ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Master Thesis: Data Science and Marketing Analytics

Interpretable Machine Learning for Attribution Modeling

A Machine Learning Approach for Conversion Attribution in Digital Marketing

Student name: Jordy Martodipoetro
Student number: 454072

Supervisor: Dr. Kathrin Gruber
Second assessor: Prof. Bas Donkers

Date final version: 15 July 2021

Abstract

This thesis explores the possibility of using a more complex machine learning approach combined with model interpretability methods to solve the attribution problem. Criteria are formulated to show how the current leading attribution methodologies, such as last-touch attribution, simple probabilistic attribution, Shapley value attribution, and logistic regression, are dominant because of their inherent algorithmic interpretability. The classifier XGBoost (eXtreme Gradient Boosting) is proposed to be used in conjunction with SHAP (Shapley Additive exPlanations) to create an attribution model. Data from a Dutch financial provider is used to create the attribution model. The XGBoost model is compared against a logistic regression model on predictive and explanatory power. Within the final section of this thesis, the predictive performance is compared. It was observed that XGBoost has a better predictive performance as opposed to logistic regression. The explanatory performance of the XGBoost + SHAP method was benchmarked against last-touch attribution, logistic regression coefficient interpretation, and logistic regression + SHAP. The most important features were consistent among the three methods.

Acknowledgements

I would like to express my gratitude to Dr. Gruber from the Erasmus School of Economics, Erasmus University Rotterdam, for all her fantastic support and help during the whole process of the thesis. It was a genuine pleasure to have her supervision, as it would not have been possible without her guidance. Even though the meetings were sometimes not as structured as I had planned, the sessions we had were really insightful. Furthermore, I would like the Dutch financial insurer for the opportunity to conduct my research for the company. It allowed me to work on a hands-on problem that can add value to the firm. I like to thank my company supervisor, with whom I could always brainstorm every Wednesday of the week, and for taking the time to have in-depth discussions on the topics discussed in this thesis. Lastly, I would like to thank my family and friends for their support during the past months while writing this thesis. It would have been much more difficult without them behind my back.

Keywords

Attribution Modeling, Digital Marketing, Conversion Attribution, Machine Learning, Explainable Machine Learning, Interpretation, Gradient Boosting, XGBoost, Logistic Regression, Shapley Value, SHAP, Insurance.

Contents

1 Introduction

1.1 Motivation

Nielsen (2021) recently reported that 2020 was the year that online advertising spending exceeded the spending for offline advertising in the Dutch advertising market for the first time, of which 52 percent was spent on online advertising. Despite the 6.3 percent decrease to 4.4 billion euros in the total media spending in 2020 as opposed to 2019 in the Netherlands, online advertising has still grown by 7 percent compared to offline advertising (Deloitte, 2021). It is expected that the total Dutch advertising spending will grow by 12 percent on average by 2022, in which the fraction of spending on online advertising is expected to grow to 64 percent (Magna, 2021).

The shift from offline to online advertising is accelerated due to the drawbacks of offline advertising. In offline advertising, it is difficult to target a certain audience resulting in less personalized advertising content. It is also an intricate process to determine the effectiveness of the advertisement on the audience as there is no tracking behavior that allows marketers to see the exposure of the offline advertisements on the audience. Online advertising, on the other hand, can track the users on the internet in their click behavior. It allows the advertisers to evaluate the online advertisements or the channels on which the advertisements are shown to internet users. However, determining the contribution of channels on the actual sale, also known as a conversion, is rather challenging to do in practice. The problem of researching the contribution of advertising campaigns towards conversion is referred to in the marketing literature as the *attribution problem* (Shao & Li, 2011).

In the past decade, initial efforts have been made to solve the attribution problem by setting up attribution methodologies based on intuition and heuristics. The most popular used attribution methodology is the last-touch attribution method, in which the last channel a visitor interacted with is given all of the credits. Nonetheless, these efforts do not yield adequate results. Rule-based attribution methodologies presume a particular (a priori) distribution regarding the contribution in a set of channels the visitor has interacted with. There is a consensus in marketing literature that these rule-based approaches do not account for important interaction effects amongst the various channels an internet user interacted with (Anderl, Becker, von Wangenheim, & Schumann, 2016; Dalessandro, Perlich, Stitelman, & Provost, 2012; Kannan & Li, 2017; Shao & Li, 2011).

As more and refined data is stored on the interactions an internet user has with an advertiser, marketers move away from rule-based attribution methodologies by addressing the attribution problem via the use of data. This had led to scholars proposing the use of simple data-driven models, such as logistic regression, probabilistic approaches, and the Shapley value for attribution modeling. Each of these methods differs in its ways of solving the attribution problem. However, the methodologies all have in common that the model itself is highly interpretable. The output yielded by a model needs the ability to be understandable terms to humans that allows for verifying

whether its reasoning is valid.

Consequently, the efforts made in the current literature focus on interpretable machine learning methods, mainly the ones named previously, that are proposed to address the attribution problem. However, it is important to note that higher interpretability often leads to sacrificing the model's accuracy. More complex machine learning models, such as gradient boosting and neural networks, often provide greater accuracy with the major disadvantage of lower interpretability. In the field of machine learning, the literature on the problem of accuracy sacrifice for higher accuracy is growing. Scholars in this field have been proposing methods, such as, SHAP that allow for the use of complex machine learning models while still providing interpretability.

1.2 Objective of this research

This thesis aims to add a novel way for creating attribution models that combines a high degree of accuracy and interpretability to the growing marketing literature implemented in Python. The data used in the thesis is obtained from a Dutch financial provider that sells car insurance in the Netherlands. For the empirical part, a specific gradient boosting algorithmic called XGBoost will be used to be compared against a simple algorithmic approach, which is logistic regression, for classification performance. The interpretability method to be used is SHAP on both XGBoost and the logistic regression and will be put against the built-in interpretation approach for logistic regression.

1.3 Structure of the thesis

The remainder of the thesis is structured as follows. Chapter 2 gives an overview of the literature focused on attribution modeling in which topics such as attribution modeling, rule-based models, data-driven models, and explainable machine learning are discussed. Chapter 3 briefly discusses the data obtained from the Dutch financial provider used for training and evaluating the models. The focus in Chapter 4 is on the methodology used in this research, where an extensive description is given of the machine learning algorithms. Chapter 4 also dives into the empirical evaluation of these methodologies. In Chapter 5, the predictive performance of the models is compared, and the feature importances are compared. Finally, Chapter 6 concludes this research on attribution modeling using machine learning with the added explainability.

2 Theoretical background

This chapter dives deeper into the theory and literature of digital marketing and attribution modeling. Section 2.1 discusses what digital marketing is. Then, Section 2.2 looks into the attribution problem and defines what attribution modeling is. Furthermore, criteria to which an attribution model should satisfy will be defined. Section 2.3 existing attribution models will be touched upon and evaluated based on the defined criteria. Lastly, Section 2.4 introduces the concept of interpretable machine learning.

2.1 Digital marketing

In the last decade, the use of digital media has become so widespread that consumers can quickly and instantaneously get access to information via the internet. According to recent statistics published by Statista (2021), 4.66 billion people use the internet to access information, 59.5 percent of the global population. This new media type has facilitated the transition of bringing conventional marketing to the digital landscape (Bughin, 2015). The result of this shift towards the digital landscape is known as digital marketing. Digital marketing differs from traditional marketing because online channels create, communicate, and deliver value to the desired target audience.

2.1.1 Digital advertising

In a global advertising report published by Letang & Stillman (2020), it is shown that digital advertising has a more significant market share of the total advertising revenues globally as opposed to non-digital advertising. Digital advertising allows a marketer to perform mass-personalization on its advertisements and is a subset of digital marketing (Durai & King, 2015; Jiang & Benbasat, 2007). *Advertising personalization* is defined as a firm-initiated action in which the advertising content, message, or visual representation is geared towards the preferences of a specific grouped audience or an individual (Arora et al., 2008). The personalization of advertisements allows marketers to target the right audience, at the right time, at the right place and is, thus, more relevant to that corresponding audience. Greater digital ad relevancy has been linked positively to a higher probability of a consumer turning into a lead or conversion (Bleier & Eisenbeiss, 2015; Hayes, Golan, Britt, & Applequist, 2020).

2.1.2 Online marketing channels

Advertisements are shown to the customer via a certain channel, and the action to be undertaken by the customer linked to the advertisement is the touchpoint. For instance, when a consumer sees an advertisement on a social media platform, the platform itself is seen as the channel where a

firm and its customers can interact (Neslin et al., 2006). Typical digital marketing channels used for digital advertising include search engines, display, email, online video, social media, affiliates, and price comparisons (de Haan, Wiesel, & Pauwels, 2016). In the literature, there is a consensus on the categorization of these channels using the origin of the interaction (Anderl, Becker, von Wangenheim, & Schumann, 2016; de Haan, Wiesel, & Pauwels, 2016; Kannan & Li, 2017). A customer can interact with a channel by its own initiative or by the initiative from the brand. Consider a customer that directly visits the brand's website directly by typing in the URL in the browser bar – this type of contact is customer-initiated. Whereas, for example, showing an advertisement to a customer using social media is seen as firm-initiated contact. The advertisements also contain a messaged action to be undertaken by the customer, for example, signing up for a newsletter on the brand's website, and this is considered a touchpoint since it is an immediate goal with the channels website and social media seen as the facilitators (Hallikainen, Alamäki, & Laukkanen, 2019).

2.1.3 Online customer journey

The shift from conventional marketing to digital marketing also enables marketers to instantly track responses from the consumer who interacted with the brand via an advertisement (Kannan & Li, 2017; Shao & Li, 2011). Brands, also seen as advertisers, employ different channels to reach customers via the internet. For each interaction the consumer has with the brand, data is gathered on the channel, the timestamp the customer interacted with the channel, and whether the customer has undertaken proceeding interactions. All these recorded interactions with the channels are then used to construct the online customer journey. Online customer journeys describe the interactions with the channels using the click pattern on an individual level before the purchase (conversion) or non-purchase (non-conversion) (Anderl, Becker, von Wangenheim, & Schumann, 2016). It is also important to note that the online customer journey can also contain interactions that are not initiated by the firm but initiated by the customer. A customer could, for instance, type in the URL of the brand and visit the website (Kannan & Li, 2017).

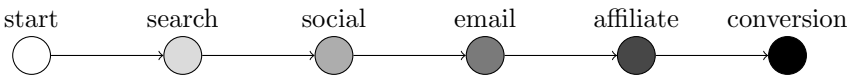


Figure 1: Graphical example of the constructed online customer journey

In Figure 1, a graphical representation of a constructed customer journey using click pattern data is used as an example that contains four interactions - excluding the start and end node. It can be observed that the first channel the customer had interaction with was a search result displayed in the search engine, followed by a social media advertisement shown by the social media channel.

Subsequently, the customer received an email from the brand and eventually interacted with an affiliate. The combination and sequence of these customer journey interactions have led to a conversion in this example. The example used in Figure 1 shows a relatively linear multi-stage customer journey indicated by interactions with touchpoints, as the journey is solely based on online click behavior. However, this constructed customer journey does not include the phases in which the customer moves during the decision-making process (Lemon & Verhoef, 2016). In the broadest sense, customer journeys include three general stages that are consistent in the marketing literature (Howard & Sheth, 1970; Lemon & Verhoef, 2016; Neslin et al., 2006). The first stage, prepurchase, embodies all customer interactions with the brand before a conversion occurs. The prepurchase stage is characterized by behavior, such as need recognition, search, and consideration. The second stage, purchase, includes behavior, such as choice, purchasing, and payment. The last stage, postpurchase, is characterized by usage, engagement, and service. According to Lemon & Verhoef (2016) the prepurchase and purchase stage are the most important stages for marketers. They imply that marketers should focus on identifying specific channels that lead customers to continue or discontinue these stages in the customer journey.

It is essential to point out that it is also possible that the customer is exposed to offline advertising, such as TV or outside bannering advertisements, and, therefore, interacts with the brand. However, the effects cannot be measured on an individual level as there is no means to track the exposure accurately other than estimation. Hence, this thesis only focuses on the online component of the customer journey. This implies that all online interactions prior to the (non-)conversion are solely taken into account.

2.2 Attribution modeling

In the previous section, it was discussed how digital marketing allows for constructing the online customer journey from the click pattern data. This tracking ability has resulted in growing popularity for attribution models. From these customer journeys, the credit of each interaction with a channel is assigned to a conversion on an individual level. Understanding the contribution of each channel on the decision to purchase or not to purchase seen from the perspective of the customer allows marketers to analyze, report, and optimize an advertising campaign (Dalessandro, Perlich, Stitelman, & Provost, 2012; Shao & Li, 2011).

The process of assigning credits to multiple channels on a user level in a customer journey is known as attribution modeling. It tries to estimate the effect of an intervention – exposure to an advertisement – has on the conversion. The estimation of the effect is done via observing the click pattern, thus, a descriptive analysis. Quantifying the degree to which a channel contributes to a conversion is a demanding task due to the rise in online channels and the complexity of online journeys (Anderl, Becker, von Wangenheim, & Schumann, 2016). The attribution of each channel can be expressed as a percentage of the total amount of conversions by channel or as an absolute

number of conversions by channel. Figure 2 is a graphical representation of the credit assignment process in attribution modeling.

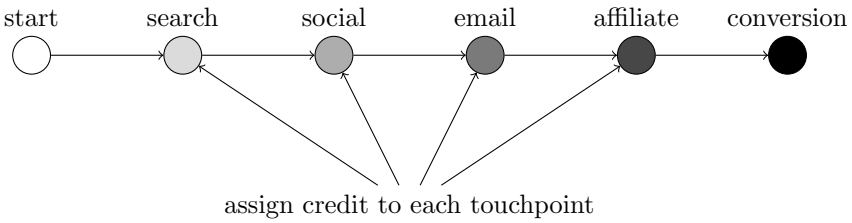


Figure 2: Graphical example of the credit assignment in attribution modeling

2.2.1 Attribution model criteria

In the field of attribution modeling, there are currently lots of attribution techniques available for use, raising the question on which of these techniques attribute the 'true' accurate contribution to a touchpoint. This fundamental question has led to scholars being concerned with formulating adequate criteria to determine what a 'good' attribution model includes. It is important to pinpoint that the relative importance of each touchpoint is inherently unobservable as there is no evaluation method on how close the generated outcome generated by attribution modeling comes to the reality of it (Dalessandro, Perlich, Stitelman, & Provost, 2012; Kelly, Vaver, & Koehler, 2018; Singh, Vaver, Little, & Fan, 2018). Hence, the topic of attribution modeling is inevitably subjective to an extent.

Nevertheless, in the literature, efforts are still made to formulate criteria. Shao & Li (2011) put forward that a 'good' attribution model should have a high degree of accuracy. The accuracy of a model indicates how correctly a model can classify whether a customer ends in a conversion or a non-conversion. The authors measure the accuracy based on the out-of-sample misclassification error rate – a high accuracy implies a low out-of-sample misclassification error rate. However, as previously said, an attribution model is a descriptive method to show how much a channel contributes to a conversion, and therefore, the main goal is not to predict (Anderl, Becker, von Wangenheim, & Schumann, 2016). Nonetheless, it can be argued that attribution modeling shows how much a channel contributes to conversion; these insights could be used in the binary classification prediction of a conversion.

Dalessandro, Perlich, Stitelman, & Provost (2012) build further on the criteria for attribution by adding that attribution models should include fairness, data-drivenness, and interpretability. Fairness reflects the ability of the model to attribute a channel according to its influence to affect the likelihood of the conversion – this is inherently a causal attribution problem as an intervention (exposure to an advertisement) is expected to impact the outcome of interest (Kelly, Vaver, & Koehler, 2018). A problem with this criteria is that the actual advertisement exposure is not

known to the advertiser. Kelly, Vaver, & Koehler (2018) and Singh, Vaver, Little, & Fan (2018) argue that fairness is theoretically only achievable by doing a real-world experiment but states this is a very costly and impractical method. Data-driven refers to the model being based on data in which the distribution of the conversion is not a priori determined. Sapp & Vaver (2016) underpin this notion by the need to also consider non-converting paths in an attribution model. By taking non-converting paths in the equation, it is implicitly assumed that advertising can also have adverse effects on the outcome. This is a reasonable thought, as there can be instances where the effect of an advertisement may cause a visitor not to convert. Lastly, interpretability refers to the model being generally accepted by all relevant parties based on statistical merit and intuitive understanding, according to Dalessandro, Perlich, Stitelman, & Provost (2012). Interpretability of the model is expressed by understanding the effects of advertisement on an individual level (local explanation) as it is a means to know how each interaction with an advertisement contributes to a conversion. The global importance should be consistent with the local importance. According to Shao & Li (2011) this also helps with the interpretability as consistent results are easier to accept by marketers. The criteria found in the literature can be summarized as follows:

1. **Ability to predict**. An attribution model should be able to accurately predict based on a customer journey prior to ending whether the path will end in a conversion or a non-conversion. The ability to predict gives an objective measure on the evaluation of the empirical performance of the model.
2. **Data-driven**. The model should be based on an algorithmic data-driven approach and not make use of a priori determined distribution of the weights of the channels. A priori based credit assignment is biased as it does not use the objective distribution of the conversion that is available in the data.
3. **All outcomes are considered**. The model should not be solely based on converting customer journey paths but also on non-converting journey paths to see which channels encourage or discourage the customers to end their journeys in a conversion or non-conversion.
4. **Fair attribution**. Estimates on the contribution of each channel to a conversion should be in accordance with the actual influence of the channel to affect the likelihood of conversion. Although it is very difficult to evaluate the causal effects in a descriptive model, from a conceptual viewpoint this is a desirable criterion.
5. **Interpretable**. An attribution model should be easily able to interpret and generally accepted. The model should be able to assign credit to an individual journey and to aggregate the credits on global level with clear interpretation.

2.3 Attribution models

Before the models are introduced, it is useful to define the mathematical notation used in this thesis. Let $i = \{1, 2, 3, ..., N\}$ denote the visitors that have interacted with the advertiser. A visitor

interaction with a channel is denoted as x_j for $j = \{1, 2, 3, ..., M\}$ channels. An interaction of visitor i with channel j is denoted by x_{ij}. The visitor can either end up in a conversion or non-conversion, which is represented by:

$$y_i = \begin{cases} 1 & \text{if visitor } i \text{ converts} \\ 0 & \text{else} \end{cases}$$

and

$$x_{ij} = \begin{cases} 1 & \text{if channel } j \text{ is present in the customer journey of customer } i \\ 0 & \text{else} \end{cases}$$

2.3.1 Heuristic-based attribution

Heuristic-based attribution is based on a non-statistical method to perform multi-credit assignment to channels. In other words, a weight distribution that is determined beforehand is used to determine the contribution each channel has on the conversion. The most popular heuristic-based attribution approaches will be discussed in this subsection.

2.3.1.1 Last-touch attribution The most common attribution method is last-touch attribution, which assigns all of the weight to the last channel. In this sense, it is making the last channel the customer interacted in the whole customer journey the most important. Last-touch attribution is intuitive as it can be argued that the last interaction as a customer has is only one step away from the conversion. However, it completely ignores the prior interactions of the customer, making it fundamentally flawed as it is positively biased towards the last interaction.

2.3.1.2 Linear touch attribution Another common attribution method is linear touch attribution, which assigns equal weights to all the channels that are present in the customer journey. It is less fundamentally flawed as it takes all the channels into account, but the weight that is assigned to each channel is still arbitrarily determined as it is not based on the actual underlying effect each channel has on the conversion.

2.3.1.3 Time decay attribution One of the other common attribution methods is time decay attribution, which assigns decaying weights to the channels that the customer has interacted with. This means that the weight is the lowest for the interaction in the beginning, and the weight

gradually increases towards the latest interaction. Time decay attribution takes all channels into account, but the weights, nevertheless, are also arbitrarily determined.

It can be concluded that despite these heuristic-based approached being computationally inexpensive to compute, the general disadvantage of using heuristics is since all of the methods assume an a priori distribution regarding the weights, none of the methods are data-driven. Subsequently, these rule-based methods are not indicative of the reality regarding the channel importance that is captured in the data.

2.3.2 Algorithmic-based attribution

Algorithmic-based attribution is based on a statistical base to perform multi-credit assignment to channels. This implies that the parameters that determine the credit assignment are derived from the data. This section gives an overview of the most relevant developments in the attribution literature. The approaches can be generally classified into three categories: a probabilistic approach, the Shapley value, and logistic regression.

2.3.2.1 Simple Probabilistic The simple probabilistic model is a non-parametric approach that is used to solve the attribution problem proposed by Shao & Li (2011). In the context of attribution modeling, the model determines the attribution of a channel based on the successful customer journeys with one or two channels interactions. The computation is rather simple, and the intuition is that first, the model learns the distribution of the channels regarding the conversion in an aggregate manner. Then, the learned conversion distribution is applied on the individual visitor level to generate the attribution of an individual channel. So, first, the conditional probability of conversion given the channel needs to be calculated:

$$P(y = 1 \mid x_j = 1) = \frac{N_{(x_j=1 \cap y=1)}}{N_{(x_j)}}. \tag{1}$$

To take the interaction between channels into account, a second-order interaction term is taken into account. Let the interaction with the second-order interaction denoted as x_k for $k = \{1, 2, 3, ..., M\}$ channels, where $k \neq j$:

$$P(y = 1 \mid x_j = 1 \cap x_k = 1) = \frac{N_{(x_j=1 \cap x_k=1; y=1)}}{N_{(x_j \cap x_k)}}. \tag{2}$$

To calculate the contribution of channel j, the conditional probability of channel j and second-order interaction probability of channels j and k are summed at each converting user level:

$$C(x_j) = \underbrace{p(y=1 \mid x_j=1)}_{\text{conditional probability}} + \underbrace{\frac{1}{2N_{j\neq k}} \sum_{j\neq k} \{p(y=1|x_j \cap x_k = 1) - p(y=1|x_j=1) - p(y=1|x_k=1)\}}_{\text{second-order interaction effect minus conditional probabilities}}. \tag{3}$$

The first term is the base conditional probability for conversion of channel j, whereas the second term is the second-order interaction probability of channels j and k minus their individual conditional probabilities. It is assumed by the authors that the net interaction effect is equally divided between the channels; thus, the sum of the second term is divided by two

The advantage of using the simple probabilistic approach for attribution modeling is that it is data-driven as it does not use a priori distribution of weights but rather uses the data to find the relative contribution of each channel. Another advantage is that this approach is pretty straightforward and easy to interpret. However, a major disadvantage of the simple probabilistic model is that it is not possible for this method to predict the probability of conversion for a visitor based on the present interactions. Furthermore, it does not take all outcomes into account regarding the customer journeys as this method only looks at the converting customer journeys. Next, it can also be argued that fairness is also lacking since the method assumes that the net interaction effect is equally divided between two channels. The question then arises if this division of the net interaction effect by two is fair.

2.3.2.2 Shapley Value The Shapley value approach is a well-established cooperative concept from Game Theory that is used to fairly distribute a payoff generated by a coalition to each individual player. In attribution modeling, a marketing campaign can be seen as the game, whereas the channels are reflected by the players. The ideal outcome of a marketing campaign is, in this case, a conversion. Coalitions refer to the customer journeys as a prospect can have interaction with all channels or with a subset of the channels. The Shapley value method calculates the differences in generated payoff when one specific player is not present in the game versus when the player is present. The difference is called the marginal contribution with respect to a certain player.

In the context of attribution modeling, a campaign is defined by a set $W = \{1, 2, 3, ..., M\}$ channels and a characteristic function $v : 2^W \rightarrow \mathbb{R}$ that maps the value of subsets of channels to real numbers. All channels given in the campaign together are referred to as the grand coalition W with M channels, whereas other possible coalitions (subsets) excluding the grand coalition W are referred to as S. Characteristic function v assigns to each coalition S the value brought by a certain coalition under the assumption that all channels in that coalition cooperate with $v(\emptyset) = 0$, where $\emptyset$ denotes an empty coalition. The equation to calculate the marginal contribution using the Shapley Value method for channel j is given by:

$$\phi_j(v) = \sum_{S \subseteq W \setminus \{j\}} \underbrace{\frac{|S|!(|W| - |S| - 1)!}{|W|!}}_{\text{weighting factor}} \underbrace{(v(S \cup \{j\}) - v(S))}_{\text{marginal conttribution}} \tag{4}$$

In Equation 4, $|W|$ is the total number of channels in the grand coalition, and $|S|$ denotes the number of channels in subset $S \subseteq W$. In the equation, the sum is given of the marginal contribution of a channel j averaged over all possible combinations in which the coalition can be build up.

The advantage of using the Shapley value for attribution modeling is that it is a data-driven approach as it uses the distribution of the importance found in the data. Furthermore, the intuition behind the Shapley value calculation is rather straightforward and is, therefore, interpretable. Another major advantage of the Shapley value is fairness property, which means that channels that do not contribute to the payoff in a game get a 0 score attributed. Nevertheless, the Shapley value itself does not have the property to generate predictions on an individual customer journey level as the Shapley value only is a measure of player importance. Another major disadvantage is that the importance weights yielded are aggregated and only take the journeys ending in conversion into account.

2.3.2.3 Logistic Regression Logistic regression is a regression model which is specific for when the dependent variable is binary. In this case of attribution modeling, $y = \{y_1, y_2, y_3, ..., y_i \mid i = 1, 2, 3..., N\}$ is the outcome, also known as a conversion or non-conversion, for customer i. $x = \{x_{1,1}, x_{1,2}, x_{1,3}, ..., x_{i,j} \mid i = 1, 2, 3, ..., N; j = 1, 2, 3, ..., M\}$ is defined as the presence of interaction with channel j for customer i. Do note that other variables representing the journey properties can also be used, such as the number of clicks per channel. However, in this case, it is about the relative importance of a channel as opposed to all channels, which falls in the scope of the thesis. Thus, in the case of logistic regression, the attribution problem is treated as a binary classification problem. The log-odds of conversion of observation i given the interactions is given by:

$$\log\left(\frac{P(y_i = 1|x_{ij})}{P(y_i = 0|x_{ij})}\right) = \beta_0 + \sum_{j=1}^{M} \beta_j x_{ij}, \tag{5}$$

where the β-coefficients reflect the overall importance of channel j in all customer journeys.

The advantage of using logistic regression for attribution modeling is that it does not use a priori distribution of weights but rather uses the data to find the relative contribution of each channel. Additionally, logistic regression is easy to interpret as the methodology behind it is intuitive how the relative contribution is computed. Logistic regression is also able to predict whether the presence of channels will lead to a conversion, and the model is trained using both converting and non-converting customer journeys in the data set. Lastly, the fairness property is assumed to be valid

with regard to the properties of logistic regression.

2.3.3 Other models

Other attribution model approaches have also been proposed in the marketing literature. Anderl, Becker, von Wangenheim, & Schumann (2016) put forward a solution for the attribution problem in which the customer journey is modeled as a Markov chain, which is a stochastic model that describes a series of possible states in where the probability of a certain state only depends on the previous state. The attribution of each channel is calculated through what is known as the removal effect. The effectiveness of channel j is determined by looking at the change in probability of conversion when channel j is removed from the customer journey. Danaher & Heerde (2018) introduce an approach that is based on the probit model by estimating the uplift in the probability of the conversion at each state in the customer journey. It uses the same principle as the Shapley Value by looking at the marginal contribution of channel j and comparing the contribution with and without channel j. However, the probit model approach uses the marginal change in the conversion probability. These models are not discussed in this thesis as these approaches are putting emphasis on the ordering of the interactions and on the spillover effects to attribute conversion credit, which is beyond the scope of this thesis.

2.3.4 Theoretical evaluation

In the previous sections, the most common approaches for the attribution problem found in the marketing literature have been discussed. Table 1 summarizes the findings of the evaluation of the attribution models based on the criteria previously discussed. A model complies with a certain criterion indicated by a '+,' whereas no compliance with a criterion is indicated by '-' in Table 1.

Table 1: Evaluation of attribution techniques based criteria

Criteria	Rule-based	Simple probabilistic	Shapley value	Logistic regression
Ability to predict	-	-	-	+
Data-driven	-	+	+	+
All outcomes	-	-	-	+
Fair attribution	-	-	+	+
Interpretable	+	+	+	+

All models comply with the post-hoc interpretability as the models are all able to yield a clear overview of which channel contributes the most to a conversion. However, the rule-based methods are not data-driven as the importance of a channel is determined a priori. The three other models are, on the other hand, data-driven. Nevertheless, the simple probabilistic model does not comply with fair attribution as the process of credit attribution is based on certain assumptions. Another major drawback is that neither the simple probabilistic model and Shapley value approach takes the non-converting journeys into account, thus, violating the criterion that states that all customer journeys should be taken into account. The model that complies with all criteria is logistic regression. Based on this conclusion, logistic regression will be used in this thesis as the benchmark regarding the evaluation for the heavier machine learning approach.

2.4 Interpretable Machine Learning

As was previously mentioned in the prior section, a good attribution model should be interpretable while also maintaining great accuracy. Great interpretability is achieved, for example, by using generalized linear models (GLMs), which are additive models where the outcome depends on the sum of inputs and parameters, resulting in the model being able to provide a clear interpretation. The logistic regression approach is such a method that is derived from the GLM. Another comprehensible methodology is the simple classification and regression tree (CART), which is a graphical model where the outcome depends on the path from the root node to the terminal node. The graphical representation allows for a clear interpretation of the features in the case of CART (Freitas, 2014). The high degree of interpretation for GLM and CART methods is referred to as *intrinsic interpretability*, i.e., the interpretability of the learner. These models are inherently interpretable due to restricting the complexity in these methodologies - also known as white-box models (Molnar, 2018). The *post-hoc interpretability* refers to the understanding of why the model has made a certain prediction without necessarily understanding the underlying mechanisms of the model - referred to as black-box models (Lipton, 2016; Molnar, 2018). In the criterion found in the literature study, *interpretability* reflects the post-hoc interpretability of a model. Therefore, in this thesis, interpretability will be used to refer to post-hoc interpretability.

For the interpretability of machine learning models, two methods are employed: *model-specific* and *model-agnostic* (Adadi & Berrada, 2018). Model-specific interpretation is specific for a certain class of machine learning algorithms. These methods leverage on the inherent structure of the machine learning model to explain the prediction. As previously used as an example, a class-specific approach could be to explain the predictions made by a decision tree that has an inherently graphical structure. Model-agnostic interpretation, on the other hand, is not specific for a certain class of machine learning models. In this case, the explanation technique considers the model as an unknown function and tries to reverse engineer the behavior based on input and output (Guidotti et al., 2018).

The interpretability can also be further categorized into two scopes: *local* and *global* interpretability (Adadi & Berrada, 2018; Lipton, 2016; Molnar, 2018). Local interpretability refers to explaining predictions on observation level, whereas global interpretability considers explaining the outcomes of the machine learning model as a whole. An example of local interpretability in the case of attribution modeling is to see how the presence or non-presence of certain channels contribute to the observation of its predicted outcome by the model. Conversely, global interpretability would try to explain which channels are important on a general level, so for all observations. In the literature, two model agnostic interpretation methods are commonly used: *LIME* and *SHAP*. LIME is the acronym for *Local Interpretable Model-agnostic Explanations*, which uses a local linear model (surrogate model) to explain the effect of each feature for an observation's prediction put forward by Ribeiro, Singh, & Guestrin (2016). SHAP stands for *Shapley Additive exPlanations* that uses the Shapley value to calculate the feature importance for each instance in the data set by looking at the prediction scores with and without the feature to be investigated proposed by Lundberg & Lee (2017). A major advantage of SHAP over LIME, is that SHAP can approximate the local explanation while still providing global explanations. The ability to explain an outcome on an observational level is a criterion found in the literature study.

3 Data

This chapter will dive into the data used in the empirical part of this research. Section 3.1 gives an impression of the background of the company from where the data is obtained. Section 3.2 describes the data set by defining the customer journey and the variables. Section 3.3 focuses on the data pre-processing. Section 3.4, focuses on the key statistics and important findings before proceeding with the empirical analysis. The last section, Section 3.5, briefly discusses how the data is processed in order for the empirical analysis.

3.1 Company background

In this research, the data is obtained from a Dutch financial provider that offers a range of financial products and services under its various brands. The company its offerings span from health to life and non-life insurance products and services in both the B2C and B2B markets in the Netherlands. The insurer advertises its offerings both above-the-line (ATL), e.g., TV and radio, and also below-the-line (BTL), e.g., email and phone. However, for research purposes, ATL and BTL cross-channel effects are not taken into account. Instead, the focus will be on below-the-line advertising as these effects on the conversion can be measured on an individual level as opposed to estimated from the mass. The provided data set include the customer journeys focused on the car insurance product of a specific brand, thus, mapped product-based and brand-based.

3.2 Data description

The customer journey data contains information regarding the source of the click (touchpoints), timestamp, and whether a conversion has taken place. Furthermore, the data is recorded from 19 January 2019 to 28 March 2021. Subsequently, The touchpoints themselves are mapped into at least two levels, which can be a combination of channel, product, user device, search engine, or product. Consequently, 462 touchpoints are observed in the data set. However, it should be noted that this level of granularity is not needed for this research as it is only of interest to determine the attribution of each channel. Hence, the level of granularity will be solely based on channel and the most important differences in intention. In Table 2, a summary is given of the data set.

Table 2: Variables in the customer journey dataset

Variable	Description
journey_id	A unique identifier on the individual journey
touchpoint	An identifier on which touchpoint was interacted with
timestamp	The exact time and date the visitor interacted
journey_step	An numeric indicator of the step in the journey
conversion	An indicator on the (non-)conversion

Table 3: Snippet of customer journey data set

journey_id	touchpoint	timestamp	journey_step	conversion
20813613	search	29-10-2019 19:24:23	1	0
20813613	phone	30-10-2019 08:11:00	2	1
...	...	...	...	...
74690286	direct	12-11-2020 09:13:35	1	0

In Table 3, it can be seen that every journey is recognized by its unique id (`journey_id`), which is assigned to a visitor whenever the website is visited for the first time - or whenever one of the three business rules regarding the customer journey is violated. In this data set, a customer journey is defined as a sequence of interactions, indicated by the `touchpoint` variable, that either leads to a conversion (buy) or non-conversion (no buy). For example, it can be observed that a customer (20813613) had the first interaction with the paid search touchpoint and then, on the following day, had a phone conversation (second interaction) with the company to purchase a car insurance. Notice that the first and step are captured by the `journey_step` variable and are also be able to be deducted from the `timestamp` variable. Another customer (74690286), for instance, interacted with the insurer by going directly to the website on 12 November 2020, so the first interaction and has not led to a conversion indicated by the '0' shown in the `conversion` column. Note that the conversion is not stored as an individual touchpoint in the dataset but rather in a separate column. It is also important to pinpoint that the insurer uses a set of rules to define a customer journey:

1. Maximum duration of a customer journey is 70 days
2. Maximum time difference between steps is 14 days
3. Maximum number of steps in a customer journey is 20

If one of these numbers is exceeded, then a new customer journey is created via the assignment of a new journey id. Lastly, it should also be outlined that whenever a visitor converts, the customer journey ends.

3.3 Data pre-processing

This section will look into the data pre-processing process to make the data ready for the empirical analysis. Subsection 3.3.1 discusses the process of making the touchpoints less specific as the current level granularity exceeds the purpose of this research. Subsection 3.3.2 focuses on creating the actual customer journeys to be used in modeling the actual attribution model.

3.3.1 Lowering the granularity of the touchpoints

As was already previously stated, the number of touchpoints amounts to 462, due to the high level of granularity on which the touchpoints are mapped by. See Appendix A for a summary of the touchpoints. Please note that in Appendix A, the number of touchpoints is 15 but in the processed data there are 47, the representation in Appendix A is purely for comprehension.

To understand which touchpoints can be merged safely to lower the detail of information, an analysis was performed on the touchpoints present in the dataset in consult with the financial provider. The mapped touchpoints that have a big enough relative difference in the ratio of conversion to non-conversion are then compared to each other within the same base mapping. For instance, the aggregated conversion rate of touchpoint `Direct` is relatively low. However, within this aggregation there are different groups for which the conversion rate differs, e.g., a direct visit to the brand's landing page for the product car insurance (`Direct_car`) has a higher conversion rate for this product as opposed to a direct visit to a non-related product (`Direct_noncar`) with regards to the conversion rate of the product car insurance. Therefore, it is important to make distinctions between certain touchpoints. In the footnote of Appendix A, the touchpoints that have been submapped are shown.

3.3.2 Creating the customer journeys

To create the actual customer journeys, the data set provided by the insurer has to be transformed. In order to perform the transformation, the following steps were taken:

- Inspect whether customer journeys have been cut off in the data set. As was previously said, the dataset contains information on the customer journey ranging from 19 January 2019 to 28 March 2021. Using this range of dates results in some of the customer journeys being cut off. For example, a conversion that has taken place on 19 January 2019 by a visitor misses information on the previous touchpoints the visitor interacted with. Similarly, the problem also occurs for visitors that interact with the touchpoints on 28 March 2021, the customer journey is being cut off resulting in no information on the further interaction and even whether a conversion has taken place. To overcome this problem, all customer journeys

are taken into account that have occurred between 19 February 2019 and 28 February 2021. If the customer journey has additional touchpoints outside this range, then the touchpoints are retrieved from the additional months, 19 January 2019 till 19 February 2019 and 28 February 2021 till 28 March 2021.

- The data set has to be coded in the correct format, which is done via one-hot encoding the touchpoints. This method will result in one customer journey per row (row-wise) in which the touchpoints are reflected by binary variables. The presence of a touchpoint in a certain customer journey is, thus, indicated by a `1`, whereas the value of `0` means that that the visitor has not interacted with the touchpoint. In Table 4, it can be seen that the visitor of customer journey `24729421` interacted with the touchpoints `direct` and `webcare` indicated by the value of `1` in the corresponding columns, which has ultimately led to a conversion indicated by the value of `1`. It is important to point out that one-hot encoding omits the order of the touchpoint and due to the nature of the Shapley Value attribution this is justified.

Table 4: Snippet of one-hot enconded customer journey data set

journey_id	affiliate	app	dealer	direct	...	webcare	other	conversion
24729421	0	0	0	1	...	1	0	1
24835022	1	0	0	0	...	0	0	0
25937402	0	1	0	1	...	1	0	1

3.4 Data insights

This section will briefly discuss the insights found in the data set provided by the insurer. Subsection 3.4.1 focuses on key statistics to describe the data set. Subsection 3.4.2 looks deeper into the distribution of the touchpoints occurring in all the customer journeys. Subsection 3.4.3 discusses the distribution of the conversion rate aggregated for all customer journeys and seen per touchpoint.

3.4.1 Key statistics of the data set

As was previously stated, the number of touchpoints used in this data set amounts to 36, which is a result of a lower level of granularity. The number of interactions captured in the data is 124,827,038, which comes from the 61,189,553 customer journeys. Furterhmore, the average journey length is 2.04 steps. Please note that this includes 1-step journeys, excluding 1-step journeys, the average journey length is 4.34 steps that last on average 272,13 hours (11 days, 8 hours and 8 minutes). Lastly, the average is conversion rate is 0.70 percent, excluding 1-step journeys the average conversion rate is 1.81 percent. See Table 5 for a summary on the key statistics.

Table 5: Key numbers in the data set

Key number indicator	Key number
Number of different touchpoints	47
Number of interactions	124,827,038
Number of journeys	61,189,553
Interacted with ≥ 3 touchpoints	10,936,861
Interacted with ≥ 5 touchpoints	5,164,183
Average journey length[1]	2.04
Average duration between steps[2]	272,13 hours
Average conversion rate[1]	0.70%

[1] Includes 1-step journeys

[2] Excludes 1-step journeys

3.4.2 Distribution of the touchpoints

Appendix B gives insight in the number of interactions with each touchpoint. It can be observed that the majority of the traffic interacts with the touchpoint `email` totaling to almost 50 percent of all interactions. 18.03 percent of the traffic interacts with the insurer via the `direct` touchpoint, meaning that the website is directly visited via the browser. 11.06 percent of the traffic interacts with the `search` touchpoint, from which 63.3 percent of the traffic comes from paid search (SEA), the remaining 36.7 percent comes from organic search (SEO). 9.62 percent of the traffic interacts with Independer, implying that almost one out of ten interactions are using the comparison website. These four touchpoints dominate the traffic regarding the interactions.

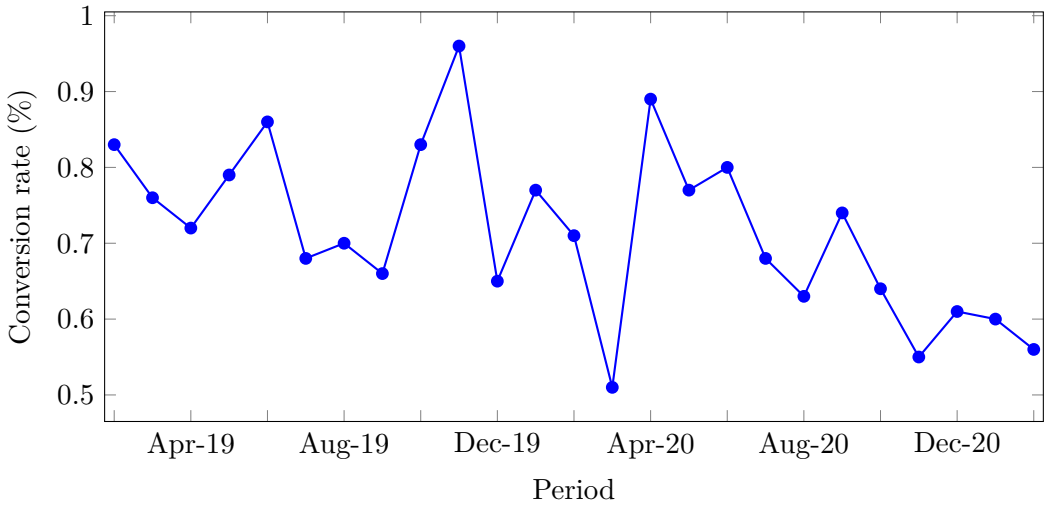
The amount touchpoints present in a customer journey is also seen to be positively correlated with the conversion rate. A one unit increase in touchpoint within a journey, is correlated with an average of 0.50 percent higher conversion rate not taking the type of touchpoint into account. As was already shown in Table 5, the majority of the customer journeys consists out of one-step customer journeys.

3.4.3 Distribution of the conversion rate

It is important to note that the data set also recorded data on the period during the COVID-19 pandemic. In 2020, the pandemic had a negative affect on the sales of car insurance in the month March as that was the beginning of the intelligent lockdown. The rest of the year, was only slightly affected as there was even an 2.8% increase in the sales of occasions in 2020 compared to 2019

totaling to 2,025,309 sold occasions (BOVAG, 2021a). Nevertheless, there were 20% less new cars being registered in 2020 as opposed to 2019, from 444,217 in 2019 to 356,051 in 2020. (BOVAG, 2021b). However, the volume of sold occasions far exceeds the volume of sold new cars in 2020. In the following year, 2021, the negative effects of the pandemic were more prominent with regards to the sales of car insurance. The beginning of 2021 was led by a second intelligent lockdown, which has resulted in less sales of both used and new cars (BOVAG, 2021c). As most of the recorded periods of data has only been slightly affected by the COVID-19 pandemic, it should not pose any problem for the empirical research. Figure 3 gives an overview of the average conversion rate of the car insurance product per month.

Figure 3: Conversion rate (%) per month for period February 2019 till February 2021



3.5 Data processing

All data is pre-processed in SAS Enterprise using SAS and SQL. Afterwards, the data is loaded in Python to be used in the empirical analysis. As the data set itself is too large to handle for Python due to memory limitations, 50 percent of data set is randomly sampled as base data. As the data itself imbalanced, the majority class had to be undersampled using weights for the training set. The test set had the original distribution to keep the data set representative to simulate the real-life scenario in which the data set is imbalanced.

4 Method

This chapter explains the method to build a multi-touch attribution model that satisfies the five criteria put forward in the previous chapter. Section 4.1 and Section 4.2 focuses on the machine learning methods used to create a classification model. Section 4.3 describes how the models will be evaluated using the classification error yielded by the model and how variability of the model is measured. Finally, section 4.4 focuses on the interpretable machine learning method for obtaining feature importance from the model.

4.1 Classification with Logistic Regression

This research focuses on the application of plain logistic regression algorithm to explore the possibility of building an adequate attribution model. In Section 3.1.1 odds and log-odds are explained to understand how the transformation is applied to logistic regression. Next in Section 3.1.2 the logistic regression approach itself is discussed.

4.1.1 Odds and log-odds

The odds of conversion ($y = 1$) is the ratio of $P(y = 1)$ to $P(y = 0)$, where $P \in [0, 1]$. In other words, it is the ratio of the outcome *conversion* to the outcome *non-conversion*. Then the odds of the event occurring are:

$$\text{odds} = \frac{P(y = 1)}{P(y = 0)}, \tag{6}$$

where the odds $\in [0, \infty]$. To make interpretation easier, taking the log of the odds allows for uniform outcomes that can be expressed in both positive or negative values:

$$\text{log-odds} = \frac{\exp(\log(\text{odds}))}{1 + \exp(\log(\text{odds}))}, \tag{7}$$

where the log-odds $\in [-\infty, +\infty]$.

4.1.2 Logistic regression

Logistic regression was previously mentioned in Chapter 2. It was stated logistic regression is a regression model that is specific for when the dependent variable is binary. Let $\mathcal{D} = \{(x_{1,1}, x_{1,2}, ..., x_{1,j}, y_1), ..., (x_{i,j}, y_i) \mid i = 1, 2, ..., N; j = 1, 2, ...M\}$ be the data set. For the thesis, the sci-kit learn logistic regression classification implementation was used in Python. Logistic regression is a GLM with a transformation that the output (Y) of the model is limited between 0 and 1. Thus, the probability of conversion for visitor i given the explanatory variables X_{ij} using Equation 8:

$$P(y_i = 1 \mid x_{ij}) = \frac{1}{1 + \exp(\beta_0 + \sum_{j=1}^{M} \beta_j x_{ij})}. \tag{8}$$

The odds are given by:

$$\frac{P(y_i = 1|x_{ij})}{1 - P(y_i = 1|x_{ij})} = \frac{P(y_i = 1|x_{ij})}{P(y_i = 0|x_{ij})} = \exp(\beta_0 + \sum_{j=1}^{M} \beta_j x_{ij}). \tag{9}$$

The log-odds are given by:

$$\log\left(\frac{P(y_i = 1|x_{ij})}{P(y_i = 0|x_{ij})}\right) = \beta_0 + \sum_{j=1}^{M} \beta_j x_{ij}. \tag{10}$$

For ease of notation, let $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i) \mid i = 1, 2, ..., N\}$ be the data set where each x_i is an input vector of features M and y_i is the corresponding label. The goal of logistic regression is to estimate the $M + 1$ unknown coefficients β through maximum likelihood estimation (MLE) that finds the best set of coefficients for which the likelihood is the greatest for the observed data while minimizing the difference between the predicted class $\hat{y}_i$ and the actual class y_i using the log-likelihood loss function:

$$\sum_{i=1}^{N} \ell(\hat{y}_i, y_i) = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(p(y_i)) + (1 - y_i) * \log(1 - p(y_i)) \right]. \tag{11}$$

4.2 Classification with XGBoost

This research focuses on the application of the XGBoost algorithm to explore the possibility of building an adequate attribution model. In Section 4.2.1 the general idea of Boosting is explained. Next in Section 4.2.2 the Gradient Descent is introduced which is the fundamental part of Gradient Boosting. Section 4.2.3 elaborates on the Gradient Boosting methodology itself. Then, Section 4.2.4 introduces XGBoost which is a newer implementation of regular Gradient Boosting that is used for the actual model creation. Finally, Section 4.2.5 focuses on tuning the hyper parameters to tune the model and Section 4.2.6 briefly shows the application of k-fold cross-validation.

4.2.1 Boosting

Boosting is an ensemble method that uses meta-learning to increase the prediction accuracy. The Boosting method was introduced by Kearns & Valiant (1989) when the authors posed the question of whether a set of weak learners could be transformed into a strong learner. A weak learner refers to any machine learning algorithm that provides an accuracy that is only slightly better than random guessing. In the case of binary classification this means that the error rate (ε) of a weak learner is always less 0.5. A strong learner, on the other, has a greater accuracy and has arbitrarily a smaller error rate as opposed to a weak learner. Schapire (1990) affirms this thought and showed that weak learners can be combined to generate a strong learner. In Boosting a model is sequentially trained and improves its predictive ability from adapting to the misclassification of the model in each iteration.

4.2.2 Gradient Descent

The Gradient Descent algorithm was introduced by Cauchy (1847) and is an iterative optimization algorithm that looks into minimizing a loss function. Consider again, $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i) \mid i = 1, 2, ..., N\}$ as the data set. Let $F(x)$ be the model that maps the explanatory variables x to outcome y. Consider $F_w(x)$, with model parameters w, that maps the explanatory variables x to outcome y. The goal is to minimize the loss function $L(z, w)$ The minimization of the loss function that models the performance of $F_w(x)$. Gradient descent looks into the derivative of the loss function $\frac{\delta L(z,w)}{\delta L(w)}$ or gradient of the loss function $\nabla L(z, w)$ to minimize the loss with respect to the parameters w.

4.2.3 Gradient Boosting

Gradient boosting was proposed by Friedman (2001) and uses the idea of boosting while using gradient descent for the minimization of the loss function. Consider again, the data set $\mathcal{D} =$

$\{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i) \mid i = 1, 2, ..., N\}$. For each observation i the differentiable loss function can be denoted by $L(y_i, F(x_i))$, which describes the loss between the actual outcome (y_i) and the predicted outcome by the model $(F(x_i))$ for observation i. Boosting uses multiple weak learners referred to as $h(x)$ to create a strong learner $F(x)$. As the model learns sequentially, the best initial guess for $F_0(x)$ would be $h_0(x)$ that minimizes the loss function:

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, \gamma). \tag{12}$$

After the calculation of $F_0(x)$, the goal of gradient boosting is to find a weak learner that gives the largest reduction in the loss function. The optimization of the loss function L is done by searching for the steepest descent in function space. For each model t that is iteratively trained on the errors of model $t-1$, the negative gradient, also called pseudo-residuals, of the loss function of the most recent model $F_{t-1}(x)$ for every observation pair (x_i, y_i) is calculated:

$$r_{it} = -\left[\frac{\delta L(y_i, F(x_i))}{\delta F(x_i)}\right]_{F(x)=F_{t-1}(x)}, \tag{13}$$

where the negative gradient r_{it} is the steepest descent in function space for observation i. Then, the weak learners are fitted h_t on r_{it} with the weak learner chosen that best approximates the pseudo-residuals. As was previously said, gradient boosting learns sequentially meaning that the weak learners are added to the one another to create a strong learner. Thus, after finding the best weak learner h_t fitted on the pseudo-residuals r_{it}, the best weak learner needs to be added to the most recent model $F_{t-1}(x)$ using a proper scaling parameter γ. The proper scaling parameters p_t to be multiplied with the best weak learner h_t is found via one-dimensional optimization over all observations n and gives:

$$\gamma_m = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, F_{t-1}(x_i) + \gamma h_t(x_i)). \tag{14}$$

The properly scaled weak learner is then added to the most recent model to create an updated model:

$$F_t(x) = F_{t-1}(x) + \gamma_t h_t(x). \tag{15}$$

The model stops training until the pre-specified number of iterations T are reached. Algorithm 1 summarizes the steps discussed previously in this subsection into algorithmic form.

Algorithm 1: Gradient Boosting

Input : $\mathcal{D} = \{(x_i, y_i) \mid i = 1, ..., n\}$, loss function $L(y, F(x))$, number of iterations T.

Output: $F_T(x)$.

1 $F_0(x) = argmin_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.
2 **for** $t = 1$ *to* M **do**
3 $\quad r_{it} = -\left[\frac{\delta L(y_i, F(x_i))}{\delta F(x_i)}\right]_{F(x)=F_{t-1}(x)}$.
4 $\quad$ Fit weak leaner $h_t(x)$ to r_{im} using the training set $\{(x_i, y_i) \mid i = 1, ..., n\}$.
5 $\quad \gamma_t = argmin_\gamma \sum_{i=1}^{N} L(y_i, F_{t-1}(x_i) + \gamma h_t(x_i))$.
6 $\quad F_t(x) = F_{t-1}(x) + \gamma_t h_t(x)$.
7 **end**

4.2.4 eXtreme Gradient Boosting (XGBoost)

The XGBoost method was introduced by Chen & Guestrin (2016) and follows the principle of gradient boosting with the major advantage that it is a more regularized implementation that controls for overfitting. For the thesis, the sci-kit learn XGBoost classification implementation was used in Python. The algorithm makes use of second-order gradients of the loss function to approximate the the true cost function resulting in real-life application where the use of XGBoost can be up to 10 times faster than regular gradient boosting algorithms (Chen & Guestrin, 2016).

The XGBoost algorithm uses a gradient boosted decision tree approach to build the model. Thus, in this case, the weak learner $f_t(x)$ is a decision tree with J terminal nodes. The loss function used for XGBoost classification is an additive function of logarithmic loss that calculates the error between predicted outcome ($\widehat{y_i}$) and the observed outcome (y_i) plus a penalty term (Ω) that penalizes the complexity of the weak learners (f_t) against overfitting. This results in a regularized loss function:

$$L = \sum_{i=1}^{N} \ell(\widehat{y}_i, y_i) + \sum_{t=0}^{T} \Omega(f_t), \tag{16}$$

$$\sum_{i=1}^{N} \ell(\widehat{y}_i, y_i) = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(p(y_i)) + (1 - y_i) * \log(1 - p(y_i)) \right], \tag{17}$$

$$\sum_{t=0}^{T} \Omega(f_m) = \gamma J_t + \frac{1}{2}\lambda ||w_t||^2, \tag{18}$$

where t represents the iteration of the T XGBoost models. γ and λ are the regularisation parameters, where penalizes for the number of the terminal node (J_t) and penalizes the weights of the terminal node (J_t) for tree (f_t) via ℓ_2 regularization. The goal is to minimize the regularized loss function with regards to its parameters. As can be derived from the regularized loss function in Equation 9, the trade-off between model accuracy and model complexity has to be found by optimization.

4.2.5 Hyperparameter tuning

The XGBoost scikit-learn implementation has a variety of hyperparameters that need to be tweaked in order to maximize classification performance while still maintaining generalization. The hyperparameters that are tuned for the XGBoost model are the following:

- **Learning rate**. The XGBoost algorithm learns sequentially, thus, the learning rate scales the weights for the features per iteration. A lower learning rate makes the boosting process more conservative and slower. The default learning rate is 0.3 in the XGBoost algorithm.
- **Maximum tree depth**. A higher maximum depth of a tree means that more splits are created for each individual decision tree per iteration until the maximum depth is reached. Increasing the maximum tree depth can lead to overfitting and higher complexity of the model. The default maximum tree depth is 6 in the XGBoost implementation.
- **Minimum child weight**. The minimum child weight refers to mininum number of instances that need to be in each terminal node to make the node be part of the model. Lowering the minimum child weight makes the model less conservative. The default minimum child weight is 1 for XGBoost.
- **Minimum split loss**. The minimum split loss specifies what minimum reduction in the loss should be to make the split in a terminal node be part of the model. Increasing the minimum split loss makes the model more conservative. The default value of the minimum split loss is 0.
- **Scale positive weights**. The scaling of positive weights is used for the presence of unbalanced classes. A higher scaling towards positive weights, favors the weights found in the positive classes. The default positive weights scale value is 1. In order determine a indicative value is to divide the positive class by the negative class $\frac{\sum_i^N y_i=1}{\sum_i^N y_i=0}$.

In order to determine what settings to use for the hyperparameters, a random grid search is

performed. This technique to tune the parameters makes random combinations of the hyperparameters to increase the generalizability for the XGBoost model. For the learning rate, the values $\{0.01, 0.05, 0.1, 0.15, 0.20\}$ were used. The best maximum tree depth was found by using the range $\{5, 10, 15, 20, 25\}$. The find the best performing minimum child weight the settings used were the values $\{1, 3, 5, 7, 9\}$. The minimum split loss was determined by using the range $\{1, 2, 3, ..., 10\}$. Lastly, the best scale positive weights was found by using the following values $\{1, 25, 50, 100, 150, 200\}$. The random combinations of hyperparameters are examined using k-fold cross-validation. The final hyperparameters used in the XGBoost model are described in the results section.

4.2.6 K-fold cross-validation with random search

K-fold cross-validation is used to determine the best distribution for the hyperparameters C of a machine learning to maintain its generalizing abilities, which is for the model to predict accurately on not seen data. In k-fold cross-validation, the training data set is split into k folds. One the $k-1$ folds, the model is trained. The minus one represents the fold that is used for evaluation, which functions as an out-of-sample test. The process repeats itself K times where the hyperparameters its values are randomly determined for each $c \subseteq C$. Then, the test error ε is averaged for all K evaluations, resulting in an evaluation measure for the out-of-sample test for a model. Additionally, this thesis also performs a split of the data in a test and training set before the actual K-fold cross-validation (outer split). The final model itself will be tested on the outer test data set.

Algorithm 2: Nested K-fold cross-validation with Random Search

Input : data set $\mathcal{D} = \{(x_i, y_i) \mid i = 1, ..., n\}$, set of hyperparamaters C, number of outer folds K_1, number of inner fold K_2

for $i = 1$ *to* K_1 **do**
  Split $\mathcal{D}$ into $\mathcal{D}_i^{train}$, $\mathcal{D}_i^{test}$, for the i'th split
  for $j = 1$ *to* K_2 **do**
    Split $\mathcal{D}_i^{train}$ into $\mathcal{D}_j^{train}$, $\mathcal{D}_j^{test}$, for the j'th split
    for *each* $RandomSample(c) \subseteq C$ **do**
      Train model $F(x)$ on $\mathcal{D}_j^{train}$ with set p
      Compute test error ε_j^{test} for model with $\mathcal{D}_j^{train}$
    end
    Select best set $c^* \subseteq C$ where ε_j^{test} is optimal
    Train model $F(x)$ with $\mathcal{D}_i^{train}$ using c^*
    Compute ε_i^{test} for model $F(x)$ with $\mathcal{D}_i^{test}$
  end
end

4.3 Evaluation of the model

In order to evaluate the predictive performance of the models proposed in the previous sections, an adequate evaluation metric has to be used. Section 3.3.1 addresses the problem of imbalance in the data set. In section 3.3.2 touches upon precision and recall which are two classification evaluation metrics. Section 3.3.3 discusses the use of the precision-recall curve (PRC) which is used in the case of severe data set imbalance.

4.3.1 Imbalanced data set

The product of car insurance has the nature that the amount of actual converting customer journeys as opposed to non-converting customer journeys is significantly less. As was seen in the previous chapter, the actual conversion rate is 0.70 percent. It is evident that a product of this nature is one which the consumer does not buy every day, e.g., toilet paper. Therefore, a more lengthy and rational decision-making process is involved resulting in a severely imbalanced data set

In this case the non-converting customer journeys are overrepresented and the converting customer journeys underrepresented. Consequently, the severely imbalanced data set creates an issue for classification metrics. Take for instance, the classification metric *accuracy* that calculates the fraction of observations correctly classified over all observations. Using this evaluation metric would result in a extremely high accuracy as the model would tends to be more biased towards the non-conversion event in the case of an imbalanced data set. However, observations that indeed converted but were incorrectly predicted are ignored using this classification metric. In the case of attribution modeling, it is also of great interest to correctly predict a converting customer based on the interactions with certain channels.

4.3.2 Precision, Recall and F1 score

Figure 4 shows the confusion matrix, also known as the error matrix, for a binary classification problem. The confusion matrix gives a global overview of the observed classifications against the predicted classifications made by the model. The number of incorrect and correct predictions are summarized by each class for the observed values. The name *confusion matrix* stems from the way this approach of a contingency table allows to see how the model confuses the two classes of conversion and non-conversion. A true positive (TP) occurs when the model correctly predicts the positive class, whereas, a false positive (FP) occurs when the model incorrectly predicts the positive class. Similarly, a true negative (TN) occurs when the model correctly predicts the negative class, whereas, a false negative (FN) occurs when the model incorrectly predicts the negative class.

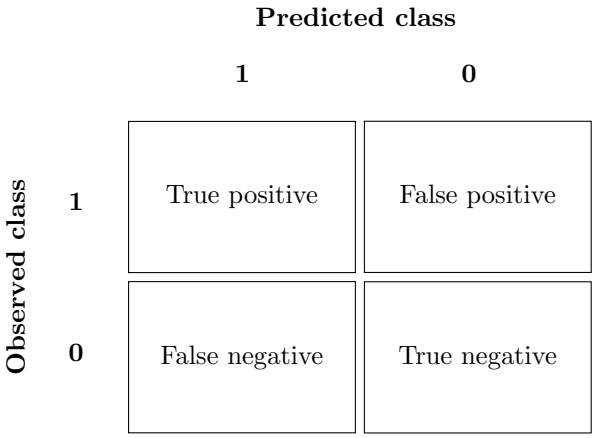


Figure 4: Confusion matrix for binary classification problem

Precision scores the fraction of correctly positive classified instances over all the predicted positive classified instances yielded by the model:

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}. \tag{19}$$

Recall scores the fraction of correctly positive classified instances over the observed positive classified instances:

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negatives}}. \tag{20}$$

The precision score evaluates how many of the positive classified observations were correctly identified of all predicted positive classified observations by the model. If the score is rather low, one could say that the model indeed classifies new observations as positive despite the imbalanced data set. However, the actual correctly predicted positive observations is rather low and, thus, not great at identifying positive observations. The recall, on the other hand, scores how many of the positive classified observations were correctly identified of all observations that were observed in the positive class. Note that both metrics are emphasizing the classification performance with regards to the positive class. In the case of attribution modeling, a conversion is a rare event.

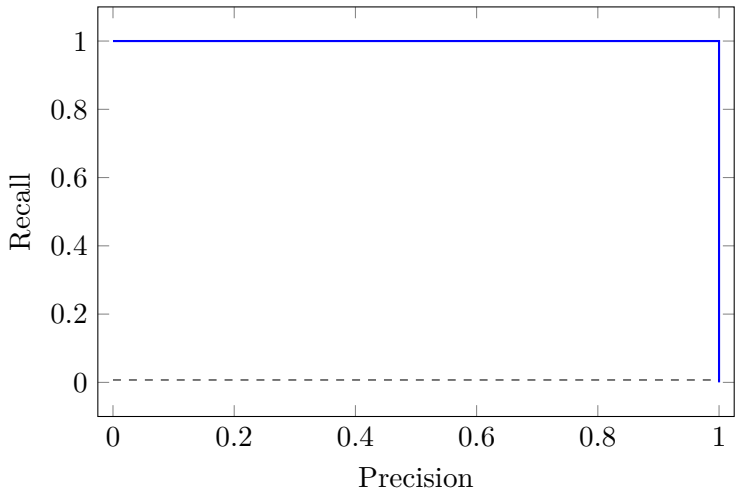
When aiming to maximize the classification performance of the model based on these two metrics, a trade-off is to be dealt with. While recall evaluates the model's ability to identify all the instances of interest (positive class) in the data, the precision scores the proportion of the instances of interest

the model was able to identify, i.e., maximizing the classification performance based on one of the two metrics leads to lowering the classification performance regarding the one of the two metrics. The F_1-score evaluates the model's predictive ability while taking both the metrics into account:

$$F_1\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{21}$$

The precision-recall curve plots the precision of against the recall at varying thresholds for a classification model. The threshold impacts the classification of the instances resulting in different recall and precision scores. A classification model that has no ability to discriminate the two classes would yield random classes or constant classes for the instances in the data set. Figure 5 illustrates the precision-recall curve. The dotted line, indicates the baseline which is the ratio between the positive and the negative class in the data set. The baseline curve would be present if the model had a random predictive ability. A perfect classifier, indicated by the blue line, would show a combination of two straight lines. A precision-recall of a model closer to the perfect classifier curve have a better predictive performance than the models closer to the baseline.

Figure 5: Precision-recall curve



To illustrate which of the machine learning approaches performs better, the precision-recall curve is used as it shows for every possible threshold what the performance of the model would be. The determination of the optimal threshold to translate the performance to a concrete confusion matrix, the F_1-score is utilized as this metric scores the model both on the recall and precision and, thus, gives the best cut-off value while optimizing recall and precision. Furthermore, the area under the curve of the precision-recall score, the average precision score, is used to compare the models for all possible thresholds.

4.4 Feature importance extraction

To extract the relative importance of a certain channel, feature importance extraction methods have to be used. Subsection 4.4.1 dives briefly into the feature importance extraction using last touch attribution. Subsection 4.4.2 discusses how to relative feature importance is calculated for logistic regression using the weights from the model. In Subsection 4.4.3 SHAP is introduced to extract the relative feature importance in both the XGBoost and logistic regression model. Ultimately, the three methods have to be compared to each other.

4.4.1 Last touch attribution

In the literature review, the last touch attribution method was already introduced. It assigns all of the weight to the last interaction making it the most important. It does this by using taking the ordering of the touchpoints into account. Despite the fact that the ordering of the interactions falls beyond the scope of this thesis, it is still of value to take the last touch attribution approach into account as it is the most common method to get the feature importance in the field of attribution modeling.

To denote the method mathematically, a new notation needs to be introduced takes the order into account. Let $i = \{1, 2, 3, ..., N\}$ denote the visitor that have who have an online customer journey with $s = \{1, 2, 3..., S_i\}$ steps in duration. Then, the channel importance for visitor i is notated as:

$$\text{importance}_i = \begin{cases} 0 \cdot x_{ij} & \text{where } s = \{1, 2, 3, ..., (S_i - 1)\} \\ 1 \cdot x_{ij} & \text{where } s = S_i \end{cases} \tag{22}$$

Consequently, for every visitor i, channel j is added to a count for a every channel (count_j). In SAS the use of a `count`-function is used to determine the relative importance of a channel. The counts are then transformed into fractions, which is seen as the probability distribution of the channels, and eventually converted to log-odds for comparison use. It should be noted that the total data set instead of the weighted data set, discussed in the previous chapter, was used for this calculation as SAS is better able to handle large data sets.

4.4.2 Feature regression weights

In logistic regression, the coefficients (βs) of the model can be used as a means of feature importance. It was previously defined that for visitor i the presence of channel j in the customer journey is denoted as x_{ij} for which the value can be either 0 or 1. A feature weight β_j, as seen in Equation

11, can be interpreted as the change in the log-odds of conversion when channel x_j is present. A $\beta_j > 0$ means a positive change in the log-odds of conversion when channel x_j is present, whereas, a $\beta < 0$ implies a negative change in the log-odds of conversion when channel x_j is present. The weights are derived from the scikit learn logistic regression model in Python.

4.4.3 SHapley Additive exPlanations (SHAP)

SHAP introduced by Lundberg & Lee (2017) is, as was previously introduced, a model-agnostic method of explaining a model, i.e., SHAP can be used on every machine learning method where an output is yielded. In this case, SHAP is applied on both the logistic regression and XGBoost model to investigate further if the feature importance is different for both approaches.

To explain the model locally, SHAP uses an additive feature attribution approach in combination with the Shapley Value. In this methodology, the explanation model is linear function of binary variables as seen in Equation 24. The original prediction function f of input features is being explained by g the explanation model in which g is the linear function with binary variables. The prediction function f is a function of x, where $x \in \mathbb{R}^M$ with M input features. Let $z' \in \{0,1\}^M$ represent the presence of the features. If the feature j is present, then $z_j^{'} = 1$, otherwise $z_j^{'} = 0$. To come to this linear model, a mapping function $h_x(x')$ is needed that transforms the original input to a simpler input denoted by x', such that $x = h_x(x')$. Thus, h_x converts the simplified vector z' back to the original input vector x. The linear additive feature function of binary variables is then defined as:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z_j^{'}, \tag{23}$$

where ϕ_j is the effect of each feature on the approximated prediction output of $f(x)$. The sum of the effect of feature j results in the approximation of $f(x)$.

The Shapley value was introduced in Chapter 2 as a means of to distribute the payoff among all players, assuming that all players collaborate. It calculates the marginal contribution of each player by considering all the possible orderings in which the player can be arranged. SHAP uses this calculation to calculate the marginal contribution of each feature j to attribute a weight ϕ_j to each feature. It uses the probability yielded by function $f(x)$ as the payoff to be distributed over the whole collection of W with M features. The possible sets of M input features is denoted as S. The order in which features are added affects the yielded probability, however, for the individual values are averages across all possible orderings calculating an average effect. Using these definitions, the formula to calculate ϕ_j can be written as follows:

$$\phi_j = \sum_{S \subseteq W \setminus \{j\}} \frac{|S|!(|W| - |S| - 1)!}{|W|!}(f_x(S \cup \{j\}) - f_x(S)), \tag{24}$$

where the input values are approximated using the conditional expectation function of the original prediction model $f_x(S)$, where S represents the set of non-zeros in $z^{'}$, and $E[f(x)|x_s]$ is the expected value of the conditional function given a set of input features S. This is denoted as $f_x(S) = f(h_x(z')) = E[f(x)|x_s]$.

The weights ϕ_j are approximated using the SHAP implementation in Python introduced by Lundberg & Lee (2017). As the XGBoost model is inherently a decision-tree model, it uses the Tree-SHAP implementation to approximate the Shapley value for each feature. In the XGBoost model, $E[f(x)|x_s]$ is estimated recursively using the structure of the tree and then used as input for Equation 25. For logistic regression, the LinearSHAP implementation is used to approximate the Shapley value for each feature. In the logistic regression model, the Shapley values are directly estimated from the weight coefficients β_j of the logit model.

In this case of attribution modeling, the outcome variable is binary, therefore, raw prediction of f_x is expressed in log-odds. The feature importance is expressed as a Shapley value on the scale of log-odds and does allow for a direct comparison with the coefficients of logistic regression. As this thesis is more interested in the magnitude of the effect, the exact contribution on the probability score is less important. Therefore, the Shapley values are used to determine the magnitude of the presence of a feature j. For global feature importance per class, the sum of Shapley Values ϕ_j per feature j for each observation i is taken with regards to the presence of feature j in the online customer journey, where $x_j \in \{0, 1\}$:

$$I_{j|x_j} = \sum_{i=1}^{N} \phi_j^{(i|x_{ij})} \tag{25}$$

5 Results

This chapter presents the empirical results of the thesis. First, section 5.1 discusses the performance of the XGBoost model compared to the logistic regression model based on the predictive power using out-of-sample observations. Then, section 5.2 outlines the model explanation performance for the three methodologies, i.e., last-touch attribution, feature regression weights, and SHAP presented in the previous chapter.

5.1 Predictive performance

5.1.1 Logistic regression model

Using the trained logistic regression model to classify the instances in the hold-out data set results in a precision score of 0.658 and a recall score of 0.629. Remember the trade-off between precision and recall discussed in the previous chapter. The slightly higher precision score shows that the model marginally overpredicts the positive class compared to the negative class. Table 8 verifies this finding as the number of instances that are false positives slightly exceeds the number of instances that are false negatives. Furthermore, it should be noted that there is a severe class imbalance in the data set, as shown in Table 8, which makes it harder for the model to learn a relation. Lastly, the best threshold for the logistic regression model is 0.613, while maximizing the F_1-score. The F_1-score using this optimal threshold for the logistic regression model results in a score of 0.643.

Table 6: Confusion matrix of the logistic regression model on the hold-out set.

		Predicted	
		Conversion	No conversion
Observed	Conversion	13,473 (0.44%)	7,081 (0.23%)
	No conversion	6,984 (0.23%)	3,031,147 (99.07%)

5.1.2 XGBoost model

In Table 7, the best set of hyperparameters for the XGBoost model are presented that were determined using a randomized grid search while maximizing the F_1-score.

Table 7: Best set of hyperparameters for XGBoost model

Hyperparameter	Value
Learning rate	0.15
Maximum three depth	15.00
Minimum child weight	3.00
Minimum split loss	7.00
Scale positive weights	1.00

Using the trained XGBoost model with the best set of hyperparameters to classify the instances in the hold-out data set results in a precision score of 0.714 and a recall score of 0.687. The slightly higher precision score shows that the model marginally overpredicts the positive class compared to the negative class. Table 9 verifies this finding as the number of instances that are false positives slightly exceeds the number of instances that are false negatives. There is also severe class imbalance present in the data set, as shown in Table 9, which makes it harder for the model to learn a relation. Furthermore, the best threshold for the XGBoost model is 0.639, while maximizing the F_1-score. The F_1-score using this optimal threshold for the XGBoost model results in a score of 0.700.

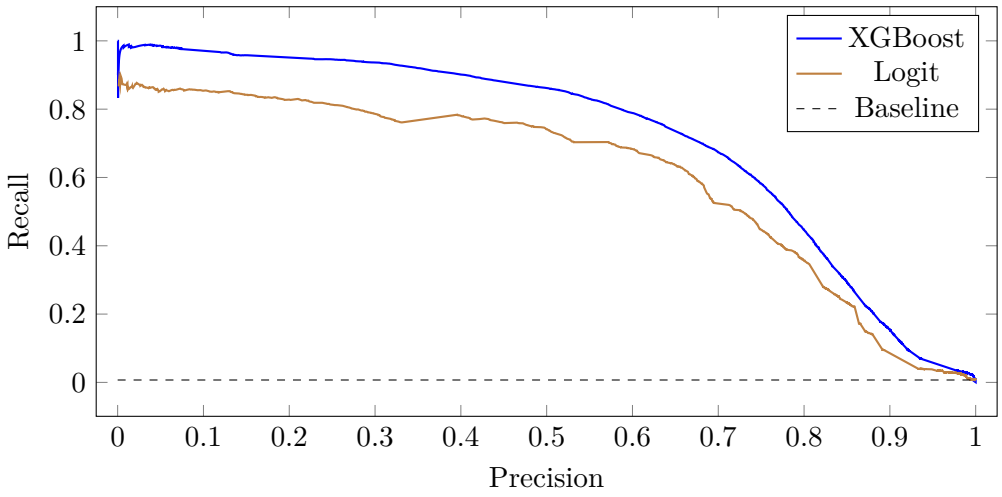
Table 8: Confusion matrix of the XGBoost model on the hold-out set.

		Predicted	
		Conversion	No conversion
Observed	Conversion	14,316 (0.47%)	7,924 (0.26%)
	No conversion	5,725 (0.19%)	3,032,406 (99.08%)

5.1.3 Model comparison

When comparing the two models, it can be seen that the XGBoost model is better able to classify unseen instances based on the higher F_1-score of 0.700 as compared to the logistic regression with a lower F_1-score of 0.639. Additionally, the recall score is higher for the XGBoost model, implying that it is able to classify 68.7 percent of the instances in the positive class in the hold-out data set correctly, whereas the logistic regression model can classify 62.9 percent of the actual positive instances correctly. Lastly, Figure 6 shows that the XGBoost model has a better classification performance despite the use of the same data set with severe imbalanced classes. It can be seen that for every possible threshold, XGBoost scores better than logistic regression. The XGBoost has an average precision score of 0.716, while the logistic regression has an average precision score of 0.607.

Figure 6: Precision-recall curve for XGBoost and logistic regression model

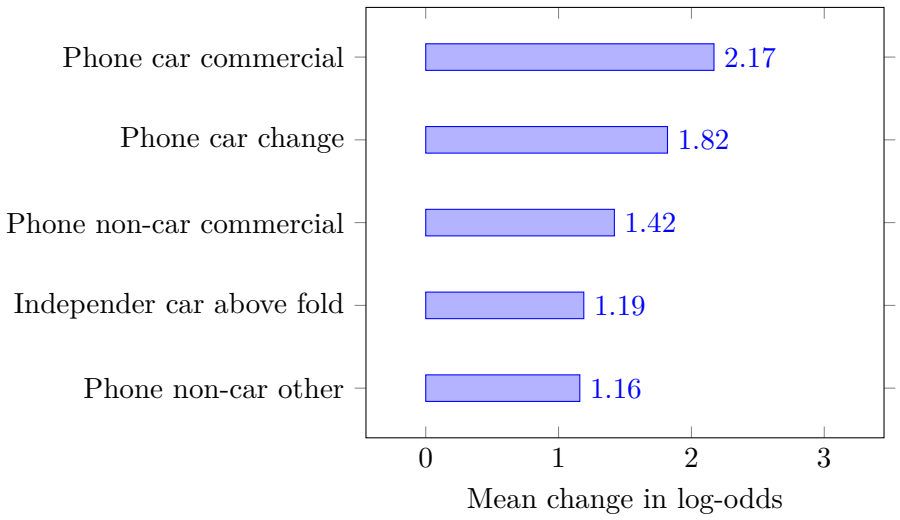


5.2 Explanatory performance

5.2.1 Last touch attribution

Performing last touch attribution on the data set, the most important feature becomes the *phone car commercial* with a mean change in log-odds of 2.17 when the feature is present in the online customer journey. Figure 7 gives an overview of the five most important features based on the absolute count of the amount of interactions.

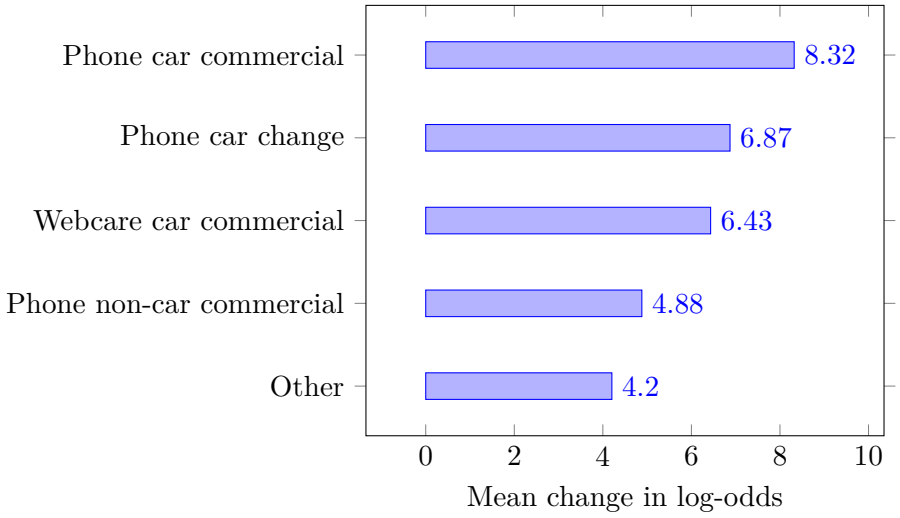
Figure 7: Five most important features based on last touch attribution



5.2.2 Feature regression weights

Logistic regression yields *phone car commercial* as the most important feature based on the positive change in log-odds when the feature is present in the online customer journey. The change in log-odds for conversion with feature *phone car commercial* is 8.32. It should be noted that the *phone car commercial* is also seen as the most important channel in the online customer journey by the last-touch attribution method. Figure 8 summarizes the five most important features found by using the weights in logistic regression.

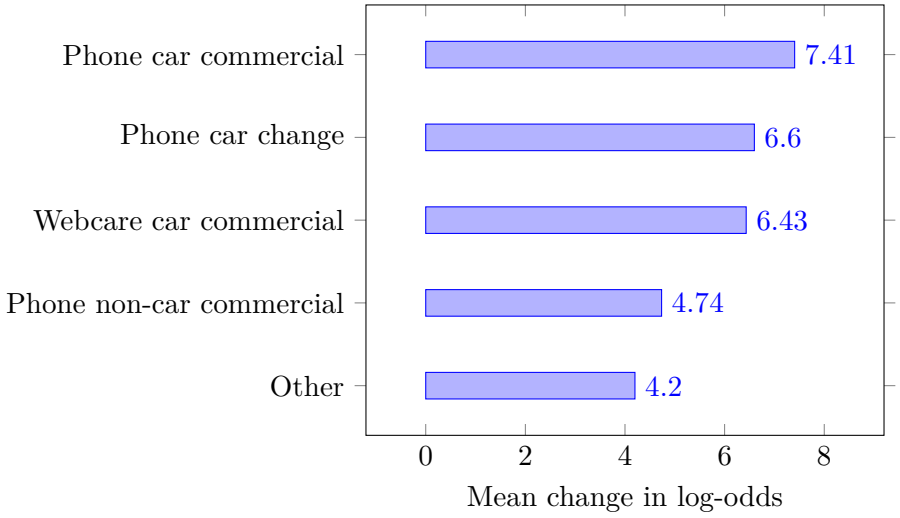
Figure 8: Five most important features based on feature weights for logistic regression



5.2.3 SHAP

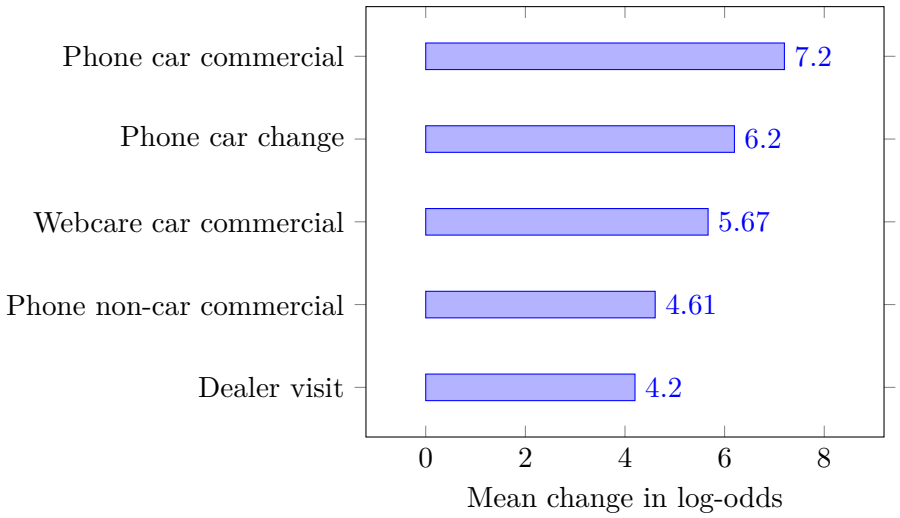
5.2.3.1 Logistic regression Using SHAP decomposition on logistic regression yields *phone car commercial* as the most important feature based on the mean Shapley value on the log-odds scale. If the feature *phone car commercial* is present in the online customer journey, then the mean increase in log-odds is 7.41. Note the consistency of the most important touchpoint *phone car commercial* with the two approaches discussed previously. Figure 9 summarizes the five most important features based based on the average impact in absolute terms on the model output magnitude in log-odds.

Figure 9: Five most important features based on logistic regression using SHAP



5.2.3.2 XGBoost Using SHAP decomposition on the XGBoost model yields *phone car commercial* as the most important feature based on the mean Shapley value on the log-odds scale. If the feature *phone car commercial* is present in the online customer journey, then the mean increase in log-odds for conversion is 7.20. Once again, note the consistency for the most important feature *phone car commercial* with the previously discussed approaches. Figure 10 summarizes the five most important features based on the mean change in log-odds for conversion if a feature is present in a customer journey.

Figure 10: Five most important features based on XGBoost using SHAP

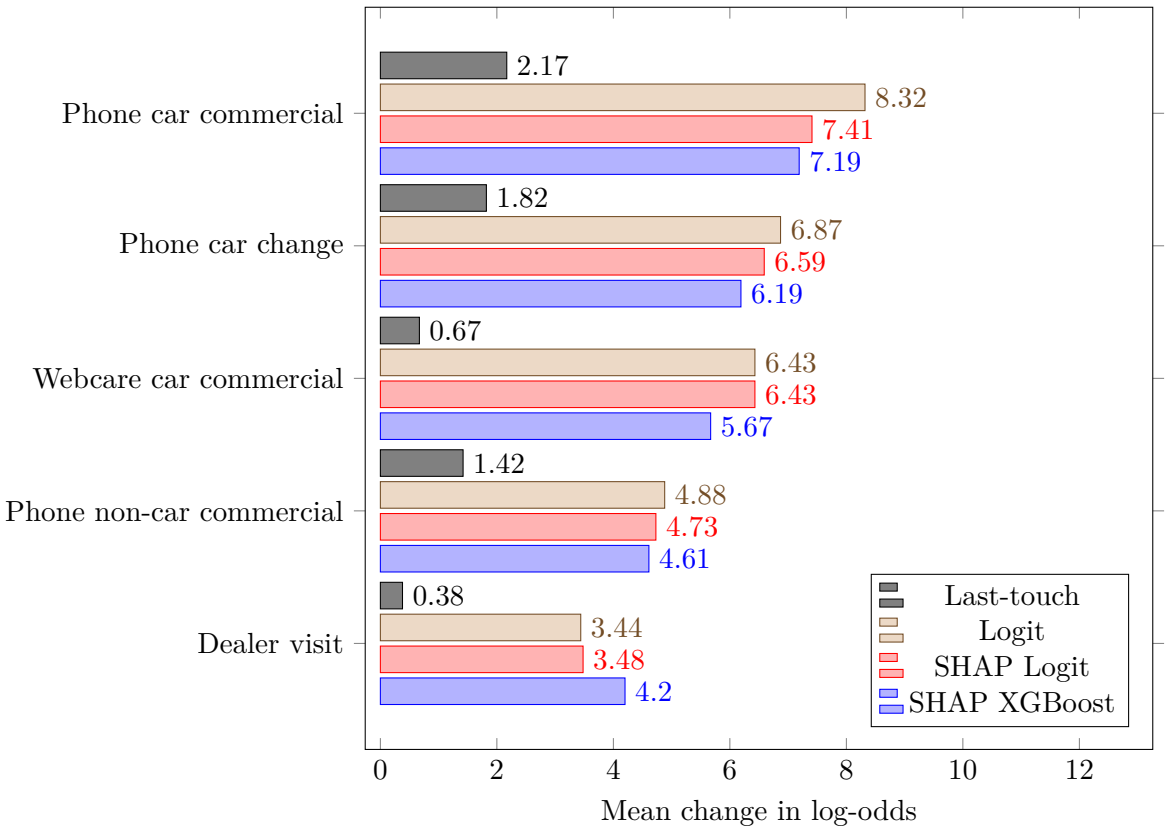


5.2.4 Explanation comparison

When comparing the three methods, it can be seen that the feature *phone car commercial* comes up as the most important channel in the online customer journey for all four approaches. However, it should be noted that the last-touch attribution is solely focusing on the number of interactions and whether the interaction is the last step. The fact that *phone car commercial* is the last interaction has to do with the nature of the product, for which lots of consumers prefer to gather information from a customer representative before converting. Nevertheless, the two data-driven approaches, feature regression weights and model + SHAP, also yield the same result.

Figure 11 summarizes the five most important features according to the three feature importance approaches discussed previously. Last-touch attribution performs the worst for feature importance extraction as the importance is determined a priori and leads to feature importances that do not represent the actual distribution in the data. Overall, the feature importances between the logistic regression coefficients and SHAP values are consistent among these two data-driven two methods. Moreover, it can be seen that the differences in the mean change in log-odds are marginal. See Appendix C for the total overview of all features in mean change in log-odds for the four approaches.

Figure 11: Five most important features for feature importance approaches



6 Conclusion and Discussion

6.1 Main findings

This thesis studied the use of the explainable machine learning method, SHAP, in combination with a more complex machine learning method for the creation of an attribution model. It was put forward that A more complex machine learning is able to capture relationships more sufficient than simpler machine learning approaches. It is shown that the use of a well-tuned XGBoost model has predictive superiority over the simpler logistic regression. The feature importance results are consistent among XGBoost/Logit + SHAP and the coefficient interpretation in the built-in logistic regression and are more accepted as 'true' as compared to last-touch attribution since last-touch attribution lacks the use of data to determine the feature importance.

6.2 Discussion

This research started with the objective to investigate the possibility of an attribution model that combines a high degree of accuracy while still allowing for interpretability. Existing marketing literature has formulated five criteria to which an attribution model should measure up. The possible candidates are the *logistic regression* model and the *XGBoost* model that both have the capability to determine their classification performance. Likewise, there is growing research on model interpretability, which has yielded methodologies regarding feature importance extraction of black-box models. The methodology used is *SHAP*, which uses the Shapley value to generate the importance of a feature. The results were in line with the expectations of the research. The XGBoost outperforms the logistic regression based on classification performance. However, in this thesis, the difference in predictive performance between logistic regression and XGBoost is marginal as no other variables other than the presence in the online customer journey were taken into account. The feature importance yielded by Logit/XGBoost + SHAP is consistent with the standard feature importance approach of logistic regression, which confirms that these two data-driven approaches are persistent and logical. In the theoretical evaluation, it was stated that rule-based methodologies, such as last-touch attribution, are fundamentally flawed due to the lack of the use of data. It is confirmed that this holds true for last-touch attribution as the feature importance yielded by this method is not consistent. The major advantage of the use of XGBoost in combination with SHAP, it that it allows one to create more complex models while still maintaining interpretability. Overall, it can be concluded that SHAP, in combination with a black-box model, e.g., XGBoost, can be seen as a new means to perform attribution modeling in the field of marketing.

6.3 Managerial implications

This research provides evidence for the use of more black-box models, such as XGBoost, while still providing interpretability that fits the purpose of attribution modeling in marketing. Despite the fact that this research only examined the feature importance based on the presence of an interaction within a customer journey, the strength of using black-box models is the addition of more user-level variables, e.g., age. The application of a more complex model allows for a deeper understanding of which channels, in combination with other user-level variables within the online customer journey, are leading to conversion through the use of SHAP. It opens up the possibility to create a holistic model that allows seeing whether a campaign for a certain audience is effective whilst taking all the other interactions with other channels into account. Organizations could benefit from this knowledge by making a better-informed decision on what to spend budget more effectively when wanting to drive the conversion rate up.

6.4 Limitations

A limitation of this research is that the order of the interactions within the customer journey was not taken into account, whereas it could be argued that the order in which the customer interacts with a channel plays an important role with respect to the nature of the customer journey. It would, therefore, be pertinent to extend the research on the introduced XGBoost + SHAP model to investigate the effect of channel ordering. Another limitation of this study is that no other variables were taken into account other than the presence of interaction within an online customer journey. Further research could include more variables, such as behavioral and demographic variables, that allow for deeper mapping within the online customer journey. Using the XGBoost approach allows for more complex relations, which can be used with SHAP to find complex underlying dependencies between variables that are related to (non-)converting journeys. Lastly, this research did not take exposure to offline channels within the customer journey into account but rather solely focused on the online customer journey based on online click behavior. It could be said that the 'true' importance of a channel can be better approximated if the effects of offline channels are also taken into consideration.

Appendix

Appendix A: Description of touchpoints

Table 9: Overview of online marketing touchpoints in the data set

Touchpoint name	Definition	Initiation type
Affiliate	Affiliate program of the financial provider redirects a visitor to their website of their focal brand for a reward given to the publisher.	Customer
App	App of the insurer's focal brand is a way on how users can interact with the company that can generate online sales and leads.	Customer
Dealer	Car dealer tries to sell car insurance for the brand to new car owners when picking up the car at the dealer.	Firm
Direct[1]	Direct visit to the website of the focal brand done by a customer typing in the brand's domain in the browser's address bar.	Customer
Display	Display advertising from the insurer containing a graphic message on it to site visitor displayed on a website.	Firm
Email[2]	Sending commercial messages via email to potential or current customers to generate leads and online sales.	Firm
File	Existing customers having their extension of car insurance executed by the insurer.	Customer/Firm
Independer[3]	Insurance-specific comparison website that redirects visitor the insurer's website for a reward given to the publisher (affiliate-based).	Customer
Referral	Word-of-mouth marketing to generate new sales and leads by incentivizing current customers by the insurer.	Customer
Phone[4]	Phone contact with the customer representative of the brand to either generate sales and leads or to change or terminate insurance contract.	Customer/Firm
Post	Physical mail post to inform customers about their current products / services or to generate news sales and leads.	Customer
Search[5]	Use of certain keywords in search engines by the consumer in where the results shown by the search engine generate online sales and leads by redirecting to the website.	Customer
Social	Social media platforms, such as, Facebook, Twitter, and LinkedIn to communicate commercial messages for online sales and leads.	Firm
Webcare[6]	Web contact with the customer representative of the brand to either generate sales and leads or to change or terminate insurance contract.	Customer
Other	Other includes all forms of advertising the insurer uses that do not fit in one of the previous mentioned categories.	Customer/Firm

[1] Direct is mapped by car/non-car

[2] Email is mapped by inbound/outbound * acquistion/commercial/newsletter/satisfaction/service/welcome/other

[3] Independer is mapped by car/non-car * above_fold/below_fold

[4] Phone is mapped by car/non-car * commercial/terminate/change/other

[5] Search is mapped by organic * car/non_car + paid * branded/non_branded

[6] Webcare is mapped by car/non-car * commercial/terminate/change/other

Appendix B: Interactions per touchpoint

Table 10: Number of interactions per touchpoint

Touchpoint	# Visits	% Visits
Affiliate	2,091,709	1.68%
App	99,928	0.08%
Dealer	41,426	0.03%
Direct	22,507,596	18.03%
Display	984,289	0.79%
Email	62,060,000	49.72%
File	803,069	0.64%
Independer	12,011,999	9.62%
Referral	3,384,166	2.71%
Phone	3,219,844	2.58%
Post	2,908,753	2.33%
Search	13,800,000	11.06%
Social	416,485	0.33%
Webcare	470,892	0.38%
Other	16,352	0.01%

Appendix C: Comparison table in mean change in log-odds per touchpoint and approach

Table 11: Comparison table in mean change in log-odds per touchpoints [A-I]

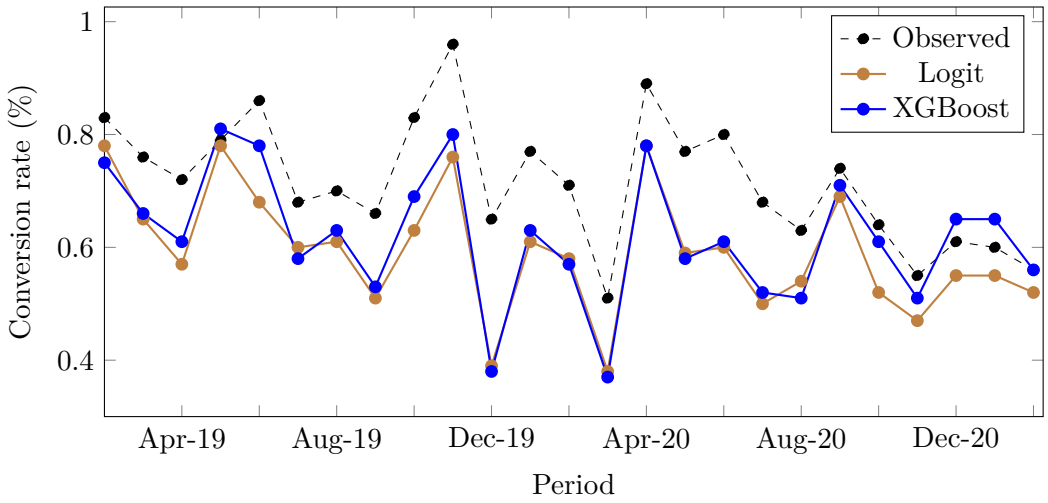
Touchpoint	Last-touch	Logit	Logit + SHAP	XGBoost + SHAP
Affiliate	0.516	1.343	1.320	1.213
App	-0.033	2.790	2.790	2.307
Dealer visit	0.385	3.438	3.438	4.201
Direct car	0.910	2.183	1.986	2.177
Direct non-car	0.213	-0.260	-0.197	0.066
Display	-1.134	0.173	0.172	0.616
Dossier	-0.224	-0.717	-0.710	0.062
Email	-0.241	-0.543	-0.391	-0.085
Email commercial	0.299	0.493	0.478	0.895
Email inbound acquistion		-1.581	-1.581	0.000
Email inbound commercial		-0.256	-0.233	-0.095
Email inbound newsletter		-0.184	-0.182	0.001
Email inbound satisfaction		0.091	0.091	0.232
Email inbound service		-1.173	-1.173	-0.575
Email inbound welcome		-0.869	-0.869	-0.510
Email newsletter	-1.757	-0.992	-0.972	-0.873
Email other		-0.166	-0.166	-0.181
Email outbound acquisition		0.586	0.586	0.000
Email outbound commercial		-0.031	-0.030	0.023
Email outbound newsletter		-0.101	-0.094	-0.056
Email outbound satisfaction		-0.193	-0.193	-0.176
Email outbound service		-1.231	-1.231	-0.607
Email outbound welcome		-1.454	-1.454	-0.666
Independer car above fold	1.186	1.703	1.413	2.216
Independer car below fold	0.009	1.057	0.994	1.147
Independer non-car above fold	-0.638	0.140	0.135	0.438
Independer non-car below fold	-1.722	0.113	0.110	0.059

Table 12: Comparison table in mean change in log-odds per touchpoints [J-Z]

Touchpoint	Last-touch	Logit	Logit + SHAP	XGBoost + SHAP
Phone car change	1.821	6.872	6.597	6.198
Phone non-car change	0.819	2.453	2.404	2.406
Phone car commercial	2.177	8.322	7.406	7.200
Phone non-car commercial	1.429	4.881	4.735	4.606
Phone car other	-0.084	0.890	0.881	1.402
Phone non-car other	1.163	1.348	1.226	1.739
Phone car terminate	0.556	3.123	3.060	2.691
Phone non-car terminate	-0.477	0.152	0.152	0.759
Post	-0.027	-1.090	-1.068	-0.324
Referral	0.587	0.699	0.650	1.057
Search car organic	0.650	2.068	1.985	2.260
Search non-car organic	-0.407	-0.696	-0.641	-0.126
Search car branded paid	1.006	3.006	2.946	2.817
Search non-car branded paid	-0.140	-0.028	-0.028	0.472
Search car non-branded paid	0.697	1.783	1.712	2.411
Search non-car non-branded paid	-1.629	-1.612	-1.451	-0.901
Social	-1.922	-0.969	-0.969	-0.312
Webcare car change	0.049	2.969	2.969	2.436
Webcare non-car change	-1.178	-0.243	-0.243	0.161
Webcare car commercial	0.676	6.434	6.434	5.668
Webcare non-car commercial	-0.237	2.408	2.408	2.833
Webcare car other	-1.595	-0.337	-0.337	-0.157
Other	-0.914	4.205	4.205	3.660

Appendix D: Conversion rate over time per model

Figure 12: Conversion rate (%) over time per model



Both the XGBoost and logistic regression model underestimate the instances that convert as the conversion rate is in most cases lower than the observed conversion rate. The conversion rate of both the XGBoost and logistic regression follow a pattern that is similar to the actual observed conversion rate, indicating that the two model are able to generalize the relations found over time. However, it seems that XGBoost overstimates more than logistic regression as can be seen in Figure 12. Most notably during the beginning of the year 2021.

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, *6*, 52138–52160.

Anderl, E., Becker, I., von Wangenheim, F., & Schumann, J. H. (2016). Mapping the customer journey: Lessons learned from graph-based online attribution modeling. *International Journal of Research in Marketing*, *33*(3), 457–474. Retrieved from https://www.sciencedirect.com/science/article/pii/S0167811616300349

Arora, N., Dreze, X., Ghose, A., Hess, J. D., Iyengar, R., Jing, B., Joshi, Y., et al. (2008). Putting one-to-one marketing to work: Personalization, customization, and choice. *Marketing Letters*, *19*(3), 305. Retrieved from https://doi.org/10.1007/s11002-008-9056-z

Bleier, A., & Eisenbeiss, M. (2015). Personalized online advertising effectiveness: The interplay of what, when, and where. *Marketing Science*, *34*(05), 627–777. Retrieved from https://doi.org/10.1287/mksc.2015.0930

BOVAG. (2021c). *Resultaat van lockdown zichtbaar in forse daling occasionverkoop* (Press release). De Bovag, RAI Vereniging en databureau RDC.

BOVAG. (2021a). *Voor het eerst meer dan 2 miljoen gebruikte personenauto's verkocht* (Press release). De Bovag, RAI Vereniging en databureau RDC.

BOVAG. (2021b). *Persbericht verloop aantal registraties personenauto's (2019-2020)* (Report). De Bovag, RAI Vereniging en databureau RDC.

Bughin, J. (2015). Brand success in an era of digital darwinism. Retrieved from https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/brand-success-in-an-era-of-digital-darwinism

Cauchy, A. (1847). Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, *25*(1847), 536–538.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Dalessandro, B., Perlich, C., Stitelman, O., & Provost, F. (2012). Causally motivated attribution for online advertising. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1–9. Retrieved from https://doi.org/10.1145/2351356.2351363

Danaher, P. J., & Heerde, H. J. van. (2018). Delusion in attribution: Caveats in using attribution for multimedia budget allocation. *Journal of Marketing Research*, *55*(5), 667–685. Retrieved from https://doi.org/10.1177/0022243718802845

de Haan, E., Wiesel, T., & Pauwels, K. (2016). The effectiveness of different forms of online advertising for purchase conversion in a multiple-channel attribution framework. *International*

Journal of Research in Marketing, *33*(3), 491–507. Retrieved from https://www.sciencedirect.com/science/article/pii/S0167811615001421

Deloitte. (2021, April). Retrieved from https://vianederland.nl/kennisbank/online-ad-spend-study-2020/

Durai, T., & King, R. (2015). Impact of digital marketing on the growth of consumerism. *Madras University Journal of Business and Finance*, *3*(2), 94–104.

Freitas, A. A. (2014). Comprehensible classification models: A position paper. *SIGKDD Explor. Newsl.*, *15*(1), 1–10. New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2594473.2594475

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of statistics*, 1189–1232. JSTOR.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Comput. Surv.*, *51*(5). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3236009

Hallikainen, H., Alamäki, A., & Laukkanen, T. (2019). Individual preferences of digital touchpoints: A latent class analysis. *Journal of Retailing and Consumer Services*, *50*, 386–393. Retrieved from https://www.sciencedirect.com/science/article/pii/S0969698918305605

Hayes, J. L., Golan, G., Britt, B., & Applequist, J. (2020). How advertising relevance and consumer–brand relationship strength limit disclosure effects of native ads on twitter. *International Journal of Advertising*, *39*(1), 131–165. Routledge. Retrieved from https://doi.org/10.1080/02650487.2019.1596446

Howard, J., & Sheth, J. (1970). The theory of buyer behavior. *Journal of the American Statistical Association*, *65*(331), 1406–1407.

Jiang, Z., & Benbasat, I. (2007). Investigating the influence of the functional mechanisms of online product presentations. *Information Systems Research*, *18*, 454–470. Retrieved from https://doi.org/10.1287/isre.1070.0124

Kannan, P. K., & Li, A. (2017). Digital marketing: A framework, review and research agenda. *International Journal of Research in Marketing*, *34*(1), 22–45.

Kearns, M., & Valiant, L. (1989). Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM*, *41*(1), 67–95. New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/174644.174647

Kelly, J., Vaver, J. G., & Koehler, J. (2018). A causal framework for digital attribution.

Lemon, K. N., & Verhoef, P. C. (2016). Understanding customer experience throughout the customer journey. *Journal of Marketing*, *80*(6), 69–96. Retrieved from https://doi.org/10.1509/

jm.15.0420

Letang, V., & Stillman, L. (2020). *Global advertising forecast - winter update* (Report Update). MAGNA Intelligence.

Lipton, Z. C. (2016). The mythos of model interpretability. Retrieved from https://doi.org/1606.03490

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Proceedings of the 31st international conference on neural information processing systems*, NIPS'17 (pp. 4768–4777). Red Hook, NY, USA: Curran Associates Inc.

Magna. (2021, June). Retrieved from https://magnaglobal.com/magna-global-advertising-forecasts-june-2021/

Molnar, C. (2018). *Interpretable machine learning: A guide for making black box models explainable.* Retrieved from https://christophm.github.io/interpretable-ml-book/

Neslin, S. A., Grewal, D., Leghorn, R., Shankar, V., Teerling, M. L., Thomas, J. S., & Verhoef, P. C. (2006). Challenges and opportunities in multichannel customer management. *Journal of Service Research*, *9*(2), 95–112. Retrieved from https://doi.org/10.1177/1094670506293559

Nielsen. (2021, June). Retrieved from https://nederlandsmedianieuws.nl/media-nieuws/Onderzoek-Nielsen-Netto-mediabestedingen-in-2020-gedaald-met-63-procent-tot-44-miljard-euro/

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, KDD '16 (pp. 1135–1144). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2939672.2939778

Sapp, S., & Vaver, J. G. (2016). Toward improving digital attribution model accuracy. Retrieved from https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45766.pdf

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*(2), 197–227. Retrieved from https://doi.org/10.1007/BF00116037

Shao, X., & Li, L. (2011, August). Data-driven multi-touch attribution models. Retrieved from https://doi.org/10.1145/2020408.2020453

Singh, K., Vaver, J. G., Little, R. E., & Fan, R. (2018). Attribution model evaluation. Retrieved from https://storage.googleapis.com/pub-tools-public-publication-data/pdf/de1c3ab14fd52301fb193237fdffd45352159d5c.pdf

Statista. (2021). Global digital population as of january 2021. Retrieved from https://www.statista.com/statistics/617136/digital-population-worldwide/