Predicting Customer Purchase Intent in E-commerce using Clickstream Data

SI671 Final Project Report

Michelle Cheng
M.S. in Information Student, School of Information
University of Michigan, Ann Arbor
michengz@umich.edu

Abstract

Currently, e-commerce platforms utilize various web tracking tools to obtain clickstream data for analyzing their users' behaviors. While most companies use these types of data to uncover overall trends, clickstream data also be helpful when it comes to predicting future actions. In this project, we aimed to explore methods on processing clickstream data to predict customer purchase intent. Both sequential and feature-based models such as Markov Chains, LSTM, and SVM were implemented, and were trained on different levels of clickstream data at multiple sequence lengths. While the feature-based SVM models outperformed the sequence models in terms of F1 score, the results from AUC showed that LSTM models are likely to outperform feature-based models when longer sequences are provided for prediction.

1. Introduction

With the prevalence of web tracking tools and services such as Google Analytics, e-commerce platforms in recent years are able to keep records of various types of behavioral data other than traditional types of information such as user demographics, purchase history, etc. One type of user behavioral data often collected nowadays are click log data, also known as clickstream data, which keeps track of users' clicks through their journey on a website. These include information regarding when a user visited a website and what pages/features they had engaged during a session. Companies can thus utilize these types of clickstream data to further inform about user behaviors and drive business decisions.

While most companies use clickstream data to aggregate data across multiple users to uncover overall trends, clickstream data can also be helpful when it comes to predicting future actions of specific customers given its sequential nature. In this project, we try to explore different methods on processing clickstream data to predict whether a given click path would lead to a successful purchase. By predicting purchases and drop-offs at an early stage of a user session, it can allow companies to target the right campaigns/promotions more accurately during a user's journey. These predictions can also provide companies insights on how user experiences can be improved on certain pages of their websites to keep users engaged.

A few research questions that were explored throughout the project include:

1. Can Sequence Modeling on clickstream data outperform common feature-based classification models?

2. Are we able to predict purchase intents with limited volume of click- stream data? What is the minimum length of clickstream sequence required to make acceptable predictions?

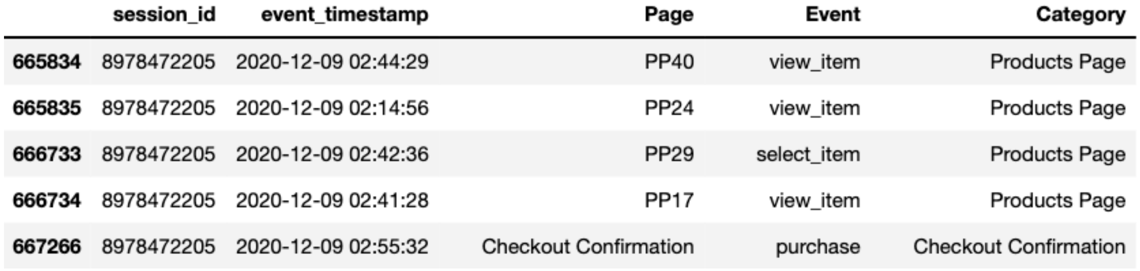
3. How does the level of interaction (e.g. page-level, event-level) affect prediction performances? Does categorization of pageviews help with improving performance?

2. Data

2.1 Data Collection

Since customer-related data is often private to companies in order to protect user information, we obtained our data from the Google Analytics 4 Sample Dataset [1] provided by BigQuery Public Dataset. It includes clickstream data of the Google Online Merchandise Store tracked by Google Analytics, which stores various attributes related to user behavior, acquisition, and transaction. For this project, we wrote SQL queries on BigQuery to export data spanning from November 1, 2020 to December 31, 2020 that are within our project scope. The final dataset includes 2,946,000 rows of click logs with 87,840 unique user sessions, where each session contains the path that a user took on the website.

In this dataset, user interaction paths can be represented as three different levels – Page, Event, and Category (Fig.1). Click paths on the page level describes the titles of the pages the user interacted with (e.g. Home, Product Page 1, Product Page 2, etc.). On the other hand, click paths on the event level describes the interaction that they had with the webpages (e.g. view item, purchase, etc.). Since the website contains a total of 43 unique product pages, we grouped similar pages into higher level categories to see whether categorization of sequences would affect prediction performances.



	session_id	event_timestamp	Page	Event	Category
665834	8978472205	2020-12-09 02:44:29	PP40	view_item	Products Page
665835	8978472205	2020-12-09 02:14:56	PP24	view_item	Products Page
666733	8978472205	2020-12-09 02:42:36	PP29	select_item	Products Page
666734	8978472205	2020-12-09 02:41:28	PP17	view_item	Products Page
667266	8978472205	2020-12-09 02:55:32	Checkout Confirmation	purchase	Checkout Confirmation

Fig.1: Samples of click logs from a single user session

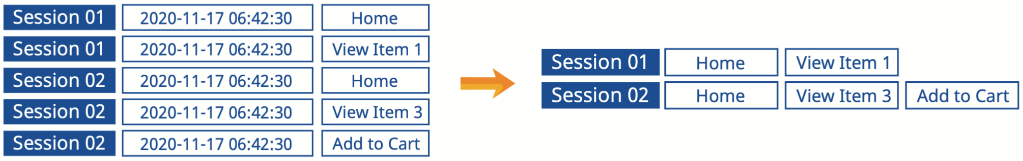
2.2 Data Preprocessing

Several preprocessing techniques were performed on the dataset prior to building our prediction models:

- **Labeling**

  Since the main goal of the project is to predict whether a given click path would lead to a successful purchase, this project is treated as a binary classification problem. Therefore, sessions in the dataset are labeled as “Purchase” (1) if a purchase event was present in its click journey, otherwise, labeled as “Non-Purchase” (0).
- **Sessionize**

  Rows of individual click logs with at different timestamps are transformed to event sequences that are grouped by sessions in preparation for modeling sequential data. (Fig. 2)This sessionization process was done by using a preprocessing function from a python package called MarkovClick [2], which deals with server log issues such as session timeouts when grouping clickstream data.



Session 01	2020-11-17 06:42:30	Home
Session 01	2020-11-17 06:42:30	View Item 1
Session 02	2020-11-17 06:42:30	Home
Session 02	2020-11-17 06:42:30	View Item 3
Session 02	2020-11-17 06:42:30	Add to Cart

Session 01	Home	View Item 1	
Session 02	Home	View Item 3	Add to Cart

Fig.2: Simple visualization of transforming click log data to sequential format

- **Undersampling**

  Out of the 87,840 unique sessions in the dataset, 95.8% of the sessions are with no purchases, while 4.2% of the sessions were successful purchases. Given that our dataset is highly imbalanced, we downsampled our dataset to a 2:1 ratio of non-purchases data and purchase data so that our models will not falsely converge by classifying everything as the majority class.

- **Preventing Data Leakage**

  Some events/pages within click paths such as “Checkout Confirmation” or “Purchase” may be strongly related to purchasing outcomes. Therefore, in order to prevent these events from “leaking” our labels, we had to make sure to exclude them so that our models could serve our purpose of predicting purchase intent based on normal click paths.

3. Methods

For this project, we implemented 3 different methods to predict purchase intent, two of them are models trained on sequential data and one of them is trained on common features in the field of machine learning.

3.1 Sequence Discrimination with Markov Chains

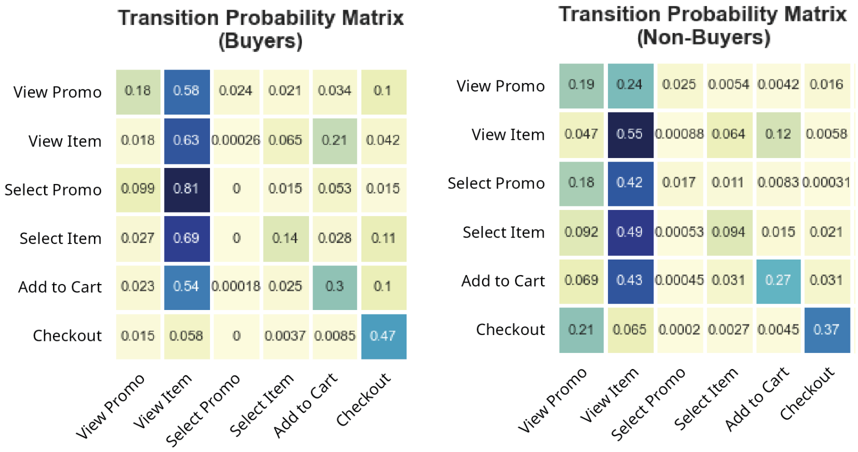
In this project, we implemented a sequence discrimination measure proposed by Durbin, R. [3], and learned from Miguéis, V. L.[4] on how it could be applied to the context of predicting customer actions through purchase sequences. They assume that sequences in each class (e.g. buyer and non-buyer) come from a specific Markov process for each class, and that we can calculate the likelihood of an observing sequence stemming from either of the classes. To calculate the likelihood of a sequence, we first create Markov transition probability matrices for each class (Fig.3) , which represent the probabilities of transitioning between states of pages/events. The log odds of transitioning from one state to the other can thus be calculated based on the transition matrices:

$$S(x) \;=\; log\frac{P(x\,|\,buyer)}{P(x\,|\,non{-}buyer)}$$

We then calculate the odds of an observing sequence originating from either a buyers class or non-buyers class based on transition probabilities of the two different classes:

$$\begin{aligned} S(View\ Item\ \rightarrow\ Select\ Item\ \rightarrow\ Add\ to\ Cart) \;&=\; log\,\frac{0.065}{0.064} + log\,\frac{0.028}{0.015} \\ &=\; 0.277 \end{aligned}$$

In this case, a positive log odds ratio means that the provided sequence is more likely to originate from the buyers class, while a negative value means the opposite.



Transition Probability Matrix (Buyers)

	View Promo	View Item	Select Promo	Select Item	Add to Cart	Checkout
View Promo	0.18	0.58	0.024	0.021	0.034	0.1
View Item	0.018	0.63	0.00026	0.065	0.21	0.042
Select Promo	0.099	0.81	0	0.015	0.053	0.015
Select Item	0.027	0.69	0	0.14	0.028	0.11
Add to Cart	0.023	0.54	0.00018	0.025	0.3	0.1
Checkout	0.015	0.058	0	0.0037	0.0085	0.47

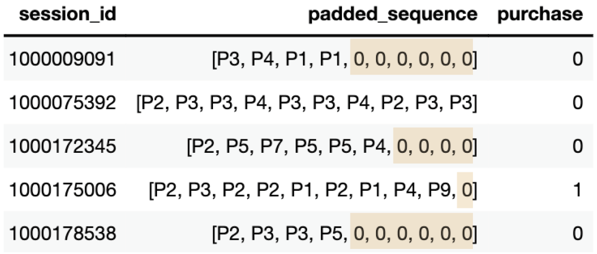
Transition Probability Matrix (Non-Buyers)

	View Promo	View Item	Select Promo	Select Item	Add to Cart	Checkout
View Promo	0.19	0.24	0.025	0.0054	0.0042	0.016
View Item	0.047	0.55	0.00088	0.064	0.12	0.0058
Select Promo	0.18	0.42	0.017	0.011	0.0083	0.00031
Select Item	0.092	0.49	0.00053	0.094	0.015	0.021
Add to Cart	0.069	0.43	0.00045	0.031	0.27	0.031
Checkout	0.21	0.065	0.0002	0.0027	0.0045	0.37

Fig.3: Markov Transition Probability Matrices

3.2 Sequence Classification with LSTM

A common method used in the field of Natural Language Processing (NLP) to classify text sequences is the Long Short Term Memory (LSTM) network, a type of Recurrent Neural Network (RNN) that has the advantage of learning longer patterns and forgets and remembers data selectively [6]. In this project, similar to what is usually done in NLP, we treated each clickstream sequence as sentences to feed our data into the LSTM [7]. Since click paths for each session in our dataset come in different lengths, each sequence is transformed into a "padded sequence" before passing in as training inputs (Fig.4).

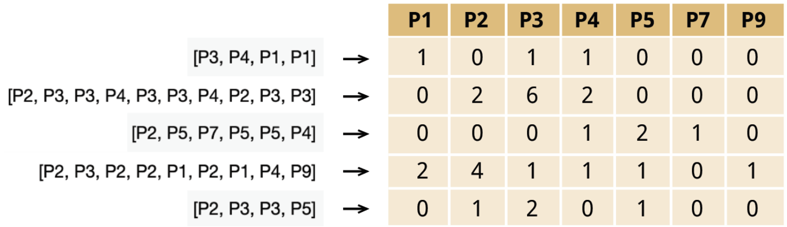


session_id	padded_sequence	purchase
1000009091	[P3, P4, P1, P1, 0, 0, 0, 0, 0, 0]	0
1000075392	[P2, P3, P3, P4, P3, P3, P4, P2, P3, P3]	0
1000172345	[P2, P5, P7, P5, P5, P4, 0, 0, 0, 0]	0
1000175006	[P2, P3, P2, P2, P1, P2, P1, P4, P9, 0]	1
1000178538	[P2, P3, P3, P5, 0, 0, 0, 0, 0, 0]	0

Fig.4: Padded sequence of a sample user session

3.3 Feature-based Standard Classification

The non-sequential features extracted for training standard classification are the amount of clicks of each page that are present in a sequence. Similar to storing term frequencies using Bag of Words (BoW) in NLP [8], we transform our data into feature vectors where each event represents a single feature (Fig.5). Standard classification models such as Support Vector Machines (SVM), Logistic Regression, and Random Forest were implemented to train on these common features. As a result, the SVM model yielded the best performance among the classifiers, and was selected as our feature-based classification model to compete with the other sequence models.



		P1	P2	P3	P4	P5	P7	P9
[P3, P4, P1, P1]	→	1	0	1	1	0	0	0
[P2, P3, P3, P4, P3, P3, P4, P2, P3, P3]	→	0	2	6	2	0	0	0
[P2, P5, P7, P5, P5, P4]	→	0	0	0	1	2	1	0
[P2, P3, P2, P2, P1, P2, P1, P4, P9]	→	2	4	1	1	1	0	1
[P2, P3, P3, P5]	→	0	1	2	0	1	0	0

Fig.5: A visualization of feature vectors representing click frequency

4. Experiments

4.1 Model Comparison

For experimentation, we compare our models from the three methods proposed above. Each model was trained and evaluated on different clickstream length cutoffs, ranging from 2 to 30 clicks per session as input data. For each sequence length, the models were also trained by multiple interaction levels – click paths on the page level, event level, and category level. As such, we compared 9 models per click-length.

We also performed hyperparameter tuning for models from each of the three methods. The Markov transition order is determined based on two order estimate criterias, Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC) [5], which are common model comparison methods for using likelihood ratios. The number of orders (k) resulting in lowest AIC and BIC values are suggested to be better models. An order of k=1 for our Markov Chain was preferred after testing with models with different orders among the two classes (Table 1). This means that, for each state in the model, it would only depend on its preceding state, in which we call it a First-Order Markov Chain Model.

For the LSTM models, we tested different batch sizes (32, 64, and 128), number of neurons (50, 100, and 300), and number of training epochs (~30). The hyperparameters selected eventually were based on validation loss (Fig.6) and evaluation scores on the validation set.

Class	Order	AIC	BIC
Purchase (Buyer)	0	358741.00	358741.00
	1	173331.35	173694.09
	2	173411.35	174136.82
Non-purchase (Non-buyer)	0	1145267.90	1145267.90
	1	505611.59	506135.84
	2	505713.59	506762.09

Table 1: AIC and BIC values by Markov orders

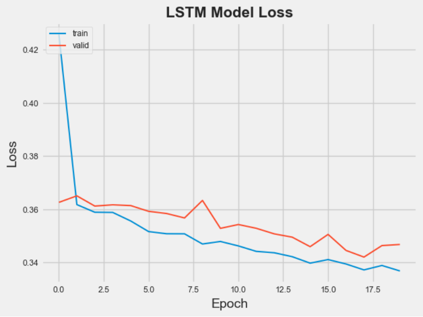


Fig.6: LSTM model loss over number of epochs

4.2 Evaluation Metrics

Since our data remained slightly imbalanced after undersampling, an important metric that we selected for evaluation is the F1 score, which is known for dealing with imbalanced datasets given its nature of finding the balance between precision and recall [9]. Another metric used for evaluating our models is the Area Under Curve (AUC). A higher AUC means that our model can achieve a higher true positive rate with a correspondingly low false positive rate. Since we are not completely certain of the probability boundaries for our models, the AUC curve allows us to compare performances at different classification thresholds.

5. Results & Discussion

As a result, when looking at F1 scores (Fig.7), the feature-based SVM model had the highest performance throughout all sequence length cutoffs, while the LSTM sequential model performed mostly second and Markov Model the last. Both the SVM and LSTM model improved significantly as we increase the cutoff length from 2 to 5, and could reach F1 scores between 0.85 to 0.90 as longer clickstream lengths are tested. The highest F1 score that the SVM model can reach at the 5-click cutoff is 0.86 (Table 2). The SVM model is also much more stable compared to the LSTM models at different cutoff lengths, which from Fig.7 we can tell that the LSTM models showed relatively more fluctuation in performance. Reasons on why the F1 scores for feature-based models outperformed the sequence models are yet to be investigated. However, a few possible factors are considered:

- There may still be room for improvement for tuning our LSTM models given that there are many hyperparameters to be tested especially in neural network models.

- A lot of the clickstreams within the dataset had only 2-3 events, which may potentially affect the training on sequence models.

- Another possibility is that click events within paths may not strongly depend on each other in the context of e-commerce clickstream sequences. As we have tested for the order estimate of the Markov Model, we found that the order that yielded the lowest AIC and BIC values wes k=1, meaning that the model performs the best when one state only depends on its preceding state.

Although the feature-based models outperformed all sequence models when evaluated using F1 score, some LSTM models performed better than the feature-based SVM when looking at the AUC outcomes (Fig.8). After the 10-click cutoff, the LSTM sequence models can reach up to AUC scores of 0.93 (Table 2), meaning that they are likely to outperform feature-based models if enough data are provided for prediction. However, the feature-based SVM models still perform better at the 5-click cutoff. This may be because that a short clickstream with only 5 events may not be enough for LSTM models to capture sequential properties at such limited length. Longer click paths may provide more information for the LSTM models to compute probabilities for each label based on sequential patterns.

In terms of interaction level, the models trained with click paths on the Event level resulted in the highest F1 scores throughout all cutoff lengths of the SVM method, meaning that the event-level features may be more valuable when it comes to predicting purchases. As for AUC, the LSTM model trained on page-level clickstreams yielded the best performance.

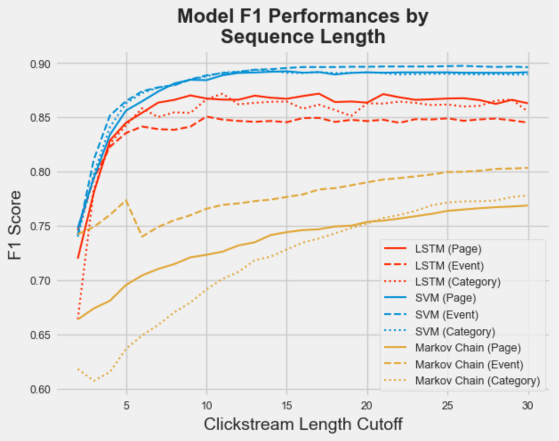


Fig.7: Model F1 Score by sequence length

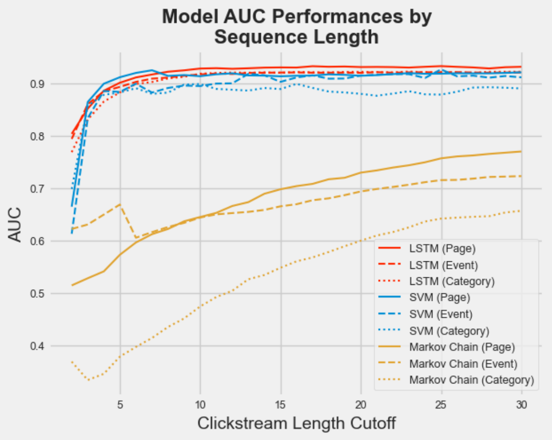


Fig.8: Model AUC by sequence length

Model	Sequence Length Cutoff	5		10		15		20		25		30	
	Interaction Level	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
Markov Chain	Page	0.695	0.573	0.723	0.645	0.744	0.698	0.753	0.729	0.764	0.757	0.769	0.770
	Event	0.773	0.669	0.766	0.644	0.776	0.665	0.790	0.693	0.799	0.715	0.803	0.723
	Category	0.637	0.378	0.691	0.474	0.728	0.548	0.752	0.599	0.771	0.642	0.778	0.657
LSTM	Page	0.845	0.901	0.867	0.928	0.867	0.930	0.863	0.931	0.867	0.933	0.863	0.931
	Event	0.835	0.893	0.851	0.917	0.845	0.920	0.846	0.920	0.849	0.921	0.845	0.921
	Category	0.842	0.882	0.866	0.918	0.864	0.920	0.863	0.922	0.862	0.921	0.855	0.922
SVM	Page	0.856	0.911	0.884	0.913	0.892	0.913	0.891	0.914	0.891	0.918	0.891	0.921
	Event	0.865	0.883	0.888	0.895	0.895	0.903	0.896	0.915	0.897	0.926	0.896	0.911
	Category	0.863	0.882	0.888	0.898	0.891	0.889	0.891	0.880	0.890	0.878	0.889	0.890

Table 2 : F1 Score and AUC by sequence length and page-levels

6. Conclusion

In this project, we aimed to explore and compare different methods on processing clickstream data to predict user purchase intent early on a user session. Models were compared at different clickstream lengths to learn how long a click path should be in order to make acceptable prediction outcomes. The models were trained with different levels of clickstream data as well to see if the types of click paths would affect performance. Overall, the feature-based standard classification SVM models outperformed the sequential models in terms of F1 score, and is also more stable compared to models trained with sequential data. The LSTM models yielded the best AUC when longer lengths of clickstreams were provided for prediction, which we hypothesize that shorter clickstreams may not be sufficient for LSTM models to capture sequential patterns. Given that most models reach F1 scores of 85% at the clickstream length of 4-5, we conclude that there should be at least 4-5 click events in observing sequences to ensure its credibility of making the right predictions. As for click path level, there were no significant patterns of whether a specific interaction level works the best for predicting purchases. However, we can conclude from the results that the LSTM model almost always obtained higher scores when training on page-level sequences, while the feature-based model worked best when training with event-level data.

7. Future Work

In this section, we propose several future steps for improvement and in-depth research. First, we propose that ways on how the LSTM models can be improved could be further explored. Hyperparameters such as learning rate, dropouts, batch size, number of neurons, etc. are all potential factors that can greatly affect the performance of their predictions. In terms of unused attributes from our dataset, time-based features such as the amount of time spent on each page can be further explored and incorporated to the feature-based classification model. To help with bringing more business insights, we can also consider performing customer segmentation using clustering methods to capture more detailed user intentions, such as whether a user is only casually browsing or predict if they would eventually abandon their cart.

References

[1] BigQuery sample dataset for Google Analytics 4 ecommerce web implementation https://developers.google.com/analytics/bigquery/web-ecommerce-demo-dataset

[2] MarkovClick. https://markovclick.readthedocs.io/en/latest/

[3] Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G. (1998). Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge university press.

[4] Miguéis, V. L., Van den Poel, D., Camanho, A. S., & Falcão e Cunha, J. (2012). Predicting partial customer churn using Markov for discrimination for modeling first purchase sequences. Advances in Data Analysis and Classification, 6(4), 337-353.

[5] Tong, H. (1975). Determination of the order of a Markov chain by Akaike's information criterion. Journal of applied probability, 12(3), 488-497.

[6] Rao, A., & Spasojevic, N. (2016). Actionable and political text classification using word embeddings and LSTM. arXiv preprint arXiv:1607.02501.

[7] Clickstream based user intent prediction with CNNs http://digital-thinking.de/deep-learning-clickstream-based-user-intent-prediction-with-anns/

[8] Venkata Raju, K., & Sridhar, M. (2020). Based sentiment prediction of rating using natural language processing sentence-level sentiment analysis with bag-of-words approach. In First International Conference on Sustainable Technologies for Computational Intelligence (pp. 807-821). Springer, Singapore.

[9] Forman, G., & Scholz, M. (2010). Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. Acm Sigkdd Explorations Newsletter, 12(1), 49-57.