UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF INFORMATION SYSTEMS

FINAL PROJECT REPORT

DATA ANALYTICS WITH R/PYTHON COURSE

TOPIC: ANALYSIS OF CUSTOMER SEGMENTATION BY RFM MODEL AND K-MEANS CLUSTERING IN ADVENTUREWORKS COMPANY

Lecturer: Nguyen Phat Dat, MA.

Group: MEP Familu

Ho Chi Minh City, June 08th, 2022

UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF INFORMATION SYSTEMS

FINAL PROJECT REPORT

DATA ANALYTICS WITH R/PYTHON COURSE

TOPIC: ANALYSIS OF CUSTOMER SEGMENTATION BY RFM MODEL AND K-MEANS CLUSTERING IN ADVENTUREWORKS COMPANY

Lecturer: Nguyen Phat Dat, MA.

Group: MEP Familu

Ho Chi Minh City, June 08th, 2022

MEMBERS OF GROUP

No.	Name	Student ID
1	**Nguyen Tran Ngoc Tram**	**K194060832**
2	Nguyen Thi To Nhi	K194060800
3	Nguyen Thanh Phong	K194060803
4	Do Thi Thanh Phuong	K194060806
5	Vu Thi Thu Thao	K194060818
6	Truong Thi Thanh Thu	K194060825

ACKNOWLEDGEMENTS

MEP Familu applied what he learned in the Data analytics with R/Python course through the process of learning theoretical knowledge as well as practical knowledge to build the project.

MEP Familu would like to express their sincere gratitude to all those who assisted them in completing the course report. The team would like to thank Mr. Nguyen Phat Dat, who provided a solid foundation of knowledge, had many comments to help the group complete the project successfully and provided solutions for the group when difficult problems arose. In addition to the main lecturers of the subject, the group thanks assistant lecturer Tran Le Tan Thinh for sharing extremely useful information.

In the limited time of the project, the team tried to come up with ideas and address the initial requirements in the best possible way. However, there are still many obstacles to overcome, mistakes are inevitable. Expect the lecturers to read and comment on the group's topic so that the group can learn from it and improve it.

COMMITMENT

During the project's implementation, the team pledges to follow all regulations, and all data and results presented in the report are accurate. All Internet, book, and textbook references are specifically cited.

If any mistakes are made, the group would like to accept full responsibility and all forms of discipline.

MEP Familu

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

RFM	Recency - Frequency - Monetary
AWC	AdventureWorks Company
EDA	Exploratory Data Analysis
CLV	Customer Lifetime Value
SSE	Sum of the squared error
WCSS	Within-Clusters-Sum-of-Squares
LTV	lifetime value
CRR	Customer Retention Rate

CHAPTER 1 PROJECT OVERVIEW

Present an overview of the content of the project, including reasons for choosing the topic, objectives, expected results, and research methods.

1.1. Reason for choosing the topic

Customers are the ones who consume and make the products and services of the business show their practical value and the most important asset to the business is the customer. If there are no customers, the goods will be backlogged and unsold, resulting in business bankruptcy.

In the competitive situation and the increasing complexity of the business environment, customer segmentation is an important factor in customer management and building an appropriate marketing strategy. Customer segmentation aims to focus and take better care of customers based on the unique characteristics of each customer group. Any business, if it wants to survive and develop long-term in the market, needs to have a set of customers for its business. However, businesses often have many different types of customers, so it can be difficult to know a specific audience to target with their marketing.

With the strong development of data science technology, Enterprises can identify their best customers through customer segmentation by applying the concept of Data Mining and Customer Relationship Management. Data Mining based on RFM model and Clustering Techniques With K-Means Algorithm allows marketers to identify specific customer groups based on demographics, interests, behaviors generating much higher response rates , plus increased customer loyalty and long-term value. RFM segmentation is an effective way to help businesses have the right strategy to accompany customers' shopping or service needs, and from there can respond promptly to these needs. Therefore, implementing customer segmentation for

businesses is essential. From an overview of the meaning of customer segmentation, our team chose the topic: "Analysis of customer segmentation by RFM model and K-Means clustering in AdventureWorks company".

1.2. Research objectives

The objective of the project is to study and find out how to segment AWC's customer segments. To do this, our team will perform the following tasks:

1. Analysis of customer segments of AWC company based on building RFM model.

2. Analysis of customer segmentation of AWC company based on the construction of unsupervised clustering- K Means.

3. In-depth analysis of AWC's customers through indicators: Cohort Analysis, Churn rate.

1.3. Research question

To accomplish the set goals, our team will ask implementation questions:

1. How are the customer segments divided by RFM and K-Means models and which segment is profitable for the company?

2. How is the demographics of each customer in each segment distributed (by their age, gender, marital status, income and occupation)?

3. In which geographic areas are the customers of each segment concentrated?

4. What is the percentage of customers who spend money on AWC's products in each segment?

5. What is the rate of customers leaving the company from 2011 to 2014?

6. Is AWC's current customer retention rate good?

1.4. Expected results

- The scope of the project's research is to understand and build a BI solution for the sales model of the company Adventure Work in terms of sales revenue and profit and data of all agents on a global scale. A multinational company that manufactures and sells bicycles made of metal and composites with markets that include North America, Europe and Asia. Understand the theory of how to divide customers by RFM and K-Means
- Apply the division of customer segments to understand the company's customers so that the company has long -term strategies for each segment.

1.5. Research object and scope

- Object: Customers of Adventure Works Cycles company until 2014
- Scope: Adventure Works Cycles, the fictitious company on which the AdventureWorks sample databases are based, is a large, multinational manufacturing company. The company manufactures and sells metal and composite bicycles to North American, European and Asian commercial markets.

1.6. How to select data

The data was obtained from the source data set **"Adventureworks2014"**. Through the data analysis process, determine the attributes needed for customer segment analysis, thereby building the data warehouse **"AWC sales"** with the dim tables and fact tables (including **DimCustomer, DimProduct, DimGeography, DimSalesTerritory , Dim Date, FactSales**). All data warehouse building processes are done on Microsoft SQL Server. After that, the team went through the ETL process using SSIS to dump the data into the built-in data warehouse. After having the complete data, the team will link Excel to the generated data set to create an xlsx data set.

1.7. Implementation Process

The research process of the subject project based on the objectives and methods is carried out according to the following process: Initially determine the reason for choosing the research topic, determine the goal of the topic, build a data warehouse. The next step is to process the data, build the RFM model and analyze the results of the model. then, perform clustering with K-means algorithm and deal with problems related to K-means algorithm. Perform customer retention analysis and customer churn analysis. Finally, evaluate and conclude the results of the project.

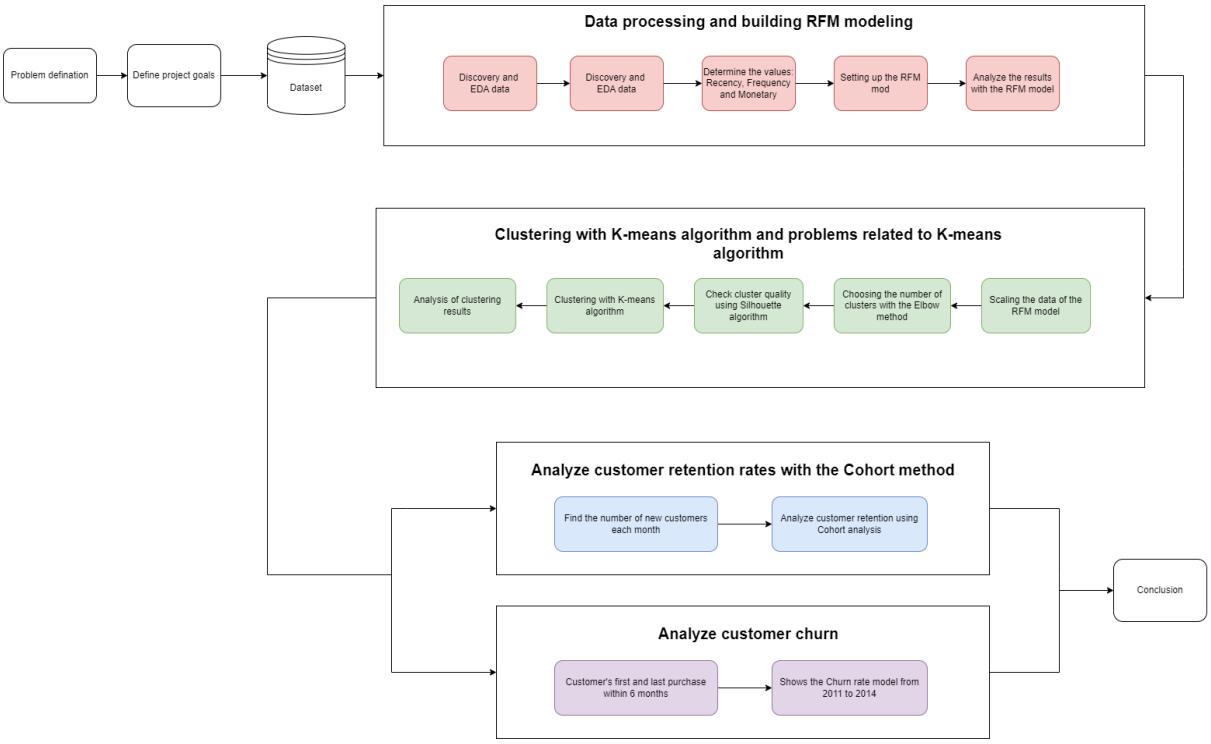
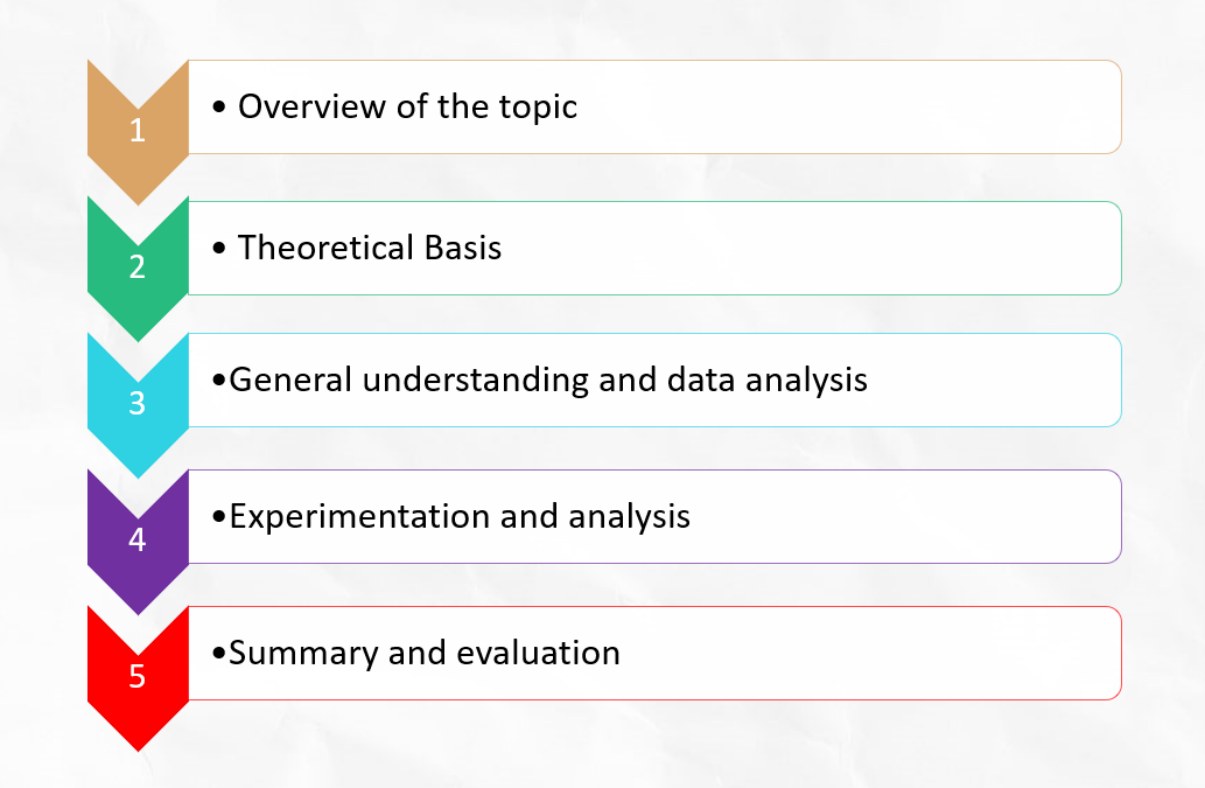


Figure 1-1 Implementation Process

1.8. Chapter overview

The overview of the report includes the following five chapters:



Chapter 1: Overview of the topic

Present an overview of the content of the project, including reasons for choosing the topic, objectives, expected results, and research methods.

Chapter 2: Theoretical Basis

Present an overview of the concept, meaning, benefits, and calculation formula of the RFM model, clustering according to K-Means, customer churn rate, and customer retention rate. From there, building a foundation for practical applied research in the customer segment.

Chapter 3. General understanding and data analysis

The content of this chapter presents an overview of the Advanture work company's data set, describing the meaning of the data shown. Perform data EDA data cleaning. Furthermore, analyzing the indices of the RFM model and K-means clustering.

Chapter 4: Experimentation and analysis

Provide data visualization charts. thereby giving an analysis of the customer segment of each model. Find out the salient features of each segment based on variables like geography, age, gender, occupation and income based on data visualization. And analyze customer churn rate and customer retention rate.

Chapter 5: Summary and evaluation

Summary of implementation content, evaluation of implementation results and development direction of the project.

CHAPTER 2 THEORETICAL BASIS

Present an overview of the concept, meaning, benefits, and calculation formula of the RFM model, clustering according to K-Means, customer churn rate, and customer retention rate. From there, building a foundation for practical applied research in the customer segment.

2.1. Overview of RFM

2.1.1. Concepts

a. Customer Segmentation

Customer Segmentation Models

Accurate customer segmentation involves tracking dynamic changes, and frequently updating new data. Although segmenting customers according to their CLV is the recommended approach, there are many types of customer segmentation models. Some of the more common types are **segmentation via cluster analysis**, **RFM segmentation**, and longevity. Some marketers might even combine one or more segmentation models in order to reach their goals.

No matter the types of segmentation models marketers decide to use, they all require marketers to create groupings of customers to serve as a first step in segmenting the customer base. Usually this will result in marketers having a series of tiers for each type of segmentation model. Marketers can then mix different tiers across models to create more defined segments. For example, mixing the highest tier of customers based on an RFM model and combining it with a low longevity tier will result in marketers having a segment of highly active, newly acquired customers.

Customer Segmentation and Machine Learning

An additional approach to customer segmentation is leveraging machine learning algorithms to discover new segments. Different from marketer-designed segmentation models, as the ones described above, machine learning customer segmentation allows advanced algorithms to surface insights and groupings that marketers might find difficult discovering on their own.

Furthermore, marketers that create a feedback loop between the segmentation model and campaign results will have ever improving customer segments. In these cases, the machine learning model will be not only able to refine its definition of segments, but also be able to identify if a specific subset of the segment is outperforming the rest, optimizing marketing performance

b. RFM

RFM Model was introduced by Hughes in 1994 for customer value analysis and effective customer segmentation. This model has been used for more than 30 years now and still remains a useful method for optimizing sales and building campaigns to engage customers.

Recency, frequency and monetary (RFM) analysis is a powerful and recognized technique in database marketing. It is widely used to rank the customers based on their prior purchasing history. RFM analysis finds use in a wide range of applications involving a large number of customers such as online purchase, retailing, etc. This method groups the customers based on three dimensions, recency(R), frequency (F) and monetary (M).

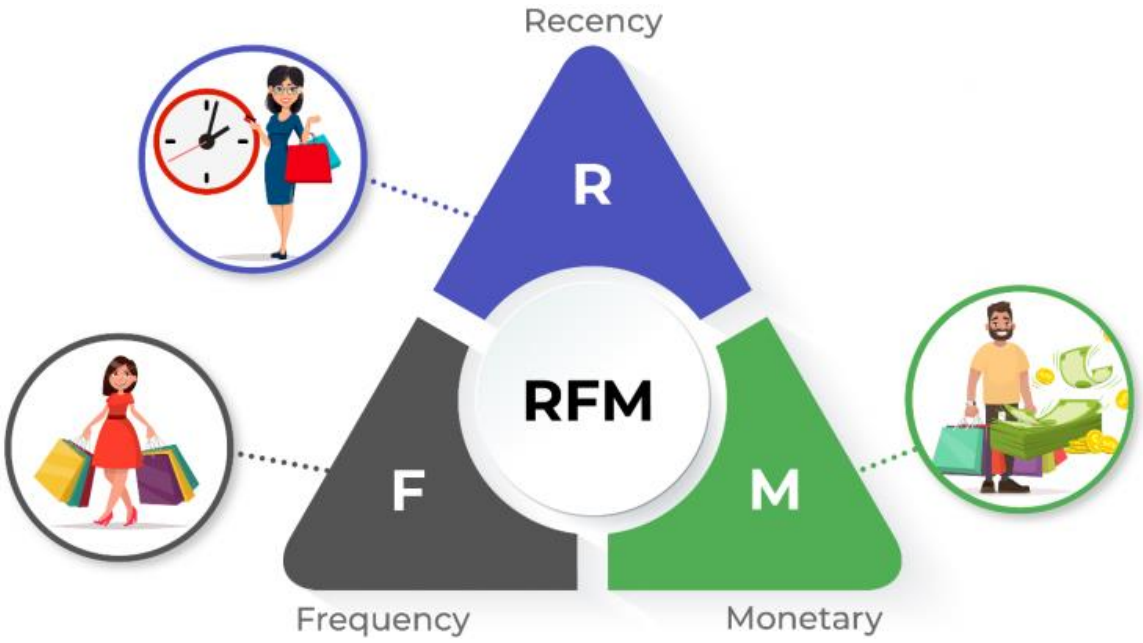


Figure 2-1 RFM Model

2.1.2. Component parts

Recency – When was the last time the customer made a purchase?

Recency value is the number of days a customer takes between two purchases. A smaller value of recency implies that the customer visits the company repeatedly in a short period. Similarly, a greater value implies that the customer is less likely to visit the company shortly.

Frequency – How many times did the customer purchase?

Frequency is defined as the number of purchases a customer makes in a specific period. The higher the value of frequency the more loyal are the customers of the company.

Monetary – How much money did the customer spend?

Monetary is defined as the amount of money spent by the customer during a certain period. The higher the amount of money spent the more revenue they give to the company.

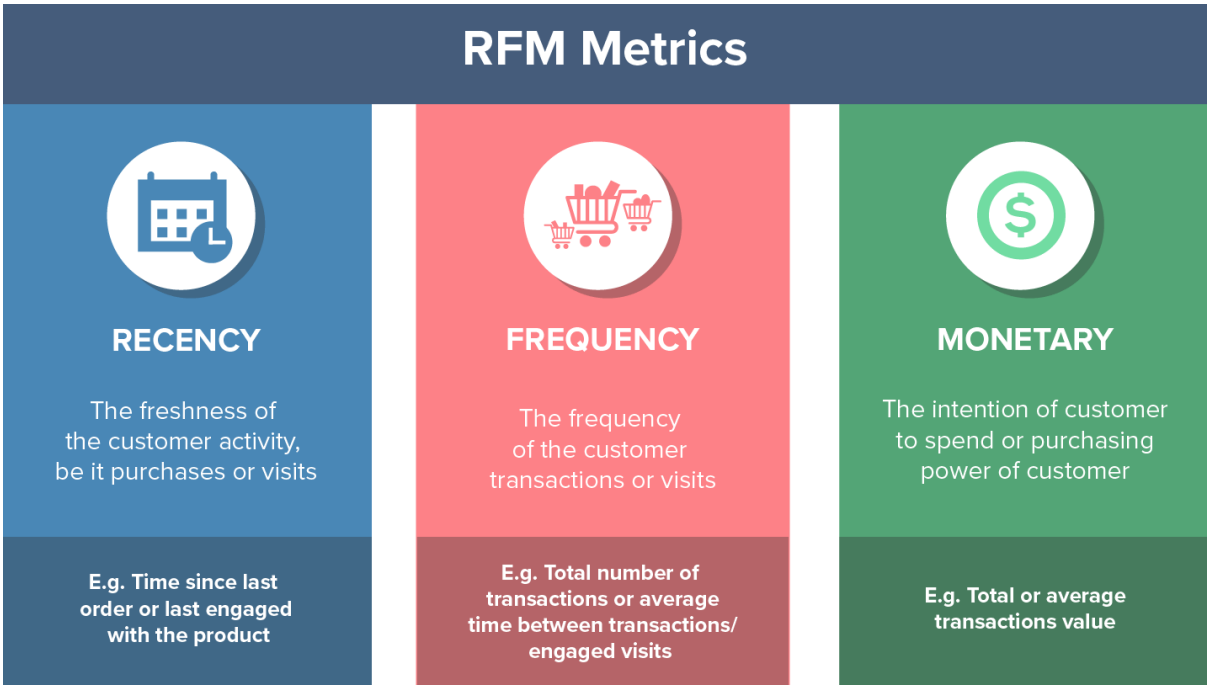


Figure 2-2 RFM Metrics

2.1.3. RFM benefits/ImportancE

Importance of RFM

RFM analysis works upon the marketing axiom (The Pareto Principle) (80:20 Rule) that "80% of a company's business comes from 20% of its customers".

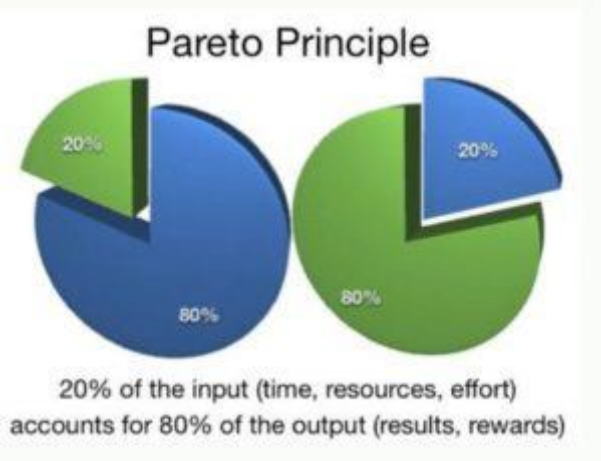


Figure 2-3 Pareto Principle

Businesses should make optimal use of their resources to gain as well as retain key customers. By adopting RFM Model Analytics, an e-commerce store can expertly target valuable customers and build promotional strategies.

RFM analysis plays a significant role in determining the relation of a customer with the company. By effective analysis of RFM, a store can attract customers by realizing their needs. It can increase customer engagement by giving custom product suggestions in accordance with their interests.

Benefits

Performing RFM analysis allows you to understand customer behavior and predict how segments could evolve in the future. The predictive character is one of the most important benefits of using RFM analysis for your business.

Save valuable resources

Instead of spending hours aggregating data and preparing spreadsheets, you focus solely on analyzing your RFM segments. Automation allows your teams to extract valuable insights to optimize strategies and increase customer lifetime value

Eliminate errors

You can’t afford errors in customer segmentation and making decisions around RFM analysis. Automated RFM solutions eliminate errors associated with manual work and offer reliable data for your growth plans

Up to date

Using automated RFM segmentation ensures that all your segments are constantly updated. You can perform regular analysis knowing that you’re looking at fresh reports, or you can select a certain period to search for trends or anomalies

Real-time actions

Having up-to-date information allows you to take advantage of opportunities that arise – like nurturing new high-value customers, or prevent negative trends – like an increasing number of complaints from a category of newly acquired customers

Consistency and traceability

Segmenting your customers based on the RFM model helps you maintain consistency in analyzing segments and traceability over the evolution of your RFM segments. This way, all departments share the same view over the segments using the same reference system

2.1.4. How to divide customers of RFM

Each customer is assigned with three different scores for recency, frequency, and monetary variables. Scoring is done on the scale from 5 to 1. The top quintile is given a score of 5, and the others are given 4, 3, 2 and 1. The scores can be assumed to have unique characterist.

Finally, all the customers are provided with scores 555, 554…,111. The customers with the score 555 can be called as the potential customers of the company since they are likely to give more profit to the company and vice versa goes with the customers having a score of 111. Depending on this RFM score, each customer can be put into a different segment.

2.1.5. Process of performing RFM segmentation

There are 5 steps to perform RFM analysis

Step 1: Collection and Collation of relevant data/values.

The RFM model involves analysis of customer transaction history. The first step is to pull out the RFM data for each customer in ascending order.

Step 2: Setting the RFM

businesses need to create custom filters in order to effectively segment the customers. This is an important aspect that will vary based on the nature of their business.

Step 3: Assigning scores

you can now assign each customer a grade based on the table above. By doing so, you're converting absolute values of transactions into chunks of similar transactions, based on RFM. Now you no longer need the absolute values mentioned in brackets, and just use the score for segmentation and analysis. After assigning scores, you can create chunks of similar customers, who have identical or similar scores in the three criteria.

Step 4: Labeling segments

The labels we use will be based on the differing characteristics of the three grades customers have received. Businesses may or may not require 125 distinct segments and can decide the number of scoring segments required and label them, based on the nature of the business.

Step 5: Creating customized strategies/tactics for relevant segments

Once businesses have segmented and labeled each customer, they can ensure personalization in all their messaging. At-risk customers can be targeted with offers, discounts, or freebies, whereas loyal customers can be provided a superior level of service in order to make them feel more valued. Recent customers can be sent information about other products that they would be interested in, whereas the Champion customer could be given greater access to products and used as a mechanism for feedback, before launching it to other customers. All of this can be done simultaneously by the business.

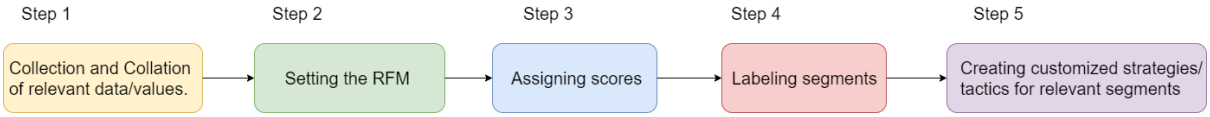


Figure 2-4 Steps of RFM model

2.2. Overview of K-Means Clustering

2.2.1. Concepts

a. Clustering

Clustering is a set of techniques used to partition data into groups, or clusters. Clusters are loosely defined as groups of data objects that are more similar to other objects in their cluster than they are to data objects in other clusters. In practice, clustering helps identify two qualities of data:

1. Meaningfulness
2. Usefulness

Meaningful clusters expand domain knowledge. For example, in the medical field, researchers applied clustering to gene expression experiments. The clustering results identified groups of patients who respond differently to medical treatments.

Useful clusters, on the other hand, serve as an intermediate step in a data pipeline. For example, businesses use clustering for customer segmentation. The clustering results segment customers into groups with similar purchase histories, which businesses can then use to create targeted advertising campaigns.

b. K-means clustering

The *k*-means clustering method is an unsupervised machine learning technique used to identify clusters of data objects in a dataset. There are many different types of clustering methods, but *k*-means is one of the oldest and most

approachable. These traits make implementing *k*-means clustering in Python reasonably straightforward, even for novice programmers and data scientists.

2.2.2. Component parts

Conventional *k*-means requires only a few steps. The first step is to randomly select *k* centroids, where *k* is equal to the number of clusters you choose. Centroids are data points representing the center of a cluster.

The main element of the algorithm works by a two-step process called expectation-maximization. The expectation step assigns each data point to its nearest centroid. Then, the maximization step computes the mean of all the points for each cluster and sets the new centroid. Here's what the conventional version of the *k*-means algorithm looks like:

Algorithm 1 *k*-means algorithm

1: Specify the number k of clusters to assign.
2: Randomly initialize k centroids.
3: **repeat**
4: **expectation:** Assign each point to its closest centroid.
5: **maximization:** Compute the new centroid (mean) of each cluster.
6: **until** The centroid positions do not change.

The quality of the cluster assignments is determined by computing the sum of the squared error (SSE) after the centroids converge, or match the previous iteration's assignment. The SSE is defined as the sum of the squared Euclidean distances of each point to its closest centroid. Since this is a measure of error, the objective of *k*-means is to try to minimize this value.

The purpose of this figure is to show that the initialization of the centroids is an important step. It also highlights the use of SSE as a measure of clustering performance. After choosing a number of clusters and the initial centroids, the

expectation-maximization step is repeated until the centroid positions reach convergence and are unchanged.

The random initialization step causes the *k*-means algorithm to be nondeterministic, meaning that cluster assignments will vary if you run the same algorithm twice on the same dataset. Researchers commonly run several initializations of the entire *k*-means algorithm and choose the cluster assignments from the initialization with the lowest SSE.

2.2.3. Benefits/breakthroughs of K Means

It works well with large datasets and it's very easy to implement.

In clustering, especially in K-means, we have the benefit of having a convergence stage in the final as it's a good indicator of stable clusters. The program stops when the best result comes out.

We can use numerous examples as data in it. It is a very adaptive type of algorithm.

It can create clusters of a variety of shapes that gives much broader importance to the data visualization part.

The clusters of k-means do not overlap with each other as they prove to be non-hierarchical.

K-means is faster than hierarchical clustering.

The clusters produced can be a lot more dense and tighter than hierarchical clustering due to the presence of globular clusters.

2.2.4. K-means's customer segmentation

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big

task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is used Elbow Method

The Elbow method is one of the most popular methods used to **select the optimal number of clusters** by fitting the model with a range of values for K in the K-means algorithm. The Elbow method requires drawing a line plot between SSE (Sum of Squared errors) vs number of clusters and finding the point representing the "**elbow point**" (the point after which the SSE or inertia starts decreasing in a linear fashion). Here is the sample elbow point. In the later sections, it is illustrated as to how to draw the line plot and find elbow point.

The formula to calculate the value of WCSS (for 3 clusters) is given below

$$WCSS= \sum_{P_i \text{ in Cluster1}} distance(P_i\ C_1)^2 + \sum_{P_i \text{ in Cluster2}} distance(P_i\ C_2)^2 + \sum_{P_i \text{ in CLuster3}} distance(P_i\ C_3)^2$$

$\sum_{Pi \text{ in Cluster1}} distance(P_i\ C_1)^2$: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

2.2.5. Process of performing K-means segmentation

To find the optimal value of clusters, the elbow method follows the below steps:

1. It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).

2. For each value of K, calculate the WCSS value.

3. Plots a curve between calculated WCSS values and the number of clusters K.

4. The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

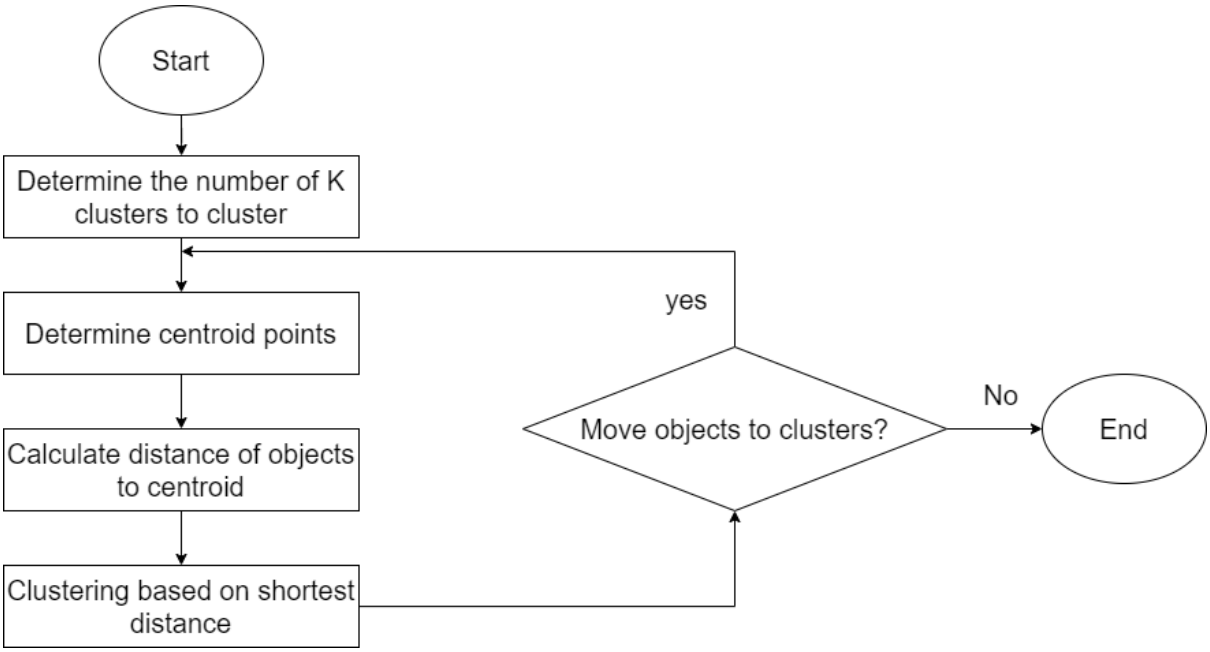


Figure 2-5 A flow chart showing the steps of K-means clustering

2.3. Overview of customer churn Churn rate

2.3.1. Concepts

What is a churn rate?

Churn rate, sometimes known as attrition rate, is the rate at which customers stop doing business with a company over a given period of time. Churn may also apply to the number of subscribers who cancel or don't renew a subscription. The higher your churn rate, the more customers stop buying from your business. The

lower your churn rate, the more customers you retain. Typically, the lower your churn rate, the better.

What is customer churn?

Customer churn refers to the natural business cycle of losing and acquiring customers. Every company — no matter the quality of its products or customer service — experiences churn. Generally speaking, the less churn you have, the more customers you keep.

2.3.2. Calculation method

Luckily, the lead way on how to measure churn rate is quite simple.

All you have to do is find out two metrics:

- How many customers were with your business at the beginning of the period, say a month
- How many customers stayed with your business at the end of the same period

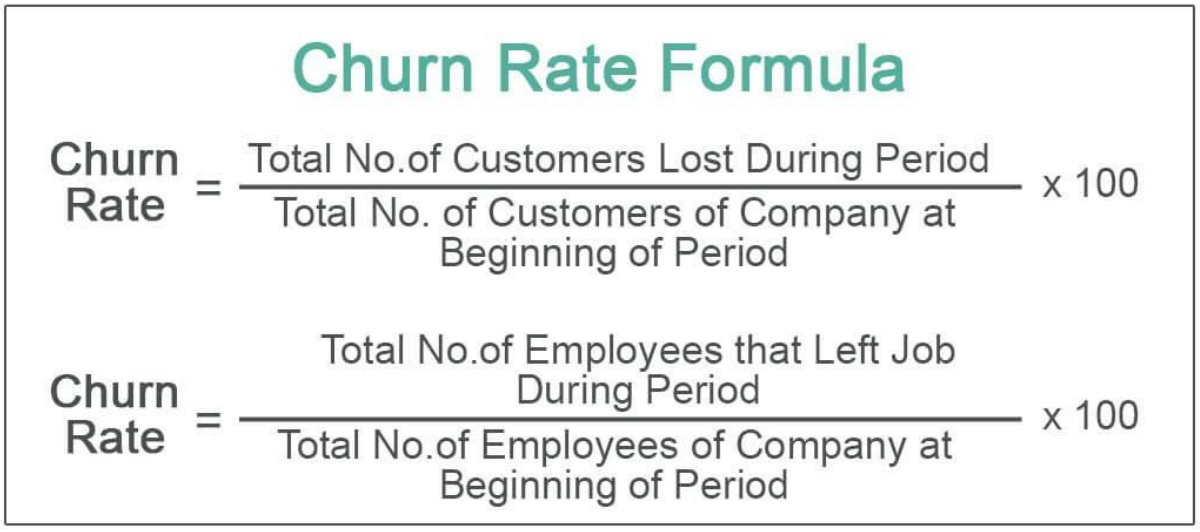


Figure 2-6 Churn rate formula

2.3.3. Meaning and benefits of implementation

Meaning of implementation

Understanding your customer churn is essential to evaluating the effectiveness of your marketing efforts and the overall satisfaction of your customers. It's also easier and cheaper to keep customers you already have versus acquiring new ones. Due to the popularity of subscription business models, it's critical for many businesses to understand where, how, and why their customers may be churning.

Benefits

Why devote effort to customer churn analysis when you could spend the time on a long list of other projects? Here's a snapshot of benefits that accrue from analyzing customer churn.

Increase profits

A business sells its products or services to make money. The ultimate goal of a customer churn analysis, then, is to increase profits by lowering customer attrition. If more customers stay around for a longer period, you should see revenues increase and profits follow.

Improve the customer experience

One of the worst reasons to lose a customer is an easily avoidable mistake, like shipping the wrong item. Understanding why customers churn can help you learn about their priorities, identify your own weaknesses and improve the overall customer experience.

Also referred to as "CX," customer experience is your customers' perception or opinion about their interactions with your business. Their view of your brand is

shaped throughout the buyer's journey, from their first interactions to post-sale support, and has a lasting impact on your company, including your bottom line.

Optimize your products and services

If customers are leaving because of specific problems with your products or services or delivery methods, you now have opportunities to improve. Not only will acting on these insights decrease customer churn, it leads to a better overall product or service that earns you more future growth.

Customer retention

The opposite of customer churn is customer retention—a business's ability to keep its customers and continue to generate revenue from them. Strong customer retention allows a business to increase the profitability of existing clients and maximize their lifetime value (LTV).

If you sell a service that costs $1,000 per month, keeping a customer for three additional months means you'll bring in an additional $3,000 in revenue per customer without spending on acquisition. The scale and dollar amount vary by business, but the concept is universal: Repeat business is profitable business.

2.4. Overview of customer retention analysis Cohort

2.4.1. Concepts

What is Cohort Analysis?

Cohort analysis is a subset of behavioral analytics that takes the data from a given eCommerce platform, web application, or online game and rather than looking at all users as one unit, it breaks them into related groups for analysis. These related groups, or cohorts, usually share common characteristics or experiences within a defined time-span.

What is Customer Retention Analysis?

Customer Retention Analysis is the process of studying metrics to understand how and why customers leave your business or churn. It helps to gain insights on a couple of notes which are pertinent to your business.

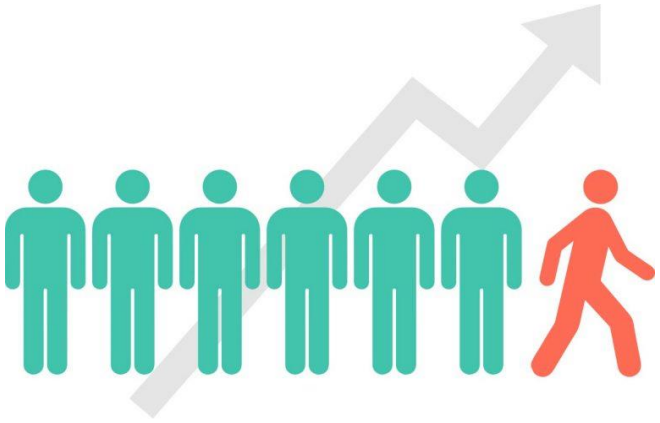


Figure 2-7 Illustration of churning customer

2.4.2. Calculation method

In all these industries, cohort analysis is commonly used to identify reasons why customers leave and what can be done to prevent them from leaving. That brings us to the calculation of Customer Retention Rate (CRR).

Customer retention rate is calculated with the help of this formula

CRR = ((E-N)/S) X 100

E – The number of customers at the end of the time period.

N – The number of customers acquired during that period.

S – The number of customers at the beginning (or start) of the period.

To measure customer retention, we find the difference between the number of customers acquired during the period from the number of customers remaining at the end of the period. This gives a true picture of retained customers.

A higher CRR means higher customer loyalty. By benchmarking your business CRR with the industry average, you can see where you stand in terms of customer retention. If CRR shows a bleak picture, corrective measures can be taken with the help of data analysis - this is where cohort analysis can help.

2.4.3. Meaning and benefits of implementation

Meaning of implementation

Evaluates the quality of your acquisition strategy

See the picture on top, with a leaking pipe? If you are not measuring customer retention, you may have a [customer] leakage problem. You may be losing existing users at a catastrophic rate while you spend your precious dollars acquiring new ones with AdWords or Facebook ads, or whatever acquisition strategy you may have.

My business model is solid because I get so many new users!

Maybe that was a little harsh… But it serves to make a point. You need to know your retention rate if you want to make informed decisions about your business model. You will also want to know how much is the lifetime value of your customers so you can evaluate if you are spending too much acquiring them, but that is out of the scope of this article!

It allows you to test what works and what doesn't

Of course, using retention related metrics is not only useful to evaluate your acquisition campaign. You can experiment with several things, like A/B testing a signup offer, or an app feature. Give A to the users who sign up in January, and give B to users who sign up in February. Measure the retention rate and check if there is a campaign that worked better than the other. Rinse. Repeat.

The bottom line is that for some people, Retention is king. And this should be logical because if you keep your customers longer, they will eventually spend more money on your product, which means increasing their lifetime value.

2.4.4. Benefits

For the long term success of any company or product, customer acquisition or customer satisfaction is essential, but the key role is played by customer retention. Why is customer retention important, and what's all the hype about, you may ask. Let's put all doubts to rest with these benefits of customer retention.

Increase in repurchasing behavior

Repurchasing and customer retention have a direct, positive relationship. Customer retention leads to repurchase leads to increased turnover as customers will keep returning. Hence, a positive relationship will have a tremendous impact on business.

Declined price sensitiveness

Retained customers are loyal to the company or product. To them, the company/product meets their demands. They find it challenging to give up with a minor increase in price because the customer benefits they get outweigh the change. It makes them more likely to accept a price rise.

Positive word of mouth

Satisfied and loyal customers advertise your product for you by positive word of mouth. Word of mouth is considered a powerful marketing tool even in the modern business age. Prospective customers are more likely to believe a fellow customer's recommendation over a brand's self-advertising. Without much effort and money, it turns out to be an economically beneficial method with better results.

These are just some of the various benefits that customer retention provides. Other advantages include decreased costs for acquiring new customers, strengthening the unique selling proposition (USP), getting qualitative feedback and more.

2.4.5. Top customer retention strategies

Provide great customer service

Customer service is key. Customers measure how useful your tool is at every step, especially when they run into a problem or face issues with your company/product. Having friendly and quick customer service at such vital moments rebuilds their confidence in the product and makes them stick for longer.

Care for your customers

There are several times customers leave when they think that you stopped caring for them. Don't isolate your customers. Take their feedback seriously, reassuringly respond to them and resolve the problems, if any.

Keep in touch with them

Communication is the most important factor in maintaining customer relationships. When you communicate with your customer, you're more aware of their needs, and you can fulfill them. In turn, they choose to stay with you and continue using your product.

Maintain a good record

Loyal customers are the ones who will stick to your products or company and will choose you over your competitors. Maintain a customer database and figure out who these loyal customers are, and then make conscious efforts to maintain a

positive relationship with them. Putler's RFM analysis can help you spot your loyal customers in just a few clicks.

Provide quality services

Providing quality products and services will give you an edge over competitors. If you lead in your field, customers themselves would not want to leave you in the first place. Focus on the quality of the services you provide to make your customers stay.

Train your employees

In an isolated state, not everyone will realize how important a loyal customer is for the company. Training your employees, informing them of the importance of customer retention and asking them to take the initiative and develop beneficial ideas for customers can effectively involve employees in strategic processes. It makes them feel empowered and responsible, motivating them to work better.

Polls and surveys

This is a fun strategy that helps retain customers. It'll also give you insight into customer perception of your company or product. Create short and interesting polls, surveys or questionnaires for your customers, asking their opinion, experience, suggestions, etc. and see your customer retention rate rise constantly.

CHAPTER 3 GENERAL UNDERSTANDING AND DATA ANALYSIS

The content of this chapter presents an overview of the Advanture work company's data set, describing the meaning of the data shown. Perform data EDA data cleaning. Furthermore, analyzing the indices of the RFM model and K-means clustering.

3.1. Understanding the AWC dataset

This analysis uses the AdventureWork Company (AWC) dataset. This dataset represents sales transactions (FactSales), customer information (DimCustomer), customer location (DimGeography), geographic location information (DimSalesTerritory) and real time customer current transaction (DimDate).

FactSales:

df_FSales

	ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	SalesTerritoryKey	SalesOrderNumber	SalesOrderLineNumber	...	UnitPriceDiscountPct	DiscountAmount	ProductStandardCost
0	310	20101229	20110110	20110105	21768	1	19	6	SO43697	1	...	0	0	2171.2942
1	346	20101229	20110110	20110105	28389	1	39	7	SO43698	1	...	0	0	1912.1544
2	346	20101229	20110110	20110105	25863	1	100	1	SO43699	1	...	0	0	1912.1544
3	336	20101229	20110110	20110105	14501	1	100	4	SO43700	1	...	0	0	413.1463
4	346	20101229	20110110	20110105	11003	1	6	9	SO43701	1	...	0	0	1912.1544
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
60393	485	20140128	20140209	20140204	15868	1	100	6	SO75122	1	...	0	0	8.2205
60394	225	20140128	20140209	20140204	15868	1	100	6	SO75122	2	...	0	0	6.9223
60395	485	20140128	20140209	20140204	18759	1	100	6	SO75123	1	...	0	0	8.2205
60396	486	20140128	20140209	20140204	18759	1	100	6	SO75123	2	...	0	0	59.4660
60397	225	20140128	20140209	20140204	18759	1	100	6	SO75123	3	...	0	0	6.9223

60398 rows × 24 columns

The above data set consists of 60398 rows which means that AWC performed 60398 transactions. The data table with 24 columns is the primary key columns that link the data tables and the columns contain the data needed for a transaction.

This project will focus on researching properties, including: CustomerKey, each of SalesOrderNumber properties belonging to a customer; the SalesAmount

attribute to represent the revenue that customer brings to the company; and the OrderDate property calculates the Recency and Frequency.

```
df_FSales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60398 entries, 0 to 60397
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ProductKey            60398 non-null  int64
 1   OrderDateKey          60398 non-null  int64
 2   DueDateKey            60398 non-null  int64
 3   ShipDateKey           60398 non-null  int64
 4   CustomerKey           60398 non-null  int64
 5   PromotionKey          60398 non-null  int64
 6   CurrencyKey           60398 non-null  int64
 7   SalesTerritoryKey     60398 non-null  int64
 8   SalesOrderNumber      60398 non-null  object
 9   SalesOrderLineNumber  60398 non-null  int64
 10  RevisionNumber        60398 non-null  int64
 11  OrderQuantity         60398 non-null  int64
 12  UnitPrice             60398 non-null  float64
 13  ExtendedAmount        60398 non-null  float64
 14  UnitPriceDiscountPct  60398 non-null  int64
 15  DiscountAmount        60398 non-null  int64
 16  ProductStandardCost   60398 non-null  float64
 17  TotalProductCost      60398 non-null  float64
 18  SalesAmount           60398 non-null  float64
 19  TaxAmt                60398 non-null  float64
 20  Freight               60398 non-null  float64
 21  OrderDate             60398 non-null  datetime64[ns]
 22  DueDate               60398 non-null  datetime64[ns]
 23  ShipDate              60398 non-null  datetime64[ns]
dtypes: datetime64[ns](3), float64(7), int64(13), object(1)
memory usage: 11.1+ MB
```

Specifically, the required information is described in the table below:

Column Name	Datatype	Description
SalesOrderNumber	Object	Unique sales order identification number
SalesOrderLineNumber	Int64	Order line number of that product in a sales order
RevisionNumber	Int64	Incremental number to track changes to the sales order over time.
OrderQuantity	Int64	The number of products which customers buy at the company
UnitPrice	Float64	Selling price of a single product
ExtendedAmount	Float64	Revenue excluding discounts
UnitPriceDiscountPct	Int64	List price minus the sale price then divided by the list price and multiplied by 100 to get a percentage.
DiscountAmount	Int64	A specific dollar amount or a percentage that will be taken off of an item
ProductStandardCost	Float64	Standard cost of 1 product.
TotalProductCost	Float64	The price of the total product is equal to the price of 1 product times the Order Quantity
SalesAmount	Float64	Total amount of one Sales minus tax
TaxAmt	Float64	Tax amount
Freight	Float64	Shipping cost
OrderDate	Datetime64	Dates the sales order was created
DueDate	Datetime64	Date the order is due to the customer
ShipDate	Datetime64	Date the order was shipped to the customer

DimSalesTerritory:

```
df_DSalesTerritory = pd.read_excel(file,'DimSalesTerritory')
df_DSalesTerritory
```

	SalesTerritoryKey	SalesTerritoryAlternateKey	SalesTerritoryRegion	SalesTerritoryCountry	SalesTerritoryGroup
0	1	1	Northwest	United States	North America
1	2	2	Northeast	United States	North America
2	3	3	Central	United States	North America
3	4	4	Southwest	United States	North America
4	5	5	Southeast	United States	North America
5	6	6	Canada	Canada	North America
6	7	7	France	France	Europe
7	8	8	Germany	Germany	Europe
8	9	9	Australia	Australia	Pacific
9	10	10	United Kingdom	United Kingdom	Europe

AWC has 3 Sales Territory Groups such as North America, Europe and Pacific. Sales Territory Country is universal with 6 countries such as: United States,

Canada, France, Germany, Australia, United Kingdom. About Region, it has 8 regions: Northwest, Southwest, Central, Canada, France, Germany, Australia, United Kingdom.

```
df_DSalesTerritory.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column                      Non-Null Count  Dtype 
---  ------                      --------------  ----- 
 0   SalesTerritoryKey           10 non-null     int64 
 1   SalesTerritoryAlternateKey  10 non-null     int64 
 2   SalesTerritoryRegion        10 non-null     object
 3   SalesTerritoryCountry       10 non-null     object
 4   SalesTerritoryGroup         10 non-null     object
dtypes: int64(2), object(3)
memory usage: 528.0+ bytes
```

Specifically, the required information is described in the table below:

Column Name	Datatype	Description
SalesTerritoryRegion	Object	Economic regions in which the company operates include 8 regions
SalesTerritoryCountry	Object	Countries in which the company distributes with 6 countries
SalesTerritoryGroup	Object	The geographical area to which the sales territory belongs divided by 3 groups

DimGeography

```
df_DGeography = pd.read_excel(file,'DimGeography')
df_DGeography
```

	GeographyKey	City	StateProvinceCode	StateProvinceName	CountryRegionCode	EnglishCountryRegionName	PostalCode	SalesTerritoryKey	IpAddressLocator
0	1	Alexandria	NSW	New South Wales	AU	Australia	2015	9	198.51.100.2
1	2	Coffs Harbour	NSW	New South Wales	AU	Australia	2450	9	198.51.100.3
2	3	Darlinghurst	NSW	New South Wales	AU	Australia	2010	9	198.51.100.4
3	4	Goulburn	NSW	New South Wales	AU	Australia	2580	9	198.51.100.5
4	5	Lane Cove	NSW	New South Wales	AU	Australia	1597	9	198.51.100.6
...	...	...	...	...	...	...	...	...	...
650	651	Mosinee	WI	Wisconsin	US	United States	54455	3	203.0.113.144
651	652	Racine	WI	Wisconsin	US	United States	53182	3	203.0.113.145
652	653	Casper	WY	Wyoming	US	United States	82601	1	203.0.113.146
653	654	Cheyenne	WY	Wyoming	US	United States	82001	1	203.0.113.147
654	655	Rock Springs	WY	Wyoming	US	United States	82901	1	203.0.113.148

655 rows × 9 columns

AWC's customers live in 655 City where the company operates. This data table shows 9 columns of data with the primary key of the table and the necessary data columns that the customer needs to provide.

```
df_DGeography.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 655 entries, 0 to 654
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   GeographyKey             655 non-null    int64
 1   City                     655 non-null    object
 2   StateProvinceCode        655 non-null    object
 3   StateProvinceName        655 non-null    object
 4   CountryRegionCode        655 non-null    object
 5   EnglishCountryRegionName 655 non-null    object
 6   PostalCode               655 non-null    object
 7   SalesTerritoryKey        655 non-null    int64
 8   IpAddressLocator         655 non-null    object
dtypes: int64(2), object(7)
memory usage: 46.2+ KB
```

Specifically, the required information is described in the table below:

Column Name	Datatype	Description
City	Object	Name of the city
StateProvinceCode	Object	ISO standard state or province code
StateProvinceName	Object	State or province description
CountryRegionCode	Object	ISO standard code for countries and regions
EnglishCountryRegionName	Object	The country or region name
PostalCode	Object	Postal code for the street address
IpAddressLocator	Object	The identifier that allows information to be sent between devices on a network

DimCustomer

```
df_DCustomer = pd.read_excel(file,'DimCustomer')
df_DCustomer.head()
```

	CustomerKey	GeographyKey	CustomerAlternateKey	FirstName	LastName	BirthDate	MaritalStatus	Gender	EmailAddress	YearlyIncome	TotalChildren	NumberChildrenAtHome	EnglishEducation	EnglishOccupation
0	11000	26	AW00011000	Jon	Yang	1971-10-06	M	M	jon24@adventure-works.com	90000	2	0	Bachelors	Professional
1	11001	37	AW00011001	Eugene	Huang	1976-05-10	S	M	eugene10@adventure-works.com	60000	3	3	Bachelors	Professional
2	11002	31	AW00011002	Ruben	Torres	1971-02-09	M	M	ruben35@adventure-works.com	60000	3	3	Bachelors	Professional
3	11003	11	AW00011003	Christy	Zhu	1973-08-14	S	F	christy12@adventure-works.com	70000	0	0	Bachelors	Professional
4	11004	19	AW00011004	Elizabeth	Johnson	1979-08-05	S	F	elizabeth5@adventure-works.com	80000	5	5	Bachelors	Professional
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18479	29479	209	AW00029479	Tommy	Tang	1969-06-30	M	M	tommy2@adventure-works.com	30000	1	0	Graduate Degree	Clerical
18480	29480	248	AW00029480	Nina	Raji	1977-05-06	S	F	nina21@adventure-works.com	30000	3	0	Graduate Degree	Clerical
18481	29481	120	AW00029481	Ivan	Suri	1965-07-04	S	M	ivan0@adventure-works.com	30000	3	0	Graduate Degree	Clerical
18482	29482	179	AW00029482	Clayton	Zhang	1964-09-01	M	M	clayton0@adventure-works.com	30000	3	0	Bachelors	Clerical
18483	29483	217	AW00029483	Jésus	Navarro	1965-06-06	M	M	jésus9@adventure-works.com	30000	0	0	Bachelors	Clerical

18484 rows × 20 columns

With a scale covering all economic regions in the world, Adventure Work has a large number of specific customers, 18484 customers (18484 rows). And in the above data table, it shows 20 columns with key columns and necessary information for the company to analyze customer buying behavior.

```
df_DCustomer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18484 entries, 0 to 18483
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   CustomerKey           18484 non-null  int64
 1   GeographyKey          18484 non-null  int64
 2   CustomerAlternateKey  18484 non-null  object
 3   FirstName             18484 non-null  object
 4   LastName              18484 non-null  object
 5   BirthDate             18484 non-null  object
 6   MaritalStatus         18484 non-null  object
 7   Gender                18484 non-null  object
 8   EmailAddress          18484 non-null  object
 9   YearlyIncome          18484 non-null  int64
 10  TotalChildren         18484 non-null  int64
 11  NumberChildrenAtHome  18484 non-null  int64
 12  EnglishEducation      18484 non-null  object
 13  EnglishOccupation     18484 non-null  object
 14  HouseOwnerFlag        18484 non-null  int64
 15  NumberCarsOwned       18484 non-null  int64
 16  AddressLine1          18484 non-null  object
 17  Phone                 18484 non-null  object
 18  DateFirstPurchase     18484 non-null  object
 19  CommuteDistance       18484 non-null  object
dtypes: int64(7), object(13)
memory usage: 2.8+ MB
```

Specifically, the required information is described in the table below:

Column Name	Datatype	Description
FirstName	Object	The first name is the name given at birth
LastName	Object	The last name (surname) represents the name of the family to which the child is born
BirthDate	Object	Date of birth.
MaritalStatus	Object	The legally defined marital state with Single (S), Married (M)
Gender	Object	The characteristics of women, men, girls and boys that are socially constructed. It includes: Male (M) and Female (F)
EmailAddress	Object	E-mail address for the person
YearlyIncome	Int64	The total value of income earned during a fiscal year
TotalChildren	Int64	A young human being below the age of puberty or below the legal age of majority.
NumberChildrenAtHome	Int64	Number of children at home
EnglishEducation	Object	The person's level of education
EnglishOccupation	Object	A person's job
HouseOwnerFlag	Int64	The number of houses the person owns
NumberCarsOwned	Int64	The number of cars the person owns
AddressLine1	Object	First street address line
Phone	Object	The phone number is to contact by mobile phone
DateFirstPurchase	Object	The date when the person started buying their first product at Adventure Works
CommuteDistance	Object	The maximum distance a worker is likely to travel each working day between the worker's residence and workplace.

DimDate

```
df_DDate = pd.read_excel(file,'DimDate')
df_DDate
```

	DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	DayNumberOfMonth	DayNumberOfYear	WeekNumberOfYear	EnglishMonthName	MonthNumberOfYear	CalendarQuarter	CalendarYear	CalendarSemester
0	20050101	2005-01-01	7	Saturday	1	1	1	January	1	1	2005	1
1	20050102	2005-01-02	1	Sunday	2	2	2	January	1	1	2005	1
2	20050103	2005-01-03	2	Monday	3	3	2	January	1	1	2005	1
3	20050104	2005-01-04	3	Tuesday	4	4	2	January	1	1	2005	1
4	20050105	2005-01-05	4	Wednesday	5	5	2	January	1	1	2005	1
...	...	...	...	...	...	...	...	...	...	...	...	...
3647	20141227	2014-12-27	7	Saturday	27	361	52	December	12	4	2014	2
3648	20141228	2014-12-28	1	Sunday	28	362	53	December	12	4	2014	2
3649	20141229	2014-12-29	2	Monday	29	363	53	December	12	4	2014	2
3650	20141230	2014-12-30	3	Tuesday	30	364	53	December	12	4	2014	2
3651	20141231	2014-12-31	4	Wednesday	31	365	53	December	12	4	2014	2

3652 rows × 15 columns

DimDate has about 3652 rows that are specific information related to dates between 2005 and 2014. And there are 15 columns that are primary key columns and information columns such as DayNumberOfWeek, EnglishDayNameOfWeek,...

```
df_DDate.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3652 entries, 0 to 3651
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   DateKey               3652 non-null   int64 
 1   FullDateAlternateKey  3652 non-null   object
 2   DayNumberOfWeek       3652 non-null   int64 
 3   EnglishDayNameOfWeek  3652 non-null   object
 4   DayNumberOfMonth      3652 non-null   int64 
 5   DayNumberOfYear       3652 non-null   int64 
 6   WeekNumberOfYear      3652 non-null   int64 
 7   EnglishMonthName      3652 non-null   object
 8   MonthNumberOfYear     3652 non-null   int64 
 9   CalendarQuarter       3652 non-null   int64 
 10  CalendarYear          3652 non-null   int64 
 11  CalendarSemester      3652 non-null   int64 
 12  FiscalQuarter         3652 non-null   int64 
 13  FiscalYear            3652 non-null   int64 
 14  FiscalSemester        3652 non-null   int64 
dtypes: int64(12), object(3)
```

Specifically, the required information is described in the table below:

Column Name	Datatype	Description
DayNumberOfWeek	Int64	The day number in a week
EnglishDayNameOfWeek	Object	The day name in a date
DayNumberOfMonth	Int64	The day number in a month
DayNumberOfYear	Int64	The day number in a year
WeekNumberOfYear	Int64	The week number in the year
EnglishMonthName	Object	The month name in a date
MonthNumberOfYear	Int64	The month number in a date
CalendarQuarter	Int64	One of the four periods of three months each of a calendar year
CalendarYear	Int64	A period of a year beginning and ending with the dates
CalendarSemester	Int64	Each period from and including January 1 through and including June 30 of each year and from and including July 1 through and including December 31 of each year
FiscalQuarter	Int64	A consecutive, three-month period within a business's fiscal year (or one-fourth of a year)
FiscalYear	Int64	A one-year period that companies and governments use for financial reporting and budgeting
FiscalSemester	Int64	A semester of the fiscal year of the Borrower (April 1-September 30 or October 1-March 31)

3.2. EDA data

Determine null values in FactSales

```
[ ]  FS_null=df_FSales.isnull().sum()
     FS_null=pd.DataFrame(FS_null,columns=['number']).sort_values(by='number',ascending=False)
     FS_null.reset_index(inplace=True)
     FS_null.head(50)
```

Find cells with null values in columns and sort those null values in order from high to low

	index	number
0	ProductKey	0
1	OrderDateKey	0
2	DueDate	0
3	OrderDate	0
4	Freight	0
5	TaxAmt	0
6	SalesAmount	0
7	TotalProductCost	0
8	ProductStandardCost	0
9	DiscountAmount	0
10	UnitPriceDiscountPct	0
11	ExtendedAmount	0

Determine null values in DimCustomer

```
[ ]  DC_null=df_DCustomer.isnull().sum()
     DC_null=pd.DataFrame(DC_null,columns=['number']).sort_values(by='number',ascending=False)
     DC_null.reset_index(inplace=True)
     DC_null.head()
```

Find cells with null values in columns and sort those null values in order from high to low

	index	number
0	CustomerKey	0
1	GeographyKey	0
2	DateFirstPurchase	0
3	Phone	0
4	AddressLine1	0

Determine null values in DimGeography

```
[ ]  DG_null=df_DGeography.isnull().sum()
     DG_null=pd.DataFrame(DG_null,columns=['number']).sort_values(by='number',ascending=False)
     DG_null.reset_index(inplace=True)
     DG_null.head()
```

Find cells with null values in columns and sort those null values in order from high to low

	index	number
0	GeographyKey	0
1	City	0
2	StateProvinceCode	0
3	StateProvinceName	0
4	CountryRegionCode	0

Determine null values in DimSalesTerritory

```
[ ]  DT_null=df_DSalesTerritory.isnull().sum()
     DT_null=pd.DataFrame(DT_null,columns=['number']).sort_values(by='number',ascending=False)
     DT_null.reset_index(inplace=True)
     DT_null.head()
```

Find cells with null values in columns and sort those null values in order from high to low

	index	number
0	SalesTerritoryKey	0
1	SalesTerritoryAlternateKey	0
2	SalesTerritoryRegion	0
3	SalesTerritoryCountry	0
4	SalesTerritoryGroup	0

Determine null values in DimDate

```
[ ]  DD_null=df_DDate.isnull().sum()
     DD_null=pd.DataFrame(DD_null,columns=['number']).sort_values(by='number',ascending=False)
     DD_null.reset_index(inplace=True)
     DD_null.head()
```

	index	number
0	DateKey	0
1	FullDateAlternateKey	0
2	DayNumberOfWeek	0
3	EnglishDayNameOfWeek	0
4	DayNumberOfMonth	0

Find cells with null values in columns and sort those null values in order from high to low

=> There are no null values in the tables in the selected dataset

3.3. RFM

3.3.1. How to calculate indicators

Determining the first purchase date, the last purchase date in the data set

```
print("Min Date",df_FSales["OrderDate"].min(), "Max Date", df_FSales["OrderDate"].max())
```

```
Min Date 2010-12-29 00:00:00 Max Date 2014-01-28 00:00:00
```

Calculating Recency: Here we are calculating recency for customers who have made a purchase with AWC company.

```
import datetime as dt

df_FSales["OrderDate"] = pd.to_datetime(df_FSales["OrderDate"])
recency = (dt.datetime(2014,2,1) - df_FSales.groupby("CustomerKey").agg({"OrderDate":"max"})).rename(columns = {"OrderDate":"Recency"})
recency["Recency"] = recency["Recency"].apply(lambda x: x.days)
recency.sort_values(by='Recency', ascending=False)
```

CustomerKey	Recency
28389	1130
27601	1126
27612	1125
27666	1122
25861	1122
...	...

Calculating Frequency: We are here calculating the frequency of frequent transactions of the customer in ordering/buying some product from the company.

```
freq = df_FSales.groupby("CustomerKey").agg({"OrderDate":"nunique"}).rename(columns={"OrderDate": "Frequency"})
freq
```

CustomerKey	Frequency
11000	3
11001	3
11002	3
11003	3
11004	3
...	...

Calculating Monetary Value: Here we are calculating the monetary value of customer spend on purchasing products from the company.

```
monetary = df_FSales.groupby("CustomerKey").agg({"SalesAmount":"sum"}).rename(columns={"SalesAmount":"Monetary"})
monetary
```

CustomerKey	Monetary
11000	8248.9900
11001	6383.8800
11002	8114.0400
11003	8139.2900
11004	8196.0100
...	...

Concat the three indexes above into a table

```
rfm = pd.concat([recency, freq, monetary],  axis=1)
rfm
```

CustomerKey	Recency	Frequency	Monetary
11000	274	3	8248.9900
11001	53	3	6383.8800
11002	343	3	8114.0400
11003	267	3	8139.2900
11004	276	3	8196.0100
...	...	...	...
29479	515	1	2049.0982
29480	199	1	2442.0300
29481	903	1	3374.9900
29482	501	1	2049.0982
29483	510	1	2049.0982

18484 rows × 3 columns

Use the qcut function to scale and mark label values

The pandas documentation describes qcut as a “Quantile-based discretization function.” This basically means that qcut tries to divide up the underlying data into equal sized bins. The function defines the bins using percentiles based on the distribution of the data, not the actual numeric edges of the bins.

```
rfm["RecencyScore"]  = pd.qcut(rfm["Recency"], 5, labels = [5, 4 , 3, 2, 1])
rfm["FrequencyScore"]= pd.qcut(rfm["Frequency"].rank(method="first"),5, labels=[1,2,3,4,5])
rfm["MonetaryScore"] = pd.qcut(rfm['Monetary'], 5, labels = [1, 2, 3, 4, 5])
rfm
```

CustomerKey	Recency	Frequency	Monetary	RecencyScore	FrequencyScore	MonetaryScore
11000	274	3	8248.9900	2	5	5
11001	53	3	6383.8800	5	5	5
11002	343	3	8114.0400	1	5	5
11003	267	3	8139.2900	2	5	5
11004	276	3	8196.0100	2	5	5
...	...	...	...	...	...	...
29479	515	1	2049.0982	1	4	4
29480	199	1	2442.0300	3	4	4
29481	903	1	3374.9900	1	4	5
29482	501	1	2049.0982	1	4	4
29483	510	1	2049.0982	1	4	4

18484 rows × 6 columns

Calculating RFM score: RFM score is calculated based upon recency, frequency, monetary value and normalized ranks. Based upon this score we divide our customers. Here we rate them on a scale of 5. Formula used for calculating rfm score is : **0.15*Recency score + 0.28*Frequency score + 0.57 *Monetary score.**

Weights can be applied equally or we can provide specific weights for each parameter based on domain knowledge or business input. Here, in the above case, we are focusing more on Frequency and Display.

```
def Score_rfm(x) : return (0.15*(x['RecencyScore']) + 0.28 *(x['FrequencyScore']) + 0.57*(x['MonetaryScore']))
rfm['score'] = rfm.apply(Score_rfm,axis=1 )
rfm.head()
```

CustomerKey	Recency	Frequency	Monetary	RecencyScore	FrequencyScore	MonetaryScore	score
11000	274	3	8248.99	2	5	5	4.55
11001	53	3	6383.88	5	5	5	5.00
11002	343	3	8114.04	1	5	5	4.40
11003	267	3	8139.29	2	5	5	4.55
11004	276	3	8196.01	2	5	5	4.55

Based on RFM score to divide customers into 5 segments:

- rfm score >4.5 : Top Customer
- 4.5 > rfm score > 4 : High Value Customer
- 4>rfm score >3 : Medium value customer
- 3>rfm score>1.6 : Low-value customer
- rfm score<1.6 :Lost Customer

```
rfm["Segment"] = np.where(rfm['score'] >
                                      4.5, "Top Customers",
                                      (np.where(
                                        rfm['score'] > 4,
                                        "High value Customer",
                                        (np.where(
    rfm['score'] > 3,
                               "Medium Value Customer",
                               np.where(rfm['score'] > 1.6,
                              'Low Value Customers', 'Lost Customers'))))))
rfm[['RecencyScore',  'FrequencyScore', 'MonetaryScore','score', 'Segment']].head(20)
```

CustomerKey	RecencyScore	FrequencyScore	MonetaryScore	score	Segment
11000	2	5	5	4.55	Top Customers
11001	5	5	5	5.00	Top Customers
11002	1	5	5	4.40	High value Customer
11003	2	5	5	4.55	Top Customers
11004	2	5	5	4.55	Top Customers
11005	2	5	5	4.55	Top Customers
11006	2	5	5	4.55	Top Customers
11007	1	5	5	4.40	High value Customer
11008	1	5	5	4.40	High value Customer
11009	2	5	5	4.55	Top Customers
11010	2	5	5	4.55	Top Customers
11011	1	5	5	4.40	High value Customer

Figure 3-1 Rating Customer based upon the RFM score

3.3.2. Customer segmentation by RFM

Use the round function to work out the number of customers in each segment forwarded

```
round(rfm.Segment.value_counts(normalize=True)*100,0)
```

```
Low Value Customers      37.0
Medium Value Customer    26.0
Lost Customers           14.0
Top Customers            12.0
High value Customer      10.0
Name: Segment, dtype: float64
```

Use the pie chart to show the segmentation of Adventureworks' customers.

The Low Value Customer segment has the most customers, accounting for 37% of the total number of customers.

The High Value Customer segment has the fewest customers, accounting for only 10% of the total number of customers.

```
[ ]  # một biểu đồ hình tròn để hiển thị tất cả các phân khúc khách hàng.
     fig = plt.pie(rfm.Segment.value_counts(), labels=rfm.Segment.value_counts().index,autopct='%.0f%%')
     plt.title('Customer segmentation by RFM method', fontsize=25)
     plt.set_cmap('winter')
```

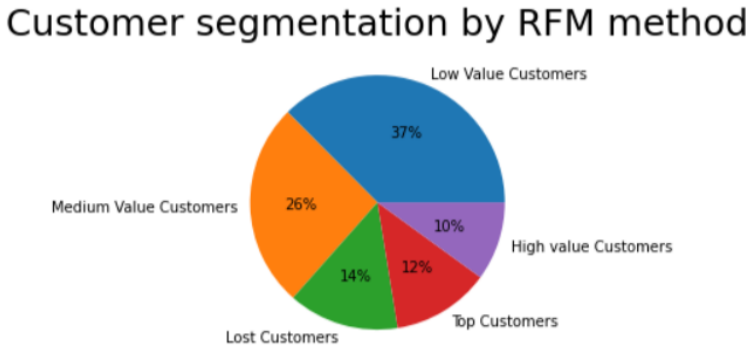


Figure 3-2 Pie *chart to show the segmentation of Adventureworks' customers.*

3.4. Clustering by K- Means

3.4.1. Handling Outliers

An outlier is a data point in a data set that is distant from all other observations. A data point that lies outside the overall distribution of the dataset.

Here the team performs outliers for Montary, Recency and Frequency

```
# Xóa outliers for Monetary
Q1 = df_kmeans.Monetary.quantile(0.05)
Q3 = df_kmeans.Monetary.quantile(0.95)
IQR = Q3 - Q1
df_kmeans = df_kmeans[(df_kmeans.Monetary >= Q1 - 1.5*IQR) & (df_kmeans.Monetary <= Q3 + 1.5*IQR)]

# Xóa outliers for Recency
Q1 = df_kmeans.Recency.quantile(0.05)
Q3 = df_kmeans.Recency.quantile(0.95)
IQR = Q3 - Q1
df_kmeans = df_kmeans[(df_kmeans.Recency >= Q1 - 1.5*IQR) & (df_kmeans.Recency <= Q3 + 1.5*IQR)]

# Xóa outliers for Frequency
Q1 = df_kmeans.Frequency.quantile(0.05)
Q3 = df_kmeans.Frequency.quantile(0.95)
IQR = Q3 - Q1
df_kmeans = df_kmeans[(df_kmeans.Frequency >= Q1 - 1.5*IQR) & (df_kmeans.Frequency <= Q3 + 1.5*IQR)]
```

3.4.2. Scaling data

Because the model's small weights are small and updated based on the prediction error, scaling the values of input X and output Y of the training dataset is an important factor. If the input is not scaled, it can lead to unstable training. In addition, if the output Y is not scaled in regression problems, it can lead to exploding gradients causing the algorithm to fail.

Data normalization is the scaling of data to a distribution where the mean of the observations is 0 and the standard deviation = 1. This technique is also known as “whitening.”. Thanks to normalization, algorithms such as linear regression, logistic regression are improved.

The normalization formula is as follows:

$$x' = \frac{x - \bar{x}}{\sigma}$$

with $\bar{x}$ and σ respectively the expectation and standard deviation of that component on the entire training data.

Standardization can be effective and even mandatory if the input data values belong to different value domains.

Standardization assumes observations with a Gaussian (bell-shaped) distribution. If the data distribution is not normally distributed, then applying standardization will not work.

To standardize the data, we need to calculate the mean and standard deviation based on the observations.

```
from sklearn.preprocessing import StandardScaler
```

Use StandardScaler to scale data.

```
# scaling các biến và lưu trữ nó trong các df khác nhau
standard_scaler = StandardScaler()
df_kmeans_norm = standard_scaler.fit_transform(df_kmeans)

# chuyển đổi nó thành khung dữ liệu
df_kmeans_norm = pd.DataFrame(df_kmeans_norm)
df_kmeans_norm.columns = ['recency','frequency','monetary']
df_kmeans_norm.head()
```

The results are returned as above.

	recency	frequency	monetary
0	0.805031	2.253885	3.149414
1	-1.145288	2.253885	2.271095
2	1.413954	2.253885	3.085863
3	0.743256	2.253885	3.097754
4	0.822681	2.253885	3.124465

3.4.3. How to find K

To determine the number of cluster clusters, we used the Elbow method.

```
ssd = []
for num_clusters in list(range(1,11)):
    model_clus = KMeans(n_clusters = num_clusters, max_iter=50)
    model_clus.fit(df_kmeans_norm)
    ssd.append(model_clus.inertia_)

# plot the SSDs for each n_clusters
plt.figure(figsize=(10,5))
plt.plot(np.arange(1,11,1), ssd)
plt.xlabel('Number of cluster', size=12)
plt.ylabel('Sum of Square Distance(SSD)', size=12)
plt.title('Elbow Curve for deciding K', size=15)
plt.show()
```

The group's results are below

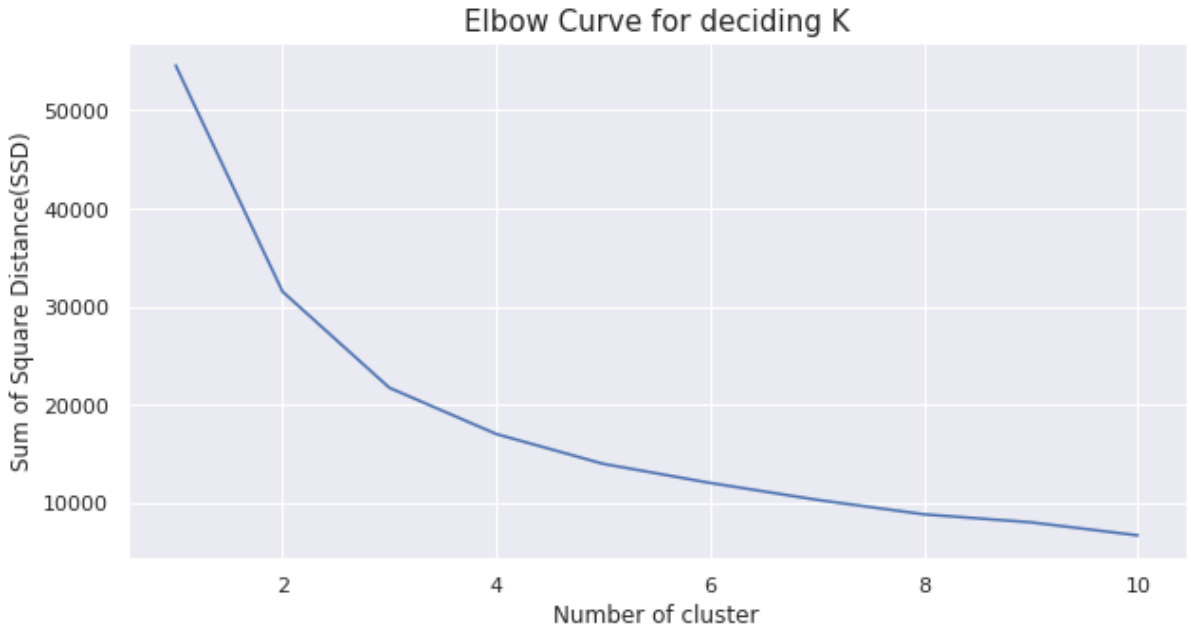


Figure 3-3 The Elbow method for deciding K

From the chart above, our team predicts k=3 based on the elbow line. However, to ensure objectivity, our team will double check with Silhouette Analysis.

Installing the Silhouette package into the library.

```
from sklearn.metrics import silhouette_score
```

```
for num_clusters in list(range(2,11)):
    # intialise kmeans
    model_clus = KMeans(n_clusters = num_clusters, max_iter=50)
    model_clus.fit(df_kmeans_norm)

    cluster_labels = model_clus.labels_

    # silhouette score
    silhouette_avg = silhouette_score(df_kmeans_norm, cluster_labels)
    print("For n_clusters={0}, the silhouette score is {1}".format(num_clusters, silhouette_avg))
```

And the results that the team achieved from this method

```
For n_clusters=2, the silhouette score is 0.4595866991194895
For n_clusters=3, the silhouette score is 0.41195789572738456
For n_clusters=4, the silhouette score is 0.43307611648752714
For n_clusters=5, the silhouette score is 0.4547766482306084
For n_clusters=6, the silhouette score is 0.4586333380077158
For n_clusters=7, the silhouette score is 0.4695171372390776
For n_clusters=8, the silhouette score is 0.4779681508725761
For n_clusters=9, the silhouette score is 0.42473893926981815
For n_clusters=10, the silhouette score is 0.44085480437018854
```

From the Silhouette analysis, we find the best cluster value when the number of clusters will be 3.

Next, the team continues to build a model with a cluster number of 3.

```
#Kmeans với k =3
model_clus3 = KMeans(n_clusters = 3)
model_clus3.fit(df_kmeans_norm)
```

```
KMeans(n_clusters=3)
```

Add Cluster column to model

```
df_kmeans['Clusters'] = model_clus3.labels_
df_kmeans.head(20)
```

CustomerKey	Recency	Frequency	Monetary	Clusters
11000	274	3	8248.99	1
11001	53	3	6383.88	1
11002	343	3	8114.04	1
11003	267	3	8139.29	1
11004	276	3	8196.01	1
11005	275	3	8121.33	1
11006	263	3	8119.03	1
11007	319	3	8211.00	1
11008	336	3	8106.31	1
11009	268	3	8091.33	1
11010	254	3	8088.04	1
11011	319	3	8133.04	1
11012	109	2	81.26	2
11013	11	2	113.96	2
11014	277	2	138.45	0
11015	379	1	2500.97	0
11016	357	1	2332.28	0
11017	110	3	6434.31	1
11018	100	3	6533.28	1
11020	399	1	2316.97	0

Show 3 clusters with Monetary, Frequency, Recency by boxplot to better illustrate for viewers

```
column = ['Recency','Frequency','Monetary']
plt.figure(figsize=(15,4))
for i,j in enumerate(column):
    plt.subplot(1,3,i+1)
    sns.boxplot(y=df_kmeans[j], x=df_kmeans['Clusters'], palette='spring')
    plt.title('{} wrt clusters'.format(j.upper()), size=13)
    plt.ylabel('')
    plt.xlabel('')

plt.show()
```

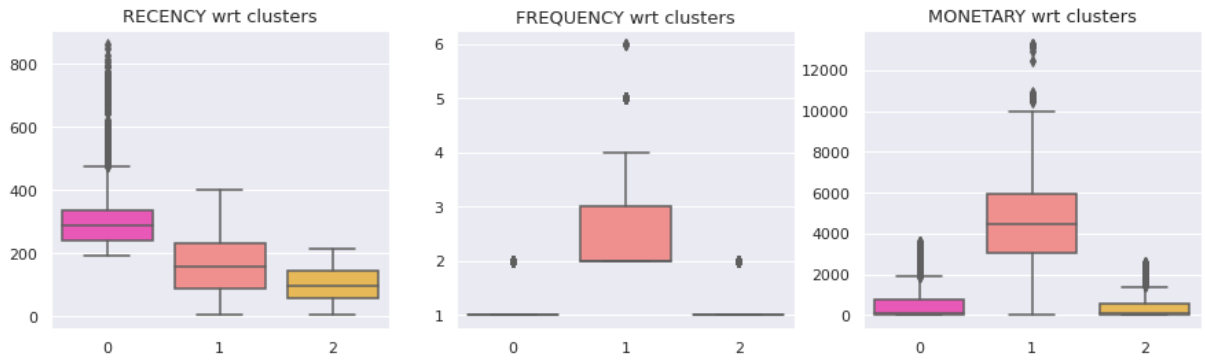


Figure 3-4 The box plot of 3 clusters with Monetary, Recency, Frequency

Represented by 3D model

```
import plotly.express as px
fig = px.scatter_3d(df_kmeans, x=df_kmeans["Recency"], y=df_kmeans["Frequency"], z=df_kmeans["Monetary"], color=df_kmeans["Clusters"])
fig.show()
```

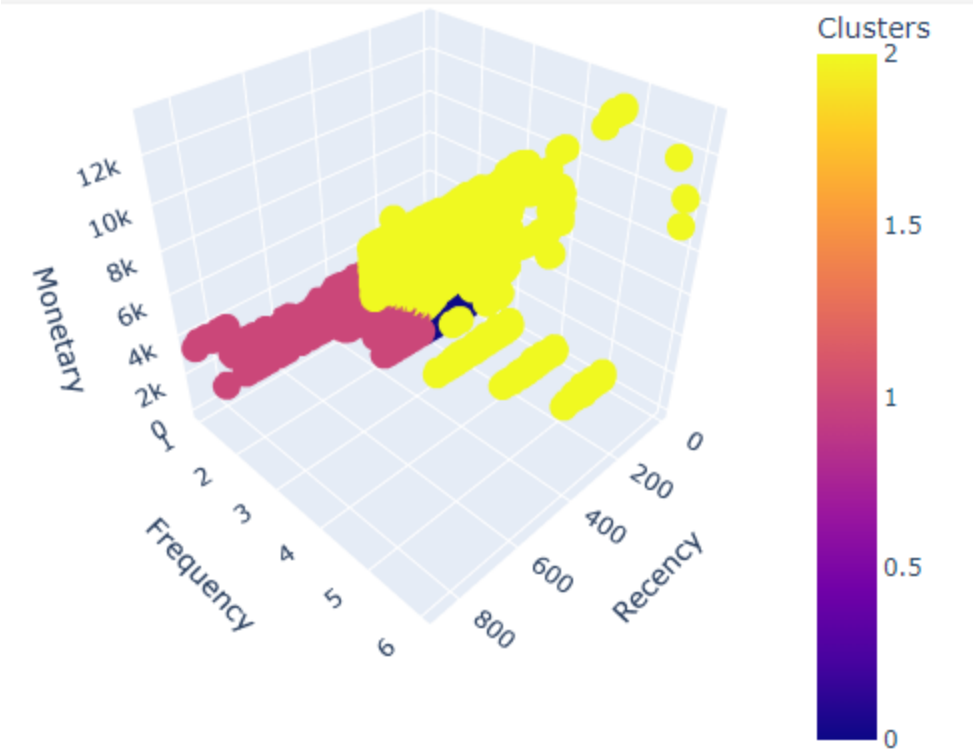


Figure 3-5 The 3D scatter model

3.4.4. Customer Segmentation by K-Means

Number of elements (customers) in each cluster (cluster)

```
round(df_kmeans.Clusters.value_counts())
```

```
2    7385
0    5892
1    4918
Name: Clusters, dtype: int64
```

Percentage (customers) by cluster

```
round(df_kmeans.Clusters.value_counts(normalize=True)*100,0)
```

```
2    41.0
0    32.0
1    27.0
Name: Clusters, dtype: float64
```

Below will be a detailed analysis of each cluster

Firstly, the cluster is 0

```
df_0 = df_kmeans[df_kmeans['Clusters']==0]
df_0
```

CustomerKey	Recency	Frequency	Monetary	Clusters
11014	277	2	138.4500	0
11015	379	1	2500.9700	0
11016	357	1	2332.2800	0
11020	399	1	2316.9700	0
11021	374	1	2371.9600	0
...	...	...	...	...
29478	220	1	2398.0500	0
29479	515	1	2049.0982	0
29480	199	1	2442.0300	0
29482	501	1	2049.0982	0
29483	510	1	2049.0982	0

5892 rows × 4 columns

According to the K-means method performed above, we can see that there are 5892 customers in cluster 0.

```
#Mô tả tứ phân vị của cụm 0
df_0.describe()
```

	Recency	Frequency	Monetary	Clusters
count	5892.000000	5892.000000	5892.000000	5892.0
mean	297.989138	1.082145	573.175774	0.0
std	82.346154	0.274609	837.291928	0.0
min	195.000000	1.000000	2.290000	0.0
25%	242.000000	1.000000	35.720000	0.0
50%	287.000000	1.000000	75.480000	0.0
75%	335.000000	1.000000	782.990000	0.0
max	864.000000	2.000000	3578.270000	0.0

Secondly, the cluster is 1.

```
df_1 = df_kmeans[df_kmeans['Clusters']==1]
df_1
```

CustomerKey	Recency	Frequency	Monetary	Clusters
11000	274	3	8248.9900	1
11001	53	3	6383.8800	1
11002	343	3	8114.0400	1
11003	267	3	8139.2900	1
11004	276	3	8196.0100	1
...	...	...	...	...
29398	358	2	3964.4700	1
29399	328	2	3936.4700	1
29400	322	2	3992.4600	1
29403	304	2	3991.9400	1
29412	316	2	2671.8796	1

4918 rows × 4 columns

For cluster 1 we have 4918 customers.

```
#Mô tả tứ phân vị của cụm 1
df_1.describe()
```

	Recency	Frequency	Monetary	Clusters
count	4918.000000	4918.000000	4918.000000	4918.0
mean	168.086824	2.331842	4372.696903	1.0
std	94.643512	0.595631	2049.885598	0.0
min	4.000000	2.000000	34.560000	1.0
25%	89.000000	2.000000	3084.022500	1.0
50%	158.000000	2.000000	4462.158200	1.0
75%	234.000000	3.000000	5923.230000	1.0
max	400.000000	6.000000	13295.380000	1.0

Finally, the cluster is 2.

```
df_2 = df_kmeans[df_kmeans['Clusters']==2]
df_2
```

CustomerKey	Recency	Frequency	Monetary	Clusters
11012	109	2	81.26	2
11013	11	2	113.96	2
11023	18	2	122.24	2
11024	190	2	56.51	2
11043	167	2	47.98	2
...	...	...	...	...
29464	109	1	756.33	2
29465	104	1	791.32	2
29470	58	1	60.47	2
29471	11	1	23.78	2
29472	153	1	88.98	2

7385 rows × 4 columns

7385 customers in cluster 2

```
#Mô tả tứ phân vị của cụm 2
df_2.describe()
```

	Recency	Frequency	Monetary	Clusters
count	7385.000000	7385.000000	7385.000000	7385.0
mean	100.642383	1.192146	477.213639	2.0
std	52.751937	0.394014	731.460511	0.0
min	4.000000	1.000000	2.290000	2.0
25%	57.000000	1.000000	38.920000	2.0
50%	97.000000	1.000000	78.960000	2.0
75%	145.000000	1.000000	588.960000	2.0
max	213.000000	2.000000	2564.920000	2.0

Sorting customer segments by K-means and adding labels for each of clusters.

According to the quartile table of the 3 clusters shown above, the group will classify those 3 clusters with labels such as High value Customers, Medium value Customers, Low Value Customers.

Table 3-1 The maximum about Monetary, Frequency, Recency with 3 clusters

Cluster	**Max Montery**	**Max Frequency**	**Max Recency**	**Label**
0	3578.27	2	864	Medium value Customers
1	13295.38	6	400	High value Customers
2	2564.92	2	213	Low value Customers

From the analysis table above the group has inferred customer segmentation by 3 labels.

```
df_kmeans["Segment_new"] = np.where(df_kmeans['Clusters'] ==
                                    1, "High value Customers",
                                    (np.where(
                                      df_kmeans['Clusters'] == 2,
                                      "Medium value Customers",
                                       "Low Value Customers")))
df_kmeans[['Recency', 'Frequency',  'Monetary','Clusters','Segment_new']].head(20)
```

And the result was achieved with the first 20 lines.

CustomerKey	Recency	Frequency	Monetary	Clusters	Segment_new
11000	274	3	8248.99	1	High value Customers
11001	53	3	6383.88	1	High value Customers
11002	343	3	8114.04	1	High value Customers
11003	267	3	8139.29	1	High value Customers
11004	276	3	8196.01	1	High value Customers
11005	275	3	8121.33	1	High value Customers
11006	263	3	8119.03	1	High value Customers
11007	319	3	8211.00	1	High value Customers
11008	336	3	8106.31	1	High value Customers
11009	268	3	8091.33	1	High value Customers
11010	254	3	8088.04	1	High value Customers
11011	319	3	8133.04	1	High value Customers
11012	109	2	81.26	2	Low value Customers
11013	11	2	113.96	2	Low value Customers
11014	277	2	138.45	0	Medium Value Customers
11015	379	1	2500.97	0	Medium Value Customers
11016	357	1	2332.28	0	Medium Value Customers
11017	110	3	6434.31	1	High value Customers
11018	100	3	6533.28	1	High value Customers

Visualization data from analyzing 3 customer segments in a new way.

```
# a pie plot
fig = plt.pie(df_kmeans.Segment_new.value_counts(), labels=df_kmeans.Segment_new.value_counts().index,autopct='%.0f%%')
plt.title('Customer segmentation by K-Means clustering', fontsize=25)
plt.set_cmap('winter')
```

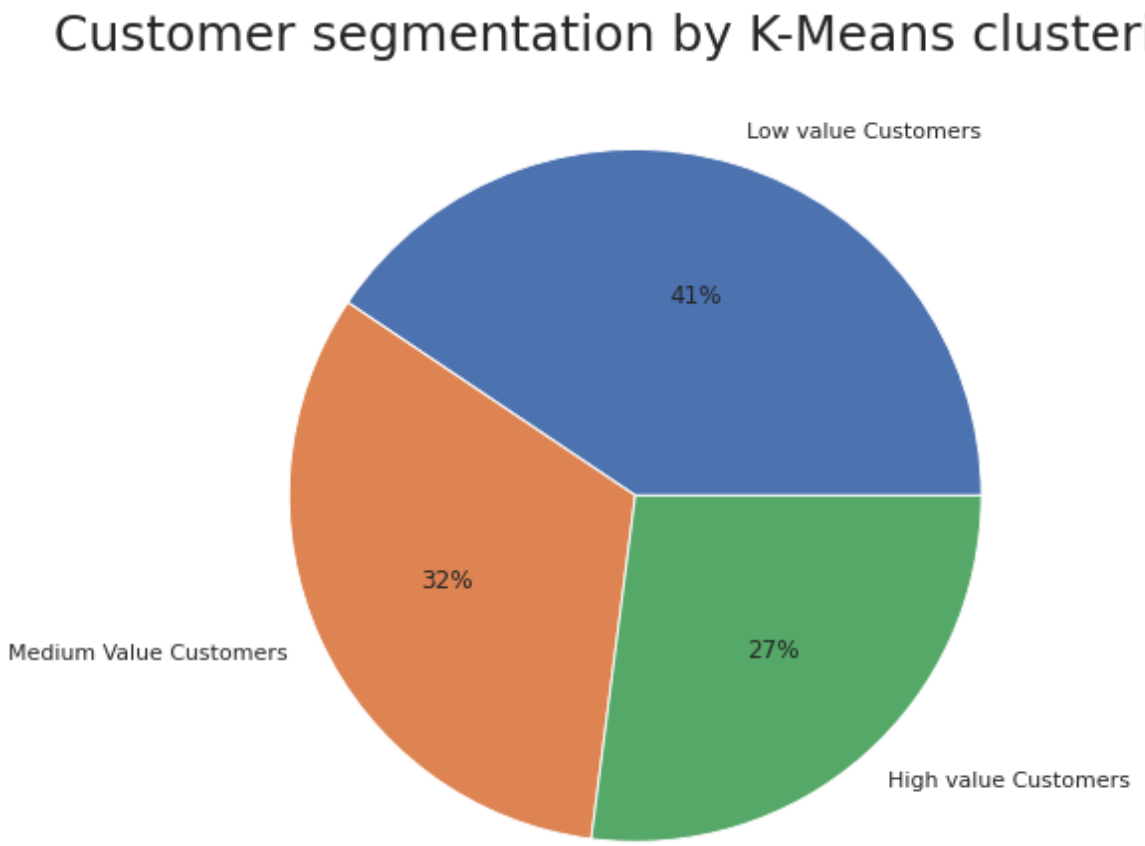


Figure 3-6 Customer segmentation by K-means clustering

CHAPTER 4 EXPERIMENTATION AND ANALYSIS

Provide data visualization charts. thereby giving an analysis of the customer segment of each model. Find out the salient features of each segment based on variables like geography, age, gender, occupation and income based on data visualization. And analyze customer churn rate and customer retention rate.

4.1. Traditional RFM data visualization and customer analytics

Merge the rfm, df_DCustomer, df_DGeography, df_FSales tables into one dataframe.

```
# merge bảng rfm và bảng DCustomer
df_data1 = pd.merge(df_DCustomer,rfm, on = ['CustomerKey'] )
df_data1.head(5)
```

	CustomerKey	GeographyKey	CustomerAlternateKey	FirstName	LastName	BirthDate	MaritalStatus	Gender	EmailAddress	YearlyIncome	...	DateF:
0	11000	26	AW00011000	Jon	Yang	1971-10-06	M	M	jon24@adventure-works.com	90000	...	
1	11001	37	AW00011001	Eugene	Huang	1976-05-10	S	M	eugene10@adventure-works.com	60000	...	
2	11002	31	AW00011002	Ruben	Torres	1971-02-09	M	M	ruben35@adventure-works.com	60000	...	
3	11003	11	AW00011003	Christy	Zhu	1973-08-14	S	F	christy12@adventure-works.com	70000	...	
4	11004	19	AW00011004	Elizabeth	Johnson	1979-08-05	S	F	elizabeth5@adventure-works.com	80000	...	

5 rows × 28 columns

Merge rfm and df_DCustomer table

```
# merge bảng vừa tạo ở trên với bảng DGeography
df_data2 = pd.merge(df_data1,df_DGeography, on = ['GeographyKey'] )
df_data2.head(5)
```

	CustomerKey	GeographyKey	CustomerAlternateKey	FirstName	LastName	BirthDate	MaritalStatus	Gender	EmailAddress	YearlyIncome	...	score	S
0	11000	26	AW00011000	Jon	Yang	1971-10-06	M	M	jon24@adventure-works.com	90000	...	4.55	Cus
1	11360	26	AW00011360	Tyrone	Serrano	1984-01-30	S	M	tyrone15@adventure-works.com	10000	...	5.00	Cus
2	11368	26	AW00011368	Edward	Miller	1984-05-24	S	M	edward28@adventure-works.com	10000	...	5.00	Cus
3	11371	26	AW00011371	Lacey	Jai	1983-11-24	M	F	lacey1@adventure-works.com	20000	...	3.01	Cus
4	11909	26	AW00011909	Nichole	She	1971-08-10	M	F	nichole0@adventure-works.com	80000	...	4.70	Cus

5 rows × 36 columns

Merge rfm, df_DCustomer and df_DGeography table

```
#merge bảng data2 với bảng FactSales
df_data3 = pd.merge(df_data2,df_FSales, on = ['CustomerKey'] )
df_data3.head(5)
```

	CustomerKey	GeographyKey	CustomerAlternateKey	FirstName	LastName	BirthDate	MaritalStatus	Gender	EmailAddress	YearlyIncome	...	UnitPriceDisc
0	11000	26	AW00011000	Jon	Yang	1971-10-06	M	M	jon24@adventure-works.com	90000	...	
1	11000	26	AW00011000	Jon	Yang	1971-10-06	M	M	jon24@adventure-works.com	90000	...	
2	11000	26	AW00011000	Jon	Yang	1971-10-06	M	M	jon24@adventure-works.com	90000	...	
3	11000	26	AW00011000	Jon	Yang	1971-10-06	M	M	jon24@adventure-works.com	90000	...	
4	11000	26	AW00011000	Jon	Yang	1971-10-06	M	M	jon24@adventure-works.com	90000	...	

5 rows × 59 columns

Merge rfm, df_DCustomer,df_DGeography and df_FSales table

Now the dataframe after merging is df_data3, then proceed to add new columns to df_data3 to serve the analysis.

Calculate and create an Age column, then divide these ages into different age groups (including Under 30, 30-39, 40-49, 50-59, 60+)

```
# Tạo cột mới cho cột tuổi
df_data3['BirthDate']= pd.to_datetime(df_data3['BirthDate'])
import datetime
CURRENT_TIME = datetime.datetime.now()
def get_age(birth_date,today=CURRENT_TIME):
    y=today-birth_date
    return y.days//365
df_data3['Age']=df_data3['BirthDate'].apply(lambda x: get_age(x))

df_data3.head(10)
```

ate	MaritalStatus	Gender	EmailAddress	YearlyIncome	...	DiscountAmount	ProductStandardCost	TotalProductCost	SalesAmount	TaxAmt	Freight	OrderDate	DueDate	ShipDate	Age
-10-06	M	M	jon24@adventure-works.com	90000	...	0	1912.1544	1912.1544	3399.99	271.9992	84.9998	2011-01-19	2011-01-31	2011-01-26	50
-10-06	M	M	jon24@adventure-works.com	90000	...	0	1265.6195	1265.6195	2319.99	185.5992	57.9998	2013-01-18	2013-01-30	2013-01-25	50
-10-06	M	M	jon24@adventure-works.com	90000	...	0	8.2205	8.2205	21.98	1.7584	0.5495	2013-01-18	2013-01-30	2013-01-25	50
-10-06	M	M	jon24@adventure-works.com	90000	...	0	1481.9379	1481.9379	2384.07	190.7256	59.6018	2013-05-03	2013-05-15	2013-05-10	50

Creating Age column

```
# Tạo nhóm tuổi mới
def cohort(Age):
    if Age < 30:
        return 'Under 30'
    elif Age <= 40:
        return '30-39'
    elif Age <= 50:
        return '40-49'
    elif Age < 60:
        return '50-59'
    else:
        return "60+"

df_data3['Age_group'] = df_data3['Age'].apply(cohort)
df_data3
```

LastName	BirthDate	MaritalStatus	Gender	EmailAddress	YearlyIncome	...	TotalProductCost	SalesAmount	TaxAmt	Freight	OrderDate	DueDate	ShipDate	profit	Age	Age_group
Yang	1971-10-06	M	M	jon24@adventure-works.com	90000	...	1912.1544	3399.99	271.9992	84.9998	2011-01-19	2011-01-31	2011-01-26	1215.8364	50	40-49
Yang	1971-10-06	M	M	jon24@adventure-works.com	90000	...	1265.6195	2319.99	185.5992	57.9998	2013-01-18	2013-01-30	2013-01-25	868.7713	50	40-49
Yang	1971-10-06	M	M	jon24@adventure-works.com	90000	...	8.2205	21.98	1.7584	0.5495	2013-01-18	2013-01-30	2013-01-25	12.0011	50	40-49

Creating Age_group column

Create a column of net profit earned. Net profit is calculated by the formula: SalesAmount - TotalProductCost - TaxAmt

```
df_data3['profit'] = df_data3['SalesAmount'] - df_data3['TotalProductCost'] - df_data3['TaxAmt']
df_data3.head(10)
```

aritalStatus	Gender	EmailAddress	YearlyIncome	...	DiscountAmount	ProductStandardCost	TotalProductCost	SalesAmount	TaxAmt	Freight	OrderDate	DueDate	ShipDate	profit
M	M	jon24@adventure-works.com	90000	...	0	1912.1544	1912.1544	3399.99	271.9992	84.9998	2011-01-19	2011-01-31	2011-01-26	1215.8364
M	M	jon24@adventure-works.com	90000	...	0	1265.6195	1265.6195	2319.99	185.5992	57.9998	2013-01-18	2013-01-30	2013-01-25	868.7713
M	M	jon24@adventure-works.com	90000	...	0	8.2205	8.2205	21.98	1.7584	0.5495	2013-01-18	2013-01-30	2013-01-25	12.0011
M	M	jon24@adventure-works.com	90000	...	0	1481.9379	1481.9379	2384.07	190.7256	59.6018	2013-05-03	2013-05-15	2013-05-10	711.4065

Creating profit column

Retrieve the critical columns needed for the analysis and create a dataframe "df_analysis" from those columns.

```
#Lấy ra những cột cần để phân tích từ bảng đã merge phía trên
important_cols=['CustomerKey','FirstName','LastName','City','StateProvinceName','EnglishCountryRegionName',
                'Age','Age_group','EnglishEducation','EnglishOccupation','Gender','MaritalStatus','YearlyIncome','Recency','Frequency','Monetary','score','Segment','SalesAmount','prof
```

```
# tạo bảng với những cột vừa lấy ra
df_analysis=df_data3[important_cols]
df_analysis
```

	CustomerKey	FirstName	LastName	City	StateProvinceName	EnglishCountryRegionName	Age	Age_group	EnglishEducation	EnglishOccupation	Gender	MaritalStatus	YearlyInc
0	11000	Jon	Yang	Rockhampton	Queensland	Australia	50	40-49	Bachelors	Professional	M	M	90
1	11000	Jon	Yang	Rockhampton	Queensland	Australia	50	40-49	Bachelors	Professional	M	M	90
2	11000	Jon	Yang	Rockhampton	Queensland	Australia	50	40-49	Bachelors	Professional	M	M	90
3	11000	Jon	Yang	Rockhampton	Queensland	Australia	50	40-49	Bachelors	Professional	M	M	90
4	11000	Jon	Yang	Rockhampton	Queensland	Australia	50	40-49	Bachelors	Professional	M	M	90
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Creating df_analysis table

After having all the necessary columns, we begin to visualize the data to analyze the characteristics of each segment.

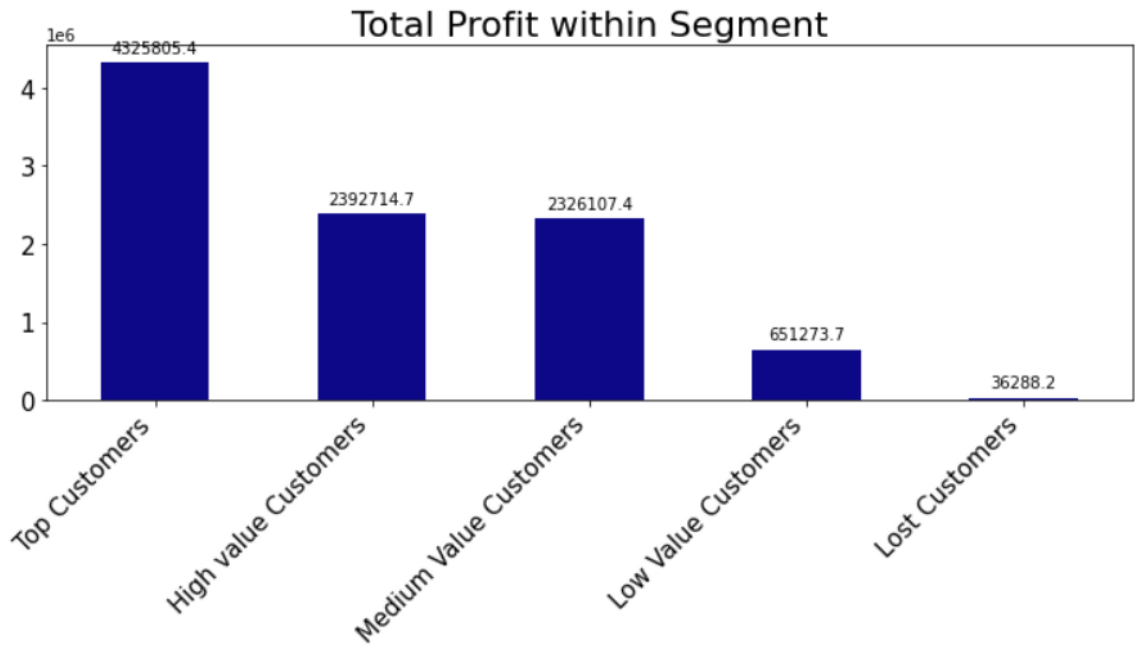


Figure 4-1 Total Profit within Segment

Looking at the chart we can see that the customer segment that contributes the most to the company is the Top Customers segment.

Next step up to visualize the data with charts:

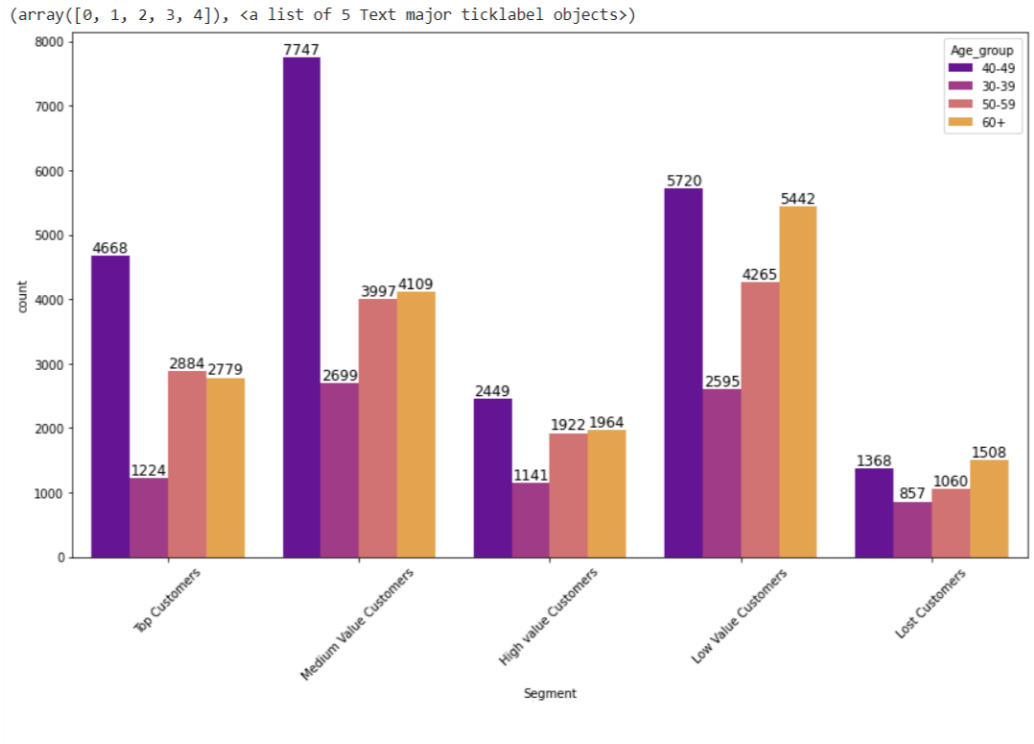


Figure 4-2 Distribution of age groups in each segment

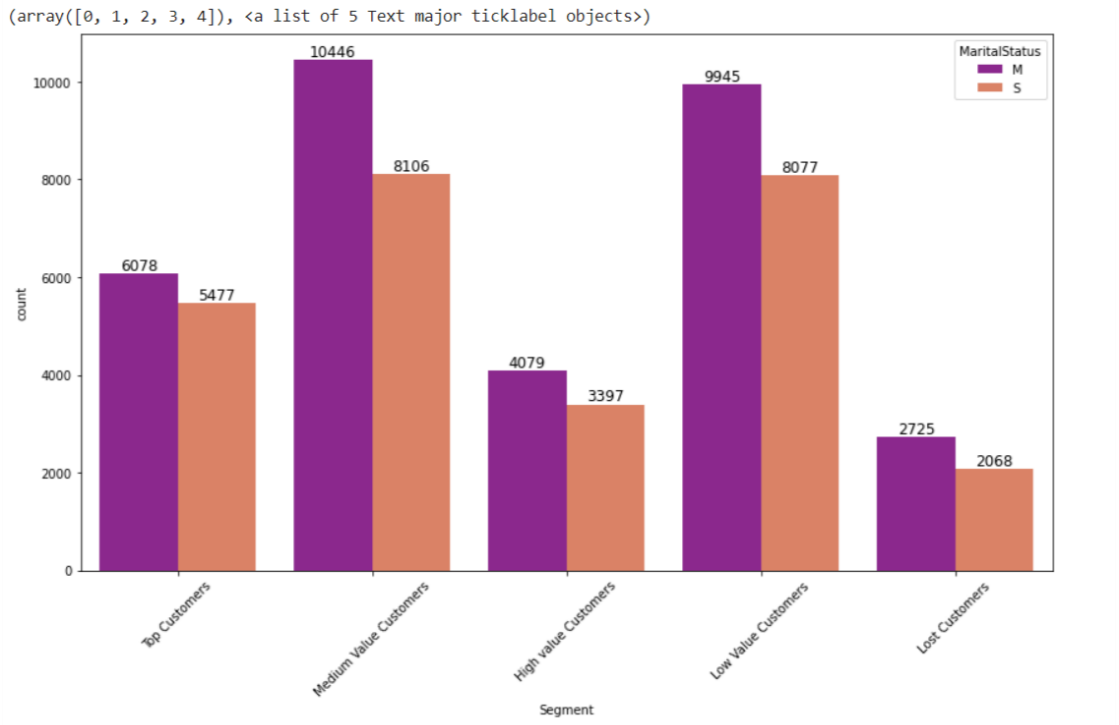


Figure 4-3 Show marital status in each segment

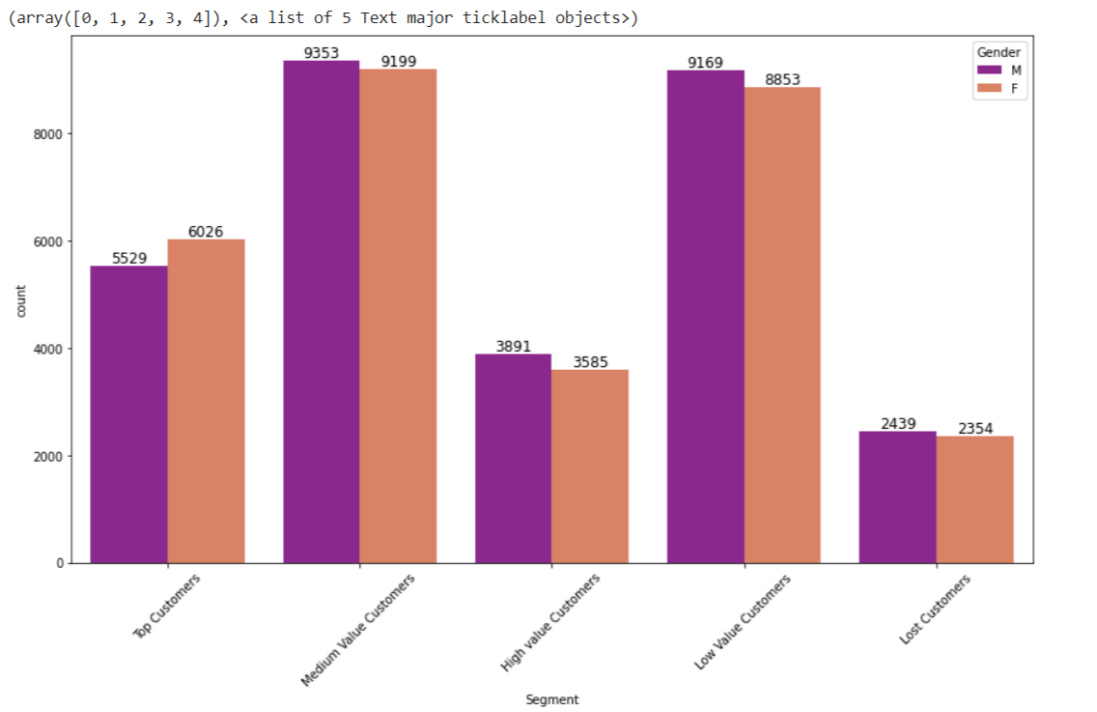


Figure 4-4 Show sex in each segment

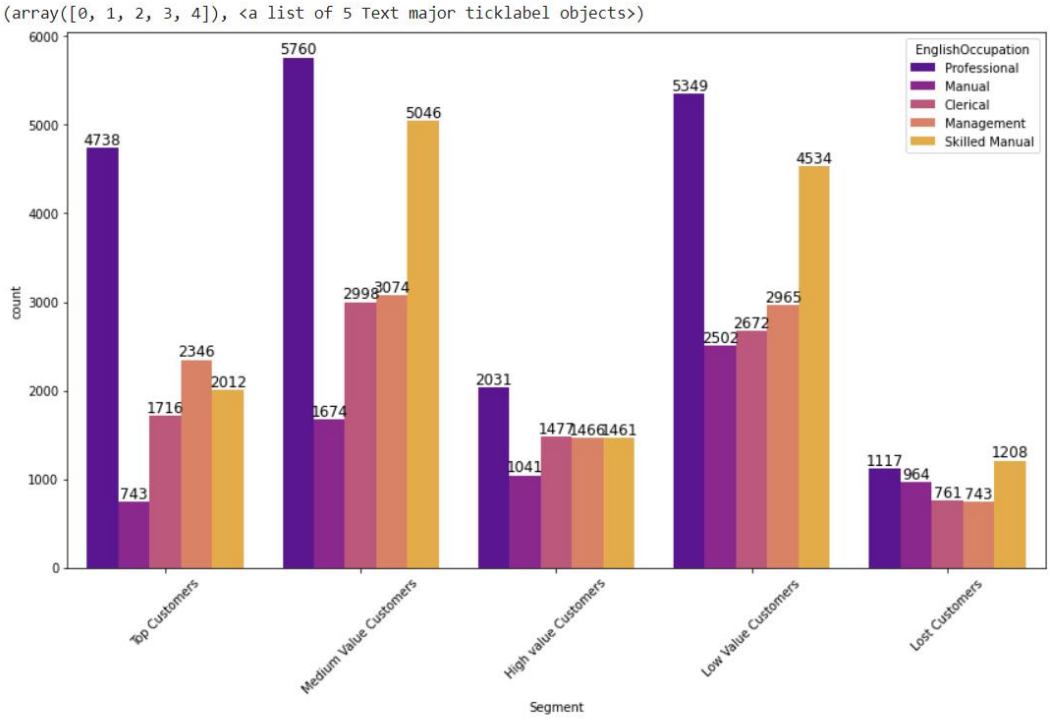


Figure 4-5 Distribution of occupations in each segment

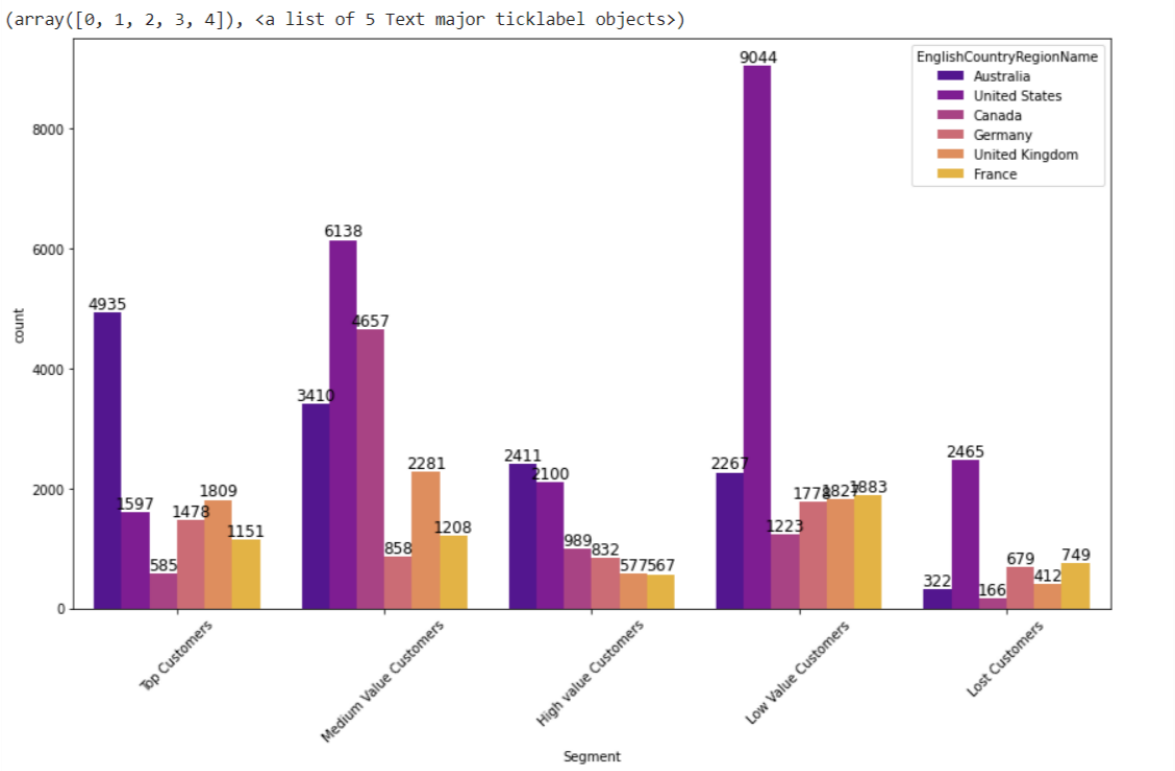


Figure 4-6 Country distribution in each segment

Calculate the average column Age, YearlyIncome, and Monetary (Monetary is the order value here is understood as customer spending)

```
#tính trung bình các cột Age, YearlyIncome, và Monetary (là giá trị đơn hàng ở đây được hiểu như chi tiêu của khách hàng)
df_avgdata = df_avgdata.groupby(['Segment'], as_index=False).mean()
df_avgdata
```

	Segment	Age	YearlyIncome	Monetary
0	High value Customers	52.673488	58184.858213	3822.074874
1	Lost Customers	53.721260	51139.161277	30.155429
2	Low Value Customers	53.607757	55640.883365	294.062634
3	Medium Value Customers	51.499299	58189.952566	1186.373188
4	Top Customers	52.143228	73065.339680	6068.560908

Average of Age, YearlyIncome, Monetary

From the chart built above, we can show the characteristics of each segment as follows

Segment 1: Top Customers *(High annual income, high spending total)*

This segment consists of middle-aged individuals who worked up a significant amount of wealth in their initial years. They also have a large spending scale and hence lead a very affluent lifestyle.

The majority of people in this segment are married, have children, and are financially secure. As a result, they are interested in purchasing bicycles for health training as well as bicycles for their children.

These people are presumed to make serious financial commitments out of all clusters due to their high spending capacity.

- This segment's most purchasing age group is **40-49**;
- The average age is **52** years;
- Predominantly **female and married**;
- This segment's clients are mostly **professional**.
- In this segment, the number of customers from **Australia** is the highest.
- Average Annual Income is **$73065** in dollars;
- Average Spending Total is **$6068**.

Segment 2: High Value Customers *(intermediate annual income, high spending total)*

This segment focuses on customers who have a sufficient but not excessive income. However, the level of spending on Adventurework products is extremely high. If customers in the Top Customer segment spend approximately 8% of their income on company products, customers in this segment spend approximately 6% of their income. As a result, this segment contributes significantly to the company's revenue and profit.

- This segment's most purchasing age group is **40-49**;
- The average age is **52** years;
- Predominantly **Male and married**;
- This segment's clients are mostly **professional**.
- In this segment, the number of customers from **Australia** is the highest.
- Average Annual Income is **$58184** in dollars;
- Average Spending Total is **$3822**.

Segment 3: Medium Value Customers *(intermediate annual income, intermediate spending total)*

This cluster consists of middle-aged customers who spend and earn money on an intermediate level.

They are careful with their spending scale as their income levels are not excessive.

These people might also be the ones with higher financial responsibilities. For instance, higher education for their children.

Suggestion: Discounts offers, Promo codes, loyalty cards, etc.

- This segment's most purchasing age group is **40-49**;
- The average age is **51** years;
- Predominantly **Male and married**;
- This segment's clients are mostly **professional**.
- In this segment, the number of customers from **the United States** is the highest.

- Average Annual Income is **$58189** in dollars;
- Average Spending Total is **$1186**.

Segment 4: Low Value Customers *(intermediate annual income, low spending total)*

This cluster consists of middle-aged people who are frugal about their spending habits.

Although the income levels are quite good, they spend very little. This might be an indicator of their financial responsibilities.

Suggestion: Membership cards, discount coupons, offers, etc could have a drastic impact on this cluster.

- This segment's most purchasing age group is **40-49**;
- The average age is **53** years;
- Predominantly **Male and married**;
- This segment's clients are mostly **professional**.
- In this segment, the number of customers from **the United States** is the highest.
- Average Annual Income is **$55640** in dollars;
- Average Spending Total is **$294**.

Segment 5: Lost Customers *(low annual income, low spending total)*

This segment consists of the older population who earn and spend less. They might be saving up for retirement.

Suggestions: Healthcare-related products can be promoted amongst this cluster. Usage of adequate discount coupons, promo codes, etc might also help.

- This segment's most purchasing age group is **40-49**;
- The average age is **53** years;
- Predominantly **Male and married**;
- This segment's clients are mostly **Skilled Manual**.
- In this segment, the number of customers from **the United States** is the highest.
- Average Annual Income is **$51139** in dollars;
- Average Spending Total is **$30**.

4.2. Data visualization and customer analysis by K-means clustering

First of all, our team will take the related columns to visualize the data into the same table as df_Kmean by keys:

```
df_Cus_Kmean = df_DCustomer[['CustomerKey','BirthDate','Gender','MaritalStatus','YearlyIncome','GeographyKey','EnglishOccupation']]
df_Geo_Kmean=df_DGeography[['GeographyKey','EnglishCountryRegionName']]

df1_Kmean=df_kmeans.merge(df_Cus_Kmean,on=(['CustomerKey']))
df_Kmean=df1_Kmean.merge(df_Geo_Kmean, on=(['GeographyKey']))

df_Kmean["BirthDate"] = pd.to_datetime(df["BirthDate"])
df_Kmean['Age']= 2014-df_Kmean['BirthDate'].dt.year
df_Kmean
```

	CustomerKey	Recency	Frequency	Monetary	Clusters	Segment_new	BirthDate	Gender	MaritalStatus	YearlyIncome	GeographyKey	EnglishOccupation	EnglishCountryRegionName
0	11000	274	3	8248.9900	1	High value Customers	1971-10-06	M	M	90000	26	Professional	Australia
1	11360	41	3	5025.8000	1	High value Customers	1976-05-10	M	S	10000	26	Manual	Australia
2	11368	62	3	5106.2400	1	High value Customers	1971-02-09	M	S	10000	26	Manual	Australia
3	11371	35	2	74.9800	2	Low value Customers	1973-08-14	F	M	20000	26	Clerical	Australia
4	11909	184	3	8105.3100	1	High value Customers	1979-08-05	F	M	80000	26	Management	Australia
...	...	...	...	...	...	...	...	...	...	...	...	...	...
18190	27040	364	1	7.2800	0	Medium Value Customers	1975-02-08	M	M	20000	524	Clerical	United States

Moreover, our team calculates the "profit" column to facilitate data visualization later:

```
[ ] #tính cột profit
    df_Kmean['profit_new'] = df_FSalesProfit['SalesAmount'] - df_FSalesProfit['TotalProductCost'] - df_FSalesProfit['TaxAmt']
    df_Kmean.head(10)
```

	CustomerKey	Recency	Frequency	Monetary	Clusters	Segment_new	BirthDate	Gender	MaritalStatus	YearlyIncome	GeographyKey	EnglishOccupation	EnglishCountryRegionName	Age
0	11000	274	3	8248.9900	1	High value Customers	1971-10-06	M	M	90000	26	Professional	Australia	43
1	11360	41	3	5025.8000	1	High value Customers	1976-05-10	M	S	10000	26	Manual	Australia	38
2	11368	62	3	5106.2400	1	High value Customers	1971-02-09	M	S	10000	26	Manual	Australia	43
3	11371	35	2	74.9800	2	Low value Customers	1973-08-14	F	M	20000	26	Clerical	Australia	41
4	11909	184	3	8105.3100	1	High value Customers	1979-08-05	F	M	80000	26	Management	Australia	35
5	11919	42	3	8171.5200	1	High value Customers	1976-08-01	M	M	60000	26	Professional	Australia	38
6	12036	116	3	69.5500	1	High value Customers	1976-12-02	F	M	40000	26	Management	Australia	38
7	12671	156	3	6802.1282	1	High value Customers	1969-11-06	M	S	70000	26	Professional	Australia	45
8	12674	150	3	6829.4696	1	High value Customers	1975-07-04	M	M	90000	26	Professional	Australia	39
9	13013	90	3	256.2400	1	High value Customers	1969-09-29	M	M	60000	26	Professional	Australia	45

```
[ ] df_FSalesProfit=df_FSales[['CustomerKey','SalesAmount','TotalProductCost','TaxAmt']]
```

+ Mã + Văn bản

```
[ ] df_FSalesProfit
```

	CustomerKey	SalesAmount	TotalProductCost	TaxAmt
0	21768	3578.2700	2171.2942	286.2616
1	28389	3399.9900	1912.1544	271.9992
2	25863	3399.9900	1912.1544	271.9992
3	14501	699.0982	413.1463	55.9279
4	11003	3399.9900	1912.1544	271.9992
...	...	...	...	...
60393	15868	21.9800	8.2205	1.7584
60394	15868	8.9900	6.9223	0.7192
60395	18759	21.9800	8.2205	1.7584
60396	18759	159.0000	59.4660	12.7200
60397	18759	8.9900	6.9223	0.7192

60398 rows × 4 columns

```
# trực quan hóa Total Profit theo Segment
segment_profit_new =df_Kmean.groupby('Segment_new').sum().sort_values('profit_new',ascending=False).iloc[:,8]
plt.figure(figsize=(10,6), tight_layout=True)
fig = segment_profit_new.plot(kind='bar', cmap='plasma', fontsize=15)
plt.xticks(rotation=45, ha='right')
plt.xlabel(' ')
plt.title('Total Profit within New Segment', fontsize=22)
for p in fig.patches:
    fig.annotate(format(p.get_height(), '.1f'),
                 (p.get_x() + p.get_width() / 2., p.get_height()),
                 ha = 'center', va = 'center',
                 xytext = (0, 9),
                 textcoords = 'offset points')
```

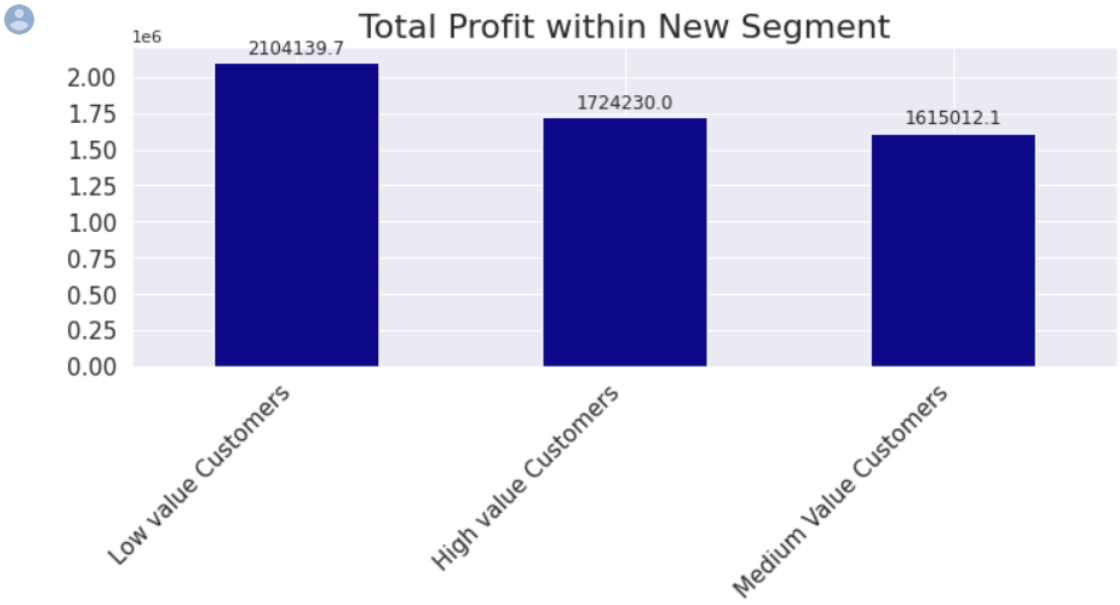


Figure 4-7 Total Profit with new Segmentation

Handling customer age:

```python
def cohort(Age):
    if Age < 30:
        return 'Under 30'
    elif Age <= 40:
        return '30-39'
    elif Age <= 50:
        return '40-49'
    elif Age < 60:
        return '50-59'
    else:
        return "60+"

df_Kmean['Age_group'] = df_Kmean['Age'].apply(cohort)
df_Kmean
```

	CustomerKey	Recency	Frequency	Monetary	Clusters	Segment_new	BirthDate	Gender	MaritalStatus	YearlyIncome	GeographyKey	EnglishOccupation	EnglishCountryRegionName
0	11000	274	3	8248.9900	1	High value Customers	1971-10-06	M	M	90000	26	Professional	Australia
1	11360	41	3	5025.8000	1	High value Customers	1976-05-10	M	S	10000	26	Manual	Australia
2	11368	62	3	5106.2400	1	High value Customers	1971-02-09	M	S	10000	26	Manual	Australia
3	11371	35	2	74.9800	2	Low value Customers	1973-08-14	F	M	20000	26	Clerical	Australia
4	11909	184	3	8105.3100	1	High value Customers	1979-08-05	F	M	80000	26	Management	Australia

```python
#chart thể hiện phân bố nhóm tuổi trong từng phân khúc


plt.figure(figsize=(12,8), tight_layout=True)
ax = sns.countplot(data=df_Kmean, x= 'Segment_new', hue='Age_group', palette='plasma')
for p in ax.patches:
    ax.text(p.get_x() + p.get_width()/2., p.get_height(), '%d' % int(p.get_height()),
            fontsize=12, color='black', ha='center', va='bottom')

plt.xticks(rotation=45)
```

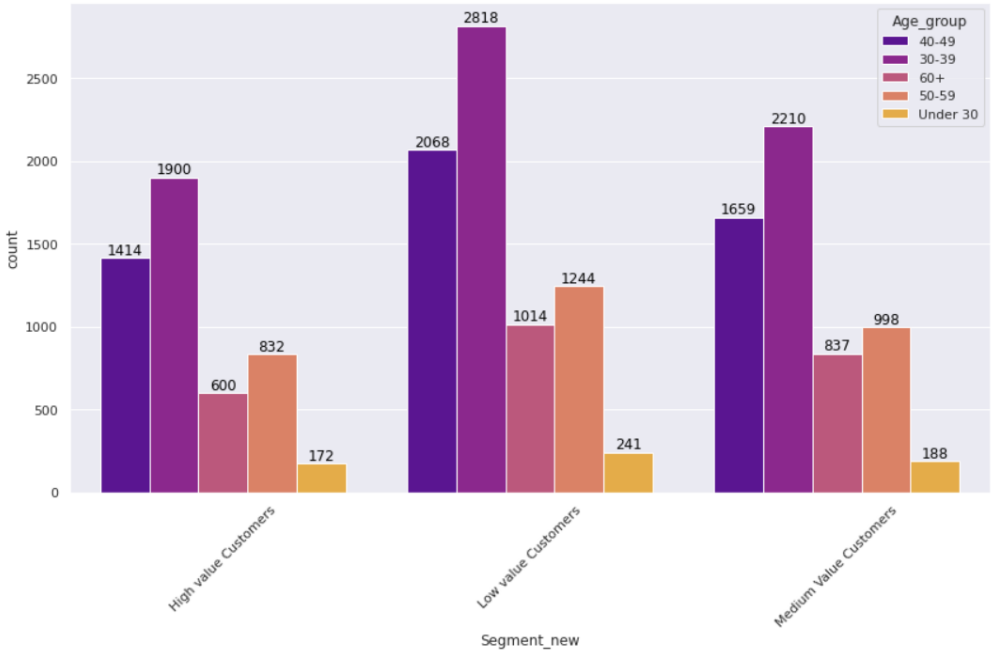


Figure 4-8 Age with new Segmentation

Data visualization with marital status column:

```
#chart thể hiện married/single trong từng phân khúc


plt.figure(figsize=(12,8), tight_layout=True)
ax = sns.countplot(data=df_Kmean, x= 'Segment_new', hue='MaritalStatus', palette='plasma')
for p in ax.patches:
    ax.text(p.get_x() + p.get_width()/2., p.get_height(), '%d' % int(p.get_height()),
            fontsize=12, color='black', ha='center', va='bottom')

plt.xticks(rotation=45)
```

```
(array([0, 1, 2]), <a list of 3 Text major ticklabel objects>)
```

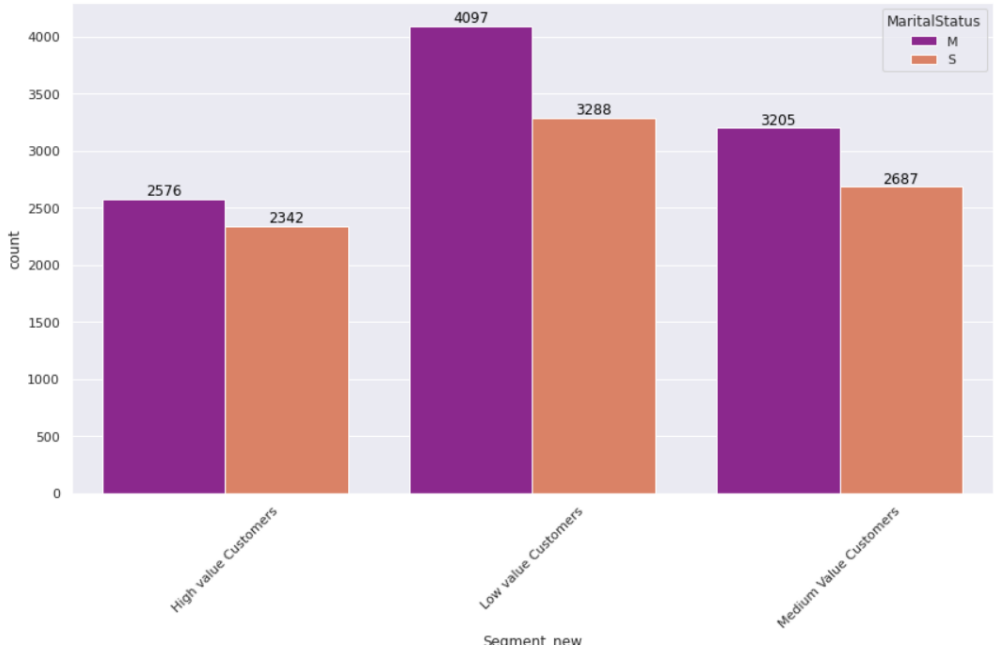


Figure 4-9 Marital Status with new Segmentation

Visualize data with gender columns:

```
[ ]  # đếm số lượng Male và Female trong từng phân khúc
     df_Kmeangender = pd.DataFrame(df_Kmean.groupby(['Segment_new','Gender'])['Gender'].count())
     df_Kmeangender
```

		Gender
Segment_new	**Gender**	
High value Customers	**F**	2470
	M	2448
Low value Customers	**F**	3620
	M	3765
Medium Value Customers	**F**	2902
	M	2990

```
[ ]  #chart thể hiện giới tính trong từng phân khúc
     plt.figure(figsize=(12,8), tight_layout=True)
     ax = sns.countplot(data=df_Kmean, x= 'Segment_new', hue='Gender', palette='plasma')
     for p in ax.patches:
         ax.text(p.get_x() + p.get_width()/2., p.get_height(), '%d' % int(p.get_height()),
                 fontsize=12, color='black', ha='center', va='bottom')
     plt.xticks(rotation=45)
```

```
(array([0, 1, 2]), <a list of 3 Text major ticklabel objects>)
```

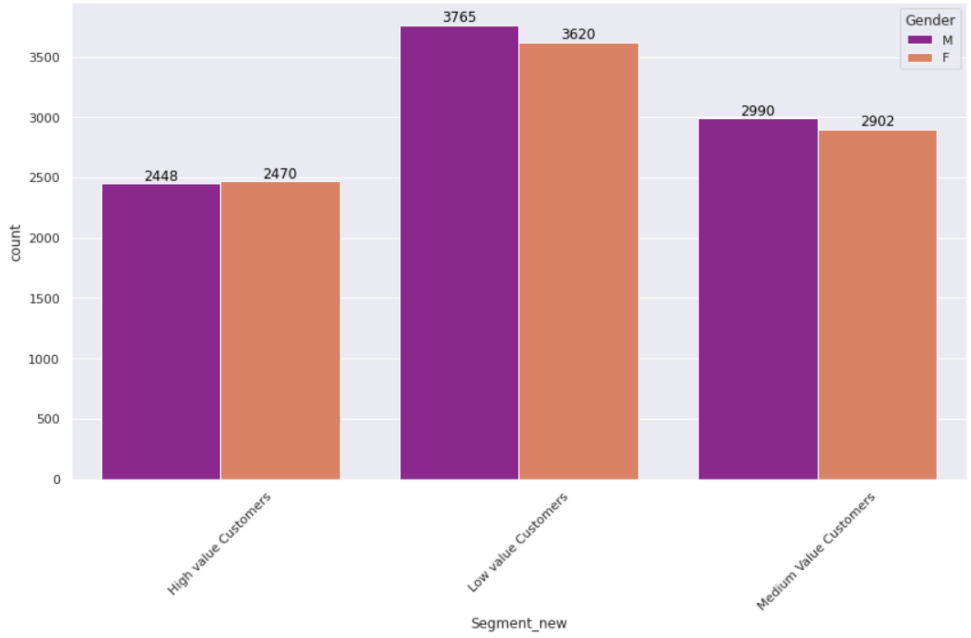


Figure 4-10 Gender with new Segmentation

Visualize data with Occupation:

```
[ ]  #chart thể hiện phân bố nghề nghiệp trong từng phân khúc
     plt.figure(figsize=(12,8), tight_layout=True)
     ax = sns.countplot(data=df_Kmean, x= 'Segment_new', hue='EnglishOccupation', palette='plasma')
     for p in ax.patches:
         ax.text(p.get_x() + p.get_width()/2., p.get_height(), '%d' % int(p.get_height()),
                 fontsize=12, color='black', ha='center', va='bottom')
     plt.xticks(rotation=45)
```

```
(array([0, 1, 2]), <a list of 3 Text major ticklabel objects>)
```

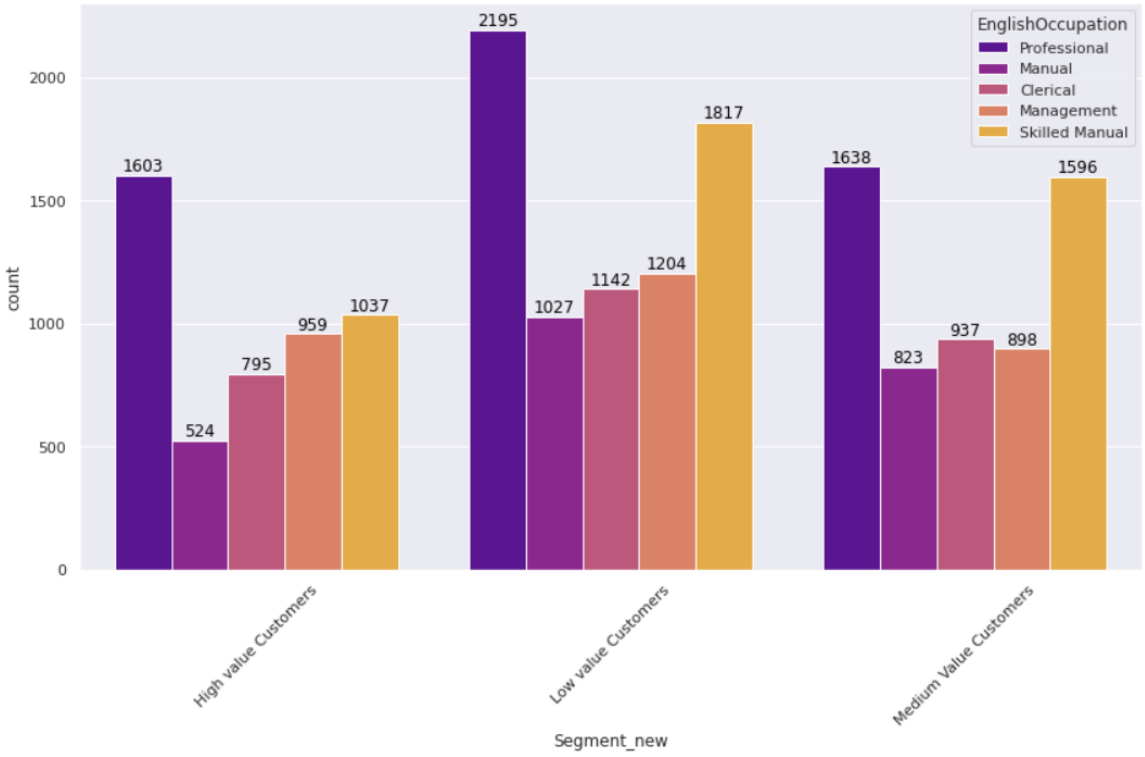


Figure 4-11 English Occupation with new Segmentation

Visualize data with geographic columns:

```
[ ]  #chart thể hiện phân bố nước trong từng phân khúc
     plt.figure(figsize=(12,8), tight_layout=True)
     ax = sns.countplot(data=df_Kmean, x= 'Segment_new', hue='EnglishCountryRegionName', palette='plasma')
     for p in ax.patches:
         ax.text(p.get_x() + p.get_width()/2., p.get_height(), '%d' % int(p.get_height()),
                 fontsize=12, color='black', ha='center', va='bottom')
     plt.xticks(rotation=45)
```

```
(array([0, 1, 2]), <a list of 3 Text major ticklabel objects>)
```

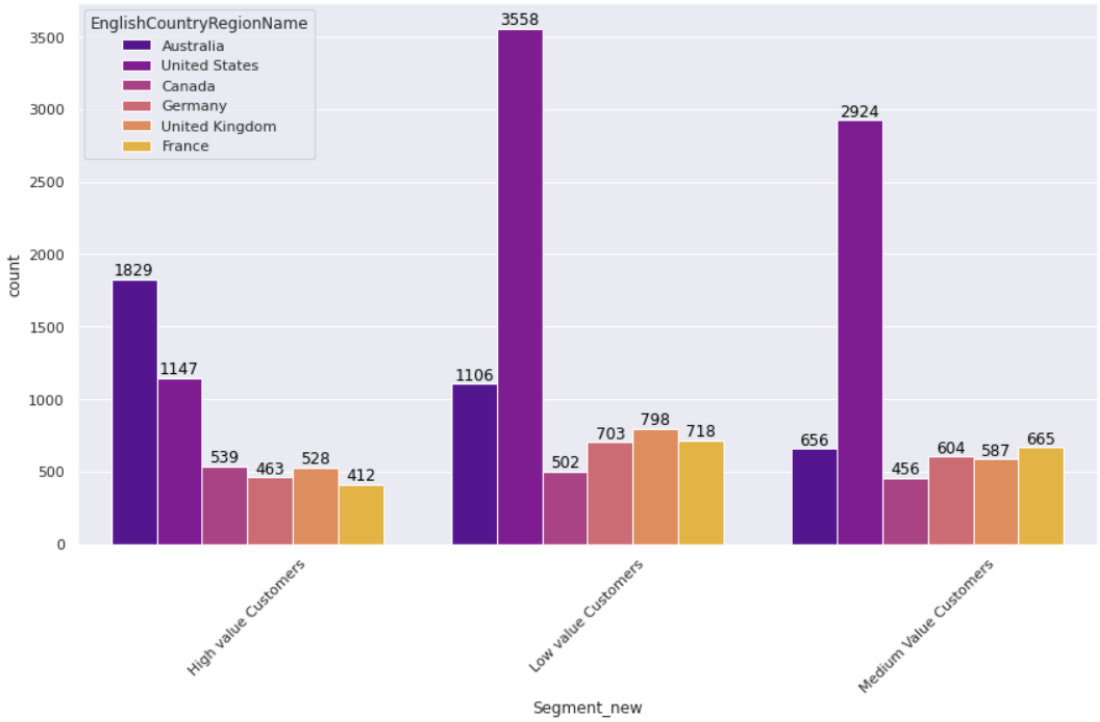


Figure 4-12 RegionName with new Segmentation

Visualize data with customer segmentation and income:

```
[ ]  ax= sns.barplot(x='Segment_new',y='YearlyIncome',palette="plasma",data=df_avgKmean)
     for p in ax.patches:
         ax.text(p.get_x() + p.get_width()/2., p.get_height(), '%d' % int(p.get_height()),
                 fontsize=12, color='black', ha='center', va='bottom')

     plt.xticks(rotation=45)
```

```
(array([0, 1, 2]), <a list of 3 Text major ticklabel objects>)
```

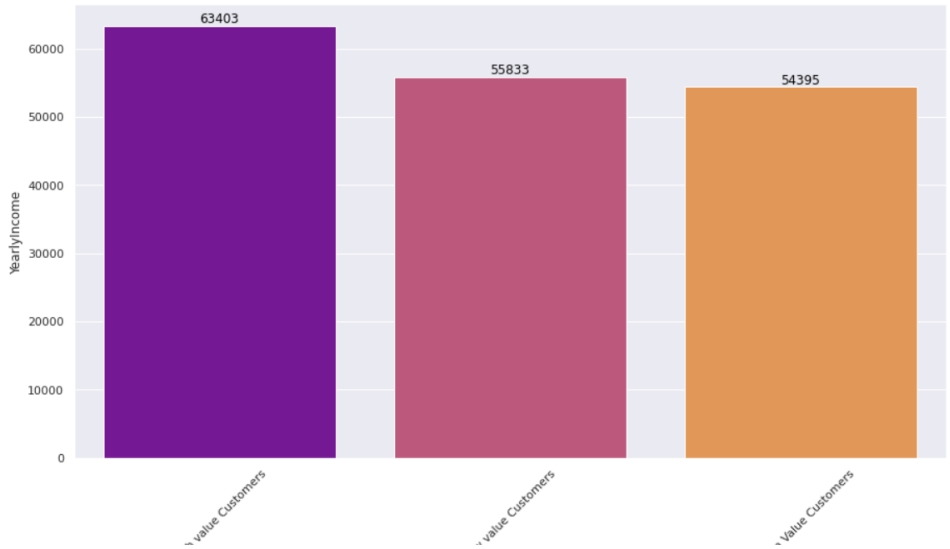


Figure 4-13 YearIncome with new Segmentation

Visualize data with customer segmentation and customer spend columns

```
[ ]  ax= sns.barplot(x='Segment_new',y='Monetary',palette="plasma",data=df_avgKmean)
     for p in ax.patches:
         ax.text(p.get_x() + p.get_width()/2., p.get_height(), '%d' % int(p.get_height()),
                 fontsize=12, color='black', ha='center', va='bottom')

     plt.xticks(rotation=45)
```

```
(array([0, 1, 2]), <a list of 3 Text major ticklabel objects>)
```

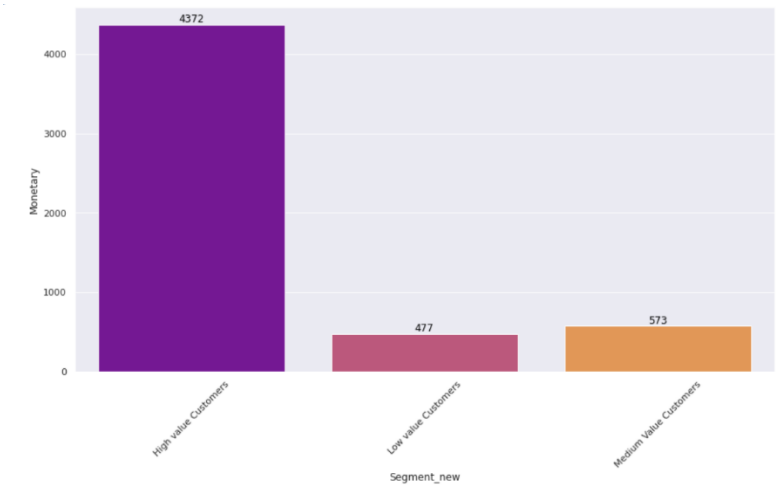


Figure 4-14 Monetary with new Segmentation

From the chart built above, we can show the characteristics of each segment as follows

Segment 1: High Customers *(high spending total, high purchase frequency, intermediate recency)*

This segment focuses on customers with the highest income compared to the other two segments. Moreover, they also spend the highest amount of money on the product of company AWC (Monetary with max is 13295 and min is 34.56).

- This segment's most purchasing age group is **30-39**;
- The average age is **44** years;
- Predominantly **female and married**;
- This segment's clients are mostly **professional**.
- In this segment, the number of customers from **Australia** is the highest.
- Average Annual Income is **$63403** in dollars;
- Average Spending Total is **$4372**

Segment 2: Medium Value Customers *(intermediate spending total, intermediate purchase frequency, high recency)*

This segment focuses on customers with moderate income. Moreover, they also spend money on AWC's products in the medium range (Monetary with max 3578 and min 2.29) but with the highest frequency (max: 864 and min 195).

Proposal: promote customers with promotions and offers on special occasions

- This segment's most purchasing age group is **30-39**;
- The average age is **45** years;

- Predominantly **Male and married**;
- This segment's clients are mostly **professional and skilled manual**.
- In this segment, the number of customers from **United States** is the highest.
- Average Annual Income is **$54395** in dollars;
- Average Spending Total is **$573**.

Segment 3: Low Value Customers *(low spending total, intermediate purchase frequency, low recency)*

This segment has the lowest frequency and recency ratio with AWC products, so the amount of money spent is also small. Therefore, the company needs to have promotions to reach more of these customers

- This segment's most purchasing age group is **30-39**;
- The average age is **45** years;
- Predominantly **Male and married**;
- This segment's clients are mostly **professional**.
- In this segment, the number of customers from **the United States** is the highest.
- Average Annual Income is **$55833** in dollars;
- Average Spending Total is **$477**.

4.3. Analysis of customer churn Churn rate

To calculate customer churn, we calculate a 6-month cycle, since AWC's core business is bicycles, accessories and sportswear, the average repurchase cycle is often longer than essential items.

```
[ ] #Churn cua 012012 -> Khach hang mua hang lan cuoi truoc 01062011
    df['01/06/2011']=df_FSales.groupby("CustomerKey").agg({"OrderDate":"max"}).rename(columns = {"OrderDate":"LastOrderDate"}) <dt.datetime(2011,6,1)
    np.count_nonzero(df['01/06/2011'])
```

```
11000
```

```
[ ] #Khach hang mua lan dau truoc 01012012
    df['01/06/2011']=df_FSales.groupby("CustomerKey").agg({"OrderDate":"min"}).rename(columns = {"OrderDate":"LastOrderDate"}) <dt.datetime(2012,1,1)
    np.count_nonzero(df['01/06/2011'])
```

```
12203
```

```
[ ] #Khach hang mua lan dau tu 012012 den 062012
    np.count_nonzero((df_FSales.groupby("CustomerKey").agg({"OrderDate":"min"}).rename(columns = {"OrderDate":"LastOrderDate"}) >=dt.datetime(2012,1,1))
    & (df_FSales.groupby("CustomerKey").agg({"OrderDate":"min"}).rename(columns = {"OrderDate":"LastOrderDate"}) <dt.datetime(2012,6,1)))
```

```
1150
```

CHURN RATES thang 01 nam 2012 = 128/230 = 5,7%

```
[ ] #Churn cua 06012012 -> Khach hang mua hang lan cuoi truoc tu 062011 012012
    np.count_nonzero((df_FSales.groupby("CustomerKey").agg({"OrderDate":"max"}).rename(columns = {"OrderDate":"LastOrderDate"}) >=dt.datetime(2011,6,1))
    & (df_FSales.groupby("CustomerKey").agg({"OrderDate":"max"}).rename(columns = {"OrderDate":"LastOrderDate"}) <dt.datetime(2012,1,1)))
```

```
134
```

```
[ ] #Khach hang mua lan dau tu 062012 den 012013
    np.count_nonzero((df_FSales.groupby("CustomerKey").agg({"OrderDate":"min"}).rename(columns = {"OrderDate":"LastOrderDate"}) >=dt.datetime(2012,6,1))
    & (df_FSales.groupby("CustomerKey").agg({"OrderDate":"min"}).rename(columns = {"OrderDate":"LastOrderDate"}) <dt.datetime(2013,1,1)))
```

```
2075
```

CHURN RATES thang 06 nam 2012 = 134/3252 = 4,1%

```
[ ] #Churn cua 0102013 -> Khach hang mua hang lan cuoi truoc tu 012012 062012
    np.count_nonzero((df_FSales.groupby("CustomerKey").agg({"OrderDate":"max"}).rename(columns = {"OrderDate":"LastOrderDate"}) >=dt.datetime(2012,1,1))
    & (df_FSales.groupby("CustomerKey").agg({"OrderDate":"max"}).rename(columns = {"OrderDate":"LastOrderDate"}) <dt.datetime(2012,6,1)))
```

```
83
```

```
[ ] #Khach hang mua lan dau tu 01/2013 den 06/2013
    np.count_nonzero((df_FSales.groupby("CustomerKey").agg({"OrderDate":"min"}).rename(columns = {"OrderDate":"LastOrderDate"}) >=dt.datetime(2013,1,1))
    & (df_FSales.groupby("CustomerKey").agg({"OrderDate":"min"}).rename(columns = {"OrderDate":"LastOrderDate"}) <dt.datetime(2013,6,1)))
```

```
4805
```

CHURN RATES thang 01 nam 2013 = 83/5193 = 1,6%

```
[ ] #Churn cua 0602013 -> Khach hang mua hang lan cuoi truoc tu 06 2012 01/2013
    np.count_nonzero((df_FSales.groupby("CustomerKey").agg({"OrderDate":"max"}).rename(columns = {"OrderDate":"LastOrderDate"}) >=dt.datetime(2012,6,1))
    & (df_FSales.groupby("CustomerKey").agg({"OrderDate":"max"}).rename(columns = {"OrderDate":"LastOrderDate"}) <dt.datetime(2013,1,1)))

    204
```

```
[ ] #Khach hang mua lan dau tu 06/2013 den 01/2014
    np.count_nonzero((df_FSales.groupby("CustomerKey").agg({"OrderDate":"min"}).rename(columns = {"OrderDate":"LastOrderDate"}) >=dt.datetime(2013,6,1))
    & (df_FSales.groupby("CustomerKey").agg({"OrderDate":"min"}).rename(columns = {"OrderDate":"LastOrderDate"}) <dt.datetime(2014,1,1)))

    7718
```

CHURN RATES thang 06 nam 2013 = 204/9915 = 2,1%

```
[ ] #Churn cua 01/2014 -> Khach hang mua hang lan cuoi truoc tu 01/2013 06/2013
    np.count_nonzero((df_FSales.groupby("CustomerKey").agg({"OrderDate":"max"}).rename(columns = {"OrderDate":"LastOrderDate"}) >=dt.datetime(2013,1,1))
    & (df_FSales.groupby("CustomerKey").agg({"OrderDate":"max"}).rename(columns = {"OrderDate":"LastOrderDate"}) <dt.datetime(2013,6,1)))

    5010
```

CHURN RATES thang 01 nam 2014 thì chưa đủ số ngày để tính được new customer từ tháng 01 -> tháng 06 năm 2014

Calculate the number of new customers who make their first purchase in every 6 months: called new customers

Calculate the number of customers who do not buy in a row in 6 months: called the rate of old customers churning.

The number of customers at the end of period T will be the number of customers at the beginning of period T + 1.

Applying the formula to calculate Customer churn rate will be graphed as shown

Visualization with charts

Then use a line chart to show the rate of change of churn rate

```
import matplotlib.pyplot as plt

plt.plot(years,churnrates)
plt.title('Churn rates by year')
plt.xlabel('years')
plt.ylabel('Churn rates')
plt.show()
```

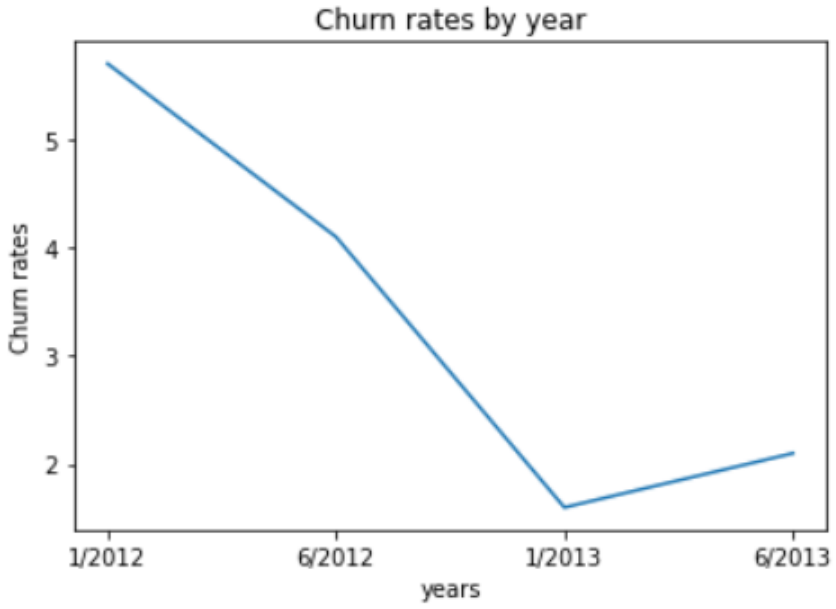


Figure 4-15 Churn rate by years

The customer churn rate tended to decrease, but by the beginning of 2013, it started to increase again. This is not good for business. Although the number of days in 2014 is not enough, it is expected that this rate will continue to increase sharply when the number of customers leaving suddenly increases (up 10 times in the previous cycles).

With a dropout rate of less than 6%, this is a low rate. For a bicycle company, this is a good ratio because bicycles are a price-sensitive commodity if there are small changes in impact. Pricing information from the company or its competitors will lead to a large variation in the customer's return rate, which indicates that the

company is competing well in the clothing sector. bicycles and gain market share. The company has well applied business and customer segmentation strategies to lead the bicycle market year by year. In the future, the company should adopt good customer segmentation models and promote effectively. It is more effective to increase customer return rate as well as reduce customer churn rate in the best way for the company.

Here is a summary table of the data for your convenience

	Jan-12	Jun-12	Jan-13	Jun-13	Jan-14
Đầu tháng	2230	3252	5193	9915	17429
Old churns	128	134	83	204	5010
New	1150	2075	4805	7718	Chưa đủ số ngày để tính
New churns					
Tổng cuối tháng	3252	5193	9915	17429	
Tỷ lệ	5.7%	4.1%	1.6%	2.1%	

4.4. Analysis of customer retention- Cohort

```
[ ]  df_cohort = df_FSales.copy()
     df_cohort.head()
```

	ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	SalesTerritoryKey	SalesOrderNumber	SalesOrderLineNumber	...	UnitPriceDiscountPct
0	310	20101229	20110110	20110105	21768	1	19	6	SO43697	1	...	0
1	346	20101229	20110110	20110105	28389	1	39	7	SO43698	1	...	0
2	346	20101229	20110110	20110105	25863	1	100	1	SO43699	1	...	0
3	336	20101229	20110110	20110105	14501	1	100	4	SO43700	1	...	0
4	346	20101229	20110110	20110105	11003	1	6	9	SO43701	1	...	0

5 rows × 24 columns

```
[ ]  #lấy chính xác giá trị tháng
     df_cohort['OrderMonth'] = df_cohort['OrderDate'].dt.strftime('%Y-%m')
     # chuyển đổi biến thành định dạng ngày giờ
     df_cohort['OrderMonth'] = pd.to_datetime(df_cohort['OrderMonth'])
```

Create OrderMonth column and assign datetime value to that data column

OrderMonth is a string representation of the year and month of a single transaction

```
[ ]  # tạo biến thứ hai 'Tháng theo nhóm'
     # nhận được ngày mua hàng đầu tiên cho mỗi khách hàng
     df_cohort['CohortMonth'] = df_cohort.groupby('CustomerKey')['OrderMonth'].transform('min')
     # chuyển đổi biến thành định dạng ngày giờ
     df_cohort['CohortMonth'] = pd.to_datetime(df_cohort['CohortMonth'])
```

Create a CohortMonth column according to the customer's purchase time and assign a date and time value to that data column, A string representation of the the year and month of a customer’s first purchase.

```
[ ]  df_cohort.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60398 entries, 0 to 60397
Data columns (total 26 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ProductKey            60398 non-null  int64
 1   OrderDateKey          60398 non-null  int64
 2   DueDateKey            60398 non-null  int64
 3   ShipDateKey           60398 non-null  int64
 4   CustomerKey           60398 non-null  int64
 5   PromotionKey          60398 non-null  int64
 6   CurrencyKey           60398 non-null  int64
 7   SalesTerritoryKey     60398 non-null  int64
 8   SalesOrderNumber      60398 non-null  object
 9   SalesOrderLineNumber  60398 non-null  int64
 10  RevisionNumber        60398 non-null  int64
 11  OrderQuantity         60398 non-null  int64
 12  UnitPrice             60398 non-null  float64
 13  ExtendedAmount        60398 non-null  float64
 14  UnitPriceDiscountPct  60398 non-null  int64
 15  DiscountAmount        60398 non-null  int64
 16  ProductStandardCost   60398 non-null  float64
 17  TotalProductCost      60398 non-null  float64
 18  SalesAmount           60398 non-null  float64
 19  TaxAmt                60398 non-null  float64
 20  Freight               60398 non-null  float64
 21  OrderDate             60398 non-null  datetime64[ns]
 22  DueDate               60398 non-null  datetime64[ns]
 23  ShipDate              60398 non-null  datetime64[ns]
 24  OrderMonth            60398 non-null  datetime64[ns]
 25  CohortMonth           60398 non-null  datetime64[ns]
dtypes: datetime64[ns](5), float64(7), int64(13), object(1)
```

check the data has been converted to datetime data type

```
[ ]  df_cohort
```

ritoryKey	SalesOrderNumber	SalesOrderLineNumber	...	ProductStandardCost	TotalProductCost	SalesAmount	TaxAmt	Freight	OrderDate	DueDate	ShipDate	OrderMonth	CohortMonth
6	SO43697	1	...	2171.2942	2171.2942	3578.2700	286.2616	89.4568	2010-12-29	2011-01-10	2011-01-05	2010-12-01	2010-12-01
7	SO43698	1	...	1912.1544	1912.1544	3399.9900	271.9992	84.9998	2010-12-29	2011-01-10	2011-01-05	2010-12-01	2010-12-01
1	SO43699	1	...	1912.1544	1912.1544	3399.9900	271.9992	84.9998	2010-12-29	2011-01-10	2011-01-05	2010-12-01	2010-12-01
4	SO43700	1	...	413.1463	413.1463	699.0982	55.9279	17.4775	2010-12-29	2011-01-10	2011-01-05	2010-12-01	2010-12-01
9	SO43701	1	...	1912.1544	1912.1544	3399.9900	271.9992	84.9998	2010-12-29	2011-01-10	2011-01-05	2010-12-01	2010-12-01
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6	SO75122	1	...	8.2205	8.2205	21.9800	1.7584	0.5495	2014-01-28	2014-02-09	2014-02-04	2014-01-01	2013-05-01

```
[ ]  def diff_month(d1, d2):
         return((d1.dt.year - d2.dt.year) * 12 + d1.dt.month - d2.dt.month)

     df_cohort['CohortPeriod'] = diff_month(df_cohort['OrderMonth'], df_cohort['CohortMonth'])
```

An integer represents a customer's stage in its "lifetime". The number represents the number of months passed since the first purchase.

```
[ ]  customer_cohort = df_cohort.pivot_table(index='CohortMonth', columns='CohortPeriod', values='CustomerKey', aggfunc='nunique')
     customer_cohort
```

CohortPeriod	0	1	2	3	4	5	6	7	8	9	...	27	28	29	30	31	32	33	34	35	36
CohortMonth																					
2010-12-01	14.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	3.0	1.0	2.0	NaN	2.0	NaN	NaN	NaN	3.0	2.0
2011-01-01	144.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	10.0	21.0	24.0	7.0	8.0	5.0	12.0	25.0	27.0	NaN
2011-02-01	144.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	8.0	12.0	12.0	19.0	7.0	4.0	29.0	28.0	NaN	NaN
2011-03-01	150.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	18.0	24.0	16.0	18.0	13.0	6.0	25.0	NaN	NaN	NaN
2011-04-01	157.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	24.0	19.0	13.0	19.0	3.0	27.0	NaN	NaN	NaN	NaN
2011-05-01	174.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	10.0	15.0	21.0	11.0	4.0	NaN	NaN	NaN	NaN	NaN
2011-06-01	230.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	10.0	23.0	17.0	22.0	NaN	NaN	NaN	NaN	NaN	NaN
2011-07-01	188.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	17.0	18.0	10.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-08-01	193.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	9.0	26.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-09-01	185.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	9.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-10-01	221.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-11-01	208.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-12-01	222.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2012-01-01	252.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2012-02-01	260.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
[ ]  cohort_size = customer_cohort.iloc[:,0]
     retention = customer_cohort.divide(cohort_size,axis=0)
     retention.index = pd.to_datetime(retention.index).date
     retention.round(3) * 100
```

CohortPeriod	0	1	2	3	4	5	6	7	8	9	...	27	28	29	30	31	32	33	34	35	36
2010-12-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	21.4	7.1	14.3	NaN	14.3	NaN	NaN	NaN	21.4	14.3
2011-01-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	6.9	14.6	16.7	4.9	5.6	3.5	8.3	17.4	18.8	NaN
2011-02-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	5.6	8.3	8.3	13.2	4.9	2.8	20.1	19.4	NaN	NaN
2011-03-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	12.0	16.0	10.7	12.0	8.7	4.0	16.7	NaN	NaN	NaN
2011-04-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	15.3	12.1	8.3	12.1	1.9	17.2	NaN	NaN	NaN	NaN
2011-05-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	5.7	8.6	12.1	6.3	2.3	NaN	NaN	NaN	NaN	NaN
2011-06-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	4.3	10.0	7.4	9.6	NaN	NaN	NaN	NaN	NaN	NaN
2011-07-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	9.0	9.6	5.3	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-08-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	4.7	13.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-09-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	4.9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-10-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-11-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-12-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2012-01-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2012-02-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
[ ]  plt.figure(figsize=(25, 15))
     plt.title('Retention Rates(in %) over one year period', size=25)
     sns.heatmap(data=retention, annot = True, fmt = '.0%', cmap="summer_r")
     plt.show()
```

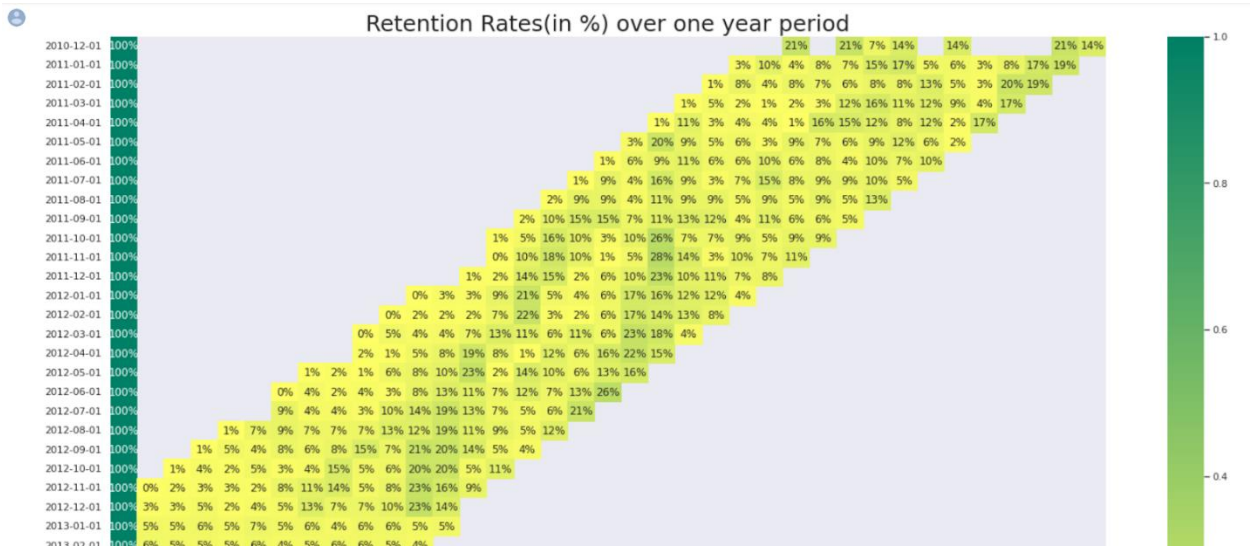


Figure 4-16 Retention rates

Customer retention is a very useful metric to understand how many of the all customers are still active.Retention gives you the percentage of active customers compared to the total number of customers.

Retention rates are often ignored, but they are actually very important. Because the cost of customer acquisition is very expensive we need to do everything to convince the client to return after their first purchase.

If your retention rate is low you will spend a budget for the acquisition channel so that more customers will arrive.

From Cohort Analysis we can see the retention rate or what percentage of customers return in the following months after the first purchase

CHAPTER 5 SUMMARY AND EVALUATION

5.1. Summary of the project implementation

Using RFM, divide your customers into five groups.In each segment, we will analyze demographics, including age, gender, marriage, income, and find out where you live. In terms of behavior, find out how much customers buy by segment. Then, using the chart to visualize the data, come up with a suggested strategy to increase the company's profits.

Use K-means to divide 3 consecutive segments and analyze each segment as above.

Predict the customer churn rate and customer retention rate.

5.2. Future work

Continue to further analyze behavior, psychology, .. to understand more deeply about customers. From there, give suitable strategies for each different segment.

From the customer segment, it is possible to build a clearer customer portrait to concretize the customer image.

Calculating more indicators in marketing combined with production activities to assess business efficiency, improve revenue and profit for the company.

5.3. Conclusion

Thanks to this research and analysis, the company's Board of Directors can propose a suitable development direction for the whole company and for each customer segment to increase customer retention rate and reduce customer churn rate. With these indicators, they relatively reflect the overall business situation.

Customer segmentation is the most important factor if you want to promote customer loyalty and retain customers. Leveraging customer segments is at the core of helping a company:

Improve product or service quality to better meet customer needs.

Offer a new or complementary product that appeals to potential consumers.

Outperform the competition.

By defining your product's strategic goals and setting up the right metrics to measure progress toward those goals, you'll gain vital product data. However, analyzing that data into insightful and strategic insights is the most important step. Equipping logical and scientific data analysis thinking, as well as gaining a deeper understanding of the data groups that solve each specific problem of the business is necessary to complete it.

REFERENCES

[1] Aditya, "Moengage," 21 09 2021. [Online]. Available: https://www.moengage.com/blog/rfm-analysis-using-rfm-segments/#NaN. [Accessed 01 06 2022].

[2] "Saleforce," [Online]. Available: https://www.salesforce.com/resources/articles/how-calculate-customer-churn-and-revenue-churn/#:~:text=Churn%20rate%2C%20sometimes%20known%20as,stop%20buying%20from%20your%20business. [Accessed 03 06 2022].

[3] "Datafun," Hieu Hoang, [Online]. Available: https://data-fun.com/cach-tinh-ty-le-churn-rate-la-gi/. [Accessed 06 2022].

[4] "ThinkZone," 06 10 2021. [Online]. Available: http://thinkzone.vn/blog/churn-rate-la-gi-cach-tinh-2-loai-churn-rate-quan-trong-trong-doanh-nghiep. [Accessed 06 2022].

[5] A. Caldwell, "Oracle netsuite," 27 01 2021. [Online]. Available: https://www.netsuite.com/portal/resource/articles/human-resources/customer-churn-analysis.shtml?mc24943=v2. [Accessed 06 2022].

[6] "Putler," [Online]. Available: https://www.putler.com/customer-retention/. [Accessed 06 2022].

[7] M. S. Iqbal, 23 05 2020. [Online]. Available: https://medium.com/@samraniqbal1991/cohort-analysis-d7f611a686a1. [Accessed 06 2022].

[8] P. Makhija, "CleverTap," [Online]. Available: https://clevertap.com/blog/cohort-analysis/. [Accessed 06 2022].

[9] F. Neves, "Towards Data Science," 21 11 2019. [Online]. Available: https://towardsdatascience.com/how-to-calculate-customer-retention-rate-a-practical-approach-1c97709d495f#:~:text=This%20percentage%20is%20nothing%20less,%25%20(4%2F10). [Accessed 06 2022].

[10] "Java T point," [Online]. Available: https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning. [Accessed 06 2022].

[11] B. Saji, "Analytics Vidhya," 21 01 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/. [Accessed 06 2022].

[12] K. Arvai, "Real Python," [Online]. Available: https://realpython.com/k-means-clustering-python/. [Accessed 06 2022].

[13] A. Christy, "ScienceDirect," [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157818304178. [Accessed 06 2022].

[14] "GeeksforGeeks," 11 2021. [Online]. Available: https://www.geeksforgeeks.org/rfm-analysis-analysis-using-python/. [Accessed 06 2022].

[15] "Nerd For Tech," 4 07 2021. [Online]. Available: https://medium.com/nerd-for-tech/customer-segmentation-using-python-e56c2b1a4c73. [Accessed 06 2022].

[16] M. GL, "Analytics Vidhya," 27 04 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/04/customer-lifetime-value-using-rfm-analysis/. [Accessed 06 2022].

[17] A. Kamath, "Moengage," 02 09 2021. [Online]. Available: https://www.moengage.com/blog/growth-tactic-1-how-to-use-cohort-analysis-to-measure-customer-retention/. [Accessed 06 2022].

[18] "TRESL," 03 06 2022. [Online]. Available: https://www.tresl.co/blog/rfm-analysis. [Accessed 06 2022].

[19] P. Makhija. [Online]. Available: https://clevertap.com/blog/rfm-analysis/. [Accessed 06 2022].

[20] Tim Ehrens, Site Editor, "TechTarget," [Online]. Available: https://www.techtarget.com/searchcustomerexperience/definition/customer-segmentation. [Accessed 06 2022].

[21] "Optimove," [Online]. Available: https://www.optimove.com/resources/learning-center/customer-segmentation. [Accessed 06 2022].

[22] "Kaggle," 04 2022. [Online]. Available: https://www.kaggle.com/code/liudmylalozhevych/kpmg-data-analytics-consulting-virtual-internship/notebook. [Accessed 06 2022].

[23] N. Shajahan, "Nerd For Tech," 04 07 2021. [Online]. Available: https://medium.com/nerd-for-tech/customer-segmentation-using-python-e56c2b1a4c73. [Accessed 06 2022].