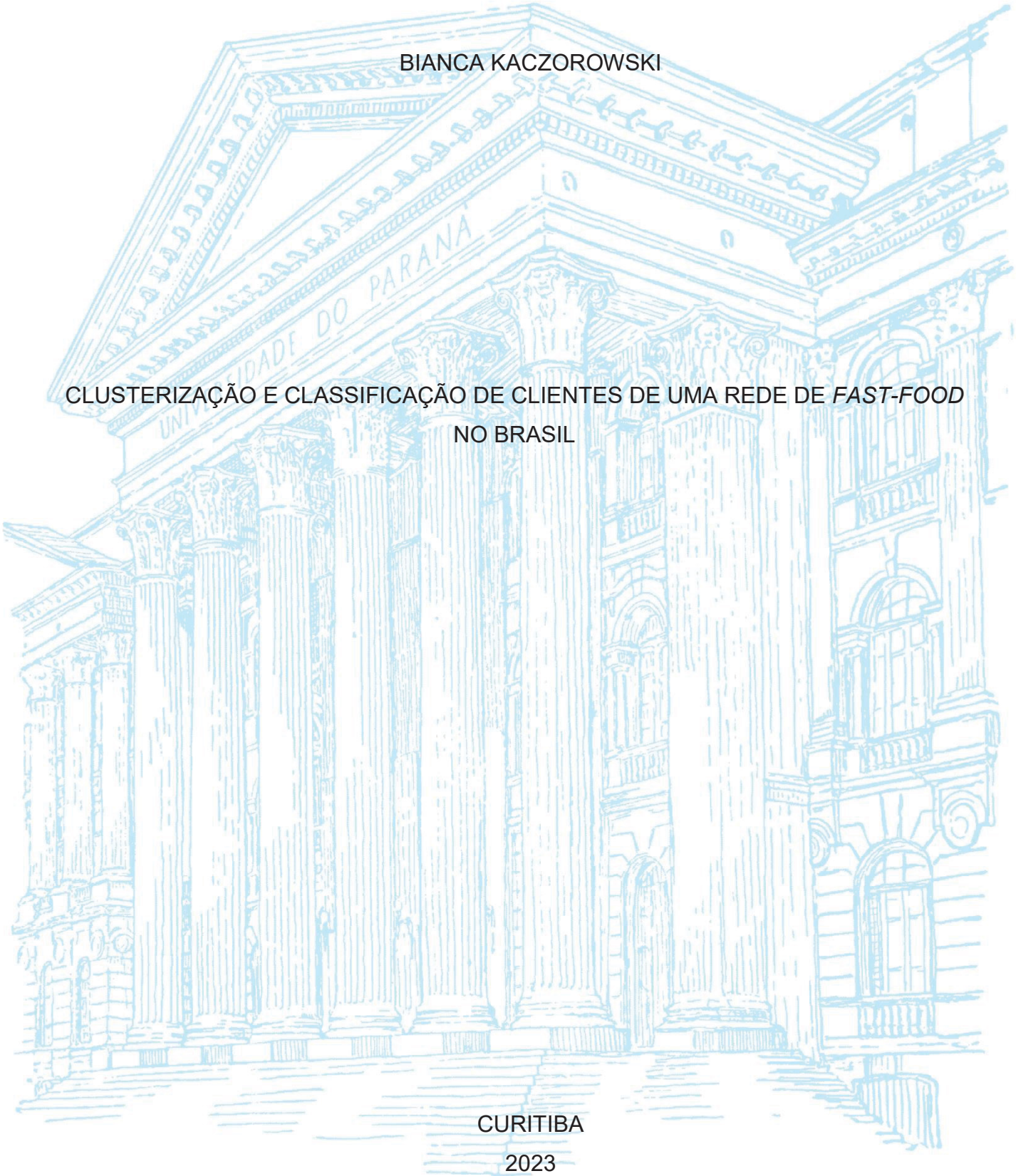
UNIVERSIDADE FEDERAL DO PARANÁ

BIANCA KACZOROWSKI



CLUSTERIZAÇÃO E CLASSIFICAÇÃO DE CLIENTES DE UMA REDE DE *FAST-FOOD* NO BRASIL

CURITIBA

2023

BIANCA KACZOROWSKI

CLUSTERIZAÇÃO E CLASSIFICAÇÃO DE CLIENTES DE UMA REDE DE *FAST-FOOD* NO BRASIL

Dissertação apresentada ao curso de Pós-Graduação em Engenharia de Produção, Setor de Tecnologia, Universidade Federal do Paraná, como requisito à obtenção do título de Mestre em Engenharia de Produção.

Orientadora: Profa. Dra. Mariana Kleina.

CURITIBA

2023

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Kaczorowski, Bianca
Clusterização e classificação de clientes de uma rede de fast-food no Brasil / Bianca Kaczorowski. – Curitiba, 2023.
1 recurso on-line : PDF.

Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia de Produção.

Orientador: Mariana Kleina

1. Restaurantes de refeições ligeiras. 2.Cluster (Sistema de computador). 3. Marketing.. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Engenharia de Produção. III. Kleina, Mariana. IV . Título.

Bibliotecário: Leticia Priscila Azevedo de Sousa CRB-9/2029

MINISTÉRIO DA EDUCAÇÃO
SETOR DE TECNOLOGIA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO ENGENHARIA DE PRODUÇÃO - 40001016070P1

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação ENGENHARIA DE PRODUÇÃO da Universidade Federal do Paraná foram convocados para realizar a arguição da dissertação de Mestrado de **BIANCA KACZOROWSKI** intitulada: **CLUSTERIZAÇÃO E CLASSIFICAÇÃO DE CLIENTES DE UMA REDE DE FAST-FOOD NO BRASIL**, sob orientação da Profa. Dra. MARIANA KLEINA, que após terem inquirido a aluna e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestra está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 31 de Março de 2023.

Assinatura Eletrônica
11/04/2023 14:57:43.0
MARIANA KLEINA
Presidente da Banca Examinadora

Assinatura Eletrônica
11/04/2023 16:06:34.0
WAGNER HUGO BONAT
Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
12/04/2023 09:12:26.0
SOLANGE REGINA DOS SANTOS
Avaliador Externo (UNIVERSIDADE ESTADUAL DO PARANÁ)

Assinatura Eletrônica
12/04/2023 15:56:15.0
MARCOS AUGUSTO MENDES MARQUES
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

A todos que apoiaram e contribuíram para realização desse trabalho.

AGRADECIMENTOS

Primeiramente, expresso minha gratidão a Deus pela saúde e por todas as bênçãos concedidas.

Agradeço ao meu pai por todo o incentivo e apoio durante os momentos mais difíceis.

Gostaria de agradecer ao Anderson pela paciência, ajuda e motivação durante todo o processo.

Também gostaria de agradecer aos professores do programa e de fora dele por todas as contribuições e sugestões valiosas realizadas ao longo das disciplinas, principalmente de Métodos de Pesquisa, mesmo em período de pandemia e também na banca de qualificação.

Por fim, gostaria de agradecer especialmente à Professora Mariana Kleina pela paciência e orientação dedicadas, mesmo à distância, e por estar sempre pronta e disposta a ajudar no que fosse necessário.

“*By failing to prepare you are preparing to fail*”

(BENJAMIN FRANKLIN, 1970)

RESUMO

Com o avanço da transformação digital nas empresas, surge o conceito de *customer centric*, no qual o cliente é colocado no centro das decisões. Nesse contexto, as empresas buscam conhecer seus clientes de forma individualizada para atender às suas necessidades e aprimorar a experiência de compra por meio da personalização. Entender o perfil do consumidor é uma vantagem competitiva, especialmente para empresas em setores em que os canais digitais não são tão representativos, como no caso dos *Quick Service Restaurants*, também conhecidos como *fast-food*. Assim, este estudo tem como objetivo a clusterização e classificação de clientes de uma rede de *fast-food* no Brasil, a fim de permitir a personalização das comunicações de *marketing*. Para isso, foram utilizados três algoritmos de clusterização (K-*Means*, Hierárquico de Ward e Modelo de Misturas Gaussianas), com o suporte do método do cotovelo para a definição do número de *clusters*, e dois algoritmos de classificação (Árvore de Decisão e Floresta Aleatória), tendo em vista o grande volume de dados em análise. O estudo usou seis variáveis, sendo três relacionadas ao modelo de Recência, Frequência e Valor e outras três relacionadas ao consumo dos produtos da empresa. De acordo com os resultados dos diferentes métodos aplicados para avaliação do desempenho dos agrupamentos formados, o K-*Means* foi o algoritmo de clusterização com melhor desempenho, além de ser o mais rápido, e o modelo de Floresta Aleatória foi selecionado para classificar os demais clientes, com uma acurácia de quase 98%.

Palavras-chave: Modelo RFV. Clusterização. Classificação. Personalização.

ABSTRACT

As companies undergo digital transformation, the concept of customer centricity emerges, placing the customer at the center of business decisions. In order to better understand and cater to individual customer needs, companies are seeking to personalize their marketing communications and improve the overall purchasing experience. In this context, this study aims to cluster and classify customers of a fast-food chain in Brazil. Three clustering algorithms (K-Means, Ward's Hierarchical, and Gaussian mixtures model) were employed along with the elbow method to determine the appropriate number of clusters, as well as two classification algorithms (Decision tree and Random forest), given the large volume of data being analyzed. The study utilized six variables, three of which were based on the Recency, Frequency, and Value model (RFM), and three were related to the customers' consumption of the company's products. According to the results of the different methods applied for evaluating the performance of the formed clusters, K-Means was the best performing and fastest clustering algorithm. Additionally, the Random forest model was chosen for customer classification, achieving an accuracy of almost 98%.

Keywords: RFM Model. Clustering. Classification. Personalization.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE SIGLAS

CRM	*Customer Relationship Management*
RFV	Recência, frequência e valor
GMM	Modelo de Misturas Gaussianas
OOB	*Out of Bag*

SUMÁRIO

1 INTRODUÇÃO

Ao longo dos últimos anos o termo transformação digital vem tomando proporções exponenciais e, com os efeitos da pandemia mundial, em 2020, provocada pelo vírus SARS-CoV-2, também conhecido como coronavírus ou COVID-19, esse processo ganhou mais importância e cadência nas empresas.

A transformação digital pode ser entendida, segundo Kane (2017), como a adoção de processos e práticas de negócio que ajudam a organização a competir de maneira eficaz no mundo cada vez mais digital. Em outras palavras, é a forma pela qual a empresa é capaz de responder às tendências digitais, o que envolve a adaptação no uso de tecnologias por parte dos clientes, parceiros, colaboradores e competidores, induzindo uma consequente mudança organizacional.

Dentre as principais alavancas, tem-se: (i) A criação de novos modelos de negócios; (ii) Conectividade, ou seja, ter todas as informações disponíveis em tempo real; (iii) Processos, buscando trazer o cliente em primeiro lugar na tomada de decisão de forma a proporcionar uma melhor experiência e com muita agilidade e (iv) *Analytics*, ou seja, fazer o uso dos dados, estudá-los e interpretá-los da melhor maneira para que isso se torne insumo para tomada de decisão, tornando-a mais assertiva (MARTINS *et al.*, 2019).

Nesse contexto surge o conceito "*customer centric"* que, em suma, significa que a empresa está orientada ao cliente, em outras palavras, pode ser definido como a alocação dos interesses dos clientes no centro das decisões da companhia. As empresas passam então a focar esforços em identificar e atender as necessidades dos clientes. Sheth *et al.* (2000) complementam que o processo de compreensão dos consumidores deve ser feito de maneira individual e não em massa como se todos tivessem o mesmo comportamento de compras.

Portanto, no que tange a experiência do usuário, que vem se tornando cada vez mais um meio de diferenciação entre as empresas no mercado, há uma crescente demanda por personalização. O uso de novas tecnologias como *cloud computing, internet* das coisas, mas principalmente *big data* e inteligência artificial tem um papel relevante nessa evolução (BRIEL, 2018). Isso porque depende da habilidade da organização de não apenas coletar, analisar e integrar os dados da base de clientes, mas sobretudo de utilizar de algoritmos como o aprendizado de máquina para

personalizar as interações de forma ágil e ajudar a construir uma conexão com os clientes.

Essas tecnologias, em conjunto com o modelo RFV – Recência, Frequência e Valor, apoiam a personalização visando quantificar vendas, promoções, preferências de consumo, intervalo entre compras, valor transacionado, entre outras informações para a construção e gerenciamento de uma relação de longo prazo com os clientes, também conhecido como *Customer Relationship Management* (CRM), e consequente fidelização e rentabilidade, fator crucial para a perpetuidade dos negócios (ANSHARI *et al.*, 2019).

Com o crescimento, nos últimos anos, do número de transações realizadas no *e-commerce* e demais plataformas digitais disponíveis cresce também a parcela de vendas identificadas das empresas, isto é, em que é possível cruzar o dado transacional com as informações do cliente. Se por um lado tem-se empresas totalmente digitais, em que não há um espaço físico para realização das compras por parte dos consumidores, existem segmentos em que as vendas de lojas físicas ainda representam quase a totalidade das vendas.

O setor de Q*uick Service Restaurant*, usualmente conhecido como *fast-food* em inglês, é um dos setores que possui um grande desafio no que tange à identificação das vendas que permite a posterior interpretação dos dados e consequente conhecimento do comportamento dos consumidores. Apenas uma pequena parcela das vendas é proveniente de compras pelo aplicativo. Dessa maneira, dentre as alternativas adotadas por uma das redes de *fast-food* no Brasil foi a criação de um programa de fidelidade, onde o cliente fornece o CPF nas compras em canais físicos em troca de pontos que, posteriormente, podem ser usados para resgatar produtos gratuitos ou com um maior desconto a depender da preferência do cliente. Dentre os inúmeros benefícios que programas como esse trazem para as companhias, a citar o aumento de frequência e retenção, é também uma estratégia que auxilia na identificação das vendas para a personalização das comunicações de *marketing.*

Com a criação do programa, a clusterização dos clientes em grupos com comportamentos semelhantes sem a escolha inicial de critérios para diferenciá-los é possível, porém não é realizada atualmente. Ao fazê-lo, aumenta-se a vantagem competitiva da companhia frente aos seus concorrentes. Isto porque os mesmos enfrentam dificuldades de identificação de vendas, que são inerentes do setor, e não

possuem a mecânica do programa de fidelidade para alavancar a coleta dos dados necessários.

Isto posto, o trabalho visa, por meio de algoritmos, a clusterização e classificação da base de clientes pertencentes ao programa de fidelidade de uma rede de *fast-food* no Brasil, possibilitando a criação, por parte da companhia, de comunicações mais efetivas e personalizadas para os clientes.

1.1 OBJETIVOS

Na presente seção são apresentados o objetivo geral e, também, os objetivos específicos do trabalho.

1.1.1 Objetivo geral

O objetivo geral do presente trabalho é a criação de grupos de clientes com perfis de hábitos de compra semelhantes, por meio da aplicação de três algoritmos de clusterização e de dois algoritmos de classificação para que a empresa em estudo possa conhecer melhor sua base de clientes e com isso identificar comportamentos para então traçar estratégias de comunicação personalizadas.

1.1.2 Objetivos específicos

Visando a obtenção do objetivo geral do trabalho propõe-se os seguintes objetivos específicos:

- Avaliar na literatura os métodos de clusterização e classificação de clientes;
- Definir, dentre as variáveis disponíveis para coleta, as que possuem a maior importância para a clusterização além do modelo de rentabilidade RFV (Recência, frequência e valor);
- Comparar o desempenho de três algoritmos de clusterização e de dois algoritmos de classificação visto que os algoritmos possuem abordagens e características diferentes e seus resultados podem ser distintos.

1.2 LIMITAÇÕES DO TRABALHO

O presente trabalho está limitado ao conjunto de dados de vendas identificadas nacionalmente do programa de fidelidade, ferramenta que permite atrelar a venda a um cliente específico, de uma rede americana de *fast-food* no Brasil.

Entretanto, como hoje a empresa tem uma comunicação massiva, entende-se que com o volume atual de dados já é possível iniciar comunicações personalizadas e começar a obter resultados positivos, ainda que pequenos, mas que possui potencial de escalabilidade ao longo do tempo com a maturidade dos dados.

1.3 JUSTIFICATIVA

A personalização é entendida como uma vantagem competitiva importante em grande parte dos setores presentes hoje no mercado, porém a estratégia de ofertas e comunicações individualizadas é um desafio substancial do qual, em geral, as empresas não estão preparadas para executar optando muitas vezes pela criação de um relacionamento de curto prazo sem foco em retenção dos clientes e extensão para uma relação de longo prazo, conforme explicitam Anshari *et al.* (2019).

Nesse cenário e, sobretudo, visando conhecer melhor o cliente a partir de dados demográficos, cadastrais e transacionais, a clusterização desses consumidores em grupos com comportamentos semelhantes entre si sem a predefinição de um critério para segmentá-los é um primeiro passo em direção à criação de valor ao cliente e ao negócio.

Essa clusterização permite otimizar a rentabilidade da companhia com a oferta dos descontos certos para os clientes certos, possibilitando a recomendação de produtos com a estratégia de *cross-sell,* termo em inglês que se refere à venda cruzada e *up-sell,* termo também em inglês que não possui uma tradução literal, porém diz respeito a uma venda adicional para aumentar o valor gasto por compra de um cliente, por exemplo.

Além disso, permite ainda a construção de uma relação de longo prazo evitando *churn* – termo que se refere à perda de um cliente, para um concorrente possivelmente, e aumentando o *life-time value* dos clientes da empresa, que se refere ao lucro gerado por um consumidor durante o seu ciclo de vida na companhia e reflete

o impacto financeiro que a empresa pode vir a ter no momento da perda desse cliente para a concorrência.

Dessa forma, o presente trabalho será o primeiro estudo do tema dentro da companhia em questão, sendo uma alternativa no conhecimento do comportamento dos clientes a partir da aplicação de algoritmos de clusterização e classificação, permitindo à companhia, não somente a posterior criação de comunicações de *marketing* baseadas nas características em comum dos grupos, como também, a difusão do uso dessa tecnologia para tal aplicação.

1.4 ESTRUTURA DO TRABALHO

Com o propósito de cumprir com os objetivos do trabalho abordados preliminarmente, a presente dissertação está estruturada em cinco capítulos, sendo o primeiro deles a presente introdução com a abordagem inicial ao tema, principais objetivos e limitações do trabalho.

No segundo capítulo apresenta-se a revisão sistemática e bibliográfica da literatura para conhecimento dos métodos de clusterização e classificação de clientes, assim como uma fundamentação teórica acerca dos conceitos envolvidos no trabalho, principalmente sobre os algoritmos utilizados e métricas de avaliação de desempenho relacionadas.

Adentrando o terceiro capítulo tem-se em detalhes os aspectos metodológicos do presente trabalho, a título de exemplo pode-se citar a classificação da pesquisa no que diz respeito a sua natureza, abordagem, e instrumentos, bem como as etapas metodológicas aplicadas para alcançar os objetivos estabelecidos previamente.

No quarto capítulo encontram-se os resultados obtidos com a aplicação, a partir da coleta dos dados, dos métodos selecionados e a avaliação das métricas de desempenho relacionadas.

Por fim, o quinto capítulo aborda as considerações finais a respeito dos resultados alcançados e sugestões para pesquisas futuras.

2 REVISÃO DA LITERATURA

O corrente capítulo visa elucidar conceitualmente temas que serão aplicados no estudo e está seccionado em 2 partes. São elas: i) Revisão bibliográfica e sistemática da literatura e ii) Referencial teórico.

2.1 REVISÃO BIBLIOGRÁFICA E SISTEMÁTICA DA LITERATURA

De forma a definir a metodologia a ser aplicada, uma revisão sistemática e bibliográfica da literatura foi realizada com o objetivo de identificar os principais trabalhos correlatos ao tema abordado, sendo possível mapear os algoritmos aplicados no que diz respeito à clusterização e classificação de clientes, assim como avaliar prováveis lacunas que possam ser estudadas na presente pesquisa, conforme propõem MacLure *et al*. (2016).

Para tal, 5 fases consecutivas foram seguidas, assim como sugerem Denyer e Tranfield (2009), De Medeiros *et al*. (2014) e Garza-Reyes (2015): i) Formulação da questão; ii) Localização dos estudos; iii) Avaliação e seleção dos estudos; iv) Análise e síntese e v) Utilização dos resultados (FIGURA 1).

FIGURA 1 – METODOLOGIA DA REVISÃO SISTEMÁTICA DA LITERATURA

FONTE: A autora (2022) com base em Denyer e Tranfield (2009), De Medeiros *et al*. (2014) e Garza-Reyes (2015).

A formulação da questão foi pautada no objetivo da revisão sistemática da literatura (RSL), a partir do qual foram definidas as palavras-chave, seja inclusão ou exclusão. Dessa forma o tema abordado é “Algoritmos para clusterização e classificação de clientes”, onde busca-se responder à 4 principais questões:

- Quais algoritmos podem ser aplicados no contexto de clusterização e classificação de clientes?

- Para esses algoritmos existe a necessidade de definição de parâmetros (Ex.: Número de *clusters*, distância, etc.)? Se sim, quais são os métodos aplicados para tal definição?
- Quais métricas de desempenho são comumente aplicadas?
- Quais variáveis transacionais são utilizadas?

As palavras-chave utilizadas são *cluster, client, customer, consumer* e RFM (Sigla em inglês que corresponde à RFV – Recência, Frequência e Valor), para nenhuma delas utilizou-se a exclusão, apenas a inclusão. Para o termo RFM buscou-se ao longo de todo o artigo, enquanto para os demais termos os mesmos deveriam estar presentes no título, resumo ou palavras-chave dos estudos. Dentre os operadores booleanos utilizados tem-se *OR* e *AND* para que seja possível identificar artigos que considerem qualquer um dos 3 termos selecionados que se referem à clientes e para que eles sejam encontrados obrigatoriamente no mesmo contexto em que a palavra *cluster* e RFM estão presentes.

Ainda sobre as palavras-chave, o sifrão ($) foi utilizado para a busca para que os termos referentes à clientes possam ser encontrados tanto no singular quanto no plural.

Com o objetivo de filtrar os artigos mais relevantes resultantes, demais critérios de avaliação foram estabelecidos, são eles: i) Tipo de documento – Selecionou-se apenas artigos e ii) Idioma – Apenas artigos publicados na língua inglesa foram selecionados. O ano de publicação não foi utilizado como filtro, considerou-se a totalidade de artigos publicados até Fevereiro de 2023, período em que a revisão bibliográfica foi realizada.

No que tange às bases, a fim de maximizar a quantidade de trabalhos relevantes encontrados, a pesquisa foi conduzida em bases de dados eletrônicas por serem mais acessíveis e permitirem a replicabilidade. Assim para localização dos estudos optou-se pela *Web of Science* e *Scopus* tendo também em vista o acesso ao conteúdo assinado disponível à instituição de ensino Universidade Federal do Paraná (UFPR).

A partir da condução por meio desse método sistemático e da apresentação resumida dos elementos utilizados no QUADRO 1, a revisão passa a ser replicável e atualizável, cumprindo o objetivo de sintetizar as evidências relativas ao tema.

QUADRO 1 – ESTRATÉGIA DE BUSCA

Estratégia de busca	Termos utilizados
Segmentação do tema	Algoritmos para clusterização e classificação de clientes
Questões de pesquisa	1. Quais algoritmos podem ser aplicados no contexto de clusterização e classificação de clientes? 2. Para esses algoritmos existe a necessidade de definição de parâmetros (Ex.: Número de *clusters*, distância, etc.)? Se sim, quais são os métodos aplicados para tal definição? 3. Quais métricas de desempenho são comumente aplicadas? 4. Quais variáveis transacionais são utilizadas?
Bases	*Web of Science* e *Scopus*
Strings de busca	*Web of science*: TS=((client$ OR customer$ OR consumer$) AND classification AND clustering) AND ALL=(RFM)
	Scopus: TITLE-ABS-KEY ((client$ OR customer$ OR consumer$) AND classification AND clustering) AND ALL (“RFM”)
Tipo de documento	Artigo
Idioma	Inglês
Ano de publicação	Todos os estudos publicados até Fevereiro de 2023

FONTE: A autora (2023).

LEGENDA: TS – Campos Título, Resumo e Palavras-chave.

Na avaliação e seleção dos estudos, apenas os estudos relevantes para o tema foram considerados. Dessa forma, excluiu-se os artigos por duplicidade tendo em vista que para a localização dos estudos selecionou-se duas bases de dados, e foi também realizada a exclusão de artigos que, a partir do título e do resumo, não possuem relação com o tema abordado na presente pesquisa, sendo, portanto, inadequados para a busca por respostas às questões previamente formuladas.

O emprego da estratégia de busca resultou em 39 artigos na base de dados *Scopus* e 23 na *Web of Science*. Aplicando a primeira regra de exclusão, referente à duplicidade, restaram 52 artigos e após a exclusão a partir da identificação da adequação dos estudos ao tema do presente estudo restaram 28 artigos que foram lidos em sua totalidade para posterior análise, síntese e utilização dos resultados (TABELA 1 e QUADRO 2).

TABELA 1 – AVALIAÇÃO E SELEÇÃO DOS ESTUDOS

Base	**Quantidade de artigos**
Scopus	39
Web of Science	23
Total de estudos	62
Estudos únicos	52
Estudos selecionados	28

FONTE: A autora (2023).

QUADRO 2 – PUBLICAÇÕES SELECIONADAS PARA A REVISÃO SISTEMÁTICA DA LITERATURA

(Continua)

Estudo	Ano	Título	Autores	Revista
E1	2020	*Classification and Identification of Loyal Customers Using Machine Learning*	Siva R. K. B.; Sai K. P.; Priya P.; Gopi S. M.; Sai K. G.	*Journal of Advanced Research in Dynamical and Control Systems*
E2	2015	*A Clusterbased Data Balancing Ensemble Classifier for Response Modeling in Bank Direct Marketing*	Amini M.; Rezaeenour J.; Hadavandi E.	*International Journal of Computational Intelligence and Applications*
E3	2015	*Twotiered Clustering Classification Experiments for Market Segmentation of Eftpos Retailers*	Singh A.; Rumantir G.	*Australasian Journal of Information Systems*
E4	2016	*Identifying and Segmenting Customers of Pasargad Insurance Company Through RFM Model RFM*	Hamdi K.; Zamiri A.	*International Business Management*
E5	2021	*Customer Segmentation and Profiling for Life Insurance Using Kmodes Clustering and Decision Tree Classifier*	Abdul-Rahman S.; Arifin N.; Hanafiah M.; Mutalib S.	*International Journal of Advanced Computer Science and Applications*
E6	2022	*Efficient Customer Segmentation in Digital Marketing Using Deep Learning with Swarm Intelligence Approach*	Wang C.	*Information Processing and Management*
E7	2007	*Research on Customer Classification Based on Fuzzy Clustering*	Zhang L.; Zhang L.; Chen S.; Cai L.; Yu Y.; Hao S.	*Journal of Computational Information Systems*

QUADRO 2 – PUBLICAÇÕES SELECIONADAS PARA A REVISÃO SISTEMÁTICA DA LITERATURA

(Continua)

Estudo	Ano	Título	Autores	Revista
E8	2016	*Using Data Mining and Neural Networks Techniques to Propose a New Hybrid Customer Behaviour Analysis and Credit Scoring Model in Banking Services Based on a Developed RFM Analysis Method*	Alborzi M.; Khanbabaei M.	*International Journal of Business Information Systems*
E9	2022	*Electric Vehicle User Classification and Value Discovery Based on Charging Big Data*	Hu D.; Zhou K.; Li F.; Ma D.	*Energy*
E10	2018	*Clustering Prediction Techniques in Defining and Predicting Customers Defection the Case of Ecommerce Context*	Rachid A.; Abdellah A.; Belaid B.; Rachid L.	*International Journal of Electrical and Computer Engineering*
E11	2023	*Research on Precision Marketing Strategy of Commercial Consumer Products Based on Big Data Mining of Customer Consumption*	Fan L.	*Journal of The Institution of Engineers (India): Series C*
E12	2019	*Customer Behavior Mining Framework CBMF Using Clustering and Classification Techniques*	Abdi F.; Abolmakarem S.	*Journal of Industrial Engineering International*
E13	2023	*Clustering Mixedtype Player Behavior Data for Churn Prediction in Mobile Games*	Perišic A.; Pahor M.	*Central European Journal of Operations Research*

QUADRO 2 – PUBLICAÇÕES SELECIONADAS PARA A REVISÃO SISTEMÁTICA DA LITERATURA

(Continua)

Estudo	Ano	Título	Autores	Revista
E14	2009	*Classifying the Segmentation Of Customer Value via RFM Model and RS Theory*	Cheng C.; Chen Y.	*Expert Systems with Applications*
E15	2020	*A Hybrid Classification Model for Churn Prediction Based on Customer Clustering*	Tang Q.; Xia G.; Zhang X.	*Journal of Intelligent and Fuzzy Systems*
E16	2018	*A Novel Model for Product Bundling and Direct Marketing in Ecommerce Based on Market Segmentation*	Beheshtian-Ardakani A.; Fathian M.; Gholamian M.	*Decision Science Letters*
E17	2016	*An Empirical Study on Customer Risk Management in Banking Industry Applying Kmeans and RFM Methods Evidence from Two Iranian Private Banks*	Farughi H.; Alaniazar S.; Mousavipour S.	*International Journal of Risk Assessment and Management*
E18	2021	*Adherence Predictor Variables in AIDS Patients an Empirical Study Using the Data Miningbased RFM Model*	Li M.; Wang Q.; Shen Y.	*AIDS Research and Therapy*

QUADRO 2 – PUBLICAÇÕES SELECIONADAS PARA A REVISÃO SISTEMÁTICA DA LITERATURA

(Continua)

Estudo	Ano	Título	Autores	Revista
E19	2016	*Knowledge Discovery from Patients Behavior via Clusteringclassification Algorithms Based on Weighted ERFM nd CLV Model an Empirical Study in Public Health Care Services*	Hosseini Z.; Mohammadzadeh M.	*Iranian Journal of Pharmaceutical Research*
E20	2022	*RFM Analysis for Customer Segmentation Using Machine Learning a Survey of a Decade of Research*	Chavhan S.; Dharmik R.; Jain S.; Kamble K.	*3C TIC*
E21	2021	*Using Improved RFM Model to Classify Consumer in Big Data Environment*	Sun G.; Xie X.; Zeng J.; Jiang M.; Huang Y.; Xiao Y.	*International Journal of Embedded Systems*
E22	2022	*Elucidating Strategic Patterns from Target Customers Using Multistage RFM Analysis*	Chattopadhyay M.; Mitra S.; Charan P.	*Journal of Global Scholars of Marketing Science*
E23	2022	*Multibehavior RFM Model Based on Improved SOM Neural Network Algorithm for Customer Segmentation*	Liao J.; Jantan A.; Ruan Y.; Zhou C.	*IEEE Access*
E24	2021	*User Value Identification Based on Improved RFM Model and Kmeans Plus Plus Algorithm for Complex Data Analysis*	Wu J.; Shi L.; Yang L.; Niu X.; Li Y.; Cui X.; Tsai S.; Zhang Y.	*Wireless Communications & Mobile Computing*

QUADRO 2 – PUBLICAÇÕES SELECIONADAS PARA A REVISÃO SISTEMÁTICA DA LITERATURA

(Conclusão)

Estudo	Ano	Título	Autores	Revista
E25	2022	*LRFMV an Efficient Customer Segmentation Model for Superstores*	Mahfuza R.; Islam N.; Toyeb M.; Emon M.; Chowdhury S.; Alam M.	*Plos One*
E26	2018	*A New Approach for Customer Clustering by Integrating the LRFM Model and Fuzzy Inference System*	Zoeram A.; Mazidi A.	*Iranian Journal of Management Studies*
E27	2018	*Hybrid Soft Computing Approach Based on Clustering Rule Mining and Decision Tree Analysis for Customer Segmentation Problem Real Case of Customercentric Industries*	Khalili-Damghani K.; Abdi F.; Abolmakarem S.	*Applied Soft Computing*
E28	2018	*LDCFR a New Model to Determine Value of Airline Passengers*	Dehghanizadeh M.; Fathian M.; Gholamian M.	*Tourism and Hospitality Research*

FONTE: A autora (2023).

A partir dos estudos selecionados, foi possível a elaboração de uma revisão bibliométrica da literatura. A FIGURA 2 ilustra a distribuição temporal dos estudos.

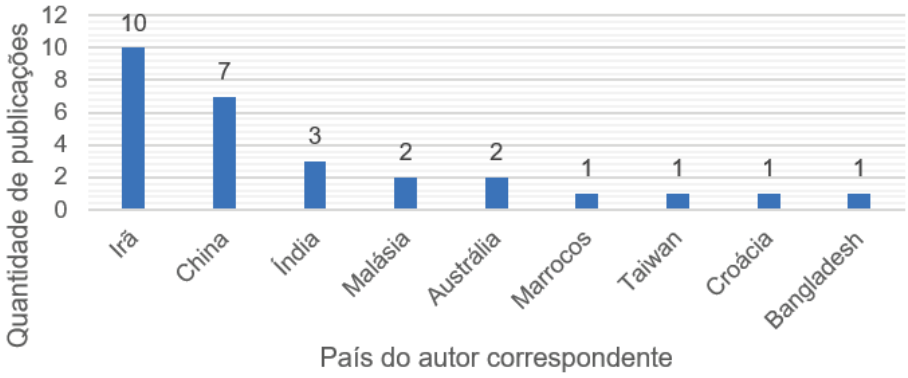
FIGURA 2 – DISTRIBUIÇÃO DOS ESTUDOS POR ANO DE PUBLICAÇÃO

FONTE: A autora (2023).

O número de publicações sobre clusterização e classificação de clientes a partir de dados transacionais, como a RFV, cresce nos últimos 5 anos, mostrando o aumento de relevância do tema.

Também foi possível analisar as publicações quanto ao país de afiliação dos autores correspondentes, a FIGURA 3 apresenta a visualização gráfica. Os países que mais se destacam são Irã e China com 10 e 7 publicações, respectivamente.

FIGURA 3 – DISTRIBUIÇÃO DOS ESTUDOS POR NACIONALIDADE

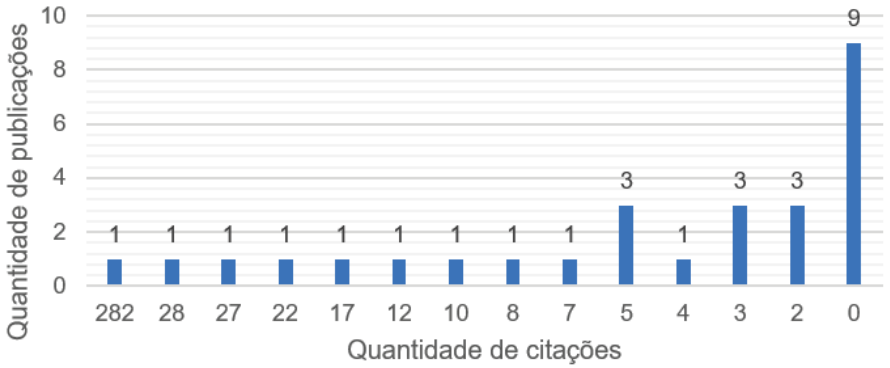


FONTE: A autora (2023).

Quanto às revistas por meio do qual os estudos foram publicados, não há um destaque. Cada um dos estudos foi publicado em uma revista diferente, dessa forma, há 28 revistas com 1 publicação cada.

Por fim, a FIGURA 4 apresenta a distribuição dos estudos por quantidade de citações onde observa-se que 68% dos artigos possuem ao menos 1 citação.

FIGURA 4 – DISTRIBUIÇÃO DOS ESTUDOS POR QUANTIDADE DE CITAÇÕES



FONTE: A autora (2023).

A avaliação e síntese dos artigos explicitada a seguir foi elaborada de forma a responder às questões formuladas para a estratégia de busca.

Siva *et al.* (2020) com o objetivo de prever a fidelidade dos clientes, utilizaram o K-*Means* como algoritmo de clusterização e, posteriormente, a árvore de decisão como classificador. O modelo RFV foi usado como base do estudo, que contou com a aplicação do método do cotovelo e do coeficiente de silhueta na definição do número de *clusters* para a clusterização, assim como o cálculo da acurácia para avaliar o desempenho do modelo de predição treinado.

Amini *et al.* (2015) propuseram um método de classificação conjunto usando uma combinação de clusterização e subamostragem, onde as previsões de vários classificadores são combinadas para obtenção de melhores resultados. O objetivo é otimizar as campanhas de *marketing* de um banco varejista português, prevendo, a partir do classificador, quais clientes são mais propensos a aderir as campanhas promocionais e assim focar no contato com os mesmos somente. Como dados utilizou-se: i) Dados pessoais do cliente – Como idade, educação, entre outros; ii) Informações bancárias – Como quantia investida, empréstimos feitos, avaliação de crédito entre outros; iii) Informações sobre o último contato – Como forma de contato

e data e iv) Informações históricas – Por exemplo resultado de campanhas ofertadas anteriormente, número total de contatos já realizados, entre outros. Como método de clusterização adotou-se o K-*Means*, e o número de *clusters* foi definido a partir do *SD Validity Index.* Já para os classificadores os autores utilizaram: i) Regressão logística (LR); ii) Árvore de decisão; iii) Rede Neural Artificial (ANN) e iv) Máquina de Vetores de Suporte (SVM). Como métodos de avaliação de desempenho usou-se: i) Acurácia da matriz de confusão; ii) Taxa de correção balanceada (BCR); iii) Área sob a curva ROC – Características Operacionais do Receptor (AUC) e iv) Taxa de verdadeiro positivo (TPR), onde comparou-se o resultado do conjunto de classificação *versus* os classificadores individualmente. O SVM foi o algoritmo com melhores resultados e que foi usado para a construção do classificador conjunto que melhorou a acurácia da predição, otimizando os recursos investidos nas campanhas de *marketing* bancárias.

Singh e Rumantir (2015) realizaram seu estudo no setor bancário, utilizou-se como base as transações bancárias de 18 dias de um dos maiores bancos australianos, o equivalente a mais de 77 milhões de transações de mais de 1 milhão de clientes. Um volume tão grande de dados torna até mesmo os cálculos básicos de RFV muito mais demorados e com um uso de recursos muito mais intenso, para isso os autores utilizaram o método de paralelização, com um *cluster* de *Intel Xeon* e *CPUs AMD Opteron* de várias velocidades com 16 núcleos, 32GB de RAM e 500GB de disco rígido. Como métodos de clusterização utilizou-se o K-*Means* e o método hierárquico de Ward. Dado a volumetria de dados, no caso do método hierárquico a criação dos *clusters* se deu por meio do processo aglomerativo com 60% dos dados e com os outros 40% os *clusters* são formados com base na distância euclidiana. Como forma de avaliação de desempenho e de obtenção do número ideal de *clusters* os autores utilizaram o índice de Dunn e o índice de Davies-Bouldin. Como forma de obter mais informações sobre os *clusters* formados pelo método hierárquico, algoritmo com melhor resultado na avaliação dos índices, os autores utilizaram a árvore de decisão e incluíram outras variáveis além da RFV para determinar se um atributo utilizado no estudo é de fato uma característica relevante de um cliente. O algoritmo de classificação usado foi avaliado por meio da acurácia da matriz de confusão.

Hamdi e Zamiri (2016) clusterizaram, por meio do método K-*Means,* clientes de uma seguradora de veículos iraniana. Para tal, utilizaram os dados transacionais de recência desde a última renovação de apólice de seguro, frequência de renovação das apólices no período analisado e valor pago pelas mesmas.

Abdul-Rahman *et al*. (2021) aplicaram a clusterização e a classificação em uma empresa do setor de seguros na Malásia para que os clientes possam, a partir da criação dos grupos, receber planos de seguros que satisfaçam as suas necessidades. Para a clusterização o método usado foi o K-*Modes*, com o apoio do método do cotovelo para a definição do número de *clusters* e de dois métodos de avaliação de desempenho, são eles: i) Coeficiente de Silhueta e ii) Índice Calinski-Harabasz. Como classificador, o algoritmo da árvore de decisão com validação cruzada de K-*Fold* foi usado e a acurácia foi apurada como avaliação da classificação realizada. Como estudo futuro os autores indicam usar o algoritmo da floresta aleatória como classificador.

Wang (2022) propõe a clusterização e segmentação de clientes de um *e-commerce* para determinar os clientes com tipo semelhante de compra de produtos. O algoritmo de clusterização usado foi o *Self Organizing Neural Network* (SONN) e a performance foi avaliada com base nos erros de quantização, referente a resolução do mapa que é inerente à construção do modelo e topológico, que mede a preservação da topologia e a continuidade do mapa. Já o classificador selecionado após a comparação da acurácia de diferentes métodos como *Deep Neural Network* (DNN), *Back Propagation Network* (BPN), árvore de decisão e *Convolutional Neural Network* (CNN) foi o DNN.

Zhang *et al*. (2007) estudaram os dados de clientes de uma empresa de eletrônicos, com base em uma extensão do modelo RFV, adotando uma variável a mais que é um índice de avaliação das informações sobre os dados transacionais dos clientes. O classificador usado foi o *Fuzzy* C-*Means.*

Alborzi e Khanbabaei (2016) propõem um modelo híbrido para avaliação da pontuação de crédito de um cliente podendo ser aplicado para classificar clientes bancários de alto valor. Como variáveis, um modelo expandido de RFV é utilizado, já para a classificação redes neurais são aplicadas.

Hu *et al.* (2022) clusterizam usuários de veículos elétricos para auxiliar nas estratégias de *marketing,* aumentar a lucratividade do setor e na fidelização do usuário. Com relação as variáveis têm-se os dados de carregamento dos veículos, como por exemplo localização das estações de carregamento usadas, tempo de carregamento, valor pago, entre outros. Os autores expandiram o tradicional modelo RFV incluindo 2 novas variáveis, *lenght* (L) e *trouble* (T), onde L significa o tempo entre o primeiro e último carregamento e T a quantidade de tentativas fracassadas de

carregar o veículo devido às falhas nos equipamentos de carregamento. Para a clusterização, 3 algoritmos foram usados, são eles: i) *Entropy-Cluster*; ii) K-*Means* e iii) *Fuzzy* C-*Means*. O DBSCAN foi usado para determinar o número ideal de *clusters* resultando em um parâmetro inicial para a aplicação do K-*Means*. Como forma de avaliação dos métodos foi calculada a inércia intra-*cluster*, resultando no *entropy-cluster* como algoritmo com melhor performance.

Rachid *et al.* (2018) estudaram os clientes do *e-commerce* de um dos maiores varejistas do Marrocos especializado em eletrônicos, moda, eletrodomésticos e artigos infantis. A clusterização utiliza não apenas o RFV, mas também uma variável adicional, o L, que foi empregada por Hu et al. (2022). O método do cotovelo e o coeficiente de silhueta foram usados para determinar o número de *clusters* para a posterior aplicação do algoritmo de clusterização K-*Means*. Para a criação do modelo de predição de *churn* – clientes que deixam de comprar com a empresa*,* 3 classificadores, com validação cruza de K-*Fold*, foram aplicados, são eles: i) Rede neural artificial; ii) Árvore de decisão simples e iii) Árvore de decisão conjunta. Para a avaliação do desempenho dos classificadores a matriz de confusão foi calculada, identificando que a árvore de decisão conjunta é o que apresenta os melhores resultados. O estudo contribui para que a empresa possa aplicar medidas de retenção de clientes para os que forem identificados pelo modelo como em risco de *churn.*

Fan (2023) aplicou, baseado no modelo RFV, 3 algoritmos de clusterização: i) K-*Means*; ii) PSO-K-*Means* – Modelo híbrido entre o K-*Means* e a otimização por enxame de partículas (PSO) e iii) IPSO-K-*Means* – Que é também um modelo híbrido como o PSO-K-*Means* porém com algumas melhorias, sendo esse último o algoritmo com melhor performance.

Abdi e Abolmakarem (2019) avaliaram os dados de clientes de uma empresa de telecomunicação. Primeiramente, a clusterização é utilizada para analisar o portfólio e os clientes são divididos com base em características sociodemográficas usando o algoritmo K-*Means*. Para a definição do número de *clusters* os autores fizeram uso do índice de Davies-Bouldin. Em seguida, a análise do agrupamento realizado é feita com base em dois critérios, a quantidade de horas que os serviços de telecomunicações são utilizados e o número de serviços selecionados pelos clientes de cada grupo. Já a segunda fase do estudo tem como objetivo a previsão do nível de atratividade dos novos clientes e também o comportamento de *churn*. Para

tal, 2 classificadores foram utilizados: i) Rede neural artificial e ii) Árvore de decisão e a matriz de confusão foi calculada de forma a avaliar o melhor modelo preditivo.

Perišic e Pahor (2023) propõem, no respectivo estudo, a clusterização de clientes do setor de jogos para celular e criação de um modelo de previsão de *churn* que leve em consideração tanto variáveis quantitativas, quanto variáveis categóricas. Para a clusterização foi usado o algoritmo *Partitioning Around Medoids* (PAM).

Cheng e Chen (2009) analisaram os dados dos clientes de uma empresa da indústria eletrônica. A clusterização se deu por meio do algoritmo K-*Means* baseado nos dados transacionais agrupados em uma RFV. Os autores propuseram a aplicação do método de classificação *RS Theory* (LEM2) que, quando tem a sua acurácia comparada aos demais modelos (Árvore de decisão, rede neural artificial e naive bayes), tem um melhor desempenho. A classificação foi utilizada com o objetivo de extrair as regras de agrupamento para que o CRM da companhia seja mais eficaz.

Tang *et al.* (2020) também propuseram a previsão de *churn* dos clientes, onde para a etapa de clusterização foi aplicado o algoritmo K-*Means* e para a etapa de previsão foi aplicado o algoritmo de classificação árvore de decisão aumentada por gradiente (GBDT).

Beheshtian-Ardakani *et al.* (2018) com o objetivo de segmentar clientes de um *e-commerce* com base na fidelidade à marca usando dados transacionais, posteriormente agrupando conforme o mercado no qual se enquadrava baseado nos dados qualitativos geográficos e demográficos e, por último, por tipo de produtos adquiridos para enfim aplicar a recomendação de produtos fizeram a comparação entre os resultados do agrupamento de 3 algoritmos de clusterização, são eles: i) *K-Means*, ii) SPSS *Two-Step,* e iii) *Self-Organized Maps* (SOM). Em todos os casos de aplicação o algoritmo que proporcionou *clusters* com maiores distinções, por meio do emprego do coeficiente da silhueta, foi o *K-Means.* Os algoritmos de classificação usados são do tipo preditores visto que o objetivo é prever, para os *clusters* formados, a quantidade de produtos adequada a ser recomendada são eles: i) Redes Neurais Artificiais (ANN); ii) Redes Bayesianas; iii) K-Vizinhos mais próximos (KNN); iv) Regressão logística (LR); v) Máquina de Vetores de Suporte (SVM) e vi) Árvore de decisão, onde o SVM é o algoritmo selecionado com maior acurácia.

Farughi *et al*. (2016) estudaram os clientes de dois bancos privados iranianos, onde o algoritmo K-*Means* foi aplicado para a clusterização com base no modelo RFV estendido: WRFM, onde o W representa pesos que são aplicados para as variáveis

do modelo tradicional RFV, calculado a partir da matriz de comparação em pares da análise hierárquica de processos.

Li *et al.* (2021) apresentam um modelo de previsão de adesão ao tratamento de pacientes com AIDS, utilizando o modelo RFV e análise de clusterização para obter variáveis preditoras de adesão. Foram analisados 257.305 dados de 16.440 pacientes diagnosticados com AIDS em Shanghai de agosto de 2009 a dezembro de 2019. Foram utilizados 3 métodos de clusterização: i) K-*Means*; ii) *Self-Organized Maps* (SOM) e iii) *Two-step clustering* e 4 algoritmos de classificação: i) Árvore de decisão, ii) Árvore de decisão (CART – *Classification and Regression Trees*); iii) Árvore de decisão (CHAID – *Chi-square Automatic Interaction Detector*) e iv) *Quick, Unbiased, Efcient, Statistical Tree* (QUEST). O K-*Means* foi considerado o melhor método de clusterização pela nota de qualidade do modelo e a árvore de decisão o melhor algoritmo de decisão com acurácia de 100%. Como resultado final, o modelo dividiu os pacientes em grupos com boa e má adesão ao tratamento de AIDS.

Hosseini e Mohammmadzadeh (2016) discutem a importância da implementação de um sistema de gerenciamento de relacionamento com o cliente (CRM) em hospitais, com o objetivo de identificar pacientes potenciais e alvo, aumentar a fidelidade e satisfação dos pacientes e maximizar a lucratividade. É proposto um modelo estendido de RFV, chamado RFMD, baseado em serviços de saúde para um hospital público no Irã, que inclui um parâmetro adicional (Duração) para estimar o valor vitalício do cliente (CLV) para cada paciente. Utilizou-se o algoritmo K-*Means* como método de clusterização, a segmentação por meio do *Customer Lifetime Value* (CLV) e a *Árvore* de Decisão (CHAID) como algoritmo de classificação para prever clientes-alvo, em potencial e leais, a fim de implementar um CRM mais forte. No final, o autor compara o método K-*Means* e a segmentação do CLV pela acurácia da árvore de decisão aplicada para os dois modelos de agrupamento, e o K-*Means* tem o melhor desempenho.

Chavhan *et al.* (2022) examinam os dados transacionais de clientes do *e-commerce* de uma loja varejista. Para a segmentação do cliente, os autores utilizam o método RFV, pois permite agrupar com base nos requisitos e direcionar as estratégias de *marketing*. Para a clusterização, 3 métodos foram aplicados: i) K-*Means*; ii) *Fuzzy* C-*Means* e iii) *Repetitive Median* K-*Means*. O *Fuzzy* C-*Means* teve a melhor performance de acordo com a largura de silhueta, porém os autores alertam para o tempo de processamento e o número de iterações do algoritmo.

Sun *et al.* (2021) reforçam a importância do relacionamento com o cliente (CRM) para a operação das empresas atualmente, principalmente na era do *big data* onde o foco passa a ser os clientes. Dessa forma, o estudo visa clusterizar os clientes com o uso do algoritmo K-*Means* para distinguir clientes com maior e menor valor para a empresa para então aplicação de estratégias de relacionamento. O método do cotovelo foi aplicado para avaliar o desempenho da clusterização.

O estudo de Chattopadhyay *et al.* (2022) tem como objetivo demonstrar um modelo de previsão de clientes lucrativos para um varejista *online* do Reino Unido. Foi aplicado o método de clusterização K-*Means* para identificar padrões de clientes com base em atributos RFV e para definir o número ideal de *cluster* foi utilizado o índice Davies-Bouldin. Após isso, os autores usaram 6 modelos de previsão: i) Modelo linear generalizado, ii) Análise discriminante linear, iii) Análise discriminante quadrático, iv) Rede neural artificial, v) Máquina de Vetores de Suporte (SVM) e vi) Spline de regressão adaptativa multivariada (MARS). Os modelos de previsão tiveram desempenho parecidos em média, porém o melhor, segundo o teste estatístico ACC e AUC, foi o Modelo linear generalizado. Os resultados mostraram que a acurácia de previsão de que um cliente faria uma compra nos próximos seis meses foi superior a 90% para os 50% melhores clientes do *Cluster* 2, que foi considerado o grupo mais rentável de clientes. Os autores sinalizam limitações de generalização, pois foi realizado em um único conjunto de dados de uma única empresa em um país e indicam que futuras pesquisas podem considerar outros dados relevantes, como dados demográficos e de produtos, para aprimorar ainda mais a seleção de clientes-alvo mais rentáveis.

Liao *et al.* (2022) propõem uma extensão ao modelo RFV baseado em múltiplos comportamentos (MB – *Multiple behavior*) de interação no *e-commerce*, como por exemplo cliques, visualizações, a ação de adicionar ao carrinho um produto, a ação de favoritar um item, entre outros. Os valores do MB-RFM são usados para agrupar os clientes em por meio do algoritmo de clusterização SOM. O modelo também provou ser mais preciso na classificação de clientes do que modelos de RFV tradicionais que consideram apenas um comportamento de interação entre usuário e item. O artigo conclui que o modelo MB-RFV pode ajudar as empresas a entender melhor seus clientes parar fazer recomendações personalizadas e estratégias de precificação e promoções.

Wu *et al.* (2021) clusterizaram, por meio do algoritmo K-*Means*++, os clientes de um *e-commerce* alimentício com base em uma extensão do modelo RFV, com a inclusão de mais duas variáveis: L – Também usado por outros autores como Hu *et al.* (2022) e Rachid *et al.* (2018) e P – Atributos de compra repetida, refere-se ao número de compras de uma categoria de bens adquiridos por um determinado usuário no tempo de referência. O número de *clusters* ideal foi selecionado com base no coeficiente de silhueta. Comparando os resultados do agrupamento formado com base no modelo RFV tradicional e do modelo com mais variáveis, os autores concluem que o modelo mais extenso promove melhores resultados de agrupamento.

O modelo RFV é amplamente utilizado para determinar segmentos de clientes lucrativos e analisar o lucro em varejistas. O modelo RFVD foi posteriormente desenvolvido, adicionando o parâmetro de Duração, como uma versão aprimorada para identificar grupos de consumidores mais relevantes e exatos. Mahfusa *et al.* (2022) propõem um novo modelo de segmentação de clientes, o RFVDV, que inclui um novo parâmetro, Volume, para mostrar a relação direta entre lucro e quantidade de produtos vendidos em varejistas. Os dados foram analisados utilizando algoritmos de clusterização: i) K-*Means*; ii) K-*Medoids* e iii) *Mini Batch* K-*Means*. O número de *clusters* ideal foi definido pelo coeficiente de silhueta. Os resultados mostram que o RFVDV, a partir do algoritmo K-*Means*, cria segmentos de clientes precisos e mantém um maior lucro em comparação com os modelos RFV e RFVD anteriores.

Zoeram e Mazidi (2018) tiveram como objetivo fornecer um método sistemático para analisar as características do comportamento de compra dos clientes, a fim de melhorar o desempenho do sistema de CRM. Para isso, foi utilizado o modelo aprimorado LRFV para analisar o valor do ciclo de vida do cliente, já que o modelo RFV mais básico não leva em consideração a lealdade dos clientes. Este estudo utiliza o *Fuzzy Inference System* (FYS) como método de clusterização. O modelo identificou 16 *clusters* em 5 grandes grupos de clientes para uma empresa de vidros e cristais. A análise dos dados permitiu identificar estratégias de *marketing* adequadas para cada grupo de clientes e para aprimorar a alocação de recursos. O estudo recomenda que a empresa se concentre em reter os clientes mais valiosos e desenvolver ferramentas promocionais apropriadas para cada grupo de clientes. Em geral, o sistema proposto pode ajudar as empresas a entender e analisar as características dos clientes e selecionar as estratégias de *marketing* adequadas para melhorar o desempenho do CRM.

Khalili-Damghani *et al.* (2018) apresentam uma abordagem híbrida de *soft computing* que utiliza clusterização, por meio do algoritmo K-*Means*, e um classificador, por meio do algoritmo da árvore de decisão, para prever em que *cluster* um novo cliente da companhia poderia ser alocado com base em variáveis não transacionais como renda, nível educacional, estrutura familiar (filhos, casamento, entre outras), demografia e estilo de vida (fumante). Para a definição do número ideal de grupos, o índice Davies-Bouldin é utilizado e para a avaliação de desempenho do classificador, a acurácia da matriz de confusão é o método aplicado pelos autores. A metodologia proposta é validada em dois estudos de caso em setores diferentes, seguros e telecomunicações, e demonstra eficácia na identificação de *leads* potencialmente lucrativos e recursos influentes para a tomada de decisão. Os resultados indicam que a metodologia pode ser aplicada em casos da vida real e contribuir para a tomada de decisão de empresas centradas no cliente, porém os autores indicam possíveis limitações do modelo como a inclusão de muitas variáveis que pode tornar o método mais lento e diminuir sua eficácia, assim outros algoritmos alternativos poderiam ser utilizados.

Dehghanizadeh *et al.* (2018) têm como objetivo classificar os clientes de uma companhia aérea com base no seu valor para que sejam oferecidos serviços personalizados. O modelo utilizado para a previsão de valor de cliente foi o LDcFR, em inglês, que utiliza as variáveis de tempo do cliente na base (*Lenght)*, distância total viajada no período, frequência de viagens e tempo desde a última viagem (Recência). Para clusterizar, os autores utilizam o *Imperialist Competitive Algorithm* (ICA), por ser considerado um dos algoritmos evolutivos mais rápido e preciso. O número ideal de *clusters* foi definido a partir do coeficiente de silhueta. Com o objetivo de determinar o valor futuro do cliente, o classificador de cadeias de Markov foi utilizado.

Após finalizada a análise dos 28 artigos, pode-se observar que, quanto ao algoritmo utilizado para a clusterização, grande parte dos estudos faz uso do *K-Means*, já com relação a classificação a árvore de decisão é o mais utilizado, seguido das redes neurais artificias e grande parte deles usados como método de previsão e não como classificador propriamente dito.

Quanto aos parâmetros iniciais para clusterização dos dados ou ainda para a avaliação de desempenho, seja da clusterização ou da classificação, nem todos os autores apresentam os métodos escolhidos. Porém para a clusterização, dentre os principais encontrados, pode-se citar o método do cotovelo, o coeficiente de silhueta

e o índice Davies-Bouldin. Já para a classificação a acurácia da matriz de confusão é o método mais utilizado.

No que diz respeito ao modelo RFV utilizado, pelo menos 35% das publicações utilizam alguma variação às 3 variáveis básicas, como por exemplo o tempo entre a primeira e última compra.

O QUADRO 3 apresenta um resumo dos estudos selecionados com as respectivas metodologias utilizadas.

QUADRO 3 – PUBLICAÇÕES SELECIONADAS PARA A REVISÃO SISTEMÁTICA DA LITERATURA

Estudo	Informações Gerais		Clusterização		Classificação	
	Setor	Variáveis	Algoritmos	Avaliação de desemp./Parâmetros	Algoritmos	Avaliação de desemp.
E1	-	RFV	K-*Means*	Cotovelo e Silhueta	Árvore de decisão	Acurácia da matriz de confusão
E2	Banco	Outros	K-*Means*	SD *Validity Index*	Regressão, árvore decisão, rede neural e máquina de vetores de suporte (SVM)	Acurácia e outros
E3	Banco	RFV	K-*Means* e Hierárquico de Ward	Índice Dunn e Davies-Bouldin	Árvore de decisão	Acurácia da matriz de confusão
E4	Seguro	RFV	K-*Means*	-	-	-
E5	Seguro	-	K-Modes	Cotovelo, Silhueta e Índice Calinski-Harabasz	Árvore de decisão	Acurácia da matriz de confusão
E6	*E-commerce*	RFV	SONN	Erros de quantização e topológico	Árvore decisão e rede neural	Acurácia da matriz de confusão
E7	Indústria eletrônica	RFV	-	-	*Fuzzy* C-*Means*	-
E8	Banco	RFV	-	-	Rede neural	-
E9	Veículos elétricos	Extensão do RFV	K-*Means*, *Fuzzy* C-*Means* e *Entropy-Cluster*	DBSCAN e inércia intra-*cluster*	-	-
E10	*E-commerce*	Extensão do RFV	K-*Means*	Cotovelo e Silhueta	Árvore decisão e rede neural	Diferentes métricas da matriz de confusão
E11	-	RFV	K-*Means*, PSO-K-*Means* e IPSO-K-*Means*	-	-	-
E12	Telecomunicação	Outros	K-*Means*	Índice Davies-Bouldin	Árvore decisão e rede neural	Diferentes métricas da matriz de confusão
E13	Jogos	Outros	PAM	-	-	-
E14	Industria eletrônica	RFV	K-*Means*	-	Modelo dos autores, árvore de decisão, rede neural, naive bayes	Acurácia da matriz de confusão
E15	-	-	K-*Means*	-	Árvore de decisão	-
E16	*E-commerce*	Extensão do RFV	K-*Means*, SPSS *Two-Step* e SOM	Silhueta	Rede neural, Rede Bayesiana, K-Vizinhos mais próximos, regressão, SVM e árvore de decisão	Acurácia da matriz de confusão
E17	Bancário	Extensão do RFV	K-*Means*	-	-	-
E18	Hospital	RFV	K-*Means*, SOM, *Two-step clustering*	-	Árvore de decisão	Acurácia da matriz de confusão
E19	Hospital	Extensão do RFV	K-*Means* e segmentação CLV	-	Árvore de decisão	Acurácia da matriz de confusão
E20	*E-commerce*	RFV	K-*Means*, *Fuzzy* C-*Means*, *Repetitive Median* K-*Means*	Silhueta	-	-
E21	-	RFV	K-*Means*	Cotovelo	-	-
E22	*E-commerce*	RFV	K-*Means*	Índice Davies-Bouldin	Modelo linear generalizado, análises discriminantes, rede neural, SVM e regressão	Acurácia da matriz de confusão
E23	*E-commerce*	Extensão do RFV	SOM	-	-	-
E24	*E-commerce*	Extensão do RFV	K-*Means*++	-	-	-
E25	*E-commerce*	Extensão do RFV	K-*Means*, K-*Medoids*, *Mini Batch* K-*Means*	-	-	-
E26	Vidros e cristais	Extensão do RFV	*Fuzzy Inference System*	-	-	-
E27	Seguro e telecom.	Outros	K-*Means*	Índice Davies-Bouldin	Árvore de decisão	Acurácia da matriz de confusão
E28	Companhia Aérea	Extensão do RFV	*Imperialist Competitive Algorithm*	-	Cadeias de Markov	-

FONTE: A autora (2023).

2.2 REFERENCIAL TEÓRICO

2.2.1 CRM

O termo *Relationship Management* surgiu na década de 1980 com o objetivo de melhorar o relacionamento direto com os clientes visto os benefícios gerados como a criação de valor adicional aos clientes (BERFENFELDT, 2010), o que passou, mais recentemente, a ser chamado de *Customer Relationship Management* (CRM).

O CRM pode ser definido como:

> Uma abordagem estratégica que se preocupa em criar valor para o acionista aprimorado por meio do desenvolvimento de relacionamentos apropriados com os principais clientes e segmentos de clientes. O CRM une o potencial das estratégias de *marketing* de relacionamento e TI para criar relacionamentos rentáveis e de longo prazo com os clientes e outras partes interessadas. O CRM oferece melhores oportunidades para usar dados e informações para entender os clientes e criar valor com eles. Isso requer uma integração interfuncional de processos, pessoas, operações e recursos de *marketing* que é ativada por meio de informações, tecnologia, e aplicações (PAYNE; FROW, 2005).

O CRM como uma área dentro das empresas nos dias atuais se faz crucial, afinal é uma estratégia de *marketing* chave no atingimento dos grandes objetivos de negócio. Construir relações com os clientes permite com que as empresas melhorem seus portfólios de produtos e campanhas promocionais. Sendo, portanto, uma abordagem de gestão que coloca o cliente no centro do processo e implementação da estratégia de *marketing* de uma empresa. Este conceito parte do pressuposto de que os clientes preferem ter um bom relacionamento de longo prazo com uma empresa ao invés de procurar outra empresa ou marca (PRADESYAH; SAPUTRI, 2022).

A eficácia do CRM pode ser determinada com base no conhecimento, capacidades e habilidades dos funcionários de uma organização na formação, manutenção, e fortalecimento do relacionamento com os diferentes tipos de clientes para atendê-los melhor do que concorrentes (HANAYSHA; MEHMOOD, 2022).

Dessa forma, um CRM efetivo, segundo Nimbalkar (2013), deve ser capaz de traçar estratégias para atração de novos clientes e de retenção dos clientes antigos. Destaca-se, portanto, conforme Wei *et al.* (2010), a importância de conhecer os

variados perfis de clientes, para encontrar os mais rentáveis em termos financeiros e então alocar recursos para estratégias de retenção.

A estratégia é, segundo Hanaysha e Mehmood (2022), inteiramente desenhada de forma a ajudar as organizações a identificar, obter, manter e nutrir as relações que serão mais rentáveis. O objetivo principal é prover para os clientes o máximo de valor e isso só é possível por meio do uso dos dados, para entender quem é esse cliente, suas respectivas necessidades e expectativas, de forma a co-criar esse valor.

Hanaysha e Mehmood (2022) afirmam que, tendo as ferramentas certas de CRM, uma organização estará pronta para reagir aos desafios emergentes e ter melhores habilidades para gerenciar banco de dados de clientes, utilizando-o como base para tomada de decisão e previsão do comportamento dos clientes para integrá-los na formação da futura estratégia de *marketing*. Portanto, o CRM fornece às empresas maiores percepções sobre seus clientes e permite que eles permaneçam com a empresa mais tempo a medida que o valor que os mesmos esperam é entregue por meio de comunicações apropriadas e oportunas ao longo da jornada desse cliente.

Greenberg (2001) define CRM analítico como o processo de captação, armazenagem, acesso, processamento, interpretação e transmissão de dados dos clientes, onde, a partir de um banco de dados, gera-se um histórico de consumo ou preferências, seja de produtos, canais de compra, forma de pagamento, entre outros. É a fonte da inteligência do processo e tem como elementos principais a identificação desse cliente e a posterior retenção, em outras palavras, é a divisão dos clientes em grupos menores que possuem características semelhantes a partir do que cada empresa entende ser relevante para o negócio naquele momento, onde é então possível definir estratégias de comunicação de *marketing* para cada um dos grupos de clientes formados.

Além de comunicações personalizadas, faz parte da atuação de CRM o chamado *up-sell* e *cross-sell*, sendo o primeiro uma opção de venda adicional, por exemplo, ao comprar uma batata frita pequena é oferecido a troca por uma batata frita grande por R$2 reais adicionais, já o segundo diz respeito a uma compra complementar, ou seja, ao comprar um sanduíche é oferecido um sorvete, à título de exemplo.

O indicador de *churn* também faz parte da rotina do CRM, que nada mais é que número de clientes que a empresa perdeu, ou seja, deixaram de comprar com a

marca ou que se descadastraram da plataforma. Tomando como exemplo um cliente que normalmente faz compras com um intervalo de 10 dias e está nesse momento a mais de 30 dias sem fazer uma nova compra, pode ser um indicativo de que ele entre para o indicador de *churn* em breve, sendo, portanto, responsabilidade de CRM avaliar estratégias de comunicação baseadas nas características de compras desse cliente com o objetivo de converte-lo em uma nova compra.

2.2.2 Modelo RFV

Nem todos os clientes gastam o mesmo valor com os produtos de uma empresa, alguns compram mais vezes, outros fizeram a compra recentemente, consequentemente nem todos os clientes devem ser contatados com o mesmo esforço e gastando o mesmo para tal. O processo de decisão de qual cliente contatar naquele momento pode ser baseado na clusterização da base de clientes (MIGLAUTSCH, 2000).

O modelo RFV, abreviação das palavras recência, frequência e valor, tem sido amplamente aplicado em diferentes áreas. A sua adoção permite com que os tomadores de decisão identifiquem quem são os clientes mais valiosos para, em seguida, desenvolver uma estratégia de *marketing* eficaz para esse grupo de clientes (WEI *et al.*, 2010).

Dentre as principais vantagens da aplicação do método, segundo destacam Wei *et al.* (2010), estão a facilidade de entendimento do modelo por parte dos tomadores de decisão e a efetividade de resumir o comportamento de compra de um cliente usando poucas variáveis e identificar os clientes mais rentáveis.

A recência representa o tempo desde a última compra do cliente em análise enquanto a frequência corresponde ao número de compras dentro do período de tempo especificado e o valor configura o valor em unidade monetária gasto por este mesmo cliente, neste mesmo período de tempo, ou ainda, no caso do valor, pode-se avaliar o gasto médio por compra reduzindo a colinearidade entre a frequência e o valor gasto (WEI *et al.*, 2010).

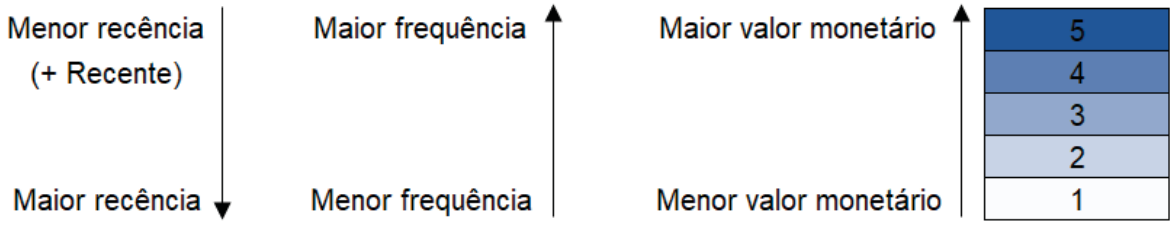
Segundo Wei *et al*. (2010), essas 3 variáveis podem ser usadas como variáveis de clusterização de clientes dado que o modelo RFV é um modelo baseado em comportamento, ou seja, ele é usado, basicamente, para analisar o comportamento dos clientes de uma empresa. Os autores ainda afirmam que a

frequência é normalmente encontrada na literatura como a variável mais importante, porém ressalta que não é algo válido para todos os tipos de empresas, cada organização possui suas especificidades tendo em vista a natureza do produto e objetivos de negócio.

Para transformar os dados transacionais dos clientes no modelo RFV é preciso dividir a lista de clientes em alguns segmentos. A quantidade é arbitrária, porém comumente encontra-se 5 divisões iguais. Para cada segmento um número, de 1 a 5, é alocado e, por exemplo, quanto mais recente a última compra, ou seja, menor a recência, maior é o número, isso porque entende-se que é um usuário com maior probabilidade de recompra, ele ainda está muito próximo à marca (WEI *et al.*, 2010). O mesmo acontece para a frequência e para o valor monetário, quanto mais compras ou maior o valor gasto por um usuário no período de tempo em análise, maior será o número aplicado. No caso da frequência, os clientes alocados no segmento 5 podem ser entendidos ou classificados como clientes mais fiéis, isto é, são clientes mais propensos a comprar os produtos da marca repetidamente. Os melhores clientes, que diz respeito a cada variável, recebem sempre a pontuação 5, enquanto os piores, 1 (MIGLAUTSCH, 2000).

A FIGURA 5 exemplifica a alocação das pontuações.

FIGURA 5 – PONTUAÇÃO DO MODELO RFV



FONTE: A autora (2022) com base em Miglautsch (2000).

Cada cliente faz parte de um único segmento do modelo RFV e é representado por um código com 3 dígitos, um representando a pontuação da frequência, outro refere-se à pontuação da variável recência e o restante é a pontuação do valor monetário, por exemplo 555 é o grupo com os melhores clientes para a empresa pois possuem alta frequência e valor monetário e uma compra recente, ao contrário do grupo 111. O número de segmentos é que define a quantidade

final de grupos de clientes, no caso de 5 segmentos, considerando as 3 variáveis, obtém-se 125 diferentes grupos, conforme Equação (1).

$$Número\ de\ grupos = Número\ de\ segmentos^{Número\ de\ variáveis}. \quad (1)$$

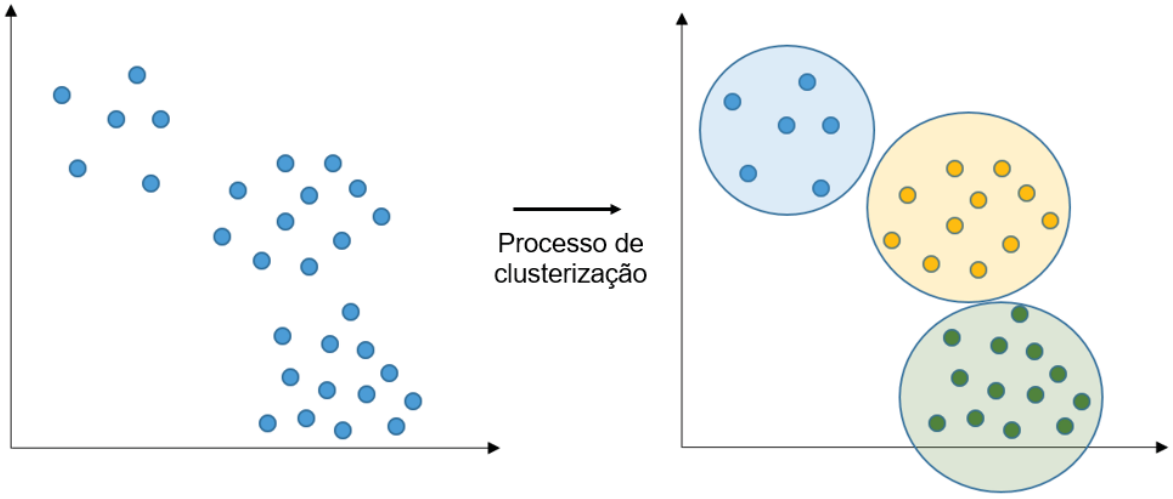
Dentre as desvantagens de aplicação do modelo, conforme cita Miglautsch (2000), tem-se os resultados insatisfatórios para grupos no topo e na base. Por ser uma divisão de grupos com a mesma quantidade de clientes, o modelo tende a agrupar comportamentos muito diferentes no topo e separar clientes com o mesmo comportamento entre o primeiro e segundo grupo. Por exemplo, se 40% da base de clientes tiver somente uma compra, então os grupos 1 e 2 terão clientes com o mesmo comportamento no que diz respeito a frequência. O autor cita ainda que não é incomum encontrar bases com até 60% dos clientes com apenas uma única compra. Enquanto no grupo de pontuação 5, as frequências podem ser muito diferentes umas das outras.

2.2.3 Clusterização

Diferentemente da segmentação onde um critério pré-definido é escolhido para caracterizar um grupo, na clusterização o algoritmo cria os grupos com base em diferentes variáveis sem que haja a necessidade de escolher uma única.

A clusterização é uma técnica de aprendizado automático que consiste em dividir um conjunto de dados, de forma não supervisionada, em grupos com objetivos semelhantes, os algoritmos não possuem uma saída pré-conhecida, ao contrário, o algoritmo se preocupa somente com os dados de entrada (ABBAS, 2008). Alguns algoritmos de clusterização necessitam como parâmetro obrigatório o número de *clusters* que se deseja formar, enquanto outros buscam detectar a quantidade de *clusters* naturais existentes no conjunto de dados de entrada. A FIGURA 6 ilustra o resultado do processo de clusterização de 30 casos de uma base de dados em 3 grupos, normalmente denominados como *clusters* ou agrupamentos.

FIGURA 6 – RESULTADO DO PROCESSO DE CLUSTERIZAÇÃO



FONTE: A autora (2022) com base em Popat *et al*. (2014).

Hruschka e Ebecken (2001) definem que, considerando um conjunto de dados de n objetos $X = \{X_1, X_2, \dots, X_n\}$, onde cada $X_i \in \mathbb{R}^p$ é um vetor de p medidas reais que dimensionam as características do objeto, estes devem ser agrupados em k *clusters* disjuntos $C = \{C_1, C_2, \dots, C_k\}$ de forma que as condições a seguir sejam respeitadas:

(i) $C_1 \cup C_2 \cup \dots \cup C_k = X$;

(ii) $C_i \neq \emptyset, \forall i,\ 1 \leq i \leq k$;

(iii) $C_i \cap C_j = \emptyset, \forall i \neq j, 1 \leq i \leq k, 1 \leq j \leq k$.

Dessa forma, um objeto não pode pertencer a mais de um *cluster* e cada *cluster* deve ter ao menos um objeto. Hruschka e Ebecken (2001) citam ainda que o valor de k geralmente é desconhecido e encontrar o melhor agrupamento para o conjunto de dados X é NP-completo e não é computacionalmente possível encontrá-lo, a não ser que n e k sejam extremamente pequenos, dado que o número de partições distintas em que pode-se dividir os n objetos em k *clusters* aumenta, conforme Equação (2).

$$Número\ de\ partições\ distintas = \frac{k^n}{n!}. \tag{2}$$

Popat *et al.* (2014) afirmam que a clusterização é uma das técnicas de mineração mais desafiadoras no processo de descoberta dos dados. Agrupar uma grande quantidade de dados é uma tarefa difícil, pois o objetivo é encontrar uma partição adequada, sem nenhum conhecimento prévio dos resultados, tentando maximizar a similaridade intra *cluster* e minimizar a similaridade inter *cluster.* Os autores resumem como sendo uma técnica de processamento de dados que resulta em grupos significativos para posterior análise, como análises estatísticas, por exemplo.

O processo para obtenção dos grupos com alguma similaridade passa antes pela etapa de pré-processamento onde a base de dados pura é: i) Limpa, ou seja, dados faltantes são retirados ou preenchidos de alguma forma; ii) As variáveis podem ser reduzidas por meio de algum método; iii) Assim como pode haver a retirada ou tratamento de *outliers* e iv) Ainda há a etapa de transformação dos dados caso faça sentido para a análise em questão. Essa nova base tratada vira um *input* para o algoritmo selecionado de clusterização que definirá quais são os grupos semelhantes como *output* do processo (POPAT *et al.*, 2014).

No processo de clusterização em si alguns algoritmos armazenam na memória principal os dados, para tal utilizam as seguintes estruturas conforme explica Han e Kamber (2001):

(i) Matriz de dados: Cada linha representa um objeto n a ser clusterizado, enquanto as colunas são as características p de cada objeto, resultando, portanto, em uma matriz n x p, conforme Equação (3).

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix}. \qquad (3)$$

(ii) Matriz de dissimilaridade: Os elementos da matriz são distâncias ou dissimilaridade entre pares de objetos i e j, resultando em uma matriz quadrada $D_{n\,x\,n}$, conforme Equação (4). Quanto menor for o valor de $d(i,j)$, mais semelhantes são os objetos de acordo com as características usadas, dessa forma, eles tendem a ficar no mesmo *cluster*. Os algoritmos de clusterização que fazem uso desse tipo de

matriz, recebem primeiro uma matriz de dados, posteriormente a transformam em uma matriz de dissimilaridade para então iniciar as etapas do método.

$$D = \begin{bmatrix} 0 & \cdots & d(1,n) \\ \vdots & \ddots & \vdots \\ d(n,1) & \cdots & 0 \end{bmatrix}. \quad (4)$$

Dentre as principais medidas de similaridade, segundo Han e Kamber (2001), tem-se: Euclidiana – conforme Equação (5), Manhattan, Minkowski e Mahalanobis. Sendo a primeira delas a mais utilizada, e pode ser definida como a distância em linha direta entre dois pontos que representam os objetos.

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \cdots + |x_{ip} - x_{jp}|^2}. \quad (5)$$

Outra distância utilizada é a Manhattan, conforme Equação (6), que também pode ser chamada de *city-block*, onde a distância é a soma dos módulos das diferenças entre todos os atributos dos dois objetos em questão, é mais fácil de ser calculada, porém a qualidade do resultado pode ser inferior se os atributos estão correlacionados (HAN; KAMBER, 2001).

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|. \quad (6)$$

Já a distância Minkowski, conforme Equação (7), é a generalização das equações apresentadas anteriormente, onde o q é um número inteiro positivo. No caso da distância Euclidiana o q é igual a 2, enquanto na Manhattan é equivalente a 1. Pode-se atribuir também pesos w a cada um dos atributos quando se deseja dar importância diferente a cada uma das características dos objetos (HAN; KAMBER, 2001).

$$d(i,j) = (w_1|x_{i1} - x_{j1}|^q + w_2|x_{i2} - x_{j2}|^q + \cdots + w_p|x_{ip} - x_{jp}|^q)^{\frac{1}{q}}. \quad (7)$$

Por fim, conforme Equação (8), a distância de Mahalanobis tende a formar *clusters* hiperelípticos e utiliza a matriz S de covariância entre os grupos, se for igual a matriz identidade, a distância de Mahalanobis é equivalente a distância Euclidiana, porém se a matriz é diagonal, então o resultado equivale a distância Euclidiana normalizada (HAN; KAMBER, 2001).

$$d(i,j) = (x_i - x_j)^T S^{-1}(x_i - x_j). \quad (8)$$

Segundo Han e Kamber (2001), um método de clusterização deve atender às seguintes condições:

i) Encontrar *clusters* com forma arbitrária – Ao considerar o espaço Euclidiano, os grupos podem ser esféricos, lineares, alongados, elípticos, cilíndricos, espirais, etc.;
ii) Encontrar *clusters* de diferentes tamanhos;
iii) Ser capaz de trabalhar com diferentes variáveis;
iv) Ser insensível a ordem de apresentação dos objetos;
v) Ser capaz de trabalhar com inúmeros atributos;
vi) Ser escalável para lidar com uma quantidade infinita de objetos;
vii) Oferecer resultados interpretáveis, compreensíveis e utilizáveis;
viii) Ser insensível na presença de ruídos visto que dados reais contém ruídos e a qualidade no resultado obtido não pode ser reduzida;
ix) Exigir o mínimo de parâmetros de entrada possível, dado que são, muitas vezes, difíceis de determinar e alguns métodos são sensíveis a tais parâmetros;
x) Aceitar restrições, uma vez que, para determinadas aplicações, podem ser de extrema importância;
xi) Descobrir o número ideal de *clusters*.

Nenhum método atende a todos esses critérios, entretanto, alguns métodos podem ser mais indicados que outros a depender da quantidade de objetos, da exigência de ter um número pré-determinado de *clusters* ou ainda do tamanho de cada *cluster*, à título de exemplo.

A efetividade dos algoritmos é algo variável, tendo em vista que grande parte requer parâmetros de inicialização que são difíceis de serem determinados,

principalmente em base de dados reais e que tendem a ser cada vez maiores. Os algoritmos são extremamente sensíveis a esses valores, ou seja, para valores muito próximos de um parâmetro os resultados do agrupamento realizado podem ser significativamente diferentes (ANKERST *et al.*, 1999).

A primeira publicação que registra o uso de um método de clusterização é de 1948, com a aplicação de um algoritmo hierárquico (SORENSON, 1948). Desde então diferentes algoritmos vêm sendo aplicados de forma a obter os grupos com similaridades e, cada um deles, resolve o problema sob diferentes perspectivas. Dentre as principais categorias pode-se citar: (i) Algoritmos de agrupamento particionais; (ii) Algoritmos de agrupamento hierárquicos e (iii) Algoritmos de agrupamento baseado em modelos. Há ainda outras categorias como baseados em grafos, grade, lógica *fuzzy*, etc.

2.2.3.1 Agrupamento particional

O agrupamento particional, também conhecido como grupo dos algoritmos de realocação iterativo, é a classe de algoritmos de clusterização considerada a mais popular, afirmam Popat *et al.* (2014). O objetivo, segundo os autores, é minimizar um determinado critério de agrupamento, a partir de uma função objetivo, ao realocar iterativamente os pontos entre os *clusters* de forma a alcançar o particionamento ideal. Esse critério, que busca-se minimizar, pode ser uma função de dissimilaridade baseada na distância, dessa forma, objetos que estão no mesmo *cluster* são semelhantes enquanto objetos em *clusters* diferentes são dissimilares.

Ao iniciar, o algoritmo escolhe k objetos como sendo os centros dos k *clusters,* posteriormente divide os demais objetos nesses grupos de forma a minimizar a distância entre o objeto em questão e o centro do *cluster*. A escolha do centro, referência para o cálculo da dissimilaridade, se dá pela média dos objetos que pertencem ao *cluster*, também conhecida por gravidade, ou define-se como referência o objeto mais próximo ao centro de gravidade do *cluster*, também chamado de medoide. A partir de então, iterações são realizadas onde os objetos mudam de *clusters* até a otimização da função objetivo (POPAT *et al.,* 2014).

A função objetivo mais usada nos métodos particionais é o erro quadrático, conforme Equação (9), onde E é a soma do erro quadrático para todos os objetos, p é

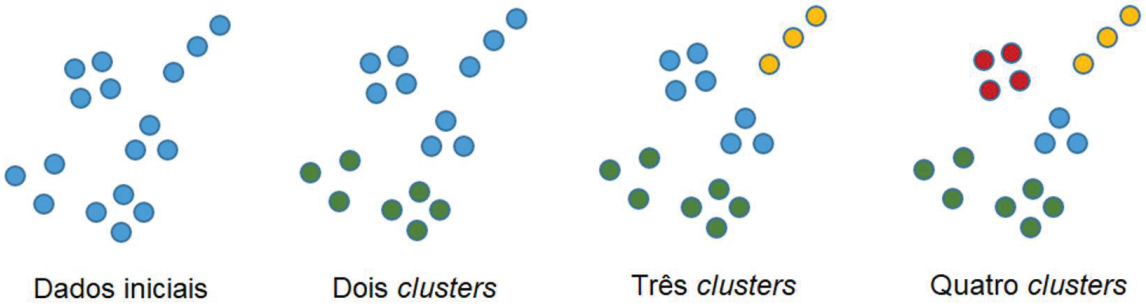
a dimensão do espaço dos objetos, e m_i é o representante do *cluster* C_i (ESTER *et al.*, 1998).

$$E = \sum_{j=1}^{k} \sum_{x \in C_i} \| p - m_i \|^2 , para\ k \in (1, n). \quad (9)$$

Guha *et al*. (1998) afirmam que os algoritmos da classe particional procuram agrupar os k *clusters* o mais compacto, denso e separado possível, porém são sensíveis quando há grande diferença nos tamanhos e geometrias dos grupos.

Os algoritmos presentes nessa classe são úteis quando a aplicação necessita um número k fixo de *clusters* (FIGURA 7), sendo uma das desvantagens, e funcionam bem com *clusters* em forma esférica. Dentre os principais algoritmos, Popat *et al.* (2014) citam o K-*Means,* PAM ou K-Medoides e CLARA (Agrupamento para grandes aplicações).

FIGURA 7 – AGRUPAMENTO POR CLUSTERIZAÇÃO



FONTE: A autora (2022) com base em Oliveira (2008).

2.2.3.1.1 K-*Means*

O *K-Means* é um algoritmo não hierárquico do tipo não supervisionado, ou seja, apenas os dados são necessários para sua execução, sem a necessidade de ter um conhecimento prévio da classe à qual cada observação pertence. Segundo Palma (2018), o método foi proposto por S. Lloyd em 1957, entretanto o trabalho só foi publicado no ano de 1982.

O algoritmo se utiliza de uma técnica iterativa para o particionamento dos dados de forma a encontrar similaridades entre os n objetos, onde, a partir da definição inicial do número de *clusters*, busca-se minimizar a distância quadrática total entre cada ponto e o centroide mais próximo, sendo o centroide o centro de cada *cluster,* equivalente à média dos valores do *cluster* (ARTHUR*;* VASSILVITSKII, 2007). A definição do número de *clusters* é a etapa mais complexa de aplicação do método pois na grande maioria das vezes não há um conhecimento sobre a base de dados de entrada.

Dado um conjunto de dados $X = \{x_1, x_{2,} \dots, x_n\}$, onde $x_n \in \mathbb{R}^p$, segundo Likas *et al.* (2003), o objetivo é dividir todos os n objetos em k grupos, conjunto de *clusters* $C = \{C_1, C_2, \dots, C_k\}$, até o ponto ideal. O critério mais utilizado para tal é o mínimo da soma dos quadrados das distâncias Euclidianas d_E do objeto x_i até o centroide c_j, também chamado de centro do *cluster*, do grupo C_j que contém x_i.

Segundo Arthur e Vassilvitskii (2007), o método pode ser descrito em 3 passos, sendo o primeiro conhecido por inicialização onde, de forma aleatória, k centroides c_j são formados, a partir do número k de *clusters* parametrizado.

Já a atribuição do *cluster* é a segunda etapa, onde, a partir dos centroides c_j gerados previamente, as distâncias $d_E(c_j, x_i)^2$ são calculadas em todos os pontos x_i do conjunto de dados, para então cada ponto ser atribuído a um centroide c_j onde a distância seja a menor possível, conforme Equação (10). Ao término da segunda fase, os n objetos já estão agrupados conforme o número k de *clusters* pré-estabelecido (ARTHUR; VASSILVITSKII, 2007).

$$\arg\ min \sum_{j=1}^{k} \sum_{i=1}^{n} \left|\left|x_i - c_j\right|\right|^2, c_j \in C. \tag{10}$$

Por fim, a terceira fase é chamada de movimentação dos centroides pois, dado que os objetos foram agrupados conforme a distância, se faz necessário um recálculo dos valores dos centroides a partir da média dos valores dos pontos de cada *cluster,* conforme Equação (11) (ARTHUR; VASSILVITSKII, 2007).

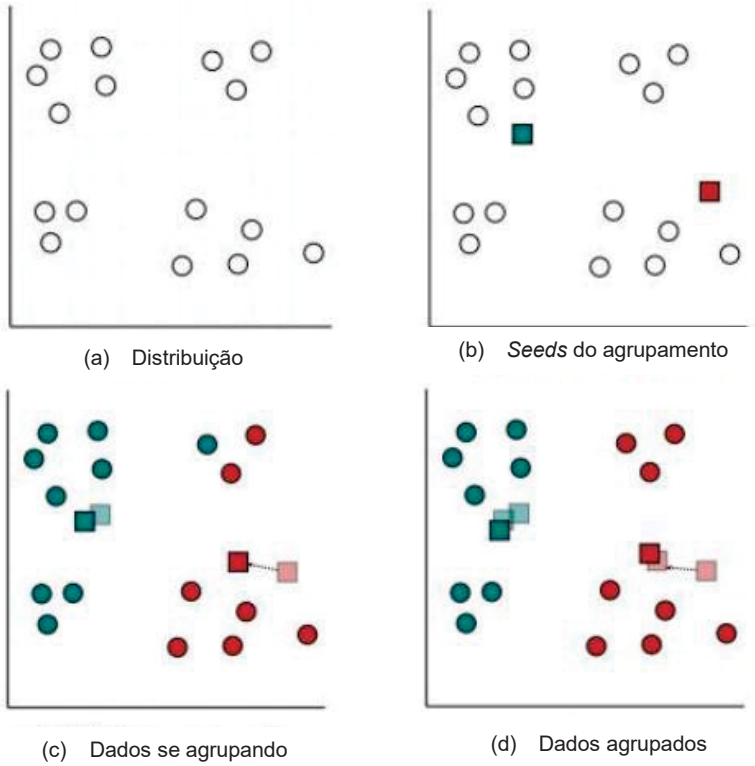
$$c_j = \frac{1}{|S_j|} \sum_{x_j \in S_j} x_j. \tag{11}$$

Onde:

S_j é o conjunto de todos os objetos atribuídos ao *cluster* C_j.

Arthur e Vassilvitskii (2007) ainda explicam que as etapas 2 e 3 são repetidas até que o *cluster* se torne estático ou que algum critério de parada que tenha sido pré-estabelecido seja alcançado, como o número máximo de iterações à título de exemplo. A FIGURA 8 ilustra o funcionamento do algoritmo.

FIGURA 8 – FUNCIONAMENTO DO ALGORITMO K-*MEANS*



FONTE: Prado (2008).

O *K-Means* é uma das técnicas de clusterização mais amplamente difundidas, devido à sua simplicidade matemática, fácil implementação e flexibilidade, Likas *et al.* (2003) afirmam ainda que é um algoritmo iterativo rápido – $O(nkt)$, onde n é o número total de padrões, k é o número de *clusters* e t é o número de iterações. A complexidade por iteração é linear ao tamanho do conjunto de dados, ao número de grupos e à dimensionalidade dos dados, sendo necessário $n \times k$ cálculos a cada iteração t do algoritmo. Porém Arthur e Vassilvitskii (2007) alertam que o algoritmo é sensível ao número k de *clusters* pré-estabelecidos, que afetam na escolha dos

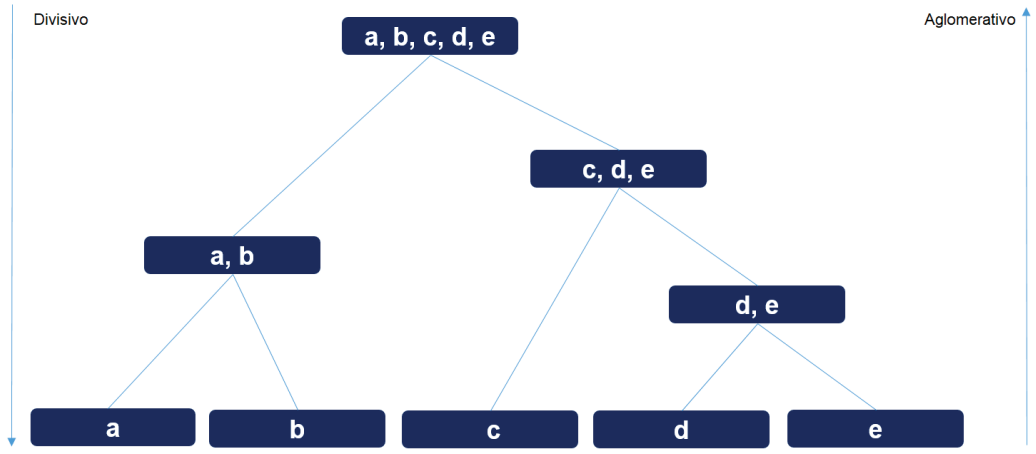
centroides iniciais, e ressaltam a importância da normalização prévia do conjunto de dados para que nenhum objeto se sobressaia frente aos demais o que pode afetar negativamente os resultados obtidos por meio da aplicação do método.

2.2.3.2 Agrupamento hierárquico

Ao contrário dos métodos particionais, que produzem agrupamentos simples, os métodos hierárquicos produzem uma série de agrupamentos relacionados. Segundo Popat *et al.* (2014), o agrupamento hierárquico divide ou mescla um conjunto de dados em uma sequência de partições aninhadas, podendo ser aglomerativo, de baixo para cima, ou divisivo, de cima para baixo. Ambos os métodos apresentam os *clusters* na forma de um dendrograma.

No caso do agrupamento hierárquico aglomerativo, conforme explicam Popat *et al.* (2014), a clusterização começa com cada objeto em um *cluster* e então continua agrupando os pares de *clusters* mais semelhantes, a partir da distância entre cada par armazenada em uma matriz de dissimilaridade simétrica, até que reste um único *cluster* que contenha todos os objetos do conjunto de dados. Já o agrupamento hierárquico divisivo tem um funcionamento contrário ao aglomerativo, começando com todos os objetos em um único *cluster* para posteriormente dividi-los em *clusters* menores até que cada objeto seja um *cluster.* A FIGURA 9 apresenta uma ilustração de como o método funciona.

FIGURA 9 – AGRUPAMENTO HIERÁRQUICO



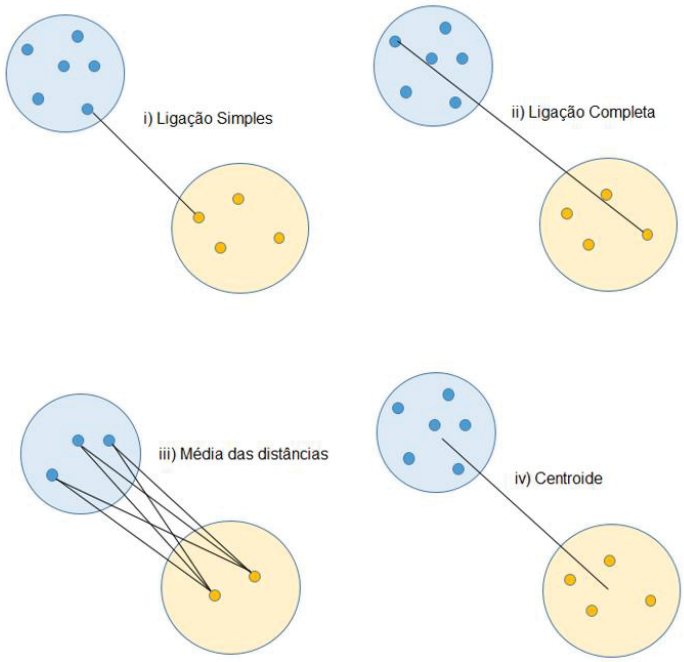
FONTE: A autora (2022) com base em Popat *et al*. (2014).

O agrupamento ou divisão dos *clusters* é feita a partir de uma medida de similaridade que reflete o grau de proximidade ou separação dos objetos. Definir uma medida de similaridade apropriada, especialmente para alguns algoritmos de clusterização, é crucial (POPAT *et al*., 2014).

O algoritmo de clusterização de ligação simples, também conhecido por método do vizinho mais próximo segundo Popat *et al*. (2014), considera que a distância entre 2 *clusters* deve ser igual a menor distância de qualquer membro de um *cluster* até qualquer outro objeto de outro *cluster*. Já o algoritmo de ligação completa, exemplificam os autores, é chamado de método do vizinho mais distante pois a distância entre 2 *clusters* deve ser a maior distância de qualquer objeto de um *cluster* até qualquer objeto de outro *cluster*.

O método da média das distâncias considera que a distância entre 2 *clusters* deve ser igual a distância média de qualquer objeto de um *cluster* até qualquer objeto de outro *cluster*. No algoritmo do centroide, a medida de similaridade é definida pela distância entre os pontos médios de cada *cluster* (centroide). Por fim, a medida de distância do método Ward é a soma das distâncias ao quadrado entre dois *clusters* (POPAT *et al.,* 2014). A FIGURA 10 ilustra algumas das diferentes medidas de similaridade existentes.

FIGURA 10 – MEDIDAS DE SIMILARIDADE DO AGRUPAMENTO HIERÁRQUICO



FONTE: A autora (2022) com base em Lauretto (2017).

O agrupamento hierárquico também pode ser entendido, segundo explicam Popat *et al.* (2014), como uma árvore binária onde as raízes representam todos os objetos do conjunto de dados a serem clusterizados. Os autores esclarecem que a vantagem dessa classe de métodos é permitir cortar a hierarquia no nível desejado de acordo com o número k de *clusters* desejado, o que o faz diferente de qualquer outra classe de algoritmos. Outra vantagem é a redução do efeito dos *clusters* iniciais na execução do método.

A complexidade da clusterização aglomerativa é de $O(n^3)$ o que o faz lento demais para bases de dados extremamente grandes, já a clusterização hierárquica divisiva é de $O(2^n)$ o que é ainda mais lento, dessa forma, Popat *et al.* (2014) concluem que os algoritmos hierárquicos aglomerativos são melhores, em termos da velocidade de execução, que os divisivos e são mais utilizados na prática.

Alguns exemplos de algoritmos dessa classe de clusterização, citados por Popat *et al.* (2014), são ROCK (Clusterização Robusta usando Ligações), BIRCH (Redução Iterativa Balanceada e Clusterização usando Hierarquias) e CURE (Clusterização Usando Representantes).

2.2.3.2.1 Método de Ward

O método de Ward, também conhecido como método da mínima variância, é um algoritmo de clusterização hierárquico alternativo ao de ligação simples onde, segundo Hair *et al.* (2005), ao invés de medir a distância diretamente, usa como medida de similaridade a variância dos *clusters,* não somente entre eles, mas dentro deles também, por meio da soma dos erros ao quadrado entre dois grupos. Ele tende a resultar em grupos com tamanhos aproximadamente iguais devido a busca pela minimização da variação interna e é mais indicado para variáveis quantitativas.

Assim como os demais algoritmos da classe hierárquica, é um método que possui uma complexidade computacional elevada, porém quando comparado aos demais é o que possui menos iterações, isso se torna uma vantagem e razão pela qual foi escolhido para aplicação, entretanto esse fator pode afetar a qualidade da clusterização realizada (SZMRECSANYI, 2012).

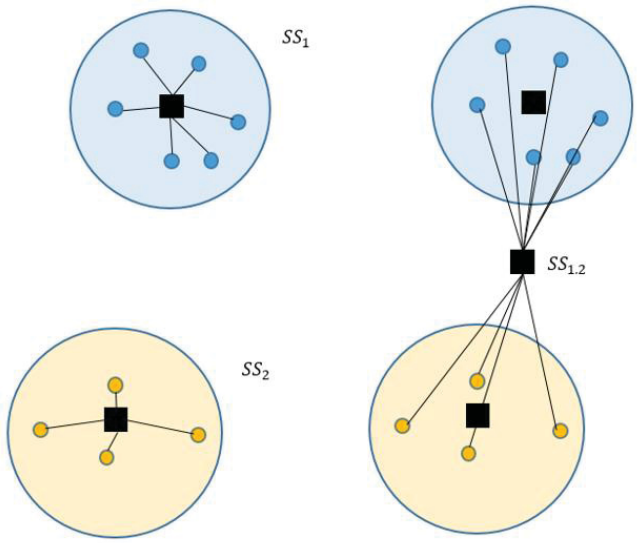
Como os outros métodos hierárquicos aglomerativos, o método Ward se inicia com n *clusters*, ou seja, um para cada objeto do conjunto de dados que vão sendo agrupados ao longo da execução do algoritmo até que reste um único *cluster* que contenha todos os n objetos. A cada etapa, os *clusters* que apresentam o menor aumento na soma total dos erros ao quadrado, isto é, a menor variância, são fundidos (HAIR *et al*., 2005).

A soma total dos erros ao quadrado SST é também chamado do custo de fusão ao combinar *clusters* l e i, conforme Equação (12), sendo sendo n_i e n_l o número de objetos no *cluster* C_i e C_l, $SS_{l,i}$ a soma dos quadrados dos erros do grupo formado pelos *clusters* l e i, SS_l e SS_i a soma dos erros quadrados dentro do respectivo *cluster* e $\bar{x}_{jl}$ *e* $\bar{x}_{ji}$ os centróides de cada um dos *clusters*. O erro começa sempre em 0 tendo em vista que na etapa inicial cada objeto é um *cluster* e cresce à medida que os grupos vão se juntando, sendo o objetivo geral do método mantê-lo o menor possível. Dado dois pares de *clusters* cujos centros estão igualmente afastados, o método preferirá fundir os menores grupos pois o número de objetos faz parte da equação assim como a separação geométrica (DUTRA *et al.,* 2004).

$$SST_{l,i} = SS_{l,i} - (SS_l + SS_i) = \frac{n_l n_i}{n_l + n_i} \sum_{j=1}^{n_i} (\bar{x}_{jl} - \bar{x}_{ji})^2. \quad (12)$$

A FIGURA 11 ilustra o funcionamento do método de Ward.

FIGURA 11 – MÉTODO DE WARD



FONTE: A autora (2022) com base em Lauretto (2017).

O método de Ward não exige como parâmetro inicial o número de *clusters*, porém é possível identificar o número de *clusters* ideal por meio do custo de fusão. Se o custo cresce muito é provável que o agrupamento está perdendo estrutura, dessa forma recomenda-se usar o k anterior ao k que provocou o aumento excessivo do custo, porém o quão grande é o aumento do custo que o torne inaceitável é subjetivo, não há nenhuma regra para tal definição (REGONDA *et al*., 2016).

2.2.3.3 Agrupamento baseado em modelos

A clusterização baseada em modelos é realizada por algoritmos que possuem como referência para cada grupo um certo modelo, ou seja, otimiza-se a curva de um modelo matemático aos objetos de um conjunto de dados em estudo. Para a definição dos grupos geralmente uma função densidade é elaborada de forma a refletir a distribuição espacial dos objetos, conforme Equação (13), onde f_k é a função densidade de probabilidade das observações x no grupo k e T_k é a probabilidade que uma observação seja proveniente do k-ésimo componente da mistura (MADHULATHA, 2012).

$$f(x) = \sum_{k=1}^{G} T_k f_k(x). \qquad (13)$$

2.2.3.3.1 Modelo de Misturas Gaussianas

Os modelos de mistura são distribuições formadas pela junção de mais de uma distribuição de probabilidade D, conhecida também como componentes da mistura, com probabilidade w, chamados de pesos da mistura. A densidade final é igual às densidades ponderadas que caracterizam essas distribuições (PORTELA, 2015).

No caso dos agrupamentos por meio de modelos de mistura, a medida de similaridade é a probabilidade *a posteriori* dos dados em relação às classes e o objetivo é a maximização do critério de agrupamento da função de verossimilhança, onde, segundo Portela (2015), utiliza-se o algoritmo *Expectation Maximization* para estimar os parâmetros que maximizam a verossimilhança dos dados.

A distribuição gaussiana, também é conhecida como distribuição normal e é uma distribuição de probabilidade contínua, descrita pela média e pelo desvio padrão. Entretanto, uma única distribuição gaussiana não pode modelar múltiplas regiões com densidades dentro de um conjunto de dados multimodal que é encontrado na prática (PATEL, KUSHWAHA, 2020).

Dessa forma, o algoritmo de clusterização se baseia, segundo Patel e Kushwaha (2020), na ideia de que cada grupo no espaço de atributos pode ser representado por uma função densidade de probabilidade, sendo descrito estatisticamente por uma soma de modelos probabilísticos, onde cada modelo descreve um *cluster.* Isto é, a soma ponderada das densidades gaussianas é conhecida como modelo de mistura de distribuições gaussianas e se baseia na Estimativa de Máxima Verossimilhança (PATEL, KUSHWAHA, 2020).

Segundo Patel e Kushwaha (2020), o algoritmo consiste em matrizes de covariância, pesos de mistura e vetores médios de cada densidade de componente presente, sendo capaz de modelar as correlações por meio da combinação linear da base de covariância diagonal.

É um algoritmo iterativo que melhora os resultados a cada iteração e forma *clusters* com forma elipsoidal. Cada *cluster* é modelado como uma distribuição gaussiana (PATEL, KUSHWAHA, 2020).

O Modelo de Misturas Gaussianas (GMM) é, portanto, uma soma de funções gaussianas com vetores de média μ_i e uma matriz de covariância Σ_i. A função gaussiana é então representada pela Equação (14), onde $g(x|w_i, \Sigma_i)$ são as

densidades das componentes gaussianas, x é o vetor de características de dimensão D, M é o número de componentes Gaussianas e os pesos da mistura são w_i. O objetivo é encontrar um conjunto de parâmetros μ_i, w_i e Σ_i para cada i em que a distribuição analítica corresponda de forma ideal à distribuição empírica. Cada uma das componentes é representada pela Equação (15), onde D é a dimensionalidade (SILVA, 2014).

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i). \tag{14}$$

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x-\mu_i)'\Sigma^{-1}(x-\mu_i)\}. \tag{15}$$

Onde busca-se a maximização da verossimilhança dos dados de treinamento, conforme Equação (16), onde T é a quantidade de observações de treino (SILVA, 2014).

$$\begin{aligned} \lambda &= arg\max_{\lambda} p(X|\lambda). \\ p(X|\lambda) &= \prod_{t=1}^{T} p(x_t|\lambda). \end{aligned} \tag{16}$$

Para tal, utiliza-se o *Expectation-Maximization* (EM), que atualiza os valores do GMM a cada iteração, sendo composto por duas etapas: i) *Expectation* e ii) *Maximization*. Na primeira etapa, se calcula a verossimilhança entre o modelo atual e os dados de treino, conforme Equação (17), onde, $g(x_t|\mu_i, \Sigma_i)$ é uma função Gaussiana D-Dimensional e para cada objeto de treino t é atribuído uma probabilidade $p(i|x_t, \lambda)$ de estar em cada *cluster* com base em sua localização x_t. Já a segunda etapa altera o modelo atual para que possua uma maior correlação com os dados, sendo estimado de acordo com as Equações (18, 19 e 20). As Equações (18) e (19) determinam a média e a matriz de covariância do agrupamento i, respectivamente; as contribuições de cada objeto para esses valores são ponderadas pela probabilidade desse objeto estar no *cluster*, que foi calculado no passo i), já a Equação (20) calcula o peso de um determinado cluster i, isso é realizado tomando a probabilidade de um objeto pertencer ao cluster i sobre todos os demais objetos. Os dois passos são repetidos até a convergência, resultando em uma estivada de cada parâmetro para cada *cluster* (SILVA, 2014).

$$p(i|x_t, \lambda) = \frac{w_i g(x_t|\mu_i, \Sigma_i)}{\sum_{i=1}^{M} w_i g(x_t|\mu_i, \Sigma_i)}. \quad (17)$$

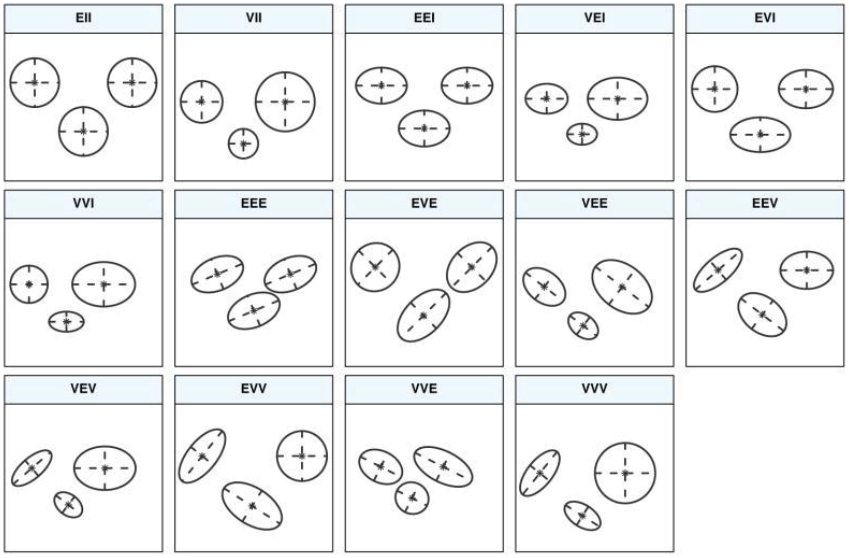
$$\mu_i = \frac{\sum_{t=1}^{T} \mathrm{p}(\, i|x_t, \lambda) x_t}{\sum_{t=1}^{T} \mathrm{p}(\, i|x_t, \lambda)}. \quad (18)$$

$$\Sigma_i = \frac{\sum_{t=1}^{T} \mathrm{p}(i|x_t, \lambda) x_t^2}{\sum_{t=1}^{T} \mathrm{p}(i|x_t, \lambda)} - \mu_i^2. \quad (19)$$

$$w_i = \frac{1}{T} \sum_{t=1}^{T} \mathrm{p}(i|x_t, \lambda). \quad (20)$$

Em uma distribuição gaussiana multivariada, o volume, forma e orientação das covariâncias podem ser iguais ou variáveis entre os grupos, dessa forma, surgem 14 modelos com diferentes características geométricas, que são apresentados na FIGURA 12 (SCRUCCA *et al.,* 2016). Sendo assim, conforme FIGURA 12 e FIGURA 13, é possível observar que o modelo VEV (V*ariable volume, equal shape, variable orientation*), a título de exemplo, possui distribuição elipsoidal, volume e orientação variáveis e, além disso, mesmo formato. O Modelo de Mistura Gaussiana (GMM), assume uma distribuição Gaussiana multivariada para cada componente. Assim, os k *clusters* são elipsoidais, centrados no vetor médio μ_k, e com outras características geométricas, como volume, forma e orientação, determinadas pela matriz de covariância Σ_k.

FIGURA 12 – MODELOS GAUSSIANOS COM DIFERENTES CARACTERÍSTICAS GEOMÉTRICAS



FONTE: Scrucca *et al.* (2016).

FIGURA 13 – CARACTERÍSTICA DOS MODELOS GAUSSIANOS

Simbolo	Distribuição	Volume	Forma	Orientação
EII	Esférico	Igual	Igual	NA
VII	Esférico	Variável	Igual	NA
EEI	Diagonal	Igual	Igual	Eixos coordenados
VEI	Diagonal	Variável	Igual	Eixos coordenados
EVI	Diagonal	Igual	Variável	Eixos coordenados
VVI	Diagonal	Variável	Variável	Eixos coordenados
EEE	Elipsoidal	Igual	Igual	Igual
EEV	Elipsoidal	Igual	Igual	Variável
VEV	Elipsoidal	Variável	Igual	Variável
VVV	Elipsoidal	Variável	Variável	Variável

FONTE: Adaptado de Rodrigues (2009).

2.2.3.4 Definição do número ideal de *clusters*

Os algoritmos de clusterização, como o K-*Means* à título de exemplo, frequentemente necessitam de alguns parâmetros para sua aplicação, dentre eles o número de *clusters*.

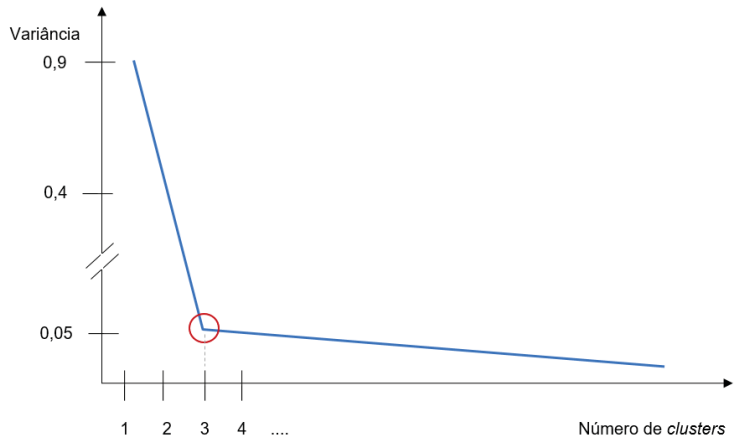
A definição do número k de *clusters* é de extrema importância para o resultado dos algoritmos a serem executados, porém é um dos parâmetros mais difíceis de ser determinado tendo em vista que não há um único método possível de ser aplicado, ou seja, não há uma única solução (BRENTARI *et al.*, 2016).

2.2.3.4.1 Método do cotovelo

O método do cotovelo, do inglês *Elbow Method*, é um método, dentre os vários existentes e que será aplicado no presente estudo, que se propõe a auxiliar em uma boa escolha do número de *clusters*, segundo Bholowalia e Kumar (2014), a partir da análise da porcentagem de variância explicada em função do número de *clusters,* ou seja, ao ter um *cluster* a mais do que o método indica como ideal os resultados obtidos com a clusterização não devem ser significativamente melhores. Os autores explanam que o percentual da variância explicada pelo número de *clusters* é plotada em um gráfico onde no eixo x são representadas as quantidades de grupos e no eixo y a variância correspondente.

No início, ao aumentar a quantidade *clusters,* o ganho em redução da variância é significativo, porém em um dado momento o ganho marginal cai drasticamente o que representa um ângulo no gráfico, como se fosse um cotovelo, o que dá origem ao nome método do cotovelo. O número de *clusters* correspondente a esse ponto ou o que possui o maior decrescimento na variância é considerado o ideal, a partir dele a variância tende a se estabilizar (BHOLOWALIA; KUMAR, 2014). A FIGURA 14 ilustra a interpretação do resultado do método.

FIGURA 14 – INTERPRETAÇÃO DO MÉTODO DO COTOVELO



FONTE: A autora (2022) com base em Bholowalia e Kumar (2014).

Segundo Cui (2020), o método do cotovelo é adequado para valores de k relativamente pequenos. O mesmo calcula a diferença quadrada de diferentes valores de k, à medida que o número de k aumenta o grau de distorção se torna menor pois o número de objetos contidos em cada grupo diminui, ficando cada vez mais próximos ao centro de gravidade do *cluster*.

A variável utilizada pelo método onde busca-se a inflexão é a variação intra-*cluster* minimizada pela soma dos quadrados, do inglês *Within-Cluster Sum-of-Squares* (WCSS), conforme Equação (21), onde k é o número de *clusters*, C_k é o centroide do *cluster* k, i varia de 1 até n, sendo n o número de objetos do conjunto de dados em análise e a distância calculada é em relação aos elementos de um *cluster* até o seu respectivo centroide. Quanto melhor a clusterização, menor o WCSS total (CUI, 2020).

$$WCSS = \sum_{x_i \in C_k} dist(x_i, c_1)^2 + \cdots + \sum_{x_i \in C_k} dist(x_i, c_k)^2. \quad (21)$$

2.2.3.5 Avaliação de desempenho

A avaliação da qualidade do agrupamento realizado pelo algoritmo pode ser baseada, segundo Palacio-Nino e Berzal (2019), na coesão, separação ou uma mistura de ambos. A coesão é uma medida intra-*cluster,* enquanto a separação é uma medida inter-*cluster.* Ambos são embasados em uma função de proximidade que determina quão similar um par de objetos é, onde as funções de similaridade, dissimilaridade e distância podem ser usadas.

Palacio-Nino e Berzal (2019) citam algumas métricas que quantificam a coesão e separação dos agrupamentos, são eles:

i) O coeficiente Calinski-Harabasz (CH), conforme Equação (22), é um deles, e se baseia na dispersão interna dos *clusters* e da dispersão entre os mesmos, sendo k o número de *clusters* e n a quantidade de objetos. Onde SSW, conforme Equação (23), é a soma dos erros ao quadrado intra-*cluster*, sendo x_i um objeto no *cluster* k e c_k o centroide. E SSB, conforme Equação (24), que é a soma dos erros ao quadrado entre *clusters,* onde busca-se maximizar a distância entre os *clusters* (GUERREIRO, 2021).

$$CH = {(SSB/(k-1))}\Big/{(SSW/(n-k))}. \quad (22)$$

$$SSW = \sum_{k=1}^{K}\sum_{i=1}^{n} dist(x_i - c_k). \quad (23)$$

$$SSB = \frac{1}{2}\sum_{k=1,l=1,l\neq k}^{K} dist(c_l - c_k). \quad (24)$$

ii) O Índice Dunn (D), conforme Equação (25), é a razão entre a menor e a maior distância entre pontos de diferentes *clusters* que deve ser maximizada.

$$D = \min_{1<i<k}\left\{\min_{1<j<k,\neq j}\left\{\frac{\delta(C_i,C_j)}{\max_{1<l<k}\{\Delta(C_l)\}}\right\}\right\}. \quad (25)$$

$$\Delta(C_i) = \max_{x,y\in C_i}\{dist(x,y)\}.$$

$$\delta(C_i, C_j) = \min_{x\epsilon C_i, y\epsilon C_j}\{dist(x,y)\}.$$

iii) O Índice Ball-Hall (BH), conforme Equação (26), é uma medida de dispersão baseado nas distâncias ao quadrado dos pontos do *cluster* em relação ao respectivo centroide. Onde o SSE, conforme Equação (27), eleva ao quadrado o resultado da equação somatória, ao contrário do que é realizado no SSW (Equação 23).

$$BH = \frac{SSE}{k}. \quad (26)$$

$$SSE = \sum_{k=1}^{K}\sum_{i=1}^{n} dist(x_i - c_k)^{\,2}. \quad (27)$$

iv) O Índice Hartigan (H), conforme Equação (28), é baseado na relação logarítmica entre a soma dos quadrados intra e entre *cluster*.

$$H = \log\left(\frac{SSB}{SSE}\right). \quad (28)$$

v) O coeficiente de Xu, conforme Equação (29), leva em consideração a dimensionalidade D dos dados, o número n de objetos e a soma dos erros ao quadrado SSE dos k *clusters.*

$$Xu = D\log_2\left(\sqrt{\frac{SSE}{Dn^2}}\right) + \log k. \quad (29)$$

vi) O Índice Davies-Bouldin mede a similaridade média entre os grupos formados e a dissimilaridade média entre os grupos adjacentes, conforme Equação (30), onde k é o número de *clusters*, S_i é a distância média entre cada instância até o centroide do cluster i, $M_{i,j}$ é a distância entre os centroides C_i e C_j. Quanto menor o valor, melhor é a qualidade do agrupamento, ou seja, um valor próximo a zero indica agrupamentos bem definidos e compactos, com boa separação entre os grupos, enquanto valores maiores indicam agrupamentos menos coesos (DAVIES; BOULDIN, 1979).

$$DB = \frac{1}{k}\sum_{i=1}^{k} D_i. \quad (30)$$

Onde:

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}}.$$

$$D_i = \max_{j \neq i} R_{i,j}.$$

vii) O coeficiente de silhueta, do inglês *silhouette coefficient*, segundo afirmam Palacio-Nino e Berzal (2019), é a métrica de avaliação da qualidade de um agrupamento mais usada, onde valores positivos apontam uma grande separação entre os grupos, ou seja, os grupos são claramente distintos. Já valores negativos evidenciam que os *clusters* não estão bem agrupados, ou seja, os objetos podem ter sido

atribuídos aos grupos de forma equivocada. Os valores do coeficiente variam de $[-1, 1]$ portanto, quanto mais próximo de 1, melhor o agrupamento realizado. Segundo Kaufman e Rousseeuw (1990), pode-se interpretar o resultado do coeficiente conforme apresenta o QUADRO 4 e o método consiste em 3 etapas descritas a seguir, entretanto, Palacio-Nino e Berzal (2019) ressaltam a alta complexidade computacional da métrica – $O(dn^2)$:

a. Para cada objeto, a distância média $a(i)$ para todos os demais objetos no mesmo *cluster* é calculada, conforme Equação (31), onde $|C_a|$ é a cardinalidade do *cluster* a, ou seja, o número de elementos do *cluster*.

$$a(i) = \frac{1}{|C_a|} \sum_{j \in C_a, i \neq j} dist(i, j). \quad (31)$$

b. Para cada objeto, a distância mínima média $b(i)$ entre um objeto de um *cluster* e outro objeto de um *cluster* diferente é calculada, conforme Equação (32).

$$b(i) = \min_{C_b \neq C_a} \frac{1}{|C_b|} \sum_{j \in C_b} dist(i, j). \quad (32)$$

c. Então, para cada objeto, o coeficiente de silhueta é determinado, conforme Equação (33), e o coeficiente de silhueta global é dado pela média do coeficiente de silhueta de cada objeto, conforme Equação (34).

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \quad (33)$$

$$S = \frac{1}{n} \sum_{i=1}^{n} s(i). \quad (34)$$

QUADRO 4 – INTERPRETAÇÃO DOS RESULTADOS DO COEFICIENTE DE SILHUETA

Resultado do coeficiente de silhueta	Interpretação possível
$0{,}70 < S \leq 1$	Grupos com estrutura muito robusta
$0{,}50 < S \leq 0{,}70$	Grupos razoavelmente unidos
$0{,}25 < S \leq 0{,}50$	Estrutura fraca, aconselha-se tentar outros métodos de clusterização
$S \leq 0{,}25$	Nenhuma estrutura encontrada

FONTE: A autora (2022) com base em Kaufman e Rousseeuw (1990).

2.2.3.5.1 MANOVA

Testes multivariados estatísticos também podem ser usados para comparar separação entre grupos formados por algoritmos diferentes com o objetivo de selecionar o que possui o melhor agrupamento. Tendo em vista somente o propósito de ranqueamento entre os algoritmos aplicados não se faz necessário que os pressupostos da MANOVA (Análise multivariada de variância), como independência, normalidade, multicolinearidade, linearidade, entre outros, sejam cumpridos.

Dentre os principais tem-se: i) De Wilk – O mais utilizado, ii) Lawley-Hotteling, iii) Pillai – O que apresenta maior robustez e iv) maior raiz de Roy (SEEBREGTS, 2022).

1. Teste de Wilk: Quanto mais próximo de 0 o Λ, mais diferente são os grupos em teste (SEEBREGTS, 2022). Conforme Equação (35), em que B é a variância entre os grupos, do inglês *between* e W é a variância intra-*cluster*, do inglês *within* (LUDWIG, 2021).

$$\Lambda = \frac{\det(W)}{\det(B+W)} = \frac{|W|}{|B+W|}. \tag{35}$$

2. Teste de Lawley-Hotelling: Ao contrário do Teste de Wilk, quanto maior o valor de Λ mais diferente são as médias (SEEBREGTS, 2022), conforme Equação (36), em que B é a variância entre os grupos e W é a variância intra-*cluster* (LUDWIG, 2021).

$$\Lambda = tr(W^{-1}B). \quad (36)$$

3. Teste de Pillai: Da mesma forma que o Teste de Lawley-Hotteling visto anteriormente, o Teste de Pillai é um teste estatístico de valor positivo, portanto, quanto maior for o valor de Λ, mais significativas são as diferenças entre as médias (SEEBREGTS, 2022), conforme Equação (37), em que B é a variância entre os grupos e W é a variância intra-*cluster* (LUDWIG, 2021).

$$\Lambda = tr(W^{-1}B(I + W^{-1}B)^{-1}). \quad (37)$$

4. Teste de maior raiz de Roy: A diferença entre as médias dos grupos é maior a medida que o valor do teste também cresce, assim como também são avaliados os testes de Pillai e Lawley-Hotelling (SEEBREGTS, 2022). O teste é dado conforme Equação (38), em que λ_j são os autovalores de $W^{-1}B$, em que B é a variância entre os grupos e W é a variância intra-*cluster* (LUDWIG, 2021).

$$max\ \lambda_j. \quad (38)$$

Portanto, para o teste de Wilks, quanto maior for a diferença entre os grupos, mais próximo de 0 será o resultado do teste. Já para os demais testes, Pillai, Hotelling-Lawley e Roy, quanto maior o valor da estatística do teste, mais significativo são as diferenças entre os grupos.

Padilla *et al*. (2007) se utilizam do teste de Hotelling-Lawley para comparar o resultado da clusterização realizada por 4 diferentes algoritmos, sendo considerado o algoritmo com melhor agrupamento o que obteve um maior valor no teste.

2.2.4 Classificação

Os algoritmos de classificação têm como objetivo, segundo Batista (2019), classificar o conjunto de dados de acordo com as características de uma base de

resposta, onde a classificação já está feita, ou seja, são usados para classificar novos dados.

É um tipo de aprendizagem de máquina supervisionada, pois o modelo irá identificar padrões nos vetores de entrada, também conhecido como base de treinamento, e classificar, em classes c, novos dados com base no que foi dado como exemplo e características padrão, por isso são também conhecidos como modelos preditivos (BATISTA, 2019).

São algoritmos que tem como objetivos encontrar um mapeamento $f : X \rightarrow Y$ e minimizar a quantidade de objetos atribuídos a classes erroneamente. O método aprende uma função matemática f com valores de atributos em seu domínio e valores de classe em sua imagem (SILVA, 2007).

Supondo que o conjunto de pares $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ tenha sido utilizado para obter a estimativa $\hat{f}$ do mapeamento f, em problemas em que se deseja fazer previsão o maior interesse não é saber se $\hat{f}(x_i) \approx y_i$, mas se $\hat{f}(x_0) \approx y_0$, em que (x_0, y_0) é um objeto de teste inédito, não utilizado para treinar o algoritmo de aprendizado. Existe, por isso, um outro conjunto de exemplos, chamado conjunto de teste e composto pelos objetos nunca vistos, que é utilizado para avaliar o desempenho de modelos preditivos (SILVA, 2007).

Depois do aprendizado, por meio dos dados de treino e teste, Silva (2007) recomenta que uma avaliação do desempenho seja realizada.

Os classificadores podem ser do tipo paramétrico ou não paramétrico.

2.2.4.1 Classificador paramétrico

Segundo Batista (2019), os classificadores paramétricos fazem uma suposição sobre a forma funcional de f, tornando mais simples o problema, visto que ajustar uma função arbitrária é menos complexo. Entretanto, a escolha normalmente não reflete a verdadeira forma da função f.

2.2.4.2 Classificador não paramétrico

Já os métodos não paramétricos, ao contrário do funcionamento dos paramétricos, não assumem explicitamente uma forma específica para f. Nesse caso, segundo afirma Batista (2019), busca-se uma estimativa que seja o mais próxima

possível dos pontos de dados, e que possui o potencial de ajustar diferentes formas possíveis para a função f. Porém, um número grande de observações é necessário para que seja viável obter uma estimativa mais precisa.

2.2.4.2.1 Árvore de decisão

O algoritmo de aprendizagem supervisionada árvore de decisão aborda um problema complexo dividindo-o em problemas mais simples e, assim por diante, até que todas as observações sejam classificadas com base nos critérios de divisão escolhidos (SHAIKHINA *et al.*, 2019). É um modelo de classificação não paramétrico, ou seja, não assume hipóteses de partida e são populares por serem facilmente interpretáveis.

Dessa forma, os dados são distribuídos como se fossem uma árvore, por isso o nome, onde as folhas são os resultados e os galhos são as condições baseadas nas entradas para o modelo, podendo os dados ser do tipo numérico, categórico ou ambos (HASTIE *et al.*, 2009).

Dado um conjunto de dados $(x_{1:n}, y_{1:n})$ em que x_i são os atributos e y_i os rótulos, a árvore de decisão vai particionar uma região R_j em sub-regiões, atribuindo um valor de saída γ_j, podendo ser a partir da moda ou das probabilidades para cada classe. Dessa forma, esse particionamento segue até que cada observação esteja sozinha em uma região R_j, ou que possua um grau máximo de pureza, ou ainda, que o critério de parada estabelecido ocorra. Ao final, haverá um número $j = 1, 2, \dots, J$ de regiões disjuntas e a predição é dada conforme a Equação (39), onde x é o vetor com os atributos, $\Theta = \{R_j, \gamma_j\}_1^J$, γ_j é a saída atribuída à região R_j, $I()$ é uma função de indicação que retorna 1 caso $x_i \in R_j$, e 0 caso contrário (JUNIOR, 2018).

$$T(x; \Theta) = \sum_{j=1}^{J} \gamma_j I(x_i \in R_j). \tag{39}$$

Os nós j da ramificação podem ser internos ou terminal. O interno é um particionamento onde determinou-se o atributo x_i e um valor de corte s_j que melhor segmenta a região R_j. Majoritariamente o corte é em duas partes (Árvore Binária), evitando uma grande fragmentação dos dados e a posterior existência de poucos pontos para os cortes subsequentes. É composto por uma regra do tipo Se-Então,

como por exemplo $x_i \leq s_j$ quando se trata de dados numéricos ou $x_i = característica_j$ quando os dados são do tipo categórico, dessa forma as saídas são verdadeiras ou falsas (HASTIE *et al.*, 2009).

Já um nó terminal está posicionado em uma das extremidades da estrutura do modelo. Este tipo de nó define o valor de saída em uma predição ou classificação, caso a amostra apresentada ao modelo atinja esta terminação da árvore (HASTIE *et al.*, 2009).

No treinamento do modelo classificador da árvore de decisão, uma métrica de impureza é calculada, normalmente utiliza-se o índice de Gini ou a Entropia Cruzada, conforme Equações (40) e (41), onde $\hat{p}_{mk}$ é a proporção de pontos para uma determinada classe k de uma região R_j (JUNIOR, 2018).

$$Índice\ de\ Gini = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}). \tag{40}$$

$$Entropia\ Cruzada = -\sum_{k=1}^{K} (\hat{p}_{mk} \log \hat{p}_{mk}). \tag{41}$$

O crescimento da árvore de decisão é um problema de otimização combinatória, onde há a escolha do melhor par (x_i, s_j) para cada nó j, com a estimativa do γ_j a partir de uma medida de impureza (JUNIOR, 2018).

2.2.4.2.2 Floresta aleatória

O algoritmo de classificação chamado floresta aleatória é um conjunto de árvores de classificação ou regressão (ALESSIA *et al.*, 2017). Essa característica o torna mais robusto e complexo, elevando o custo operacional, em contrapartida, possui melhores resultados tendo em vista a redução da variância. É um modelo que lida melhor com dados multicolineares. A saída do modelo segue a Equação (42), onde B é o número total de árvores, $T()$ é a resposta de uma árvore b para um vetor de entrada x_i e Θ_b representa os parâmetros da árvore (JUNIOR, 2018).

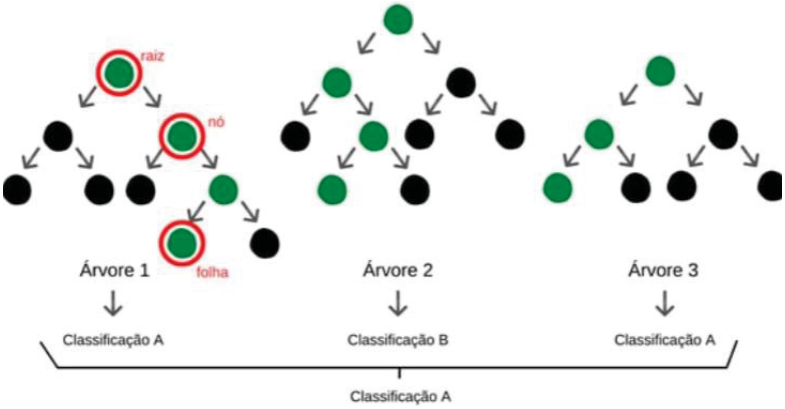
$$\hat{f}^B(x_i) = \frac{1}{B} \sum_{b=1}^{B} T(x_i, \Theta_b). \tag{42}$$

Na maioria das vezes, o modelo é treinado de acordo com o método *bagging*, que diz que a combinação de modelos de classificadores leva a um melhor resultado (ALESSIA *et al.*, 2017).

Todas as árvores de decisão na floresta aleatória são modelos separados. Cada uma delas utiliza um subconjunto de características aleatórias para prever um alvo que depois se acumulam para prever um alvo mais preciso (ALESSIA *et al.*, 2017).

Com as árvores de decisão treinadas, com diferentes características de construção e regras, cada uma delas dá uma resposta para o objeto x_i. O algoritmo da floresta aleatória retorna a resposta majoritária, no caso exemplificado pela FIGURA 15, a resposta majoritária das árvores é a classificação A, dessa forma, a floresta assume como A a resposta para esse objeto (ARIZA *et al.*, 2022).

FIGURA 15 – FUNCIONAMENTO DO ALGORITMO DA FLORESTA ALEATÓRIA



FONTE: Ariza *et al.* (2022).

O algoritmo é treinado com conjuntos de dados do mesmo tamanho do conjunto de dados de treino, chamados de *bootstraps*, criados a partir de uma reamostragem aleatória dos dados de treino. Uma vez que uma árvore é construída, um conjunto de *bootstraps*, que não inclua nenhum registro específico do conjunto de dados original (amostra *out-of-bag* - OOB) é usado como conjunto de teste. A taxa de erro da classificação de todos os conjuntos de teste é a estimativa OOB do erro generalizado (ALESSIA *et al.*, 2017).

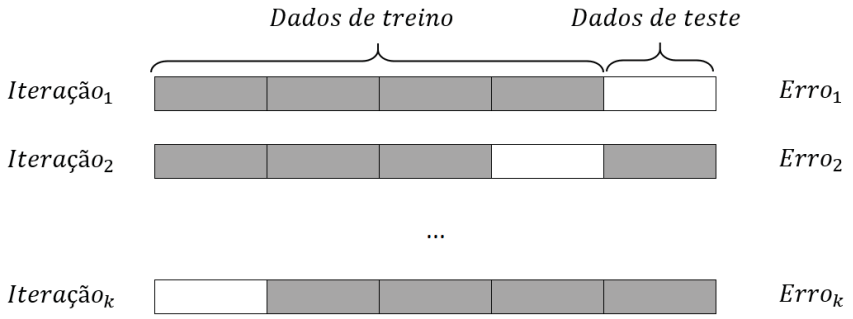
Para dividir um nó binário da melhor maneira, o índice de Gini deve ser maximizado. Um baixo Gini ou igual a 0, quer dizer que o nó é puro, é o mesmo que dizer que uma das variáveis é mais significante no particionamento dos dados nos dois

grupos. Por outro lado, quando o Gini se aproxima do valor 1, o nó é impuro, ou seja, aumenta-se o número de classes uniformemente distribuídas neste nó. Dessa forma, ele pode ser usado para classificar a importância das características para um problema de classificação (ALESSIA *et al.*, 2017).

2.2.4.3 Validação cruzada de K-*Fold*

A validação cruzada é uma técnica usada na validação de um modelo de predição, onde os dados de treino são separados em k blocos e a cada iteração são usados $k-1$ blocos como dados de treino e a validação ocorre com o último bloco, assim como ilustra a FIGURA 16. Normalmente os valores selecionados para k são 5, 10 e 20. Ao final, seleciona-se o modelo com o menor erro (ANGUITA *et al.*, 2012).

FIGURA 16 – EXEMPLO DO FUNCIONAMENTO DA VALIDAÇÃO CRUZADA K-FOLD, COM K=5



FONTE: A autora (2023).

2.2.4.4 Matriz de confusão

Para avaliar o modelo de um classificador, uma das medidas de desempenho utilizadas é a matriz de confusão que indica a acurácia, que é o percentual de acertos sobre todas as previsões do algoritmo. Dado os objetos de entrada, tanto de treino quanto posteriormente de teste, os mesmos serão classificados, podendo as classes estarem corretas ou não, dessa forma a precisão é avaliada conforme a Equação (43) (SILVA, 2007).

$$Acurácia = \frac{Quantidade\ de\ acertos}{Quantidade\ de\ objetos\ classificados}. \quad (43)$$

A matriz de confusão, também conhecida por matriz de erro, é, segundo Patro e Patra (2014), uma forma visual de avaliar a performance do modelo de classificação, onde cada linha representa o valor correto da classe de acordo com a base de treino ou teste, enquanto cada coluna representa o valor previsto, de acordo com o modelo, para aquele objeto. Dessa forma, todos os dados corretamente previstos se localizam na diagonal e os erros fora dela, sendo uma forma clara de identificá-los, conforme ilustra o QUADRO 5. A acurácia do exemplo dado no QUADRO 5 é de 70%, pois, conforme a Equação (44), é a soma dos acertos que estão presentes na diagonal do quadro ($5 + 3 + 11 = 19$), dividido pelo total de previsões ou classificações realizadas que é a soma de todo o quadro, ou seja, todos os valores da diagonal somados ao que estão fora dela ($19 + 3 + 2 + 2 + 1 = 27$).

QUADRO 5 – EXEMPLO DE UMA MATRIZ DE CONFUSÃO

Classes de entrada	Classes previstas		
	Gato	Cachorro	Coelho
Gato	5	3	0
Cachorro	2	3	1
Coelho	0	2	11

FONTE: A autora (2023) com base em Patro e Patra (2014).

3 METODOLOGIA

A presente seção descreve o método de pesquisa aplicado e as etapas metodológicas do trabalho, desde o planejamento, passando pelos métodos de coleta e análise dos dados até a interpretação dos resultados a partir de indicadores de avaliação de desempenho dos métodos de clusterização e classificação aplicados, abordando também os instrumentos utilizados.

3.1 MÉTODO DE PESQUISA

O método de pesquisa do trabalho pode ser definido a partir da classificação da pesquisa, unidade de análise e seleção do público alvo descritos a seguir.

3.1.1 Classificação da pesquisa

O trabalho desenvolvido é da natureza de pesquisa aplicada dado que o modelo desenvolvido pode ser aplicado de maneira prática e imediata, como propõe Silva e Menezes (2005), para resolução de problemas específicos reais no campo de CRM.

Quanto à abordagem do problema, a dissertação desenvolvida trata-se de um estudo predominantemente quantitativo uma vez que, segundo Silva e Menezes (2005), se utilizam dados quantificáveis, ou seja, podem ser traduzidos em números para análise ou classificação.

No que diz respeito ao processo de raciocínio ou também chamada de lógica de investigação, a pesquisa pode ser classificada como dedução, posto que o estudo se baseia na premissa de que clientes de uma mesma empresa possuem comportamentos de compra diferentes e, dessa forma, é possível separá-los em diferentes grupos (ANDRADE, 2001).

Considera-se o trabalho aqui descrito como uma pesquisa exploratória dado que, segundo Gil (2002), o objetivo desse tipo de estudo é proporcionar esclarecimento e maior compreensão do tema. A construção de grupos semelhantes que funcionam como uma hipótese para que seja possível uma posterior aplicação conclusiva das estratégias pela empresa corrobora com a classificação adotada.

Tendo em vista o objetivo de estudar um fenômeno contemporâneo e em seu contexto real de aplicação, a presente dissertação pode ser enquadrada como um estudo de caso quanto aos seus instrumentos técnicos (BENBASAT *et al.*, 1987).

A TABELA 2 apresenta um resumo das informações descritas anteriormente.

TABELA 2 – CLASSIFICAÇÃO DA PESQUISA

Classificação	**Enquadramento da pesquisa**
Natureza da pesquisa	Aplicada
Abordagem do problema	Quantitativo
Lógica de investigação	Dedução
Objetivos de pesquisa	Exploratório
Procedimentos técnicos	Estudo de caso

FONTE: A autora (2023).

3.1.2 Unidade de análise

Na presente pesquisa os clientes de uma empresa americana de *fast-food* no Brasil, membros do programa de fidelidade da companhia, fazem parte da unidade de análise, juntamente com os dados históricos de compras e categoria dos produtos adquiridos pelos mesmos.

3.1.3 Seleção do público-alvo

O critério para a seleção apenas das informações referentes aos membros do programa de fidelidade baseou-se na disponibilidade dos dados de venda identificados possibilitando a coleta e extração para posterior aplicação de algoritmos, permitindo a clusterização e classificação dos clientes.

Dessa maneira, a pesquisa teve como população alvo, grupo para o qual se deseja obter informação, a companhia de *fast-food* americana no Brasil com a população de pesquisa limitada aos membros do programa de fidelidade na sua totalidade. O mesmo foi lançado ao mercado no ano de 2021 e, somente em 2022 passou a funcionar em todos os canais de venda, sendo, portanto, o espaço temporal

do estudo. Os dados necessários para execução foram extraídos das bases de armazenamento internas da empresa.

3.2 ETAPAS METODOLÓGIAS

O fluxograma da FIGURA 17 apresenta a metodologia aplicada no estudo de clusterização e classificação englobando a seleção do conjunto de dados, redução e seleção das variáveis, aplicação dos algoritmos e avaliação dos resultados.

FIGURA 17 – ETAPAS METODOLÓGICAS DA PESQUISA

(Continua)

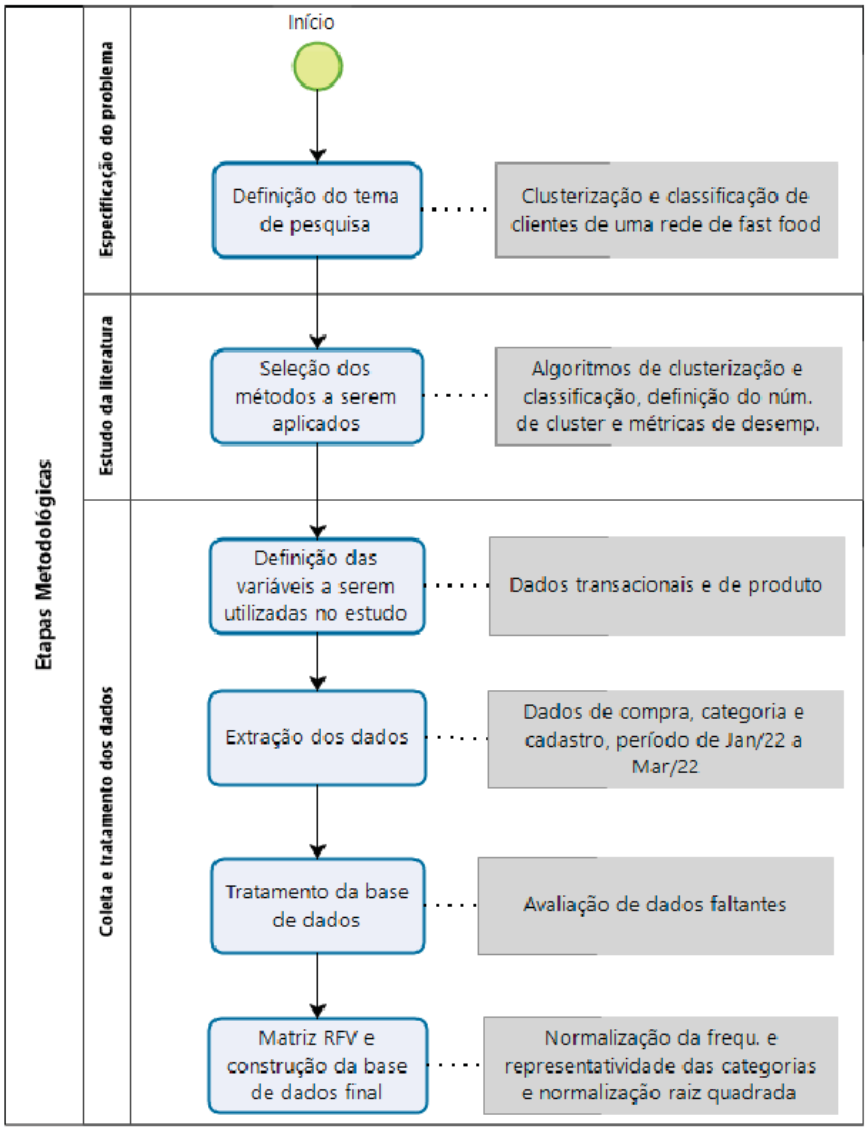
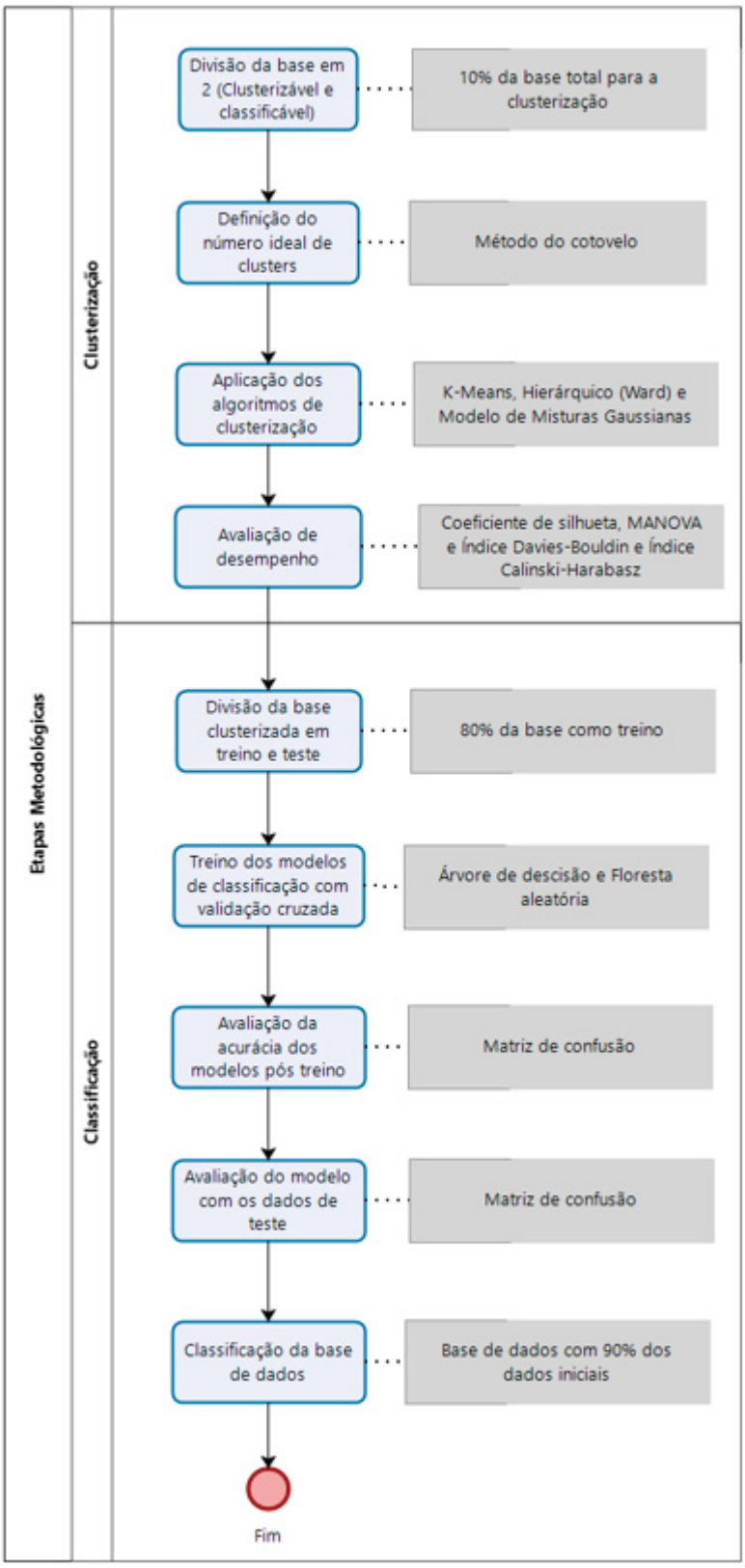


FIGURA 17 – ETAPAS METODOLÓGICAS DA PESQUISA

(Continuação)



FONTE: A autora (2023).

3.2.1 Seleção das variáveis

Em um primeiro momento, diferentes variáveis que poderiam fazer parte do estudo foram listadas (QUADRO 6) e, posteriormente, uma análise qualitativa foi realizada de forma a selecionar somente as variáveis que tem um maior potencial de impacto para a companhia, tendo em vista a maturidade da mesma no que diz respeito à clusterização de clientes para a personalização das comunicações de *marketing*.

QUADRO 6 – POSSÍVEIS VARIÁVEIS PARA O ESTUDO

(Continua)

Tipo	Variável	Descrição	Unidade	Exemplo
Produto	TP	Tipo	-	Sanduíche, sobremesa, acompanhamento, etc.
	CAT	Categoria	-	*Core, Premium*, etc.
	LAN	Lançamento	Binário	Sim ou não
	REC	Recompensa do programa de fidelidade	-	Não, parcial e gratuita.
Preço	Desc_SD	Desvio padrão do desconto	-	-
	MP	Margem do produto	%	-
	PLAT	Plataforma de desconto	-	Cupom do aplicativo, *Menu Board*, 9,90, etc.
Cliente	ID	Número identificador do cliente	-	-
	GEN	Gênero	Binário	F ou M
	IDD	Idade	Anos	-
	CAD	Data de cadastro no programa de fidelidade	Dias	-

QUADRO 6 – VARIÁVEIS POSSÍVEIS PARA O ESTUDO

(Conclusão)

Tipo	Variável	Descrição	Unidade	Exemplo
Transacionais	QNT	Quantidade vendida	Unidades	-
	VL	Valor bruto de venda	R$	-
	DT	Data de compra	dd/mm/aaaa	-
	LJ	Loja de compra	-	-
	SC	Participação do canal de compra	%	*Drive-Thru*, Totem, Quiosque, etc.
	FP	Forma de pagamento	-	Pix, carteiras digitais, dinheiro, etc.
	RFV	Combinação dos dados transacionais: R - Recência, F – Frequência V - Valor	-	-

FONTE: A autora (2023).

Foram selecionadas para o estudo 6 variáveis classificadas em 2 tipos principais:

(i) Dados relacionados aos produtos adquiridos: Para que seja possível identificar qual é o tipo do produto preferido pelo cliente a quantidade de compra de sanduíches, acompanhamentos e sobremesas foram usadas no estudo. É importante ressaltar que a companhia possui outras classificações como bebidas e itens extras – por exemplo uma carne extra no sanduíche. Porém, o volume de itens extras em comparação aos demais produtos é próximo a 1% do total de itens vendidos e o consumo de bebidas normalmente já está atrelado ao consumo de sanduíches e acompanhamentos. Dessa forma, esses outros dois tipos de produtos citados não foram considerados;

(ii) Transacionais: Dados relacionados às compras identificadas dos clientes, são elas: recência, frequência de compra e o valor gasto.

As demais variáveis do tipo produto e transacionais não foram utilizadas no estudo tendo em vista o objetivo da companhia de dar os primeiros passos em direção à comunicação personalizada, adicioná-los poderia trazer uma complexidade de execução em um momento em que a companhia não está madura o suficiente para tal.

Já com relação as demais variáveis do tipo clientes, não as utilizar foi uma escolha que levou em consideração não somente o momento da empresa no que diz respeito à personalização, mas também à disponibilidade de dados. Os dados demográficos e geográficos dos clientes solicitados no momento do cadastro no aplicativo da marca para telefones portáteis, porém não eram obrigatórios. Dessa forma, é possível encontrar muitos clientes na base de dados sem esses campos devidamente preenchidos e, em 2022, a empresa deixou de solicitar esses dados entendendo que não era algo que ela utilizava no dia a dia e causava certa fricção no momento da jornada do cliente conhecida como aquisição. É importante salientar que atualmente a empresa, apesar de possuir ofertas regionalizadas, também está em uma fase inicial de utilização dessa estratégia de precificação.

Entende-se que as comunicações de *marketing* personalizadas podem ajudar na rentabilidade das empresas ao oferecer o desconto no momento certo para o cliente certo, porém nesse primeiro momento optou-se por não adicionar ao modelo de clusterização variáveis do tipo preço, novamente pela maturidade da empresa com relação ao tema, mas também pelo esforço de obtenção desses dados. Hoje a companhia possui limitações no que diz respeito à extração e consultas da base de dados e também às permissões de acesso a esse tipo de informação que é extremamente sensível.

3.2.2 Coleta dos dados

Foram coletados dados diretamente de dois bancos de armazenamento de dados da empresa: i) *SAP Analytics Cloud* e ii) *Salesforce.* O período de coleta foi de Janeiro de 2022 a Março de 2022.

O lançamento do programa de fidelidade, que permite a identificação das compras, se deu em Fevereiro de 2021, porém a omnicanalidade, funcionamento do programa em todos os canais de venda da companhia (Exemplo: *Delivery*, *drive-thru,*

totem de autoatendimento, etc), foi implementada em Dezembro de 2021, o comportamento de compra dos clientes é melhor mensurado desde então, por isso considerou-se como período de início da análise Janeiro de 2022.

Dentre os dados transacionais coletados, tem-se a data da compra, o número da nota fiscal eletrônica, os quais são necessários somente para construção da base para extração, sendo que, dessa forma, cada linha da base de dados constituirá uma compra diferente. Além da data da compra e do número da nota fiscal, a base é constituída também pelo código de identificação do cliente criado internamente na companhia para cumprir a Lei Geral de Proteção de Dados (LGPD - Lei n° 13.709/2018) vigente no Brasil atualmente, e, ainda, o valor total bruto da compra considerando os impostos. A taxa de frete cobrada nas compras no canal *delivery*, por não ser repassada à companhia e sim aos parceiros logísticos, não foi considerada (FIGURA 18). Esses dados dizem respeito somente às compras identificadas pelos clientes.

FIGURA 18 – COLETA DE DADOS TRANSACIONAL

		Medidas	VENDA BRUTA - LOYALTY TOTAL
Data	ID Customer Unificado	NFe	
Jan 1, 2022	00014066-0b98-4a92-b032-71865de66236	32493284923842930001406	R$ 51,90
	0001413f-4eb1-4302-9355-53bb4e530403	02934823173348326437264	R$ 15,00
		28472347284729384728341	R$ 15,00

FONTE: A autora (2023).

Para os dados relacionados aos produtos adquiridos considerou-se a data da compra, o código de identificação do cliente e quantidade vendida de cada uma das categorias de produtos (FIGURA 19). Esses dados dizem respeito somente às compras identificadas pelos clientes.

FIGURA 19 – COLETA DE DADOS RELACIONADOS AOS PRODUTOS ADQUIRIDOS

	Subcategoria	ACOMPANHAMENTOS	ADD ON	BEBIDA	SANDUICHE	SOBREMESA
Data	ID Customer Unificado					
Mar 1, 2022	00014066-0b98-4a92-b032-71865de66236	-	-	-	1	-
	0001413f-4eb1-4302-9355-53bb4e530403	1	-	1	2	-
	00017494-34ea-4208-9778-861777470509	2	-	2	1	-

FONTE: A autora (2023).

Além disso, ainda para os dados relacionados aos produtos, uma base foi exportada com a quantidade total de itens vendidos pela companhia de cada categoria mensalmente, independente da identificação dos clientes, para que possa ser calculado posteriormente o quanto cada categoria contribui percentualmente para o total de itens vendidos.

De forma a utilizar para o estudo somente usuários que continuam cadastrados no programa de fidelidade da empresa, afinal são esses clientes que a companhia poderá posteriormente ativar via comunicações de CRM (*e-mail marketing, push notification* e *SMS)*, o *status* do usuário deve ser usado como um filtro que servirá também para desconsiderar da base de dados os clientes que são suspeitos de fraude, eliminando parte dos dados *outliers*. Isso porque a companhia possui uma inteligência para detectar comportamentos fraudulentos a partir dos dados de compra, principalmente relacionado a frequência dentro de um dado intervalo de tempo, à título de exemplo tem-se o operador de caixa que insere o próprio CPF ou de terceiros nas compras para atingir a meta mensal de identificação de compras da loja. Dessa forma, apenas usuários com o *status* ativo ("*active*") serão considerados. Importante ressaltar que a variável não será usada como parte do conjunto de dados para clusterização e posterior classificação.

Os dados relacionados ao tempo de cadastro dos clientes no programa de fidelidade também são úteis para o estudo, mas não como variável de clusterização (FIGURA 20). A frequência do modelo RFV será normalizada pelo período pelo qual o usuário faz parte do programa de fidelidade. É algo de extrema importância pois, no exemplo em que um cliente *A* e um cliente *B* tem 2 compras entre Janeiro/2022 e Março/2022 a frequência mensal de ambos seria de 0,66, porém se o cliente *A* se cadastrou em Dezembro/2021 e o *B* em Março/2022, o cliente *A* continua com uma frequência mensal de 0,66, enquanto o *B* passa a ter uma frequência mensal de 2, visto que é cliente da companhia somente há um mês.

FIGURA 20 – DADOS DO USUÁRIO

FieloPLT__JoinDate__c	FieloPLT__Status__c	V_ExternalId__c
2020-12-16	Active	72b28e6a-2791-410a-a9de-fbb4110336be
2020-12-17	Active	d0668f78-1b3b-49e4-947f-5cda3644c26f
2020-12-17	Active	6731bf76-fa26-4bbd-8ded-af0b3b891eca
2020-12-17	Active	ea26f177-78c4-4d2a-9ffa-a3b0c3e4f950
2020-12-18	Active	35bda306-9534-46d1-abcb-f737f18ac34b
2021-01-08	Active	f7ef751e-2ef3-4691-974f-6890934be391
2021-01-11	Active	38dc3c59-34b3-407c-a720-ba3e6203a05a

FONTE: A autora (2023).

3.2.3 Tratamento da base de dados

A partir da coleta dos dados foi necessário realizar um tratamento das bases por meio da verificação da padronização dos códigos identificados dos clientes. Conforme ilustrado pela FIGURA 21, caso algum dos códigos não estivesse no formato alfanumérico de GUID (*globally unique identifier)* o mesmo era ajustado para tal formato, evitando assim, a perda dos dados referentes aquele cliente.

FIGURA 21 – TRATAMENTO DO CÓDIGO DO CLIENTE DA BASE DE COMPRAS

Data	ID Customer Unificado	NFE	VENDA BRUTA - LOYALTY TOTAL
15/01/2022	ffff9409-7efb-43bc-8a20-951acde8980c	5,32201E+43	R$ 60,90
15/01/2022	{"customer_id":"8b6e87ba-b7f1-4309-a207-5350d17d074e"}	4,12201E+43	R$ 5,00

FONTE: A autora (2023).

Na mesma base pode haver dados de venda bruta com valor 0, isso porque caso o cliente tenha resgatado uma recompensa do programa de fidelidade, esse produto sai de graça, porém esses dados não foram desconsiderados tendo em vista que é importante registrar que houve uma visita nas lojas da marca desse cliente, não afetando a recência e frequência (FIGURA 22).

FIGURA 22 – IMPACTO NA MATRIZ RFV AO TRATAR A BASE: VALOR DE VENDA IGUAL A 0

Exemplo de base com o valor bruto 0

Data	Cliente	Valor	NFe
Ontem	Cliente 1	R$ 0,00	1
Anteontem	Cliente 1	R$ 20,00	2
Anteontem	Cliente 1	R$ 10,00	3

Impacto ao tirar a transação com o valor 0

-	Valor	Frequência	Recência
Cliente 1	R$ 30,00	2	2

Impacto ao não tirar o valor 0

-	Valor	Frequência	Recência
Cliente 1	R$ 30,00	3	1

FONTE: A autora (2023).

No caso do campo do identificador da nota fiscal estar vazio, os dados foram mesmo assim considerados de modo a não interferir nas informações relacionadas ao valor gasto e a recência do referido cliente. (FIGURA 23).

FIGURA 23 – IMPACTO NA MATRIZ RFV AO TRATAR A BASE: NFe VAZIO

Exemplo de base com o código na nota fiscal vazio

Data	Cliente	Valor	NFe
Ontem	Cliente 1	R$ 20,00	N/A
Ontem	Cliente 1	R$ 10,00	1

Impacto ao tirar a transação com o valor vazio

-	Valor	Frequência	Recência
Cliente 1	R$ 10,00	1	1

Impacto ao não tirar o valor vazio

-	Valor	Frequência	Recência
Cliente 1	R$ 30,00	2	1

FONTE: A autora (2023).

Foram retirados da base de cadastro, conforme explicitado anteriormente, os clientes com *status* diferente de ativo.

3.2.4 Matriz RFV e dados finais para clusterização

Para a viabilizar a construção da matriz RFV, foi necessário desenvolver uma base exclusiva, contendo o código identificador do cliente, a data das compras e o valor bruto de venda. Nesse caso, cada linha da base corresponde a uma compra diferente.

Na sequência, a matriz RFV foi então gerada, a partir da função *rfm_table_order* do pacote *rfm* do *software* de programação computacional R. Para a elaboração da variável recência, foi considerada com data de referência o primeiro dia do mês seguinte ao término do período sob análise.

Não utilizou-se para o estudo a escala da RFV de 1 a 5 como apresenta a literatura, mas sim, os valores reais do conjunto de dados, como por exemplo o total gasto no período pelo cliente.

Com a RFV criada, uniu-se então a ela os dados de cadastro dos membros (data de cadastro e *status*), onde foi possível filtrar os clientes com *status* diferente de ativo e desconsiderar da matriz.

Para a normalização da frequência, conforme explicado na seção 3.2.2, primeiro calculou-se a quantidade de dias entre o cadastro e o primeiro dia do mês subsequente ao mês de término do período de análise para o caso em que o cadastro tenha sido realizado ao longo do período de análise. Já para o caso em que o cadastro tenha ocorrido antes do início do período de análise considerou-se para a variável dias a quantidade de dias do período de análise, 90 dias. Em seguida, foi realizada a divisão dos dias em meses considerando um mês com 30 dias e, por fim, a divisão da frequência total pelo número de meses, resultando na frequência mensal de cada cliente (FIGURA 24).

FIGURA 24 – EXEMPLO DE NORMALIZAÇÃO DA FREQUÊNCIA DOS CLIENTES

Exemplo com o período de análise entre 01/01/2022 e 31/03/2022

Cliente	Cadastro	Dia subsequente ao fim do período	Dias entre cadastro e fim do período de análise	Frequência total no período de análise	Meses na base	Frequência Mensal
00014066-0b98-4a92-b032-71865de66236	31/07/2021	01/04/2022	90	3	3	1,0
0001413f-4eb1-4302-9355-53bb4e530403	01/01/2022	01/04/2022	90	6	3	2,0
00017494-34ea-4208-9778-861777470509	01/02/2022	01/04/2022	59	2	2	1,0

FONTE: A autora (2023).

Novas variáveis como o total gasto pelo cliente no mês e o valor médio de compra também foram calculados.

A partir da extração dos dados de produtos, a distribuição percentual mensal e do total do período de análise da quantidade vendida pela companhia de cada categoria de produtos foi calculada.

Ainda com relação aos dados de produtos, na base que inclui a identificação, calculou-se a quantidade de produtos comprados de cada categoria por cliente, mensalmente e no total do período de análise. Posteriormente, a distribuição percentual da quantidade de itens comprados por cliente foi realizada e, por fim, novas variáveis foram criadas, uma para cada categoria, referente a normalização do percentual de compra daquele cliente pela distribuição daquela mesma categoria na venda total da companhia.

Dessa forma, um valor igual a 1, representa que o cliente compra aquela categoria na mesma média que a companhia a vende de forma geral, entretanto valores maiores que 1, por exemplo, indicam um consumo acima da média (FIGURA 25). Essa informação é relevante para comunicações de CRM personalizadas, em caso de uma recência muito alta, uma ativação com a categoria principal de consumo pode ser um atrativo, assim como um desconto em uma categoria que a companhia se destaca de forma geral, mas que o cliente não é habituado a consumir também pode ser uma estratégia interessante.

FIGURA 25 – EXEMPLO DE NORMALIZAÇÃO DO PERCENTUAL DE REPRESENTATIVIDADE DAS CATEGORIAS DE PRODUTOS

Visão Companhia	Sanduíche	Acompanhamento	Sobremesa	Total
Quantidade vendida	500	400	100	1000
Participação	50%	40%	10%	100%

Visão Cliente	Sanduíche	Acompanhamento	Sobremesa	Total
Quantidade comprada	8	1	1	10
Participação	80%	10%	10%	100%

Normalização	Sanduíche	Acompanhamento	Sobremesa
Participação companhia	50%	40%	10%
Participação cliente	80%	10%	10%
Normalizado	1,60	0,25	1,00

FONTE: A autora (2023).

Em seguida, para cada cliente a quantidade de itens comprados de cada categoria de produtos da companhia foi adicionada ao conjunto de dados para completar a base utilizada na clusterização.

Por fim, tirou-se as colunas que foram usadas como apoio, resultando em uma base igual a exemplificada pelo QUADRO 7.

QUADRO 7 – EXEMPLO DO CONJUNTO DE DADOS

Identificador	Recência	Frequência Mensal	Valor médio/compra	Sanduíche	Acompanhamento	Sobremesa
Cliente 1	1	4	R$ 80,00	1,6	0,6	1
Cliente 2	10	8	R$ 300,00	1	0,8	1,3

FONTE: A autora (2023).

Parte dos *outliers,* relacionados a frequência, já foram retirados quando os clientes suspeitos de fraude foram retirados da base de dados. Nenhum outro método de retirada de dados atípicos será aplicado tendo em vista que ao retirá-los um grupo específico de clientes pode estar sendo desconsiderado.

A base foi então normalizada, com a utilização do método raiz quadrada (ARRUDA, 1959), de forma a equilibrar a escala, lidar com a assimetria e mitigar o impacto dos dados *outliers* no estudo.

3.2.5 Clusterização

Com a base final, retirou-se aleatoriamente uma amostra pequena (10% do total), tendo em vista que posteriormente seria aplicado um classificador.

O método do cotovelo, amplamente difundido na literatura, foi então aplicado no *software* R, onde calculou-se a soma dos quadrados intra *cluster* no algoritmo K-*Means* para diferentes valores de k, usando como base a amostra retirada para a definição do número k de *clusters* e que serviu como parâmetro inicial para os 3 algoritmos de clusterização.

O primeiro algoritmo executado no *software* R foi o K-*Means*, por meio da função *kmeans*. O segundo foi o algoritmo hierárquico de Ward com a função *hclust* e *dist*, cuja distância utilizada foi a euclidiana. Por último, o modelo de mistura gaussiana

foi aplicado e a função utilizada foi a *mclust* do pacote com o mesmo nome. Todas as funções utilizadas são do *software* R que é a ferramenta em uso para o estudo em conjunto com a ferramenta *RStudio* de visualização. A escolha dos métodos ocorreu, para o *K-Means,* pela ampla aplicação na literatura e, para o método de Ward, pela redução da complexidade computacional dentre os métodos hierárquicos que se apresentam como uma lacuna na aplicação de clusterização de clientes por meio de dados transacionais como o modelo RFV e o modelo de misturas gaussianas também pela lacuna apresentada na revisão sistemática da literatura realizada.

3.2.6 Análise de resultados de clusterização

Com o objetivo de avaliar a melhor clusterização realizada dentre os 3 algoritmos aplicou-se o coeficiente de silhueta, encontrado na revisão da literatura como uma das métricas indicadas para essa avaliação. A função para mensurar o melhor agrupamento foi a *silhouette* do pacote *cluster*.

Para a análise de variância multivariada (MANOVA), com os 4 testes estatísticos: Wilks, Traço de Pillai, Traço de Hotelling-Lawley e Raíz Máxima de Roy, foi usada a função *manova*.

Outros dois índices também foram medidos para avaliar o agrupamento realizado pelos algoritmos de clusterização. De forma a calcular o índice Davies-Bouldin a função utilizada foi a *index.DB*, já para o índice Calinski-Harabasz a função usada foi a *index.G1*, ambas as funções do pacote *clusterSim*.

3.2.7 Classificação

Com a definição dos grupos de clientes com perfis semelhantes com o melhor agrupamento segundo as métricas citadas anteriormente, a base de dados agora devidamente clusterizada foi aleatoriamente separada em 2 grupos: treino e teste.

O grupo treino representa 80% da base e foi utilizado para treinar o modelo de classificação a partir da validação cruzada de *K-Fold*, onde o número de subdivisões k foi igual a 5, assegurando uma melhor modelagem, que não estará enviesado pela formação do conjunto de dados de treino. Para isso utilizou-se a função *train* do pacote *rpart*.

3.2.8 Análise de resultados da classificação

Dentre os modelos de classificação, aplicou-se 2: Árvore de decisão (Método *rpart* na função *train*) e Floresta aleatória (Método *rf* na função *train*). A escolha da árvore de decisão se deu pela ampla difusão na literatura e da floresta aleatória pela lacuna apresentada na revisão sistemática da literatura realizada.

O algoritmo selecionado para dar continuidade ao estudo foi o que obteve uma maior acurácia no treino com a validação cruzada.

Com o modelo definido, os dados de teste foram classificados e a acurácia foi então medida pela matriz de confusão.

Por fim, a base de dados reservada para classificação antes mesmo da aplicação da clusterização volta a ser utilizada agora no modelo de classificação aprovado. As bases após a clusterização e classificação são agrupadas novamente agora obtendo uma coluna a mais, o *cluster* ao qual cada cliente pertence.

4 RESULTADOS

A corrente seção tem como propósito apresentar os resultados obtidos a partir da aplicação da metodologia detalhada no capítulo 3.

Para execução das funções no software R, cujos códigos são apresentados no Apêndice A, foi utilizado um computador Intel® Core™ i7-10510U 1,8GHz e memória de 8GB.

4.1 COLETA E TRATAMENTO DOS DADOS

Para os dados relacionados a cadastro (Data de cadastro e *status* do cliente) encontrou-se 4.836.887 clientes, considerando desde o lançamento do programa, em Fevereiro de 2021 até o final do mês de Março de 2022. Ao retirar da base usuários com *status* diferente de ativo, restaram 4.821.371 clientes, uma redução de mais de 15,5 mil clientes. Não foram encontrados dados faltantes ou fora do padrão.

Já com relação aos dados de compras, foram 4.339.023 transações identificadas no período de Janeiro de 2022 a Março de 2022. Nessa base, além dos ajustes realizados no código identificador do cliente, avaliou-se também os dados faltantes. 11.916 registros não possuíam o número da nota fiscal e mais 65.593 transações eram somente de resgate de recompensas e que, portanto, possuem o valor 0, somados representam aproximadamente 2% da base total, porém optou-se por não retirá-los tendo em vista os impactos já abordados na metodologia.

Nos dados relacionados a produtos, a base final contendo já a distribuição normalizada dos produtos mostra que as 4.339.023 de transações foram realizadas por 1.640.320 clientes.

4.2 MATRIZ RFV E DADOS FINAIS PARA CLUSTERIZAÇÃO E CLASSIFICAÇÃO

Inicialmente foi utilizado uma base de dados referente ao período de Janeiro de 2022 a Julho de 2022 para que o histórico de compras fosse ao menos de 6 meses. Um histórico maior seria mais adequado em virtude da frequência de compra do segmento de *fast-food* não ser tão elevado como de um supermercado, por exemplo. Porém, não foi possível aplicar a função de construção da matriz RFV, por insuficiência de memória do computador utilizado (FIGURA 26). Dessa forma o

período de análise final escolhido para o estudo foi de apenas 3 meses, de Janeiro de 2022 a Março de 2022.

FIGURA 26 – ERRO DE MEMÓRIA DO COMPUTADOR: MATRIZ RFV

```
> rfm_result <- rfm_table_order(orders, customer_id, order_date, revenue, analysis_date) #Matriz RFV
Error in `dplyr::summarise()`:
! Problem while computing `amount = sum(revenue)`.
i The error occurred in group 376314: customer_id = "1b18f2a5-b7ac-478c-b774-b39a5b794ac3".
Caused by error:
! memory exhausted (limit reached?)
Run `rlang::last_error()` to see where the error occurred.
>
```

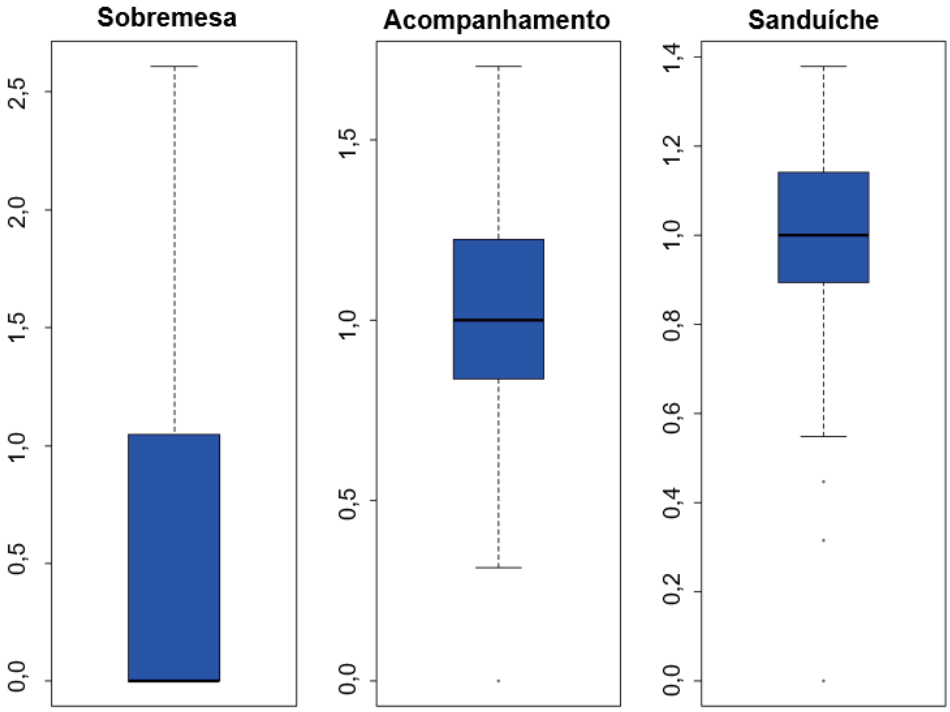
FONTE: A autora (2023).

Ao construir a matriz RFV e retirar os usuários com *status* diferente de ativo restaram 1.626.960 clientes, não sendo encontrado dado vazio.

A base foi então normalizada pelo método raiz quadrada. As FIGURA 27 e FIGURA 28 fornecem uma representação visual das medidas estatísticas básicas do conjunto de dados de cada variável após a normalização. A frequência mensal e o valor médio de compra apresentam inúmeros dados *outliers*, conforme método de visualização *boxplot*.

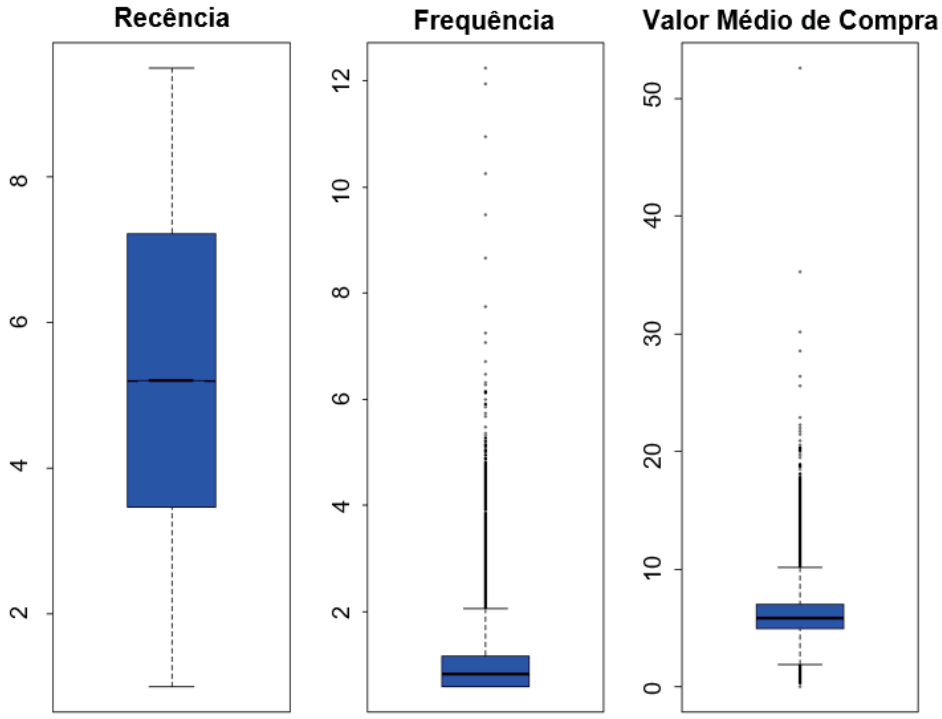
Já a FIGURA 29 fornece a representação contínua da distribuição de probabilidade dos dados, também após a normalização. Pelos histogramas também é possível notar a presença de dados *outliers,* principalmente avaliando as variáveis relacionadas as categorias de produto. A recência é a variável com melhor ajuste.

FIGURA 27 – BOXPLOT DAS VARIÁVEIS: SOBREMESA, ACOMPANHAMENTO E SANDUÍCHE



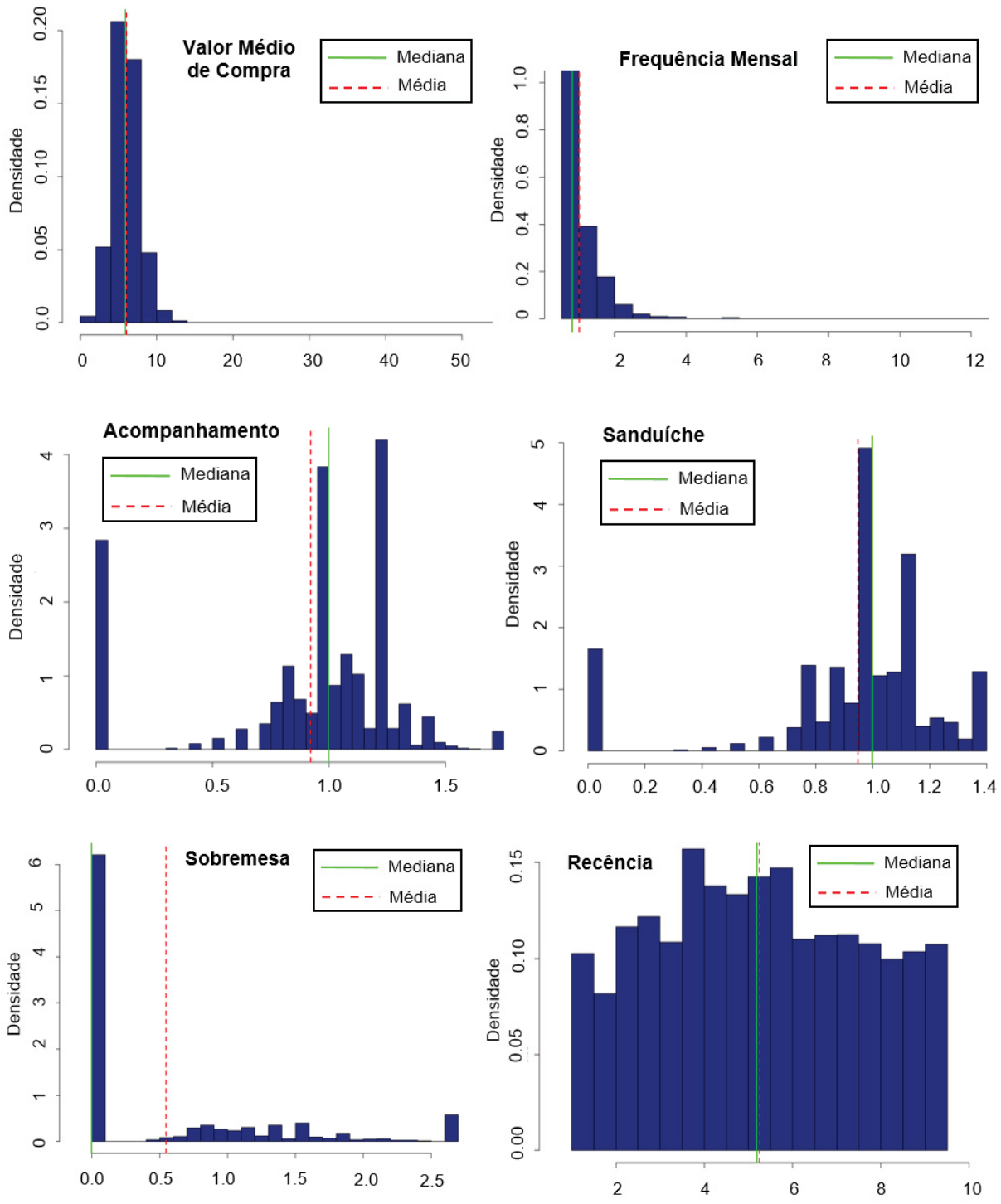
FONTE: A autora (2023).

FIGURA 28 – BOXPLOT DAS VARIÁVEIS: RECÊNCIA, FREQUÊNCIA MENSAL E VALOR MÉDIO DE COMPRA



FONTE: A autora (2023).

FIGURA 29 – HISTOGRAMA DAS VARIÁVEIS NORMALIZADAS



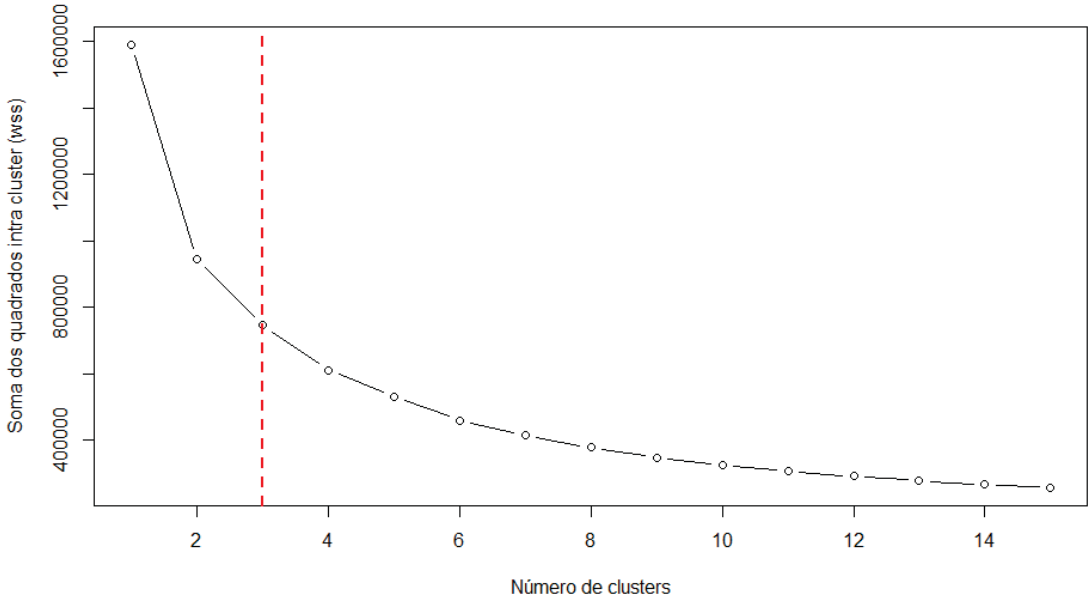
FONTE: A autora (2023).

4.3 CLUSTERIZAÇÃO

Com a base final para clusterização, parte dela foi separada para a clusterização, um total de 162.696 clientes, enquanto a base para classificação ficou com o restante dos clientes, 1.464.264.

Foi aplicado então o método do cotovelo, onde o número máximo de *clusters* foi limitado a 15, a partir dele é possível concluir que o número ideal de cluster k é igual a 3, pois é o onde o gráfico faz um cotovelo e ao aumentar mais que 3 o número de *clusters*, o ganho é muito menor, sendo marginal (FIGURA 30).

FIGURA 30 – MÉTODO DO COTOVELO



FONTE: A autora (2023).

O resultado do método do cotovelo foi então utilizado como parâmetro inicial para os 3 algoritmos de clusterização.

4.3.1 K-*Means*

O QUADRO 8 apresenta o valor médio, já desnormalizado, de cada variável utilizada para a formação de cada grupo por meio do algoritmo K-*Means* que foi executado em menos de 1 segundo. O *cluster* 1 é formado por 73.659 clientes

(45,3%), o 2 tem 44.603 clientes (27,4%), já o último grupo é composto por 44.434 clientes (27,3%).

QUADRO 8 – MÉDIA DAS VARIÁVEIS PARA O AGRUPAMENTO K-*MEANS*

Cluster	Recência	Frequência Mensal	Valor médio/compra	Sanduíche	Acompanhamento	Sobremesa
1	10	1,70	R$ 33,39	0,91	0,90	0,34
2	48	0,57	R$ 60,16	1,09	1,09	0,06
3	50	0,66	R$ 21,52	0,70	0,59	0,63

FONTE: A autora (2023).

4.3.2 Método de Ward

Não foi possível aplicar o Método Ward pela mesma razão apresentada anteriormente para a matriz RFV, memória insuficiente do computador. Tal erro ocorreu na determinação do parâmetro inicial do algoritmo hierárquico, ou seja, a matriz de distância. Sendo assim, na tentativa de contornar esse inconveniente, diferentes distâncias foram utilizadas, porém sem sucesso, conforme apresentado na FIGURA 31 e FIGURA 32, as quais ilustram os erros mencionados.

FIGURA 31 – ERRO DE MEMÓRIA DO COMPUTADOR: DISTÂNCIA *MANHATTAN*

```
>  d <- dist(data, method = "manhattan") # É construída a matriz de distancias entre cada elemento da base de dados
Error: cannot allocate vector of size 98.6 Gb
>
```

FONTE: A autora (2023).

FIGURA 32 – ERRO DE MEMÓRIA DO COMPUTADOR: DISTÂNCIA EUCLIDIANA

```
> d <- dist(data, method = "euclidean") # É construída a matriz de distancias entre cada elemento da base de dados
Error: cannot allocate vector of size 98.6 Gb
>
```

FONTE: A autora (2023).

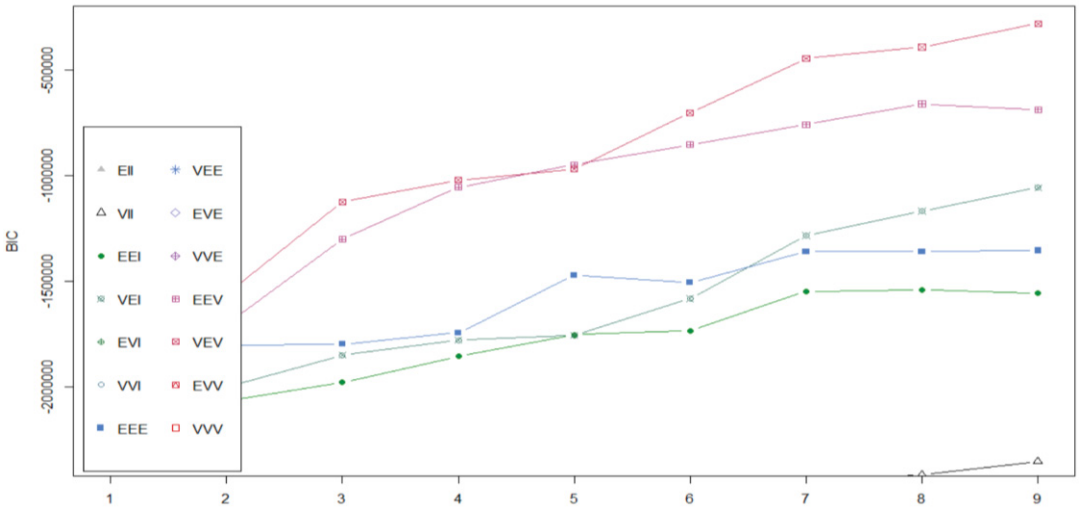
A quantidade de dados que a função suportou foi de aproximadamente 10 mil clientes, 1% da base total de clientes e 10% da base que já estava sendo usada para clusterização.

Para o montante de mais de 15 milhões de clientes que a empresa possui e que cresce a cada dia, 1% é uma amostra muito pequena e, dado que o modelo será usado na prática pela companhia. Um algoritmo que possui limitações como essa não é o ideal, mesmo que a clusterização não seja realizada com frequência.

4.3.3 Modelo de Misturas Gaussianas

Para a clusterização por meio de modelo de misturas gaussianas, a métrica do critério de informação bayesiano (BIC) é usada como definição do número ideal de *clusters* (RELVAS, 2020). Quanto maior for seu valor, melhor será o modelo para representar o conjunto de dados, dessa forma, o modelo indica que 9 é o número ideal de grupos (FIGURA 33). Entretanto, novamente, o viés da pesquisadora foi importante, 9 grupos é muito para que a empresa comece a trabalhar comunicações personalizadas, dessa forma, o número de grupos encontrado pelo método do cotovelo, 3, é mais razoável para esse contexto e, portanto, foi colocado como parâmetro inicial para o modelo.

FIGURA 33 – CRITÉRIO DE INFORMAÇÃO BAYESIANO (BIC)



FONTE: A autora (2023).

De acordo com a FIGURA 33, considerando 3 grupos, o modelo VEV (*Variable volume, Equal shape, Variable orientation*) foi o que se mostrou mais adequado conforme o BIC. Sendo assim, conforme FIGURA 12FIGURA 13, é possível observar

que tal modelo possui distribuição elipsoidal, volume e orientação variáveis e, além disso, mesmo formato.

O QUADRO 9 apresenta o valor médio de cada variável utilizada para a formação de cada grupo, já desnormalizado. O *cluster* 1 é formado por 50.091 clientes (30,8%), o 2 tem 100.102 clientes (61,5%), já o último grupo é composto por 12.503 clientes (7,7%). O modelo foi executado em 17 segundos.

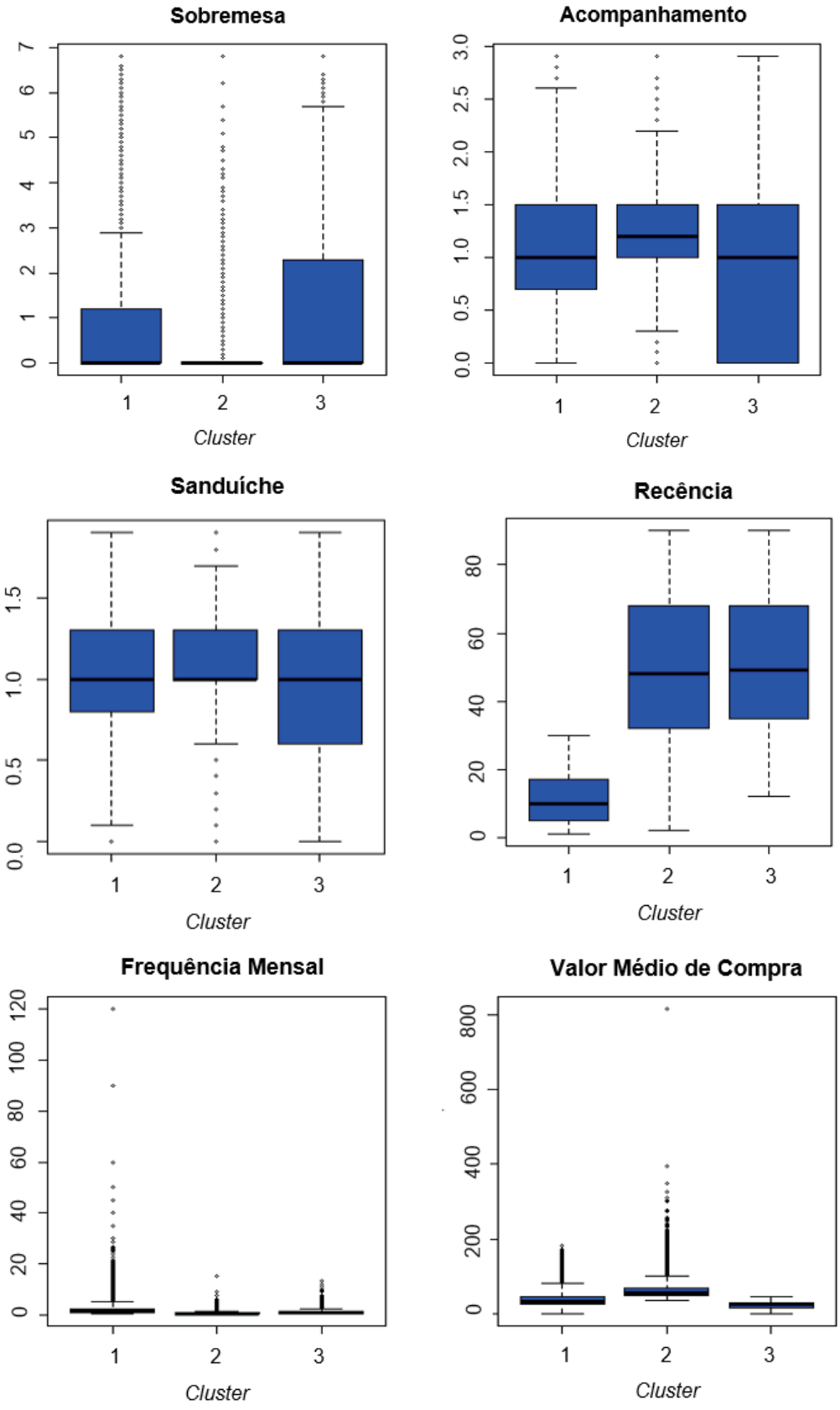
QUADRO 9 – MÉDIA DAS VARIÁVEIS DO MODELO DE MISTURA GAUSSIANA

Cluster	Recência	Frequência Mensal	Valor médio/compra	Sanduíche	Acompanhamento	Sobremesa
1	21	1,61	R$ 36,30	0,86	0,87	1,37
2	31	0,81	R$ 40,23	1,14	1,02	4,41
3	31	0,92	R$ 10,75	0,00	0,04	6,03

FONTE: A autora (2023).

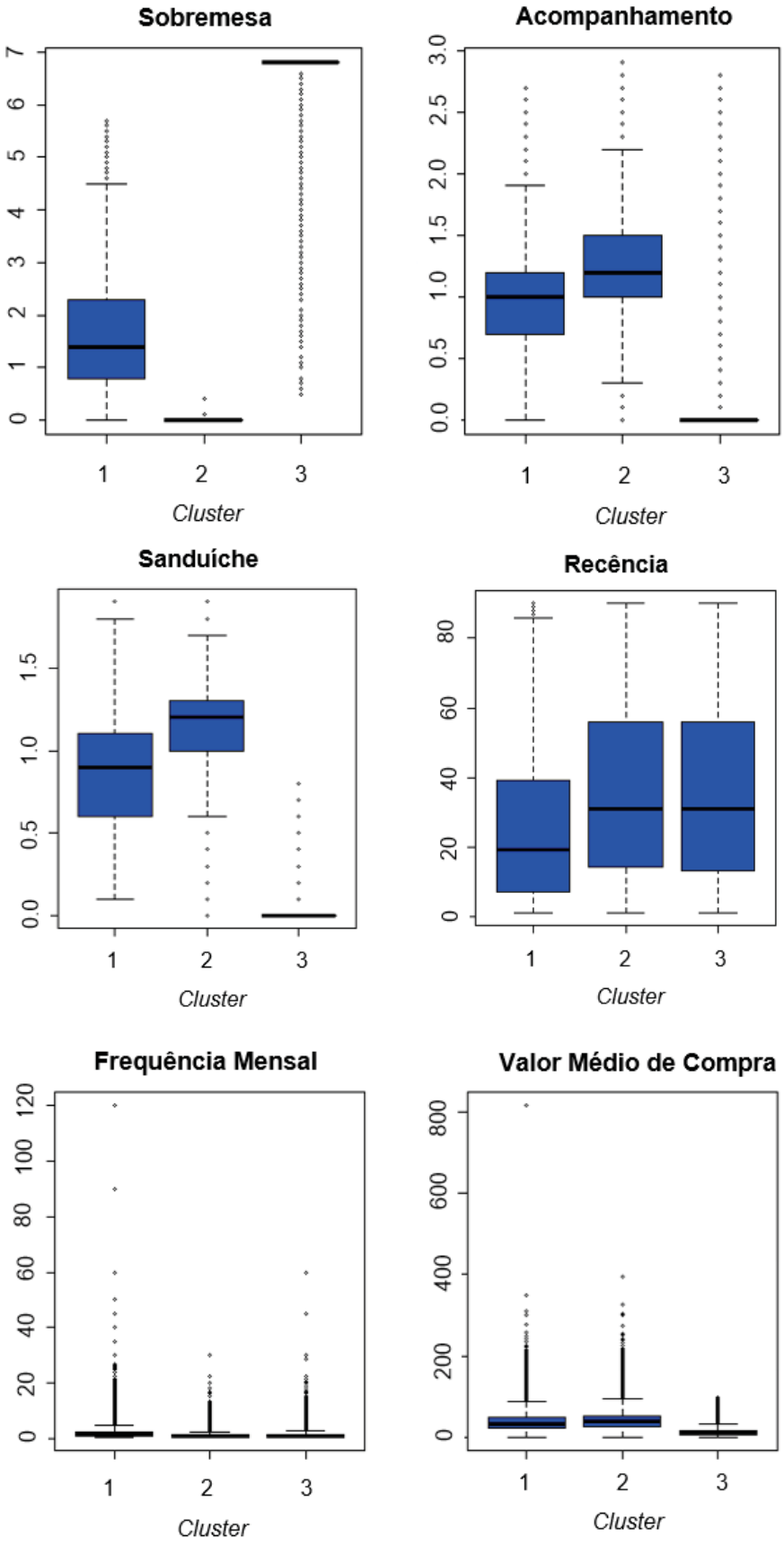
As FIGURA 34 e FIGURA 35 ilustram as medidas estatísticas básicas do conjunto de dados desnormalizados de cada variável após a clusterização.

FIGURA 34 – BOXPLOT POR VARIÁVEL DO AGRUPAMENTO K-*MEANS*



FONTE: A autora (2023).

FIGURA 35 – BOXPLOT POR VARIÁVEL DO AGRUPAMENTO MODELO DE MISTURAS GAUSSIANAS



FONTE: A autora (2023).

Por meio delas (FIGURA 34 e FIGURA 35) é possível identificar que foram feitos agrupamentos diferentes, as variáveis que mais se assemelham são a frequência mensal e o valor médio de compra. Com relação aos grupos formados por meio do algoritmo K-*Means* nota-se que as variáveis que mais diferem os grupos são a recência, valor médio de compra e acompanhamento. Já para o agrupamento do modelo de misturas gaussianas tem-se uma variação maior no que diz respeito as três variáveis de produtos (acompanhamento, sobremesa e sanduíche).

4.4 AVALIAÇÃO DE DESEMPENHO DA CLUSTERIZAÇÃO

Para avaliar o melhor agrupamento, quatro métricas foram utilizadas: Coeficiente de silhueta, MANOVA, índice Davies-Bouldin e índice Calinski-Harabasz.

4.4.1 Coeficiente de silhueta

A determinação do desempenho da clusterização pelo coeficiente de silhueta não foi possível de ser realizada, tanto para o agrupamento obtido pelo algoritmo K-*Means* quanto para o obtido pelo algoritmo GMM, uma vez que ocorreu um erro relacionado a disponibilidade de memória computacional, conforme mensagem apresentada pela FIGURA 36.

FIGURA 36 – ERRO DE MEMÓRIA: COEFICIENTE DE SILHUETA

```
>      silhueta_kmeans<-silhouette(results_kmeans$cluster, dist(data)) # the result closer to 1 implies high clustering quality
Error: cannot allocate vector of size 98.6 Gb
```

FONTE: A autora (2023).

4.4.2 MANOVA

Para os 4 testes estatísticos realizados, o melhor agrupamento é o do Modelo de Mistura Gaussiana (QUADRO 10), visto que para os testes de Pillai, Hotelling-Lawley e Roy os valores do algoritmo GMM são maiores que o do K-*Means* e, para o teste de Wilks, o valor é menor, assim como descrito na seção 2.2.3.5.1.

QUADRO 10 – RESULTADO TESTES ESTATÍSTICOS

Teste estatístico	K-*Means*	Modelo Mistura Gaussiana (GMM)	Referência	Melhor Modelo
Wilks	0,17	0,02	Menor	GMM
Pillai	1,13	1,64	Maior	GMM
Hotelling-Lawley	3,04	13,82	Maior	GMM
Roy	2,28	11,19	Maior	GMM

FONTE: A autora (2023).

4.4.3 Índice Davies-Bouldin

Levando em consideração a avaliação pelo índice Davies-Bouldin, o melhor agrupamento é o do algoritmo K-*Means* (QUADRO 11), visto que quanto menor o valor do índice, mais coeso é o agrupamento realizado.

QUADRO 11 – RESULTADO ÍNDICE DAVIES-BOULDIN

Índice	K-*Means*	Modelo Mistura Gaussiana (GMM)	Referência	Melhor Modelo
Davies-Bouldin	1,23	3,45	Menor	K-*Means*

FONTE: A autora (2023).

4.4.4 Índice Calinski-Harabasz

Já com relação a avaliação do índice Davies-Bouldin, o melhor agrupamento também é o do algoritmo K-*Means* (QUADRO 12), visto que quanto maior o valor do índice, melhor é o agrupamento formado.

QUADRO 12 – RESULTADO TESTES ESTATÍSTICOS

Ìndice	K-*Means*	Modelo Mistura Gaussiana (GMM)	Referência	Melhor Modelo
Calinski-Harabasz	92.038,25	11.888,96	Maior	K-*Means*

FONTE: A autora (2023).

Dos 3 tipos de avaliação de desempenho, considerando que não foi possível a aplicação do coeficiente de silhueta, 2 mostram o K-*Means* como algoritmo com o melhor agrupamento, sendo um deles, o índice Davies-Bouldin, um dos mais

utilizados na literatura. Dessa forma, o algoritmo K-*Means* segue como base para a continuidade do estudo.

4.5 CLASSIFICAÇÃO

Para o treinamento do modelo de aprendizado de máquina supervisionado de classificação usou-se a base de 162.696 clientes já clusterizados pelo algoritmo selecionado K-*Means* como base de treino e teste.

Dessa forma a base de treino é composto por 130.160 clientes, enquanto a base de teste é composta por 32.536 clientes, tanto para a árvore de decisão quanto para o algoritmo da floresta aleatória.

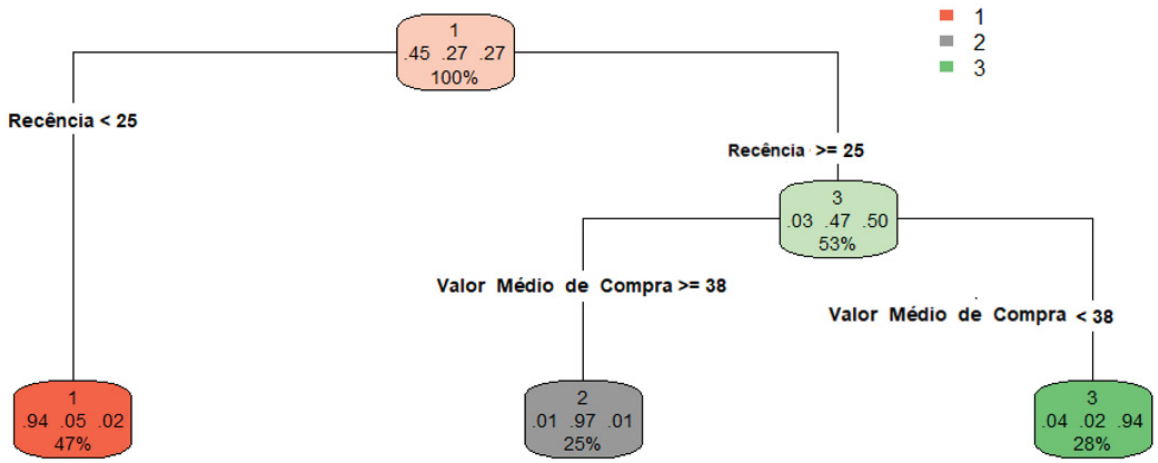
4.6 AVALIAÇÃO DE DESEMPENHO DA CLASSIFICAÇÃO

4.6.1 Árvore de decisão

O algoritmo de classificação árvore de decisão teve uma acurácia de 95,5% no treino, com execução em 6 segundos, e no teste essa acurácia, a partir da matriz de confusão, foi de 94,4%.

A FIGURA 37 ilustra as variáveis mais representativas identificadas pelo modelo para fazer a classificação dos clientes nos três grupos, que são a recência e o valor médio de compra. Cada nó mostra: i) A classe prevista para aquele conjunto de dados, nesse caso sendo cada classe um *cluster*; ii) A probabilidade de cada classe para o conjunto de dados que está no nó e iii) O percentual de observações em cada nó. Para o grupo 1 a única variável avaliada para classificá-lo dessa forma é a recência menor que 25. Já para os grupos 2 e 3, além da recência maior ou igual a 25, outra variável também é usada. Para o grupo 2, entram todos os clientes que possuem valor médio de compra maior ou igual a R$38, já para o grupo 3, são classificados os clientes com valor médio de compra inferior a R$38.

FIGURA 37 – ÁRVORE DE DECISÃO



FONTE: A autora (2023).

4.6.2 Floresta aleatória

O algoritmo de classificação floresta aleatória teve uma acurácia de 98,0% no treino, a execução levou pouco mais de 13 minutos. No teste essa acurácia, a partir da matriz de confusão, foi de 97,9%. Dessa forma, apesar de um tempo maior de execução, o algoritmo selecionado para a classificação dos demais dados de clientes da matriz RFV é o algoritmo da floresta aleatória.

Considerando que a floresta aleatória apresentou melhor resultado para o agrupamento para treino e teste, o restante dos dados, 1.464.264 clientes, também foram classificados por meio da mesma técnica. Lembrando que, a quantidade de clientes corresponde a 90% da base RFV original.

Assim, a base total correspondente a 1.626.960 clientes, foi dividida em 3 *clusters*, sendo 90% via classificação (floresta aleatória) e 10% via clusterização (K-*Means*). Dessa forma, o primeiro grupo contém 732.481 clientes (45%), o segundo grupo é formado por 439.251 clientes (27%) e o último compreende 455.228 clientes (28%).

O QUADRO 13 apresenta a média final de cada *cluster* e a FIGURA 38 ilustra essas médias. Apesar de não ter a ilustração das diferentes árvores de decisão usadas pelo algoritmo, o resultado das médias e medianas indicam que a recência em

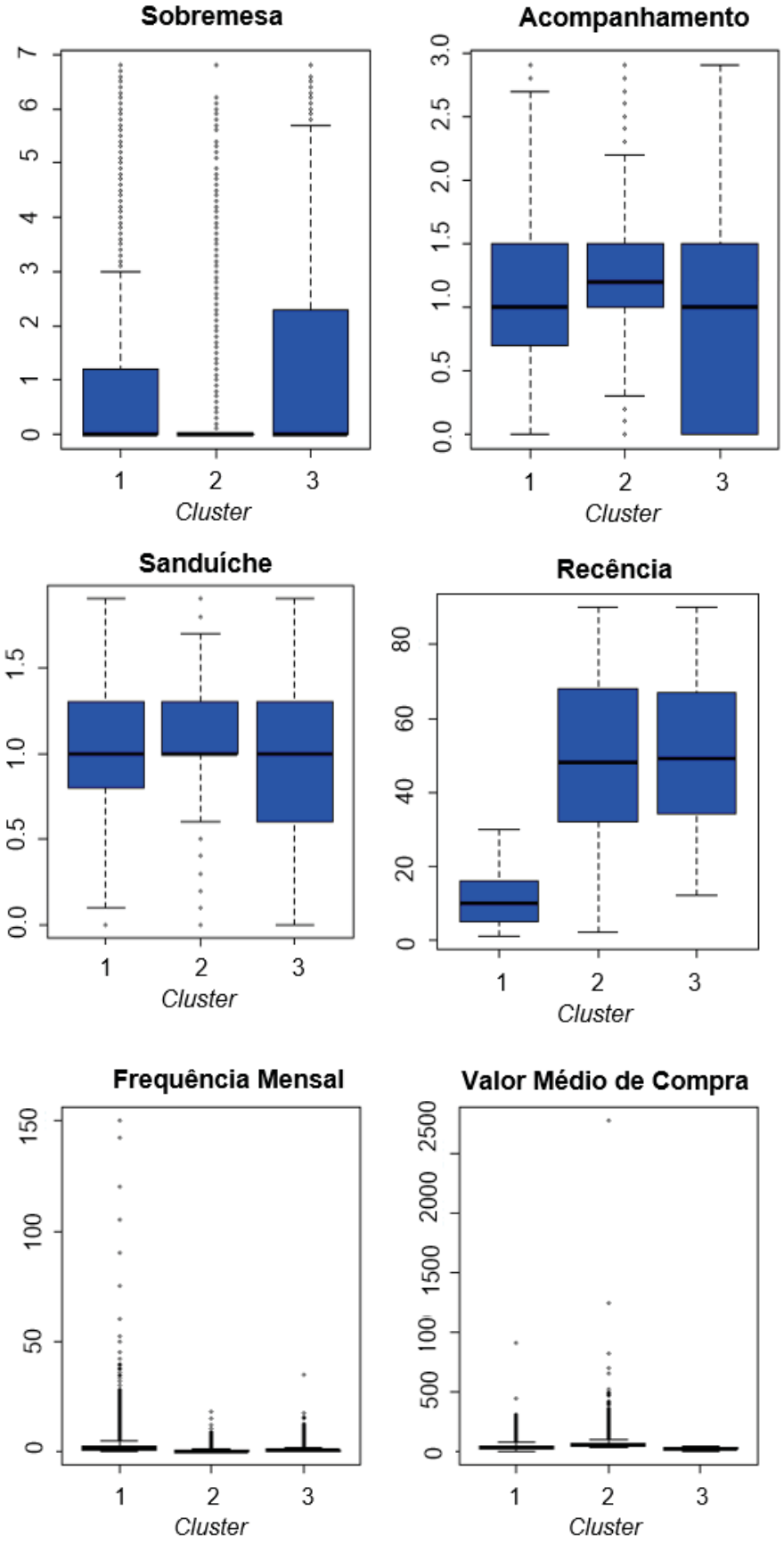
conjunto com o valor médio de compra são as variáveis mais significativas para a construção dos grupos.

QUADRO 13 – RESULTADO DO AGRUPAMENTO FINAL

Cluster	Recência	Frequência Mensal	Valor médio/compra	Sanduíche	Acompanhamento	Sobremesa
1	11	2,21	R$ 35,84	1,00	1,04	0,92
2	50	0,63	R$ 61,71	1,11	1,16	0,30
3	51	0,75	R$ 23,03	0,92	0,88	1,59

FONTE: A autora (2023).

FIGURA 38 – MÉDIAS DO AGRUPAMENTO FINAL



FONTE: A autora (2023).

A partir desses dados, é possível identificar, para cada grupo, comportamentos distintos de compra entre eles, de forma que a companhia possa passar a enviar comunicações de *marketing* personalizadas para os clientes de forma a rentabilizar melhor os envios, dado que uma comunicação que condiz com a jornada do cliente tende a ter uma conversão maior.

Observando o QUADRO 13, nota-se que o primeiro grupo é formado por clientes com uma alta frequência, de aproximadamente 2 visitas ao mês, com um valor médio de compras em linha com a média da companhia (R$34,66). É um *cluster* extremamente importante para companhia, visto que são os clientes que possuem uma frequência alta, dessa forma a recência desse grupo deve ser acompanhada de perto para que se evite o *churn*, ou seja, a perda dos clientes desse grupo.

Já o segundo grupo, por exemplo, contém clientes com a menor frequência dentre todos os grupos, que no período de 3 meses varia entre 1 e 2 visitas, com um valor médio de compra elevado quando comparado com a média da companhia que é de R$34,66. Não parece ser um grupo focado em descontos, assim como também parece ser um grupo em que a ocasião de consumo não é individual, ou seja, podem ser casais, por exemplo. Uma estratégia de comunicação para esse grupo pode ser relacionada a itens *premium* do portfólio de produtos, sem desconto, e que ressaltem os benefícios e diferenciais do produto, ou ainda, a comunicação de ofertas para 2 ou mais pessoas.

O último grupo apresenta clientes com uma frequência mensal próxima de 1 a 2 visitas em 3 meses, assim como o grupo 2, porém com o menor valor médio de compra dos 3 grupos de clientes, o que é baixo para os padrões da companhia. Pode ser um sinal de compras em um canal específico, como por exemplo pelo aplicativo, que possuem um valor médio de compra em linha com o apresentado no grupo, próximo dos R$25, ou ainda de que usam mais as ferramentas de desconto da companhia, como os cupons do aplicativo. São clientes que possuem uma oportunidade de aumentar tanto a frequência quanto o valor médio de compras, dessa forma, incentivos como missões de aceleração de ganho de pontos em compras para troca por produtos grátis pode ser um estímulo interessante de ser comunicado.

A recência dos grupos está bem em linha com a frequência apresentada, já no caso dos produtos, o que se destaca é a categoria de sobremesas para os grupos 2 e 3, o grupo 3 consome mais desse tipo de produto que a média dos demais clientes da companhia e no grupo 2 esse comportamento é inverso, é um grupo que consome menos

que a média da companhia, é possível que uma comunicação de um lançamento de uma nova sobremesa não terá uma conversão tão relevante quanto em comparação ao grupo 3.

5 CONCLUSÕES

O presente estudo teve como objetivo a clusterização e a posterior classificação dos clientes de uma rede de *fast-food* no Brasil, levando em consideração somente os dados de compra dos clientes.

Para tal foram selecionadas as variáveis da matriz RFV e do volume de compra dos itens das 3 principais categorias de produtos da companhia: Sanduíche, acompanhamento e sobremesa. Além disso, 3 métodos de clusterização foram testados: K-*Means*, Método de Ward e Modelo de Misturas Gaussianas, 4 métodos de avaliação dos agrupamentos propostos: Coeficiente de silhueta, análise de variância multivariada (MANOVA), índice Davies-Bouldin e índice Calinski-Harabasz e ainda 2 algoritmos de *machine learning* supervisionados para classificação: Árvore de decisão e Floresta aleatória.

Como resultado da clusterização, com a definição do número de *clusters* ideal igual a 3 a partir do método do cotovelo, o K-*Means* foi o algoritmo com a execução mais rápida, de menos de 1 segundo, em comparação com o modelo de misturas gaussianas, apesar de ser rápido também, com um total de 17 segundos. Foi também o K-*Means* o algoritmo com os melhores resultados nas avaliações de desempenho dos agrupamentos formados, sendo, portanto, o algoritmo escolhido para dar sequência ao presente estudo.

Importante ressaltar que não foi possível computar o algoritmo hierárquico tendo em vista o erro de memória computacional encontrado que não permitiu finalizá-lo. Assim como não foi possível aplicar o coeficiente de silhueta como método de avaliação de desempenho dos algoritmos de clusterização pela mesma razão.

No que diz respeito a classificação, o algoritmo da árvore de decisão obteve uma acurácia de 95,5% no treino e 94,4% no teste, com um tempo de execução de 6 segundos. Já o algoritmo da floresta aleatória no treino teve uma acurácia de 98,0% enquanto no teste se manteve em 97,9% também, o tempo de execução foi mais longo de 13 minutos, porém, dado a diferença no resultado de acurácia do modelo, o algoritmo selecionado para a classificação dos demais clientes nos 3 grupos foi o da floresta aleatória.

Com a base completa de clientes foi possível avaliar a diferença das médias das variáveis selecionadas para o estudo, traçar hipóteses sobre o comportamento de

compra e desenhar pelo menos uma estratégia de comunicação de *marketing* personalizada possível.

Apesar dos resultados serem satisfatórios, ao longo do estudo constatou-se limitações na execução de diferentes funções no *software* R por memória computacional excedida, principalmente em métodos que são difundidos na literatura como o coeficiente de silhueta para avaliação dos agrupamentos realizados, construção da matriz RFV e do algoritmo de clusterização hierárquico. Nesse sentido, para trabalhos futuros, sugere-se o uso de uma máquina com configurações melhores que as apresentadas na metodologia como ferramenta de uso no presente trabalho e a construção de funções ao invés da utilização de funções prontas de bibliotecas do *software* R.

Além disso a aplicação de outros métodos também é visto como oportunidade futura, principalmente na aplicação de outros índices de avaliação de desempenho da clusterização, o cálculo do erro padrão para variabilidade interna dos *clusters*, análise de componentes principais para avaliar as variáveis principais usadas nos agrupamentos, avaliar a variabilidade dos *k-folds*, assim como um outro modelo de clusterização, o *TreeKDE* que é uma combinação de uma árvore de decisão com o modelo baseado em densidade de Kernel.

Vê-se também como trabalho futuro a oportunidade da aplicação prática de comunicações personalizadas para cada um dos grupos formados, onde, a partir de um grupo teste e outro controle é possível avaliar se o *marketing* personalizado de fato tem uma conversão maior e é benéfico em termos estratégicos e de rentabilidade para a companhia para que seja possível desenhar estratégias mais consistentes e construa-se um *roadmap* de evoluções nessa atuação.

Por fim, pode-se ainda validar hipóteses levantadas brevemente no presente estudo, de que o grupo 3, por exemplo, possui um valor médio de compra relativamente mais baixo pois são clientes que compram regularmente com cupons de desconto ou que compram em um canal de compras específico, ou ainda, tomando como exemplo o segundo agrupamento de ter um valor médio de compra elevado por ser um cliente que visita as lojas da marca com a família ou que compra itens considerados como *premium* sem desconto por valorizar o tamanho e sabor diferenciado dos sanduíches dessa subcategoria.

REFERÊNCIAS

ABBAS, O. A. Comparisons between data clustering algorithms.**The International Arab Journal of Information Technology**, Jordânia, v.5, n. 3, p. 320-325, jul 2008.

ABDI F.; ABOLMAKAREM S. Customer Behavior Mining Framework (CBMF) using clustering and classification techniques. **Journal of Industrial Engineering International**, v. 15, p. 1-18, dec 2019. DOI: 10.1007/s40092-018-0285-3.

ABDUL-RAHMAN, S*. et al*. Customer Segmentation and Profiling for Life Insurance using K-Modes Clustering and Decision Tree Classifier. **International Journal of Advanced Computer Science and Applications**, v. 12, n. 9, p. 434-444, 2019. DOI: 10.14569/IJACSA.2021.0120950.

ALESSIA, S. *et al.* Random Forest Algorithm for the Classification of Neuroimaging Data in Alzheimer's Disease: A Systematic Review. **Frontiers in Aging Neuroscience**, v. 9, 2017. DOI: 10.3389/fnagi.2017.00329.

AMINI, M. *et al.* A Cluster-Based Data Balancing Ensemble Classifier for Response Modeling in Bank Direct Marketing. **International Journal of Computational Intelligence and Applications**, v.14, n. 4, 2015. DOI: 10.1142/S1469026815500224.

ANDRADE, M. M. **Introdução à metodologia do trabalho científico**. 2ed. São Paulo: Atlas. P.111, 2001.

ANGUITA, D. *et al.* The ‘K’ in K-fold Cross Validation. **ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning**, Bruges (Belgium), 2012.

ANKERST, M. *et al.* OPTICS: Ordering Points To Identify the Clustering Structure. **ACM SIGMOD international conference on Management of data**. ACM Press, 1999. DOI: 10.1145/304181.304187

ANSHARI, M. *et al.* Customer relationship management and big data enabled: Personalization & customization of services**. Applied Computing and Informatics,** v. 15, n 2. p. 94-101, 2019. DOI: 10.1016/j.aci.2018.05.004.

ARIZA, V. M. P. *et al.* Uso do algoritmo "Floresta Aleatória" na identificação do comportamento da população na busca por serviços de saúde após o início da pandemia do novo coronavírus. **AtoZ – Novas práticas em informação e conhecimento**, 2022.

ARRUDA, H. V. Aplicação da transformação raiz quadrada, na analise da variância de dados experimentais. **Bragantia**, 1959. DOI: https://doi.org/10.1590/S0006-87051959000100034.

ARTHUR, D.; VASSILVITSKII, S. *K-Means*++: The Advantages of Careful Seeding**. Conference: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on**

Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007.

BATISTA, M. A utilização de algoritmos de aprendizado de máquina em problemas de classificação. Dissertação (Mestrado - Programa de Pós-Graduação em Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, p. 109, 2019.

BEHESHTIAN-ARDAKANI, A. *et al*. A novel model for product bundling and direct marketing in e-commerce based on market segmentation. **Decision Science Letters**, v. 7, n. 1, p. 39-54, 2018. DOI: 10.5267/j.dsl.2017.4.005.

BENBASAT, I. *et al.* The case research strategy in studies of information systems. **MIS Quarterly**, v. 11, n. 3, p. 369-386, 1987.

BERFENFELDT, J. Customer Relationship Management. Dissertação Mestrado (Departamento de administração e ciências sociais) — Lulea University of Technology, Suécia, 2010.

BHOLOWALIA, P.; KUMAR, A. EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. **International Journal of Computer Applications**, v. 105, n. 9, 2014.

BRENTARI, E. *et al*. Clustering ranking data in market segmentation: a case study on the Italian McDonald's customers' preferences. **Journal of Applied Statistics**, v. 43, n. 11, p. 1959–1976, 2016.

BRIEL, F. The future of omnichannel retail: A four-stage Delphi study. **Technological Forecasting and Social Change**, v. 132, p. 217-229, 2018.

CHATTOPADHYAY, M. *et al*. Elucidating strategic patterns from target customers using multi-stage RFM analysis. **Journal of Global Scholars of Marketing Science**, 2022. DOI: 10.1080/21639159.2022.2080094.

CHENG C.; CHEN Y. Classifying the segmentation of customer value via RFM model and RS theory. **Expert Systems with Applications**, v. 36, n. 3, p. 4176-4184, 2009. DOI: 10.1016/j.eswa.2008.04.003.

CUI, M. Introduction to the K-Means Clustering Algorithm Based on the Elbow Method. **Accounting, Auditing and Finance (2020)** 1: 5-8, Clausius Scientific Press, Canada, 2020. DOI: 10.23977/accaf.2020.010102.

DAVIES, D. L.; BOULDIN, D. W. A cluster separation measure. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 1, n. 2, p. 224-227, 1979.

DEHGHANIZADEH, M. *et al*. LDcFR: A new model to determine value of airline passengers. **Tourism and Hospitality Research**, v. 18, n. 3, p. 357-366, 2018. DOI: 10.1177/1467358416663821.

DENYER, D.; TRANFIELD, D. **Producing a systematic review**. In D. A. Buchanan & A. Bryman (Eds.), The SAGE handbook of organizational research methods (p. 671–689). London: Sage Publications Ltd. 2009.

DE MEDEIROS, J. F. *et al*. Success factors for environmentally sustainable product innovation: A systematic literature review. **J. Clean. Prod**, v. 65, p. 76–86, 2014. DOI: 10.1016/j.jclepro.2013.08.035.

DUTRA, R.M. O. *et al*. O Método Ward de Agrupamento de Dados e sua Aplicação em Associação com os Mapas Auto-Organizáveis de Kohonen. **Workcomp Sul**, Florianópolis, 2004.

ESTER, M. *et al*. Incremental Clustering for Mining in a Data Warehousing Environment. **Proceedings 24th Int. Conf. Very Large Data Bases, VLDB**, p. 323-333, 1998.

FAN, L. Research on Precision Marketing Strategy of Commercial Consumer Products Based on Big Data Mining of Customer Consumption. **Journal of the Institution of Engineers (India): Series C**, v. 104, n. 1, p. 163-168, 2023. DOI: 10.1007/s40032-022-00908-7.

FARUGHI H. *et al*. An empirical study on customer risk management in banking industry: Applying k-means and RFM methods (evidence from two Iranian private banks). **International Journal of Risk Assessment and Management**, v. 19, n. 4, p. 315-330, 2016. DOI: 10.1504/IJRAM.2016.079610.

GARZA-REYES, J. A. Green lean and the need for Six Sigma. **Int. J. Lean Six Sigma**, v. 6, p. 226–248, 2015. DOI: 10.1108/IJLSS-04-2014-0010.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4a. ed. São Paulo: Atlas, 2002.

GREENBERG, P. CRM at the speed of light: Capturing and keepin gustomers in Internet real time. **[S.l.]: Elsevier**, 2001.

GUERREIRO, M. Análise de Métodos de Agrupamento de Dados para Detecção de Anomalias na Precificação e Categorização de Peças da Indústria Automotiva**.** 2021. 84 p. Dissertação (Mestrado) – Ciência da Computação da Universidade Tecnológica Federal do Paraná (UTFPR), Ponta Grossa.

GUHA, S., R. *et al.* CURE: an efficient clustering algorithm for large databases. **ACM Sigmod International Conference on Management of Data**, p. 73–84, 1998.

HAIR, J. F. *et al*. **Análise multivariada de dados**. Trad. Adonai S. Sant'Anna e Anselmo C. Neto. 5 ed. Porto Alegre: Bookman, 2005.

HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques.** San Diego: Academic Press, 2001.

HANAYSHA, J. R.; MEHMOOD, K. An Exploration of the Effect of Customer Relationship Management on Organizational Performance in the Banking Sector.

International Journal of Customer Relationship Marketing and Management, v. 13, n. 1, 2022. DOI: 10.4018/IJCRMM.2022010101.

HASTIE, T. *et al*. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. New York: Springer-Verlag, 2009.

HOSSEINI Z.; MOHAMMADZADEH M. Knowledge discovery from patients' behavior via clustering-classification algorithms based on weighted eRFM and CLV model: An empirical study in public health care services. **Iranian Journal of Pharmaceutical Research**, v. 15, n. 1, p. 355-367, 2016.

HRUSCHKA, E. R.; EBECKEN, N. F. F. A Genetic algorithm for cluster analysis. **IEEE Transactions on Evolutionary Computation**, 2001.

HU, D. *et al*. Electric vehicle user classification and value discovery based on charging big data. **Energy**, v. 249, 2022. DOI: 10.1016/j.energy.2022.123698.

JUNIOR, W. J. A. Métodos de Otimização Hiperparamétrica: Um Estudo Comparativo Utilizando Árvores de Decisão e Florestas Aleatórias na Classificação Binária. Dissertação de Mestrado (Programa de Pós-Graduação em Engenharia Elétrica) – Universidade Federal de Minas Gerais, p. 82, 2018.

KANE, G. C. Digital maturity, not digital transformation. MIT Sloan Management Review, 2017. Disponível em: https://sloanreview.mit.edu/article/digital-maturity-not-digital-transformation/. Acesso em: 13 de Junho de 2021.

KAUFMAN, L.; ROUSSEEUW, P. J. **Finding groups in data – An introduction to cluster analysis.** Wiley, NY, 1990.

KHALILI-DAMGHANI, K. *et al*. Hybrid soft computing approach based on clustering, rule mining, and decision tree analysis for customer segmentation problem: Real case of customer-centric industries. **Applied Soft Computing**, v. 73, p. 816-828, 2018. DOI: 10.1016/j.asoc.2018.09.001

LAURETTO, M. Análise de Agrupamentos (Clusters). 2017. Disponível em: http://www.each.usp.br/lauretto/cursoR2017/04-AnaliseCluster.pdf. Acesso em: 03/04/2022.

LI, M. *et al*. Adherence predictor variables in AIDS patients: an empirical study using the data mining-based RFM model. **AIDS Research and Therapy**, v. 18, n. 1, 2021. DOI: 10.1186/s12981-020-00326-8.

LIAO, J. *et al*. Multi-Behavior RFM Model Based on Improved SOM Neural Network Algorithm for Customer Segmentation. **IEEE Access**, v. 10, p. 122.501 – 122.512, 2022. DOI: 10.1109/ACCESS.2022.3223361.

LIKAS, A. *et al*. The global K-Means clustering algorithm. **Pattern Recognition**, v. 36, n. 2, p. 451 – 461, fev 2003. DOI: 10.1016/S0031-3203(02)00060-2.

LUDWIG, G. Técnicas de clustering baseadas em otimização de outras funções objetivo. Instituto de Matemática, Estatística e Computação Científica – Unicamp, Campinas – SP, p. 21, 2021.

MACLURE, K. *et al.* Reviewing the literature, how systematic is systematic? **Int J Clin Pharm**, March 2016. DOI: 10.1007/s11096-016-0288-3.

MADHULATHA, T. S. An overview on clustering methods. **IOSR Journal of Engineering**, v. 2, n. 4, p. 719-725, 2012.

MARTINS, H. *et al*. Transformações digitais no Brasil: insights sobre o nível de maturidade digital das empresas no país. Mckinsey, 2019. Disponível em: <https://www.mckinsey.com/br/our-insights/transformacoes-digitais-no-brasil#>. Acesso em: 03 de Fevereiro de 2021.

MIGLAUTSCH, J. R. Thoughts on RFM scoring. **Journal of Database Marketing**, v. 8, n. 1, p. 67–72, 2000. DOI: https://doi.org/10.1057/palgrave.jdm.3240019.

NIMBALKAR, D. Data Mining using RFM Analysis. **International Journal of Scientific & Engineering Research**, v. 4, n. 12, dez, 2013.

OLIVEIRA, T. Segmentação de Imagens Coloridas Utilizando Técnicas de Agrupamento de Dados. 2008. 112 p. Dissertação (Mestrado) – Ciências de Computação e Matemática Computacional. Instituto de Ciências Matemáticas e de Computação – ICMC/USP, São Carlos.

PADILLA, G *et al.* Comparison of Several Clustering Methods in Grouping Kale Landraces. **J. Amer. Soc. Hort. Sci.** 132 (3), p. 387-395, 2007.

PALACIO-NINO, J-O; BERZAL, F. Evaluation Metrics for Unsupervised Learning Algorithms. **ArXiv:1905.05667**, 2019. DOI: doi.org/10.48550/arXiv.1905.05667.

PALMA, L. F. Agrupamento de dados: k-médias**.** Trabalho de conclusão de curso de ciências exatas e tecnológicas da UFRB, 2018.

PATEL, E.; KUSHWAHA, D. S. **Clustering Cloud Workloads: K-Means vs Gaussian Mixture Model**. Procedia Computer Science, v. 171, p. 158-167, 2020.

PATRO, V. M.; PATRA, M. R. Augmenting Weighted Average with Confusion Matrix to Enhance Classification Accuracy. **Transactions on Machine Learning and Artificial Intelligence**, v. 2, n. 4; p. 77-91, 2014.

PAYNE, A.; FROW, P. A strategic framework for customer relationship management. **Journal of marketing**, SAGE Publications Sage CA: Los Angeles, CA, v. 69, n. 4, p.167–176, 2005.

PERIŠIC, A.; PAHOR, M. Clustering mixed-type player behavior data for churn prediction in mobile games. **Central European Journal of Operations Research**, v. 31, n. 1, p. 165-190, 2023. DOI: 10.1007/s10100-022-00802-8.

POPAT, S. K. *et al*. Review and Comparative Study of Clustering Techniques. **International Journal of Computer Science and Information Technologies**, v. 5 n. 1, 2014, p. 805-812.

PORTELA, N. M. **Modelo de Mistura de Gaussianas Fuzzy Contextual**. Tese de doutorado, Programa de Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, p. 132, 2015.

PRADESYAH, R.; SAPUTRI, W. Customer Relationship Management in maintening and increasing the number of customers. **International Seminar of Islamic Studies**, v. 3, n. 1, 2022.

PRADO, T. C. Segmentação de Imagens Coloridas Utilizando Técnicas de Agrupamento de Dados. TCC (graduação) - Universidade Federal de Santa Catarina. Centro Tecnológico. Curso de Ciências da Computação. Florianópolis, Santa Catarina, 2008.

RACHID, A. *et al*. Clustering prediction techniques in defining and predicting customers defection: The case of e-commerce context. **International Journal of Electrical and Computer Engineering**, v.8, n. 4, p. 2367-2383, 2018. DOI: 10.11591/ijece.v8i4.pp2367-2383.

REGONDA, S. K. *et al*. Using climate regionalization to understand Climate Forecast System Version 2 (CFSv2) precipitation performance for the Conterminous United States (CONUS). **Geophysical Research Letters**, v. 43, n. 12, 2016. DOI: 10.1002/2016GL069150.

RELVAS, C. E. M. Agrupamento baseado em modelos de mistura de gaussianas com covariáveis. Tese de doutorado, Instituto de Matemática e Estatística da Universidade de São Paulo, p. 75, 2020.

RODRIGUES, F. Métodos de agrupamento na análise de dados de expressão gênica. Tese de mestrado, Centro de ciências exatas e de tecnologia da Universidade Federal de São Carlos, p.95, 2009.

SCRUCCA, L. *et al.* mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models. **R J.**, v. 8, n. 1, p.289-317, ago 2016.

SEEBREGTS, C. Utilização De Algoritmo K-Means Para Definição De Domínios Litológicos. Monografia de Conclusão de Curso, Engenharia de Minas da Universidade Federal de Ouro Preto – MG, p. 72, 2022.

SHAIKHINA, T. *et al*. Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation. **Biomedical Signal Processing and Control**, v. 52, p. 456–462, 2019.

SHETH, J. N. *et al*. The Antecedents and Consequences of Customer-Centric Marketing. **Journal of the Academy of Marketing Science**, v. 28, n. 1, p. 55–66, 2000. DOI: 10.1177/0092070300281006.

SILVA, E. L.; MENEZES, E. M. (2005). **Metodologia da Pesquisa e Elaboração de Dissertação**. 4ª ed., UFSC, Florianópolis, SC, 138p, 2005.

SILVA, J. P. D. Algoritmos de Classificação baseados em Análise Formal de Conceitos. Dissertação (Programa de Pós-Graduação em Ciência Da Computação), Instituto de Ciências Exatas, Universidade Federal de Minas Gerais, p.115, 2007.

SILVA, S. S. Segmentação de Imagens Utilizando Combinação de Modelos de Misturas Gaussianas. Dissertação de Mestrado, Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, p. 80, 2014.

SINGH, A; RUMANTIR, G. Two-tiered clustering classification experiments for market segmentation of EFTPOS retailers. **Australasian Journal of Information Systems,** v. 19, p. 117-132, 2015.

SIVA; R. K.B *et al.* Classification and identification of loyal customers using machine learning. **Journal of Advanced Research In Dynamical and Control Systems**, 2020.

SORENSON, T. A Method of Establishing Groups of Equal Amplitudes in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons. **Kongelige Danske Videnskabernes Selskab**, Biologiske Skrifter, vol. 5, p. 1-34, 1948.

SUN, G. *et al*. Using improved RFM model to classify consumer in big data environment. **International Journal of Embedded Systems**, v. 14, n. 1, p. 54-64, 2021. DOI: 10.1504/IJES.2021.111976.

SZMRECSANYI, B. Grammatical Variation in British English Dialects: A Study in Corpus-Based Dialectometry. **Cambridge University Press**, 2012.

TANG, Q. *et al*. A hybrid classification model for churn prediction based on customer clustering. **Journal of Intelligent and Fuzzy Systems**, v. 39, n. 1, p. 69-80, 2020. DOI: 10.3233/JIFS-190677.

VARELLA, C. Análise Multivariada Aplicada As Ciências Agrárias. Pós-Graduação em Agronomia Ciência do Solo: CPGA – CS. Universidade Rural do Rio de Janeiro, p.6, 2007. Disponível em: http://www.ufrrj.br/institutos/it/deng/varella/Downloads/multivariada%20aplicada%20as%20ciencias%20agrarias/Apresenta/regressao%20linear%20multipla.pdf. Acesso em: 06/03/2023.

WANG, C. Efficient customer segmentation in digital marketing using deep learning with swarm intelligence approach. **Information Processing & Management**, v. 59, n. 6, 2022. DOI: 10.1016/j.ipm.2022.103085.

WEI, J. *et al*. A review of the application of RFM model. **African Journal of Business Management**, Taiwan, v. 4, n. 19, p. 4199-4206, dez 2010.

WU, J. *et al.* User Value Identification Based on Improved RFM Model and *K-Means*++ Algorithm for Complex Data Analysis. **Wireless Communications and Mobile Computing**, 2021. DOI: 10.1155/2021/9982484.

ZOERAM, A.; MAZIDI, A. A New Approach for Customer Clustering by Integrating the LRFM Model and Fuzzy Inference System. **Iranian Journal of Management Studies**, v. 11, n. 2, p. 351-378, 2018. DOI: 10.22059/ijms.2018.242528.672839.

APÊNDICE A – *SCRIPT* DESENVOLVIDO EM R

```
#IMPORTANDO OS DADOS

  library("readxl")

  #Cadastro e Status
    cadastro_status_1<-read_excel("011220_010421.xlsx") #Dezembro/20 e Janeiro a Março/21
    cadastro status 2<-read excel("010421 010521.xlsx") #Abril/21
    cadastro_status_3<-read_excel("010521_010621.xlsx") #Maio/21
    cadastro_status_4<-read_excel("010621_010721.xlsx") #Junho/21
    cadastro status 5<-read excel("010721 010821.xlsx") #Julho/21
    cadastro_status_6<-read_excel("010821_010921.xlsx") #Agosto/21
    cadastro status 7<-read excel("010921 011021.xlsx") #Setembro/21
    cadastro_status_8<-read_excel("011021_011121.xlsx") #Outubro/21
    cadastro status 9<-read excel("011121 011221.xlsx") #Novembro/21
    cadastro_status_10<-read_excel("011221_010122.xlsx") #Dezembro/21
    cadastro status 11<-read excel("010122 010222.xlsx") #Janeiro/22
    cadastro_status_12<-read_excel("010222_010322.xlsx") #Fevereiro/22
    cadastro status 13<-read excel("010322 010422.xlsx") #Março/22
    #cadastro_status_14<-read_excel("010422_010522.xlsx") #Abril/22
    #cadastro status 15<-read excel("010522 010622.xlsx") #Maio/22
    #cadastro_status_16<-read_excel("010622_010722.xlsx") #Junho/22
    #cadastro status 17<-read excel("010722 010822.xlsx") #Julho/22

    cadastro_status_total <- rbind (cadastro_status_1,
                                    cadastro status 2,
                                    cadastro_status_3,
                                    cadastro status 4,
                                    cadastro_status_5,
                                    cadastro status 6,
                                    cadastro_status_7,
                                    cadastro status 8,
                                    cadastro_status_9,
                                    cadastro status 10,
                                    cadastro_status_11,
                                    cadastro status 12,
                                    cadastro_status_13
                                    #,cadastro status 14,
                                    #cadastro_status_15,
                                    #cadastro_status_16,
                                    #cadastro status 17
                                    ) #Criando uma única base

    cadastro_status_total<-cadastro_status_total[,c(-1)] #Retirando coluna desnecessária:
[Fielo PLT Member]

    cadastro status total ativo NA<-
subset(cadastro_status_total,cadastro_status_total$FieloPLT__Status__c == 'Active') #Pegando
somente os usuários ativos

    cadastro status total ativo <- na.omit(cadastro status total ativo NA) #Limpeza da base
```

```
    #BASE FINAL DE CADASTROS
    cadastro status total ativo$FieloPLT  JoinDate  c<-
as.Date(cadastro_status_total_ativo$FieloPLT__JoinDate__c, format = "%Y-%m-%d") #Formatando
as datas para tipo date

#---------------------------------------------------------------------

  #Compras
    compras_1<-read_excel("010122_150122.xlsx") #Janeiro/22
    compras 2<-read excel("160122 310122.xlsx") #Janeiro/22
    compras_3<-read_excel("010222_150222.xlsx") #Fevereiro/22
    compras 4<-read excel("160222 220222.xlsx") #Fevereiro/22
    compras_5<-read_excel("230222_280222.xlsx") #Fevereiro/22
    compras_6<-read_excel("010322_100322.xlsx") #Março/22
    compras 7<-read excel("110322 200322.xlsx") #Março/22
    compras 8<-read excel("210322 310322.xlsx") #Março/22
    #compras 9<-read excel("010422 050422.xlsx") #Abril/22
    #compras_10<-read_excel("060422_150422.xlsx") #Abril/22
    #compras_11<-read_excel("160422_230422.xlsx") #Abril/22
    #compras_12<-read_excel("240422_300422.xlsx") #Abril/22
    #compras 13<-read excel("010522 050522.xlsx") #Maio/22
    #compras_14<-read_excel("060522_120522.xlsx") #Maio/22
    #compras 15<-read excel("130522 210522.xlsx") #Maio/22
    #compras_16<-read_excel("220522_260522.xlsx") #Maio/22
    #compras 17<-read excel("270522 310522.xlsx") #Maio/22
    #compras_18<-read_excel("010622_060622.xlsx") #Junho/22
    #compras 19<-read excel("070622 120622.xlsx") #Junho/22
    #compras_20<-read_excel("130622_180622.xlsx") #Junho/22
    #compras 21<-read excel("190622 240622.xlsx") #Junho/22
    #compras 22<-read excel("250622 300622.xlsx") #Junho/22
    #compras 23<-read excel("010722 060722.xlsx") #Julho/22
    #compras_24<-read_excel("070722_130722.xlsx") #Julho/22
    #compras_25<-read_excel("140722_200722.xlsx") #Julho/22
    #compras_26<-read_excel("210722_270722.xlsx") #Julho/22
    #compras_27<-read_excel("280722_310722.xlsx") #Julho/22

    #As bases já estão limpas (alguns campos não estavam com o customer_id diretamente, mas
sim: "{"customer id":"2ab92473-4a86-4cee-8f35-
8476f0ea21d5","name":"Emilio","cpf":"xxxxxxxxx"}")

    compras_jan <- rbind (compras_1, compras_2)
    compras fev <- rbind (compras 3, compras 4, compras 5)
    compras_mar <- rbind (compras_6, compras_7, compras_8)
    #compras abr <- rbind (compras 9, compras 10, compras 11, compras 12)
    #compras mai <- rbind (compras 13, compras 14, compras 15, compras 16, compras 17)
    #compras jun <- rbind (compras 18, compras 19, compras 20, compras 21, compras 22)
    #compras_jul <- rbind (compras_23, compras_24, compras_25, compras_26, compras_27)


    compras_total_NA <- rbind (compras_jan, compras_fev, compras_mar)#, compras_abr,
compras mai, compras jun, compras jul) #Criando uma única base

    compras total<-compras total NA[,c(-2, -3)] #Retirando coluna desnecessária: Customer id
antes do ajuste e NFe code
```

```
    #Limpeza da base
    #compras total 0<-na.omit(compras total) #Limpeza da base (Se não tirar o que poderiam
ser 3 transações eu vou considerar como 1, se tirar eu deixo de considerar transações dos
clientes)

    #compras total <- subset(compras total 0, compras total 0$`VENDA BRUTA - LOYALTY
TOTAL`!=0.0) #Retirando transações zeradas (se tirar não considera essa visita, mas se não
tirar afeta o valor gasto)


    #BASE FINAL DE COMPRAS
    compras total$Data<-as.Date(compras total$Data, format = "%Y-%m-%d") #Formatando as
datas para tipo date

#---------------------------------------------------------------------

  #Categorias

    library("csv")

    cat jan <- read.csv("Share jan22.csv")
    #share_jan <- data.frame(categoria = c('Sobremesa', 'Acompanhamentos', 'Sanduiches'),
volume = c(3976738, 9492855, 14683951))
    #share_jan$share <- share_jan$volume/sum(share_jan$volume)
    #share jan$share <- round(share jan$share,4)

    cat fev <- read.csv("Share fev22.csv")
    #share_fev <- data.frame(categoria = c('Sobremesa', 'Acompanhamentos', 'Sanduiches'),
volume = c(3832631, 8443851, 12760971))
    #share fev$share <- share fev$volume/sum(share fev$volume)
    #share fev$share <- round(share fev$share,4)

    cat_mar <- read.csv("Share_mar22.csv")
    #share_mar <- data.frame(categoria = c('Sobremesa', 'Acompanhamentos', 'Sanduiches'),
volume = c(4146773, 9555491, 14215677))
    #share mar$share <- share mar$volume/sum(share mar$volume)
    #share_mar$share <- round(share_mar$share,4)

    #cat_abr <- read.csv("Share_abr22.csv")
    #share abr <- data.frame(categoria = c('Sobremesa', 'Acompanhamentos', 'Sanduiches'),
volume = c(3899120, 9762838, 14728140))
    #share abr$share <- share abr$volume/sum(share abr$volume)
    #share_abr$share <- round(share_abr$share,4)

    #cat mai <- read.csv("Share mai22.csv")
    #share mai <- data.frame(categoria = c('Sobremesa', 'Acompanhamentos', 'Sanduiches'),
volume = c(3968062, 10077991, 15120644))
    #share_mai$share <- share_mai$volume/sum(share_mai$volume)
    #share_mai$share <- round(share_mai$share,4)

    #cat jun <- read.csv("Share jun22.csv")
    #share_jun <- data.frame(categoria = c('Sobremesa', 'Acompanhamentos', 'Sanduiches'),
volume = c(3989827, 10064322, 14599097))
    #share_jun$share <- share_jun$volume/sum(share_jun$volume)
    #share jun$share <- round(share jun$share,4)

    #cat_jul <- read.csv("Share_jul22.csv")
```

```
    #share_jul <- data.frame(categoria = c('Sobremesa', 'Acompanhamentos', 'Sanduiches'),
volume = c(4896274, 10897318, 15788019))
    #share_jul$share <- share_jul$volume/sum(share_jul$volume)
    #share jul$share <- round(share jul$share,4)

    base cat <- rbind (cat jan, cat fev, cat mar)#, cat abr, cat mai, cat jun, cat jul)
#base única

    #library(plyr)
    #library(dplyr)
    #library(tidyr)

    #share <- data.frame(categoria = c('Sobremesa', 'Acompanhamentos', 'Sanduiches'), volume
= c(28709065, 68294666, 101896499)) #Criando tabela com quantidade de compras dos items
total
    share <- data.frame(categoria = c('Sobremesa', 'Acompanhamentos', 'Sanduiches'), volume
= c(11956142, 27492197, 41660599)) #Criando tabela com quantidade de compras dos items total
    share$share <- share$volume/sum(share$volume) #Criando o share
    share$share <- round(share$share,4)
    share <- share[,-2] #Tirando o volume
    share <- as.data.frame(t(share)) #transpondo a tabela
    #share <- dplyr::rename(share, Sobremesa = V1, Acompanhamentos = V2, Sanduiche = V3)
#Renomeando as colunas
    library(gdata)
    share <-rename.vars(share, c("V1","V2", "V3"),
c("Sobremesa","Acompanhamentos","Sanduiche"))
    share <- share [-1,] #Tirando a linha que seria do nome das colunas
    share$Sobremesa <- as.numeric(share$Sobremesa) #Transformando o share em dado numérico
para fazer conta depois
    share$Acompanhamentos <- as.numeric(share$Acompanhamentos)
    share$Sanduiche <- as.numeric(share$Sanduiche)

    cat_agrup <- with(base_cat,table(customer_loyalty_id, node5_product_segment)) #Base
agrupada com apenas 1 linhas por cliente
    cat <- as.data.frame.matrix(cat_agrup) #Transformando a tabela em dataframe
    cat$customer id <- rownames(cat agrup) #Ajustando para ter uma coluna de id (variável)

    cat$total <- rowSums(cat[,1:3]) #Somando o total de itens comprados para fazer o share
depois
    cat$share sobremesa <- round(cat$SOBREMESA/cat$total,4) #Share das categorias (%)
    cat$share_acompanhamento <- round(cat$ACOMPANHAMENTOS/cat$total,4)
    cat$share sanduiche <- round(cat$SANDUICHE/cat$total,4)
    #cat$total2 <- rowSums(cat[,5:7]) #Double check

    cat$sobremesa normalizado <- round(cat$share sobremesa/share$Sobremesa,1) #normalização
do share -> 1 consumo igual a média da cia, <1 abaixo da média, >1 acima da média da cia
    cat$acompanhamento_normalizado <-
round(cat$share_acompanhamento/share$Acompanhamentos,1)
    cat$sanduiche_normalizado <- round(cat$share_sanduiche/share$Sanduiche,1)


    #BASE FINAL DE CATEGORIAS
    cat vf <- cat[,c(-1,-2,-3,-5,-6,-7,-8)]



#---------------------------------------------------------------------
```

```
#Matriz RFV

    orders <- data.frame(customer_id=compras_total$`ID Customer Unificado_Ajustado`,
                                 order date=compras total$Data,
                                 revenue=compras_total$`VENDA BRUTA - LOYALTY TOTAL`)
#Criando uma base somente com as informações necessárias e na ordem correta

    #analysis date <- lubridate::as date("2022-08-01", tz = "UTC") #Data para referência da
recência
    analysis date <- lubridate::as date("2022-04-01", tz = "UTC") #Data para referência da
recência

    library(rfm)
    rfm_result <- rfm_table_order(orders, customer_id, order_date, revenue, analysis_date)
#Matriz RFV
    rfm result final<-as.data.frame(rfm result$rfm) #Convertendo a tabela em dataframe

    rfm_result_final<-rfm_result_final[,c(-6,-7,-8,-9)] #Deixando as colunas necessárias
somente (escala da RFV não será usada)

    cadastro status total ativo<-dplyr::rename(cadastro status total ativo,
c("customer_id"="V_ExternalId__c")) #Renomear para fazer o inner join das colunas de id
    rfm ativos<-merge(x=rfm result final,y=cadastro status total ativo,by = "customer id")
#Considerando somente usuários não suspeitos de fraude
    rfm ativos <- rfm ativos[,-7] #Tirando coluna desnecessária: Status ativo

    #rfm ativos$Fim periodo<-lubridate::as date("2022-08-01", tz = "UTC") #Data de
referencia para o cálculo de dias para normalizar a frequencia
    rfm ativos$Fim periodo<-lubridate::as date("2022-04-01", tz = "UTC") #Data de referencia
para o cálculo de dias para normalizar a frequencia
    rfm ativos$dias<-rfm ativos$Fim periodo-rfm ativos$FieloPLT  JoinDate  c #Calculando os
dias entre o cadastro e o fim do período de compras analisado para normalizar a frequencia
    datainicio<-lubridate::as_date("2022-01-01", tz = "UTC") #Data de início do período de
compras avaliado
    #datafim<-lubridate::as_date("2022-08- ", tz = "UTC") #Data final do período de compras
avaliado
    datafim<-lubridate::as_date("2022-04-01", tz = "UTC") #Data final do período de compras
avaliado
    dias<-datafim-datainicio #dias do período de compras
    rfm ativos$dias[rfm ativos$dias > dias] <- dias #Se o usuário se cadastrou antes do
período de compras analisado ele considera o período completo do pe´riodo de compras em
análise
    rfm_ativos$dias<-as.numeric(rfm_ativos$dias) #Transformando em número para fazer a
normalizaçao depois
    rfm ativos$freq mensal<-(rfm ativos$transaction count/(rfm ativos$dias/30)) #Frequencia
mensal normalizada
    rfm_ativos$freq_mensal<-round(rfm_ativos$freq_mensal,2) #Arredondando para 2 casas
decimais

    rfm_ativos$spend_mensal<-(rfm_ativos$amount/(rfm_ativos$dias/30)) #Spend mensal
normalizado
    rfm_ativos$spend_mensal<-round(rfm_ativos$spend_mensal,2) #Arredondando para 2 casas
decimais

    rfm ativos$TM<-rfm ativos$amount/rfm ativos$transaction count #TM
    rfm_ativos$TM<-round(rfm_ativos$TM,2)
```

```
    rfm_ativos<-merge(x=rfm_ativos,y=cat_vf,by = "customer_id") #Juntando o share da
categoria com a rfv

    rfm ativos 0<-na.omit(rfm ativos) #Limpeza da base: valores NA
    rfm_ativos_0$total<-rowSums(rfm_ativos_0[,12:14])

    rfm_ativos <- subset(rfm_ativos_0, rfm_ativos_0$total!=0.00) #Retirando share total
zerado

    qnt clientes<-nrow(rfm ativos) #Quantidade de clientes

    #BASE FINAL RFV
    rfm_ativos<- rfm_ativos[,c(-2,-6,-7,-8, -15)] #Tirando colunaS desnecessáriaS (dias,
data final do periodo, total, data de cadastro, data última compra)
    write.csv(rfm ativos, 'rfm ativos.csv')

#---------------------------------------------------------------------

#NORMALIZAÇÃO DOS DADOS:

      rfm ativos normalizando<-read.csv("rfm ativos.csv")
      rfm_ativos_normalizando<-rfm_ativos_normalizando[,-1]

      rfm_ativos_normalizando$recency_days_sqrt<-sqrt(rfm_ativos_normalizando$recency_days)
      rfm ativos normalizando$TM sqrt<-sqrt(rfm ativos normalizando$TM)
      rfm_ativos_normalizando$freq_mensal_sqrt<-sqrt(rfm_ativos_normalizando$freq_mensal)
      rfm ativos normalizando$sanduiche normalizado sqrt<-
sqrt(rfm_ativos_normalizando$sanduiche_normalizado)
      rfm ativos normalizando$sobremesa normalizado sqrt<-
sqrt(rfm ativos normalizando$sobremesa normalizado)
      rfm ativos normalizando$acompanhamento normalizado sqrt<-
sqrt(rfm_ativos_normalizando$acompanhamento_normalizado)
      rfm_ativos <-  rfm_ativos_normalizando [,c(-3,-4,-6)]

      par(mfrow=c(1,3))
      boxplot(rfm ativos$recency days sqrt, col="blue", main="Recência",notch=T, pch=19)
      boxplot(rfm_ativos$freq_mensal_sqrt, col="blue", main="Frequência Mensal")
      boxplot(rfm ativos$TM sqrt, col="blue", main="Valor Médio de Compra (R$)")

      par(mfrow=c(1,3))
      boxplot(rfm_ativos$sobremesa_normalizado_sqrt, col="blue", main="Sobremesa")
      boxplot(rfm ativos$acompanhamento normalizado sqrt, col="blue", main="Acompanhamento")
      boxplot(rfm_ativos$sanduiche_normalizado_sqrt, col="blue", main="Sanduiche")

dens <- density(rfm ativos$TM sqrt)
      histograma=hist(rfm ativos$TM sqrt, col="darkblue", border="black",
          probability = T,xlab="Valor médio de compra",
          ylab="Densidade", xlim = range(dens$x), main="Histograma")
      abline(v = c(median(rfm_ativos$TM_sqrt), mean(rfm_ativos$TM_sqrt)),
             col = c("green", "red"),
             lwd = c(2,2),
             lty=c(1,2))
      # acrescentando quadro com legenda
      legend(x="topright", #posicao da legenda
             c("Mediana","Média"), #nomes da legenda
             col=c("green","red"), #cores
             lty=c(1,2), #estilo da linha
```

```
dens <- density(rfm_ativos$recency_days_sqrt)
      histograma=hist(rfm ativos$recency days sqrt, col="darkblue", border="black",
                      probability = T,xlab="Recência",breaks= 30,
                      ylab="Densidade", xlim = range(dens$x), main="Histograma")
      abline(v = c(median(rfm_ativos$recency_days_sqrt),
mean(rfm ativos$recency days sqrt)),
             col = c("green", "red"),
             lwd = c(2,2),
             lty=c(1,2))
      # acrescentando quadro com legenda
      legend(x="topright", #posicao da legenda
             c("Mediana","Média"), #nomes da legenda
             col=c("green","red"), #cores
             lty=c(1,2), #estilo da linha
             lwd=c(2,2)) #grossura das linhas

      dens <- density(rfm ativos$freq mensal sqrt)
      histograma=hist(rfm_ativos$freq_mensal_sqrt, col="darkblue", border="black",
                      probability = T,xlab="Frequência Mensal",breaks= 30,
                      ylab="Densidade", xlim = range(dens$x), main="Histograma")
      abline(v = c(median(rfm ativos$freq mensal sqrt), mean(rfm ativos$freq mensal sqrt)),
             col = c("green", "red"),
             lwd = c(2,2),
             lty=c(1,2))
      # acrescentando quadro com legenda
      legend(x="topright", #posicao da legenda
             c("Mediana","Média"), #nomes da legenda
             col=c("green","red"), #cores
             lty=c(1,2), #estilo da linha
             lwd=c(2,2)) #grossura das linhas

      dens <- density(rfm_ativos$sobremesa_normalizado_sqrt)
      histograma=hist(rfm_ativos$sobremesa_normalizado_sqrt, col="darkblue", border="black",
                      probability = T,xlab="Sobremesa",breaks= 30,
                      ylab="Densidade", xlim = range(dens$x), main="Histograma")
      abline(v = c(median(rfm ativos$sobremesa normalizado sqrt),
mean(rfm_ativos$sobremesa_normalizado_sqrt)),
             col = c("green", "red"),
             lwd = c(2,2),
             lty=c(1,2))
      # acrescentando quadro com legenda
      legend(x="topright", #posicao da legenda
             c("Mediana","Média"), #nomes da legenda
             col=c("green","red"), #cores
             lty=c(1,2), #estilo da linha
             lwd=c(2,2)) #grossura das linhas

      dens <- density(rfm_ativos$acompanhamento_normalizado_sqrt)
      histograma=hist(rfm_ativos$acompanhamento_normalizado_sqrt, col="darkblue",
border="black",
                      probability = T,xlab="Acompanhamento",breaks= 30,
                      ylab="Densidade", xlim = range(dens$x), main="Histograma")
      abline(v = c(median(rfm ativos$acompanhamento normalizado sqrt),
mean(rfm_ativos$acompanhamento_normalizado_sqrt)),
             col = c("green", "red"),
             lwd = c(2,2),
             lty=c(1,2))
```

```
      # acrescentando quadro com legenda
      legend(x="topright", #posicao da legenda
             c("Mediana","Média"), #nomes da legenda
             col=c("green","red"), #cores
             lty=c(1,2), #estilo da linha
             lwd=c(2,2)) #grossura das linhas

      dens <- density(rfm ativos$sanduiche normalizado sqrt)
      histograma=hist(rfm_ativos$sanduiche_normalizado_sqrt, col="darkblue", border="black",
                      probability = T,xlab="Sanduíche",breaks= 30,
                      ylab="Densidade", xlim = range(dens$x), main="Histograma")
      abline(v = c(median(rfm ativos$sanduiche normalizado sqrt),
mean(rfm_ativos$sanduiche_normalizado_sqrt)),
             col = c("green", "red"),
             lwd = c(2,2),
             lty=c(1,2))
      # acrescentando quadro com legenda
      legend(x="topright", #posicao da legenda
             c("Mediana","Média"), #nomes da legenda
             col=c("green","red"), #cores
             lty=c(1,2), #estilo da linha
             lwd=c(2,2)) #grossura das linhas

#---------------------------------------------------------------------

#SELECIONAR BASE AMOSTRAL

dataset_sem_outliers<-rfm_ativos[,c(-2,-3,-4,-5,-6,-7)]

      library(ggplot2)
      library(lattice)
      library(caret)

      set.seed(100)
      particao = createDataPartition(1:dim(dataset_sem_outliers)[1],p=.1) #Colocar valor
pequeno
      dataset_clusterizavel = dataset_sem_outliers[particao$Resample1,]
      dataset classificavel = dataset sem outliers[- particao$Resample1,]

      data <-dataset clusterizavel[,-1]

#---------------------------------------------------------------------

#MÉTODO DO COTOVELO

     library(knitr)
     library(rmarkdown)
     library(readxl)
     library(kableExtra)
     library(factoextra)
     library(gridExtra)

     k.max <- 15
```

```
     #BASE PARA CLUSTERIZAÇÃO
     wss <- sapply(1:k.max,function(k){kmeans(data, k, nstart=50,
iter.max = 15)$tot.withinss}) # WSS é a distância total dos pontos de dados de seus
respectivos centróides do cluster
     wss
     plot(wss)

#-----------------------------------------------------------------------

#ALGORITMOS DE CLUSTERIZAÇÃO

  #1. K-Means

     Inicio_kmeans<-Sys.time()
     results kmeans <- kmeans(data, 3, iter.max = 5)
     Fim kmeans<-Sys.time()
     results kmeans
     kmeanstabela<-as.data.frame(results_kmeans$centers)
     kmeanstabela$recency_days<-kmeanstabela$recency_days_sqrt^2
     kmeanstabela$TM<-kmeanstabela$TM_sqrt^2
     kmeanstabela$freq mensal<-kmeanstabela$freq mensal sqrt^2
     kmeanstabela$sobremesa_normalizado<-kmeanstabela$sobremesa_normalizado_sqrt^2
     kmeanstabela$acompanhamento normalizado<-kmeanstabela$acompanhamento normalizado sqrt^2
     kmeanstabela$sanduiche_normalizado<-kmeanstabela$sanduiche_normalizado_sqrt^2


     resultado kmeans aux <- table(dataset clusterizavel$customer id,
results_kmeans$cluster) #Qual id está em qual cluster
     resultado kmeans <- as.data.frame.matrix(resultado kmeans aux) #Transformando a tabela
em dataframe
     resultado kmeans$customer id <- rownames(resultado kmeans aux) #Ajustando para ter uma
coluna de id (variável)

  #2. Hierárquico

     d <- dist(data, method = "euclidian") # É construída a matriz de distancias entre cada
elemento da base de dados
     results hierarq <- hclust(d, "ward.D")

     # Plotagem do dendograma originado pelos resultados (results).
     plot(results_hierarq)

     fviz_dend(results_hierarq, k = 3,# numero de clusters
               cex = 0.5, # label size
               k colors = c("#2E9FDF", "#00AFBB", "#E7B800", "#FC4E07"),
               color labels by k = TRUE, # color labels by groups
               rect = TRUE # Add rectangle around groups
     )


     rect.hclust(results hierarq, k = 3, border = "red")
     clusters_hierarq <- cutree(results_hierarq, 3)
     clusters hierarq

     resultado hierarq aux <- table(dataset clusterizavel$customer id, clusters hierarq)
#Qual id está em qual cluster
```

```
     resultado_hierarq <- as.data.frame.matrix(resultado_hierarq_aux) #Transformando a
tabela em dataframe
     resultado_hierarq$customer_id <- rownames(resultado_hierarq_aux) #Ajustando para ter
uma coluna de id (variável)

     plot(x[], col = clusters hierarq)

     require("scatterplot3d")
     scatterplot3d(x[]) #3D

     library(rgl)
     plot3d(x[,1:3], col=clusters hierarq, main="hierarchical clusters") #3D

     #Resultado dos clusters
     library (dplyr)
     resultado hierarq<-left join(resultado hierarq, dataset clusterizavel %>%
select(`customer id`, `recency days`,`freq mensal`,`TM`, `sobremesa normalizado`,
`acompanhamento_normalizado`, `sanduiche_normalizado` ),by = c("customer_id" =
"customer_id"))
     resultado_hierarq_1<-subset(resultado_hierarq, resultado_hierarq$`1`>=1)
     hierarq 1<-resultado hierarq 1[,c(-1,-2,-3,-4)]
     colMeans(hierarq_1)
     resultado hierarq 2<-subset(resultado hierarq, resultado hierarq$`2`>=1)
     hierarq_2<-resultado_hierarq_2[,c(-1,-2,-3,-4)]
     colMeans(hierarq 2)
     resultado_hierarq_3<-subset(resultado_hierarq, resultado_hierarq$`3`>=1)
     hierarq 3<-resultado hierarq 3[,c(-1,-2,-3,-4)]
     colMeans(hierarq_3)
     summary(hierarq 3)


  #3. Model Based

     library(mclust)
     #GMT<- Mclust(data)
     Inicio gmm<-Sys.time()
     GMT <- Mclust(data,3) #or specify number of clusters
     Fim gmm<-Sys.time()
     Fim_gmm- Inicio_gmm
     summary(GMT$BIC)
     plot(GMT, what = 'BIC', xlab = "Número de componentes",ylim = range(GMT$BIC[,-(1:2)],
na.rm = TRUE),
          legendArgs = list(x = 'bottomleft'))
     summary(GMT, parameters=TRUE) #Médias as variáveis de cada cluster
     GMT$modelName #Modelo usado
     GMT$G #Número ótimo de clusters
     #plot(GMT, what=c("classification")) #The types of what argument are: "BIC",
"classification", "uncertainty", "density"
     GMT$classification


     resultado_modelo_aux <- table(dataset_clusterizavel$customer_id, GMT$classification)
#Qual id está em qual cluster
     resultado_modelo <- as.data.frame.matrix(resultado_modelo_aux) #Transformando a tabela
em dataframe
     resultado_modelo$customer_id <- rownames(resultado_modelo_aux) #Ajustando para ter uma
coluna de id (variável)
```

```
     GMMtabela<-as.data.frame(t(GMT$parameters$mean))
     GMMtabela$recency days<-GMMtabela$recency days sqrt^2
     GMMtabela$TM<-GMMtabela$TM_sqrt^2
     GMMtabela$freq mensal<-GMMtabela$freq mensal sqrt^2
     GMMtabela$sobremesa_normalizado<-GMMtabela$sobremesa_normalizado_sqrt^2
     GMMtabela$acompanhamento normalizado<-GMMtabela$acompanhamento normalizado sqrt^2
     GMMtabela$sanduiche_normalizado<-GMMtabela$sanduiche_normalizado_sqrt^2

#-----------------------------------------------------------------------

#ESCOLHA DO MELHOR MÉTODO

  #1. Coeficiente de silhueta

     require("cluster")
     silhueta kmeans<-silhouette(results kmeans$cluster, dist(data)) # the result closer to
1 implies high clustering quality
     fviz_silhouette(silhueta_kmeans)

     silhueta_hierarq<-silhouette(clusters_hierarq, dist(data))
     fviz silhouette(silhueta hierarq)

     silhueta modelo<-silhouette(GMT$classification, dist(data))
     fviz_silhouette(silhueta_modelo)

  #2. MANOVA

     #K-Means

library(gdata)

     resultado_kmeans1<-subset(resultado_kmeans, resultado_kmeans$`1`>0)
     resultado_kmeans1<-resultado_kmeans1[,c(-2,-3)]
     resultado_kmeans1 <-rename.vars(resultado_kmeans1, c("1","customer_id"),
c("cluster","customer_id"))

     resultado_kmeans2<-subset(resultado_kmeans, resultado_kmeans$`2`>0)
     resultado kmeans2<-resultado kmeans2[,c(-1,-3)]
     resultado_kmeans2$`2`<-2
     resultado kmeans2 <-rename.vars(resultado kmeans2, c("2","customer id"),
c("cluster","customer_id"))

     resultado_kmeans3<-subset(resultado_kmeans, resultado_kmeans$`3`>0)
     resultado kmeans3<-resultado kmeans3[,c(-1,-2)]
     resultado kmeans3$`3`<-3
     resultado kmeans3 <-rename.vars(resultado kmeans3, c("3","customer id"),
c("cluster","customer_id"))

     resultado_kmeans_clusterizado<-rbind(resultado_kmeans1,resultado_kmeans2,
resultado_kmeans3)

     library (dplyr)
     dataset clusterizado kmeans<-left join(dataset clusterizavel,
resultado_kmeans_clusterizado)
     dataset clusterizado kmeans$cluster<-as.character(dataset clusterizado kmeans$cluster)
```

```
     dataset_clusterizado_kmeans$recency_days<-
dataset clusterizado kmeans$recency days sqrt^2
     dataset_clusterizado_kmeans$TM<-dataset_clusterizado_kmeans$TM_sqrt^2
     dataset clusterizado kmeans$freq mensal<-dataset clusterizado kmeans$freq mensal sqrt^2
     dataset_clusterizado_kmeans$sobremesa_normalizado<-
dataset clusterizado kmeans$sobremesa normalizado sqrt^2
     dataset_clusterizado_kmeans$acompanhamento_normalizado<-
dataset clusterizado kmeans$acompanhamento normalizado sqrt^2
     dataset_clusterizado_kmeans$sanduiche_normalizado<-
dataset clusterizado kmeans$sanduiche normalizado sqrt^2

     par(mfrow=c(1,3))
     boxplot(dataset_clusterizado_kmeans$recency_days ~ dataset_clusterizado_kmeans$cluster,
col="blue", main="Recência", ylab = "Recência",
             xlab = "Cluster")
     boxplot(dataset clusterizado kmeans$freq mensal ~ dataset clusterizado kmeans$cluster,
col="blue", main="Frequência Mensal", ylab = "Frequência Mensal",
             xlab = "Cluster")
     boxplot(dataset_clusterizado_kmeans$TM ~ dataset_clusterizado_kmeans$cluster,
col="blue", main="Valor Médio de Compra", ylab = "Valor Médio de Compra",
              xlab = "Cluster")
     par(mfrow=c(1,3))
     boxplot(dataset clusterizado kmeans$sobremesa normalizado ~
dataset_clusterizado_kmeans$cluster, col="blue", main="Sobremesa", ylab = "Sobremesa",
             xlab = "Cluster")
     boxplot(dataset_clusterizado_kmeans$acompanhamento_normalizado ~
dataset clusterizado kmeans$cluster, col="blue", main="Acompanhamento", ylab =
"Acompanhamento",
             xlab = "Cluster")
     boxplot(dataset clusterizado kmeans$sanduiche normalizado ~
dataset clusterizado kmeans$cluster, col="blue", main="Sanduiche", ylab = "Sanduíche",
             xlab = "Cluster")

     res.kmeans <- manova(cbind(recency_days_sqrt, freq_mensal_sqrt, TM_sqrt,
sobremesa_normalizado_sqrt, acompanhamento_normalizado_sqrt,sanduiche_normalizado_sqrt) ~
cluster, data = dataset clusterizado kmeans)
     summary(res.kmeans, test='Pillai')
     summary(res.kmeans, test='Wilks')
     summary(res.kmeans, test='Hotelling-Lawley')
     summary(res.kmeans, test='Roy')

     #hierarq
     resultado_hierarq1<-subset(resultado_hierarq, resultado_hierarq$`1`>0)
     resultado hierarq1<-resultado hierarq1[,c(-2,-3)]
     resultado hierarq1 <-rename.vars(resultado hierarq1, c("1","customer id"),
c("cluster","customer id"))

     resultado_hierarq2<-subset(resultado_hierarq, resultado_hierarq$`2`>0)
     resultado_hierarq2<-resultado_hierarq2[,c(-1,-3)]
     resultado_hierarq2$`2`<-2
     resultado hierarq2 <-rename.vars(resultado hierarq2, c("2","customer id"),
c("cluster","customer_id"))

     resultado_hierarq3<-subset(resultado_hierarq, resultado_hierarq$`3`>0)
     resultado hierarq3<-resultado hierarq3[,c(-1,-2)]
     resultado_hierarq3$`3`<-3
```

```
     resultado_hierarq3 <-rename.vars(resultado_hierarq3, c("3","customer_id"),
c("cluster","customer id"))

     resultado hierarq clusterizado<-rbind(resultado hierarq1,resultado hierarq2,
resultado_hierarq3)
     library (dplyr)
     dataset_clusterizado_hierarq<-left_join(dataset_clusterizavel,
resultado hierarq clusterizado)
     dataset_clusterizado_hierarq$cluster<-
as.character(dataset clusterizado hierarq$cluster)

     res.hierarq <- manova(cbind(recency days, freq mensal, TM, sobremesa normalizado,
acompanhamento_normalizado,sanduiche_normalizado) ~ cluster, data =
dataset_clusterizado_hierarq)
     summary(res.hierarq, test='Pillai')
     summary(res.hierarq, test='Wilks')
     summary(res.hierarq, test='Hotelling-Lawley')
     summary(res.hierarq, test='Roy')

     #GMM
     resultado mod1<-subset(resultado modelo, resultado modelo$`1`>0)
     resultado_mod1<-resultado_mod1[,c(-2,-3)]#, -4, -5, -6, -7, -8, -9)]
     resultado mod1 <-rename.vars(resultado mod1, c("1","customer id"),
c("cluster","customer_id"))

     resultado_mod2<-subset(resultado_modelo, resultado_modelo$`2`>0)
     resultado mod2<-resultado mod2[,c(-1,-3)]#, -4, -5, -6, -7, -8, -9)]
     resultado_mod2$`2`<-2
     resultado mod2 <-rename.vars(resultado mod2, c("2","customer id"),
c("cluster","customer id"))

     resultado_mod3<-subset(resultado_modelo, resultado_modelo$`3`>0)
     resultado_mod3<-resultado_mod3[,c(-1,-2)]#, -4, -5, -6, -7, -8, -9)]
     resultado_mod3$`3`<-3
     resultado_mod3 <-rename.vars(resultado_mod3, c("3","customer_id"),
c("cluster","customer id"))

  resultado mod clusterizado<-rbind(resultado mod1, resultado mod2, resultado mod3)#,
resultado_mod4, resultado_mod5, resultado_mod6, resultado_mod7, resultado_mod8,
resultado mod9)
     library (dplyr)
     dataset clusterizado mod<-left join(dataset clusterizavel, resultado mod clusterizado)
     dataset_clusterizado_mod$cluster<-as.character(dataset_clusterizado_mod$cluster)
     dataset clusterizado mod$recency days<-dataset clusterizado mod$recency days sqrt^2
     dataset clusterizado mod$TM<-dataset clusterizado mod$TM sqrt^2
     dataset clusterizado mod$freq mensal<-dataset clusterizado mod$freq mensal sqrt^2
     dataset_clusterizado_mod$sobremesa_normalizado<-
dataset_clusterizado_mod$sobremesa_normalizado_sqrt^2
     dataset_clusterizado_mod$acompanhamento_normalizado<-
dataset_clusterizado_mod$acompanhamento_normalizado_sqrt^2
     dataset clusterizado mod$sanduiche normalizado<-
dataset_clusterizado_mod$sanduiche_normalizado_sqrt^2

     par(mfrow=c(1,3))
     boxplot(dataset clusterizado mod$recency days ~ dataset clusterizado mod$cluster,
col="blue", main="Recência", ylab = "Recência",
             xlab = "Cluster")
```

```
     boxplot(dataset_clusterizado_mod$freq_mensal ~ dataset_clusterizado_mod$cluster,
col="blue", main="Frequência Mensal", ylab = "Frequência Mensal",
             xlab = "Cluster")
     boxplot(dataset clusterizado mod$TM ~ dataset clusterizado mod$cluster, col="blue",
main="Valor Médio de Compra", ylab = "Valor Médio de Compra",
             xlab = "Cluster")
     par(mfrow=c(1,3))
     boxplot(dataset clusterizado mod$sobremesa normalizado ~
dataset_clusterizado_mod$cluster, col="blue", main="Sobremesa", ylab = "Sobremesa",
             xlab = "Cluster")
     boxplot(dataset_clusterizado_mod$acompanhamento_normalizado ~
dataset clusterizado mod$cluster, col="blue", main="Acompanhamento", ylab =
"Acompanhamento",
             xlab = "Cluster")
     boxplot(dataset clusterizado mod$sanduiche normalizado ~
dataset clusterizado mod$cluster, col="blue", main="Sanduiche", ylab = "Sanduíche",
             xlab = "Cluster")

     res.GMM <- manova(cbind(recency_days_sqrt, freq_mensal_sqrt, TM_sqrt,
sobremesa_normalizado_sqrt, acompanhamento_normalizado_sqrt,sanduiche_normalizado_sqrt) ~
cluster, data = dataset clusterizado mod)
     summary(res.GMM, test='Pillai')
     summary(res.GMM, test='Wilks')
     summary(res.GMM, test='Hotelling-Lawley')
     summary(res.GMM, test='Roy')

#3. Indice davis-bouldin

     library(clusterSim)

     #K-Means
     DB_kmeans<-index.DB(data, results_kmeans$cluster, centrotypes="centroids")

     #GMM
     DB_GMM<-index.DB(data, GMT$classification , centrotypes="centroids")


  #4. Calinski

     #library(cluster.stats)
     CH_Kmeans<-round(index.G1(data,results_kmeans$cluster),digits=2)


     CH GMM<-round(index.G1(data,GMT$classification),digits=2)

   #Algoritmo escolhido
     dataset_clusterizado <- dataset_clusterizado_kmeans

     library(plotly)
     library(dplyr)
     plot ly(dataset clusterizado kmeans, x=~TM, y=~freq mensal, z=~recency days,
color=~cluster) %>% add_markers(size=1.5)

     par(mfrow=c(1,6))
     boxplot(dataset clusterizado$recency days ~ dataset clusterizado$cluster, col="blue",
main="Recência")
```

```
     boxplot(dataset_clusterizado$freq_mensal ~ dataset_clusterizado$cluster, col="blue",
main="Frequência Mensal")
     boxplot(dataset_clusterizado$TM ~ dataset_clusterizado$cluster, col="blue",
main="Ticket Médio")
     boxplot(dataset_clusterizado$sobremesa_normalizado ~ dataset_clusterizado$cluster,
col="blue", main="Sobremesa Normalizado")
     boxplot(dataset_clusterizado$acompanhamento_normalizado ~ dataset_clusterizado$cluster,
col="blue", main="Acompanhamento Normalizado")
     boxplot(dataset_clusterizado$sanduiche_normalizado ~ dataset_clusterizado$cluster,
col="blue", main="Sanduiche Normalizado")


#---------------------------------------------------------------------

#CLASSIFICADOR

  #TREINO E TESTE

dataset_clusterizado<-dataset_clusterizado_normalizado[,c(-2,-3,-4,-5,-6,-7)]
     names( dataset_clusterizado)[3:8] <- c("Recência", "Valor_Médio_de_Compra",
"Frequência Mensal", "Sobremesa","Acompanhamento","Sanduíche")

     set.seed(100)
     particao = createDataPartition(1:dim(dataset_clusterizado)[1],p=.80)
     dataset treino id = dataset clusterizado[particao$Resample1,]
     dataset_teste_id = dataset_clusterizado[- particao$Resample1,]

     dataset_treino <- dataset_treino_id[,-1]
     dataset teste <- dataset teste id[,-1]

     #Treino

library(rpart)
     #train_tree = rpart(cluster~., data = dataset_treino, method = "class")
     #print(train_tree)
     #summary(train tree)
     #plot(train_tree)
     #text(train tree, pretty = 0, cex = 0.6)


     #validação cruzada
     fitControl <- trainControl(method = "cv",
                                number = 5)


     inicio tree<-Sys.time()
     train_tree <- train(cluster~., data = dataset_treino, method = "rpart", trControl =
fitControl)  #Árvore de decisão
     fim_tree<-Sys.time()
     fim_tree-inicio_tree
     print(train tree)

     #plot(train tree)

     library(rpart.plot)
     rpart.plot(train_tree$finalModel, type=4)
     text(train_tree, pretty = 0, cex = 0.6)
```

```
     dataset_treino.cols = dataset_treino[,1:6]
     inicio forest<-Sys.time()
     train_forest <- train(cluster~., data = dataset_treino, method = "rf", trControl =
fitControl, metric = "Accuracy",
                           importance = TRUE,
                           nodesize = 14,
                           ntree = 800,
                           maxnodes = 24)  #Floresta aleatória
     fim_forest<-Sys.time()
     print(train forest)
     rpart.plot(train_forest$finalModel, type=4)


     train_forest$finalModel$err.rate
      oob.err.data <- data.frame(
       Quantidade de arvores = rep(1:nrow(train forest$finalModel$err.rate), 4),
       Legenda = rep(c("OOB","1","2","3"), each = nrow(train forest$finalModel$err.rate)),
       Erro = c(train_forest$finalModel$err.rate[,"OOB"],
train_forest$finalModel$err.rate[,"1"], train_forest$finalModel$err.rate[,"2"],
train_forest$finalModel$err.rate[,"3"]))

     #install.packages("e1071")
     library(e1071)
     library(caret)
     ggplot(data = oob.err.data, aes(x = Quantidade de arvores, y= Erro)) +
geom_line(aes(color = Legenda))


     plot(train forest)
     text(train forest, pretty = 0, cex = 0.6)

     #Teste

     #data_test_new <- dataset_teste   # Duplicate test data set
     #data_test_new$customer_id[which(!(data_test_new$customer_id %in%
unique(dataset treino$customer id)))] <- NA  # Replace new levels by NA
     inicio_teste_forest<-Sys.time()
     test forest predict = predict(train forest, dataset teste)
     fim_teste_forest<-Sys.time()

     ## confusion matrix
     confusion matrix <- table(dataset teste$cluster,test forest predict)
     percentual_acuracia <- 100*sum(diag(confusion_matrix))/sum(confusion_matrix)
     print(paste("acuracia:",percentual acuracia,"%"))

     inicio teste tree<-Sys.time()
     test_tree_predict = predict(train_tree, dataset_teste)
     fim_teste_tree<-Sys.time()

     ## confusion matrix
     confusion matrix <- table(dataset teste$cluster,test tree predict)
     percentual_acuracia <- 100*sum(diag(confusion_matrix))/sum(confusion_matrix)
     print(paste("acuracia:",percentual acuracia,"%"))

  #CLASSIFICAÇÃO

     dataset_clusterizado<-dataset_clusterizado_normalizado[,c(-2,-3,-4,-5,-6,-7)]
```

```
     names( dataset_clusterizado)[3:8] <- c("Recência", "Valor_Médio_de_Compra",
"Frequência Mensal", "Sobremesa","Acompanhamento","Sanduíche")

     dataset classificavel$recency days<-dataset classificavel$recency days sqrt^2
     dataset_classificavel$TM<-dataset_classificavel$TM_sqrt^2
     dataset classificavel$freq mensal<-dataset classificavel$freq mensal sqrt^2
     dataset_classificavel$sobremesa_normalizado<-
dataset classificavel$sobremesa normalizado sqrt^2
     dataset_classificavel$acompanhamento_normalizado<-
dataset classificavel$acompanhamento normalizado sqrt^2
     dataset_classificavel$sanduiche_normalizado<-
dataset classificavel$sanduiche normalizado sqrt^2

     dataset_classificavel_sem_id <- dataset_classificavel[,c(-1, -2, -3, -4, -5, -6, -7)]
     names( dataset classificavel sem id)[1:6] <- c("Recência", "Valor Médio de Compra",
"Frequência Mensal", "Sobremesa","Acompanhamento","Sanduíche")
     inicio classificacao<-Sys.time()
     tree_predict = predict(train_forest, newdata = dataset_classificavel_sem_id) #Colocar
base que não foi clusterizada
     fim_classificacao<-Sys.time()
     print(tree predict)

     tree predict<-as.numeric(tree predict)
     dataset_classificado<-dataset_classificavel[,c(-2, -3, -4, -5, -6, -7)]
     names( dataset classificado)[2:7] <- c("Recência", "Valor Médio de Compra",
"Frequência_Mensal", "Sobremesa","Acompanhamento","Sanduíche")
     dataset classificado$cluster<-tree predict

     library(dplyr)
     dataset classificado<-dplyr::select(dataset classificado,
customer id,cluster,Recência,Valor Médio de Compra,Frequência Mensal,Sobremesa,Acompanhament
o,Sanduíche)

     dataset_final <- rbind (dataset_classificado, dataset_clusterizado)
     write.csv(dataset_final, "dataset_final.csv")
     ------

  library("csv")

     agrupamento final <- read.csv("dataset final.csv")
     agrupamento_final<-agrupamento_final[,c(-1,-2)]



     par(mfrow=c(1,3))
     boxplot(agrupamento final$Recência ~ agrupamento final$cluster, col="blue",
main="Recência", ylab = "Recência",
             xlab = "Cluster")
     boxplot(agrupamento_final$Frequência_Mensal ~ agrupamento_final$cluster, col="blue",
main="Frequência Mensal",ylab = "Frequência Mensal",
             xlab = "Cluster")
     boxplot(agrupamento_final$Valor_Médio_de_Compra ~ agrupamento_final$cluster,
col="blue", main="Valor Médio de Compra", ylab = "Valor Médio de Compra",
             xlab = "Cluster")

     par(mfrow=c(1,3))
```

```
     boxplot(agrupamento_final$Sobremesa ~ agrupamento_final$cluster, col="blue",
main="Sobremesa", ylab = "Sobremesa",
             xlab = "Cluster")
     boxplot(agrupamento final$Acompanhamento ~ agrupamento final$cluster, col="blue",
main="Acompanhamento", ylab = "Acompanhamento",
             xlab = "Cluster")
     boxplot(agrupamento_final$Sanduíche ~ agrupamento_final$cluster, col="blue",
main="Sanduiche", ylab = "Sanduíche",
             xlab = "Cluster")

     agrupamento_final_1 <- subset(agrupamento_final,agrupamento_final$cluster<=1)
     agrupamento final 2 aux <- subset(agrupamento final,agrupamento final$cluster>=2)
     agrupamento_final_2 <-
subset(agrupamento_final_2_aux,agrupamento_final_2_aux$cluster<3)
     agrupamento final 3 <- subset(agrupamento final,agrupamento final$cluster>=3)

     summary(agrupamento final 1)
     summary(agrupamento_final_2)
     summary(agrupamento_final_3)
```