
olist Product Recommendation Engine
CAPSTONE FINAL PRESENTATION

Data Set

Kaggle - Brazilian E-Commerce Public Dataset by Olist
https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce

Description:

Olist is the largest department store in Brazilian marketplaces, connecting small businesses from all over Brazil to customers online and aiding in shipping products to customers as well. The data set contains real commercial data of 100k orders, though has been anonymised to protect customers' personal data

Capstone Goal

Original:

Segment buyers and sellers into ranked customer groups and provide Olist with statistics of these segments to aid in targeted advertising of current products and future marketing campaigns to grow these customer groups

Adapted:

Construct a product recommendation engine that, given a specific customer, will output product recommendations ranked by the products similarity to the customers previous purchases

Customer Segmentation

RFM Ranking (Buyers and Sellers):

Recency:

Measure of when the customer last made an order

Frequency:

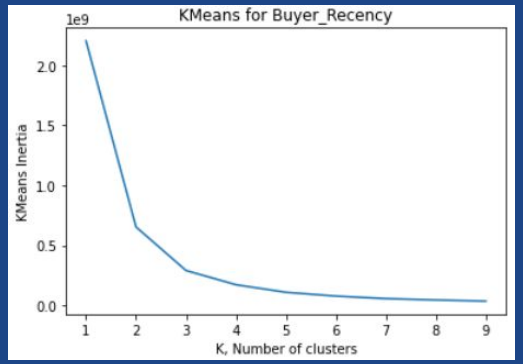
Number of orders the customer has in the dataset

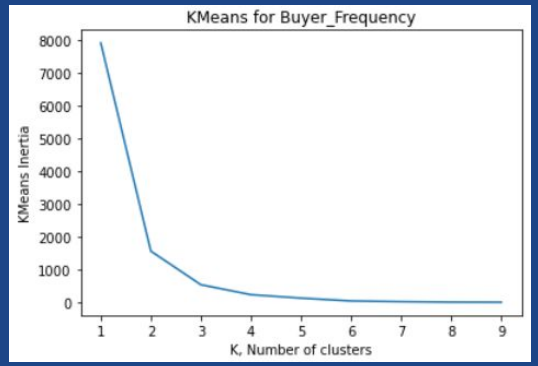
Monetary:

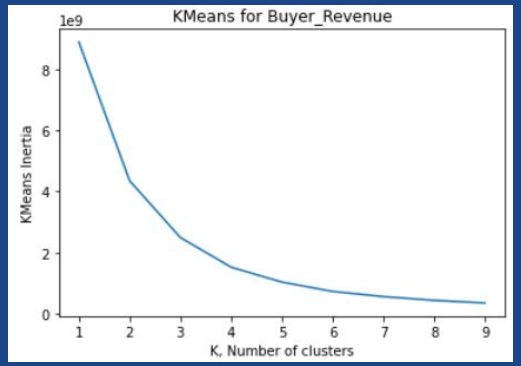
Total amount the customer has spent/made in the dataset

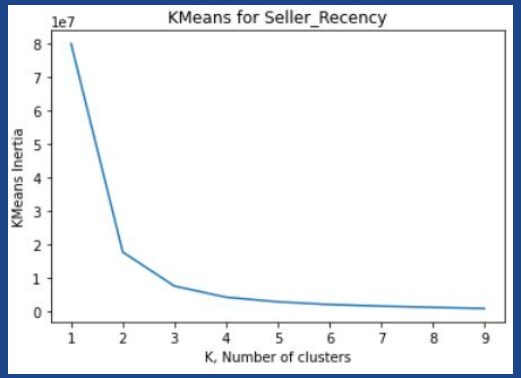
Customer Segmentation

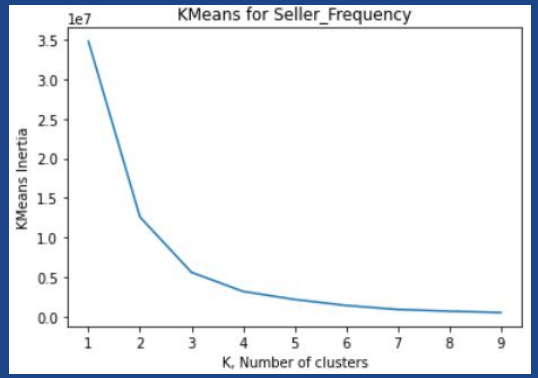
RFM KMeans Clustering

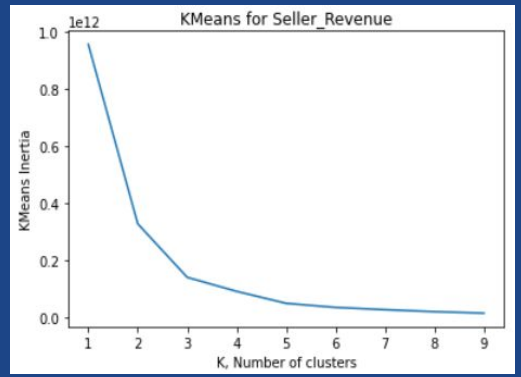












Customer Segmentation

RFM KMeans Clustering - Results

Cluster Counts

Customer	Recency	Frequency	Monetary
Buyer	4	4	5
Seller	4	4	5

Overall Scores - Buyers

Buyer_OverallRFMScore	Buyer_Recency	Buyer_Frequency	Buyer_Revenue
0	486.761675	1.000000	96.817523
1	336.774760	1.011934	123.618145
2	212.951759	1.029242	145.506507
3	98.333307	1.046431	170.908309
4	112.441015	1.278782	469.445107
5	111.177611	1.663540	875.757711
6	104.621469	2.180791	1434.112863
7	102.297468	2.778481	2405.420506
8	97.207547	4.150943	3225.849434
9	69.083333	4.166667	4316.088333

Overall Scores Sellers

Seller_OverallRFMScore	Seller_Recency	Seller_Frequency	Seller_Revenue
0	525.327273	3.523636	739.880145
1	318.578804	7.076087	1200.217310
2	161.557613	8.923868	1680.105700
3	33.487252	13.410057	2236.358215
4	34.827027	59.443243	13024.881946
5	23.024691	126.543210	19654.099012
6	22.560976	232.658537	46439.367073
7	14.851852	425.074074	53798.847037
8	22.625000	638.875000	128986.017500
9	7.375000	1187.125000	175086.316250
10	10.600000	1252.600000	254983.110000

Customer Segmentation

RFM - Grouping

Buyers -

Buyer_OverallRFMScore	
0	13490
1	21870
2	26264
3	25565
4	4925
5	1599
6	531
7	158
8	53
9	12

Buyer_RFM_Ranking	
0) Lowest - Overall = 0	13490
1) Low - Overall > 0	48134
2) Medium - Overall > 2	32089
3) High - Overall > 5	689
4) Highest - Overall > 7	65

Sellers -

Seller_OverallRFMScore	
0	275
1	368
2	486
3	1412
4	185
5	162
6	41
7	27
8	8
9	8
10	5

Seller_RFM_Ranking	
0) Lowest - Overall = 0	275
1) Low - Overall > 0	2266
2) Medium - Overall > 4	347
3) High - Overall > 6	89

Goal Shifting

The decision to abandon Customer Segmentation analysis was based on the lack of demographic data past the geographic data

The data set was much better suited to making a product recommendation engine for customers by:

- Separate dataset into category groups
- Find similarity between groups in a category
- Determine which group the customer is in
- Make recommendations based on products in the customer’s group and those in groups similar

Category Groups

Every product in the set has a product Id, category group feature, as well as the id of the customer and seller.

Can group by pivot tables:

Rows	Columns	SIze
Customer Id	Product Category	(93161, 31)
Seller Id	Product Category	(2918,31)
Customer Id with >1 Product Id purchase	Product Id	(2868, 4520)

Category Groups - size reduction

- Normalize to percentage (nearest 5%)

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Sample 1	0	1	32	1	45
Sample 2	77	0	1	0	1
Sample 3	1	19	1	19	29
Sample 4	17	1	16	0	25
Sample 5	34	4	12	5	17
Sample 6	0	0	31	0	43
Sample 7	17	14	5	13	0
Sample 8	23	0	23	1	34

=>

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Sample 1	0	0	40	55	0
Sample 2	95	0	0	0	0
Sample 3	0	25	0	40	25
Sample 4	25	0	25	40	0
Sample 5	45	5	15	20	5
Sample 6	0	0	40	55	0
Sample 7	30	25	10	0	25
Sample 8	25	0	25	40	0

- Remove Duplicate Rows and columns

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Sample 1	0	0	40	55	0
Sample 2	95	0	0	0	0
Sample 3	0	25	0	40	25
Sample 4	25	0	25	40	0
Sample 5	45	5	15	20	5
Sample 6	0	0	40	55	0
Sample 7	30	25	10	0	25
Sample 8	25	0	25	40	0

=>

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Sample 1	0	0	40	55	0
Sample 2	95	0	0	0	0
Sample 3	0	25	0	40	25
Sample 4	25	0	25	40	0
Sample 5	45	5	15	20	5
Sample 7	30	25	10	0	25

=>

	Cat 1	Cat 2	Cat 3	Cat 4
Sample 1	0	0	40	55
Sample 2	95	0	0	0
Sample 3	0	25	0	40
Sample 4	25	0	25	40
Sample 5	45	5	15	20
Sample 7	30	25	10	0

Category Groups - size reduction

- Rename Grouped Rows, and use as column

	Cat 1	Cat 2	Cat 3	Cat 4
Sample 1	0	0	40	55
Sample 2	95	0	0	0
Sample 3	0	25	0	40
Sample 4	25	0	25	40
Sample 5	45	5	15	20
Sample 7	30	25	10	0

=>

	Group Name	Cat 1	Cat 2	Cat 3	Cat 4
0	A	0	0	40	55
1	B	95	0	0	0
2	C	0	25	0	40
3	D	25	0	25	40
4	E	45	5	15	20
5	F	30	25	10	0

Results of Reduction

Group	Category	Original Size	New Group Id	New Size
Customer Id	Product Category	(93161, 31)	*CustomerGroupId*	(536, 31)
Seller Id	Product Category	(2918, 31)	*SellerGroupId*	(1031, 31)
Customer Id	Product Id	(2868, 4520)	*CustomerProductGroupId*	(2810, 3349)
(only customers with more than one product_ids used)				

Similarity Scores - row a vs row b

- Euclidean Distance

Sqrt of sum of squared distances

$$\sqrt{\sum_{i=0}^{n} (a_i - b_i)^2}$$

	1	2	3	4	5	6
A	100	0	0	40	60	35
B	0	50	33	10	0	30
C	0	50	33	50	40	30
D	0	0	33	0	0	0
E	0	0	0	0	0	5

>

	A	B	C	D	E
A	0	134.59	118.8	132.34	126.89
B	134.59	0	56.57	59.16	65.68
C	118.8	56.57	0	86.6	91.18
D	132.34	59.16	86.6	0	33.38
E	126.89	65.68	91.18	33.38	0

- Modified Euclidean

Modify Rows

$$amod = [\, a_i \text{ if } a_i \text{ != 0, else } -b_i \,]$$

$$bmod = [\, b_i \text{ if } b_i \text{ != 0, else } -a_i \,]$$

Take Euclidean Dist

$$\sqrt{\sum_{i=0}^{n} (amod_i - bmod_i)^2}$$

	1	2	3	4	5	6
A	100	0	0	40	60	35
B	0	50	33	10	0	30
C	0	50	33	50	40	30
D	0	0	33	0	0	0
E	0	0	0	0	0	5

>

	A	B	C	D	E
A	0	263.97	234.27	264.68	248.39
B	263.97	0	89.44	118.32	124.02
C	234.27	89.44	0	173.21	177.15
D	264.68	118.32	173.21	0	66.75
E	248.39	124.02	177.15	66.75	0

Similarity Scores - row a vs row b

- Cosine Similarity

Treat a and b as vectors

Find angle between them using definition of dot product

	1	2	3	4	5	6
A	100	0	0	40	60	35
B	0	50	33	10	0	30
C	0	50	33	50	40	30
D	0	0	33	0	0	0
E	0	0	0	0	0	5

>

	A	B	C	D	E
A	1	0.17	0.46	0	0.27
B	0.17	1	0.79	0.49	0.44
C	0.46	0.79	1	0.36	0.32
D	0	0.49	0.36	1	0
E	0.27	0.44	0.32	0	1

Dot Product

$$a \bullet b = \sum_{i=0}^{n} (a_i * b_i)$$

$$a \bullet b = |a||b|cos(\theta_{ab})$$

$$cosine\ similarity = arccos(\frac{a \bullet b}{|a||b|})$$

Interpretation

0 - orthogonal (no similarity)

1 - parallel (perfect similarity)

Similarity Scores - Performance

- Model
  - Agglomerative Clustering (allows for custom distance measurement and using cosine similarity)
  - Multiple Linkages (average, complete)
  - K clusters (1 - 100)
- Output
  - Average Silhouette Score of samples:

$$a = \text{mean distance between the samples of the cluster this sample is a part of}$$
$$b = \text{distance between the sample and the nearest cluster that the sample is not a part of}$$

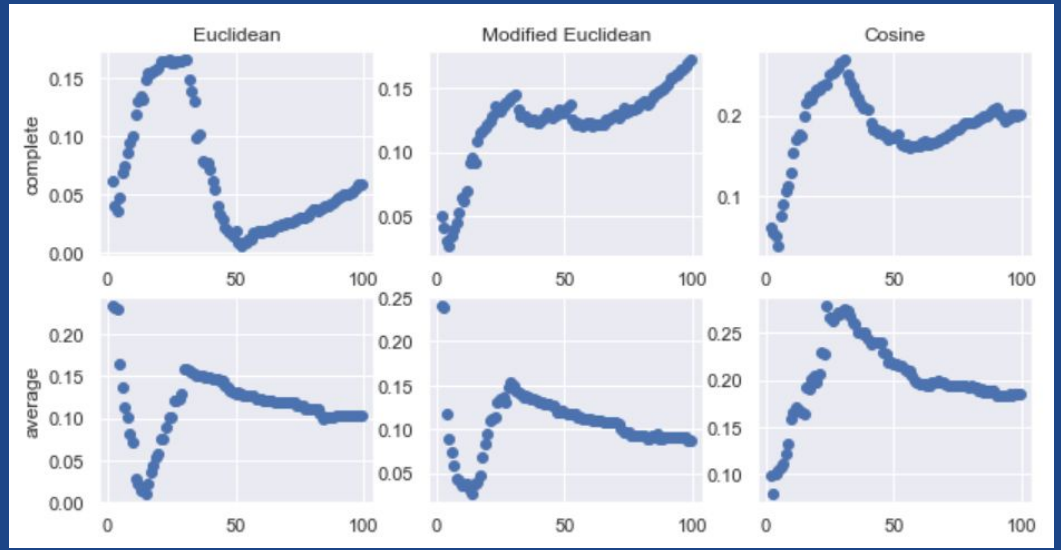
$$\text{Silhouette Score} = \frac{(b-a)}{max(a,b)}$$

Similarity Scores - Performance

Customer Id vs Product Category

New Group Id	New Size
CustomerGroupId	(536, 31)
SellerGroupId	(1031, 31)
CustomerProductGroupId	(2810, 3349)

Distance Measurement	Linkage	Max_Silhouette_Score	Number_Of_Clusters
Euclidean	complete	0.166001	24
Modified Euclidean	complete	0.172348	99
Cosine	complete	0.269330	31
Euclidean	average	0.233287	2
Modified Euclidean	average	0.239850	2
Cosine	average	0.278595	24

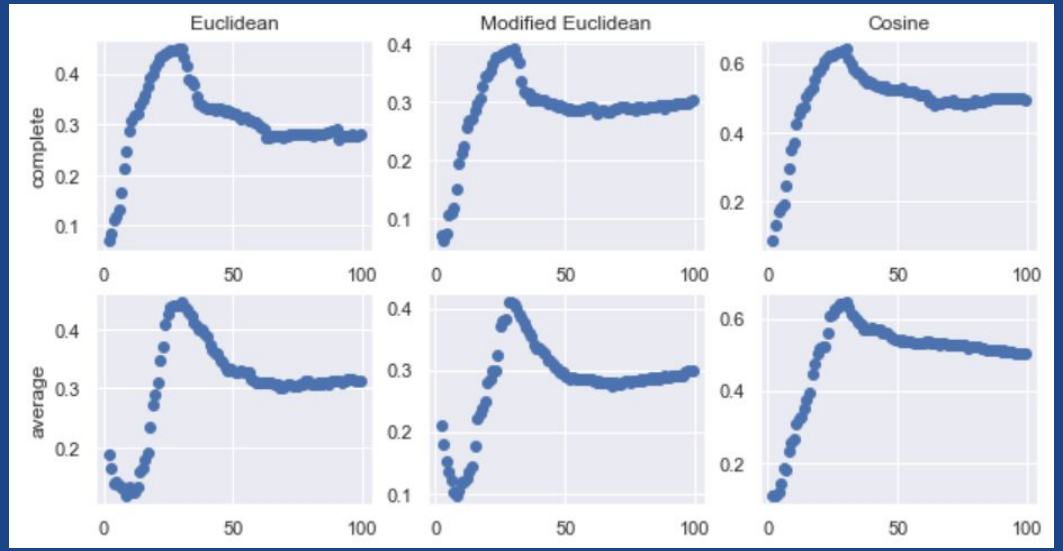


Similarity Scores - Performance

Seller Id vs Product Category

New Group Id	New Size
CustomerGroupId	(536, 31)
SellerGroupId	(1031, 31)
CustomerProductGroupId	(2810, 3349)

Distance Measurement	Linkage	Max_Silhouette_Score	Number_Of_Clusters
Euclidean	complete	0.448936	30
Modified Euclidean	complete	0.392026	30
Cosine	complete	0.642732	30
Euclidean	average	0.445347	30
Modified Euclidean	average	0.408833	29
Cosine	average	0.643737	30

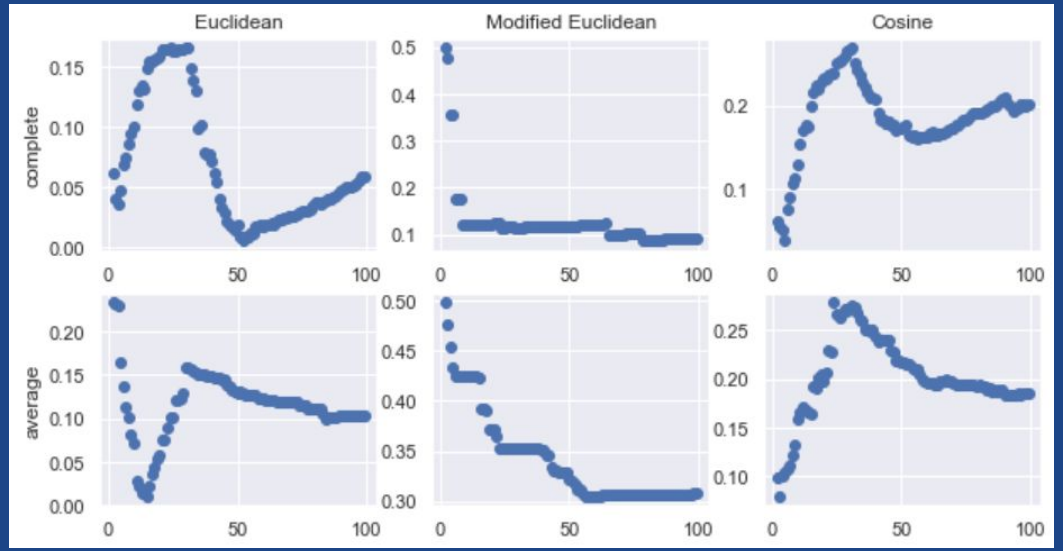


Similarity Scores - Performance

Customer Id vs Product Id

New Group Id	New Size
CustomerGroupId	(536, 31)
SellerGroupId	(1031, 31)
CustomerProductGroupId	(2810, 3349)

Distance Measurement	Linkage	Max_Silhouette_Score	Number_Of_Clusters
Euclidean	complete	0.166001	24
Modified Euclidean	complete	0.497897	2
Cosine	complete	0.269330	31
Euclidean	average	0.233287	2
Modified Euclidean	average	0.497897	2
Cosine	average	0.278595	24



Product Recommendation Engine

Based on

- CSV for each Category Group
  - Row: Group Id, Number of Customers in Group, Category %
- CSV for Cosine Similarity
  - Row: Group Id, similarity to every other group in order

Recommendations produced using:

- Similarity Table for known customers (Group assigned)
- Category Group Table to Similarity Table for new customers
- Agglomerative Clustering with ‘average’ linkage and cosine similarity
- With RFM Scores ranking

Product Recommendation Engine

Input:

- Customer Id, Similarity Minimum

Output:

- Product ids the customer has not bought previously based on the products of the following, if their similarity is high enough
  - Customers who have purchased products in categories in similar proportions to the user
  - Sellers who sell products in categories in similar proportions to the user's purchases
  - Customers who purchased certain products in similar proportions to the user

Product Recommendation Engine - Example

User Id - ‘3e43e6105506432c953e165fb2acf44c’
Minimum Similarity - 0.8

Recommendations:

product_id	SellerGroupId_CosSim	SellerGroupId_Aglom	CustomerGroupId_CosSim	Total	product_category_name_english
0b814a3c8fa6dbb849df7c28c1bd6831	0.0	0.802525	0.916698	1.719223	Furniture_Bedroom
1b0fb6d6a05121d4b76a49f44c2c427b	0.0	0.802525	0.916698	1.719223	Furniture_Bedroom
odee5506e0699fa985e30a00b701b17a	0.0	0.802525	0.916698	1.719223	Furniture_Bedroom
aff39c649de8c4e36d325880de8f4338	0.0	0.802525	0.916698	1.719223	Furniture_Bedroom
428f4b96ba9f630e761b333673039ae6	0.0	0.802525	0.916698	1.719223	Furniture_Bedroom
...	...	...	...	...	...
2eefd8ed7a9782380fe14a1efdec7418	0.0	0.000000	0.943983	0.943983	Furniture_Home
cebad0ed16ecd450b97d2be843d3da86	0.0	0.000000	0.943983	0.943983	NaN
3512f777c335f8297b5ad43416b69cf8	0.0	0.000000	0.935601	0.935601	Furniture_Bedroom
7f9f228320c43765cfe30cdc090e0be4	0.0	0.000000	0.935601	0.935601	Furniture_Bedroom
a02d0123079f4ae96001ba2010d1a2df	0.0	0.000000	0.935601	0.935601	Construction

Product Recommendation Engine - Example

User Id - ‘3e43e6105506432c953e165fb2acf44c’
Minimum Similarity - 0.8
Customer’s Buying Habits -

Furniture_Bedroom	7
Furniture_Home	3
Construction	1
Technology	1
HomeAppliance	1

Top 200 Recommendations

Furniture_Bedroom	159
Furniture_Home	26
Construction	7
Gardening	5
HomeAppliance	1
Home	1

Top 200 Recommendations with RFM

Furniture_Bedroom	185
Furniture_Home	9
HomeAppliance	1
Beauty	1
Office	1
Home	1
Construction	1

Product Recommendation Engine Updating

Method provided that will take a list of new orders and for every customer in the new order set

- Add them to appropriate group based on all their orders (old and new), and add new group for them if needed
- Update count of members for each group
- If new group created, add corresponding row and column in Similarity Score Table

If any group is now empty, remove corresponding row from Category Group and corresponding row and column from Similarity Table. Then update all group numbers in tables and orders.

Improvements/ Future Work

Alternative Clustering to Agglomerative

- Chosen for use of custom distances, which were not the best performing similarity score
- Cannot predict new samples, must be fitted on every use

Limiting Recommendations

- All products from similar buyers and similar sellers returned
- Seller's product scores dominate, should be weighted further and capped to maximum number of recommendations

Increasing Variety

- Customer's top Product Category dominates
- Recommendations could be proportioned based on user's orders

Final Note

- Quality of recommendations could not be determined during this capstone, as it ultimately depends on user interaction with the recommendations and the desired effect from Olist

- Code has been provided for generating CSV files that could be converted to SQL tables, generating recommendations based on these files, and updating these files for new orders. This code can be easily converted to run on a live production environment.

Questions?