UTILIZAÇÃO DE CLUSTERIZAÇÃO PARA SEGMENTAÇÃO DE CLIENTES A PARTIR DE DADOS DE VAREJO

Henrique José Wilbert, Aurélio Faustino Hoppe – Orientador

Curso de Bacharel em Ciências da Computação
Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

hwilbert@furb.br, aureliof@furb.br

Resumo: *Apesar de existirem várias formas de identificar comportamentos de clientes, poucas extraem esse valor a partir de informações já existentes em uma base de dados, muito menos extraem características relevantes. Este trabalho apresenta o desenvolvimento de um protótipo utilizando os atributos recência, frequência e monetário (RFM) para a segmentação de clientes de uma base de dados de varejo. Para isso, utilizou-se o algoritmo de clusterização k-means. A avaliação da qualidade dos clusters foi obtida através dos índices de validação interna: Silhouette, Calinski Harabasz e Davies Bouldin. Uma vez não obtido o consenso entre os três, foi aplicado três índices de validação externa: estabilidade global, estabilidade por cluster e estabilidade SLSa. Foram obtidos seis segmentos de clientes, identificados pelo seu comportamento único: clientes perdidos, clientes desinteressados, clientes recentes, clientes menos recentes, clientes leais e melhores clientes. Seu comportamento foi evidenciado e analisado, indicando tendências e preferências.*

Palavras-chave*: Varejo. Comportamento de clientes. Clusterização. K-means. RFM. Segmentação. Índices de validação externos. Silhouette. Calinski-Harabasz. Davies Bouldin.*

1 INTRODUÇÃO

Com a evolução da tecnologia de informação a partir dos anos 90, grandes empresas adotaram sistemas de gerenciamento na forma de softwares Enterprise Resource Planning (ERP). Estes softwares auxiliam em suas rotinas à nível operacional, seja no controle do estoque, fiscal, financeiro, transacional e até recursos humanos (NIJHER, 2014). A partir disso, alcançou-se um patamar de eficiência nunca concebido, visto que registros antes realizados em papel e caneta, passaram a ser produzidos automaticamente. Ainda segundo os autores, em paralelo a informatização desses processos, houve também um crescimento da quantidade de dados armazenados referentes à produtos, clientes, transações, gastos e receitas.

Diante deste contexto, avançaram-se também as táticas de marketing direto, como por exemplo, o envio de catálogos por correio, até ofertas altamente objetivas para indivíduos selecionados, cujas informações transacionais estavam presentes na base de dados. O foco das relações empresa-cliente volta-se então à clientes que já possuem um cadastro com a empresa, visto que o custo para adquirir um cliente novo através de publicidade é muito maior que o custo de alimentar uma relação já existente (SRIVASTAVA; CHANDRA; SRIVASTAVA, 2019).

Segundo Reinartz, Thomas e Kumar (2005, p. 77), quando empresas tratam os gastos entre aquisição e retenção de clientes, destinar menos recursos para a retenção impactará em uma lucratividade menor à longo prazo, comparando-se a investimentos menores em aquisição de clientes. Ainda segundo os autores, no conceito de relações de retenção, atribui-se grande ênfase à lealdade e lucratividade de um cliente, sendo lealdade a tendência do cliente comprar com a empresa e a lucratividade, a medida geral de quanto lucro um cliente traz à empresa através de suas compras.

De acordo com Nguyen, Sherif e Newby (2007, p. 114), com o avanço da gerência das relações com clientes foram abertas novas vias pelas quais sua lealdade e lucratividade pode ser cultivada, atraindo uma crescente demanda por parte de empresas, visto que a adoção destes meios permite que as organizações melhorem seu serviço ao consumidor, consequentemente gerando renda. Com isso, diferentes ferramentas acabam sendo utilizadas, como sistemas de recomendação que, geralmente em ramos e-commerce, levam em conta várias características pertinentes ao comportamento do cliente, construindo um perfil próprio que será utilizado para realizar a recomendação de um produto que talvez seja de seu interesse. Outra ferramenta pertinente à lucros e lealdade é a segmentação, que visa separar uma única massa de clientes em segmentos homogêneos em termos de comportamento, permitindo o desenvolvimento de campanhas, decisões e estratégias de marketing especializadas à cada grupo de acordo com suas características (TSIPTSIS; CHORIANOPOULOS, 2009, p.4).

Roberts, Kayande e Stremersch (2014) afirmam que as ferramentas de segmentação apresentam o maior impacto dentre as decisões de marketing disponíveis, indicando uma grande procura por tais ferramentas ao longo da próxima década. Dolnicar, Leisch e Grün (2018) indagam que a segmentação de clientes apresenta muitos benefícios caso implementada corretamente, dentre os principais está a introspecção por parte da empresa sobre os tipos de clientes que ela possui, e consequentemente, seus comportamentos e necessidades. Por outro lado, Dolnicar, Leisch e Grün (2018) também destacam que caso a segmentação não seja aplicada corretamente, a execução da prática em sua totalidade gera

um desperdício de recursos, visto que a falha retorna segmentos não condizentes com o comportamento real, deixando a empresa que aplicou com nenhuma informação válida sobre os clientes que ela possui.

Em relação a segmentação de clientes, algumas métricas tornam-se relevantes nos contextos aos quais estão inseridas. Segundo Kumar (2008, p. 29), o modelo Recency Frequency Monetary (RFM), é utilizado em empresas de venda por catálogo, enquanto empresas de *high-tech* tendem a usar Share of Wallet (SOW) para implementar suas estratégias de marketing. Já o modelo Past Customer Value (PCV), geralmente é utilizado em empresas de serviços financeiros. Dentre os modelos citados, o RFM é o que possui maior facilidade de aplicação em diversas áreas de comércio, varejo e supermercados, visto que são necessários apenas os dados transacionais (vendas) dos clientes, dos quais são obtidos os atributos de Recência (R), Frequência (F) e Monetário (M).

A partir desses dados, segundo Tsiptsis e Chorianopoulos (2009, p. 335), é possível detectar bons clientes a partir das melhores pontuações de RFM. Se o cliente efetuou uma compra recentemente, seu atributo R será alto. Caso ele compre muitas vezes ao longo de um determinado período, seu atributo F será maior. Por fim, caso seus gastos totais forem significativos, terá um atributo M alto. Ao categorizar o cliente dentro destas três características, é possível obter uma hierarquia de importância, tendo os clientes que possuem valores RFM altos no topo, e clientes que possuem valores baixos na base. Apesar destas possibilidades para a segmentação, o modelo padrão original é um tanto quanto arbitrário, segmentando os clientes em quintis, cinco grupos com 20% dos clientes, não atentando-se às nuances e todas a interpretações que a base de clientes pode possuir. Além disso, o método também pode produzir uma grande quantidade de grupos (até 125), que por muitas vezes, não representam significativamente os clientes de um estabelecimento.

Com o aumento da quantidade de dados e do trabalho manual requerido para segmentação, Alelyani, Tang e Liu (2014) indagam que a automatização desse processo se tornou indispensável, tendo como uma de suas principais técnicas, o *clustering*. Esta técnica consiste em categorizar dados sem rótulo em grupos chamados *clusters*, cujos integrantes são parecidos entre si e diferentes de integrantes de outros *clusters*, com base nas características analisadas. Dentre os algoritmos de clusterização, o algoritmo K-means é um dos mais populares, sendo simples de implementar e dispondo de extensos estudos sobre seus comportamentos (FRÄNTI; SIERANOJA, 2019). No contexto de avaliação, Hämäläinen, Jauhiainen e Kärkkäinen (2017) destacam que a qualidade de uma solução pode ser medida através dos índices de validação, que consideram a compactação dos dados dos *clusters* e sua separação com outros *clusters*, permitindo a obtenção de um grau de certeza maior ao considerar um resultado de segmentação advindo de um algoritmo de clusterização.

Diante da importância da segmentação de clientes, e da crucialidade de extração de suas características comportamentais de maneira efetiva, este trabalho apresenta a criação de um protótipo que utilize os atributos do modelo RFM em conjunto com o algoritmo de clusterização K-means. Cuja função será extrair de maneira automática as informações de uma base de dados real de varejo, com o objetivo de identificar diferentes segmentos de clientes com base em seus comportamentos. Para a validação da quantidade de clusters foram utilizados três índices internos (Silhouette, Calinski-Harabasz e Davies-Bouldin) e três índices externos (estabilidade global, estabilidade por *cluster* e estabilidade SLSa - Segment Level Stability across solutions), para evidenciar a qualidade das soluções obtidas.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão descritos os assuntos que servirão de base para a realização deste trabalho. A seção 2.1 trata do tema de *clustering*, e a seção 2.2 discorre sobre os trabalhos relacionados.

2.1 CLUSTERING

Para Cherkassky e Mulier (2007), *clustering* se trata do problema de separar um conjunto de dados em grupos chamados de "*clusters*" baseado em alguma medida de similaridade. O objetivo é encontrar um conjunto de *clusters* dos quais as amostras dentro dos mesmos são mais similares entre si do que quando comparadas com amostras de outros *clusters*. A análise destes *clusters* consiste no fato de que a medida de similaridade entre eles é escolhida subjetivamente baseado na sua habilidade de criar *clusters* interessantes ao analista. Cherkassky e Mulier (2007) classificam os algoritmos em dois tipos principais: (i) Hierárquicos, seguindo uma estrutura de árvores; (ii) Particionais, que geram *clusters* a partir de sucessivas secções, cujos métodos são identificados em dois grupos: particionais onde cada dado é atribuído à um e somente à um *cluster* e particionais que podem pertencer à vários *clusters*.

Um dos algoritmos de clusterização particional mais utilizados é o K-means. De acordo com Ghosh e Kumar (2013), o K-means é o método de particionamento mais aplicado para analisar dados, separando os objetos em *clusters* mutuamente exclusivos (K) de maneira que os objetos de cada *cluster* fiquem tão perto entre si quanto possível, porém tão longe quanto possível de objetos em *clusters* diferentes. Cada *cluster* possui um ponto central (centroide), cuja localização é obtida através da média da localização de todos os pontos pertencentes ao *cluster*. Segundo Ghosh e Kumar (2013), o algoritmo baseia-se na constante atualização da posição dos centroides e recálculo dos pontos mais próximos, sendo que inicialmente os k-centroides são aleatoriamente distribuídos no espaço. O algoritmo acaba sua execução quando

nenhum ponto muda de *cluster* ou nenhum centroide se move. Formalmente, o algoritmo K-means pode ser resumido nos seguintes passos:

a) definição de K – escolher um determinado número de *clusters* desejados;
b) inicialização – escolher k pontos de partida para ser usados como centroides dos *clusters*;
c) classificação – examinar cada ponto no conjunto de dados e designá-lo ao *cluster* cujo centroide seja mais próximo;
d) cálculo de centroides – quando cada ponto no conjunto de dados estiver designado a um *cluster*, é necessário recalcular os novos k centroides, com base na média das distâncias dos dados contidos em cada *cluster*;
e) critério de convergência – os passos c e d precisam ser repetidos até nenhum dado mudar de *cluster* ou nenhum centroide mudar de posição.

Rendón *et al.* (2011) denotam que cada algoritmo de clusterização depende das características do conjunto de dados, bem como os parâmetros de inicialização requeridos por cada um. Caso os parâmetros passados forem incorretos, o resultado de clusterização pode acabar sendo pior que o esperado, não condizendo com os reais *clusters* intrínsecos ao conjunto de dados. Rendón *et al.* (2011) afirmam que para melhor definir os parâmetros que se encaixam à uma determinada solução, são necessários valores guias, os quais apresentam métricas para se basear na decisão de quais valores utilizar como parâmetro. Para esta finalidade, são utilizados os índices de validação de *clusters*.

Para Hämäläinen, Jauhiainen e Kärkkäinen (2017), um índice de validação avalia a qualidade do resultado de um algoritmo de clusterização, procurando achar a separação que melhor se encaixa com a natureza dos dados. O número de *clusters* dado como parâmetro para vários algoritmos deveria ser decidido com base na estrutura natural dos dados, porém nem sempre há uma solução clara sobre o melhor número de *clusters*. Segundo Rendón *et al.* (2011), existem dois principais métodos de validação de *clusters*: os baseados em critérios internos e os baseados em critérios externos.

Hämäläinen, Jauhiainen e Kärkkäinen (2017) destacam que os índices de validação internos medem a tangibilidade do objetivo, sendo esta, uma alta similaridade entre os dados dentro de um *cluster* e uma alta diferença entre os dados de *clusters* diferentes. Essas medidas são chamadas de separação intra e inter-cluster, respectivamente. Uma boa medida de separação intra-cluster possui números baixos, já uma boa medida de separação inter-cluster possui números altos. Os autores ainda destacam que nenhum índice de validação é perfeito para qualquer contexto, alguns índices são mais adequados à diferentes tipos de dados. Devido à isso, recomenda-se utilizar múltiplos índices para a análise de *clusters*. Rendón *et al.* (2011) ressaltam que, apesar dos índices de validação internos se basearem nas medidas de compactação (separação intra-cluster) e separabilidade (separação inter-cluster), nem sempre estas medidas são utilizadas de maneira igual. Alguns índices realizam comparações entre estas medidas com base em pares de *clusters*, outros aplicam uma média geral levando em consideração todos os *clusters* contidos numa solução.

Segundo Draszawka e Szymański (2011), índices externos de validação podem ser utilizados quando a segmentação real dos dados é conhecida *a priori*. Uma vez conhecidas as categorias ou classes de cada dado, é possível comparar esta informação com os *clusters* criados pelo algoritmo. Este tipo de validação é preferencial quando se procura o melhor algoritmo de clusterização para um determinado conjunto de dados. Os autores descrevem que as características utilizadas pelos índices externos geralmente levam em consideração a similaridade dos diferentes resultados, bem como a variabilidade entre soluções.

2.2 TRABALHOS CORRELATOS

Esta seção agrega três trabalhos relacionados ao contexto dos temas apresentados. No Quadro 1 é descrito o modelo de segmentação denominado Length, Recency, Frequency, Monetary and Periodicity (LRFMP) proposto por Peker, Kocyigit e Eren (2017), ao qual considera a longevidade e a periodicidade. O Quadro 2 apresenta o trabalho de Hidayat *et al*. (2020), que consiste na aplicação do modelo Recency Frequency Monetary (RFM) em conjunto com o algoritmo Fuzzy c-means (FCM) para clusterização de escolas. Por fim, no Quadro 3 detalha-se o modelo R+FM, sendo uma versão modificada do modelo original RFM aplicada em clientes de uma empresa de e-commerce (TAVAKOLI et al., 2018).

Quadro 1 – LRFMP model for customer segmentation in the grocery retail industry: a case study

Referência	Peker, Kocyigit e Eren (2017)
Objetivos	Propor um modelo baseado no RFM para a classificação de segmentos de clientes no contexto de padarias. Identificar diferentes segmentos de padarias baseado no modelo proposto. Bem como realizar análises sobre o comportamento dos segmentos obtidos.
Principais funcionalidades	Foi criado um modelo baseado no modelo RFM, denominado Lenght, Recency, Frequency, Monetary, Periodicity (LRFMP), que utiliza a longevidade e periodicidade. A partir do modelo aplicado aos dados, foi realizada a clusterização do conjunto de 10.471 clientes, obtendo cinco *clusters*. Após isso, foram realizadas análises sobre os grupos de clientes, categorizando-os pelas suas características mais importantes, e aplicando inferências para possíveis ações promocionais.

Ferramentas de desenvolvimento	Para a extração de características dos clientes, o modelo LRFMP foi desenvolvido tendo o contexto de padarias em mente, sendo adicionados os atributos de longevidade (duração da relação cliente-empresa em dias) e periodicidade (regularidade com que visitas à padaria são feitas, medida pelo desvio padrão dos tempos entre as visitas) para capturar informações pertinentes ao segmento de mercado de padarias. A normalização dos dados foi feita aplicando o algoritmo z-score, deixando os valores com uma média de 0 e desvio-padrão de 1. A clusterização foi feita com o algoritmo K-means, utilizando para validação de k os índices de Silhouette, Calinski-Harabasz e Davies-Bouldin, gerando uma solução com cinco *clusters* de clientes. A descrição dos *clusters* é feita pela análise descritiva das características LRFMP de cada *cluster*, indicando valores altos ou abaixo do normal.
Resultados e conclusões	Os autores indicam a descoberta de cinco segmentos de clientes: clientes leais de alta contribuição, clientes leais de baixa contribuição, clientes incertos, clientes perdidos de alta contribuição e clientes perdidos de baixa contribuição. Baseado nas características de cada segmento, os autores denotaram sua importância para o ramo de padarias, chamando atenção para possíveis oportunidades de fortalecimento na relação empresa-cliente focado nas peculiaridades de cada segmento. Os autores também atentaram para a contribuição que o modelo LRFMP agregou ao destacar comportamentos oriundos dos atributos L e P, visto que facilitou o agrupamento em *clusters* mais significativos.

Fonte: elaborado pelo autor.

Quadro 2 – Segmentation of university customers loyalty based on RFM analysis using Fuzzy c-means clustering

Referência	Hidayat et al. (2020)
Objetivos	Utilizar o modelo RFM em conjunto com Fuzzy C-Means (FCM) para a clusterização de escolas. Analisar possíveis escolas leais num contexto de acordo de cooperação com a universidades. Detectar diferentes segmentos com base na potencialidade de relações.
Principais funcionalidades	Hidayat et al. (2020) adaptaram o modelo RFM para o contexto de escolas, levando à uma medida de lealdade atrelada aos alunos que finalizam o ensino médio e engajam na universidade estudada, contando com alunos de 342 escolas. Os dados RFM das escolas foi processado pelo algoritmo FCM e validado com o índice Partition coefficient index (PCI).
Ferramentas de desenvolvimento	A definição dos atributos RFM foi realizada com base nos alunos pertencentes às escolas do estudo. A recência foi definida pela quantidade de vezes que a escola apresentou pelo menos um aluno para admissão à universidade ao longo de cinco anos, a frequência foi definida pela média de alunos apresentados a cada ano e o atributo monetário foi definido pela quantidade total de alunos apresentados. Os atributos foram separados em intervalos de desempenho, resultando em uma pontuação de 1 a 5 para cada atributo, sendo 5 o melhor desempenho e 1 o pior. O algoritmo FCM foi aplicado em conjunto com o índice de validação PCI, de maneira que a melhor solução de acordo com o índice foi de quatro *clusters*, com um índice médio de 0,86 (sendo 0 um resultado ruim e 1 um resultado ótimo).
Resultados e conclusões	Os autores denotam que os quatro *clusters* adquiridos podem ser definidos pelas suas características principais, que se resumem à: alto potencial, potencial, baixo potencial e muito baixo potencial. Hidayat et al. (2020) indagam que, com base nesta segmentação, a universidade pode filtrar possíveis relações indesejadas e focar em uma cooperação com escolas que de fato influenciam no fluxo de alunos novos.

Fonte: elaborado pelo autor.

Quadro 3 – Customer segmentation and strategy development based on user behavior analysis, RFM model and data mining techniques: a case study

Referência	Tavakoli et al (2018)
Objetivos	Propor um modelo baseado no RFM que se adeque às relações entre os seus próprios atributos. Identificar as melhores métricas para medir os atributos do modelo criado. Clusterizar clientes utilizando o modelo proposto no contexto de lojas *e-commerce*. Adquirir segmentos de clientes com comportamento identificável a partir das características RFM estabelecidas, bem como aplicar uma campanha ativa de relacionamento com os clientes de um segmento escolhido.
Principais funcionalidades	Foram definidas métricas para os atributos RFM, bem como também criado um modelo novo, R+FM, que considera a dependência entre as variáveis F e M e atribui diferentes pesos de acordo com sua importância para o ramo *e-commerce*. A partir do modelo R+FM, foi realizada a clusterização do conjunto de 3 milhões clientes, obtendo dez *clusters* separados em três categorias de recência. Após isso, foram realizadas análises sobre os grupos gerados, e foi aplicado uma campanha de SMS para um segmento específico.

Ferramentas de desenvolvimento	O modelo R+FM foi desenvolvido com a métrica de recência em dias, arbitrariamente separando os clientes em ativos (menos de 90 dias desde a última compra), em processo de perda (tendo sido feita a última compra entre 90 e 365 dias) e perdidos (última compra foi a mais de um ano). Após a separação por recência foram aplicados pesos aos atributos de frequência e monetário, para equilibrar as características num grau de importância condizente com o ramo de *e-commerce*, resultando num multiplicador de 1,3 para frequência e 0,7 para monetário. A clusterização foi feita com o algoritmo K-means, utilizando um valor fixo para o número de *clusters* decidido com base nas necessidades da empresa estudada, resultando em 4 *clusters* para os clientes ativos, 3 *clusters* para os clientes perdidos e 3 para os clientes em processo de perda. A descrição dos *clusters* é feita pela interpretação dos valores médios dos atributos dos *clusters*, e após isso foi aplicada uma campanha de SMS com promoções para todos os *clusters* de clientes ativos.
Resultados e conclusões	Segundo os autores, os clientes foram segmentados em quatro grupos de clientes ativos: alto valor, médio valor alto monetário, médio valor alta frequência, baixo valor. Foram então estipuladas possíveis campanhas para cada segmento, sendo aplicada uma campanha de SMS com promoções que visavam aumentar atributos em segmentos que possuíam pouco, com incentivos à múltiplas compras para grupos de baixa frequência, descontos em compras grandes para grupos com baixo atributo monetário, e incentivos de lealdade para clientes valiosos. Tavakoli et al (2018) comentam que a campanha resultou em incrementos de até 14 dólares nos valores médios do atributo monetário.

Fonte: elaborado pelo autor.

3 DESCRIÇÃO DO PROTÓTIPO

Nesta seção são descritos os aspectos mais relevantes do protótipo desenvolvido. A subseção 3.1 trata da especificação, onde são expostos os requisitos do protótipo. Logo após, é abordada a implementação, onde é apresentado o desenvolvimento do protótipo em si junto, com seus resultados.

3.1 ESPECIFICAÇÃO

Para a descrição das funções do protótipo, são expostos os Requisitos Funcionais (RF) no Quadro 4 e os Requisitos Não Funcionais (RNF) no Quadro 5.

Quadro 4 – Requisitos Funcionais

RF01 - Adquirir os dados transacionais de clientes a partir de um banco de dados.
RF02 - Filtrar os clientes com informações irregulares.
RF03 - Extrair dos clientes as características (recência, frequência e monetária) utilizadas no modelo RFM.
RF04 - Normalizar os dados para evitar disparidades nas escalas dos atributos.
RF05 - Exibir num gráfico 3D a localização dos clientes a partir da pontuação das características RFM.
RF06 - Segmentar em *clusters* os clientes com base nos atributos RFM.

Fonte: elaborado pelo autor

Quadro 5 – Requisitos Não Funcionais

RNF01 - Utilizar o algoritmo de clusterização K-means para a segmentação dos clientes.
RNF02 - Aplicar os índices de validação interna Silhouette, Calinski-Harabasz e Davies-Bouldin para validação da qualidade dos *clusters*.
RNF03 - Aplicar os índices de validação externa de estabilidade global, estabilidade por *cluster* e estabilidade SLSa.
RNF04 - Utilizar a linguagem Python para o desenvolvimento do protótipo.

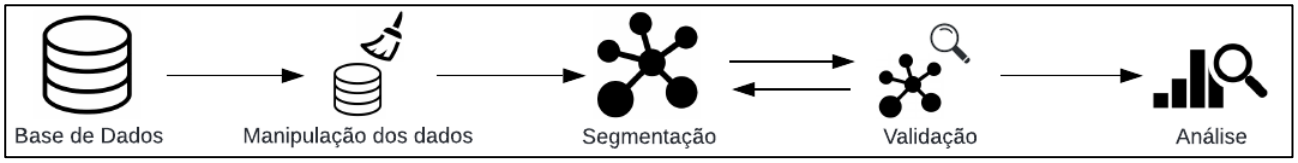
Fonte: elaborado pelo autor

Nos apêndices A e B, encontram-se os diagramas de componentes e de atividades, demonstrando os principais módulos que fazem parte do protótipo, bem como suas principais funções e atividades expostas em sequência.

3.2 IMPLEMENTAÇÃO E RESULTADOS

Para o desenvolvimento deste protótipo, foram realizados os seguintes passos: obtenção dos dados de clientes a partir da base fornecida, manipulação dos dados, segmentação, validação e análise, conforme ilustra a Figura 1.

Figura 1 – Processo utilizado para segmentar clientes



Fonte: elaborado pelo autor.

3.2.1 Obtenção dos dados

O primeiro passo refere-se à obtenção dos clientes e informações pertinentes a partir de uma base de dados de um software de gestão comercial. O segmento de mercado da base em questão é focado na venda de roupas masculinas e femininas. Foram extraídos 1845 clientes com informações do período entre 01/01/2016 e 01/12/2021, consolidando dados de tabelas de vendas para adequar-se com a Tabela 1, que representa a estrutura necessária para realizar a segmentação baseada nos atributos RFM.

Tabela 1 – Estrutura dos dados obtidos

ID	Recência	Frequência	Monetário
38	139	65	37176

Fonte: elaborado pelo autor.

Um registro possui um número de identificação (ID) próprio referente ao seu número de identificação no banco de dados original. Também é gravado a sua recência, representando a quantidade de dias desde a última compra. A frequência contabiliza as compras feitas durante o período estabelecido. Por fim, a informação "monetário" representa o total gasto em R$ dentro do período considerado.

Obteve-se cada um dos atributos RFM a partir da extração de todas as vendas realizadas por frente de caixa para um determinado cliente (comércio). A recência foi adquirida calculando a diferença em dias entre a data da última compra e a data final do período estabelecido para a obtenção dos dados (01/12/21). No caso do cliente da Tabela 1, sua recência é de 139 dias desde sua última compra (15/07/21). A frequência foi adquirida totalizando a quantidade de vendas realizadas para o cliente no determinado período. Para o cliente da Tabela 1, sua frequência contabiliza 65 compras no período de 01/01/2016 até 01/12/21. Por fim, o atributo monetário foi criado a partir da soma dos totais de cada venda, resultando em um valor de R$ 37.176,00.

3.2.2 Manipulação dos dados

Nesta etapa foram realizados procedimentos de remoção de dados inconsistentes como vendas vazias, tipos de operações não adequadas como recebimentos e pagamentos de vendas feitas à crediário, e remoção de clientes sem vendas na loja. Com essas operações, 97 clientes foram removidos, resultando num total de 1748 clientes na base. Em seguida, aplicou-se uma normalização dos atributos, visto que o algoritmo K-means utiliza uma medida de distância, e o intervalo de valor dos atributos varia conforme sua natureza (monetário pode apresentar valores na casa dos milhares, enquanto os outros atributos distribuem-se em centenas), podendo influenciar negativamente nos resultados.

Para a normalização dos atributos foi utilizado o método Min-Max. Segundo Saranya e Manikandan (2013), a normalização pelo método Min-Max realiza uma alteração linear nos dados, que são transformados para um novo intervalo. Conforme exposto na Equação 1, tendo um valor *v* de um atributo *A* de intervalo $[min_A, max_A]$, este é transformado para o novo intervalo $[novo\_min_A, novo\_max_A]$, que no caso desta aplicação encontra-se entre 0 e 1.

$$\frac{v - min_A}{max_A - min_A} \left(novo_{max_A} - novo\_min_A\right) + novo\_min_A \quad (1)$$

A partir da aplicação deste método, o registro exposto na Tabela 1 têm seus valores convertidos e representados na Tabela 2, tendo como valor máximo 1 e mínimo 0.

Tabela 2 - Estrutura dos dados obtidos após normalização

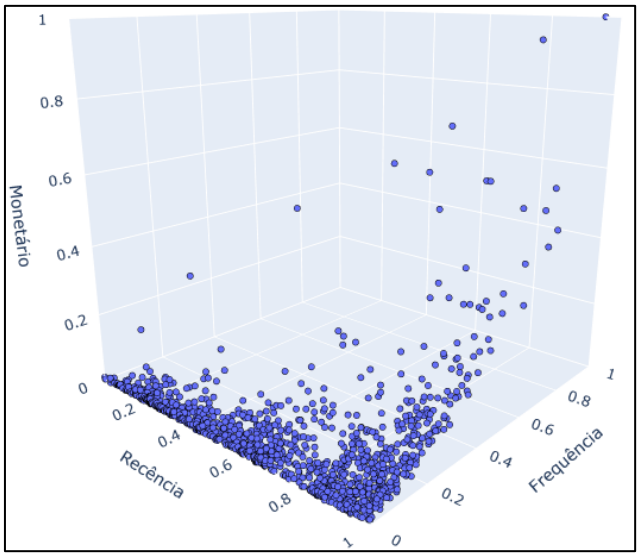
ID	Recência	Frequência	Monetário
38	0,0074928	0,71910112	0.43890863

Fonte: elaborado pelo autor.

Valores próximos a 1 indicam que o atributo do cliente em questão é alto em relação a todos os outros clientes, valores próximos a 0 indicam que o atributo é baixo em relação aos outros. Ou seja, o cliente com o melhor atributo de frequência terá 1 como seu valor, caso contrário terá 0. Uma exceção é o atributo de recência, que devido ao formato em que foi adquirido, acaba possuindo valores inversos, tendo uma recência boa caso o valor for próximo de 0 e ruim caso fique perto de 1 (quanto menos dias desde a última compra, melhor). Por motivos de simplicidade e consistência de medidas, foi aplicada uma simples transformação dos valores de recência, subtraindo o valor de 1. Desta maneira, valores baixos transformam-se em valores altos, e vice-versa, contribuindo para uma melhor análise, visto que agora clientes com um desempenho RFM bom tendem a ter todos os atributos próximos de 1.

No Apêndice C encontram-se os principais códigos-fonte referentes à etapa de manipulação dos dados, expondo as rotinas de filtragem de vendas e a aplicação do método Min-Max. Após a manipulação dos dados, é possível exibir cada cliente em um gráfico 3D, com cada eixo representando um atributo como na Figura 2. É possível identificar que apesar dos dados não fornecerem uma distribuição de *clusters* naturais, eles apresentam uma estrutura própria, com uma quantidade grande de clientes aglomerados no canto esquerdo do gráfico indicando uma baixa frequência, distribuídos sobre diversos intervalos de recência, com poucos clientes de atributo monetário alto.

Figura 2 – Representação 3D dos clientes



Fonte: elaborado pelo autor.

3.2.3 Validação Interna

As etapas de segmentação e validação foram realizadas em paralelo, visto que o algoritmo utilizado, K-means, requer a especificação do número de *clusters* desejados. Logo, dispôs-se de validações internas e externas para auxiliar na decisão. Arbelaitz *et al*. (2013) concluem em seu estudo através de uma análise estatística, que dos 30 índices internos pesquisados, 10 provam ser recomendáveis para utilização. No topo desta lista, encontram-se os índices de Silhouette, Calinski-Harabasz e Davies-Bouldin.

De acordo com Rousseeuw (1987), para gerar o índice de Silhouette de um dado, são necessárias apenas duas coisas: os *clusters* obtidos e o conjunto das distâncias entre todos os dados observados, sendo calculado para cada *i* o seu respectivo índice Silhouette *s(i)*. Calcula-se também a média de dissimilaridade das distâncias de *i* com o resto dos dados do *cluster* de *i*, denotado por *a(i)*. No passo seguinte, é obtido o valor mínimo entre as distâncias de *i* e qualquer outro *cluster* (é descoberto então o *cluster* vizinho de *i*, ou seja, o *cluster* com que *i* mais se encaixaria caso não estivesse em seu *cluster* original), denotado por *b(i)*. Este processo pode ser resumido pela Equação 2, que resulta em num número entre -1 e 1, sendo -1 uma categorização ruim do objeto *i* (não condizente com seu *cluster* atual) e sendo 1 uma categorização ótima. Para obter a qualidade da clusterização em geral, é obtida a média de *s(i)* para todos os objetos *i* do conjunto de dados.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2}$$

Para Calinski e Harabasz (1974), o seu índice Variance Rate Criterion (VRC), demonstrado na Equação 3, considera: a soma dos quadrados entre grupos (Between Group Sum of Squares - BGSS) que retrata a variância entre *clusters* levando em conta a distância de seus centroides até o centroide global; e a soma dos quadrados dentre grupos (Within Group Sum of Squares - WGSS) que retrata a variância dentro de *clusters* levando em conta as distâncias dos pontos em um *cluster* até o seu centroide. Considera-se também o número de observações/dados *n* e o número de *clusters* *k*. Quando este índice é utilizado, procura-se maximizar o resultado conforme o valor de *k* é alterado.

$$VRC = \frac{BGSS}{\mathrm{k}-1} \Big/ \frac{WGSS}{\mathrm{n}-\mathrm{k}} \tag{3}$$

Davies e Bouldin (1979) denotam que o objetivo de seu índice é definir uma medida de separação de *clusters* $R(S_i, S_j, M_{ij})$ que permita a computação da similaridade média de cada *cluster* com o seu *cluster* mais similar (vizinho), o valor mais baixo possível seria o resultado ideal. Com S_i sendo a medida de dispersão do *cluster i*, S_j sendo a medida de dispersão do *cluster j* e M_{ij} sendo a distância entre os *clusters i* e *j*, conforme Equação 4.

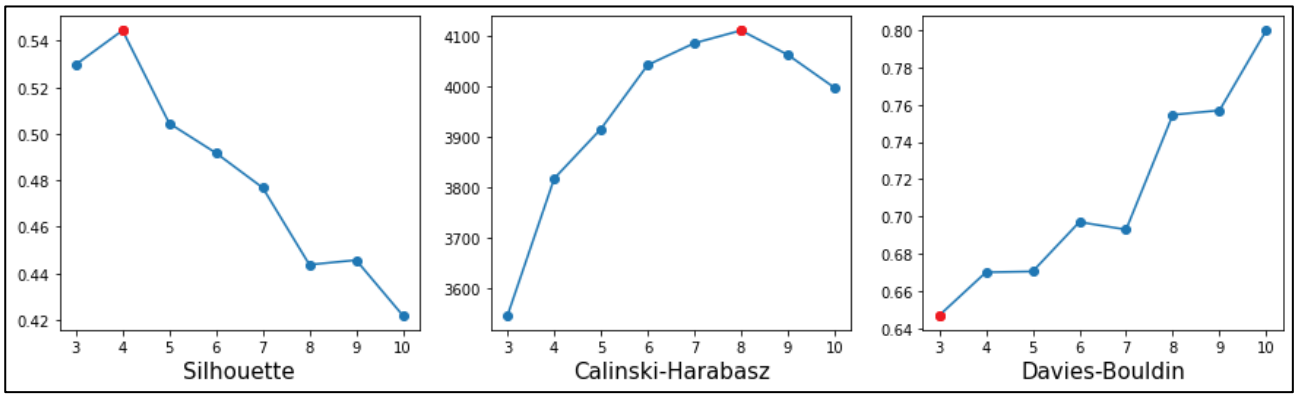
$$R_{ij} \equiv \frac{S_i + S_j}{M_{ij}} \quad \bar{R} \equiv \frac{1}{N}\sum_{i=1}^{N} R_i \tag{4}$$

De acordo com Davies e Bouldin (1979), primeiro obtêm-se R_{ij} de todos os *clusters*, isto é, a razão de distâncias inter e intra-cluster entre o *cluster i* e *j*. Após isso, obtêm-se R_i (o valor mais alto de R_{ij}) identificando para cada *cluster*,

o *cluster* vizinho ao qual ele mais se assemelha. Por fim, é calculado o índice em si $\bar{R}$, sendo este, a soma total das similaridades de *N clusters* com seus vizinhos mais próximos.

Foram geradas 8 soluções de segmentação com o algoritmo K-means, partindo de k=3 até k=10. Após isso, os melhores resultados entre as soluções k de acordo com cada índice foram obtidos. De acordo com a Figura 3, o índice de Silhouette sugeriu 4 *clusters*, enquanto Calinski-Harabasz sugeriu 8 e Davies-Bouldin 3. Ressalta-se que na interpretação do índice de Silhouette e Calinski-Harabasz é escolhido o maior valor, já no índice de Davies-Bouldin é selecionado o menor valor. No Apêndice D encontram-se os principais códigos-fonte referentes à etapa de validação interna, expondo as rotinas utilizadas para cada índice de validação.

Figura 3 – Índices de validação interna

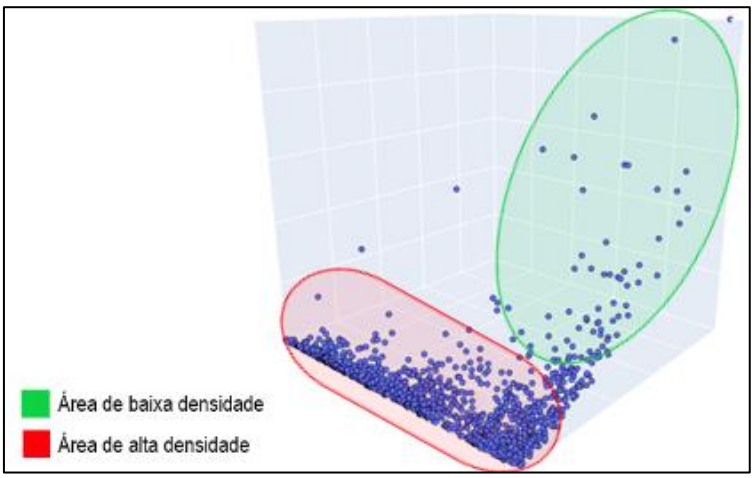


Fonte: elaborado pelo autor.

As sugestões de número de *clusters* a partir dos índices apresentaram uma alta variabilidade, causando grande incerteza na detecção do número de *clusters*. Este resultado é comum entre conjuntos de dados que não possuem *clusters* ocorridos naturalmente. Segundo Dolnicar, Leisch e Grün (2018, p. 157), dados de consumidores tipicamente não contém segmentos naturais, dificultando a obtenção do número ideal de *clusters* a partir de índices internos de validação.

Ademais, várias características na distribuição dos dados podem afetar os índices internos de validação, Liu et *al*. (2010) destacam que diferentes densidades, ruído, formas arbitrárias, e *clusters* muito próximos limitam os resultados e introduzem desafios adicionais ao processo de estimação da quantidade de *clusters*. Mais especificamente, os índices de Silhouette e Davies-Bouldin sofrem com a proximidade de *clusters*, e Calinski-Harabasz desempenha mal em distribuições de tamanho desigual. Todas essas características citadas se fazem presentes ao visualizar a distribuição dos dados na Figura 4, que além de apresentar tamanhos diferentes nos possíveis *clusters*, demonstra uma aglomeração de dados em um lado específico da distribuição e uma densidade baixa em áreas de alto atributo monetário.

Figura 4 – Densidades da distribuição de clientes



Fonte: elaborado pelo autor.

3.2.4 Estabilidade global

Com a incerteza gerada pelos índices internos de validação, faz-se necessário adquirir outras visões sobre o conjunto de dados, de forma que seja possível assegurar um número de *clusters* ideal com uma boa margem de certeza. Com isso, foram aplicados índices externos de validação. Como não existem *clusters* "verdadeiros" ou dados de teste com categorias *a priori* para fazer a comparação externa, foi utilizada uma medida de estabilidade global, definida por

Ernst e Dolnicar (2017) onde a informação externa é composta por soluções com diferentes quantidades de *clusters*. Esta medida utiliza dois conceitos principais: o *bootstrapping* para a seleção aleatória de amostras, e o Adjusted Rand Index (ARI) para a medida de similaridade entre duas soluções.

Segundo Roodman et *al*. (2019), métodos de *bootstrapping* consistem na geração de amostras que representam o conjunto de dados original e a aplicação de avaliações em tais amostras, de maneira que uma avaliação em uma amostra geralmente equivale à uma avaliação no conjunto de dados original. Os métodos de *bootstrapping* podem também utilizar o conceito de substituição, onde uma amostra pode conter dados repetidos do conjunto original, criando uma característica de aleatoriedade e variância no conjunto de amostras.

De acordo com Robert, Vasseur e Brault (2021), o Rand Index (RI) é uma medida de similaridade entre duas soluções de clusterização z e z', podendo ser definida pela Equação 5. Onde *a* é o número de pares de elementos que foram atribuídos ao mesmo *cluster* em z e z'. *b* é o número de pares de elementos que foram atribuídos ao mesmo *cluster* na solução z, mas em *clusters* diferentes na solução z'. *c* é o número de pares de elementos que foram atribuídos a *clusters* diferentes na solução z, mas em *clusters* iguais na solução z', e por fim, *d* é o número de pares de elementos que estão em *clusters* diferentes tanto em z quanto em z'. Robert, Vasseur e Vincent (2021) ressaltam que *a* e *d* podem ser interpretados como concordâncias, enquanto *b* e *c* podem ser interpretados como discordâncias entre as soluções z e z' avaliadas. O Índice de Rand resulta em um valor entre 0 e 1, sendo 1 uma concordância perfeita dos *clusters* entre as soluções e 0 uma discordância total.

$$RI = \frac{a + d}{a + b + c + d} \tag{5}$$

Segundo Santos e Embrechts (2009), RI possui alguns problemas conhecidos, como nem sempre apresentar um valor 0 para soluções completamente aleatórias, e variar positivamente conforme o número de *clusters* nas soluções aumenta. Para isso medidas diferentes foram criadas com intuito de corrigir tais problemas, uma destas medidas é o Adjusted Rand Index (ARI) exibido na Equação 6. Com Índice sendo o resultado de RI, Índice Esperado sendo o índice de RI esperado numa ocasião em que as observações são aleatoriamente atribuídas à diferentes *clusters*, e Índice Máximo sendo o valor máximo possível de RI. O índice ARI varia entre -1 e 1, sendo -1 um valor para uma dissimilaridade alta e 1 um valor para uma similaridade alta entre duas soluções.

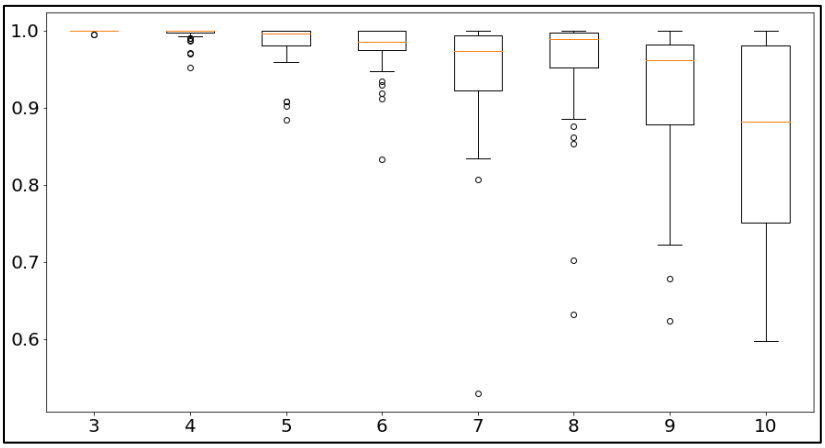
$$ARI = \frac{\text{Índice} - \text{Índice Esperado}}{\text{Índice Máximo} - \text{Índice Esperado}} \tag{6}$$

A partir dos dois conceitos apresentados, faz-se possível aplicar a medida de estabilidade global sugerida por Ernst e Dolnicar (2017), dividida nos seguintes passos:

a) criar 50 pares de amostras *bootstrap* com substituição a partir dos dados;
b) realizar a clusterização de cada par de amostras com k *clusters*;
c) calcular o valor ARI do par clusterizado, gerando um valor de -1 a 1;
d) repetir os passos "b" e "c" até atingir o número k desejado.

No Apêndice E encontra-se o código-fonte referente ao índice de estabilidade global, expondo cada passo implementado. Após o término desse algoritmo, têm-se 50 valores ARI para cada k analisado. É possível então, representar os valores num gráfico boxplot como apresentado na Figura 5, onde o eixo horizontal representa as soluções com diferentes números de clusters e o eixo vertical representa o valor do índice ARI, as formas de caixa representam 50% dos valores e os traços da parte externa representam os outros 50%. Valores *outliers* (valores fora da distribuição padrão) são representados por círculos fora da parte externa e o traço laranja indica a média dos valores. Com esse gráfico é obtida uma visão concreta da estabilidade de cada solução com k *clusters*. Observa-se que o valor de ARI tende a diminuir conforme k é aumentado, indicando uma variação maior nas possíveis diferenças entre os *clusters* de cada solução, isto é, quanto maior o valor de k, execuções diferentes de K-means resultarão em soluções de clusterização totalmente diferentes.

Figura 5– *Boxplot* de ARI para cada número de *clusters*



Fonte: elaborado pelo autor.

Após a análise do gráfico *boxplot*, evidencia-se que a partir de 6 *clusters* o valor de ARI entre as soluções varia constantemente de forma negativa. Logo, tornam-se viáveis as soluções com 4, 5 e 6 *clusters*, visto que estas possuem estabilidade desejável em relação às soluções com números k maiores, e ainda permitem uma análise mais detalhada de cada *cluster*. A opção com k=3 não foi considerada pelo fato de possuir poucos *clusters*, agregando clientes diferentes num mesmo grupo, tornando a solução mais generalizada e com poucos detalhes discerníveis em cada *cluster*.

3.2.5 Estabilidade por Cluster

A estabilidade global permite analisar as soluções com respeito à sua mudança conforme várias execuções de um algoritmo de clusterização, porém não permite uma análise detalhada na estrutura específica das soluções, isto é, os *clusters*. Dito isso, após selecionados três candidatos de segmentação (k=4, k=5 e k=6), é possível calcular a estabilidade por *cluster* descrita por Hennig (2007), sendo esta semelhante ao método anterior, porém com um foco em *clusters* ao invés de soluções inteiras. Esta estabilidade permite detectar *clusters* instáveis dentro de soluções estáveis e vice-versa, auxiliando posteriormente em análises descritivas e seleção das soluções em si, visto que providencia uma visão por *cluster*, facilitando na escolha de um segmento em potencial de clientes. O método utiliza o conceito de *bootstrapping* e o índice de Jaccard para o cálculo de estabilidade.

De acordo com Lee et *al.* (2019), o Índice de Jaccard mede a similaridade entre dois conjuntos de dados *A* e *B*, levando em consideração a união e intersecção destes conjuntos, como expresso na Equação 7. A parte superior representa a intersecção de *A* com *B*, contendo então valores comuns aos dois conjuntos. A parte inferior representa a união de *A* com *B*, contendo todos os valores de *A* e todos os valores de *B*, subtraindo então os valores comuns aos dois para evitar sua duplicação. O índice de Jaccard retorna um valor entre 0 e 1, sendo 1 um valor que representa a similaridade total entre os dois conjuntos, e 0 o valor que representa a dissimilaridade total entre os conjuntos.

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{7}$$

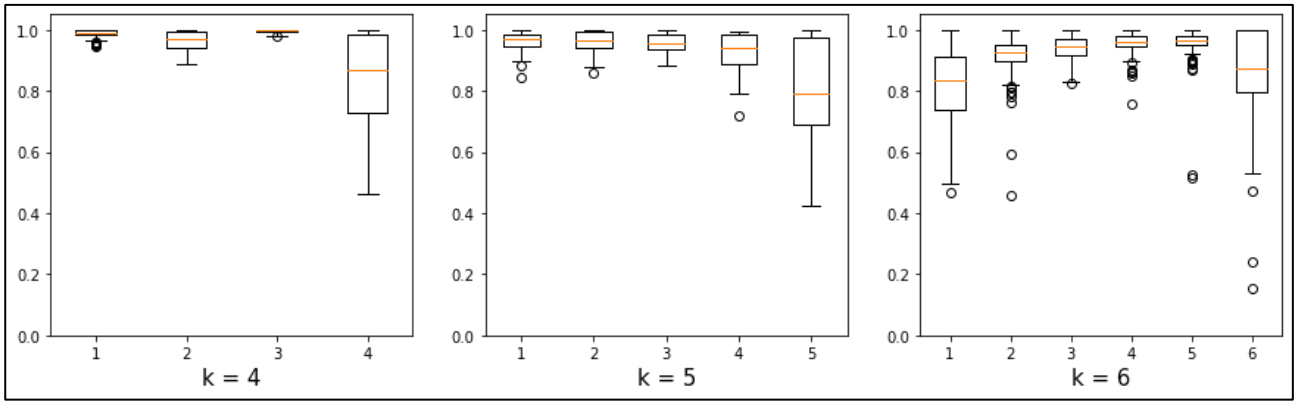
Em suma, o método descrito por Hennig (2007) realiza uma amostragem por *bootstrap* do conjunto original, comparando através do índice de Jaccard cada *cluster* pertencente à solução original com sua representação *bootstrap*, gerando um índice para cada *cluster*. O algoritmo pode ser descrito com os seguintes passos:

a) criar 100 amostras *bootstrap* com substituição a partir dos dados da solução original;
b) realizar a clusterização da amostra;
c) extrair os dados comuns entre a solução original e a amostra (lembrando que a amostra pode conter dados repetidos ou não conter alguns dados do conjunto original);
d) para cada *cluster* do conjunto original, calcular o índice de máximo de Jaccard entre ele e cada *cluster* da amostra *bootstrap*;
e) repetir a partir do passo "b" para o restante das amostras.

No Apêndice F encontra-se o código-fonte referente ao índice de estabilidade por *cluster*, expondo cada passo implementado. A execução do algoritmo resulta em 100 valores em um intervalo de 0 a 1 para cada *cluster*, é possível então exibir eles em um gráfico boxplot. O eixo horizontal de cada gráfico representa os diferentes clusters contidos em uma solução, já o eixo vertical representa o valor do índice de Jaccard, Permitindo visualizar de maneira intuitiva a estabilidade de cada *cluster* dentro de uma solução. Como existem três candidatos à solução (k=4, k=5 e k=6), foi aplicado o algoritmo para cada um resultando na Figura 6, onde é possível comparar as soluções com relação à estabilidade de seus *clusters* em particular. Observa-se nas soluções com k=4 e k=5 que o último *cluster* possui grande instabilidade,

podendo chegar à valores de Jaccard próximos de 0,4. Já na solução com k=6, a estabilidade do último *cluster* varia com menos intensidade. A solução ainda apresenta uma instabilidade no primeiro *cluster*, indicando uma possível repartição de um *cluster* grande e instável em dois *clusters* menores e mais estáveis.

Figura 6– *Boxplot* de Jaccard para cada *cluster* de cada solução



Fonte: elaborado pelo autor.

3.2.6 Estabilidade SLSa

Outro método de análise de possíveis soluções com respeito à quantidade de *clusters* é o Segment Level Stability across solutions (SLSa), apresentado por Dolnicar e Leisch (2017), que avalia a estabilidade à nível de *cluster* ao longo de várias soluções, e permite identificar mudanças nas estruturas dos *clusters* como junções e separações, providenciando informações sobre o histórico de um *cluster* referente à sua composição. Este método aplica o conceito de *relabeling* e utiliza a medida de entropia formulada por Shannon (1948).

Dolnicar e Leisch (2017) denotam que para implementar o algoritmo SLSa efetivamente, é necessária a aplicação de *relabeling*, que remete à nomeação consistente de *clusters* ao longo de possíveis soluções. Mais especificamente, é o ato de identificar *clusters* iguais pertencentes a soluções diferentes e atribuir o mesmo nome a eles, de maneira que o seu rastreamento se faça possível. Para aplicar o *relabeling* em um conjunto de soluções, Dolnicar e Leisch (2017) propõem o algoritmo descrito no Quadro 10 do Apêndice G. Para o conjunto de dados utilizado neste trabalho, apesar do foco estar nas soluções candidatas com k=4, k=5 e k=6, optou-se por aplicar o *relabeling* nas soluções com k=2 até k=9 para um melhor entendimento do processo de formação dos *clusters*.

De acordo com Dolnicar e Leisch (2017), a medida de entropia representa a incerteza em uma distribuição de probabilidades (p_1, p_2, p_3, ... p_k). É descrita pela Equação 8, onde p_j é a distribuição de probabilidades em questão. O valor de entropia máximo consiste numa distribuição de probabilidades onde todos os valores são iguais, resultando num valor de entropia H=1. O valor de entropia mínimo consiste numa distribuição de probabilidades onde somente um dos valores é 1, [0,0,1,0,0] por exemplo, resultando num valor de entropia H=0 e, no contexto do algoritmo, sinalizando que todos os dados de um *cluster* em uma solução são os mesmos que todos os dados de outro *cluster* de uma solução anterior.

$$H = -\sum_{i=1}^{n} p_j \log p_j \tag{8}$$

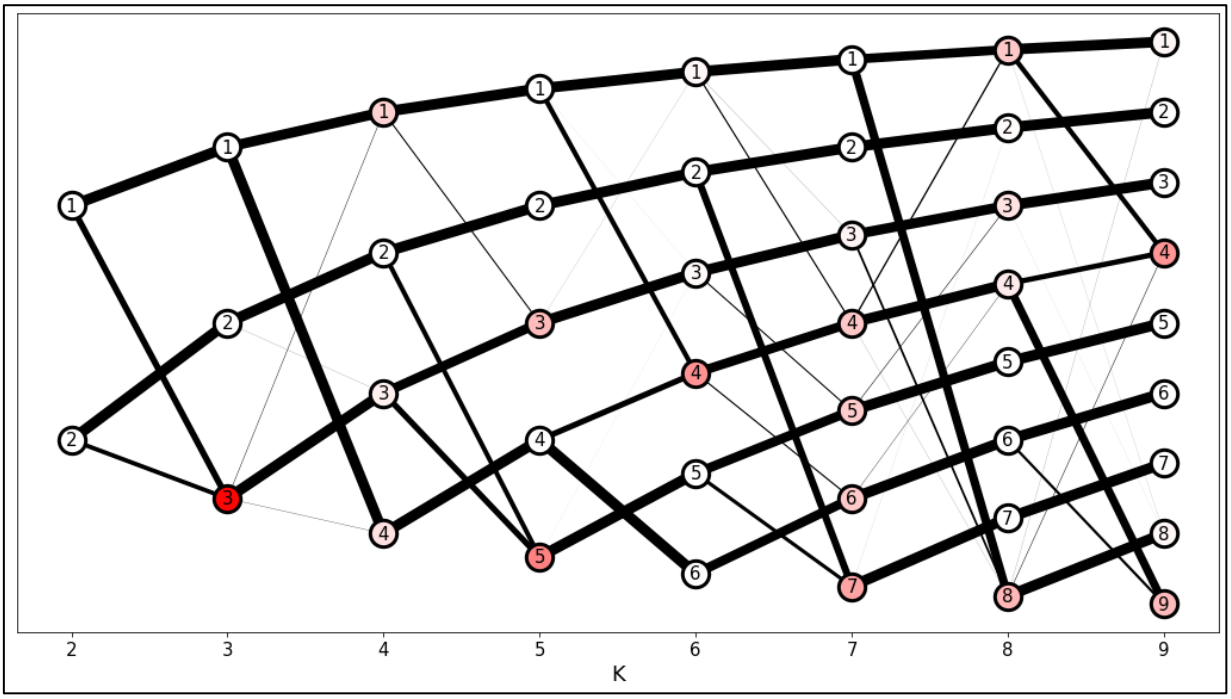
Para aplicar o cálculo SLSa de Dolnicar e Leisch (2017), é necessário calcular a medida de entropia H de cada *cluster* S_l^i (*cluster l* pertencente à solução *i*) em relação a todos os *clusters* da solução anterior S_j^{i-1} (*clusters j* = 1, ... k_{i-1} pertencentes à solução anterior *i-1*). Logo, o valor SLSa de um segmento *l* pertencente à uma solução com k_i segmentos é definido pela Equação 9, onde um valor mínimo de 0 representa a pior estabilidade possível, enquanto 1 indica a melhor estabilidade possível. Em suma, um *cluster* com SLSa = 1 equivale à um *cluster* que não foi formado a partir de outros *clusters*, e sim perdurou ao longo das soluções de k segmentos, enquanto um *cluster* com SLSa = 0 foi criado a partir de dois ou mais *clusters* na solução *k*-1 anterior.

$$SLSa(S_l^i) = -1\frac{H}{\log(k_{i-1})} \tag{9}$$

No Apêndice H encontra-se o código-fonte referente ao índice SLSa, expondo cada passo implementado. Após o cálculo de SLSa para cada *cluster* de cada solução até k=9, é possível representar os valores no grafo exibido na Figura 7, partindo de uma solução com dois *clusters* no canto esquerdo e terminando em uma solução com nove *clusters* no canto direito. *Clusters* com baixo valor SLSa estão coloridos com um tom de vermelho conforme sua instabilidade. As linhas em preto representam a totalidade de clientes pertencentes à um *cluster* que são atribuídos a outro *cluster* na solução seguinte, linhas grossas indicam uma quantidade maior, e muitas linhas à esquerda de um *cluster* indica que ele foi gerado

a partir de vários outros. É possível observar que o *cluster* de número 3 na solução de três *clusters* possui um nível alto de instabilidade, visto que ele foi criado a partir dos dados do *cluster* 1 e 2 na solução anterior (representando efetivamente metade de cada *cluster* da solução anterior). Outros *clusters* seguem o mesmo comportamento, mais especificamente, os *clusters* criados a partir de uma nova solução (os últimos *clusters* de cada coluna) na maioria das vezes são produtos de uma junção de partes de outros *clusters*.

Figura 7– Grafo SLSa de soluções com k=2 a k=9

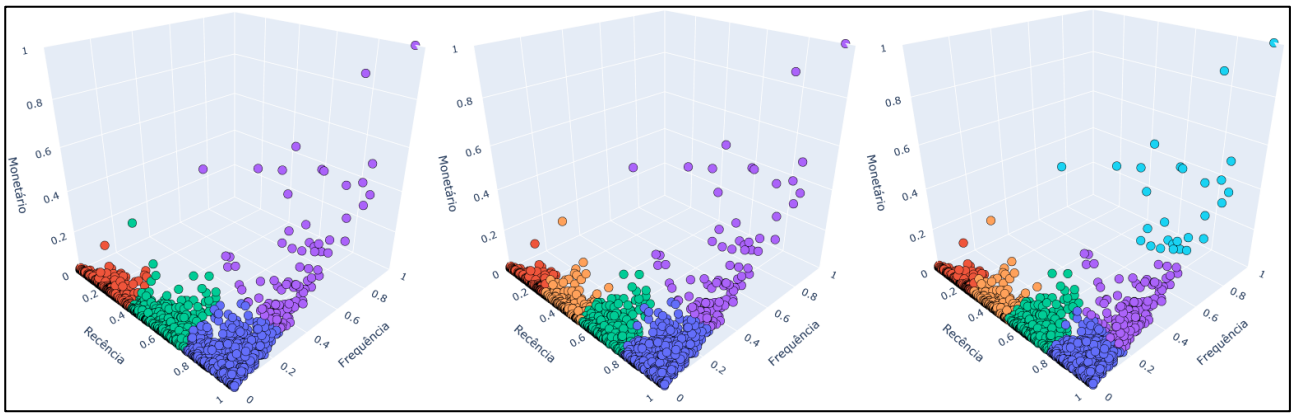


Fonte: elaborado pelo autor.

Nota-se que após a solução 6, quase todos os *clusters* das soluções seguintes apresentam alguma quantidade de instabilidade, sendo formados a partir de dois ou mais *clusters* em soluções anteriores salvo algumas exceções. Das soluções candidatas (4,5 e 6) somente a 6 apresenta uma distribuição satisfatória de *clusters* estáveis, com cinco *clusters* tendo somente um pai na solução anterior.

Para uma melhor visualização, a Figura 8 expõe a transição dos *clusters* ao longo das diferentes soluções candidatas. É possível notar que o *cluster* 5 (em laranja) foi criado na solução 5 a partir de dados provenientes dos *clusters* 2 (em vermelho) e 3 (em verde). Da mesma forma, o *cluster* 6 (em ciano) na solução 6 foi criado a partir de metade dos dados do *cluster* 4 (em lilás), que consequentemente foi deslocado em direção ao *cluster* 1 (em roxo), resultando na aparente "junção" entre duas metades de *clusters.*

Figura 8 – Gráfico dos *clusters* das soluções 4,5 e 6



Fonte: elaborado pelo autor.

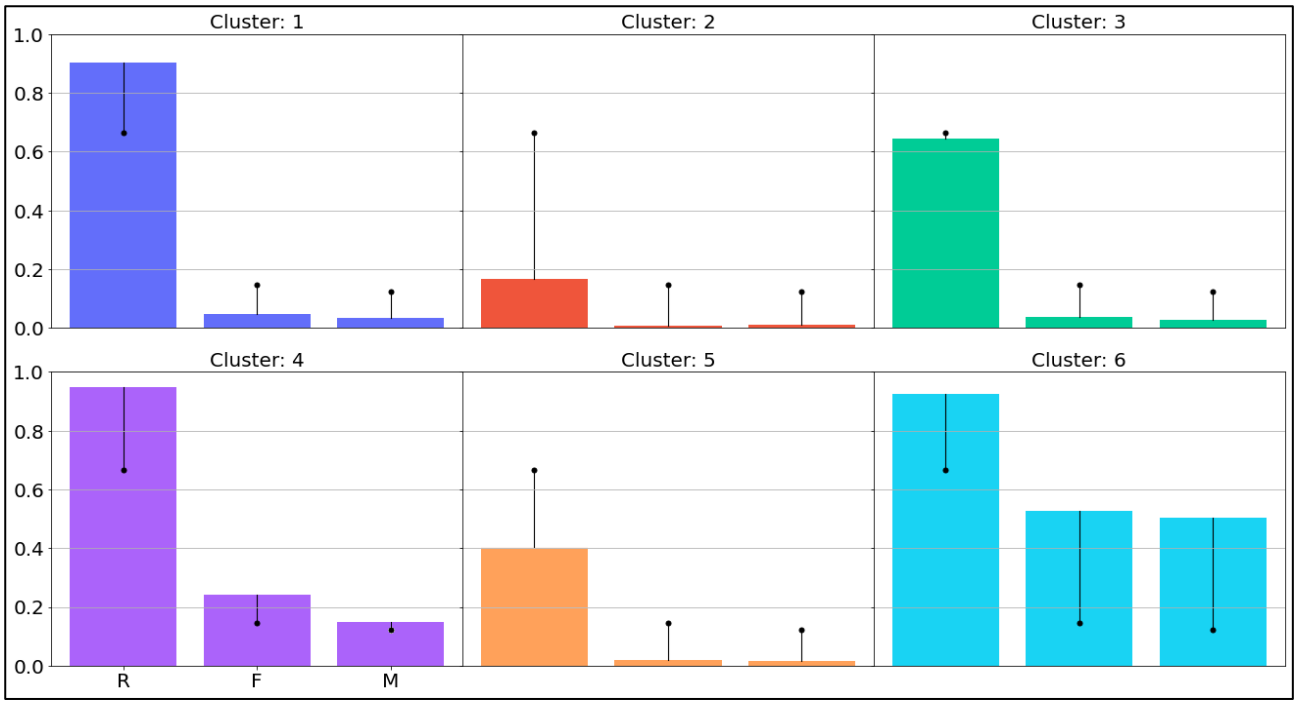
Logo mais, os clientes do *cluster* 6 (em ciano) são completamente absorvidos em outro *cluster* nas soluções menores, apesar de apresentarem características únicas como o fato de possuir os três atributos RFM altos em comparação ao resto dos *clusters*. Portanto, foi escolhida a solução com 6 *clusters*, pois representa de maneira satisfatória todos os

tipos de clientes presentes no conjunto de dados, bem como dispõe uma estabilidade global aceitável (acima de 0,95) e uma instabilidade por *cluster* tolerável (somente um *cluster* formado a partir de deslocamentos).

3.2.7 Perfil dos Clusters

Uma vez obtida a solução desejada, é necessário analisar os *clusters* contidos nela, de maneira que seja facilmente entendido seu perfil e quais características são realmente relevantes. Witschel, Loo e Riesen (2015) indagam que antes de se beneficiar dos resultados, um analista precisa entender a essência de cada *cluster*, ou seja, quais são as características compartilhadas entre os clientes de um *cluster* que os diferenciam dos demais. A partir desta ideia, foi criado um gráfico de barras que apresenta a média de cada característica RFM de cada *cluster* contido na solução k=6, exposto na Figura 9. Cada barra representa um atributo RFM, e sua altura é definida pela média do atributo em questão no cluster. Nesta representação, cada atributo possui um ponto preto referente à média da solução inteira, permitindo comparar se o atributo do *cluster* realmente se destaca em relação a todos os outros.

Figura 9 – Gráfico de perfil dos *clusters* da solução k=6



Fonte: elaborado pelo autor.

Ao analisar a Figura 9 levando em conta cada atributo RFM, é possível ter as seguintes interpretações:

a) Os *clusters* 2 e 5 possuem uma recência, frequência e atributo monetário abaixo da média geral, indicando possivelmente um tipo de cliente que não frequenta mais a loja (*cluster* 2) ou está em processo de parar de frequentar (*cluster* 5). Os *clusters* 2 e 5 possuem respectivamente 390 e 338 clientes, representando cerca de 41% de todos os clientes cadastrados;
b) Os *clusters* 1 e 3 apresentam uma recência alta, porém frequência e monetário baixos, indicando um tipo de cliente novo que ainda não é familiar à loja, ou está em processo de desenvolver uma relação de visitas frequentes, ou até um cliente antigo que frequentou a loja recentemente. De qualquer maneira, estes *clusters* podem representar o fluxo de clientes que compraram recentemente na loja. Os *clusters* 1 e 3 possuem respectivamente 474 e 379 clientes, representando cerca de 48% de todos os clientes cadastrados;
c) Os *clusters* 4 e 5 apresentam atributos RFM acima da média, indicando clientes leais que compram frequentemente e gastam um dinheiro total alto em relação aos outros. O *cluster* 6 em particular, apresenta os maiores valores entre todos os *clusters*, representando os melhores clientes da loja. Os seus atributos RFM são expressivamente maiores, porém este *cluster* contém somente 28 clientes. O *cluster* 4 também possui menos clientes que os outros *clusters*, com 139 no total. Os dois *clusters* juntos representam um total de 167 clientes, cerca de 11% de todos os clientes cadastrados.

Com as informações geradas pelos perfis dos *clusters*, é possível obter um resumo sucinto dos tipos de clientes que frequentam a empresa, sendo estes: clientes perdidos (com baixa recência, frequência e monetário), clientes em processo de perda (com recência abaixo da média, frequência e monetário baixos), clientes recentes (com alta recência, porém frequência e monetário baixos), clientes menos recentes (com alta recência, porém menor que os clientes recentes,

e uma frequência e monetário mais baixos que os clientes recentes), clientes leais (recência, frequência e monetário altos) e por fim os melhores clientes (melhores atributos RFM possíveis).

3.2.8 Descrição dos Clusters

A partir da análise de variáveis de segmentação, torna-se viável a implementação de campanhas promocionais, ações de incentivo e até métodos de resgate de clientes perdidos. Contudo, a análise não está necessariamente finalizada, segundo Dolnicar, Leisch e Grün (2018), um dos passos importantes após obter os perfis de *clusters* é o processo de descrição. A descrição de *clusters* consiste na análise individual dos *clusters* a partir de variáveis externas ao processo de clusterização, chamadas de variáveis descritivas. Estas variáveis podem conter informações como: idade, sexo, localização, padrão de compra, informações provenientes de questionários, e outras características pertinentes ao âmbito da empresa.

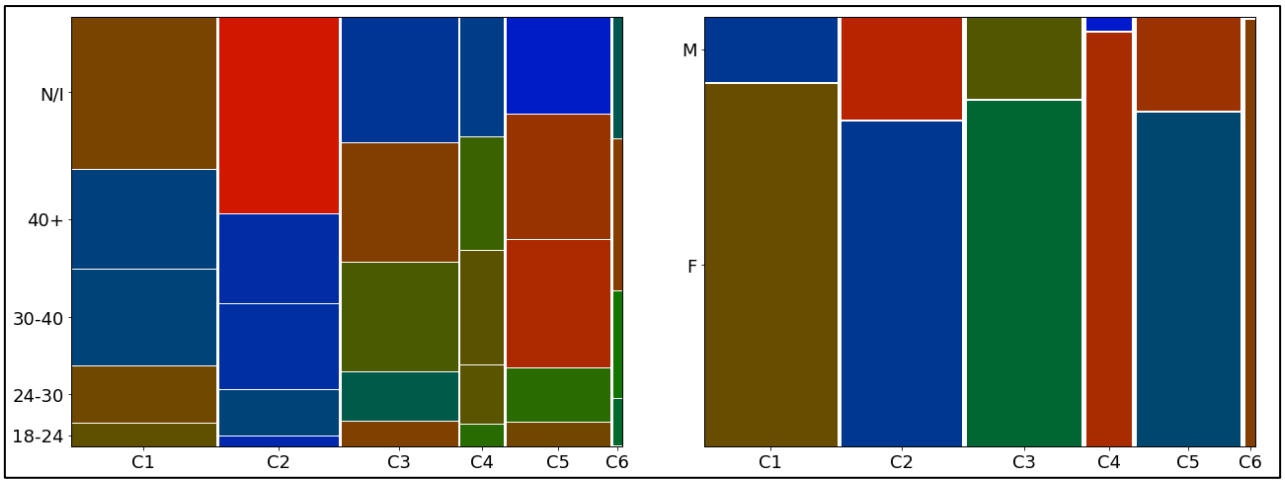
Como a base de dados possui várias informações elegíveis, foram escolhidas cinco para o processo de descrição dos *clusters*: idade, sexo, tempo de contato com a loja, quantidade de compras por estação e taxa de devoluções. Após a extração dos dados descritivos, foram utilizados gráficos de mosaico para a exposição. Este tipo de gráfico é semelhante ao gráfico de barras, porém exibe as informações em células que possuem o seu tamanho referente à quantidade de informações observadas, podendo variar em largura conforme a quantidade de clientes/compras em um *cluster* e em altura conforme o percentual da variável observada em comparação ao percentual das outras variáveis.

Outro conceito pertinente ao gráfico de mosaico é o modelo estatístico aplicado chamado de distribuição bimodal, que exibe variações anormais na distribuição de valores baseando-se numa hipótese de independência das variáveis. Desta maneira, valores maiores que o esperado (acima de dois desvios padrões, ou fora do limite de 95% dos valores) são exibidos em tons vermelhos de maior intensidade, valores menores que o esperado são exibidos em tons azuis de maior intensidade, e valores normais adquirem a cor verde. Com esta visão é possível observar características únicas de *clusters* que possuem variações anormais.

Em relação às informações descritivas utilizadas, a variável de idade foi transformada em uma variável ordinal. Esta variável parte dos 18 a 24 anos, considerando intervalos de idade de seis anos em diante para cada categoria, sendo a penúltima para clientes com mais de 40 anos e a última uma categoria representa uma falta de informação no cadastro. A variável de sexo disponível no banco de dados consiste nas categorias “masculino” e “feminino”.

O resultado dos gráficos aplicados a estas variáveis pode ser observado na Figura 10, que apresenta o gráfico de idade à esquerda e o gráfico de sexo à direita, cada gráfico exibe no eixo vertical as categorias das variáveis descritivas analisadas e no eixo horizontal os *clusters*. Como a distribuição das células ocorre de acordo com a variável observada e a quantidade de observações no *cluster*, o tamanho de cada uma varia em largura e altura. Tomando o *cluster* 6 (C6) como exemplo, sua largura é fina devido à baixa quantidade de clientes que possui, e a altura de cada célula pertencente à ele depende do percentual que cada categoria representa em relação as outras categorias do mesmo *cluster*, caso uma categoria possua 99% dos clientes, ela ocupará 100% da célula, como no *cluster* 6 (C6) do gráfico à direita.

Figura 10 – Gráfico de mosaico das idades (esquerdo) e sexo (direito)



Fonte: elaborado pelo autor.

Ao analisar a Figura 10, é possível ter as seguintes interpretações:

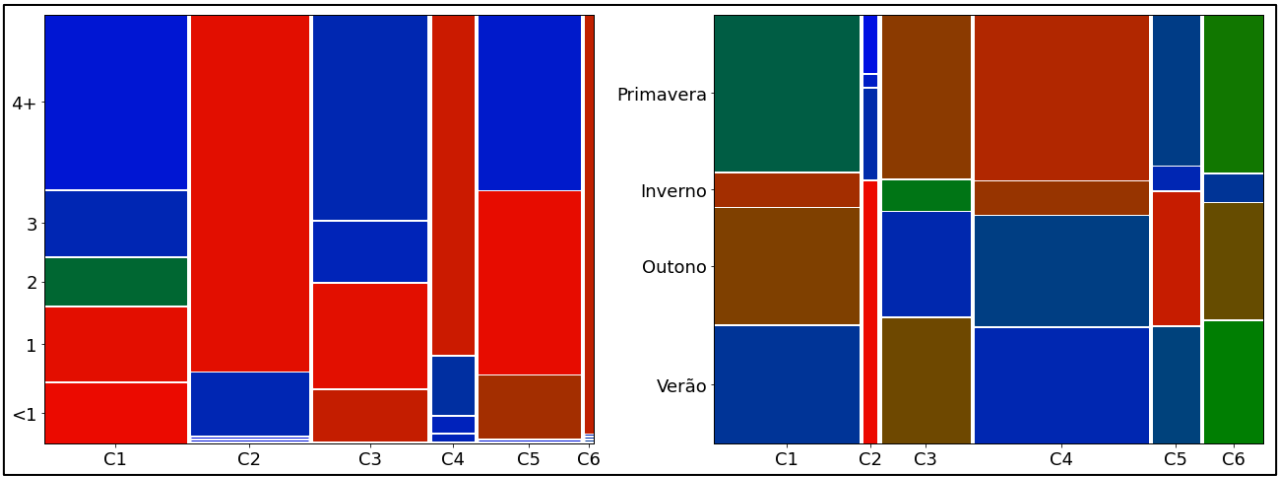
a) em relação à idade (gráfico à esquerda), o *cluster* de clientes recentes (C1) possui número menor que esperado (células em azul) de clientes adultos e idosos e maiores concentrações de jovens adultos e clientes sem informação, indicando que pode haver um fluxo de pessoas jovens sendo atraídas pela loja. Já o *cluster* dos clientes perdidos (C2) possui um número maior que esperado de clientes que não informaram idade,

indicando uma certa resistência com preenchimento de cadastros. O *cluster* de clientes sendo perdidos (C5) possui uma quantidade maior que esperada de clientes com idade maior que 30, indicando uma possível insatisfação com os produtos oferecidos à essa faixa etária, informação que é corroborada pelo fato de que o fluxo de clientes recentes (C1) possui mais jovens que o esperado.

b) em relação ao sexo (gráfico à direita), os clientes mais importantes (pertencentes aos *clusters* C4 e C6) são majoritariamente mulheres e estão em quantidade acima do esperado, mesmo com a loja oferecendo linhas masculinas, indicando uma preferência feminina pelas roupas oferecidas. Esta informação é corroborada pelo fato de que os *clusters* com clientes perdidos ou em processo de serem perdidos (C2 e C5) possuírem uma maior quantidade de homens que mulheres, indicando uma possível falta de engajamento masculino com as opções oferecidas.

As duas outras variáveis que permitem uma exibição através de mosaico são: a quantidade de anos desde o cadastro de um cliente na empresa e a quantidade de compras realizada durante cada estação. Para a primeira, foi estabelecido o intervalo de menos de um ano (<1), um, dois, três e mais de quatro anos. Para a segunda o intervalo é composto pelas quatro estações (Verão, Outono, Inverno e Primavera). Os gráficos gerados são exibidos na Figura 11, que segue a mesma estrutura da figura anterior.

Figura 11 – Gráfico de mosaico do tempo de cadastro (esquerdo) e total de compras por estação (direito)



Fonte: elaborado pelo autor.

A partir da análise da Figura 11, é possível ter as seguintes interpretações:

a) em relação ao tempo de cadastro dos clientes (gráfico à esquerda), é possível identificar que os *clusters* com clientes recentes (C1 e C3) possuem uma quantidade maior de clientes recém cadastrados que o normal, assim como clientes com um ano de cadastro, permitindo identificar que estes *clusters* apresentam um fluxo de clientes novos. Os *clusters* com os melhores clientes (C4 e C6) possuem muitos clientes cadastrados a mais de quatro anos (no *cluster* C6 são todos os clientes), indicando que os clientes com desempenho RFM bom raramente são clientes novos, necessitando uma longa relação com a loja. Por fim, os *clusters* C2 e C5 que representam clientes perdidos ou em processo de perda, apresentam grande quantidade de clientes cadastrados a mais de quatro anos, fato que justifica a característica de clientes perdidos.

b) em relação à quantidade de compras por estação (gráfico à direita), o gráfico expõe as preferências de cada *cluster* em relação à estações específicas, exibindo uma preferência geral pelas coleções de Inverno, Verão e Outono. O *cluster* de clientes perdidos (C2) apresenta uma grande taxa de compras feitas durante o verão, podendo indicar uma certa insatisfação com a linha desta estação, visto que os clientes deste *cluster* não frequentam mais a loja. O *cluster* com segundo melhor desempenho RFM (C4), apresenta a maior quantidade de compras de todos os outros *clusters* (denotado pela largura das células), destas vendas, maior do que o normal foi a frequência de compras na primavera, indicando uma preferência deste grupo pela linha desta estação.

A última variável analisada, taxa de devoluções de compras (transação ou venda que contém pelo menos uma devolução), foi obtida através da razão entre a quantidade de devoluções de um *cluster* e a sua quantidade de vendas total. Desta maneira, a Tabela 3 exibe os percentuais de devolução de cada *cluster*.

Tabela 3 – Taxa de devoluções por *cluster*

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
9,09%	6,93%	8,11%	11,81%	8,68%	17,50%

Fonte: elaborado pelo autor.

Com base nos percentuais apresentados, nota-se que os *clusters* com melhor desempenho RFM (*cluster* 4 e 6) possuem as maiores taxas de devolução (11,81% e 17,50% respectivamente), indicando uma alta seletividade entre seus clientes. O *cluster* de clientes perdidos (*cluster* 2) apresenta a menor taxa de devolução (6,93%), indicando que um cliente insatisfeito raramente realiza uma devolução, e simplesmente não frequenta mais a loja ao invés de trocar o produto e tentar comprar novamente.

4 CONCLUSÃO

A segmentação de clientes permite uma análise aprofundada do comportamento dos clientes de uma empresa. Com os dados certos, perfis antes obscuros podem ser identificados, a partir de informações outras vezes consideradas sem utilidade além da camada operacional de vendas e cadastros de uma empresa. Este trabalho teve como iniciativa a numeração e identificação destes perfis, para isso, utilizou-se o banco de dados de uma empresa real de varejo de roupas, contendo informações cadastrais e transacionais de 1845 clientes. Foi atribuído a cada cliente suas características com base no modelo RFM, sendo realizada posteriormente a limpeza e manipulação dos dados, de maneira que se adequem ao algoritmo de clusterização utilizado, K-means.

Para a validação da solução de *clusters* bem como sua quantidade, foram utilizados três índices internos de validação (Silhouette, Calinski-Harabasz e Davies-Bouldin), e quando eles não foram conclusivos o bastante para a definição da quantidade, utilizou-se os seguintes índices externos de validação: medida de estabilidade global com base no índice ARI, medida de estabilidade por *cluster* com base no índice de Jaccard, e o método SLSa a partir da medida de entropia. Após selecionar três soluções candidatas (com 4, 5 e 6 *clusters*) a partir da estabilidade global, a estabilidade por *cluster* apresentou melhor resultado na solução com 6 *clusters*, sendo então confirmada e detalhada a partir do método SLSa, demonstrando o processo de divisão e junção dos *clusters* ao longo das iterações com diferentes números para o parâmetro *k* do algoritmo K-means.

Desta forma, a solução com 6 *clusters* foi escolhida, e seus *clusters* foram apresentados num gráfico contendo suas características RFM, de maneira que seus perfis fossem detectados com base nas inferências realizadas a partir de seus atributos. Com o perfilamento dos *clusters*, foram nomeados seis segmentos com base nas suas peculiaridades: clientes perdidos (com baixa recência, frequência e monetário), clientes em processo de perda (com recência abaixo da média, frequência e monetário baixos), clientes recentes (com alta recência, porém frequência e monetário baixos), clientes menos recentes (com alta recência, porém menor que os clientes recentes, e uma frequência e monetário mais baixos que os clientes recentes), clientes leais (recência, frequência e monetário altos) e por fim os melhores clientes (melhores atributos RFM possíveis).

Após o destaque do perfil de cada segmento através das variáveis de segmentação RFM, realizou-se uma análise a partir de variáveis descritivas com base nos dados disponíveis na base de dados. Os segmentos foram avaliados através de gráficos de mosaico e tabelas com base na sua idade, sexo, tempo de cadastro, compras por estação e devoluções, sendo apontadas particularidades presentes em cada variável descritiva, como possíveis tendências dos segmentos, fluxos anormais, quantidades fora do padrão, dentre outras.

Desta maneira, o objetivo de identificar diferentes segmentos de clientes com base em seu comportamento foi atingido. Apesar dos índices de validação interna não apresentarem um consenso entre o número de *clusters* naturais, foi possível obter uma garantia de estabilidade dos segmentos através dos índices externos. Dito isso, é evidente que apesar de não existirem *clusters* naturais, ainda assim foi possível obter segmentos significativos, contendo características destacáveis que os diferenciam entre si, permitindo discernimentos posteriores sobre os tipos de clientes que frequentam o estabelecimento, extrapolando para os tipos de clientes em geral do ramo de varejo.

Ademais, o presente trabalho contribui para a comunidade acadêmica, pela aplicação de modelos (RFM), índices (três internos e três externos), métodos (normalização Min-Max, bootstraping, Índice de Jaccard e ARI) e algoritmo K-means, em uma base de dados real, analisando sua influência em dados com uma distribuição diferente de dados de treinamento (cujas características comumente apresentam *clusters* bem definidos, diferentemente de uma base com dados reais). Uma conclusão derivada da aplicação de tais técnicas a este conjunto de dados é de que nem sempre os índices de validação internos apresentam um consenso sobre a quantidade de *clusters*, necessitando a utilização de outros tipos de validação. Além disso, foi demonstrado que informações valiosas para o setor de varejo de roupas e possivelmente outros setores podem ser extraídas de uma base de dados com informações transacionais e cadastrais, indicando um valor intrínseco à dados que muitas vezes são somente armazenados e raramente analisados em contexto de *clusters* de clientes.

Diante do exposto, o presente trabalho pode ser complementado a partir das seguintes propostas: utilização do método RFM em conjunto com K-means aplicado à uma base de dados de um ramo diferente de varejo, como por exemplo, supermercados, concessionárias, imobiliárias, dentre outras; aplicação de diferentes índices internos e externos para a validação da qualidade dos *clusters* sob diferentes visões; utilização de outras variáveis descritivas, como tempo gasto por compra, linhas de produtos mais comprados e quantidade de produtos por compra; aplicação de questionários, para utilizar em conjunto com a análise dos perfis, cruzando as variáveis com base no *cluster* questionado.

REFERÊNCIAS

ALELYANI, Salem; TANG, Jiliang; LIU, Huan. Feature Selection for Clustering: a review. In: AGGARWAL, Charu C.; REDDY, Chandan K. (ed.). **Data Clustering**: algorithms and applications. [S.l.]: Chapman And Hall/Crc, 2014. Cap. 2. p. 29-61.

ARBELAITZ, Olatz *et al*. An extensive comparative study of cluster validity indices. **Pattern Recognition**, [S.l.], v. 46, n. 1, p. 243-256, jan. 2013. Elsevier BV. Disponível em: http://dx.doi.org/10.1016/j.patcog.2012.07.021 Acesso em: 14 jun. 2022.

C.SARANYA; G.MANIKANDAN. A Study on Normalization Techniques for Privacy Preserving Data Mining. **International Journal Of Engineering And Technology**, [S.l.], v. 5, n. 3, p. 2701-2704, jun. 2013. Disponível em: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.411.1996&rep=rep1&type=pdf. Acesso em: 18 jun. 2022.

CALINSKI, T.; HARABASZ, J. A dendrite method for cluster analysis. **Communications In Statistics - Theory And Methods**, [S.l.], v. 3, n. 1, p. 1-27, 1974. Informa UK Limited. http://dx.doi.org/10.1080/03610927408827101.

CHERKASSKY, Vladimir S.; MULIER, Filip. Methods for data reduction and dimensionality reduction. In: CHERKASSKY, Vladimir S.; MULIER, Filip. **Learning from data**: concepts, theory, and methods. 2. ed. Hoboken: Ieee Press, 2007. Cap. 6, p. 191.

CHRISTY, A. Joy *et al*. RFM ranking – An effective approach to customer segmentation. **Journal Of King Saud University - Computer And Information Sciences**, [S.l.], v. 33, n. 10, p. 1251-1257, dez. 2021. Elsevier BV. Disponível em: http://dx.doi.org/10.1016/j.jksuci.2018.09.004. Acesso em: 10 jun. 2022.

DRASZAWKA, Karol; SZYMAńSKI, Julian. External Validation Measures for Nested Clustering of Text Documents. In: RYŜKO, Dominik *et al* (ed.). **Emerging Intelligent Technologies in Industry**. [S.l.]: Springer Berlin, 2011. Cap. 4. p. 207-225. Disponível em: https://link.springer.com/book/10.1007/978-3-642-22732-5. Acesso em: 12 jun. 2022.

DAVIES, David L.; BOULDIN, Donald W. A Cluster Separation Measure. **Ieee Transactions On Pattern Analysis And Machine Intelligence**, [S.l.], v. -1, n. 2, p. 224-227, abr. 1979. Institute of Electrical and Electronics Engineers (IEEE). Disponível em: http://dx.doi.org/10.1109/tpami.1979.4766909. Acesso em: 01 maio 2022.

DOLNICAR, Sara; LEISCH, Friedrich. Using segment level stability to select target segments in data-driven market segmentation studies. **Marketing Letters**, [S.L.], v. 28, n. 3, p. 423-436, 1 mar. 2017. Disponível em: https://www.researchgate.net/publication/314165290_Using_segment_level_stability_to_select_target_segments_in_data-driven_market_segmentation_studies. Acesso em: 09 jun. 2022.

DOLNICAR, Sara; LEISCH, Friedrich; GRÜN, Bettina. **Market Segmentation Analysis:** Understanding It, Doing It, and Making It Useful. [S.L.]: Springer Nature, 2018. 324 p. Disponível em: https://library.oapen.org/handle/20.500.12657/51281. Acesso em: 11 jun. 2022.

ERNST, Dominik; DOLNICAR, Sara. How to Avoid Random Market Segmentation Solutions. **Journal Of Travel Research**, [S.L.], v. 57, n. 1, p. 69-82, 6 jan. 2017. Disponível em: https://www.researchgate.net/publication/312149300_How_to_Avoid_Random_Market_Segmentation_Solutions. Acesso em: 11 jun. 2017.

FRÄNTI, Pasi; SIERANOJA, Sami. How much can k-means be improved by using better initialization and repeats? **Pattern Recognition**, [S.L.], v. 93, n. 1, p. 95-112, set. 2019. Disponível em: https://doi.org/10.1016/j.patcog.2019.04.014. Acesso em: 21 jun. 2022.

GHOSH, Soumi; KUMAR, Sanjay. Comparative Analysis of K-Means and Fuzzy C-Means Algorithms. **International Journal Of Advanced Computer Science And Applications**, [S.L.], v. 4, n. 4, p. 35-39, 2013. The Science and Information Organization. http://dx.doi.org/10.14569/ijacsa.2013.040406. Disponível em: https://thesai.org/Publications/ViewPaper?Volume=4&Issue=4&Code=IJACSA&SerialNo=6. Acesso em: 07 nov. 2021.

GUSTRIANSYAH, Rendra; SUHANDI, Nazori; ANTONY, Fery. Clustering optimization in RFM analysis Based on k-Means. **Indonesian Journal Of Electrical Engineering And Computer Science**, [S. l], v. 18, n. 1, p. 470-477, abr. 2020. Mensal. Disponível em: http://ijeecs.iaescore.com/index.php/IJEECS/article/view/20264. Acesso em: 02 set. 2021.

HÄMÄLÄINEN, Joonas; JAUHIAINEN, Susanne; KÄRKKÄINEN, Tommi. Comparison of Internal Clustering Validation Indices for Prototype-Based Clustering. **Algorithms**, [S.l.], v. 10, n. 3, p. 105-119, 6 set. 2017. MDPI AG. http://dx.doi.org/10.3390/a10030105.

HENNIG, Christian. Cluster-wise assessment of cluster stability. **Computational Statistics & Data Analysis**, [S.L.], v. 52, n. 1, p. 258-271, set. 2007. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0167947306004622. Acesso em: 11 jun. 2022.

HIDAYAT, Syahroni *et al*. Segmentation of university customers loyalty based on RFM analysis using fuzzy c-means clustering. **Jurnal Teknologi Dan Sistem Komputer**, [S.L.], v. 8, n. 2, p. 133-139, 11 mar. 2020. Institute of Research and Community Services Diponegoro University (LPPM UNDIP). http://dx.doi.org/10.14710/jtsiskom.8.2.2020.133-139. Disponível em: https://jtsiskom.undip.ac.id/index.php/jtsiskom/article/view/13352. Acesso em: 09 jun. 2022.

HUGHES, Arthur M. **Strategic Database Marketing 4e**: the masterplan for starting and managing a profitable, customer-based marketing program. 4. ed. [S. l.]: McGraw-Hill, 2011. 608 p.

KUMAR, V. **Managing Customers for Profit**: strategies to increase profits and build loyalty. Upper Saddle River: Pearson Prentice Hall, 2008. 296 p.

LEE, Shinho *et al*. Android Malware Similarity Clustering using Method based Opcode Sequence and Jaccard Index. In: 2019 INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGY CONVERGENCE (ICTC), 10., 2019, Jeju. **Proceedings…** [S.L.]: Ieee, 2019. p. 178-183. Disponível em: https://ieeexplore.ieee.org/abstract/document/8939894. Acesso em: 11 jun. 2022.

LIU, Yanchi *et al*. Understanding of Internal Clustering Validation Measures. In: 2010 IEEE INTERNATIONAL CONFERENCE ON DATA MINING, 10., 2010, Sydney. **Proceedings…** [S.L.]: Ieee, 2010. p. 911-916. Disponível em: https://ieeexplore.ieee.org/abstract/document/5694060. Acesso em: 11 jun. 2022.

NGUYEN, Thuyuyen H.; SHERIF, Joseph S.; NEWBY, Michael. **Strategies for successful CRM implementation**. Information Management & Computer Security, [S.l.], v. 15, n. 2, p. 102-115, maio 2007. Disponível em: https://www.emerald.com/insight/content/doi/10.1108/09685220710748001/full/html?journalCode=imcs. Acesso em: 26 set. 2021.

NIJHER, Harshjot Singh. **Exploring Critical Success Factors of ERP Implementation in United Nations Types of Organizations**: relationship between factors impacting user experience. 2014. 183 f. Dissertação (Mestrado) - Curso de Administração, Concordia University, [S.L.], 2014. Disponível em: https://spectrum.library.concordia.ca/id/eprint/979062/. Acesso em: 18 jun. 2014.

MIGLAUTSCH, J R. Thoughts on RFM scoring. **Journal Of Database Marketing & Customer Strategy Management**, [S.L.], v. 8, n. 1, p. 67-72, ago. 2000. Springer Science and Business Media LLC. http://dx.doi.org/10.1057/palgrave.jdm.3240019. Disponível em: https://link.springer.com/article/10.1057/palgrave.jdm.3240019. Acesso em: 07 nov. 2021.

PAPADIMITRIOU, Christos H.; STEIGLITZ, Kenneth. **Combinatorial Optimization**: algorithms and complexity. New Jersey: Prentice Hall, 1982. 496 p.

PEKER, Serhat; KOCYIGIT, Altan; EREN, P. Erhan. LRFMP model for customer segmentation in the grocery retail industry: a case study. **Marketing Intelligence & Planning**, [S.l.], v. 35, n. 4, p. 544-559, 6 maio 2017. Emerald. http://dx.doi.org/10.1108/mip-11-2016-0210. Disponível em: https://www.emerald.com/insight/content/doi/10.1108/MIP-11-2016-0210/full/html. Acesso em: 07 set. 2021.

PETRISON, Lisa A.; BLATTBERG, Robert C.; WANG, Paul. Database marketing: past, present, and future. **Journal Of Direct Marketing**, [S.l.], v. 11, n. 4, p. 109-125, mar. 1997. Wiley. http://dx.doi.org/10.1002/(sici)1522-7138(199723)11:43.0.co;2-g. Disponível em: https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1522-7138(199723)11:4%3C109::AID-DIR12%3E3.0.CO;2-G. Acesso em: 19 set. 2021.

RASHID, Mohammad A.; HOSSAIN, Liaquat; PATRICK, Jon David. The Evolution of ERP Systems: a historical perspective. In: NAH, Fiona Fui-Hoon. **Enterprise Resource Planning**: solutions and management. Hershey: Irm Press, 2001. p. 35-50. Disponível em: https://books.google.com.br/books?id=qBcJwDWk4ioC&lpg=PR1&ots=9MrXoQhaRL&dq=Enterprise%20Resource%20Planning%BR&pg=PR1#v=onepage&q=Enterprise%20Resource%20Planning:%20Solutions%20and%20Management&f=false. Acesso em: 19 set. 2021.

REINARTZ, Werner; THOMAS, Jacquelyn S.; KUMAR, V. Balancing Acquisition and Retention Resources to Maximize Customer Profitability. **Journal Of Marketing**, [S.l.], v. 69, n. 1, p. 63-79, jan. 2005.

RENDÓN, Eréndira *et al*. Internal versus External cluster validation indexes. **International Journal Of Computers And Communications**, [S. L.], v. 5, n. 1, p. 27-34, fev. 2011. Disponível em: http://www.universitypress.org.uk/journals/cc/20-463.pdf. Acesso em: 12 jun. 2022.

ROBERT, Valerie; VASSEUR, Yann; BRAULT, Vincent. Comparing High-Dimensional Partitions with the Co-clustering Adjusted Rand Index. **Journal Of Classification**, [S.L.], v. 38, n. 1, p. 158-186, 14 nov. 2020. Disponível em: link.springer.com/article/10.1007/s00357-020-09379-w. Acesso em: 11 jun. 2022.

ROBERTS, John H.; KAYANDE, Ujwal; STREMERSCH, Stefan. From academic research to marketing practice: exploring the marketing science value chain. **International Journal Of Research In Marketing**, [S.L.], v. 31, n. 2, p. 127-140, jun. 2014. Disponível em: https://doi.org/10.1016/j.ijresmar.2013.07.006. Acesso em: 19 jun. 2022.

ROODMAN, David *et al*. Fast and wild: bootstrap inference in stata using boottest. **The Stata Journal**: Promoting communications on statistics and Stata, [S.L.], v. 19, n. 1, p. 4-60, mar. 2019. Disponível em: https://journals.sagepub.com/doi/abs/10.1177/1536867X19830877. Acesso em: 11 jun. 2022.

ROUSSEEUW, Peter J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. **Journal Of Computational And Applied Mathematics**, [S.l.], v. 20, n. 0, p. 53-65, nov. 1987. Elsevier BV. Disponível em: http://dx.doi.org/10.1016/0377-0427(87)90125-7. Acesso em: 2 jun. 2022.

SAFARI, Fariba; SAFARI, Narges; MONTAZER, Gholam Ali. Customer lifetime value determination based on RFM model. **Marketing Intelligence & Planning**, [S.l.], v. 34, n. 4, p. 446-461, 6 jun. 2016. Emerald. http://dx.doi.org/10.1108/mip-03-2015-0060.

SANTOS, Jorge M.; EMBRECHTS, Mark. On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification. In: ARTIFICIAL NEURAL NETWORKS – ICANN 2009, 19., 2009, Cyprus. **Proceedings…** [S.L.]: Isbn, 2009. p. 175-184.

SHANNON, C. E.. A Mathematical Theory of Communication. **Bell System Technical Journal**, [S.l.], v. 27, n. 3, p. 379-423, jul. 1948. Disponível em: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x. Acesso em: 12 jun. 2022.

SRIVASTAVA, S.K., CHANDRA, B., SRIVASTAVA, P. The Impact of Knowledge Management and Data Mining on CRM in the Service Industry. In: NATH, V., MANDAL, J. (eds.). **Nanoelectronics, Circuits and Communication Systems**. Springer Singapore, 2018. v. 511. p. 37–52. Disponível em: https://doi.org/10.1007/978-981-13-0776-8_4

TAVAKOLI, Mohammadreza *et al*. Customer Segmentation and Strategy Development Based on User Behavior Analysis, RFM Model and Data Mining Techniques: a case study. In: 2018 IEEE 15TH INTERNATIONAL CONFERENCE ON E-BUSINESS ENGINEERING (ICEBE), 15., 2018, Xiam. **Proceedings…** [S.l.]: Ieee, 2018. p. 119-126. Disponível em: https://www.researchgate.net/publication/330027350_Customer_Segmentation_and_Strategy_Development_Based_on_User_Behavior_Analysis_RFM_Model_and_Data_Mining_Techniques_A_Case_Study. Acesso em: 11 set. 2021.

TSIPTSIS, Konstantinos K.; CHORIANOPOULOS, Antonios. **Data Mining Techniques in CRM**: inside customer segmentation. Chichester: John Wiley & Sons, 2009. 374 p.

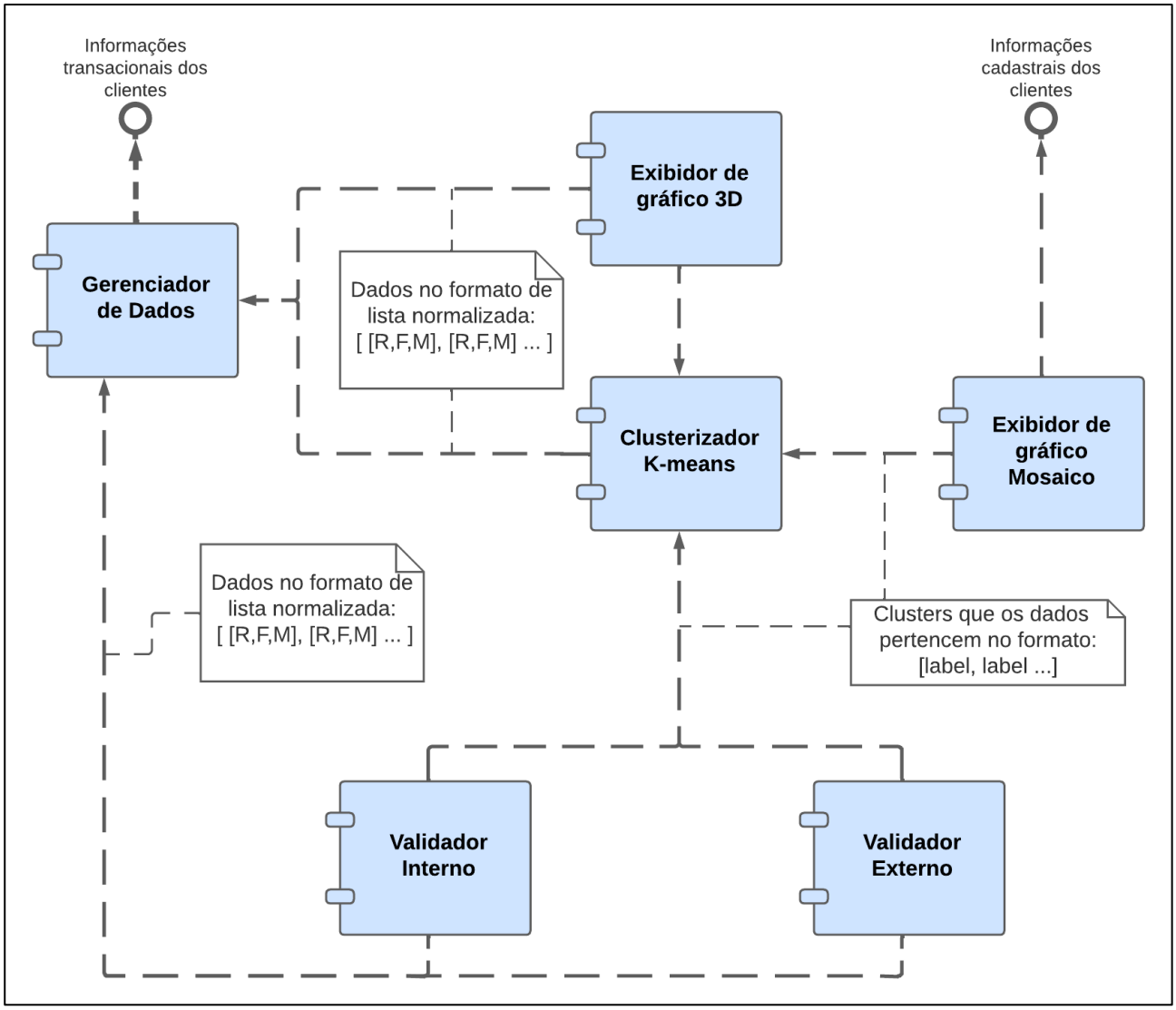
VERHOEF, Peter C et al. The commercial use of segmentation and predictive modeling techniques for database marketing in the Netherlands. **Decision Support Systems**, [S.l.], v. 34, n. 4, p. 471-481, mar. 2003. Disponível em: https://liacs.leidenuniv.nl/~puttenpwhvander/library/Others/segmpredmodel-hoekstra.pdf. Acesso em: 19 set. 2021.

WITSCHEL, Hans Friedrich; LOO, Simon; RIESEN, Kaspar. How to Support Customer Segmentation with Useful Cluster Descriptions. In: ADVANCES IN DATA MINING: APPLICATIONS AND THEORETICAL ASPECTS ICDM 2015, 15., 2015, Hamburg. **Proceedings [...] .** [S.L.]: Springer, 2015. p. 17-31. Disponível em: https://link.springer.com/chapter/10.1007/978-3-319-20910-4_2. Acesso em: 11 jun. 2022.

APÊNDICE A – DIAGRAMA DE COMPONENTES

A Figura 12 apresenta o diagrama dos principais componentes do protótipo de segmentação. O componente primário, dos quais os outros componentes subsequentes dependem, é o `Gerenciador de dados`, que extrai, normaliza e organiza os dados das informações transacionais dos clientes no formato adequado (lista dos clientes, com cada cliente sendo representado pelo conjunto de seus atributos RFM no modelo de lista [R, F, M]). Um componente adicional é o Exibidor de gráfico 3D que utiliza os dados do gerenciador e, dependendo da necessidade, depende dos dados do Clusterizador K-means. O componente principal do protótipo é o Clusterizador K-means, que realiza a clusterização dos clientes com base nos dados informados pelo Gerenciador. Os componentes do tipo Validador, utilizam em conjunto os dados providenciados pelo Gerenciador e pelo Clusterizador, que fornece os *labels* dos clientes (qual *cluster* pertencem, no modelo de lista [label]). Por fim, o componente Exibidor de gráfico Mosaico utiliza as informações cadastrais dos clientes, em conjunto com as informações de qual *cluster* pertencem, providenciadas pelo Clusterizador.

Figura 12 – Diagrama de componentes

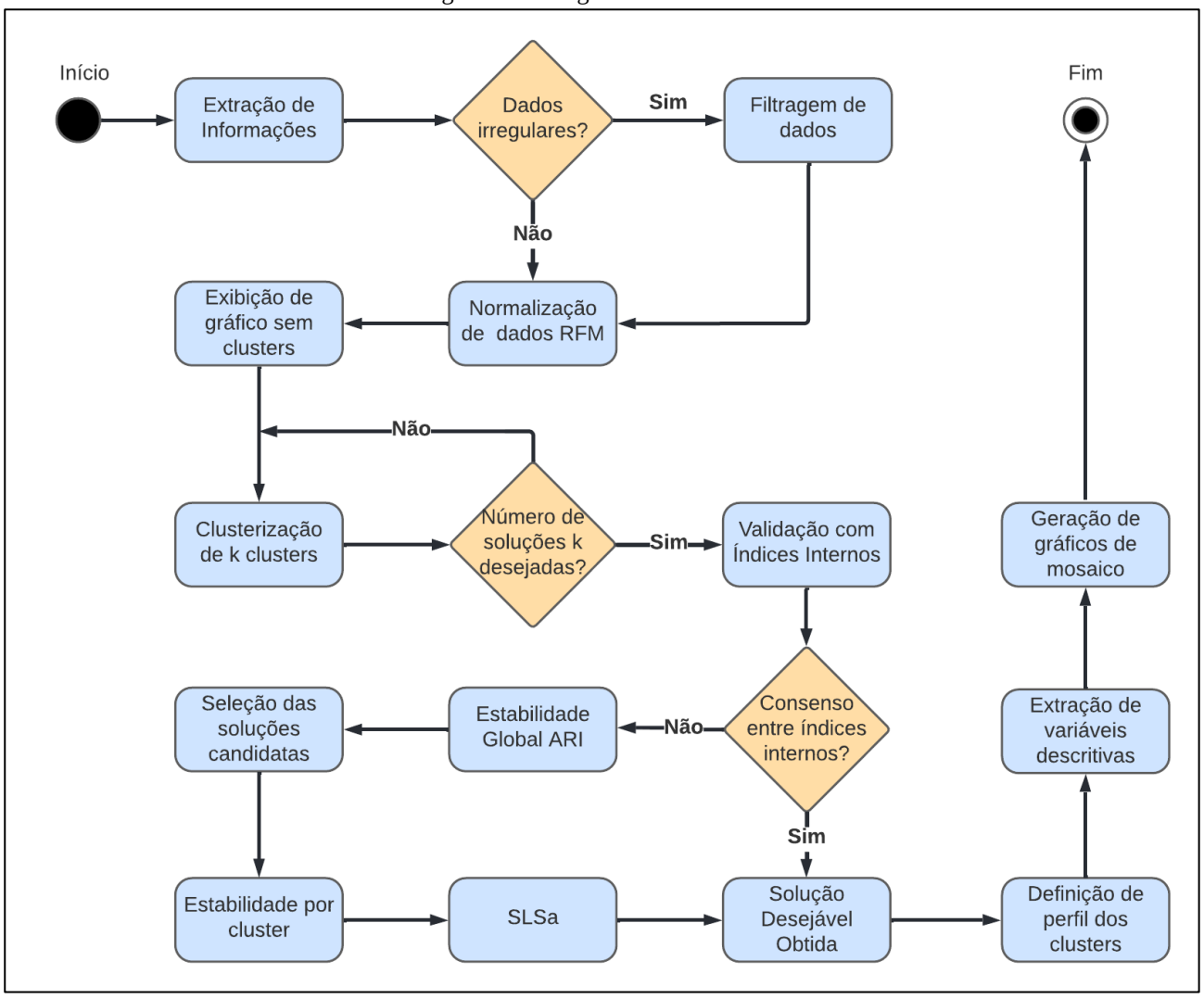


Fonte: elaborado pelo autor.

APÊNDICE B – DIAGRAMA DE ATIVIDADES

A Figura 13 exibe o diagrama do fluxo das principais atividades do protótipo no processo de segmentação. O primeiro passo consiste na extração de informações da base de dados. Caso os dados estiverem irregulares (como vendas vazias, clientes sem vendas, clientes vazios), estes são filtrados. O passo de normalização de dados é necessário para o resto das atividades. É exibido o gráfico dos clientes antes da clusterização para providenciar uma visão sobre a distribuição dos dados. A clusterização consiste em gerar *k clusters*, de acordo com a quantidade *k* desejada de possíveis soluções, tendo cada solução uma quantidade diferente de *clusters*. Após a clusterização, é validado a qualidade das soluções através dos índices internos, que podem chegar num consenso em relação à quantidade *k* ideal de *clusters*. Caso o consenso não aconteça, é necessário validar através de índices externos, começando pela estabilidade global, com isso obtém-se as soluções candidatas, que são submetidas pela análise de estabilidade por cluster auxiliada da estabilidade SLSa. Após uma solução desejável ser obtida, é realizado o perfilamento de cada *clusters* (agora segmento). São extraídas as variáveis descritivas desejadas de cada segmento, e por fim, são exibidas em gráficos de mosaico.

Figura 13 – Diagrama de atividades



Fonte: elaborado pelo autor.

APÊNDICE C – CÓDIGOS-FONTE DE MANIPULAÇÃO DE DADOS

O Quadro 6 mostra as principais ações do código-fonte referente à manipulação dos dados. Estes trechos são executados logo após a extração de dados do banco, que está omitida por escolha, pois revela a estrutura do banco de dados do software de gestão que o criou. A linha 1 é executada logo após esta extração, criando a variável `rfm`, sendo esta uma lista com cada id, recência, frequência e monetário de cada cliente (no formato [`id_cliente, atributo_R, atributo_F, atributo_M`]). Nas linhas 2 a 4, são criadas as principais variáveis que serão utilizadas no resto do protótipo, contendo os atributos (variável `atributos`) no formato [`atributo_R, atributo_F, atributo_M`] e os ids (variável `ids`) contendo somente os identificadores de cada cliente no formato [`id_cliente`]. A linha 6 trata-se do laço de repetição onde são removidos os clientes cujo atributo frequência for zero, removendo então, clientes sem vendas. Por fim, as linhas 15 e 16 utilizam o função `MinMaxScaler()` da biblioteca `sklearn.preprocessing` para transformar a lista de atributos em uma lista normalizada através do método `fit_transform(atributos)`.

Quadro 6 – Códigos-fonte da manipulação dos dados

```
1  rfm.append([cliente[0],recencia,freq, int(soma)])
2  atributos = np.array(rfm)
3  ids = atributos[:,0]
4  atributos = np.array(atributos[:,1:]).astype(float)
5  removeu = True
6  while removeu:
7      removeu = False
8      i = 0
9      for i in range(len(atributos)):
10         if(atributos [i][1] == 0):
11             atributos = np.delete(atributos, (i), axis=0)
12             rfm = np.delete(rfm, (i), axis=0)
13             removeu = True
14             break
15 scaler = MinMaxScaler()
16 atributos = scaler.fit_transform(atributos)
```

Fonte: elaborado pelo autor.

APÊNDICE D – CÓDIGOS FONTE DOS ÍNDICES INTERNOS DE VALIDAÇÃO

O Quadro 7 mostra as principais ações do código-fonte referente aos índices internos de validação. Na linha 1 é definido o intervalo de possíveis parâmetros para K, sendo escolhido o intervalo de 3 a 10 *clusters*. Na linha 5 são geradas 8 soluções K-means, partindo de 3 a 10 *clusters*, as soluções são então guardadas numa lista (variável `clusters_gerados`). As linhas 10, 14 e 19 respectivamente realizam a validação através dos índices de Silhouette, Calinski-Harabasz e Davies-Bouldin, para cada solução da lista de `clusters_gerados`. Os algoritmos de validação são chamados através do pacote `metrics` da biblioteca `sklearn`. Por fim, os valores de desempenho de cada iteração sobre os *clusters* gerados são guardados nas variáveis `scoreS`, `scoreC` e `scoreD`, para ser posteriormente exibidos.

Quadro 7 – Códigos-fonte dos índices internos

```
 1  range_n_clusters = [3, 4, 5, 6, 7, 8, 9, 10]
 2  clusters_gerados = []
 3  scoreS,scoreC,scoreD = [],[],[]
 4
 5  for n_clusters in range_n_clusters:
 6      clusterer = KMeans(n_clusters=n_clusters, random_state=0)
 7      cluster_labels = clusterer.fit_predict(atributos)
 8      clusters_gerados.append(list((cluster_labels, n_clusters)))
 9
10  for i in range(len(range_n_clusters)):
11      silhouette_avg = silhouette_score(atributos, clusters_gerados[i][0])
12      scoreS.append(silhouette_avg)
13
14  for i in range(len(range_n_clusters)):
15      calinski_harabasz_avg = metrics.calinski_harabasz_score(atributos,
16  clusters_gerados[i][0])
17      scoreC.append(calinski_harabasz_avg)
18
19  for i in range(len(range_n_clusters)):
20      davies_bouldin_avg = metrics.davies_bouldin_score(atributos,
21  clusters_gerados[i][0])
22      scoreD.append(davies_bouldin_avg)
```

Fonte: elaborado pelo autor.

APÊNDICE E – CÓDIGO-FONTE DO ÍNDICE DE ESTABILIDADE GLOBAL

O Quadro 8 apresenta a implementação em código fonte do algoritmo de estabilidade global. Na linha 1, a variável `n_clusters` denota as quantidades de *clusters* que serão avaliadas. A linha 5 remete à geração de amostras com dados selecionados aleatoriamente do conjunto original, podendo obter dados repetidos, sendo geradas 50 amostras. A linha 11 denota o laço de repetição referente à duplicação de cada amostra (linhas 12 e 13), clusterização do par (linhas 16 e 19), e comparação das soluções através do índice ARI (linha 21), cujo método `adjusted_rand_score` foi importado da biblioteca `Sklearn`. Os valores retornados pelos índices são adicionados à lista de índices para a solução com K-clusters determinada, e após repetidos os passos para as outras 49 amostras, os valores são guardados na variável `índices_rand` para posterior exibição. Ressalta-se que a clusterização foi feita utilizando sementes (números aleatórios de inicialização do algoritmo) diferentes para o algoritmo K-means, caso contrário, o valor de ARI para qualquer par de clusterizações seria 1 (similaridade total).

Quadro 8 – Código-fonte do algoritmo estabilidade global

```
1  n_clusters = [3, 4, 5, 6, 7, 8, 9, 10]
2  indices_rand = []
3  n_samples=len(atributos)
4  samples = []
5  for _ in range(50):
6      samples.append(atributos[np.random.randint(atributos.shape[0], size=n_samples),
7  :])
8
9  for i in n_clusters:
10     rand_k = []
11     for j in range(50):
12         par1 = samples[j]
13         par2 = samples[j]
14
15         clusterer = KMeans(n_clusters=i, random_state=random.randint(0,4294967295))
16         cluster_labels_par1 = clusterer.fit_predict(par1)
17
18         clusterer = KMeans(n_clusters=i, random_state=random.randint(0,4294967295))
19         cluster_labels_par2 = clusterer.fit_predict(par2)
20
21         rand_k.append(adjusted_rand_score(cluster_labels_par1,cluster_labels_par2))
22     indices_rand.append(rand_k)
```

Fonte: elaborado pelo autor.

APÊNDICE F – CÓDIGO-FONTE DO ÍNDICE DE ESTABILIDADE ENTRE SOLUÇÕES

O Quadro 9 apresenta a implementação em código-fonte do algoritmo de estabilidade entre soluções. O método recebe como parâmetros o número de *clusters* na solução e a solução original. A linha 8 inicia o laço de repetição para a comparação das 100 amostras. A linha 13 realiza a clusterização da amostra, é então realizada a intersecção entre os dados do conjunto de dados original e os dados da amostra resultando na variável `originais_I_amostra`. A linha 16 realiza a intersecção dos dados do *cluster* original e os dados comuns resultando na variável `cluster_original_I_originais_I_amostra`. No laço da linha 19 é listado os valores de Jaccard para os dados do *cluster* original (somente com dados pertencentes ao conjunto de dados da amostra) e os dados do *cluster* da amostra, adicionados na variável `lista_jac`. Com a lista populada, é descoberto o *cluster* da amostra que mais se assemelha ao *cluster* original (linha 28), os valores das 100 amostras são então guardados na variável `lista_jac_max` para futura exposição.

Quadro 9 – Código-fonte do algoritmo estabilidade entre soluções

```
1  def estabilidade_clusters(n_clusters,cluster):
2      original = cluster.labels_
3      lista_jac_max = []
4      for i in range(n_clusters):
5          lista_jac_k = []
6          ids = [id_ for id_, x in enumerate(original) if x == i]
7          cluster_original_id = [id_atributos[id_] for id_ in ids]
8          for j in range(100):
9              amostra = amostras[j]
10             amostra_features = [list(x[1:])[0] for x in amostra]
11             clusterer = KMeans(n_clusters=n_clusters,
12 random_state=random.randint(0,4294967295))
13             clustering_amostra = clusterer.fit_predict(amostra_features)
14             originais_I_amostra = [value for value in id_atributos if value[0] in
15 [x[0] for x in amostra]]
16             cluster_original_I_originais_I_amostra = [value for value in
17 cluster_original_id if value[0] in [x[0] for x in originais_I_amostra]]
18             lista_jac = []
19             for k in range(n_clusters):
20                 ids = [id_ for id_, x in enumerate(clustering_amostra) if x == k]
21                 cluster_amostra_id = [amostra[id_] for id_ in ids]
22                 cluster_amostra_I_originais_I_amostra = [value for value in
23 originais_I_amostra if value[0] in
24                 [x[0] for x in cluster_amostra_id]]
25                 lista_jac.append(jaccard(
26                 cluster_original_I_originais_I_amostra,
27                 cluster_amostra_I_originais_I_amostra))
28             lista_jac_k.append(max(lista_jac))
29         lista_jac_max.append(lista_jac_k)
30     return lista_jac_max
```

Fonte: elaborado pelo autor.

APÊNDICE G – ALGORITMO DE RELABELING E CÓDIGO FONTE

Dolnicar e Leisch (2017) propõem o algoritmo presente no Quadro 10, que permite rastrear segmentos ao longo de partições (soluções) com diferentes números de *clusters*. Considerando P_1, P_2. ... , P_m como uma série de m partições (soluções) com números de *clusters* $K_1 < K_2 < ... < K_m$, os seguintes passos aplicam o conceito de *relabeling* de maneira que o *cluster* K_1 da partição P_1 é o mesmo *cluster* que o K_1 da partição P_2 e assim por diante.

Quadro 10 – Passos do Algoritmo de *relabel*

1. Clusterizar dados com qualquer algoritmo de clusterização para obter a partição P_1, atribua 1 para *i*.
2. Calcular a tabela de contingência *T* de K_i x K_{i+1} para as partições P_i e P_{i+1}. Nomeie as colunas de *T* para 1, . . . , K_{i+1}.
3. Encontrar os K_i *clusters* em P_{i+1} que possuem concordância máxima com os *clusters* de P_i resolvendo um problema de atribuição de soma linear (LSAP) na tabela *T*. Ou seja, utilizar LSAP para permutar as colunas de *T* de maneira que a diagonal principal da matriz das primeiras K_i colunas seja máxima. Permutar as colunas junto com seus nomes.
4. Inserir o restante K_{i+1} - K_i à direita da coluna que mais se encaixa (maior número de dados em comum) em P_i. Se o elemento da linha *j* é o máximo, inserir entre as colunas *j* e *j+1*.
5. Nomeie os *clusters* de P_{i+1} de acordo com as colunas permutadas da tabela *T*.
6. Repetir até $i = m-1$.

Fonte: Dolnicar e Leisch (2017).

O Quadro 11 apresenta a implementação em código fonte do algoritmo de *relabel* apresentado no Quadro 10. Nas linhas 1 a 3 são criadas as variáveis utilizadas no algoritmo, sendo `range_clusters` a quantidade limite do laço de *clusters* que será feita a análise. Os *clusters* são guardados na variável `lista_particoes`, e os *labels* já alinhados são guardados na variável `lista_labels`. A linha 4 remete ao passo 1 do algoritmo original, onde é feita a partição em si dos dados através de K-means. A linhas 10 e 11 remetem à tabela de contingência do passo 2 e ao problema LSAP do passo 3, funções importadas das bibliotecas `Pandas` e `Scipy`, respectivamente. Então, os *labels* originais (variável `label_original`) são transformados nos *labels* adequados da solução de k+1 *clusters* (variável `label_relabel`) através da manipulação de colunas da tabela de contingência.

Quadro 11 – Código-fonte do algoritmo de r*elabel*

```
1  range_clusters = 10
2  lista_particoes = []
3  lista_labels = []
4  for i in range(2,range_clusters+2):
5      lista_particoes.append(KMeans(n_clusters=i, random_state=0).fit(atributos))
6  cluster_i = np.array(lista_particoes[0].labels_)
7
8  for i in range(2,range_clusters+1):
9      cluster_i_1 = np.array(lista_particoes[i-1].labels_)
10     contingencia = crosstab(cluster_i, cluster_i_1)
11     label_original,label_relabel =
12 linear_sum_assignment(contingencia,maximize=True)
13     for j in range(len(label_original)):
14         if(label_original[j] != label_relabel[j]):
15             for k, item in enumerate(cluster_i_1):
16                 if(item == label_relabel[j]):
17                     cluster_i_1[k] = label_original[j]
18                 else:
19                     if(item == label_original[j]):
20                         cluster_i_1[k] = label_original[j]+100
21                         mudou = True
22             if mudou:
23                 label_relabel = [label_original[j]+100 if x == label_original[j]
24 else x for x in label_relabel ]
25     for m,item in enumerate(cluster_i_1):
26         if(item not in label_original):
27             cluster_i_1[m]=i
28     lista_labels.append(cluster_i)
29     if(i!=range_clusters):
30         cluster_i = cluster_i_1
```

Fonte: elaborado pelo autor.

APÊNDICE H – CÓDIGO FONTE DO ALGORITMO SLSA

O Quadro 12 apresenta o código fonte do algoritmo SLSa. O método recebe como parâmetro a lista de *labels* das soluções com diferentes números de *clusters*. O algoritmo sempre leva em conta duas variáveis, `item`, que representa a primeira solução com *k clusters*, e `próximo`, que representa a solução *k+1 clusters*. Para cada *cluster* na solução contida em `próximo`, é calculado a distribuição do pertencimento a partir dos *clusters* da solução anterior (`item`). Com a distribuição dos percentuais de pertencimento guardada na variável `ps`, é calculado a medida de entropia para cada *cluster* da solução contida em `próximo`, guardada na variável `entropia`. A estabilidade é calculada então a partir da medida de entropia, e é guardada em uma lista de estabilidades por *cluster* (variável `estabilidade_cl_i`), que por sua vez é guardada em uma lista de estabilidades por solução (variável `estabilidades`).

Quadro 12 – Código-fonte do algoritmo SLSa

```
1  def slsa(lista_labels):
2      estabilidades = []
3      for i,item in enumerate(lista_labels):
4          if(i < len(lista_labels)-1):
5              estabilidade_cl_i = []
6              proximo = lista_labels[i+1]
7              label_k = [[x,y] for x,y in enumerate(item)]
8              label_k_1 = [[x,y] for x,y in enumerate(proximo)]
9
10             for j in range(len(set(proximo))):
11                 ps = []
12                 total = np.count_nonzero(proximo == j)
13                 for k in range(len(set(item))):
14                     p = len([x for x in [x[0] for x in label_k if x[1] == k] if x
15 in [x[0] for x in label_k_1 if x[1] == j]])/total
16                     ps.append(p)
17                 entropia = 0
18                 for pj in ps:
19                     if pj != 0:
20                         entropia += pj*log(pj,2)
21                 entropia = (entropia*(-1))/log(len(set(item)),2)
22
23                 estabilidade_cl_i.append([1 - entropia, ps])
24             estabilidades.append(estabilidade_cl_i)
25     return estabilidades
```

Fonte: elaborado pelo autor.