

University of Groningen
&
University of Münster

Master Thesis

Leveraging Unstructured Data in Channel Attribution Theory

Chair: Prof. Dr. Jaap Wieringa
Faculty of Economics and Business

Prof. Dr. Raoul Kübler
Marketing Center Münster

Supervisor: Prof. Dr. Jaap Wieringa
Issuing Date: February 12, 2019
Due Date: June 17, 2019

Submitted by: Felix Lehmkuhle
Aquamrijnstraat 367
9743 PJ Groningen

Phone: +49 157 88928717
E-mail: f.lehmkuhle@student.rug.nl
Student number: S3836207 (Groningen)
418481 (Münster)

Executive Summary

Digitalization disrupts the business landscape in nearly every area – it poses both major challenges and great opportunities. The extensive volume and velocity of information regularly overwhelms customers’ decision-making process. An effective marketing channel strategy is more important than ever to attract awareness, increase consideration and finally stimulate a purchase interest. As the decision process varies along the purchase funnel, an analysis of channel effects on different stages reveals valuable insights.

To withstand the demanding developments in the age of big data, corporations need to value and exploit data analytics as a key function in successful business strategies. The vast amount of structured and unstructured data offers great potential to understand customers’ decision-making process in more detail. In particular, unstructured data can serve as a tool to improve channel attribution accuracy and separate channel effects in distinctive subgroups. However, corporations frequently do not exploit all available resources but limit their analysis to structured data.

A holistic literature review proves that also marketing research does not analyze channel effectiveness along a purchase funnel while benefiting from the incorporation of unstructured data.

This study expands a basic clickstream dataset by additional variables constructed from image and text data. I use the Microsoft Azure Face Recognition API to categorize image pictures and apply a sentiment analysis to group Google Ad descriptions into three categories each. Subsequently, I run multiple Random Forest algorithms and use the predicted outcomes to calculate Shapley Values displaying channels’ effectiveness. The results indicate the potential of unstructured data to improve channel attribution accuracy. Further, the categorization of paid search advertising based on emotionality displayed in Google Ad descriptions reveals a higher impact of neutral than strong positive or positive ads. These findings recommend the advantage of presenting insightful information and specific discount statements in ad descriptions.

In conclusion, this study aligns channel attribution theory with a purchase funnel framework and elaborates on the potential of unstructured data. In doing so, it partly improves attribution accuracy and detects effects of search advertising subgroups.

Preface

Since I strongly believe in the power of data analytics, I value both traditional econometric methodologies and latest developments in machine learning approaches as tools to manage current and upcoming challenges in the age of Big Data.

During my Bachelor program in Business Administration at the University of Münster, I realized my passion for quantitative marketing. As I did not learn how to develop own quantitative models as an undergraduate, I was glad to elaborate on this topic during the following Master program in Marketing and Finance. At that time, I started to learn about Machine Learning and Artificial Intelligence, which are amazing fields full of opportunities. Willing to further improve my own knowledge, the Marketing Intelligence program at the University of Groningen provided the great chance to spend my second master year abroad. During the last year lectures as “Data Science and Marketing Analytics” or “Market Models” did not only teach me to extract valuable information from large databases but also provided me with the chance to learn in an inspiring environment. Being aware of the excellent study conditions, I saw the master thesis as a possibility to further improve my skills in data analytics. The collaboration with a data science start-up and an international Dutch cooperation created optimal working conditions. I am very grateful for the collaboration over the past months.

My special gratitude goes to Prof. Dr. Jaap Wieringa who did not only served as an excellent thesis supervisor but also helped me as an outstanding lecturer. Further, I want to stress the great support of Piet Peeperkorn. He was always accessible to discuss problems and enormously motivated to answer my questions. Finally, I express my thanks to Gilian Ponte, Amira Awad, Nerina Dombrink, and Sophie Ladwein for always having an open ear and the perseverance for extensive discussions.

I am very grateful for the last two years which not only posed an exciting intellectual challenge but also helped to grow my own personality.

Thank you



Felix Lehmkuhle

Table of Contents

Table of Contents	I
List of Figures	III
List of Tables.....	IV
List of Abbreviations.....	V
List of Symbols	VI
List of Appendices.....	VII
1 Introduction	1
2 Conceptual background	4
2.1 Terminology	4
2.2 Literature review	4
2.2.1 Purchase funnel concept.....	5
2.2.2 Channel attribution concept.....	7
2.2.2.1 Channel overview.....	7
2.2.2.2 Channel attribution in practice	10
2.2.2.3 Channel attribution in research	12
2.3 Methodological framework.....	18
2.3.1 Framework overview.....	18
2.3.2 Framework description.....	21
3 Data.....	22
3.1 Data source.....	22
3.2 Data description	23
3.2.1 Purchase funnel stages.....	24
3.2.2 Channels	25
3.2.3 Situational and customer characteristics	27
3.3 Dataset conception	28
4 Methodology.....	30

4.1	Unstructured data analysis	30
4.1.1	Categorization of image data.....	31
4.1.2	Categorization of Google Ad data.....	33
4.2	Structured data analysis.....	36
4.2.1	Random Forest	37
4.2.2	Shapley Value.....	39
5	Results	41
5.1	Unstructured data analysis	41
5.1.1	Categorization of image data.....	41
5.1.2	Categorization of Google Ad data.....	42
5.2	Structured data analysis.....	43
5.2.1	Random Forest	43
5.2.2	Shapley Values.....	46
6	Discussion.....	49
6.1	Conclusion.....	49
6.2	Managerial implications.....	51
6.3	Limitations and future research.....	52
	References	55
	Appendix	67

List of Figures

Figure 1: Channel overview	10
Figure 2: Modeling framework	20
Figure 3: Purchase funnel stages per day	25
Figure 4: Channel interactions per day.....	26
Figure 5: Situational and customer characteristics.....	28
Figure 6: Training dataset construction.....	30
Figure 7: Emotionality recognition	33
Figure 8: Prediction dataset construction	39
Figure 9: Emotionality of product pictures	42
Figure 10: Emotionality of Google Ad descriptions	43
Figure 11: Variable importance.....	45
Figure 12: Shapley Values	48

List of Tables

Table 1: Funnel frameworks.....	7
Table 2: Literature overview	16
Table 3: Variable operationalization	23
Table 4: Prediction results	45
Table 5: Shapley Values.....	48

List of Abbreviations

CPC	cost-per-click advertising
DCNN	deep convolutional neural network
e.g.	for example
EX	exponential method
FT	first touch method
i.a.	amongst others
ID	identifier
i.e.	that is
LIME	Local Interpretable Model-agnostic Explanations
LT	last touch method
MAE	mean absolute error
RMSE	room mean squared error
SD	structured data
SHAP	Shapley Additive Explanations
U	uniform method
UD	unstructured data
URLs	Uniform Resource Locators
vs.	versus

List of Symbols

number of

List of Appendices

Appendix A: Outlier analysis	67
Appendix A.A : Basic dataset	67
Appendix A.B : Customer journey dataset.....	67
Appendix C: Programming code	70

1 Introduction

“Half the money I spend on advertising is wasted; the trouble is I don't know which half” – this famous dictum emphasizes the challenge of measuring advertisements' effectiveness. Today, the topic seems to be more relevant than ever as the impact of marketing is highly dependent on its accountability (Rust et al. 2004, p. 76; Verhoef and Leeflang 2009, p. 14; Webster 2006, p. 6). International corporations like Procter & Gamble cut their digital advertising budget by over 100 million US Dollar as they have not been able to accurately measure its performance (Burell and Terlep 2017).

Channel attribution models usually rely on simple heuristics like the first or last touch point approach (Anderl et al. 2016, p. 457; Berman 2018, p. 771; Li and Kannan 2014, p. 41). However, allocating a final purchase to a single advertising effect does not account for the impact of all other channels, customers get in touch with during their decision-making process. Hence, corporations do not capture several channel effects although they actually spend money on them. A comprehensive attribution methodology is indispensable to precisely measure advertisements' return on investment.

Marketing research points out the importance of channel attribution to optimize budget allocation (e.g. Barajas et al. 2016; Danaher and van Heerde 2018; Kannan and Li 2017; Kannan, Reinartz, and Verhoef 2016; Li et al. 2016; Webster 2006). Nonetheless, Danaher and van Heerde (2018) contradict this statement and argue why profit-maximizing instead of attribution-based allocation should be used. They stress the descriptive character of attribution models which do not provide information about an optimal distribution. Even though descriptive models indeed do not solve an optimization problem, they disclose channels' effectiveness. Subsequently, the revealed information can be used to adequately allocate the advertising budget. Thus, attribution-based methodologies still provide valuable information to optimize channel usage.

Nowadays, an extensive database allows to track customer journeys in detail. This offers new opportunities to measure channel attribution. Digitalization fuels the development while exerting an enormous impact on the corporate landscape in total and on marketing in particular (Bradlow et al. 2017; Erevelles, Fukawa, and Swayne 2016; Moe and Ratchford 2018). As many marketers especially value the

usage of online advertising, this study scrutinizes the effect of digital channels (Braun and Moe 2013, p. 753).

Digitalization expands the purchase funnel by adding multiple new touchpoints that affect customers along the entire decision process (Lemon and Verhoef 2016, p. 77). To date attribution models often limit their analysis to the final purchase decision. However, an incorporation of different funnel stages helps to get a better understanding of customers' decision process as their emotional states change along the shopping process (Frambach, Roest, and Krishnan 2007; Gardial et al. 1994; Haan, Wiesel, and Pauwels 2016; Kannan, Reinartz, and Verhoef 2016; Mittal, Kumar, and Tsiros 1999).

New technological possibilities increase the volume, velocity and variety of data. Additional to the tremendous amount of structured data (SD), unstructured data (UD) constantly gains importance both in research and practice (Bradlow et al. 2017, p. 88; Erevelles, Fukawa, and Swayne 2016, p. 898). As it delivers insights beyond numeric information (e.g. corporations' communication style), attribution research should exploit this potential (Bradlow et al. 2017, p. 88; Erevelles, Fukawa, and Swayne 2016, p. 898). In spite of these benefits, many marketing departments and researchers do not incorporate UD in channel attribution theory as it is quite challenging to handle compared to SD (Blumberg and Atre, p. 42; Erevelles, Fukawa, and Swayne 2015, p. 898). Thus, both marketing managers and researchers miss promising opportunities to improve attribution models.

Research questions

This study focuses on the potential of combining SD with UD in channel attribution research and specifically scrutinizes the following two research questions:

- 1. Does the incorporation of unstructured data increase channel attribution accuracy along an e-commerce website funnel?*
- 2. Does unstructured data help to evaluate the effectiveness of channel sub-groups along an e-commerce website funnel?*

Results

Six additional variables, representing clicks on product images (happy, mixed, neutral) and search advertising categories (strong positive, positive, neutral), increase

the prediction accuracy of customers' awareness and consideration during the purchase process. The improved estimates cause a considerable increase in channel attribution precision compared to an analysis solely based on SD.

Besides, the results disclose a benefit of neutral Google Ad descriptions compared to strong positive and positive versions in cost-per-click advertising (CPC). The usage of neutral descriptions consistently leads to a higher customer awareness, consideration, and purchase intention along the purchase funnel.

Contribution

As marketeers face the challenge to allocate the advertising budget in the most effective way, a well-grounded understanding of channel attribution is indispensable. Based on a combination of a structured clickstream dataset with unstructured image and text data, this study carves out how corporations can benefit from a vast amount of data.

Facial expressions of fashion models in product images represent emotional features on the e-commerce platform. As hedonic website characteristics influence customers' clicking behavior, image characteristics provide insightful information (Childers et al. 2001, p. 527; Cyr et al. 2009, pp. 539–540; Rosen and Purinton 2004, p. 784). Further, an inclusion of Google Ad descriptions subdivides CPC into different classes.

As additional variables constructed from UD (image and text features) improve the prediction of customers' behavior along an e-commerce website funnel, I subsequently attribute better estimates to preceding channel contacts. Besides, the categorization of CPC subgroups reveals detailed channel effects and helps to optimize the usage of search engine advertising.

In conclusion, this study expands current attribution literature by incorporating UD in order to improve channel attribution accuracy and analyze the effects of separate CPC subgroups. Further, it provides a well-structured framework based on a purchase funnel concept and applies latest machine learning algorithms.

Structure

After the introduction, the second chapter deals with the conceptual background including a definition of the key terminology, an extensive literature review and a

derivation of the underlying framework. Chapter three describes the dataset and explains the variable construction used for the analysis in chapter four. Before tree-based machine learning methods analyze the data, categorization techniques carve out patterns in UD as additional explanatory variables. Chapter five delineates the results and identifies channel impacts on customers' purchase behavior. Finally, chapter six concludes the findings, presents managerial implications and discusses limitations as well as opportunities for future research.

2 Conceptual background

2.1 Terminology

Before I combine SD with UD and evaluate channels' effectiveness, a distinct definition of the key terminology is valuable. Therefore, I differentiate *structured data* from *unstructured data* and clearly specify the *channel* term.

The exact distinction between both data types (structured versus unstructured) is blurry, whereat highly UD is a concurrent, nonnumeric representation of multiple dimensions (Balducci and Marinova 2018, pp. 558–560). As it is difficult to capture UD in rows and columns, meta-data helps to organize the information in a database (Blumberg and Atre, p. 42). UD expands SD by audios, videos, images and textual information. In total, it accounts for approximately 95% of the entire available data volume, commonly referred to as big data (Erevelles, Fukawa, and Swayne 2015, p. 898; Gandomi and Haider, pp. 137–138). In this study, I specify UD as information represented in images and text data.

Although researchers regularly devote their interest to marketing channels, they usually do not start with a specific definition of the key terminology (e.g. Barajas et al. 2016; Danaher and van Heerde 2018; Kannan and Li 2017). Instead, they use “channel” as an umbrella term encompassing all direct and indirect touchpoints customers get in touch with along the entire purchase process. In this study I base the definition on the conceptualization provided by Neslin et al. (2006, p. 96) and define a “channel” as an “*online medium through which customers can enter an e-commerce platform*”.

2.2 Literature review

A channel attribution framework based on purchase funnel stages ensures a well-structured analysis. I describe a general funnel concept and present all attribution

studies applying it so far. Afterwards, I focus channels commonly discussed in the marketing literature. As the investigation of channels' effectiveness is much-noticed, I explain attribution theories most often used in practice and conclude with theoretical methodologies. In doing so, I highlight existing research gaps and carve out the main contributions of this study.

2.2.1 Purchase funnel concept

Customers' decision journey includes touchpoints with various channels along the purchase process (Lemon and Verhoef 2016, p. 77). Instead of a direct choice whether or not to purchase something, the decision process starts on a broader base. Typically, customers encounter a pre-purchase, purchase and post-purchase stage (Frambach, Roest, and Krishnan 2007, p. 26; Gensler, Verhoef, and Böhm 2012, p. 994; Lemon and Verhoef 2016; Neslin et al. 2006, p. 97). Due to information gathered on each level, customers stepwise adapt their decisions (Huneke, Cole, and Levin 2004, p. 67).

At the pre-purchase stage customers become aware of different products by gathering all kinds of information (Balasubramanian, Raghunathan, and Mahajan 2005, pp. 16–17). Using this knowledge, they create a consideration set containing all products of interest. Subsequently, they evaluate the alternatives and decide for or against a product at the purchase stage (Frambach, Roest, and Krishnan 2007, p. 29; Lemon and Verhoef 2016, p. 77). In doing so, they compare all advantages and disadvantages or just rely on emotional buying impulses. Finally, customers reach the post-purchase step that includes all touchpoints after the purchase (Frambach, Roest, and Krishnan 2007, p. 30; Lemon and Verhoef 2016, p. 77). Corporations engage their customers to create a long-term relation and motivate them for a re-purchase (Frambach, Roest, and Krishnan 2007, p. 30).

Although most studies in attribution research do not follow a funnel framework, some researchers apply it as a general conceptualization. Table 1 provides a chronological overview of all research papers incorporating a funnel framework in attribution theory.

While Wiesel, Pauwels, and Arts (2011) differentiate between “visits – leads – quote requests – orders”, Abhishek, Fader, and Hosanagar (2012) structure their

analysis due to customer engagement states “disengaged – active – engaged – conversion”. Li and Kannan (2014) base their work on the stages “consideration – visit – purchase”, Hoban and Bucklin (2015) separate between “non-visitor – visitor – authenticated user – converted customer”, and Ghose and Todri-Adamopoulos (2016) differentiate between “search – visit – conversion”. Expanding basic AIDA (“awareness – consideration – desire – action”) concept, Batra and Keller (2016) present an extensive framework encompassing “needs – awareness – examination – learning – liking – payment – commitment – consumption – satisfaction – loyalty – engagement – advocacy”. Both Haan, Wiesel, and Pauwels (2016) and Kireyev, Pauwels, and Gupta (2016) orient their framework toward the website structure (“homepage – product page – shopping basket – check out”; “search clicks – search conversion”). Similar to Abhishek, Fader, and Hosanagar (2012), Colicev, Kumar, and O'Connor (2018) relate their work to customers’ cognitive states by using “awareness – consideration – purchase intent – satisfaction” as a funnel framework conceptualization.

Most studies use four or more funnel stages and appreciate a comprehensive understanding of the pre-purchase stage. To evaluate factors leading customers to a purchase, researchers subdivide the first stage into different levels. Thus, funnel frameworks in attribution literature usually do not follow firmly anchored three-stage classification (pre-purchase – purchase – post-purchase) but apply a deeper subdivision (e.g. Frambach, Roest, and Krishnan 2007; Gensler, Verhoef, and Böhm 2012; Lemon and Verhoef 2016; Neslin et al. 2006). This structure ensures a sound understanding of the decision process with a particular focus on impact factors leading customers to a purchase.

Besides, researchers in attribution theory often define funnel stages as cognitive states (e.g. Abhishek, Fader, and Hosanagar 2012; Batra and Keller 2016; Colicev, Kumar, and O'Connor 2018; Li and Kannan 2014). This structure grounds in a much-noticed research paper published by Shocker et al. (1991) who recommend customers’ state of mind to model the evolutionary decision process. Customers start at a broad awareness level and specify their consideration along the purchase process.

The concepts constituted in this section and summarized in table 1 serve as the foundation for my own framework in section 2.3. I adapt the typical three-stage

classification (pre-purchase – purchase – post-purchase) based on the advantages of conceptualizations applied in attribution research.

Table 1: Funnel frameworks

Author	Year	Context	Funnel framework
Wiesel, Pauwels, and Arts	2011	office equipment	visits leads quote requests orders
Abhishek, Fader, and Hosanagar	2012	automotive	disengaged active engaged conversion
Li and Kannan	2014	hospitality	consideration visit purchase
Ghose and Todri	2015	experience good	search visit conversion
Batra and Keller	2016	-	needs awareness examination learning liking payment commitment consumption satisfaction loyalty engagement advocation
Haan, Wiesel, and Pauwels	2016	i.a. electronics, fashion	homepage product page shopping basket check out
Kireyev, Pauwels, and Gupta	2016	financial service	search clicks search conversion
Colicev, Kumar, and O'Connor	2018	i.a. automotive, fashion	awareness consideration purchase Intent satisfaction

Source: author's own illustration.

2.2.2 Channel attribution concept

2.2.2.1 Channel overview

Although customers already encounter multiple channels along the purchase funnel, the technological development continually creates new contact points (Frambach,

Roest, and Krishnan 2007; Gensler, Verhoef, and Böhm 2012). In this day and age, marketing managers spend around two thirds of their advertising budget on digital media (McIntyre and Virzi 2018, p. 9). As all channels impact customers in a different way, an accurate evaluation of channel effectiveness poses a crucial challenge (Braun and Moe 2013, p. 755; Danaher and Dagger 2013, p. 517; Guo 2012). Corporations typically use paid, owned and earned channels to interact with customers (Batra and Keller 2016, p. 129; Stephen and Galak 2012, p. 624). Next to paid channels like display advertising, CPC, e-mails, and affiliate marketing, owned channels comprise organic search results and direct website visits. Additionally, customers can rely on referrals by other customers, which is denoted as earned channels. Alongside these three categories, the influence of social media marketing is sharply increasing (Colicev, Kumar, and O'Connor 2018, p. 100; Felix, Rauschnabel, and Hinsch 2017, p. 118; Kumar et al. 2016, p. 1). By offering the opportunity to place paid ads, owned marketing campaigns and an environment for customer-to-customer interactions, social media constitutes a platform for all three channel types.

Paid channels

Display advertising encompasses different categories like prospecting, retargeting or video ads (Ghose and Todri-Adamopoulos 2016, p. 889). More than one of these advertising types usually impact customers' decision process and influence their likelihood to visit a website (Hoban and Bucklin 2015, pp. 375–376). Likewise, it positively impacts final conversion rates and repurchase probabilities (Ghose and Todri-Adamopoulos 2016, pp. 900–901; Kireyev, Pauwels, and Gupta 2016, p. 475; Manchanda et al. 2006, p. 98). The distinct effects differ due to customers' position on the purchase funnel (Ghose and Todri-Adamopoulos 2016, p. 891). In general, a display ad on early funnel stages exerts stronger influence.

Many corporations spend a large proportion of the marketing budget on search engine advertising (Li et al. 2016, p. 831). The success of CPC mainly depends on the word choice and its position on the search result page (Ghose and Yang 2009, pp. 1605–06; Rutz, Trusov, and Bucklin 2011, p. 663). Broader keywords stimulate significantly more return visits than narrow keywords. Additionally, longer keywords lead to lower click-through rates than shorter.

E-mail marketing does not belong to the latest achievements in marketing. But it is still an often-used tool by a lot of corporations (Zhang, Kumar, and Cosguner 2017, pp. 851–853). As a low-cost communication method, it enables marketing managers to contact customers without much effort.

Another method of paid search advertising is affiliate marketing. It is similar to display advertising on search engines but shows a key difference. Affiliate marketing does not pledge the vendor to pay in advance (Edelman and Brandi 2015, p. 2). Instead, the affiliate only receives money if the forwarded customers engage in a product purchase.

Owned channels

Many visitors use paid search advertising in the beginning but change to direct website accesses over time (Rutz, Trusov, and Bucklin 2011, p. 646). Therefore, customers directly visiting a website often show a long-term relation with the company.

Another way in which customers access a website are organic search results (Yang and Ghose 2010). They appear below paid search ads and are not sponsored by corporations. Search engine algorithms determine their page ranking which corporations try to improve by search engine optimization.

Earned channels

Referral programs usually depict deliberately established firm initiatives that reward existing customers if they recruit new ones (Guo 2012, p. 373). Customers highly appreciate recommendations by other customers as they assume them to be particularly trustworthy (Chevalier and Mayzlin 2006; Trusov, Bucklin, and Pauwels 2009). Furthermore, referral programs foster customer engagement which strengthens the relationship to the corporations. Thus, they do not only affect customers receiving a recommendation but also customers actively participating by giving recommendations.

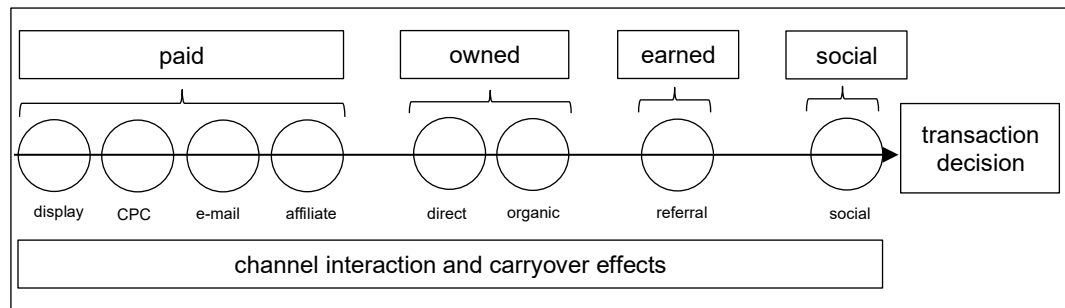
Social media

Driven by the digital disruption of business processes, more and more corporations establish social media as their main communication medium (Chang, Yu, and Lu 2015, p. 1; Felix, Rauschnabel, and Hinsch 2017, p. 118; Moorman, Christine, pp. 44–57). It offers customers (by user generated content) and marketing departments

(by firm generated content) new opportunities to interact with each other (Colicev, Kumar, and O'Connor 2018; Kumar et al. 2016). While user generated content impacts customers' awareness and satisfaction, firm generated content is more effective regarding customers' consideration and purchase intention.

In conclusion, customers can access a website through multiple channels (see figure 1). During the entire journey they typically encounter several touchpoints. Channel impacts do not only depend on the funnel position but also differ due to interaction and carryover effects. Preceding channel contacts can influence the effect of the current touchpoint (interaction effect) or unfold their effect with a time lag (carryover effect).

Figure 1: Channel overview



Arbitrary channel sequence.

Source: author's own illustration.

2.2.2.2 Channel attribution in practice

Corporations often base channel attribution methods on simple heuristics not accounting for interaction and carryover effects (Anderl et al. 2016, p. 457; Berman 2018; Kannan, Reinartz, and Verhoef 2016, p. 450; Li and Kannan 2014, p. 41). The “last touch method” (LT), the “first touch method” (FT), the “uniform method” (U), as well as the “exponential method” (EX) are relevant approaches in practice. Due to their inherent simplicity, these methods can lead to a mediocre allocation of the advertising budget.

The LT, which completely assigns the final conversion to the last touchpoint customers get in touch with before buying a product, is the most widespread approach in practice (Berman 2018, p. 771; Danaher and van Heerde 2018, pp. 667–669). As customers typically encounter more than one channel before a purchase, the LT disregards other touchpoints although they could be relevant in explaining the conversion behavior. Additionally, it does not incorporate interaction or carryover effects between different advertising sources. Therefore, it does not capture dynamics

between channels. Besides, the LT does not consider the influence of broad keywords as an entry to the purchase funnel. Customers usually start their path to purchase with broad keywords and specialize their search behavior over time (Ghose and Yang 2009, p. 1613; Li et al. 2016, p. 836). So, narrow keywords are more closely linked to a conversion than broad keywords. Furthermore, it under-estimates the effects of referral, e-mail and display channels while overvaluing the impact of CPC (Kannan and Li 2017, p. 36; Li and Kannan 2014, p. 42; Xu, Duan, and Whinston 2014, p. 1394).

The FT is very similar to the LT, just focusing on the other end of the purchase funnel. It strengthens the importance of the initial touchpoint by allocating total credits to the first channel contact (Li et al. 2016, p. 833). As the FT also focuses on a single touchpoint, it neither captures interaction nor carryover effects. Instead of appreciating specific keywords, the FT supports generic ones since customers typically start browsing with unspecified search terms.

Since both the FT and the LT only consider a single channel, corporations established alternative approaches like the U and the EX (Li and Kannan 2014, p. 41). Although they neither incorporate specific interactions nor carryover effects, both methods deal with multiple touchpoints. The U spreads the impact leading to a conversion evenly across all channels. Therefore, it assumes that the first channel impacts the final purchase decision equally strong as the last channel. Even though the EX also includes all touchpoint effects, it values each contact point exponentially higher the closer it gets to the final conversion decision.

Recently the Shapley Value approach, which has already been proposed by Lloyd S. Shapley in a game theoretic context, is getting popular in business practice since Google established the methodology as its data-driven attribution solution (Berman 2018; Google LLC 2019b, p. 780, Shapley 1953, pp. 307–318). It ascribes purchase behavior to preceding channel contacts as the average marginal impact of all conceivable channel orders (Li and Kannan 2014, p. 51).

Summing up, practitioners use different concepts to measure channel attribution. Although most of them typically rely on heuristic approaches as the LT or FT, methodologies as the Shapley Value offer opportunities to more accurately assess channel effectiveness.

2.2.2.3 Channel attribution in research

During the last years many researchers directed their attention towards channel attribution theory (Kannan, Reinartz, and Verhoef 2016, p. 449). The Marketing Science Institute (MSI) even emphasized the topic's relevance by ranking attribution as a number one research priority for 2016 – 2018 (Marketing Science Institute 2016, p. 5). Attribution modeling provides an opportunity to gain detailed knowledge on the effectiveness of different channels (Kannan, Reinartz, and Verhoef 2016, p. 449).

Despite a research focus on channel attribution, the existing literature differs greatly in (1) the incorporation of unstructured data, (2) the application of a purchase funnel framework, (3) the usage of various statistical methods, and (4) the focus on one or more channels (see table 2).

The latest version of MSI research priorities 2018 – 2020 stresses the importance of integrating UD in research (Marketing Science Institute 2018, p. 14). This paper mainly elaborates on the potential of UD in channel attribution research. Therefore, the following review chronologically presents the existing attribution literature subdivided into two parts according to the incorporation of UD.

Attribution research not using unstructured data

As the marketing budget allocation is a topic of high interest, numerous studies focus on channel attribution theory. However, many researchers do not use the opportunities of advanced data analytics (integration of UD) to develop new channel attribution models.

Manchanda et al. (2006) analyze the effect of banner advertising on purchase probabilities using a hierarchical Bayesian model. Shao and Li (2011) amplify this study and present a model dealing with the impact of six digital channels. They propose a bagged logistic regression as a classification model and subsequently use simple probabilistic models to attribute channels' effectiveness. Wiesel, Pauwels, and Arts (2011) combine attribution research with a purchase funnel framework and exert a vector autoregression model to study the effect of online advertising on offline sales. Besides, they explicitly separate between firm-initiated and customer-initiated contact. Abhishek, Fader, and Hosanagar (2012) also align their analysis with a purchase funnel but characterize single steps as latent customer states. They apply a

hidden Markov model to analyze how display ads and CPC impact customers along the funnel. Dalessandro et al. (2012) compare customers' conversion behavior with and without touchpoint contacts. They execute different logistic regressions types and evaluate channel contacts based on Shapley Values. Danaher and Dagger (2013) present a comprehensive study assessing the effects of ten channels consisting of both online and offline touchpoints (Type II Tobit model). Li and Kannan (2014) conduct a probabilistic model accounting for carryover and interaction effects to study the impact of different online channels along three purchase funnel stages. Similar to Abhishek, Fader, and Hosanagar (2012), Xu, Duan, and Whinston (2014) make use of a Bayesian model to explore the effects of display and search advertising. Zhang, Wei, and Ren (2014) rely on Hazard models in order to assess channels' impact on customers' clicking behavior. Hoban and Bucklin (2015) focus on display advertising and align a Bayesian model with a purchase funnel framework. Ghose and Todri-Adamopoulos (2016) run a vector autoregression model to evaluate display ads while explicitly differentiating between active and passive search as dependent variables. Using Markov graphs, Anderl et al. (2016) map the customer journey to construe the impact of seven different online channels. As already several researchers before, Barajas et al. (2016) deal with the impact of display advertising (potential causal outcome model). Ji, Wang, and Zhang (2016) add social media and paid search to the analysis. They execute a probabilistic model and incorporate a time decay to account for the long-term effects. Kireyev, Pauwels, and Gupta (2016) construct a vector autoregression model to examine attribution dynamics and spillover effects between display and search advertising. In their research outlook they point out the benefit of studies analyzing channel effects on different stages of the purchase funnel. Like Dalessandro et al. (2012), Berman (2018) base his analysis on Shapley Values carving out the disadvantages of the LT. Ren et al. (2018) propose a quite novel approach based on a dual attention recurrent neural network to evaluate multi-touch attribution.

Research papers using unstructured data

In other marketing disciplines the combination of SD and UD is already very popular (e.g. Balducci and Marinova 2018; Dhar and Chang 2009; Erevelles, Fukawa, and Swayne 2016; Liu, Singh, and Srinivasan 2016; Nam, Joshi, and Kannan 2017; Nam and Kannan 2014; Pang, Lee, and Vaithyanathan 2002). In attribution research

however, the usage of both data types is still limited. Although not many, some researchers incorporate these developments – above all by Google Ad keywords and text mining methodologies.

Ghose and Yang (2009) examine the impact of CPC with a Bayesian model. Next to search terms' ranking they include content characteristics for a keyword-specific consideration. In a follow-up study they expand their first analysis by incorporating organic search terms (Yang and Ghose 2010). Equivalent to their previous study, they respect contextual differences and base their findings on a Bayesian approach. Besides dissecting search terms, Goldfarb and Tucker (2011) analyze the impact of matching display ads to website content (Bayesian model). Similar to Ghose and Yang (2009), Rutz, Trusov, and Bucklin (2011) elaborate on a keyword-specific analysis. In particular, they focus on the indirect effect of paid search terms on future direct website visits (Bayesian model). Haan, Wiesel, and Pauwels (2016) present an extensive study comprising seven online and two offline channels. They use a vector autoregression model to measure the long-term effectiveness and control for contextual differences. Kumar et al. (2016) scrutinize the influence of firm generated content by differentiating between valence, receptivity and customer susceptibility (difference-in-difference regression). Li et al. (2016) enlarge the paid search research by studying the effect of general and specific keywords along a purchase funnel (three-stage-least-square regression). Colicev, Kumar, and O'Connor (2018) elaborate on social media advertising and add user generated content to their analysis. Applying a vector autoregression model, they examine how neutral valence, positive valence and vividness of firm generated content as well as valence and volume of user generated content affect customers on four funnel stages. Sahni, Wheeler, and Chintagunta (2018) identify the research gap in e-mail marketing and apply a Hidden Markov model to analyze different forms of personalization.

From all studies presented, Haan, Wiesel, and Pauwels (2016) provide the most similar framework to this conceptualization as they incorporate UD, apply a purchase funnel framework and analyze the effect of different channels. However, their study mainly carves out the difference between the LT and more sophisticated analyses. Instead, I emphasize the benefit of incorporating UD to improve attribution accuracy and evaluate CPC subgroups.

The full potential of incorporating UD in channel attribution research is far from being tapped. Many researchers limit their application to text data and do not benefit from the great volume of image data. The success of social media platforms as Instagram underlines the importance of visual data (Statista 2019). This study expands the usage of UD by categorizing product images displayed on an e-commerce platform and subsequently incorporates the extracted information in channel attribution models. Moreover, most researchers focus on the analysis of paid search term categories but do not consider differences in ad descriptions (e.g. Ghose and Yang 2009; Rutz, Trusov, and Bucklin 2011). Thus, I shift the focus on analyzing the effect of different Google Ad descriptions.

Developing new methodologies dealing with UD is a state-of-the-art research topic. In addition, the integration of machine learning algorithms in marketing research is still limited (Marketing Science Institute 2018, p. 14). This paper highlights the opportunities of UD in attribution research, presents a theory-based conceptual framework, and applies latest machine learning algorithms. To the best of my knowledge, no researcher has provided a similar comprehensive study in channel attribution literature before.

Table 2: Literature overview

Author	Year	Context	Method	Unstructured data	Purchase funnel	Different channels
<i>Unstructured data not used</i>						
Manchanda et al.	2006	Healthcare and beauty	Bayesian Model	×	×	✓
Shao and Li	2011	Consumer software	Bagged Logistic Regression	×	×	✓
Wiesel, Pauwels, and Arts	2011	Office equipment	Vector Autoregression	×	✓	✓
Abhishek, Fader, and Hosanager	2012	Automotive	Hidden Markov Model	×	✓	✓
Dalessandro et al.	2012	FMCG, Telecommunication	Logistic Regression	×	×	✓
Danaher and Dagger	2013	Electronics, Fashion	Type II Tobit Model	×	×	✓
Li and Kannan	2014	Hospitality	Probabilistic Model	×	✓	✓
Xu, Duan, and Whinston	2014	Electronics	Bayesian Model	×	×	✓
Zhang, Wie, and Ren	2014	-	Hazard Model	×	×	✓
Ghose and Todri	2015	Experience good	Vector Autoregression	×	✓	×
Hoban and Bucklin	2015	Financial service	Bayesian Model	×	✓	×
Anderl et al.	2016	Online travel, Fashion	Markov Graphs	×	×	✓
Barajas et al.	2016	Telecommunication, Transport	Potential Outcome Model	×	×	×
Ji, Wang, and Zhang	2016	-	Probabilistic Model	×	×	✓

Author	Year	Context	Method	Unstructured data	Purchase funnel	Different channels
<i>Unstructured data not used</i>						
Kireyev, Pauwels, and Gupta	2016	Financial service	Vector Autoregression	✗	✓	✓
Berman	2018	-	Probabilistic Model	✗	✗	✓
Ren et al.	2018	-	Neural Network	✗	✗	✓
<i>Unstructured data used</i>						
Ghose and Yang	2009	-	Bayesian Model	✓	✗	✗
Yang and Ghose	2010	-	Bayesian Model	✓	✗	✓
Goldfarb and Tucker	2011	i.a. Automotive, Fashion	Bayesian Model	✓	✗	✗
Rutz, Trusov, and Bucklin	2011	Automotive	Bayesian Model	✓	✗	✗
Haan, Wiesel, and Pauwels	2016	i.a. Electronics, Fashion	Vector Autoregression	✓	✓	✓
Kumar et al.	2016	Beverage	Difference-in-Difference	✓	✗	✓
Li et al.	2016	Jewelry	Three-stage-least-square	✓	✓	✗
Colicev, Kumar, and O'Connor	2018	i.a. Automotive, Fashion	Vector Autoregression	✓	✓	✗
Sahni, Wheeler, and Chintagunta	2018	Education	Hidden Markov Model	✓	✗	✗
<i>This study</i>	<i>2019</i>	<i>Fashion</i>	<i>Random Forest</i>	✓	✓	✓

Source: author's own illustration.

2.3 Methodological framework

2.3.1 Framework overview

Funnel frameworks are beneficial to ensure a well-structured analysis as they organize the decision process into separate steps (Batra and Keller 2016, p. 138). This study includes single funnel stages and provides a better understanding of channel attribution compared to frameworks entirely focusing on the transaction stage. Even if channel contacts do not cause a purchase, they can be useful to impact customers' overall attitude.

In section 2.2.1 I describe a general funnel framework and describe all attribution studies applying a purchase funnel conceptualization (see table 1). Based on the advantages of funnel frameworks in attribution theory, I adapt the typical three-stage classification (pre-purchase – purchase – post-purchase) by especially valuing the pre-purchase stage. Thus, I subdivide the first stage to precisely evaluate channel impacts leading customers to a purchase. Another advantage of funnel conceptualizations in attribution theory is the usage of customers' state of mind. Understanding their cognitive states helps to get a better impression of the underlying decision-making process. Therefore, I adopt the usage of different customer states and specifically focus on “awareness” (pre-purchase), “consideration” (pre-purchase), and “purchase”.

In contrast to most frameworks in attribution research, I do not expand the funnel structure subdivision as I want to maintain distinctive thresholds between funnel stages. A separation into three levels both ensures unambiguous stage definitions and accurately models customers' decision process. As I do not have access to data about the post-purchase stage, I do not consider it in the analysis.

Similar to Haan, Wiesel, and Pauwels (2016), I operationalize customers' awareness as clicks on product overview pages, purchase consideration as clicks on product detail pages, and the purchase itself as the transaction revenue. In doing so, I closely align the analysis with a typical website structure. Customers usually start their buying process by browsing across multiple pages (awareness). If they are interested in an item, they visit the product detail page (consideration) and finally decide whether to buy a product (purchase).

The analysis stepwise explains customers' (1) awareness, (2) consideration, and (3) purchase. Based on this structure, I evaluate the effectiveness of eight digital channels along an e-commerce website funnel (see figure 2).

To consider customers' way through the funnel, I consecutively expand the framework by including awareness to model consideration, and subsequently awareness and consideration to model purchase (Bucklin and Sismeiro 2003, p. 258; Inman, Winer, and Ferraro 2009, p. 27; Kollat and Willett 1967, p. 24; Mallapragada, Chandukala, and Liu 2016, p. 31; Schwarz 2004).

Next to the eight digital channels and customer states (awareness and consideration), situational and user characteristics exert influence on customers' individual behavior (Batra and Keller 2016, p. 131; Blanco, Sarasa, and Sanclemente 2010, p. 668; Danaher and van Heerde 2018, p. 680; Mallapragada, Chandukala, and Liu 2016, p. 32). Therefore, I include these aspects as control variables.

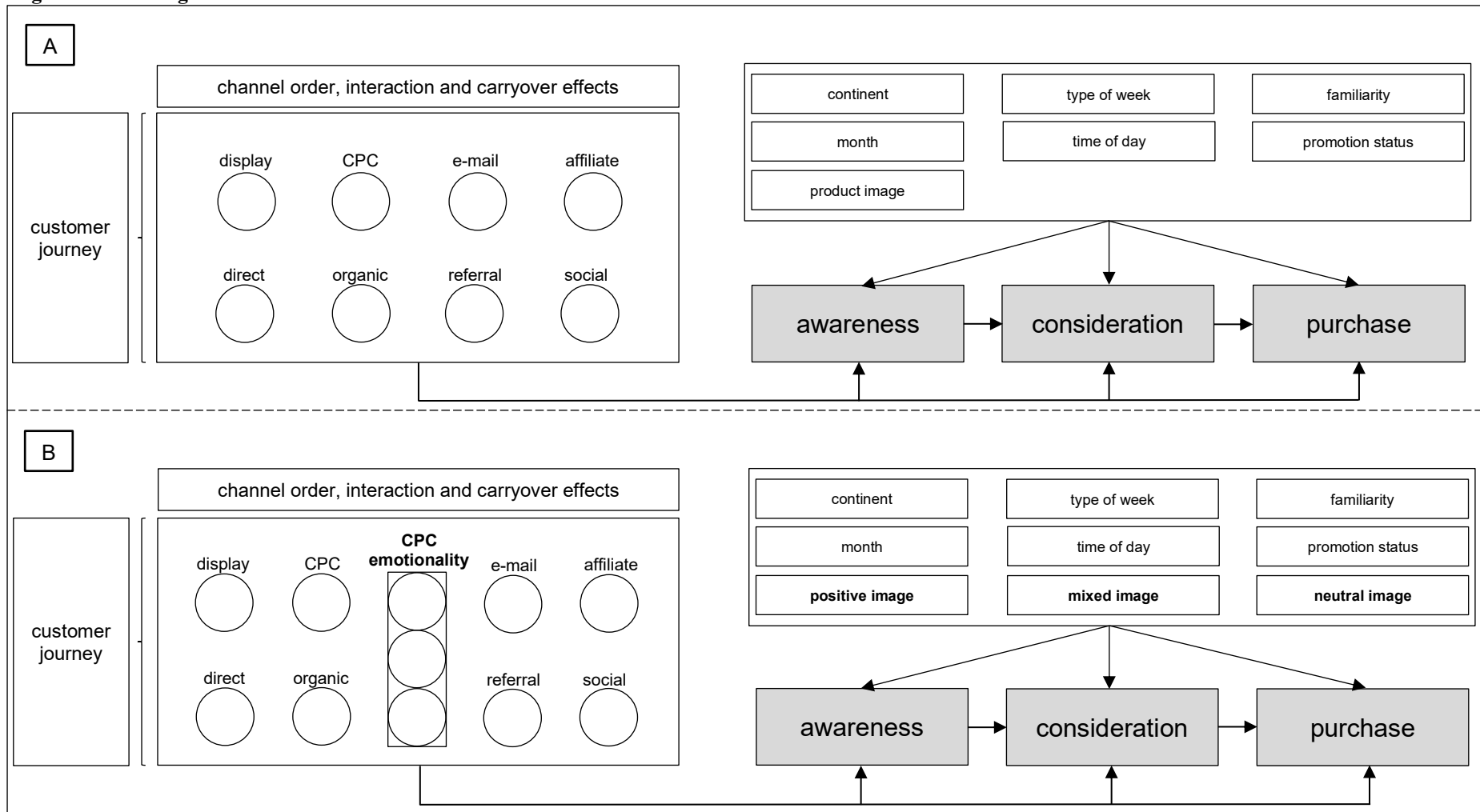
In a separate analysis I expand the framework by the incorporation of UD (see figure 2.B). Website characteristics like corporations' communication style impact customers' clicking behavior. Therefore, I include this information in form of facial expressions in product images and text features in Google Ad descriptions (Blanco, Sarasa, and Sanclemente 2010, p. 668; Childers et al. 2001, p. 527; Cyr et al. 2009, pp. 539–540; Rosen and Purinton 2004, p. 784). As the additional data helps to predict customer states, the improved estimates subsequently serve as a foundation to accurately evaluate channel attribution. Therefore, I hypothesize:

H1: Incorporating unstructured data increases channel attribution accuracy along an e-commerce website funnel.

A categorization of Google Ad descriptions reveals insights beyond numeric figures (Bradlow et al. 2017, p. 88; Erevelles, Fukawa, and Swayne 2016, p. 898). A CPC subdivision is useful as customers react differently to various communication styles (Ghose and Yang 2009; Li et al. 2016). Therefore, I hypothesize:

H2: Incorporating unstructured data reveals the effectiveness of CPC sub groups along an e-commerce website funnel.

Figure 2: Modeling framework



Legend: A – unstructured data not included; B – unstructured data included.
Source: author's own illustration.

2.3.2 Framework description

Purchase funnel stages

Usually customers start their shopping process by browsing the website. They get an overview of different offers and encounter several product overview pages. I define this stage as the awareness stage. If customers find an interesting item, they move on to the next decision level. Product detail pages provide specific information and the opportunity to inspect products from different perspectives. As customers consider the product in detail, I specify this funnel step as the consideration stage. Finally, customers ponder products' utility and decide whether to buy something at the purchase stage.

Obviously, the shopping behavior is no straightforward process. Instead, customers move back and forth between different purchase funnel stages. After considering individual products, they return to an overview page and get aware of another alternative. This process continues until customers finally decide whether to buy something.

In total, customer journeys comprise all product overview (awareness stage) and detail page clicks (consideration stage) before a final purchase decision. Customers do not follow a pre-defined buying process encompassing one interaction at the awareness stage, one at the consideration stage and a final decision at the purchase stage. Instead, they dynamically switch between different levels of the funnel.

Channels

Although offline channels still exist in the advertising environment, this study focuses on the effectiveness of eight online channels: display advertising, CPC, e-mail marketing, affiliate marketing, direct search, organic search, customer referral, and social media marketing.

I model the purchase process as a customer journey including all channel contacts until a final purchase decision. As customers usually switch between different channels, every journey consists of a unique channel order and length. Therefore, I capture order, interaction, and carryover effects between channels.

Many corporations spend a large portion of the marketing budget on CPC (Li et al. 2016, p. 831). Hence, an in-depth analysis of different conceptualizations is valuable. A content categorization of all Google Ad descriptions classifies the search channel into three groups, whereby each represents a particular emotionality type (strong positive, positive, neutral). In doing so, attribution models comprising both SD and UD, do not only indicate if customers interacted with CPC but also states the exact sub-category. Therefore, I scrutinize the effectiveness of three different CPC categories.

Situational and customer characteristics

Every channel contact takes place in a specific situation and environment. To account for this heterogeneity, the attribution model needs to control for situational information and unique customer characteristics (Batra and Keller 2016, p. 131; Blanco, Sarasa, and Sanclemente 2010, p. 668; Blattberg, Briesch, and Fox 1995, p. 128; Danaher and van Heerde 2018, p. 680; Mallapragada, Chandukala, and Liu 2016, p. 32). Therefore, the continent (Europe – Asia – America – Africa – Oceania), the promotion status (membership day – no membership day), the month (i.e. March – April), the type of week (no weekend – weekend), the time of day (morning – afternoon – evening – night) and customers' website familiarity (not familiar – familiar) provide information to explain customer states along the purchase funnel. Whereas the baseline attribution model (without UD) only includes the overall interaction frequency with product images, the extended model (combining SD and UD) incorporates the number of contacts with different image categories (positive, mixed, neutral).

3 Data

3.1 Data source

The e-commerce platform of an international Dutch corporation, which is active in over 20 countries, serves as the foundation for the analysis in this study. The corporation operates in its sector as a market leader in north-western Europe. It follows an omnichannel strategy while especially valuing the importance of customer interactions with digital channels. An annual e-commerce growth rate of about 40% reflects the corporation's focus on further amplifying a digital business strategy.

3.2 Data description

The provided dataset covers the time period from March 11, 2019 until April 7, 2019 and comprises 1,376,906 website sessions. 30 minutes of inactivity, midnight or an access through a new advertising campaign end an old session and start a new one. 8,046 sessions show neither an interaction on a product overview page (awareness stage) nor on a product detail page (consideration stage). Thus, I exclude them from further analysis which reduces the dataset extent to 1,368,860 observations.

The channels analyzed in this study account for 1,335,781 e-commerce website accesses (97.58% of all sessions). In total, 33 variables provide information about eight digital channels (display advertising, CPC including ad descriptions, e-mail marketing, affiliate marketing, direct search, organic search, customer referral, and social media marketing), three funnel stages (awareness stage, consideration stage, purchase stage), the corresponding transaction revenue, as well as seven situational and customer characteristics (customer identifier (ID), date, continent, promotion status, month, type of week, time of day, website familiarity, product image links) (see table 3).

Table 3: Variable operationalization

Variable	Dataset term	Scale	Description
channels			
display advertising	display	0 / 1	e-commerce entered via display?
CPC	cpc	0 / 1	e-commerce entered via cpc?
Google Ad description	ad_description	text	Google Ad descriptions
e-mail marketing	email	0 / 1	e-commerce entered via email?
affiliate marketing	affiliate	0 / 1	e-commerce entered via affiliate?
direct search	direct	0 / 1	directly entered e-commerce?
organic search	organic	0 / 1	e-commerce entered via organic?
customer referral	referral	0 / 1	e-commerce entered via referral
social media marketing	social	0 / 1	e-commerce entered via social?
funnel stages			
awareness	views_product_ overview	0–200	clicks on product overview pages

Variable	Dataset term	Scale	Description
funnel stages			
consideration	views_product_detail	0–188	clicks on product detail pages
purchase	transactions	0–5	number of transactions
transaction revenue			
transaction revenue	transaction_revenue	0–33,503	transaction revenue
situational information and customer characteristics			
customer ID	user_id	integer	unique customer identification
date	date	ISO 8601	date of website session
Europe	Europe	0 / 1	e-commerce entered from Europe?
Asia	Asia	0 / 1	e-commerce entered from Asia?
America	Americas	0 / 1	e-commerce entered from America?
Africa	Africa	0 / 1	e-commerce entered from Africa?
Oceania	Oceania	0 / 1	e-commerce entered from Oceania?
promotion day	promotion_day	0 / 1	e-commerce entered on promotion day?
no promotion day	no_promotion_day	0 / 1	e-commerce not entered on promotion day?
March	March	0 / 1	e-commerce entered in March?
April	April	0 / 1	e-commerce entered in April?
no weekend	no_weekend	0 / 1	e-commerce entered between Monday and Friday?
weekend	weekend	0 / 1	e-commerce entered on Saturday or Sunday?
morning	morning	0 / 1	e-commerce entered between 06:00 am and 12:00 pm?
afternoon	afternoon	0 / 1	e-commerce entered between 12:00 pm and 06:00 pm?
evening	evening	0 / 1	e-commerce entered between 06:00 pm and 12:00 am?
night	night	0 / 1	e-commerce entered between 12:00 am and 06:00 am?
not familiar	not_familiar	0 / 1	e-commerce website entered for first time?
familiar	familiar	0 / 1	e-commerce website not entered for first time?
product image links	photo_urls	text	links to product images that customers have viewed

Source: author's own illustration.

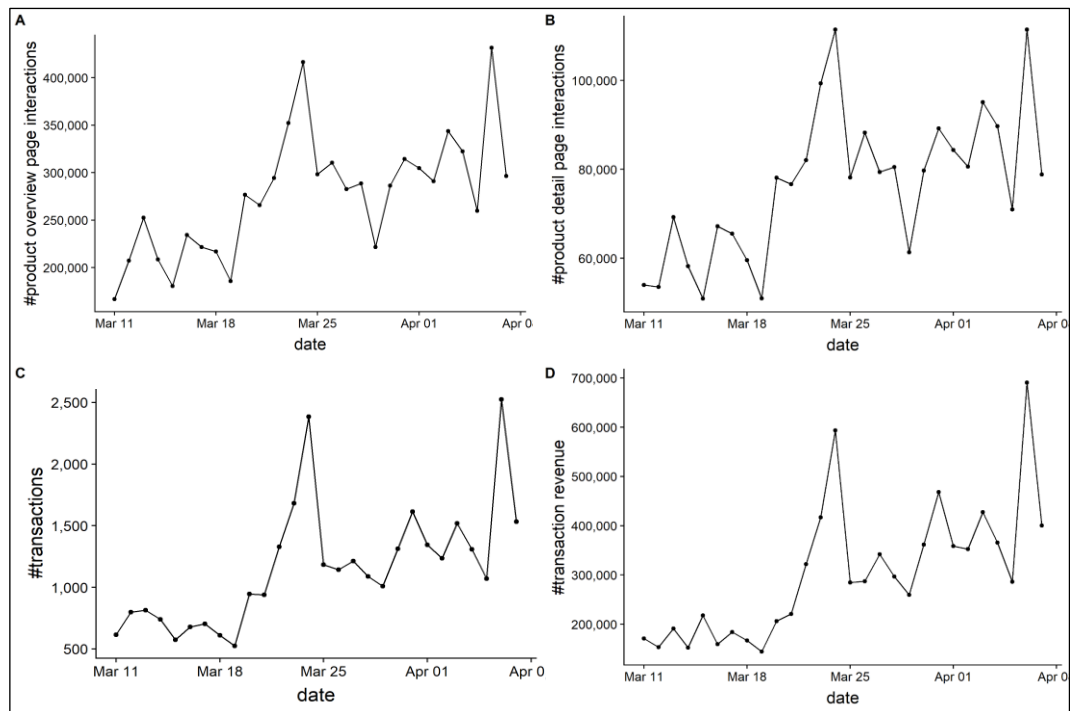
3.2.1 Purchase funnel stages

Throughout the purchase process customers pass a three-step website funnel. The dataset encompasses 7,731,143 product overview page interactions (awareness stage), 2,214,694 product detail page interactions (consideration stage), and 32,450

transactions (transaction stage) with a value of 8,480,307 Euro. All three funnel stages show a large range: product overview page interactions vary between one and 200 clicks per session, product detail page interactions between one and 188 clicks per session, and the revenue between 0.40 Euro and 33,503 Euro per session. Although the peak values constitute extremes, I do not exclude them from the analysis as they can reveal valuable insights and do not cause issues in tree-based algorithms (see appendix A.A). Especially sessions with a high transaction revenue (e.g. a store purchase) can be useful to understand channel effectiveness in depth.

All funnel stages show a similar development over time (see figure 3.A – figure 3.D). As they are strongly connected, this represents a sensible finding. The overall funnel stage development (see figure 3) matches the distribution of channel contacts (see figure 4). Both indicate an increasing trend with two peak values on March 24 and April 6, 2019, on which the corporation offered special price promotions.

Figure 3: Purchase funnel stages per day



A – product overview page interactions per day; B – product detail page interactions per day; C – transactions per day; D – transaction revenue per day.
Source: author's own illustration.

3.2.2 Channels

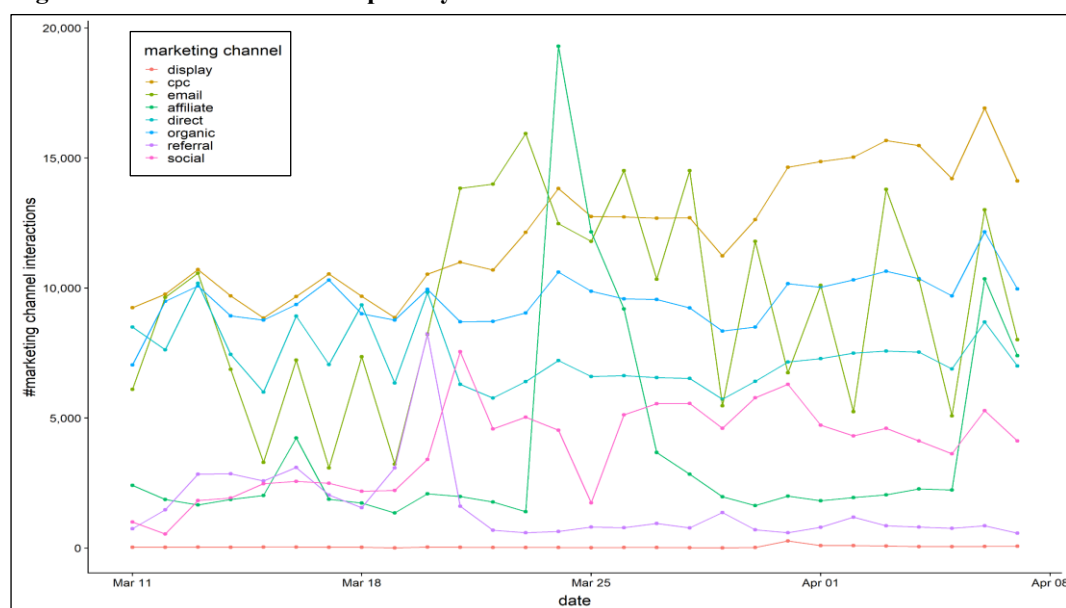
Customers entered the e-commerce store by clicking on display advertising (1,292 sessions; i.e. 0.10%), interacting with CPC (340,913 sessions; i.e. 25.50%), using an e-mail-link (262,584; i.e. 19.70%), getting in touch with affiliate marketing

(107,120 sessions; i.e. 8.02%), directly searching the store (205,020 sessions; i.e.15.30%), clicking on organic search results (267,230 sessions; i.e. 20.00%), trusting customer referrals (43,832 sessions; i.e. 3.28%) or coming from a social media website (107,790 sessions; i.e. 8.07%). Thus, by far the most sessions resulted from paid channels (53.32%), followed by owned channels (35.30%), social media platforms (8.07%), and earned channels (3.28%).

In 64.54% of all cases in which customers accessed the store via CPC, the dataset provides a Google Ad description. This generally consists of one to three short sentences displaying a brand statement or promoting a special offer. During the considered time period the corporation used 145 different descriptions. The five most common account for 86.68% of all observations which include an ad description.

Figure 4 displays customers' daily channel interactions over time. It reveals relatively strong fluctuations for e-mail advertising and affiliate marketing. Both channels reached the peak at the weekend between March 22 and March 24, 2019 at which the corporation offered price promotions. Also, on April 6, 2019 discounts led to high e-commerce accesses via all channels. Whereas CPC, e-mail marketing, organic search results, and direct website visits led most customers to the website, advertisements on social networks, affiliate marketing, referral programs, and display ads caused a lower number of interactions. In total, a slight increasing channel interaction trend is observable. A constant rise in CPC underlines this development.

Figure 4: Channel interactions per day



Source: author's own illustration.

3.2.3 Situational and customer characteristics

In total, 822,167 unique customers visited the website from March 11 until April 7, 2019. As the considered dataset encompasses 1,335,781 website sessions, a single customer engaged in 1.6247 sessions on average.

The great majority of website sessions resulted from countries within Europe (98.62%), followed by North and South America (1%), Asia (0.23%), Africa (0.08%), and Oceania (0.02%). Therefore, almost all customers accessed the website from a European country (mainly Netherlands). As 0.04% of all sessions do not indicate a country, for these observations no continent information is available (see figure 5.A).

The dataset covers a time period of 28 days, of which 21 days were in March and the remaining seven days in April. The ratio of session frequency approximately reflects this dispersion as 72.84% of sessions took place in March and 27.16% in April. Thus, the first week of April showed a slight disproportionately high fraction (see figure 5.B).

A similar imbalance appears in the session distribution at weekends versus weekdays. Whereas weekdays comprised 71.43% of all days, they accounted for only 68.89% of all sessions. This indicates a higher customer activity Saturdays and Sundays. Generally, customers have more time at weekends and enjoy using it for apparel shopping (see figure 5.C).

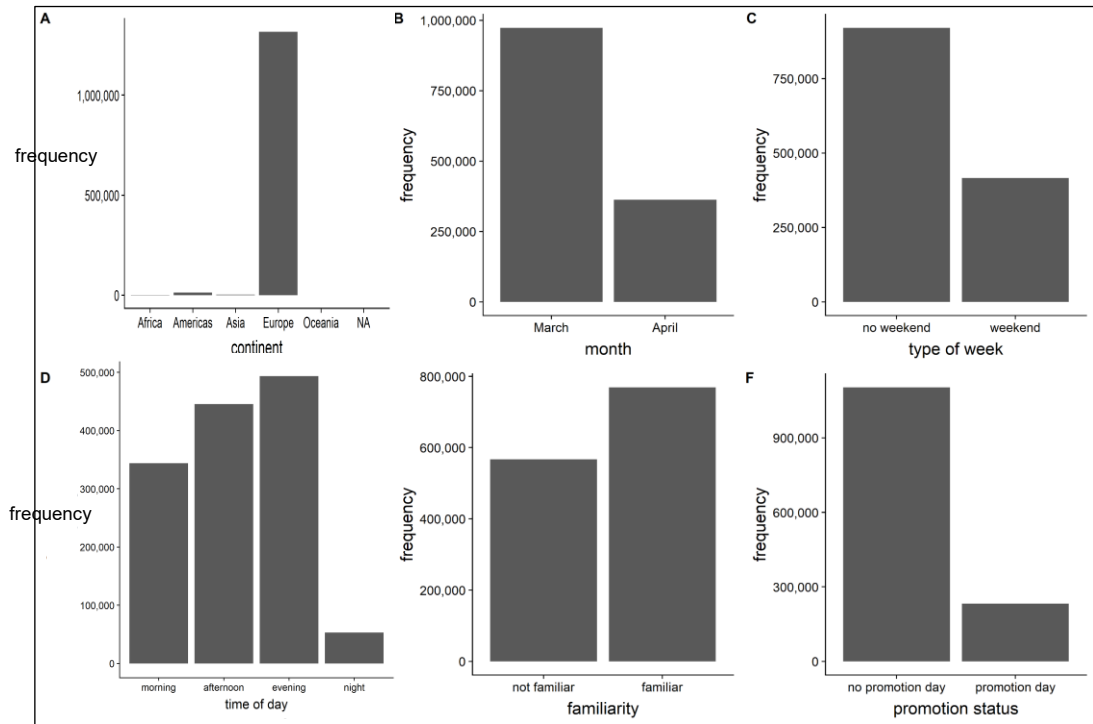
The number of session interactions rose from 25.75% in the morning, to 33.35% in the afternoon and finally 36.94% in the evening. Thus, customer interest was especially intensive between 06:00 pm and 12:00 am. Afterwards, the website engagement dropped sharply to 3.96% between 12:00 am and 06:00 am. Similar to the weekend phenomenon, customers like to use leisure time after work for hedonic activities as spending money on fashion (see figure 5.D).

42.43% of all sessions resulted from customers entering the website for the first time. Thus, almost every second customer represented a new acquisition. This high attraction rate underlines the interest of new customers. It reflects the corporation's focus on a strong digital presence and contributes to an annual e-commerce growth rate of about 40% (see figure 5.E).

Between March 11 and April 7, 2019, the corporation offered two promotion periods over a length of four days in total. During these days, the corporation lowered prices and engaged in particularly strong advertising. 17.43% of all sessions fell on these days (see figure 5.F).

If customers get aware of a product they are interested in, they visit the product detail page for an explicit consideration and the opportunity to take a closer look at products. In 46.25% of all sessions customers clicked at least on one product image. This study includes all pictures displayed in the first place on the product detail page (1,953 pictures). On average, customers interacted with two product images per session.

Figure 5: Situational and customer characteristics



A – continent distribution; B – promotion status distribution; C – month distribution; D – type of week distribution; E – time of day distribution; F – familiarity distribution.
Source: author's own illustration.

3.3 Dataset conception

Based on the data described in previous sections, I create a customer journey dataset. I first select all journeys consisting of a single session and subsequently extract all multi-session journeys. To construct the latter, I chronologically order the observations for every customer ID. Thereafter, I group all session for an individual customer on a transaction level, whereby only the first multi-session journey remains in the dataset. As the considered time period is limited to one month, I assume that

all sessions before a final purchase decision belong to the same journey. Finally, I combine all single-session with the grouped multi-session journeys into a final dataset.

Every observation represents a single customer journey including the added number of product overview page interactions, product detail page interactions, transactions, and the corresponding revenue. Further it includes the frequency of all channel contacts (see figure 6). It provides information about the continent from which customers accessed the e-commerce store, as well as the added number of sessions for each month, type of week, and time of day. Finally, it indicates customers' website familiarity, the number of sessions at price promotion days and delivers all links to product pictures on which customers clicked during their purchase journey.

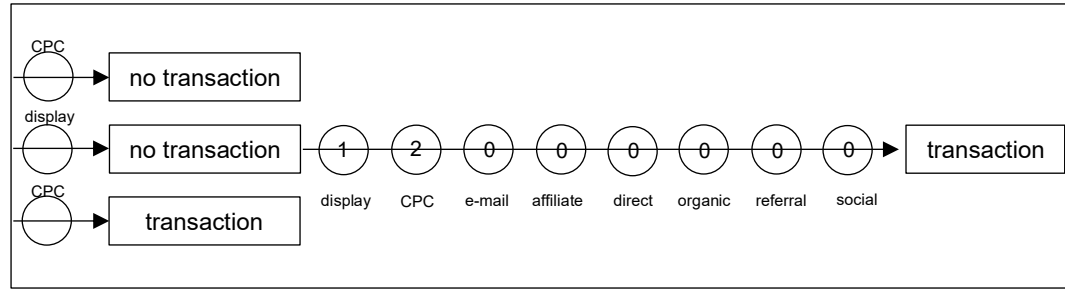
The resulting dataset comprises 812,721 customer journeys. On average a customer interacted with 0.002 display ads, 0.41 CPC, 0.31 e-mails, 0.13 affiliate ads, 0.25 direct accesses, 0.32 organic search results, 0.05 referrals, and 0.13 social ads during a journey. This led to 9.33 clicks on product overview pages (awareness stage) and 2.59 clicks on product detail pages (consideration stage) on average. In total, 3.90% of all journeys resulted in at least one transaction with a mean revenue of 263.04 Euro (median: 139.96 Euro).

The majority of all journeys took place in Europe (98.06%), followed by North and South America (1.45%), Asia (0.27%), Africa (0.07%), and Oceania (0.03%). A small portion resulted from website accesses across different continents (0.06%). For the remaining customer journeys, no continent information is available (0.06%). More than two-thirds of all journeys started and finished in March (69.29%), whereas a quarter completely occurred in April (24.39%). All other journeys comprised sessions in both months (11.32%). 63.23% of journeys ensued at weekdays and 27.22% at weekends. The remaining sessions took place at both weekdays and weekends (9.56%). The journey frequency successively increased during the day but sharply dropped at night (morning: 20.33%, afternoon: 26.77%, evening: 31.81%, night: 3.39%). 17.70% of journeys included website sessions at different times of day. The distribution of website familiarity is almost balanced (familiar: 45.58%, not familiar: 54.42%), whereat even more journeys resulted from unfamil-

iar customers. Although promotion days (four out of 28 days) typically arouse special awareness, they did not lead to a remarkable increase in customer journeys (promotion days: 14.34%, non-promotion days: 85.66%).

The final dataset contains multiple extreme values (see appendix A.B). However, all of them remain in the dataset as they do not represent irregular data. Additionally, they represent valuable information and do not cause problems in tree-based models.

Figure 6: Training dataset construction



Here illustrated by the example of channel contacts.

Source: author's own illustration.

4 Methodology

UD constantly gains increasing importance both in research and practice (Bradlow et al. 2017, p. 88; Erevelles, Fukawa, and Swayne 2016, p. 898). But there is not only a rapid development in data availability but also in methodologies evaluating this data (Liu, Singh, and Srinivasan 2016 p. 364). Machine learning techniques provide opportunities to analyze data beyond rule-based approaches as they autonomously identify optimal learning criteria. Researchers decide which information to include but do not specify learning factors in advance. Therefore, machine learning algorithms often involve a lack of interpretability as the learning process commonly resembles a black box model. However, model agnostic techniques as the Shapley Value approach provide possibilities to comprehend the underlying process.

I first expand the basic customer journey dataset by information captured in product images and Google Ad descriptions. Thereafter, I use the extended version to evaluate channel attribution in more detail.

4.1 Unstructured data analysis

This study uses computer vision and text mining methods to benefit from information captured in product images and Google Ad descriptions. The algorithms

structure both into three categories each with regard to emotionality. Thus, six variables expand the previously developed dataset and serve as additional explanatory input data in supervised machine learning methods.

4.1.1 Categorization of image data

Digitalization and particularly the power of social media platforms fuel the availability of image data. However, researchers often do not incorporate visual information. In marketing only a few studies extract image features or label visual data to expand the dataset (e.g. Dzyabura, El Kihal, and Ibragimov 2018; Joo, Wilbur, and Zhu 2016; Klostermann et al. 2018).

In this study I apply computer vision methodology on image data and evaluate the emotionality of fashion models' facial expression. First, I select all unique Uniform Resource Locators (URLs) leading to product images and bundle the 1,953 URLs in a character vector. Afterwards, I apply the Microsoft Azure Face Recognition application programming interface (API) to evaluate inherent emotionality (Microsoft; Yu and Zhang 2015). In contrast to Klostermann et al. (2018), I do not utilize the Google Cloud API as Microsoft's algorithm provides more detailed information and distinguishes better between single emotions.

The algorithm first detects faces by applying a hierarchical process consisting of three different methodologies (Yu and Zhang 2015). It starts with several cascading classifiers proposed by Chen et al. (2014). Although this method already delivers a relatively high recognition accuracy, it shows issues for detecting profile faces. Thus, a second approach based on a deep convolutional neural network (DCNN) investigates pictures for which the first algorithm did not detect faces (Zhang and Zhang 2014). It is especially successful in evaluating profile and non-frontal faces. However, it lacks regarding pictures in which the largest face is not suitable for the emotion recognition. Finally, a third algorithm that applies a mixture of several tree-based methods examines all pictures for which no face has been covered during the first both steps (Zhu and Ramanan 2012). The combination of all three algorithms ensures a high face detection veracity which is essential to evaluate the emotionality in a second step.

In order to construct a robust emotionality detector, randomly distorted images (perturbed images) expand the training dataset (Yu and Zhang 2015). The incorporation

of rotated, skewed, and rescaled images improves the algorithms ability to handle a broad collection of miens.




A DCNN analyzes the image data and evaluates eight emotion types (anger, contempt, disgust, fear, happiness, neutral, sadness, surprise) on a continuous zero-to-one scale (Yu and Zhang 2015). It includes seven hidden layers (five convolutional and two fully connected layers) and three stochastic pooling layers. Both layer types use rectified linear unit transformation to incorporate non-linear mapping. All pooling layers base on random sampling, a 3x3 filter window and a stride of two. Therefore, each pooling layer divides the dimensionality (layer length and height) in halves. The fully connected layers use dropout mechanisms to reduce overfitting. Finally, the last stage includes a “softmax” layer to standardize the output values between zero and one. A negative log-likelihood function measures the model loss and represents the target function for the optimization process. However, the image recognition does not depend on a single DCNN. Instead ensemble learning methods combine several DCNNs to strengthen the model robustness.

As the DCNN evaluates the emotionality based on facial expressions, it requires the image to display a face. For this reason, I do not consider images without a face in the analysis.

Obviously, the emotion variety is limited as corporations do not want to communicate negative feelings. Nonetheless, facial expressions differ regarding indicated happiness. Thus, I divide them into pictures that clearly show happy faces, pictures that indicate happiness and pictures with neutral facial expressions. Based on the underlying distribution and a visual inspection, I categorize all images with a happiness valuation above 0.8 as “happy”, images with a neutrality valuation above 0.8 as “neutral” and images with a valuation for both happy and neutral below or equal to 0.8 as “mixed” (see figure 7). Finally, I save all evaluated image URLs in separate character vectors which I use to create three new columns with information about how many happy, neutral and mixed images customers saw during the entire journey.

To sum up, I implement the Microsoft Azure Face Recognition API (DCNN) and evaluate all unique product images with respect to emotionality. The three additional variables about customers' interactions with different image categories expand the journey dataset.

Figure 7: Emotionality recognition

neutral emotionality		mixed emotionality		happy emotionality	
					
anger	0	anger	0	anger	0
contempt	0.123	contempt	0	contempt	0
disgust	0	disgust	0	disgust	0
fear	0	fear	0	fear	0
happiness	0.008	happiness	0.712	happiness	1
neutral	0.868	neutral	0.285	neutral	0
sadness	0.001	sadness	0	sadness	0
surprise	0	surprise	0.003	surprise	0

Source: author's own illustration based on Microsoft (2019).

4.1.2 Categorization of Google Ad data

To benefit from the vast amount of text data, researchers often apply different mining algorithms in their analyses (e.g. Archak, Ghose, and Ipeiritis 2011; Homburg, Ehm, and Artz 2015; Netzer et al. 2012; Schweidel and Moe 2014). These provide an opportunity to extract differences by splitting the content into several components. Especially in sentiment analyses text mining enjoys great popularity (Pang and Lee 2008; Pang, Lee, and Vaithyanathan 2002; Schweidel and Moe 2014).

Also, in attribution theory researchers occasionally apply text mining while mainly focusing on the effect of different sponsored search term categories (e.g. Ghose and

Yang 2009; Joo, Wilbur, and Zhu 2016; Rutz, Trusov, and Bucklin 2011). This study however elaborates on a new topic – namely, Google Ad descriptions. Based on a bag of words algorithm, I subdivide all descriptions into three categories with regard to emotionality. Thereby, I do not only consider if a customer accessed an e-commerce platform via a Google Ad but also include the ad descriptions’ content.

First, I select all 145 unique ad descriptions which the corporation used during the considered time period. As research methodology works better for English text data, I apply the Google Cloud translation API to translate sentences from Dutch to English (Google LLC 2019a). As both languages do not considerably differ in their cultural linguistic usage, a translation does not constitute any problems. Following, that I manually correct translation errors caused by apostrophes and mutated vowels. The revised dataset serves as the foundation for the emotionality analysis.

I treat every Google Ad description as a unique “bag of words” to represent its inherent polarity as a continuous number (Hu and Liu 2004; Rinker 2019). A higher indicator reflects a more positive message. Equation 1 illustrates how the algorithm assigns a polarity value (δ) to each description. Whereas the numerator captures the aggregated polarity, the denominator accounts for the description length (polarity term density).

$$\delta = \frac{x_i^T}{\sqrt{n}} \quad (1)$$

δ – Google Ad polarity; x_i^T – summed polarity of context clusters; n – number of words in Google Ad.

The algorithm scans the Google Ad for positive (negative) polarity signals and ascribes them an emotionality value based on a sentiment lexicon developed by Hu and Liu (2004). It includes more than 7,000 words marked as positive or negative. I slightly change the collection by adding “cheap” as a positive instead of negative word.

The algorithm creates context clusters consisting of four preceding and two subsequent words around every captured indicator. To calculate the overall cluster polarity, it reweighs the evaluated emotionality values based on amplifiers and de-amplifiers. These valence shifters are subtracted from each other and multiplied by an additional valence weight (default: 0.80). As an even number of negators cancel

each other out, only an uneven number of negators reverse the polarity. Finally, all individual cluster polarities sum up to the total description polarity (equation 2).

$$x_i^T = \sum \left((1 + c * (x_i^A - x_i^D)) * w * (-1)^{\sum x_i^N} \right) \quad (2)$$

x_i^T – context cluster polarity; c – valence weight; x_i^A – amplification indicator; x_i^D – de-amplification indicator ; w – polarity weight; x_i^N – negation frequency.

An amplification indicator (e.g. very) strengthens the emotional impact of a polarity word. However, a negation indicator can offset this effect. Thus, equation 3 only captures positive amplifications.

$$x_i^A = \sum \left((1 - w_{neg}) * x_i^a \right) \quad (3)$$

x_i^A – amplification indicator; w_{neg} – negation indicator ; x_i^a – amplification count.

In contrast to an amplification, a de-amplification (e.g. slightly) decreases the emotional intent. Its effect equals the number of de-amplifiers reassessed by the negation and amplification frequency (equation 4). Though, the de-amplification effect is limited to minus one (equation 5).

If the Google Ad description includes both an uneven number of negators and at least one amplifier, equation four embodies the negative amplification effect.

$$x_i^{D'} = \sum (-w_{neg} * x_i^a + x_i^d) \quad (4)$$

x_i^D – de-amplification indicator; $x_i^{D'}$ – help de-amplification indicator.

$$x_i^D = \max (x_i^{D'}, -1) \quad (5)$$

$x_i^{D'}$ – help de-amplification indicator; w_{neg} – negation indicator; x_i^a – amplification count; x_i^d – de-amplification count.

Naturally, an even number of negators cancels the negation effect. Therefore, a binary indicator (one for negation) represents the negation effect (equation 6).

$$w_{neg} = \left(\sum x_i^N \right) * mod 2 \quad (6)$$

w_{neg} – negation indicator; x_i^N – negation frequency.

The algorithm evaluates all Google Ad descriptions in the dataset. Of course, the polarity variation in Google Ad descriptions is inherently limited. Negative values are not sensible as the corporation tries to communicate a positive image.

In the following, I construct three vectors including all strong positive, positive, and neutral descriptions. Based on the underlying polarity distribution, I define a description as strong positive for a polarity value greater or equal to 0.8, as positive for a value between 0.5 and 0.8 and as neutral for a value below 0.5. Before I use these vectors to expand the customer journey dataset, I reallocate the categorization back to the Dutch descriptions. Finally, I use the retranslated vectors to loop over the data in order to add three columns indicating which types of Google Ad descriptions customers encountered during their journey.

In conclusion, I apply the Google Cloud translation API in combination with a sentiment analysis algorithm to structure unstructured Google Ad descriptions. The revealed categories provide information about corporations' communication style and expand the customer journey dataset by three columns describing the number of contacts with strong positive, positive, and neutral ad descriptions during an entire journey.

4.2 Structured data analysis

The expanded dataset includes the initial clickstream data and emotionality features extracted from website images and Google Ad descriptions. Thus, it serves as a sophisticated foundation for a detailed channel attribution analysis. Before subdividing the data into a training (20,000 customer journeys), a validation (20,000 customer journeys), and a test set (100,000 customer journeys), I randomly reshuffle the complete dataset.

Rapid progress in machine learning methodology allows to predict response variables without knowing the relation between independent and dependent variables in advance. Random Forests (RFs) – an ensemble technique of single-tree-based algorithms – belong to the most advanced forms of machine learning methods and promise accurate prediction results (Delen and Zolbanin 2018, p. 192; Putka, Beatty, and Reeder 2018, p. 695). However, the number of marketing researchers using this methodology is still relatively low (e.g. Coussement and Bock 2013; Coussement and van den Poel 2008).

In this study, I construct multiple RFs to model customer states along the purchase funnel. Subsequently, I apply the Shapley Value approach to accurately evaluate channel effects.

4.2.1 Random Forest

In 1996, Breiman paved the way for more sophisticated tree-based machine learning algorithms by proposing the “bagging” methodology as an ensemble technique to improve prediction accuracy and reduce variance. Based on multiple bootstrap samples, the average response (decision trees: majority response) of different trees depicts a final prediction.

However, bagged trees are usually very similar and suffer from multicollinearity. Thus, Breiman (2001) developed the RF approach which improves the results by decreasing collinearity and restricting overfitting issues. In contrast to bagged trees, RFs only consider a variable subset to determine the optimal split at each node (Liaw and Wiener 2002, p. 18). This procedure also limits potential multicollinearity issues between single variables.

Since the response variables (awareness: clicks on product overview pages, consideration: clicks on product detail pages, purchase: transaction revenue) are not binary but continuous, I use regression instead of decision trees. I train all algorithms on the training set, compare different hyperparameter tunings on the validation set, and finally evaluate their performance on the test set. In doing so, I try to minimize the mean absolute error (MAE) and root mean squared error (RMSE), whereby the latter specifically penalizes large miscalculations. Furthermore, I maximize the explained variance.

I start the tuning process by considering the default hyperparameter values (mtry: number of variables/3 = 12, node size: 5, max depth: 30, sample size: number of observations = 20,000) (Liaw 2019). As I do not face a strict time restriction, I do not optimize the parameters by Bayesian updating but apply a grid search to test different specifications. Although the Bayesian procedure is very time efficient, the model outcomes are usually not as good as separately testing different parameter combinations. The results of a single Bayesian optimized RF at the awareness stage confirm the inferiority compared to a stepwise grid search procedure.

Lower values for mtry (number of variables considered at each node) improve the model stability by decreasing the correlation between trees (Probst, Wright, and Boulesteix 2019, pp. 3–6). Besides, increasing the node size (minimum number of observations in a terminal node) substantially lowers the computation time without

a strong impact on models' performance. Although a higher node size typically reduces the depth of RFs, I provide the opportunity to grow larger trees by altering the max depth parameter. Finally, I decrease the sample size to further reduce multicollinearity between trees and limit overfitting issues. To test these parameter values, I apply a grid search which compares the performance of all hyperparameter combinations (mtry: 6, 8, 10, 12; node size: 10, 15; max depth: 20, 30, 40; sample size: 14,000, 16,000). As I do not face any overfitting issues and include only literature-based variables, I do not exclude single variables from the analysis.

On each purchase funnel stage, I compare the results of RFs not incorporating UD with the accuracy of RFs incorporating UD (clicks on strong positive Google Ads, clicks on positive Google Ads, clicks on neutral Google Ads, clicks on happy product images, clicks on mixed product images, clicks on neutral product images) (baseline RFs). By comparing both model specifications, I scrutinize if UD improves prediction results and subsequently increases channel attribution accuracy. As an interaction with a product image naturally implicates a click on a product detail page, I do not include image variables at the consideration stage.

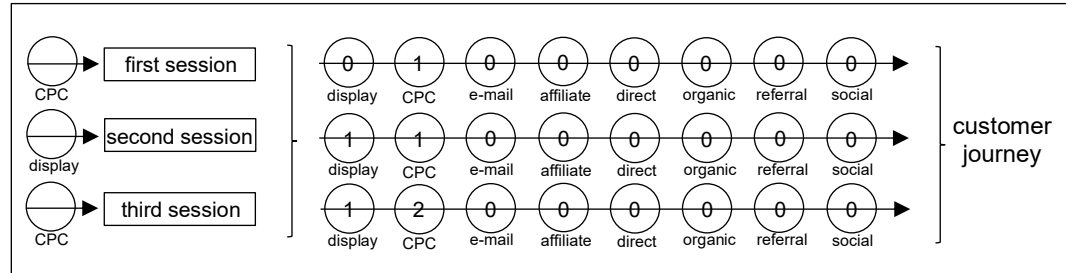
All RFs show a weak performance in predicting transaction revenue. Therefore, I create an additional model that predicts the transaction probability. Since the algorithm uses a binary transaction indicator as dependent variable, I apply a decision instead of regression tree. Whereas the former selects a node split to maximize the Gini impurity, the latter minimizes the variance (Probst, Wright, and Boulesteix 2019, p. 5). In order to identify the best prediction model, I base the decision on the prediction accuracy (proportion correctly classified), sensitivity (proportion of positive correctly classified) and specificity (portion of negative correctly classified).

After tuning the hyperparameters and evaluating the results of both model specifications on aggregated customer journeys (see figure 6), I use the best performing algorithm to predict customer states on journeys separated in their underlying sessions. In contrast to the training, validation, and test set which include both converting and non-converting customer journeys, the prediction dataset only comprises journeys with at least one transaction. Furthermore, it consecutively adds up all sessions during one journey in order to illustrate the underlying purchase process (see figure 8). Thus, also the predicted variables represent cumulative values which indicate the expected clicks on product overview pages (awareness stage), clicks on

product detail pages (consideration stage), the transaction revenue (purchase stage) and the transaction probability (purchase stage) if the customer journey would end after the considered session.

The results enable me to evaluate channels' effectiveness along the purchase funnel. Therefore, I compute Shapley Values on the predicted values in a following step.

Figure 8: Prediction dataset construction



Here illustrated by the example of channel contacts.

Source: author's own illustration.

4.2.2 Shapley Value

Different model agnostic techniques as “Local Interpretable Model-agnostic Explanations (LIME)” (Ribeiro, Singh, and Guestrin 2016), “Shapley Additive Explanations (SHAP)” (Lundberg and Su-In 2017), and the “Shapley Value” (Shapley 1953, pp. 307–318) provide opportunities to understand machine learning procedures in more detail. Especially the latter attracts high attention in channel attribution research (Berman 2018; Dalessandro et al. 2012; Li et al. 2016). Since Google recently established the Shapley Value approach as its data driven attribution methodology, many corporations value it as a channel effectiveness indicator (Google LLC 2019b). Due to the high practical relevance and its upcoming popularity in attribution research, I apply Shapley Values in order to attribute customer states to preceding channel contacts.

The Shapley Value approach originated as a cooperative game theoretic model that defines players' value as their average marginal contribution to all subgroups of the game (Shapley 1953, pp. 307–318). Dalessandro et al. (2012) transferred this concept to channel attribution theory while defining players as channels and subgroups as customer journeys. The Shapley Value concept bases on a well-grounded theoretical foundation and constitutes one of few modeling approaches fulfilling the symmetry axiom (interchangeable players receive same value), null-player axiom (player without contribution does not receive any value), and additivity axiom (if

we can separate the contribution, we can separate the values) (Shapley 1953, pp. 307–318).

Actually, the Shapley Value approach considers all possible player combinations and allocates the values dependent on players' contribution to final outcomes. However, customer journeys could theoretically be endless consisting of an extensive number of channel contacts which would lead to an infinite large number of combinations. Therefore, I limit the examination to all customer journeys which are present in the dataset. Since I consider customers' individual behavior, I account for the heterogeneity among customers. Furthermore, I predict the response variables on a session-level dataset. Thus, I model journeys at the most detailed level and allow channel effects to hinge on unique customer and situational characteristics. In doing so, I capture interaction, carryover, and order effects between channels. This is a considerable advantage over approaches like Markov models which do not account for the heterogeneity of channel effects across different customer journeys.

As the variables consecutively sum up for all sessions in a customer journey, the predicted dependent variables represent cumulative numbers. They show the expected customer behavior until the current session (e.g. clicks on product overview page until end of session i). To determine the marginal effect of a channel, I dissect the summated dependent variables. Therefore, I subtract preceding sessions from following ones to withdraw the stepwise summation (e.g. predicted clicks on product overview pages in session i – predicted clicks on product overview pages in session $i-1$). Thereafter, I add all marginal effects for each channel type (inclusive three additional subgroups for CPC). However, channels can indicate a high contribution solely caused by a high frequency. Hence, I divide the summed marginal effects by the number of customer contacts to calculate final Shapley Values. As I calculate the contributions of each sponsored Google Ad type (strong positive, positive, neutral), I evaluate channel impacts in more detail.

In conclusion, I first expand the considered dataset by six additional columns representing customers' contact with different types of product images and Google Ad descriptions. Subsequently, I use the enlarged customer journey dataset to train RFs in order to predict customer states along the purchase funnel. I apply the tuned su-

pervised machine learning algorithms on journeys presented in their underlying sessions. Finally, I calculate Shapley Values based on the predicted results to evaluate channels' effectiveness.

5 Results

5.1 Unstructured data analysis

Computer vision methodology reveals customers' interaction frequency with different product image categories. Besides, a sentiment analysis of Google Ad descriptions discloses information on channel contacts with CPC subgroups. So, image and text data expand the basic dataset by six additional variables indicating customers' contact frequency with different image categories and CPC types along the entire journey.

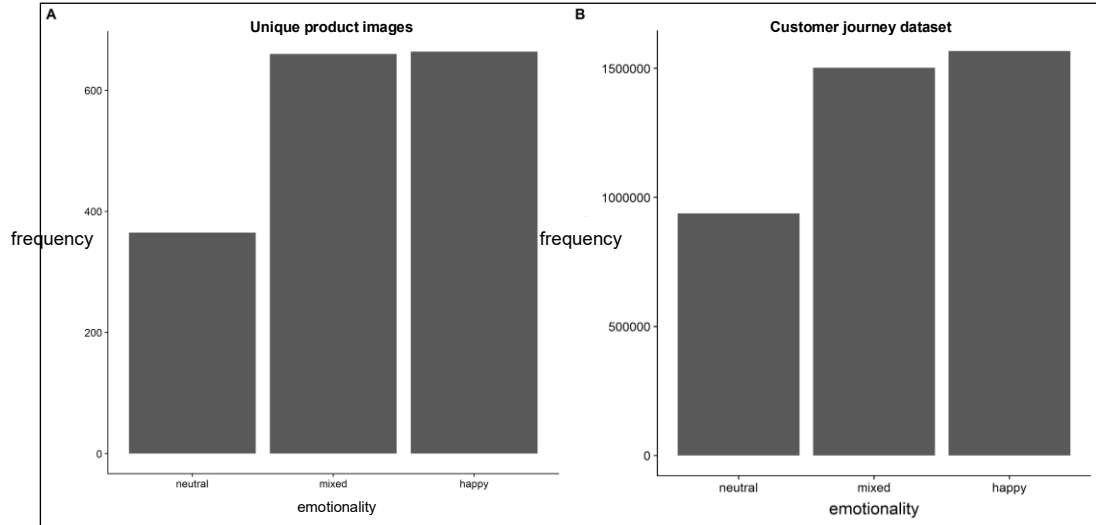
5.1.1 Categorization of image data

Customers viewed 1,953 different product images of which 264 did not show the model's face. For these, the computer vision algorithm does not evaluate emotionality leading to 1,689 assessed images in total. An approximately equal share spreads across happy (664) and mixed images (660), whereas the portion of neutral images is much lower (365) (see figure 9.A). The customer journey dataset reflects this distribution including 937,492 neutral, 1,502,605 mixed, and 1,566,974 happy images (see figure 9.B). Thus, during an average journey a customer viewed 1.15 neutral, 1.85 mixed, and 1.93 happy images.

The contact distribution of different image types shows isolated extreme values (contacts with happy images > 600, contacts with mixed images > 600, contacts with neutral images > 300) (see appendix A.B). However, these numbers are just related to long journeys with multiple sessions in which customers clicked on product images.

In conclusion, customers most often clicked on happy and mixed images. This is sensible as they jointly account for almost 80% of all images. Nevertheless, the number of interactions with neutral images were slightly higher (23.40%) than their portion on all images suggests (21.81%).

Figure 9: Emotionality of product pictures



Source: author's own illustration.

5.1.2 Categorization of Google Ad data

During the considered time period (March 11, 2019 – April 7, 2019) customers interacted with 145 different Google Ad descriptions. The minority shows strong positive emotions (11), followed by positive (27) and neutral emotionality (107) (see figure 10.A). This rank order alters in the customer journey dataset which comprises 11,308 positive ads, 41,238 strong positive, and 160,483 neutral descriptions (see figure 10.B). So, customers interacted more often with a strong positive compared to a merely positive Google Ad although the latter accounts for 2.5 times as many ads. During an average journey, a customer clicked on 0.19 neutral, 0.05 strong positive, and 0.02 positive ads.

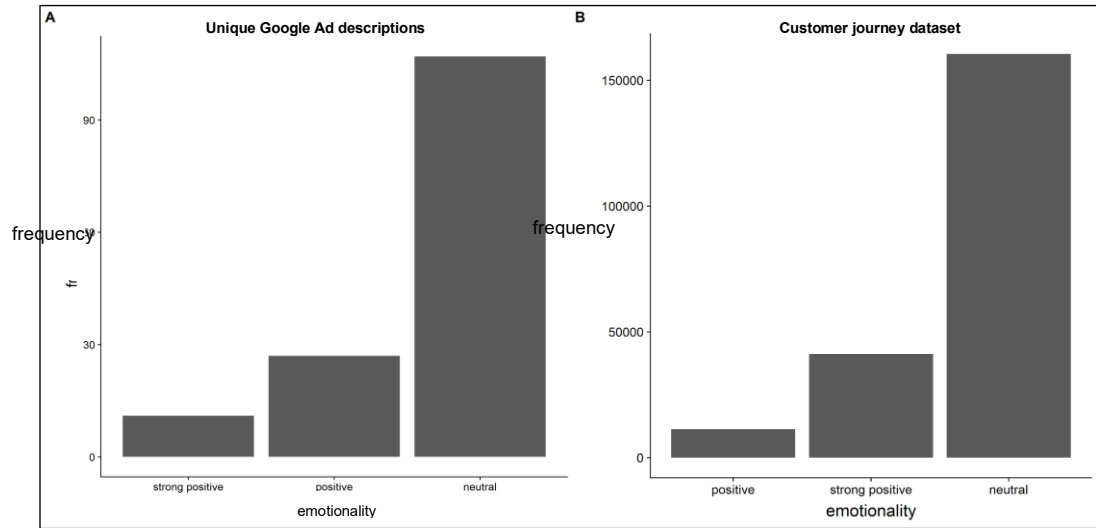
Single journeys show excessively high Google Ad contacts (contacts with neutral Google Ads > 100, contacts with strong positive Google Ad descriptions > 80, contacts with positive Google Ad descriptions > 40) (see appendix A.B). Like extreme numbers of image contacts, these values do not represent irregularities as they just represent long customer journeys with many sessions.

The results reveal a large gap between contact frequency with neutral compared to positive or strong positive Google Ad descriptions. However, it does not disclose the effectiveness of each type. Hence, I include all three CPC subgroups into channel attribution analysis to analyze their impact on each funnel stage.

Overall, I categorize UD and combine the extracted information with the initial journey dataset. By this, I create an extensive structured dataset as a foundation for

the following attribution analysis. Subsequently, I apply RFs combined with a Shapley Value approach to assess individual channel effects.

Figure 10: Emotionality of Google Ad descriptions



Source: author's own illustration.

5.2 Structured data analysis

Based on the expanded journey dataset, I construct RFs to model clicks on product overview pages (awareness stage), clicks on product detail pages (consideration stage), transaction revenue (purchase stage) and transaction probability (purchase stage). Thereafter, I apply the trained models on a new dataset. Finally, I evaluate channel effectiveness based on Shapley Values.

5.2.1 Random Forest

To calculate precise Shapley Values, it is beneficial to predict customer states as accurately as possible. So, I compare the results of two optimized RFs on every funnel stage - one without and the other with UD.

The incorporation of six additional variables, constructed from UD, increases the explained variance at the awareness stage by around 0.89 percentage points (see table 4). Furthermore, it reduces the MAE from 4.0301 to 3.9468 and the RMSE from 8.2917 to 8.0514. As the usage of UD improves the prediction accuracy, a better evaluation of channels' effectiveness is possible.

At the consideration stage incorporating UD leads to slightly better results. It increases the explained variance from 79.21% to 79.30% and reduces the MAE from

1.1675 to 1.1531. The RMSE slightly rises from 2.3994 to 2.4092. Thus, it marginally improves the prediction and subsequently channel attribution analysis.

With regard to transaction revenue, the prediction accuracy is relatively bad. Besides, the inclusion of UD does not enhance the results. This leads to an explained variance of 13.74% (with UD: 10.78%), a MAE of 15.3617 (with UD: 15.7246), and a RMSE of 126.7251 (with UD: 127.0237). As these results cannot serve as a reliable basis to attribute customer states to preceding channel contacts, I develop another model at the purchase stage using transaction as a binary variable.

To predict customers' transaction decision, I run a RF decision tree and use prediction accuracy, sensitivity, and specificity as evaluation criteria. This dramatically improves the results. The incorporation of UD marginally decreases the prediction accuracy from 96.67% to 96.58% and increases the sensitivity from 99.37% to 99.49%. However, the specificity drops rather strongly from 29.67% to 24.24%. As only a minority of all customers purchases at least one product (in test dataset: 3.8960%), the classification algorithm faces difficulties in predicting a transaction. This explains the much lower prediction accuracy regarding specificity.

Figure 11.A and figure 11.B strengthen the benefit of incorporating UD at the first both funnel stages. Especially, customers' contact frequency with different image types reveals useful information. Indeed, the three image variables show the highest prediction power (highest mean decrease in mean squared error) at the awareness stage (Liaw 2019). Especially mixed and happy images provide useful information. At the consideration stage, I only add CPC subgroups as additional independent variables. As product image clicks automatically implicate an interaction with the consideration stage, an incorporation of image categories is not sensible. Although CPC subgroups are not the most influential variables, they stronger impact RF's structure than multiple channels and cause an increase in the prediction accuracy. Therefore, their inclusion provides a valuable resource.

At the purchase stage UD does not improve prediction results in both model specifications (transaction revenue and transaction decision). The variable importance measurement (mean decrease in Gini index) underlines instead the impact of preceding funnel stages and the overall number of image clicks (see figure 11.C and figure 11.D) (Liaw 2019).

Although incorporating UD does not improve prediction results at the purchase stage, they partly support the first hypothesis at the awareness and consideration stage. Therefore, UD can represent a valuable resource to increase channel attribution accuracy along an e-commerce website funnel.

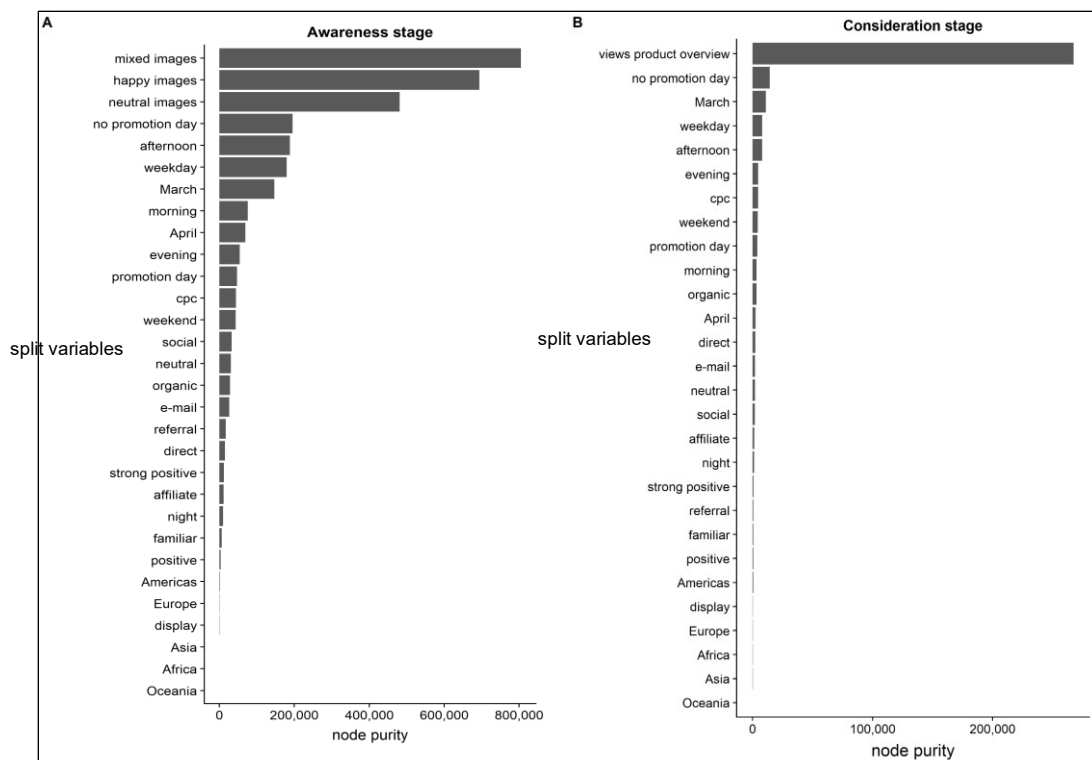
Even though the variable importance helps to understand RFs in more detail, they do not deliver information on channels' effectiveness. To evaluate the effectiveness of individual channels, a further model agnostic technique is necessary. Thus, I calculate Shapley Values of all eight channel types. As I am especially interested in the effect of CPC, I include all three subgroups for an in-depth examination.

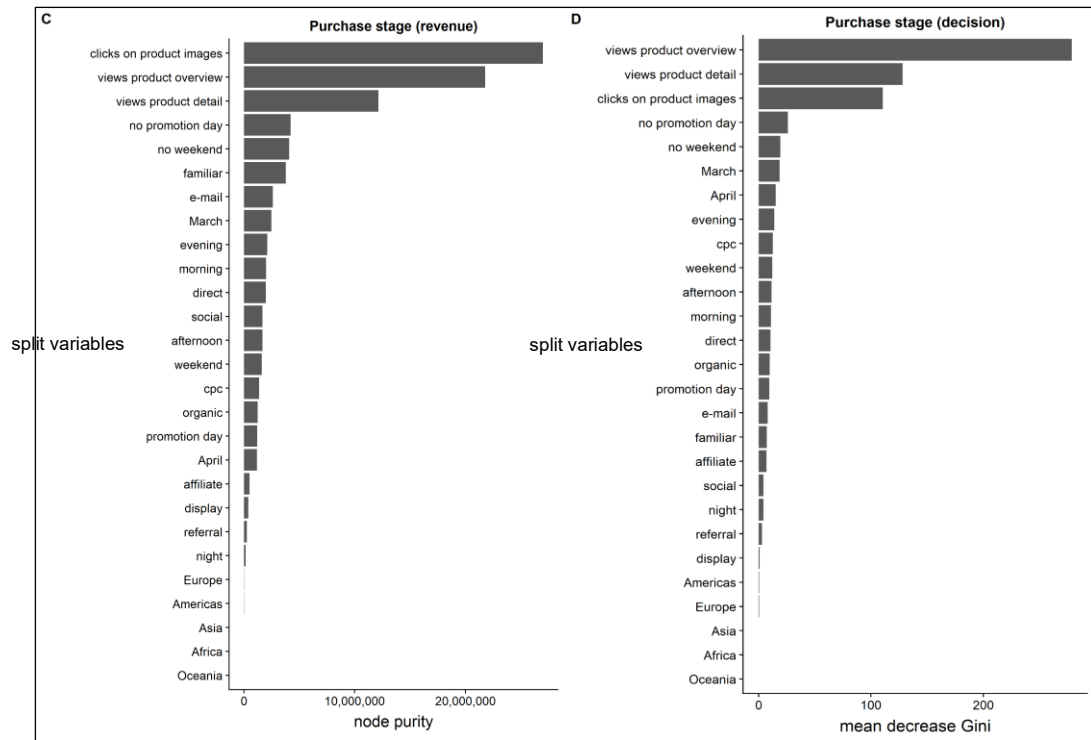
Table 4: Prediction results

	UD not included			UD included		
	MAE	RMSE	Explained variance	MAE	RMSE	Explained variance
Awareness	4.0301	8.2917	70.28%	3.9468	8.0514	71.17%
Consideration	1.1675	2.3994	79.21%	1.1531	2.4092	79.30%
Purchase (revenue)	15.3617	126.7251	13.74%	15.7246	127.0237	10.78%
	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity
Purchase (decision)	96.67%	99.37%	29.67%	96.58%	99.49%	24.24%

Source: author's own illustration.

Figure 11: Variable importance





Source: author's own illustration.

5.2.2 Shapley Values

As customers' decision-making varies during the purchase process, a stepwise analysis along funnel stages helps to better explain channel impacts (Frambach, Roest, and Krishnan 2007; Gardial et al. 1994; Haan, Wiesel, and Pauwels 2016; Kannan, Reinartz, and Verhoef 2016; Mittal, Kumar, and Tsiros 1999). The results indicate how the effectiveness differ at the awareness, consideration, and purchase stage.

To calculate advertisings' return on investment and optimize a channel strategy, a detailed understanding of individual effects is worthwhile. UD cannot only increase the overall prediction exactness and thus improve attribution accuracy but can also define separate channel subgroups. Three categories based on emotionality displayed in Google Ad descriptions reveal effect differences between CPC specifications.

For the first both funnel stages the rank order of channel effects is almost the same (see table 5). Customers who entered the e-commerce store through advertising on social media platforms indicate the highest Shapley Values (awareness stage: 15.3806, consideration stage: 7.1631). Subsequently, CPC exerts the second highest impact (awareness stage: 14.8668, consideration stage: 6.0494.). The classification into three subgroups reveals that neutral Google Ad descriptions constitute the most

effective CPC type (awareness stage: 16.1356, consideration stage: 6.5589), followed by strong positive (awareness stage: 15.7394, consideration stage: 6.1521), and positive descriptions (awareness stage: 13.9506, consideration stage: 5.6031). Thus, neutral and strong positive Google Ads show an even stronger impact on the first funnel stage than ads displayed on social media platforms. Customers who visited the e-commerce store by clicking on organic search results show the third highest values (awareness stage: 13.2700, consideration stage: 5.0287). Thereafter, direct site visits led to an average marginal increase in product overview page interactions (awareness stage) by 12.5528 clicks and product detail page interactions (consideration stage) by 5.0287 clicks. Following referral programs (awareness stage: 11.4601, consideration stage: 5.0287) and display ads (awareness stage: 10.6029, consideration stage: 4.6081) impact customers in descending order. Whereas at the awareness stage e-mail marketing (10.4122) exerts a stronger impact than affiliate marketing (9.0588), at the consideration stage affiliate marketing (4.4618) stronger impacts customers compared to e-mail marketing (4.3688).

As Shapley Values of the transaction revenue base on an insufficiently low prediction accuracy, I do not interpret these values. Instead, I analyze channel effects on the purchase stage by considering customers' overall transaction decision. In line with the effect order at the awareness and consideration stage, advertisements on social media platforms indicate the strongest impact (0.2052) at the purchase stage. Following, CPC shows the second highest value (0.1828). Again, three subgroups provide detailed insights into search advertisings, whereby neutral Google Ad descriptions exert the highest effect (0.2049), followed by strong positive (0.1920), and positive descriptions (0.1674). Customers who entered the e-commerce store via organic search results indicate the third highest Shapley Value (0.1684), while customers searched directly the e-commerce store show the fourth highest value (0.1604). Finally, affiliate marketing (0.1562), e-mail-links (0.1347), referral programs (0.1336), and display ads (0.1315) decreasingly affect customers' transaction decision. As the rank order of channel effectiveness varies between funnel stages, the results prove a dependence on customers' position in the purchase funnel.

In conclusion, Shapley Values offer a profound evaluation of all eight digital channels. The results of three CPC subgroups ensure a detailed understanding of search advertisings' effectiveness. Hence, it proves the second hypothesis stating that UD

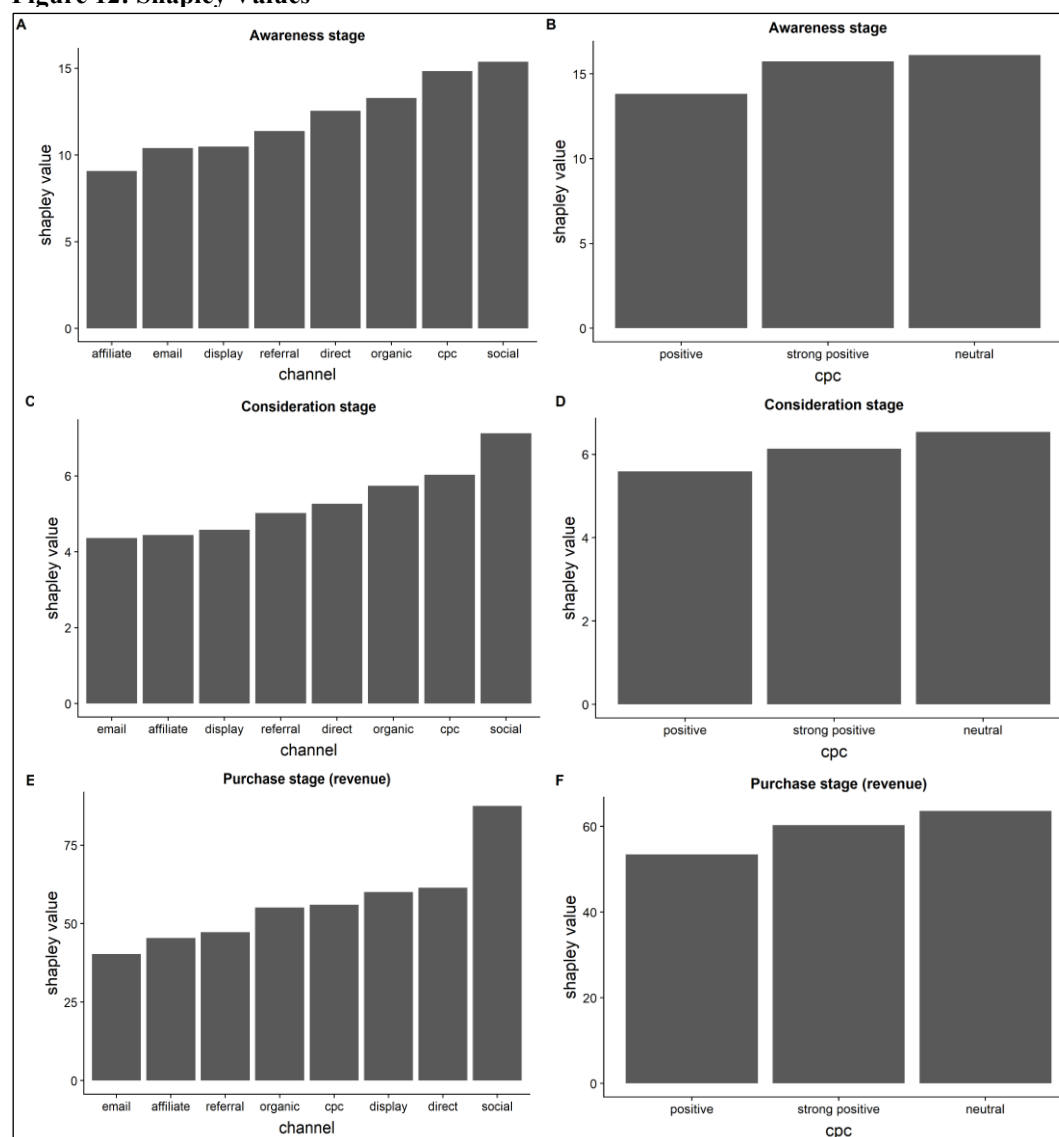
reveals effects of CPC subgroups along an e-commerce website funnel. Figure 12 represents how channel effects differ between funnel stages and definitely illustrates which channels are especially valuable. It foregrounds the influence of ads on social media and highlights neutral descriptions as the most valuable CPC type.

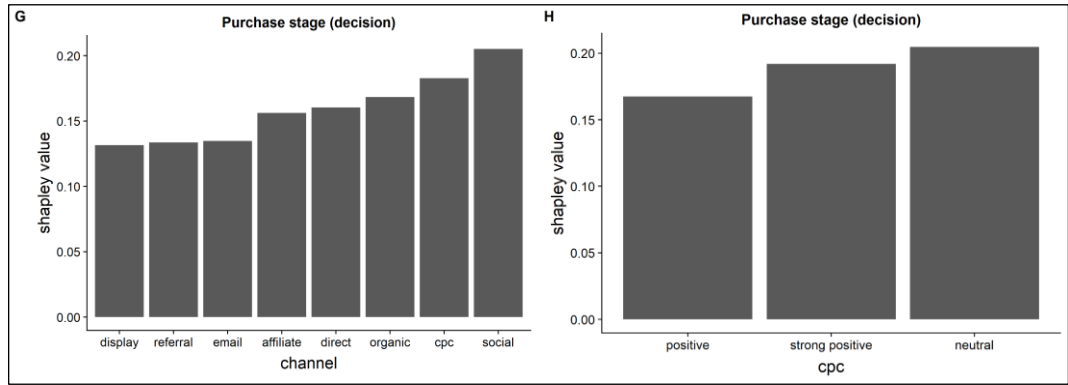
Table 5: Shapley Values

	Awareness	Consideration	Purchase (revenue)	Purchase (decision)
display	10.6029 (10.86%)	4.6081 (10.79%)	58.2784 (12.88%)	0.1315 (10.33%)
cpc	14.8668 (15.23%)	6.0494 (14.17%)	56.0448 (12.38%)	0.1828 (14.36%)
strong positive	15.7394 (34.35%)	6.1521 (33.59%)	60.1663 (33.98%)	0.1920 (34.03%)
positive	13.9506 (30.44%)	5.6031 (30.59%)	53.2658 (30.08%)	0.1674 (29.67%)
neutral	16.1356 (35.21%)	6.5589 (35.81%)	63.6485 (35.94%)	0.2048 (36.30%)
e-mail	10.4122 (10.67%)	4.3688 (10.23%)	39.8954 (8.82%)	0.1347 (10.58%)
affiliate	9.0588 (9.28%)	4.4618 (10.45%)	46.3757 (10.25%)	0.1562 (12.27%)
direct	12.5528 (12.86%)	5.2898 (12.39%)	61.4487 (13.58%)	0.1604 (12.60%)
organic	13.2700 (13.60%)	5.7338 (13.43%)	54.8901 (12.13%)	0.1684 (13.23%)
referral	11.4601 (11.74%)	5.0287 (11.67%)	47.6062 (10.52%)	0.1336 (10.50%)
social	15.3806 (15.76%)	7.1631 (16.77%)	87.9836 (19.44%)	0.2052 (16.12%)

Source: author's own illustration.

Figure 12: Shapley Values





Source: author's own illustration.

6 Discussion

6.1 Conclusion

This study elaborates on the benefits of incorporating UD in channel attribution research. Based on an extensive literature overview, I underline untapped potential and carve out several research gaps. As most studies focus on SD or limit UD to a CPC keyword classification, I expand existing research by the usage of image and text features. As channel effects depend on customers' funnel position, a stepwise analysis along different stages offers detailed insights.

I apply computer vision methodology (DCNN) on image data and sentiment analysis on text data to create six additional variables. These reveal customers' interaction frequency with different product image categories (positive emotionality, mixed emotionality, neutral emotionality) and CPC subgroups (strong positive emotionality, positive emotionality, neutral emotionality). The adoption of variables constructed from UD improves the prediction of customers' awareness and consideration. Besides, the subdivision into CPC categories reveals important effect differences in search engine advertising.

Next to incorporating different data types, I elaborate on the potential of machine learning algorithms in marketing research. As I use a DCNN and multiple RFs, this study relies on latest developments in the machine learning environment.

The rank order of channel effects does not considerably differ between funnel stages. Referral programs, display ads, email-links, and affiliate marketing constantly belong to the group of less effective channel types. However, within this group the effectiveness remarkably changes from the consideration to the purchase stage. The Shapley Value of referral programs switches from second worst to the fifth position,

display ads deteriorate from the sixth position to the least effective channel type, e-mails improve from the last to the sixth rank, and affiliate marketing improves from the seventh to the fifth position. Thus, display ads and referral programs are more effective in arousing customers' awareness and consideration than impacting their final purchase intention. Instead affiliate and e-mail marketing are better to stimulate customers' final buying interest than increasing preceding awareness and consideration.

Social media advertising indicates the highest Shapley Values on all three funnel stages. It offers great opportunities to impact customers' awareness, consideration and purchase intention. Therefore, social media ads do not only arouse customers' interest but also affect purchase decisions. The huge popularity of Instagram, YouTube or Snapchat provides a unique opportunity to attract and engage customers. Social media platforms shift marketing communications from a one-way to a two-way interaction. This strengthens the customer-firm relationship and fosters the development of brand communities (Kim and Ko 2012, p. 1480; Vries, Gensler, and Leeflang 2012, p. 83). Customers often perceive advertising on social media as an entertaining inspiration rather than an annoying sales message. Besides, corporations with a large network face the opportunity to address many customers' without spending money on sponsored advertising. This further improves the channel effectiveness and underlines the potential of social media ads.

A strong impact of CPC on customer states along the purchase funnel confirms previous research findings that emphasize the role of search advertising as an effective marketing channel (Ghose and Yang 2009; Li et al. 2016). However, this study goes beyond existing literature by analyzing the effect of different Google Ad description categories. Consistent results across all funnel stages show the strongest effect for neutral descriptions, followed by strong positive and positive versions. Customers seem to follow an "all-or-nothing" interest as they prefer either Google Ad descriptions without any emotion or with a strong positive emotion over ads displaying a mixture. The strong appreciation of neutral descriptions suggest that customers are especially interested in collecting helpful information and fact-based discount depictions.

It is rather surprising that website visits via social ads, CPC, or organic search results show higher Shapley Values than direct store accesses. However, customers

directly visiting an e-commerce platform are commonly familiar with the store structure. As they already know which product categories they are interested in, their clicking process is rather straightforward. This explains lower Shapley Values at the awareness and consideration stage. Nonetheless, the results still contradict the expectation that familiar customers show a higher purchase probability. Continuously, long-term customers just check new product offers without a specific purchase intention on a regular basis. Therefore, a lower Shapley Value at the purchase stage is reasonable.

Shapley Values do not indicate how managers should optimally allocate the marketing budget since further aspects like advertising costs are not considered. The results do not represent prescriptive values which recommend a specific course of action. Nonetheless, they reveal impacts of different marketing channels. Managers can combine these findings with cost figures and strategic consideration (e.g. which customer state should be particularly focused on) to improve overall channel strategies. Thus, Shapley Values constitute an important component in channel attribution analysis.

To conclude, this study shows how UD can improve overall channel attribution accuracy, reveals the effect size of CPC subgroups and scrutinizes channels' impact on different purchase funnel steps.

6.2 Managerial implications

Although corporations have access to a large volume and variety of data, they do not use its full potential to optimize attribution methods. Alike, corporations often rely on simple heuristics but do not benefit from rapid progress in machine learning algorithms. If they however decide to adopt these approaches, the attribution methodology of Google Analytics represents a popular option. However, it is ambiguous how Google comes up with predictions before allocating them to preceding channel contacts. This study provides a guideline to improve attribution analyses. Marketing managers should appreciate the value of UD to increase attribution accuracy and assess the effectiveness of channel subgroups.

In general, channels with a strong (weak) performance on one funnel stage, also indicate a strong (weak) impact on both other stages. Even though channels' effectiveness does not considerably differ between funnel stages, slight variations

(awareness / consideration vs. purchase stage) reveal a dependence on customers' funnel position. Marketing managers should keep the funnel framework in mind to clearly define which channel should focus on which customer state and distinctively evaluate channel effects.

Advertisements on social media platforms exert the strongest impact on customers along the purchase funnel. It underlines the importance of attracting and engaging customers in digital brand communities. Marketing managers need to ensure that corporations keep up with the emergence of new communication mediums. They should heavily invest in a strong social media presence as a main advertising tool.

Besides, search advertising constitutes a valuable marketing activity. Both CPC and organic search results belong to the most effective channel types. Therefore, marketing departments have to spend money on sponsored ads and improve organic search terms' ranking by search engine optimization. In doing so, they should not only think about which particular search terms to promote but also esteem the related Google Ad descriptions. The results of this study recommend the usage of neutral emotionality. Corporations should focus on insightful information and specific discount statements.

6.3 Limitations and future research

This study grounds on an extensive database, latest machine learning algorithms and a well-structured attribution framework. Nonetheless, several methodological and content-related limitations offer opportunities for future research.

Obviously, customers do not click on every marketing channel they are observing along the purchase funnel. Though, a channel can also impact the decision-making process without causing a click-interaction. As the underlying dataset only covers channels on which customers clicked, it does not capture every channel effect. Eye-tracking methodologies could possibly find a remedy to understand which channels attract visual contacts but do not cause a direct interaction.

To model customers' purchase behavior, I restructure the initial session-level data into a journey dataset. Therefore, I extract and combine numerous single- and multi-session journeys. Due to simplicity reasons, I only include the first multi-session journey for every customer. Likewise, I limit the training dataset (20,000 journeys),

validation dataset (20,000 journeys) and test dataset (100,000 journeys) due to computing power restrictions. Although this does not bias the results, I do not exploit all available information in the data. However, it is questionable if extending the dataset would improve the results as the marginal increase in prediction power decreases with the number of observations. Furthermore, I group all sessions before a final purchase decision into one customer journey. As the considered time period is limited to one month, this constitutes a comprehensible approach. Nevertheless, further research should analyze if different journey conceptualizations (e.g. journey ends if time gap between consecutive sessions exceeds a pre-defined threshold) lead to different results.

Clicks on product overview pages, product detail pages, and transactions do not perfectly explain customers' state of mind. However, it is essential to understand why customers decide in a certain way. Alternative dependent variables such as the time spend on a website could be helpful to model customers' purchase process and get a more complete picture of channels' effectiveness.

The underlying dataset does not provide Google Ad descriptions for all customers who clicked on CPC. Thus, the analysis of strong positive, positive, and neutral ad descriptions bases on a lower number of observations. But also in this case, it is not guaranteed that more descriptions would significantly improve model outcomes.

Incorporating UD to estimate customer states along a purchase funnel, enhances the prediction accuracy at the awareness and consideration stage. Although the results indicate the potential of UD, it only leads to a relatively small improvement. Future research should elaborate on this by including more information from image and text data. Especially, the former offers a huge variety of components which this study has not yet incorporated. The Microsoft Azure Image Recognition API provides excellent opportunities to benefit from all kinds of image features. Another opportunity is the development of an own DCNN to extract specific visual information of interest. Valuing the importance of social media advertising, the inclusion of UD offers huge potential.

This study combines the results of a supervised machine learning methodology (RFs) with the Shapley Value approach to evaluate channel effectiveness. Although machine learning algorithms often allow to improve the prediction accuracy, they commonly resemble a black box model not revealing variable parameters. Future

research should explicitly compare the results of a channel attribution analysis based on machine learning approaches with the outcomes of “standard” econometric methodologies.

The Shapley Value approach attributes predicted results entirely to preceding channel contacts. However, also other factors like customers’ personal motivation, financial resources, or previous buying behavior cause significant impact. Thus, Shapley Values should be carefully interpreted and rather be used to evaluate channels’ effectiveness rank order. Future research should compare the application of Shapley Values with results of alternative model agnostic techniques as LIME or SHAP.

Additional explanatory variables could help to improve the prediction accuracy. Thus, marketing research should strive to optimize model specifications by incorporating further customer and situational characteristics. Particularly, the differentiation between mobile and desktop channel interactions could reveal interesting research findings and help to improve models’ validity.

This study does not present a sufficient guideline to finally optimize channel strategies since managers have to consider additional components as financial restrictions. Future research should elaborate on this by developing a methodology which includes all necessary components to deduce an optimal marketing budget allocation.

References

- Abhishek, Vibhanshu, Peter Fader, and Kartik Hosanagar (2012), "Media Exposure Through the Funnel: A Model of Multi-Channel Attribution," working paper, Hein College/Wharton School, Carnegie Mellon University / University of Pennsylvania.
- Anderl, Eva, Ingo Becker, Florian von Wangenheim, and Jan H. Schumann (2016), "Mapping the Customer Journey: Lessons Learned from Graph-Based Online Attribution Modeling," *International Journal of Research in Marketing*, 33 (3), 457–74.
- Archak, Nikolay, Anindya Ghose, and Panagiotis G. Ipeirotis (2011), "Deriving the Pricing Power of Product Features by Mining Consumer Reviews," *Management Science*, 57 (8), 1485–509.
- Balasubramanian, Sridhar, Rajagopal Raghunathan, and Vijay Mahajan (2005), "Consumers in a Multichannel Environment: Product Utility, Process Utility, and Channel Choice," *Journal of Interactive Marketing*, 19 (2), 12–30.
- Balducci, Bitty and Detelina Marinova (2018), "Unstructured Data in Marketing," *Journal of the Academy of Marketing Science*, 46 (4), 557–90.
- Barajas, Joel, Ram Akella, Marius Holtan, and Aaron Flores (2016), "Experimental Designs and Estimation for Online Display Advertising Attribution in Marketplaces," *Marketing Science*, 35 (3), 465–83.
- Batra, Rajeev and Kevin L. Keller (2016), "Integrating Marketing Communications: New Findings, New Lessons, and New Ideas," *Journal of Marketing*, 80 (6), 122–45.
- Berman, Ron (2018), "Beyond the Last Touch: Attribution in Online Advertising," *Marketing Science*, 37 (5), 771–92.
- Blanco, Carlos F., Raquel G. Sarasa, and Carlos O. Sanclemente (2010), "Effects of Visual and Textual Information in Online Product Presentations: Looking for the Best Combination in Website Design," *European Journal of Information Systems*, 19 (6), 668–86.
- Blattberg, Robert C., Richard Briesch, and Edward J. Fox (1995), "How Promotions Work," *Marketing Science*, 14 (3), 122–32.

- Blumberg, Robert and Shaku Atre (2003), “The Problem with Unstructured Data,” *Data Management Review*, 13 (4), 42–49.
- Bradlow, Eric T., Manish Gangwar, Praveen Kopalle, and Sudhir Voleti (2017), “The Role of Big Data and Predictive Analytics in Retailing,” *Journal of Retailing*, 93 (1), 79–95.
- Braun, Michael and Wendy W. Moe (2013), “Online Display Advertising: Modeling the Effects of Multiple Creatives and Individual Impression Histories,” *Marketing Science*, 32 (5), 753–67.
- Breiman, Leo (1996), “Bagging Predictors,” *Machine Learning*, 24 (2), 123–40.
- (2001), “Random Forests,” *Machine Learning*, 45 (1), 5–32.
- Bucklin, Randolph E. and Catarina Sismeiro (2003), “A Model of Web Site Browsing Behavior Estimated on Clickstream Data,” *Journal of Marketing Research*, 40 (3), 249–67.
- Burell, Alexandra and Sharon Terlep (2017), “P&G Cuts More than \$100 Million in ‘Largely Ineffective’ Digital Ads,” (accessed May 30, 2019), [available at <https://www.wsj.com/articles/p-g-cuts-more-than-100-million-in-largely-ineffective-digital-ads-1501191104>].
- Chang, Yu-Ting, Hueiju Yu, and Hsi-Peng Lu (2015), “Persuasive Messages, Popularity Cohesion, and Message Diffusion in Social Media Marketing,” *Journal of Business Research*, 68 (4), 777–82.
- Chen, Dong, Shaoqing Ren, Yichen Wei, Xudong Cao, and Jian Sun (2014), “Joint Cascade Face Detection and Alignment,” in *Computer Vision – ECCV 2014. Lecture Notes in Computer Science*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, eds. Zurich, Springer International Publishing, 109–22.
- Chevalier, Judith A. and Dina Mayzlin (2006), “The Effect of Word of Mouth on Sales: Online Book Reviews,” *Journal of Marketing Research*, 43 (3), 345–54.
- Childers, Terry L., Christopher L. Carr, Joann Peck, and Stephen Carson (2001), “Hedonic and Utilitarian Motivations for Online Retail Shopping Behavior,” *Journal of Retailing*, 77 (4), 511–35.

- Colicev, Anatoli, Ashish Kumar, and Peter O'Connor (2018), "Modeling the Relationship Between Firm and User Generated Content and the Stages of the Marketing Funnel," *International Journal of Research in Marketing*, 36 (1), 100–16.
- Coussement, Kristof and Koen W. de Bock (2013), "Customer Churn Prediction in the Online Gambling Industry: The Beneficial Effect of Ensemble Learning," *Journal of Business Research*, 66 (9), 1629–36.
- and Dirk van den Poel (2008), "Integrating the Voice of Customers Through Call Center Emails into a Decision Support System for Churn Prediction," *Information & Management*, 45 (3), 164–74.
- Cyr, Dianne, Milena Head, Hector Larios, and Bing Pan (2009), "Exploring Human Images in Website Design: A Multi-Method Approach," *MIS Quarterly*, 33 (3), 539–66.
- Dalessandro, Brian, Claudia Perlich, Ori Stitelman, and Foster Provost (2012), "Causally Motivated Attribution for Online Advertising," in *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*, Ying Li, Ti_Yan Liu, Tao Quin, Esin Saka, James Shanahan, and Dou Shen, eds. New York, ACM Press, 1–9.
- Danaher, Peter J. and Tracey S. Dagger (2013), "Comparing the Relative Effectiveness of Advertising Channels: A Case Study of a Multimedia Blitz Campaign," *Journal of Marketing Research*, 50 (4), 517–34.
- and Harald J. van Heerde (2018), "Delusion in Attribution: Caveats in Using Attribution for Multimedia Budget Allocation," *Journal of Marketing Research*, 55 (5), 667–85.
- Delen, Dursun and Hamed M. Zolbanin, "The Analytics Paradigm in Business Research," *Journal of Business Research*, 90 (June), 186–95.
- Dhar, Vasant and Elaine A. Chang (2009), "Does Chatter Matter? The Impact of User-Generated Content on Music Sales," *Journal of Interactive Marketing*, 23 (4), 300–07.

- Dzyabura, Daria, Siham El Kihal, and Marat Ibragimov (2018), “Leveraging the Power of Images in Predicting Product Return Rates,” working paper, New York Universit / Frankfurt School of Finance and Management.
- Edelman, Benjamin and Wesley Brandi (2015), “Risk, Information, and Incentives in Online Affiliate Marketing,” *Journal of Marketing Research*, 52 (1), 1–12.
- Erevelles, Sunil, Nobuyuki Fukawa, and Linda Swayne (2016), “Big Data Consumer Analytics and the Transformation of Marketing,” *Journal of Business Research*, 69 (2), 897–904.
- Felix, Reto, Philipp A. Rauschnabel, and Chris Hinsch (2017), “Elements of Strategic Social Media Marketing: A Holistic Framework,” *Journal of Business Research*, 70, 118–26.
- Frambach, Ruud T., Henk C.A. Roest, and Trichy V. Krishnan (2007), “The Impact of Consumer Internet Experience on Channel Preference and Usage Intentions Across the Different Stages of the Buying Process,” *Journal of Interactive Marketing*, 21 (2), 26–41.
- Gandomi, Amir and Murtaza Haider (2015), “Beyond the Hype: Big Data Concepts, Methods, and Analytics,” *International Journal of Information Management*, 35 (2), 137–44.
- Gardial, Sarah F., D. S. Clemons, Robert B. Woodruff, David W. Schumann, and Mary J. Burns (1994), “Comparing Consumers’ Recall of Prepurchase and Postpurchase Product Evaluation Experiences,” *Journal of Consumer Research*, 20 (4), 548–60.
- Gensler, Sonja, Peter C. Verhoef, and Martin Böhm (2012), “Understanding Consumers’ Multichannel Choices Across the Different Stages of the Buying Process,” *Marketing Letters*, 23 (4), 987–1003.
- Ghose, Anindya and Vilma Todri-Adamopoulos (2016), “Toward a Digital Attribution Model: Measuring the Impact of Display Advertising on Online Consumer Behavior,” *MIS Quarterly*, 40 (4), 889–910.

- and Sha Yang (2009), “An Empirical Analysis of Search Engine Advertising: Sponsored Search in Electronic Markets,” *Management Science*, 55 (10), 1605–22.
- Goldfarb, Avi and Catherine Tucker (2011), “Online Display Advertising: Targeting and Obtrusiveness,” *Marketing Science*, 30 (3), 389–404.
- Google LLC (2019a), “Cloud Translation Documentation,” (accessed May 14, 2019), [available at <https://cloud.google.com/translate/docs/>].
- (2019b), “Data-Driven Attribution Methodology. Understand the Science Behind Data-Driven Attribution,” (accessed May 1, 2019), [available at <https://support.google.com/analytics/answer/3191594?hl=en>].
- Guo, Zhiling (2012), “Optimal Decision Making for Online Referral Marketing,” *Decision Support Systems*, 52 (2), 373–83.
- Haan, Evert de, Thorsten Wiesel, and Koen Pauwels (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.
- Hoban, Paul R. and Randolph E. Bucklin (2015), “Effects of Internet Display Advertising in the Purchase Funnel: Model-Based Insights from a Randomized Field Experiment,” *Journal of Marketing Research*, 52 (3), 375–93.
- Homburg, Christian, Laura Ehm, and Martin Artz (2015), “Measuring and Managing Consumer Sentiment in an Online Community Environment,” *Journal of Marketing Research*, 52 (5), 629–41.
- Hu, Minqing and Bing Liu (2004), “Mining and Summarizing Customer Reviews,” in *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Won Kim, Ronny Kohavi, Johannes Gehrke, and William DuMouchel, eds. New York, ACM Press, 168–77.
- Huneke, Mary E., Catherine Cole, and Irwin P. Levin (2004), “How Varying Levels of Knowledge and Motivation Affect Search and Confidence During Consideration and Choice,” *Marketing Letters*, 15 (2/3), 67–79.

- Inman, J. Jeffrey, Russell S. Winer, and Rosellina Ferraro (2009), "The Interplay Among Category Characteristics, Customer Characteristics, and Customer Activities on In-Store Decision Making," *Journal of Marketing*, 73 (5), 19–29.
- Ji, Wendi, Xiaoling Wang, and Dell Zhang (2016), "A Probabilistic Multi-Touch Attribution Model for Online Advertising," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, Snehasis Mukhopadhyay, Yunyao Li, Parikshit Sondhi, ChengXiang Zhai, Elisa Bertino, Fabio Crestani, Javed Mostafa, Jie Tang, Luo Si, Xiaofang Zhou, and Yi Chang, eds. New York, 1373–82.
- Joo, Mingyu, Kenneth C. Wilbur, and Yi Zhu (2016), "Effects of TV Advertising on Keyword Search," *International Journal of Research in Marketing*, 33 (3), 508–23.
- Kannan, P. K. and Hongshuang "A." Li (2017), "Digital Marketing: A Framework, Review and Research Agenda," *International Journal of Research in Marketing*, 34 (1), 22–45.
- , Werner Reinartz, and Peter C. Verhoef (2016), "The Path to Purchase and Attribution Modeling: Introduction to Special Section," *International Journal of Research in Marketing*, 33 (3), 449–56.
- Kim, Angella J. and Eunju Ko (2012), "Do Social Media Marketing Activities Enhance Customer Equity? An Empirical Study of Luxury Fashion Brand," *Journal of Business Research*, 65 (10), 1480–86.
- Kireyev, Pavel, Koen Pauwels, and Sunil Gupta (2016), "Do Display Ads Influence Search? Attribution and Dynamics in Online Advertising," *International Journal of Research in Marketing*, 33 (3), 475–90.
- Klostermann, Jan, Anja Plumeyer, Daniel Böger, and Reinhold Decker (2018), "Extracting Brand Information from Social Networks: Integrating Image, Text, and Social Tagging Data," *International Journal of Research in Marketing*, 35 (4), 538–56.
- Kollat, David T. and Ronald P. Willett (1967), "Customer Impulse Purchasing Behavior," *Journal of Marketing Research*, 4 (1), 21–31.

- Kumar, Ashish, Ram Bezawada, Rishika Rishika, Ramkumar Janakiraman, and P. K. Kannan (2016), "From Social to Sale: The Effects of Firm-Generated Content in Social Media on Customer Behavior," *Journal of Marketing*, 80 (1), 7–25.
- Lemon, Katherine N. and Peter C. Verhoef (2016), "Understanding Customer Experience Throughout the Customer Journey," *Journal of Marketing*, 80 (6), 69–96.
- Li, Hongshuang and P. K. Kannan (2014), "Attributing Conversions in a Multi-channel Online Marketing Environment: An Empirical Model and a Field Experiment," *Journal of Marketing Research*, 51 (1), 40–56.
- , ———, Siva Viswanathan, and Abhishek Pani (2016), "Attribution Strategies and Return on Keyword Investment in Paid Search Advertising," *Marketing Science*, 35 (6), 831–48.
- Liaw, Andy (2019), "randomForest. Classification and Regression with Random Forest," [available at <https://www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/randomForest>].
- and Matthew Wiener (2002), "Classification and Regression by randomForest," 2 (3), 18–22.
- Liu, Xiao, Param V. Singh, and Kannan Srinivasan (2016), "A Structured Analysis of Unstructured Big Data by Leveraging Cloud Computing," *Marketing Science*, 35 (3), 363–88.
- Lundberg, Scott M. and Lee Su-In (2017), "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, 31 (December), 4765–74.
- Mallapragada, Girish, Sandeep R. Chandukala, and Qing Liu (2016), "Exploring the Effects of "What" (Product) and "Where" (Website) Characteristics on Online Shopping Behavior," *Journal of Marketing*, 80 (2), 21–38.
- Manchanda, Puneet, Jean-Pierre Dubé, Khim Y. Goh, and Pradeep K. Chintagunta (2006), "The Effect of Banner Advertising on Internet Purchasing," *Journal of Marketing Research*, 43 (1), 98–108.

- Marketing Science Institute (2016), *Research Priorities 2016-2018*. Cambridge, Mass.: Marketing Science Institute.
- (2018), *Research Priorities 2018-2020*. Cambridge, Mass.: Marketing Science Institute.
- McIntyre, Ewan and Anna M. Virzi (2018), *CMO Spend Survey 2018-2019. Marketers Proceed into Uncharted Waters with Confidence*. Stamford, CT: Gartner.
- Microsoft, “Face Recognition API,” (accessed May 15, 2019), [available at <https://docs.microsoft.com/en-us/azure/cognitive-services/face/overview>].
- Microsoft, “Face Recognition API Demonstration version,” (accessed June 16, 2019), [available at <https://azure.microsoft.com/de-de/services/cognitive-services/face/#recognition>].
- Mittal, Vikas, Pankaj Kumar, and Michael Tsiros (1999), “Attribute-Level Performance, Satisfaction, and Behavioral Intentions over Time: A Consumption-System Approach,” *Journal of Marketing*, 63 (2), 88–101.
- Moe, Wendy W. and Brian T. Ratchford (2018), “How the Explosion of Customer Data Has Redefined Interactive Marketing,” *Journal of Interactive Marketing*, 42, A1-A2.
- Moorman, Christine (2018), *The CMO Survey. Predicting the Future of Markets, Tracking Marketing Excellence, and Improving the Value of Marketing Since 2008*. Durham, NC: Fuqua School of Business.
- Nam, Hyoryung, Yogesh V. Joshi, and P. K. Kannan (2017), “Harvesting Brand Information from Social Tags,” *Journal of Marketing*, 81 (4), 88–108.
- and P. K. Kannan (2014), “The Informational Value of Social Tagging Networks,” *Journal of Marketing*, 78 (4), 21–40.
- Neslin, Scott A., Dhruv Grewal, Robert Leghorn, Venkatesh Shankar, Marije L. Teerling, Jacquelyn S. Thomas, and Peter C. Verhoef (2006), “Challenges and Opportunities in Multichannel Customer Management,” *Journal of Service Research*, 9 (2), 95–112.

- Netzer, Oded, Ronen Feldman, Jacob Goldenberg, and Moshe Fresko (2012), “Mine Your Own Business: Market-Structure Surveillance Through Text Mining,” *Marketing Science*, 31 (3), 521–43.
- Pang, Bo and Lillian Lee (2008), “Opinion Mining and Sentiment Analysis,” *Foundations and Trends in Information Retrieval*, 2 (2), 1–135.
- , ———, and Shivakumar Vaithyanathan (2002), “Thumbs up?,” in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, unknown, eds. Morristown: Association for Computational Linguistics, 79–86.
- Probst, Philipp, Marvin N. Wright, and Anne-Laure Boulesteix (2019), “Hyperparameters and Tuning Strategies for Random Forest,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9 (3), 1–15.
- Putka, Dan J., Adam S. Beatty, and Matthew C. Reeder (2018), “Modern Prediction Methods: New Perspectives on a Common Problem,” *Organizational Research Methods*, 21 (3), 689–732.
- Ren, Kan, Yuchen Fang, Weinan Zhang, Shuhao Liu, Jiajun Li, Ya Zhang, Yong Yu, and Jun Wang (2018), “Learning Multi-touch Conversion Attribution with Dual-Attention Mechanisms for Online Advertising,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, Alfredo Cuzzocrea, Assaf Schuster, Haixun Wang, James Allan, Norman Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Broder, Mohammed Zaki, Selcuk Candan, and Alexandros Labrinidis, eds. New York, ACM Press, 1433–42.
- Ribeiro, Marco T., Sameer Singh, and Carlos Guestrin (2016), “Model-Agnostic Interpretability of Machine Learning,” in *Proceedings of ICML Workshop on Human Interpretability in Machine Learning*, Been Kim, Kush R. Varshney, and Adrian Weller, eds. New York, 91–95.
- Rinker, Tyler, “Sentiment Analysis,” (accessed May 14, 2019), [available at <https://www.rdocumentation.org/packages/qdap/versions/2.3.2/topics/polarity>].
- Rosen, Deborah E. and Elizabeth Purinton (2004), “Website Design,” *Journal of Business Research*, 57 (7), 787–94.

- Rust, Roland T., Tim Ambler, Gregory S. Carpenter, V. Kumar, and Rajendra K. Srivastava (2004), "Measuring Marketing Productivity: Current Knowledge and Future Directions," *Journal of Marketing*, 68 (4), 76–89.
- Rutz, Oliver J., Michael Trusov, and Randolph E. Bucklin (2011), "Modeling Indirect Effects of Paid Search Advertising: Which Keywords Lead to More Future Visits?," *Marketing Science*, 30 (4), 646–65.
- Sahni, Navdeep S., Christian Wheeler, and Pradeep Chintagunta (2018), "Personalization in Email Marketing: The Role of Noninformative Advertising Content," *Marketing Science*, 37 (2), 236–58.
- Schwarz, Norbert (2004), "Metacognitive Experiences in Consumer Judgment and Decision Making," *Journal of Consumer Psychology*, 14 (4), 332–48.
- Schweidel, David A. and Wendy W. Moe (2014), "Listening in on Social Media: A Joint Model of Sentiment and Venue Format Choice," *Journal of Marketing Research*, 51 (4), 387–402.
- Shao, Xuhui and Lexin Li (2011), "Data-Driven Multi-Touch Attribution Models," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chid Apte, Joydeep Ghosh, and Padhraic Smyth, eds. New York, ACM Press, 258–64.
- Shapley, Lloyd S. (1953), "A Value for n-Person Games," in *Contributions to the Theory of Games*, Vol. 2, Harold W. Kuhn and Albert W. Tucker, eds. Princeton, Princeton University Press, 307–18.
- Shocker, Allan D., Moshe Ben-Akiva, Bruno Boccara, and Prakash Nedungadi (1991), "Consideration Set Influences on Consumer Decision-Making and Choice: Issues, Models, and Suggestions," *Marketing Letters*, 2 (3), 181–97.
- Statista (2019), "Anzahl der Täglich Aktiven Instagram Stories Nutzer Weltweit in Ausgewählten Monaten von Oktober 2016 bis Januar 2019 (in Millionen)," (accessed June 2, 2019), [available at <https://de.statista.com/statistik/daten/studie/659687/umfrage/taeglich-aktive-nutzer-von-instagram-stories-weltweit/>].
- Stephen, Andrew T. and Jeff Galak (2012), "The Effects of Traditional and Social Earned Media on Sales: A Study of a Microlending Marketplace," *Journal of Marketing Research*, 49 (5), 624–39.

- Trusov, Michael, Randolph E. Bucklin, and Koen Pauwels (2009), “Effects of Word-of-Mouth versus Traditional Marketing: Findings from an Internet Social Networking Site,” *Journal of Marketing*, 73 (5), 90–102.
- Verhoef, Peter C. and Peter S.H. Leeflang (2009), “Understanding the Marketing Department’s Influence within the Firm,” *Journal of Marketing*, 73 (2), 14–37.
- Webster, Frederick E. (2006), “Back to the Future: Integrating Marketing as Tactics, Strategy, and Organizational Culture,” *Journal of Marketing*, 69 (4), 1–25.
- Wiesel, Thorsten, Koen Pauwels, and Joep Arts (2011), “Practice Prize Paper – Marketing’s Profit Impact: Quantifying Online and Offline Funnel Progression,” *Marketing Science*, 30 (4), 604–11.
- Xu, Lizhen, Jason A. Duan, and Andrew Whinston (2014), “Path to Purchase: A Mutually Exciting Point Process Model for Online Advertising and Conversion,” *Management Science*, 60 (6), 1392–412.
- Yang, Sha and Anindya Ghose (2010), “Analyzing the Relationship Between Organic and Sponsored Search Advertising: Positive, Negative, or Zero Interdependence?,” *Marketing Science*, 29 (4), 602–23.
- Yu, Zhiding and Cha Zhang (2015), “Image based Static Facial Expression Recognition with Multiple Deep Network Learning,” in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, Zhengyou Zhang, Phil Cohen, Dan Bohus, Radu Horaud, and Helen Meng, eds. New York, ACM Press, 435–42.
- Zhang, Cha and Zhengyou Zhang, “Improving Multiview Face Detection with Multi-Task Deep Convolutional Neural Networks,” in *Proceedings of the 2014 IEEE Winter Conference on Applications of Computer Vision*, Walter Scheirer, Ruigang Yang, and Charles Stewart, eds. New York, IEEE Computer Society, 1036–41.
- Zhang, Xi, V. Kumar, and Koray Cosguner (2017), “Dynamically Managing a Profitable Email Marketing Program,” *Journal of Marketing Research*, 54 (6), 851–66.
- Zhang, Ya, Yi Wei, and Jianbiao Ren, “Multi-touch Attribution in Online Advertising with Survival Theory,” in *Proceedings of the 2014 IEEE International*

Conference on Data Mining, Ravi Kumar, Hannu Toivonen, Jian Pei, Joshua Z. Huang, and Xindong Wu, eds. New York, IEEE Computer Society, 687–96.

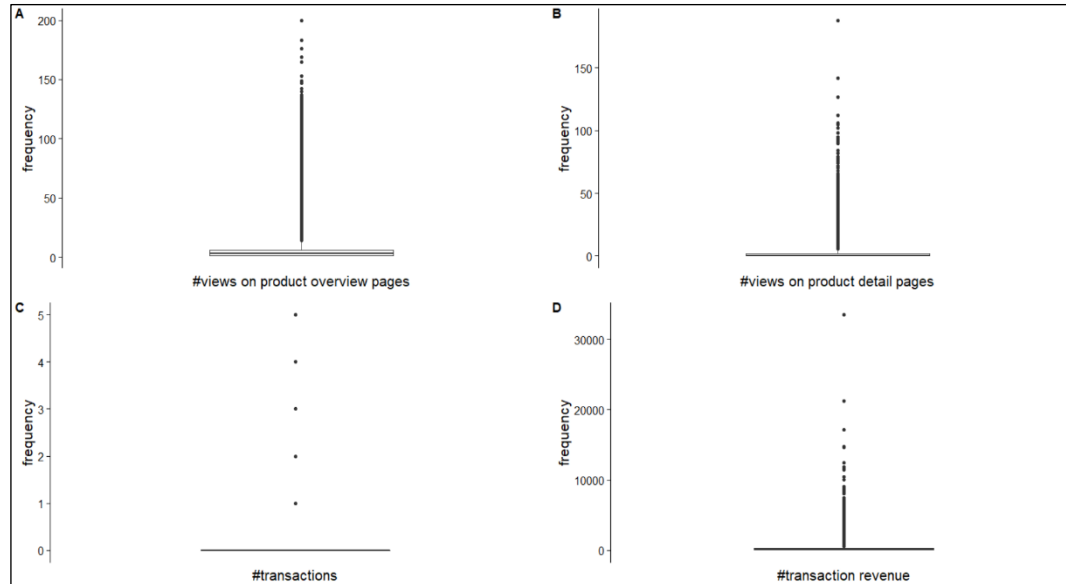
Zhu, Xiangxin and Deva Ramanan, “Face Detection, Pose Estimation, and Landmark Localization in the Wild,” in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, unknown, eds. New York, IEEE Computer Society, 2879–86.

Appendix

Appendix A: Outlier analysis

Appendix B.C: Basic dataset

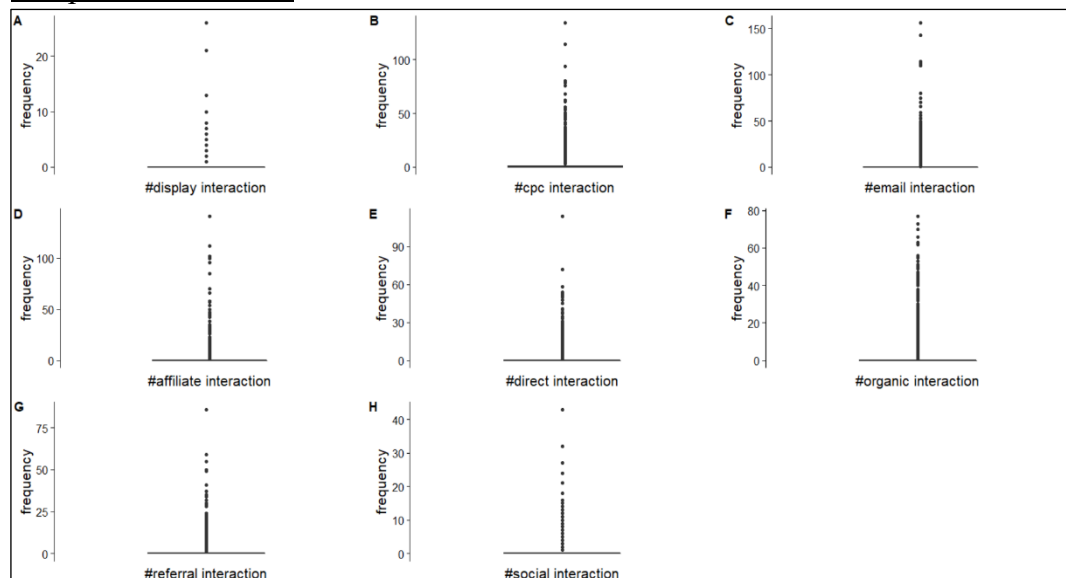
Dependent variables



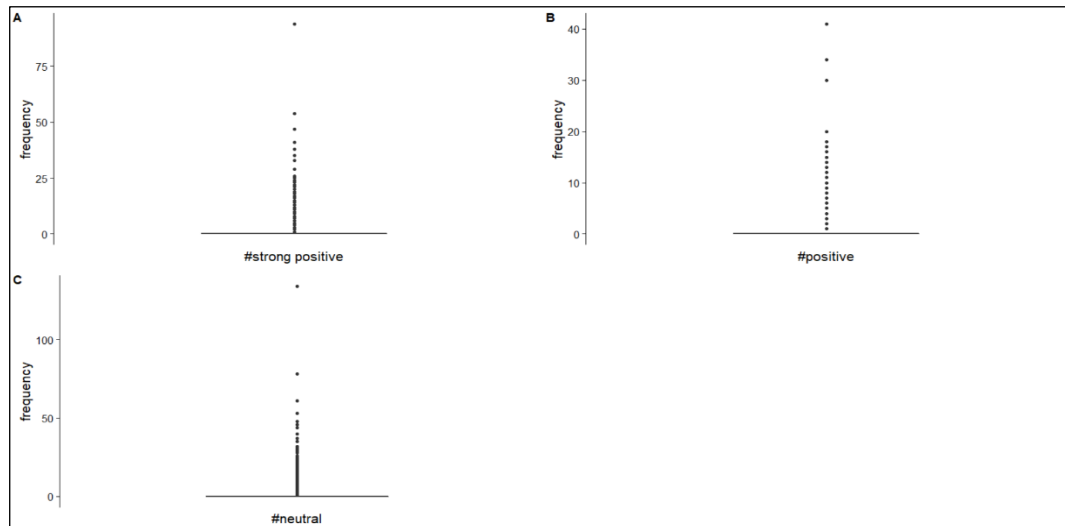
A – product overview page interactions; B – product detail page interactions; C – transactions; D – transaction revenues.

Appendix D.E : Customer journey dataset

Independent variables

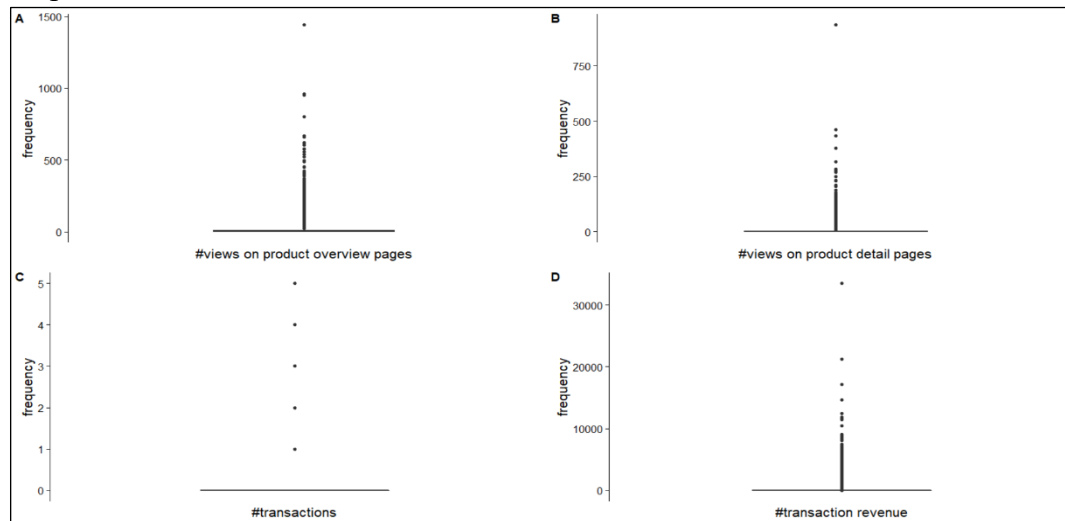


A – display interactions; B – CPC interactions; C – e-mail interactions; D – affiliate interactions; E – direct interactions; F – organic interactions; G – referral interactions; H – social interactions.



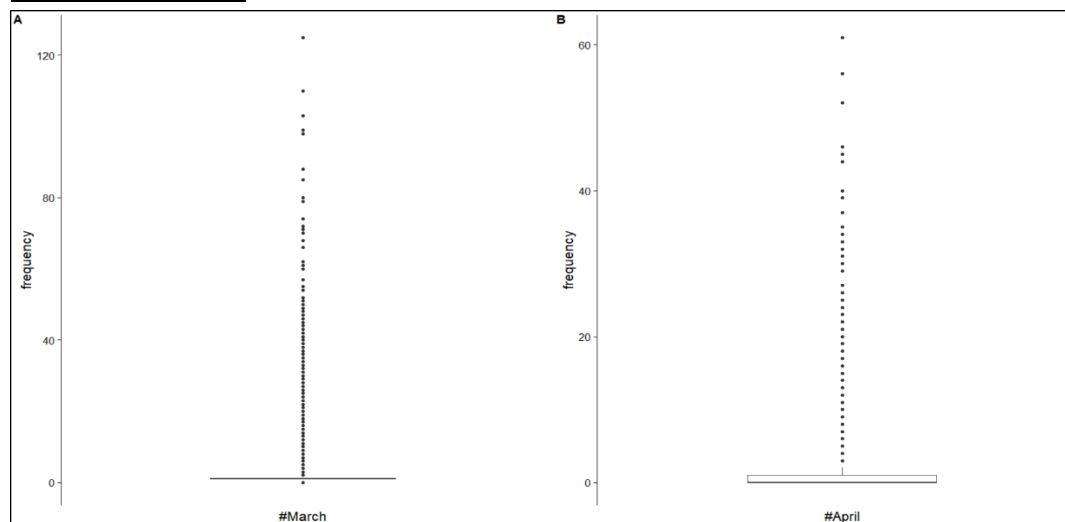
A – strong positive Google Ad interactions; B – positive Google Ad interactions; C – neutral Google Ad interactions.

Dependent variables

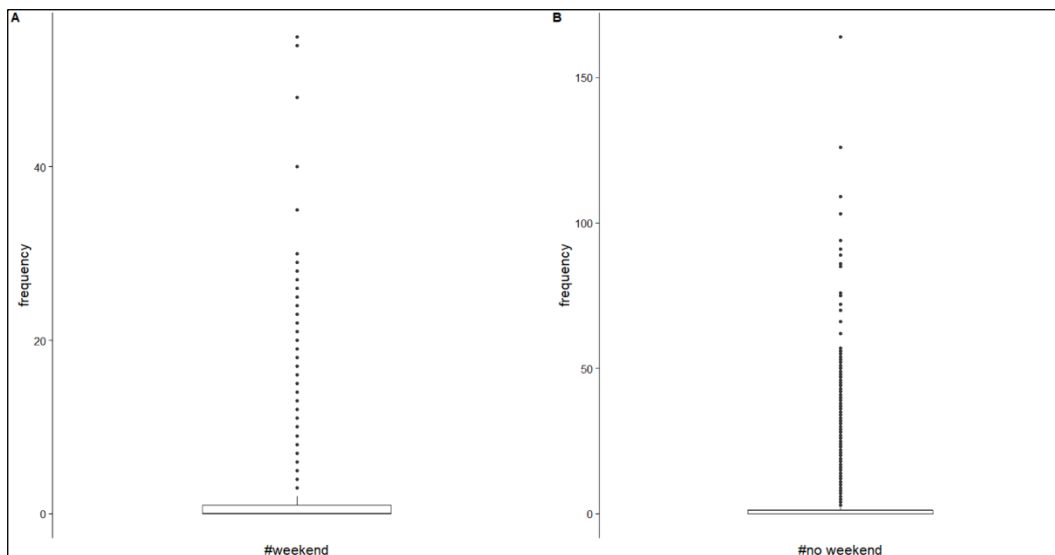


A – product overview page interactions; B – product detail page interactions; C – transactions; D – transaction revenues.

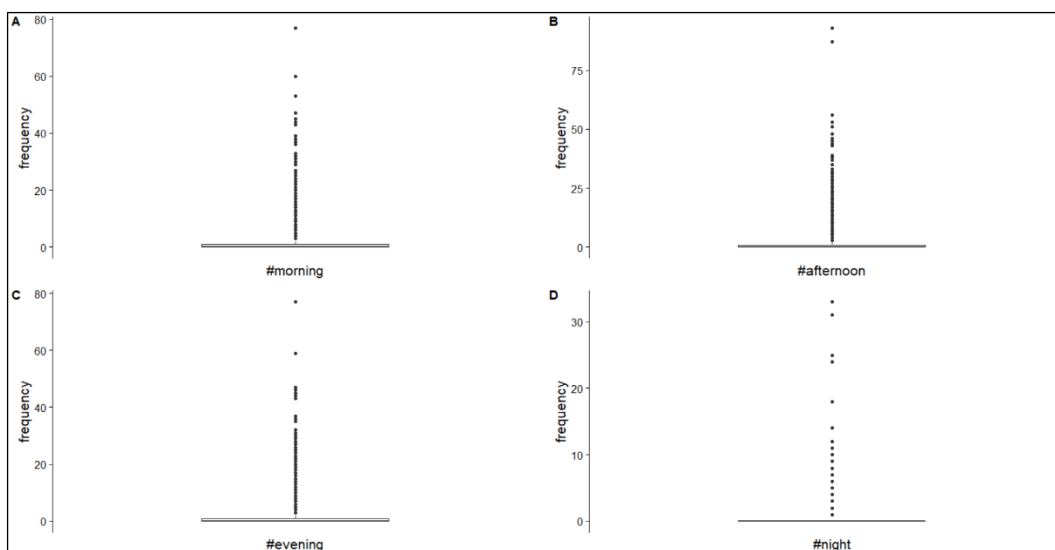
Control variables



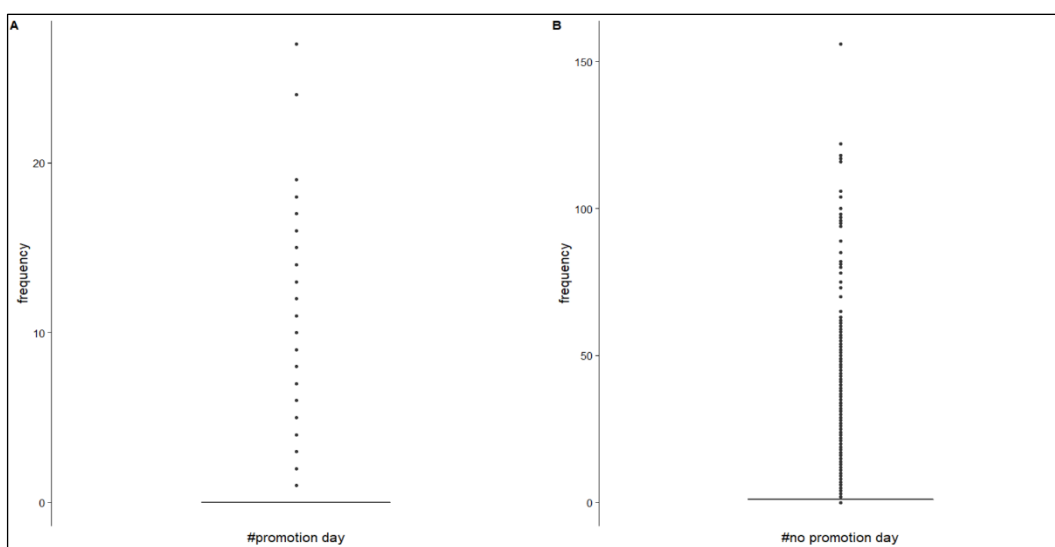
A – March; B – April.



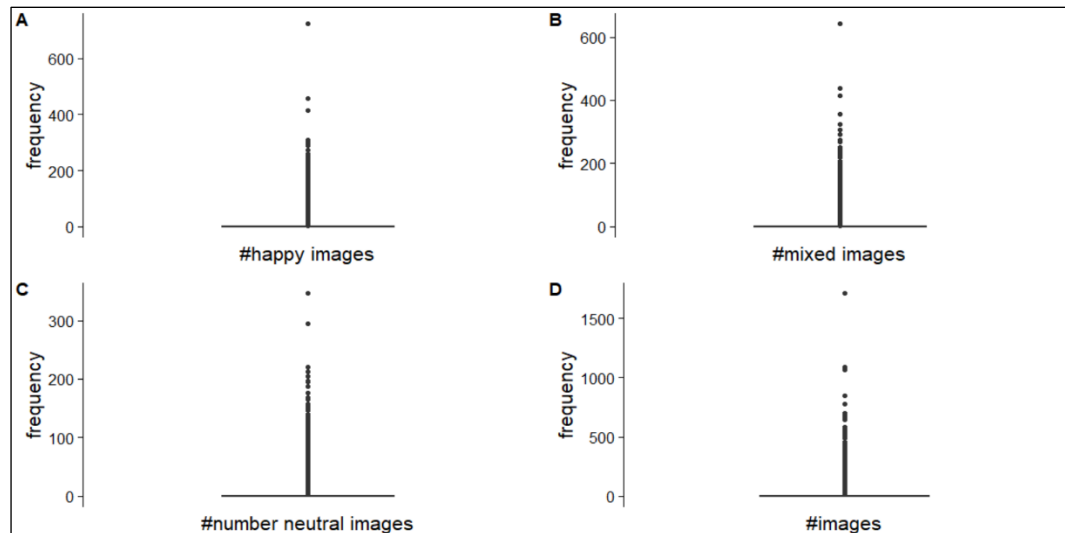
A – weekend; B – no weekend.



A – morning; B – afternoon; C – evening; D – night.



A – promotion day; B – no promotion day.



A – happy image interactions; B – mixed image interactions; C – neutral image interactions; D – image interactions.

Appendix F: Programming code

```
# Intro -----
rm(list = ls())
setwd("C:\\Users\\felix\\Documents\\0.Uni_Groningen_aktuell\\4. Semester\\Master thesis\\Data & Analysis\\Master_thesis_analysis\\FinalData\\Dataset")

## Load packages
library(tidyverse)
library(httr)
library(lubridate)
library(data.table)
library(stringr)
library(fastDummies)
library(Hmisc)

library(scales)
library(countrycode)
library(cowplot)

library(translateR)
library(tidytext)
library(stringr)
library(NLP)
library(tm)
library(qdap)
library(broom)

library(randomForest)
library(caret)
library(MlBayesOpt)
```

```
library(Metrics)
library(ggraph)
library(igraph)
```

```
Sys.setlocale("LC_TIME", "English")
```

Load Clickstream Data -----

```
clickstream_data_1 <- read.csv("dataset1.csv", header = TRUE, stringsAsFactors
= FALSE, na.strings = "", encoding = "UTF-8")
class(clickstream_data_1)
dim(clickstream_data_1)
summary(clickstream_data_1)
str(clickstream_data_1)
colnames(clickstream_data_1)
```

```
clickstream_data_2 <- read.csv("dataset2.csv", header = TRUE, stringsAsFactors
= FALSE, na.strings = "", encoding = "UTF-8")
class(clickstream_data_2)
dim(clickstream_data_2)
summary(clickstream_data_2)
str(clickstream_data_2)
colnames(clickstream_data_2)
```

```
clickstream_data_3 <- read.csv("dataset3.csv", header = TRUE, stringsAsFactors
= FALSE, na.strings = "", encoding = "UTF-8")
view(clickstream_data_3)
class(clickstream_data_3)
dim(clickstream_data_3)
summary(clickstream_data_3)
str(clickstream_data_3)
colnames(clickstream_data_3)
```

Combine Datasets

```
clickstream_data <- rbind(clickstream_data_1,clickstream_data_2,click-
stream_data_3)
view(clickstream_data)
class(clickstream_data)
dim(clickstream_data)
summary(clickstream_data)
str(clickstream_data)
colnames(clickstream_data)
```

Load Google Ad Data -----

```
ad_description_df <- read.csv2("ad_description_df.csv", header = TRUE, string-
sAsFactors = FALSE, na.strings = "", encoding = "UTF-8")
head(ad_description_df)
colnames(ad_description_df) <- "content"
```

```

# Load Image Data -----
image_urls <- read.csv("image_url.csv", header = FALSE, stringsAsFactors =
FALSE, fileEncoding="UTF-8-BOM")
class(image_urls)
colnames(image_urls) <- "url"
image_vector_url <- as.vector(image_urls$url)

# Data Wrangling - Clickstream Data -----
## Naming
column_names <- c("user_id", "session_number", "date_time_utc",
"date_time_cet", "time_of_day", "marketing_channel", "country", "medium",
"source", "campaign", "ad_content", "keyword", "campaign_code", "cam-
paign_id", "ad_group_id", "ad_creative_id", "ad_criteria_id", "ad_page",
"ad_slot", "ad_criteria_parameters", "ad_gcl_id", "ad_customer_id", "ad_net-
work_type", "ad_boom_user_list_id", "ad_video", "ad_type", "ad_approval_sta-
tus", "ad_description", "ad_headlines", "destination_url", "display_url",
"views_product_overview", "views_product_detail", "transactions", "transac-
tion_revenue", "photo_urls", "product_categories", "product_detail_pages")
length(column_names)
colnames(clickstream_data) <- column_names

## Select necessary columns
clickstream_data_selected <- clickstream_data %>% select("user_id", "ses-
sion_number", "date_time_cet", "time_of_day", "marketing_channel", "country",
"ad_description", "views_product_overview", "views_product_detail", "transac-
tions", "transaction_revenue", "photo_urls")
dim(clickstream_data_selected)
str(clickstream_data_selected)

## Replace [NULL] by NA
clickstream_data_selected <- lapply(clickstream_data_selected, function(x)
ifelse(x=="[NULL]", NA, x))
clickstream_data_selected <- as.data.frame(clickstream_data_selected)

## Create multiple channel columns
clickstream_data_selected$marketing_channel <- as.character(click-
stream_data_selected$marketing_channel)
clickstream_data_selected$marketing_channel[clickstream_data_selected$mar-
keting_channel=="(none)"] <- "direct"

clickstream_data_selected$marketing_channel <- as_factor(clickstream_data_se-
lected$marketing_channel)
levels(clickstream_data_selected$marketing_channel)
clickstream_data_selected$marketing_channel <- factor(clickstream_data_se-
lected$marketing_channel, levels = c("display", "cpc", "email", "affiliate", "di-

```

```
rect","organic","referral","social","nontransactionalmail","nontransaction-
almail","remarketing","folder","page","app","banner","web","merk-url","influ-
encermarketing"))
levels(clickstream_data_selected$marketing_channel)
```

```
clickstream_data_selected <- dummy_cols(clickstream_data_selected, select_col-
umns = "marketing_channel")
setnames(clickstream_data_selected, old = c("marketing_channel_email","mar-
keting_channel_social","marketing_channel_display","marketing_channel_affili-
ate","marketing_channel_direct","marketing_channel_organic","market-
ing_channel_referral","marketing_channel_nontransactionalmail","market-
ing_channel_cpc","marketing_channel_remarketing","marketing_chan-
nel_folder","marketing_channel_page","marketing_channel_app","market-
ing_channel_nontransactionalmail","marketing_channel_banner","market-
ing_channel_web","marketing_channel_merk-url","marketing_channel_influenc-
ermarketing"), new=c("email","social","display","affiliate","direct","organic",
"referral","nontransactionalmail","cpc","remarket-
ing","folder","page","app","nontransactionalmail","banner","web","merk-
url","influencermarketing"))
```

Adjust date and time variable

```
clickstream_data_selected <- clickstream_data_selected %>% mu-
tate(date_time_cet_2=date_time_cet)
clickstream_data_selected <- clickstream_data_selected %>% sepa-
rate(date_time_cet_2, c("date","time_cet"), sep = "T")
clickstream_data_selected$time_cet <- substr(clickstream_data_se-
lected$time_cet,0,5)
```

```
clickstream_data_selected$time_of_day <- factor(clickstream_data_se-
lected$time_of_day, levels = c("morning","afternoon","evening","night"))
clickstream_data_selected <- dummy_cols(clickstream_data_selected, select_col-
umns = "time_of_day")
setnames(clickstream_data_selected, old = c("time_of_day_even-
ing","time_of_day_afternoon","time_of_day_morning","time_of_day_night"),
new=c("evening","afternoon","morning","night"))
```

Include weekday and month

```
clickstream_data_selected$month <- lubridate::month(clickstream_data_se-
lected$date_time_cet,abbr=FALSE, label=TRUE)
clickstream_data_selected$weekday <- lubridate::wday(clickstream_data_se-
lected$date_time_cet, abbr=FALSE, label=TRUE)
clickstream_data_selected$date_time_cet <- NULL
```

Months

```
clickstream_data_selected <- dummy_cols(clickstream_data_selected, select_col-
umns = "month")
setnames(clickstream_data_selected, old = c("month_March", "month_April"),
new=c("March", "April"))
```

Weekday vs. Weekend

```
clickstream_data_selected$type_of_week <- ifelse(clickstream_data_selected$weekday %in% c("Saturday", "Sunday"), "weekend", "no_weekend")
clickstream_data_selected <- dummy_cols(clickstream_data_selected, select_columns = "type_of_week")
setnames(clickstream_data_selected, old = c("type_of_week_weekend", "type_of_week_no_weekend"), new = c("weekend", "no_weekend"))
```

Website familiarity

```
clickstream_data_selected$familiarity <- ifelse(clickstream_data_selected$session_number == 1, "not_familiar", "familiar")
clickstream_data_selected <- dummy_cols(clickstream_data_selected, select_columns = "familiarity")
setnames(clickstream_data_selected, old = c("familiarity_familiar", "familiarity_not_familiar"), new = c("familiar", "not_familiar"))
```

Continents

```
unique(clickstream_data_selected$country)
length(clickstream_data_selected[clickstream_data_selected$country == "(not set)", "user_id"])
```

```
clickstream_data_selected$continent <- countrycode(sourcevar = clickstream_data_selected$country, origin = "country.name", destination = "continent")
```

```
length(clickstream_data_selected[clickstream_data_selected$continent == "NA", "user_id"])
length(clickstream_data_selected[clickstream_data_selected$country == "(not set)", "user_id"])
length(clickstream_data_selected[clickstream_data_selected$country == "Kosovo", "user_id"])
length(clickstream_data_selected[clickstream_data_selected$country == "St. Martin", "user_id"])
```

```
unique(clickstream_data_selected$continent)
clickstream_data_selected[clickstream_data_selected$country == "Kosovo", "continent"] <- c("Europe", "Europe", "Europe", "Europe")
clickstream_data_selected[clickstream_data_selected$country == "St. Martin", "continent"] <- c("Americas")
```

```
clickstream_data_selected <- dummy_cols(clickstream_data_selected, select_columns = "continent")
setnames(clickstream_data_selected, old = c("continent_Europe", "continent_Asia", "continent_Americas", "continent_Africa", "continent_Oceania", "continent_NA"), new = c("Europe", "Asia", "Americas", "Africa", "Oceania", "NA"))
```

Include promotion column

```
clickstream_data_selected$promotion_day_overview <- 0
```

```
clickstream_data_selected$promotion_day_overview[clickstream_data_selected$date %in% c("2019-03-22", "2019-03-23", "2019-03-24", "2019-04-06")]
<- 1
```

```
clickstream_data_selected <- dummy_cols(clickstream_data_selected, select_columns = "promotion_day_overview")
setnames(clickstream_data_selected, old = c("promotion_day_overview_1", "promotion_day_overview_0"), new = c("promotion_day", "no_promotion_day"))
```

```
## Change data types
str(clickstream_data_selected)
```

```
clickstream_data_selected$ad_description <- as.character(clickstream_data_selected$ad_description)
clickstream_data_selected$photo_urls <- as.character(clickstream_data_selected$photo_urls)
```

```
clickstream_data_selected$date <- as_date(clickstream_data_selected$date)
clickstream_data_selected$time_cet <- as.ITime(clickstream_data_selected$time_cet)
```

```
levels(clickstream_data_selected$month) <- c("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")
clickstream_data_selected$weekday <- factor(clickstream_data_selected$weekday, levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))
```

```
levels(clickstream_data_selected$weekday) <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
clickstream_data_selected$type_of_week <- as.factor(clickstream_data_selected$type_of_week)
```

```
clickstream_data_selected$familiarity <- as.factor(clickstream_data_selected$familiarity)
```

```
clickstream_data_selected$country <- as.factor(clickstream_data_selected$country)
clickstream_data_selected$continent <- as.factor(clickstream_data_selected$continent)
```

```
clickstream_data_selected$promotion_day_overview <- as.integer(clickstream_data_selected$promotion_day_overview)
```

```
str(clickstream_data_selected)
```

```
## Order columns for better overview
```

```
clickstream_data_selected <- clickstream_data_selected[c("date", "user_id",  
"views_product_overview", "views_product_detail", "transactions", "transac-  
tion_revenue", "marketing_channel", "display", "cpc", "email", "affiliate", "di-  
rect", "organic", "referral", "social", "nontransactionalmail", "remarketing",  
"folder", "page", "app", "nontransactionalmail", "banner", "web", "merk-url", "in-  
fluencemarketing", "ad_description", "country", "continent", "Europe", "Asia",  
"Americas", "Africa", "Oceania", "month", "March", "April", "type_of_week",  
"weekend", "no_weekend", "weekday", "time_of_day", "morning", "afternoon",  
"evening", "night", "time_cet", "session_number", "familiarity", "familiar",  
"not_familiar", "promotion_day_overview", "promotion_day", "no_promo-  
tion_day", "photo_urls")]  
str(clickstream_data_selected)
```

```
## Check NAs
```

```
col_clickstream_data_selected <- colnames(clickstream_data_selected)  
number_NAs_clickstream_data_selected <- vector(mode = "double", length =  
length(col_clickstream_data_selected))  
names(number_NAs_clickstream_data_selected) <- col_clickstream_data_se-  
lected
```

```
for (i in seq_along(col_clickstream_data_selected)) {  
  number_NAs_clickstream_data_selected[i] <- sum(is.na(clickstream_data_se-  
lected[col_clickstream_data_selected[i]]))  
}
```

```
number_NAs_clickstream_data_selected #(transaction_revenue: change later to 0,  
ad_descriptions: change later to "", continent: due to missing country informations  
(no problem in final datasets - due to grouping and filtering))
```

```
## Check missing values
```

```
number_missing_clickstream_data_selected <- vector(mode = "double", length =  
length(col_clickstream_data_selected))  
names(number_missing_clickstream_data_selected) <- col_clickstream_data_se-  
lected
```

```
for (i in seq_along(number_missing_clickstream_data_selected)) {  
  number_missing_clickstream_data_selected[i] <- sum(is_empty(click-  
stream_data_selected[col_clickstream_data_selected[i]]))  
}
```

```
number_missing_clickstream_data_selected
```

```
## Delete "errors" (views_product_overview=0)
```

```
clickstream_data_selected <- clickstream_data_selected %>% filter(views_pro-  
duct_overview!=0)
```

```

## Check outliers
#"views_product_overview", "views_product_detail", "transactions", "transaction_revenue"
a <- c("views_product_overview", "views_product_detail", "transactions", "transaction_revenue")
b <- c("views on product overview pages", "views on product detail pages", "transactions", "transaction revenue")

independent_variables <- c("display", "cpc", "email", "affiliate", "direct", "organic", "referral", "social")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(clickstream_data_selected[clickstream_data_selected$marketing_channel %in% independent_variables,], aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

boxplot_vpo_csd <- plot_list[[1]]
boxplot_vpd_csd <- plot_list[[2]]
boxplot_tra_csd <- plot_list[[3]]
boxplot_tre_csd <- plot_list[[4]]

plot_grid(boxplot_vpo_csd, boxplot_vpd_csd, boxplot_tra_csd, boxplot_tre_csd,
labels = "AUTO")

## In-depth check for outliers
clickstream_data_selected %>% filter(marketing_channel %in% independent_variables & views_product_overview>150)
clickstream_data_selected %>% filter(marketing_channel %in% independent_variables & views_product_detail>125)
clickstream_data_selected %>% filter(marketing_channel %in% independent_variables & transaction_revenue>20000)

## Save
write.csv(clickstream_data_selected, file = "C:\\Users\\felix\\Documents\\0.Uni_Groningen_aktuell\\4. Semester\\Master thesis\\Data & Analysis\\Master_thesis_analysis\\FinalData\\Dataset\\clickstream_data_selected.csv")
view(clickstream_data_selected)

# Data Wrangling - Google Ad Disruption Data -----
## Create unique ad description dataset
ad_description_df <- unique(ad_description_df$content)
ad_description_df <- as.data.frame(ad_description_df)

```



```

colnames(ad_description_df) <- "content"
ad_description_df$content <- as.character(ad_description_df$content)

## Translate ad description from dutch to english using google cloud translation
API
translated_ad_description <- translate(dataset=ad_description_df,
                                       content.field="content",
                                       google.api.key="AIzaSyAV1w8ueLAkw-
Xg1D2B4JKj6wajMDOWES4",
                                       source.lang="nl",
                                       target.lang="en")

head(translated_ad_description)
colnames(translated_ad_description) <- c("dutch","english")

## Save
write.csv(translated_ad_description, file = "C:\\Users\\felix\\Docu-
ments\\0.Uni_Groningen_aktuell\\4. Semester\\Master thesis\\Data & Analy-
sis\\Master_thesis_analysis\\FinalData\\Dataset\\ad_description.csv")

## Load translated ad descriptions
translated_ad_description <- read.csv("ad_description.csv", header = TRUE,
stringsAsFactors = FALSE, na.strings = "")

## Create dataframe for english version
ad_description_english <- as.data.frame(translated_ad_description$english)
head(ad_description_english)
colnames(ad_description_english) <- "english"
ad_description_english$english <- as.character(ad_description_english$english)

## Correct translation errors
...

## Polarity analysis
description_polarity <- polarity(text.var = ad_description_english$english,
                                grouping.var = ad_description_english$english,
                                polarity.frame = key.pol[x!=c("cheap"),],
                                negators = negation.words,
                                amplifiers = amplification.words,
                                deamplifiers = deamplification.words)

polarity_analysis <- counts(description_polarity)
view(polarity_analysis)

#Correcting errors
polarity_analysis$polarity <- ifelse(polarity_analysis$polarity<0,(-1)*polar-
ity_analysis$polarity,polarity_analysis$polarity)

#Classification

```

```

polarity_analysis$strong_positive <- if_else(polarity_analysis$polarity>=0.8,1,0)
polarity_analysis$positive <- if_else(polarity_analysis$polarity>=0.5 & polarity_analysis$polarity<0.8,1,0)
polarity_analysis$neutral <- if_else(polarity_analysis$polarity<0.5,1,0)

## Check NAs
col_polarity_analysis <- colnames(polarity_analysis[7:9])
number_NAs_polarity_analysis <- vector(mode = "double", length =
length(col_polarity_analysis))
names(number_NAs_polarity_analysis) <- col_polarity_analysis

for (i in seq_along(col_polarity_analysis)) {
  number_NAs_polarity_analysis[i] <- sum(is.na(polarity_analysis[col_polarity_analysis[i]]))
}

number_NAs_polarity_analysis

## Check missing values
number_missing_polarity_analysis <- vector(mode = "double", length =
length(col_polarity_analysis))
names(number_missing_polarity_analysis) <- col_polarity_analysis

for (i in seq_along(number_missing_polarity_analysis)) {
  number_missing_polarity_analysis[i] <- sum(is_empty(polarity_analysis[col_polarity_analysis[i]]))
}

number_missing_polarity_analysis

#Create polarity vectors
ad_description_df$strong_positive <- polarity_analysis$strong_positive
ad_description_df$positive <- polarity_analysis$positive
ad_description_df$neutral <- polarity_analysis$neutral

strong_positive_description <- as.vector(ad_description_df$content[ad_description_df$strong_positive==1])
positive_description <- as.vector(ad_description_df$content[ad_description_df$positive==1])
neutral_description <- as.vector(ad_description_df$content[ad_description_df$neutral==1])

# Data Wrangling - Image Data -----
## Define API
end.point <- "https://westeurope.api.cognitive.microsoft.com/face/v1.0/detect"

## Define API_KEY
key1 <- "cc0ca543bb434ecd9e8d0f95cb8914cb"

```

```

## Access API
image_info <- list()

for (i in seq_along(image_vector_url)) {
  image <- paste("img",i,sep = "_")
  if(i %% 20 == 0){Sys.sleep(61)}
  image_info[[i]] <- as.list(assign(image,POST(url = end.point,
                                         add_headers(.headers = c("Ocp-Apim-Subscription-
Key" = key1)),
                                body = paste('{ "url":', "'",
                                image_vector_url[i], "'", '}', sep = "'"),
                                query = list(returnFaceAttributes = "emotion"),
                                accept_json()))))
}

## Extract emotions
extract_emotions <- function(x){
  as.data.frame((content(x)[[1]]$faceAttributes$emotion))
}

image_emotion_df <- data.frame(Anger=numeric(),Contempt=numeric(),Dis-
gust=numeric(),Fear=numeric(),Happiness=numeric(),Neutral=numeric(),Sad-
ness=numeric(),Surprise=numeric())

for (i in c(1:length(image_vector_url))) {
  tryCatch({
    image_emotion_df[i,] <- extract_emotions(image_info[[i]])
  }, error=function(e){cat("ERROR :", "NA", "\n")})
}

print(image_emotion_df)
class(image_emotion_df)
dim(image_emotion_df)
str(image_emotion_df)
## Add image_urls as first column
image_emotion_df$url <- image_urls$url
image_emotion_df <- image_emotion_df[,c("url", "Anger", "Contempt", "Fear",
"Happiness", "Neutral", "Sadness", "Surprise")]

## Create evaluation columns
image_emotion_df$Strong_Happy <- if_else(image_emotion_df$Happi-
ness>0.8,1,0)
image_emotion_df$Strong_Neutral <- if_else(image_emotion_df$Neu-
tral>0.8,1,0)
image_emotion_df$Mixed <- if_else(image_emotion_df$Neutral<=0.8 & im-
age_emotion_df$Happiness<=0.8,1,0)

```

```

image_emotion_df$Else <- if_else(image_emotion_df$Strong_Happy==0 & im-
age_emotion_df$Strong_Neutral==0 & image_emotion_df$Mixed==0,1,0)
image_emotion_df$No_Face <- if_else(is.na(image_emotion_df$Anger),1,0)

view(image_emotion_df)

## Save
write.csv(image_emotion_df, file = "C:\\Users\\felix\\Documents\\0.Uni_Gro-
ningen_aktuell\\4. Semester\\Master thesis\\Data & Analysis\\Master_thesis_anal-
ysis\\FinalData\\Dataset\\evaluated_image_data.csv")

## Load evaluated Image Data
evaluated_image_data <- read.csv("evaluated_image_data.csv", header = TRUE,
stringsAsFactors = FALSE, fileEncoding="UTF-8-BOM")
class(evaluated_image_data)
dim(evaluated_image_data)
str(evaluated_image_data)

## Check NAs
col_evaluated_image_data <- colnames(evaluated_image_data)
number_NAs_evaluated_image_data <- vector(mode = "double", length =
length(col_evaluated_image_data))
names(number_NAs_evaluated_image_data) <- col_evaluated_image_data

for (i in seq_along(col_evaluated_image_data)) {
  number_NAs_evaluated_image_data[i] <- sum(is.na(evaluated_im-
age_data[col_evaluated_image_data[i]]))
}

number_NAs_evaluated_image_data #(NAs due to pictures without model face
shown)
nrow(evaluated_image_data[evaluated_image_data$No_Face==1,])

## Check missing values
number_missing_evaluated_image_data <- vector(mode = "double", length =
length(col_evaluated_image_data))
names(number_missing_evaluated_image_data) <- col_evaluated_image_data

for (i in seq_along(number_missing_evaluated_image_data)) {
  number_missing_evaluated_image_data[i] <- sum(is_empty(evaluated_im-
age_data[col_evaluated_image_data[i]]))
}

number_missing_evaluated_image_data

## Create image-emotion vectors
happy_images <- evaluated_image_data %>% select(url,Strong_Happy) %>% fil-
ter(Strong_Happy==1) %>% select(url)

```

```

mixed_images <- evaluated_image_data %>% select(url,Strong_Neutral) %>%
filter(Strong_Neutral==1) %>% select(url)
neutral_images <- evaluated_image_data %>% select(url,Mixed) %>% fil-
ter(Mixed==1) %>% select(url)
na_images <- evaluated_image_data %>% select(url,No_Face) %>% fil-
ter(No_Face==1) %>% select(url)

happy_images_url <- as.vector(happy_images$url)
mixed_images_url <- as.vector(mixed_images$url)
neutral_images_url <- as.vector(neutral_images$url)

# Exploratory Data Analysis - Clickstream Data -----
summary(clickstream_data_selected)
str(clickstream_data_selected)

## Independent variables
## How often marketing channel was used to access website (absolute values)
#(a)Absolute marketing channel interaction in total
number_absolute <- vector(mode = "double", length = length(independent_varia-
bles))
names(number_absolute) <- independent_variables

for(i in seq_along(independent_variables)){
  number_absolute[i] <- sum(clickstream_data_selected[,independent_varia-
bles[i]])
}

absolute_frequency_a <- number_absolute
absolute_frequency_a

#(a)Absolute marketing channel interaction in total
bar_absolute_channel_interaction <- clickstream_data_selected %>%
select(marketing_channel) %>%
filter(marketing_channel %in% independent_variables) %>%
ggplot(aes(x=marketing_channel))+
geom_bar()+
xlab("marketing channel")+
scale_y_continuous(name = "#marketing channel interaction", la-
bels=scales::comma)+
theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"))+
labs(fill="marketing channel")

print(bar_absolute_channel_interaction)

#(b)Absolute marketing channel interaction per day
absolute_frequency_b <- clickstream_data_selected %>%
select(date, marketing_channel) %>%
filter(marketing_channel %in% independent_variables) %>%

```

```
group_by(date,marketing_channel) %>%
summarise(absolute_frequency=n())
```

#(b)Absolute marketing channel interaction per day

```
point_absolute_channel_interaction_daily <- ggplot(absolute_frequency_b,
aes(x=date, y=absolute_frequency, color=marketing_channel))+
  geom_point()+
  geom_line()+
  scale_y_continuous(name = "#marketing channel interactions", la-
bels=scales::comma)+
  theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"), leg-
end.justification = "top")+
  labs(col="marketing channel")
```

```
print(point_absolute_channel_interaction_daily)
```

How often marketing channel was used to access website (relative values)

#(c)Relative marketing channel interaction in total

```
number_relative <- vector(mode = "double", length = length(independent_varia-
bles))
names(number_relative) <- independent_variables
```

```
for(i in seq_along(independent_variables)){
  number_relative[i] <- sum(clickstream_data_selected[,independent varia-
bles[i]]) / nrow(clickstream_data_selected%>% filter(marketing_channel %in%
independent_variables))
}
```

```
relative_frequency_a <- clickstream_data_selected %>%
  select(marketing_channel) %>%
  filter(marketing_channel %in% independent_variables) %>%
  count(marketing_channel) %>%
  mutate(relative_frequency=n/nrow(clickstream_data_selected %>% filter(mar-
keting_channel %in% independent_variables)))
relative_frequency_a
```

#(c)Relative marketing channel interaction in total

```
bar_relative_channel_interaction <- ggplot(relative_frequency_a, aes(x=market-
ing_channel,y=relative_frequency))+
  geom_bar(stat = "identity")+
  ylab("marketing channel interaction (%)")+
  xlab("marketing channel")+
  theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"))+
  labs(fill="marketing channel")
```

```
print(bar_relative_channel_interaction)
```

#(d)Relative marketing channel interaction per day

```
relative_frequency_b <- clickstream_data_selected %>%  
  select(date, marketing_channel) %>%  
  filter(marketing_channel %in% independent_variables) %>%  
  group_by(date,marketing_channel) %>%  
  summarise(absolute_frequency=n()) %>%  
  mutate(channel_frequency=sum(absolute_frequency),relative_frequency=absolute_frequency/channel_frequency)
```

#(d)Relative marketing channel interaction per day

```
point_relative_channel_interaction_daily <- ggplot(relative_frequency_b,  
aes(x=date, y=relative_frequency, color=marketing_channel))+  
  geom_point()+  
  geom_line()+  
  ylab("marketing channel interaction (%)")+  
  theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"), legend.justification = "top")+  
  labs(col="marketing channel")
```

```
print(point_relative_channel_interaction_daily)
```

#Overview IV graphics

```
plot_grid(bar_absolute_channel_interaction, bar_relative_channel_interaction, labels = "AUTO")  
plot_grid(point_absolute_channel_interaction_daily, point_relative_channel_interaction_daily, labels = "AUTO")
```

Dependent variables

```
dependent_variables <- c("views_product_overview", "views_product_detail",  
"transactions", "transaction_revenue")
```

```
clickstream_data_selected$transaction_revenue[is.na(clickstream_data_selected$transaction_revenue)] <- 0
```

How often product overview pages, product detail pages have been viewed + number of transactions

```
number_absolute_DV <- vector(mode = "double", length = length(dependent_variables))  
names(number_absolute_DV) <- dependent_variables
```

```
for(i in seq_along(dependent_variables)){  
  number_absolute_DV[i] <- sum(clickstream_data_selected[clickstream_data_selected$marketing_channel %in% independent_variables,dependent_variables[i]])  
}
```

```
absolute_frequency_DV_a <- number_absolute_DV  
absolute_frequency_DV_a <- data.frame(views_transactions=dependent_variables, absolute_frequency=absolute_frequency_DV_a)
```

```

str(absolute_frequency_DV_a)
absolute_frequency_DV_a$views_transactions <- factor(absolute_frequency_DV_a$views_transactions, levels = c("views_product_overview", "views_product_detail", "transactions", "transaction_revenue"))

absolute_views <- absolute_frequency_DV_a %>% select(views=views_transactions, absolute_frequency) %>%
  filter(views %in% c("views_product_overview", "views_product_detail"))

absolute_transactions <- absolute_frequency_DV_a %>% select(transactions=views_transactions, absolute_frequency) %>%
  filter(transactions=="transactions")

absolute_transaction_revenue <- absolute_frequency_DV_a %>% select(transaction_revenue=views_transactions, absolute_volume=absolute_frequency) %>%
  filter(transaction_revenue=="transaction_revenue")

#(a)Absolute views in total
views_total <- ggplot(absolute_views, aes(x=views, y=absolute_frequency))+
  geom_bar(stat="identity")+
  scale_y_continuous(name = "#views", labels=scales::comma)+
  scale_x_discrete(name="purchase funnel stages",labels=c("product overview page", "product detail page"))+
  theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))

print(views_total)

#(b)Absolute transactions in total
transactions_total <- ggplot(absolute_transactions, aes(x=transactions, y=absolute_frequency))+
  geom_bar(stat="identity")+
  scale_y_continuous(name = "#views", labels=scales::comma)+
  scale_x_discrete(name="purchase funnel",labels=c("transactions"))+
  theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
print(transactions_total)

#(c)Absolute transaction volume in total
transaction_volume_total <- ggplot(absolute_transaction_revenue, aes(x=transaction_revenue, y=absolute_volume))+
  geom_bar(stat="identity")+
  scale_y_continuous(name = "#volume", labels=scales::comma)+
  scale_x_discrete(name="purchase funnel",labels=c("transaction_revenue"))+
  theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))

print(transaction_volume_total)

```



```

#(a)Absolute views, (b)transactions, and (c) transaction_revenue
plot_grid(views_total, transactions_total, transaction_volume_total, labels =
"AUTO")

# Views, transactions, and transaction revenue per day
absolute_frequency_DV_b <- clickstream_data_selected %>%
  select(date, marketing_channel, views_product_overview, views_product_detail,
transactions, transaction_revenue) %>%
  filter(marketing_channel %in% independent_variables) %>%
  group_by(date) %>%
  summarise(views_product_overview=sum(views_product_overview),
views_product_detail=sum(views_product_detail), transactions=sum(transac-
tions), transaction_revenue=sum(transaction_revenue))

absolute_frequency_DV_b_long <- gather(absolute_frequency_DV_b,
"views_transactions_transaction_revenue", "frequency", 2:5)
absolute_frequency_DV_b_long$views_transactions_transaction_revenue <- fac-
tor(absolute_frequency_DV_b_long$views_transactions_transaction_revenue,
levels = c("views_product_overview", "views_product_detail", "transactions",
"transaction_revenue"))

#(d)Absolute views product overview per day
absolute_views_product_overview_per_day <- absolute_frequency_DV_b_long
%>% select(date, views=views_transactions_transaction_revenue, frequency)
%>% filter(views == "views_product_overview")

#(d)Absolute views product overview per day
views_product_overview_per_day <- ggplot(absolute_views_product_over-
view_per_day, aes(x=date, y=frequency))+
  geom_point()+
  geom_line()+
  scale_y_continuous(name = "#product overview page interactions", la-
bels=scales::comma)+
  theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"), leg-
end.position = "none")
print(views_product_overview_per_day)

#(e)Absolute views product detail per day
absolute_views_product_detail_per_day <- absolute_frequency_DV_b_long
%>% select(date, views=views_transactions_transaction_revenue, frequency)
%>%
  filter(views == "views_product_detail")

#(e)Absolute views product overview per day
views_product_detail_per_day <- ggplot(absolute_views_product_detail_per_day,
aes(x=date, y=frequency))+
  geom_point()+

```

```

geom_line()+
scale_y_continuous(name = "#product detail page interactions", la-
bels=scales::comma)+
theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"), leg-
end.position = "none")

```

```

print(views_product_detail_per_day)

```

#(f)Absolute transactions per day

```

absolute_transactions_per_day <- absolute_frequency_DV_b_long %>% se-
lect(date, transactions=views_transactions_transaction_revenue, frequency) %>%
filter(transactions=="transactions")

```

#(f)Absolute transactions per day

```

transactions_per_day <- ggplot(absolute_transactions_per_day, aes(x=date, y=fre-
quency))+
geom_point()+
geom_line()+
scale_y_continuous(name = "#transactions", labels=scales::comma)+
theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"), leg-
end.position = "none")+
labs(col="marketing channel")

```

```

print(transactions_per_day)

```

#(g)Absolute transaction revenue per day

```

absolute_transaction_revenue_per_day <- absolute_frequency_DV_b_long %>%
select(date, transaction_revenue=views_transactions_transaction_revenue, fre-
quency) %>%
filter(transaction_revenue=="transaction_revenue")

```

#(g)Absolute transaction revenue per day

```

transaction_revenue_per_day <- ggplot(absolute_transaction_revenue_per_day,
aes(x=date, y=frequency))+
geom_point()+
geom_line()+
scale_y_continuous(name = "#transaction revenue", labels=scales::comma)+
theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"), leg-
end.position = "none")+
labs(col="marketing channel")

```

```

print(transaction_revenue_per_day)

```

#(d)Views product overview (e)Views product detail and (f)transactions, and (g) transaction revenue per day

```
plot_grid(views_product_overview_per_day,views_product_detail_per_day,
transactions_per_day, transaction_revenue_per_day, labels = "AUTO")
```

```
## Controls
```

```
##(a)Continent
```

```
continent_number <- clickstream_data_selected %>%
  filter(marketing_channel %in% independent_variables) %>%
  select(continent) %>%
  group_by(continent) %>%
  count(continent)
```

```
bar_continent_number <- ggplot(data=continent_number)+
  geom_bar(aes(x=continent, y=n), stat="identity")+
  scale_y_continuous(name = "frequency", labels=scales::comma)+
  theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
```

```
print(bar_continent_number)
```

```
##(b)Month
```

```
month_number <- clickstream_data_selected %>%
  filter(marketing_channel %in% independent_variables) %>%
  select(month) %>%
  group_by(month) %>%
  count(month)
```

```
bar_month_number <- ggplot(data=month_number)+
  geom_bar(aes(x=month, y=n), stat="identity")+
  scale_y_continuous(name = "frequency", labels=scales::comma)+
  theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
```

```
print(bar_month_number)
```

```
##(c)Time of week
```

```
weekend_number <- clickstream_data_selected %>%
  filter(marketing_channel %in% independent_variables) %>%
  select(type_of_week) %>%
  group_by(type_of_week) %>%
  count(type_of_week)
```

```
bar_weekend_number <- ggplot(data=weekend_number)+
  geom_bar(aes(x=type_of_week, y=n), stat="identity")+
  scale_x_discrete(name="type of week",labels=c("no weekend", "weekend"))+
  scale_y_continuous(name = "frequency", labels=scales::comma)+
  theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
```

```
print(bar_weekend_number)
```

#(d)Time of day

```
time_of_day_number <- clickstream_data_selected %>%
  filter(marketing_channel %in% independent_variables) %>%
  select(time_of_day) %>%
  group_by(time_of_day) %>%
  count(time_of_day)

bar_time_of_day_number <- ggplot(data=time_of_day_number)+
  geom_bar(aes(x=time_of_day, y=n), stat="identity")+
  xlab("time of day")+
  scale_y_continuous(name = "frequency", labels=scales::comma)+
  theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))

print(bar_time_of_day_number)
```

#(e)Familiarity

```
familiarity_number <- clickstream_data_selected %>%
  filter(marketing_channel %in% independent_variables) %>%
  select(familiarity) %>%
  group_by(familiarity) %>%
  count(familiarity)

bar_familiarity_number <- ggplot(data=familiarity_number)+
  geom_bar(aes(reorder(x=familiarity,n), y=n), stat="identity")+
  scale_x_discrete(name = "familiarity")+
  scale_y_continuous(name = "frequency", labels=scales::comma)+
  theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))

print(bar_familiarity_number)
```

#(f)Promotion day

```
promotion_day_number <- clickstream_data_selected %>%
  filter(marketing_channel %in% independent_variables) %>%
  select(promotion_day_overview) %>%
  group_by(promotion_day_overview) %>%
  count(promotion_day_overview)

promotion_day_number <- as.data.frame(promotion_day_number)
colnames(promotion_day_number) <- c("promotion_status", "n")
promotion_day_number[promotion_day_number$promotion_status==0,"promotion_status"] <- "no promotion day"
promotion_day_number[promotion_day_number$promotion_status==1,"promotion_status"] <- "promotion day"

bar_promotion_day_number <- ggplot(data=promotion_day_number)+
  geom_bar(aes(x=promotion_status, y=n), stat="identity")+
  xlab("promotion status")+
  scale_y_continuous(name = "frequency", labels=scales::comma)+
  theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))

print(bar_promotion_day_number)
```

```

scale_x_discrete(name="promotion status")+
scale_y_continuous(name = "frequency", labels=scales::comma)+
theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"))

print(bar_promotion_day_number)

#Overview control graphics
plot_grid(bar_continent_number, bar_month_number, bar_weekend_number,
bar_time_of_day_number, bar_familiarity_number, bar_promotion_day_number,
labels = "AUTO")

# Exploratory Data Analysis Google Ad Discription Data -----
## On Ad level
#Sessions with Google Ad descriptions
nrow(clickstream_data_selected[!is.na(clickstream_data_selected$ad_descrip-
tion),])

#Sessions accessed via cpc
nrow(clickstream_data_selected[clickstream_data_selected$cpc==1,])

#Unique Google Ad descriptions
length(ad_description_df$content)

#Most commonn Google Ad descriptions
ad_description_vector <- as.vector(clickstream_data_selected[!is.na(click-
stream_data_selected$ad_description),"ad_description"])
number_ad_descriptions <- table(ad_description_vector)
number_ad_descriptions <- sort(number_ad_descriptions)
view(number_ad_descriptions)

## On word level
#Single word usage
data(stop_words)
single_words <- ad_description_english %>%
  mutate(id=row_number()) %>%
  unnest_tokens(word, english) %>%
  anti_join(stop_words)
single_words %>% count(word, sort = TRUE)

single_words %>%
  count(word, sort = TRUE) %>%
  filter(n>10) %>%
  mutate(word=reorder(word,n)) %>%
  ggplot(aes(word,n))+
  geom_col()+
  xlab("words")+
  coord_flip()+

```

```
theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
```

```
#Polarity ad descriptions
```

```
polarity <- c("strong_positive", "positive", "neutral")
number_polarity <- vector(mode = "double", length = length(polarity))
names(number_polarity) <- polarity
```

```
for (i in seq_along(polarity_analysis$english)) {
  number_polarity[i] <- polarity_analysis %>% select(polarity[i]) %>% sum()
}
```

```
number_polarity <- as.data.frame(number_polarity)
colnames(number_polarity) <- "frequency"
```

```
polarity_unique_ads <- ggplot(number_polarity, aes(x=reorder(rownames(number_polarity), frequency), y=frequency))+
  geom_bar(stat = "identity")+
  theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))+
  scale_x_discrete(name="emotionality", labels=c("strong positive", "positive", "neutral"))+
  ggtitle("Unique Google Ad descriptions")
```

```
polarity_unique_ads
```

```
# Exploratory Data Analysis - Image Data -----
```

```
## Not evaluated images
```

```
#Number of sessions in which at least one image has been seen
```

```
nrow(clickstream_data_selected[clickstream_data_selected$photo_urls!="null",])
```

```
#Number of different product images looked at
```

```
length(image_vector_url)
```

```
## Evaluated images
```

```
emotions <- c("happy", "mixed", "neutral", "NA")
picture_emotions <- vector(mode = "double", length = 4)
names(picture_emotions) <- emotions
picture_emotions[1] <- count(happy_images)
picture_emotions[2] <- count(mixed_images)
picture_emotions[3] <- count(neutral_images)
picture_emotions[4] <- count(na_images)
```

```
picture_emotions <- as.data.frame(picture_emotions)
picture_emotions <- t(picture_emotions)
rownames(picture_emotions) <- NULL
colnames(picture_emotions) <- "n"
picture_emotions <- as.data.frame(picture_emotions)
picture_emotions$emotions <- c("happy", "mixed", "neutral", "NA")
```

```

picture_emotions <- picture_emotions[, c("emotions", "n")]

polarity_unique_images <- ggplot(picture_emotions, aes(x=reorder(emotions,n),
y=n))+
  geom_bar(stat = "identity")+
  scale_x_discrete(name="emotionality",limits=c("neutral","mixed","happy"))+
  scale_y_continuous(name = "frequency", labels=scales::comma)+
  theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"))+
  ggtitle("Unique product images")

polarity_unique_images

# Create final datasets -----
##Combine Clickstream Dataset with Google Ad Data
clickstream_data_selected$strong_positive <- 0
clickstream_data_selected$strong_positive[clickstream_data_selected$ad_de-
scription %in% strong_positive_description==TRUE] <- 1

clickstream_data_selected$positive <- 0
clickstream_data_selected$positive[clickstream_data_selected$ad_description
%in% positive_description==TRUE] <- 1

clickstream_data_selected$neutral <- 0
clickstream_data_selected$neutral[clickstream_data_selected$ad_description
%in% neutral_description==TRUE] <- 1

# Change data types
ad_emotion_vector <- c("strong_positive", "positive", "neutral")
for(i in seq_along(ad_emotion_vector)){
  clickstream_data_selected[,ad_emotion_vector[i]] <- as.integer(click-
stream_data_selected[,ad_emotion_vector[i]])
}

## Combine Clickstream Dataset with Image Dataset
clickstream_data_selected$number_happy_images <- str_count(click-
stream_data_selected$photo_urls, paste(happy_images_url, collapse="|"))
clickstream_data_selected$number_mixed_images <- str_count(click-
stream_data_selected$photo_urls, paste(mixed_images_url, collapse="|"))
clickstream_data_selected$number_neutral_images <- str_count(click-
stream_data_selected$photo_urls, paste(neutral_images_url, collapse="|"))
clickstream_data_selected <- clickstream_data_selected %>% mutate(num-
ber_images=number_happy_images+number_mixed_images+number_neu-
tral_images)
colnames(clickstream_data_selected)

#Mean and median of images looked at
mean(clickstream_data_selected[clickstream_data_se-
lected$photo_urls!="null", "number_images"])

```

```

median(clickstream_data_selected[clickstream_data_selected$photo_urls!="null","number_images"])

str(clickstream_data_selected)

## Save
write.csv(clickstream_data_selected, file = "C:\\Users\\felix\\Documents\\0.Uni_Groningen_aktuell\\4. Semester\\Master thesis\\Data & Analysis\\Master_thesis_analysis\\FinalData\\Dataset\\final_clickstream_data_selected.csv")

## Limit to necessary data
combined_data <- clickstream_data_selected %>% filter(marketing_channel
%in% independent_variables) %>%
select("date", "time_cet",
       "user_id",
       "views_product_overview", "views_product_detail", "transactions",
       "transaction_revenue",
       "marketing_channel", "display", "cpc", "email", "affiliate", "direct",
       "organic", "referral", "social",
       "ad_description", "strong_positive", "positive", "neutral",
       "Europe", "Asia", "Americas", "Africa", "Oceania",
       "March", "April",
       "weekend", "no_weekend",
       "morning", "afternoon", "evening", "night",
       "familiar", "not_familiar",
       "promotion_day", "no_promotion_day",
       "number_happy_images", "number_mixed_images",
       "number_neutral_images", "number_images")

combined_data <- combined_data[order(combined_data$user_id, combined_data$date, combined_data$time_cet),]

## Format transaction revenue and Google Ad column
combined_data$transaction_revenue[is.na(combined_data$transaction_revenue)]
<- 0
combined_data$ad_description[is.na(combined_data$ad_description)] <- ""

## Group to customer journeys
#Select (a) single-session journeys and (b) multi-session journey
user_vector <- as.vector(combined_data$user_id)

#(a) Identify single-session journeys
combined_data$instruction_2 <- ""
for (i in seq_along(user_vector)) {

```



```

  ifelse(combined_data$transactions[i]>0 & combined_data$user_id[i]==combined_data$user_id[i-1] & combined_data$transactions[i-1]>0,combined_data$instruction_2[i] <- "single-session journey", combined_data$instruction_2[i] <- "")
}
combined_data$instruction_2[1] <- ""

```

```

combined_data$instruction_3 <- ""
for (i in seq_along(user_vector)) {
  ifelse(combined_data$transactions[i]>0 & combined_data$user_id[i]!=combined_data$user_id[i-1],combined_data$instruction_3[i] <- "single-session journey", combined_data$instruction_3[i] <- "")
}
combined_data$instruction_3[1] <- ""

```

#(b) Identify multi-session journeys

```

combined_data$instruction <- ""
for (i in seq_along(user_vector)) {
  ifelse(combined_data$transactions[i]>0 & combined_data$user_id[i]==combined_data$user_id[i-1] & combined_data$transactions[i-1]==0,combined_data$instruction[i] <- "group with previous columns showing same user_id", combined_data$instruction[i] <- "")
}
combined_data$instruction[1] <- ""

```

#Check if identification is correct

```

combined_data %>% filter(transactions>0 & instruction=="" & instruction_2=="" & instruction_3=="")

```

#(a) Create dataset for single-session journeys

```

single_session_dataset <- combined_data %>% filter(instruction_2=="single-session journey" | instruction_3=="single-session journey")
single_session_dataset$instruction <- NULL
single_session_dataset$instruction_2 <- NULL
single_session_dataset$instruction_3 <- NULL

```

```

single_session_dataset$time_cet <- NULL
single_session_dataset$date <- NULL
single_session_dataset$ad_description <- NULL
single_session_dataset$transaction_revenue[is.na(single_session_dataset$transaction_revenue)] <- 0

```

#(b) Create dataset for multi-session journeys

```

help <- combined_data %>% filter(instruction=="group with previous columns showing same user_id") %>% select(user_id)
help_vector <- as.vector(help$user_id)
multi_session_dataset <- combined_data %>% filter(instruction_2=="" & instruction_3=="" & user_id %in% help_vector)
multi_session_dataset$instruction_2 <- NULL

```

```
multi_session_dataset$instruction_3 <- NULL
```

#(b) Limit rows to one multi-session journey per customer

```
multi_session_dataset$instruction_4 <- ""
for (i in seq_along(multi_session_dataset$user_id)) {
  if(multi_session_dataset$instruction[i]!=""){
    multi_session_dataset$instruction_4[(i+1):length(multi_session_da-
taset$user_id)][multi_session_dataset$user_id[(i+1):length(multi_session_da-
taset$user_id)]==multi_session_dataset$user_id[i]] <- "x"
  }
}
multi_session_dataset <- multi_session_dataset %>% filter(instruction_4!="x")
```

```
multi_session_dataset$instruction <- NULL
multi_session_dataset$instruction_4 <- NULL
multi_session_dataset$transaction_revenue <- as.numeric(multi_session_da-
taset$transaction_revenue)
multi_session_dataset$transaction_revenue[is.na(multi_session_dataset$transac-
tion_revenue)] <- 0
multi_session_dataset$ad_description[is.na(multi_session_dataset$ad_descrip-
tion)] <- ""
```

#(b) Group multi-session journeys

```
grouped_multi_session_dataset <- multi_session_dataset %>% group_by(user_id)
%>% summarise(views_product_overview=sum(views_product_overview),
views_product_detail=sum(views_product_detail), transactions=sum(transac-
tions), transaction_revenue=sum(transaction_revenue), display=sum(display),
cpc=sum(cpc),email=sum(email), affiliate=sum(affiliate), direct=sum(direct), or-
ganic=sum(organic), referral=sum(referral), social=sum(social),
strong_positive=sum(strong_positive), positive=sum(positive), neutral=sum(neu-
tral), familiar=sum(familiar), not_familiar=sum(not_familiar), Europe=sum(Eu-
rope), Asia=sum(Asia), Americas=sum(Americas), Africa=sum(Africa), Oce-
ania=sum(Oceania), morning=sum(morning), afternoon=sum(afternoon), even-
ing=sum(evening), night=sum(night), weekend=sum(weekend), no_week-
end=sum(no_weekend), March=sum(March), April=sum(April), promo-
tion_day=sum(promotion_day), no_promotion_day=sum(no_promotion_day),
number_happy_images=sum(number_happy_images), number_mixed_im-
ages=sum(number_mixed_images), number_neutral_images=sum(number_neu-
tral_images), number_images=sum(number_images))
```

#(b) Adapt grouped multi-session dataset

```
grouped_multi_session_dataset$familiar[grouped_multi_session_dataset$famil-
iar>0] <- 1
grouped_multi_session_dataset$not_familiar[grouped_multi_session_dataset$fa-
miliar==1] <- 0
grouped_multi_session_dataset$not_familiar[grouped_multi_session_da-
taset$not_familiar>1] <- 1
```

```

grouped_multi_session_dataset$Europe[grouped_multi_session_dataset$Europe>0] <- 1
grouped_multi_session_dataset$Asia[grouped_multi_session_dataset$Asia>0] <- 1
grouped_multi_session_dataset$Americas[grouped_multi_session_dataset$Americas>0] <- 1
grouped_multi_session_dataset$Africa[grouped_multi_session_dataset$Africa>0] <- 1
grouped_multi_session_dataset$Oceania[grouped_multi_session_dataset$Oceania>0] <- 1

```

#Combine (a)single-session journey dataset with (b)multi-session journey dataset

```

single_session_dataset_adapted <- single_session_dataset
single_session_dataset_adapted$marketing_channel <- NULL
converting_dataset <- rbind(single_session_dataset_adapted, grouped_multi_session_dataset)

```

#Combine converting dataset with (c)not-converting dataset

```

converting_user <- converting_dataset$user_id
converting_user <- as.vector(converting_user)

```

```

not_converting_dataset <- combined_data %>% filter(user_id %nin% converting_user)

```

#(c) Group not-converting journeys

```

grouped_not_converting_dataset <- not_converting_dataset %>%
group_by(user_id) %>% summarise(views_product_overview=sum(views_product_overview), views_product_detail=sum(views_product_detail), transactions=sum(transactions), transaction_revenue=sum(transaction_revenue), display=sum(display), cpc=sum(cpc),email=sum(email), affiliate=sum(affiliate), direct=sum(direct), organic=sum(organic), referral=sum(referral), social=sum(social), strong_positive=sum(strong_positive), positive=sum(positive), neutral=sum(neutral), familiar=sum(familiar), not_familiar=sum(not_familiar), Europe=sum(Europe), Asia=sum(Asia), Americas=sum(Americas), Africa=sum(Africa), Oceania=sum(Oceania), morning=sum(morning), afternoon=sum(afternoon), evening=sum(evening), night=sum(night), weekend=sum(weekend), no_weekend=sum(no_weekend), March=sum(March), April=sum(April), promotion_day=sum(promotion_day), no_promotion_day=sum(no_promotion_day), number_happy_images=sum(number_happy_images), number_mixed_images=sum(number_mixed_images), number_neutral_images=sum(number_neutral_images), number_images=sum(number_images))

```

#(c) Adapt grouped not-converting dataset

```

grouped_not_converting_dataset$familiar[grouped_not_converting_dataset$familiar>0] <- 1
grouped_not_converting_dataset$not_familiar[grouped_not_converting_dataset$familiar==1] <- 0

```

```

grouped_not_converting_dataset$not_familiar[grouped_not_converting_da-
taset$not_familiar>1] <- 1

grouped_not_converting_dataset$Europe[grouped_not_converting_dataset$Eu-
rope>0] <- 1
grouped_not_converting_dataset$Asia[grouped_not_converting_dataset$Asia>0]
<- 1
grouped_not_converting_dataset$Americas[grouped_not_converting_da-
taset$Americas>0] <- 1
grouped_not_converting_dataset$Africa[grouped_not_converting_dataset$Af-
rica>0] <- 1
grouped_not_converting_dataset$Oceania[grouped_not_converting_dataset$Oce-
ania>0] <- 1

#Combine converting dataset with (c)not-converting dataset
final_dataset_1 <- rbind(converting_dataset, grouped_not_converting_dataset)
final_dataset_1 <- final_dataset_1[order(final_dataset_1$user_id),]

#Save
write.csv(final_dataset_1, file = "C:\\Users\\felix\\Documents\\0.Uni_Gro-
ningen_aktuell\\4. Semester\\Master thesis\\Data & Analysis\\Master_thesis_anal-
ysis\\FinalData\\Dataset\\final_dataset_1.csv")

## Create additional dataset for Shapley Value approach (select finally converting
consumer paths, keep session level, and consecutively add up variables for each
journey)
#Double Google Ad columns to keep original one after adding up
multi_session_dataset$strong_positive_help <- multi_session_dataset$strong_pos-
itive
multi_session_dataset$positive_help <- multi_session_dataset$positive
multi_session_dataset$neutral_help <- multi_session_dataset$neutral

#As dataset need do have same number of columns, same for single-session da-
taset
single_session_dataset$strong_positive_help <- single_session_da-
taset$strong_positive
single_session_dataset$positive_help <- single_session_dataset$positive
single_session_dataset$neutral_help <- single_session_dataset$neutral

#Consecutively add up variables in multi-session journeys
for(i in 2:length(multi_session_dataset$user_id)){
  if(multi_session_dataset$user_id[i]==multi_session_dataset$user_id[i-
1]){multi_session_dataset$views_product_overview[i] <- multi session da-
taset$views_product_overview[i-1]+multi_session_dataset$views_product_over-
view[i];
  multi_session_dataset$views_product_detail[i] <- multi_session_da-
taset$views_product_detail[i-1]+multi_session_dataset$views_product_detail[i];

```

```

multi_session_dataset$transactions[i] <- multi_session_dataset$transactions[i-
1]+multi_session_dataset$transactions[i];
multi_session_dataset$transaction_revenue[i] <- multi_session_dataset$transac-
tion_revenue[i-1]+multi_session_dataset$transaction_revenue[i];
multi_session_dataset$display[i] <- multi_session_dataset$display[i-
1]+multi_session_dataset$display[i];
multi_session_dataset$scpc[i] <- multi_session_dataset$scpc[i-1]+multi_ses-
sion_dataset$scpc[i];
multi_session_dataset$affiliate[i] <- multi_session_dataset$affiliate[i-
1]+multi_session_dataset$affiliate[i];
multi_session_dataset$direct[i] <- multi_session_dataset$direct[i-1]+multi_ses-
sion_dataset$direct[i];
multi_session_dataset$organic[i] <- multi_session_dataset$organic[i-
1]+multi_session_dataset$organic[i];
multi_session_dataset$referral[i] <- multi_session_dataset$referral[i-
1]+multi_session_dataset$referral[i];
multi_session_dataset$social[i] <- multi_session_dataset$social[i-1]+multi_ses-
sion_dataset$social[i];
multi_session_dataset$strong_positive[i] <- multi_session_dataset$strong_posi-
tive[i-1]+multi_session_dataset$strong_positive[i];
multi_session_dataset$positive[i] <- multi_session_dataset$positive[i-
1]+multi_session_dataset$positive[i];
multi_session_dataset$neutral[i] <- multi_session_dataset$neutral[i-
1]+multi_session_dataset$neutral[i];
multi_session_dataset$morning[i] <- multi_session_dataset$morning[i-
1]+multi_session_dataset$morning[i];
multi_session_dataset$afternoon[i] <- multi_session_dataset$afternoon[i-
1]+multi_session_dataset$afternoon[i];
multi_session_dataset$evening[i] <- multi_session_dataset$evening[i-
1]+multi_session_dataset$evening[i];
multi_session_dataset$night[i] <- multi_session_dataset$night[i-1]+multi_ses-
sion_dataset$night[i];
multi_session_dataset$weekend[i] <- multi_session_dataset$weekend[i-
1]+multi_session_dataset$weekend[i];
multi_session_dataset$no_weekend[i] <- multi_session_dataset$no_weekend[i-
1]+multi_session_dataset$no_weekend[i];
multi_session_dataset$March[i] <- multi_session_dataset$March[i-
1]+multi_session_dataset$March[i];
multi_session_dataset$April[i] <- multi_session_dataset$April[i-1]+multi_ses-
sion_dataset$April[i];
multi_session_dataset$promotion_day[i] <- multi_session_dataset$promo-
tion_day[i-1]+multi_session_dataset$promotion_day[i];
multi_session_dataset$no_promotion_day[i] <- multi_session_dataset$no_pro-
motion_day[i-1]+multi_session_dataset$no_promotion_day[i];
multi_session_dataset$number_happy_images[i] <- multi_session_dataset$num-
ber_happy_images[i-1]+multi_session_dataset$number_happy_images[i];
multi_session_dataset$number_mixed_images[i] <- multi_session_dataset$num-
ber_mixed_images[i-1]+multi_session_dataset$number_mixed_images[i];

```

```

    multi_session_dataset$number_neutral_images[i] <- multi_session_da-
taset$number_neutral_images[i-1]+multi_session_dataset$number_neutral_im-
ages[i];
    multi_session_dataset$number_images[i] <- multi_session_dataset$number_im-
ages[i-1]+multi_session_dataset$number_images[i]
  }
}

```

```

multi_session_dataset$time_cet <- NULL
multi_session_dataset$date <- NULL
multi_session_dataset$ad_description <- NULL

```

#Combine part_1 and part_2

```

final_dataset_2 <- rbind(multi_session_dataset, single_session_dataset)
view(final_dataset_2)

```

#Save

```

write.csv(final_dataset_2, file = "C:\\Users\\felix\\Documents\\0.Uni_Gro-
ningen_aktuell\\4. Semester\\Master thesis\\Data & Analysis\\Master_thesis_anal-
ysis\\FinalData\\Dataset\\final_dataset_2.csv")

```

Data Wrangling - Final Dataset 1 -----

##(a) Final dataset 1

```

final_dataset_1 <- read.csv("final_dataset_1.csv", header = TRUE, stringsAsFac-
tors = FALSE, na.strings = "", encoding = "UTF-8")
dim(final_dataset_1)
str(final_dataset_1)

```

Delete unnecessary column

```

final_dataset_1$X <- NULL

```

Change naming

```

setnames(final_dataset_1, old = c("number_happy_images", "number_mixed_im-
ages", "number_neutral_images", "number_images"), new=c("happy_images",
"mixed_images", "neutral_images", "images"))

```

Change data tpye

```

familiarity_vector <- c("familiar", "not_familiar")
for(i in seq_along(familiarity_vector)){
  final_dataset_1[,familiarity_vector[i]] <- as.integer(final_dataset_1[,familiar-
ity_vector[i]])
}

```

continent_vector <- c("Europe", "Asia", "Americas", "Africa", "Oceania")

```

for(i in seq_along(continent_vector)){
  final_dataset_1[,continent_vector[i]] <- as.integer(final_dataset_1[,conti-
nent_vector[i]])
}

```

```

## Check NAs
summary(final_dataset_1)

variables_1 <- colnames(final_dataset_1)
number_NAs_1 <- vector(mode = "double", length = length(variables_1))
names(number_NAs_1) <- variables_1

for (i in seq_along(variables_1)) {
  number_NAs_1[i] <- sum(is.na(final_dataset_1[variables_1[i]]))
}

number_NAs_1

#Check missing values
number_missing_values_1 <- vector(mode = "double", length = length(variables_1))
names(number_missing_values_1) <- variables_1

for (i in seq_along(variables_1)) {
  number_missing_values_1[i] <- sum(is_empty(final_dataset_1[variables_1[i]]))
}

number_missing_values_1

## Check outliers
#"display", "cpc", "email", "affiliate", "direct", "organic", "referral", "social"
a <- c("display", "cpc", "email", "affiliate", "direct", "organic", "referral", "social")
b <- c("display interaction", "cpc interaction", "email interaction", "affiliate interaction", "direct interaction", "organic interaction", "referral interaction", "social interaction")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_1, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

boxplot_dis_1 <- plot_list[[1]]
boxplot_cpc_1 <- plot_list[[2]]
boxplot_ema_1 <- plot_list[[3]]
boxplot_aff_1 <- plot_list[[4]]
boxplot_dir_1 <- plot_list[[5]]
boxplot_org_1 <- plot_list[[6]]

```

```

boxplot_ref_1 <- plot_list[[7]]
boxplot_soc_1 <- plot_list[[8]]

plot_grid(boxplot_dis_1, boxplot_cpc_1, boxplot_ema_1, boxplot_aff_1, box-
plot_dir_1, boxplot_org_1, boxplot_ref_1, boxplot_soc_1, labels = "AUTO")

```

#In-depth check for outliers

```

final_dataset_1 %>% filter(display>20)
final_dataset_1 %>% filter(cpc>80)
final_dataset_1 %>% filter(email>100)
final_dataset_1 %>% filter(affiliate>100)
final_dataset_1 %>% filter(direct>60)
final_dataset_1 %>% filter(organic>60)
final_dataset_1 %>% filter(referral>50)
final_dataset_1 %>% filter(social>20)

```

#"strong_positive", "positive", "neutral"

```

a <- c("strong_positive", "positive", "neutral")
b <- c("strong_positive", "positive", "neutral")

```

```

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_1, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

```

```

boxplot_spo_1 <- plot_list[[1]]
boxplot_pos_1 <- plot_list[[2]]
boxplot_neu_1 <- plot_list[[3]]

```

```

plot_grid(boxplot_spo_1, boxplot_pos_1, boxplot_neu_1, labels = "AUTO")

```

#In-depth check for outliers

```

final_dataset_1 %>% filter(strong_positive>60)
final_dataset_1 %>% filter(positive>20)
final_dataset_1 %>% filter(neutral>50)

```

*#"views_product_overview", "views_product_detail", "transactions", "transac-
tion_revenue"*

```

a <- c("views_product_overview", "views_product_detail", "transactions", "trans-
action_revenue")
b <- c("views on product overview pages", "views on product detail pages",
"transactions", "transaction revenue")

```



```

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_1, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

```

```

boxplot_vpo_1 <- plot_list[[1]]
boxplot_vpd_1 <- plot_list[[2]]
boxplot_tra_1 <- plot_list[[3]]
boxplot_tre_1 <- plot_list[[4]]

```

```

plot_grid(boxplot_vpo_1, boxplot_vpd_1, boxplot_tra_1, boxplot_tre_1, labels = "AUTO")

```

#In-depth check for outliers

```

final_dataset_1 %>% filter(views_product_overview>1000)
final_dataset_1 %>% filter(views_product_detail>500)
final_dataset_1 %>% filter(transaction_revenue>20000)

```

#"March", "April"

```

a <- c("March", "April")
b <- c("March", "April")

```

```

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_1, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

```

```

boxplot_mar_1 <- plot_list[[1]]
boxplot_apr_1 <- plot_list[[2]]

```

```

plot_grid(boxplot_mar_1, boxplot_apr_1, labels = "AUTO")

```

#In-depth check for outliers

```

final_dataset_1 %>% filter(March>100)
final_dataset_1 %>% filter(April>50)

```

#"weekend", "no_weekend"

```

a <- c("weekend", "no_weekend")
b <- c("weekend", "no weekend")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_1, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

boxplot_wee_1 <- plot_list[[1]]
boxplot_nwe_1 <- plot_list[[2]]

plot_grid(boxplot_wee_1, boxplot_nwe_1, labels = "AUTO")

#In-depth check for outliers
final_dataset_1 %>% filter(weekend>30)
final_dataset_1 %>% filter(no_weekend>125)

#"morning", "afternoon", "evening", "night"
a <- c("morning", "afternoon", "evening", "night")
b <- c("morning", "afternoon", "evening", "night")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_1, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

boxplot_mor_1 <- plot_list[[1]]
boxplot_aft_1 <- plot_list[[2]]
boxplot_eve_1 <- plot_list[[3]]
boxplot_nig_1 <- plot_list[[4]]

plot_grid(boxplot_mor_1, boxplot_aft_1, boxplot_eve_1, boxplot_nig_1, labels = "AUTO")

#In-depth check for outliers
final_dataset_1 %>% filter(morning>60)
final_dataset_1 %>% filter(afternoon>60)

```

```

final_dataset_1 %>% filter(evening>60)
final_dataset_1 %>% filter(night>20)

#"promotion_day", "no_promotion_day"
a <- c("promotion_day", "no_promotion_day")
b <- c("promotion day", "no promotion day")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_1, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

boxplot_pda_1 <- plot_list[[1]]
boxplot_npd_1 <- plot_list[[2]]

plot_grid(boxplot_pda_1, boxplot_npd_1, labels = "AUTO")

#In-depth check for outliers
final_dataset_1 %>% filter(promotion_day>20)
final_dataset_1 %>% filter(no_promotion_day>125)

#"number_happy_images", "number_mixed_images", "number_neutral_images",
"number_images"
a <- c("happy_images", "mixed_images", "neutral_images", "images")
b <- c("happy images", "mixed images", "number neutral images", "images")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_1, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

boxplot_him_1 <- plot_list[[1]]
boxplot_mim_1 <- plot_list[[2]]
boxplot_nim_1 <- plot_list[[3]]
boxplot_ima_1 <- plot_list[[4]]

```

```
plot_grid(boxplot_him_1, boxplot_mim_1, boxplot_nim_1, boxplot_ima_1, labels
= "AUTO")
```

```
#In-depth check for outliers
```

```
final_dataset_1 %>% filter(happy_images>400)
final_dataset_1 %>% filter(mixed_images>400)
final_dataset_1 %>% filter(neutral_images>250)
final_dataset_1 %>% filter(images>1000)
```

```
write.csv(final_dataset_1, file = "C:\\Users\\felix\\Documents\\0.Uni_Gro-
ningen_aktuell\\4. Semester\\Master thesis\\Data & Analysis\\Master_thesis_anal-
ysis\\FinalData\\Dataset\\final_dataset_1.csv")
```

```
# Data Wrangling - Final Dataset 2 -----
```

```
final_dataset_2 <- read.csv("final_dataset_2.csv", header = TRUE, stringsAsFac-
tors = FALSE, na.strings = "", encoding = "UTF-8")
dim(final_dataset_2)
str(final_dataset_2)
```

```
## Delete unnecessary column
```

```
final_dataset_2$X <- NULL
```

```
## Change naming
```

```
setnames(final_dataset_2, old = c("number_happy_images", "number_mixed_im-
ages", "number_neutral_images", "number_images"), new=c("happy_images",
"mixed_images", "neutral_images", "images"))
```

```
## Check NAs
```

```
summary(final_dataset_2)
```

```
variables_2 <- colnames(final_dataset_2)
number_NAs_2 <- vector(mode = "double", length = length(variables_2))
names(number_NAs_2) <- variables_2
```

```
for (i in seq_along(variables_2)) {
  number_NAs_2[i] <- sum(is.na(final_dataset_2[variables_2[i]]))
}
```

```
number_NAs_2
```

```
## Check missing values
```

```
number_missing_values_2 <- vector(mode = "double", length = length(varia-
bles_2))
names(number_missing_values_2) <- variables_2
```

```
for (i in seq_along(variables_2)) {
  number_missing_values_2[i] <- sum(is_empty(final_dataset_2[variables_2[i]]))
}
```

```

}

number_missing_values_2

## Check outliers
#"display", "cpc", "email", "affiliate", "direct", "organic", "referral", "social"
a <- c("display", "cpc", "email", "affiliate", "direct", "organic", "referral", "so-
cial")
b <- c("display interaction", "cpc interaction", "email interaction", "affiliate inter-
action", "direct interaction", "organic interaction", "referral interaction", "social
interaction")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_2, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

boxplot_dis_2 <- plot_list[[1]]
boxplot_cpc_2 <- plot_list[[2]]
boxplot_ema_2 <- plot_list[[3]]
boxplot_aff_2 <- plot_list[[4]]
boxplot_dir_2 <- plot_list[[5]]
boxplot_org_2 <- plot_list[[6]]
boxplot_ref_2 <- plot_list[[7]]
boxplot_soc_2 <- plot_list[[8]]

plot_grid(boxplot_dis_2, boxplot_cpc_2, boxplot_ema_2, boxplot_aff_2, box-
plot_dir_2, boxplot_org_2, boxplot_ref_2, boxplot_soc_2, labels = "AUTO")

#"strong positive", "positive", "neutral"
a <- c("strong_positive", "positive", "neutral")
b <- c("strong positive", "positive", "neutral")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_2, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

```

```

boxplot_spo_2 <- plot_list[[1]]
boxplot_pos_2 <- plot_list[[2]]
boxplot_neu_2 <- plot_list[[3]]

plot_grid(boxplot_spo_2, boxplot_pos_2, boxplot_neu_2, labels = "AUTO")

#"views_product_overview", "views_product_detail", "transactions", "transaction_revenue"
a <- c("views_product_overview", "views_product_detail", "transactions", "transaction_revenue")
b <- c("views on product overview pages", "views on product detail pages", "transactions", "transaction revenue")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_2, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

boxplot_pov_2 <- plot_list[[1]]
boxplot_pdv_2 <- plot_list[[2]]
boxplot_tra_2 <- plot_list[[3]]
boxplot_re_2 <- plot_list[[4]]

plot_grid(boxplot_pov_2, boxplot_pdv_2, boxplot_tra_2, boxplot_re_2, labels = "AUTO")

#In-depth check for outliers
final_dataset_2 %>% filter(transaction_revenue>20000)

#"March", "April"
a <- c("March", "April")
b <- c("March", "April")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_2, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

```

```

}

boxplot_mar_2 <- plot_list[[1]]
boxplot_apr_2 <- plot_list[[2]]

plot_grid(boxplot_mar_2, boxplot_apr_2, labels = "AUTO")

#"weekend", "no_weekend"
a <- c("weekend", "no_weekend")
b <- c("weekend", "no weekend")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_2, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

boxplot_wee_2 <- plot_list[[1]]
boxplot_nwe_2 <- plot_list[[2]]

plot_grid(boxplot_wee_2, boxplot_nwe_2, labels = "AUTO")

#"morning", "afternoon", "evening", "night"
a <- c("morning", "afternoon", "evening", "night")
b <- c("morning", "afternoon", "evening", "night")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_2, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

boxplot_mor_2 <- plot_list[[1]]
boxplot_aft_2 <- plot_list[[2]]
boxplot_eve_2 <- plot_list[[3]]
boxplot_nig_2 <- plot_list[[4]]

plot_grid(boxplot_mor_2, boxplot_aft_2, boxplot_eve_2, boxplot_nig_2, labels =
"AUTO")

```

```

# "promotion_day", "no_promotion_day"
a <- c("promotion_day", "no_promotion_day")
b <- c("promotion day", "no promotion day")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_2, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

boxplot_pda_2 <- plot_list[[1]]
boxplot_npd_2 <- plot_list[[2]]

plot_grid(boxplot_pda_2, boxplot_npd_2, labels = "AUTO")

# "happy_images", "mixed_images", "neutral_images", "images"
a <- c("happy_images", "mixed_images", "neutral_images", "images")
b <- c("happy images", "mixed images", "number neutral images", "images")

plot_list <- list()
for(i in seq_along(a)){
  c <- ggplot(final_dataset_2, aes_string(y=a[i]))+
    geom_boxplot()+
    scale_x_discrete(paste("#", b[i], sep = ""))+
    ylab("frequency")+
    theme(panel.background = element_rect(fill = "white"), panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))
  plot_list[[i]] <- c
}

boxplot_him_2 <- plot_list[[1]]
boxplot_mim_2 <- plot_list[[2]]
boxplot_nim_2 <- plot_list[[3]]
boxplot_ima_2 <- plot_list[[4]]

plot_grid(boxplot_him_2, boxplot_mim_2, boxplot_nim_2, boxplot_ima_2, labels = "AUTO")

# In-depth check for outliers
final_dataset_2 %>% filter(happy_images>250)
final_dataset_2 %>% filter(mixed_images>250)
final_dataset_2 %>% filter(neutral_images>200)
final_dataset_1 %>% filter(images>700)

```



```
write.csv(final_dataset_2, file = "C:\\Users\\felix\\Documents\\0.Uni_Gro-
ningen_aktuell\\4. Semester\\Master thesis\\Data & Analysis\\Master_thesis_anal-
ysis\\FinalData\\Dataset\\final_dataset_2.csv")
```

```
# Exploratory Data Analysis - Final Dataset 1 -----
```

```
## IVs
```

```
number_absolute_IV_1 <- vector(mode = "double", length = length(independ-
ent_variables))
```

```
names(number_absolute_IV_1) <- independent_variables
```

```
for(i in seq_along(independent_variables)){
  number_absolute_IV_1[i] <- sum(final_dataset_1[,independent_variables[i]])
}
```

```
number_absolute_IV_1
```

```
mean_iv <- c("")
```

```
for(i in seq_along(independent_variables)){
  mean_iv[i] <- mean(final_dataset_1[,independent_variables[i]])
}
```

```
median_iv <- c("")
```

```
for(i in seq_along(independent_variables)){
  median_iv[i] <- median(final_dataset_1[,independent_variables[i]])
}
```

```
#Google Ad descriptions
```

```
google_ad_descriptions <- c("strong_positive", "positive", "neutral")
```

```
number_absolute_GA_1 <- vector(mode = "double", length =
length(google_ad_descriptions))
```

```
names(number_absolute_GA_1) <- google_ad_descriptions
```

```
for(i in seq_along(google_ad_descriptions)){
  number_absolute_GA_1[i] <- sum(final_dataset_1[,google_ad_descriptions[i]])
}
```

```
number_absolute_GA_1 <- as.data.frame(number_absolute_GA_1)
```

```
colnames(number_absolute_GA_1) <- "frequency"
```

```
number_absolute_GA_1
```

```
polarity_ads_final_1 <- ggplot(number_absolute_GA_1, aes(x=reor-
der(rownames(number_absolute_GA_1),frequency), y=frequency))+
  geom_bar(stat = "identity")+
  theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"))+
  scale_x_discrete(name="emotionality", labels=c("positive", "strong positive",
"neutral"))+
  ggtitle("Customer journey dataset")
```

```

polarity_ads_final_1

plot_grid(polarity_unique_ads, polarity_ads_final_1, labels = "AUTO")

## DVs
number_absolute_DV_1 <- vector(mode = "double", length = length(dependent_variables))
names(number_absolute_DV_1) <- dependent_variables

for(i in seq_along(dependent_variables)){
  number_absolute_DV_1[i] <- sum(final_dataset_1[,dependent_variables[i]])
}

number_absolute_DV_1

#Number of journeys with at least one transaction
length(final_dataset_1$transactions[final_dataset_1$transactions>0])

#Mean and median of product overview and product detail page interactions
mean(final_dataset_1$views_product_overview)
mean(final_dataset_1$views_product_detail)

median(final_dataset_1$views_product_overview)
median(final_dataset_1$views_product_detail)

#Mean and median of transaction revenue for journeys with at least one transaction
mean(final_dataset_1$transaction_revenue[final_dataset_1$transactions>0])
median(final_dataset_1$transaction_revenue[final_dataset_1$transactions>0])

## Controls
#(a)Continent
sum(final_dataset_1$Europe)
sum(final_dataset_1$Asia)
sum(final_dataset_1$Americas)
sum(final_dataset_1$Africa)
sum(final_dataset_1$Oceania)

#(b)Month
sum(final_dataset_1$March)
sum(final_dataset_1$April)

#(c)Weekend
sum(final_dataset_1$weekend)
sum(final_dataset_1$no_weekend)

#(d)Time of day
sum(final_dataset_1$morning)

```

```

sum(final_dataset_1$afternoon)
sum(final_dataset_1$evening)
sum(final_dataset_1$night)

#(e)Familiarity
sum(final_dataset_1$familiar)
sum(final_dataset_1$not_familiar)

#(f)Promotion status
sum(final_dataset_1$promotion_day)
sum(final_dataset_1$no_promotion_day)

#(g)Image Data
image_emotions <- c("happy_images", "mixed_images", "neutral_images", "im-
ages")
number_absolute_IM_1 <- vector(mode = "double", length = length(image_emo-
tions))
names(number_absolute_IM_1) <- image_emotions

for(i in seq_along(image_emotions)){
  number_absolute_IM_1[i] <- sum(final_dataset_1[,image_emotions[i]])
}

number_absolute_IM_1 <- as.data.frame(number_absolute_IM_1)
colnames(number_absolute_IM_1) <- "frequency"
number_absolute_IM_1

polarity_images_final_1 <- ggplot(number_absolute_IM_1, aes(x=reor-
der(rownames(number_absolute_IM_1), frequency), y=frequency))+
  geom_bar(stat = "identity")+
  scale_x_discrete(name="emotionality", limits=c("neutral_images", "mixed_im-
ages", "happy_images"), labels = c("neutral", "mixed", "happy"))+
  theme(panel.background = element_rect(fill = "white"), panel.grid = ele-
ment_line(colour = "grey92"), text = element_text(size=16, family = "sans"))+
  ggtitle("Customer journey dataset")
polarity_images_final_1

plot_grid(polarity_unique_images, polarity_images_final_1, labels = "AUTO")

## Correlations
cor(final_dataset_1[,independent_variables])
# Analysis -----
# Train, Validate & Test RFs -----
str(final_dataset_1)

## Splitting the data
set.seed(123)
final_dataset_1_shuffled <- final_dataset_1[sample(1:nrow(final_dataset_1),
nrow(final_dataset_1), replace = F),]

```

```

x.train <- final_dataset_1_shuffled[1:20000, ]
x.validate <- final_dataset_1_shuffled[20001:40000, ]
x.test <- final_dataset_1_shuffled[40001:140000, ]

## Random Forest Regression Trees
#Product overview page
#(a) without UD
#Training
#Default values
rf_pop_UD <- randomForest(views_product_overview ~ display + cpc + email +
  affiliate + direct + organic + referral + social +
    Europe + Asia + Americas + Africa + Oceania +
    March + April +
    weekend + no_weekend +
    morning + afternoon + evening + night +
    familiar +
    promotion_day + no_promotion_day+
    images,
  data = x.train)

print(rf_pop_UD)
plot(rf_pop_UD)

#Bayesian optimization - Hyperparameter tuning (mtry, min_node_size)
drop <- c("user_id", "views_product_detail", "transactions", "transaction_reve-
  nue", "images", "not_familiar")

tuned_rf <- rf_opt(train_data = x.train[, !names(x.train) %in% drop],
  train_label = views_product_overview,
  test_data = validate_sample[, !names(validate_sample) %in% drop],
  test_label = views_product_overview,
  mtry_range = c(1L, ncol(x.train[, !names(x.train) %in% drop])-1),
  min_node_size = c(1L, as.integer(sqrt(nrow(x.train))))),
  acq = "ucb",
  init_points = 4,
  n_iter = 10
)

tuned_rf_param <- data.frame(tuned_rf$Best_Par)
colnames(tuned_rf_param) <- "value"
mtry_opt <- tuned_rf_param$value[1]
min_node_size_opt <- tuned_rf_param$value[2]

test <- randomForest(views_product_overview ~ display + cpc + email + affiliate
  + direct + organic + referral + social +
    Europe + Asia + Americas + Africa + Oceania +
    March + April +
    weekend + no_weekend +
    morning + afternoon + evening + night +

```

```

        familiar +
        promotion_day + no_promotion_day+
        images,
        mtry = mtry_opt,
        nodesize = min_node_size_opt,
        data = validate_sample)

pred <- predict(object = test, newdata = validate_sample)
rmse(actual = validate_sample$views_product_overview, predicted = pred)
mae(actual = validate_sample$views_product_overview, predicted = pred)
#Apply grid search as Bayesian updating does not lead to improved results

#Grid search - Hyperparameter tuning
mtry <- seq(6,ncol(x.train)/3,2)
nodesize <- c(10,15)
maxdepth <- seq(20,40,10)
sampsize <- nrow(x.train) * c(0.7,0.8)

hyper_grid <- expand.grid(mtry = mtry, nodesize = nodesize, maxdepth =
maxdepth, sampsize = sampsize)

models_1_a <- list()
for(i in 1:nrow(hyper_grid)){
  mtry <- hyper_grid$mtry[i]
  nodesize <- hyper_grid$nodesize[i]
  maxdepth <- hyper_grid$maxdepth[i]
  sampsize <- hyper_grid$sampsize[i]

  models_1_a[[i]] <- randomForest(views_product_overview ~ display + cpc +
email + affiliate + direct + organic + referral + social +
        Europe + Asia + Americas + Africa + Oceania +
        March + April +
        weekend + no_weekend +
        morning + afternoon + evening + night +
        familiar +
        promotion_day + no_promotion_day+
        images,
        mtry = mtry,
        nodesize = nodesize,
        maxdepth = maxdepth,
        sampsize = sampsize,
        data = x.train)
}

#Validation
#RMSE comparison
rmse_values_1_a <- c()
for(i in 1:length(models_1_a)){
  model <- models_1_a[[i]]

```

```

    pred <- predict(object = model, newdata = x.validate)
    rmse_values_1_a[i] <- rmse(actual = x.validate$views_product_overview, predicted = pred)
  }

min(rmse_values_1_a)
match(min(rmse_values_1_a), rmse_values_1_a)
hyper_grid[match(min(rmse_values_1_a), rmse_values_1_a),]
optimal_model_rmse_1_a <- models_1_a[[match(min(rmse_values_1_a), rmse_values_1_a)]]

#MAE comparison
mae_values_1_a <- c()
for(i in 1:length(models_1_a)){
  model <- models_1_a[[i]]
  pred <- predict(object = model, newdata = x.validate)
  mae_values_1_a[i] <- mae(actual = x.validate$views_product_overview, predicted = pred)
}

min(mae_values_1_a)
match(min(mae_values_1_a), mae_values_1_a)
hyper_grid[match(min(mae_values_1_a), mae_values_1_a),]
optimal_model_mae_1_a <- models_1_a[[match(min(mae_values_1_a), mae_values_1_a)]]

#Decision:
#Choose MAE model as higher higher explained variance and lower MAE
#Test
#MAE model
pred <- predict(object = optimal_model_mae_1_a, newdata = x.test)
rmse(actual = x.test$views_product_overview, predicted = pred)
mae(actual = x.test$views_product_overview, predicted = pred)

#(b) with SD
#Training
#Default values
rf_pop_SD <- randomForest(views_product_overview ~ display + cpc +
  strong_positive + positive + neutral + email + affiliate + direct +
  organic + referral + social +
  Europe + Asia + Americas + Africa + Oceania +
  March + April +
  weekend + no_weekend +
  morning + afternoon + evening + night +
  familiar +
  promotion_day + no_promotion_day +
  happy_images + mixed_images + neutral_images,
  data = x.train)

```

```

print(rf_pop_SD)
plot(rf_pop_SD)

#Grid search - Hyperparameter tuning
mtry <- seq(6,ncol(x.train)/3,2)
nodesize <- c(10,15)
maxdepth <- seq(20,40,10)
sampsize <- nrow(x.train) * c(0.7,0.8)

hyper_grid <- expand.grid(mtry = mtry, nodesize = nodesize, maxdepth =
maxdepth, sampsize = sampsize)

models_1_b <- list()
for(i in 1:nrow(hyper_grid)){
  mtry <- hyper_grid$mtry[i]
  nodesize <- hyper_grid$nodesize[i]
  maxdepth <- hyper_grid$maxdepth[i]
  sampsize <- hyper_grid$sampsize[i]

  models_1_b[[i]] <- randomForest(views_product_overview ~ display + cpc +
    strong_positive + positive + neutral + email + affiliate +
    direct + organic + referral + social +
    Europe + Asia + Americas + Africa + Oceania +
    March + April +
    weekend + no_weekend +
    morning + afternoon + evening + night +
    familiar +
    promotion_day + no_promotion_day +
    happy_images + mixed_images + neutral_images,
    mtry = mtry,
    nodesize = nodesize,
    maxdepth = maxdepth,
    sampsize = sampsize,
    data = x.train)
}

#Validation
#RMSE comparison
rmse_values_1_b <- c()
for(i in 1:length(models_1_b)){
  model <- models_1_b[[i]]
  pred <- predict(object = model, newdata = x.validate)
  rmse_values_1_b[i] <- rmse(actual = x.validate$views_product_overview, pre-
dicted = pred)
}

min(rmse_values_1_b)
match(min(rmse_values_1_b), rmse_values_1_b)
hyper_grid[match(min(rmse_values_1_b), rmse_values_1_b),]

```

```

optimal_model_rmse_1_b <- models_1_b[[match(min(rmse_values_1_b),
rmse_values_1_b)]]

#MAE comparison
mae_values_1_b <- c()
for(i in 1:length(models_1_b)){
  model <- models_1_b[[i]]
  pred <- predict(object = model, newdata = x.validate)
  mae_values_1_b[i] <- mae(actual = x.validate$views_product_overview, pre-
dicted = pred)
}

min(mae_values_1_b)
match(min(mae_values_1_b), mae_values_1_b)
hyper_grid[match(min(mae_values_1_b), mae_values_1_b),]
optimal_model_mae_1_b <- models_1_b[[match(min(mae_values_1_b),
mae_values_1_b)]]

#Decision:
#Choose MAE model as higher higher explained variance and lower MAE

#Test
#MAE model
pred <- predict(object = optimal_model_mae_1_b, newdata = x.test)
rmse(actual = x.test$views_product_overview, predicted = pred)
mae(actual = x.test$views_product_overview, predicted = pred)

#Variable Importance
var_imp_pop <- as.data.frame(importance(optimal_model_mae_1_b))
var_imp_pop$channels <- rownames(var_imp_pop)
colnames(var_imp_pop) <- c("node_purity", "channels")
var_imp_pop <- var_imp_pop[order(var_imp_pop$node_purity, decreasing =
TRUE),]

var_imp_pop_plot <- ggplot(var_imp_pop, aes(x=reorder(channels, node_purity),
y=node_purity))+
  geom_bar(stat = "identity")+
  coord_flip()+
  scale_x_discrete(name="split variables", labels = c("Oceania", "Africa", "Asia",
"display", "Europe", "Americas", "positive", "familiar", "night", "affiliate", "strong
positive", "direct", "referral", "e-mail", "organic", "neutral", "social", "weekend",
"cpc", "promotion day", "evening", "April", "morning", "March", "weekday", "af-
ternoon", "no promotion day", "neutral images", "happy images", "mixed im-
ages"))+
  scale_y_continuous(name="node_purity", labels = comma)+
  ggtitle("Awareness stage")
var_imp_pop_plot

#Plot exemplifary graph

```



```

tree_func <- function(final_model,
                      tree_num) {
  # get tree by index
  tree <- randomForest::getTree(final_model,
                                k = tree_num,
                                labelVar = TRUE) %>%
    tibble::rownames_to_column() %>%
    # make leaf split points to NA, so the 0s won't get plotted
    mutate(`split point` = ifelse(is.na(prediction), `split point`, NA))
  # prepare data frame for graph
  graph_frame <- data.frame(from = rep(tree$rowname, 2),
                            to = c(tree$`left daughter`, tree$`right daughter`))
  # convert to graph and delete the last node that we don't want to plot
  graph <- graph_from_data_frame(graph_frame) %>%
    delete_vertices("0")
  # set node labels
  V(graph)$node_label <- gsub("_", " ", as.character(tree$`split var`))
  V(graph)$leaf_label <- as.character(tree$prediction)
  V(graph)$split <- as.character(round(tree$`split point`, digits = 2))
  # plot
  plot <- ggraph(graph, 'dendrogram') +
    theme_bw() +
    geom_edge_link() +
    geom_node_point() +
    geom_node_text(aes(label = node_label), na.rm = TRUE, repel = TRUE) +
    geom_node_label(aes(label = split), vjust = 2.5, na.rm = TRUE, fill = "white") +
    geom_node_label(aes(label = leaf_label, fill = leaf_label), na.rm = TRUE,
                    repel = TRUE, colour = "white", fontface = "bold",
                    show.legend = FALSE) +
    theme(panel.grid.minor = element_blank(),
          panel.grid.major = element_blank(),
          panel.background = element_blank(),
          plot.background = element_rect(fill = "white"),
          panel.border = element_blank(),
          axis.line = element_blank(),
          axis.text.x = element_blank(),
          axis.text.y = element_blank(),
          axis.ticks = element_blank(),
          axis.title.x = element_blank(),
          axis.title.y = element_blank(),
          plot.title = element_text(size = 18))
  print(plot)
}

check <- which(optimal_model_rmse_1_b$forest$ndbigtree == min(optimal_model_rmse_1_b$forest$ndbigtree))
tree_func(final_model = optimal_model_rmse_1_b, tree_num = check)
#obviously way to extensive to plot

```

```

#Product detail page
#(a) without UD
#Training
#Default values
rf_pdp_UD <- randomForest(views_product_detail ~ views_product_overview +
                           display + cpc + email + affiliate + direct + organic + referral +
                           social +
                           Europe + Asia + Americas + Africa + Oceania +
                           March + April +
                           weekend + no_weekend +
                           morning + afternoon + evening + night +
                           familiar +
                           promotion_day + no_promotion_day,
                           data = x.train)

print(rf_pop_UD)
plot(rf_pop_UD)

#Grid search - Hyperparameter tuning
mtry <- seq(6,ncol(x.train)/3,2)
nodesize <- c(10,15)
maxdepth <- seq(20,40,10)
sampsize <- nrow(x.train) * c(0.7,0.8)

hyper_grid <- expand.grid(mtry = mtry, nodesize = nodesize, maxdepth =
maxdepth, sampsize = sampsize)

models_2_a <- list()
for(i in 1:nrow(hyper_grid)){
  mtry <- hyper_grid$mtry[i]
  nodesize <- hyper_grid$nodesize[i]
  maxdepth <- hyper_grid$maxdepth[i]
  sampsize <- hyper_grid$sampsize[i]

  models_2_a[[i]] <- randomForest(views_product_detail ~
                                   views_product_overview + display + cpc + email + affiliate
                                   + direct + organic + referral + social +
                                   Europe + Asia + Americas + Africa + Oceania +
                                   March + April +
                                   weekend + no_weekend +
                                   morning + afternoon + evening + night +
                                   familiar +
                                   promotion_day + no_promotion_day,
                                   mtry = mtry,
                                   nodesize = nodesize,
                                   maxdepth = maxdepth,
                                   sampsize = sampsize,
                                   data = x.train)

```

```

}

#Validation
#RMSE comparison
rmse_values_2_a <- c()
for(i in 1:length(models_2_a)){
  model <- models_2_a[[i]]
  pred <- predict(object = model, newdata = x.validate)
  rmse_values_2_a[i] <- rmse(actual = x.validate$views_product_detail, predicted
= pred)
}

min(rmse_values_2_a)
match(min(rmse_values_2_a), rmse_values_2_a)
hyper_grid[match(min(rmse_values_2_a), rmse_values_2_a),]
optimal_model_rmse_2_a <- models_2_a[[match(min(rmse_values_2_a),
rmse_values_2_a)]]

#MAE comparison
mae_values_2_a <- c()
for(i in 1:length(models_2_a)){
  model <- models_2_a[[i]]
  pred <- predict(object = model, newdata = x.validate)
  mae_values_2_a[i] <- mae(actual = x.validate$views_product_detail, predicted =
pred)
}

min(mae_values_2_a)
match(min(mae_values_2_a), mae_values_2_a)
hyper_grid[match(min(mae_values_2_a), mae_values_2_a),]
optimal_model_mae_2_a <- models_2_a[[match(min(mae_values_2_a), mae_val-
ues_2_a)]]

#Decision:
#MAE and RMSE lead to same optimal parameters. Thus, same results.
#Test
#RMSE model
pred <- predict(object = optimal_model_rmse_2_a, newdata = x.test)
rmse(actual = x.test$views_product_detail, predicted = pred)
mae(actual = x.test$views_product_detail, predicted = pred)

#(b) with SD
#Training
#Default values
rf_pdp_SD <- randomForest(views_product_detail ~ views_product_overview +
display + cpc + strong_positive + positive + neutral + email +
affiliate + direct + organic + referral + social +
Europe + Asia + Americas + Africa + Oceania +
March + April +

```

```

        weekend + no_weekend +
        morning + afternoon + evening + night +
        familiar +
        promotion_day + no_promotion_day,
        data = x.train)

print(rf_pdp_SD)
plot(rf_pdp_SD)

#Grid search - Hyperparameter tuning
mtry <- seq(6,ncol(x.train)/3,2)
nodesize <- c(10,15)
maxdepth <- seq(20,40,10)
sampsize <- nrow(x.train) * c(0.7,0.8)

hyper_grid <- expand.grid(mtry = mtry, nodesize = nodesize, maxdepth =
maxdepth, sampsize = sampsize)

models_2_b <- list()
for(i in 1:nrow(hyper_grid)){
  mtry <- hyper_grid$mtry[i]
  nodesize <- hyper_grid$nodesize[i]
  maxdepth <- hyper_grid$maxdepth[i]
  sampsize <- hyper_grid$sampsize[i]

  models_2_b[[i]] <- randomForest(views_product_detail ~
        views_product_overview + display + cpc + strong_positive
        + positive + neutral + email + affiliate + direct + organic +
        referral + social +
        Europe + Asia + Americas + Africa + Oceania +
        March + April +
        weekend + no_weekend +
        morning + afternoon + evening + night +
        familiar +
        promotion_day + no_promotion_day,
        mtry = mtry,
        nodesize = nodesize,
        maxdepth = maxdepth,
        sampsize = sampsize,
        data = x.train)
}
#Validation
#RMSE comparison
rmse_values_2_b <- c()
for(i in 1:length(models_2_b)){
  model <- models_2_b[[i]]
  pred <- predict(object = model, newdata = x.validate)
  rmse_values_2_b[i] <- rmse(actual = x.validate$views_product_detail, predicted
= pred)

```

```

}

min(rmse_values_2_b)
match(min(rmse_values_2_b), rmse_values_2_b)
hyper_grid[match(min(rmse_values_2_b), rmse_values_2_b),]
optimal_model_rmse_2_b <- models_2_b[[match(min(rmse_values_2_b),
rmse_values_2_b)]]

#MAE comparison
mae_values_2_b <- c()
for(i in 1:length(models_2_b)){
  model <- models_2_b[[i]]
  pred <- predict(object = model, newdata = x.validate)
  mae_values_2_b[i] <- mae(actual = x.validate$views_product_detail, predicted =
pred)
}

min(mae_values_2_b)
match(min(mae_values_2_b), mae_values_2_b)
hyper_grid[match(min(mae_values_2_b), mae_values_2_b),]
optimal_model_mae_2_b <- models_2_b[[match(min(mae_values_2_b),
mae_values_2_b)]]

#Decision:
#Choose MAE model as higher higher explained variance and lower MAE

#Test
#MAE model
pred <- predict(object = optimal_model_mae_2_b, newdata = x.test)
rmse(actual = x.test$views_product_detail, predicted = pred)
mae(actual = x.test$views_product_detail, predicted = pred)

#Variable Importance
var_imp_pdp <- as.data.frame(importance(optimal_model_mae_2_b))
var_imp_pdp$channels <- rownames(var_imp_pdp)
colnames(var_imp_pdp) <- c("node_purity", "channels")
var_imp_pdp <- var_imp_pdp[order(var_imp_pdp$node_purity, decreasing =
TRUE),]

var_imp_pdp_plot <- ggplot(var_imp_pdp, aes(x=reorder(channels, node_purity),
y=node_purity))+
  geom_bar(stat = "identity")+
  coord_flip()+
  theme(panel.background = element_rect(fill = "white"), panel.grid =
element_line(colour = "grey92"), text = element_text(size=16,
family = "sans"))+
  scale_x_discrete(name="split variables", labels = c("Oceania", "Asia", "Africa",
"Europe", "display", "Americas", "positive", "familiar", "referral",
"strong positive", "night", "affiliate", "social", "neutral", "e-mail", "direct",

```

```

"April", "organic", "morning", "promotion day", "weekend", "cpc", "evening",
"afternoon", "weekday", "March", "no promotion day",
"views product overview"))+
scale_y_continuous(name="node purity", labels = comma)+
ggtitle("Consideration stage")
var_imp_pdp_plot

plot_grid(var_imp_pop_plot, var_imp_pdp_plot, labels = "AUTO")

#Transaction revenue
#(a) without UD
#Training
#Default values
rf_tr_UD <- randomForest(transaction_revenue ~ views_product_overview +
  views_product_detail + display + cpc + email + affiliate + direct
  + organic + referral + social +
  Europe + Asia + Americas + Africa + Oceania +
  March + April +
  weekend + no_weekend +
  morning + afternoon + evening + night +
  familiar +
  promotion_day + no_promotion_day +
  images,
  data = x.train)

print(rf_tr_UD)
plot(rf_ptr_UD)

#Grid search - Hyperparameter tuning
mtry <- seq(6,ncol(x.train)/3,2)
nodesize <- c(10,15)
maxdepth <- seq(20,40,10)
sampsize <- nrow(x.train) * c(0.7,0.8)

hyper_grid <- expand.grid(mtry = mtry, nodesize = nodesize, maxdepth =
maxdepth, sampsize = sampsize)

models_3_a <- list()
for(i in 1:nrow(hyper_grid)){
  mtry <- hyper_grid$mtry[i]
  nodesize <- hyper_grid$nodesize[i]
  maxdepth <- hyper_grid$maxdepth[i]
  sampsize <- hyper_grid$sampsize[i]

  models_3_a[[i]] <- randomForest(transaction_revenue ~
    views_product_overview + views_product_detail + display
    + cpc + email + affiliate + direct + organic + referral +
    social +
    Europe + Asia + Americas + Africa + Oceania +

```

```

    March + April +
    weekend + no_weekend +
    morning + afternoon + evening + night +
    familiar +
    promotion_day + no_promotion_day +
    images,
    mtry = mtry,
    nodesize = nodesize,
    maxdepth = maxdepth,
    sampsize = sampsize,
    data = x.train)
}

#Validation
#RMSE comparison
rmse_values_3_a <- c()
for(i in 1:length(models_3_a)){
  model <- models_3_a[[i]]
  pred <- predict(object = model, newdata = x.validate)
  rmse_values_3_a[i] <- rmse(actual = x.validate$transaction_revenue, predicted =
pred)
}

min(rmse_values_3_a)
match(min(rmse_values_3_a), rmse_values_3_a)
hyper_grid[match(min(rmse_values_3_a), rmse_values_3_a),]
optimal_model_rmse_3_a <- models_3_a[[match(min(rmse_values_3_a),
rmse_values_3_a)]]

#MAE comparison
mae_values_3_a <- c()
for(i in 1:length(models_3_a)){
  model <- models_3_a[[i]]
  pred <- predict(object = model, newdata = x.validate)
  mae_values_3_a[i] <- mae(actual = x.validate$transaction_revenue, predicted =
pred)
}

min(mae_values_3_a)
match(min(mae_values_3_a), mae_values_3_a)
hyper_grid[match(min(mae_values_3_a), mae_values_3_a),]
optimal_model_mae_3_a <- models_3_a[[match(min(mae_values_3_a), mae_val-
ues_3_a)]]

#Decision
#RMSE lead higher explained variance and lower RMSE

#Test
#RMSE model

```

```

pred <- predict(object = optimal_model_rmse_3_a, newdata = x.test)
rmse(actual = x.test$transaction_revenue, predicted = pred)
mae(actual = x.test$transaction_revenue, predicted = pred)

```

#(b) with SD

#Training

#Default values

```

rf_tr_SD <- randomForest(transaction_revenue ~ views_product_overview +
  views_product_detail + display + cpc + strong_positive +
  positive + neutral + email + affiliate + direct + organic + referral
  + social +
  Europe + Asia + Americas + Africa + Oceania +
  March + April +
  weekend + no_weekend +
  morning + afternoon + evening + night +
  familiar +
  promotion_day + no_promotion_day +
  happy_images + mixed_images + neutral_images,
  data = x.train)

```

```

print(rf_tr_SD)

```

```

plot(rf_tr_SD)

```

#Grid search - Hyperparameter tuning

```

mtry <- seq(6,ncol(x.train)/3,2)

```

```

nodesize <- c(10,15)

```

```

maxdepth <- seq(20,40,10)

```

```

sampsize <- nrow(x.train) * c(0.7,0.8)

```

```

hyper_grid <- expand.grid(mtry = mtry, nodesize = nodesize, maxdepth =
maxdepth, sampsize = sampsize)

```

```

models_3_b <- list()

```

```

for(i in 1:nrow(hyper_grid)){
  mtry <- hyper_grid$mtry[i]
  nodesize <- hyper_grid$nodesize[i]
  maxdepth <- hyper_grid$maxdepth[i]
  sampsize <- hyper_grid$sampsize[i]

```

```

  models_3_b[[i]] <- randomForest(transaction_revenue ~
    views_product_overview + views_product_detail + display
    + cpc + strong_positive + positive + neutral + email +
    affiliate + direct + organic + referral + social +
    Europe + Asia + Americas + Africa + Oceania +
    March + April +
    weekend + no_weekend +
    morning + afternoon + evening + night +
    familiar +
    promotion_day + no_promotion_day +

```



```

        happy_images + mixed_images + neutral_images,
        mtry = mtry,
        nodesize = nodesize,
        maxdepth = maxdepth,
        sampsize = sampsize,
        data = x.train)
}

#Validation
#RMSE comparison
rmse_values_3_b <- c()
for(i in 1:length(models_3_b)){
  model <- models_3_b[[i]]
  pred <- predict(object = model, newdata = x.validate)
  rmse_values_3_b[i] <- rmse(actual = x.validate$transaction_revenue, predicted =
pred)
}

min(rmse_values_3_b)
match(min(rmse_values_3_b), rmse_values_3_b)
hyper_grid[match(min(rmse_values_3_b), rmse_values_3_b),]
optimal_model_rmse_3_b <- models_3_b[[match(min(rmse_values_3_b),
rmse_values_3_b)]]

#MAE comparison
mae_values_3_b <- c()
for(i in 1:length(models_3_b)){
  model <- models_3_b[[i]]
  pred <- predict(object = model, newdata = x.validate)
  mae_values_3_b[i] <- mae(actual = x.validate$transaction_revenue, predicted =
pred)
}

min(mae_values_3_b)
match(min(mae_values_3_b), mae_values_3_b)
hyper_grid[match(min(mae_values_3_b), mae_values_3_b),]
optimal_model_mae_3_b <- models_3_b[[match(min(mae_values_3_b),
mae_values_3_b)]]

#Decision
#Choose RMSE model as higher higher explained variance and lower RMSE

#Test
#RMSE model
pred <- predict(object = optimal_model_rmse_3_b, newdata = x.test)
rmse(actual = x.test$transaction_revenue, predicted = pred)
mae(actual = x.test$transaction_revenue, predicted = pred)

```

#Variable Importance

```
var_imp_tr <- as.data.frame(importance(optimal_model_mae_3_a))
var_imp_tr$channels <- rownames(var_imp_tr)
colnames(var_imp_tr) <- c("node_purity", "channels")
var_imp_tr <- var_imp_tr[order(var_imp_tr$node_purity, decreasing = TRUE),]

var_imp_tr_plot <- ggplot(var_imp_tr, aes(x=reorder(channels, node_purity),
y=node_purity))+
  geom_bar(stat = "identity")+
  coord_flip()+
  theme(panel.background = element_rect(fill = "white"),
        panel.grid = element_line(colour = "grey92"), text = element_text(size=16,
        family = "sans"))+
  scale_x_discrete(name="split variables", labels = c("Oceania", "Africa", "Asia",
  "Americas", "Europe", "night", "referral", "display", "affiliate", "April",
  "promotion day", "organic", "cpc", "weekend", "afternoon", "social", "direct",
  "morning", "evening", "March", "e-mail", "familiar", "no weekend",
  "no promotion day", "views product detail", "views product overview",
  "clicks on product images"))+
  scale_y_continuous(name="node purity", labels = comma)+
  ggtitle("Purchase stage (revenue)")
var_imp_tr_plot
```

#Transaction binary

##(a) without UD

#Training

```
x.train[x.train$transactions>1,"transactions"] <- 1
x.train$transactions <- as.factor(x.train$transactions)
```

```
x.validate[x.validate$transactions>1,"transactions"] <- 1
x.validate$transactions <- as.factor(x.validate$transactions)
```

```
x.test[x.test$transactions>1,"transactions"] <- 1
x.test$transactions <- as.factor(x.test$transactions)
```

#Default values

```
rf_tr_UD <- randomForest(transactions ~ views_product_overview +
  views_product_detail + display + cpc + email + affiliate + direct
  + organic + referral + social +
  Europe + Asia + Americas + Africa + Oceania +
  March + April +
  weekend + no_weekend +
  morning + afternoon + evening + night +
  familiar +
  promotion_day + no_promotion_day +
  images,
  data = x.train)
```

```

print(rf_tr_UD)
plot(rf_tr_UD)

#Grid search - Hyperparameter tuning
mtry <- seq(6,ncol(x.train)/3,2)
nodesize <- c(10,15)
maxdepth <- seq(20,40,10)
sampsize <- nrow(x.train) * c(0.7,0.8)

hyper_grid <- expand.grid(mtry = mtry, nodesize = nodesize, maxdepth =
maxdepth, sampsize = sampsize)

models_4_a <- list()
for(i in 1:nrow(hyper_grid)){
  mtry <- hyper_grid$mtry[i]
  nodesize <- hyper_grid$nodesize[i]
  maxdepth <- hyper_grid$maxdepth[i]
  sampsize <- hyper_grid$sampsize[i]

  models_4_a[[i]] <- randomForest(transactions ~ views_product_overview +
                                views_product_detail + display + cpc + email + affiliate +
                                direct + organic + referral + social +
                                Europe + Asia + Americas + Africa + Oceania +
                                March + April +
                                weekend + no_weekend +
                                morning + afternoon + evening + night +
                                familiar +
                                promotion_day + no_promotion_day +
                                images,
                                mtry = mtry,
                                nodesize = nodesize,
                                maxdepth = maxdepth,
                                sampsize = sampsize,
                                data = x.train)
}

#Validation
confusion_matrix_4_a <- list()
for(i in seq_along(models_4_a)){
  model <- models_4_a[[i]]
  pred <- predict(model, newdata=x.validate,type = "class")
  confusion_matrix_4_a[[i]] <- confusionMatrix(pred,x.validate$transactions)
}

accuracy_4_a <- c()
for(i in seq_along(models_4_a)){
  accuracy_4_a[i] <- confusion_matrix_4_a[[i]]$overall[1]
}

```

```
match(max(accuracy_4_a), accuracy_4_a)
optimal_model_4_a <- models_4_a[[match(max(accuracy_4_a), accuracy_4_a)]]
```

```
#Test
```

```
pred <- predict(object = optimal_model_4_a, newdata = x.test, type = "class")
confusionMatrix(pred, x.test$transactions)
```

```
 #(b) with SD
```

```
#Training
```

```
#Default values
```

```
rf_tr_SD <- randomForest(transactions ~ views_product_overview +
  views_product_detail + display + cpc + strong_positive +
  positive + neutral + email + affiliate + direct + organic + referral
  + social +
  Europe + Asia + Americas + Africa + Oceania +
  March + April +
  weekend + no_weekend +
  morning + afternoon + evening + night +
  familiar +
  promotion_day + no_promotion_day +
  happy_images + mixed_images + neutral_images,
  data = x.train)
```

```
print(rf_tr_SD)
```

```
plot(rf_tr_SD)
```

```
#Grid search - Hyperparameter tuning
```

```
mtry <- seq(6, ncol(x.train)/3, 2)
```

```
nodesize <- c(10, 15)
```

```
maxdepth <- seq(20, 40, 10)
```

```
samplesize <- nrow(x.train) * c(0.7, 0.8)
```

```
hyper_grid <- expand.grid(mtry = mtry, nodesize = nodesize, maxdepth =
  maxdepth, samplesize = samplesize)
```

```
models_4_b <- list()
```

```
for(i in 1:nrow(hyper_grid)){
  mtry <- hyper_grid$mtry[i]
  nodesize <- hyper_grid$nodesize[i]
  maxdepth <- hyper_grid$maxdepth[i]
  samplesize <- hyper_grid$samplesize[i]
```

```
models_4_b[[i]] <- randomForest(transactions ~ views_product_overview +
  views_product_detail + display + cpc + strong_positive +
  positive + neutral + email + affiliate + direct + organic +
  referral + social +
  Europe + Asia + Americas + Africa + Oceania +
  March + April +
  weekend + no_weekend +
```

```

        morning + afternoon + evening + night +
        familiar +
        promotion_day + no_promotion_day +
        happy_images + mixed_images + neutral_images,
        mtry = mtry,
        nodesize = nodesize,
        maxdepth = maxdepth,
        sampsize = sampsize,
        data = x.train)
}

#Validation
confusion_matrix_4_b <- list()
for(i in seq_along(models_4_b)){
  model <- models_4_b[[i]]
  pred <- predict(model, newdata=x.validate,type = "class")
  confusion_matrix_4_b[[i]] <- confusionMatrix(pred,x.validate$transactions)
}

accuracy_4_b <- c()
for(i in seq_along(models_4_b)){
  accuracy_4_b[i] <- confusion_matrix_4_b[[i]]$overall[1]
}

match(max(accuracy_4_b), accuracy_4_b)
optimal_model_4_b <- models_4_b[[match(max(accuracy_4_b), accuracy_4_b)]]

#Test
pred <- predict(object = optimal_model_4_b, newdata = x.test, type = "class")
confusionMatrix(pred, x.test$transactions)

#Variable Importance
var_imp_tb <- as.data.frame(importance(optimal_model_4_a))
var_imp_tb$channels <- rownames(var_imp_tb)
colnames(var_imp_tb) <- c("mean_decrease_GINI", "channels")
var_imp_tb <- var_imp_tb[order(var_imp_tb$mean_decrease_GINI, decreasing =
TRUE),]

var_imp_tb_plot <- ggplot(var_imp_tb, aes(x=reorder(channels, mean_de-
crease_GINI), y=mean_decrease_GINI))+
  geom_bar(stat = "identity")+
  coord_flip()+
  theme(panel.background = element_rect(fill = "white"),
        panel.grid = element_line(colour = "grey92"), text = element_text(size=16,
family = "sans"))+
  scale_y_continuous(name="mean decrease Gini", labels = comma)+
  scale_x_discrete(name="split variables", labels = c("Oceania", "Africa", "Asia",
"Europe", "Americas", "display", "referral", "night", "social", "affiliate",
"familiar", "e-mail", "promotion day", "organic", "direct", "morning",

```

```

"afternoon", "weekend", "cpc", "evening", "April", "March", "no weekend",
"no promotion day", "clicks on product images", "views product detail",
"views product overview"))+
ggtitle("Purchase stage (decision)")
var_imp_tb_plot

#Combined Plots Variable Importance
plot_grid(var_imp_pop_plot, var_imp_pdp_plot, labels = "AUTO")
plot_grid(var_imp_tr_plot, var_imp_tb_plot, labels = c("C", "D"))
plot_grid(var_imp_pop_plot, var_imp_pdp_plot, var_imp_tr_plot,
var_imp_tb_plot, labels = "AUTO")

# Shapley Value - Channel attribution -----
## Use model to predict product overview pages, product detail pages and trans-
action revenue - final dataset 2
#Product overview page clicks
final_dataset_2$pop <- predict(object = optimal_model_mae_1_b, newdata = fi-
nal_dataset_2)
final_dataset_2$pop_margin <- final_dataset_2$pop

for(i in 2:length(final_dataset_2$user_id)){
  if(final_dataset_2$user_id[i]==final_dataset_2$user_id[i-1]){
    final_dataset_2$pop_margin[i] <- final_dataset_2$pop[i]-final_da-
taset_2$pop[i-1]
  }
}

#Channels in general
value_pop <- c()
for(i in seq_along(independent_variables)){
  value_pop[i] <- as.vector(final_dataset_2 %>%
    filter(marketing_channel==independent_variables[i]) %>%
    select(pop_margin) %>%
    summarise(sum_pop=sum(pop_margin)))
}

value_pop <- as.data.frame(value_pop)
colnames(value_pop) <- independent_variables
value_pop

#Strong positive, positive, neutral
strong_positive_value_pop <- sum(final_dataset_2 %>% filter(strong_posi-
tive_help==1) %>% select(pop_margin))
positive_value_pop <- sum(final_dataset_2 %>% filter(positive_help==1) %>%
select(pop_margin))
neutral_value_pop <- sum(final_dataset_2 %>% filter(neutral_help==1) %>% se-
lect(pop_margin))

```

```

#Assess efficiency
#(a)
value_pop <- t(value_pop)
value_pop <- as.data.frame(value_pop)
colnames(value_pop) <- "value"

value_pop$frequency <- 0
for(i in seq_along(independent_variables)){
  value_pop$frequency[i] <- sum(final_dataset_2$marketing_channel==independent_variables[i])
}

value_pop <- value_pop %>% mutate(shapley_value_pop=value/frequency)
rownames(value_pop) <- independent_variables
value_pop

#(b)
shapley_value_strong_positive_pop <- strong_positive_value_pop/nrow(final_dataset_2 %>% filter(strong_positive_help==1))
shapley_value_positive_pop <- positive_value_pop/nrow(final_dataset_2 %>% filter(positive_help==1))
shapley_value_neutral_pop <- neutral_value_pop/nrow(final_dataset_2 %>% filter(neutral_help==1))

shapley_vaule_ad_description_pop <- data.frame(shapley_value_strong_positive_pop, shapley_value_positive_pop, shapley_value_neutral_pop)
colnames(shapley_vaule_ad_description_pop) <- c("strong positive", "positive", "neutral")
shapley_vaule_ad_description_pop <- t(shapley_vaule_ad_description_pop)
colnames(shapley_vaule_ad_description_pop) <- c("shapley_value")
shapley_vaule_ad_description_pop <- as.data.frame(shapley_vaule_ad_description_pop)

#Plot
shapley_value_pop_bar <- ggplot(value_pop, aes(x=reorder(rownames(value_pop),shapley_value_pop), y=shapley_value_pop))+
  geom_bar(stat = "identity")+
  theme(panel.background = element_rect(fill = "white"),
    panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))+
  xlab("channel")+
  ylab("shapley value")+
  ggtitle("Awareness stage")
shapley_value_pop_bar

shapley_value_pop_2_bar <- ggplot(shapley_vaule_ad_description_pop,
  aes(x=reorder(rownames(shapley_vaule_ad_description_pop),shapley_value),
  y=shapley_value))+
  geom_bar(stat = "identity")+

```

```

theme(panel.background = element_rect(fill = "white"),
panel.grid = element_line(colour = "grey92"), text = element_text(size=16,
family = "sans"))+
xlab("cpc")+
ylab("shapley value")+
ggtitle("Awareness stage")
shapley_value_pop_2_bar

plot_grid(shapley_value_pop_bar, shapley_value_pop_2_bar, labels = "AUTO")

#Product detail page clicks
final_dataset_2$pdp <- predict(object = optimal_model_mae_2_b, newdata = fi-
nal_dataset_2)
final_dataset_2$pdp_margin <- final_dataset_2$pdp

for(i in 2:length(final_dataset_2$user_id)){
  if(final_dataset_2$user_id[i]==final_dataset_2$user_id[i-1]){
    final_dataset_2$pdp_margin[i] <- final_dataset_2$pdp[i]-final_da-
taset_2$pdp[i-1]
  }
}

#Channels in general
value_pdp <- c()
for(i in seq_along(independent_variables)){
  value_pdp[i] <- as.vector(final_dataset_2 %>%
    filter(marketing_channel==independent_variables[i]) %>%
    select(pdp_margin) %>%
    summarise(sum_rev=sum(pdp_margin)))
}

value_pdp <- as.data.frame(value_pdp)
colnames(value_pdp) <- independent_variables
value_pdp

#Strong positive, positive, neutral
strong_positive_value_pdp <- sum(final_dataset_2 %>% filter(strong_posi-
tive_help==1) %>% select(pdp_margin))
positive_value_pdp <- sum(final_dataset_2 %>% filter(positive_help==1) %>%
select(pdp_margin))
neutral_value_pdp <- sum(final_dataset_2 %>% filter(neutral_help==1) %>% se-
lect(pdp_margin))

#Assess efficiency
#(a)
value_pdp <- t(value_pdp)
value_pdp <- as.data.frame(value_pdp)
colnames(value_pdp) <- "value"

```



```

value_pdp$frequency <- 0
for(i in seq_along(independent_variables)){
  value_pdp$frequency[i] <- sum(final_dataset_2$marketing_channel==independent_variables[i])
}

```

```

value_pdp <- value_pdp %>% mutate(shapley_value_pdp=value/frequency)
rownames(value_pdp) <- independent_variables
value_pdp

```

#(b)

```

shapley_value_strong_positive_pdp <- strong_positive_value_pdp/nrow(final_dataset_2 %>% filter(strong_positive_help==1))
shapley_value_positive_pdp <- positive_value_pdp/nrow(final_dataset_2 %>% filter(positive_help==1))
shapley_value_neutral_pdp <- neutral_value_pdp/nrow(final_dataset_2 %>% filter(neutral_help==1))

```

```

shapley_vaule_ad_description_pdp <- data.frame(shapley_value_strong_positive_pdp, shapley_value_positive_pdp, shapley_value_neutral_pdp)
colnames(shapley_vaule_ad_description_pdp) <- c("strong positive", "positive", "neutral")
shapley_vaule_ad_description_pdp <- t(shapley_vaule_ad_description_pdp)
colnames(shapley_vaule_ad_description_pdp) <- c("shapley_value")
shapley_vaule_ad_description_pdp <- as.data.frame(shapley_vaule_ad_description_pdp)

```

#Plots

```

shapley_value_pdp_bar <- ggplot(value_pdp, aes(x=reorder(rownames(value_pdp),shapley_value_pdp), y=shapley_value_pdp))+
  geom_bar(stat = "identity")+
  theme(panel.background = element_rect(fill = "white"),
    panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))+
  xlab("channel")+
  ylab("shapley value")+
  ggtitle("Consideration stage")
shapley_value_pdp_bar

```

```

shapley_value_pdp_2_bar <- ggplot(shapley_vaule_ad_description_pdp,
  aes(x=reorder(rownames(shapley_vaule_ad_description_pdp),shapley_value),
  y=shapley_value))+
  geom_bar(stat = "identity")+
  theme(panel.background = element_rect(fill = "white"),
    panel.grid = element_line(colour = "grey92"), text = element_text(size=16, family = "sans"))+
  xlab("cpc")+
  ylab("shapley value")+
  ggtitle("Consideration stage")

```

```
shapley_value_pdp_2_bar
```

```
plot_grid(shapley_value_pdp_bar, shapley_value_pdp_2_bar, labels = "AUTO")
```

```
#Transaction revenue
```

```
final_dataset_2$tr <- predict(object = optimal_model_mae_3_a, newdata = fi-  
nal_dataset_2)
```

```
final_dataset_2[final_dataset_2$tr<0,"tr"] <- 0
```

```
final_dataset_2$tr_margin <- final_dataset_2$tr
```

```
for(i in 2:length(final_dataset_2$user_id)){  
  if(final_dataset_2$user_id[i]==final_dataset_2$user_id[i-1]){  
    final_dataset_2$tr_margin[i] <- final_dataset_2$tr[i]-final_dataset_2$tr[i-1]  
  }  
}
```

```
#Channels in general
```

```
value_tr <- c()
```

```
for(i in seq_along(independent_variables)){  
  value_tr[i] <- as.vector(final_dataset_2 %>%  
    filter(marketing_channel==independent_variables[i]) %>%  
    select(tr_margin) %>%  
    summarise(sum_rev=sum(tr_margin)))  
}
```

```
value_tr <- as.data.frame(value_tr)
```

```
colnames(value_tr) <- independent_variables
```

```
value_tr
```

```
#Strong positive, positive, neutral
```

```
strong_positive_value_tr <- sum(final_dataset_2 %>% filter(strong_posi-  
tive_help==1) %>% select(tr_margin))
```

```
positive_value_tr <- sum(final_dataset_2 %>% filter(positive_help==1) %>% se-  
lect(tr_margin))
```

```
neutral_value_tr <- sum(final_dataset_2 %>% filter(neutral_help==1) %>% se-  
lect(tr_margin))
```

```
#Assess efficiency
```

```
##(a)
```

```
value_tr <- t(value_tr)
```

```
value_tr <- as.data.frame(value_tr)
```

```
colnames(value_tr) <- "value"
```

```
value_tr$frequency <- 0
```

```
for(i in seq_along(independent_variables)){  
  value_tr$frequency[i] <- sum(final_dataset_2$marketing_channel==independ-  
ent_variables[i])
```

```

}

value_tr <- value_tr %>% mutate(shapley_value_tr=value/frequency)
rownames(value_tr) <- independent_variables
value_tr

#(b)
shapley_value_strong_positive_tr <- strong_positive_value_tr/nrow(final_dataset_2 %>% filter(strong_positive_help==1))
shapley_value_positive_tr <- positive_value_tr/nrow(final_dataset_2 %>% filter(positive_help==1))
shapley_value_neutral_tr <- neutral_value_tr/nrow(final_dataset_2 %>% filter(neutral_help==1))

shapley_vaule_ad_description_tr <- data.frame(shapley_value_strong_positive_tr,
shapley_value_positive_tr, shapley_value_neutral_tr)
colnames(shapley_vaule_ad_description_tr) <- c("strong positive", "positive",
"neutral")
shapley_vaule_ad_description_tr <- t(shapley_vaule_ad_description_tr)
colnames(shapley_vaule_ad_description_tr) <- c("shapley_value")
shapley_vaule_ad_description_tr <- as.data.frame(shapley_vaule_ad_description_tr)

#Plots
shapley_value_tr_bar <- ggplot(value_tr, aes(x=reorder(rownames(value_tr),shapley_value_tr), y=shapley_value_tr))+
  geom_bar(stat = "identity")+
  theme(panel.background = element_rect(fill = "white"),
  panel.grid = element_line(colour = "grey92"), text = element_text(size=16,
  family = "sans"))+
  xlab("channel")+
  ylab("shapley value")+
  ggtitle("Purchase stage (revenue)")
shapley_value_tr_bar

shapley_value_tr_2_bar <- ggplot(shapley_vaule_ad_description_tr, aes(x=reorder(rownames(shapley_vaule_ad_description_tr),shapley_value), y=shapley_value))+
  geom_bar(stat = "identity")+
  theme(panel.background = element_rect(fill = "white"),
  panel.grid = element_line(colour = "grey92"), text = element_text(size=16,
  family = "sans"))+
  xlab("cpc")+
  ylab("shapley value")+
  ggtitle("Purchase stage (revenue)")
shapley_value_tr_2_bar

#Transaction binary

```

```

tb <- predict(object = optimal_model_4_a, type = "prob",newdata = final_da-
taset_2)
tb <- as.data.frame(tb)
final_dataset_2$tb<- tb`1`

final_dataset_2$tb_margin <- final_dataset_2$tb

for(i in 2:length(final_dataset_2$user_id)){
  if(final_dataset_2$user_id[i]==final_dataset_2$user_id[i-1]){
    final_dataset_2$tb_margin[i] <- final_dataset_2$tb[i]-final_dataset_2$tb[i-1]
  }
}

#Channels in general
value_tb <- c()
for(i in seq_along(independent_variables)){
  value_tb[i] <- as.vector(final_dataset_2 %>%
    filter(marketing_channel==independent_variables[i]) %>%
    select(tb_margin) %>%
    summarise(sum_rev=sum(tb_margin)))
}

value_tb <- as.data.frame(value_tb)
colnames(value_tb) <- independent_variables
value_tb

#Strong positive, positive, neutral
strong_positive_value_tb <- sum(final_dataset_2 %>% filter(strong_posi-
tive_help==1) %>% select(tb_margin))
positive_value_tb <- sum(final_dataset_2 %>% filter(positive_help==1) %>% se-
lect(tb_margin))
neutral_value_tb <- sum(final_dataset_2 %>% filter(neutral_help==1) %>% se-
lect(tb_margin))

#Assess efficiency
#(a)
value_tb <- t(value_tb)
value_tb <- as.data.frame(value_tb)
colnames(value_tb) <- "value"

value_tb$frequency <- 0
for(i in seq_along(independent_variables)){
  value_tb$frequency[i] <- sum(final_dataset_2$marketing_channel==independ-
ent_variables[i])
}

value_tb <- value_tb %>% mutate(shapley_value_tb=value/frequency)
rownames(value_tb) <- independent_variables
value_tb

```

```

#(b)
shapley_value_strong_positive_tb <- strong_positive_value_tb/nrow(final_da-
taset_2 %>% filter(strong_positive_help==1))
shapley_value_positive_tb <- positive_value_tb/nrow(final_dataset_2 %>% fil-
ter(positive_help==1))
shapley_value_neutral_tb <- neutral_value_tb/nrow(final_dataset_2 %>% fil-
ter(neutral_help==1))

```

```

shapley_vaule_ad_description_tb <- data.frame(shapley_value_strong_posi-
tive_tb, shapley_value_positive_tb, shapley_value_neutral_tb)
colnames(shapley_vaule_ad_description_tb) <- c("strong positive", "positive",
"neutral")
shapley_vaule_ad_description_tb <- t(shapley_vaule_ad_description_tb)
colnames(shapley_vaule_ad_description_tb) <- c("shapley_value")
shapley_vaule_ad_description_tb <- as.data.frame(shapley_vaule_ad_descrip-
tion_tb)

```

#Plots

```

shapley_value_tb_bar <- ggplot(value_tb, aes(x=reor-
der(rownames(value_tb),shapley_value_tb), y=shapley_value_tb))+
  geom_bar(stat = "identity")+
  theme(panel.background = element_rect(fill = "white"),
  panel.grid = element_line(colour = "grey92"), text = element_text(size=16,
  family = "sans"))+
  xlab("channel")+
  ylab("shapley value")+
  ggtitle("Purchase stage (decision)")
shapley_value_tb_bar

```

```

shapley_value_tb_2_bar <- ggplot(shapley_vaule_ad_description_tb,
  aes(x=reorder(rownames(shapley_vaule_ad_description_tb),shapley_value),
  y=shapley_value))+
  geom_bar(stat = "identity")+
  theme(panel.background = element_rect(fill = "white"),
  panel.grid = element_line(colour = "grey92"), text = element_text(size=16,
  family = "sans"))+
  xlab("cpc")+
  ylab("shapley value")+
  ggtitle("Purchase stage (decision)")
shapley_value_tb_2_bar

```

```

plot_grid(shapley_value_pop_bar, shapley_value_pop_2_bar, shap-
ley_value_pdp_bar, shapley_value_pdp_2_bar, labels = "AUTO")
plot_grid(shapley_value_tr_bar, shapley_value_tr_2_bar, shapley_value_tb_bar,
shapley_value_tb_2_bar, labels = c("E", "F", "G", "H"))

```

I hereby certify that I have composed my master thesis

Leveraging unstructured data in channel attribution theory

independently and without outside help, and that I have specially marked and quoted the sources of all the passages taken literally by other authors, as well as all remarks that closely follow the thought of other authors.

Groningen, June 17, 2019

Felix Lehmkuhle

(signature)