Impact of Promotional Codes During Checkout on Fingerhut Customers

Peppermint Mocha: Karen Nguyen[1,2,†], Alan Wang[1,2,†], Ananda Gowda[1,2,†], and Deana Moghaddas[1,2,†]

[1]Department of Mathematics, UCLA, Los Angeles, CA 90095, USA.

[2]Department of Statistics, UCLA, Los Angeles, CA 90095, USA.

[†]These authors contributed equally to this work.

March 24, 2023

Executive Summary

Fingerhut is looking to incentivize consumers to make a purchase, but in a strategic manner to effectively maximize revenue. To address this problem, one of the questions Bluestem Brands raises is how they can go about predicting a user's behavior and activity to help inform whether it is worth it to incur marketing spend on this particular user. They want to uncover which users are more likely to convert based on their website activity and accessible features; these customers would be the population to target for marketing tactics since the tactics have a higher probability of succeeding.

For our case, our group narrows the problem raised by Bluestem Brands by investigating the effect of deploying promotional codes during checkout on conversion and purchase rates, since we have clear data on the usage of this marketing technique. We can see in Fig. 1 that promotional code failure leads to lower purchase rates compared to applying a successful code, so we want to explore the impact of applying promotional codes on purchase probabilities.

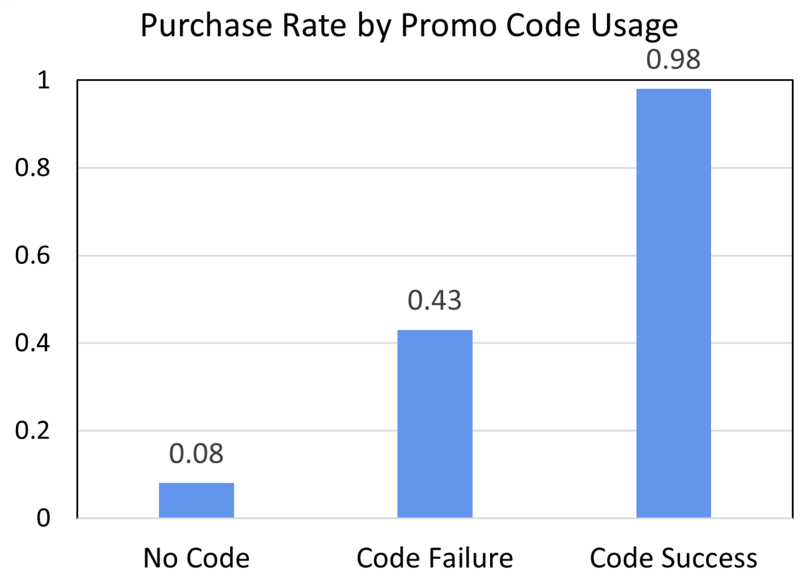

Figure 1: Difference in purchase rates based on promo code utilization.

We analyze individual user and session behavior and characteristics by looking at features such as their lifetime order totals, behavior score, visit duration, event list, etc. Using this information, we examine the probability that an individual customer would make a purchase with and without a promotional code in each visit session; the difference in these two probabilities is the metric we are interested in. We want to investigate whether there will be an increase in the probability of purchasing when offered a promotional code and if there is, how substantial the increase is. We hope that this will benefit Bluestem Brands by informing them which types of users they should ultimately target with promotional codes to best improve purchase rate and in turn, increase revenue.

To investigate our problem, we first select a variety of features that we thought would best represent and distinguish types of customers; these include a number of categorical and numeric features that we found to be statistically significant from preliminary testing. Following this, we build an Artificial Neural Network in conjunction with a Recurrent Neural Network and input our selected features into this model to train it to predict the probability of purchasing for each visit session. With our model, we simulate the impact of promotional codes by predicting purchase probabilities with and without code usage. We examine the gain in probability for each individual visit session and group visits into four bins based on their gain in percentages: 0-25%, 25-50%, 50-75%, and 75-100%. Lastly, we summarize the features of visits in each bin and analyze what types of users are distinctly represented to make conclusions on which users Bluestem Brands should mainly target promotional codes to.

In the end, our model signifies that Bluestem Brands should look to target promotional codes at new users, indicated by features such as low lifetime order totals, high order recencies, etc., in order to maximize the probability of conversion. Conversely, Bluestem Brands should stray away from offering promotional codes to returning users with high behavior scores, high lifetime order totals, etc., because promotional codes have the least significant effect on these users.

Our analysis aims to dive deeper into one of the various marketing tactics of Bluestem Brands and discover how the company should optimally employ it. This report contains a summary of our group's methodologies and procedures, the results and findings, as well as plans for future work and development.

Contents

1 Introduction **5**

2 Problem Statement **5**

3 Methodology **6**

3.1 Data 6

3.2 Assumptions 7

3.3 Data Cleaning 7

3.4 Feature Engineering 7

3.5 Statistical Testing 8

3.6 Modeling 10

3.6.1 Artificial Neural Network + Recurrent Neural Network 10

3.6.2 Influence of Promotional Codes 12

3.7 Results 13

4 Conclusions **15**

4.1 Summary of Results 15

4.2 Limitations 15

4.3 Next Steps 16

References **18**

1 Introduction

Fingerhut is an online shopping company that aims to build or rebuild consumer credit by providing products that can be financed through payment plans, allowing for the subsequent staggering of credit card bill payments. With millions of customers visiting their website, Fingerhut is able to collect vast amounts of data on customer behavior, preferences, and purchasing patterns. This data can be analyzed to create more effective and personalized marketing strategies in order to drive growth in sales.

In order to optimize their marketing strategies to suit the profile of a specific user, companies will oftentimes use data to analyze each individual's behavior and characteristics, and then aggregate users into separate groups based on similarities to create a more simplified view of customers. Companies can then target marketing according to the tastes of each group and even test the effectiveness of marketing campaigns to decide which type of user to target with certain campaigns.

2 Problem Statement

A common marketing tactic used to increase a user's likelihood of purchasing is by providing promotional codes. However, some types of users leap at the opportunity to use a promotional code, while others remain indifferent. According to a survey conducted by RetailMeNot, 80% of shoppers made their first purchase with a brand because of an offer or discount [1]. Characteristics such as whether a customer is a first-time buyer or information like a user's website activity can inform us on how the introduction of a promotional code may or may not drive users to purchase. This analysis can also aid in the individualizing of promotional codes according to the profile of a user, and prevent unnecessary marketing expenses. Our objective is to help Fingerhut decide which types of users they should ideally invest marketing tactics on, specifically promotional codes, and which users they should approach with alternative strategies. Using a machine learning model, for each visit session v, we want to find the difference between the probability of purchasing given a promo code and the probability of purchasing without being given a promo code, i.e. $P(\text{purchase}|v, \text{promo}) - P(\text{purchase}|v, \text{no promo})$. To examine which types of users experience the greatest changes in probability and which experience the least, we group users into four bins based on this change in probability of purchasing and investigate the range of traits these users have in common.

3 Methodology

3.1 Data

The dataset made available to this project contains web log data of all user activities on the Fingerhut website for the week of December 10, 2022, through December 16, 2022. It contains 42,730,149 rows, one row for one event and 283 columns, and one column for one attribute tracked by the Adobe Analytics system.

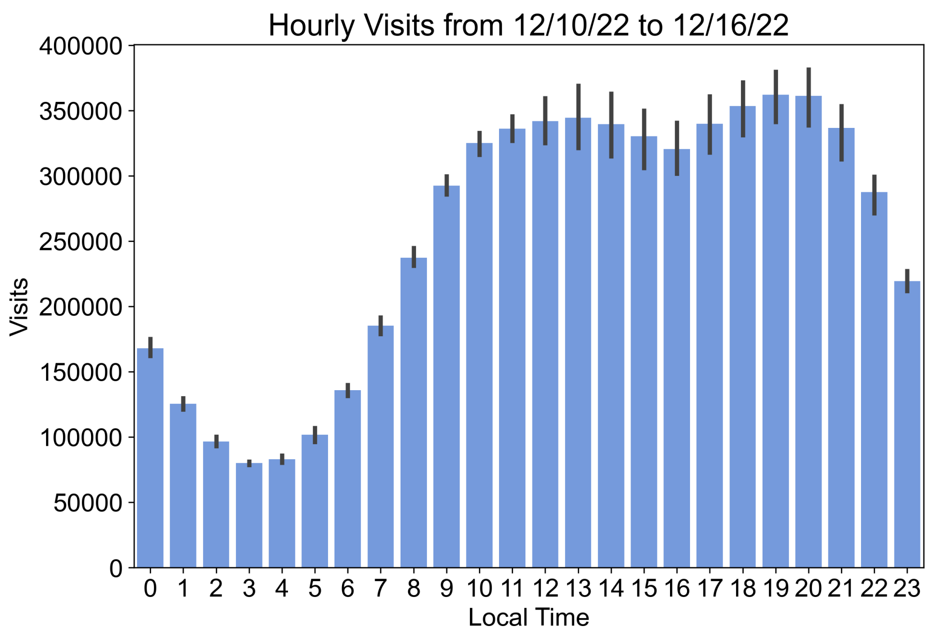

Figure 2: Average number of unique visit sessions per hour across 7 days. The bars represent the 95% confidence interval.

Grouping by visit id, the dataset contains the activity of 2,041,286 unique visit sessions. The average number of unique visit sessions per hour across the week is shown in Fig. 2.

3.2 Assumptions

To execute our analysis and ensure our results are interpretable, we make the following assumptions:

- To model how the probability of a purchase changes based on user access to a promotional code, we must assume that a user will use a new promotional code at checkout if one is provided during checkout.
- We assume the features we have chosen to include in our model are representative of overall user behavior.

3.3 Data Cleaning

Given our goal is to incentivize users to make a purchase with a promotional code, we first filter the dataset to unique sessions that reach the checkout page. In addition, we no longer need to convert customers after they make a first purchase, so we further filter the data to only keep the activities up to each session's first Checkout Confirmation (purchase) page; if a session does not end with a purchase, we keep all its activities.

The dataset after cleaning contains around 30% sessions with purchase, indicating an imbalanced dataset.

3.4 Feature Engineering

For every visit session, we compute preliminary user attributes and session attributes shown in Table 1. We select this initial subset of features based on obtained business knowledge and intuition and will perform statistical testing on the numeric and categorical variables in Section 3.5 to finalize our selection.

For the `event list` feature, we clean the `pagename` column to 33 types of pages, such as home page, account page, different product category page, error page, etc., and then we apply ordinal encoding. Note that we use 1-33 instead of 0-32 here to avoid non-learning due to zero gradient in our model training. Then, for each visit session, we extract a list of all pages visited throughout the session.

User Attributes	Session Attributes
Order Recency	Number of PDP Visits
Behavior Score	Number of Cart Visits
Lifetime Order Totals	Visit Duration
Number of Previous Visits	Device Type
	Promo Attempt
	Cart Features: Size; Price Total, Range, Std. Dev.
	Event List

Table 1: Preliminary User and Session Attributes for Modeling

In addition, the user attributes all contain `NaN` values that must be imputed in order to properly train our model:

- In the `order recency` column, `NaN` values are substituted with the maximum value present in the data, as the value represents the time since that customer made a purchase.

- For `behavior score`, we replace `NaN` values with 300, the minimum value in the column since larger behavior scores are given to more active Fingerhut users.

- `NaN` values in `lifetime order totals` are given a value of zero. Missing values in this column mean that the user has never made a purchase.

- `NaN` values in `number of previous visits` are also given a value of zero, as missing values mean the user has never visited Fingerhut.

3.5 Statistical Testing

After engineering the preliminary features, we conduct statistical testing to evaluate the relationship between each feature and the target variable: purchases vs. non-purchases. We want to discover the significance of each variable and determine whether they should be kept in our model. The two statistical tests we choose to perform are the Two-sample Z-test and the Chi-Square Test of Homogeneity; the former is for our numeric features, and the latter is for our categorical features.

Regarding the Two-sample Z-test, we investigate whether the means of the numeric feature for each group (purchases vs. non-purchases) are significantly different. If they are significantly different, this means that the feature can contribute to classifying the two groups [2]. Otherwise, the feature does not aid in explaining the distinction between the two groups and should not be considered in our predictive model.

For instance, we plot the visit duration in Fig. 3 grouped by sessions with and without purchases, and it can be observed that there is a difference in the amount of time that users spend on the site between visits that end with a purchase and visits that do not end in a purchase. The feature has a corresponding p-value of < 0.001.

After performing statistical tests on each of our numeric features mentioned in Section 3.4, we find that all of the preliminary features we engineered have significant p-values, so we keep all of them in our modeling.

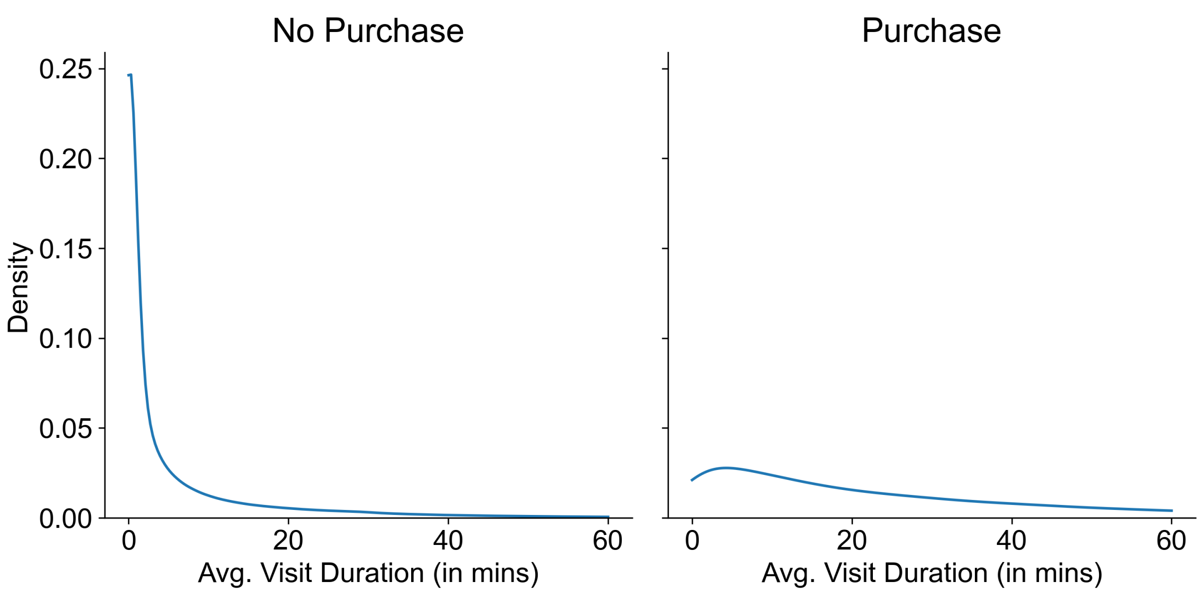

Figure 3: Distribution of time spent on Fingerhut.

Likewise, the Chi-Square Test of Homogeneity is executed for our potential categorical features. We analyze whether the purchase rates between each category of the feature are significantly different. This can help us determine if some categories of the feature are more likely to purchase than other categories, and thus will help us predict a purchase. If there are no significant differences in purchase rates between categories, then the categorical feature is inconsequential for our purposes as it gives no information about purchasing.

After we find that the p-value is significant for a feature, we continue our testing to look at pairwise comparisons, as shown in Fig. 4, in order to see which specific pairs significantly differ in purchase rate. Based on the pairwise results, we transform the categorical feature `device type`. After testing, we keep all other categorical features.

Device Type	Non-Purchase	Purchase	Purchase Rate
Desktop	239541	9063	0.170
Mobile	921389	106642	0.104
Tablet	408872	46921	0.103
Unknown	65	6	0.085

Chi-Squared Statistics	P-Value
10546	0

1) Construct contingency table and carry out initial Chi-Square Test

Pairs	P-Value
Desktop – Mobile	0
Desktop – Tablet	0
Desktop – Unknown	0.203
Mobile – Tablet	0.4936
Mobile – Unknown	0.8361
Tablet – Unknown	0.8361

2) Execute Chi-Square Test with pairwise comparisons

Categories BEFORE	Categories AFTER
• Desktop • Mobile • Tablet • Unknown	• Desktop • Other *since Desktop was the category that was mostly different from the others*

3) Transform categories based on which pairs are significantly different

Figure 4: Process flow of the statistical testing for `device type`, a categorical feature.

3.6 Modeling

3.6.1 Artificial Neural Network + Recurrent Neural Network

Instead of using traditional machine learning models to predict the probability of purchasing for each visit session, we apply a combination of an Artificial Neural Network (ANN) and a Recurrent Neural Network (RNN) for the following three reasons:

1. A neural network applies non-linear transformations to the variables, so it does not overly rely on a single variable for prediction, which is imperative for a classification task on an imbalanced dataset like ours.

2. Given that each session may result in a different number of activities, the length

of `event list` is variable. Furthermore, it is sequential data so it demonstrates a recurrent property.

Unlike traditional machine learning models that take in a single number and are unable to capture sequential relationships, RNNs have connections between nodes that can create a cycle, allowing output from some nodes to affect subsequent input into the same nodes. This is most suitable for our task.

3. Lastly, most traditional classification models only output the result but not the probability. However, for neural networks, we can apply a sigmoid transformation

$$\sigma(x) = \frac{1}{1+e^{-x}} \in (0,1)$$

to convert the output to a probability of purchasing.

Our model architecture is shown in Fig. 5.

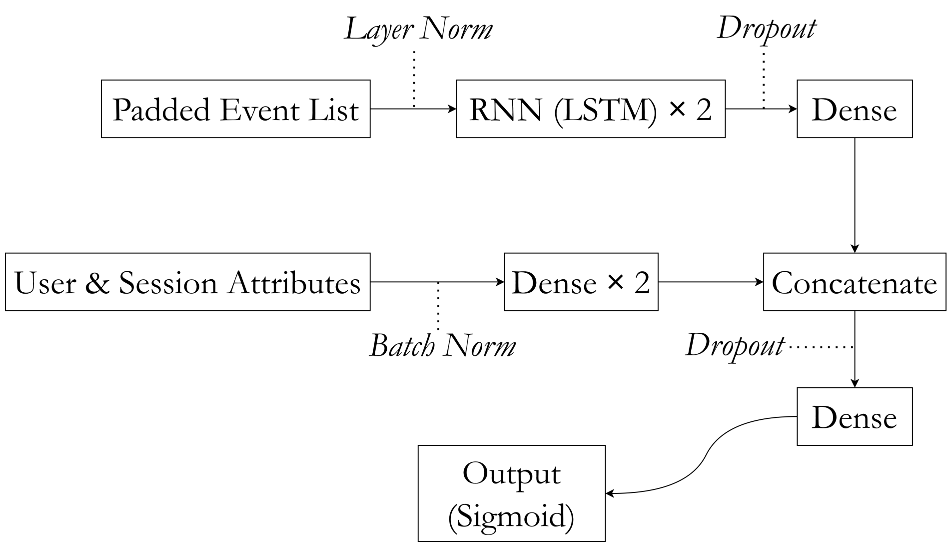

Figure 5: The architecture of our ANN + RNN model.

We first pad each `event list` to a length of 400, with 0's at the beginning, so that the most important events - the last few events before reaching the checkout page - are aligned across instances. If an event list is originally longer than 400 entries, we keep only the last 400 events. We choose 400 as 99.5% of sessions have less than 400 events.

Then, we feed the padded event lists to an RNN; in our case, we stack two layers of Long Short-Term Memory (LSTM) layers. The actual mathematical implementation of LSTM is beyond the scope of this project, but LSTM utilizes an input gate, a forget gate, and a value gate to store information on past data and relate it to the new data along the training process [3]. We apply Bidirectional LSTM so that the model can learn in both directions of the sequential event list.

For all other features, we feed them into an ANN, where we stack two Dense (Fully-Connected) layers. The output is then concatenated with the output from the LSTM. After concatenation, the model now captures features from sequential data and non-sequential data alike and can be used to output the probability of purchasing with a Sigmoid layer.

To boost the model performance, we also utilize certain regularization techniques:

- To speed up training, we apply layer normalization for sequential data and batch normalization for non-sequential data. Layer normalization works per training instance while batch normalization works across all instances; both adaptively shift the mean and variance of the data to values that the model "prefers" over time during training to accelerate training [1].
- To prevent overfitting, we add two Dropout layers. Dropout layers randomly set some input units to 0 at each step during training. When the weight corresponding to one feature is set to 0, the model is forced to learn from the remaining features. This makes sure that the model does not overly rely on any single feature and thus, prevents overfitting.

3.6.2 Influence of Promotional Codes

To determine the effects of promotional codes on purchase rate, we use the probabilities returned by the model and leverage the `promo_attempt` column. As shown in Fig. 6, we input each session (i.e. row) into the model twice, once with `promo_attempt` as 1 and once as 0. This simulates the same session with and without a promotional code and returns two

[1] Like many other techniques in neural network and deep learning, unfortunately, a rigorous mathematical proof of why layer and batch normalization speed up training and yield better performance is still missing; it comes purely from empirical results [4].

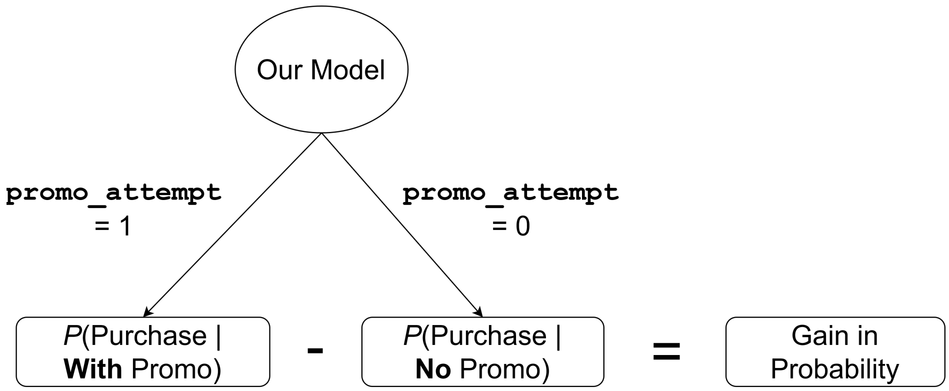

Figure 6: Prediction phase of our model.

differing purchase probabilities. Taking the difference of the aforementioned probabilities results in the change in probability from adding or removing a promotional code from that session.

3.7 Results

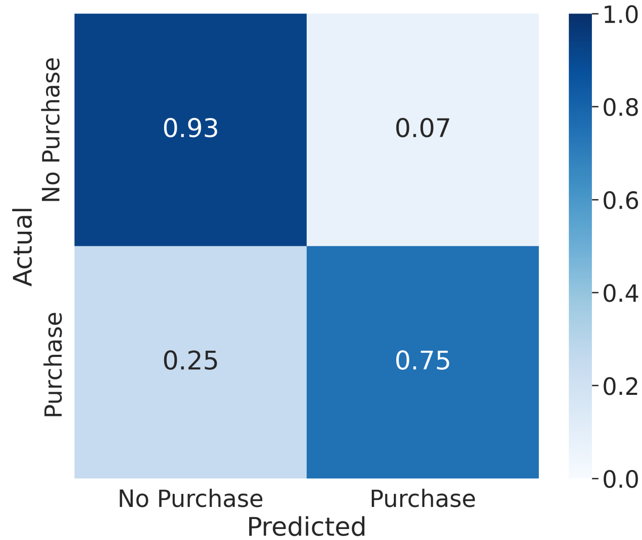

Figure 7: Confusion matrix for neural network model.

Our model is an effective classifier of customer purchases. It achieves an accuracy of 88% and a macro average F1-score of 0.85. Fig. 7 displays the model's confusion matrix, in which the decimal labels represent the proportion of visits correctly classified in the validation set grouped by the visit sessions' true labels. Although the model appears to be slightly biased towards predicting a visit will not end in a purchase, it still correctly classifies 75% of sessions that result in a purchase.

Because of the binary nature of the neural network's output, it also serves as a probabilistic model that provides the likelihood of a customer making a purchase. As previously described, the influence of promotional codes on the probability of purchase can be obtained by utilizing this ability. By partitioning users into four bins - 0-25%, 25-50%, 50-75%, and 75-100% - based on this gain in probability, generalizations can be made on how different types of visits can be characterized.

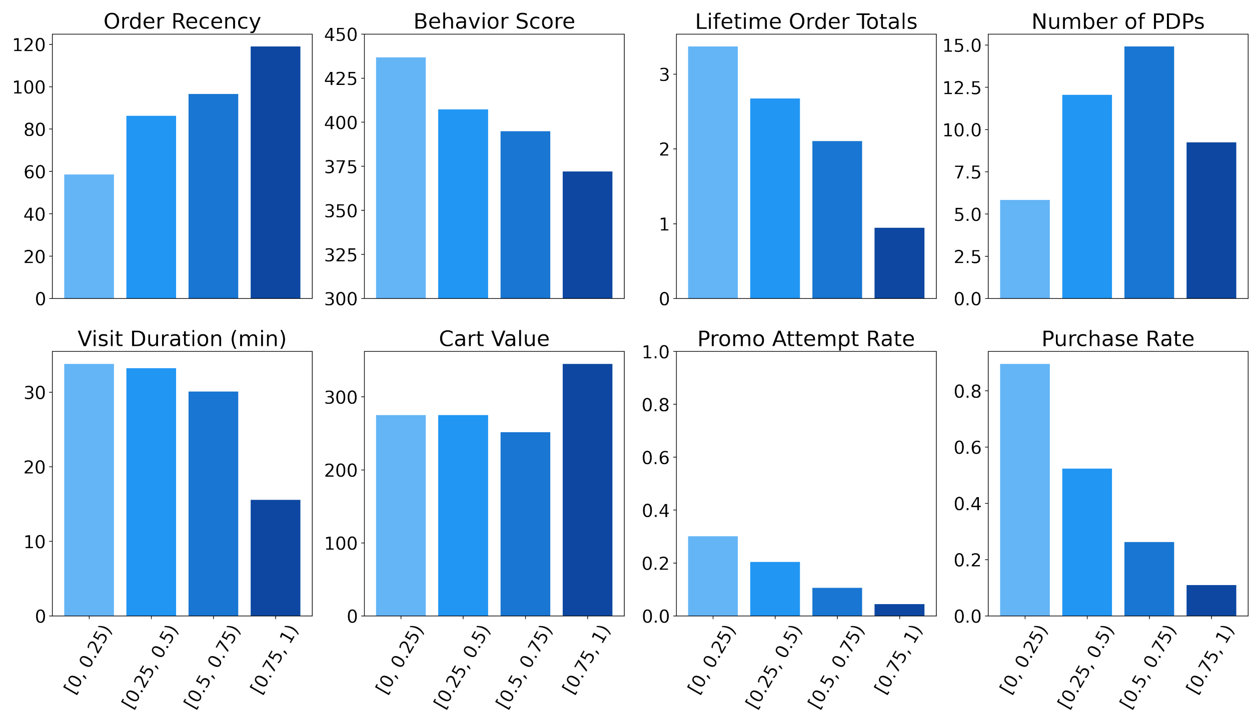

Figure 8: Average user and session attributes by the gain in purchase probability.

Fig. 8 highlights some of the average user and session attributes in each bin. The leftmost bar on each plot represents the group with the lowest gain in probability (0 to 25%). Users in this group are observed to have high behavior scores, lifetime order totals, and actual purchase rates among other attributes. This group is determined to be represented by returning users.

The rightmost bar on each plot represents the group with the highest gain in probability (75 to 100%). Users in this group are observed to have low behavior scores, lifetime order totals, and actual purchase rates among other attributes, opposite of the returning users. This group is determined to be comprised of new users.

On one hand, we find that new users are the most influenced by promotional codes. On the other hand, the more active a user is on Fingerhut, the less they are influenced by promotional codes. Once reaching checkout, 90% of the current returning users are already making purchases regardless of having a promotional code applied.

4 Conclusions

4.1 Summary of Results

- The neural network model is effective at determining whether a user will complete a purchase once reaching checkout and reveals an increase in a customer's purchase probability when a promotional code is attempted.
- Users that are heavily influenced by promotional codes exhibit characteristics such as low behavior scores, lifetime order totals, and actual purchase rates. They can be classified as new users.
- Users that are minimally influenced by promotional codes display characteristics such as high behavior scores, lifetime order totals, and actual purchase rates. They can be classified as returning users.

Our recommendations for Fingerhut include:

- New users with a low purchase probability at checkout should be targeted with promotional codes to maximize the likelihood of a completed purchase.
- Fingerhut should avoid targeting promotional codes at returning users with a low gain in probability.

4.2 Limitations

There are several limitations to our current modeling.

First, within our dataset, we are unable to distinguish how a customer acquired the promotional codes which they apply to the Fingerhut site. Because our project focuses on the effect of introducing a promotional code during the checkout phase, it is potentially problematic that our dataset does not indicate when the user actually received the promotional code. Consequently, we are not able to filter to users who were presented a promotion code at the checkout phase specifically. This introduces potential bias; for instance, a user may have reached checkout with a promotional code obtained off-site. Thus, they may be predisposed towards purchasing as they likely came to the site intending on making a purchase. Without more information on when or where this promotional code is given, we cannot assess that the promotional code is what solely swayed the user towards purchasing at checkout.

Another limitation of our model is its lack of applicability toward regular, non-holiday times of the year. Because the data that we used to train our model is both from the holiday season and from a limited time period of seven days, the results of our model may be skewed as they do not represent typical days throughout the year.

4.3 Next Steps

Our next steps involve both addressing the limitations of our model and enhancing it to capture more specificities.

A possible solution to address our inability to distinguish when a customer acquired a promotional code is to acquire more data about promotional code types from Fingerhut. This could potentially be in the form of more website tracking or a post-purchase survey indicating where the promotional code was received. This new information would help to exclude the sessions where the promotional codes are received before the checkout phase so that the model could be less biased toward purchasing.

There are also several ways in which we can expand our model. Our model currently predicts the change in probability of purchase due to any type of promotional code. However, our predictions could be more accurate if we could explore the effect of specific types of promotional codes on users. Some types of promotional codes can be more appealing to customers or more suitable to what they have in their cart, and certain types of users may prefer certain types of promotional codes over others. The results could inform recommendations on how Fingerhut can target specific monetary values of promotional codes in order to optimize the probability of purchase.

However, a model that recommends the promotional code which optimizes gain in probability of purchasing may consistently pick the code that produces the greatest reduction in price. This is a universally strong indicator of whether a user will purchase or not, but does not necessarily serve to optimize revenue. Therefore, we could also look at how revenue changes in the presence of different types of promotional codes. This can provide relevant information in situations where the application of a promotional code results in a greater probability of purchase, but lower revenue than desired by Fingerhut.

References

[1] Galen Crout. *What are promotional codes and how do they work?* Oct. 2021. URL: https://www.shopify.com/sg/blog/what-are-promotional-codes-and-how-do-they-work.

[2] Ashish. *Statistical tests: Feature selection using statistical tests*. Aug. 2022. URL: https://www.analyticsvidhya.com/blog/2021/06/feature-selection-using-statistical-tests/.

[3] Jonathan C. Kao. *Lecture notes in Neural Network and Deep Learning*. UCLA ECE C247, 2023.

[4] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.