Fingerhut FreshStart
Customer Activity Analysis

Junwon Choi, Alyssa Liu, Ronan Nayak

Statistics M148: Experience of Data Science
Department of Statistics & Data Science, UCLA

Executive Summary

This report focuses on Fingerhut's FreshStart customer data from 2021-2023, with the goal of exploring factors influencing new customers' decisions to activate their accounts and initiate their first purchase. Through data cleaning, manipulation, and modeling techniques, we successfully identified insights into customer behaviors during their journeys, the effectiveness of promotional campaigns, and the key factors contributing to the achievement of an “ideal” journey.

Key Findings

1. Customer Behavior Patterns: The study used Markov Chains to identify customer behavior patterns, focusing on stages in customer journeys rather than events to draw clear insights, revealing that certain stages in the customer journey significantly influence the flow between different events.

2. Promotional Influence: Our analysis on promotions reveals an impact on customer activation rates. Promotional activities, notably "Campaignemail Clicked", had the most impact on whether a customer eventually activated their accounts or not. However, this conflicted with our analysis on successful customer journeys, which showed that the “Campaignemail Clicked” rate was slightly lower for successful customers than unsuccessful customers. This insight suggests that email campaigns warrant a more detailed investigation, specifically on the promotion types and content. Additionally, many customers quit their journeys after starting the initial steps of making a purchase, indicating that they may not be satisfied or impressed with the available inventory.

3. Journey to Activation and Purchase: Through classification models, we discovered that the initial stages of a customer's journey and the number of days passed since the start of their journey highly influenced their progression to successful transactions. Our models indicate that early engagement and starting event/stage of a journey can significantly enhance customer activation and purchase rates.

Strategic Recommendations

- Optimize customer journey stages, particularly focusing on enhancing engagement in the early stages of the journey.
- Tailor promotions more strategically, focusing on offering customers promotions for specific products they’ve viewed, pushing them to start their journeys at specific stages, and exploring additional promotional methods other than email campaigns.
- Consider segmenting customers based on their interaction patterns and tailor communications and offers in order to maximize success rates.

Conclusion

Our analysis provides Fingerhut with actionable insights into customer behavior patterns, the impact of promotional strategies, and recommendations to optimize successful customer journeys. By focusing on strategic engagement and personalized promotions, Fingerhut can improve its FreshStart program's effectiveness, ultimately driving more successful customer journeys.

Statement of Problem

While there were various directions that we could have taken to analyze the provided dataset, our primary goal was to identify key factors that drive Fingerhut's FreshStart customers to activate their accounts and make their first purchases. We avoided integrating complex models to represent customer behavior and rather focused on simplifying the dataset as best as efficiently as possible. We believed that this would not only help us interpret and deliver our findings but also provide insight to Fingerhut on more simplified yet useful methods when categorizing and predicting their customer behavior.

We aimed to provide an answer to the following research questions based on Fingerhut's FreshStart customer data from 2021 to 2023.

1. What do typical customer behavior patterns look like? Are certain actions on Fingerhut's platform more or less likely to lead to another action?
2. Which customer behaviors or features make them more likely to follow the "ideal journey" as outlined by Fingerhut?
3. Which promotional materials launched by Fingerhut are more likely to result in initial purchases by customers?

Data

Data Issues & Cleaning

There were several issues and ambiguities with the provided dataset (export.csv). Below we identify the issues discovered and the steps we took to address them. We first note that export.csv contains 64,911,906 rows.

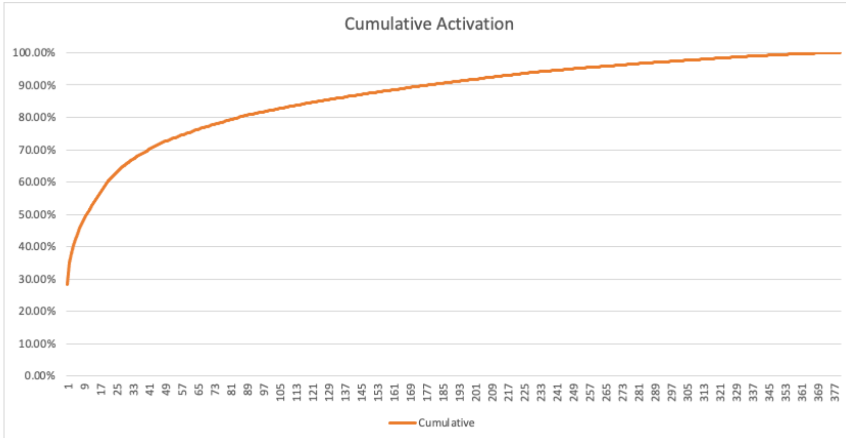
After taking an initial look at the dataset and the event_definition dataset, it became clear that the events "promotion_created" and "campaignemail_clicked" did not have corresponding ed_id values. We took the liberty of manually adding the following labels: "Promotion Created" corresponds with ed_id of 1, and "Campaignemail Clicked" corresponds with ed_id of 24.

Ignoring the journey_steps_until_end column, we found 8,058,871 duplicate rows. Although repeated activities from individual customers can contain useful information, we later confirmed that system-wide records caused by internal testing at Fingerhut have the same events and timestamps logged multiple times per customer. These duplicate records were removed from the dataset, and we reindexed the journey_steps_until_end column for each customer.

We wanted to focus on investigating customer journeys long-term, specifically what factors would help determine a successful customer from an unsuccessful one. The original dataset had 68,351 customers with multiple accounts, which took up ~4% of the entire dataset. In order to standardize the data more and control for multiple journeys, we decided to treat each unique customer_id and account_id combination as a separate customer. From this point on, when we refer to customers, we will be referring to rows with a unique customer_id and account_id identifier.

From there, we took a deeper look at customers who had multiple account activations. After some data wrangling, we discovered that 1,319,326 customers had no account activation, 414,268 had one account activation, and 2,173 had either two or three account activations. It appears that those customers with two or three account activations had multiple journeys in the data. Again, to control for multiple journeys and to focus on investing successful versus unsuccessful journeys, we decided to remove the customers who had more than one account activation. These customers also only accounted for about 0.001% of total customers and 0.4% of our total dataset.

Another issue we encountered was differentiating between unsuccessful journeys and incomplete journeys. The latest date in the original dataset was in September of 2023, leaving many customers with incomplete journeys rather than unsuccessful ones.



[Figure 1: Cumulative Activation Rate - provided by Fingerhut]

According to Fingerhut and the visualization above, if it has been over 60 days since activation with no order placed, it's very rare that a user will end up placing an order - meaning that this is the primary differentiating factor between incomplete and unsuccessful customer journeys. That being said, we removed a customer from the dataset if they had activated their account in the last 60 days but not yet placed an order because of the possibility of them placing an order after the dataset had been cut off.

Additional Data Manipulation

To prepare the data for further analysis and modeling, as well as ensuring efficient memory management, we generated different datasets suitable for various modeling applications. Below we outline the two main datasets that we generated and the models that they were used in.

The first format that we created was a wide-format dataframe, where each customer corresponds to a single row. Attributes such as ed_id, event_name, event_timestamp, journey_steps_until_end were stored as lists for each customer. All these attribute lists have the same length for each customer.

	customer_id	account_id	ed_id	event_name	event_timestamp	journey_steps_until_end
0	-2147206560	2082689427	[12, 1, 19, 5, 11, 3, 4, 6, 4]	[application_web_approved, promotion_created, ...	[2023-05-02 20:20:18+00:00, 2023-05-02 20:58:1...	[1, 2, 3, 4, 5, 6, 7, 8, 9]
1	-2145360520	1467252181	[19, 19, 19, 19, 19, 19, 19, 3, 19, 12, 4, 4, ...	[application_web_view, application_web_view, a...	[2022-01-08 00:40:01+00:00, 2022-01-08 00:40:0...	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...
2	-2144991170	-2081722203	[19, 19, 19, 19, 19, 19, 3, 19, 12, 4, 4, 4, 4...	[application_web_view, application_web_view, a...	[2022-09-19 14:47:59+00:00, 2022-09-19 14:48:0...	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...
3	-2142086624	-484534102	[2, 12, 4, 4, 4, 5, 6, 1, 1, 6, 24, 1, 21	[campaign_click, application_web_approved, bro...	[2021-07-07 06:00:00+00:00, 2021-07-07 18:58:0...	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...

[Figure 2.1: Wide-format dataframe]

This data structure allowed us to identify 1,665,431 unique customers and more efficiently manage the dataset. Based on the attribute lists, we were able to extract key information for areas that we planned to analyze. Figure 2.2 below displays how we performed one-hot encoding based on the customers' application submission, activation, and promotion exposure status.

ed_id	stage	event_timestamp	application	activation	promotion_exposure	place_order_web	place_order_phone	order
[2, 12, 1, 4, 4, 4, 11, 1, 5, 6, 1, 1, 4, 11, ...	[Discover, Apply for Credit, Promotion Created...	[2021-09-03T06:00:00.000Z, 2021-09-03T21:46:20...	1	1	1	1	0	
[2, 4, 11, 4, 4, 11, 4, 4, 4, 11, 4, 4, 5, 4, ...	[Discover, First Purchase, First Purchase, Fir...	[2021-02-20T06:00:00.000Z, 2021-02-20T22:23:39...	1	1	1	1	0	
[19, 19, 19, 19, 19, 19, 19, 19, 19, 19, 3, 19, 19, 12, 4,...	[Apply for Credit, Apply for Credit, Apply for...	[2022-12-11T07:43:02.000Z, 2022-12-11T07:43:04...	1	0	1	0	0	
[12, 2, 22]	[Apply for Credit, Discover, Discover]	[2023-07-02T13:43:31.000Z, 2023-07-02T19:43:31...	1	0	1	0	0	

[Figure 2.2: Wide-format dataframe: one-hot encoding]

Figure 2.3 further shows how we extracted exactly which promotional materials each customer was exposed to. More specifically, the following ed_id values were taken as promotional materials:

ed_id	event_name
1	Promotion Created
2	campaign_click
9	customer_requested_catalog_digital
20	catalog_email_experian
21	catalog_mail
24	Campaignemail Clicked

place_order_web	place_order_phone	order_shipped	place_order	promotion_type	2	9	20	21	1	24
1	0	1	1	[1, 2]	1	0	0	0	1	0
1	0	1	1	[1, 2]	1	0	0	0	1	0
0	0	0	0	[2]	1	0	0	0	0	0
0	0	0	0	[2]	1	0	0	0	0	0

[Figure 2.3: Wide-format dataframe: one-hot encoding (promotion types)]

This particular dataframe was then utilized in the logistic regression model designed to identify key promotional variables driving customer purchase behavior.

Through this manipulation, we were also able to calculate the following numbers for activation and purchase rates:

<table>
<tr><td rowspan="4">Total
(1,665,431)</td><td rowspan="2">Activated
(410,862)</td><td>Purchase
(354,404)</td><td>21.28% of total</td></tr>
<tr><td>No purchase
(56,458)</td><td>3.39% of total</td></tr>
<tr><td rowspan="2">Not activated
(1,254,569)</td><td>Purchase
(25,433)</td><td>1.53% of total</td></tr>
<tr><td>No purchase
(1,229,136)</td><td>73.80% of total</td></tr>
</table>

[Figure 3.1: Percentage calculation based on activation and purchases]

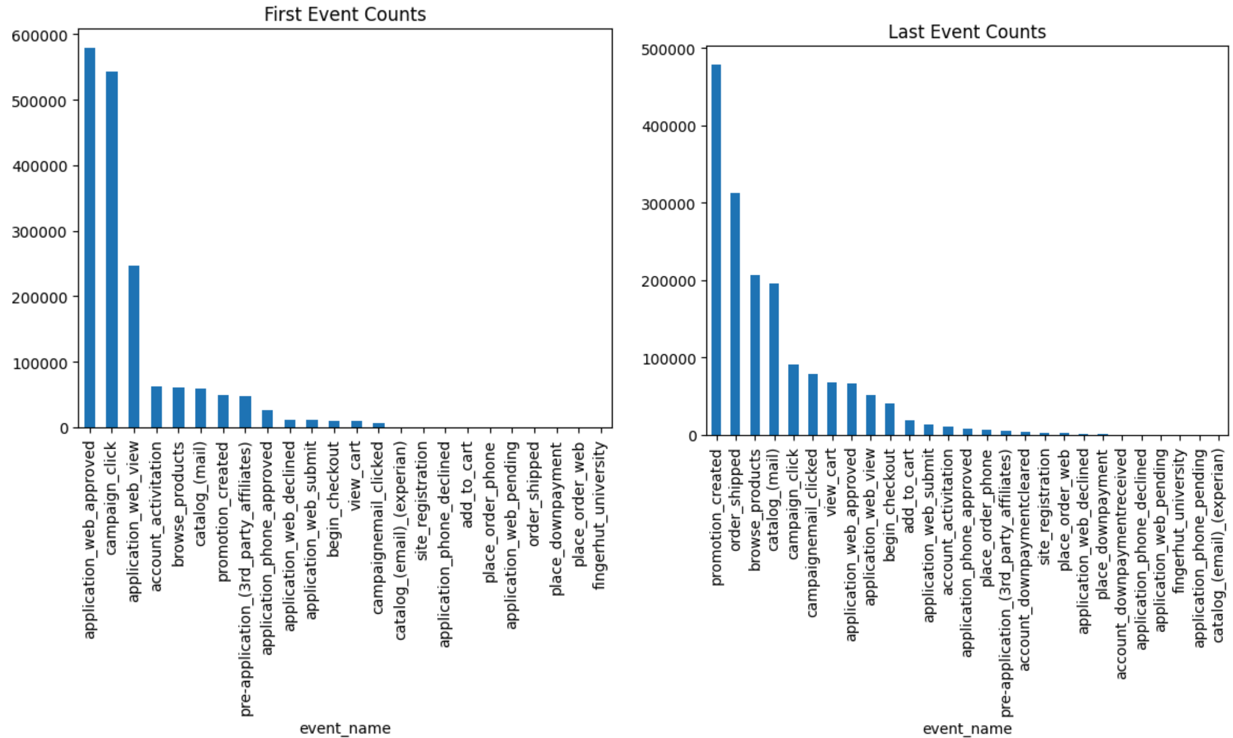
<table>
<tr><td rowspan="2">Total promotion exposure customers
(1523909)</td><td>Activated
(372432)</td><td>24.44% of total promotion exposure customers</td></tr>
<tr><td>Not activated
(1151477)</td><td>75.56% of total promotion exposure customers</td></tr>
</table>

[Figure 3.2: Percentage calculation within customers exposed to promotions]

Though the wide format data was much easier to follow and read, a major limitation arose from this, as machine learning models in Python cannot typically take in columns of lists as data, especially since the lists in the data are all of varying lengths. We decided to turn to feature engineering instead, but we were aware that we wouldn't be able to capture all the intricacies of the original data.

The first concept that we focused on modeling was predicting whether or not a customer would have a successful journey. During the feature engineering process, we made sure to only include attributes that could be determined at the beginning of or during a customer journey. For example, we included the first event and the first stage in the customer journey, rather than things like the last event or last stage in the journey.

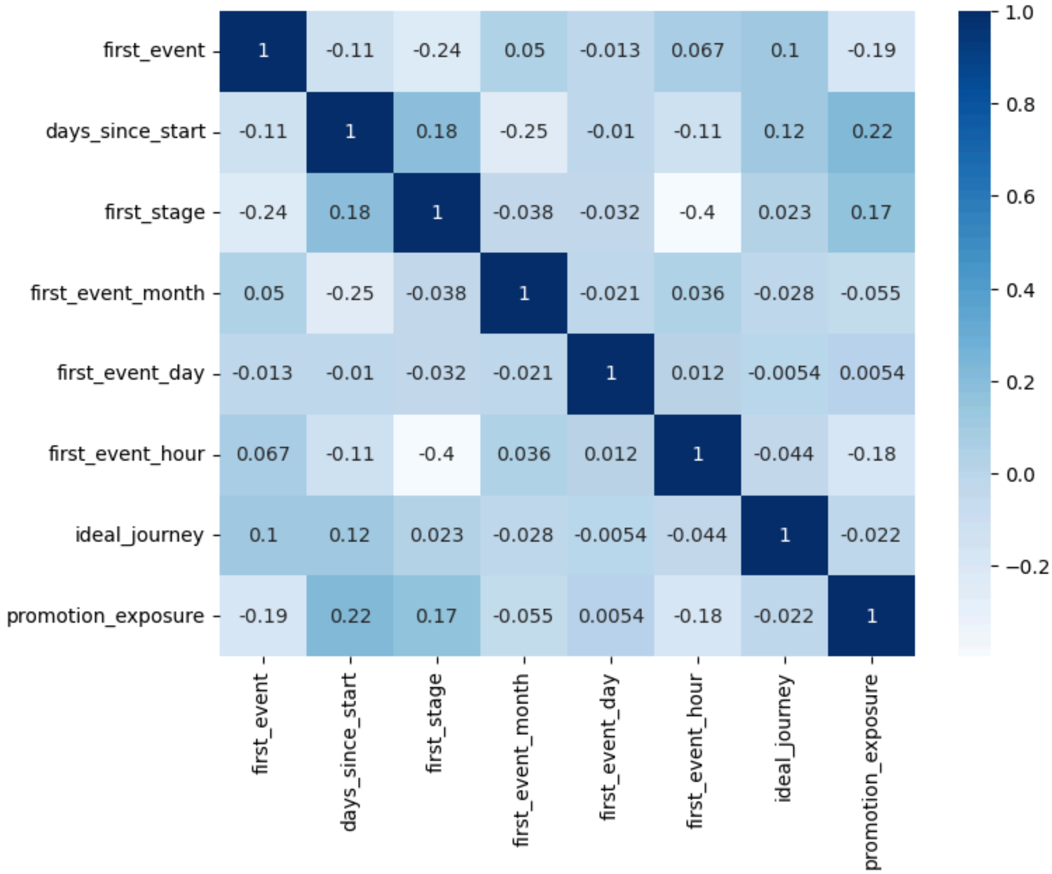
The exact features that we created were first_event, days_since_start (the number of days between the first date in the customer journey and the dataframe cutoff date), first_stage, first_event_month (the month when the first event took place), first_event_day, first_event_hour, promotional_exposure (whether or not the customer was exposed to promotional material), and ideal_journey. ideal_journey was a binary variable that had 1 if the customer journey contained all attributes of what was defined to be an ideal journey as outlined by Fingerhut: apply for credit, first purchase, downpayment, and order shipped, and 0 otherwise.



[Figure 4: Frequencies of first and last event counts]

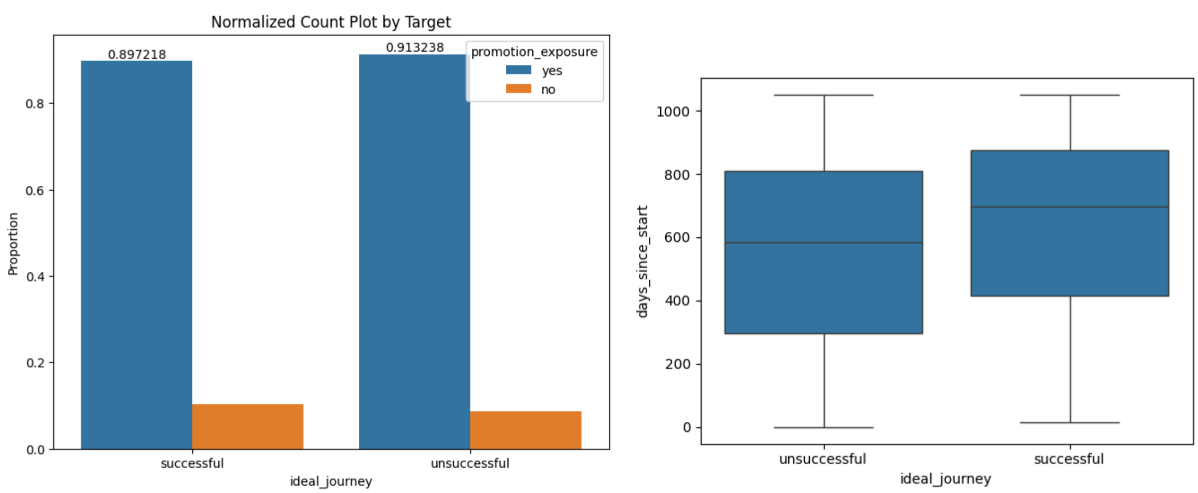
We proceeded with classification models with ideal_journey as the target variable. We also scaled the data using sklearn's standard scaler, to account for distance-based algorithms and control for differences in scale.

To ensure that the features aren't highly correlated, we generated the following heatmap:



[Figure 5: Heatmap of feature-engineered variables]

Before implementing classification models, we perform some initial exploration of variable distribution between the two classes of the target variable.



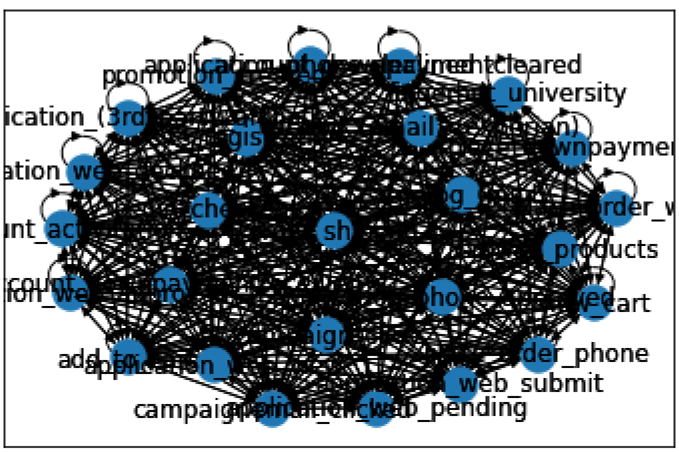
[Figure 6: Assessing dataset balance based on target variable (ideal_journey)]

Methods & Results

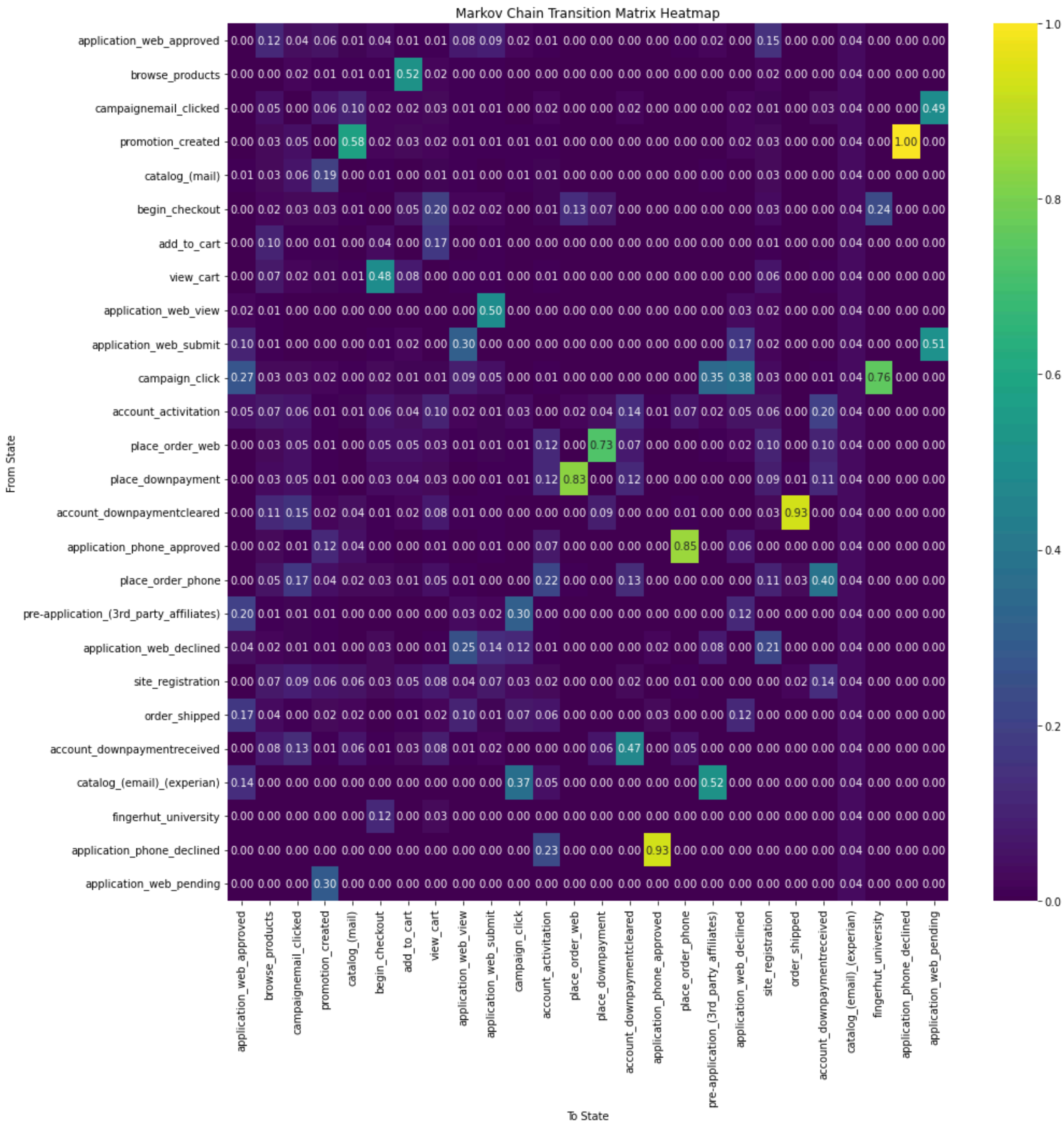
Markov Chains: Customer Patterns

Markov Chains were used to test our initial hypothesis of past customer behavior and sequences predicting future behavior. Using a randomly sampled version of the original dataset with duplicates removed, we tried to create Markov chains utilizing all event names from the event_name column. Unfortunately, this resulted in incredibly dense and convoluted heatmaps and Markov chain visualizations due to the enormous number of unique customers as well as the number of events. These models were created using the networkX library in Python, chosen for its ability to represent and analyze Markov chains. Due to no specific Markov chain functionality within this package, directed graphs were used as the building blocks for these chains, and led to further issues as illustrated below. This methodology led to an incomprehensible visualization accompanied by a heatmap.

The main assumptions to create these Markov chains included declaring that an event is unique and thus the same event cannot happen twice in succession. This assumption was determined due to the generally nearby timestamps of many repeated events, indicating a double click, and not a new event state. Despite this assumption that would naturally result in a heatmap with 0.0 diagonal values, there is one column catalog_mail_experian that was never visited by any customer and thus was given equal probabilities at all points. From this, this event was declared insignificant and removed. Despite the uninterpretable visualization, after deciding to use the mchmm library instead, the heatmap values were corroborated when using the same assumptions.



[Figure 7: Markov visualization incorporating all events]

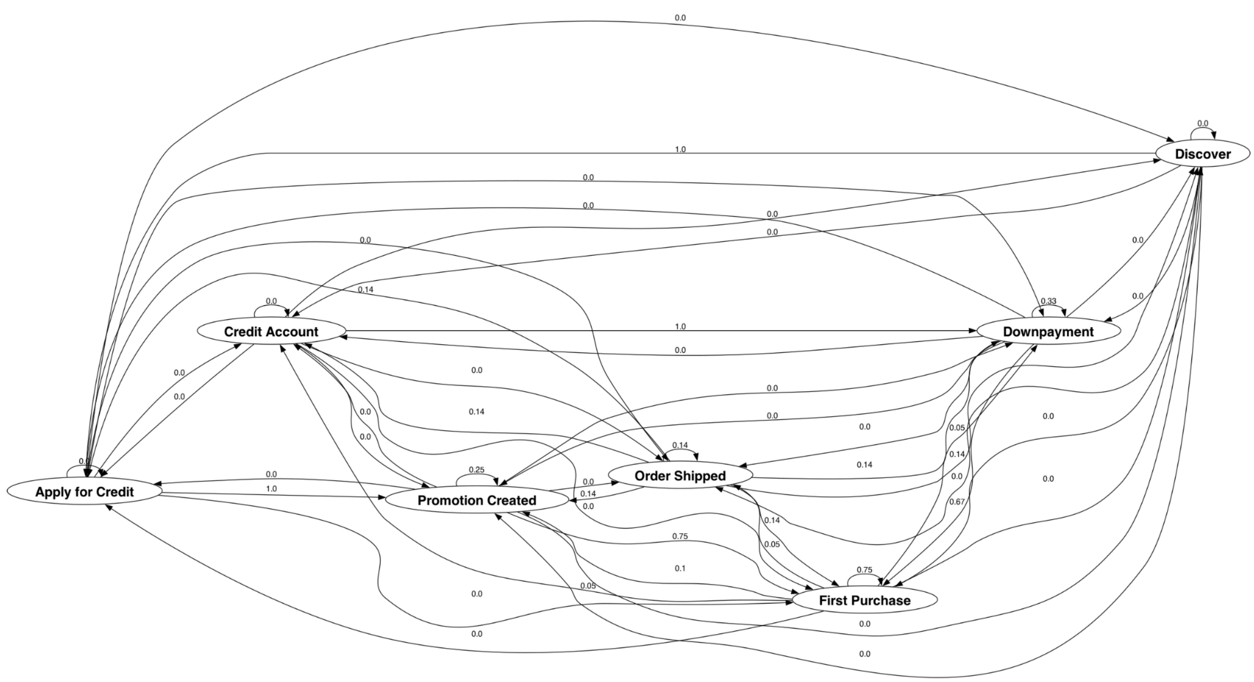


[Figure 8: Heatmap incorporating all events]

Following the creation of these Markov chains with the NetworkX library, we migrated to the mchmm library, chosen for its easy creation of Markov chains directly from sequences of observed data. This library also provided valuable tools in regards to computing and manipulating transition matrices as well as visualizing the structure of these chains. We tried to mitigate the issues of these overly crowded Markov chains, inspecting how visualizations would look for individual customer sequences with the goal of determining the value of incorporating all event names as unique steps in the customer path sequence. We experimented with removing 'insignificant' event steps but this resulted in an inaccurate representation of customer progressions.

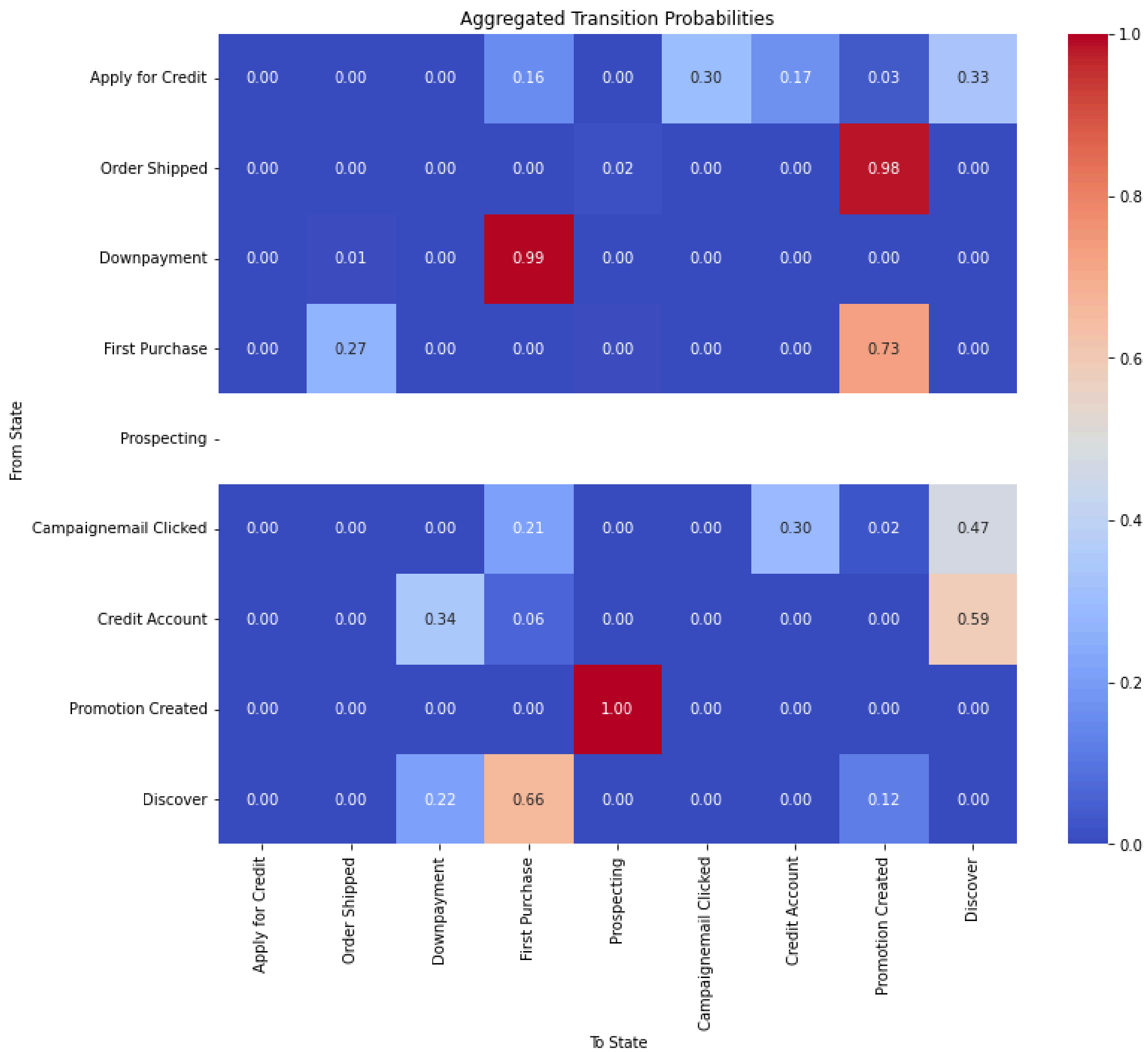
Our ultimate resolution to this issue was to group the event names down to their stages, as defined in the provided event_definitions dataset. This segmentation of events provided a reasonable method of inspecting customer journeys with the exception of the lack of assignment for event_id's '1' and '24', corresponding to 'Promotion Created' and 'Campaign Email Clicked', respectively. These were each granted their own Stages and were named 'Promotion Created' and 'Campaignemail Clicked', accordingly. The fact that these two events were granted their own stage is something to keep in mind when interpreting the following Markov chain representations. The data was then cleaned and manipulated to create only one row for each unique Customer ID. This new dataset contained relevant columns ed_id in which each row provided a list object of the ed_ids visited by each customer in order of occurrence, and 'stage' which provided a list object of the Stage names as defined by the event_definitions dataset as well as the newly created 'Promotion Created' and 'Campaignemail Clicked' stages and an additional 'Order Shipped' stage only corresponding to the 'order_shipped', also in order of occurrence. The 'order_shipped' event was given its own stage as it was determined to be the end goal of an ideal customer journey and thus significant on its own. Each of these stages included between one and eight unique events, as can be found in the event_definition documentation, and provided means for a much cleaner understanding and visualization of the sequences of events taken by each customer.

The assumption that an event cannot be repeated became insignificant as there were multiple events per stage. This can be seen below in the visualization of the sequence of events for a singular specific customer.



[Figure 9: Single Customer Markov Visualization with Engineered Stages]

Following the cleaning of the dataset with the creation of these list object rows (ed_id, stage), there were still over 1.6 million rows of unique customers, resulting in a desire to randomly sample for the creation of these Markov chains. 16,000 unique customers were randomly selected and a corresponding Markov chain was created for the customer's sequence. We then created an aggregated matrix based on observed transitions within the unique Markov chains. These aggregated transition probabilities were then visualized with a heatmap with the assumption that a stage cannot repeat in succession. This assumption was created with the goal of remaining consistent with the rest of the analysis which does not allow for repeated events.



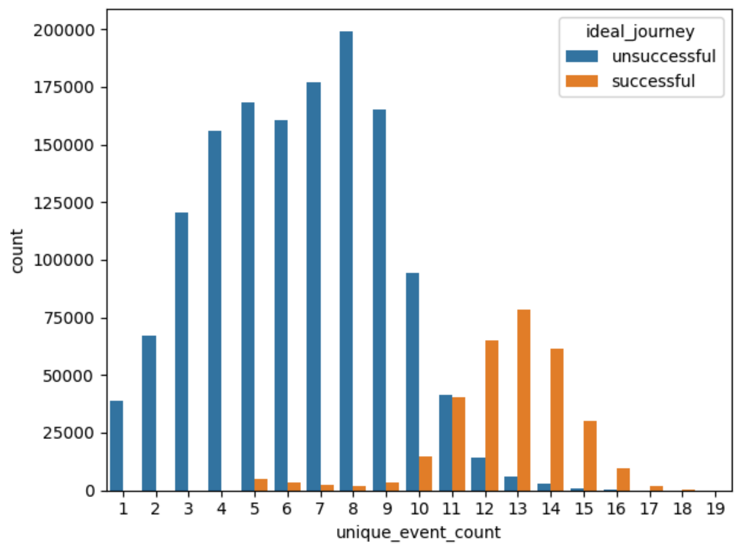
[Figure 10: Aggregated transition probabilities heatmap]

From this transition matrix as well as the earlier transition matrices we find that the prospecting stage is irrelevant. This stage includes catalog_email_experian and catalog_main, events that should be removed as event flags in the customer journey. We find that a downpayment leads to the first purchase with almost certainty, as makes sense given the process of buying within the Fingerhut, Freshstart Credit Program. We also find some informative results. For example, we find that more than 50% of the time, events relating to account activation and review lead to customers engaging in some form of discovery that includes clicking on a campaign, requesting a digital catalog, visiting any of the site's 3rd party affiliates, or site registration. Following this discovery phase, users make their first purchase 66% of the time, indicating a correlation between this stage and users engaging in a productive way with the Fingerhut, FreshStart Credit Program service.

This aggregation of event names into stages poses some issues within this representation and interpretability. Because this only maps the exact subsequent step, there is an unexpected lack of correlation in the heatmap. For example, we may not expect 'Order Shipped' to lead to 'Promotion Created' consistently, yet it does 99% of the time. This is because 'Order Shipped' is the last stage in the sequence before 'Promotion Created' begins it again. These results lead to gaps in the interpretability of these Markov chains in some cases, despite the insight that they provide. The size and unique events that comprise this dataset restrict other Markov chain approaches and thus we progressed to other methods of providing valuable insight into customer behavior such as classification and clustering techniques.

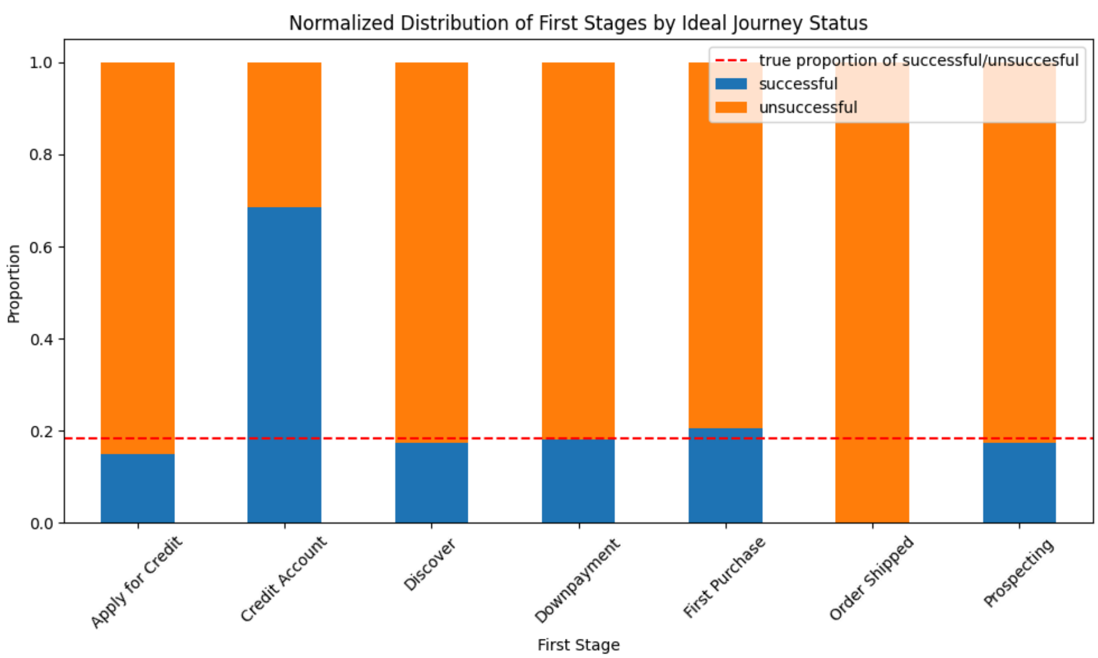
Successful vs. Unsuccessful Ideal Journeys EDA

We begin with some preliminary EDA to get an idea of what exactly differentiates customers with successful journeys and those with unsuccessful journeys.

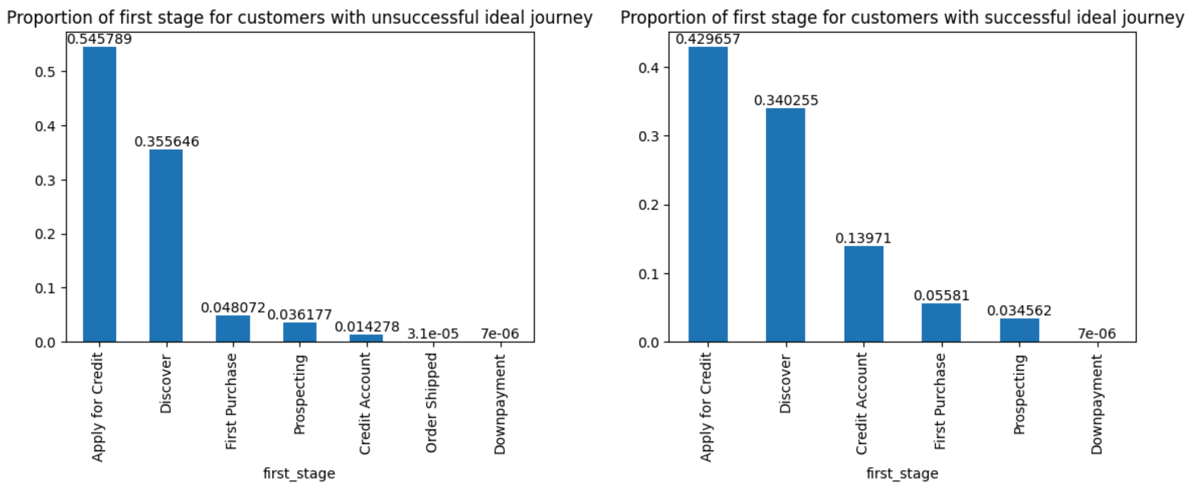


[Figure 11: Count of unique events for successful vs unsuccessful journeys]

We also conducted an inspection of first stage for successful vs. unsuccessful journeys:



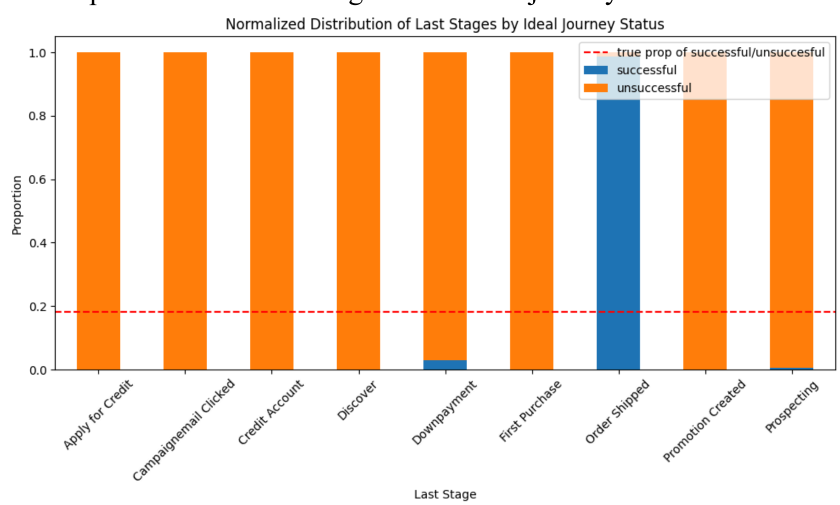
[Figure 12: Normalized first stage distribution for successful vs unsuccessful journeys (stacked)]



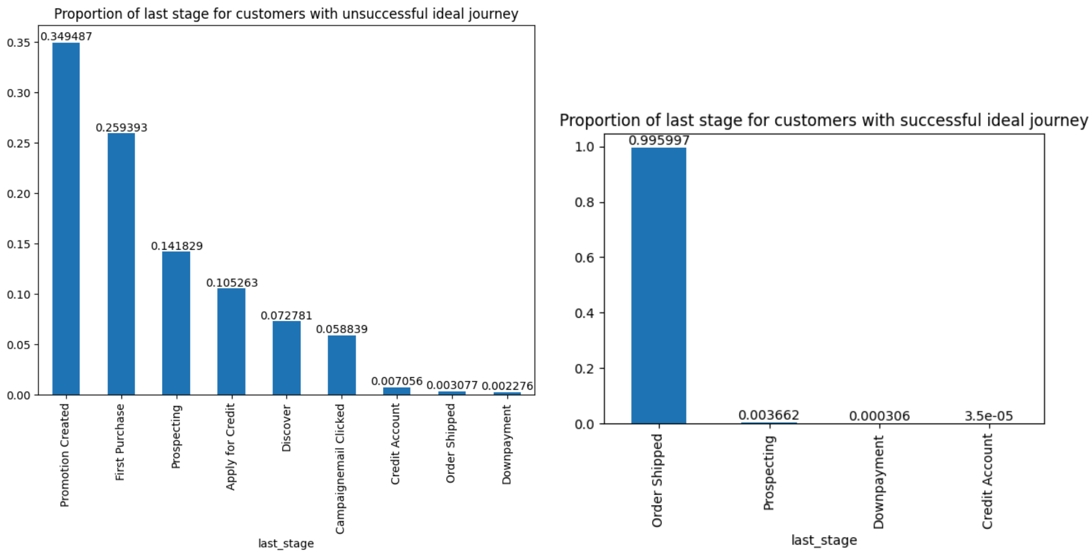
[Figure 13: Normalized first stage distribution for successful vs unsuccessful journeys]

Interestingly, it seems that there are certain customers who have had an order shipped without necessarily reaching the previous steps of an ideal journey. However, after further inspection, we see that there is a 99% correlation between having an order shipped and having an ideal journey for customers, which is what we would expect.

We perform a similar inspection for the last stage of customer journeys:

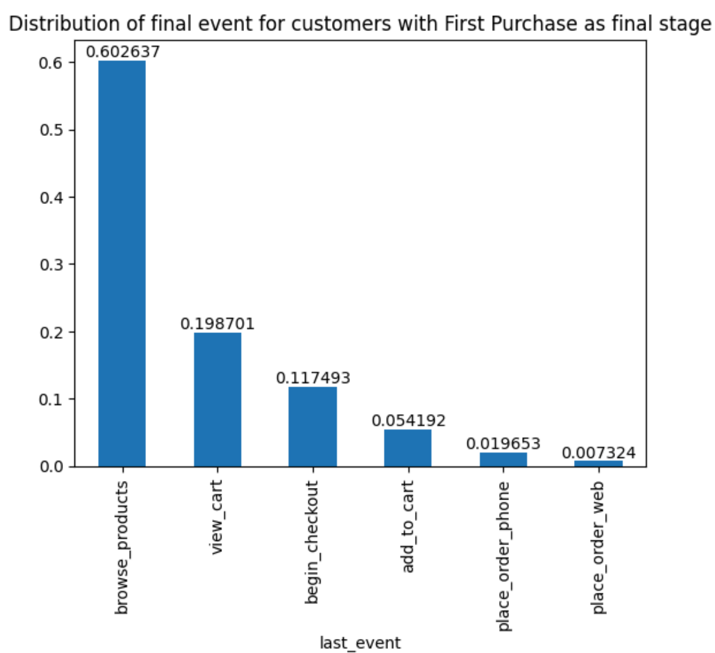


[Figure 14: Normalized last stage distribution for successful vs unsuccessful journeys (stacked)]



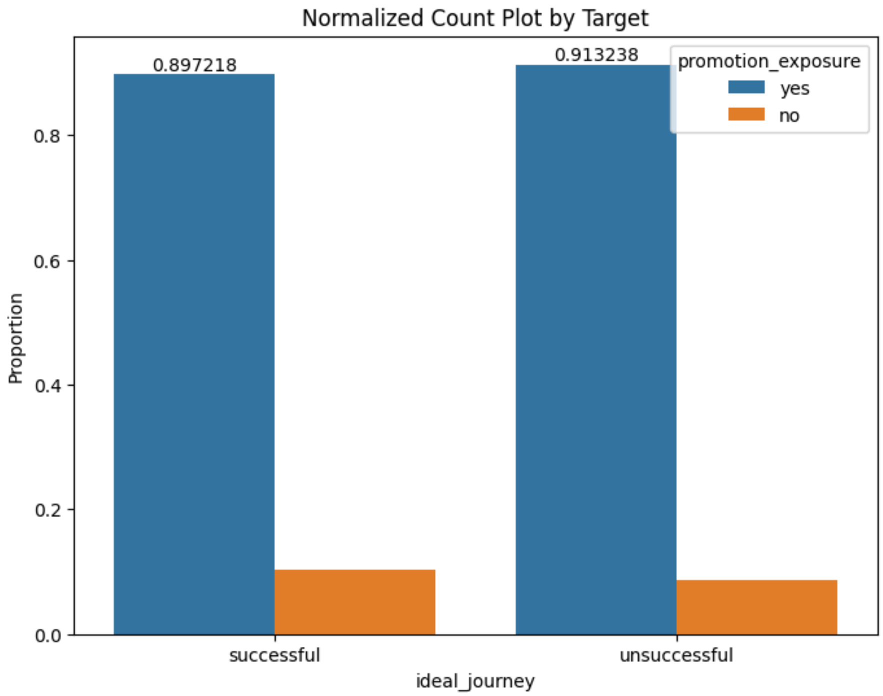
[Figure 15: Normalized last stage distribution for successful vs unsuccessful journeys]

As expected, almost all the customers with successful ideal journeys end with orders shipped as their last stage. We can see that around 35% of customers with unsuccessful journeys have promotions created as their last step. About 26% of customers also end with First Purchase as their last stage.

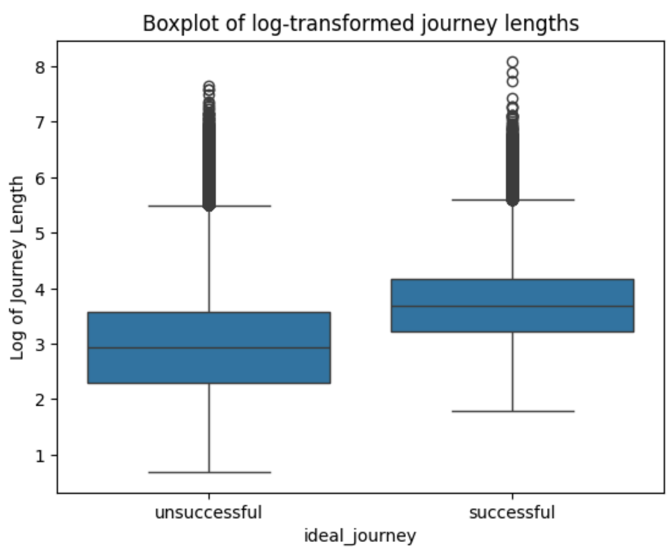


[Figure 16: Normalized final event distribution for customers with “first purchase” as final stage]

Here, we see that many customers who have First Purchase as their final stage actually just stop after they’ve browsed products for a bit. Very few eventually make it to the checkout or place order events.



[Figure 17: Normalized counts of promotion exposure for successful vs unsuccessful journeys]
Interestingly, a slightly higher proportion of customers with unsuccessful journeys were exposed to promotions, indicating that Fingerhut promotions may not necessarily be crucial to whether a customer has a successful journey or not.



[Figure 18: Log-transformed journey lengths for successful vs unsuccessful journeys]

Additional exploratory data analysis reveals that customers with successful ideal journeys generally have more unique events in their journeys, which makes sense because many unsuccessful customers don't make it through the initial stages of a journey. Journey lengths are generally a bit longer for successful customers, likely because many unsuccessful customers quit before they get far in the journey.

Classification Model: Successful vs. Unsuccessful Ideal Journeys

We began with a basic logistic regression model. After 10-fold cross-validation, the average accuracy obtained was about 82%. At first glance, this may seem like a high number, but because the dataset is imbalanced with about 82% of customers with an ideal journey and 18% without an ideal journey, this model doesn't perform better than the baseline. If we simply just guessed every customer to have an ideal journey, we would have obtained the same accuracy. This is also confirmed by the extremely low recall and F1 scores of 0.005 and 0.011, respectively.

	precision	recall	f1-score	support
unsuccessful	0.82	1.00	0.90	465695
successful	0.83	0.01	0.01	104706
accuracy	-	-	0.82	570401
macro avg	0.82	0.50	0.46	570401
weighted avg	0.82	0.82	0.74	570401

[Figure 19: Logistic regression model performance]

The dataset imbalance means the model ends up performing well on the majority class of the data but not well on the minority class. From here, we used several different approaches to try and increase the accuracy to be above baseline.

The first was passing in the argument class_weight = 'balanced', which adjusts the class weights to be inversely proportional with class frequencies. As you can see below, there was a decrease in overall accuracy, but the other metrics increased significantly, meaning the model can perform much better on the minority class.

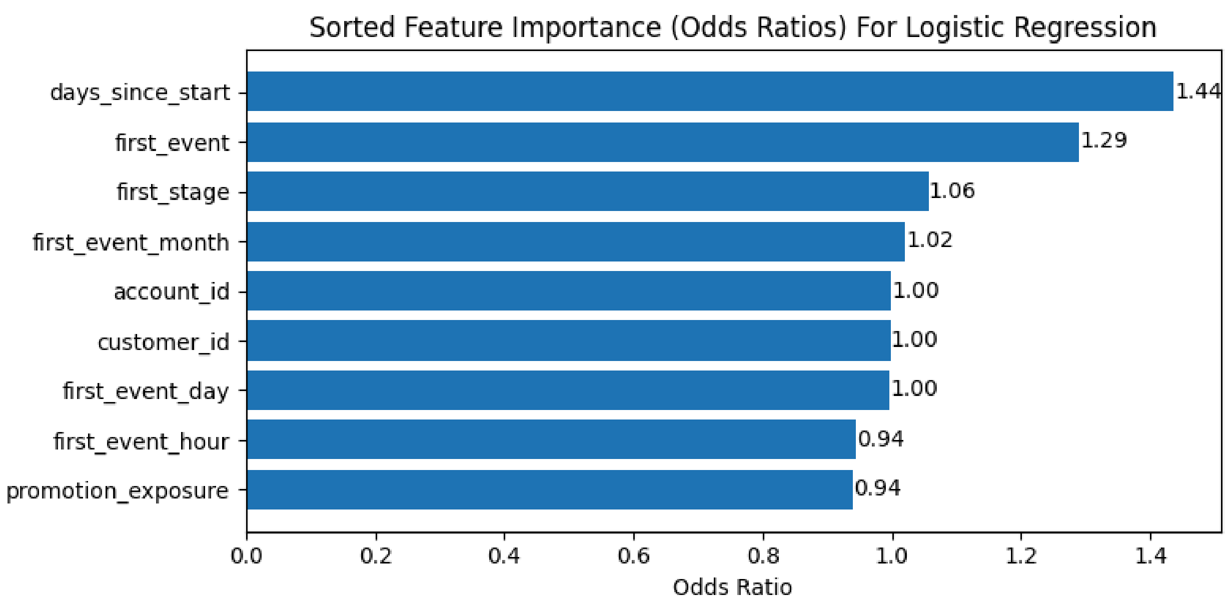
	precision	recall	f1-score	support
unsuccessful	0.86	0.57	0.68	465695
successful	0.23	0.58	0.33	104706
accuracy	-	-	0.57	570401
macro avg	0.55	0.58	0.51	570401
weighted avg	0.74	0.57	0.62	570401

[Figure 20: Logistic regression model performance - balanced class]

We also investigated the coefficients for each predictor, and got the following results:

Predictor	Coefficient	Odds Ratio
days_since_start	0.3724	1.4512
first_event	0.2768	1.3189
first_event_hour	-0.0817	1.0851
first_stage	0.0407	1.0415
first_event_month	0.0162	1.0163
first_event_day	-0.0021	0.9979
promotion_exposure	-0.1027	0.9024

[Figure 21: Coefficient of logistic regression predictors]



[Figure 22: Logistic regression odds ratios for predictor variables]

This indicates that the most important variables in the classification model are the number of days since the start of the customer's journey and the first event in their journey. For every one-day increase in the number of days since the start of the customer's journey, the odds of a successful journey increase by 1.44 times. There is a naturally occurring order to the stages in the dataset, according to the ID numbers provided in the definition data frame. Based on this, it seems that the further the first stage is in the order, the more likely the customer will be to finish the journey. More specifically, for every one-unit increase in the first stage ID, the odds of a successful journey increase by 1.29.

We repeated the same process for a Random Forest Classifier with similar but more impressive results.

	precision	recall	f1-score	support
unsuccessful	0.86	0.68	0.75	465695
successful	0.26	0.51	0.34	104706
accuracy	-	-	0.65	570401
macro avg	0.56	0.59	0.55	570401
weighted avg	0.75	0.65	0.68	570401

[Figure 23: Random forest classifier model performance]

This model yielded higher metrics than most of the other models, so we proceeded with hyperparameter tuning for the Random Forest Classifier. We fit three folds for 10 possible hyperparameter combinations and experimented with the number of trees in the forest, max depth of the tree, min number of samples to split an internal node, min number of samples at a leaf node, and the number of features to consider when looking for the best split.

When tuning, we used AUC-ROC score rather than accuracy to determine which hyperparameter combination yielded the best performing model, as this metric works better for imbalanced datasets. The best-performing model had an AUC-ROC score of 0.6654029813394534, and we obtained the following results:

	precision	recall	f1-score	support
unsuccessful	0.87	0.67	0.76	465695
successful	0.27	0.55	0.36	104706
accuracy	-	-	0.65	570401
macro avg	0.57	0.61	0.56	570401
weighted avg	0.76	0.65	0.68	570401

[Figure 24: Random forest classifier model performance after hyperparameter tuning]

While we see slight improvements in the model's overall performance, we also explored additional gradient boosting machines, as these tend to be robust to overfitting and can perform well on imbalanced datasets.

We followed the same hyperparameter tuning steps for XGB Classifier, but this time we implemented scaling of label weights. We used the scale_pos_weight parameter to control the balance of positive and negative weights with a value of 4.462067794531725, which was the number of negative labels (unsuccessful) / the number of positive labels (successful). We also experimented with the number of estimators, learning rate, subsample size, max depth of a tree, the proportion of columns to be random samples for the trees, and the min sum of weights of observations required in a child. The best-performing model obtained an AUC_ROC score of 0.6677974466833927, only slightly higher than the random forest performance.

	precision	recall	f1-score	support
unsuccessful	0.87	0.65	0.74	465695
successful	0.27	0.57	0.36	104706
accuracy	-	-	0.64	570401
macro avg	0.57	0.61	0.55	570401
weighted avg	0.76	0.64	0.67	570401

[Figure 25: XGB classifier model performance after hyperparameter tuning]

The most complex models we experimented with were neural networks. The output of the first neural network is shown below. The model had three layers and used both ReLU and sigmoid activation. We proceeded with binary cross-entropy loss and an Adam optimizer. We also used the ReduceLROnPlateau callback in Keras to reduce the model's learning rate once the validation loss stopped improving. We ran the model for 50 epochs but still saw very minimal improvements in both the validation loss and the accuracy.

```
Epoch 1/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 96s 1ms/step - accuracy: 0.6001 - loss: 0.6380 - val_accuracy: 0.6038 - val_loss: 0.6347 - learning_rate: 0.0010
Epoch 2/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 87s 1ms/step - accuracy: 0.6055 - loss: 0.6339 - val_accuracy: 0.6051 - val_loss: 0.6339 - learning_rate: 0.0010
Epoch 3/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 86s 1ms/step - accuracy: 0.6066 - loss: 0.6331 - val_accuracy: 0.6042 - val_loss: 0.6344 - learning_rate: 0.0010
Epoch 4/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 87s 1ms/step - accuracy: 0.6066 - loss: 0.6331 - val_accuracy: 0.6053 - val_loss: 0.6332 - learning_rate: 0.0010
Epoch 5/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 100s 1ms/step - accuracy: 0.6065 - loss: 0.6324 - val_accuracy: 0.6059 - val_loss: 0.6329 - learning_rate: 0.0010
Epoch 6/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 97s 1ms/step - accuracy: 0.6073 - loss: 0.6322 - val_accuracy: 0.6055 - val_loss: 0.6331 - learning_rate: 0.0010
Epoch 7/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 119s 2ms/step - accuracy: 0.6069 - loss: 0.6322 - val_accuracy: 0.6062 - val_loss: 0.6333 - learning_rate: 0.0010
Epoch 8/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 114s 2ms/step - accuracy: 0.6073 - loss: 0.6324 - val_accuracy: 0.6056 - val_loss: 0.6333 - learning_rate: 0.0010
Epoch 9/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 108s 2ms/step - accuracy: 0.6069 - loss: 0.6323 - val_accuracy: 0.6062 - val_loss: 0.6325 - learning_rate: 0.0010
Epoch 10/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 111s 2ms/step - accuracy: 0.6072 - loss: 0.6319 - val_accuracy: 0.6059 - val_loss: 0.6330 - learning_rate: 0.0010
Epoch 11/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 108s 2ms/step - accuracy: 0.6080 - loss: 0.6319 - val_accuracy: 0.6043 - val_loss: 0.6331 - learning_rate: 0.0010
Epoch 12/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 111s 2ms/step - accuracy: 0.6072 - loss: 0.6318 - val_accuracy: 0.6065 - val_loss: 0.6326 - learning_rate: 0.0010
Epoch 13/50
...
Epoch 49/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 96s 1ms/step - accuracy: 0.6092 - loss: 0.6306 - val_accuracy: 0.6075 - val_loss: 0.6315 - learning_rate: 1.0000e-05
Epoch 50/50
70588/70588 ━━━━━━━━━━━━━━━━━━━━ 101s 1ms/step - accuracy: 0.6094 - loss: 0.6303 - val_accuracy: 0.6076 - val_loss: 0.6315 - learning_rate: 1.0000e-05
```

[Figure 25: Keras neural network training]

We ran a similar model implementation in PyTorch but received similar results. In both neural networks, the similar values for loss and validation loss indicate that the model is likely underfitting, even with the learning rate being adjusted.

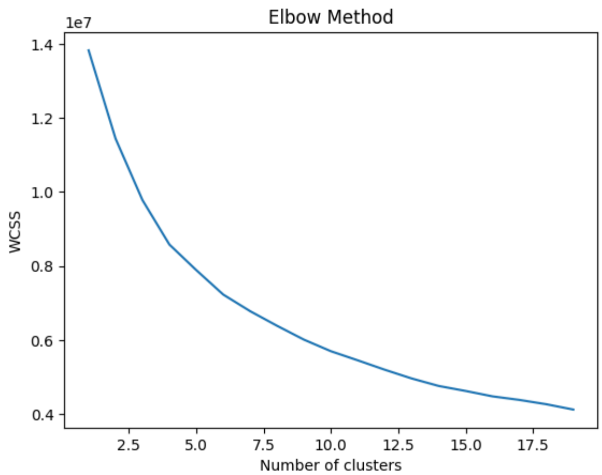
So while our AUC_ROC scores and other metrics exceeded baseline measurements (baseline for AUC_ROC would be 0.5), there is still a long way to go in improving the model to predict whether or not a customer would eventually reach a successful journey.

The underfitting and generally low scores/loss are likely a result of the limited features that we created, as we limited the features to those that could be extracted from any customer journey, even one that just began. Unsurprisingly, time since the beginning of the customer's journey and first event appeared to be the most important features in this classification problem.

Clustering Approach

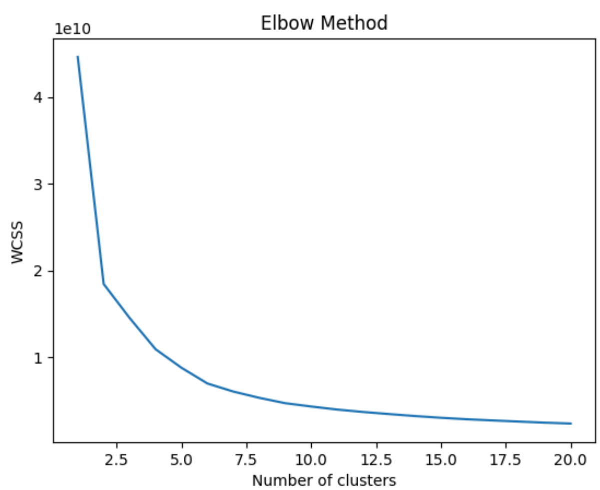
For clustering, we decided to use the same feature-engineered data as used in the classification models and similarly scaled the data with sklearn's standard scaler.

For 1-20 clusters, we plotted WCSS (the sum of variance between the observations in each cluster) against the number of clusters and identified where the "elbow" of the curve was. This point appeared at around 6 clusters and indicates where the WCSS starts to plateau as clusters increase, which is what we proceeded with.



[Figure 26.1: Elbow curve of scaled feature-engineered data]

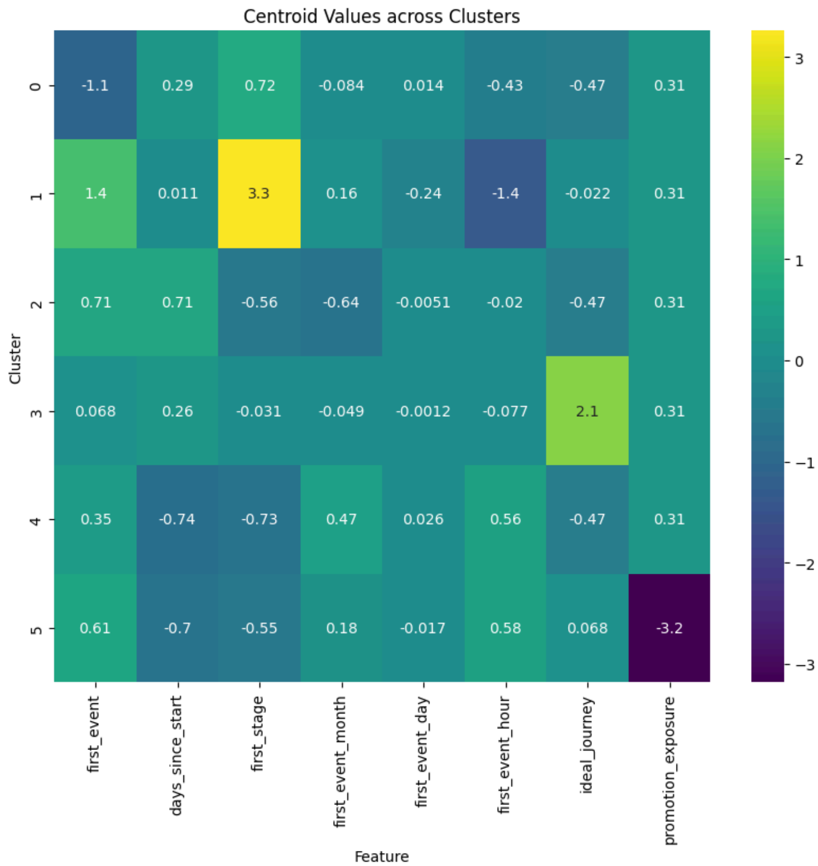
We address a clear limitation here, which is that the elbow isn't super distinct, but there is a clear drop-off point in the decrease in WCSS. We also ran the same code on a more descriptive dataset, which included one-hot-encoded columns for whether a journey contained specific stages and events, and we obtained the following curve:



[Figure 26.2: Elbow curve of scaled feature-engineered data (additional features)]

The elbow here is much more distinct, confirming that 6 is the number of clusters that we should proceed with.

To visualize the differences between the clusters, we first create a heatmap with the different centroid values for each cluster:



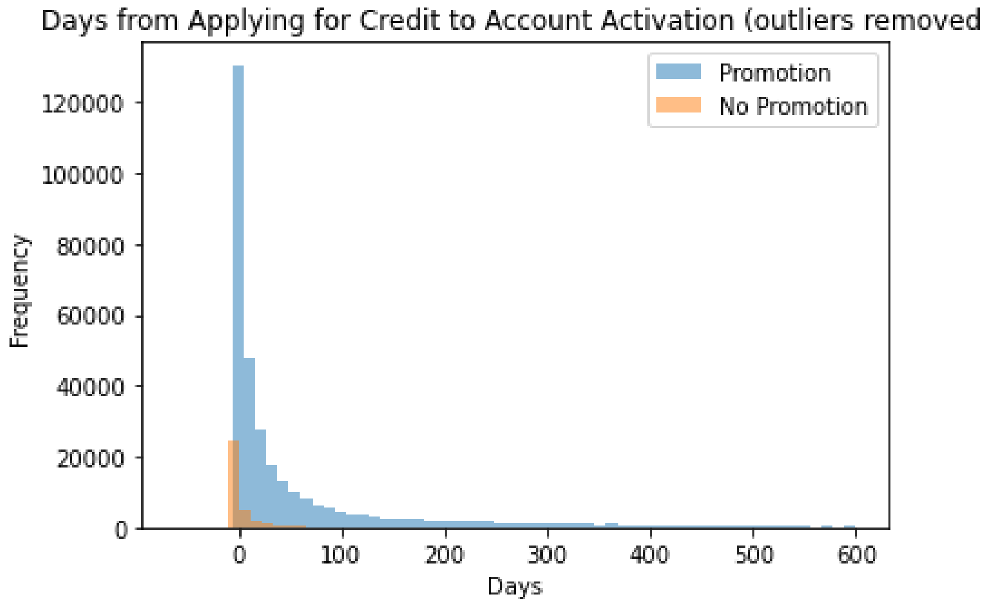
[Figure 27: Comparison of centroid values across clusters]

When examining the heatmap above, the most clear differences between centroids appear in the first_event, first_stage, and first_event_hour columns, indicating that these features may be the most important when differentiating between customer types.

Promotion EDA

Based on the results from our initial classification model investigating factors that contribute to determining successful vs unsuccessful journeys, we discovered that exposure to promotions led to less success. Although it had a relatively low contribution to determining the success of customer journeys overall, we felt it was necessary to further analyze whether or not promotions influenced activation rates. To investigate this issue further, we implemented logistic regression to determine whether or not promotions are an effective strategy for Fingerhut, and if so, which promotions have been the most effective so far in motivating decisions.

We begin with plotting a histogram to assess the duration of customers applying for credit and activating their accounts based on their exposure to promotions:



[Figure 28: Days between applying for credit and activating account]

We can see that most customers are exposed to some type of promotional material when applying for credit and activating their accounts. Though not conclusive, this may be indicative of promotions influencing activation rates with FreshStart customers. We note that on a relative scale, both groups of customers show a similar distribution, with most customers activating their accounts within a couple of days of applying for credit.

Classification Model: Effective Promotions

As outlined above, we utilized our wide-format data frame to one-hot encode the types of promotions customers received, as well as whether or not they were exposed to promotions at all.

	precision	recall	f1-score	support
Activation	0.86	0.50	0.64	250911
No activation	0.33	0.75	0.46	82176
accuracy	-	-	0.57	333087
macro avg	0.60	0.63	0.55	333087
weighted avg	0.73	0.57	0.59	333087

[Figure 29: Logistic regression model (target: activation, predictors: promotion types) performance - balanced class]

Predictor (ed_id / event_name)		Coefficient	Odds Ratio
24	Campaignemail Clicked	0.6607	1.9362
1	Promotion Created	0.3192	1.3760
-	promotion_y_n	0.0378	1.03855
9	customer_requested_catalog_digital	0.0	1.0
2	campaign_click	-0.1431	0.8666
20	catalog_email_experian	-0.3926	0.6753
21	catalog_mail	-1.4094	0.2443

[Figure 30: Coefficient and odds ratio of logistic regression predictors: promotion types]

The average accuracy after 10-fold cross-validation, the average accuracy obtained was around 56.60%. We note that the model could benefit from more detailed categorization or specification of promotion types within Fingerhut's database.

Based on odds ratio values, we observed that 'Campaignemail Clicked' has the biggest influence on customer account activation rates. We could attribute clicking campaign emails to providing a direct access point for customers to explore the site and eventually leading to account activation.
Given that approximately 86.26% of customers who activated their accounts ended up making a first purchase, setting activation rates as a key performance indicator (KPI) remains valid for Fingerhut.

Discussion

Based on the Markov Chain analysis and visualization, we find that simplification of the customer journey into stages as opposed to unique events is an effective way to analyze and visualize customer behaviors. This approach provided cleaner insights into how customers move through different phases of engagement with Fingerhut. We implore Fingerhut to use the findings from the Markov Chain analysis to identify stages that have conversion rates that exist but can be improved (e.g., 'Apply for Credit' to 'Discover') as well as those with unexpectedly low conversion rates (e.g., 'Promotion Created' to 'Discover') and invest in understanding what drives conversions, or lack thereof, at these points. Consider A/B testing different approaches to optimize these key stages. We also recommend tailoring communications and promotions to customers based on their current stage in the journey. For customers at critical conversion points, targeted incentives or information could help them progress toward the next stage. Beyond stages, we recommend the segmentation of customers based on their pathways through this Markov Chain, or further Markov Chain Analysis. Personalizing experiences based on these segments and recognizing that different customer groups may have unique needs and preferences could be beneficial.

Given the aggregation of events and sampling of customers, some transitions in the aggregated heatmap may still be counterintuitive or lack clear interpretation, such as the consistent transition from 'Order Shipped' to 'Promotion Created'. Caution should be exercised in drawing direct causations from these transitions. Also, despite the relatively large random sample taken for Markov Chain analysis, it is still recommended to be wary as the dataset has many unique customers and randomly sampling 16,000 unique customers from a large dataset introduces the potential for sampling bias. The selected subset may not fully represent the broader customer base, impacting the generalizability of findings. It is also recommended to be aware that the assumption that the same event cannot happen twice in succession may not fully capture the nuances of customer behavior. Future analyses should consider the potential implications of this assumption on findings.

For further research, we would advise exploring simplified models and conveying the necessity to have access to more comprehensive data. Integrating external factors or further customer attributes into the analysis can provide a more holistic view of customer behavior and uncover additional, meaningful insights. The limited nature of the dataset allowed for customer behavior analysis that could have been more insightful and allowed for an understanding of what specifically drives customer paths given more customer and internal data. Given the complexity of customer journeys, future analyses could benefit from further simplifying models by focusing on critical transitions and stages that have the most significant impact on customer behavior.

Based on the models and visualizations created to compare successful journeys and unsuccessful journeys, promotion exposure didn't seem to have a huge impact on whether a customer would be successful or not. 91.32% of unsuccessful customers were exposed to promotions, which was higher than the 89.72% of successful customers who were exposed. About 35% of unsuccessful customers also ended their journeys after receiving a promotion from Fingerhut.

However, we did observe that if we set activation rate as our primary KPI, then online campaign emails were effective tools at driving activations. From these results, we would recommend that Fingerhut keep singular factors such as activations as the KPI rather than setting a more strict ideal journey of customers. Furthermore, the results from the promotion analysis provide motivation to investigate which specific types of campaign emails or online promotions are more effective at increasing the activation rates of customers. We would also recommend reminding customers if they have items in their cart or giving them specific promotions for products they may have viewed.

The same goes for sending out reminders or promotions for customers based on the first event in their journey. Customers with successful ideal journeys only made up about 20% of the dataset, but they made up about 70% of customers who had Credit Account as their first stage. This indicates that customers who start with a Credit Account as their first stage have a significantly higher chance of successfully completing the ideal journey as outlined by Fingerhut.

The other stages that appear first in Figure 12 have a pretty comparable split to 80% unsuccessful and 20% successful journeys. Based on this, Fingerhut could focus on sending reminders and promotions to customers who do not have Credit Account as their first stage. Alternatively, when sending out promotional emails to customers who have not yet created accounts, you could suggest they start with events that correspond with Credit Account, as this could lead to higher rates of successful ideal journeys.

Additionally, about 26% of customers end with the First Purchase stage, 60% of which specifically browse products as their last event. Because of these high proportions, we would recommend that Fingerhut expand the range of their available inventory for purchase, as it seems many customers don't proceed with actually making a purchase after seeing what's available for purchase.

Overall, in the logistic regression model for classifying successful and unsuccessful journeys, the variables with the odds ratios furthest from one ended up being the number of days since the start of the customer's journey, as well as the starting stage of the customer's journey. Since these are the variables that have the largest impact on the odds of whether a journey is successful or not, these are the variables we should focus most on controlling for new customers, whether that is through promotional emails or other strategies.