Implementation of a Visual Analytics dashboard for Sales Analysis and Customer Segmentaton in Retail

Visual Analytics 2020/2021 - Alessandra Monaco (1706205)

1 Introduction

In Marketing Supply Chain, being a retailer involves a lot of decision making. Retail industries buy goods from manufacturers (the producers) or wholesalers (that sell goods in large quantities) and sell them to end customers with higher prices. In order to maximize their profits, a key point is to decide which products to buy from the manufacturers/wholesalers, trying to predict how much profitable they could be: they have to understand the market, in other words, they have to understand the customers' needs. Moreover, advertisement plays an important role to increase sales, especially in online retail, but it requires choosing which products to promote, trying to understand which ones are more likely to capture the customers' interests. However, different customers may have different necessities, tastes, and they can also have a different importance for the decision making processes according to their purchase behavior: a customer that spends a lot of money and purchases often is more "valuable" than a customer that purchased just few times. Therefore, customer tastes are not the only focus during decision making: many other aspects should be taken into account, such as loyalty, retention, monetary value.
In this setting, a well-known strategy is *Customer Segmentation* (aka *Market Segmentation*), that aims to split customers in discrete subgroups with similar characteristics. The characteristics that are taken into account in the segmentation may vary between different retail industries: each retailer can adopt its own model, however they all share the need of tools that allow for deep and detailed analyses of huge amounts of data. In this research we propose the use of Visual Analytics for a fast and interactive market segmentation and analysis for a multiple retailer use case.
Section 2 provides an overview of past works on sales analysis and segmentation, and describes similar solutions to the proposed one. **Section 3** is a detailed description of the dataset used. **Section 4** describes the system from a front-end and back-end point of view, providing details about both the analytic and the visualization, and providing examples of possible analyses. Finally, **Section 5** concludes the research defining the main advantages of Market Analysis, Segmentation and Visual Analytics.

2 Related Works

A recent implementation of a sales analysis dashboard has been proposed by Ricky Akbar (2020). Their work involves the entire Business Intelligence (BI) life-cycle, including the design of a Data Warehouse (DW), the implementation of Extract-Transform-Load pipelines, and the creation of a visualization system for analyses using Tableau Software. This last part of the project is a starting point for our work, being useful to understand what kind of sales information can be interesting and important to analyze. They propose a set of interactive visualizations, in particular:

1. bubble-size chart : it analyzes sales by category; each product category is a bubble and its size encodes the total number of sales for that product category;
2. sales by brand bar graph : for each brand the length of the bar encodes the total number of sales for that brand;
3. total sales line chart (drill-down) : a standard line chart is empowered with the capability of changing the level of detail, namely the line chart can aggregate by date, by month, by year;
4. total payment of sales line chart (drill-down) : same as previous, but considering on y axis the daily/-monthly/yearly profit instead of the total number of sales;

5. total sales bar chart + profit line chart by week-day : for each day of the week the length of the correspondent bar encodes the total number of sales on that weekday for the entire time-interval considered in the database, while the position of the point in the x-axis of the line-chart encodes the total payment of sales;
6. most expensive product sold for each type (including product price);
7. top most sold products, with the correspondent number of sales.

We customized some of the ideas below to create our own dashboard in d3.js. First, given that the products in our dataset are organized in categories and subcategories, we exploit a more hierarchical representation, using a treemap instead of the bubble-size chart (1) to analyze sales by categories and subcategories. As in (3) and (4), we plot a line chart to display profits and sales, but using a single line chart and a data selector to choose whether to visualize sales or profits, and an aggregation selector to choose the level of aggregation, trying to simulate the drill-down operations that are typical of BI and DW. Lastly, (5) was an inspiration for our calendar heatmap, in which we do not only analyze sales by week-day as they did, but also analyze monthly/seasonal patterns. We do not analyze brands or individual products because we do not have such information in our dataset.

Ricky Akbar (2020) work was a good starting point, but it is not enough for our purposes: our goal is to extend this kind of sales analysis to customer segments, requiring a customer segmentation methodology and a segment-based filtering functionality.
Regarding the first need, different types of segmentation have been proposed during time. As Ron Kohavi (2004) suggests in *"Visualizing RFM Segmentation"*, RFM is a very old technique that has been used for more than 50 years by marketers, to help them to visualize and quickly identify important customer segments. Customers are splitted in bins basing on 3 behavioral attributes: Recency of purchase, Frequency of purchase and Monetary value of purchase. The goal of this approach is to identify customers that are more likely to buy again in the future, basing on some simple observations:

- customers that have purchased recently are more likely to buy again;
- frequent buyers are more likely to purchase again than unfrequent customers;
- big spenders respond better to messages.

The paper's approach follows Arthur Huges RFM analysis, ranking each behavioral attribute into five ranges, but binning by threshold and not by equal frequency (as Arthur Huges suggests). Since looking at the 3-digit number of the combined scores (from 111 to 555) is not enough for a satisfying analysis, the authors suggest possible visualizations than can give more insights about the segments, such as RFM heatmap, RF 2d scatterplot where segments are squares, colors encode the average monetary value and size encodes the number of customers in the segment; also RFM 3d scatterplot has been proposed, and histograms to analyze demographic information of segments.

Recently, with the development of Machine Learning algorithms, new segmentation approaches have been proposed. A possibility is the use of unsupervised clustering algorithms to split customers into groups with similar characteristics. What exactly are these characteristics depends on the features that are given as input.
Hossain (2017) in *"Customer Segmentation using Centroid Based and Density Based Clustering Algorithms"* applies two different clustering algorithms for market segmentation, considering as customers' features the annual spending on 8 different items. The research shows that both the density-based DBSCAN and centroid-based K-Means were able to create meaningful clusters, and DBSCAN was also good in detecting outliers, namely anomalous customers having different spending habits. No visualization has been proposed by this paper for cluster analysis.

A combined approach of the previous was proposed by Rahul Shirole (2021) in *"Customer Segmentation using RFM Model and K-Means Clustering"*. The idea is that, instead of considering 5x5x5 distinct segments, or grouping by combined scores as in Ron Kohavi (2004) research, K-Means takes as features the RFM scores of each customer. Setting the number of clusters to 4, the authors were able to segment the customers into Class A, Class B, Class C and Class D, such that Class A generates the highest revenue, and Class D the lowest.

There are plethora of other possible approaches to segmentation, such as Neural Networks, business rule-based, Self-organizing map or supervised techniques. Different techniques may provide different

insights, that can be complementary, so there is not always a "best" technique, it depends on business goals and data. We chose to provide two distinct and complementary types of segmentations: (i) rfm-based, inspired by Ron Kohavi (2004), but applying equal-frequency binning instead of threshold-based, and using the RFM heatmap proposed by them, and (ii) unsupervised segmentation based on K-Means, inspired by Hossain (2017), but performed on the Principal Components and not on raw data.
For this last computation, cluster visualization and interpretation may be critical if the dataset is large and multidimensional.
When the dataset is multidimensional, a reasonable technique is the use of parallel coordinates. Julian Heinrich (2015) in *"Big Data Visual Analytics with Parallel Coordinates"* shows the use of parallel coordinates together with advanced analytics such as clustering (K-Means) and dimensionality reduction (PCA) to discover structure in the data. Dealing with Big Data, however, can produce a lot of clutter in the visualization due to the high density of the lines. To improve the visibility of the chart, we took their suggestion to color the lines according to the clustering results: in this way we are also able to understand which features mostly characterize clusters and understand the created groups.
In the next sections we will describe the mentioned visualizations in details.

3 Dataset

The dataset used in this project is from Kaggle[1] and describes sales of a fictional multiple retailer[2]. It includes 3 csv files: *Customers.csv* stores demographic information about customers, such as id, day of birth, gender and the code of the city in which he is located; *prod_cat_info.csv* stores information regarding inclusion relationships between product categories; *Transactions.csv* stores purchases made by customers for certain product categories, including the product quantity, the date, the total amount spent, the store type and other less relevant information. The dataset contains 6 product macro-categories (Bags, Books, Clothing, Electronics, Footwear, Home and Kitchen), 23 053 transactions and 5 647 customers (but only 5 506 of them have at least one associated transaction).
To limit file access and build a quickly responsive system, the original dataset is merged into a single (and more redundant) csv file (*full_data.csv*) with shape (23 053, 15), storing the following information: transaction id, customer id, transaction date, code and name of the product subcategory, code and name of the product category, product quantity, rate, tax, total amount spent in this transaction, store type, customer day of birth, customer gender, customer city code.

4 System Implementation

A front-end dashboard has been implemented in d3.js framework. The server is created using Flask python package, and analytics run in python, exploiting some well-known data science libraries, such as Scikit-learn, Pandas and Numpy.

4.1 Analytics

4.1.1 RFM Segmentation

The RFM Segmentation groups customers basing on 3 metrics, defined as follows:
- **Recency** : number of days that have been passed since last purchase;
- **Frequency** : total number of purchases until most recent date in the dataset;
- **Monetary** : total amount of money spent by the customer.

The creation of the segments needs a scoring method to group the R, F, M values. A possibility is to use thresholds, building if-then rules (ex: if R value < 10, then it gets assigned a score R=1 and so on..), but this may require to assess and adjust thresholds when we repeat the segmentation after some time, and there is also the risk that some segments remain empty (no customer falls into them), which is not what we want. A more standard and easier approach is to use **quantile-based scoring** (aka percenteil-based grouping, or

[1] https://www.kaggle.com/darpan25bajaj/retail-case-study-data

[2] **Multiple Retailing**, often referred to as "Multi-channel Retailing", is an approach in retail business in which products are offered to customers through multiple retail channels (internet, physical stores, ...). All the channels share the ownership and management.

equal-frequency binning), such that, for each metric, the customers are grouped into equal-size groups. The steps to implement this approach are quite easy and intuitive: for each metric (R, F, M), we (i) sort customers from worst to best value[3], then (ii) we choose the number of groups that we want to obtain[4] (for our dataset the best choice was to use quartiles, because there were not too many customers), and finally (iii) we break the customers into the selected number of groups such that each group has the same size, assigning labels from 1 (worst) to 4 (best).
At the end of this process we end up with 4x4x4=64 distinct segments, making the analysis quite difficult. We will see how visualization techniques can help in the correspondent section.
To make the segmentation more interesting, for each customer we also compute a total score by summing R,F and M scores (ex: if R=2, F=3, M=4, total score=2+3+4=9), and we assign threshold-based labels:

- if total score >= 9, then label="Gold";
- if 8 <= total score < 9, then label="Can't loose them";
- if 7 <= total score < 8, then label="Loyal";
- if 6 <= total score < 7, then label="Potential";
- if 5 <= total score < 6, then label="Promising";
- if 4 <= total score < 5, then label="Needs Attention";
- if total score < 4, then label="Require Activation";

4.1.2 Unsupervised Segmentation

Unsupervised Segmentation aims to automatically split customers into groups with similar interests, basing on purchase behaviors.
In practice, for each customer, we create a vector such that each component is the number of purchases made by that customer for products belonging to a given subcategory, so every feature vector has a length equal to the total number of subcategories (23). Features are normalized with a Standard Scaler and Principal Component Analysis is applied, then the vectors are clustered with K-Means algorithm. Applying PCA before a clustering algorithm like the K-Means may help to produce much clearer and defined clusters. Moreover, reducing the dimensionality of the data can improve K-Means efficiency, speeding up the computation. The number of principal components and the number of clusters can be selected by the end-user, with the limit of 2-12 components and 2-8 clusters.

4.2 Visualizations

Figure 1 shows the implemented dashboard. In the next subsections we will analyze each visualization independently.

External Filters (a)

On the top of the dashboard, the user can select the data on which he wants to run the analytical computations (rfm analysis and unsupervised clustering). It is possible to choose a gender, a city and store type. For instance, a retailer may want to offer a discount only to most proficient women for some reason, and perform the analytics on female gender only; or, a retailer may want to open a new shop near a certain city, so it may be useful to analyze interests of customers that live near that city; or, he may want to analyze only the purchases of the tele-shop, to understand customer needs and use the generated insights for better tele-advertisements. Another possibile scenario is when a retailer needs a better warehouse organization and wants to understand what kind of products are better to keep in the MBR shop because they are better aligned with the needs of customers that purchased in that shop.
Notice that these filters are all applied on the original dataset before running the computations (this is why they are called "external"). A reset button is used to clear all selections on all the views and reset the original visualization.

[3] This means that frequency and monetary values are sorted in ascending order, while recency is sorted in descending order, since customers that have purchased more recently (few days before=>low recency) are "better" than customers that have purchased a long time ago (high recency).

[4] In other words we choose if we want quartile (scores 1-4), quintile (scores 1-5) and so on.

Purchase Trends (b),(c),(d)

As the title suggests, this part of the dashboard is focused on analyzing purchase trends.
View (**a**) helps to understand if the sales trend is increasing, decreasing, stable, unstable or fluctuating during time, if there are particularly profitable days (peaks), or particularly bad days/periods. The linechart is a very standard visualization technique for analyzing trends in time-series data, but still very effective. The user can choose to display profits (i.e. amount of money purchased by customers each day) or sales (i.e. number of products bought by customers each day). There is also the possibility to aggregate the total profits/sales by day, by month or by year. Since it may be hard to understand the precise values of the x (date) and y (profit/sales) axes from the chart, the precise values of the coordinates are displayed when the mouse goes over a data point. To make the visual analysis easier even when a lot of data are displayed, it is possible to zoom with the mouse on a particular time period; as a result of this interaction, view (**c**) and (**d**) will be filtered, too.
View (**c**) displays many aggregating information that can help the retailer to understand how proficient is the current selection[5], computing the total profits, the total number of sales and the average daily profit.
Finally, view (**d**) displays the number of sales for each subcategory in a treemap, that is built basing on the inclusion relationships between subcategories and categories. The colour encodes the category that the subcategory belongs to, the size of the rects encodes the total number of sales. With such a view it will be much easier and quick for the end-user to understand the categorical organization of products and detect if some category/subcategory is purchased more than the others. Numerical details and labels are displayed too for more precise information. To deal with space limitation, if the rectangle is not large enough to handle label text and numerical information, those details can be seen on a tooltip that is displayed on mouse over the rectangle. Subcategory rectangles can be selected/deselected to filter the visualizations (**b**), (**c**), (**e**).

Purchase Seasonalities (e)

The calendar heatmap is another well-known visualization that can help to quickly visualize seasonalities. A calendar heatmap uses colored cells, typically in a monochromatic scale, to show number of events (profits/sales) for each day in a calendar view. Days are arranged into columns by week and grouped by month and years; week-days, months and years are labeled. This enables to quickly identify patterns and answers to questions like "Are there particular week-days or months in which customers generally buy more?", "Are there patterns that are repeated over the years/months?".

Unsupervised Customer Segmentation (f),(g),(h),(i),(j)

This part of the dashboard is focused on unsupervised clusters visualization. Selector in (**f**) allows the end-user to select the number of principal components to perform PCA and the number of clusters to run the K-Means algorithm. After the computation, some important metrics are reported in table (**f**): silhouette score[6], inertia[7] and percentage of total variance explained[8] by PCA. Notice that, in clustering settings, metrics are not enough to understand which are the best hyperparameters (number of components and number of clusters): you also have to understand if the created clusters are meaningful. This is why we do not simply choose the n component and n clusters that give highest silhouette, but let the user look at the metrics and analyze the clusters with all the provided visualizations to adjust the model hyperparameters and trigger the computation.
Dealing with multidimensional data requires appropriate visualizations: parallel coordinates are often a good choice, but if the dataset is too large the visualization can suffer from cluttering due to overplotting, and the result may be too much confusing to derive insights. We took the suggestion of Julian Heinrich (2015), colouring the lines according to the cluster they belong to but, differently from the paper, we plot only the

[5] For "selection" we mean a particular time period, or a particular set of subcategories, or a particular cluster, or a particular rfm segment.

[6] The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from 1 to +1 and is defined as 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.

[7] Inertia is the sum of squares for all dataset points to their closest centroid.

[8] The total variance explained is the sum of the variance explained by each component, in percentage. The fraction of variance explained by a principal component is the ratio between the variance of that principal component and the total variance.

original dimensions (the authors of the paper plot also cluster label, PCA 1, PCA 2 as dimensions of the parallel coordinates), because our dataset already has a lot of dimensions and this can add confusion. We plotted 2 PCA components in (**h**), instead.
Since the order of the axes impacts how the reader understands the data, we give to the user the possibility to randomly shuffle the order of the dimensions, looking for patterns. At the beginning, dimensions are sorted according to their category and subcategory. Categories are sorted alphabetically. Axes labels are truncated (they are too long), but the full label can be seen on a tooltip on mouse over. The tooltip background colour respects the category colour of the treemap.
Clusters can be selected using the interactive legend in (**i**) (multi-option selector), or clicking in the scatterplot (**h**) or clicking in the parallel coordinates (**g**). When a cluster is selected, the filtering is applied almost on the entire dashboard: (**a**), (**c**), (**d**) and (**e**) show only transactions related to the customers belonging to that cluster, while (**l**) table contains only the selected cluster customers. This allows an in-depth analysis of the cluster, from different perspectives: proficiency ((**c**)), purchase trend ((**b**)) and seasonalities ((**e**)), emerging categories and subcategories considering the entire cluster ((**d**)), individual and precise values of purchased subcategories, inter-cluster correlations between purchased subcategories ((**g**)), individual demographic and rfm details of the single customers ((**l**)), cluster cohesion ((**h**)).

Rfm Customer Segmentation (k)

This visualization is inspired by Ron Kohavi (2004) work: it shows RFM segments using a RF heatmap where the colour saturation represents the average monetary value of each segment cell. We make the chart more interactive by allowing the end-user to select one or many cells. The result of this interaction is similar to the cluster selection: all the other visualizations are filtered considering customers that belong to the selected segment. Due to the equal-frequency binning, it may happen that some segments have similar characteristics, and this is why we allow to select more than one segment cell at a time. When a cell is selected, the border is highlighted; on mouse over, a tooltip shows segment size and precise value of average monetary, together with the segment label that we created.

Segment Details (l)

Even though we choose to focus on group of customers (segment) instead of individual customers in our analysis, having individual information may be useful too. Consider, for instance, the following settings:

- the retailer identifies clusters of customers that have common interests, and wants to send them advertisements tailed on cluster needs;
- there are customers that have purchased frequently in the past, but recently they stopped to purchase: the retailer wants to encourage them to buy again, sending tailed advertisements to re-activate them;
- the retailer wants to offer discounts to its "Gold" customers (customers with high frequency, recency and monetary values);
- the retailer identifies a highly proficient cluster and wants to send discounts to the Big Spenders of the cluster to increase their loyalty and satisfaction.

All the described settings require to know exactly who are the individual customers (especially the customer identifier). We provide individual customer details in a simple table that can be sorted according to each column: id, day of birth, gender, city, cluster, recency value, frequency value, monetary value, R score, F score, M score and lastly, the colour of the rfm segment. This may be useful if the retailer wants to consider only a subset of the created clusters/segments due to further analyses and take only the correspondent ids. Moreover, this table can also be useful to inspect precise values of recency frequency and monetary, because in (**k**) we just have the average monetary and the R,F segment rank, and not the values. If any of cluster and rfm segment is selected, the table shows only customers belonging to the segment, together with an indication of the segmentation method used (K-Means or Rfm), otherwise, if no selection is applied, all customers are displayed. The total number of selected customers is displayed, too.

4.3 Other comments

The use of Colour

Colour can be considered as the most pre-attentive feature. However, using too many colours may disturb the user and produce the opposite effect, removing the attention from important insights. Furthermore, different

goals require different colour scales and choices: categorical data need contrasting colours, continuous data are often more understandable with monochromatic scales with different levels of saturation, to quickly understand if a metric is increasing or decreasing.
Our dashboard contains many visualizations, and most of them require colours encoding for faster and improved insights generation. Colours create associations in the different visualizations (ex: same colour in the scatter plot, parallel coordinates and segment table indicates the same cluster), but if a cluster colour is similar to a category colour of the treemap, the user may create the wrong association or get confused. With the help of ColorBrewer we created 2 qualitative palettes but with different saturation and light levels to distinguish between them: colours with high level of light and low saturations are used to encode product categories, while colours with high saturation levels and lower light are used to encode clusters. For the continuous scales, we use sequential scales with interpolator, provided by d3 and based on ColorBrewer (can be found here). On the one side, using the same colour scale for both average monetary value and daily profit/sales may be confusing, but, on the other side, using different colour scales may again add confusion due to the large number of colours used in the dashboard. Our compromise is to select different colour scales but based on the same colour (purple), but in **(e)** we use an interpolator of blue and purple, while in **(k)** an interpolator of purple only. We tried to keep all the other parts of the dashboard as much neutral as possible, with gray backgrounds and buttons, and white or gray texts.

Overview, Details, Interaction

We tried to always provide an overview, but at the same time, to offer details to the user. This is done through interactions (zooming, filtering by select/unselect), and with an intense use of tooltips with precise values displayed on mouse over in almost each visualization. We use the font-size to focus the attention on particular details (ex: the total customers, the total profits,...).

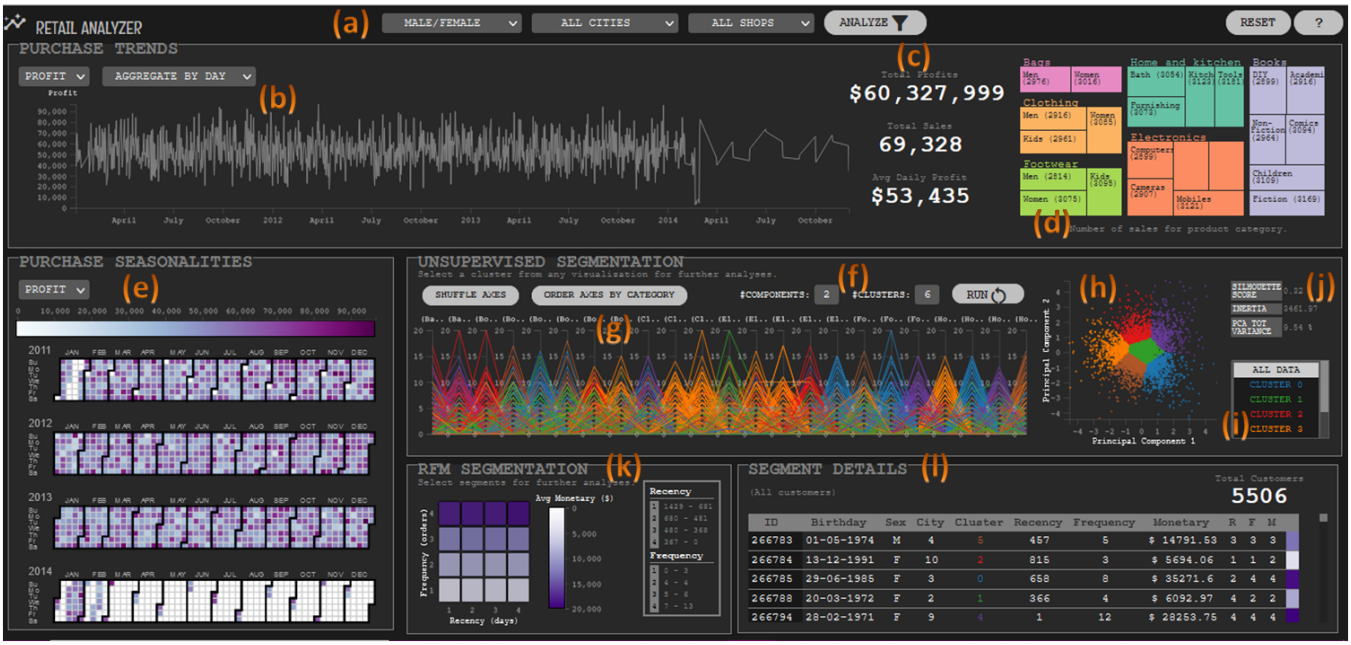

Figure 1: A screenshot of the implemented dashboard.

5 Conclusions

The implemented dashboard allows for an interactive, visual analysis for market and customer segmentation in retail. The advantages of such analysis are various:

- better resource allocation : you buy from the producers only what is likely to be purchased by customers;
- more effective targeted advertisement, focused on groups of customers with similar characteristics;
- more proficient marketing, treating customers differently depending on their importance and proficiency;
- improved customer satisfaction : the market is driven by customers' needs and interests.

References

A. S. M. Shahadat Hossain. Customer Segmentation using Centroid Based and Density Based Clustering Algorithms. *3rd International Conference on Electrical Information and Communication Technology (EICT)*, 2017.

Bertjan Broeksema Julian Heinrich. Big Data Visual Analytics with Parallel Coordinates. *Big Data Visual Analytics (BDVA)*, 2015.

Saraswati Jadhav Rahul Shirole, Laxmiputra Salokhe. Customer Segmentation using RFM Model and K-Means Clustering. *International Journal of Scientific Research in Science and Technology (IJSRST)*, 8(3), 2021.

Mohammad Hafiz Hersyah Miftahul Jannah Ricky Akbar, Meza Silvana. Implementation of Business Intelligence for Sales Data Management Using Interactive Dashboard Visualization in XYZ Stores. *International Conference on Information Technology Systems and Innovation (ICITSI)*, 2020.

Rajesh Parekh Ron Kohavi. Visualizing RFM Segmentation. *Proceedings of the 2004 SIAM international conference*, 2004.