Erasmus School of Economics
Bachelor Thesis - Double Bachelor in Econometrics and Economics

Multi-touch Marketing Attribution using Recurrent Neural Networks

Author: Anshul Bhat (495924)

Supervisor: Dr. Kathrin Gruber

Second Assessor: Luuk van Maasakkers

Date final version: 3 July, 2022

Abstract

In online advertising, customers are exposed to a sequence of digital channels, e.g., social media advertising, search engine advertising and optimisation, and responsive websites, before making a conversion decision, i.e., purchases, sign-ups, or subscriptions. Measuring the individual contribution of each channel to a conversion event is a critical task for marketers to maximise the cost effectiveness of their marketing strategy. Commonly used attribution approaches such as rule-based heuristics fail to model the sequential nature of the customer journey while more advanced techniques like Markov chains are unable to model long-term dependencies along the customer journey. In response, we propose a long short-term memory framework, capable of processing long-term channel interplay effects, for the multi-touch attribution problem. An attention mechanism is used to learn channel attributions from the conversion prediction objective and a second LSTM model that distinguishes between channel impressions and clicks is also proposed. Applying our analysis to two datasets, we find that the LSTM models demonstrate substantial predictive performance gains over heuristic, logistic regression and Markovian baseline models, generating slightly different attribution scores. Our results also indicate that the difference between clicks and impressions on a channel's conversion contribution is negligible.

Contents

1 Introduction

The advent of online digital advertising can be traced back to 1994 when AT&T purchased a small rectangular space on HotWired magazine's website to display the world's first banner ad, at a cost of US$30,000. The three decades since have seen a veritable explosion in the adoption of digital marketing strategies with an estimated digital ad-spend of US$183.1bn in 2021, accounting for 64.2% of total advertising expenditure (Adgate, 2021). The evolution of smart devices and the internet landscape has contributed to the development of a wide variety of online ads and digital channels, including responsive websites, search engine marketing, affiliate marketing, in-app advertising, social media, email and mobile advertising (McCarthy, 2021). Customers may be exposed to any of these channels on various devices, possibly interacting with the same channel more than once, before reaching a potential conversion, that is, a purchase, sign-up or subscription. Anderl, Becker, Von Wangenheim, and Schumann (2016) define this sequence of advertising interactions (aka touchpoints) preceding a conversion decision as an online customer journey. For marketers to maximise their return on online advertising expenditure, evaluating which ad-interaction in the customer journey actually contributed to the conversion is critical. Attribution allows firms to identify which channels offer the largest contribution to user conversions.

Common approaches to attribution include rudimentary rule-based heuristics, such as first-touch attribution (FTA) or last-touch attribution (LTA), which ignore the influence of any ad-interactions besides the first or last ones in the customer journey (Wang, Zhang, Yuan, et al., 2017). Similar rule-based solutions exists for multi-touch attribution models (MTA) as well. For example, linear attribution assigns equal credit to all touchpoints in the customer journey, while time decay assigns fixed percentages based on the chronological order of touchpoints. Not only do such approaches ignore the sequential nature of customer journeys and the timing between ad impressions (H. Li & Kannan, 2014), they also fail to incorporate any differences in the effect on user behaviour between channels. For example, newsletters stored in an email inbox can be repeatedly accessed and often drive conversion through other channels, especially by directing traffic to the company website (Breuer, Brettel, & Engelen, 2011). These effects would be undervalued by first-touch and last-touch heuristics.

To tackle the aforementioned shortcomings, several data-driven attribution models have been developed in recent years. Shao and Li (2011) proposed a bagged logistic regression model, which was then extended by Dalessandro, Perlich, Stitelman, and Provost (2012) using causal inference methods as an answer to the lack of interpretability. However, unbiased estimation of causal parameters proves difficult under certain assumptions and the model still does not consider the sequential nature of the customer journey. Kireyev, Pauwels, and Gupta (2016) developed a vector autoregressive model, which does consider the temporal factor but does not account for spillover effects, i.e. the impact of channel visits on a conversion through other channels (H. A. Li & Kannan, 2013). A survival theory-based model presented by (Zhang, Wei, & Ren, 2014) acknowledges that different digital advertising channels can have different different impacts on user conversions but makes the critical assumption that said impact invariably deteriorates over time. Anderl et al. (2016) propose a graph-based Markovian model to determine channel contribution. In their proposed framework, the sequential nature of customer journeys is modelled as first and higher order Markov chains. Transition probabilities are used to investigate channel spillover and carryover effects (the impact of channel visits on a conversion through subsequent visits to the same channel). Higher order Markov chains allow for analysis of channel (sub)sequences and associated spillover effects. To elucidate, a user's decision to convert or visit a particular channel may be dependent not just on the previous channel, but on the order of a given subset of channels visited previously. We refers to these dependencies as higher-order dependencies. Theoretically, Markov models can account for these higher-order dependencies. In practice, however, the suggested framework suffers from scalability issues due to computational limits on the order of the Markov chain. This stems from the fact that the state space increases exponentially with the order of the Markov chain (Graves, Wayne, & Danihelka, 2014).

Neural networks are another increasingly popular tool in the field of MTA problems, especially recurrent neural networks (RNN). RNN models have long been used for speech recognition and machine translation tasks for the same reason that they are apt for MTA: they provide a flexible means of processing sequential dependencies in data (Du, Zhong, Nair, Cui, & Shou, 2019). As Du et al. (2019) explain, the RNN architecture allows them to efficiently store information about the past, making them suitable to handle non-Markovian dependencies. That being said, the most important quality of an attribution model is its 'explainability'. Beyond interpreting customer journeys and accurately associating them with conversions or non-conversions, their parameters should justify these results and answer the question: Why was the customer journey associated with a conversion or non-conversion? In other words, attribution models must be able to identify which specific touchpoints in the customer journey contributed to the conversion decision and quantify these contributions. As is the case for most neural networks, simple RNNs are a black box - we do not know how individual nodes in the neural network work together to produce a final output. In the case of a base RNN model, it is virtually impossible to glean insights from node weights to ascertain the dergee of influence of each touchpoint on a customer's conversion decision. Clearly, RNNs require some additional mechanisms to obtain explainable results in the form of attribution scores for each channel. In view of the benefits and limitations of neural networks relative to frequently used attribution modelling techniques, we propose the following research question:

Can we obtain explainable results from neural network attribution models, that are comparable to commonly used techniques in an multi-touch attribution setting?

However, due to the vanishing gradient problem, deep RNNs struggle to handle long-term dependencies (Hochreiter, 1998). Long short-term memory (LSTM) was proposed as a solution to this vanishing gradient problem plaguing simple RNNs (Hochreiter & Schmidhuber, 1997). Within the field of neural network-based attribution modelling, several papers focus on the LSTM framework in their quest for higher predictive performance. Ren et al. (2018) propose a Dual-attention Recurrent Neural Network (DARNN) for MTA that learns the attribution values through an attention mechanism directly from the conversion estimation objective. Yang, Dyer, and Wang (2020) combined an LSTM-based conversion prediction model with an additive feature attribution model. Deep Neural Net with Attention for Multi-touch Attribution, or DNAMTA for short, is a similar LSTM-based conversion prediction model that also incorporates user-context information such as demographic information (Arava, Dong, Yan, Pani, et al., 2018). This brings us to the first research subquestion:

How does an LSTM-based neural network approach to conversion prediction and multi-touch attribution compare to other commonly used approaches, such as rule-based heuristics, logistic regression and Markov chains?

Further, as Ren et al. (2018) point out, the vast majority of proposed MTA solutions disregard differences in user pre-conversion behaviour, particularly the difference between ad clicks and ad impressions. An impression occurs when a digital advertisement is displayed to a visitor on their screen, whereas a click, as the name might suggest, involves the visitor clicking on the advertisement to be directed to an online property (Commercial Web Services, 2011). Most studies either treat clicks and impressions equally, or attribute conversion either solely to clicks or solely to impressions (Ren et al., 2018). The degree of user interaction with a given advertisement could substantially influence conversion, thereby providing additional valuable insights into true channel attribution. This motivates our second research subquestion:

To what extent can additional user-advertisement interaction information - for example, click or impression - improve the performance of the LSTM-based model?

2 Related Works

Data-driven MTA solutions remain a prominent area of interest for marketing scholars. In this section, we attempt to synthesise the main findings of academic literature related to attribution modelling - beginning with a broader view of the prevalent data-driven attribution models, followed by popular neural network solutions.

In the logistic regression model proposed by Shao and Li (2011), the explanatory variables are the number of touches in each advertising channel. The estimated coefficients are interpreted to reflect the attribution score of each channel. However, the exact method of deriving attribution scores from the estimated coefficients is left unexplained. Alongside their logistic regression, (Shao & Li, 2011) suggested a probabilistic model that derived attribution from first and second order conditional conversion probabilities. They found that bootstrap aggregating (bagging) improved the predictive performance of logistic regression, yielding more accurate predictions than the probabilistic model. Danaher and Van Heerde (2018) use a probit model to derive an attribution formulation that factors in carryover and interaction effects, discovering a proportional relationship between the marginal effectiveness of a channel times its number of exposures. Other studies explore the use of time series models to predict and attribute conversion. De Haan, Wiesel, and Pauwels (2016) investigate the difference between firm-initiated contact (FIC) and customer-initiated contact (CIC) with a structural vector autoregression (SVAR) model, concluding that CIC was 26.7 times more effective than FIC. The multivariate time series model put forth by Kireyev et al. (2016) suggests that the dynamic interaction between paid search and display ads increases their effectiveness over time. Xu, Duan, and Whinston (2014) present a Bayesian hierarchical framework that models advertisement clicks and purchases as dependent random events in continuous time. Their model outperforms several benchmark MTA models and demonstrates that the conversion rate measure is biased against display advertisements relative to paid search. Markov chain analysis is a popular branch of MTA solutions as well. Anderl et al. (2016) model the customer journey as first and higher-order Markov walks, using rule-based heuristics and two logit models as benchmarks. Their findings confirm that heuristic approaches can produce misleading results that can lead to incorrect attribution conclusions, and identify both carryover and spillover effects. However, due to computational limits on the order of the Markov chain, they restrict their higher-order analysis to an order of 4.

RNN attribution models, specifically LSTM-based attribution models with their capability of processing long-term sequential dependencies, offer an answer to the scalability issues associated with the Markovian framework. Although neural networks and LSTM-based models have been known to outperform simpler models such as logistic regression, they remain a comparatively unexplored area of research. Much of this has to do with the fact that LSTMs, as is the case with most neural networks, are difficult to interpret. Hence, several studies have been dedicated to developing new techniques to estimate conversion attribution with LSTMs. The Dual-attention Recurrent Neural Network implemented by Ren et al. (2018) was named as such because it applies the attention mechanism to both user clicks and conversion predictions. According to them, the attention mechanism simultaneously contributes to prediction accuracy while naturally learning the attribution of all touchpoints. DNAMTA (Arava et al., 2018) estimates attribution the same way but with a single attention layer. In addition to their attention attribution score, they make use of fractional and incremental attribution scores. The DeepMTA model proposed by Yang et al. (2020) is comprised of two stages: the first is a phased LSTM framework to model the customer journey, the second an additive features explanation model to interpret the first stage by calculating the weight of each touchpoint using Shapley values and linear regression.

3 Data

Our research is based on three publicly available clickstream datatsets. As Bucklin and Sismeiro (2009) explain, the internet allows for efficient, unobtrusive collection of information on a user's online activity, known as clickstream data, including information on customer journeys. This section provides an overall description of the datasets used, an overview of the required data pre-processing, as well as some descriptive statistics in Table 1 for context.

3.1 Kaggle Dataset

For the Kaggle dataset[1], each row represents a touchpoint in a customer journey. It contains information on 586,000 touchpoint observations, with variables denoting the user id, the marketing channel involved, the timestamp of the interaction, the nature of the interaction (impression or conversion), whether or not the user converted, and the value of the potential conversion. Assuming that each user id represents a unique customer journey, the timestamp information enables us to generate rows for separate customer journeys. If the final touchpoint for a given user id is associated with a conversion, this implies that the corresponding customer journey ended in a conversion. Further processing involves generating a channel visits variable for each channel to count the number of visits to each channel in a given customer journey. To map the last four touchpoints in a customer journey, four dummy variables channel t, with t = 1, 2, 3, 4 are generated, for each channel. These dummies are set to 1 if, counting from the end of the customer journey, the given channel exists at the specified position t, and 0 otherwise.

Table 1: *Descriptive statistics for datasets.*

Description	Kaggle	Criteo
Number of different channels	5	10
Number of impressions	586737	937840
Number of clicks	-	369785
Number of customer journeys	240108	189213
Average journey length	2.444	4.957
Number of conversions	17639	15648
Journey conversion rate	0.073	0.083

[1]https://www.kaggle.com/code/hughhuyton/multitouch-attribution-modelling/notebook

3.2 Criteo Dataset

Published by Criteo AI Labs (Diemert Eustache, Meynet Julien, Galland, & Lefortier, 2017), this dataset contains data on over 16.5 million touchpoints, a sample of 30 days of Criteo live traffic data. For each touchpoint, it provides the timestamp, the associated user id and campaign/channel id. Variables indicating whether or not a conversion occurred in the 30 days after the interaction, and whether or not the user actually clicked on the impression are also included.

Since this dataset contains nearly 700 channels, many of which are associated with very few user ids, we evaluate a subset of the data comprising only the 10 most frequently visited channels. This is vital to ensure ease of attribution interpretation. After grouping on the basis of user ids, any groups that include visits to channels outside the list of the 10 most frequent are discarded. We generate separate customer journeys using a process similar to Ren et al. (2018). For users with multiple conversions events, their touchpoint sequences are split up such that each sequence contains a maximum of one conversion, thereby representing a customer journey, while ensuring that each sequence has a minimum length of 3 and a maximum length of 20. Additional data pre-processing to identify the number of visits to each channel and the final four touchpoint channels is carried out as previously indicated. Further, we utilise the click indicator variable for each touchpoint to construct a *click_info* variable in the form of a vector. For all customer journeys, the length of the vector is equal to 10, as each index corresponds to a unique channel. The value at a given index denotes the number of clicks on that channel during the customer journey.

4 Methodology

For our methodology we make use of the baselines (last-touch, first-touch, logistic regression) and Markovian models presented by Anderl et al. (2016). This research is supplemented by our addition of two long short-term memory neural networks.

Since the true causal relationship between online channel impressions and user conversions is unknown for our real-world datasets, it is impossible to evaluate the accuracy of attribution results produced different models. A credible attribution model should, however, be able to correctly predict conversion events (Lodish, 2001). For this reason, predictive performance serves as the primary basis for comparison of our attribution models. To this end, we use the area under the receiver operating characteristic (ROC) curve (AUC) as a measure of predictive accuracy. AUC exhibits a number of desirable properties when compared to overall accuracy, chief among them being that it is invariant to a priori class probabilities (Bradley, 1997). To reduce bias of our prediction results, they are calculated across 10 cross-validation repetitions.

Before presenting the models used in our analysis, it is useful to introduce some notation. First, let $c \in \{1, 2, ..., C\}$ denote our customer. Each customer c has a unique customer journey. For customer c, their customer journey is represented by a sequence of $v \in \{1, 2, ..., V_c\}$ touchpoints, and a conversion decision y_c which is equal to 1 if the user converts and 0 otherwise. The v^{th} touchpoint in customer c's journey can be denoted by $s_{c,v}$. A touchpoint in any given customer journey involves a visit to marketing channel $k \in \{1, 2, ..., K\}$, so a visit of customer c to channel k at touchpoint v can be written as $s_{c,v} = k$.

4.1 Baseline Models

4.1.1 Rule-based Heuristics

As previously mentioned, two common rule-based heuristics for attribution modelling are last-touch (LTA) and first-touch (FTA) attribution. As their names might suggest, LTA attributes 100% of the conversion credit to the final touchpoint in the customer journey, while FTA attributes 100% of the conversion credit to the first touchpoint in the customer journey. Let $a_{c,v}$ refer to the conversion credit attribution to touchpoint v in customer c's journey.

$$a_{c,v}^{LTA} = \begin{cases} 0, & v = \{1, 2, ..., (V_c - 1)\} \\ 1, & v = V_c \end{cases} \tag{1}$$

$$a_{c,v}^{FTA} = \begin{cases} 0, & v = \{2, ..., V_c\} \\ 1, & v = 1 \end{cases} \tag{2}$$

For LTA (FTA), calculating the overall attribution A_k of conversion credits to a given channel k is a simple matter of dividing the total number of customer journeys that ended in conversion with that channel at the final (first) touchpoint, by the total number of customer journeys that ended in a conversion.

$$A_k^{LTA} = \frac{\sum_{c=1}^{C} I[s_{c,V_c} = k, y_c = 1]}{\sum_{c=1}^{C} y_c} \tag{3}$$

$$A_k^{FTA} = \frac{\sum_{c=1}^{C} I[s_{c,1} = k, y_c = 1]}{\sum_{c=1}^{C} y_c} \tag{4}$$

Given the last (first) touchpoint of a customer journey, the empirical probability of conversion can be calculated for LTA (FTA) by dividing the number of customer journeys with a given channel at the last (first) touchpoint that results in a conversion, by the total number of customer journeys with that channel at the last (first) touchpoint.

$$p^{LTA}(y_i = 1 | s_{c,V_c}) = \frac{\sum_{c=1}^{C} I[s_{c,V_c} = k, y_c = 1]}{\sum_{c=1}^{C} I[s_{c,V_c} = k]} \tag{5}$$

$$p^{FTA}(y_i = 1 | s_{c,1}) = \frac{\sum_{c=1}^{C} I[s_{c,1} = k, y_c = 1]}{\sum_{c=1}^{C} I[s_{c,1} = k]} \tag{6}$$

4.1.2 Logistic Regression

The choice of the non-linear logistic regression functional form is justified insofar as the dependent variable, i.e. the conversion decision, is binary. Unfortunately, existing literature does not clearly delineate a method to extract attributions from logistic regression coefficient estimates. For that reason, they are primarily used as baselines for predictive accuracy comparisons against the other models, in the context of this study.

The first of our logistic regressions uses the number of visits to each channel in a customer journey as explanatory variables to model the probability of a conversion, as implemented by Shao and Li (2011). The variable $x_{c,k}$ represents the number of times customer c visited channel k along their customer journey. The model can be specified as follows:

$$logit(y_c) = \alpha + \sum_{k=1}^{K} \beta_k x_{c,k}, \tag{7}$$

where

$$x_{c,k} = \sum_{v=1}^{V_c} I[s_{c,v} = k], \tag{8}$$

As before, we can estimate the conversion probability of a customer given their touchpoint history:

$$\hat{p}^{log1}(y_c = 1 | \{x_{c,1}, x_{c,2}, ..., x_{c,K}\}) = \Lambda(\hat{\alpha} + \sum_{k=1}^{K} \hat{\beta}_k x_{c,k}), \tag{9}$$

where $\Lambda(x) = \frac{1}{1+e^{-x}}$, the logistic function. As explained in Section 2, this model fails to appropriately model the sequential nature of the customer journey. The second logistic regression attempts to tackle this shortcoming by including the order of the final four touchpoints as the explanatory variable. This is accomplished using the dummy variable $d_{c,k,t}$ which is set to 1 if customer c visits channel k at touchpoint t in their customer journey, and 0 otherwise. In this case, for customer c, $t \in \{(V_c - 3), (V_c - 2), (V_c - 1), (V_c)\}$, an element of the subset of the last 4 touchpoints in the customer journey. The second logit model can be specified in the following manner:

$$logit(y_c) = \gamma + \sum_{k=1}^{K} \sum_{t=(V_c-3)}^{V_c} \delta_{k,t} d_{c,k,t}. \tag{10}$$

Similarly, given the final four touchpoints of a customer journey, the conversion probability can be calculated:

$$\hat{p}^{log2}(y_c = 1|\{d_{c,1,(V_c-3)}, ..., d_{c,K,V_c}\}) = \Lambda(\hat{\gamma} + \sum_{k=1}^{K} \sum_{t=(V_c-3)}^{V_c} \hat{\delta}_{k,t} d_{c,k,t}). \tag{11}$$

4.2 Markov Chains

The following Markovian attribution framework draws directly from the structure outlined in Anderl et al. (2016). In their paper, customer journeys are modelled as chains in a directed Markov graph. Markov chains are probabilistic models that describe sequences of events or system states, predicated on the notion that the probability of a system transitioning to a given state, depends solely on the current state of the system. Markov graphs are represented by a set of states, S, and a transition matrix, W, containing the probabilities of the system moving from a given state to any other state in the state space, including that state itself. For example, in Markov graph $M = \{S, W\}$:

$$S = \{s_1, s_2, ..., s_n\} \tag{12}$$

$$w_{i,j} = p\left(X_t = s_j \mid X_{t-1} = s_i\right), \quad 0 \leq w_{i,j} \leq 1, \quad \sum_{j=1}^{N} w_{i,j} = 1 \quad \forall i \tag{13}$$

In the first order Markov graph, a state correpsonds to one channel. Anderl et al. (2016) also add three special states: a START state to denote the starting point of every customer journey, a CONVERSION state for customer journeys associated with a successful conversion, and an absorbing NULL state for customer journeys without a conversion. Furthermore, the transition probability w_{k_1,k_2} denotes the probability that a visit to channel k_1 at a given touchpoint, is followed by a visit to channel k_2 at the next touchpoint. For a given sample of customer journeys with channel set or states $H = \{h_1, h_2, ..., h_n\}$, an example Markov graph could be specified as $G = \{E, F\}$, where state space $E = \{START, CONVERSION, NULL\} \cup H$ and F is the transition matrix. This transition matrix can be computed empirically from the data.

We now move to the process of assigning conversion credits to each channel. In Anderl et al. (2016), attribution is based on an ad factor called the 'removal effect'. Simply put, the 'removal effect' of a channel k (RE_k) is the change in the probability of reaching the CONVERSION state from the START state, after the removal of k from the channel set. It is further proved that the 'removal effect' is equivalent to the product of two other ad factors, namely, 'visit' and 'eventual conversion' ad factors. For channel k, 'visit' indicates the probability of ever passing k on a random walk beginning in the START state, while 'eventual conversion' refers to the probability of reaching the CONVERSION state from k. Although Anderl et al. (2016) do not clearly indicate their preferred method of deriving the 'removal effect', we opt for the stochastic simulation method available in the ChannelAttribution package (Altomare, Loris, & Altomare, 2016). This involves using the empirically computed transition probabilities to simulate customer journeys in the form of random walks beginning in the START state, as well as random walks beginning with each channel. The 'eventual conversion' and 'visit' values for each channel are estimated on this set of simulations. Once the 'removal effects' have been obtained, the subsequent attribution ($A_{k,Markov}$) reported for each channel is equal to its 'removal effect' as a percentage of the sum of 'removal effects' for all channels. This ensures that attributions add up to 1, allowing for easier comparison of models.

$$A_k^{Markov} = \frac{RE_k}{\sum_{k=1}^{K} RE_k} \tag{14}$$

In response to studies suggesting web users exhibit behaviour that may violate the Markov assumption (Chierichetti, Kumar, Raghavan, & Sarlos, 2012), Anderl et al. (2016) formulate higher order Markov models. For a Markov model of order r, transition probabilities now depend on the past r observations. Key to the higher order framework is recognising that a Markov chain of order $r > 1$ across state space H can be represented as a first order Markov chain across state space H_r, composed of r-tuples of states in H. For example, the transition probability $P(X_t = h_a \mid X_{t-1} = h_b, X_{t-2} = h_c, X_{t-3} = h_d) \equiv P(X_t = \{h_a, h_b, h_c\} \mid X_{t-1} = \{h_b, h_c, h_d\})$. The 'removal effect' of each tuple can be calculated by applying the aforementioned simulation technique to this first order translation. The 'removal effect' of each channel k is then obtained by averaging the 'removal effects' of all tuples containing channel k at their final index.

Conversion predictions can also be extracted from a Markov graph. In the case of a first order model, the estimated conversion probability is equal to the transition probability of moving from the channel at the final touchpoint of a given customer journey, to the CONVERSION state. Analogously, for a higher order (r) graph, this is equal to the transition probability of moving from the final r-tuple of channels in the customer journey, to the CONVERSION state.

$$\hat{p}^{Markov}(y_c = 1|s_{c,V_c} = k) = p(X_t = CONVERSION|X_{t-1} = s_{c,V_c} = k) \tag{15}$$

$$\hat{p}^{Markov}(y_c = 1|\{(s_{c,(V_c-r)} = k_i), ..., (s_{c,V_c} = k_{i+r})\}) = p(X_t = CONVERSION|X_{t-1} = \{k_i, ..., k_{i+r}\}) \tag{16}$$

We make use the of the publicly available ChannelAttribution Python package to calculate transition probabilities and implement the stochastic simulation method to derive removal effects (Altomare et al., 2016).

4.3 LSTM-based Models

4.3.1 Recurrent Neural Networks

Recurrent neural networks, or RNNs, are a class of neural networks for processing sequential data, typically in the form of vectors x_t where $t \in \{1, ..., T\}$ (Goodfellow, Bengio, & Courville, 2016). Unlike feedforward neural networks, RNNs incorporate a feedback loop that enable them to retain information from previous events. RNNs can be represented as a chain of repeating modules of a neural network, each of which take an input vector, x_t, and produce an output vector, hidden state h_t. However, the output of a module is not influenced by the current input alone, but also by the history of previous inputs (Karpathy, 2015). As the network learns to use the previous state as a summary of past input sequences (Goodfellow et al., 2016):

$$h_t = f(x_t, h_{t-1}, \theta) \tag{17}$$

Predictions can be derived from RNN outputs h_t by first applying an appropriate activation function, such as the hyperbolic tangent ($tanh$), and then multiplying the result by a weight matrix V, for hidden layer to output layer connections (Goodfellow et al., 2016). It is also possible to tailor RNNs to various input-output size combinations. Examples include sequence output (one to many), sequence input (many to one), and sequence input and output (many to many).

4.3.2 Long Short-Term Memory

Although RNNs can store information from past inputs to produce current outputs, as the gap between relevant past inputs and and desired present outputs increases, their performance deteriorates. This issue was first noticed by Hochreiter (1998), who termed it the vanishing gradient problem. During backpropagation, the error gradients at deeper layer in the RNN are calculated as the product of gradients of all the nodes' activation functions (typically sigmoid functions) with respect to their weights. As the number of layers in the network increases, the value of this product begins to approach 0, and the gradient vanishes as it moves backwards through the net, leading to poor weight updates and ineffective training.

The Long short-term memory (LSTM) network is a special type of RNN, proposed by Hochreiter and Schmidhuber (1997) as a solution capable of learning long term dependencies. The fundamental differences between RNN and LSTM lies in the structure of their repeated modules. Generally, RNNs contains a single activation layer, whereas LSTM modules contain four interacting layers and a cell state in addition to the hidden state. The following description of an LSTM network largely adheres to the notation and structure outlined in Olah (2015).

Three gate units, based on sigmoid layers, are used to control the flow of information through the cell state. The first gate, appropriately named the 'forget gate', determines what information is retained by the cell state and what is discarded from it. For every element in the previous cell state vector c_{t-1}, a sigmoid layer σ in the 'forget gate' outputs a value between 0 and 1 indicating the proportion of information to be retained (0 being retain nothing, 1 being retain everything). This layer f_t takes the previous hidden state h_{t-1} and the current input vector x_t as inputs.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{18}$$

The 'input gate' decides what new information is to be stored in the cell state, and is composed of two parts. In the first, the current input vector x_t and the previous hidden state h_{t-1} are passed through a sigmoid layer to once again identify the important elements (i_t). In the second, the same information is passed through a *tanh* layer to generate a candidate vector $\tilde{c}_t$ that could be added to the cell state.

$$\begin{aligned} i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \\ \tilde{c}_t &= \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_c\right) \end{aligned} \tag{19}$$

The next to step is to actually compute the new cell state vector c_t by multiplying the previous cell state c_{t-1} by f_t (forgetting unimportant information) and adding $i * \tilde{c}_t$ (storing only important new information).

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{20}$$

The final 'output gate' ascertains which elements of the current cell state c_t are relevant for the output by once again passing the previous hidden state h_{t-1} and the current input x_t through a sigmoid layer in o_t. The cell state c_t is subjected to a hyperbolic tangent transformation before being multiplied by this gating signal to produce the final output hidden state h_t.

$$\begin{aligned} o_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \\ h_t &= o_t * \tanh\left(c_t\right) \end{aligned} \tag{21}$$

4.3.3 Attention

We employ an attention mechanism to derive attribution results for each channel from our LSTM outputs. Attention was first introduced by Bahdanau, Cho, and Bengio (2014) to improve the performance of encoder-decoder architectures used in sequence-to-sequence (seq2seq) neural networks, in which two RNNs work in combination to transform one sequence into another. Such models are commonly used for neural machine translation tasks. The encoder converts an input sequence into a vector, which is then read by the decoder and transformed into the desired output. Seq2seq models usually encounter difficulty processing long input sequences as they only use the final hidden state output of the encoder as the context vector for the decoder. This mean that the encoder condenses the entire input sequence in a single hidden state vector. The longer the sequence, the more difficult it becomes to capture all contextual information or the relation between elements, and the poorer the performance of the seq2seq model. The attention mechanism solves this by utilising all the encoder hidden states across the sequence during the decoding process. The decoder has access to the entire input sequence and can select or pay ‘attention’ to specific elements to generate an output.

There are two types of attention mechanisms: Bahdanau attention (Bahdanau et al., 2014) and Luong attention (Luong, Pham, & Manning, 2015). Their differences lie in the calculation of an alignment score and the choice of decoder hidden state. The alignment score $score_t$, quantifies the ’attention’ the decoder pays to each encoder output. The alignment score of the Bahdanau model is given by:

$$\tilde{score}_t = v_a(\tanh(\mathrm{W_a}(s_{t-1}, H))), \tag{22}$$

where H denotes the encoder hidden states, s_{t-1} denotes the previous decoder hidden state, and v_a and W_a are weight matrices to be learned. Since we are only dealing with a single output, the previous decoder hidden state is the initial decoder hidden state. The Bahdanau alignment function is also referred to as the ’concat’ function (Luong et al., 2015) or ’additive attention’ (Vaswani et al., 2017) for the manner in which the encoder hidden states are concatenated with the initial decoder hidden state.

In addition to the ’concat’ alignment function, Luong et al. (2015) specify two multiplicative’ alternatives:

$$\tilde{score}_t = s_t^T H \tag{23}$$

$$\tilde{score}_t = s_t^T W_a H. \tag{24}$$

The *dot* alignment score (23) is a simple dot product of the encoder hidden states H and current decoder hidden state s_t, while the *general* alignment score (24) is a dot product of the decoder hidden state s_t and a linear transformation of the encoder states H.

For our analysis, we will use Bahadanau additive attention. We now provide a general overview of the attention process. First, the encoder generates a hidden state for each element in the input sequence. The initial hidden state of the decoder is set equal to the final hidden state of the encoder. The following step involves calculation of an alignment score $score_t$, which quantifies the ’attention’ the decoder pays to each encoder output. Luong proposes three specifications for the alignment scoring function. The alignment scores or attention weights are run through a softmax layer so that they all add up to 1.

$$score_t = \mathrm{softmax}(\tilde{score}_t) \tag{25}$$

A context vector w_t is created by multiplying the encoder outputs by the attention weights.

$$w_t = score_t H \tag{26}$$

4.3.4 Model Architecture

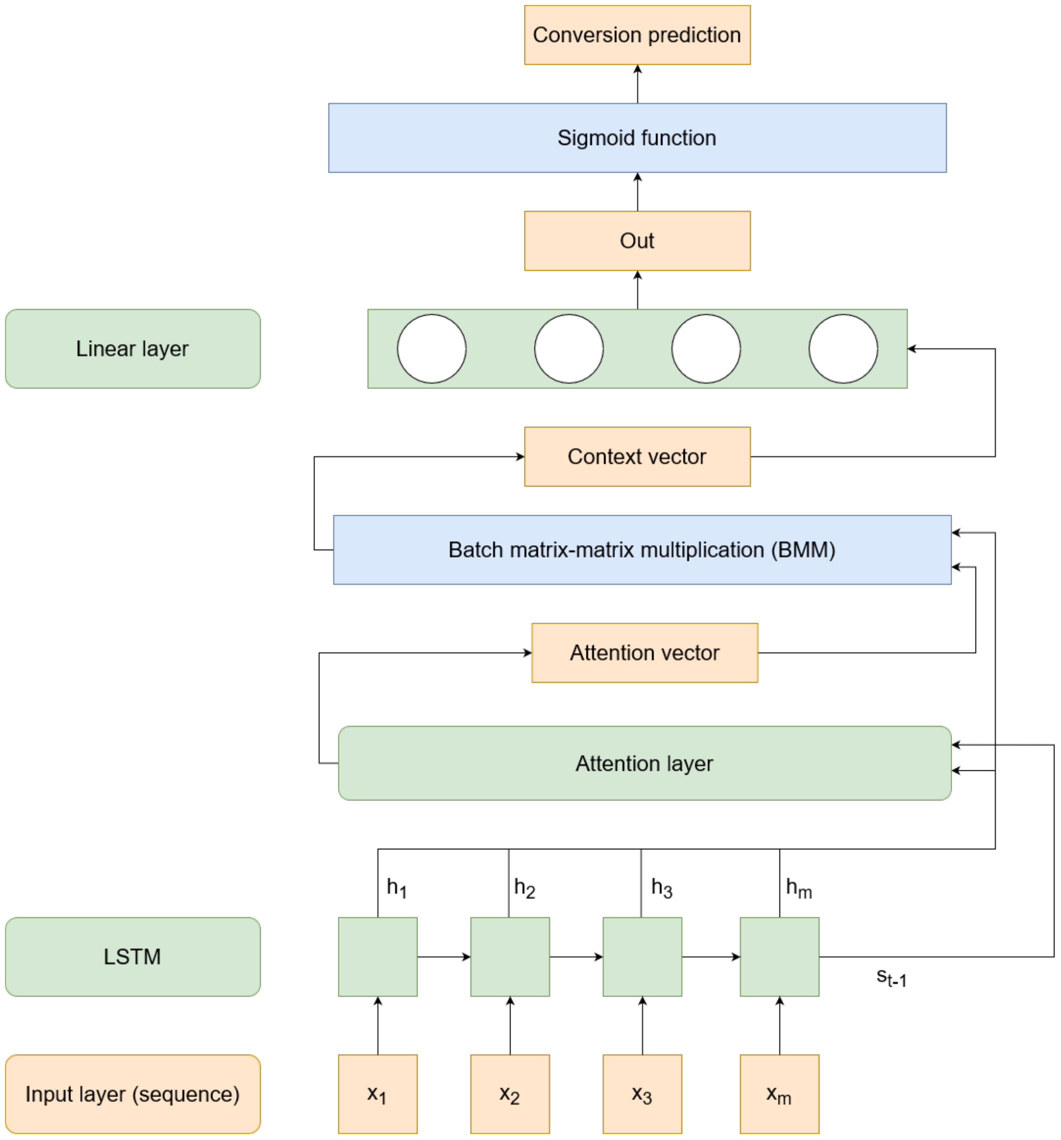
As we wish to predict a binary output (conversion decision) given an input sequence (the customer journey), we implement a many-to-one encoder-decoder network with attention in PyTorch. The 'PackedSequence' PyTorch class allows us to consider variable length inputs. Variable length input sequences in the same batch are zero-padded such that all inputs are the same length, equal to that of the longest input. Additionally, the original length of each sequence is stored in the 'PackedSequence' object. Input sequences are further transformed into sequences of one-hot encoded vectors to improve predictive accuracy.

The encoder module uses a single layer bidirectional LSTM. Bidirectional recurrent neural networks connect two hidden layers in opposite directions, training in positive and negative time directions along the input sequence. The encoder outputs a tensor containing all the hidden states for each element in the sequence, and a tensor containing the final hidden state for each element in the sequence. The latter, as previously explained, is used as the initial hidden state for the decoder. The attention module calculates attention weights as detailed in the previous section, outputting an attention vector for each input sequence. Since several input sequences in a batch have been zero-padded and we do not want to pay any attention to these pads, we use a masking function to ensure that attention is only calculated over the original length of the input sequence. The decoder calculates a context vector for each input by multiplying the encoder output and the attention weights. Passing the context vector through a linear layer followed by a sigmoid layer yields our conversion predictions. A visual representation of the base LSTM model can be found in Figure 1.

For our second model incorporating click information, the network is almost identical to the 'non-click' network specified above. Additional steps include generating a click vector for each input sequence or customer journey. The length of this vector is equal to the number of unique channels in the dataset and each element denotes the number of clicks in a given channel along a customer journey. This vector is then fed through a linear layer, the output of which is concatenated with conversion prediction obtained in the final step of the 'non-click' network. The concatenated vector is passed through another linear layer to generate the final, click-incorporating conversion prediction.

Both models must also include a mechanism to account for the class imbalance problem. Class imbalance describes a situation wherein one or more classes are more frequently occurring than

Figure 1: *Visual representation of base LSTM model.*



other classes. Imbalanced datasets introduce a bias in the learning process, resulting in predictions that are skewed towards the majority class(es). Undersampling and oversampling are two methods for handling class imbalance, however, both have their drawbacks. Undersampling involves deleting samples from the majority class which implies a loss of valuable information, whereas oversampling involves adding samples to the minority class which could lead to overfitting and longer training times. Instead, we address class imbalance by adding a sample weight parameter to our loss function, thereby leaving the distribution of our data intact. By attaching higher weights to a minority class and a majority weight for a majority class, we introduce a stronger penalty for minority class probability prediction errors, mitigating the aforementioned bias. Binary cross-entropy was chosen as the loss function, as is appropriate for a classification task. In both the

Kaggle and Criteo datasets, the conversions are vastly outnumbered by non-conversions. Therefore, we attach a higher weight to conversions which is equivalent to the ratio of non-conversions to conversions.

For both networks, the widely popular Adam optimiser was used with a learning rate between 0.01 and 0.001 depending on the results for each dataset. After some experimentation, it was found that a batch size of 128 and 5 epochs yielded good results.

4.3.5 Attention Attributions

For each input batch, our LSTM model outputs an attention matrix the same length of the batch, in which each row contains the attention weights for the input sequence in the corresponding row of the input batch. Each element in a given input sequence denotes a touchpoint in the customer journey and accordingly, each element in its attention weight vector denotes the attribution of that touchpoint. We represent the attribution of the v'th touchpoint in customer c's journey as $\alpha_{c,v}$. The unnormalised attribution $\hat{A}_{k,LSTM}$ for a given channel k is equal to the mean attribution of channel k across all customer journeys containing that channel ending in a conversion:

$$\hat{A}_{k,LSTM} = \frac{\sum_{c=1}^{C_k} \sum_{v=1}^{V_c} y_c \alpha_{c,v} I[s_{c,v} = k]}{C_k}, \tag{27}$$

where C_k is the number of customer journeys containing channel k, y_c is a binary variable indicating a conversion and $s_{c,v}$ denotes the v'th touchpoint in customer c's journey. For comparison of our attribution results to other models we normalise $\hat{A}_{k,LSTM}$.

$$A_{k,LSTM} = \frac{\hat{A}_{k,LSTM}}{\sum_{k=1}^{K} \hat{A}_{k,LSTM}} \tag{28}$$

5 Results

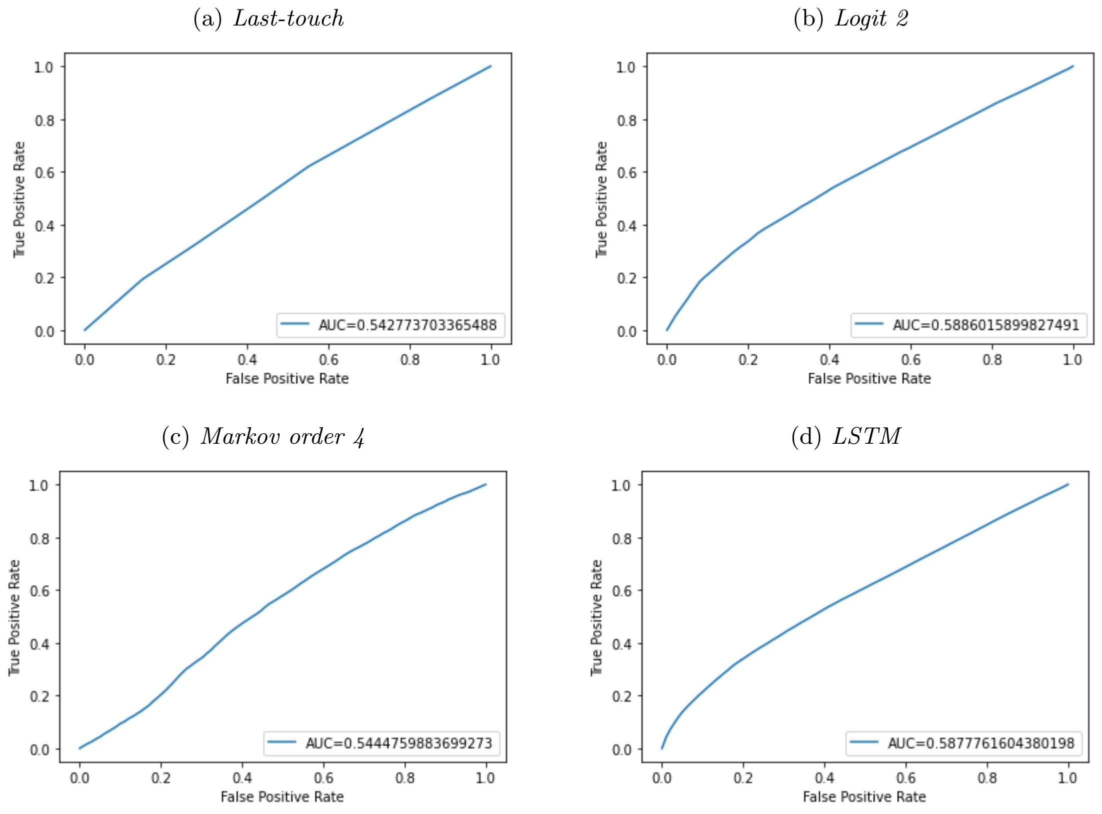
5.1 Predictive Accuracy

The first step in analysing our results is establishing the credibility of our proposed LSTM attribution model. To do so, we compare its conversion predictive performance against that of the baseline models and Markov models presented in Anderl et al. (2016). As mentioned in the Methodology section, we use the (area under the) ROC curve as a measure of predictive accuracy. The ROC curve visualises the tradeoff between the sensitivity (true positive rate or TPR) and the specificity (1 - false positive rate or FPR). Sensitivity indicates the model's ability to correctly predict conversions, while specificity indicates the model's ability to correctly predict non-conversions. Ideally, models should be both specific and sensitive which is demonstrated by a TPR tending to 1 and an FPR tending to 0. Given that ROC curves plot the FPR on the x-axis and TPR on the y-axis, better performing models yield curves close to the top-left of the graph. For random classifiers, the TPR and FPR are equivalent and the ROC lies along the 45-degree diagonal. ROC performance can be summarised by a scalar value, the area under the ROC curve or AUC. The higher the AUC, the better the predictive performance of the model. Random guessing has an AUC of 0.5 while a perfect predictor has an AUC of 1. The predictive accuracy results can be found in Table 2 while the out-of-sample ROC curves can be found in Figures 2 and 3. Note that, unless otherwise specified, predictive performance refers to the out-of-sample predictions.

Table 2: *Predictive accuracy.*

						Markov					
Sample	Dataset	LTA	FTA	Logit 1	Logit 2	Order 1	Order 2	Order 3	Order 4	LSTM	LSTM (click)
In-sample	Kaggle	0.544	0.544	0.587	0.589	0.495	0.524	0.539	0.567	0.589	na
	Criteo	0.673	0.666	0.686	0.688	0.672	0.675	0.676	0.675	0.696	0.706
Out-of-sample	Kaggle	0.543	0.544	0.586	0.589	0.516	0.546	0.550	0.544	0.588	na
	Criteo	0.673	0.666	0.686	0.687	0.639	0.638	0.632	0.598	0.695	0.705

Figure 2: *AUC, ROC curves for Kaggle dataset.*



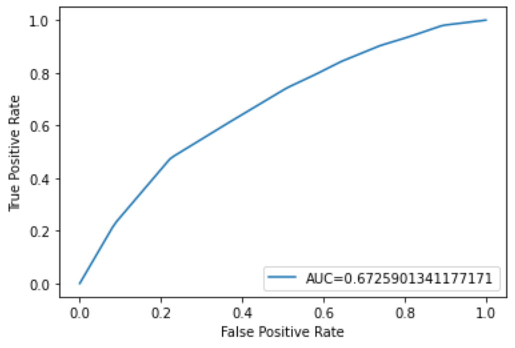
We find that the LSTM model (without click information) achieves better in-sample and out-of-sample predictive performance than nearly all the other models for both the Kaggle and Criteo dataset. The only exception is the logit 2 model for the Kaggle dataset. Notably, in terms of out-of-sample predictions for the Criteo dataset, the Markov models are outperformed not only by the LSTM model, but also the baseline heuristics and logistic regressions. For the Kaggle dataset, the Markov model performance is comparable to heuristics but still below that of the logistic regressions. Markov model conversion predictions are based on empirically calculated transition probabilities, making them very sensitive to class imbalance. That the number of conversions are vastly outweighed by the number of non-conversions (see Table 1) for both datasets, could suggest a reason for the relatively poor performance of the Markov models. By contrast, the LSTM framework accounts for the class imbalance by adding a weight parameter to its loss function which contributes to the improvements in its predictive power. Furthermore, for the Criteo dataset, as the order of the Markov model increases, predictive performance decreases. For the Kaggle dataset on the other hand, predictive performance increases up to order 3 and then decreases. One could infer that the exact order of ad-interactions has more of an influence on conversions for the Kaggle dataset than the Criteo dataset. Alternatively, higher order Markov models have too many parameters and could be overfitting the training data, as evidenced by the difference between the in-sample and the out-of-sample predictive performance of the Markov models for the Criteo dataset.

Because heuristics like first-touch and last-touch attribution focus only on a single touchpoint in the customer journey, theoretically, they tend to perform a lot better when datasets contain a large number of short customer journeys comprising only one channel visit before the conversion decision. Table 2 indicates a relatively high predictive accuracy for the heuristic models, approximately equal to the LSTM model performance for the Criteo dataset, and to the best performing Markov model for the Kaggle dataset. The latter result seems reasonable considering an average customer journey length of 2.44 touchpoints in the Kaggle dataset (see Table 1). Lower average journey lengths imply a larger number of short customer journeys. Notice too that the difference between the predictive accuracy of the heuristic model and that of the optimal model is much larger for the Kaggle dataset than the Criteo dataset. Based on the aforementioned theory, the relatively superior performance of the heuristic for the Criteo datatset could be attributed to a potentially shorter average customer journey length. However, Table 1 shows that the average customer journey length for the Criteo dataset is more than twice that of the Kaggle dataset, implying a lower proportion of short customer journeys. Coupled with the impressive performance of the logit models over the (higher-order) Markov models, these results reiterate the notion that for the Criteo dataset, the selection of channels a user interacts with has a larger impact on conversion decisions than the order in which the user interacts with them.

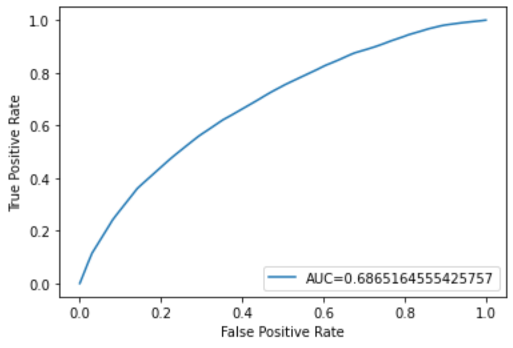
To assess the benefit of adding click information to our model, we direct our attention to graphs (d) and (e) in Figure 3, and to the last two columns in Table 2. Although incorporating click information improves the performance of our LSTM model for the Criteo dataset, the difference is not substantial (an AUC increase of 0.01). Considering that more than a third of all ad-interactions in the dataset were clicks, we can disregard the possibility that the available click information was

Figure 3: *AUC, ROC curves for Criteo Dataset*

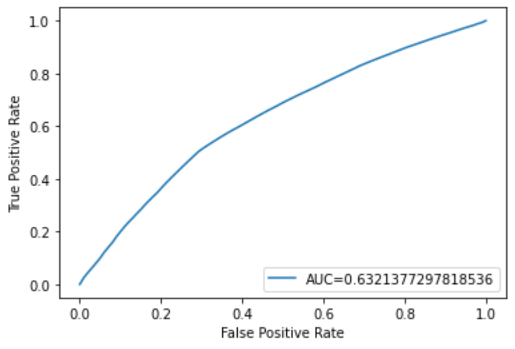
(a) *Last-touch*



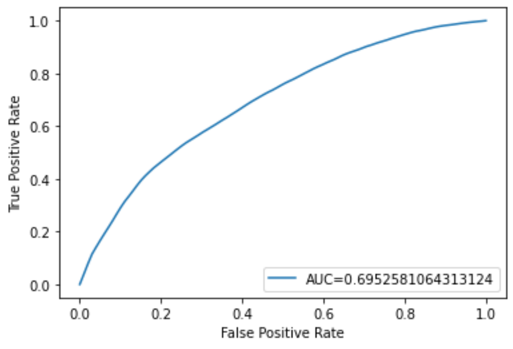
(b) *Logit 2*



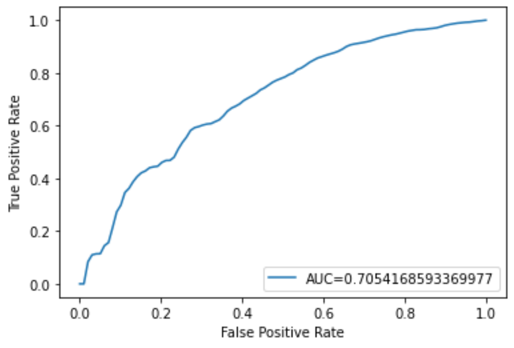
(c) *Markov order 3*



(d) *LSTM*



(e) *LSTM with click info*



insufficient for the neural network to learn. Rather, it appears that a click on a particular channel has little to no additional effect on user conversion compared to a regular channel impression.

The core takeaways from our predictive performance analysis are as follows. In terms of out-of-sample performance, the LSTM model outperforms all the other baseline and Markov models for the Criteo dataset, and all but the logit 2 model for the Kaggle dataset. This accuracy in its predictions lends credibility to its attribution scores. The Markov models are among the poorest performing models, a results which could be attributed to the class imbalance between conversions and non-conversions for both datasets. Finally, adding customer click information to the LSTM model yields no considerable predictive accuracy improvement over the base LSTM model.

5.2 Attribution Results

Next, we take a closer look at the attribution results of our proposed LSTM framework and compare it to the attribution of heuristic baselines and the Markovian framework. The attribution scores for the Kaggle dataset can be found in Table 4 and in Table 6 for the Criteo dataset. Our first observation is that compared to the Kaggle dataset, the spread between LSTM attribution scores and the average attributions scores of the other models for the Criteo dataset is far smaller. This is indicated by a lower root mean squared difference between LSTM attributions and the average of the heuristic and Markov attributions for the Criteo dataset (3.28 percentage points) than the Kaggle dataset (7.54 percentage points). The comparison also reveals that for both datasets, the LSTM models flatten the distribution of attribution scores across all channels, leading to a more even share of contribution between all the channels. We find that the standard deviation of the LSTM attributions scores for the Kaggle dataset is 4.918 percentage points, more than 2 percentage points less than the most evenly distributed heuristic or Markov model. The smoothing effect by the LSTM models is less pronounced for the Criteo dataset, with LSTM attributions standard deviations of 6.739 percentage points (without click information) and 7.047 percentage points (with click information), against an minimum standard deviation of 7.690 percentage points for the heuristic or Markov models.

Table 3: *Estimation results for Logit model 1, Kaggle dataset.*

Channel	Coefficient (β)	Odds ratio (e^{β})
Intercept	-2.726***	
Facebook	0.078***	1.081
Instagram	0.079***	1.082
Online display	0.038***	1.039
Online video	0.096***	1.100
Paid search	0.035***	1.036

Note: *p < 0.1, **p < 0.05, ***p < 0.01

Table 4: *Attribution results for Kaggle dataset, in percentages.*

			Markov				
Channel	FTA	LTA	Order 1	Order 2	Order 3	Order 4	LSTM
Facebook	29.350	30.053	30.558	29.864	29.595	29.405	20.468
Instagram	13.204	12.722	17.498	18.412	17.932	18.000	15.827
Online Display	12.246	12.127	12.083	11.829	11.398	11.485	17.233
Online Video	18.232	19.321	16.293	15.820	16.492	16.717	28.261
Paid Search	26.969	25.778	23.569	24.075	24.582	24.394	18.211

We then turn our attention to the Kaggle dataset for which the Logit 1 estimation results and model attribution results can be found in Tables 3 and 4. Attribution scores derived from the baselines and Markovian frameworks show little variation, with a maximum difference of roughly 5 percentage points between the highest and lowest attribution score for any given channel. All six of these models rank Facebook, Paid Search and Online Display as first, second and last in terms of attribution, while Instagram and Online Video, nearly equal in terms of contribution, share the third and fourth spots. The LSTM attribution scores on the other hand deviate significantly from the average attribution scores of the other models. It attributes the most conversion credits to Online Video, the least to Instagram, and a roughly equal amount to Online Display and Paid Search. As mentioned previously, we have no ground truth to compare these attribution scores to, making it impossible to judge how accurate they are. However, our logit model results and existing attribution theory could provide some justification for these results. As Table 3 shows, the Online Video coefficient is significant and the largest of all channels. The fact that it is the stronger predictor of conversion events could provide some explanation for its high LSTM attribution score. Xu et al. (2014) found that the last-touch heuristic is biased towards search engine advertising. Our results for Paid Search are consistent with their research, as illustrated by the 7 percentage point difference between the last-touch and LSTM attribution scores for paid search. With regards to the comparatively high LSTM attribution score for Online Display, we turn to Ilfeld and Winer (2002) for an explanation. According to their research, as display advertising can be considered a form of banner advertising aimed at raising brand awareness rather than driving final conversions, they are often underrepresented by single-touch heuristic approaches. In Anderl et al. (2016), for the two datasets including the Display channel, the Markov order 3 model does attribute a greater percentage of conversion credits to the Display channel than both first-touch and last-touch models; however, the absolute difference is small ($<$ 2 percentage points). A similar phenomenon can be found in our results as well. Along with online display ads, Ilfeld and Winer (2002) also consider social media, such as Facebook and Instagram, a form of banner advertising. Our results for the Instagram channel appear to follow the same pattern as our Online Display results, that being underestimation by the heuristic models relative to the Markov and LSTM models, in line with Ilfeld and Winer (2002). The results for the Facebook channel on the other hand stand in contrast to their findings, with an LSTM attribution score approximately 10 percentage points lower than the heuristic attribution scores. Furthermore, Ghose and Todri-Adamopoulos (2016) show that branding pre-roll advertising, more specifically a 15-30 second video advertisement, though having a strong effect on driving traffic to the advertiser's website though organic search, has little to no effect on a customer' conversion probability. Meanwhile, our LSTM model attributes the largest percentage of conversion credits to Online Video. These opposing results could be explained by differences in video placement (e.g. website pop-ups or autoplay before a user-selected video), video duration, the goal of the video ad campaign (e.g. sales or website traffic) and a multitude of other factors.

The Criteo dataset does not specify the exact form of online channel, instead, each channel is assigned a number. The lack of information about the various channel structures renders comparison of individual channel attribution scores meaningless. Looking at Table 6, we once again notice that attribution scores derived by the baseline and Markov models are quite similar for all channels. Another interesting observation is that all the channel coefficient estimates from the Logit 1 model are negative (see Table 5). An increase in the number of visits to any channel is therefore associated with a decrease in the probability of a conversion event. As such, shorter customer journeys in the Criteo dataset are more likely to end in a conversion than longer ones. LSTM with click attribution scores are largely similar to LSTM attribution scores, with the root mean squared difference between the two less than 0.5 percentage points. Considering the minimal difference in predictive accuracy between the two models (see Table 2), this result is unsurprising and reinforces the claim that a click on a given channel does not improve its contribution to a conversion event, relative to a regular impression of that channel.

Table 5: *Estimation results for Logit model 1, Criteo dataset.*

Channel	Coefficient (β)	Odds ratio (e^{β})
Intercept	-1.548***	
1	-0.076***	0.927
2	-0.178***	0.837
3	-0.593***	0.553
4	-0.189***	0.828
5	-0.338***	0.713
6	-0.016***	0.985
7	-0.245***	0.782
8	-0.240***	0.787
9	-0.269***	0.764
10	-0.357***	0.700

Note: *$p < 0.1$, **$p < 0.05$, ***$p < 0.01$

Table 6: *Attribution results for Criteo dataset, in percentages.*

			Markov					
Channel	FTA	LTA	Order 1	Order 2	Order 3	Order 4	LSTM	LSTM (click)
1	24.546	25.045	24.142	25.110	25.259	24.362	17.822	18.670
4	13.561	13.273	14.210	13.476	13.856	13.714	9.959	9.236
8	4.723	4.787	4.849	4.921	4.188	4.671	8.057	7.862
2	13.433	13.292	12.633	12.522	12.909	12.211	10.303	10.317
5	4.850	4.454	5.382	6.157	5.595	5.696	5.369	5.087
9	6.116	6.122	6.021	5.626	5.990	6.549	7.237	7.166
6	22.041	22.604	21.217	21.185	21.211	21.979	25.196	25.828
10	3.272	3.176	3.309	2.813	2.921	3.416	5.721	5.662
3	2.230	1.981	2.680	2.655	2.238	2.066	2.103	2.200
7	5.228	5.266	5.557	5.535	5.833	5.337	8.232	7.972

6 Conclusion

In this study, we introduced an explainable multi-touch attribution model based on a recurrent neural network that was capable of handling long-term dependencies, such as spillover and carryover effects, between channels in a customer journey. Using a long short-term memory framework, we were able to model the sequential nature of customer journeys and with the help of an attention mechanism, our model was able to evaluate the overall conversion contribution of each channel in the dataset. We further propose a second LSTM model that incorporates the number of clicks on each channel in the customer journey, to analyse the additional influence of a click on a given channel over an impression.

In the absence of a ground truth to measure the results of our attribution framework against, predictive accuracy offered a suitable means of model comparison. Along those lines, two commonly used heuristics, first-touch and last-touch attribution, and two logistic regression models were selected as baselines, in addition to the Markovian framework proposed by Anderl et al. (2016). Note that the logistic regressions illustrate the strength of a channel as a predictor of a conversion rather than its contribution to the conversion, making them more appropriate as a benchmark for a model's predictive accuracy rather than an attribution model. Applying our analysis to two publicly available clickstream datasets, we show that both our LSTM models achieve a higher in-sample and out-of-sample predictive accuracy than nearly all of the other models, asserting the validity of their attribution results. The Markov models are among the poorest performing, a result that could be attributed to the class imbalance between conversions and non-conversions within both datasets. Although our LSTM attributions scores differ from the attribution scores calculated by the baseline and Markov models (which themselves show little variation), the deviation is fairly small. The LSTM models also have a smoothing effect on the amplitude of channel attributions, ensuring a more even distribution of conversion credits between channels relative to the baseline and Markov models. With regards to the LSTM framework incorporating click information, we find no substantial improvement in the predictive accuracy over the base LSTM model, nor do we find a significant difference in their attribution scores. This leads us to the conclusion that there is little to no difference between the impact of a channel click or impression on a conversion.

Our research has several limitations that open avenues for future research. Chief among them is the availability of real world clickstream data. Our analysis is applied to only two datasets, producing company-specific results that may suffer from a lack of generalisability. Furthermore, of the two datasets, only Kaggle specifies the exact type of channel. For future studies, multiple datasets, ideally sharing several online channels, could be considered. Doing so would allow for calculation of generalisable attributions for individual channels in a multi-touch setting. The average customer journey length of the Kaggle dataset also implies a large proportion of journeys comprised of one or two touchpoints. Shorter journeys render complex attribution techniques like our LSTM framework redundant as single-touch heuristics prove adequate. Separately, although our LSTM model is capable of processing long term spillover and carryover effects in a more computationally efficient manner than Markov chains for example, its current framework is unable to explain or visualise these channel relationships. Additional research is necessary to develop neural network based solutions to mapping out channel interplay effects. Other improvements to our models include incorporating unique user information or using alternative attribution scores (e.g. removal effect, Shapley values). Our models derive attribution scores based on a channel's contribution to a binary conversion decision. However, some conversions could have a higher value than others, for example, one purchase conversion might be worth US$10 and another might be worth US$100. Attribution models that reflect channel contribution to the overall monetary value of conversions would be extremely useful for businesses. Another prevalent issue is the absence of true causal relationship information between channel visits and conversions. Future research could make use synthetic data to establish a ground truth to compare the accuracy of attributions against.

Deploying the LSTM attribution framework that we propose has numerous managerial implications. To begin with, marketers could enhance the effectiveness and streamline the budget of their marketing strategies by substituting financial resources away from channels with a low attribution scores to those with stronger contributions. The ability to calculate conversion probabilities given a customer journey enables marketers to discern the most effective or persuasive channel to present to the user next, thereby facilitating the implementation of real-time bidding strategies and further improving marketing cost efficiency. By modifying the definition of the binary conversing variable, marketers can also track the real-time performance of individual channels with regards to different marketing goals such as brand awareness or sales.

References

Adgate, B. (2021, 12). Agencies Agree; 2021 Was A Record Year For Ad Spending, With More Growth Expected In 2022. *Forbes*. Retrieved from `https://www.forbes.com/sites/bradadgate/2021/12/08/agencies-agree-2021-was-a-record-year-for-ad-spending-with-more-growth-expected-in-2022/`

Altomare, D., Loris, D., & Altomare, M. D. (2016). *Package 'channelattribution'.* Opgehaald van Cran. r-project. org: https://cran. rproject. org/web/packages . . . .

Anderl, E., Becker, I., Von Wangenheim, F., & Schumann, J. H. (2016). Mapping the customer journey: Lessons learned from graph-based online attribution modeling. *International Journal of Research in Marketing*, *33*(3), 457–474.

Arava, S. K., Dong, C., Yan, Z., Pani, A., et al. (2018). Deep neural net with attention for multi-channel multi-touch attribution. *arXiv preprint arXiv:1809.02230*.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, *30*(7), 1145–1159.

Breuer, R., Brettel, M., & Engelen, A. (2011). Incorporating long-term effects in determining the effectiveness of different types of online advertising. *Marketing Letters*, *22*(4), 327–340.

Bucklin, R. E., & Sismeiro, C. (2009). Click here for internet insight: Advances in clickstream data analysis in marketing. *Journal of Interactive marketing*, *23*(1), 35–48.

Chierichetti, F., Kumar, R., Raghavan, P., & Sarlos, T. (2012). Are web users really markovian? In *Proceedings of the 21st international conference on world wide web* (pp. 609–618).

Commercial Web Services. (2011, 04). *Clicks vs. Impressions?* Retrieved from `https://www.commercialwebservices.com/blog/2011/04/20/clicks-vs-impressions/`

Dalessandro, B., Perlich, C., Stitelman, O., & Provost, F. (2012). Causally motivated attribution for online advertising. In *Proceedings of the sixth international workshop on data mining for online advertising and internet economy* (pp. 1–9).

Danaher, P. J., & Van Heerde, H. J. (2018). Delusion in attribution: Caveats in using attribution for multimedia budget allocation. *Journal of Marketing Research*, *55*(5), 667–685.

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.

Diemert Eustache, Meynet Julien, Galland, P., & Lefortier, D. (2017). Attribution modeling increases efficiency of bidding in display advertising. In *Proceedings of the adkdd and targetad workshop, kdd, halifax, ns, canada, august, 14, 2017* (p. To appear). ACM.

Du, R., Zhong, Y., Nair, H., Cui, B., & Shou, R. (2019). Causally driven incremental multi touch attribution using a recurrent neural network. *arXiv preprint arXiv:1902.00215*.

Ghose, A., & Todri-Adamopoulos, V. (2016). Toward a digital attribution model. *MIS quarterly*, *40*(4), 889–910.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT press.

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), 107–116.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Ilfeld, J. S., & Winer, R. S. (2002). Generating website traffic. *Journal of Advertising Research*, *42*(5), 49–61.

Karpathy, A. (2015). The unreasonable effectiveness of recurrent neural networks. *Andrej Karpathy blog*, *21*, 23.

Kireyev, P., Pauwels, K., & Gupta, S. (2016). Do display ads influence search? attribution and dynamics in online advertising. *International Journal of Research in Marketing*, *33*(3), 475–490.

Li, H., & Kannan, P. (2014). Attributing conversions in a multichannel online marketing environment: An empirical model and a field experiment. *Journal of Marketing Research*, *51*(1), 40–56.

Li, H. A., & Kannan, P. (2013). Attribution modeling: understanding the influence of channels in the online purchase funnel. *Marketing Science Institute Working Paper Series*(12).

Lodish, L. M. (2001). Building marketing models that make money. *Interfaces*, *31*(3_supplement), S45–S55.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

McCarthy, K. (2021, 07). *Digital Marketing Channels: The 7 Most Essential.* Retrieved from `https://act-on.com/blog/digital-marketing-7-essential-channels/`

Olah, C. (2015, 08). *Understanding LSTM Networks – colah's blog.* Retrieved from `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`

Ren, K., Fang, Y., Zhang, W., Liu, S., Li, J., Zhang, Y., ... Wang, J. (2018). Learning multi-touch conversion attribution with dual-attention mechanisms for online advertising. In *Proceedings of the 27th acm international conference on information and knowledge management* (pp. 1433–1442).

Shao, X., & Li, L. (2011). Data-driven multi-touch attribution models. In *Proceedings of the 17th acm sigkdd international conference on knowledge discovery and data mining* (pp. 258–264).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wang, J., Zhang, W., Yuan, S., et al. (2017). Display advertising with real-time bidding (rtb) and behavioural targeting. *Foundations and Trends® in Information Retrieval*, *11*(4-5), 297–435.

Xu, L., Duan, J. A., & Whinston, A. (2014). Path to purchase: A mutually exciting point process model for online advertising and conversion. *Management Science*, *60*(6), 1392–1412.

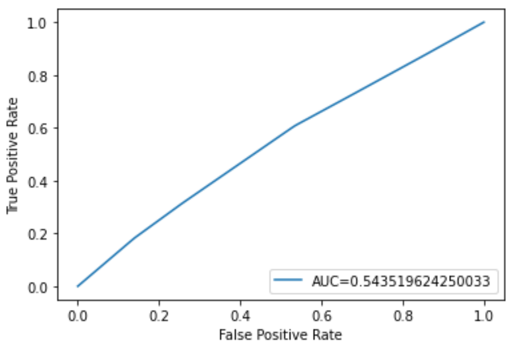
Yang, D., Dyer, K., & Wang, S. (2020). Interpretable deep learning model for online multi-touch attribution. *arXiv preprint arXiv:2004.00384*.

Zhang, Y., Wei, Y., & Ren, J. (2014). Multi-touch attribution in online advertising with survival theory. In *2014 ieee international conference on data mining* (pp. 687–696).

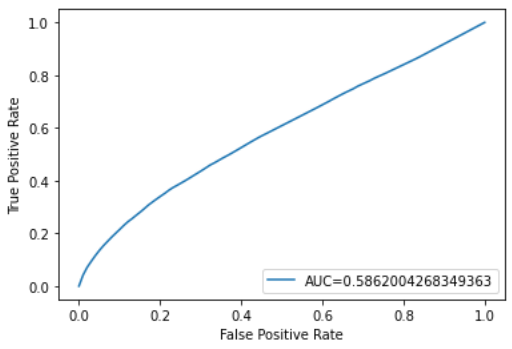
Appendix A

Figure 4: *AUC, ROC curves for Kaggle dataset.*

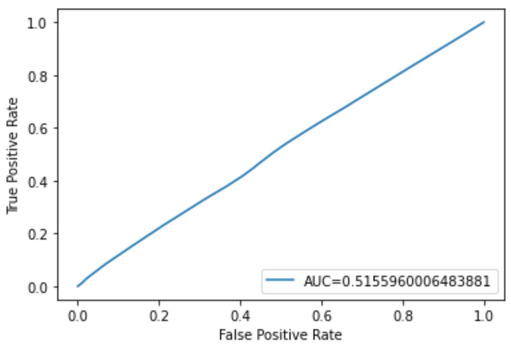
(a) *First-touch*



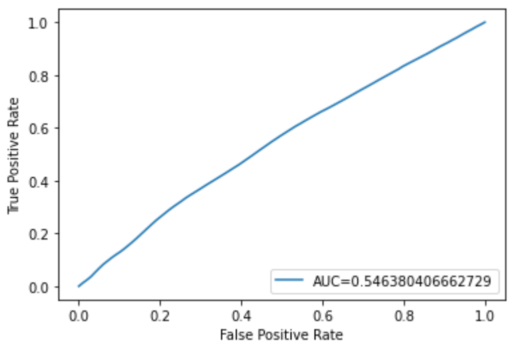
(b) *Logit 1*



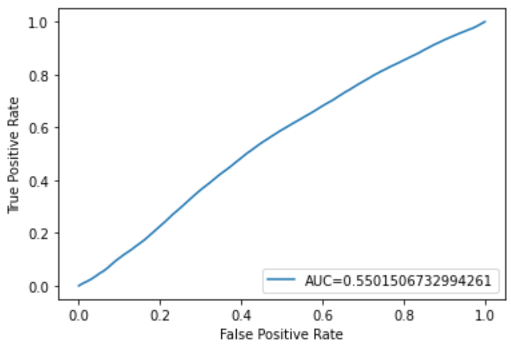
(c) *Markov order 1*



(d) *Markov order 2*



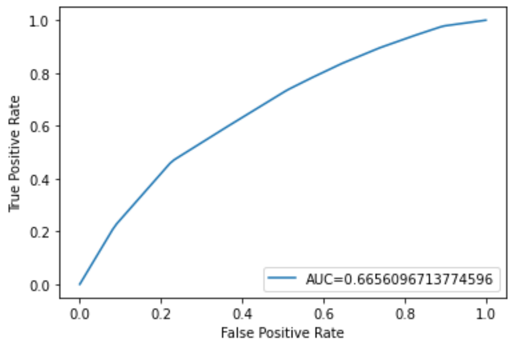
(e) *Markov order 3*



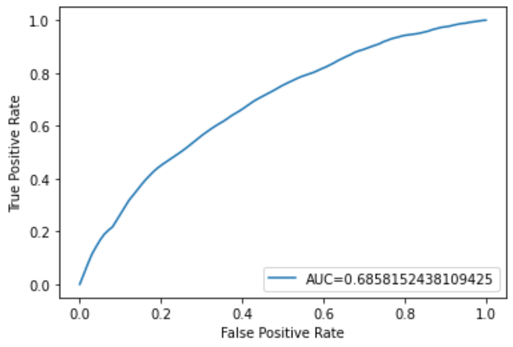
Appendix B

Figure 5: *AUC, ROC curves for Criteo dataset.*

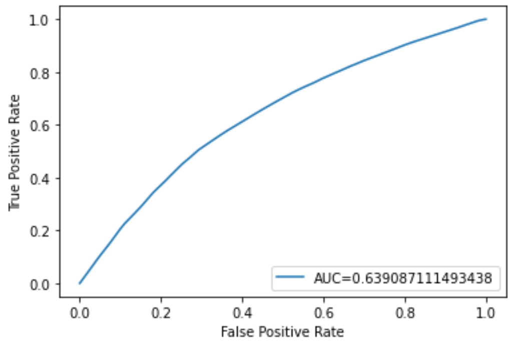
(a) *First-touch*



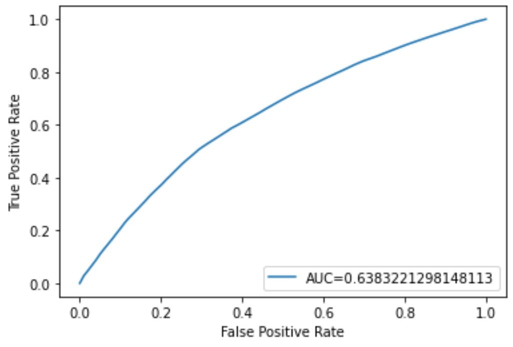
(b) *Logit 1*



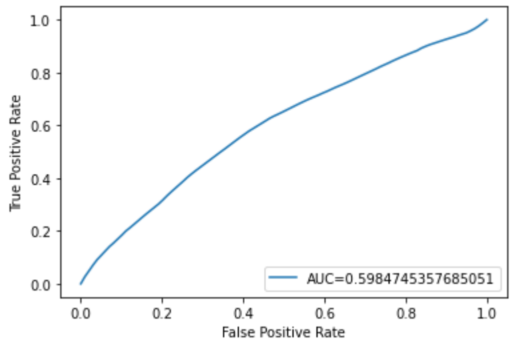
(c) *Markov order 1*



(d) *Markov order 2*



(e) *Markov order 4*


True Positive Rate
False Positive Rate
AUC=0.5984745357685051