

Customer Segmentation Using Machine Learning

Akshay Sreekumar Nair (191IT103)
Information Technology
National Institute Of Technology Karnataka
Surathkal, India 575025
e-mail: a.191it103@nitk.edu.in

Abstract—Effective management of customer’s knowledge leads to efficient Customer Relationship Management. Predicting relevant segmentation groups for customers using clustering, in this case k-means, gives a fair idea about the customer’s behaviour. The data inferred from this data mining then may later be used to segregate customer’s based on their needs and also, improve marketing strategies.

Index Terms—Customer Relationship Management, K-means, Data Mining, Customer satisfaction, Marketing.

I. INTRODUCTION

Customer Segmentation plays an important role for a company to customize and make further improvements its relation with the customers. With customer segmented into a variety of groups, we find similar characteristics in each of the customer’s behaviours and need. Then, those are generalized into groups to satisfy demands of those customer segments targeting specific customer groups. This data can then be used to market specific products based on the customer’s need.

II. BACKGROUND & RELATED WORK

There exists a clear trend that customers are switching their shopping mode from traditional outlets to internet, eliminating the time and geographical constraints on shopping. [Zha+21] Customer segmentation divides customers into groups based on common characteristics, which is useful for banks, businesses, and companies to improve their products or service opportunities. [Abd21] Effective management of customer’s knowledge leads to efficient Customer Relationship Management (CRM). To accurately predict customer’s behaviour, clustering, especially K-means, is one of the most important data mining techniques used in customer relationship management marketing, with which it is possible to identify customers’ behavioural patterns and, subsequently, to align marketing strategies with customer preferences so as to maintain the customers. [ZE20]

Algorithm	Davis-Bouldin Index	Silhouette Coefficient	Dunn Index
Hierarchical Clustering	1.834418	0.289862	0.57328051
K-means Clustering	2.017857	0.311893	0.64092362

Fig. 1. different validation metrics for the Hierarchical and K-means clustering [Abd21]

Previously conducted researches on customer segmentation holds the fact of customer satisfaction, marketing strategy

improvements and also helped target specific products. Publications suggesting these facts,

- H. -H. Zhao, X. -C. Luo, R. Ma and X. Lu, "An Extended Regularized K-Means Clustering Approach for High-Dimensional Customer Segmentation With Correlated Variables," in IEEE Access, vol. 9, pp. 48405-48412, 2021, doi: 10.1109/ACCESS.2021.3067499.
- Abdulhafedh, Azad. (2021). Incorporating K-means, Hierarchical Clustering and PCA in Customer Segmentation. 3. 12-30. 10.12691/jcd-3-1-3.
- Zare, Hamed Emadi, Sima. (2020). Determination of Customer Satisfaction using Improved K-means algorithm. Soft Computing. 24. 10.1007/s00500-020-04988-4.

III. PROBLEM STATEMENT

Products are offered randomly without checking the previous order details or calculating the probability of the customer buying a product and acting accordingly. This is achieved by segregating the customers based on their need using k-means clustering.

A. Objectives

- Improve customer satisfaction
- Reduce wastage of unwanted products
- Improve marketing strategy
- Targeting audience on the basis of their needs.

IV. DESIGN METHODOLOGY & REQUIREMENT ANALYSIS

Clustering is an unsupervised learning method that divides the feature space into clusters or groups of similar objects. In general, clustering is used for pattern recognition. In this paper, we will be following the following steps for achieving our goal,

- 1) Segmentation Basis
- 2) Data Preparation
- 3) Segmentation with K-means Clustering
- 4) Hyper-parameter Tuning
- 5) Visualization and Interpretation of the Results

A. Segmentation Basis

Approach in terms of customer’s behavioural aspect (other cases might include geographical preferences, etc.) The primary features used in this case are:

- Number of products ordered
- Average return rate

- Total spending.

	customer_id	products_ordered
0	1000661	1
1	1001914	1
2	1002167	3
3	1002387	1
4	1002419	2

Fig. 2. Number of products ordered

	average return rate	count of unit return rate
0	0.000000	24823
9	0.013889	1
10	0.066667	1
8	0.083333	1
12	0.125000	1
5	0.166667	2
6	0.200000	2
4	0.250000	5
1	0.333333	13
11	0.400000	1
3	0.500000	9
7	0.666667	2
2	1.000000	13

Fig. 3. Average return rate

	customer_id	total_spending
0	1000661	260.0
1	1001914	79.2
2	1002167	234.2
3	1002387	89.0
4	1002419	103.0

Fig. 4. Total spending

B. Data Preparation

The used dataset comprises of data of about 25000 customers. The dataset is well formatted without any NULL values. The three features as mentioned in the Segmentation Basis will be calculated using customer_id. The plotly library will help in data visualization. Pandas and numpy will come in handy for Data preparation.

	variant_id	customer_id	order_id	net_quantity	gross_sales	discounts	returns	net_sales	taxes	total_sales	return
count	7.005200e+04	7.005200e+04	7.005200e+04	70052.000000	70052.000000	70052.000000	70052.000000	70052.000000	70052.000000	70052.000000	70052.000000
mean	2.442320e+11	6.013091e+11	5.506075e+13	0.701179	61.776302	-4.949804	-10.246051	46.580348	9.123636	55.703982	
std	4.255079e+12	6.223201e+12	2.587640e+13	0.739497	31.800689	7.769972	25.154677	51.802690	10.305236	61.920557	
min	1.001447e+07	1.000661e+06	1.000657e+13	-3.000000	0.000000	-200.000000	-237.500000	-237.500000	-47.500000	-285.000000	
25%	2.692223e+07	3.295985e+06	3.270317e+13	1.000000	51.670000	-8.340000	0.000000	47.080000	8.375000	56.227500	
50%	4.494514e+07	5.566107e+06	5.522207e+13	1.000000	74.170000	0.000000	0.000000	63.330000	12.660000	76.000000	
75%	7.743106e+07	7.815352e+06	7.736876e+13	1.000000	79.170000	0.000000	0.000000	74.170000	14.840000	89.000000	
max	8.422212e+13	9.977409e+13	9.999554e+13	6.000000	445.000000	0.000000	0.000000	445.000000	63.340000	445.000000	

Fig. 5. Statistics of various columns

	products_ordered	average_return_rate	total_spending
0	1	0.0	260.0
1	1	0.0	79.2
2	3	0.0	234.2
3	1	0.0	89.0
4	2	0.0	103.0

Fig. 6. customers.head()

Distribution of the Features

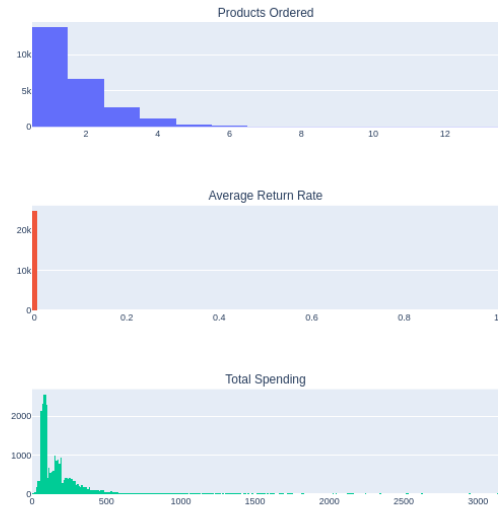


Fig. 7. Distribution of the features

All 3 distributions are positively skewed distributions. Products ordered shows a power-law distribution and average return rate of 99% of the customers are 0.

The data is to be scaled now, for K-means clustering. Logarithmic transformation is a suitable transformation for skewed data. This will scale down proportionally the 3D space which our data is spread, yet preserving the proximity between the points.



Fig. 8. Distribution of the features after Scaling

	products_ordered	average_return_rate	total_spending	log_products_ordered	log_average_return_rate	log_total_spending
0	1	0.0	260.0	0.693147	0.0	5.564520
1	1	0.0	79.2	0.693147	0.0	4.384524
2	3	0.0	234.2	1.386294	0.0	5.460436
3	1	0.0	89.0	0.693147	0.0	4.499810
4	2	0.0	103.0	1.098612	0.0	4.644391

Fig. 9. First few entries after Scaling

	log_products_ordered	log_average_return_rate	log_total_spending
0	0.693147	0.0	5.564520
1	0.693147	0.0	4.384524
2	1.386294	0.0	5.460436
3	0.693147	0.0	4.499810
4	1.098612	0.0	4.644391
...
24869	1.098612	0.0	5.560682
24870	1.098612	0.0	5.495117
24871	0.693147	0.0	4.499810
24872	1.098612	0.0	5.590987
24873	0.693147	0.0	4.174387

24874 rows × 3 columns

Fig. 10. Features to be used as an input to the model

C. Segmentation with K-means Clustering

The outcome is now ready to be clustered using the k-means algorithm.

The k-means algorithm works in the way,

- 1) Initialize k=n centroids=number-of-clusters
- 2) Assign each data point to the closest centroid based on euclidean distance, thus forming the groups
- 3) Move centers to the average of all points in the cluster

Repeat steps 2 and 3 until convergence.
The algorithm tries to minimize — optimize the within-cluster sum-of-squared-distances or inertia of each cluster.

$$\sum_{i=0}^n \min_{\mu_j \in C} (||x_i - \mu_j||^2)$$

Fig. 11. Mathematical expression of within-cluster sum-of-squared-distances or inertia where X is the points in the cluster and μ is the current centroid.

```
In [30]: kmeans_model = KMeans(init='k-means++',
                                max_iter=500,
                                random_state=42)

In [31]: kmeans_model.fit(customers.iloc[:,3:])

# print the sum of distances from all examples to the center of the cluster
print("within-cluster sum-of-squares (inertia) of the model is:", kmeans_model.inertia_)

within-cluster sum-of-squares (inertia) of the model is: 1066.631474556988
```

Fig. 12. Creating K-Means Model

Selecting the optimal clusters number is of primary concern. The number of clusters is suitable if,

- Samples in a cluster are as similar as possible;
- Samples belonging to different clusters are as incommensurable as possible.

That is, the clusters should have the maximum possible compression and should be as far from each other as possible. For a proper clustering, both criteria must be met, since if only the compression criterion is held, then any data can be considered as a cluster. In this case, the best clustering is to get the entire data into a cluster, so that the distance between each cluster is zero. [ZE20]

The next stage comprises of selecting the optimal value of k.

D. Hyper-parameter Tuning

While selecting k, we are going to decide against the optimization criteria of the K-means, inertia, using elbow method.

We are going to build different K-means models with k values 1 to 15, and save the corresponding inertia values. With the elbow method, we are going to select the k value where the decrease in the inertia stabilizes.

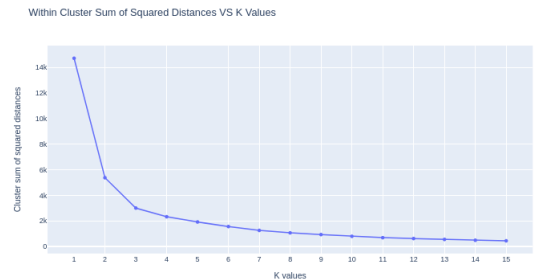


Fig. 13. Visualization for the selection of number of segments

When $k=1$, the data isn't grouped yet, hence inertia is the highest. Inertia continues to decrease as the value of k increases. At $k=4$, the descent stabilizes and continues linearly afterwards, forming an elbow at $k=4$. This points out the optimal number of customer group is 4.

```
In [41]: customers.head()
Out[41]:
```

	products_ordered	average_return_rate	total_spending	log_products_ordered	log_average_return_rate	log_total_spending	clusters
0	1	0.0	260.0	0.693147	0.0	5.564520	1
1	1	0.0	79.2	0.693147	0.0	4.384524	0
2	3	0.0	234.2	1.386294	0.0	5.460436	3
3	1	0.0	89.0	0.693147	0.0	4.499810	0
4	2	0.0	103.0	1.098612	0.0	4.644391	1

Fig. 14. customers.head()

```
In [43]: customers.tail()
Out[43]:
```

	products_ordered	average_return_rate	total_spending	log_products_ordered	log_average_return_rate	log_total_spending	clusters	is_center
24873	1.000000	0.000000	64.000000	0.693147	0.000000	4.174387	0	0
24874	1.014963	0.001153	76.551009	0.700601	0.001152	4.350036	0	1
24875	1.527302	0.000547	159.911940	0.927152	0.000547	5.080857	1	1
24876	3.951300	0.000559	581.869314	1.599650	0.000559	6.367963	2	1
24877	2.400057	0.000519	284.126659	1.223792	0.000519	5.652933	3	1

Fig. 15. customers.tail()

E. Visualization and Interpretation of the Results

When k is now initialized to 4, the visualization is as represented in the following graph,

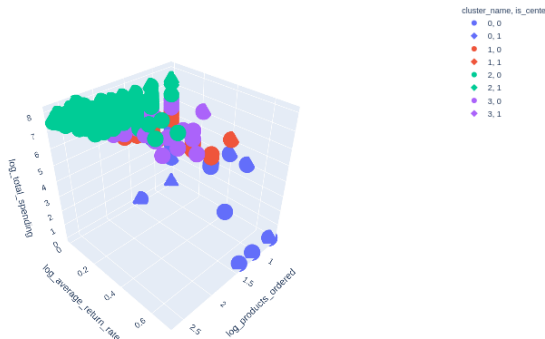


Fig. 16. Visualizing Customer Segmentation

Data points are shown in spheres and centroids of each group are shown with cubes. We get 4 different customer groups,

- 1) Blue: Customers who ordered at least one product, with maximum total spending of 100 and having the highest average return rate. They might be the newcomers of the e-commerce website.
- 2) Red: Customers who ordered 1 to 4 products, with average total spending of 150 and a maximum return rate of 0.5.
- 3) Purple: Customers who ordered 1 to 4 products, with average total spending of 300 and a maximum return rate of 0.5.

- 4) Green: Customers who ordered 1 to 13 products, with average total spending of 600 and average return rate as 0. It makes the most favourable customer group for the company.

Customer Groups	Customer Group Magnitude
0	10468
1	7236
2	5100
3	2074

Fig. 17. Cardinality

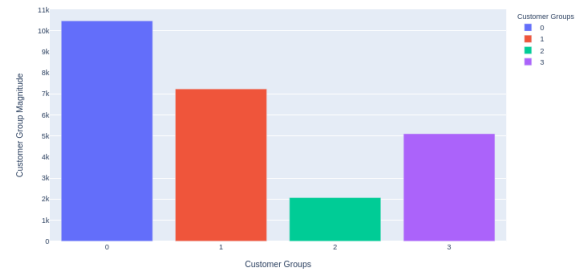


Fig. 18. Cluster Magnitudes

V. RESULTS

The optimal number of customer groups were found out to be 4 in this case, which may vary depending on the datasets. The four customer groups found had varying characteristics and are to be dealt with accordingly.

VI. ANALYSIS & DISCUSSION

A single dataset comprising of 25000 customers was taken into account for this experiment. The data set was void of NULL values. Three primary features were selected to work upon. After selection of the features, the final version of pre-processed data was fed and the datasets with the most relevant features were segregated and taken into account.

On taking a further look at the cluster magnitudes,

- Blue group comprises of 42% of total customers, marking a huge share. Any further improvements made in this category would indeed contribute to the high increase in revenue.
- Red and purple group together comprise of 50% of total customers. They have a maximum return rate of 0.5.
- Green group consists of 8% of the total customers, they order multiple products and they are highly likely to keep them. It is the most favourable group for the company. To maintain and possibly expand this group, special deals and pre-product launches might help. Moreover, they can play a huge role in impacting the new audience, contribute significantly for the expansion of the customer base.

VII. CONCLUSION

The customer segmentation model was approached from a behavioural point of view, and the primary features taken into account to achieve this effect were number of products ordered, average return rate and total spending for each customer.

The dataset initially was random. It could not differentiate into potential customer sweet spots or categorize them into any certain groups. On applying the k-means clustering the dataset could be categorized into various groups and accordingly the outcomes of target specific audience could be achieved.

VIII. FUTURE WORK

The standard k-means algorithm was implemented for clustering in this article. To further reduce clustering errors, a modification of the k-means algorithm could be used.

Implementing this project in the real-world would further ensure consistency and practical application of the model. The future research can be directed in this aspect.

Maliciousness level of customers could be determined using data mining methods such as association rules and the rules extraction.

IX. INDIVIDUAL CONTRIBUTION

Name: Akshay Sreekumar Nair

Contributions,

- Collect Dataset
- Design model
- Pre-processing Dataset
- Scaling
- Implement k-means algorithm
- Hyper-parameter tuning
- Visualization and interpretation of results obtained
- Prepare Report
- Presentation work

ACKNOWLEDGMENT

I would like to express my sincere gratitude towards Professor Ram Mohana Reddy Guddeti for his continued support and guidance in making this work possible.

REFERENCES

- [1] [Zha+21]
An Extended Regularized K-Means Clustering Approach for High-Dimensional Customer Segmentation With Correlated Variables.
Author image of Hong-Hao Zhao Hong-Hao Zhao, Xi-Chun Luo, Rui Ma
<https://ieeexplore.ieee.org/document/9381869>
- [2] [Abd21]
Incorporating K-means, Hierarchical Clustering and PCA in Customer Segmentation.
Azad Abdulhafedh
https://www.researchgate.net/publication/349094412_Incorporating_K-means_Hierarchical_Clustering_and_PCA_in_Customer_Segmentation
- [3] [ZE20]
Determination of Customer Satisfaction using Improved K-means algorithm.
Hamed Zare, Sima Emadi
https://www.researchgate.net/publication/341745185_Determination_of_Customer_Satisfaction_using_Improved_K-means_algorithm