Segmenting Customers using K-Means, RFM and Transaction Records

Adeline Ong
Mar 16 · 6 min read

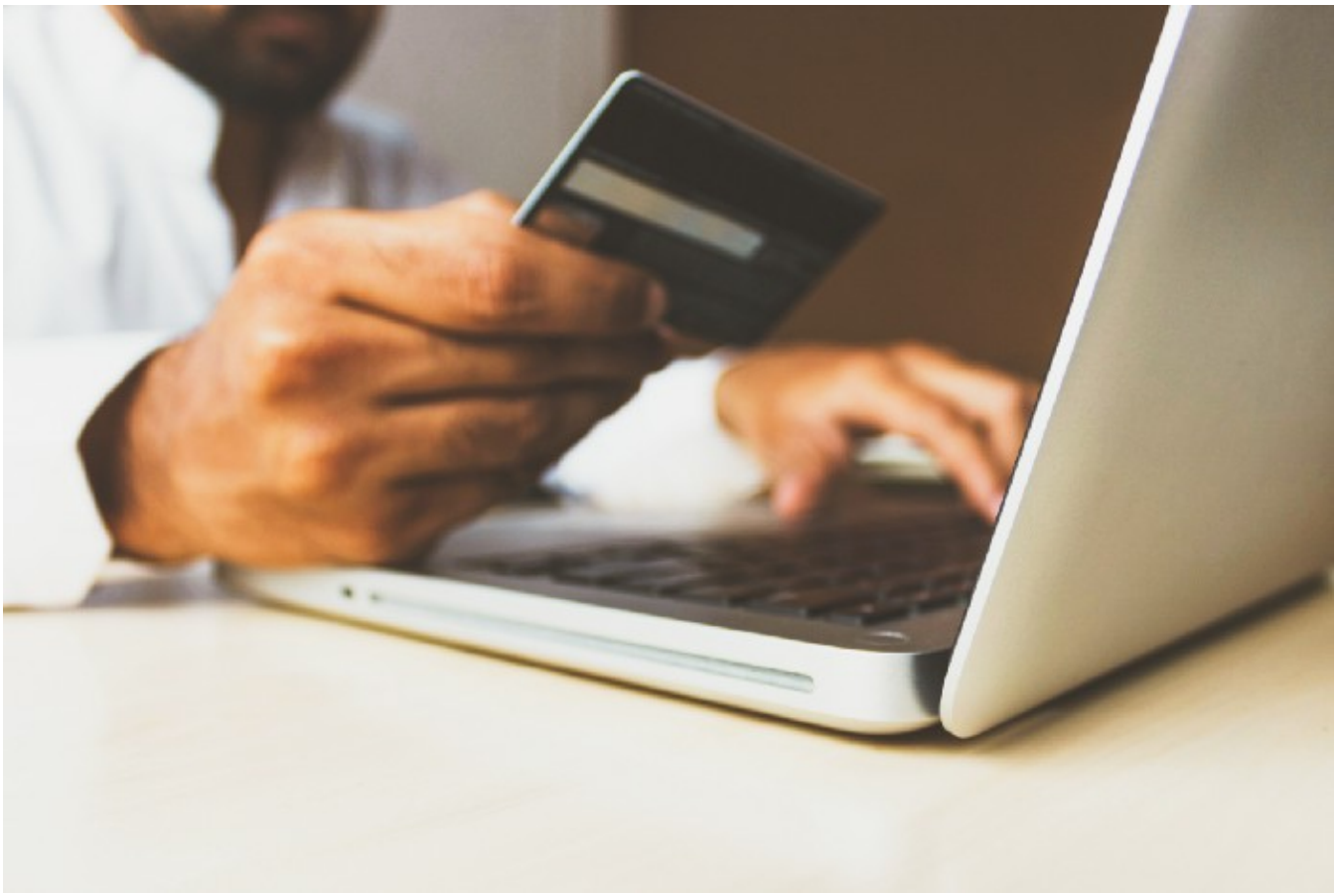

Photo by rupixen.com on Unsplash

In this article, I will walk you through how I applied K-means and RFM segmentation to cluster online gift shop customers based on their transaction records.

Introduction

When I was in college, I started a simple e-store selling pet products. Back then, I only collected enough customer information to make the sale, and get my products to them.

Simply put, I only had their transaction records and addresses.

Back then, I didn't think I had enough information to perform any useful segmentation. However, I recently came across an intuitive segmentation approach called **RFM** (**R**ecency **F**requency **M**onetary Value), which can be easily applied to basic customer transaction records.

About RFM Segmentation

Here's what each letter of **RFM** means:

- **R**ecency: How long has it been since the customer last purchased from you (e.g. in days, in months)?
- **F**requency: How many times has the customer purchased from you within a fixed period (e.g. past 3 months, past year)
- **M**onetary Value: How much has the customer spent at your store within a fixed period (which should be the same period set for **F**requency).

We can group customers, and come up with business recommendations based on RFM scores. For example, you could offer promotions to reengage customers who have not bought from your store recently. You could further prioritize your promotional strategy by focusing on customers who used to buy frequently and spend at least average monetary value.

Using K-Means Instead of the Traditional Approach

The traditional RFM approach requires you to manually rank customers from 1 to 5 on each of their RFM features. Two ways to define ranks would be to create groups of equal intervals (e.g. range/5), or categorize them based on percentiles (those up to 20th percentile would form a rank).

Since we are data scientists, why not use an unsupervised learning model to do the job? In fact, our model might perform better than the traditional approach since it groups customers based on their RFM values, instead of their ranking.

The Data

The dataset was from UCL's machine learning repository. The file contained 1 million customer transaction records for a UK-based online gift store for the period between

2009 to 2010, and 2010 to 2011. There were two sheets in the excel file (one for each year), and each sheet had the same 8 features:

- Customer ID
- Country (I didn't really look at this since most customers were UK-based as well)
- Invoice Code
- Invoice Date
- Stock Code
- Stock Description
- Unit Price
- Unit Quantity

Data Cleaning

Since both datasets contained the same features, I appended one to the other. Following this, I dropped rows that had:

- Missing Customer ID
- Missing Stock Description
- Abnormal Stock Codes that did not conform to the expected format, such as Stock Codes that started with letters, and had less than 5 digits. These tended to be from manual entries (Stock Code 'M'), postage costs (Stock Code 'DOT') and cancelled orders (Stock Codes starting with 'C'). However, I retained Stock Codes that ended with letters, as these tended to indicate product variations (e.g. pattern, color).

After creating RFM features for each customer (see Feature Engineering), I also removed extreme outliers that were more than 4 standard deviations away from the mean. Removing extreme outliers is important because they can skew unsupervised learning models that use distance-based measures.

Feature Engineering

To derive a customer's Recency, I calculated the time difference (in days) between the latest purchase in the combined dataset, and the customer's last purchase. Lower scores indicate a more recent purchase, which is better for the store.

I created features that corresponded to each customer's frequency of purchase (over the 2 year period) and total spend (Monetary Value) through aggregation:

- **Frequency**: Count the number of unique Invoice Codes per customer
- **Monetary Value**: Sum the price of all items purchased

I also created other features, which I thought would be useful cluster descriptors:

- Total spend per invoice
- Time (in days) between orders

Choosing an Unsupervised Learning Model using Silhouette Score

Silhouette score can be used to evaluate the quality of unsupervised learning models where the ground truth is unknown. Silhouette score measures how similar an observation is to its own cluster, as compared to other clusters.

Values closer to 1 indicate better cluster separation, while values near 0 indicate overlapping clusters. Avoid values that are negative.

I applied 3 unsupervised learning models to the data, and chose go with K-Means because it had the best silhouette scores regardless of the number of clusters.

Silhouette scores of unsupervised learning models by number of clusters

Choosing the Number of K-Means Clusters

To choose the number of clusters (n_clusters), I took into account each cluster's silhouette score. Optimally, every cluster's coefficient value should be higher than the mean silhouette score (in the graph, each cluster's peak should exceed the red dotted line). I also took into account the RFM values of each cluster.

I differed the number of K-Means clusters and examined the RFM values and silhouette scores of the models. I decided to go with n_clusters =5 instead of anything less despite a lower silhouette score because an important customer segment that had good RFM values only appeared when n_clusters = 5. Clusters that appeared beyond n_clusters = 5 were less critical because they had poorer RFM scores.

Table depicting silhouette scores across n number of clusters, and whether each cluster's coefficient value was higher than the mean silhouette score within each model

Visualizing and Describing the Clusters

Having choose a unsupervised learning model, and a suitable number of clusters, I visualized the clusters using a 3D plot.

3D plot depicting customer segments derived using RFM segmentation and K-Means

Clusters 4 and 2 have better RFM scores and represent the store's core customers. The other 3 clusters appear to be more causal customers who purchase less frequently.

Core Customers

Based on this dataset, 18% of customers are core customer, and they contributed to 62% of revenue for the past two years. They spend a lot, purchase frequently, every one or two months, and are still engaged with the store. As the typical price of the online store's products tends to be low, the clusters' average spend suggest that they are purchasing in large quantities, so they are probably wholesalers and smaller shops that resell the store's goods.

Table describing key features of core customers. Non-percentage figures represent averages.

Casual Customers

As for casual customers, I'd like to highlight Cluster 0 (which I've called Gift Hunters) as they are most critical to the store. They contributed to about a quarter of revenue, which is a lot more than the other casual clusters. They tended to purchase from the store once every quarter in small amounts, which suggest that they are individuals buying for special occasions.

Table describing the key features of casual customers. Non-percentage figures represent averages.

Possible Promotional Strategies to Pursue

Given the features of the clusters, I propose the following promotional strategies for key groups:

- **Wholesalers**: Given their small numbers, it might make sense to engage them directly to build goodwill and loyalty. It would be best to lock them in with a custom solution.
- **Small Shops**: Explore cashback discounts that can be used during subsequent purchases. This will also lower their cost and encourage them to spend more.

- **Gift Hunters**: Engage them just before special occasions and encourage them to spend more by giving them free gifts for a minimum spend that is higher than their current mean spend of 347 pounds.

To End Off...

I think RFM segmentation pairs very well with unsupervised learning models, as they remove the need for marketers to manually segment their customer records. I hope I've illustrated how meaningful customer segments can be created from very basic customer information. For more details, you can look at my notebook. It contains code and details about the other models that I explored.

Thanks for reading!
