Instacart Market Basket Analysis

An Exercise in Predicting Future Customer Purchases

Final Capstone Project
LaShawn Gaines

Background Information

Instacart operates in a rapidly growing and competitive online grocery shopping market. As the company expands, it becomes increasingly important to understand the nuances of customer behavior and to possess the ability to cater to their individual preferences. In a December 9, 2022 blog post, Instacart noted a potential for an e-commerce penetration rate as high as 35% by 2027. The COVID-19 pandemic helped spark significant growth in the grocery e-commerce business, and now e-commerce retailers like Instacart are looking for ways to maintain that momentum.

Market Basket Analysis is a powerful tool that can be used to uncover patterns in customer purchasing behavior by revealing which products are frequently purchased together. On May 3, 2017, Instacart released a public dataset with 200,000 anonymized Instacart users and a sample of 3 million of their grocery orders. It was a call to aspiring data scientists to build models to predict future purchases for Instacart customers using Market Basket Analysis.

Key Stakeholders

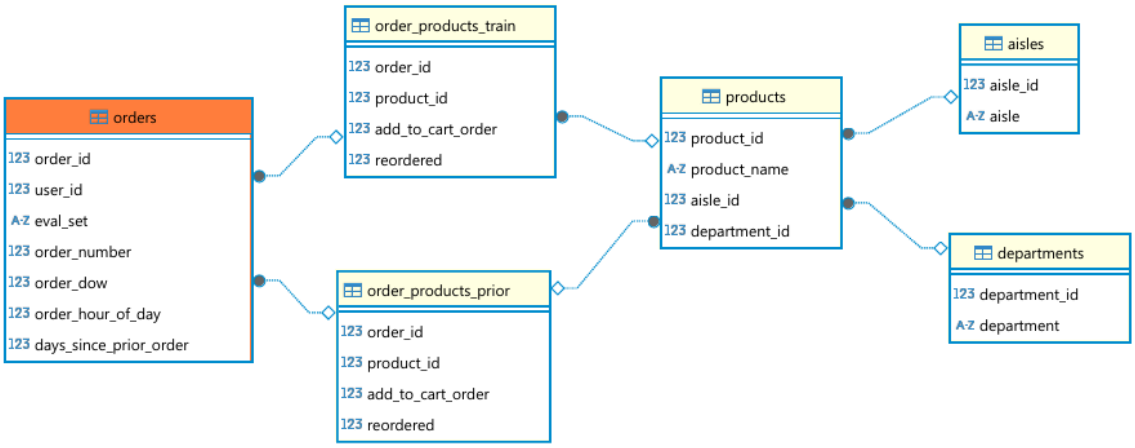
The key stakeholders for this market basket analysis for Instacart could be:

- Instacart Leadership: especially those responsible for making strategic decisions for the company
- Instacart Marketing team: may be able to use potential insights gained in order to drive customer engagement, retention and purchases
- Instacart customers: may have a better experience based on the insights gained and actions taken as a result of the project

About the Dataset

Instacart released this dataset on kaggle as part of a machine learning competition in 2017. The winners would receive a cash prize and a fast track through Instacart's employee recruitment process. This dataset was contained within 6 relational files

containing information for over 3 million grocery orders (each order contains 4 to 100 products) for 200,000 Instacart users. The schema is below:



One of the kaggle contest creators, jeremystan, provided some clarification on the data tables. I have summarized the content of these tables below:

1. The **orders** table contains 3.4 million order records spanning 206 thousand Instacart users. The data has been anonymized to protect the user's personal information. For example, purchase dates have been removed, and have been replaced with only a sequential number for each order and user (1 = first, n=nth). Instacart also categorized the transactions into three categories: prior, train, and test.
2. The **products** table contains 50 thousand records, 1 for each product offered. They are categorized by aisle_id and department_id.
3. The **aisle** table contains 134 unique rows, with the corresponding aisle names.
4. The **departments** table contains 21 unique rows, with the corresponding department names.
5. The **order_products_SET** tables (one for train and one for prior) contain over 30 million rows of data. Each row provides a product and the numerical sequence in which each item was placed into the user's shopping cart.

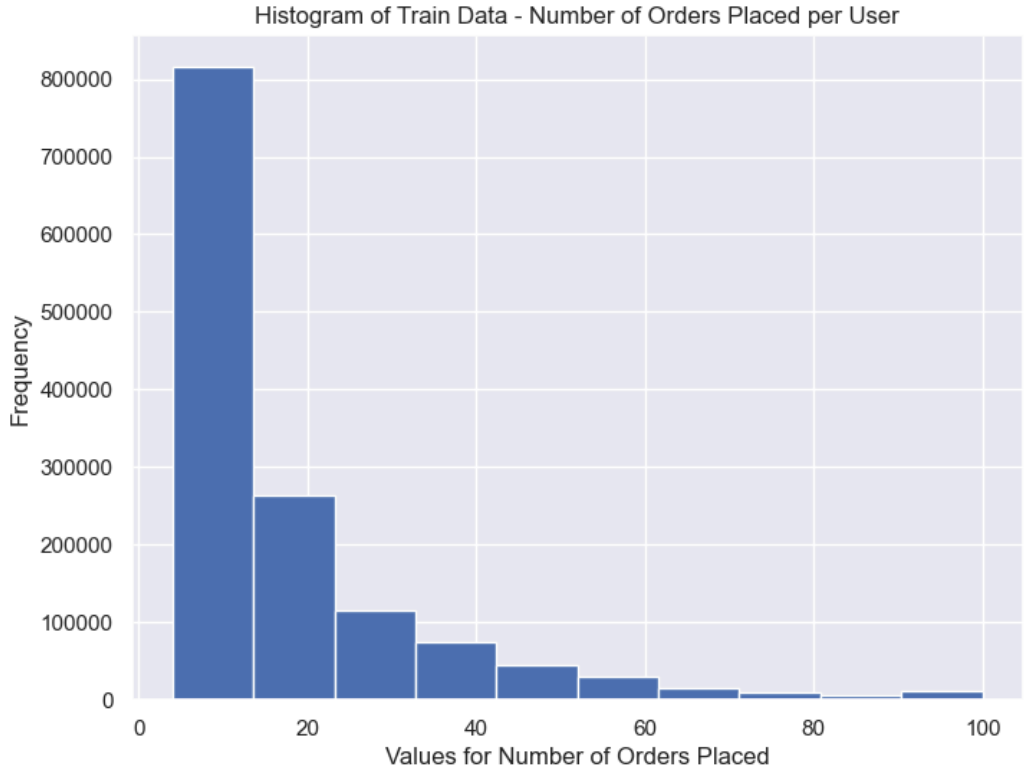
Data Wrangling

The dataset from Instacart was very clean. Each column for the tables listed, all contained appropriate data types and only one table contained missing data. The only

table that contained missing values was the **orders** table. The days_since_prior order field was capped at 30 days. Of the 3.4 million records in the **orders** table, nearly 2.1 million had missing values. At this stage of the project, no updates were needed for the dataset.

EDA

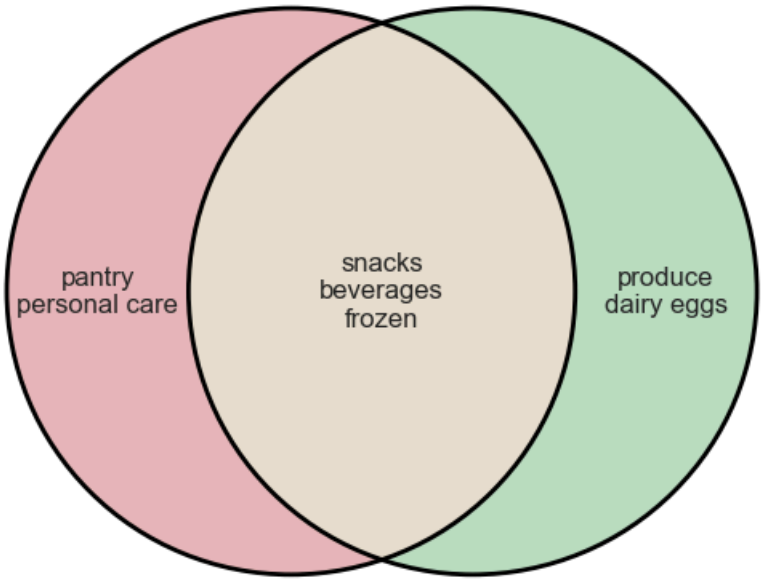
For my first pass at EDA, I leveraged the YData profiling tool to analyze the data in my products and order_products_SET tables. All of the correlations noted were fairly obvious, for example, the re-order status is highly positively correlated with the number of orders placed. However it did help to identify some areas of skewness with the datasets. In particular, the number of orders placed is an example of this. The mean number of orders placed per customer is 17, with a median value of 11 and max value is 145. This will factor into our customer segmentation process.



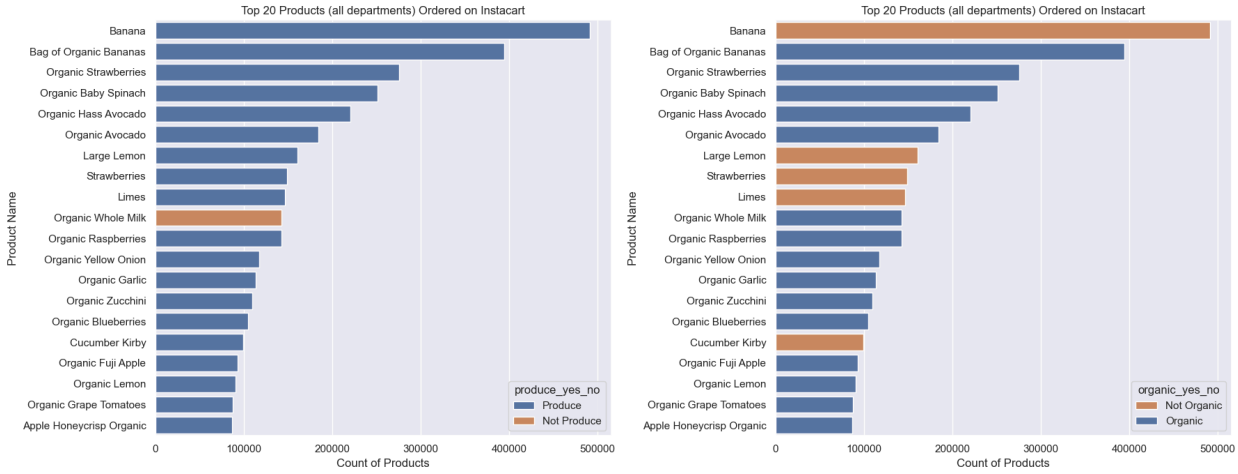
Interestingly enough, through the EDA process I noted that the number one product ordered by Instacart users was bananas. Bananas are categorized within the top department for orders, which is Produce. This does not align with Instacart's product offerings. The top product offering on the platform is personal care products. While it

holds the top spot in product offerings, personal care products rank only 14th in user orders. This could be useful information for the vendors selling goods through the Instacart application. When comparing the top 5 products offered on the platform vs the top 5 products ordered from the platform, 3 departments appear in both lists. Those three product departments are snacks, beverages and frozen goods.

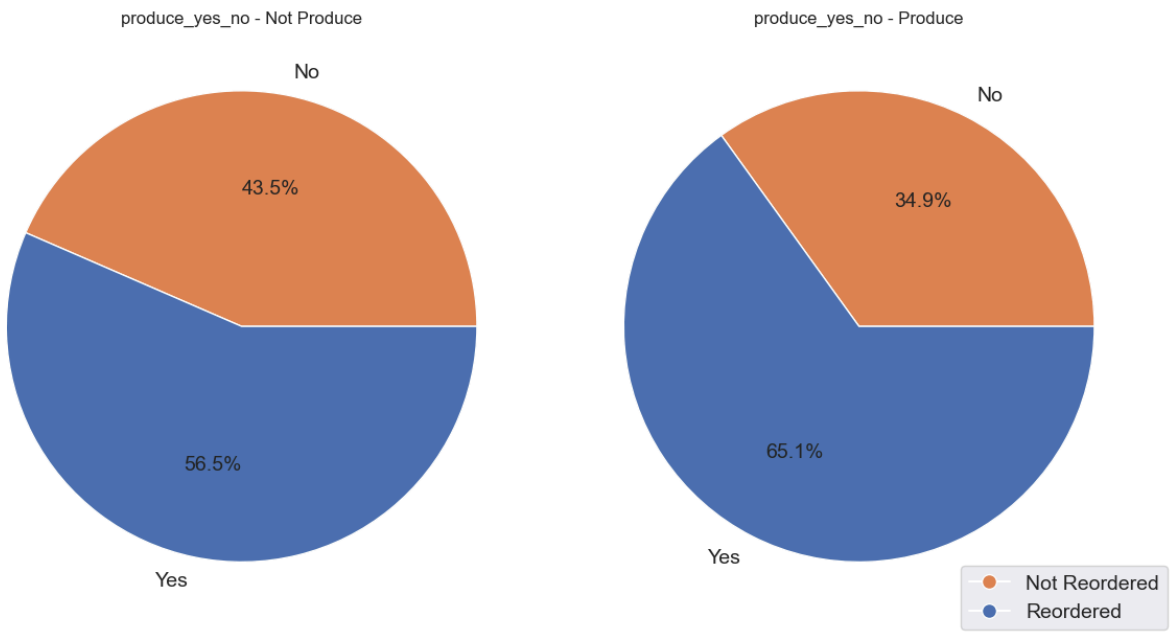
Venn Diagram of Top 5 Departments by Product Offerings and Orders/Sales



I drilled into the top 2 product categories for user orders (dairy/eggs and produce) and noted that Organic products are quite popular as well. Fifteen of the top 20 products ordered on the Instacart platform are of the organic variety and only 1 dairy/eggs product ranks in the top 20, Organic Milk.



Given that the most actively ordered products on the Instacart platform can be categorized into the produce department and are of the organic variety of foods, I continued to dig further for evidence that these factors may be related to whether a customer will reorder a product or not. There appears to be some relationship between these categories and the likelihood of reordering items.



By generating a correlation heatmap, I was able to determine some weak correlations between reordered status and the following categories:

- Items amongst the first ten items added to the cart appears to have a weak positive correlation with whether it is reordered

- Items ordered between 8 and 14 days prior, also appears to have a mild positive correlation with reorder status

Lastly, I also noted a slight negative correlation between purchases of produce and dairy/eggs.

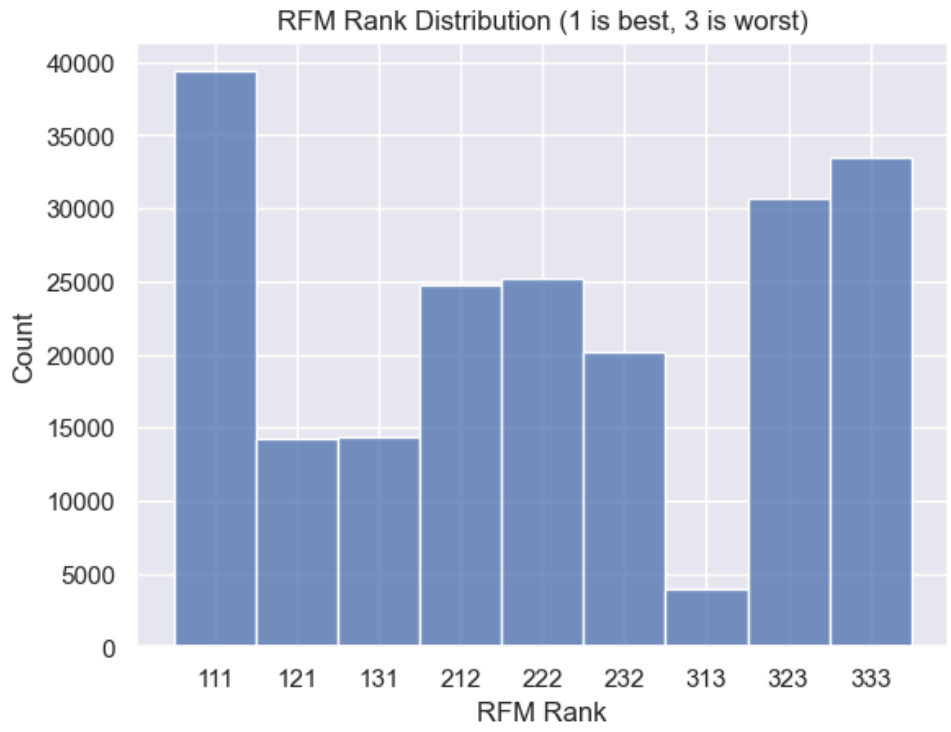
Customer Segmentation with RFM and K-means Clustering

From a marketing perspective, I decided to see if the Instacart user population could be segmented to derive further insights and to perhaps strengthen any predictions for products purchased. This posed quite a challenge, as Instacart did not provide any demographic data for the roughly 200,000 users in this sample dataset. In order to protect the anonymity of the users, even product pricing and sales generated per order were excluded from this dataset. Without the benefit of having demographic data, I decided to leverage the RFM model as a way to segment my users.

RFM is a behavioral consumer segmentation method that leverages three key metrics to categorize customers:

- **Recency** informs us about how recently a customer made their last purchase. A lower recency value typically indicates that the customer is more engaged with our business. Recency was provided by Instacart in the days_since_prior_order in the orders table. Of the 3.4 million records in the orders table, roughly 2 million contained NaN values. The documentation was silent as to what the NaN values might represent. Rather than assume either 0 days or over 30 days, we replaced all NaN values with the median value of 8 days. Lastly, we assigned the max recency value from all of each users' orders placed to represent their recency score.
- **Frequency** measures how often a customer purchases from us within a specific time period. A higher frequency would be considered better. We were able to derive the frequency value for each customer, by counting the number of unique order_ids associated with each user.
- **Monetary Value** allows us to measure the total amount a customer has spent with us over a specific timeframe. As previously noted, Instacart removed all information pertaining to sales amounts or pricing information for the individual products to maintain privacy for their users. We made the assumption that customers who purchased more items, would also therefore have a higher monetary value. We used the total number of items purchased per user as a proxy value for our monetary value.

Based on the count of customers within each segment, it appears that the RFM segment containing the highest rankings for each of the RFM categories (segment =111) is our largest group of customers. However, the second largest group of customers is segment 333, containing our least engaged, least frequent and lowest monetary value.

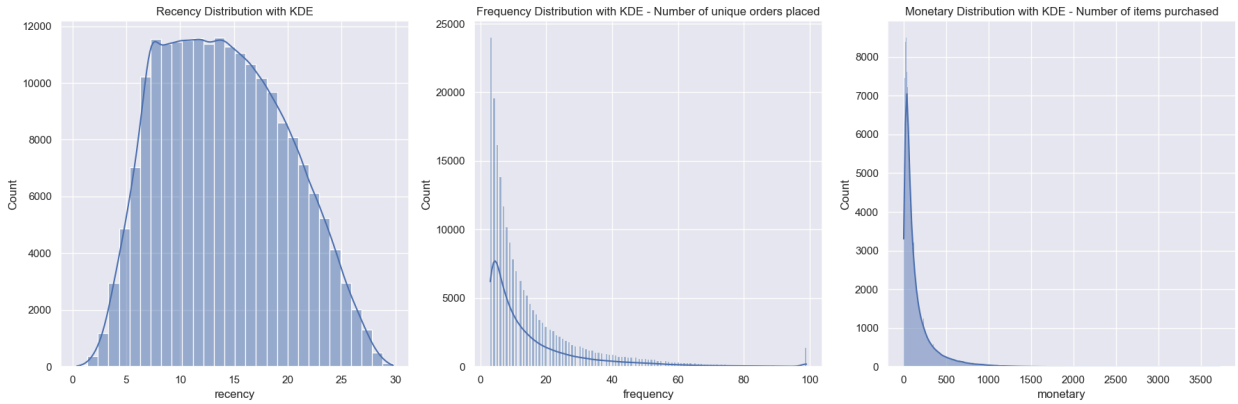


This information could be used to send targeted marketing promotions and suggestions to the Instacart users to perhaps motivate the middle of the road customers to become more engaged and to reduce the chance of churn.

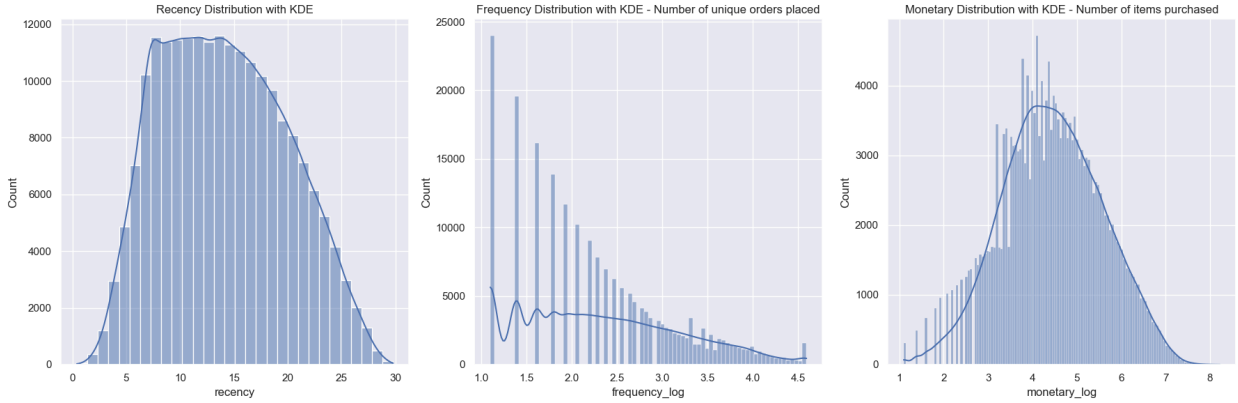
Before utilizing the K-means clustering method, I pre-processed the data by completing the following tasks:

- Review the distribution of our features and resolve any issues with skewness
- Normalize our data
- Determine the ideal number of clusters using an elbow plot

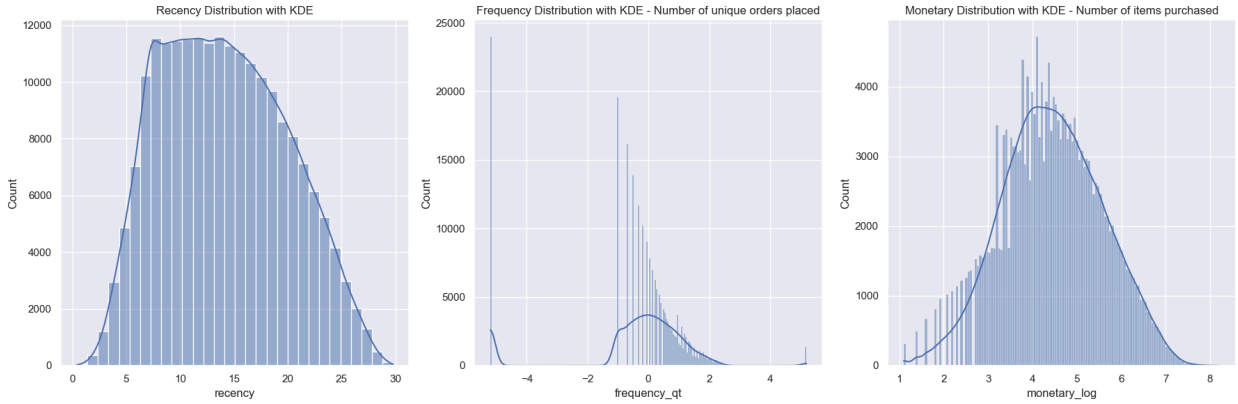
Upon my initial review of the distribution for recency, frequency and monetary value, only recency appears to have a normal distribution. Both frequency and monetary value are skewed to the right.



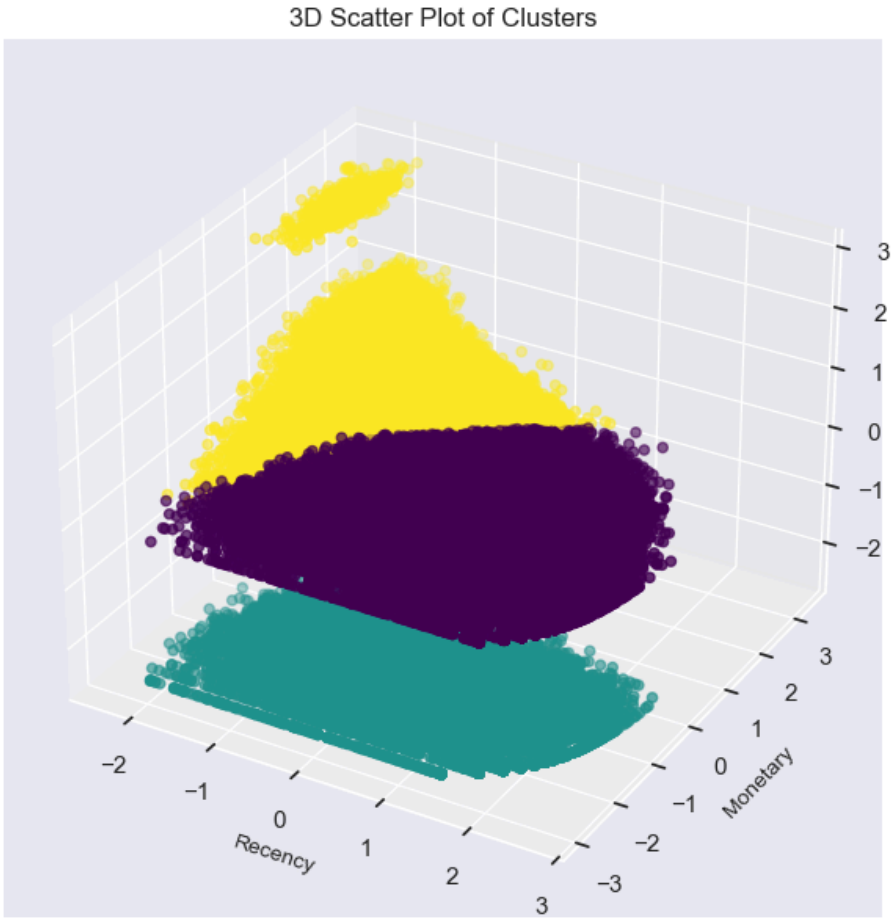
Initially, I attempted to normalize frequency and monetary value using logarithmic transformation as each of the features only contain positive values. This normalized our monetary value data, but still left frequency skewed to the right.



Lastly, we obtained a little normalcy to the frequency feature leveraging quantile transformation.

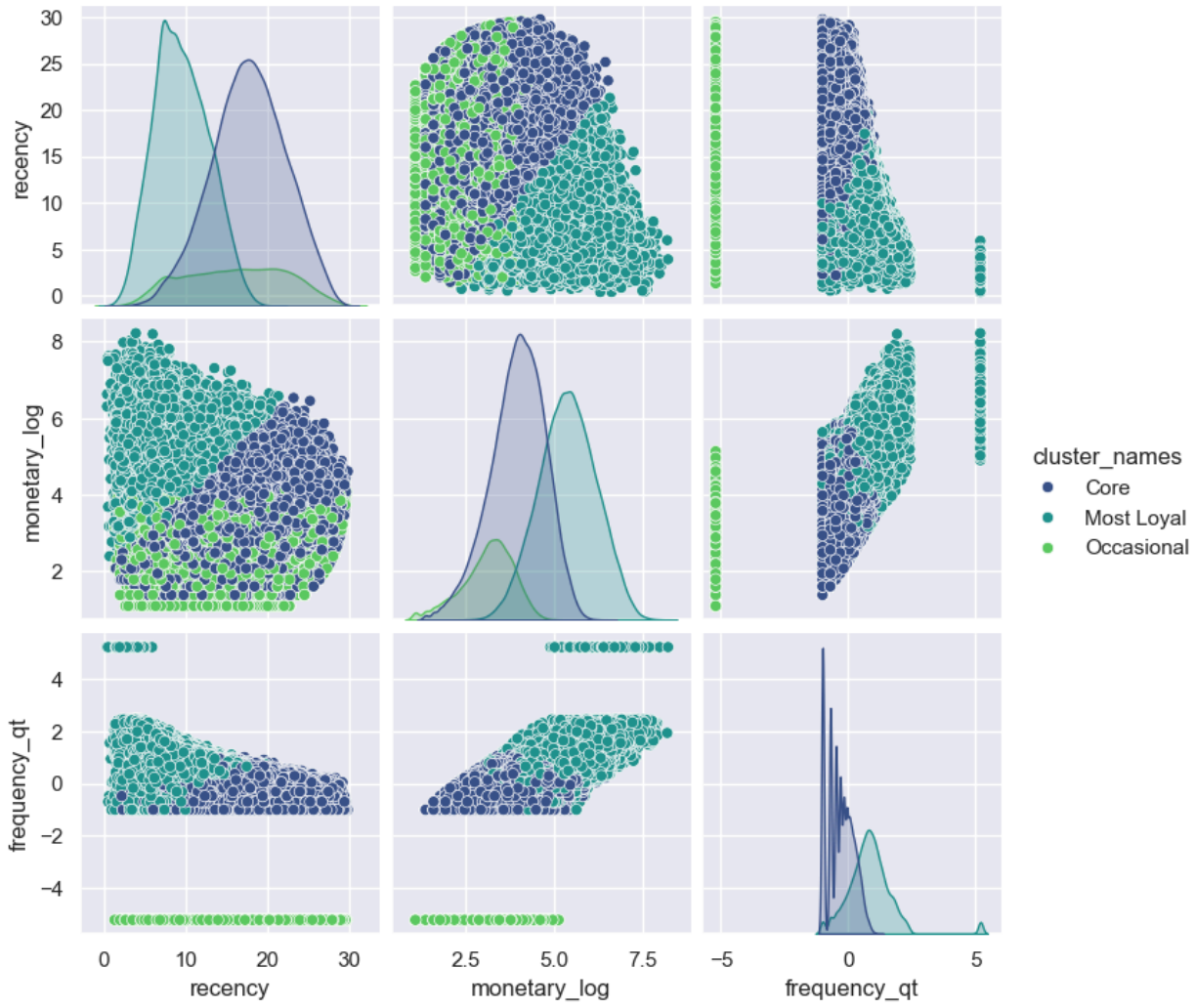


After reducing the skewness of our features, we normalized our data using sklearn's Standard Scaler. We also leveraged an elbow plot to find a good starting point for n_clusters, which was determined to be 3 clusters. Our initial result is below:



Next, I flattened our data using t-SNE to better visualize our clusters.

Lastly, using pairplots patterns emerged. This allowed me to assign characteristics to the 3 clearly identifiable segments. With this information, we could target each of the customer segments with specific promotions and or discounts to potentially shift their purchase behaviors.



Market Basket Analysis - the Challenges

The biggest challenge that I faced with this dataset was the sheer size of the combined orders_products table, which contained over 30 million rows of data. This was a challenge I expected as the Apriori algorithm is known to be computationally expensive. My initial plan to reduce the dataset using my newly identified customer segments, still left me with a large dataset of 25 million rows, even when focussing solely on our most loyal customers. Therefore, I scrapped the idea of using customer segments to reduce my dataset and settled on taking a random sample.

Random sampling added a new challenge for splitting the dataset. I had to split my data between train and test, based on the construct of time. Meaning, that the test set should include the very last order for each user and the training set would be all prior orders for each customer. To accomplish this task, I created a random sample of

Instacart users. I began with 10% of the customer sample, approximately 20,000 customers. The memory challenge still persisted. I also tried sample sizes of 5% and 2.5% of customers. Even shutting down all processes on my machine other than my VS code, was of very little help. All still posed a memory challenge for my computing resources, and I ultimately settled for a customer sample size of 1%.

Even the significantly reduced dataset containing 1% of customers, would still pose a challenge for leveraging the FP Growth model. Therefore, I created the following models: a baseline, a tweaked baseline, an apriori rules based model, and the eclat model.

Market Basket Analysis - the models

The baseline model will build a cart using the avg. cart size per user_id and the popularity of products ordered. We will used the following metrics:

- their average cart size, n (as measured by total number of products purchased / total number of orders)
- all Instacart items will be ranked from most frequently purchased item to least frequently purchased item (rank=1 is most frequent, so lower is better)
- construct the cart to include the Top n Instacart items purchased

This baseline model produced an F1 score of 14.1%.

Next, I tweaked the baseline model to be a bit more personalized specific to each user's purchase history. At its core, this model assumes that users will reorder various products from time to time. Based on order frequency by user, I created a cart based on each user's top products. It used the following metrics:

- their average cart size, n (as measured by total number of products purchased / total number of orders)
- all the individual customer's purchases will be ranked from most frequently purchased item to least frequently purchased item (rank=1 is most frequent, so lower is better)
- build a cart comprised of the top n items from each customer's specific top purchased items

This tweaked baseline model produced an F1 score of 32.7%.

Next, I created an Apriori Rules Association model for predicting future purchases for each user. This method of association is a popular data mining technique and was one of the first used in market basket analysis. It is used to uncover potential relationships between items within a dataset, which meet a minimum support threshold. In this case, the minimum support represents the proportion of the transactions which contain the itemset. This model will generate rules of association that says if item A is in a cart, then item B is also likely to be in that cart. A great example of this is oftentimes when someone purchases a new mobile phone, they are highly likely to also purchase a new mobile phone case as well.

Before creating the association rules, the data must be formatted properly. The orders_products table contains one row per item purchased. Therefore, each customer order initially contained multiple rows of data. In order to use the apriori method, the data must be formatted to contain
A single row per shopping cart, with one product per column. These baskets must also be encoded as boolean type, indicating whether or not the product is included in the basket or not.

After generating the association rules, I was able to define a function to recommend a shopping cart for each user based on the antecedents and consequents within the rules. I leveraged the following information to build a cart:

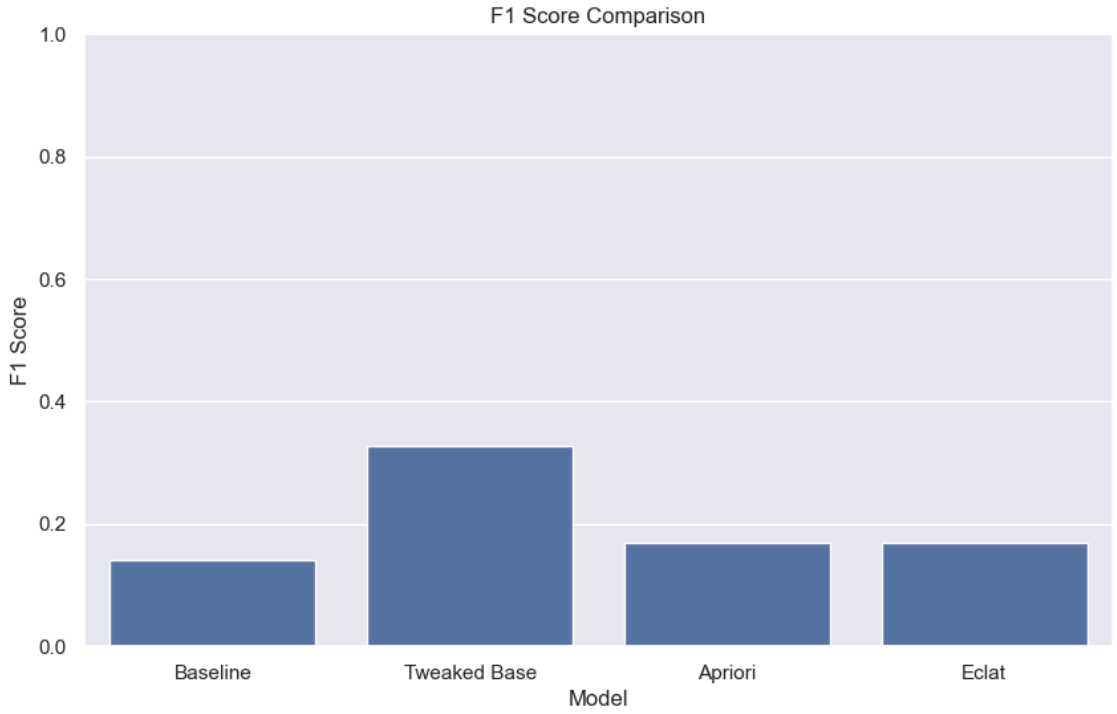
- the average cart size per customer (n), this will tell us how many products should likely be within the customers next order
- Assume that the initial item in a customers basket will be the product that they order most frequently
- add each subsequent item to the cart based on the association rules by identifying the consequent item for the last item added to the cart after each iteration up to n times.

The apriori model produced an F1 score of 16.9%, only slightly better than our baseline model. As the apriori model is based on the shopping habits of all customers as a whole, vs the shopping patterns of the individual customers, it is not surprising that while slightly better than the baseline model, it is not better than the tweaked baseline model.

Lastly, given the computational expense of using the apriori model, I also used the ECLAT algorithm. It is one of the suggested methodologies for larger datasets. It is very similar in nature to the apriori rules association model, except it uses a vertical data format where each item is associated with a list of transactions in which it appears.

The frequent itemsets are found by intersecting the transaction_id's list to create new itemsets.

Its F1 score was essentially the same as the Apriori model, with a score of 16.9%.



Recommendations

In this particular instance, leveraging a simple model that builds a cart based on the frequency of how often a customer purchases an item, produces the highest accuracy score when it comes to predicting what will be in their next shopping cart. This is the model that I would propose we start with, given our CPU constraints. For future enhancements, I would consider leveraging Apache Spark given its scalability when working with larger datasets. This would allow me to utilize larger sample sizes, and potentially to use additional algorithms such as FP-Growth (included within the Spark MLlib library) and XGBoost (using the XGBoost4J-Spark library) in my market basket analysis.

Another recommendation for Instacart would be to increase the visibility of organic produce on the platform. During the EDA process, we noted that customers were more likely to reorder organic produce items vs non-organic items. Instacart may want to ensure that these items are highly visible and available on its platform to drive revenue and to maintain or improve customer satisfaction.

Lastly, Instacart could leverage the customer segmentation data to develop targeted marketing campaigns for its customers. The analysis showed that the largest segment represents its most loyal and engaged customers. However, the second largest segment appeared to be its least engaged or non-active users. For its most loyal customers, this could simply be to offer suggestions or reminders to reorder their frequently purchased items at regular intervals. For the customers who are not as engaged, they may want to offer one-time discounts to come back or create a loyalty program like the Kroger Plus card.