ONLINE RETAIL CUSTOMER SEGMENTATION

Sushwet Kumar Pandey, Data Science Trainee, AlmaBetter, Bangalore

INTRODUCTION:

Customer segmentation is the process of classifying customers into groups based on their shared behaviour or other attributes. The groups should be homogeneous within them and should also be heterogeneous to each other. The main goal is to identify customers that are most profitable and loyal and the ones who churned out, to prevent further loss of customer by redefining company policies. Having large number of customers, each with different needs it is difficult to find which customer is most important for business and target them with appropriate strategy.

PROBLEM STATEMENT:

Identify major customer segments on a transnational dataset which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail.

DATA DESCRIPTION:

• **InvoiceNo:** Nominal, 6-digit integral number uniquely assigned to each transaction.

• **StockCode:** Nominal, 5-digit integral number uniquely assigned to each distinct product.

• **Description:** Nominal, product(item) name.

• **Quantity:** Numeric, quantities of each product per transaction.

• **InvoiceDate:** Numeric, the day and time when each transaction was generated.

• **UnitPrice:** Numeric, product price per unit in sterling.

• **CustomerID:** Nominal, 5-digit integral number uniquely assigned to each customer.

• **Country:** Nominal, name of the country where each customer resides.

DATA PREPARATION:

INSPECTING DATASET:

1. Importing some useful libraries like pandas, matplotlib, seaborn, and NumPy.
2. Importing the dataset and checking the top 5 and last 5 rows to get an overall idea.
3. Coming to some exploration of the dataset, by checking the info() which give us some intuition about null

values as well as data types of columns present in it.

4. Now we move towards descriptive summary of the dataframe which give some quantitative idea about the dataset i.e., average values of the columns, frequency of the values, variability and dispersion concerns on how spread out the values are.

FEATURE ENGINEERING:

Feature engineering is the pre-processing step of machine learning, which is used to transform raw data into features that can be used for creating a predictive model using Machine learning or statistical Modelling. Feature engineering is finding the useful variables to be used in a predictive model. The new features are created by mixing existing features using addition, subtraction, ratios, and these new features have great flexibility.

Data cleaning: There are features which have some null entries corresponding to it, which we need to drop or treat by some technique like fill method.

Dropping duplicates: In the dataset there are some observations which are kindly as same as the previous or next entries, so this needs to be dropped.

One hot encoding: One hot encoding is the popular encoding technique in machine learning. It is a technique that converts the categorical data in a form so that they can be easily understood by machine learning algorithms and hence can make a good prediction.

EXPLORATORY DATA ANALYSIS:

Exploratory Data Analysis or (EDA) is understanding the data sets by summarizing their main characteristics often plotting them visually. This step is very important especially when we arrive at modelling the data in order to apply Machine learning. Plotting in EDA consists of Histograms, Box plot, Scatter plot and many more. It often takes much time to explore the data. Through the process of EDA, we can ask to define the problem statement or definition on our data set which is very important.

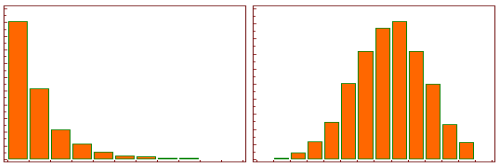
For data analysis, Exploratory Data Analysis (EDA) must be our first step. Exploratory Data Analysis helps us to –

- To give insight into a data set.
- Understand the underlying structure.
- Extract important parameters and relationships that data holds between them.
- Test underlying assumptions

LOG TRANSFORMATION:

Log transformation is a data transformation method in which it replaces each variable x with a log(x). The choice of the algorithm base is usually left up to the analyst and it would depend on the

purposes of statistical modelling. In this article, we will focus on the natural log transformation. The nature log is denoted as ln. When our original continuous data do not follow the bell curve, we can log transform this data to make it as "normal" as possible so that the statistical analysis results from this data become more valid. In other words, the log transformation reduces or removes the skewness of our original data. The important caveat here is that the original data has to approximately follow a log-normal distribution. Otherwise, the log transformation won't work.



RECENCY, FREQUENCY, MONETARY MODEL(RFM):

RFM segmentation allows marketers to target specific clusters of customers with communications that are much more relevant for their particular behaviour and thus generate much higher rates of response, plus increased loyalty and customer lifetime value. Like other segmentation methods, RFM segmentation is a powerful way to identify groups of customers for special treatment. RFM stands for recency, frequency and monetary.

- **Recency**: How much time has elapsed since a customer's last activity or transaction with the brand?
- **Frequency**: How often has a customer transacted or interacted with the brand during a particular period of time?
- **Monetary**: Also referred to as "monetary value," this factor reflects how much a customer has spent with the brand during a particular period of time.

Interpretation

If the RFM of any customer is 444. His Recency is good, frequency is more and Monetary is more. So, he is a customer who spends well and is frequent.

If the RFM of any customer is 111. His Recency is low, frequency is low and Monetary is low. So, this customer is neither loyal nor an exclusive customer.

If the RFM of any customer is 144. His purchased a long time ago but buys frequently and spends more. And so on.

Like this we can come up with number of segments for all combinations of R,F and M based on our use case. Higher the RFM score, more valuable the customer is.

STANDARD SCALING VARIABLES:

Standard Scaler helps to get standardized distribution, with a zero mean and standard deviation of one (unit variance). It standardizes features by subtracting the mean value from the feature and then

dividing the result by feature standard deviation.

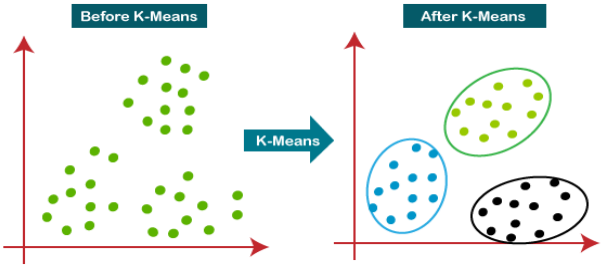
- The fit(data) method is used to compute the mean and std dev for a given feature so that it can be used further for scaling.
- The transform(data) method is used to perform scaling using mean and std dev calculated using the. fit() method.
- The fit_transform() method does both fit and transform.

QUANTILE BASED CLUSTERING:

Now we are introducing K-quantiles clustering. K-quantiles clustering can be computed by a simple greedy algorithm in the style of the classical Lloyd's algorithm for K-means. It can be applied to large and high-dimensional dataset. It allows for within-cluster skewness and internal variable scaling based on within-cluster variation. Different versions allow for different levels of parsimony and computational efficiency. Although K-quantiles clustering is conceived as nonparametric, it can be connected to a fixed partition model of generalized asymmetric Laplace-distributions. The consistency of K-quantiles clustering is proved, and it is shown that K-quantiles clusters correspond to well separated mixture components in a non-parametric mixture. In a simulation, K-quantiles clustering is compared with a number of popular clustering methods with good results. A high-dimensional micro array dataset is clustered by K-quantiles.

K-MEANS CLUSTERING:

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabelled dataset into different clusters. Here K defines the number of pre-defined



clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

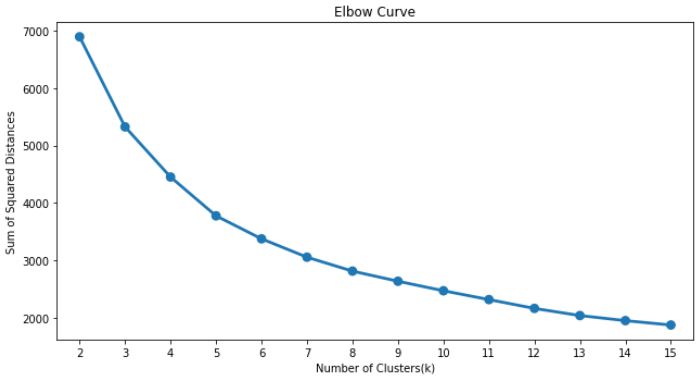
“It is an iterative algorithm that divides the unlabelled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.”

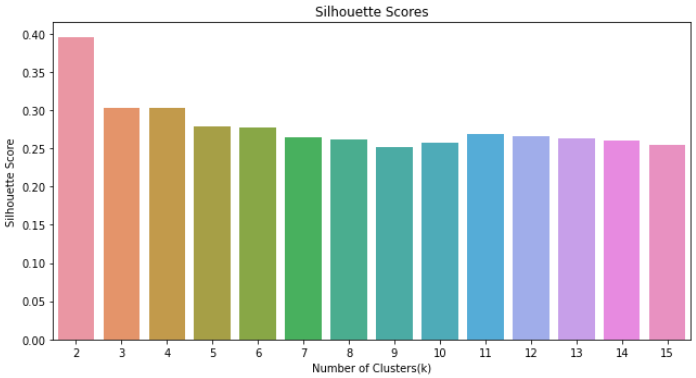
Here we applied two techniques to optimise the best value of K.

ELBOW METHOD FOR OPTIMAL VALUE OF K IN K-MEANS:

A fundamental step for any unsupervised algorithm is to determine the optimal number of clusters into which the data may be clustered. The Elbow Method is one of the most popular methods to determine this optimal value of k.

We now define the following: -





Distortion: It is calculated as the average of the squared distances from the cluster centres of the respective clusters. Typically, the Euclidean distance metric is used.

Inertia: It is the sum of squared distances of samples to their closest cluster centre.

We iterate the values of k from 1 to 15 and calculate the values of distortions for each value of k and calculate the distortion and inertia for each value of k in the given range.

SILHOUETTE SCORE:

Silhouette Coefficient or silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1.

1: Means clusters are well apart from each other and clearly distinguished.

0: Means clusters are indifferent, or we can say that the distance between clusters are not significant.

-1: Means clusters are assigned in the wrong way.

Silhouette Score = (b-a)/max(a,b) where

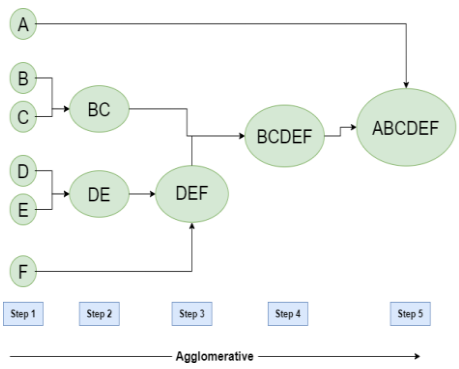
a= average intra-cluster distance i.e. the average distance between each point within a cluster, b= average inter-cluster distance i.e. the average distance between all clusters.

HIERARCHICAL CLUSTERING:

A Hierarchical clustering method works via grouping data into a tree of clusters. Hierarchical clustering begins by treating every data point as a separate cluster. Then, it repeatedly executes the subsequent steps:

In Hierarchical Clustering, the aim is to produce a hierarchical series of nested clusters. A diagram called Dendrogram (A Dendrogram is a tree-like diagram that statistics the sequences of merges or splits) graphically represents this hierarchy and is an inverted tree that describes the order in which factors are merged (bottom-up view) or cluster are break up (top-down view).

The basic method to generate hierarchical clustering are:



1.

1. **Agglomerative**:

Initially consider every data point as an individual Cluster and at every step, merge the nearest pairs of the cluster. (It is a bottom-up method). At first every data set is considered as individual entity or cluster. At every iteration, the clusters merge with different clusters until one cluster is formed.

2. **Divisive**:

We can say that the Divisive Hierarchical clustering is precisely the opposite of the Agglomerative Hierarchical clustering. In Divisive Hierarchical clustering, we take into account all of the data points as a single cluster and in every iteration, we separate the data points from the clusters which aren't comparable. In the end, we are left with N clusters.

DBSCAN CLUSTERING :

Clustering analysis is basically an Unsupervised learning method that divides the data points into a number of groups, such that the data points in the same groups have similar properties and data points in different groups have different properties in some sense.

Fundamentally, all clustering methods use the same approach i.e., first we calculate similarities and then we use it to cluster the data points into groups or batches. Here we will focus on **Density-based spatial clustering of applications with noise (DBSCAN)** clustering method.

Clusters are dense regions in the data space, separated by regions of the lower density of points. The DBSCAN algorithm is based on this intuitive notion of "clusters" and "noise". The key idea is that for each point of a cluster, the neighbourhood of a given radius has to contain at least a minimum number of points.

DBSCAN algorithm requires two parameters:

eps: It defines the neighbourhood around a data point i.e., if the distance between two points is lower or equal to 'eps' then they are considered as neighbours. If the eps value is chosen too small then large part of the data will be considered as outliers. If it is chosen very large then the clusters will merge and majority of the data points will be in the same clusters. One way to find the eps value is based on the k-distance graph.

MinPts: Minimum number of neighbours (data points) within eps radius. Larger the dataset, the larger value of MinPts must be chosen. As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1. The minimum value of MinPts must be chosen at least 3.

CONCLUSION:

We started with a quantile based simple segmentation model first then moved to more complex models because simple implementation helps having a first glance

at the data and know where/how to exploit it better.

Then we moved to k-means clustering and visualized the results with different number of clusters. As we know there is no assurance that k-means will lead to the global best solution. We moved forward and tried Hierarchical Clustering and DBSCAN clustering as well.

We didn't obtain a clearly separated clusters as the Cluster assignments are muddled.

Segments depends on how the business owners want plans to use the results, and the level of granularity they want to see in the clusters. Based on that, various methods of clustering can be further applied be it on RFM variables or directly on the transactional dataset.

Keeping these points in view we clustered the major segments based on our understanding as per different criteria as shown below in the summary data frame.

S.No.	CLUSTERER	CRITERION	SEGMENTS
1	QUANTILE BASED	Quantile	4
2	K-MEANS	Elbow	5
3	K-MEANS	Silhouette	2
4	K-MEANS	Elbow and silhouette	4
5	AGGLOMERATIVE	Dendogram(y=70)	2
6	AGGLOMERATIVE	Dendogram(y=50)	3
7	DBSCAN	nan	3