus case is dependent on the status of the default outcome, the proponent starts the dialogue with this argument. When the default outcome is justified, any admissible dialogue tree of d can serve as the explanation. Figure 4.6 shows an example of what an admissible tree for Miss Jale could look like.

When the non-default outcome prevails, no admissible dialogue trees exist. In those cases, the authors deploy a *maximal dialogue tree*, a tree in which no opponent nodes can be attacked, for this purpose.

4.4 AF-CBA

Building on AA-CBR, Prakken (2020) proposed a new approach, which we will call A Fortiori Case-Based Argumentation (AF-CBA). AF-CBA is designed to explain predictions of other classifiers in a model-agnostic way. Instead of determining an outcome by itself - as AA-CBR does - it takes the prediction of another classifier as the starting point.

Prakken (2020) introduces in the second part of his article an explanation method

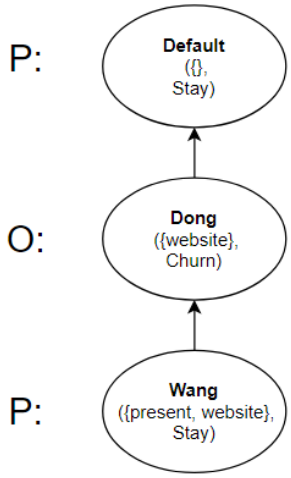

Figure 4.6: *Possible explanation tree for customer Jale - Churn example AA-CBR*

which can process multi-valued features. We will only consider this method and, therefore, slightly adjust our running example. We transform the binary feature 'website' into a multi-valued feature: instead of representing whether the customer has an account on the website, it now represents the number of times the customer logged into the website (0 - ∞). Table 4.3 shows our renewed data set.

customer	gift	high-cost	present	website	churn
Miss Hill	1	0	0	5	0
Mister Nash	1	1	1	3	1
Mister Wang	0	0	1	6	0
Miss Dong	0	0	0	1	1

Table 4.3: *Churn example data set*

AF-CBA is based on Horty's factor- and dimension-based result models of precedential constraint (Horty 2011; Horty 2019). In these models, Horty uses *a fortiori reasoning* to reason about new cases. According to this way of reasoning, a new case must adopt the outcome of another case if all their differences make the new case even stronger for that outcome.

In order to speak about one case being stronger for an outcome than another, we must first introduce some definitions and notation. All definitions are adopted from Prakken (2020), although sometimes with small notational differences.

Prakken (2020) refers to features as *dimensions*. Given a dimension d, he defines a *value assignment* as a pair (d, v), where v represents the value. A list of value assignments is called a *fact situation*. A *case* is then defined as a pair $c = (F, outcome(c))$, where F is a fact situation, and $outcome(c) \in \{o, \overline{o}\}$, where o and $\overline{o}$ represent the two available outcome values. Miss Hill can be represented as the following case: ([(gift, 1), (high-cost, 0), (present, 0), (website, 5)], Stay).

A case base is a set of cases relative to a set D of dimensions; all cases assign values to a dimension d if and only if $d \in D$. Like Prakken (2020), we will use $F(c)$ to denote the fact situation of case c, and $v(d, c)$ to denote the value of dimension d in case c.

Given two value assignments (d, x) and (d, y), we write:

$$x \leq_o y$$

to denote that value y for d favors outcome o at least as strongly as value x for d. For example, for the dimension 'website', we could state that a value of 1, favors the outcome Churn at least as strongly as the value 3, written: $3 \leq_{Churn} 1$. Note that the ordering does not tell us whether the values 1 or 3 favor outcome Churn; all we know is that value 1 promotes churning at least as strong as value 3.

The expression x $\leq_o$ y is equal to y $\geq_o$ x. Prakken (2020) defines a dimension as a tuple $d = (V, \leq_o, \leq_{\overline{o}})$, where V is the set of values and $\leq_o$ and $\leq_{\overline{o}}$ are two partial orders on V such that $v \leq_o v'$ iff $v' \leq_{\overline{o}} v$. Now we can introduce Horty's preference relation on fact situations as:

Definition 1. [Preference relation on fact situations.] Let F and F' be two fact situation with the same set of dimensions. Then F' is at least as strong as F for outcome o , written $F \leq^o F'$, if and only if, for all $(d, v) \in F$ and all $(d, v') \in F'$ it holds that $v(d) \leq^o v'(d)$.

In other words, a fact situation F is at least as strong for outcome o as fact situation F' when the value of F is at least as strong for outcome o as the value of F' on every dimension. We say that a case base is *inconsistent* if and only if it includes factor sets X and Y, such that for some new fact situation F we could have $X \leq_o F$ and $Y \leq_{\overline{o}} F$. A case base that is not inconsistent is *consistent*.

Given the preference relation on fact situations, the a fortiori rule of Horty applied to cases is defined as:

Definition 2. [A fortiori rule.] Let CB be a case base and F a fact situation given a set of dimensions D. Then, given CB, assigning outcome o to F is *forced* if and only if there exists a case $c = (F', s) \in CB$ such that $F' \leq^o F$.

In other words, we must assign the outcome of case c to fact situation F when F is at least as strong for $outcome(c)$ as fact situation F' of case c. Assigning both outcome o and $\overline{o}$ to F can only be forced if the case base is inconsistent.

Before we can apply a fortiori reasoning to our example, we must establish the tendencies of the features to favor outcomes. For this strength ordering, Prakken (2020) argues that it is inadequate to represent factors - binary features - in the same way as other dimensions. He states that for some factors we want to avoid regarding one value as favoring outcome o, written $pro - o$, and the other value as discouraging outcome o, denoted as $con - o$. Take for example the factor 'gift'. The presence of this factor seems a factor pro-Stay. However, not receiving a gift does not seem to be a con-Stay factor, but rather a neutral outcome. He, therefore, decides to treat factors as a different type of dimensions. Each factor comes with a partial function t_d: V -> o, o' that assigns to zero, one or both values of the dimension (v and $\overline{v}$) an outcome, such that:

1. if $t_d(v) = o$ then $t_d(\overline{v}) = \overline{o}$ or $t_d(\overline{v})$ is undefined
2. if $t_d(v) = o$ then $(\overline{v}) <_o v$

From now on, we will refer to two-valued dimensions as *factors* and to multi-valued dimensions as *dimensions*.

For our running example, we will make some assumptions about the tendencies of the features to favor outcomes. Prakken (2020) does not specify a way to determine these tendencies but assumes this information to be available. We will assume that $t_d(v)$ = Stay for factors 'gift' and 'present' with value 1, and call these factors pro-Stay. Next, we assume that $t_d(v)$ = Churn for 'high-cost' with value 1, and call this a pro-Churn factor. We will leave the other values of the factors undefined. For our multi-valued dimension - 'website' - we assume that the higher a value on this dimension is, the stronger the value becomes for the outcome Stay: $v(d) >_{Stay} v'(d)$ and $v(d) <_{Churn} v'(d)$ if and only if $v > v'$. In other words, the more often a customer logs into the website, the more likely we assume the customer to stay.

When all differences between a case c and fact situation F, make F at least as strong for $outcome(c)$ as c, we say that c has no *relevant differences* with F.

Prakken (2020) formally defines these differences as:

Definition 3. [Relevant differences between cases] Let $c = (F(c), outcome(c))$ and $f = (F(f), outcome(f))$ be two cases. The set $D(c, f)$ of relevant differences between c and f is defined as follows:

1. If $outcome(c) = outcome(f) = o$ then $D(c, f) = \{(d, v) \in F(c) \mid v(d, c) >_o v(d, f)\}$
2. If $outcome(c) \neq outcome(f)$ where $outcome(c) = o$ then $D(c, f) = \{(d, v) \in F(c) \mid v(d, c) <_o v(d, f)$.

As proven in Prakken (2019), a simple criterion can be used to determine whether a case is forced:

Proposition 1. Let CB be a case base and f a case with with fact situation F. Then deciding f for o is forced given CB if and only if there exists a case c with outcome o in CB such that $D(c, f) = \emptyset$.

This allows us to start reasoning about the outcomes of new cases. Given a case base consisting of our example customers *Hill, Nash, Wang and Dong*, consider first a fact situation F: [(gift, 0), (high-cost, 0), (present, 1), (website, 8)]. Because each value assignment of F is at least as strong for outcome Stay as that of *Wang*, F would be forced to obtain $outcome(Wang)$.

Now suppose that we would have the following focus customer, Miss Jale (Table 6.2). We represent her fact situation as: *Jale* = [(gift, 0), (high-cost, 0), (present, 1), (website, 5)].

customer	gift	high-cost	present	website	churn
Miss Jale	0	0	1	5	?

Table 4.4: *Churn example focus customer*

In this scenario, *Jale* has relevant differences with all cases in the case base. Therefore, neither assigning outcome Stay, nor Churn is forced. To handle these situations, Prakken (2020) introduces a top-level model of case-based explanation dialogues. Like in AA-CBR, dialogues are formalized as the grounded game of an AAF. We will informally sketch the dynamics of the explanation dialogues; for the formalization and further details, we refer to the article of Prakken (2020).

In the game, the proponent tries to justify the prediction of the classifier. To justify the prediction, the proponent cites an example case - which we will call *precedent* - from the case base that received an outcome equal to the prediction. The opponent than tries to distinguish the focus case from the precedent, while the proponent aims to *explain away* all of their differences.

When the cited precedent has no relevant differences with the focus case, this leads directly to a fortiori justification of the prediction. When there are relevant differences between the two cases, the opponent comes into play. The opponent can distinguish the precedent on its relevant differences with the focus case or cite a counterexample. After that, the proponent has the opportunity to defend against those attacks, showing the differences can be compensated or do not matter.

There is a fixed structure of the dialogue tree, which only allows branches to have a maximum length of three moves. A visualization of the high-level structure of the dialogue is presented in Figure 4.7.

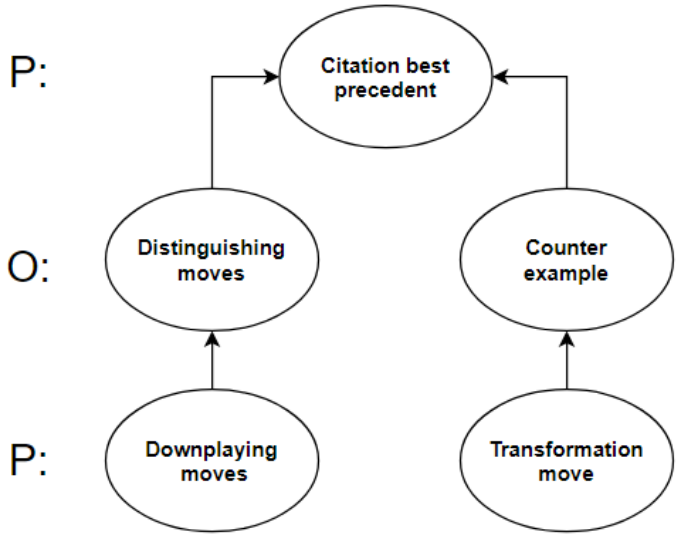

Figure 4.7: *The high-level structure of the dialogue game tree*

We will explain the different moves in some more detail, after which we will apply the approach to our example.

For the game, we will assume that there is some classification model that predicts for any focus case f an outcome $prediction(f) \in \{o, \overline{o}\}$. The proponent

starts the dialogue of a focus case by citing a *best precedent*. A best precedent for focus case f and $prediction(f)$, is defined as any case $c \in CB$ that meets the following two conditions:

1. $outcome(c) = prediction(f)$
2. There is no case $c' \in CB$ with $outcome(c') = prediction(f)$ and $D(c', f) \subset D(c, f)$

The second condition entails that a best precedent must have a minimal subset of relevant differences among the cases with that outcome.

The opponent has two approaches to reply to the citation. We will first consider the distinguishing moves. A distinguishing move points out any relevant differences between the focus case and the cited precedent. Prakken (2020) defines two types of distinguishing moves for factors:

> **MissingPro**(c, x): the focus case lacks pro-factors x of precedent c
> **NewCon**(c, x): the focus case contains con-factors x that are not in precedent c

and one for dimensions:

> **Worse**(c, x): the focus case has less favorable values on dimensions x than precedent c

To defend against those attacks, the proponent can apply downplaying moves. A downplaying move can be used to neutralize a distinguishing move the opponent played. Downplaying is then used as a way to argue why an attack does not undermine the representativeness of the selected case. Two concepts constitute the basis of the downplaying moves for factors:

> **Substitution**: factors can substitute each other, removing the difference on which the cases were distinguished
> **Cancellation**: factors can cancel out each others effect, removing the difference on which the cases were distinguished

and one the basis for downplaying dimensions:

> **Compensation**: more favorable values on some dimensions, can compensate for less favorable values on other dimensions

The input for the downplaying moves can be found in the differences between the focus case and precedent that are not part of the 'relevant differences'. Prakken

(2020) does not link any conditions to the application of downplaying moves himself - the moves are even allowed to be empty. His approach leaves the specifications of such conditions to top-level knowledge; he assumes an unspecified set *sc* of rules to prescribe which downplaying moves can be applied.

The second way the opponent can respond to the citation is by citing a counterexample. For focus case f, $prediction(f)$ and cited precedent p, any case $c \in CB$ that meets the following two conditions is allowed to be used as a counterexample, attacking p:

1. $outcome(c) \neq prediction(f)$
2. $D(p, f) \not\subset D(c, f)$

The second condition entails that the set of relevant differences of the precedent may not be a proper subset of the relevant differences of the counterexample.

The proponent can defend against a counterexample by using the same concepts used for the downplaying moves. These concepts are applied to *transform* the own cited precedent into a case that has no relevant differences with the focus case. By demonstrating that the precedent can be transformed in a case without relevant differences, the proponent defeats the counterexample.

We will now apply the dialogue game to the running example, considering Miss Jale as our focus case. Her fact situation looks like this:

$$Jale = [(\text{gift}, 0), (\text{high-cost}, 0), (\text{present}, 1), (\text{website}, 5)]$$

Our case base consists of the four customer from the data set, transformed into cases:

$Hill$ = ([(gift, 1), (high-cost, 0), (present, 0), (website, 5)], Stay)
$Nash$ = ([(gift, 1), (high-cost, 1), (present, 1), (website, 3)], Churn)
$Wang$ = ([(gift, 0), (high-cost, 0), (present, 1), (website, 6)], Stay)
$Dong$ = ([(gift, 0), (high-cost, 0), (present, 0), (website, 1)], Churn)

We will suppose some black-box classifier predicted $Jale$ to stay. The proponent now starts the game by citing a best precedent. In our case base, we have two cases with an outcome equal to the prediction Stay: $Hill$ and $Wang$. The relevant differences of these cases with $Jale$ are:

$$D(Hill, Jale) = \{(\text{gift}, 1)\} \text{ and } D(Wang, Jale) = \{(\text{website}, 6)\}$$

As none of the sets is a proper subset of the other, $Hill$ and *Wang* could both be cited by the proponent. In this example, we will suppose the proponent cites case $Hill$. As $Hill$ has relevant differences with *Jale*, the opponent can distinguish the precedent on its differences with the focus case. Specifically, the opponent can play *MissingPro(Hill, gift)* pointing out the fact that *Jale* lacks the pro-factor gift of precedent $Hill$.

In addition, the opponent can play a counterexample. A counterexample must be a case with the opposite outcome, so either *Nash* or *Dong*. As $D(Hill, Jale)$ is no proper subset of $D(Nash, Jale)$, nor $D(Dong, Jale)$, both cases could be applied as counterexamples. For the course of the game, it never matters which counterexample is chosen, as the response of the proponent is independent of the properties of the counterexample.

In response to the attacks of the opponent, the proponent will look for downplaying possibilities. Apart from the 'relevant differences', *Jale* and $Hill$ also differ on the factor 'present'. *Jale* was present at the last customer event, whereas $Hill$ was not. This pro-factor can be used to downplay *MissingPro(Hill, gift)* with *pSubstitutes(present, gift, Hill)* stating that the missing pro-factor 'gift' is in a sense still in *Jale*, as it can be substituted by the extra pro-factor 'present'.

Finally, the extra pro-factor can also be used to reply to the counterexample. The proponent can transform $Hill$ into a new case $Hill'$ = ([(gift, 0), (high-cost, 0), (present, 1), (website, 5)], Stay) making use of the move *pSubstitutes(present, gift, Hill)*. The new case $Hill'$ is identical to *Jale*, so cannot have any relevant differences.. Using this transformed case, the proponent replies $Transformed(Hill, Hill')$ to attack the counterexample. The dialogue game tree of the example is visualized in Figure 4.8.

Using the information generated with the dialogue, Prakken (2020) proposes that AF-CBA should at least present to the explainee:

1. Whether the focus case is *forced*, meaning the prediction follows from the a fortiori rule.
2. *If the focus case is forced*
   every precedent with no relevant differences can be used as the explanation
   Else,
   the sequence of downplaying moves derived from the winning strategy is shown. This sequence explains what needs to be accepted to

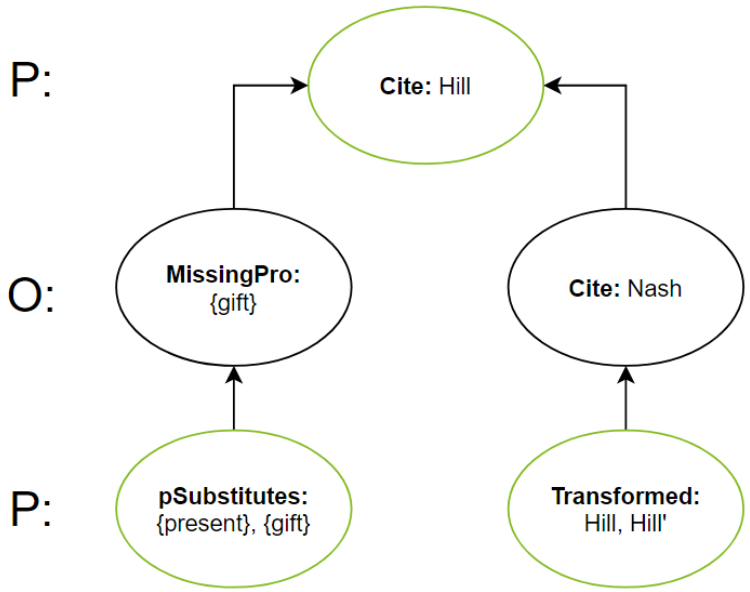

Figure 4.8: *Example dialogue game tree*

make the prediction forced.

4.5 Research approach

In this research, we aim to further investigate the possibilities of CBA in relation to the challenges we face with black-box algorithms. We will take the proposal of Prakken (2020) as our starting point. This approach has not been tested yet and deviates in two interesting ways from AA-CBR. Instead of functioning as a classifier itself, AF-CBA is designed to explain the predictions of a black box. Secondly, the method allows the inclusion of multi-valued input features, broadening the scope of application. By bringing the proposed method into practice, the research aims to contribute to answering the following research question:

RQ) *To what extent is CBA applicable for making machine learning more interpretable?*

The approach taken in this research is a form of *Design science*. Within this field of science, researchers try to improve a problem context and to answer knowledge questions by designing and investigating artefacts (Wieringa 2014).

Five sub-questions will be addressed to answer the main question. These questions arise at different levels of the research process. The first questions focus on AF-CBA specifically. Later, we will take a broader perspective and consider the general possibilities of applying CBA to obtain interpretable machine learning.

When applying AF-CBA, information about the impact of different features on the prediction should be collected. This is necessary to establish the tendencies of the features and to enable reasoning about the differences between cases. How this information can be generated is the focus of the first sub-question.

S1) *What kind of method is applicable for generating feature information to be used in AF-CBA?*

Prakken (2020) proposes that feature information could be generated by consulting an expert to provide a factor hierarchy manually. Due to the practical constraints coming along with that, this research will experiment with generating information about the input features automatically. Using a feature importance estimator the direction and strength of the impact of a feature on a particular outcome will be calculated. Those scores can then be applied to compare cases and decide which features can be used to compensate for others.

As previously discussed, deciding on the way cases are being selected poses a challenge to every CBR system. The second sub-question is dedicated to that topic.

S2) *Which cases should be selected to be used in AF-CBA?*

Answering this question includes looking for a suitable distance function, as well as defining the search algorithms for ‘best precedents’ and ‘counterexamples’. There is another challenge regarding the number of cases that will be selected. In the dialogue tree of AF-CBA, only a single precedent and counterexample are used. This number of examples would seem reasonable in the ideal scenario in which a case base only consists of instances for which the ‘ground truth’, or at least some well-established outcome, is known. For example, when lawyers

draw analogies to past cases or discuss their relevant similarities and differences, they can, in principle, rely on the decisions made in a single case.

In the case of a model that is trained based on statistical relations, the relations between the input-features and the outcome in the training data are usually far from being 'ground truths'. Although we aim features to cover the complete causal picture, in practice, this is rarely the case. The training data of a churn prediction model could contain two customers with precisely the same input-features, of which one stayed at the company and one churned. Alternatively - more extreme - there could be a customer for which all the input-features predicted green light, who due to some unknown reason left the company.

This seems to make the approach suggested by Prakken (2020) problematic when applied to a machine-learning model trained on statistical data in which the features are not covering the complete causal picture. When there is an extreme case in the data, one for which the outcome is unexpected based on the features, that case could be selected as best precedent or counterexample for cases for which we would not want to find supporting evidence. A potential solution could be to refer to multiple cases or statistics of cases, instead of to a single case.

After the single case or multiple cases have been selected, they can be attacked in the argument game. An attacking move distinguishes the selected case from the focus case on relevant criteria. The application algorithm of the downplaying moves will be the focus of the third sub-question.

S3) *How to apply downplaying moves in AF-CBA?*

The topics of investigation in this section will be the order in which downplaying moves should be applied and the criteria to which features must adhere to compensate for each other.

A considerable disadvantage of the model-agnostic approach, as mentioned by Rudin (2019), is the fact that the explanation method is usually not entirely faithful to the black-box model. This problem applies to AF-CBA, which explanations are generated almost completely independently from the prediction model. In this section, we consider the possibilities for removing this separation between the explanation system and the prediction model.

S4) *What are the possibilities for closing the gap between AF-CBA and the prediction model?*

We will approach this question with two sub-questions. First, we will investigate whether we can enable the CBA-system to generate acceptable predictions itself, making the black-box model superfluous. Replacing the black box by the explanation system would eliminate the problems attached to a model-agnostic approach.

S4.1) *Have interpretable CBA-systems potential for taking over black-box models?*

In order to answer that question, we will transform AF-CBA into a new system which is able to make predictions by itself. After that, the accuracy of the model in predicting new outcomes can be measured. Results will be compared with AA-CBR and other (black-box) classifiers. Based on the difference in accuracy and interpretability between the CBA-systems and black-box models, there will be reasoned about whether there is still independent value to the black box.

When there is independent value to a black box, we might want to hold on to it. In the second part, we will investigate whether it is possible to incorporate further information about a black-box model in the CBA-system, while still holding on to a model-agnostic approach.

S4.2) *Are there model-agnostic possibilities to incorporate further information about the black-box model in CBA?*

Based on the work of Weerts, Ipenburg, and Pechenizkiy (2019), we will elaborate on an alternative approach. In this approach, we collect new information about the black box by measuring its predictions for the instances in the training data.

In the last subsection will be evaluated to what extent machine-learning explanations created with CBA are suitable for human users.

S5) *To what extent are explanations generated by CBA-systems suitable for users?*

To test the suitability of the systems for users, we will build user interfaces for two different types of CBA explanation systems. We then test these systems by conducting a user experiment.

5. Evaluation of AF-CBA

In this chapter, we will run experiments to evaluate AF-CBA, the explanation method proposed by Prakken (2020). We will first introduce the three data sets that were used for the experiments. After that, we discuss how the data instances were transformed into cases, making up the case bases. We will then conduct the experiments and evaluate the method based on the results.

5.1 About the data sets

For the experiments, we used three publicly available data sets: Churn (*Telco Customer Churn* 2018), Mushroom (Dua and Graff 2019) and Graduate Admission (Acharya, Armaan, and Antony 2019). All data sets are tabular, consisting of several features, together with an outcome variable. The Churn data set contains information about customers of a telecom service. The outcome variable *Churn* represents whether a customer continued using a company's telecom services or churned (cancelled the subscription). The Mushroom set consists of descriptions of hypothetical samples corresponding to species of mushrooms in the Agaricus and Lepiota Family. All features of this set are categorical, and the outcome variable represents whether the mushroom is either definitely edible or (possibly) poisonous. Čyras et al. (2019) also applied the Mushroom data set to test ANNA, which will allow us to compare our results. The Admission set consists of features - such as exam scores - with which a prediction can be made about whether an applicant will be admitted to a Master Program.

We made a heterogeneous selection of data sets in terms of consistency. The Churn data set is of a highly statistical nature; on average, we can tell which profiles are likely to stay or churn based on the features, but there are lots of

exceptions.

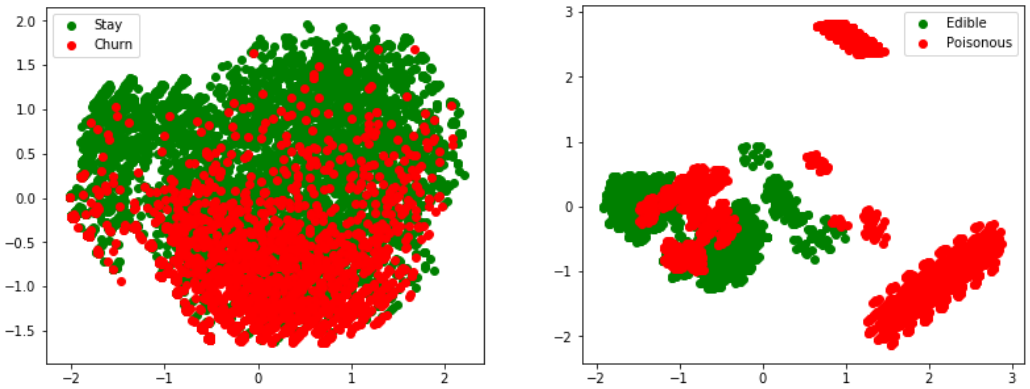

Figure 5.1: *A Principal Component Analysis of the Churn (left) and Mushroom (right) data sets.*

In the Mushroom data set, on the other hand, the features do seem to possess enough information to consider the outcome variables some 'truth'; cases with the same features must have the same outcome. Figure 5.1 shows the different structures of the Churn and Mushroom data sets, as visualized with a Principal Component Analysis, transforming the data points into two dimensions. The Admission data set forms a middle ground between the statistical Churn and consistent Mushroom data set. Further details about the data sets can be found in Table 5.1.

	Mushroom	Churn	Admission
number of instances	8124	7032	500
distribution outcome	48% poisonous	27% churned	7.8% refused
number of features	22	21	7
number of categorical features	22	18	1
number of continuous features	0	3	6

Table 5.1: *Data set statistics*

5.1.1 Implementation

To prepare the data for AF-CBA, we made a couple of modifications. The outcome values of the Graduate Admission data set were transformed into binary

values by replacing every value below 0.5 with 0, and other values with 1. We removed one feature from the Mushroom data set - 'veil-type' - for which all instances have the same value assignment.

We used an automatic approach to establish the tendencies of features to promote outcomes. All Mushroom and part of the Churn features consist of categorical values. To simplify these assignments, those features were transformed into binary features. This was done by creating one binary feature for every categorical value. This left us with data sets that consist only of binary and continuous features plus a binary outcome. For every feature, we measured its correlation with the two outcome values in the data set. When a feature has a positive correlation with outcome value s, we assumed that $t_d(v) = s$ for value 1. In all other cases, we left $t_d(v)$ undefined. For dimensions - the continuous features - we used a similar approach. When a dimension has a positive correlation with outcome s, we assumed every value v that is greater than value v' to be more favorable for outcome s.

Every row of data was transformed into a case. The feature values of an instance were represented as a *fact situation*: a list of pairs, (F, v), where F is the feature and v the value for that instance. A case is then a pair $Case(F, o)$, where F is the fact situation, and o the outcome value for that instance. Per data set, we created one case base - a list of cases - consisting of all instances.

5.1.2 Consistency

Before applying AF-CBA, we measured the consistency of the three case bases. We say that a case base is inconsistent if and only if it includes cases X and Y with $outcome(X) = s$ and $outcome(Y) = \bar{s}$, such that $X <_s Y$; Y is at least as good for outcome s as X. In other words, there is a case which received outcome $\bar{s}$, while its features are more favorable for the opposite outcome, outcome s, than those of a case with outcome s. In that scenario, a new case F with the same features as Y would be forced to receive both outcomes s and $\bar{s}$ according to the model of precedential constraints. A case base is consistent if and only if it is not inconsistent.

For our implementation, the Mushroom case base appeared to be consistent. This was different for the other case bases; for 20% of the Admission cases and 45% of the Churn cases, a case that is more favorable for that outcome but

received the opposite outcome could be found. This inconsistency seems caused by a couple of 'exceptional' cases. As Table 5.2 shows, removing respectively 3.4% and 11.1% of the most inconsistent cases, results in consistent Admission and Churn case bases.

	Mushroom	Churn	Admission
percentage consistent cases	100%	55%	80%
n removals for consistent CB	0 (0%)	780 (11.1%)	17 (3.4%)

Table 5.2: *Consistency statistics*

5.2 Experiments

In this section, we will present the results of the conducted computer experiments. Further interpretation and discussion of the results will be the topic of the next section. Unless stated otherwise, the experiments use for every data set a case base consisting of 500 randomly selected cases. For a single experiment, every case in the case base is used once as the focus case, while the 499 remaining cases are used as candidate precedents.

5.2.1 Selection of precedents

The first step to explaining a focus case for AF-CBA is to cite a *best precedent*. In this experiment, we will take a look at this selection process. Prakken (2020) defines a best precedent as a case in the case base that:

1. received the same outcome as the focus case
2. has a minimal set of relevant differences with the focus case compared to other cases in the case base

Multiple cases can meet both criteria. Table 5.3 shows the average and standard deviation of the number of best precedents that can be found.

When there are no relevant differences between a case and a precedent, the proponent has a trivial winning strategy in the argument game. In case of a trivial winning strategy, the opponent can not distinguish the focus case from the precedent. Technically, the opponent might be able to respond with a counterexample

that has no relevant differences either. However, this move could simply be defeated by citing the precedent again; a transformation is not needed as the precedent already has no relevant differences. The structure of the argument games will, therefore, be the same for every best precedent. As shown by Figure 5.2, a substantial percentage of focus cases has a trivial winning strategy, especially in the Admission case base.

	Mushroom	Churn	Admission
all cases	26.27 (29.20	9.16 (9.07)	105.92 (116.38))
non-trivial cases	39.55 (28.53)	14.05 (9.44)	6.22 (5.18)

Table 5.3: *Average and standard deviation of the number of best precedents*

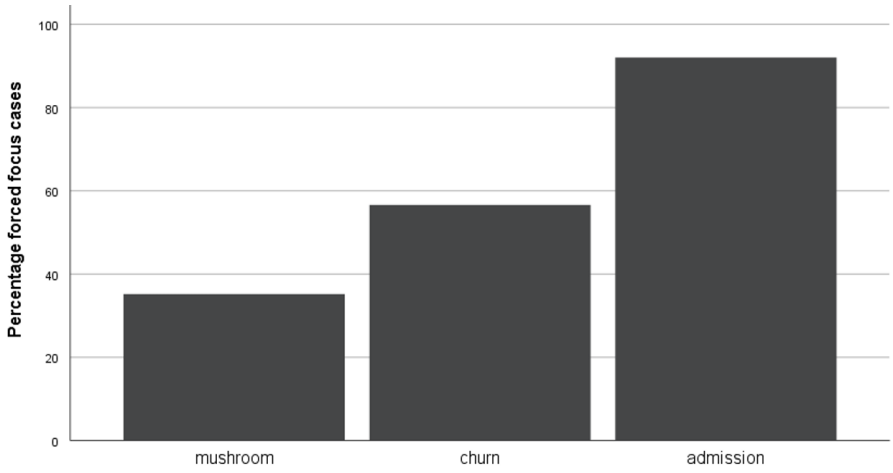

Figure 5.2: *Percentage of focus cases with a trivial winning strategy per data set*

The structure of the argument game can differ between best precedents that do have relevant differences with the focus case. A simple measure with which best precedents can be compared is whether any empty downplaying moves are needed to defend against attacks on the citation. An empty downplaying move is a way of saying that the differences between the focus case and precedent cannot be downplayed by other features, but still do not matter. This could be seen as the weakest form of attack.

To obtain an idea of whether there exist relevant differences between the best precedents that are selected for a focus case, we divided selections of best precedents into three groups: selections in which none, a part of or all of the prece-

dents need empty downplaying moves. In Table 5.4, the distribution over these three groups is shown for all focus cases with a non-trivial winning strategy.

	Mushroom	Churn	Admission
no precedents need empty downplay	75.6%	42.8%	65.0%
some precedents need empty downplay	24.4%	56.7%	35.0%
all precedents need empty downplay	0.0%	0.5%	0.0%

Table 5.4: *Percentages of focus cases with a non-trivial winning strategy for which none, some or all of the best precedents need empty downplay*

In only a single case of the Churn case base, was it necessary to use empty downplaying moves to defend any of the best precedents. In other cases, the system could at least defend part of the best precedents against the distinguishing moves by pointing to compensating features.

5.2.2 Resulting outcomes

The resulting explanations should express to the user whether the focus case is forced and if not, under which assumptions the outcome would be forced (Prakken 2020). The necessary assumptions are expressed in the argument game in the form of downplaying moves. If all downplaying moves are considered valid; either because they are part of definitions added as top-level information or because the user finds them acceptable, the citation is said to be justified, meaning that the focus case can rightfully adopt the outcome of the precedent.

For the Admission case base, a typical explanation of this form could look like the one below. In the explanation, *dimension D (x/y)* denotes that on dimensions D, the focus case has value x and the precedent value y.

> Outcome 'Admitted' is forced if: Compensates {University Rating (0.25/0.0), SOP (0.38/0.25), LOR (0.5/0.25), CGPA (0.27/0.14)}, can downplay Worse{GRE Score (0.1/0.24)}

Given this explanation, the user can decide whether the amount with which the student scored better on University Rating, SOP, LOR and CGPA offers sufficient compensation for the lower score on GRE.

The Mushroom case base provides us with less compact results, such as the following example (feature values are abbreviated for reasons of space):

> Outcome 'Poisonous' is forced if: pSubstitutes{spore-print-color: w, cap-surface: y, odor: y, ring-type: e, habitat: p, stalk-root: ?, bruises: f, stalk-surface-below-ring: k, stalk-color-above-ring: p, gill-color: b, ring-number: o, gill-size: n} & cCancels{ring-type: p, bruises: t, ring-number: t, stalk-color-above-ring: w, cap-shape: b, habitat: m, stalk-surface-below-ring: s, gill-size: b, odor: n}, can downplay: MissingPro{stalk-shape: e, gill-color: r, cap-surface: s, cap-color: p, spore-print-color: r, stalk-root: b} and cSubstitutes{ring-type: p, bruises: t, ring-number: t, stalk-color-above-ring: w, cap-shape: b, habitat: m, stalk-surface-below-ring: s, gill-size: b, odor: n} & pCancels{spore-print-color: w, cap-surface: y, odor: y, ring-type: e, habitat: p, stalk-root: ?, bruises: f, stalk-surface-below-ring: k, stalk-color-above-ring: p, gill-color: b, ring-number: o, gill-size: n}, can downplay: NewCon{cap-shape: x, cap-color: n, stalk-shape: t}

5.2.3 Playing devil's advocate

Even more interesting than explanations of correct predictions, may be to see what the explanation system tells us in case of an incorrect prediction. For this experiment, we used every case of the case bases again once as a focus case, but now we switched its outcome to the opposite - the incorrect outcome. As Figure 5.3 shows, the number of focus cases for which a trivial winning strategy exists is far lower when the outcome is incorrect. For a consistent case base, such as the Mushroom case base, this number is zero.

For the cases without a trivial winning strategy, we investigated whether there was a difference in the argument games when the outcome was incorrect compared to correct. Figure 5.4 shows what percentage of the total number of differences that were pointed at in a game consists of *defense features*: features that support the downplaying moves. A percentage of 50% would mean that on average every feature in a distinguishing attack by the opponent, can be downplayed by one defensive feature.

As the bar-plot shows, the relative number of defense features is lower for the incorrect outcomes for all three case bases. For the Churn case base this difference

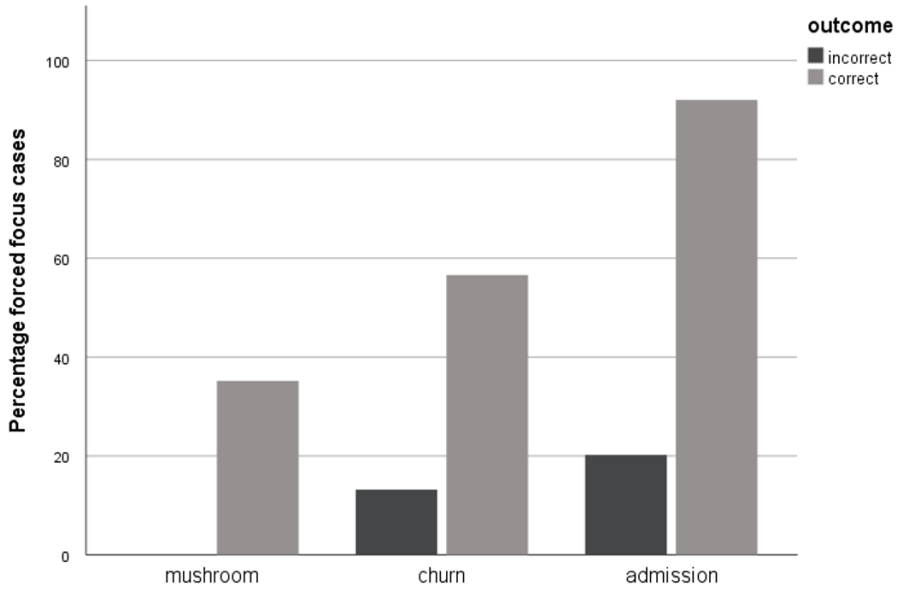

Figure 5.3: *Percentage of focus cases with a trivial winning strategy per data set for correct and incorrect outcomes*

is less than 10%, while the percentage of defense features decreases for the other two case bases with more than 50%.

5.2.4 Adding top-level information

An interesting property of AF-CBA is that it allows the addition of top-level information to the explanation system. The system can incorporate this information in the downplaying moves.

An eye for an eye

A simple rule that we could add is that all features are of equal importance. This would mean that for every feature that causes a relevant difference, one compensating feature needs to be presented to downplay the attack.

As in the previous paragraph, we wanted to measure how the system handled correct, as well as incorrect inputs. Therefore, we generated one argument game

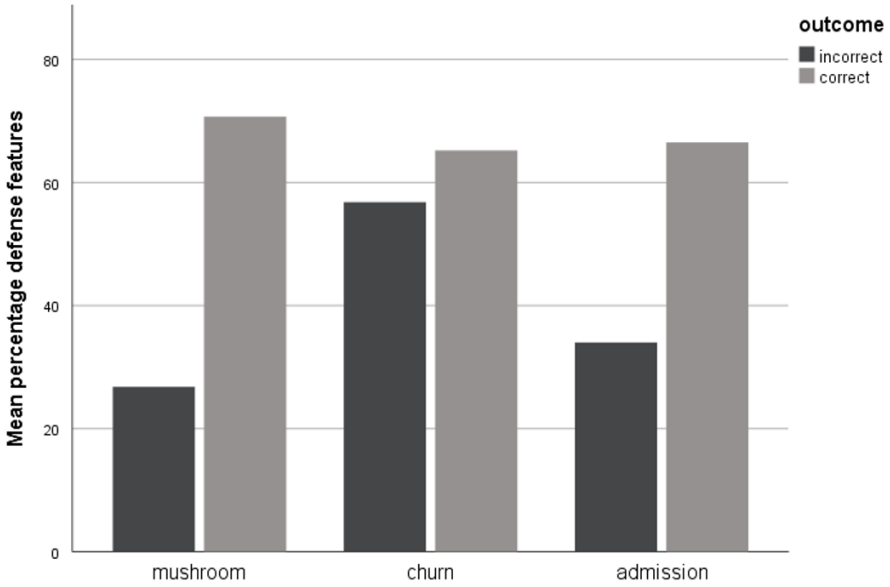

Figure 5.4: *Average percentage defense features of all differences in a game for correct and incorrect outcomes*

for the actual outcome, and one for the opposite (incorrect) outcome of every focus case. For an argument game, a precedent was randomly selected from the selection of best precedents.

For this experiment, we did not allow for empty downplaying moves - an eye must be compensated by an eye - and only considered a downplaying move successful if it contained at least the number of compensating features as the attack. Figure 5.5 shows the percentage of outcomes that was considered justified in this experiment, meaning the cited precedent could be defended against all attacks and was thus part of the grounded extension.

For all case bases, more than 80% of the correct outcomes was considered justified with this eye-for-an-eye rule. Incorrect outcomes were accepted less frequently, though for the Admission and Churn case base still more than 30% of the time.

Differences in the importance of features seem to form a problem for this approach. A feature that has a negligible positive correlation with the outcome is treated in the same way as a pro-feature that correlates strongly.

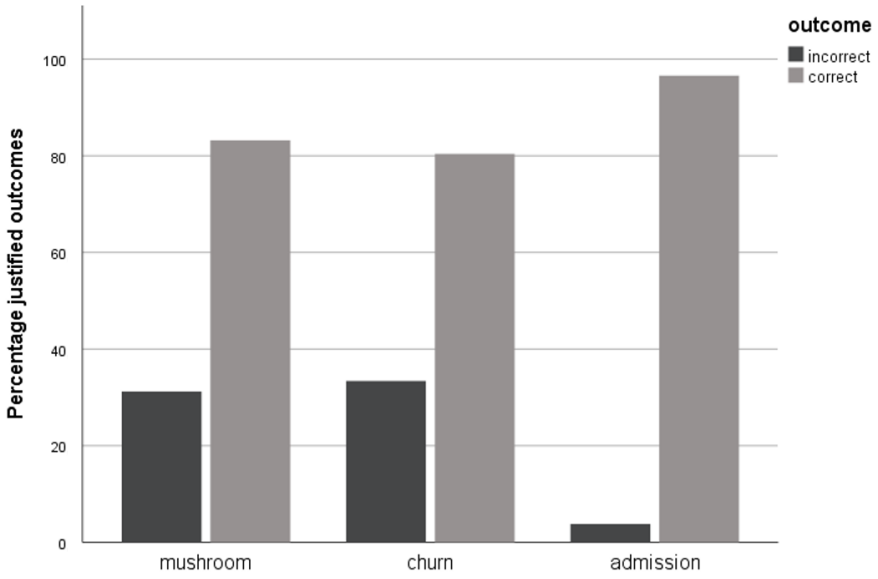

Figure 5.5: *Percentage justified outcomes for the correct and incorrect outcomes using the one-one rule*

Correlation game

In an attempt to address this problem, we set up a new experiment making use of some sort of correlation game. The set up was identical to the previous experiment, except the measurement of successful downplay. The success of a downplaying move was this time dependent on whether the collective impact of its features - measured as their summed absolute correlation - was at least equal to the collective impact of the features used in the distinguishing move. If the collective impact appeared to be less, the downplaying move was considered invalid, making the outcome of the focus case to be considered unjustified.

As Figure 5.6 shows, the correlation game did not result in a substantial improvement compared to the eye-for-an-eye approach. The percentages of justified outcomes are nearly identical.

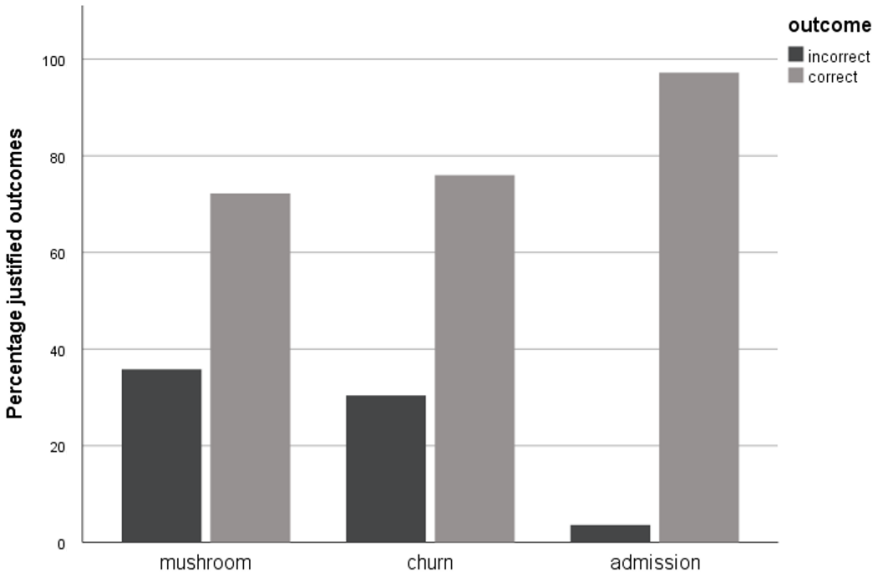

Figure 5.6: *Percentage justified outcomes for the correct and incorrect outcomes using the correlation game*

5.2.5 Using actual predictions

In the experiments so far, we have considered the output of AF-CBA when presented with only correct or only incorrect predictions as input. In an actual application, however, the system would receive the predictions of another machine-learning model as input. It would be interesting to see whether the system responds differently to the specific instances that another model predicts incorrectly. In this final experiment, we will compare the percentage of focus cases for which a trivial winning strategy exists given correct and incorrect predictions of another classifier.

We first made a selection of classifiers to test which models perform best on our data sets. We selected three classifiers which are currently most popular in practical applications: DecisionTree, Support Vector Machine and Naive Bayes (Das and Behera 2017). We also added a popular meta-algorithm used in combination with a DecisionTree, named AdaBoost. Finally, we added a *white-box* classifier, in the form of a Logistic Regression. We used built-in classifiers from the Python *sklearn* library. Details about the models can be found in Appendix B.

Per data set, every model was trained on a random selection of 80% of all data and tested on the remaining 20%. We measured the accuracy of a model by dividing the number of correct predictions by the total number of predictions. To improve the reliability of the results, averages over ten runs on random selections of the data were used.

The analysis made clear that there exist multiple models that reach 100% prediction accuracy on the Mushroom data set. Testing AF-CBA with the inputs of such as model, would result in the exact same experiment we conducted before (Table 5.3). We, therefore, only continued this experiment with the Churn and Admission data set. On the Churn data set, AdaBoost performed with an accuracy of 80.2% best. The logistic regression model was most accurate on the Admission data set, prediction 92.9% of the instances correctly. The performances of all models can be found in Table 7.2 in Chapter 7.

For this experiment, we used - again - 500 instances per data set. As we wanted to measure the predictions of the classifiers on unseen data, we applied 5-fold cross-validation. This means that we split the set into five test sets of 100 instances. Per set, the 400 remaining instances constituted the case base and were used for training the classifier. In this way, we measured the predictions of the classifier and corresponding output of AF-CBA for all 500 instances.

Figure 5.7 shows the percentages of focus cases with a trivial winning strategy for correct and incorrect predictions. As we can see, this percentage is substantially higher for correct predictions on both data sets.

When we compare Figure 5.7 with Figure 5.3, we see that the percentage of trivial winning strategies is higher for incorrect predictions than for incorrect outcomes. This could be explained by the fact that an instance for which the outcome is difficult to predict for another classifier, will likely be more challenging to judge correctly for AF-CBA as well.

5.3 Discussion

Implementing the approach of Prakken (2020) offered a couple of interesting results. First of all, we found that for a substantial part of the focus cases, a trivial winning strategy existed for the correct outcome. When there was no trivial winning strategy available, almost every case could still be defended with

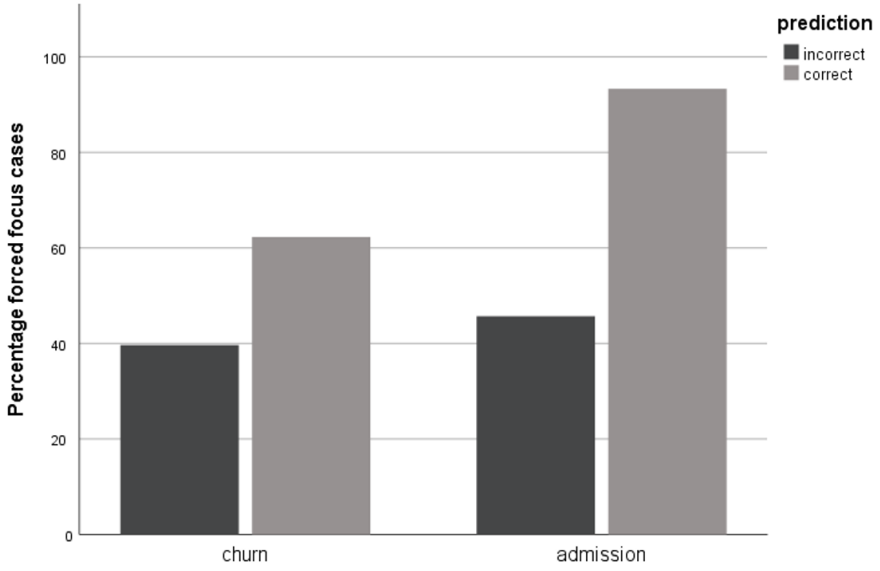

Figure 5.7: *Percentage of focus cases with a trivial winning strategy per data set for correct and incorrect predictions of the classifier. Adaboost was used for the churn predictions, a logistic regression for the admission predictions.*

a non-empty downplaying move. The system was clearly less positive in case of incorrect outcomes. For the incorrect outcomes, a far lower number of trivial winning strategies existed, and the relative number of defense features was lower. For the Churn and Admission set, we saw that the percentage of trivial winning strategies also appeared to be lower for incorrect predictions. Although lower, AF-CBA could still find a trivial winning strategy for about 40% of all incorrect predictions.

The 'raw presentation' of the outcomes of the explanation system in the Admission case base seemed comprehensible for a human user. With many more features, however, such as in the Mushroom case base, this was no longer the case.

After adding a simple rule as top-level knowledge, the system considered correct outcomes more often justified than incorrect outcomes. There seemed to be no relevant difference between using the eye-for-an-eye or correlation rule. Although the system clearly showed sensitivity for the correctness of the outcomes, the percentages still leave room for improvement.

In the following sections, five technical limitations of the current approach are discussed. In the final section, we will discuss a more fundamental challenge for AF-CBA.

5.3.1 Best and better precedents

The definition of a best precedent defined by Prakken (2020) allows many cases to qualify (Table 5.3). The simplest solution to pick one of those for the argument game would be to use random selection, as was done in some of the experiments. However, there might be differences between the best precedents which make certain citations to be preferred above others. This can especially be relevant when there is no trivial winning strategy. When there is a trivial winning strategy, all of the best precedents have no relevant differences with the focus case at all, leaving the opponent without relevant possibilities to attack the citation. For those focus cases, it does not make a difference for the structure of the argument tree which of the best precedents is selected.

As became clear (Table 5.4), a substantial part of the focus cases with non-trivial strategies has a selection of best precedents of which some are in need of empty downplaying moves in the argument game, while others are not. This gives the impression that it could be valuable to sharpen the definition of a 'best precedent'.

5.3.2 Separation of factors and dimensions

Prakken (2020) makes a distinction between factors and dimensions. Factors are defined as features with boolean values, such as yes/no or male/female. Dimensions are all features that can take more than two values, such as age. He acknowledges that, at first sight, it seems that factors are simply a special case of dimensions with only two values 0 and 1 where $0 < s$ 1 and $1 < s'$ 0.

After that, he rejects this idea, arguing that we do not always want to regard one value of a factor as *pro-s*, and the other as *con-s*. As an example, he uses the factor bribed. Suppose that the presence of this factor in a lawsuit would mean that the defendant made others do something for him or her by giving them money. The presence of this factor (represented as outcome 1), would be a pro-factor for the plaintiff. However, the absence of the factor bribed (outcome

0), does not seem to count as a factor con the plaintiff; it seems neutral with respect to that outcome. Prakken (2020), therefore, concludes to treat factors and dimensions differently.

This 'neutral state' of a value does not seem unique to factors. Suppose we would have the feature bribed, but this time as a dimension representing the amount of money that was paid to bribe others. In that case, a high amount would clearly favor the plaintiff, whereas an amount of 0 seems neutral with respect to that outcome.

Both of these scenarios, however, do not seem to pose problems for the explanation system given the way Horty (2019) treats dimensions. Using his strength ordering, we do not need to indicate whether a value favors a particular side. All we need to know is how the impact of the value relates to that of the other possible values. This is clear in both examples; the higher the number, the stronger the case for the plaintiff.

Separating factors from dimensions, on the other hand, does seem to have undesired consequences. In the argument game, factors and dimensions cannot help each other by defending against distinguishing moves. The effect of this is very clear in the Admission case base, which exists of six dimensions and a single factor. Whenever the factor is used in a distinguishing move, the proponent needs to answer with an empty downplay move, independent of how well the dimension values could compensate for this. An example of such a situation is shown below.

> Outcome 'Not admitted' is forced if: cSubstitutes{} & pCancels{}, can downplay: NewCon{Research}

In such scenarios, the distinction between factors and dimensions seems problematic. A user would like to know whether any of the other features can compensate for the difference, independent of whether that compensation comes from factors or dimensional features.

5.3.3 Added value of the counterexample

A citation of a precedent can not only be attacked by pointing out any differences, but also by citing a counterexample (Prakken 2020). A counterexample is a case from the case base with the opposite outcome of the focus case and precedent. This seems an interesting addition, as humans tend to prefer contrastive explanations (Miller 2019); explanations that tell you not just why P, but why P instead of Q.

In AF-CBA, the proponent can attack a counterexample by transforming the precedent into a case with no relevant differences with the focus case using downplaying moves. This handling of counterexamples, unfortunately, does not give the user information about how the precedent and counterexample relate to each other, nor how the counterexample relates to the focus case. Moreover, the transformation into a case with no relevant differences using the downplaying moves does not add information that was not in the game already. All relevant differences are pointed out by the MissingPro, NewCon and Worse attacks, after which the downplaying moves show whether the precedent can be defended or not. The questions whether the distinguishing moves can be downplayed and whether the counterexample can be downplayed, will always receive the same answer. This gives the impression that counterexamples should be applied differently.

5.3.4 Feature overload

As stated by Molnar (2019), the interpretability of example-based methods stands or falls by the comprehensibility of a single instance in the data set. To enable users to comprehend an instance, the features must not only be meaningful but also limited in number.

In the ‘raw’ example outputs, we observed a substantial difference in comprehensibility between the output for the admission set - consisting of just seven features - and the mushroom set, which is rich in features. Possibly the implementation could be improved by using a partial ordering on the categorical features, instead of transforming them into binary ones. However, also in that scenario, every case consists of 22 features. It seems necessary to look for possibilities to decrease the number of features used in the method or presented to the user.

5.3.5 Inconsistencies

When transforming the data sets into case bases, we found that both the Churn- and the Admission data set were inconsistent for our implementation. This means that while using the case base, focus cases exist for which AF-CBA would find a trivial winning strategy for both outcomes.

This seems to be problematic. The existence of a trivial winning strategy is the highest form of confidence in the prediction the system can express. A system that confidently supports prediction p, as well as $\overline{p}$, will and should not constitute much trust.

A possible solution could be to restrict the usage of AF-CBA to consistent data sets. However, for our only consistent case base - the Mushroom set - we found that classifiers could learn to predict instances with 100 % accuracy. This gives the impression that the class of data sets that are relevant for our purposes but also consistent may be limited. Therefore, we will suggest a different solution in the next chapter.

5.3.6 Explaining or comparing?

Apart from the technical challenges that this concrete implementation of AF-CBA faces, there seems to be a more fundamental challenge as well. Instead of explaining how a black-box model comes to a prediction, AF-CBA explains whether a different, more interpretable model would justify the prediction made. As a result, the explanation can fully deviate from the way the prediction model works; it is almost like we are comparing the predictions of two different models.

As this type of explanation system is more concerned with justifying the black box than with explaining it, we will from now call this approach a *justification system*. There seem to be a couple of disadvantages to applying justification systems. First of all, the construction we obtain feels artificial and may confuse end users. A model we do not know about is in the lead, while some other model is explaining its own reasoning - who to trust? Many of the reasons why we want an explanation - such as building trust, learning from the prediction model and debugging - may continue to exist in this construction.

Moreover, the explanation system *must* be worse than the prediction model.

Otherwise - as Rudin (2019) lined out - the explanation system would make the prediction model redundant. Given that the explanation system is worse, the system might distract the user from following correct predictions of the black-box.

Although the system does not explain the inner workings of the black box, it does present extra information to the user, which may lead to new insights. For example, when 'grey zone' decisions need to be made, a user could consider the information from the argumentation model to make a better-informed decision. Besides, we cannot claim that the explanation system is just a separate classifier, as AF-CBA takes the prediction of the black box as the starting point.

All in all, we can conclude that the added value of the current justification is not self-evident and needs further experimentation. In the next chapter, we will investigate whether we can improve the justification system by searching for solutions to the technical limitations found. In addition, we will explore two new directions which diverge from the justification approach taken in AF-CBA.

6. A new argument framework

In this chapter, we will build on Prakken (2020) and try to come up with solutions for the limitations found. In the first three sections, we will suggest adjustments to the original justification system and propose a new abstract framework of argumentation. The changes in approach will be motivated per topic. After that, we will investigate two new directions that aim to close the gap between the CBA-system and the prediction model.

6.1 Combining factors and dimensions

In the previous chapter, we argued that a sharp distinction between factors and dimensions is problematic. When factors and dimensions play in 'different leagues' within the argument game, it is impossible to compensate a worse dimension with a factor, and visa versa. In this proposal, we eliminate this difference and treat factors and dimensions equally. All value assignments are still represented as pairs of the form (d, v), where d is the dimension and v the value for that instance. In case of a factor - a binary feature - v can only take two possible values: 0 (absence) and 1 (presence).

As the running example, we will reuse part of our Churn scenario. In this scenario, we have a small data set with information about three previous customers (Table 6.1).

We also assume knowledge of the tendencies of the features to promote outcomes. We define a dimension as a pair $d(f,t)$, where f is the name of the feature and $t \in \{o, \overline{o}\}$ its tendency. When t is equal to outcome o, d is called a $pro-o$ dimension: the higher the value on feature d, the more outcome o is

customer	gift	high-cost	present	website	churn
Miss Hill	1	0	0	5	0
Mister Nash	1	1	1	3	1
Mister Wang	0	0	1	6	0

Table 6.1: *Churn example data set*

promoted. As before, we assume that *gift*, *present* and *website* are pro-Stay; *High-cost* is assumed to be pro-Churn.

We transform every instance of the training data into a *case*, a pair (F, o), where F denotes the fact situation and o the outcome. We can represent the case for Miss Hill as:

$Hill$ = ([(gift, 1), (high-cost, 0), (present, 0), (website, 5)], Stay)

To refer to elements of cases, we will introduce some extra notation. We say $Outcome_C$ to denote the outcome of case C. For case C, we define Pro_C as a list of all feature names f for dimensions (f,t) in which $Outcome_C$ is equal to t. Con_C denotes the list of all feature names f for dimensions (f,t) in which $Outcome_C$ is unequal to t. For dimension d and case C, $\text{Value}_C(d)$ refers to the value of C for d.

Suppose that we have some new customer, Miss Jale (Table 6.2). Miss Jale is our *focus case*: we do not know her decision yet, and so she is the focus of our prediction task.

customer	gift	high-cost	present	website	churn
Miss Jale	0	0	1	3	?

Table 6.2: *Churn example focus customer*

As we do not know the outcome yet, we represent the focus case simply as a fact situation:

$Jale$ = [(gift, 0), (high-cost, 0), (present, 1), (website, 3)].

To reason about the outcome of the focus case, we need a way to compare the focus case to other cases. For this purpose, we define two types of differences between the focus case and the precedents: negative differences (ND) and positive

differences (PD). ND are equal to the relevant differences in AF-CBA and represent differences which make the focus case less likely to imitate the outcome of the precedent.

Definition 4. [Negative differences (ND)] Let F be a focus case and P a precedent. The set $ND(F,P)$ of negative differences between F and P is defined as follows.

$ND(F,P) = \{f \in Pro_P \,|\, Value_P(f) > Value_F(f)\} \cup \{f \in Con_P \,|\, Value_P(f) < Value_F(f)\}\}$.

The negative differences between our focus case *Jale* and $Hill$, $ND(Jale, Hill)$, are: {*gift, website*}. Both are features in Pro_{Hill} for which the value of $Hill$ is higher than that of *Jale* - making *Jale* less likely to imitate $Hill$s decision to stay.

PD are the opposite differences and describe the differences which make the focus case stronger for the outcome.

Definition 5. [Positive differences (PD)] Let F and P be two cases. The set $PD(F,P)$ of positive differences between F and P is defined as follows.

$PD(F,P) = \{\{f \in Pro_P \,|\, Value_P(f) < Value_F(f)\} \cup \{f \in Con_P \,|\, Value_P(f) > Value_F(f)\}$.

There is one positive difference between *Jale* and $Hill$: *present*. $Hill$ was not present at the latest customer event, while *Jale* was. We call this a positive difference as it makes *Jale* more likely to imitate the decision of $Hill$ to stay.

The forms of attack towards the citation of a precedent, as shown in Figure 6.1, will be similar to but simpler than those of Prakken (2020). As we no longer apply different rules for factors, two of the possible distinguishing moves - MissingPro and NewCon - no longer apply.

A precedent can still be attacked by a *Worse* move, pointing out any negative differences. The proponent can defend against this attack by playing a *Compensates* move, stating that the positive differences can compensate for the negative differences. As in Prakken (2020), a Compensates move may be empty, while Worse has to include at least a single feature.

So far, our approach can be seen as a special case of AF-CBA in which we apply factors as dimensions. However, in AF-CBA, counterexamples can also attack precedents. Counterexamples will play a different role in our argument game, as will be explained in the next section,

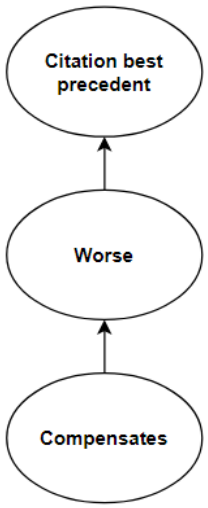

Figure 6.1: *Structure of the attack and defense of the citation in the argument game*

6.2 Promoted counterexamples and winning precedents

In the previous section, we omitted the counterexample as a possible attack on the precedent. In this proposal, counterexamples will have a different position within the argumentation framework. Instead of treating them as abstract entities, we will enable the properties of the counterexamples to influence their roles in the argument framework. To achieve this, we promote the counterexamples; they will be treated equally to the precedents. Any case - independent of whether its outcome matches the prediction of the focus case - can be used as the first citation of the proponent.

We now have as many precedents as the case-base size and need a method to find the best, or *winning*, examples. An intuitive approach is to select the cases that are most similar to the focus case for this purpose. In machine learning, this approach is known as a *k-nearest neighbors model*. Such a model bases a new prediction on the k nearest neighbors of the case in the feature space.

Interestingly, the selection of precedents in Prakken (2020) can be seen as a variation on the nearest neighbors algorithm in which a distinction is made between *relevant differences* (negative differences) and other differences (positive dif-

ferences). His definition of a best precedent attempts to minimize the distance in feature space for negative differences while ignoring the feature differences that make the new case even more likely to obtain the precedent's outcome. This refined notion of distance from Prakken (2020) seems promising; it enables us to find examples that would be overlooked in a nearest neighbors approach - they would be too far away in absolute feature space.

As was shown in the previous chapter, important differences can exist between cases within the selection of best precedents as defined in Prakken (2020). His definition allows every case with a minimal set of relevant differences with the focus case to be selected as precedent. We want to sharpen this definition in such a way that only the best examples from these precedents remain.

It seems difficult to improve the definition without having extra information about the importance of features. We could, for example, choose the best precedent with the minimal number of differences with the focus case. Although this might be better than random selection, it is not completely satisfying. There can be multiple cases with the same number of differences that are not equally strong precedents. Besides, a lower number of differences does not guarantee the case to be a better example; two small differences can be less important than a single big difference.

We, therefore, suggest an approach which takes an estimation of the importance of every feature into consideration. The method that is used for these estimations is unspecified; expert opinions, automatic feature importance extractions or other methods can be used. The only requirement is that the estimations assign to every feature a real, non-negative number - the weight (w) - which meets the following two conditions:

1. the importance of a feature f for instance i can be calculated as:

$$Importance_i(f) = n(Value_i(f)) * w(f) \tag{6.1}$$

   where $n(Value_i(f))$ represents the normalized value - such that all feature values fall between 0 and 1 - of x for feature f.

2. if $Importance_i(x)$ equals $Importance_i(y) * n$, then x is estimated to be n times as important as y

Using this information, we will quantify the differences between cases. The importance of a difference on feature f between case i and j can be calculated

as:

$$Importance_{D(i,j)}(f) = abs(Importance_i(f) - Importance_j(f)) \tag{6.2}$$

In other words, the importance of a difference is calculated as the absolute difference in the importance of the feature values for those cases. By summing up these values of importance, we can calculate for every precedent the total importance of the negative- and the positive differences with the focus case.

$$Importance_{ND(i,j)} = \sum_{i=1}^{length(ND)} importance_{D(i,j)}(ND_i) \tag{6.3}$$

$$Importance_{PD(i,j)} = \sum_{i=1}^{length(PD)} importance_{D(i,j)}(PD_i) \tag{6.4}$$

This results in one value representing the strength of the Worse attack, and one value representing its compensating strength.

Coming back to our example, we assume the four features to have the following weights (Table 6.3):

gift	high-cost	present	website
0.2	0.9	0.7	0.15

Table 6.3: *Weights of the features*

To calculate the difference in feature importance between $Jale$ and $Hill$, we first have to normalize their feature values. We determine the minimum (MIN) and maximum (MAX) value for that feature from all cases in the case base. We then calculate the normalized value as:

$$n(v) = \frac{v - MIN}{MAX - MIN} \tag{6.5}$$

The results of this calculation are shown in the first two rows of Table 6.4.

Now we can calculate the importance of the feature values for $Jale$ and $Hill$ by multiplying the normalized values with the corresponding weights. As a final

step, we take the absolute difference between the values of $Jale$ and $Hill$ to obtain the importance of their differences. Table 6.4 shows the results of these calculations.

category	gift	high-cost	present	website
$n(Value_{Jale})$	0	0	1	0
$n(Value_{Hill})$	1	0	0	0.75
$Importance_{Jale}$	0	0	0.7	0
$Importance_{Hill}$	0.2	0	0	0.11
$Importance_{D(Jale,Hill)}$	0.2	0	0.7	0.11

Table 6.4: *Normalized feature values, feature importance and difference importance for Jale and Hill*

We define the *balance* of a Compensates attack to be the importance of the Worse attack subtracted from the importance of the Compensates attack. The higher the value of this balance, the better the precedent can compensate for the negative differences with positive ones. In the case of Jale and Hill, the $Importance_{ND(Jale,Hill)}$ is 0.2 + 0.11 = 0.31. As the importance of the positive differences, $Importance_{PD(Jale,Hill)}$, is 0.7, the balance of Hills compensating move is positive: 0.7 - 0.31 = 0.39. We say that compensating move A is *stronger* than compensating move B, when the balance value of A is higher than that of B.

With these new definitions, we can set up the argument framework in such a way that it can distinguish between the precedents. More specifically, we want the framework to make a selection of *winning precedents*, which consists of all precedents that are part of the grounded extension.

First, we add the following rule to our argument framework - enabling us to distinguish between compensating moves:

> A attacks B if A and B are compensating moves and A is a stronger compensating move than B

This rule causes the precedents with weaker compensating moves to be excluded from the grounded extension. Note that the application of this rule is not restricted to compensating moves that defend the same precedent, a compensating

move defending precedent P, can attack any weaker compensating move defending precedent P'.

There may also be cases that do not even need a compensating move because they lack any negative differences. Then we can apply the a fortori rule of Horty (2011). To prioritize these cases, we also add the following rule:

> A attacks B if A is a precedent with no negative differences with the focus case and B is a compensating move

This rule causes precedents that don't have any negative differences to eliminate all precedents that have such differences, by attacking their compensating defenses.

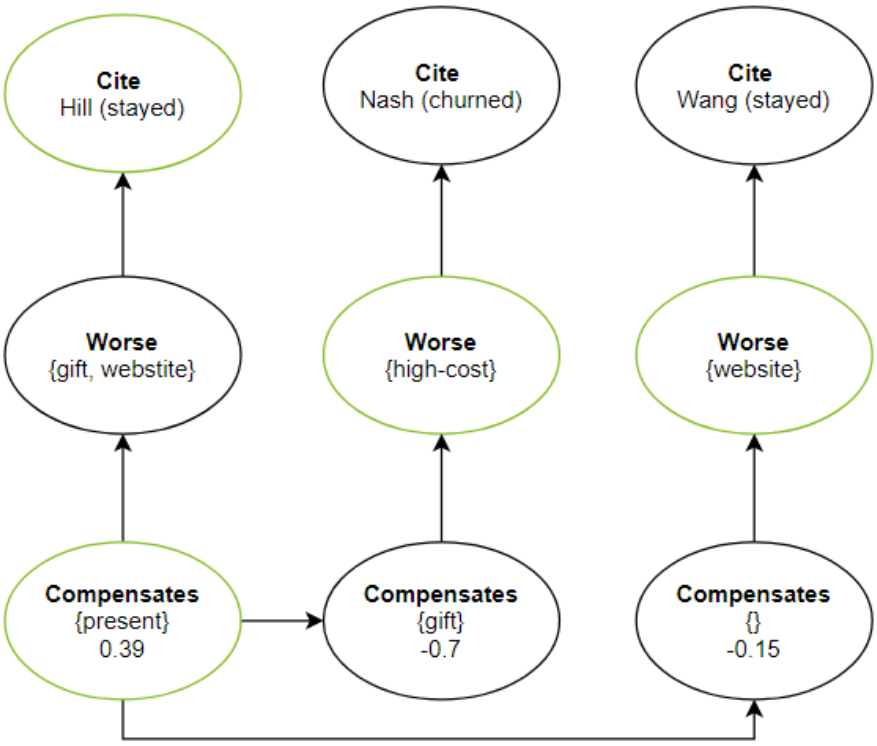

Figure 6.2: *Final argument framework of the Churn example. Green arguments are part of the grounded extension.*

Applying these rules to the example, we obtain the argument framework presented in Figure 6.2. *Hill* is the only winning precedent as her compensating move is stronger than those of the other two customers. Given a finite case base, there must always be at least one winning precedent.

Proposition 2. Let *CB* be a non-empty, finite case base and *F* a focus case. Then the set of winning precedents cannot be empty.

Proof. Suppose that there are no winning precedents. Then all precedents must be attacked by arguments that are in the grounded extension. The only argument that can attack a precedent is a Worse move, so all precedents must be attacked by a Worse move. As a Compensates move may be empty, it can always attack a Worse move. Thus, every Worse move will be attacked by a Compensates move, defending the precedent. These Compensates moves must all be attacked; otherwise, the precedents will be in the grounded extension, which is in contradiction with the supposition that there are no winning precedents. There are two ways in which a Compensates move can be attacked:

1. A Compensates move can be attacked by a stronger Compensates move
2. A Compensates move can be attacked by a precedent with no negative differences with the focus case

With attack 1) the Compensates move that attacks another Compensates move, must be strictly stronger. This means that there must be at least one Compensates move that is not attacked by any other Compensates moves, as it is the strongest. When a Compensates move is unattacked, the corresponding precedent must be part of the grounded extension as well, because the only argument attacking it (Worse) is successfully defeated. This would mean that the set of winning precedents is not empty. In case of attack 2) the precedent that attacks the Compensates move must be a winning precedent; it has no negative differences so is not attacked by any arguments. Both scenarios contradict the supposition that there is no winning precedent, so we conclude that the set of winning precedents cannot be empty for finite case bases. □

When all winning precedents have the same outcome, this outcome will be the justified prediction for the focus case. Later, we will explain what the system will do when there are multiple winning precedents with different outcomes.

6.3 Dealing with inconsistencies

Suppose that we add a new customer to the example case base, Mister Trent (Table 6.5).

customer	gift	high-cost	present	website	churn
Mister Trent	1	0	1	7	1

Table 6.5: *Churn example extra customer*

Mister Trent received a gift, was present on the latest customer event and is not part of a high-cost category. He also logged in to the website seven times. Based on our information about the tendencies of features, Trent seems a clear candidate to stay. However, Trent decided to churn. As Figure 5.1 from the previous chapter shows, divergent customers such as Mister Trent also exist in the actual Churn Telecom data set.

Cases that received a surprising outcome, like customer Trent, can be problematic for our approach. Even though Trent decided to churn, we don't want our model to predict for similar customers that they will churn; we have all reason to believe a similar customer would stay. Unfortunately, these surprising cases tend to have a big impact on the argument game. Suppose that we would generate an argument game for *Jale* after *Trent* is added to the case base. We can see that all differences between *Trent* and *Jale* are positive; they make *Jale* even more likely to churn. *Trent* would be the only winning precedent, attacking all compensating moves.

To deal with this problem, we transform our case base into a *consistent* one. As defined in previous chapters, we call a case base consistent if and only if there are no cases X and Y with $outcome(X) = s$ and $outcome(Y) = \bar{s}$, such that $X < sY$; Y is at least as good for outcome s as X.

In our example, the addition of Mister Trent causes inconsistency in the case base. Both *Hill* and *Wang* stayed, while *Trent*, who churned, has a profile that is at least as good as theirs for the outcome Stay. To achieve a consistent case base, we iteratively remove the most inconsistent cases until we are left with a consistent case base. The inconsistency of a case is measured as the number of times the case can be used to show an inconsistency with another case. For *Trent* this would be the largest number (2), so his case would be removed from the case base. The process ends there, as we are left with a consistent case base.

The algorithm we use to check the consistency and count the number of inconsistencies per case is shown in pseudo code below:

```
consistent = True
inconsistent_dict = { }
for case in CB do
    for other_case in CB with outcome(case) ≠ outcome(other_case) do
        if other_case is at least as good for outcome(case) as case then
            consistent = False
            inconsistent_dict[other_case] += 1
        end if
    end for
end for
return consistent, inconsistent_dict
```

The function returns a binary variable 'consistent' stating whether the current case base is consistent. As long as this variable is False, we iteratively remove the most inconsistent case from the inconsistent_dictionary. Note that this algorithm is not deterministic; when multiple cases have the same number of inconsistencies, one of those cases is randomly chosen to be removed.

Now that we have introduced the final concept of the Argument Game, we can give a formal account of the argumentation framework.

Definition 6. Given a finite, consistent case base CB and a focus case $f \notin CB$, our abstract argumentation framework AAF is a pair $\langle A, attack \rangle$ where:

- $A = CB \cup M$, where $M =$
  $\{Worse(c,x) \mid x \neq \emptyset \text{ and } x = ND(c,f)\} \cup$
  $\{Compensates(c,y,x,b) \mid y = PD(c,f),\ x = ND(c,f) \text{ and } b = Balance(c,f)\}$
- A attacks B if and only if:
  - $B \in CB$ and A is of the form $Worse(B,x)$, or
  - B is of the form $Worse(c,x)$ and A is of the form $Compensates(c,y,x,b)$, or
  - A is of the form $Compensates(c,y,x,b)$ and B is of the form $Compensates(c',y',x',b')$ and $b > b'$, or
  - $A \in CB$ and B is of the form $Compensates(c,y,x,b)$ and A has no negative differences with the focus case: $ND(A,f) = \emptyset$.

6.4 Feature selection

Large numbers of features form another challenge for the approach. In our running example, we considered only four features, making the comparisons comprehensible for humans. For real machine learning applications, the number of features in the data can be ten, or even a hundred times as much. Problems that absolutely need large numbers of features seem unsuitable for our approach. For other problems, we might be able to reduce the number of features to an acceptable level.

In this work, we will use an automatic approach to feature selection. The idea of this approach is to eliminate as many features as we can without deteriorating the predictive model. We put this into practice by removing one by one the features that have the least importance, measured by the absolute weights of a logistic regression. We continue this process until the accuracy of the prediction model starts decreasing. The process is illustrated in the pseudo code below:

```
old-accuracy, new-accuracy = 0, 0
while old accuracy <= new accuracy do
    old-accuracy = new-accuracy
    new-accuracy = the model's current accuracy
    if new-accuracy ≥ old-accuracy then
        best features = current features
    end if
    measure the importance of the current features
    select the feature f with the least importance
    remove feature f from the data set
end while
return best features
```

In the algorithm above the elimination of features stops as soon as the accuracy decreases. In certain applications, increased comprehensibility of the cases might outweigh a small drop in accuracy. The designers of the system could then play around with the number of features and prediction accuracy to find the best balance.

6.5 Presenting justifications

In this section, we will discuss how the formal framework can be used to present justifications to a user. Concretely, we will design an interface in which the system justifies predictions for the Churn data set. We apply this interface in a user experiment, which is the topic of Chapter 8.

The generated argumentation framework for a focus case represents an extensive overview of the interaction between all cases in the case base. This large framework could be applied to enable the user to cite an example case by itself, creating an interactive system in which the user can ask contrastive questions, such as 'Why did customer C receive outcome o, whereas customer C' received outcome o'?'. We will leave this to future work and continue with a simpler presentation. Aiming to disclose the most relevant information to the user, we will only allow cases to be presented when they are *best performing* precedents for that outcome in the game. Given a precedent $P = (F, o)$, P is a best performing precedent for F if it would be a winning precedent in an argument game for F in which only precedents with outcome o are allowed. In other words, P is the strongest precedent for its outcome.

Using a case base consisting of only best performing precedents, we can generate a grounded game in a similar fashion as AF-CBA does. The proponent begins by citing a precedent P that received the same outcome as the prediction for focus case F. Then there are two possibilities: either the precedent P has no negative differences with the focus case. This would make it a trivial winning strategy; the opponent is directly out of moves. We call a precedent with no negative difference with the focus case *forced*.

Alternatively, the proponent can point out the negative differences between P and F by playing a Worse move. The proponent has one way of replying to this move: with a Compensates attack. This move is always available, as it is allowed to be empty. Now, there are two ways in which the opponent could attack the Compensates move. Either by citing a precedent with no negative differences with the focus case or by attacking the Compensates move with a stronger one. In both cases, the opponent wins the game, declaring the prediction to be unjustified. When both options are not available to the opponent, the citation of the proponent is justified. In that case, the counterexample is unjustified, unless its Compensates move is exactly as strong as that of P. The counterexample cannot both have no negative differences and be unable to attack P with the focus case,

as we work with a consistent case base.

The size of the difference between the best examples from both outcomes varies; sometimes it is a close call, and as we saw, there is even a scenario possible in which cases for both sides are winning precedents. To nuance the judgment of the justification system, we can make it non-binary. In our work, we used four output options of the system: convincing, somewhat convincing, somewhat unconvincing and unconvincing. Given two non-forced best-performing precedents *same* and *counter*, where *same* has an outcome equal to the prediction and *counter* unequal, we would say the prediction is somewhat convincing if either:

$$Balance_{same} = Balance_{counter}$$

or both:

a) $Balance_{same} > Balance_{counter}$
b) $Importance_{ND(same)} > Importance_{ND(counter)}$ or $Importance_{PD(same)} < Importance_{PD(counter)}$

In the last scenario, *same* has a greater balance but more negative differences or less positive differences with the focus case than *counter*. Symmetrically, the system would output 'somewhat unconvincing' when we replace a) with $Balance_{same} < Balance_{counter}$.

Together with one of the four judgments, we present a textual explanation to communicate the reasoning used by the system. These explanations are generated with templates. All templates can be found in Appendix A. Figure 6.3 shows what the output of the system looks like for customer Sanders.

Along with the specified output, we support the judgment by presenting the cases and comparison - the positive- and negative differences - using visualizations. The weights allow us to present the importance of the features by means of feature summary visualizations. Bars represent the direction and the strengths of the impact of a feature. The width of the bar of feature f for case c is established relative to $Importance_c(f)$. The color represents whether a higher value on this dimension favors the outcome Churn (red) or Stay (green). The original feature names were transformed into little phrases. Figure 6.4 shows what the presentation of Mister Sanders looks like.

Finally, we represent comparisons in similar fashion as cases, using small templates for the text. The bar color represents whether the difference makes the fo-

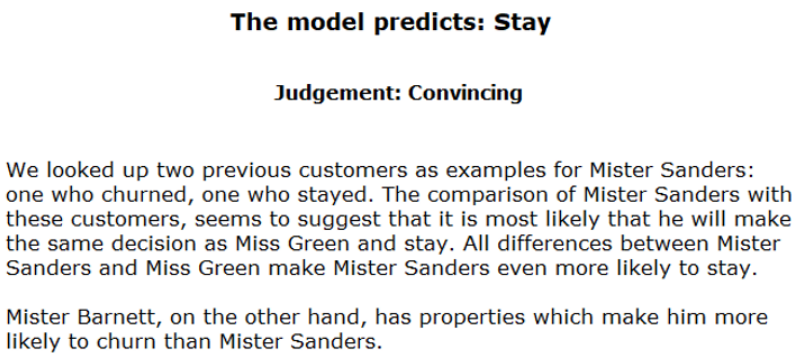

Figure 6.3: *Judgment and explanation for example customer Sanders*

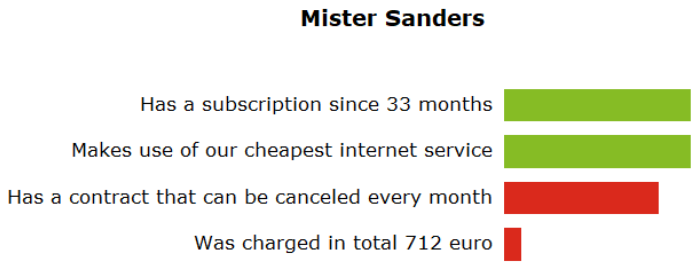

Figure 6.4: *Example presentation of a case*

cus case more likely to Churn (red) or Stay (green) than the precedent. The size for a difference between case c and p is established relative to $abs(Importance_c(f) - Importance_p(f))$. Figure 6.5 shows the comparison with the winning precedent for our example customer.

For reasons of space and in an attempt to prevent information overload, only one precedent together with its comparison is shown in the interface at the same time. The user can switch between the two precedents - *same* and *other* - using a button. The complete structure of the screen in the interface is shown in Figure 6.6.

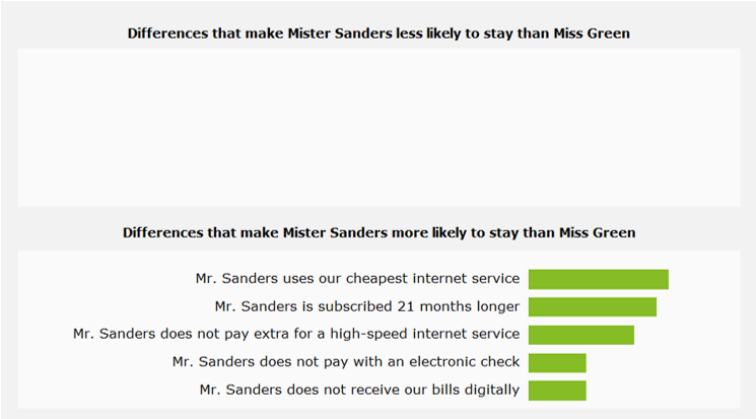

Figure 6.5: *Positive and negative differences between focus case Sanders and precedent Green. In this example, Sanders has only positive differences with Green*

6.6 Two alternative directions

In the previous chapter, we have discussed the disadvantages that come with the fact that the justification system operates - to a large extent - independently from the black-box model. In this section, we will propose two ideas that aim to close the gap between the predictive model and the explanation system. The cleanest solution would be to replace the black-box model by an interpretable CBA-system. The first section will discuss this option. In the second section, we will discuss a possibility inspired by the work of Weerts, Ipenburg, and Pechenizkiy (2019). In this approach, we hold on to the black box but make an attempt to monitor what is going on inside of it.

6.6.1 Classifying independently

Instead of using our proposed CBA-system to explain predictions of some machine-learning model, we can also apply the system as an independent classifier. Like a k-nearest neighbors algorithm, the method searches for examples in the training data. Instead of using some distance function to find nearest neighbors, our system creates an Abstract Argumentation Framework to select the most suitable, *winning*, precedents. In contrast to AF-CBA, our new proposal does not need a prediction from another model as input. It can independently

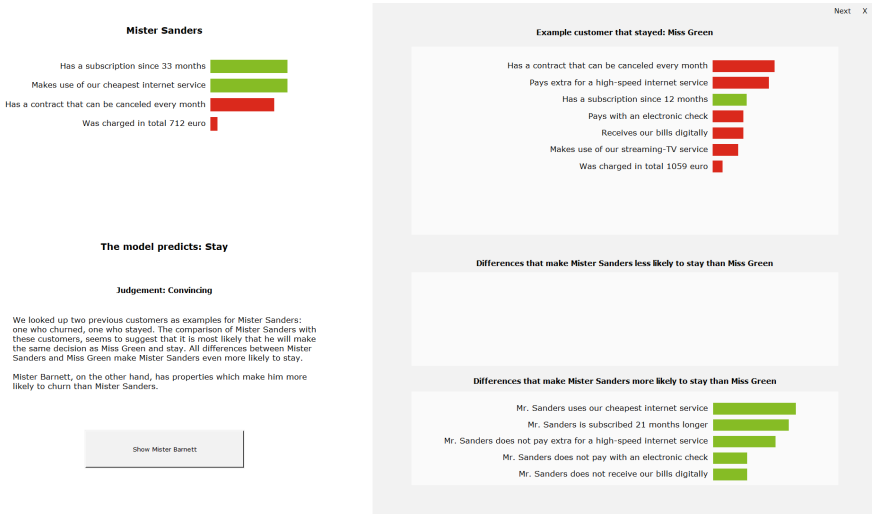

Figure 6.6: *User interface justification system. Top left, the focus case is presented. Below, the prediction of the black-box model and the output of the justification system. On the right side, the precedent and its comparison with the focus case are shown. The button in the left button corner enables the user to switch between the two precedents.*

make predictions of the outcome for new cases by selecting the most common outcome among the winning precedents.

When the explanation system takes over the role of the classifier, the explanations become faithful to what is actually going on in the classifier. This would be an ideal solution in terms of transparency but might affect the accuracy in a negative way. In the next chapter, we will investigate how well this explainable classifier can perform on the Churn, Mushroom and Admission data sets.

6.6.2 Monitoring the black box

According to our current state of knowledge, not every black box can be replaced by an interpretable system without diminishing the performance. In this section, we will discuss whether we can gain further insights into black boxes while still using a model-agnostic approach.

Prakken (2020) assumed two forms of access to the black box: access to its training data and access to predictions of the model, given input instances. Weerts,

Ipenburg, and Pechenizkiy (2019) interestingly combined these two; they collected the predictions of the black box for the instances in the training data. With this information, a new dimension can be added to our argumentation; we now know the outcomes of similar instances ánd whether the model predicted these outcomes correctly. This allows us to focus our reasoning on the question: 'Is the prediction reliable?' instead of 'Does our system justify the prediction?'

To display the outcomes and predictions for cases in the neighborhood of the focus case, Weerts, Ipenburg, and Pechenizkiy (2019) used a two dimensional scatter plot. In their plot, the distance between two cases roughly represents the distance in feature space according to some distance function. We adopted their approach to a large extent, and applied a two-dimensional scatter plot to visualize the neighborhood of the focus case. As our distance function, we used the importance of features weighted by the logistic regression. To transform the weighted feature values into two dimensions, a forced-directed graph was created, using the Python *forcelayout* library. This method applies a physical model of attraction and repulsion (Gibson, Faith, and Vickers 2013) to reduce dimensions. It aims to lay out a two-dimensional graph optimally, representing the actual spacing in multi-dimensional feature space.

We decided to include the five nearest neighbors in weighted feature space in the plot. These neighbors were selected from the complete case base. We consciously did not use the consistent case base, as this undermines the goal of the system to present the reliability of the model in the neighborhood. The number of cases to display was chosen arbitrarily, attempting to find a balance between completeness and selectivity. Experimentation with different numbers is left for further research.

We used colors to represent the outcome of the neighbors and the correctness of the model's prediction. The color of the border of the square represented whether the customer stayed (green) or churned (red). The color inside the square represented whether the outcome was correctly (white) or incorrectly (black) predicted by the black box. Figure 6.7 shows an example of the plot for customer Sanders.

To enable the user to compare the neighbors, we made the presentation interactive. The neighbours were presented as buttons. Pressing a button, enabled the user to view the details of the neighbor and its positive and negative differences with the focus case. This information, as well as the focus case and prediction of the model, were presented in the exact same way as in the justification system.

Figure 6.7: *Example monitor presentation for a customer from the Churn set*

The monitor system did not output an explicit argumentation, nor suggest a conclusion. It was left to the user to apply its own argumentation on the information. It would be interesting to investigate the possibilities of formalizing argumentation based on this information in future research.

We ended this chapter with three directions of applying CBA for the challenges we face regarding black boxes. In Chapter 7, we will further investigate the possibility of creating a CBA-classifier. Chapter 8 presents the results of a small user experiment we conducted, comparing the justification- and monitor method.

7. Evaluation of classifiers

In this chapter, we will investigate the potential of an explainable classifier, which builds on the concepts found in Prakken (2020). In addition, we aim to establish the impact of feature selection and filtering the case base. We will take a broad perspective on CBA and move away from the formal frameworks of argumentation we have discussed so far. In this way, we hope to test the suitability of the concepts applied in the argumentation frameworks.

For the experiments, the Mushroom, Churn and Admission data sets are reused. In the first section, we compare the classification performances of different variations on the algorithm presented in the previous chapter. These tests will be performed with a case base consisting of all the training data, as well as with a consistent case base. After that, we select the best performing algorithms and use those to investigate whether feature selection can help improve the outcomes.

7.1 Comparing algorithms

The algorithm introduced in the previous chapter adopts three concepts from the approach of Prakken (2020):

C1) A precedent without any negative differences is always preferred over a precedent with negative differences
C2) Negative differences make a precedent less suitable than positive differences
C3) Negative differences can be compensated by positive differences

Although these concepts apply intuitively within a law context, it is unclear whether these rules are suitable for reasoning about different kinds of machine-

learning problems. All three concepts rely strongly on the idea that we can make a successful distinction between negative and positive differences, which may not be the case for every data set.

To obtain a better understanding of their suitability, we will define variations on the algorithm in which we drop one or multiple of these concepts. After the name of every algorithm, we list which concepts are applied. Let us first summarize the original algorithm, which we will call *priority + balance*.

Algorithm 1) priority + balance (C1, C2 and C3)

1. If there are precedents with no negative differences with the focus case
   (a) Predict the most common outcome among these precedents
2. Else
   (a) Select all precedents with a maximal weighted balance (positive differences - negative differences) with the focus case
   (b) Predict the most common outcome among these precedents

We would like to test whether C1, assigning priority to precedents with no negative differences, improves performance. Therefore, we define the algorithm *balance*, which leaves out the priority rule:

Algorithm 2) balance (C2, C3)

1. Select all precedents with a maximal weighted balance (positive differences - negative differences) with the focus case
2. Predict the most common outcome among these precedents

We are also interested in testing whether C3, to compensate for negative differences with positive ones, improves performance. We define an alternative algorithm, which ignores the positive differences and only tries to minimize negative differences:

Algorithm 3) minimize negative (C1, C2)

1. select all precedents with minimal weighted negative differences with the focus case
2. predict the most common outcome among these precedents

Finally, we would like to test the performance when we drop the concept of positive and negative differences. For this purpose, we define the following nearest neighbors approach:

Algorithm 4) nearest neighbor ()

1. select all precedents with minimal weighted differences with the focus case
2. predict the most common outcome among these precedents

To compare the performance of the different algorithms, we measured their performance on the complete Mushroom (8124 instances), Churn (7032 instances) and Admission (500 instances) set. The data sets are divided into a random selection of 80% training data and 20% test data. The training set is used to run a logistic regression to determine the weights of the features. Using these weights, the tendency of features is determined and all training instances are turned into cases, making up the case base. The performance of an algorithm is then measured on the test set using either the entire case base or a consistent subset - the result after running the consistency algorithm. The performance is measured as the accuracy, calculated as the numbers of correct predictions divided by the total number of predictions. To obtain a more stable evaluation, we repeat this whole process ten times and report the average outcomes. Table 7.1 shows the results of the experiment.

Algorithm	Mush	Churn	Admission
Priority + balance (all)	**1.000** (0.00)	0.716 (0.03)	0.907 (0.03)
Balance (all)	**1.000** (0.00)	0.691 (0.03)	0.888 (0.03)
Minimize negative (all)	**1.000** (0.00)	0.733 (0.03)	0.907 (0.03)
Nearest neighbor (all)	**1.000** (0.00)	0.717 (0.02)	**0.917** (0.03)
Priority + balance (consistent)	**1.000** (0.00)	0.738 (0.04)	0.903 (0.03)
Balance (consistent)	**1.000** (0.00)	0.727 (0.04)	0.884 (0.03)
Minimize negative (consistent)	**1.000** (0.00)	**0.756** (0.03)	0.902 (0.03)
Nearest neighbor (consistent)	**1.000** (0.00)	0.746 (0.02)	0.913 (0.02)

Table 7.1: *Mean and standard deviation of the accuracy of the different algorithm. The Mushroom and Churn data set were run 10 times. Due to the small size, the Admission set was run 100 times. Per run 80% training - and 20% test instances are selected randomly*

All of the algorithms reached a 100% accuracy when trained on 80% of the Mushroom data set. Note that this set is already consistent, so the consistency algorithm did not change the case base in any way. *Minimize negative* - which minimizes negative differences while ignoring positives - appeared to score best on the Churn set. *Nearest neighbor* performed best on the Admission set.

The *Balance* algorithm - which minimizes negative and maximizes positive differences - was least successful.

The consistency algorithm appeared to pay off for the Churn set; all case-based methods performed better using a consistent case base. Interestingly, the algorithms slightly worse with a consistent case base on the Admission set.

7.1.1 Comparison with non case-based classifiers

All the algorithms in the previous experiment used some form of case-based reasoning. To compare their performances, we measured the accuracy of the five different classifiers introduced in Chapter 5 using the same set-up. Table 7.2 shows their results.

Logistic Regression	**1.000** (0.00)	0.798 (0.01)	**0.929** (0.02)
Decision Tree	**1.000** (0.00)	0.728 (0.01)	0.898 (0.03)
Naive Bayes	0.958 (0.01)	0.726 (0.01)	0.859 (0.03)
Support Vector Machine	**1.000** (0.00)	0.796 (0.01)	0.922 (0.02)
Adaboost	**1.000** (0.00)	**0.802** (0.01)	0.916 (0.02)

Table 7.2: *Mean and standard deviation of the accuracy on the five alternative classifiers. The Mushroom and Churn data set were run 10 times; the Admission set 100 times. Per run 80% training - and 20% test instances are selected randomly.*

Compared to the best alternative classifiers in the selection, the case-based models reached a lower accuracy on the Churn set (4.6% less), a slightly lower accuracy on the Admission set (1.2% less) and an equal accuracy on the Mushroom set.

7.2 Feature selection

In the previous section, we only considered performance in terms of accuracy. As we aim to create an explainable system, comprehensibility is as important. One way to improve the comprehensibility is to limit the number of features used in the model. Decreasing the number of features does not have to lead to a sacrifice in performance. In fact, feature selection is a known tool to improve classification performance (Dash and Liu 1997).

7.2.1 The selection algorithm

The subject of feature selection is a field of study in itself; extensive experimentation falls beyond the scope of this research. In this work, we will only consider the effects of using a simple algorithm introduced in the previous chapter.

When we applied the feature selection algorithm, we noticed that the algorithm sometimes terminates at a local maximum, which is not the global maximum. In that case, the accuracy decreases when the least important feature is removed, so the selection algorithm stops. However, when even more features would be removed, the accuracy would increase again. To deal with this problem, we relaxed the condition for continuing searching for the best set of features. Instead of terminating searching as soon as there is some drop in accuracy, we slightly changed the algorithm to:

```
old accuracy, new accuracy, best accuracy = 0, 0, 0
relaxation = 0.05
while old accuracy <= new accuracy + relaxation do
    old-accuracy = new-accuracy
    new-accuracy = the model's current accuracy
    if new accuracy ≥ best accuracy then
        best accuracy = new accuracy
        best features = current features
    end if
    measure the importance of the current features
    select the feature f with the least importance
    remove feature f from the data set
    (make the case base consistent)
end while
return best features
```

The *relaxation value* allows the algorithm to continue, even though the previous feature removal resulted in a small drop in accuracy. Note that the algorithm still returns a set of best-performing features; the if-statement makes sure the return variable is only updated when the performance is at least as good.

The relaxation value of 0.05 was chosen arbitrarily. A low value can minimize computing power by forcing the algorithm to stop when continuing seems unlikely to result in any improvement. When this is of no concern, a higher value

could be used, allowing the algorithm to consider every number of features.

A second short-cut we applied to save computing power, is to drop a bunch of features whose importance falls below a certain threshold. This was especially useful in the Mushroom data set, consisting of 116 binary features. Of these features, 58 appeared to be at least ten times less important than the most important feature, and these did not contribute to the performance. We removed these all together, before starting the selection algorithm.

We must note that removing features can cause inconsistencies in the case base. This can be illustrated by the fact that removing all features, would cause an inconsistency for every two cases with a different outcome; their features are then at least as good for the outcome of the other case. When we want to use the feature selection algorithm with a consistent case base, the consistency algorithm must be applied after every feature removal.

7.2.2 Applying feature selection

For the feature selection process, we used the combination of the case-based classifier and consistency setting that performed best as the prediction model: we used *Nearest neighbor* on the complete Mushroom- and Admission set and *Minimize negative* on a consistent Churn case base. For the selection algorithm, we used all 500 instances of the Admission set, and a random selection of 1000 instances from the Churn and Mushroom set.

	Mushroom	Churn	Admission
Number of features before	116	30	7
Best accuracy before	1.000 (0.00)	0.756 (0.03)	0.917 (0.03)
Number of features after	17	8	4
Best accuracy after	1.000 (0.00)	0.792 (0.01)	0.933 (0.02)

Table 7.3: *Number of features and accuracy on the best algorithm (Nearest neighbor for Mushroom and Admission, Minimize negative for Churn) before and after feature selection*

Table 7.3 shows the number of features and accuracy on the best algorithm before and after the selection. A substantial decrease in features was possible for all three data sets. The performance on the Mushroom set stayed the same, while

the feature selection benefited the performance of the Churn (+ 3.6%) and Admission (+ 1.6%) classifiers.

Classifier	Mushroom	Churn	Admission
Priority + balance (all)	**1.000** (0.00)	0.751 (0.01)	0.909 (0.03)
Balance (all)	**1.000** (0.00)	0.612 (0.01)	0.875 (0.04)
Minimize negative (all)	**1.000** (0.00)	0.751 (0.01)	0.912 (0.03)
Nearest neighbor (all)	**1.000** (0.00)	0.753 (0.01)	0.913 (0.02)
Priority + balance (consistent)	**1.000** (0.00)	0.790 (0.01)	**0.933** (0.02)
Balance (consistent)	**1.000** (0.00)	0.743 (0.01)	0.929 (0.02)
Minimize negative (consistent)	**1.000** (0.00)	0.792 (0.01)	0.930 (0.03)
Nearest neighbor (consistent)	**1.000** (0.00)	0.790 (0.01)	0.932 (0.02)
Logistic Regression	0.999 (0.00)	0.786 (0.01)	0.926 (0.02)
Decision Tree	**1.000** (0.00)	0.759 (0.01)	0.899 (0.03)
Naive Bayes	0.990 (0.00)	0.709 (0.01)	0.876 (0.03)
Support Vector Machine	**1.000** (0.00)	0.782 (0.01)	0.921 (0.02)
Adaboost	**1.000** (0.00)	**0.793** (0.01)	0.912 (0.02)

Table 7.4: *Mean and standard deviation of the accuracy on the complete data sets after feature selection using the four case-based methods, with all data and with a consistent CB, and five different classifiers. The Mushroom and Churn data set were run 10 times; the Admission set 100 times. Per run 80% training - and 20% test instances are selected randomly*

Finally, Table 7.4 shows the performances of all models after feature selection. After feature selection, both the Churn and Admission set performed best with a consistent case base. Interestingly, the feature selection did not benefit the non-case-based classifiers; their performances slightly diminished.

7.3 Discussion

In this chapter, we experimented with different algorithms applying none, some or all of the three concepts derived from AF-CBA. The performance of the different algorithms appeared to be relatively similar. *Balance* performed worst. In contrast to *Priority + balance* and *Minimize negative*, this algorithm does not prioritize minimizing negative differences; it simply tries to maximize the weighted balance (positive differences minus negative differences). This seems to suggest

that minimizing negative differences is more important than maximizing positive ones.

Feature selection appeared to be highly effective. It caused the data instances to become substantially more comprehensible. In the Mushroom data set more than 85% of all features could be removed. Moreover, feature selection also helped to improve the accuracy on the Churn and Admission set. Creating a consistent case base appeared to be a successful approach as well. After feature selection, both the Churn and Admission set performed better on all case-based algorithms using a consistent case base.

The Mushroom data set appeared not too suitable for comparing the systems; all algorithms performed perfectly. A motivation for selecting this data set was to compare AF-CBA to the approach of Čyras et al. (2019), called AA-CBR. We implemented AA-CBR and found that, after feature selection, the system reached 100% accuracy as well. Further analysis revealed that the Mushroom data set consists - after our feature selection - of only 38 unique cases. Therefore, for every focus case, there can be found multiple exact copies in the case base; every case-based algorithm should be well suited to find those. As AA-CBR only works with binary features, the other data sets are unsuitable for the comparison. It would be interesting to compare the two approaches in future research.

In this chapter, we only considered predictive performance in terms of accuracy. Admittedly, this is a limited representation of the performance of the models. It would be interesting to consider other metrics - such as recall and precision - and information about the precedents the algorithms base their analysis around. Potentially this could lead to further insights about the applicability of the different concepts.

Another limitation of the experiment was the usage of data sets. On all three data sets, a logistic regression performs fairly well. Replacing this classifier is not our goal; it seems unlikely that a case-based system can be considered more interpretable than a logistic regression. Further research must reveal whether case-based classifiers could also replace black boxes when other white boxes fail to do so.

8. User experiment

8.1 Introduction

An online user experiment was conducted to gain an understanding of the interaction between users and the systems. Concretely, we wanted to investigate two questions:

1. How do users perceive the Case-Based Argumentation explanation systems in terms of convenience, trust and insight?
2. How do the Case-Based Argumentation explanation systems influence the response of users towards the model's predictions?

In the experiment, two types of Case-Based Argumentation systems were used: a justification- and a monitor system. To compare these systems, we also applied two other ways of presenting the predictions: with absence of an explanation (the control condition) and with an explanation in the form of feature importance.

In this chapter, we will first introduce the experimental case. After that, we will formulate our research hypotheses, describe the method used and present the results. In the final section, we will discuss the implications of the results and the validity of the experiment.

8.2 Experimental Case

In the experiment, the participants played the role of a telecommunication employee. Their company aimed to retain as many customers as possible while

minimizing their cost. To achieve this, the company wanted to offer a special discount to the customers who were most likely to churn: cancel their subscription. During the experiment, the participants were in the first stage of this process; their task was to estimate the likelihood of customers to churn.

The participants were assisted by a predictive model that made a binary prediction (stay or churn) for every customer. In the control condition, the participants only saw the customer profile and the prediction of the model. In the three other conditions, the CBA-systems or feature-importance method provided them with extra information that could help them make their estimation.

8.3 Hypotheses

The nature of the experimental research is exploratory, as the two CBA-systems have - to the best of our knowledge - not previously been tested on users. To investigate whether there seem to be differences between the presentation methods in the interactions with users, we will first assume that all methods lead to the same results. By testing these *null-hypotheses*, we aim to detect potential differences.

We formulate four hypotheses concerning the first research question to establish whether the users perceive the ways of presenting predictions differently. First, we articulate three null-hypotheses, stating that all forms of presenting are perceived equally:

Hypothesis 1. The users will consider the four forms of presenting the predictions equally convenient.

Hypothesis 2. The users will perceive predictions of the model in the four forms of presenting the predictions as equally trustworthy.

Hypothesis 3. The users will consider the four forms of presenting the predictions as equally insightful.

We would also like to investigate whether the level of familiarity with machine learning of the user is related to the way the different systems are perceived. Therefore, we articulate the following null-hypothesis:

Hypothesis 4. Users who are unfamiliar with machine learning will rate the

presentation systems in the same way as users who are familiar with machine learning.

In addition to the way the users perceive the methods, we are also interested in the way the methods influence their estimations. What we would like to know is whether the estimates of the users are *better* when using certain explanation methods. However, it is nontrivial what being a better estimation on a scale from 0 to 100 entails. Even though we do know that a customer stayed, this does not mean that 0 (will very likely stay) is the most reasonable estimation for that customer. It may be that 30%, or even 80%, of the customers with the same profile would, in fact, churn. If we knew what the best estimation would be, the role of a user would arguably be superfluous.

Instead of defining 'best estimations', we can base our analysis on the way the estimations relate to the predictions of the black-box model. During every task, the participants receive a binary prediction of the model: Stay or Churn. Using the prediction and the information presented by the presentation method, the participants make their own estimation on a scale. A user who would strictly follow the binary prediction of the model would estimate 0 in case of the prediction 'Stay' and 100 in case of 'Churn'. Any points on the scale removed from these ends could be seen as an expression of doubt towards the prediction. Using this notion, we could say that a user can add value to the predictions of a model by expressing doubt in cases of incorrect predictions while supporting correct predictions.

First, we will analyze whether the general amount of doubt expressed towards the predictions is equal among the presentation methods:

Hypothesis 5. The users will express the same amount of doubt towards the predictions of the model for the four ways of presenting the predictions.

After that, we will take a look at the difference in doubt between correct and incorrect predictions. We call the extra doubt that is expressed in cases of incorrect predictions the *added doubt*. We will test whether the added doubt is equal among the presentation methods with the following null-hypothesis:

Hypothesis 6. The users will express the same amount of added doubt towards incorrect predictions of the model for the four ways of presenting the predictions.

For the two CBA-systems, we would also like to establish whether the content

of the information the system shows influences the amount of doubt the users have towards the predictions of the model. We formulate the following null-hypotheses, stating there is no difference in doubt:

Hypothesis 7. The users will have the same amount of doubt towards predictions when the justification system supports the prediction, as when it does not.

Hypothesis 8. The users will have the same amount of doubt towards predictions when the monitor system shows supporting evidence for the prediction, as when it does not.

8.4 Method

Participants A total of 20 persons participated in the experiment; 10 of them identified themselves as a woman, 9 as a man, and one preferred not to indicate gender. Their ages ranged between 22 and 57, with an average age of 28. The sample was high-educated; 17 of the participants completed a Bachelor's Degree, 7 also a Master's Degree, and 1 finished a PhD. Nearly all participants, 19, were at least a little bit familiar with machine learning; 16 participants were at least a little bit familiar with explaining machine learning. Table 8.1 shows their self-reported familiarity.

	Familiarity with	
	machine learning	explaining machine learning
Not at all	1	4
A little bit	7	6
Somewhat	4	7
Very	8	3

Table 8.1: *Frequencies of self-reported familiarity with (explaining) machine learning in the sample*

Participants were recruited via the social circle of the researchers. They did not receive any remuneration for their participation.

Materials Informed consents were used containing the time the experiment would take - approximately 40 minutes - and the rights of the participants: confi-

dential and anonymous processing of their information, and the right to quit the experiment at any point. The informed consent can be found in Appendix A.

During the experiment, two types of questions were asked: estimation- and review questions. Estimation questions were asked to complete one task and read: 'How likely does it seem to you that *name* will churn?'. After every condition, consisting of one practice, and four actual tasks, the participants were asked the following three review questions:

- How convenient did you find your task during this section?
- How trustworthy did you consider the predictions to be during this section?
- How insightful was the information you received during this section to you?

Responses to these questions were collected on a visual analogue scale; participants specified their response by indicating a position along a continuous line between two endpoints. Figure 8.1 shows an example of such a scale. The submitted position on the scale results in a number between 0 (Not likely at all) and 100 (Very likely).

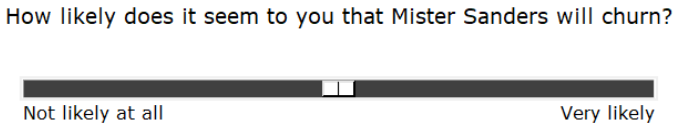

Figure 8.1: *Example of the estimation task for the customer Mister Sanders*

Visual analogue scales have been found to be preferred over categorical scales (Funke and Reips 2012). They do not restrict answers to a small number of response options, which enables us to detect little differences. Moreover, the continuous data collected with scales can be used for a greater number of statistical tests (Gerich 2007).

The explanation systems were implemented in Jupyter Notebooks using Python. A supporting user interface was built using the *tkinter* library. For the Churn case scenario, we used the Telco Churn data set *Telco Customer Churn* (2018), which was introduced in Chapter 5. In the previous chapter, we saw that the AdaBoost model appeared to be most accurate on this data set. This model was

applied as the black-box predictive model, using the *sklearn* library. The same 16 customers were used for all estimation tasks. These customers were classified into four groups of four. Within these groups, three customers received a correct prediction of the black-box model and one an incorrect.

The justification- and monitor systems were presented to the user in the interactive interface, as described in Chapter 6. The feature importance condition contained only part of this information: all information regarding the focus customer. Figure 8.2 shows what the information in this condition looks like for the example customer. In the presentation form in which an explanation was absent, the feature importance information - represented as colored bars - was left out as well. Table 8.2 presents an overview of the properties of the different presentation methods.

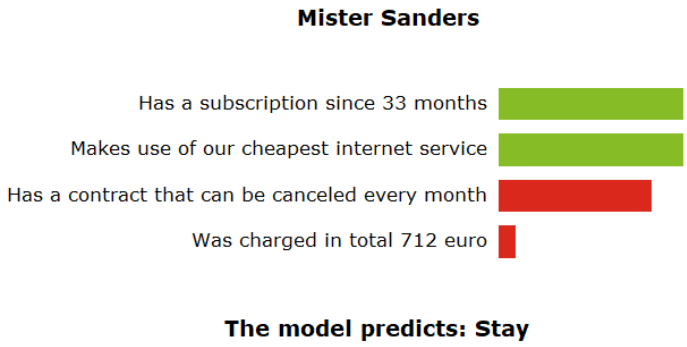

Figure 8.2: *Example of the information presented in the feature importance condition*

Presentation method	Customer profile	Model prediction	Feature importance	Comparisons with previous customers
No explanation	X	X		
Feature importance	X	X	X	
Justification	X	X	X	X
Monitor	X	X	X	X

Table 8.2: *Properties of the presentation methods. A cross (X) represents that a property is present within the method*

After the experiment, participants completed a short survey. The survey measured their age, gender, level of education, level of familiarity with machine

learning and level of familiarity with explaining machine learning. The survey can be found in Appendix A.

Skype, Teams or Zoom - depending on the participant's preferences - was used to connect with the participants during the experiment.

Procedure The experiments were conducted via online video calls. During these calls, the participants could perform the experiment by taking over the control of the laptop of the researcher. The participants were allowed to ask questions or give comments at any point of the experiment.

The general procedure of the experiment is visualized in Figure 8.3. First, the participants were asked for their agreement to participate in the research with an informed consent. After that, the experimental case and general set-up of the experiment were introduced. All text used for explanation of the experiment to the user can be found in Appendix A.

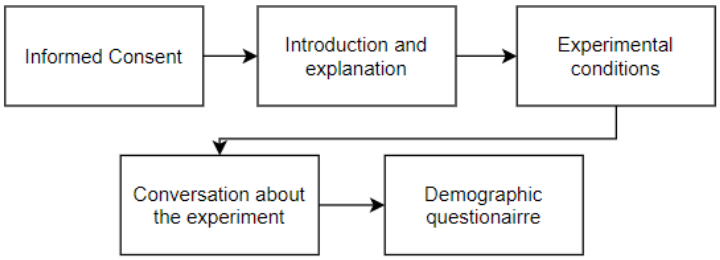

Figure 8.3: *General procedure*

Next, the experimental conditions started. There were four conditions; one for every presentation method. Every participant took part in all of them. The procedure within these conditions, as visualized in Figure 8.4, was the same for every condition. The participant first received an explanation of the condition. After that, the participant practiced the task with one example customer. The example customer used for the practice task was the same for all participants in all conditions. After the example, the participants were prompted to express any remaining questions. When they were ready, they started with the actual task, in which they estimated the likelihood of churning for four customers.

The structure of a single estimation task is presented in Figure 8.5. The information page showed the prediction from the black-box model, the profile of the

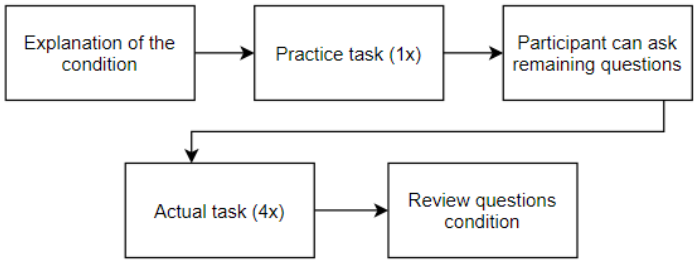

Figure 8.4: *Procedure experimental conditions*

customer and possibly - depending on the condition - extra information. The question page only consisted of the estimation question, as shown in the example in Figure 8.1. Participants could switch between the information-page and the question-page until they submitted an answer to the question.

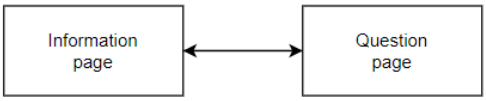

Figure 8.5: *Structure single estimation*

The order in which the conditions were presented was randomly distributed over all participants. The customer groups and conditions were randomly paired, such that each customer group appeared equally often together with every condition in the research.

8.5 Results

We analyzed the data using SPSS statistics version 25. The first section concerns the analysis of the review scores (hypotheses 1-4). In the second section, the churn estimations are analyzed (hypotheses 5-8).

8.5.1 Reported convenience, trust and insight

Three one-way repeated measures analyses of variance (ANOVA's) were used to compare reported convenience, trust and insight of the four different explanation methods. Figure 8.6 shows the average review scores from the sample per method.

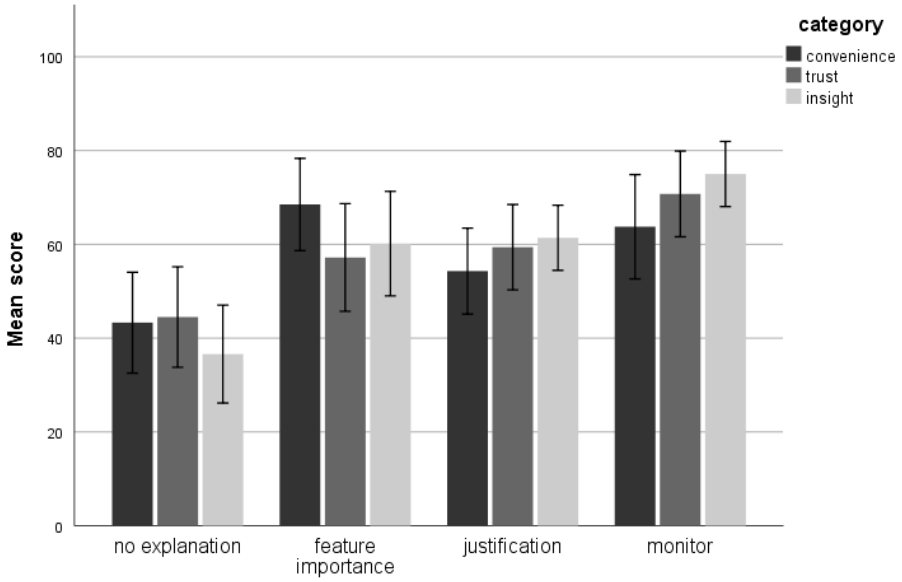

Figure 8.6: *Average reported convenience, trust and insight per method. Error bars show the 95% confidence interval.*

Given the small sample size (20), a Shapiro-Wilk test was used to check for a possible violation of the normality assumption. The Shapiro-Wilk statistics indicated that the assumption of normality was supported for most distributions. However, the assumption of normality was violated for reported trust using the monitor method ($p = .011$), and insight using the monitor ($p = .022$) and feature importance method ($p = .037$). Therefore, a more conservative p-value of .01 instead of .05 was used to conduct significance tests for these distributions.

The F_{MAX} for the convenience- (1.47), trust- (1.59) and insight (2.58) test demonstrated homogeneity of variances. Mauchly's test indicated that the assumptions of sphericity were not violated for the convenience and insight test, but was violated for the trust test. Therefore, the degrees of freedom for this ANOVA were adjusted by multiplying them by the Huynh-Feldt Epsilon.

The ANOVA results of the convenience test show that the sample reported some methods to be more convenient than others, $F(3, 57) = 4.84$, $p < .001$, $\eta^2 = .203$, rejecting hypothesis 1. Pairwise comparisons revealed that the feature importance method ($M = 68.5$, $SD = 21.0$) was reported to be significantly more convenient than no explanation ($M = 43.3$, $SD = 23.0$).

The second hypothesis was also rejected, as the trust test-results show that the sample reported predictions to be more trustworthy using some methods than others, $F(2.50, 47,46) = 7.42$, $p = .001$, $\eta^2 = .281$. Pairwise comparisons further revealed that using the feature importance method ($M = 57.2$, $SD = 24.5$) or the monitor method ($M = 70.8$, $SD = 19.5$) predictions were reported to be significantly more trustworthy than during absence of an explanation ($M = 44.5$, $SD = 22.9$).

Finally, the insight test-results show that the sample reported some methods to be more insightful than others, $F(3, 57) = 16.68$, $p < .001$, $\eta^2 = .467$, rejecting hypothesis 3. Pairwise comparisons revealed that the absence of an explanation ($M = 36.6$, $SD = 22.3$) was reported to be significantly less insightful than the presence of any of the three explanation methods: feature importance ($M = 60.2$, $SD = 23.8$), justification ($M = 61.4$, $SD = 14.8$) and monitor ($M = 75.0$, $SD = 14.8$). Due to the conservative alpha value used, the monitor method was not shown to be significantly more insightful than the justification method ($p = .023$).

We also wanted to establish whether the methods were perceived differently among participants with a different level of familiarity with machine learning. Participants rated their familiarity on a scale of four: not at all, a little bit, somewhat or very familiar with machine learning. We used this scores to split the sample into two: one group that is (relatively) unfamiliar with machine learning - members answered ‘not at all’ or ‘a little bit’ - and a group that is familiar - members answered ‘somewhat’ or ‘very’. The unfamiliar group contained 8 participants, the familiar group 12. Figure 8.7 shows the average cumulative scores of convenience, trust and insight per method for the two groups.

Independent samples t tests were used to compare the cumulative scores of reported convenience, trust and insight for the four methods from participants that are unfamiliar with machine learning to those of participants that are familiar. None of the t-tests was statistically significant, so we cannot reject hypothesis 4, stating that methods are perceived equally among the two groups.

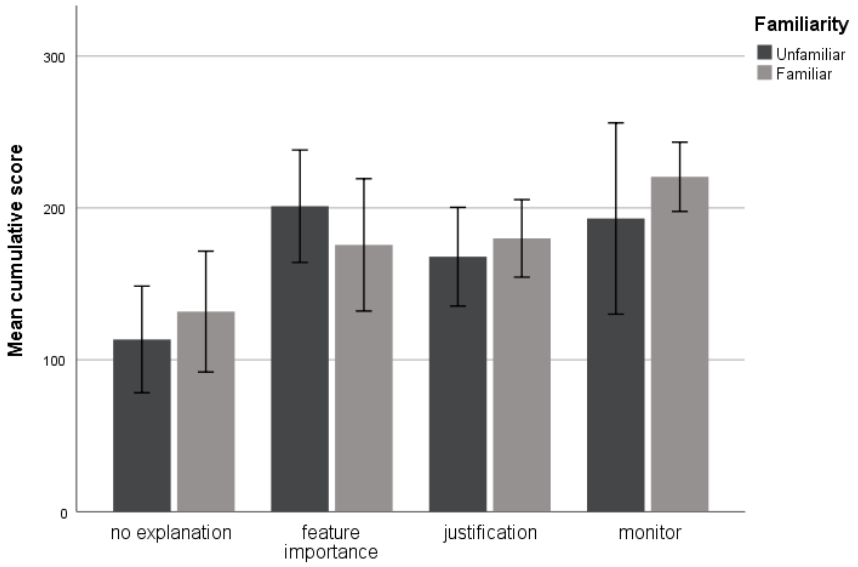

Figure 8.7: *Average cumulative score of convenience, trust and insight per method for participants that are unfamiliar or familiar with machine learning. Error bars show the 95% confidence interval.*

To sum up, the ANOVA results showed that:

1. The feature importance method was considered more convenient than the absence of an explanation
2. The predictions of the model were considered more trustworthy when using the feature importance or monitor method than in case of absence of an explanation
3. The three explanation methods were considered more insightful than the absence of an explanation

A t-test comparing the cumulative scores of participants that are unfamiliar with machine-learning to the scores of familiar participants found no significant differences between the groups.

8.5.2 Churn estimations

The second part of the analysis concerns the estimations the participants made during the tasks. We transformed their estimates into variables representing the amount of doubt participants expressed towards the predictions of the Black Box. A participant that would have no doubt at all, was assumed to follow the predictions of the model 100%. We call this response *no_doubt*, and define it as:

$$no\_doubt(prediction) = \left\{ \begin{array}{ll} 0, & \text{when prediction == 'Stay'} \\ 100, & \text{when prediction == 'Churn'} \end{array} \right\}$$

Now we can calculate the doubt towards a prediction for participant p as:

$$doubt_{\mathrm{p}}(prediction) = abs(no\_doubt(prediction) - estimation_{\mathrm{p}}(prediction))$$

We used an ANOVA to test whether there is a general difference between the explanation methods in the amount of doubt the sample expresses towards the predictions. Table 8.3 shows the average amount of expressed doubt in the predictions per method, as measured within the sample.

	no explanation	feature importance	justification	monitor
M (SD)	28.8 (12.7)	25.3 (11.6)	31.5 (10.8)	26.7 (7.3)

Table 8.3: *Mean and standard deviation of the amount of doubt per method*

Boxplots and Shapiro-Wilk statistics indicated that the assumption of normality was supported; Fmax was 3.049, demonstrating homogeneity of variances; and Mauchly's test indicated that the assumption of sphericity was not violated. The ANOVA results did not give reason to reject null-hypothesis 5, stating that there is no difference in the overall amount of doubt the sample expresses towards the predictions throughout the different methods, $F(3, 57) = 2.11$, $p = .109$.

The previous test gave some insight into the general tendency to follow the predictions of the black box model per method. In an ideal scenario, the doubt towards the prediction model would be high in cases of an incorrect prediction, and low in cases of a correct one. To test whether we observe this behaviour, we calculated the average doubt on correct and incorrect predictions separately for each method. Then we calculated the added doubt for incorrect predictions as:

added doubt incorrect = average doubt incorrect - average doubt correct

The higher the added doubt for incorrect predictions, the better the sample could distinguish between correct and incorrect predictions of the black-box model. Table 8.4 shows the average added amount of doubt for incorrect predictions per method.

	no explanation	feature importance	justification	monitor
M (SD)	11.1 (4.3)	13.2 (4.1)	9.8 (5.2)	22.3 (4.4)

Table 8.4: *Mean and standard deviation of the added amount of doubt for incorrect predictions per method*

We used an ANOVA to test whether the distributions in added doubt for incorrect predictions differed significantly per method. All assumptions of normality, homogeneity of variances and sphericity were met. The ANOVA results show that null-hypothesis 6, stating that there is no difference in added amount of doubt for incorrect predictions cannot be rejected, $F(3, 57) = 1.43$, $p = .242$.

Finally, we wanted to establish how the information presented by the CBA-systems relates to the doubt users express towards the predictions. Our experimental design makes the analysis of these questions challenging. We refrained from manipulating the outputs of the CBA-systems, which resulted in uneven numbers of data points per output category. Within a condition - consisting of four customers - participants received certain outputs multiple times while missing out on other possible outputs of the system. These disparities make the structure of the data unsuitable for most statistical tests. Instead, we will base our analysis on scatter plots, displaying all relevant data points, and confidence intervals of the mean per output category.

The justification system provided three types of judgments towards the prediction of the black box: ‘convincing’ (in 60 data points), ‘somewhat convincing’ (10 data points), or ‘unconvincing’ (10 data points). The fourth type of judgment, ‘somewhat unconvincing’, did not occur within the selection of customers. Figure 8.8 shows all doubt-scores measured per judgment of the system.

Optically, a trend seems to exists in the expressed doubt; the less convincing the CBA-system judges the predictions, the higher the doubt. Table 8.5 shows the averages and 97.5% confidence intervals per judgment. We can declare the amount of doubt expressed presented with a ‘convincing’- and ‘unconvincing’ judgment to differ significantly using an *alpha* of 0.05; their 97.5% confidence intervals do not overlap. Hypothesis 7, assuming an equal amount of doubt towards predic-

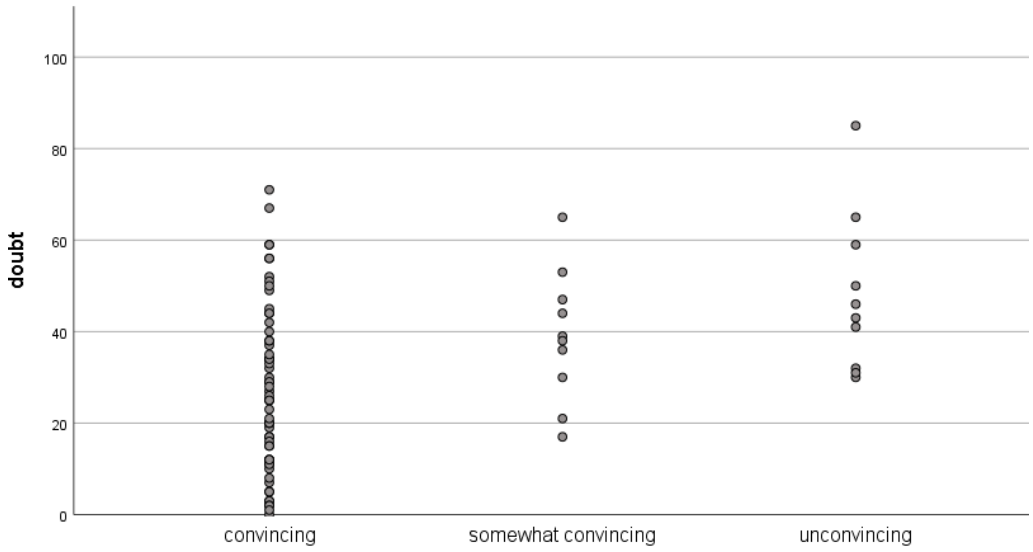

Figure 8.8: *The amount of doubt expressed towards the predictions per judgment of the justification system. The scatter-plot shows all 80 data points measured in this condition during the experiment.*

tions when the justification system supports the prediction as to when it does not, must be rejected.

	convincing	somewhat convincing	unconvincing
M (SD)	27.52 (18.1)	39.00 (14.38)	48.20 (17.4)
97.5% CI	22.15 - 32.89	26.79 - 51.21	33.41 - 62.99

Table 8.5: *Mean, standard deviation and 97.5% confidence interval of the doubt expressed per judgment of the justification system*

The monitor system has less clear categories of output; the system does not make any judgments but presents information about the five nearest neighbors to the focus case. We chose to simplify the output into two percentages: the percentage of neighbors which made the same decision as is predicted for the focus case, and the percentage of neighbors for which the model predicted that decision correctly. The higher both percentages, the more reassuring we would expect the outcome to be for the model's prediction. We multiplied the two percentages to obtain one variable, representing how reassuring the output was. A 50% cut-off point was used to classify the outputs as reassuring (above 50% or higher)

or alarming (below 50%). The classification resulted in 35 reassuring and 45 alarming data points, presented in a scatter plot in Figure 8.9.

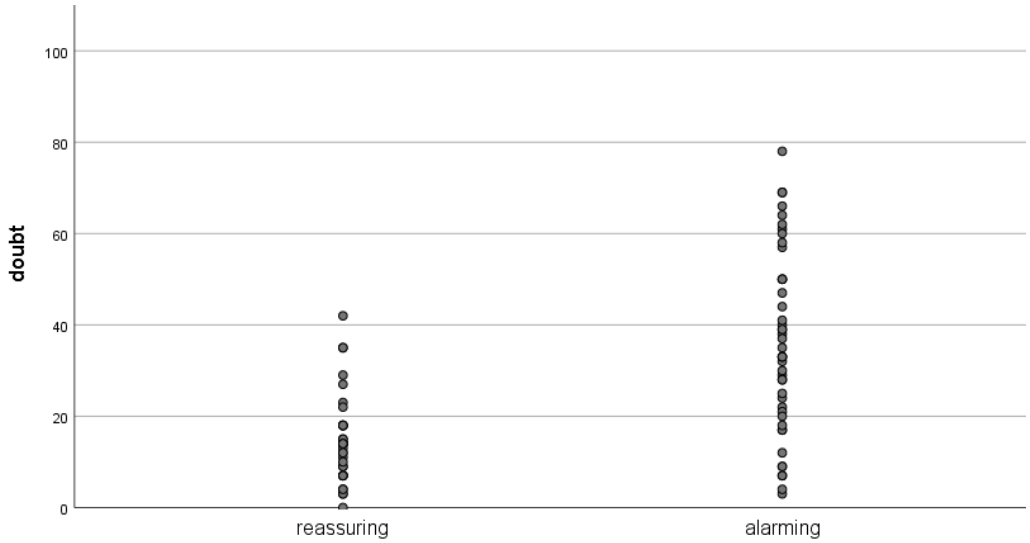

Figure 8.9: *The amount of doubt expressed towards the predictions for reassuring and alarming outputs of the monitor system. The scatter-plot shows all 80 data points measured in this condition during the experiment.*

When presented a reassuring output, all participants expressed a doubt below 50% towards the prediction. For alarming outputs, the doubt ranged up to 78%. Table 8.6 shows the averages and 97.5% confidence intervals per category. With a substantial difference - larger than ten doubt-scores - between the two confidence intervals, we reject the final hypothesis and declare the amount of doubt to differ significantly between reassuring and alarming outputs.

	reassuring	alarming
M (SD)	14.77 (9.65)	35.89 (19.85)
97.5% CI	10.95 - 18.60	29.02 - 42.76

Table 8.6: *Mean, standard deviation and 97.5% confidence interval of the doubt expressed with reassuring and alarming output of the monitor system*

8.6 Discussion

In this section, we will discuss the experimental results. We will complement these results with comments participants made during and after the experiment. After that, we will discuss the validity of the research.

8.6.1 Experimental findings

In this Churn experiment, we collected two types of information: review scores and task data in the form of estimations. The review scores can give us insight into how the users experienced the different systems. Within the sample, the presentation of feature importance was considered most convenient. This result may be related to the fact that multiple participants remarked that the large amount of information presented by the CBA-systems made it hard to process. Though the time used per condition was not measured, the tasks using the CBA-systems seemed to take substantially longer. It would be interesting to investigate whether the information presented by the CBA-systems could be reduced or be presented in a more convenient way.

The sample reviewed the predictions with all forms of explanations more trustworthy than without. Feature importance and monitor system were shown to significantly increase the trust in the black box predictions. Within the sample, the monitor system increased the trust the most. Some participants remarked that the justification system did not feel trustworthy. Two reasons were mentioned for this:

1. The example customers differ a lot from the focus case
2. It is unconvincing to base the judgment on only two examples

The justification system tried to minimize the negative differences while maximizing positive ones. This strategy made the system select examples of customers who had almost no similarities with the focus case, but many positive differences. This way of reasoning appeared unintuitive to some. As stated by Sørmo, Cassens, and Aamodt (2005), the usefulness of CBR is dependent on the ability of users to confirm the similarity assessment. Possibly, the trustworthiness could have been increased by explaining the reasoning of the system in more detail.

Further explanation of the system might have also helped with regard to the second reason. During the experiment, it was unknown to the users that the examples came from a consistent subset of the case base, which information might have increased their trust. Still, it may be the case that for some users, reasoning with a small number of examples does not constitute trust.

Participants rated all three methods significantly more insightful than the absence of an explanation. The monitor system was rated most insightful. Comparing the scores of the CBA-systems, we see that the sample rated the monitor system somewhat more positive on all three measures.

We found no significant differences between users familiar and unfamiliar with machine learning. Though, the mean scores, as shown in Figure 8.7, seem to suggest that a difference might be found with a larger sample. Unfamiliar participants rated the feature importance method highest, whereas the monitor system was the favorite for familiar participants. This stresses the importance of testing the explanation methods with potential users; plots might suit machine-learning experts, but be less appropriate for other types of users.

The average amount of doubt expressed towards the estimations was similar for the four methods. An ideal system would increase doubt in case of incorrect predictions while supporting trust in correct ones. In this experiment, the added doubt in case of incorrect predictions was largest when using the monitor system, but no significant difference compared to the other methods was found. Further experimentation using more test items or a larger sample is required to establish their impact.

Within the sample, alarming sounds from the CBA-systems went along with higher expressions of doubt towards the predictions. When the justification system judged a prediction unconvincing, users expressed on average a doubt of about 50%, choosing the middle ground between the two disagreeing systems. This gives the indication that the model's predictions and judgments of the justification system were treated as equally important.

8.6.2 Validity of the experimental research

With the experiment, we obtained some insights into the suitability of the different methods and uncovered points of attention for further development. Admittedly, the experiment was small; both in sample size and in the number of tasks

per system. The size of the experiment especially fell short for measuring the impact of the systems on the task performance. Given the unpredictable behaviour that sometimes occurs in the churn set, many estimations would be needed to draw hard conclusions about the performance of the users. With a small selection, unfounded predictions might be rewarded by unexpected decisions. More experimentation is needed to answer questions related to performance.

The brief exposure of the participants to the presentation methods might threaten external validity. Only one practice example and four actual tasks were used per condition. Besides, the participants did not receive guidelines on how to interpret the information. The briefness of the exposure and lack of training might have disadvantaged the systems consisting of larger amounts of information. More extensive exposure to the systems could enable improved generalizations to the long-term usage of the systems.

9. Conclusion

This thesis aimed to investigate the applicability of CBA for the goal of making machine learning more interpretable. In this final chapter, we will try to answer the questions we posed in the beginning. Before we come to the main research question, we will formulate an answer to all of the sub-questions. We end the chapter with suggestions for future research.

9.1 Conclusions of the research

S1) *What kind of method is applicable to generate feature information to be used in AF-CBA?*

In this work, we combined AF-CBA with the automatic generation of feature information. Specifically, we used weights that were generated by a logistic regression. The structure of the information created with the logistic regression - assigning features a positive or negative weight - can easily be incorporated in a CBA system. Furthermore, this approach is fast and does not require any expertise on the problem.

This final benefit is, at the same time, a vulnerability of the approach. All features are represented as numbers; we ignore their special properties and the relations they might have with other features. This simplification of the content may lead to limitations in the reasoning of the system. We, for example, assumed that the importance of a dimension has a linear relation with its value. This may be an assumption you do not want to make. Furthermore, with our current approach, we did not find much evidence for compensating negative differences by positive ones. It may be that an approach which specifies particular feature relations

could support this idea. The method of Prakken (2020) allowed for the addition of top-level (expert) information. In our current approach, expert knowledge could be used to adjust the weights. It would be interesting to investigate the possibilities for allowing more complicated forms of top-level information, such as rules, as well.

All in all, using an automatic approach to generating feature information seems applicable to AF-CBA. With this approach, a basis of quantitative information about all features is efficiently generated. For some problems, this information may be sufficient; for other problems, it could be a useful first step. A logistic regression was found to be a suitable method for this, but other approaches could be interesting to try out as well.

S2) *Which cases should be selected to be used in AF-CBA?*

Given all training data, we have to determine which instances to include in the case base. The right selection of cases is dependent on the goal of the application.

The goal of AF-CBA is to answer the question: is the prediction justified? In order to answer this question, the system aims to base the argumentation on the best examples that can be found in the training data. When such a system relies on individual examples, it is hindered by the appearance of unexpected outcomes within the training data. We see two possibilities to tackle this problem:

1. Filter the case base
2. Require the system to use multiple cases as evidence

In this work, we investigated the first option by applying an algorithm that automatically filtered the case base. The algorithm removes the most inconsistent cases from the case base until all inconsistencies are removed. The first results with this approach seem promising; the final performances on the Churn and Admission data sets improved for all case-based classifiers.

S3) *How to apply downplaying moves in AF-CBA?*

In the argument game of Prakken (2020), downplaying moves are ways of arguing that the differences between a focus case and precedent do not matter or can be compensated. He distinguishes between different types of downplaying moves: four for factors and one for dimensions. His formalization allows for multiple orders of applying downplaying moves, possibly leading to different results.

Instead of tackling the problem of order, we changed the structure of argumentation. The main reason for that was that the distinction between factors and dimensions appeared unnecessary and caused problems in the reasoning. We decided to treat factors equal to dimensions, which left us with only a single downplaying move - removing the problem of order.

In addition, we argue that the success of defending a precedent should depend on its relative strength compared to other available precedents, counterexamples included. We used the automatically generated feature information to weight the strength of downplaying moves. Using this strength ordering, we enabled stronger downplaying moves to attack others. In this way, only the strongest downplaying moves in the game can become part of the grounded extension.

In later experiments, we moved away from the formal structure of argumentation frameworks. While we no longer used argumentation frameworks, the idea of downplaying moves remained relevant; (how) can cases compensate for their shortcomings as examples?

In our experiments, we did not find much evidence for the success of the concept of compensation. The *Nearest neighbor* and *Minimize negative* algorithms, which do not apply the concept of compensation, performed best. As stated at the first sub-question, our method did not take any relations between the features into account; all positive differences were assumed to be able to compensate for all sorts of negative differences. Another approach that specifies specific compensation relations between features might work out differently.

S4) *What are the possibilities for closing the gap between AF-CBA and the prediction model?*

We distinguished between two approaches for eliminating this gap: removing (S4.1) or monitoring (S4.2) the black box.

S4.1) *Have interpretable CBA-systems potential for taking over black-box models?*

Whenever possible in terms of accuracy, it is a cleaner solution to replace a black box, then to explain it. When the explanation system and classifier become one, the complete process becomes more transparent.

In our experiments, we saw that the CBA-systems performed comparably in terms of accuracy to mainstream classifiers. We can, therefore, conclude that black boxes could well be replaced by CBA-systems. However, on our data

sets, other interpretable systems were also able to perform well. The potential of case-based classifiers to replace black boxes on more complex problems can, therefore, not be assessed based on our research. This also applies to the potential of AA-CBR, as the Mushroom data set did not appear to be a suitable test for the system.

S4.2) *Are there model-agnostic possibilities to incorporate further information about the black-box model in CBA?*

The work of Weerts, Ipenburg, and Pechenizkiy (2019) inspired us to include a new source of information in the system: the predictions of the black box on the training data. Using this information, the reasoning of the CBA-system can be directed towards the black box. Displaying this information to users in a two-dimensional plot was received well in the user experiment. The incorporation of this information in the argumentation of the explanation system is left to further research.

S5) *To what extent are the explanations generated by the model suitable for users?*

In terms of the criteria specified by Miller (2019), the suitability for users can be improved. The explanations do have a contrastive set-up, but the user cannot pose contrastive questions itself. In addition, the systems do not adjust the displayed information to the knowledge of the user. Lastly, the user experiment made clear that the large amount of information presented by the CBA-systems was difficult to process fully; this gives the impression that the explanations should be more selective.

The user experiment showed that the justification- and monitor system were considered more insightful than the absence of an explanation. The sample rated the monitor system higher on the scales of convenience, trust and insight than the justification system. Conversations with the users made clear that the justification system was considered unintuitive by some. The system minimized negative differences while maximizing positive ones. This allowed the system to select examples with few similarities with the focus case, as long as their differences were positive. The dissimilarity between the focus- and example case evoked distrust by some users.

With the small number of tasks - four per presentation system - no difference could be demonstrated between the task performance using the explanation systems. The results seem to hint at a better performance using the monitor

system, but further research must reveal whether that is the case.

RQ) *To what extent is CBA applicable for making machine learning more interpretable?*

CBA offers broad possibilities to machine-learning interpretability. In this work, we moved from testing the concrete formal approach of Prakken (2020) to taking a broader perspective on the possibilities of CBA. We distinguished between two roles for CBA-systems: justifying or monitoring the predictions of a classifier.

The justification approach comes with some challenges. When it is applied as an explanation system for a black box, the construction feels artificial; the predictions of the black box are explained by presenting how an alternative, *white-box* classifier would come to the same - or different - conclusion. The added value of such a design does not speak for itself and needs further experimentation. A cleaner solution would be to use the justification system as a classifier, justifying its own predictions to the user. Further research could investigate whether there are problems for which CBA approaches proof more suitable than the usage of other white-box classifiers.

The monitor system - inspired by the work of Weerts, Ipenburg, and Pechenizkiy (2019) - proved to be an interesting alternative. This system enables the user to analyze the trustworthiness of predictions of the black box based on the local feature space. Instead of functioning as a separate classifier, it tries to uncover the strengths and vulnerabilities of the black box. The system could be valuable in assisting human judgment in high-stake decisions, exposing potential weaknesses of the prediction model.

9.2 Future work

Many suggestions for future work have been made already. In relation to the justification- and classification system, our first recommendation would be to further investigate the suitability of concepts underlying the reasoning about differences and similarities between data instances. Possibly, the addition of extra information about the features could prove useful.

For the monitor system, we would recommend experimenting with presenting different numbers of cases. This could be done in an interactive manner, allow-

ing the user to ask for specific information. Besides, it would be interesting to establish whether some explicit form of reasoning could be generated that can assist the user. Finally, further user experimentation can identify the consequences of the system on human decision making.

Bibliography

Acharya, Mohan S., Asfia Armaan, and Aneeta S. Antony (2019). "A Comparison of Regression Models for Prediction of Graduate Admissions". In: *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*. DOI: 10.1109/iccids.2019.8862140.

Adadi, Amina and Mohammed Berrada (2018). "Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6, pp. 52138–52160.

Cocarascu, Oana, Kristijonas Cyras, and Francesca Toni (2018). *Explanatory predictions with artificial neural networks and argumentation*.

Cunningham, Pádraig, Dónal Doyle, and John Loughrey (2003). "An evaluation of the usefulness of case-based explanation". In: *5th International Conference on Case-Based Reasoning*. Springer, pp. 122–130.

Čyras, Kristijonas, David Birch, Yike Guo, Francesca Toni, Rajvinder Dulay, Sally Turvey, Daniel Greenberg, and Tharindi Hapuarachchi (2019). "Explanations by arbitrated argumentative dispute". In: *Expert Systems with Applications* 127, pp. 141–156.

Cyras, Kristijonas, Ken Satoh, and Francesca Toni (2016). "Abstract argumentation for case-based reasoning". In: *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Das, Kajaree and Rabi Narayan Behera (2017). "A survey on machine learning: concept, algorithms and applications". In: *International Journal of Innovative Research in Computer and Communication Engineering* 5.2, pp. 1301–1309.

Dash, Manoranjan and Huan Liu (1997). "Feature selection for classification". In: *Intelligent data analysis* 1.3, pp. 131–156.

Doshi-Velez, Finale and Been Kim (2017). "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608*.

Doyle, Dónal, Pádraig Cunningham, Derek Bridge, and Yusof Rahman (2004). "Explanation oriented retrieval". In: *7th European Conference on Case-Based Reasoning*. Springer, pp. 157–168.

Dua, Dheeru and Casey Graff (2019). *UCI Machine Learning Repository*. URL: http://archive.ics.uci.edu/ml.

Dung, Phan Minh (1995). "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games". In: *Artificial intelligence* 77.2, pp. 321–357.

Edwards, Lilian and Michael Veale (2017). "Slave to the algorithm: Why a right to an explanation is probably not the remedy you are looking for". In: *Duke L. & Tech. Rev.* 16, pp. 1–65.

Freitas, Alex A. (2019). "Automated machine learning for studying the trade-off between predictive accuracy and interpretability". In: *3th International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, pp. 48–66.

Friedman, Jerome H. (2001). "Greedy function approximation: a gradient boosting machine". In: *Annals of Statistics*, pp. 1189–1232.

Funke, Frederik and Ulf-Dietrich Reips (2012). "Why semantic differentials in web-based research should be made from visual analogue scales and not from 5-point scales". In: *Field methods* 24.3, pp. 310–327.

Gentner, Dedre, Jeffrey Loewenstein, and Leigh Thompson (2003). "Learning and transfer: A general role for analogical encoding." In: *Journal of Educational Psychology* 95.2, pp. 393–408.

Gerich, Joachim (2007). "Visual analogue scales for mode-independent measurement in self-administered questionnaires". In: *Behavior Research Methods* 39.4, pp. 985–992.

Gibson, Helen, Joe Faith, and Paul Vickers (2013). "A survey of two-dimensional graph layout techniques for information visualisation". In: *Information visualization* 12.3-4, pp. 324–357.

Guidotti, Riccardo, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi (2019). "A survey of methods for explaining black box models". In: *ACM Computing Surveys* 51.5, 93:1–93:42.

Heider, Fritz and Marianne Simmel (1944). "An experimental study of apparent behavior". In: *The American Journal of Psychology* 57.2, pp. 243–259.

Horty, John (2011). "Rules and reasons in the theory of precedent". In: *Legal Theory* 17.1, pp. 1–33.

— (2019). "Reasoning with dimensions and magnitudes". In: *Artificial Intelligence and Law* 27.3, pp. 309–345.

Karacapilidis, Nikos, Brigitte Trousse, and Dimitris Papadias (1997). "Using case-based reasoning for argumentation with multiple viewpoints". In: *International Conference on Case-Based Reasoning*. Springer, pp. 541–552.

Kim, Been, Rajiv Khanna, and Oluwasanmi O Koyejo (2016). "Examples are not enough, learn to criticize! criticism for interpretability". In: *Advances in Neural Information Processing Systems*. Vol. 29, pp. 2280–2288.

Kolodner, Janet L. (1992). "An introduction to case-based reasoning". In: *Artificial Intelligence Review* 6.1, pp. 3–34.

Lipton, Peter (1990). "Contrastive explanation". In: *Royal Institute of Philosophy Supplements* 27, pp. 247–266.

Lipton, Zachary C. (2018). "The mythos of model interpretability". In: *Queue* 16.3, pp. 31–57.

Lundberg, Scott M. and Su-In Lee (2017). "A unified approach to interpreting model predictions". In: *Advances in Neural Information Processing Systems*. Vol. 30, pp. 4765–4774.

Malle, Bertram F. (2006). *How the mind explains behavior: Folk explanations, meaning, and social interaction*. Mit Press.

Miller, Tim (2019). "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial Intelligence* 267, pp. 1–38.

Mitchell, Tom Michael (2006). *The discipline of machine learning*. Vol. 9. Carnegie Mellon University, School of Computer Science, Machine Learning.

Modgil, Sanjay and Martin Caminada (2009). "Proof theories and algorithms for abstract argumentation frameworks". In: *Argumentation in artificial intelligence*. Springer, pp. 105–129.

Molnar, Christoph (2019). *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/.

Prakken, Henry (2019). "Comparing Alternative Factor-and Precedent-Based Accounts of Precedential Constraint." In: *JURIX*, pp. 73–82.

— (2020). "A top-level model of case-based argumentation for explanation". In: *Proceedings of the ECAI 2020 Workshop on Dialogue, Explanation and Argumentation for Human-Agent Interaction (DEXA HAI 2020)*. to appear.

Rathi, Shubham (2019). "Generating Counterfactual and Contrastive Explanations using SHAP". In: *arXiv preprint arXiv:1906.09293*.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). "Model-agnostic interpretability of machine learning". In: *arXiv preprint arXiv:1606.05386*.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2018). "Anchors: High-precision model-agnostic explanations". In: *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 1527–1535.

Robeer, Marcel Jurriaan (2018). "Contrastive explanation for machine learning". Utrecht University. MA thesis.

Rudin, Cynthia (2019). "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature Machine Intelligence* 1.5, pp. 206–215.

Sørmo, Frode, Jörg Cassens, and Agnar Aamodt (2005). "Explanation in case-based reasoning–perspectives and goals". In: *Artificial Intelligence Review* 24.2, pp. 109–143.

Telco Customer Churn (2018). https://www.kaggle.com/blastchar/telco-customer-churn. Version 1.

Tetlock, Philip E. and Richard Boettger (1989). "Accountability: A social magnifier of the dilution effect." In: *Journal of Personality and Social Psychology* 57.3, pp. 388–398.

Wachter, Sandra, Brent Mittelstadt, and Chris Russell (2017). "Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GPDR". In: *Harv. JL & Tech.* 31, pp. 841–892.

Weerts, Hilde JP, Werner van Ipenburg, and Mykola Pechenizkiy (2019). "Case-Based Reasoning for Assisting Domain Experts in Processing Fraud Alerts of Black-Box Machine Learning Models". In: *arXiv preprint arXiv:1907.03334*.

Wieringa, Roel J. (2014). *Design science methodology for information systems and software engineering*. Springer.

Yao, Quanming, Mengshuo Wang, Hugo Jair Escalante, Isabelle Guyon, Yi-Qi Hu, Yu-Feng Li, Wei-Wei Tu, Qiang Yang, and Yang Yu (2018). "Taking Human out of Learning Applications: A Survey on Automated Machine Learning". In: *arXiv preprint arXiv:1810.13306*.

A. Details user experiment

A.1 Informed Consent

You are invited to participate in a research project about Explainable Artificial Intelligence. This experiment should take about 40 minutes to complete. Responses will be kept confidential and will be processed anonymously. Participation is voluntary and you have the option to quit the experiment at any point. If you have any questions about the research, please contact the researcher Rosa Ratsma (*mail address*).

A.2 Explanation experiment

In this experiment, you play the role of a telecommunication employee. Your task in the company is to decrease the number of customers that churns: cancels their subscription. To achieve this, you can offer customers a special discount. However, giving discounts to every customer is too costly for your company. To maximize your client retention, while minimizing your costs, you want to estimate which clients have the highest risk of churning next month. These are the clients you want to contact to offer the discount.

During this experiment, you are in the first stage of this process: you want to estimate which of the customers is at high risk of churning. Profiles of customers will be presented to you, after which you can indicate on a scale how likely it seems to you that these customers will churn.

You are not on your own. There is a prediction model that will assist you during this task. This model has learned from data about previous customers. It can

predict outcomes for new customers, and these predictions are correct about 80% of the time.

The experiment consists of four sections. The sections differ in the kind, and the amount of information that you can use to make your estimation. After every section, you will be asked to indicate how convenient you experienced your task during that section and how insightful and trustworthy the information was to you.

A.3 Explanations of the experimental conditions

Condition 1: No explanation

In this section, you will only get two sources of information: a profile of a customer and the prediction of the model for this customer.

Condition 2: Feature importance

In this section, you will be shown the profiles of the customers together with colored lines indicating the impact that the properties seem to have on customers decisions.

Condition 3: Justification system

In this section, another system will try to assist you by comparing the current customer with previous customers whose decision we already know. The system will select two previous customers: one that stayed and one that churned. Based on the comparison between the current customer and the previous customers, the system will try to help you judge whether the prediction of the model seems convincing.

Condition 4: Monitor system

In this section, we will show you comparisons with previous customers whose decision we already know. You will not only see whether these previous customers churned or stayed, but also whether the model could predict their decision correctly.

The five customers that are most similar to the current customer will be visualized in a plot. The distance between customers in the plot represents the similarity; the closer the more similar. If you click on a customer in the plot, the comparison with the current customer will be shown in the right part of the screen.

Text used after every condition explanation

You can switch between the information-page and question-page with the left and right arrow of your keyboard. Let's first practice with one example.

A.4 Templates justification system

Below the templates used by the justification system in the user experiment. The bold text specifies the scenario in which the template is applied. The nouns in *italics* are replaced by the names, outcomes and pronouns of the specific cases, when the explanation is presented to a user.

Both have differences, winner more positive, less negative: We looked up two previous customers as examples for *focus*: one who churned, one who stayed. The comparison of *focus* with these customers, seems to suggest that it is more likely that *he/she* will make the same decision as *winner* and *winner-outcome*.

Although *focus* has differences with both customers that make *him/her* less likely to imitate their decisions, these differences seem less important and can better be compensated by other properties in the comparison with *winner*.

Both have differences, winner more positive and more negative: We looked up two previous customers as examples for *focus*: one who churned, one who stayed. The comparison of *focus* with these customers, seems to suggest that it is somewhat more likely that *he/she* will make the same decision as *winner* and

winner-outcome.

Although *focus* has differences with both customers that make *him/her* less likely to imitate their decisions, these differences can best be compensated by other properties in the comparison with *winner*.

Both have differences, winner less positive and less negative: We looked up two previous customers as examples for *focus*: one who churned, one who stayed. The comparison of *focus* with these customers, seems to suggest that it is somewhat more likely that *he/she* will make the same decision as *winner* and *winner-outcome*.

Although *focus* has differences with both customers that make *him/her* less likely to imitate their decisions, these differences seem less important in the comparison with *winner*.

Winner is equal to focus: We looked up two previous customers as examples for *focus*: one who churned, one who stayed. The comparison of *focus* with these customers, seems to suggest that it is most likely that *he/she* will make the same decision as *winner* and *winner-outcome*. *winner* has exactly the same profile as *focus*.

loser, on the other hand, has differences with *focus* that make *focus* less likely to imitate the decision of *loser* to *loser-outcome*.

Winner has positive and no negative differences: We looked up two previous customers as examples for *focus*: one who churned, one who stayed. The comparison of *focus* with these customers, seems to suggest that it is most likely that *he/she* will make the same decision as *winner* and *winner-outcome*. All differences between *focus* and *winner* make *focus* even more likely to *winner-outcome*.

loser, on the other hand, has properties which make *him/her* more likely to *loser-outcome* than *focus*.

A.5 Survey questions

1. What is your age?
2. What gender do you identify as?
   (a) Female

   (b) Male
   (c) Prefer not so say
   (d) Other ...
3. What is the highest degree or level of education that you have completed?
   (a) Primary school
   (b) High school
   (c) Bachelor's Degree
   (d) Master's Degree
   (e) PhD or higher
   (f) Prefer not to say
   (g) Other ...
4. How familiar are you with machine learning?
   (a) Not at all
   (b) A little bit
   (c) Somewhat
   (d) Very
5. How familiar are you with explaining machine learning?
   (a) Not at all
   (b) A little bit
   (c) Somewhat
   (d) Very

B. Details computer experiments

B.1 Classifiers

The following classifiers from the Python *sklearn* library were used:

1. *DecisionTreeClassifier()*
2. *SVC(kernel='linear')*
3. *GaussianNB()*
4. *LogisticRegression(solver = 'lbfgs')*
5. *AdaBoostClassifier()*

B.2 Feature selection

Below the resulting selection of features after running the feature selection algorithm are presented per data set.

Mushroom

odor_a, odor_c, odor_f, odor_l, odor_n, odor_p, gill-size_b, gill-size_n, gill-color_b, stalk-surface-above-ring_k, stalk-surface-below-ring_y, ring-type_f, spore-print-color_k, spore-print-color_n, spore-print-color_r, spore-print-color_u, population_c

Churn

tenure, MonthlyCharges, InternetService_Fiber optic, Contract_Month-to-month

Admission

GRE Score, TOEFL Score, LOR , CGPA

B.3 Digital Appendix

Data sets and Python files can be found on Dropbox using the following link:
https://www.dropbox.com/sh/ghw9qjpvw7collc/AAB5Nb-9j_6vJFWSUHa8VSz1a?dl=0