UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA
DCET - DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LAVERTY DIDERONE DE ASSIS LADEIA

AVALIAÇÃO DA EFICÁCIA DOS ALGORITMOS DE CLUSTERING PARA SEGMENTAÇÃO DE CLIENTES DE UMA INDÚSTRIA DE PROCESSAMENTO DE PROTEÍNA ANIMAL COM BASE EM SEUS HISTÓRICOS DE COMPRA

VITÓRIA DA CONQUISTA – BA
JUNHO - 2023

LAVERTY DIDERONE DE ASSIS LADEIA

AVALIAÇÃO DA EFICÁCIA DOS ALGORITMOS DE *CLUSTERING* PARA SEGMENTAÇÃO DE CLIENTES DE UMA INDÚSTRIA DE PROCESSAMENTO DE PROTEÍNA ANIMAL COM BASE EM SEUS HISTÓRICOS DE COMPRA

Projeto entregue à disciplina Trabalho Supervisionado II como requisito parcial para obtenção do Grau de Bacharel em Ciência da Computação pela Universidade Estadual do Sudoeste da Bahia.

Orientadora: Prof.ª Dra Maísa Soares dos Santos Lopes

VITÓRIA DA CONQUISTA – BA

JUNHO – 2023

LAVERTY DIDERONE DE ASSIS LADEIA

AVALIAÇÃO DA EFICÁCIA DOS ALGORITMOS DE *CLUSTERING* PARA SEGMENTAÇÃO DE CLIENTES DE UMA INDÚSTRIA DE PROCESSAMENTO DE PROTEÍNA ANIMAL COM BASE EM SEUS HISTÓRICOS DE COMPRA

Aprovada em 21/06/2023

BANCA EXAMINADORA

Profª Dra Maísa Soares dos Santos Lotes
Universidade Estadual do Sudoeste da Bahia – UESB
Orientadora

Profª Dra Alzira Ferreira Silva
Universidade Estadual do Sudoeste da Bahia – UESB
Convidada

Prof Dr Gidevaldo Novais dos Santos
Universidade Estadual do Sudoeste da Bahia – UESB
Convidado

Dedico este trabalho em homenagem a meu pai Antônio Dadier Ladeia (*in memoriam*) e a minha mãe Eleide Gomes de Assis Ladeia (*in memoriam*) por me ensinarem o preceito base da educação e da moral, que me fizeram chegar até aqui.

AGRADECIMENTOS

São tantos os agradecimentos que até me preocupo em ser injusto e esquecer de pessoas tão amadas. Portanto, inicio meus agradecimentos à Deus, pela vida, por minha alma, por me fazer acordar todas as manhãs e me fazer apto a viver a vida. Sem Ti, Pai, nada sou.

Agradeço à minha família, primeiramente aos que já não estão mais comigo, digo fisicamente, mas que em pensamento jamais estiveram longe. Agradeço ao meu pai, à minha mãe e ao meu avô João Leôncio. Vocês foram, e são, fundamentais em minha vida. Agradeço aos que estão aqui comigo, ao meu lado, perto ou longe, em todos os momentos, vocês são o meu lastro, a minha base, a minha motivação, a minha inspiração, vocês são os meus amores. Agradeço, então, aos meus filhos Caio e Dan, à minha esposa Emarry, aos meus irmãos Lerley e Leliany, à minha tia Dete, aos meus primos, à minha sogra Dona Nalva. Cada um de vocês regou essa semente que agora floresceu e tenho certeza que dará muitos frutos.

Agradeço aos meus amigos, àqueles que estão em minha vida desde a minha infância, aos meus amigos da universidade que em muitas situações dividiu comigo o esforço dessa graduação. Agradeço a Celina, o meu anjo “uesbiano”. Agradeço aos meu amigos e irmãos do Maria de Nazaré pelo apoio, acolhimento e incentivo.

Por fim, agradeço aos meus mestres, que tenho a honra e a felicidade de também chamá-los de amigos. Agradeço a Profª. Dra. Maísa, a Profª. Dra. Alzira, a Profª. Dra. Kátia, ao Prof. Dr. Hélio, ao Prof. Dr. Roque, ao Prof. Msc. Stênio, ao Prof. Dr. Gidevaldo (Gil). Levo comigo os seus ensinamentos, as conversas, os sonhos e os projetos. Levo comigo o que há de mais precioso para um indivíduo o conhecimento e as experiências.

Amo todos vocês, obrigado!

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

O *clustering* é uma técnica de aprendizagem não supervisionada que particiona o conjunto de dados de entrada em grupos com base na similaridade das características de suas amostras (BAKSHI; BAKSHI, 2018). Essa técnica é bastante aplicada em ambientes corporativos, onde se tem um grande volume de dados e deseja extrair conhecimento a partir dele, utilizado durante o processo de tomada de decisão ou durante o planejamento estratégico da organização (ENHOLM, 2022). Neste contexto, este trabalho se propôs a realizar uma pesquisa, de carácter exploratório, com objetivo de avaliar a eficácia de três algoritmos de *clustering*: o *K-Means*, o *Hierarchical Clustering* e DBSCAN, utilizando como caso de uso o histórico de compras dos clientes de uma indústria de processamento de proteína animal localizada em Vitória da Conquista – Bahia. Para a avaliação dos algoritmos foram consideradas duas abordagens: (i) a primeira considerou as métricas de valores absolutos como, o coeficiente de silhueta e o número de clusters; (ii) a segunda, de carácter mais intuitivo, considerou os princípios de homogeneidade e heterogeneidades dos clusters criados. Mas, antes que os algoritmos fossem avaliados, foi necessário que se cumprisse todas as etapas envolvidas no processo de criação dos modelos de *machine learning*, para isso, foi utilizada a metodologia de desenvolvimento *Cross-Industry Standard Process for Data Mining* (CRISP-DM). De acordo com os critérios de avaliação definidos, concluiu-se que, para o conjunto de dados utilizados, o algoritmo *Hierarchical Clustering* foi o mais eficaz.

Palavras-chave: inteligência artificial; *machine learning; clustering; K-Mean; Hierarchical Clustering;* DBSCAN.

ABSTRACT

Clustering is an unsupervised learning technique that partitions the input dataset into groups based on the similarity of the characteristics of their samples (BAKSHI; BAKSHI, 2018). This technique is widely applied in corporate environments, where you have a large volume of data and want to extract knowledge from it, used during the decision-making process or during the organization's strategic planning (ENHOLM, 2022). In this context, this work proposed to carry out an exploratory research, with the objective of evaluating the effectiveness of three clustering algorithms: K-Means, Hierarchical Clustering and DBSCAN, using as a use case the customers' purchase history of an animal protein processing industry located in Vitória da Conquista - Bahia. For the evaluation of the algorithms, two approaches were considered: (i) the first considered metrics of absolute values such as the silhouette coefficient and the number of clusters; (ii) the second, more intuitive, considered the principles of homogeneity and heterogeneity of the clusters created. But, before the algorithms were evaluated, it was necessary to fulfill all the steps involved in the process of creating machine learning models, for this, the Cross-Industry Standard Process for Data Mining (CRISP-DM) development methodology was used. . According to the defined evaluation criteria, it was concluded that, for the dataset used, the Hierarchical Clustering algorithm was the most effective.

Keywords: artificial intelligence; *machine learning; clustering; K-Mean; Hierarchical Clustering;* DBSCAN.

SUMÁRIO

1 INTRODUÇÃO

Segundo Ludermir (2021), atualmente o mundo vive uma nova revolução industrial, caracterizada pela integração de tecnologias emergentes que alicerçam modelos de negócio disruptivos. Tecnologias como a inteligência artificial (IA), a robótica e a internet das coisas (IoT) têm sido adotadas por empresas e indústrias impulsionando processos de mudança, permitindo que a produção se torne mais eficiente, flexível e personalizada, fazendo com que essas empresas se tornem mais competitiva, impactando diretamente na forma de como as pessoas consomem seus produtos e serviços (LUDERMIR, 2021).

Das tecnologias de ponta, a inteligência artificial gradativamente tem ganhado mais espaço no setor produtivo, muito por conta da sua versatilidade. Ludermir (2021) aponta três motivos que contribuíram para a expansão da IA : (i) a capacidade que a inteligência artificial tem em resolver problemas complexos, característica que a abordagem algorítmica tradicional não se mostra tão eficiente; (ii) redução significativa dos custos de hardware e o aumento do poder computacional o que permitiu o investimento em pesquisa e adoção por parte das empresas; (iii) a imensa disponibilidade de dados produzidos pela internet, pelos sensores e atuadores, presentes no equipamentos utilizados pelas indústrias e outras empresas, pelos softwares de gestão que integram as diferentes áreas das empresas e pelas mídias sociais.

De acordo com a pesquisa realizada pela Deloitte em 2018, que contou com a participação de 1.900 empresas de sete países (Austrália, Canadá, China, Alemanha, França, Reino Unido e Estados Unidos), foi apontado que 75% dos gestores entrevistados consideram que as tecnologias relacionadas com a inteligência artificial são muito ou criticamente importantes para o sucesso de seus negócios, isso representa um aumento de 81% se comparado com 2016 (LOUCKS, 2019).

Ainda em relação ao quão amplo é o espectro de aplicação da inteligência artificial, Enholm (2022) cita duas perspectivas que ajudam a compreender melhor porque a IA tem sido adotada pelo mercado. A primeira é em relação a aplicação da inteligência artificial na automatização de tarefas e processos. A segunda trata-se de uma perspectiva assistencial, na qual a IA se integra à experiência humana apoiando e otimizando decisões e ações.

Em se tratando da perspectiva assistencial da IA, atualmente as técnicas de aprendizado de máquinas podem ser aplicadas a grandes volumes de dados gerados pelas empresas, sendo possível extrair informações e conhecimento que sirva de subsídio no processo de tomada de decisão que, por sua vez, pode gerar valor em vários setores ou segmentos. Um exemplo claro que pode representar muito bem este cenário é a segmentação dos clientes de uma empresa a partir dos dados gerados por seus sistemas de gestão. Ao longo de anos, esses *softwares* registram seus bancos de dados milhões de transações armazenadas em dezenas, ou até mesmo centenas, de tabelas. Antes da evolução da IA esses milhares de dados eram esquecidos, ou quando lembrados, eram utilizados em relatórios engessados com informações limitadas.

Nesse contexto, este trabalho se propõe em realizar um estudo de caso para aplicação de técnicas de *machine learning* para criar *clusters* de clientes que pertencem a uma indústria de processamento de proteína animal em Vitória da Conquista – BA baseando-se em seus perfis de compra. Através desse estudo de caso é possível demonstrar como a inteligência artificial pode se conectar às estratégias de negócio de uma empresa, extraindo *insights* e padrões ocultos a partir dos dados por elas gerados, auxiliando o processo de tomada de decisão.

1.1 Objetivo

De forma geral, esse trabalho tem como objetivo avaliar a eficácia algoritmos de *clustering* (agrupamento) baseados em técnicas de aprendizagem de máquina não supervisionados, utilizando como caso de uso o histórico das compras dos clientes de uma indústria de processamento de proteína animal localizada em Vitória da Conquista - BA.

1.1.1 Objetivos específicos

Como objetivos específicos este trabalho se propõe a:

- Realizar a análise exploratória e preparação do conjunto de dados contendo o histórico das vendas dos clientes da indústria de processamento de proteína animal ao longo dos últimos 3 (três) anos.

- Criar e treinar os modelos de *clustering* com base nos algoritmos de aprendizagem de máquina não supervisionado.
- Validar os resultados obtidos modelos implementados.
- Analisar a eficácia dos modelos com base nos resultados obtidos.

1.2 Justificativa

Em um mundo cada vez mais competitivo, as empresas são compelidas a responderem mais rapidamente às incessantes mudanças apresentadas pelo mercado. Para isso, as organizações precisam ter acesso às informações corretas, no tempo certo e em um formato adequado para que as decisões possam ser tomadas de forma eficiente (BENABDELLAH, BENGHABRIT; BOUHADDOU, 2019). Em um cenário tão dinâmico, as organizações estão recorrendo às tecnologias emergentes como ferramentas estratégicas para alcançar os melhores resultados e obter vantagens competitivas (WEILL e WOERNER, 2017, *apud* BORGES *et al*., 2021).

Nos últimos tempos, a ascensão da inteligência artificial e o seu desenvolvimento em diversas áreas do conhecimento têm atraído a atenção de muitas organizações. O uso da IA tem gerado valor em várias dimensões dos negócios, tais como a automatização de processos, a capacidade de extrair perspectivas valiosas a partir dos dados para auxiliar na tomada de decisões, o engajamento dos clientes e empregados, o *design* e entrega de novos produtos e serviços (BORGES *et al*., 2021).

Pela capacidade de emular o desempenho humano na realização de tarefas que requer cognição, as tecnologias de inteligência artificial têm sido bastante eficazes no apoio à tomada de decisão, sobretudo no que diz respeito ao processamento de grandes volumes dados em um curto período de tempo.

Movendo-se nessa direção, encontra-se a Frigosol. Formado por grupo de três empresas, a Frigosol é uma indústria de processamento de proteína animal localizada no sudoeste baiano que atende uma região formada por quarenta cidades totalizando aproximadamente 1.700 clientes, considerando o mercado interno e externo, no qual ao longo de cinco anos, foram realizadas quase 190 mil vendas. Seus gestores entendem que para atingir melhores resultados, principalmente na área comercial, é imprescindível que se faça uma gestão eficiente de seus clientes. Desse modo, o primeiro passo é, então, identificar quais são os perfis dos seus clientes, tendo como base o histórico de vendas.

Dos dados gerados a partir das vendas é possível extrair características e padrões que permitam associar os clientes que possuam similaridades entre si, formando grupos. Mas como extrair essas informações? Quais técnicas e tecnologias podem ser aplicadas para que seja possível identificar esses padrões? Neste cenário, como essas informações podem agregar valor ao negócio da empresa?

1.3 Considerando as limitações de orçamento por parte das pequenas e médias empresas que dificultam o acesso aos ambientes computacionais de alto desempenho, este trabalho se propõe a resolver o problema descrito acima, criando modelos de *machine learning* utilizando técnicas de aprendizagem de máquina não supervisionadas que permitam o agrupamento dos clientes de acordo com os seus históricos de venda, sem que seja necessária uma infraestrutura computacional de alto poder computacional. Uma vez que os grupos de clientes sejam definidos pelos modelos de IA, a gestão comercial da empresa pode traçar estratégias mais eficientes ou até mesmo gerar novos modelos que otimizem suas vendas.Metodologia

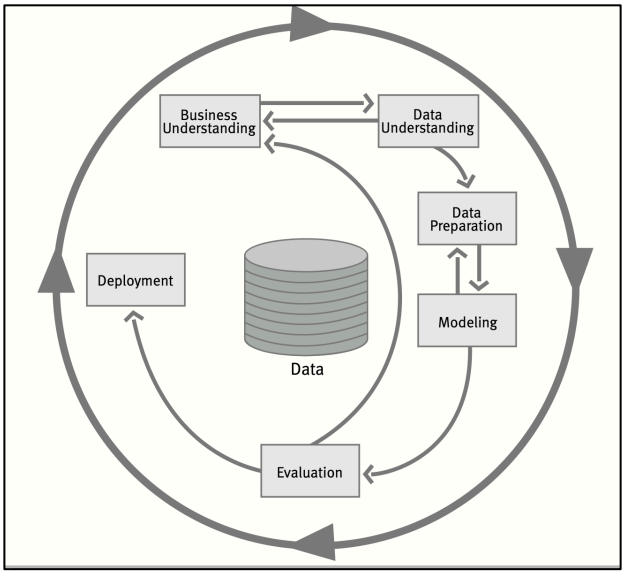
Este trabalho é uma pesquisa aplicada, cujo objetivo é gerar conhecimentos a partir de uma aplicação prática, dirigida à solução de problemas específicos (PRODANOV e FREITAS, 2013). Considerando sua finalidade, este trabalho é uma pesquisa exploratória, pois visa avaliar os algoritmos de *clustering* para segmentar clientes de uma indústria de processamento de proteína animal com base em seus históricos de compra e o seu desenvolvimento seguiu as seguintes etapas:

- Levantamento bibliográfico com a finalidade de aprofundar o conhecimento e o entendimento dos conceitos de inteligência artificial, aprendizagem de máquina e, em específico, as técnicas de aprendizado não supervisionado e os algoritmos de *clustering* .
- Aplicação prática das técnicas de *clustering* em um conjunto de dados extraídos de um ambiente corporativo real.
- Análise dos resultados alcançados a partir das etapas anteriores e apresentação do algoritmo de melhor desempenho considerando o contexto e os tipos de dados utilizados na pesquisa.

Quanto à metodologia de desenvolvimento, foi adotado o *Cross-Industry Standard Process for Data Mining* (CRISP-DM). Concebida em 1996 por um grupo de empresas precursoras na área de mineração de dados e *data warehouse*, o CRISP-DM se propõe a criar um modelo de processo padrão para projetos de mineração de dados neutro em termos de indústria, ferramenta e aplicativo (CHAPMAN *et al*., 2000).

Apoiada por grande parte das indústrias de mineração de dados e financiada pela Comissão Europeia, o CRISP-DM foi publicado nos anos 2000. Esta metodologia define seis fases para o ciclo de vida de um projeto de mineração de dados, conforme mostra a Figura 1 - Fases do modelo de referência CRISP-DM.

Figura 1 - Fases do modelo de referência CRISP-DM



Fonte: Chapman *et al.* (2000)

Com base na Figura 1, Chapman *et al*. (2000) explica a natureza cíclica dos projetos de mineração de dados, enfatizando que a ordem em que as fases são realizadas não é fixa ou sequencial, sendo comum a necessidade de retroceder ou avançar entre elas. As setas representam as dependências entre as fases de modo que o resultado obtido em cada fase influencia a escolha da próxima fase ou tarefa específica.

A seguir cada uma das fases são abordadas brevemente:

- entendimento do negócio: a fase inicial foca na compreensão dos objetivos do projeto e no entendimento dos requisitos sob uma perspectiva do negócio. Esse conhecimento é então convertido para uma definição do problema e, de forma preliminar, é projetado um plano para alcançar os objetivos (CHAPMAN *et al*., 2000). Provost e Fawcett (2016) acrescenta que nesta fase o problema é reformulado repetidas vezes em um processo de descoberta até que se consiga uma formulação de uma solução aceitável;

- entendimento dos dados: Provost e Fawcett (2016) diz que os dados são a matéria-prima disponível a partir da qual a solução de um projeto de mineração de dados será desenvolvida. Desse modo, a fase de entendimento dos dados inicia com a coleta dos dados, identificando suas fontes, estimando os possíveis custo para adquiri-los e segue com as atividades que permitam a familiarização com os dados, identificando possíveis problemas de qualidade, limitações e, até mesmo, a descoberta dos primeiros insights (CHAPMAN *et al.*, 2000). Nesta fase os dados são “escavados” para que seja possível revelar a estrutura do problema (PROVOST; FAWCETT, 2016);
- preparação dos dados: Segundo Chapman *et al*. (2000), a fase de preparação dos dados compreende todas as atividades para definir o conjunto final de dados utilizados como entrada para o treinamento dos modelos (CHAPMAN *et al.*, 2000). Nesta fase os dados são manipulados e convertidos em formatos que rendam melhores resultados (PROVOST; FAWCETT, 2016);
- modelagem: Nesta etapa são selecionadas várias técnicas de modelagem, incluindo aplicação de algoritmos e otimização de seus parâmetros. Para um mesmo problema pode-se aplicar diferentes técnicas de modelagem buscando alcançar os melhores resultados. É bastante comum voltar à fase de preparação dos dados, para que sejam feitos ajustes em seus formatos (CHAPMAN *et al*., 2000);
- avaliação: O objetivo desta fase é fazer um exame minucioso avaliando e estimando os resultados obtidos a fim de garantir que as entregas estão válidas e confiáveis (PROVOST; FAWCETT, 2016). A natureza das avaliações são quantitativas e qualitativas e em resumo esta fase ajuda a garantir que os modelos satisfaçam os objetivos de negócio (PROVOST; FAWCETT, 2016). A pergunta chave para esta fase é: há alguma questão importante do negócio que não foi suficientemente considerada (CHAPMAN *et al*., 2000)?;
- implantação: nesta fase os modelos criados são colocados em uso real de maneira que o conhecimento obtido passa ser organizado e apresentado de uma forma em que os interessados possam usá-lo (CHAPMAN *et al*., 2000). A depender dos objetivos do negócio, a implantação dos modelos pode ser simples como gerar alguns relatórios ou complexa que envolve decisões de negócio em tempo real (PROVOST; FAWCETT, 2016).

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentados os conceitos e técnicas relacionados à inteligência artificial, aprendizagem de máquina e algoritmos de *clustering*, que são temas aderentes ao objeto de pesquisa deste trabalho.

2.1 Inteligência artificial

Nas últimas décadas a inteligência artificial (IA) tem chamado bastante atenção pelos avanços e resultados alcançados (ENHOLM, 2022). Reconhecimento de imagens, sistema de recomendação, carros autônomos e inúmeras outras aplicações modernas são possíveis devido aos avanços da inteligência artificial. Contudo, ao contrário do que se pensa, a IA não é tão recente. Em 1950 a IA já se estabelecia como uma disciplina acadêmica (HEANLNEIN; KAPLAN, 2019). Naquela época, pesquisadores e cientistas já imaginavam construir máquinas que fossem capazes de executar tarefas cognitivas como os seres humanos (BENBYA; PACHIDI; JAVENPAA, 2021). Um bom exemplo disso é o teste de Turing (LUDERMIR, 2022). De forma simples, o teste de Turing consiste em um humano conseguir identificar se ele está conversando com outro humano ou com uma máquina. Caso o humano não consiga fazer essa distinção, é um indicativo de que o sistema é inteligente de modo que o sistema passou no teste (TURING, 1950, *apud* LUDERMIR, 2022).

Contudo, o termo inteligência artificial só foi utilizado oficialmente em 1956 pelos pesquisadores Marvin Minsky e Jonh MacCarthy durante o workshop Dartmouth *Research Project on Artificial Intelligence*.

Entre as décadas de 1960 e 1970 as pesquisas foram intensificadas, abrangendo contribuições de vários outros campos como: biologia, linguística, psicologia, ciências cognitivas, neurociência, matemática, filosofia, engenharia e ciência da computação (BENBYA; PACHIDI; JAVENPAA, 2021). Mas, mesmo com alguns avanços, a maior parte das expectativas criadas pelos principais investidores não foram correspondidas. Segundo Heanlnein e Kaplan (2019), a falta inicial no progresso da IA na época foi a maneira específica na qual os primeiros sistemas tentaram replicar a inteligência humana, assumindo que essa inteligência pudesse ser formalizada e reconstruída como um conjunto de declarações de "*if-else*". Outro ponto

chave que corroborou para que as pesquisas em IA estagnassem foi a falta de capacidade de processamento dos computadores naquele período (BENBYA; PACHIDI; JAVENPAA, 2021). Heanlnein e Kaplan (2019) trazem um exemplo de interrupção de uma pesquisa liderada pelo psicólogo canadense Donald Hebb que tentava replicar o processo de um neurônio humano. A pesquisa de Hebb levaria, mais tarde, ao desenvolvimento de outras relacionadas às redes neurais (HEANLNEIN; KAPLAN, 2019).

Por mais 20 anos as pesquisas ainda apresentaram resultados tímidos diante das expectativas criadas, principalmente àquelas criadas pelo mercado. Mas, na primeira década do ano 2000 as redes neurais voltaram na forma de *Deep Learning*, com destaque para o programa AlphaGo desenvolvido pela Google. O AlphaGo foi um programa criado para jogar o jogo de tabuleiro Go utilizando uma rede neural. A rede criada apresentou bons resultados de performes o que chamou a atenção de outras indústrias no sentido de visualizar novas oportunidades de aplicação (HEANLNEIN; KAPLAN, 2019). Esse avanço é de tal forma importante que a base desse algoritmo é utilizada em várias aplicações hoje em dia.

Mas em termos conceituais, a inteligência artificial possui um conceito formal? Segundo ENHOLM *et al*. (2022), são várias as definições de inteligência artificial, mas todas buscam sempre fazer uma distinção da IA com as tecnologias convencionais. Antes mesmo de apresentar as definições de inteligência artificial, é importante entender os termos “inteligência” e “artificial” separadamente. De acordo com Lichtenthaler (2017, *apud* ENHOLM et al., 2022), inteligência envolve atividade mentais como aprendizado, raciocínio e compreensão. Já o termo artificial corresponde a algo que é feito por humanos em vez de ocorrer naturalmente. Desse modo, Taguimdje (*et al*. 2022, *apud* ENHOLM *et al*., 2022) diz que a inteligência artificial é algo que faz as máquinas simularem a inteligência.

Com claro entendimento dos significados dos termos “inteligência” e “artificial”, Enholm *et al*. (2019) considera que, de forma geral, existem duas principais definições de IA. A primeira define a IA como “uma ferramenta que resolve uma tarefa específica que pode ser impossível ou muito demorada para um ser humano concluir” (DEMLENHER; LAUMER, 2020, *apud* ENHOLM *et al*., 2022). A segunda definição considera a IA como “um sistema que imita a inteligência humana e os processos cognitivos, como interpretar, fazer inferências e aprender” (MIKALEF; GUPTA, 2021, *apud* ENHOLM *et al*., 2022). Como complementação a essas duas definições mais

gerais, existem ainda outras mais específicas de outros estudiosos conforme consta na Tabela 1.

Tabela 1 - Conceito de inteligência artificial por diversos autores

Autores e Data	Definição
KOLBJORNSRUD *et al*. (2017)	Computadores e aplicações que sentem, compreendem, agem e aprendem
AFIOUNI (2019)	É um conceito geral para sistemas de computadores capazes de executar tarefas que usualmente necessita de uma inteligência humana natural, sendo baseado em regras ou não
LEE *et al*. (2019)	Sistemas inteligentes criados para usar dados, análises, e observações para realizar certas tarefas sem a necessidade de ser programadas para isso.
WANG *et al*. (2019)	Conceito amplo que captura o comportamento inteligente de uma máquina
MAKARIUS *et al*. (2020)	A capacidade de um sistema de interpretar corretamente dados externos, aprender com esses dados e usar esses aprendizados para atingir metas e tarefas específicas por meio de adaptação flexível
SCHMIDT *et al*. (2020)	O esforço de imitar capacidades cognitivas e humanas em computadores
DEMLENHNER; LAUMER (2020)	Sistema de computador que tem a habilidade de perceber, aprender, julgar, ou planejar sem ser explicitamente programado para seguir regras predeterminadas ou sequências de ações através de todo o processo
WAMBA-TAGUIMDJE (2020)	Conjunto de teorias e técnicas usadas para criar máquinas capazes de simular inteligência. É um termo geral que envolve o uso de computadores para modelos de comportamento inteligente com o mínimo de intervenção humana
MIKALEF; GUPTA (2021)	A capacidade de um sistema de identificar, interpretar, fazer inferências e aprender com os dados para alcançar objetivos organizacionais e sociais predeterminados

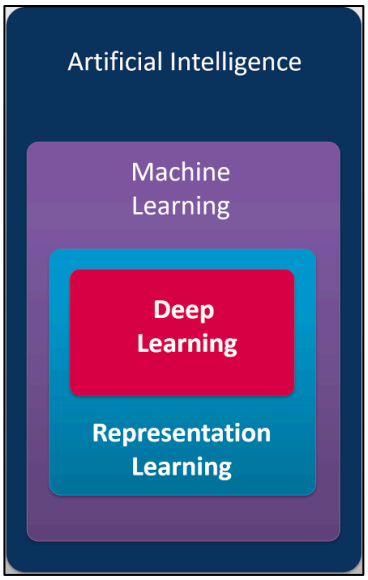
Fonte: ENHOLM *et al*. (2022)

Conforme apresentado na tabela acima, não existe uma única definição para a inteligência artificial, existem várias com abordagens que as diferenciam sutilmente.

São essas nuances que ajudaram a formar os campos, ou disciplinas, da IA como, por exemplo: a *machine Learning*, *deep Learning.* (BORGES *et al*., 2021).

A Figura 2 ilustra melhor o relacionamento entre as disciplinas. Conforme apresenta Borges *et al*. (2021), o *Deep Learning* é um subconjunto do *Machine Learning*. Contudo, o *Deep Learning* não são todas as abordagens de aprendizagem do *Machine Learning*. O *Machine Learning*, por sua vez, é um subconjunto da inteligência artificial.

Figura 2 - Diagrama da representação do relacionamento entre IA, Machine Learning e Deep Learning



Fonte: Borges *et al.* (2021)

2.2 *Machine Learning*

Como mencionado anteriormente, um dos maiores desafios no início do desenvolvimento da inteligência artificial era a capacidade em executar tarefas que são facilmente resolvidas por humanos (ENHOLM *et al*., 2022), como a capacidade de extrair padrões dos dados e adquirir conhecimento a partir deles (ABRAMSON *et al*., 1963, *apud* ENHOLM *et al*., 2022). Tradicionalmente, os *insights* eram obtidos do conjunto de dados desenvolvendo regras de decisões manualmente (BAKSHI; BAKSHI, 2018). Para algumas situações essa estratégia se demonstrava viável, principalmente quando os humanos tinham um bom entendimento do processo e do modelo. Contudo, essa mesma estratégia apresentava desvantagens como o fato de

utilizar usar regras codificadas à mão. Nesse caso, se uma tarefa vinculada à uma regra sofresse alteração toda a regra deveria ser reescrita, fora o fato de que para projetar essas regras era necessário um especialista humano (BAKSHI; BAKSHI, 2018).

À medida em que os algoritmos de *machine learning* eram melhorados, as aplicações baseadas em computadores adquiriram a habilidade de detectar padrões automaticamente a partir dos dados e agir sem serem explicitamente programadas (MURPHY 2012, *apud* ENHOLM *et al*., 2022). É nesse sentido que Ludermir (2022) salienta quão importante são os dados para o *machine learning*, pois somente a partir de uma grande quantidade de exemplos os algoritmos conseguem gerar conhecimento.

Assim como na inteligência artificial, existem na literatura várias definições para o aprendizado de máquina. Schimidt *et al*. (2020) define *machine learning* como uma abordagem indutiva, na qual as regas de decisão são identificadas a partir dos dados usando métodos estatísticos (*apud* ENHOLM *et al*., 2022). Afiouni (2019) por sua vez, define *machine learning* como sendo o subconjunto da IA capaz de "aprender" com os dados e fazer previsões em que as regras sejam ditadas por humanos.

Quanto aos tipos de supervisão, os algoritmos de aprendizagem de máquina, podem ser classificados em três categorias: supervisionado, não supervisionado e por reforço. Na próxima seção, cada uma dessas categorias são abordadas com mais detalhe.

2.2.1 Aprendizado supervisionado

Segundo Sharma *et al*. (2022), o aprendizado de máquina supervisionado é uma técnica em as máquinas são treinadas em um conjunto de dados rotulados e o resultado desse treinamento é uma saída determinada. Os dados rotulados são dados de entrada que são pré-associados com a saída mais adequada do conjunto de dados (SHARMA *et al*., 2022). Sendo assim, o objetivo do aprendizado supervisionado é encontrar uma função de mapeamento que direciona o algoritmo a encontrar no conjunto de dados o mapeamento mais adequando da variável de entrada (x) com a variável de saída (y) com base no conjunto de dados de treinamento (SHARMA *et al*., 2022). Uma vez que os algoritmos são treinados, os modelos são capazes de fazer a

predição do valor ou da classe dos dados que nunca foram vistos (BAKSHI; BAKSHI, 2018). Portanto, os algoritmos de aprendizagem supervisionada aprendem com os pares de entrada e saídas, mas é necessário o papel de um “supervisor” que orienta a forma de como obtém a saída desejada a partir dos exemplos utilizados durante o processo de treinamento (BAKSHI; BAKSHI, 2018).

Existem dois tipos de aprendizado supervisionado: classificação e regressão. No aprendizado supervisionado por classificação a saída de todas tarefas tem um alvo ou rótulo associado a ela (SHARMA *et al*., 2022). O objetivo é encontrar um valor discreto associado a qualquer classe particular ou específica, avaliando esse valor com base da correção do desempenho do algoritmo em relação ao conjunto de dados do problema (SHARMA *et al*., 2022). A classificação algumas vezes é dividida em classificação binária, referindo-se à classificação de duas classes, e a classificação multiclasses, que corresponde à classificação entre duas ou mais classes (BAKSHI; BAKSHI, 2018). No contexto do aprendizado supervisionado por regressão, o propósito consiste em prever um valor que se aproxime ao máximo do valor de saída atual. Assim, a avaliação é realizada por meio do cálculo do erro do conjunto de dados, conforme mencionado em Sharma et al. (2022).

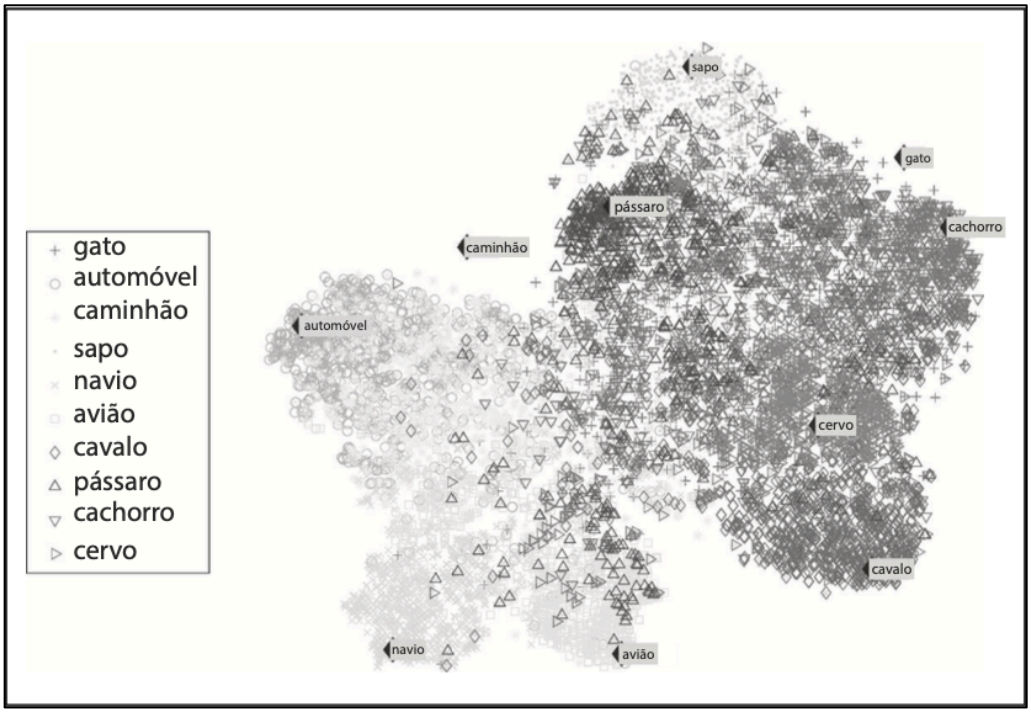
2.2.2 Aprendizado não supervisionado

O aprendizado de máquina não supervisionado é uma técnica em que os dados utilizados no conjunto de treinamento não são rotulados (LUDERMIR, 2022), isso quer dizer que, para um conjunto de dados de entrada (x) não existirá variável de saída correspondente para as entradas informadas (SHARMA, 2022). Nessa abordagem, os algoritmos buscam modelar uma estrutura básica ou de distribuição a partir dos dados de treinamento a fim de descobrir padrões ocultos (SHARMA, 2022). Por meio da similaridade e dos padrões encontrados nos dados, os algoritmos conseguem identificar regras que representam a relação entre os objetos (SHARMA, 2022).

Segundo Géron (2019), são quatro os tipos ou técnicas de aprendizagem não supervisionados: *clustering*, visualização, redução de dimensionalidade, e associação. O *clustering* é uma técnica de aprendizagem não supervisionada que particiona o conjunto de dados de entrada em grupos por meio das relações de similaridade que existem entre os itens (BAKSHI; BAKSHI, 2018). Essa técnica é

bastante aplicada em situações problemas em que se tenha um grande volume de dados e busca extrair padrões de similaridade, correlacionando os dados de forma a criar grupos ou classificações dando significado aos dados (BAKSHI; BAKSHI, 2018). Em relação à técnica de visualização, Bakshi e Bakshi (2018) a explica como sendo uma abordagem de mapeamento de dados complexos apresentando-os em uma melhor forma de visualização, sem que ocorra a transformação desses dados. Desse modo, estes algoritmos tentam preservar ao máximo de estrutura dos dados evitando a sobreposição durante a sua visualização (GÉRON, 2019). O t-SNE é um exemplo de algoritmo não supervisionado do tipo visualização. A Figura 3 ilustra a saída de um algoritmo t-SNE, na qual foi possível visualizar em duas dimensões um conjunto de complexo formado por muitas características. Na Figura 3 é possível identificar vários grupos uns relacionados com animais, outros com meio de transporte. Supondo que esse conjunto de dados tenha diversas características, o algoritmo t-SNE conseguiu apresentar os dados separando bem os grupos. (BAKSHI; BAKSHI, 2018). Quanto à redução de dimensionalidade, trata-se de uma técnica de aprendizagem não supervisionada cujo objetivo é simplificar os dados sem perder muitas informações (GÉRON, 2019). Esta técnica é bastante utilizada para extrair as características mais relevantes de um conjunto de dados que possuem muitas variáveis. Por fim, a associação é uma técnica de aprendizagem não supervisionada baseada em regras (BAKSHI; BAKSHI, 2018) encontradas a partir das relações existentes entre os atributos (GÉRON, 2019). Um bom exemplo da aplicação dessa técnica é encontrar a relação das vendas de algum produto em relação a venda de outros produtos baseados no comportamento e padrões dos clientes.

Figura 3 - Exemplo de uma visualização gerada pelo ao algoritmo t-SNE



Fonte: Géron (2019)

2.2.3 Aprendizado por reforço

Adotando uma abordagem diferente do aprendizado supervisionado e do aprendizado não supervisionado, na qual o aprendizado ocorre a partir dos dados, o aprendizado por reforço é orientado à experiência guiado por um ambiente de recompensa e punição (AFIOUNI, 2023). Segundo Li (2017), o aprendizado por reforço acontece a partir das experiências que surgem através das interações com entidades externas. A ideia que sustenta essa técnica são os objetivos definidos por um agente humano e as recompensas que são baseadas nas melhores estratégias encontradas para alcançar os objetivos propostos. As estratégias, por sua vez, são possíveis combinações de ações e passos, também conhecidas como políticas (AFIOUNI, 2023).

Afiouni (2023) menciona como uma das principais vantagens do aprendizado por reforço o mecanismo de aprendizagem dinâmico, que permite com que o aprendizado nunca pare, sendo perpetuado através de sucessivas interações, em vez de observações dos dados. Embora as interações se apresentem como uma das principais vantagens do aprendizado por reforço, ela também representa um grande desafio, isso pelo fato da busca contínua por melhores estratégias o que demanda um

grande número de interações até que se alcance a estratégia aceitável (AFIOUNI, 2023). As interações crescem cada vez mais conforme as aplicações se tornam mais complexas.

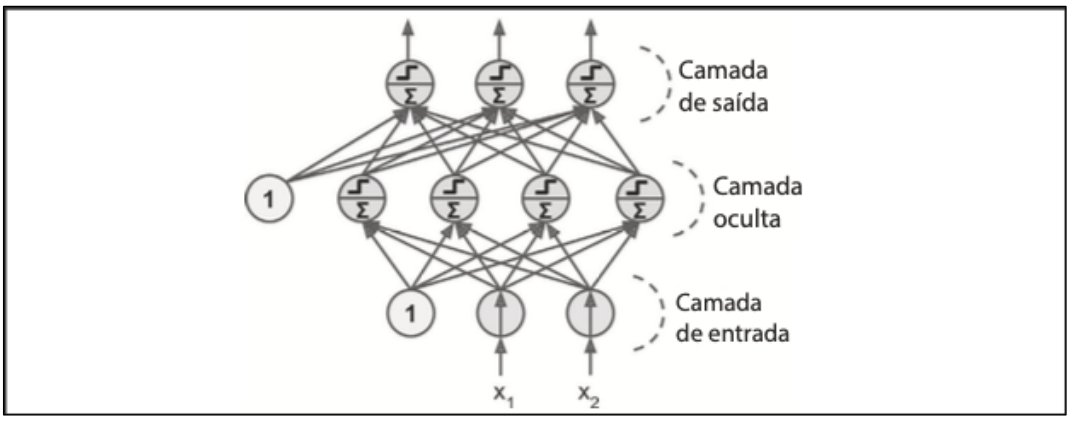
Em um contexto prático, a abordagem de aprendizado por reforço é utilizada por muitos sites como YouTube e Amazon onde os usuários estão continuamente avaliando seus conteúdos e produtos, de modo que, as recomendações refletem as avaliações de seus usuários (AFIOUNI, 2023).

2.3 *Deep Learning*

Como explica Borges *et al*. (2021), o *deep learning* é um subconjunto do *machine learning*, dessa forma, entender o que os diferencia é, de certa forma, uma maneira de obter um melhor entendimento. Basicamente diferença entre o *machine learning* e o *deep learning* está em suas arquiteturas. A arquitetura de *machine learning* tradicional possui basicamente duas camadas, uma de entrada e outra de saída (LI, 2017). O *deep learning*, por sua vez, se caracteriza por uma estrutura multi-camadas (AFIOUNI, 2023), onde, entre as camadas de entrada e saída existem uma ou mais camadas ocultas (LI, 2017). Em resumo, segundo Afiouni (2023), a diferença entre *machine learning* e *deep learning*, é que o segundo faz uso de arquiteturas de redes neurais artificiais.

A Figura 4, apresenta uma estrutura básica da arquitetura de uma rede neural artificial, onde percebe-se claramente as camadas de entrada e saída; e a camada oculta entre essas duas.

Figura 4 - Arquitetura de uma Rede *Neural Multi Layer Perceptron*



Fonte: GÉRON, A. (2019)

Conforme Li (2017) explica, em cada camada, com exceção da camada de entrada, a entrada de cada unidade – antigamente chamada de neurônio - é a soma das unidades das camadas anteriores. Em todas as camadas, exceto na camada de entrada, em cada entrada de cada unidade soma-se os pesos de cada unidade da cama anterior. Li (2017) continua informando que para aplicar a entrada de cada unidade e obter a nova representação da entrada anterior, utiliza-se uma função de transformação não linear, ou uma função de ativação, ou uma função logística. Entre os links de cada unidade de uma camada para outra existem os pesos. Depois dos cálculos que seguem da entrada para a saída, na camada de saída e em cada camada oculta, calcula-se o derivativo regresso, de modo que ocorre uma repropagação dos gradientes em direção à camada de entrada fazendo com que os pesos possam ser atualizados para otimizar alguma função de perda. Essa arquitetura é a arquitetura básica para muitas redes neurais a exemplo a CNN (*convolutional neural network*), a RNN (*recorrente neural network*), a LSTM (*long short term memory network*), etc, sendo que cada uma delas foi desenvolvida para resolver problemas específicos (LI, 2017).

2.4 Técnicas de agrupamento ou *clustering*

O *clustering* é uma técnica cujo objetivo é agrupar um conjunto de dados em *clusters*, de modo que em cada *cluster* sejam agregados aos objetos similares (BENABDELLAH; BENGHABRIT; BOUHADDOU, 2019). O agrupamento dos objetos ocorre por processos repetitivos e iterativos onde a cada iteração busca detectar as

semelhanças e padrões nos atributos presentes no conjunto dos dados (TRIPATHI; BHARDWAJ; ESWARAN, 2018).

Os algoritmos de *clustering* se diferem quanto à abordagem utilizada no processo de agrupamento dos objetos. Desse modo, para um melhor entendimento, estes são classificados em 5 categorias: (i) os algoritmos baseados em particionamento; (ii) algoritmos hierárquicos; (iii) algoritmos baseados em densidade; (iv) algoritmos baseados em grade; (v) e os algoritmos baseados em modelo (TRIPATHI; BHARDWAJ; ESWARAN, 2018).

Tripathi, Bhardwaj e Eswaran (2018) explicam cada uma dessa categorias. Nos algoritmos baseados em particionamento, no início, todos os pontos de dados (objetos) fazem parte de um único *cluster*, assim, por meio de sucessivas iterações os pontos de dados são posicionados entre os *clusters*. Os algoritmos de *clustering* hierárquico criam uma decomposição hierárquica dos dados (USAMA *et al*., 2019), podendo usar duas abordagens: aglomerativa (*bottom-up*), onde cada observação inicia no seu próprio *cluster* e, então, os pares desses segmentos formados são combinados em direção à hierarquia superior; e a abordagem divisiva (*top-down*), na qual todas as observações começam em um *cluster*. A partir deste primeiro *cluster* ocorre repetidas divisões gerando diferentes *clusters*. Tanto na abordagem aglomerativa, quanto na divisiva, o resultado pode ser visualizado como um dendrograma. Quanto aos algoritmos baseados em densidade os *clusters* são definidos como regiões de maior densidade e os objetos são diferenciados como núcleo, ruídos e pontos de fronteira. Os algoritmos de agrupamento baseados em grade dividem os conjuntos de dados em estruturas de grade contendo várias células. Por fim, os algoritmos de agrupamento baseados em modelos agrupam os dados com base em técnicas de modelos estatísticos.

A seguir são apresentados 5 algoritmos de agrupamentos considerados como os mais aderentes ao escopo deste trabalho. Com base nas revisões e análises das pesquisas correlacionadas foram avaliados quatro critérios que ajudaram a definir o nível de aderência dos algoritmos, foram eles: popularidade, flexibilidade, aplicabilidade em conjunto de dados industriais e a capacidade de lidar com dados de alta dimensão (BENABDELLAH; BENGHABRIT; BOUHADDOU, 2018).

2.4.1 *Principal Component Analysis* (PCA)

O PCA, ou análise de componente principal, é um procedimento estatístico que utiliza transformação ortogonal nos dados para converter um conjunto de dados com *n* número de variáveis possivelmente correlacionadas em um conjunto com um número *k* de variáveis, não correlacionadas menor que *n,* chamadas de componentes principais (USAMA, 2019). Os componentes principais são organizados em ordem decrescente de variância, de forma que o primeiro componente atende a maior variância o último a menor (GE *et al*., 2017).

O PCA é amplamente aplicado nas análises exploratória dos dados, sobretudo na redução de dimensionalidades em que os dados de entrada com *n* dimensões são reduzida para *k* dimensões sem perder informações críticas nos dados (USAMA, 2019). Uma vez que a redução de dimensionalidade é aplicada, outras análises podem ser feitas mais facilmente como, por exemplo, a visualização de dados, a detecção de anormalidades e a detecção de *outliers* (GE *et al.*, 2017).

2.4.2 *K-Means*

O *K-Means* é um algoritmo baseado no princípio do particionamento (MEHTA; MEHRA; VERMA, 2021), amplamente usado por sua simplicidade e eficiência (TRIPATHI; BHARDWAJ; ESWARAN, 2018). Seu objetivo é particionar os dados de exemplo em *k clusters*, de modo que cada dado pertença ao *cluster* mais próximo (GE *et al*., 2017). Em cada *k clusters*, há um ponto central chamado de *centroids*. Inicialmente estes *centroids* são colocados ao longo dos pontos dos dados. Cada ponto de dados é associado ao *centroid* com a menor distância. Depois que cada ponto de dados for associado, os *centroids* de novos grupos são recalculados. Os passos descritos são repetidos até que a movimentação dos *centroids* acabe, ou seja, quando a função objetiva está completa e como resultado final são obtidos os *k clusters*. A função objetiva é o erro quadrático, que corresponde à distância entre os pontos de dados e os *centroids* do *cluster* o qual ele está associado (TRIPATHI; BHARDWAJ; ESWARAN, 2018).

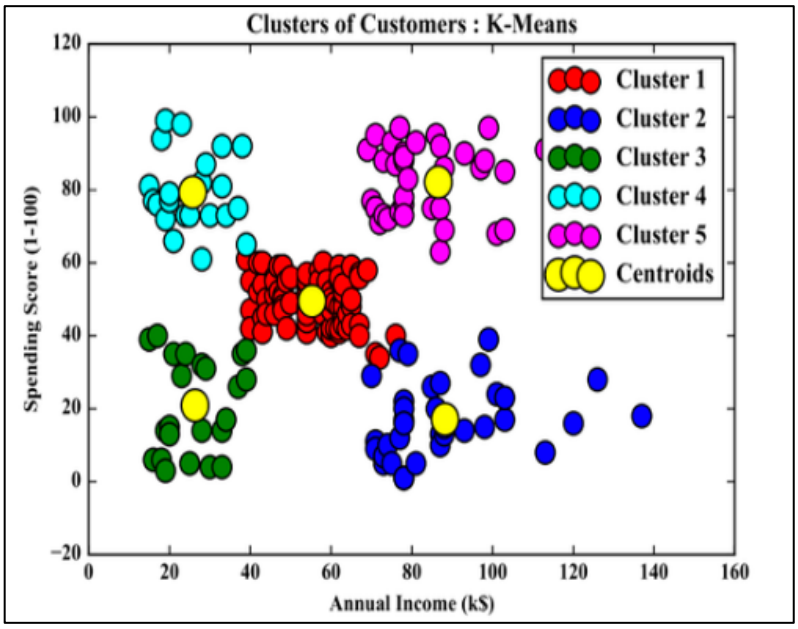
Apesar da simplicidade e eficiência, o *K-Means* apresenta algumas restrições que devem ser consideradas. A primeira delas é a necessidade de informar o número de *clusters* antes que o algoritmo seja inicializado. Essa questão é importante uma vez que não existe uma regra que defina qual o número ideal para *k*, mas existem

técnicas que podem auxiliar a definição desse número. Outro aspecto é que mesmo após o final do processo de definição dos *clusters*, não há garantia de que os *clusters* encontrados estão em uma configuração ótima, mesmo quando a função objetiva for encerrada (TRIPATHI; BHARDWAJ; ESWARAN, 2018).

Em relação às possíveis aplicações, sobretudo no contexto industrial, o *K-Means* tem sido usado em muitos domínios como, por exemplo, em aplicações que identifiquem os diferentes tipos de falha nos processos industriais, na definição de grades de produtos (GE, *et al*., 2017) e, muito alinhada com o objetivo deste trabalho, em aplicações de segmentação de clientes (TRIPATHI; BHARDWAJ; POOVMMAL, 2018).

A Figura 5 apresenta o gráfico *scatter plot* como uma perspectiva visual da saída do *K-Means* na segmentação de clientes considerando a pontuação de gastos em relação às receitas. Conforme é possível observar foram definidos 5 *clusters* e para melhor evidenciá-los foram destacados os *centroids* de cada *cluster*.

Figura 5 - Segmentação de clientes - K-Means



Fonte: (TRIPATHI; BHARDWAJ e POOVMMAL, 2018)

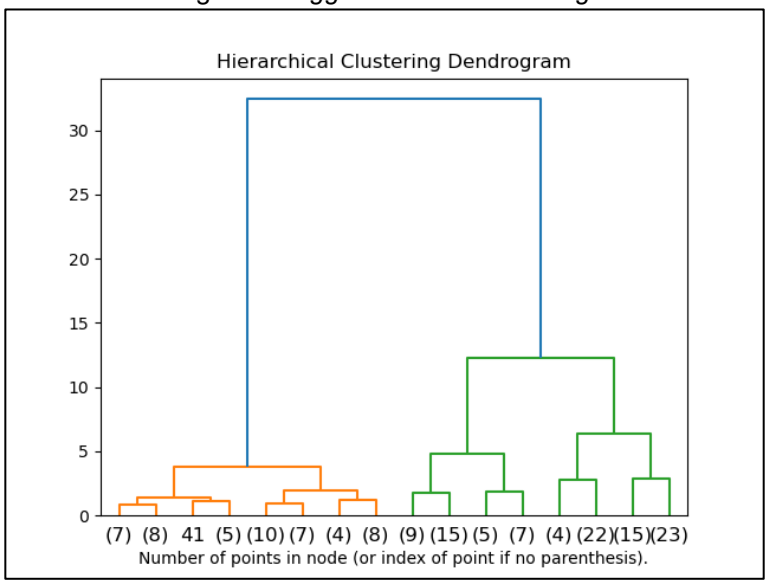
2.4.3 *Agglomerative clustering*

O algoritmo *agglomerative clustering* implementa o método de agrupamento hierárquico em uma abordagem *bottom-up* (USAMA *et al*.,2019). Nesta abordagem, os *N* clusters são formados para diferentes pontos de dados, mesclando os pontos de

dados de acordo com um método de ligação (MEHTA; MEHRA; VERMA, 2021). Conforme explica Abbas (2021), os objetos pertencem inicialmente a uma lista conjuntos individuais S1,...,Sn e por meio de uma função de custo o algoritmo encontra o conjunto de pares {Si, Sj} da lista que é mais barata para ser fundida. Diferente da abordagem de particionamento, para os algoritmos hierárquicos não exigem que seja especificado o número de *k* de clusters em sua inicialização (USAMA *et al*.,2019), capacitando-o manipular qualquer forma de similaridade e distância (ABBAS, 2023).

Conforme consta no manual do *Scikit-Learn* (2007), o dendrograma é das formas utilizadas para representar a estrutura hierárquica criada pelo *agglomerative clustering*. Semelhante a uma estrutura de árvore, o dendrograma é apresentado na Figura 6.

Figura 6 - Representação da hierarquia dos clusters segundo o algoritmo *Agglomerative Clustering*



Fonte: Scikit-Learn (2007)

Outro aspecto relevante do algoritmo de agrupamento aglomerativo são as possíveis estratégias utilizadas para a ligação entre os clusters. Segundo a biblioteca *Scikit-Learn* (2007), são disponibilizadas 4 estratégias de ligação: *ward* que minimiza a soma dos quadrados das distâncias dentro de todos os clusters, a qual minimiza a variância; *average* minimiza a média das distâncias entre todas as observações dos dois pares de *clusters*; *complete* minimiza a distância máxima entre todas as

observações dos dois pares de *clusters*; e *single* minimiza a distância entre as observações mais próximas do pares dos *clusters* (SCIKIT-LEARN, 2007).

Na abordagem aglomerativa, o dendrograma é iniciado considerando cada amostra (ponto de dado) como sendo um *cluster* individualizado. A cada iteração é calculada a proximidade entre os *clusters* com base no critério de ligação selecionado e, então, os pares de *clusters* são fundidos. Esse processo se repete até que reste somente um único *cluster* (DOBILAS, 2021).

2.4.4 DBSCAN

O *Density Based Spatial Clustering of Applications with Noise* (DBSCAN) trata-se de um algoritmo de *clustering* baseado em noções de densidade proposto por Ester *et al*. (1996). Para formalizar as noções intuitivas de *cluster* e ruídos Ester *et al*. (1996) apresentaram os conceitos de densidade de vizinhança baseada em uma função de distância entre dois pontos. Desse modo, a partir de um ponto arbitrário, e com base em um determinado raio, define-se uma quantidade mínima de pontos que deve pertencer a região definida pelo raio informado, garantindo uma densidade desejável para aquela região (ESTER *et al*., 1996). Portanto, para todos os pontos dentro de um *cluster*, a distância entre eles é sempre menor que o raio predefinido (NAMAGANDA-KIYIMBA; MUTALE, 2020). Os pontos com uma densidade acima do limite especificando farão parte de um outro *cluster* (BIRANT; KUT, 2007).

Diferente dos algoritmos baseados na abordagem de particionamento, o DBSCAN não exige que seja informado o *k* de *clusters* em sua inicialização, ao invés disso, o DBSCAN exige que sejam informados outros dois parâmetros: o Eps e o MinPts, que correspondem ao valor máximo do raio para que os pontos possam fazer parte da região que definirá os *clusters* e o número mínimo de pontos que estarão contidos nos *clusters* respectivamente (BIRANT; KUT, 2007). Além disso, o DBSAN tem como habilidade a capacidade de descobrir *clusters* com formatos arbitrários, tal como, linear, côncavo, oval, etc (BIRANT; KUT, 2007). Tais características tem provado a capacidade do DBSCAN de processar grande volumes de dados.

Uma das desvantagens identificadas para o DBSCAN é a capacidade de capturar apenas certos tipos de pontos de ruídos quando existem *clusters* de diferentes densidades. Além disso, o DBSCAN não apresenta bons resultados quando

os *clusters* estão muitos próximos. Ele tem um melhor desempenho quando os *clusters* estão mais distantes uns dos outros (BIRANT; KUT, 2007).

2.4.5 *Guassian Mixture Models*

O algoritmo *Gaussian Mixture Models* (GMM) utiliza o método probabilístico suave de *clustering*, em que os clusters são descritos de acordo com diferentes distribuições Gaussianas (BIARNES, 2020). Esta estratégia faz com que a modelagem dos dados seja mais flexível se comparada a outros algoritmos de agrupamento como o *K-Means* (FOLEY, 2019).

Com base no conjunto de dados o algoritmo GMM gera várias distribuições Gaussianas e para cada distribuição assume-se um *cluster*. Os pontos de dados que apresentem uma mesma distribuição Gaussiana pertencem ao mesmo *cluster* (FOLEY, 2019). Os vetores das médias (μ) e na matriz de covariância (∑) do conjunto de dados servem de base para definir os parâmetros que serão utilizados para gerar as distribuições Gaussianas. O desafio, então, é escolher os parâmetros mais apropriados para gerar as distribuições (BIARNES, 2020).

Para escolher os parâmetros mais adequados para gerar as distribuições Gaussianas, o GMM usa o algoritmo de maximização de expectativa, do inglês *expectation-maximization* (EM). Abstraindo a complexidade matemática do algoritmo, este possui duas etapas: a etapa de estimativa e a etapa de maximização. Na primeira etapa (etapa E) o algoritmo busca estimar as variáveis com base nos pesos, nas médias e na matriz de covariância. Na segunda etapa o algoritmo maximiza as variáveis em relação aos parâmetros. Esse processo continua até que o algoritmo converge, ou seja, a função de perda não muda (FOLEY, 2019).

Todos os algoritmos apresentados anteriormente são aderentes à natureza do problema, ou seja, agrupamento de clientes com base no históricos de suas compras. No entanto, somente com os aspectos teóricos não é suficiente para definir qual deles será o mais indicado. Para essa definição, faz-se necessário criar, testar e treinar os modelos equivalentes e com base nas métricas obtidas será possível definir qual algoritmo é o mais indicado na aplicação do problema. As próximas seções deste trabalho contemplará todas as fases de criação dos modelos e análise dos seus resultados com base nos dados disponibilizados pela Frigosol.

3 DESENVOLVIMENTO

Seguindo a metodologia CRISP-DM, abordada no item 1.3 deste trabalho, este capítulo descreve os processos e atividades realizadas relacionadas às quatro primeiras etapas desta metodologia. Incialmente é apresentado o entendimento do negócio; em seguida são descritas as atividades envolvidas durante a etapa de entendimentos dos dados; a terceira etapa compreende as ações realizadas na etapa de preparação dos dados; e a quarta etapa descreve as atividades realizadas durante a construção dos modelos de *machine learning*.

3.1 Entendimento do negócio

Cleypaul (2021) considera o entendimento do negócio como uma etapa estratégica para os projetos de *machine learning* pois, segundo ele, entender o negócio, seus processos e as regras envolvidas ajuda na descoberta de insights e no levantamento de hipóteses que poderão ser aplicadas nas etapas subsequentes. Chapman *et al*. (2000) complementa dizendo que entender o negócio aprimora a compreensão real do problema que se pretende resolver (CHAPMAN *et al*., 2000).

Cleypaul (2021) recomenda que antes de qualquer atividade seja feita uma pesquisa inicial, em um contexto mais amplo, buscando responder perguntas básicas como: qual é o propósito geral do negócio a ser analisado? Qual área de atuação? Quem são seus clientes e em quais são os seus mercados? Quais são seus produtos comercializados ou produzidos? A partir dessas perguntas tomou-se conhecimento que a empresa se trata de uma indústria de processamento de proteína animal atuando, principalmente, no mercado baiano, com sede em Vitória da Conquista, cujos principais clientes são outras empresas que atuam no ramo varejistas e restaurantes. Os produtos comercializados são todos de origem animal, proveniente do beneficiamento da carne bovina e suína. Estas respostas foram obtidas através das reuniões realizadas com os especialistas da própria empresa e pesquisas realizadas através da *Internet*.

Mesmo em um contexto geral, esta pesquisa ajudou a mapear os problemas mais comuns enfrentados por empresas deste segmento e, portanto, ajudou também a construir um *background* preliminar de conhecimento que, por sua vez, auxiliou na formulação de prováveis perguntas mais específicas abordadas durante a etapa da

análise exploratória dos dados. Durante esta etapa foram realizadas cinco reuniões com o time de especialista da empresa e duas visitas à fábrica, com o propósito de explorar a dinâmica da empresa e identificar informações que, por alguma razão, ficaram desapercebidas. Por vezes, nessas reuniões, foi apresentada uma necessidade recorrente: uma forma de impulsionar as vendas, buscando alcançar melhores resultados. Foi a partir dessa necessidade que surgiu uma pergunta de carácter mais técnico: como as técnicas de *machine learning* poderiam contribuir para atender as necessidades apresentadas?

Da pergunta levantada, surgiu a primeira hipótese: é possível alcançar melhores resultados nas vendas entendendo o comportamento de compras dos seus clientes. O que se propõe é que a partir do histórico de vendas dos clientes, é possível extrair padrões de compras, descobrir correlações entre os dados, viabilizando a criação de um mecanismo automático de agrupamento de clientes por similaridade. Ora, identificando o grupo no qual o cliente faça parte, a equipe de vendas terá acesso a novas informações, mais confiáveis, que permita elaborar as estratégias de vendas mais efetivas. Dessa hipótese, surgiram outras perguntas que incitam a curiosidade sobre os dados como, por exemplo: quais os produtos mais comprados pelos clientes? Existe sazonalidade em suas compras? Existe relação entre os produtos comprados e localização dos seus clientes? Tais perguntas foram apenas as primeiras, servindo de ponto de partida para a análise exploratória realizada nas seções seguintes.

3.2 Entendimento dos dados

3.2.1 Definição da fonte de dados e sua extração

O entendimento dos dados inicia-se com a definição das fontes que serão utilizadas no processo de construção do *dataset*. E em uma das visitas realizadas à empresa foi apresentado o software de gestão (ERP) utilizado por ela. Este *software* é utilizado pelas áreas de produção, estoque, departamento comercial, compreendendo vendas, gestão de produtos e clientes da empresa. O seu banco de dados armazena um histórico de transações dos últimos cinco anos, sendo a principal fonte de dados utilizada pelas ferramentas de geração de relatórios e *dashboards*, que auxiliam os gestores nas tarefas de análise, validação, conferência e acompanhamento dos processos internos da empresa.

Compreendendo o potencial desta ferramenta e objetivando um primeiro contato sobre os dados, foi solicitada e prontamente disponibilizada uma amostra preliminar dos dados contendo informações sobre os produtos vendidos, as quantidades vendidas, data da venda, valor unitário do produto, valor total vendido de cada produto, os grupos e subgrupos nos quais os produtos fazem parte, o cliente para qual o produto foi vendido e o número da nota fiscal referente à venda. Desta análise, foi possível obter informações básicas sobre os dados como: seus tipos, atributos, provável volumetria e se, realmente, de tais dados poderiam ser extraídos padrões, definir correlações e gerar inferências.

Ao final desta primeira análise, concluiu-se que todas as informações necessárias à construção do *dataset* poderiam sim, ser obtidas a partir do sistema de banco de dados do *software* de gestão. Desse modo, foi formalizado o termo de aceso aos dados garantindo as condições de confidencialidade e assegurando os direitos e deveres conforme prever a Lei geral de proteção aos dados. Com o aceso aos dados garantido, o próximo passo foi definir quais seriam as tabelas de onde os dados seriam extraídos. Neste ponto em específico o apoio do time técnico de manutenção do *software* deu o suporte necessário definida a *view* que fornece os dados aos vários relatórios e gráficos utilizados pela equipe de gestão de vendas como sendo a fonte para a extração dos dados.

A extração dos dados foi realizada pela consulta à *view* mencionada, definido um período de três anos, iniciando no dia 01/02/2020 até o dia 31/03/2023. Os dados foram disponibilizados no formato de planilha eletrônica encaminhados pela empresa.

3.2.2 Inspeção geral dos dados

Com o acesso aos dados foi dado início a etapa da análise exploratória. Esta etapa visa aprofundar o entendimento sobre dados através de consultas e apresentação dos resultados por meio de gráficos e tabelas. A partir daí, ocorre o processo de interpretação dos dados como uma espécie de mineração, onde são extraídos os *insights* sobre os dados. Todo esse entendimento adquirido é bastante aproveitado nas próximas etapas. Para executar as atividades mencionadas foram utilizados um conjunto de *softwares*, bibliotecas e *frameworks* que funcionaram como uma verdadeira “caixa de ferramentas” de manipulação dos dados. A linguagem de programação utilizada foi o Python. Esta linguagem foi escolhida pelo fato de já

possuir nativamente a maioria das bibliotecas e *frameworks* comumente usados, abstraindo a complexidade do desenvolvimento das rotinas de limpeza, visualização e transformação dos dados, aumentando a produtividade na execução destas tarefas. O Jupyter Notebook foi a ferramenta escolhida para implementação do fluxo de trabalho, que compreende a execução e a exploração dos dados. Por fim, como ferramentas auxiliares para manipulação e visualização dos dados, foram utilizadas as bibliotecas Numpy[1], Pandas [2], Matplotlib[3] e Seaborn[4].

Com o ambiente de trabalho configurado, iniciaram-se as ações pertinentes à inspeção dos dados. O primeiro passo foi observar a estrutura dos dados e após análise preliminar, constatou-se que o *dataset* era formado por 61 colunas e 116.037 linhas, onde cada linha corresponde a um item vendido de uma das vendas realizadas durante o período definido. Cada linha possuía 61 colunas, referindo-se às informações de cada produto vendido, como nome do produto, preço unitário, quantidade, o cliente para o qual o produto foi vendido, grupo do produto, subgrupo e outras informações conforme consta no dicionário de dados, disponível no anexo I deste trabalho. Das 61 colunas, 26 eram do tipo float64, 23 do tipo *object* (*string*), 10 do tipo int64 (inteiro de 64 bits) e dois do tipo *datetime* (data e hora).

O próximo passo foi verificar os dados faltantes. A Tabela 2 apresenta todas as colunas (*features*) as quais foram identificados dados faltantes e a quantidade de linhas afetadas pela ausência dos dados.

Tabela 2 - Quantidade de linhas faltas por coluna

Colunas com valores faltantes	**Quantidade de linhas com valores faltantes**
NFS_DATA_CANCELAMENTO	116037
NFP_PECAS	849
PPR_TOTAL_PESO	8894
NFS_LTCRR_LOTE	1666
NFS_LTCRR_DATA	1631
GRUPO_CLIENTE	52848
GRUPO_CLIENTE_DESCRICAO	52848
ATIVIDADE_CLIENTE	36

Fonte: autoria própria,(2023).

[1] https://numpy.org/doc/stable/index.html
[2] https://pandas.pydata.org/docs/index.html
[3] https://matplotlib.org/stable/index.html
[4] https://seaborn.pydata.org

Os dados faltantes são tratados de acordo com o nível de relevância de cada coluna em relação ao negócio que está sendo analisado e em relação ao índice proporcional de dados faltantes *dataset*. Por exemplo, a coluna NFS_DATA_CANCELAMENTO possui 100% de dados faltantes, isso significa dizer que esta coluna não possui dados, portanto não faz sentido mantê-la no *dataset*. A coluna TIPO_FRETE possui somente 20% de seus dados como válidos, ou seja, 80% de dados faltantes, o que representa pouca quantidade de dados disponível, consequentemente, também será removida. Outras colunas como GRUPO_CLIENTE e GRUPO_CLIENTE_DESCRICAO apesar de apresentar alto índice de dados faltantes, não foram desconsideradas de imediato pelo fato de fazerem referência a uma possível classificação dos clientes. Estas colunas foram analisas com mais detalhes, a fim de certificar à sua relevância.

A próxima análise diz respeito às colunas que apresentam valores discretos. Conforme apresenta a Tabela 3 foram seis as colunas analisanda e para cada coluna foram demonstrados os valores estatísticos básicos e faixa dos valores mínimo e máximo.

Tabela 3 - Informações estatísticas das colunas de valores discretos mais relevantes

Colunas	**Média**	**Desvio Padrão**	**Mínimo**	**Máximo**
NFP_QTDE_PRODUTO	333.62	2.132,97	-40.000	67.920,00
NFP_TOTAL_PRODUTO	3735,38	22.212,16	-279.370,00	2.557.262,40
NFP_PRECO_UNITARIO	37,09	890,19	-3.190,00	170.000,00
NFS_VALOR_TOTAL_NOTA	14.138,91	52.507,10	-279.370,00	2.557.262,40
PPR_TOTAL_PESO	338,10	2.037,81	0	50.250,00
NFP_TOTAL_PRODUTO_BRUTO	3.739,21	22.241,32	-279.370,00	2.557.263,34

Fonte: autoria própria (2023)

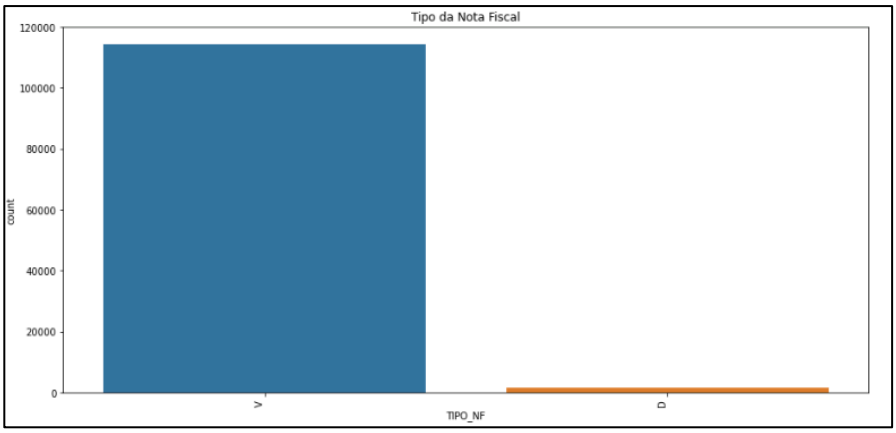
O que chama atenção em relação às informações apresentadas na Tabela 3 foi a presença de valores negativos na faixa de valores mínimos pois, sabe-se que é impossível existem vendas negativas. Outro ponto importante foi a presença valores zerados na coluna PPR_TOTAL_PESO. Finalmente, outro ponto de atenção está relacionado com grande diferença entre os valores máximo e mínimo de cada coluna. Por meio destas observações sugere-se a presença de valores discrepantes (*outliers*). Em relação aos *outliers*, estes serão tratados nas próximas etapas.

Outro ponto analisado durante a inspeção dos dados foi em relação às colunas que, aparentemente, apresentam valores categóricos como, por exemplo, as colunas TIPO, TIPO_NF, GRUPO_CLIENTE_DESCRICAO, GRUPO_PRODUTO, SUBGRUPO_PRODUTO e ATIVIDADE_CLIENTE.

Conforme apresenta a

, a coluna TP_NF possui dois possíveis valores: “V” e “D”, na qual “V ” corresponde às notas fiscais de venda e “D” às notas fiscais de devolução. De início, percebe-se um desbalanceamento entre esses dois valores. Mas quanto ao significado, somente as notas fiscais de venda são relevantes

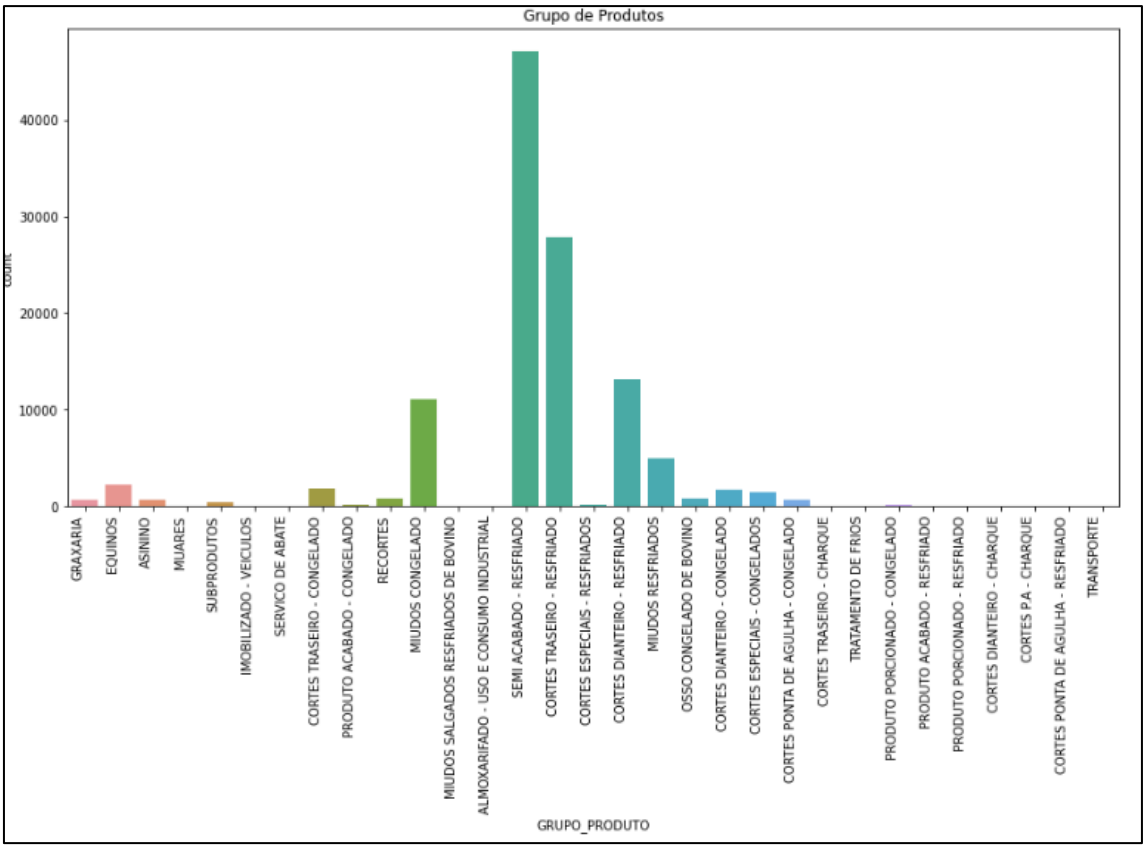
Figura 7 - Distribuição dos dados em relação à coluna TIPO_NF



Fonte: autoria própria, (2023)

A próxima coluna analisada foi a coluna de GRUPO_PRODUTOS. Esta coluna contém a classificação dos produtos, sendo encontrados em 31 grupos conforme ilustra a Figura 8.

Figura 8 - Distribuição dos dados em relação à coluna GRUPO_PRODUTO

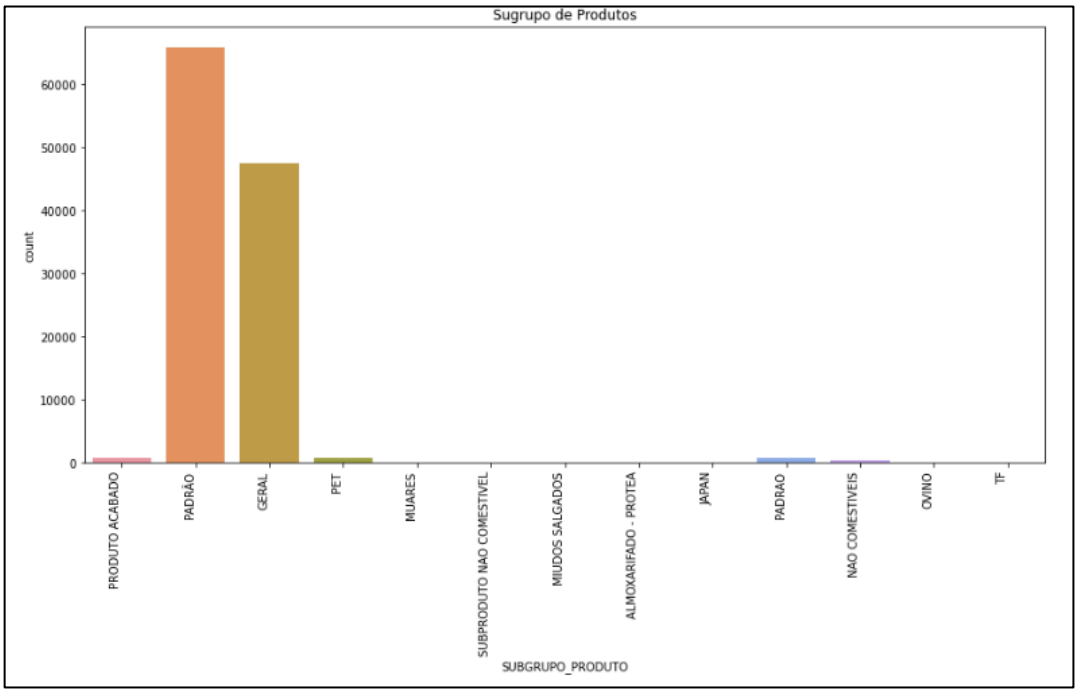


Fonte: autoria própria, (2023)

Conforme observado na Figura 8, os grupo de produtos MIUDOS CONGELADOS, SEMI ACABADO – RESFRIADO, CORTES TRASEIRO – RESFRIADO, CORTES DIANTEIRO – RESFRIADO, MIUDOS RESFRIADOS concentram mais de 90% das vendas. Dando continuidade à análise desta coluna, de imediato alguns grupos de produto chamaram atenção por não serem aderentes ao escopo deste trabalho como, por exemplo, os grupos de produto GRAXARIA, IMOBILIZADO -VEICULOS, ALMOXARIFADO – USO E CONSUMO INDUSTRIAL, TRANSPORTE, EQUINOS e MUARES. Estas colunas são tratadas na etapa de preparação dos dados.

Outra coluna que também se propõe a fazer a classificação dos produtos é a coluna SUBGRUPO_PRODUTO. Nela foram identificados 13 subgrupos. Percebeu-se um grande desbalanceamento dos seus dados, onde 99% dos dados concentram-se nos subgrupos PADRÃO e GERAL. Foi também observado uma duplicidade semântica em relação a dois subgrupos PADRÃO e PADRAO. Esses ruídos serão tratados durante a limpeza dos dados na etapa de preparação dos dados.

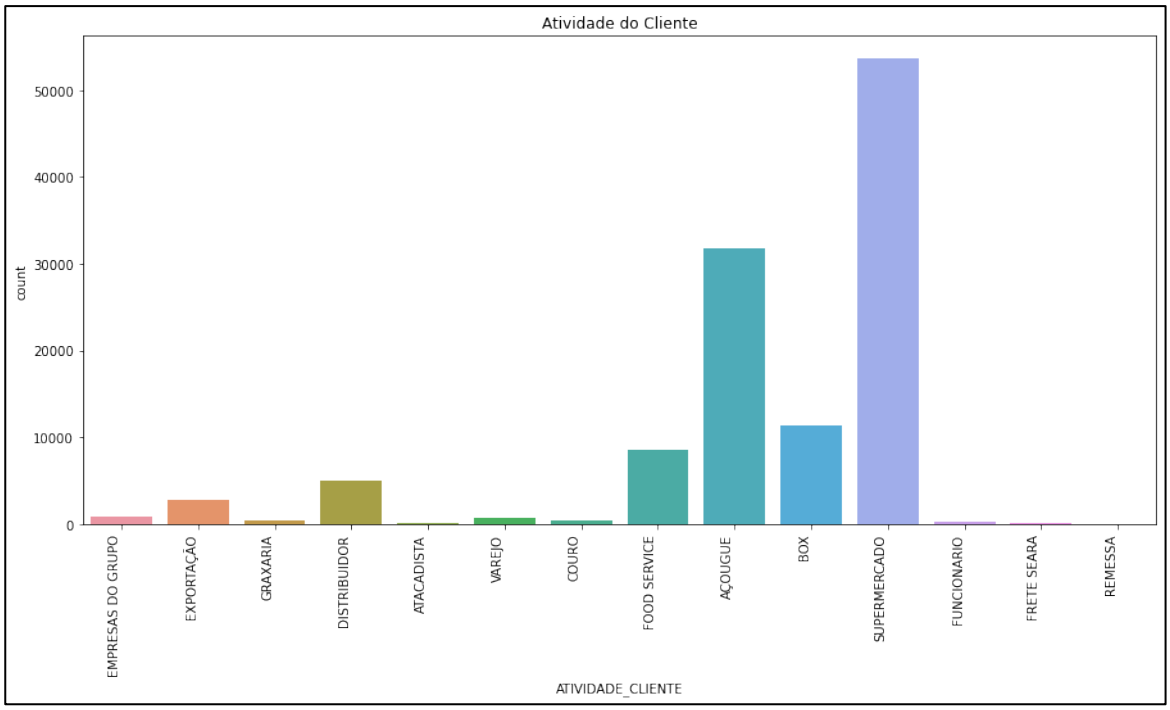
Figura 9 - Distribuição dos dados em relação à coluna SUBGRUPO_PRODUTO



Fonte: autoria própria (2023)

A última coluna que supostamente apresentavam dados categóricos foi a coluna ATIVIDADE_CLIENTE. Esta coluna refere-se a classificação dos clientes quanto às suas atividades econômicas. Conforme apresenta a Figura 10 foram encontradas 14 atividades, onde 4 delas concentram a maioria dos produtos vendidos. Outras atividades não apresentam significância em relação ao objetivo deste trabalho, são elas: EMPRESAS DO GRUPO, EXPORTAÇÃO, GRAXARIA, FUNCIONARIO, FRETE SEARA e REMESSA.

Figura 10 - Distribuição dos dados em relação à coluna ATIVIDADE_CLIENTE



Fonte: autoria própria (2023)

3.3 Preparação dos dados

A qualidade dos dados é um fator preponderante para determinar a eficiência dos modelos de *machine learning* (GE *et al*., 2017). Portanto, nesta etapa são descritas as atividades que foram realizadas relacionadas ao pré-processamento, limpeza e transformação. O resultando obtido foi o *dataset* final que foi utilizado durante etapa de modelagem dos algoritmo de *clustering*.

3.3.1 Definição das colunas relevantes

Geralmente, em projetos em envolvem ciência de dados e *machine learning*, não são todas as *features* disponíveis no *dataset* inicial que serão consideradas relevantes. Neste caso, cada coluna precisa ser analisada, avaliando a sua semântica e o quão aderente esta coluna é em relação aos termos do modelo de negócio em questão (CHAPMAN *et al*., 2000). Nesta situação, o dicionário de dados apresenta-se como importante ferramenta de apoio, auxiliando no processo de seleção destas colunas. A Tabela 4 apresenta a relação das colunas consideradas como não

relevantes em relação ao escopo deste trabalho. Para cada grupo de colunas foram apresentas suas respectivas justificativas de não serem relevantes.

Tabela 4 - Colunas descartadas por não serem relevantes

(continua)

Colunas descartadas	Justificativa
FILIAL_CODIGO FILIAL_FANTASIA, CF_FANTASIA, FILIAL_MARCA	São colunas que fazem referência às colunas NFS_EMPRESA e EMP_NOME e, portanto, trata-se de informações complementares.
NFS_NRO_NF e NFS_SERIE	São colunas que se referem aos identificadores únicos das notas fiscais de vendas e a série da nota fiscal. Os identificados únicos não contribuem para a definição dos padrões de similaridade dos dados, portanto, podem ser descartados.
PRODUTO_COD_IMPORTACAO	Trata-se da coluna que corresponde ao código de importação para o produto. Como não serão considerados produto de importação, esta coluna foi descartada.
NFP_RATEIO_DESC_TOTAL, TL_DESCONTO, NFP_NAT_OP, CFOP_SIMP, NFP_SIT_TRIBUTARIA, NFP_BASE_SUBST_TRIB, NFP_VALOR_ICMS_SUBST_TRIB, NFP_VALOR_COFINS, NFP_BC_COFINS, NFP_BC_ICMS, NFP_VALOR_PIS, NFP_VALOR_ICMS, NFP_BC_PIS, PRODUTO_CLASS_FISCAL	São colunas que apresentaram que fazem menção à taxas ou situações tributárias e valores de impostos. Apesar das informações tributárias ser possível extrair padrões de similaridade dos clientes, para um contexto de relacionado ao perfil de compra dos clientes, não seja útil. Por esse motivo, essas colunas foram desconsideradas.

Tabela 5 - Colunas descartadas por não serem relevantes

(conclusão)

Colunas descartadas	Justificativa
NFP_RATEIO_FRETE, TL_FRETE, TIPO_FRETE, NFS_FRM_MTORISTA, NFS_VEICULO, MOTORISTA, VEICULO, VEND_CODIGO, VEND_NOME.	São colunas que fazem referência aos dados de transporte, frete e identificação dos vendedores que realizaram a venda. Segundo os analistas da empresa, tais informações não contribuem para qualquer tipo de definição de padrão de similaridade entre os clientes.
NFS_LTCRR_LOTE, NFS_LTCRR_DATA	São colunas que fazem referência a número e data do lote do produto vendido. Não existe correlação entre estas informações e padrões de similaridade de clientes, portanto, foram desconsideradas

Fonte: autoria própria, (2023)

Além dos aspecto semântico, o processo de seleção das colunas considera também o índice dos dados faltantes para cada uma delas. Sob essa perspectiva, as colunas NFS_DATA_CANCELAMENTO e TIPO_FRETE apresentaram alto índice de dados faltantes. Como mais de 80% de seus dados eram nulos, ambas as colunas foram eliminadas do *dataset*.

Outras duas outras colunas que também apresentaram alto índice de dados faltantes, precisaram de uma análise mais detalhada para confirmar o seu grau de relevância. Conforme identificado na análise exploratória, suspeitava-se que as colunas GRUPO_CLIENTE e GRUPO_CLIENTE_DESCRICAO estariam relacionadas a algum tipo de classificação dos clientes. Contudo, identificou-se que estas colunas faziam menção ao grupo empresarial a qual o cliente fazia parte. Percebeu-se que estas informações não eram relevantes a serem utilizadas como critério de agrupamento dos clientes, por essa razão foram removidas do *dataset*.

A Figura 11 apresenta o resultado da definição das colunas selecionadas. É importante ressaltar que a relação apresentada nesta figura ainda não corresponde as colunas do *dataset* final.

Figura 11 - Resultado da seleção das colunas consideradas relevantes

```
NFS_EMPRESA                          int64
EMP_NOME                            object
NFS_NUMERO                           int64
NFS_DATA_CANCELAMENTO              float64
NFP_PRODUTO                          int64
NFS_CLIENTE                          int64
NFP_UNIDADE_PRODUTO                 object
NFP_QTDE_PRODUTO                   float64
NFP_TOTAL_PRODUTO                  float64
NFP_PRECO_UNITARIO                 float64
NFS_DATA_EMISSAO           datetime64[ns]
NFP_PRODUTO_DESCRICAO               object
NFP_PECAS                          float64
CF_RAZAO                            object
CID_DESCRICAO                       object
CID_UF                              object
PPR_TOTAL_PESO                     float64
TIPO                                object
NFS_VALOR_TOTAL_NOTA               float64
NFP_TOTAL_PRODUTO_BRUTO            float64
TIPO_NF                             object
DIA_SEMANA                           int64
DATA_SEMANA                         object
GRUPO_PRODUTO                       object
SUBGRUPO_PRODUTO                    object
ATIVIDADE_CLIENTE                   object
dtype: object
```

Fonte: autoria própria, (2023)

3.3.2 Limpeza

A limpeza dos dados busca eliminar os ruídos identificados sobre os dados durante a análise exploratória. Com base nos resultados encontrados e nos *insights* gerados foram implementados métodos de consultas e aplicações de filtros manipulando os dados de forma que seja possível selecionar somente àqueles de maior relevância.

Diante de algumas descobertas, percebeu-se que no *dataset* existiam vendas associadas 3 diferentes empresas, conforme apresenta a Tabela 6. No entanto, segundo o analista de negócio, somente as vendas relacionadas à FRIGOSOL – FRIGORIFICO SUL BAHIA deveriam ser consideradas, pois correspondiam aos produtos de interesse da empresa. Sendo assim, foram removidas todas amostras de dados cuja coluna NFS_EMPRESA apresentaram os valores iguais a 1 ou 4.

Tabela 6 - Quantidade de produtos vendidos por empresa

NFS_EMPRESA	EMP_NOME	Quantidade de produtos vendidos
3	FRIGOSOL – FRIGORIFICO SUL BAHIA – ME	112.346
1	FRIGORIFICO REGIONAL SUDOESTE LTDA	3.688
4	SEARA SERV DE TRANSP E COM DE UTIL DOMES LTDA-EPP	3

Fonte: autoria própria (2023)

Seguindo na abordagem na seleção dos dados relevantes, foi analisadas as amostras de dados segundo os tipos de notas fiscais. Segundo a análise exploratória, o tipo “V” da tabela TIPO_NF correspondiam as vendas e o tipo “D” correspondiam as devoluções. Para este trabalho somente as notas fiscais do tipo “V” faziam sentido, portanto todas as amostras cuja coluna TIPO_NF apresentasse o valor “D” foram desconsideradas.

A próxima coluna avaliada foi GRUPO_PRODUTO. Como descrito na etapa de inspeção esta coluna é utilizada para classificação de produtos. Conforme análise, constatou-se que os grupos GRAXARIA, IMOBILIZADO – VEÍCULOS, ALMOXARIFADO – USO E CONSUMO INDUSTRIAL, TRANSPORTE, EQUINOS, ASININO e MUARES não foram considerados relevantes. Desta lista, os 4 primeiros itens estão relacionados aos produtos que não são de consumo humano ou correspondem a veículos, adubos, etc. Quanto aos 3 últimos, referem-se a produtos destinados à exportação não sendo produtos consumidos pelo mercado interno. Portanto, todas as amostras de dados relacionados aos grupos mencionados foram removidos do *dataset*.

Quanto à coluna SUBGRUPO_PRODUTO, foram aplicados os mesmos critérios aplicados na coluna GRUPO_PRODUTO, ou seja, foram desconsiderados os subgrupo de produtos que não correspondiam aos produtos aptos ao consumo humano ou que são comercializados no mercado interno. Portanto, os subgrupo de produtos PET, MUARES, SUBPRODUTO NAO COMESTIVEL, ALMOXARIFADO – PROTEA, JAPAN, NAO COMESTIVEIS, OVINO e TF foram removidos do *dataset*.

A coluna TIPO, por sua vez, classifica à venda em 5 diferentes tipos: PRODUTO ACABADO, MATERIA PRIMA, SERVICOS, OUTROS e CONSUMO. Dos tipos listados o tipo PRODUTO ACABADO representa 99% dos dados. Como os

outros tipos estão relacionadas a tipos de vendas não relevantes e, ainda, possuem poucas amostras de dados, não convém mantê-los no *dataset*.

Por fim foram analisados os dados da coluna ATIVIDADE_CLIENTE. De acordo com a análise exploratória, esta coluna refere-se a classificação dos clientes em relação às suas atividades econômicas. No *dataset* foram identificadas 13 diferentes atividades, contudo as atividades FUNCIONARIO, EMPRESAS DO GRUPO, FRETE SEARA, REMESSA e COURO não estão relacionadas diretamente com vendas aos clientes. Conforme confirmado pelo analista de negócio estas vendas estão correspondem às vendas feitas a funcionários, à empresas do próprio grupo, remessa, frentes e venda de couros. Após análise e validação junto ao analista da empresa conclui-se que tais vendas não aderem aos escopo do trabalho, portanto, todos os dados associados as mesmas foram removidas do *dataset*. Ainda em relação à coluna ATIVIDADE_CLIENTE, foi constatado que 4 clientes estavam associados a mais de uma atividade. Neste caso, decidiu-se por associar cada cliente a somente uma atividade e, portanto, foi selecionada a atividade com mais vendas associadas.

Um outro ponto que requer atenção durante o processo de limpeza dos dados é a presença de registros duplicados ou dados que possuem o mesmo significado. Comumente, estas situações correspondem a pequenas diferenças detectadas em colunas descritivas causados, por exemplo, por erros de digitação. Para lidar com estas situações foram implementadas 3 funções: remover_acentos, get_duplicated e merge_duplicados. A primeira função remove acentos e caracteres especiais como forma de padronizar os dados. A segunda função identifica os registros duplicados e a terceira faz a consolidação desse dados.

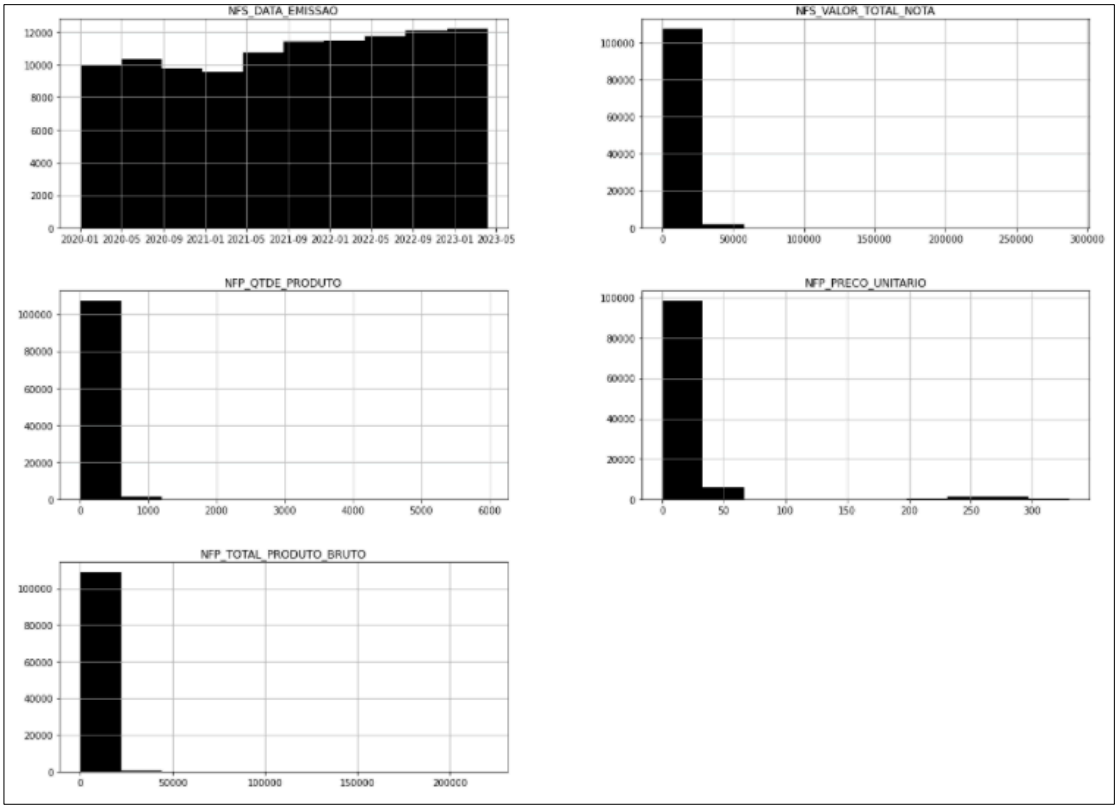
3.3.3 *Outilers*

Segundo Lv (2019), o *outlier* corresponde a um pequeno subconjunto de pontos de dados, onde suas medições desviam de forma significativa da maioria das amostras do conjunto de dados aos quais eles pertence. Os *outliers* podem impactar diretamente na eficiência dos modelos de *machine learning* seja aumentando o tempo de processamento do modelo ou gerando resultados imprecisos.

Uma estratégia inicial para identificar a presença de possíveis *outliers* no *dataset* é gerar histogramas das colunas que armazenam dados discretos. Os histogramas apresentam como os dados estão distribuídos e essa análise pode ser o

passo inicial para análise mais detalhadas utilizando outras ferramentas. Sendo assim, foram gerados os histogramas paras as colunas NFS_DATA_EMISSAO, NFS_VALOR_TOTAL_NOTA, NFP_QTDE_PRODUTO, NFP_PRECO_UNITARIO e NFP_TOTAL_PRODUTO_BRUTO conforme apresenta a Figura 12.

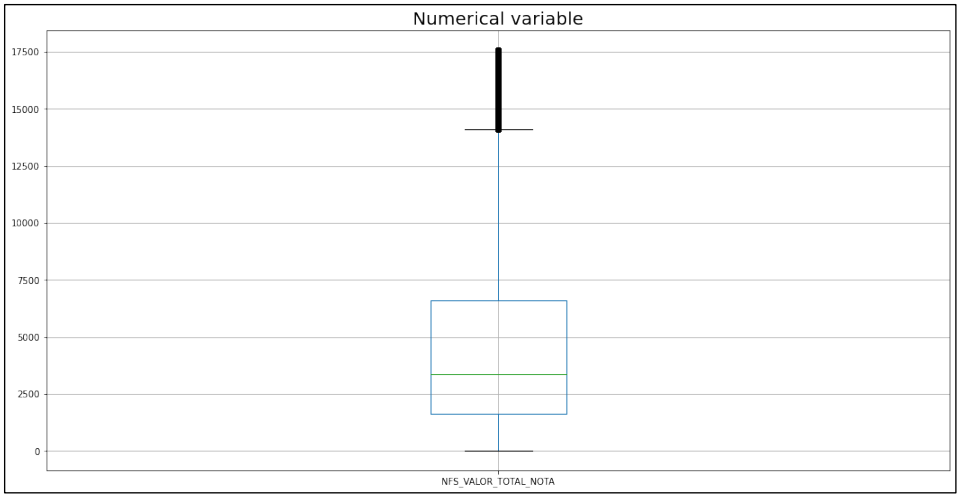
Figura 12 - Histograma das colunas numéricas



Fonte: elaborado pelo autor, (2023)

De acordo com os gráficos acima, as colunas NFS_VALOR_TOTAL_NOTA, NFP_QTDE_PRODUTO, NFP_PRECO_UNITARIO e NFS_TOTAL_PRODUTO_BRUTO apresentaram uma distribuição em seus dados bastante irregular, sendo forte indício de *outliers*. Para uma melhor análise foram gerados gráficos *boxplot* os quais fornecem informações detalhadas A Figura 13 representa o *boxplot* dos dados da coluna NFS_VALOR_TOTAL_NOTA.

Figura 13 - Gráfico *boxplot* para detecção de outliers utilizando o intervalo interquartílico

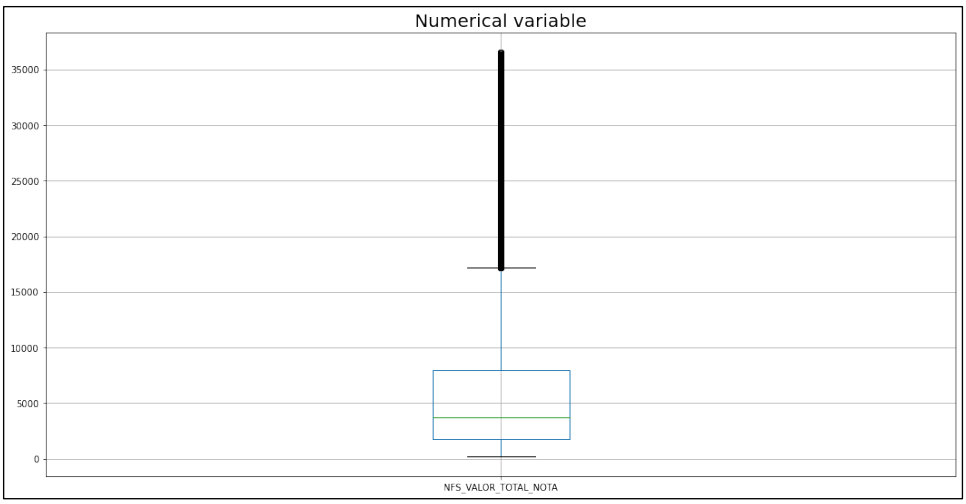


Fonte: elaborado pelo autor (2023)

O *boxplot* separa dos dados em quartis. O 1º quartil corresponde à primeira linha horizontal inferior do retângulo e representa 25% dos dados abaixo de R$ 1.743,00. O 2º quartil corresponde a mediada, representada pela linha verde dentro do retângulo e apresenta um valor de R$ 3.728,00. O 3º quartil corresponde a terceira linha horizontal do retângulo com valor de R$ 8.078,00 representando 75% dados. O limite de detecção dos *outliers* é calculado pelo intervalo interquartílico (IQR) que corresponde a diferença do 3º quartil com o 1º quartil. O IQR corresponde a 50% dos dados. A partir do IQR obtém-se o limite inferior que é calculado pelo valor do 1º quartil menos uma vez e meia o IQR. O limite superior é calculado pelo 3º quartil mais uma vez e meia o IQR. Com base nesses cálculos define-se o limite superior de detecção de *outlier* em R$ 17.580,00, que representa 7% dos dados. O mesmo processo aplicado para análise da coluna NFS_VALOR_TOTAL_NOTA foi também aplicado para as colunas NFP_QTDE_PRODUTO, NFP_PRECO_UNITARIO e NFP_TOTAL_PRODUTO_BRUTO.

Outro método utilizado para determinar os limites para detecção de *outliers* foi eliminar todos os registros maiores que o 99º percentil e todos os registros menores do que o 1º percentil. A Figura 14 apresenta o *boxplot* resultante desta abordagem.

Figura 14 - Gráfico boxplot para detecção de outliers considerando os valores extremos que se encontram acima e abaixo dos 99º e 1º percentis



Fonte: elaborado pelo autor (2023)

O valor limite encontrado foi de R$ 37.000,00. Como os limites encontrados apresentaram uma diferença significativa, decidiu-se por analisar as vendas que se encontravam acima desses valores. Com o auxílio do especialista de negócios da empresa, conclui-se que as vendas acima do limite de R$ 37.000,00 representavam situações de vendas pontuais e atípicas, relacionadas às vendas promocionais ou vendas pontuais feitas aos clientes do ramo atacadista. Conclui-se que todos os registros com NFS_VALOR_TOTAL_NOTA acima de R$ 37.000,00 seriam outliers e, portanto, seriam desconsiderados.

3.3.4 Transformação dos dados

Como parte da estratégia para alcançar uma melhor performance dos algoritmos de *machine learning*, são aplicadas técnicas de transformação dos dados deixando-os em um formato mais adequado. Estas transformações envolve alterações em seus tipos, mudanças em suas escaldas e, em alguns casos, novas *features* são criadas buscando melhorar a representação do negócio que está sendo analisado (GE *et al*., 2017).

A conclusão de todas as etapas anteriores, permite que se tenha um entendimento mais profundo dos dados e do negócio. Neste sentido, foi constatado que o *dataset* inicial não apresentava as características apropriadas para que pudesse extrair os padrões de similaridade necessários para a formação dos clusters.

Basicamente, as características até então disponíveis estavam mais relacionadas aos produtos vendidos do que aos próprios clientes em si, apesar dos clientes representarem uma dessas características. Através deste entendimento e com base nos insights gerados durante a análise exploratória, conclui-se que seria necessário a criação de novas *features* que pudessem melhor representar as características de compra dos clientes. Sobre o processo de criação de novas *features*, Chapman *et al*. (2000) informa que, por vezes, esta tarefa é bastante intuitiva, ou seja, não existe um processo padrão que possa ser a todas as situações, em todos os projetos, a decisão de qual *feature* será criada depende muito do domínio que o cientista de dados tem sobre os dados e sobre os negócios. A Tabela 7 apresenta a relação das novas *features* criadas a partir do *dataset* inicial.

Tabela 7: Relação das novas *features* que compõe o *dataset* final

Feature	**Descrição**
FATURAMENTO_TOTAL	Corresponde a soma de todas as vendas feitas.
MEDIA_FATURAMENTO_MENSAL	Média aritmética mensal das compras realizadas por cada cliente.
FATURAMENTO_ULTIMA_COMPRA	Valor total da última compra realizada pelo cliente.
NRO_DIAS_ULTIMA_COMPRA	Número de dias desde a última compra, tendo como a data base o dia 30/03/2023.
NRO_ ITENS_COMPRADOS	Quantidade total de produtos compradas por cada cliente
QTDE_DE_COMPRAS_DIA_SEMANA	Somatório de todas as compras realizadas em cada dia da semana
CIDADE	Cidade do cliente
PRODUTOS_MAIS_COMPRADO	Lista dos 5 produtos mais comprados por cada cliente (cesta de produtos)

Fonte: elaborado pelo autor, (2023)

A Figura 15 apresenta um trecho de código da rotina de processamento dos dados desenvolvida para a criação das novas *features*.

Figura 15 - Rotina para processamento dos dados para o novo dataset

```
colunas_finais = [
    'CLIENTE',
    'CIDADE',
    'DIAS_PRIMEIRA_COMPRA',
    'DIAS_ULTIMA_COMPRA',
    'QUANTIDADE_COMPRAS_REALIZADAS',
    'VOLUME_TOTAL_COMPRADO',
    'QUANTIDADE_PRODUTOS_COMPRADOS',
    'VALOR_TOTAL_COMPRADO',
    'VALOR_MEDIO_COMPRA',
    'INTERVALO_DIAS_COMPRAS'
]

dados = []
for name, group in frigosol.groupby(['CF_RAZAO','CID_DESCRICAO']):
    linha = []
    # CLIENTE
    linha.append(name[0])

     # CIDADE
    linha.append(name[1])

    data_base = datetime.strptime('2023-03-31', '%Y-%m-%d')
    primeira_compra = group['NFS_DATA_EMISSAO'].min()
    ultima_compra = group['NFS_DATA_EMISSAO'].max()

    # DIAS_PRIMEIRA_COMPRA
    dias_primeira_compra = abs((data_base - primeira_compra).days)
    linha.append(dias_primeira_compra)

    # DIAS_ULTIMA_COMPRA
    dias_ultima_compra = abs((data_base - ultima_compra).days)
    linha.append(dias_ultima_compra)

    # QUANTIDADE_COMPRAS_REALIZADAS
    quantidade_compras = len(group['NFS_NUMERO'].unique())
    linha.append(quantidade_compras)

    # VOLUME_TOTAL_COMPRADO
    linha.append(group['NFP_QTDE_PRODUTO'].sum())

    # QUANTIDADE_PRODUTOS_COMPRADOS
    linha.append(group['NFS_NUMERO'].count())

    # VALOR_TOTAL_COMPRADO
    faturamento_total = group['NFP_TOTAL_PRODUTO'].sum()
    linha.append(faturamento_total)

    # VALOR_MEDIO_COMPRA
    linha.append((faturamento_total / quantidade_compras))

    # INTERVALO_DIAS_COMPRAS
    intervalo_compras = (dias_primeira_compra - dias_ultima_compra)/quantidade_compras
    linha.append(intervalo_compras)

    dados.append(linha)
    print('----------------')
    print('CLIENTE: ' + name[0])
```

Fonte: autoria própria (2023)

Nesta rotina os produtos são agrupados por clientes e da relação destes produtos são calculadas as novas *features*. Ao final deste processamento, as 100.000 linhas existentes no *dataset* inicial foram condensadas para 1.524, onde cada linha do novo *dataset* corresponde a um cliente.

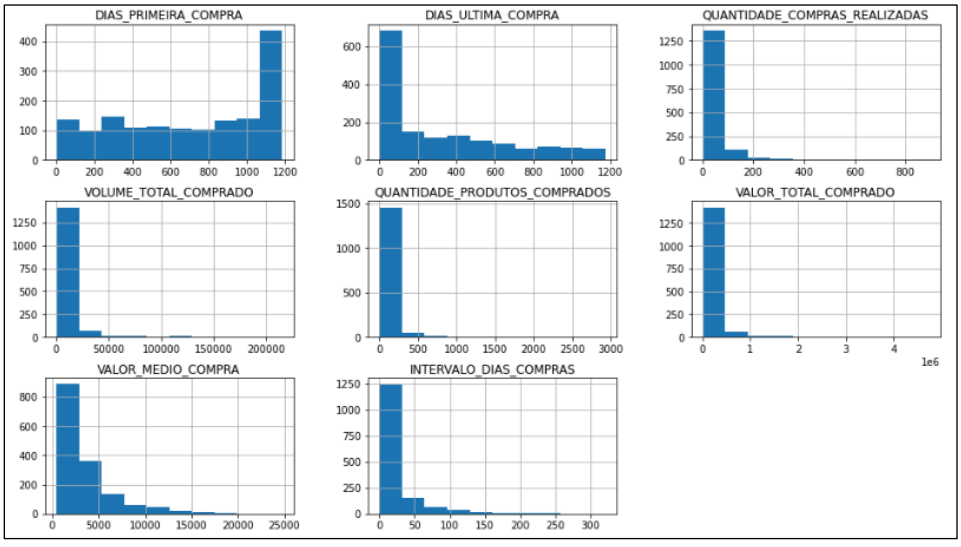
Ao final deste processo, foi realizada mais uma análise, agora sobre os novos dados. Basicamente foi feita uma avaliação entender como estes dados estavam distribuídos. Mais uma vez foram geradas informações estatísticas básicas conforme apresenta a Tabela 8. Observa-se que a distribuição dos dados em todas as *features* varia bastante. Por exemplo, a coluna VOLUME_TOTAL_COMPRADO varia de 3,00 a 215.470,00 entre o valor máximo e o mínimo, já a coluna INTERVALO_DIAS_COMPRA varia de 0 a 321,33. Estas informações são visualmente apresentadas pelos histogramas apresentados na Figura 16.

Tabela 8 - Informações estatísticas básicas no novo *dataset*

Colunas	Média	Desvio Padrão	Mín.	Máx.
DIAS_PRIMEIRA_COMPRA	714,44	389,88	1,00	1.185,00
DIAS_ULTIMA_COMPRA	316,33	344,80	0,00	1.175,00
QUANTIDADE_COMPRAS_REALIZADAS	34,78	63,17	1,00	891,00
VOLUME_TOTAL_COMPRADO	6853,19	17.358,50	3,00	215.470,00
QUANTIDADE_PRODUTOS_COMPRADOS	67,54	166,02	1,00	2.928,00
VALOR_TOTAL_COMPRADO	141.034,63	365.852,28	433,14	4.728.499
VALOR_MEDIO_COMPRA	3.444,75	3.157,65	433,14	24775,70
INTERVALO_DIAS_COMPRAS	22,20	33,03	0,00	321,33

Fonte: autoria própria (2023)

Figura 16 - Gráficos que apresentam a distribuição dos dados do novo dataset



Fonte: autoria própria (2023)

Escalas tão diferentes impacta negativamente na performance nos modelos de *machine learning*, sobretudo em sua acurácia e no tempo de processamento

utilizando durante o processo de treinamento. Para resolver este problema foram aplicadas duas técnicas de transformação de dados: a normalização e a padronização. Segundo Bruce e Bruce (2019), a padronização é uma técnica que deixa os dados de todas as *features* em escalas semelhantes subtraindo os dados da média e depois dividindo pelo desvio padrão. A normalização força que os dados fiquem entre uma faixa que varia de 0 e 1. Estas transformações garantem as *features* influenciem o modelo de forma mais equilibrada caso fosse considerada a escala original. A biblioteca do *Scikit-Learn* disponibiliza as classes *StandardScaler* e *MinMaxScaler* do pacote *sklearn.preprocessing* que facilita estas transformações. A título de exemplo, a Figura 17 exibe as 5 primeiras linhas com o resultado da normalização e padronização dos dados.

Figura 17 - Gráficos que apresentam a distribuição dos dados do novo *dataset*

	DIAS_PRIMEIRA_COMPRA	DIAS_ULTIMA_COMPRA	QUANTIDADE_COMPRAS_REALIZADAS	VOLUME_TOTAL_COMPRADO	QUANTIDADE_PRODUTOS_COMPRADOS
0	0.588682	0.558298	0.006742	0.003636	0.004100
1	0.105574	0.000851	0.014607	0.028029	0.008541
2	0.065034	0.066383	0.000000	0.000149	0.000000
3	0.996622	0.001702	0.278652	0.509445	0.242228
4	0.996622	0.001702	0.243820	0.274418	0.168432

Fonte: autoria própria (2023)

3.4 Modelagem

De acordo com Chapman *et al*. (2000), a fase de modelagem possui 4 etapas, que são: seleção da técnica de modelagem, design de teste, construção do modelo e avaliação do modelo.

3.4.1 Seleção das técnicas de modelagem

Como subsídio para o processo de seleção das técnicas de modelagem, foram considerados os algoritmos de agrupamento apresentados na seção 2 e as características dos dados associados ao *dataset* identificadas na fase de preparação dos dados. A partir daí, buscou-se identificar as características que apresentavam

correspondência e compatibilidade entre as características dos algoritmos e as características dos dados. Em relação ao *dataset* foram observados a sua volumetria, o número de *features* existentes, os tipos de dados de cada *features* e a presença e importância de se ter ou não *outliers*. Quanto aos algoritmos, foram analisados o seu desempenho, as estratégias adotadas para a criação dos *clusters* e a sensibilidade em relação aos dados discrepantes ou desbalanceados. Após estas análise, foram selecionados três algoritmos: o *K-Means*, o DBSCAN e o *Hierarchical clustering*.

Um dos principais aspectos observados para selecionar o *K-Means* foi o fato dele ser amplamente referenciado em diversos estudos e artigos científicos como, por exemplo, Ge *et al.* (2017), Saxena *et al.* (2017), Abbas (2008), Usama (2019) e Tripathi, Bhardwaj e Poovammal (2018), comprovando a sua versatilidade e consistência. Além disso, é importante destacar o seu bom desempenho, justificado pela estratégia de criação dos clusters (ABBAS, 2008). No *K-Means*, é obrigatório que o número de clusters seja informado antes que o algoritmo seja treinado, portanto, esta é uma informação prévia. O simples fato de já conhecer o número de clusters que se pretende criar já diminui a quantidade de iterações necessárias para realocar os membros dos *clusters* (ABBAS, 2008).

A justificativa pela qual o DBSCAN foi selecionado está relacionada com a estratégia utilizada para criar os *clusters*. No DBSCAN a quantidade de *clusters* não precisa ser informada previamente (BIRANT; KUT, 2007). Segundo Birant e Kut, esta habilidade tem como referência a densidade de suas amostras (também de chamadas de pontos), na qual é obtida contando o número de amostras em uma região de raio específico ao redor daquela amostra. Caso uma amostra apresente densidade acima do limite especificado, um novo *cluster* é criado (BIRANT; KUT, 2007).

Já *o Hierarchical clustering* foi selecionado por apresentar uma abordagem de agrupamento diferente dos outros dois algoritmos. Este algoritmo analisa a hierarquia dos pontos de dados (amostras) e como eles se movem dentro do *cluster* ou fora dele (TRIPATHI; BHARDWAJ; POOVAMMAL, 2018). Dessa forma, é possível escolher o número de *clusters* a partir do dendrograma criado pelo algoritmo, bastando para isso, selecionar a distância máxima deseja entre os *clusters* e, a partir daí, traça-se uma linha de corte naquela posição definindo o número máximo de *clusters* (TRIPATHI; BHARDWAJ; POOVAMMAL, 2018). Esta estratégia permite que os *clusters* seja combinados ou divididos de acordo com sua hierarquia (ABBAS, 2008).

3.4.2 Design de teste

Segundo Chapman *et al*. (2000), a segunda etapa da fase de modelagem consiste na definição dos procedimentos que deverão ser adotados para testar a qualidade e validade dos modelos criados. Enquanto outras técnicas de aprendizagem utilizam métricas triviais para avaliar o desempenho dos modelos como, por exemplo, a acurácia, a precisão e o *recall*, no aprendizado não supervisionado não se pode levar em conta somente os valores absolutos dos rótulos encontrados pelos algoritmos, deve-se avaliar a separação dos dados similares supondo quais membros pertencem ao mesmo grupo analisando as suas semelhanças com base em suas características (SCIKIT-LEARN, 2007).

Sendo assim, para avaliação e validação dos modelos foram consideradas duas abordagens: (i) a primeira considera as métricas de valores absolutos, neste caso, foram considerados o número de clusters e o coeficiente de silhueta de cada modelo (BAGIROV, ALIGULIYEV, e SULTANOVA, 2023); (ii) a segunda, de carácter mais intuitivo, se apoia nos princípios de homogeneidade e heterogeneidade.

Conforme explicado na seção 2.4.2, alguns algoritmos de *clustering* exigem que seja informado em sua inicialização o número de *clusters* que se pretende formar. Segundo com Dinh, Fujinami e Huynh (2019), esse número é informado de forma aleatória, o que pode causar uma superestimação ou subestimação dos números de *clusters*. Para resolver esta questão, foi adotado o método *elbow* para encontrar o número de *clusters* para os algoritmos que exigem esse número em sua inicialização. O método *elbow* será explicado em mais detalhes na próxima seção. Em relação ao coeficiente de silhueta, trata-se de um coeficiente que identifica o quão compacto o *cluster* está e o quão separados os *clusters* estão entre si (BAGIROV, ALIGULIYEV e SULTANOVA, 2023)

Quanto aos princípios de homogeneidade e heterogeneidade, o primeiro diz respeito à similaridade das amostras dentro de cada *cluster* e o segundo corresponde a dissimilaridade entre amostras de *clusters* diferentes (RACHWAT , 2023).

3.4.3 Construção dos modelos

A terceira etapa da fase de modelagem corresponde à construção dos modelos *clustering* e, para isso, foi utilizada a biblioteca *Scikit-Learn*. Escrita em Python, o *Scikit-Learn* oferece uma interface de programação simplificada, onde são disponibilizados pacotes que reúnem classes que implementam os diversos algoritmos de *machine learning* (BUITINCK *et al*., 2013). Através dessas classes os modelos são instanciados e treinados através dos seus métodos e parâmetros

Como mencionado na etapa anterior, alguns dos algoritmos de *clustering* requer que o número de *clusters* que se pretende criar seja informado previamente, de modo que, este parâmetro deve ser passado no momento em que os modelos são instanciados. Mas, como saber se o número de *clusters* informado na inicialização do modelo é o mais adequado? Para solucionar esta questão foi adotado o método do cotovelo (*elbow method*), que consiste em tentar diferentes valores para o parâmetro que define o número de *clusters* que se pretende criar. Os valores testados são valores inteiros entre 1 a raiz quadrada do número de observações do *dataset*. Então, para cada iteração é obtida o valor de inércia, que corresponde à soma do quadrado das distâncias entre cada ponto e o ponto central de cada *cluster* criado pelo modelo. Com isso, espera-se que se obtenha valores menores para esta métrica a medida que o valor número de *cluster* aumenta. Em relação aos modelos que implementam os algoritmos que não exigem que sejam informados o número de *cluster* em sua inicialização, esta técnica não é aplicada e, portanto, os modelos são instanciados, treinados e, posteriormente, são apresentados seus os resultados.

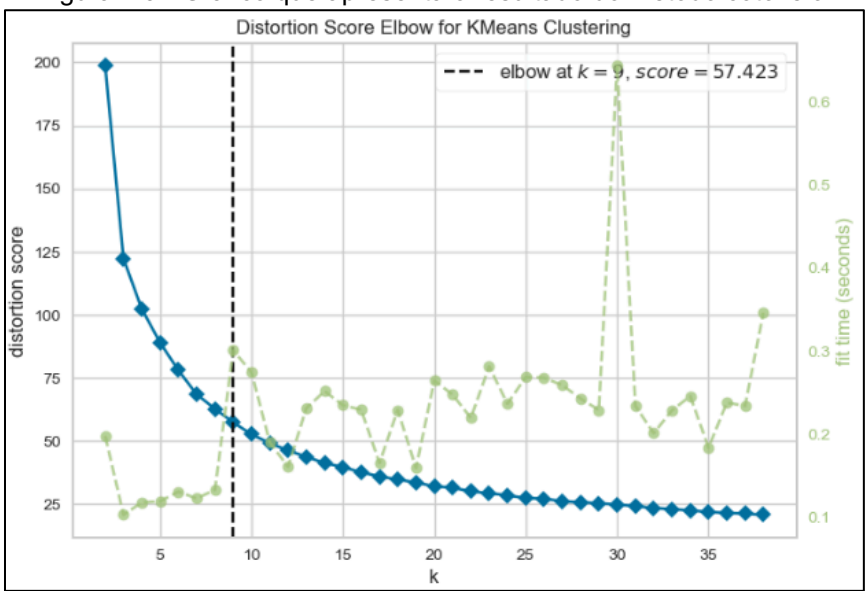
Conforme descrito na etapa de seleção da técnica de aprendizagem, o primeiro modelo criado foi o *K-Means*. O *K-Means* é um dos algoritmos que requer que o número de *clusters* que se pretende criar seja informado previamente. Então, conforme explicado, foi aplicado o método do cotovelo, o qual apresentou o número de clusters igual a 9. A Figura 18 apresenta o trecho de código que demostra a implementação do método do cotovelo e a Figura 19 apresenta o gráfico com o resultado obtido.

Figura 18 - Trecho de código com implementação em Python do método cotovelo

```
from yellowbrick.cluster import KElbowVisualizer

end = int(math.sqrt(df.shape[0]))

kmeans = KMeans(random_state=42)
elbow = KElbowVisualizer(kmeans, k=(2, end))

elbow.fit(df.iloc[:,0:8])
elbow.show()
```

Fonte: autoria própria (2023)

Figura 19 - Gráfico que apresenta o resultado do método cotovelo

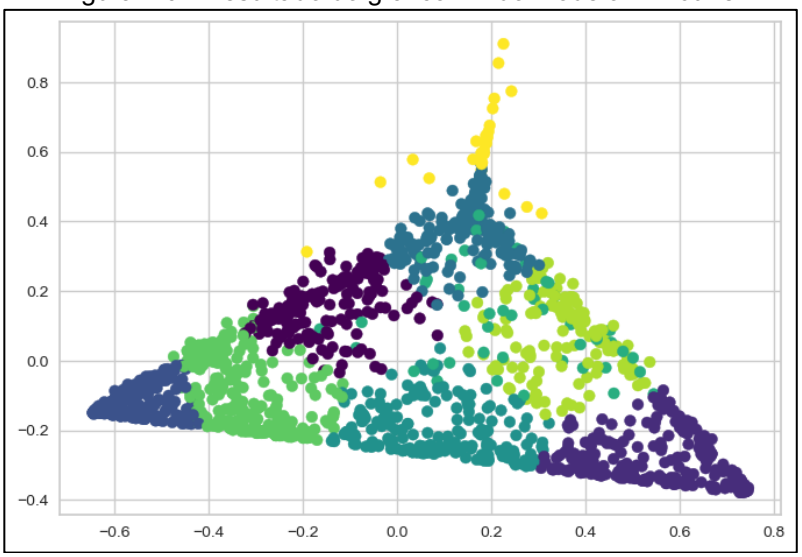


Fonte: autoria própria (2023)

Conhecendo o número ideal de *clusters*, o modelo do *K-Means* foi instanciado e treinado utilizando o *dataset* obtido na etapa de preparação dos dados. O *Scikit-Learn* oferece alguns parâmetros que permitem que os modelos sejam ajustados minuciosamente. Para este trabalho foram considerados os parâmetros mais relevantes de cada modelo. No *K-Means*, além do número de clusters, foram selecionados *random_state* utilizado para reproduzir clusters exatos repetidas vezes, esse parâmetro garante que uma amostra pertencerá sempre a um mesmo cluster mesmo o modelo sendo executado repetidas vezes. O valor para este parâmetro pode ser um número inteiro aleatório que seja abaixo de 1234 (SCKIT-LEARN, 2007). Outro parâmetro refere-se ao número máximo de iterações (*max_int*) que o modelo pode executar até que se encontre a localização dos *centróides* de cada *cluster*. O último parâmetro corresponde a qual algoritmo de aproximação de expectativa será utilizado, podendo ter duas opções: a opção padrão que é o "*lloyd*" e o "*elkan*". Em relação a este parâmetro foram testadas as duas opções, mas os resultados foram os mesmos.

Após a criação do modelo, o modelo foi treinado e seu resultado foi apresentado em um gráfico bidimensional conforme apresenta a Figura 20. O modelo bidimensional facilita a compreensão do resultado alcançado, embora o *dataset* seja multidimensional. Para chegar em uma representação bidimensional foi utilizado um algoritmo de redução de dimensionalidade PCA.

Figura 20 - Resultado do gráfico 2D do modelo *K-Means*



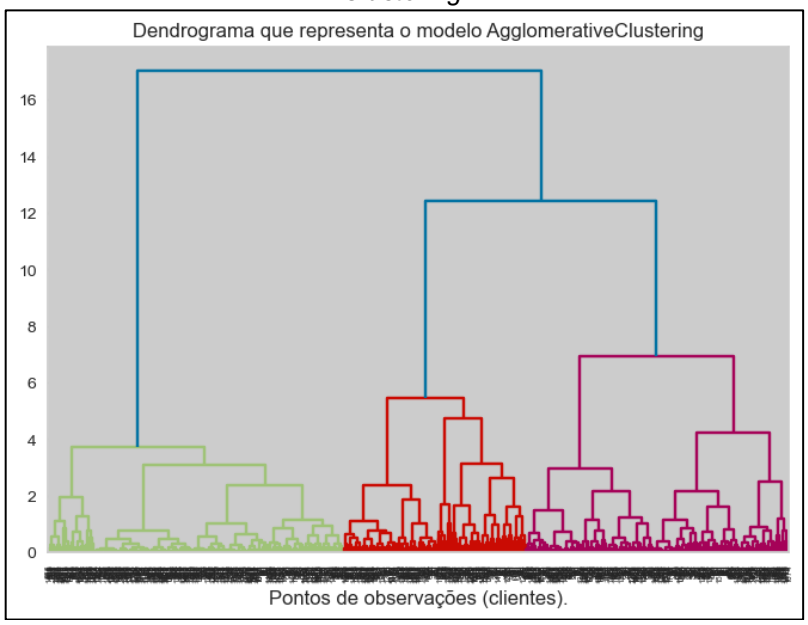
Fonte: autoria própria

O segundo modelo criado foi o *Hierarchical Clustering*. Conforme descrito na seção 2.4.3, este algoritmo define uma estrutura de agrupamento hierárquico onde os clusters podem ser fundidos ou separados sucessivamente de acordo com a abordagem e estratégia de ligação selecionada (SCIKIT-LEARN, 2007).

Segundo Dobilas (2021), antes o modelo hierárquico seja criado é recomendado que se construa o dendrograma como uma forma de representar a estrutura hierárquica dos clusters. O dendrograma é uma ferramenta que auxilia na identificação do número de *clusters* mais adequado para o modelo. Este número é identificado tracejando uma linha de corte horizontal que divide a estrutura hierárquica do dendrograma, de modo que, a quantidade de ramificações que ficarem abaixo desta linha corresponde ao número de *clusters* procurado. Contudo, para definir a posição onde a linha de corte será tracejada é necessário avaliar as alturas das linhas verticais que correspondem à distância de cada *cluster*. A partir do topo, a altura de cada linha vertical é calculada até o ponto de junção inferior. Compara-se as linhas verticais que estão no mesmo nível e escolhe a linha com a maior altura. De acordo com a **Erro! Fonte de referência não encontrada.**, a maior linha vertical é a primeira

linha da esquerda para direita cujo ponto de junção está próximo ao valor 4 do eixo y. Deste ponto para baixo verifica-se que existe outra linha vertical que seja maior do que a anterior. Como não existe uma linha maior, o ponto onde a linha de corte será traça será próximo do valor 4 do eixo y. Ao tracejar a linha, consegue cortar 7 ramificações, que corresponde ao número de *clusters* que será utilizado para na criação do modelo. A Figura 21 apresenta o dendrograma criado a partir do *dataset* disponibilizado na etapa de preparação dos dados.

Figura 21 - Dendrograma criado para o modelo *Agglomerative Clustering*



Fonte: autoria própria (2023)

Para criação do modelo e do dendrograma foram utilizadas as classes *AgglomerativeClustering* e *dendrogram*, ambas disponíveis na biblioteca *Scikit-Learn*. A Figura 22 exibe a função responsável pela criação do dendrograma apresentado na Figura 21.

Figura 22 - Função implementada para criar o dendrograma

```python
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram

def plot_dendrogram(modelo, **kwargs):
    # Matriz de ligação para plotar o dendrograma

    # contagem do número de amostra em cada nó
    counts = np.zeros(modelo.children_.shape[0])
    n_samples = len(modelo.labels_)
    for i, merge in enumerate(modelo.children_):
        current_count = 0
        for child_idx in merge:
            if child_idx < n_samples:
                current_count += 1  # leaf node
            else:
                current_count += counts[child_idx - n_samples]
        counts[i] = current_count

    linkage_matrix = np.column_stack(
        [modelo.children_, modelo.distances_, counts]
    ).astype(float)

    # Plotagem do dendrograma
    dendrogram(linkage_matrix, **kwargs)



hac = AgglomerativeClustering(distance_threshold=0, n_clusters=None)
hac = hac.fit(X)

plt.title("Dendrograma que representa o modelo AgglomerativeClustering")
# plot the top three levels of the dendrogram
plot_dendrogram(hac, truncate_mode="level", p=10)
plt.xlabel("Pontos de observações (clientes).")
plt.show()
```

Fonte: autoria própria (2023)

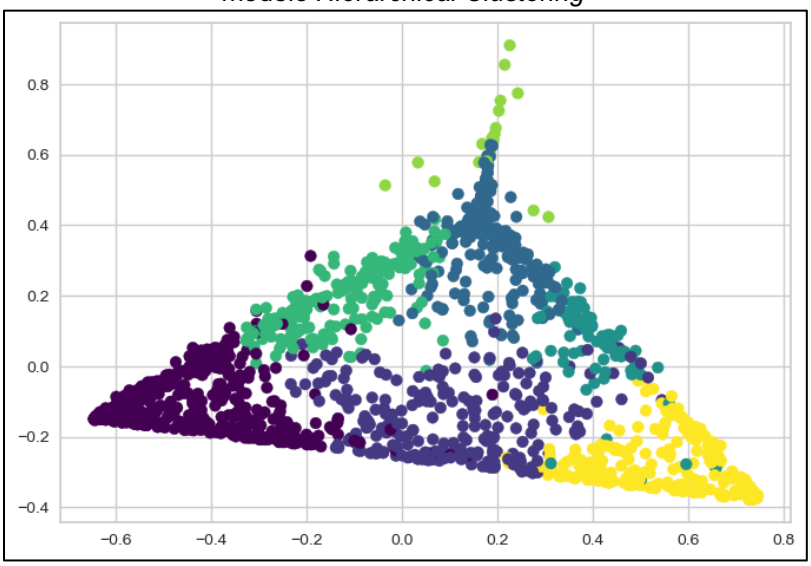
O trecho de código apresentado na Figura 23 cria o modelo *Agglomerative Clustering* com 7 *clusters*. Em seguida, para melhor representação dos clusters formados, utiliza-se o algoritmo de redução de dimensionalidade PCA. O resultado final é apresentado na Figura 24.

Figura 23 – Trecho de código para criação do modelo *Agglomerative Clustering*

```python
hac = AgglomerativeClustering(n_clusters=7)
hac = hac.fit(X)
labels = hac.labels_

X = df.iloc[:,0:8]
pca = PCA(2)

plot_columns = pca.fit_transform(X)
plt.scatter(x=plot_columns[:,0], y=plot_columns[:,1], c=labels, cmap="viridis")
plt.show()
```

Fonte: autoria própria (2023)

Figura 24 - Apresentação bidimensional dos clusters formados no modelo *Hierarchical Clustering*



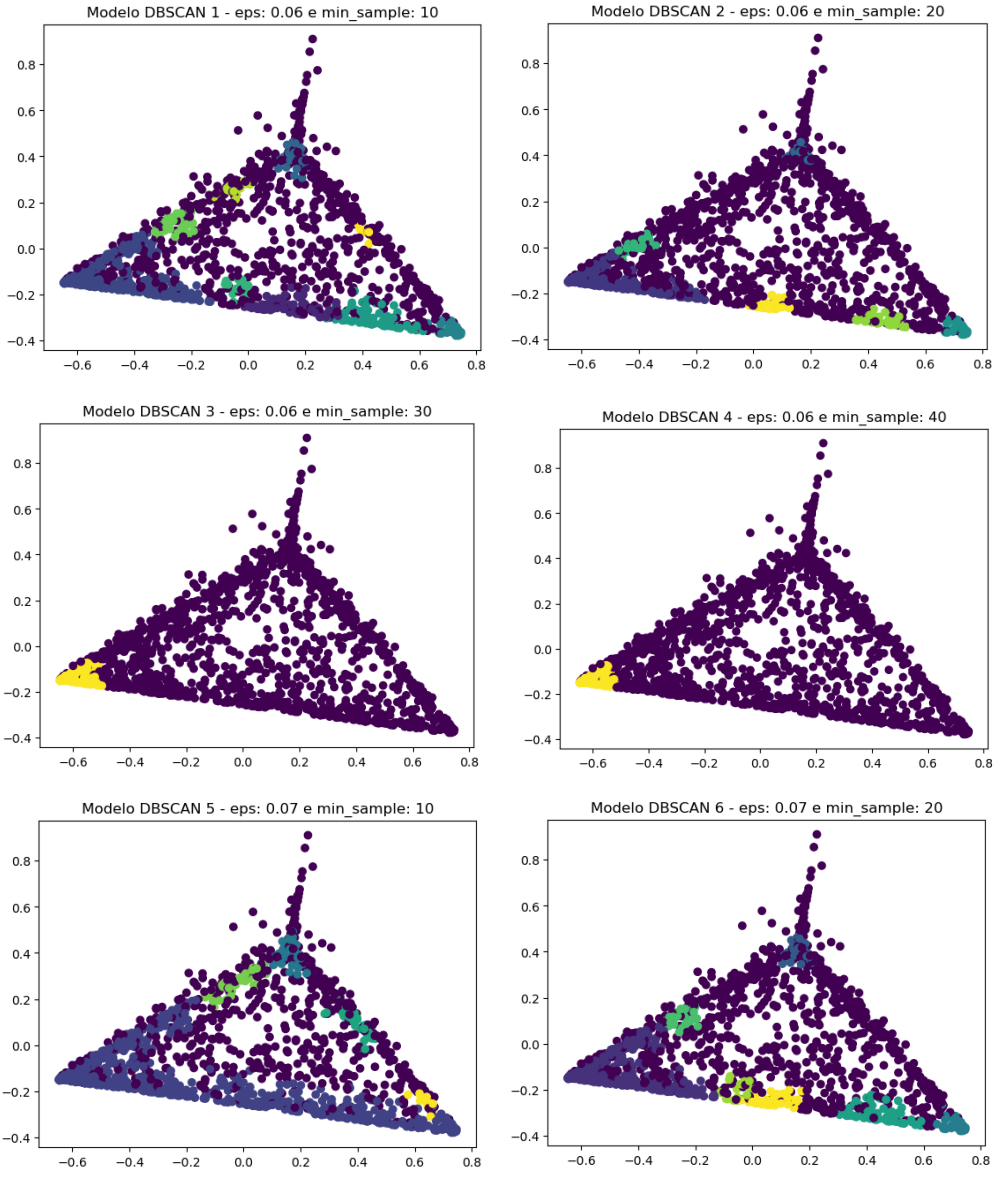
Fonte: autoria própria (2023)

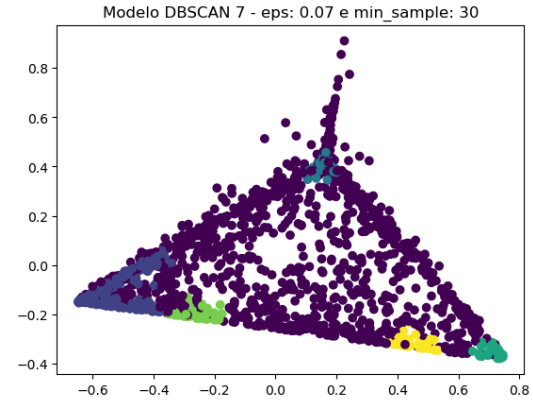
O último modelo foi criado a partir do algoritmo DBSCAN. Como explicado na seção 2.4.4, os *clusters* no DBSCAN são definidos de acordo com as áreas de alta e baixa densidade. Embora no DBSCAN não seja obrigatório informar o número de *clusters* em sua na inicialização, existem outros dois outros parâmetros que são requeridos: o *min_sample* (MinPts) e o *eps*. *O min_sample* corresponde ao número mínimo de amostras que precisão estar próximas (ou vizinhas) de um determinado ponto para que o *cluster* seja formado. E o *eps* especifica o raio da vizinhança, que está relacionado com a distância máxima para que dois pontos sejam considerados vizinhos um do outro (SCIKIT-LEARN, 2007).

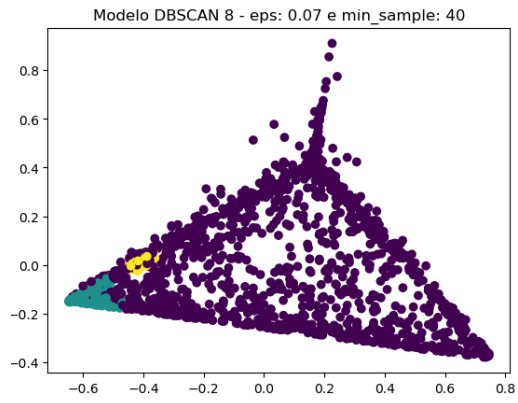
Entendendo o impacto que esses parâmetros têm no processo de criação dos *clusters*, Khandelwal (2021) adverte: se for informado um valor muito pequeno para o *eps* (raio de vizinhança) muitas amostras não serão associadas a nenhum *cluster*, consequentemente, muitas delas serão consideradas ruídos. Por outro lado, se o valor do *eps* for muito grande, isto pode fazer com que *clusters* muito próximos sejam fundidos, com isso, serão formados grupos grandes que poderiam ser divididos em grupos menores representando melhor o ambiente que está sendo modelado. Encontrar os valores mais adequados para o *eps* e para o *min_sample* é um processo que envolve sucessivas tentativas, onde para cada valor, o resultado obtido precisa ser avaliado e o recomendado é que seja avaliado pelo especialista do negócio (KHANDELWAL, 2021).

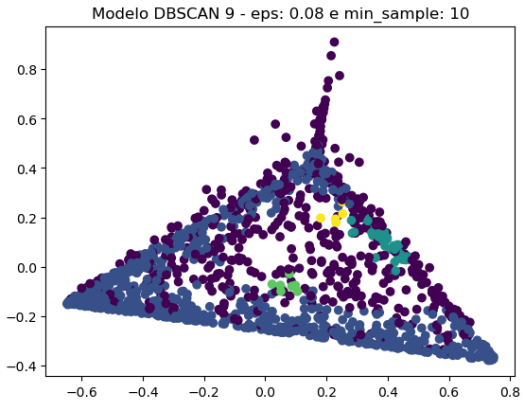
Com base nas explicações de Khandelwal (2021), foram criados 20 diferentes modelos baseado no DBSCAN, onde cada modelo corresponde ao resultado da combinação dos parâmetros *eps* e *min_sample*. A Figura 25 apresenta em formato de gráficos 2D os clusters formados por cada modelo.

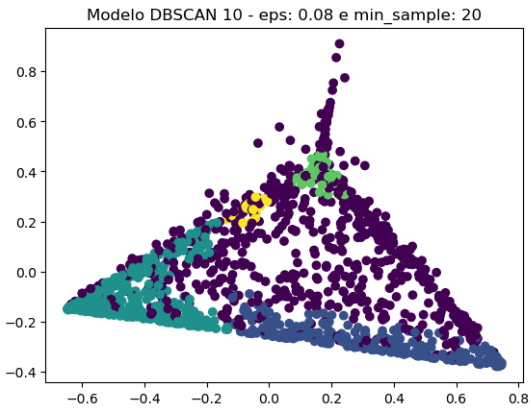
Figura 25 - Apresentação dos resultados do modelo DBSCAN para a faixa de valores definidas para os parâmetros eps e min_sample

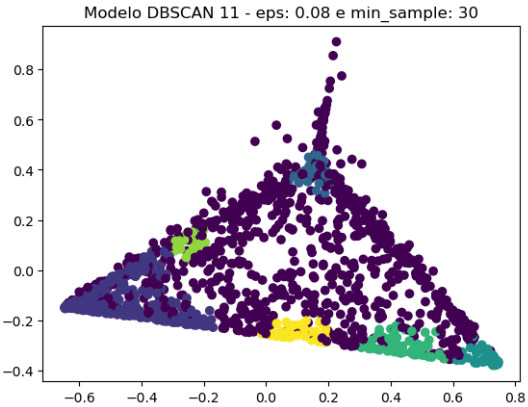


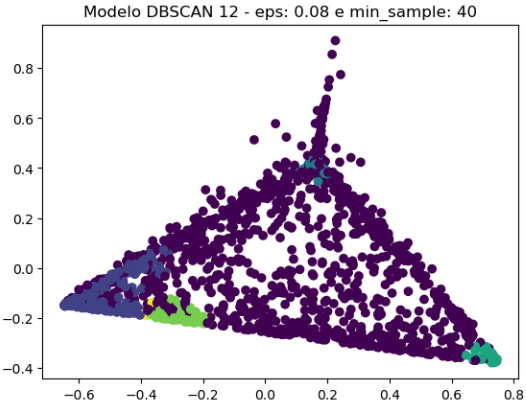

Modelo DBSCAN 7 - eps: 0.07 e min_sample: 30

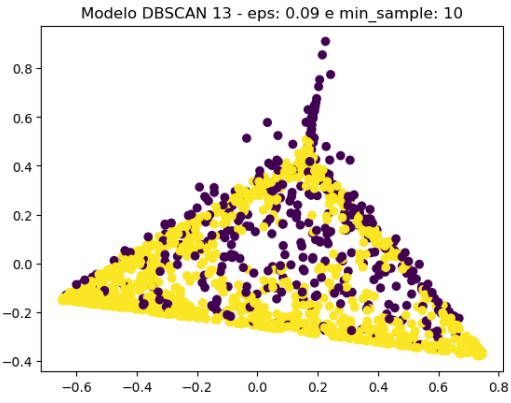

Modelo DBSCAN 8 - eps: 0.07 e min_sample: 40

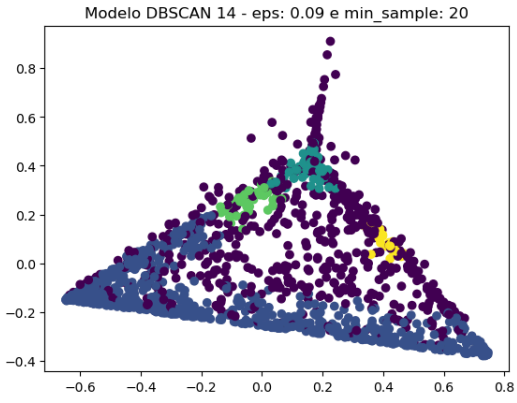

Modelo DBSCAN 9 - eps: 0.08 e min_sample: 10

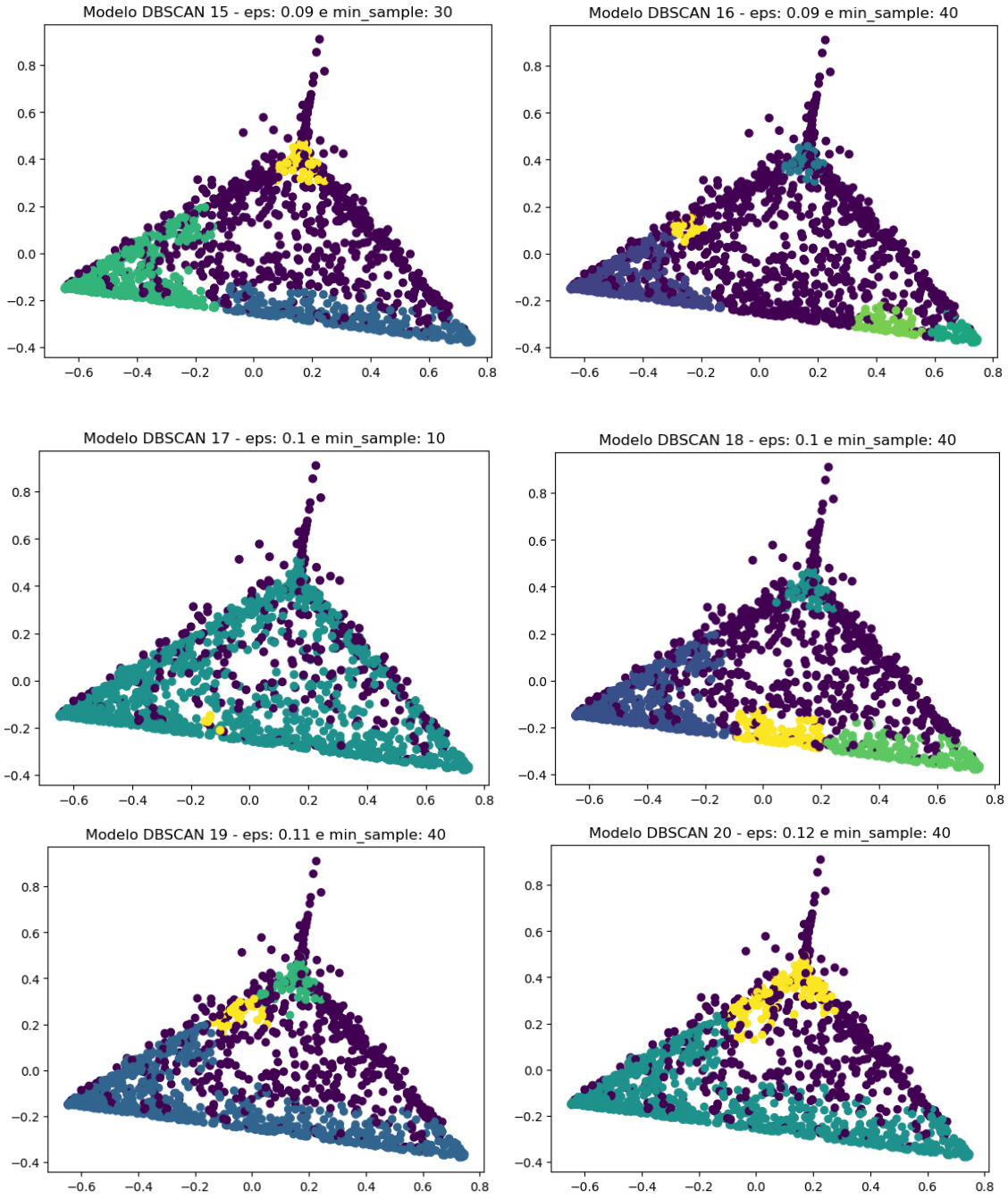

Modelo DBSCAN 10 - eps: 0.08 e min_sample: 20


Modelo DBSCAN 11 - eps: 0.08 e min_sample: 30


Modelo DBSCAN 12 - eps: 0.08 e min_sample: 40


Modelo DBSCAN 13 - eps: 0.09 e min_sample: 10


Modelo DBSCAN 14 - eps: 0.09 e min_sample: 20

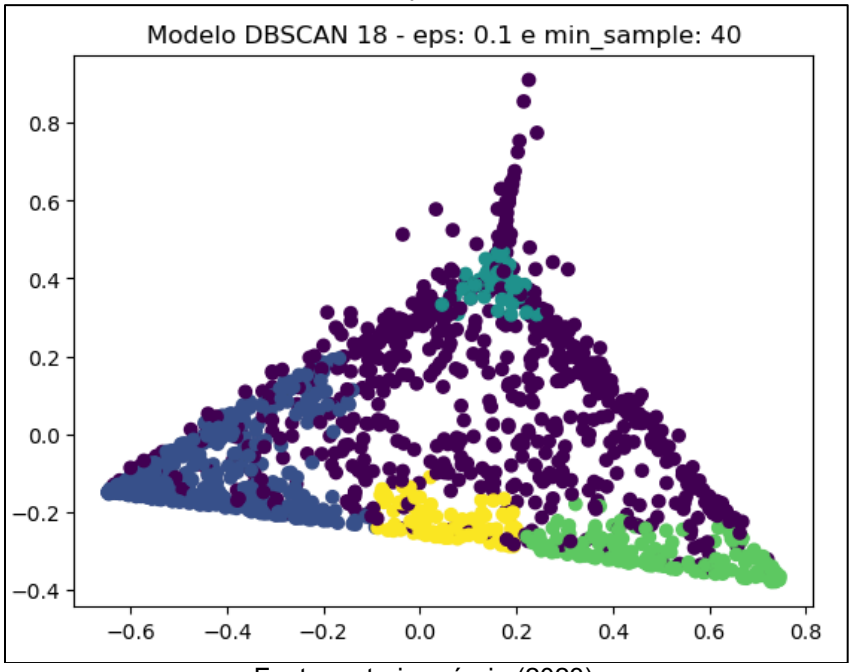


Fonte: autoria própria (2023)

O objetivo desta abordagem foi testar cada conjunto de parâmetros analisando seus resultados até que se fosse selecionado o modelo baseado no algoritmo DBSCAN que apresentasse o melhor representado no agrupamento dos clientes segundo seus perfis de compra. De acordo com os gráficos apresentados na Figura 25, alguns modelos foram descartados imediatamente por não corresponderem a um cenário que representasse a situação real, isso porque alguns cluster apresentaram um número de amostras insignificantes e um único *cluster* com

quase todas as amostras. Os modelos descartados foram os modelos de número 1, 2, 3, 4, 5, 7, 8, e 12. Considerando a sobreposição dos *clusters*, os modelos 9, 13, 14, 17 e 20 também foram descartados. Por fim, restaram os modelos de número 6, 10, 11, 15, 16, 18 e 20 que foram validados pelo especialista do negócio, onde, segundo sua percepção, o modelo 18 apresentou melhor resultado. Este modelo é apresentado na Figura 26.

Figura 26 - Modelo 18 selecionado pelo especialista de negócio da empresa



Fonte: autoria própria (2023)

4 AVALIAÇÃO

Como mencionado no item 3.4.2, os critérios de avaliação dos algoritmos de *clustering* consideram métricas e métodos diferentes dos que são utilizados pelos algoritmos de aprendizagem supervisionada. Neste caso, conforme proposto na seção supracitada, a avaliação dos algoritmos objeto deste trabalho, feita a partir dos modelos criados, considerou duas abordagens: (i) as métricas de valores absolutos, formadas pelo número de *clusters* e o coeficiente de silhueta; e (ii) os princípios de homogeneidade e heterogeneidade.

Em relação à primeira abordagem, a Tabela 9 apresenta os resultados dos critérios de avaliação relacionados aos números de *clusters* e coeficiente de silhueta de cada um dos modelos criados.

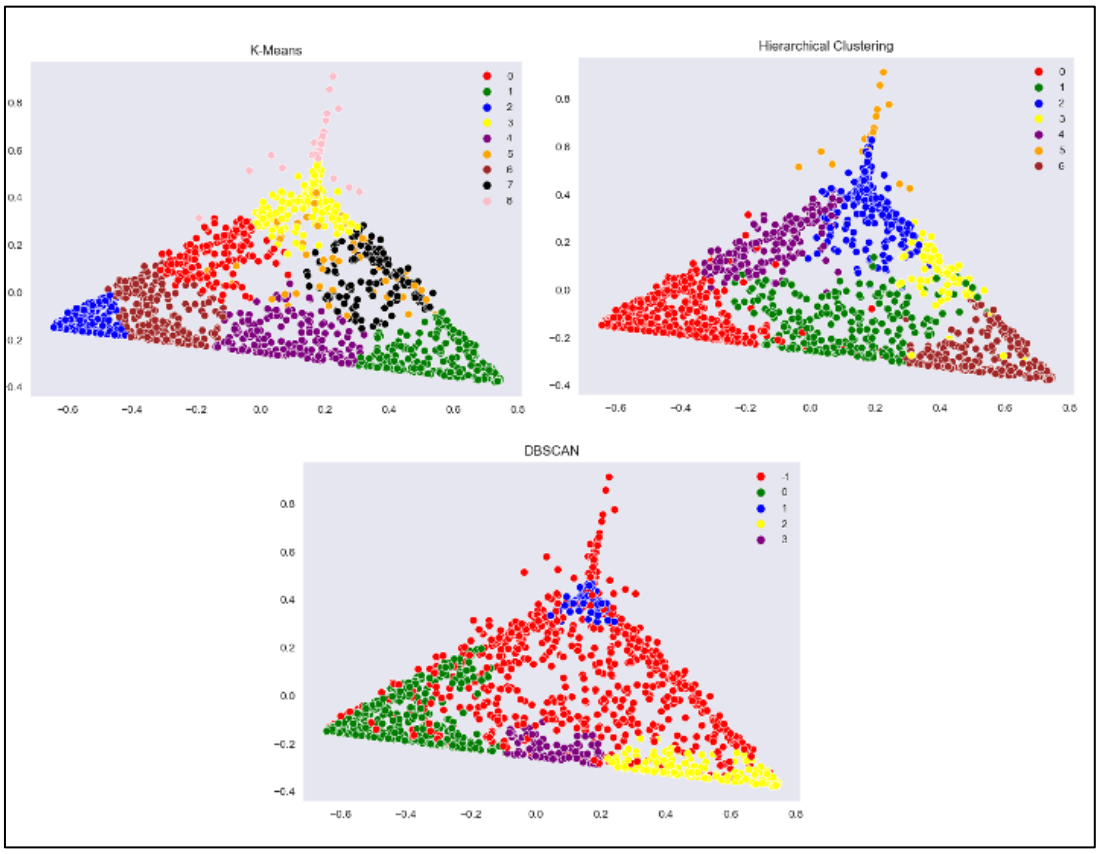
Tabela 9 – Número de clusters e coeficiente de silhueta dos modelos K-*Means*, *Hierarchical Clustering* e DBSCAN

Modelo	**Número de *clusters***	**Coeficiente de Silhueta**
K-Means	9	0,3423
Hierarchical Clustering	7	0,3438
DBSCAN	5	0,1408

Fonte: autoria própria (2023)

A título de contextualização, o manual do *Scikit-Learn* (2007) informa que o coeficiente de silhueta pode variar entre -1 a 1, sendo que os valores próximos a -1 são considerados piores e os valores próximos a 1 são os que apresentam os melhores resultados. Já os valores próximos de 0 estão relacionados a *clusters* sobrepostos. Conforme apresenta a Tabela 9, nota-se que os coeficientes de silhueta dos modelos *K-Means* e *Hierarchical Clustering* apresentaram resultados de 0,3424 e 0,3438 respectivamente, considerados resultados satisfatórios, apesar de não estarem tão próximos de 1, mas estão suficientemente distantes de 0. Em relação ao modelo DBSCAN, o resultado não foi tão satisfatório, pelo fato de o coeficiente de silhueta apresentar o valor de 0,1408, ou seja, mais próximo de 0, indicando sobreposição dos *clusters*. A Figura 27 apresenta estas informações de forma mais clara, onde as amostras do grupo -1 do DBSCAN se misturam com o grupo 0, caracterizando uma sobreposição de *clusters*. Já os *scatters plots* dos modelos *K-Means* e *Hierarchical Clustering* mostram os *clusters* mais homogêneos.

Figura 27 - Comparativo dos clusters formados pelos modelos *K-Means, Hierarchical Clustering* e DBSCAN



Fonte: autoria própria (2023)

Ao analisar os clusters formados pelos modelos *K-Means* e *Hierarchical Clustering* observa-se bastante semelhança entre eles. Primeiro em relação ao número de clusters. O modelo *K-Mean* apresenta 9 *clusters* e o *Hierarchical Clustering* 7. Ao compará-los percebe-se que os *clusters* 0 e 3 do modelo *Hierarchical Clustering* foram subdivididos nos *clusters* 2-6 e 5-7 respectivamente no modelo *K-Mean*. Mas em relação ao *K-Means* observa-se uma sobreposição dos *clusters* 5 e 7, o que não é observado no modelo *Hierarchical Clustering*.

Ainda em relação aos *clusters*, o modelo DBSCAN formou 5 grupos, o que não representa um número muito distante do *K-Mean* e do *Hierarchical Clustering*. Entretanto, o *cluster* rotulado com o valor -1 do modelo DBSCAN, possui 674 amostras. O ponto de atenção é que, segundo Scikit-Learn (2007), o *cluster* rotulado como -1 correspondem aos ruídos, ou seja, valores incomuns. Mas como 45% do *dataset* pode ser considerado como ruído? Percebe-se também que este *cluster* rotulado como -1 no DBSCAN foi dividio em 4 *clusters* tanto no modelo *K-Means*

quanto no modelo *Hierarchical Clustering*. Para finalizar, nota-se que esse *cluster* do DBSCAN não é denso comparando-o com outros *clusters*.

Considerando os critérios de heterogeneidade e homogeneidade, a Tabela 10 apresenta os resultados desta avaliação para cada modelo.

Tabela 10 - Resultado da avaliação de homogeneidade e heterogeneidade dos modelos

Modelo	**Homogeneidade**	**Heterogeneidade**
K-Mean	SIM	SIM
Hierarchical Clustering	SIM	SIM
DBSCAN	SIM	NÃO

Fonte: autoria própria (2023)

Para obtenção dos resultados relacionados à homogeneidade dos modelos foram selecionados 4 *clusters* de cada modelo, chegando a um total de 12 *clusters*. Para cada 4 *clusters* foram selecionadas, de forma aleatória, 5 amostras. Em seguida, para cada grupo de 5 amostras pertencente a um *cluster* foi realiza a comparação de suas *features* buscando comprovar a similaridade entre elas. O modelo foi considerando homogêneo uma vez que todas as amostras de todos os seus *clusters* foram também consideradas similares. Neste critério todos os modelos foram considerados homogêneos.

Em relação a avaliação de heterogeneidade entre os *clusters*, para cada modelo foi selecionada 1 amostra de cada *clusters*. Portanto, para o modelo *K-Means* foram selecionadas 9 amostras, para o modelo *Hierarchical Clustering* foram selecionadas 7 amostras e para o DBSCAN foram selecionadas 5 amostras. Para cada grupo de amostras selecionadas foi avaliado o nível de dissimilaridade entre as amostras. Caso todas as amostras fossem dissimilares entre si, o modelo foi considerado heterogêneo. Conforme apresentado na Tabela 10 somente o modelo DBSCAN não foi considerado heterogêneo.

5 CONCLUSÃO

O objetivo de pesquisa deste trabalho foi avaliar, por meio de pesquisa exploratória, a eficácia dos algoritmos de *clustering* na segmentação dos clientes de uma indústria de processamento de proteína animal com base em seus históricos de compra. Neste sentido, a conclusão deste trabalho apoia-se nas seguintes avaliações:

- A efetividade das técnicas de aprendizagem de máquina não supervisionadas utilizadas na solução do problema proposto.
- A eficácia dos modelos construídos a partir dos algoritmos *K-Means*, *Hierarchical Clustering* e DBSCAN com base no estudo de caso proposto.
- A qualidade dos dados e as características dos algoritmos como fatores de impacto nos resultados dos modelos.

Em relação a efetividade das técnicas de aprendizagem de máquina não supervisionadas, de acordo com os resultados apresentados na etapa de avaliação dos modelos, foi possível observar que os três algoritmos, o *K-Means, Hierarchical Clustering* e DBSCAN, foram capazes de encontrar os padrões de similaridade entre as variáveis de cada amostra, permitindo que cada cliente fosse rotulado de acordo com o padrão de similaridade encontrado. Desse modo, conclui-se que todos os algoritmos se demonstraram efetivos quanto a segmentação dos clientes considerando o conjuntos de dados utilizando durante o processo de treinamento dos modelos.

Quanto à eficácia dos modelos, conclui-se que o modelo *Hierarchical Clustering* foi o mais eficaz. Essa decisão está amparada em dois critérios abordados durante a fase de avaliação dos modelos. O primeiro refere-se ao coeficiente de silhueta. Como consta na seção 4 deste trabalho, o modelo *Hierarchical Clustering* apresentou o valor de 0,3423, acima dos modelos *K-Means* e DBSCAN, embora o coeficiente do modelo *K-Means* esteja bem próximo do *Hierarchical Clustering*. O segundo critério diz respeito às avaliações de homogeneidade e heterogeneidade conforme mencionado na fase de avaliação dos modelos, esses critérios correspondem aos aspectos mais intuitivos de validação dos modelos, nos quais são considerados o grau de similaridade das amostras dentro dos *clusters* e o grau de dissimilaridade das amostras entre os *clusters*. Nesse aspecto, os modelos *Hierarchical Clustering* mais uma vez apresentaram o melhor resultado, seguido pelo

K-Means que apresentou uma sobreposição dos clusters 5 e 7. O modelo DBSCAN não apresentou um resultado satisfatório em relação a heterogeneidade, justificado pela sobreposição de vários *clusters* e baixa densidade no *cluster* que possui o maior número de amostras.

Em relação à qualidade dos dados e às características dos algoritmos, conclui-se que ambos os aspectos impactam diretamente no desempenho dos modelos. Esta relação foi comprovada durante o processo de treinamento dos modelos, onde foi possível observar que a cada ajuste feito na etapa de preparação e transformação dos dados repercutia em melhores resultados dos modelos, sobretudo na formação dos *clusters*. Quanto às características dos algoritmos, constatou-se que os algoritmos *Hierarchical Clustering* e *K-Means* possuem características que melhor adapta ao problema proposto, em virtude das abordagens utilizadas pelos algoritmos na formação de seus clusters. O mesmo não aconteceu com o DBSCAN justamente por utilizar uma estratégia diferente, baseada na densidade de suas amostras. De acordo com Birant e Kut (2007), o DBSCAN se adapta melhor a *dataset* com maior volumetria.

De forma geral, os três modelos criados conseguiram resolver o problema de segmentação dos clientes, contudo os modelos *K-Means e Hierarchical Clustering* apresentaram melhores os resultados, mas o *Hierarchical Clustering* foi o mais eficaz.

REFERÊNCIAS

ABBAS, O. Comparisons Between Data Clustering Algorithms. **International Arab Journal of Information Technology**, v. 5, n. 3, p. 320-325, 1 jul. 2008. Disponível em: https://www.researchgate.net/publication/220413756. Acesso em: 9 jan. 2023.

AFIOUNI, R. Organizational learning in the rise of machine learning. *In:* **40th International Conference on Information Systems**, Munich, ed. 40, 12 dez. 2019. Disponível em: https://aisel.aisnet.org/icis2019/business_models/business_models/2. Acesso em: 3 jan. 2023.

BAGIROV, A.; ALIGULIYEV, R.; SULTANOVA, N. Finding compact and well-separated clusters: Clustering using silhouette coefficients. **Pattern Recognition**, v. 135, 10 mar. 2023. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0031320322006239>. Acessado em: 09 mai. 2023.

BAKSHI, K; BAKSHI, K. Considerations for artificial intelligence and machine learning: Approaches and use cases. **The international IEEE Aerospace Conference**, Big Sky, Montana, US, ano 2018, ed. 39, p.1-9, 28 jun. 2018. DOI 10.1109/AERO.2018.8396488. Disponível em: https://ieeexplore.ieee.org/document/8396488. Acesso em: 4 jan. 2023.

BENABDELLAH, A; BENGHABRIT, A.; BOUHADDOU, I. A survey of clustering algorithms for an industrial context. **Procedia Computer Science**, v. 148, p. 291-302, 2019. Disponível em: https://www.sciencedirect.com/science/article/pii/S1877050919300225. Acesso em: 17 nov. 2022.

BENBYA, Hind; PACHIDI, Stella; JAVENPAA, Sirkka. Artificial Intelligence in Organizations: Implications for Information System Research. **Journal of Association for Information Systems**, v. 22, n. 2, 9 mar. 2021. Special Section, p. 281-252. DOI 10.17705/1jais.00662. Disponível em: https://aisel.aisnet.org/jais/vol22/iss2/10. Acesso em: 12 dez. 2022.

BIARNES, A. Gaussian Mixture Models and Expectation-Maximization: A full explanation. **Towards data Science**, 2020. Disponível em: https://towardsdatascience.com/gaussian-mixture-models-and-expectation-maximization-a-full-explanation-50fa94111ddd. Acesso em: 10 jan 2023.

BIRANT, D.; KUT, A. ST-DBSCAN: An algorithm for clustering spatial–temporal data. **Data & Knowledge Engineering**, v. 60, n. 1, p. 208-211, 13 mar. 2007. DOI https://doi.org/10.1016/j.datak.2006.01.013. Disponível em: https://www.sciencedirect.com/science/article/pii/S0169023X06000218. Acesso em: 16 dez. 2022.

BORGES, A. F. S. *et al*. The strategic use of artificial intelligence in the digital era: Systematic literature review and future research directions. **International Journal of Information Management, Elsevier Ltd**, v. 57, 2021. DOI 10.1016/j.ijinfomgt.2020.102225. Disponível em: https://www-scopus.ez10.periodicos.capes.gov.br/record/display.uri?eid=2-s2.0-85090862788&origin=resultslist&zone=contextBox. Acesso em: 20 nov. 2021.

BRUCE, P.; BRUCE, A. **Estatística prática para cientistas de dados**: 50 conceitos essenciais. Altas Books, 2019. 378 p. ISBN 978-85-508-1300-4.

BUITINCK, L. *et al*. API design for machine learning software: experiences from the sckit-learn project. *In:* **European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases**, 2013. Disponível em: < https://arxiv.org/abs/1309.0238 >. Acesso em: 02 abr 2023.

CHAPMAN, P. *et al*. **CRISP-DM 1.0**: Step-by-step data mining guide. 1. ed. Chicago, IL: IBM Corporation, 2000. 73 p. v. 1. Disponível em: https://web.archive.org/web/20220401041957/https://www.the-modeling-agency.com/crisp-dm.pdf. Acesso em: 28 nov. 2022.

CLEYPAUL, D. CRISP-DS: Cyclic Methodology for Data Science Projects. **Magrathea**, 31 out. 2021. Disponível em: https://blog.magrathealabs.com/crisp-ds-cyclic-methodology-for-data-science-projects-10c7d00fbc85. Acesso em: 9 jan. 2023.

DINH, D.; FUJINAMI, T.; HUYNH, V. Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient. *In:* **International Symposium on Knowledge and Systems Sciences**, v. 1103 p. 1-7, 2019. Disponível em: <https://link.springer.com/chapter/10.1007/978-981-15-1209-4_1#citeas>. Acesso em: 10 mai. 2023.

DOBILAS, S. HAC: Hierarchical Agglomerative Clustering - Is it better than K-Means. **Towards Data Science**, 9 maio 2021. Disponível em: https://towardsdatascience.com/hac-hierarchical-agglomerative-clustering-is-it-better-than-k-means-4ff6f459e390. Acesso em: 11 maio 2023.

ENHOLM, I.M. *et al*. Artificial Intelligence and Business Value: a literature review. **Information Systems Frontiers**, v. 24, p. 1709-1734, 2022. Disponível em: < https://link.springer.com/article/10.1007/s10796-021-10186-w#citeas>. Acesso em: 18 nov. 2022.

FOLEY, D. Gaussian Mixture Models. **Towards data Science**, 2019. Disponível em: https://towardsdatascience.com/gaussian-mixture-modelling-gmm-833c88587c7f. Acesso em: 10 jan 2023.

GE, Z. *et al*. Data Mining and Analytics in process industry: The role of machine learning. **IEEE Access**, v.5, p. 20590-20616, 19 set. 2017. DOI 10.1109/ACCESS.2017.2756872. Disponível em: https://ieeexplore.ieee.org/document/8051033. Acesso em: 12 jan. 2023.

GÉRON, A. **Mãos à obra aprendizado de máquina com Scikit-Learn e TensorFlow**: Conceitos, ferramentas e técnicas para construção de sistemas inteligentes. Rio de Janeiro: Altas Books, 2019. 576 p. ISBN 978-85-508-0902-1.

HEANLNEIN, M.; KAPLAN, A. A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence. **California Management Review**, v. 61, n. 4, p. SI, 8 ago. 2019. DOI https://doi-org.ez10.periodicos.capes.gov.br/10.1177/0008125619864925. Disponível em: https://journals-sagepub-com.ez10.periodicos.capes.gov.br/doi/10.1177/0008125619864925. Acesso em: 5 dez. 2022.

KHANDELWAL, R. DBSCAN - Desity-base spatial clustering for application with noise: The density-based unsupervised clustering algorithm robust to outliers. **The Startuo**, 30 jan. 2023. Disponível em: https://medium.com/swlh/dbscan-density-based-spatial-clustering-for-applications-with-noise-476d95c1f14a. Acesso em: 1 maio 2023.

LI, Y. Deep Reinforcement Learning: An Overview. **arXiv**. 25 jan. 2017. Disponível em: https://arxiv.org/abs/1701.07274. Acesso em: 3 jan. 2023.

LOUCKS, J. *et al*. **Future in the balance**? How countries are pursuing an AI advantage. Insights from Deloitte's State of AI in the Enterprise, 2ª Edition survey [online]. Deloitte, 2019. Disponível em: <https://www2.deloitte.com/us/en/insights/focus/cognitive-technologies/ai-investment-by-country.html. Acesso em: 15 de nov. 2022.

LUDERMIR, T. B. Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências. **Estudos Avançados**, v. 35, n. 101, p. 85-94, 2021. Disponível em: < https://www.revistas.usp.br/eav/article/view/185035>. Acesso em: 17 nov. 2022.

LV, Y. *et al*. A New Outilier Detection Method Based on Machine Learning. *In:* **2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)**. 2019. p. 1-7. DOI 10.1109/ICSIDP47821.2019.9173217. Disponível em: <https://ieeexplore-ieee-org.ez10.periodicos.capes.gov.br/document/9173217>. Acesso em: 20 abr. 2023.

NAMAGANDA-KIYIMBA, J.; MUTALE, J. An Optimal Rural Community PV Microgrid Design Using Mixed Integer Linear Programming and DBSCAN Approach. **SAIEE ARJ**, v. 111, n. 3, p. 111-

119, 2020. Disponível em: <http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S1991-16962020000300004&lng=en&nrm=iso>. Acesso em: 07 abr. 2023.

NIAKSU, O. CRISP Data Mining Methodology Extension for Medical Domain. Baltic **Jornal Modern Computing**, Vilnius, v. 3, n. 2, p. 92-109, 2015. Disponível em: https://www.researchgate.net/publication/277775478_CRISP_Data_Mining_Methodology_Extension_for_Medical_Domain. Acesso em: 8 fev. 2023.

MCKINNEY, W. **Python para análise de dados**: Tratamento de dados com Pandas, Numpy e Jupyter. 3. ed. São Paulo: Novatec, 2023. 784 p. ISBN 978-85-7522-841-8

PRODANOV, C. C.; FREITAS, E. C. **Metodologia do Trabalho Científico**: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico. 2ª. ed. Novo Hamburgo: FEEVALE, 2013. 275 p. ISBN 978-85-7717-358-3.

PROVOST, F.; FAWCETT, T. **Data science para negócio**. O que você precisa saber sobre mineração de dados e pensamento analítico de dados. 1ª. ed. Rio de Janeiro: Altas Books, 2016.

RACHWAT, A *et al*. Determining the quality of a dataset in clustering terms. **Applied Sciences**, v. 13, n. 2942, ed. 5, 24 fev. 2023. DOI https://doi.org/10.3390/app13052942. Disponível em: https://www.mdpi.com/2076-3417/13/5/2942. Acesso em: 9 maio 2023.

RIAHI, Y.; RIAHI, S. Big Data and Big Data Analytics: Concepts, types and technologies. **International Journal of Research and Engineering**, v. 5, n. 9, p. 524-528, 2018. Disponível em: < https://www.researchgate.net/publication/328783489_Big_Data_and_Big_Data_Analytics_Concepts_Types_and_Technologies/link/5be2b85da6fdcc3a8dc40690/download >. Acesso em: 12 nov. 2022.

ROCHA, I. F.; KISSIMOTO, K. O. Artificial intelligence and internet of things adoption in operations management: Barriers and benefits. **Revista de Administração Mackenzie**, v. 23, n. RAM, Rev. Adm. Mackenzie, 2022. Disponível em: <https://www.scielo.br/j/ram/a/mGpm3mhb5vZ5VLPbmmfYBwt/citation/?lang=pt#> Acesso em: 18 nov. 2022.

SARAVANAN, R.; SUJATHA, P. State of Art Techniques on Machine Learning Algorithms: A perpective of supervised learning approches in data classification. *In:* **2018 Second Internation Conference on Intelligent Computing and Control Systems**, Madurai, India, ano 2018, p. 945-949, 15 jun. 2018. DOI 10.1109/ICCONS.2018.8663155. Disponível em: https://doi-org.ez10.periodicos.capes.gov.br/10.1109/ICCONS.2018.8663155. Acesso em: 2 jan. 2023.

SAXENA, A. *et al.* A review of clustering techniques and developments. **Neurocomputing,** v. 267, p. 664-681, 2017. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0925231217311815?via%3Dihub. Acesso em: 03 marc. 2023.

SCIKIT-LEARN. **Machine Learning in Python**, 2007. Disponível em: < https://scikit-learn.org/stable/index.html >. Acesso em: 09 mai. 2023.

SHARMA, A. *et al*. Supervised and unsupervised prediction application of machine learning. *In*: **INTERNATIONAL CONFERENCE ON CYBER RESILIENCE**, 2022, Dubai. Anais eletrônicos […] Dubai: IEEE, 2022. p. 1-5. Disponível em: https://ieeexplore.ieee.org/document/9996063. Acesso em: 5 dez. 2022.

SICHMAN, J. S. Inteligência Artificial e sociedade: avanços e riscos. **Estudos Avançados**, v. 35, n. 101, p. 37-50, 2021. DOI https://doi.org/10.1590/s0103-4014.2021.35101.004. Disponível em: http://c4ai.inova.usp.br/. Acesso em: 2 dez. 2022.

TRIPATHI, S.; BHARDWAJ, A.; ESWARAN, P. Approaches to Clustering in Customer Segmentation. International **Journal of Engineering & Technology**, v. 7, n. 312, 20 jul. 2018. 12, p. 802-807. DOI 10.14419/ijet.v7i3.12.16505. Disponível em: https://www.sciencepubco.com/index.php/ijet/article/view/16505. Acesso em: 13 jan. 2023.

USAMA, M. *et al*. Unsupervised Machine Learning for Network: Techniques, Applications and Research Challanges. **IEEE Access**, v. 7, p. 65579-65615, 14 maio 2019. DOI 10.1109/ACCESS.2019.2916648. Disponível em: http://ieeexplore.ieee.org/document/8713992. Acesso em: 10 jan. 2023.