Daniel Heinsch
5-8-21

Olist Case Analysis

Introduction:

Rafael Soto is a lead marketing analyst at the Brazilian e-commerce store, Olist. Over the last 4 years he has helped Olist reach out farther than they ever have before. Due to their fairly large presence in the E-commerce market, Olist has to stay ahead of the competition through advanced marketing and behavioral analytics. Recently, an intern from America has come to Brazil to learn under him, that intern being me. After a few rounds of successful interviews and an extensive sign-up process, I traveled there to broaden my analytical domain.

Headquartered in Curitiba, Olist is a large department store in Brazil. There it connects thousands of small business owners and distributors through its E-Commerce platform. Much like Amazon, one may go onto their app or website and place an order for any item they listed. After it is purchased, the item will be shipped from the store of purchase, to the awaiting customer, after which, they are asked to review the product. For every item that is sold, Olist receives a portion of that revenue, reflecting the general structure of their business model. Just starting in 2015, Olist has already amassed a sizable merchant base with thousands of weekly customer purchases.

The goal of Olist is to expand these small business owners to new customers, something that a small, physical store would never be able to by themselves. Due to Olist's meteoric rise, recent investor support of 5 million dollars, and relatively little competition in Brazil, they are looking to take advantage of their large collection of data and expand on their data analytics division.

The data engineers got back to Rafael, and he sent me nine separate datasets that detail customer habits, their transactions, order details, and more supplementary data that ranged from 2016 to 2018, covering over 735 days.

Rafael presents a few key questions to be answered through these datasets. **1)** How can we get a deeper understanding of Olist customers? **2)** With our understanding of them, how can we use this to continue Olist's expansion? **3)** What are some other key aspects of the Olist platform that should be analyzed, and how would these discoveries benefit Olist? **4)** Develop a short positioning statement for each proposition and present it / them to Rafael and his team.

Rafael explains that this project is an important part of my internship, and to do well would reflect a promising future for my position at Olist.

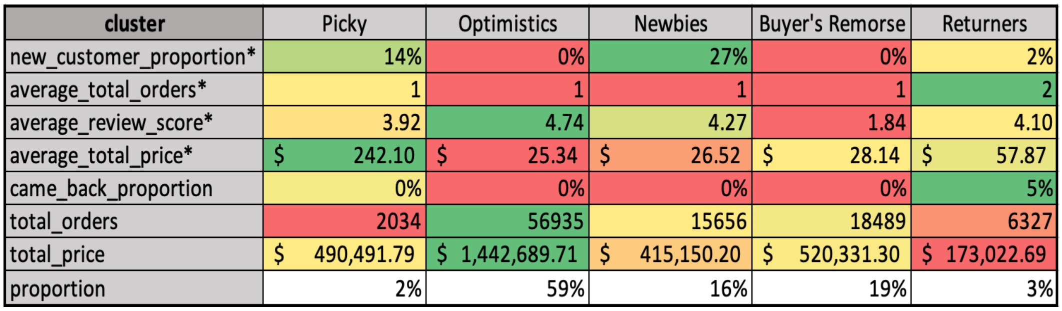
Customer Report:
I decided that my angle of attack would be to identify and group the customer base in some way, through this we may target the customers that return the best results and allowing Olist to expand through an increase in revenue while decreasing extra costs. My first step in doing this was to clean the data and collect some general statistics on the customer's behavior. Immediately, I was shocked to find that most customers never actually return, with only 3.12% of all customers making more than one purchase. And even though the average product rating was above 4 out of 5, this clearly didn't matter. Additionally, those 97% of customers who never returned made up 94% of all the revenue gained from Olist and the sellers. Not only would it be harder to identify these customers, creating an effective targeting position may also prove to be challenging. This added an extra level of difficulty that I did not anticipate. It isn't easy to target a customer that only shops once. Regardless of the obstacles ahead, I pressed on.

With such a high proportion of revenue that comes from customers that do not return, it would be wise to either convince customers to stay or encourage them to spend more from their one order than what they initially intended. I decided that the latter would be a good option, which can be achieved through collaborative filtering.

I needed to start moving variables into the sparse Customers dataset. These variables would need to be only the most important and insightful metrics available. Olist's goal is to expand on their growing business, and this is primarily done through monetary gain. I figured that measuring a customer's sentiment, their total price contribution, and their number of orders stand as the most effective metrics. Additionally, determining their stability and their recency of joining are promising candidates.

With all this data surrounding each customer, I thought of different ways that one could group them. The best method that came to mind was through clustering. If I could take the customers and the most important metrics that I created and pinpoint the cluster(s) that benefit Olist the most. After picking the best segment, I would develop a customer positioning statement for Rafael.

I went ahead with using the 5 variables that I created. I ran multiple clustering algorithms and picked the optimal cluster count. After segmenting the customers into 5 clusters, I aggregated the variables that were included in the algorithm while also adding a few more. I did this to measure the Clusters' reflection of the factors that did not play a part in the algorithm. With a little bit of formatting and color coding, I sat down to look at the results [Figure 1].



cluster	Picky	Optimistics	Newbies	Buyer's Remorse	Returners
new_customer_proportion*	14%	0%	27%	0%	2%
average_total_orders*	1	1	1	1	2
average_review_score*	3.92	4.74	4.27	1.84	4.10
average_total_price*	$ 242.10	$ 25.34	$ 26.52	$ 28.14	$ 57.87
came_back_proportion	0%	0%	0%	0%	5%
total_orders	2034	56935	15656	18489	6327
total_price	$ 490,491.79	$ 1,442,689.71	$ 415,150.20	$ 520,331.30	$ 173,022.69
proportion	2%	59%	16%	19%	3%

Figure 1: Customer Clusters

Cluster 0 [The Picky]

This cluster contains those who spend, on average, much more than any other cluster. Making up only 2% of all the customers, they buy very few, but expensive items. Their high expenses allow them to surpass cluster 2, which is nearly eight times larger than them. Even with the promising returns, they do not seem to be a trustworthy cluster to rely on.

Cluster 1 [The Optimistics]

With an astounding 59% of the total customer base, these customers are content with buying a cheap item they found online. They clearly know what they are getting, and leave amazing reviews, with an average of 4.7 out of 5 from a total of around 57,000 customers. Although these people loved their product, they did not find any reason to return to the platform, being one of the three clusters that have absolutely no returners. Although they contribute the most to the overall revenue, they spend the least on average.

Cluster 2 [The Newbies]

The newbies hold the largest percentage of new customers. Much like its peers, the Newbies care very little about sticking around, and spend the second lowest in total and on average. This Cluster seems to be very content with their purchase, however, this does not seem to affect their buying behavior.

Cluster 3 [The Buyer's Remorse]

The Buyer's Remorse cluster can be described in its name. With easily the lowest average reviews, they clearly hated the products that they purchased. Their unwillingness to return isn't all that noticeable as even the highest of raters refuse to come back, but they clearly regret their purchases.

Cluster 4 [The Returners]

With customers who have returned making up a whopping 5% of the total cluster's population, this cluster's 'came back' count is 15 times higher than the next highest, The Picky cluster. Except for 288 individuals, all remaining 5,189 returning customers fall under the Returners. This Cluster has the second highest average total price paid, but their total price is smallest when compared to other clusters.

Additionally, they only make up 3% of the total customer population and they unfortunately do not make up for it in any way.

None of these groups show much loyalty, and the cluster with the highest rate of returning is not only small, but also the least appealing in terms of revenue. Instead, it is more important to try and target a cluster that makes up a large portion of the total revenue while also encouraging them to increase the price of their one order. This could be achieved through The Optimistics as they easily make up the largest share of total revenue along with their very high sentiment towards the products they have purchased. Additionally, their average order cost is the lowest, but if that were to be increased, then it would result in huge returns. Using customer reviews in a collaborative filtering system would likely be the most effective way to encourage extra purchases per order, and thus increasing the Optimistic's total average price.

Due to the lack of customer reviews in the customer to product pivot table, and the fact that the segmentations result in an even sparser table, segmented collaborative filtering would not be possible. So instead, I decided to run a general recommendation system that used all customers. As stated above, the customer's review score would represent the sentiment and correlation between every product category through an item-based collaborative filtering system.

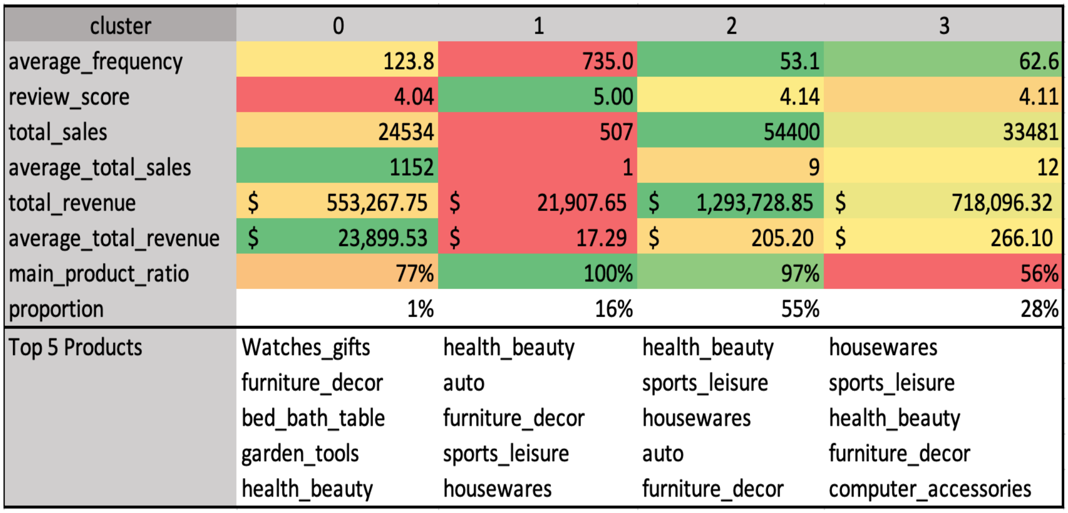
With this completed, and in addition to customer segmentation and targeting, I felt satisfied with answering Rafael's first two questions. However, I needed to step back a little, up until this point my analysis has focused almost entirely on customers. Rafael recommended that I look at the other datasets, so I turned my attention to the merchant base on the Olist app.

Merchant Report:

The pursuit of customer analysis is only but one side of Olist's clients. Not only do customers use Olist to buy, but Sellers use it to sell. This is the client base that actually pays Olist to use their platform and increasing the total amount of suppliers should prove to be wise.

Rafael was pleased to find that the average total sales of a given seller was around 36 items, with only 16% of all merchants only ever selling one product. This would allow for more insightful interactions and comparisons to be done between sellers. These Sellers, on average, sell one item every 178 days, where the highest performers sell one or more items daily. Getting the frequency of one's transactions reflect the amount of traffic brought in by a given Seller. Their average total revenue is about 4,399 BRL -or around 842 US dollars-, which paired with an average total sale of around 36 items, shows that Olist's platform centers around fairly small businesses.

Similar to the customers data, I planned to segment the suppliers into clusters. With this, I would be able to identify the sellers that report the highest returns and best overall metrics. I was very satisfied with the metrics used to cluster the customers, so similar variables will be used for seller segmentation.



cluster	0	1	2	3
average_frequency	123.8	735.0	53.1	62.6
review_score	4.04	5.00	4.14	4.11
total_sales	24534	507	54400	33481
average_total_sales	1152	1	9	12
total_revenue	$ 553,267.75	$ 21,907.65	$ 1,293,728.85	$ 718,096.32
average_total_revenue	$ 23,899.53	$ 17.29	$ 205.20	$ 266.10
main_product_ratio	77%	100%	97%	56%
proportion	1%	16%	55%	28%
Top 5 Products	Watches_gifts furniture_decor bed_bath_table garden_tools health_beauty	health_beauty auto furniture_decor sports_leisure housewares	health_beauty sports_leisure housewares auto furniture_decor	housewares sports_leisure health_beauty furniture_decor computer_accessories

Figure 2: Seller Clusters

Cluster 0 [The Goliaths]
The Goliath cluster makes up for its name not in proportion, but in total sales and average total revenue. Making up only 1% of the total seller base, they make a little under half of the highest grossing cluster. These businesses are far and away the biggest entities on the platform and provide a stable flow of revenue.

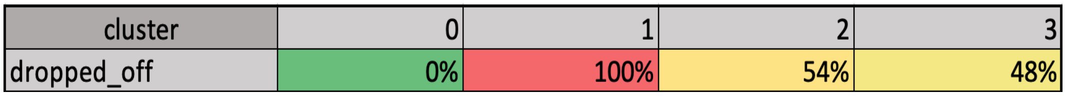
Cluster 1 [The Davids]
The davids are the smallest of small businesses on Olist. They are likely hobbyists or small mom and pop shops that tried out the E-commerce method, only to ditch it for traditional, physical, selling. This cluster encompasses all or nearly all of the sellers that have only sold one item, which is reflected in their low revenue and sales. Additionally, David is the second smallest cluster with around 16% total proportion. On the bright side, their average product reviews are 5 out of 5, though this unfortunately does not outweigh David’s other negative aspects.

Cluster 2 [The Foundation]
The foundation cluster makes up the bedrock of the platform. With the highest overall total revenue and largest cluster proportion, these are the types that allow Olist to flourish. These types are heavily specialized, even for their massive proportions, averaging at a 97% main product ratio.

Cluster 3 [The Mundane]
The Mundane cluster is a non-specialized, larger cluster. They are not particularly notable as their metrics do not place first in any category. Although not special, the Mundane are also not the worst in any category, making them average in every aspect.

Unfortunately for Daivd, Goliath wins this time. Minimizing the David cluster would be ideal, while propping up notable clusters like Goliath and Foundation. Focusing directly on Goliath may seem like a tempting target, however, it is important to note that one shouldn't focus too heavily on a few large clients. For one reason or another, they may decide to leave the platform, and to rely on their loyalty is precarious. On the other hand, increasing the Goliath by one or two percent of the overall seller base could lead to an immediate surge in revenue. Additionally, these Goliath businesses are likely to be more stable and would not be as likely to leave Olist in case of some economic downturn.

My initial reaction was to pick Foundation for their large size and high total revenue, but first I wanted to measure Goliaths' 'loyalty', I created one final variable that would check to see if a Seller had 'dropped off' where it checks if the days between the product last sold and now was greater than the average sale frequency.



cluster	0	1	2	3
dropped_off	0%	100%	54%	48%

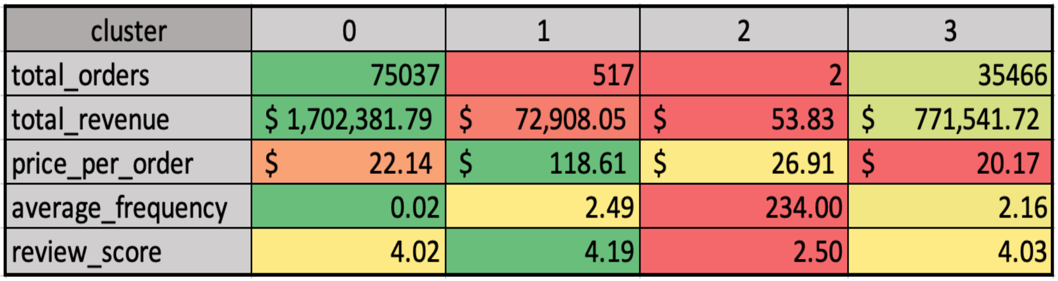
Figure 3: Seller Drop Off

As it turned out, Goliath measured to be the most loyal, with all 22 of the businesses keeping a steady rate of sales and not slowing down. This is amazing when compared to Foundation, where nearly half of their seller base has performed less than expected. To conclude the merchant clustering portion, the Goliath cluster should be the target for the Seller side of the Olist clientbase.

Wrapping back to Rafael's initial questions, the merchant cluster analysis was another aspect of the Olist datasets that, when analyzed, could lead to the benefit of Olist. With a higher amount of these 'larger' small businesses, it would mean a massive jump in income for each new client.

Moving forward, the analysis of the merchant base revealed that much of their top products repeat. This would indicate that certain products are much more lucrative than others. If a product analysis is performed, then it may be possible to compare its results with the seller clusters. This would allow Olist to not only target the best sellers, but it would also allow them to figure out which businesses to contact based on their product selections.

I collected a few variables for the products dataset. These variables are very similar to both the sellers and customers, but still contain the most important data for the questions at hand. We are looking for product clusters that show high total revenue, high total orders, and frequent transactions. From the earlier analyses, it seems as though the average review score is not necessarily an indicator for quality clusters, and we may choose clusters with a relatively low rating. It is important to note that this 'relatively low' rating would likely score at around 4 out of 5 and still reflects an overall positive sentiment. Once again I ran the cluster analysis, this time on the products dataset.



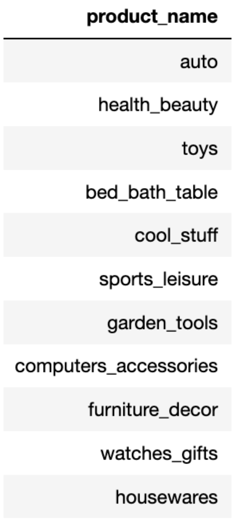
cluster	0	1	2	3
total_orders	75037	517	2	35466
total_revenue	$ 1,702,381.79	$ 72,908.05	$ 53.83	$ 771,541.72
price_per_order	$ 22.14	$ 118.61	$ 26.91	$ 20.17
average_frequency	0.02	2.49	234.00	2.16
review_score	4.02	4.19	2.50	4.03

Figure 4: Product Clusters

There is not much that is needed to be said about the product clusters. After a quick glance, I already knew which one to pick. Cluster 0 was the most popular, frequent, and highest grossing cluster. Although lacking in the highest review score, it still boasts an average of around 4. Cluster 3 is another fine contender, but still lacked even half of the total orders and had a smaller average price per order. Cluster 2 was not even worth mentioning, where it contains a mesly 2 total orders and most likely should not have been considered as its own cluster. Cluster 1 is less appealing than Cluster 3, and although it has high reviews and a massive price per order, it does not compete with the returns provided by Clusters 0 or 2.

When looking back at the customer clusters, the Optimistic Cluster was chosen, they had a low average total price (same as 'average price per order' as none of them returned). This matches with product cluster 0 as its average price per order is also very low. This is likely due to customers in the Optimistics making up a large portion of product cluster 0.

I decided to move forward with Cluster 0 and looked at which products made it up [Figure 5]. The eleven product categories serve as the main product attractions to Olist and targeting the businesses that supply them will come with a great advantage. The 5 highest performing products in each merchant cluster also reflect the success of cluster 0's products, where every top product is listed within Cluster 0.



product_name
auto
health_beauty
toys
bed_bath_table
cool_stuff
sports_leisure
garden_tools
computers_accessories
furniture_decor
watches_gifts
housewares

Figure 5: Cluster 0 Products

Positioning Statement: Customers

As Olist expands ever further across Brazil, the analytics needs to keep up with the data that is collected. One such aspect of Olist data concerns the customers. Being able to properly identify, understand, and target the most important customers will continue to feed back into Olist's revenue stream. One of the largest issues that Olist is facing, is that customers are not returning. In order to combat this, there must be an item recommendation system in place to encourage these customers to purchase more per order. Other methods to keep customers from leaving are also an option, however this would require much more time, knowhow, and man power to solve. Additionally, the customer segment, Optimistic, should be the target customer base in which Olist targets. This cluster makes up the largest share of all revenue while also having the lowest average order price. Increasing this average by a little bit through item-based collaborative filtering will result in a massive increase in revenue.

Positioning Statement: Sellers and Products

Olist's merchant sector acts as the sole source of revenue for Olist's E-commerce platform. Although customers indirectly pay Olist, the sellers personally pay them. The type of merchant ranges from a one-and-done to a larger small business and everything in between. Olist now has to decide which seller type provides the most benefit for Olist's current and future positions. Through the clustering analysis of said sellers, it is clear that the cluster, Goliath, would provide the most benefit to focus on. Although Goliath is currently less profitable than other clusters, just a slight increase in its size would lead to an increase in income multiple times as large as any other cluster. Additionally, the businesses under Goliath are stable entities that return consistent sales reports.

While analyzing the highest performing merchant clusters, a clustering for products was also performed. The products that fell under the chosen cluster, Zero, are the same products listed in each merchant cluster's best products. Targeting the larger small businesses and reaching out to other businesses that specialize in products under cluster 0 would lead to the highest increase in Olist performance.

Conclusion:

This case has been a great learning experience for the whole business analytics workflow, from business understanding to evaluation, and everything in between.

Due to the customer return rate being so low, it was difficult to think of a practical customer target that a business could use. Although I was faced with this obstacle, the customer cluster decisions left me fairly satisfied, and instead used the customer return problem as my main reason for applying the recommendation system.

The item based collaborative filtering would not work with the clustered customers. The reason why this occurred was because of the low customer return rate, which left the correlation matrix to be very sparse and sensitive, where it alphabetically reported the listed items. Because of this, I decided that a general recommendation system would work much better. Finally, I would have like to have done more analysis with geo data, but because of time constraints and the geolocation dataset's anonymity, this proved too difficult.

Appendix:

A) Data Descriptions: 9 datasets

The original collection of Olist data was broken down into 9 separate datasets. Each set is relatively clean where any variable with a notable number of null records were never important in the first place. Each dataset also contained one or more '[variable]_id' that allowed connections between other datasets.

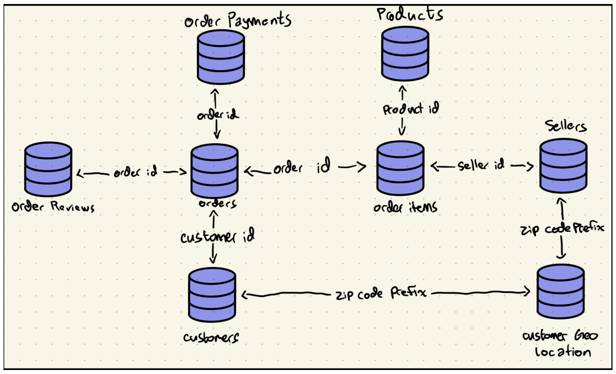


Figure 6: Datasets & Connections

olist_customers_dataset [customers]:
The customers dataset contains basic geolocation information for each unique customer. Other than these variables and two customer ID keys, this dataset contains little personal information on them.

olist_geolocation_dataset [geo]:
The geo locations dataset contains anonymized -but specific- location data for both sellers and customers.

olist_order_items_dataset [items]:
Made up of mostly ID variables, this dataset contains some information on the prices, and dates for each product sold in all orders.

olist_order_payments_dataset [payments]:
This dataset contains payment information for every unique order.

olist_order_reviews _dataset [reviews]:
This dataset contains review information for every unique order

olist_orders_dataset [orders]:
This dataset contains multiple date and status information for every unique order.

olist_products_dataset [products]:
This dataset contains anonymized information for each unique product.

olist_sellers_dataset [sellers]:
This dataset contains basic geolocation information for each unique seller.

product_category_name_translations [translations]:
The dataset contains product categories in both Portuguese and English

These are the descriptions of the cleaned datasets (including clusters).

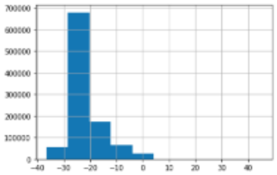
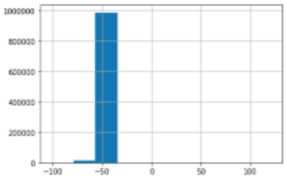
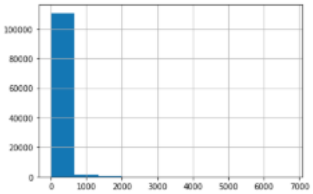
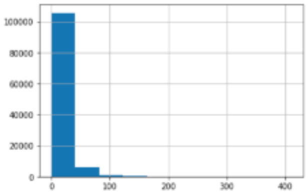
Cleaned Datasets:

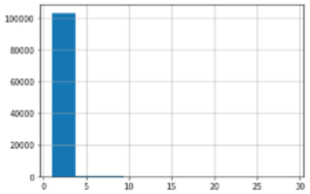
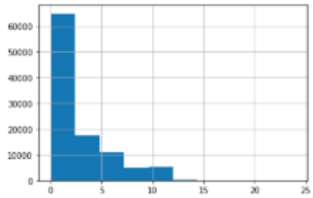
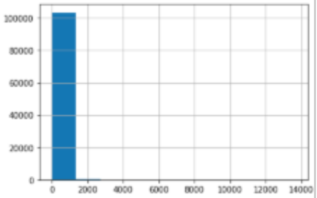
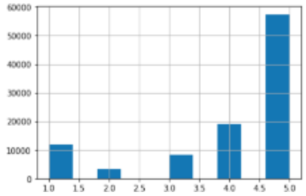
CustomerCleaningFull:
The dataset contains product categories in both Portuguese and English

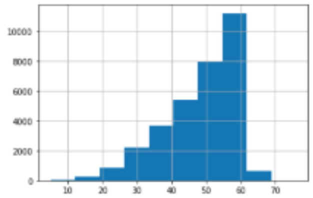
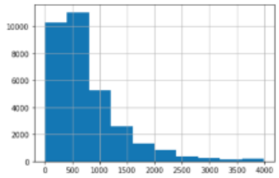
B) Data Dictionary

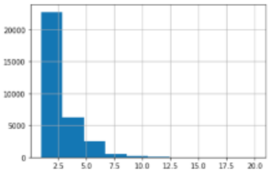
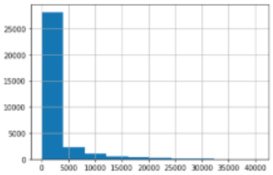
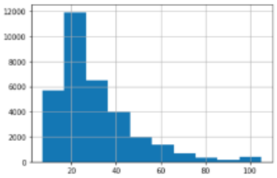
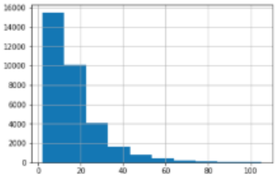
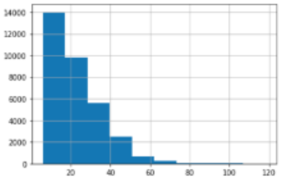
Olist Original Data

variable name	description	type	min	max	categories	Distribution
olist_customers_dataset						
customer_id	key to the orders dataset. Each order has a unique customer_id.	Char			99441	
customer_unique_id	unique identifier of a customer.	Char			96096	
customer_zip_code_prefix	first five digits of customer zip code	Char			14994	
customer_city	Customer city name	Char			4119	
customer_state	Customer state name	Char			27	
olist_geolocations_dataset						
geolocation_zip_code_prefix	first 5 digits of zip code	Char			19015	

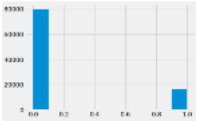
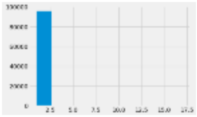
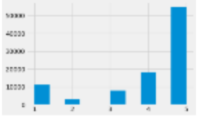
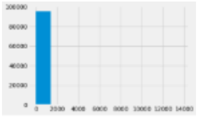
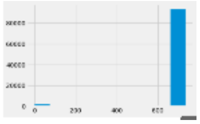
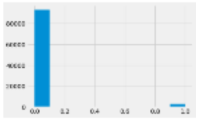
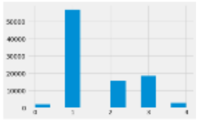
geolocation_lat	latitude	Geo				
geolocation_lng	longitude	Geo				
geolocation_city	City name	Char			8011	
geolocation_state	State Name	Char			27	
olist_order_items_dataset						
order_id	Order Identifier	Char			98666	
order_item_id	sequential number identifying number of items included in the same order.	Char			21	
product_id	Product Unique ID	Char			32951	
seller_id	Seller Unique ID	Char			3095	
shipping_limit_date	Shows the seller shipping limit date for handling the order over to the logistic partner.	Num	9/19/2016	4/9/2020		
price	Given item Price	Num	$0.85	$6,735.00		
freight_value	item freight value item (if an order has more than one item the freight value is splitted between items)	Num	$0.00	$409.68		

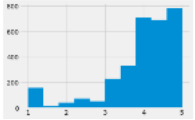
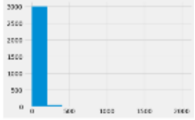
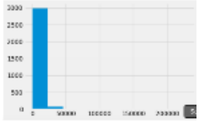
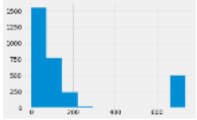
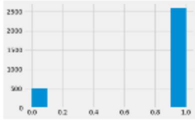
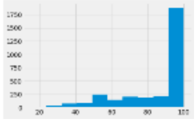
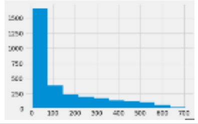
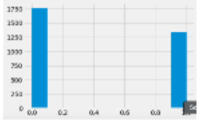
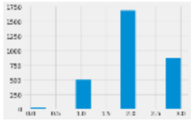
olist_order_payments_dataset						
order_id	Order Identifier	Char				
payment_sequential	A customer may pay an order with more than one payment method. If he does so, a sequence will be created to accommodate all payments.	Num	1	29		
payment_type	method of payment chosen by the customer.	Char			5	
payment_installments	number of installments chosen by the customer.	Num	0	24		
payment_value	Transaction value	Num	0	13,664		
olist_order_reviews_dataset						
review_id	Review id	Char			100000	
order_id	Unique Id of the Order	Char			100000	
review_score	Given score from the customer	Num	1	5		
review_comment_title	The title of the comment	Char			100000	
review_comment_message	Comment message from the review left by the customer, in Portuguese.	Char			100000	
review_creation_date	Shows the date in which the satisfaction survey was sent to the customer.	Num	10/2/2016	8/31/2018		

review_answer_timestamp	Shows satisfaction survey answer timestamp.	Num	10/7/2016	10/29/2018		
olist_orders_dataset						
order_id	unique identifier of the order.	Char			99441	
customer_id	key to the customer dataset. Each order has a unique customer_id.	Char			99441	
order_status	Reference to the order status (delivered, shipped, etc).	Char			8	
order_purchase_timestamp	Shows the purchase timestamp.	Num	9/4/2016	10/17/2018		
order_approved_at	Shows the payment approval timestamp.	Num	9/15/2016	9/3/2018		
order_delivered_carrier_date	Shows the order posting timestamp. When it was handled to the logistic partner.	Num	10/8/2016	9/11/2018		
order_delivered_customer_date	Shows the actual order delivery date to the customer.	Num	10/11/2016	10/17/2018		
order_estimated_delivery_date	Shows the estimated delivery date that was informed to customer at the purchase moment.	Num	9/30/2016	11/12/2018		
olist_products_dataset						
product_id	unique product identifier	Char			32951	
product_category_name	root category of product, in Portuguese.	Char			74	
product_name_length	number of characters extracted from the product name.	Num	5	76		
product_description_length	number of characters extracted from the product description.	Num	4	3992		

product_photos_qty	number of product published photos	Num	1	20		
product_weight_g	product weight measured in grams.	Num	0	40425		
product_length_cm	product length measured in centimeters.	Num	7	105		
product_height_cm	product height measured in centimeters.	Num	2	105		
product_width_cm	product width measured in centimeters.	Num	6	118		
olist_sellers_dataset						
seller_id	seller unique identifier	Char			3095	
seller_zip_code_prefix	first 5 digits of seller zip code	Char			2246	
seller_city	seller city name	Char			611	
seller_state	Seller state name	Char			23	
olist_translations_dataset						
product_category_name	category name in Portuguese	Char			71	
product_category_name_english	category name in English	Char			71	

Cleaned Data

variable name	description	data type	max	min	categories	distribution
Customers						
customer_unique_id	key to the orders dataset. Each order has a unique customer_id.	Char				
customer_id	unique identifier of a customer.	Char				
customer_zip_code_prefix	first five digits of customer zip code	Char				
customer_city	Customer city name	Char				
customer_state	Customer state name	Char				
first_order	Timestamp of a customer's first order	Time	9/4/2016	10/17/2018		
new_customer	Indicates whether the customer's first purchase was within 90 days of the latest date	Bool				
total_orders	Total number orders for a given customer	Num	1	17		
review_score	Average of all reviews given by a customer	Num	1	5		
total_price	Total price paid for all orders for a given customer	Num	$0.00	$13,664.08		
favorite_product	The highest occuring product that a customer purchased	Char				
average_frequency	The average difference of days between product purchases for a customer	Num	0	735		
came_back	Indicates whether a customer has purchased more than once	Bool				
cluster	The cluster number for a given Customer	Num				
Sellers						
seller_id	Unique Seller ID	Char			3095	
seller_zip_code_prefix	First 5 numbers of their zip code	Char			2246	

seller_city	The city that the seller lives in	Char			611	
seller_state	The state that the seller lives in	Char			23	
review_score	The average review score for a particular Seller	Num	1	5		
total_sales	The total amount of orders that contained a Seller's item	Num	1	2036		
total_revenue	the total amount of revenue generated from a Seller	Num	$3.50	$229,472.63		
favorite_product	The most sold product category from a Seller	Char			68	
average_frequency	The average distance of days between product sales	Num	0	735		
multiple_sales	Indicates whether a Seller has sold more than one item	Bool				
main_product_ratio	The ratio of a Seller's favorite product with total number of products sold	Num	15.38%	100.00%		
days_since_last	Number of Days since last Sale	Num	6	706		
dropped_off	Indicates whether a Seller has stopped selling according to their average frequency	Bool				
cluster	The cluster number for a given Seller	Num				

C) Clean Datasets Wrangling Process

Python libraries used:

Pandas **Provided the core structure of all dataframes. Pandas boasts a large number of methods that allow ease of data manipulation and iteration through both columns and rows.**

E.X: using .join() method to combine two datasets based on a given key

Datetime **Compatible with pandas, this core library allowed arithmetic to be used on datatime objects.**

E.X: creating the average_frequency variable through subtracting all subsequent dates for a unique customer / seller, and then extracting the equivalent number of days from this.

Sklearn **One of the most popular machine learning and data science libraries, this package was used for the standardization and clustering (kmeans) for all three datasets of interest.**

E.X: iterating the clustering of an array of the scaled customer's features with an increasing K, where the sum of squares error was appended to an SSE array.

Matplotlib **Another popular package, this library supports the creation of visualizations in an easy way.**

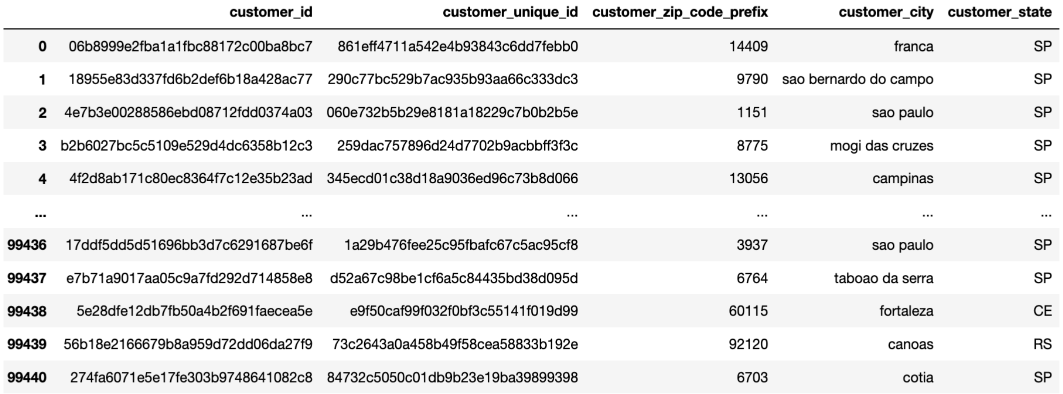
E.X: created an elbow chart for all 10 SSE customer clustering values.

Cleaning Code		
IPYNB	**PDF**	**Description**
CustomersCleaningFull.ipynb	CustomerCleaningFull.pdf	These notebooks step through the process of cleaning and creating the variables use for each analysis.
SellersCleaningFull.ipynb	SellersCleaningFull.pdf	
ProductsCleaningFull.ipynb	ProductCleaningFull.pdf	
Resulting CSV's		
CSV		**Description**
customers_c.csv		Clean customers-based dataset with the added variables.

sellers_c.csv	Clean sellers-based dataset with the added variables.
products_c.csv	Clean products-based dataset with the added variables.
cus_items.csv	This dataset was created within the Sellers cleaning code. This is just a simple dataset that contains customer_unique_id, product_name, and review_score.

- *Full step by step process of cleaning and wrangling is documented in the Code*

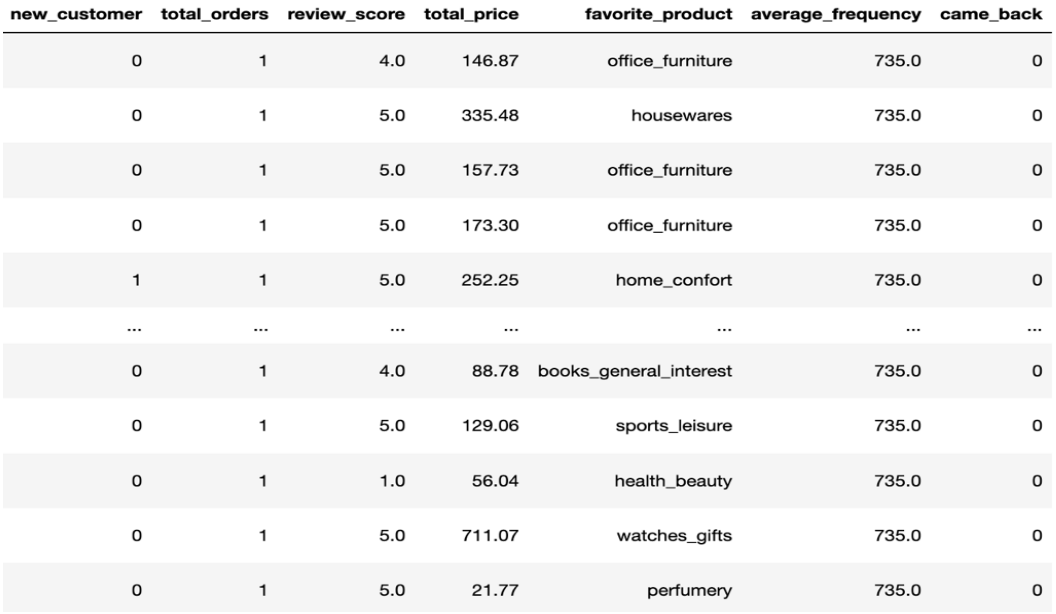
I knew right away that I wanted to segment these Customers into more homogenous groups. To do this, I wrote down every variable held in each dataset (see dictionary) and highlighted the features that would be the most beneficial for Olist's in terms of growth and revenue. Olist, like any other business, wants to make money, and a Customer's total price paid, their total order count, and their average review score to be ideal choices. Starting with these three variables, I began with the customers dataset. With only a few initial features in it, I needed to collect the three variables from the orders dataset. Both total price and total orders could be calculated in there, but I had to access the reviews dataset to calculate average review score.



	customer_id	customer_unique_id	customer_zip_code_prefix	customer_city	customer_state
0	06b8999e2fba1a1fbc88172c00ba8bc7	861eff4711a542e4b93843c6dd7febb0	14409	franca	SP
1	18955e83d337fd6b2def6b18a428ac77	290c77bc529b7ac935b93aa66c333dc3	9790	sao bernardo do campo	SP
2	4e7b3e00288586ebd08712fdd0374a03	060e732b5b29e8181a18229c7b0b2b5e	1151	sao paulo	SP
3	b2b6027bc5c5109e529d4dc6358b12c3	259dac757896d24d7702b9acbbff3f3c	8775	mogi das cruzes	SP
4	4f2d8ab171c80ec8364f7c12e35b23ad	345ecd01c38d18a9036ed96c73b8d066	13056	campinas	SP
...	...	...	...	...	...
99436	17ddf5dd5d51696bb3d7c6291687be6f	1a29b476fee25c95fbafc67c5ac95cf8	3937	sao paulo	SP
99437	e7b71a9017aa05c9a7fd292d714858e8	d52a67c98be1cf6a5c84435bd38d095d	6764	taboao da serra	SP
99438	5e28dfe12db7fb50a4b2f691faecea5e	e9f50caf99f032f0bf3c55141f019d99	60115	fortaleza	CE
99439	56b18e2166679b8a959d72dd06da27f9	73c2643a0a458b49f58cea58833b192e	92120	canoas	RS
99440	274fa6071e5e17fe303b9748641082c8	84732c5050c01db9b23e19ba39899398	6703	cotia	SP

Figure 7: Initial Customer Dataset

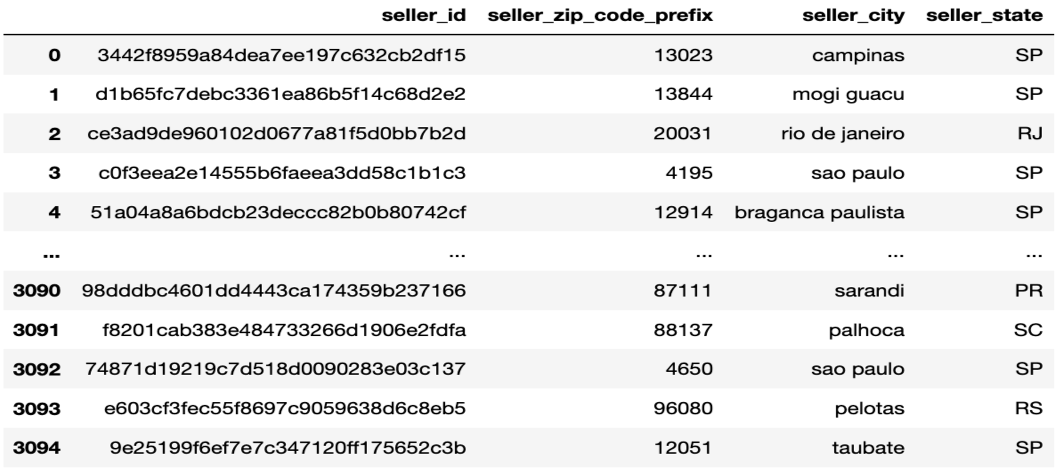
Calculating and creating these three variables was a fairly easy task, and I thought about my next course of action. I figured measuring a customer's behavior would be another solid choice, so I decided on three more variables to create; new customer, favorite product, and average frequency. These features could help with the lack of customer Who data and allow for more accurate targeting. Now with all 6 of these variables stored inside of the customers dataset [Figure 8], I felt more confident in my customer-based dataset. After I encountered the customer return problem, I created a 'came back' variable that check whether a customer has made more than one purchase so that I could monitor the issue in the future clusters.



new_customer	total_orders	review_score	total_price	favorite_product	average_frequency	came_back
0	1	4.0	146.87	office_furniture	735.0	0
0	1	5.0	335.48	housewares	735.0	0
0	1	5.0	157.73	office_furniture	735.0	0
0	1	5.0	173.30	office_furniture	735.0	0
1	1	5.0	252.25	home_confort	735.0	0
...	...	...	...	...	...	...
0	1	4.0	88.78	books_general_interest	735.0	0
0	1	5.0	129.06	sports_leisure	735.0	0
0	1	1.0	56.04	health_beauty	735.0	0
0	1	5.0	711.07	watches_gifts	735.0	0
0	1	5.0	21.77	perfumery	735.0	0

Figure 8: Cleaned Customer Dataset

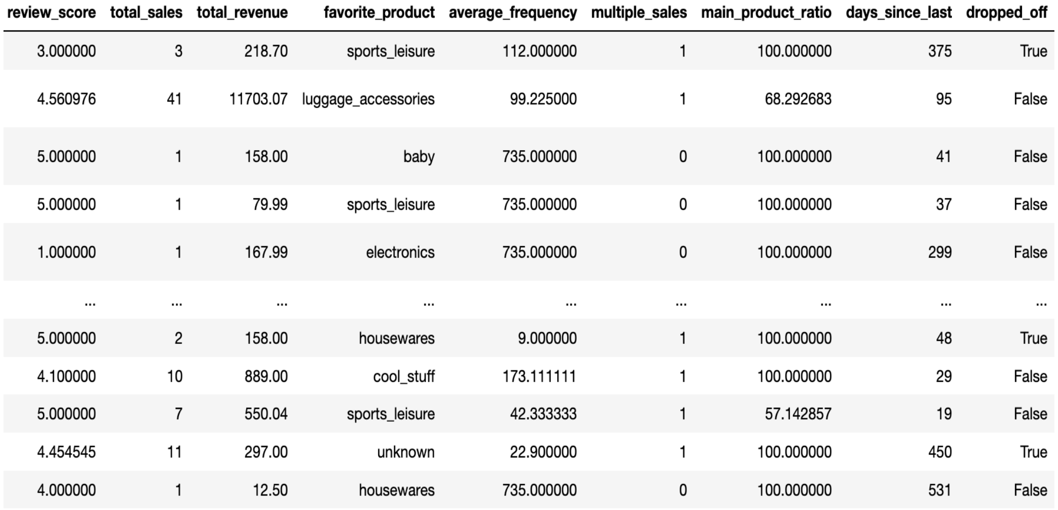
Now that these were completed, I move onto the sellers. The initial dataset contained a little information on geographical data. My immediate plan was to calculate similar variables to customers and move them into Sellers.



	seller_id	seller_zip_code_prefix	seller_city	seller_state
0	3442f8959a84dea7ee197c632cb2df15	13023	campinas	SP
1	d1b65fc7debc3361ea86b5f14c68d2e2	13844	mogi guacu	SP
2	ce3ad9de960102d0677a81f5d0bb7b2d	20031	rio de janeiro	RJ
3	c0f3eea2e14555b6faeea3dd58c1b1c3	4195	sao paulo	SP
4	51a04a8a6bdcb23deccc82b0b80742cf	12914	braganca paulista	SP
...	...	...	...	...
3090	98dddbc4601dd4443ca174359b237166	87111	sarandi	PR
3091	f8201cab383e484733266d1906e2fdfa	88137	palhoca	SC
3092	74871d19219c7d518d0090283e03c137	4650	sao paulo	SP
3093	e603cf3fec55f8697c9059638d6c8eb5	96080	pelotas	RS
3094	9e25199f6ef7e7c347120ff175652c3b	12051	taubate	SP

Figure 9: Initial Sellers Dataset

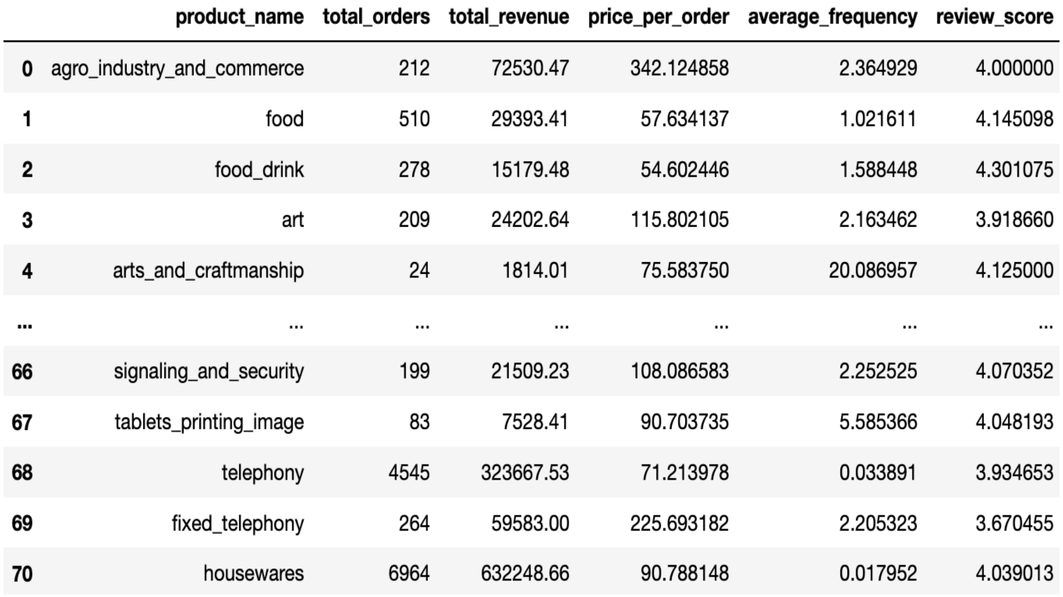
The variables that I decided to use calculate were as follows; average review score, total sales, total revenue, favorite product, average frequency, multiple sales, main product ratio, days since last, and dropped off. Those first 5 variables were the same variables calculated in the customers dataset, so they followed a similar code structure. Multiple sales is an boolean value that indicates whether a seller has sold more than one of their products. Main product ratio is the ratio between the count of the seller's favorite product with the total count of sales. Days last since is the number of days from the latest day to the date of their last sale. This is used to calculate the 'dropped_off' variable. This variable aims to roughly measure the stability of a given Seller by checking whether the days since last purchase is greater than their average frequency.



review_score	total_sales	total_revenue	favorite_product	average_frequency	multiple_sales	main_product_ratio	days_since_last	dropped_off
3.000000	3	218.70	sports_leisure	112.000000	1	100.000000	375	True
4.560976	41	11703.07	luggage_accessories	99.225000	1	68.292683	95	False
5.000000	1	158.00	baby	735.000000	0	100.000000	41	False
5.000000	1	79.99	sports_leisure	735.000000	0	100.000000	37	False
1.000000	1	167.99	electronics	735.000000	0	100.000000	299	False
...	...	...	...	...	...	...	...	...
5.000000	2	158.00	housewares	9.000000	1	100.000000	48	True
4.100000	10	889.00	cool_stuff	173.111111	1	100.000000	29	False
5.000000	7	550.04	sports_leisure	42.333333	1	57.142857	19	False
4.454545	11	297.00	unknown	22.900000	1	100.000000	450	True
4.000000	1	12.50	housewares	735.000000	0	100.000000	531	False

Figure 10: Cleaned Sellers Dataset

Finally moving onto the last dataset, the products dataset was created from scratch, where each record holds a unique product name. The variables calculated for this dataset were, again, the same 5 variables from both of the previous datasets.



	product_name	total_orders	total_revenue	price_per_order	average_frequency	review_score
0	agro_industry_and_commerce	212	72530.47	342.124858	2.364929	4.000000
1	food	510	29393.41	57.634137	1.021611	4.145098
2	food_drink	278	15179.48	54.602446	1.588448	4.301075
3	art	209	24202.64	115.802105	2.163462	3.918660
4	arts_and_craftmanship	24	1814.01	75.583750	20.086957	4.125000
...	...	...	...	...	...	...
66	signaling_and_security	199	21509.23	108.086583	2.252525	4.070352
67	tablets_printing_image	83	7528.41	90.703735	5.585366	4.048193
68	telephony	4545	323667.53	71.213978	0.033891	3.934653
69	fixed_telephony	264	59583.00	225.693182	2.205323	3.670455
70	housewares	6964	632248.66	90.788148	0.017952	4.039013

Figure 11: Cleaned Product Dataset

With the 3 cleaned datasets, I could move onto the clustering and recommendations.

D) Data Clusters Wrangling Process

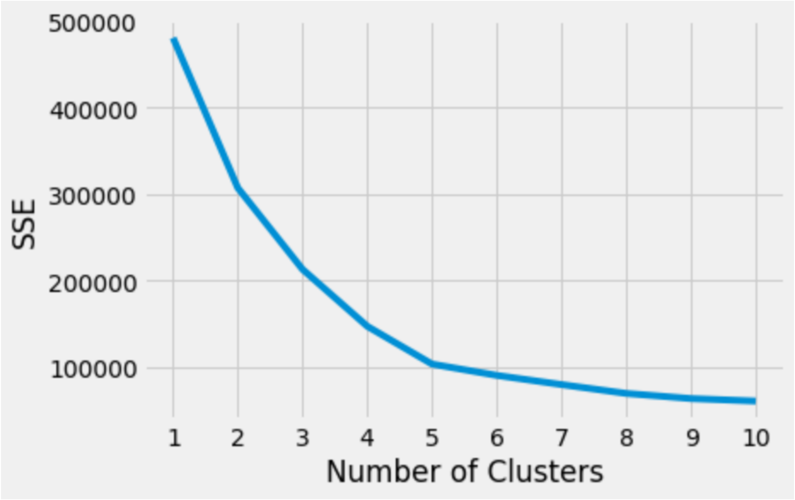
Clustering Code		
IPYNB	**PDF**	**Description**
CustomerClustering.ipynb	CustomerClustering.pdf	With the clean customers, sellers, and products datasets, these new notebooks were used to take the important features within a given dataset, clustering them multiple times to find the optimal k, and appending the cluster numbers to the cleaned dataset. Afterwards a table was created based on the average metrics between the variables of interest and the clusters. This was performed on all 3 datasets.
SellersClustering.ipynb	SellersClustering.pdf	
ProductClustering.ipynb	ProductClustering.pdf	
Recommendations.ipynb	Recommendations.pdf	This notebook details that creation and analysis of the item-based collaborative filtering method
Resulting CSV's		
CSV		**Description**
CustomerClustering.csv		The original cleaned Customers, Sellers, and Products Datasets but including the additional cluster variable that indicated a record's cluster number Rows > Unique C / S / P
SellerClustering.csv		
ProductClustering.csv		
cluster_customers.csv		Table comparing customer variable statistics with customer clusters. Rows > Variables Columns > Clusters
cluster_sellers.csv		
cluster_products.csv		
recommendations.csv		This table details the top 20 recommended choices for a given product category.

- *Full step by step process of cleaning and wrangling is documented in the Code*

Starting out with clustering, I took the clean datasets that were created in the first half and saved each of the features / important variables into their own array. After this I standardized each feature for proper clustering analysis. No categorical variables were use in my final analysis, so I did not need to dummify anything.

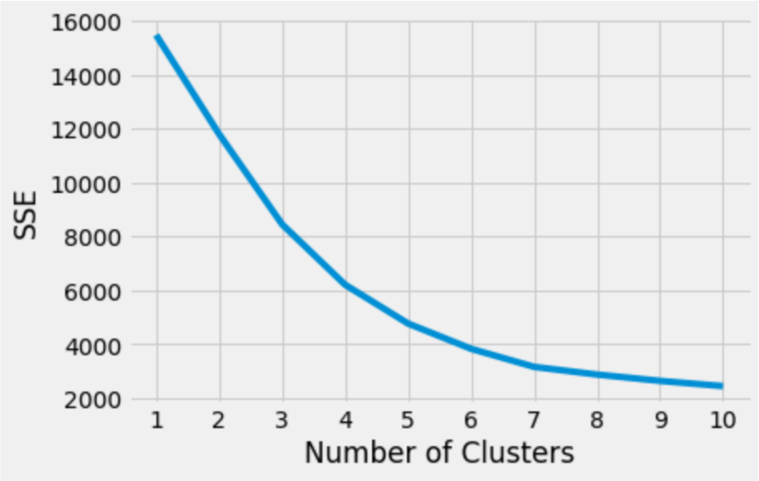
For Customers, Sellers, and Products, I ran 10 K-Means clustering algorithms with an increasing k for each iteration. Each cluster result returned the labels for every record and the total Sum of Squares Error. Using the SSE, I created an elbow graph to decide what k I wanted to use for each cluster.

Customers



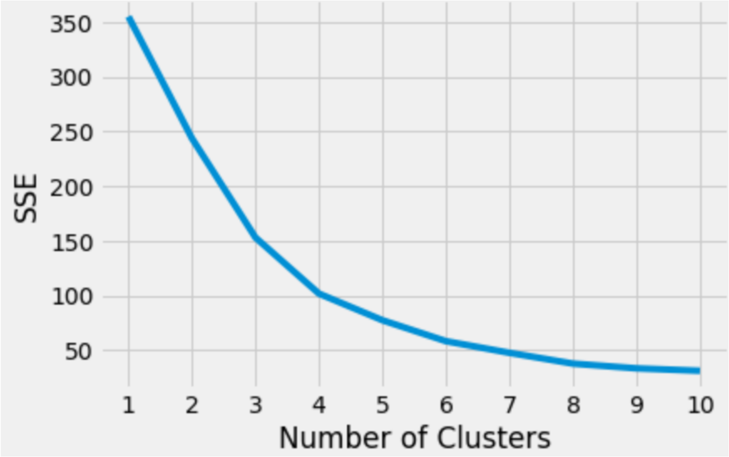
For customers clustering, the elbow point for the graph occurs at k = 5. With this I proceeded with k=5.

Sellers



The seller’s point was harder to pinpoint, but I decided on a k=4.

Products

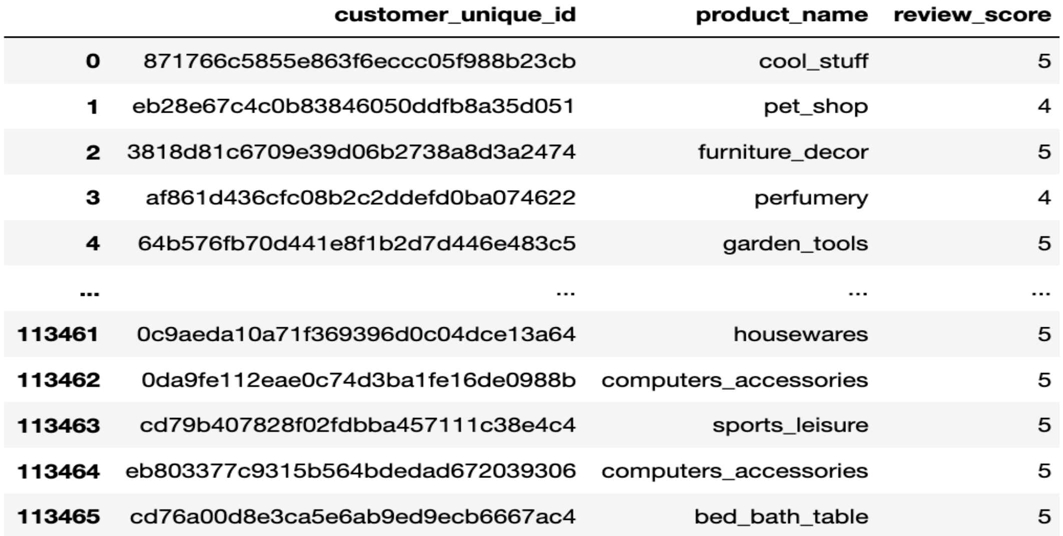


Similar to sellers, a k=4 for Products was the point to pick in my opinion.

After running the final clustering algorithms on each dataset, I saved the cluster labels / numbers back into the Clean datasets, which were saved to '[dataset]CleaningFull'. After this was done, I could now start collecting variable statistics based off of the clusters. This process of creating clustered variable statistics was very repetitive, however, I had to make sure I was using the correct measure of center based on the variable's distribution. Some variables, particularly customer features, did not make it into the final analysis as the first few trials with said variable included resulted in clusters that were poor in performance. However, I did add a few other metrics to some of the clusters for additional analysis.

	Cluster Variables	Outside Variables
Customers	- New Customer Proportion - Average Total Orders - Average Review Score - Average Total Price	- Came Back Proportion - Total Orders - Total Price - Proportion
Sellers	- Average Frequency - Review Score - Total Sales - Total Revenue	- Average total sales - Average total revenue - Price per item - Dropped off - Proportion - Top 5 products
Products	- Total Orders - Total Revenue - Price Per Order - Average Frequency - Review Score	

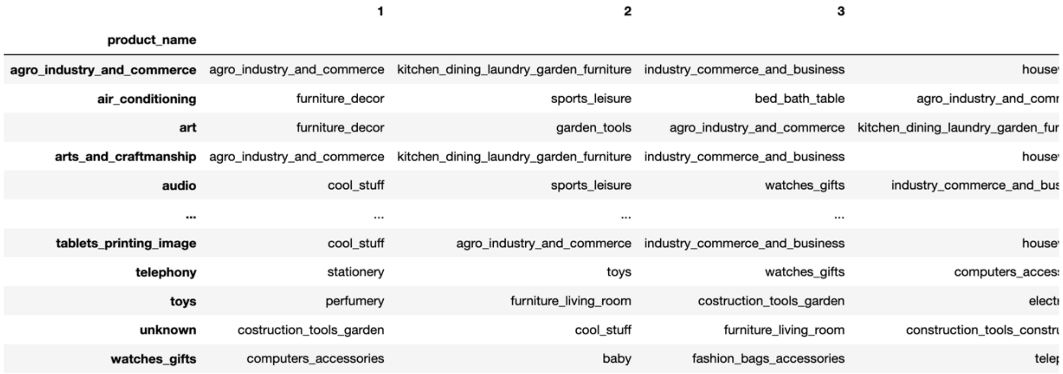
The final topic is the recommendation system that I created. I initially a dataset for it in the sellers cleaning code. This is due to the fact that the sellers code already contained the work that brought customer_unique_id, product_name, and review_score together. With a dataset containing those variables, I saved it for the Recommendation code.



	customer_unique_id	product_name	review_score
0	871766c5855e863f6eccc05f988b23cb	cool_stuff	5
1	eb28e67c4c0b83846050ddfb8a35d051	pet_shop	4
2	3818d81c6709e39d06b2738a8d3a2474	furniture_decor	5
3	af861d436cfc08b2c2ddefd0ba074622	perfumery	4
4	64b576fb70d441e8f1b2d7d446e483c5	garden_tools	5
...	...	...	...
113461	0c9aeda10a71f369396d0c04dce13a64	housewares	5
113462	0da9fe112eae0c74d3ba1fe16de0988b	computers_accessories	5
113463	cd79b407828f02fdbba457111c38e4c4	sports_leisure	5
113464	eb803377c9315b564bdedad672039306	computers_accessories	5
113465	cd76a00d8e3ca5e6ab9ed9ecb6667ac4	bed_bath_table	5

Figure 12: Recommendation Variables

With this I created a new notebook titled recommendations.ipynb that took these variables and created a pivot table that compared customers and products by average review. With this, I was able to get a correlation matrix for all items, and after removing the diagonal 1s, I got the 20 highest correlations for every product and listed them in a table. With this, we have a working item-based collaborative filtering method. It is not perfect, however, and products with not correlations will default to the first alphabetical product item, that being Agro_industry_and_commerce.



	1	2	3	
product_name				
agro_industry_and_commerce	agro_industry_and_commerce	kitchen_dining_laundry_garden_furniture	industry_commerce_and_business	house
air_conditioning	furniture_decor	sports_leisure	bed_bath_table	agro_industry_and_comr
art	furniture_decor	garden_tools	agro_industry_and_commerce	kitchen_dining_laundry_garden_fur
arts_and_craftmanship	agro_industry_and_commerce	kitchen_dining_laundry_garden_furniture	industry_commerce_and_business	house
audio	cool_stuff	sports_leisure	watches_gifts	industry_commerce_and_bus
...	...	...	...	
tablets_printing_image	cool_stuff	agro_industry_and_commerce	industry_commerce_and_business	house
telephony	stationery	toys	watches_gifts	computers_acces
toys	perfumery	furniture_living_room	costruction_tools_garden	electi
unknown	costruction_tools_garden	cool_stuff	furniture_living_room	construction_tools_constru
watches_gifts	computers_accessories	baby	fashion_bags_accessories	telep

Figure 13: Item-Based Recommendations

Notes:

1) The code and csv files may contain variables that were never used in the final analyses, but were left in.

2) Due to the clustering algorithms changing the final results when reran, I avoided touching these parts of the code after generating the ones that I used.
3) All prices are presented in terms of Dollars and not Brazilian Reals, which is what the original data is based on

References:

Olist, Sionek, A., Dabague, Galeazzi, T., Oliveira, M., & Horning, W. (n.d.). *Brazilian E-Commerce Public Dataset by Olist*. Kaggle. Retrieved May 3, 2021, from https://www.kaggle.com/olistbr/brazilian-ecommerce

Olist. (2021). Owler. https://www.owler.com/company/olist