ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics
Master Thesis Data Science & Marketing Analytics

Evaluating the feasibility of supervised machine learning models for multi-touch attribution.

Danny Samuel van Tol
697647

Supervisor: dr. K Gruber
Second accessor: dr. CS Bellet

Data final version: 1-8-2024

Abstract

This research has used real-world e-commerce data to implement heuristic and data-driven attribution models. These models have been compared to demonstrate the most efficient and best-performing models that can be utilized for customer journey analysis. The results of this research have indicated that of the heuristic attribution models, the time-decay model has shown to have the highest conversion rate and, therefore, the highest performance. Using this heuristic model would be recommended if implementing data-driven models is not possible. The data-driven models have all performed well, although some are better. The bagged logistic regression is shown to be the worst performing, followed by the pre-determined random forest model. One of the best-performing attribution models is the fine-tuned random forest model. The most efficient customer journey analysis model is the recurrent neural network model with long short-term memory architecture. This recurrent neural network will provide the best results as it can handle time-oriented features common in customer journey analysis.

Table of Contents

1 Introduction

Digital advertising is an essential element of digital marketing that aims to reach customers through digital channels regarding a product, brand, or service. The digital marketing environment is becoming increasingly important as it has embedded itself into the communication strategies of virtually all companies that partake in advertising (McStay, A., 2017).

Billions of dollars worldwide are invested in digital advertising mediums. However, the maximum allocative efficiency has not yet been achieved, which can be attributed to marketing inefficiencies. A multitude of causes creates these inefficiencies, one being the ad effect measurement, which is an incorrect estimation of the incremental effects that advertisements have based on the behavior of consumers. This can lead to a distorted expectation in which firms assume future demand is higher or lower than it will be, causing an inefficient budget allocation, which results in losses. Advertisers view this ad measurement effect as the leading concern, indicating the current situation's severity (Gordon et al., 2020).

In advertisement campaigns, users can be exposed to multiple channels, which could lead to a conversion. Measuring the precise effect each channel had on the conversion is where research becomes complex. Traditional attribution models generally fail to measure the channel effects as these models are based on simple heuristics that make rule-based decisions instead of conducting statistical analysis on the underlying data. These traditional attribution models handling data in such a manner could cause the model to fail to capture the actual influence of each touchpoint in the advertising campaign. Only 1 in 10 companies regard the traditional last-touch attribution model as efficient. However, 50% of the companies in 2016 still used this inefficient method for budget allocation (Thurber, 2016). Therefore, more sophisticated attribution models must be used to attribute credits to multiple channels (Shao & Li, 2011).

The following research starts with a literature review in which the main concept of the customer journey and the models that can be utilized for this journey will be analyzed. The heuristic models, which are the single-touch, last-touch, linear, time-decay, and the data-driven models being the bagged logistic regression, random forest, and recurrent neural network, demonstrate their relevance and success after implementation in prior research and will therefore be used for this research. The performance metrics and interpretation methods necessary to evaluate the models on the data will be explored in the last section of the literature review. The following section, data will discuss general information about the dataset, how the data has been transformed to be properly used and the process of how multiple datasets have been merged to create the data on which the models will operate. The data section in addition will demonstrate the distribution of the target variable in the data and the correlation plots of the predictors before and after balancing. The outcome of implementing the mentioned models will be demonstrated

graphically and in detail in the results. The results will elaborate on that the time-decay is the best-performing heuristic model as it has the highest conversion rate and that the neural network with long short-term memory (LSTM) architecture is the best-performing data-driven model as it has the highest accuracy, specificity, and overall performance. The results are followed by the conclusion, in which the most important findings are demonstrated and compared to prior research. This section will additionally answer the research question. The last section of the main body of the research discusses the limitations of the current research and future research recommendations.

The main body of the research will be followed by the appendix, which starts with a reference list of all the literature that has been consulted. This includes additional plots and information useful for reproducing this research's results.

This research promises a brighter future for advertisers. It demonstrates which models yield the highest return on investment, offers transparency in the results delivered by each medium, and maximizes the ad effect measurement. By showcasing the actual effect of each ad channel, this research can guide advertisers toward more efficient and effective digital advertising strategies. The following research question will be answered to achieve this promise: *Evaluating the feasibility of supervised machine learning models for multi-touch attribution.*

This research and its findings will be valuable for most companies engaged in digital advertising, particularly those utilizing multiple channels. The insights gained from this study's results will enhance managerial decision-making by providing a comprehensive understanding of simple and complex attribution models using real-world data.

2 Literature review

2.1 Customer Journey

The customer journey is generally used as a metaphor for taking a customer's perspective while interacting with a company or product (Halvorsrud & Kvale, 2017).
It can use used as an analysis method from which insights can be gained regarding the behavior of customers in various touchpoints as the customer journey describes the path a customer takes while interacting with a product or service. This method therefore can analyze the impact of individual advertisements on the customer behavior with the use of attribution or other data-driven models (Koch et al., 2023).

The findings of the existing recent and relevant literature regarding customer experience and journey analysis are essential for understanding the strengths and weaknesses of using certain models for

specific datasets and how, in general, particular models perform while being implemented.

Generally customer journey literature has been proliferating; however, the current literature regarding this subject appears incoherent due to its diverse theoretical background (Tueanrat, Papagiannidis, & Alamanos, 2021). The customer journey can be described as a multidimensional construct that focuses on a customer's cognitive, emotional, sensory, behavioral, and social responses to a particular firm's offering during the customer's path to purchase. Customer journey analysis is considered a greenfield, and the research on this subject is relatively new (Lemon & Verhoef, 2016).

The following articles based on attribution models have been chosen from existing literature that has implemented and tested these models on channel data. These models will be compared and critically evaluated for their efficiency and relevancy. This process will contribute to the literature regarding the customer journey analysis, as only the most relevant and useful models will be considered.

2.2 Heuristic Attribution Models

Starting with single-touch attribution models like first-touch and last-touch. The article by De Haan et al. (2016) found that using a simple attribution model like last-click attribution is a popular method. However, such attribution models overlook the influence and synergy of all other channels in the customer journey. The article suggests that multi-touch attribution channels are preferred as they capture the synergy of channels.

An article by Berman (2018) found that in many cases, the last-touch attribution model even lowers profits for global advertisers compared to not using it at all. The familiarity with stakeholders is indicated as the only advantage of the last-touch attribution model. This article suggests that an alternative model that uses Shapely values, which give a less extreme credit allocation than a last-touch attribution model, increases profits for advertisers.

Another article by Leguina et al. (2020) includes the advantages of first- and last-touch attribution. This article suggests that using first- and last-touch attribution models is not computationally expensive and is beneficial in its simplicity. Additionally, it demonstrates that the first-touch attribution model emphasizes the initial interaction, which can help understand how customers are initially drawn to a brand.

The time-decay and linear attribution models assign credits to multiple touchpoints in the customer journey. Therefore, these models recognize the importance of not just the first and last touchpoints as in the single-touch attribution models.

According to an article by Nisar and Yeung (2017), the time-decay attribution model adjusts credit

so that the closer a touchpoint in the customer journey is to conversion, the more credit it receives.

The article by Leguina et al. (2020) found that the touchpoints closest to conversion impact the model's accuracy the most and are, therefore, essential to consider when building data-driven attribution models.

Another article by Sakly and Ouazan (2016) demonstrates that rule-based time-decay and linear attribution models are more sophisticated than single-touch attribution models but still tend to make bold assumptions that could corrupt the model's output. In an article by Yuvaraj et al. (2018), the cause of this corrupted output is demonstrated as the article indicates that the the linear attribution model assumes that all channels a customer interacts with have contributed equally to the user's total conversion. The article by Leguina et al. (2020), again but regarding the linear model, states that the linear attribution model has performance limitations based on the obtained results, the reasoning being that the touchpoints closer to conversion should be assigned more credits instead of an equal amount to all touchpoints. An article by Gaur et al. (2024) states that the linear attribution model lacks adaptability to the dynamic nature of customers' browsing patterns, rendering them static compared to data-driven attribution models. However, the article states that the linear attribution model is still the most frequently used by marketers to distribute the budget among channels with the last-touch and first-touch models, even though these models are unreliable.

According to the mentioned articles, the time-decay attribution model seems most helpful in conducting research, even though it is still not as reliable as data-driven attribution models. However, since marketers most frequently use the first-touch, last-touch, and linear attribution models, it is deemed essential to use all these heuristic models to answer the research question, as these heuristic models can be used as a benchmark to compare with the more sophisticated data-driven attribution models.

2.3 Bagged Logistic Regression Model

A data-driven model shown to be superior for customer journey analysis, in contrast to the heuristic models mentioned, is the bagged logistic regression model. This model has been proposed and used in an article by Shao and Li (2011). In this article, it has been discussed that using a bagged logistic regression for attribution modeling shows an equal amount of considerable classification accuracy compared to a standard logistic regression. Both models, therefore, have a high accuracy regarding classification. The added benefit of the bagged logistic regression is an increased stability regarding the estimates for individual advertising channel contributions.

The disadvantages of bagged logistic regression, though, are numerous. According to the article, the model performs well regarding user classification in a customer journey but lacks the interpretive aspect

specifically of the customer journey, which is crucial for attribution. Additionally, the article indicates that tree-based methods outperform the bagged logistic regression. Another article regarding the logistic regression model by Wu et al. (2019) states that currently, random forest and the logistic regression are popular models that are applied to customer journey data. However, it seems that this bagged logistic model has only been properly and generally knowingly been applied once. The articles of Kannan et al. (2016), Kadyrov and Ignatov (2019), and lastly, the most recent article from Yang, Dyer, and Wang (2020) only mention the bagged logistic model being applied in the article of Shao and Li (2011) therefore this model seems to be underrepresented in the research regarding customer journey analysis.

2.4 Random Forest

Efron's article (2020) states that random forest is an efficient model for large datasets containing data regarding automation tasks like online shopping.

High accuracy is essential when classifying the contribution of each channel to the conversion. A firm's budget allocation could be based on this information, maximizing the ad measurement effect. Random Forest is regarded as a model with higher accuracy than the traditional heuristic models previously mentioned and could very well have higher accuracy than the bagged logistic regression.

The article by Churchill et al. (2024) praises the random forest for outperforming traditional models regarding predictive accuracy and all models in customer journey analysis in predicting negatives. The neural network outperforms the random forest model in terms of balanced accuracy in the article.

A third article by Nygård and Mezei (2020) indicates that the random forest model should be included in customer journey analysis as it is robust and has superior predictive performance. This model solves the complications of other tree-based models like the bagging model as random forest creates random subsets of predictors that are used for building trees, which helps in de-correlating trees individually, reducing the overall variance, as stated in an article by Hegde, Wallace, and Gray (2015). According to the article of Nygård and Mezei (2020), random forest has a greater general performance than logistic regression and neural network models. This statement of the random forest performing better than the neural network model contradicts the article of Churchill et al. (2024) regarding the general performance of random forest and neural networks in customer journey analysis. An additional benefit of random forest for attributing credits to channels is its ability to automatically estimate the synergetic effects between channels, as indicated by the article of Sinha, Saini, and Anadhavelu (2014).

2.5 Neural Networks & Long Short-Term Memory

The Neural Network attribution model has shown, as claimed by an article by Churchill et al. (2024), that neural networks can be used for customer journey analysis as this model enables firms to measure the relative importance of each type of touchpoint, and it can quantify the impact of each touchpoint within a touchpoint type. The model is efficient in handling high-dimensional data and delivering superior predictive accuracy. The neural network emerged from numerous models tested as the model with the highest area under the roc curve value, indicating its superior predictive accuracy as shown by the results of Churchill et al. (2024).

A neural network model that has been improved upon in terms of customer journey analysis is, according to an article by Li, Arava, Dong, Yan, and Pani (2018), the deep neural network model. This model is refined by incorporating the LSTM architecture. This integrated architecture assists in capturing the underlying hidden complex action patterns of the customer journey.

An article by Lang and Rettenmeier (2017) indicates that the Recurrent Neural Network (RNN) is a natural fit for modeling and predicting customer behavior data as this type of data is sequential in nature. The model has shown higher predictive accuracy than logistic regression. The benefit of an RNN model over other neural network models is its ability to predict customer behavior without feature engineering. It can predict individual customer's current and future behavior, which can be advantageous for efficient budget allocation in the present and future. However, the article does propose that improved model architecture could provide a more promising future for the use of the model.

This RNN model can be further improved using the previously mentioned RNN architecture LSTM. According to an article by Le and Zuidema (2016), RNN models without an LSTM architecture have significant problems with exploding or vanishing gradients. This problem arises when error signals propagating from the root in a parse tree to the child nodes shrink quickly, causing difficulties in capturing long-term dependencies. This, in turn, could decrease the learning efficiency of the RNN model. The RNN model in Lang and Rettenmeier (2017) could have been more efficient if it had the LSTM architecture.

According to Le and Zuidema (2016), the LSTM architecture is superior to the standard RNN architecture in overcoming the vanishing gradient problem and capturing long-term dependencies. Regarding LSTM in attribution modeling for customer journeys, an article by Kindbom (2021) indicates that LSTM is especially advantageous in situations with large amounts of data and longer customer journeys and will, in this regard, perform better than simple probabilistic, logistic regression and last-touch attribution models. This article additionally states that using real-world data for customer journey analysis with LSTM and additional performance metrics could provide interesting results, as using LSTM has been a relatively unexplored model type for attribution modeling.

However, Wu et al. (2019) state that the logistic and random forest models score better regarding the AUC score than the LSTM model, but both the random forest and the LSTM model perform better in general compared to the logistic regression. The bagged logistic model is indicated to be better performing than the logistic regression but less efficient than the random forest and LSTM models.

The statements contradicting each other regarding whether a random forest or neural network has higher predictive accuracy will create an additional opportunity to demonstrate which model is most efficient for channel data.

The articles regarding RNN and LSTM indicate that this combination of deep learning will yield a sophisticated model for attribution modeling. The statement that the RNN, combined with LSTM, is relatively unexplored regarding attribution modeling and with real-world data will fill the current research gap on this topic and provide meaningful results to the scientific community.

2.6 SHAP Values & Markov Model

The article of Kindbom (2021) regarding LSTM in customer journey analysis states additionally that deep learning models like LSTM have been criticized for being difficult to interpret. A solution to this is to use the SHapley Additive exPlanations (SHAP) values. The SHAP values aid in interpreting predictions of any machine learning model (Lundberg & Lee, 2017). An article by Yang, Dyer, and Wang (2020) solves the issue stated by Kindbom (2021) by demonstrating that SHAP values, which can be applied to the features of a machine learning model, utilize the concepts of both Shapely values and Lime and is specifically catered to explaining machine learning models. Therefore, SHAP values could provide more valuable results than only Shapely or Lime values. The article by Merikanto (2022) is in line with the article of Yang, Dyer, and Wang (2020 and states that the SHAP values can be used to explain the effects of customer attributes regarding conversions. These values can be applied to any machine learning model to interpret predictions.

A model that is used in particular articles regarding customer journey analysis that will not be incorporated in this research is the Markov Chain Model. This model has shown its use in specific situations in customer journey analysis. An easy-to-interpret model and a model that only remembers the most recent action but does not utilize any step made in the past. The article by Vermeer and Trilling (2020) indicates these characteristics of the Markov model. It has been stated previously in the article of Lang and Rettenmeier (2017) that RNN is a natural fit for customer journey data as the model can handle sequential data well. The article by Vermeer and Trilling (2020) states that the Markov model cannot handle multiple sequences as it has no memory. An additional article by Wu et al. (2019) supports this claim. It adds that the LSTM architecture of RNN is, in comparison to the Markov model, able to

efficiently exploit patterns for modeling sequential data. Therefore the Markov model can be deemed unnecessary while using the RNN model with LSTM architecture.

The mentioned articles have been examined and chosen based on the criteria of the articles being related and applied to customer journey analysis, being relevant, relatively recent, and having been peer-reviewed. The articles that are not peer-reviewed have been chosen as they still have reached a high level of relevancy and recency to be included.

2.7 Model Performance Metrics

It has been indicated that the heuristic models are not making use of statistical analysis and are not data-driven therefore, it is not possible to compare the heuristic models to the data-driven models with the exact same performance metrics. However, a metric that can be utilized to compare the heuristic models is the corresponding conversion rate these models give as a result from their credit allocation. This metric is used in an article by Ji, W., Wang, X., & Zhang, D. (2016) which demonstrates that heuristic models, like the first, last-touch, or the linear and time-decay models, can be evaluated with the use of this metric.

Typical performance metrics used to evaluate and compare data-driven models are accuracy, precision, recall, and F1 score. These metrics have been used in the article by Kindbom (2021) to compare LSTM to other models. In addition to these metrics, the Area Under the Curve - Receiver Operator Characteristic (ROC- AUC) Curve is a reliable metric for comparing model performance, according to an article by Bhatta (2022). This article delves deeper into the many performance metrics papers used to compare models. It is in line with the article of Kindbom (2021), which states that the metrics used in that paper are helpful for model comparison.

3 Data

The real-world data used for this research has been extracted from the website Kaggle.com on a sub-page titled E-commerce multichannel direct messaging 2021-2023. As mentioned in the title, the real-world datasets included in this sub-page are based on anonymized time series data from 2021 until 2023. The data is collected from a medium-sized online store in Russia by a company named REES46. The dataset can be used without charge on the condition that the original source and the Kaggle.com source are mentioned, which is https://rees46.com/ and https://www.kaggle.com/datasets/mkechinov/direct-messaging/data?select=campaigns.csv respectively. The following steps are described in detail to facilitate other researchers' replication of this paper's results.

3.1 Data transformation

Data transformation is crucial for enhancing the efficiency of machine learning models, removing bias, and creating accurate results. An example of this is that the data type of the *campaign_type* column in the *campaigns.csv* dataset is a character data type and has three unique values. “bulk,” “trigger,” or “transactional.” Simply dummy encoding the column to have these three values indicated as 1, 2, and 3, respectively, would, according to an article by Al-Shehari and Alsowail (2021), create bias as machine learning models will assume the column to have an ordinal relationship indicating that the “bulk’’ value has a higher importance compared to the other values as it is the first unique value. Applying these values with an ordinal relationship to a machine learning model will create incorrect results and bias, which can be avoided with one-hot encoding. This method produces a separate dummy column for each unique value. These can then be applied to a machine-learning model.

The *campaigns.csv* dataset included multiple columns that had to be binary-encoded. These columns included unique values ‘’f’’ and ‘’t’’ indicating false and true, which were binary encoded to 0 and 1, respectively. The last step of data transformation for this dataset is to transform the data type character of the date and time columns *started_a*t and *finished_at* into usable columns for machine learning models. Therefore, these columns had to be transformed using the *lubridate* package in R. to make them usable.

The *client_first_purchase_data.csv* dataset only had one column that had to be transformed: the *first_purchase_data* column. This column is a character data type and, therefore, needed to be transformed for the usability of machine learning models with the *lubridate* package. The *holidays.csv* had only one column in need of transformation. In this dataset, the *date* column was required to be converted into the correct data format. An additional binary column called *is_holidays* has been added to *holidays.csv* to indicate if a holiday is present in a row corresponding to a specific date.

The dataset messages-demo.csv had multiple columns that had to be one-hot encoded: *channel*, *platform*, *message_type*, and *email_provider*. Only the three most common unique values were included for one-hot encoding for this last column, as encoding many values would increase the dataset's dimensionality. The remaining values were classified as *other*. This one-hot encoding creates multiple new binary columns from the unique values of the encoded columns. This process makes the original one-hot encoded columns redundant, which will be removed. The columns *is_opened*, *is_unsubscribed*, *is_hard_bounced*, *is_soft_bounced*, *is_complained*, *is_blocked,* and *is_purchased* had unique values of either *f*, indicating false or *t*, indicating true. Models cannot read these values; therefore, they have been binary-encoded in 0 for *f* and 1 for *t*. Lastly, columns *sent_at*, *opened_first_time_at, opened_last_time_at*, *clicked_first_time_at*, *clicked_last_time_at*, *hard_bounced_at*, *soft_bounced_at*, *complained_at* and

purchased_at indicate time and date. These columns were the data class "character," which made them useful for incorporating into models. These columns have, therefore, been converted into the correct date-time format by the POSIXct function in R.

The columns in the datasets that have not been mentioned were already in a proper state that did not need data transformation. The full descriptions explaining each column and its data type can be observed in Appendix 10.1.

3.2 Data Merging

The merging of datasets is a necessary and beneficial process. Combining and utilizing multiple datasets will bring higher value than using separate datasets (Chen et al., 2014). The merging process starts with two datasets: the *client_first_purchase_date.csv* and the *messages-demo.csv*. These datasets were merged based on a column that both have in common: *client_id*. This creates a combined dataset named *merged_dataset*. The next step in the process is to merge *merged_data* with the *campaigns.csv* dataset. The campaigns dataset has the column *id,* which is the identification number of the corresponding campaign, and the *merged_dataset* has the *campaigns_id* column, originally from the *messages-demo.csv* dataset, as both *campaign_id* and *id* refer to the same campaigns. It is, therefore, possible to merge both datasets based on these two columns, creating the *merged_data* dataset. The last dataset that has to be merged is the *holidays.csv* dataset. This dataset has been merged with the *merged_data* dataset based on the *is_purchased* column in the *merged_data* dataset, with its origin in the *messages-demo.csv* dataset and the date column in the *holidays.csv* dataset creating the last merged dataset, *final_data*.

3.3 Balancing for the Training and Test Data

The dataset used for this research has a clear data imbalance. The conversion rate for the customers who received a message is only 0.12%. The distribution of this target variable indicating the conversion rate, named *is_purchased*, can be observed on the left side of *Figure 1*.

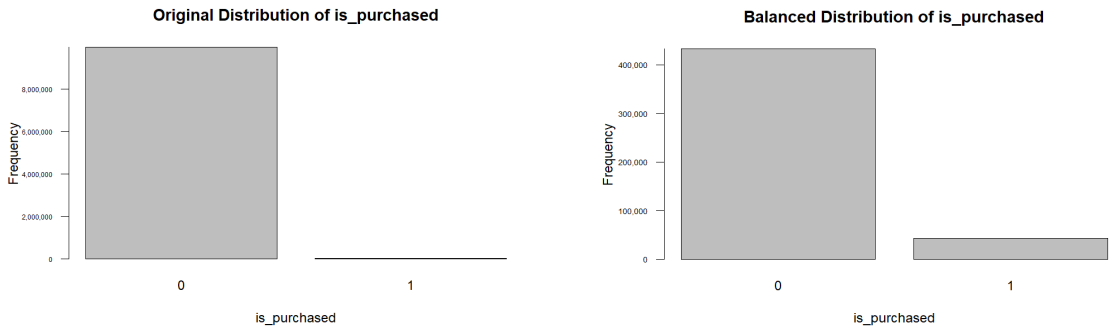

Figure 1 Distributions of the target variable is_purchased

An imbalanced dataset is one of the most considerable threats to the efficiency of machine learning models (Karatas et al., 2020). This imbalance causes classification models to be overwhelmed by the prevalent class and will, therefore, ignore the rare examples (Menardi & Torelli, 2012). One article by Rout et al. (2017) states that unbalanced datasets cause standard classifier learning algorithms like logistic regression, random forest, and neural networks to fail in creating accurate results. Therefore, the decision has been made to balance the majority and minority class of *is_purchased*.

An article by Thabtah et al. (2020) indicates that balancing a dataset equally among majority and minority classes will cause the performance metrics to perform sub-optimal. The solution instead is to balance the majority and minority classes such that the majority class holds 90% of the data and the minority class 10%. This balancing will create the highest value for the performance metrics. One method to achieve this balance is SMOTE (Synthetic Minority Oversampling). According to an article by Wongvorachan et al. (2023), this method synthetically creates additional data, in this case for the minority class of *is_purchased,* to achieve the 90/10% balance needed for the desired performance metrics. However, according to the article, this method has a sub-optimal performance with high-dimensional data and tends to overfit by introducing noise. An alternative method without SMOTE's disadvantages is to conduct an undersampling of the majority class. The article of X. Liu et al. (2009) indicates that undersampling, which uses a subset of the majority class, is an efficient method for class-imbalance problems. The drawback of this method is that it utilizes only a fraction of the majority class, which could cause helpful information to be neglected.

To utilize the most efficient balancing method, the undersampling method, and to simultaneously remove its drawback. It has been decided to balance the data by utilizing the original dataset *messages.csv*, of which the mentioned *messages-demo.csv* dataset is a subset. The *messages.csv* dataset has 172 million messages included in comparison to the 10 million in *messages-demo.csv*. The *messages.csv* dataset has, because of its size, been split into multiple random subsets. The minority class

values, of which the target variable is_purchased == 1 in one of these subsets, have been extracted and infused into the *messages-demo.csv* dataset. Increasing the minority class of the target variable from a value of 12340 to 43352. This enlargement of the minority class has created an advantageous situation in which less data of the majority class has to be undersampled. This results, in turn, in that the majority class retains more information. The final result of this undersampling process, in which the majority class was undersampled until it was in the ratio of 90/10, can be observed in the graph on the right side of *Figure 1*.

3.4 Feature Correlations After Data Balancing

The plots in *Figure 2* indicate the correlations between the predictor variables after the data balancing. These plots are essential in clarifying the relations between these predictor variables and indicating any signs of multicollinearity. The signs of multicollinearity refer to the linear relationship between predictors, which could cause major inefficiencies in estimating the model parameters (Alin, 2010). The plots indicate in dark colors the strong levels of correlation, which range from dark blue, which is a strong positive correlation, to dark red, which indicates a strong negative correlation. In general, strong correlation relations are indicated as having a value above 0.7 or below -0.7. The lighter the blue or red, the less strong the correlation is, with white indicating no sign of correlation.

The correlation plot on the left of *Figure 2* indicates a strong positive correlation between *channel.mobile_push* and *message_type.bulk* with a value of 0.9393. The strong negative correlation in this plot is between the variables *channel.email* with *channel.mobile_push* giving a value of -0.9997 and *channel.email* with *messages_type.bulk,* giving a value of -0.9389.

The correlation plot on the right of *Figure 2* indicates a strong positive correlation between the variables *camp_campaign_typebulk* with *camp_channelmobile_push* giving a correlation value of 0.936. Between *camp_campaign_typebulk* with *camp_topicsale.out* giving a value of 0.9995. Between *camp_campaign_typetransactional* with *camp_channelemail* giving a value of 0.9386 and *camp_channelmobile_push* with *camp_topicsale.out,* giving a value of 0.9355.
The strong negative correlations are between *camp_campaign_typebulk* with *camp_topicother* giving a negative correlation value of -1. Between *camp_channelmobile_push* with *camp_topicother* giving a value of -0.9356 and lastly, *camp_topicother* with *camp_topicsale*.out giving a value of -0.9999.

The values indicated by the correlation plots demonstrate evidence of multicollinearity, with some values even indicating almost or perfect linearity. This could cause the results of statistical analysis models like the logistic regression to be statistically insignificant (Daoud, 2017). Therefore, while building these statistical models, it will be required regarding efficiency to manually remove variables

that are multicollinear.

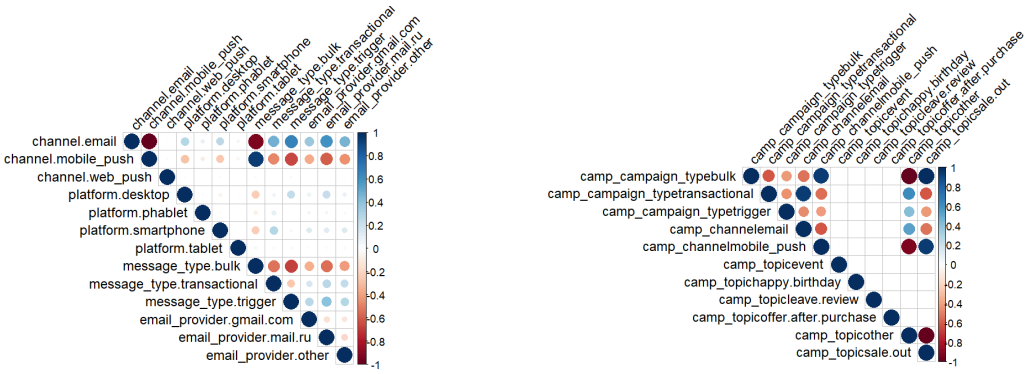

Figure 2 Correlation Plots of Messages and Campaign Touchpoints After Balancing

To understand if the balancing process has succeeded, it is essential to compare the correlation plots before and after this balancing process with each other. From the difference, it can be assumed if the relationships between features have changed too dramatically, as it would prevent models their ability to learn semantic information (Schwartz & Stanovsky, 2022). The correlation plots before balancing are indicated in *Figure 3* here, it is noticeable in the the correlation plot on the left in *Figure 3* that the correlations are relatively the same as after balancing. The only noticeable difference is that the plot on the left of *Figure 3* has for the small positive or negative correlations slightly less visibility. This indicates that the minority class in every feature is slightly less prevalent, which is expected from an unbalanced dataset.

The large differences can be observed in the correlation plot on the right side of *Figure 3*. From this plot, it can be inferred that the features *camp_topicevent*, *camp_topicleave.review*, *camp_topic_other,* and *camp_topicaoffer.after.purchase* and their respective correlations with other features are not present in the right side correlation plot of *Figure 3*, in comparison to the right side plot of *Figure 2*. This observation indicates that the correlation plot before balancing on the right side of *Figure 3* had a minority class, which for every feature, as well the class of the event happening, of these features was too small to properly infer a relationship from between these features mentioned and the other features in the plot. It is, however, contradictory to the comparison of the left side plots of *Figure 2* and *3,* in which it is the case that the correlations have become stronger. This could be due to the dominance of certain classes, leading to higher correlations between variables associated with these classes.

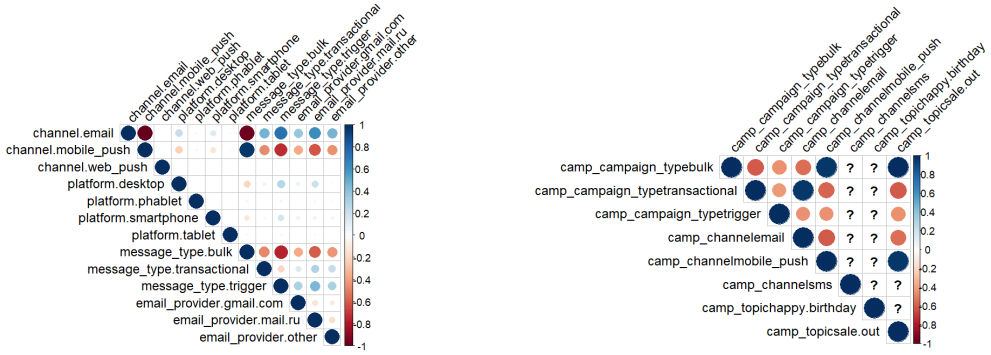

Figure 3 Correlation Plots of Messages and Campaign Touchpoints Before Balancing

These results provide evidence of the necessary process of balancing the dataset. This process creates more possibilities for calculating the relationships between predictors, which is necessary for the models that will be implemented in this research.

4 Methodology

4.1 Goals of Methodology

The methodology will delve further into the general mathematical and logical structure of the models implemented for this research. The general model description will demonstrate the components of a model, followed by the implementation of this model in the R environment.

4.2 Heuristic Attribution Models

The first-touch attribution model is a traditional rule-based model that allocates 100% of the credits to the first channel a customer has interacted with and 0% to all other channels. This model, therefore, assumes that the conversion has been caused only by the first channel a customer interacted with and assumes that all other channels had no contribution to the conversion. The first touch attribution model can be defined with the following formula:

$$Attribution_{\{FirstTouch\}}(T_i) = \sum_{\{j \in U_i\}} C_j$$

Where T_i can be defined as the representation of the i-th touchpoint, U_i represents the unique users for whom T_i is the first touchpoint, and C_j is the conversion value for user j, which is 1 if the user converted or 0 if not.

The last-touch attribution model is also a traditional rule-based model. This model allocates 100% of the credits to the last channel a customer has interacted with and 0% to all other channels. It assumes that only the channel closest to the conversion has contributed fully to the conversion. This assumption, however, demonstrates a significant flaw in this model as it fails to consider any other channel, leading to an un-optimized budget allocation, which the first-touch attribution model suffers under as well (Nisar & Yeung, 2017). The last touch attribution model can be defined with the following formula:

$$Attribution_{\{LastTouch\}}(T_i) = \sum_{\{j \in U_i\}} C_j$$

Where T_i can be defined as the representation of the i-th touchpoint, U_i represents the unique users for whom T_i is the last touchpoint, and C_j is the conversion value for user j , which is 1 if the user purchased or 0 if not. Both the first and last-touch attribution models are straightforward and simplistic, hence their popularity.

The heuristic multi-touch attribution models consider multiple channels for the path to purchase and are therefore considered more reliable than the first and last-touch attribution models.
The linear attribution model is one of these multi-touch attribution models. This model assigns an equal number of credits to all the channels considered in a customer's path to purchase. For a four-point customer journey, this model would allocate 25% of the credits to each channel (Sakly & Ouazan, 2016). This rule-based method indicates that marketers should direct budget allocation uniformly among channels customers interact with (Yuvaraj et al., 2018).

$$Attribution_{Linear}(T_i) = \sum_{j \in U_i} \frac{C_j}{n}$$

Where $Attribution_{Linear}(T_i)$ is the credit assigned to the touchpoint T_i in a linear attribution model. The U_i indicates the set of all the touchpoints in the customer journey. The C_j part is regarding the credit that is assigned to each touchpoint. Lastly, n is the number of touchpoints in the customer journey. The sum of the credits allocated will always be equal to the total credit.

The last multi-touch heuristic attribution used for this research is the time-decay model. This model adjusts credit so that the closer an impression is to conversion, the more credit it receives. Credit allocation is progressively increasing across the customer journey, indicating that this rule-based model assumes higher importance for credits closer to conversion (Nisar & Yeung, 2017). The advertiser can decide which proportionality of credit allocation is distributed among the proximity of the touchpoints to the conversion (Leguina, Rumín, & Rumín, 2020).

$$Attribution_{TimeDecay}(T_i) = \sum_{j \in U_i} C_j e^{-k(t_i - t_j)}$$

Where $Attribution_{TimeDecay}(T_i)$ refers to the attribution credit assigned to touchpoint T_i in a time-decay attribution model. The U_i indicates the set of all the touchpoints in the customer journey. The C_j in the formula indicates the credit assigned to each touchpoint. The symbol e represents the base of the natural logarithm, which is approximately equal to 2.71828. The k symbol is a parameter that controls the rate of decay which is a constant that is defined by the advertiser and determines how quickly the credit decreases as one moves away from the purchase event. The symbol t_i indicates the time of the purchase. Lastly, t_j is the time of the touchpoint T_j . Each touchpoint T_j gets a credit that decays exponentially based on its distance in time from the purchase T_i. The type of time-decay model used for this research is the linear time-decay model, as the touchpoints are not far from each other timewise in the customer journey.

These heuristic models are not known to give accurate results for managerial decision-making. However, they are still widely used among marketers, and budget allocation decisions are based on their results (Thurber, 2016). Therefore, comparing these models to data-driven models is essential to clearly distinguish their efficiency (Gaur, Bharti, & Bajaj, 2024).

4.3 The Bagged Logistic Regression

The bagged logistic regression is composed of two components. The logistic regression and the bagging process. The logistic regression is an integral model for describing the data relationship between a response variable and multiple explanatory variables. The logistic regression is the most used model to analyze this data. The main difference between this logistic model and another popular model, linear regression, is that the logistic model has a binary outcome variable instead of a continuous one. From a mathematical point of view, the logistic regression has shown to be a very flexible and effortless model to implement (Hosmer & Lemeshow, 2000). The logistic regression is heavily related to the linear regression, which can be explained by the following formula of the logistic regression:

$$Probability\ of\ outcome\ (\hat{Y}_i) = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_i X_i}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_i X_i}}$$

Here $\hat{Y}_i$ represents the estimated probability of being in one of the binary outcome categories (i) rather than representing the continuous outcome in the linear regression model. The $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_i X_i$ part of the formula represents the linear regression equation for independent variables expressed in a logit scale instead of the non-logit scale in the original linear regression. The logit scale is used because the outcome variable of the logistic regression has to be either 0 or 1, while the outcome variable of the linear regression can take any value. The logit scaling of the linear regression can be observed in the following formula to clarify this process.

$$\ln(\hat{Y}/1 - \hat{Y}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \beta_i X_i$$

Logistic regression is known to conduct maximum likelihood estimation. This process identifies, using iterative cycles, the strongest linear combination of independent variables, increasing the likelihood of the observed outcome.

To create an accurate logistic model, it is essential to conduct proper independent variable selection and fittingly build the model. The use of multilinear variables has to be avoided. This would indicate that variables are heavily correlated with each other. Regarding the appropriate number of variables to select for this model, it is most efficient to select the fewest independent variables with the highest explanatory power.

The last step in creating an efficient logistic regression model is to conduct proper model building linked to the variable selection. Three main model-building processes can be applied. The first is the direct approach. This process is useful when no predetermined hypotheses are stated, indicating the importance of some variables over other variables. In this process, all variables are simultaneously put into the logistic regression with no regard for the order or importance of the variable. If there is a predetermined hypothesis stating the importance of some variables over others, then the use of sequential/hierarchical is recommended. In this process of model building, the variables are added sequentially to see if they could further improve the model. This process is primarily useful in indicating the patterns of a causal relationship between independent and dependent variables. However, the causal patterns this process explains can become fairly complex if the number of variables increases, making drawing conclusions from the model difficult. The last model-building process is stepwise regression, which identifies the independent variables that require removal based on predetermined statistical criteria. However, this model is considered to be controversial as it can create models that are not reasonable from

a logical perspective (Stoltzfus, 2011).

The second component of the bagged logistic regression is the bagging process. Bagging is one of the most used ensemble methods. The application of bagging in classification tasks has led to significant improvements. Bagging is the process of bootstrap aggregating to create ensembles. This process involves training different classifiers with bootstrapped replicas of the original training dataset. This can be defined as a new dataset being created to train every classifier by randomly drawing with replacement data from the original dataset. This process causes the diversity of the data to be retained within the resampling procedure by using different data subsets. When an unknown instance is presented to each classifier in this process, a majority vote is carried out to infer the class.

$Bagged\ Classifier\ Output$: $H(x) = sign\left(\sum_{t=1}^{T} h_t(x)\right)$ where $h_t \in [-1,1]$ are the induced classifiers.

Algorithm: Bagging Process

Input: Training set S, Number of iterations T, Bootstrap size n
Output: Ensemble of classifiers $\{h1, h2, \ldots, hT\}$

1. For $t = 1$ to T **do**
   2. $S_t \leftarrow$ RandomSampleReplacement(n, S) // Create a bootstrap sample from the original dataset
   3. $h_t \leftarrow I(S_t)$ // Train a weak learner on the bootstrap sample to produce an induced classifier
4. **end for**
5. Return the ensemble of classifiers $\{h1, h2, \ldots, hT\}$

Where S is the training set, T the number of iterations, n the bootstrap size and I the weak learner. Step one initializes the loop to run T iterations. Step two creates a bootstrap sample from the original dataset. In step three, a weak learner I gets trained on the bootstrap sample to produce an induced classifier. In step four, the process is repeated until T classifiers are trained, and, lastly, in step 5, the ensamble of classifiers is returned. Aggregating these T classifier predictions will cause the formation of the final bagged classifier. The bagging process can also partition large datasets into smaller subsets, which are used to train different classifiers. This process includes two components: Rvotes, in which random subsets are created, and Ivotes, which creates consecutive datasets based on the importance of

instances. Important instances improve diversity, while difficult instances, identified by Out-Of-Bag (OBB) classifiers, are misclassified by the ensemble. Difficult instances are always added to the next data subset, whereas easy instances are less likely to be included (Galar et al., 2012).

This component can develop an important graph from which it can be inferred what the right number of bootstrap samples is to use, called the OBB error. Each predictor is learned from a bootstrap sample of training examples in the bagging process. The output of a bag, a set of predictors, is decided by voting. This OBB estimate is therefore based on involving the votes of each predictor whose training examples have been omitted from the sample. There are no additional predictors generated. Therefore, the OBB estimate is more time-efficient than a 10-fold cross-validation.

The Bagging process has an additional time-efficient essential metric used to determine the appropriate number of bootstrap samples called the OBB error. Every predictor was trained on a bootstrap sample drawn from the training data for bagging. This eventually leads to the final bagging model in which, by using majority voting, a prediction is made by aggregating the predictions of the predictors. The OBB error estimation is calculated by evaluating the performance of each predictor on the training set, which has not been included in the bootstrap sample. This means that for every instance in the training set, there are multiple predictors whose votes can be used to estimate for that instance. This process of OBB estimation is useful because it removes the need for additional predictors or the need to conduct a 10-fold cross-validation (Bylander, 2002).

4.4 The Random Forest Model

The Random Forest (RF) is an ensemble classifier that creates multiple decision trees by using a randomly selected subset of training samples and variables. This random selection is implemented in two ways. The first random selection is that the RF model samples the dataset with a replacement, which is the same process as bootstrap sampling, also known as bagging. The data excluded in this process, which is indicated as the OBB data, will be useful for testing the decision trees developed from the in-sample data. The second random selection occurs in the nodes of the decision trees. At each node in the trees, several predictors will be chosen. In the last step of the process, the RF model will test all possible thresholds for the selected variables in which the model decides which combination of the variable threshold results in the best split. The best split can be defined as the split that most efficiently separates the cases from the controls until the model reaches a level where only nodes are left containing cases or controls or by a pre-determined endpoint. This process is repeated until the RF model is complete (Rigatti, 2017). This process of these two random selections causes the RF model to create a higher accuracy than single decision tree models (Speiser et al., 2019). The RF classifier is increasingly used due to the accuracy of the predictions, which is among the highest in the classification setting. The model can

efficiently handle overfitting, highly dimensional data, and multicollinearity within a relatively short time.

The most important tuning parameters that must be considered for the RF model are nTree, which decides the number of trees the RF model should consist of, and the Mtry, which decides the number of features that must be considered for splitting at each node. It is recommended for these tuning parameters of the RF model to set the nTree to 500 and the Mtry to the square root of the number of input variables (Belgiu & Drăguţ, 2016). However, another article contradicts this statement by indicating that tuning the parameters of the RF model will improve performance (Probst et al., 2019). Another way to consider the RF tuning parameters is to create the OBB error plot, which is commonly used to decide on the number of nTree to use for the RF model and can additionally be used to determine the Mtry by evaluating the plot. Therefore, the OBB error plot is, just as in the bagging model, an excellent plot to evaluate the model's performance.

The RF model can create two Variable Importance Measures (VIMs): the Mean Decrease Accuracy (MDA) and the Mean Decrease Gini (MDG). These two measures indicate a ranking from the top being relevant variables to the bottom indicating irrelevant variables. Both measures are deemed essential as the stability of feature selection has been indicated as equally important to the high classification accuracy (H. Wang et al., 2016).

The following formula describes the RF process:

The final prediction of the random forest, which is indicated as the finite forest estimate, is obtained by averaging the predictions of all M trees.

$$m_{M_n}(x\,;\Theta_j,\ldots,\Theta_m D_n) = \frac{1}{M}\sum_{j=1}^{M} M_n(x\,;\Theta_j,\ D_n)$$

Where each individual tree in the forest produces a prediction denoted as $m_n\,(x\,;\Theta_j,\ D_n)$ and $m_{M_n}(x\,;\Theta_j,\ldots,\Theta_m D_n)$ is the finite forest estimate, which indicates the average predictions over all M trees. M_n is the j-th tree estimate. The x symbol is the data point for which the prediction is made, Θ_j is the random variable representing the randomness in the j-th tree construction, and, D_n is the training set used by the model to create trees. Lastly, M is the total number of trees in the random forest.

This process of aggregating decreases the variance and increases the predictive accuracy of the RF model.

4.5 The Recurrent Neural Network with LSTM Architecture

The Recurrrent Neural Network (RNN) is a type of neural network which is most efficient in processing data of sequential nature. This sequential data consists of vectors x_t with the time step of

$t = 1, \ldots, T$. In contrast to other neural networks, an RNN model makes use of feedback loops for it processing of data. These loops allow the memorization of information from the previous time steps. Especially the output h_t of each of the hidden layers in the forward pass of the neural network is dependent the current input vectors x_t and on the previous state h_{t-1}. Therefore, the formula of the RNN model can be defined as:

$$h_t = f(h_{t-1}, x_t)$$

The predictions of the model, after applying an activation function like tanh, are created based on the outputs h_t. Activation functions are non-linear functions that compute the hidden layer values by mapping a real-valued input towards a predefined range. After this process the neural network will then be trained by making use of an algorithm named backpropagation through time. This algorithm stores the states in the forward pass and then computes the gradients with respect to the weights of the backward pass. However the issue with this last process of the algorithm is that the gradients that are computed accrros the time steps tend to vanish or explode. This results in an unstable learning or the weights stop updating.

The Long Short-Term Memory (LSTM) is an architecture for neural networks that is increasing in popularity for attribution modeling. Multiple versions exist of this architecture which are all designed to learn long-term dependencies across sequences. This architecture is insensitive to differences in hyperparameters and is able to generalize well. The architecture consists of that each repeating model of the LSTM model consists of four neural network layers that interact. In the core, the cell state transfers information which is being regulated by three gates. The forget gate layer makes decisions on what information to make use of. This forget layer is followed by the input gate layer, which assists in determining what new information should be stored in the cell state. Lastly, the output of the cell state is filtered and transferred to the next time step. This architecture ensures that the exploding or vanishing gradient issue of the RNN model is solved, according to the article by Wu et al. (2019).

The recurrent neural network has, similarly to the neural network, hyperparameters which are essential for the model to function efficiently. One of these hyperparameters is the batch size. The batch size decides the number of samples the model uses before updating the internal model parameters. The batch can be defined as a for-loop that iterates over one or multiple samples and can make predictions. When the batch is finished, the predictions that have been made will be compared to the output variables which have been expected to create a calculation of the error. This error is then used to update the algorithm of the model, improving the model. There are three possibilities for tuning the batch size. The

first is if the batch size is equal to the size of the training set, which is called batch gradient descent. The second is if the batch size is equal to one sample, then the batch learning algorithm is called stochastic gradient descent. Lastly, when the batch size is more than one sample but less than the size of the training set then the learning algorithm is called mini-batch gradient descent. Research has indicated that the best-performing size is 32. As it is relatively small for a batch size, it will be more robust than larger batch sizes (Kandel & Castelli, 2020). Another important hyperparameter is the epoch. The number of epochs defines the number of times the learning algorithm will go through the training set. One epoch can be defined as each sample in the training set having the opportunity to update the internal model parameters, where an epoch consists of one or multiple batches (Brownlee, 2018). Another important hyperparameter is the learning rate which, which controls how large of a step to take in the direction of the negative gradient (Zeiler, 2012). In other words, the parameter decides the steps in minimizing the loss function and the error. This parameter is essential because an inefficiently set learning rate can lead to poor local solutions where the value of the loss function is not better than other local solutions. An inefficient learning rate, therefore, causes the neural network to perform sub-optimal (Takase et al., 2018). It has additionally been shown that setting the learning rate low would cause poor generalizability but a high optimization for the training loss, which means that the error will be less and the accuracy higher (Li et al., 2019).

An important regularizartion technique that is shown to improve a neural network is the ridge regularization as well known as the L2 regularization it is a technique used for improving a models generalizabilityy. It does this by imposing constraints on the parameters of a neural network model and adds penalties in this model to the objective function during the optimization process (Ni, Fang, & Huttunen, 2021).

The RNN will produce multiple important plots. One of these important plots is the plot showcasing the validation accuracy with the training accuracy. If the lines of these two metrics in the plot are near each other in proximity then it can be inferred that the neural network model can generalize well to unseen data. The other important plot is the validation loss plot which is a key metric used to evaluate the performance of a RNN model on a validation dataset (Alzubaidi et al., 2021).

4.6 SHAP Values

The SHapley Additive exPlanations (SHAP) values are a combination of Local Interpretable Model-agnostic Explanations (LIME) and Shapley value estimation.

Lime is a method that interprets model predictions by basing these predictions on a locally approximation of the model around a given prediction. Lime makes use of a local linear explanation model and is an instance-based explainer that generates simulated data points around an instance by

making use of random perturbation and provides explanations by fitting a weighted, sparse linear model over the predicted responses from the perturbed points. The resulting explanations of LIME are locally faithful to an instance regardless of classifier type (Zafar & Khan, 2021).

The Shapley value method can be described by several components. These include fairness, symmetry, and efficiency. The original setting of this method is a cooperative game that consists of a player set and a scalar-valued characteristic function that can define the value of the subsets of the players. In such a game, the shapley values offer an innovative way to distribute the collective value of the team across individuals. Therefore, the individual features in a machine learning model can be inferred from the collective value of the model with shapley. To apply this method to machine learning, we will need to define two components. These are the player set and the characteristic function. The player set must be represented by a set of input features or data points. The characteristic function can be described as the goodness of fit for a model or out-of-sample model performance (Rozemberczki et al., 2022).

The SHAP is a popular feature-attribution mechanism that uses game-theoretic notions to measure the influence of individual features on the prediction of a machine-learning model. The explanations of this method determine the influence of a given feature by systematically computing the expected value of the machine learning model given a subset of the features. The SHAP values therefore depend on the predictive model as well ass the assumptions on the underlying data distribution (Van Den Broeck et al., 2022). The SHAP method produces an important plot. This plot depicts a summary of the estimated SHAP values of the model colored by feature values for all the main feature and their interaction effects. These are ranked from top to bottom by importance. This plot, called the SHAP summary plot, accurately identifies the important features and quantifies the amount of feature contributions of the model

4.7 Performance Metrics

Performance metrics are essential to compare models with each other. It is common to create confusion matrices for models so these metrics can be inferred from them. A confusion matrix is a 2x2 table that indicates true positives, false positives, true negatives, and false negatives values. These values can be used to calculate performance metrics like sensitivity, as well known as recall, specificity, precision accuracy, and the f1 score. The following formulas, one to five, are used to calculate the values of these metrics.

1. $Sensivity = \frac{TP}{TP + FN}$ which is the fraction of positive cases predicted as positive.
2. $Specificity = \frac{TN}{TN + FP}$ which is the fraction of negative cases predicted as negative

3. $Precision = \frac{TP}{TP + FP}$ which is the fraction of truly positive cases from all cases the model predicted positive

4. $Accuracy = \frac{TP + TN}{TP + FN + TN + FP}$ which is the fraction of cases the model correctly predicted

5. $F1\ Score = \frac{2TP}{2TP + FP + FN}$ which is the harmonic mean of positive predictive value and sensitivity

The performance metric that is highly useful for data-driven model comparison is the The Area Under the Receiver Operating Characteristic Curve (AUC-ROC). It demonstrates the predictive capabilities of a model with a value. For $Auc = 1$ The model has perfect prediction capability. For $0.5 < AUC < 1$ The model performs better than random guessing. The closer to 1, the better the model's performance. For $AUC = 0.5$ The model performs no better than random guessing. Lastly, for $Auc < 0.5$ The model performs worse than random guessing, which may indicate that the model is inversely predicting the classes (Erickson & Kitamura, 2021).

The performance metric which is helpful for heuristic models is the conversion rate. This rate indicates the average conversion rate that results from the calculations of a heuristic model. From this conversion rate, it is possible to infer how well these models are performing at finding touchpoints that contribute highly to the conversion (Ji, Wang, & Zhang, 2016).

5 Results

5.1 Heuristic Models

The process of implementing the first-touch attribution model has given results indicated in *Table 1*. This table showcases the campaign touchpoints, which are the first touchpoints a customer can interact with. This table has ten feature conversion rates in order of high to low. These rates are necessary to indicate the conversion rate comparison metric of the first-touch heuristic model. It can be observed that the most occurring touchpoint is the *camp_topicother* with a *n_total* value of 303438. This value is linked to the n_conversions value of 40255. Dividing the *n_conversions* by the *n_total* results in the conversion_rate of $\frac{40255}{303438} = 13.27\%$. Therefore, the conversion rate most prevalently indicated by the first-touch attribution model is 13.27%.

Table 1 Touchpoint Conversion Rate campaign

Touchpoint Name	n_conversions	n_total	conversion_rate
camp_topicleave.review	11	11	100.00
camp_topicoffer.after.purchase	5	5	100.00
camp_topichappy.birthday	1	1	100.00
camp_campaign_typetransactional	30746	188264	16.33
camp_channelemail	28834	193040	14.94
camp_topicother	40255	303438	13.27
camp_campaign_typetrigger	9509	115095	8.26
camp_channelmobile_push	5009	168737	2.97
camp_campaign_typebulk	3097	173513	1.78
camp_topicsale.out	3080	173412	1.78

Note: n_conversions is the number of conversions of the touchpoint, and n_total is the total amount of touchpoint occurrence. The *conversion_rate* is in percentages.

The conversion rate of the last-touch attribution model has been calculated similarly to that of the first-touch attribution model. In *Table 2*, the results of this model are demonstrated. The conversion rates of the 13 message touchpoints are in order of top to bottom. This table indicates that the touchpoint with the highest number of conversions of the messages is the *channel.email* touchpoint. The *n_conversions* for this touchpoint is 38161, and the *n_total* is 199215. Dividing the *n_conversions* by the *n_total* results in a *conversion_rate* of $\frac{38161}{199215} = 19.16\%$. This result indicates that the conversion rate that is most prevalently indicated by the last-touch attribution model is 19.16%

Table 2 Touchpoint Conversion Rate Messages

Touchpoint Name	n_conversions	n_total	conversion_rate
platform.phablet	2137	2407	88.78
platform.smartphone	17859	22521	79.30
platform.tablet	425	580	73.28
channel.web_push	49	81	60.49
platform.desktop	12912	25872	49.91
message_type.transactional	31116	79672	39.06
email_provider.other	13934	60968	22.85
email_provider.gmail.com	8033	41741	19.24
channel.email	38161	199215	19.16
email_provider.mail.ru	16181	96517	16.76
message_type.trigger	9509	115095	8.26

Touchpoint Name	n_conversions	n_total	conversion_rate
platform.	9975	425448	2.34
channel.mobile_push	5142	277576	1.85
message_type.bulk	2727	282105	0.97

The linear attribution model will be calculated as indicated in the appendix model preparation 10.2.4. The total number of clients who made a purchase is 19077. The total sum of these customers' conversion rates and unique touchpoints is 2902510.11. Dividing these two values will give an average touchpoint conversion rate of $\frac{2902510.11}{19077} = 152.15$. This value 152.15 then has to be divided by the average amount of touchpoints customers have. This has been calculated to be 6.635844. $\frac{152.15}{6.635844} = 22.92$. Therefore the *conversion_rate* of the linear attribution model is 22.92%.

The time-decay attribution model has 100 credits to allocate among the touchpoints. The following linear time-decay process is to be applied. The customers who interacted with the *campaigns* touchpoints, which are the first touchpoints, will have their respective conversion rates halved, and the customers who interacted with the *messages* touchpoints will have the sum of their respective conversion rates of the *messages* touchpoints multiplied by 1.5. The resulting average channel conversion rate by conducting this time-decay process results in a conversion rate of 37.63%.

The first-touch attribution model has a conversion rate of 13.27%. The last-touch attribution model gradually increases its conversion rate compared to the first-touch model, reaching 19.16%. The linear attribution model increases its conversion rate slightly compared to the last-touch model, reaching 22.92%. The time-decay attribution model increases significantly over the linear attribution model, reaching 37.63%.

From these results, multiple conclusions can be drawn. The first is that of the one-touch attribution models. The last touch has the highest conversion rate and, therefore, would be considered the most useful, which can explain its popularity compared to the first-touch model. Additionally, it provides evidence that touchpoints closer to the purchase event are more valuable as indications than earlier touchpoints, which aligns with the literature.

The next conclusion is that of the heuristic multi-touch attribution models. The time-decay model is more efficient in its allocative process than the linear attribution model. This also indicates that a model that allocates a higher value to touchpoints closer to the purchase event will yield superior results. These results for the multi-touch attribution models are in line with the literature, which states that the

multi-touch attribution models are more efficient in comparison to the single-touch attribution models and that the time-decay is the most efficient heuristic attribution model which should be used if the available models are only the heuristic attribution models.

5.2 Bagged Logistic Regression Results

The graph of Figure 4 indicates the OBB Error on the y-axis, ranging from 0.032 to 0.037, and the corresponding bootstrap samples on the x-axis, ranging from 0 to 32. The OBB error is visibly fluctuating, especially until around 23 bootstrap samples. The model shows fewer fluctuations after these first 23 samples, indicating that the model's performance is becoming increasingly consistent with additional bootstrap samples. The OBB rate ranging from 0.032 to 0.037 indicates that the model has a generalization error rate of around 3.4 to 3.6%. This rate suggests that the model reasonably predicts the target variable *is_purchased*.

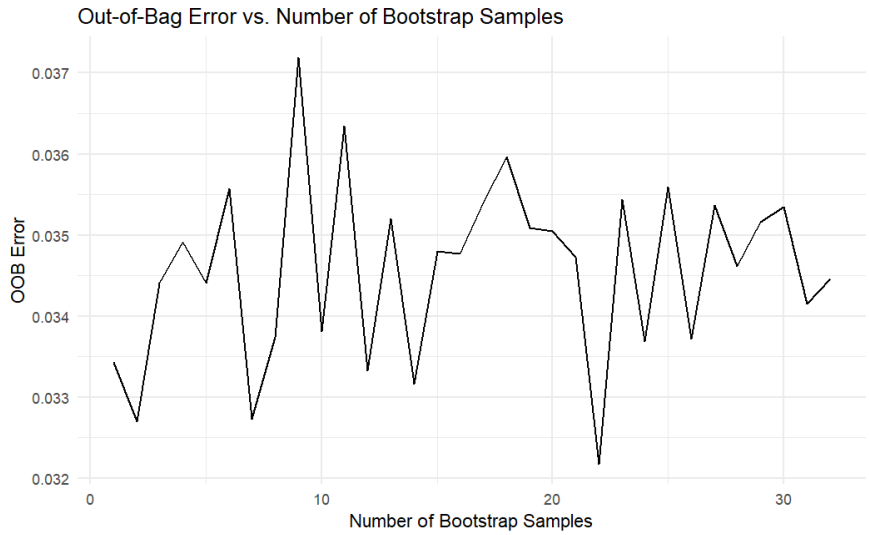

Figure 4 Out-Of-Bag Error and Number of Bootstrap Samples

The variable importance plot of Figure 5 indicates how the features of the bagged logistic regression impact the predictive performance of the model. It can be observed in this plot that the *message_type.bulk* feature is the most important feature with an importance value significantly exceeding all other features of 0.17. The feature with the second best impact on the predictive performance of this model is *camp_campaign_typetrigger,* with an importance value of 0.12. This feature is followed by *platform.smartphne* with a value of 0.09, *platform.desktop* with a value of 0.08, *camp_channelemail* with

a value of 0.04, *email_provider.mail.ru* with 0.03, *platform.phablet* with 0.02 and *email_provider.gmail.com* with a score slightly above 0,01. The feature *platform.tablet* has a slightly positive but negligible importance value regarding the predictive performance of the model, and the feature *channel.web_push* has no importance value as this feature brings no improvement to the predictive performance.

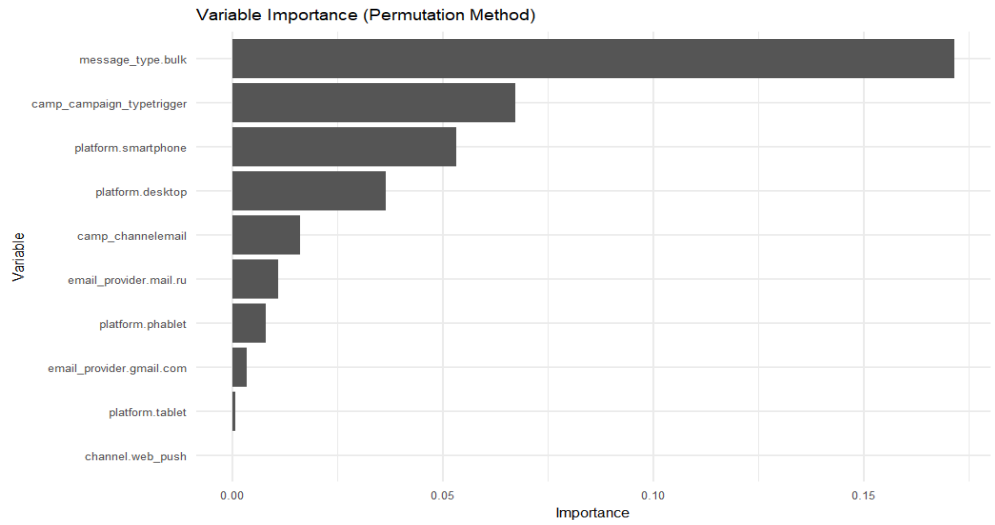

Figure 5 Variable Importance Plot of Bagged Logistic Regression

The performance metrics and their corresponding values have been calculated for the Bagged Logistic model in *Table 3*. This table indicates a high accuracy for the model with a value of 0.9638. This value indicates the portion of both positive and negative classified instances out of the total instances. The value indicates a correct classification of 96.38% of these instances in the test set. The recall, also known as the sensitivity, indicates the proportion of true positive instances out of all the actual positive instances. The recall value for this model is 0.9920, which corresponds to the model successfully identifying 99.20% of the positive instances. This high value indicates that the model performs well in indicating the purchase events. The specificity indicates the proportion of true negative instances out of all the actual negative instances. The value for this metric being 0.6819 indicates the model correctly classified 68.19% of the negative instances. This result suggests that the model is less effective in detecting the negative class than the positive class. The F1-Score indicates a harmonic mean of precision and recall, balancing the two metrics. The value for this metric is 0.9803, corresponding to a percentage of 98.03%. This percentage suggests a strong balance between precision and recall. Lastly, the ROC-AUC metric indicates the Area Under the Receiver Operating Characteristic Curve, which measures the model's ability to discriminate between positive and negative instances across all threshold values. The value of this metric

is 0.9653. This indicates that the model has an excellent discriminatory ability, as this metric's value of 96.53% is considered high. However, to achieve these results, it was necessary to reduce the number of predictors from 26 to 14 because the model could not handle mutually exclusive or perfectly separated predictors.

It can be concluded from the Bagged Logistic Regression that this model is highly effective at predicting purchases but slightly less effective at predicting non-purchases. The overall performance is strong, with excellent accuracy, recall, and ROC-AUC values.

5.3 Random Forest Results

The results of the Random Forest (RF) model with pre-determined parameters, a nTree value of 500, and a Mtry of $\sqrt{26} = 5$. Where 26 is the number of predictors. Will be compared to a fine-tuned RF model with grid-search and optimization of the nTree with the OBB error and nTree graph. The performance metrics of the RF model with pre-determined parameters and the RF model with fine-tuned parameters are indicated in *Table 3*. The most efficient RF model based on performance metrics will be used to compare with other models. The bold letters in *Table 3* indicate the performance metric of the corresponding model, which is an improvement compared to the other model. It can be observed that the accuracy and recall are the same for both models. Indicating no improvement or deterioration. However, the metrics accuracy, specificity, f1-score, and ROC-AUC indicate a clear improvement in the fine-tuned model compared to the pre-determined model. These results prove the opposite of the article of Belgiu and Drăguţ (2016). This article states that the mTry should be set to the square root of the predictors used for efficiency. The results support the statements in the article of Probst et al. (2019), stating that parameter tuning will enhance the performance of the RF model.

The OBB error with the corresponding nTree graph of both models can be observed in Figure 6. The pre-determined model is indicated on the left side of the figure, and the fine-tuned model is on the right. For both models, it can be observed that initially increasing the number of trees significantly lowers the OBB error. However, the pre-determined model demonstrates an increase in OBB error from 100 trees, which decreases again slowly around 350 trees. The fine-tuned model demonstrates a stabilization of the OBB error near 90 trees. These results suggest an increase in efficiency in the fine-tuned model compared to the pre-determined model. The model can achieve a lower and stabilized OBB error with fewer trees. These results further indicate the superiority of the fine-tuned RF model over the pre-determined model.

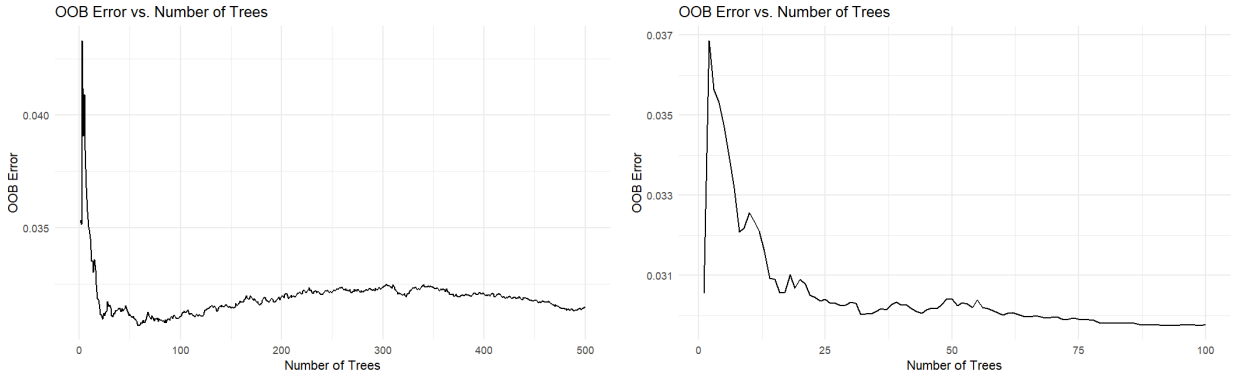

Figure 6 Pre-determined and Fine-Tuned OBB Error vs Number of Trees graphs.

The variable importance plot results of the fine-tuned RF model can be observed in Figure 7. The plot on the left indicates on the y-axis, the touchpoints used in the fine-tuned RF model ranked from top to bottom based on their importance. The x-axis indicates the mean decrease in accuracy. This metric indicates how much the model's accuracy decreases if a particular variable is omitted. Variables with higher values indicate that a variable is more important for an accurate prediction.

Platform.smartphone is indicated as the most important touchpoint for an accurate prediction of the model, followed by the other platform touchpoints, *platform.desktop*, *platform.phablet,* and *platform.tablet*. This indicates that the platform a client uses is a significant indicator of whether they decide to purchase. The less but still important variables are the message touchpoints *message_type.transactional*, *message_type.bulk,* and the channel touchpoints *channel.web_push*, *channel.mobile_push* and *channel.email*. The least important variables regarding their influence on accuracy are the topic touchpoints *camp_topicother*, *camp_topicsale.out*, *camp_topicoffer.after.purchase*, *camp_topichappy.birthday* and the channel touchpoint *camp_channelsms*. These variables could be removed, and the accuracy would remain the same.

The plot on the right indicates on the y-axis, the touchpoints used in the fine-tuned RF model, which is also ranked from top to bottom based on the contribution these touchpoints provide in reducing the uncertainty of the model. The mean decrease in gini on the y-axis indicates how much impurity reduction these touchpoints are associated with. The touchpoints regarding the platform the clients use *platform.smartphone*, *platform.desktop*, together with the message touchpoint *message_type.transactional*, provide the highest level in the mean decrease gini, indicating that these variables are the most influential in improving the model's predictive power by providing the most information gain. The touchpoints contributing nothing to improving the model's predictive power are

topic touchpoints *camp_topicleave.review*, *camp_topicoffer.after.purchase*, *camp_topicevent*, *camp_topichappy.birthday*, and the *channel touchpoint camp_channelsms*. The rest of the touchpoints contribute moderately to reducing the RF model's impurity.

These two plots conclude that both plots indicate the significant importance of the RF model of clients using the smartphone and desktop platforms and the message type touchpoint *message_type.transactional*. The touchpoints *camp_topicleave.review*, *camp_topicoffer.after.purchase*, *camp_topicevent*, *camp_channelsms*, and *camp_topichappy.birthday* have a negligible impact on the model's performance. These variables could be removed from the RF model without significantly affecting accuracy, leading to a more efficient and streamlined model.

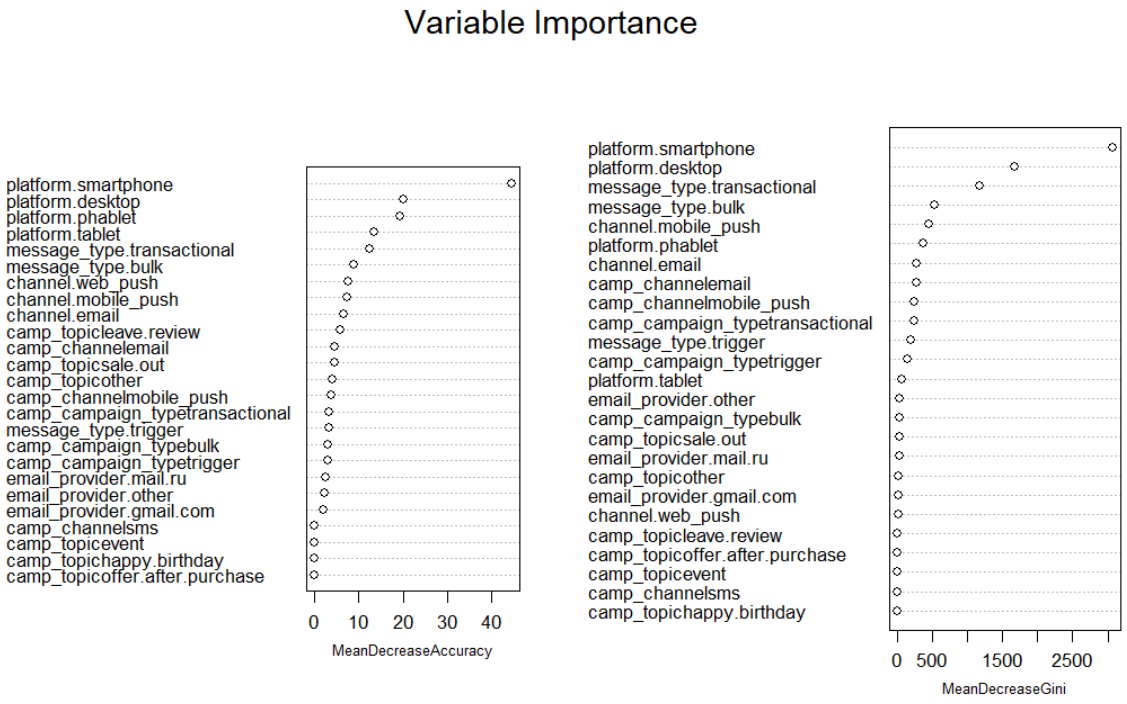

Figure 7 Variable Importance Plots Of Fine-Tuned RF Model.

5.4 Recurrent Neural Network with Long Short-Term Memory

The decision has been made to optimize the hyperparameters of the RNN model in the following way. The number of epochs has been set to 50, the learning rate to 0.0001, and the batch size to 32 to create the best-performing RNN model. The epoch was set to 50 in regarding computational efficiency. The learning rate was set low as with this rate, the model has a high accuracy, which is part of increasing the performance metrics, and the batch size was set to 32 as this would aid in creating a robust model.

Additionally, the model was only able to function by implementing L2 regularization. The model has been able to use all predictors in the data plus additional features like *total_campaigns, total_messages, total_purchases, avg_time_since_first_purchase, avg_time_since_last_open, avg_time_since_last_click, avg_time_since_unsubscribe avg_time_since_complaint, avg_campaign_duration*, these additional features will provide the model with results that are based on time predictors which should increase the fit of the model to the customer journey data of this research.

The left side plot in *Figure 8* indicates that both the train loss and validation loss are significantly decreasing until around an epoch of 9. The decrease further continues until around epoch 40 after which the decrease stagnates. This continuing decrease indicates that the RNN model is significantly learning and improving itself from the training and validation sets, which stagnates from an epoch of 40. It can as well be inferred from this graph that there is no sign of overfitting, as the training and validation loss lines are close to each other. Therefore the RNN model is generalizing well to unseen data. It can be observed in the right side graph of *Figure 8* that from an epoch value of 0 until approximately 2 that the training and validation accuracy for the model is rapidly increasing. This rapid increase is followed by a stabilization from epoch 2 until approximately epoch 34 as the accuracy of the RNN model for both the training and validation set are stable between a value of 0.95 and 0.96. The last significant increase takes place from approximately epoch 34 to 39 as the both the values of the training and validation set re-stabilize, which demonstrates that the RNN model needed an epoch value of around 39 to be stable in its resulting accuracy. The close alignment between the lines of both sets indicates that the RNN model is just as indicated on the left side of *Figure 8*, not overfitting and generalizing well to unseen data.

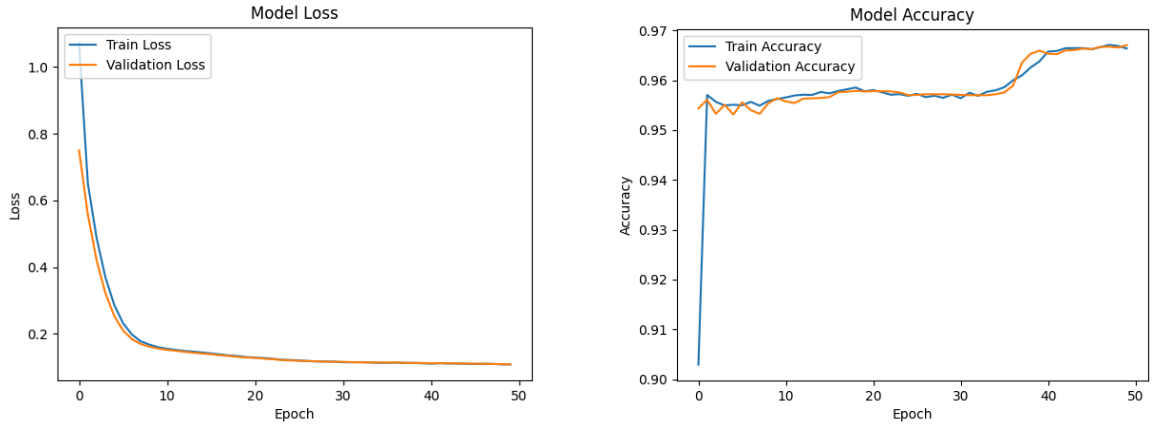

Figure 8 The model loss and model accuracy plots.

The results of the RNN model are described in *Table 3*. Here, it can be observed with an accuracy metric value of 0.9721 that 97.21% of the predictions made by the RNN model are correct. This high level of accuracy indicates the model's sufficient predictive performance. The recall for this model is 0.9949, indicating that the RNN model can correctly identify 99.49% of the actual positive cases. The specificity of 0.9365 indicates that 93.65% of the negative cases are correctly identified by the model, and the F1 score of 0.7023 suggests that the model has a moderate balance between precision and recall.

5.5 Comparison of Models

Table 3 Model Metric Comparison

Performance Metric	Bagged Logistic Regression Metrics	Pre-Determined Random Forest Metrics	Fine-Tuned Random Forest Model Metrics	Recurrent Neural Network Metrics
Accuracy	0.9638	0.9675	0.9697	**0.9721**
Recall	0.9920	**0.9991**	**0.9991**	0.9949
Specificity	0.6819	0.6516	0.6753	**0.9365**
F1 Score	0.9803	0.9825	**0.9836**	0.7023
ROC-AUC	**0.9653**	0.8254	0.8372	0.8530

Note: The bold numbers indicate that these values are the highest or equal to the highest value out of all the models.

It can be observed in *Table 3* that the bagged logistic regression has the worst performance in all metrics except for the ROC-AUC value. This result indicates that this model would be essential in cases where distinguishing between positive and negative classes across all threshold levels is vital. Even though all other metrics are lower, the model does come close in regards of accuracy, recall and f1 score to the other models. This model is followed by the pre-determined random forest, which performs worse than the fine-tuned random forest model in all metrics except for recall. Therefore, if there is no possibility to fine-tune hyperparameters and the correct identification of actual correct cases is the most important, then the pre-determined random forest model is essential to implement. The result of the fine-tuned model performing better in general than the pre-determined random forest model contradicts the findings of (Belgiu & Drăguţ, 2016) and supports the findings of (Probst et al., 2019). If it is possible to fine-tune the random forest model, correctly identifying actual correct cases, the balance of effectively identifying true positive cases and minimizing false positives is of importance then choosing for this fine-tuned RF model would be vital. Lastly, the recurrent neural network with LSTM architecture is shown to have the highest accuracy and the highest specificity. This RNN model should be used in situations where general correctness and the necessity to reduce false negatives are vital.

The RNN model is shown to utilize customer journey features that indicate the time of a touchpoint, have no signs of overfitting, can generalize well and have the highest accuracy and specificity. For this reason, it has been decided that the recurrent neural network is the best-performing data-driven attribution model for customer journey analysis. This model will, therefore be further interpreted with the use of SHAP values. This finding of the RNN model being the best performing contradicts the findings of Nygård and Mezei (2020), who state that the RF model should be better for customer journey analysis than neural network models, including the RNN model.

5.6 Shap Analysis

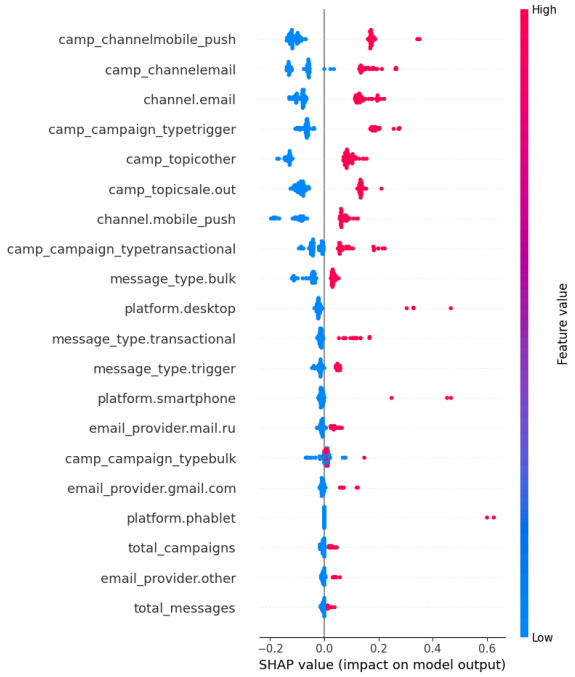

Figure 9 SHAP interpretation of the RNN model.

These results and the possibility to utilize features based on time have made the decision clear to apply the SHAP value interpretation to this model, which can be observed in *Figure 9*. It can be observed from this plot that *camp_channelmobile_push*, *camp_channelemail and channel.email* are among the most important features, having a significant impact on the model's predictions and that *camp_campaign_typetrigger*, and *camp_topicother* also show substantial impact as indicated by their position near the top of the list. For c*amp_channelmobile_push* the high values in red are associated with higher SHAP values, indicating that high values of this feature increase the output of the model, while *camp_channelmobile_push*, low values, as indicated in blue, are associated with lower SHAP values,

indicating that low values of this feature decrease the model output. The same can be stated regarding these features: *camp_channelmobile_push*, *camp_channelemail*, and *channel.email. camp_topicother* and *camp_topicsale.out* that have a mix of high and low values impacting the prediction in both directions, suggesting a more complex relationship with the target variable. The features *email_provider.mail.ru, camp_campaign_typebulk, email_provider.gmail.com, platform.phablet, total_campaigns, email_provider.other, and total_messages* have SHAP values centered around zero, indicating that they have little to no effect on the model output. These predictors with no effect could potentially cause issues for the predictive performance of the RNN model. Therefore, the SHAP value of these predictors further justifies the use of the L2 regularization technique applied to the RNN model.

5.7 Channel Attribution Results

The results of the model's predictive performance with channel attribution for the data-driven models have been indicated in *Table 4*. These results could potentially indicate which channels are most important for the conversion in the dataset. These normalized values indicate when negative for a touchpoint regarding the data-driven model that this touchpoint has a negative effect on the model's predictive performance. Positive values indicate a positive effect of model performance, and a value of zero indicates no positive or negative effect on the predictive performance of the model..

It can be observed that the bagged logistic regression uses the fewest touchpoints, which can be attributed to the model's inability to handle multicollinearity. The RNN has the second fewest touchpoints as it has automatically excluded variables it deemed unimportant. It is noticeable that the majority of the touchpoints the RNN model excluded are the same touchpoints the bagged logistic model excluded.

The platform touchpoints *platform.desktop*, *platform.phablet*, *platform.smartphone,* and *platform.tablet* in general showcase the highest positive values of the channel attribution, indicating the importance of these touchpoints regarding their impact on the predictive performance for conversions. Only the pre-determined RF model indicates a negative value for *platform.phablet,* which can further provide evidence of the fine-tuned RF model being a better alternative. The *camp_campaign_typetrigger*, *message_type.bulk,* and *email_provider.gmail.com* showcase mostly negative attribution values across all data-driven models, indicating that their removal would benefit most models.

Regarding individual models, it can be observed for the bagged logistic regression that excluding the *camp_campaign_typetrigger* and *message_type.bulk* touchpoints would be most sensible as these touchpoints have the lowest value of all the touchpoints in this model. The pre-determined RF model has five touchpoints with the lowest value, which are *camp_channelsms*, *camp_topicevent*, *camp_topichappy.birthday*, *camp_topicleave.review,* and *camp_topicoffer.after.purchase*. These touchpoints share the negative value of -0.55, and their removal would significantly benefit the

performance of this model. The fine-tuned RF model shares the same touchpoints with the lowest value that should be removed, which are *camp_channelsms*, *camp_topicevent*, *camp_topichappy.birthday*, *camp_topicleave.review,* and *camp_topicoffer.after.purchase* with a value of -0.51. Lastly, the RNN model has two touchpoints indicating a significantly negative impact on model performance that should be removed, which are *camp_channelemail* and *camp_channelmobile_push*.

Comparing these results to the heuristic models, it can be observed that for the campaign touchpoints mentioned in *Table 1,* which indicates the first touchpoint of a customer, is that nearly all of these touchpoints in this table are indicated to have a negative effect on the predictive performance of the models in *Table 4*. The only touchpoints and models for which this is not the case are *camp_channelemail* with the pre-determined RF model and *camp_campaign_typebulk* with the RNN model. This could provide further evidence of the lesser importance of the first touchpoints in a broader context, taking as many variables and effects into consideration. Additionally, it can be observed for the second touchpoints customers interact with in *Table 2* that the first six touchpoints with the highest conversion rates are indicated to have a majority of positive values in *Table 4*, except for *platform.tablet*. These touchpoints are *platform.phablet*, *platform.smartphone*, *platform.desktop,* and *message_type.transactional*. This observation indicates that not only does the literature support the claim that touchpoints closer to conversion, as in *Table 2,* are more important indicators of a conversion, but it is additionally as well supported by the data-driven models. The values in *Table 4* directly indicate the importance of touchpoints in influencing the predictive ability of models, and the values additionally, indirectly indicate the importance of the touchpoints in influencing the event of a conversion.

Table 4 Channel Attribution Scores

Touchpoint Name	Bagged Logistic Regression Normalized Score	Pre-Determined Random Forest Normalized Score	Fine-Tuned Random Forest Normalized Score	Recurrent Neural Network Normalized Scores
camp_campaign_typetrigger	-1.53	-0.22	-0.3	-0.77
camp_channelemail	-0.66	0.02	-0.13	-1.08
channel.web_push	0.40	-0.54	-0.50	X
platform.desktop	0.96	1.66	0.90	0.00
platform.phablet	1.21	-0.11	0.01	1.30
platform.smartphone	1.09	3.95	3.94	0.43
platform.tablet	1.10	-0.49	-0.43	X
message_type.bulk	-1.32	0.47	0.25	-0.21
email_provider.gmail.com	-0.49	-0.53	-0.49	1.08
camp_campaign_typebulk	X	-0.49	-0.47	0.87
camp_campaign_typetransactional	X	-0.22	-0.18	-0.21

Touchpoint Name	Bagged Logistic Regression Normalized Score	Pre-Determined Random Forest Normalized Score	Fine-Tuned Random Forest Normalized Score	Recurrent Neural Network Normalized Scores
camp_channelmobile_push	X	-0.10	-0.16	-1.29
camp_channelsms	X	-0.55	-0.51	X
camp_topicevent	X	-0.55	-0.51	X
camp_topichappy.birthday	X	-0.55	-0.51	X
camp_topicleave.review	X	-0.55	-0.51	X
camp_topicoffer.achter.purchase	X	-0.55	-0.51	X
camp_topicother	X	-0.49	-0.49	-0.64
camp_topicsale.out	X	-0.47	-0.47	-0.43
channel.email	X	0.05	-0.11	-0.86
channel.mobile_push	X	0.07	0.15	-0.43
channel.web_push	X	-0.54	-0.50	X
message_type.transactional	X	1.34	1.20	0.09
message_type.trigger	X	-0.18	-0.23	0.22
email_provider.mail.ru	X	-0.50	-0.48	0.65
email_provider.other	X	-0.46	-0.46	1.73

Note: The X indicates that there is no value for this touchpoint corresponding to the model.

6 Conclusion

This research has been inspired by the complexity of the contributions that channels can have in a customer journey. This complexity has led to the creation of the research question: *Evaluating the feasibility of supervised machine learning models for multi-touch attribution.* Answering this research question could pave the way to solving this complexity and the many issues that arise from it. The literature on this topic has indicated that the heuristic models are inefficient compared to data-driven models. However, the literature did indicate that a significant number of companies still make use of these heuristic models for their customer journey analysis, of these heuristic models which are the first-touch, last-touch, time-decay, and linear attribution. The time-decay is most reliable as it considers multiple touchpoints while assigning more credits to touchpoints closer to conversion. Therefore, as these models are still widely used, it is important to consider these as a benchmark to the data-driven models and to demonstrate their indicated inefficiencies.

The bagged logistic regression model has been chosen as the first data-driven model as it has far higher stability than the logistic regression. The model has shown to have high accuracy, indicating its predictive ability. However, it has only been properly applied once to this kind of research and data. Therefore, this model has been chosen to have its results contribute to this research gap and for its predictive success. The RF model has been chosen for this research as it has been implemented frequently

with this kind of data. The RF model has been shown to be an efficient model for large datasets with data regarding automation tasks like online shopping, which this research is utilizing. The RF model is shown to be robust and has predictive superiority over traditional models. The last model, the RNN model, has been chosen as numerous articles indicate it to be the most efficient model for customer journey analysis. Adding the LSTM architecture to this model solves the issue this model has with vanishing, as well called the exploding gradient problem. The real-world e-commerce data chosen for this research is based on millions of messages with dozens of touchpoints, providing ample opportunity to implement the mentioned models on customer journey data. The models have been implemented on the data, and therefore, the following information will answer the research question:

The conversion rate has been used as the performance metric for the heuristic models. The results of implementing these models are that the first-touch attribution model has the lowest conversion rate and the last-touch attribution model shows the highest conversion rate of the single-touch attribution models. In the multi-touch attribution models, the linear attribution is shown to have the lowest and the time-decay model is shown to have the highest conversion rate. This time-decay model should therefore be utilized if there is no possibility to use a data-driven attribution model as it has the highest conversion rate out of all heuristic models. The time-decay model being the best performing multi-touch heuristic model and the heuristic model in general combined with the result that the last-touch model is the best performing single-touch heuristic model indicates that channels closer to conversion are of higher importance according to the results of the heuristic models.

The bagged logistic regression model has shown to be the worst performing model in performance metrics except for its ROC-AUC score, even though the model is generally not far off the other models in the other metrics.

The pre-determined random forest model is shown to perform worse than the fine-tuned random forest model only having an equal recall, therefore, fine-tuning should be utilized to create the best-performing random forest model. The fine-tuned random forest model has the highest recall, which is shared with the pre-determined random forest model and F1 score. The recurrent neural network model has the highest accuracy and specificity of all models. This model is indicated to be the best-performing model for customer journey data as, in combination with the well-performing metrics, it can additionally encompass time-orientated predictors into its research, which are essential for the customer journey.

Comparing the heuristic models channel conversion rates to the values of the channels for the data-driven models influencing the predictive performance has given multiple important insights. The most important insight is that, as indicated in the literature and further supported by the results of this comparison, the first touchpoints, which are the campaign touchpoints, are of lesser importance than the

later touchpoints. Lastly, the touchpoints which are indicated by both the heuristic and data-driven models to be the most important for conversions are *platform.phablet*, *platform.smartphone*, *platform.desktop,* and *message_type.transactional*. These touchpoints are deemed the most valuable for the e-commerce platform from which this data has been extracted

The results of this research will contribute to increasing the maximum budget allocation efficiency by reducing marketing efficiencies like the ad effect measurement. It does this by indicating the models that are most efficient in their conversion rate for heuristic models, which is the time-decay model, and the best performing and best-fit model for customer journey analysis is the recurrent neural network model. The models can therefore be applied by firms on customer journey data, which will result in a better understanding of which channels contribute the most to conversion and which do not. Lastly, combining the heuristic and data-driven model results can enhance the support and evidence for claims regarding the importance of touchpoints in the customer journey.

7 Discussion

7.1 Limitations of the research

This research has a few limitations. The first limitation is that the conversion_rate metric for the heuristic attribution models cannot directly be compared to the performance metrics of the data-driven attribution models. Even though the data-driven models are shown to be better in not being rule-of-thumb, still for these data-driven, it was not quantifiable how much worse the heuristic models were compared to the data-driven ones.

The second limitation is that the RNN model could not function without L2 regularization. Other models could function with manually removing multicolinear or issue-causing predictors, but for the RNN model, this was not the case. This issue could jeopardize the comparison between the models their metrics, as the RNN model had L2 regularization while the other models did not. The last limitation is that the data-driven models have used a sample of 100.000 observations from the dataset to account for computational efficiency. However using the entire dataset or an increase in observations would create more robust results. This could be stated regarding the number of epoch in the RNN model as well. It was visible that the accuracy of the train and validation set increased significantly and unexpectedly at epoch 34. Having increased the epoch to 100 or 200 could have shown increases as well, which would have improved the general accuracy of the model.

8 Recommendations for future research

The first recommendation is that future research should focus on creating a comparison metric for heuristic and data-driven models for customer journey data. This could yield interesting results as it is currently known that data-driven models are superior, but not to what extent. These results would give a clear indication to the large number of companies that are making use of only heuristic models that using data-driven models gives a substantial increase in proper budget allocation as they perform better, as indicated by this research and the literature. Additionally, these heuristic models have been largely ignored and cast aside in recent research because they are deemed inferior to data-driven models. However, it is still important to include these models as they are still widely used and therefore relevant. Keeping these heuristic models in future research will provide a larger quantity of evidence against the use of only applying these models.

The models that were used in this research are shown to be the best performing according to the literature. However, it would still be useful to compare models that were shown to be inefficient for customer journey analysis data like the markov model. Including a larger number of data-driven and possibly heuristic models will clarify how these models compare to each other on this type of data and which model truly is most superior. Implementing these models could further support the results of this research in that the RNN model is the most efficient model for customer journey analysis. Furthermore, it would be interesting to evaluate all these models on multiple customer journey datasets using more observations and for the RNN model to use more epochs to create more robust and generalizable results.

9 References

Alin, A. (2010). Multicollinearity. *Wiley Interdisciplinary Reviews*. *Computational Statistics*, *2*(3), 370–374. https://doi.org/10.1002/wics.84

Al-Shehari, T., & Alsowail, R. A. (2021). An insider data leakage detection using One-Hot encoding, synthetic minority oversampling and machine learning techniques. *Entropy*, *23*(10), 1258. https://doi.org/10.3390/e23101258

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A. Q., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00444-8

Arava, S. K., Dong, C., Yan, Z., & Pani, A. (2018). Deep neural net with attention for multi-channel multi-touch attribution. *arXiv preprint arXiv:1809.02230*. Retrieved from https://arxiv.org/abs/1809.02230

Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, *114*, 24–31. https://doi.org/10.1016/j.isprsjprs.2016.01.011

Berman, R. (2018). Beyond the last touch: attribution in online advertising. *Marketing Science*, *37*(5), 771–792. https://doi.org/10.1287/mksc.2018.1104

Bhatta, I. (2022). Optimizing Marketing Channel Attribution for B2B and B2C with Machine Learning Based Lead Scoring Model (Doctoral dissertation, Capitol Technology University). Retrieved from https://www.proquest.com/openview/65d82057fee3c27039e2429bc1d72b00/1?pq-origsite=gscholar&cbl=18750&diss=y

Burk, S. (2006). A Better Statistical Method for A/B Testing in Marketing Campaigns. *Marketing Bulletin*, *17*. Retrieved from https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ab09d3493c8943dda5a1a79304bad79c24b12d9c

Brownlee, J. (2018). What is the Difference Between a Batch and an Epoch in a Neural Network. *Machine learning mastery*, *20*, 1-5. Retrieved from: https://deeplearning.lipingyang.org/wp-content/uploads/2018/07/What-is-the-Difference-Between-a-Batch-and-an-Epoch-in-a-Neural-Network_.pdf

Bylander, T. (2002). Estimating generalization error on Two-Class datasets using Out-of-Bag estimates. *Machine Learning*, *48*(1/3), 287–297. https://doi.org/10.1023/a:1013964023376

Chai, C. P. (2020). The importance of data cleaning: Three visualization examples. *Chance*, *33*(1), 4–9. https://doi.org/10.1080/09332480.2020.1726112

Chen, M., Mao, S., & Liu, Y. (2014). Big Data: a survey. Journal on Special Topics in *Mobile Networks and Applications/Mobile Networks and Applications*, *19*(2), 171–209. https://doi.org/10.1007/s11036-013-0489-0

Churchill, V., Li, H. A., & Xiu, D. (2024). Unraveling consumer purchase journey using neural network models. *Social Science Research Network*. https://doi.org/10.2139/ssrn.4793154

Daoud, J. I. (2017). Multicollinearity and regression analysis. *Journal of Physics. Conference Series*, *949*, 012009. https://doi.org/10.1088/1742-6596/949/1/012009

De Haan, E., Wiesel, T., & Pauwels, K. (2016). The effectiveness of different forms of online advertising for purchase conversion in a multiple-channel attribution framework. *International Journal Of Research in Marketing*, *33*(3), 491–507. https://doi.org/10.1016/j.ijresmar.2015.12.001

Efron, B. (2020). Prediction, estimation, and attribution. *International Statistical Review*, *88*(S1). https://doi.org/10.1111/insr.12409

Erickson, B. J., & Kitamura, F. (2021). Magician's corner: 9. Performance metrics for machine learning models. *Radiology: Artificial Intelligence*, *3*(3), e200126. Retrieved from https://pubs.rsna.org/doi/full/10.1148/ryai.2021200126

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2011). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(4), 463-484. doi: 10.1109/TSMCC.2011.2161285

Gaur, J., Bharti, K., & Bajaj, R. (2024). Maximizing marketing impact: heuristic vs ensemble models for attribution modeling. *Global Knowledge, Memory, and Communication*. https://doi.org/10.1108/gkmc-04-2023-0112

Gordon, B. R., Jerath, K., Katona, Z., Narayanan, S., Shin, J., & Wilbur, K. C. (2020). Inefficiencies in digital advertising markets. *Journal of Marketing*, *85*(1), 7–25. https://doi.org/10.1177/0022242920913236

Halvorsrud, R., & Kvale, K. (2017). Strengthening customer relationships through Customer Journey Analysis. In *Edward Elgar Publishing eBooks*, 183-200. https://doi.org/10.4337/9781785369483.00021

Hegde, C., Wallace, S., & Gray, K. (2015, September). Using trees, bagging, and random forests to predict rate of penetration during drilling. In *SPE Middle East Intelligent Oil and Gas Symposium* (p. D011S001R003). SPE. https://doi.org/10.2118/176792-M

Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression*. https://doi.org/10.1002/0471722146

Ji, W., Wang, X., & Zhang, D. (2016, October). A probabilistic multi-touch attribution model for online advertising. *In Proceedings of the 25th acm international on conference on information and knowledge management* (pp. 1373-1382). https://doi.org/10.1145/2983323.2983787

Kadyrov, T., & Ignatov, D. I. (2019). Attribution of customers' actions based on machine learning approach. *MPRA Paper*, *2479,* 77–88. https://publications.hse.ru/mirror/pubs/share/direct/334106305.pdf

Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, *6*(4), 312–315. https://doi.org/10.1016/j.icte.2020.04.010

Kannan, P., Reinartz, W., & Verhoef, P. C. (2016). The path to purchase and attribution modeling: Introduction to special section. *International Journal of Research in Marketing*, *33*(3), 449–456. https://doi.org/10.1016/j.ijresmar.2016.07.001

Karatas, G., Demir, O., & Sahingoz, O. K. (2020). Increasing the performance of Machine Learning-Based IDSs on an imbalanced and Up-to-Date dataset. *IEEE Access*, *8*, 32150–32162. https://doi.org/10.1109/access.2020.2973219

Kindbom, H. (2021). Investigating the Attribution Quality of LSTM with Attention and SHAP: Going Beyond Predictive Performance. Retrieved from https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1596411&dswid=-4496

Koch, C., Lindenbeck, B., & Olbrich, R. (2023). Dynamic customer journey analysis and its advertising impact. *Journal of Strategic Marketing*, 1–20. https://doi.org/10.1080/0965254x.2023.2171475

Lang, T., & Rettenmeier, M. (2017, April). Understanding consumer behavior with recurrent neural networks. In *Workshop on Machine Learning Methods for Recommender Systems*. Retrieved from https://userpage.fu-berlin.de/tlang/pub/2017-lang-rettenmeier-mlrec.pdf

Le, P., & Zuidema, W. (2016). Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs. *arXiv preprint arXiv:1603.00423*. Retrieved from https://arxiv.org/abs/1603.00423

Leguina, J. R., Rumín, Á. C., & Rumín, R. C. (2020). Digital Marketing Attribution: Understanding the user path. *Electronics*, *9*(11), 1822. https://doi.org/10.3390/electronics9111822

Lemon, K. N., & Verhoef, P. C. (2016). Understanding customer experience throughout the customer journey. *Journal of Marketing*, *80*(6), 69–96. https://doi.org/10.1509/jm.15.0420

Liu, X., Wu, J., & Zhou, Z. (2009). Exploratory undersampling for Class-Imbalance learning. *IEEE Transactions on Systems*, *Man and Cybernetics*. *Part B*. *Cybernetics*, *39*(2), 539–550. https://doi.org/10.1109/tsmcb.2008.2007853

Li, Y., Wei, C., & Ma, T. (2019). Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in neural information processing systems*, *32*. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2019/file/bce9abf229ffd7e570818476ee5d7dde-Paper.pdf

Lunardon, N., Menardi, G., & Torelli, N. (2014). ROSE: a Package for Binary Imbalanced Learning. *The R Journal*, *6*(1), 79. https://doi.org/10.32614/rj-2014-008

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*. Retrieved from https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Menardi, G., & Torelli, N. (2012). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, *28*(1), 92–122. https://doi.org/10.1007/s10618-012-0295-5

Merikanto, K. (2022). Using machine learning to predict purchase potential from customer data. Retrieved from https://www.theseus.fi/handle/10024/746946

Nisar, T. M., & Yeung, M. (2017). Attribution Modeling In Digital Advertising. *Journal of Advertising Research*, *58*(4), 399–413. https://doi.org/10.2501/jar-2017-055

Nygård, R., & Mezei, J. (2020). Automating Lead Scoring with Machine Learning: An Experimental Study. *Proceedings of the . . . Annual Hawaii International Conference on System Sciences/Proceedings of the Annual Hawaii International Conference on System Sciences*. https://doi.org/10.24251/hicss.2020.177

Probst, P., Wright, M. N., & Boulesteix, A. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery/Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery*, *9*(3). https://doi.org/10.1002/widm.1301

Rigatti, S. J. (2017). Random Forest. *Journal of Insurance Medicine*, *47*(1), 31–39. https://doi.org/10.17849/insm-47-01-31-39.1

Rout, N., Mishra, D., & Mallick, M. K. (2017). Handling Imbalanced Data: a survey. *In Advances in intelligent systems and computing* (pp. 431–443). https://doi.org/10.1007/978-981-10-5272-9_39

Rozemberczki, B., Watson, L., Bayer, P., Yang, H., Kiss, O., Nilsson, S., & Sarkar, R. (2022). The Shapley value in machine learning. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2202.05594

Sakly, S., & Ouazan, H. M. (2016). Toward a Dynamique Multitouch Attribution Model for Marketing. Retrieved from https://www.researchgate.net/profile/Sami-Sakly/publication/309416430_Toward_a_dynamic_attribution_model_for_marketing/links/580f7a0608aee15d49120d13/Toward-a-dynamic-attribution-model-for-marketing.pdf

Schwartz, R., & Stanovsky, G. (2022). On the Limitations of Dataset Balancing: The Lost Battle Against Spurious Correlations. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2204.12708

Shao, X., & Li, L. (2011, August). Data-driven multi-touch attribution models. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 258-264). Retrieved from https://dl.acm.org/doi/abs/10.1145/2020408.2020453

Sinha, R., Saini, S., & Anadhavelu, N. (2014). Estimating the incremental effects of interactions for marketing attribution. *IEEE*. https://doi.org/10.1109/besc.2014.7059518

Speiser, J. L., Miller, M. E., Tooze, J., & Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. *Expert Systems With Applications*, *134*, 93–101. https://doi.org/10.1016/j.eswa.2019.05.028

Stoltzfus, J. C. (2011). Logistic Regression: A brief primer. *Academic Emergency Medicine*, *18*(10), 1099–1104. https://doi.org/10.1111/j.1553-2712.2011.01185.x

Takase, T., Oyama, S., & Kurihara, M. (2018). Effective neural network training with adaptive learning rate based on training loss. *Neural Networks*, *101*, 68–78. https://doi.org/10.1016/j.neunet.2018.01.016

Thabtah, F., Hammoud, S., Kamalov, F., & Gonsalves, A. (2020). Data imbalance in classification: experimental evaluation. *Information Sciences*, *513*, 429–441. https://doi.org/10.1016/j.ins.2019.11.004

Thurber, K. (2016). Customer-centricity and marketing attribution: Here is why it matters and how to get started. *Applied Marketing Analytics*, *2*(2), 121-126. Retrieved from https://www.ingentaconnect.com/content/hsp/ama/2016/00000002/00000002/art00005

Tueanrat, Y., Papagiannidis, S., & Alamanos, E. (2021). Going on a journey: A review of the customer journey literature. *Journal of Business Research*, *125*, 336-353. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0148296320308584

Van Den Broeck, G., Lykov, A., Schleich, M., & Suciu, D. (2022). On the Tractability of SHAP Explanations. *Journal of Artificial Intelligence Research/the Journal of Artificial Intelligence Research*, *74*, 851–886. https://doi.org/10.1613/jair.1.13283

Vardarsuyu, M., & Sunaoğlu, Ş. K. (2022). The Journey of Customer Identification: A Systematic Literature Review and Directions for Further Investigation. *Eskişehir Osmangazi Üniversitesi İktisadi ve İdari Bilimler Dergisi*, *17*(2), 561-583. Retrieved from https://dergipark.org.tr/en/pub/oguiibf/issue/70614/1079602

Vermeer, S., & Trilling, D. (2020). Toward a better understanding of news user journeys: A Markov chain approach. *Journalism Studies*, *21*(7), 879-894. Retrieved from https://www.tandfonline.com/doi/full/10.1080/1461670X.2020.1722958

Wang, B., Liu, L. C., Koong, K. S., & Bai, S. (2009). Effects of daily and "woot‐off" strategies on e‐commerce. *Industrial Management & Data Systems*, *109*(3), 389-403. Retrieved from https://www.emerald.com/insight/content/doi/10.1108/02635570910939407/full/html?casa_token=0h1D3Py1CpEAAAAA:ZKM7HPcmnKeAuGeMDKu0-6nuS7KiQhma7CPAKDwxYd9l7PEnLnEeVc8tSgm3ccoKv1perVWAihN4RlNdjeYXsemijkPoHZF5yDm6S6W6aaSVHLxZWA

Wang, H., Yang, F., & Luo, Z. (2016). An experimental study of the intrinsic stability of random forest variable importance measures. *BMC Bioinformatics*, *17*(1). https://doi.org/10.1186/s12859-016-0900-5

Wang, X., Gong, G., Li, N., & Qiu, S. (2019). Detection analysis of epileptic EEG using a novel random forest model combined with grid search optimization. *Frontiers in Human Neuroscience*, *13*. https://doi.org/10.3389/fnhum.2019.00052

Wongvorachan, T., He, S., & Bulut, O. (2023). A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining. *Information*, *14*(1), 54. https://doi.org/10.3390/info14010054

Wu, Q., Hsu, W. L., Xu, T., Liu, Z., Ma, G., Jacobson, G., & Zhao, S. (2019, January). Speaking with actions-learning customer journey behavior. In *2019 IEEE 13th International conference on semantic computing (ICSC)* (pp. 279-286). IEEE. doi: 10.1109/ICOSC.2019.8665577

Ni, X., Fang, L., & Huttunen, H. (2021). Adaptive L2 Regularization in Person Re-Identification. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 9601-9607). IEEE. https://doi.org/10.1109/ICPR48806.2021.9412481

Yang, D., Dyer, K., & Wang, S. (2020). Interpretable deep learning model for online multi-touch attribution. *arXiv preprint arXiv:2004.00384*. Retrieved from https://arxiv.org/abs/2004.00384

Yuvaraj, C. B., Chandavarkar, B. R., Kumar, V. S., & Sandeep, B. S. (2018). Enhanced Last-Touch Interaction Attribution Model in Online Advertising. *IEEE*. https://doi.org/10.1109/discover.2018.8674079

Zafar, M. R., & Khan, N. (2021). Deterministic local interpretable Model-Agnostic explanations for stable explainability. *Machine Learning and Knowledge Extraction*, *3*(3), 525–541. https://doi.org/10.3390/make3030027

Zeiler, M. D. (2012). ADADELTA: an Adaptive Learning Rate Method. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1212.5701

Zhang, J., & Chen, L. (2019). Clustering-based undersampling with random over sampling examples and support vector machine for imbalanced classification of breast cancer diagnosis. *Computer Assisted Surgery*, *24*(sup2), 62–72. https://doi.org/10.1080/24699322.2019.1649074

Zhu, G., Wu, Z., Wang, Y., Cao, S., & Cao, J. (2019). Online purchase decisions for tourism e-commerce. *Electronic Commerce Research and Applications*, *38*, 100887. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S156742231930064X

10 Appendix

10.1 Data Overview Table

Table 10.1 Data Overview

Feature Description	Feature Definition	Data Class
client_id	Unique identification number for clients	Numeric
first_purchase_data	The moment the client was added to the system due to purchasing an item.	POSIXct
purchased_at	The exact time and date the purchase happened	POSIXct
campaign_id	A comprehensive identifier used across all campaigns to link individual messages to specific marketing efforts, enabling analysis of campaign performance and user engagement. The "id" serves as an internal reference for specific campaign types.	Numeric
message_id	Special identifier for messages.	Character
date	Date of message being sent.	POSIXct
sent_at	Date in combination with the exact time the message was sent.	POSIXct
is_opened	Binary indication if the message was opened or not	Numeric
opened_first_time_at	Date and time the message was opened for the first time.	POSIXct
opened_last_time_at	Date and time the message was opened for the last time.	POSIXct
is_clicked	Binary if a message was clicked on by a customer or not.	Numeric
clicked_first_time_at	Date and time of the first moment a customer clicked on a message.	POSIXct
clicked_last_time_at	Date and time of the last moment a customer clicked on a message.	POSIXct
is_unsubscribed	Binary indication if a person is unsubscribed or not.	Numeric
unsubscribed_at	Time and date a customer unsubscribed.	POSIXct
is_complained	Binary indication if a customer complained or not.	Numeric
complained_at	Time and date a customer complained.	POSIXct

Feature Description	Feature Definition	Data Class
is_purchased	Binary indication if the customer received a message, clicked, and purchased 24 hours after click.	Numeric
channel.email	Email is used as a channel to send a message to a customer.	Numeric
channel.mobile_push	Mobile_push is used as a channel to send a message to a customer.	Numeric
channel.web_push	Web_push is used as a channel to send a message to a customer.	Numeric
platform.desktop	A desktop is used as a device to open a message.	Numeric
platform.phablet	A phablet is used as a device to open a message.	Numeric
platform.smartphone	A smartphone is used as a device to open a message.	Numeric
platform.tablet	A tablet is used as a device to open a message.	Numeric
platform.	All other devices used to open a message are not indicated.	Numeric
message_type.bulk	The type of message send being a bulk message.	Numeric
message_type.transactional	The type of message send being a bulk message.	Numeric
message_type.trigger	The type of message send being a trigger message.	Numeric
email_provider.gmail.com	The domain part of the email (for email messages) is gmail.com.	Numeric
email_provider.mail.ru	The domain part of the email (for email messages) is mail.ru.	Numeric
email_provider.other	The domain part of the email (for email messages) is a non-mentioned email.	Numeric
started_at	The start of the bulk campaign date and time.	POSIXct
finished_at	The end of the bulk campaign date and time.	POSIXct
camp_campaign_typebulk	The type of campaign is a bulk campaign.	Numeric
camp_campaign_typetransactional	The type of campaign is a transactional campaign.	Numeric
camp_campaign_typetrigger	The type of campaign is a trigger campaign.	Numeric
camp_channelemail	Email is used as a channel to send the campaign to a customer.	Numeric
camp_channelmobile_push	Mobile_push is used as a channel to send the campaign to a customer.	Numeric

Feature Description	Feature Definition	Data Class
camp_channelmultichannel	Multiple channels are used by the campaign to send messages to customers.	Numeric
camp_channelsms	SMS is used as a channel to send the campaign to a customer.	Numeric
camp_topicevent	The campaign topic is an event.	Numeric
camp_topichappy.birthday	The campaign topic is wishing the customer a happy birthday.	Numeric
camp_topicleave.review	The campaign topic is asking the customer to leave a review.	Numeric
camp_topicoffer.after.purchase	The campaign topic is asking the customer to leave a review.	Numeric
camp_topicother	The campaign topic is different from the ones mentioned.	Numeric
camp_topicsale.out	The campaign topic is a sale out.	Numeric
is_holiday	Binary indicator if there was a holiday present on a day of customer data.	Numeric

Note: The following information, except for the data class, has been inferred from the notebook data overview page: https://www.kaggle.com/code/mkechinov/direct-messaging-campaigns-dataset-overview

11 Appendix Data Reproducibility

11.1 Data Description

The data includes email, web push, mobile push, and SMS channels. The campaign types in the data are bulk, triggers, and transactional. The data has to be cleaned to be used effectively.
The data is divided into four datasets. The first dataset is *holidays.csv*, which includes two columns, one regarding the date of the event and one regarding the sixteen most significant holidays. This data is included as holidays have a substantial effect on e-commerce (Wang et al., 2009). The conversion rate could spike during or near holidays, which could bias the results if not accounted for (Zhu et al., 2019).

The second dataset is the *client_first_purchase_date.csv.* This dataset includes two columns. One column indicates the first purchase date of a customer. This purchase automatically registers the customer in the e-commerce firm's system to use for future campaigns. The second column describes the client ID. This is the personal identification number of the client who made a purchase. This number is essential for customer journey analysis as individual customer paths and a customer's relationship to the company can be analyzed (Vardarsuyu & Sunaoğlu, 2022).

The third dataset is *campaigns.csv*. This dataset contains 19 columns with information regarding the ID a campaign belongs to. The campaign type can be bulk, a campaign regarding messages sent for sellouts and before holidays to stimulate sales and bring back customers. The second campaign type is transactional messages, which are used for some kind of information delivery process, for example, bonuses added or order delivery status changed. The last campaign type is *trigger messages,* e.g., abandoned carts, which are sent automatically based on the customer's behavior. The campaign topic, start of the campaign, end of the campaign, and numerous columns regarding the contents of the campaign message are included in this dataset. These columns will be indicated in full with their descriptions after the data cleaning process.

The fourth dataset is the *messages-demo.csv*. This dataset contains 32 columns with information regarding 10 million messages sent to clients registered in the system. This dataset is a randomized part of the complete messages file with 172 million messages. The dataset includes the following columns: campaign type, which indicates the type of campaign a message belongs to. The channel type indicates the channel used for sending the message, whether email, mobile push, or other. The exact time and date a message has been opened. If the link in the message has been clicked on and if this eventually leads to a conversion with the exact time of this conversion added.

Lastly, the unique message ID and the client ID are included. These last two columns allow for the customer path analysis as all messages and their consequences can be traced to specific customers. As with the previous dataset, the columns in this dataset that are eventually used will be stated in full after the data cleaning to avoid naming unused or impractical columns.

11.2 Data Cleaning

Data cleaning is essential in preparing data for analysis. It removes impractical columns or rows in a dataset and increases the data quality. Chai's article (2020) demonstrates this importance. The dataset *campaigns.csv* includes multiple columns that have to be removed. These columns are *ab_test*, *is_test*, *position* and *hour_limit*. The column *ab_test* indicates if a bulk campaign has been utilized for A/B testing. This form of testing is used to test multiple ideas and compare their efficiency (Burk, 2006). The column has only 1% true and 0% false values regarding campaigns. The other values are 99% null, which indicates an absence of values in this column. Therefore, this column will be excluded as the significant level of null values influences the variability of the column. This column will not contribute to customer journey outcomes. The 18th column in the dataset *is_test* indicates if a campaign states to the customer that it is a test campaign. The values for this column are distributed as 99% null and 1% false.

Additionally, the notebook section of the Kaggle source indicates that this column can be ignored for research, further proving its redundancy. The 19th column *position* indicates the position of trigger campaigns created by certain events. This column has 99% null and 1% other values, demonstrating its unuseability. The 10th column, *hour_limit,* indicates the number of messages sent per hour. This column almost entirely consists of NA values and will be removed from the *campaigns.csv* data set. Lastly, the columns regarding the subject of a message, which are columns 7 and 11 until 17, will be removed as these columns add significant dimensionality but do not add importance to the customer journey. Therefore, columns *subject_length*, *subject_with_personalization*, *subject_with_deadline*, *subject_with_emoji*, *subject_with_bonuses*, *subject_with_discount, subject_with_saleout* and *total_count* will be removed from the dataset. The same process will be applied to columns 9 and 10. *Warmup_mode* and *hour_limit* indicate when and if it has taken place that a bulk campaign message has been sent. This process has left the *campaign.csv* dataset with the remaining 16 columns.

The data of messages-demo.csv has been randomly sampled for 10% to decrease the computational cost of data cleaning, transforming, and using the data in models. The dataset messages-demo.csv includes a column called *category*. The datasets page on Kaggle indicates that this 7th column is unusable and will, therefore, be removed from the dataset. Other columns in the messages-demo.csv had to be removed as these are shown to be unusable on Kaggle, is the 1st column named *id*. This column indicates the message sequence ID. The columns created_at, and updated_at have not been clarified on Kaggle except that they were deemed unnecessary. The 10th column in this dataset *stream* indicates the string value of a device type where a message was opened. The column includes one unique value, which is *desktop*. This column with one unique value will not create value for any models discussed and will be removed to reduce the dimensionality of the dataset. The 22nd column, *hard_bounced_at,* indicates if a message has been hard bounced and will be removed due to nearly having 100% missing values. The 21st column, *is_hard_bounced,* is heavily related to this column and, therefore, will be removed because of redundancy. The 24th column, *soft_bounced_at,* indicates the time a message has been soft bounced*,* the 23rd column, *is_soft_bounced,* indicates if a message was softbounced*; the* 28th column, *blocked_at,* indicates the time a customer blocked the messages*,* and the 27th column, *is_blocked,* were in an identical situation; consequently, these four columns were also removed. This process has left the messages-demo.csv dataset with 21 columns.

The *client_first_purchase_date.csv* dataset with two columns and the *holidays.csv* dataset with two columns did not require data cleaning and will be kept in their original state with both datasets consisting of two columns.

Figure A1 indicates the data distribution after using the original 20 GB file and the product of infusion a sample of the minority class of the target variable *is_purchased* into the merged dataset of this research to create a larger minority class.

11.3 Data distributions

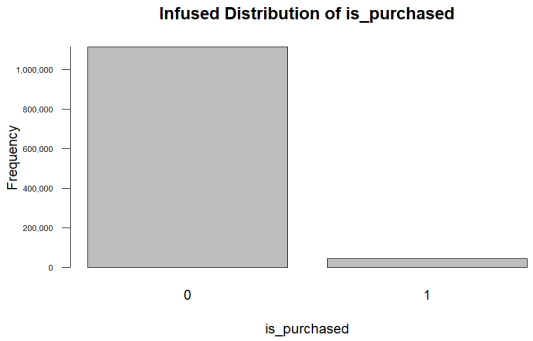

Figure A1 Infused Distribution of the is_purchased target variable

11.4 Model Preparations

The implementation of the first-touch attribution model has been prepared by creating a new object in R. that includes only the channels regarding the campaign. The reason is that the campaign channels are the first touchpoint a customer interacts with. These are catered to the customer before they even receive the message. This object will then be used to calculate the total occurrence of each campaign channel in a row with a conversion. After creating the list of the most common first-touch campaign channels, it will be necessary to calculate the average conversion rate per channel. This can be calculated by dividing the total amount of time a channel occurs in the data for each channel by the total amount of time they happen in a row with a conversion. By doing this, the channel's conversion rate, which, on average, is deemed the most important in the first-touch attribution model, can be compared to the other heuristic models.

For clarity, this conversion rate will be expressed mathematically.

$$ConversionRate_{FirstTouch} = \ Avg\ CR\ of\ Most\ Common\ First\ Touch\ Channel$$

Where avg is an abbreviation for average and CR for conversion rate.

After creating the output for this model, it will be necessary to move on to the next single-touch heuristic model, the last-touch attribution model. This model only includes the message channels. This is

because the message channels are the last touchpoint a customer can interact with before purchasing. This model also has a list created with the total occurrence of a channel in rows with conversions. Here, the conversion rate on average per channel is calculated as well. According to this model, the most important channel is the most occurring message channel in rows with conversions.
The mathematical expression for the conversion rate of this model is:

$$ConversionRate_{LastTouch} = Avg\ CR\ of\ Most\ Common\ Last\ Touch\ Channel$$

The linear attribution model has been prepared differently. This model uses multiple channels for its credit allocation. The model uses both the campaign and message channels, assuming equal importance among all channels regardless of whether they are close or far from conversion. Therefore, an object, including all channels, has been created for this model. From this object, a list of the most occurring channels in a row with a conversion is made. This list includes the corresponding average conversion rate of the channels. To calculate the final conversion rate of this model, it is necessary to calculate the number of touchpoints a customer has and the respective conversion rate of each channel. These values then need to be added together and divided by the total number of customers that conducted a purchase and, lastly, divided by the average touchpoints of all the customers to get a single conversion rate value for the linear attribution model.

$$ConversionRate_{Linear} = \frac{\left(\frac{\sum CR\ Channels\ Purchasing\ Customers}{Total\ Purchasing\ Customers}\right)}{Avg\ Touchpoints\ Customers}$$

The time-decay attribution model also uses multiple channels for credit allocation. This model will make use of all touchpoints for its calculation. The touchpoints nearest to the conversion event are deemed the most important and, therefore, will receive 50% more credits than the first campaign touchpoints, which will receive 50% fewer credits. The linear time-decay distribution has been chosen as the exponential time-decay distribution was deemed too extreme for the small time difference in campaigns and message touchpoints. This allocative distribution is deemed necessary as the message touchpoints are closer to the conversion event and, therefore have according to this model a higher importance regarding the purchasing behavior of customers. Multiplying the campaigns and messages touchpoints with 0.5 and 1.5 respectively, summing their average conversion rate values, and dividing this between the number of customers that conducted a purchase will create a single conversion rate value for the time-decay attribution model.

$$ConversionRate_{TimeDecay} = \sum_{i=1}^{n} (\frac{i}{n} Avg\ Cr\ Channel_{n-i+1})$$

Where n is the number of channels, i the channel position in the path to purchase, $Avg\ Cr\ Channel_{n-i+1}$ is the conversion rate of the(n−i+1)-th channel from the purchase event, and $\frac{i}{n}$ is the proportional weight of the channel based on its order.